综合表 7.2 和图 7.2，BWMMS 能够很好支持，更多客户端以共享模式访问同一文件集，元数据服务表现出较强的扩展能力。

7.3.2 私有使用模式的扩展能力评估

相对共享模式而言，私有使用模式是系统更为常见的使用方式。在这种模式中，用户主要访问各自私有目录中的文件，用户间没有文件的共享。本节评估固定服务器规模下、用户负载变化带来的聚合吞吐率的变化，以及固定用户负载情况下、元数据服务器规模变化带来的聚合吞吐率变化两种情况。

7.3.2.1 负载变化的扩展能力评估

系统包括 6 个元数据服务器。测试采用 postmark[postmark2002]进行，测试文件集的大小范围为 4kB-16kB，每次文件 I/O 的块大小为 4kB。在每个客户的测试文件集生成阶段，postmark 首先创建 10 个目录，然后在这 10 个目录中总共创建 10000 个文件。在事务处理阶段，它将进行 50000 次事务[3]请求，事务处理速度通过“指定的事务数（50000）/事务阶段所用时间”计算而得。评估共测试 1、2、4 和 6 个客户端共 4 种情况，结果如表 7.3 所示。

表 7.3 私有模式聚合吞吐率随客户端变化

客户端数目	1	2	4	6
聚合吞吐率(ops)	367	522	935	1,320
吞吐率加速比	1	1.4223	2.5477	3.5967

从加速比看，在私有模式下，BWMMS 还不能支持聚合元数据吞吐率随客户端的线性扩展能力。

表 7.4 私有模式既定服务器的 MS 负载情况

客户端数目	1	2	4	6
MS 聚合服务时间(s)	0.80	3.68	15.79	40.72
平均 MS 和 BS 通信时间(ms)	6.65	7.30	7.98	8.74
平均元数据迁移时间(ms)	17.86	20.89	35.14	55.51

但是，结合表 7.4 的服务器负载情况的分析结果表明，其受限于后端绑定服务器当前的实现结构。从表 7.4 的“MS 和 BS 的平均通信时间”和“MS 聚合服务时间”看，BS 的决策结果能够将元数据请求分布到多个服务器，元数据请求负载将由多个服务器分担，服务器进程的聚合服务时间表现出增长幅度放缓的趋势。

但是，“元数据迁移的平均时间”表明，客户端数目的增加，将导致活跃元数据数目的增加，增加元数据服务器间交互的复杂度。由于 BS 仅以各个元数据服务器当前活跃元数据数目为决策参数，并且元数据缓存管理策略没有及时地将本 MS 真实活跃的元数据数目提供给 BS，元数据分布策略将关联元数据分布到不同元数据服务器的概率增大，

[3] Postmark 的事务定义见第三章分布策略对比评估部分。

系统的跨服务器请求数目出现增长趋势，需要 BS 转发的请求数目增加，导致平均元数据迁移时间的增长。

7.3.2.2 服务器变化的扩展能力评估

为验证元数据服务器增加带来的系统聚合元数据处理能力的提升，本节评测在 7 个客户端下元数据服务器数目从 1、2、4 到 6 个时的系统聚合事务吞吐率的变化趋势。7 个客户端采用 postmark 进行测试。测试文件集由位于 10 个目录下的 10000 个文件构成，每个文件的大小范围为 4kB-16kB，每次文件 I/O 的块大小为 4kB，测试完成 50000 次事务请求。系统聚合事务吞吐率如表 7.5 所示。

表 7.5 私有模式聚合吞吐率随服务器变化

服务器数目	1	2	4	6
聚合吞吐率(ops)	328	633	997	1,564
吞吐率加速比	1	1.9299	3.0396	4.7683

表 7.6 的元数据服务器负载情况，能够说明影响客户端聚合元数据吞吐率的原因。

表 7.6 私有模式既定客户端的 MS 负载情况

服务器数目	1	2	4	6
MS 聚合服务时间(s)	84.72	58.42	52.87	43.78
MS 和 BS 平均通信时间(ms)	9.32	7.43	8.03	9.07
平均元数据迁移时间(ms)	0	15.48	15.87	15.86

从表 7.6 来看，通过元数据分布策略的作用，元数据请求分布到多个服务器，使得元数据服务器的聚合服务时间呈现降低趋势。

随着服务器数量的增加，元数据服务器之间的交互复杂化。由于 BS 当前的单一进程结构实现，MS 的请求在 BS 串行执行，导致“MS 和 BS 的平均通信时间”呈现增加趋势。同时，出于平衡服务器负载目的，请求分布策略将增加跨服务器请求的数目，元数据迁移出现的概率增加，导致“元数据迁移的平均时间”呈现出增加的趋势。这将进一步阻塞 BS 的处理，导致系统性能的降低。

综合以上分析，对私有使用模式而言，尽管不能提供线性的扩展能力，BWMMS 仍然能够支持客户端和服务器数目增加带来的聚合元数据吞吐率的增加。目前其扩展能力表现不够好的关键原因在于，BS 的实现还不能很好地支持请求的并行处理。

7.3.3 创建删除空文件的扩展能力评估

尽管统计结果表明，文件创建/删除的比例非常少，仍可能存在需要大量文件创建/删除的应用。本节评估系统对密集的文件创建/删除应用的支持情况。

每个客户端在私有目录下创建 10000 个空文件，创建完成后，删掉这些空文件。测试使用 Linux® 的 time 工具收集所用时间。客户端的请求处理速率 OPER(i)=10000/TIME(i)，其中，i 表示第 i 个客户。TIME（i）为测试所耗的时间，单

位为秒。OPER（i）是通过计算得到的 I/O 速率，单位为每秒完成的操作数目(ops)。将每次测试所有客户端的请求处理速率相加，得到聚合请求处理速率：

$$\mathrm{OPER} = \sum_{i=1}^{n} \mathrm{OPER}(i), \text{其中} n \text{为客户数量}$$

系统测试 6 个元数据服务器规模下，客户数量分别为 1、2、4 和 6 个的创建和删除的聚合吞吐率，结果如表 7.7 所示。

表 7.7 创建/删除聚合吞吐率随客户端变化

客户端数目	1	2	4	6
创建聚合吞吐率(ops)	263.50	264.13	235.54	235.16
创建吞吐率加速比	1	1.002	0.894	0.892
删除聚合吞吐率(ops)	222.27	201.82	193.18	187.64
删除吞吐率加速比	1	0.908	0.869	0.844

表 7.8 是创建和删除空文件测试过程的服务器负载情况。图 7.3 是 BS 服务进程的 CPU 占用率随时间的变化曲线图。

表 7.8 创建/删除的 MS 负载情况

客户端数目	1	2	4	6
MS 聚合服务时间(s)	0.16	0.33	0.75	1.12
MS 和 BS 平均通信时间(ms)	0.570	0.601	0.752	0.741

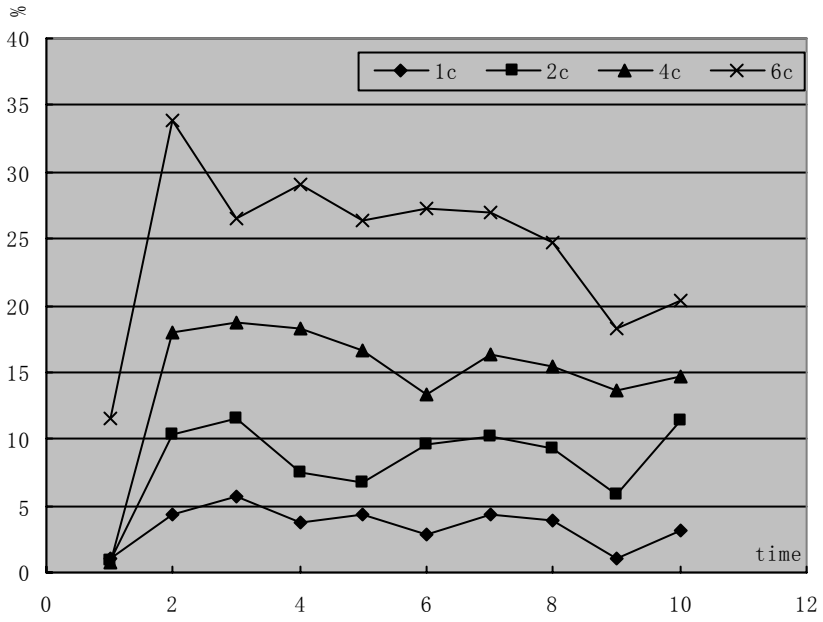

图 7.3 创建/删除 BS 的 CPU 占用率

结合表 7.8 的创建/删除文件的服务器负载和图 7.3 的 BS 服务进程的 CPU 占用率可以看出，由于文件创建过程需要为新创建的文件决定宿主映射决策，元数据缓存将不能起到提高处理效率的作用，每一个文件创建都需要 BS 参与。所以，随着应用负载的增加，BS 的负载相应地增加。BS 的处理能力和元数据创建的处理过程，限制文件创建聚

合吞吐率的提升，文件创建的聚合吞吐率随客户端负载增加而降低。

同样，由于跨服务器的文件删除请求需要通过 BS 完成元数据的迁移。尽管可能的元数据迁移请求数目很少，但它可能阻塞 BS，其他请求不能得到处理。所以，随着客户端数量的增加，系统聚合的文件删除吞吐率表现出下降的趋势。

总之，目前 BWMMS 对具有频繁的文件创建/删除的应用支持还不够,需要根据应用的实际需要进行优化。

7.4 对新兴应用的扩展支持评估

生物信息计算是近几年新兴的高性能计算应用。本节通过生物计算广泛采用的 BLAST（Basic Local Alignment Search Tool），评估 BWMMS 的元数据服务对新兴的生物计算应用的支持能力。生物处理仪获取规模庞大、文件大小较小的原始数据，经过计算处理，获得待比对的序列。BLAST 是用来比对生物序列的一级结构(如不同蛋白质的氨基酸序列或不同基因的 DNA 序列)的算法，使用待比对序列进行序列相似度的比对。BLAST 主要表现为固定数量文件的反复读请求。最坏情况下，为完成一条序列的比对，它需要读取目标数据库所有的记录。其需要的计算时间很长，长时间的文件读写请求表现出庞大的元数据请求。

本测试采用的序列库的大小为 2GB。通过 BLAST 软件包提供的实用程序 FORMATDB 的参数控制，序列库被格式化为 60 组子库，每组包括后缀分别为 phr、pin 和 psq 的 3 个文件,每组文件的大小和约为 40MB。目标库共有 180 个文件,总和为 2.4GB 左右。测试使用 4 个客户端完成 211,359 条序列的比对，总共需要 435,214,997 次元数据访问 RPC 通信。测试采用 Linux®的 time 实用程序收集比对过程需要的时间。为明确系统的扩展能力，分别测试元数据服务器数量为 1、2 和 3 个的情况。其结果如表 7.9 所示：

表 7.9 BLAST 聚合处理时间

服务器数目	1	2	3
BLAST 聚合时间(h)	260.32	230.27	208.64
聚合时间递减速度(%)	0	11.54	19.85

从 BLAST 运行总时间看，其时间随元数据服务器数目增加的降低比例不很明显，大约在 10%左右。但是，收集到的时间分布表明，BLAST 绝大部分工作在用户态进行。

通过系统态时间，扣除与元数据访问不相干的时间，得到大致的元数据访问时间，才能够更合理地表现元数据请求开销。可以从系统态时间扣除的主要时间是数据访问时间。从统计到的数据访问看，应用读取文件数据的请求数目大致在 210,000 个，每次读取 1MB 的文件。测试初期获得的 BWFS 文件数据读取速度是 37.99MB/s。根据这两个值进行计算，读取文件数据的时间大概是 1.54 小时左右。所以，从系统态时间中扣除这部分时间后，得到元数据访问的大致时间，如表 7.10 所示。

表 7.10 BLAST 元数据访问时间分析

服务器数目	1	2	3
系统态时间(h)	5.89	4.96	4.01
计算的元数据访问时间(h)	4.35	3.42	2.47
元数据时间递减速度(%)	0	21.38	31.82

元数据服务器的服务时间是各个元数据服务器处理元数据请求花费的时间。结合它来分析系统的元数据处理，能更好地说明元数据服务能力。测试的聚合服务时间如表 7.11 所示：

表 7.11 BLAST 元数据服务器聚合服务时间分析

服务器数目	1	2	3
MS 聚合工作时间(s)	20,475. 15	15,726.22	12,895.66
聚合工作时间递减速度(%)	**0**	**23.19**	**37.02**
MS 与 BS 通信时间(s)	0.138	0.264	0.369
MS 与 SN 通信时间(s)	3,269.36	6,386.92	6,801.37
MS-SN 时间所占比例(%)	**15.97**	**40.61**	**52.74**

从服务器聚合服务时间看，元数据服务器数目的增加能够带来近 20%的 MS 聚合服务时间的减少。BS 当前的单一进程结构，将导致元数据服务器的请求的串行处理，MS 与 BS 的通信时间将随服务器数目的增加而增加，这反映在“MS 与 BS 通信时间”。尽管绝对值的变化不大，但其随服务器的变化比例的增长速度非常快。同时，由于元数据访问的小读小写特征，频繁的元数据读写，将导致元数据读写时间随服务器数目增加而延长，这反映在“MS 与 SN 通信时间”。“MS-SN 时间所占比例”是“MS 与 SN 的通信时间”占整体元数据服务时间的比例，它表现出快速增加趋势。这表明，通过改善 BS 的实现结构、优化元数据服务器的元数据读写性能，系统的元数据服务效率将得到很大提升，可能获得近线性的元数据服务扩展能力。

7.5 本章小结

元数据存储服务和元数据请求服务关键问题的解决方案得到原型实现。系统实现平台是 RedHat® Linux® 8.0，由 1 个 SN、1 个 BS、多个 MS 和多个 AS 构成。

本章评估其对传统应用和新兴应用的扩展支持能力。

针对传统应用的评估表明，从结构和策略上讲，BWMMS 的对称服务器结构和动态元数据请求分布策略，能够很好地支持系统规模的扩展需要。系统能够将负载尽可能均衡地分布到各个元数据服务器，提升系统的处理能力。目前的系统原型实现存在限制扩展能力的不足，且对具有大量频繁的文件创建/删除的应用的扩展支持很差。

针对生物信息计算应用 BLAST 的评估结果表明，服务器数目增加能够带来元数据请求处理时间近 20%左右的减少。在未来工作中，通过改善 BS 的实现结构、优化元数据读写访问，系统将可能获得近线性的元数据服务扩展能力。