Massive Solutions

Кластерный комплекс МГУ «Ломоносов»

Руководство пользователя

05 апреля 2011 г.

Содержание

Введение

Этот документ является руководством по использованию вычислительного кластерного комплекса МГУ «Ломоносов» (T500) под управлением операционной системы (ОС) Clustrx.

Документ предназначен для пользователей с неадминистративным уровнем полномочий, работа которых с кластером заключается в выполнении на нем своих вычислительных задач.

В руководстве описана вся последовательность операций, выполняемых пользователем кластера: от входа в операционную среду Clustrx до получения результатов расчетов. Рассмотрена работа как через графический интерфейс пользователя (GUI), так и через интерфейс командной строки, представляющий собой стандартную Linux-оболочку bash.

Доступ к графическому и командному интерфейсам предоставляется пользователям администратором кластера. Кроме данного руководства, пользователям необходимо следовать указаниям администратора по всем вопросам работы с кластером.

Информация о доступных на кластерном комплексе версиях библиотек, компиляторов и другого программного обеспечения соответствует версии образа вычислительного узла **CNL08**.

Другую информацию по текущей конфигурации кластера, необходимую для успешного выполнения вычислительных задач, можно получить у администратора.

Глава 1. Вход пользователя на кластер

В распоряжение пользователя кластера предоставляется два вида интерфейсов (точек доступа), каждый из которых дает возможность выполнять все операции, связанные с вычислительными задачами:

- стандартный командный интерфейс Linux bash с доступом по протоколу **ssh** через соответствующий клиент;
- графический Web-интерфейс Clustrx с доступом через интернет-браузер.

Возможность работы в любом из интерфейсов зависит от уровня привилегированности вашей группы пользователей, установленного администратором. В зависимости от него те или иные средства и возможности могут оказываться недоступными.

1. Командный интерфейс

Постановка задачи

Пользователь подключается к командному интерфейсу кластера через ssh-клиент, например, пакет OpenSSH для Linux или PuTTY для Windows. Организация доступа пользователей к кластеру возложена на администратора. На текущий момент предполагается вход пользователей по ключам.

Порядок выполнения

1. Пользователю следует сгенерировать пару ключей (публичный и приватный), по которым он будет входить на кластер, и передать публичный ключ администратору кластера, чтобы он использовал его при создании учетной записи пользователя.
2. Войдя на кластер, пользователь попадает в стандартную командную оболочку bash для построчного набора и исполнения команд.
3. После первого входа на кластер пользователь должен сменить автоматически предоставленный ему пароль, чтобы затем иметь возможность входить в графический интерфейс пользователя.

1.1. Генерирование ключей

Постановка задачи

Для доступа к кластеру по ssh с локальных машин пользователей под управлением Linux и Windows необходимо сгенерировать пару ключей, приватный и публичный, после чего передать публичный ключ администратору кластера.

Ключевая пара для входа на кластер должна иметь один из двух форматов:

- OpenSSH RSA (желательно) / DSA;

- RFC (поддерживается).

Требуемая длина ключа составляет 2048 бит.

Порядок выполнения

1. Если на локальной машине (не на кластере) используется пакет OpenSSH в среде Linux/UNIX:

   a. Для генерирования пары ключей с требуемыми свойствами выполните команду:

   ```
   ssh-keygen -t rsa -b 2048
   ```

   b. Когда вас попросят, укажите пароль для ключа.

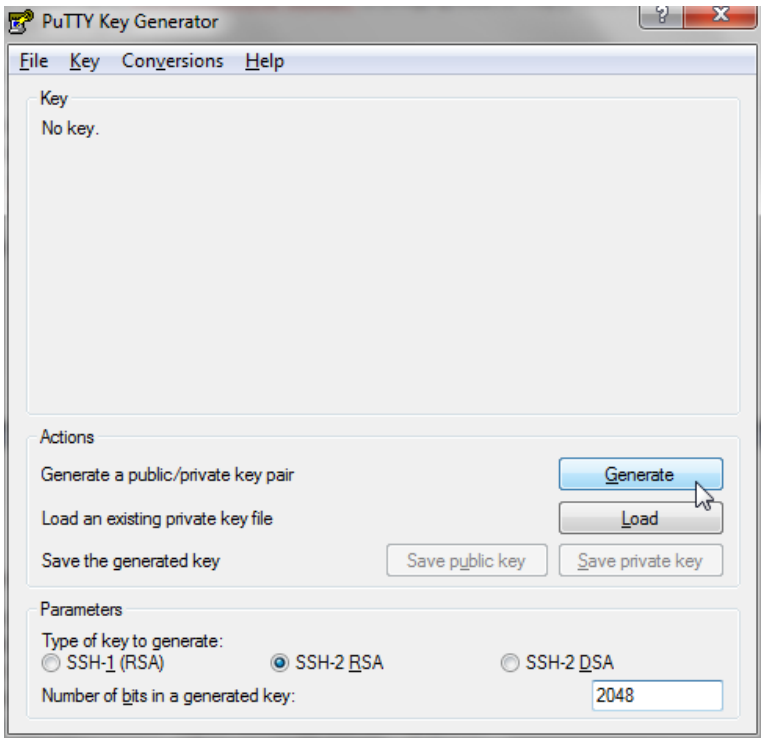
   Не оставляйте ключ беспарольным, это создает проблемы с безопасностью и затрудняет ход работы на кластере.

   Приватный ключ следует оставить в домашнем каталоге пользователя, а публичный — передать администратору.

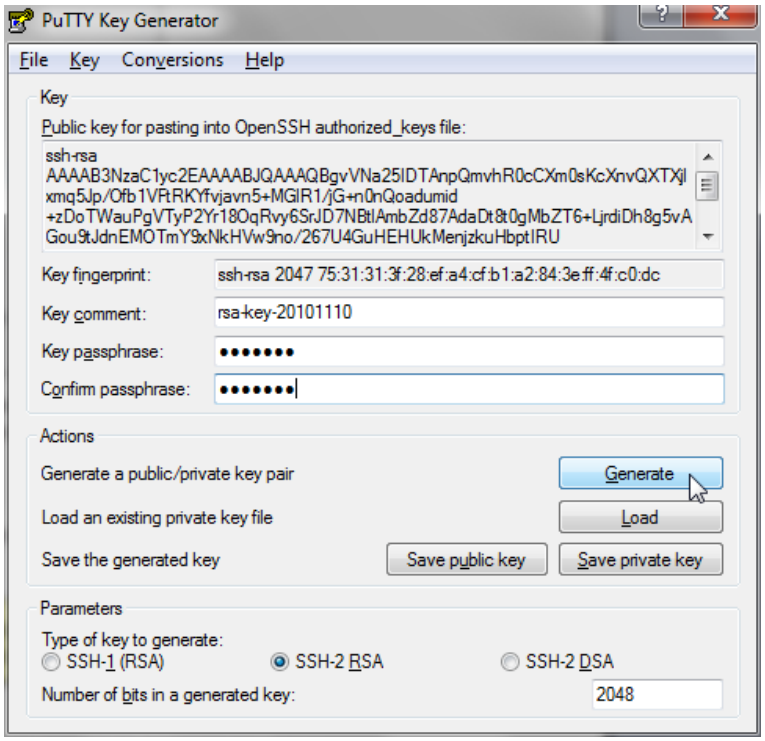
2. Если на локальной машине используется система Windows, то для генерирования ключей можно воспользоваться ssh-клиентом PuTTY, файловым клиентом WinSCP (в их состав входит генератор ключей PuttyGen) или другим ssh-клиентом, поддерживающим нужные возможности.

   Приведем пример использования утилиты PuttyGen в составе приложения PuTTY для Windows.

   a. Запустите программу, выставьте тип ключа **SSH-2 RSA**, длину **2048** бит и щелкните на кнопке **Generate**.



   b. Когда начнется генерирование ключа, вас попросят провести мышью по пустой области окна для рандомизации ключа. Затем появится окно:



c. Сохраните публичный ключ, нажав на кнопку **Save public key** и выбрав для ключа имя файла, например, `id_rsa.pub`.

   Этот ключ передается администратору кластера.

d. Введите и подтвердите произвольный непустой пароль для ключа в полях **Key passphrase** и **Confirm passphrase**. Нажмите на кнопку **Save private key** для сохранения приватного ключа под тем же именем файла, но с другим расширением (PuTTY предлагает `.ppk`).

Эти ключи будут использоваться для доступа к кластеру через клиент PuTTY.

3. В случае использования других программ генерирования ключей обратитесь к их документации.

1.2. Вход на кластер

Вход на кластер выполняется из Linux или Windows через соответствующие ssh-клиенты, установленные у пользователя.

Предварительные замечания

- Администратор должен предоставить имя пользователя, адрес/имя хоста и номер порта для доступа.
- Учетная запись пользователя должна быть создана с применением переданного им публичного ключа.

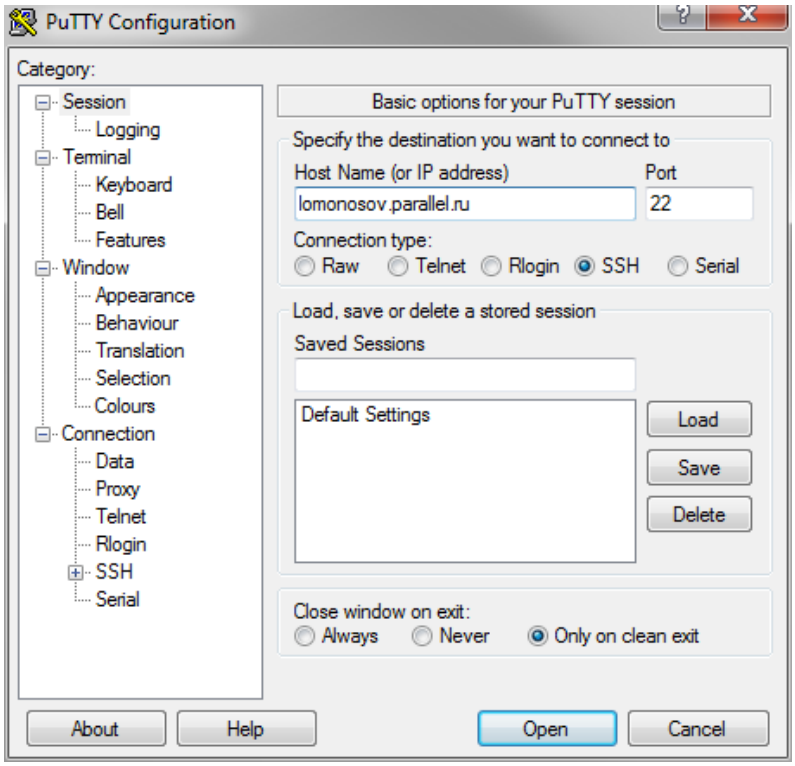
Порядок выполнения

1. Для входа с помощью утилит пакета OpenSSH из среды Linux введите команду:

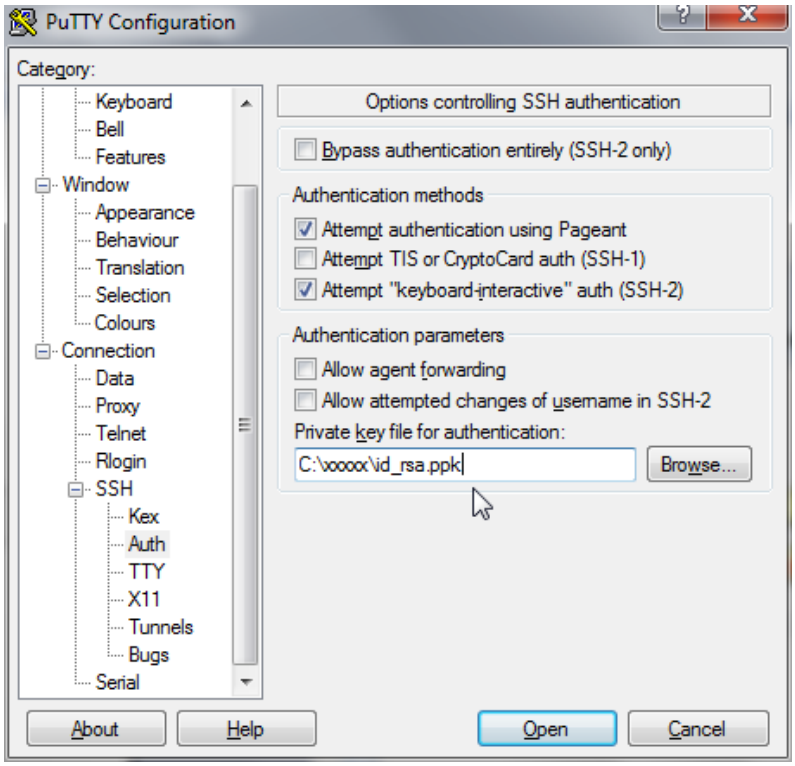
```
ssh -p<port> <username>@<host>
```

2. Для входа из Windows с помощью программы PuTTY выполните следующее:

   a. В категории **Session**, укажите в поле **Host Name (or IP address)** предоставленный администратором хост, а в поле **Port** — номер порта.



   b. В категории **Connection** → **SSH** → **Auth**, укажите приватный ключ в поле **Private key file for authentication**.



c. Щелкните на кнопке **Open**. Откроется окно терминала, в котором вам будет предложено ввести имя пользователя. Введите его и нажмите <Enter> для входа на кластер.

   Командная оболочка появится в этом же окне консольного терминала.

1.3. Смена пароля после первого входа

Постановка задачи

В то время как вход в командный интерфейс выполняется по ключу, вход в графический интерфейс требует ввода пароля. При создании учетной записи автоматически генерируется пароль, доступный для просмотра из домашнего каталога пользователя на кластере, а потому небезопасный. Пользователю следует сменить его на новый пароль через командный интерфейс после первого же входа на кластер.

Порядок выполнения

1. Просмотрите автоматически сгенерированный для вас пароль командой

   ```
   cat ~/.password
   ```

2. Введите команду **`passwd`** и поменяйте текущий пароль на любой желательный (не менее 8 символов, из букв и цифр):

   a. В ответ на ожидание введите текущий пароль, просмотренный в пункте 1. Нажмите <Enter>.
   b. Введите новый пароль, нажмите <Enter>.

   После этого вы получаете право доступа к графическому интерфейсу пользователя под вашим именем и новым, установленным вами паролем. Новый па-

роль не будет храниться в файле в вашем домашнем каталоге, и просмотреть его командой **cat** будет нельзя.

3. Удалите файл с начальным, уже недействительным, паролем, введя команду

```
rm ~/.password
```

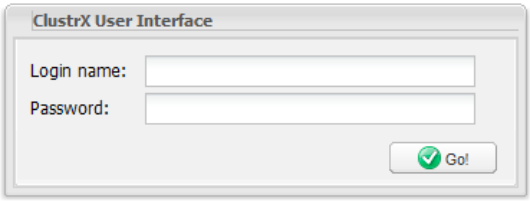
2. Графический интерфейс

Постановка задачи

Обращение к кластеру можно осуществлять через интерактивный графический Web-интерфейс пользователя ОС Clustrx, работающий в интернет-браузере. Он предоставляет пользователю как режим запуска новой задачи на выполнение, так и режим отслеживания ее выполнения в общей очереди кластера.

Порядок выполнения

1. Для входа в Web-интерфейс введите предоставленный администратором кластера адрес в адресную строку браузера. Появится окно входа:

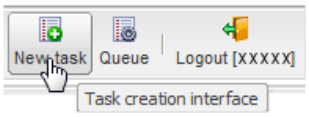


2. Введите в нем ваше имя (логин) и пароль, щелкните на кнопке **Go**.

   Откроется графический интерфейс пользователя, вид и возможности которого отличаются для пользователей разного уровня привилегированности. Так, обычные пользователи увидят только отведенные им каталоги и состояние своих собственных вычислительных задач, смогут ставить их в очередь, останавливать, забирать данные. Привилегированные пользователи (например, администраторы групп) смогут делать это и для других задач.

   После входа в окне браузера появится один из режимов работы с графическим интерфейсом (какой именно — зависит от настроек и полномочий пользователя, например, **Queue**).

3. Для входа в режим постановки задачи в очередь щелкните на кнопке **New Task** в этой панели.



   В зависимости от статуса пользовательской учетной записи и ее настроек интерфейс может сразу после входа открыться в режиме **New Task**.

Глава 2. Работа с файлами на кластере

Постановка задачи

Для обеспечения работы пользовательских вычислительных задач бывает необходимо загрузить на кластер те или иные файлы данных, а также получить результаты работы с кластера. Основные операции с файлами данных или результатов задачи:

- выгрузка из локального местонахождения на сервер кластера;
- загрузка с сервера в локальное местонахождение пользователя.

Порядок выполнения

- Для передачи файлов могут использоваться утилиты пакета OpenSSH через командную строку Linux.
- В графическом Web-интерфейсе ОС Clustrx имеется файловый менеджер для управления файлами пользователя.
- Для пользователей, у которых на локальной машине установлена система Windows, есть возможность использовать программы с графическим интерфейсом для передачи файлов по ssh-соединениям.

1. Работа с OpenSSH в Linux/UNIX

1.1. Использование команды `scp`

Постановка задачи

Для передачи данных на кластер и загрузки их с кластера может использоваться команда **`scp`** пакета OpenSSH.

Порядок выполнения

1. Передача архива данных на кластер:

   ```
   scp <data.tar.bz2> <user>@<host:>
   ```

2. Если необходимо выгрузить каталог, а не файл, команду следует дополнить ключом **`r`**:

   ```
   scp -r <datadir> <user>@<host:>
   ```

   Первый аргумент следует заменить на реальный архив данных по локальному пути, а второй — на реальное местонахождение пользователя на кластере.

3. Для передачи данных в сторону пользователя используется следующая команда:

   ```
   scp <user>@<host>:</path/data.tar.bz2> <c:\localpath>
   ```

4. Для передачи каталогов используется та же команда с ключом **`r`**.

```
scp –r <user>@<host>:</path/datadir> <c:\localpath>
```

Первый аргумент следует заменить на реальное местонахождение данных пользователя на кластере, а второй — на его локальный путь.

1.2. Использование утилиты `sshfs`

Постановка задачи

Для удобства в работе с файлами на кластере может использоваться утилита **`sshfs`** — клиент для SSH-соединений, позволяющий смонтировать в локальный каталог файловую систему удаленного хоста и работать с ней как с локальным ресурсом.

Порядок выполнения

- Простейший синтаксис команды:

```
sshfs <hostname>: <mountpoint>
```

По умолчанию подключается домашний каталог, но можно также указать каталог после двоеточия, если это позволят допуски пользователя.

См. также

➥ **`man sshfs`** — системная документация к данной утилите с дополнительными ключами и аргументами.

1.3. Использование программы `sftp`

Постановка задачи

Для передачи файлов с кластера и на кластер может использоваться программа **`sftp`** из пакета OpenSSH.

Порядок выполнения

- Команда для загрузки файла `file1` с сервера и записи его в текущий каталог под именем `file1` (или `file2`, если это имя указано):

```
sftp [[user@]host[:<file1> [<file2>]]]
```

- Команда для выполнения **`batchfile`**, содержащего команды **`sftp`**, после подключения к серверу:

```
sftp -b <batchfile> [<user>@]host]
```

В качестве имени файла может быть указан символ **`-`**. В этом случае **`batchfile`** читается из стандартного потока ввода.

См. также

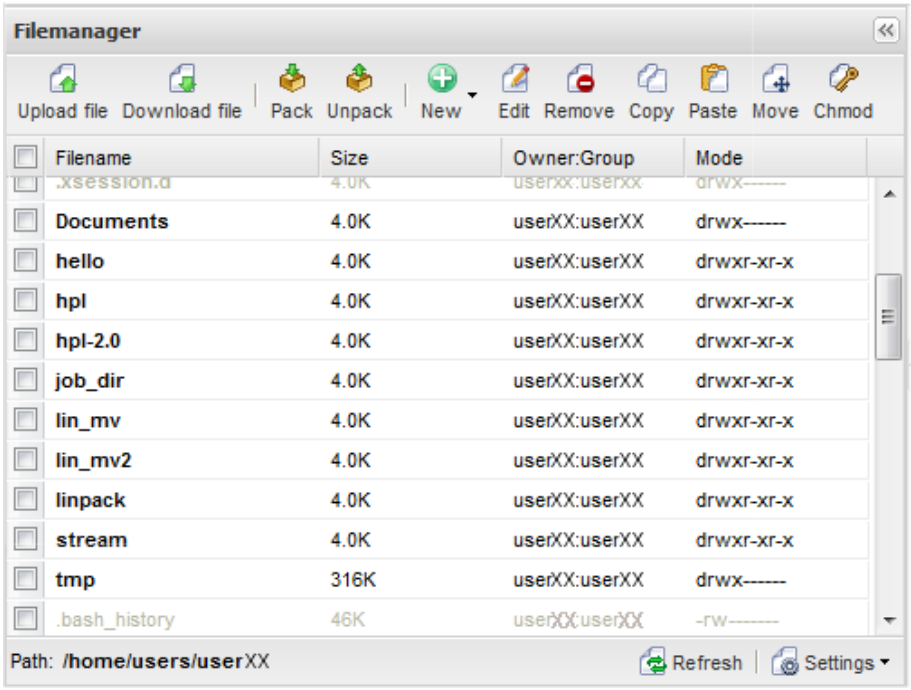
➥ **`man sftp`** — системная документация к программе.

➥ http://www.openbsd.org/cgi-bin/man.cgi?query=sftp&sektion=1 — то же в Интернете.

2. Работа с файловым менеджером Clustrx

Постановка задачи

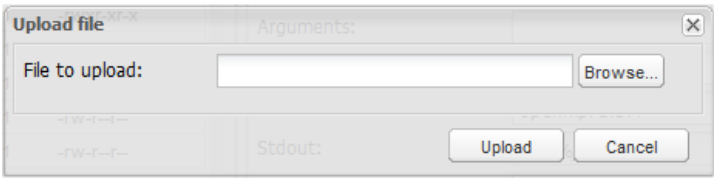
Все операции создания, копирования, перемещения, удаления файлов и т.п. могут выполняться через файловый менеджер графического Web-интерфейса пользователя ОС Clustrx, который находится в левой части окна режима **New Task**. Для этого служит панель кнопок в его верхней части:



Порядок выполнения

1. Две первые кнопки, **Upload file** и **Download file** — позволяют обмениваться файлами между локальным расположением пользователя и его домашним каталогом на кластере.

   a. По щелчку на кнопке **Upload file** откроется диалоговое окно для выбора файла на локальном компьютере:

   Щелкните на кнопке **Browse** для выбора файла, а затем на **Upload** для его загрузки в домашний каталог на кластере.

   b. Для копирования файла из домашнего каталога на кластере выберите файл в списке, отметьте его «птичкой» в крайней левой клетке, и щелк-

ните на кнопке **Download**, после чего вам будет предложено сохранить файл у себя локально, обычными средствами браузера.

2. Остальные кнопки верхней панели также выполняют полезные функции работы с файлами:

   a. **Pack** / **Unpack** используются для упаковки/распаковки архивов.
   b. **New** позволяет создавать новые файлы и каталоги.
   c. **Edit** дает возможность редактировать текстовые файлы в окне редактора.
   d. **Remove**, **Copy**, **Paste**, **Move** служат для обычных операций удаления, копирования, вставки, перемещения файлов (например, между подкаталогами домашнего каталога пользователя) по аналогии с обычными средствами операционных систем и пользовательских интерфейсов.
   e. **Chmod** изменяет режим доступа к файлу.

3. Передача файлов из Windows

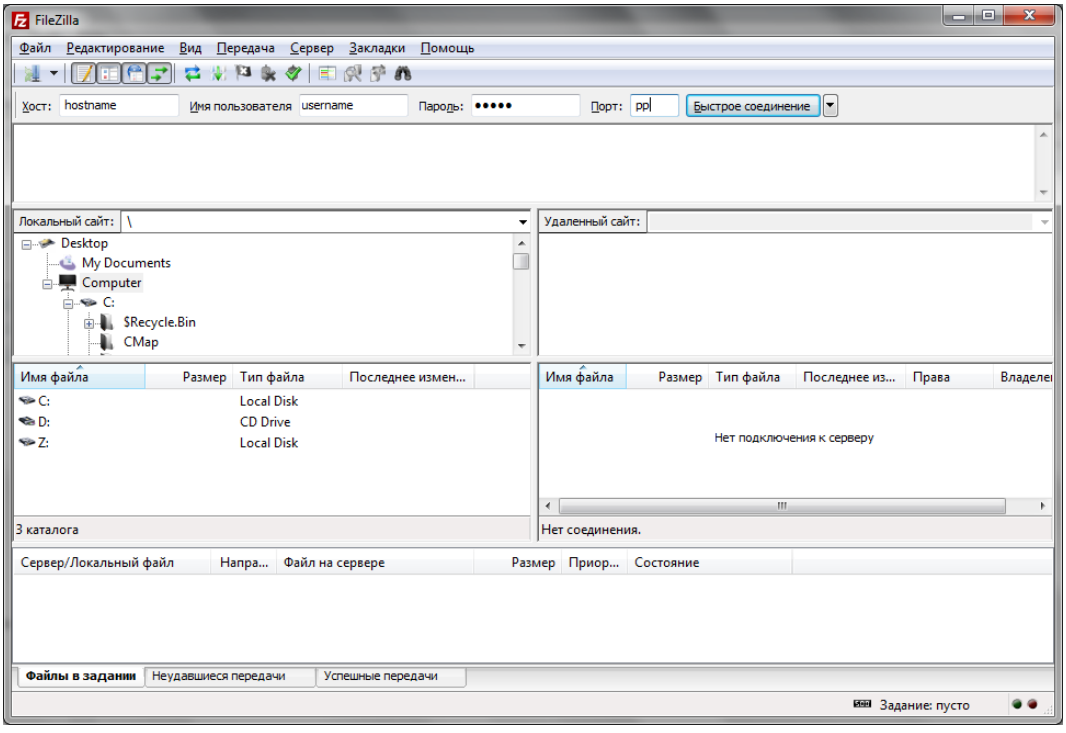
Постановка задачи

Для пользователей, у которых на локальной машине установлена система Windows, существуют реализации OpenSSH с графическим интерфейсом — например, WinSCP или FileZilla — которые поддерживают передачу файлов и не требуют работы с командной строкой.

Для примера рассмотрим бесплатную программу FileZilla, доступную на `http://sourceforge.net/projects/filezilla/`.

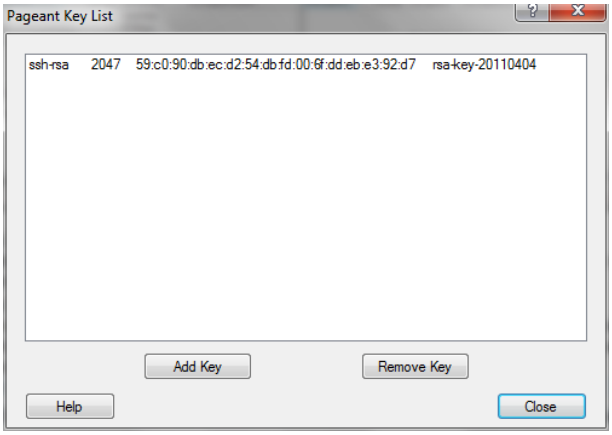
Порядок выполнения

1. Запустите программу FileZilla. Ее рабочее окно имеет вид:



2. Введите в полях **Хост**, **Имя пользователя**, **Пароль**, **Порт** соответствующие данные, предоставленные администратором, и нажмите на кнопку **Быстрое соединение**.

3. Для входа на кластер не по паролю, а по ключу (что является стандартным способом для пользователей) необходимо предварительно, до попытки соединения с кластером через FileZilla, запустить агент программы PuTTY (Pageant) и указать в нем свой приватный ключ. Пример ключа в окне Pageant:



FileZilla взаимодействует с этим агентом и использует заданные в нем ключи для входа на кластер.

Не пользуйтесь собственными средствами FileZilla для работы с ключами. Так, если выбрать в FileZilla пункт меню **Редактирование** → **Настройки** и в открывшемся окне настройки войти в раздел **Соединение** → **SFTP**, то программа предложит выбрать ключ для входа на кластер. Однако воспользоваться этим средством для входа на кластер в настоящее время нельзя, т.к. FileZilla не поддерживает ключи, защищенные паролями, а именно такие используются на кластерном комплексе.

Результат

После установки соединения на правой панели появится содержимое домашнего каталога пользователя на кластере. На левой панели, как видно из рисунка, показано содержимое локального диска пользователя. После открытия соединения между этими двумя панелями становится возможен обмен файлами (копирование, перемещение), как во многих распространенных файловых менеджерах для Windows. Можно также открывать файлы для удаленного редактирования, используя программы, установленные пользователем у себя в локальной системе.

Глава 3. Компиляция программ

Постановка задачи

Для компиляции вычислительных задач пользователя на кластере выделено специальное виртуальное окружение под именем `compiler`. В нем установлены все доступные версии MPI и компиляторов, см. Раздел 1.

Среда запуска вычислительной задачи (на вычислительных узлах кластера) отделена от среды компиляции. В среде компиляции запускать задачу на выполнение запрещается.

Для последующего корректного запуска скомпилированной программы необходимо согласовать среду выполнения со средой компиляции — в частности, по используемой версии MPI (Глава 4, Раздел 8).

Среда компиляции в системном отношении представляет собой то же самое, что и среда выполнения на вычислительных узлах (Compute Node Linux), плюс средства разработки: компиляторы, библиотеки и другие пакеты.

Порядок выполнения

- Для входа в среду компиляции введите команду

  ```
  ssh compiler
  ```

  При этом должно быть открыто SSH-соединение с кластером, см. Глава 1, Раздел 1.

- Для возвращения из среды компиляции в основную командную оболочку пользователя (откуда можно запускать задачи на выполнение) введите команду **`exit`**.

1. Выбор MPI и компилятора

Постановка задачи

При наличии нескольких версий MPI и компиляторов они устанавливаются по нестандартным путям (с префиксами). Одновременно можно работать с одной версией MPI и одним комплектом компиляторов. Для выбора версии MPI и/или компилятора необходимо вручную настраивать переменные окружения, что неудобно. Для облегчения процесса работы с несколькими версиями MPI и компиляторов создана утилита **`mpienv`**. Ее основные возможности:

- просмотр списка доступных MPI и компиляторов;
- выбор необходимой версии MPI и/или компилятора;
- запоминание последнего выбранного состояния.

Порядок выполнения

- Синтаксис командной строки утилиты **mpienv**:

```
mpienv -s|-l|-m <MPI> -c <COMPILER>
```

Опция	Описание
-l	Просмотр списка доступных MPI и компиляторов.
-m	Установка текущей версии MPI.
-c	Установка текущей версии компилятора.
-s	Просмотр текущих настроек окружения.
-r	Считывание файла настроек и применение записанной в нем конфигурации.
-h	Вывод справки об использовании команды.

- Настройки окружения пользователя, выполненные с помощью опций **mpienv**, применяются для его текущей сессии, а также сохраняются в файл, находящийся в домашнем каталоге пользователя (обычно это `~/.mpienv`).
  - Для разных сессий пользователя, использующих один домашний каталог, будет использован один и тот же файл сохранения настроек, потому текущие настройки могут не совпадать с записанными в этом файле.
  - При логине пользователя автоматически применяются записанные в файле сохранения настройки. В случае отсутствия этого файла будут использованы версии MPI и компилятора, заданные по умолчанию.
- Пример просмотра списка доступных версий:

```
$ mpienv -l
MPI             COMPILER
openmpi-1.2
                gnu-4.1
                gnu-4.3
                gnu-4.4
                intel-11.1.072
openmpi-1.3
                gnu-4.1
                gnu-4.3
                gnu-4.4
                intel-11.1.072
openmpi-1.4
                gnu-4.1
                gnu-4.3
                gnu-4.4
                intel-11.1.072
impi-4.0
                intel-11.1.072
```

- Пример просмотра текущих MPI и компилятора:

```
$ mpienv -s
MPI: openmpi-1.4
COMPILER: gnu-4.3
```

- Пример выбора MPI и компилятора:

```
$ mpienv -m openmpi-1.4 -c intel-11.1.072
```

- Для `impi-4.0.1` предоставляется два вида команд-«оберток» над компиляторами, облегчающих работу с этой версией MPI:
  - `mpi**` — используются компиляторы `gcc`;
  - `mpii**` — используются компиляторы Intel.

  Пример просмотра версии компилятора `gcc`:

```
$ mpicc --version
x86_64-alt-linux-gcc (GCC) 4.4.5 20101001 (ALT Linux 4.4.5-tmc0)
Copyright (C) 2010 Free Software Foundation, Inc.
...
```

  Пример просмотра версии компилятора Intel:

```
$ mpiicc --version
icc (ICC) 12.0.0 20101006
Copyright (C) 1985-2010 Intel Corporation. All rights reserved.
```

2. Компиляторы GNU

Для просмотра имеющихся в `compiler`-окружении версий компиляторов GNU используйте команду

```
mpienv -l
```

Выбор компилятора (а также, при необходимости, версии MPI) см. Раздел 1.

Использование требуемого компилятора можно посмотреть по его системной документации. Для этого введите следующую команду, подставив имя конкретного компилятора:

```
man <compiler>
```

В `compiler`-окружении вместе с компиляторами установлены все необходимые для них ресурсы, в т.ч. документация к ним.

В неинтерактивном SSH-сеансе связи с `compiler`-окружением компилятор `mpicc` в настоящее время не обнаруживается. При вводе команды

```
$ ssh compiler mpicc
```

может выдаваться ошибка `mpicc: command not found`. В интерактивном сеансе эта проблема отсутствует.

В качестве временного решения используйте команду

```
ssh compiler "sh -l -c mpicc"
```

3. Компиляторы Intel

Компиляторы Intel могут дать существенный выигрыш в ресурсоемких задачах. Для просмотра имеющихся в `compiler`-окружении версий компиляторов Intel используйте команду

```
mpienv -l
```

Выбор компилятора (а также, при необходимости, версии MPI) см. Раздел 1.

Использование требуемого компилятора можно посмотреть по его системной документации. Для этого введите следующую команду, подставив имя конкретного компилятора:

```
man <compiler>
```

В `compiler`-окружении вместе с компиляторами установлены все необходимые для них ресурсы, в т.ч. документация к ним.

4. Библиотеки

В среде компиляции (`compiler`-окружении) установлено большое количество библиотек для работы вычислительных приложений пользователей и других средств разработки.

Как в среде компиляции, так и в исполнительной среде на вычислительных узлах пользователь находится в командной оболочке Linux **`bash`**. Это означает, что ему доступны стандартные средства Linux для просмотра установленных в системе RPM-пакетов и других ресурсов, например, команда, выводящая список всех пакетов:

```
rpm -qa [| less]
```

С помощью этой команды (см. ее документацию по **`man rpm`**) можно выяснить полный состав установленных средств разработки.

При возникновении вопросов по работе с установленным в среде компиляции программным обеспечением третьей стороны обращайтесь к документации соответствующих программных продуктов.

Глава 4. Запуск задач из командной строки

1. Просмотр доступных ресурсов

Постановка задачи

Пользователь имеет возможность определить, какие разделы (*partitions*) существуют на кластере, какие узлы они включают, и общее состояние кластера. Это может оказаться полезно и перед тем, как ставить задачу в очередь, и во время ее выполнения.

Порядок выполнения

- Введите команду **`sinfo`**:

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
test         up   infinite     97  down* compiler, ...
test         up   infinite     64   idle node1-038-[01-32],..
regular*     up   infinite   1236 drain* node1-006-[01-32], ...
regular*     up   infinite     54  down* node1-001-15,...
regular*     up   infinite     32  drain node1-003-27,...
regular*     up   infinite   1029  alloc node1-001-[01-14],...
regular*     up   infinite    849   idle node1-001-[24-32],...
hdd          up   infinite      3 drain* node2-001-[01-02],...
hdd          up   infinite      5  down* node2-018-[05,09],...
hdd          up   infinite      1   comp node2-026-10
hdd          up   infinite     10  drain node2-001-03,...
hdd          up   infinite     16  alloc node2-019-[07-10],...
hdd          up   infinite    225   idle node2-001-[04-10],...
main         up   infinite   1239 drain* node1-006-[01-32],...
main         up   infinite     59  down* node1-001-15,...
main         up   infinite      1   comp node2-026-10
main         up   infinite     42  drain node1-003-27,...
main         up   infinite   1045  alloc node1-001-[01-14],...
main         up   infinite   1074   idle node1-001-[24-32],...
cell         up   infinite     23  down* node3-001-[04-08],...
cell         up   infinite      3   idle node3-001-[01,03,09]
```

  - Троеточия заменяют длинные списки узлов, выведенные в нескольких строках.
  - Символ * после имени раздела указывает на то, что это — раздел по умолчанию для постановки задач.
  - Все разделы находятся в состоянии `UP`. Это означает, что они доступны для запуска задач.
  - Информация о вычислительных узлах раздела сгруппирована по их состояниям. В этом случае узлы `compiler`, `node1-025-[01-32]` и т.д. находятся в состоянии `down`.
  - Звездочка * после состояния `down` указывает на то, что узлы не отвечают.

– Используется краткое представления списка узлов с общим префиксом `node1-025-` и числовыми диапазонами или отдельными числами.

См. также

➥ **`man sinfo`** — системная документация по команде **`sinfo`**. У команды есть много опций, позволяющих просматривать информацию о кластере, фильтровать и форматировать ее в зависимости от потребностей.

2. Просмотр очереди задач

Постановка задачи

Пользователь имеет возможность просмотреть список текущих задач в очереди на кластере. Это может оказаться полезно и перед тем, как ставить свою задачу в очередь, и во время ее выполнения.

Порядок выполнения

- Используйте команду **`squeue`**:

```
$ squeue
JOBID PARTITION  NAME     USER  ST  TIME  NODES NODELIST(REASON)
25879   regular   run   userXX   R  0:09      1 node1-061-28
25880   regular   run   userXX   R  0:06      1 node1-090-07
25881   regular   run   userXX   R  0:05      1 node1-090-13
25882   regular   run   userXX   R  0:04      1 node1-001-26
```

Здесь:

– Поле `JOBID` отображает идентификатор задачи.

– Поле `ST` — состояние задачи. Четыре задачи находятся в состоянии выполнения (`R`, сокращение от `Running`).

– Поле `TIME` показывает, сколько времени задача запущена, в формате *`дни-часы:минуты:секунды`*.

– Поле `NODELIST(REASON)` показывает, на каких узлах запущена задача, или причину, почему эта задача все еще находится в ожидании. Типичные причины ожидания задач:

  - `Resources` — ожидание, пока появится необходимое количество свободных ресурсов;
  - `Priority` — в очереди имеется более приоритетная задача.

См. также

➥ **`man squeue`** — системная документация по команде **`squeue`**. У команды есть много опций, позволяющих просматривать информацию о текущих задачах, а также форматировать и фильтровать ее.

3. Интерактивное выполнение задачи

Постановка задачи

Рассматривается пример запуска задачи пользователем в интерактивном режиме, с занятием определенного количества ресурсов кластера.

Нельзя выполнять запуск задачи в `compiler`-окружении. Среды компиляция и выполнения задачи — это отдельные рабочие среды, см. Глава 3.

Порядок выполнения

- Задача с именем **`hostname`** запускается на трех узлах (ключ **`N3`**) с помощью команды **`srun`**:

```
$ srun -N3 hostname
node1-006-01.lomonosov.parallel.ru
node1-006-02.lomonosov.parallel.ru
node1-006-03.lomonosov.parallel.ru
```

  Раздел (партиция) кластера не указан, поэтому будет использоваться заданный по умолчанию.

  Также по умолчанию будет запущен один процесс на узел.

- В немного измененном примере запустим четыре процесса (ключ **`n4`**):

```
$ srun -n4 hostname
node1-006-01.lomonosov.parallel.ru
node1-006-01.lomonosov.parallel.ru
node1-006-01.lomonosov.parallel.ru
node1-006-01.lomonosov.parallel.ru
```

  Количество узлов не указано, поэтому по умолчанию будет занято необходимое количество.

См. также

➥ **`man srun`** — системная документация к команде **`srun`**. Команда имеет много опций для управления тем, какой ресурс будет выделен и как процессы распределятся между выделенными ресурсами.

4. Запуск задачи в пакетном режиме

Постановка задачи

Рассматривается запуск вычислительной задачи пользователем на кластере в пакетном режиме.

Нельзя выполнять запуск задачи в `compiler`-окружении. Среды компиляция и выполнения задачи — это отдельные рабочие среды, см. Глава 3.

Порядок выполнения

- Для работы задачи в этом режиме необходимо создать скрипт ее запуска.

  В качестве примера воспользуемся скриптом **run_hostname.sh**, в котором вызывается **srun**.

```
$ cat run_hostname.sh
#!/bin/sh
srun hostname

$ sbatch -N3 ./run_hostname.sh
$ cat slurm-20415.out
node1-006-01.lomonosov.parallel.ru
node1-006-03.lomonosov.parallel.ru
node1-006-02.lomonosov.parallel.ru

$ sbatch -n4 ./run_hostname.sh
sbatch.bin: Submitted batch job 20416

$ cat slurm-20416.out
node1-006-01.lomonosov.parallel.ru
node1-006-01.lomonosov.parallel.ru
node1-006-01.lomonosov.parallel.ru
node1-006-01.lomonosov.parallel.ru
```

  При использовании команды **srun** в скриптах ей не нужно передавать дополнительных ключей для описания ресурсов и запускаемых процессов. Это необходимо только для **sbatch**.

5. Распределение процессорных ресурсов для вычислительных задач

Постановка задачи

В текущей версии менеджера ресурсов (SLURM) для ОС Clustrx имеются особенности при выделении и отображении запрашиваемого количества ресурсов — вычислительных узлов и их процессоров — а также при распределении параллельных процессов между узлами. Это следует иметь в виду при постановке задачи в очередь и анализе ее выполнения.

Порядок выполнения

- Чтобы запросить определенное количество вычислительных узлов (ВУ) для задачи, используется ключ **-N**. Пример:

```
$ sbatch -N 2048 run_hostname
```

  Задача будет запущена на 2048 ВУ, с одним процессом на каждом узле.

- Чтобы запросить определенное количество параллельных процессов для задачи, используется ключ **-n**. Пример:

```
sbatch -n 1024 run_hostname
```

Задача будет запущена в виде 1024 параллельных процессов. Каждый процесс будет занимать 1 ядро CPU. Количество выделенных ВУ будет зависеть от количества ядер CPU на них.

- При совместном использовании ключей **-N** и **-n** производится коррекция в сторону большего количества выделяемых ВУ, если этого требует значение ключа **-n**.

  Количество параллельных процессов задачи будет строго соответствовать ключу **-n**. Но при этом:

  – если количества ВУ, запрашиваемого ключом **-N**, недостаточно для размещения процессов, то оно будет скорректировано в большую сторону до нужной величины;

  – если количество ВУ по ключу **-N** больше, чем минимально необходимо для размещения процессов, то процессы будут равномерно распределяться по затребованному количеству ВУ.

  Примеры:

  a. Переопределение количества ВУ в большую сторону:

  ```
  $ sbatch -N4 -n64 run_hostname
  ```

  Выделяется 8 ВУ с 8 процессами на каждом (по количеству процессоров на одном ВУ). Значение `-N4` переопределяется на `-N8`.

  b. Равномерное распределение по узлам:

  ```
  $ sbatch -N4 -n8 run_hostname
  ```

  Выделяется 4 ВУ, с 2 процессами на каждом.

- Совместно с предыдущими ключами можно использовать ключ **--ntasks-per-node**, задающий максимально допустимое количество параллельных процессов, запускаемых на отдельном узле. Этот ключ изменяет поведение ключей **-N** и **-n**, и тем самым может усложнить расчет выделяемых ресурсов.

  Примеры:

  a. Совместное использование `n` и `ntasks-per-node`:

  ```
  $ srun -n4 --ntasks-per-node=2 hostname
  ```

  Задаче будет выделено 2 ВУ, с 2 процессами на каждом — всего 4 процесса.

  b. Совместное использование `N` и `ntasks-per-node`:

  ```
  $ srun -N4 --ntasks-per-node=2 hostname
  ```

  Задаче будет выделено 4 ВУ, с 2 процессами на каждом — всего 8 процессов.

  c. Совместное использование трех ключей:

  ```
  $ srun -N8  -n8 --ntasks-per-node=2 hostname
  ```

  Задаче будет выделено 8 ВУ, с 1 процессом на каждом — всего 8 процессов. Приоритет — по значению `N`.

  d. Совместное использование трех ключей:

```
$ srun -N4  -n32 --ntasks-per-node=2 hostnam
```

Задаче будет выделено 16 ВУ, с 2 процессами на каждом — всего 32 процесса. Приоритет — по значениям `n` и `ntasks-per-node`, значение `N` переопределяется в большую сторону.

При использовании ключа `--ntasks-per-node` возможно неправильное отображение требуемого количества ВУ для задачи в состоянии `PENDING`. Фактически отображается то количество ВУ, которое требовалось бы без применения этого ключа. При переходе в состояние `RUNNING` количество ВУ отображается правильно.

6. Запуск однопроцессорных задач

Постановка задачи

Запуск не-MPI-задач пользователя через команду **`sbatch`** может выполняться посредством простого системного скрипта `/usr/bin/run`. Это особенно удобно при постановке в очередь однопроцессорной задачи, одной или нескольких.

Также этот скрипт можно использовать для запуска задач, скомпилированных под MVAPICH2.

Порядок выполнения

- Запуск задачи выглядит следующим образом:

```
sbatch -n <num> run /path/to/executable
```

Пример:

```
$ sbatch -n2 run hostname
sbatch: Submitted batch job 5067

$ cat slurm-5067.out
node1-001-26.lomonosov.parallel.ru
node1-001-26.lomonosov.parallel.ru
```

7. Отмена выполнения задачи

Постановка задачи

Уже запущенную либо ожидающую в очереди на кластере вычислительную задачу можно снять — прекратить или отменить ее выполнение. Это делается с помощью команды **`scancel`**, которой передается в качестве аргумента идентификатор задачи.

Порядок выполнения

1. Чтобы определить идентификатор нужной задачи, выполните команду **`squeue`** для просмотра очереди запущенных вами задач:

```
$ squeue
JOBID PARTITION  NAME    USER  ST  TIME  NODES NODELIST(REASON)
25888   regular   run  userXX   R 1:19      1 node1-001-26
```

Если идентификатор задачи известен, этот шаг можно опустить.

2. Выполните команду **`scancel`** для отмены задачи:

```
$ scancel 25888
```

3. Теперь можно проверить, снята ли задача, снова просмотрев очередь:

```
$ squeue
JOBID PARTITION  NAME    USER  ST  TIME  NODES NODELIST(REASON)
```

Результат

Вычислительная задача с идентификатором 25888 убрана из очереди.

8. MPI-задачи

Постановка задачи

Для корректного запуска MPI-задач необходимо, чтобы исполнительная среда совпадала по версии MPI со средой, в которой задача была скомпилирована, см. Глава 3, Раздел 1.

В настоящей версии системы Clustrx, установленной на кластерном комплексе, среда компиляции (`compiler`-окружение) отделена от исполнительной среды, см. Глава 3. Нельзя запускать задачу на выполнение в той же среде компиляции, где она была скомпилирована.

Порядок выполнения

- Если компиляция была выполнена в среде, настроенной с помощью утилиты **`mpienv`** (Глава 3, Раздел 1), и настройки среды компиляции после этого не изменялись, то для согласования с нею исполнительной среды достаточно выполнить команду

```
mpienv -r
```

- Если настройки среды компиляции или исполнительной среды по MPI каким-либо образом изменялись, то для корректного запуска задачи необходимо перед ее выполнением явно указать версию MPI, с которой она компилировалась:

```
mpienv -m <mpi_version>
```

8.1. Запуск OpenMPI-задач

Постановка задачи

Запуск OpenMPI-задач на кластере поддерживается только в пакетном режиме. Для этого служит команда **sbatch** с использованием стандартного скрипта **ompi**.

Порядок выполнения

1. Выбрать нужную версию OpenMPI (Глава 3, Раздел 1). В случае необходимости выполнить компиляцию и перейти непосредственно к запуску.
2. При необходимости явно задать размер стека для OpenMPI-задачи воспользуйтесь одним из двух способов:
   a. При работе в командной оболочке `bash` установите переменные `OMP_STACKSIZE` или `KMP_STACKSIZE` с помощью команды **env** *имя_переменной=значение*.
   b. При работе в графическом интерфейсе запуска задач (Глава 5, Раздел 1) установите те же переменные в списке **Environment** под центральной областью параметров задачи. В поле **Variable** следует ввести имя переменной, в поле **Value** — ее значение.
3. Выполнить запуск следующей командой:

```
sbatch [<sbatch-options>] ompi <path-to-program>
       [<program-options>]
```

Пример

Запустим программу **./hello** на 8 процессов (опция **n8**) и посмотрим результат выполнения:

```
$ sbatch -n8 ompi ./hello
sbatch.bin: Submitted batch job 20428

$ cat slurm-20428.out
Hello, world, I am 0 of 8
Hello, world, I am 1 of 8
Hello, world, I am 2 of 8
Hello, world, I am 3 of 8
Hello, world, I am 5 of 8
Hello, world, I am 7 of 8
Hello, world, I am 4 of 8
Hello, world, I am 6 of 8
```

8.2. Запуск MVAPICH-задач

Постановка задачи

Запуск MVAPICH-задач поддерживается в пакетном режиме и в режиме реального времени (интерактивном).

Порядок выполнения

1. Выбрать нужную версию OpenMPI (Глава 3, Раздел 1). В случае необходимости выполнить компиляцию и перейти непосредственно к запуску.

2. Запуск в пакетном режиме осуществляется командой **sbatch** с использованием стандартного скрипта **mvapich**:

```
sbatch [<sbatch-opts>] mvapich <path-to-prog> [<prog-opts>]
```

   Например, запустим программу **./hello** в виде 16 процессов (**-n16**) на 2 узла (**-N2**) и посмотрим результат выполнения:

```
$ sbatch -n16 -N2 mvapich ./hello
sbatch.bin: Submitted batch job 20529

$ cat slurm-20529.out
Hello, world, I am 0 of 16
Hello, world, I am 8 of 16
Hello, world, I am 9 of 16
Hello, world, I am 10 of 16
Hello, world, I am 11 of 16
Hello, world, I am 12 of 16
Hello, world, I am 13 of 16
Hello, world, I am 14 of 16
Hello, world, I am 15 of 16
Hello, world, I am 1 of 16
Hello, world, I am 2 of 16
Hello, world, I am 4 of 16
Hello, world, I am 5 of 16
Hello, world, I am 6 of 16
Hello, world, I am 7 of 16
Hello, world, I am 3 of 16
```

3. Запуск MVAPICH-задачи в интерактивном режиме реального времени осуществляется командой **srun** с использованием ключа **--mpi=mvapich**:

```
srun --mpi=mvapich [<sbatch-opts>] <path-to-prog> [<prog-opts>]
```

   Например, запустим программу **./hello**, 5 процессов (**-n5**):

```
$ srun --mpi=mvapich -n5 ./hello
Hello, world, I am 0 of 5
Hello, world, I am 1 of 5
Hello, world, I am 2 of 5
Hello, world, I am 3 of 5
Hello, world, I am 4 of 5
```

8.3. Запуск задач с Intel MPI

Предварительные замечания

Выбор версии MPI и компилятора, а также компиляцию см. Глава 3, Раздел 3, Глава 3, Раздел 1.

Порядок выполнения

- В системе установлен скрипт **impi** для запуска Intel MPI-приложений. Он позволяет выполнять запуск в виде:

```
sbatch [<sbatch-opts>] impi <path-to-program> [<prog-opts>]
```

Пример запуска задачи на 12 узлах и результаты ее вывода:

```
$ sbatch -n 12 impi ./hello
sbatch.bin: Submitted batch job 11853

$ cat slurm-11853.out
Hello, world, I am 0 of 12
Hello, world, I am 5 of 12
Hello, world, I am 1 of 12
Hello, world, I am 4 of 12
Hello, world, I am 8 of 12
Hello, world, I am 3 of 12
Hello, world, I am 9 of 12
Hello, world, I am 11 of 12
Hello, world, I am 7 of 12
Hello, world, I am 6 of 12
Hello, world, I am 2 of 12
Hello, world, I am 10 of 12
```

8.4. Запуск MVAPICH2-задач

Постановка задачи

Запуск MVAPICH2-задач поддерживается в пакетном режиме и в режиме реального времени (интерактивном).

Порядок выполнения

- Запуск выполняется точно так же, как для не-MPI-задачи (Раздел 3, Раздел 4).

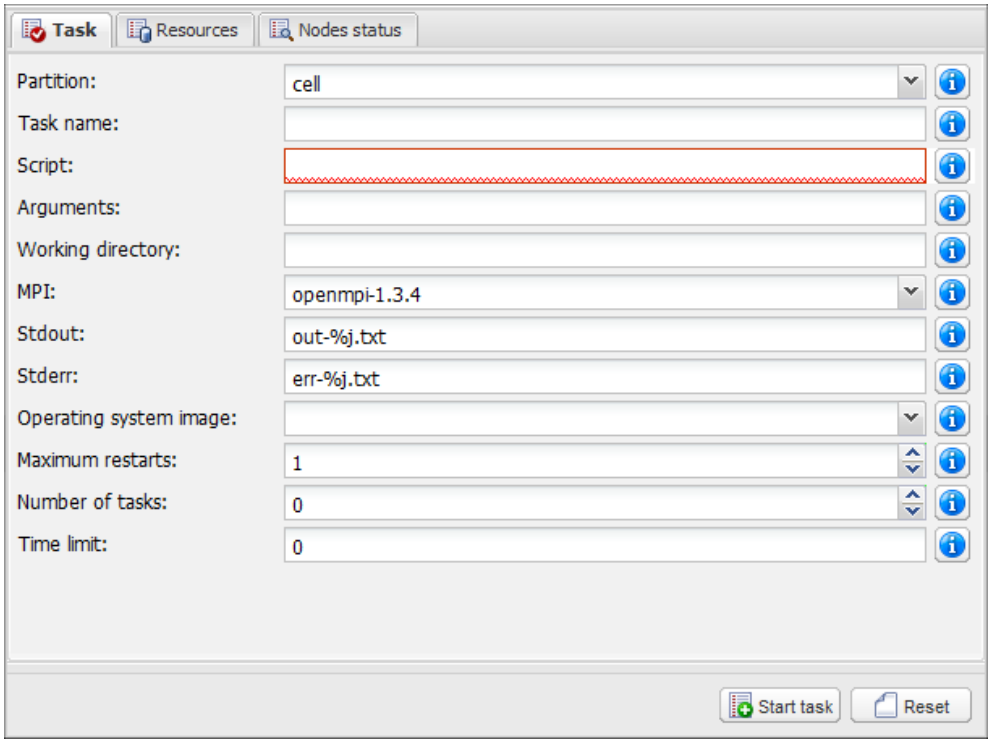
Пример

```
$ srun -n 12 hello.mv2
Hello, world, I am 1 of 12
Hello, world, I am 2 of 12
Hello, world, I am 8 of 12
Hello, world, I am 4 of 12
Hello, world, I am 7 of 12
Hello, world, I am 3 of 12
Hello, world, I am 0 of 12
Hello, world, I am 10 of 12
Hello, world, I am 6 of 12
Hello, world, I am 9 of 12
Hello, world, I am 5 of 12
Hello, world, I am 11 of 12
```

Глава 5. Графический интерфейс управления задачами

1. Запуск задачи на выполнение

При входе в режим **New Task** графического Web-интерфейса (см. Глава 1, Раздел 2) открывается окно для постановки задач в очередь. В центральной его части задаются параметры задачи и выполняется ее запуск на выполнение:



По каждому полю можно получить дополнительную информацию, щелкнув на кнопке справа от него. Для задачи задается (сверху вниз):

- раздел-партиция кластера;
- имя самой задачи;
- скрипт для ее запуска;
- аргументы командной строки;
- рабочий каталог;
- версия MPI;
- стандартные потоки вывода и ошибок;
- образ операционной системы для загрузки на узел;
- максимальное количество перезапусков;

- максимальное количество параллельных процессов (шагов задачи);
- временной лимит выполнения задачи.

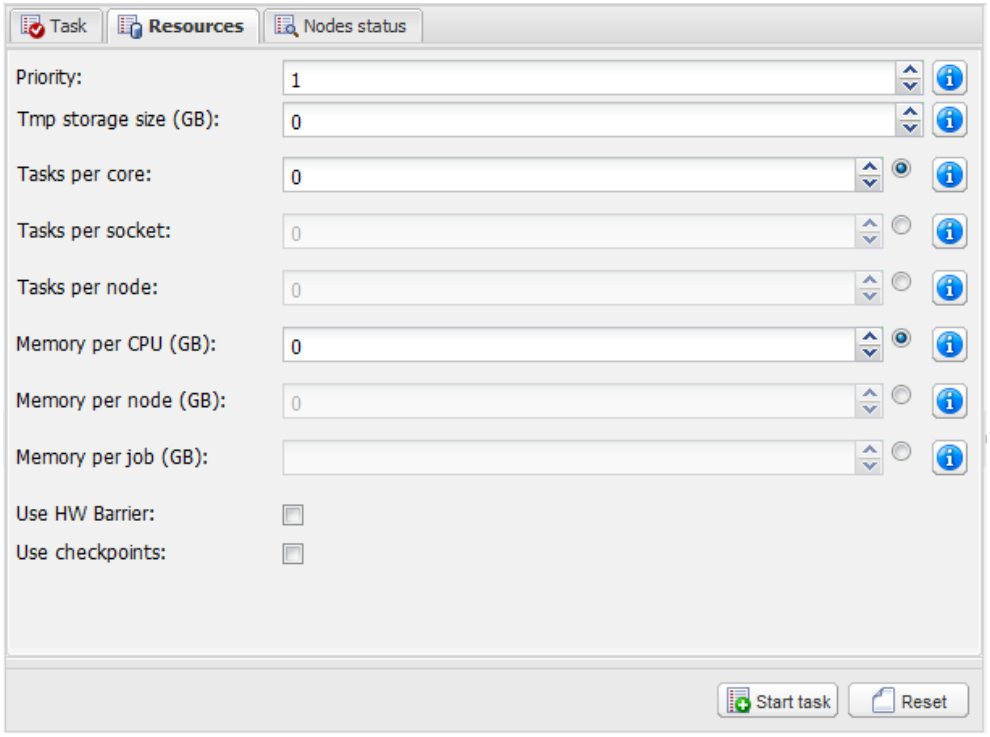
Кнопка **Start task** запускает задачу, а кнопка **Reset** восстанавливает параметры по умолчанию (из профиля).

Следующая вкладка в этой же области окна, **Resources**, позволяет задать лимиты ресурсов для задачи:

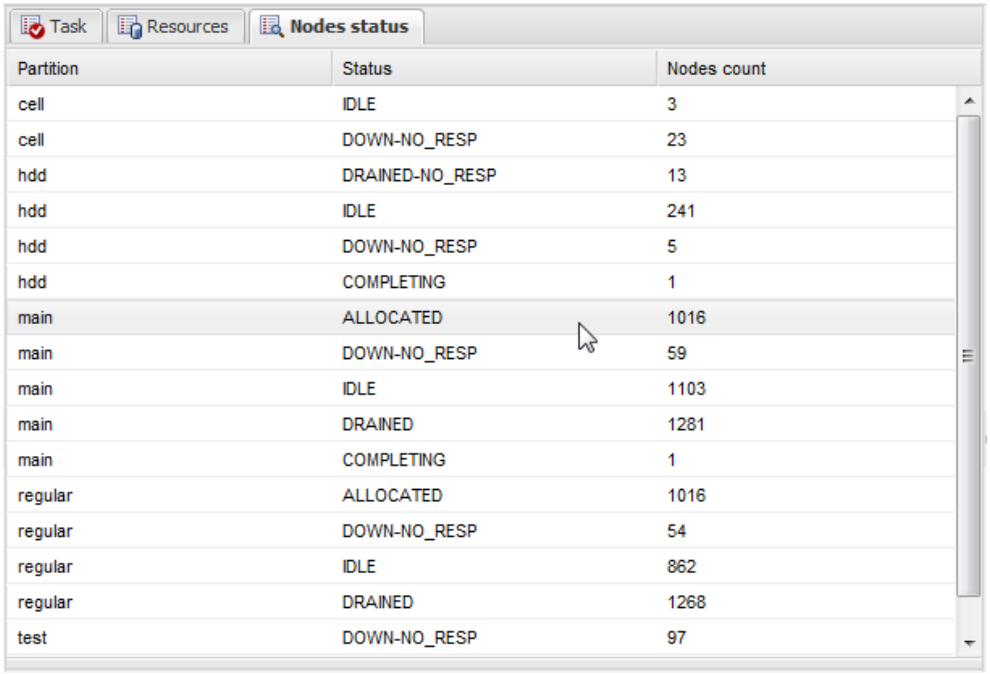
- объем временного хранилища;
- количество задач на ядро, сокет и узел;
- объем памяти на процессор, узел и задачу;
- использование барьерной сети (для синхронизации сохранения данных приложений в промежуточных контрольных точках);

В текущей версии только библиотека OpenMPI из всех версий MPI (см. вкладку **Task**) корректно работает с барьерными сетями.

- использование самих контрольных точек.



Третья вкладка центральной части окна, **Nodes status**, показывает текущее состояние вычислительных узлов кластера согласно номенклатуре SLURM. Узлы упорядочены по разделам и состояниям, указано общее количество узлов в каждом состоянии.



Partition	Status	Nodes count
cell	IDLE	3
cell	DOWN-NO_RESP	23
hdd	DRAINED-NO_RESP	13
hdd	IDLE	241
hdd	DOWN-NO_RESP	5
hdd	COMPLETING	1
main	ALLOCATED	1016
main	DOWN-NO_RESP	59
main	IDLE	1103
main	DRAINED	1281
main	COMPLETING	1
regular	ALLOCATED	1016
regular	DOWN-NO_RESP	54
regular	IDLE	862
regular	DRAINED	1268
test	DOWN-NO_RESP	97

Кроме центральной части, в режиме **New task** имеются вспомогательные средства для выбора параметров и данных задачи.

Левая часть окна – это файловый менеджер (**File manager**) для просмотра каталогов пользователя и управления его файлами (см. Глава 2, Раздел 2). Имена файлов можно перетаскивать отсюда мышью в любые поля интерфейса, где требуется задать файл.

В правой части окна интерфейса (**Profiles**) можно добавить, изменить или удалить профили, то есть наборы параметров задачи, заданных в окне. Системные профили (**System Profiles**) доступны для изменений только администраторам, тогда как пользовательские (**User Profiles**) может создавать и править пользователь. Если щелкнуть на профиле, параметры из него переносятся в поля окна.

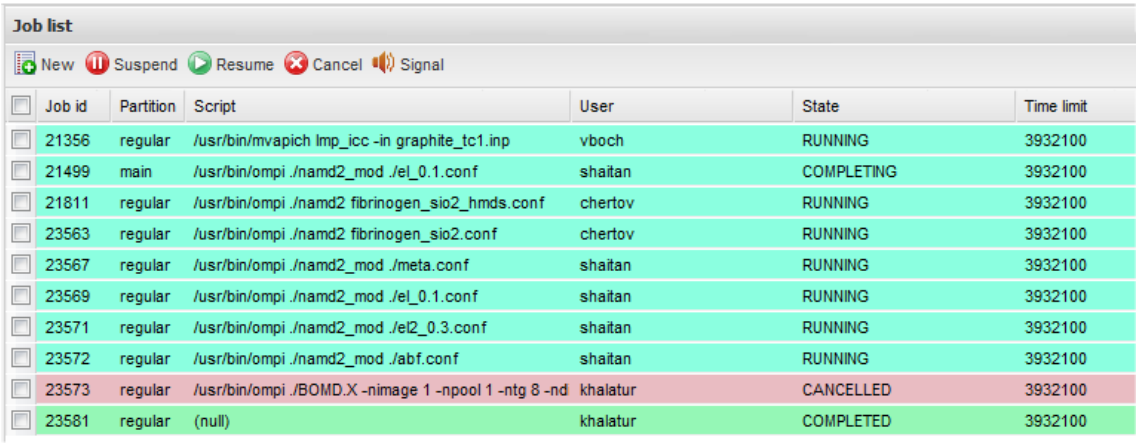
Под центральной частью окна задаются дополнительные параметры окружения задачи. **Prologs** и **Epilogs** — это скрипты для выполнения перед и после задачи. Для задачи можно задать переменные окружения в таблице **Environment**.

Здесь же задаются зависимости (**Dependencies**) между задачами и извещения (**Notifications**) о различных возможных событиях в ходе выполнения задачи. Типы зависимостей должны быть определены в конфигурационной базе данных. Это, например, запуск или приостанов задачи после определенного режима завершения другой задачи, и т.п.

Извещения о завершении задачи или других событиях в настоящей версии не реализованы.

2. GUI — управление очередью задач

Щелкните на кнопке **Queue** на панели навигации в верхнем правом углу окна браузера, где открыт графический Web-интерфейс. Откроется составное окно для слежения за состоянием очереди вычислительных задач и управления ею. В центральной его части находится список задач, находящихся в очереди, и ряд элементов управления для изменения их состояния.



Job id	Partition	Script	User	State	Time limit
21356	regular	/usr/bin/mvapich lmp_icc -in graphite_tc1.inp	vboch	RUNNING	3932100
21499	main	/usr/bin/ompi ./namd2_mod ./el_0.1.conf	shaitan	COMPLETING	3932100
21811	regular	/usr/bin/ompi ./namd2 fibrinogen_sio2_hmds.conf	chertov	RUNNING	3932100
23563	regular	/usr/bin/ompi ./namd2 fibrinogen_sio2.conf	chertov	RUNNING	3932100
23567	regular	/usr/bin/ompi ./namd2_mod ./meta.conf	shaitan	RUNNING	3932100
23569	regular	/usr/bin/ompi ./namd2_mod ./el_0.1.conf	shaitan	RUNNING	3932100
23571	regular	/usr/bin/ompi ./namd2_mod ./el2_0.3.conf	shaitan	RUNNING	3932100
23572	regular	/usr/bin/ompi ./namd2_mod ./abf.conf	shaitan	RUNNING	3932100
23573	regular	/usr/bin/ompi ./BOMD.X -nimage 1 -npool 1 -ntg 8 -nd	khalatur	CANCELLED	3932100
23581	regular	(null)	khalatur	COMPLETED	3932100

Окно видно всем пользователям, но некоторые элементы и операции зависят от уровня полномочий пользователей и могут быть доступны не всем. Так, рядовому пользователю видны только его собственные задачи в списке, а администраторы групп (*groupadmin*-ы) могут видеть задачи всех пользователей группы.

Для каждой задачи в этом списке указан ее идентификатор, раздел кластера, пользователь, состояние выполнения или завершения, лимит времени, причина прекращения ее работы (по номенклатуре SLURM). Кнопка **New** открывает интерфейс запуска новой задачи.

Остальные кнопки над списком предназначены для функций администрирования. В режиме рядового пользователя они не видны.

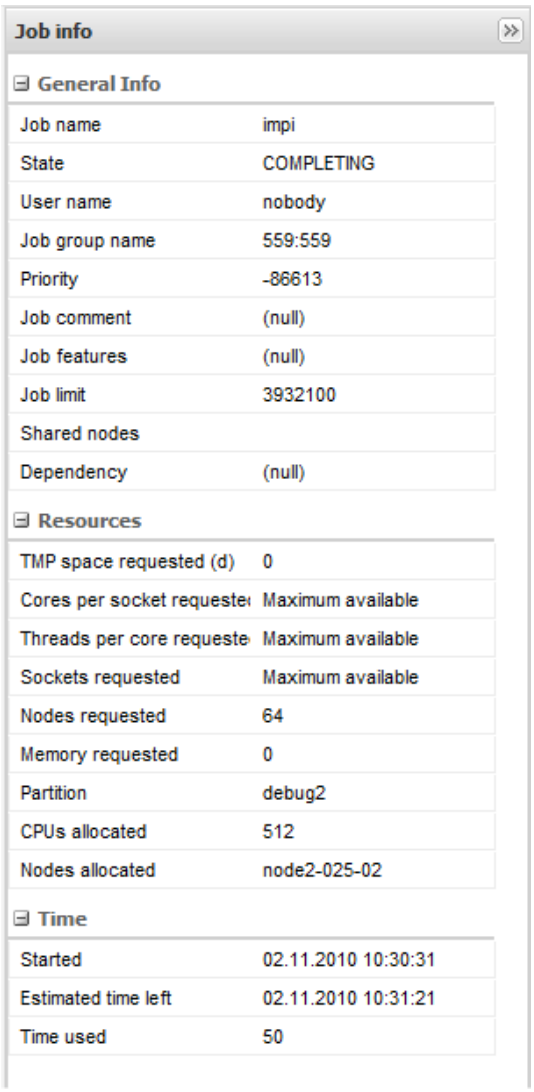
Слева и справа от списка задач находятся вспомогательные панели для просмотра и фильтрования информации о задачах.

Левая панель, **Filters**, позволяет отобрать информацию для центральной области по критериям пользователей, состояний задач, разделов кластера. Для этого достаточно поставить или снять соответствующие «птички».

Список возможных состояний задач:

- **PENDING** – задачи в очереди, еще не запущенные на выполнение;
- **RUNNING** – выполняемые задачи;
- **SUSPENDED** – задачи, приостановленные администратором или владельцем;
- **COMPLETING** – завершающие свое выполнение задачи;
- **COMPLETED** – нормально завершившиеся задачи;
- **CANCELLED**, **TIMEOUT**, **FAILED** – аварийно завершившиеся задачи.

Если выбрать задачу в центральном списке окна интерфейса, справа от списка в области **Job info** (см. рисунок) появится подробная информация об этой задаче. Информация состоит из трех категорий: общая (**General Info**), ресурсы задачи (**Resources**) и время ее выполнения (**Time**).


Job info
General Info
Job name impi
State COMPLETING
User name nobody
Job group name 559:559
Priority -86613
Job comment (null)
Job features (null)
Job limit 3932100
Shared nodes
Dependency (null)
Resources
TMP space requested (d) 0
Cores per socket requeste Maximum available
Threads per core requeste Maximum available
Sockets requested Maximum available
Nodes requested 64
Memory requested 0
Partition debug2
CPUs allocated 512
Nodes allocated node2-025-02
Time
Started 02.11.2010 10:30:31
Estimated time left 02.11.2010 10:31:21
Time used 50

Глава 6. Копирование результатов

Постановка задачи

Вычислительная задача пользователя может выводить данные (результаты расчетов и пр.) в файлы, находящиеся на кластере в домашнем каталоге этого пользователя. Если пользователю необходимо просмотреть эти данные или выполнить их обработку, он может скопировать их к себе на локальную машину несколькими способами.

Порядок выполнения

- С помощью команды-утилиты **scp** (Глава 2, Раздел 1.1) и других средств SSH в среде Linux (Глава 2, Раздел 1.2 или Глава 2, Раздел 1.3).
- С помощью файлового менеджера в Web-интерфейсе Clustrx, режим **New Task** (Глава 2, Раздел 2).
- С помощью оболочек для передачи файлов в среде Windows (Глава 2, Раздел 3).

Список изменений

Дата	Раздел	Описание изменений
05.05.2011	Глава 4, Раздел 8.1	Добавлено задание размеров стека для OpenMPI-задач.
04.04.2011	Глава 2, Раздел 3	Уточнены условия работы в FileZilla при входе на кластер по ключу.
30.03.2011	Глава 4, Раздел 5	Добавлен раздел об управлении распределением процессоров и узлов при запуске задач.
	Глава 1, Раздел 1.2	Добавлены реквизиты входа на кластер через Windows-клиент PuTTY (изменен рис.)
21.03.2011	Дизайн	Изменен дизайн обложки и заголовков
23.02.2011	Глава 3, Раздел 2	Документировано временное решение для обращения к компилятору `mpicc` в неинтерактивном сеансе.
	Информация о документе	Добавлен список изменений и выходные данные документа.

Информация о документе

Разработчик

В. Бродовой

Информационный вклад

А. Бандура, С. Горенко, М. Кузнецов, С. Рябчун, И. Устинов, М. Шигорин

Публикация и права

Massive Solutions Ltd., 2010-2011

info@massivesolutions.co.uk

http://www.massivesolutions.co.uk