

# Инструкция по работе с многопроцессорной техникой

Алексей Сальников

11 февраля 2011 г.

## 1 Введение

Обычно, в тот момент, когда пользователь подаёт заявку на использование многопроцессорной техники, ему дают доступ в вычислительный центр, где собственно находится многопроцессорная техника. Вычислительный центр состоит из многих компонент. Схематически вычислительный центр изображён на рисунке 1. Далее описаны основные компоненты вычислительного центра.

- **Вычислительный система** — объединённый некоторой специально организованной коммуникационной средой набор вычислительных узлов. Вся эта совокупность узлов выполняет задачи пользователя в порядке очереди и представляется как единая система, управляемая с помощью специального компьютера или нескольких компьютеров. (Это определение, вообще говоря, верно только для кластеров.)
- **Узел вычислительной системы** — отдельный компьютер, который работает под управлением своей собственной операционной системы и который объединён сетью с другими узлами. На узлах вычислительной системы происходит расчёт задач пользователя.
- **Файловое хранилище** — подсистема (возможно, целая сеть компьютеров), предназначенная для хранения данных (результаты экспериментов, вычислений, резервные копии и т. п.)
- **Интерфейсная машина (front end)** — машина, предназначенная для взаимодействия системы и удалённых пользователей. На неё осуществляется удалённый доступ, на ней пользователь имеет доступ к файловому хранилищу и с неё пользователь осуществляет запуск заданий на вычислительной системе.
- **Управляющий сервер** — сервер, контролирующий работу вычислительной системы. Вообще говоря, обычно он недоступен пользователю, однако через него происходит управление очередью на вычислительной системе и включение/отключение отдельных узлов и всей системы в целом.

Вычислительный центр может состоять сразу из нескольких вычислительных систем. Например, в НИВЦ МГУ доступны сразу вычислительные системы «Чёбышев МГУ» и Ломоносов МГУ. Обсудим для начала вопросы, связанные с удалённым доступом на интерфейсную машину в вычислительном центре.

Для работы будут доступны следующие вычислительные системы:

- Regatta — IBM pSeries 690. 16-процессорная машина с 64 Гб оперативной памяти. Подробная информация о машине доступна на сайте <http://www.regatta.cs.msu.su/>

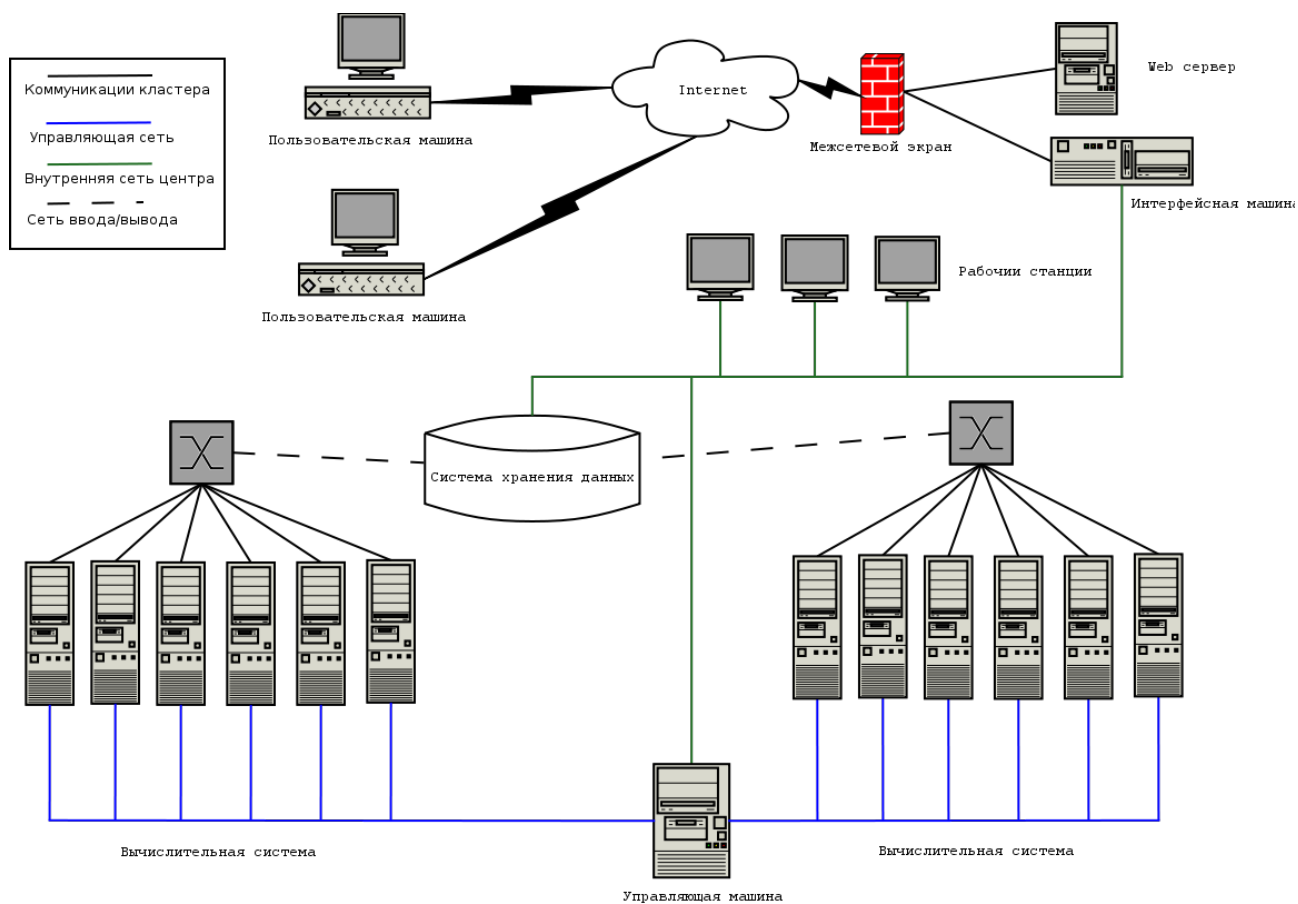


Рис. 1. Структура вычислительного центра

[p690.htm](#). В текущий момент на машине установлена операционная система Linux. Адрес интерфейсной машины [regatta.cs.msu.su](#). Адрес технической поддержки [hpcadm@cs.msu.su](mailto:hpcadm@cs.msu.su).

- BlueGene — BlueGene P. 2048-процессорная установка. Каждый процессор имеет четыре ядра. Подробная информация о машине доступна на сайте <http://hpc.cs.msu.su/bgp>. Для доступа на машину в текущий момент необходимо получать OpenVPN сертификат. Адрес интерфейсной машины [bluegene1.hpc](#) или явно по IP адресу 10.6.7.50. Адрес технической поддержки [hpcadm@cs.msu.su](mailto:hpcadm@cs.msu.su).
- Чебышёв МГУ — Набор узлов на процессорах Intel. Структура довольно сложная, на один узел доступно 8 ядер. Всего узлов доступно 1250. Подробная информация о машине доступна на сайте [http://parallel.ru/cluster/skif\\_msu.html](http://parallel.ru/cluster/skif_msu.html). Авторизация пользователей производится строго по SSH-ключам. Адрес интерфейсной машины [skif-mgu.parallel.ru](#). Адрес технической поддержки [support@parallel.ru](mailto:support@parallel.ru).

## 2 Удалённый доступ

Как правило, интерфейсная машина является UNIX-сервером. В UNIX сообществе принято, что удалённый доступ на машину производится по протоколу SSH. После установления SSH соединения пользователь получает терминал на удалённой машине в котором запущен SHELL. Подробно обсуждать команды SHELL и особенности использования терминалов UNIX, а также структуру файловой системы UNIX в настоящем руководстве не предполагается.

Для обеспечения дополнительных мер безопасности для попадания во внутреннюю сеть от внешних пользователей требуют создания специального зашифрованного туннелированного соединения, которое может быть осуществлено с помощью VPN (Virtual Private Network). Остановимся для начала на использовании VPN.

## 2.1 Создание VPN-соединения на факультет ВМК МГУ

Информация о том, как пользоваться компьютерной сетью факультета ВМК, доступна на сайте <http://wiki.cs.msu.su/Main/WebHome>. В частности, по поводу VPN можно прочитать здесь <http://wiki.cs.msu.su/Main/OpenVPN>. Факультет ВМК предполагает, что пользователи будут использовать OpenVPN для организации VPN-соединения. Получить VPN-сертификат можно, оставив заявку на сайте <http://support.cs.msu.su>. Предположим, что системные администраторы факультета выдали VPN-сертификат, в данном случае он будет выглядеть как архив с некоторым набором файлов, среди которых должен быть файл с именем `cmc.ovpn`. Этот файл является конфигурацией клиентской части OpenVPN.

В случае работы из под Windows необходимо скачать OpenVPN клиент с сайта <http://openvpn.net/index.php/open-source/downloads.html>. После установки данной программы следует воспользоваться инструкциями с сайта факультета <http://wiki.cs.msu.su/Main/OpenVPNConfigWindows>.

В случае с UNIX системами необходимо поставить пакет в операционной системе под названием `openvpn`. После успешной его установки необходимо распаковать выданный системными администраторами архив в какой-либо каталог. После этого вы заходите в указанный каталог и запускаете в нём с правами пользователя `root` команду `openvpn`, указывая параметры: `openvpn --config cmc.ovpn`. Если всё будет хорошо, то у вас запустится `openvpn`-клиент, который займёт собой консоль и создаст новый сетевой интерфейс, в Linux он называется `tun0`. Таблица маршрутизации должна при этом выглядеть следующим образом:

```
salnikov@ice:~/openvpn$ route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
10.20.100.86 * 255.255.255.255 UH 0 0 0 tun0
vpn.cmc.msu.ru angel.seminar 255.255.255.255 UGH 0 0 0 eth1
10.20.0.1 10.20.100.86 255.255.255.255 UGH 0 0 0 tun0
10.6.6.96 10.20.100.86 255.255.255.240 UG 0 0 0 tun0
10.6.7.0 10.20.100.86 255.255.255.0 UG 0 0 0 tun0
10.5.55.0 * 255.255.255.0 U 0 0 0 eth1
10.3.0.0 10.20.100.86 255.255.0.0 UG 0 0 0 tun0
default 10.20.100.86 128.0.0.0 UG 0 0 0 tun0
128.0.0.0 10.20.100.86 128.0.0.0 UG 0 0 0 tun0
default angel.seminar 0.0.0.0 UG 0 0 0 eth1
```

После установки `openvpn`-соединения появится возможность зайти на интерфейсную машину по протоколу SSH и тем самым получить терминальный доступ к вычислительной системе.

## 2.2 Доступ по протоколу SSH

В большинстве UNIX систем в качестве SSH клиента принято использовать пакет OpenSSH (<http://www.openssh.org/>). Обычно оказывается достаточно поставить пакет с именем `ssh`, если он по каким-то причинам всё ещё не стоит на машине.

OpenSSH позволяет на основе пар открытого и приватного ключей организовать авторизацию без указания пользователем пароля на удалённой машине. Пользователь сперва локально создаёт публичный и приватный ключи. Для этого используется программа

ssh-keygen с ключём `-t` например `ssh-keygen -t dsa`. В результате в каталоге `$HOME/.ssh` должны появиться файлы `$HOME/.ssh/id_dsa.pub` и `$HOME/.ssh/id_dsa`. Публичный ключ необходимо предоставить системному администратору вычислительного центра. Системный администратор запишет публичный ключ в специальный файл (по умолчанию `$HOME/.ssh/authorized_keys`), который в дальнейшем будет использоваться для идентификации пользователя. После выполнения этой процедуры с локальной машины на удалённую можно заходить по протоколу SSH, вызвав команду `ssh` с указанием имени пользователя и имени удалённой машины, например так: `ssh my_user_name@regatta.hpc`. Если при авторизации необходимо использовать не стандартный `id_dsa` файл с ключём, а какой-то другой, то его можно указать с помощью ключа `-i`. Если необходимо на удалённой машине запускать приложения, требующие системы Xwindows, то в опциях команды `ssh` необходимо указать ключ `-X`, в противном случае приложения будут выдавать сообщение о том, что они не могут открыть дисплей.

Применяя те же методы шифрования, что и для терминального доступа, можно передавать файлы в сети безопасным с точки зрения несанкционированного чтения способом. Для этой цели в SSH есть механизм SCP — Secure Copy (команда `scp`) и SFTP — Secure File Transfer Protocol (команда `sftp`). Рассмотрим некоторые наиболее важные ключи команды `scp`.

- `-p` — Сохраняет время модификации, время доступа и права доступа как у файла-оригинала.
- `-r` — Рекурсивное копирование дерева каталогов с подкаталогами.

В целом синтаксис командной строки команды `scp` схож с синтаксисом командной строки `cp`. Допустим, нам необходимо скопировать каталог с именем `abc` на удалённую машину `xyz.nodomain` где существует учётная запись `nouser`. Удалённо каталог будет иметь имя `abc` и находится в домашнем каталоге пользователя. Команда будет выглядеть так:  
`scp -r abc nouser@xyz.nodomain:`

После двоеточия можно указать путь в файловой системе удалённой машины (по умолчанию путь отсчитывается от домашнего каталога пользователя). Если не указать ничего, то это будет домашний каталог.

Для передачи файлов можно использовать также встроенную в программу Midnight Commander (команда `mc`) обёртку над `ssh`. В этом случае возможно перетаскивание файлов из левой панели в правую и наоборот, где в одной из панелей отображается файловая система удалённой машины.

## 2.3 SSH в Windows

В Windows существует несколько SSH клиентов. Наиболее популярный из них пакет программ `putty`. Пакет является проектом с открытым исходным кодом, и его можно бесплатно скачать с сайта <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>. Возможна установка пакета без внесения записей в реестр Windows и без административных прав. В пакет входят следующие программы:

- `putty.exe` — графический интерфейс, на каждое устанавливаемое соединение появляется новое окно Windows. Сперва необходимо через графический интерфейс задать настройки соединения.
- `pscp.exe` — аналог `scp` в OpenSSH с точностью до большинства ключей. Работает только в командной строке.
- `psftp.exe` — аналог `sftp` в openSSH. Работает только в командной строке.

- `plink.exe` — аналог команды `ssh` пакета OpenSSH. Работает только в командной строке.
- `puttygen.exe` — аналог `ssh-keygen`, но работает в графическом режиме и создаёт файлы с ключами в формате, отличном от формата OpenSSH, но форматы легко приводятся друг к другу. Таким образом можно использовать пару приватного и публичного ключей, полученных в `putty` для OpenSSH.

Для передачи файлов по протоколу SSH для Windows существует надстройка над `putty`, которая называется WinSCP. Эта программа также является проектом с открытым исходным кодом и свободно скачивается с сайта <http://winscp.net>. Интерфейс основан на перетаскивании файлов из левой панели в правую и наоборот.

### 3 Компиляция программ

Обсудим теперь вопросы, связанные с компиляцией программ на различных многопроцессорных системах. Как правило, на любой вычислительной системе присутствуют компиляторы GNU: `gcc` — компилятор C, `g++` — компилятор C++ и `gfortran` — компилятор Fortran90. Эти компиляторы отличаются универсальностью и способностью скомпилировать в том числе не очень аккуратно написанный исходный код. Кроме описанных компиляторов могут присутствовать «фирменные» компиляторы, предназначенные для наиболее тонкого учёта особенностей именно той аппаратуры, для которой они выдают исполняемый код. Также почти всегда присутствует разнообразный набор программных библиотек, как переносимых, так и использующих особенности «целевой» аппаратуры.

Для параллельного программирования практически везде можно использовать одну или несколько реализаций MPI, а также OpenMP, который является частью компиляторов, и библиотеку `pthread`.

При распараллеливании программы на общую память довольно часто используют технологию стандарта OpenMP. Компиляторы GNU, начиная с версии 3.4, поддерживают стандарт OpenMP 2.0. (В текущий момент готовится стандарт версии 3.1.) Для того, чтобы можно было воспользоваться OpenMP из `gcc`, `g++` или `gfortran`, необходимо убедиться, что в операционной системе помимо самих компиляторов установлена библиотека `libgomp`. После этого можно компилировать исходный код с указанием ключа `-fopenmp`. Этот ключ приводит к тому, что компилятор начинает понимать прагмы и псевдокомментарии (для FORTRAN) OpenMP, а наличие библиотеки `libgomp` приведёт к тому, что будет возможным использование функций OpenMP, таких как `omp_get_wtime()`.

При распараллеливании программы на распределённую память обычно используют MPI. Для компиляции MPI-программ принято использовать скрипты, которые с указанием дополнительных параметров вызывают стандартные компиляторы C/C++ и FORTRAN. Итак, обычно используются:

- `mpicc` — компиляция C-программ использующих MPI-библиотеку
- `mpicxx` — компиляция C++-программ использующих MPI-библиотеку
- `mpif77` — компиляция FORTRAN77-программ использующих MPI-библиотеку
- `mpif90` — компиляция FORTRAN90-программ использующих MPI-библиотеку

После компиляции программа обычно запускается на выполнение командой `mpirun` или `mpiexec` с указанием фактического необходимого числа процессоров. Однако на практике практически на любой вычислительной системе присутствует понятие очереди задач

пользователей, поэтому вместо вызова программы `mpirun` происходит запуск специального скрипта, который добавляет задачу в очередь, а непосредственный запуск осуществляется самой системой очередей и скрыт от пользователя.

Остановимся более подробно на особенностях доступных вычислительных систем с точки зрения сборки исходного кода программы.

### 3.1 BlueGene P

Информацию по этой системе можно найти на сайте <http://hpc.cs.msu.su>.

### 3.2 Regatta

Вмеру устаревшую информацию об этой системе можно найти на сайте <http://www.regatta.cs.msu.su>.

### 3.3 Чебышёв МГУ

На СКИФ-МГУ «Чебышёв» установлено несколько реализаций MPI. Для переключения между ними используется программа `mpi-selector`. Чтобы получить список реализаций, наберите `mpi-selector --list`. Для выбора нужной реализации `mpi-selector --set your-mpi-variant`. Программа `mpi-selector` меняет переменные окружения, которые выставляются для пользователя в момент входа в систему, поэтому для того, чтобы изменения вступили в силу, необходимо выйти и заново войти на интерфейсную машину.

На машине доступны три вида компиляторов:

- GNU компиляторы: `gcc`, `g++`, `gfortran` (OpenMP включается опцией `-fopenmp`).
- Intel компиляторы: `icc`, `icpc`, `ifort` (OpenMP включается опцией `-openmp`).
- Pentium GCC компиляторы: `pgcc`, `pgCC`, `pgf77`, `pgf90` (OpenMP включается опцией `-mp`).

Более подробную информацию об этой системе можно найти на сайте <http://parallel.ru/cluster/>.

## 4 Запуск задач пользователя

Как уже было сказано ранее, практически всегда на вычислительных системах запуск заданий производится через систему очередей. Пользователь ставит задачу в очередь, указывая при этом набор необходимых ему ресурсов таких как: число процессоров, общее время выполнения задачи, возможные пожелания на размещение задачи по узлам вычислительной системы, число необходимых лицензий какого либо программного продукта, если таковой имеется и предоставляет ограниченное число лицензий на запуск. После попадания в очередь задача начинает по ней двигаться с целью достижения момента своего запуска. Здесь нужно быть достаточно внимательным, поскольку некоторые системы очередей ставят задачу в очередь, даже если известно, что ресурсов для запуска этой задачи недостаточно в принципе. Скорость продвижения по очереди зависит от запрошенного объёма ресурсов, в зависимости от выставленного пользователем приоритета и в зависимости от приоритета самого пользователя.

Рассмотрим, как осуществляется запуск заданий и отслеживание их состояний в различных вычислительных системах.