ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

ЛАБОРАТОРНАЯ РАБОТА № 1

«РАСПРЕДЕЛЁННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ»

Автор: С.Н. Мамойленко

Новосибирск - 2016

Оглавление

Цель работы

Лабораторная работа направлена на развитие следующих общекультурных, общепрофессиональных и профессиональных компетенций:

- способностью к профессиональной эксплуатации современного оборудования и приборов (ОК-8);
- владением методами и средствами получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе в глобальных компьютерных сетях (ОПК-5);
- способностью проектировать распределенные информационные системы, их компоненты и протоколы их взаимодействия (ПК-8);
- способностью к программной реализации распределенных информационных систем (ПК-13);

В результате изучения дисциплины должно быть сформировано понимание архитектуры распределенных высокопроизводительных вычислительных систем, основных принципов их эксплуатации и использования для переработки информации, а также получены базовые умения и навыки по проектированию и разработке распределённого программного обеспечения.

Теоретическое введение

1. Распределённые вычислительные системы

Современным инструментарием высокопроизводительной обработки информации являются распределённые вычислительные системы (РВС). В архитектурном смысле по классификации Майкла Флинна[1], такие системы относятся к классу MIMD (Multiple Instruction Multiple Data, Множественные потоки Команд и Множественные потоки Данных). Все ресурсы РВС, как программные, так и аппаратурные, являются логически и технические распределёнными. Это означает, что в таких системах нет ни одного общего (разделяемого) ресурса и все вычислительные элементы (узлы, машины) являются полноценными самостоятельными вычислительными устройствами, программным образом настраиваемые так, чтобы функционально представляться как единая система.

В списке Top-500 (Июнь 2016 года) [2] самых высокопроизводительных вычислительных систем мира к распределённым относится 431 система (что составляет 86,2%) и это количество неуклонно растет (см. рисунок 1). Следует отметить, что в указанном списке распределённые вычислительные системы называются *кластерами*. Этот термин стал популярным после выпуска в 1984 году компанией DEC своей системы VAXcluster [3]. Суть термина «кластер» отражает главную особенность распределённых вычислительных систем – совокупность независимых («слабо связанных») вычислительных элементов, которая используется для решения одной задачи. В дальнейшем укреплением этого термина в профессиональной сфере стал реализованный в 1995 году проект Beowulf [4], который дал начало эре создания коммерческих распределённых вычислительных систем из серийно выпускаемых персональных компьютеров и серверов и телекоммуникационных систем.

В нашей стране созданием высокопроизводительных средств обработки информации занимаются с 1950-х годов [5]. Работы в этой области проводились в Институте математики Сибирского отделения Академии наук СССР под руководством Э.В. Евреинова. В дальнейшем эти работы были продолжены научной школой член-корреспондента РАН В.Г. Хорошевского [6]. В основу создания отечественных РВС легла концепция вычислительных систем с программируемой структурой, сфор-

[1] https://ru.wikipedia.org/wiki/Таксономия_Флинна
[2] https://www.top500.org/statistics/list/
[3] https://en.wikipedia.org/wiki/VMScluster
[4] https://ru.wikipedia.org/wiki/Beowulf_(кластер)
[5] Две книги Хорошевского
[6] http://csc.sibsutis.ru/node/25

мированной к началу 1970-х годов. Первоначально таким средствам обработки информации было дано название – однородные вычислительные системы.

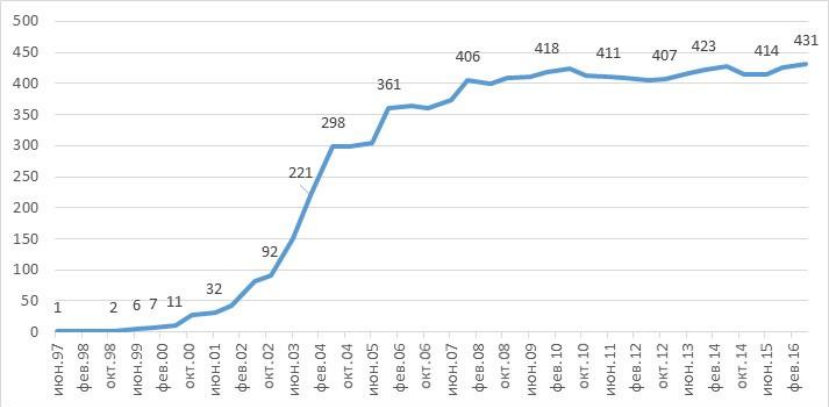


Рисунок 1 – Количество распределенных вычислительных систем в списках ТОП-500

Функционирование распределённых вычислительных систем базируется на модели коллектива вычислителей [..], в основу которой положены следующие принципы:

- параллельности и независимости функционирования структурных элементов;
- переменности макроструктуры, т.е. возможности программировать коммутацию каналов между узлами системы;
- однородности макроструктуры, гомогенности состава, т.е. программной совместимости всех вычислительных устройств системы, регулярности связей между ними.

Техническая реализация модели коллектива вычислителей основывается на следующих принципах:

- модульность – принцип, предопределяющий формирование вычислительной системы из унифицированных элементов, которые функциональны и конструктивно закончены, имеют средства сопряжения с другими элементами, разнообразие которых составляет полный набор. На практике принято такие модули называть *элементарными машинами*.
- близкодействие – принцип построения вычислительных систем, обуславливающий такую организацию информационных взаимодействий между модулями-вычислителями, при которой каждый из них может непосредственно (без «посредников») обмениваться информацией с весьма ограниченной частью модулей-вычислителей. Взаимодействие между удалёнными вычислителями (не имеющими непосредственной связи друг с другом) осуществляется только с помощью промежуточных вычислителей;
- асинхронность функционирования – все элементы РВС функционируют независимо и не используют какого-либо единого генератора тактовых импульсов;
- децентрализованность управления – каждый элемент РВС имеет собственную независимую систему управления, которая настраивается таким образом, чтобы, вступая во взаимодействие с другими управляющими системами, реализовывать общую функцию РВС;
- распределённость ресурсов – все технические и программные ресурсы распределены между модулями РВС.

Архитектурными принципами распределённых вычислительных систем являются:

- масштабируемость – способность к наращиванию и сокращению ресурсов, возможность варьирование производительности;
- универсальность – способность выполнения алгоритма (как последовательного, так и параллельного) решения любой задачи;
- производительность. В отличие от ЭВМ, построенных на модели вычислителя, вычислительные системы не имеют принципиальных ограничений в повышении производительности;
- реконфигурируемость – способность изменять количество вычислителей и структуру связей под решение конкретной задачи;
- надежность и живучесть – способность сохранять заданный уровень производительности на определённом временном отрезке и в условиях отказов элементов РВС;

- самоконтроль и самодиагностика – наличие инструментария выявления сбоев и отказов в РВС;

- технико-экономическая эффективность – позволяет достичь минимальных показателей стоимости владения РВС.

2. Пространственно-распределённая мультикластерная вычислительная система СибГУТИ

Центром параллельных вычислительных технологий СибГУТИ совместно с Лабораторией вычислительных систем Института физики полупроводников им. А.В. Ржанова Сибирского отделения Российской академии наук создана и развивается мультикластерная вычислительная система. Система (см. рисунок 2) объединяет 8 пространственно-распределенных кластерных вычислительных систем, причем кластеры A-F расположены в Центре параллельных вычислительных технологий СибГУТИ, а кластеры G, H – в Лаборатории вычислительных систем Института физики полупроводников им. А.В. Ржанова Сибирского отделения РАН.

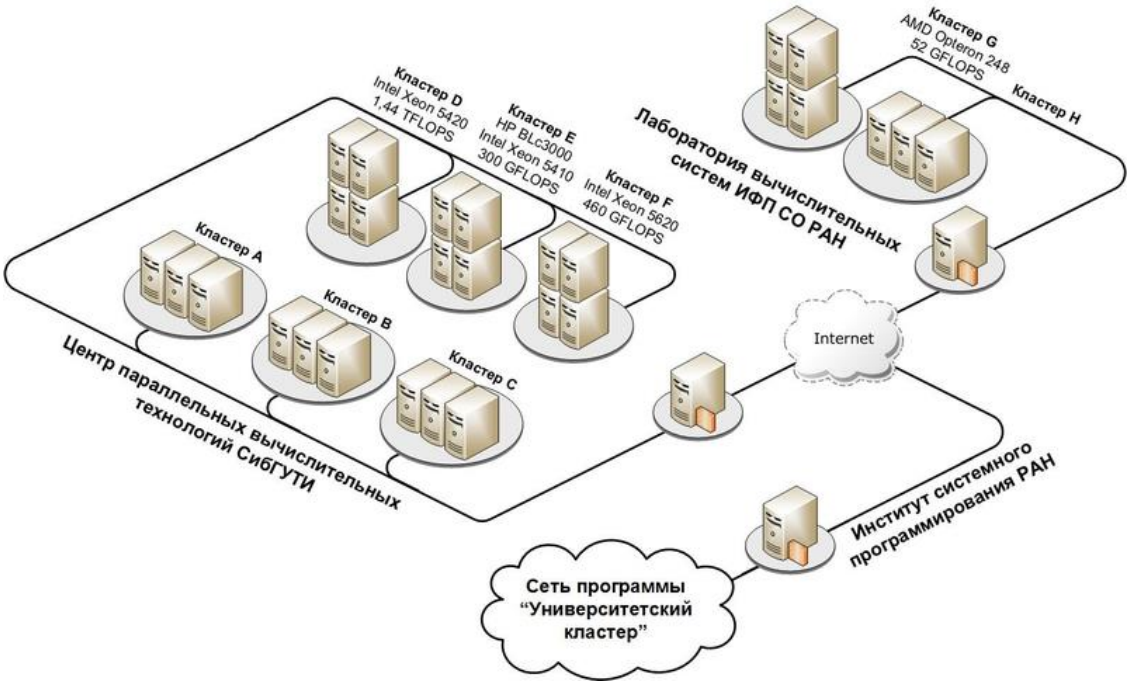


Рисунок 2 - Конфигурация пространственно-распределенной мультикластерной вычислительной системы (октябрь 2015 г.)

В рамках выполнения лабораторной работы будет использован один из сегментов этой системы – кластер Jet (сегмент D).

3. Архитектура распределённой вычислительной системы Jet

Распределённая вычислительная система (кластер) Jet состоит из 18 двухпроцессорных (Intel Xeon E5420, 4 ядра, 2,5 ГГц) вычислительных узлов и одного двухпроцессорного внешнего узла (Intel Xeon E5420, 4 ядра, 2,5 ГГц). Общее количество вычислительных ядер в РВС Jet составляет 144 штуки. Каждый вычислительный узел имеет 8 Гбайт оперативной памяти, внешний узел оснащён оперативной памятью объемом 16Гбайт.

Для хранения информации каждый вычислительный узел оснащён локальным жестким диском объемом 500 Гбайт, внешний узел имеет два диска по 500 Гбайт. Доступ к пользовательским данным внешнего узла внутри кластера предоставляется с использованием сетевой файловой системы NFS.

Коммуникационная среда кластера представлена двумя сетями стандарта Gigabit Ethernet (топология каждой сети – звезда с одним коммутатором). При этом одна сеть является сервисной, а вторая сеть используется в ходе выполнения вычислений.

Конструктивно распределённая вычислительная система Jet представлена в виде двух 19-ти дюймовых стоек, в которых размещены вычислительные узлы, телекоммуникационное оборудование и система бесперебойного питания.

Вычислительная система Jet функционирует под управлением операционной системы GNU/Linux (Fedora). Распределение ресурсов системы для решения задач осуществляется с использованием системы очередей OpenPBS/Torque. В качестве планировщика ресурсов используется стандартный FCFS планировщик PBS_sched.

Детальное описание архитектуры указанного кластера и его технические характеристики можно найти на сайте Центра параллельных вычислительных технологий [7].

4. Профессиональная эксплуатация распределённых вычислительных систем

Пользователи эксплуатируют распределённые вычислительные системы для решения своих профессиональных задач, заключающихся в выполнении некоторого программного обеспечения, реализующего определённый алгоритм переработки информации. Информация, необходимая для выполнения программного обеспечения (в некоторых случаях и само программное обеспечение) передается в систему посредствам сетевого взаимодействия. Аналогично получают результаты выполнения программного обеспечения.

4.1 Общие принципы взаимодействия с распределёнными вычислительных системами

Пользователи взаимодействуют с РВС в многопользовательском терминальном режиме. Это означает, что к ресурсам вычислительных систем одновременно (псевдопараллельно) предоставлен доступ нескольким пользователям, взаимодействующих с системой при помощи устройств ввода и вывода, совокупность которых, называется терминалом (см. рисунок 3). Терминалы соединены с распределённой системы с использованием телекоммуникационных каналов, в том числе глобальных компьютерных сетей.

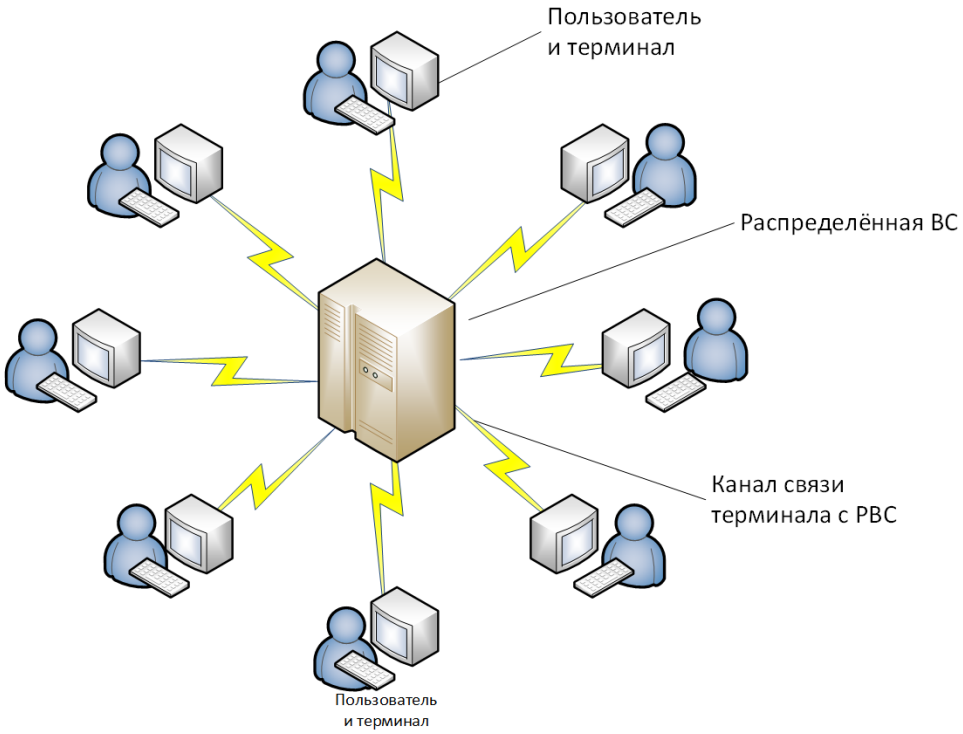


Рисунок 3 – Многопользовательский терминальный режим доступ
к ресурсам распределённых вычислительных систем

Терминалы могут иметь различную архитектуру и быть как специализированными, т.е. разработанными специально для предоставления доступа к вычислительным ресурсам и входящими в со-

[7] http://cpct.sibsutis.ru

став вычислительной системы, так и унифицированными, т.е. автономными электронно-вычислительными машинами, функционирующими самостоятельно и использующие распределённое программное обеспечение для организации доступа к ресурсам распределённой вычислительной системы (персональными компьютерами, мобильными устройствами, кассовыми терминалами и т.п.).

а) б)

Рисунок 4 – Пример унифицированных терминалов
(а – кассовый терминал, б – терминал оплаты)

Унифицированные терминалы эмулируют терминальный доступ с использованием распределённого программного обеспечения и специальных сетевых протоколов трансляции информации. В РВС, функционирующих под управлением систем GNU/Liunux или иных UNIX-подобных систем для терминального доступа в текстовом режиме чаще всего применяется сетевой протокол SSH (англ. Secure Shell). В кластерах, функционирующих под управлением операционной системы Windows, используется графический режим доступа, собственное программное обеспечение и протокол удалённого вызова процедур RPC.

В высокопроизводительных вычислительных системах доступ в интерактивном режиме чаще предоставляется только к внешнему узлу. Выполнение пользовательского программного обеспечения на вычислительных узлах осуществляется в пакетном режиме.

4.2 Терминальный доступ к распределённой вычислительной системе Jet

Для доступа к РВС Jet необходимо подключиться к ресурсу `ssh://jet.cpct.sibsutis.ru:22`[8]. В результате подключения будет открыт сеанс интерактивного взаимодействия пользователя с внешним узлом (англ. frontend).

На первом этапе подключения по протоколу SSH пользователю необходимо пройти аутентификацию, т.е. указать имя пользователя и ввести пароль. Если удаленный сервер проверит совпадение предоставленных имени пользователя и его пароля, с тем, что хранится на сервере, то будет подтверждена подлинность предоставленных данных и пользователю будет предоставлен доступ к системе (см. рисунок 3).

```
[user@host ~]$ ssh user@jet.cpct.sibsutis.ru
user@jet.cpct.sibsutis.ru's password:
Last login: Wed Aug 17 13:14:36 2016 from 1.2.3.4
[user@jet ~]$
```

Рисунок 3 – Пример начала SSH-сессии при доступе к ресурсам РВС Jet

[8] Такая запись ресурса означает, что взаимодействие осуществляется по протоколу SSH с узлом cpct.sibsutis.ru и используется сетевой порт с номером 22.

В представленных примерах показан терминальный доступ с использованием стандартной для операционных систем GNU/Linux консольной реализации клиента SSH. Существуют также и иные реализации терминального программного обеспечения, например, утилита PuTTY, активно используемая пользователями операционной системы Microsoft Windows. Для мобильных устройств, функционирующих под управлением Android, широкое распространение получил клиент JuiceSSH от компании Google.

Протокол SSH допускает для аутентификации пользователей использовать распределённую технологию ассиметричной цифровой подписи открытым и закрытым ключами. Для этого пользователь должен сгенерировать пару ключей и поместить на удалённый сервер, к которому он планирует подключаться, открытый ключ в файл `~/.ssh/authorized_keys`[9]. В процессе аутентификации сервер проверит подлинность цифровой подписи клиента и подтвердит его аутентификацию. Пример конфигурирования автоматической авторизации представлен на рисунке 5. Для того, чтобы в процессе аутентификации не запрашивались пароль доступа к закрытому ключу в процессе его записи необходимо использовать пустое значение для парольной фразы. После копирования файлов проверяется работоспособность автоматической аутентификации пользователей.

```
[user@host ~]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/sergey/.ssh/id_rsa.pub.
The key fingerprint is:
18:df:46:20:0d:ee:f9:b6:34:d5:34:98:b2:e9:35:f5 user@host.local
The key's randomart image is:
+--[ RSA 2048]----+
|      oo.        |
|     . ... o     |
|      o . + +    |
|     . = * + o   |
|      + S * . E  |
|       o + .     |
|        *        |
|       o o       |
|        .        |
+-----------------+
[user@host ~]$ scp .ssh/id_rsa.pub user@jet.cpct.sibsutis.ru:
user@jet.cpct.sibsutis.ru's password:
id_rsa.pub                                    100%  404     0.4KB/s   00:00
[user@host ~]$ ssh user@jet.cpct.sibsutis.ru  "cat ./id_rsa.pub >> \
./.ssh/authorized_keys"
user@jet.cpct.sibsutis.ru's password:
[user@host ~]$ ssh user@jet.cpct.sibsutis.ru
Last login: Wed Aug 17 13:55:14 2016 from 1.2.3.4
[user@jet ~]$
```

Рисунок 5 – Пример конфигурирования автоматической аутентификации пользователей

Трансляция информации (передача файлов) в систему или обратно осуществляется с использованием протокола SCP (см. рисунок 6).

[9] Следует отметить, что к файлу обязательно должен быть ограничен доступ (режим 600).

```
[user@host ~]$ scp file.txt user@jet.cpct.sibsutis.ru:
user@jet.cpct.sibsutis.ru's password:
file.txt                                      100%    0     0.0KB/s   00:00
[user@host ~]
```

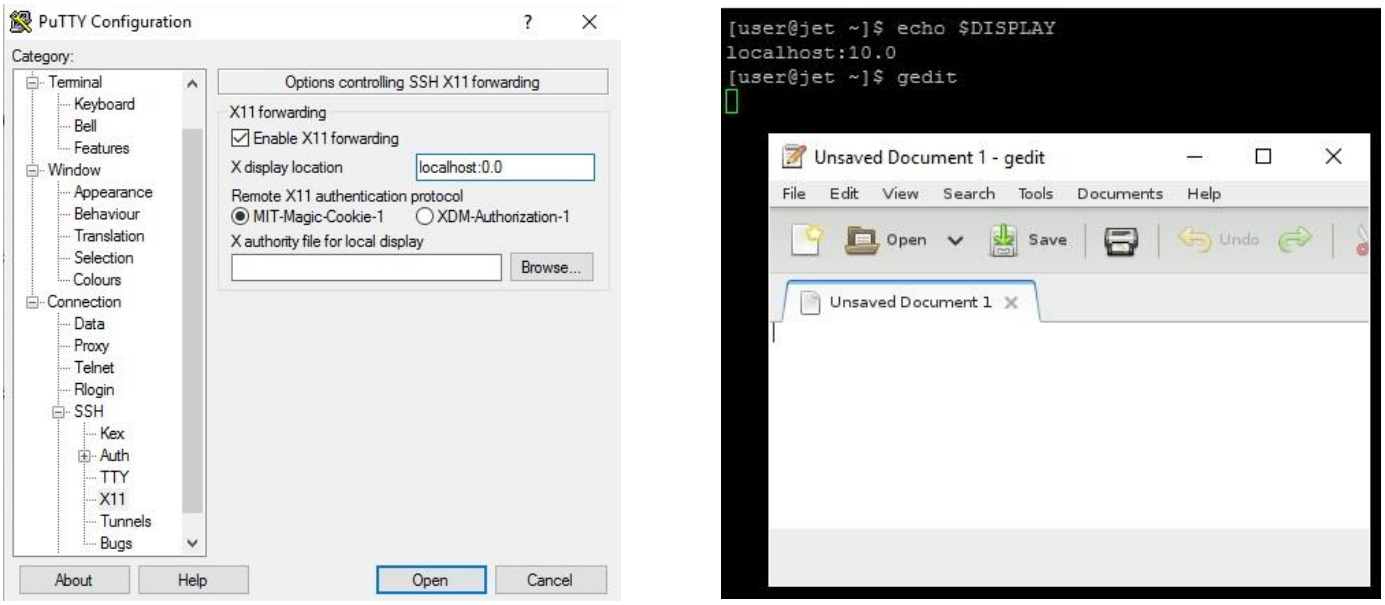
Рисунок 6 – Пример копирования файла в РВС Jet

4.3 Подготовка пользовательского программного обеспечения и исходных данных

Получив доступ к внешнему узлу РВС Jet, пользователь может подготовить необходимое ему программное обеспечение и исходные данные. Принципы работы в терминальном режиме такие же, как при работе с обычной командной оболочкой (командной строкой). В РВС Jet по умолчанию используется командная оболочка Bash.

Пользователям РВС Jet доступен набор средств разработки программного обеспечения: компиляторы (gcc, g++, icc, gfortran и т.п.), отладчики (gdb), средства автоматизации сборки (make, autotools) и т.п.

Используя технологию X11-forwarding возможно применять утилиты, имеющие графический интерфейс. Для этого на компьютере клиента должно быть запущено программное обеспечение XServer и при подключении должен быть включен режим X11-forwarding. Для клиента ssh это делается с использованием опции `-X`, в клиенте PuTTY установкой галочки в меню SSH/X11 (см. пример на рисунке 7).



а) б)

Рисунок 7 – Использование графического терминального режима, клиент SSH – PuTTY, графический сервер Xming (а – настройка X11 forwarding, б – пример запуска графического приложения gedit)

4.4 Управление задачами распределённой вычислительной системы Jet

Для управления ресурсами РВС используется система очередей OpenPBS/Torque и планировщик PBS_shed. Узнать о имеющихся вычислительных узлах и их состоянии можно с использованием команды pbsnodes (см. рисунок 8). В информации, выводимой этой команды, можно узнать: имеются ли свободные вычислительные ядра (параметр state), состояние узла (включен или выключен), список задач, для решения которых назначены ресурсы этого вычислительного узла (jobs) и т.п.

```
[user@jet ~]$ pbsnodes
cn1
     state = free
     power_state = Running
```

```
      np = 8
      ntype = cluster
      jobs = 0/3309.jet.cluster.local,1-5/3310.jet.cluster.local
      status = rectime=1471424096,macaddr=…
      mom_service_port = 15002
      mom_manager_port = 15003
…
[user@jet ~]$
```

Рисунок 8 – Пример использовании команды pbsnodes (часть вывода результатов пропущена)

Для того, чтобы использовать ресурсы РВС пользователь должен описать задачу, т.е. определить какие ресурсы ему необходимы и задать перечень действий, который должна выполнить система, чтобы запустить пользовательское программное обеспечение на выполнение на выделенных ресурсах. При этом, указанная последовательность начинает выполняться на одном из выделенных вычислительных узлов (как правило, это узел с наименьшим логическим номером в системе из множества узлов, выделенных для решения задачи).

Описание задачи оформляется в виде специального файла, написанного с использованием нотации пакетных файлов командной оболочки Bash (см. рисунок 9). В первых строках этого файла в виде комментариев, начинающихся с префикса `#PBS` описываются параметры задачи:

- `N` – наименование задачи, которое будет использовано для отображения её состояния в очереди задач;

- `l` – список ресурсов, которые необходимы для решения задачи. Ресурсы указываются в следующем виде: `nodes=<количество вычислительных узлов>:ppn=<количество вычислительных ядер>`. Подробнее об описании ресурсов можно прочитать в техническом описании `qsub`.

- `j` – объединение стандартных потоков вывода (запись `oe` означает, что поток вывода ошибок объединяется со стандартным потоком вывода и они перенаправляются в один файл).

После описания ресурсов записываются необходимые команды оболочки Bash, которые необходимо выполнить для запуска пользовательского программного обеспечения.

```
#PBS -N Config
#PBS -l nodes=1:ppn=2
#PBS -j oe
cd $PBS_O_WORKDIR
env
cat $PBS_NODEFILE
sleep 20
```

Рисунок 9 – Пример описания задачи для распределённой вычислительной системы

Для того, чтобы сообщить пользовательскому приложению о фактически выделенных ресурсах распределённой вычислительной системы планировщик OpenPBS/Torque специальным образом формирует пользовательское окружение, помещая в него ряд переменных и создает служебные файлы (см. таблицу 1).

Таблица 1 – Переменные среды окружения

Имя переменной	Назначение переменной
`$PBS_O_WORKDIR`	Путь к рабочему каталогу, в котором должно выполняться пользовательское программное обеспечение
`$PBS_JOBID`	Полный идентификатор задачи

$PBS_NUM_NODES	Количество вычислительных узлов, запрошенных для решения задачи
$PBS_JOBNAME	Имя задачи, которую пользователь указал в описании (параметр -N)
$PBS_NODEFILE	Пусть к файлу, в котором содержится описание вычислительных ресурсов, выделенных для решения задачи. В файле в каждой строке указано сетевое имя узла, выделенного для запуска программного обеспечения.

Чтобы поставить задачу в очередь используется команда `qsub` (см. рисунок 10) Все параметры, которые указываются в файле с описанием задачи возможно указать прямо в командной строке в виде опций. В этом случае в файле с описанием задачи останется только перечень выполняемых действий.

```
[user@jet ~]$ qsub myjob.job
3371.jet.cluster.local
[user@jet ~]$ qstat
Job ID               Name          User        Time Use S Queue
-------------------- ------------- ---------- -------- - -----
3365.jet              Config        user             0 R debug
3366.jet              Config        user             0 R debug
3367.jet              Config        user             0 R debug
3368.jet              Config        user             0 R debug
3369.jet              Config        user             0 R debug
3370.jet              Config        user             0 R debug
3371.jet              Config        user             0 R debug
[user@jet ~]$
```

Рисунок 10 – Пример работы с очередью задач распределённой вычислительной системы

Вывести состояние очереди(ей) задач можно с использованием команды `qstat`. При этом для каждой задачи выводится следующая информация: идентификатор, имя задачи, пользователь, время нахождения в очереди, состояние, имя очереди (в которой находится задача).

Удалить задачу из очереди возможно с помощью команды `qdel`. Изменить параметры задачи позволяет утилита `qalter`.

Система управления ресурсами OpenPBS/Torque позволяет запускать задачи, требующие интерактивного взаимодействия с пользователем. В этом случае из описания задачи считываются только параметры. После того, как ресурсы будут выделены на одной из вычислительных машин системы будет запущена командная оболочка и её ввод/вывод будет перенаправлен через `qsub` пользователю. Дальше пользователь может взаимодействовать с удалённой командной оболочной обычным образом. Чтобы запустить задачу в интерактивном режиме необходимо указать опцию `-I`, в командной строке запуска `qsub`.

5. Проектирование распределённого программного обеспечения

Распределённым программным обеспечением (далее распределённое ПО) называют такое программное обеспечение, которое состоит из нескольких функционально независимых частей, настроенных так, чтобы решать общую единую задачу. Взаимодействие частей распределённого ПО осуществляется посредствам трансляции информации посредством современных компьютерных технологий и/или глобальных компьютерных сетей.

Проектирование распределённого ПО, по сути, ничем не отличается от проектирования любого другого программного обеспечения, за одним исключением. На этапе проектирования архитектуры распределённого ПО необходимо выделить независимые функциональные блоки так, чтобы взаимодействие между ними было минимальным.

На этапе детального проектирования необходимо строго определить интерфейс взаимодействия независимых модулей, определить протокол взаимодействия и форматы передачи данных.

Задание на лабораторную работу

Базовые задания.

1. Подключитесь к ресурсу `ssh://jet.cpct.sibsutis.ru:22`. После первого подключения измените пароль для своей учетной записи.
2. Подготовьте программное обеспечение, реализующее алгоритм умножения двух прямоугольных матриц целых чисел. Размеры матриц задаются параметрами командной строки. Исходные матрицы генерируются псевдослучайным образом (стандартный генератор). Исходные матрицы и результат их перемножения выводятся в стандартный поток вывода. Язык программирования и средства разработки, с использованием которых будет реализовано программное обеспечение, выбираются из числа доступных в распределённой вычислительной системе Jet.
3. Опишите задачу, требующую для своего решения один узел и одно ядро на нем. При решении задачи должно запускаться программное обеспечение, реализованное в п. 2. Размеры перемножаемых матриц задаются в виде трех чисел: M (количество строк в 1 таблице), N (количество столбцов в 1 таблице и строк во 2 таблице), P (количество столбцов 2 таблицы), где M – линейный номер задач в её полном идентификаторе, N – удвоенное значение M, P – длина строки, содержащей сетевое имя узла, на котором выполняется задача.
4. Поставьте задачу по умножению матриц в очередь. Посмотрите состояние очереди. Дождитесь завершения решения задачи и убедитесь в правильности выполнения программного обеспечения.

Основные задания.

5. Сконфигурируйте учетную запись в системе Jet так, чтобы при подключении по SSH использовалась пара открытый/закрытый ключ. Убедитесь, что теперь при подключении к системе, аутентификация происходит автоматически (без запроса паролей).
6. Используя технологию X11 forwarding запустите графический редактор gedit и создайте в домашнем каталоге на основном узле системы Jet текстовый файл, в котором разместите информацию о себе: номер группы, учетная запись, ФИО.
7. Создайте описание задачи, требующей для своего решения два вычислительных узла и три вычислительных ядра на каждом. Запустите задачу в интерактивном режиме. После получения доступа к командной строке выведите на экран содержимое всех переменных среды окружения, начинающихся с префикса PBS_. Выведите на экран содержимое файла, в котором указаны имена узлов, выделенных для решения задачи. Используя команду `mount` определите каким образом сконфигурирована файловая система на вычислительном узле.

Задания повышенной сложности.

8. Разработайте распределённое программное обеспечение реализующее следующий функционал:
   - Модуль вывода информации о состоянии вычислительных узлов системы Jet. Запуск модуля возможен только на внешнем узле. Состоянием узлов считаем значения полей: state, power_state и jobs.

- Модуль отображения состояния распределённой вычислительной системы. Использует информацию, получаемую от первого модуля или загружаемую из локального файла (текущее состояние распределённой системы должно сохранятся в локальный файл).

Интерфейс модуля отображения состояния распределённой системы значения не имеет (может быть текстовый, графический и т.п.)

Контрольные вопросы

1. Что является современным инструментарием высокопроизводительной обработки информации?
2. К какому классу (по Флинну) относится архитектура распределённых вычислительных систем?
3. Что отражает список Top-500?
4. Кем и где была предложена концепция однородных вычислительных систем?
5. На какой модели базируется функционирования распределённых вычислительных систем?
6. Назовите принципы, лежащие в основе модели коллектива вычислителей?
7. Какие принципы положены в основу технической реализации модели коллектива вычислителей?
8. Какими архитектурными принципами обладает распределённые вычислительные системы?
9. Опишите мультикластерную пространственно-распределённую вычислительную систему Центра параллельных вычислительных технологий СибГУТИ.
10. Что такое терминальный доступ? Как выглядит терминал и из каких элементов он состоит? Как терминал взаимодействует с вычислительной системой?
11. Какие протоколы используются для реализации терминального доступа с использованием локальных и/или глобальных сетей?
12. Зачем используется технология ассиметричной цифровой подписи в процессе терминального доступа?
13. Опишите процесс профессиональной эксплуатации распределённых вычислительных систем.
14. Что такое X11-forwarding? Как её использовать?
15. Каким образом можно узнать состояние узлов вычислительной системы?
16. Что такое паспорт задачи и как он устроен?
17. Каким образом поставить задачу в очередь? Изменить её параметры? Удалить задачу из очереди?
18. Какие особенности имеет процесс проектирования распределённого программного обеспечения?