

**研究生学位论文开题报告**

**报告题目　 　 云应用敏感的混合存储**

**性能保障技术研究**

**学生姓名 唐震 学号 201218015029014**

**指导教师 黄涛 职称 研究员**

**学位类别 工学博士**

**学科专业 计算机软件与理论**

**研究方向 网络分布计算理论与技术**

**培养单位 中国科学院软件研究所**

**填表日期 2015年12月15日**

**中国科学院大学制**

**填 表 说 明**

1. 本表内容须真实、完整、准确。
2. “学位类别”名称填写：哲学博士、教育学博士、理学博士、工学博士、农学博士、医学博士、管理学博士，哲学硕士、经济学硕士、法学硕士、教育学硕士、文学硕士、理学硕士、工学硕士、农学硕士、医学硕士、管理学硕士等。
3. “学科专业”名称填写： “二级学科”全称。

**目　录**

[1. 选题的背景及意义 4](#_Toc439152779)

[1.1. 选题背景 4](#_Toc439152780)

[1.1.1. 固态盘的背景及其特点 4](#_Toc439152781)

[1.1.2. 混合存储 5](#_Toc439152782)

[1.1.3. 混合存储与云计算的结合 7](#_Toc439152783)

[1.2. 选题意义 8](#_Toc439152784)

[1.2.1. 目前应用模式的局限性 8](#_Toc439152785)

[1.2.2. 服务质量违约带来的后果 8](#_Toc439152786)

[2. 国内外本学科领域的发展现状与趋势 9](#_Toc439152787)

[2.1. SSD缓存分配 9](#_Toc439152788)

[2.2. 数据预取 10](#_Toc439152789)

[2.3. 全局IO控制 10](#_Toc439152790)

[3. 课题主要研究内容、预期目标 11](#_Toc439152791)

[3.1. 研究内容 11](#_Toc439152792)

[3.1.1. 云应用敏感的混合存储SSD资源分配 11](#_Toc439152793)

[3.1.2. 云应用敏感的混合存储数据预取 12](#_Toc439152794)

[3.1.3. 云应用敏感的混合存储全局控制 13](#_Toc439152795)

[3.2. 预期目标 14](#_Toc439152796)

[4. 拟采用的研究方法、技术路线、实验方案及其可行性分析 14](#_Toc439152797)

[4.1. 研究方法 14](#_Toc439152798)

[4.2. 技术路线 14](#_Toc439152799)

[4.3. 实验方案 15](#_Toc439152800)

[4.4. 可行性分析 15](#_Toc439152801)

[5. 已有科研基础与所需的科研条件 15](#_Toc439152802)

[6. 研究工作计划与进度安排 16](#_Toc439152803)

[参考文献 17](#_Toc439152804)

# 选题的背景及意义

## 选题背景

### 固态盘的背景及其特点

固态盘（Solid State Drives，SSD）是一种新的存储设备，主要使用非易失的闪存芯片（NAND Flash）作为存储介质，由闪存阵列及对应的相关电路和主控芯片组成，并最终封装成标准的磁盘接口形式。

闪存技术提出较早，但由于成本原因，一直未能广泛应用，直到近年来随着半导体技术的提升以及成本的降低，固态盘才逐步进入消费领域。消费级的固态盘出现在2006年，Samsung于2006年3月发布了基于NAND闪存的固态盘，具备57Mbps的读速度和32Mbps的写速度，是当时主流的传统磁记录式机械硬盘（Hard Disk Drives, HDD）的1.5倍，但功耗只有机械硬盘的5%[1]。此后，固态盘受到了学术界和工业界的持续关注，其在存储系统中的应用也越来越广泛。各大著名闪存厂商，如Intel，Samsung，Toshiba，Plextor等以及传统的机械硬盘公司如Seagate和Western Digital都推出了相应的产品以抢占这一新兴存储市场份额。各大公司和企业也都已经或计划在他们的存储系统中采用固态盘以提高性能和降低能耗[2, 3]。国际著名调查机构IDC在2008年的一项预测表明，2011年将约有300万块固态盘被用来支持企业的应用需求，创造12亿美元的市场[4]。由于在性能、能耗、可靠性等方面的优势，固态盘正被广泛地应用在存储系统中，包括个人移动终端存储、工作站、服务器及高性能计算中心[5-7]。

固态盘与传统的机械硬盘在读写数据的原理上有着本质的不同。固态盘的存储是基于闪存的，目前成熟的闪存技术包括单层存储（Single Level Cell，SLC）、多层存储（Multi Level Cell，MLC）和三层存储（Triple Level Cell，TLC），它们的成本依次降低，但与之对应的使用寿命和读写速度也依次降低，进行纠错的代价依次升高。

固态盘使用闪存阵列作为主要数据存储形式，其内部是由主控芯片进行统一管理的，可以提供较高的并行访问能力[8, 9]，这也意味着固态盘具有较高的随机读写能力。而具体到每个闪存芯片，其支持读（Read）、写（Write）和擦除（Erase）三种操作。读操作通过检测存储单元的电压状态来判断当前单元所存储的信息；写操作通过改变存储单元的电压来改变其中的信息，但只能将其从“1”变为“0”；擦除操作用于将存储单元的信息重置为“1”。因此，只有经过擦除操作，对应的存储单元才能再次接受写操作。而与之对应的是，读操作和写操作的延迟通常是微秒级的，但擦除操作的延迟是毫秒级的，这也导致固态盘的读写操作的速度呈现非对称的特点，固态盘也需要消耗较多的时间来擦除旧数据，进行垃圾回收操作，平衡擦除和读写操作，保证读写性能稳定，防止固态盘在全盘写入之后写性能急剧降低。

闪存芯片维持稳定电压的能力是有限的，这也意味着闪存芯片只具有有限的擦写周期（Erase Cycle），当超过这一阈值时，闪存芯片将变得不稳定，存储在其上的信息将不可识别，可靠性也会降低[10-15]。针对前文所述的三种主流的闪存技术，SLC的寿命约为100000个擦写周期，MLC的寿命约为10000个擦写周期，而TLC的寿命约为1000~1500个擦写周期。

表 1[[1]](#footnote-1)比较了目前主流固态盘和机械硬盘的性能和性价比。综合上述固态盘的特性以及表 1的数据，可以发现固态盘相对于机械硬盘有如下特点：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **产品名称** | **容量** | **顺序读/写**  **（单位：MB/s）** | **随机4K读/写**  **（单位：IOPS）** | **价格/每GB价格**  **（单位：元）** |
| Intel 750 | 1.2 TB | 2500/1200 | 460,000/290,000 | 7,499/6.25 |
| Intel 730 | 480 GB | 550/470 | 89,000/74,000 | 1,999/4.16 |
| Intel 535 | 240 GB | 540/490 | 41,000/80,000 | 1,099/4.58 |
| WD Blue SSHD | 4 TB | 150/150 | 60/120 | 1,199/0.30 |
| WD Black | 4 TB | 180/180 | 80/150 | 1,499/0.38 |

表 1：主流固态盘和机械硬盘的对比

优势：

1. 固态盘相比机械硬盘而言没有机械动作，即启动电机及移动磁头的代价，因此降低了耗电量。
2. 固态盘没有机械运动的部件，不会因为机械故障（撞击、跌落等）而损坏，具有较高可靠性。
3. 固态盘并行度高，随机读写速度快，随机读写的每秒IO请求数（IOPS）是机械硬盘的100~500倍。
4. 固态盘的顺序读写速度相对机械硬盘有一定的优势，顺序读写速度（MB/s）约为机械硬盘的5倍。

劣势：

1. 固态盘的存储单元擦写寿命有限，而机械硬盘的寿命远高于固态盘的寿命。
2. 固态盘的读写速度是不对称的，机械硬盘的读写速度是对称的，这也意味着使用者需要考虑固态盘频繁进行擦除操作时带来的性能下降。
3. 固态盘的成本较高，与机械硬盘相比每GB容量的成本约为机械硬盘的4~6倍。

因此，综合考虑固态盘和机械硬盘的优劣，如何将两者混合使用构成混合存储系统，充分发挥两者优势，成为工业界和学术界关注的问题。

### 混合存储

随着固态盘的普及，基于固态盘和机械硬盘的混合存储成为研究热点。混合存储是一种集固态盘和机械硬盘为一体，以大容量、高性能和低成本为目标的异构性非易失存储应用模式。其设计思想在于使性能好、价格高的固态盘在存储系统中发挥杠杆作用，发挥固态盘和机械硬盘的各自优势并弥补对方的短处，让系统以接近机械硬盘的价格提供近似固态盘的性能。

由于固态盘具有非易失性，因此在数据不需要共享，且所有读写请求均不绕过固态盘的情形下，可以将固态盘作为非易失性缓存（Non-Volatile Cache，NV Cache），安全地将脏数据保存在固态盘上，提高读写性能。

针对固态盘的这一特点，产业界目前存在两个主要的成熟应用模式，一是将小容量固态盘内置于机械硬盘中，用作非易失性缓存，两者构成混合硬盘（Hybrid Disk，产业界也称作SSHD），二是使用独立的固态盘用作非易失性缓存，辅以存储管理应用来加速独立的机械硬盘。

2006 年，三星公司在消费领域推出了混合磁盘[16, 17]。混合硬盘的特点主要有：

1. 混合硬盘可以借助固态盘缓存来过滤频繁读写的小规模数据，在不发回机械硬盘进行读写操作时可以将机械硬盘停转，降低耗电量。
2. 混合硬盘可以在机械硬盘因为感应到震动而停机时继续通过内置的固态盘提供服务，从而提高了可用性。
3. 将内置的固态盘作为读写缓存，可以提高混合硬盘的整体性能，尤其是小规模频繁随机读写的性能。由于缓存具有非易失性，混合硬盘中可以安全地使用回写（Write back）的写缓存策略以最大化写操作性能。

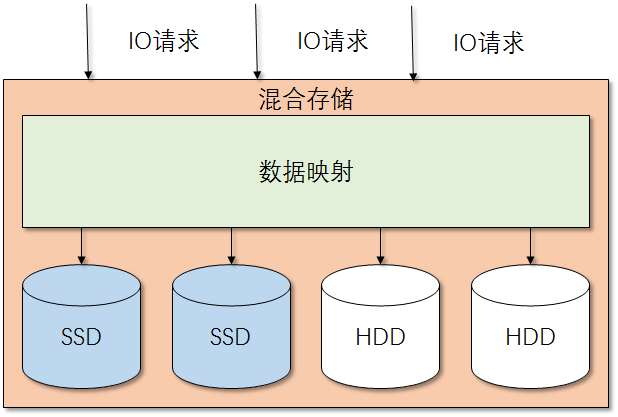


图 1：混合存储原理图

1. 与固态盘相比，混合硬盘的成本较低。

使用独立固态盘作为独立机械盘的缓存的应用模式也得到了如Intel等厂商的关注。内置于Intel快速存储技术（Intel Rapid Storage Technology，Intel RST）[18]解决方案中的Intel智能响应技术（Intel Smart Response Technology，Intel SRT）[19]使用不低于16GB的独立固态盘作为独立机械硬盘的缓存，通过判断IO数据块的收益，将对用户而言高价值的数据块（如频繁访问的操作系统文件、应用程序和文档等）缓存至固态盘，而不缓存将对用户而言低价值的数据块（后台服务访问的数据等）。这一技术适用于桌面环境下提高系统对用户的响应能力，因此目前被各大PC厂商广泛用于台式机及笔记本中以提升存储性能、可靠性并降低能耗，如典型的16GB固态盘配合1TB机械硬盘的形式[20]。

另一方面，在学术界，研究人员也从不同角度出发提出了多种固态盘和机械盘的存储系统，主要研究方向包括降低能耗[21, 22]、利用缓存提高性能[23-26]、提高缓存寿命[27]等，目标场景也由单纯的桌面系统演进至数据中心的服务器[28, 29]。

典型的混合存储原理如图 1所示。混合存储中最重要的是数据映射模块。数据映射模块将底层的存储部件（固态盘和机械硬盘）组织成一个统一的存储空间供上层操作系统及应用使用。数据映射模块通常包含映射表和具体的映射策略以及监测IO请求特点的检测模块。根据固态盘和机械硬盘之间的组织方式和应用场景，混合存储可以分为如下三种类型[30]：

1. 层次型混合存储：机械硬盘作为永久存储，而固态盘作为机械硬盘上层的非易失性缓存。层次型混合存储系统将固态盘作为缓存层，在系统原先的内存缓存（DRAM缓存）之外扩展了一个辅助缓存。在层次型混合存储系统中，应用经常访问的数据块被缓存至固态盘中。由于固态盘是非易失的，数据块的后续IO操作可以直接发送至固态盘，从而减小响应时间。
2. 逆置型混合存储：固态盘作为永久存储，而机械硬盘作为固态盘的写缓存。逆置型混合存储系统将速度较慢的机械硬盘作为固态盘的写缓存，在写请求发送至固态盘之前先写入机械硬盘，待积累到一定程度的顺序写操作之后再写回固态盘。这种混合存储系统的应用模式主要权衡了固态盘的寿命，以及固态盘在读操作上的高性能。
3. 并列型混合存储：固态盘和机械硬盘均作为永久存储，使用同一地址空间，通过I/O 重定向和数据迁移来优化性能。在这种应用模式下，固态盘和机械盘共同构成了存储系统。与前两种应用模式不同，并列型混合存储系统中并没有将某一种存储设备作为缓存，而是将其组合成一个统一的存储空间。在这一应用模式中，数据映射模块主要负责控制数据的具体放置，并进行适当的迁移和重映射操作。

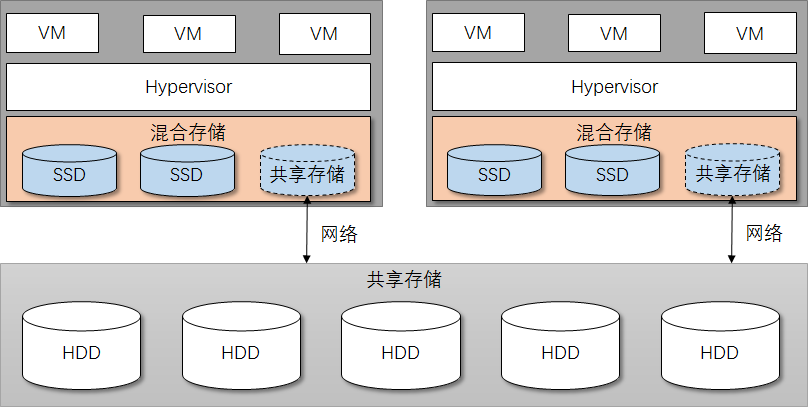


图 2：混合存储在云计算中的主要应用模式

而在虚拟化环境下，云应用的广泛部署对支撑虚拟机运行的混合存储提出了新的挑战。

### 混合存储与云计算的结合

在虚拟化环境下，混合存储的应用模式有极大区别。虚拟化环境中的物理节点通常用于承载虚拟机。而为了保证虚拟机的可用性和可靠性，虚拟机通常都是保存在一个统一的后端存储系统中，而非物理节点本地。而与之相对应的，缓存系统通常部署在物理节点本地，以消除网络开销并获得较高的性能。

因此，混合存储系统的组织模式应当进行改变，以适应虚拟化环境下的需求。dm-cache[31-33]是一个使用快速块设备来加速慢速块设备的缓存组件，并于2013年4月合并到Linux 3.9内核中[34, 35]。Facebook也基于dm-cache设计了FlashCache[36]，作为后端服务的加速手段。

云应用是一种新型的应用承载模式，是将应用承载与虚拟机上的伺服模式。主流云计算平台，如Amazon EC2[37]和Microsoft Azure[38]、阿里云等，都基于虚拟化技术，提供了按需租用虚拟机的模式，供用户自行部署不同类型应用，以构成弹性伸缩的云应用集群。在云应用的背景下，混合存储也得到也越来越多的关注。

混合存储在云计算环境下的主流应用模式是作为虚拟机的磁盘缓存，如图 2所示。学术界针对混合存储在云计算环境下的应用也进行了相关研究工作，如Mercury[39]、vCacheShare[40]、Capo[41]、Centaur[42]等。

目前，主流的云计算厂商，如Amazon EC2、Microsoft Azure、阿里云等，都提供了基于SSD或混合存储的虚拟机存储方案。主流的存储服务提供商，如EMC、IBM和NetApp等，都提供了混合存储的方案用以支撑虚拟机的后端存储。

Gartner报告[43]指出，服务器SSD迎来爆发式增长（2015同比增长51%），到2017年，SSD市场将比2014年增长5倍，混合存储将成为服务器市场的主要使用模式之一。IDC的报告[44]指出，企业级的全闪存阵列（All Flash Array，AFA）及混合闪存阵列（Hybrid Flash Array，HFA）迎来快速增长，混合闪存阵列相比全闪存阵列具有更高的增长率和利润。可以预见混合存储在云计算环境下的应用将越来越广泛。

## 选题意义

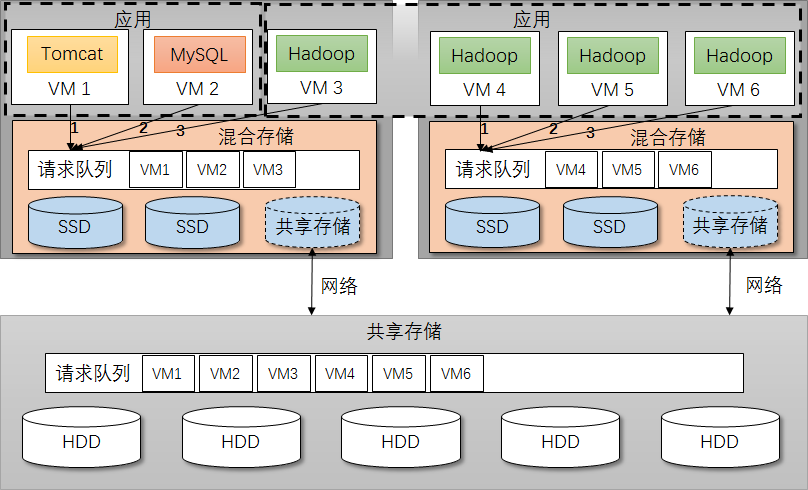


图 3：混合存储的局限性

### 目前应用模式的局限性

云应用的场景是多租户的，与混合存储面向单一租户的使用模式存在矛盾。而目前，混合存储的主要应用模式并没有考虑到上层应用的特点，从宏观上看仍然是先来先服务的，如图 3所示。而对于主流的云应用类型，目前混合存储的应用模式都没有考虑到相关的关键特性。对事务型应用，如Tomcat和MySQL共同构成的云应用服务器集群，混合存储没有考虑到云应用各个组件对存储的不同需求，从而无法差异化地提供存储性能保证，导致应用的响应时间增加；对大数据处理型应用，如Hadoop，混合存储没有感知到大数据处理平台上执行的任务及其具体的调度，从而无法根据应用的特点和需求执行数据预取，导致应用上所运行任务的完成时间增加。而从虚拟机集群的角度看，目前的混合存储应用模式存在多条独立的IO请求队列，未能实现全局统一管理，从而难以保证云应用端到端的性能。当前混合存储应用模式的这些局限性会导致服务质量违约并带来严重后果。

### 服务质量违约带来的后果

目前混合存储的应用模式难以很好地实现资源管理，易造成服务质量违约。对事务型应用，服务质量违约会导致应用的响应时间增加，影响用户体验；对大数据处理型应用，服务质量违约会导致任务执行时间延长，影响用户的分析和决策。

早先的相关测试[45]表明，不合理的SSD资源分配会带来巨大的性能差异。如图 4所示，针对4个不同的应用T1、T2、T3和T4分配SSD缓存，带来的吞吐率差异在3倍以上。

而同时，由于云应用的各个组件均依赖于混合存储，混合存储上的延迟反应到云应用上影响会放大5倍。有报告[46]指出，Amazon后端存储的响应时间每延长0.2秒，最终反映到前端Web页面的响应时间会延迟1秒，会导致用户访问量降低11%，用户满意度降低16%，而Amazon的年销售额也会因此减少16亿美元。在用户访问量降低的同时，也极大影响了用户的访问体验，最终对品牌形象造成负面影响。

# 国内外本学科领域的发展现状与趋势

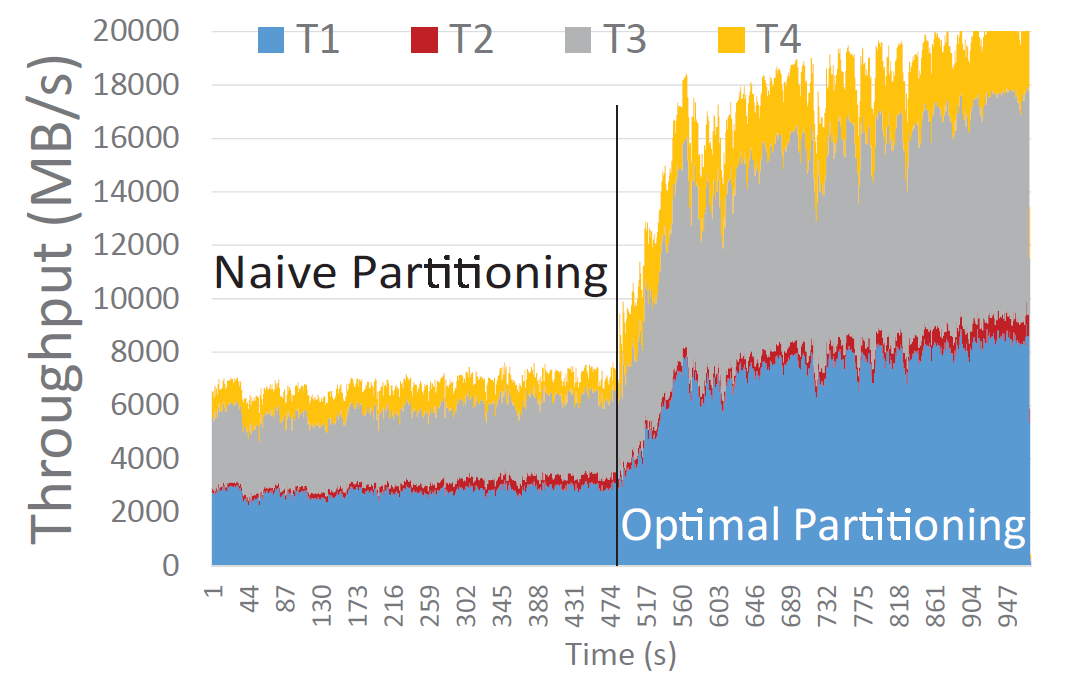


图 4：不同缓存分配方案带来的吞吐率影响

高质低廉的存储服务是云计算的核心竞争力，各大云计算提供商也在努力提升混合存储的服务质量，在学术界，近年来存储[32, 47]和系统[29, 48]的顶级会议，如FAST、OSDI等，也越来越多地关注混合存储的相关研究。

混合存储的服务质量保障在近年成为研究热点，而其中亟待解决的问题是如何优化主流类型的云应用，如事务型应用和大数据处理型应用的性能，以及如何保障云应用端到端的性能和服务质量。

针对事务型应用，主要的优化手段是为虚拟机分配SSD缓存；针对大数据处理型应用，主要的优化手段是针对那些对应用性能影响较大的数据进行预取；在云应用端到端服务质量保障方面，主要手段是进行全局的IO控制。

## SSD缓存分配

目前国内外针对SSD缓存分配的主要研究工作有如下几点：

Centaur[42]通过缓存分区（Cache Partitioning），为不同的虚拟机分配独立的、不同大小的缓存，从而获得相比使用统一缓存（所有虚拟机共享同一个SSD缓存）而言更高的缓存利用率和性能。作者发现缓存的命中率随着缓存大小的变化而变化，在到达一定阈值之后分配更多的缓存并不能使得命中率有效提高，因此可以基于缓存命中率变化曲线确定虚拟机的SSD缓存分配。Centaur基于在线测试和反馈的方法，通过尝试和评价不同的缓存分配方案最终选择整体缓存命中率最高的缓存分配方案。Centaur也将这一方法应用到了其他关键性能指标上，如提升整体吞吐率或降低整体IO延迟等。

S-CAVE[49]基于测试和历史数据分析的方法，根据缓存命中率及由缓存相关指标导出的数据，如rECS（有效缓存空间比例）、已分配缓存空间、早先的分配决策等，来决定缓存的分配。S-CAVE通过检测rECS和命中率的变化，基于CLOCK算法，根据上一个时间片内rECS和命中率的改变量分为4种情形并适用不同的决策和缓存大小调整方法，最终实现虚拟机整体性能的提升。

以上主流的SSD缓存分配相关工作都基于缓存命中率或相关指标展开，属于命中率敏感的缓存分配，最终的调整目标也是与缓存本身的特点相关的，如提高缓存命中率，降低IO延迟等。

这些工作存在的局限性是它们都将虚拟机独立对待，针对不同的虚拟机分配缓存，没有考虑到应用的不同组件部署在多台虚拟机上的情形，也没有考虑到虚拟机之间的关联，从而不能很好地应对多虚拟机部署的事务型云应用。

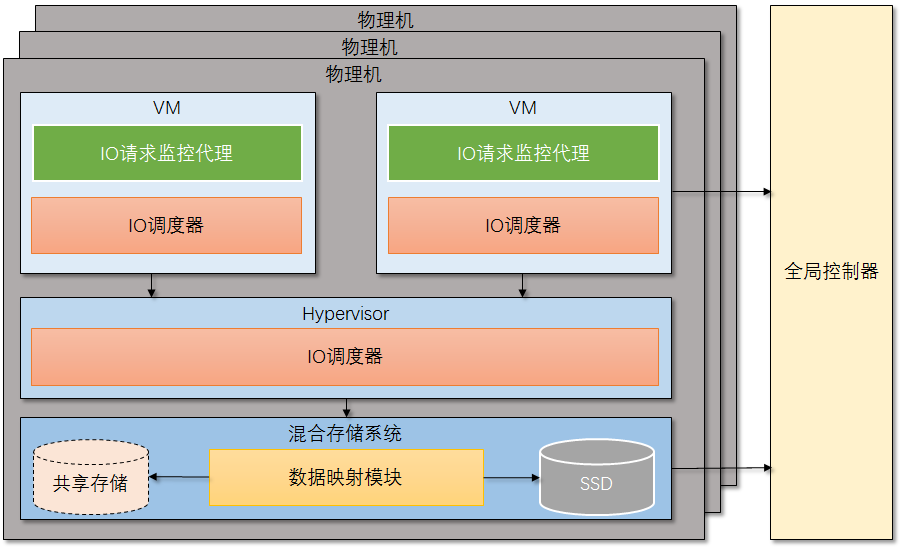


图 5：系统架构图

## 数据预取

目前国内外针对云计算环境下数据预取的主要研究工作有如下几点：

HAT（Hotness Aware Hit）[50]采用了基于历史数据分析的方法，利用缓存的页面引用队列（Page Reference Queue）来区分不同类型的数据，并将缓存划分为3块区域（Hot、Warm、Cold）从而区别对待缓存页面。对不同的区域HAT采用了不同的存储策略，将Hot区域的页面存储在内存中，将Warm区域的页面存储在内存和SSD中，将Cold区域的页面存储在磁盘中。HAT进一步修改了缓存置换算法，为不同类型的缓存命中情况设定不同的处理策略，最终实现系统性能提升。

CRAID[51]将缓存整合至RAID系统中，为RAID阵列中的每一块磁盘分配了独立的缓存空间（Cache Partition），用以存放当前频繁访问的数据，而剩余空间用于存储暂时未访问的数据。CRAID同时监控了IO请求并导出了数据访问模式，以识别热点数据并进行相应处理。CRAID重定向了IO请求，针对不同类型的访问请求和工作负载类型，执行不同的预取和缓存置换操作，以实现RAID的性能和性价比的提升。

这些工作存在的局限性是它们都基于统计和历史数据的方法，需要依赖先验知识或历史数据，以修正模型。同时，它们也没有考虑到多节点部署的应用以及应用之间的相关关系，不适用于负载和访问模式在运行时快速变化的场景。

## 全局IO控制

目前国内外针对云计算环境下全局IO控制的主要研究工作有如下几点：

IOFlow[52]针对不同虚拟机的IO请求进行了标记，将虚拟机的IO请求抽象为IO流的形式（类似于SDN中的流），之后通过控制IO流在网络上的传输以实现针对IO请求的控制。IOFlow提供了一组API，以针对特定的IO请求流设定资源控制和优先级策略，以实现细粒度的IO控制。在具体的控制方面，IOFlow拦截了IO请求发送到网络上以及到达目标计算机时的两组IO驱动（Windows的SMB），在这两个拦截点应用策略，实现全局的IO路径控制。最终IOFlow实现了虚拟机的服务质量保证。

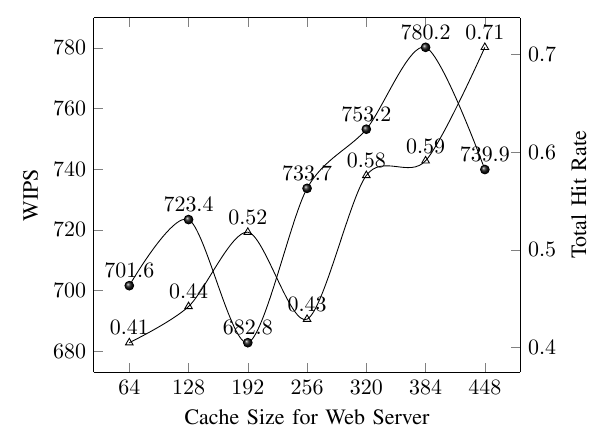


图 6：10万数据规模的测试结果

Marori[45]是IOFlow的一个扩展，它将缓存作为IOFlow中IO路径的路由目标，即可将指定的IO请求导向至缓存，从而实现全局的缓存分配和调度，提升虚拟机的性能。Marori也提供了一组缓存相关的API，允许分配并管理缓存。最终Marori可以在保证虚拟机服务质量的同时控制IO请求对缓存的使用，以提高缓存利用率并提升性能。

这些工作存在的局限性是它们都从网络的角度来进行IO的流控制，更类似于将SDN利用到了存储领域中。因此，它们的“全局控制”及“控制与数据分离”的概念只覆盖了IO栈在网络传输中的部分，即IO请求在网络上的交互过程，没有考虑到IO请求在具体的IO栈中的调度。因此，当应用集群规模较大、拓扑复杂，且有多个应用集群存在时，仍需要大量的人工介入，为每台虚拟机定义特定规则，覆盖IO栈的各方面，方可保证控制的准确性。

# 课题主要研究内容、预期目标

## 研究内容

课题主要从混合存储的视角切入，在混合使用SSD和机械硬盘的存储系统中感知云应用行为，监控云应用执行环境，提升性能并提供保障。系统的架构图如图 5所示。

主要分为如下3项主要研究内容工作：

### 云应用敏感的混合存储SSD资源分配

虚拟化环境（云计算环境）下一种典型应用模式是事务型应用，而事务型应用通常会在多台虚拟机上部署同一个应用的多个组件，并进行协作，这些组件对IO性能（带宽和IOPS）有着不同的需求，即应用本身的IO性能需求是不均衡的。

但同时，这些应用对IO性能的需求是稳定的，亦即在运行时的IO性能需求只会与负载相关等比例变化，但各组件的需求并不会变化。

一个典型的场景如部署在同一台物理服务器上的中小规模的Web应用程序。负载均衡器、Web服务器和数据库服务器部署在不同的虚拟机上，这三个组件访问的数据不同（负载均衡器和Web服务器主要存取静态内容，数据库服务器需要频繁操作数据库文件），模式也不同（读写请求所占比例不同，读写的数据大小和访问模式不同）。

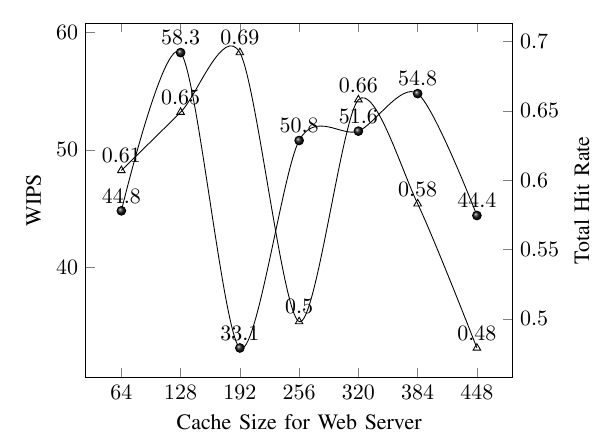


图 7： 10万数据规模的测试结果

先前的工作通常将每一台虚拟机作为一个独立的个体来看待，并在其上进行固态盘的缓存优化技术研究，主要优化缓存本身的特性，如命中率和利用率等，最终达到整体的平均命中率最大（亦即失效率最小），以达到性能优化的目的。但由于未考虑到应用本身的特点和部署应用的各虚拟机之间的关联，导致方法所导出的最优配置并不能达到应用整体的性能最优。而从另一方面看，简单的“先来先服务”模式的固态盘缓存构成的混合存储系统并不能达到最佳的性能。

早先的实验结果表明，对于一个简单的具有10万条数据的TPC-W基准应用，其静态内容大小约为100MB，数据库大小约为500MB。将其完全放置在机械硬盘（HDD）上时，测试得到的WIPS在150左右浮动，而完全放置在固态盘（SSD）上时，测试得到的WIPS在750~800之间浮动。但采用默认的调度策略将512MB大小的固态盘（已能基本满足应用需要）用作机械硬盘的缓存时，测试得到的WIPS在600~750之间浮动，如图 6所示。性能有一定提升但距离理论值仍有较大的距离，存在很大的优化空间，不同配置的性能差异也在2倍左右。

而当数据规模增大时，性能差异将更为明显。在数据规模在100万的情形下，全放置在HDD上时WIPS在10~15间浮动，全放置在SSD上时WIPS在120~130之间浮动，但使用512MB大小的固态盘作为缓存时，WIPS在30~60之间浮动，如图 7所示。性能差异在2倍左右。

另外，在这两个场景下，都存在某些数据点的命中率相同的情形，在命中率相同时最终的性能也不同。

在这一典型应用模式的驱动下，亟需一种云应用敏感的混合存储SSD资源分配，可以感知应用所部署的多个组件对IO性能的需求，并进行固态盘的缓存分配及数据放置，最终实现应用整体的性能最优。

### 云应用敏感的混合存储数据预取

虚拟化环境中另一个典型的应用模式是并行处理和大数据处理应用。典型场景如MPI并行处理应用以及Hadoop大数据处理应用。与传统的事务型应用不同，并行处理和大数据处理应用同样部署在多台虚拟机上，但其规模较大，无法由一台物理机承载，需要进行全局视角下的管理与控制。另外，这类应用对于IO性能的需求是均衡的，但在实际执行过程中由于任务分配和调度的不同会出现倾斜，即各个节点对IO性能的需求在运行时是动态改变的，这也会导致对应用整体性能的影响。

图 8：不同缓存分配对任务执行时间的影响（TestDFSIO）

如图 8所示，先前的实验结果表明，对于一个多台物理机构成的集群，在其上承载10个Hadoop节点，在完全不使用SSD缓存或是将所有数据放置在SSD上时，其任务的执行时间是相对稳定的。但调整部分节点的缓存大小，即提升节点本身的IO吞吐率，对整体Job的完成时间影响是不稳定的，同时带来的整体IO带宽的提升也是不稳定的。究其原因，主要是由于应用承载的任务不同及具体执行时的资源调度不同，导致应用在运行时对IO性能的需求动态变化，同时应用的多节点之间的资源分配缺乏协调，导致无法均衡地分配IO资源，以致某些节点成为瓶颈并拖慢整体的执行时间，导致优化效果不显著或不稳定。

因此，亟需一种云应用敏感的混合存储数据预取，从全局视角衡量应用本身的行为，并据此动态调整资源供给和预取数据，以达到性能调优及提升SSD使用效率的目标。

### 云应用敏感的混合存储全局控制

虚拟化环境不是独占的。将多租户环境，且各租户的应用类型各不相同的情况下，进行IO服务质量差分显得尤为重要。云计算平台需要为应用划分优先级并保障高优先级应用的性能，其中的重要部分是IO性能。

事务型应用和大数据处理应用具有截然不同的优先级。事务型应用是延迟敏感的，而大数据处理应用通常是离线处理，且对执行时间并不敏感。但从另一个角度看，部署在同一个云计算平台上的应用对IO性能的需求是类似的，单纯基于前述的IO性能优化并不能实现应用的服务质量保证，因此需要借助服务质量保障技术。

而从另一个角度看，从虚拟机到混合存储系统的IO路径是相对较长的，需要经过虚拟机本身的文件系统及底层IO调度器，Xen Hypervisor的从Domain U到Domain 0的IO调度器，以及Xen Hypervisor本身的底层IO调度器。这些IO调度器都决定了请求的优先级，但由于它们只能感知到有限的信息，不能从全局视角感知应用的拓扑、优先级以及性能保证等，从而引入管理开销和额外的性能开销。

因此，亟需一种云应用敏感的混合存储全局控制从全局视角感知应用的服务质量需求和优先级需求，同时统筹IO路径上的所有IO调度器，完成合理的资源调度和优先级选择，最终提升应用的整体性能。

## 预期目标

论文的预期目标主要包括：

1 实现云应用敏感的混合存储IO性能管理框架

2 研究一种云应用敏感的混合存储SSD资源分配方法，以优化事务型云应用的性能，降低应用的响应时间

3 研究一种云应用敏感的混合存储数据预取方法，以优化大数据处理型云应用的性能，减小任务的完成时间

4 研究一种云应用敏感的混合存储IO全局调度方法，以优化云应用端到端的性能，保障服务质量

5 进行实验对原型系统进行验证

6 撰写学术论文3~4篇

7 申请专利及软件著作权

# 拟采用的研究方法、技术路线、实验方案及其可行性分析

## 研究方法

针对云应用敏感的混合存储SSD资源分配，拟采用基于权重的方法。主要挑战在于如何刻画应用组件的IO需求及如何确定SSD分配策略。

针对云应用敏感的混合存储数据预取，拟采用基于性能建模及启发式的方法。主要挑战在于如何刻画不同类型数据对云应用性能的影响，以及如何具体进行数据预取。

针对云应用敏感的混合存储全局控制，拟采用基于策略的方法。主要挑战在于如何标识IO请求以及如何进行全局调度，并最终确定应用的优先级并规划IO资源。

## 技术路线

拟采用如下技术路线：

1 对国内外相关工作进行进一步调研与分析。

2 研究针对混合存储的全局管理框架，实现全局控制器、部署在虚拟机上的IO请求感知模块、部署在虚拟化主机上的数据映射模块以及各层次的IO调度器。

3 研究针对事务型云应用的SSD资源分配技术，研究相应的机制和策略。扩展IO请求感知模块，对应用中不同组件的IO需求进行感知；扩展全局控制器，根据不同组件的IO需求，确定同一应用部署的多台虚拟机的权重，最终反馈给数据映射模块进行具体的调整；扩展数据映射模块，实现缓存策略。

4 研究针对大数据处理型应用的数据预取技术，研究相应的机制和策略。扩展IO请求感知模块，对应用的行为模式和任务调度进行感知；扩展全局控制器，根据应用的行为模式和任务调度确定需要进行数据预取的节点，并确定数据对应用性能的影响；扩展数据映射模块，实现数据的预取策略。

5 研究混合存储的全局控制，研究相应的机制和策略。扩展IO请求感知模块，实现IO请求标记；接管虚拟机内部的IO调度器；扩展数据映射模块，感知IO标记并接管数据映射模块中的IO调度部分；接管Hypervisor底层的IO调度器。扩展全局控制器，从全局视角管理各层次的IO调度器。

6 应用到OnceCloud中，用类似生产系统的场景验证可行性与有效性。

## 实验方案

拟将软件工程技术研发中心自主开发的OnceCloud作为研究的目标云平台。虚拟机中部署的事务型应用主要基于Java平台，采用Apache Tomcat（自主研发的OnceAS）作为应用服务器，MySQL作为数据库，nginx作为负载均衡器。拟采用如下基准：

1 Bench4Q：TPC-W测试基准

2 RUBiS：在线交易基准

虚拟机中部署的并行处理应用主要基于MPI，采用MPICH作为MPI的标准实现。拟采用如下基准：

1 NPB（NAS Parallel Benchmark）：NASA提供的基准测试

2 HPL（High Performance Linpack）：高性能计算的Linpack基准测试

虚拟机中部署的大数据处理应用主要基于Apache Hadoop。拟采用如下基准：

1 TestDFSIO：Hadoop自带的HDFS性能基准测试

2 filebench：Hadoop自带的HDFS文件系统性能基准测试

3 Gridmix2：Hadoop集群的基准测试

针对虚拟机端到端的性能优化，拟采用SPEC的SPECvirt基准，来评价虚拟机上部署多个应用集群以及多多租户环境下的端到端性能。

## 可行性分析

固态硬盘与机械硬盘（或慢速存储设备）构成混合存储的应用模式得到了学术界和工业界长期的关注，近年来也有大量的文献关注虚拟化环境下固态硬盘对存储系统的性能优化。一些主流的应用模式，如使用固态硬盘作为机械硬盘的缓存等，已经得到充分验证，相对成熟，例如dm-cache和bcache已经并入了Linux内核的主线（mainline），由此衍生出的FlashCache成功用于Facebook。在另一方面，从全局视角控制IO的技术近年来也得到了充分关注，Microsoft Research的一个团队成功实现了IO请求的流控制（IOFlow），并进行了一系列扩展。

论文的工作主要弥补现有工作在应对多租户的云平台环境下多虚拟机部署的事务型应用和大数据处理应用性能优化方面的不足，前述的技术路线也被证明是可行的。

# 已有科研基础与所需的科研条件

软件工程技术研发中心拥有自主研发的云计算平台OnceCloud，可以为论文工作的开展提供有力支撑。OnceCloud基于高度定制化的Xen Hypervisor和Docker容器，可以较好地支撑内核级别的功能扩展，从而可以允许针对Xen Hypervisor和Docker底层的混合存储数据映射模块、IO调度模块的修改以及针对Linux的内核IO调度器。

本人参与了OnceCloud的研发工作，在之前的研发过程中负责分布式存储和网络功能虚拟化的相关工作，对OnceCloud的架构和关键模块具有深刻的理解，具有扎实的科研基础。

科研条件方面，软件工程技术研发中心拥有大规模的刀片服务器，并配置有一定数量的SSD，可以支撑虚拟化环境下混合存储的相关研究工作。

# 研究工作计划与进度安排

论文工作时间计划从2015年12月至2017年6月。预期计划和进展如下：

1 研发混合存储的IO性能管理框架，包括虚拟机中的IO请求监控代理、全局控制器、IO调度器及混合存储的数据映射模块。

2 研究针对事务型云应用的SSD资源分配技术，需要扩展IO请求监控代理以感知应用组件的IO请求，扩展数据映射模块并为全局控制器添加对应的策略。

3 研究针对大数据处理型应用的数据预取技术，需要扩展IO请求监控代理以感知应用的任务调度，扩展数据映射模块并为全局控制器添加对应的策略。

4 研究混合存储的全局控制，以实现云应用端到端的服务质量保证。扩展IO请求监控代理以标记IO请求，扩展数据映射模块并为全局控制器添加对应的策略。

5 进行实验验证

6 撰写博士论文，期间完成学术论文3~4篇，并申请专利及软件著作权

# 参考文献

1. Samsung. *Samsung Launches NAND Flash-based Solid State Disk for Mobile PCs*. 2006 [cited 2006; Available from: <http://www.samsung.com/semiconductor/insights/news/4236>.

2. Narayanan, D., et al., *Migrating server storage to SSDs: analysis of tradeoffs*, in *Proceedings of the 4th ACM European conference on Computer systems*. 2009, ACM: Nuremberg, Germany. p. 145-158.

3. Claburn, T. *Google Plans To Use Intel SSD Storage In Servers*. 2008 [cited 2015; Available from: <http://www.networkcomputing.com/storage/google-plans-to-use-intel-ssd-storage-in-servers/d/d-id/1067741>.

4. Reinsel, D. and J. Janukowicz, *Datacenter SSDs: Solid footing for growth.* IDC Corporation, Techical Report, 2008.

5. Wu, M. and W. Zwaenepoel, *eNVy: a non-volatile, main memory storage system*, in *Proceedings of the sixth international conference on Architectural support for programming languages and operating systems*. 1994, ACM: San Jose, California, USA. p. 86-97.

6. Caulfield, A.M., L.M. Grupp, and S. Swanson, *Gordon: using flash memory to build fast, power-efficient clusters for data-intensive applications*, in *Proceedings of the 14th international conference on Architectural support for programming languages and operating systems*. 2009, ACM: Washington, DC, USA. p. 217-228.

7. Andersen, D.G., et al., *FAWN: a fast array of wimpy nodes*, in *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*. 2009, ACM: Big Sky, Montana, USA. p. 1-14.

8. Chen, F., D.A. Koufaty, and X. Zhang, *Understanding intrinsic characteristics and system implications of flash memory based solid state drives*, in *Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*. 2009, ACM: Seattle, WA, USA. p. 181-192.

9. Feng, C., L. Rubao, and Z. Xiaodong. *Essential roles of exploiting internal parallelism of flash memory based solid state drives in high-speed data processing*. in *High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on*. 2011.

10. Grupp, L.M., et al., *Characterizing flash memory: anomalies, observations, and applications*, in *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*. 2009, ACM: New York, New York. p. 24-33.

11. Mohan, V., et al. *How I learned to stop worrying and love flash endurance*. in *Proceedings of the 2nd USENIX conference on Hot topics in storage and file systems*. 2010. USENIX Association.

12. Wu, G. and X. He, *Reducing SSD read latency via NAND flash program and erase suspension*, in *Proceedings of the 10th USENIX conference on File and Storage Technologies*. 2012, USENIX Association: San Jose, CA. p. 10-10.

13. Wu, Q., G. Dong, and T. Zhang. *Exploiting heat-accelerated flash memory wear-out recovery to enable self-healing SSDs*. in *USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage)*. 2011.

14. Lee, S., et al., *Lifetime management of flash-based SSDs using recovery-aware dynamic throttling*, in *Proceedings of the 10th USENIX conference on File and Storage Technologies*. 2012, USENIX Association: San Jose, CA. p. 26-26.

15. Pan, Y., G. Dong, and T. Zhang, *Exploiting memory device wear-out dynamics to improve NAND flash memory system performance*, in *Proceedings of the 9th USENIX conference on File and stroage technologies*. 2011, USENIX Association: San Jose, California. p. 18-18.

16. Samsung. *What is Hybrid HDD?* 2007 [cited 2015; Available from: <http://www.samsung.com/jp/consumer/learningresources/hdd/whitepapers/pop_white_hybridhdd.html>.

17. Bisson, T. and S.A. Brandt. *Reducing Hybrid Disk Write Latency with Flash-Backed I/O Requests*. in *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2007. MASCOTS '07. 15th International Symposium on*. 2007.

18. Intel. *Intel Rapid Storage Technology*. 2012 [cited 2015; Available from: <http://www.intel.com/content/www/us/en/architecture-and-technology/rapid-storage-technology.html>.

19. Intel. *What is Intel Smart Response Technology?* 2012 [cited 2015; Available from: <http://www.intel.cn/content/dam/www/public/us/en/documents/technology-briefs/smart-response-technology-brief.pdf>.

20. Dell. *Business Laptops*. 2015 [cited 2015; Available from: <http://www.dell.com/us/business/p/laptops>.

21. Chen, F., S. Jiang, and X. Zhang, *SmartSaver: turning flash drive into a disk energy saver for mobile computers*, in *Proceedings of the 2006 international symposium on Low power electronics and design*. 2006, ACM: Tegernsee, Bavaria, Germany. p. 412-417.

22. Kgil, T. and T. Mudge, *FlashCache: a NAND flash memory file cache for low power web servers*, in *Proceedings of the 2006 international conference on Compilers, architecture and synthesis for embedded systems*. 2006, ACM: Seoul, Korea. p. 103-112.

23. Koltsidas, I. and S.D. Viglas, *Flashing up the storage layer.* Proc. VLDB Endow., 2008. **1**(1): p. 514-525.

24. Taeho, K., D. Roberts, and T. Mudge. *Improving NAND Flash Based Disk Caches*. in *Computer Architecture, 2008. ISCA '08. 35th International Symposium on*. 2008.

25. Qing, Y. and R. Jin. *I-CASH: Intelligently Coupled Array of SSD and HDD*. in *High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on*. 2011.

26. Makatos, T., et al., *Using transparent compression to improve SSD-based I/O caches*, in *Proceedings of the 5th European conference on Computer systems*. 2010, ACM: Paris, France. p. 1-14.

27. Soundararajan, G., et al., *Extending SSD lifetimes with disk-based write caches*, in *Proceedings of the 8th USENIX conference on File and storage technologies*. 2010, USENIX Association: San Jose, California. p. 8-8.

28. Chen, F., D.A. Koufaty, and X. Zhang, *Hystor: making the best use of solid state drives in high performance storage systems*, in *Proceedings of the international conference on Supercomputing*. 2011, ACM: Tucson, Arizona, USA. p. 22-32.

29. Mesnier, M., et al., *Differentiated storage services*, in *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*. 2011, ACM: Cascais, Portugal. p. 57-70.

30. 刘洋, *层次混合存储系统中缓存和预取技术研究*. 2013, 华中科技大学.

31. Van Hensbergen, E. and M. Zhao, *Dynamic policy disk caching for storage networking.* IBM Research Report, 2006.

32. Koller, R., et al., *Write policies for host-side flash caches*, in *Proceedings of the 11th USENIX conference on File and Storage Technologies*. 2013, USENIX Association: San Jose, CA. p. 45-58.

33. Arteaga, D. and M. Zhao, *Client-side Flash Caching for Cloud Systems*, in *Proceedings of International Conference on Systems and Storage*. 2014, ACM: Haifa, Israel. p. 1-11.

34. Linux. *Linux 3.9*. 2013 [cited 2013; Available from: <http://kernelnewbies.org/Linux_3.9>.

35. Linux. *Device-Mapper Cache*. 2013 [cited 2015; Available from: <https://www.kernel.org/doc/Documentation/device-mapper/cache.txt>.

36. Facebook. *Releasing Flashcache*. 2010; Available from: <https://www.facebook.com/notes/mysql-at-facebook/releasing-flashcache/388112370932/>.

37. Amazon. *Amazon Elastic Compute Cloud*. 2015 [cited 2015; Available from: <http://aws.amazon.com/cn/ec2/>.

38. Microsoft. *Microsoft Azure*. 2015 [cited 2015; Available from: <https://azure.microsoft.com/zh-cn/>.

39. Byan, S., et al. *Mercury: Host-side flash caching for the data center*. in *Mass Storage Systems and Technologies (MSST), 2012 IEEE 28th Symposium on*. 2012.

40. Meng, F., et al., *vCacheShare: automated server flash cache space management in a virtualization environment*, in *Proceedings of the 2014 USENIX conference on USENIX Annual Technical Conference*. 2014, USENIX Association: Philadelphia, PA. p. 133-144.

41. Shamma, M., et al., *Capo: recapitulating storage for virtual desktops*, in *Proceedings of the 9th USENIX conference on File and stroage technologies*. 2011, USENIX Association: San Jose, California. p. 3-3.

42. Koller, R., A.J. Mashtizadeh, and R. Rangaswami. *Centaur: Host-Side SSD Caching for Storage Performance Control*. in *Autonomic Computing (ICAC), 2015 IEEE International Conference on*. 2015.

43. Gartner, *Magic Quadrant for Solid State Arrays*. 2015.

44. IDC, *Worldwide All-Flash Array and Hybrid Flash Array 2014-2018 Forecast and 1H14 Vendor Shares*. 2014.

45. Stefanovici, I., et al., *Software-defined caching: managing caches in multi-tenant data centers*, in *Proceedings of the Sixth ACM Symposium on Cloud Computing*. 2015, ACM: Kohala Coast, Hawaii. p. 174-181.

46. SmartBear. *The Cost of Poor Web Performance* 2012 [cited 2015; Available from: <http://blog.smartbear.com/web-performance/the-cost-of-poor-web-performance-infographic/>.

47. Gulati, A., I. Ahmad, and C.A. Waldspurger, *PARDA: proportional allocation of resources for distributed storage access*, in *Proccedings of the 7th conference on File and storage technologies*. 2009, USENIX Association: San Francisco, California. p. 85-98.

48. Angel, S., et al., *End-to-end performance isolation through virtual datacenters*, in *Proceedings of the 11th USENIX conference on Operating Systems Design and Implementation*. 2014, USENIX Association: Broomfield, CO. p. 233-248.

49. Luo, T., et al., *S-CAVE: effective SSD caching to improve virtual machine storage performance*, in *Proceedings of the 22nd international conference on Parallel architectures and compilation techniques*. 2013, IEEE Press: Edinburgh, Scotland, UK. p. 103-112.

50. Lv, Y., et al., *Hotness-aware buffer management for flash-based hybrid storage systems*, in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 2013, ACM: San Francisco, California, USA. p. 1631-1636.

51. Miranda, A. and T. Cortes, *CRAID: online RAID upgrades using dynamic hot data reorganization*, in *Proceedings of the 12th USENIX conference on File and Storage Technologies*. 2014, USENIX Association: Santa Clara, CA. p. 133-146.

52. Thereska, E., et al., *IOFlow: a software-defined storage architecture*, in *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*. 2013, ACM: Farminton, Pennsylvania. p. 182-196.

1. 数据来源：

   Intel固态盘性能数据 http://ark.intel.com/zh-cn#@SolidStateDrives

   Western Digital机械硬盘性能数据 http://www.wdc.com/wdproducts/library/SpecSheet/ [↑](#footnote-ref-1)