|  |
| --- |
| **ЗАО «РСК Технологии»** |
| **Высокопроизводительная вычислительная система для решения задач биоинформатики**  **(с комплектом расширения)** |
| Руководство пользователя |
| **69573991.009.МФТИ.И3.2** |
|  |
| МОСКВА  2012 |

АННОТАЦИЯ

Данный документ является частью конструкторской документации высокопроизводительной вычислительной системы для решения задач биоинформатики (здесь и далее подразумевается система с установленным комплектом расширения), разрабатываемой на основании контракта № К12-021, заключенным между Государственным образовательным учреждением высшего профессионального образования «Московский физико-технический институт (государственный университет)» и ЗАО «РСК Технологии».

Документ представляет собой руководство пользователя высокопроизводительной вычислительной системы для решения задач биоинформатики, созданной в Государственном образовательном учреждении высшего профессионального образования «Московский физико-технический институт (государственный университет)» в городе Долгопрудный.

СОДЕРЖАНИЕ

1. Назначение и условия применения 6

2. Подготовка к работе 9

2.1 Требования к квалификации пользователя 9

2.2 Вход в систему и аутентификация 9

2.2.1 Общие принципы 9

2.2.2 Получение реквизитов для удалённого доступа 9

2.2.3 Удалённый терминальный доступ 9

3. Описание операций 11

3.1 Терминальный доступ 11

3.1.1 Получение удалённого терминального доступа к системе 11

3.1.2 Получение имени текущей рабочей директории 11

3.1.3 Просмотр содержимого директории 11

3.1.4 Просмотр атрибутов избранного файла в директории 11

3.2 Использование SFTP 12

3.2.1 Получение удалённого доступа к системе 12

3.2.2 Получение имени текущей рабочей директории 12

3.2.3 Получение списка файлов в текущей директории 12

3.2.4 Загрузка файла 12

3.2.5 Получение файла 12

3.2.6 Выход из клиента 12

3.3 Конфигурация среды пользователя 13

3.3.1 Использование Environmental Modules 13

3.3.2 Список доступных модулей 13

3.3.3 Выбор компилятора 14

3.3.4 Замена компилятора 14

3.3.5 Использование библиотек 14

3.4 Планировщик задач (SLURM) 14

3.4.1 Состояние планировщика задач 14

3.4.2 Просмотр состояния очереди задач 15

3.4.3 Запуск MPI-задач 15

3.4.4 Освобождение выделенных узлов 20

3.4.5 Удаление задачи 20

3.4.6 Переменные окружения, используемые планировщиком SLURM 20

3.5 Общие ресурсы 21

Список сокращений 22

Данное руководство пользователя составлено для высокопроизводительной вычислительной системы для решения задач биоинформатики, разрабатываемой на основании контракта № АЭ11-050, заключенным между Государственным образовательным учреждением высшего профессионального образования «Московский физико-технический институт (государственный университет)» и ЗАО «РСК Технологии».

Высокопроизводительная вычислительная система для решения задач биоинформатики (далее – вычислительная система) – вычислительная система, реализованная на базе инновационной системы с жидкостным охлаждением РСК «Торнадо». Основной функцией данной вычислительной системы является решение задач с высокой вычислительной нагрузкой.

Кроме этого, вычислительная система обладает такими свойствами, как:

* высокой энергоэффективностью решения за счет отвода тепла с помощью жидкостной системы охлаждения;
* высокой вычислительной плотностью, необходимой для реализации сверхбольших суперкомпьютеров с высокоскоростными вычислительными сетями;
* высокой отказоустойчивостью за счет отсутствия движущихся частей, таких как вентиляторы системы охлаждения и жёсткие диски.

Настоящий документ является достаточным для ознакомления конечным пользователем вычислительной системы для решения задач с высокой вычислительной нагрузкой.

# Назначение и условия применения

Основные технические характеристики вычислительной системы представлены в таблице 1–1.

| Табл. 1–. Технические характеристики вычислительной системы | | | |
| --- | --- | --- | --- |
| **Параметр** | **Значение для одного узла** | **Значение для Вычислителя в целом** | |
| Теоретическая пиковая производительность Вычислителя | – | 83 148 Гфлопс | |
| Теоретическая пиковая производительность одного узла | 371,2 Гфлопс | – | |
| Тип используемых процессорных чипов | Intel Xeon E5-2690  8 ядер, тактовая частота ядра 2,90 ГГц, QPI 8,0 ГТ/с, размер кэша 20 МБ | | |
| Характеристики коммуникационной сети Вычислителя | Infiniband QDR, 40 Гбит/с, обеспечивается половинная бисекционная пропускная способность для всех вычислительных узлов. | | |
| Сеть Ethernet | Обеспечение доступа ко всем вычислительным узлам со скоростью 1 Гбит/с | | |
| Количество процессорных чипов | 2 | | 448 |
| Количество процессорных ядер | 16 | | 3 584 |
| Объем памяти | 64 ГБ (4 ГБ на ядро) | | 14 336 ГБ |
| Количество узлов | - | | 224 |
| Тип дисков | 600 ГБ SATA HDD | | |
| Количество дисков | 1 | | 224 |

Функциональная схема высокопроизводительной вычислительной системы представлена на рисунке 1–1.

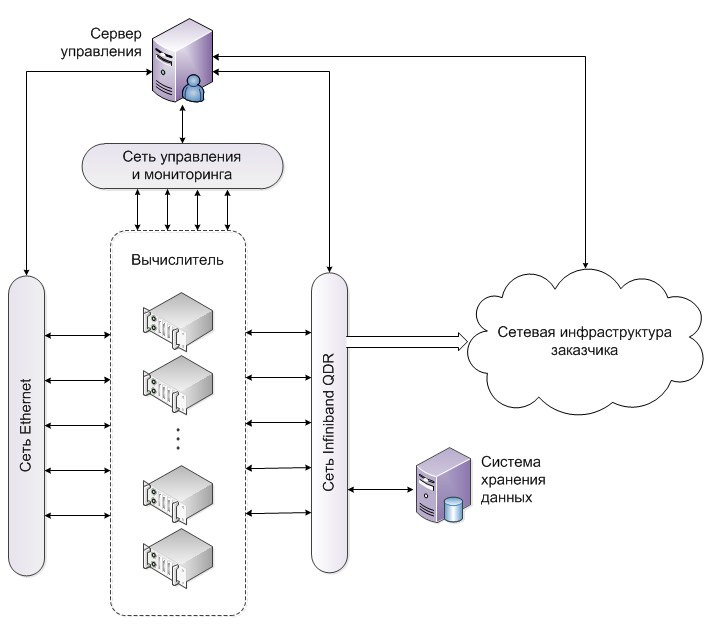


Рис. –. Функциональная схема вычислительной системы

В состав вычислительной системы входят Вычислитель, сервер управления Вычислителем и система хранения данных. Для связи компонентов в единую систему и с сетевой инфраструктурой заказчика используется несколько сетей, каждая из которых служит для выполнения определённых функций.

Вычислитель представляет собой кластерную систему из множества узлов, объединённых между собой высокопроизводительной коммуникационной сетью, реализованной по технологии Infiniband QDR. Коммуникационная сеть поддерживает реализацию основных примитивов библиотеки MPI версии 2.0 и построена по топологии, обеспечивающей половинную бисекционную пропускную способность.

Все узлы Вычислителя и сервер управления подключены к сети Gigabit Ethernet, которая используется для осуществления удалённого доступа к вычислительным ресурсам системы.

Табл. –. Описание системы

|  |  |
| --- | --- |
| Планировщик ресурсов SLURM | |
| Очередь пользовательских заданий | Имя – work  (просмотр состояния «sinfo –p work») |
| Диапазон адресов узлов | node001-node224 |
| Установленное по умолчанию время выполнения задачи (пользователь может установить самостоятельно) | 20 минут |
| Максимально доступное время выполнения задачи | 1 день |
| Network | |
| Точка доступа администраторов системы – сервер управления вычислителем | Адрес - cluster.iscalare.mipt.ru |
| Единая точка доступа пользователей - сервер доступа пользователей «login» | Адрес - cluster.iscalare.mipt.ru |
| Адресный план комплекса | |
| Сеть управления – eth0 | 10.4.4.0/22 |
| Сеть транспортная – Ib0 | 10.4.8.0/22 |
| Сеть доступа к модулям управления узлов (BMC) | 10.4.0.0/22 |
| Доступные модули пакета Environmental Modules | |
| parallel/mpi.intel/4.0.3.008 | Intel MPI |
| compilers/composer\_xe/2011\_sp1.6.233 | Компиляторы Intel Composer XE |
| compilers/cplusplus/gnu/4.4.6 | Компилятор GNU C |
| compilers/cplusplus/intel/12.1.0 | Компилятор Intel C++ |
| compilers/fortran/gnu/4.4.6 | Компилятор GNU Fortran |
| dot | Добавляет текущую директорию к переменной окружения PATH |
| launcher/slurm(default) | Использование инструментов пакета SLRUM для запуска MPI задач (srun) |
| launcher/intel | Использование инструментов пакета Intel MPI для запуска MPI задач (mpiexec.hydra) |

# Подготовка к работе

## Требования к квалификации пользователя

Квалификация пользователя, допускаемого к эксплуатации вычислительной системы, должна обеспечивать эффективное функционирование системы во всех заданных режимах.

Пользователь должен пройти общую и специальную подготовку по работе со средствами вычислительной системы и средствами вычислительной техники.

Общая подготовка должна включать в себя получение навыков работы с программным обеспечением в объеме навыков пользователей вычислительной системы.

Специальная подготовка должна включать в себя получение навыков работы с системным и прикладным обеспечением вычислительной системы в объеме навыков ее использования.

## Вход в систему и аутентификация

### Общие принципы

Политика работы вычислительной системы подразумевает интерактивный вход пользователя на консоли вычислительных узлов, который возможен только из планировщика задач, основной интерфейс к которому представлен на сервере управления. В каждый конкретный момент времени планировщик выделяет вычислительный узел в единоличное использование пользователю, запросившему такой доступ.

### Получение реквизитов для удалённого доступа

Необходимые параметры и настройки для обеспечения доступа предоставляет Оператор вычислительной системы.

Для доступа к системе необходима учётная запись и пароль, а также адрес сервера управления.

### Удалённый терминальный доступ

Удалённый терминальный доступ пользователя обеспечивается средствами SSH (Secure Shell).

В среде Linux используется приложение ssh, для операционных систем семейства Microsoft – PuTTY.Приложение доступно для свободного скачивания по адресу http://www.chiark.greenend.org.uk/~sgtatham/putty/.

Для удалённой работы с файлами используется подсистема SSH — SFTP, описанная ниже.

# Описание операций

## Терминальный доступ

Каталог пользователя доступен на каждом узле вычислительной системы в директории вида /home/user, где user — имя пользователя.

В примере представлен вывод клиента SSH для командной строки.

### Получение удалённого терминального доступа к системе

% ssh user@mipt

Last login: Fri Aug 3 14:50:50 2012 from 178.209.98.254

user@mipt-head:~$

### Получение имени текущей рабочей директории

user@mipt-head:~$ pwd

/home/user

### Просмотр содержимого директории

user@mipt-head:~$ ls

schrodinger test2

### Просмотр атрибутов избранного файла в директории

user@mipt-head:~$ ls -lad act\_test1.tgz

-rw-r--r-- 1 user users 3305845 Sep 22 2011 act\_test1.tgz

Примеры переходов по дереву директорий:

user@mipt-head:~$ cd opt/

user@mipt-head:~/opt$ pwd

/home/user/opt

user@mipt-head:~/opt$ ls -la

total 32

drwxr-xr-x 8 user users 4096 Oct 17 2011 .

drwxr-xr-x 32 user users 4096 Aug 3 14:50 ..

drwxr-xr-x 2 user users 4096 Oct 19 2011 bin

drwxr-xr-x 6 user users 4096 Oct 17 2011 fftw

drwxr-xr-x 6 user users 4096 Oct 17 2011 gromacs

drwxr-xr-x 2 user users 4096 Oct 19 2011 include

drwxr-xr-x 3 user users 4096 Oct 19 2011 lib

drwxr-xr-x 4 user users 4096 Oct 17 2011 share

user@mipt-head:~/opt$ cd ..

user@mipt-head:~$ cd /

user@mipt-head:/$ pwd

/

user@mipt-head:/$ cd

user@mipt-head:~$ pwd

/home/user

user@mipt-head:~$

## Использование SFTP

### Получение удалённого доступа к системе

В примере представлен вывод клиента SFTP для командной строки:

% sftp mipt

Connected to mipt.

### Получение имени текущей рабочей директории

sftp> pwd

Remote working directory: /home/user

### Получение списка файлов в текущей директории

sftp> ls

schrodinger test2

sftp>

### Загрузка файла

sftp> put test

Uploading test to /home/user/test

test

100% 0 0.0KB/s 00:00

### Получение файла

sftp> get test

Fetching /home/user/test to test

### Выход из клиента

sftp> bye

## Конфигурация среды пользователя

### Использование Environmental Modules

Для управления множественными версиями различных прикладных программных пакетов и библиотек на вычислительной системе установлен пакет Environmental Modules.

Данный пакет позволяет гибко настраивать переменные окружения и пакетных задач для использования тех или иных версий программного обеспечения и отслеживания их зависимостей. Кроме того использование Environmental Modules позволяет гибко управлять разными версиями приложения.

Пакет состоит из модулей, описанных в module files, которые доступны по адресу: /opt/basis/modules/Modules.

Пользователь может создавать свой набор пользовательских файлов в директории, в которой осуществляется запуск задачи.

Конфигурация подсистемы осуществляется путём правки конфигурационных файлов в директории **/opt/basis/modules/Modules/default/modulefiles** головного (head) сервера кластера.

Каждый модуль содержит информацию, необходимую для настройки окружения под конкретное приложение. Настройка осуществляется через задание переменных PATH, MANPATH, INCLUDE, LD\_LIBRARY\_PATH и т.д.

Модули могут быть динамически и автоматически подгружены и выгружены в свободном режиме. Поддерживаются все популярные командные интерпретаторы shell, включая bash, ksh, zsh, sh, csh, tcsh, в том числе такие, как perl.

По умолчанию при входе в систему для пользователя автоматически загружаются следующие модули:

* модуль Intel Composer XE 2013 (compilers/composer\_xe/2011\_sp1.6.233);
* модуль Intel MPI (parallel/mpi.intel/4.0.3.008)

Для просмотра подгруженных модулей необходимо выполнить следующую команду:

$ module list

Currently Loaded Modulefiles:

1) compilers/composer\_xe/2011\_sp1.6.233 2) parallel/mpi.intel/4.0.3.008

### Список доступных модулей

Для просмотра списка доступных модулей необходимо выполнить команду:

bash-4.1$ module avail

### Выбор компилятора

Для того чтобы подгрузить компилятор C++ (по умолчанию − Intel) необходимо выполнить команду:

$ module load compilers/cplusplus

$ module list

Currently Loaded Modulefiles:

1) modules 2) compilers/cplusplus/intel/12.1.0

$ $CC –version

icc (ICC) 12.1.0 20110811

Copyright (C) 1985-2011 Intel Corporation. All rights reserved.

### Замена компилятора

Для того чтобы заменить Intel C/C++ на GNU необходимо выполнить команду:

$ module switch compilers/cplusplus/intel/13.1.1 compilers/cplusplus/gnu/4.4.6

### Использование библиотек

Для выбора библиотеки Intel MPI необходимо выполнить команду:

$ module load parallel/mpi.intel

$ module list

Currently Loaded Modulefiles:

1) compilers/cplusplus/intel/12.1.0 2) parallel/mpi.intel/4.0.3.008

$ mpiexec --version

Intel(R) MPI Library for Linux, 64-bit applications, Version 4.0 Update 3 Build 20110824

Copyright (C) 2003-2011 Intel Corporation. All rights reserved.

## Планировщик задач (SLURM)

### Состояние планировщика задач

Для получения информации о состоянии и использовании очереди задачи необходимо выполнить команду “**sinfo**”:

# sinfo

PARTITION AVAIL TIMELIMIT NODES STATE NODELIST

work\* up 5-00:00:00 10 drain\* node[049,053,105-112]

work\* up 5-00:00:00 4 down\* node[085,093,163,197]

work\* up 5-00:00:00 1 comp node018

work\* up 5-00:00:00 19 drain node[007,024,028-029,032,035-036,038,046-047,052,056,060,063,067,084,094,102,104]

work\* up 5-00:00:00 126 alloc node[001-006,008-017,019-023,025-027,030-031,033-034,037,039-045,048,050,055,057-059,061-062,064,066,069-072,074-081,083,086-092,095-101,103,113-114,116-118,120-126,128-137,139-150,154,158,160,162,211-224]

work\* up 5-00:00:00 59 idle node[051,068,115,119,138,151-153,155-157,159,161,164-196,198-210]

work\* up 5-00:00:00 5 down node[054,065,073,082,127]

Дополнительные опции:

**-n <nodelist>** – статус по конкретным группам вычислительных узлов;

**-p <partition name>** – статус по узлам конкретного раздела.

### Просмотр состояния очереди задач

Для получения списка активных задач необходимо использовать команду “**squeue**”:

user@head:~$ squeue

JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)

3807 work bash user R 1:23 2 node[075-076]

Дополнительные опции:

**-w <nodelist>** – статус по конкретным группам вычислительных узлов;

**-p <partition name>** – статус по узлам конкретного раздела.

### Запуск MPI-задач

MPI-задачи на кластере могут быть запущены в пакетном или интерактивном режимах.

#### Пакетный режим запуска задачи

Пользователь создаёт скрипт для запуска задачи и использует утилиту “**sbatch**”. Задача отправляется в очередь и будет выполнена, как только будут доступны запрошенные ресурсы. Вывод результата выполнения задачи будет записан в файл slurm-<номер задачи в очереди>.out в директории, откуда осуществлялся запуск задачи.

Основные ключи утилиты “**sbatch**”:

–N, --nodes указывает количество необходимых узлов

–n, --ntasks общее количество запущенных процессов

--ntasks-per-node задает количество процессов запускаемых на каждом вычислительном узле

–t, --time время доступности выделенных ВУ (в минутах)

-p, --partition выделение ресурсов в указанной партиции

--gres Выделение нод с указанными ресурсами

В качестве параметра утилите “**sbatch**” передаётся исполняемый скрипт, который будет запущен на первом из выделенных вычислительных узлов. Внутри скрипта осуществляется запуск задачи с помощью утилит “**srun**” или “**mpiexec.hydra**”.

Утилита “**sbatch**” позволяет задавать все свои опции внутри скрипта запуска. Формат задания опций:

#SBATCH <опция>

#### Запуск с помощью утилиты “srun”

Необходимо, чтобы был загружен модуль *“***launcher/slurm**”:

module load launcher/slurm

Скрипт для пакетного запуска X экземпляров MPI-программы hello\_mpi на каждом из Y вычислительных узлов с использованием “srun”:

**hello\_hydra.sh**

#!/bin/sh

srun --ntasks-per-node X -n X\*Y ./hello\_mpi

**Запуск скрипта hello\_hydra.sh**

bash-4.1$ module load launcher/slurm

bash-4.1$ sbatch -N Y ./hello\_hydra.sh

Submitted batch job 135

Вывод скрипта будет записан в файл slurm-<номер задачи в очереди>.out в директории, откуда осуществлялся запуск задачи.

Пример скрипта для запуска X экземпляров MPI-задачи на каждом из Y вычислительных узлов (ограничение по времени 1 минута) с помощью утилиты “srun” и заданием опций внутри скрипта:

**srun\_template.sh.**

#!/bin/sh

# Set timelimit

#SBATCH --time=1

# Use partition

#SBATCH --partition=work

# Number of allocated nodes

#SBATCH --nodes=Y

# Number of tasks per node

#SBATCH --ntasks-per-node=X

# Number of tasks

#SBATCH --ntasks=X\*Y

srun ./hello\_sym

Пример скрипта расположен в /opt/basis/scripts/srun\_template.sh.

**Запуск скрипта srun\_template.sh**

bash-4.1$ sbatch ./srun\_template.sh

#### Запуск с помощью утилиты “mpiexec.hydra”

Необходимо, чтобы модуль «**launcher/intel**» был загружен.

Пример для пакетного запуска X экземпляров MPI-программы hello\_mpi на каждом из Y вычислительных узлов с использованием “**mpiexec.hydra**”:

**hello\_hydra.sh**

#!/bin/sh

mpiexec.hydra –perhost X -n X\*Y ./hello\_mpi

**Запуск скрипта hello\_hydra.sh**

bash-4.1$ module load launcher/intel

bash-4.1$ sbatch -N Y ./hello\_hydra.sh

Submitted batch job 135

Вывод скрипта будет записан в файл slurm-<номер задачи в очереди>.out в директории, откуда осуществлялся запуск задачи.

Пример скрипта для запуска X экземпляров MPI-задачи на каждом из Y вычислительных узлах (ограничение по времени 1 минута) с помощью утилиты “**mpiexec.hydra**” и заданием опций внутри скрипта:

**mpiexec\_template.sh**

#!/bin/sh

# Set timelimit

#SBATCH --time=1

# Use partition

#SBATCH --partition=work

# Number of allocated nodes

#SBATCH --nodes=Y

# Number of tasks per node

#SBATCH --ntasks-per-node=X

mpiexec.hydra -perhost $SLURM\_NTASKS\_PER\_NODE -n \ $(($SLURM\_JOB\_NUM\_NODES\*$SLURM\_NTASKS\_PER\_NODE)) ./hello\_sym

Пример скрипта расположен в /opt/basis/scripts/mpiexec\_template.sh:

**Запуск скрипта mpiexec\_template.sh**

bash-4.1$ sbatch ./mpiexec\_template.sh

#### Интерактивный режим запуска задачи

Пользователь либо использует для запуска задачи утилиту “**srun**”, либо самостоятельно выделяет необходимое количество вычислительных узлов и запускает задачу с помощью утилиты “**mpiexec.hydra**” или “**srun**”. Вывод результата выполнения задачи будет произведён в консоль.

Основные ключи утилиты “**srun**”:

–N, --nodes указывает количество необходимых узлов

--ntasks-per-node задает количество процессов запускаемых на каждом вычислительном узле

–n, --ntasks общее количество запущенных процессов

–t, --time время доступности выделенных ВУ (в минутах)

-p, --partition выделение ресурсов в указанной партиции

--gres Выделение нод с указанными ресурсами

Пример запуска X экземпляров MPI-задачи на Y вычислительных узлах:

1. Загрузить при необходимости модуль “launcher/slurm” командой:

module load launcher/slurm

1. Использовать команду “srun” для запуска MPI-задачи:

srun –N Y --ntasks-per-node=X -n X\*Y ./hello\_mpi

После запуска задачи результат выполнения будет выведен на экран:

Hello, world, I am 6 of 6

Hello, world, I am 5 of 6

Hello, world, I am 4 of 6

Hello, world, I am 3 of 6

Hello, world, I am 2 of 6

Hello, world, I am 1 of 6

#### Интерактивный запуск MPI-задачи (“mpiexec.hydra”)

Основные ключи утилиты “**mpiexec.hydra**”:

-perhost <n> Запустить n процессов на каждом узле

-n Общее количество процессов

Для получения информации обо всех доступных опциях утилиты “mpiexec.hydra” необходимо использовать команду:

mpiexec.hydra --help

Основные ключи утилиты “salloc”:

–N, --nodes указывает количество необходимых узлов

–t, --time время доступности выделенных ВУ (в минутах)

-p, --partition выделение ресурсов в указанной партиции

--gres Выделение нод с указанными ресурсами

Для получения информации обо всех доступных опциях утилиты “**salloc**” необходимо использовать команду:

man salloc

Пример запуска X экземпляров MPI-задачи на Y хостах:

* Выделить Y вычислительных узлов из кластера:

salloc –N Y

При наличии ресурсов будет выведено сообщение об успешном выделении вычислительных узлов:

salloc: Granted job allocation 302

* Загрузить при необходимости модуль **launcher/intel** командой:

module load launcher/intel

* Использовать команду “**mpiexec.hydra**” для запуска MPI-задачи:

mpiexec.hydra -perhost X -n X\*Y ./hello\_mpi\_new

После запуска задачи результат выполнения будет выведен на экран.

Hello, world, I am 6 of 6

Hello, world, I am 5 of 6

Hello, world, I am 4 of 6

Hello, world, I am 3 of 6

Hello, world, I am 2 of 6

Hello, world, I am 1 of 6

### Освобождение выделенных узлов

Для освобождения узлов, выделенных командой “**salloc**”,необходимо воспользоваться комбинацией клавиш “Ctrl+D” или командой “**exit**”.

### Удаление задачи

Для удаления задачи используется команда:

scancel <JobID>

Для удаления запущенной задачи необходимо знать её идентификатор (ID). Для удаления задачи с известным ID выполните команду:

user@head:~$ scancel 3807

salloc: Job allocation 3807 has been revoked.

### Переменные окружения, используемые планировщиком SLURM

При выделении ресурсов или запуске задач планировщик автоматически прописывает в переменные окружения актуальную служебную информацию. Ниже приведён список этих переменных с описанием:

* **$SLURM\_JOB\_CPUS\_PER\_NODE** – количество процессорных ядер, которое может быть использовано задачей на каждом выделенном вычислительном узле (по умолчанию, 32);
* **$SLURM\_JOBID** – идентификатор текущей аллокации ресурсов;
* **$SLURM\_JOB\_ID** – аналогично $SLURM\_JOBID;
* **$SLURM\_JOB\_NODELIST** – список выделенных вычислительных узлов;
* **$SLURM\_JOB\_NUM\_NODES** – количество выделенных вычислительных узлов;
* **$SLURM\_NNODES** – аналогично $SLURM\_JOB\_NUM\_NODES;
* **$SLURM\_NODE\_ALIASES** – псевдонимы выделенных вычислительных узлов (не используется в данной инсталляции);
* **$SLURM\_NODELIST** – аналогично $SLURM\_JOB\_NODELIST;
* **$SLURM\_SUBMIT\_DIR** – путь до директории, в которой находился текущий пользователь в момент выделения ресурсов;
* **$SLURM\_TASKS\_PER\_NODE** – количество процессов, которые могут быть одновременно запущены на одном вычислительном узле (по умолчанию равно количеству ядер, в данной конфигурации − 32).

## Общие ресурсы

Вычислительная система имеет выделенную директорию /share для общего доступа. В данной директории расположены следующие ресурсы:

* Modules – файлы конфигурации пакета Environmental Modules;
* Parallel – пакеты и библиотеки для параллельных вычислений;
* Intel – директория с установленными на вычислительную систему пакетами от компании Intel (Cluster Studio и т.д.);
* различные компоненты, устанавливаемые администратором: библиотеки, компиляторы, прикладные пакеты.

# Список сокращений

|  |  |
| --- | --- |
| Термин | Описание |
| SSH | Secure SHell — «безопасная оболочка» — сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование соединений (например, для передачи файлов). |
| SFTP | SSH File Transfer Protocol — протокол прикладного уровня, предназначенный для копирования и выполнения других операций с файлами поверх надёжного и безопасного соединения. |
| SLURM | Менеджер ресурсов с открытым кодом для вычислительных систем под управлением Linux. |

**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Изм | Номера листов (страниц) | | | | Всего листов (страниц) в док. | № разреш.  документа | Подпись | Дата | Примечание |
| Измененных | Замененных | Новых | Аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |