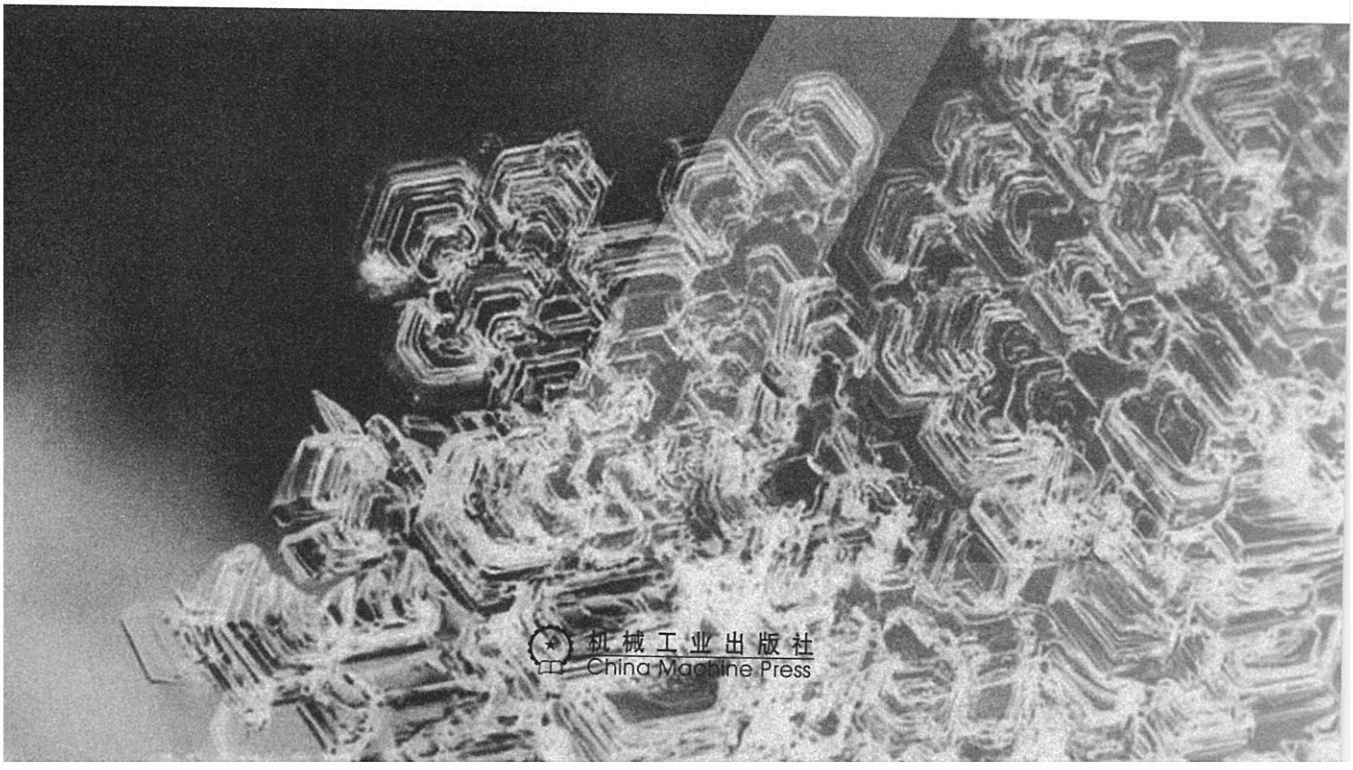
云计算与虚拟化技术丛书

Essential Virtual SAN
Administrator's Guide to VMware VSAN

VMware Virtual SAN 权威指南

[美] Cormac Hogan Duncan Epping 著 徐炯 译
VMware中国研发中心 审校



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

VMware Virtual SAN 权威指南 /（美）霍根（Hogan，C.），（美）埃平（Epping，D.）著；徐炯译 .—北京：机械工业出版社，2014.9
（云计算与虚拟化技术丛书）

ISBN 978-7-111-48023-5

I. V… II. ①霍… ②埃… ③徐… III. 虚拟处理机－指南 IV. TP338-62

中国版本图书馆 CIP 数据核字（2014）第 211744 号

本书版权登记号：图字：01-2014-5107

VMware Virtual SAN 权威指南

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）
责任编辑：关 敏
责任校对：殷虹
印 刷：北京市荣盛彩色印刷有限公司
版 次：2014 年 10 月第 1 版第 1 次印刷
开 本：186mm × 240mm 1/16
印 张：14.75
书 号：ISBN 978-7-111-48023-5
定 价：59.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991 88361066
投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259
读者信箱：hzjsj@hzbook.com

Foreword 推荐序

Virtual SAN 是一个变革性的软件定义存储（SDS）产品，它是基于服务器端存储的共享分布式对象存储系统。分布式存储系统在业界和学术界已经发展多年，但是基于 Server 端存储的共享分布式存储产品发展缓慢。高速闪存和高速网络技术的发展与推广使我们能够重新审视传统 SAN 存储和服务器端存储的性能价格优劣及不同的应用场景。Virtual SAN 的成功在于挑战传统，勇于创新。

类似的分布式对象存储系统产品不少（比如 Ceph——我在 UCSC 读博士期间同师门学友的博士论文的产物），但是我认为真正能够荣登企业级 Hypervisor-converged 分布式对象存储产品的目前只有 Virtual SAN 一个。

VMware Virtual SAN 产品和研发团队就像个初创公司。从研发开始到产品正式发布整整四年，中间经历难熬的多个开发、测试和纠错周期，几经反复，一言难尽。曾经有一段时间，为了能够尽早把高质量的产品推出给客户，研发团队 100 多个日夜连续加班赶进度。Virtual SAN 中国研发团队直接参与了 Virtual SAN 1.0 核心技术的研发，并经历了产品发布前那些繁忙和揪心的日子。令人欣慰的是，结果是好的。Virtual SAN 发布时受欢迎程度就像多年前 ESX 刚刚推出时一样，在前三个月已经拥有 300 多个各行各业的客户（客户分布在美国和中国等国家）。在 VMworld 2014 大会上，Virtual SAN 也受到客户和合作伙伴的倾力追捧。客户最喜欢 Virtual SAN 的简单易用（Simplicity）。这正是产品和研发团队在设计产品时极力追求的简单无上限（No Limits ——这里指无上限不是无底线）目标。

在旧金山参加 VMworld 2014 大会期间，我特意约两位作者 Cormac Hogan 和 Duncan Epping 见面，相谈甚欢。两位作者是 VMware 公司著名的技术博主，其中 Cormac 是 VMware 存储架构师，Duncan 是 VMware 云产品的首席架构师。如果没记错的话，这是 Duncan 第 7 次写书（Cormac 是第一次写对外出版的书）。他们告诉我，这次写书与以往不同，因为这次

写书是基于他们在参与产品开发测试讨论和产品推广过程中（通过写大量的博文）使用 Virtual SAN 的实践——实践是检验好产品的唯一标准。作者在接触和推广试用 Virtual SAN 的过程中，觉得 Virtual SAN 技术太酷太好用了，自然而然就想到要写一本书来为读者介绍这中间的奥秘。

我通读了本书的英文原版书。全书通俗易懂、图片丰富，作者精通存储和分布式系统，对 VMware 和其他存储产品也是造诣颇深。作者不是底层工程师，也没有读过 Virtual SAN 产品的源代码，但是作者在书中对产品和技术的描述及把握适度，没有把本书写成一本纯产品手册或者代码层次的技术书。每一个技术章节，特别是中间的体系结构和技术细节（包括很多数据流图），都经过与 Virtual SAN 资深架构师和首席工程师讨论才拍板定调。

本书中文版的一个特点是它不是简单的“直译”。我见过好多翻译过来的书（包括当前较热门的 SDN、OpenStack 等专题），都是硬邦邦的直译，很多术语晦涩难懂，让人觉得还不如读英文原版。本书译者徐炯对存储有直接使用经验，并对翻译精益求精。在翻译之前和期间，他一直与我们 Virtual SAN 中国研发团队保持及时的沟通，确保书中内容在技术术语的使用上和语意上都能与我们达成一致的意见。

我认为这是一本当下在软件定义存储领域难得的好书。不管你是分布式对象存储新手还是专家，本书都会让你受益匪浅。

成功属于那些敢于承担风险和勇于创新的人！

林才学　博士

VMware 中国研发中心高级研发经理，Virtual SAN 中国研发团队负责人

2014 年 9 月

The Translator's Words 译者序

两周前当本书的翻译接近尾声的时候，2014 年度的 VMworld 大会在美国旧金山开幕了，会上传来了令人震惊的消息，VMware 公司推出了超融合基础架构 EVO:RAIL 和 EVO:RACK。

“融合”这个词这些年来越来越热。是呀，原本各自为政甚至是井水不犯河水的网络、服务器和存储开始相互渗透合并，其速度之快趋势之猛以至于连“融合”这个词本身都 hold 不住了，非得要搬出“超融合”才能压得住阵脚。VMware 公司的 Virtual SAN 就是 VMware 公司的这个超融合架构的核心技术。

几个月前接手这本关于 VMware Virutal SAN（VSAN）技术的最新图书的翻译工作的时候，我就已经深深感受到这种技术给我们带来的震撼。作为一家纳斯达克上市的美国公司在中国工厂的 IT 技术经理，我也同时负责中国工厂的基础架构和数据中心的一些架构设计工作。去年（2013 年）我们才部署完毕 FlexPod，这是 Cisco+NetApp+VMware 合作推出的统一计算的架构，它非常巧妙地利用思科 UCS 技术将服务器的所有配置都剥离出来放在配置文件中，使得服务器纯粹成为提供计算资源和内存资源的盒子。但是 FlexPod 仍然利用了 NetApp 的共享的集中式存储，就好像 VCE 还是需要利用 EMC 的集中式存储一样。集中式存储的最大缺点就是贵，而 VSAN 利用服务器的本地存储和闪存加速技术，带给用户高性能的同时将存储成本降低到了普通机架式服务器硬盘的价位。这种低价的冲击影响是巨大的，甚至是致命的。

正如 Charles Fan 在为本书写序时说的那样，这对存储企业来说“是一种破坏性的创新”。随着企业内几乎所有的应用包括关键业务应用都随着服务器虚拟化集中到了 VMware 平台上，企业级集中式存储的应用场景也就慢慢集中到为 VMware 提供存储。此时，完全可以用 VSAN 来取代集中式存储。几年之后，目前这种中小型企业级集中式存储甚至可能会因此而被市场所彻底摒弃。

非常高兴能有机会对一本介绍前沿技术的书籍进行翻译。接到这项任务的时候，本书的英文版本还没有完全定稿（英文版于今年 8 月出版），于是我也获得了很多与本书作者 Cormac

Hogan 和 Duncan Epping 沟通的机会。这两位作者都是虚拟化领域的大拿，各自拥有自己的博客网站，发表过很多极有价值的博客文章。多年前我就是他们博客的订阅者。能有机会和自己的偶像合作非常令人激动和愉快。翻译本书的过程也是我学习的过程，在这个过程中，两位作者给予了我很多支持，在此表示深深的感谢。

还要感谢华章公司的关敏编辑和王春华编辑，她们是把这本书交到我手里的红娘，不仅如此，她们的认真仔细帮助我纠正了很多错误。此外，我也获得了 VMware 中国研发中心林才学博士及其领导下的开发团队的技术支持，在此一并谢过。

翻译本书的过程既是愉快的也是痛苦的，愉快是因为新知识的收获，痛苦是因为本书的翻译是在繁忙的工作之余利用挤出来的个人时间完成的，时间紧张而我又是拖延症患者，因此常常被小鞭子逼着抽着去做这项文字工作。所以，最后的最后，要深深地感谢我的妻子刘峥嵘同学，不是你的小鞭子时时敲打，我是不可能及时完成这项任务的。

徐炯

2014 年 9 月 8 日中秋夜

Foreword 序 一

2012年早些时候我加入了VMware公司，当时我很荣幸地受命领军交付下一代的vSphere——我们的旗舰产品，这个荣誉让我既羞愧又惊喜。进入角色几个月后，存储组加入团队中，我很荣幸能和这样一群敬业的工程师紧密合作。他们正在做的东西非常特别——相信这将是存储历史上一个重要的转折点。

我们开始打造一款分布式的可容错的专门为虚拟环境优化的存储系统。我们的目标是构建一个具备所有共享存储品质（弹性、性能、可扩展性等）的产品，但是这个产品不需要特殊的硬件也不需要专门的软件来维护，可以直接运行在x86服务器上。只需要插入硬盘和SSD，vSphere会搞定剩下的一切。加上基于策略的管理框架和新的运营模型，存储管理将变得前所未有地简单。

我们遇到了很多困难——就像所有长期的软件项目一样，长夜漫漫、士气低落、主次纠葛、计划多变。尽管如此，我们坚持下来了。2013年6月特别痛苦，当时团队已经准备好发布vSphere 5.5了，但是我不得不告诉团队还不能交付VSAN，相反，在宣布产品“可以商用”之前，它们必须经过更广泛的公开beta测试和更多严苛的实验。

风险实在太高，特别是因为这是VMware在软件定义的存储领域的一次突袭，是我们软件定义的数据中心愿景的关键组成部分。

当然，他们很失望。因为我们不能在VMworld的舞台上展示这个产品，但是大家坚持了下来。我觉得我们的选择是正确的。经过了6个月和12 000名beta测试员的不懈努力，Virtual SAN终于完成了：它可靠、成熟，做好了登台表演的准备。VSAN可以从用于小型分支办公室的最少3个节点的配置，横向扩展到数TB存储、上百万IOPS的可以支持整个企业所有存储需求的庞然大物。

我们的团队交付的是业界真正特别优秀的产品：一个完全分布式的与hypervisor无缝融合的存储架构。

VSAN 不是外挂，它是内核。

关于 VSAN 已经有了很多文章，都提到它将给存储世界带来一种全新的理念。然而本书很不一般。Duncan 和 Cormac 在整个项目过程中始终与我们的开发团队密切合作。他们不仅极度熟悉 VSAN 架构，也亲手进行过大规模的部署和管理。他们是专家。

Ben Fathi

VMware CTO

Foreword 序 二

今年的早些时候，我有幸参加了克莱顿·克里斯坦森（Clayton Christensen）的一个研讨会。他的极具创意的作品——《创新者的困境》（*Innovator's Dilemma*）是我最喜欢的商业读本之一，因此能有机会亲耳聆听克莱顿本人的深刻见解实在是太棒了。整个研讨会中我都有一个幻觉，好像房间里面没有其他人，只有克莱顿和我，而且我们一直在探讨 Virtual SAN。

讨论的主题是——VSAN 到底是一种破坏性创新还是维持性创新？

在克莱顿的书里面是这样定义的：维持性创新是一种使事物更好、更快、更强大的技术进步，可以满足客户不断增长的需求。维持性创新不需要改变当前的业务模式、业务流程或目标客户。这是大公司如何变得更大的方法。只要给它们足够的资源和客户关系，它们总能通过维持性创新战胜较小的公司。

然而，总是会发生某些技术进步超过了客户需求发展的情况。此时，创新来自于底部。这些创新通过提供一种不同的方式来解决问题。在开始的时候它们可能无法提供同样水准的特性或性能，但是它们更便宜、更简单，常常会带来更多的和不同领域的客户，有时候它们会彻底改变业务模式。这就是破坏性创新。对于现任的市场领导者来说，采纳破坏性创新极为困难。这种类型的创新会重新定义整个业界，新的领导者由此而诞生。

那么，VSAN 是一种破坏性创新还是维持性创新呢？这个问题看上去好像很傻，当然这是一种破坏性创新。它是一种极度简单的、纯粹软件化的、融合在 hypervisor 中的分布式存储解决方案，它完全和 vSphere 整合在了一起，可以运行在普通商用服务器上。它对存储的经济模型和消费模型都做出了重新定义。尽管它还缺少（就目前而言）一些典型的存储的特性和优点，但是它提供了比传统企业级存储阵列更为简单易用的品质，以更低的价格卖给一群与以往不同的存储管理员用户。因此，这是一种典型的破坏性创新，就好像傻瓜相机一样。比起“真正的”相机，傻瓜相机开始只有很少的功能，但是它们却非常简单，目标客户是完全不同

的一群用户。你猜后来怎样了？很快傻瓜相机的用户数量就超过了传统相机。

那为什么会有这样的问题？作为一款存储产品，是的，VSAN 毫无疑问是一款革命性的产品，它会颠覆整个存储工业并将其推向一个新的时代。然而，如果换一个角度来看，作为 vSphere 服务器虚拟化平台的自然延伸，进化成软件定义的数据中心，这是一种维持性创新。VSAN 仍然卖给同样的 vSphere 客户，并帮助他们实现更多的功能。它将服务器环境扩展成一个融合性的基础架构，将 vSphere 抽象层通过基于策略的自动化从计算领域扩展到了存储领域。

因此，我们有一半胜算掌握在我们自己手里。这是一个维持性创新和破坏性创新的组合。一方面它构筑在我们的 VMware 原生的 hypervisor 平台之上，扩展了提供给客户的价值，另一方面它同时是一种将要重塑存储工业的破坏性创新，这个产品对存储会造成的影响就像当年 vSphere 对服务器造成的一样。

VSAN 产品是整个 VSAN 产品团队过去 4 年辛勤劳动的结果。这个团队不仅仅包括核心架构师、程序员、测试员和产品经理，Duncan 和 Cormac 也是团队的 2 个关键人物。他们带来了真实世界的经验和客户的想法，他俩也是将我们的想法带到全世界的两个最有力的声音。我非常高兴他们能及时推出这本书，希望你们能和我一样觉得它非常有用。VSAN 是一个非常特别的产品，它会给整个业界带来持续的影响。欢迎你加入我们这个令人兴奋的旅程中。

Charles Fan

VMware 研发、存储和可用性高级副总裁

Preface 前 言

说到虚拟化及其依赖的底层基础架构，经常会提起一个组件——存储。原因相当简单：在很多环境中，存储是痛点。尽管存储市场已经因为闪存技术的引入发生了变化，很多传统的存储问题得到了缓解，但是很多机构还没能采纳这些新的架构，因而仍然会遇到挑战。

存储问题的范围包括运营上的复杂性到性能问题甚至是可用性的限制。这些问题中的大部分都起因于同样的根本问题：老旧的系统架构。这是因为大多数存储平台架构是在虚拟化技术出现之前开发出来的，而虚拟化已经改变了使用这些共享存储平台的方法。

某种程度上，可以说是虚拟化迫使存储业界去寻找新的方法来构建存储系统。不再是通过单台服务器连接到单台存储设备（也称为逻辑单元或简写为 LUN），虚拟化通常由一台（或多台）物理服务器承载很多虚拟机连接到一个或多个存储设备上。这不仅仅增加了这些存储系统的负载，也改变了工作负载的模式并增加了对总容量的需求。

可以想象，对于大多数存储管理员来说，这要求思考模式的大改变。LUN 的大小应该是多少？对性能有什么要求？最终需要多少个磁盘？这些 LUN 将提供何种数据服务？虚拟机将存放在什么地方？不仅思考模式要改变，而且要求和其他 IT 团队协力合作。过去服务器管理员、网络管理员和存储管理员都可以活在他们自己的独立的小天地内，现在他们需要相互沟通并齐心协力才能保证他们构建的平台的高可用性。在过去，一个错误（例如错误配置或过低的置备）只会影响一台服务器，现在则会影响很多虚拟机。

当虚拟化出现时，我们对于如何运营和构建 IT 基础架构的思维曾经发生过集体性的根本变化。如今集体性的转变再次发生，这一次是由软件定义的网络和软件定义的存储引起的。但我们不应该再重复历史，重复那些在虚拟化首次出现时我们曾经犯过的错误。我们应该坦率而开放地和数据中心管理员们讨论这个问题，并一起迎接数据中心架构和运营的革命。

写作本书的动机

我俩很早就都参与进了 VSAN 的产品开发，并立刻意识到这是每个人都会提起的产品，人们会想要了解更多的东西。在各种美妙的来来往往的交谈和探讨中，我们认识到这些信息不曾被记录下来过。考虑到我俩都是狂热的博客作者，我们决定各自独立地开始撰写文章。很快我们就攒起了大量的材料，多到不可能将它们用博客的方式都发表出来，于是我们决定合力出版一本书。初步接触后，VMware 出版社愿意出版这本书。

读者对象

本书的目标读者是和 VMware vSphere 环境相关的 IT 专业人员。你最好已经用过一阵子 VMware vSphere，或许已经参加过 vSphere 的课程，例如“安装、配置和管理”课程。本书不是一本初学者读物，但是书中提供的信息应该已经足够覆盖各种不同水平的管理员和架构师。

如何使用本书

本书分 10 章，分别如下：

- 第 1 章概要介绍了软件定义的存储和 VSAN。
- 第 2 章从物理和虚拟的角度描述了对 VSAN 安全地进行实施所需的要求。
- 第 3 章介绍了安装和配置 VSAN 的各个步骤。
- 第 4 章介绍了基于策略的存储管理。
- 第 5 章深入介绍了 VSAN 的架构细节。
- 第 6 章描述了虚拟机存储策略是如何用来简化虚拟机部署的。
- 第 7 章描述了常用的管理和维护任务的步骤。
- 第 8 章覆盖了 Virtual SAN 和其他 VMware 功能及产品之间的互操作性。
- 第 9 章提供了多个例子来介绍如何设计一个 VSAN 群集，包含了一些容量规划的练习。
- 第 10 章覆盖了各种可用于进行 VSAN 排错和监控的（命令行）工具。

致谢

我们两个作者都在 VMware 公司工作。在本书中表达的意见都是我们根据自己对产品的经验表达的个人意见。本书中的陈述不一定反映出 VMware 公司的意见和观点。

我们要感谢 Christos Karamanolis 和 Paudie O’Riordan，作为我们的技术编辑，他们始终让我们保持坦诚的态度。当然，我们要感谢 Virtual SAN 工程团队，特别要指出 2 个名字：

Christian Dickmann 和 Christos Karamanolis（再次感谢！），他们在 VSAN 方面的渊博知识和对 VSAN 的深入理解贯穿到了本书的各个角落。我们还想感谢 William Lam、Wade Holmes、Rawlinson Rivera、Simon Todd、Alan Renouf 和 Jad El-Zein 对本书提供的帮助和贡献。

最后，我们想感谢 VMware 公司的管理团队（Phil Weiss、Adam Zimman 和 Mornay van der Walt）对本书和其他项目的支持。

VSAN 加油！

Cormac Horgan 和 Duncan Epping

关于作者 *About the Authors*

Cormac Hogan 是 VMware 集成工程团队的存储架构师。他是位于爱尔兰科克市的 VMware EMEA[⊖]总部 2005 年的第一批雇员之一，曾经在 VMware 技术市场部门和支持部门任职。Cormac 撰写过很多存储相关的白皮书，并做过大量关于存储最佳实践和新特性方面的演讲。Cormac 是 CormacHogan.com 网站的站长，这是一个专注在存储和虚拟化方面的博客网站。

可以通过 twitter 关注他：@CormacJHogan。

Duncan Epping 是 VMware 研发中心的首席架构师。他负责在现有产品和功能中挖掘新的潜力，并为 VMware 寻找新的业务机会。Duncan 专门研究软件定义的存储、hypervisor 融合平台以及高可用解决方案。Duncan 是最早的 VMware 认证设计专家之一（VCDX007），他是 Yellow-Bricks.com 网站的站长和多本著作的作者，包括 VMware vSphere 群集技术深入探讨（*VMware vSphere Clustering Technical Deepdive*）系列。

可以通过 twitter 关注他：@DuncanYB。

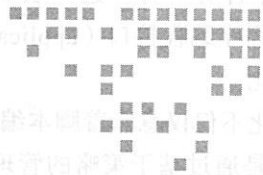
⊖ EMEA 指的是 Europe, Middle East and Africa。——译者注

About the Technical Reviewers 关于技术审校者

Christos Karamanolis 是 VMware 存储和可用性工程部的首席架构师及首要工程师。他在分布式系统、容错、存储和存储管理领域有超过 20 年的研发经验。他是 Virtual SAN 这一新的分布式存储系统和 vSphere 基于策略的存储管理栈（S-PBM，VASA）的架构师。早先他曾做过 ESX 存储栈（NFS 客户端和 vSCSI 过滤器）和容灾产品（vSphere Replication）。在 2005 年加入 VMware 之前，Christos 作为研究员在 HP Labs 待过几年，从事新一代存储产品的研发。他的职业生涯始于帝国理工学院，在那里他是一名助理教授，与此同时他还是一个独立的 IT 咨询师。他参与合作了超过 20 篇研究报告，发布在经同行评议的杂志和会议上，并拥有 24 项专利。他拥有英国伦敦大学帝国理工学院颁发的分布式计算的博士学位。

Paudie O'Riordan 是 VMware 研发部门的资深集成联络员。在此之前他在 EMC 工作（1996—2007），历任 IT 管理员、EMC 全球技术支持、公司系统工程师和 EMC 研发中心的首席软件工程师。在 VMware 的前几年，他曾任 VMware 全球服务部的高级资深技术支持工程师。他拥有 VCP 证书（VCP4）。

目 录 *Contents*



第 1 章 Chapter 1

VSAN 概述

本章将把你带入软件定义的数据中心的世界，不过我们将主要关注存储方面。本章首先讨论软件定义的数据中心的基本前提，随后深入到软件定义的存储的概念及其相关的解决方案，例如服务器存储区域网络（Server SAN）。

1.1 软件定义的数据中心

在 2012 年 VMware 的年度大会 VMworld 上，VMware 分享了对于软件定义的数据中心（software-defined datacenter，SDDC）的愿景。SDDC 是 VMware 的公有云和私有云的架构，在其中将数据中心所有的重要组成部分——计算、存储、网络以及相关的服务全部都进行虚拟化。将数据中心的各个组件虚拟化使得 IT 团队更加灵活，降低了运营的复杂性，减少了成本，并同时增加了可用性和敏捷性，最终将大大缩短把新服务投向市场的时间。

要达到这些目的，仅仅是实现所有组件本身的虚拟化是不够的，其使用的平台必须拥有以全自动的方式来安装和配置的能力。更重要的是，它应该能使你无须过多操作就能智能地管理和监控基础架构。这就是软件定义的数据中心的意义所在！就像 VMware 的高级副总裁 Raghu Raghuram 所概括的：软件定义的数据中心的精要就是“抽象化、池化和自动化”。

抽象化、池化和自动化都是通过在物理资源上引入额外的层面实现的，这个层面通常是指**虚拟化层**。我想本书大多数的读者对计算虚拟化的领军产品 VMware vSphere 都会比较熟悉，但是熟悉网络虚拟化——有时候也指软件定义的网络（software-defined network，SDN）的解决方案——的读者就可能比较少了。在这个领域 VMware 提供的解决方案叫做 NSX，这是在收购来的 Nicira 公司的解决方案的基础上构建而成的。NSX 之于网络就相当

于 vSphere 之于计算一样。这些层面不仅对物理资源进行虚拟化，还允许你将它们池化，并且提供应用程序编程接口（application programming interface, API）来允许你将所有的运营活动都自动化。

然而自动化不仅仅意味着脚本编写，例如虚拟机（及其相关联的资源）的置备自动化的一个重要环节是通过基于策略的管理来实现的。预定义的策略使你得以用快速、便捷、一致和可重复的方式来置备虚拟机。计算策略的一个例子就是定义在资源池或 vApp 容器上的资源特性。这些特性使你可以从预留（reservation）、限制（limit）和优先级（priority）等方面量化资源策略。网络策略的范围可以涵盖从安全到服务质量等各个方面。遗憾的是，存储却往往大大受限于物理存储设备提供的特性，很多时候无法满足许多客户的需求和期望。

本书将讨论 VMware 的 SDDC 的存储组件，具体来说，就是一款从 VMware vSphere 5.5 Update 1 开始才发布的、名为 Virtual SAN（VSAN）的新产品将怎样来切入这个愿景。我们将从底层的实施细节来探讨如何实施、如何将其整合到现有的平台中、如何利用其功能，以及如何进行扩容。不过，在开始之前，了解一下 VSAN 对于更宽泛的软件定义的存储来说意味着什么还是有帮助的。

1.2 软件定义的存储

软件定义的存储是一个被很多厂商广为使用甚至到了滥用地步的一个术语。因为每一家的定义都不同，所以还是让我们先来引用一下 VMware 的定义：

软件定义的存储是将工业标准服务器的存储提供出来并通过软件控制层面实现存储的自动化和池化。它将存储的置备和管理的方法简化到了极致，并利用工业标准服务器的存储大大降低了成本。（资料来源：http://cto.vmware.com/vmwares-strategy-for-software-defined-storage/。）

软件定义的存储产品是一个将硬件抽象化的解决方案，它使你可以轻松地将所有资源池化并通过一个友好的用户界面（UI）或 API 来提供给消费者。一个软件定义的存储的解决方案使得你可以在不增加任何工作量的情况下进行纵向扩展（Scale-Up）或横向扩展（Scale-out）。

很多人坚持认为软件定义的存储就是将传统存储设备的功能移到了主机上。这从存储设备的虚拟化版本（例如惠普的 StoreVirtual VSA 系列产品）出现以来，逐渐进化到运行在各种不同的硬件平台上的各种解决方案（例如 Nexenta 的解决方案），成为一种趋势。自此一个新的时代来临了。

1.3 超融合 / 服务器 SAN 解决方案

当今世界，超融合（hyper-convergence）/ 服务器 SAN 解决方案分为两种：

- 超融合设备（hyper-converged appliance）

❑ 纯软件解决方案（software only solution）

超融合解决方案是一类在单个机箱内提供完整的虚拟机平台解决方案的设备。这个机箱里通常含有多个安装了虚拟管理程序（hypervisor）[⊖]的商用 x86 服务器，并利用虚拟存储设备或一个基于内核的存储栈将本地存储汇聚到一个大的共享池中。现在市场上可以见到的典型产品有 Nutanix、Scale Computing、SimpliVity 和 Pivot3。图 1-1 显示的就是一个典型的此类设备，它在一个 2U 高的机箱里面集成了 4 台主机。

图 1-1 超融合存储供应商常用的硬件

问题来了：如果这只是些安装了 hypervisor 的传统 x86 服务器加上一个虚拟存储设备，那么相比传统存储系统它的优势在哪里？超融合平台的优点是：

❑ 投产时间短，安装、部署时间不超过 4 小时

❑ 易于管理和集成

❑ 能同时在容量和性能上进行横向扩展

❑ 相比传统环境更低的总购置成本

这些解决方案总是以单一库存单位（stock keeping unit，SKU）的方式出售，并且通常提供统一的支持服务。这可以避免在产品支持问题上相互扯皮。然而，对很多人来说，困难来自于这些解决方案在硬件和配置上过于死板。超融合厂商采用的硬件常常不是自己所偏好的硬件供应商，因此在涉及系统更新和补丁，甚至是在布线和上架等问题时引发了很多可操作性层面的质疑。其实这就是一个信任问题。某些人被服务器厂商 X 洗脑后根本不会考虑其他品牌，而另一些人却可能完全不喜欢品牌 X。这就是基于软件的存储解决方案的切入点。

纯软件的存储解决方案又分两类。现在最常见的解决方案是虚拟存储设备（virtual storage appliance，VSA）。VSA 解决方案是以虚拟机方式部署的，它们安装于物理硬件上的 hypervisor 层上。VSA 允许你将底层的物理资源池化为一个共享的存储设备。VSA 的例子有 VMware 的 vSphere Storage Appliance、Maxta、惠普的 StoreVirtual VSA 和 EMC Scale IO。纯软件解决方案的优点是通常可以利用现有的硬件，只要它们存在于硬件兼容列表（hardware compatibility list, HCL）中即可。大多数情况下，这个硬件兼容列表和 hypervisor 所支持的硬件类似，只有少数关键组件例外，如磁盘控制器和闪存设备。

VSAN 也是一种纯软件的解决方案，然而，VSAN 和上面提到的这些略有不同。VSAN 位于一个不同的层面上，它不是一个基于 VSA 的解决方案。

1.4 Virtual SAN 简介

对于软件定义的存储，VMware 计划把重点放在一系列本地存储、共享存储以及存储 /

⊖ hypervisor 指安装在裸设备上的极为精简的虚拟化中间层，虽然有时候可以译作虚拟管理程序，但是为了行文顺畅，本书以下章节中均保留 hypervisor 不作翻译。——译者注

数据服务的 VMware 创新项目上。一言以蔽之，VMware 想让 vSphere 成为存储服务的平台。

存储曾经是一种在项目初期配置、部署完毕并且在其整个生命周期都不会变更的存在。如果需要变更某个正被虚拟机使用的 LUN 或卷的某些特性或属性，大多数情况下，原始 LUN 或卷必须被删除并重建。这是一种干扰生产的、有风险的并需要耗费大量时间的操作，因为它需要在 LUN 或卷之间迁移数据。这甚至可能需要几个星期来进行协调沟通。

有了软件定义的存储后，虚拟机存储需求可以动态地实例化，而不需要重建 LUN 或卷。随着时间的推移，虚拟机的工作负载和需求可能会发生变化，底层的存储可以在任何时候根据工作负载来进行调整。这就是 VSAN 所想要实现的目标：通过主机上的软件层面来将底层的硬件**集成**、**抽象化**和**池化**，以此来提供存储服务和服务水平协议（service level agreement, SLA）的**自动化**。

软件定义的存储的一个关键要素是基于存储策略的管理（storage policy-based management，SPBM）。这也是 vSphere 5.5 发行版的一个关键特性。SPBM 可以看作是在 vSphere 5.0 时引入的 VMware 存储配置文件（Storage Profile）特性的下一代进化版本。存储配置文件的着眼点更侧重于如何确保虚拟机在置备时能选择到正确的存储设备，而在 vSphere 5.5 中 SPBM 则是 VMware 如何实现基于软件的存储的极为重要的组件。

通过 SPBM 和 vSphere API，底层存储技术表现为一个具有多种功能的抽象化的存储空间池，并展现给 vSphere 管理员用于虚拟机置备。这些功能与性能、可用性或存储服务（例如精简置备、压缩、复制等）相关。由此 vSphere 管理员可以用这么一组功能（这组功能是运行在虚拟机上的应用程序所需要的）创建一个**虚拟机存储策略**（或配置文件）。在部署的时候，vSphere 管理员选择一个虚拟机存储策略，SPBM 将这个虚拟机存储策略推送给存储层和数据存储，使之明白其中的要求并用于存储选择。这意味着虚拟机总能被创建在能满足虚拟机存储策略中的要求的合适的底层存储上。

如果虚拟机工作负载或其 I/O 的模式随时间的推移发生了变化，只需要简单地对特定虚拟机（甚至只需对虚拟磁盘）应用一个新的虚拟机存储策略即可。在这个新的虚拟机存储策略中包含反映新的工作负载的需求和特性。之后，策略就可以无须任何管理员人为干预、无缝地被实现（比照而言，很多传统存储系统都需要手工将虚拟机或虚拟磁盘迁移到一个不同的数据存储上才能实现同样的功能）。为了实现和 vSphere 的无缝集成并提供 SPBM 功能，开发出了 VSAN。

1.5 什么是 Virtual SAN

Virtual SAN 是 VMware 推出的一种新的存储解决方案，它的 beta 版本在 2013 年发布，并于 2014 年 3 月正式开放给公众。VSAN 完全集成在 vSphere 中，它是一种基于对象的存储系统，是虚拟机存储策略的平台，这种存储策略的目标是为了帮助 vSphere 管理员简化虚拟机存储放置的决策。它完全支持并与 vSphere 的核心特性，诸如 vSphere 高可用性

（HA）、分布式资源调度（DRS）以及 vMotion 等深度集成在一起，参见图 1-2。

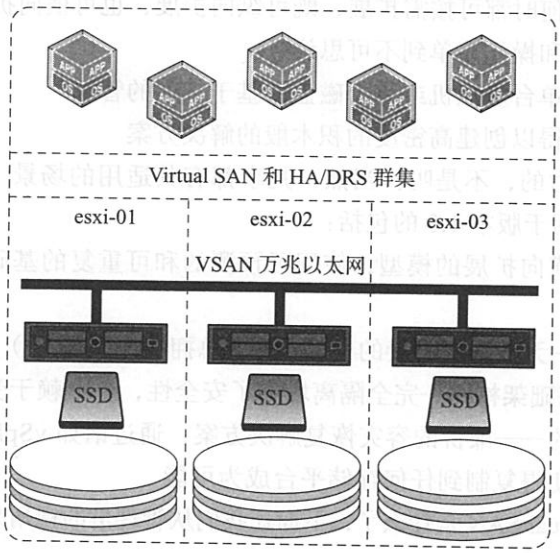


图 1-2　VSAN 群集概览

VSAN 的目标是在提供弹性的同时提供横向扩展存储的能力。从 QoS 的角度来考虑，其目标还在于创建虚拟机存储策略以在每台虚拟机甚至是每个虚拟磁盘的粒度上来定义性能和可用性水平。

VSAN 是一种基于软件的分布式存储解决方案，它直接构建在 hypervisor 中。它不是已有的其他解决方案所采用的那种虚拟设备（virtual appliance），而应该被认为是一种基于内核的解决方案，是 hypervisor 的一部分。技术上来说，这并不完全准确，因为对应于性能和响应速度的关键组件（例如数据路径和群集）是位于内核中的，而其他组件可以被认为是“控制层面”（control plane）的一部分，通常以原生用户空间代理（native user-space agent）方式被实施。虽然如此，对于 VSAN，除了你早已熟悉的 VMware vSphere 本身之外，不需要安装任何其他东西。

Virtual SAN 意味着简单。无须多言，它就是那么简单。试一下吧，它就是简单到只需要为 Virtual SAN 的传输创建一块 VMkernel 网络接口卡（network interface card，NIC）并在群集级别上启用即可（如图 1-3 所示）。当然，为了提供最佳的用户体验，的确有一些建议和前提条件。不用担心，第 2 章会细细为你讲来。

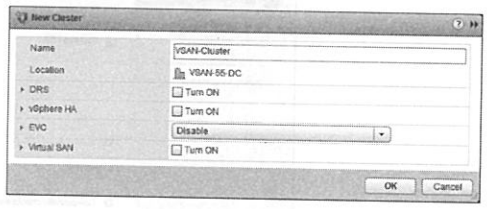


图 1-3　两次点击启用 VSAN

现在我们知道了它易于使用且配置简单，那么像 VSAN 这样的解决方案到底有什么优点呢？主要的卖点在哪里呢？

- **软件定义的**——使用工业标准的硬件
- **弹性**——无论何时都可按需扩展，既可纵向扩展，也可横向扩展
- **简单**——管理和操作简单到不可思议
- **自动**——针对单台虚拟机或单个磁盘的基于策略的管理
- **融合**——使你得以创建高密度的积木般的解决方案

听上去好有竞争力的，不是吗？当然，凡事都有其适用的场景。Virtual SAN 1.0 也有特定的应用场景。适用于版本 1.0 的包括：

- **虚拟桌面**——横向扩展的模型，使用可预测的和可重复的基础架构模块来降低成本和简化运营
- **测试和开发**——无须购买昂贵的存储（降低总拥有成本 TCO）并可快速置备
- **管理或 DMZ 基础架构**——完全隔离增加了安全性，不依赖于受其管理的资源
- **容灾恢复的目的**——廉价的容灾恢复解决方案，通过诸如 vSphere Replication 之类的特性来启用，使得复制到任何存储平台成为可能

现在我们知道 Virtual SAN 是什么了，下面让我们从管理员的视角来看一下它是什么样的。

1.6 从管理员角度来看 VSAN 的样子

当 VSAN 启用的时候，一个共享的数据存储就开始展现给该 VSAN 群集中的所有主机。这就是 VSAN 强大的地方——它以数据存储的形式出现。就像现在所有其他存储解决方案一样，这个数据存储能用来存储虚拟机及其所关联的组件，例如虚拟磁盘、交换文件和虚拟机配置文件。当你在部署一台新的虚拟机的时候，你将会看见一个熟悉的界面，其中列出了可用的数据存储，包括基于 VSAN 的数据存储（如图 1-4 所示）。

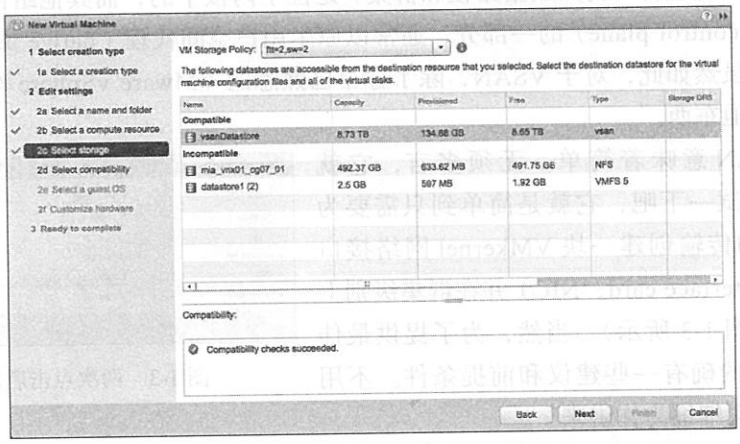


图 1-4 一个普通的数据存储

这个 VSAN 数据存储是在主机的本地存储资源之外形成的。通常来说，在一个启用了

VSAN 的群集中所有的主机都要为这个共享的数据存储贡献出自己的性能（闪存）和容量（磁盘）。这意味着当群集逐渐扩大的时候，数据存储也会随之成长。VSAN 因此被称为横向扩展（在群集中添加主机）的存储系统，但是也允许纵向扩展（给主机增加资源）。

每台为 VSAN 群集贡献存储容量的主机都至少需要有一块闪存盘和一块传统磁盘。为了构成共享存储，VSAN 要求群集中至少有 3 台主机把自己的存储提供给共享的数据存储使用，其他群集中的主机可以不贡献自己的存储而仅仅只是利用这些存储资源。图 1-5 显示一个具有 4 台主机的群集，其中的 3 台（esxi-01、esxi-02 和 esxi-03）贡献了自己的存储，第四台主机没有提供自己的存储，而只是作为消费者去使用存储资源。尽管一个带有不贡献存储的主机的非统一的群集[⊖]在技术上是可行的，但是为了总体上获得更好的利用率、更佳的性能和更高的可用性，我们强烈建议创建一个所有的主机都贡献自己的存储的统一群集。

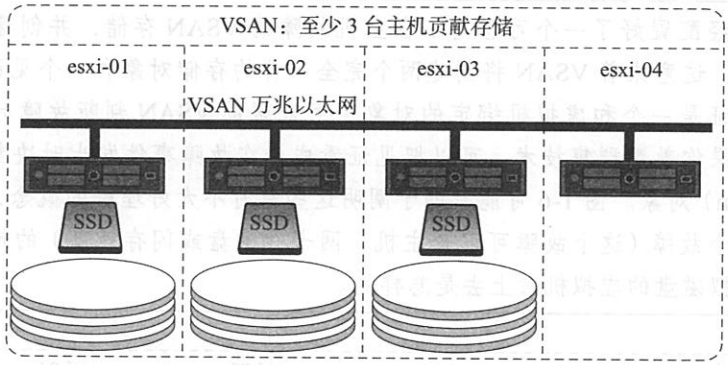


图 1-5 非统一的 VSAN 群集的示例

无论是从容量还是网络连接的角度来看，现在 VSAN 的局限性都来自于 vSphere 群集。这意味着连接到一个 VSAN 数据存储的主机最多可达 32 台，每台主机最多可以支持 100 台虚拟机，综合起来就是总计可达 3200 台虚拟机运行在一个 32 主机构成的 VSAN 群集上，其中 2048 台虚拟机受 vSphere HA 的保护。

可以想象如果只使用普通磁盘，很难从性能上提供良好的用户体验。为了提供最佳的用户体验，VSAN 采用了闪存。闪存资源被用于读缓存和写缓冲，每个写 I/O 都会先写入闪存，并最终批量写回磁盘。而读 I/O 操作则不同，要看需要读取的数据是否在缓冲区内。完美的情况下所有的读 I/O 都出自闪存。第 5 章将深入探讨读缓冲和写缓存的机制与技术细节。

为了保证部署的虚拟机可以带有某些特性，VSAN 允许你为每台虚拟机或每块虚拟磁盘分别配置策略。这些策略可以帮助你实现为工作负载定义的服务水平目标（service level objective，SLO），这可以是性能相关的特性——例如读缓冲或磁盘条带，但也可以是可用性相关的特性——例如用来保证虚拟机磁盘（以及其他重要文件的）的策略副本的放置。

⊖ 非统一的群集，原文为 nonuniform cluster，指不是由完全一致的主机所构成的群集。——译者注

如果曾经用过虚拟机存储策略，你或许会产生疑惑——是不是保存在同一个 VSAN 数据存储上的所有虚拟机都必须配置相同的虚拟机存储策略呢？事实并非如此。VSAN 允许在同一个数据存储上给不同的虚拟机提供不同的策略，甚至是给同一个虚拟机的不同虚拟磁盘提供不同的策略。

如前所述，通过利用策略，弹性等级可以根据每块虚拟磁盘的粒度来进行配置。一个镜像副本会存在于多少台主机和多少磁盘上将取决于所选择的策略。因为 VSAN 使用由策略定义的镜像副本来提供弹性，所以不需要本地 RAID 组。换而言之，主机提供给 VSAN 存储空间的磁盘应该只需要简单地提供一组磁盘即可。

不管你定义的策略可以容忍 1 台主机故障，还是要容忍 3 台主机同时发生故障，VSAN 都会保证你的对象有足够多的副本被创建出来。下面的例子描述了 VSAN 和大多数现有的其他虚拟存储解决方案之间的主要区别，以及这对于 VSAN 有多么重要。

示例 我们已经配置好了一个可容忍 1 台主机故障的 VSAN 存储，并创建了一个新的虚拟磁盘。这意味着 VSAN 将创建两个完全一样的存储对象和一个见证（witness）对象。见证是一个和虚拟机绑定的对象，用来帮助 VSAN 判断故障时谁将赢得所有权。如果你熟悉群集技术，可以把见证看成是在故障事件发生时决断所有权的仲裁（quorum）对象。图 1-6 可能有助于阐明这些有时不太好理解的概念，它描述了可以容忍一个故障（这个故障可以是主机、网卡、磁盘或闪存设备）的情况下一台带有一块虚拟磁盘的虚拟机看上去是怎样的。

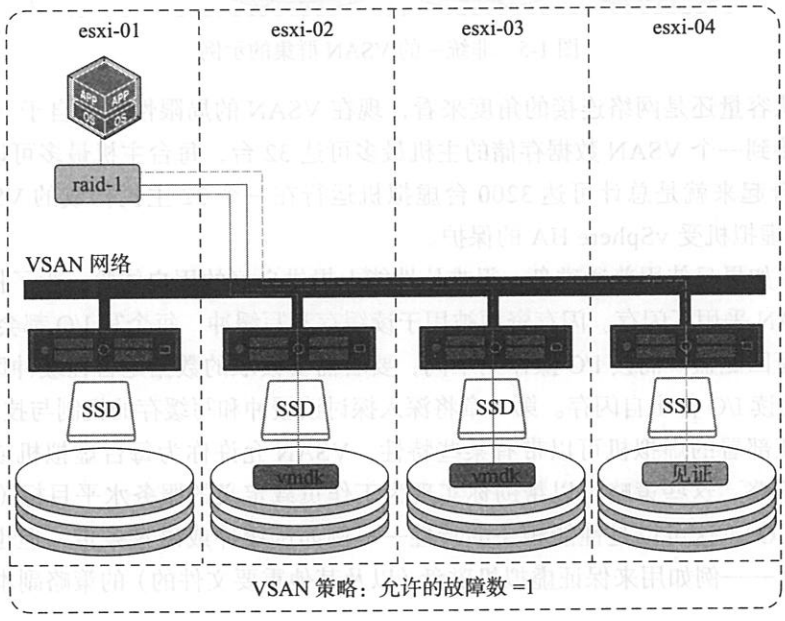


图 1-6 VSAN 允许的故障数

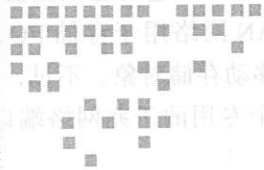
在图 1-6 中，虚拟机运行在第一台主机（esxi-01）上，而它的虚拟磁盘位于群集的其他主机（esxi-02 和 esxi-03）上。在这个场景中，VSAN 网络用于存储 I/O，使得虚拟机可以在群集中自由移动而无须随着计算资源的迁移而移动存储对象。不过，这导致了实施 VSAN 所需要的第一个必要条件：VSAN 要求最少一个专用的千兆网络端口，不仅如此，VMware 推荐给 VSAN 网络提供万兆以太网连接。

是的，这看上去可能还是有点复杂，但平心而论，VSAN 将所有这些复杂性都遮掩掉了，我们将在本书余下的章节中进一步阐述这一点。

1.7 小结

总结一下，vSphere Virtual SAN（VSAN）是一个基于 hypervisor 的全新分布式存储平台，它汇聚了计算和存储资源，使你得以通过基于策略的管理来定义以虚拟机为粒度的服务水平目标。它使你得以用一种前无古人的简单且高效的方法来控制可用性和性能。

初窥皮毛之后，是时候进一步进行探讨了。第 2 章将描述安装和配置 VSAN 所需要的前提条件。



Chapter 2 第 2 章

VSAN 部署的前提条件和要求

在深入 VSAN 的安装和配置之前，有必要先讨论一下安装的前提条件和要求。VMware vSphere 5.5 Update 1（U1）是每一个基于 VSAN 的虚拟基础架构的基石。

2.1 VMware vSphere 5.5

VMware vSphere 5.5 U1 包含两个主要组成部分：vCenter Server 管理平台和 ESXi 虚拟管理程序（hypervisor）。要安装和配置 VSAN，这两者缺一不可。

VMware vCenter Server 为 VMware vSphere 环境提供了一个集中的管理平台。它是一个用来置备新的虚拟机、配置主机并进行许许多多与管理虚拟基础架构相关的运营工作的解决方案。

要运行一个完全支持 VSAN 的环境所必须的最低版本是 vCenter Server 5.5 U1。无论是 Windows 版本的 vCenter Server 还是 vCenter Server 虚拟设备（vCenter Server Appliance，VCSA）都可以用来管理 VSAN。VSAN 的管理和监控是通过 vSphere Web Client 来进行的，这也要求版本必须至少是 5.5 Update 1。对于那些希望对配置、监控或管理进行某种程度的（甚至全部）自动化的管理员来说，VSAN 还可以完全通过命令行界面（CLI）和 vSphere 应用程序编程接口（API）来进行配置与管理。尽管单个群集只能有一个 VSAN 数据存储，但是一个 vCenter Server 实例可以管理多个 VSAN 和计算群集。

2.1.1 ESXi 5.5 U1

VMware ESXi 是一个企业级虚拟化产品，它允许你在一台独立的服务器上以完全相

互隔离的方式运行一个操作系统的多个实例。它是一个裸设备的解决方案，这意味着它无须借助客户操作系统并且自身所占的空间极小。ESXi 是当今世界上绝大多数虚拟化环境的基础。

为了形成一个受支持的 VSAN 群集，需要至少 3 台 ESXi 主机（每台主机均具有本地存储并提供存储给 VSAN 数据存储使用）。这是为了让群集满足最低的可用性要求——可以容忍至少一台主机发生故障。VSAN 群集在一个群集中支持最多 32 台 ESXi 主机，而在 beta 版发布的时候只能支持 8 台。这些 ESXi 主机必须至少是版本 5.5 U1。

一台主机内存的最小推荐值是 6GB。如果你的主机配置的磁盘组数量已达上限，我们建议这台主机至少配置 32GB 内存。主机对内存的要求直接和主机中配置的物理磁盘的数量以及磁盘组的数量相关。关于这个问题，你会在第 9 章中了解更多细节。

2.1.2　ESXi 主机引导的考虑因素

为基于 VSAN 的基础架构安装 ESXi 时，把 ESXi 镜像安装在什么地方有多种选择：本地磁盘、USB 闪存驱动器或 SD 卡。当前版本的 VSAN 不支持 ESXi 的无状态启动（自动部署方式）。选择将 ESXi 安装到 USB 闪存或 SD 卡的额外好处是无须为镜像浪费一块磁盘，于是这块磁盘就可以被 VSAN 用作创建分布式的、共享的、VSAN 数据存储，来部署虚拟机。

对于内存小于 512GB 的主机来说，是可以从 USB 或 SD 卡引导的。对于内存配置超过了 512GB 的主机，ESXi 需要安装在一块本地磁盘上。这是因为 VSAN 必须使用核心 dump 分区来存储 VSAN trace 文件，这些 trace 文件将被用于 VMware 全球支持服务和 VMware 工程师团队进行故障的根本原因分析。VSAN trace 文件将在第 10 章中详细探讨。请注意，当将 ESXi 安装在 USB 或 SD 卡上的时候，设备应该至少有 8GB 内存。

如果主机没有 USB 或 SD 卡而把 ESXi 安装在一块本地磁盘上时，这块本地磁盘将无法加入一个磁盘组，因而无法用于提供存储给 VSAN 数据存储。正因为如此，在磁盘插槽数量有限的环境中，我们建议使用 USB 或 SD 卡。

2.2　VSAN 的要求

在开始启用 VSAN 之前，我们强烈建议 vSphere 管理员首先验证一下环境是否满足了所有的前提条件和要求。下面的列表中我们还增加了一些从基础架构的角度来增强弹性的建议：

- ❑ 至少 3 台 ESXi 主机
- ❑ 每台 ESXi 主机至少 6GB 内存
- ❑ VMware vCenter Server
- ❑ 每台为 VSAN 数据存储提供存储的主机至少有一块硬盘

- ❑ 每台为 VSAN 数据存储提供存储的主机至少有一个闪存设备
- ❑ 用于安装 ESXi 的可引导设备
- ❑ 推荐使用支持直通（Pass-through）/JBOD 模式的磁盘控制器
- ❑ 专用于 VSAN VMkernel 接口的千兆以太网端口

2.2.1 VMware 硬件兼容性指南

在开始安装配置 ESXi 之前，建议先根据 VMware 官方的 VSAN 兼容性指南来验证一下硬件配置。VSAN 兼容性指南可以在下面的网站找到：

http://vmwa.re/vsanhcl

VSAN 对磁盘、闪存设备和磁盘控制器有严格的要求。因为选择众多，配置一台完美的 VSAN 主机可能是一件复杂的事情。在逐个讨论这些组件之前，你应该知道还有另外一个选择：VSAN Ready Nodes。

2.2.2 VSAN Ready Nodes

较之自行选择组件，VSAN Ready Nodes 是一个很棒的替代方案。很多厂商帮你做好了功课，创建了这些叫做 VSAN Ready Nodes 的硬件组合。这些 VSAN Ready Nodes 由经过测试和认证的硬件组成，在我们看来，它们可以提供额外的保证。VSAN Ready Nodes 也已经列在兼容性指南列表中了，如图 2-1 所示。

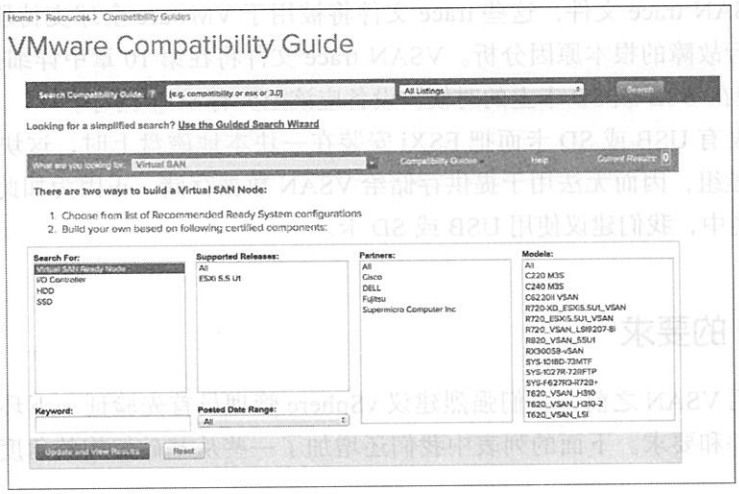


图 2-1 Virtual SAN Ready Nodes 配置

对于那类更具有冒险精神的管理员，或者那些偏好某个未在 VSAN Ready Nodes 兼容指南中列出的品牌或厂商的管理员，就必须关注各个组件（例如存储控制器和磁盘驱动器）的兼容性。下面几节将重点探讨这些需要考虑的因素。

2.2.3　存储控制器

每台加入 VSAN 群集的 ESXi 主机都需要一个磁盘控制器。这个磁盘控制器最好具有通常称为**直通模式**（Pass-through mode）、HBA **模式**或者 JBOD **模式**的功能。换而言之，磁盘控制器应该能够直接控制底层作为独立驱动器的磁盘或固态硬盘（SSD），而无须经过其上的 RAID 层。于是，ESXi 的操作就可以无须被控制器截取并解释，而直接对磁盘进行。为虚拟机定义策略属性（诸如可用性和性能属性）时，VSAN 会负责磁盘的 RAID 配置。在 VSAN 兼容性指南中列出了已经成功通过测试的磁盘控制器。

在配置新服务器的时候，每家服务器厂商都有很多不同的磁盘控制器可供选择。下面列出了最常见的服务器品牌及其磁盘控制器，包括一些常被诸如 SuperMicro 这类品牌使用的通用 LSI 磁盘控制器：

- Dell PERC H200
- HP H220i
- IBM ServeRAID M1015 SAS/SATA
- LSI 2008，LSI 9207-8i，LSI 9211-8i，LSI 9240-8i

在写作本书的时候，这些磁盘控制器已经被列入了兼容性指南列表中。但是在采购前你应该比对一下最新版本的 VMware 兼容性指南来验证你要购买的控制器是否列在其中。有时候，兼容性指南会因为固件版本的变更而变动。

在 VSAN beta 版时，我们在很多低端主板上常见的 AHCI 板载控制器上遇到了一些麻烦。这些磁盘控制器通常用于家庭实验室，但是，ESXi 自带的这些控制器的驱动程序有一个缺陷，导致磁盘会随机出现“Permanent Device Loss”（永久性设备丢失）状态。这些 AHCI 控制器还会引起性能低下，因为它们并不是为 VSAN 用例设计的。这些控制器通常不在 VMware 兼容性指南中，因此不推荐将它们用于生产环境或者较为正式的实验环境。

在某些场景下，硬件可能已经购买好了，而现有的磁盘控制器不支持直通模式。这时候可以使用 RAID-0。

1. 磁盘控制器 RAID-0

对于不支持直通 /HBA/JBOD 模式的磁盘控制器，VSAN 支持通过 RAID-0 配置的磁盘驱动器。RAID-0 的卷如果在配置中只包含 1 个磁盘驱动器，就可以被 VSAN 使用。这对磁盘和 SSD 都是一样的。这个操作可以通过磁盘控制器的软件或固件来实现。管理员必须了解，当 SSD 使用了 RAID-0 配置时，往往就不会被 VSAN 认作一个闪存设备，因为此时其闪存的特性被 RAID-0 设置屏蔽了。如果这种情况发生了，你必须用 esxcli 命令行来标注这个磁盘驱动器为闪存设备。下面的例子会告诉你应该如何操作。还有一个例子来说明如何解决另一种设备识别的问题——如何标注一个设备为本地（local）。在某些环境中，一些 ESXi 主机本地的设备会被认作为共享卷，这是因为某些 SAS 控制器允许被多个主机同时

访问。在这种情况下，尽管设备是本地的（local），但是它们会被显示为共享的而非本地的（not local）。这些设备不会被用于 VSAN，因为目前 VSAN 不支持使用 SAN 或其他类型的共享卷。在下面的例子中，我们创建了一条规则，将设备 mpx.vmhba2:C0:T0:L0 标注成本地设备和闪存设备：

```
esxcli storage nmp satp rule add -satp VMW_SATP_LOCAL -device
mpx.vmhba2:C0:T0:L0 -option "enable_local enable_ssd"
```

当使用 RAID-0 而不是直通模式的时候，必须考虑某些影响。当使用直通模式时，驱动器在大多数情况下都会被直接识别出来，无须将其配置成“本地”（local）或“固态硬盘”（SSD）。而使用 RAID-0 时，驱动器会绑定在某个 RAID-0 配置上，这意味着驱动器和 RAID-0 配置是一一对应的。如果这个驱动器出现故障且需要更换为一个新的驱动器，那么这个一一对应的关系就会被打破，新的驱动器替换上来的时候就必须重新手动建立一个新的 RAID-0 配置来与之对应。因此在使用 RAID 控制器配置的时候，就会有额外的工作量。而采用直通模式时，只需要简单地移除并插入新磁盘即可。

2. 性能和 RAID 缓存

VMware 已经对各种不同类型的磁盘控制器和 RAID 控制器进行了很多性能测试。大多数情况下，直通模式和 RAID-0 配置之间的性能差异可以忽略不计。

当使用 RAID-0 配置的时候，应该禁用存储控制器的写缓存，让 VSAN 获得全部的控制权。当 RAID-0 配置中的存储控制器写缓存无法被完全禁用的时候，应该将存储控制器的缓存配置成 100% 用于读缓存，这也是一种有效地禁用写缓存的方法。

2.2.4 磁盘

在给 VSAN 群集提供磁盘容量的时候，每台 ESXi 主机必须至少拥有一块磁盘。额外的磁盘显然可以提供更多容量，也能提升性能，因为虚拟机存储对象可以被条带化并分散到多个磁盘上去。此外，在需要的时候更多数量的磁盘可以提供更多容量均衡选择。每个磁盘将成为一个磁盘组的一部分。一个 VSAN 主机最多可以有 5 个磁盘组，每个磁盘组最多可以包含 7 块磁盘，因此最多可达 35 块磁盘，如图 2-2 所示。

从高容量的 7200 RPM 的 SATA 驱动器到低容量但性能更好的 15K RPM 的 SAS 驱动器，VSAN 支持各种类型的磁盘。尽管虚拟机存储 I/O 性能大部分是由闪存提供的，还是有必要指出任何涉及磁盘读写的 I/O 操作还是和磁盘转轴提供的性能直接相关联的。第 9 章会提供各种例子来展示选择 SATA 磁盘或 SAS 磁盘所造成的影响。现在我们将给出下面的列表，列明不同磁盘类型的磁盘能提供的 IOPS 数：

- 7200 RPM SATA：80 IOPS
- 10 000 RPM SAS：130 IOPS
- 15 000 RPM SAS：175 IOPS

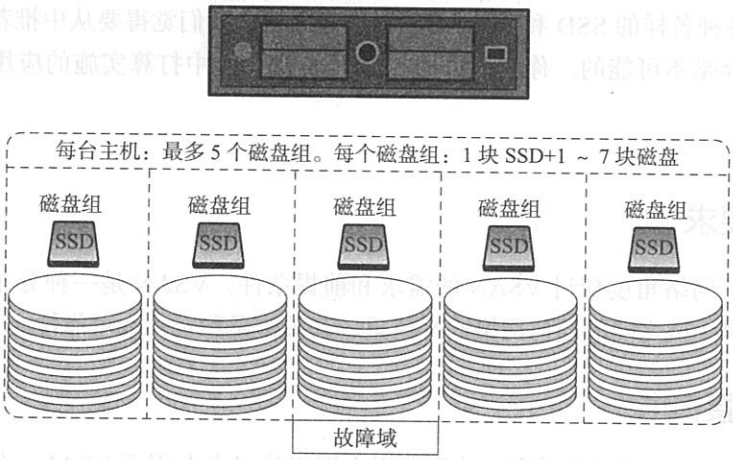


图 2-2　磁盘和磁盘组最大配置

2.2.5　闪存设备

给 VSAN 群集提供磁盘容量时，每台 ESXi 主机必须至少拥有一个闪存设备。这个闪存设备位于一组磁盘的前端，被 VSAN 用作写缓存和读缓冲。每个磁盘组都需要一个闪存设备。VSAN 中每个主机最多可以有 5 个磁盘组，因此每台主机最多可以有 5 个闪存设备。一台主机中的闪存容量越大，提供的性能就越高，因为更多的 I/O 可以被缓冲 / 缓存。

为了得到最佳的 VSAN 性能，VMware 推荐使用至少 20 000 IOPS、每天每单位覆写 10 次的情况下可以有 5 年生命周期的固态硬盘。VMware 支持从固态硬盘（SSD）到 PCIe 闪存设备等各种类型的闪存。VMware 兼容性指南中列出了受支持的 PCIe 闪存设备和固态硬盘列表。在购买新设备之前，建议先查询 VMware 兼容性指南以保证自己的配置是受支持的。

下面列出的是 VMware 兼容性指南中闪存设备的分级：

- Class A：每秒写入 2 500 ～ 5 000 次
- Class B：每秒写入 5 000 ～ 10 000 次
- Class C：每秒写入 10 000 ～ 20 000 次
- Class D：每秒写入 20 000 ～ 30 000 次
- Class E：每秒写入 30 000 + 次

有个问题常会被提起：“能否使用消费级别的 SSD？这样 VSAN 能工作吗？”从技术角度来看，使用消费级别的 SSD，VSAN 也可以完美地工作。然而，大多数消费级别的 SSD 的使用寿命较短，耐用性保证较低，而且性能差异较大（往往性能较差）。尽管从价格角度来看，消费级别的 SSD 可能颇具吸引力，但是我们想强调 VSAN 无论是在读还是写操作上都非常倚重闪存来进行缓冲和缓存，当（闪存）驱动器出现故障的时候会影响到 SDD 绑定的整个磁盘组。闪存设备故障时，整个磁盘组都会被标注为不健全的。

在调研了各种各样的 SSD 和 PCIe 的闪存设备之后，我们觉得要从中推荐一个品牌或某一种类型的闪存是不可能的。你应该根据 VSAN 虚拟环境中打算实施的应用的需求来做出决定。

2.3 网络要求

这一节将从网络角度探讨 VSAN 的需求和前提条件。VSAN 是一种分布式的存储解决方案，因此它对主机之间的通信网络非常倚重，其关键是稳定性和可靠性。

2.3.1 网络接口卡

每台 ESXi 主机必须至少具有一块千兆以太网络接口卡专用于 VSAN。然而，作为最佳实践，VMware 和本书作者都推荐使用万兆网卡。出于冗余的考虑，可以在每一台主机上都配置网卡绑定。我们认为这是最佳实践，但这并不是构建一个完整功能的 VSAN 群集所必需的。

2.3.2 受支持的虚拟交换机类型

无论是 VMware vSphere Distributed Switch ™（VDS）还是 VMware 标准交换机（VSS）均支持 VSAN。使用分布式交换机具有一些优点，这在第 3 章会详细介绍。在 VSAN 的最初发布版中，不支持其他任何虚拟交换机类型。

2.3.3 VMkernel 网络

在每台想要加入 VSAN 群集的 ESXi 主机上，都必须创建一个用于 VSAN 通信的 VMkernel 端口。这个标记为 Virtual SAN Traffic（虚拟 SAN 流量）的 VMkernel 端口是自 vSphere 5.5 中新出现的类型。这个端口用于群集内节点之间的通信，并且当一个特定的虚拟机运行在某一台 ESXi 主机上而构成这台虚拟机的文件的真正数据块又落在群集中另外一台 ESXi 主机上的时候，这个端口也用于读和写操作。在这种情况下，I/O 将通过群集内主机间的网络传递，如图 2-3 所示，VMkernel 接口 vmk02 用于 VSAN 群集内所有主机之间的 VSAN 流量传输，位于 esxi-01 上的虚拟机所有的读和写操作都要用到 VSAN 网络。

2.3.4 VSAN 网络流量

VSAN 使用的协议是个专有协议。VMware 没有公布这个协议的规范，就像 VMware 的 vMotion、Fault Tolerance、vSphere Replication 以及其他 VMware 专有协议等其他 VMware 产品和特性一样。VSAN 网络用于 3 种不同的流量类型，由于这些流量带来了对物理网络交换机配置的一些要求，所以了解清楚它们是怎么回事是很重要的。

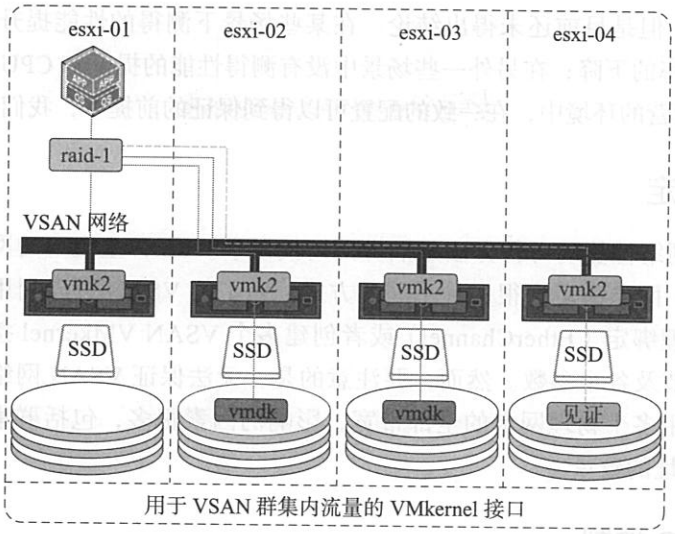


图 2-3　VSAN 流量

- **组播心跳（Multicast heartbeat）**——这类流量用来发现加入到群集中的所有主机，并且判断主机状态。和其他类型的流量相比，组播心跳只产生非常少的数据包。
- **来自群集服务（CMMDS）的组播和单播数据包**——这类流量进行元数据（例如对象放置和统计信息）的更新。它比心跳产生的网络流量多一些，但是仍然只占非常小的百分比。
- **存储流量（例如读和写）**——这是网络流量的主要部分。群集中任何主机和其他主机的通信都是通过单播进行的。

为了保证 VSAN 主机可以正常通信，要求物理交换机允许两层组播的流量。如果 VSAN 群集中的 ESXi 主机之间无法通过 VSAN 流量网络进行组播通信，VSAN 群集将无法正确形成。

组播是一个关键的组成部分，尽管它仅占整个网络流量中很小的比例。由于存储的读写 I/O 都要经过网络，VSAN 群集中大部分流量都是存储流量，保证最优的网络带宽是非常重要的。如果可能不要使用那些把组播流量转换成广播流量方式传输的低端交换机，VMware 建议使用真正支持组播流量的物理交换机。

2.3.5　巨型帧

VSAN 网络支持巨型帧（jumbo frame）。我们相信，每个 VSAN 部署都是不同的，不管是从服务器硬件的角度来看还是从网络硬件的角度来说都是如此。正因为如此，很难说应该推荐使用还是反对使用巨型帧。此外，在非全新配置的环境中实施巨型帧会带来一些运营上的影响。如果巨型帧的配置没能从端到端保持一致，就可能出现网络故障。尽管到目前为止，在 VSAN 环境中实施不一致的巨型帧配置还未出现过问题。我们曾做过一些测试试图证

明巨型帧的优点，但是目前还未得出结论。在某些场景下测得的性能提升了 15% 同时也观察到了 CPU 使用率的下降；在另外一些场景中没有测得性能的提升和 CPU 使用率的下降。

在一个成熟运营的环境中，在一致的配置可以得到保证的前提下，我们建议采用巨型帧。

2.3.6 网卡绑定

另一个优化网络性能的可行方法是捆绑网络接口卡。ESXi 主机上的网络接口卡绑定对 VSAN 是透明的。网卡绑定有很多种不同的方式。为了使 VSAN 可以利用多个物理网卡端口，可以实施物理绑定（EtherChannel）或者创建多个 VSAN VMkernel 接口。第 3 章将深入探讨配置的细节及各个参数。然而，要注意的是，无法保证 VSAN 网络流量是否能在同一时间内完全利用多个物理网卡的全部带宽，影响的因素很多，包括群集的大小、网卡的数量和不同 IP 地址的数量。

2.3.7 网络 I/O 控制

尽管建议使用万兆网卡，但是并非要将这些万兆网卡仅仅专用于 VSAN 网络，它们是可以与其他网络流量共享的。然而，你可能需要考虑使用网络 I/O 控制（NIOC）来保证在网络拥堵的情况下 VSAN 流量仍能获得一定数量的网络带宽。尤其是当这块万兆网卡和诸如 vMotion 之类的流量共享时，因为声名狼藉的 vMotion 可是会在任何可能的情况下吃掉所有带宽的哦。使用 NIOC 必须创建分布式交换机（VDS），因为它不支持标准交换机（VSS）。幸运的是，分布式交换机已经包含在 VSAN 许可证中了。

第 3 章会列举各种例子来说明在不同类型的网络配置的情况下应该怎样配置 NIOC。

2.3.8 防火墙端口

当启用 VSAN 时，有一些 ESXi 防火墙端口会自动在 VSAN 群集的 ESXi 主机上开启（包括入口和出口两个方向）。这些端口将用于群集主机之间的通信以及在 ESXi 主机上的存储提供程序之间进行通信传输。表 2-1 列出了一些 VSAN 专用的网络端口。

表 2-1 VSAN 启用的 ESXi 端口和协议

名称	端口号	协议
Cmmds	12345、23451	UDP
RDT	2233	TCP
Vsanvp	8080	TCP

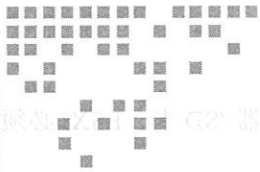
2.4 小结

尽管配置 VSAN 事实上只需要两次鼠标点击即可完成，但是花点时间检查并保证所有前提条件都已满足且准备工作都已经到位是非常重要的。稳定的存储平台要从基石开始打造。在开始进入第 3 章之前，你应该再回顾一下这个检查列表来确认所有前提条件都已经满足了：

- ❑ vSphere 5.5 U1
- ❑ 至少 3 台主机
- ❑ 每台主机至少 6GB 内存
- ❑ 主机内存超过 512GB 时，不支持 USB 闪存驱动器 /SD 卡，ESXi 必须安装在磁盘上
- ❑ 使用认证的磁盘控制器
- ❑ 每台提供容量的主机至少一个认证的闪存设备
- ❑ 每台提供容量的主机至少一块认证的磁盘驱动器
- ❑ 专用于 VSAN 的千兆网卡或者共享的万兆网卡
- ❑ 在 VSAN 的二层网络上启用组播

下面两项额外的建议不是实施一个完整功能的 VSAN 所必需的，但是对于生产环境来说或许也是必要的：

- ❑ 用于 VSAN 的网络交换机冗余性
- ❑ 实施端到端的一致的巨型帧



Chapter 3 第 3 章

VSAN 的安装与配置

本章具体描述安装和配置的过程，以及在进行 VSAN 群集部署之前可能需要考虑的所有初始准备工作的步骤。你将学到如何正确创建一个网络和存储设备的知识，以及一些关于如何部署最优的 VSAN 环境的有用技巧和窍门。

3.1 VSAN 网络

网络连接是任何 VSAN 群集的心脏。VSAN 群集的主机不仅将网络用于虚拟机 I/O，还用于主机间的状态通信。一致且正确的网络配置是 VSAN 部署成功的关键。由于大多数磁盘 I/O 会往返于一台远程主机，因此 VMware 建议使用万兆以太网基础架构。应该指出的是，尽管千兆以太网也完全受支持，但在大规模部署时它可能会成为瓶颈所在。

VMware vSphere 提供了两种不同的虚拟交换机类型，这两者都可以用于 VSAN。

- VMware 标准虚拟交换机（VSS）提供了从虚拟机和 VMkernel 端口到外部网络的连接，但是它仅存在于一台 ESXi 主机本地。
- vSphere 分布式交换机（VDS）为横跨多台 ESXi 主机的虚拟交换机管理提供了集中控制。除了 VMware 标准虚拟交换机可以提供的功能之外，它还可以提供额外的网络特性，例如网络 I/O 控制（NIOC），可以为你的网络提供服务质量管理（QoS）。尽管 VDS 通常需要特定的 vSphere 版本，但是 VSAN 已经包含了 VDS 许可，而不管你正在运行的是什么版本的 vSphere。

3.2 为 VSAN 服务的 VMkernel 网络

所有参与 VSAN 网络的 ESXi 主机都需要相互通信。vSphere 5.5 引入了一个新的

VMkernel 类型，叫做 Virtual SAN Traffic（虚拟 SAN 流量），用于 VSAN 流量的传输。只有当 VSAN VMkernel 端口在加入到 VSAN 群集的每一台 ESXi 主机上都存在的时候，VSAN 群集才会成功构建起来。在构建 VSAN 群集之前，vSphere 管理员必须在每一台群集内的 ESXi 主机上都创建一个 VSAN VMkernel 端口（参见图 3-1）。

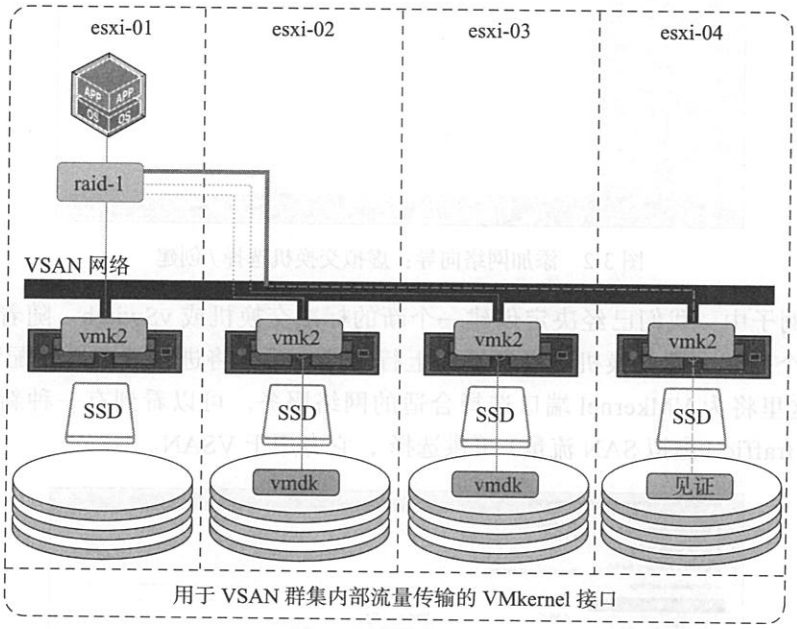


图 3-1　用于 VSAN 群集内流量传输的 VMkernel 接口

如果 VSAN 没有 VMkernel 网络，群集就无法成功构建。如果 VSAN 群集内的 ESXi 主机之间无法建立通信，只有一台 ESXi 主机会加入进 VSAN 群集，其他 ESXi 主机将无法加入。当群集中的 ESXi 主机之间出现通信困难的时候，会显示一条警告消息。如果在所有 VMkernel 端口都创建出来之前就建立了群集，同样会显示一条关于主机之间存在通信困难的警告消息。直到所有的 VMkernel 端口都创建完成，通信建立起来后，群集才会成功构建起来。

3.3　VSAN 网络配置之 VMware 标准交换机

通过 VMware 标准交换机创建一个用于 VSAN 网络流量的端口组是相对简单的。在安装 ESXi 主机的时候，一个 VMware 标准交换机已经自动创建，并用来承载 ESXi 网络管理流量和虚拟机流量。可以使用这个现存的标准交换机以及与之关联的与外部网络通信的上行链路来创建一个用于 VSAN 流量的新的 VMkernel 端口。或者可以选择为 VSAN 网络流量的 VMkernel 端口创建一个新的标准交换机（见图 3-2），并为它选择一些新的上行链路。

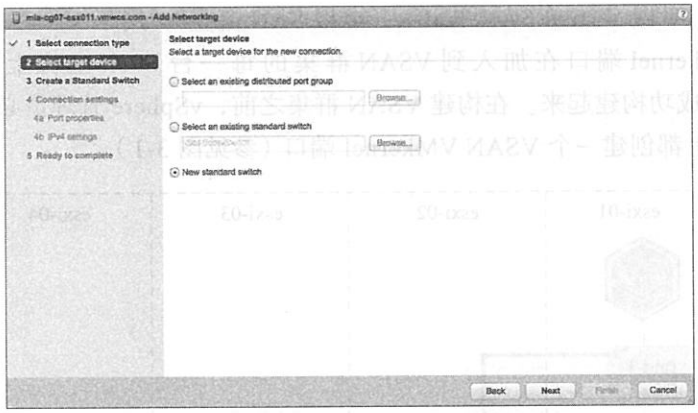


图 3-2　添加网络向导：虚拟交换机选择 / 创建

在这个例子中，我们已经决定创建一个新的标准交换机或 vSwitch。随着添加网络向导，在为这个新的标准交换机选择合适的上行链路之后，将进行端口属性配置（如图 3-3 所示）。在这里将为 VMkernel 端口选择合适的网络服务。可以看到有一种新的流量类型 Virtual SAN traffic（虚拟 SAN 流量）可供选择，它专用于 VSAN。

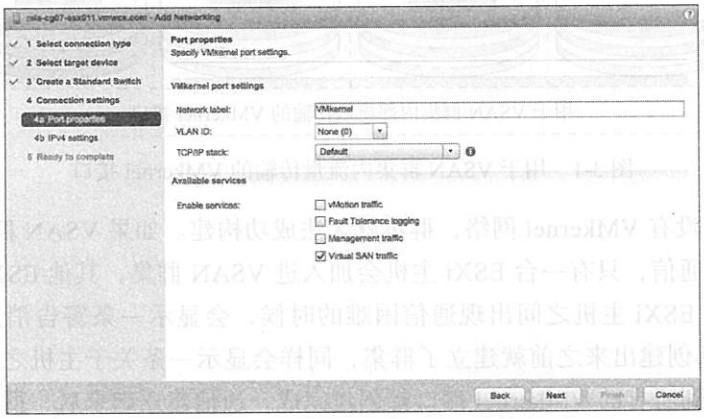


图 3-3　在端口上启用虚拟 SAN 流量服务

完成向导之后，你就获得了一个配置了传输 VSAN 流量的 VMkernel 端口组的标准交换机。当然，你必须在 VSAN 群集中的每一台 ESXi 主机上重复这个步骤。

3.4　VSAN 网络配置之 vSphere 分布式交换机

VSAN 要使用 VDS，需要配置一个分布式端口组来承载 VSAN 流量。创建分布式端口组后，就可以在独立的 ESXi 主机上创建 VMkernel 接口来使用这个分布式端口组。接下来将详细描述这个过程。

第 1 步：创建分布式交换机

尽管 VMware 官方文档中没有明确指出应该使用哪个版本的分布式交换机，但是我们建议为 VSAN 创建最新版本（v5.5）的分布式交换机，这是笔者进行 VSAN 测试时用的版本。注意，如果使用了版本 5.5 的分布式交换机，那么所有连接到这个 VDS 的 ESXi 主机都必须运行 ESXi 版本 5.5，而不能用老版本的 ESXi。

在创建分布式交换机时的一个步骤是选择是否启用 NIOC。我们建议保留默认值——启用。稍后会讨论 VSAN 环境中 NIOC 的数值（应该如何设置）。

第 2 步：创建分布式端口组

创建分布式端口组的步骤相对简单：

1. 打开 vSphere Web 客户端，导航到 vCenter Server 清单中的 vSphere 分布式交换机对象。

2. 选择创建新的分布式端口组。

3. 给分布式端口组提供一个名字。

4. 设置端口组的属性，例如绑定类型、端口分配，以及可以链接到端口组的端口数量（如图 3-4 所示）。

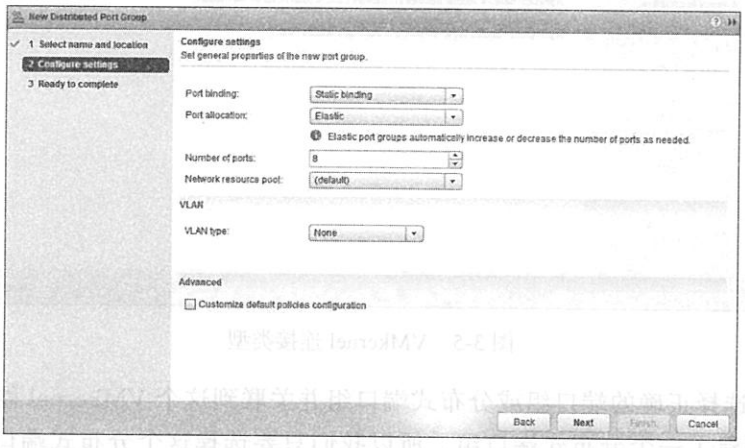


图 3-4　分布式端口组设置

在创建端口组时的一个重要决定是，端口分配（port allocation）设置以及关联到端口组的端口数量。注意，默认端口数量是 8，默认的端口分配设置是弹性（Elastic）。这意味着当所有 8 个端口都分配完之后，新的一组 8 个端口会创建出来。将分配类型设置成弹性的端口组会随着分配的设备数量的变化自动增加或减少端口数量。当端口绑定（Port binding）设置成静态绑定（Static binding）时，一个端口会在连接到分布式端口组的时候被分派给 VMkernel 端口。如果计划创建一个 16 主机或一个 32 主机的 VSAN 群集，你可能需要考虑

将端口数量配置得更大一些而不是默认的 8 个。这意味着当维护或出现故障时，主机能有足够的端口可以分配，直到它可以重新加入群集为止。这也意味着交换机无须承担额外的开销来删除和重新添加端口。

在创建分布式交换机和分布式端口组的时候，有很多其他选项可以选择，例如端口绑定的类型。这些选项在官方的 VMware vSphere 文档中有详细的描述。尽管我们在此稍微多讨论了一些关于端口分配的细节，但是大多数配置都不在本书讨论的范围之内，不熟悉这些选项的读者可以在官方文档中找到解释。不过，对于 VSAN 部署来说，这些分布式交换机和端口组的配置选项即使只是简单地保留其默认值，也是没问题的。

第 3 步：创建 VMkernel 端口

当分布式端口组创建好之后，就可以开始在 ESXi 主机上创建 VMkernel 端口了。给 ESXi 主机添加网络的第一步是选择一个合适的连接类型。对于 VSAN 网络流量而言，连接类型是 VMkernel Network Adapter（如图 3-5 所示）。

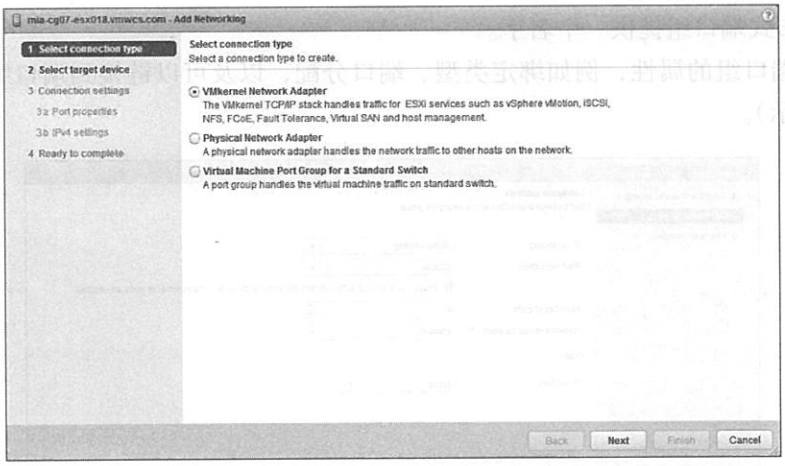


图 3-5　VMkernel 连接类型

下一步是选择正确的端口组或分布式端口组并关联到这个 VMkernel 网络适配器。前面我们已经创建了一个分布式端口组，所以我们只要选择这个分布式端口组即可（如图 3-6 所示）。

分布式端口组选定后，就该为这个 VMkernel 端口选择合适的连接设置了。在连接设置的第一部分，要配置的是端口属性，在这里要选择 VMkernel 端口相关联的服务。这个例子中，要创建一个用于 VSAN 流量的 VMkernel 端口，所以要选择启用的服务是 Virtual SAN traffic，如图 3-7 所示。默认情况下，只有一个 TCP/IP 堆栈可以选，其他网络堆栈类型或许可以在 VMware 官方的文档中找到，它们超出了本书讨论的范畴，不过这里可以简单提一下：ESXi 主机上可以配置不同类型的网络堆栈，它们具有不同的属性（例如每个网络堆

栈各自关联的默认网关）。

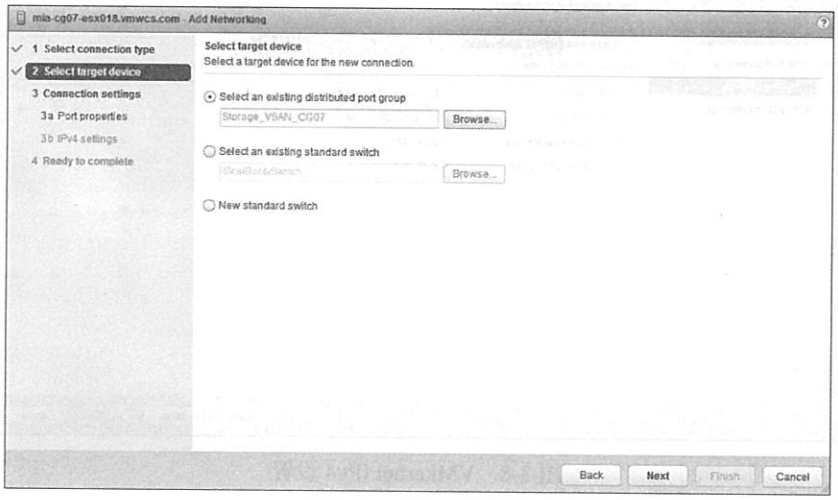


图 3-6　VMkernel 目标设备

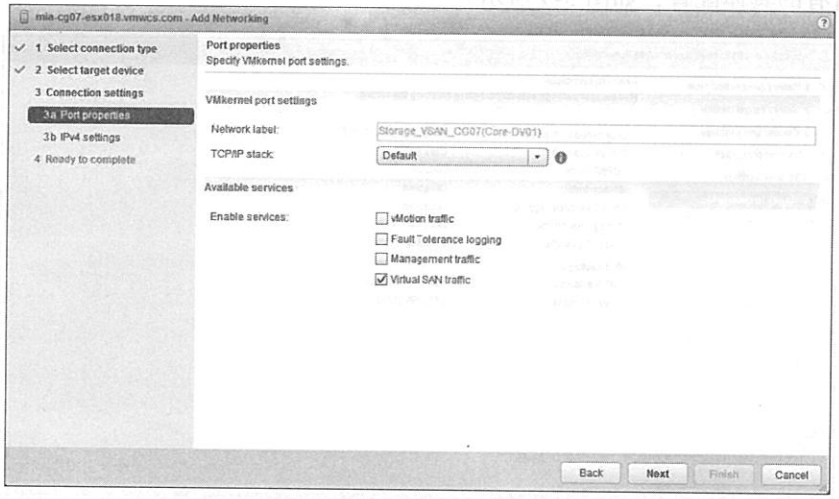


图 3-7　VMkernel 端口属性

正确的服务（Virtual SAN traffic）选好后，下一步是进行 VMkernel 适配器的 IPv4 设置（如图 3-8 所示）。VSAN 1.0 不支持 IPv6。对于 IPv4 有两个可选项：DHCP 或静态。动态主机配置协议（Dynamic Host Configuration Protocol，DHCP）是一个标准的网络协议，它用来给网络上的其他设备提供具体的网络配置。如果选择了 DHCP，在网络上必须存在一台有效的 DHCP 服务器来给 ESXi 主机的这个 VMkernel 端口提供有效的 IPv4 信息。这个例子中选择使用静态配置，所以必须提供一个有效的 IP 地址和子网掩码。对于 VSAN 网络接口，不管是 DHCP 还是静态 IP 都是受支持的。

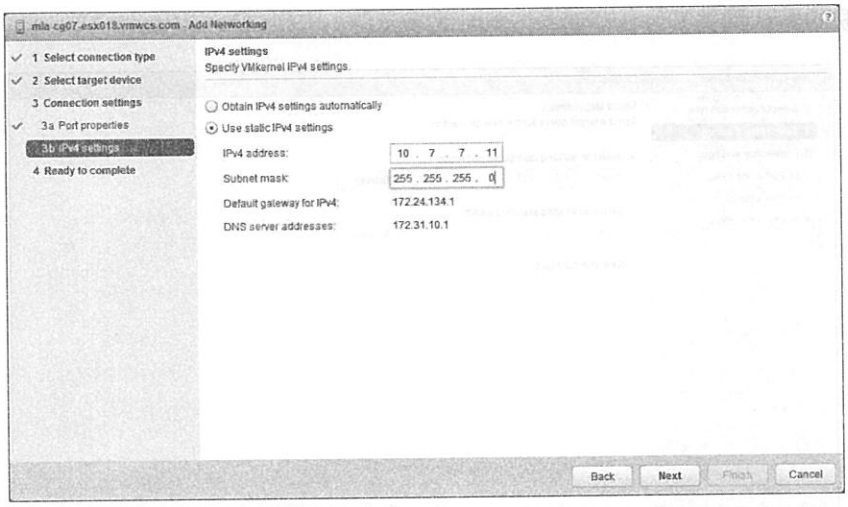


图 3-8 VMkernel IPv4 设置

在 VMkernel 端口的细节都设置完毕之后，在最终创建 VMkernel 端口前，可以再重复检查一次所有的具体配置，如图 3-9 所示。

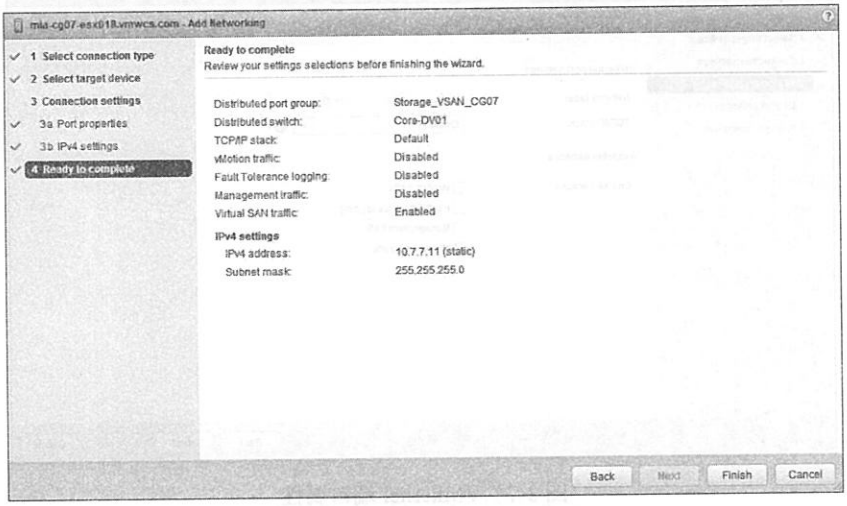


图 3-9 VMkernel 配置即将完成

这个 VMkernel 端口配置必须在 VSAN 群集中的每台 ESXi 主机上重复进行。配置结束时，为了成功创建 VSAN 群集所进行的网络配置准备工作就完成了。

3.5 可能发生的网络配置问题

如果 VSAN VMkernel 没能配置正确，在 VSAN 群集对象的 Manage（管理）页面上

的 Virtual SAN → General（常规）部分会显示一条配置警告消息。点击 Misconfiguration detected（检测到配置错误）消息旁边上的 i 图标，关于网络状态的进一步具体信息会显示出来（如图 3-10 所示）。

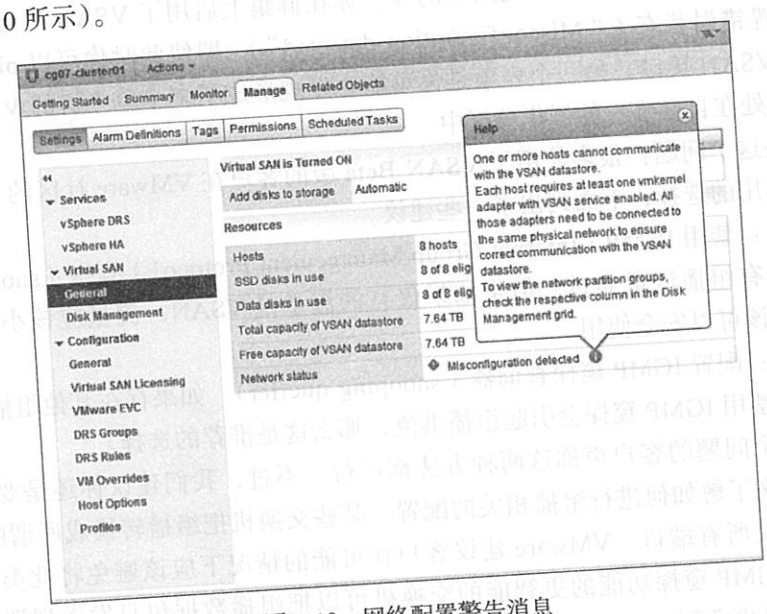


图 3-10　网络配置警告消息

另一个可以观察到 VSAN 通信故障的地方是在 Summary（摘要）页，如图 3-11 所示。如果主机无法和群集中的其他主机通信，Summary 页会显示“Host cannot communicate with all other nodes in the VSAN enabled cluster”（主机无法与已启用 VSAN 群集中的所有其他节点进行通信）。此时，你需要回到 VMkernel 端口属性页面进行检查，确保设置是正确的。

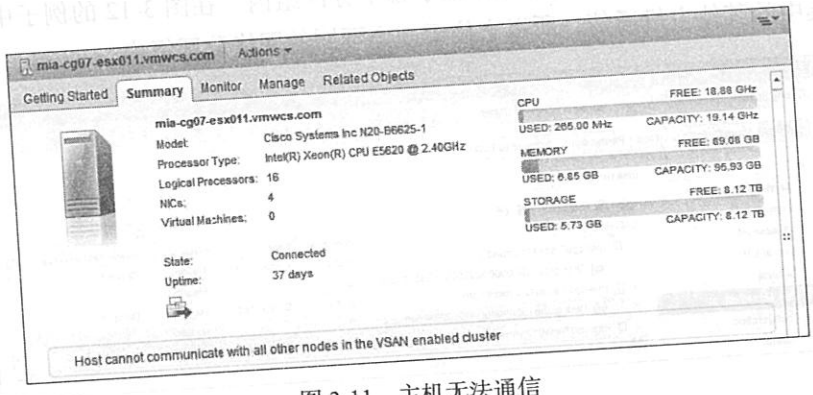


图 3-11　主机无法通信

另一个常常困扰很多客户的问题就是对组播流量传输的依赖。VSAN 的一个必要条件是允许组播流量在 VSAN 群集中的 ESXi 主机之间的 VSAN 网络上传输。不过，组播仅用

于相对不频繁的操作，例如，VSAN 群集中主机的初次发现，以及群集中主机之间持续的“心跳”检查。

那么缺乏组播支持是怎样表现出来的呢？你在群集上启用了 VSAN 后会看见网络状态显示一种配置错误状态（“Misconfiguration detected”），即使此时你可以 ping/vmkping 通所有主机的 VSAN 接口。另一个表象是你可能会发现形成了多个单主机的 VSAN 群集，每个 ESXi 主机处在自己唯一的群集分区中。

如何解决这个问题？很多使用了 VSAN Beta 版的客户在 VMware 社区的 VSAN 论坛中讨论过一些可用的选择，下面列出了一些建议。

- **选择 1**：禁用 IGMP（Internet Group Management Protocol）窥探（snooping）。这会允许**所有**组播流量通过，但是如果仅有的流量是 VSAN，流量应该小到可以忽略，所以应该可以安全使用。
- **选择 2**：配置 IGMP 窥探查询器（snooping querier）。如果存在其他组播流量，而且你担心禁用 IGMP 窥探会引起组播洪流，那么这是推荐的选择。

遇到过这个问题的客户声称这两种方法都可行。不过，我们建议你还是要参考交换机供应商的文档来了解如何进行组播相关的配置。某些交换机把组播转换成所谓的有效广播，数据包将被发往所有端口。VMware 建议客户在可能的情况下应该避免将此类交换机用于 VSAN。具有 IGMP 窥探功能的更智能的交换机可以把组播数据包只发送到那些需要的端口，这类交换机更适用于 VSAN 部署。原因是非智能交换机只是简单地把组播流量转换成广播流量，这可能会导致网络洪流并影响连在同一个交换机上的非 VSAN 主机。

最后需要解释的是如何判断哪台或哪些主机处于隔离于群集的状态。最方便的方法是在 VSAN 的 Manage(管理）页面的 Disk Managment(磁盘管理）中查看 Disk Groups(磁盘组）视图。这个视图包含一个名为 Network Partition Group（网络分区组）的列，其中会显示一个组号来突出显示某台特定的主机现在位于哪个分区组内。在图 3-12 的例子中，一台主机无法与群集中的其他主机通信，所以它位于一个不同的网络分区组内。

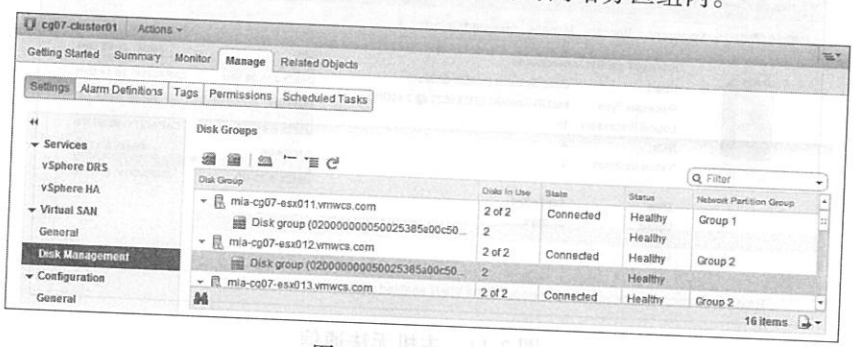


图 3-12　网络分区组

如果群集被成功地创建并且所有主机都能相互通信，在这个视图中的所有主机都将具有相同的网络分区组号。

3.6　网络 I/O 控制配置示例

如前所述，网络 I/O 控制（NIOC）可以用来保证 VSAN 群集的通信和 I/O 传输所需的带宽。只有在 vSphere 分布式交换机（VDS）中才能配置 NIOC，而在 VMware 标准交换机（VSS）中是不支持的。VDS 的确只在某些高版本的 vSphere 中才提供，不过 VSAN 已经包含了 VDS，而不管你使用的是哪个版本的 vSphere。

如果你正在使用一个较早版本的（早于 5.5）的分布式交换机，为了使用 VSAN，我们建议你把它升级到版本 5.5，虽然 vSphere 的文档中没有特别指出这一点。这个建议只是为了小心起见，因为我们所有的 VSAN 测试都是基于这个版本的分布式交换机的。

在 vSphere 5.5 版本中，NIOC 包含一种新的流量类型，叫做“虚拟 SAN 流量”，并对 VSAN 流量提供服务质量（QoS）。虽然对于某些 VSAN 群集环境来说，服务质量配置可能不是必需的，但是在 VSAN 流量正好被共享的同一块万兆网卡上的其他流量类型所影响的时候，有这样的特性就很棒了。vMotion 流量天生的特性就是“爆发性的”，可能会试图占用一个网卡端口上的所有可用带宽，这样就会影响到网卡上共享的其他流量类型，包括 VSAN 流量。在这种情况下，使用 NIOC 就可以避免这种自发的拒绝服务攻击（DoS attack）。

设置 NIOC 相当简单，并且只要配置完成，它就能为所有主机之间的 VSAN 流量保证一定的带宽。当 VDS 生成时，NIOC 默认是启用的。如果这个功能在初始创建分布式交换机的时候被禁用了，那么它还可以再次被启用。方法是这样的：首先，用 vSphere Web 客户端在 vCenter Server 清单中选择 VDS，然后导航到 Manage（管理）页，接着选择 Resource Allocation（资源分配）视图，就会显示出 Network I/O Control（网络 I/O 控制）配置选项（如图 3-13 所示）。

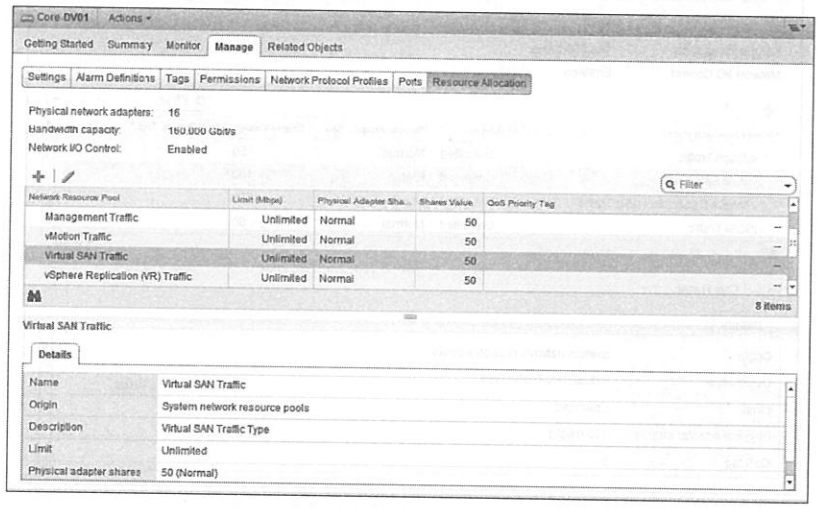


图 3-13　NIOC 资源分配

要为 VSAN 流量改变 NIOC 的资源分配，只需要简单地编辑 Virtural SAN Traffic 网络

资源池（network resource pool）的属性即可。图 3-14 显示了有哪些可以更改的配置选项。

默认情况下，Limit（限制）被设置成 Unlimited（不受限制），Physical adapter shares（物理适配器份额）值设置为 50，QoS tag（Qos 标记）设置为 none。Unlimited 意味着 VSAN 网络流量在没有拥堵的时候可以使用全部网络带宽。如果拥堵发生，Physical adapter shares 就开始发挥作用。这个份额会与分配给其他流量类型的份额进行比较来决定哪种流量类型能得到更高的优先级。

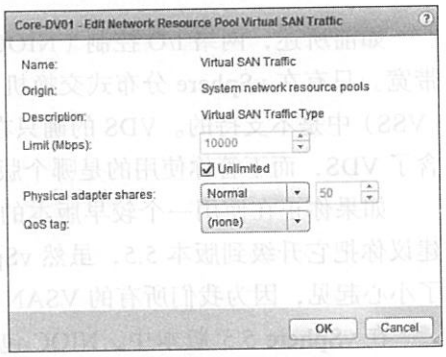


图 3-14 NIOC 配置

有必要进一步解释一下 QoS tag（QoS 标记）。仅仅在主机上定义网络流量的优先级是不够的，当特定的网络流量类型离开某台 ESXi 主机的时候，需要通知整个基础架构这种网络流量类型的优先级，这就是 QoS 标记发挥作用的地方了。QoS 标记其实是 IEEE 802.1p 标记。IEEE 802.1p 标记用一个 3 位的字段来表示优先等级，也叫作 PCP（Priority Control Point，优先级控制点），它位于以太网帧头部。这允许网络数据包分成一种或几种不同的流量类型。标记的数值越大，流量的优先级就越高。这使得外部设备（例如交换机和路由器）得以识别哪些流量的优先级比较高。在图 3-15 的例子中，VSAN 流量的 QoS 标记设置成了 5，份额值设置成了 High（高），这相当于把份额值设置成了 100。

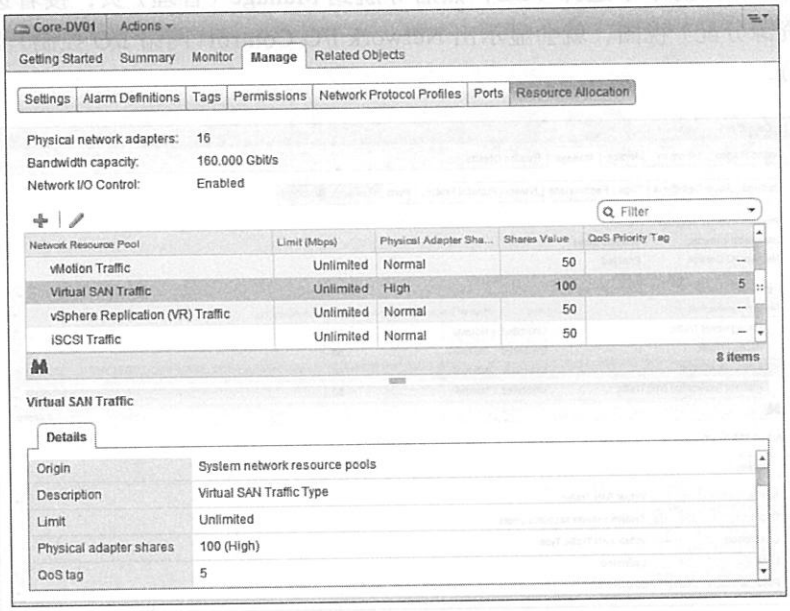


图 3-15 VSAN 流量网络的资源池

VMware 推荐在 VSAN 的部署上使用万兆以太网基础架构。在此类部署中，通常使用

两个万兆以太网端口，分别连接到两台具有万兆以太网能力的物理交换机来提高可用性。因此不同类型的流量将需要共享整个网络带宽，此时 NIOC 可就是无价之宝了。

我们不推荐给 VSAN 流量设定一个 Limit（限制）。原因是，这个 Limit 是一个“硬性的”限制设定。换句话说，如果对 VSAN 流量配置了 2Gbps 的 Limit，即使网络上还有多余的可用带宽，流量也会被限制在 2Gbps。因此我们不推荐使用 Limit 设定。我们建议使用 shares（份额）来根据资源使用和需求情况对各种流量类型做出“虚限制”。

3.7 设计考量：分布式交换机和网络 I/O 控制

为了提供服务质量（QoS）和性能的可预测性，VSAN 和 NIOC 应该携手共进。在讨论配置选项之前，下列网络类型应该列入考虑范围内：

- 管理网络
- vMotion 网络
- Virtual SAN 网络
- 虚拟机网络

这个设计考量假设为了可用性已经准备好了万兆的冗余网络连接和一对冗余的交换机。基于使用的网络交换机类型的不同，我们将描述两个场景：

1. **不具备**链路聚合能力的冗余万兆以太网交换机配置。
2. **具备**链路聚合能力的冗余万兆以太网交换机配置。

注意 链路聚合（IEEE 802.3ad）使得用户可以在网络设备之间用多条链路来进行连接。它将多条物理连接捆绑成一条逻辑连接，并提供某种程度的冗余性和带宽提升。

对于这两种配置，都建议创建以下端口组和 VMkernel 接口：

- 1 个管理网络 VMkernel 接口
- 1 个 vMotion VMkernel 接口（所有接口都在同一个子网内）
- 1 个 VSAN VMkernel 接口
- 1 个虚拟机端口组

为了简化配置，应该创建单个 VSAN 和 vMotion 的 VMkernel 接口。创建多个 VSAN 和 vMotion 的 VMKernel 接口也是可以的，不过创建配置的时候，**必须**将这些 VSAN VMkernel 接口放在不同的子网里。

为了保证不同类型的流量分别在不同的物理端口上传输，我们将利用标准分布式交换机的能力。接下去我们还会告诉你如何使用份额（share）来避免“嘈杂的邻居”。

场景 1：不具备“链路聚合”能力的冗余万兆以太网交换机

在这个配置中有两个独立的万兆以太网上行链路。出于简单性的理由，建议把流量分

隔并将一条万兆以太网上行链路专用于 VSAN。下面是每种流量类型推荐的最小带宽：

- 管理网络：1GbE
- vMotion VMkernel 接口：5GbE
- 虚拟机网络：2GbE
- VSAN VMkernel 接口：10GbE

不同的流量类型将共享同一条上行链路。管理网络、虚拟机网络和 vMotion 网络流量配置成共享上行链路 1，而 VSAN 流量配置成使用上行链路 2。通过这种网络配置方式，在 VSAN 群集处于正常或标准的运行状态下时，所有不同类型的流量都会获得足够的带宽。

为了保证没有一种流量类型会在竞争发生的时候影响其他流量类型，需要配置 NIOC 并设置份额管理机制。

在这个场景的练习中，当定义流量类型的网络份额（shares）时，假设只有一个物理端口可用，且所有流量类型共享同这一个物理端口。

这个场景还使用了最坏情况法来进行考量——即在故障发生时，也能保证性能。通过这种方法，我们可以确保 VSAN 始终拥有 50% 的带宽，同时还给其余流量类型提供足够的带宽，以避免自己引发的可能的拒绝服务攻击。

表 3-1 列出了不同流量类型推荐配置的份额值。

表 3-1 根据不同流量类型推荐的份额和限制配置值（场景 1）

流量类型	份额	限制
管理网络	20	/
vMotion VMkernel 接口	50	/
虚拟机端口组	30	/
Virtual SAN VMkernel 接口	100	/

在为不同类型的流量选择上行链路的时候，应该把不同类型的流量相互隔离，以提供可预见性，同时避免“嘈杂的邻居”的干扰。建议进行以下配置：

- 管理网络 VMkernel 接口 = 使用明确故障切换顺序 = 上行链路 1 活动 / 上行链路 2 备用
- vMotion VMkernel 接口 = 使用明确故障切换顺序 = 上行链路 1 活动 / 上行链路 2 备用
- 虚拟机端口组 = 使用明确故障切换顺序 = 上行链路 1 活动 / 上行链路 2 备用
- Virtual SAN VMkernel 接口 = 使用明确故障切换顺序 = 上行链路 2 活动 / 上行链路 1 备用

为了拥有可预见性，建议在 Teaming and failover 部分选择 Use explicit failover order（使用明确故障切换顺序）选项（见图 3-16）。这个选项——明确故障切换顺序——始终使用活动适配器列表中排在最前面的那条上行链路来传递故障切换检测条件。

将流量分隔开可以优化存储性能，同时也能给 vMotion 和虚拟机流量提供足够的带宽（见图 3-17）。尽管使用“基于负载的绑定”（LBT，Load Based Teaming）机制也能实现，但是请注意，LBT 负载均衡周期是 30 秒，这在共享的上行链路出现“突发”流量的时候可能会引起短时间的拥堵。此外请注意，在进行网络故障排错的时候，它可能会在跟踪物理网

卡端口和 VMkernel 接口之间的关系方面造成一些困难。因此，这个方法也在网络配置上提供了一种简洁性。

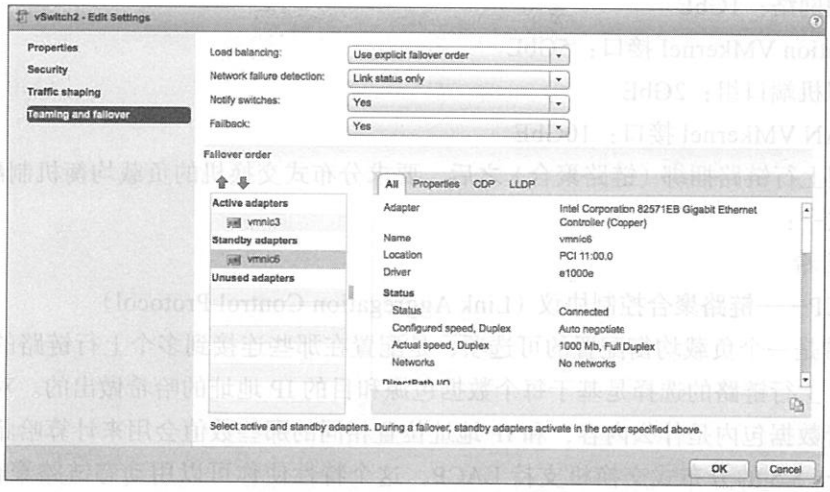


图 3-16　使用明确故障切换顺序

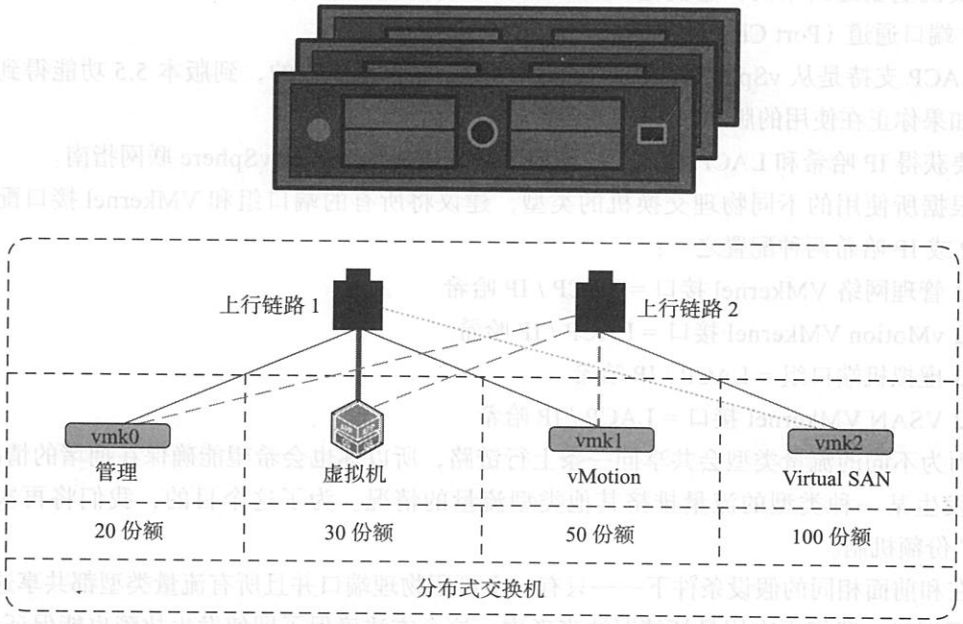


图 3-17　分布式交换机、故障切换顺序和 NIOC 配置

场景 2：具备“链路聚合”能力的冗余的万兆以太网交换机

在这种场景下，两台万兆以太网上行链路配置成一种捆绑的方式（常称为 EtherChannel

或链路聚合）。因为物理交换机具备这样的功能，所以虚拟层面的配置便极其简单。我们仍然以前面推荐的最低带宽作为设计的考量因素：

- 管理网络：1GbE
- vMotion VMkernel 接口：5GbE
- 虚拟机端口组：2GbE
- VSAN VMkernel 接口：10GbE

当物理上行链路捆绑（链路聚合）之后，要求分布式交换机的负载均衡机制配置成下面两种选择之一：

- IP 哈希
- LACP——链路聚合控制协议（Link Aggregation Control Protocol）

IP 哈希是一个负载均衡配置的可选项，是配置在那些连接到多个上行链路的 VMkernel 接口上的。上行链路的选择是基于每个数据包源和目的 IP 地址的哈希做出的。对于非 IP 数据包，不管数据包内是什么内容，和 IP 地址位置相同的那些数值会用来计算哈希。

vSphere 5.5 的分布式交换机支持 LACP。这个特性使你可以用动态链路聚合的方法把 ESXi 主机和物理交换机连接起来。链路聚合组（Link Aggregation Group，LAG）是在分布式交换机上创建出来的，它们把 ESXi 主机上的物理网卡的带宽聚合起来，并依次连接到 LACP 端口通道（Port Channel）。

LACP 支持是从 vSphere 分布式交换机版本 5.1 开始引入的，到版本 5.5 功能得到了增强。如果你正在使用的版本较早，则应该升级到版本 5.5。

要获得 IP 哈希和 LACP 支持的更多细节，可以参考官方的 vSphere 联网指南。

根据所使用的不同物理交换机的类型，建议将所有的端口组和 VMkernel 接口配置成 LACP 或 IP 哈希两种配置之一：

- 管理网络 VMkernel 接口 = LACP / IP 哈希
- vMotion VMkernel 接口 = LACP / IP 哈希
- 虚拟机端口组 = LACP / IP 哈希
- VSAN VMkernel 接口 = LACP / IP 哈希

因为不同的流量类型会共享同一条上行链路，所以你也会希望能确保在拥堵的情况下，不会发生某一种类型的流量排挤其他类型流量的情况。为了这个目的，我们将再次使用 NIOC 份额机制。

在和前面相同的假设条件下——只有一个可用物理端口并且所有流量类型都共享这同一个物理端口，我们再次用最坏情况法来考虑。这个方法确保了即使发生故障也能保证性能。通过这种方法，我们可以确保 VSAN 始终能获得 50% 的带宽并同时能给其他类型的流量留出足够的带宽，以避免自己引发的可能的拒绝服务攻击。

当两条上行链路都可用时，VSAN 得到的带宽相当于 10GbE。当只有一条上行链路可用时（由于网卡故障或者维护的原因），可用带宽减半，相当于 5GbE 带宽。

表 3-2 列出了不同流量类型推荐配置的份额值。

表 3-2　根据不同流量类型推荐的份额和限制配置值（场景 2）

流量类型	份额	限制
管理网络	20	/
vMotion VMkernel 接口	50	/
虚拟机端口组	30	/
VSAN VMkernel 接口	100	/

图 3-18 描述了这种配置的场景。

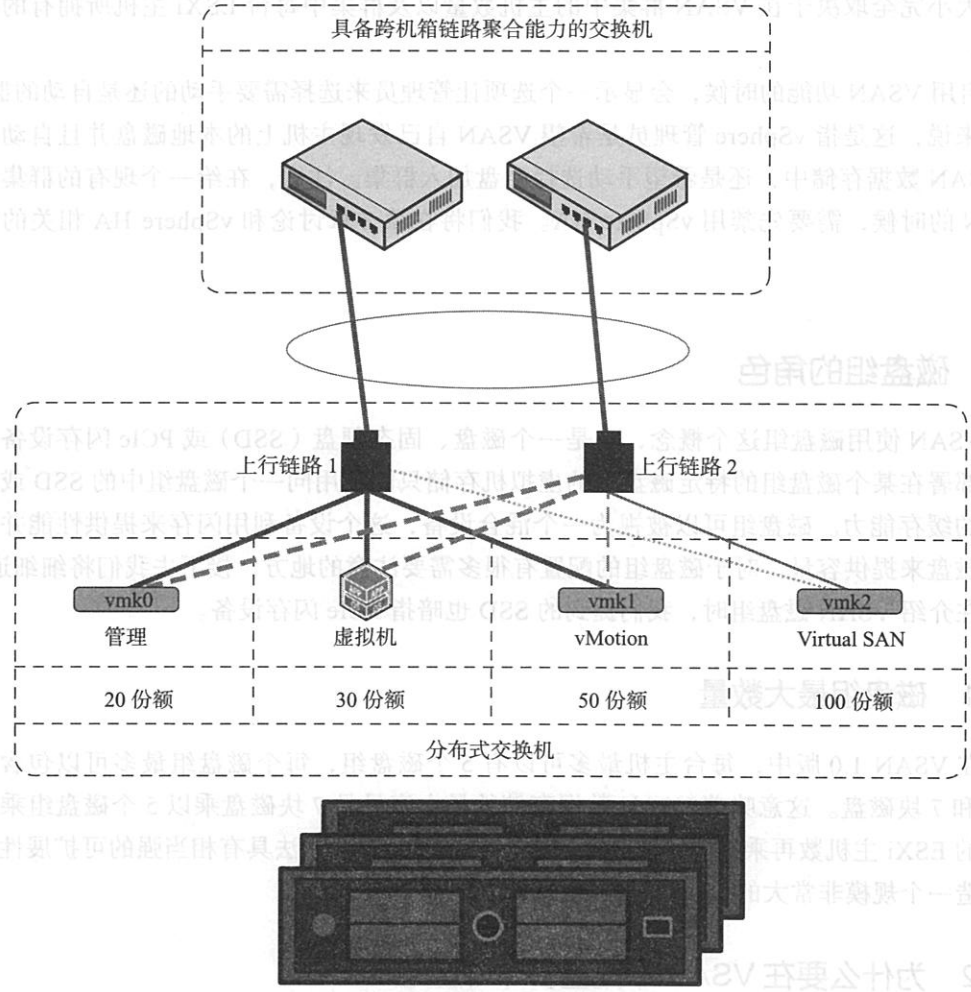


图 3-18　分布式交换机的链路聚合配置

这里讨论的两种场景都应该可以给你的 VSAN 群集提供一个优化的网络配置。

3.8 创建 VSAN 群集

VSAN 群集的创建在很多方面都和 vSphere DRS 或 HA 群集的创建完全一样。群集对象是在 vCenter server 清单中创建的，你可以选择先启用 VSAN 群集的功能，然后再添加主机到这个群集；或者也可以先添加主机到群集，之后再在群集上启用 VSAN。在一个群集上启用 VSAN 的结果就是在这个 VSAN 群集中的所有 ESXi 主机都可以访问一个共享的分布式的 VSAN 数据存储。在 VSAN 1.0 版中，一个 VSAN 群集只能创建一个 VSAN 数据存储。因此，所有的本地存储都将被这个 VSAN 数据存储享用。

VSAN 数据存储是由这个群集中每一台 ESXi 主机的本地存储构成的。VSAN 数据存储的大小完全取决于在 VSAN 群集中的主机数量以及群集中每台 ESXi 主机所拥有的磁盘数量。

启用 VSAN 功能的时候，会显示一个选项让管理员来选择需要手动的还是自动的群集。简单来说，这是指 vSphere 管理员是希望 VSAN 自己发现主机上的本地磁盘并且自动添加到 VSAN 数据存储中，还是希望手动选择磁盘加入群集。注意，在给一个现有的群集配置 VSAN 的时候，需要先禁用 vSphere HA。我们将在第 8 章讨论和 vSphere HA 相关的配置变更。

3.9 磁盘组的角色

VSAN 使用磁盘组这个概念，它是一个磁盘、固态硬盘（SSD）或 PCIe 闪存设备的容器。部署在某个磁盘组的特定磁盘上的虚拟机存储只能利用同一个磁盘组中的 SSD 或闪存设备的缓存能力。磁盘组可以被视为一个混合设备，这个设备利用闪存来提供性能并同时利用磁盘来提供容量。对于磁盘组的配置有很多需要注意的地方，接下去我们将细细道来。后面在介绍 VSAN 磁盘组时，我们提到的 SSD 也暗指 PCIe 闪存设备。

3.9.1 磁盘组最大数量

在 VSAN 1.0 版中，每台主机最多可以有 5 个磁盘组，每个磁盘组最多可以包含 1 块 SSD 和 7 块磁盘。这意味着 VSAN 数据存储的最大容量是 7 块磁盘乘以 5 个磁盘组乘以群集中的 ESXi 主机数再乘以每块磁盘的容量。看见没，这种方法具有相当强的可扩展性，可以打造一个规模非常大的分布式数据存储。

3.9.2 为什么要在 VSAN 中配置多个磁盘组

磁盘组最多只能包含一块 SSD。当 vSphere 管理员想把 ESXi 主机中的多块 SSD 加入到一个 VSAN 群集的时候，就必须创建多个磁盘组。如果性能很重要的话，管理员可

以提高 SSD 对磁盘的比率。SSD 对磁盘的比率越高，用于 I/O 加速的缓存就越多。或者，vSphere 管理员也可以决定在所有的磁盘组中保持一致的 SSD 对磁盘的比率，以保证虚拟机性能的一致性。我们将在第 5 章中具体讨论 SSD 的作用。现在，理解 SSD 不会对 VSAN 数据存储的容量做出贡献，而只是用作虚拟机的 I/O 加速器（70% 用作读缓存，30% 用作写缓存）就足够了。

另一个采用多个磁盘组的原因是它允许 vSphere 管理员来定义故障域。在有多个磁盘组的情况下，每个磁盘组里面都含有一块 SSD 和多块磁盘，当某块 SSD 发生故障的时候，故障域就会被限制在那块故障 SSD 所在的磁盘组里面的磁盘。而如果磁盘组规模很大包含很多磁盘，一块 SSD 的故障就可能影响很多虚拟机。在设计磁盘组配置的时候，应该将故障域问题考虑进去。

图 3-19 显示了一个包含有 5 台主机的 VSAN 群集，每台主机带有 1 块 SSD 和 7 块磁盘。任何一台主机中的 SSD 出现故障，只有同一个磁盘组中的磁盘会受到影响。SSD 故障不会影响群集中其他主机或它们的磁盘组。事实上如果同一台主机上还有其他磁盘组，它们也同样不会受到这块 SSD 故障的影响。所以说，一个磁盘组可以用来定义一个故障域。

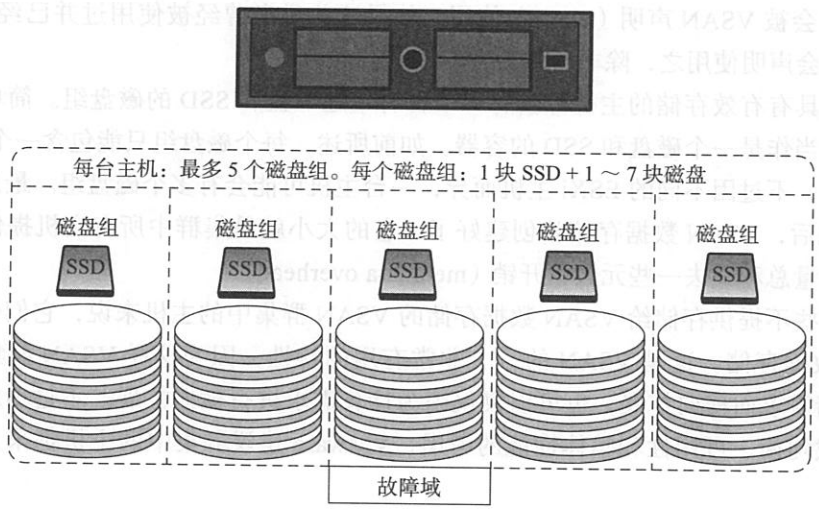


图 3-19　磁盘组定义了故障域

3.9.3　SSD 与磁盘的比率

从硬件角度设计 VSAN 环境的时候，必须认识到 VSAN 的性能高度依赖于 SSD。根据经验，VMware 建议闪存容量对虚拟磁盘容量总数的比率至少要达到 10%，这个比率是在不考虑“可允许的故障数”的前提下算出的。VMware 也支持更低的比例，但是较大的比率能事实上改善虚拟机的性能，因为更多的 I/O 能被缓存。在 VSAN 环境中，SSD 将用作虚拟机的读缓冲和写缓存。

10% 这个数值是基于大多数工作数据集[⊖]的比率是 10% 这样一个假设得出的。根据这个经验数据（仅仅是经验数据）来进行配置意味着运行在虚拟机中的应用程序的活动数据应该都位于闪存之中。

例如，假设我们有 100 台虚拟机，每台虚拟机有 100GB 虚拟磁盘，预期平均使用率是 50GB。此时，可计算得出下面的结论：

$$10\% \times (100 \times 50\text{GB}) = 500\text{GB}$$

这个数字是闪存容量的总数，应该除以主机的数量。如果你有 5 台主机，那么这个例子中可以算出每台主机建议配置的闪存容量为 100GB。

3.9.4 自动添加磁盘到 VSAN 磁盘组

用手动模式来控制并配置 VSAN 在某些情况下可能有用，但是大多数场景下让 VSAN 自行处理或许更合理。如果创建 VSAN 群集时选择了“ Automatically Add Disks”（自动添加磁盘），那么 VSAN 就会自动发现主机上的本地磁盘和本地固态硬盘（SSD）并且在群集中的每台主机上自动创建磁盘组。注意，这些 SSD 和磁盘只有在不具有任何分区的空的状态下才会被 VSAN 声明（claim）使用，如果这些磁盘曾经被使用过并已经含有数据，VSAN 就不会声明使用之，除非事先把它们清空。

每一台具有有效存储的主机都会有一个含有本地磁盘和 SSD 的磁盘组。简单来说，磁盘组可以被当作是一个磁盘和 SSD 的容器。如前所述，每个磁盘组只能包含一个 SSD 和最多 7 个磁盘，不过因不同的 ESXi 主机而异，一台主机可能会有多个磁盘组。最后，当所有这些都完成后，VSAN 数据存储就创建好了，它的大小就是集群中所有主机提供的所有这些磁盘的容量总和减去一些元数据开销（metadata overhead）。

对于那些不提供存储给 VSAN 数据存储的 VSAN 群集中的主机来说，它们仍然可以访问 VSAN 数据存储。这是 VSAN 的一个非常有用的特性，因为这样 VSAN 群集不仅仅可以因为存储需求而横向扩展，也可以仅仅因为计算需求进行横向扩展。不过请注意，出于更好地负载均衡、可用性和整体性能的考虑，VMware 建议群集中的主机都采用完全一样的配置。

3.9.5 处理 Is_local 还是 Is_SSD 的问题

注意，尽管自动模式会声明本地（local）磁盘，大多数带有 SAS 控制器的 ESXi 主机会把它们的硬盘认作为远程（remote），因此 VSAN 不会自动声明这些磁盘。在这种情况下，即使群集是配置成自动模式的，vSphere 管理员仍将不得不手工创建磁盘组。

尽管第 2 章已经讨论过这个问题了，还是值得在这里重申一次：某些 SAS 控制器可以在多台主机之间共享设备。当 ESXi 检测到这类控制器时，会把这些设备标记为共享的，即

⊖ 工作数据集（working data set）指的是正被使用的数据。——译者注

使它们只由单台 ESXi 主机使用也是如此。不仅如此，当 VSAN 发现共享标记时（flag 的实际配置为 Is_Local: False），为了以防万一出现多台 ESXi 主机共享这个设备的情况，这些磁盘不会被声明。只有当这些设备被标记为 local 后 VSAN 才会去声明它们。这就是为什么人为干预并标记这些设备为 local 是必要的了。你可能会觉得这个方法有点过于谨慎，但是 VSAN 最不想见到的就是错误地声明了一块不该被声明的磁盘。

根据使用的闪存类型的不同，也会发生无法识别闪存设备的情况。在这两种情况下，可以通过使用 esxcli 命令将设备手工标记成 local 或 ssd。下面是如何标记磁盘为 SSD 的操作示例。尽管我们已在第 2 章中讨论 RAID-0 配置时提到过这条命令，还是值得让我们再次回顾一下并继续深入探讨。

第一步是要创建一个新的 SATP 规则。这个规则包含要标记为 SSD 的磁盘设备，并把 --option 的参数值设置为 enable_ssd。在这个例子中我的设备 ID 是 mpx.vmhba0:C0:T0:L0:

```
# esxcli storage nmp satp rule add --satp VMW_SATP_LOCAL --device
mpx.vmhba0:C0:T0:L0 --option=enable_ssd
```

下一步是重新声明 (Reclaim) 设备，以应用这个新的规则：

```
# esxcli storage core claiming reclaim -d mpx.vmhba0:C0:T0:L0
```

最后验证这个设备是否已经被视为 SSD 设备了：

```
# esxcli storage core device list --device=mpx.vmhba0:C0:T0:L0
mpx.vmhba0:C0:T0:L0
   Display Name: Local VMware Disk (mpx.vmhba0:C0:T0:L0)
   .
   .
   .
   Model: Virtual disk
   Revision: 1.0
   SCSI Level: 2
   Is Pseudo: false
   Status: on
   Is RDM Capable: false
   Is Local: true
   Is Removable: false
   Is SSD: true
   Is Offline: false
   Is Perennially Reserved: false
   Thin Provisioning Status: unknown
   .
   .
   .
```

当 SAS 控制器背后的设备属性显示成 Is_Local: false 时，类似的过程可以用来更改 Is_Local 参数。这条命令必须对所有 VSAN 要声明的磁盘（不论是磁盘还是 SSD）都运行一

遍，不仅如此，还必须对 VSAN 群集中所有主机上需要用于 VSAN 数据存储的但却被认作 remote 的磁盘一一执行这条命令。

3.9.6 手工添加磁盘到 VSAN 磁盘组

前面我们提过，在创建 VSAN 群集的时候你可以选择手工添加磁盘。如果选择了这个选项，VSAN 群集仍然会创建，只是这时 VSAN 数据存储的初始大小为 0 字节。管理员必须一台一台地在每台主机上手工创建磁盘组并且将磁盘（每磁盘组可以是 1 ～ 7 块）和 SSD（每磁盘组最多 1 块）加入到这些磁盘组。每当一台主机上的磁盘组创建完成，VSAN 数据存储就会根据已经添加的磁盘容量增长。注意，用作读缓冲（read cache）和写缓存（write buffer）的 SSD 是不被计入 VSAN 数据存储的容量的。

你可能会想这个选项什么时候才用得上呢？可能的原因如下：当 VSAN 在构建磁盘组的时候，它总试图用一种前后一致的方式来进行。然而，因为使用的是很多种不同类型的服务器，尤其是那些使用 SAS 来连接磁盘的服务器，手工的方法或许比自动模式更重要。SAS 通过唯一标识符来定义设备而不是用基于端口的方式。因此在第一台主机上插槽 1 中的磁盘可能加入了 ESXi 主机 1 的磁盘组 1，而在 ESXi 主机 2 上第一个插槽里面的磁盘可能加入的却是磁盘组 2。当不管因为什么原因需要更换磁盘的时候，最最重要的是把正确的磁盘移除并更换成一块新磁盘。正因为如此，vSphere 管理员可能希望手工配置磁盘组，以便能轻松地对磁盘进行标识。

3.9.7 磁盘组创建示例

创建磁盘组的操作仅在群集创建时选择了手工模式后才需要进行。如果群集是以自动模式创建的，那么磁盘组就会自动帮你创建好，并且自动加入主机上所有可用的磁盘。创建一个磁盘组的方法非常简单，不过如前面所提过的，有一些限制条件需要注意：

- 每个磁盘组最多只能有 1 块 SSD
- 每个磁盘组最多可以有 7 块磁盘

如果一台主机含有 7 块以上的磁盘或者多于 1 块的 SSD，那么可以创建多个磁盘组。要创建一个磁盘组，群集必须首先配置成手动模式，如图 3-20 所示。

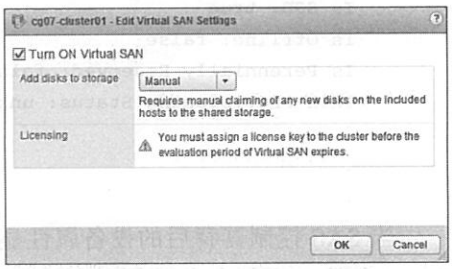


图 3-20 启用 VSAN

当群集创建后，VSAN 不会试图声明存储设备。下一步就是手工创建磁盘组。用 vSphere Web Client 导航到 VSAN 管理下面的 Disk Management（磁盘管理）部分时，选择群集中的某一台主机，点击图标来创建一个新的磁盘组。这将会把主机上所有可用的磁盘（包括 SSD 和磁盘）显示出来，如图 3-21 所示。

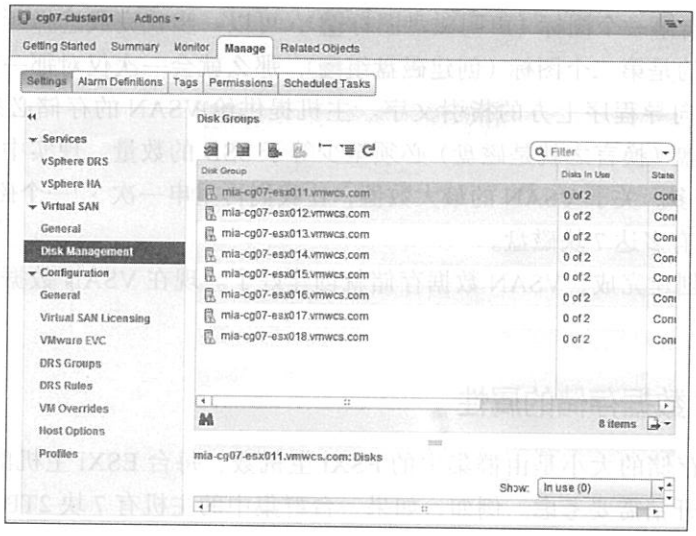


图 3-21　VSAN 磁盘管理

此时 vSphere 管理员有很多可选项。如果愿意，管理员可以决定声明所有主机上的所有磁盘，或者他们可以一台一台地依次为每台主机创建磁盘组。如果磁盘是显示为 not local（例如那些位于 SAS 控制器后面的磁盘），第一个选项会很有用。然而为了在更细的粒度上进行控制，管理员可能喜欢每次只为一台主机创建磁盘组。

当你决定手工创建磁盘组时，vSphere Web Client 提供了一个非常直白的用户界面（UI），你可以在这个用户界面中为每台主机选择磁盘和 SSD，如图 3-22 所示。

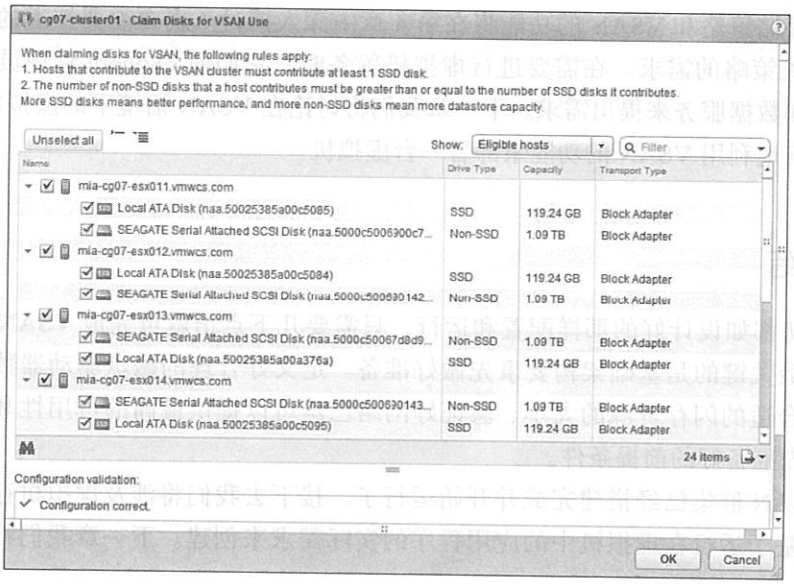


图 3-22　为 VSAN 声明磁盘

如果选择点击第一个图标（声明磁盘图标），可以一步就完成选择所有主机上的所有磁盘。如果点击的是第二个图标（创建磁盘组），那么就会一次仅对那一台主机上的磁盘进行操作。注意向导程序上方的指引文字。主机提供给 VSAN 的存储必须至少包括一块 SSD。非 SSD 硬盘（换言之就是磁盘）必须至少等于 SSD 的数量。现实中，磁盘的数量较之 SSD 总会多很多。关于 VSAN 的最大数值，让我们再重申一次：一个磁盘组仅能包括 1 块 SSD 但是可以有多达 7 块磁盘。

一旦磁盘组创建完成，VSAN 数据存储就创建好了。现在 VSAN 数据存储就可以用于部署虚拟机了。

3.9.8 VSAN 数据存储的属性

VSAN 数据存储的大小是由群集中的 ESXi 主机数、每台 ESXi 主机的磁盘数决定的。还有一些元数据开销需要考虑。例如，如果一台群集中的主机有 7 块 2TB 的磁盘，整个群集有 8 台主机，原始容量将有：

$$7 \times 2\text{TB} \times 8 = 112\text{TB}$$

现在，你的 VSAN 已经配置好了。

VSAN 数据存储一旦构建完成，它的很多功能、属性就会在 vCenter Server 的界面中显示出来。这些功能用于为 VSAN 数据存储上的虚拟机及其关联的虚拟磁盘创建合适的虚拟机存储策略。功能包括条带宽度、组件的故障容忍程度、强制置备和已置备容量。然而，在开始部署虚拟机之前，首先需要理解如何创建合适的虚拟机存储策略来满足虚拟机内运行着的应用的需求。

虚拟机存储策略和 VSAN 的功能将在第 4 章中深入探讨。现在需要知道的是这些功能形成了虚拟机策略的需求。在需要进行虚拟机置备时，它们使 vSphere 管理员可以基于性能、可用性和数据服务来提出需求。下一章我们将讨论在 VSAN 语境下的虚拟机存储策略，以及如何正确地利用 VSAN 的功能来部署一台虚拟机。

3.10 小结

如果一切都如设计好的那样配置和运行，只需要几下点击就可完成 VSAN 的配置。然而，最重要最关键的是基础架构要事先做好准备。定义好合理的磁盘驱动器数量、为性能考虑决定好合适的闪存资源的大小、验证好网络已经可以提供最高的可用性和性能，这些都是必须事先准备好的前提条件。

现在 VSAN 群集已经搭建完成并开始运行了。接下去我们将涉及虚拟机存储策略。这些策略必须基于运行在虚拟机中的应用程序的实际需求来创建。下一章我们将探讨如何实现之。

第 4 章 Chapter 4

VSAN 相关的虚拟机存储策略

VMware 在 vSphere 5.0 中提供了一个叫做配置文件驱动的存储（Profile-Driven Storage）功能。这个功能使得 vSphere 管理员在部署虚拟机时可以轻松地选择正确的数据存储。对数据存储的选择是基于数据存储的能力做出的。在虚拟机的整个生命周期里，配置文件驱动的存储让管理员可以检查其底层的存储是否仍然处于**兼容状态**——换句话说，虚拟机所在的数据存储是否仍然能为虚拟机提供正确的功能？为什么说这个功能很有用？这是因为当虚拟机不管因什么理由被迁移到另外一个数据存储的时候，管理员仍能保证数据存储可以继续满足其需求。即使虚拟机在迁移到一个数据存储前没有检查过目的存储的能力，管理员仍可以随时通过 vSphere 客户端[⊖]来检查虚拟机所在存储的合规性，并可以在当前存储不再满足虚拟机对存储的需求时采取改正措施（即把虚拟机迁回一个合规的数据存储）。

然而，虚拟机存储策略以及存储策略驱动的管理在此基础上更进了一步。上一段我们描述了一种由存储驱动的存储服务质量——在同一个数据存储上的所有虚拟机会继承该数据存储的能力。而对于 VSAN，存储服务质量不再存在于数据存储之上，而是由与虚拟机和虚拟磁盘（VMDK）相关联的虚拟机存储策略来实施的。

4.1 在 VSAN 环境中引入基于存储策略的管理

VSAN 使用一种叫做**基于存储策略的管理**（storage policy-based management，SPBM）的改进方法来部署虚拟机。所有部署在 VSAN 数据存储上的虚拟机都必须使用一种虚拟机

⊖ 由于只有 vSphere Web 客户端才能支持 VSAN 相关的配置，因此本书中如无特殊说明，vSphere 客户端均指 vSphere Web 客户端。——译者注

存储策略。即使没有特别创建过，也会有一个默认策略分配给虚拟机。虚拟机存储策略包含一个或多个 VSAN 功能，这些功能将在本章一一介绍。当 VSAN 群集配置完成，VSAN 数据存储已经创建出来后，vCenter Server 中就会多出一组 VSAN 相关的功能。当群集成功配置后，这些由 VASA[⊖]存储提供程序（马上还会介绍更多内容）带来的功能可在虚拟机部署到 VSAN 数据存储上时对每个虚拟机（及每个 VMDK）设置可用性、容量以及性能策略。

如前所述，这和之前 vSphere 版本中的虚拟机存储配置文件机制大为不同。对于虚拟机存储配置文件来说，其功能和数据存储相关联，用于虚拟机放置决策。

现在我们拥有了一种机制，通过它我们可以指明虚拟机及 VMDK 的要求，并用这些要求来创建一个策略。然后把这个策略传递给存储层，令其为这台虚拟机创建一个满足策略需求的存储对象。事实上，一台虚拟机可以拥有多个与之关联的策略，不同的策略作用于不同的 VMDK。

让我们来解释一下功能、策略和配置文件。功能就是底层存储提供可用性、性能和可靠性等方面的能力，这些功能可以在 vCenter 中看到，并被用来创建一个虚拟机存储策略（后面简称策略）。策略可能包含一个或多个功能，而这些功能反映了虚拟机或者运行在虚拟机上的应用程序的需求。早先版本的 vSphere 使用术语**配置文件**，现在它们被称为**策略**。

在 VSAN 数据存储上部署虚拟机和之前的 vSphere 版本的方法差别很大。以前，管理员要先给一组 ESXi 主机提供一个 LUN 或卷（volume）来存放虚拟机，对于块存储来说，还需要进行分区、格式化并构建 VMFS 文件系统来创建一个数据存储来存储虚拟机文件。对于网络附加存储（NAS）来说，则要把一个 NFS 卷挂载到 ESXi 主机上来形成数据存储。不可能为这些 VMDK 指定 RAID-0 的条带宽度，或者为其指定 RAID-1 的副本。

对于 VSAN 来说，部署虚拟机的方法很不一样。必须要考虑 VM 上运行的应用程序的可用性、性能和可靠性等因素。基于这些要求，就必须要创建一个合适的虚拟机存储策略并在部署时将其与虚拟机关联起来。

在 VSAN 的最初发行版中有 5 种功能，如图 4-1 所示。

你可以在创建虚拟机存储策略时选择所需的功能。虚拟机存储策略对于 VSAN 部署非常重要，因为它们定义了一台虚拟机是如何部署到 VSAN 数据存储上的。通过虚拟机存储策略，你可以定义 VMDK 的 RAID-0 条带组件的数量或者一个 VMDK 的 RAID-1 镜像副本的数量。

接下去的章节中我们会关注在创建虚拟机存储策略时何处应该使用这些功能，以及何时应该将它们设置成与默认值不同的数值。记住，一个虚拟机存储策略可以包含一个或多个功能。

⊖ VASA 即 vSphere APIs for Storage Awareness 的缩写，字面上翻译就是存储感知的 vSphere 应用程序编程接口，它使得 vSphere 可以直接调用存储本身的特性（例如硬件级别的快照和复制），并同时将存储的功能反映在 vSphere 管理软件中。——译者注

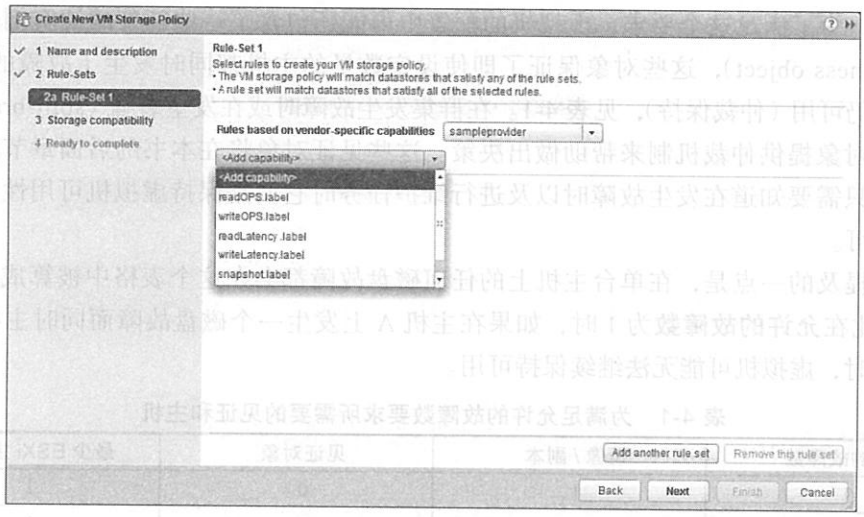


图 4-1　用于虚拟机存储策略的 VSAN 功能

在 VSAN 的最初发行版本中，VM 存储策略具有 5 种可供选择的功能。作为管理员，你可以决定在策略中加入哪些功能，当然这应该取决于虚拟机的要求。例如，虚拟机要求的性能和可用性是怎样的？这些功能包括：

- 允许的故障数（Number of Failures to Tolerate）
- 每个对象的磁盘带数（Number of Disk Stripes per Object）
- 闪存读取缓存预留（Flash Read Cache Reservation）
- 对象空间预留（Object Space Reservation）
- 强制置备（Force Provisioning）

接下来我们将详细描述这些 VSAN 功能。

4.1.1　允许的故障数

允许的故障数（FTT）是一个设定在存储对象上的要求，它是在仍能保证对象可用的情况下，群集中允许出现的主机、网络或磁盘同时发生故障的数量。如果这个属性被设定了，说明该配置必须至少包含**允许的故障数**+1 个副本。你可以把它看作 RAID-1 配置场景，在这种配置中虚拟机存储对象是被镜像了的。然而，镜像是跨 ESXi 主机的，如图 4-2 所示。

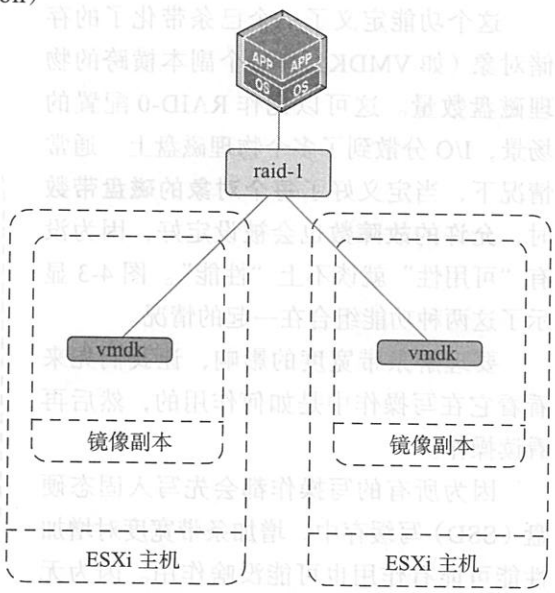


图 4-2　反映在 RAID-1 配置中的允许的故障数

注意，为了满足这个要求，虚拟机的配置中可能还包含了一些额外的被实例化的见证对象（witness object），这些对象保证了即使设定数量的主机都同时发生了故障的情况下，对象数据仍可用（仲裁保持），见表 4-1。在群集发生故障时或在发生裂脑（split-brain）情况时，见证对象提供仲裁机制来帮助做出决策。这些见证对象将在本书的后面章节里详细讨论，现在只需要知道在发生故障时以及进行维护任务时它们是保持虚拟机可用性的必要组成部分即可。

值得提及的一点是，在单台主机上的任何磁盘故障都会在这个表格中被算成一次“故障”。因此在**允许的故障数**为 1 时，如果在主机 A 上发生一个磁盘故障而同时主机 B 又出现了问题时，虚拟机可能无法继续保持可用。

表 4-1 为满足允许的故障数要求所需要的见证和主机

允许的故障数	镜像 / 副本	见证对象	最少 ESXi 主机数
0	1	0	1
1	2	1	3
2	3	2	5
3	4	3	7

如果在虚拟机部署的时候没有选择任何策略，默认策略会将**允许的故障数**设成 1。在创建一个新策略的时候，**允许的故障数**的默认值也是 1。这意味着即使没有在策略中明确说明这个功能，它也已经暗含在内了。

4.1.2 每个对象的磁盘带数

这个功能定义了一个已条带化了的存储对象（如 VMDK）其每个副本横跨的物理磁盘数量。这可以比作 RAID-0 配置的场景，I/O 分散到了多个物理磁盘上。通常情况下，当定义好了**每个对象的磁盘带数**时，**允许的故障数**也会被设定好，因为没有“可用性”就谈不上“性能”。图 4-3 显示了这两种功能组合在一起的情况。

要理解条带宽度的影响，让我们先来看看它在写操作中是如何作用的，然后再看读操作。

因为所有的写操作都会先写入固态硬盘（SSD）写缓存中，增加条带宽度对增加性能可能有作用也可能没啥作用。因为无法保证增加的新条带会使用不同的 SSD，

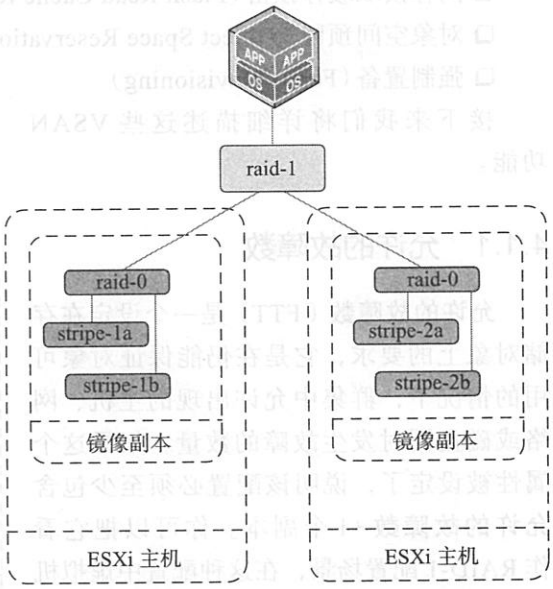


图 4-3 当条带宽度设成 2 且允许的故障数设成 1 时的存储对象配置情况

新的条带可能会被放置在一块位于同一个磁盘组里的磁盘上，因此新的条带会使用同一块 SSD。如果新的条带被置于一个不同的磁盘组中（不管是在同一台主机还是不同的主机上），就会利用到一块不同的 SSD，因此可能会带来性能的提升。不过，作为管理员，你无法控制这种行为的发生。唯一增加条带宽度确保可以增加性能的情况发生在大量的写操作从 SSD 回写到磁盘上时。在这种情况下，增加条带可以提高回写性能。

从读的角度来说，增加条带宽度在大量读缓冲没有命中的情况下会对性能有所帮助。举例来说，假设一台虚拟机每秒有 2000 次读操作，并且读缓冲命中率是 90%，那么有 200 个读操作需要由磁盘直接提供服务。在这种情况下，单块 150 IOPS 的磁盘不足以提供所有的读操作，增加条带宽度就可以起到作用，满足虚拟机对磁盘 I/O 的要求。

一般来说，默认的条带宽度值 1 应该可以满足大多数甚至所有虚拟机的工作负载。只有确认了大块回写或大量读缓存没有命中问题存在并成了性能限制时才需要更改条带宽度值。

4.1.3　闪存读取缓存预留

闪存读取缓存预留是 SSD 或闪存设备上的闪存容量预留用作存储对象的读缓存的数量，其数值表示为存储对象（VMDK）逻辑大小的一个百分比，这个百分比的数值最多可以精确到小数点后 4 位。这么细的粒度是必要的，这样管理员才能表示小于 1% 的单位。以 1TB 的 VMDK 为例，如果读缓存预留只能以每次 1% 的粒度来增加，这就意味着每次要增加 10GB，这在大多数情况下对单台虚拟机来说都太多了。

注意，获得缓存能力不需要手工设置一个预留值。所有虚拟机都会平分一块 SSD 硬盘的读缓存。这个预留值应该不作任何设置（默认值），除非的确有一个读性能问题需要解决而且你确信独占一些读缓存可以解决问题。如果在虚拟机存储策略中添加了这个功能并将数值设置为 0，那么使用这个策略的虚拟机就不会具有任何读缓存。在最初发布的 VSAN 版本中，没有在多台使用读缓存的虚拟机之间按比例分享资源的机制，因此，每台虚拟机都平等地分享读缓存。

4.1.4　对象空间预留

部署在 VSAN 上的所有对象都是精简置备的，这意味着虚拟机部署的时候不会预留任何空间，只有当虚拟机存储增长的时候空间才被使用。**对象空间预留**功能定义了虚拟机存储对象在初始化的时候可能预留的逻辑空间的百分比，它是预留的空间数量占总对象地址空间的一个百分比。这个属性用来定义一个厚置备的存储对象。当**对象空间预留**值设为 100% 时，虚拟机存储对容量的要求都会被预先保留（厚置备）。这是厚置备延迟置零（Lazy Zeroed Thick, LZT）格式的，而不是厚置备置零（Eager Zeroed Thick, EZT）格式的。LZT 和 EZT 的区别在于 EZT 的虚拟磁盘在创建的时候就被置零而 LZT 虚拟磁盘仅仅在第一次写的时候才置零。

4.1.5 强制置备

如果这个参数被设成一个非零值，那么即便数据存储不满足虚拟机存储策略里面的设定，对象也会被置备。在虚拟机 Summary（摘要）页和相关的虚拟机存储策略视图中，这台虚拟机会被显示成不合规。如果群集中没有足够的空间来满足即使是一个副本的预留要求，即便启用了**强制置备**，置备还是会失败。当群集具有额外的资源时，VSAN 会将此对象置为合规状态。记住，这个参数只能在有绝对必要时作为例外情况使用。如果将其作为默认情况使用，将把虚拟机及其关联的所有数据置于危险之地。请慎用！

4.2 VASA 供应商提供程序

作为 VSAN 群集创建步骤的一部分，每台 ESXi 主机都要在 vCenter 上注册一个 VSAN **存储提供程序**（Storage Provider）。它利用了 VASA 将 VSAN 的功能展现在 vCenter Server 界面上，并且利用这些功能可以构建出虚拟机存储策略，最终用于在 VSAN 数据存储上部署虚拟机。如果你熟悉 VASA 并且在传统的存储环境上用过的话，你会发现这些功能很眼熟。不过传统存储环境要使用 VASA 功能，需要对特定的存储进行一些额外的配置来添加**存储提供程序**。而对于 VSAN 来说，vSphere 管理员不需要担心**存储提供程序**，因为它们在 VSAN 群集创建时已经自动注册好了。

4.2.1 VASA 简介

VASA 允许存储供应商把它们存储的功能发布到 vCenter Server 并在 vSphere Web 客户端中显示出来。VASA 还可以提供诸如存储健康状态、配置信息、容量以及精简置备情况等信息，使 VMware 得以感知存储内部的运行状况。以前，VASA 和传统存储是这样工作的：存储阵列将其功能告知给 VASA 存储提供程序，然后存储提供程序再通知 vCenter Server，最后用户就可以在 vSphere Web 客户端上看见存储阵列的这些功能。通过虚拟机存储策略，这些存储的功能得以浮现在 vSphere Web 客户端的界面上，帮助管理员从空间、性能和服务水平协议（SLA）等方面来选择正确的存储。对于传统存储阵列是这样的，现在对于 VSAN 也是。不过，在使用 VASA 和虚拟机存储策略的流程上，传统存储和 VSAN 还是有一个显著的区别：对于传统存储，VASA 只是把数据存储的功能展现出来，vSphere 管理员还是必须自己选择合适的存储来放置虚拟机；而对于 VSAN，你在虚拟机存储策略中定义好虚拟机存储所需要的那些功能，随后这个策略被推送给 VSAN，通知它这就是对存储的要求。然后 VASA 会告诉你 VSAN 是否能满足这些要求，它针对每个存储对象，有效地传递合规性信息。主要区别在于现在此功能是双向模式的，而以前 VASA 只是提供功能信息（单向的），现在它不仅展现功能信息，还验证虚拟机的要求是否根据策略的内容被满足了。

4.2.2　存储提供程序

存储提供程序看上去是什么样的？请看图 4-4。当一个 VSAN 群集被创建时，群集中每台 ESXi 主机就会在 vCenter Server 上注册 VASA 存储提供程序。在一个 4 节点的 VSAN 群集上的 VASA VSAN 存储提供程序配置就类似如此。

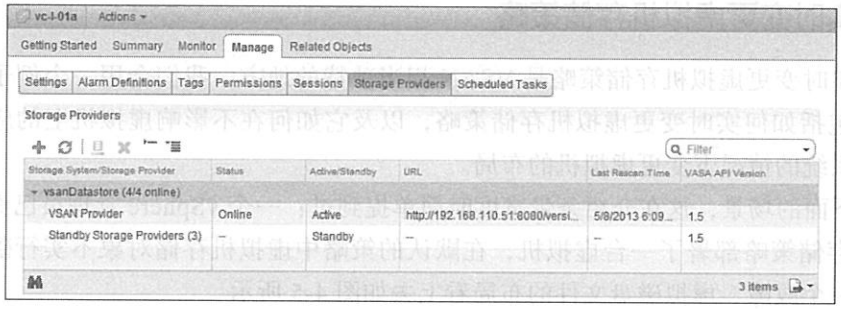


图 4-4　VSAN 群集被创建时添加的 VSAN 存储提供程序

要查看存储提供程序的状态，打开 Web 客户端，导航到 vCenter Server 清单，选择 Manage（管理）标签页，然后选择 Storage Provider（存储提供程序）视图。应该总是有一个 VSAN Provider 处于联机状态，其他存储提供程序应该处于备用状态。

在超过 8 台 ESXi 主机的 VSAN 群集中，用户界面中就会有超过 8 个 VASA 存储提供程序。为了显示更清晰，列表会被缩短到只显示 8 个存储提供程序。备用存储提供程序的数量仍然会正确显示，只是你不能对它们进行查询。

4.3　VSAN 存储提供程序：高可用

你可能会问为什么每台 ESXi 主机都要注册存储提供程序，原因是高可用性。如果一台 ESXi 主机发生故障，群集中的另一台 ESXi 主机可以接管这些 VSAN 功能发布的任务。如果再回头看一下图 4–4 所示的存储提供程序，你会发现仅有一个 VSAN 提供程序是联机（online）的，来自这个 4 节点群集中其他 3 台 ESXi 主机的存储提供程序都处于备用（standby）状态。如果当前联机的存储提供程序掉线或出现故障，不管这是什么原因造成的（多半是因为主机故障），备用提供程序中的一个会被提升为活动（active）状态。

对于 vSphere 管理员创建 VSAN 群集的任务来说，基本上不需要对存储提供程 序进行什么操作，这里只是给你提供一个参考。然而，如果真的遇到了 VSAN 的功能无法在虚拟机存储策略上显示出来的问题，那就有必要到这个配置页来看一看，确认一下至少有一个存储提供程序是活动的。如果不存在活动的存储提供程序，那么在试图创建一个虚拟机存储策略的时候，你就无法发现任何 VSAN 功能。此时，作为排错的一个步骤，你可以考虑点击存储提供程序页面上的刷新图标（橙色环状箭头）来刷新一下存储提供程序。

应该注意的是，VASA 存储提供程序完全不参与 VSAN 的数据传输路径。如果存储提供程序发生了故障，完全不会影响运行在 VSAN 数据存储上的虚拟机。失去存储提供程序的影响仅仅是丢失了对相关功能的可见性，所以你将无法再创建新的存储策略。然而已经在运行的虚拟机和策略则不受影响。

4.3.1 实时变更虚拟机存储策略

可以实时变更虚拟机存储策略是 VSAN 相当独特的地方。我们会用一个例子来解释这个概念，包括如何实时变更虚拟机存储策略，以及它如何在不影响虚拟机上的应用程序或客户操作系统的情况下变更虚拟机的布局。

考虑下面的场景，这在介绍条带宽度时简单提到过：一个 vSphere 管理员已经通过默认的虚拟机存储策略部署了一台虚拟机，在默认的策略中虚拟机存储对象不实行磁盘条带化并能容忍一个故障。虚拟磁盘文件的布局看上去如图 4-5 所示。

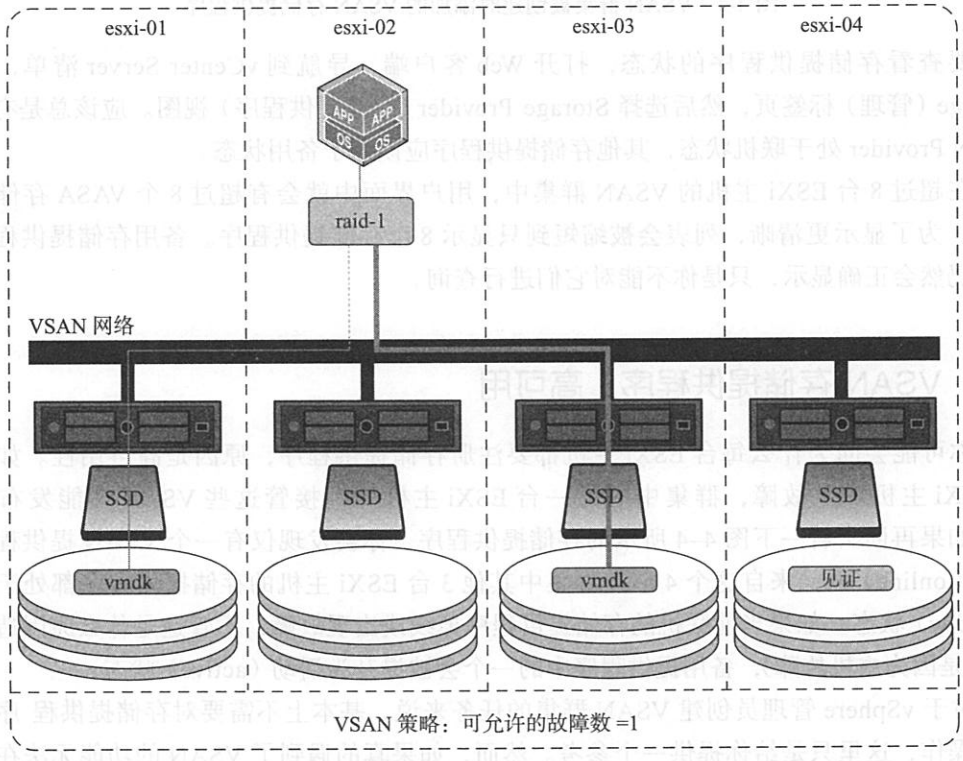


图 4-5 允许的故障数 = 1 时的 VSAN 策略

这台虚拟机和其上的应用程序开始时表现不错，运行得很令人满意，读缓冲命中率达到了 100%。然而，随着时间的推移，加入到 VSAN 群集的虚拟机数量越来越多。vSphere 管理员开始注意到部署在 VSAN 上的虚拟机现在只能达到 90% 的读缓冲命中率了。这意味

着 10% 的读操作需要由磁盘来提供服务。在峰值时这台虚拟机每秒有 2000 次读操作，因此有 200 次读取需要由磁盘来提供服务（就是缓冲没有命中的 10% 读取）。磁盘的性能为每块磁盘 150 IOPS，这意味着单块磁盘无法满足额外的 200 IOPS。为了满足虚拟机对 I/O 的要求，vSphere 管理员做出了正确的决定：创建一个跨 2 块磁盘的 RAID-0 条带。

在 VSAN 上，vSphere 管理员有 2 个选择来实现之。

第一个选择是，简单地修改现有的关联到这台虚拟机的虚拟机存储策略，添加一个条带宽度的要求。然而，这个方法会更改所有使用这个策略的虚拟机的存储布局。

另一个方法是，创建一个全新的策略，这个策略和之前的策略完全一样，只是额外再多加一个条带宽度的功能。然后将这个新策略附加到受缓冲无法命中困扰的虚拟机上。当这个新策略关联上了虚拟机，管理员可以促发一次新的 / 更新过的策略与虚拟机的同步。VSAN 会负责更改底层的虚拟机存储布局使之满足新策略的要求，而更改的同时虚拟机仍在运行。实现方法是在原先的存储对象仍然存在的同时，构建新的存储对象，并加入额外的组件（在这个例子中 RAID-0）。

更改虚拟机存储策略的方法有两种：要么是编辑现有的虚拟机存储策略并加入条带宽度等于 2 的新功能；要么是创建一个全新的虚拟机存储策略并包括**允许的故障数** =1 和**条带宽度** =2 这 2 个参数。你可能更希望使用后者，因为最初的策略可能会有其他虚拟机在使用，编辑那个策略会影响所有使用它的虚拟机。当新策略创建出来时，它可以同此虚拟机及其存储对象相关联，这在 Web 客户端中有很多方法来实现。事实上，如果必要的话可以在每一个虚拟机存储对象的粒度上来更改策略。

变更完成后，代表新配置的新组件（例如 RAID-0 条带）将进入重新配置中的状态。这在保持原有对象的同时，会临时构建一个额外的对象，所以 VSAN 数据存储上必须有额外的空间来承载这种实时变更。当新对象准备完毕，新的配置也完成之后，原有对象将被丢弃。

现在虚拟机存储对象可以反映出 Web 客户端中的变更了——例如，同时具备 RAID-0 条带和 RAID-1 副本配置的存储对象如图 4-6 所示。

作为对比，让我们看看很多传统存储阵列上要实现同样的目的所需进行哪些操作。这至少包括以下这些步骤：

- 从原有数据存储迁移虚拟机
- 删除所涉及的 LUN 或卷
- 创建一个满足新的存储要求的新的 LUN（不同的 RAID 等级）
- 对于块存储来说，还可能需要重新以 VMFS 格式来格式化 LUN

最后，你还必须把虚拟机重新迁移回新的数据存储。在新的存储对象创建好并同步完成后，旧的存储对象将自动被移除。注意，在需要的时候 VSAN 的条带化可以跨磁盘、磁盘组和主机，例如在图 4-6 所描绘的例子中，条带 S1a 和 S1b 位于同一台主机上，但是条带 S2a 和 S2b 位于不同的主机上。

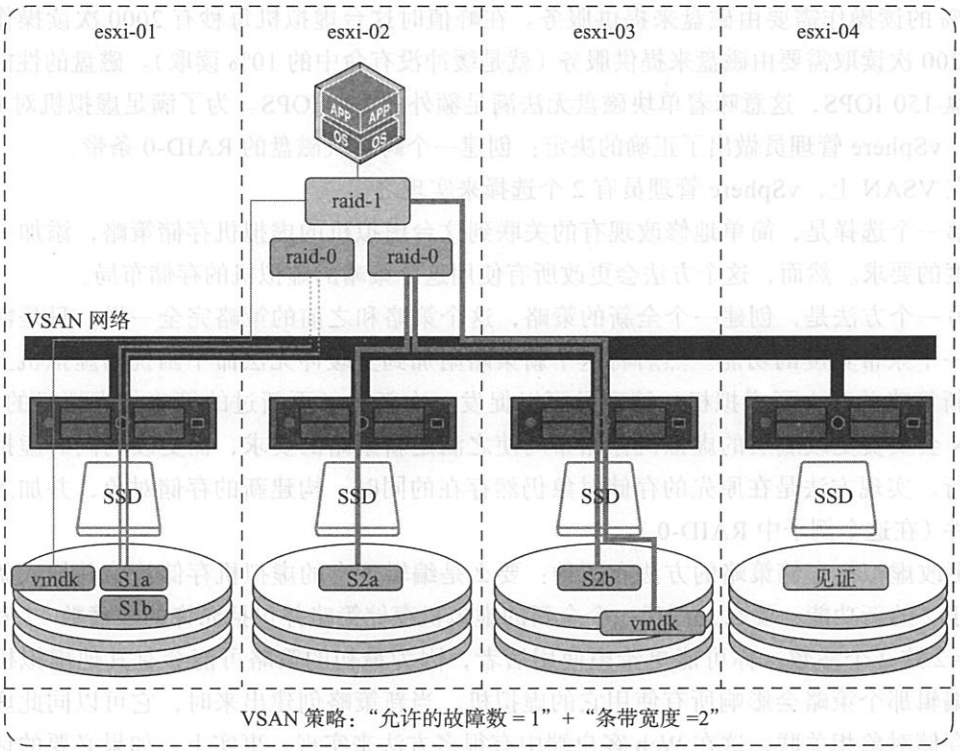


图 4-6 VSAN RAID-0 和 RAID-1 配置

我们还未曾提及的是，对于这样的配置更改，是一定会产生额外的见证对象的。对于一台能持续不断地访问其组件的虚拟机，必须在群集中保证有超过 50% 的存储对象的组件是可用的。因此，对虚拟机存储策略的变更会导致额外的见证组件被创建出来。

通过下面的过程，你实际上可以在 vSphere 用户界面中看见配置变更的发生过程。选择要变更的虚拟机，点击 Manager（管理）标签，然后选择 VM Storage Policies（虚拟机存储策略）视图，如图 4-7 所示。尽管这个视图没有显示所有的虚拟机存储对象，它确实已经显示了 VM Home（虚拟机主页）名字空间，并且可以看见 VMDK。

4.3.2 对象、组件和见证

到目前为止，本章已经介绍了很多新概念，包括一些新的术语。第 5 章中，还将覆盖很多更深入的内容，包括对象、组件，当然还要介绍见证磁盘，包括在虚拟机存储策略中一些特别功能是如何影响虚拟机存储对象的。现在只需要理解对于 VSAN，一台虚拟机不再是一组文件的集合，而是一组存储对象的集合即可。VSAN 有四种类型的存储对象：

- 虚拟机主页名字空间（VM Home namespace）
- 虚拟磁盘（VMDK）
- 虚拟机交换文件（VM swap ）

❑ 增量盘（delta disk）

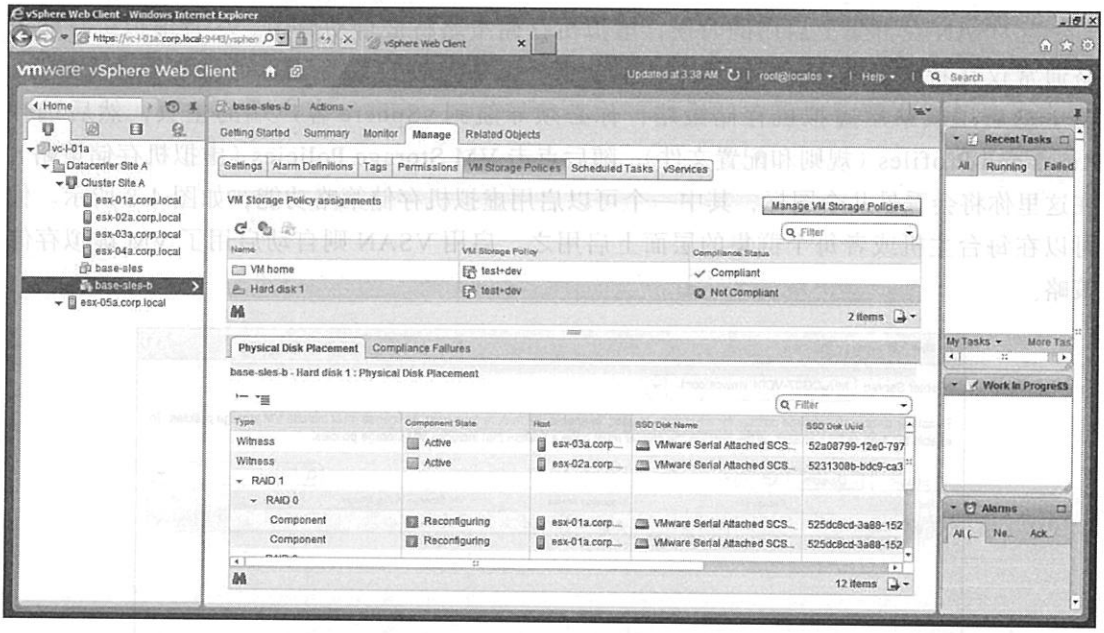


图 4-7　在 vSphere Web 客户端的虚拟机存储策略视图中显示组件的重新配置

尽管在 vSphere Web 客户端中仅显示虚拟机主页名字空间和 VMDK（硬盘），我们将在第 10 章介绍 VSAN 各种可用的不同监控工具时，告诉你如何查看其他存储组件（也就是 delta 和虚拟机交换文件）的方法。

4.4　虚拟机存储策略

虚拟机存储策略和 vSphere 5.0 引入的虚拟机配置文件的工作模式一模一样，简单说就是创建一个策略来包含虚拟机置备的要求。与原先的存储配置文件特性比较起来，其主要的区别在于存储等级（storage level）是如何起作用的。就存储配置文件来说，只需要在置备虚拟机的时候简单地使用策略中的要求来选择合适的数据存储即可。然而就存储策略而言，就不仅仅是选择合适的数据存储这么简单，还要求通知底下的存储层关于这台虚拟机特定的可用性和性能的要求。所以即使 VSAN 数据存储是目的存储，如果虚拟机是通过虚拟机存储策略置备的，策略中的设置可能会引发额外的要求。例如可能的陈述是这样的：这台虚拟机具有为保证可用性需要多个虚拟机文件的副本的要求、为保证高性能对条带宽度和读缓冲的要求，以及精简置备的要求。

虚拟机存储策略既保存在 VSAN 中，也保存在 vCenter 的清单数据库中。每个对象都将其策略保存在其自己的元数据中，这意味着 vCenter 并不是部署虚拟机存储策略的必要条件，因此即使因某些原因 vCenter Server 不可访问，策略仍然可以继续起作用。

4.4.1 启用虚拟机存储策略

当 VSAN 在群集上启用的时候，虚拟机存储策略将被自动启用。尽管虚拟机存储策略通常仅在特定的 vSphere 版本中才能获得，添加 VSAN 许可证也可以获得这个特性。要手动启用或禁用虚拟机存储策略，你必须导航到 vSphere 客户端的主页，然后选择 Rules and Profiles（规则和配置文件），随后点击 VM Storage Policies（虚拟机存储策略），在这里你将会看见几个图标，其中一个可以启用虚拟机存储策略功能，如图 4-8 所示。你可以在每台主机或者每个群集的层面上启用之。启用 VSAN 则自动启用了 VM 虚拟存储策略。

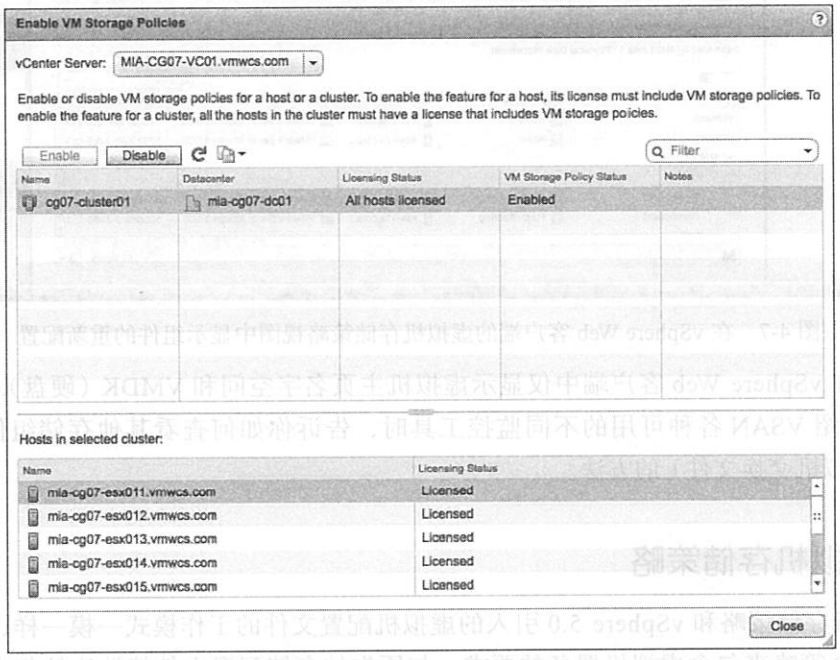


图 4-8 当 VSAN 启用时虚拟机存储策略自动被启用

注意 禁用 VSAN 不会自动在群集上禁用虚拟机存储策略。若想禁用虚拟机存储策略，用户仍然必须手动禁用。

4.4.2 创建虚拟机存储策略

一旦虚拟机存储策略启用后，这个窗口中的另一个图标使得 vSphere 管理员可以创建单独的策略。前面我们曾经提过，很多关于可用性和性能的 VASA 功能都是通过 VASA 展现出来的，在这里管理员必须决定虚拟机中运行的应用程序到底对性能和可用性有什么要求。例如，管理员要求这个虚拟机在保持运行时能容忍多少个组件（主机、网络和磁盘）发生故

障？又比如运行在虚拟机中的应用程序对 IOPS 的要求高不高？如果高的话，应该给这台虚拟机提供多少读缓存才是合适的，才能满足对性能的要求？其他考量因素还包括对这台虚拟机应该采用精简置备还是厚置备。

另一个值得注意的地方是，vSphere 5.5 还支持使用标记来进行置备。因此，除了使用 VSAN 数据存储的功能来创建虚拟机存储策略以满足要求外，还可以创建基于标记的策略。使用基于标记的策略超出了本书讨论的范畴，但是你或许可以从 vSphere 存储的文档集中找到更进一步的信息。

4.4.3　在虚拟机置备时分配虚拟机存储策略

虚拟机存储策略的分配是在虚拟机置备过程中完成的。在 vSphere 管理员选择目的数据存储时，他必须从可用的虚拟机存储策略下拉菜单中选择合适的策略。随后数据存储会被分成兼容和不兼容数据存储两类，从而得以让 vSphere 管理员为虚拟机放置做出合适且正确的选择。

匹配数据存储并不意味着数据存储必须满足虚拟机存储策略的要求，这只是意味着数据存储了解了这一组要求已经置于策略之中。如果没有足够的资源来满足策略的要求，虚拟机的置备仍然有可能会失败。

我们在 4 节点群集的例子中创建了一个包含允许的故障数等于 2 的策略。4 节点的群集无法满足这个策略的要求，但是当策略创建的时候，VSAN 数据存储会显示为一个可匹配的资源，并且，在使用这个策略置备虚拟机时，VSAN 数据存储也会显示为合规的，如图 4-9 所示。

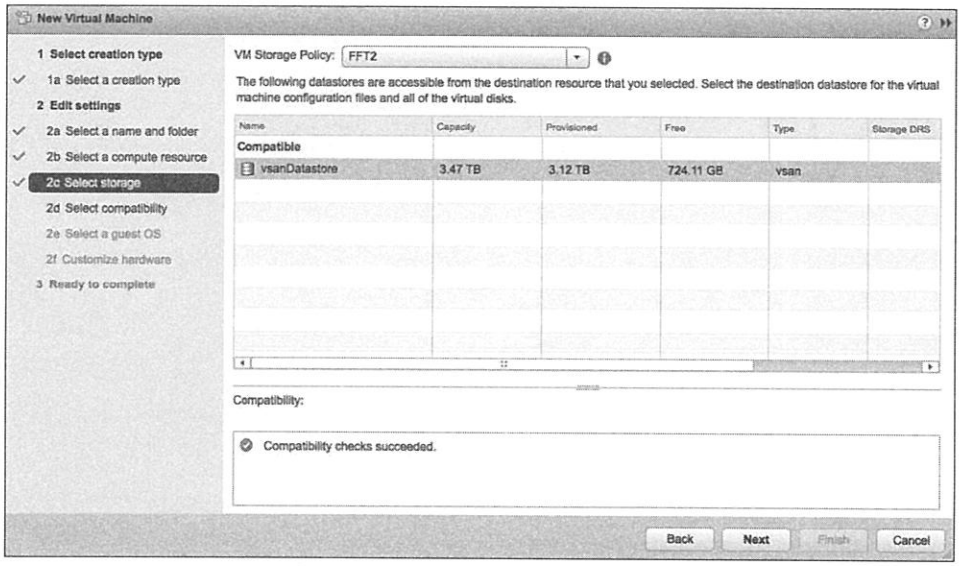
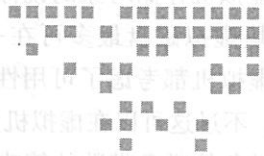


图 4-9　即使策略不被满足，VSAN 数据存储仍然显示为合规

但是如果继续部署虚拟机，虚拟机的创建操作会失败。这是一个值得记住的重点：仅仅因为 VSAN 告诉你某个特定的策略是合规的，绝对不意味着你可以用这个策略成功部署一台虚拟机。

4.5 小结

你以前可能已经用过虚拟机存储**配置文件**。虚拟机存储策略与之有着显著的区别。尽管我们仍然使用 VASA——存储感知的 vSphere API，虚拟机存储策略使我们可以把存储 QoS 从数据存储转到虚拟机上。虚拟机（或者更进一步来说运行在虚拟机上的应用程序）现在可以提出它们自己特定的策略需求，这包含底层存储关于性能、可靠性和可用性的能力。



第 5 章 Chapter 5

架构细节

本章讲述了 Virtual SAN 底层的架构细节。我们已经提起过不少 VSAN 架构的内容，包括使用闪存作为 I/O 的缓存、将 VSAN 功能展现出来的 VASA 角色、虚拟机存储策略、见证盘以及对于直通 RAID 控制器的需求等。

本章将深入探讨这些特性，并介绍由 VSAN 引入的一些新的架构概念和术语。尽管大多数 vSphere 管理员不会接触这些底层的结构，但是对组成 VSAN 的服务具有一些大致的了解对排错或分析日志文件还是有用的。在探讨这些底层细节之前，让我们首先来介绍 VSAN 的一个核心概念：分布式 RAID/RAIN。

5.1 分布式 RAID

VSAN 使用分布式 RAID，为虚拟机提供高可用性和最佳性能。从可用性角度来说，分布式 RAID 意味着 VSAN 环境可以容忍一台或多台 ESXi 主机（或主机上的组件例如磁盘）故障而继续为其上所有的虚拟机提供其全部功能。而为了确保虚拟机性能最佳，VSAN 分布式 RAID 提供了将虚拟磁盘散布到多个物理磁盘和主机上去的能力。

然而，值得说明的一点是，通过使用存储策略，虚拟机的可用性和性能现在可以针对单台虚拟机来设置，甚至可以针对单块虚拟磁盘来设置。管理员可以通过存储策略来定义 VSAN 群集中的一台虚拟机可以容忍多少主机故障或者多少磁盘故障，并可以定义一块虚拟磁盘可以散布到多少主机和磁盘上。如果选择不在存储策略中配置可用性要求，那么主机或磁盘的故障就肯定会影响虚拟机的可用性。

VSAN 在主机之间使用 RAID-1（同步镜像）来满足对系统中存储对象的可用性和可

靠性的要求。虚拟机存储对象的镜像拷贝数量取决于虚拟机存储策略。根据虚拟机存储策略的不同，一块虚拟磁盘最多可在一个 32 节点的 VSAN 上拥有 3 个镜像。默认情况下，VSAN 部署的虚拟机都考虑了可用性因素——VSAN 上部署的每台虚拟机的存储对象都有一个镜像拷贝。不过这可以在虚拟机置备时通过选择不同的策略来改变。

根据每个对象的磁盘带数的策略设置，一个虚拟磁盘对象可能会被条带化到很多物理磁盘上来达到期望的性能要求。可以通过 RAID-0 增强虚拟机存储对象的性能，不过条带配置并不总是增强性能的必要条件。在本章稍后我们将解释原因，并说明在什么时候在虚拟机存储策略中增加 VMDK 的条带宽带可以带来性能提升。

你或许会奇怪为什么 VMware 不使用更能有效利用空间的方法，诸如 RAID-5 或 RAID-6，而要使用 RAID-1？使用 RAID-1 背后的原因是 RAID-5 和 RAID-6 使用的是“读 – 变更 – 写”的方法，如果写操作不能填充一个完整的条带宽度，很多写就会要求额外的磁盘读操作，这可能会要求更多的（而且很可能更小的）磁盘驱动器来保持稳定的存储性能。这增加了系统的总拥有成本。

5.2 对象和组件

VSAN 数据存储是一种**对象存储系统**，虚拟机是由大量不同的存储对象组成的，而不像之前是由一组文件组成的。理解这一点很重要。

到目前为止我们还未曾详述对象和组件。所以，在深入不同对象的各种细节之前，让我们先从 VSAN 中对象和组件的定义及概念讲起。

对象指的是一个独立的存储块设备，它与 SCSI 语义兼容。它可以根据需要来创建，并且大小没有限制，不过在 VSAN 的最初发行版本中 VMDK 的大小上限是 2TB 减 512 字节。对象现在取代 LUN 成了 VSAN 的主要存储单元。在 VSAN 中最典型的存储块设备就是独立的 VMDK、虚拟机主页名字空间、虚拟机交换文件和增量盘（如果 VMDK 拍过快照的话）。VSAN 中的每个“对象”都有其自己的 RAID 树，将策略要求映射成物理设备上实际的布局。如果你在部署虚拟机的时候选择了某一个虚拟机存储策略，那么策略中关于可用性和性能的这些要求就会被应用到虚拟机对象上。

组件是对象的 RAID 树上的叶子——这意味着，一“片”组件是存放在一个特定的“闪存设备 + 磁盘”的组合（一个物理磁盘组）上的。组件通过闪存获得了透明[⊖]的缓冲 / 缓存能力，其数据则静静地躺在磁盘上。

在 VSAN 数据存储上，虚拟机具有 4 种不同类型的对象，每台虚拟机都可以由这些对象中的部分组合而成，这些对象如下：

- 虚拟机主页（VM Home）或“名字空间目录”（namespace directory）

⊖ 对于存储组件来说是，闪存所提供的缓冲 / 缓存功能是透明的，也就是说，组件不关心也不知道自己的数据是通过闪存中的缓存 / 缓冲提供的还是直接访问磁盘得来的。——译者注

- 交换文件对象（如果虚拟机处于开启状态）
- 虚拟磁盘 /VMDK
- 增量盘（建立快照之后每个对象都有自己的增量盘）

另有一个组件叫做见证（witness），它非常重要，也很特殊。尽管它不直接给虚拟机提供存储空间，但是不可否认的是，当群集中的虚拟机存储对象出现故障的时候，作为必要的仲裁对象，它是非常关键的。稍后我们还会讨论见证组件，现在让我们先关注虚拟机存储对象。

在这4种对象中，虚拟机名字空间需要进一步解释一下。所有虚拟机文件，包括VMDK、增量（快照）和交换文件都存放在VSAN上一块叫做**虚拟机名字空间**的地方。在虚拟机主页名字空间中最典型的文件有 .vmx 虚拟机描述文件、.log 日志文件、.vmdk 磁盘描述文件、快照增量盘描述文件以及所有其他可能在虚拟机主页目录中找到的文件。

那么组件是怎么回事？ VSAN上的每个存储对象都可以被看成一棵RAID树，每片树叶就是一个组件。例如，如果一个VMDK的条带宽带是2，那么这个VMDK上就会配置一个RAID-0条带并横跨2块磁盘。这个VMDK是一个对象，组成它的每一片条带就是这个对象的一个组件。

类似地，如果定义了这块VMDK在群集中应该至少容忍一个故障（主机、磁盘或网络），就要对这个VMDK对象创建一个RAID-1镜像——在群集内的一台主机上有一个副本组件，同时在另一台主机上有另一个副本组件。最后，如果同时需要配置条带和可用性，那么条带组件将会在主机之间被镜像，最终形成了RAID0+1配置。

注意，增量盘是在对虚拟机拍摄快照的时候创建出来的，它会继承其母盘的策略（条带宽度、副本数等）。

交换文件对象是在虚拟机开机的时候创建的。

5.2.1 组件的限制

关于VSAN中的组件有两个相关的限制。因为这两个限制都是硬性限制并且最终会影响群集中单台主机上可以运行的虚拟机数量，理解这些限制是非常重要的。VSAN 1.0的限制如下：

- **每台主机的最大组件数**：3000
- **每个对象的最大组件数**：64（包括条带和副本）

主机的组件包括处于关闭状态的虚拟机的组件。VSAN将这些组件分派到群集中的各台主机上，并试图保持一种均衡的分布。然而，某些主机拥有的组件仍然有可能比其他一些主机更多。所以这就是为什么VMware推荐的最佳实践是使得一个VSAN群集的所有主机尽可能地保持类似（甚至是完全一致）的配置。在设计和部署一个VSAN群集的时候，组件的数量限制是设计和部署VSAN群集时重要的决策因素，这将会在第9章中进一步深入探讨。

管理员可以使用 vSphere Web 客户端查看虚拟机主页名字空间和虚拟机的 VMDK 等对象及其组件的布局，图 5-1 提供了一个此类布局的示例。这台虚拟机有一块硬盘，这块硬盘在 2 台不同的主机上拥有镜像，你可以在 Host 这一列看见这些组件[⊖]的位置。

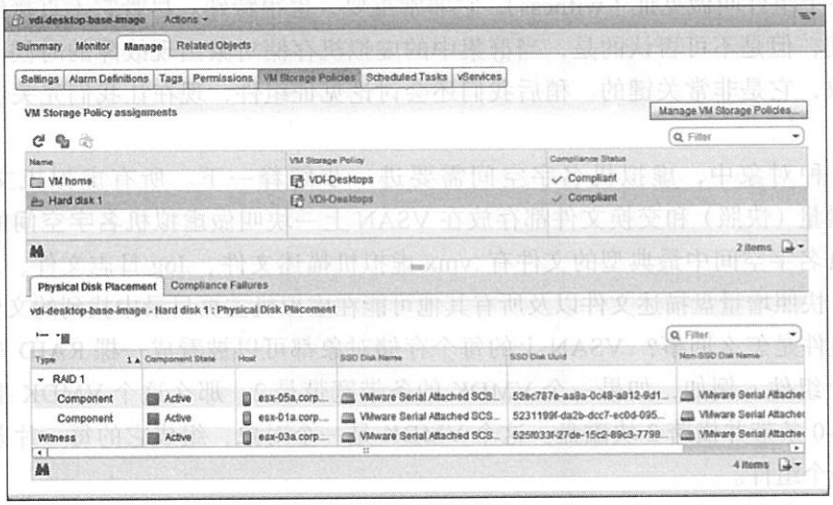


图 5-1　物理磁盘布局

5.2.2　虚拟机存储对象

如前所述，4 种存储对象是：虚拟机主页名字空间、虚拟机交换文件、VMDK 和增量盘，如图 5-2 所示。

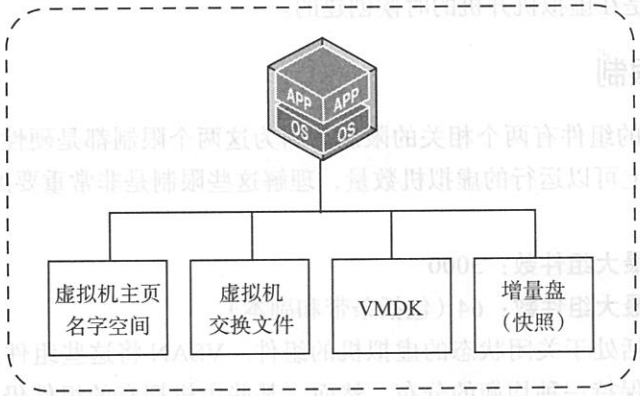


图 5-2　虚拟机存储对象

现在我们将讨论虚拟机存储策略中定义的特性将如何影响这些存储对象。注意，并非

⊖ 这些组件指的是组成这一个 RAID-1 镜像的 VMDK 对象的副本（replica）组件和见证（witness）组件。

所有的虚拟机存储对象都会实施这些策略。

5.2.3 虚拟机主页名字空间

名字空间包含不属于特定对象的其他所有内容，如以下这些内容（但不仅限于此）：

- .vmx 虚拟机描述文件、.vmdk 磁盘描述文件部分、VMX 使用的 .log 日志文件
- 用于 VMware Horizon View 的 CBRC[⊖]的摘要文件
- 内存快照（拍摄快照时生成的）
- vSphere Replication 和 Site Recovery Manager 的文件
- 客户自定义的文件
- 由其他软件解决方案产生的文件

VSAN 使用 VMFS 作为每台虚拟机的名字空间对象的文件系统，来存放有关虚拟机的文件。这是一个功能完整的普通的 VMFS，也就是说它具有完整的支持群集的能力，我们可以将它用在那些依赖 VMFS 的功能中（例如 vMotion、vSphere HA）。如果你去每台 ESXi 主机查看文件系统，你会发现它表现为一个自动挂载的子目录。然而，尽管用起来和普通 VMFS 一样，它却没有其他环境中的那些 VMFS 所具有的那些限制，因为连接到这些虚拟机主页名字空间 VMFS 卷的主机数量最多是 2 台（例如，在 vMotion 发生时），而在传统环境下同样的 VMFS 卷往往是同时被几十台主机共享的。换而言之，VSAN 对这些“普通”的 VMFS 卷的用法是完全不同的，它可以具有更高的扩展性和更佳的性能。

用于虚拟机主页的虚拟机存储策略是特殊的。虚拟机主页存储对象的大多数需求和 VMDK 都不一样。设想一下，像虚拟机主页名字空间这样的对象需要什么闪存当读缓冲或者条带功能吗？不需要！这就是为什么即便策略中声明这些能力也不起作用的原因。不过有一个功能的确是生效的——允许的故障数。这个功能使虚拟机得以在群集中多个硬件发生故障的时候仍然可用。由于对于虚拟机主页存储对象来说，高性能不是主要诉求，继承下来的 VMDK 的设置会被覆盖掉，条带宽度（Stripe Width）总是设成 1，读取缓存预留（Read Cache Reservation）总是设成 0%，而且对象空间预留（Object Space Reservation）被设成 0%，所以它总是精简置备的。这使得虚拟机主页名字空间不会无谓消耗资源，而把可用资源留给那些需要的对象，例如 VMDK。

另外一个要点是，如果策略中设置了强制置备（Force provisioning），虚拟机主页名字空间对象会继承这个属性，这意味着，即使资源总量不足，虚拟机也会被部署。

5.2.4 虚拟机交换文件

对于虚拟机交换文件对象（VM Swap），默认的策略中允许的故障数是 1。无须将对象空间预留（Object Space Reservation）设置成 100%，交换对象也会被事先 100% 的置备好。

⊖ CBRC 是 Content-Based Read Cache 的缩写，意思是基于内容的读缓冲。——译者注

从接入控制的角度来说，这意味着如果没有足够的磁盘空间来生成虚拟机交换对象，VSAN将无法部署虚拟机。

5.2.5 VMDK 和增量盘

现在你知道了，当虚拟机部署的时候，虚拟机主页名字空间和虚拟机交换文件（.vswp）有其自己的默认策略。因此，只有 VMDK 和这些磁盘文件的快照文件（增量盘）才遵循设置在虚拟机存储策略中的那些属性。

因为 VSAN 对象有可能是由多个组件构成的，部署的时候每个 VMDK 和增量盘都会有其自己的 RAID 树配置。

5.2.6 见证和副本

作为 RAID 树的一部分，每个对象通常都有几个副本（副本也是一种组件）。我们曾经提起过，一个或多个见证组件**可能**会随着虚拟机存储对象的创建而创建。在 RAID 树中，见证是每个对象的一部分。它是组成 RAID 树的叶子，但是不含数据，只包括元数据（metadata）。见证是当 VSAN 群集中发生了故障后进行仲裁决断的时候用来打破平局的裁判。

让我们用一个最简单的例子来解释下这么设计的目的：假设要部署的虚拟机的配置是条带宽度为 1 且允许的故障数为 1。这种情况下，需要为每台虚拟机创建 2 个副本。因此，RAID-1 设置就足够了。然而，在 2 个副本的情况下，如果主机之间失联，将无法分辨这到底是主机故障还是网络分区的情况。因此，需要在配置中引入一个第三方，这就是见证。VSAN 中的一个对象要被认定为可用，必须满足以下两个条件：

- RAID 树必须允许数据访问（RAID-1 必须至少有一个完好的副本，RAID-0 必须所有的条带都完好）。
- 必须有超过 50% 的组件可用。

在前面的例子中，只有当能同时访问一个副本和一个见证，或者同时访问两个副本的时候，才能够访问这个对象。这样，在出现网络分区的情况下，至少有部分群集可以访问这个对象。

一个常见的问题是，见证是否消耗 VSAN 数据存储的空间？每个见证组件在 VSAN 数据存储上大概会占用 2MB 空间用来存储元数据。尽管大多数空间是被虚拟机及其磁盘所占据，相对而言见证占据的空间很小，但是在做容量规划和扩展性考量时，这也是需要考虑的因素之一。

5.2.7 对象布局

另一个常常被提起的问题是，VSAN 环境中的对象到底是怎么分布的？如前所述，虚

拟机主页空间是用来存储虚拟机配置文件的，它是 VMFS 格式的。所有其他虚拟机磁盘对象（不论是 VMDK 还是快照）都以其各自的方式被实例化成一种分布式存储对象。

尽管我们觉得没必要去关心对象到底存在什么地方，我们能够理解你的心情，对于这样一种新型的解决方案，你可能希望会对对象存放的物理位置有一个更好的了解。VMware 觉得管理员可能会有这样的期望，因此，vSphere 用户界面中有地方可以让管理员来查看虚拟机对象的布局，并可看见组成一个存储对象的每个组件（条带、副本、见证）存放的位置，如图 5-3 所示。

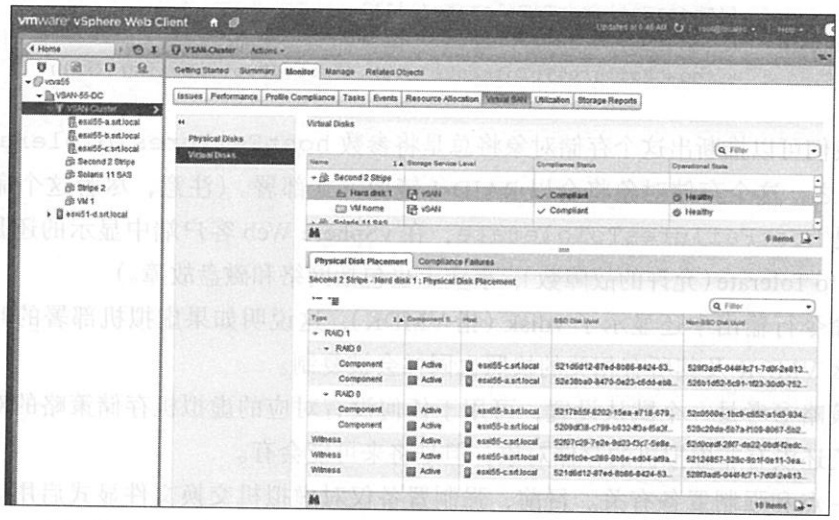


图 5-3 RAID-1、RAID-0 和见证

出于可用性的考虑，VSAN 绝不会让不同的副本（镜像）组件共用同一台主机。

注意，这里我们看不见虚拟机交换文件对象。交换文件 UUID 目前无法通过 VIM 应用程序可编程接口来获得，因此无论是 Ruby vSphere Console（RVC，将在第 10 章介绍）还是 vSphere Web 客户端都无法显示其信息。不过，有一个办法可以获取交换文件的信息，稍后会演示。快照 / 增量盘对象在 vSphere 用户界面中也不可见，不过这些对象默认是继承其 VMDK 母盘的策略配置的（也就是说其对象分布和其 VMDK 母盘是一致的）。

我们已经说过不少关于虚拟机存储策略的概念了，现在让我们再深入一步。

默认虚拟机存储策略

VMware 鼓励管理员们不要依赖默认策略的设置而是去创建自己的 VSAN 策略。不过，如果你决定在 VSAN 数据存储上部署虚拟机时不选择任何策略，那么默认策略就会被应用。默认策略有一些非常特殊的特性，当管理员在（不管是什么原因）没有选择策略的情况时，它可以防止虚拟机及其相关数据置于风险之中。这种情况相当常见，我们就见过很多次管理员在匆忙中创建了虚拟机但是忘记了选择策略的情况。不过，的确需要强调的是，VMware 强烈建议管理员们创建自己的虚拟机存储策略，即使需求和默认策略完全一致。这

仅仅只是因为一点：自建策略使得管理员可以获取有意义的合规性检查报告。VMware 还强烈建议管理员不要去编辑或变更默认策略。不过在某种必要情况下可能需要更改，我们很快会对此进行探讨。

默认策略可以通过 esxcli 命令行来观察，让我们来试一下：

```
~ # esxcli vsan policy getdefault
Policy Class  Policy Value
------------  ------------------------------------------------------
cluster       (("hostFailuresToTolerate" i1))
vdisk         (("hostFailuresToTolerate" i1))
vmnamespace   (("hostFailuresToTolerate" i1))
vmswap        (("hostFailuresToTolerate" i1) ("forceProvisioning" i1))
~ #
```

由此我们可以推断出这个存储对象将总是将参数 hostFailuresToTolerate 设置为 1。换而言之，这个存储对象将会以 RAID-1 镜像方式部署。（注意，尽管这个命令行界面 CLI 中说是 hostFailuresToTolerate，在 vSphere Web 客户端中显示的还是 Numbers of Failures to Tolerate（允许的故障数），事实上也包括网络和磁盘故障。）

注意命令行输出中还显示了 vdisk（指 VMDK），这说明如果虚拟机部署的时候未选择策略，那么 VMDK 及其所有相关的快照也同样会被复制。

群集策略参考是一个默认设置，可用于任何没有对应的虚拟机存储策略的对象。此时此刻，我们还没有非虚拟机的存储对象，但是将来可能会有。

最后一点和强制置备有关。目前，强制置备仅对虚拟机交换文件显式启用。如果你想在一个单主机的 VSAN 群集上引导一台 vCenter Server，在当前配置下是无法实现的。这是因为默认策略中 hostFailuresToTolerate 是 1，如果只有一台主机可用，VSAN 将无法满足这个条件，所以虚拟机将无法创建。要想使之可行，必须更改默认策略，将所有对象类型 forceProvisioning 都改成 1。可以用下面的命令行来实现：

```
~ # esxcli vsan policy setdefault -c vdisk
-p "((\"hostFailuresToTolerate\" i1) (\"forceProvisioning\" i1))"
~ # esxcli vsan policy setdefault -c vmnamespace -p
"((\"hostFailuresToTolerate\" i1) (\"forceProvisioning\" i1))"
```

这会导致默认 VSAN 策略发生如下变化：

```
~ # esxcli vsan policy getdefault
Policy Class  Policy Value
------------  ------------------------------------------------------
cluster       (("hostFailuresToTolerate" i1))
vdisk         (("hostFailuresToTolerate" i1) ("forceProvisioning" i1))
vmnamespace   (("hostFailuresToTolerate" i1) ("forceProvisioning" i1))
vmswap        (("hostFailuresToTolerate" i1) ("forceProvisioning" i1))
```

注意，现在我们已经了解了默认策略，接下去让我们一起来看看管理员可以怎样定义策略。

5.3 VSAN 软件组件

本节将简要介绍组成分布式软件层的一些软件组件。

这些信息大多数都不会在 vSphere 管理员的日常工作中用到。由于只要动几下鼠标就可完成所有的安装配置，VSAN 将纷繁复杂藏于极简的实施背后。不过，就像前面的介绍中曾提过的，你可能会时不时地在 vSphere 用户界面中或是在 VMkernel 日志中看见关于这些组件的消息，我们觉得还是有必要把这些幕后的主要组件提一下，给你一些关于这些组件是如何运作的背景知识。而且，当你开始使用 RVC（第 10 章会介绍）时，大量输出中都会提到这些软件组件，这是为什么我们要在这里简要介绍它们的另一个原因。

VSAN 架构由 4 个主要组件组成，如图 5-4 所示，后面我们会进一步详细探讨。

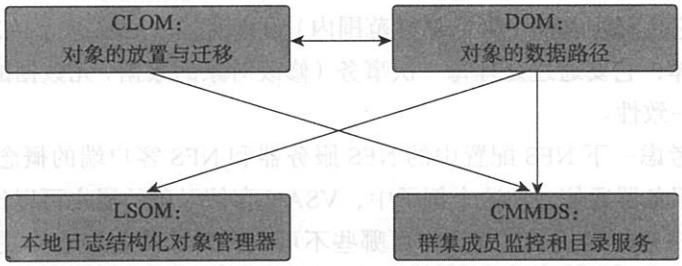


图 5-4 VSAN 的软件组件

5.3.1 组件管理

VSAN 的本地日志结构化对象管理器（Local Log Structured Object Manager，LSOM）作用于物理磁盘层面。VSAN 是靠 LSOM 来给虚拟机存储对象组件提供存储空间的（位于 ESXi 主机的本地磁盘上）。说起组件这个术语，我们指的是组成 RAID-0 的条带组件或组成 RAID-1 的副本组件。因此，LSOM 是作用于 ESXi 主机的磁盘或 SSD 上的。回想一下，SSD 被用作放置在磁盘前端的读缓冲和非易失性写缓存。

关于 LSOM 的另一种表述是：它负责为 VSAN 群集提供存储的一致性。这句话内在的意思是，它存储着组成虚拟机存储对象的组件、所有的配置信息以及虚拟机存储策略。

LSOM 负责上报设备层的事件，如设备的健康状态等。如果设备发生临时性偶发错误时，LSOM 也负责重试出错的 I/O 操作。

LSOM 还能辅助进行对象的恢复。每次 ESXi 主机启动时，LSOM 会进行 SSD 日志恢复，这会引发一次对所有日志的读取来保证内存状态是最新的和正确的。这意味着加入 VSAN 群集的 ESXi 主机比未加入 VSAN 群集的 ESXi 主机在重新启动时要花更多的时间。

5.3.2 对象的数据路径

分布式对象管理器（Distributed Object Manager，DOM）给建立在本地（LSOM）组件

之上的对象提供分布式的数据访问路径。DOM 负责将分布在 VSAN 群集中多台 ESXi 主机上的本地组件创建成可靠的、可容错的虚拟机存储对象，这是通过向存储对象实施分布式 RAID 类型来实现的。

DOM 还负责处理各种不同类型的故障，例如设备无法访问主机的 I/O 问题。当发生了意外的主机故障后进行恢复时，DOM 必须重新同步每个对象的所有组件。一个组件每隔一段时间就发布一次 bytesToSync 值说明同步操作正在进行。

5.3.3 对象的归属

本章中我们时不时会提起对象属主（object owner），让我们再更详细地表述一下什么是对象属主。对于群集中的每一个存储对象，VSAN 都会为其选出一个属主。属主可以被视为是存储机头，它负责协调谁（在 VSAN 范围内）可以对这个对象进行 I/O 操作。属主基本上是这么一个实体，它要通过处理每一次事务（修改对象的数据 / 元数据的操作）来保证分布式对象的数据一致性。

作为类比，考虑一下 NFS 配置中的 NFS 服务器和 NFS 客户端的概念。只有特定的客户端可以成功与服务器通信。在这个例子中，VSAN 存储对象的属主可以比作 NFS 服务器，它决定了哪个客户端可以进行 I/O 操作而哪些不可以。对象属主概念中的最后一部分是组件管理器，它可以被视为 LSOM 的网络前端（换而言之，VSAN 中的一个存储对象是如何被访问的）。

对象属主通过与组件管理器通信来找到 RAID 树上的叶子（即存储对象的组件）。通常情况下只有一个客户端访问对象，然而在 vMotion 操作时，多个客户端可能会访问同一个对象。绝大多数情况下，对象属主和客户端位于 VSAN 群集的同一个节点上。

5.3.4 对象的放置与迁移

群集级别对象管理器（Cluster Lever Object Manager，CLOM）负责保证对象的配置与其策略匹配（也就是说，请求的条带宽度已被实施或者已经置备了足够数量的副本来满足虚拟机的可用性要求）。CLOM 是这样运作的：它接受分配给对象的策略，然后进行大量不同的探索来找到现在群集中可满足策略要求的配置。与此同时，它还保持着 VSAN 中所有节点之间资源利用的平衡。

然后 DOM 实施由 CLOM 给予的配置。CLOM 在群集的不同 ESXi 主机之间分发组件。CLOM 试图形成某种程度上的平衡，不过某些主机比其他主机具有更多组件的情况并不少见，例如已用容量 / 预留容量或已用的闪存读取缓存 / 预留的缓存比其他主机更多。

VSAN 群集上的每个节点都运行着 CLOM 的一个实例。CLOM 的每个实例都对本主机上 DOM 支配的对象的配置和策略合规性负责，因此它需要和 CMMDS（Cluster Monitoring, Membership, and Directory Service，群集监控、成员和目录服务）通信来获取归属转移的情况。CLOM 只和本机的实体通信，它不使用网络。

5.3.5 CMMDS

CMMDS 的目的是发现、建立和维护群集的相互联网的节点成员。它管理着物理群集资源清单，例如主机、设备、网络以及存储对象元数据信息（如策略、分布式 RAID 配置等等），并把它们存放在内存数据库中。对象元数据还总是会保存一份在磁盘上。它还对节点和网络路径的故障发现负责。

其他软件组件通过浏览目录和订阅更新来学习群集拓扑和对象配置的变化。例如，DOM 可以利用目录的内容决定哪个节点存储对象的组件以及决定可到达节点的路径。

注意，仅当主机之间的组播网络连接是全网状的拓扑时 CMMDS 才会构建一个群集（并选举出主控）。

CMMDS 用来选举对象的“属主”。对象的属主会对特定对象进行所有的 RAID 操作，如前所述。

5.3.6 主机角色（主控、备用和代理）

当 VSAN 群集形成时，你可能会注意到 VSAN 群集中的每台 ESXi 主机都具有一个特定的角色（这可以通过 esxcli 命令行来看到）。这些角色仅用于 VSAN 群集服务。群集服务（CMMDS）负责维护一个最新的目录，包括磁盘、磁盘组和 VSAN 群集中每台 ESXi 主机上的对象。这和管理群集中对象或对象的 I/O 操作完全无关，仅仅是允许群集中的节点可以追踪到其他节点。群集服务是基于主控（包括备用）和代理的，所有的节点都会将更新发送给主控，主控然后把这些更新通过顺序可靠的 VSAN 特定的组播协议重新分发给代理。这就是为什么 VSAN 网络必须具有处理组播流量的能力的原因（前面提到过）。角色是在群集发现过程中赋予的，这个过程也是 VSAN 群集中的 ESXi 主机选举主控的时候。vSphere 管理员对于群集成员会获得何种角色没有控制权。

一个常见的问题是为什么需要备用角色？这是因为如果当前主控角色的 ESXi 主机发生了灾难性的故障而没有备用的话，所有 ESXi 主机必须根据新选举出来的主控重新刷新自己全部目录的内容使之与新的主控保持一致。这意味着群集中的所有节点可能都需要把从自己的角度所获知的群集的目录的内容发送给主控。如果有备用的话，就不需要把这些信息在网络上重新传输一次了，并且可以加速新主控节点的选举过程。

重点是对于用户或 vSphere 管理员来说，被选举成为主控角色的 ESXi 节点并没有特别的功能或其他显而易见的不同之处。因为主控是自动选举出来的，就算它发生了故障，由于这个节点和其他节点没有功能上的区别，因此在主控节点上的进行操作还是和在其他节点上进行完全没有关系。

5.3.7 可靠数据报传输

可靠数据报传输（Reliable Datagram Transport，RDT）是 VSAN 的内部通信机制。它默

认使用 TCP 作为传输层，并根据需要创建或删除 TCP 连接（套接字）。

RDT 是构建在 VSAN 群集服务之上的，群集服务使用心跳来决定链路状态。如果发现了链路故障，RDT 就会中断这条路径上的连接并另选一条健康的路径。

当需要对一个 VSAN 对象进行操作时，DOM 使用 RDT 来与 VSAN 对象的属主对话。因为 RDT 承诺可靠交付，其使用者可以在路径或节点故障时依靠它来重新发起请求，这可能会导致对象归属的变化并变更到通往对象属主的新的路径。CMMDS（通过心跳和监控功能）和 RDT 负责处理超时（timeout）及路径故障。

5.4 磁盘格式

在研究不同 I/O 相关的数据流之前，让我们来简要地聊一聊 VSAN 的磁盘格式（On-disk Format）。

5.4.1 闪存设备

VMware 对 VSAN 的闪存设备使用其私有的磁盘格式。闪存设备的读缓冲部分具有其自己的磁盘格式，而写缓存部分则另有一种日志结构的格式。这两种格式都是新的和特别设计的，用来在闪存设备固件提供的基本功能之外增加其耐久能力。

5.4.2 磁盘

可能会令某些人惊讶的是，VMware 在 VSAN 上继续使用 VMFS，但不是传统的 VMFS。这是 VSAN 特有的一个新格式，叫做 VMFS Local（VMFS-L）。VMFS-L 是 VSAN 中每台 ESXi 主机的本地存储的磁盘文件系统的格式。标准的 VMFS 文件系统是特别为群集环境设计的，多台主机可以共享一个数据存储。它不是为单主机 / 本地磁盘环境设计的，而且肯定不是为分布式数据存储设计的。对于分布式存储来说，就要用到 VMFS-L 了。首先，群集的磁盘锁定及与其相关的 VMFS 心跳被移除了。这些功能仅在多台主机共享文件系统的时候才需要用到，而在单主机的时候是不需要的。现在新的锁管理器（Lock Manager）取代了在卷上设定 SCSI 预留（SCSI reservation）来锁定元数据的方法，完全不再需要使用 SCSI 预留了。VMFS-L 也不再需要磁盘心跳机制了，现在只需要简单地更新一下内存中的心跳副本（因为其他主机无须知道这个锁）就足够了。测试显示，通过这些变化，VMFS-L 置备磁盘所需要的时间只有标准 VMFS 的一半。

5.5 VSAN I/O 流

在接下去的几个段落中，我们将跟踪 I/O 流，看看当一台虚拟机部署到 VSAN 数据存

储上的时候，其客户操作系统中的应用程序的读写操作是怎样的。我们将观察当条带宽度设置成 2 的时候的读操作，还将观测当允许的故障数为 1 的时候写操作都又是怎样的。这将有助于理解底层的 I/O 流，并帮助你了解在设定其他功能值时的 I/O 流。在开始之前，让我们先来看看在 I/O 路径中闪存的作用。

5.5.1 SSD 的作用

如前面的章节所述，SSD（在这里此术语等同于闪存设备）对 VSAN 来说有 2 个用途：读缓冲和写缓存，这大幅提升了虚拟机的性能。在某些方面，VSAN 可以比作是市场上大量的“混合”（hybrid）存储解决方案，那些混合存储解决方案也使用了把 SSD 和磁盘组合在一起来提高 I/O 性能的方法，但是它们基于低成本的 SATA 或 SAS 磁盘驱动器，仅具有横向扩展容量的能力。

读缓冲的目的

读缓冲的目的是维护一个经常被虚拟机访问的磁盘块的列表。当缓冲命中的时候可以减少读 I/O 的延迟，缓冲命中的意思就是磁盘块位于缓冲区内，不需要从磁盘取回。虚拟机中应用程序正在读取的真正的数据块可能不在虚拟机运行的同一台主机上，在这种情况下，VSAN 会参考目录服务来找到这个数据块是否位于群集中另外一台 ESXi 主机的缓冲中。如果发生了缓冲未命中的情况，数据将会直接从磁盘取回，当然，这会引起延迟惩罚，并可能影响到 VSAN 每秒输入 / 输出操作的数量（IOPS）。我们已经讨论过目录服务，简单地说这个服务会维护一个 VSAN 群集中的对象列表，其中包括对象属主和对象位置等信息，可以用作读取时的参考。

VSAN 总是试图确保读请求是发往同一个镜像副本的，这样数据块只会在群集中被缓存一次；换句话来解释：它只存在于一块 SSD 的缓冲区内，这块 SSD 位于镜像副本所在的那台 ESXi 主机上。因为缓存空间是相对昂贵的，这个机制可以对 VSAN 所需的缓存进行优化。正确地设置 VSAN 缓冲的大小和 SSD 会对稳态下的性能产生非常显著的影响。

写缓存的目的

写缓存表现为一种回写的缓存。写在进入 SSD 上的准备阶段时被确认。事实上，我们使用 SSD 作为写缓存同样也减少了写操作的延迟。

因为写会先进入 SSD，我们必须确保这个数据块会有另外一个拷贝存放在 VSAN 群集的某个地方。VSAN 中的所有虚拟机都具有一个可用性策略设置来保证虚拟机数据至少另有一个可用的拷贝，包括写缓存的内容。一旦客户操作系统中的应用程序发起了写操作，这个动作就会同时写入到本地写缓存（本地指存储对象所在的主机）和一台或多台远程主机上的写缓存中，这个写缓存是 VMDK 存储对象的组件所在的磁盘组上关联的 SSD 或闪存设备。

这意味着当一台主机发生故障时，我们还另有一份缓存内的数据，所以不会发生数据丢失。虚拟机只需简单地重新使用这个被复制的缓存副本及其被复制的磁盘数据即可。

5.5.2 剖析 VSAN 读操作

对于 VSAN 数据存储中的对象来说，是有可能存在多个副本（RAID-1）的，这发生在虚拟机存储策略中的允许的故障数的值大于 0 的时候。换而言之，读可能跨副本发生。根据磁盘上的逻辑块地址（Logical Block Address，LBA）的不同，不同的读请求可能会被发往不同的副本。这保证了 VSAN 不会消耗不必要的读缓存。

举例来说，当虚拟机中的应用程序发起了一个读请求，就会询问群集服务（CMMDS）来判断谁是数据的属主。如果数据块是位于属主的读缓冲内，那么读请求就直接由 SSD 读缓冲提供服务。如果读缓冲未能命中，那么说明数据块不在缓冲区内，下一步就会从磁盘读取数据。

如前所述，对象的属主会把读请求分散到组成这个对象的各个组件上，这样一个给定的数据块就最多只会在 1 个节点上被缓存起来，这最大限度地有效利用了 SSD 缓存。很多情况下，群集服务返回的主机和数据属主主机不是同一台，也就是说这时数据存在另一台 ESXi 主机的 SSD 或磁盘上，可能需要通过网络来进行传输。一旦取到数据，就会反馈给请求者的 ESXi 主机，告知读取服务已经提供给了应用程序。

图 5-5 给出的就是 VSAN 上读操作的相关步骤的示意图。在这个特定的例子中，条带宽度设成了 2，虚拟机存储对象的条带分布到了不同主机的磁盘上（用 VSAN 术语来说，每个条带都是一个组件）。注意，stripe-1a 和 stripe-1b 位于同一台主机上，而 stripe-2a 和 stripe-2b 位于不同的主机上。在我们的例子中，需要从 stripe-2b 来读取数据。当引用群集服务时，数据属主不拥有虚拟机上的应用程序试图读取的数据块，因此，读请求通过万兆以太网络来取回数据块。

5.5.3 剖析 VSAN 写操作

现在我们知道了读的工作原理，让我们来看看写操作。当部署一台新的虚拟机的时候，组件是存储在多台主机上的。VSAN 并没有数据本地化的意图，因此有可能出现虚拟机在 esxi-01 上运行（从 CPU 和内存角度来看）而其虚拟机存储组件实际上是存放在 esxi-02 和 esxi-03 上的情况，如图 5-6 所示。

当虚拟机中的应用程序发起一个写操作时，对象属主克隆这个写操作。**并发**的写请求同时通过万兆网络发往 esxi-02 上的 SSD 以及 esxi-03 上的 SSD。当数据写入 SSD 后，会发送写入确认（ACK）并触发“准备操作”。属主等待所有 2 台主机的 ACK 信号后完成 I/O。稍后这个写入会作为批量处理的一部分最终回写到磁盘上。各主机的回写操作都是相互独立的，也就是说，esxi-02 的回写操作的时间可能和 esxi-03 不同。这无须协调一致因为不同主机的情况不同，例如缓存空间填满的速度、剩余空间的大小以及数据将存放在磁盘的什么地方都可能是不同的。

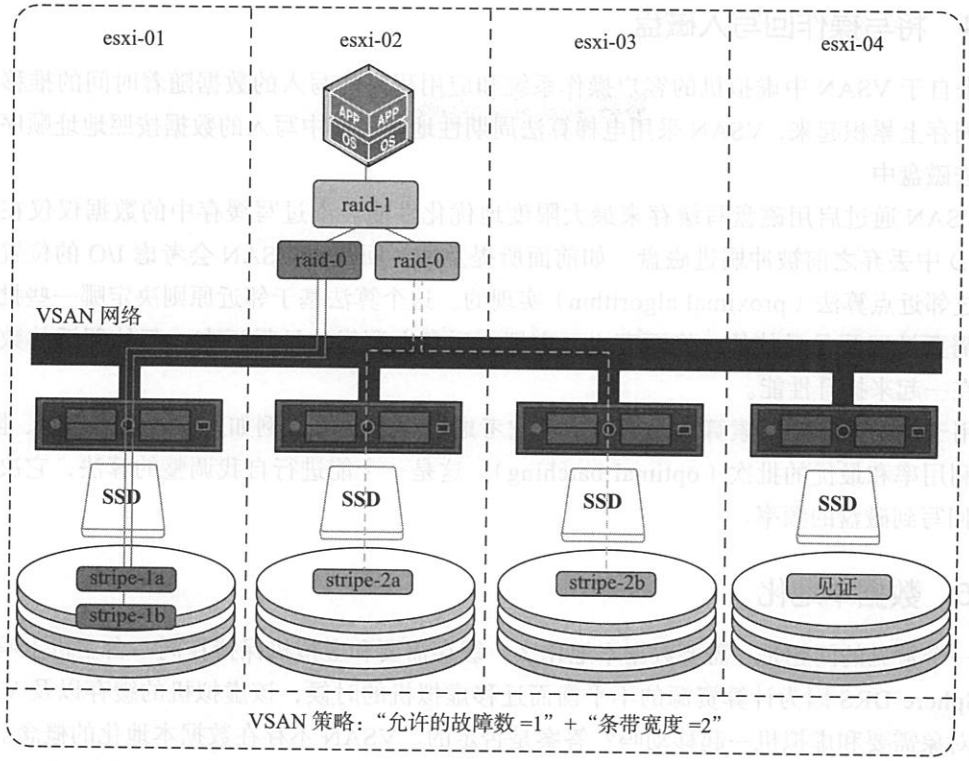


图 5-5　VSAN I/O 流：允许的故障数 =1 + 条带宽度 =2

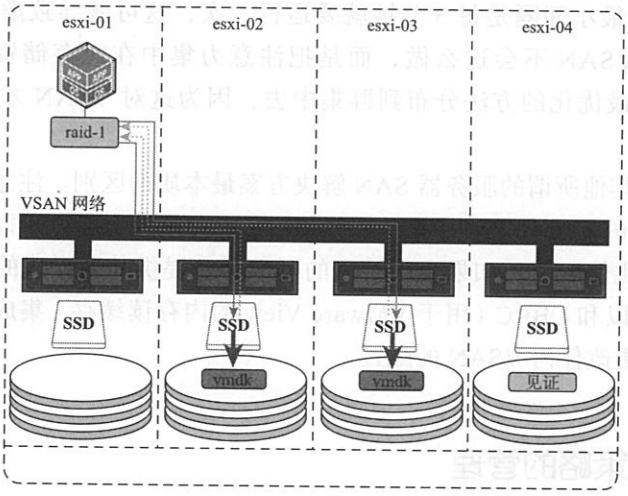


图 5-6　VSAN I/O 流：写确认

5.5.4 将写操作回写入磁盘

来自于 VSAN 中虚拟机的客户操作系统和应用程序的写入的数据随着时间的推移会慢慢在闪存上累积起来，VSAN 采用电梯算法周期性地将缓存中写入的数据按照地址顺序“冲刷”进磁盘中。

VSAN 通过启用磁盘写缓存来最大限度地优化性能。不过写缓存中的数据仅仅在数据从 SSD 中丢弃之前被冲刷进磁盘。如前面所提及的，回写时 VSAN 会考虑 I/O 的位置。这是通过邻近点算法（proximal algorithm）实现的。这个算法基于邻近原则决定哪一些批次的写入需要被回写是最佳的。换而言之，它把在磁盘上逻辑地址靠近在一起的邻近的数据块组合在一起来提升性能。

用于这个操作的探索算法是成熟的，它考虑了很多参数，例如进入 I/O 的速率、队列、磁盘利用率和最优的批次（optimal batching）。这是一个能进行自我调整的算法，它决定了 SSD 回写到磁盘的频率。

5.5.5 数据本地化

一个常见的问题是：需要数据本地化么？缓存需要和虚拟机保持在同一台主机上吗？每当 vSphere DRS 因为计算资源的不平衡而迁移虚拟机的时候，该虚拟机的缓存以及 VMDK 存储对象需要和虚拟机一起移动吗？答案是否定的。VSAN 不存在数据本地化的概念。

原因很简单：考虑到读 I/O 网络传输最多只有一跳（one hop），万兆以太网引起的延迟比起其他延迟（例如内核延迟）小到可以忽略不计，它甚至比闪存的延迟还要小，而把数据迁来迁去带来的好处远远抵不上其耗费的成本（请特别考虑一下这样的事实：默认的 vSphere DRS 最小间隔是每 5 分钟就要运行一次，这可能导致虚拟机每 5 分钟就会迁移一次）。所以 VSAN 不会这么做，而是把注意力集中在对存储资源的负载均衡上，将它们用最高效和最优化的方法分布到群集中去，因为这对 VSAN 才是更有收益和成本有效的。

这是 VSAN 和其他所谓的服务器 SAN 解决方案最本质的区别。注意我们特别指明是**读** I/O，因为出于弹性的考量，数据总是需要被存储在多台主机上的，所以无论是何种解决方案写 I/O 都会有一跳的距离。如果由于特别的要求必须提供某种形式的数据本地化，我可以告诉你 VSAN 可以和 CBRC（用于 VMware View 的内存读缓存）集成，因此你可以启用之，而完全不需要更改任何 VSAN 的配置。

5.6 基于存储策略的管理

在本书第 1 章中说过，基于存储策略的管理（SPBM）在 VMware 软件定义的存储的愿景中扮演着非常重要的角色。第 4 章覆盖了和 VSAN 结合在一起的 SPBM 的基础知识，并

讲述了管理员如何使用 SPBM 来为虚拟机定义需求组合，特别是还定义了用于虚拟机上运行着的应用程序的需求组合。这些需求组合被推送到存储层面，随后存储层面会检查这个虚拟机的存储对象是否可以被以这个需求组合的定义来实例化。例如，是否有足够的条带来满足条带宽度的要求？或者群集中是否有足够的主机来容错（满足允许的故障数的要求）？如果这些需求可以满足，那么此 VSAN 数据存储被认为是匹配资源，并会在置备向导程序中标注出来。之后，当虚拟机部署的时候，在其存储的 summary（概要）窗口就会被显示为合规的。如果 VSAN 数据存储过量配置了，或者无法满足条带的性能要求，就不会在置备向导中显示为匹配资源。如果即使有资源不匹配的情况而虚拟机仍然部署到了这个 VSAN 数据存储上，那么虚拟机的 summary（概要）窗口就会显示不合规。

概括一下，SPBM 基于放置在虚拟机存储策略中的要求，为传统环境中的虚拟机选择合适的数据存储提供了一种自动化的策略驱动的机制。在一个启用了 VSAN 的环境中，SPBM 决定了虚拟机是如何置备以及如何分布的。

让我们再进一步看看基于 VSAN 的基础架构中的 SPBM 的概念。

5.7　VSAN 的功能

本节介绍虚拟机存储策略中的 VSAN 功能。这些功能在群集配置成功时通过 VSAN 数据存储的 VASA 提供程序展现出来，特别要提的是可用性和性能策略可以针对每个**对象**分别进行设置。我们特地没有说**虚拟机**，这是因为对象甚至可以只是一块虚拟磁盘。

如果你对于这些功能进行了过量配置（也就是说，在策略中配置的能力不能被 VSAN 数据存储所满足），VSAN 数据存储就不再会在置备时显示成为一个匹配的资源，并且虚拟机会在 Summary（概要）页中显示为 noncompliant（不合规）。

在谈到 VSAN 功能的主题时，我们会再次深入地对它们进行探究。

5.7.1　策略设置：允许的故障数

第 4 章我们曾讨论过哪些虚拟机存储策略会影响虚拟机存储对象，现在让我们深入研究允许的故障数（缩写为 FTT）的细节，这可能是 VSAN 提供的功能里面最常用的。

这个功能在存储对象上设置一个条件，要求其必须能容忍至少 n 个故障（这里 n 指的是群集中同时发生的故障数，包括主机、网络或磁盘故障），还能同时保证存储对象的可用性，从而使得虚拟机可以无中断地持续运行，或（因故障类型不同）被 vSphere HA 重启。如果虚拟机存储策略中配置了这个属性，说明存储对象必须至少包含 n+1 个副本。

注意，只有当虚拟机的存储对象可用时，VSAN 数据存储上的虚拟机才能被访问。让我们来回顾一下曾经介绍过的一些概念：VSAN 数据存储上的虚拟机拥有很多存储对象，包括虚拟机主页名字空间、虚拟机交换文件、VMDK 和快照增量（snapshot delta）。为了保证虚拟机持续可访问，组成虚拟机存储对象的组件必须至少有 50% 可用。

让我们用一个简单的例子来说明。如果你部署的虚拟机的唯一的策略设置是允许的故障数等于 1，VMDK 存储对象的部署情况可能如图 5-7 所示。

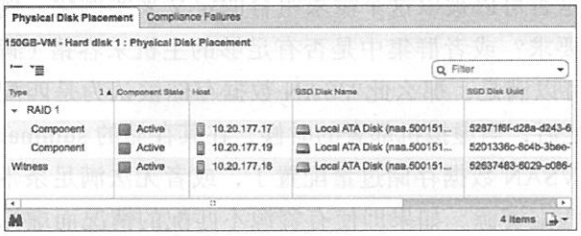


图 5-7　FTT=1 时磁盘放置的简单例子

存储对象是由组件构成的，理解这一点很重要。组成 RAID-1 镜像存储对象的两个组件，一个位于 host 17，另一个位于 host 19，正是这些镜像了的数据副本使得容错成为可能。那么，位于 host 18 上的见证对象是什么呢？嗯，还记得必须要有 50% 以上的组件可用么？ 在这个例子中，如果没有见证组件，当 host 17 发生故障时，你就失去了 1 个组件（50%）。即使此时你仍然还有一个有效的副本，已经无法满足保证 50% 以上的组件可用的条件了。这就是我们需要见证盘的原因。在裂脑（split-brain）情况下，见证还被用来判断谁仍然位于群集中。

见证对象本身实质上就是一些元数据，大小仅有 2MB，所占空间不多。但你创建的存储对象的组件越多，额外的见证对象也越多，如图 5-8 所示。这完全取决于 RAID 配置以及 VSAN 如何决定对象的放置。

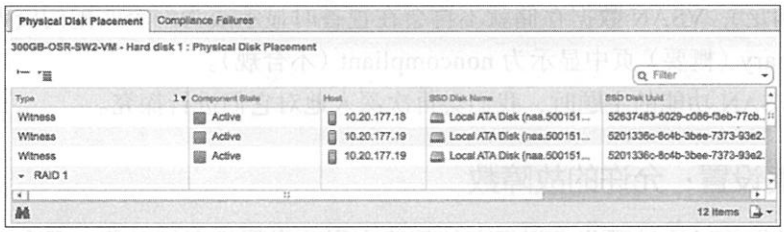


图 5-8　可能需要创建额外的见证

现在你理解了见证的概念，下一个问题是：要容忍 n 个故障，VSAN 群集需要多少台主机？表 5-1 给出了答案。

表 5-1　允许的故障数、副本数和需要的主机数之间的关系

允许的故障数 (n)	RAID-1 副本数 (n + 1)	VSAN 群集中需要的主机数 ($2n$ + 1)
1	2	3
2	3	5
3	4	7

如果试图设置的允许的故障数的值大于 VSAN 群集的能力，这是不允许的，设置不会

成功。图 5-9 的例子描述的就是试图在一个三节点的群集上将允许的故障数设置为 2。

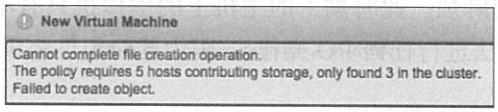


图 5-9　允许的故障数设置需要特定数量的主机支持

5.7.2　允许的故障数的最佳实践

允许的故障数的推荐值是 1，除非你有特别强的可用性要求希望虚拟机可以承受超过 1 个的故障。注意，增加允许的故障数会要求额外的磁盘可用容量，用于创建额外的副本。

VSAN 有多种管理流程来警告和提供保护，以免因意外移除主机造成 VSAN 无法满足指定的虚拟机对允许的故障数的要求。

问题来了：VSAN 群集最少需要几台主机？从支持角度说，答案是 3。然而当你需要进行系统维护并且想在维护时段中仍能保持同样的可用性水平的时候怎么办？

要满足允许的故障数为 1 的策略，任何时候都需要最少 3 台主机。即使一台主机发生故障，你仍可访问数据，因为 3 台主机具有 2 个镜像副本和一个见证，你总是可以具有大于 50% 的可用组件。但是当你如图 5-10 所示将其中的一台置于维护模式时会怎样呢？

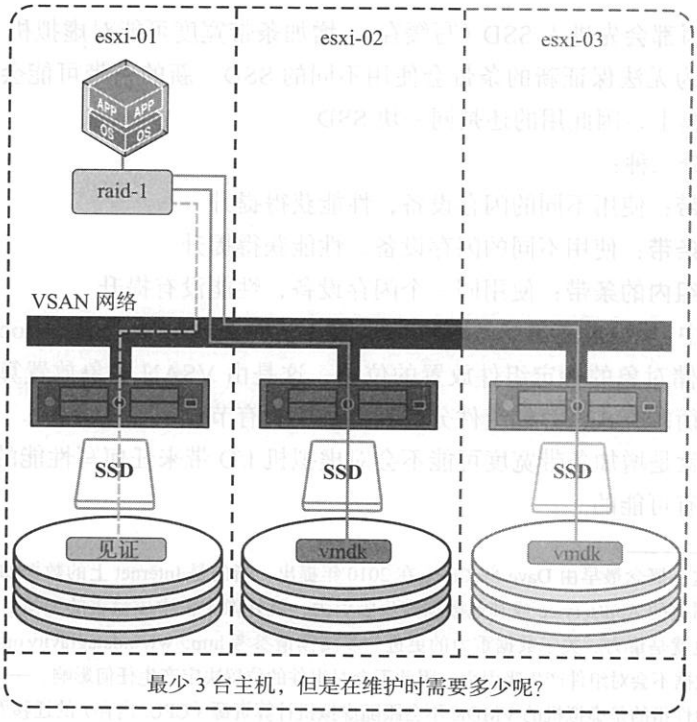


图 5-10　VSAN 所需最少主机数

当剩下的 2 台主机按照预期正常运行时，所有虚拟机会持续运行。如果此时一台主机发生了故障，你就会遇到麻烦。此时剩下的主机上的虚拟机只有不到 50% 的组件了，结果虚拟机将无法重启（也无法进行任何 I/O 操作），因为此时组件将没有属主。

5.7.3 策略设置：条带宽度

第二常用的功能非**每个对象的磁盘带数**莫属，为了行文更顺畅我们将用“条带宽度”来指代。首先要讨论的是条带化对 VSAN 环境下的虚拟机有什么好处？我们提及这个问题的原因是你需要了解 VSAN 中所有的 I/O 会先通过闪存。更准确地说，所有的写都会先进入写缓存，而所有的读也将尽可能地从读缓冲区中提供，如果数据不在读缓冲区中（读缓冲未命中）则读取将不得不由磁盘提供服务。

那么增加条带宽度有什么用呢？

- 如果虚拟机存储对象被条带化分布到不同的磁盘组，或被条带化到另一台主机的磁盘上，有可能提高虚拟机存储对象的写性能。
- 当读缓冲未能命中时。
- 数据块从闪存批量回写到磁盘上时可能会有性能提升。

让我们再详细解释一下。

性能：写

因为所有的写都会先进入 SSD（写缓存），增加条带宽度可能对虚拟机 I/O 性能有提升，也可能没有。因为无法保证新的条带会使用不同的 SSD。新的条带可能会位于同一个磁盘组的另外一块磁盘上，因此用的还是同一块 SSD。

条带的情况分三种：

- **跨主机条带**：使用不同的闪存设备，性能获得提升
- **跨磁盘组条带**：使用不同的闪存设备，性能获得提升
- **同一磁盘组内的条带**：使用同一个闪存设备，性能没有提升

在当前版本中，VSAN 不具备数据重力[⊖]（data gravity）或本地性[⊜]（locality）作为参照，因此不能指定存储对象的特定组件放置的位置，这是由 VSAN 对象放置算法决定的，它会试图通过一种平衡的方式将存储组件分散到群集中所有节点的磁盘上去。

因此，结论就是增加条带宽度可能不会对虚拟机 I/O 带来任何写性能的提升，但是潜在的性能提升还是有可能的。

⊖ 数据重力这一概念最早由 Dave McCrory 在 2010 年提出，指的是 Internet 上的数据如果在局部数量巨大的话，会对周边的 App/service 或其他数据产生吸引力，就好像宇宙中质量重的天体对其他天体产生吸引力一样——也就是重力。关于数据重力的更进一步阅读请参考 http://www.datagravity.org。这里作者的意思是 VSAN 的数据不会对组件产生吸引力，因此不会对组件的位置决定产生任何影响。——译者注

⊜ 不具备本地性指的是虚拟机的 VMDK 不会跟随虚拟机计算资源（CPU/ 内存）的迁移而迁移，即不一定在虚拟机计算资源的“本地”。——译者注

性能：读缓冲未命中

现在来看看增加条带宽度的下一条理由。这有可能是这么做的最主要的原因。当虚拟机的数据集合太大，或者在工作负载随机程度非常高的情况下，读缓冲未命中率（带来的压力）超过了一块磁盘的吞吐率，那么在读取时使用多块磁盘就可以带来好处。

从读取的角度看，当发生很多读缓冲未命中时，增加条带宽度会有所帮助。例如，一台虚拟机每秒有 2000 次读操作，它的读命中率是 90%，那么就有 200 个读操作需要由磁盘来提供。这种情况下，因为单块磁盘只能提供 150 IOPS，无法满足所有的读操作，所以此时增加条带宽度会有帮助，可以满足虚拟机对 I/O 的要求。

如何判断是否存在读缓冲未命中？不幸的是，目前还不能在 vSphere Web Client 中获取这个信息。不过 RVC VSAN Observer Tool 提供了很多详细的信息，包括读缓冲命中率（如图 5-11 所示）。在这个例子里，读缓冲命中率是 100%，意味着没必要增加条带宽度，因为所有 I/O 都是由闪存提供的。

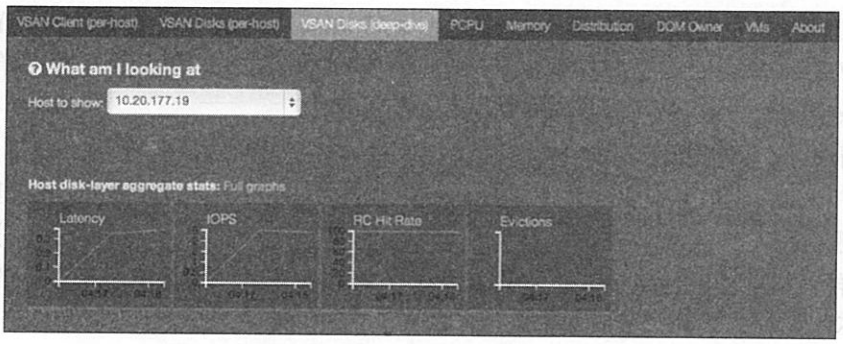


图 5-11 命中率为 100%，Evictions 为 0

所以，当 VSAN 中发生了读缓冲未命中的情况，I/O 会部分由磁盘提供，如果单块磁盘不足以处理所有请求时，增加条带宽度可以带来 I/O 性能的提升。

性能：SSD 回写

增加条带宽度的最后一个理由和把数据块从 SSD 回写到磁盘有关。关于 SSD 的磁盘回写有个重要的考量因素：VSAN 上运行的工作负载到底是何种类型的？举个例子，如果你正在部署虚拟桌面并且有几百台虚拟机，那么很可能 VSAN 的磁盘回写总是需要用到所有的磁盘，变更条带宽度就不起什么作用（因为所有磁盘已经都在用了）。如果你的虚拟机中 99 台都没什么写操作，但有一台却进行大量的写，只有在这种情况下，将这台虚拟机的条带宽度改大才可能会有性能的提升。

如何才能知道 SSD 已有大量数据块需要回写呢？你可以用 VSAN Observer Tool，它在 vCenter Server 5.5 U1 或更新的版本中提供。VSAN Observer Tool 是 RVC 的组成部分，图 5-12 所示的截图是从 VSAN Disks（deep-dive）视图的 Device-leve stats（设备级状态）选项中获取的。第 10 章中我们将详细讲述 RVC 和 VSAN Observer Tool。

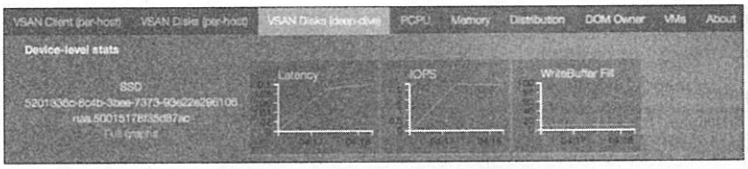


图 5-12 VSAN Observer Tool 的 SSD 信息

5.7.4 策略设置之外的 VSAN 条带化

在 Web 客户端可以看到组件的位置，如果你经常会去观察的话，或许已经发现 VSAN 在没特别指明的情况下已经把 VMDK 条带化了。又或者你明明指定了条带宽度为 2，但是却观察到创建出来的条带宽度为 3。当空间受限时，VSAN 会以自己判断觉得合适的方式来分拆 VMDK。VSAN 会在一个虚拟磁盘（VMDK）大于任何单个可用空闲空间时对磁盘使用条带化。VSAN 隐藏了这样的事实：即使主机上只有空间较小的几个物理磁盘，管理员仍然可以创建容量非常大的 VMDK。因此，即使在虚拟机的存储策略中没有设置过条带宽度，大的 VMDK 文件也会被分拆到多个磁盘上。VSAN 会在虚拟磁盘大于任一单个磁盘时使用磁盘条带。

默认情况下，即使策略并没有设置条带化的要求，一个对象副本也会在达到 256GB 时被分拆，并可以跨磁盘条带化成多个 256GB 的大块数据（条带）。条带化的分拆甚至可能发生在对象还没达到 256GB 的时候，只要因为磁盘空闲空间的原因让 VSAN 觉得分拆是有益的就会这样。注意，不是因为每到 256GB 就会产生标准分拆，就意味着所有新的大数据块会拆分到不同的磁盘上去，这取决于总体性能的平衡和空闲空间。

接下去我们来展示一些我们做过的测试，或许能把问题讲述得更加清楚一些。

测试 1

我们在一个由多个 136GB 磁盘所组成的 VSAN 数据存储上创建了一个 150GB 的虚拟机，并设定了允许的故障数（FTT）为 1 的策略。我们得到了一个 VMDK，它是一个简单的 RAID-1 配置，具有 2 个组件，每个副本都含有 1 个组件（因此没有条带化）。这是因为虚拟机在 VSAN 数据存储上默认就是精简置备的，所以即使我们创建的虚拟机是 150GB，因为精简置备的关系，它仍然可以存在于一个 136GB 的磁盘上，如图 5-13 所示。

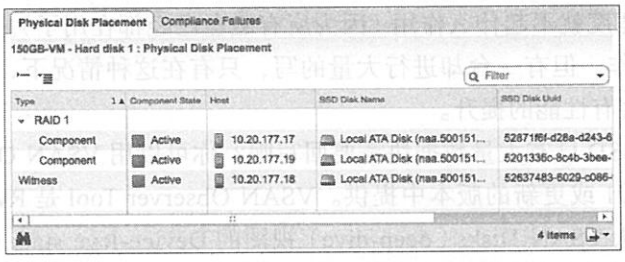


图 5-13 简单物理磁盘放置：单见证组件

测试 2

我们创建的虚拟机还是 150GB，策略中配置了 FTT 为 1 以及对象空间预留（OSR）为 100%。这次 VMDK 还是 RAID-1 配置的，但是每个副本都是由 2 个组件组成的 RAID-0。OSR 本质上就是把虚拟机配置成厚置备。因为我们要保证供给的空间，虚拟机必须至少扩展到 2 块磁盘，因此用到了条带化。

测试 3

我们创建了一个 300GB 的虚拟机，策略设置了 FTT 为 1、OSR 为 100% 以及条带宽度（SW）为 2。和前面一样，我们的配置是一个虚拟磁盘的 RAID-1 镜像，不过这次每个副本都是一个由 3 个组件组成的 RAID-0。这个配置中，即使 SW 设成 2，VMDK 要求的空间仍然太大了，甚至超过了 2 块磁盘能提供的容量，所以第 3 块磁盘必须加入进来，如图 5-14 所示。

我们可以得出结论，即使策略中没有特别指明条带宽度，条带化仍然会被用于那些比单块磁盘更大的 VMDK 上。

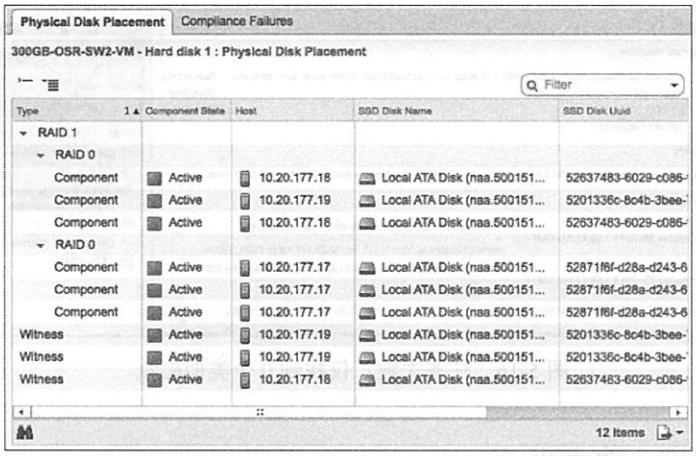


图 5-14 更复杂的部署：需要多个见证组件

5.7.5 条带宽度的最大值

在这个 VSAN 的最初发行版本中，条带宽度的最大值可以被设成 12。这可以是跨同一台主机上的多个磁盘，也可以是跨主机的。记得吗？在同时定义条带宽度（SW）和 FTT 时，至少要有 SW × FTT 个磁盘才能满足策略的要求。这意味着 SW 和 FTT 的数值越大，对象及其组件的放置就越复杂。虚拟机存储策略的**每个对象的磁盘带数**这个设置的意思是条带所跨磁盘数的最小值，如果有必要，VSAN 会增加额外的条带。

图 5-15 显示的是一个虚拟机存储策略的截图，点击 information 图标后会显示更多的细节。帮助窗口中的 HDD 代表着硬盘驱动器，也就是本书中一直提起的磁盘。

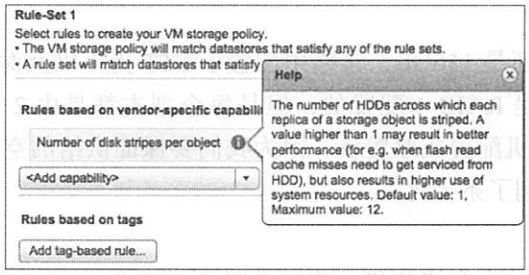


图 5-15 每个对象的磁盘条带数

5.7.6 条带宽度配置错误

你可能会问，如果 vSphere 管理员要求的 VSAN 群集实现的 SW 策略设置是不可用或不可能实现的，会发生什么情况？图 5-16 显示的就是其导致的报错信息。基本上，虚拟机的部署不会成功，并报错说没有足够数量的磁盘来满足定义的策略的要求。

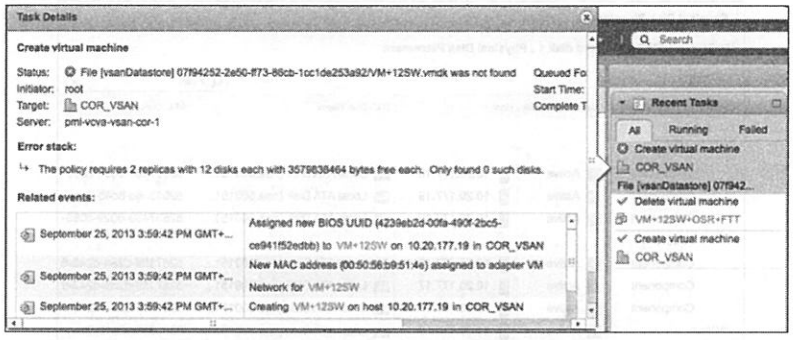


图 5-16 任务失败：仅找到 0 个类似磁盘

5.7.7 条带宽带：块大小

在探讨 SW 问题之后常被提起的一个问题是：条带的数据分段（segment）是否是特定的？换句话说，当在 VSAN 存储策略中定义条带宽度时，其组件容量增大的步进增量是多大的块？ VSAN 使用的条带分段大小是 1MB，如图 5-17 所示，1MB 的条带分段 1 会存入 esxi-02，当下一个 1MB 写入的时候，1MB 的条带分段 2 会存入 esxi-03，以此类推。

5.7.8 条带宽度最佳实践

读过本节之后，你应该对增加条带宽度可能会导致组件放置的复杂程度理解得更为透彻了。VSAN 具有很多自有的逻辑来智能地处理对象放置，我们建议不要去增加条带宽度，除非你已经明确找到了（可由增加条带宽度解决的）严重的性能问题，诸如读缓冲不能命中

或磁盘回写的性能问题。

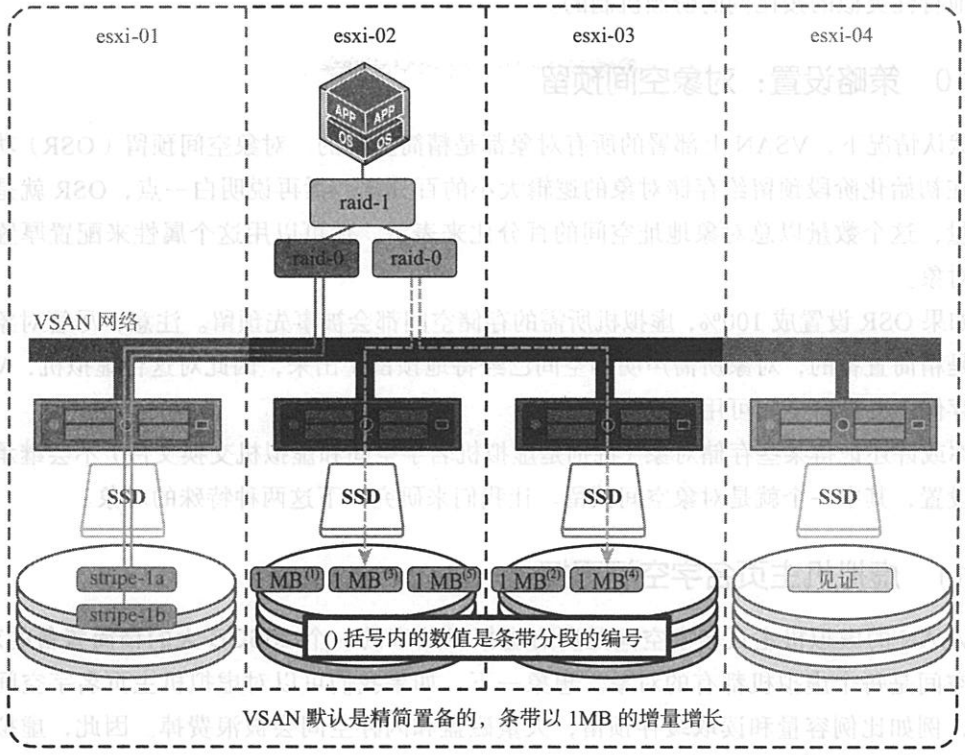


图 5-17 VSAN I/O 流：条带化以 1MB 为增量

要知道所有 I/O 都要先经过闪存层。所有写 操作先要写入闪存，然后才会批量回写到磁盘上。对于读来说，操作会先试图访问闪存层，如果读缓冲未能命中，那么数据块将从磁盘读取，因而，如果所有的读请求都能由闪存层来满足，就完全没必要增加条带宽度，因为这不会带来任何好处。因此事先做好算术并合理配置基于闪存的缓存大小，会比事后增加条带宽度更为有效！

5.7.9 策略设置：闪存读取缓存预留

这个策略设置是配置在 SSD 上为存储对象读缓存预留的闪存容量的大小，其值是存储对象逻辑大小的百分比，这个百分比数值可以配置到小数点后 4 位。如此精细的粒度是有必要的，这样管理员可以在小于 1% 的尺度上进行配置。拿 1TB 的磁盘为例，如果我们配置的读取缓存预留的增量只能以 1% 作为单位，就意味着每次增加缓存预留都至少增加 10GB，这在大多数情况下对单台虚拟机来说都太多了。

注意，要使用缓存是无须设置此预留值的。除非你正试图解决一个现实的性能问题，这个预留值应该被设为 0。

在 VSAN 的最初版本中，对于此资源（缓存资源）是没有那种 vSphere 管理员所熟悉的和其他特性类似的按比例的份额机制的。

5.7.10 策略设置：对象空间预留

默认情况下，VSAN 上部署的所有对象都是精简置备的。对象空间预留（OSR）功能定义了在初始化阶段预留给存储对象的逻辑大小的百分比。话再说明白一点，OSR 就是空间预留量，这个数量以总对象地址空间的百分比来表示。你可以用这个属性来配置厚置备的存储对象。

如果 OSR 设置成 100%，虚拟机所需的存储空间都会被事先预留。注意，尽管对象本身仍然是精简置备的，对象所需声明的空间已经特地预留了出来，因此对这台虚拟机，VSAN 数据存储不会没有空间可用。

你或许还记得某些存储对象（特别是虚拟机名字空间和虚拟机交换文件）不会继承某些策略设置，其中一个就是对象空间预留。让我们来研究一下这两种特殊的对象。

5.7.11 虚拟机主页名字空间再探

VSAN 的虚拟机主页名字空间是所有虚拟机共享的一个 **256GB 大的精简置备的对象**。名字空间是每个虚拟机都有的对象。想象一下，如果我们可以对虚拟机主页名字空间配置策略，例如比例容量和读取缓存预留，大量磁盘和闪存空间会被浪费掉。因此，虚拟机主页空间有自己特别的策略，如下所示：

每个对象的磁盘带数：1

闪存读取缓存预留：0%

强制置备：启用

对象空间预留：0%（精简）

FTT 策略设置继承自虚拟机存储策略。因此如果有客户创建了一个包含 FTT 设置的虚拟机存储策略，虚拟机主页名字空间将具有同样的能力来确保策略中指定的故障数可被容忍。

5.7.12 交换文件再探

虚拟机交换文件遵循和虚拟机名字空间一样的规则，使用同样的默认策略，也就是 1 个磁盘条带、0% 的读取缓存预留和 0% 的对象空间预留。有一点不同，这和 FTT 有关，虚拟机交换文件的 FTT 为 1，它不从虚拟机存储策略继承。

不过，虚拟机交换文件没有自己的名字空间，所以并不像虚拟机主页名字空间那样受限于 256GB 大小的精简置备的存储对象。

用于虚拟机交换文件策略的默认值也同样无法被（设定的）策略所覆盖。

5.7.13 如何查看虚拟机交换文件存储对象

我们已经知道，虚拟机交换文件是组成虚拟机的对象之一，其他对象还有虚拟机主页名字空间、VMDK 和快照增量。不幸的是，在 VSAN 中，你无法从用户界面的虚拟机对象列表中看见虚拟机交换文件，这不可避免地引发了下面的问题：如何检查和验证虚拟机交换文件对象的策略及其消耗掉的资源？

事实上，要做到这点是需要知道些小窍门的。因为即使你试图使用 RVC 命令行——vsan.vm_object_info，你也只能得到关于虚拟机主页名字空间、VMDK 和快照增量的信息（第 10 章将详细介绍 RVC 命令行），虚拟机交换文件的信息再一次遁形了。要获取虚拟机交换文件的信息，首先必须从虚拟机交换文件的描述文件（descriptor file）中获得 UUID 信息。要做到这一点，一种方法是通过 SSH 到 VSAN 群集中的某台 ESXi 主机上，并在 ESXi shell 中使用 cat 命令行来显示虚拟机交换文件的描述文件的内容，并找到 objectID 这一项。举例如下：

```
# cat win1-6e39614a.vswp
# Object DescriptorFile
version = "1"
objectID = "vsan://c7c0a552-7851-b20b-8d05-1cc1de253a92"
.
.
```

一旦找到描述符后，就可以将其用在 RVC 命令行中间，来显示真实的交换文件对象了。这条命令是 vsan.object_info，它有 2 个参数，第一个参数是群集号，第 2 个是 UUID：

```
/localhost/CH-Datacenter/computers> ls
0 CH-Cluster (cluster): cpu 86 GHz, memory 45 GB
/localhost/CH-Datacenter/computers> vsan.object_info 0 c7c0a552-7851-b20b-
8d05-1cc1de253a92
DOM Object: c7c0a552-7851-b20b-8d05-1cc1de253a92 (owner: 10.20.177.17,
policy: hostFailuresToTolerate = 1, forceProvisioning = 1,
proportionalCapacity = 100)
Witness: 048fa852-ac82-539b-a3ed-1cc1de253a92 (state: ACTIVE (5),
host: 10.20.177.19, md: naa.5000c5002bd78a5f, ssd: naa.50015178f35d87ac)
RAID_1
Component: fc8ea852-0603-7190-4bf6-1cc1de253a92 (state: ACTIVE (5),
host: 10.20.177.18, md: naa.5000c5002bd62be3, ssd: naa.50015178f35d86ee)
Component: 4d6aa852-0238-f7e6-c93c-1cc1de253a92 (state: ACTIVE (5),
host: 10.20.177.17, md: naa.5000cca00b33fc20, ssd: naa.50015178f35d8e33)
/localhost/CH-Datacenter/computers>
```

现在，我们得到了虚拟机交换文件的信息，从中可以看见以下内容：

- hostFailuresToTolerate 设成了 1，这说明虚拟机交换文件配置了 RAID-1（镜像）。
- forceProvisioning 设成了 1。这意味着即使当前策略无法满足，也必须总是置备虚拟机交换对象。

- proportionalCapacity 设成了 100%，这意味着用于交换文件的空间的确是完全预留的。

从空间利用的角度来看，我们可以推导出 VSAN 上部署的虚拟机的交换文件会消耗（已配置内存－内存预留）×（FTT+1）的容量的磁盘空间。在大多数环境中，这意味着消费的磁盘空间基本上是虚拟机已置备内存数量的两倍，因为大多数客户不设置预留值。

当然，目标是使这个信息最终能更简单地被访问到，不过现在如果你需要这个信息的话，此方法应可满足你的要求。虚拟机交换文件（.vswp）是考虑 VSAN 存储容量时的重要因素，请务必仔细考量。第 9 章中提供了一个公式来帮助计算。

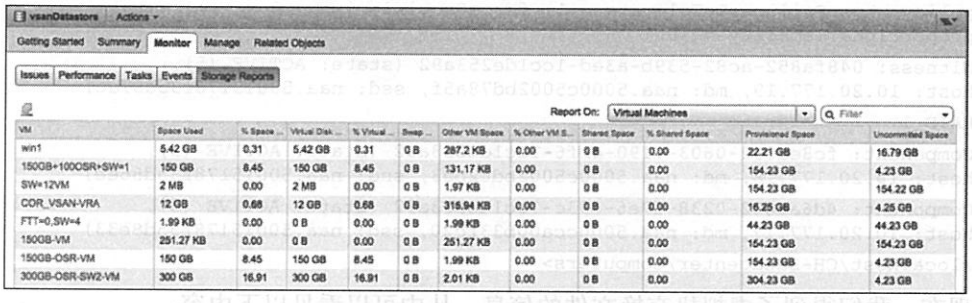
5.7.14 增量盘 / 快照的告诫

大多数时候，VMDK 增量（也常常称为快照）将继承关联在其原盘上的策略。在此 VSAN 的第一个发布版中，vSphere 管理员也可以给链接克隆[⊖]（linked clone）设置一个虚拟机存储策略。对于链接克隆来说，策略仅针对链接克隆（顶层增量盘）本身，而不是针对原盘的，但这在用户界面中是不显示的。VMware Horizon View 和 VMware vCloud Director 通过 vSphere API 使用这个功能。

现在你知道了可以预留空间并且磁盘是精简置备的，你或许会想知道哪里可以找到虚拟机消耗了多少空间和有多少空间预留等信息。

5.7.15 验证空间的实际使用量

在用户界面中选择 VSAN 数据存储，点 Monitor 页，再点击 Storage Reports，就可以看见一个漂亮的视图，显示了每台虚拟机占有空间的实际使用量，如图 5-18 所示。注意，默认视图中不会自动显示所有的列，你需要手工把它们添加进来。



VM	Space Used	% Space ...	Virtual Disk ...	% Virtual ...	Swap ...	Other VM Space	% Other VM S...	Shared Space	% Shared Space	Provisioned Space	Uncommitted Space
win1	5.42 GB	0.31	5.42 GB	0.31	0 B	287.2 KB	0.00	0 B	0.00	22.21 GB	16.79 GB
150GB+100OSR+SW=1	150 GB	8.45	150 GB	8.45	0 B	131.17 KB	0.00	0 B	0.00	154.23 GB	4.23 GB
SW=12VM	2 MB	0.00	2 MB	0.00	0 B	1.97 KB	0.00	0 B	0.00	154.23 GB	154.22 GB
COR_VSAN-VRA	12 GB	0.68	12 GB	0.68	0 B	316.94 KB	0.00	0 B	0.00	16.25 GB	4.25 GB
FTT=0,SW=4	1.99 KB	0.00	0 B	0.00	0 B	1.99 KB	0.00	0 B	0.00	44.23 GB	44.23 GB
150GB-VM	251.27 KB	0.00	0 B	0.00	0 B	251.27 KB	0.00	0 B	0.00	154.23 GB	154.23 GB
150GB-OSR-VM	150 GB	8.45	150 GB	8.45	0 B	1.99 KB	0.00	0 B	0.00	154.23 GB	4.23 GB
300GB-OSR-SW2-VM	300 GB	16.91	300 GB	16.91	0 B	2.01 KB	0.00	0 B	0.00	304.23 GB	4.23 GB

图 5-18 VSAN 数据存储到底消耗了多少空间

关于对象空间预留（OSR）这里有一些有趣的内容。如前所述，所有部署到 VSAN 上的虚拟机本质上都是精简置备的。在图 5-18 的例子中，我们部署了一台叫作 150GB-VM 的虚

⊖ 链接克隆是虚拟桌面的一种，这种类型的虚拟桌面操作系统完全一致，都来自于同一个母盘。——译者注

拟机，没有对其配置 OSR。可以看见这台虚拟机的虚拟磁盘大小是 0 字节。

第 2 个例子中，我们部署了一台叫作 150GB-OSR-VM 的虚拟机，并对其配置了 100% 的 OSR，可以看见它的虚拟磁盘的大小是 150GB。

5.7.16 策略设置：强制置备

我们曾多次提起这个功能：强制置备。如果这个参数设成了一个非零的值，那么即使虚拟机存储策略无法被数据存储满足，对象还是会被置备。然而，如果群集中没有足够的空间来满足至少一个副本对预留的要求，即使启用了强制置备，置备仍然会失败！

现在我们已经知道这些不同的功能是怎么回事了，接下去让我们来看看当故障发生时 VSAN 如何利用这些功能。

5.7.17 见证和副本：故障场景

对于 VSAN，故障场景总是一个热门话题。应该如何配置？故障时 VSAN 又会怎样反应？本节我们会用几个简单的场景来描述特定情况下 VSAN 会如何应对。

下面的例子中涉及一个 4 主机的 VSAN 群集。我们会在不同的 FTT 和 SW 设置下讨论在主机故障情况下是如何表现的。请知晓这里的例子只是为了说明问题，解释在需要进行对象放置决策时 VSAN **可能**会做出的决定。VSAN 可能会选择任何可以满足客户需求的配置（指允许的故障数和条带宽度）。例如，如果 FTT 和 SW 的值更高，放置选择中的见证的数量和主机的数量就可能会比下面所示例子中的数值更高。

示例 1：允许的故障数为 1 且条带宽度为 1

在这第一个例子中，条带宽度被设为了 1。因此，没有配置条带，只是一个简单的对象实例。然而，需求是必须能容忍一块磁盘或一台主机故障，所以必须要创建一个副本（组件的 RAID-1 镜像）。不过在这个配置中需要 1 个见证来避免裂脑情况。裂脑情况下，esxi-01 和 esxi-03 都仍然在持续运行，但是互相之间的通信中断了。哪台主机能和见证通信就是拥有有效的数据拷贝的主机。在这些配置中，数据放置可能如图 5-19 所示。

图 5-19 中，当发生了一台主机故障或者是一块磁盘故障，数据仍然可以被访问。如果 esxi-04 发生了故障，由于还是具有大于一半的组件，esxi-02 和 esxi-03 继续可以提供数据访问。然而，如果 esxi-03 和 esxi-04 都出现了故障，就无法满足简单多数的条件，数据就无法被访问了。注意，在这个场景中，从计算资源的角度来说，虚拟机是运行在 esxi-01 上的，而对象的组件则是存储在 esxi-02/03/04 上的。

示例 2：允许的故障数为 1 且条带宽度为 2

再来看一看另一个例子，这次条带宽度增加到了 2，意味着每个组件必须至少被分拆到 2 个磁盘上，不过这可能会是同一台主机上的 2 块磁盘或是不同主机上的 2 块磁盘。图 5-20 显示了存储对象的一种可能的分布方式。

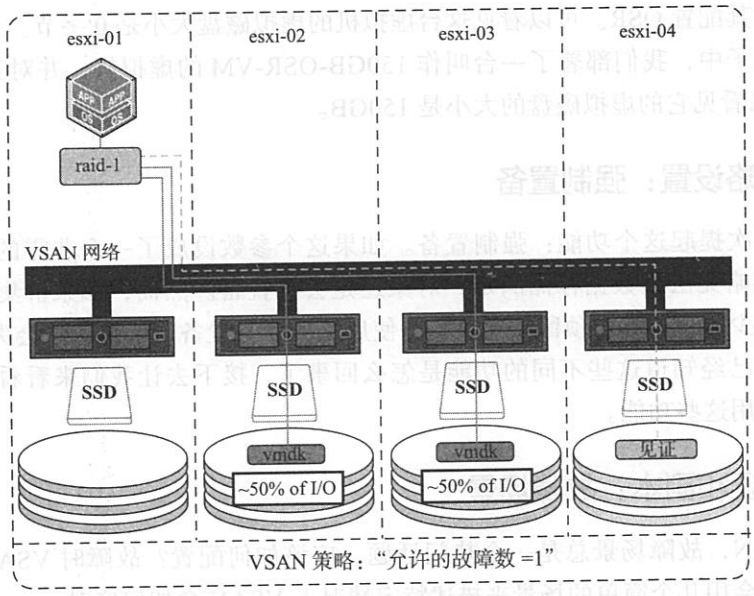


图 5-19　允许的故障数 =1

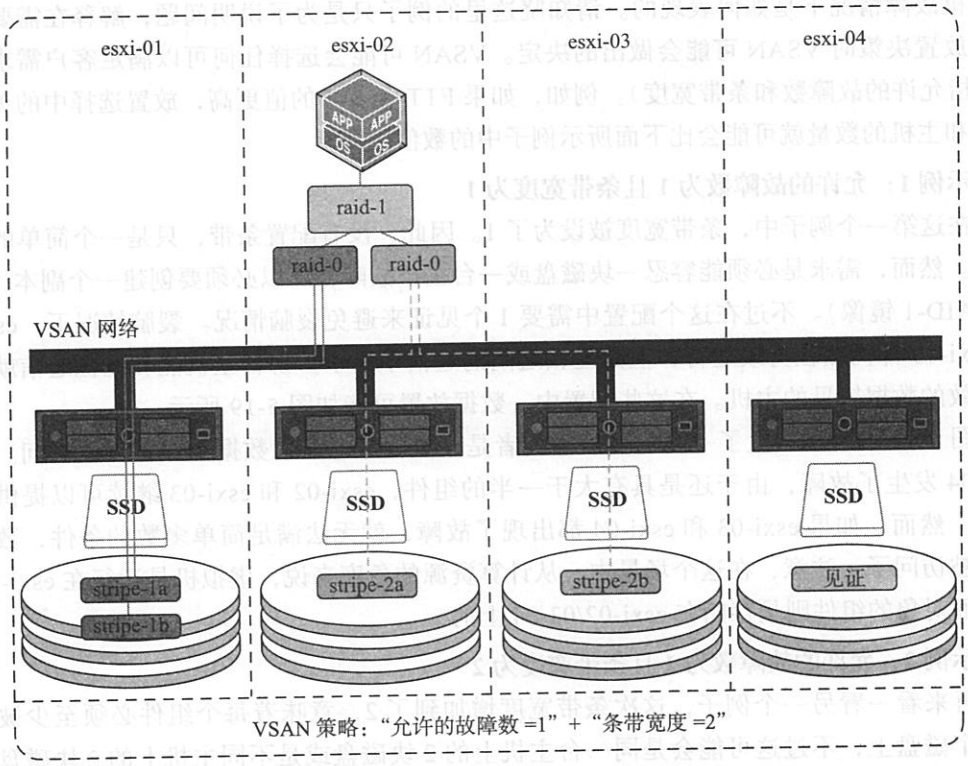


图 5-20　允许的故障数为 1 且条带宽带为 2

可以看到在这个例子中，VSAN 把第 1 个条带（RAID-0）的 2 个组件都放在了 esxi-01 上而把第 2 个条带的组件分开放置到了 esxi-02 和 esxi-03 上。因为允许的故障数为 1，我们配置了 RAID-1 镜像。在这个配置中，也用到了见证。为什么这个示例中也需要见证？想象一下，假设 esxi-01 发生了故障，这会同时影响 esxi-01 上的 2 个组件，现在我们有 2 个组件发生了故障，而另外 2 个仍然在 esxi-02 和 esxi-03 上运行得好好的。在这种情况下，我们仍然需要一个见证盘来避免裂脑的情况。

注意，如果每个 RAID-0 中各坏一个组件的话，数据将无法被访问，这是因为 RAID-1 的 2 个副本都受影响了。因此，esxi-01 上的一个磁盘故障和 esxi-02 上的一个磁盘故障会导致虚 拟机无法被访问，直到磁盘故障被修复为止。由于见证盘不包含数据，在这种情况下也无能为力。注意，此时故障数大于 1，而我们的策略设置成了只能容忍 1 个故障。

示例 3：允许的故障数为 2 且条带宽度也为 2

在最后这个例子中，允许的故障数被设成了 2，意味着还需要再多一个副本。因为每个副本都由 2 个条带化了的组件构成，所以在 VSAN 数据存储上还需要额外增加 2 个组件。现在，部署图看上去可能如图 5-21 所示。

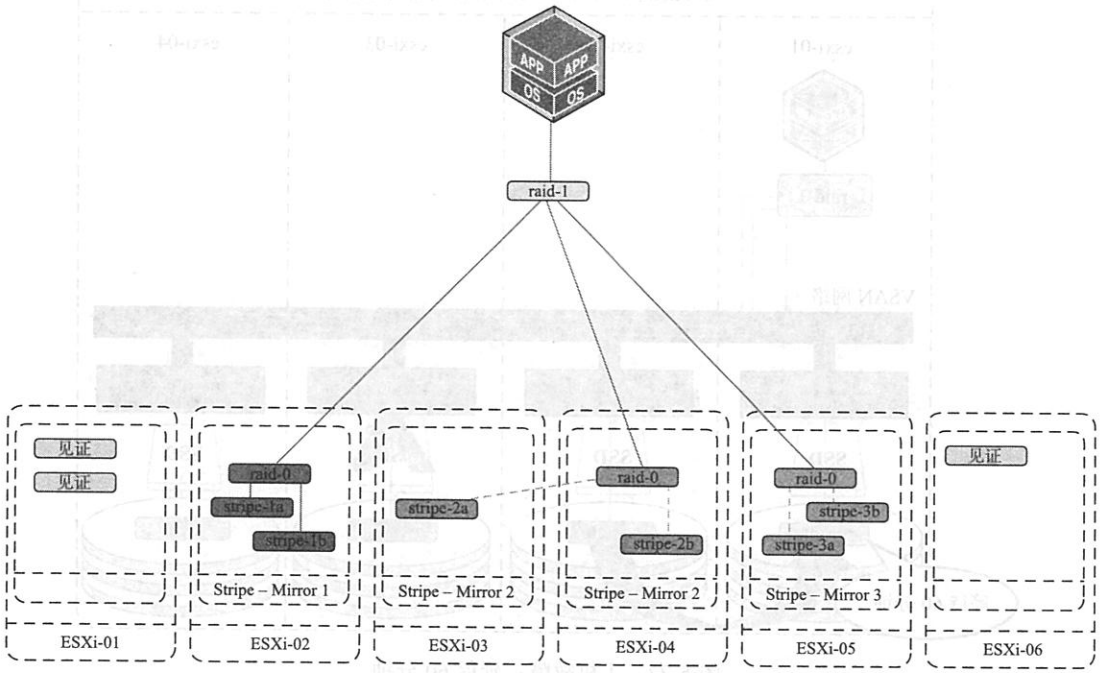


图 5-21 允许的故障数为 2 且条带宽度也为 2

条带组件前面已经解释过，大家应该很清楚了。类似地，现在出现了第 3 个 RAID-0 副本，这也可以理解。但是出现了 3 个见证盘怎么解释呢？嗯，考虑下这样一种情况：esxi-02 和 esxi-05 都坏了，这相当于 4 个组件失联，为了具有超过半数的裁决权，只剩下 2 个组

件是不够的，至少要 5 个对象才能满足超过半数的条件。这就是这个配置中需要 3 个见证的原因。这样，在 2 台主机故障的情况下才能保证数据仍然可以被访问。

如果故障发生了怎么办？ VSAN 会如何反应呢？

5.7.18 从故障中恢复

当故障被检测到时，VSAN 会判断哪个对象有组件在那台故障设备上。根据故障类型的不同，VSAN 会决定是立刻采取行动还是等一段时间（60 分钟），这取决于 VSAN 是否知道设备发生了什么情况。例如，当主机故障时，VSAN 通常不知道发生的原因是什么，甚至可能不知道到底发生了什么情况——是主机故障还是网络故障？是暂时的还是持久的，等等。如果这种情况发生了，受影响的组件会被标注为处于 “*absent*”（失联）状态。假设我们的例子中的情况是一个永久性的主机故障。

一旦 VSAN 意识到组件失联，就开始启动一个 60 分钟的计时器。如果组件在 60 分钟内恢复，VSAN 会同步副本。如果组件没能恢复，VSAN 会创建一个新副本，如图 5-22 所示。

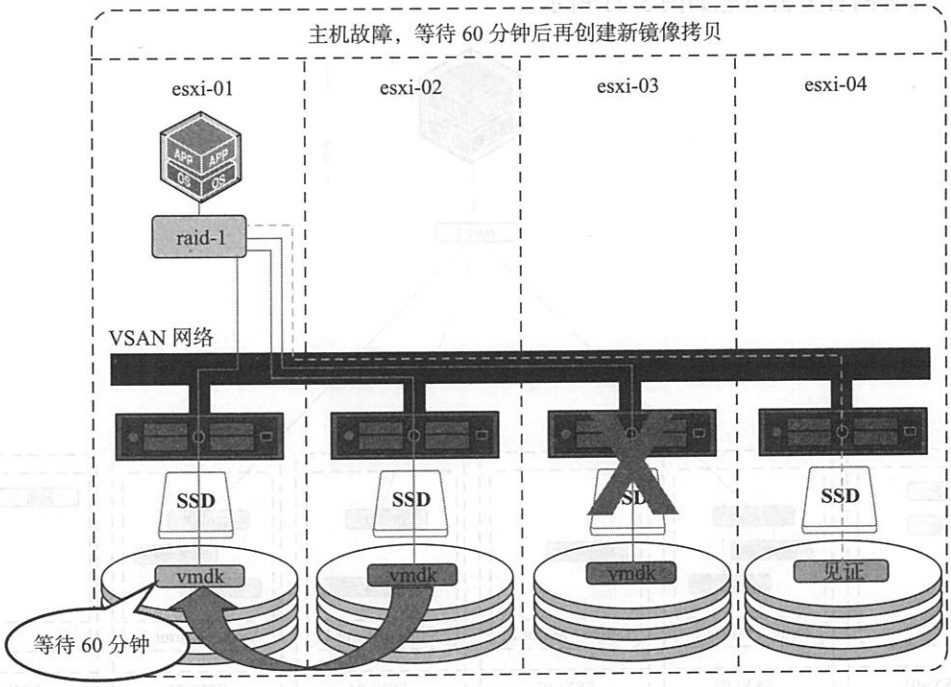


图 5-22 主机故障：延迟 60 分钟

注意，你可以在高级设置中减小这个超时（timeout）值，方法是在每台 ESXi 主机的高级设置区，更改 `VSAN.ClomRepairDelay` 参数。如果你真的想更改这个参数，我们强烈建议你在群集中所有主机上保持相同的值，这可以通过脚本方式实现。并且要定期监控这个参数实施的一致性以避免因不一致导致的问题（在更改 ESXi 高级设置前，请参考

VMware 文档或咨询 VMware 技术支持）。

前面说过，在某些场合下 VSAN 会对故障立刻做出反应。这取决于故障的类型——例如磁盘或闪存设备故障。很多情况下，控制器或设备本身能够指出发生了什么故障，并会告诉 VSAN 设备（故障）不太可能很快（在一个合理的时间内）恢复，于是 VSAN 会立刻响应，把所有受影响的组件（如图 5-23 中的 VMDK）标注为“*degraded*”（已降级），并立刻创建一个新的镜像拷贝。

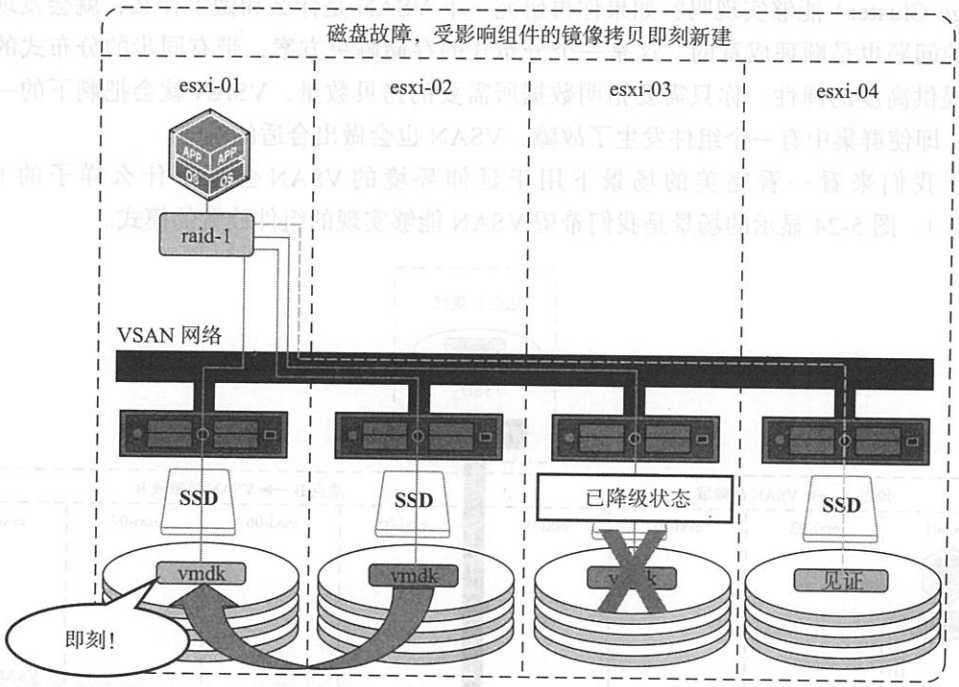


图 5-23 磁盘故障：即刻复制镜像拷贝

当然，在创建镜像之前 VSAN 会验证是否有足够的空间来存放这个新的拷贝。

如果故障在 60 分钟内就恢复了，或者在还未完成副本的创建之前就恢复了，VSAN 会决定到底是完成副本创建更好呢还是同步“旧”组件更合理。这个决定取决于一个概念——**重新配置**（reconfiguration）。

VSAN 中重新配置的发生有很多原因。首先，用户可能会选择改变对象的策略，而当前的配置或许不再能符合新的策略的规定，所有新的配置必须计算出来并应用到对象上。其次，群集中的磁盘或节点可能会发生故障。如果一个对象丢失了其配置中的一个组件，它也可能会不再能满足策略。

重新配置可能是最消耗资源的任务，因为在大多数情况下都会产生大量数据的转移。为了保证普通的虚拟机 I/O 不会被重新配置任务影响，VSAN 具有这样的能力：把重新配置任务（对资源的请求）限制在一定范围内，从而不影响虚拟机的性能。

现在你已经知道了 VSAN 对各种故障是如何反应的，知道它如何处理裂脑的情况。你或许会想知道是否可以用 VSAN 来创建一个延伸群集（stretched cluster）解决方案。让我们再进一步探讨一下。

5.7.19 延伸性 VSAN

问题再一次出现：利用 VSAN 的 vSphere 城域存储群集（vMSC，vSphere Metro Storage Cluster）能够实现吗？如果你再研究一下 VSAN 是什么和做了什么，就会发现问出这样的问题也是顺理成章的。这是一个分布式的存储解决方案，带有同步的分布式的缓存层来提供高度的弹性。你只需要指明数据所需要的拷贝数量，VSAN 就会把剩下的一切都搞定。即使群集中有一个组件发生了故障，VSAN 也会做出合适的响应。

让我们来看一看完美的场景下用于延伸环境的 VSAN 会是个什么样子的（假设 FTT=1）。图 5-24 显示的场景是我们希望 VSAN 能够实现的组件放置的模式。

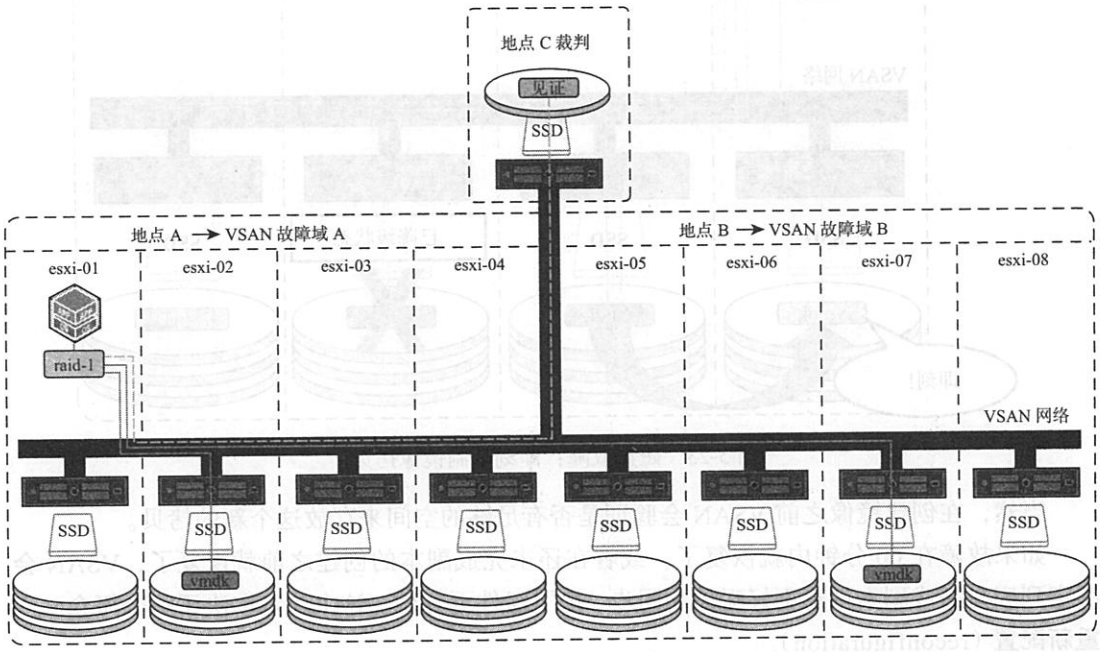


图 5-24 延伸性 VSAN：我们希望见到的情况

现在要明白的是，图 5-24 显示的是我们所希望的延伸性 VSAN 环境的情况，不幸的是，目前 VSAN **做不到**这一点。再来看一下图 5-25 中的示例，就会清楚为什么在当前的时点将 VSAN 用于延伸场景不是个好主意了。

现在已知的问题如下：

- **对象放置**：无法控制副本存放的位置。你希望第 2 个镜像拷贝存放在 B 处，但现在无法做到，因为在 VSAN v1.0 版本中无法定义“故障域”。

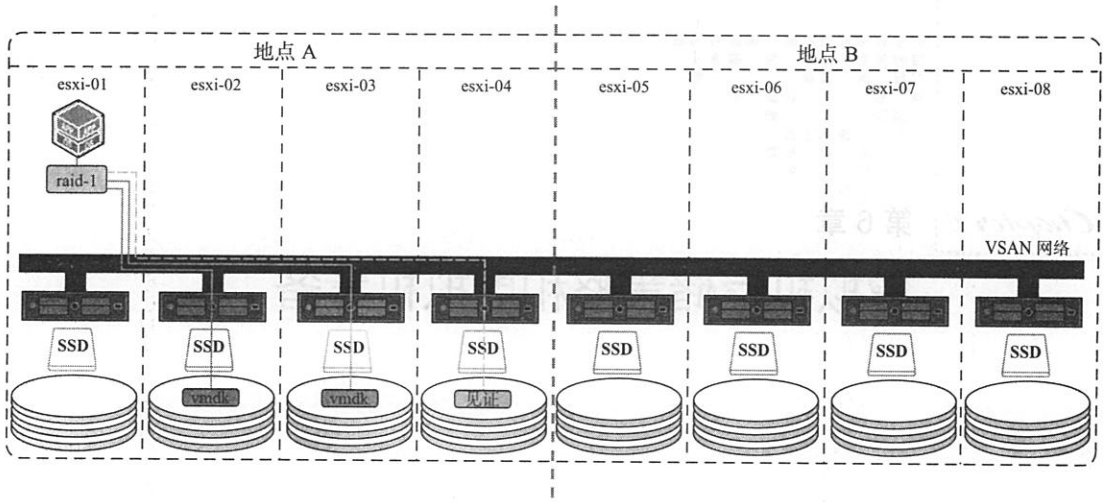


图 5-25 延伸性 VSAN：现状

- **见证放置**：你希望在发生分区隔离的情况下能有第三方场所来打破平局。但是现在见证组件可能放置在任何位置，因为 VSAN 的放置不受控制。
- **支持**：VMware 还未测试并认证过远距扩展的 VSAN，这意味着它还不受支持。

写作本书的时候，关于 VSAN 是否可以用来创建 vMSC 的问题，答案仍然是：不行，不支持远距扩展的 VSAN 群集。这当然仍然是 VMware VSAN 工程师队伍的兴趣所在，将来，我们或许会看见上述问题的解决方案。

5.8 小结

VSAN 具有一个特别的架构，既有前瞻性又同时具备可扩展性。它设计用来处理极端的 I/O 负载并可以应付不同的故障，关键就在于基于策略的管理。创建策略阶段做出的决策决定了 VSAN 数据存储及其工作负载的灵活性、性能和弹性将可以达到何种程度。

Chapter 6 第 6 章

虚拟机存储策略和虚拟机置备

本章介绍虚拟机置备工作流的一些例子。前面已经学习了 VSAN 的虚拟机存储策略的各种功能，这些功能可以用于添加一个虚拟机存储策略和在一个可用的 VSAN 数据存储上部署一台虚拟机。本章将覆盖如何使用这些功能来创建合适的虚拟机存储策略并讨论虚拟机存储对象在部署到 VSAN 数据存储之后的布局。

6.1 策略设置：FTT=1

让我们从创建一个非常简单的虚拟机存储策略开始。然后来检查将这个策略部署虚拟机到 VSAN 数据存储时会发生什么。创建的第一个策略只有一个功能设置——允许的故障数为 1。这意味着任何用这个策略部署到 VSAN 数据存储上的虚拟机都会配置一个额外的数据镜像拷贝（副本），这样如果 VSAN 群集中发生了一个故障，VSAN 存储对象仍然完全可用。让我们用实际行动来验证一下。在动手之前，让我们先画出预期的结果，如图 6-1 所示。

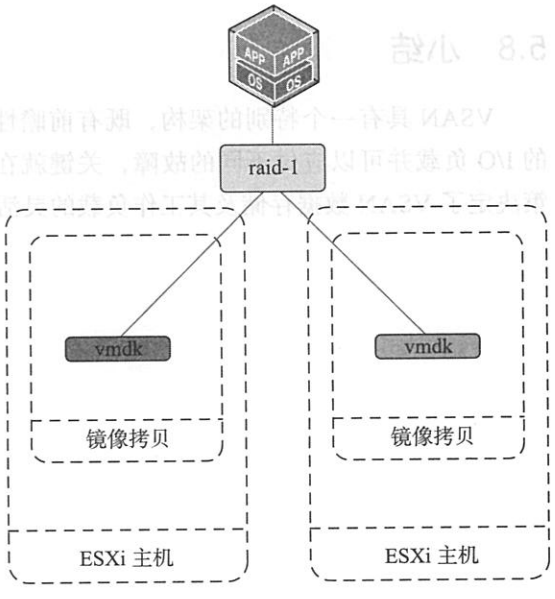


图 6-1 VSAN I/O 流：FTT 为 1

这个 VSAN 环境中具有好多台 ESXi 主机，每台 ESXi 主机都只有一个磁盘组，每

个磁盘组都只含一块 SSD 和一块磁盘。这个 VSAN 群集已经启用，并且 ESXi 主机已经组成了一个 VSAN 数据存储。我们将在这个数据存储上部署一台新的虚拟机。

让我们首先来回顾一下虚拟机存储策略创建的过程。这个创建过程在第 4 章中已经非常具体地介绍过了，大家还学到了可用在 VSAN 数据存储上部署虚拟机的各种不同的功能，或许你还记得，虚拟机存储策略中的 5 种功能是：

- ❑ 允许的故障数
- ❑ 每个对象的磁盘带数
- ❑ 闪存读取缓存预留
- ❑ 对象空间预留
- ❑ 强制置备

我们从最简单的开始，让第一个虚拟机存储策略只包含一个简单的功能——允许的故障数为 1。

首先，在 vSphere Web 客户端的 VM Storage policies（虚拟机存储策略）页面中创建一个新策略。这将会打开 Create New VM Storage Policy（创建新虚拟机存储策略）的新窗口，如图 6-2 所示。

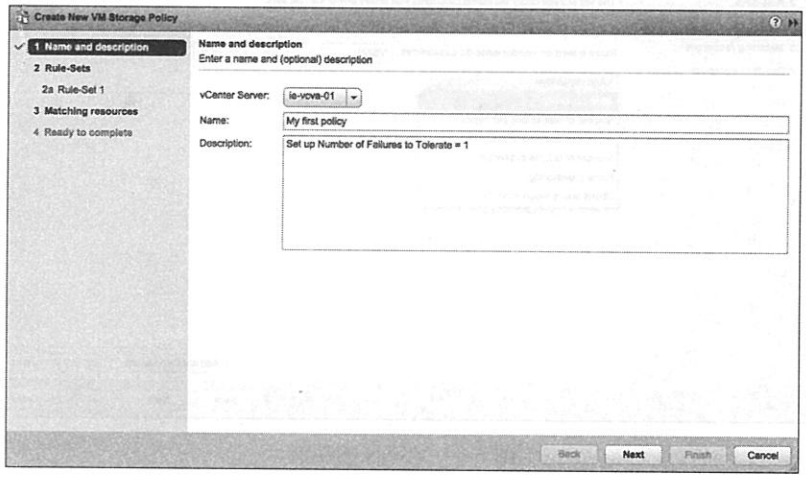


图 6-2　创建一个虚拟机存储策略

下一个屏幕会显示关于规则集（Rule-sets）的信息。规则集是把多个规则组合在一起的方法，通过这种方法，虚拟机可以根据所选的要满足的规则的不同，部署到不同的数据存储上去。出于练习的目的，我们只创建了一个规则集。这个向导程序显示了关于规则集的额外信息，如图 6-3 所示。

下一个屏幕开始给 VSAN 添加我们自己的规则集。首先要把 Vendor 从 None 改成 VSAN，在 <Add capability> 下拉菜单中就会出现新增的选项。此时点击 <Add capability>，就会显示支持的这些功能，如图 6-4 所示。

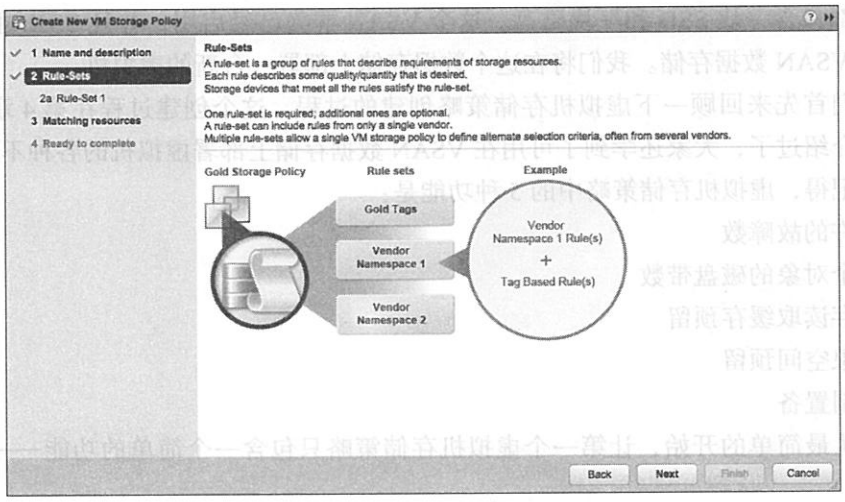


图 6-3 规则集

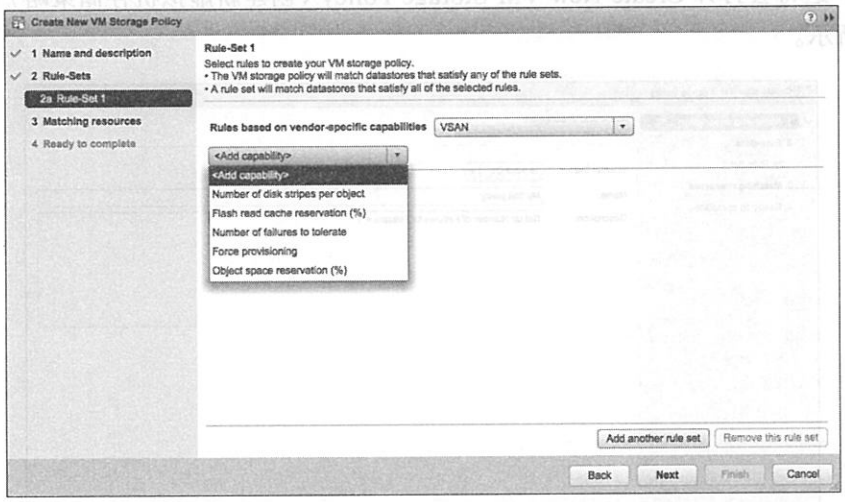


图 6-4 VSAN 的功能

对于第一个策略，我们想要添加的功能是允许的故障数，并把它设为 1，如图 6-5 所示。

在这个向导程序中，还有很多其他特性，例如 Add tag-based rules（添加基于标记的规则）和 Add another rule set（添加其他规则集）按钮。这些超出了本书的讨论范畴，不过你可以在官方的 vSphere 文档中找到额外的信息。

点击 Next 进入向导程序的 Matching resources（匹配的资源）窗口，此时 VSAN 数据存储应该会显示出来，如图 6-6 所示。这意味着 VSAN 数据存储已经理解了 VM 存储策略的内容（亦即功能）。

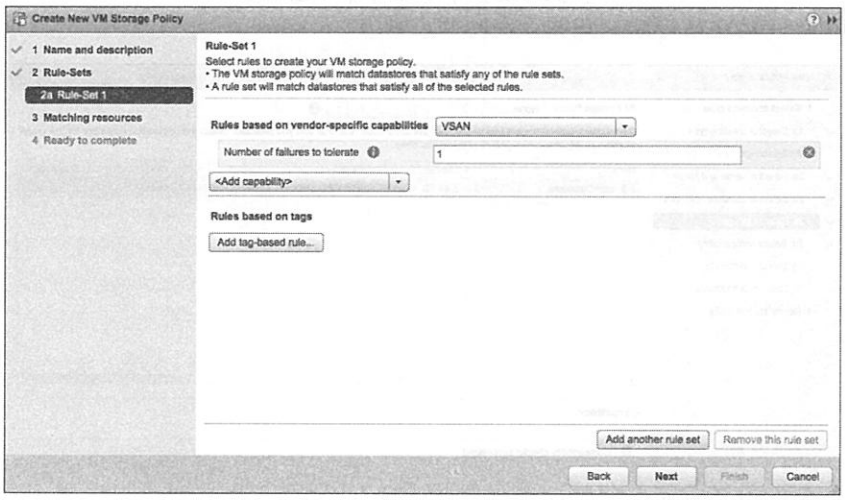


图 6-5　可允许的故障数设为 1

小心　注意，VSAN 数据存储在 Matching Resources（匹配的资源）窗口显示出来并不意味着 VSAN 数据存储可以用来置备虚拟机。策略可能会包含一个不切实际的条带宽度或 FTT 设定，VSAN 群集无法满足。这个屏幕仅仅意味着 VSAN 理解了策略的内容，这是一个非常重要的区别。

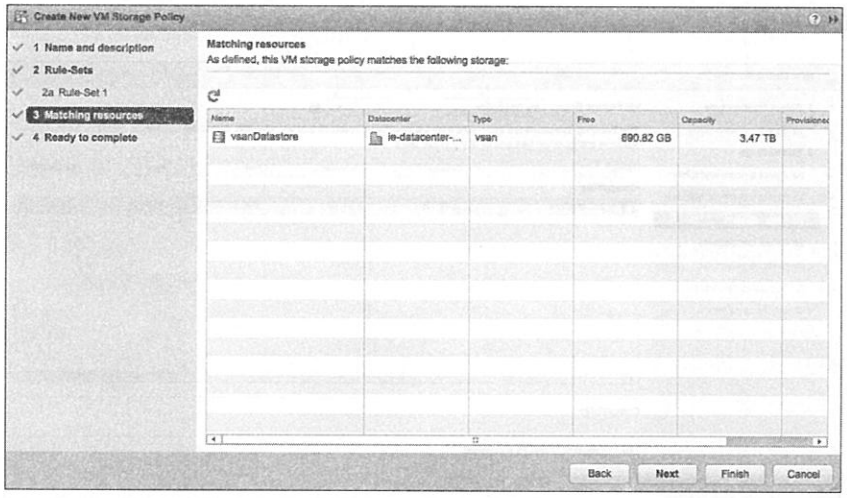


图 6-6　匹配的资源

再次检查策略并点击 Finish（完成）来创建策略。恭喜！你已经创建了第一个虚拟机存储策略。现在可以再进一步用这个策略去部署一台新的虚拟机。部署一台新的虚拟机和以前的方法一模一样。唯一的区别在于存储选择的步骤，默认情况下，是不会选择任何虚拟

机存储策略的。默认的设置是 None，如图 6-7 所示。

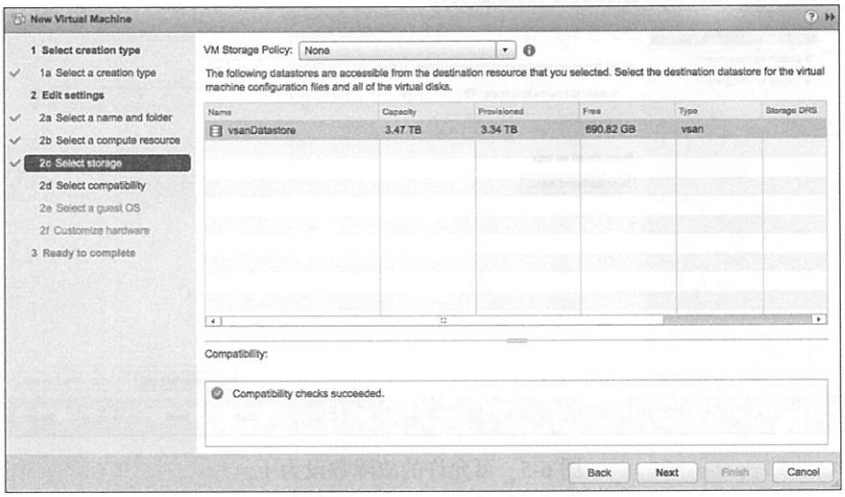


图 6-7　没有选择任何策略

不过，当选择了新的虚拟机存储策略（My first policy）后，就可以看见 VSAN 数据存储是兼容的，如图 6-8 所示。就像在创建一个新的虚拟机存储策略向导程序（Greate New VM Storage Policy wizard）中匹配的资源那部分一样，这仅仅意味着 VSAN 数据存储理解了策略的内容，并不意味着 VSAN 群集可以满足要求，只有在虚拟机真正部署的时候才能知道是不是满足要求。

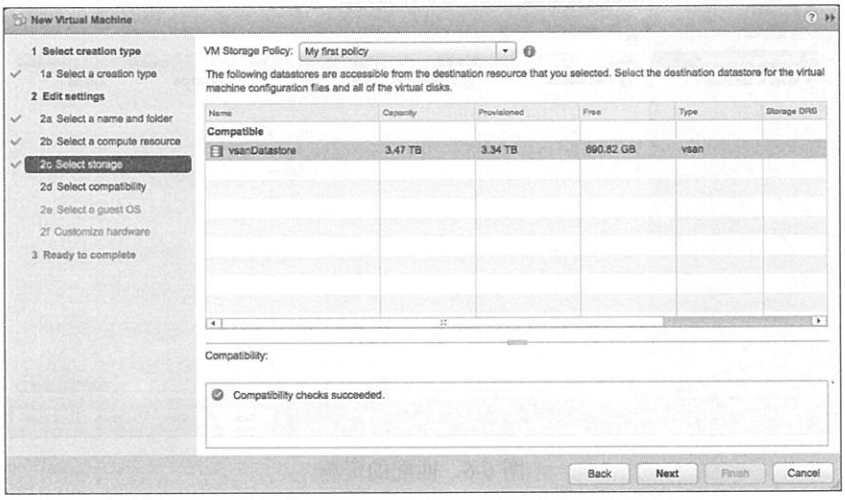


图 6-8　选中 My first policy，并且 VSAN 数据存储是兼容的

当虚拟机部署完成后，点击 Manage（管理）导航到虚拟机视图，并选择 VM Storage Policies（虚拟机存储策略），如图 6-9 所示。从这里我们可以看见虚拟机存储对象的布局，

例如虚拟机主页名字空间和虚拟机磁盘文件（VMDK）。虚拟机主页名字空间是存储 .vmx 虚拟机描述文件和其他虚拟机所必需的配置文件的地方。这些组成 VSAN 数据存储上的虚拟机的存储对象在第 5 章中已经具体讨论过了。

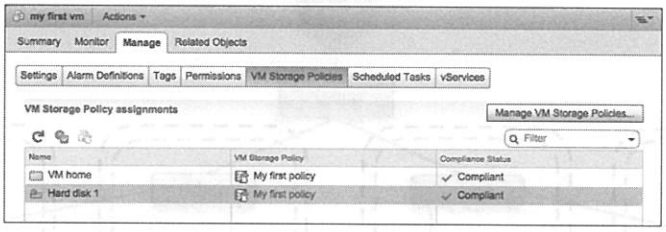


图 6-9 合规性状态是 Compliant（合规的）

可以看见，2 个对象都是合规的。换而言之，它们都满足虚拟机存储策略中定义的功能要求。这意味着虚拟机可以容忍 VSAN 群集中的一个故障并仍然具有一个全功能的可用的存储对象。此时选择 Physical Disk Placement（物理磁盘放置位置）选项卡可以进一步对这 2 个对象（虚拟机主页和硬盘）进行观察，在这里可以看见组件的 RAID-1（镜像）配置，如图 6-10 所示。

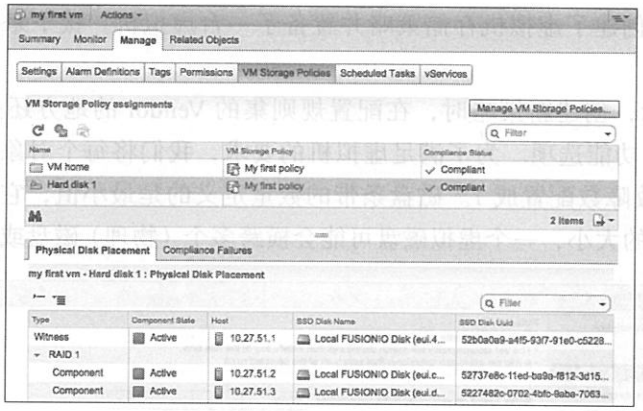


图 6-10 物理磁盘放置位置

6.2 策略设置：FTT=1, SW=2

接下来我们试试另一个策略设置，添加另外一个功能。这次，我们将使用一个比第一个例子具有更多资源的群集来满足额外的要求。这次我们将要求将可允许的故障数设为 1 并将每个对象的磁盘带数设为 2。我们先来创建这样一个虚拟机存储策略并用这个策略来部署一台虚拟机，然后看看对不同的虚拟机存储对象的布局会产生何种影响。在这个例子中，我们会设定一个将 RAID-0 条带配置做成了镜像的 RAID-1 配置，这就是 4 个磁盘组件，其中每个 RAID-0 条带含有 2 个组件，然后再两两镜像组成 RAID-1 的配置。图 6-11 显示的

是一个逻辑视图。注意，因为每台 ESXi 主机只有一块物理磁盘，所以每个 RAID-0 配置将会横跨至少 2 台主机。

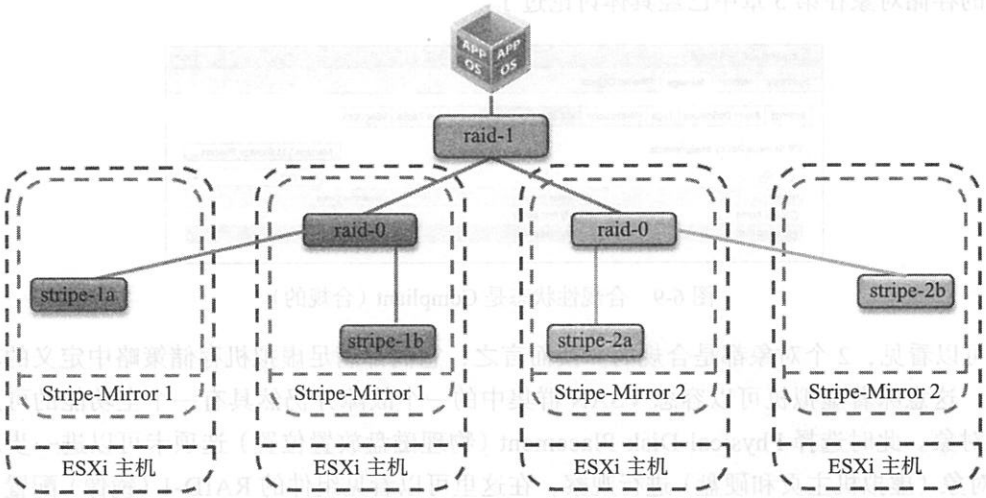


图 6-11　VSAN I/O 流：跨 2 台主机的条带化

现在我们已经创建了虚拟机存储策略并置备了一台虚拟机。接下来看看我们的理论到底对不对。

如图 6-12 所示，创建新策略时，在配置规则集的 Vendor 的地方还是要选择 VSAN 以显示所需的 VSAN 功能选项。为了满足虚拟机的要求，我们将每个对象的磁盘带数配置成 2，并将可允许的故障数配置成 1。磁盘条带的数量定义的是最小值，它还取决于虚拟磁盘的大小和物理磁盘的大小，一个虚拟磁盘可能会横跨多个（物理）磁盘或主机。

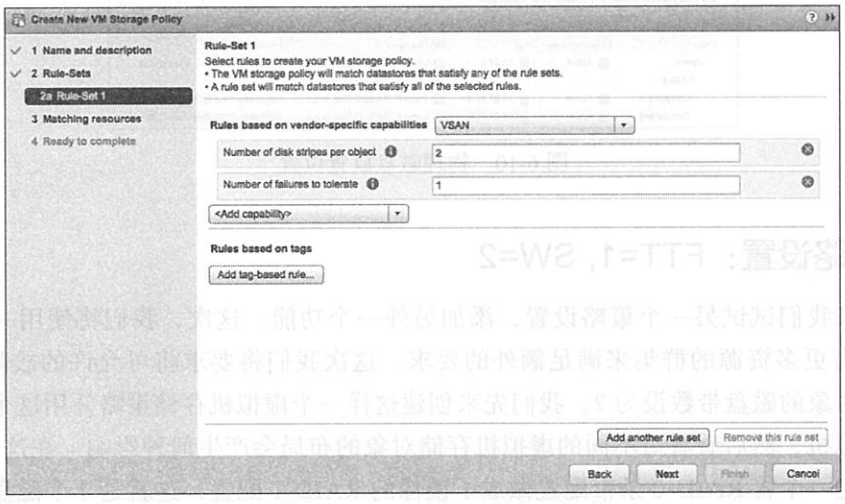


图 6-12　虚拟机存储策略：FTT= 1 且 SW = 2

现在已经创建了一个新的虚拟机存储策略，接下来看看虚拟机置备工作流，如图 6-13 所示。

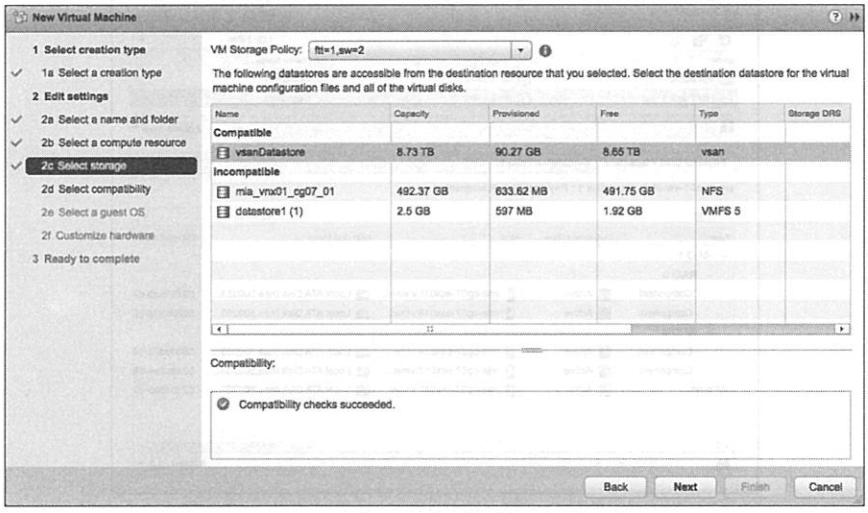


图 6-13　VSAN 数据存储和策略 ftt=1, sw=2 兼容

在这个例子中，我们特地把新建的策略起名为 ftt=1, sw=2，现在可以看见可用的数据存储就分成了 2 个不同的类别：

- Compatible（兼容）
- Incompatible（不兼容）

如你所见，选择了新建的虚拟机存储策略后，只有 VSAN 数据存储是兼容的。这是因为只有 VSAN 数据存储才能理解置于虚拟机存储策略中的这些功能要求，而其他数据存储（本地 VMFS 和 NFS）都无法理解这些策略要求，因此被放在了不兼容这个分类中。不过如果你仍然想放的话，这些不兼容的数据存储也是可以选择的。如果选择了一个不兼容的数据存储，就会收到警告，告知数据存储与给定的虚拟机存储策略不匹配，并且这个策略会显示为不适用。

部署完虚拟机之后，我们再来检查一下物理磁盘的布局情况，如图 6-14 所示。

从图 6-14 中可以看出，为了满足虚拟机存储策略中关于 FTT 的要求，已经创建了 RAID-1（镜像）配置。现在你还可以看见一些额外的信息——每个副本都是由一个 RAID-0 条带配置组成的，每个条带包含 2 个组件，这满足了 SW=2 的要求。

这里还创建了一个见证盘。现在要指明的是，见证盘的数量与组件在群集中的主机和磁盘上分布的情况直接相关。如果这是一个 3 节点的群集，可能会需要创建几个额外的见证盘来确保故障时（尤其是主机故障时）虚拟机对象的组件仍然有 50% 可用。这个例子中我们用到的 VSAN 群集有 8 个节点，因为组件被分散到了每一台单独的 ESXi 主机上，因此单个见证盘就足够保证在一台主机故障的时候保持可用组件数大于 50%。

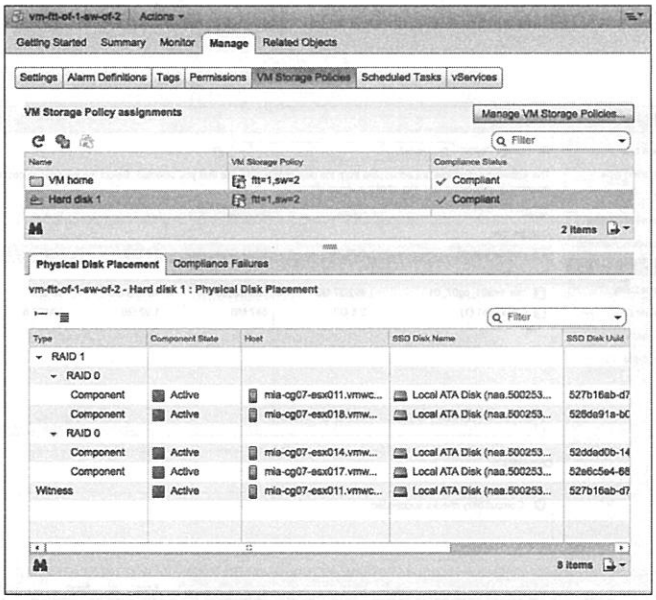


图 6-14　物理磁盘放置位置，策略为 ftt=1, sw=2

有趣的是，VM Home（虚拟机主页）名字空间不会去匹配每个对象的磁盘带数的要求，它只会满足允许的故障数的要求。因此，如果去查看虚拟机主页名字空间的话，会看见组件没有 RAID-0 配置，如图 6-15 所示。

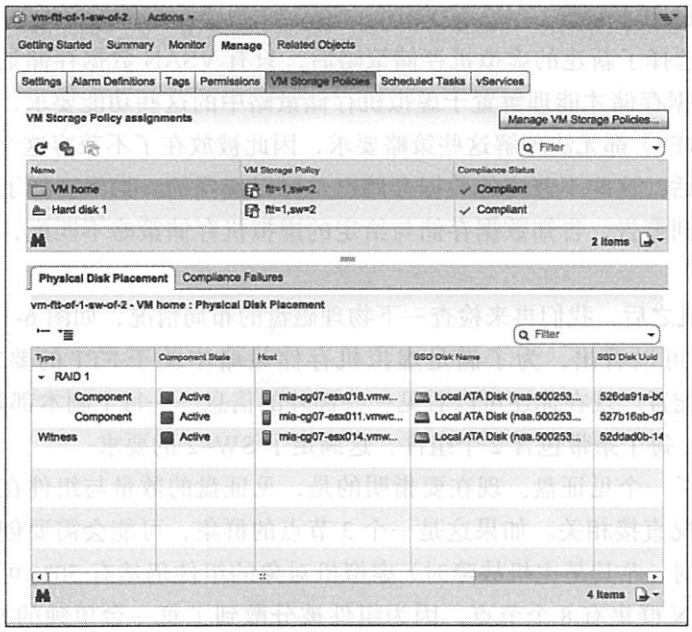


图 6-15　虚拟机主页名字空间没有条带宽度的配置

6.3　策略设置：FTT=2，SW=2

在下面这个例子中，我们创建了另一个虚拟机存储策略，将每个对象的磁盘带数设成 2，并把允许的故障数设成 2，这意味着 VSAN 群集必须能容忍 2 个不同的故障（主机、网络或磁盘故障）。考虑到设定要容忍的“双主机故障”和 2 个磁盘条带，可能的磁盘布局如图 6-16 所示。

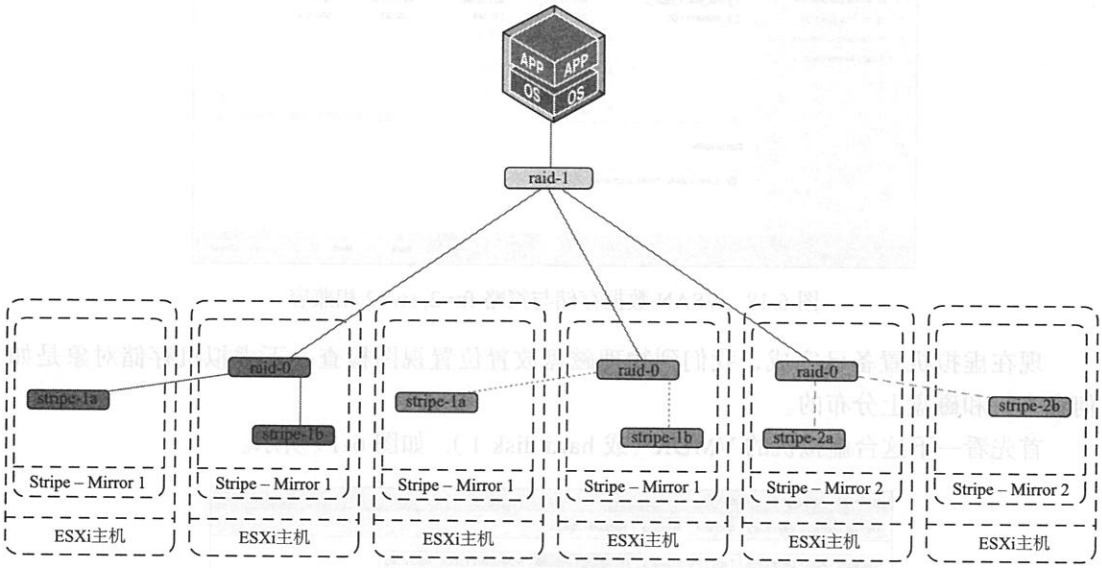


图 6-16　VSAN I/O 流：允许 2 个故障且把条带宽带设为 2

首先，根据要求创建策略，如图 6-17 所示。

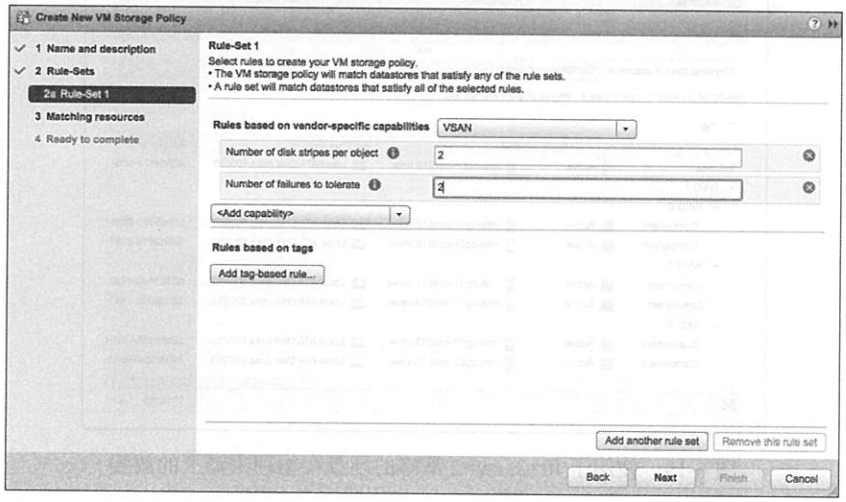


图 6-17　设置为 FTT =2, SW =2

接下来用这个策略部署一台新虚拟机，如同预想的一样，只有 VSAN 数据存储兼容这个 ftt=2，sw=2 的虚拟机存储策略，如图 6-18 所示。

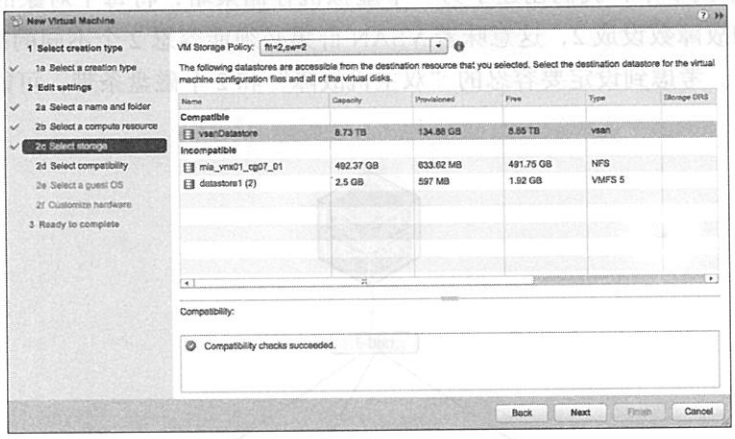


图 6-18　VSAN 数据存储与策略 ftt=2, sw=2 相兼容

现在虚拟机置备已完成，我们到物理磁盘放置位置视图检查一下虚拟机存储对象是如何在主机和磁盘上分布的。

首先看一下这台虚拟机的 VMDK（或 hard disk 1），如图 6-19 所示。

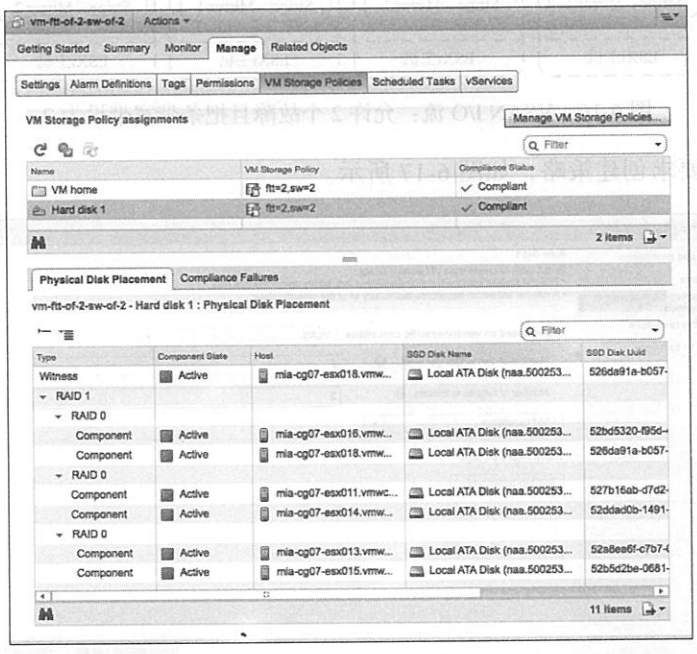


图 6-19　使用了 ftt=2, sw=2 策略的硬盘在物理磁盘上的放置

现在我们可以看见 VSAN 对这台虚拟机的虚拟磁盘实施了额外的 RAID-0 条带配置。

对 RAID-0 条带配置来说，至少要有一个 RAID-0 条带配置中的所有组件都必须完好无损。这就是需要第 3 个 RAID-0 条带配置的原因了。你或许会想，如果第一个 RAID-0 条带配置中的第一个组件丢失，同时第 2 个 RAID-0 条带配置中的第 2 个组件丢失，或许此时 VSAN 还能利用剩下的组件来保持存储对象可用。事实并非如此。要在群集中容忍 2 个同时故障，第 3 个 RAID-0 条带配置是必需的，因为 2 个故障可能会导致 2 个 RAID-0 条带配置同时无法使用。这也是所有的 RAID-0 条带配置都要在一个 RAID-1 配置中保持镜像的原因。如图 6-19 中可看见的那样，组件保存在这个 8 节点 VSAN 群集中 6 台不同的 ESXi 主机上，它们分别是：mia-cg07-esx11、mia-cg07-esx13、mia-cg07-esx14、mia-cg07-esx15、mia-cg07-esx16 以及 mia-cg07-esx018。

接下来看一看虚拟机主页名字空间，如图 6-20 所示。

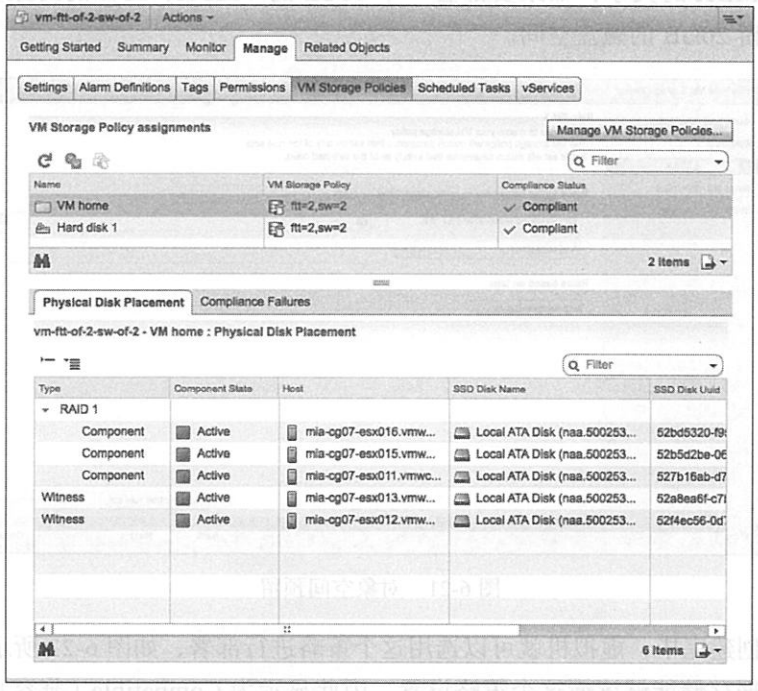


图 6-20　当策略是 ftt=2, sw=2 时，虚拟机主页在物理磁盘上的放置

前面已经说过，虚拟机主页名字空间不实施磁盘条带策略，但是的确会实施允许的故障数策略。所以布局视图中没有 RADI-0 配置，但我们仍可看见 RAID-1 镜像配置，其中通过 3 个副本来满足虚拟机存储策略中 FTT 设成 2 的要求。此外我们还可以在这里观察到见证盘数量的增加。记住，要使这个对象保持在线状态，虚拟机主页名字空间对象必须有 50% 以上的组件保持在线，从而在即使丢失 2 个副本的情况下，仍然有一个副本（也就是虚拟机主页名字空间的一个拷贝）可用。因此，即使 2 个故障同时发生让我们损失了 2 个配置副本，我们仍然有超过 50% 的组件可用。

6.4 策略设置：FTT=1,OSR=50%

下一个例子中我们将要探索一个不同的功能。如前所述，VSAN 上部署的所有对象默认都是精简置备的。这意味着它们一开始是不占用磁盘空间的，但是会随着虚拟机内客户操作系统的运行慢慢地按需增长。不过，使用虚拟机存储策略中的对象空间预留这个策略设置，可以在虚拟机部署的时候预先为其保留一定百分比的磁盘空间。默认情况下，对象空间预留的值是 0%，这就是 VSAN 数据存储上的虚拟机都是精简置备的原因。如果想要为一台虚拟机预留所有的空间（也就是“厚置备”的磁盘），你可以用这个参数来实现——把对象空间预留值设置成 100%。而这个例子中我们会设成一个中间值。

我们从在虚拟机部署时预留 50% 磁盘空间的例子开始，如图 6-21 所示。这个百分比的数值指的是 VMDK 的大小，如果要部署一个 40GB 的 VMDK，50% 的对象空间预留值就意味着应该预留 20GB 的磁盘空间。

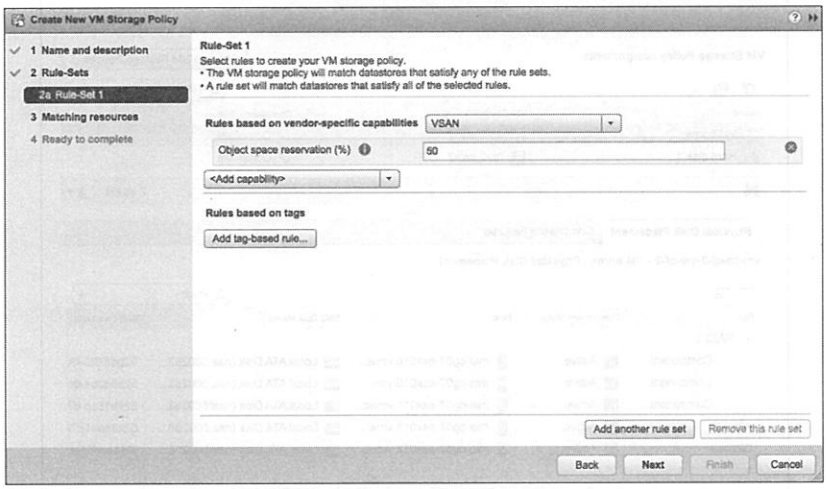


图 6-21 对象空间预留

一旦策略创建完毕，虚拟机就可以选用这个策略进行部署，如图 6-22 所示。

VSAN 数据存储能够读懂这个策略设置，因此显示为 Compatible（兼容），而其他数据存储则被标注为 Incompatible（不兼容）。善于观察的你或许已经发现了我们没有选择可允许的故障数这个条件。当然，我们应该设置这个参数来保证虚拟机的高可用性。不过，就算没有专门指明，可允许的故障数为 1 这个设置总是隐性存在的，因此，如果策略中没有特别设定 FTT 的值，就会按照 FTT=1 来实施。可以通过检查物理磁盘放置位置（Physical Disk Placement）视图，查看到底有没有 RAID 配置来确认这一点。只有当 FTT 明确地在策略中设置为 0 时，才不具备 RAID-1 的配置。

首先，我们在 VM home（虚拟机主页）名字空间视图中验证了的确存在 RAID-1 配置，如图 6-23 所示。

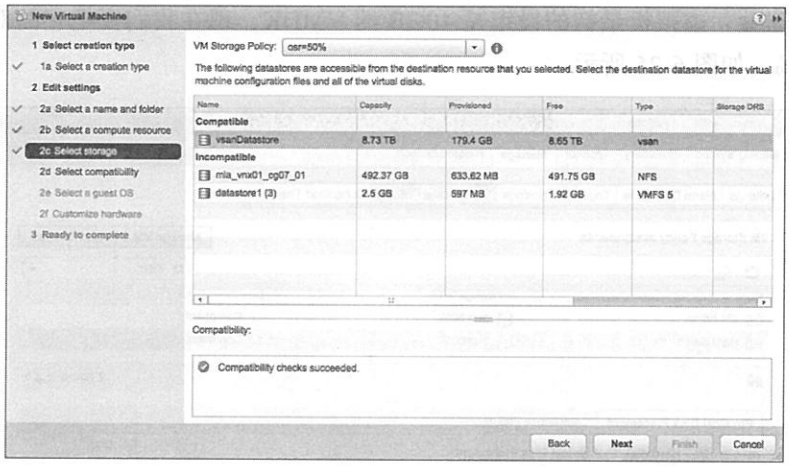


图 6-22　VSAN 数据存储理解对象空间预留的要求

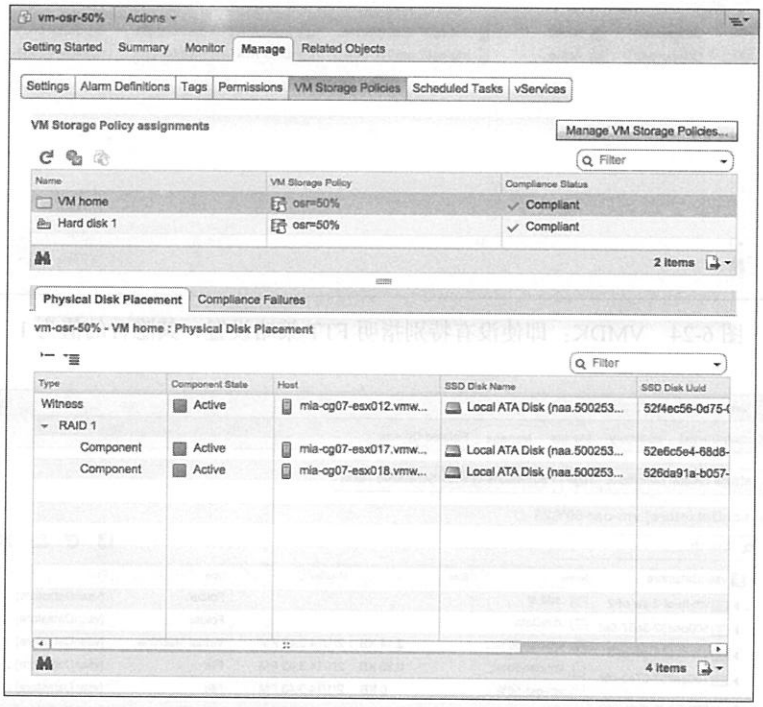


图 6-23　虚拟机主页：即使没有在策略中指定也可以推算出允许的故障数

我们还可以确认硬盘也具有镜像配置，即使没有专门在策略中进行设定，它还是具有可以容纳 1 个故障的能力，如图 6-24 所示。

然而，让我们回到最初提起的额外要求上——它要求保留虚拟机所需磁盘空间的 50%。要查看 VMDK 到底消耗了多少空间，请使用 vSphere Web 客户端并导航到 Datastore >

Manager > Files，虚拟机最初的设定是 40GB 的 VMDK，现在我们要把它的对象空间预留值设置为 50%，如图 6-25 所示。

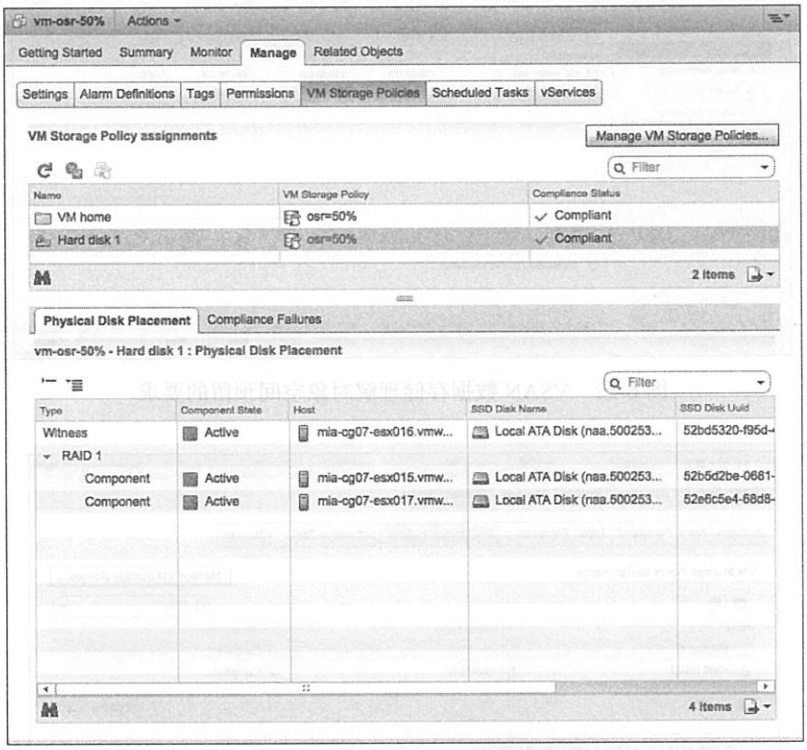


图 6-24　VMDK：即使没有特别指明 FTT 策略设置，其隐含的值为 1

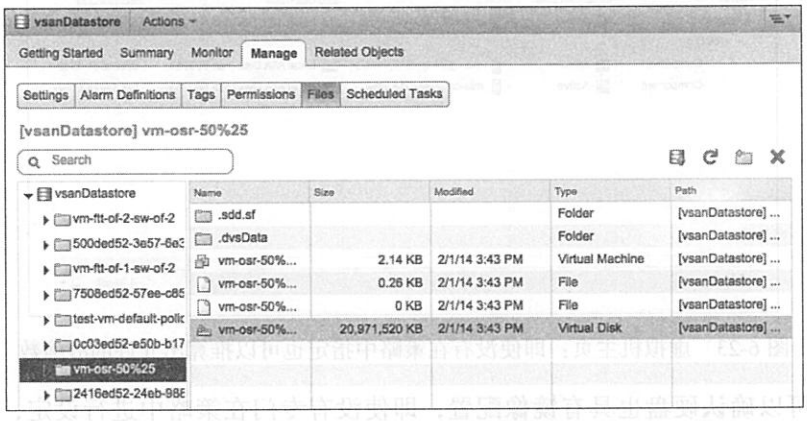


图 6-25　对象空间预留为 50%，需要保留 40GB 中的 20GB

如前所述，我们可以看见 40GB 的虚拟机硬盘文件已经部署好了，并且已经预留了 20GB 的磁盘空间，等于虚拟机存储策略里面为这台虚拟机配置的值——50%。

6.5　策略设置：FTT=1,OSR=100%

现在来看一下最后一个策略，这次要给我们的虚拟磁盘预留全部 100% 的空间。步骤和前面一样，就是先创建一个包含对象空间预留要求的策略，只是这次把值设成 100 而不是 50，如图 6-26 所示。你或许已经猜到了，这意味着我们将预先保留虚拟机磁盘的全部空间，与一个厚置备格式的虚拟磁盘文件类似。

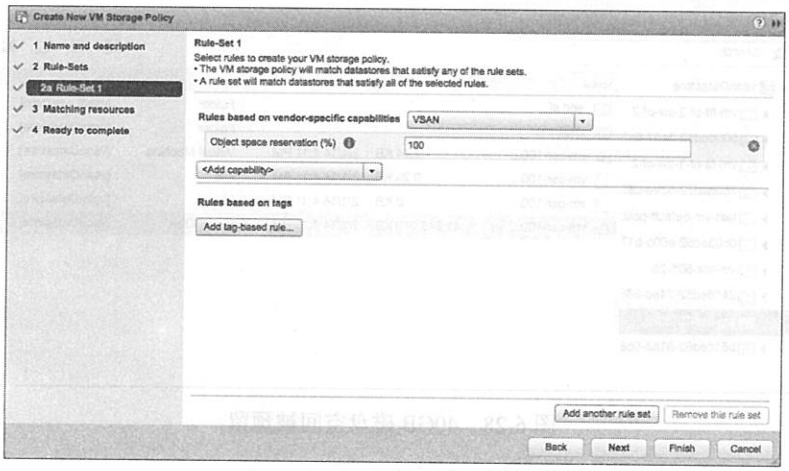


图 6-26　对象空间预留为 100%

设置的步骤和前面一样，策略仅仅包含一个设置——对象空间预留，只是这一次要设成 100%。和上次一样，虽然策略里面没有明确说明，但是隐含着可允许的故障数的设定为 1。

再一次，我们在部署虚拟机的时候选择这个策略，并且验证 VSAN 数据存储和所选的这个策略兼容，如图 6-27 所示。

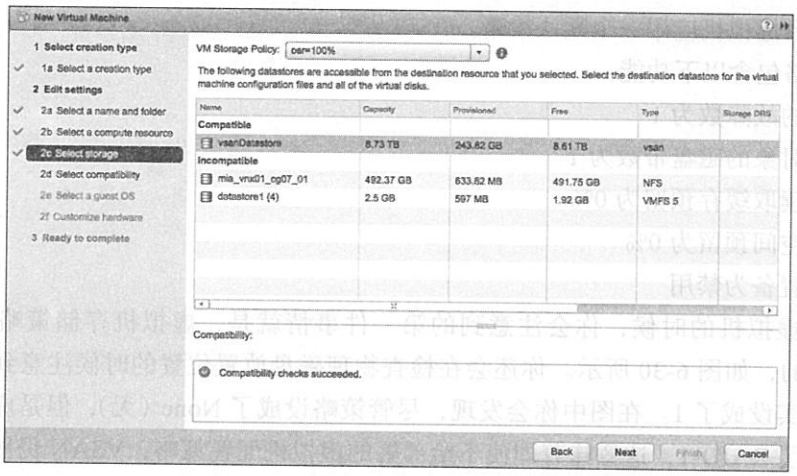


图 6-27　VSAN 数据存储和虚拟机存储策略兼容

下一步是判断为这个 VMDK 文件真正预留了多少磁盘空间？我们再次通过 Datastore > Manage > Files 视图来看一下到底事先预留了多少空间（如图 6-28 所示）。因为对象空间预留被设成了 100%，这次应该看见全部 40GB 的空间被预留了。

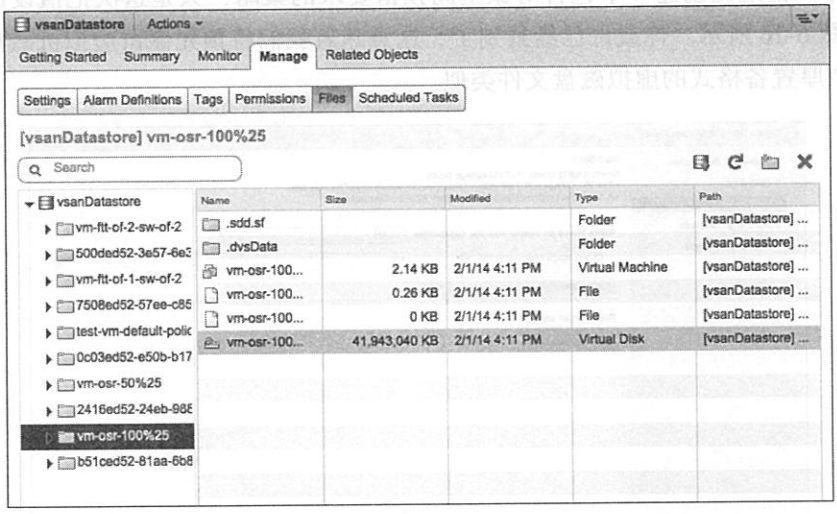


图 6-28　40GB 磁盘空间被预留

果然，全部 40GB 磁盘空间被事先预留了。

6.6　默认策略

你或许已经猜到 VSAN 有一个默认策略。这意味着如果对 VSAN 数据存储上部署的某台虚拟机没有选择任何策略（虚拟机存储策略选择的地方设成了 None，如图 6-29 所示），那么会对这台虚拟机应用一个默认策略。

默认策略包含以下功能：

- 允许的故障数为 1
- 每个对象的磁盘带数为 1
- 闪存读取缓存预留为 0%
- 对象空间预留为 0%
- 强制置备为禁用

在部署虚拟机的时候，你会注意到的第一件事情就是，虚拟机存储策略是被设置成 None（无）的，如图 6-30 所示。你还会在检查物理磁盘放置位置的时候注意到可允许的故障数的值其实设成了 1。在图中你会发现，尽管策略设成了 None（无），但是虚拟机对象已经被配置成了 RAID-1，这意味着即使不给部署的虚拟机配置策略，VSAN 仍然会自动通过默认策略来提供（高）可用性。

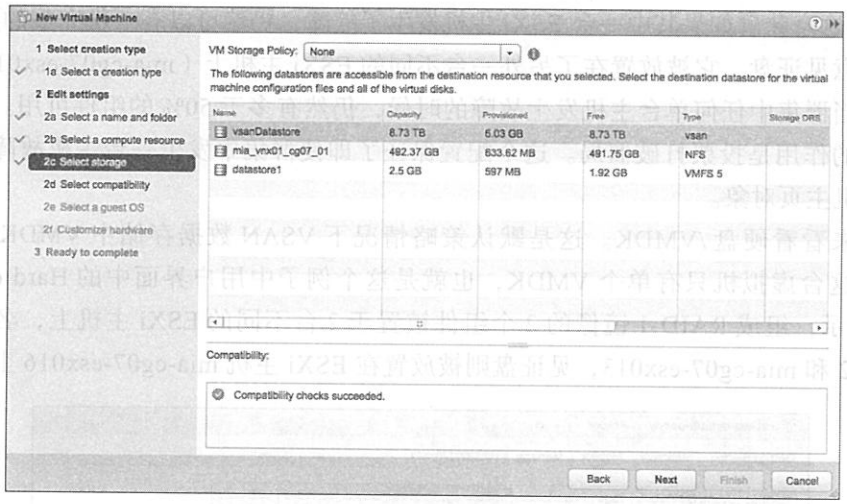


图 6-29　不选择策略的结果是使用默认策略

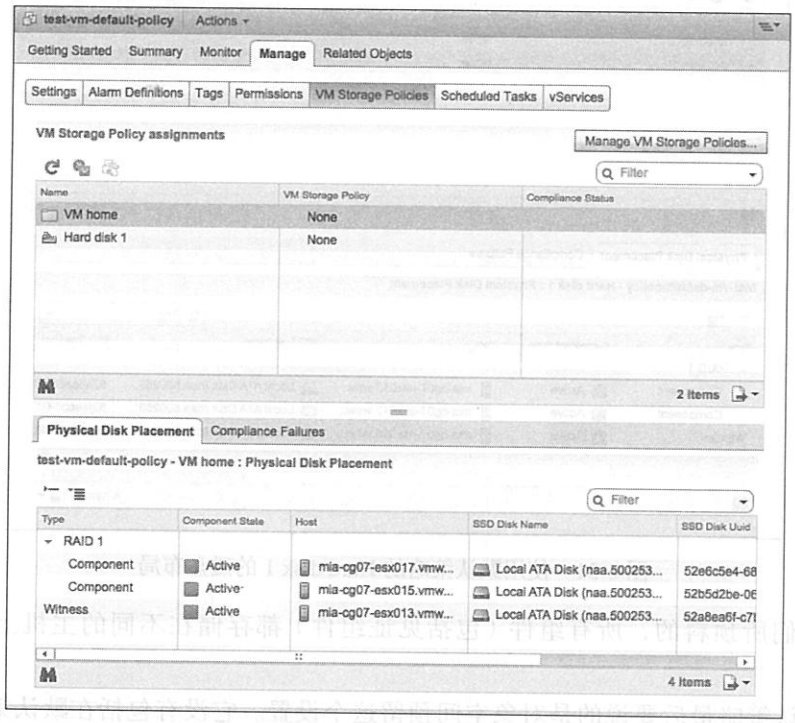


图 6-30　允许的故障数 =1 是默认策略的一部分

如图 6-30 所示，这台虚拟机并没有关联任何虚拟机存储策略。然而，如果观察一下虚拟机主页，我们会发现它已经被自动配置成了 RAID-1，数据已经有了镜像拷贝。组成 RAID-1 镜像的 2 个组件放置在 2 台不同的 ESXi 主机上（mia-cg07-esx017 和 mia-cg07-

esx015）。这意味着如果其中一台 ESXi 主机发生了故障，仍然可以有一份完整的数据可用。另外要注意见证盘，它被放置在了另外一台不同的 ESXi 主机上（mia-cg07-esx013）。这是为了确保当群集中任何单台主机发生故障的时候，仍然有多于 50% 的组件可用。见证在这个配置中的作用是投票打破僵局。这个配置保证了即使群集中发生一台主机故障仍然可以访问虚拟机主页对象。

接下来看看硬盘 /VMDK。这是默认策略情况下 VSAN 数据存储中 VMDK 的布局情况。因为这台虚拟机只有单个 VMDK，也就是这个例子中用户界面中的 Hard disk 1。如图 6-31 所示，组成 RAID-1 镜像的 2 个组件被置于 2 台不同的 ESXi 主机上，名字是 mia-cg07-esx12 和 mia-cg07-esx013，见证盘则被放置在 ESXi 主机 mia-cg07-esx016 上。

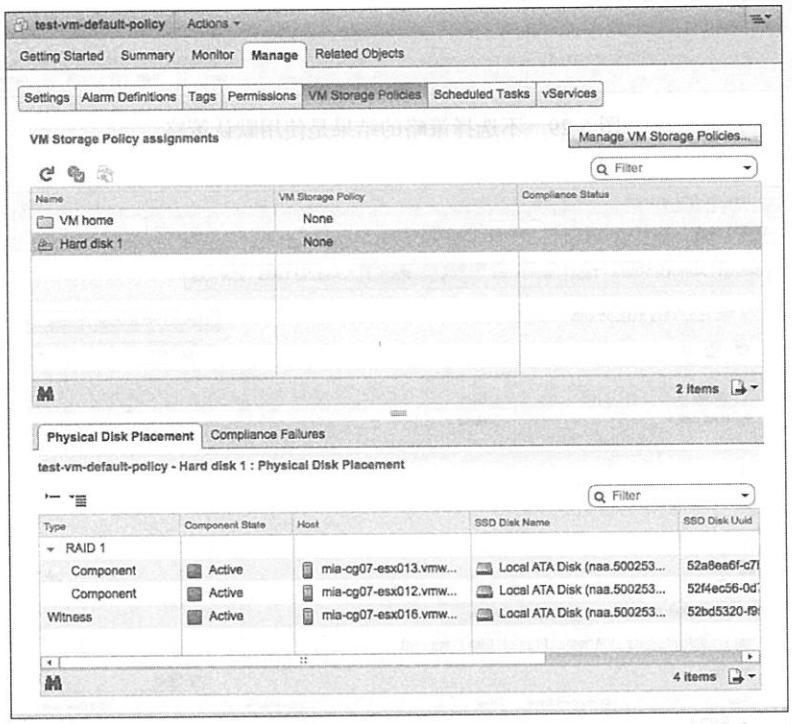


图 6-31　使用默认策略的 Hard disk 1 的磁盘布局

正如我们所预料的，所有组件（包括见证组件）都存储在不同的主机上来保证高可用性。

关于默认策略最后要说的是对象空间预留这个设置。它没有包括在默认策略中。而是在创建虚拟机的向导程序中设置 thin（精简置备）还是 thick（厚置备）的磁盘时实施的。如果部署虚拟机时不更改创建虚拟机向导程序中的默认配置，VSAN 数据存储上就会部署一块厚置备延迟置零（lazy zeroed thick, LZT）的 VMDK，这和将对象空间预留设置成 100% 一样。

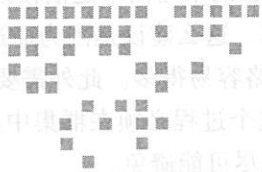
尽管存在着默认策略，VMware 建议管理员创建自己的策略而不是依赖默认策略来部署虚拟机。VMware 还提醒大家对于修改默认策略要谨慎。这么建议是因为通过在 vCenter 中创建一个虚拟机存储策略来满足需求要比修改默认策略容易得多。此外需要注意，编辑默认策略只能通过 ESXi 主机上的命令行来完成，而且这个过程必须在群集中的每一台 ESXi 主机上重复进行。这可能会引起人为的错误，所以应该尽可能避免。

6.7　小结

本章覆盖了虚拟机存储策略的创建和 VSAN 数据存储上的虚拟机部署。有一个策略设置没有包括在本章范围内，那就是闪存读取缓存预留。这只是因为从虚拟机布局的角度来看，这个设置无法在 vSphere Web 客户端的用户界面中观察到。不过这个值也是以百分比的方式配置的，这与对象空间预留配置的方式完全一致，也是 VMDK 大小的一个百分比。例如，在一个 40GB 的 VMDK 上设置了 1% 的闪存读取缓存预留，就会在闪存上保留 400MB 的读缓存用于那台指定的虚拟机。不过，如前所述，这无法通过 vSphere Web 客户端观察到。在第 10 章中，我们会告诉你如何通过使用 Ruby vSphere Console（RVC）来检查闪存读取缓存预留的数值。

没在本章讨论的另外一个策略设置是强制置备。同样，当这个设置配置进策略并用于虚拟机部署的时候，其效果也无法通过 vSphere Web 客户端观察到。如果强制置备用于部署一台虚拟机，那么只要一组完整的存储对象可以置备到 VSAN 数据存储上，那么虚拟机就可以部署成功。策略可能会包含诸如允许的故障数或条带宽度或闪存读取缓存预留等要求，只要强制置备被设定了，即使这些要求不能满足，虚拟机也会被部署。不过，此时虚拟机会在 vSphere Web 客户端的用户界面中被标注为不合规。当可以获得额外的资源的时候，虚拟机就会利用这些额外的资源进行重新配置（reconfigure）使之满足合规性的要求。一旦资源可用，VSAN 就会自动强制这些策略而无须管理员人工发起（重新配置）。现在你应该明白利用强制置备来部署不符合策略要求的虚拟机是危险的，这可能会在群集发生故障的时候导致虚拟机不可用。

有些虚拟机存储策略的表现可能不是显性的，例如默认策略设置，又例如允许的故障数为 1 这个特性已经隐藏在策略中了，此外对于某些虚拟存储对象只能应用部分策略设置，这些细微的差别已经在第 5 章中非常具体地解释过了。



Chapter 7 第 7 章

管理和维护

本章涵盖了常见的 VSAN 管理和维护流程及任务，还提供了一些常见工作流和与日常管理工作相关的例子。

7.1 主机管理

VMware VSAN 是一个既可横向扩展又可纵向扩展的存储架构，这意味着可以无缝地给 VSAN 群集添加额外的存储资源。这些存储资源可以是磁盘、包含 SSD 的完整的磁盘组，也可以是带有存储容量的主机。如果你有过一些 vSphere 环境管理的经验，就不会对 VSAN 的管理是如此简单感到惊讶——添加存储容量真的就像给群集加一台新的主机一样简单。让我们来深入研究一下其中的一些任务。

7.1.1 添加主机到群集

添加主机到 VSAN 群集的方法相当简单直接。当然，你必须保证主机满足 VSAN 的要求或推荐做法，例如专用的千兆网卡端口（建议万兆），又比如若主机要提供额外的存储空间则需要至少一块 SSD 和一块硬盘（HDD）。此外，尽管一些配置（如用于 VSAN 通信的 VMkernel 端口）也可以在主机加入群集以后再设置，它们应该提前配置好。在主机成功加入群集之后，根据新主机中额外 HDD 的大小，需要观察 VSAN 数据存储到底增加了多少空间。请注意，SSD 不会增加 VSAN 数据存储的容量。为了完整表述，使用 vSphere Web 客户端将主机添加到 VSAN 群集所需的步骤如下。

1. 右键点击群集对象并选择 Add Host。

2. 填入服务器的 IP 地址或主机名，如图 7-1 所示。

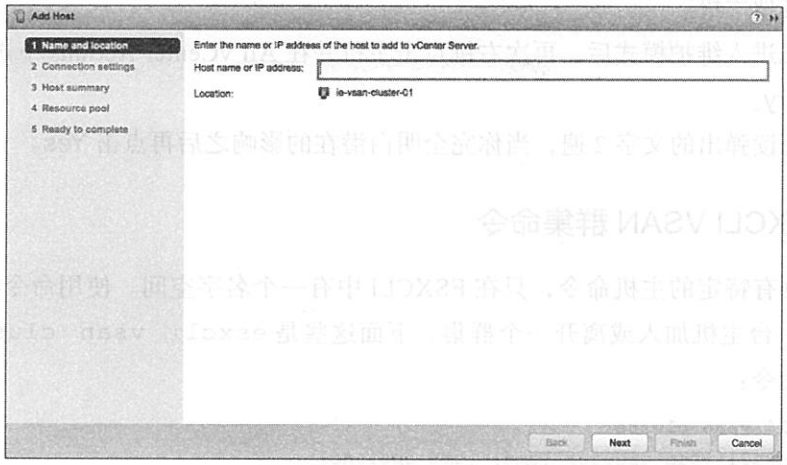


图 7-1　添加一台主机到群集

3. 填入用户账号（通常是 root）和密码。

4. 接受 SHA1 指纹算法的选项。

5. 在 Host Summary 窗口点击 Next。

6. 选择将要使用的许可证。

7. 如果需要可以启用 lockdown 模式并点击 Next。

8. 在 Resource Pool 选择的时候点击 Next。

9. 最后点击 Finish，完成添加主机到群集的操作。

如果你的群集配置成自动模式，这就完成了！如果没有配置成自动模式，需要手动创建磁盘组。本章后面会教你怎么做。

7.1.2　从群集中移除主机

如果你想从群集中移除一台主机，必须首先确保主机已经被置于维护模式，这将在下一节中详细讨论。主机被成功置于维护模式之后，就可以安全地从 VSAN 群集移除了。下面是使用 vSphere Web 客户端将一台主机从群集中移除的步骤。

1. 右键点击主机，然后选择 Enter Maintenance Mode，并从图 7-2 所示的屏幕中选择合适的 VSAN 数据迁移选项，然后点击 OK。

2. 现在所有虚拟机将会被迁移（vMotion）到其他主机。

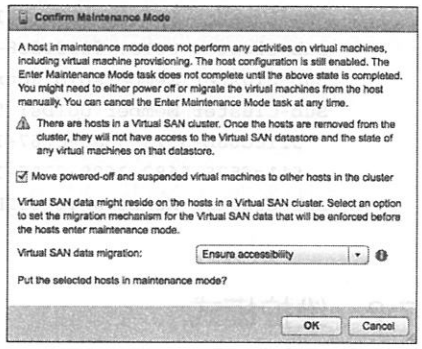


图 7-2　进入维护模式

3. 当迁移完成时，根据所选 VSAN 数据迁移选项的不同，VSAN 组件上的数据可能也会被复制到其他主机。

4. 当主机进入维护模式后，再次右键点击主机并在 All vCenter Actions 下选择 Remove from Inventory。

5. 仔细阅读弹出的文字 2 遍，当你完全明白潜在的影响之后再点击 Yes。

7.1.3 ESXCLI VSAN 群集命令

VSAN 没有特定的主机命令，只在 ESXCLI 中有一个名字空间。使用命令行（CLI）命令，可以使一台主机加入或离开一个群集。下面这些是 `esxcli vsan cluster` 命令行的部分基本命令：

```
~ # esxcli vsan cluster
Usage: esxcli vsan cluster {cmd} [cmd options]

Available Commands:
  get      Get the information of the VSAN cluster that this host is joined to.
  join     Join the host to a given VSAN cluster.
  leave    Leave the VSAN cluster the host is currently joined to.
  restore  Restore the persisted VSAN cluster configuration.
~ #
```

经常在排错练习时用到的一个命令是 `get` 命令。`get` 命令可以用来在命令行获取群集的配置信息，常用来比较主机之间的配置，例如：

```
~ # esxcli vsan cluster get
Cluster Information
   Enabled: true
   Current Local Time: 2013-03-18T12:09:11Z
   Local Node UUID: 511b62c3-96e6-434e-6839-1cc1de253de4
   Local Node State: MASTER
   Local Node Health State: HEALTHY
   Sub-Cluster Master UUID: 511b62c3-96e6-434e-6839-1cc1de253de4
   Sub-Cluster Backup UUID: 511cc68b-352a-5cae-cf67-1cc1de252264
   Sub-Cluster UUID: 523845c8-73c9-5d99-0393-9ef20a328714
   Sub-Cluster Membership Entry Revision: 10
   Sub-Cluster Member UUIDs: 511b62c3-96e6-434e-6839-1cc1de253de4,
     511cc68b-352a-5cae-cf67-1cc1de252264,
     511cd526-5682-3688-8206-1cc1de253a92
   Sub-Cluster Membership UUID: 56092451-245f-9c0c-29f6-1cc1de253de4
```

7.2 维护模式

前面在讨论从 VSAN 群集中移除主机时我们简要地提到了维护模式，在 VSAN 的环境

中维护模式包含了新的功能，接下去我们会详细解释。以前，当一台 ESXi 主机被置于维护模式时，所需做的只是将其上所有虚拟机迁出那台主机，然而，在 VSAN 环境下，进入维护模式时还提供了数据迁移选项。VSAN 维护模式的选项和数据清空有关，具体介绍如下。

- **确保可访问性（Ensure Accessibility）**：这个选项会在进入维护状态的主机上迁出足够多的数据，以确保即使这台主机宕机后所有的虚拟机存储对象仍可被访问到。VSAN 并不迁出所有数据，而是检查那些会在主机被置于维护模式时丢失简单多数组件或数据可用性的存储对象，对这些对象创建足够数量的拷贝来解决这个问题。这台主机上的对象的组件如果没有高可用性，就会因为主机置于维护模式而无法被访问到，这种情况下，VSAN（或者更精确地说是 CLOM）就必须重新配置这些对象。例如，当虚拟机配置了 FTT 为 0 时，这种情况就可能发生。确保可访问性是进入维护模式向导程序中的默认选项，也是 VMware 的推荐选项。
- **迁移全部数据（Full Data Migration）**：这个选项会迁出所有数据，本质上就是为置于维护模式的那台主机本地磁盘存放的每一块数据都创建一个拷贝。VSAN 并不需要完全从那台进入维护模式的主机复制数据，而是可以（并也会利用）从所有拥有对象副本的主机复制数据，以避免在进入维护模式的主机上制造瓶颈。换言之，在一个 8 台主机的群集中，当一台主机用迁移全部数据的方式进入维护模式时，所有 8 台主机都可能参与受影响组件的重建过程。只有在所有受影响的对象都完成重新配置、所有组件都被放置到群集中另外的主机上并且合规性得到保证的情况下，主机才会成功进入维护模式。
- **不迁移数据（No Data Migration）**：这个选项不对存储对象进行任何操作。如果主机在进入维护模式之后关机，情况等同于主机崩溃。

注意，当主机进入维护模式后，VSAN 仍然在主机上进行操作、访问和提供数据服务。只有在主机从群集中移除或者关机时，VSAN 才会停止使用该主机（或者，当然，当你决定把全部数据都迁出，那么当数据迁移完成，“旧”的组件被移除后，VSAN 也不再使用该主机）。

图 7-3 显示的是当一台或多台主机置于维护模式时可以选择的选项，默认的预先选定的数据迁移建议是 Ensure Accessibility（确保可访问性）。

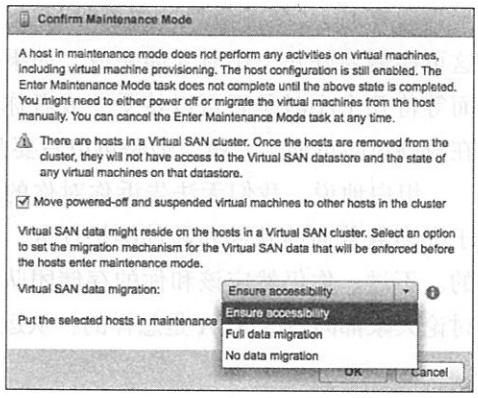


图 7-3　维护模式的选项

维护模式数据迁移选项的建议

先拿普通存储环境做个比较：在升级的时候，通常会采用滚动升级的方法——也就是说如果有 2 个控制器，当其中一个在升级的时候会由另外一个来处理 I/O。即使是这样，也还

是有风险的。区别在于，作为一个虚拟化管理员，**你拥有更多弹性**，因此会**希望某些特性**可以达到预期效果，例如 vSphere High Availability（HA）。扪心自问，你**愿意**承受何种程度的风险，并且**能够**承受何种程度的风险？

从 VSAN 的角度来说，当需要把一台主机置于维护模式时，请考虑下列问题：

- **为什么要将主机置于维护模式？**是要升级主机并且可预见的不可用时间很短？还是彻底将一台主机从群集中移除？问题的答案对于要选择维护模式的哪个数据迁移选项非常重要。
- **现有多少台主机？**只有 3 台主机时，你只能选择 Ensure Accessibility（确保可访问性），这是因为 VSAN 总是需要至少 3 台主机来进行存储（2 个副本和 1 个见证）。
- **迁移需要多长时间？**
  - 用的磁盘是什么类型的？ SAS 还是 SATA？
  - 网络是万兆还是千兆？
  - 群集有多大？
- **需要把数据从 1 台主机迁往另一台以保持可用性水平吗？**只有现存的组件需要被迁移，而不是主机的“原始空间”，也就是说，如果 8TB 空间只使用了 6TB，那么只有 6TB 需要迁移。
- **是否只需要确保数据可访问性，并可以承受维护期间潜在的宕机风险？**只有处于风险中的组件才会被移走。例如，如果 6TB 空间中只有 500GB 处于风险中，那么就只有那 500GB 才会迁走。

关于维护模式的数据迁移模式还有一些需要说明。当为了增加可用性水平而选择迁移全部数据时，“维护窗口”也被延长了，因为需要通过网络在主机之间复制几 TB 的数据，这可能需要好几个小时才能完成。如果你的 ESXi 主机重启需要大概 20 分钟，为数据迁移而等待几个小时是否可以接受？或者你还是愿意承受一些风险，通知用户因为维护本身存在较高风险可能会宕机，但是可以只要几分钟就完成而不是几个小时。

坦白地说，我们无法告诉你对你的组织来说哪种方法是最好的。不过我们的确觉得对于大多数普通的软件和硬件维护任务，用确保可访问性维护模式数据迁移选项是可以接受的。不过，你仍然应该和你的存储团队讨论**所有的**方法并了解它们的工作流，和业务伙伴讨论大家都同意的 SLA 是怎样的？从运营的角度来看哪种才是最合适的？

7.3 磁盘管理

如前所述，VSAN 的设计目标之一是能够横向扩展存储容量，这要求有能力添加新磁盘、把磁盘替换为容量更大的磁盘或简单地更换故障磁盘。本节将讨论在 VSAN 环境下完成这些任务的步骤。

7.3.1　添加一个磁盘组

第 2 章中介绍过如何添加一个磁盘组，不过为了完整起见，这里再列一下具体的步骤。

1. 在左侧窗格中点击 VSAN 群集。
2. 在右侧选择 Manage。
3. 点击 Settings 和 Disk Management。
4. 如图 7-4 所示，在群集中选择一台主机并点击带有绿色加号（+）的图标。

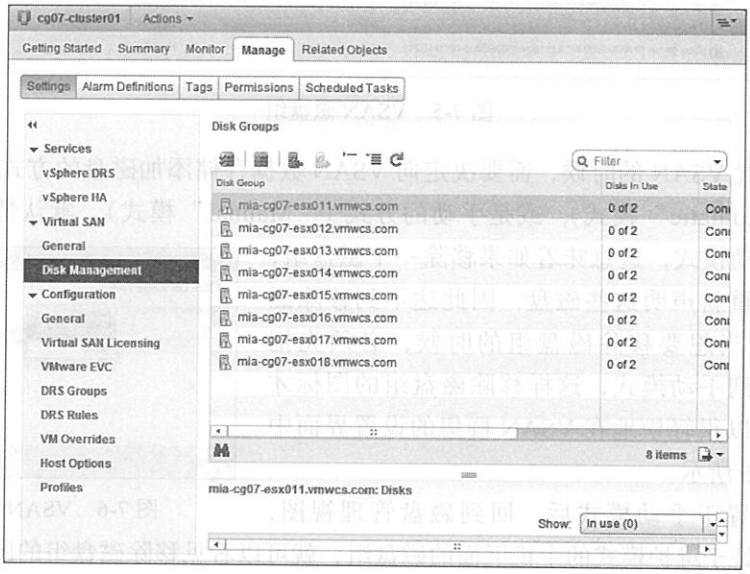


图 7-4　VSAN 磁盘管理

5. 选择所有要加入此磁盘组的磁盘和闪存设备并点击 OK。

现在一个磁盘组就创建好了，这只需要几秒钟的时间。

注意，如果想这么做，你也可以在所有主机上同时创建新的磁盘组。这很简单，只需要在点击新建磁盘组图标的时候不要选择主机，就能做到在群集的所有主机上添加一个新的磁盘组。

7.3.2　移除一个磁盘组

在开始这个任务之前，我们建议先把要进行磁盘组移除操作的 ESXi 主机置于维护模式。这对于移除磁盘组的操作不是必要步骤，但是我们相信大多数管理员都希望在删除前先把此磁盘组中的虚拟机组件移到群集中的其他磁盘组上。如果你不进行这个迁出数据的步骤，就可能造成“降级的”组件，这样在 VSAN 进行重新配置的时候这些对象就不再具有高可用性。因此我们建议（主机）要先置于维护模式。如果你打算在进入维护模式时迁移全部数据，你应该先确认群集中是否具有足够的磁盘空间。

完成这个步骤后，主机就进入了维护模式，如图 7-5 所示，现在就可以移除磁盘组了。不过，根据 VSAN 配置方式的不同，移除磁盘组的图标可能不会显示在磁盘组视图中。

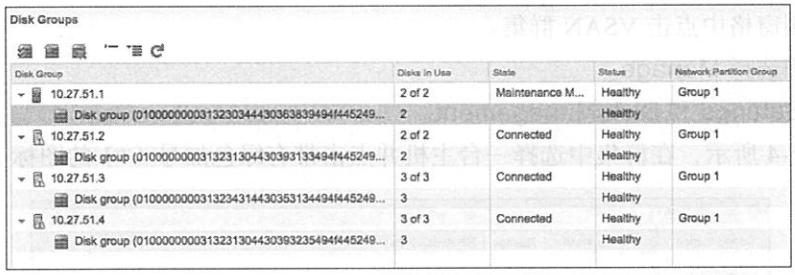


图 7-5　VSAN 磁盘组

在最初配置 VSAN 的时候，需要决定向 VSAN 数据存储添加磁盘的方式，这可以是全自动的（“Automatic”模式），或是手动的方式（“Manual”模式）。默认情况下，VSAN 会被配置为自动模式，这意味着如果移除一个磁盘组，VSAN 会立刻重新声明这些磁盘，因此达不到移除磁盘组的目的。当想要移除磁盘组的时候，必须先把 VSAN 群集改到手动模式，这样移除磁盘组的图标才会出现。这可以很轻松地在 VSAN 群集的设置界面中完成，如图 7-6 所示。

图 7-6　VSAN 群集模式

把 VSAN 置于手动模式后，回到磁盘管理视图，点击选择那台处于维护模式的主机下面的磁盘组，就可以看见移除磁盘组的图标了（有一个红色的 X）。现在可以进行移除磁盘组的操作了。

7.3.3　向磁盘组添加磁盘

如果 VSAN 配置成自动模式，就不存在这个问题。VSAN 群集会自动声明新的或现存的磁盘并向 VSAN 数据存储提供容量。

然而，如果群集配置成手动模式，就需要手动去向磁盘组添加新磁盘用以给 VSAN 数据存储增加容量。用 vSphere Web 客户端可以轻松地完成这个操作。先导航到 VSAN 群集，选择 Manage（管理）页，然后选择 VSAN Disk Management（磁盘管理），接下去新的磁盘就可以被声明用于磁盘组了。如果你的磁盘没有显示出来，请再对磁盘控制器做一次重新扫描。选定一台主机，并点击 Claim Disks（声明磁盘），此时会显示出一个群集中所有 ESXi 主机的列表以及每台主机的可用磁盘。另外还有一个方法，可以让你选择特定的磁盘组，并在每一台主机上依次添加磁盘到磁盘组。根据所选方法的不同，简单地选择想要加入到 VSAN 群集的磁盘然后点击 OK。图 7-7 显示了一个 4 节点群集，其中要声明的磁盘都已经被选定。

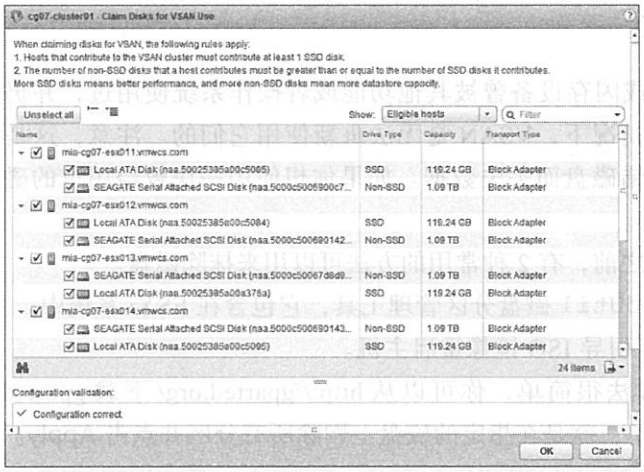


图 7-7　声明磁盘

7.3.4　从磁盘组中移除磁盘

和前面讨论的移除磁盘组类似，只有在群集处于手动模式的时候，才可以通过 vSphere Web 客户端来将磁盘从磁盘组中移除。如果群集处于自动模式，VSAN 群集会直接重新声明刚刚被移除的磁盘。在群集置于手动模式的情况下，导航 VSAN 群集的 Disk Management（磁盘管理）部分，选择一个磁盘组，此时移除磁盘的图标（带有红色 X 的一块磁盘）就会在用户界面（UI）中显示出来，如图 7-8 所示。请注意，当群集处于自动模式的时候，这个图标是不可见的。

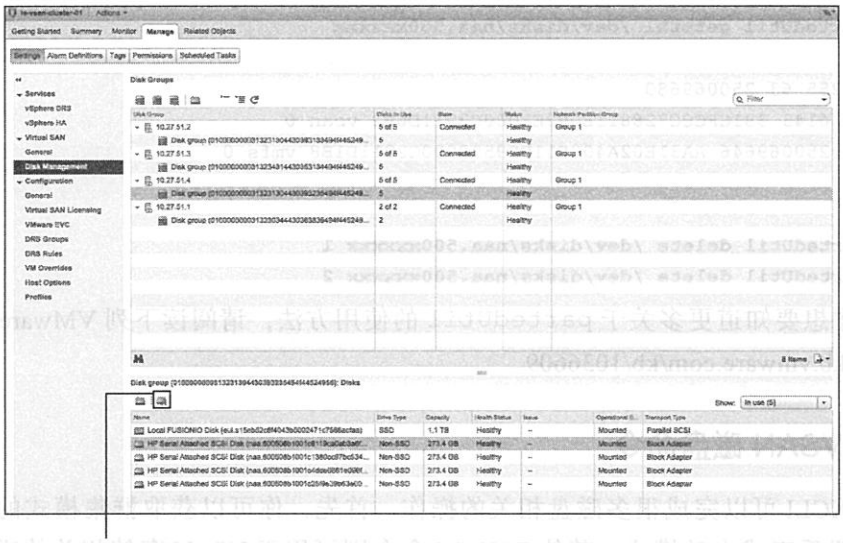


图 7-8　从磁盘组移除磁盘

7.4 抹除磁盘

有时候，磁盘或闪存设备曾被其他功能或者操作系统使用过，并仍然拥有分区甚至是文件系统，在这种情况下，VSAN 是无法重新使用它们的。注意，这是刻意为之的，目的是为了防止用户选错磁盘而丢失数据。如果你想使用一块曾经用过的磁盘，你需要手工抹除磁盘的所有数据。

在用于 VSAN 之前，有 2 种常用的方法可以用来抹除磁盘。

- 使用 partedUtil 磁盘分区管理工具，它包含在 ESXi 系统中。
- 用 gparted 可引导 ISO 镜像重启主机。

用 gparted 的方法很简单，你可以从 http://gparted.org/ 下载 gparted 的 ISO 镜像文件。而使用的方法很简单，就是在指定的磁盘上删除所有分区并点击 Apply。

警告 抹除磁盘的任务具有破坏性，在抹除磁盘后想要恢复数据几乎是不可能的。

使用 ESXi 内置的 partedUtil 方法稍微有点复杂，因为这是一个命令行工具。下面是使用 partedUtil 工具来抹除一块磁盘的必要步骤。如果你不确定要抹除哪个设备，请务必用 esxcli storage core device list 命令再重复检查一次设备 ID。

```
esxcli storage core device list:
 ~ # partedUtil get  /dev/disks/naa.500xxxxxx
15566 255 63 250069680
1 2048 6143 0 0
2 6144 250069646 0 0

~ # partedUtil getptbl /dev/disks/naa.500xxxxxx
gpt
15566 255 63 250069680
1 2048 6143 381CFCCC728811E092EE000C2911D0B2 vsan 0
2 6144 250069646 AA31E02A400F11DB9590000C2911D1B8 vmfs 0
~ #

~ # partedUtil delete /dev/disks/naa.500xxxxxx 1
~ # partedUtil delete /dev/disks/naa.500xxxxxx 2
```

如果你想要知道更多关于 partedUtil 的使用方法，请阅读下列 VMware 知识库文章：http://kb.vmware.com/kb/1036609。

ESXCLI VSAN 磁盘命令

用 ESXCLI 可以完成很多磁盘相关的操作。首先，你可以获取群集模式的信息或是给群集设置手工或自动模式。其他 ESXCLI 命令则可用于 VSAN 存储相关的磁盘和磁盘组，你可以添加、移除或列出一个磁盘组中的磁盘。这将同时列出 SSD 和 HDD。下面的

esxcli vsan storage list 命令的输出重点显示出：2 个设备是否是 SSD，以及它们属于哪个磁盘组。

```
~ # esxcli vsan storage list
naa.5000c5002bd7526f
   Device: naa.5000c5002bd7526f
   Display Name: naa.5000c5002bd7526f
   Is SSD: false
   VSAN UUID: 52db9f60-57b8-ad88-70eb-889f3c72b5e1
   VSAN Disk Group UUID: 521f9dda-efda-4718-e75d-aec63eb6fbd4
   VSAN Disk Group Name: naa.500253825000c296
   Host UUID: 519f364d-ef04-8d94-8ad4-1cc1de252264
   Cluster UUID: 520bff2a-badc-0cdd-7be7-70e5e5ae032f
   Used by this host: true
   In CMMDS: true
   Checksum: 11848694795517181960
   Checksum OK: true
naa.500253825000c296
   Device: naa.500253825000c296
   Display Name: naa.500253825000c296
   Is SSD: true
   VSAN UUID:
   VSAN Disk Group UUID: 521f9dda-efda-4718-e75d-aec63eb6fbd4
   VSAN Disk Group Name: naa.500253825000c296
   Host UUID: 00000000-0000-0000-0000-000000000000
   Cluster UUID: 00000000-0000-0000-0000-000000000000
   Used by this host: true
   In CMMDS: true
   Checksum: 12800345249350977942
   Checksum OK: true
```

如前所述，通过命令行也可以从磁盘组中移除磁盘或闪存设备，不过这应该尤其小心，最好是通过前面几页提到过的用户图形界面（UI）来完成。

7.5 故障场景

我们已经在第 5 章中探讨过一些故障场景，并解释了“失联”（absent）组件和“已降级”（degraded）组件的不同之处。不过，再从运营的角度来理解一下磁盘、SSD 或主机故障会造成怎样的影响也不错。在开始讨论之前，让我们首先来回顾一下 2 种不同的故障状态，因为这是从运营角度考量的基础。

- Absent（失联）：VSAN 并不知道消失的组件发生了什么。典型的例子是主机故障。此时 VSAN 会默认等待 60 分钟，之后再创建新的副本组件。
- Degraded（已降级）：VSAN 知道消失的组件发生了什么。一个典型的例子是发生在一块 SSD 或一块磁盘已经彻底坏掉的时候。此时 VSAN 立刻就会创建新的组件，来

使得受影响的对象尽快合规于其所选的策略。

现在你知道了不同的状态有哪些，让我们再看一看“最”常见的故障类型，以及它们的影响。

7.5.1 磁盘故障

磁盘故障可能是任何存储环境中最常见的故障了，VSAN 也不例外。原因很简单：磁盘有活动部件。当然问题在于 VSAN 如何处理磁盘故障？如果故障发生时正好在磁盘上有一个读或写的操作会发生什么？

如果存储组件返回一个读错误，VSAN 会去检查是否存在副本组件，如果有则从那个副本读取。默认情况下每个对象被创建时都配置成 FTT 为 1，这意味着每个对象总有 2 个完全一样的副本组件可用。故障发生在读取时有 2 种不同的情况，第一种情况是问题可以解决，另一种情况是问题无法解决。当问题是可以解决的时候，I/O 错误会被汇报给对象的所有者，对象的所有者则会发起组件重构。当组件重构完成时，故障组件会被删除。然而，如果因为某种原因，没有副本组件存在（这种情况的可能性很小，因为这需要管理员手工创造一个特别的策略才行），VSAN 就会报告这个虚拟机出现了 I/O 错误。

写故障也会传送到对象所有者，也同样会在 VSAN 群集中另外的磁盘上触发组件重构。当组件重构完成时，群集目录（CMMDS）会被更新。注意，闪存设备（它没有出错）会继续用缓存来提供读取服务。

在某一个（或多个）组件因为故障而处于重建的过程中，现在的 vSphere Web 客户端中没有显示有多少需要同步的数据。不过 Ruby vSphere Console（RVC）工具中有一条非常有用的命令 `vsan.resync_dashboard` 可以让你来进行验证。

```
/localhost/CH-Datacenter/computers/cluster> vsan.resync_dashboard
2013-12-12 16:56:58 +0000: Querying all VMs on VSAN ...
2013-12-12 16:56:58 +0000: Querying all objects in the system from 10.20.177.18 ...
2013-12-12 16:56:59 +0000: Got all the info, computing table ...
+-----------------------------------------------------------------------------+----------------+---------------+
| VM/Object                                                                   | Syncing objects | Bytes to sync |
+-----------------------------------------------------------------------------+----------------+---------------+
| win1                                                                        | 1              |               |
| [vsanDatastore] 9a3f9352-346a-f78d-3360-1cc1de253de4/win1-000001.vmdk       |                | 48.00 GB      |
+-----------------------------------------------------------------------------+----------------+---------------+
| Total                                                                       | 1              | 48.00 GB      |
+-----------------------------------------------------------------------------+----------------+---------------+
/localhost/CH-Datacenter/computers/cluster>
```

7.5.2 闪存设备故障

如果闪存设备不可访问会发生什么情况？当闪存设备不可访问时，那块闪存设备支持的所有硬盘都会无法被访问。闪存设备故障等同于闪存设备背后的所有磁盘故障。从本质上来说，当一个闪存设备故障时，整个磁盘组被认为是“已降级的”。如果 VSAN 群集中有多余的容量，它就会试图在另一台主机或磁盘上重新配置存储对象。

因此，从运营和架构决策的角度来看，根据使用到的主机类型的不同，创建多个小的磁盘组可能要比单个大磁盘组好，因为一个磁盘组可以被视为一个故障域，如图 7-9 所示。

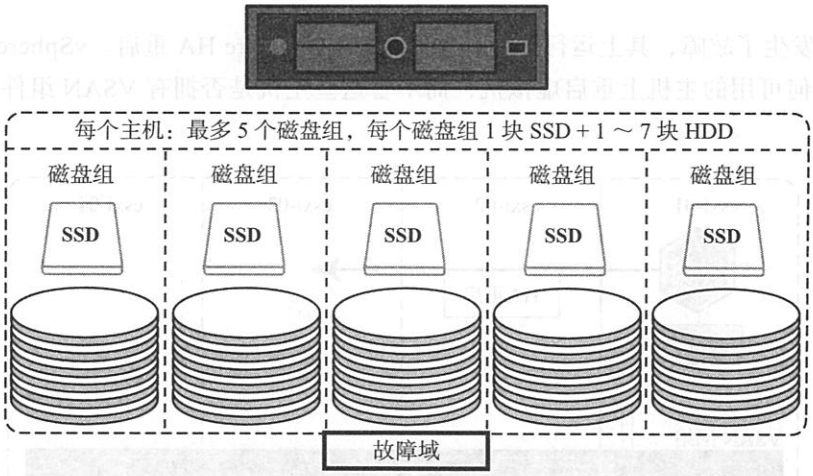


图 7-9　VSAN 磁盘组

7.5.3　主机故障

假设 VSAN 虚拟机存储策略已经创建并设置了允许的故障数为 1，VSAN 群集中的一台主机故障和连接到常规存储设备的普通群集中的主机故障类似。当然，主要的不同点在于当故障 VSAN 主机恢复正常时，它会含有不同步的对象组件。幸运的是，VSAN 带有一种机制，一旦主机恢复就会立刻同步所有组件。

出现主机故障情况 60 分钟后，VSAN 会启动组件重建，因为此时主机能在合理的时间内恢复的可能性已经很小了。当存储对象重构完成时，群集目录（CMMDS）会被更新。

如果原先故障的主机恢复并重新加入了群集，VSAN 会检查对象重构状态。如果对象已经在其他一个或多个节点上完成了重构，就不会有其他动作。如果对象重构仍在进行中，而且 VSAN 认为同步原先故障主机上的组件更快，那么就会恢复原先主机上的副本组件，并丢弃最近的（但是未完成同步的）拷贝。反之，（如果对象重构已经完成），那么就会丢弃故障主机上的原组件。

你现在可能会对 VSAN 组件的重新同步的工作原理产生一些疑惑。当对象的组件由于主机、网络或磁盘故障而无法同步的事件发生时，VSAN 会维护一个被变更的数据块的点位图（bitmap）。在故障修复之后，这个点位图使得由 2 个或更多组件构成的对象得以恢复。让我们用一个例子来进行解释。假设现在有一台主机上含有对象 X 的副本 A，当这台主机出现问题与群集的其余部分相隔离时，X 剩下的组件超过了所需的最小数量因此数据仍然可以被访问到，所以能继续工作并对读写操作提供服务。当 A “失联”时，所有对 X 进行

的写操作会被 VSAN 持续不断地记录进一个点位图。如果包含有副本 A 的被隔离主机恢复了，且 VSAN 决定将其和对象 X 的其他组件重新整合到一起，点位图就会被用来同步组件 A。

当主机发生了故障，其上运行的所有虚拟机会被 vSphere HA 重启。vSphere HA 可能会在群集中任何可用的主机上重启虚拟机，而不管这些主机是否拥有 VSAN 组件，如图 7-10 所示。

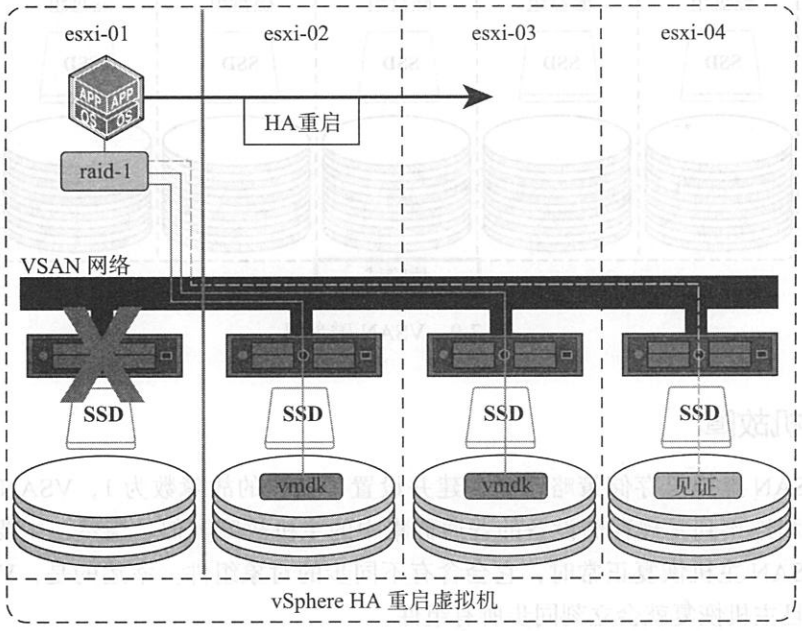


图 7-10 VSAN 主机 1 发生了故障，HA 触发重启操作

当发生主机隔离的事件时，vSphere HA 能够也会重启受影响的虚拟机。因为这种情况略略有点复杂，让我们再进一步讨论一下。

7.5.4 网络分区

网络分区（Network Partition）[⊖]可能发生在网络发生故障时。换而言之，某些主机在群集的这一边，而剩下的主机则位于群集的另一边。当出现网络分区事件时，VSAN 会显示相关网络错误配置问题的警示信息。

在前面的章节中解释过主机和磁盘故障的场景，现在是时候描述一下 VSAN 群集是如何处理隔离和分区的了。让我们首先来看一个典型的场景并基于这个例子来解释一下在网络发生分区时到底发生了些什么。

在图 7-11 描述的场景中，VSAN 中有一台虚拟机运行在 ESXi-01 上。这台虚拟机是通

⊖ 也叫网络隔离。——译者注

过一个设置了 FTT 为 1 的虚拟机存储策略来置备的。

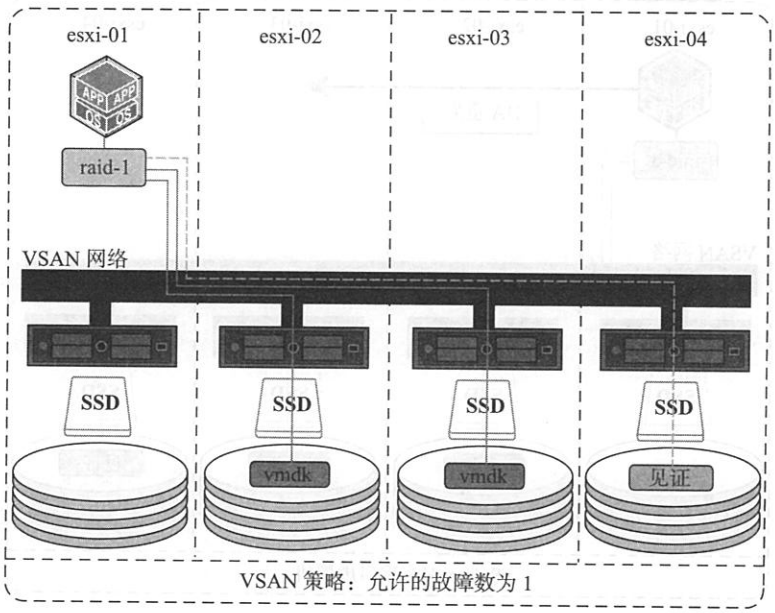


图 7-11　VSAN I/O 流：允许的故障数为 1

因为 VSAN 具有这样的能力——虚拟机运行的主机可以不拥有这台虚拟机的活动存储组件，于是问题就出现了：当网络被隔离的时候会发生什么情况？如你所能想象的一样，这里 VSAN 网络起着重要的作用，当你意识到它还被 HA 用作网络的心跳，你会发现它的作用原来是如此巨大。注意，vSphere HA 网络是被 VSAN 自动重新配置的，用来保证正确的网络可以处理这些场景。如果这种情况发生的话，下面的这些步骤描述了 vSphere HA 和 VSAN 是如何响应一个隔离事件的：

1. HA 将会探测到无法从 esxi-01 接收到心跳。
2. HA 主控主机会试图 ping 从属主机 esxi-01。
3. HA 会宣告从属主机 esxi-01 不可用。
4. VM 虚拟机会在某一台其他主机上被重启（在这个例子中是 esxi-03，如图 7-12 所示）。

还有个问题：如果网络的故障很糟糕，esxi-01 和 esxi-02 位于网络分区的同一侧，这会发生什么后果？嗯，这就是见证发挥作用的时候了。先来看看图 7-13，这能帮助大家理解。

这里的情况复杂程度略有增加。现在有 2 个分区，一边跑着虚拟机并且也有虚拟机的虚拟磁盘（VMDK），另一个分区则有一个 VMDK 副本和一个见证。猜猜看会发生什么？对了，VSAN 会用见证来判断哪一边的分区具有简单多数的组件，并且基于结果来决定哪一边的分区获胜。在这个例子中，分区 2 具有超过 50% 的对象组件，因此是胜者。这意味着虚拟机将会在 esxi-03 或 esxi-04 上被 vSphere HA 重启。注意，只有在你配置了相应的主机隔离响应时，分区 1 的 虚拟机才会被关机。

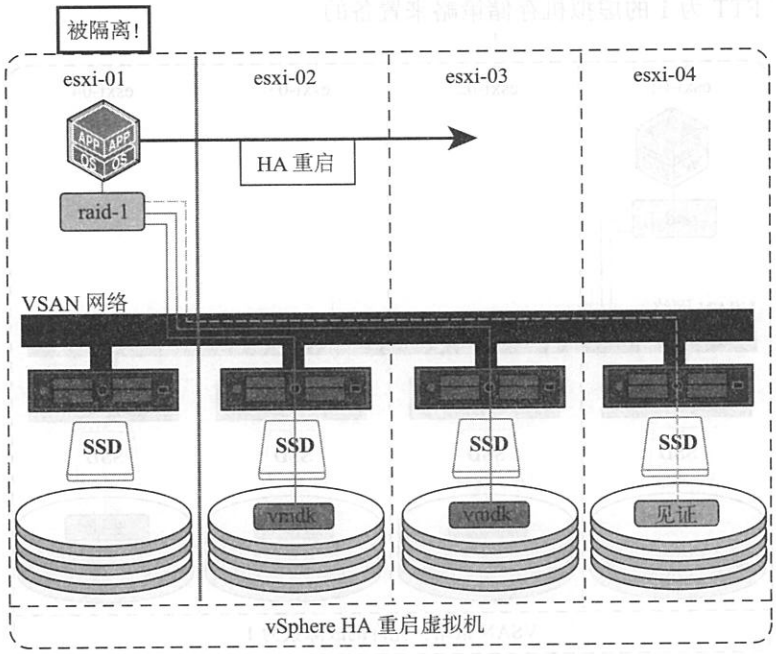


图 7-12　有一台主机被隔离的 VSAN 分区：HA 重启虚拟机

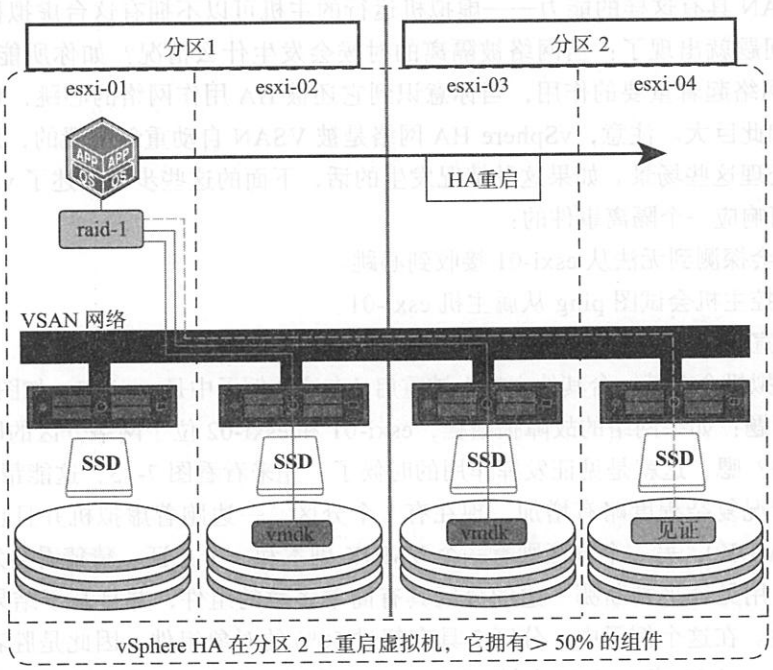


图 7-13　VSAN 分区，多台主机被隔离：HA 重启虚拟机

技巧　我们想强调的是，这个配置是我们强烈推荐的！（隔离响应→关机。）

但是如果 esxi-01 和 esxi-04 都被隔离了，会发生什么情况？图 7-14 显示了这种场景。

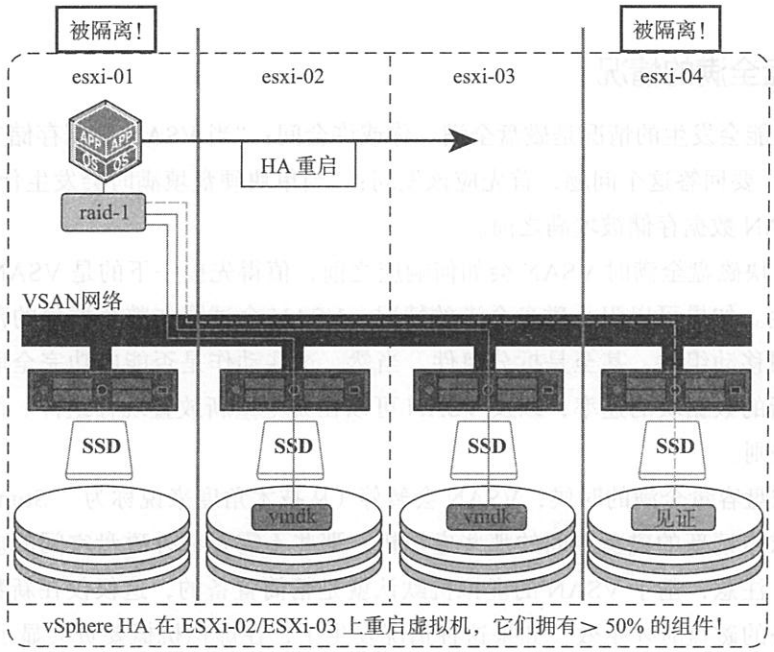


图 7-14　VSAN 中有 2 台主机被隔离：HA 重启虚拟机

还记得在前面讨论过的规则吗？

基于某个分区中可用组件的百分比来宣告胜者。

如果某个分区能够访问超过 50% 的（某个对象的）组件，就获胜。对于每个对象来说，最多只能有一个获胜分区。这意味着当 esxi-04 和 esxi-01 都被隔离时，只有 esxi-02 或 esxi-03 可以重启此虚拟机，因为此对象有 66% 的组件存在于群集的这个分区中。

要防止这些情况发生，必须保证 VSAN 网络是高可用的，例如通过网卡绑定和冗余的网络交换机，这些曾在第 3 章中讨论过。

如果因为某些原因 vCenter 不可用了，如果要获取 VSAN 网络的信息，你可以通过 ESXi 的命令行来实现。通过 CLI 管理员可以检查或移除 VSAN 网络配置。下面的例子中，你可以发现用于群集通信的 VMkernel 网络接口及其使用的 IP 协议：

```
~ # esxcli vsan network list
Interface
   VmkNic Name: vmk2
   IP Protocol: IPv4
```

```
    Interface UUID: 06419f51-ec79-0b57-5b3e-1cc1de252264
    Agent Group Multicast Address: 224.2.3.4
    Agent Group Multicast Port: 23451
    Master Group Multicast Address: 224.1.2.3
    Master Group Multicast Port: 12345
    Multicast TTL: 5
~ #
```

7.5.5 磁盘全满的情况

另一个可能会发生的情况是磁盘全满。你或许会问："当 VSAN 数据存储空间全满会发生什么情况？"要回答这个问题，首先应该发问："当单块硬盘填满时会发生什么？"因为这会发生在 VSAN 数据存储被填满之前。

在解释一块磁盘全满时 VSAN 会如何响应之前，值得先提一下的是 VSAN 会试图防止这种情况发生。如果可以阻止磁盘全满的情况，VSAN 会试图在整个群集的范围内平衡容量，或者来回移动组件，甚至是拆分组件。当然，这些动作是否能成功完全取决于虚拟机声明和填充新的数据块的速率，以及 VSAN 可以在哪里重新放置现存组件。简单地说，就是遵循物理法则。

当发生磁盘容量全满的时候，VSAN 会暂停（从技术角度来说称为"Stun"）试图写数据并为写请求申请新的磁盘空间的那些虚拟机。那些不需要额外磁盘空间的虚拟机可以继续正常运行。注意，基于 VSAN 的虚拟机默认就是精简置备的，这仅仅在新数据块分配给这个精简置备的磁盘时才生效。如果这种情况发生了，在虚拟机概要页会显示如图 7-15 这样的报错信息。

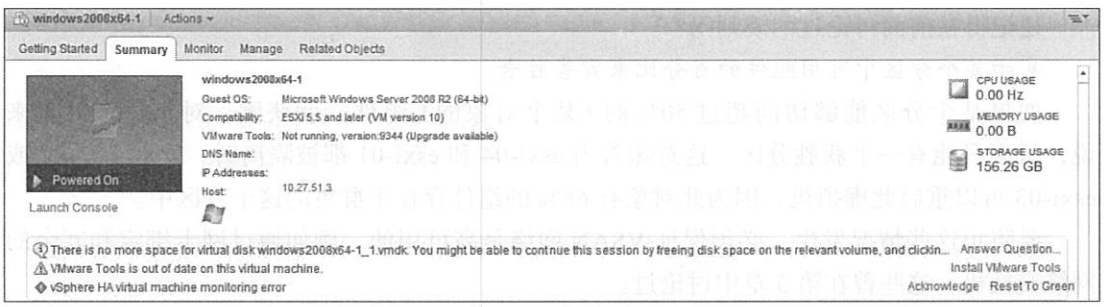


图 7-15 没有多余空间的信息

这和数据存储到达容量上限时可以观察到的 VMFS 的表现一模一样。当额外的磁盘容量添加到 VSAN 数据存储之后，"停滞的"虚拟机可能可以通过 vSphere Web 客户端来恢复运行。通过 Monitor（管理）> Virtual SAN > Physical Disks（物理磁盘）视图（如图 7-16 所示），管理员应该能看见每块磁盘都各有多少容量已经被消耗掉了。

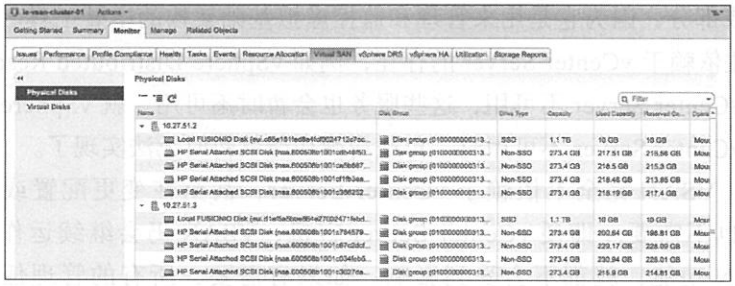


图 7-16　监控物理磁盘

7.6　精简置备的考量

默认情况下，所有部署到 VSAN 数据存储的虚拟机都是精简置备的。这样做最大的好处就是虚拟机不会占用任何未使用的空间容量。在数据中心中的虚拟机有 40% ～ 60% 的未用空间的情况并不少见。可以想象，如果虚拟机是厚置备的，不仅会平添不少成本，而且从组件放置的角度来看还会使 VSAN 的弹性减小。

当然，从运营的角度来看精简置备，如果你严重地过量分配空间而且很多虚拟机都声明新的磁盘空间，总是会填满 VSAN 数据存储空间的。这和采用 NFS 或使用虚拟机精简配置的 VMFS 的环境没什么区别。幸运的是，通过 Web 客户端用户界面，有很多地方可以检查容量，图 7-17 显示的是例子之一——Summary（概要）页。

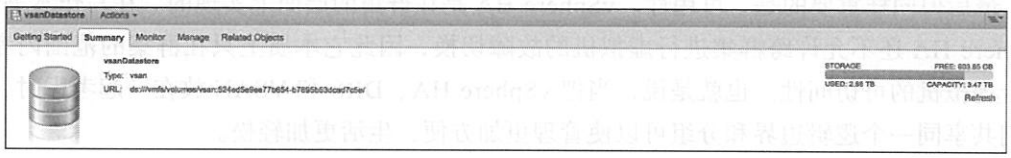


图 7-17　VSAN 数据存储的容量

当然，当到达一定的阈值时，vCenter Server 会发出警报来确保管理员已经知晓了潜在的可能发生的问题。默认情况下，会在超过 75% 的阈值时触发一个黄色三角形的感叹号的警报（严重性：警告），当到达 85% 时会发出另外一个警报（严重性：严重），如图 7-18 所示。

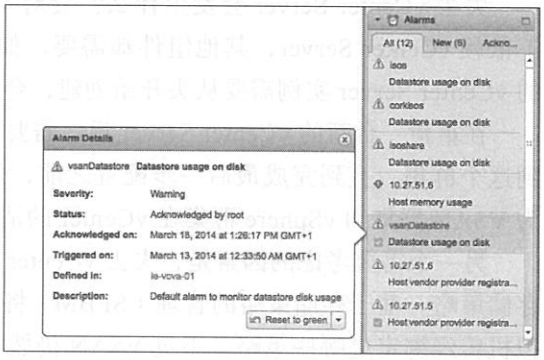


图 7-18　数据存储使用警示

7.7　vCenter 管理

vCenter Server 是大多数 vSphere 部署项

目中的重要组成部分，因为它是用来管理和监控虚拟基础架构的主要工具。过去，vSphere 中的新功能往往依赖于 vCenter Server 的存在，例如 vSphere Distributed Resource Scheduler（DRS）。如果 vCenter Server 不可用，这些服务也会暂时不可用。就 vSphere DRS 的例子来说，这意味着 vCenter Server 不可用时，负载均衡功能就暂时无法实现了。

幸运的是，VSAN 完全不依赖于 vCenter Server，甚至在变更配置或创建一个新的 VSAN 群集时也不需要。即便 vCenter Server 宕机，VSAN 仍会继续运作，因此只要虚拟机用了 VSAN 功能，就都不会受到影响。如果有需要，所有的管理任务都可以通过 ESXCLI（就此事还可用 RVC）来完成。还有疑惑？是的，这种方法是完全受 VMware 支持的。

现在你可能会问为什么 VMware 决定将 VSAN 群集和 vSphere HA 与 DRS 放在一起管理，尤其是在 VSAN 和 vCenter Server 之间没什么直接依存关系的情况下。有几个原因，让我们在探讨 vCenter Server 故障场景之前先来简单解释一下。

把 VSAN 群集和 vSphere HA 与 DRS 群集放在一起的主要原因是用户体验。现在，在配置 / 启用 VSAN 的时候，只需要在 vSphere Web 客户端界面中，在群集属性中点击一次鼠标即可完成。这样轻松的用户体验主要是靠已经成型的计算群集（由一组逻辑的 ESXi 主机构成）实现的。

这不仅可以使部署更为轻松，也简化了升级流程和其他维护任务，这些任务通常是在群集的范围内完成的。在此之上，计算资源的容量规划和规模决策都是在群集的粒度上完成的，通过群集结构的统一，可以方便进行存储规模的规划。

最后但同样重要的是：可用性。vSphere HA 是在群集的层面上实现的。在写作本书时，vSphere HA 还不允许跨群集进行虚拟机的故障切换，因此它本质上只在群集的范围内考虑每台虚拟机的可访问性。也就是说，当把 vSphere HA、DRS 和 VSAN 放在一起考虑时，让它们共享同一个逻辑边界和分组可以使管理更加方便，生活更加轻松。

7.7.1 vCenter Server 故障场景

失去 vCenter Server 会发生什么？会对 VSAN 有何影响？如何重建环境？即使 VSAN 不依赖 vCenter Server，其他组件却需要。如果，只是如果，vCenter Server 发生故障，新的 vCenter Server 实例需要从头开始创建，会对 VSAN 环境有何影响？

在重建一台新的 vCenter Server 后，需要重新定义一个启用 VSAN 的群集并将主机加回到这个群集。直到完成最后一步配置之前，你将会一直收到一个“配置错误”的警报，因为 VSAN 群集和 vSphere 群集在 vCenter 的清单（inventory）中不匹配。

另一个需要考虑的因素是，失去 vCenter Server 意味着失去管理员曾经创建过的虚拟机存储策略。基于存储策略的管理（SPBM）将不会知道之前的虚拟机存储策略，也不知道虚拟机曾经绑定过哪些策略。不过 VSAN 仍然清楚明白地知道管理员的要求并继续实施着这些策略。现在，通过 UI 是无法导出现有策略的，但是在 vSphere 5.5 中已经有一个用于虚

拟机存储策略的应用程序编程接口（API）可以做到这一点。

关于虚拟机存储策略API的重要一点是，它是作为一个vCenter Server中的独立的API端点发布的，因此不能通过普通vSphere API访问到。要使用这个API，必须连接到SPBM服务器，这需要一个认证的vCenter Server会话来实现。这个API可以用来导出和导入策略。你可以在下面的William Lam的文章中找到如何获取现有虚拟机存储策略的一个例子：http://www.virtuallyghetto.com/2013/11/restoring-vsan-vm-storage-policy.html。通过利用这个例子和公开的SPBM API，就可以开发出导入和导出虚拟机存储策略的脚本。

7.7.2 在VSAN上运行vCenter Server

一个常见的支持问题是，VMware是否支持在VSAN群集上安装用于管理VSAN的vCenter Server？问题来源于当VSAN数据存储出现故障无法访问时，其上的虚拟机包括vCenter Server也会无法运行。于是在失去了vCenter Server之后（因此诸如RVC等工具也无法使用），就无法对VSAN环境进行排错。幸运的是，VSAN完全可以通过ESXi主机上的ESXCLI命令行来进行管理。因此，开始时那个问题的答案是：是的，VMware支持用户将vCenter Server安装在VSAN上（而且这也是受支持的），但是显然，在少数的vCenter Server不在线的情况下需要对VSAN进行管理或进行问题排错，用户体验就不会那么好了。这是一个需要仔细考虑并谨慎做出的决定。

7.7.3 vCenter Server引导过程

如果可以在ESXi上运行vCenter Server，如何从头创建并开始部署呢？在典型的绿地项目[⊖]中，没有其他外部存储，VSAN必须在部署vCenter Server之前就已经投入使用。

virtuallyghetto.com的William Lam描述了一个可以实现上述目标的操作流程。要获得完整的流程和关于VSAN及自动化的更多文章，请参见William Lam的网站（virtuallyghetto.com），他友好地授权我们在本书中使用他提供的内容。为了方便读者，我们将这个在单服务器VSAN数据存储上部署并引导一台vCenter Server的过程简短地进行了总结，步骤如下。

1. 在物理主机上安装ESXi 5.5。从技术角度来说，一台主机就可以开始这个部署过程了，但是你可能希望再准备好2台额外的主机，除非你不在乎vCenter Server在遇到硬件故障的时候能否被恢复。

2. 你必须在计划置备vCenter Server的ESXi主机上修改默认VSAN存储策略，运行以下2条ESXCLI命令行来启用“强制置备”：

```
esxcli vsan policy setdefault -c vdisk -p
  "((\"hostFailuresToTolerate\" i1) (\"forceProvisioning\" i1))"
```

⊖ 绿地项目（Greenfield deployment）指的是全新的从零开始构建的项目。

```
esxcli vsan policy setdefault -c vmnamespace -p
 "((\"hostFailuresToTolerate\" i1) (\"forceProvisioning\" i1))"
```

3. 用下列 ESXCLI 命令来确认 VSAN 默认策略已经修改正确了：

```
~ # esxcli vsan policy getdefault
Policy Class  Policy Value
____  ____________________
cluster       (("hostFailuresToTolerate" i1))
vdisk         (("hostFailuresToTolerate" i1) ("forceProvisioning" i1))
vmnamespace   (("hostFailuresToTolerate" i1) ("forceProvisioning" i1))
vmswap        (("hostFailuresToTolerate" i1) ("forceProvisioning" i1))
```

4. 你必须识别出第一台 ESXi 主机上的用于 VSAN 数据存储的磁盘。用下列 ESXCLI 命令来获取信息：

```
esxcli storage core device list
```

5. 要获取某个特定设备的具体信息来分辨这到底是一块 SSD 还是常规的磁盘，可以在命令行中用 -d 参数并加上设备名：

```
esxcli storage core device list -d <disk identifier>
```

6. 识别出要使用的磁盘后，写下磁盘名，这将在下面的步骤中用到。在这个例子中，我们只有一块 SSD 和一块磁盘。

7. 在可以创建 VSAN 数据存储之前，我们首先需要创建一个 VSAN 群集。在还没有 vCenter Server 时要创建 VSAN 群集需要的参数之一是，唯一标识了 VSAN 群集的 UUID。UUID 的格式是这样的：*nnnnnnnn-nnnn-nnnn-nnnn-nnnnnnnnnnnn*，其中 *n* 是一个十六进制数字。你可以自己编造一个或者用这个在线 UUID 生成器来生成一个：http://www.uuidgenerator.net/。

8. 要创建一个 VSAN 群集，用以下 ESXCLI 命令并加上 -u 选项和前一步骤中生成的 UUID：

```
esxcli vsan cluster join -u <UUID>
```

9. 群集创建完成后，可以运行下列 ESXCLI 命令来获取 VSAN 群集的信息：

```
esxcli vsan cluster get
```

10. 接下来我们要把 ESXi 主机的磁盘添加进来以创建这个单节点的 VSAN 数据存储。要这么做，我们需要运行下列 ESXCLI 命令，并输入在前面的步骤获取的 SSD 和 HDD 的磁盘设备名称：

```
esxcli vsan storage add -d <HDD-DISK-ID> -s <SSD-DISK-ID>
```

-d 选项指明的是常规磁盘，而 -s 选项指明 SSD 盘。如果有多块磁盘，你就需要输入多次 -d 选项。你还可以用下列 ESXCLI 命令来查看已经用于 VSAN 数据存储的磁盘清单：

```
esxcli vsan storage list
```

11. 为了减少一个额外的步骤，还可以在第一台 ESXi 主机上用 ESXCLI 命令启用 VSAN 流量类型，并且也可以为其他两台主机提前做同样的操作。这个步骤不是马上必需的，因为它可以稍后在 vCenter Server 搭建起来后通过 vSphere Web 客户端来完成。你需要创建一个 VMkernel 接口或是选择一个已有的 VMkernel 接口来启用 VSAN 流量类型，ESXCLI 命令行如下：

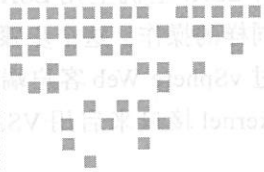
```
esxcli vsan network ipv4 add -i <VMkernel-Interface>
```

12. 此时，你已经在这个单台 ESXi 主机上配置出了一个有效的 VSAN 数据存储。你可以通过登录到 vSphere C# 客户端来进行验证，此时应该可以看见 VSAN 数据存储已经挂载到了 ESXi 主机上。现在可以部署 vCenter Server Applicance 5.5 OVA/OVF 到数据存储并启动这虚拟机了。

在部署完 vCenter Server 后，你应该尽快将剩余的主机都加入到群集中。此外，你还需要创建一个新的虚拟机存储策略，并将其和 vCenter Server 虚拟机相关联，并确保 vCenter Server 虚拟机和这个新的策略匹配。

7.8　小结

通过本章的演示可以知道，VSAN 很容易进行横向或纵向的扩展。即使配置成手工方式，添加新的主机或新的磁盘仍然只需要点击几下鼠标即可。对于那些偏好命令行的管理员来说，ESXCLI 是 vSphere Web 客户端的一个极好的替代品。对于偏好 PowerShell 的管理员来说，VMware 也已经发布了一个小工具来提供额外的命令使你可以通过 VMware PowerCLI 来管理 VSAN。可以在这里找到更多信息：https://labs.vmware.com/flings/powercli-extensions。



Chapter 8 第 8 章

互操作性

本章着眼于 VSAN 与其他 vSphere 核心技术及 VMware 产品之间的互操作性，如果存在不兼容或额外需要考虑的地方，我们将在相应的章节特别强调。我们将尽可能保证本章中讨论的组件间的互操作性内容是正确的，但是 VMware 可能会在任何时候做出变动并更新产品，因此，我们强烈建议参考官方的关于 VSAN 和相应产品的文档以及相关产品的发布说明（Release Notes）来获取最新的信息。VMware 还提供了一个在线的互操作性指南 / 矩阵的网站，这也是我们强烈推荐的，可在下面的网址找到：

http://partnerweb.vmware.com/comp_guide2/sim/interop_matrix.php

可以在 VMware 知识库文章 KB2006028 中找到 *VMware Product Compatiblity Guide*（VMware 产品兼容性指南）。

注意，现在 VMware 具有大量软件产品和功能的组合，并且还在持续地定期发布新产品和新功能。本章不可能包罗万象，只是探讨了那些在与客户和合作伙伴的会议及谈话中发现的令我们感兴趣的产品和功能。再提一次，VMware 官方文档是真相的源头，因为随着时间的推移，产品的可支持性和互操作性毫无疑问都会发生变化。我们强烈建议大家在部署 VSAN 及相关产品的时候先读一读每个产品的发布说明，以了解产品的最新变化。

8.1 vMotion

vSphere vMotion 可以实现主机之间在线迁移虚拟机而无须任何宕机时间，无论是虚拟机上的客户操作系统还是运行着的应用程序。这个功能完全支持部署在 VSAN 数据存储上的虚拟机。VSAN 上的 vMotion 用户体验和传统 SAN 或 NAS 数据存储上的完全一致，而

且群集中的所有主机都共享数据存储。如前所述，在当前发布的 VSAN 版本中没有数据本地性，换而言之，一台虚拟机的计算资源可以位于某一台 ESXi 主机上，而其数据则可以存放在 VSAN 群集中另外一台完全不同的 ESXi 主机上。这意味着，VSAN 群集中的虚拟机在主机之间的 vMotion 只需要关心虚拟机计算资源的移动，而不需要关心自身的数据迁移。管理员可以在 VSAN 群集中所有的 ESXi 主机之间迁移运行中的虚拟机，包括那些不将本地存储提供给 VSAN 数据存储使用的但是仍可访问 VSAN 数据存储的 ESXi 主机。

将 VSAN 数据存储上的虚拟机的 vMotion 和传统数据存储上的虚拟机的 vMotion 进行比较时，有一个额外的现象需要特别指出。在 VSAN 群集发生网络分区（Network Partition）时，从一台被隔离的 ESXi 主机上能够看见 VSAN 数据存储并不意味着主机需要访问到属于虚拟机的所有存储对象。当你试图 vMotion 一台虚拟机时，vMotion 会检查目的 ESXi 主机是否能访问这台虚拟机的存储对象，如果不能，vMotion 会失败，运行中的虚拟机会留在原先的 ESXi 主机上。

当然，VSAN 群集上的网络分区也扮演着 vSphere High Availibility（HA）的角色，你将在本章后面的部分了解到更多细节。

8.2 Storage vMotion

因为所有 VSAN 群集上部署的虚拟机都共享同一个分布式数据存储，因此 Stoarge vMotion 本身是不存在的。也就是说，无法将虚拟机的存储对象在群集内移来移去。VSAN 拥有其自己的算法来进行虚拟机存储对象的初始放置。

不过，因为加入到 VSAN 群集的 ESXi 主机可能也会有其他类型的存储，例如 NAS 或 SAN，你可能会问，是不是可以将虚拟机在线地从 VSAN 数据存储迁移到同台主机的 VMFS 或 NFS 数据存储上，或者反过来迁移？是的，虚拟机在不同数据存储类型之间（包括 VSAN）的在线迁移是完全受支持的。下面列出了受支持的 Storage vMotion 的可能的情况。

- 从 VMFS 迁往 VSAN
- 从 NFS 迁往 VSAN
- 从 VSAN 迁往 VMFS
- 从 VSAN 迁往 NFS

vSphere 的迁移机制（vMotion）在 vSphere 5.5 中被增强了，现在可以做到同时将虚拟机的计算部分和存储部分从一台 ESXi 主机和数据存储迁移到另一台 ESXi 主机和数据存储，而没有任何宕机时间。结果是这使得跨网络的主机间在线迁移不再需要共享存储。虚拟机还可以在不同群集之间进行迁移。vSphere 管理员也可以利用增强 vMotion 功能在不同的 VSAN 群集之间迁移虚拟机。

8.3 vSphere HA

vSphere HA 完全支持 VSAN 群集，它为群集中的虚拟机提供了额外可用性。然而为了能正确地与 VSAN 互操作，VMware 对 vSphere HA 进行了很多重大的改动。

尽管在最初的发布版本中 VSAN 支持 32 个节点，而且 VSAN 群集中每台 ESXi 主机都支持 100 台虚拟机，vSphere HA 只能在每个数据存储中保护总共 2048 台虚拟机。这意味着如果你把 VSAN 中的虚拟机数量推到 3200 台的极限值，vSphere HA 将不能保护全部这些虚拟机。在这个例子中如果你想用 vSphere HA 来保护所有 3200 台虚拟机，一种可行的解决这个极限问题的变通办法是构建多个较小的群集。

8.3.1 vSphere HA 通信网络

在非 VSAN 环境中，vSphere HA 代理的通信是通过管理网络进行的；在 VSAN 环境中，vSphere HA 代理的通信是通过 VSAN 网络进行的。背后的原因是我们希望当网络故障发生时 vSphere HA 和 VSAN 主机位于相同的分区中。这就避免了故障时 vSphere HA 和 VSAN 对网络分区的理解不同造成的潜在冲突——不同分区拥有不同的存储组件和对象的子集。这样 VSAN 群集代理和 CMMDS 就可以提供关于 VSAN 群集中分区的一致的视图，使得 VSAN 可以在不同的虚拟机的对象可访问性上做出准确的判断，进而可以在给定的分区（即具有简单多数组件的可被虚拟机访问的那个分区）上重启虚拟机。

VSAN 环境中的 vSphere HA 默认情况下继续把管理网络的默认网关用作隔离检测。我们猜测大多数 VSAN 环境中管理网络和 VSAN 网络很可能是共享同一个物理架构的（尤其是万兆以太网环境）。不过如果 VSAN 和管理网络是在不同的物理架构中，建议将默认的 vSphere HA 隔离检测地址从管理网络改到 VSAN 网络中。如前所述，默认情况下隔离检测地址是管理网络的默认网关，VMware 建议在 VSAN 中的 vSphere HA 要使用 VSAN 网络中的 IP 地址来做隔离检测。要 vSphere HA 不使用默认网关而使用 VSAN 网络中的一个 IP 地址来做隔离检测，下面的设置（在 vSphere HA 的高级设置中）必须改变：

- das.useDefaultIsolationAddress=false
- das.isolationAddress0=<VSAN 网络中的某个 IP 地址 >

然而如果 VSAN 网络中没有合适的隔离检测地址（因为 VSAN 需要 L2 网络，其中可能没有配备网关），那么可以保留默认的隔离地址在管理网络中。关于 vSphere HA 高级配置的更多细节可以在以下知识库文章中找到：http://kb.vmware.com/kb/2033250。

另一个值得注意的不同点是关于网络重新配置的。如果在 VSAN 层面对 VSAN 网络进行了变更，这无法被 vSphere HA 自动探测到。因此，vSphere 管理员必须手动发起一次 vSphere HA 群集重新配置以使这些变更可以被探测到。

8.3.2 vSphere HA 心跳数据存储

关于 VSAN 上的 vSphere HA，另一个值得注意的不同点是 VSAN 数据存储无法用作心跳数据存储。在传统的 SAN 或 NAS 数据存储环境下，当发生 vSphere HA 群集的分区情况时这些心跳对于判断虚拟机的归属起到了非常重要的作用。当 vSphere HA 部署在传统共享存储（SAN/NAS）上时，这个特性特别有用，因为它可以在分区之间进行同一层面上的沟通协调。由于 VSAN 不具有共享数据存储，VSAN 环境就不具备这个特性。vSphere HA 不将 VSAN 数据存储用于心跳，也不会允许用户将其设置为心跳数据存储。VSAN 使用网络上的群集服务（clustering service）来进行非常快速的故障探测。

然而，需要注意的是，如果 VSAN 群集中的 ESXi 主机分区也同时能访问共享存储（无论是 VMFS 还是 NFS），这些传统的数据存储会被用于 vSphere HA 心跳。

8.3.3 vSphere HA 元数据

vSphere HA 需要为群集中的每台虚拟机保存其保护元数据。在传统数据存储上，这是保存在每个数据存储的根目录下的，取名为 .vSphere-HA。在 VSAN 中，做法有点不一样。vSphere HA 保护信息不是保存在数据存储的根目录下，而是保存在虚拟机的名字空间元数据中，和虚拟机的那组常用配置文件存放在一起。

8.3.4 vSphere HA 接入控制

关于 vSphere HA 和 VSAN 之间互操作性还有另外一个值得讨论的因素。在配置 vSphere HA 时，有一个需要做出的决定就是关于接入控制的。接入控制保证了 vSphere HA 可以留出一些资源，这样在故障时能具有足够的资源可以重启虚拟机。

注意，在故障恢复的时候，VSAN 是不知道有接入控制机制存在的。VSAN 不存在这样的自动机制来留出一部分空闲资源用于保证不会发生过量分配。

如果发生了故障，VSAN 会试图利用群集中剩余节点上的所有剩余资源来开启虚拟机并保证其处于合规状态。如果资源过量分配，VSAN 群集中可能发生的多个故障会占用 VSAN 数据存储上全部的可用空间，因此当 VSAN 和 vSphere HA 一起使用的时候，必须要提前做好谨慎的容量规划。

在规划和设计一个 VSAN 环境的时候，建议将“重建容量”考虑进去。在第 9 章中将学到如何实现这个目标。为简单起见，建议将 VSAN（手工）接入控制的设定和所选的 vSphere HA 接入控制设置保持一致。

8.3.5 vSphere HA 推荐设置

当在启用了 vSphere HA 的 VSAN 环境中发生了主机隔离事件时，vSphere HA 会实施配置的隔离响应举措。在 vSphere HA 配置中对隔离事件可以选择以下 3 种不同类型的响应：

- 保持打开电源
- 请关闭电源，然后进行故障切换
- 请关闭，然后进行故障切换[⊖]

推荐的配置是在主机隔离事件发生时，让 vSphere HA 自动关闭主机上的虚拟机电源，因此，“隔离响应”应该设置成“请关闭电源，然后进行故障切换”，而不是默认的“保持打开电源”。

注意，那个“请关闭电源，然后进行故障切换”动作类似于从物理主机上拔掉电源线。虚拟机的进程事实上是突然中断的——这不是一次干净的关机。不过，在隔离事件发生时，VSAN 不太可能在隔离的主机上进行磁盘写入，因此关闭电源是推荐做法。如果 ESXi 主机被分区了，虚拟机也不太可能访问到存储对象的单独多数的组件。

8.3.6 受 vSphere HA 保护的 VSAN 和非 VSAN 虚拟机

对基于 VSAN 的虚拟机，是否应该重启虚拟机的信息是保存在虚拟机的主页名字空间对象中的；对非基于 VSAN 的虚拟机（也就是位于 SAN 或 NAS 数据存储上的虚拟机），这个信息是保存在虚拟机主数据存储的受保护列表文件中的。

无论是对 VSAN 虚拟机还是非 VSAN 虚拟机，vSphere HA 主控主机更新和访问其重启信息都是类似的。

在保护一台虚拟机的时候，如果虚拟机是非 VSAN 虚拟机，在虚拟机主数据存储中的保护列表文件会被更新。

如果受保护的是一台 VSAN 虚拟机，则会在虚拟机主页名字空间对象元数据中插入一个保护关键字。

在 vSphere HA 主控主机被选举出来后，主控主机需要决定哪些虚拟机要受保护。如果同时存在 VSAN 数据存储和传统的 SAN/NAS 数据存储，vSphere HA 主控主机会不仅读取传统数据存储上的保护列表文件，也会从虚拟机主页名字空间对象元数据中获取 vSphere HA 保护关键字。

8.4 DRS

VSAN 完全支持 DRS 来进行基于 CPU 和内存资源的虚拟机初始放置。DRS 也可以置于全自动模式，因此在发生 CPU 或内存资源不平衡的时候，虚拟机的计算资源可以通过 vMotion 在线迁移到群集中的另一台主机上。我们已经在 vMotion 一节介绍过这些，不过因为在 VSAN 上的虚拟机的计算资源和存储资源是完全无关的（例如，虚拟机的计算资源位

⊖ 这里完全参考了 vSphere Web 客户端中文版中的译文，更准确的意思是：请关闭客户机操作系统，然后进行故障切换。——译者注。

于 ESXi 主机 1，而其存储对象的第一个副本可能在 ESXi 主机 2 上，而且存储对象的第二个副本可能在 ESXi 主机 3 上），vMotion 操作（由 DRS 发起）可以无缝地在 VSAN 群集上的 ESXi 主机之间移动虚拟机。

8.5 Storage DRS

Storage DRS（SDRS）用于在多个共享式数据存储之间持续进行虚拟机的负载均衡或是初始放置。因为 VSAN 只提供单一的分布式数据存储，从这个角度来看，初始放置和负载均衡与 SDRS 无关。性能和可用性都基于虚拟机存储策略中的设置项，并且虚拟机存储对象和组件初始被 VSAN 放置在什么地方也取决于策略中的这些要求。

SDRS 可以继续在 VSAN 群集中用于 VMFS 和 NFS 数据存储，如果这些存储类型存在的话。不过，你可以把 VSAN 数据存储包含在一个数据存储群集中。

8.6 Storage I/O Control

VSAN 不支持 Storage I/O Control（SIOC），原因和 SDRS 类似。通过在虚拟机存储策略中设置的要求，VSAN 会决定放置这些对象 / 组件的最佳方法以满足这些要求。当瓶颈出现时，SIOC 会抑制同一个共享式的 SAN 或 NAS 数据存储的主机上的队列深度，所以这个特性和“分布式的”VSAN 数据存储无关，因为 VSAN 数据存储只使用本地存储。在 VSAN 上 SIOC 会被自动禁用。SIOC 是用来发现和管理 VMware 不能控制的存储上的性能问题的，而 VSAN 具有内建的决策机制，可以用来均衡虚拟机存储对象和组件。

如果导航到 VSAN 数据存储，选择 Manage（管理）页，在 General（常规）底下的 Datastore Capabilities（数据存储功能）的地方会看见 Storage I/O Control 显示为 Not supported（不受支持），如图 8-1 所示。

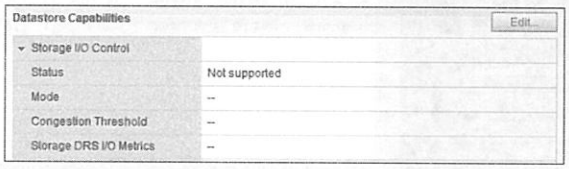


图 8-1 Storage I/O Control 不受 VSAN 支持

8.7 分布式电源管理

分布式电源管理（Distributed Power Management，DPM）是一个在资源利用率较低的时候关闭 ESXi 主机并在需要更多资源的时候重新开启这些主机的工具。它和 VSAN 不兼容，

也被自动禁用了。禁用 DPM 的原因是一台 ESXi 主机尽管其上可能没有运行任何一台虚拟机，仍然有可能包含某虚拟机的存储对象的组件（例如 RAID-0 条带或 RAID-1 镜像）。出于这个原因，不希望将 ESXi 主机置于备用 / 关机状态。在启用了 VSAN 的群集上不允许开启 DPM，如果你这么做了，会显示如图 8-2 那样的报错信息。

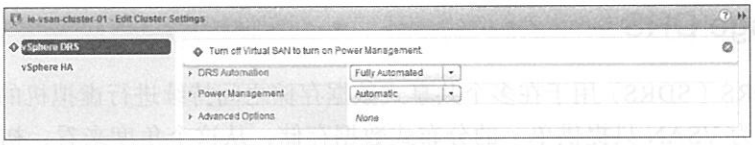


图 8-2　DPM 不受 VSAN 支持

8.8　VMware Data Protection

vSphere Data Protection（VDP）和 VDP Advanced（VDPA）版本 5.5.6（2014 年 3 月 11 日发布）都受 VSAN 支持。更早的 VDP/VDPA 版本则不受 VSAN 支持。VDP 虚拟设备可以运行在 VSAN 数据存储上，并可以备份和恢复 VSAN 数据存储上的虚拟机。事实上，用 VSAN 来承载备份和恢复的基础架构是 VSAN 产品团队给 VSAN 定义的功用之一。它同时具备全套的恢复操作，例如 Restore to Original Location（恢复到原位置）和 Create New（创建新副本）。

在安装过程中，管理员会被提示为 3 个 256GB 的虚拟磁盘选择存储位置，这 3 个磁盘是保存备份的地方。此时可以选择 VSAN 数据存储，如图 8-3 所示。

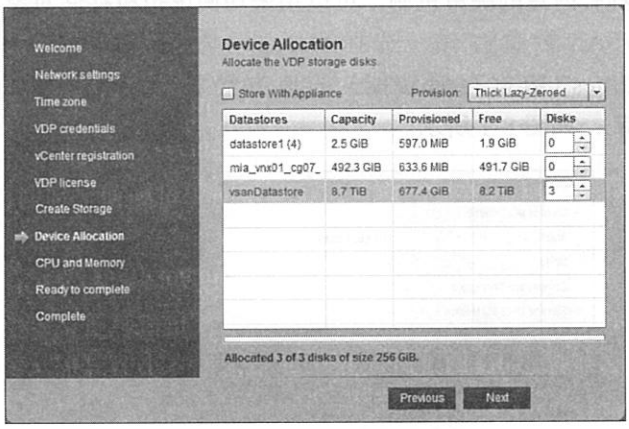


图 8-3　VDP 可以使用 VSAN 作为存储

虚拟机磁盘对象（VMDK）没有部署虚拟机存储策略，这意味着它们将继承默认策略，即允许的故障数为 1。计划在 VSAN 数据存储上安装 VDP 或 VDPA 的管理员需要注意，尽管 VDP 要求的是 3 个 256GB，真正需要的空间将会是这个数量的两倍，这是因为允许

的故障数为 1 的策略（如果实施的策略中 FTT 值更大的话，空间会需要更多）。这是因为默认策略不会使用对象空间预留设置，而是在部署磁盘的时候用传统的精简置备或厚置备的方法。如果（在创建虚拟机时）选择了默认策略 None 的话，部署的磁盘会采用默认的厚格式。

8.8.1 使用 VDP 从 VSAN 数据存储备份虚拟机

在 VSAN 上的虚拟机和传统数据存储上的虚拟机备份的方式完全一样，没有需要额外考虑的地方。注意，虚拟机存储策略的信息是不随虚拟机备份的，因此，当进行恢复的时候，也没有虚拟机存储策略信息随着虚拟机被恢复出来，恢复的仅仅只是虚拟机数据而已。恢复后用户可以选择某个策略来和恢复出来的虚拟机进行关联。

8.8.2 使用 VDP 将虚拟机恢复到 VSAN 数据存储

恢复一台从 VSAN 数据存储上备份过的虚拟机和恢复一台传统数据存储上的虚拟机完全一样。通过选择“Restore to original location”（恢复到原始位置）或是“create new”（创建新副本），虚拟机可以被恢复到原来的位置或是新的位置。当选择“Restore to original location”时，恢复出来的虚拟机会保留其原来的虚拟机存储策略，如图 8-4 所示。因此，不需要在恢复之后重新将虚拟机存储策略应用到虚拟机上。

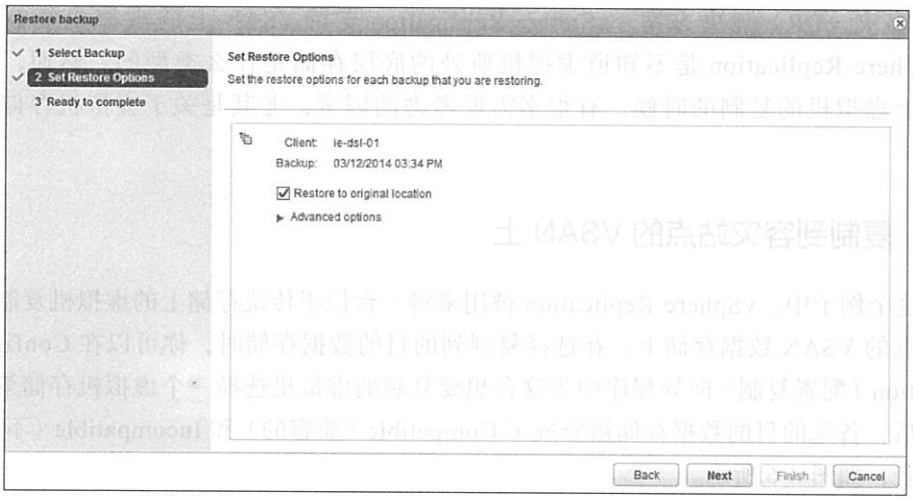


图 8-4 VDP 可以恢复虚拟机到原来的位置

当选择“create new”的恢复方法时，虚拟机存储策略不会随虚拟机恢复出来，因为这个信息没有随虚拟机备份起来。虚拟机恢复后关联的是默认策略（也就是允许的故障数为 1）。因此被恢复到 VSAN 数据存储上的虚拟机的存储策略需要在恢复完成后重新应用到此虚拟机上。注意，如有必要，Advanced Options（高级选项）允许你选择另外一个恢复的目

的数据存储，如图 8-5 所示。

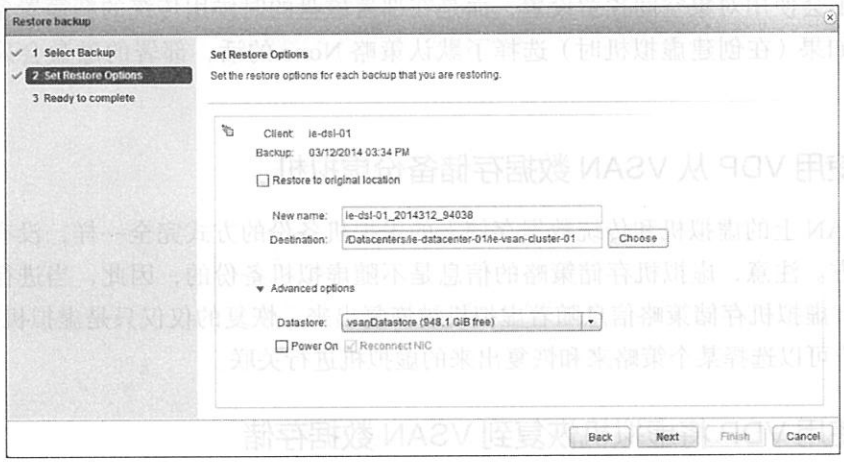


图 8-5 VDP 可以恢复到另一个位置

8.9 vSphere Replication

VMware 完全支持 vSphere Replication 版本 5.5.1（2014 年 3 月 11 日发布）用于 VSAN 来提供容灾（DR）解决方案。vSphere Replication 支持 VSAN 上的虚拟机复制。事实上，vSphere Replication 是不知道虚拟机所处的底层存储是什么类型的。然而，当进行 VSAN 上虚拟机的复制的时候，有很多需要考虑的因素，尤其是关于虚拟机存储策略的方面。

8.9.1 复制到容灾站点的 VSAN 上

在这个例子中，vSphere Replication 被用来将一台位于传统存储上的虚拟机复制到一个容灾站点的 VSAN 数据存储上。在选择复制到的目的数据存储时，你可以在 Configuration Replication（配置复制）向导程序中为这台想要复制的虚拟机选择一个虚拟机存储策略。选择策略后，备选的目的数据存储被分成了 Compatible（兼容的）和 Incompatible（不兼容的）两个列表，如图 8-6 所示。

8.9.2 恢复虚拟机

恢复虚拟机后，在 VSAN 数据存储上将它启动起来，虚拟机及其虚拟磁盘会和在“Configure Replication”（配置复制）步骤中指定的虚拟机存储策略关联。不过，在恢复的最初阶段，VMDK（的合规状态）被显示为 out of date（已过期），如图 8-7 所示。

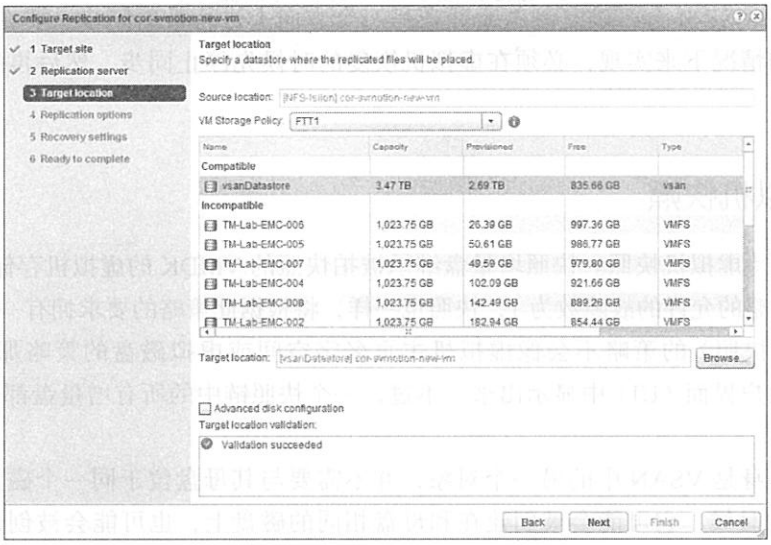


图 8-6　VSAN 和 vSphere Replication 的互操作性

图 8-7　硬盘合规性状态显示为 out of date（已过期）

这是因为虚拟机起初配置的是默认策略（允许的故障数为 1），要让虚拟机合规并重新配置策略要求，管理员必须点击那个图标来“给所选的虚拟机重新应用虚拟机存储策略”。在此时，根据所需配置的策略组件，虚拟机硬盘可能会被报为不合规。不过，当配置完成的时候，虚拟机会被认为是合规的。

没有一种方法可以通过 vSphere Replication 对被恢复的虚拟机自动进行重新保护（或自动对复制动作进行回退），这不是 VSAN 的限制，而是 vSphere Replication 本身的运作方式使然。要做到自动化，需要用 Site Recovery Manager（SRM）来配合 vSphere Replication。

在这个例子中，SRM 会打破复制关系并在另一个方向上进行重新同步。要在只用 vSphere Replication 的情况下来实现，必须在虚拟机恢复的时候先停止同步，然后再配置一个相反方向的复制。

8.10 虚拟机快照

VSAN 支持虚拟机快照。快照增量盘继承被拍快照的 VMDK 的虚拟机存储策略。因此，如果虚拟机策略的允许的故障数为 1，快照也一样，将根据此策略的要求拥有一个副本。

增量盘（快照）的策略不会像虚拟机主页名字空间或虚拟磁盘的策略那样在 vSphere Web 客户端用户界面（UI）中显示出来，不过，一个快照链中的所有增量盘都会继承其母盘同样的策略。

快照盘本身是 VSAN 中的另一个对象，并不需要与其母盘位于同一个磁盘上。当快照被创建出来的时候，它可能会被创建在和母盘相同的磁盘上，也可能会被创建在另一块不同的磁盘上，这取决于诸如容量等因素。总之，就算快照被创建在和母盘同一块磁盘上，只要这块磁盘容量快满的时候，VSAN 就可能会将某些组件无缝地移动到其他磁盘上。移动一个组件通常不会产生能被察觉到的性能影响，因为 VSAN 会通过创建新的组件并在后台同步数据的方法来移动组件，而此时虚拟机的 I/O 仍主要由闪存设备 /SSD 来提供。

VSAN 无法对整个 VSAN 数据存储拍取快照。

8.11 vCloud Director

VSAN 已经被测试通过可以和 vCloud Director 5.5.1 一同工作，并可以用来取代传统的 SAN 或 NAS 数据存储。VSAN 的虚拟机存储策略也能被 vCloud Director 识别，并表现得也和常规策略与数据存储类似。图 8-8 显示了 VMware 兼容性指南中 VSAN 和 vCloud Director 的互操作性矩阵。

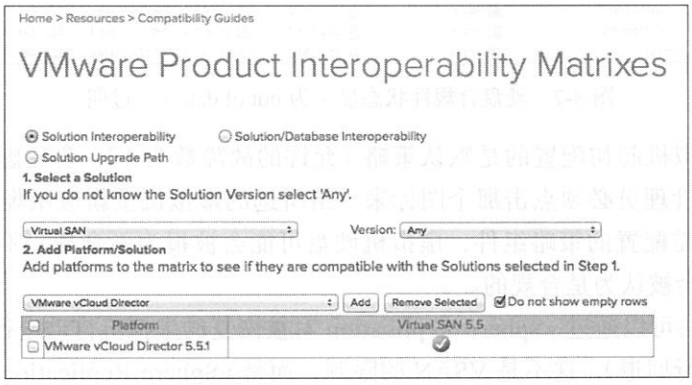


图 8-8 VSAN 和 vCloud Director 的互操作性矩阵

8.12 VMware Horizon View

VSAN 的主要使用案例之一就是 VMware Horizon View——这是 VMware 的虚拟桌面产品。部署一个虚拟桌面基础架构（virtual desktop infrastructure，VDI）的主要障碍之一就是，能满足虚拟桌面性能要求的存储的成本居高不下。在很多案例中都因为合适的存储成本太高或者现有的存储性能太差，因而虚拟桌面项目被取消或是暂时叫停了。

VMware Horizon View 支持把 VSAN 用作其存储平台。VSAN 天生的横向扩展能力加上 SSD 层面（所有 I/O 都由闪存提供）提供的高性能使 VSAN 成了 VMware Horizon View 的极佳伙伴。

当配合其他 vSphere 功能，例如 vSphere Storage Accelerator（也被称为 Content Based Read Cache，CBRC）时，通过 VMware Horizon View 部署在 VSAN 上的虚拟桌面，可以给对 VDI 感兴趣的客户提供非常棒的产品组合。

8.12.1 用于 Horizon View 的 VSAN 支持

VMware 在 2014 年 3 月 11 日发布了 VMware Horizon View 版本 5.3.1 来支持 VSAN。除此之外，这个版本的 Horizon View 没有其他新功能。VSAN 数据存储既可以用于链接克隆（linked-clone）的桌面池，也可以用于完全克隆（full-clone）的桌面池。

8.12.2 用于 VMware View 的虚拟机存储策略

对于 Horizon View 5.3.1，虚拟桌面关联的磁盘使用默认的 VSAN 策略部署。默认策略不仅用于链接克隆桌面池也用于完全克隆桌面池。用于 Horizon View 部署在 VSAN 数据存储上的全部虚拟桌面的默认策略如下：

- 每个磁盘对象的带数：1
- 允许的故障数：1
- 对象空间预留：0%
- 闪存读取缓存预留：0%

在前面的章节中，我们曾经建议不要修改默认策略。我们说过，唯一可能需要更改默认策略的情况是要将 vCenter 部署到初始只有一个节点的 VSAN 群集上时。现在我们将讨论另外一个可能需要编辑默认策略的理由：如果客户想要为通过 Horizon View 5.3.1 部署的虚拟桌面的磁盘配置一个不同的策略，唯一的方法是修改默认策略。这只能通过 vSphere 命令行接口（ESXCLI）来实现，详情请参考 VMware 知识库文章 KB2073795：http://kb.vmware.com/kb/2073795。

我们希望在 VMware Horizon View 将来的发布版本中能提供新的方法，可以为每个虚拟桌面存储对象创建自定义的可配置策略，不过，目前的 Horizon Veiw 5.3.1 发布版中还无法实现。

8.12.3 Horizon View 的配置

在 VMware View 中设置桌面池的时候，方法和设置其他池没什么区别，唯一的不同就是在选择数据存储的时候要选择 VSAN。这里必须为桌面池的链接克隆选择一个数据存储，如图 8-9 所示，我们已经选择了 VSAN 数据存储。

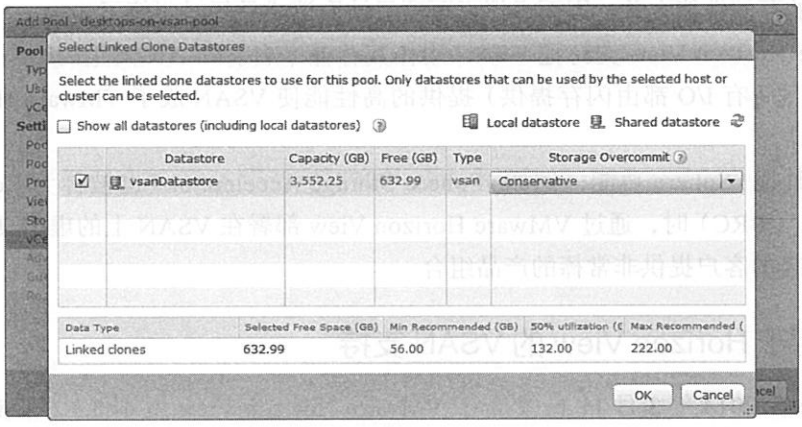


图 8-9 为链接克隆选用 vsanDatastore

桌面池创建完成后，现在可以回到 vSphere Web 客户端中来检查已经部署的桌面了。

首先应该要指出的是，基于链接克隆的地址池使用虚拟机快照来为链接克隆桌面创建副本对象。如果原始的虚拟机（及其增量快照）已经配置了一个虚拟机存储策略，那么副本同样也会关联到这个策略。然而，因为在 Horizon View 5.3.1 版本中是无法直接对虚拟机存储策略进行管理的，建议的做法是对所有对象都使用默认策略，包括对将要用于虚拟桌面镜像的虚拟机。图 8-10 显示的是一个副本对象的虚拟机存储策略，其中并没有使用默认的 None，而是关联了一个叫做 FTT1 的策略。

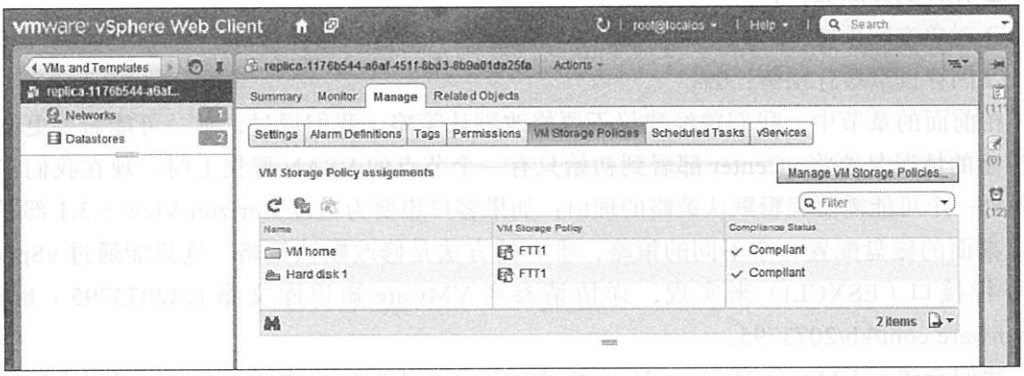


图 8-10 副本从原始虚拟机继承虚拟机存储策略

现在，让我们把注意力转回到桌面。每个桌面都有 4 个硬盘，每个硬盘都使用默认策

略。这4个硬盘如下：

1. 操作系统盘，从副本克隆。

2. 用户数据盘，或persistent disk（简写为UDD），用来存放Windows用户配置文件，这样它们就不会受View Composer的操作影响（例如更新、重新编译和重新均衡）。

3. Internal disk（内部磁盘），存储了计算机账户、密码来保证桌面刷新时和域之间的可连接性。此外，Quickprep和Sysprep的配置是保存在这个磁盘上的。

4. System Disposable Disk（系统可丢弃磁盘，简写为SDD），存放了Windows页面文件和临时文件（是非持续性磁盘，会在用户会话结束时被自动删除）。

在vSphere Web客户端中导航到虚拟机存储策略，可以查看每个桌面的虚拟机存储策略，如图8-11所示。

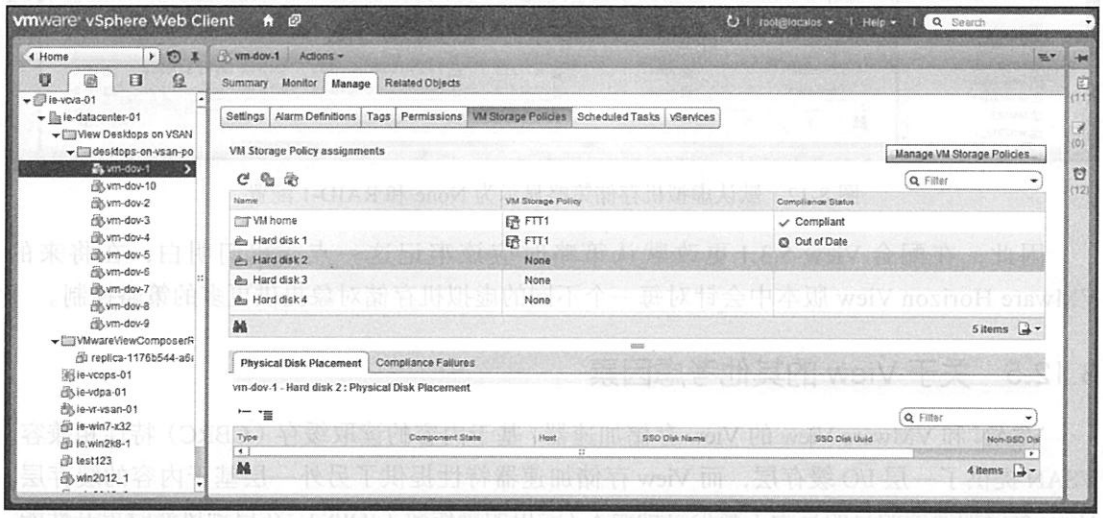


图8-11 桌面硬盘使用的是默认虚拟机存储策略

注意，虚拟机存储策略不会明示默认策略，只是简单地标注为None。然而，如果你查看任何一个磁盘，就会发现允许的故障数属性显示为RAID-1配置。如图8-12所示的就是Hard Disk 4的情况。

8.12.4 更改默认策略

大家已经了解了对于运行在VSAN上的View 5.3.1，一切都是默认策略在起作用。基础虚拟机及其相关联的快照应该部署在VSAN数据存储上并应用默认策略。副本在创建的时候将继承这个默认策略，继而为这个桌面创建的一组磁盘（操作系统盘、内部磁盘、persistent disk、disposable disk、checkpoint disk）都会继承相同的默认策略。

更改默认策略的唯一方法是使用ESXCLI命令行。不过再强调一次，你应该考虑这样的事实——更改默认策略会影响所有的对象，例如设置了10%的闪存读取缓存预留，你就

会对所有的磁盘对象都保留 10% 的读取缓存，包括内部磁盘（只有一些账号信息和 Sysprep 信息）。在这个例子中，你最好让缓存平均地被所有对象所共享。

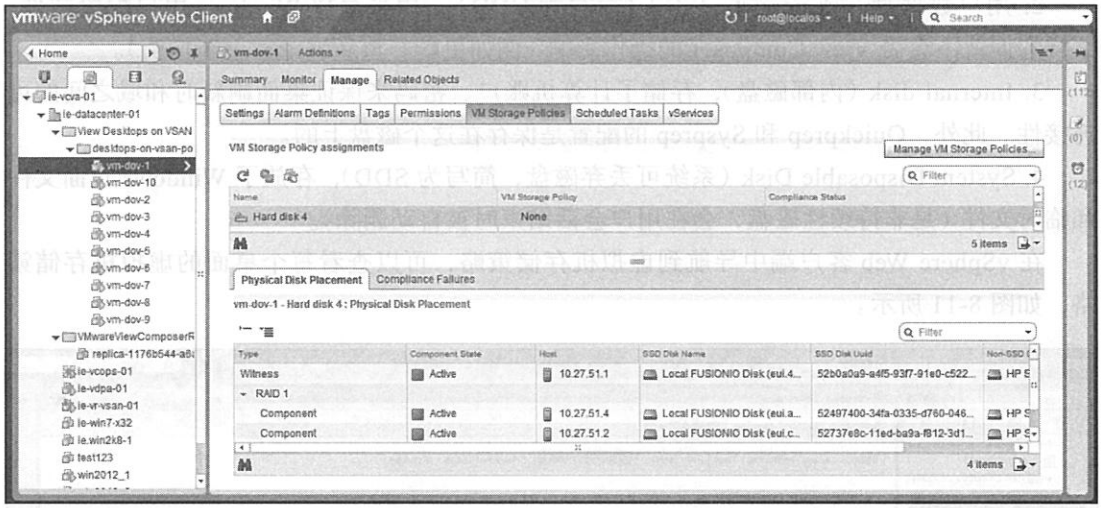


图 8-12　默认虚拟机存储策略显示为 None 和 RAID-1 配置

因此，在配合 View 5.3.1 更改默认策略时应该牢记这一点。我们明白，在将来的 VMware Horizon View 版本中会针对每一个不同的虚拟机存储对象提供更多的策略控制。

8.12.5　关于 View 的其他考虑因素

VSAN 和 VMware View 的 View 存储加速器 / 基于内容的读取缓存（CBRC）特性相兼容。VSAN 提供了一层 I/O 缓存层，而 View 存储加速器特性提供了另外一层基于内容的缓存层。View 存储加速器的目的是为了减少每秒输入 / 输出的操作数（IOPS），在启动风暴时提升性能。

另外一个考虑因素是 Horizon View 和 VSAN 在一起时消除了对由 Horizon View 提供的副本分层（Replica tiering）特性的需求。VSAN 通过 SSD 层提供了读取缓存，消除了 View 对副本分层功能的需求。事实上，当 VSAN 数据存储是虚拟桌面部署的目的数据存储时，View 5.3.1 不再支持副本分层。

最后一个考虑因素是 SE Sparse 磁盘格式不再被 VSAN 数据存储支持。因此，只有完全克隆或使用 vmfsSparse 格式（也称为 redo log 格式）的链接克隆部署的桌面才被支持。

8.13　vCenter Operations

vCenter Operations 版本 5.8（2013 年 12 月 10 日发布）支持 VSAN 并可以用来检查某些 VSAN 数据存储的特性。这个版本的 vCenter Operations 会把 VSAN 数据存储和其他类型的传统数据存储（例如 SAN 或 NFS 数据存储）一样看待。vCenter Operations 的一个很棒

的特性是其数据存储关系视图。这里你可以看见用到 VSAN 数据存储的所有 ESXi 主机和所有虚拟机，如图 8-13 所示。

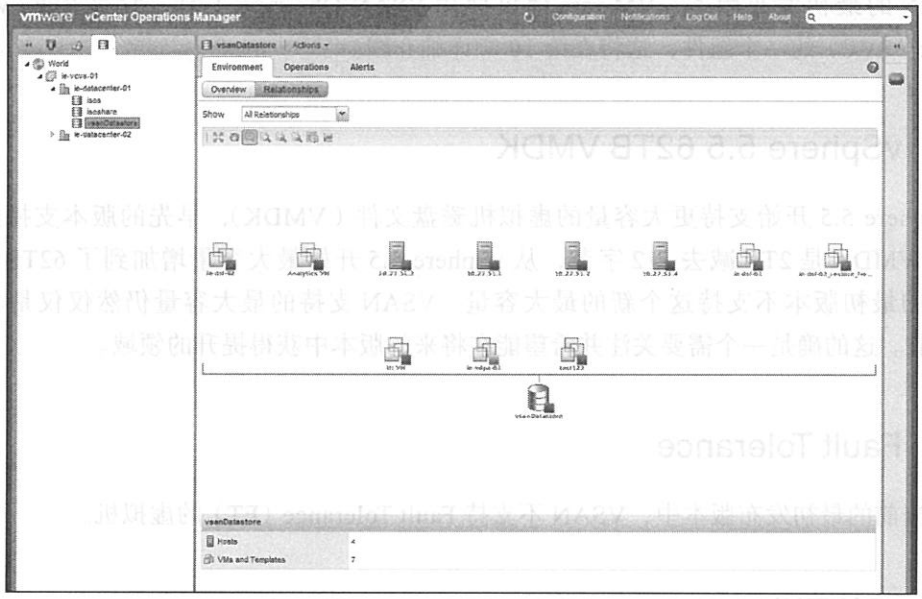


图 8-13　vCenter Operations 5.8 和 VSAN 数据存储关系视图

你或许会希望看见所有的统计信息，但是，事实并非如此。与常规的数据存储不同，VSAN 数据存储中并非所有的统计信息都可见。在 vCenter Operations 版本 5.8 中查看 VSAN 数据存储，报告中只含有容量信息，而没有磁盘 I/O 或工作负载的图表显示出来，如图 8-14 所示。这是因为在 VSAN 的最初发布版本中，这些统计信息没有提交给 vCenter 显示出来，因此也无法被 vCenter Operations Manager 获取。

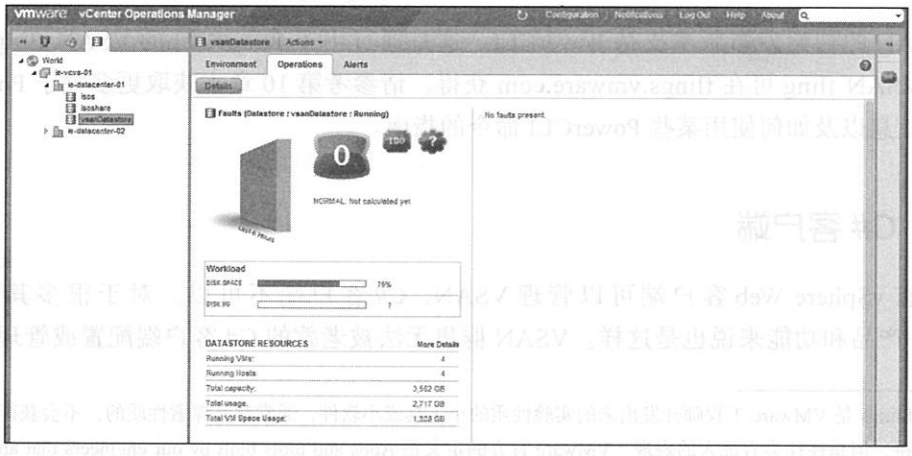


图 8-14　vCenter Operations Manager 5.8 只汇报部分 VSAN 数据存储信息

已经在项目计划中的下一个发布版本的 vCenter Operations Manager 将提供对 VSAN 的完整支持，将能够理解所有 VSAN 对象、特殊的统计信息、分布式的特性、内容等。对于 VSAN 的最初发布版本，VMware 建议使用 VSAN Observer 工具，它是 Ruby vSphere Console（RVC）的一个组件，这将在第 10 章详细介绍。

8.14 vSphere 5.5 62TB VMDK

vSphere 5.5 开始支持更大容量的虚拟机磁盘文件（VMDK），早先的版本支持的最大容量的 VMDK 是 2TB 减去 512 字节，从 vSphere 5.5 开始最大容量增加到了 62TB。不过 VSAN 的最初版本不支持这个新的最大容量。VSAN 支持的最大容量仍然仅仅是 2TB 减 512 字节。这的确是一个需要关注并希望能在将来的版本中获得提升的领域。

8.15 Fault Tolerance

在当前的最初发布版本中，VSAN 不支持 Fault Tolerance（FT）的虚拟机。

8.16 延伸群集 /vSphere Metro Storage Cluster

在 VSAN 的最初发布版本中不支持延伸群集（Stretched Cluster）或 vSphere Metro Storage Cluster（vMSC）。我们理解对这个功能有很多需求，也相信 VMware 团队正在密切关注这个功能，希望在下一个版本中这个功能可以获得支持。

8.17 PowerCLI

VSAN 的初始发布版本支持 PowerCLI。这是通过用于 PowerCLI 的 VSAN fling[⊖]来实现的，VSAN fling 可在 flings.vmware.com 获得。请参考第 10 章来获取更多关于 PowerCLI 支持的信息以及如何使用某些 PowerCLI 命令的指南。

8.18 C# 客户端

只有 vSphere Web 客户端可以管理 VSAN，C# 客户端不可以。对于很多其他新的 vSphere 产品和功能来说也是这样。VSAN 群集无法被老款的 C# 客户端配置或管理，它只

⊖ Flings 是 VMware 工程师开发出来的实验性质的小工具或小软件，通常都是探索性质的，不会获得正式的支持，但是往往会有惊人的表现。VMware 官方的定义是 Apps and tools built by our engineers that are intended to be played with or explored. ——译者注。

能通过 Web 客户端来成功配置和管理。VMware 不支持对 VSAN 功能使用 C# 客户端。

8.19 vCloud Automation Service

vCloud Automation Service（vCAC）为 vSphere 基础架构提供了集中的置备和管理。从 VSAN 集成的角度来看，vCAC 6.0.1 可以利用 VSAN 数据存储，但是所有通过 vCAC 部署的虚拟机都将使用默认虚拟机存储策略。要更改虚拟机存储策略只能在 vCAC 之外实现。图 8-15 显示了 VSAN 和 vCAC 之间的产品互操作性。

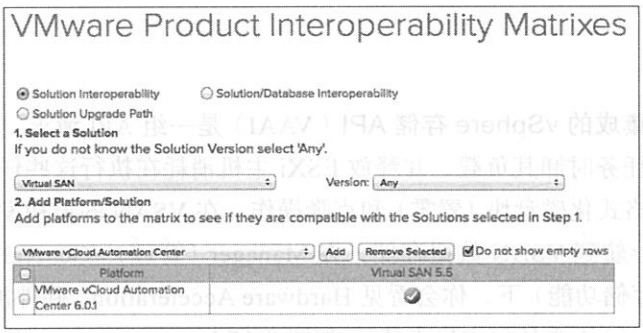


图 8-15 VSAN 和 vCloud Automation Center 的互操作性

关于如何将 vCAC 和 VSAN 集成的讨论超出了本书讨论的范畴，不过，我们在 VMware 的同事 Jad El-Zein，写过一篇信息含量非常丰富的博客文章，介绍了对于 vCAC 版本 6.0 和 VSAN 上的虚拟机存储策略哪些集成是可以实现的。欲知详情请访问博客文章 http://www.virtualjad.com/2014/03/using-vsan-storage-policies-in-vcloud.html。

8.20 主机配置文件

VSAN 支持主机配置文件（Host Profiles）特性。通过主机配置文件，我们可以先配置好要加入 VSAN 群集的第一台主机，保存下它的主机配置文件，并利用它的配置应用到要加入 VSAN 群集的其他 ESXi 主机上。如果你计划创建一个 8 节点、16 节点甚至 32 节点的群集，这将非常有用。图 8-16 显示的是用于 VSAN 群集上的主机的主机配置文件。

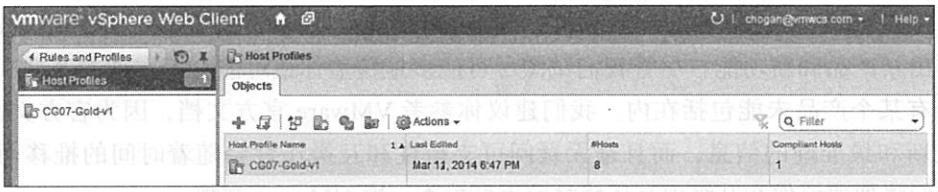


图 8-16 VSAN 和主机配置文件的互操作性

8.21 Auto-Deploy

Auto-deploy 是 ESXi 主机自动化部署的一种机制，它扩展并利用了主机配置文件。当前版本的 VSAN 不支持 Auto-Deploy。

8.22 RDM

当前版本的 VSAN 不支持 RDM（Raw Device Mappings，裸设备映射）。

8.23 VAAI

用于存储阵列集成的 vSphere 存储 API（VAAI）是一组 API 请求，用来在对存储阵列进行某些常规存储任务时卸其负载，并释放 ESXi 主机消耗在执行这些任务上的资源。典型的负载卸载任务是格式化磁盘块（置零）和克隆操作。在 VSAN 的最初发行版本中是不支持 VAAI 的。如果你导航到 VSAN 数据存储，在 Manager（管理）、General（常规）、Datastore Capabilities（数据存储功能）下，你会看见 Hardware Acceleration（硬件加速）被标注为 Not supported on any host（不受任何主机支持），如图 8-17 所示。

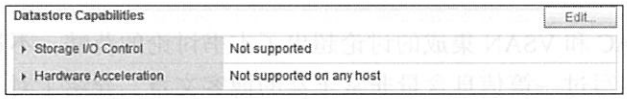


图 8-17 VSAN 不支持 VAAI

8.24 微软群集服务

VSAN 不支持微软群集服务（MSCS），主要的原因是因为它不支持物理模式的 RDM，而这是在 ESXi 主机的虚拟机上实施 MSCS 节点的必要条件。

8.25 小结

如前所述，VMware 拥有大量软件产品组合和各种各样的功能特性，并且还在定期地不断推出新产品和新功能。尽管我们试图尽可能多地覆盖你最可能会感兴趣的产品和功能，还是会有某个产品未能包括在内。我们建议你参考 VMware 官方文档，因为官方文档中包含了最新和最准确的信息，而且毫无疑问可支持性和互操作性会随着时间的推移而变化，因此我们强烈建议你在升级现有环境之前重新检查一次 VMware 文档。

第 9 章 Chapter 9

设计 VSAN 群集

本章将手把手地带着你遍历一次设计完美的 VSAN 群集的所有步骤。我们将利用本书各章中提到的知识和技巧来确保设计出的 VSAN 群集能满足你的技术要求和业务需求。在进行各种练习之前，我们想要强调的是在 VMware 兼容性指南（VMware Compatibility Guide，VCG）中包含了一个预定义配置的列表，叫做 VSAN Ready Nodes。如果你对自行选择每个硬件组件来搭建自己的服务器不感兴趣，清单中列出的硬件厂商（在写作本书时包括思科、IBM、Supermicros 和 DELL[⊖]）提供的 VSAN Ready Node 配置是非常棒的选择。

在开始下面的 2 个设计练习之前，我们想先讨论一下 VSAN 1.0 版的几个限制。

9.1 容量限制

在设计一个 VSAN 1.0 环境的时候，有几个限制必须要重视。其中一些很简单直接，另外一些则不那么明显。简单概括如下：

- ❑ 每个群集最多 32 台主机
- ❑ 每台主机最多 100 个虚拟机
- ❑ 每个群集最多 3 200 台虚拟机
- ❑ VSAN 数据存储上受 vSphere HA 保护的虚拟机数量最多 2 048 台
- ❑ 最多 5 个磁盘组
- ❑ 每个磁盘组最多 7 块磁盘

⊖ 在翻译本书时，最新版本的 VSAN Ready Nodes 列表中还包含 HP 的几个服务器型号，请记得一定要参考 VMware 官方网站上最新版本的列表。——译者注

- 每个磁盘组最多 1 个闪存设备
- 每台主机最多 3 000 个组件

如前所述，其中大多数很容易理解，其中有 3 点需要进一步解释：

- VSAN 数据存储上受 vSphere HA 保护的虚拟机的最大数量
- 群集中主机的最大数量
- 组件的最大数量

VSAN 数据存储上受 vSphere HA 保护的虚拟机的最大数量是 2 048。这不仅仅是 VSAN 群集的限制，所有基于 VMFS 和 NFS 的数据存储都是如此。注意，因为对于开启新的虚拟机并没有强制性的阻拦限制，当在 VSAN 数据存储上超过 2 048 台的限制之后仍然可以开启新的虚拟机，只是这些虚拟机将不会受 vSphere HA 保护。

这个限制的原因在于所谓的 vSphere HA “开机列表”以及由此带来的 HA 内在的运作方式。这个列表文件会跟踪所有虚拟机及其开启状态，它具有 2 048 个空位来登记虚拟机。当所有空位都被占据时，额外开启的虚拟机就无法被登记在案，因此无法被 vSphere HA 保护。

第 2 件值得注意的事情是，若要把群集中的主机数从 16 台提升到 32 台，需要到每台 ESXi 主机的高级设置中去进行额外的设置。扩展到 32 台主机的这个选项默认情况下是不启用的，因为启用后 VSAN 会要求每台主机额外多使用 150MB 内存。注意，这个选项重启后才能生效。要想获取更多信息，请参考 VMware 知识库文章 KB2073930（http://kb.vmware.com/kb/2073930）。简而言之，你需要在每台主机上通过用户界面（UI）中的高级设置或通过下面的命令行来进行配置：

```
esxcli system settings advanced list -o /CMMDS/goto11
```

第 3 个值得讨论的限制是组件的最大数量。VSAN 1.0 中组件的最大数量是每台主机 3 000 个。不同类型的对象在 VSAN 数据存储中可能会含有一个或多个组件：

- 虚拟机名字空间
- 虚拟机交换文件
- 虚拟机磁盘
- 虚拟磁盘快照

毫无疑问，本书读到这里你已经很清楚虚拟机会拥有一个名字空间、一个交换文件并且通常有一个磁盘。很重要的一点是，理解允许的故障数这个设置非常重要，它会影响组件的数量。配置的 FTT 的值越大，意味着拥有的对象的组件会越多。也就是说，当配置 FTT 为 1 时，意味着磁盘对象将有 2 个镜像（换而言之 2 个组件）。对于条带宽度也是同样。如果它被增大到超过 1 的数字，组件数量也会增加。因此，如果一个虚拟机磁盘（VMDK）对象被条带化分拆到 2 个磁盘上，你将具有 2 个组件。除此之外，一个组件的最大尺寸是 256GB，这意味着如果你有一个 512GB 的虚拟机磁盘对象，这个对象会被配置成 2 个 256GB 的组件。这在进行容量配置和扩容决策的时候非常重要。

让我们通过一个例子来分析一下更改存储策略会带来怎样的影响。

9.2　允许的故障数为 1 且条带宽度为 1

这意味着每个虚拟机的最小组件数量是 6。不过，如果虚拟机配置了 2 个磁盘，那么总组件数量将增加到 8。如果对该虚拟机拍了 1 个快照，将会因每个磁盘而增加 2 个组件，最终的组件数将达到 12。

- 1 个虚拟机名字空间对象 = 2 个组件（因为弹性的要求）
- 1 个虚拟机交换文件对象 = 2 个组件（因为弹性的要求）
- 2 个虚拟机磁盘对象 = 4 个组件（因为弹性的要求）
- 每个磁盘 2 个虚拟机快照文件对象 = 4 个组件（因为弹性的要求）

正如我们所演示的那样，组件的数量会快速增长，如果在一台主机上运行 100 个虚拟机，每台虚拟机都有 5 个磁盘和 5 个虚拟机快照，那么在使用标准策略（允许的故障数为 1）的情况下，组件数计算如下：

- 100 × 虚拟机名字空间对象 = 100 × 2 组件 = 200
- 100 × 虚拟机交换文件对象 = 100 × 2 组件 = 200
- 100 × 5 个虚拟机磁盘对象 = 500 × 2 组件 = 1 000
- 100 × 5 个虚拟机磁盘对象每个都有 5 个快照对象 = 5 × 500 × 2 组件 = 5 000

可以看出，“仅仅”在 5 个磁盘和 5 个快照的情况下就导致组件数超过 6 000。你可能会想知道是否可以验证当前的组件总数。你将在第 10 章中学到如何用 RVC 来获取此信息。

我们不会把这个例子延伸到下面的 2 个场景中，因为大多数环境可能都不会与之类似，但是我们的确坚信理解这些限制是非常重要的。

9.3　闪存磁盘比

在 VSAN Beta 版中，从容量的角度 VMware 建议闪存磁盘比设成 10%。这个建议随着时间推移到今日被改成下面的版本了：

在考虑允许的故障数之前，一般建议将 Virtual SAN 的闪存容量配置成预期消耗存储空间的 10%。

让我们用一个简短的例子来解释这个变化。假设我们的环境如下：

- 100 台虚拟机
- 每个虚拟机 50GB
- 预计 VMDK 中有 50% 的空间被消耗

根据老的建议，这意味着我们建议配备 10% × 100 VM × 50GB × 2（FTT = 1）= 1000GB 的闪存容量。注意，消耗掉多少存储没有被考虑进来，只是计算了包括镜像对象在内的

总存储容量。而根据新的建议，这意味着我们的建议是 10% × 100 VM × （50% × 50GB）= 250GB 闪存容量。这次我们没有把镜像考虑进来，而是只考虑了消耗掉的总磁盘空间。

假设我们使用的主机数是最低要求，也就是 3 台。这意味着在老的场景中我们需要给每台主机配置的闪存容量是 330GB（1000GB/3 台主机），而通过新的公式只需要在 3 主机的群集中给每台主机配备 85GB 的闪存（250GB/3）。因为较少的读缓冲和写缓存，这当然会有性能上的影响。记住读缓冲和写缓存之间有 70/30 的分配比，那么在这种情况下结果是怎样的？

- 59.5GB 与 231GB 读缓冲
- 25.5GB 与 99GB 写缓存

在下面的练习中，我们将采用“预期消费容量的 10%”这个规则。

9.4 性能设计

性能当然是设计一个 VSAN 基础架构的重要关注点。在接下来要描述的各种场景中，我们的关注点放在容量设计以及部分性能方面。当论及性能时，我们会讨论由磁盘提供的 IOPS 数，而不是由闪存提供的 IOPS 数。

你应该已经在本书中学到，VSAN 严重依靠闪存设备来提供所需的性能。闪存被同时用作为读缓冲和写缓存，因此，如同前面章节所描述过的那样，不正确地配置闪存的容量可能会对工作负载造成严重的性能影响。用于 33 台虚拟机的 59.5GB 读缓冲和 231GB 读缓冲之间的区别是巨大的。就减少磁盘的访问来说，每台虚拟机具有 1.8GB 还是 7GB 缓存的确是有区别的。对虚拟机性能来说，不仅仅是闪存设备的容量有关系，闪存的类型也有关系。

总而言之，所有这些提到过的最佳实践都只是推荐做法，它们对于大多数环境都适用，但是你的环境可能有所不同，可能会有更多苛刻的应用程序。在集群上运行着的应用程序的总活动（热点）数据才是关键所在。现实中，你必须要对此做出估算。不幸的是，在写作本书的时候，还没有什么工具可以帮你来完成这一任务。不过，VMware 正在开发一些工具，能够在群集闪存容量的需求上给你提供一些更精确的数据参考。

第 2 章曾经列出了闪存设备的分级，VMware 用这种分级来对此类设备可以达到何种程度的性能表现提供一个参考。

VMware 兼容性指南中标出的指定闪存设备等级列表如下：

- Class A：每秒 2 500 ～ 5 000 次写操作
- Class B：每秒 5 000 ～ 10 000 次写操作
- Class C：每秒 10 000 ～ 20 000 次写操作
- Class D：每秒 20 000 ～ 30 000 次写操作
- Class E：每秒 30 000 + 次写操作

为了演示不同设备之间的区别，我们将某些设备的理论性能表现举例如下：

- FusionIO ioDrive2，800GB，230k 随机写 IOPS，215k 随机读 IOPS（使用 4K 大小的数据块）（http://www.fusionio.com/data-sheets/iodrive2）
- Intel S3700，800GB，36k 随机写 IOPS，75k 随机读 IOPS（使用 4K 大小的数据块）（http://www.intel.com/content/www/us/en/solid-state-drives/solid-state-drives-dc-s3700-series.html）
- Samsung SM1625，800GB，23k 随机写 IOPS，100k 随机读 IOPS（使用 4K 大小的数据块）（http://www.samsung.com/global/business/semiconductor/news-events/press-releases/detail? newsId=12244）

如果你在 4 台主机上运行了 400 台虚拟机，使用 FusionIO 闪存卡比起使用 Intel S3700 可能会有非常大的性能差异。那么想象一下在 16 台主机上运行 2000 台虚拟机会怎样！当然，这二者之间的价格差异也同样显著，但是这的确是一个需要重视的考虑因素。

磁盘控制器的影响

一个常被问起的问题是，磁盘控制器队列深度对 VSAN 环境的性能会带来怎样的影响。考虑一下从虚拟机逐层向下直到设备本身的不同层面的各种队列，可能用如图 9-1 所示的图解来描述最能说明问题。

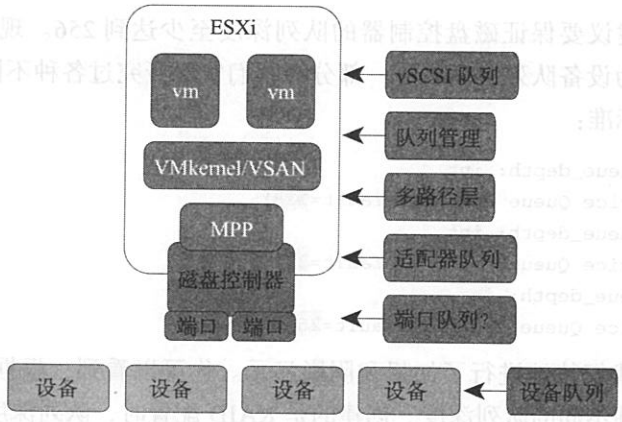


图 9-1　不同层面的排队

图 9-1 显示了 6 个不同层次的队列形式，当然现实中可能会有更多种类的缓存和队列（不过我们尽可能保持合理的简化并解释潜在的瓶颈会发生的层面）。在客户操作系统内，vSCSI 适配器具有一个队列。下一个层面是 VSAN，也有其自己的队列和 I/O 管理。接下去，I/O 数据流会通过多路径层到达主机上的各种不同设备，再下面一个层面中，磁盘控制器也有一个队列，而且（根据使用的控制器类型）有可能每个磁盘控制器端口上各存在一个队列。最后而且也是很重要的一点，每个设备（也就是磁盘）会有一个队列。

仔细观察图 9-1，你会发现很多虚拟机的 I/O 将流过同一个磁盘控制器，并且 I/O 会发往或来自一个或多个设备（通常会是多个设备）。这可能会是第一个真正的潜在瓶颈点——

磁盘控制器的队列深度。

假设有 4 个 SATA 磁盘，每个磁盘的队列深度是 32，这意味着合在一起需要并行处理 128 个 I/O。那么如果磁盘控制器只能处理 64 个请求会怎样？这将导致 64 个 I/O 被保留在 VMkernel/VSAN 上。如你所见，如果能保证磁盘控制器的队列能保存的数量和设备队列能保存的数量总数相同（或更多）就太好了，这样 VSAN 就可以在不受磁盘控制器限制的情况下进行队列整形。

谈起磁盘控制器，不同生产制造商产品之间队列最大深度的区别非常巨大，甚至同一制造商的不同型号之间也有明显差异。表 9-1 列出了 5 种流行的磁盘控制器及其队列深度，这个示例只是为了说明没有经过调研就做出决定会错得多么离谱。

表 9-1 磁盘控制器队列深度

制造商	磁盘控制器	队列深度
Dell	PERC H710 Adapter	975
HP	Smart Array P420i	1 020
Intel	C602 AHCI (Patsburg)	31（每端口）
LSI	2008	25
LSI	2308	600

对于 VSAN，建议要保证磁盘控制器的队列深度至少达到 256。现在磁盘控制器只是公式的一部分，因为设备队列占据了另一部分。我们曾经研究过各种不同的控制器和设备，下面是一些典型的标准：

```
mpt2sas_raid_queue_depth: int
   Max RAID Device Queue Depth (default=128)
mpt2sas_sata_queue_depth: int
   Max SATA Device Queue Depth (default=32)
mpt2sas_sas_queue_depth: int
   Max SAS Device Queue Depth (default=254)
```

最重要的部分我们特地进行了加粗和阴影显示。你可以看到，根据所用设备类型的不同，控制器具有 3 种不同的队列深度。创建的是 RAID 配置时，队列深度是 128；而当直连一个 SAS 驱动器的时候（常称之为 VSAN SAS 或直通），队列深度是 254。最常见的 SATA 设备默认队列深度只有 32，可以预见到，这又会是一个瓶颈点。不过，幸运的是，SATA 较小的队列深度的问题可以通过替换成 NL-SAS 驱动器（近线串行连接 SCSI）来解决，这种驱动器的队列深度要大得多。

可以通过使用 `esxcfg-info -s | grep "==+SCSI Interface" -A 18` 命令行来验证控制器的队列深度，这条命令会显示出 SCSI 接口的大量信息，其中也包括队列深度。

注意，甚至是所用的固件和驱动程序也会对控制器和设备的队列深度造成影响。我们强烈建议使用 VSAN 兼容性指南中列出的驱动程序。如图 9-2 所示，这是某个特定磁盘控制器的详情页面（http://www.vmware.com/resources/compatibility/ search.php?deviceCategory =

vsan）。有时候，你会在驱动程序更新后看见队列深度从 25 增加到 600。可以想象，这将对性能造成多大的影响啊！

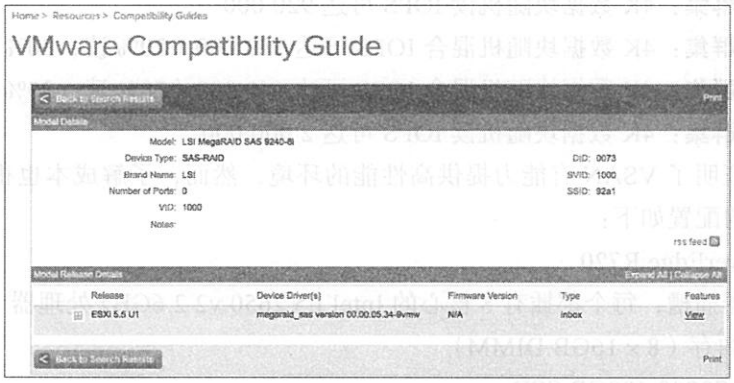


图 9-2　设备驱动程序详情

此时我们通常会遇到这样的问题：NL-SAS 和 SATA 驱动器对比是怎样的？因为 NL-SAS 本质上只是使用了 SAS 连接器的 SATA 驱动器，它的优势在哪里？ NL-SAS 驱动器有以下优点：

- 双端口，可用于冗余路径
- 能连接一个设备到多台电脑
- 支持完整的 SCSI 命令集
- 比 SATA 更快的接口（多达 20%），无 STP（串行 ATA 隧道协议）开销
- 更深的命令行队列（深度）

从成本角度来看，大多数生产商的 NL-SAS 和 SATA 之间的价差几乎可以忽略。写作本书时，4TB 驱动器在不同电商网站上的平均价差是 30 美元。考虑到其带来的实际好处，我们强烈推荐为 VSAN 选购 NL-SAS 而不是 SATA。

读完本节，你应该很清楚对于 VSAN 设计而言磁盘控制器是个关键组件，不仅仅是磁盘控制器本身，还包括固件和设备驱动程序，这些都会对 VSAN 的性能带来很大的影响。在部署 VSAN 的时候，我们强烈建议你确保驱动程序的版本是 VSAN 兼容性列表中推荐的版本，或者在需要的时候升级到最新版本。

9.5　VSAN 的性能

很难对性能进行预测，因为工作负载各不相同而且硬件组合也各式各样，这些都会带来不同的结果。在 VSAN 发布之后，VMware 公布了多个性能测试的数据：（http://blogs.vmware.com/vsphere/2014/03/supercharge-virtual-san-cluster-2-million-iops.html）。结论令人印象深刻。但是请注意，这些实验不能保证你的环境中可以实现同样的结果。这些理论测

试不需要（而且大多数情况下也不可能）和你环境中的 I/O 模式是相同的（因此结论也会不同）。虽然如此，我们觉得下面的这些数据还是值得分享的。

- **16 主机群集**：4K 数据块随机读 IOPS 可达 920 000
- **16 主机群集**：4K 数据块随机混合 IOPS 可达 320 000（70% 读，30% 写）
- **32 主机群集**：4K 数据块随机混合 IOPS 可达 640 000（70% 读，30% 写）
- **32 主机群集**：4K 数据块随机读 IOPS 可达 2 000 000

我们已经证明了 VSAN 有能力提供高性能的环境，然而，了解成本也很重要。达到上述数字的系统的配置如下：

- Dell PowerEdge R720
- 双处理器插槽，每个都插有 8 核心的 Intel E5 2650 v2 2.6GHz 处理器
- 128GB 内存（8 × 16GB DIMM）
- 1 × Intel S3700 400GB SSD
- 7 × 10K RPM 1.1 TB SAS 磁盘
- LSI 9207-8i 磁盘控制器
- Intel 82599EB 10GbE 网卡

我们对配置进行了部分微调，以对需要进行海量 I/O 的大型工作负载环境进行优化并扩展到 32 主机。下面列出了这些变更：

- 允许 VSAN 构建 32 主机群集（advanced setting: /adv/CMMDS/goto11）
- vSphere 网络堆栈大小增加到 512MB（advanced setting: /adv/Net/TcpipHeapMax）
- VMware Paravirtual SCSI（PVSCSI）开启时间参数最好支持高 I/O 大规模的工作负载："vmw_pvscsi.cmd_per_lun=254 vmw_pvscsi.ring_pages=32".（PVSCSI 适配器是虚拟的高性能存储适配器，可以带来极高的吞吐量和较低的 CPU 利用率）
- 禁用所有电源管理特性
- 每台主机使用单个 VSAN VMkernel 端口并具有万兆以太网上行链路

注意所有这些 VMware 公开提供的性能测试和参考架构都是基于万兆以太网络的。在我们的设计场景中，将使用万兆以太网作为黄金标准，因为这是 VMware 强力推荐的，它能够增加吞吐量并降低延迟。不同的配置选项，包括如何使用 NIOC 已经在第 3 章中讨论过了。

VMware View 性能

除了性能的理论最大值之外，VMware 已经通过 View Planner 和各种群集配置（从 3 台到 16 台主机的群集）验证过 VSAN 的性能。这个测试将 VSAN 的性能和全闪存阵列进行了对比——通过比较相同数量的 VSAN 主机和非 VSAN 的连接到外部（全闪存）存储系统的主机上可以运行多少个桌面来显示 VSAN 的开销：

承载了桌面虚拟机的主机具有 16 个 Intel Xeon E5-2690 处理器内核，每个内核 2.9GHz。

主机具有 256GB 物理内存，足够运行 100 个内存 1GB 的 Windows 7 虚拟机。在 VSAN 配置中，每台主机具有 2 个磁盘组，每个磁盘组都具有一个 200GB PCI-e 固态硬盘和 6 个 300GB 15K RPM 的 SAS 硬盘。

View Planner 基准测试的结果是：在 3 主机的 VSAN 群集上可以运行 305 个桌面，在 8 主机的群集上可以运行 803 个桌面，在 16 主机的群集上可以运行 1615 个桌面，可以看出 VSAN 可以做到线性扩展，响应时间的性能指标没有下降。相比全闪存阵列，VSAN 并没有显著的额外开销。在同样的 8 主机群集的情况下，全闪存阵列可以支持 805 个桌面，仅比 VSAN 群集多了 2 个，这点小小的差别完全可以忽略不计。

更重要的是，即使是在响应时间上，VSAN 的性能也和全闪存阵列接近，如图 9-3 所示。然而，全闪存阵列的价格要贵得多。根据 VMware 的数据，VSAN 方案的成本大约只有全闪存阵列的 25%。

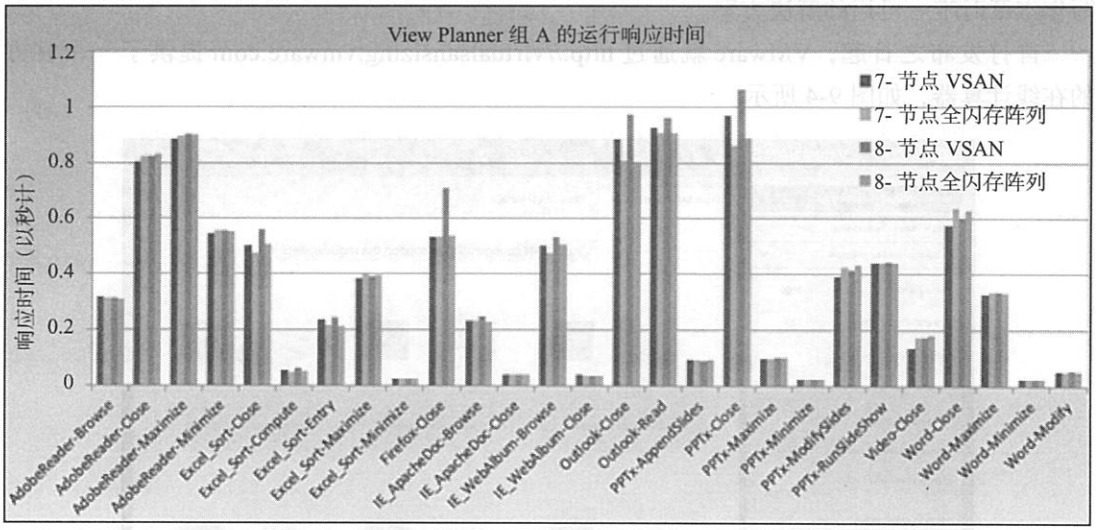


图 9-3　View 运行的响应时间

可以在下列 VMware 网站找到更多测试结果：

- VDI Benchmarking Using View Planner on VMware Virtual SAN Beta – Part 1
  http://blogs.vmware.com/performance/2013/10/vdi-benchmarking-using-view-planner-on-vmware-virtual-san-vsan.html
- VDI Benchmarking Using View Planner on VMware Virtual SAN Beta – Part 2
  http://blogs.vmware.com/performance/2013/11/vdi-benchmarking-using-view-planner-on-vmware-virtual-san-part-2.html
- VDI Benchmarking Using View Planner on VMware Virtual SAN Beta – Part 3
  http://blogs.vmware.com/performance/2013/11/vdi-benchmarking-using-view-planner-on-vmware-virtual-san-part-3.html

- VDI Benchmarking Using View Planner on VMware Virtual SAN GA（vSphere 5.5 U1）http://blogs.vmware.com/performance/2014/03/vdi-performance-benchmarking-vmware-virtual-san-5-5.html

现在，让我们来看一看某些性能方面的制约因素，并从容量的角度来探讨一下如何设计 VSAN 群集。

9.6 设计和容量规划工具

在开始第一个设计之前，我们想告诉你一些可以帮助进行 VSAN 基础架构设计和容量规划的工具。本章中提供的这些场景案例都使用了 VSAN Calculator（VSAN 计算器），你可以在这里访问到：http://vmwa.re/vsancalc。虽然这个计算器并非官方出品的工具，但却是在写作本章时唯一可用的解决方案。

自打发布之日起，VMware 就通过 http://virtualsansizing.vmware.com 提供了一个官方的在线计算器，如图 9-4 所示。

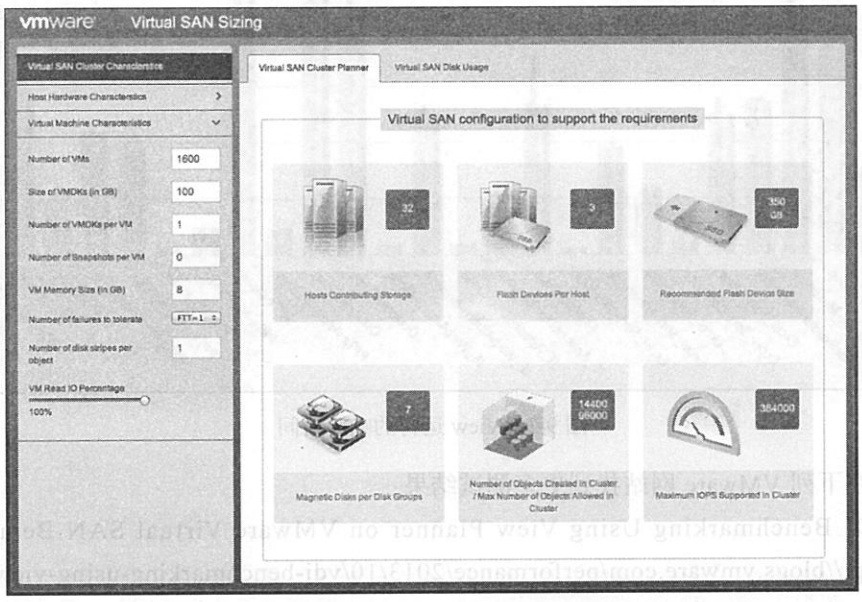


图 9-4 官方的 VMware VSAN 容量规划计算器

VMware 的这个官方的 VSAN 容量规划计算器使你可以根据特定的参数来进行设计，这些参数列举如下：

- 虚拟机的数量
- 虚拟机磁盘（VMDK）的大小
- VMDK 的数量

- 快照的数量
- 读写 I/O 比

我们强烈建议你使用这些工具来验证自己的设计和决策，以此来保证工作负载的最佳性能和可用性。而其中更推荐的是 virtualsansizing.vmware.com，因为它是 VMware 官方支持的容量规划计算器。

9.7　场景 1

在设计 VSAN 环境时，充分理解虚拟机的需求是非常重要的。这里显示的不同的例子会告诉你不同的决策会带来怎样的后果。任何设计都是从收集需求开始的。在这个例子中，我们收集到的参数如下：

- 平均每台虚拟机 1.5 个 vCPU
- 平均每台虚拟机 5GB 内存
- 平均每台虚拟机 54GB 磁盘空间
- 每台虚拟机预期的磁盘消耗为 50%

在这个环境中，目前已有 173 台虚拟机，并且在接下去的 18 个月内会增长到 250 台，这意味着我们的 VSAN 基础架构应该有能力提供以下这些资源：

- 250 × 1.5 vCPU = 375 vCPUs
- 250 × 5GB = 1250 GB 内存
- 250 × 54GB = 13500 GB 磁盘空间

我们会考虑采用 5:1 的 vCPU/ 内核比，因为虚拟机总共需要 375 个 vCPU，除以 5，结果是我们共需要 75 个处理器内核。

再来深入一点地看看存储需求。在开始计算之前，我们需要知道这些虚拟机需要何种程度的弹性。在我们的计算中，会考虑将允许的故障数设置为 1，我们还增加了 10% 的额外磁盘空间来保存元数据（metadata）和临时快照。如果你的环境中需要为快照提供更多的空间，在进行计算练习之前不要忘记将这个因子考虑进去。

在将以上因素考虑进去之后，公式看上去是这个样子的：

（虚拟机数量 × 磁盘平均容量 + 虚拟机数量 × 内存平均容量）×（FTT +1）+ 10% 余量

这里将内存平均容量计算在内是因为每台虚拟机会在磁盘上创建一个大小等同于内存配置容量的交换文件。使用前面提到的业界标准的平均值，结果是这样的：

（250×54 + 250×5）× 2 =（13,500 + 1250）× 2 = 29500GB + 10% 余量 = 32450GB

将结果除以 1024 再进位取整，总共需要的存储空间是 32TB。现在我们知道将需要 32TB 磁盘空间、1250GB 内存和 75 个处理器内核，接着让我们来研究一下如何配置硬件。

决定主机配置

我们将从 2U 主机这种最常见的机型开始研究。在这个例子中，我们决定选用 Dell R720XD，其外观如图 9-5 所示。这台服务器已经对存储空间进行了优化，可以选配 12 个 3.5 英寸的磁盘驱动器或 24 个 2.5 英寸的磁盘驱动器。Dell R720XD 是一台双处理器服务器，最多支持 768GB 内存，两个处理器插槽上都可以配备任何 4 核到 12 核的 CPU。关于 Dell R720XD 的更多细节可到官方网站获得：http://www.dell.com/us/business/p/poweredge-r720xd/pd。

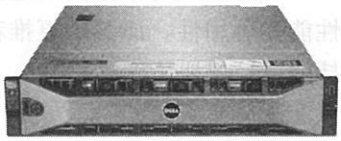


图 9-5 Dell R720XD 服务器

我们环境的必要条件如下：

- （根据 VSAN 的要求）最少 3 台主机
- 32TB 的磁盘总原始容量
- 75 个 CPU 内核
- 1250GB 内存
- 在故障情况下最小的过量配置（高可用要求 N + 1）

如前所述，Dell 720XD 可配置 12 核处理器，每台服务器 2 个 CPU 的情况下最多可以配置 24 个处理器内核。这意味着从 CPU 的角度考虑，我们大致需要 3 台主机。然而，因为考虑到有故障情况下过量配置的需求，则至少需要配置 4 台主机。从内存角度考虑，每台主机可以提供 768GB 内存，考虑到我们需要 1250GB 的内存，2 台主机就足够了。因为同时也要考虑故障情况下过量配置和最优的内存配置，我们决定每台主机配置 512GB 内存。

当然，价格总是重要因素，我们强烈建议基于 CPU、内存和磁盘配置来比较价格。在这个例子中，我们不会考虑价格因素，因为计算机组件的报价一直在变化。本例中，我们决定使用 4 台主机，因为从 CPU 和内存角度出发最多需要 4 台主机。我们会根据这个结果来进行磁盘配置的设计。

存储的容量设计略微妙。让我们来看一下以下 4 种不同的方案。我们知道总共需要 32TB 的存储，我们还知道一个 VSAN 群集最少需要 3 台主机。考虑到我们可以选择 3.5 英寸磁盘或 2.5 英寸磁盘，根据我们前面配置中选择了 4 台主机，总共可以最多拥有 96 个 2.5 英寸磁盘插槽或 48 个 3.5 英寸磁盘插槽。在决定磁盘类型的时候还应该考虑到每组磁盘（最多 7 个）需要 1 个闪存设备。

一个重要的考虑因素是由磁盘和闪存设备共同提供的 IOPS 数。一个典型的 3.5 英寸 7200RPM 的 SATA 驱动器可以提供大概 80 IOPS，而一块 2.5 英寸的 10K RPM 的 SAS 磁盘可以提供 150 IOPS。（IOPS 数值来自于 http://www.techrepublic.com/blog/the-enterprise-cloud/calculate-iops-in-a-storage-array/）从容量的角度来说，SATA 磁盘的范围是 1TB 到 4TB，而现在最大的 SAS 磁盘也只有 1.2TB。我们知道共需 32TB，让我们来速算一下看看

这对决策会有怎样的潜在影响。我们将采用最极端的情况来进行最大程度上的对比，这意味着用容量大但慢速的 SATA 磁盘来和小容量但相对快的 SAS 磁盘进行比较。

- 32TB / 4TB = 约 8 个 SATA 磁盘 = 640 IOPS（从 SATA 磁盘可获得的性能）
- 32TB / 0.6TB = 约 54 个 SAS 磁盘 = 8100 IOPS（从 SAS 磁盘可获得的性能）

可以看出，这 2 种极端例子的结果差异巨大。尽管 VSAN 已经被设计成利用闪存设备作为主要的性能来源，这还是一个非常重要的设计考虑因素，因为当数据需要被回写到磁盘上时或当读缓冲未能命中而数据块需要从磁盘上直接读取时，这些 IOPS 仍然会被用上。在这个例子中，我们决定使用 SAS 磁盘，这样即使数据不在缓存中时仍然能获得不错的性能。问题是为什么配置中不选用 1.2TB 的磁盘呢？答案很简单，因为不是每个硬件厂商都提供 1.2TB 磁盘，并且它的价格仍然相对较贵。让我们来算一下，如前所述，我们总共需要 54 个 600GB 的 SAS 磁盘，分布到 4 台主机上，平均每台主机 14 块盘。

为了保证我们的 VSAN 群集能提供最优的用户体验，根据经验，我们将采用虚拟机预期消耗磁盘总容量的 10% 来作为闪存容量。在我们的例子中，我们最多会有 250 台虚拟机，这些虚拟机拥有总共 54GB 的虚拟磁盘空间，其中我们预期会实际消耗 50% 的空间，根据推荐值计算得出的闪存容量如下：

10% ×（250 台虚拟机 ×（预期消耗 50% × 54GB 总磁盘空间））= 675GB

考虑到我们将配置 4 台主机，这意味着每台主机上的闪存空间需要 675GB 除以 4 约等于 169GB。注意，因为每个磁盘组最多 7 个磁盘，我们的配置中每台主机会有 14 块磁盘，因此必须最少配置成 2 个磁盘组。每个磁盘组需要有其自己的闪存设备。根据 VMware 兼容性指南，就 Dell 的配置和最低每秒 20000 次写操作的性能要求（D 类和 E 类满足要求），我们决定从下列闪存设备中进行选择：

- 200GB SAS 写密集型（WI）6Gbps 2.5 英寸 SSD
- 400GB SAS 写密集型（WI）6Gbps 2.5 英寸 SSD

要为每台主机提供 170GB 的闪存容量，我们只需要为每台主机配置 1 块 200GB 的闪存盘。但是考虑到 2 个磁盘组，最终必须配置 2 块 200GB 的闪存盘。

最后一个重要的配件是磁盘控制器。VMware 建议配置直通控制器。Dell 为 R720XD 提供了 2 个型号：H310 和 H710P。H310 是一款标准的直通控制器，而 H710P 提供了诸如缓存、自我加密驱动器和其他各种高级功能。考虑到 VSAN 会使用自己的缓存逻辑，我们决定选择 H310。

最终 4 台 Dell R720XD 主机（带有 2.5 英寸驱动器插槽）的配置如下：

- 2 颗 12 核 E5-2695 处理器
- 512GB 内存
- 磁盘控制器：Dell H310
- 14 块 600GB SAS 10K RPM 磁盘
- 2 块 200GB SAS（WI）SSD

9.8 场景 2

在前面的场景中，我们例子中虚拟机的规模算是中等大小的，因此群集规模比较小。这一次将需要在环境中运行 1500 台虚拟机。这将导致一系列额外的挑战，因此和前面的场景差异较大。这个例子中虚拟机的参数如下：

- ❑ 平均每台虚拟机 1 个 vCPU
- ❑ 平均每台虚拟机 6GB 内存
- ❑ 平均每台虚拟机 50GB 磁盘空间
- ❑ 每台虚拟机预期的磁盘消耗为 75%

这个环境中当前有 12 台虚拟机，最终会在接下去的 12 个月内增长到 1500 台虚拟机。因此，VSAN 基础架构应该能提供以下资源：

1 500 × 1 vCPU = 1 500 vCPU

1 500 × 6GB = 9000GB 内存

1 500 × 50GB = 75000GB 磁盘空间

我们将采用的 vCPU/ 内核比为 7:1。考虑到总共需要 1500 个 vCPU，除以 7 得到 215 个 CPU 内核。而总的内存需求是 9000GB。

接下来让我们一起进一步研究一下存储的需求。在开始计算之前，我们还是需要看一下虚拟机对弹性的要求。这次我们将允许的故障数设为 1，并将在故障时重建存储对象。因为在这个案例中考虑了备用的磁盘容量，因此确保了发生一台主机故障时所有虚拟机都能被保护。我们还增加了 10% 的额外磁盘空间来保存元数据和临时快照。

将以上因素考虑进来以后公式看上去是这样的：

（虚拟机数量 × 磁盘平均容量 + 虚拟机数量 × 内存平均容量）×（FTT + 1）+ 10% 余量

这里将内存平均容量计算在内是因为每台虚拟机会在磁盘上创建一个大小等同于内存配置容量的交换文件。使用前面提到的业界标准平均值，结果是这样的：

（1 500 × 50 + 1 500 × 6）× 2 =（75 000 + 9000）× 2 = 168000GB + 10% 余量 = 184800GB

将结果除以 1024 再进位取整，总共需要的存储空间是 181TB。现在我们知道将需要 181TB 磁盘空间、9000GB 内存和 215 个处理器内核，接着让我们来研究一下如何配置硬件。

决定主机配置

在这个例子中，客户指出它们将惠普的硬件设备作为其环境中的标准配置，而且希望使用 HP ProLiant DL380p Gen8 的 2U 主机（见图 9-6）。DL380p 可以配备 2 颗处理器（每 CPU 最多 12 核）和 768GB 内存（24 条 DIMM 插槽），并具有 25 个磁盘插槽。

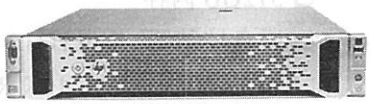


图 9-6 HP DL380

我们对环境的要求是：

- ❑ 最少 3 台主机（根据 VSAN 的要求）

- 181TB 原始磁盘容量
- 215 颗 CPU 内核
- 9000GB 内存

如前所述，HP DL380p 可以配备最多 12 核处理器，因此每台双处理器的主机可以有 24 颗内核。因此，从 CPU 角度来计算，我们大概最少需要 9 台主机。从 内存角度来看，每台主机可以提供 768GB 内存，考虑到我们需要 9000GB 内存，那么至少需要 12 台主机才能满足内存的需要。当然，价格总是重要的考虑因素，我们强烈建议你基于 CPU、内存和磁盘配置来比较价格。因为计算机组件的报价总是在持续变动，这里我们不会考虑价格因素。

再来看一看存储的选择。我们知道一共需要 181TB 存储。接下来我们将从磁盘大小及性能来比较几种不同的配置。

- 181TB = 约 91 * 2TB = 7 280 IOPS
- 181TB = 约 151 * 1.2TB = 22 650 IOPS
- 181TB = 约 201 * 0.9TB = 30 150 IOPS
- 181TB = 约 301 * 0.6TB = 45 150 IOPS

我们知道每台服务器可以配备 25 个 2.5 英寸磁盘驱动器或 12 个 3.5 英寸的磁盘驱动器。从磁盘角度来看，如果使用 0.6TB 的磁盘则需要 15 台主机。你可能会质疑为什么需要 15 台主机，因为 301/25 小于 15。但是我们必须考虑到还需要为 SSD 准备一些插槽。前面的例子你或许还记得，每个磁盘组（最多 7 块磁盘）需要一块 SSD。如果你创建了 3 个磁盘组，每个磁盘组 7 块磁盘，则会需要 3 块 SSD，共使用 24 个磁盘插槽。这样，总共 301 块磁盘除以每台主机 21 块磁盘结果是 14.3 台主机，进位后得到 15 台主机。

因为我们希望同时具有横向扩展和纵向扩展的能力，所以最终决定使用 20 台主机，并采用 900GB 的磁盘驱动器。这让我们留下了足够的空槽位，可以在需要的时候添加存储容量。总共需要 181TB 空间，分配到 20 台主机上每台大约 9TB。我们将使用 12 个 900GB 的磁盘驱动器而不是 11 个，因为 12 是个偶数，我们希望均衡每个磁盘组的磁盘个数，让数量保持一致都是 6 个，这样每台主机上会配置 2 个磁盘组。

使用闪存空间的 10% 规则，我们需要 1500 ×（75% × 50GB），也就是总共 5625GB 的闪存容量。考虑到我们会在群集中配置 20 台主机，也就是每台主机大约 280GB。由于闪存设备的常规容量是 100、200、400 和 800GB，我们将选用 200GB 的驱动器，这是因为主机有 2 个磁盘组会需要 2 个 SSD。由于我们的配置略微超过了需求，应该不会有任何的性能惩罚。

最后列出 20 台 HP DL380p（25 SFF）主机的配置清单：

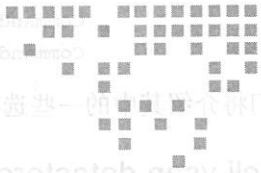
- 双处理器，6 核 E5-2630
- 512GB 内存
- 磁盘控制器：HP H220i

- 12 块 900GB 10K RPM SAS 磁盘
- 2 块 200GB 写密集型 SAS SSD

9.9 小结

如同本章中的这些例子所说明的那样，仔细选择硬件的确非常重要。例如，选择了某种类型的磁盘，可能会导致潜在的性能问题。而其他组件如磁盘控制器，也会对 VSAN 的运行带来某种影响。

你应该在做出购买决策之前深思熟虑。如前所述，VMware 提供了一个绝好的选择——VSAN Ready Node 计划。这些服务器针对 VSAN 进行了优化和配置，简化了由不同硬件选择带来的复杂性。



第 10 章 Chapter 10

排错、监控和性能

本章讨论可在 VSAN 环境中进行监控和排错的可用的扩展工具集，并介绍如何利用这些工具快速诊断并解决 VSAN 的问题。

VSAN 可以利用这些现成的工具以及专用于 VSAN 的内建工具。本章将覆盖以下工具。

- ESXCLI：ESXi 主机的命令行界面（CLI）
- Ruby vSphere Console（RVC）：管理 vCenter 实例的常规工具，但还可以扩展支持 VSAN 管理
- VSAN Observer：利用 RVC 的一款基于 Web 的性能工具
- Esxtop：ESXi 主机性能监控工具

也应该关注传统的监控工具，例如 vSphere Web 客户端，它可继续用于 VSAN，提供单个虚拟机及其相关 VMDK 的性能视图。

10.1 ESXCLI

ESXi 5.5U1 引入了一个新的 ESXi 命令行（ESXCLI）指令集：esxcli vsan。它拥有一组额外的命令参数，用于检查、监控和配置 VSAN 群集，如下：

```
~ # esxcli vsan
Usage: esxcli vsan {cmd} [cmd options]
Available Namespaces:
  datastore             Commands for VSAN datastore configuration
  network               Commands for VSAN host network configuration
  storage               Commands for VSAN physical storage configuration
  cluster               Commands for VSAN host cluster configuration
```

```
maintenancemode        Commands for VSAN maintenance mode operation
policy                 Commands for VSAN storage policy configuration
trace                  Commands for VSAN trace configuration
```

接下来我们将介绍其中的一些选项。

10.1.1 esxcli vsan datastore

esxcli vsan datastore 指令集可用来进行 VSAN 数据存储配置。它在最初的 VSAN 发行版本中能做的事情很少，只能用来获取和设置 VSAN 数据存储的名字。VSAN 数据存储的默认名字是 vsanDatastore。如果你有想法要更改 vsanDatastore 这个名字，将不得不在群集中的每台 ESXi 主机上不断重复这条命令，因此通过 vSphere Web 客户端在群集层面上进行更改或许更佳。如果你正通过同一个 vCenter Server 管理多个 VSAN 群集，我们强烈建议你给每个 VSAN 数据存储起一个独一无二且容易辨识的名字。

```
~ # esxcli vsan datastore
Usage: esxcli vsan datastore {cmd} [cmd options]

Available Namespaces:
  name                 Commands for configuring VSAN datastore name.

~ # esxcli vsan datastore name
Usage: esxcli vsan datastore name {cmd} [cmd options]
Available Commands:
  get                  Get VSAN datastore name.
  set                  Configure VSAN datastore name. In general, Rename
                       should always be done at cluster level. Across a VSAN
                       cluster VSAN datastore name should be in sync.

~ # esxcli vsan datastore name get
   Name: vsanDatastore
```

10.1.2 esxcli vsan network

这个指令集用于 VSAN 的网络配置。某种程度上说，它比前面的 *datastore* 指令集还要有用，因为它可以列出当前配置、清除当前配置、恢复 VSAN 网络配置（这里指的是 ESXi 引导过程中的配置恢复，不是指由用户发起的恢复动作）以及从 VSAN 网络配置中移除一个接口。

```
~ # esxcli vsan network
Usage: esxcli vsan network {cmd} [cmd options]
Available Namespaces:
  ipv4                 Commands for configuring IPv4 network for VSAN.
Available Commands:
  clear                Clear the VSAN network configuration.
```

```
  list                  List the network configuration currently in use by
                        VSAN.
  remove                Remove an interface from the VSAN network
                        configuration.
  restore               Restore the persisted VSAN network configuration.
```

```
~ # esxcli vsan network list
Interface
   VmkNic Name: vmk6
   IP Protocol: IPv4
   Interface UUID: f4c8e352-f46e-c538-42ad-0011010700df
   Agent Group Multicast Address: 224.2.3.4
   Agent Group Multicast Port: 23451
   Master Group Multicast Address: 224.1.2.3
   Master Group Multicast Port: 12345
   Multicast TTL: 5
```

这里有趣的地方是组播信息。回顾一下第 2 章，你或许会记起 VSAN 要求加入到群集的 ESXi 主机之间允许组播流量通过。

另一个值得注意的地方是，现在 VSAN 只支持 IPv4，在这个最初的发行版本中不支持 IPv6。不过最吸引人的地方是组播的细节。Agent Group Multicast Port 对应于 *cmmds* 端口，这个端口在 VSAN 启用的时候在 ESXi 防火墙上被打开。第一个 IP 地址 224.2.3.4 用于主控和备用之间的通信，而第二个地址 224.1.2.3 用于代理。`esxcli vsan network list` 是一个有用的命令，可以在网络分区的情况下用来查看网络配置和网络状态。

其他对诊断网络问题有用的命令如下：

- `esxcli network diag ping`：测试 VMkernel 端口的响应。
- `esxcli network ip neighbor list`：显示 ARP 缓存中的网络上所有其他 VSAN 节点。
- `esxcli network ip connection list`：显示 UDP 连接信息。
- `tcpdump-uw`：嗅探网络流量。

10.1.3 esxcli vsan storage

这个指令集用于存储配置，通过其选项可以选择 VSAN 声明磁盘的方式，并能够给 VSAN 添加和移除物理磁盘。

`esxcli vsan storage automode` 命令可用来获取或设置自动声明（auto-claim）。如果它被设置成禁用，群集就处于 manual 模式（手动模式）。

```
~ # esxcli vsan storage
Usage: esxcli vsan storage {cmd} [cmd options]
Available Namespaces:
```

```
  automode          Commands for configuring VSAN storage auto claim mode.
Available Commands:
  add               Add physical disk for VSAN usage.
  list              List VSAN storage configuration.
  remove            Remove physical disks from VSAN disk groups.

~ # esxcli vsan storage automode
Usage: esxcli vsan storage automode {cmd} [cmd options]
Available Commands:
  get                 Get status of storage auto claim mode.
  set                 Configure storage auto claim mode
```

要显示某台 VSAN 中特定的 ESXi 主机上已经声明并被使用的磁盘和固态盘（SSD），可用 list 选项。这个特殊的配置只有一个磁盘和一个 SSD，并且在字段 Used by this host 边上都注明了 true 标志，说明它们都已经被 VSAN 声明过了。Is SSD 字段指明设备的类型（false 指磁盘，true 指 SSD）。

```
~ # esxcli vsan storage list
naa.5000c5006900c7cf
   Device: naa.5000c5006900c7cf
   Display Name: naa.5000c5006900c7cf
   Is SSD: false
   VSAN UUID: 526977e9-92fb-b7c1-0c88-4ed9ed73b0d1
   VSAN Disk Group UUID: 527b16ab-d7d2-ac98-229b-019ccdde17a3
   VSAN Disk Group Name: naa.50025385a00c5085
   Used by this host: true
   In CMMDS: true
   Checksum: 641692210327909576
   Checksum OK: true

naa.50025385a00c5085
   Device: naa.50025385a00c5085
   Display Name: naa.50025385a00c5085
   Is SSD: true
   VSAN UUID: 527b16ab-d7d2-ac98-229b-019ccdde17a3
   VSAN Disk Group UUID: 527b16ab-d7d2-ac98-229b-019ccdde17a3
   VSAN Disk Group Name: naa.50025385a00c5085
   Used by this host: true
   In CMMDS: true
   Checksum: 8907154225867547805
   Checksum OK: true
```

如果你想用 ESXCLI 来给 VSAN 中的某个磁盘组添加新磁盘，可以用 add 选项。根据硬盘是磁盘还是 SSD，选项有所不同（分别对应为 -d|--disks 或 -s|--ssd）。注意，只有空的且没有分区信息的硬盘才能加入 VSAN。

remove 选项可以用来从 VSAN 的磁盘组里面移除磁盘和 SSD。毋庸置疑，使用这条

命令时你必须非常小心，从 VSAN 的磁盘组中移除磁盘应该被视为一次维护任务。remove 选项会从指定硬盘（通过一个参数提供）中移除所有分区信息（因此也包含所有 VSAN 信息）。注意，如果从一个磁盘组中移除一个 SSD，整个磁盘组都会变得不可访问。

如果你有磁盘曾经用于 VSAN 而现在又打算派其他用场（VMFS、裸设备映射【RDM】或某些场合下把 SSD 用作 vSphere 闪存读取缓存），可以使用 remove 选项来清除遗留在磁盘上的 VSAN 分区信息。

关于磁盘和磁盘控制器的其他有用的命令如下：

- `esxcli storage core adapter list`：显示驱动程序和适配器的描述，这对于检查适配器是否位于硬件兼容性列表（HCL）中非常有用。
- `esxcfg-info -s | grep "==+SCSI Interface" -A 18`：这条命令将显示大量信息，其中最重要的是显示设备的队列深度，这对于性能来说非常重要。
- `esxcli storage core device smart get -d` *XXX*：显示关于驱动器的 SMART 统计信息，尤其对 SSD 而言它是一条非常有用的命令，可以显示 SSD 的耗损状况以及 SSD 的整体健康度。
- `esxcli storage core device stats get`：显示硬盘的整体统计信息。

10.1.4　esxcli vsan cluster

`esxcli vsan cluster` 命令允许执行命令的 ESXi 主机获取 VSAN 群集信息，还可以让其加入或离开一个 VSAN 群集。在 vCenter Server 不可用的情况下，这条命令非常有用，它可以从 VSAN 群集中移除一台特定的主机。恢复功能不是给用户自行发起使用的，而是在 ESXi 启动的过程中用于从配置文件中恢复有效的群集配置。

```
~ # esxcli vsan cluster
Usage: esxcli vsan cluster {cmd} [cmd options]

Available Commands:
  get             Get the information of the VSAN cluster that this host
                  is joined to.
  join            Join the host to a given VSAN cluster.
  leave           Leave the VSAN cluster the host is currently joined
                  to.
  restore         Restore the persisted VSAN cluster configuration.
```

这条命令的 get 选项很有用，可以收集本地 ESXi 主机（节点）的健康信息及其在群集中的角色信息。在下面的例子中，你将看到这台 ESXi 主机是一个 *agent*（代理）并且其健康状态是 healthy（正常）。如第 5 章中曾经讨论过的，其他状态有 *master*（主控）和 *backup*（备用）。这些状态与主机在群集服务（CMMDS）中扮演的角色有关。如果需要复习一下 VSAN 中主机的不同角色，请参考第 5 章。

```
~ # esxcli vsan cluster get
```

```
Cluster Information
   Enabled: true
   Current Local Time: 2014-02-08T10:11:15Z
   Local Node UUID: 52b20dab-3f82-819c-c5df-0011010700df
   Local Node State: AGENT
   Local Node Health State: HEALTHY
   Sub-Cluster Master UUID: 52b228a4-2235-2422-1b19-00110107007f
   Sub-Cluster Backup UUID: 52b20efb-e15b-01b9-fb23-00110107009f
   Sub-Cluster UUID: 52ce6421-83d3-6415-5e51-3c6e6ee31b62
   Sub-Cluster Membership Entry Revision: 9
   Sub-Cluster Member UUIDs:
     52b228a4-2235-2422-1b19-00110107007f,
     52b20efb-e15b-01b9-fb23-00110107009f,
     52b20e01-7cd7-b9b5-b704-0011010700af,
     52b22966-99f2-93fa-81d9-00110107003f,
     52b20d6c-9920-87ad-5d4d-0011010700bf,
     52b228ae-8a0d-2756-bf5d-00110107005f,
     52b20dab-3f82-819c-c5df-0011010700df,
     52cc513c-c042-9524-79d8-00110107001f
   Sub-Cluster Membership UUID: 76eae352-1441-9b05-9845-00110107007f
```

从上面命令行的输出结果中我们可以看见的另外一条有用的信息是群集子成员的UUID。这个字段中一共有8个值，说明这是个8节点的群集。这条命令可以显示出每台主机都各自认为群集中有哪些节点，在对网络隔离问题进行排错时相当有用。

10.1.5 esxcli vsan maintenancemode

maintenancemode是一个有趣的命令行选项。你或许会觉得这条命令可以用来进入或退出维护模式，但事实并非如此。这个命令唯一能做的就是取消一个进行中的VSAN维护模式的操作。不过，这仍然是一条非常有用的命令，尤其是在下面这种情况下：当你决定将一台主机置于维护模式并已经选择了Full Data Migration（迁移全部数据）选项，然后想停止这个进行中的数据迁移（因为它需要很长时间）并改换成Ensure Access（确保可访问性）模式时。

```
~ # esxcli vsan maintenancemode
Usage: esxcli vsan maintenancemode {cmd} [cmd options]
Available Commands:
  cancel              Cancel an in-progress VSAN maintenance mode operation.
```

10.1.6 esxcli vsan policy

本书的前几章已经详细介绍了虚拟机存储策略，其中详细地讨论过当虚拟机部署的时候不选择任何虚拟机存储策略对虚拟机存储对象会产生什么影响。VSAN会给虚拟机存储对象关联上默认存储策略，下面这个esxcli vsan policy指令集是检查并更改默认存储策略

的一种方法。

```
~ # esxcli vsan policy
Usage: esxcli vsan policy {cmd} [cmd options]

Available Commands:
  cleardefault          Clear default VSAN storage policy values.
  getdefault            Get default VSAN storage policy values.
  setdefault            Set default VSAN storage policy values.

~ # esxcli vsan policy getdefault
Policy Class  Policy Value
------------  ------------------------------------------------------
cluster       (("hostFailuresToTolerate" i1))
vdisk         (("hostFailuresToTolerate" i1))
vmnamespace   (("hostFailuresToTolerate" i1))
vmswap        (("hostFailuresToTolerate" i1) ("forceProvisioning" i1))
```

这里可以看见构成一台 VSAN 数据存储上的虚拟机的各种虚拟机存储对象，还能看见默认策略的值。尽管策略值被称为 host failures to tolerate（允许的主机故障），它事实上等同于 vSphere Web 客户端中看见的 number of failures to tolerate（允许的故障数）。所有的对象都可以容忍群集中发生的至少一个故障并维持可用。类 vdisk 指的是虚拟机磁盘对象（VMDK），它也覆盖了快照增量。类 vmnamespace 就是虚拟机主页名字空间，其中存放着虚拟机的配置文件、元数据文件和日志文件。策略类 vmswap 当然就是虚拟机交换文件。关于 vmswap 最后要提醒的是它具有 forceProvisioning 属性，这意味着即使 VSAN 群集中没有足够的空间来满足 2 个虚拟机交换文件副本以容忍一个故障，VSAN 仍将置备虚拟机，只不过此时只能提供一个交换文件实例。关于强制置备的细节可以在本书的很多其他章节中找到。

如果你想要变更默认策略，帮助文件中包括大量的信息来介绍每一个策略。对默认策略进行配置的命令如下：

```
# esxcli vsan policy setdefault <-p|--policy> <-c|--policy-class>
```

诚然，我们并未覆盖默认策略中所有这 5 项策略设置，但是如果你希望这么做的话，当然完全可以把它们全部包括进来。不过正如前面多次重点提起的那样，更改默认策略必须谨慎，例如，给允许的故障数或闪存读取缓存预留设置了不合理的默认值可能会导致无法置备任何虚拟机的后果。从 esxcli vsan policy setdefault 命令的帮助输出结果中，可以显示更多策略设置的细节，展示如下。

- cacheReservation：为存储对象预留用作读取缓冲的闪存容量。显示为对象逻辑大小的一个百分比。仅用于解决读取的性能问题。预留的闪存容量不能用于其他对象，而未被预留的闪存被所有对象平均共享。其值表现为百万分之几，默认值为 0，最大值为 1000000。

- forceProvisioning：若此选项值为 yes，则即使在存储策略中的要求无法被群集中的现有资源满足的情况下，对象仍然会被置备。当要求的资源可用时，VSAN 会将对象置于合规状态。默认值为 No。
- hostFailuresToTolerate：定义了一个存储对象可以容忍的主机、磁盘或网络故障的数量。要容忍 n 个故障，就需要创建 n+1 个对象拷贝，并要求分布到 2n+1 台主机的存储上去。默认值为 1，最大值为 3。
- stripeWidth：每个存储对象副本条带化后横跨的磁盘数。这个数值如果大于 1 可能会带来更好的性能（例如当闪存读取缓冲未命中后需要直接由磁盘提供服务时），但是增加 stripeWidth 并不能保证性能提升。默认值为 1，最大值为 12。
- proportionalCapacity：是存储对象逻辑大小的一个百分比，用于表示虚拟机置备（厚置备）时预留的空间。存储对象的剩余部分则通过精简置备来提供。默认值为 0%，最大值为 100%。

另一个值得提起的用于 setdefault 命令的参数是 -c|--*policy-class* 选项。这就是要更改默认值的那个 VSAN 策略类。其选项包括 cluster、vdisk、vmnamespace 和 vmswap，命令中必须包含其中之一。

事实上，随着 VMware Horizon View 5.3.1 版本的发布，这个版本开始支持在 VSAN 上部署 View 桌面机，桌面存储对象通过默认策略进行部署。然而，这些默认值可能并非对 VSAN 上的每个 View 部署都是最合适的。尽管 VMware 强烈建议其客户在 Horizon View 5.3.1 中采用默认策略，客户可能还是需要基于其自身的需要（如第 8 章所讨论过的那样）来修改策略。默认策略可以通过以上所提及的 ESXCLI 命令行来进行修改。

最后要提一下 cluster，它是策略类选项之一，但是和 vmnamespace、vdisk 或 vmswap 不同，它不是一个虚拟机存储对象。这个选项用于包罗所有 VSAN 数据存储上部署的不属于虚拟机存储对象的那些部分。尽管我们不觉得有任何非虚拟机存储对象类型会放置在 VSAN 数据存储上，为了万一会用到这个参数，我们还是在这里先提一下。

10.1.7 esxcli vsan trace

esxcli vsan trace 是一个排错和诊断的工具，不应该在没有 VMware 全球支持服务（Global Support Services，简称 GSS）人员指导的情况下使用。它被设计用来捕获 VSAN 内部的一些诊断信息，用于进行进一步分析。我们在此列出其选项只是为了完整起见。

```
~ # esxcli vsan trace
Usage: esxcli vsan trace {cmd} [cmd options]

Available Commands:
  set        Configure VSAN trace. Please note: This command is not
             thread safe.

~ # esxcli vsan trace set -h
```

```
Usage: esxcli vsan trace set [cmd options]

Description:
  set                 Configure VSAN trace. Please note: This command is not
                      thread safe.

Cmd options:
  -f|--numfiles=<long>  Log file rotation for VSAN trace files.
  -p|--path=<str>       Path to store VSAN trace files.
  -r|--reset            When set to true, reset defaults for VSAN trace
                        files.
  -s|--size=<long>      Maximum size of VSAN trace files in MB.
```

10.1.8 用于 VSAN 排错的其他非 ESXCLI 命令

除了 esxcli vsan 系列命令之外，还有一些 ESXi 主机上的命令行指令可能对监控和排错有所帮助。

osls-fs

osls-fs 比起所有其他命令来说不仅仅是一个排错的指令。它可用于显示 VSAN 数据存储中的内容。这条命令不在搜索路径中，但是可以在下面列出的位置找到。在这条命令中，我们列举了 VSAN 数据存储中一台虚拟机目录中的内容。当 vSphere Web 客户端中的数据存储文件视图不能正确显示的时候，或是由于这样那样的原因被认为显示的信息不准确时，这条命令就非常有用了。

```
~ # cd /vmfs/volumes/vsanDatastore
~ # /usr/lib/vmware/osfs/bin/osfs-ls win2k8x64-1_1/

.fbb.sf
.fdc.sf
.pbc.sf
.sbc.sf
.vh.sf
.pb2.sf
.sdd.sf
.6cc6d752-a00c-ad67-268e-0010185def78.lck
win2k8x64-1.vmdk
win2k8x64-1.nvram
win2k8x64-1.vmx
win2k8x64-1.vmxf
win2k8x64-1.vmsd
win2k8x64-1_2.vmdk
.dvsData
.f6ccd752-5843-0da9-4149-0010185def78.lck
win2k8x64-1_1.vmdk
```

```
vmware.log
.d8dad752-a897-8b6a-8a96-0010185def78.lck
```

cmmds-tool

cmmds-tool 是另一个 ESXi 主机带有的有用的排错命令，可以用来显示很多 VSAN 信息。可以显示的信息包括配置、元数据、群集的状态、群集中主机的状态以及虚拟机存储对象的状态。很多其他上层诊断工具都利用了 cmmds-tool 收集到的信息。可以想象，它带有大量的选项。

```
~ # cmmds-tool
usage: cmmds-tool <cmd> <options>
commands:
   add                 Adds an entry from stdin. On successful exit the
                       entry is guaranteed to be in the directory
   delete              Deletes matching entries. On successful exit the
                       entry will be deleted from the directory
   dump                Dumps first matching entry to stdout
   find                Finds matching entries
   wait                Waits for a matching entry to appear
   waitdump            Waits for a matching entry to appear and dumps
                       the entry to stdout
   waitformembership   Waits for a membership entry to appear
   whoami              Get the node's uuid as used in the sub-cluster
   amimember           Check if I am the member in the current sub-cluster
   readdump            Reads a cmmds directory dump from a file
                       (specified with -d/--dumpfile) and
                       o/p to stdout in a given format specified
                       using -f option.
options:
   -o/--owner=<uuid>:          Entry owner
   -u/--uuid=<uuid>:           Entry uuid
   -t/--type=<int>|<name>:     Entry type
   -r/--rev=<int>:             Entry revision (-1 for latest)
   -i/--timeout=<int>:         Max time for wait (0 for infinite wait)
   -f/--format=<fmt>:          Output format (fmt should be one of
                               json/python/simple. Default is 'simple'
   -d/--dumpfile=<file>:       Filename to read the cmmds dump from.
   -p/--print-dump-hdr:        When CMMDS dump is read off the file, should
                               the dump file header be printed as well
   -v/--verbose=<int>:         Verbosity level
   -h/--help:                  Print this help text
```

find 选项可能是最有用的一个选项，尤其是当你想要收集某台虚拟机背后真实的存储对象的信息时。例如，可以看见某个特定对象的健康状态。在这个例子中，我们想要发现 UUID 是 52777432-f127-f001-d081-800a04cafb0e 的磁盘对象的额外的信息。

```
~ # cmmds-tool find -u 52777432-f127-f001-d081-800a04cafb0e
owner=52ca9e00-b362-a040-eb2a-984be1047ad4(Health: Healthy)
  uuid=52777432-f127-f001-d081-800a04cafb0e type=DISK rev=0
  [content = (1587068342272 1100 1+10000000 1200000000 1+0 13400000
  10 i10 i15 11600000 10 116777216 i0
  5214b1a5-b6b1-3a3a-f8b6-ee4ecc7a8d0e
  "52d7b4db-515e86a0-383a-001b21168828")], errorStr=(null)
owner=52ca9e00-b362-a040-eb2a-984be1047ad4(Health: Healthy)
  uuid=52777432-f127-f001-d081-800a04cafb0e type=HEALTH_STATUS rev=0
  [content = (i0 1319401449472)], errorStr=(null)
owner=52ca9e00-b362-a040-eb2a-984be1047ad4(Health: Healthy)
  uuid=52777432-f127-f001-d081-800a04cafb0e type=DISK_USAGE rev=645
  [content = (118874368 10 10 10 10)], errorStr=(null)
owner=52ca9e00-b362-a040-eb2a-984be1047ad4(Health: Healthy)
  uuid=52777432-f127-f001-d081-800a04cafb0e type=DISK_STATUS rev=645
  [content = (1430420688896 10 10 i10 18 10 1261888 10 10 10 10 10 10
  10 10 1+0 i4)], errorStr=(null)
```

这个命令确实还有很多其他选项可用。例如，-o <*owner*> 可以用来显示所有属主是 <*owner*> 的对象的信息。这可能会产生数量可观的输出结果。

Type 是另一个选项，它可用 -t 来指明。根据输出结果，可显示的类型包括 DISK、HEALTH_STATUS、DISK_USAGE 和 DISK_STATUS。其他类型还包括 DOM_OBJECT、DOM_NAME、POLICY、CONFIG_STATUS、HA_METADATA 和 HOSTNAME 等。

下面这条命令用于从一个 4 节点群集中列出主机名：

```
~ # cmmds-tool find -t HOSTNAME

owner=52caa324-d534-150c-d007-984be1047764(Health: Healthy)
  uuid=52caa324-d534-150c-d007-984be1047764 type=HOSTNAME rev=0
  [content = ("cs-tkmt-h03")], errorStr=(null)

owner=52ca9198-f7bf-3c3c-93c2-984be104893e(Health: Healthy)
  uuid=52ca9198-f7bf-3c3c-93c2-984be104893e type=HOSTNAME rev=0
  [content = ("cs-tkmt-h02")], errorStr=(null)

owner=52ca9e00-b362-a040-eb2a-984be1047ad4(Health: Healthy)
  uuid=52ca9e00-b362-a040-eb2a-984be1047ad4 type=HOSTNAME rev=0
  [content = ("cs-tkmt-h04")], errorStr=(null)
owner=52c6c45b-e7f4-31e0-7797-984be10a24d4(Health: Healthy)
  uuid=52c6c45b-e7f4-31e0-7797-984be10a24d4 type=HOSTNAME rev=0
  [content = ("cs-tkmt-h01")], errorStr=(null)
```

可以看出，这条命令非常强劲，它可以让你在 ESXi 主机上进行大量调查研究和排错工作。例如当一个大型群集中有一块磁盘发生了故障，这条命令可以用于发现哪个存储对象受到了这个故障的影响。再提醒一次，使用这条命令时需要小心，如果你有顾虑，可以选择在 VMware 支持人员的指导下使用。

vdq

vdq 命令有 2 种用途，它的确是 ESXi 主机上的一个非常好的排错工具。这个命令的第一个选项可以告诉你 ESXi 主机上的磁盘是否适用于 VSAN，如果不适用，又是什么原因造成的。

这条命令的第 2 个选项用于在 VSAN 启用之后，可以用它来显示磁盘映射信息，也就是显示哪个 SSD（或闪存设备）和哪些磁盘组合在一起组成了一个磁盘组。

首先，让我们运行这个命令来查询所有适用于 VSAN 使用的硬盘。第一个输出结果来自于一台不含有 VSAN 的主机。

```
~ # vdq -q
[
   {
      "Name"      : "naa.600508b1001c1184075bd1f8c2c882ec",
      "VSANUUID"  : "",
      "State"     : "Ineligible for use by VSAN",
      "Reason"    : "Has partitions",
      "IsSSD"     : "0",
      "IsPDL"     : "0",
   },

   {
      "Name"      : "naa.6000d3100046c500000000000000010",
      "VSANUUID"  : "",
      "State"     : "Ineligible for use by VSAN",
      "Reason"    : "Has partitions",
      "IsSSD"     : "0",
      "IsPDL"     : "0",
   },

]
```

前面这个例子中，没有硬盘可以用于 VSAN，因为它们都已经含有分区。下面的输出结果来自于一台已经启用了 VSAN 的主机：

```
~ # vdq -q
[
   {
      "Name"      : "mpx.vmhba32:C0:T0:L0",
      "VSANUUID"  : "",
      "State"     : "Ineligible for use by VSAN",
      "Reason"    : "Has partitions",
      "IsSSD"     : "0",
      "IsPDL"     : "0",
   },

   {
```

```
        "Name"     : "eui.48f8681115d6416c00247172ce4df168",
        "VSANUUID" : "52b0a0a9-a4f5-93f7-91e0-c52287f74668",
        "State"    : "In-use for VSAN",
        "Reason"   : "None",
        "IsSSD"    : "1",
        "IsPDL"    : "0",
    },

    {
        "Name"     : "naa.600508b1001c530aff02e0c5c7971e1d",
        "VSANUUID" : "52236e03-b7f5-10a7-88e3-f0b41fe207a8",
        "State"    : "In-use for VSAN",
        "Reason"   : "Non-local disk",
        "IsSSD"    : "0",
        "IsPDL"    : "0",
    },

]
```

可以看出，有 2 个硬盘已经被 VSAN 声明，第 3 个硬盘不符合条件，因为它已经含有分区。在这个例子中，不符合条件的磁盘是 ESXi 主机的引导磁盘。这条命令还指明了哪个硬盘是 SSD（IsSSD）以及那个硬盘处于永久性设备丢失状态（IsPDL）。

这条命令第 2 个有用的选项是列出所有 VSAN 磁盘的映射，换而言之，哪些闪存设备和哪些磁盘在同一个磁盘组中。下面是一个输出的例子（包含了 -H 参数以使输出结果更易让人读懂）：

```
~ # vdq -i -H
Mappings:
   DiskMapping[0]:
           SSD:  eui.48f8681115d6416c00247172ce4df168
            MD:  naa.600508b1001c530aff02e0c5c7971e1d
```

这条命令显示了 SSD 和磁盘（MD）之间的关系。如果你想要从命令行了解某台特定主机上磁盘组的布局，这条命令就非常有用了，尤其是当你在那台 ESXi 主机上配置了多个磁盘组的时候，它可以快速告诉你哪块 SSD 位于哪些磁盘的前端。

尽管本节介绍的一些命令都相当有用，但它们都是在某一台 ESXi 主机的基础上来检查和监控 VSAN 的，管理员肯定还需要一些可以让他们从群集整体的角度来进行检查的工具。VMware 早在开发 VSAN 时就非常清楚这一点，因此扩展了 Ruby vSphere Console（RVC）来提供 VSAN 的完整的群集角度的视图。下节将深入探讨 RVC。

10.2　Ruby vSphere Console

上一节介绍了用于 VSAN 的基于 ESXi 主机的命令行，本节将介绍一款工具，它使

你可以从群集的角度来进行 VSAN 的管理。VMware vCenter Server 5.5 U1 包含一个新的组件叫做 Ruby vSphere Console（RVC），它也可以在 VMware vCenter Virtual Appliance（VCVA）中找到。如同介绍中所描述的那样，RVC 是一个可编程的接口，允许管理员来查询 vCenter、群集、主机、存储和网络的状态。就 VSAN 而言，存在相当多的可编程扩展程序来显示你对某个 VSAN 群集想了解的几乎所有内容。本节将介绍 RVC 中的这些 VSAN 扩展。

可以通过 RVC 连接到任何 vCenter Server。在 VCVA 上，可以通过 Secure Shell（SSH）登录并运行 `rvc <user>@<vc-ip>` 命令。

在基于 Windows 的 vCenter 环境中，需要打开命令行界面并导航到 c:\Program Files\VMware\Infrastructure\VirtualCenter Server\support\rvc。在这里可以找到 rvc.bat 文件。你或许需要编辑这个文件，添加上可以用于登录到 vCenter Server 的适当的账号（默认账号是 Administrator@localhost）。账号设置完成后，只需要简单地运行 rvc.bat，输入密码，就可以连通了。

登录后，你会看见一个虚拟文件系统，vCenter Server 实例位于其根目录中。现在可以用诸如 cd 和 ls 这种命令来四处打量了，也可以用 tab 来快速补充完整命令行。这个文件系统的结构模拟了 vSphere Client 中的清单目录树。因此，可以运行 `cd <vCenter Server>`，然后输入 `cd <datacenter>`。可以用 ~ 来指代当前的 datacenter，其下所有的群集都位于“Computers”目录下。注意，当你在文件夹 / 目录中四处浏览时，列出的内容都带有一些数值。这些数值可能也能用作快捷方式。例如，在下面的例子中，vCenter 目录中只有一个 datacenter，它关联了一个数字 0，我们可以 cd 到 0，而不需要输入这个 datacenter 的全名。

```
> ls
0 /
1 mia-cg07-vc01/
> cd 1
/mia-cg07-vc01> ls
0 mia-cg07-dc01 (datacenter)
/mia-cg07-vc01> cd 0
/mia-cg07-vc01/mia-cg07-dc01> ls
0 storage/
1 computers [host]/
2 networks [network]/
3 datastores [datastore]/
4 vms [vm]/
```

10.2.1 VSAN 命令

若想学习任何命令，可以输入 `<command> -help`，也可以使用命令 `help` 和 `help <command-namespace>`（例如 `help vm` 或 `help vm.ip`）来学习更多命令。因为我们

主要对 VSAN 感兴趣，让我们来看看对 VSAN 监控和排错到底有哪些可用的命令吧。

```
/mia-cg07-vc01/mia-cg07-dc01> help vsan
Commands:
enable_vsan_on_cluster: Enable VSAN on a cluster
disable_vsan_on_cluster: Disable VSAN on a cluster
cluster_change_autoclaim: Enable VSAN on a cluster
host_consume_disks: Consumes all eligible disks on a host
host_wipe_vsan_disks: Wipes content of all VSAN disks on a host
host_info: Print VSAN info about a host
cluster_info: Print VSAN info about a cluster
disks_info: Print physical disk info about a host
cluster_set_default_policy: Set default policy on a cluster
object_info: Fetch information about a VSAN object
disk_object_info: Fetch information about all VSAN objects on a given
physical disk
cmmds_find: CMMDS Find
fix_renamed_vms: This command can be used to rename some VMs which get
  renamed by the VC in case of storage inaccessibility. It is possible
  for some VMs to get renamed to vmx file path. eg.
  "/vmfs/volumes/vsanDatastore/foo/foo.vmx". This command will rename
  this VM to "foo". This is the best we can do. This VM may have been
  named something else but we have no way to know. In this best effort
  command, we simply rename it to the name of its config file (without
  the full path and .vmx extension of course!).
vm_object_info: Fetch VSAN object information about a VM
disks_stats: Show stats on all disks in VSAN
whatif_host_failures: Simulates how host failures impact VSAN resource
  usage
observer: Run observer
resync_dashboard: Resyncing dashboard
vm_perf_stats: VM perf stats
enter_maintenance_mode: Put hosts into maintenance mode
lldpnetmap: Gather LLDP mapping information from a set of hosts
check_limits: Gathers (and checks) counters against limits
object_reconfigure: Reconfigure a VSAN object
obj_status_report: Print component status for objects in the cluster.
apply_license_to_cluster: Apply license to VSAN
check_state: Checks state of VMs and VSAN objects
reapply_vsan_vmknic_config: Unbinds and rebinds VSAN to its vmknics
recover_spbm: SPBM Recovery

To see commands in a namespace: help namespace_name
To see detailed help for a command: help namespace_name.command_name
```

这里显示的所有命令都需要以 vsan 开头。因此，要运行 enable_vsan_on_cluster，你必须用命令 vsan.enable_vsan_on_cluster。记得吗？命令行是可以补完整的，所以每条命令你都只需要键入开头的几个字符然后用 Tab 按键来补完剩下的字

符（或者显示符合你当前键入命令的所有可选命令）。

当然，还有一组 RVC 命令也是 VSAN 管理员感兴趣的。这组命令是 SPBM 命令集，可以用于任何关于虚拟机的存储策略。下面列出了 RVC 中的 SPBM 命令：

```
/mia-cg07-vc01/mia-cg07-dc01> help spbm
Commands:
profile_delete: Delete a VM Storage Profile
profile_apply: Apply a VM Storage Profile. Pushed profile content to
  Storage system
profile_create: Create a VM Storage Profile
device_change_storage_profile: Change storage profile of a virtual disk
check_compliance: Check compliance
namespace_change_storage_profile: Change storage profile of VM
  namespace
vm_change_storage_profile: Change storage profile of VM namespace and
  its disks
device_add_disk: Add a hard drive to a virtual machine

To see commands in a namespace: help namespace_name
To see detailed help for a command: help namespace_name.command_name
```

稍后我们会回来探讨 SPBM，但现在我们先来研究一下 RVC 中的 VSAN 命令行，了解它们对监控、排错和解决 VSAN 的问题有何帮助。你必须用 vsan. 开头来运行所有相关的 VSAN 命令。

若要对某个特定的命令获得更多的帮助，只需要简单地在命令前面加上 `help` 即可。

enable_vsan_on_cluster 和 disable_vsan_on_cluster

这两条命令如其字面意思所说的那样——用于启用或禁用群集上的 VSAN。除此之外唯一的方法就是通过 vSphere Web 客户端来完成，你无法用 ESXCLI 实现同样的功能。用 ESXCLI 可以将 ESXi 主机加入或退出群集，扩展 VSAN 的规模或是缩小之，但是无法在群集层面上启用或禁用 VSAN 服务。如下面的命令所示（其中数字 0 代表 ls 输出中的 0 指代的那个群集）：

```
/mia-cg07-vc01/mia-cg07-dc01/computers> ls
0 cg07-cluster01 (cluster): cpu 126 GHz, memory 696 GB
/mia-cg07-vc01/mia-cg07-dc01/computers> vsan.disable_vsan_on_cluster 0
```

重新启用 VSAN 同样简单直接：

```
/mia-cg07-vc01/mia-cg07-dc01/computers> vsan.enable_vsan_on_cluster 0
```

host_info

大多数情况下，导航到你感兴趣的特定对象是非常容易的。你可以轻松地在命令行中用主机的数字快捷方式导航到相应位置，然后使用 `vsan.host_info` 命令，如下面的例子所示：

```
/mia-cg07-vc01/mia-cg07-dc01> ls
0 storage/
1 computers [host]/
2 networks [network]/
3 datastores [datastore]/
4 vms [vm]/
/mia-cg07-vc01/mia-cg07-dc01> cd 1
/mia-cg07-vc01/mia-cg07-dc01/computers> ls
0 cg07-cluster01 (cluster): cpu 126 GHz, memory 697 GB
/mia-cg07-vc01/mia-cg07-dc01/computers> cd 0
/mia-cg07-vc01/mia-cg07-dc01/computers/cg07-cluster01> ls
0 hosts/
1 resourcePool [Resources]: cpu 126.58/126.58/normal, mem 697.07/697.07/
normal
/mia-cg07-vc01/mia-cg07-dc01/computers/cg07-cluster01> cd 0
/mia-cg07-vc01/mia-cg07-dc01/computers/cg07-cluster01/hosts> ls
0 mia-cg07-esx011.vmwcs.com (host): cpu 2*8*2.39 GHz, memory 103.00 GB
1 mia-cg07-esx012.vmwcs.com (host): cpu 2*8*2.39 GHz, memory 103.00 GB
2 mia-cg07-esx013.vmwcs.com (host): cpu 2*8*2.39 GHz, memory 103.00 GB
3 mia-cg07-esx014.vmwcs.com (host): cpu 2*8*2.39 GHz, memory 103.00 GB
4 mia-cg07-esx015.vmwcs.com (host): cpu 2*8*2.39 GHz, memory 103.00 GB
5 mia-cg07-esx016.vmwcs.com (host): cpu 2*8*2.39 GHz, memory 103.00 GB
6 mia-cg07-esx017.vmwcs.com (host): cpu 2*8*2.39 GHz, memory 103.00 GB
7 mia-cg07-esx018.vmwcs.com (host): cpu 2*8*2.39 GHz, memory 103.00 GB
/mia-cg07-vc01/mia-cg07-dc01/computers/cg07-cluster01/
  hosts> vsan.host_info 0
VSAN enabled: yes
Cluster info:
  Cluster role: agent
  Cluster UUID: 52ce6421-83d3-6415-5e51-3c6e6ee31b62
  Node UUID: 52b20dab-3f82-819c-c5df-0011010700df
  Member UUIDs: ["52b228a4-2235-2422-1b19-00110107007f",
    "52b20efb-e15b-01b9-fb23-00110107009f", "52b20e01-7cd7-b9b5-b70
4-0011010700af", "52b22966-99f2-93fa-81d9-00110107003f",
    "52b20d6c-9920-87ad-5d4d-0011010700bf", "52b228ae-8a0d-2756-bf5
d-00110107005f", "52b20dab-3f82-819c-c5df-0011010700df",
    "52cc513c-c042-9524-79d8-00110107001f"]
Storage info:
  Auto claim: no
  Disk Mappings:
    SSD: Local ATA Disk (naa.50025385a00c5085) - 119 GB
    MD: SEAGATE Serial Attached SCSI Disk (naa.5000c5006900c7cf) - 1117 GB
NetworkInfo:
  Adapter: vmk6 (10.7.7.11)
```

在这个例子中，我们先进入主机文件夹然后运行命令，命令行中使用了数字 0，它代表

着列表中的第一台主机 mia-cg07-vc01.vmwcs.com。这句命令显示的是关于群集、网络和存储的综合信息，从中可以看见群集角色和成员、显示了一块磁盘和一块 SSD 的磁盘映射关系，以及在这台主机上用于 VSAN 流量的 VMkernel 适配器相关的网络信息。如果用 esxcli 命令行的话需要很多命令才能显示同样的结果。看，RVC 功能的强大马上就显现出来了。

host_consume_disks 和 host_wipe_vsan_disks

接下来的命令用来让 VSAN 群集中某台特定主机声明所有未声明的空的本地硬盘。这通过 host_consume_disk 命令来实现。显然，这用于在手工模式下创建 VSAN；如果 VSAN 被配置成自动模式，空的本地硬盘会自动声明。

然而，如果你给一台 VSAN 主机添加了新的硬盘，而且这些硬盘曾经用于其他用途，已经存在数据或分区信息，那么可以用 host_wipe_vsan_disks 命令来创建新的干净的空白的可被 VSAN 利用的硬盘。

cluster_info

下面这条命令是我们介绍的第一条可以提供 VSAN 群集的集中视图的命令。要获取群集配置的全景，从这条命令开始可能是最佳选择。再一次，进入 RVC 清单列表的群集对象中并输入命令 vsan.cluster_info，或者简单地在运行命令时键入群集对象的全路径即可。这条命令等同于你到群集中的每台主机上去运行 vsan.host_info 命令。前面我们说起过 vsan.host_info 可以显示群集、主机、磁盘和网络信息，现在一条命令就可以显示每一台主机的这些信息。我们不会在下面列举这个 8 节点 VSAN 群集中所有 8 台主机的信息，但是从下面截取的输出中你也应该可以略知一二了。

```
/mia-cg07-vc01/mia-cg07-dc01/computers> vsan.cluster_info 0
Host: mia-cg07-esx011.vmwcs.com
  VSAN enabled: yes
  Cluster info:
    Cluster role: agent
    Cluster UUID: 52ce6421-83d3-6415-5e51-3c6e6ee31b62
    Node UUID: 52b20dab-3f82-819c-c5df-0011010700df
    Member UUIDs: ["52b228a4-2235-2422-1b19-00110107007f",
      "52b20efb-e15b-01b9-fb23-00110107009f", "52b20e01-7cd7-b9b5-b
704-0011010700af", "52b22966-99f2-93fa-81d9-00110107003f",
      "52b20d6c-9920-87ad-5d4d-0011010700bf", "52b228ae-8a0d-2756-b
f5d-00110107005f", "52b20dab-3f82-819c-c5df-0011010700df",
      "52cc513c-c042-9524-79d8-00110107001f"]
  Storage info:
    Auto claim: no
    Disk Mappings:
      SSD: Local ATA Disk (naa.50025385a00c5085) - 119 GB
      MD: SEAGATE Serial Attached SCSI Disk
        (naa.5000c5006900c7cf) - 1117 GB
```

```
    NetworkInfo:
      Adapter: vmk6 (10.7.7.11)

Host: mia-cg07-esx012.vmwcs.com
    VSAN enabled: yes
    Cluster info:
      Cluster role: agent
      Cluster UUID: 52ce6421-83d3-6415-5e51-3c6e6ee31b62
      Node UUID: 52b20d6c-9920-87ad-5d4d-0011010700bf
      Member UUIDs: ["52b228a4-2235-2422-1b19-00110107007f",
        "52b20efb-e15b-01b9-fb23-00110107009f", "52b20e01-7cd7-b9b5-b
704-0011010700af", "52b22966-99f2-93fa-81d9-00110107003f",
        "52b20d6c-9920-87ad-5d4d-0011010700bf", "52b228ae-8a0d-2756-b
f5d-00110107005f", "52b20dab-3f82-819c-c5df-0011010700df",
        "52cc513c-c042-9524-79d8-00110107001f"]
    Storage info:
      Auto claim: no
      Disk Mappings:
        SSD: Local ATA Disk (naa.50025385a00c5084) - 119 GB
        MD: SEAGATE Serial Attached SCSI Disk (naa.5000c500690142df)
          - 1117 GB
NetworkInfo:
    Adapter: vmk6 (10.7.7.12)
```

这里你可以再次看见群集角色和成员、显示了磁盘和 SSD 的磁盘映射关系，以及在这台主机上用于 VSAN 流量的 VMkernel 适配器相关的网络信息。我们相信你会同意这条命令对于获取群集配置的概览非常有用。

disks.info

如前所述，VSAN 只会声明那些本地的空白的（换而言之不存在分区表的）磁盘。如果你发现 VSAN 由于某些原因没有声明一块磁盘，`disks.info` 是一条很好的命令，可以用来检查为什么无法声明磁盘。这条命令可以显示磁盘是否已经被 VSAN 使用、是否可被 VSAN 利用，或是否不符合条件无法被 VSAN 使用。通常来说，如果磁盘不符合条件，例如已分过区，`disks.info` 会显示出分区信息。输出结果可能会相当长，这是因为要显示某些磁盘标识符或这些磁盘在某些情况下的状态，因此我们在本书中就不列出这条命令的输出结果了。

cluster_set_default_policy

本书前面已经非常深入地讨论过默认策略的细节。在前面 ESXCLI 的章节，我们事实上已经交待过默认策略设置的一些命令。你或许还记得我们说过如果要用 ESXCLI 命令行方式来更改默认策略，你必须到每台主机上一台一台地修改。好吧，现在这条命令可以让我们一次性地将群集中所有主机上的默认策略都修改掉。假设默认策略没能满足你的要求，或许你希望默认策略具有一个更高数值的允许的故障数，又或者希望条带宽度更大，又或

许你希望在默认策略中预留一些读取缓存，在这些场合下，你可以使用 cluster_set_default_policy 命令。首先，你需要创建一个希望当成默认策略的策略。这可以在使用策略的任何虚拟机的 vmprofiles 文件夹下获得（配置文件 [profiles] 是策略 [policy] 先前的名称），然后你可以通过 cluster_set_default_policy 命令指向那个策略，将那条策略变成新的默认策略，并用于没有特别指定虚拟机存储策略的任何虚拟机。

object_reconfigure

你可以对现已存在的 VSAN 对象使用 RVC 命令 vsan.object_reconfigure 来改变当前的虚拟机存储策略并分配给它们一个新的策略。如果你觉得有必要给某个特殊的对象增加允许的故障数或增大条带宽度，这条命令可以帮到你。

object_info、disk_object_info 和 vm_object_info

这 3 条 RVC 命令放在一起介绍是因为它们在某种程度上相互关联，都和显示群集中对象的具体信息有关，分别针对磁盘、群集和虚拟机对象。让我们从 vm_object_info 命令说起，因为它在这 3 个命令之中可能更具有教育性。要使用这条命令，首先导航到某一台虚拟机并将该虚拟机作为命令参数。下面的示例中，我们选择了一台只含有单台虚拟机的主机并对其使用该命令。然后 RVC 会检查群集中所有的 ESXi 主机来找出哪台 ESXi 主机含有与此虚拟机相关的对象。此时此刻，你应该很清楚这样一个事实：VSAN 很可能不会把虚拟机的存储对象和虚拟机的计算资源放在同一台主机上。事实上，虚拟机存储对象（包括副本）很可能位于和其计算资源所处的主机完全不同的群集中的一台主机上。

下面的输出结果为了可读性的原因做了删节：

```
/mia-cg07-vc01/mia-cg07-dc01/computers/cg07-cluster01/hosts/mia-cg07-esx011.vmwcs.com> cd vms
/mia-cg07-vc01/mia-cg07-dc01/computers/cg07-cluster01/hosts/mia-cg07-esx011.vmwcs.com/vms> ls
0 test-vm-default-policy: poweredOff
/mia-cg07-vc01/mia-cg07-dc01/computers/cg07-cluster01/hosts/mia-cg07-esx011.vmwcs.com/vms> vsan.vm_object_info 0
2014-02-08 11:15:16 +0000: Fetching VSAN disk info from mia-cg07-esx011.vmwcs.com (may take a moment) ...
2014-02-08 11:15:17 +0000: Fetching VSAN disk info from mia-cg07-esx012.vmwcs.com (may take a moment) ...
VM test-vm-default-policy:
  Namespace directory
    DOM Object: 0c03ed52-e50b-b174-4eea-0011010700df (owner: mia-cg07-esx015.vmwcs.com, policy: hostFailuresToTolerate = 1)
      Witness: 0c03ed52-8b83-33b6-898a-0011010700df (state: ACTIVE (5), host: mia-cg07-esx013.vmwcs.com, md: naa.5000c50067d8d9f7, ssd: naa.50025385a00a376a)
      RAID_1
        Component: 0c03ed52-2aa3-32b6-36a8-0011010700df (state: ACTIVE (5), host: mia-cg07-esx015.vmwcs.com, md: naa.5000c5006900c843, ssd: naa.50025385a00c508e)
        Component: 0c03ed52-7457-31b6-20ba-0011010700df (state: ACTIVE (5), host: mia-cg07-esx017.vmwcs.com, md: naa.5000c500690143d7, ssd: naa.50025385a00c5088)
  Disk backing: [vsanDatastore] 0c03ed52-e50b-b174-4eea-0011010700df/test-vm-default-policy.vmdk
    DOM Object: 1203ed52-5649-8be2-c1ec-0011010700df (owner: mia-cg07-esx013.vmwcs.com, policy: hostFailuresToTolerate = 1, proportionalCapacity = 100)
      Witness: 1203ed52-424b-3e11-89a7-0011010700df (state: ACTIVE (5), host: mia-cg07-esx016.vmwcs.com, md: naa.5000c5006901415b, ssd: naa.50025385a00c5090)
      RAID_1
        Component: 1203ed52-df00-3d11-dbd1-0011010700df (state: ACTIVE (5), host: mia-cg07-esx012.vmwcs.com, md: naa.5000c500690142df, ssd: naa.50025385a00c5084)
        Component: 1203ed52-1e00-3b11-8d8e-0011010700df (state: ACTIVE (5), host: mia-cg07-esx013.vmwcs.com, md: naa.5000c50067d8d9f7, ssd: naa.50025385a00a376a)
```

前面的输出结果中列出了 2 个虚拟机对象：虚拟机主页名字空间目录和虚拟机磁盘 / VMDK。这 2 个存储对象都具备 RAID-1 副本配置，说明它们关联了允许的故障数的策略设置。由于每个对象都只有 2 个组件，我们可以假设这台虚拟机的策略中允许的故障数被设成了 1。这个结论可以通过查看每个对象的 DOM 对象那一行来进行验证，它明确地显示出了策略设置。没错，我们确实可以看见这两者的允许的故障数都设成了 1。虽然这个描述

符说的是允许的主机故障（hostFailuresToTolerate），它其实也涵盖了其他组件例如磁盘和网络。另一个有意思的地方是这个例子中的 VMDK 的 proportionalCapacity 是 100，意味着它在 VSAN 数据存储上是厚置备的，而不是默认的精简置备的。

当然，这个输出结果中最有用的部分可能还是这些组件存放在哪里的具体位置信息。你可以看见主机、磁盘 ID 和磁盘组中面向磁盘的 SSD 的 ID。而且，你最终还可以看见这些组件是否健康。在这个例子中，所有组件都显示为 Active（活动）状态，也就是正常的。

最后，我们还可以获得见证组件的信息。在本书的这个阶段本该无须再多作解释，只需要知道它们在故障发生的时候起了很重要的作用就可以了。当群集发生故障时，它们参与到投票中，使得存储对象可以满足超过半数的要求并保持继续可用。

现在我们已经检查过了关联到虚拟机的对象的信息，有时候也需要知道哪些对象 / 组件实际上位于哪块物理磁盘上。这就是 disk_object_info 命令发挥用处的时候了。通过这条命令，可以显示被 VSAN 声明的物理磁盘里面的内容并让 RVC 显示出这块磁盘中的组件。这条命令是针对群集发起的，不过要求提供磁盘标识符作为额外的参数。这个参数很容易获取，前面的 vm_object_info 命令就已经提供了这个信息，你也可以通过 ESXCLI 或 vSphere 用户图形界面来获得。如果我们使用了从前面的命令中获得的磁盘的 NAA ID，就可以显示出该磁盘驱动器上的所有对象和组件。

还有，这是另一条提供了很多有用信息的命令。在这个例子中，我们切换到了另一个不同的 VSAN 群集来显示输出的多样性。下面这个命令的输出由于可读性的原因已经做了删节，这次我们留下了“DOM Objects”：

```
/localhost/ie-datacenter-01/computers> ls
0 ie-vsan-cluster-01 (cluster): cpu 109 GHz, memory 331 GB
/localhost/ie-datacenter-01/computers> vsan.disk_object_info 0 naa.600508b1001c36662525d1d217c882f87
2014-03-14 11:37:57 +0000: Fetching VSAN disk info from hosts (may take a moment) ...
2014-03-14 11:38:00 +0000: Done fetching VSAN disk infos
Physical disk naa.600508b1001c3662525d1d217c882f87 (525a5dee-1f35-b3b5-cac9-c55d047ea601):
  DOM Object: 16782053-64de-7b11-a685-001517a69c72 (owner: 10.27.51.2, policy: hostFailuresToTolerate = 1, proportionalCapacity =
100)
    Context: Part of VM ie-vdpa-01: Disk: [vsanDatastore] 2e742053-2895-3151-e630-001517a69c72/ie-vdpa-01_2.vmdk
    Witness: 17782053-0097-fb35-a525-001517a69c72 (state: ACTIVE (5), host: 10.27.51.2, md: naa.600508b1001cdb46f505bf98eef9e9b4,
ssd: eui.c68e151fed8a4fcf0024712c7cc444fe)
    Witness: 17782053-9c0b-fa35-cf2a-001517a69c72 (state: ACTIVE (5), host: 10.27.51.2, md:
**naa.600508b1001c3662525d1d217c882f87**, ssd: eui.c68e151fed8a4fcf0024712c7cc444fe)
    Witness: 17782053-2abf-fa35-44b9-001517a69c72 (state: ACTIVE (5), host: 10.27.51.4, md: naa.600508b1001c8119ca0ab3a6fe0d2b19,
ssd: eui.a15eb52c6f4043b5002471c7886acfaa)
    RAID_1
      RAID_0
        Component: 17782053-b020-f935-eb15-001517a69c72 (state: ACTIVE (5), host: 10.27.51.3, md:
naa.600508b1001c784579103bf9baf41797, ssd: eui.d1ef5a5bbe864e27002471febdec3592)
        Component: 17782053-744b-f835-a4ca-001517a69c72 (state: ACTIVE (5), host: 10.27.51.3, md:
naa.600508b1001c034feb6ff0871db13c4b, ssd: eui.d1ef5a5bbe864e27002471febdec3592)
      RAID_0
        Component: 17782053-2a68-f735-e951-001517a69c72 (state: ACTIVE (5), host: 10.27.51.4, md:
naa.600508b1001c1380cc97bc5345465552, ssd: eui.a15eb52c6f4043b5002471c7886acfaa)
        Component: 17782053-b807-f635-1568-001517a69c72 (state: ACTIVE (5), host: 10.27.51.2, md:
naa.600508b1001cdb465505bf98eef9e9b4, ssd: eui.c68e151fed8a4fcf0024712c7cc444fe)
```

这个输出结果的美妙之处在于它显示出了对象及其上下级关系，换而言之，你可以知道特定组件是哪个虚拟机存储对象的一部分。不仅如此，它还能显示组成那个存储对象的其他组件，以及这些其他组件在 VSAN 群集中的位置。它还通过把 ** 放置在磁盘标识符的

前面和后面，将其突出显示出来。这个磁盘标识符就是我们提供给前面的命令作为参数使用的。

从这个输出结果中我们可以推导出这块我们感兴趣的磁盘包含相当多的组件。其中一些组件是虚拟机磁盘 /VMDK 的一部分，其他组件则用来组成虚拟机的名字空间目录，另一些则可能是虚拟机交换文件的一部分。这个输出结果对于排错极为有用，因为它可以让你看见磁盘上组件的状态。这个输出结果中所有的组件都显示为好的、正常的和活动的状态。

接下来最后要介绍的是 object_info 命令。它现在使用真实的 VSAN 对象标识符来显示存储对象的状态。因此，无须查看隶属于一台虚拟机的所有对象，或是某块磁盘上的所有组件，我们也可以直接查看群集中的单个存储对象的详细信息了，无论这是一个名字空间目录还是一块虚拟机磁盘。这对于追踪某个特定对象的状态非常有用，尤其是从日志文件显示的相关消息中获得了一个对象 ID，而你又无法确定这个对象属于哪个虚拟机或哪个磁盘时。在下面的例子中，我们选择了一块虚拟机磁盘的 ID，并运行了下面的命令，命令中用到了群集和对象 ID 作为参数。

```
/localhost/ie-datacenter-01/computers> vsan.object_info 0 d4502253-e81d-00b8-6351-0010185def78
DOM Object: d4502253-e81d-00b8-6351-0010185def78 (owner: 10.27.51.3, policy: hostFailuresToTolerate = 1, proportionalCapacity = [0, 100])
  Witness: d5502253-10e6-321a-656d-0010185def78 (state: ACTIVE (5), host: 10.27.51.1, md: naa.600508b1001c530aff02e0c5c7971e1d, ssd:
eui.48f8681115d6416c00247172ce4df168)
  RAID_1
    Component: d5502253-3461-321a-d275-0010185def78 (state: ACTIVE (5), host: 10.27.51.3, md: naa.600508b1001c034feb6ff0871db13c4b, ssd:
eui.d1ef5a5bbe864e27002471febdec3592)
    Component: d5502253-b00b-311a-ff45-0010185def78 (state: ACTIVE (5), host: 10.27.51.2, md: naa.600508b1001c3662525d1d217c882f87, ssd:
eui.c68e151fed8a4fcf0024712c7cc444fe)
  Extended attributes:
    Address space: 43285303296B (40.31 GB)
    Object class: vdisk
    Object path: /vmfs/volumes/vsan:524ed5e9ee77b654-b7895b53dcad7c5e/c2cd2153-24a4-45a5-b0a1-001517a69c72/hbrdisk.RDID-7b82ee6b-b12e-4364-b72c-
c0ba2a4101d0.12.61153313595688.vmdk
```

这给了我们一个组成此对象的不错的详细视图。我们发现它带有 RAID-1 配置和 2 个组件，说明它的策略设置中包含允许的故障数为 1。所有这些可以在输出结果中 DOM Object 这一行得到验证——包括 hostFailuresToTolerate = 1。我们还可以发现对象类型是 vdisk，说明这是一块虚拟机磁盘（VMDK）。再一次，这条命令显示了关于每个组件的有用位置信息以及所有组件都处于良好的 Active（活动）状态的信息。

通过使用 object_info 命令，管理员应该可以追踪到每个不同的存储对象都属于群集中的哪个虚拟机，并验证它们是否处于良好的工作状态。如果发生了问题，需要管理员来找出组件并进行诊断和排错，这些命令是无价之宝。

cmmds_find

cmmds_find 命令极为有用，尤其是当一台 ESXi 主机贴出了一条报错信息，其中带有一个特定 VSAN 对象的 ID 时，你可以用这条命令来找出这个对象究竟和谁相关。vsan.cmmds_find（cmmds 指群集、监控、成员和目录服务）命令可用于显示 VSAN 群集中某个特定对象的信息。

-t DISK_USAGE

要显示群集中磁盘使用率的信息时可以用这个选项。其中最重要的结果是 Health 这一列，它将直接告诉你磁盘是否处于不健康状态。

```
/localhost/vsphere5.5-u1> vsan.cmmds_find ~/computers/vsphere5.5-u1/ -t DISK_USAGE
+---+------------+--------------------------------------+--------------+---------+----------------------------------------+
| # | Type       | UUID                                 | Owner        | Health  | Content                                |
+---+------------+--------------------------------------+--------------+---------+----------------------------------------+
| 1 | DISK_USAGE | 52777432-f127-f001-d081-800a04cafb0e | 10.27.51.4   | Healthy | {"capacityReserved"=>18874368,         |
|   |            |                                      |              |         |  "iopsReserved"=>0,                    |
|   |            |                                      |              |         |  "throughPutReserved"=>0,              |
|   |            |                                      |              |         |  "12CacheReserved"=>0,                 |
|   |            |                                      |              |         |  "11CacheReserved"=>0}                 |
| 2 | DISK_USAGE | 5214b1a5-b6b1-3a3a-f8b6-ee4ecc7a8d0e | 10.27.51.4   | Healthy | {"capacityReserved"=>0,                |
|   |            |                                      |              |         |  "iopsReserved"=>0,                    |
|   |            |                                      |              |         |  "throughPutReserved"=>0,              |
|   |            |                                      |              |         |  "12CacheReserved"=>0,                 |
|   |            |                                      |              |         |  "11CacheReserved"=>0}                 |
| 3 | DISK_USAGE | 5269ded0-91a2-4986-4d5e-e946d252730f | 10.27.51.3   | Healthy | {"capacityReserved"=>14680064,         |
|   |            |                                      |              |         |  "iopsReserved"=>0,                    |
|   |            |                                      |              |         |  "throughPutReserved"=>0,              |
|   |            |                                      |              |         |  "12CacheReserved"=>0,                 |
|   |            |                                      |              |         |  "11CacheReserved"=>0}                 |
| 4 | DISK_USAGE | 52f226e9-7664-34a1-b6be-3f27c4d2671e | 10.27.51.3   | Healthy | {"capacityReserved"=>7509901312,       |
|   |            |                                      |              |         |  "iopsReserved"=>0,                    |
|   |            |                                      |              |         |  "throughPutReserved"=>0,              |
|   |            |                                      |              |         |  "12CacheReserved"=>0,                 |
|   |            |                                      |              |         |  "11CacheReserved"=>0}                 |
| 5 | DISK_USAGE | 5251dfbe-2106-16ca-e6eb-c8389a7eebdd | 10.27.51.3   | Healthy | {"capacityReserved"=>0,                |
|   |            |                                      |              |         |  "iopsReserved"=>0,                    |
|   |            |                                      |              |         |  "throughPutReserved"=>0,              |
|   |            |                                      |              |         |  "12CacheReserved"=>0,                 |
|   |            |                                      |              |         |  "11CacheReserved"=>0}                 |
| 6 | DISK_USAGE | 52de5de0-ff71-02fa-b671-558abd3280e1 | 10.27.51.4   | Healthy | {"capacityReserved"=>25165824,         |
|   |            |                                      |              |         |  "iopsReserved"=>0,                    |
|   |            |                                      |              |         |  "throughPutReserved"=>0,              |
|   |            |                                      |              |         |  "12CacheReserved"=>0,                 |
|   |            |                                      |              |         |  "11CacheReserved"=>0}                 |
+---+------------+--------------------------------------+--------------+---------+----------------------------------------+
```

-t DISK

`-t DISK` 选项相当有用，因为它会显示群集中所有硬盘的详细信息。显示结果中的 Content 列带有一些有趣的属性，包括容量以及硬盘是否是一块 SSD（`isSsd`）。

-u UUID

`-u UUID` 选项使得管理员可以获得特定组件的更详细的信息。

从前面的输出中可以看出，还有不少其他的类型，包括 HEALTH_STATUS 和 DISK_STATUS。这些都可以通过 `-t` 选项传递给 `vsan.cmmds_find` 命令。

我们仅仅只是稍微介绍了一下这条命令的皮毛，用多了以后你就会注意这条命令非常像 ESXCLI 里面的 `cmmds-tool` 命令。没错，很多 `cmmds-tool find` 的选项也可以用于这条命令。

fix_renamed_vms

关于 `fix_renamed_vms` 命令没有太多可以解释的。如其字面意思那样，vCenter

```
+-------------------------+---------+-----------+------+----------+
| mia-cg07-esx015.vmwcs.com | 1     | 1117.75 GB | 2 %  | 2 %      |
| mia-cg07-esx017.vmwcs.com | 1     | 1117.75 GB | 2 %  | 2 %      |
| mia-cg07-esx011.vmwcs.com | 1     | 1117.75 GB | 0 %  | 0 %      |
| mia-cg07-esx014.vmwcs.com | 1     | 1117.75 GB | 0 %  | 0 %      |
| mia-cg07-esx018.vmwcs.com | 1     | 1117.75 GB | 0 %  | 0 %      |
| mia-cg07-esx016.vmwcs.com | 1     | 1117.75 GB | 0 %  | 0 %      |
| mia-cg07-esx012.vmwcs.com | 1     | 1117.75 GB | 4 %  | 4 %      |
| mia-cg07-esx013.vmwcs.com | 1     | 1117.75 GB | 4 %  | 4 %      |
+-------------------------+---------+-----------+------+----------+

Simulating 1 host failures:
The command shows current VSAN disk usage, but also simulates how
disk usage would evolve under a host failure. Concretely the simulation
assumes that all objects would be brought back to full policy
compliance by bringing up new mirrors of existing data.
The command makes some simplifying assumptions about disk space
balance in the cluster. It is mostly intended to do a rough estimate
if a host failure would drive the cluster to being close to full.

+-----------------------------------------+-----------------------+
| Host with most data on it:              |mia-cg07-esx013.vmwcs.com |
| Data to be newly mirrored:              |40.76 GB               |
| Capacity before failure:                |8815.06 GB free, 1% used  |
| Capacity after failure, before re-mirroring:|7738.08 GB free, 1% used  |
| Capacity after failure, after re-mirroring: |7697.31 GB free, 2% used  |
+-----------------------------------------+-----------------------+
```

如同这条命令的输出结果中说的那样，模拟一台主机故障背后的原因是为了检查一旦发生故障时，如果要重建虚拟机存储对象以满足策略要求，是否会发生任何磁盘驱动器容量不足的情况。这是条非常有用的命令，可用来判断 VSAN 在发生一个故障时是否能继续满足对虚拟机的可用性要求。它对于容量规划也是非常有用的，因为如果一台主机故障意味着策略合规性不再能满足，那么再发生一个故障就可能导致虚拟机不可用。这个故障可能是一个主机、磁盘或网络故障。

在前面的输出结果中，群集中有 8 台主机，但是每台都只贡献了一块磁盘。然而，这个群集的利用率非常低，所以一台主机的故障对虚拟机存储对象所需占用的磁盘空间容量的影响是非常小的，甚至可以忽略不计。在这个群集上距离填满磁盘空间还很远呢。

接下来我们看看 vsan.disks_stats 命令。这条命令基本上是用来告诉你从对象的角度来看群集在磁盘之间的负载均衡好不好。它也可以显示关于可用磁盘空间和已消耗的磁盘空间的信息。它显示的其他有用信息还包括性能——例如读 / 写操作和延迟。不过，性能问题会在下一节中更详尽地探讨。

让我们来看一看输出结果的一些例子，首先是一个全群集的视图。为了方便阅读我们对下面的输出结果做了删节，移除了后 5 台主机的输出结果。

```
/mia-cg07-vc01/mia-cg07-dc01/computers> vsan.disks_stats 0
+----------------------+---------------------------+------+------+------------+------+----------+-------+----------+
|                      |                           |      | Num  | Capacity   |      |          |       | Device   |
| DisplayName          | Host                      | isSSD| Comp | Total      | Used | Reserved | Used  | Latencies|
+----------------------+---------------------------+------+------+------------+------+----------+-------+----------+
| naa.50025385a00c5085 | mia-cg07-esx011.vmwcs.com | SSD  | 0    | 83.47 GB   | 0 %  | 0 %      | 0r/0w | 0d/0q/0k |
| naa.5000c5006900c7cf | mia-cg07-esx011.vmwcs.com | MD   | 5    | 1117.75 GB | 0 %  | 0 %      | 0r/0w | 0d/0q/0k |
+----------------------+---------------------------+------+------+------------+------+----------+-------+----------+
| naa.50025385a00c5084 | mia-cg07-esx012.vmwcs.com | SSD  | 0    | 83.47 GB   | 0 %  | 0 %      | 0r/0w | 0d/0q/0k |
| naa.5000c500690142df | mia-cg07-esx012.vmwcs.com | MD   | 4    | 1117.75 GB | 4 %  | 4 %      | 0r/0w | 0d/0q/0k |
+----------------------+---------------------------+------+------+------------+------+----------+-------+----------+
| naa.50025385a00a376a | mia-cg07-esx013.vmwcs.com | SSD  | 0    | 83.47 GB   | 0 %  | 0 %      | 0r/0w | 0d/0q/0k |
| naa.5000c50067d8d9f7 | mia-cg07-esx013.vmwcs.com | MD   | 4    | 1117.75 GB | 4 %  | 4 %      | 0r/0w | 0d/0q/0k |
+----------------------+---------------------------+------+------+------------+------+----------+-------+----------+
```

可以看出组件的分布相对统一，横跨了群集中的所有主机和所有磁盘。VSAN 自己的初始放置算法控制着这一切。它会列出每一台主机、主机上的每一块磁盘和 SSD 以及组件计数器、容量和各种性能计数器。除了可以对群集运行这条命令外，我们还可以对某台主机来运行这条命令（通过进入到主机对象上），但是这不会显示更多额外的信息。

这条命令对于验证群集是否均衡极其有用。这就是 VMware 在其最佳实践中强烈建议将 VSAN 群集中的 ESXi 主机配置成完全一致的原因。这可以将组件最优地均衡分布到群集中的所有组件和磁盘上去。如果在一组不同的 ESXi 主机上组建 VSAN 群集，就可能导致一个不均衡的 VSAN 群集，这样其中某些主机的负担就可能比其他重，并可能会引起性能问题。

最后这条统计信息命令是 `vsan.vm_perf_stats`。当将某台虚拟机作为一个参数来运行这条命令时，它会提供每台虚拟机的每秒输入 / 输出操作数（IOPS）、吞吐量和延时信息。这对于调查特定单台虚拟机的性能问题极为有用。这条命令会收集 2 组性能信息，每隔 20 秒一次。然后输出样本时间段内的性能平均值。不过，VSAN 通过 VSAN Observer 工具提供了一个查看性能信息的更好方法，后面马上就会详细介绍这个工具。

resync.dashboard

`resync.dashboard` 是条非常有用的命令，用它可以检查 VSAN 中的存储对象的重新同步状态以及同步的速度有多快。很多情况下会发生重新同步，包括发生了主机故障需要在群集中其他地方重建（存储）组件，或者的确发生了策略变更需要重新创建一个新的对象并和老的对象同步。还记得吗？对虚拟机存储策略的变更是可以在线进行的。此时新的存储布局要求的新副本被创建出来并和原先已存在的组件进行同步。当同步完成时，原先的组件就会被丢弃。所有这些都发生在虚拟机开机并持续运行着的同时。

`vsan.resync_dashboard` 命令是在 RVC 中对群集对象发起的。它会显示所有正在重新同步的对象，并会显示还在等待同步的剩余的字节数。下面的例子来自于我们的一个群集，这里我们将一台主机置于维护模式并请求进行完全数据迁移，这意味着 VSAN 需要

将进入维护模式的那台主机上的所有组件和对象都迁移到群集中的另外一台 ESXi 主机上。我们可以看见一个对象正在被同步并还剩下 0.18GB 字节的数据等待同步。

```
2014-02-10 12:02:47 +0000: Querying all VMs on VSAN ...
2014-02-10 12:02:47 +0000: Querying all objects in the system ...
2014-02-10 12:02:48 +0000: Got all the info, computing table ...
+-----------------------------------------+-----------------+---------------+
| VM/Object                               | Syncing objects | Bytes to sync |
+-----------------------------------------+-----------------+---------------+
| vm-ftt-of-1-sw-of-2                     | 1               |               |
|    [vsanDatastore] vm-ftt-of-1-sw-of-2.vmx |               | 0.18 GB       |
+-----------------------------------------+-----------------+---------------+
| Total                                   | 1               | 0.18 GB       |
+-----------------------------------------+-----------------+---------------+
```

enter_maintenance_mode

可以用 `enter_maintenance_mode` 命令来将参与 VSAN 群集的 ESXi 主机置入维护模式。关于维护模式可用选项的更多信息请参考第 7 章。

lldpnetmap

对于网络问题的诊断和排错，`lldpnetmap` 是一个极为有用的 RVC 命令。链路层发现协议（LLDP，Link Layer Discovery Protocol）是一个供应商中立的链路层协议，它被网络设备用来广播自己的能力。对于熟悉思科 CDP（Cisco Discovery Protocol）的管理员来说，这是一个和供应商无关的中立的等价的协议。对 VSAN 群集运行这条命令，可以显示群集中所有主机的 LLDP 信息（如果它们存在的话）。

apply_license_to_cluster

`apply_license_to_cluster` 是另一条简单的 RVC 命令，可以用它来给 VSAN 群集添加许可证——如果你不想通过 vSphere Web 客户端的话。它有一个选项 `--license-key`。

check_limits

当 VSAN 群集中虚拟机数量较大的时候，除了 `vsan.check_state` 之外，我们建议定期运行 `check_limits` 命令。在本书的很前面需求一章中，我们讨论过 VSAN 的很多限制条件。其中之一就是在 VSAN 群集中的一台 ESXi 主机上可以存在的组件数量。最初的 VSAN 发布版本这个数量是 3000。`vsan.check_limits` 命令的输出结果同时显示了现存的组件数量和数量限制。VSAN 的另一个限制是每台主机上的 RDP（可靠数据报协议）可以承载的接口连接数（number of socket connections），这个信息和当前在用的接口总数一起在结果中显示出来。出于更佳的阅读体验考虑，我们对此条命令的输出结果进行了删节。

```
/mia-cg07-vc01/mia-cg07-dc01/computers> vsan.check_limits 0
2014-02-08 13:10:41 +0000: Gathering stats from all hosts ...
2014-02-08 13:10:43 +0000: Gathering disks info ...
2014-02-08 13:10:43 +0000: Fetching VSAN disk info from <hosts> (may take a moment)
+---------------------------+------------------+-------------------------+
| Host                      | RDT              | Disks                   |
+---------------------------+------------------+-------------------------+
| mia-cg07-esx011.vmwcs.com | Assocs: 13/20000 | Components: 6/3000      |
|                           | Sockets: 15/10000| naa.5000c5006900c7cf: 0% |
|                           | Clients: 0       | naa.50025385a00c5085: 0% |
|                           | Owners: 2        |                         |
| mia-cg07-esx012.vmwcs.com | Assocs: 5/20000  | Components: 4/3000      |
|                           | Sockets: 7/10000 | naa.50025385a00c5084: 0% |
|                           | Clients: 0       | naa.5000c500690142df: 3% |
|                           | Owners: 0        |                         |
| mia-cg07-esx013.vmwcs.com | Assocs: 16/20000 | Components: 5/3000      |
|                           | Sockets: 14/10000| naa.5000c50067d8d9f7: 3% |
|                           | Clients: 0       | naa.50025385a00a376a: 0% |
|                           | Owners: 2        |                         |
+---------------------------+------------------+-------------------------+
```

reapply_vsan_vmknic_config

reapply_vsan_vmknic_config 命令在 ESXi 主机的某个特定的 VMkernel 网卡上重新配置 VSAN 网络。当物理网络层面发生变化并需要 VSAN 网络识别出这些变化的时候，这条命令就很有用。下面的输出显示的是输入命令后的结果：

```
/localhost/ie-datacenter-01/computers/ie-vsan-cluster-01/hosts>
vsan.reapply_vsan_vmknic_config
Host: 10.27.51.2
  Reapplying config of vmk2:
    AgentGroupMulticastAddress: 224.2.3.4
    AgentGroupMulticastPort: 23451
    IPProtocol: IPv4
    InterfaceUUID: 80381f53-bc26-2968-3cf2-001517a69c72
    MasterGroupMulticastAddress: 224.1.2.3
    MasterGroupMulticastPort: 12345
    MulticastTTL: 5
  Unbinding VSAN from vmknic vmk2 ...
  Rebinding VSAN to vmknic vmk2 ...
```

recover_spbm

假设你失去了管理 VSAN 群集的 vCenter Server，在部署一台新的 vCenter Server 后，剩下的任务是如何重建虚拟机存储策略。但是怎样才能找回曾经的策略设置呢？recover_spbm 命令真的可以帮到你。

首先，它可以显示 VSAN 上的哪些虚拟机没有配置策略。这句话的意思是这些虚拟机的策略没有在 vCenter 中定义过。在可能的情况下，虚拟机会继续遵照它们原本的策略设

置，即使 vCenter Server 不再可用。一旦这些虚拟机（它们运行着 vCencer 中不存在的策略）被 recover_spbm 命令所发现，你就可以用一些命令选项来恢复它们。

```
/localhost/ie-datacenter-01/computers/ie-vsan-cluster-01/hosts>
vsan.recover_spbm 0
2014-03-14 12:23:02 +0000: Fetching Host info
2014-03-14 12:23:02 +0000: Fetching Datastore info
2014-03-14 12:23:02 +0000: Fetching VM properties
2014-03-14 12:23:02 +0000: Fetching policies used on VSAN from CMMDS
2014-03-14 12:23:03 +0000: Fetching SPBM profiles
2014-03-14 12:23:03 +0000: Fetching VM <-> SPBM profile association
2014-03-14 12:23:03 +0000: Computing which VMs do not have a SPBM Profile
...
2014-03-14 12:23:03 +0000: Fetching additional info about some VMs
2014-03-14 12:23:03 +0000: Got all info, computing after 0.76 sec
2014-03-14 12:23:03 +0000: Done computing

Found 0 missing SPBM Profiles.
Found 0 entities not associated with their SPBM Profiles.
```

在前面的输出结果中，没有虚拟机是运行在未定义的策略之下。vCenter Server 拥有一个完整的被所有虚拟机使用着的策略列表。还存在很多额外的命令选项，例如查看丢失策略的细节、重建丢失的策略，以及自动关联策略和虚拟机。当真的发生 vCenter Server 故障或有需要将 VSAN 迁移到一台新的 vCenter Server 时，这是一条非常有用的 RVC 命令。

10.2.2 SPBM 命令

在 RVC 中另外一组非常有用的命令是 SPBM（基于存储策略的管理）命令。可以想象，这些是作用于存储策略的非常有用的命令。

下面的输出结果显示的是这些命令列表，其中大多数相当直接。

```
> help spbm
Commands:
profile_delete: Delete a VM Storage Profile
profile_apply: Apply a VM Storage Profile. Pushed profile content to
Storage system
profile_create: Create a VM Storage Profile
device_change_storage_profile: Change storage profile of a virtual disk
check_compliance: Check compliance
namespace_change_storage_profile: Change storage profile of VM namespace
vm_change_storage_profile: Change storage profile of VM namespace and its
disks
device_add_disk: Add a hard drive to a virtual machine

To see commands in a namespace: help namespace_name
To see detailed help for a command: help namespace_name.command_name
```

在 VSAN 中，还有很多可用于检查 SPBM 设置的 RVC 命令。现在，你知道要在 VSAN

上部署一台虚拟机，必须先给虚拟机创建一个虚拟机存储策略，它可以用来定义虚拟机磁盘的镜像副本的数量（允许的故障数）或 VMDK 的条带宽度。SPBM 是控制 VSAN 这个方面的底层技术。下面让我们来看一下 RVC 中的 SPBM 扩展。

首先来看一看 SPBM 扩展，在 RVC 中一共有 8 个，它们的名字字面意思直白，相当易于理解：

```
spbm.check_compliance
spbm.profile_apply
spbm.device_add_disk
spbm.profile_create
spbm.device_change_storage_profile
spbm.profile_delete
spbm.namespace_change_storage_profile
spbm.vm_change_storage_profile
```

如果你在 RVC 中导航到一台虚拟机，就可以对单个虚拟机或设备使用这些命令。让我们来看一些例子。首先我要查的是一台名叫 win1 的特定虚拟机的合规性。

```
/localhost/CH-Datacenter/vms> ls
0 Discovered virtual machine/
1 VMware vCenter Operations Manager: cpu 0.00/-0.00/normal, mem 0.00/-0.00/
normal
2 win1: poweredOn
3 win2: poweredOn
4 win3: poweredOn
5 win4: poweredOn
6 win5: poweredOn
7 win6: poweredOn
8 vSphere Data Protection 5.5: poweredOn

/localhost/CH-Datacenter/vms> spbm.check_compliance 2
+-----------------+---------+------------+
| VM/Virtual Disk | Profile | Compliance |
+-----------------+---------+------------+
| win1            | FT=1    | compliant  |
|   Hard disk 1   | FT=1    | compliant  |
+-----------------+---------+------------+
Number of 'compliant' entities: 2
```

下一步是应用一个新的配置文件。这个配置文件可以在 ~/storage/vmprofiles 下找到。在这个例子中，我们有 2 个可用的配置文件：

```
/localhost/CH-Datacenter> ls storage/vmprofiles/
0 FT=1
1 FT=1,SW=2
```

现在我们要用 spbm.vm_change_storage_profile 命令把其中一台虚拟机的配置文件从 FT=1（允许的故障数 =1）更改为 FT=1,SW=2（允许的故障数 =1 且磁盘对象的带

数 =2），如下所示：

```
/localhost/CH-Datacenter/vms> ls
0 Discovered virtual machine/
1 VCops-VM: cpu 0.00/-0.00/normal, mem 0.00/-0.00/normal
2 win1: poweredOn
3 win2: poweredOn
4 win3: poweredOn
5 win4: poweredOn
6 win5: poweredOn
7 win6: poweredOn
8 VDP-VM: poweredOn
/localhost/CH-Datacenter/vms> spbm.vm_change_storage_profile 2 -p ~/
storage/vmprofiles/FT=1,SW=2/
ReconfigVM win1: success
```

当然，重新配置需要花一点时间。通过输入 spbm.check_compliance 命令，可以观察到属于这台刚刚更改过存储策略的虚拟机的硬盘 1 的状态现在变成了 noncompliant（不合规）：

```
/localhost/CH-Datacenter/vms> spbm.check_compliance 2
+-----------------+------------+--------------+
| VM/Virtual Disk | Profile    | Compliance   |
+-----------------+------------+--------------+
| win1            | FT=1,SW=2  | compliant    |
|   Hard disk 1   | FT=1,SW=2  | nonCompliant |
+-----------------+------------+--------------+
Number of 'compliant' entities: 1
Number of 'nonCompliant' entities: 1
```

当然，我们还可以用非常有用的 vsan.resync_dashboard 命令来查看当重新配置进行的时候，还有多少数据仍然在同步中（输出结果被截短了并移除了虚拟机 ID）。

```
/localhost/CH-Datacenter/computers> ls
0 CH-Cluster (cluster): cpu 86 GHz, memory 45 GB
/localhost/CH-Datacenter/computers> vsan.resync_dashboard 0
2013-12-12 16:56:58 +0000: Querying all VMs on VSAN ...
2013-12-12 16:56:58 +0000: Querying all objects in the system from 10.20.177.18 ...
2013-12-12 16:56:59 +0000: Got all the info, computing table ...
+-----------------------------------------+----------------+-------------+
| VM/Object                               | Syncing objects | Bytes to sync |
+-----------------------------------------+----------------+-------------+
| win1                                    | 1              |             |
| [vsanDatastore] <vmid>/win1-000001.vmdk |                | 48.00 GB    |
+-----------------------------------------+----------------+-------------+
| Total                                   | 1              | 48.00 GB    |
+-----------------------------------------+----------------+-------------+
```

可以重复运行这个命令，当 Bytes To Sync 变成 0 时，所有内容就都同步完了。我相信你会同意，这条命令相当有用。

最后说一句：如果你想使用的 SPBM 命令必须要以一个“设备”作为参数，你必须使用在 ~/vms/device/ 中找到的磁盘设备。

10.2.3　用于 VSAN 的 PowerCLI

我们知道，很多客户都对 vCloud 套件中的任务自动化非常感兴趣。对 VSAN 也一样。为了尽快让你用上自动化，VMware 研发中心以 fling 的形式发布了一组 VSAN PowerCLI 命令集（cmdlets）。

一旦从 VMware fling 网站（flings.vmware.com）下载这个模块并正确安装之后，一组 VSAN 的命令集就可以使用了。

针对 VSAN 的新命令集如下：

- Get-VsanDisk
- Get-VsanDiskGroup
- New-VsanDisk
- New-VsanDiskGroup
- Remove-VsanDisk
- Remove-VsanDiskGroup
- New-Cluster
- Set-Cluster
- New-VMHostNetworkAdapter
- Set-VMHostNetworkAdapter

表 10-1 展示了如何使用这些命令集的例子。

表 10-1　用于 VSAN 的 PowerCLI 命令集

任务	命令（Cmdlet）
从一台特定的主机查询磁盘 mpx.vmhba2:C0:T1:L0	`Get-VsanDisk -VMHost MyVMHost -CanonicalName "mpx.vmhba2:C0:T1:L0"`
查询含有规范名称为 mpx.vmhba2:C0: T2:L1 的 VSAN 磁盘组	`Get-VsanDiskGroup -VMHost MyVMHost -CanonicalName "mpx.vmhba2:C0:T2:L1"`
获取磁盘组信息并向其中添加一个磁盘	`$dg = Get-VsanDiskGroup -VMHost MyVMHost -CanonicalName "mpx.vmhba2:C0:T2:L1"` `$d = New-VsanDisk -VsanDiskGroup $dg -CanonicalName "mpx.vmhba3:C0:T2:L0"`
创建一个新的 VSAN 磁盘组，其中包含一个磁盘和一个 SSD	`New-VsanDiskGroup -VMHost MyVMHost -SolidStateCanonicalName "mpx.vmhba2:C0:T1:L0" -HardDiskCanonicalName "mpx.vmhba3:C0:T1:L0"`

从 VSAN 磁盘组中移除一个磁盘 mpx.vmhba3:C0:T2:L0	`$dg = Get-VsanDiskGroup -VMHost MyVMHost -CanonicalName "mpx.vmhba2:C0:T2:L1"`
	`Get-VsanDisk -VsanDiskGroup $dg -CanonicalName "mpx.vmhba3:C0:T2:L0" |`
	`Remove-VsanDisk`

从主机移除一个磁盘组	`$dg = Get-VsanDiskGroup -VMHost MyVMHost -CanonicalName "mpx.vmhba2:C0:T2:L1"` `Remove-VsanDiskGroup -VsanDiskGroup $dg`

10.3 VSAN 和 SPBM API

很多对自动化感兴趣的管理员可能也会对 VMware 公开的 VSAN 和 SPBM 的应用程序可编程接口（API）感兴趣。针对 VSAN 和 SPBM 的特定 API 可以作为 vSphere 5.5 API 的一部分来获得。如果愿意的话，它可以让你通过可编程的方式来访问很多 VSAN 操作并将任务自动化。尽管这可能并非对所有人都有用，为了完整起见，我们还是决定将这部分内容包括进来。然而，我们强烈建议对通过这些 API 来开发自动化应用感兴趣的管理员去 API 说明规范中获取进一步的细节。

和 RVC 一样，你应该知道 VSAN 有 2 类主要的操作：

- 针对 VSAN 的操作。
- 针对虚拟机存储策略的操作。

让我们来看一些例子。

10.3.1 启用 / 禁用 VSAN（自动声明）

使用 `ReconfigureComputeResource_Task()` 并设置 `spec->vsanConfig->enabled` 成为 true 或 false，并将 `spec->vsanConfig->defaultConfig->autoClaimStorage` 设置成 true。

10.3.2 手工磁盘声明

每台 ESXi 主机都在 `configManager->vsanSystem` 提供了一个 vsanSystem 管理器，通过下列方法来提供磁盘管理：

- `AddDisks_Task()`
- `InitializeDisks_Task()`
- `QueryDisksForVsan()`
- `QueryHostStatus()`
- `RemoveDisk_Task()`
- `RemoveDiskMapping_Task()`
- `UpdateVsan_Task()`

10.3.3 更改虚拟机存储策略

下列命令可以用来为虚拟机主页和虚拟磁盘更改虚拟机存储策略：

使用 `ReconfigVM_Task()` 并将 `spec->vmProfile` 设置成虚拟机存储策略 ID。

也可以仅用来更改虚拟机磁盘的虚拟机存储策略：

使用 `ReconfigVM_Task()` 并将 `spec->deviceChange->device` 设成特定的虚拟机磁盘以重新配置并将 `spec->deviceChange->vmProfile` 设置成虚拟机存储策略 ID。

10.3.4　进入维护模式

使用 EnterMaintenanceMode_Task() 并将 spec->maintenanceSpec->vsanMode->objectAction 设置成特定的数据访问模式。

10.3.5　在 VSAN 数据存储中创建和删除目录

可以使用 DatastoreNamespaceManager，它提供了下列 2 种方法：

- CreateDirectory()
- DeleteDirectory()

10.3.6　CMMDS

CMMDS 是用于访问底层 CMMDS 对象和磁盘管理 API 的内置的 VSAN 管理器。

每台 ESXi 主机都提供了一个 vsanInternalSystem 管理器（configManager->vsanInternalSystem），它提供了用于访问 VSAN 底层系统的以下方法：

- QueryCmmds()
- QueryObjectsOnPhysicalVsanDisk()
- QueryPhysicalVsanDisks()
- QueryVsanObjects()

10.3.7　SPBM

对于虚拟机存储策略而言，VSAN 利用了 SPBM 框架。它允许管理员创建策略来定义特定的存储功能，例如性能和可靠性，并应用于某个虚拟机。SPBM API 在 vCenter Server 中表现为一个独立的 API 端点，如果你想利用 VSAN 存储的功能创建额外的虚拟机存储策略，就会需要用到它。表 10-2 列出了某些 SPBM API 调用。

表 10-2　SPBM API 调用

任务	SPBM API 调用
查询已定义的可用虚拟机存储配置文件列表	PbmQueryProfile()
创建虚拟机存储策略	PbmCreate()
删除虚拟机存储策略	PbmDelete()
检查虚拟机存储策略的合规性	PbmCheckCompliance()
给定虚拟机主页或虚拟磁盘，查询其关联的虚拟机存储配置文件	PbmQueryAssociatedEntity()
给定虚拟机存储配置文件，查询其关联的虚拟机主页或虚拟磁盘	PbmQueryAssociatedProfiles()

10.4　在 ESXi 上对 VSAN 进行诊断排错

到目前为止，我们已经介绍了很多以群集为中心的工具，例如 RVC 工具。尽管我们的

确讨论过一些可以给管理员在 ESXi 主机上使用的 ESXCLI 命令行，你还需要知道去哪里才能找到错误信息和日志文件。本节会特别关注要监控的日志文件，以及你可能会用到的诊断 VSAN 的其他一些工具。

10.4.1 日志文件

你可以在 ESXi 主机的以下位置找到 VSAN 日志文件，如表 10-3 所示。

表 10-3 VSAN 日志文件的位置

日志文件描述	日志文件位置
CLOM（群集级别对象管理器日志）	/var/log/clomd.log
OSFS（作为一个文件系统，代表 VSAN 对象存储）	/var/log/osfsd.log
vCenter/ESXi 通信	/var/log/hostd.log /var/log/vpxa.log
VSAN 供应商提供器	/var/log/vsan/vpd.log
ESXi 日志	/var/log/vmkernel.log /var/log/vobd.log /var/log/vmkwarning.log

你还可以针对 VSAN 的主要软件组件例如 LSOM、RDT、DOM 和 PLOG 来搜寻参考信息。GSS（VMware 全球支持服务）建议在对 VSAN 问题进行排错的时候，先搜索 VMkernel 的日志文件，来找到含有这些关键字的项。如果你不熟悉这些软件组件，请重读第 5 章，其中提供了这些软件组件及其扮演的角色的详细信息。

10.4.2 VSAN Trace 工具

我们曾在本章的 ESXCLI 一节简要提起过 trace 工具。VSAN 使用一种压缩的二进制 trace 格式来记录每个 I/O 的多条日志消息。trace 存放在 /var/log/vsantraces/ 目录下。这些 trace 不是人类可读懂的格式，在查看前必须先释放出来。要解码 VSAN trace 成人类可读懂的“日志消息”格式，可以在 ESXi 主机上运行以下命令：

```
# cd /var/log/vsantraces/
# zcat <file>.gz | /usr/lib/vmware/vsan/bin/vsanTraceReader.py > <file>.txt
```

命令执行后，*<file>*.txt 中会包含 trace 的人类可读懂的格式。

10.4.3 VSAN VMkernel 模块和驱动程序

ESXi 5.5 发布版中预置了构建 VSAN 群集需要的组件，无须额外将 VIB 或软件组件添加到主机，就可以成功创建一个 VSAN 群集并构建起一个可横向扩展的 VSAN 数据存储。

VSAN 成功配置后，你会观察到新的 VMkernel 模块被加载进来，用来进行 VSAN 的实施。这些 VMkernel 模块的名字是 vsan、rdt、plog 和 lsomcommon。

当虚拟机进行写操作时，写会先进入 SSD，随后 VSAN 会有规律地将 SSD 中的内容回写到磁盘中。plog 模块实现了 VSAN 的电梯算法，它会检查磁盘的物理布局并决定何时将 SSD 中的内容冲刷进磁盘。

vsan 模块可被认为是同时用于 LSOM 和 DOM 组件的模块，因为这些模块相互交织在一起，lsomcommon 也包含这些组件的共享代码。

rdt 模块即可靠数据报传输（Reliable Datagram Transport）模块，它负责跨群集的 VSAN 通信。

10.5　性能监控

管理存储的最重要的一个方面是要有能力进行监控并对性能问题进行诊断和排错。VSAN 也一样。本节我们将与你分享 vSphere 管理员可以用来监控并对 VSAN 性能相关问题进行排错的各种工具。

10.5.1　用于 VSAN 的 ESXTOP 性能计数器

在 VSAN 的最初发布版本中，`esxtop` 中没有专用于 VSAN 数据存储的性能计数器。不过，当你想要在 ESXi 主机上检查虚拟机活动、VMDK 性能、主机状态、内存使用以及磁盘活动时，它仍然是一个非常有用的工具。`esxtop` 很容易使用，在 ESXi 主机的 shell 提示符下，简单地键入 **`esxtop`** 即可。图 10-1 显示了一些 `esxtop` 输出结果的例子。

```
 1:33:20pm up 49 days 15:04, 514 worlds, 0 VMs, 0 vCPUs; CPU load average: 0.01, 0.01, 0.02
PCPU USED(%): 1.3 3.9 1.3 0.5 1.5 0.3 0.5 0.1 4.1 0.4 0.3 0.1 0.4 0.5 0.2 0.4 AVG: 1.0
PCPU UTIL(%): 2.9 4.2 3.2 1.4 2.8 0.7 1.3 0.5 4.4 0.9 0.8 0.5 0.9 1.0 0.6 1.1 AVG: 1.7
CORE UTIL(%): 6.6     3.6     3.2     1.5     4.9     0.9     1.4     1.3     AVG: 2.9

    ID      GID NAME             NWLD  %USED  %RUN  %SYS  %WAIT %VMWAIT  %RDY  %IDLE %OVRLP  %CSTP  %MLMTD  %SWPWT
21789709 21789709 esxtop.10111864    1   3.82  3.66  0.00  95.41      -  0.00   0.00   0.00   0.00   0.00   0.00
  2019     2019 hostd.100001493    26   3.79  3.63  0.00 2572.30      -  0.03   0.00   0.00   0.00   0.00   0.00
     2        2 system             121   2.39  3.46  0.00 11985.29      -  0.60   0.00   0.00   0.00   0.00   0.00
  2465     2465 sh.1000015170       1   0.48  0.49  0.00   98.58      -  0.00   0.00   0.00   0.00   0.00   0.00
16577438 16577438 cmmdsd.10085114    2   0.36  0.14  0.26  198.00      -  0.00   0.00   0.03   0.00   0.00   0.00
     8        8 helper             166   0.30  0.30  0.00 16447.55      -  0.03   0.00   0.00   0.00   0.00   0.00
  1324     1324 vmkiscsid.10000     2   0.13  0.03  0.11  198.13      -  0.01   0.00   0.00   0.00   0.00   0.00
  3161     3161 vpxa.1000015525    11   0.07  0.11  0.00 1089.66      -  0.01   0.00   0.02   0.00   0.00   0.00
  2097     2097 rhttpproxy.1000     8   0.06  0.08  0.00  792.46      -  0.05   0.00   0.01   0.00   0.00   0.00
  1686     1686 vmware-usbarbit     2   0.04  0.04  0.00  198.11      -  0.00   0.00   0.00   0.00   0.00   0.00
16577352 16577352 clomd.100851145    1   0.03  0.03  0.00   99.03      -  0.00   0.00   0.00   0.00   0.00   0.00
21789616 21789616 sshd.1011186413    1   0.03  0.03  0.00   99.03      -  0.00   0.00   0.00   0.00   0.00   0.00
  1216     1216 net-lacp.100001    3   0.02  0.03  0.00  297.20      -  0.02   0.00   0.00   0.00   0.00   0.00
    10       10 ft                  5   0.02  0.02  0.00  495.39      -  0.00   0.00   0.00   0.00   0.00   0.00
     9        9 drivers            13   0.01  0.02  0.00 1288.04      -  0.01   0.00   0.00   0.00   0.00   0.00
  3537     3537 openwsmand.1000     3   0.01  0.02  0.00  297.19      -  0.01   0.00   0.00   0.00   0.00   0.00
  1867     1867 logchannellogge     1   0.01  0.01  0.00   99.06      -  0.00   0.00   0.00   0.00   0.00   0.00
   891      891 vmsyslogd.10000     4   0.01  0.01  0.00  396.34      -  0.00   0.00   0.00   0.00   0.00   0.00
  4185     4185 sfcb-ProviderMa    10   0.01  0.01  0.00  990.67      -  0.00   0.00   0.00   0.00   0.00   0.00
  2815     2815 dcbd.1000015353     1   0.00  0.01  0.00   99.06      -  0.00   0.00   0.00   0.00   0.00   0.00
  1548     1548 chardevlogger.1     1   0.00  0.01  0.00   99.07      -  0.00   0.00   0.00   0.00   0.00   0.00
11277055 11277055 osfsd.1085790890    1   0.00  0.01  0.00   99.06      -  0.00   0.00   0.00   0.00   0.00   0.00
  1915     1915 sensord.1000014     1   0.00  0.00  0.00   99.07      -  0.00   0.00   0.00   0.00   0.00   0.00
  4556     4556 sfcb-ProviderMa     9   0.00  0.01  0.00  891.61      -  0.00   0.00   0.00   0.00   0.00   0.00
11277140 11277140 swapobjd.100579     1   0.00  0.00  0.00   99.06      -  0.00   0.00   0.00   0.00   0.00   0.00
   952      952 vobd.1000014253     18   0.00  0.00  0.00 1783.55      -  0.00   0.00   0.00   0.00   0.00   0.00
```

图 10-1　esxtop 输出结果

在 `esxtop` 运行界面中输入 `h` 可以访问帮助文档 。下面这些是可用的选项：

- `c`: CPU
- `i`: Interrupt（中断）
- `m`: Memory（内存）

- n: Network（网络）
- d: Disk adapter（磁盘适配器）
- u: Disk device（磁盘设备）
- v: Disk VM（磁盘虚拟机）
- p: Power management（电源管理）

不过，我们的确还有一个工具，可以提供以 VSAN 为中心的性能统计信息，它叫做 VSAN Observer 工具。后面我们会作介绍。

10.5.2 用于 VSAN 的 vSphere Web 客户端性能计数器

与 esxtop 类似，vSphere 客户端也没有针对 VSAN 数据存储的性能计数器。如果你在 vCenter Server 清单中导航到 VSAN 群集对象，选择 Monitoring（监控）页，再选择 Performance（性能）视图，有一个选项可以更改图表。可以发现这里也没有任何针对 VSAN 数据存储的性能图表。

不过，无论是对虚拟机还是 VMDK，vSphere 客户端中的性能视图都非常完美，即使该虚拟机是部署在 VSAN 数据存储上也一样。图 10-2 显示的性能信息视图里面强调的是 VMDK 的读写延迟。

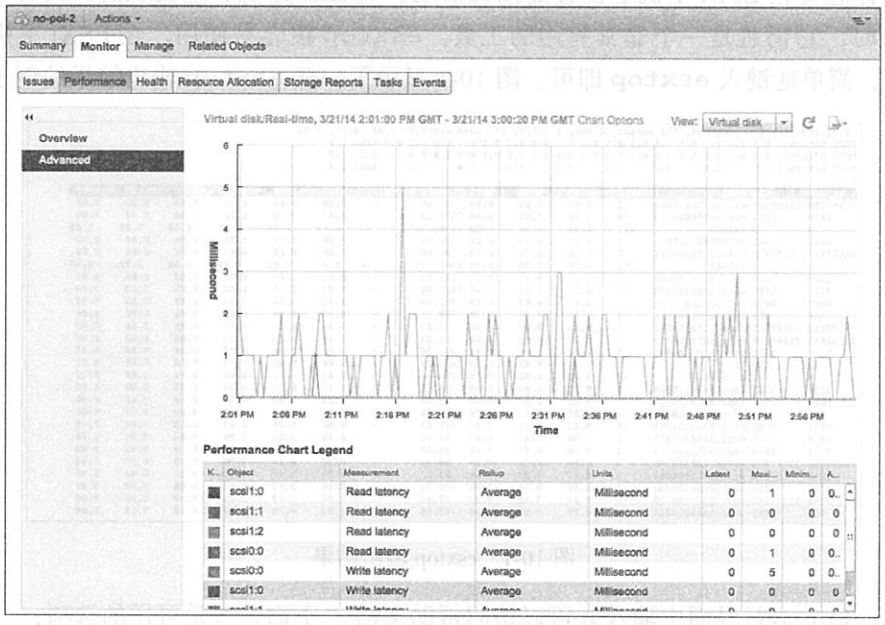


图 10-2 vSphere Web 客户端的性能视图

如前面在 esxtop 一节提过的一样，VSAN Observer 工具可以用来显示关于 VSAN 性能的信息。接下来介绍这个工具。

10.5.3　VSAN Observer

vSphere 5.5 U1 的 vSphere Web 客户端带有很多内建的 VSAN 管理功能。例如，你可以在 vSphere Web 客户端中找到 VSAN 数据存储及其虚拟机层面的性能统计信息。然而，如果要查看更深入的 VSAN 性能，深入物理磁盘层面，理解缓存命中率，追究观察到的延迟原因等，vSphere 5.5 U1 的 vSphere Web 客户端就不能提供这些层面的细节了，这些就是 VSAN Observer 发挥作用的地方了。

VSAN Observer 是随着版本 5.5–U1 的 vSphere vCenter Server 发布的。它是用于 vSphere 管理的交互式命令行 shell 工具 RVC（Ruby vSphere Console）的一个组成部分。RVC 存在于 vSphere 5.5–U1 的 Windows 版本的 vCenter Server 和 VCVA（vCenter 虚拟设备版本）中。

让我们在深入到 VSAN Observer 能做什么之前，先介绍一下如何部署这个工具及需要的前提条件。

VSAN Observer 前提条件

VSAN Observer 是一款性能工具，是特别开发出来用于显示 VSAN 性能信息的。它需要一款现代的 Web 浏览器和 Interent 连接（因为某些开源软件组件需要下载）。它还需要 vCenter Server 5.5 U1，不管是 Linux 的虚拟设备版本还是 Windows 版本都可以。Ruby vSphere Console（RVC）已经在 vCenter Server 5.5U1 中内置了。

有 2 种部署方法：

- 可以使用管理 VSAN 群集的 vCenter Server 生产环境中的 RVC。
- 可以额外部署一台 vCenter Server，仅仅为了使用其中的 RVC 和 VSAN Observer 工具。

在实验环境中，前者更方便。管理员需要知晓 VSAN Observer 开启了一个不加密的非安全的 HTTP 服务器。在生产环境 vCenter Server 上做这些可能会违反安全策略，这就是 VMware 提供了另一个独立选项的原因。在这种情况下，额外部署一台服务器来运行 RVC 可能是一个更好的方案。

运行 `vsan.observer` 命令并将群集名作为一个参数将会启动 VSAN Observer。这条命令会每隔 *x* 秒从 vCenter Server 和 VSAN 收集一次统计信息。用于收集统计信息的默认的间隔是 60 秒，不过你也可以自己通过 `--interval` 参数设置一个更小或更大的值。当前它将连续收集 2 个小时的信息。

通常，我们会和 `-run-webserver` 选项一起运行这条命令，它会在 8010 端口上开启一个不加密的 HTTP Web 服务器。可以通过 `--port` 选项来更改端口号。因为之前我们已经介绍过如何在一个 Windows 版本的 vCenter Server 上开启 RVC，现在让我们来看看在 vCenter Server 的 Linux 虚拟设备版本上让 RVC 运行起来的步骤（这也会启动 VSAN Observer）：

1. 在 vCenter Server 虚拟设备上开启一个 SSH 会话：

```
ssh root@<name or ip of your VCVA>
```

2. 使用 root 账号和 vCenter 名启动 RVC，例如：

```
rvc root@localhost
```

3. 现在，用 cd 命令进入到 vCenter 对象下（每一层下都可以用 ls 来查看对象名字）。如果按 <tab> 键，它会补完数据中心对象的名字：

```
cd localhost/<Name-of-your-datacenter> /
```

4. 现在再用一次 cd 命令。第一个对象是 computers，第二个则是你的群集名。在我的例子中如下所示：

```
cd computers/<Name-of-your-VSAN-cluster>/
```

5. 现在你可以用以下命令开启 VSAN Observer 了：

```
vsan.observer . --run-webserver --force
```

6. 现在你可以看见 Observer 每 60 秒会查询一次统计信息，如前所述，你可以按 Ctrl+C 来终止之。数据收集会在 2 小时后自动停止。

完成这些准备工作的步骤后，现在可以来深入检视一下 VSAN 的性能数据了。首先打开一个浏览器并指向 http://<*rvc-vc-ip*>:<*observer-port*>。<*rvc-vc-ip*> 是运行 RVC 的主机的 IP 地址，而不是你要监控的 vCenter Server 的 IP 地址（尽管它们可能是相同的）。端口默认是 8010，但是你可能已经用 --port 选项更改过了。我们推荐使用 Google Chrome，不过任何当代浏览器应该都可以。Internet Explorer 8 不能算是一个当代浏览器，但是某种程度上也还能用。更早版本的 IE 肯定会出问题。

图 10-3 显示的是首次打开 VSAN Observer 时的页面。

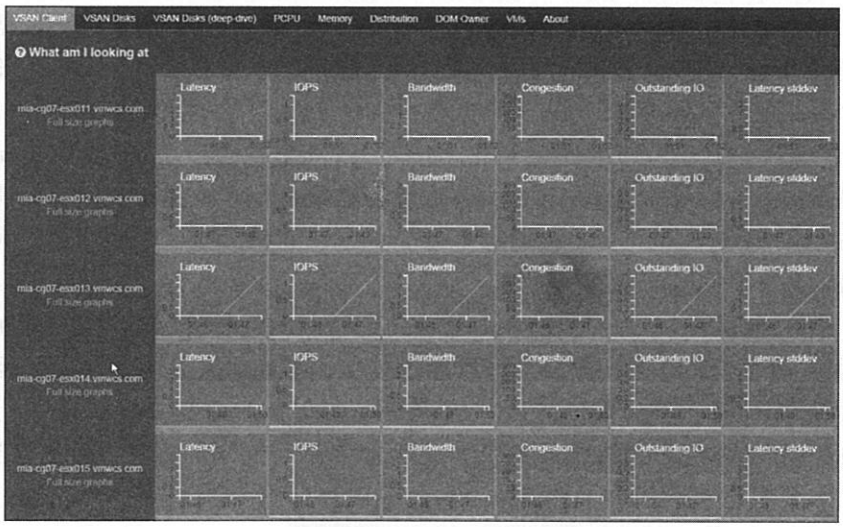


图 10-3　VSAN Observer：VSAN 客户端视图

你还可以要求生成一个统计信息包，这是一个较小的归档文件，其中包含有 Web 浏览器显示的同样信息，你可以下载保存、email 给同事或 email 给 VMware 支持人员。命令行如下：

```
vsan.observer <cluster> -generate-html-bundle /tmp
```

这条命令会将统计信息打包放入 vCenter Server 的 /tmp 目录下。

这条命令会持续运行，直至你按 Ctrl+C 让它停下为止。注意，它会将 Observer 在内存中会话的全部历史记录保留下来直到你按 Ctrl+C 为止，这意味着如果你连续运行几个小时，它会吃掉几个 GB 的内存。这是你可能希望将 VSAN Observer 运行在一台专用的 vCenter Server 上的另外一个原因。

检查 VSAN Observer 性能数据

当你第一次打开 VSAN Observer 时，异常问题的主要指示器会在问题的图形下显示一条红色的下划线来表明它已经超出了正常的操作边界。VSAN Observer 中的图像通常在正常状态下会显示为绿色，或是在没有收集到数据或没有足够信息时显示为灰色。红色就是需要开始调查的信号灯，它会在样本收集期间有 20% 的样本超出设定的阈值范围时显示出来。

VSAN Observer 的用户界面是通过一个个子系统组织起来的。你应该先从 VSAN 客户端视图着手，这个视图会给你一个从 VSAN 收集起来的虚拟机服务等级的概览。VSAN 群集中的每台主机（下文中简称为“ VSAN 客户端”）都在使用着分布在群集中所有其他主机上的存储，所以在 VSAN 客户端的主机 A 上发现的性能问题可能事实上是由于主机 B 上的磁盘超负荷造成的。

“ VSAN 磁盘”视图允许你从磁盘角度来研究 VSAN，它通过研究如何从本地磁盘提供 I/O 服务来检查节点是如何向 VSAN 数据存储贡献存储的。然后你可以进一步一台一台主机地深入研究 VSAN 的磁盘层面，检查 VSAN 是如何在 SSD 和 HDD 之间分配 I/O 的。

图 10-4 显示了 VSAN 磁盘视图。从图中可以看出，这里显示了大量的信息。这个视图显示了所有的内容，包括延迟、IOPS、带宽、拥塞、突出的 I/O 和延迟的标准差（说明了延迟偏离平均值的程度）。再说一次，你要看的是带有红色下划线的那些图表，这些高亮显示出一种反常状态，是磁盘相关问题调查的切入点。对于延迟，阈值水平被设成了 30 毫秒。带宽是以每秒千字节（KB/s）来衡量的。拥塞程度的范围从 1 ～ 255，1 意味着无拥塞，255 意味着完全堵塞。拥塞的阈值设在了 75。

VSAN 和 ESXi 的其余部分共享计算资源，也就是说，VSAN 会消耗掉一小片主机的 CPU 和内存资源，而这台主机上同时还运行着虚拟机。VSAN 被设计成不会消耗超过 10% 的主机 CPU 资源。你可以通过 Observer 相关的页面来检查 VSAN PCPU（物理 CPU）和内存的消耗，这或许对于发现因 CPU 或内存限制而引起的性能瓶颈有所帮助。

图 10-5 显示了内存消耗状态，其中不仅仅包括 VSAN 组件的消耗，还包括了其他参与到 VSAN 群集的各种 ESXi 主机对内存的消耗。

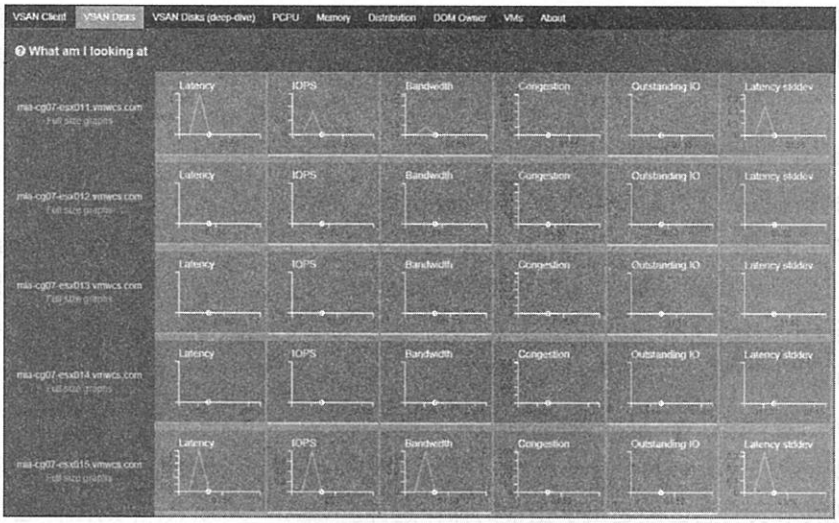


图 10-4 VSAN Observer：VSAN 磁盘

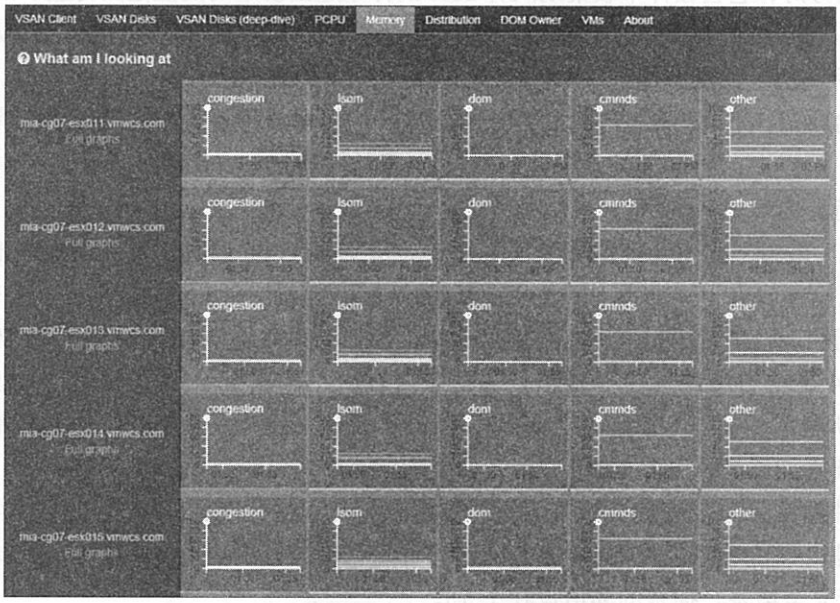


图 10-5 VSAN Observer：内存

因为 VSAN 是一种以虚拟机为中心来管理的存储（针对每个虚拟磁盘的虚拟机存储策略），你也可以在 VSAN Observer 中就每台虚拟机甚至是每块虚拟磁盘的层面来进行性能观测。要这么做，在 VM Home（虚拟机主页）选择你想要获得更多细节信息的那台虚拟机即可。

图 10-6 显示的是一台虚拟机的虚拟机主页（VM Home）。对于组成此对象的每个组件，延迟、IOPS、读缓冲（RC）命中率和缓存清空等信息全都显示在这里。对于可能表现出性

能问题的特定虚拟机来说，这些优质的信息可以用来判断组成此存储对象的组件是否存在问题。清空（Evictions）是一种事实参考值，用来表示缓存中正被从缓存刷入到磁盘的数据项的状态，其较高的数值意味着可能存在缓存源端竞争，暗示闪存的大小可能规划不合理。读缓冲命中率也是一个有趣的图形，因为任何低于 100 的值都说明存在缓冲未命中，从而导致读取一个数据块将不得不直接访问磁盘来实现，这会增加延迟。

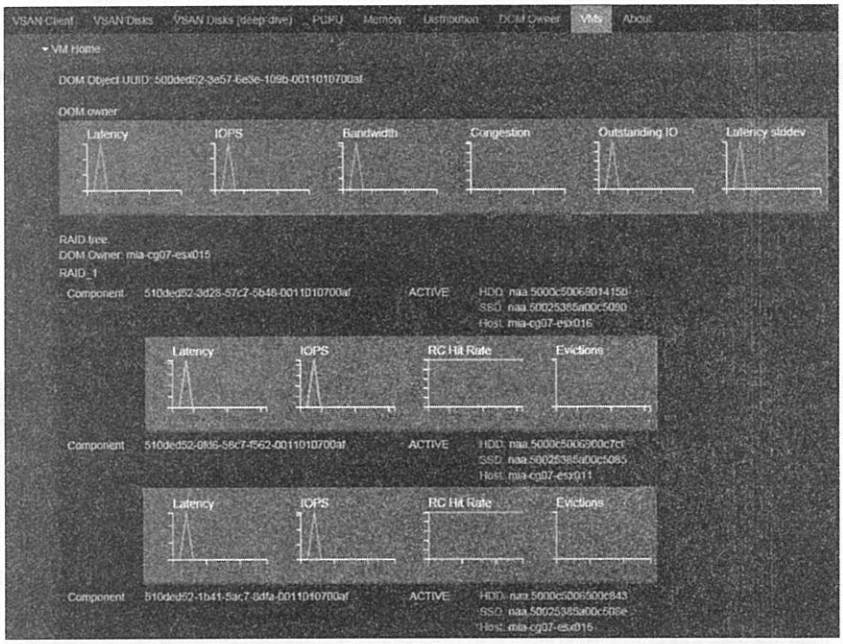


图 10-6　VSAN Observer：虚拟机视图

最后但也是非常重要的一点，还有一些辅助信息页（关于群集均衡、对象分布、显著的群集事件等）。每次你切换页面，图像都会自动更新以反映出后台 RVC 收集到的最新数据。

大多数页面都包含如何解释图像中展示的信息的相关内容。然而，很多信息还是需要你熟悉存储性能的知识。可以这么说，你对 VSAN Observer 工具用得越多，你就越熟悉怎样才能让自己的环境运行在稳定状态，那么当异常状况发生需要进行诊断和排错的时候，这个工具就越有用。

如你所想，关于能用 VSAN Observer 来做什么我们只是泛泛地介绍了些皮毛而已。

10.6　VSAN Observer 使用示例

为了更好地演示 VSAN Observer 的强大之处，我们找到了 Simon Todd——VMware 支持团队中 VSAN 方向的技术领军人物之一，询问他使用 VSAN Observer 来对 VSAN 的性能问题进行排错的经验。Simon 很爽快地分享了下面这个例子，包括问题症状和根本原因。

Simon 分享的案例中的问题描述是这样的："在 I/O 测试中，虚拟机很慢，某些虚拟机不可访问"。在问题被迅速确认之后，排错过程始于日志文件调取。首先，DOM 汇报称有一个操作花费了太多时间（DOMTraceOpTookTooLong），如图 10-7 所示。

```
2013-12-05T17:21:38.356392 [26141117] [cpu27] [22a870d2 OWNER write]
DOMTraceOpTookTooLong:2645: {'op': 0x4136c85f78c0, 'obj': 0x412eca2b4c40,
'objUuid': '58cc9852-d985-9aae-8bd9-0006f62b11ec', 'time msec': 103129}
```

图 10-7　DOMTraceOpTookTooLong

调查组还发现 LSOM 报告了一个非常高的 Congestion（拥塞值）255，如图 10-8 所示。

```
2013-12-05T17:21:35.586223 [27666780] [cpu3] [3184]
LSOMTraceRcLSOMVAAllocFailure:1055: {'status': 'VMK_NO_MEMORY'}
2013-12-05T17:21:35.586224 [27666781] [cpu3] [3184] LSOMTraceRcScanError:
3332: {'rclPacked': 0x0, 'objectSlot': 14000, 'rclIndex': 0, 'rdt': 0x0,
'inFlight': 0x0}
2013-12-05T17:21:35.586224 [27666782] [cpu3] [3184]
LSOMTraceRcDomCompletion:1461: {'status': 'VMK_NO_MEMORY', 'time':
'00:00:00.000003', 'heapCongestion': 0, 'vaCongestion': 0, 'iopsCongestion':
0, 'congestion': 255}
```

图 10-8　Congestion（拥塞值）高达 255

现在是时候转向 VSAN Observer 来看看它对定位日志文件中发现的问题的根本原因是否有帮助。当 VSAN Observer 开启并打开 VSAN 客户端视图后，马上发现系统存在显著的延迟问题，如图 10-9 所示。并且和日志中找到的拥塞报错一样，这里也显示有拥塞问题。

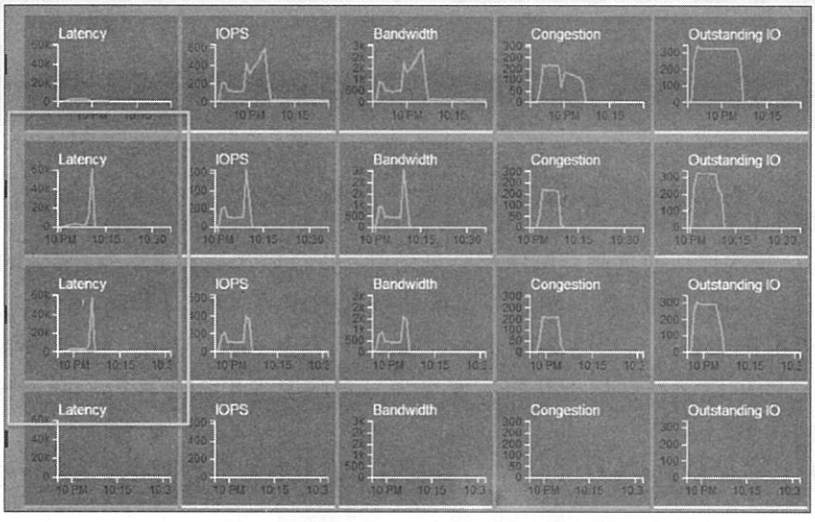


图 10-9　延迟问题

这些异常情况通过红色的下划线来指明，可以马上引起管理员的注意。点击图像将其放大可以显示进一步的细节。在这个例子中，我们展开了其中一个延迟的图像，关于延迟值的进一步细节展现出来。如图 10-10 所示，图像中的读取延迟值异常高，到达了 60 000 毫秒的区间（或 60 秒）。

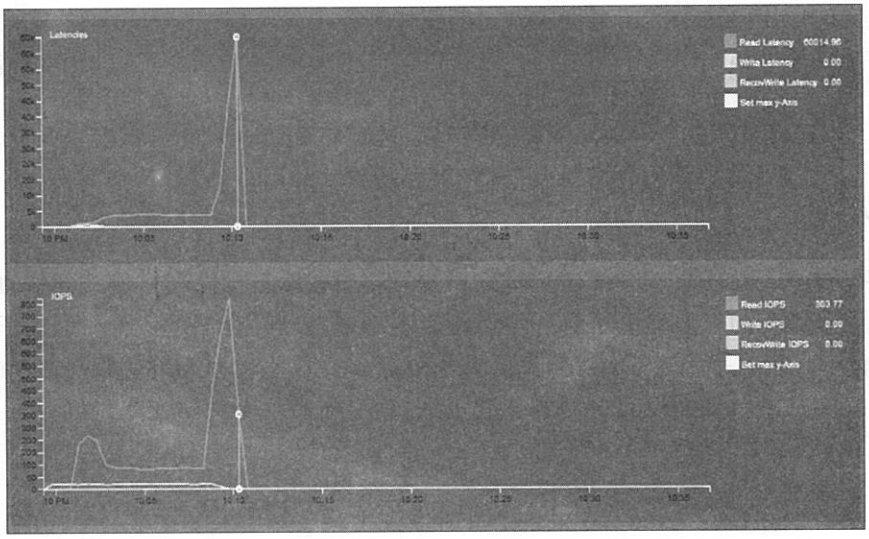


图 10-10　延迟和 IOPS 的细节

引起我们注意的一个事实是，IOPS 值相对较低，但延迟值却很高。为什么会这样？还需要进一步进行研究。如前所述，还有其他一些有趣的 VSAN Observer 视图可以看一看。下一个有意思的视图就是虚拟机（视图）。在这个视图中，我们发现某一个虚拟机组件存在大量的缓存清空（cache evictions），因此，看上去延迟和拥塞的根本原因在于闪存设备无法满足工作负载。它需要经常性地从缓存中清空数据来为新的数据块腾出空间。如图 10-11 Eviction 图像中的橙色峰值所示。

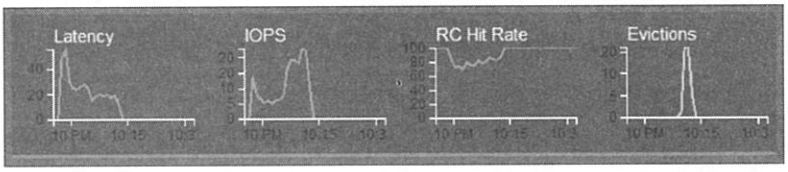


图 10-11　Evictions

于是支持团队询问客户，在这个环境中使用了哪种类型的闪存设备。结果发现在这个环境中使用的闪存设备是 Class A 闪存，最高 IOPS 只有 2500。每台主机的闪存设备都已经被客户部署的工作负载给撑爆了，这导致了大家不希望看见的结果。不幸的是，VMware 支持团队无法帮助这个客户修复问题，因为他们使用的是不受 VMware 支持的闪存设备（VMware 不支持任何类型的 Class A 闪存设备）。

希望这个实用案例能在某种程度上反映出 VSAN Observer 的强大，它不仅仅能告诉你如何使用它来确认瓶颈，还可以用来反映出对于闪存设备类型的选择会对基础架构的性能造成何种影响。你应该通过持续观察，让自己熟悉自己的环境，怎样才算是正常状态怎样算是不正常。尽管当数值超过阈值范围时会显示红色的下划线来报警，这也在某种程度上

起到了一定的作用。这个实用案例应该还能使你理解正确规划 VSAN 环境的重要性，所以再重申一次，你应该复习一下第 9 章，以获得如何正确规划 VSAN 群集大小的指导，并避免类似这里讨论的问题在你的环境中发生。

10.7 小结

现在我们可以清楚地知道，对 VSAN 环境的监控和诊断排错有一整套可用的工具。我们从很多 VMware 的客户那里了解到他们再也不希望自己的存储是一个“黑匣子”——关于其性能的可见性几乎为零。有了这样一套命令行（CLI）的扩展包和图像界面（UI）工具，客户就可以钻入 VSAN 操作的底层来一探究竟了。

推荐阅读

高性能CUDA应用设计与开发：方法与最佳实践

作者：（美）Rob Farber ISBN：978-7-111-40446-0 定价：59.00元

CUDA并行程序设计：GPU编程指南

作者：（美）Shane Cook ISBN：978-7-111-44861-7 定价：99.00元

CUDA专家手册：GPU编程权威指南

作者：（美）Nicholas Wilt ISBN：978-7-111-47265-0 定价：85.00元

GPU高性能编程CUDA实战

作者：（美）Jason Sanders 等 ISBN：978-7-111-32679-3 定价：39.00元

大规模并行处理器程序设计（英文版·第2版）

作者：（美）David B. Kirk 等 ISBN：978-7-111-41629-6 定价：79.00元

推荐阅读

企业虚拟化实战——VMware篇

作者：张巍 ISBN：978-7-111-27544-2 定价：59.00元

VMware、Citrix和Microsoft虚拟化技术详解与应用实践

作者：马博峰 ISBN：978-7-111-40319-7 定价：109.00元

VMware vSphere 5虚拟数据中心构建指南

作者：（法）Eric Maillé 等 ISBN：978-7-111-41677-7 定价：59.00元

VMware vSphere部署的管理和优化

作者：（美）Sean Crookston 等 ISBN：978-7-111-42543-4 定价：59.00元

VMware 站点恢复管理器管理实践

作者：（英）Mike Laverick ISBN：978-7-111-45735-0 定价：79.00元