尽管元数据迁移对系统有以上两个方面的负面影响。但是，元数据请求统计比例、BWMMS 元数据分布管理的动态重分布特性、以及 BWMMS 的负载决策方法，使得其影响尽可能地减到最小。

尽管元数据迁移可能导致集中请求成为跨服务器请求，影响元数据的逻辑性。但是，1）仅在文件移动请求中，目录迁移后还可能继续使用。统计结果表明，文件移动请求所占比例<0.01%[Hendricks2006]，对系统的影响甚微；2）尽管极小比例的文件移动请求将可能导致跨服务器请求的出现。但是，由于 BWMMS 元数据请求分布管理的动态重分布特性，在通过 MS2 访问 bar 下面的、分布在 MS1 的文件 foo1、foo2 之前，MS1 可能已经根据元数据活跃性，释放掉 foo1、foo2 的管理权限。这样，可以通过请求分布决策，foo1、foo2 可以重新分布到 MS2，减少 bar 被迁移带来的跨服务器请求数量。由于与具体应用密切相关，较难通过实验评估元数据请求分布管理的动态重分布特性对元数据迁移的积极影响程度。

对于公平性的影响，系统能够通过有效的负载信息分析，选择处理请求的服务器。通过 BS 全局的决策，元数据将从预期负载较重的服务器，向预期较轻的服务器迁移，避免服务器负载不平衡的加重，可能进一步促进服务器间负载的平衡。

5.3.7 迁移协议的容错分析

迁移协议容错分析基于的模型是多个进程构成的、通过异步消息进行通信的进程集合。尽管系统的可能故障包括网络因故障分割为多个子网、服务器硬件故障、或者系统软件故障等多个方面，其最终表现为在既定时间内，消息发起进程没有得到消息接收进程的答复。在没有收到答复情况下，消息发起进程可能重发请求，也可能放弃请求。协议必须能够保证系统不会因为一对进程间的消息重发、或者消息丢弃而出现不一致。根据图 5.2，协议容错分析包括 MS1-BS，BS-MS2，MS2-SN 和 MS1-SN 之间的通信过程。

1. MS2 和 SN 之间。MS2 需要将缓存中的元数据写回到设备。在成功更新设备后，更改元数据的分布信息。当在指定时间内没有收到 SN 的回应时，如果 MS2 重新写设备，此时系统仍然仅有 MS2 拥有该元数据的宿主权限，所以不存在覆盖掉其他服务器写回的有效元数据的风险；如果 MS2 放弃写设备，它认为协议处理出错，不会释放元数据的宿主权限。即使 SN 已经将元数据写回到设备，后面的写操作覆盖掉它，也不会丢失有用信息。
2. BS 和 MS2 之间。如果 BS 再次发送消息，要求 MS2 释放元数据权限。MS2 收到重发的消息，检查自己缓存的元数据分布信息后，将由于没有宿主权限而不再写设备，仅返回“正确处理”结果给 BS。如果 BS 放弃重发，它将返回“协议错误”给 MS1。如果在发送“协议错误”结果给 MS1 后，收到 MS2 返回的“协议正确处理”消息，BS 按照协议正确处理，更新元数据的分布信息，等待

MS1 收到“协议处理错误”结果后的重试。

3. MS1 和 BS 之间。如果 MS1 再次发送消息，BS 根据元数据分布信息，返回“协议正确”处理结果。如果 MS1 放弃重发消息。那么，它将返回“请求处理出错”给应用，应用根据需要可以重新发起，或者放弃。
4. MS1 和 SN 之间，由于前面的协议过程已经正确更改 MS2、BS 和 MS1 的元数据分布信息。此时，仅有 MS1 拥有元数据写权限。所以，MS1 和 SN 间任意次的读元数据请求都将获得有效的元数据信息，不会出现不一致。

总之，元数据迁移协议不仅能够保证从存储设备正确读写元数据，还能够保证“同一个活跃元数据，有且仅有一个宿主元数据服务器”。并且，由于 BS 的信息是所有 MS 记录的宿主权限项信息的并集。所以，元数据分布信息管理机制和策略能够支持 BS 故障、MS 都不能有故障，或者 BS 存活、任意 MS 故障的情况。对于 BS 和任意 MS 同时故障的支持，还需要进行深入的研究。

5.4 与两阶段提交协议对比分析

BWMMS 与两阶段提交协议的对比分析包括消息数量和更新磁盘次数，对比的请求采用跨服务器的文件创建、删除、以及涉及 4 个元数据的文件移动请求的最坏情况，对比结果如表 5.3 所示，其中的数据对格式是“(通信消息数量,设备内容更新次数)”。由于通过网络刷写大量数据到存储服务器的时间占整个协议处理时延的比例非常大，设备内容更新次数的变化，反映协议的效率变化。

表 5.3 BWMMS 动态迁移协议和两阶段提交协议对比结果

	两阶段提交协议	动态迁移协议	时间减少比例
创建文件	(2,2)	(1,1)	50%
删除文件	(2,2)	(2,1)	50%
移动文件	(6,4)	(6,3)	25%

从表中可以看出，BWMMS 动态迁移协议至少可以获得 25%的协议时延减少。并且，BWMMS 动态迁移协议不需要在协议过程中以同步写方式保存重要状态，其错误恢复也不需要多个服务器通过复杂的错误恢复协议协同，对系统的正常性能影响较小。

5.5 元数据迁移协议的影响评估

本节从不同比例的跨服务器请求对系统影响的角度，评估元数据迁移协议的影响。评估用例采用修改过事务构成的 postmark。原始 postmark 是每进行一次读文件所有内容/追加写文件内容的同时，还需要创建一个指定大小的文件/删除已有文件。这种模式中，创建/删除必定出现，称之为“100%”模式。后面的表述用“α”表示一次事务中创建/删除出现的概率,，比如α=5%表示 100 个事务中将有 5 个创建/删除的出现。本测试评估了α为 0%、3%、5%、20%和 100%共 5 种情况，其结果如图 5.3 所示。

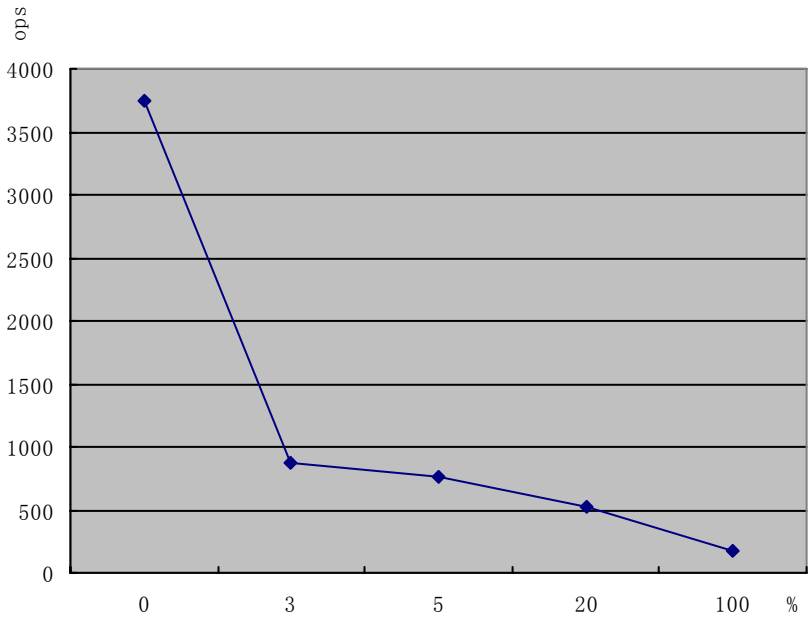

图 5.3 跨服务器请求比例的影响评估结果

表 5.4 是各种比例测试的服务器开销情况。

表 5.4 元数据迁移对服务器负载的影响

	0%	3%	5%	20%	100%
元数据迁移请求数目	0	31	48	59	65
MS 与 BS 平均通信时间(ms)	0.84	0.91	1.51	2.51	3.57
MS 聚合服务时间(s)	44.23	48.78	60.64	80.10	200.43
平均 RPC 处理速度	50,017.94	47,569.93	39,511.64	36,522.79	28,745.29

表中数据表明，随着文件创建/删除比例的增加，各个服务器需要迁移元数据的请求数量增加，导致 BS 用于元数据迁移请求处理的开销增大，影响 BS 处理 MS 的元数据分布决策请求，导致“MS 与 BS 的平均通信时间”呈增加趋势。并且，由于 MS 在处理元数据迁移请求过程中，对 AS 请求的响应不够，导致 AS 请求超时的概率增加，“平均 RPC 处理速度”呈现降低的趋势。NFS 的 trace 分析结果[Hendricks2006]表明，需要通过迁移完成处理的跨服务器请求的比例大约在 0.01%，其本身对系统的影响不会很大，但阻塞正常请求。所以，如果 BS 采用独立线程处理迁移请求，它们对正常元数据分布请求的阻塞影响将大幅度降低，元数据请求的处理能力将得到很大提高。

5.6 本章小结

在集群环境中，更改文件系统的跨服务器请求的原子性将影响文件系统的一致性。尽管其所占比例非常小，不到 1%。但是，传统的两阶段提交等协议，对系统性能和可恢复性的影响都比较大，探讨轻量级原子性保证协议显得非常必要。

以集中共享存储架构和对称元数据服务器结构为支持，BWMMS 提出“通过改变活跃元数据的宿主，完成元数据在服务器间的迁移，跨服务器请求涉及到的活跃元数据集

中到单个服务器”的动态迁移协议。在本地文件系统技术支持下，元数据请求由单个服务器集中处理。对比分析表明，相对于传统的两阶段提交协议而言，在最坏情况下，BWMMS 动态迁移协议也可以获得 25%左右的处理时延降低。并且其错误恢复更加简单，对系统的影响更加微小。

本章最后通过实验评估元数据迁移对系统的影响。对于跨服务器请求比例非常小的应用，BWMMS 采用的轻量级元数据迁移协议，对系统整体性能的影响非常微弱。结果同时表明，跨服务器请求的比例将影响系统的处理能力，要求在未来研究中进一步加强对不同特征的应用模式研究，促进系统进一步的演进。

第六章 元数据请求并发与同步控制

请求的并发和同步控制是分布式系统的重要研究内容。如何控制请求的并发，避免系统出现死锁是分布式系统必须解决的重要问题。在 BWMMS 中，通过 BS 转发元数据迁移请求的轻量级跨服务器请求原子性保证协议，加重系统的死锁可能。如何通过控制元数据请求的并发和同步，完成死锁的检测和消除是本章的主要研究内容。

BWMMS 包括元数据分布信息管理和文件系统元数据访问两个层次的并发。通过将元数据分布信息和文件系统元数据的同步控制分离，BWMMS 在元数据分布信息层进行请求的同步控制，通过分析系统可能的请求并发情况，尽可能地并发元数据分布信息请求和元数据请求，提升系统的扩展能力。

6.1 问题描述

分布式系统的死锁问题，来源于分布式计算环境中的并发进程对共享资源的竞争[Gray1981][Massey1986][Lee2001][Singhal1989][Krivokapic1999][Kshemkalyani1999]。死锁预防、死锁避免、以及死锁检测和消除[Wang1998][Terekhov1999][Gonzales1999][Ling2006]是现有解决分布式系统死锁问题的三种主要方式。分布式死锁检测和消除策略以其不需要预防分布式死锁的出现、动态地检测和消除死锁的特点，成为分布式系统死锁问题解决的主要趋势。

明确请求之间存在的时间和因果关系[Lamport1978][Schwarz1994]，有助于分布式系统的全局镜像维护、死锁问题的检测和消除。在分布式计算领域，已有大量的针对请求因果关系的研究[Fishburn1985][Dielh1992][Katz1990][Meldal1991][Schmuck1991][King2003][Zhu2003]。

投机执行可以提升系统的处理性能[Chang1999][Fraser2003][Nightingale2006][Jefferson1987][Franklin1996][Zhang1999]。它记录进程间存在的数据依赖。在进程依赖的数据不可用时，它根据预测的数据提前执行，提高进程的并行度。当依赖的数据出现最终结果后，只有在预测错误时，它才根据最终结果修正投机执行的进程。

在 BWMMS 中，分布式元数据请求通过元数据迁移，转换成集中请求处理。为简化元数据迁移协议，元数据迁移请求通过集中决策服务器转发。第 4 章已经验证 AS 的元数据请求间的并发。本章主要关注元数据迁移请求和正常元数据请求之间的并发控制问题。系统可能存在的请求并发如图 6.1 所示。