

Оптимизация потоков данных в центре обработки геномных данных при помощи распределенной файловой системы Lustre¹

Р.Н. Арифуров, Н.В. Попова, С.А. Науменко

Аннотация. Описывается архитектура центра обработки данных высокопроизводительного секвенирования. Для повышения производительности подсистемы ввода-вывода предлагается использование специализированных хранилищ данных с аппаратными RAID-контроллерами, распределение нагрузки по нескольким серверам данных с помощью распределенной файловой системы Lustre, выделение специализированных томов под ресурсоемкие задачи при помощи инфраструктуры Fiber Channel и кластерной файловой системы OCFS2.

Ключевые слова: центр обработки данных, big data, геномика, оптимизация, распределенная файловая система.

Введение

Проект секвенирования (прочтения, расшифровки) генома человека длился с 1990 по 2003 год [1, 2]. В нем участвовали десятки лабораторий по всему миру: последовательность генома была разделена между лабораториями, каждый участок, в свою очередь, был поделен на небольшие последовательности, которые секвенировались методом Сэнгера при помощи автоматизированных секвенаторов. Затраты на проведение этого проекта составили около трех миллиардов долларов США. В результате был получен качественный референсный геном человека длиной 3.5 млрд. нуклеотидов, разработаны методы сбора и анализа данных.

Дробление ДНК при секвенировании неизбежно: будучи плотно упакованной длинной молекулой, при манипуляциях она испытывает разрывы. Поэтому создание секвенатора, который бы прочитывал целые хромосомы, проблематично.

Однако эта естественная особенность процесса секвенирования привела к созданию технологий шотган-секвенирования (shotgun sequencing) и высокопроизводительного секвенирования (high throughput sequencing, «next generation sequencing», NGS).

Основная идея обоих методов заключается в том, что исходная геномная последовательность амплифицируется (многократно копируется), а затем фрагментируется на короткие перекрывающиеся подпоследовательности длиной 100-1000 нуклеотидов. После прочтения фрагментов исходная последовательность восстанавливается при помощи специальных алгоритмов сборки генома. Разработка таких алгоритмов и реализация соответствующих программных систем развилась в отдельную область биоинформатики [3].

Современные высокопроизводительные секвенаторы, например Illumina HiSeq 2000, позволяют получить за 2 недели около 600 млрд. нуклеотидов в коротких чтениях, что составля-

¹ Проект выполнен при поддержке РФФИ (грант №12-07-31261) и министерства образования и науки РФ (грант № 11.G34.31.0008).

ет около 17 геномов человека с покрытием 10х. Стоимость реагентов на такой запуск составляет около 25 тыс. долларов США [4].

Новый дешевый способ получения геномных данных открыл новые возможности в биологии и медицине. Он позволяет получать информацию о геномных последовательностях культурных и диких растений и животных, данные об изменчивости внутри популяций, данные о соматических мутациях в опухолевых тканях, данные о метилировании ДНК, структуре хроматина, взаимодействиях ДНК-белок, данные об экспрессии генов [5].

Высокопроизводительное секвенирование стало одной из ключевых технологий современной биомедицинской науки. Крупнейшие американские и европейские университеты имеют в своем составе центры секвенирования. В Китае организован Пекинский институт геномики [6], который оснащен 128 секвенаторами Illumina Hiseq 2000 и который насчитывал в 2011 году 4000 сотрудников. Крупнейшими геномными центрами являются также Wellcome Trust Sanger Institute [7] в Великобритании и Broad Institute of MIT and Harvard [8] в США.

В настоящее время в России высокопроизводительные секвенаторы используются в лаборатории эволюционной геномики факультета биоинженерии и биоинформатики МГУ [9], в центральном НИИ эпидемиологии Роспотребнадзора [10], в лаборатории эволюционной геномики Института общей генетики РАН [11], в лаборатории геномики Отдела генетики и селекции Центра защиты леса Красноярского края [12] и многих других лабораториях.

Выполнение научных проектов с участием высокопроизводительного секвенатора требует специализированного аппаратного и программного компьютерного обеспечения. В процессе работы секвенатор генерирует до 5ТВ изображений, которые затем распознаются программным обеспечением секвенатора и далее хранятся в виде текстовых файлов, содержащих короткие строки — чтения. Суммарный объем этих файлов измеряется в десятках — сотнях гигабайт, в зависимости от объема проекта. Проекты с наибольшим объемом данных — это медицинские проекты, требующие множества повторностей и контролей, проекты секвениро-

вания больших полных геномов (геномы длиной более 1 млрд. нуклеотидов), проекты, связанные с популяционной геномикой, когда секвенируются геномы множества особей из популяции, проекты изучения дифференциальной экспрессии генов.

Область создания и использования суперкомпьютеров в России развита хорошо. Достаточно упомянуть суперкомпьютеры «Ломоносов» [13], «Чебышёв» [14], установленные в НИВЦ МГУ [15], компьютеры суперкомпьютерного центра РАН [16]. Существует компетентное экспертное сообщество [17], ежегодные конференции и журнал «Суперкомпьютеры» [18].

Традиционная архитектура кластерного суперкомпьютера имеет вид, представленный на Рис. 1. Ее особенностями являются наличие большого количества вычислительных ядер, скомпонованных в серверы-лезвия и графические ускорители, быстрая сеть между вычислительными узлами и сравнительно небольшое хранилище данных. Такой суперкомпьютер обычно предназначен для решения задач физики высоких энергий, вычислительной гидродинамики, квантовой химии, а также в тех случаях, когда компьютер используется многими пользователями, каждый из которых решает задачу, требующую интенсивных вычислений.

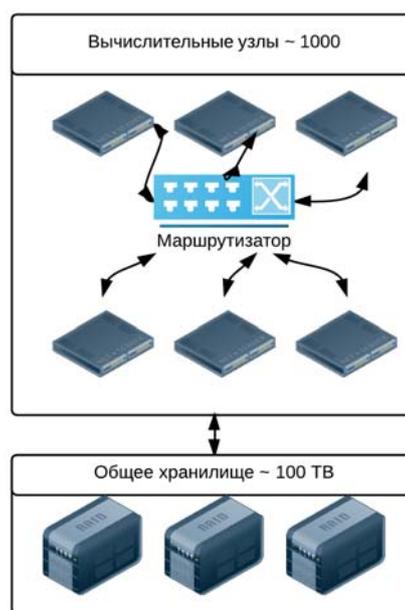


Рис. 1. Архитектура кластерной вычислительной системы

К сожалению, на компьютерах такой архитектуры сложно решать задачи подготовки и анализа геномных данных, для которых требуется небольшое количество вычислительных ядер (порядка сотен на группу исследователей), однако требуются файловые хранилища большого объема и гарантированные полосы пропускания ввода-вывода по сети для вычислительных узлов. Подобные вычислительные комплексы используются в вычислительной геофизике для обработки данных с датчиков, в сотовых компаниях и называются центрами обработки данных (ЦОД) (Рис. 2). Разработка, эксплуатация и оптимизация таких центров представляет собой новую область суперкомпьютерной отрасли — область так называемых «больших данных», или big data.

1. Центр обработки данных для лаборатории эволюционной геномики

В лаборатории эволюционной геномики факультета биоинженерии и биоинформатики МГУ (ФББ МГУ) создан центр обработки данных и вычислительный кластер [19] (далее ЦОД). Данные в хранилища поступают с двух секвенаторов, обработка данных и исследования ведутся силами нескольких лабораторий по удаленному доступу (Рис. 3).

ЦОД включает 3 системы хранения, каждая объемом по 144 ТБ. Две системы хранения соединены с серверами данных по протоколу SAS (Рис. 4). Ресурсы третьей системы хранения

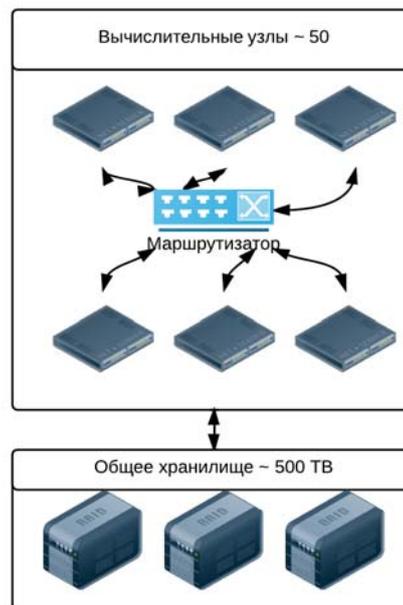


Рис. 2. Архитектура ЦОД

доступны в сети хранения данных по протоколу Fiber Channel на скорости 4 Гб/с.

30 вычислительных узлов представляют собой двухпроцессорные серверы, имеющие по 24 вычислительных ядра и по 48 Гб оперативной памяти. Дополнительный четырехпроцессорный узел с 48 ядрами и 512 Гб оперативной памяти предназначен для сборки геномов.

Два узла управления предназначены для резервирования управления, а также для приема данных с секвенаторов.

Сеть хранения данных включает узел с большой памятью, два узла управления и 10 вычислительных узлов. Сеть обмена данными вклю-

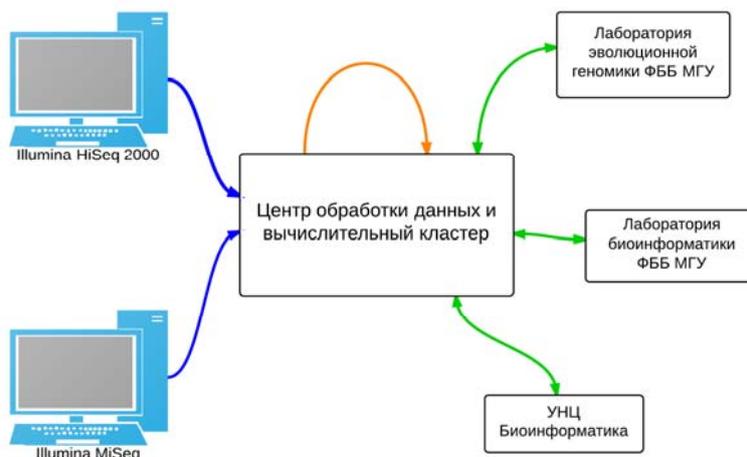


Рис. 3. Потоки данных в центре обработки геномных данных

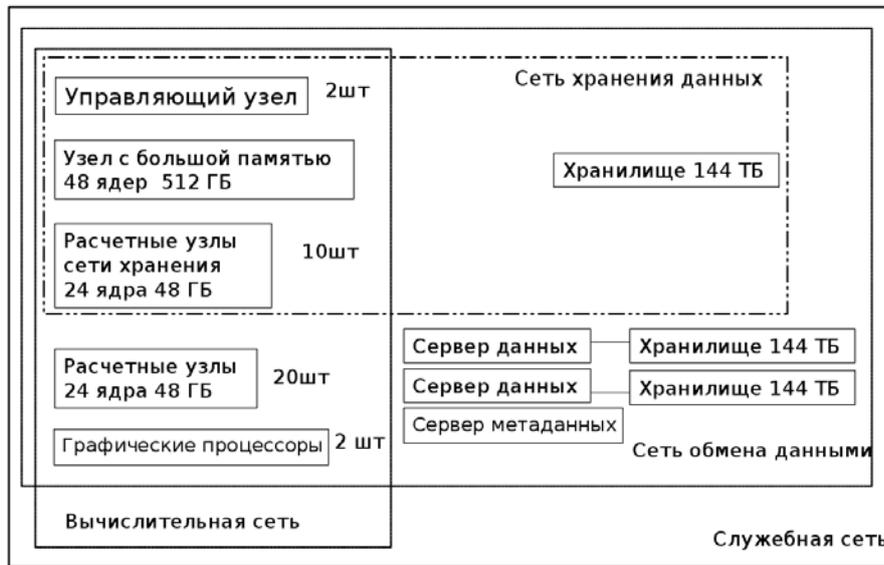


Рис. 4. Основные структурные элементы ЦОД

чает все узлы. Вычислительная сеть предназначена для передачи данных в процессе вычислений. Служебная сеть объединяет все устройства.

Мощное устройство бесперебойного питания с двойным преобразованием фильтрует входной ток и служит защитой в случае кратковременных отключений питания.

В качестве базовой операционной системы в ЦОД используется Scientific Linux [20]. Для управления вычислительными заданиями, управления конфигурациями, мониторинга используются системы torque [21], puppet [22-23], nagios [24]. Предусмотрена рассылка смс-сообщений в критических ситуациях и автоматическое выключение системы при длительном отключении электричества.

2. Способы организации передачи данных в ЦОД

Традиционный в научных лабораториях метод организации передачи данных от файловых серверов к вычислительным серверам — это сетевая файловая система NFS (network file system) [25] (Рис.5).

Достоинство этого метода в простоте и прозрачности его настройки. Недостатком при высокой загрузке (большом количестве запросов на чтение-запись от вычислительных серверов) становится то, что файл-сервер переходит в режим критической загрузки и время отклика становится настолько большим, что становится невозможной работа даже с небольшими объемами

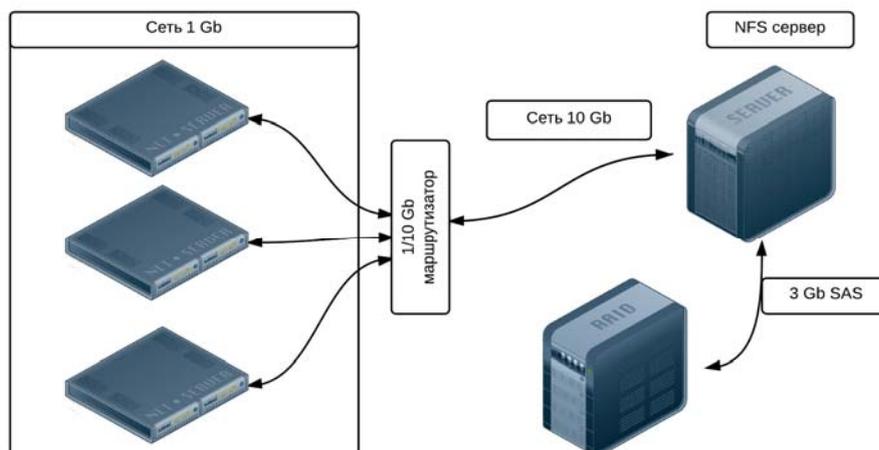


Рис. 5. Архитектура NFS

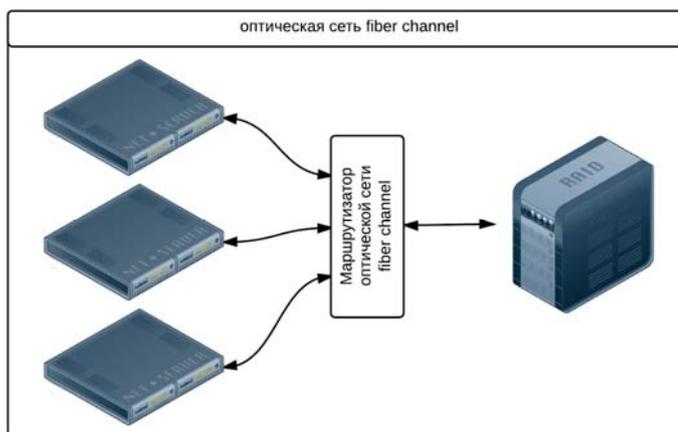


Рис. 6. Архитектура SAN

данных (например, редактирование пользователями своих программ). В случае ЦОД полоса пропускания от системы хранения до файл-сервера составляет 300 МБ/с (по протоколу SAS) или 400 МБ/с (по протоколу Fiber Channel) [26]. Каждый вычислительный сервер может получать данные на скорости 100 МБ/с. Таким образом, при наличии более четырех задач с интенсивным вводом-выводом, запущенных на разных вычислительных узлах, общая производительность системы резко уменьшается.

Альтернативным способом организации передачи данных может быть симметричная кластерная файловая система (Рис. 6). В этом случае каждый вычислительный узел, обладающий адаптером Fiber Channel, получает доступ к файловому хранилищу на скорости 400 МБ/с. Конфликты при доступе к данным разрешаются при помощи блокировок на уровне кластерной файловой системы.

Однако тестирование таких конфигураций показывает, что под нагрузкой при одновременной работе нескольких вычислительных узлов скорость ввода-вывода для одного узла значительно снижается (Табл. 1).

Подобные конфигурации (SAN - storage area network) с кластерными файловыми системами подходят для организации отказоустойчивых решений, а также для процессов с редкой передачей больших объемов данных на высокой скорости (например, таким образом можно было бы организовать обработку геномных данных в большом центре секвенирования, когда присутствует постоянный поток работ: первичная об-

работка данных – сборка геномов – аннотация). Однако для организации вычислительного кластера с высокой загрузкой на подсистему ввода-вывода это решение не подходит.

Распределенная файловая система Lustre позволяет нагрузить несколько серверов данных, и, соответственно, хранилищ, увеличив суммарную полосу пропускания (Рис. 7). Данные от клиентских вычислительных узлов разбиваются на небольшие порции, которые равномерно распределяются по серверам данных и хранилищам, таким образом, максимально полно используется полоса пропускания от серверов данных к хранилищам.

Табл. 1. Скорость чтения-записи для файловых систем NFS, OCFS2, GFS2

dd (write), oflag=direct			
Количество узлов	Средняя скорость записи с одного узла, МБ/с		
	OCFS2	GFS2	NFS4
1	120	59	57
2	78	54	53
3	64	45	51
dd (read), oflag=direct			
Количество узлов	Средняя скорость записи с одного узла, МБ/с		
	OCFS2	GFS2	NFS4
1	274	124	29
2	178	90	29
3	124	73	29

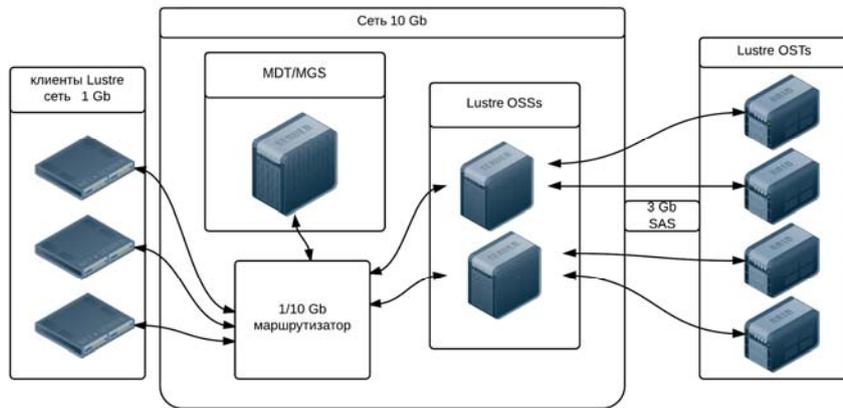


Рис. 7. Архитектура распределенной файловой системы Lustre

3. Архитектура файловой системы Lustre

Lustre [27] — распределенная сетевая файловая система, разработанная компанией Sun/Oracle [28], в данный момент разработка файловой системы передана открытому сообществу [29,30].

Архитектура Lustre включает в себя:

1. управляющий сервер/MGS (Management Server);
2. сервер метаданных/MDS (Metadata Server, зачастую объединенный с управляющим сервером);
3. серверы хранения данных/OSS (Object Storage Server);
4. клиенты Lustre.

В ЦОД развертывание файловой системы осуществляется на следующей аппаратной архитектуре (Рис 7):

- управляющий сервер объединен с сервером метаданных, на нем выделен раздел 100 ГБ (аппаратный RAID 10) для хранения метаданных;
- два сервера хранения данных, по 4 объекта хранения данных на каждом (12,7 ТБ аппаратный RAID 5);
 - клиенты на вычислительных узлах: 32;
 - сеть: серверы 10 Гб/с, клиенты 1 Гб/с.

В нашем случае оценка снизу для суммарной полосы пропускания составляет $300 \text{ МБ/с} * 2 * 2 = 1200 \text{ МБ/с}$ (Рис. 7), что значительно повышает возможности кластера по обработке данных.

4. Установка Lustre

Необходимые для установки пакеты доступны по адресам:

<http://downloads.whamcloud.com/public/lustre/>
<http://downloads.whamcloud.com/public/e2fsprogs/>

Для установки компонентов Lustre на серверы (управляющий, метаданных и хранения) помимо пакетов Lustre необходимо также установить модифицированное ядро. На клиентских вычислительных узлах установка модифицированного ядра не обязательна (в случае установки модифицированного ядра на клиентские вычислительные узлы, на них необходимо устанавливать пакеты, помеченные *).

Пакеты, необходимые для установки, в зависимости от типа указаны в Табл. 2.

Для автоматизации процесса установки в директории с пакетами был создан локальный yum репозиторий:

```
#createrepo /local_repo (необходимо установить пакет createrepo)
```

Затем в директории /etc/yum.repos.d/ был создан файл, содержащий информацию о локальном репозитории:

```
[local_repo]
name=lustre local
baseurl=file:///local_repo
gpgcheck=0
```

После этого для установки на сервере выполняется команда:

```
#yum install lustre lustre-ldiskfs
```

на клиенте:

```
#yum install lustre-client
```

Табл. 2. Пакеты для установки Lustre

Пакет Lustre	Описание	Установить на	
		Сервер	Клиент
RPM пакеты модифицированного ядра			
kernel- <ver>_lustre.<ver>	Для OEL и RHEL	X*	
kernel-ib-<ver>	Пакет Lustre OFED, требуется, если используется сеть Infiniband	X*	X
RPM пакеты модулей Lustre			
lustre-modules-<ver>	Для модифицированного ядра	X*	
lustre-client-modules-<ver>	Для клиентов		X
RPM пакеты утилит Lustre			
lustre-<ver>	Пакет утилит Lustre, включает утилиты пользовательского пространства для настройки и запуска Lustre	X*	
lustre-client-<ver>	Утилиты Lustre для клиентов		X
lustre-ldiskfs-<ver>	Lustre-модифицированный пакет модуля ядра для файловой системы ldiskfs	X	
e2fsprogs-<ver>	Пакет утилит, используемый для обслуживания файловой системы ldiskfs	X	

Все необходимые зависимости устанавливаются автоматически.

Кроме того, на всех серверах, как вычислительных, являющихся клиентами Lustre, так и серверах данных и сервере метаданных, необходимо отключить подсистему безопасности SELinux. Для этого в файле `/etc/sysconfig/selinux` необходимо заменить `SELINUX=enforcing` на `SELINUX=disabled`.

После этого требуется перезагрузить сервер для применения настроек Selinux и загрузки в установленное ядро.

На сервере метаданных необходимо создать и смонтировать файловую систему:

```
#mkfs.lustre --fsname=lustre --mgs --mdt --
index=0 /dev/sdX
#mount -t lustre /dev/sdaX /mnt/mdt/
```

Для автоматического монтирования необходимо добавить запись в `/etc/fstab`:

```
LABEL=lustre-MDT0000 /mnt/mdt
lustre defaults,_netdev 0 0
```

В данном случае монтирование осуществляется по метке (`lustre-MDT0000`), которая созда-

ется при форматировании (метка формируется следующим образом — сначала название файловой системы, указанное в опции `--fsname`, в данном случае `lustre`, затем описание роли, которую исполняет диск — `MDT` или `OST`, затем номер, соответствующий индексу указанному при форматировании в опции `--index`). Монтирование по метке нельзя применять в случае использования `multipathIO`.

Для создания файловых систем на серверах хранения данных необходимо выполнить следующие команды.

```
#mkfs.lustre --fsname=
lustre --mgsnode=192.168.0.104@tcp0 --ost --
index=N /dev/sdX
```

где `192.168.0.104@tcp0` — сетевой адрес и идентификатор интерфейса, с которым работает Lustre на сервере метаданных. По умолчанию используется первый сетевой интерфейс в системе (`eth0`), которому присваивается идентификатор `tcp0`. Если необходимо использовать другой сетевой интерфейс или Infiniband, то нужно создать конфигурационный файл

/etc/modprobe.d/lustre.conf, в котором прописать соответствующие интерфейсы, например,
 options lnet networks=
 tcp0(eth0), tcp1(eth1), o2ib

Монтирование созданных объектов хранения (OST — object storage target)
 #mount -t lustre /dev/sdX /mnt/ostN/

Для автоматического монтирования в файл /etc/fstab для каждого OST необходимо добавить запись:

```
LABEL=lustre-OST000N          /mnt/ostN
lustre defaults,_netdev      0      0
```

В данном случае, как и MDT, OST монтируются с помощью метки.

После создания и монтирования объектов хранения (OST) на серверах хранения необходимо смонтировать файловую систему на клиентских вычислительных узлах:

```
#mount -t lustre 192.168.0.104@tcp0:/lustre
/mnt/lustre
```

Для автоматического монтирования в файл /etc/fstab необходимо добавить запись

```
192.168.0.104@tcp0:/lustre /mnt/lustre
lustre defaults,_netdev 0 0
```

Опция `_netdev` указывает на то, что монтирование должно осуществляться после инициализации сети.

Проверить доступность файловой системы `lustre` можно, выполнив на любом клиентском вычислительном узле команду

```
$ lfs df -h
```

Видно, что суммарный объем файловой системы составляет 101.8Т.

5. Администрирование Lustre

Основной утилитой управления файловой системой `Lustre` является `lctl`. Она позволяет получить доступ к большей части параметров файловой системы. Далее приведен пример применения данной утилиты, а также утилиты `lfs_migrate`, для миграции файлов между `OST`.

Миграция файлов между OST:

```
lfs_migrate <файл или директория>
```

При миграции файл может снова быть записан на исходный `OST`. Чтобы избежать этого, необходимо на `MDS` выполнить команду:

```
#lctl --device <id> deactivate
```

Список устройств и их номера можно получить при помощи команды:

```
#lctl dl
```

Для активации устройства необходимо выполнить:

```
#lctl --device <id> activate
```

Кроме того, важными являются настройки распределения файлов по объектам хранения (`OST`), так называемый страйпинг (`striping`). По умолчанию каждый файл хранится только на одном `OST`, а размер страйпа составляет 1 МВ. Проверить установки страйпинга можно командой:

```
lfs getstripe <файл или директория>
```

UUID		bytes	Used	Available	Use%	Mounted on
lustre-MDT0000_UUID	105.9G	465.8M	98.4G	0%		/mnt/lustre[MDT:0]
lustre-OST0000_UUID	12.7T	420.4M	12.1T	0%		/mnt/lustre[OST:0]
lustre-OST0001_UUID	12.7T	420.4M	12.1T	0%		/mnt/lustre[OST:1]
lustre-OST0002_UUID	12.7T	420.4M	12.1T	0%		/mnt/lustre[OST:2]
lustre-OST0003_UUID	12.7T	420.4M	12.1T	0%		/mnt/lustre[OST:3]
lustre-OST0004_UUID	12.7T	420.4M	12.1T	0%		/mnt/lustre[OST:4]
lustre-OST0005_UUID	12.7T	420.4M	12.1T	0%		/mnt/lustre[OST:5]
lustre-OST0006_UUID	12.7T	420.4M	12.1T	0%		/mnt/lustre[OST:6]
lustre-OST0007_UUID	12.7T	420.4M	12.1T	0%		/mnt/lustre[OST:7]
filesystem summary:	101.8T	3.3G	96.7T	0%		/mnt/lustre

Изменить настройки страйпинга для отдельного файла или для целой директории можно с помощью команды *lfs setstripe*.

Для просмотра статистики для какого-либо OST в режиме реального времени можно использовать утилиту *llobdstat*. Утилита *llshowmount* показывает список узлов, на которых смонтирована файловая система Lustre. Утилита *tune2fs.lustre* изменяет параметры конфигурации дисков Lustre (OST, MDT).

Утилита *lustre_rsync* предназначена для репликации файловой системы Lustre (источник данных) на какую-либо другую файловую систему (приемник данных). Данная утилита получает список файлов и директорий с сервера метаданных, поэтому ей не требуется фактически обходить дерево файловой системы. За счет этого утилита является эффективной по времени выполнения.

Утилита *lfs* предоставляет следующие возможности:

- проверка статусов MDT и OST серверов — *lfs check mdt|osts|servers*;
- получение информации о свободном месте — *lfs df [-i] [-h] [path]*;
- поиск файлов по различным параметрам — *lfs find*;
- управление квотами — *lfs quota, quotacheck, quotachown, quotainv, quotaon, quotaoff, setquota*;
- управление страйпингом — *lfs getstripe, setstripe*;

Для проверки и восстановления файловой системы используется утилита *lfsck*.

6. Ранжирование потребителей каналов ввода-вывода как способ оптимизации загрузки

Существующие планировщики вычислительных задач предназначены для планирования процессорных ресурсов и использования памяти, и не предназначены для планирования каналов доступа к дисковым ресурсам. В целях управления каналами доступа к удаленным сетевым дискам можно тестировать нестандартное использование прокси-серверов, регулирующих объем трафика.

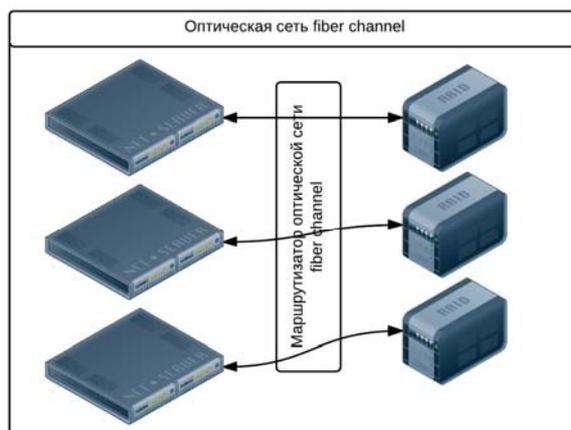


Рис. 8. Выделенные тома для крупных потребителей ресурса

Опыт эксплуатации ЦОД показывает, что постоянный мониторинг занятости каналов, через которые осуществляется ввод-вывод, позволяет выявить те проекты, которые являются наиболее крупными потребителями ресурсов. В геномике таким проектами могут быть анализ больших эукариотических геномов, анализ популяций больших геномов, проекты с сотнями транскриптомов. Объем данных по подобным проектам может составлять несколько терабайт, с рабочими потоками данных несколько сотен мегабайт в секунду [31].

Таким образом, появляются крупные потребители ресурса ввода-вывода, потоки которых сравнимы с суммарным потоком остальных потребителей (средних, порождающих потоки в 10 МБ/с, и малых, с потоками в 1 МБ/с).

Архитектура ЦОД позволяет выделять для крупных потребителей на время проекта отдельные тома, подключая их к выделенным серверам (Рис. 8). Это позволяет гарантировать такому проекту скорость доступа 400 МБ/с с одновременной разгрузкой канала для большинства проектов.

7. Сравнение файловых систем NFS и Lustre

Файловая система Lustre по сравнению с NFS обеспечивает следующие преимущества.

1. Масштабируемость в обе стороны.

Суммарный размер файловой системы и общая полоса пропускания легко увеличивают-

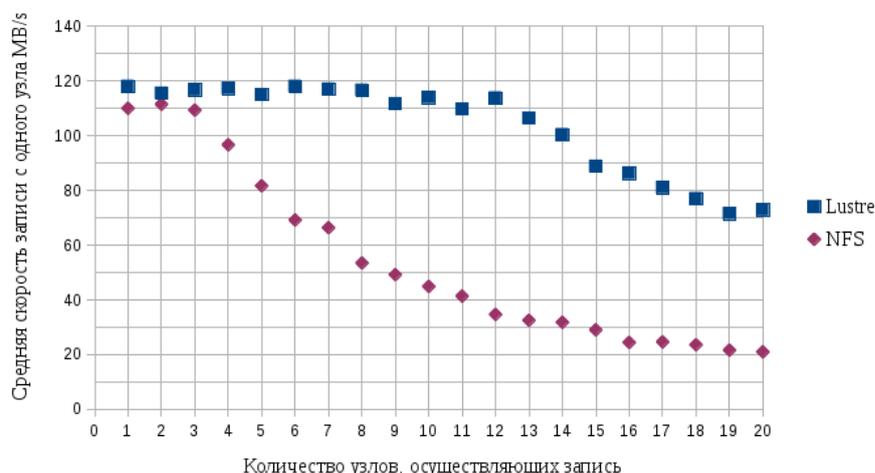


Рис. 9. Сравнение производительности файловых систем Lustre и NFS

ся путем добавления новых серверов данных или добавления OST к уже существующим. Также могут быть добавлены новые клиентские вычислительные узлы.

2. Прозрачная замена жестких дисков в хранилище.

Можно вывести из работы любой из OST, распределив хранящиеся на нем данные между остальными, извлечь данный OST из файловой системы и произвести физическую плановую замену жестких дисков без остановки работы всей файловой системы.

3. Более высокая производительность.

Производительность NFS сервера ограничена пропускной способностью шины от одного сервера до системы хранения. При использовании Lustre суммарная производительность складывается из суммы производительностей серверов данных. На Рис. 9 показана скорость записи с одного вычислительного узла при различной нагрузке (от 0 до 20 одновременно пишущих узлов). Видно, что производительности Lustre и NFS примерно равны только в случае, когда нагрузка составляет 3 узла и менее. При большем количестве узлов производительность Lustre значительно выше.

Заключение

С ростом количества внедрений высокопроизводительных секвенаторов в биологических и медицинских лабораториях необходимо развертывание новых центров обработки данных.

Нами представлена разработка архитектуры и оптимизации потоков данных в центре, обслуживающем 2 секвенатора, около 50 пользователей и до 400 ТБ данных. В качестве основных рекомендаций при проектировании и развертывании новых ЦОД в области геномики представляются использование:

- специализированных систем хранения с дублирующими аппаратными RAID-контроллерами;
- жестких дисков массового производства в модификации RAID Edition объемом 2-3 ТБ;
- инфраструктуры fiber channel;
- мощного ИБП;
- распределенной файловой системы Lustre;
- локальных хранилищ для самых крупных проектов;
- кластерной файловой системы в случае организации конвейера обработки данных;
- систем мониторинга, управления конфигурациями, управления задачами;
- узлов с большой памятью, если в задачи центра входит сборка больших геномов и транскриптомов.

Литература

1. Human genome project information // http://web.ornl.gov/sci/techresources/Human_Genome/index.shtml (дата обращения: 30 сентября 2013 г.).
2. International Human Genome Sequencing Consortium (IHGSC). Finishing the euchromatic sequence of the human genome. // Nature. - 2004. - 431. - 931.-945.

3. Miller J.R., Koren S., Sutton G.. Assembly algorithms for next-generation sequencing data. // *Genomics*. - 2010. - 95. - 315-327.
4. Specifications sheet illumina sequencing // http://res.illumina.com/documents/systems/hiseq/datasheet_hiseq_systems.pdf (дата обращения: 30 сентября 2013 г.)
5. Soon W.W., Hariharan M. and Snyder M.P. High-throughput sequencing for biology and medicine. // *Molecular Systems Biology*. - 2013. - 9. - 640.
6. BGI // <http://www.genomics.cn/en/index> (дата обращения: 30 сентября 2013 г.).
7. Home page – Wellcome Trust Sanger Institute // <http://www.sanger.ac.uk/> (дата обращения: 30 сентября 2013 г.).
8. Broad Institute of MIT and Harvard // <http://www.broadinstitute.org/> (дата обращения: 30 сентября 2013 г.).
9. Лаборатория эволюционной геномики ФББ МГУ // <http://evolgenomics.fbb.msu.ru> (дата обращения: 30 сентября 2013 г.).
10. Центральный научно-исследовательский институт эпидемиологии Роспотребнадзора // www.pcr.ru (дата обращения: 30 сентября 2013 г.).
11. Laboratory of Evolutionary Genomics // <http://rogaevlab.ru> (дата обращения: 30 сентября 2013 г.).
12. Отдел генетики и селекции - Центр защиты леса Красноярского края // <http://www.cz124.ru/struktura/otdel-genetiki-i-selektii.html> (дата обращения: 30 сентября 2013 г.).
13. Суперкомпьютер "Ломоносов" // <http://www.msu.ru/lomonosov/science/computer.html> (дата обращения: 30 сентября 2013 г.).
14. MSU SUPERCOMPUTERS: "CHEBYSHEV" // <http://hpc.msu.ru/?q=node/60> (дата обращения: 30 сентября 2013 г.).
15. Научно-исследовательский вычислительный центр МГУ // <http://www.srcc.msu.ru/news.htm> (дата обращения: 30 сентября 2013 г.).
16. МСЦ РАН – Главная страница // <http://www.jssc.ru> (дата обращения: 30 сентября 2013 г.).
17. Информационно-аналитический центр по параллельным вычислениям // <http://parallel.ru/> (дата обращения: 30 сентября 2013 г.).
18. Журнал "Суперкомпьютеры" // <http://supercomputers.ru/> (дата обращения: 30 сентября 2013 г.).
19. Арифюлов Р.Н., Науменко С.А. Опыт создания центра обработки данных и вычислительного кластера для лаборатории эволюционной геномики. // Информационные технологии и системы (ИТиС'12): сборник трудов конференции. - М.: ИППИ РАН, 2012, с. 324-327. ISBN 978-5-901158-19-7. // <http://itas2012.iitp.ru/pdf/1569601141.pdf> (дата обращения: 08.10.2013).
20. Scientific Linux // <https://www.scientificlinux.org/> (дата обращения: 30.09.2013).
21. Torque Resource Manager // <http://www.adaptivecomputing.com/products/open-source/torque/> (дата обращения: 30.09.2013).
22. Puppet Labs: The Leading IT Automation Software Solution // <http://puppetlabs.com> (дата обращения: 30.09.2013).
23. James Turnbull, Jeffrey McCune. Pro Puppet. — Apress 2011.
24. Nagios. // <http://www.nagios.org/>. (дата обращения: 30.09.2013).
25. Shepler S., Eisler M., Noveck D. Network File System (NFS) Version 4 Minor Version 1 Protocol. IETF RFC 5661. - January 2010.
26. Troppens U., Muller-Friedt W., Wolafka R., Erkens R., Hausteine N. Storage Networks Explained. Basics and Application of Fibre Channel SAN, NAS, iSCSI, InfiniBand and FcoE, Second Edition. - 2009 John Wiley & Sons Ltd.
27. Lustre. // http://wiki.lustre.org/index.php/Main_Page (дата обращения: 30.09.2013).
28. Lustre File System | Sun | Oracle // <http://www.oracle.com/us/products/032293.htm> (дата обращения: 30.09.2013).
29. Whamcloud Lustre 2.4. // <http://wiki.whamcloud.com/display/PUB/Lustre+2.4> (дата обращения: 30.09.2013).
30. Development | Lustre – OpenSFS // <http://lustre.opensfs.org/contribute-code/> (дата обращения: 30.09.2013).
31. Арифюлов Р.Н., Науменко С.А. Опыт эксплуатации центра обработки данных и вычислительного кластера в лаборатории эволюционной геномики. // Информационные технологии и системы (ИТиС'13): сборник трудов конференции. - М.: ИППИ РАН, 2013, с. 248-250. ISBN 978-5-901158-23-4. // <http://itas2013.iitp.ru/pdf/1569759149.pdf> (дата обращения: 08.10.2013).

Арифюлов Ренат Надирович. Младший научный сотрудник Лаборатории эволюционной геномики Факультета биоинженерии и биоинформатики МГУ им. М.В. Ломоносова. Окончил Российский химико-технологический университет им. Д.И. Менделеева в 2012 году. Область научных интересов: информационные технологии, linux, датацентр.
E-mail: arifulovrenat@gmail.com

Попова Нина Владимировна. Аспирантка Факультета биоинженерии и биоинформатики МГУ им. М.В. Ломоносова. Окончила МГУ им. М.В. Ломоносова в 2009 году. Область научных интересов: эволюционная геномика.
E-mail: nina.tolmacheva@gmail.com

Науменко Сергей Анатольевич. Научный сотрудник Института проблем передачи информации им. А.А. Харкевича РАН и Лаборатории эволюционной геномики Факультета биоинженерии и биоинформатики МГУ им. М.В. Ломоносова. Окончил Харьковский национальный университет радиоэлектроники в 2004 году. Автор 7 печатных работ. Область научных интересов: молекулярная эволюция, эволюционная геномика, linux, датацентр. E-mail: nauменко@iitp.ru