

# Работа 1. Основы работы с кластерными вычислительными системами.

Большинство вычислительных систем работают под управлением операционных систем на ядре Linux. Это значит, что основные принципы работы будут общими. Различные системы управления ресурсами вычислительных систем так же имеют общие черты. Работу с вычислительной системой можно разделить на несколько этапов:

- подключение (включая передачу необходимых данных);
- формирование и запуск задачи;
- отслеживание процесса выполнения;
- анализ результатов.

Рассмотрим каждый из этапов более подробно.

## **Подключение.**

Практически все без исключения вычислительные системы под управлением Linux для удаленного сетевого доступа используют протокол SSH. Существуют различные клиентские приложения практически для всех операционных систем (в том числе, и для работающих на мобильных устройствах). В \*nix системах ssh-клиент обычно устанавливается по-умолчанию; для Windows одним из наиболее распространенных клиентов является приложение PuTTY.

При работе в \*nix системе для подключения к удаленному компьютеру с именем *remote.host.dom* по протоколу SSH с именем пользователя *username* достаточно выполнить команду

```
ssh username@remote.host.dom
```

После ввода пароля пользователя будет доступен интерфейс командной строки с приглашением, имеющим, например, такой вид:

```
\[username@remote.host.dom ~\] $
```

В данном случае символ ~ обозначает домашний каталог, в котором находится пользователь после подключения, а после символа \$ ожидается ввод команд.

При работе в Windows системе необходимо запустить и настроить программу PuTTY, окно которой приведено на рис. 1. В поле Host Name необходимо ввести имя удаленного компьютера, при необходимости можно исправить значение поля Port. Для подключения необходимо нажать кнопку Open. Далее потребуется ввести имя пользователя и пароль. При успешном подключении пользователю будет доступен тот же интерфейс командной строки.

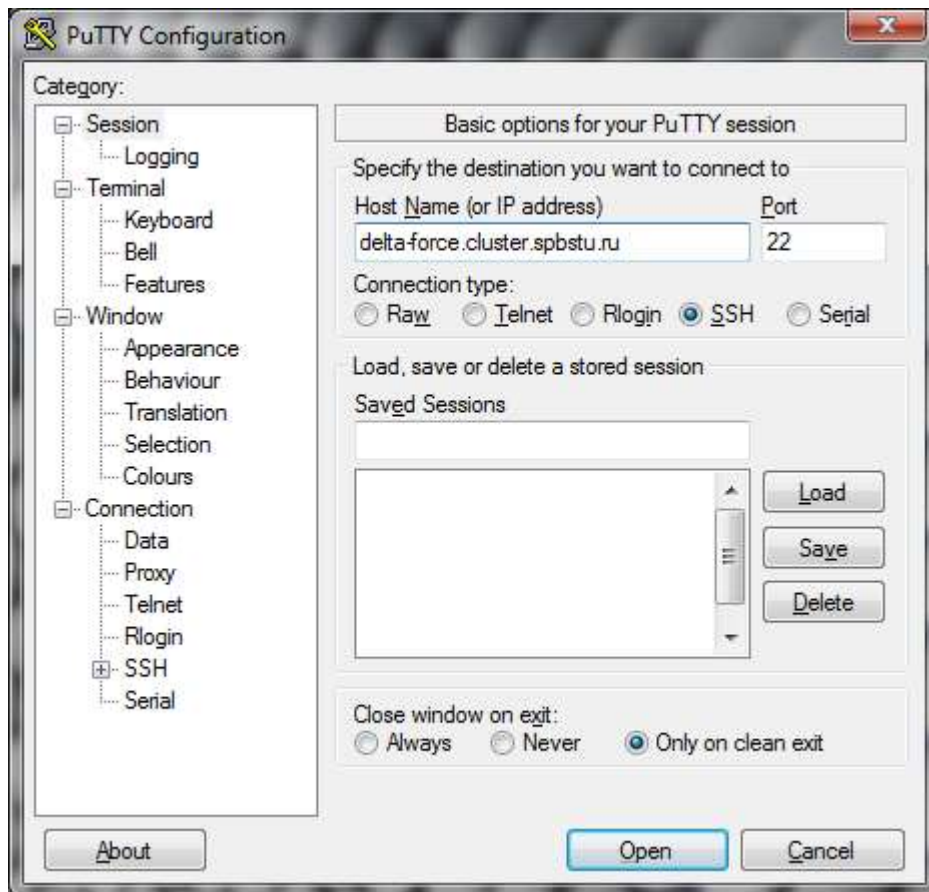


Рисунок 1 Рабочее окно программы PuTTY

Для передачи файлов в Linux системе обычно используется утилита `scp`, работающая в текстовом режиме. Для Windows будет удобно воспользоваться программой WinSCP, имеющей интуитивно понятный интерфейс. Настройки программы производятся по аналогии с PuTTY.

## **Основные принципы**

Для работы в командной строке потребуется знание основных принципов устройства ОС Linux.

Linux (в отличие от Windows) является регистрозависимой системой, т.е. важен используемый регистр: каталоги `MyHome` и `myhome` могут существовать одновременно.

Система имеет иерархическую файловую систему: все файлы и каталоги расположены в дереве с общим корнем, обозначаемым символом `/`. Все пути к файлам и каталогам делятся на абсолютные (указываются относительно корня) и относительные (относительно текущего каталога). В задании относительных путей используются следующие обозначения: символом точка обозначается текущий каталог, двумя точками – каталог уровнем выше. Пример:

`/usr/local/bin/somefile` – абсолютный путь к файлу `somefile` в каталоге `/usr/local/bin`;

`./somefile` – относительный путь к файлу `somefile`, находящемуся в текущем каталоге;

.././somefile – относительный путь к файлу, находящему в каталоге двумя уровнями выше.

## Основные команды

В качестве основных команд предлагается изучить команды работы с файлами, каталогами и процессами.

pwd	имя текущего каталога
cd	смена текущего каталога
ls	просмотр содержимого каталога; с ключом <code>-l</code> выводит информацию в длинном формате с указанием размера файлов и прав доступа; <code>-h</code> размер файлов в привычных единицах: кб, мб, гб и т.д.
mkdir	создание каталогов; с ключом <code>-p</code> создает структуру вложенных каталогов
rm	удаление файла; с ключом <code>-r</code> удаление каталога
echo	вывод текста в поток стандартного вывода
cat	вывод содержимого файла
less	просмотр содержимого файла или вывода программы в постраничном режиме, выход по клавише <code>q</code>
tail	вывод последних строк файла; с ключом <code>-n</code> выводит <code>n</code> последних строк, вместо 10 по умолчанию
ps	активные процессы пользователя
top	просмотр активных процессов в интерактивном режиме (диспетчер задач), выход по клавише <code>q</code>
mc	файловый менеджер

Ввод и вывод программ можно перенаправить: символ `<` используется для перенаправления потока стандартного ввода, `>` - для потока стандартного вывода.

Например, команда

```
echo some text > somefile
```

запишет строку "some text" в файл с именем somefile.

## Формирование задачи

Планировщики задач принимают задания как из командной строки, так и из файла, оформленного специальным образом.

Существует несколько основных систем управления ресурсами (системы управления очередями заданий): SLURM, Torque, SunGridEngine, Cleo. Рассмотрим использование системы SLURM (Simple Linux Utility for Resource Management). Основные утилиты описаны в таблице:

sinfo	информация о текущем состоянии узлов, управляемых SLURM
squeue	список текущих задач; по-умолчанию выдается список сначала исполняемых, затем ожидающих задач в порядке приоритета исполнения
salloc	резервирование ресурсов в режиме реального времени; обычно вызывается для резервирования ресурсов и запуска командной оболочки с возможностью удаленного доступа на выделенные узлы. Далее в полученной командной оболочке с помощью команды <code>sgun</code> запускаются параллельные задачи
sbatch	запуск пакетных заданий. В качестве параметра передается скрипт-файл, содержащий одну или более команд для запуска параллельных задач

scancel	отмена ожидающих или исполняющихся задач. Так же может использоваться для посылы произвольного сигнала всем процессам, относящихся к задаче
smap	информация о задачах и узлах в псевдографическом виде
srun	запуск параллельных задач. Принимает большое количество параметров, определяющих требования к ресурсам: минимальное/максимальное количество узлов, количество процессоров, имена определенных узлов, отдельные характеристики узлов (требования к памяти, дисковому пространству и т.п.). Задача может содержать множество подзаданий, выполняемых последовательно или параллельно на независимых или общих узлах (в пределах области резервирования задачи)

Для проверки работы системы можно выполнить простую команду:

```
srun -n 1 hostname
```

которая запрашивает 1 ядро и выполняет на нем команду hostname, поэтому в результате работы будет выведено имя вычислительного узла, на который попала задача.

Сформируем скрипт с аналогичным действием. В текстовом редакторе файлового менеджера или другом текстовом редакторе создать файл test.slurm со следующим содержимым:

```
#!/bin/bash
#SBATCH -n 1
hostname
```

Далее его можно отдать на выполнение командой

```
sbatch test.slurm
```

Строки, начинающиеся со #SBATCH, указывают системе требуемые параметры, основные из которых перечислены в таблице.

-n	требуемое количество процессорных ядер
-N	требуемое количество вычислительных узлов
-J	имя задачи
-o	имя файла для перенаправления стандартного потока вывода
-e	имя файла для перенаправления стандартного потока ошибок
-p	имя очереди для исполнения задачи
-t	максимальное время исполнения задачи

## Задания

Подключиться к удаленной системе и изучить описанные выше команды.