ЗАО «РСК Технологии»
121170, Москва, Кутузовский пр-кт 36, стр.23

МОДЕРНИЗАЦИЯ СУПЕРКОМПЬЮТЕРА МОДЕЛИ МВС-10П В ЧАСТИ РАЗДЕЛА НА ОСНОВНЫХ УНИВЕРСАЛЬНЫХ ПРОЦЕССОРАХ (ОП)

Программное обеспечение «РСК БАЗИС»

Инструкция пользователя

69573991.031.МСЦ.И3.2

МОСКВА
2017

СОДЕРЖАНИЕ

АННОТАЦИЯ

Данный документ является частью комплекта документацииМодернизация суперкомпьютера модели МВС-10П в части раздела на основных универсальных процессорах (ОП) (далее Комплект расширения), разрабатываемого на основании контракта №161202 от 23 декабря 2016 года, заключенного между МСЦ РАН (далее Заказчик) и ЗАО «РСК Технологии» (далее Исполнитель).

Данный документ представляет собой руководство пользователя программного обеспечения РСК «БазИС»(RSC BasIS, BasIS, РСК БАЗИС, БАЗИС)версии 2.1, разработанного ЗАО «РСК Технологии».

ГЛОССАРИЙ

ВК	Вычислительный комплекс
ПО	Программное обеспечение
Git	Система контроля версий
CLI	Интерфейс командной строки
BMC	Baseboard Management Controller — встроенный контроллер управления
CPU	Central Processing Unit — центральный процессор.
DHCP	DynamicHostConfigurationProtocol— протокол динамического конфигурирования узла, автоматическое получение сетевых настроек
Infiniband	Высокоскоростная коммутируемая последовательная шина
IPMI	Intelligent Platform Management Interface — интеллектуальный интерфейс управления платформой
LAN	LocalAreaNetwork— локальная компьютерная сеть
LDAP	Упрощённый протокол доступа к каталогам, протокол LDAP
MPI	MessagePassingInterface— программный интерфейс (API) для передачи информации, который позволяет обмениваться сообщениями между процессами, выполняющими одну задачу
NIC	Network Interface Controller — сетевой контроллер
RAID	Redundant array of independent disks — избыточный **массив** независимых **жёстких дисков**
SEL	SystemEventLog— аппаратный журнал системы
SFTP	SSHFileTransferProtocol— 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
SSH	SecureShell— «безопасная оболочка» — сетевой протокол прикладного уровня, позволяющий производить удалённое

	управление операционной системой и туннелирование соединений (например, для передачи файлов)
TCP/IP	Протокол управления передачей / межсетевой протокол
xCAT	Extreme Cloud Administration Toolkit

ПРИМЕРЫ СТИЛЕЙ, ИСПОЛЬЗУЕМЫХ В ДОКУМЕНТЕ

Вычислитель	Термин, наименование
$HOME/.ssh	Путь к файлу
dumpxCATdb	Команда
# /etc/init.d/xcatd stop	# - команда, выполняемая от суперпользвоателя (root)
$ pwd	$ - команда, выполняемая от обычного пользователя.

1 ВВЕДЕНИЕ

Полное наименование: Программное обеспечение РСК «БазИС» (RSC BasIS, BasIS, РСК БАЗИС, БАЗИС, ПО).

Назначение:ПО управления «РСК БАЗИС» предназначено для обеспечения работоспособности различных Вычислительных комплексов, разрабатываемых ЗАО «РСК Технологии».

Вычислительный Комплекс – это совокупность аппаратных и программных,интегрированных для решения вычислительных задач.

Более подробные сведения о назначении приведены в документе «Общее описание системы».

Общие сведения: ПО управления Вычислительным комплексом “РСК БазИС” представляет собой набор программных компонент, интегрированных друг с другом для решения задач управления ВК.

Вычислительный Комплекс – это совокупность аппаратных и программных компонент:

- Аппаратные компоненты включают:
  - Инфраструктурные модули.
  - Сетевые модули.
  - Компоненты хранения данных.
  - Вычислительные сервера.
  - Сервера управления
  - Сервера доступа.
- Программные компоненты включают:
  - ПО управления «РСК БАЗИС».
  - Прикладное ПО.

Важно! Состав аппаратных и программных компонентов может отличаться от приведенного выше и зависит от конфигурации Вычислительного комплекса.

Инфраструктурные модули обеспечивают базовые требования функционирования ВК – охлаждение, электропитание и прочее.

*Сетевые моду*ли – сетевые коммутаторы, маршрутизаторы и кабельная подсистема, обеспечивающие внутренние и внешние связи элементов комплекса.

Компоненты хранения данных – внешние дисковые накопители и сервера доступа к ним, для организации централизованных ресурсов хранения.

Вычислительные сервера – ключевой компонент комплекса, выполняющий необходимые конечному пользователю вычисления.

Сервера управления – сервера, которые производят координацию всех подсистем ВК. Содержат в своем составе головной управляющий сервер, на который происходит первоначальная установка ПО управления.

Также могут содержать произвольное количество дополнительных серверов управления, в зависимости от размера ВК и требуемых функций.

Сервера доступа – сервера, на который происходит первоначальный вход пользователя. Также являются серверами, откуда происходит диспетчеризация задач для пакетной обработки.

Прикладное программное обеспечение – ПО, запускаемое конечными пользователями с целью решения прикладных задач.

2 ОПИСАНИЕ ВЫЧИСЛИТЕЛЬНОГО КОМПЛЕКСА

2.1 Структура Вычислительного комплекса и назначение его частей

Вычислительный Комплекс состоит из совокупности аппаратных и программных компонент. В их число входят:

- Аппаратные компоненты
  - Инфраструктурные компоненты
  - Сетевые компоненты
  - Компоненты хранения данных
  - Вычислительные сервера
  - Сервера управления и доступа
- Программные компоненты
  - ПО управления ВК
  - Прикладное ПО

2.1.1 Аппаратные компоненты

Инфраструктурные компоненты обеспечивают базовые требования функционирование ВК – охлаждение, электропитание и прочее.

Сетевые компоненты – сетевые коммутаторы, маршрутизаторы и кабельная подсистема, обеспечивающие внутренние и внешние связи элементов комплекса.

Компоненты хранения данных – внешние дисковые накопители и сервера доступа к ним, для организации централизованных ресурсов хранения.

Вычислительные сервера – ключевой компонент комплекса, выполняющий необходимые конечному пользователю вычисления.

Сервера управления – сервера, которые производят координацию всех подсистем ВК. Содержат в своем составе головной управляющий сервер, на который происходит первоначальная установка ПО управления.

Также могут содержать произвольное количество дополнительных серверов управления, в зависимости от размера ВК и требуемых функций.

Сервера доступа – сервера, на который происходит первоначальный вход пользователя. Также являются серверами, откуда происходит диспетчеризация задач для пакетной обработки.

2.1.2 Программные компоненты

ПО управления Вычислительным комплексом “РСК БазИС” представляет собой программных компонент, интегрированных друг с другом для решения задач управления ВК.

Прикладное программное обеспечение – ПО, запускаемое конечными пользователями с целью решения прикладных задач.

2.2 Основные характеристики Вычислительного комплекса

Полный перечень входящих в Вычислительный комплекс компонент отписывается в документе «Общее описание системы».

3 ПОДГОТОВКА К РАБОТЕ

3.1 Требования к квалификации пользователя

Квалификация пользователя, допускаемого к эксплуатации Вычислительного комплекса, должна обеспечивать эффективное функционирование комплекса во всех заданных режимах.

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

Общая подготовка должна включать в себя получение навыков работы с программным обеспечением в объеме навыков пользователей Вычислительного комплекса.

Специальная подготовка должна включать в себя получение навыков работы с системным и прикладным обеспечением Вычислительного комплекса в объеме навыков его использования.

3.2 Общий принцип использования

Взаимодействие с Вычислительным комплексом происходит удаленно через использование консольного интерфейса.

Вычислительный комплекс спроектирован для мультипользовательской одновременной работы, поэтому для управления выделением ресурсов в рамках комплекса установлен планировщик ресурсов SLURM.

Политика работы Вычислительного комплекса подразумевает несколько этапов взаимодействия пользователя с системой:

1. Интерактивный вход пользователя на консоль сервера входа по протоколу ssh.
2. Получение доступа к вычислительным узлам через планировщик задач SLURM в двух вариантах: в интерактивном и пакетном режимах:
   2.1. В интерактивном режиме пользователь запрашивает у планировщика требуемое количество вычислительных узлов, после чего ожидает их выдачи. В случае успешного выделения узлов планировщиком (о чем система сообщит в консоли текущей сессии), пользователь может получить прямой ssh-доступ к выданным узлом на запрошенное время;
   2.2. В пакетном режиме запуск осуществляется с помощью сценария, представляющего собойshell-скрипт. Планировщик размещает сценарий в очередь планирования и сам принимает решение о дате и месте ее запуска.Скрипт будет запущен на первом из выделенных узлов.

В любом случае каждый узел выделяется пользователю в единоличное пользование в рамках конкретной задачи.

3.3 Получение реквизитов для удаленного доступа

3.3.1 Удаленный доступ и авторизация

Необходимые параметры и настройки для обеспечения доступа предоставляет Оператор Вычислительного кластера.

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

Вместо пароля может использоваться авторизация по ssh-ключам, что является более безопасной схемой.

3.3.2 Удаленный доступ по паролю

Для доступа по паролю необходимо воспользоваться утилитой ssh:

```
$ ssh rsc@login
```

```
rsc@login's password:
Last login: Tue Nov  3 14:19:41 2015
[rsc@login ~]$
```

3.3.3 Управление ssh-ключами

Для авторизации по ключу пользователю необходимо иметь предварительно подготовленную пару, состоящую из публичного и приватного ssh-ключей.

На платформе Windows для этого можно воспользоваться утилитой puttygen.exe, на платформе unix – утилитой ssh-keygen.

Сначала необходимо выложить свой публичный ключ в файл ключей, находящейся в своей пользовательской директории на кластере по адресу ~/.ssh/authorized_keys. Для этого можно воспользоваться либо авторизацией по паролю, либо попросить выполнить эту операцию Оператора.

Для самостоятельного добавления ключа воспользуйтесь утилитой *ssh-copy-id*:

```
$ ssh-copy-id rsc@login
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s),  to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you  are prompted now it is to install the new keys
rsc@login's password:
Number of key(s) added: 1
Now try logging into the machine, with:   "ssh 'rsc@login'"
and check to make sure that only the key(s) you wanted were added.
```

После этого становится возможно зайти на сервер доступа без указания пароля:

```
$ ssh rsc@login
Last login: Tue Nov  3 15:50:57 2015 from 89.207.88.26
[rsc@login ~]$
```

3.3.4 Доступ на внутренние сервера комплекса

При первом входе пользователя на Вычислительный комплекс автоматически генерируется ssh-ключ, предназначенный для дальнейшего доступа на вычислительные узлы. Данный ключ хранится в пользовательской директории**.ssh.**

4 СТРУКТУРА ДИРЕКТОРИЙ

4.1 Пользовательская директория

Директория пользователя находится по адресу **/home/<имя пользователя>.**

4.2 Общие директории

На кластере существуют следующие общие директории:

Путь	Назначение
/opt/basis	Общие системные файлы
/opt/software	Директория для установки прикладного программного обеспечения

4.3 Загрузка и выгрузка данных

Для загрузки данных на кластер необходимо использовать любой клиент с поддержкой протокола SSH.

Для Windows можно использовать клиент WinSCP(**https://winscp.net/)**, для unix систем встроенный клиент scp.

5 ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

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

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 EnvironmentModules позволяет гибко управлять разными версиями приложений.

Пакет состоит из модулей, описанных в modulefiles, которые доступны в директории **/opt/basis/modules**, а также системных директориях **/etc/modulefiles** и **/usr/share/Modules/modulefiles**.

Пользователь может создавать свой набор пользовательских файлов в своей домашней директории.

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 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.

Для просмотра подгруженных модулей необходимо выполнить следующую команду:

```
$ module list
Currently Loaded Modulefiles:
  1)  mpi/openmpi-x86_64
```

Для просмотра списка доступных модулей необходимо выполнить команду:

```
$ module avail

--------------------------- /usr/share/Modules/modulefiles ----------------------
dot         module-git  module-info modules     null        use.own

--------------------------- /etc/modulefiles ------------------------------------
mpi/openmpi-x86_64
```

Для подгрузки (или выгрузки) модуля необходимо выполнить команду:

```
$ module load mpi/openmpi-x86_64
```

или

```
$ module unload mpi/openmpi-x86_64
```

5.1 Управление списком загружаемых по умолчанию модулей

При первом входе пользователя системы в его домашнем каталоге создается файл **.modules**. Списком загружаемых модулей можно управлять с помощью редактирования файла **$HOME/.modules** или ключей команды module.

> **Важно! Из-за особенностей логики работы компонента, для корректного функционирования механизма автозагрузки модулей в списке загружаемых модулей должен присутствовать модуль null, который не выполняет никаких действий**

Просмотр текущего списка осуществляется с помощью следующей команды:

```
$ module initlist
```

Например:

```
$ module initlist
bash initialization file $HOME/.modules loads modules:
null
```

Добавление модуля в автозагрузку осуществляется с помощью следующей команды:

```
$ module initadd <modulefile>
```

Например:

```
$ module initadd compilers/cplusplus/gnu/4.4.6
```

Удаление модуля из автозагрузки осуществляется с помощью следующей команды:

```
$ module initrm <modulefile>
```

Например:

```
$ module initrm compilers/composer_xe/2013_sp1
Removedcompilers/composer_xe/2013_sp1
```

Более подробную информацию о использовании пакета Environment Modules можно найти в разделе ссылочной документации.

5.2 Поставляемое ПО

5.2.1 Компиляторы, библиотеки

Для компиляции прикладного программного обеспечения на кластере установлен стандартный набор библиотек и компиляторов из пакета IntelParallelStudioClusterEdition.

Для его использования необходимо подгрузить модуль *intel*

5.2.2 MPI runtime

Для работы параллельных MPI-приложений на кластере установлена библиотека MPI из пакета IntelParallelStudioClusterEdition.

Для ее подгрузки необходимо подгрузить модуль *parallel/mpi.intel*

5.2.3 CUDA

Данный раздел актуален для кластеров, содержащих GPU Nvidia.

Для поддержки графических ускорителей Nvidia установлен набор библиотек CUDA.

Для ее подгрузки необходимо подгрузить модуль *nvidia/cuda-7.5.*

6 ЗАПУСК ЗАДАЧ

Для выполнения задач на кластере необходимо предварительно скомпилированное приложение запустить с использованием инструментов установленной на кластере библиотеки MPI.

6.1 Компиляция задач

Если приложение поставляется в исходном коде, то тогда необходимо осуществить предварительную сборку согласно инструкции по сборке данного ПО.

6.2 Описание планировщика задач

Для управления ресурсами на кластере установлен планировщик SLURM. Все взаимодействие с вычислительными ресурсами кластера осуществляется только через него.

Основные инструменты планировщика задач:

- *sinfo* – просмотр статуса кластера
- *squeue* – просмотр очереди задач
- *salloc* – интерактивное выделение вычислительных узлов
- *sbatch* – пакетный запуск задач

6.3 Просмотр статуса кластера

На кластере может быть предусмотрено несколько очередей (разделов) для запуска задач.

Для получения информации о состоянии и использовании очереди задач необходимо выполнить команду *sinfo*:

```
[rsc@login ~]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
compute*     up 14-00:00:0      5  alloc node[2-6]
```

Дополнительные опции:

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

-p<partitionname> – статус по узлам конкретного раздела.

6.4 Просмотр очереди задач

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

```
[rsc@login ~]$ squeue
JOBID  PARTITION     NAME             USER  ST       TIME  NODES NODELIST(REASON)
67             compute        huge_tem        rsc      R       33:30   5        node[2-6]
```

Дополнительные опции:

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

-p<partitionname> – статус по узлам конкретного раздела.

6.5 Запуск MPI-задач

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

Пакетный режим – стандартный режим работы на кластере. Интерактивный режим чаще используется для отладки работы приложений.

Перед запуском задачи, используя EnviromentalModules, выберите:

- необходимую библиотеку mpi (по умолчанию *parallel/mpi.intel*)
- способ запуска задачи (для общих случаев рекомендуется *launcher/slurm*)

Способ запуска определяет связь между планировщиком задач и библиотекой MPI. В поставку включены два сценария:

- *launcher/slurm* – для общих случаев
- *launcher/mpiexec*– когда требуется ручная подстройка параметров mpiexec

6.5.1 Пакетный режим

Для работы в пакетном режиме пользователю необходимо сначала создать исполняемый сценарий, в котором описано правило запуска задачи.

Затем данный сценарий (в виде скрипта с выставленным битом выполнения) передается утилите sbatch в качестве параметра. Он будет запущен на первом из выделенных вычислительных узлов.

Обратите внимание, что в сценарии запуска необходимо указать требуемые модули пакета EnvironmentModules для загрузки.

Пример содержимого сценария:

```
#!/bin/sh

# Set timelimit
#SBATCH --time=1-0:0

# Number of allocated nodes
#SBATCH --nodes=5

# Number of tasks per node
#SBATCH --ntasks-per-node=10

# Enable Environment Modules
source /etc/profile.d/modules-basis.sh

# Load launcher module
module load launcher/slurm

BINARY=/opt/basis/scripts/hello_sym.mic

srun $BINARY
```

После этого необходимо добавить задачу в общую очередь задач, используя утилиту *sbatch*:

```
[rsc@login ~]$ sbatch mpirun_template.sh
Submitted batch job 27
```

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

6.5.2 Интерактивный режим запуска задачи

В интерактивном режиме запуска задачи пользователь запускает приложение самостоятельно, при этом управление терминальной сессией переходит к задаче и пользователь не может выполнять другие действия.

Для остановки задачи в интерактивном режиме можно использовать комбинацию клавиш Ctrl-C.

Для интерактивной работы с узлами используется утилита salloc. Ниже приведен типовой сценарий работы в интерактивном режиме:

```
[rsc@login ~]$ module load launcher/slurm parallel/mpi.intel
[rsc@login ~]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
compute*     up 14-00:00:0      5   idle node[2-6]
[rsc@login ~]$ salloc -N 5
salloc: Granted job allocation 26
[rsc@login ~]$ srun ./hello_sym.mic
```

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

6.5.3 Запуск задач на модулях PetaStream

Так как узлы PetaStream являются отдельными узлами планирования, то запуск задач на них также осуществляется с помощью библиотеки MPI и утилит SLURM, с учетом нескольких требований:

- Пользователи не должны модифицировать переменные окружения:
  - I_MPI_DEVICE (не должна быть установлена)
  - I_MPI_FABRICS (должна быть автоматически установлена в ‘dapl’)
  - I_MPI_DEPL_PROVIDER (должна быть установлена автоматически в соответствии с местоположением узла в модуле)
- Должен быть подгружен модуль работы с IntelXeonPhi:
  - *mic_pmi* в случае использования *launcher/slurm*(см. раздел Запуск MPI-задач)
  - *mic* в случае использования *launcher/mpiexec*

6.5.3.1 Пакетный режим

Шаблон сценария:

```
#!/bin/sh

# Set timelimit
#SBATCH --time=30

# Use partition
#SBATCH --partition=mic

# Number of allocated nodes
#SBATCH --nodes=4

# Number of tasks per node
#SBATCH --ntasks-per-node=2

# Enable Environment Modules
source /etc/profile.d/modules-basis.sh

# Load launcher & mic modules
module load mic_pmi launcher/slurm

# Enable MPI Debug
export I_MPI_DEBUG=${I_MPI_DEBUG:-0}

BINARY=/opt/basis/scripts/hello_sym.mic

srun $BINARY
```

Запуск задачи:

```
user15@login:~
$ sbatch /opt/basis/scripts/mpiexec_template.sh
Submitted batch job 466
```

Результат выполнения будет сохранен в файле вида **slurm-${SLURM_JOB_ID}.out**. Файл будет сохранен в текущей директории, откуда производился запуск:

```
user15@login:~
$ cat slurm-466.out
Master rank     0 (122 threads) of    16 with PID 16883 is running on ps-mic0
 Slave rank     1 (122 threads) of    16 with PID 16884 is running on ps-mic0
 Slave rank     2 (122 threads) of    16 with PID 12668 is running on ps-mic1
 Slave rank     3 (122 threads) of    16 with PID 12669 is running on ps-mic1
 Slave rank     4 (122 threads) of    16 with PID 12572 is running on ps-mic2
 Slave rank     5 (122 threads) of    16 with PID 12573 is running on ps-mic2
 Slave rank     6 (122 threads) of    16 with PID 12273 is running on ps-mic3
 Slave rank     7 (122 threads) of    16 with PID 12274 is running on ps-mic3
 Slave rank     8 (122 threads) of    16 with PID 11947 is running on ps-mic4
```

6.5.3.2 Интерактивный режим

Подгрузка модулей:

```
$ module load mic_pmi launcher/slurm parallel/mpi.intel
```

Выделение узлов:

```
$ salloc -N 8 --ntasks-per-node=1
salloc: Pending job allocation 199
salloc: job 199 queued and waiting for resources
salloc: job 199 has been allocated resources
salloc: Granted job allocation 199
```

Запуск задачи:

```
$ srun  /opt/basis/scripts/hello_sym.mic
Master rank     0 (244 threads) of     8 with PID  5579 is running on ps-mic8
 Slave rank     1 (244 threads) of     8 with PID  5544 is running on ps-mic9
 Slave rank     2 (244 threads) of     8 with PID  5538 is running on ps-mic10
 Slave rank     3 (244 threads) of     8 with PID  5536 is running on ps-mic11
 Slave rank     4 (244 threads) of     8 with PID  5539 is running on ps-mic12
 Slave rank     5 (244 threads) of     8 with PID  5535 is running on ps-mic13
 Slave rank     6 (244 threads) of     8 with PID  5541 is running on ps-mic14
 Slave rank     7 (244 threads) of     8 with PID  5541 is running on ps-mic15
```

6.6 Управление задачей

6.6.1 Получение подробной информации о задаче

Для получения подробной информации о задаче необходимо воспользоваться утилитой *scontrol*

6.6.2 Удаление задачи

Для удаления задачи используется команда *scancel.* Для удаления запущенной задачи необходимо знать её идентификатор (ID).

```
[rsc@login ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                67   compute huge_tem      rsc  R    1:14:58      5 node[2-6]
[rsc@login ~]$ scancel 67
[rsc@login ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
```

6.6.3 Переменные окружения SLURM

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

- **$SLURM_JOB_CPUS_PER_NODE** – количество процессорных ядер, которое может быть использовано задачей на каждом выделенном вычислительном узле;
- **$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;
- **$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

7 ТИПИЧНЫЕ ПРОБЛЕМЫ И ПУТИ ИХ РЕШЕНИЯ

7.1 Обращение в службу технической поддержки

Для обращения в техническую поддержку РСК необходимо открыть обращение в системе заявок.

Открывать обращения в ней можно, отправив электронное письмо с описанием проблемы по адресу rt@rsc-tech.ru

7.1.1 Порядок обращения в службу технической поддержки

При возникновении проблем необходимо придерживаться следующих рекомендаций, которые помогут РСК оперативно на них реагировать:

1. Каждую проблему формулировать в отдельном запросе. Это поможет РСК вести тщательное исследование и решение проблемы до конца, что было бы затруднительно, если в одном запросе указано несколько разных проблем с разной степенью детализации.
2. К каждой проблеме необходимо иметь алгоритм ее воспроизведения, позволяющий повторить поведение системы вплоть до получения ошибки (например, команда запуска и сообщение об ошибке).
3. К каждой проблеме прикладывать изначальные условия для запуска, а именно:
   - Имя пользователя
   - Рабочая директория для запуска
   - Точная команда запуска
   - Подгруженные модули (комманда modulelist)
   - Настройки среды окружения (комманда env)
   - Возникающая ошибка (здесь может быть содержание в произвольной форме, описывающее поведение системы, появляющиеся сообщения или файлы журналов запуска задачи, пути к ним и их содержание)

8 СЫЛОЧНАЯ ДОКУМЕНТАЦИЯ

1. xCat
   - http://xcat.sourceforge.net
2. Puppet
   - http://docs.puppetlabs.com/puppet/3/reference/
   - Компонент augeas: http://augeas.net
   - Компонент munge: https://code.google.com/p/munge/
3. Пакет Environmental Modules
   - http://modules.sourceforge.net/man/module.html
4. Планировщик задач SLURM
   - Основная документация: http://slurm.schedmd.com
5. Сервер Директорий пользователей 389 Direcotory Server
   - http://directory.fedoraproject.org

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

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