

# Руководство пользователя кластера ВЦ ДВО РАН

Версия 5.0  
Мальковский С.И.  
2010 г., 2013 г.  
Версия 4.0  
Сапронов А. Ю.  
28 августа 2005 г.

(при редакции: Шаповалов Т.С. (tsh@list.ru) от 20.02.2008)

## Аннотация

Данное руководство содержит минимально необходимый объем информации для работы на кластере ВЦ ДВО РАН: описание процесса регистрации, сведения по работе в ОС Linux (вход в систему, работа с каталогами и файлами, мониторинг) и работе с MPI программами и не параллельными программами на кластере (компиляция, запуск, остановка, работа с очередями). В тексте под термином параллельная программа подразумеваются только MPI программы.

Команды и переменные командного интерпретатора, названия программ, листинги, непосредственный ввод/вывод консоли выделены моноширинным шрифтом.

Вопросы относительно работы кластера следует отправлять на e-mail: [support@hpc.febras.net](mailto:support@hpc.febras.net).

Вопросы относительно этого документа (ошибки, неточности, предложения) можно отправлять на e-mail: [support@hpc.febras.net](mailto:support@hpc.febras.net).

## Содержание

Аннотация.....	1
1. Регистрация.....	2
2. Вход в систему.....	2
2.1. Вход с Windows-машины.....	2
2.2. Вход с терминала Linux.....	3
3. Копирование файлов.....	3
4. Работа на кластере.....	5
4.1. Навигация.....	5
4.1.1. Консоль.....	5
4.1.2. mc.....	7
4.2. Редактирование файлов.....	7
4.2.1. mc-edit.....	7
4.2.2. Emacs, VIM.....	8
4.3. Компиляция программ.....	8
4.3.1. Замечания по разработке программ на отдельной машине.....	10
4.4. Запуск задач.....	10
4.4.1. Диспетчеризация задач.....	10
4.4.2. Система очередей.....	10
4.4.3. Постановка задачи в очередь.....	11
4.4.4. Запуск интерактивных программ.....	12
4.4.5. Запуск непараллельных программ.....	12
4.4.6. Состояние пользовательских задач.....	13
4.4.6. Остановка задач.....	14
6. Мониторинг.....	14
6.1. Web-интерфейс.....	14
6.2. Консоль.....	14
7. Справочная информация.....	15

## 1. Регистрация

Регистрация пользователей на кластере происходит автоматически – посредством web-интерфейса. Загрузить html страницу с формой для регистрации можно по адресу: <http://hpc.febras.net/cluster/registration.html> или пройти по ссылке с главной страницы сайта кластера: <http://hpc.febras.net>.

В регистрационной форме пользователь должен заполнить ряд обязательных (выделены символом «\*») полей:

**Фамилия, Имя, Отчество** Заполнение разрешено как кириллицей, так и латинскими буквами.

**Email (сообщения кластера)** По этому адресу будут отправляться сообщения связанные с пользовательскими заданиями, а также организационные сообщения связанные с регистрацией пользователя, ошибками в работе и т.п.

**Телефон** Номер контактного телефона.

**Место работы** Полное наименование места работы или учебы.

**Логин** Имя пользователя применяемое для входа на кластер. Разрешено применение только латинских букв и цифр. Минимальный размер этого поля 4 символа.

**Пароль** Желаемый пароль для входа на кластер. Разрешено применение только латинских букв и цифр. Минимальный размер этого поля 8 символов.

**Пароль (еще раз)** То же, что и предыдущее. Служит для контроля корректности ввода пароля.

**Цель регистрации** В этом поле пользователь может оставить дополнительную информацию для администратора, которую он считает важной.

После заполнения формы и нажатия на кнопку «Отправить регистрационную информацию» заявка на регистрацию автоматически будет отправлена администратору кластера. Заявка рассматривается администратором, решение о регистрации согласовывается, в случае необходимости, с руководством. При отсутствии ошибок заполнения формы и принятия положительного решения о регистрации будет создана учетная запись пользователя. На контактный e-mail будет отправлено сообщение, содержащее краткие инструкции для подключения.

## 2. Вход в систему

Для работы с системой пользователь должен иметь свою учетную запись на управляющем узле кластера. Регистрация пользователя на кластере происходит в соответствии с предыдущей частью руководства. После регистрации пользователь получает свое имя (логин), пароль и домашнюю директорию. Если имя пользователя, например, будет `user`, то домашняя папка находится в `/home/user`.

Пользователи имеют возможность работать на кластере с любой машины, находящейся в сети института и интернет. Для входа в систему пользователю необходим адрес сервера ([mercury.febras.net](http://mercury.febras.net)), а также имя и пароль, указанные при регистрации.

### 2.1. Вход с Windows-машины

Работа с системой осуществляется по безопасному протоколу SSH при помощи какого-либо ssh-клиента. Клиент должен поддерживать протокол версии 2. Рекомендуется использовать [PuTTY](http://PuTTY). Эта программа является свободно распространяемой и проста в использовании.

После запуска программы (рис. 2) пользователь должен выбрать протокол ssh и в поле «Host Name (or IP address)» указать адрес сервера. Нажатие на «Open» приведет к отправке запроса на подключение. В случае успешного подключения к серверу будет предложено ввести имя (логин), а затем и пароль.

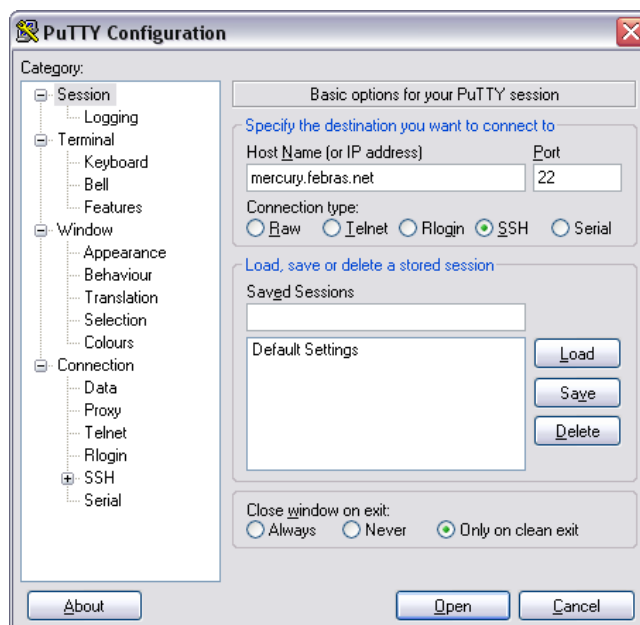


Рис. 2: Окно ssh-клиента PuTTY

При вводе пароля символы на экране не отображаются. Если все введено правильно, то пользователь автоматически окажется в своей домашней директории. Этот каталог доступен пользователю с любого узла кластера.

**Примечание.** На кластере существует единое дисковое пространство для директорий /opt (только чтение) и /home. Все узлы используют дисковый массив сервера посредством сетевой файловой системы NFS. Файл записанный на одном из узлов кластера автоматически становится доступен на любом другом.

Работа в ssh-сессии происходит в терминальном (текстовом, консольном) режиме. Необходимо помнить, что консоль Linux, в отличие от Windows, различает регистр вводимых символов, то есть mydoc.txt и mydoc.TXT не одно и то же. После входа на экране отображается консоль командного интерпретатора в следующем формате имя\_пользователя@машина текущий\_каталог:

```
[user@mercury ~]$
```

## 2.2. Вход с терминала Linux

В любой дистрибутив ОС Linux входит терминальный ssh-клиент (обычно OpenSSH). Минимальный формат команды для подключения к кластеру таков:

```
[user@localhost ~]$ ssh mercury.febras.net -l имя_пользователя
```

## 3. Копирование файлов

При работе пользователю придется копировать файлы со своей машины на кластер (программы, входные данные, библиотеки) и обратно (листинги, результаты расчетов). Сделать это можно через протокол SSH.

Копирование файлов по протоколу SSH является наиболее безопасным при удаленном доступе. Благодаря шифрованию этот протокол обеспечивает должную защиту передаваемой по сети информации. Копирование файлов осуществляется при помощи утилиты `scp` (для windows пользователей – [WinSCP](#)).

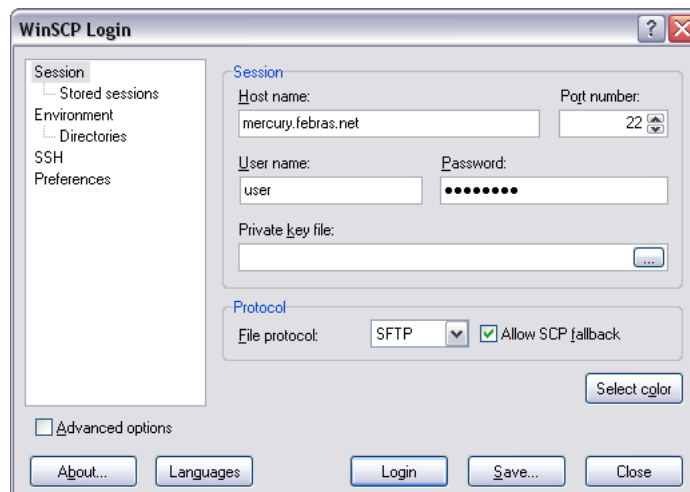
Ниже представлены примеры, иллюстрирующие передачу файла с локального компьютера пользователя в его домашнюю директорию на кластере.

### Примеры:

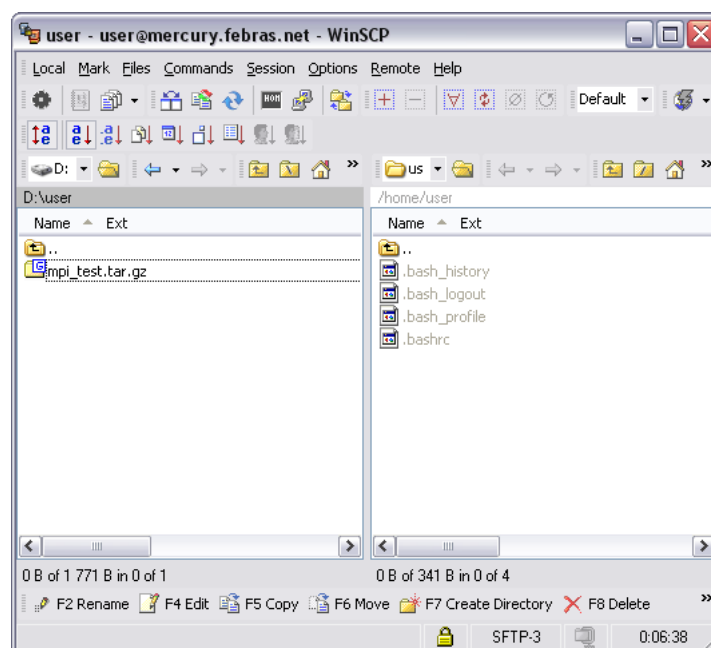
Использование WinSCP в ОС Windows:

1. Запускаем приложение и указываем в соответствующих полях адрес сервера (или его до-

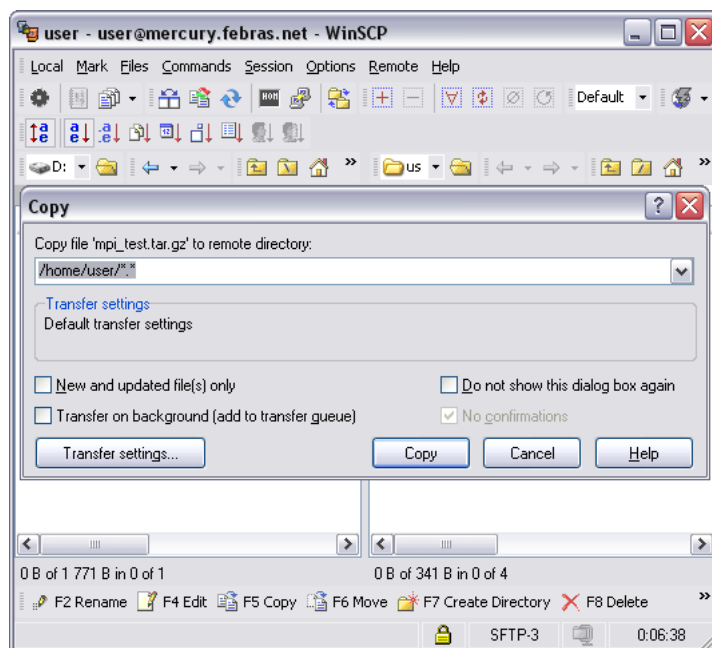
менное имя), имя пользователя и пароль. Чтобы при последующих подключениях не вводить эту информацию снова, сохраняем текущую сессию, нажав на кнопку «Save...». После этого подключаемся к кластеру путем нажатия на кнопку «Login».



2. При первом подключении появится запрос о том, доверяете ли вы удаленному серверу. Для продолжения подключения нужно нажать кнопку «Yes».
3. После подключения откроется окно программы, напоминающее двухпанельный файловый менеджер. В левой панели отображается содержимое одной из директорий вашего компьютера, в правой — содержимое директории кластера (по умолчанию — вашей домашней директории на кластере).



4. В правой панели перейдите в директорию с файлами, которые необходимо передать на кластер. В левой укажите директорию на кластере, в которую необходимо скопировать файлы. Выделите в левой панели файлы для передачи и нажмите клавишу «F5». Появится запрос на копирование, который необходимо принять, нажав кнопку «Сору».



5. После завершения копирования вы увидите переданные файлы в правой панели.
6. Копирование файлов с кластера на локальный компьютер осуществляется аналогично.

Использование команды `scp` в ОС Linux:

Рассмотрим синтаксис команды, для копирования файла из текущей директории в удаленную директорию кластера:

```
scp ./mpi_test.tar.gz user@mercury.febras.net:/home/user
```

Первым обязательным параметром этой команды является копируемый файл (папка) – источник. В показанном примере это `./mpi_test.tar.gz`. Строка `user@mercury.febras.net:/home/user` – адрес приемника. Формат адреса: **имя\_пользователя@адрес\_узла\_назначения:файл\_или\_папка\_приемник**.

Для того, чтобы скопировать локальную директорию с поддиректориями на кластер, воспользуйтесь следующей командой:

```
scp -r ./mpi_test user@mercury.febras.net:/home/user
```

Ключ `-r` указывает, что мы хотим скопировать не отдельный файл, а директорию с поддиректориями целиком. Остальные параметры команды остались прежними.

Копирование файлов с кластера осуществляется аналогично.

## 4. Работа на кластере

Для работы на кластере удобно пользоваться файловым менеджером `mc`, который включает редактор текстовых файлов. Универсальным способом работы на кластере является работа через консоль. Оба способа кратко описаны ниже.

### 4.1. Навигация

#### 4.1.1. Консоль

На кластере пользователю доступен на запись только его домашний каталог. Для работы с файлами и каталогами существует несколько полезных команд (Табл. 1):

Таблица 1: Некоторые полезные команды

Команда	Описание
---------	----------

<b>ls</b>	Показать список папок и файлов в текущем каталоге
<b>pwd</b>	Показать имя текущего каталога
<b>cd</b> название_каталога	Сменить текущий каталог
<b>mkdir</b> название_каталога	Создать каталог
<b>clear</b>	Очистка экрана
<b>cp</b> что куда	Копирование файлов и каталогов
<b>tar</b>	Работа с архивами

Подробную справку по этим командам, а также по всем другим можно получить, набрав в консоли:

```
man имя_команды
```

Более расширенную информацию по большинству программ можно получить, набрав:

```
info имя_команды
```

При работе в консоли удобно пользоваться клавишей автоподстановки **Tab**. Если набрав несколько первых символов программы или файла пользователь нажмет на клавишу **Tab**, то система дополнит их до полного имени, если имеется только единственный вариант дополнения. Если вариантов несколько, то появится их список. Введение дополнительных символов, которые однозначно определяют нужное имя и нажатие клавиши **Tab** приводит к выбору нужного имени файла или команды.

#### Примеры:

Определение в какой папке находится пользователь.

```
[user@mercury etc]$ pwd
/etc
```

Переход в домашний каталог.

```
[user@mercury etc]$ cd /home/user
```

Создание папки **/home/user/test**

```
[user@mercury ~]$ mkdir /home/user/test
```

Копирование файла **mpi\_test.tar.gz** из текущей (домашней) директории в только что созданную. Переход в эту директорию.

```
[user@mercury ~]$ cp ./mpi_test.tar.gz /home/user/test/
[user@mercury ~]$ cd /home/user/test
```

Распаковка архива, формата **tar.gz**:

```
[user@mercury test]$ tar -xzf ./mpi_test.tar.gz
```

Просмотр содержимого директории:

```
[user@mercury test]$ ls
mpi_test mpi_test.tar.gz
[user@mercury test]$ ls ./mpi_test
test.c test.cpp test.f test.f90
```

### 4.1.2. mc

В работе на кластере удобно использовать файловый менеджер Midnight Commander (mc) (Рис. 3). mc очень похож на FAR и Norton Commander.

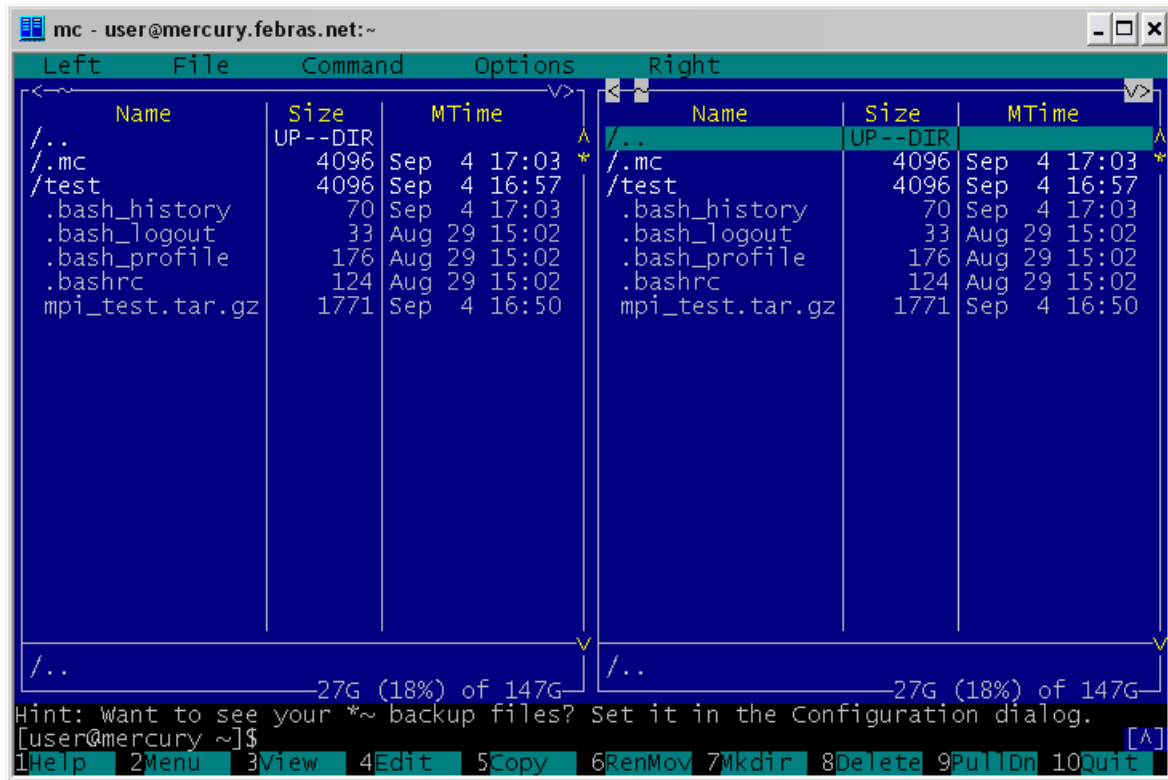


Рис. 3: Файловый менеджер mc

Этот ФМ можно запустить из консоли:

```
[user@mercury ~]$ mc -ca
```

Ключ `-ca` необходим для корректного отображения специальных символов в ssh-сессии.

## 4.2. Редактирование файлов

### 4.2.1. mc-edit

Пользователю рекомендуется основную часть исходных файлов подготавливать на своей локальной машине в привычных текстовых редакторах. Для внесения небольших изменений в файлы очень хорошо подходит встроенный в mc редактор.

Для запуска этого редактора необходимо установить курсор на интересующий файл и нажать клавишу F4 (Рис. 4). Для создания и редактирования нового файла — «Shift-F4».

```

mc - user@mercury.febras.net:~/test/mpi_test/mpi_test.qsub
mpi_test.qsub [-M--] 6 L:[ 1+12 13/ 14] *(153 / 154b)= . 10 0x0A
#PBS -k oe
#PBS -l nodes=2:ppn=16
#PBS -r n
#PBS -m abe
#PBS -q vl_mercury
#PBS -N mpi_test
#!/bin/sh

cd /home/user/test/mpi_test
mpiexec ./mpi
exit 0_

```

Рис. 4: Встроенный в mc редактор

В этом редакторе присутствуют все основные средства, имеющиеся в подобных программах: копирование, вставка, поиск, замена, подсветка синтаксиса, ряд других.

#### 4.2.2. Emacs, VIM

Если необходимы более сложные средства редактирования, то можно использовать редакторы vi/vim и emacs. Оба этих редактора имеют более чем тридцатилетнюю историю и стали культовыми в мире UNIX. vim является более простым (по функциональности), но менее наглядным редактором. Основная функция vim — работа с исходными текстами программ. emacs — это больше чем редактор. С его помощью можно осуществлять профессиональную верстку; набор, запуск и отладку программ; прием/отправление электронной почты и новостей; навигацию по жесткому диску; использование в качестве ежедневника и много другое.

Недостаток этих редакторов — достаточно сложное управление (особенно vim), которое нельзя назвать интуитивно понятным пользователям, привыкшим к работе в ОС Windows. За полным перечнем доступных команд редакторов рекомендуется обратиться к соответствующим руководствам.

### 4.3 Компиляция программ

На кластере (на 15.12.2013) поддерживаются следующие компиляторы языков программирования для архитектуры x86\_64:

Таблица 2: Компиляторы на кластере

Компилятор	Путь к файлу компилятора	Язык
Intel C 14.0.1	/opt/intel/composer_xe_2013_sp1.1.106/bin/intel64/icc	C
Intel C++ 14.0.1	/opt/intel/composer_xe_2013_sp1.1.106/bin/intel64/icpc	C++
Intel Fortran 14.0.1	/opt/intel/composer_xe_2013_sp1.1.106/bin/intel64/fort	Fortran 77/90
GNU C 4.1.2	/usr/bin/gcc	C
GNU C++ 4.1.2	/usr/bin/g++	C++
GNU Fortran 4.1.2	/usr/bin/gfortran	Fortran 90



GNU Fortran 3.4.6	/usr/bin/g77	Fortran 77
-------------------	--------------	------------

В переменной среды «PATH» для всех пользователей прописаны пути к компиляторам Intel для архитектуры x86\_64. Поэтому, например, при вводе команды:

```
[user@mercury ~]$ icc
```

будет запускаться 64-битная версия компилятора Intel C. В переменной среды «LD\_LIBRARY\_PATH» также указаны пути к библиотекам с архитектурой x86\_64.

В качестве реализации MPI библиотеки на кластере (на 15.12.2013) поддерживается Intel MPI. Библиотека доступна со всех узлов кластера и расположена по пути /opt/intel/impi/4.1.2. Пути к 64-битной версии данной библиотеки добавлены в переменные среды «PATH» и «LD\_LIBRARY\_PATH».

Для компиляции mpi программ лучше всего использовать обёртки к компиляторам, чем вручную прописывать для этого специальные флаги. Так, например, чтобы скомпилировать mpi программу, написанную на языке Fortran, нужно воспользоваться оберткой mpiifort. Данная команда вызовет компилятор Intel Fortran, с указанием всех необходимых флагов.

Распишем соответствия между обертками и соответствующими им компиляторами: mpiicc – icc, mpiicsc – icsc, mpiifort – ifort, mpigcc – gcc, mpigxx – g++, mpif77 – g77, mpif90 – gfortran.

Для того, чтобы посмотреть какие опции компилятора указываются при вызове обертки, можно воспользоваться следующей командой:

```
[user@mercury ~]$ mpiicc -show
icc -I/opt/intel/impi/4.1.2.040/intel64/include
-L/opt/intel/impi/4.1.2.040/intel64/lib -Xlinker -enable-new-dtags -Xlinker
-rpath -Xlinker /opt/intel/impi/4.1.2.040/intel64/lib -Xlinker -rpath
-Xlinker /opt/intel/mpi-rt/4.1 -lmpigf -lmpi -lmpigi -ldl -lrt -lpthread
```

Как видно из данного вывода, единственными опциями, которые может потребоваться указать при вызове компилятора могут оказаться опции оптимизации.

Рассмотрим пример, иллюстрирующий компиляцию mpi приложения:

```
[user@mercury mpi_test]$ mpiifort test.f90 -o ./mpi -O2 -funroll-loops
-march=core2 -mmmx
```

В данном примере, учитывая специфику научных приложений, особое внимание уделено параметрам, отвечающим за оптимизацию вычислений с плавающей точкой.

Таблица 3: Некоторые параметры компиляторов

Параметр	Описание
-O3	Включение полной автоматической оптимизации компилятора (рекомендуемое -O2)
-funroll-loops	Включение раскрутки циклов
-march=core2	Этот параметр указывает компилятору использовать команды доступные начиная с Core 2
-mmmx	Разрешение использования набора инструкций mmx
-o имя_файла	Имя выходного файла. Если этот параметр не указан, то по умолчанию готовая к выполнению программа будет называться a.out

Обычно эти параметры оптимизации позволяют программе работать существенно быстрее.

Для продуктивной работы настоятельно рекомендуется ознакомиться с полным перечнем команд компиляторов.

### 4.3.1 Замечания по разработке программ на отдельной машине

Практически все реализации MPI поддерживают запуск параллельных приложений в режиме эмуляции на отдельно взятой рабочей станции. Это можно делать как на Linux, так и Windows машинах.

В Linux рекомендуется использовать пакет OpenMPI, а для создания MPI приложений на Windows машинах можно использовать пакет MPICH в версии для Windows. Для успешного портирования программ с Windows на Linux не следует использовать расширения предоставляемые средами программирования, такими как Visual Studio и Borland Builder.

Подготовленные исходные коды программ лучше всего компилировать на кластере.

## 4.4 Запуск задач

### 4.4.1 Диспетчеризация задач

Для диспетчеризации задач на кластере используется система PBS Torque. С её помощью пользователь может отправлять свои задачи на исполнение, снимать их с исполнения и получать информацию по текущему статусу задачи.

Данная система построена на основе очередей, где под очередь понимается набор пользовательских процессов (программ, задач) выполняющихся в рамках системы диспетчеризации. Каждой очереди сопоставлен ряд атрибутов, в зависимости от которых к задаче будут применены те или иные действия. Типичными атрибутами являются название (идентификатор) очереди, её приоритет, доступные ресурсы, количество задач. В общем случае термин очередь не означает, то что программы в ней будут выполняться строго последовательно.

Чтобы поставить задачу на исполнение, пользователь должен добавить ее при помощи команды `qsub` в какую-либо очередь. Очереди отличаются друг от друга совокупностью ресурсов, которыми они обладают.

### 4.4.2 Система очередей

На данный момент действуют 3 очереди: `vl_mars`, `vl_mercury`, `vl_pluto`. В таблице 4 представлены отличия между очередями.

Таблица 4: Очереди на кластере

	<code>vl_mars</code>	<code>vl_mercury</code>	<code>vl_pluto</code>
Доступно узлов	4	5	8
Процессоры, установленные на узлах	<i>Six Core AMD Opteron™ 8431</i>	<i>Quad Core Intel® Xeon® E5450 EM64T</i>	<i>Dual Core Intel® Xeon® 5060 EM64T</i>
Процессоров на узле	4	2	2
Вычислительных ядер на узле	24	8	4
Памяти на узле	96GB	16GB	4GB
Сеть передачи данных	InfiniBand	InfiniBand	Gigabit Ethernet

Никаких ограничений по времени исполнения и количеству доступных ресурсов для этих очередей не установлено.

Для получения информации об очередях, можно выполнить команду `qstat -q`.

```
[user@mercury ~]$ qstat -q
server: mercury1
Queue          Memory CPU Time Walltime Node  Run Que Lm  State
-----
```

vl_mars	--	--	--	--	9	1	--	E	R
vl_pluto	--	--	--	--	5	0	--	E	R
vl_mercury	--	--	--	--	4	0	--	E	R
					-----	-----			
					18	1			

**Queue** – имя очереди; **Run** – число выполняемых задач; **Que** – число задач, ожидающих начала выполнения.

Команда `qstat -Qf имя_очереди` позволяет получить информацию о конкретной очереди.

#### 4.4.3 Постановка задачи в очередь

Для постановки задачи в очередь на исполнение используется команда `qsub`. Данная команда принимает в качестве параметра имя скрипта, в котором описываются требуемые задачей ресурсы и указываются команды, исполняемые при запуске. Рассмотрим пример, иллюстрирующий запуск ранее скомпилированной программы на 2 узлах кластера, с использованием 8 процессоров на каждом.

```
[user@mercury mpi_test]$ cat mpi_test.qsub
#PBS -k oe
#PBS -l nodes=2:ppn=8
#PBS -r n
#PBS -M user@mail.com
#PBS -m abe
#PBS -q vl_mercury
#PBS -N mpi_test
#!/bin/sh

cd /home/user/test/mpi_test

mpirun ./mpi

exit 0
[user@mercury mpi_test]$ qsub mpi_test.qsub
66330.mercury1
```

Если команда выполнена успешно, то на экране отобразится идентификатор задачи (в данном случае это 66330.mercury1), в противном случае появится сообщение об ошибке. Ошибки пользовательской программы (неправильная компиляция и т.п.) проявятся только при переходе задачи к активному состоянию.

**Примечание.** Весь вывод программы в стандартный поток и в поток ошибок перенаправляется в файлы, находящиеся в домашней директории пользователя. Названия таких файлов имеют формат `имя_задачи.(e/o)порядковый_номер`. Для запущенной задачи это будут: `mpi_test.e66330` – для потока ошибок и `mpi_test.o66330` – для стандартного потока вывода.

Прокомментируем каждую из строчек скрипта `mpi_test.qsub`.

Указание сброса потока вывода (o) и потока ошибок (e):

```
#PBS -k oe
```

Требуемое количество узлов (2) и процессоров на каждом из них (8):

```
#PBS -l nodes=2:ppn=8
```

Является ли задача перезапускаемой (задачей с контрольными точками); y — является, n — не является:

```
#PBS -r n
```

Почтовый адрес пользователя:

```
#PBS -M user@mail.com
```

Какие сообщения отправляются на указанный адрес (a — ошибка в выполнении задачи, b — начало выполнения, e — завершение выполнения):

```
#PBS -m abe
```

Идентификатор очереди:

```
#PBS -q vl_mercury
```

Название задачи:

```
#PBS -N mpi_test
```

Указание необходимого командного интерпретатора:

```
#!/bin/sh
```

Переход в директорию с исполняемым файлом:

```
cd /home/user/test/mpi_test
```

Запуск приложения:

```
mpirun ./mpi
```

Выход:

```
exit 0
```

#### 4.4.4 Запуск интерактивных программ

Программы, использующие стандартный ввод, называются интерактивными. Как правило, такие программы после запуска требуют от пользователя ввода данных. При постановке задачи в очередь любая программа переводится в фоновый режим. В этом режиме ввод данных пользователем в запущенную программу невозможен. Для передачи данных таким программам используется механизм перенаправления стандартных потоков ввода/вывода.

Для перенаправления подготавливается текстовый файл, содержимое которого в точности представляет собой данные, вводимые пользователем.

Например, если программа `solver` предполагает ввод в первой строке размерности матрицы, а во второй количества итераций, то текстовый файл `input.txt` будет иметь вид:

```
[user@mercury solver]$ cat input.txt
10000000
1000
```

После каждого числа обязателен символ новой строки. Запуск программы на выполнение производится так:

```
solver < input.txt
```

Скрипт для постановки в очередь задания, в рамках которого будет выполняться интерактивная программа, будет выглядеть следующим образом:

```
[user@mercury solver]$ cat job.qsub
#PBS -k oe
#PBS -l nodes=2:ppn=8
#PBS -r n
#PBS -M user@mail.com
#PBS -m abe
#PBS -q vl_mercury
#PBS -N solver
#!/bin/sh

cd /home/user/test/solver

mpirun ./solver < ./input.txt

exit 0
```

#### 4.4.5 Запуск непараллельных программ

Запуск непараллельных программ практически ничем не отличается от запуска параллельных программ. Единственное отличие заключается в том, что в `qsub` скрипте такой программы необходимо

указать, что для её работы необходим только один логический процессор:

```
#PBS -l nodes=1:ppn=1
```

Также в этом скрипте необходимо запускать непосредственно исполняемый файл программы, то есть не использовать для запуска `mpirun`.

Вот как будет выглядеть `qsub` скрипт для запуска непараллельной программы:

```
[user@mercury test]$ cat job.qsub
#PBS -k oe
#PBS -l nodes=1:ppn=1
#PBS -r n
#PBS -M user@mail.com
#PBS -m abe
#PBS -q vl_mercury
#PBS -N sample_job
#!/bin/sh

cd /home/user/test

./sample

exit 0
```

#### 4.4.6 Состояние пользовательских задач

Для получения информации об очередях и задачах пользователя используется команда `qstat`. Выполнение этой команды без параметров покажет все задачи пользователя и их состояние.

```
[user@mercury ~]$ qstat
Job id          Name          User          Time Use S Queue
-----
700.mercury1   mpi_test      user          00:10:40 R vl_pluto
701.mercury1   sample_job    user          0 Q vl_mars
702.mercury1   solver        user          0 Q vl_mercury
```

**Job id** — идентификатор задачи, полученный при выполнении `qsub`; **Name** — имя задачи; **User** — имя пользователя, запустившего задачу; **Time Use** — процессорное время, потраченное задачей; **S (State)** — состояние задачи (**R** – задача выполняется, **Q** – ожидает в очереди); **Queue** — очередь.

В данном случае пользователю `user` принадлежат три задачи, одна из которых (700.mercury1) выполняется в очереди `vl_pluto`, а две других ожидают в очередях `vl_mars` и `vl_mercury`.

С помощью команды `qstat -n идентификатор_задачи` можно получить список узлов, на которых выполняется конкретная задача. Эта информация полезна при мониторинге эффективности использования вычислительных ресурсов с использованием системы Ganglia, так как позволяет отслеживать состояние только используемых задачей узлов.

```
[user@mercury ~]$ qstat -n 701
mercury.febras.net:
Job ID          Username Queue      Jobname      SessID NDS  TSK  Req'd Req'd  Elap
-----
701.mercury1   user      vl_mars   sample_job  9456    1  --   --   --  R 166:1
mars1/11+mars1/10+mars1/9+mars1/8+mars1/7+mars1/6+mars1/5+mars1/4+mars1/3
+mars1/2+mars1/1+mars1/0
```

Для получения более подробной информации о конкретной задаче можно запустить команду `qstat -f идентификатор_задачи`.

#### 4.4.6 Остановка задач

Остановка программы производится командой `qdel идентификатор_задачи`:

```
[user@mercury ~]$ qdel 700
```

Этой командой задача, стоящая в очереди, убирается из нее, а выполняющаяся задача снимается с выполнения. Следующая по очереди и приоритету задача встает на выполнение.

Задача снимается в течении некоторого времени, поэтому при вызове `qstat` непосредственно после `qdel` удаленная задача все еще может быть отражена в таблице.

## 6. Мониторинг

### 6.1 Web-интерфейс

Мониторинг кластера реализован при помощи системы Ganglia. Эта система позволяет следить за ресурсами кластера посредством web-интерфейса.

Система мониторинга находится по адресу <http://mercury.febras.net/ganglia> (или по ссылке с главной страницы сайта кластера).

Для мониторинга пользователю доступно большое число типов ресурсов: загруженность процессора, оперативная память, загрузка сети, средняя загрузка, количество процессов и ряд других. Имеется возможность наблюдать как за всеми узлами в кластере (по одному параметру), так и за каждым (по всем параметрам).

### 6.2 Консоль

Кроме графического интерфейса существует несколько полезных консольных команд для мониторинга.

Команда `pbsnodes имя_узла` позволяет получить информацию о конкретном узле: тип, состояние, количество процессоров, выполняющиеся задачи. Ниже представлен фрагмент вывода этой команды.

```
[user@mercury ~]$ pbsnodes mercury2
mercury2
  state = job-exclusive
  np = 8
  properties = mercury
  ntype = cluster
  jobs = 0/69079.mercury1, 1/69079.mercury1, 2/69079.mercury1,
3/69079.mercury1, 4/69079.mercury1, 5/69079.mercury1, 6/69079.mercury1,
7/69079.mercury1
  status = opsys=linux,uname=Linux mercury2 2.6.18-128.el5xen #1 SMP Wed Jan
21 11:12:42 EST 2009 x86_64,sessions=30274 30299 30300 30301 30302 30303 30304
30305
30306,nsessions=9,nusers=1,idletime=6750117,totmem=34966476kb,availmem=32764152
kb,physmem=16090112kb,ncpus=8,loadave=8.07,netload=1610728519638,state=free,job
s=69079.mercury1,varattr=,rectime=1285741337
```

**state** – состояние узла (**job-exclusive** – все ресурсы узла заняты; **free** – на узле есть свободные ресурсы для запуска заданий; **offline** – узел временно выведен из эксплуатации, запуск заданий на нем невозможен; **down** – узел выключен); **np** – число процессорных ядер на узле; **jobs** – задачи, запущенные на узле.

При выполнении команды `pbsnodes` без указания параметров будет выведена информация обо всех узлах кластера.

Для получения списка свободных узлов (или узлов с каким-либо другим состоянием) можно воспользоваться командой `pbsnodes -l free` (здесь **free** – состояние узла).

```
[user@mercury ~]$ pbsnodes -l free
mercury1          free
mercury4          free
mercury5          free
mars3             free
```

## 7. Справочная информация

Описание основных команд при работе в ОС Linux – [http://www.info.jinr.ru/unixinfo/pc/lin\\_os.html](http://www.info.jinr.ru/unixinfo/pc/lin_os.html)

Документация к системе диспетчеризации заданий PBS Torque — <http://www.clusterresources.com/torquedocs21/>