Lustre

概述：

Lustre提供了一种新的模块化存储框架，包括各种存储管理功能、网络、锁定和大规模存储目标，所有这些都旨在支持小到非常大的集群的可伸缩集群文件系统。 Lustre这个名字体现了“Linux”和“Cluster”。 Lustre侧重于在大型计算机集群中使用的可伸缩性，但通过组成系统的模块的实现和部署方面的微小变化，同样可以很好地服务于较小的商业环境。 Lustre运行在不同的网络上，包括目前的以太网和Quadrics。

Lustre起源于卡内基梅隆大学的Coda项目的研究。 它已经引起了存储行业的几家公司的兴趣，它们为设计做出了贡献，并资助了一些实施。现在它是Sun微系统公司开发的通用公共许可(GNU)，Lustre克服了许多传统分布式记录（文件）框架中的执行、范围和可扩展性问题。

在Luster文件系统上提出了一个零假设工作，这是HPC（高性能计算)的一个步骤。 该假设接受了来自各种门户网站的文章的研究范围，并以Sun团队在文件系统研究领域开发的更好的方法来处理问题，并构建了一个更好的可伸缩文件系统：Lustre。

由于Lustre框架具有很大程度的可扩展结构，Lustre系统在逻辑超级计算中表现突出，而且在制造、天然气和石油、金融业和丰富的媒体中也表现突出。Lustre的用户有POSIX接口,可以同时访问相互记录对象。

解决的问题：

Lustre是一种非常重要的存储架构，它将构建的、公开的和创造性的协议合并为明确的、网络神经信息存储和易于访问的解决方案。

Lustre在一堆令人满意的框架中给出了较高的I/O turnout，并且利用信息的情况进一步提供了从信息在实际存储中的位置上的自由，从关键点故障中保护，以及从集群中立即检索。 Lustre可接受框架重构和主机或系统停电。

架构：

Lustre具有三个主要组件：元数据服务器（MDS），对象存储服务器（OSS）和客户端，它是基于对象的文件框架。

Lustre将块设备应用于记录信息和元数据存储，并且每个块设备都可以由一个Lustre管理来监督。Lustre文件系统的总数据限制是所有单个OST限制的总和。通过标准POSIX I / O框架调用访问用户并同时使用数据。

MDS（元数据服务器）提供与元数据相关的服务，而元数据客户端（MDC）则是这些服务的受益者。每个文件系统将元数据目标与一个元数据服务器（MDS）关联。每个文件的元数据都由MDT存储。元数据存储的参数是记录元数据，例如记录名称，注册表层次结构和访问方式。

Lustre框架的配置数据由管理服务器（MGS）提供。对象存储服务器（OSS）的作用是提供信息和发现的块设备。对象存储客户端（OSC）是框架提供的功能的用户/客户端。单个或多个OST（对象存储目标）与OSS关联，并且实际记录数据对象由OSTs文件系统保存。

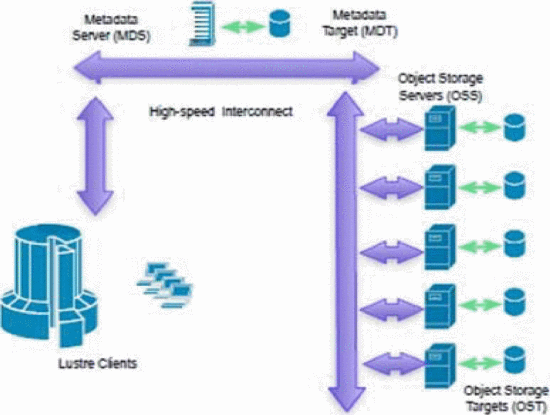


图 1lustre架构

由lustre拓展的架构：

1客户Lustre：-Lustre文件系统是我们测试平台的基础。Lustre服务器与Linux虚拟文件系统之间的接口由Lustre客户端进行。这将承载应用程序并访问OST以获取所需数据。

2前端处理器-前端处理器连接到端口。前端处理器的主要任务是一次服务Lustre客户的请求。单个请求可以同时由前端处理器接受和确认。处理器数量越多，发生的同步I / O服务就越多。

3缓存-这是最快的内存组件。缓存负责处理Lustre客户端生成的许多请求。每个前端处理器都分配有固定大小的缓存，通过使用缓存分区来将缓存分配给多个处理器。在某些情况下，缓存作为单个大内存组件工作。如果Lustre客户端请求的数据位于高速缓存中，则前端处理器将数据直接从高速缓存提供给客户端。

4后端处理器-这些是磁盘和后端处理器之间的接口。

5磁盘-这些是已驻留数据的实际物理存储组件。磁盘存储所有来自Lustre客户端的数据，并根据Lustre客户端的请求提供服务。这些磁盘根据需要使用各种RAID方法进行配置。

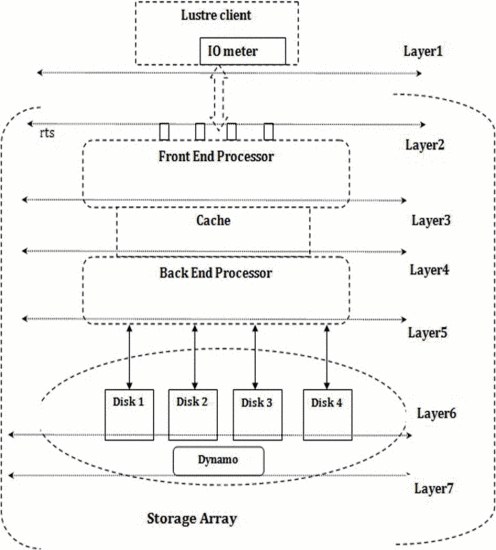


图 2由luster拓展的架构

关键技术：

Lustre通过提供多个抽象来增强执行和可扩展性。

通过元数据服务器(MDS)查找文件，并在文件系统级别将文件作为对象处理。 所有与记录相关的操作都由MDS（元数据服务器）提供，例如记录创建、记录、文件搜索、协调文件到对象存储目标(OST)的I/O请求以及目录和文件属性操作。基于价值的文件框架事务记录由MDS以及元数据更改保存，并且由于该文件系统的设备和网络故障影响一个MDS不影响系统框架的执行而使系统状态（集群）和恢复正常。

与其他文件系统框架一样，Lustre框架对于每个习惯文件、catalos、典型符号连接和特殊文件都有一个相同的i节点。 正常的记录i-节点在OSTs上保存对象引用，这些对象引用保留记录信息，而不是引用真实的记录信息。 在记录系统框架中，新的记录导致分配一个i节点并建立其基本属性。 在Lustre中创建一个新记录会导致用户请求一个MDS，该MDS为记录创建i-node，然后请求对象存储目标(OSTs)来创建真正保留文档的对象信息。

对象的元数据作为扩展属性保存在i-节点中。 通过RAID方法，在OSTs上添加的对象保留与记录相关的信息，并对少数对象存储目标进行标记。 在Object Storage Target的内部，信息被真正地追求并写入OBD。 对于最近制作的文件，用户和OSTs之间的I/O是通过请求OBD跟踪和写入信息来完成的。 元数据服务器的更新是在需要与新记录连接的额外名称空间更改时完成的。

实际保留和分配到对象存储目标和基本的基于对象的磁盘的持久冲突鼓励了系统的进步，它可以在智能磁盘驱动器的最新时代的外观上给予额外的执行增强，在设备中提供基于对象的分配信息管理服务。 基于集群的记录系统与少数存储设备生产者有效地协调，在磁盘驱动器设备中创建合并的对象基础磁盘支持。 这个场景实际上相当于SCSI标准引领智能设备的路径，它将很大一部分硬件通信传输到设备接口和驱动硬件控制器。 如此敏锐的OBD感知设备可以为目前Lustre的建立提供即时的执行升级。

结论：

Lustre是一种最新的存储架构，具有许多优点和分布式文件框架，在面向网络的计算环境中为企业网络、计算集群和共享数据提供了可伸缩性、更高的性能和灵活性。 Lustre为文件I/O使用对象存储模型。 文件系统更改和高层文件的事务记录由元数据服务器(MDS)维护)。 分散对象存储目标(OSTs)负责真正的文件框架I/O，并负责与邻居或网络存储设备(称为基于对象的磁盘(OBDs)进行接口)。 Lustre是一个开放的标准，例如Linux、XML、LDAP、SNMP，可以很容易地从开源库和现有的文件系统框架中访问，从而提供一个可靠、强大和可伸缩的文件系统框架。 为了消除停机时间和提高文件系统的可访问性，Lustre使用了复制、复杂、尖端的故障转移恢复方法。这样，Lustre获得了更高的性能和生产力。

Lustre的发明者正在有效地与硬件生产商合作，帮助发明即将到来的智能存储设备时代，硬件增强可以进一步将数据处理从分布式文件框架的软件部分卸载到存储设备本身。 该文件系统在下一代搜索引擎等搜索项目中有了进一步的实现。因为以该文件系统为核心，可以实现完整的可伸缩系统。