

Изучение особенностей платформы Globus Toolkit 5.0.3

Globus Toolkit представляет собой набор модулей для построения виртуальной организации распределенных вычислений. Каждый модуль определяет интерфейс, используемый высокоуровневыми компонентами, и имеет реализацию для различных сред выполнения. Вкупе они образуют виртуальную машину Globus. Все модули могут быть разбиты на следующие группы (рис. 1):

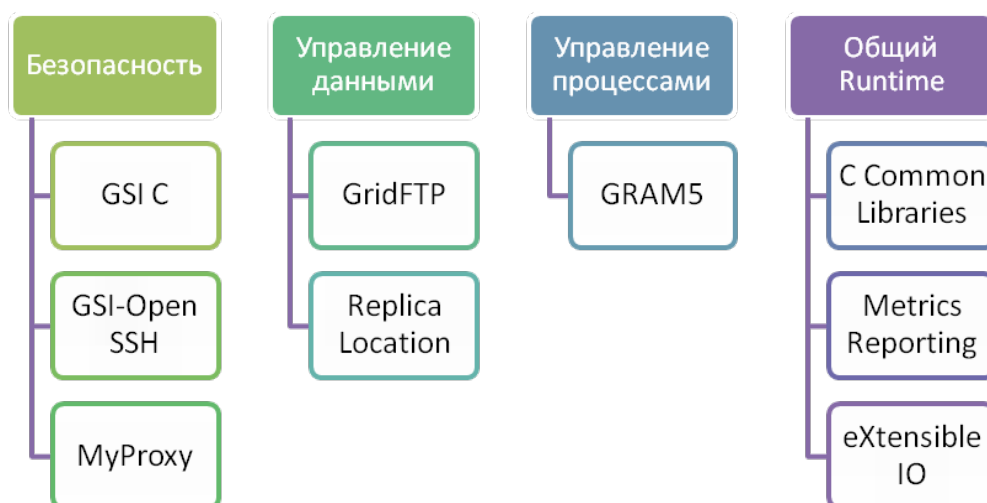


Рис. 1. Архитектура Globus Toolkit 5.0.3

- модули безопасности,
- модули управления данными,
- модули управления процессами,
- общий Runtime.

Управление процессами

Globus Toolkit 5 предоставляет доступ к вычислительным процессам при помощи компонента GRAM5 (Grid Resource Allocation and Management). Благодаря нему можно добавлять, контролировать, отслеживать и

отменять задания в grid'е. Однако, фактически, GRAM5 не является планировщиком заданий – это набор сервисов и клиентов для взаимодействия с целым рядом локальных планировщиков заданий посредством стандартного текстового протокола.

GRAM не является чем-то цельным, он состоит из компонентов, причем как на протокольном, так и на программном уровне реализации. Такой компонентный подход позволяет скрыть все подробности реализации и абстракции для простоты работы.

Сервисная модель включает в себя два демона:

- **globus-gatekeeper** – демон и GSI-библиотека, используемые для защиты передачи данных и для координации сервисов
- **globus-job-manager** – демон, реализующий планировку задач и функционал для передачи данных или файлов

Чтобы на данном вычислительном узле можно было удаленно запускать на исполнение программы, на нем должен выполняться специальный процесс называемый Gatekeeper. Пользовательские приложения формируют запросы к GRAM на специальном языке RSL (Resource Specification Language). Gatekeeper работает в привилегированном режиме и выполняет следующие функции:

- производит взаимную аутентификацию с клиентом;
- анализирует RSL запрос;
- отображает клиентский запрос на учетную запись некоторого локального пользователя;
- запускает от имени локального пользователя специальный процесс, называемый Job Manager, и передает ему список требующихся ресурсов.

После того, как Gatekeeper выполнит свою работу, Job Manager запускает задание (процесс или несколько процессов) и производит его дальнейший мониторинг, сообщая клиенту об ошибках и других событиях. Gatekeeper запускает только один Job Manager для каждого пользователя,

который управляет всеми заданиями данного пользователя. Когда заданий больше не остается, Job Manager завершает работу.

С точки зрения безопасности, GRAM5 является очень защищенным компонентом. Протоколы передачи данных основаны на SSL. После однократной авторизации, каждая копия задания выполняется под локальным POSIX-пользователем. При этом ограничиваются возможность мониторинга и контролирования выполнения задачи для тех, кто авторизован лишь локально. GRAM берет на себя задачу протоколирования всех событий и ведет полную историю действий задания и критических системных операций.

GRAM5 обеспечивает высокопроизводительную передачу файлов между компьютерными ресурсами и внешними накопителями информации, как до так и во время выполнения задания. Доступ к информации можно получить через HTTP, GASS и GridFTP.

Управление данными

Управление данными в Globus Toolkit 5 осуществляется посредством инструментов GridFTP. Протокол GridFTP предоставляет безопасную, устойчивую к ошибкам, быструю и эффективную передачу данных (особенно – в больших объемах).

Globus Toolkit предоставляет:

- серверную реализацию globus-gridftp-server
- клиент - командную строку globus-url-copy
- набор библиотек для разработчиков

Для доступа к данным используется клиент, именуемый globus-url-copy (хотя существуют также и сторонние разработки с интерактивным интерфейсом). Этот клиент позволяет получить доступ к информации через различные протоколы (hhttp, https, ftp, gsiftp, file). Клиент представляет собой простую командную строку, которая может являться элементом других сценариев, например:

```
globus-url-copy      gsiftp://remote.host.edu/path/to/file
file:///path/on/local/host
```

Помимо GridFTP в набор для управления данными вполне входит такой компонент, как RLS (Replica Location Service). Как видно из названия, сервис предназначен для контроля за расположением дубликатов данных, т.е. для помощи в определении физического расположения файла. Это позволяет получать доступ к различным копиям одного и того же файла (например, выбирать файл, расположенный на ближайшем сервере)

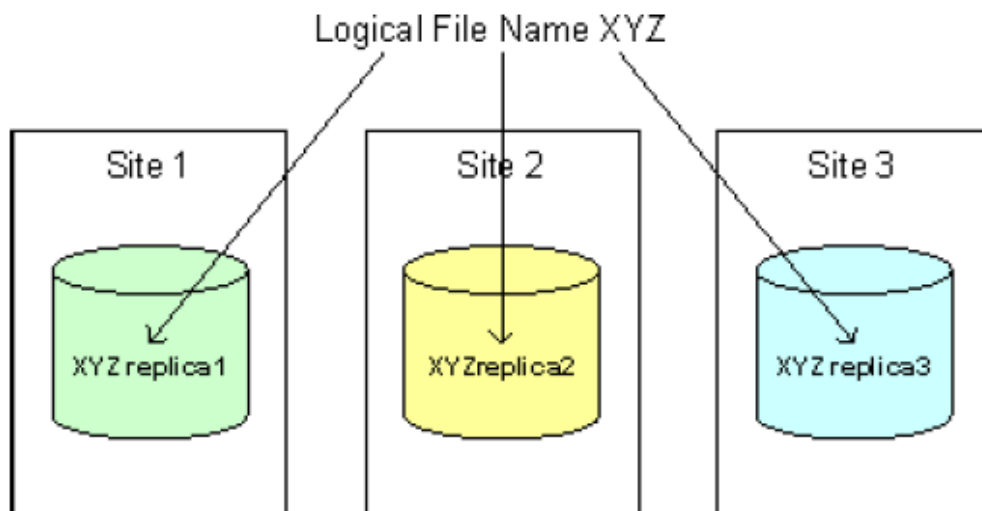


Рис. 2. Принцип работы RLS

Как видно из рисунка 2, файл, имеющий **логическое имя XYZ** на физических носителях будет иметь различные **физические имена**.

Безопасность

В основе безопасности в Globus Toolkit 5 лежат принципы авторизации и сертификации (X.509), реализуемые с помощью компонентов защиты. Они позволяют контролировать действия всех пользователей в grid'е, защищая передаваемые данные. Также эти компоненты поддерживают ре-

дактирование полномочий пользователей, сохранение информации о членстве в группах доступа.

В состав инструментов безопасности входят следующие компоненты:

Grid Security Infrastructure (GSI) – компонент, предоставляющий API для аутентификации, авторизации и сертификации. GSI основана на надежной и широко используемой инфраструктуре криптографии с открытым ключом (Public Key Infrastructure – PKI). В качестве идентификаторов пользователей и ресурсов в GSI используются цифровые сертификаты X.509. В работе с сертификатами X.509 и в процедуре выдачи/получения сертификатов задействованы три стороны:

1. Центр Сертификации (Certificate Authority – CA) – специальная организация, обладающая полномочиями выдавать (подписывать) цифровые сертификаты. В GT5 эта задача выполняется компонентом SimpleCA.
2. Подписчик – это человек или ресурс, который пользуется сертификационными услугами CA. SimpleCA включает в сертификат данные, предоставляемые подписчиком (имя, организация и пр.) и ставит на нем свою цифровую подпись.
3. Пользователь – это человек или ресурс, полагающийся на информацию из сертификата при получении его от подписчика. Пользователи могут принимать или отвергать сертификаты, подписанные CA.

В Globus Toolkit используются два типа сертификатов X.509:

1. Сертификат пользователя (User Certificate) – этот сертификат должен иметь каждый пользователь, работающий с grid-системой. Сертификат пользователя содержит информацию об имени пользователя, организации, к которой он принадлежит, и центре сертификации, выдавшем данный сертификат.
2. Сертификат узла (Host Certificate) – это сертификат должен иметь каждый узел (ресурс) grid-системы. Сертификат узла аналогичен сер-

тификату пользователя, но в нем вместо имени пользователя указывается доменное имя конкретного вычислительного узла.

Для управления сертификатами X.509 и полномочиями в GT5 используется компонент **MyProxy**. MyProxy сочетает в себе онлайн репозиторий учетных записей и онлайн Центр Сертификации, позволяющие пользователям безопасно получить доступ к своим учетным записям где и когда потребуется. Пользователь с помощью команды `mypoxy-login` аутентифицируется и получает права доступа, включающие в себя CA-сертификат и Списки Аннулированных Сертификатов (CRLs).

Последним компонентом безопасности является **GSI-OpenSSH** – модифицированный OpenSSH, в котором добавлена работа с прокси-сертификатами X.509 для обеспечения доступа к удаленным системам и передачи данных без ввода паролей, полагаясь на достоверный проху credential для аутентификации (это комбинация прокси-сертификата и отвечающего ему ключа).

Структура grid'а с точки зрения запущенных сервисов

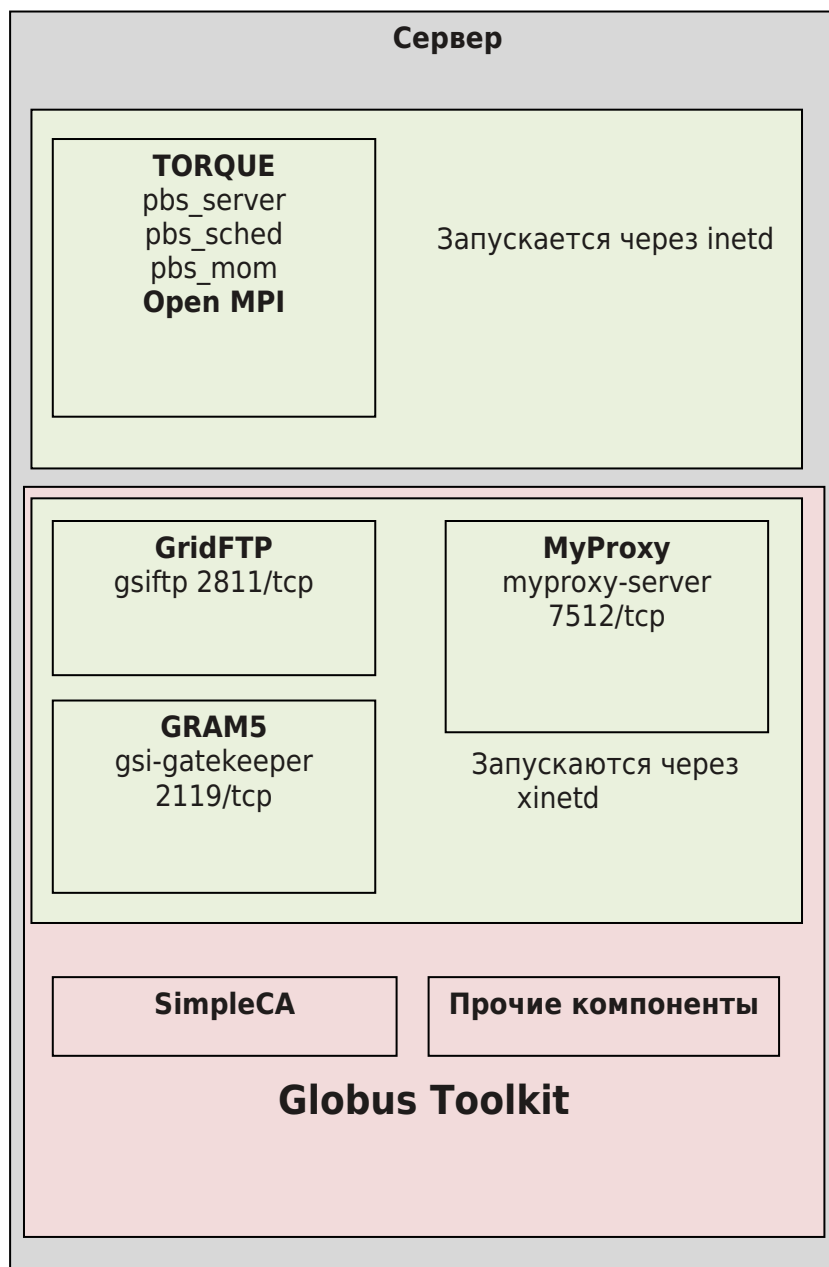


Рис.3 Структура серверной машины

На серверной машине с помощью службы **xinetd** будут запущены демоны `gsiftp`, `gsigatekeeper` и `myproxy-server`, которые будут обеспечивать сетевое взаимодействие. С помощью службы `inetd` будут запускаться все сервисы кластерного ПО, настроенные на серверной машине: `pbs_server`, `pbs_mom` и `pbs_sched`. Также на серверной машине будут установлены

SimpleCA, Open MPI, а также необходимые для работы платформы компоненты Globus Toolkit.



Рис.4 Структура клиентской машины

На клиентской машине с помощью службы **xinetd** будут запущены только демон `gsiftp`, которые необходим для обеспечения доступа к данным. С помощью службы `inetd` будет запускаться сервис `pbs_mom`, позволяющий использовать машину в качестве вычислительного узла. Также, как и на серверной машине, будут установлены SimpleCA, Open MPI, GRAM5, MyProxy и другие компоненты Globus Toolkit.

Установка Globus Toolkit 5.0.3

Установка на первую машину

Globus Toolkit является программным обеспечением с открытым исходным кодом, написанным на С. На официальном сайте The Globus Alliance (<http://www.globus.org>) доступны все выпущенные версии Toolkit'а, в том числе и последние сборки. Globus Toolkit 5.0.3 может быть установлен на Apple OS X, RedHat, Fedora Core, Debian, SuSE, FreeBSD и Solaris (или бинарно совместимые с ними системы). Для развертывания grid будут использоваться машины с операционной системой Debian 6 Squeeze.

Прежде чем приступить к установке и сборке самого Globus Toolkit 5.0.3, необходимо убедиться, что на машине установлены необходимые программы и библиотеки, а именно:

Openssl 0.9.7+

Libssl (development + shared libraries)

Zlib (development + shared libraries)

gcc

g++

tar

sed

make

vim (можно заменить любым другим текстовым редактором)

xinetd (необходима для конфигурирования демонов)

ssh

После того, как система будет готова к установке, необходимо скачать и распаковать инсталлятор. Затем, сконфигурировать его и запустить сборку.

```

globus@GlobusNode:~$ tar xjf gt5.0.3-all-source-installer.-
tar.gz
globus@GlobusNode:~$ cd gt5.0.3-all-source-installer
globus@GlobusNode:/gt5.0.3-all-source-installer$ ./configure
--prefix=/sandbox/globus/
checking build system type... i686-pc-linux-gnu
configure: creating ./config.status
config.status: creating Makefile
root@GlobusNode:~/gt5.0.3-all-source-installer$ make |
tee installer.log
cd gpt-3.2autotools2004 && OBJECT_MODE=32 ./build_gpt
build_gpt ==> installing GPT into /sandbox/globus-5/
Your build completed successfully. Please run make install.
root@GlobusNode:~/gt5.0.3-all-source-installer$ make install
/sandbox/globus-5//sbin/gpt-postinstall
...
..Done

```

Теперь для удобства дальнейшей работы добавим подключение команд Globus Toolkit в файл /etc/bash.bashrc. Таким образом мы избавим себя от обязанности подключать команды при открытии нового терминала:

```

root@GlobusNode:~#vim /etc/bash.bashrc

```

Добавим в начало файла следующие две строки:

```

export GLOBUS_LOCATION=/sandbox/globus-5
source $GLOBUS_LOCATION/etc/globus-user-env.sh

```

Сохраняем, подключаем заново файл /etc/bash.bashrc и перезапускаем терминал.

```

root@GlobusNode:~#source /etc/bash.bashrc

```

После сборки Globus Toolkit'a машина готова к конфигурации. В этот момент решается, будет ли данная машина серверным узлом или клиентом. В зависимости от этого идет дальнейшая конфигурация. Сделаем копию виртуальной машины, и из этой копии уже потом будем настраивать вторую машину.

Настройка первой машины может быть разбита на следующие этапы:

1. Настройка Simple-CA
2. Настройка MyProxy сервера
3. Настройка GridFTP

4. Настройка GRAM5

Этап 1. Настройка Simple-CA

На данном этапе будет настроена аутентификация и авторизация пользователей с помощью сертификатов. Потребуется сертификаты как для пользователей, так и для сервисов. Для сервисов они будут совпадать с сертификатами хоста. Для создания сертификатов будет использоваться **SimpleCA**, которая поставляется вместе с Globus Toolkit.

Прежде чем приступить к настройке сертификатов необходимо проверить файл `/etc/hosts`.

```
root@GlobusNode:/# vim /etc/hosts
127.0.0.1      GlobusNode    localhost
127.0.1.1      GlobusNode.WORKGROUP    GlobusNode.localdomain
.....
```

Главное – чтобы после 127.0.0.1 стояло GlobusNode без указаний на рабочую группу, иначе в дальнейшем может возникнуть конфликт имен. Можно приступать к установке и настройке **SimpleCA**. Запускаем скрипт автоматической конфигурации SimpleCA:

```
root@GlobusNode:~# cd /gt5.0.3-all-source-installer/quickstart
root@GlobusNode:gt5.0.3-all-source-installer/quickstart# perl
gt-server-ca.pl -y
Setting up /sandbox/globus-5/
Please enter a password of at least four characters for the
CA: <CApassword>
Confirm password: <CApassword>
Creating a new simpleCA, logging to gt-server-ca.log...
Running setup-gsi...
Your CA hash is: d58f5223
It is located at /sandbox/globus-
5//share/certificates/d58f5223.0
y
Your host DN is /O=Grid/OU=GlobusTest/OU=simpleCA-
globusnode/CN=host/globusnode
The hostcert is located at /sandbox/globus-5/etc/hostcert.pem
```

· В этот момент выполнение кода приостановится. Нужно будет ввести 'y' и нажать Enter для продолжения. В этот момент должен возникнуть запрос на подтверждение создания SimpleCA, но из-за ошибки в скрипте вопрос не выводится.

SimpleCA установилась и был выписан сертификат для нашей машины. Проверяем, создались ли необходимые файлы:

```
root@GlobusNode:~# ls /root/.globus/
simpleCA
root@GlobusNode:~# ls /root/.globus/simpleCA/
cacert.pem      certs      crl
globus_simple_ca_d58f5223_setup-0.20.tar.gz
grid-ca-ssl.conf  index.txt  index.txt.attr
index.txt.old  newcerts  private    serial
serial.old
```

Все необходимые файлы успешно созданы. Последним шагом на данном этапе будет копирование подписанных и доверенных сертификатов в /etc/grid-security. Это рабочая директория, в которой будут храниться все необходимые для аутентификации сертификаты:

```
root@GlobusNode:~# mkdir /etc/grid-security
root@GlobusNode:~# mv $GLOBUS_LOCATION/etc/host*.pem
/etc/grid-security/
root@GlobusNode:~# cp -r $GLOBUS_LOCATION/share/certificates/
/etc/grid-security/
```

Этап 2. Настройка MyProxy сервера

На данном этапе будет создан и сконфигурирован MyProxy сервер на GlobusNode. Он используется для хранения и выдачи сертификатов пользователей и сервисов. Копируем из директории Globus'а конфигурационный файл по умолчанию и редактируем его:

```
root@GlobusNode:~# cp $GLOBUS_LOCATION/share/myproxy/myproxy-
server.config /etc
root@GlobusNode:~# vim /etc/myproxy-server.config
```

Необходимо раскомментировать в файле /etc/myproxy-server.config строки 18-24 (т.е. убрать символ решетки в начале следующих строк):

```
accepted_credentials  ""
authorized_retrievers  ""
```

```

default_retrievers      "*"
authorized_renewers     "*"
default_renewers        "none"
authorized_key_retrievers "*"
default_key_retrievers  "none"

```

Теперь необходимо добавить в файл `/etc/services` информацию о новом сервисе – `myproxy-server`. Воспользуемся существующим файлом модификации и допишем его в конец файла `/etc/services`.

```

root@GlobusNode:~# cat
$GLOBUS_LOCATION/share/myproxy/etc.services.modifications >>
/etc/services

```

Проверяем результат выполненной команды:

```

root@GlobusNode:~# tail /etc/services
fido          60179/tcp    # fidonet EMSI over TCP
# Local services
myproxy-server 7512/tcp    # Myproxy server

```

Сервис успешно был добавлен в список. Теперь необходимо создать конфигурационный файл службы `myproxy-server`. Также воспользуемся заготовкой файла, поставляемой вместе с Globus Toolkit и скопируем ее в директорию `/etc/xinetd.d/`

```

root@GlobusNode:~# cp
$GLOBUS_LOCATION/share/myproxy/etc.xinetd.myproxy /etc/xinet-
d.d/myproxy

```

Исправляем файл с настройками сервиса. Комментарий

```

root@GlobusNode:~# vim /etc/xinetd.d/myproxy
service myproxy-server
{
    socket_type    = stream #Тип сокета - поток
    protocol      = tcp    #Протокол - TCP
    wait          = no     #Сервис работает в многопоточном режиме
    user          = root    #Сервис запускается от пользователя root
    server        = /sandbox/globus-5/sbin/myproxy-server #Путь к
исполняемому файлу
    env           = GLOBUS_LOCATION=/sandbox/globus-5
LD_LIBRARY_PATH=/sandbox/globus-5/lib #Переменные среды
    disable       = no     #Сервис активен
}

```

Перезапускаем службу xinetd и проверяем работоспособность сервиса myproxy-server.

```
root@GlobusNode:~# /etc/init.d/xinetd reload
Reloading internet superserver configuration: xinetd.
root@GlobusNode:~# netstat -an | grep 7512
tcp    0      0 0.0.0.0:7512        0.0.0.0:*          LISTEN
```

MyProxy сервер запущен и успешно прослушивает указанный в /etc/services порт - 7512. По этому порту к нему будут выполняться подключения. Теперь необходимо создать сертификат для тестового пользователя. Создаем в системе реального пользователя с полным именем *QuickStart User* и логином *quser*. При создании сертификата будет необходимо указать два пароля. Первый - это пароль, который будет в дальнейшем использоваться для авторизации пользователя *quser* (минимум 6 символов), а второй – пароль, который был указан при создании **SimpleCA** для доступа к ней.

```
root@GlobusNode:~ # myproxy-admin-adduser -c "QuickStart User"
-l quser
Enter PEM pass phrase: <например, qpassword>
Verifying - Enter PEM pass phrase: <qpassword>
```

```
To sign the request
please enter the password for the CA key: <пароль от SimpleCA>
```

```
The new signed certificate is at:
/root/.globus/simpleCA/newcerts/02.pem
```

```
using storage directory /var/myproxy
Credential stored successfully
Certificate subject is:
/O=Grid/OU=GlobusTest/OU=simpleCA-globusnode/CN=QuickStart
User
```

SimpleCA подписала для пользователя *quser* сертификат и поместила его в свою директорию. Последний шаг – создаем файл *grid-mapfile* под *root*'ом, который будет использоваться для авторизации. Копируем целиком строку */O=Grid/OU=...* из предыдущего вывода:

```
root@GlobusNode:/etc/grid-security# vim /etc/grid-
security/grid-mapfile
"/O=Grid/OU=GlobusTest/OU=simpleCA-globusnode/CN=QuickStart
User" quser
```

Этап 3. Настройка GridFTP

Теперь, когда уже настроена аутентификация, можно приступать к сервисам. Сначала установим GridFTP – сервис для передачи данных по протоколу gsiftp

Создаем новый файл **/etc/xinetd.d/gridftp** и редактируем его, подобно файлу myproxy-server.

```
root@GlobusNode:/etc/grid-security# vim /etc/xinetd.d/gridftp
service gsiftp
{
instances          = 100 #Максимум соединений - 100
socket_type         = stream #Тип сокета - поток
wait                = no #Сервис работает в многопоточном режиме
user                = root #Сервис запускается от пользователя root
env += GLOBUS_LOCATION=/sandbox/globus-5
env += LD_LIBRARY_PATH=/sandbox/globus-5/lib #Переменные среды
server              = /sandbox/globus-5/sbin/globus-gridftp-server #Путь
к исполняемому файлу
server_args          = -i #Запускается как inetd сервис
log_on_success      += DURATION #Параметр ведения журнала
disable             = no #Сервис активен
}
```

Добавляем в конец файла **/etc/services** информацию о новом сервисе - gsiftp.

```
root@GlobusNode:/etc/grid-security# vim /etc/services
# Local services
myproxy-server      7512/tcp  # Myproxy server
gsiftp              2811/tcp
```

Перезапускаем службу xinetd и проверяем работоспособность сервиса gsiftp.

```
root@GlobusNode:/etc/grid-security# /etc/init.d/xinetd reload
Reloading internet superserver configuration: xinetd.
root@GlobusNode:/etc/grid-security# netstat -an | grep 2811
tcp        0      0 0.0.0.0:2811          0.0.0.0:*            LISTEN
```


Сервис работает и успешно слушает порт 2811.

Этап 4. Настройка GRAM5

Приступим к установке GRAM5 – менеджера вычислительных ресурсов. Можно настроить несколько различных Локальных Менеджеров Ресурсов (LRM) с которыми будет работать GRAM, но в этом руководстве будет рассмотрен лишь простой "fork" jobmanager, отвечающий за запуск задачи на локальной машине.

Создаем файл **/etc/xinetd.d/gsigatekeeper** и настраиваем новую службу.

```
root@GlobusNode:/etc/grid-security# vim /etc/xinetd.d/gsigate-
keeper
service gsigatekeeper
{
    socket_type = stream #Тип сокета - поток
    protocol = tcp #Протокол - TCP
    wait = no #Сервис работает в многопоточном режиме
    user = root #Сервис запускается от пользователя root
    env = LD_LIBRARY_PATH=/sandbox/globus-5/lib #Переменная
среды
    server = /sandbox/globus-5/sbin/globus-gatekeeper #Путь к
исполняемому файлу
    server_args = -conf /sandbox/globus-5/etc/globus-gatekeep-
er.conf #Путь к конфигурационному файлу для службы
    disable = no #Сервис активен
}
```

Добавляем в конец файла **/etc/services** информацию о новом сервисе – gsigatekeeper.

```
root@GlobusNode:/etc/grid-security# vim /etc/services
# Local services
myproxy-server 7512/tcp # Myproxy server
gsiftp         2811/tcp
gsigatekeeper  2119/tcp
```

Перезапускаем xinetd и проверяем работоспособность сервиса.

```
root@GlobusNode:/etc/grid-security# /etc/init.d/xinetd reload
```

```
Reloading internet superserver configuration: xinetd.  
root@GlobusNode:/etc/grid-security# netstat -an | grep 2119  
tcp        0      0 0.0.0.0:2119      0.0.0.0:*        LISTEN
```

Gatekeeper теперь автоматически запускается с xinetd, слушает порт 2119, и сконфигурирован, чтобы принимать и перенаправлять задачи fork jobmanager'у.

Установка на вторую машину

Сделаем копию ранее скопированной виртуальной машины с уже установленным Globus Toolkit'ом, и настроим эту машину как клиента, с которого будем передавать задачи на сервер. На этой машине также необходимо создать пользователя *QuickStart User* с логином *quser*. Затем, в настройках сети необходимо поменять сетевое имя машины на GlobusNode2. Соответственно должен быть переписан файл **/etc/hosts**.

Помимо этих изменений на обеих машинах следует внести в файл **/etc/hosts** IP-адреса другой машины, чтобы они могли найти друг друга.

```
root@GlobusNode2:/# vim /etc/hosts  
127.0.0.1      GlobusNode2  localhost  
127.0.1.1      GlobusNode2.WORKGROUP  GlobusNode2.localdo-  
main  
.....  
192.168.231.147 Globusnode
```

Проверим, что машины видят друг друга:

```
root@GlobusNode:/# ping globusnode2  
root@GlobusNode2:/# ping globusnode
```

Этап 1. Настройка безопасности на второй машине

Настройка безопасности на второй машине сводится к копированию созданных SimpleCA файлов. Необходимо скопировать папку \$GLOBUS_LOCATION/share/certificates с первой машины на вторую:

```
globus@GlobusNode2:~$ scp -r globusnode:/sandbox/globus-  
5/share/certificates /sandbox/globus-5/share  
d58f5223.signing_policy      100% 1337 1.3KB/s 00:00  
globus-host-ssl.conf.d58f5223 100% 2709 2.7KB/s 00:00  
d58f5223.0                  100% 924 0.9KB/s 00:00  
grid-security.conf.d58f5223  100% 1324 1.3KB/s 00:00  
globus-user-ssl.conf.d58f5223 100% 2809 2.7KB/s 00:00
```

Теперь создадим на первой машине (GlobusNode) сертификат для второй машины (globusnode2), но для пользователя *quser*:

```
root@GlobusNode:/home/globus# myproxy-admin-addservice
-c "globusnode2" -l quser
Enter PEM pass phrase: <qpassword>
Verifying - Enter PEM pass phrase: <qpassword>
To sign the request
please enter the password for the CA key: <CApassword>
The new signed certificate is at:
/root/.globus/simpleCA/newcerts/03.pem

using storage directory /var/myproxy
Credential stored successfully
Certificate subject is:
/O=Grid/OU=GlobusTest/OU=simpleCA-globusnode/CN=globusnode2
```

Данный сертификат был добавлен в папку /root/.globus/simpleCA/ и отвечает за возможность машины с именем globusnode2 выполнить вход под пользователем *quser*. Теперь получим этот сертификат на второй машине с помощью команды *myproxy-retrieve*, указав в качестве параметров имя сервера, имя нашей машины и имя пользователя, для которого выпущен сертификат.

```
root@GlobusNode2:/home/globus# myproxy-retrieve -s globusnode
-k "globusnode2" -l quser
Enter MyProxy pass phrase: <qpassword>
Credentials for quser have been stored in
/etc/grid-security/hostcert.pem and
/etc/grid-security/hostkey.pem.
```

Разрываем соединение с MyProху, так как оно больше не нужно.

```
root@GlobusNode2:/home/globus# myproxy-destroy -s globusnode
-k "globusnode2" -l quser
MyProxy credential 'globusnode2' for user quser was successfully removed.
```

Теперь у второй машины есть все необходимые сертификаты. Последний шаг – создать **grid-mapfile** и вписать в него то же, что и на первой машине. Это необходимо для корректной авторизации машин:

```
root@GlobusNode2:~# vim /etc/grid-security/grid-mapfile
"/O=Grid/OU=GlobusTest/OU=simpleCA-globusnode/CN=QuickStart
User" quser
```

Этап 2. Настройка второй машины: GridFTP

Установка GridFTP идентична установке на первой машине, описанной в этапе 3. Следующие команды необходимо выполнить:

```
root@GlobusNode2:/# vim /etc/xinetd.d/gridftp
root@GlobusNode2:/# vim /etc/services
root@GlobusNode2:/# /etc/init.d/xinetd reload
Reloading internet superserver configuration: xinetd.
```

Установка менеджера распределенных ресурсов

TORQUE – менеджер распределенных ресурсов, являющийся одной из версий **PBS (Portable Batch System)**. Он используется для обеспечения удобного доступа к вычислительным ресурсам и служит заменой локальному планировщику заданий. Ввиду легкости установки и настройки **TORQUE** можно считать идеальным вариантом для установки в grid'е в качестве менеджера ресурсов. Вкупе с OpenMPI и с Globus Toolkit 5 он позволит распараллеливать задачи между несколькими машинами, являющимися элементами grid'a, и соединенными при помощи обычных сетевых протоколов.

Архитектура **TORQUE** следующая: На сервере работают демоны **pbs_server** (координирует работу всего **TORQUE**), **pbs_sched** (планировщик заданий, выделяет под конкретную задачу имеющиеся ресурсы) и **pbs_mom** (демон вычислительного узла, отвечающий за непосредственный запуск задачи на конкретном узле. На машинах-клиентах в зависимости от целей устанавливаются **pbs_mom** (если это должен быть один из вычислительных узлов) и/или **pbs_client** (если это клиентская машина, с которой будут запускаться задачи).

1. Настройка серверной машины

Скачиваем дистрибутив с сайта Cluster Resources и распаковываем его:

```
globus@GlobusNode:~$ tar xzvf torque-3.0.1.tar.gz
globus@GlobusNode:~$ cd torque-3.0.1/
```

Конфигурируем и собираем TORQUE. Эти действия необходимо выполнять с правами суперпользователя.

```
root@GlobusNode:/home/globus/torque-3.0.1# ./configure
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
.....
Ready for 'make'.

root@GlobusNode:/home/globus/torque-3.0.1# make
Making all in src
make[1]: Entering directory `/home/globus/torque-3.0.1/src'
.....

root@GlobusNode:/home/globus/torque-3.0.1# make install
Making install in src
make[1]: Entering directory `/home/globus/torque-3.0.1/src'
.....
```

Конфигурируем pbs_server: Для этого используем конфигурационный скрипт (предварительно остановив pbs сервер командой qterm -t quick).

```
root@GlobusNode:~$ qterm -t quick
root@GlobusNode:~$ /home/globus/torque-3.0.1/torque.setup
globus
initializing TORQUE (admin: globus@GlobusNode)
.....
```

Запускаем все необходимые на серверной машине службы: pbs_server, pbs_mom и pbs_sched.

```
root@GlobusNode:~$ pbs_server
root@GlobusNode:~$ pbs_mom
root@GlobusNode:~$ pbs_sched
```

Для удобства работы добавляем службы TORQUE в автоматический запуск в init.d. Тогда они будут стартовать вместе с системой и их не придется запускать вручную. Для каждой из служб копируем конфигурационный файл из contrib/init.d/ в /etc/init.d/, а затем обновляем информацию о службе с параметрами по умолчанию.

```
root@GlobusNode:~$ cp contrib/init.d/debian.pbs_mom
/etc/init.d/pbs_mom
root@GlobusNode:~$ cp contrib/init.d/debian.pbs_sched
/etc/init.d/pbs_sched
```

```

root@GlobusNode:~$ cp contrib/init.d/debian.pbs_server
/etc/init.d/pbs_server

root@GlobusNode:~$ update-rc.d pbs_mom defaults
root@GlobusNode:~$ update-rc.d pbs_sched defaults
root@GlobusNode:~$ update-rc.d pbs_server defaults

```

Теперь необходимо указать какие машины будут использоваться в качестве вычислительных нодов. Для этого необходимо добавить информацию о нодах в файл /var/spool/torque/server_priv/nodes

```

root@GlobusNode:~$ vim /var/spool/torque/server_priv/nodes

## This is the TORQUE server "nodes" file.
##
.....
## for more information, please visit:
##
## http://www.clusterresources.com/torquedocs/nodeconfi...
globusnode
globusnode2
root@GlobusNode:~$ qterm -t quick
root@GlobusNode:~$ pbs_server

```

2. Настройка вычислительных узлов

На вычислительном узле необходимо сконфигурировать демоны **pbs_mom** и **pbs_client**. Будем выполнять удаленную настройку с машины-сервера.

Для начала соберем необходимые пакеты:

```

root@GlobusNode:/home/globus/torque-3.0.1# make packages
Building packages from /home/globus/torque-3.0.1/tpackages
rm -rf /home/globus/torque-3.0.1/tpackages
.....
Building ./torque-package-doc-linux-i686.sh ...
Done.

```

The package files are self-extracting packages that can be copied and executed on your production machines. Use --help for options.

Теперь скопируем созданные пакеты на вторую машину с помощью команды scp.

```

root@GlobusNode:/# scp torque-package-mom-linux-i686.sh

```

```

globusnode2:/tmp/.
torque-package-mom-linux-i686.sh 100% 602KB 601.5KB/s 00:00
root@GlobusNode:/# scp torque-package-clients-linux-i686.sh
globusnode2:/tmp/.
root@globusnode2's password:
torque-package-clients-linux-i686.sh 100% 472KB 472.3KB/s
00:00

```

С помощью команды ssh также удаленно установим скопированные пакеты:

```

root@GlobusNode:/home/globus/torque-3.0.1# ssh globusnode2
/tmp/torque-package-mom-linux-i686.sh --install
root@globusnode2's password:
Installing TORQUE archive...
Done.
root@GlobusNode:/home/globus/torque-3.0.1# ssh globusnode2
/tmp/torque-package-clients-linux-i686.sh --install
root@globusnode2's password:
Installing TORQUE archive...
Done.

```

Добавим **pbs_mom** в автозапуск, как это было сделано на серверной машине:

```

root@GlobusNode:~$ scp contrib/init.d/debian.pbs_mom
globusnode2:/etc/init.d/pbs_mom
root@GlobusNode2:~$ update-rc.d pbs_mom defaults

```

Запустим **pbs_mom**, немного подождем и проверим работоспособность обоих нодов.

```

root@GlobusNode2:~$ pbs_mom
globus@GlobusNode2:~$ pbsnodes -a
globusnode
state = free
np = 1
ntype = cluster
.....

globusnode2
state = free
np = 1
ntype = cluster
.....

```

Оба нода успешно функционируют, их статус – free, т.е «свободен», готов к вычислениям. Но прежде, чем приступить к вычислениям, необходимо настроить права доступа.

3. Права доступа.

Следующим шагом будет выдача клиент-машине прав на добавление задач. Это можно сделать, выполнив на сервере команду `qmgr` со следующим параметром:

```
root@GlobusNode:/# qmgr -c 'set server submit_hosts =
globusnode2'
```

Теперь необходимо разрешить команде `scp` копирование результатов вычисления на машину пользователя. Поскольку все вычисления будут производиться от пользователя `quser`, то для него и будет создан доступ без пароля. Необходимо сгенерировать доверенный ключ и скопировать его на вычислительный узел. Проще всего это сделать командами `ssh-keygen` и `ssh-copy-id`.

```
quser@GlobusNode:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/quser/.ssh/id_rsa):
<жмем Enter>
Enter passphrase (empty for no passphrase): <жмем Enter>
Enter same passphrase again: <жмем Enter>
Your identification has been saved in /home/quser/.ssh/id_rsa.
Your public key has been saved in /home/quser/.ssh/id_rsa.pub.
```

```
...
quser@GlobusNode:~$ ssh-copy-id globusnode2
The authenticity of host 'globusnode2 (192.168.231.148)' can't
be established.
RSA key fingerprint is
cc:ca:74:65:34:86:6a:43:f8:d7:f5:0b:b1:02:35:34.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'globusnode2,192.168.231.148' (RSA)
to the list of known hosts.
quser@globusnode2's password:
Now try logging into the machine, with "ssh 'globusnode2'",
and check in:
```

```
.ssh/authorized_keys
```

to make sure we haven't added extra keys that you weren't expecting.

Теперь скопируем этот же ключ и на серверную машину, чтобы можно было помещать результаты вычислений в папку `/home/quser/.globus/job/` ... на сервере, из которой они потом уже будут пересылаться на вторую машину.


```
quser@GlobusNode:~/globus/job/globusnode$ ssh-copy-id  
globusnode  
quser@globusnode's password:  
Now try logging into the machine, with "ssh 'globusnode'", and  
check in:
```

```
.ssh/authorized_keys
```

to make sure we haven't added extra keys that you weren't expecting.

4. Установка jobmanager-pbs.

Теперь, когда TORQUE корректно установлен, можно приступить к конфигурированию Globus Toolkit. Оно сводится к сборке пакета gram5-pbs и установке его.

```
globus@GlobusNode:~$ cd gt5.0.3-all-source-installer/  
globus@GlobusNode:~/gt5.0.3-all-source-installer$ make  
gram5-pbs
```

Экспортируем путь, куда был установлен TORQUE и обновляем пакет Globus Toolkit. Наши настройки не будут удалены – в результате команды gpt-postinstall -force лишь будет перепроверен набор пакетов и установлен новый – gram5-pbs.

```
globus@GlobusNode:~/gt5.0.3-all-source-installer$ export  
PBS_HOME=/var/spool/torque  
globus@GlobusNode:~/gt5.0.3-all-source-installer$  
gpt-postinstall -force  
running /sandbox/globus-5/setup/globus/setup-globus-common..  
Changing to /sandbox/globus-5/setup/globus ]  
creating globus-sh-tools-vars.sh  
creating globus-script-initializer  
.....
```

Проверяем, правильно ли установился gram5-pbs:

```
globus@GlobusNode:~$ ls -l /sandbox/globus-5/etc/  
grid-services/  
итого 8  
lrwxrwxrwx 1 globus globus 15 Май 30 14:40 jobmanager -> job-  
manager-fork  
-rw-r--r-- 1 globus globus 152 Май 31 18:33 jobmanager-fork  
-rw-r--r-- 1 globus globus 151 Май 31 18:33 jobmanager-pbs
```

Jobmanager-pbs успешно установлен.

Тестирование grid'a

Прежде чем приступить к тестированию grid'a необходимо авторизоваться на сервере. Для этого заходим под пользователем *quser* и выполняем команду:

```
quser@GlobusNode2:~$ myproxy-logon -s globusnode
Enter MyProxy pass phrase: <qpassword>
A credential has been received for user quser in
/tmp/x509up_u1001.
```

Для добавления задач в Globus Toolkit 5.0.3 предусмотрены следующие команды:

- globusrun
- globus-job-run
- globus-job-submit

Самая простая из них - globus-job-run имеет следующую структуру:

globus-job-run <сервис выполнения> <путь к исполняемому файлу>

```
quser@GlobusNode2:~$ globus-job-run globusnode /usr/bin/whoami
quser
```

```
quser@GlobusNode2:~$ globus-job-run globusnode /bin/hostname
GlobusNode
```

Если задача выполняется длительное время, то есть возможность добавлять задачу в очередь и продолжить работу в консоли, а впоследствии получить результат вычислений. Для этого будет удобно воспользоваться командой globus-job-submit, которая после выполнения возвращает идентификатор задачи.

```
quser@GlobusNode2:~$ globus-job-submit globusnode /bin/date
https://globusnode:38129/16145810981669093481/1838060202852406
4396/
quser@GlobusNode2:~$ globus-job-get-output
https://globusnode:38129/16145810981669093481/1838060202852406
4396/
Tue May 31 14:33:25 MSD 2011
```

Также могут быть полезными команды `globus-job-status` (возвращает статус задачи), `globus-job-cancel` (удаляет задачу) и `globus-job-clean` (удаляет задачу и все созданные в процессе выполнения файлы). Все они как и `globus-job-get-output` в качестве параметра получают идентификатор задачи

Наиболее гибкой и функциональной является команда `globusrun`. Она может передавать специальные файлы (скрипты) написанные с использованием разметки RSL (Resource Specification Language), что позволит выполнять очень сложные команды. Также допускается употребление RSL-конструкций непосредственно в тексте команды (в этом случае конструкция окружается одинарными кавычками `'`). `Globusrun` может работать как `globus-job-submit`, т.е. возвращать идентификатор задачи, а может и дожидаться результатов выполнения, например:

```
quser@GlobusNode2:~$ vim a.rsl
&(executable=gsiftp://globusnode2/bin/hostname)
quser@GlobusNode2:~$ globusrun -f a.rsl -s -r globusnode
GlobusNode
```

```
quser@GlobusNode2:~$ globusrun -s -r globusnode
'&(executable=/bin/hostname)'
GlobusNode
```

Стандарт RSL включает в себя более 30 различных атрибутов, позволяющих очень гибко настроить задачу. Так, например, можно задать набор входных файлов, перенаправить поток ошибок и вывода в файл, указать максимальное допустимое время выполнения, ограничить программу по вычислительным ресурсам, подключить дополнительные библиотеки, указать метод запуска программы и многое другое. Пример:

```
quser@GlobusNode2:~$ cat b.rsl
&(rsl_substitution = (GRIDFTP_SERVER
"gsiftp://globusnode2:2811"))
(executable=/bin/ls) (arguments=-alt /tmp/my_echo)
(file_stage_in = ($(GRIDFTP_SERVER)/bin/echo /tmp/my_echo))
(file_clean_up=/tmp/my_echo)

quser@GlobusNode2:~$ globusrun -f b.rsl -s -r globusnode
quser@GlobusNode2:~$ cat ~/stdout
-rw-r--r-- 1 quser quser 24228 Nov 21 07:16 /tmp/my_echo
```

```
quser@GlobusNode2:~$ ls /tmp/my_echo
ls: /tmp/my_echo: No such file or directory
```

TORQUE как кластерное ПО предоставляет свой набор инструментов для управления задачами: основные команды это qsub (для добавления задач), qdel (для удаления) и qstat (статистика кластера). Также есть свой набор переменных среды, который позволяет управлять выполнением задачи (контроль ресурсов, входных данных и т.п.). Однако, использование TORQUE в совокупности с Globus Toolkit позволяет перенаправлять задачи на TORQUE, с использованием команд Globus и синтаксиса RSL, что дает мощный инструментарий для работы с процессами (как то распараллеливание с помощью MPI и работа с огромными объемами данных, под которые оптимизирован GridFTP).

Чтобы передать задачу TORQUE при запуске задачи надо указать jobmanager, который будет контролировать выполнение задачи. За выполнение локальных задач отвечает jobmanager-fork (используется по умолчанию), а за переадресацию кластерному ПО – jobmanager-pbs.

```
quser@GlobusNode2:~$ globus-job-run globusnode:/jobmanager-pbs
gsiftp://globusnode2/home/quser/helloworld.out
Hello, World!
```

```
quser@GlobusNode2:~$ globus-job-run globusnode: /jobmanager-pbs -np 2 -x '&(jobtype=mpi)' hellompi.out
Hello, MPI (rank: 0, count: 2)
Hello, MPI (rank: 1, count: 2)
```

```
quser@GlobusNode2:~$ qstat
Job id          Name    User Time Use S Queue
11.globusnode STDIN quser 00:00:00 C batch
12.globusnode STDIN quser 00:00:00 C batch
```

Заключение

В результате проведенных исследований были изучены особенности платформы Globus Toolkit 5.0.3, возможности ее применения совместно с кластерным ПО (TORQUE 3.0.1). Был произведен анализ существующих статей и публикаций по тематике данной работы, в результате которого были разработаны методические указания, позволяющие эффективно развернуть grid-вычисления.

На основании созданных методических указаний был развернут grid на двух виртуальных машинах, который был протестирован, оптимизирован и подготовлен к дальнейшему расширению.

Список литературы

1. Высокопроизводительные вычисления и кластерные системы. Обзор и сравнение: Globus Toolkit, Unicore, gLite. И. Парошина <http://www.-parallelcomputing.ru/archives/412>
2. Grid-вычисления. Википедия.
<http://ru.wikipedia.org/wiki/Грид>
3. Обзор Globus Toolkit
<http://gridclub.ru/software/gt.html>
4. Globus Toolkit 5.0.3 Quickstart
<http://www.globus.org/toolkit/docs/5.0/5.0.3/admin/quickstart/#quickstart>
5. TORQUE Administrator Guide
<http://www.adaptivecomputing.com/resources/docs/torque/index.php>
6. GT 5.0.3 GRAM5: User's Guide
<http://www.globus.org/toolkit/docs/5.0/5.0.3/execution/gram5/user>