**Распределенные файловые системы**

Две главные цели.

1. Сетевая прозрачность.

Самая важная цель - обеспечить те же самые возможности доступа к файлам, распределенным по сети ЭВМ, которые обеспечиваются в системах разделения времени на централизованных ЭВМ.

1. Высокая доступность.

Другая важная цель - обеспечение высокой доступности. Ошибки систем или осуществление операций копирования и сопровождения не должны приводить к недоступности файлов.

Файловый сервис - это то, что файловая система предоставляет своим клиентам, т.е. интерфейс с файловой системой.

Файловый сервер - это процесс, который реализует файловый сервис.

Пользователь не должен знать, сколько файловых серверов имеется и где они расположены.

Так, как файловый сервер обычно является обычным пользовательским процессом, то в системе могут быть различные файловые серверы, предоставляющие различный сервис (например, UNIX файл сервис и MS-DOS файл сервис).

**Lustre** — распределённая файловая система массового параллелизма, используемая обычно для крупномасштабных кластерных вычислений. Название Lustre является контаминацией, образованной словами Linux и cluster. Реализованный под лицензией GNU GPL, проект предоставляет высокопроизводительную файловую систему для кластеров с десятками тысяч узлов сети и петабайтными хранилищами информации.

Файловые системы Lustre используются в компьютерных кластерах, начиная от небольших кластеров рабочих групп и заканчивая масштабными географически распредёленными кластерами. Пятнадцать суперкомпьютеров из мирового «Топ-30» используют файловые системы Lustre, в том числе самый быстрый в мире суперкомпьютер — K computer (на 2011 год).

Файловые системы Lustre могут поддерживать десятки тысяч клиентских систем, десятки петабайт (PBs) памяти для хранения данных и пропускную способность ввода-вывода в сотни гигабайт в секунду (GB/s). Благодаря высокой масштабируемости Lustre, такие области бизнеса, как провайдеры Интернет, финансовые организации, индустрия нефти и газа устанавливают файловые системы Lustre в своих центрах обработки данных.

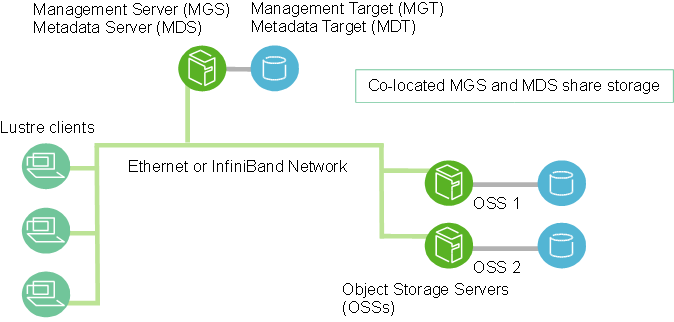
**История**

Архитектура файловой системы Lustre была разработана в рамках исследовательского проекта в 1999 Питером Браамом (Peter Braam), старшим научным сотрудником Университета Карнеги — Меллон. Браам покинул его, чтобы основать собственную компанию Cluster File Systems, реализовавшую Lustre 1.0 в 2003 году. В 2007 году Sun Microsystems поглотила Cluster File Systems и включила Lustre в свой пакет программного обеспечения для высокопроизводительных систем, намереваясь использовать преимущества технологии Lustre в файловой системе Sun ZFS и операционной системе Solaris. В ноябре 2008 года Браам покинул Sun Microsystems для работы над другой файловой системой, оставив разработку архитектуры Lustre Эрику Бартону (Eric Barton) и Андреасу Дилджеру (Andreas Dilger).

В 2010 году Oracle поглотила Sun и к ней перешёл бизнес по поддержке и разработке Lustre. В апреле 2010 года Oracle объявила о намерении ограничить платную поддержку для новой Lustre 2.0, предоставляя её только на оборудовании Oracle или оборудовании, разработанном одобренными Oracle поставщиками. При этом Lustre осталась доступной всем пользователям под лицензией GPL и существующие заказчики Lustre 1.8 продолжили получать техподдержку от Oracle.

В декабре 2010 Oracle объявила о прекращении развития Lustre. Выпуск Lustre 1.8 переведен на обслуживание только для сопровождения, что создало неопределенность по поводу дальнейшего развития файловой системы. Согласно этому объявлению разработка и поддержка новых версий Lustre осуществляется сообществом разработчиков, включая Whamcloud, Xyratex, OpenSFS, European Open Filesystems (OFS) SCE и другими. В том же году Эрик Бартон и Андреас Дилджер покинули Oracle и основали стартап Whamcloud, сфокусированный на развитии Lustre.

В июле 2012 года Whamcloud была приобретена Intel. Это произошло после того, как Whamcloud выиграла контракт для министерства энергетики США на расширение Lustre в рамках первой фазы проекта экзафлопных вычислений под названием FastForward на период до 2018 года. После этого OpenSFS перевела все контракты по разработке Lustre в Intel.

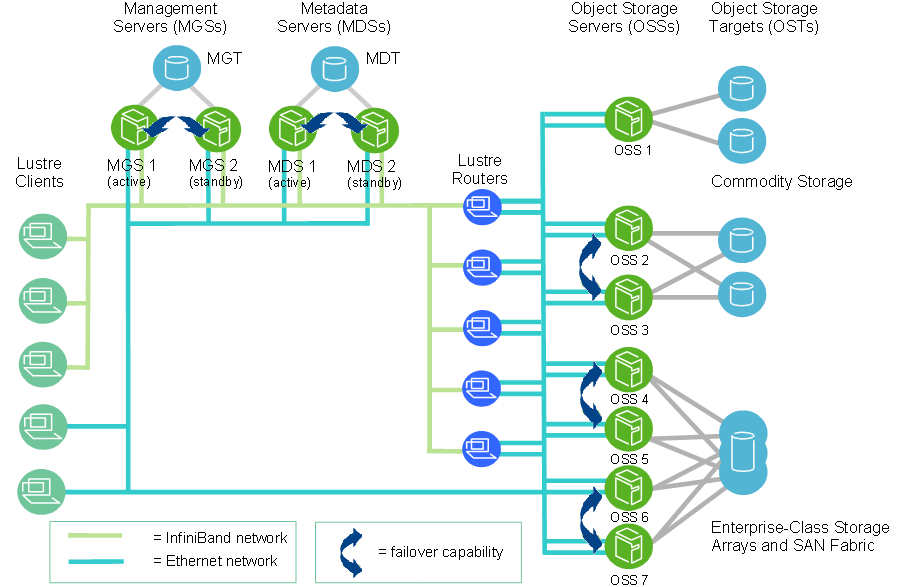
**Архитектура**

Файловая система Lustre содержит следующие основные функциональные модули:

1. **Management Server (MGS)** – **сервер управления** – содержит информацию о файловых системах в кластере, которой он делится с другими компонентами (в т.ч. и с клиентами). Он может быть расположен вместе с сервером метаданных, но предпочтительно выделять для него собственное дисковое пространство, чтобы управлять им независимо.
2. **Metadata Server (MDS)** **– сервер метаданных** – предоставляет метаданные, хранящиеся на соединенных с ним **целях метаданных (metadata target — MDT).**
3. **Metadata Target (MDT) – цель метаданных** – хранит метаданные о пространстве имен, например имена файлов, каталогов, права доступа, а также карту размещения файлов.

До версии 2.3 включительно было возможно иметь только одну MDT в файловой системе, которая могла обслуживаться несколькими MDS, но во взаимоисключающем режиме. В версии 2.4 появилась возможность хранить метаданные файловой системы на нескольких MDT, благодаря **Distributed Namespace Environment (DNE) – распределенному пространству имен**: метаданные корня файловой системы хранились на главной (primary) MDT, а метаданные подпапок можно было хранить на нескольких MDT. В версии 2.8 появилась возможность разделять файлы одной директории по разным узлам MDT, такие папки называются **разделенными (striped).**

1. **Object Storage Server (OSS)** – **сервер хранения объектов** – хранит данные файлов из одного или нескольких **целей хранения объектов (object storage targets — OST).** В зависимости от оборудования сервера OSS обычно обслуживает от двух до восьми OST, емкостью до 16TB, а каждая OST управляет одной локальной дисковой файловой системой. Ёмкость файловой системы Lustre определяется суммой ёмкостей, предоставляемых OST.
2. **Object Storage Target (OST)** **– цель хранения объектов** – пользовательские данные хранятся на одном или нескольких объектах, а каждый объект – на OST. Имеется возможность конфигурировать количество объектов, на которые делится файл, в целях повышения производительности.
3. **Клиенты**, обращающиеся и использующие данные. Lustre предоставляет всем клиентам унифицированное пространство имен для всех файлов и данных в файловой системе, используя стандартную семантику POSIX, а также обеспечивает параллельный когерентный доступ по записи и чтению к файлам в файловой системе/ Таким образом, файловая система Luster для клиента является виртуальной файловой системой Linux, а всю заботу об операциях чтения/записи берет на себя ПО Luster.

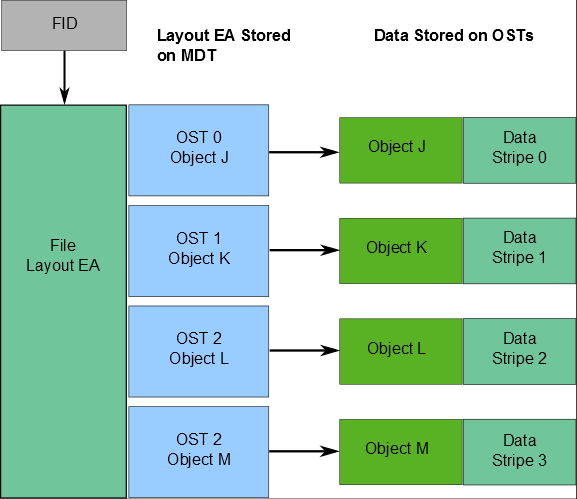


Компоненты MDT, OST, а также клиенты могут находиться на одном и том же узле, но при инсталляции их обычно помещают на отдельные узлы, взаимодействующие по сети. Уровень сети Lustre (Lustre Network — LNET) поддерживает несколько коммуникационных платформ, включая Infiniband, TCP/IP через Ethernet и другие сети, Myrinet, Quadrics, а также проприетарные сетевые технологии. Lustre также использует преимущества удаленного прямого доступа к памяти, если это возможно, для увеличения пропускной способности и снижения доли использования ЦП.

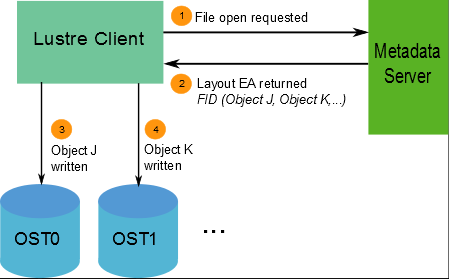
Хранилище, используемое для дублирующих файловых систем MDT и OST, делится на части и может быть организовано через управление логическими томами (разные физические тома представляются как один) и/или RAID (объединение нескольких дисков в один логический элемент), как правило, форматированными в файловой системе ext4. Сервера Lustre OSS и MDS считывают, записывают и изменяют данные в формате, введённым этими файловыми системами.

OST является выделенной файловой системой, экспортирующей интерфейс в байтовые регионы объектов для операций чтения/записи. MDT является выделенной файловой системой для контроля доступа к файлам и сообщений киентам о том, какие объект(ы) входят в структуру файла. В настоящее время MDT и OST используют для хранения данных улучшенную версию ext4, называемую ldiskfs. В 2008 Sun начала проект портирования Lustre в Sun ZFS/DMU для хранения внутренних данных, продолжающийся как проект с открытым кодом.

**Доступ к данным**



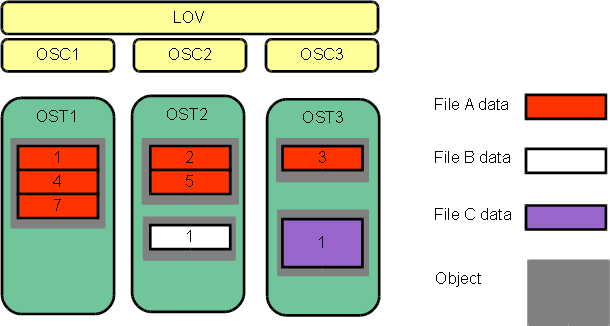
Информация о том, на каких OST данные расположены хранится в атрибуте Имя файла (Layout EA) на узле MDT. Файл идентифицируется FID (128 бит). Если файл является обычным (не папка или ссылка), то этот атрибут указывает на один или несколько объектов, которые хранятся на OST. Таким образом, данные разделяются на части, которые могут храниться на удаленных узлах.



При доступе клиента к файлу поиск имени файла (Layout EA) выполняется в MDS. В результате файл либо создается от имени клиента, либо клиенту возвращается местоположение существующего файла. При операциях чтения или записи клиент интерпретирует местоположение на уровне логического тома объекта, отображающего смещение и размер в один или несколько объектов, каждый из которых расположен на отдельной OST. Затем клиент блокирует диапазон файлов для работы и исполняет одну или несколько операций параллельного чтения или записи непосредственно в OST. При таком подходе устраняются узкие места взаимодействия клиента с OST, так что общая пропускная способность, доступная клиенту для чтения и записи, масштабируется почти линейно с ростом количества OST в файловой системе.

Клиенты не модифицируют объекты файловой системы OST напрямую, делегируя эту задачу OSS. Этот подход обеспечивает масштабируемость для крупномасштабных кластеров и суперкомпьютеров, а также улучшает безопасность и надежность. В противоположность этому блочные разделяемые файловые системы, например Global File System и OCFS должны поддерживать прямой доступ к хранилищу для всех клиентов в файловой системе, увеличивая риск повреждения файловой системы со стороны неправильных клиентов.

**Объекты данных и разделение данных**



В традиционных дисковых файловых системах UNIX, структура данных inode содержит базовую информацию о каждом файле, например о том, где хранятся данные, содержащиеся в файле. Файловая система Lustre также использует inode, но inode на MDT указывает на один или несколько объектов OST, ассоциированных с файлом, а не на блоки данных. Эти объекты реализуются как файлы OST. При открытии файла клиентом операция открытия передает множество указателей объекта и их расположение из MDS клиенту, после чего клиент может непосредственно взаимодействовать с узлом OSS, хранящим объект, что позволяет клиенту выполнять ввод-вывод в файл без последующего взаимодействия с MDS.

Если с MDT inode ассоциирован только один объект OST, этот объект содержит все данные файла Lustre. Если с файлом ассоциировано более одного объекта, данные файла «разделены» среди объектов подобно RAID 0. Разделение файла между несколькими объектами дает существенные преимущества в производительности. При использовании разделения, максимальный размер файла не ограничен размерами одной цели. Ёмкость и совокупная пропускная способность ввода-вывода масштабируется с ростом числа OST, по которым разделен файл. Кроме того, поскольку блокировка каждого объекта для каждой OST управляется независимо, добавление частей (OST) масштабирует возможности блокировки ввода-вывода в файл пропорционально. Каждый файл в файловой системе может иметь различное размещение при разделении, так что ёмкость и производительность можно оптимально настроить для каждого файла.

**Блокировки**

Lustre использует менеджера распределённой блокировки в стиле VMS для защиты целостности данных и метаданных каждого файла. Доступ и модификация файла Lustre полностью когерентна для всех клиентов. Блокировки метаданных управляются MDT, хранящим inode файла с использованием 128-битного идентификатора Lustre File Identifier (FID, состоящего из номера последовательности и идентификатора объекта), используемого как имя ресурса. Блокировки метаданных делятся на несколько частей, защищающих процесс обнаружения файла (владелец и группа файла, разрешения и режим, а также ACL (Access Control List) — список контроля доступа, который определяет, кто или что может получать доступ к конкретному объекту, и какие именно операции разрешено или запрещено этому субъекту проводить над объектом., состояние inode (размер каталога, содержимое каталога, количество ссылок, временные метки), размещение (разделение файла), а также специальные атрибуты. Клиент может получить несколько частей блокировки метаданных для одного inode одним запросом RPC, но в настоящее время предоставляются только блокировки для чтения inode. MDS управляет всеми модификациями inode, чтобы избежать конкуренции за ресурс и сейчас возможен только один узел, получающий блокировки на запись inode.

Блокировки данных файла управляются OST, по которым разделен каждый объект файла, с использованием байтовых блокировок экстентов. Клиенты могут получить перекрывающиеся блокировки чтения экстента для части или всего файла, что позволяет существовать нескольким параллельным читателям для одного файла, а также неперекрывающиеся блокировки записи экстента для областей файла. Это позволяет многим клиентам Lustre получить параллельный доступ к файлу для чтения и записи, избегая проблем при вводе-выводе файла. На практике клиенты Linux управляют своим кэшем данных в единицах страниц, поэтому они будут запрашивать блокировки, всегда являющиеся целым, кратным размеру страницы (4096 байт в большинстве клиентов). При запросе клиентом блокировки экстента OST может предоставить блокировку на больший экстент, чтобы сократить количество запросов блокировки, выполняемых клиентом. Реальный размер предоставленной блокировки зависит от нескольких факторов, включая количество уже выполненных блокировок, наличие конфликтующих блокировок на запись, и числа запросов на блокировку. Установленная блокировка никогда не будет меньше требуемого экстента. Блокировки экстента OST используют Lustre FID в качестве имени ресурса для блокировки. Поскольку число серверов блокировки экстента растет с увеличением числа OST в файловой системе, это также увеличивает совокупную производительность блокировки файловой системы и одного файла, если он разделен по нескольким OST.

**Сетевые возможности**

В кластере с файловой системой Lustre, системная сеть, соединяющая серверы и клиентов, реализуется посредством Lustre Networking (LNET), предоставляющей коммуникационную инфраструктуру, требуемую файловой системе Lustre. Дисковое хранилище соединяется с MDS и OSS традиционными технологиями сетей хранения данных.

LNET поддерживает много общеизвестных типов сетей, например InfiniBand и IP, позволяя осуществить одновременный доступ для сетей разных типов с маршрутизацией между ними. Удаленный прямой доступ к памяти разрешен при поддержке его нижележащей сетью, такой как Quadrics Elan, Myrinet и InfiniBand. Возможности высокой доступности и восстановления позволяют прозрачное восстановление при использовании отказоустойчивых серверов.

LNET предоставляет пропускную способность через сети Ethernet (GigE) свыше 100 MB/s, пропускную способность до 3 GB/s через каналы InfiniBand QDR, пропускную способность свыше 1 GB/s через интерфейсы 10GigE.

**Высокая доступность**

Высокая доступность файловой системы Lustre включает надежный механизм отказоустойчивости и восстановления, обеспечивающий прозрачную перезагрузку серверов при неисправности. Взаимодействие версий между младшими версиями программного обеспечения Lustre позволяет обновить сервер, выключив его (или переключив на резервный сервер), выполнить обновление и перезапустить сервер, причем все активные задания продолжают работать, обнаруживая задержку только при переводе хранилища на резервный сервер.

Lustre MDS конфигурируются как активная/пассивная пара, а OSS обычно развертываются в конфигурации активный/активный, обеспечивающей надежность без существенной перегрузки. Часто резервный MDS является активным MDS для другой файловой системы Lustre, поэтому в кластере нет простаивающих узлов.