Работа на кластере Lxfarm

В.Тихомиров

Версия 2.0, май 2006

Содержание

1 Введение **2**

2 Конфигурация и состав Lxfarm **2**

3 Вход в систему **4**

4 Сервисы Lxfarm **5**

4.1 Web-сервер 5
4.2 Дополнительное дисковое пространство 5
4.3 Резервное копирование файлов 6
4.4 Система электронной почты 7
4.5 Система мониторинга 8
4.6 Файловая система AFS 9

5 Программное обеспечение **10**

6 Выполнение заданий в фоновом режиме **11**

7 Система управления заданиями PBS **13**

7.1 Принцип работы PBS 13
7.2 PBS очереди на Lxfarm 14
7.3 Постановка задания в очередь 15
7.4 Ввод-вывод в системе PBS 16
7.5 Файл задания в PBS 18
7.6 Управление заданиями в PBS 19

8 Работа в системе GRID **19**

9 Разные замечания и советы **20**

1 Введение

Lxfarm МИФИ – это ферма (кластер) компьютеров, выступающая как узел GRID – всемирной распределенной системы вычислительных центров. Кроме того, Lxfarm предоставляет часть своих ресурсов локальным (не-GRID) пользователям – сотрудникам и студентам МИФИ. Компьютеры используются для проведения расчетов и хранения больших объемов данных, главным образом, в области ядерной физики, физики элементарных частиц и космических лучей. Ферма подключена к 1 Гбит/сек каналу связи, соединяющему ее с другими центрами в Москве и за рубежом.

В этом документе описываются сервисы Lxfarm и порядок работы локальных (не-GRID) пользователей на ферме. Вопросы, связанные с работой в системе GRID, рассматриваются очень коротко. Поскольку все компьютеры фермы работают под операционной системой Linux, предполагается, что пользователь достаточно хорошо знаком с этой ОС.

Данный документ периодически обновляется и размещается на сайте Lxfarm http://www.lxfarm.mephi.ru в разделе "Документация". Все замечания и предложения по содержанию можно направлять на имя Владимира Тихомирова по электронной почте по адресу Tikhomirov@mephi.ru.

2 Конфигурация и состав Lxfarm

Lxfarm является гетерогенной системой компьютеров на базе процессоров Intel, работающих под управлением операционной системы Linux. В настоящее время в состав Lxfarm входят 13 компьютеров, которые по их функциональному назначению можно разделить на следующие группы:

- Компьютеры с именами от ui01.lxfarm.mephi.ru до ui03.lxfarm.mephi.ru. На эти компьютеры локальные (не-GRID) пользователи могут входить "извне" по сети. Они предназначены для интерактивной работы, отладки программ и несложного счета. В системе GRID эти компьютеры выполняют роль UI – User Interface.

  ui01 работает на процессоре Celeron с рабочей частотой 733 МГц и имеет 256 Мбайт RAM. ui02 и ui03 – на процессоре Pentium-IV 2.4 ГГц, 512 Мбайт RAM.

- Компьютеры с именами от wn001.lxfarm.mephi.ru до wn003.lxfarm.mephi.ru – более мощные машины на базе процессоров Xeon с частотой 2.8 ГГц и 1 Гбайт RAM (wn001 и wn002 –

двухпроцессорные). Предназначены для проведения расчетов, требующих значительных затрат компьютерного времени. Эти компьютеры недоступны для пользователей в интерактивном режиме. Задания на них направляются через систему PBS (см. ниже, раздел 7) и выполняются в пакетном (batch) режиме. В системе GRID эти компьютеры выполняют роль WN (Worker Nodes).

- Сервисные компьютеры:
  - lxfarm01.mephi.ru – файл-сервер для хранения домашних каталогов локальных (не-GRID) пользователей и общего программного обеспечения. На этом же компьютере работает система авторизации локальных пользователей.
  - lxfarm02.mephi.ru – файл-сервер, используемый для предоставления локальным пользователям дополнительного дискового пространства (каталог /scratch00, см. раздел 4.2).
  - lxfarm07.mephi.ru – сервер, управляющий очередями в системе PBS. В GRID выполняет роль CE (Computing Element).
  - lxfarm08.mephi.ru – файл-сервер, выполняющий роль SE (Storage Element) и MON (Monitoring Element) в GRID-узле.
  - se2.lxfarm.mephi.ru – файл-сервер для хранения данных эксперимента PAMELA.
  - backup.lxfarm.mephi.ru – компьютер для резервного копирования рабочих каталогов локальных пользователей, программного обеспечения общего назначения, а также некоторых системных файлов и каталогов.
  - www.lxfarm.mephi.ru – Web-сервер кластера Lxfarm. Используется также для мониторинга работы компьютеров фермы.

  Все сервисные компьютеры недоступны для входа обычным пользователям.

На всех компьютерах фермы установлена операционная система Scientific Linux CERN версии 3.0.6, основанная на дистрибутиве RedHat Enterprise Linux 3, ядро (kernel) 2.4. Компьютеры подключены к источникам бесперебойного питания и работают в круглосуточном режиме.

Везде в дальнейшем мы будем пользоваться обозначениями “UI компьютер” и “WN компьютер”, даже если речь не идет о работе в системе GRID. Для обычных пользователей UI будет означать компьютер, на который открыт интерактивный доступ, а WN – компьютер, на котором выполняются задания в фоновом режиме под управлением системы PBS.

Количество, имена и конфигурация компьютеров на Lxfarm могут со временем измениться. Актуальная информация доступна через систему мониторинга на сайте Lxfarm (раздел 4.5). На UI можно также воспользоваться командами **top**, **df**, **cat /proc/cpuinfo** и **cat /proc/meminfo** для быстрого получения информации о ресурсах компьютера. Компьютеры WN, как уже отмечалось, недоступны для пользователей и, следовательно, на них невозможно выполнить приведенные выше команды. Некоторую информацию о количестве процессоров и объеме оперативной памяти на WN можно получить с помощью команды **pbsnodes -a**, выполненной на любом UI компьютере. Заметим, что на некоторых компьютерах работают процессоры, поддерживающие технологию Intel Hyper Threading. В этом случае для большинства команд и утилит каждый физический процессор представляется в виде двух виртульных CPU (команда **pbsnodes** является исключением: она показывает реальное число процессоров).

3 Вход в систему

Вход по сети на кластер Lxfarm возможен только на компьютеры, имена которых начинаются на "ui". Вход возможен только по протоколу SSH (вход по telnet закрыт из соображений безопасности). На внешнем компьютере, работающем под управлением операционной системы семейства UNIX/Linux с установленным пакетом ssh, необходимо выполнить, например, команду

`ssh username@ui01.lxfarm.mephi.ru`

где `username` – ваше имя пользователя на кластере Lxfarm.

Если вы планируете входить на Lxfarm с компьютера под управлением операционной системы семейства Windows, то на нем необходимо установить пакет программ, поддерживающих SSH протокол. Один из таких пакетов – PuTTY – можно скачать по адресу:
http://www.chiark.greenend.org.uk/~sgtatham/putty

Корневой домашний каталог пользователя на Lxfarm имеет имя `/home/username`, где `username` – ваше имя пользователя. **Важно:** Физически каталоги пользователей являются частью сетевой файловой системы NFS (см. ниже), что делает содержимое этих каталогов идентичным на всех компьютерах фермы.

Отметим, что по умолчанию домашние каталоги пользователей закрыты для чтения кем-либо, кроме владельца. Если вы хотите, чтобы ваши файлы были доступны для чтения другим пользователям системы, выполните команду

```
chmod a+xr $HOME
```

С любого из UI компьютеров, куда вошел пользователь, он с помощью команды ssh может перейти на любой другой UI компьютер кластера. Отсюда же можно направлять задания на WN (см. раздел 7).

Сервис NIS (Network Information Service) обеспечивает единую систему авторизации и идентификации пользователей на всех компьютерах кластера Lxfarm. Посредством сетевой файловой системы NFS (Network File System) осуществляется "экспорт" домашних каталогов пользователей, а также каталогов общего программного обеспечения с сервера lxfarm01.mephi.ru на все UI и WN компьютеры. Таким образом, сервисы NIS и NFS обеспечивают для пользователей единую рабочую среду: на любом из компьютеров, куда пользователю открыт доступ или куда он направляет свои задания, он имеет одно и то же имя, пароль, рабочую оболочку (SHELL), рабочий каталог и программное обеспечение.

Перенос файлов между компьютерами фермы Lxfarm и любыми другими следует проводить с помощью команд **scp** и **sftp**, работающих по протоколу SSH. Вход на компьютеры фермы по протоколу ftp закрыт из соображений безопасности.

Важно: Если вы хотите сменить свой пароль на всех компьютерах фермы, то для этого необходимо выполнить команду **yppasswd** на любом из UI. Не используйте обычную Linux команду **passwd**, поскольку она меняет пароль только на том компьютере, где была выполнена. Аналогично, для смены рабочей оболочки (SHELL), используемой по умолчанию, вместо стандартной Linux команды **chsh** следует применять команду **ypchsh**.

4 Сервисы Lxfarm

4.1 Web-сервер

Web-сервер предоставляет информацию о кластере Lxfarm, содержит различную документацию и ссылки на Интернет ресурсы. Адрес главной страницы веб-сайта Lxfarm: http://www.lxfarm.mephi.ru.

4.2 Дополнительное дисковое пространство

В настоящий момент отсутствуют какие-либо ограничения на размер дискового пространства, занимаемого файлами пользователя. Однако необходимо понимать, что рано или поздно такой момент может наступить. Поэтому рекомендуется держать большие и редко используемые файлы не в своем домашнем каталоге `/home/username`, а в специальном разделе `/scratch00`. Этот раздел физически расположен на сервере lxfarm02 и экспортируются с него посредством сетевой файловой системы NFS на все UI и WN компьютеры.

В разделе `/scratch00` существуют каталоги `/scratch00/username`, где `username` – имя пользователя на Lxfarm. Пользователям дано право записи в этот каталог. Таким образом, вы можете переместить туда большие и редко используемые файлы из своего текущего каталога:

```
mv filename /scratch00/username/
```

Если данный файл необходим для чтения какой-либо программе, запущенной, например, из раздела `/home/username/programs`, то можно не копируя файл из `/scratch00/username`, просто создать символическую ссылку:

```
cd /home/username/programs
ln -s /scratch00/username/filename
```

Аналогичный прием с символической ссылкой можно применить и в случае, если ваша программа при работе создает очень большой выходной файл: направив ссылку на раздел `/scratch00/username`, вы тем самым заставите программу делать физическую запись именно в этот раздел, а не в каталог `/home/...`, откуда программа запущена.

4.3 Резервное копирование файлов

Ежедневно на кластере Lxfarm осуществляется резервное копирование файлов. Полностью копируются все файлы из рабочих каталогов пользователей (раздел `/home`), раздел `/csoft` с программным обеспечением общего назначения, а также ряд системных файлов и каталогов. Такое резервное копирование, помимо страховки на случай сбоев в системе, в некоторых случаях позволяет пользователям восстанавливать свои случайно удаленные файлы.

Резервные копии рабочих каталогов пользователей физически расположены на сервере backup.lxfarm.mephi.ru и экспортируются с него посредством сетевой файловой системы NFS на все UI компьютеры, т.е. на те компьютеры, куда пользователям разрешен интерактивный вход. Здесь сохраненные копии пользовательских файлов доступны для чтения в каталоге `/ubackup/username`, где `username` – имя пользователя.

Система резервного копирования устроена следующим образом. Каждые сутки около 5 часов ночи сохраняются все вновь созданные в течение последних 24-х часов файлы. Если какой-либо файл за это время был модифицирован, то в качестве копии хранится только последний вариант файла. Если пользователь удалил в разделе `/home` какие-то свои файлы, то резервная копия этих файлов в разделе `/ubackup` будет храниться до конца календарного месяца. В начале каждого месяца осуществляется “зеркалирование”, т.е. точное копирование существующих в данный момент в `/home` файлов в раздел `/ubackup`. Таким образом, в этот момент удаленные пользователем из `/home` файлы также перестают существовать и в системе резервного копирования. Это делается, чтобы избежать неограниченного роста размера раздела `/ubackup`.

Важно: Отметим, что для файлов из раздела `/scratch00` резервное копирование не проводится.

4.4 Система электронной почты

Электронная почта на Lxfarm используется главным образом различными сервисами для посылки сообщений о работе системы администратору фермы. Однако сообщения могут направляться и обычным пользователям. Например, сервисы, выполняющие задания в фоновом режиме, такие, как команда **at** (раздел 6) или система PBS (раздел 7), по окончании своей работы направляют пользователям сообщения, содержащие стандартный поток вывода и/или ошибок, т.е. тот текст, который в случае интерактивного выполнения заданий обычно выводится на экран. В принципе, находясь на одном из компьютеров UI, пользователь может просматривать эти сообщения, например, с помощью программы **pine**. Однако необходимо учитывать следующее.

Почтовый ящик для входящих писем (а это – обычный файл) расположен не в каталоге пользователя, а в некотором системном каталоге. Таким образом, это физически разные файлы на разных компьютерах фермы. Сервис **at** всегда выполняет задание на том компьютере, на котором эта команда была запущена. Поэтому, чтобы просмотреть, скажем, выходной поток задания, выполненного командой **at** на ui01, пользователь должен войти на ui01 и открыть там свой почтовый ящик. Если задание было запущено командой **at** на ui02, то необходимо войти и посмотреть почту на ui02 и т.д. Это не очень удобно, тем более, что пользователь может и не помнить, на каком именно компьютере он запускал задание. В этом случае ему придется просматривать свои почтовые ящики на всех UI компьютерах.

Решить эту проблему можно, организовав перенаправление электронной почты с компьютеров Lxfarm на какой-либо другой адрес пользователя. Для этого в корневом каталоге пользователя необходимо создать файл с именем `.forward`. Файл может выглядеть следующим образом:

```
username
vasya@mail.ru
```

Здесь `username` – имя пользователя на ферме Lxfarm. Эта строка необходима **только** если вы хотите, помимо пересылки писем, также сохранять их и на локальном UI или WN компьютере. Вторая строка указывает на адрес `vasya@mail.ru`, по которому следует перенаправлять письма. Поскольку файл `.forward` расположен в корневом каталоге пользователя, а этот каталог через систему NFS экспортируется на все UI и WN компьютеры фермы, то перенаправление писем на адрес `vasya@mail.ru` также будет производиться со всех UI и WN компьютеров.

В системе PBS ситуация сложнее: здесь сообщение должно пройти с одного компьютера, где выполнялось задание (WN) на другой, с которого это задание было направлено (UI), а это невозможно из-за ограничений, накладываемых организацией почты в МИФИ. Поэтому при работе в PBS следует избегать пересылки выходных потоков заданий по e-mail, а направлять их в дисковый файл пользователя. В соответствующем разделе показано, как это сделать.

Важно: Пожалуйста, настройте свою почту на Lxfarm указанным выше способом с использованием файла `.forward`. И не забывайте в PBS заданиях перенаправлять выходные потоки на диск. В противном случае вы рискуете тем, что не увидите сообщений, которые посылаются вам сервисами типа **at** или PBS.

4.5 Система мониторинга

Система мониторинга фермы дает информацию о загрузке компьютеров, размерах занятого и свободного дискового пространства, сетевом трафике и т.п. Система доступна на сайте http://www.lxfarm.mephi.ru в разделе "Монитор системы".

Возможность быстрой оценки текущего состояния фермы предоставляет команда **gstat -a**, которую можно выполнить на любом компьютере. Эта команда показывает текущее состояние и загрузку всех компьютеров фермы, например:

```
CLUSTER INFORMATION
       Name: Lxfarm
      Hosts: 13
Gexec Hosts: 0
 Dead Hosts: 0
  Localtime: Sun May 7 12:16:31 2006

CLUSTER HOSTS
Hostname                     LOAD                       CPU              Gexec
 CPUs (Procs/Total) [     1,     5, 15min] [  User,  Nice, System, Idle]

www.lxfarm.mephi.ru
    1 (    0/   54) [  0.00,  0.00,  0.00] [   0.1,   0.0,   0.0,  99.9] OFF
lxfarm01.mephi.ru
```

```
    2 (    1/  107) [  1.00,  1.00,  1.00] [   0.1,  49.9,   1.1,  48.5] OFF
ui01.lxfarm.mephi.ru
    1 (    2/   57) [  1.00,  1.00,  1.00] [   0.0, 100.1,   0.0,  -0.1] OFF
ui02.lxfarm.mephi.ru
    1 (    2/   65) [  1.00,  1.00,  1.00] [   0.1, 100.0,   0.0,   0.0] OFF
wn002.lxfarm.mephi.ru
    2 (    2/   76) [  2.00,  2.00,  2.00] [   0.0,  50.0,   0.2,  48.9] OFF
wn001.lxfarm.mephi.ru
    2 (    2/   62) [  2.01,  2.02,  1.98] [  50.0,   0.0,   0.1,  49.8] OFF
```

Здесь необходимо обратить внимание на три столбца LOAD: в них показана загрузка процессоров. Соответствующие числа примерно равны количеству задач, выполняемых на данном компьютере, усредненных за время 1, 5 и 15 минут. Пользователь может учесть эти данные для того, чтобы перейти или направить задание на менее загруженный компьютер (разумеется, при этом необходимо учитывать относительную мощность компьютеров).

4.6 Файловая система AFS

AFS представляет собой глобальную сетевую файловую систему. Файлы, расположенные на AFS, доступны с любого компьютера в мире, где установлена программа AFS-клиент. Полный путь ко всем файлам в системе AFS начинается как `/afs/cell/`, где `cell` – имя так называемой ячейки AFS, поддержка которой осуществляется соответствующим AFS-сервером. Многие научные центры и университеты имеют собственные ячейки AFS.

Каталоги и файлы в AFS имеют собственную систему прав доступа, дополнительную к обычным UNIX правам. Как правило, бóльшая часть файлов на AFS относится к программному обеспечению различных экспериментов и проектов и доступна для чтения. Кроме этого, существуют личные каталоги пользователей, авторизованных в данной ячейке AFS. Доступ к таким каталогам защищен паролем.

На компьютерах UI установлена клиентская часть системы AFS. Это означает, что с этих компьютеров можно получить доступ к AFS файлам и каталогам соответственно имеющимся у пользователей привилегиям. Например, в `/afs/cern.ch/project` содержатся каталоги, относящиеся к множеству различных проектов, поддерживаемых ЦЕРН. В каталоге `/afs/cern.ch/sw` можно найти дистрибутивы и откомпилированные программы таких пакетов, как `ROOT` и `GEANT4`. В каталоге `/afs/cern.ch/atlas` содержатся файлы, относящиеся к эксперименту ATLAS. Все эти каталоги, а также множество других, причем не только из ячейки `cern.ch`, доступны пользователям Lxfarm для чтения.

Если вы авторизованы на одном из AFS-серверов, то можете получить полный доступ, в том числе и на запись, к своим каталогам на соответствующей AFS ячейке. Для этого необходимо выполнить команду
`klog -cell afs_cell -principal username`
где `afs_cell` – имя соответствующей AFS ячейки, а `username` – имя пользователя на AFS-сервере. Если `username` совпадает с вашим именем на Lxfarm, то этот параметр можно опустить. Также можно опустить параметр afs_cell в случае, если это – `cern.ch`.

Следует помнить, что сгенерированный командой **klog** ключ доступа к AFS ячейке действует ограниченное (по умолчанию – 24 часа) время, после чего команду необходимо повторить. Кроме того учтите, что этот ключ хранится не в каталоге пользователя, а в системном каталоге того компьютера, где была выполнена команда **klog**. Поэтому при переходе на другой UI компьютер команду **klog** необходимо выполнить заново.

5 Программное обеспечение

На всех компьютерах фермы установлены компиляторы с языков программирования FORTRAN (**g77**), C (**gcc**) и C++ (**g++**). Конкретную версию компилятора можно узнать, выполнив соответствующую команду с опцией `--version`.

Бóльшая часть программного обеспечение общего назначения установлена в каталоге `/csoft`, который экспортируется системой NFS с сервера lxfarm01 на все другие компьютеры кластера. В настоящий момент установлены следующие пакеты программ.

- В каталоге `/cern` установлены фортрановские библиотеки и программы пакета CERNLIB. Заметим, что одновременно может быть установлено несколько версий библиотеки. В системных файлах автозагрузки для всех пользователей Lxfarm задана переменная окружения (environment) `CERN_LEVEL=pro`, а в переменную `PATH` добавлен путь `/cern/pro/bin`. Каталог `pro` является символической ссылкой на одну из версий CERNLIB, которая, таким образом, используется по умолчанию. Если вы хотите использовать другую из установленных версий CERNLIB, то необходимо соответственно изменить переменные `CERN_LEVEL`, `CERN_ROOT` и `PATH`.

- Пакет CLHEP, также разработанный в ЦЕРН, содержит библиотеки программ для базовых функций и операций, написанных на C++. Этот пакет используется, в частности, в GEANT4.

- Многофункциональный пакет ROOT различных версий установлен в каталоге `/csoft/root`.
- В каталоге `/csoft/geant4` расположен пакет GEANT4. В качестве программ визуализации для GEANT4 можно использовать пакет DAWN, который установлен в `/csoft/dawn`. GEANT4 откомпилирован с созданием как статических, так и загружаемых библиотек.
- В каталоге `/csoft/geant4_vmc` можно найти пакет Geant4 VMC – виртуальный Монте-Карло интерфейс между ROOT и GEANT4.
- Две программы для моделирования ядро-ядерных взаимодействий высоких энергий: VENUS и UrQMD версии установлены в каталогах `/csoft/venus` `/csoft/urqmd` соответственно.
- Пакет atlsim установлен в `/csoft/atlsim`.
- В каталоге `/csoft/alice` установлено несколько версий пакета ALIROOT эксперимента ALICE.

В каталоге `/csoft/distrib` можно найти дистрибутивы некоторых из упомянутых программ.

Программы обновляются по мере появления новых версий. В раздел `/csoft` по просьбе пользователей могут быть установлены и другие программы, особенно большие по объему занимаемого дискового пространства и такие, которыми пользуются сразу несколько человек.

6 Выполнение заданий в фоновом режиме

У пользователей могут встречаться задания, длительность исполнения которых составляет несколько часов или даже дней. Такие задания неудобно выполнять в интерактивном режиме: во-первых, для этого нужно все это время держать открытой интерактивную сессию, во-вторых, в случае даже кратковременного разрыва связи между локальным компьютером пользователя и удаленной машиной, на которой исполняется задание, выполнение этого задания будет прервано. В подобной ситуации гораздо удобнее запускать задание в фоновом режиме.

Для выполнения заданий в фоновом режиме можно воспользоваться командой **at**. Аргументами этой команды являются имя файла задания, где записаны команды, которые необходимо исполнить, а также момент времени, когда задание начнет выполняться. **Важно:** Задание выполняется на том же компьютере, на котором была исполнена команда **at**.

Таким образом, на ферме Lxfarm это – UI компьютеры. Количество заданий, одновременно выполняемых на UI компьютере, как в интерактивном режиме, так и запущенных командой **at**, никак не ограничено, поэтому задания на UI компьютерах могут конкурировать между собой. Об этом необходимо помнить и стараться запускать задания на менее загруженных компьютерах, учитывая также их относительную мощность.

Файл задания (`bscript` в нашем примере) в простейшем случае может состоять из одной строки с обращением к выполняемой программе:
```
program
```
где `program` – имя выполняемой программы с указанием полного пути к ней, если это необходимо. Запустить такой командный файл `bscript` можно следующим образом:
```
at -f bscript time
```
где `time` – желаемое время запуска. Простейшие примеры формата времени `time` таковы: ключевое слово `now` означает, что необходимо запустить командный файл немедленно, а, например, 19:40 – в ближайшие 19 часов 40 минут. Формат задания времени `time` можно посмотреть в справочной странице команды **at**. Напомним, что текущее время в системе выдается командой **date**.

Отметим, что в момент начала исполнения задания текущим каталогом становится тот, в котором находился пользователь в момент выполнения команды **at**. Заметим также, что файл задания не обязательно должен иметь атрибут права доступа `x`.

Поскольку задание выполняется в фоновом режиме, то пользователь должен позаботиться о потоках ввода-вывода, т.е. о том, как передать необходимые программе данные, которые в интерактивном режиме вводятся с клавиатуры, и куда будет направлен поток, который в интерактивном режиме выводится на экран. Допустим, исполняемая программа **scitrd** требует ввода с клавиатуры последовательно трех строк:
```
1 10.
RNDM 346781
10000
```
Тогда файл `bscript` может выглядеть так:
```
(echo ‘‘1 10.’’; echo ‘‘RNDM 346781’’; echo ‘‘10000’’) | scitrd
```
Другой способ: создать файл, скажем, `scitrd.inp`, прописать в нем три строки с входными параметрами, а в файле задания `bscript` написать:
```
scitrd <scitrd.inp
```
Выходной поток и поток ошибок, которые в интерактивном режиме выводятся на экран, команда **at** по окончании работы пересылает пользователю по e-mail в почтовый ящик того компьютера, с которого она была запущена. Пользователь может перенаправить эти потоки в диско-

вый файл, скажем, с именем `scitrd.out`, написав в `bscript`:
`scitrd <scitrd.inp >&scitrd.out`
Необходимо однако понимать, что в данном примере в файл `scitrd.out` будет направлен только выходной поток собственно программы **scitrd**. Если ошибка произошла до или после выполнения программы, скажем, не был найден выполняемый файл `scitrd`, то сообщение о такой ошибке все же будет направлено по e-mail. Посмотрите раздел 4.4, чтобы настроить свою электронную почту на ферме.

С помощью команды **atq** можно узнать о состоянии своих заданий, направленных на исполнение командой **at**. При необходимости можно удалить задание – команда **atrm**.

7 Система управления заданиями PBS

PBS (Portable Batch System) – это система управления заданиями, выполняемых в фоновом (batch) режиме на кластере из нескольких компьютеров. Задачей системы является эффективное распределение заданий по компьютерам с учетом имеющихся ресурсов, а также ресурсов, запрашиваемых пользователем. Пользователь, составив задание и направив его на выполнение, избавлен от необходимости просматривать состояние всех компьютеров на кластере, чтобы определить – какой из них обладает необходимыми ресурсами и свободен в данный момент.

Система PBS является одной из важнейших составляющих GRID узла МИФИ. С ее помощью по WN компьютерам распределяются задания, приходящие на ферму из сети GRID со всего мира. Одновременно PBS на Lxfarm используется для распределения заданий, направляемых на WN локальными (не-GRID) пользователями.

7.1 Принцип работы PBS

Система PBS состоит из сервера, принимающего и распределяющего задания, компьютеров-клиентов, с которых задания могут направляться на сервер, и вычислительных компьютеров, где эти задания выполняются. На Lxfarm роль сервера выполняет lxfarm07, клиентскими машинами являются все UI компьютеры, а выполняются задания на WN компьютерах.

В PBS существует несколько очередей (queue), которые не привязаны к определенному вычислительному компьютеру, а принадлежат всей системе PBS в целом. Каждая очередь имеет имя и обладает набором атрибутов, наиболее важными из которых являются максимальное процессорное время, в течение которого задание может исполняться, а также список пользователей, которые имеют право направлять задания в данную очередь. Пользователь составляет свое задание и направляет его в определенную очередь, указав либо имя очереди, либо необходимое для выполнения задания процессорное время. Пользователь может запросить и другие ресурсы, например, объем оперативной памяти, необходимый для выполнения задания.

Система PBS настроена таким образом, что на каждом вычислительном компьютере может выполняться ограниченное число заданий, равное числу процессоров на данном компьютере. Управляющий сервер направляет задание на тот вычислительный компьютер, который удовлетворяет запрошенным в задании пользователя критериям и в данный момент является свободным. Если все процессоры на всех WN заняты, задание остается ждать в очереди. Когда какой-либо процессор на WN освобождается, на нем начинает выполняться одно из таких ждущих заданий. Какое именно задание и из какой именно очереди – определяется достаточно сложной системой приоритетов. Эта система может меняться со временем, например, в какие-то периоды времени давая более высокий приоритет заданиям, направляемым на ферму одним из ЦЕРНовских экспериментов через систему GRID, а в другие периоды – напротив, заданиям локальных пользователей. При прочих равных условиях приоритет будут иметь те задания, которые требуют для выполнения меньшее компьютерное время и которые раньше встали в очередь.

7.2 PBS очереди на Lxfarm

В настоящий момент в системе PBS на Lxfarm существуют следующие очереди.

- Очереди в системе GRID. Задания в эти очереди могут направляться только авторизованными в системе GRID пользователями, являющимися членами соответствующих виртуальных организаций (VO). В настоящее время существуют три таких очереди с именами `atlas`, `pamela` и `dteam`. Максимальное время выполнения заданий в этих очередях – 48 часов. Локальные пользователи не могут направлять задания в эти очереди.

- Очереди для локальных (не-GRID) пользователей.

  - `veryshort` – очередь для выполнения заданий с запрошенным процессорным временем до 15 минут включительно. Эта очередь предназначена, главным образом, для отладки заданий.

– `short` – очередь с временем свыше 15 и до 90 минут включительно.

– `middle` – очередь с временем свыше 90 минут и до 8 часов включительно.

– `long` – очередь с временем свыше 8 и до 48 часов включительно.

– `verylong` – очередь с временем свыше 48 и до 120 часов включительно.

– `unlimited` – очередь с неограниченным временем счета.

Отметим, что очередь `unlimited` обычно не включена. Чтобы направить задание в эту очередь, необходимо обратиться к системному администратору фермы.

Важно: Если в задании не указано ни имя очереди, ни требумое процессорное время, то оно по умолчанию будет направлено в очередь `middle`.

Имена всех существующих очередей и краткую информацию об их текущем состоянии можно получить с помощью команды **qstat -q**.

7.3 Постановка задания в очередь

Пользователь может направить задание на выполнение в системе PBS с любого из UI компьютеров фермы. Постановка задания в очередь выполняется командой **qsub**. В качестве обязательного аргумента выступает имя файла задания. Кроме того, желательно указать запрашиваемое для выполнения задания процессорное время, либо имя очереди. В противном случае, как уже отмечалось, задание будет направлено в очередь `middle`. Примеры:

`qsub -q long bscript`

направит задание, описанное в файле bscript, в очередь `long`.

`qsub -l cput=9:00:00 bscript`

запросит для выполнения задания, описанного в файле `bscript`, 9 часов 00 минут 00 секунд процессорного времени. Таким образом, данное задание также попадет в очередь `long`. Отметим однако, что в случае, когда процессорное время в явном виде не запрошено, а указано имя очереди, то предполагается, что пользователь запросил максимально возможное для данное очереди время. Поэтому если в приведенных выше примерах оба задания окажутся в состоянии ожидания (т.е. все процессоры будут заняты), то при прочих равных условиях раньше на выполнение уйдет задание из второго примера, поскольку у него запрошенное время меньше, чем у первого.

Файл задания (`bscript` в нашем примере) в простейшем случае может состоять из одной строки с обращением к выполняемой программе:

```
program
```

где `program` – имя выполняемой программы с указанием полного пути к ней, если это необходимо. Отметим, что файл задания не обязательно должен иметь атрибут `x`.

Важно: В начале выполнения файла задания в системе PBS (в отличие от команды **at**) предполагается, что пользователь оказывается в корневом каталоге своей домашней директории, а не в том каталоге, из которого была выполнена команда **qsub**. Если программа **program** находится не в корневом каталоге, а скажем, в каталоге `simu`, то для ее выполнения необходимо, чтобы либо путь к программе входил в переменную `PATH` пользователя, либо этот путь был явным образом задан в файле задания `bscript`:

```
simu/program
```

Другой способ решения проблемы – явным образом перейти в каталог `simu` в файле `bscript`:

```
cd simu
program
```

Последний способ особенно удобен, если в каталоге `simu`, помимо самой исполняемой программы, находятся также файлы ввода и/или вывода.

Более сложные варианты файла задания будут приведены в следующих разделах.

При направлении задания на PBS пользователь может запросить и другие ресурсы, кроме процессорного времени. Запрашиваемые ресурсы указываются опцией `-l` команды **qsub** и перечисляются через запятую. Например, команда

```
qsub -l cput=30:00,mem=500mb bscript
```

запросит для выполнения задания 30 минут процессорного времени и 500 Мбайт оперативной памяти.

Некоторые свойства и текущее состояние компьютеров, на которых выполняются задания под управлением системы PBS, можно посмотреть, набрав команду **pbsnodes -a**.

7.4 Ввод-вывод в системе PBS

При работе в системе PBS важно понимать, каким образом осуществляются операции ввода и вывода.

- **Ввод с клавиатуры.** Поскольку в системе PBS программа работает в фоновом режиме, то пользователю необходимо позаботиться о том, чтобы обеспечить программу теми данными, которые в интерактивном режиме должны были вводиться с клавиатуры. В разделе 6 рассказано, как это можно сделать.

- **Вывод на экран.** По умолчанию система PBS во время выполнения программы записывает потоки вывода и ошибок (то, что при интерактивной работе обычно выводится на экран) в два временных файла на том компьютере, где исполняется программа. По окончании выполнения задания PBS пытается переслать эти файлы на тот компьютер, с которого было запущено задание (была выполнена команда **qsub**). Однако на кластере Lxfarm запрещены те протоколы, которые позволяют копировать файлы с компьютера на компьютер без ввода пароля. Поэтому попытка переслать файл с потоками вывода/ошибок оказывается неудачной, о чем PBS информирует пользователя, посылая ему e-mail. Но e-mail также не может пройти с одного компьютера, где выполнялось задание (WN) на другой (UI), с которого оно было направлено, из-за ограничений, накладываемых организацией сети в МИФИ.

  Чтобы решить проблему, можно перенаправить выходные потоки выполняемой программы в файл на диске, как это показано в разделе 6. Другое (рекомендуемое) решение – применить опцию `-k` в команде **qsub**:
  `qsub -q long -k oe bscript`
  Эта опция указывает на необходимость сохранить (keep) файлы с записью потока вывода (`o`) и ошибок (`e`) на том компьютере, где программа выполнялась, а не пересылать их на компьютер, с которого было направлено задание. Файлы потоков вывода и ошибок по умолчанию будут называться `bscript.oNNN` и `bscript.eNNN` соответственно, где `NNN` – номер задания в PBS системе. Обычно бывает удобно объединить эти файлы в один, что можно сделать с помощью опции `-j` (join):
  `qsub -q long -k oe -j oe bscript`
  По завершении задания этот файл под именем `bscript.oNNN` записывается в домашний каталог пользователя на том компьютере, где выполнялось задание, т.е. на одном из WN. А поскольку домашний каталог пользователя экспортируются посредством сетевой файловой системы NFS на все UI и WN компьютеры, то файл `bscript.oNNN` автоматически оказывается доступен и на любом из UI компьютеров.

- **Чтение и запись с диска и на диск.** Как уже отмечалось, благодаря сетевой файловой системе NFS пользователь имеет один и тот же набор своих каталогов и файлов на всех UI и WN компьютерах. Поэтому, если программе необходимы какие-то входные дисковые файлы, или она выводит что-то на диск, то нет необходимости заботиться о том, чтобы пересылать эти файлы между UI и WN. Это замечание относится и к каталогу `/scratch00`, о котором говорилось в разделе 4.2.

7.5 Файл задания в PBS

Как уже отмечалось, простейший файл задания в системе PBS может состоять из единственной строки, в которой вызывается исполняемая программа. Однако этот файл может быть и более сложным, включая в себя как команды рабочей оболочки пользователя, так и различные опции команды **qsub**. Рассмотрим такой пример командного файла `bscript`.

```
#PBS -k oe -j oe
date
cd $PBS_O_WORKDIR
echo HOST: $HOSTNAME
echo PWD=$PWD
scitrd <scitrd.inp
date
```

В первой строке файла задания можно указывать опции команды **qsub**. В этом случае строка должна начинаться с символов `#PBS`. В данном случае заданы опция `-k`, которая указывает на необходимость оставить файлы с потоками вывода и ошибок на компьютере, где будет выполняться задание, и опция `-j`, объединяющая эти потоки в одни файл. Команды **date** покажут время начала и конца выполнения задания. Специальная переменная окружения `$PBS_O_WORKDIR` указывает на тот каталог, из которого была выполнена команда **qsub**. Таким образом, строка
`cd $PBS_O_WORKDIR`
сделает этот каталог текущим. Затем две команды **echo** выведут информацию о компьютере, на котором выполняется задание и о текущем каталоге. Наконец, строка
`scitrd <scitrd.inp`
запускает программу **scitrd**, которая читает из файла `scitrd.inp` данные потока ввода. После окончания задания потоки вывода и ошибок окажутся записанными в файл `bscript.oNNN` в головном каталоге пользователя.

Опции, не записанные в первой строке файла задания, можно по-прежнему указывать в командной строке:
`qsub -q short bscript`

7.6 Управление заданиями в PBS

Информацию о заданиях, находящихся под управлением системы PBS, можно получить с помощью команды **qstat**. Команда имеет множество опций, наиболее информативными из которых являются **qstat -n** и **qstat -f**. Команда **qselect**, позволяет выводить информацию не обо всех заданиях, а только о тех, которые удовлетворяют заданным в команде критериям отбора.

В случае необходимости удалить задание, как еще стоя́щее в очереди, так и уже выполняющееся, можно командой **qdel NNN**, где `NNN` – номер задания, которое сообщается при направлении задания командой **qsub**, а также фигурирует в выводе команды **qstat**.

С помощью команды **qalter** можно вносить некоторые изменения в параметры задания, которые еще не начали исполняться. Команда **qmove** перенаправляет задание, находящееся в состоянии ожидания, в другую очередь.

Команда **qhold** переводит задание, ожидающее в очереди, в "замороженное" (hold) состояние: задание будет находится в нем, даже если есть свободные процессоры. "Разморозить" такое задание можно командой **qrls**.

Для всех команд системы PBS имеются manual page на UI компьютерах. Поэтому вы можете получить полную информацию о любой команде `command` и ее опциях, набрав в командной строке
`man command`

Описание некоторых команд и советы пользователю по работе в системе PBS можно найти в документе "Portable Batch System. Administrator Guide.", расположенном на сайте Lxfarm http://www.lxfarm.mephi.ru в разделе "Документация".

8 Работа в системе GRID

Ферма Lxfarm функционирует как GRID узел системы LCG (LHC Computing GRID), нацеленной на моделирование и обработку данных экспериментов на коллайдере LHC в Европейском Центре Ядерных Исследований (ЦЕРН). На ферме установлено программное обеспечение LCG версии 2_7_0. Имя фермы в GRID – ru-Moscow-MEPHI-LCG2.

Кроме Worker Nodes (WN) на ферме установлены следующие элементы LCG сайта: CE (Computing Element), SE (Storage Element) и MON (Monitoring Element), совмещенный с SE. Ферма не имеет своего RB (Resource Broker), а пользуется услугами RB GRID узла НИИЯФ МГУ. В настоящей момент вычислительные и дисковые ресурсы фермы в GRID сети предоставлены виртуальным организациям (VO) atlas, pamela и dteam.

Компьютеры фермы с именами, начинающимися на “ui”, сконфигурированы как User Interface в системе LCG. На них пользователь автоматически получает все переменные окружения (environment), необходимые для работы в GRID. Пользователи, имеющие GRID сертификаты, могут на UI компьютерах направлять свои задачи в GRID, пересылать файлы и т.д. Заметим, что при создании с помощью команды **grid-proxy-init** временного прокси-сертификата, он записывается в каталог `/tmp`, который является локальным на каждом UI компьютере. Поэтому при переходе с одного на другой UI компьютер команду **grid-proxy-init** необходимо выполнить заново.

Подробно работа пользователей в системе LCG описана в документе “LCG-2 User Guide”, который можно найти в разделе “Документация” на сайте Lxfarm http://www.lxfarm.mephi.ru.

9 Разные замечания и советы

Пользователь должен следить за объемом, занимаемом на диске его файлами. Для этого можно применять команду **du**. Так,

```
du -s $HOME
```

покажет в килобайтах суммарный размер, занимаемый на диске всеми вашими файлами и каталогами. Команда

```
du -sS
```

выдаст суммарный размер всех файлов в текущем каталоге, не включая сюда вложенные каталоги. А

```
du -S | sort -n
```

выведет на экран список каталогов, находящихся внутри текущего, в порядке возрастания размера занимаемого ими дискового пространства.

Иногда при аварийном завершении программ в текущем каталоге создаются файлы с именем `core.NNN`, где NNN – номер процесса. Этот файл представляет собой “посмертный” слепок оперативной памяти, сброшенный программой на диск. Если вы не эксперт и не собираетесь анализировать этот файл, то удалите его, тем более, что часто эти файлы имеют довольно большой размер.

Иногда после переустановки операционной системы или в результате переконфигурации компьютеров на Lxfarm, пользователь при входе на такой компьютер по протоколу SSH может получать “страшные” сообщения примерно такого содержания:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@       WARNING: POSSIBLE DNS SPOOFING DETECTED!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
The RSA1 host key for lxfarm01 has changed,
and the key for the according IP address 194.67.77.111
is unchanged. This could either mean that
DNS SPOOFING is happening or the IP address for the host
and its host key have changed at the same time.
Offending key for IP in /root/.ssh/known_hosts:23
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA1 host key has just been changed.
The fingerprint for the RSA1 key sent by the remote host is
1b:48:4c:fc:02:ed:98:8e:83:24:99:03:c9:99:90:b1.
Please contact your system administrator.
Add correct host key in /root/.ssh/known_hosts to get rid of this message.
Offending key in /root/.ssh/known_hosts:11
RSA1 host key for lxfarm01 has changed and you have requested strict checking.
Host key verification failed.
```

В некоторых случаях пользователю даже не удастся войти на компьютер. Это может быть связано с тем, что в результате переконфигурации на компьютере был изменен файл с электронным ключом, используемым при работе протокола SSH. В этом случае необходимо аккуратно удалить из файла `.ssh/known_hosts`, расположенного на том компьютере, с которого выполняется вход, все строки, содержащие имя и/или IP адрес удаленного компьютера, на который вы пытаетесь войти. Если данный рецепт не помогает – удалите весь файл `.ssh/known_hosts`.

Все компьютеры на ферме настроены таким образом, что при появлении на соответствующем сервере в ЦЕРН обновленных версий пакетов, установленных на Lxfarm, это обновление автоматически происходит и на компьютерах фермы. Это означает, что версии некоторых системных программ, в частности – компиляторов, могут незначительно меняться со временем. Потенциально (хотя и очень маловероятно) это может привести к проблемам при запуске уже откомпилированных программ. В таком случае откомпилируйте их заново.