Кластер САФУ

Подключение и работа с MPI

Содержание:

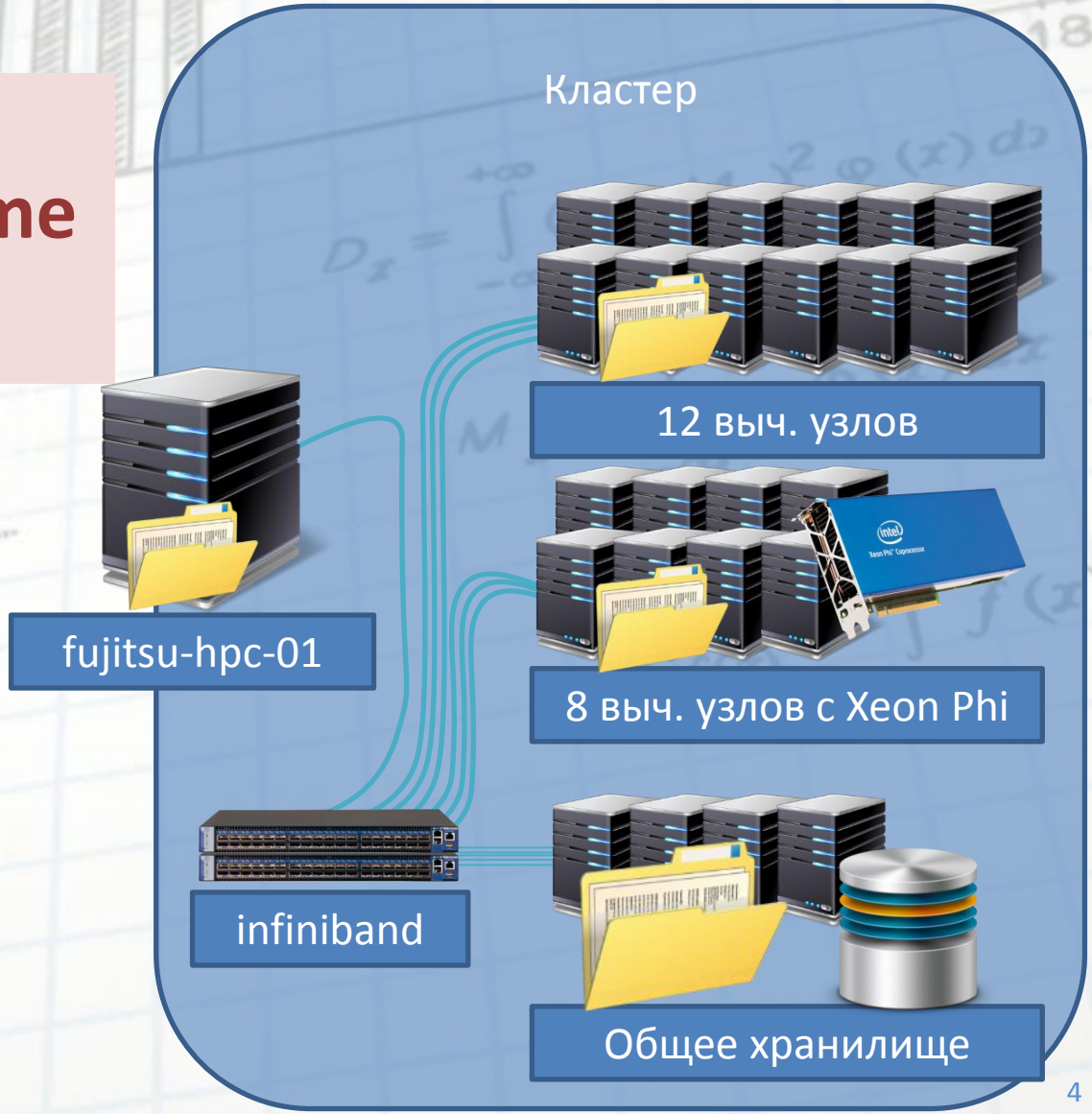
- Архитектура кластера
- Инструменты для удалённой работы

ПОДКЛЮЧЕНИЕ

Архитектура

Файловая система

- **На всех узлах директория /home является общей!**



Подключение

- Центральный узел:
  - SSH: fujitsu-hpc-01.narfu.ru
- Инструменты:
  - PuTTY – SSH-клиент
    - командная строка
  - WinSCP – SCP-клиент
    - работа с файлами
  - Xming – X-сервер
    - работа с графическими приложениями (например, некоторые компоненты ANSYS)

PuTTY

Создайте новый профиль:

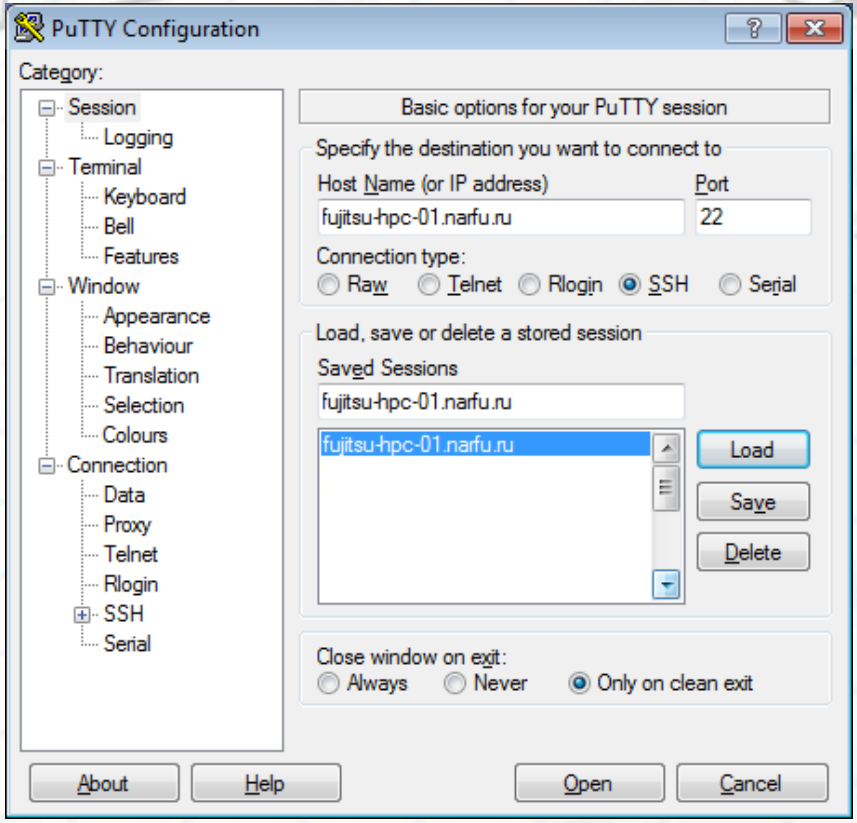
- Host: fujitsu-hpc-01.narfu.ru
- Port: 22
- Connection type: SSH

Установите поддержку Unicode: Window\Translation:

- Remote character set: UTF-8

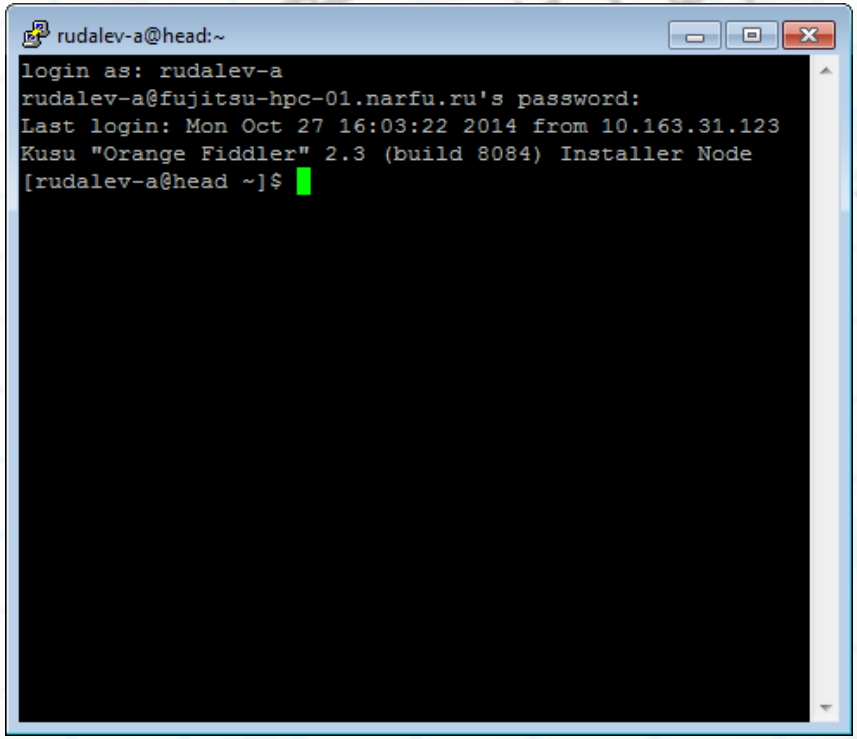
Если нужна поддержка приложений с графическим интерфейсом, установите: Connection\SSH\X11:

- ☑ Enable X11 forwarding



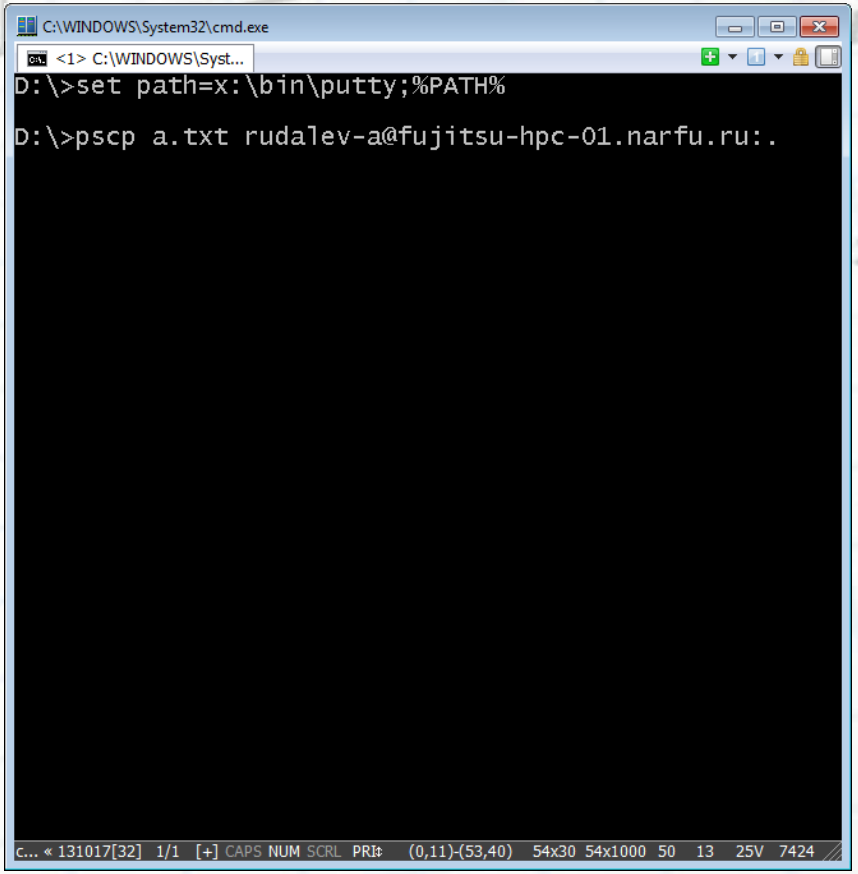
PuTTY: вход в систему

- Введите имя пользователя
- Введите пароль
  Внимание: при вводе пароля символы не печатаются на экране!!!
- В приглашении «командной строки» указывается имя пользователя и имя узла.
- Внутреннее имя головного узла: head



Копирование файлов: scp

- **scp** – стандартная UNIX-программа для удалённого копирования файлов по протоколу SSH
- **pscp** – реализация scp для MS Windows в пакете PuTTY
- Рекомендация: используйте программу **ConEmu** при работе в командной строке MS Windows



Копирование файлов: WinSCP

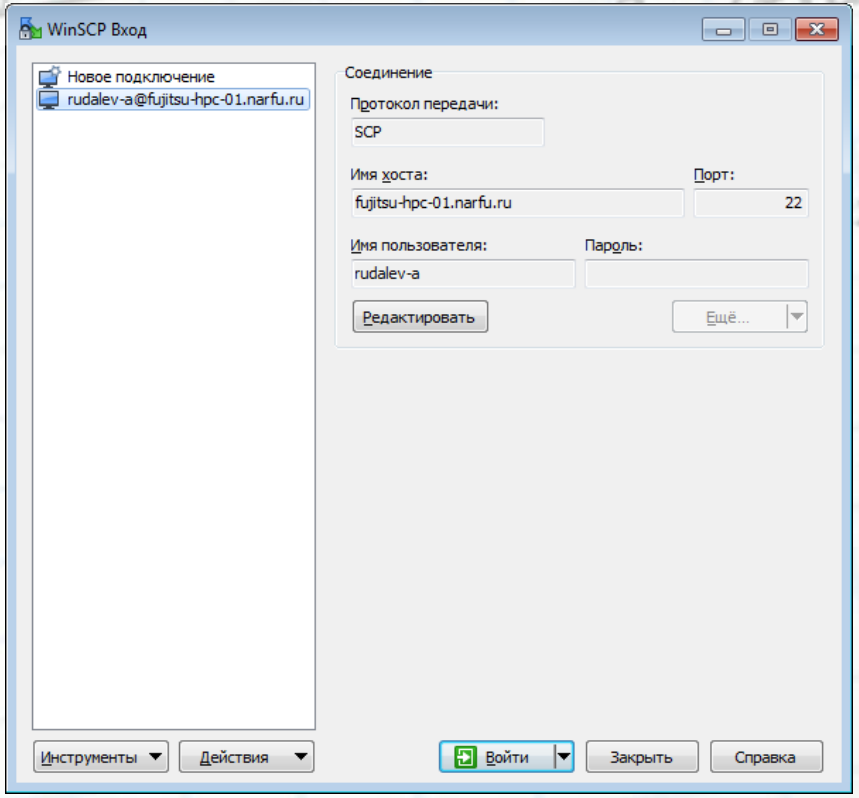
- **WinSCP** – реализация SFTP/FTP/SCP графического клиента для MS Windows

Необходимо указать:

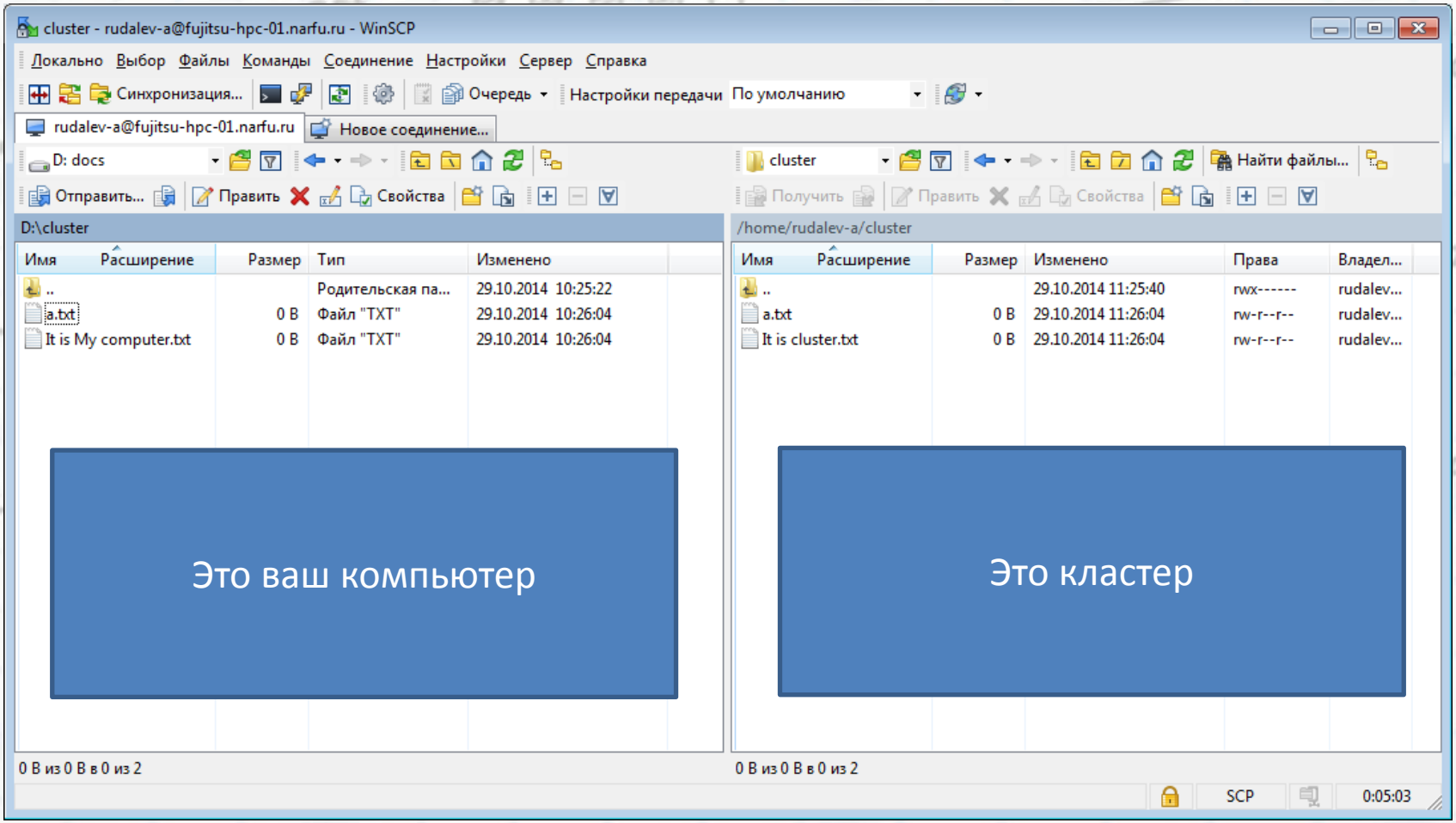
- Протокол передачи: SCP
- Имя хоста: fujitsu-hpc-01.narfu.ru
- Имя пользователя: Ваш логин

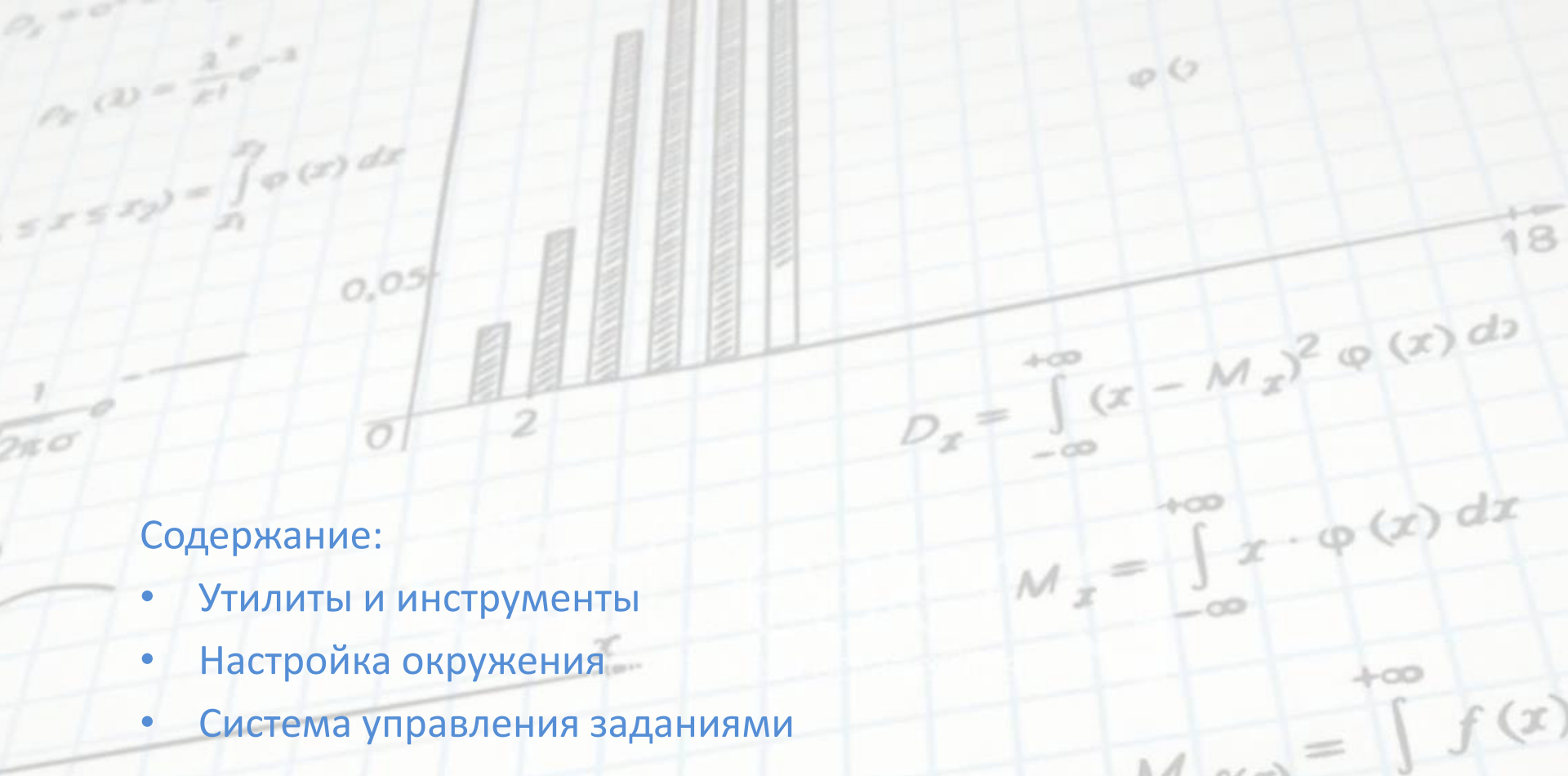
Не указывайте пароль:

- тогда он будет запрашиваться при подключении



Копирование файлов: WinSCP





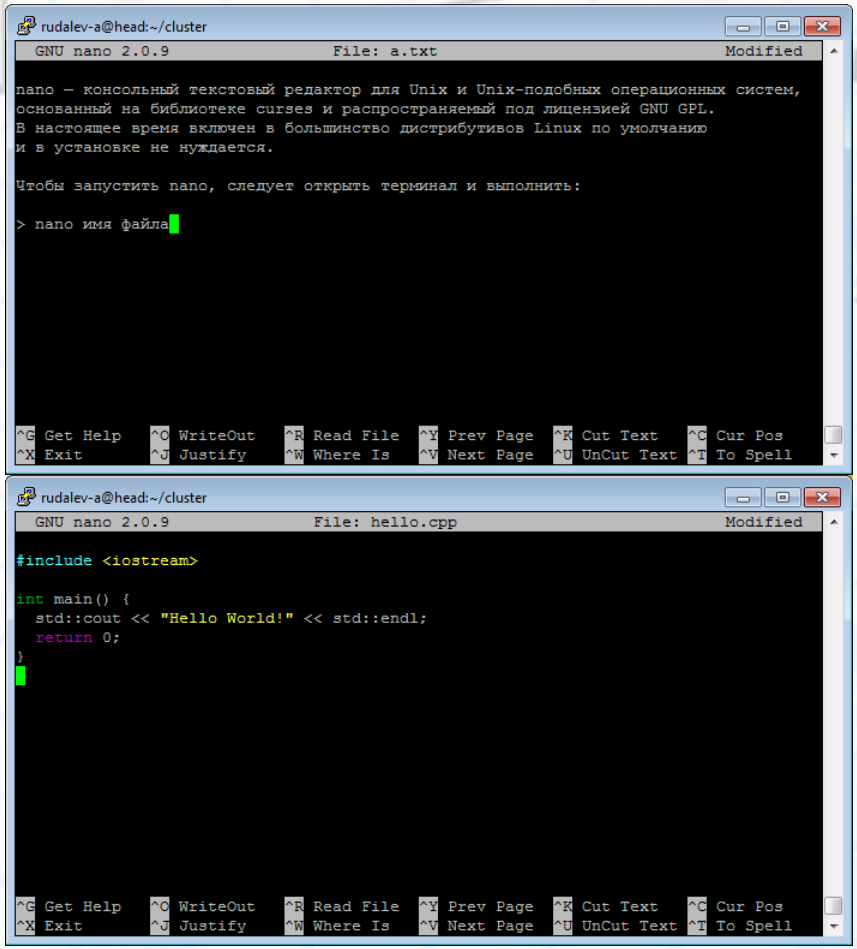
Содержание:

- Утилиты и инструменты
- Настройка окружения
- Система управления заданиями

РАБОТА НА КЛАСТЕРЕ

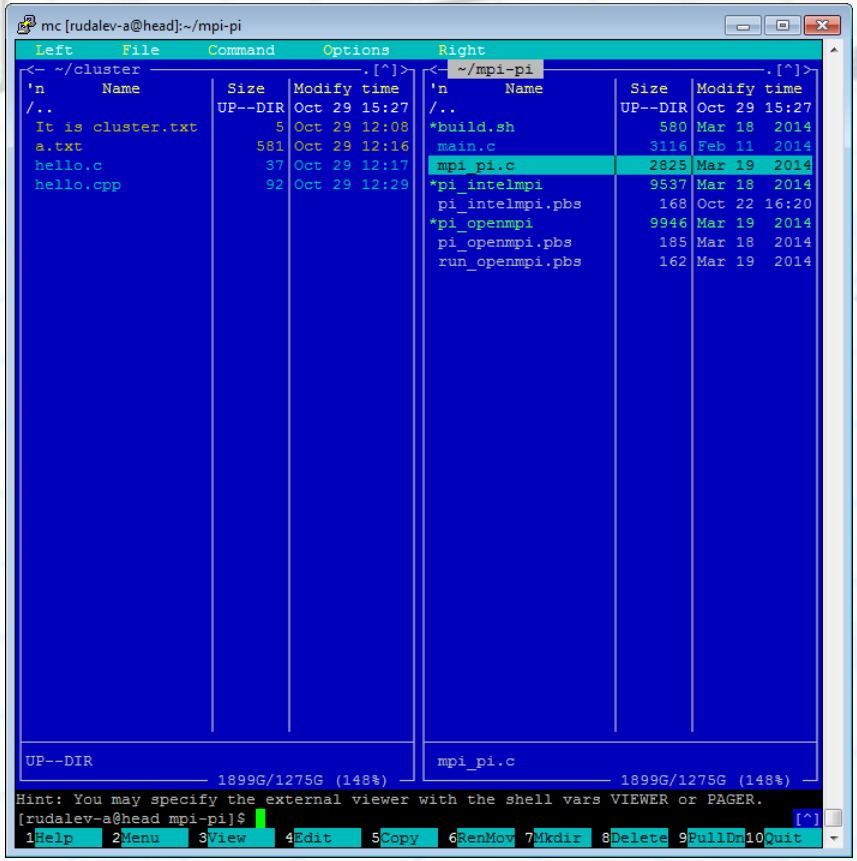
Редактирование файлов: nano

- **nano** – консольный текстовый редактор
- Запуск: > nano имя_файла
- Управление через сочетания клавиш Ctrl+…. С.м. подсказку в низу
  - Ctrl+O – записать изменения, нужно подтвердить имя файла нажав Enter
  - Ctrl+X – выйти из программы
- Поддерживает подсветку синтаксиса, если она включена в файле ~/.nanorc
  - например, для C/C++: include /usr/share/nano/c.nanorc
- Так же есть и классические **vi** и **emacs**



Файловый менеджер: mc

- **mc** – классический консольный двухпонельник для UNIX
- Запуск:
  > mc



- Просмотреть файл: F3
- Изменить файл: F4
- Скопировать: F5
- Переместить: F5
- Переименовать: Shift+F6
- Удалить: F8
- Выход: F10

Environment modules

- Средство управления переменными окружения
- Список активированных модулей:
  - > module list
- Список доступных модулей:
  - > module avail
- Активация модуля:
  - > module load ИМЯ_МОДУЛЯ
  - Используйте автодополнение по клавише Tab
- Замена модуля
  - > module switch ЧТО НА_ЧТО

PBS

- Система управления заданиями
- Файл задания
- Просмотр очереди:
  - > qstat
- Помещение задания в очередь:
  - > qsub файл.pbs
- Удаление из очереди:
  - > qdel ID_задания

PBS: Пример задания

```
[rudalev-a@head ~]$ cat Hello.pbs  <- Вывод содержимого
файла

#!/bin/bash                    <- Кто исполнит скрипт
#PBS -N HelloWorld             <- Краткое название задачи
echo Hello world!              <- Команда

[rudalev-a@head ~]$ qsub Hello.pbs <- Запуск скрипта в PBS

809.Head                       <- ID задачи

[rudalev-a@head ~]$ cat HelloWorld.o809

Hello world!                   <- Вывод STDOUT

[rudalev-a@head ~]$ cat HelloWorld.e809

                               <- Вывод STDERR
```

PBS: Пример задания

```
#!/bin/bash
#PBS -N HelloWorld
#PBS -l select=5                    <- запросить 5
ядер
#PBS -l walltime=00:01:00 <- макс. время
работы

echo Hello world! I am main process
date +"Date: %Y-%m-%d"
echo Time: `date +%H:%M:%S.%N`
echo Main host: `/bin/hostname`
echo Can run on: `cat ${PBS_NODEFILE}`
echo - PBS Environment ----------
export | grep PBS
echo ----------------------------
```

PBS: Пример задания

```
[rudalev-a@head ~]$ qsub Hello.pbs
810.head

[rudalev-a@head ~]$ cat HelloWorld.o810

Hello world! I am main process
Date: 2014-10-29
Time: 14:25:50.516809978
Main host: cn20
Can run on: cn20.fujitsu-hpc cn20.fujitsu-hpc cn20.fujitsu-hpc cn20.fujitsu-hpc cn20.fujitsu-hpc
- PBS Environment -----------
declare -x PBS_ENVIRONMENT="PBS_BATCH"
declare -x PBS_JOBCOOKIE="0000000024CC5E53000000001CB8383E"
declare -x PBS_JOBDIR="/home/rudalev-a"
declare -x PBS_JOBID="810.head"
declare -x PBS_JOBNAME="HelloWorld"
declare -x PBS_MOMPORT="15003"
declare -x PBS_NODEFILE="/var/spool/PBS/aux/810.head"
declare -x PBS_NODENUM="0"
declare -x PBS_O_HOME="/home/rudalev-a"
declare -x PBS_O_HOST="head.fujitsu-hpc"
declare -x PBS_O_LANG="en_US.UTF-8"
declare -x PBS_O_LOGNAME="rudalev-a"
declare -x PBS_O_MAIL="/var/spool/mail/rudalev-a"
declare -x PBS_O_QUEUE="workq"
declare -x PBS_O_SHELL="/bin/bash"
declare -x PBS_O_SYSTEM="Linux"
declare -x PBS_O_WORKDIR="/home/rudalev-a"
declare -x PBS_QUEUE="workq"
declare -x PBS_TASKNUM="1"
-----------------------------
```

Ключи PBS

- #PBS -l select=5
  - Запросить 5 ядер для 5 процессов
  - Для большинства **MPI-приложений** его будет достаточно
  - PBS_NODEFILE: cn20.fujitsu-hpc cn20.fujitsu-hpc cn20.fujitsu-hpc cn20.fujitsu-hpc cn20.fujitsu-hpc
- #PBS -l select=5:ncpus=10
  - Запросить по 10 ядер для 5 процессов (всего 50), каждый процесс будет использовать 10 потоков
  - Для **OpenMP-приложений** можно использовать: -l select=1:ncpus=10
  - PBS_NODEFILE: cn20.fujitsu-hpc cn20.fujitsu-hpc cn16.fujitsu-hpc cn16.fujitsu-hpc cn14.fujitsu-hpc
- #PBS -l nodes=5:ppn=2:ncpus=20
  - Запросить по 2 ядра на 5 разных вычислительных узлах, на каждом узле будет использовано 20 ядер
  - Этот вариант используется когда необходимо **целиком зарезервировать узлы**, запретив запускать на них другие задачи.
  - PBS_NODEFILE: cn20.fujitsu-hpc cn20.fujitsu-hpc cn16.fujitsu-hpc cn16.fujitsu-hpc cn14.fujitsu-hpc cn14.fujitsu-hpc cn17.fujitsu-hpc cn17.fujitsu-hpc cn15.fujitsu-hpc cn15.fujitsu-hpc

Содержание:

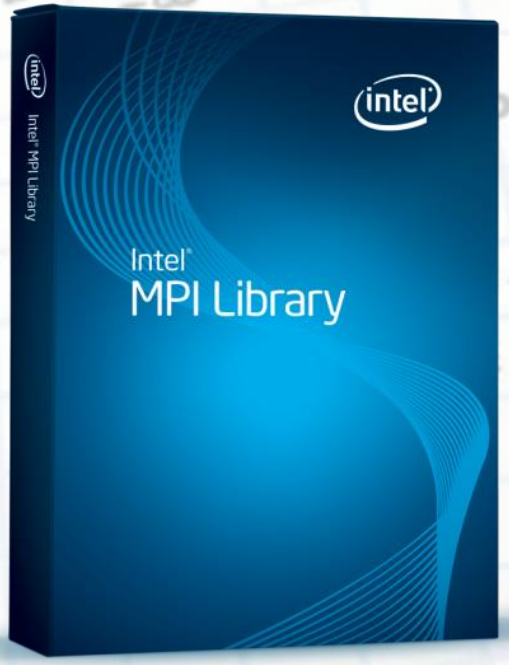
- Реализации MPI
- Настройка окружения
- Компиляция
- Запуск

КОМПИЛЯЦИЯ И ЗАПУСК MPI-ПРИЛОЖЕНИЙ

Реализации MPI

OpenMPI

Intel MPI



Компиляция

- Настройка окружения:
  - OpenMPI: > module load openmpi-pbs
  - Intel MPI: > module load intel/mpi
- Компиляция
  - Си: > mpicc файл.c -o исп_файл
  - C++: > mpicxx файл.cxx -o исп_файл
  - F77: > mpif77 файл.f77 -o исп_файл
  - F90: > mpif77 файл.f90 -o исп_файл

Требования к MPI-программе

- Запуск в пакетном режиме через систему **PBS**
- Запуск MPI-приложений разрешён только на вычислительных узлах
- Для mpirun не надо указывать ключи определяющие количество потоков. Все данные будут взяты из переданных переменных окружения
- Не забывайте в скрипте запуска переходить в рабочую директорию и заново определять переменные окружения, например, через modules load

Hello World: пример на Си

```c
/* C Example
   file: mpi_hello.c
*/
#include <stdio.h>
#include <mpi.h>


int main (argc, argv)
     int argc;
     char *argv[];
{
  int rank, size;

  MPI_Init(&argc, &argv); // starts MPI
  MPI_Comm_rank(MPI_COMM_WORLD, &rank); // get current process id
  MPI_Comm_size(MPI_COMM_WORLD, &size); // get number of processes
  printf( "Hello world from process %d of %d\n", rank, size );
  MPI_Finalize();
  return 0;
}
```

Hello World: PBS-скрипт

```
#!/bin/bash
#PBS -N mpi_hello
#PBS -l select=10
#PBS -l walltime=00:05:00

cd ${PBS_O_WORKDIR}
module load openmpi-pbs

mpirun mpi_hello
```

Hello World: Компиляция и запуск

- Подключите OpenMPI

```
> module load openmpi-pbs
```

- Создайте файлы mpi_hello.c и mpi_hello.pbs:

```
> nano mpi_hello.c
> nano mpi_hello.pbs
```

- Скомпилируйте:

```
> mpicc  mpi_hello.c -o mpi_hello
```

- Запустите:

```
> qsub mpi_hello.pbs
888.head
```

- Посмотрите результат:

```
> cat mpi_hello.o888
```

Содержание:

- Запуск offload-приложения на Intel Xeon Phi

КОМПИЛЯЦИЯ И ЗАПУСК ПРИЛОЖЕНИЙ ДЛЯ XEON PHI

Xeon Phi: offload example ¼ main.cpp

```
#include <stdio.h>
#include <omp.h>

#pragma offload_attribute(push, target(mic))
int getThreadCount()
{
    int thread_count;

    #pragma omp parallel
    {
        #pragma omp single
            thread_count = omp_get_num_threads();
    }
    return thread_count;
}
#pragma offload_attribute(pop)
...
```

Xeon Phi: offload example 2/4
main.cpp

```
int main()
{
    printf("Intel CPU:\n");
    printf("Number of threads on CPU: %d\n", getThreadCount());

    int number_of_coprocessors =
        _Offload_number_of_devices();

    printf("Intel Xeon Phi:\n");
    printf("Number of coprocessors: %d\n",
        number_of_coprocessors);

    for (int i = 0; i < number_of_coprocessors; ++i)
    {
        int thread_count;
        #pragma offload target(mic:i)
        {
            thread_count = getThreadCount();
        }
        printf("Number of threads on MIC%d: %d\n", i,  thread_count);
    }

    return 0;
}
```

Xeon Phi: offload example 3/4

```
#!/bin/bash
#PBS -N lab1-mics
#PBS -l select=1:ncpus=20:nmics=1
#PBS -l walltime=00:05:00

cd ${PBS_O_WORKDIR}
echo Main host: `/bin/hostname`
export OMP_NUM_THREADS=
module load intel/composer_xe
module load intel/mpi

./lab1_thread_test
```

Xeon Phi: offload example 4/4

```
$ module load intel/composer_xe
$ module load intel/mpi
$ icc -O2 -openmp main.cpp -o
lab1_thread_test
```

Содержание:

- Проброс портов

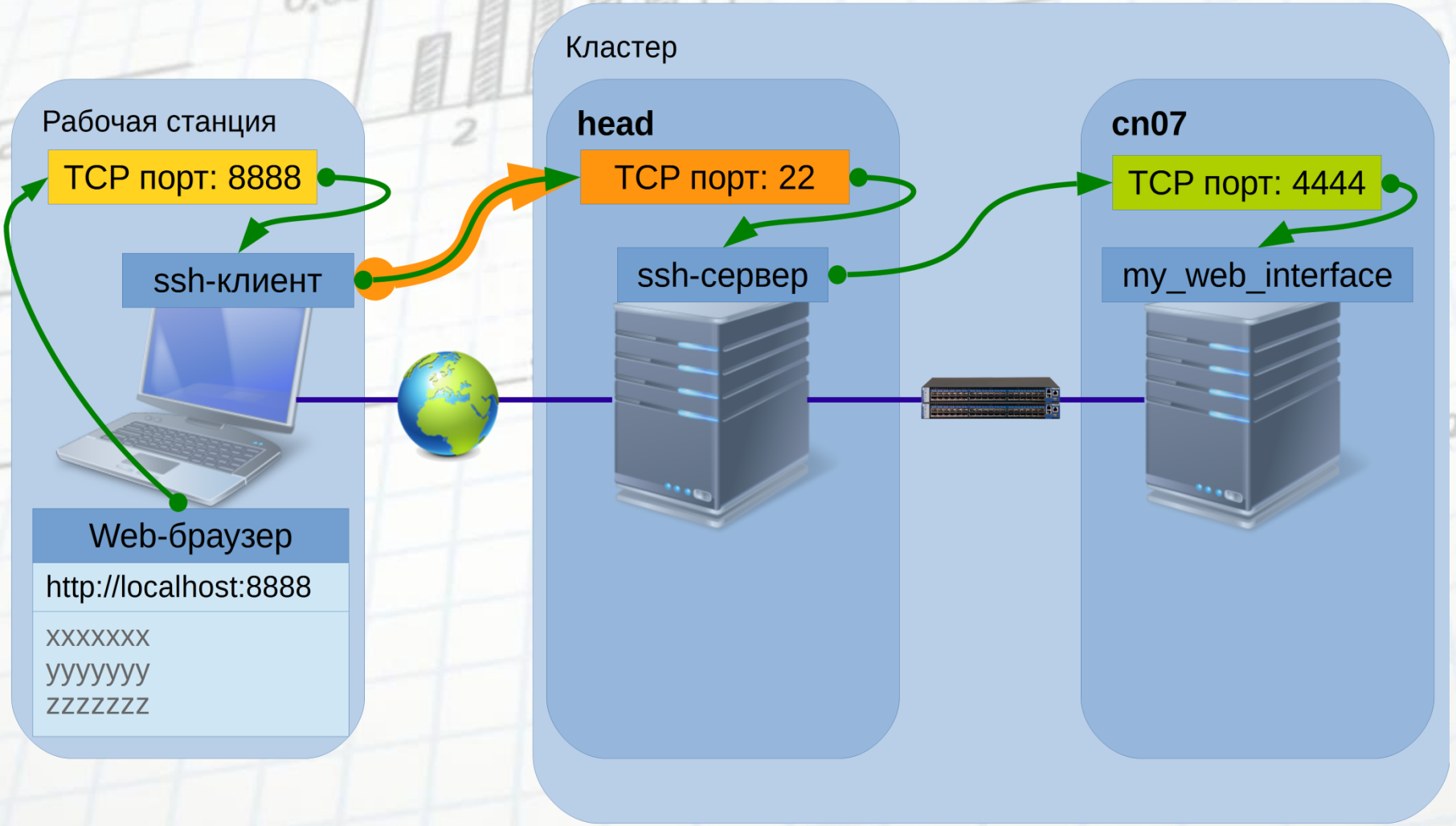
ДОПОЛНИТЕЛЬНО

Проброс портов: Задача

- **Пример:** На кластере запущенна задача, которая имеет Web-интерфейс для мониторинга результатов. Он предоставляется процессом my_web_interface, работающим на 4444 tcp-порте вычислительного узла cn07.
- **Необходимо:** С рабочего компьютера пользователя подключится к этому web-интерфейсу.
- **Проблема:** Если в web-браузере на компьютере пользователя набрать адрес http://cn07:4444/ - то подключения не будет (вычислительный узел спрятан за головным узлом кластера).
- **Решение:** Запустить на рабочем компьютере посредника, который откроет локальный tcp-порт (например, 8888) и будет через ssh-соединение к головному узлу кластера «соединит» его с портом 4444 вычислительного узла cn07. Тогда в адресной сроке Web-браузера на рабочем компьютере можно ввести http://localhost:8888/

Проброс портов

```
*nix:    ssh -N -L 8888:cn07:4444 user@head
win:     plink -N -L 8888:cn07:4444 user@head
```



Контактное лицо:

Рудалёв Александр Васильевич <**a.rudalev@narfu.ru**>

В ДОБРЫЙ ПУТЬ!