Computer Science and Application 计算机科学与应用, 2021, 11(11), 2609-2618
Published Online November 2021 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2021.1111264

Hans汉斯

面向天河二号Lustre文件系统优化实践

曾凌波，杜云飞，万　文，颜　辉*，钟　英

中山大学国家超级计算中心，广东 广州

收稿日期：2021年10月2日；录用日期：2021年11月1日；发布日期：2021年11月8日

摘　要

近年来，随着高性能计算技术的飞速发展，Lustre文件系统作为高性能计算重要组成部分越来越受到重视，对于Lustre文件系统元数据服务的高可靠性一直是研究的重点，针对Lustre文件系统元数据高可靠性的研究，本文设计了一种双MDT冗余的架构保证了元数据服务的高可靠性；同时随着应用复杂度的提高和计算规模的增加，对Lustre文件系统的性能要求也越来越高，特别是在处理海量小文件和I/O密集型应用时，对Lustre文件系统元数据的I/O性能提出了更高的要求，为了提高元数据的I/O性能，本文通过升级MDT元数据底层硬件设备来提升元数据的I/O性能和提升整体文件系统的I/O性能，满足新的应用对文件系统IO的要求。

关键词

文件系统，元数据，高可靠，优化

Optimization Practice of Lustre File System for Tianhe No. 2

Lingbo Zeng, Yunfei Du, Wen Wan, Hui Yan*, Ying Zhong

Nation Supercomputer Center in Guangzhou, Sun Yat-sen University, Guangzhou Guangdong

Received: Oct. 2nd, 2021; accepted: Nov. 1st, 2021; published: Nov. 8th, 2021

Abstract

In recent years, with the rapid development of high-performance computing technology, the Lustre file system has been paid more and more attention as an important component of high-performance computing. The high reliability of the metadata service of the Lustre file system has always been the focus of research. This paper has designed a dual MDT redundant architecture to

*通讯作者。

文章引用: 曾凌波, 杜云飞, 万文, 颜辉, 钟英. 面向天河二号 Lustre 文件系统优化实践[J]. 计算机科学与应用, 2021, 11(11): 2609-2618. DOI: 10.12677/csa.2021.1111264

ensure the high reliability of metadata services; at the same time, as the application complexity increases and the calculation scale increases, the performance requirements of the Lustre file system are also getting higher and higher, especially when dealing with massive small files and I/O intensive applications. I/O performance of the metadata of the Lustre file system has raised higher requirements. In order to improve the I/O performance of metadata, this article upgrades the underlying hardware equipment of MDT metadata to improve the metadata I/O performance and to improve the overall file system I/O performance to meet the new application requirements for file system IO.

Keywords

File System, Metadata, High Reliability, Optimization

1. 引言

伴随着高性能计算的不断发展和普及，高性能计算已经被广泛的运用于海洋数值预报、能源勘探、材料分析、工业仿真等领域，对工业转型和国民经济的发展具有重要的推动作用，高性能集群已经变得越来越重要。以海洋数值预报为例，随着海洋和气候变化研究的不断深入，海洋观测资料的数量和种类不断增多，海洋数值预报模式正逐步朝着更高分辨率(分辨率越高，网格越精细)、更多物理过程(方程组方程数目增加)和更快计算速度的方向发展[1] [2]。海洋数值预报模式的快速发展也对高性能计算集群的性能提出了新的要求。Lustre [3]文件系统作为高性能计算集群中重要的组成部分，它的稳定性、可靠性以及性能优化一直都是业界研究的重点。

Lustre 文件系统作为大规模高性能计算集群数据存储组件，元数据服务作为文件系统客户端访问后端文件无法绕开的一个环节，元数据服务故障将直接导致整个文件系统不可用，所以为了能提供持续的元数据服务，必须要对 Lustre 文件系统元数据架构进行高可用设计提高文件系统整体的可用性；同时，Lustre 文件系统作为共享文件系统，当客户端请求量到达一定规模时文件系统性能瓶颈表现明显用户体验极差，针对这一问题，本文对 Lustre 文件系统元数据 I/O 优化策略[4]进行了分析和讨论。

2. Lustre 文件系统介绍

Lustre 是一个开源、全局单个命名空间、符合 POSIX 标准的分布式并行文件系统；Lustre 具有高可扩展性、高性能两大特性，能够支持数万客户端系统、PB 级存储容量以及数百 GB 的聚合 I/O 吞吐量，其基本结构如图 1 示。

Lustre 文件系统[3]由元数据服务器(MDS)、对象存储服务器(OSS)、客户端三部分组成。MDS 负责向客户端提供整个文件系统的元数据(元数据存储在 MDT 中)，管理整个文件系统的全局命名空间，维护整个文件系统的目录结构、用户权限以及文件系统元数据一致性。OSS 负责对象数据的存储，将 I/O 数据保存到由它管理的后端对象存储设备(OST)中。客户端通过标准的 POSIX 接口向用户提供对文件系统的访问，用户通过客户端可以透明的访问整个文件系统中的数据。当客户端读写文件时，从 MDS 得到文件信息，从 OSS 中得到数据。

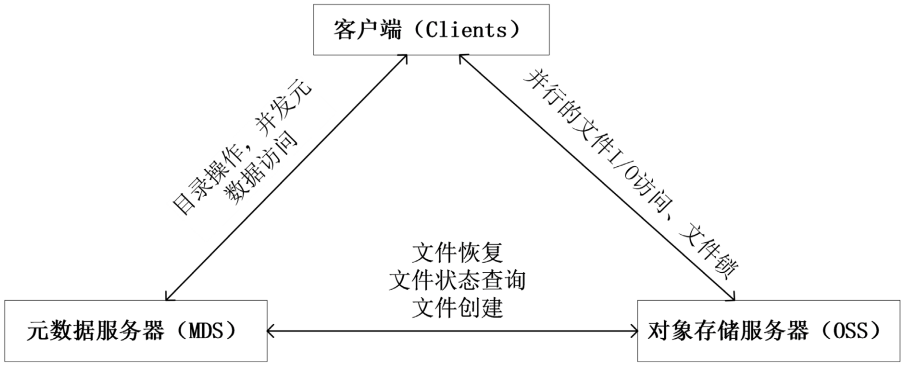

Figure 1. Lustre basic structure diagram
图 1. Lustre 基本结构图

3. Lustre 文件系统元数据高可用设计

众所周知 Lustre 文件系统客户端访问后端对象存储服务时必须先请求元数据服务，客户端从元数据服务中获取到所访问文件的元数据然后才能真正访问后端文件；从客户端访问后端文件的请求路径不难发现元数据服务对于整个文件系统至关重要，一旦元数据服务发生故障将直接导致整个文件系统服务不可用或者用户后端数据丢失，所以在生产系统中必须要考虑元数据服务的高可用架构以及 MDT 的冗余设计。

3.1. 元数据高可用设计介绍

元数据服务高可用涉及两个方面：元数据服务器的冗余设计和元数据目标 MDT 的冗余设计。Lustre 文件系统官方推荐部署方式如图 2 所示，使用的是两个元数据服务器(采用 Active/Standby 模式)共享一个元数据设备的模式对外提供元数据服务，这种方式解决了一个元数据服务器宕机导致的元数据服务停服的情况，但是没能彻底解决元数据服务高可用的问题，一旦共享的元数据设备出现故障一方面文件系统服务不能正常使用，另外可能会导致整个文件系统所有用户数据丢失。

为了解决单 MDT 单点故障的问题，本文设计了双 MDT 冗余备份架构替换单一 MDT 的方案，解决了元数据目标 MDT 单点故障问题，真正保证了整个元数据服务的高可用。

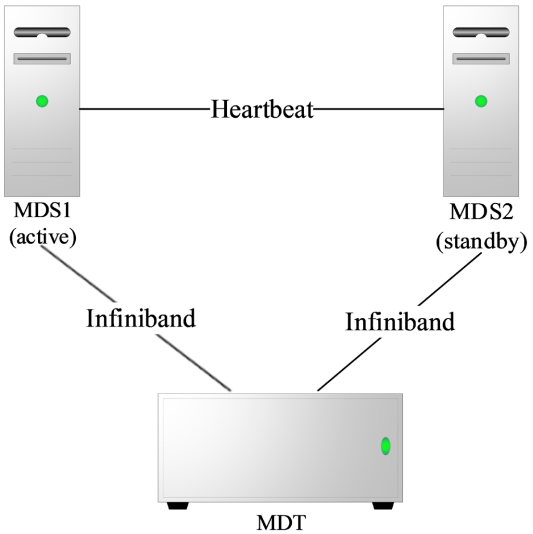

Figure 2. Single MDT architecture diagram
图 2. 单 MDT 架构图

图 3 中主备元数据服务器各自对应一个 MDT 设备，主备 MDT 设备之间的数据同步通过 drbd [5]实现，主备元数据服务器状态通过 heartbeat [6]心跳监控实现；新架构解决了元数据服务器状态自动监控和元数据自动同步复制，真正实现元数据服务的高可用。

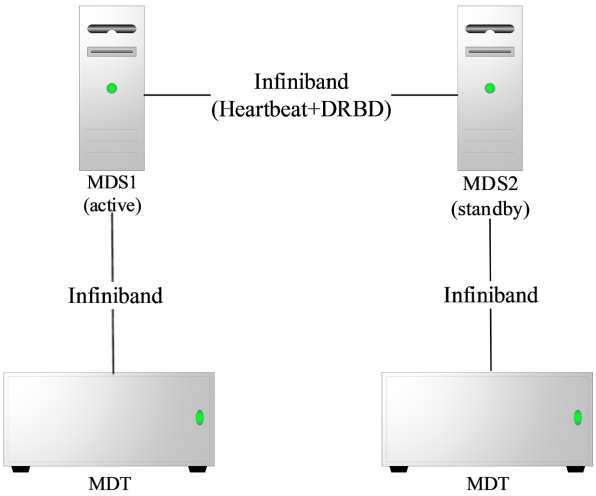

Figure 3. Double MDT structure diagram
图 3. 双 MDT 结构图

3.2. 元数据复制实现

图 3 中两个元数据服务器的元数据需要进行实时同步保证主备 MDT 元数据一致性，本文通过 drbd 实时数据同步技术来实现主备元数据的同步。

drbd 具体同步流程如下：主节点写入的数据通过 drbd 设备存储到主节点的存储设备中，同时，数据会通过网络发送到备份节点相应的 drbd 设备，最终将数据写入备份节点的存储设备中。

drbd 工作流程如图 4 示。左边为主元数据服务器(实线箭头)，右边为备份元数据服务器(虚线箭头)。当主元数据服务器接收到元数据请求时，drbd 会将接收到的元数据复制一份。一份存储到主元数据服务器对应的 MDT 设备中，另一份通过网络(TCP/IP 协议)传输复制到备份元数据服务器的 drbd 设备。备份数据服务器的 drbd 设备接收到元数据后，将元数据存储到备份元数据服务器对应的 MDT 设备中。

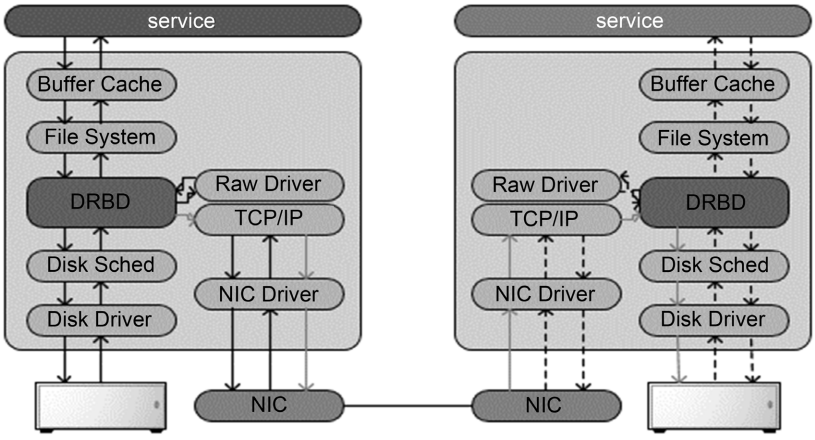

Figure 4. Drbd work flow chart
图 4. Drbd 工作流程图

在元数据同步中，主元数据服务器上的 drbd 首先会把元数据写入到对应的 MDT 设备中，同时通过网络将需要更新的元数据复制到备份元数据服务器的 drbd 设备，最终将元数据写入到备份元数据服务器对应的 MDT 设备中至此完成元数据同步。

3.3. 元数据服务器状态监测

本文中元数据服务器冗余策略采用 active/standby [7]模式，主元数据服务器和备份元数据服务器需要定期进行心跳监测确认彼此状态，通过在主备元数据服务器上部署 heartbeat 实现主备元数据服务器状态自动监测。

主备元数据服务器间通过心跳信息和心跳反馈来确认彼此状态，如果主元数据服务器在一定时间内未收到备份元数据服务器的心跳信息，则判定备份元数据服务器离线，结合监控系统将备份元数据服务器离线消息通知给管理员，管理员及时对备份元数据服务器进行处理；如果备份元数据服务器在一定时期内未收到主元数据服务器的心跳信息，则判断主元数据服务器离线，备份服务器开始接管主元数据服务器对外提供元数据服务，同时结合监控系统将主元数据服务器离线的消息通知给管理员进行处理。

主备元数据服务器之间通过 heartbeat 进行心跳监测，需要特别注意脑裂[8]情况的出现，脑裂会使元数据写入混乱，导致主备元数据不一致造成用户数据丢失，在实际生产环境中可以从以下三个方面来防止脑裂的出现：

1) 同时使用串口和以太网连接，使用两条心跳线路，实现心跳线路冗余。

2) 检测到脑裂时强行关闭一个心跳节点，例如，在备份元数据服务器上发现心跳故障通过 IPMI 发送关机命令到主节点。

3) 做好脑裂的监控报警，在问题发生时能快速人为介入将影响降低到最低。

3.4. 实验结果

图 3 中展示了重新设计后的元数据服务器架构，主备 MDS 服务器对应主备两个 MDT 设备，针对新的元数据架构做了大量的实验来获取和对比 drbd 主备元数据同步时间的开销以及元数据服务切换时文件系统恢复所花时间。

3.4.1. Drbd 主备元数据同步时间开销

创建不同数目文件对比主备 MDT 进行元数据同步的时间开销，Lustre 文件系统中每个元数据大约为 2.5 KB，drbd 配置文件中设置传输速率为 200 M/s，根据创建的文件数可以估算出需同步的元数据规模以及理论同步所需时间；在实际生产系统中 drbd 的传输速率只有默认设置传输速率的 1/3 即 60 M/s。

需同步元数据规模 = 创建文件数量 * 单个元数据大小

理论同步时间 = 需同步元数据规模/drbd 理论传输速率

实际同步时间 = 需同步元数据规模/drbd 实际传输速率

通过上面 3 个公式能计算出创建不同规模文件元数据理论同步时间和实际同步时间，具体数据如表 1 示：

Table 1. Drbd metadata synchronization time cost table
表 1. Drbd 元数据同步时间开销表

创建文件数量(个)	需同步数据规模(M)	理论同步时间(s)	实际同步时间(s)
10,000	24.4	0.122	0.488
100,000	244	1.22	4.88

Continued

1,000,000	2437	12.1856	51.424
10,000,000	24,371	121.856	487.424

实际生产系统中，文件系统每秒平均创建文件的个数约为 50 个，根据表 1 中的数据表明，同步这 50 个文件的元数据可以忽略不计，在实际生产系统中元数据同步时间开销基本趋于实时，完全满足实际生产需要。

3.4.2. 元数据服务切换文件系统恢复时间开销

当元数据服务故障时主备元数据进行服务切换时，Lustre 文件系统客户端需要和新的元数据服务器进行重连接，文件系统恢复的时间开销基本花在了客户端和新元数据服务的重连上，表 2 中给出了不同规模客户端在元数据服务发生切换时文件系统恢复所需时间。

Table 2. File system recovery schedule when switching metadata services of clients of different sizes
表 2. 不同规模客户端下元数据服务切换时文件系统恢复时间表

文件系统客户端数量(个)	文件系统恢复时间(s)
50	10
100	23
200	41
500	108
1000	211
2000	420

表 2 中对不同规模客户端文件系统恢复时间进行了对比，通过数据可以发现随着客户端数量的增加文件系统恢复时间也是在递增的，在 2000 个客户端规模下文件系统恢复的时间也只需要 420 s 左右，完全满足实际生产系统的要求。

3.4.3. 测试结果总结

综合 3.4.1 和 3.4.2 两小节的测试结果，从元数据复制时间开销到元数据服务切换文件系统恢复时间都是能满足实际生产系统的需求的。

4. Lustre 文件系统元数据 IO 优化策略

4.1. 背景介绍

以往高性能计算应用对文件的读写访问特征主要注重聚合带宽性能，对元数据的性能要求不高[9]，Lustre 文件系统采用机械磁盘阵列作为 MDT 就能满足大部分应用需求。然而，随着高性能计算技术的发展和普及以及大数据、深度学习和高性能计算应用的融合，高性能应用中 I/O 密集型应用增多，继续使用机械磁盘阵列作为 MDT 会使得元数据服务成为整个文件系统的瓶颈[10]，机械盘阵 MDT 的 I/O 性能无法支持大规模的 I/O 请求，导致客户端文件访问出现卡顿、应用运行时间增加，更严重的情况会导致 I/O 请求超时，引发 Lustre 文件系统不稳定，甚至出现文件系统故障。

4.2. 元数据性能优化策略

天河 2 集群现有存储系统使用 NetApp 机械盘阵[11]作为 MDT 设备，在使用过程中，随着计算规模

的扩大，大数据、深度学习和高性能计算应用的深度融合，Lustre 文件系统通过横向扩展来提升 I/O 带宽，在整个存储系统 I/O 吞吐量增加的情况下，MDT 设备的 I/O 性能成为了整个存储系统的瓶颈[12]。

磁盘设备 IOPS 性能体现在对小文件的随机读写，现在主流的盘阵主要使用 NL-SAS 的机械硬盘作为存储介质，相比 SATA 和 SAS 接口的机械盘，性价比相对较高。机械硬盘通过磁头寻道和盘片的旋转来实现数据的存取，它的工作原理决定了它的随机存取数据性能比较差，相反，固态硬盘无论在顺序存取性能还是随机存取性能相比机械硬盘优势明显。固态硬盘除了性能优势明显外，相比机械硬盘能耗更低、防震抗摔性强、体积更小的优势，不过固态硬盘价格贵、容量小、寿命短是它的缺点。

结合固态硬盘和机械硬盘的优缺点，本文分别选用 SATA SSD、NetApp SAN 盘阵两种存储设备作为 MDT 进行 IOPS 性能对比测试。其中，SATA SSD 盘阵由 8 块 1.92TB 的 IntelSATASSD 组成，NetApp SAN 盘阵由 10 块 3TB NL-SAS 9600 转机械磁盘组成，硬盘阵列模式皆为 RAID 10 模式。选用 mdtest 元数据基准测试工具，通过对比测试来进一步验证 SSD 在 IOPS 性能上的优势。

4.3. 实验结果

根据元数据性能优化策略，本文选用 SATA SSD、NetApp SAN 盘阵两种存储设备作为 MDT 进行 IOPS 性能对比测试，运用 mdtest 元数据基准测试工具对目录创建、目录 stat、目录删除、文件创建、文件 stat、文件删除六种操作进行测试，取六种测试结果总和的平均值为 MDT 的 IOPS 性能。

图 5、图 6 中分别为目录和文件测试对比，使用 SATA SSD 的 MDT 在目录创建、目录 stat、文件创建、文件 stat 和文件删除五种操作测试中性能都远远好于使用 NetApp SAN 盘阵的 MDT，只有在目录删除操作测试中性能比 NetApp SAN 盘阵差。

Figure 5. Comparison of directory operation test
图 5. 目录操作测试对比图

Figure 6. Comparison of file operation test
图 6. 文件操作测试对比图

图 7 中为 mdtest 测试的 IOPS 平均值，测试结果显示，在不同规模下，使用 SATA SSD 的 MDT 的 IOPS 性能相比使用 NetApp SAN 盘阵的 MDT 的 IOPS 性能提升了近 60%~70%，性能提升明显。

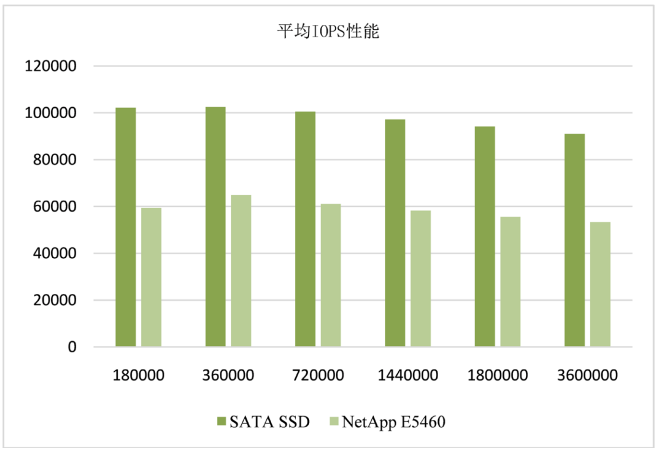

Figure 7. Comparison of average IOPS performance
图 7. 平均 IOPS 性能对比

图 8 显示的是不同规模下两种存储介质 mdtest 测试时后端磁盘设备利用率的变化曲线，从图中可知，在文件系统 IOPS 达到峰值之后，采用 SATA SSD 作为 MDT，在上述测试规模中，磁盘的平均利用率大概在 20%左右，说明在此测试规模下 SSD 磁盘 RAID 组远远没有达到性能瓶颈。而采用 NetApp SAN 盘

阵作为 MDT，在上述所有的测试规模中，磁盘平均利用率一直为 100%，说明测试所给的 IO 请求已经足够多，NetApp SAN 盘阵在 I/O 请求处理上已经满负荷，此时盘阵性能存在瓶颈。

通过以上测试结果可知，使用 SATA SSD 的 MDT 总体性能表现优秀，虽然目录删除性能较差，但在日常的生产使用过程中，涉及到目录删除的操作很少。因此，Lustre 文件系统可以使用固态硬盘作为 MDT 的存储介质，MDT 并不需要大容量的存储空间，同时满足 MDT 高 IOPS 的需求。

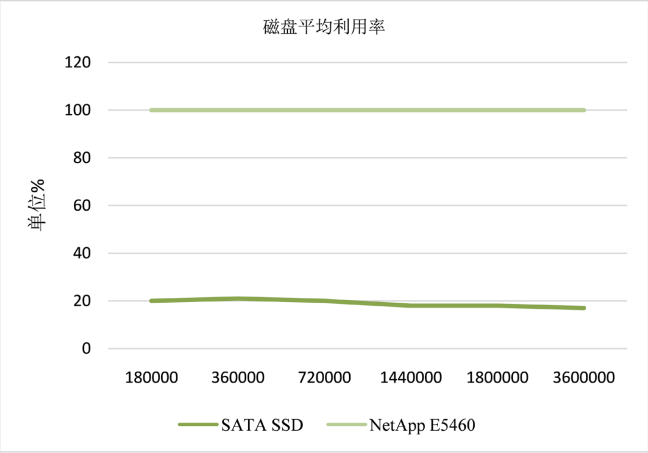

Figure 8. Comparison of average disk average utilization
图 8. 平均磁盘平均利用率对比

5. 结束语

针对 Lustre 文件系统的元数据高可用方案的设计与实施从根本上解决了 Lustre 元数据服务单点故障问题，保证了文件系统服务的高可用和文件系统数据的高可靠；另外结合高性能应用的 I/O 特征的分析，对 Lustre 元数据 MDT 物理设备进行了升级，通过提高 MDT 元数据设备的 I/O 性能提升整体文件系统的 I/O 性能，确保了 I/O 密集型应用在天河 2 系统上的顺利运行。

目前对于 Lustre I/O 性能的优化目前只是在硬件层面对设备进行了升级，下一步工作将针对 Lustre 软件层进行优化，进一步提升文件系统整体的 I/O 性能。

基金项目

国家重点研发计划支持项目(2018YFC1406205)；国家自然科学基金项目(U1811464)；广东省引进创新创业团队项目(2016ZT06D211)和广东省省级科技计划项目(2019B020208014)。

参考文献

[1] Shu, Q., Qiao, F.L., Song, Z.Y. and Yin, X.Q. (2013) A Comparison of Two Global Ocean-Ice Coupled Models with Different Horizontal Resolutions. *Acta Oceanologica Sinica*, **32**, 1-11. https://doi.org/10.1007/s13131-013-0335-z

[2] 宋振亚, 刘卫国, 刘鑫, 苏天赟, 刘海行, 尹训强. 海量数据驱动下的高分辨率海洋数值模式发展与展望[J]. 海洋科学进展, 2019, 37(2): 161-170.

[3] Liang, J. and Nie, R.H. (2015) Lustre File System Based on Object Storage. *Computer Engineering and Design*, **36**, 1666-1670.

[4] Chen, Q., Chen, Z.N. and Jiang, J.H. (2014) MDDS: A Method to Improve the Metadata Performance of Parallel File System for HPC. *Journal of Computer Research and Development*, **51**, 1663-1670.

[5] Chen, J.-X. and Liu, X.-J. (2011) Analysis and Improvement of Distributed Replicated Block Device. *Computer Engineering and Design*, **32**, 3599-3601, 3806.

[6] Wang, H. and Sun, X.-Y. (2012) Application of Heartbeat and Drbd in Large-Capacity OLT. *Modern Electronics Technique*, **35**, 131-134, 137.

[7] Li, Y. (2020) Research and Implementation of High Available Clusters Based on Linux. *Computer Applications*, **39**, 35-38.

[8] Gong, T.-N. and Zhou, S.-M. (2012) High Availability Cluster of Linux Based on DRBD. *Computer and Information Technology*, **20**, 63-65.

[9] Li, L.L., Wu, W.G. and Sun, L.X. (2012) Performance Optimization of Fine-Grained I/O in Parallel File System Lustre. *Computer Engineering and Applications*, **48**, 88-92.

[10] Liu, G.M., Zou, D. and Zhang, C. (2009) Research on Lustre-Oriented Storage Acceleration with Solid State Disk. *Journal of Computer Research and Development*, **46**, 371-375.

[11] Mao, X.-F., Hou, X.-M. and Ma, H. (2016) VLBI Storage System of Computer Based on Disk Array. *Computer Systems & Applications*, **25**, 107-111.

[12] Li, Z., Zhou, E.Q. and Liao, X.K. (2009) Filter Cache: A Method for Improving I/O Performance of Lustre File System. *Journal of Computer Research and Development*, **46**, 71-77.