D-Grid

InGrid

Innovative Grid-Entwicklungen für Ingenieurwissenschaftliche Anwendungen

AP 4.3: Sicherheits- und Vertrauensmodelle

D 4.3.2a-1
Pilotumgebung Enhanced Security - 1. Prototyp

Autoren	**Organisation**
Alfred Geiger	TS-SfR
Niels Fallenbeck	Universität Marburg
Hans-Joachim Picht	Universität Marburg
Matthias Schmidt	Universität Marburg
Christian Schridde	Universität Marburg
Matthew Smith	Universität Marburg

Internal Reviewer

Version history:

Version	Date
0.1	18. März 2007
0.2	19. März 2007
0.3	22 März 2007

Table of Contents

Übersicht

Aufbau des Gesamtsystems

Dieser Abschnitt gibt einen Überblick über alle Komponenten, die an der Ausführung eines Jobs beteiligt sind. Der genau Ablauf wird im Folgenden detailliert erläutert. Zur Illustration ist in Abbildung 1 die entstandene Infrastruktur dargestellt.

Der Anlaufpunkt für Endanwender, hier speziell Ingenieure, ist das in AP3 entwickelte Gridsphere-Portal. Um einen neuen Job abzuschicken, müssen dem Portal die zu bearbeitenden Daten übergeben werden. Alle anschließenden Vorgänge sind für den Benutzer transparent. Das Portal kann entweder direkt mit dem Globus Head Node kommunizieren (Pfad A in der Abbildung), oder über die Workflow Engine (Pfad B). Beide Computer sind in einer demilitarisierten Zone platziert.

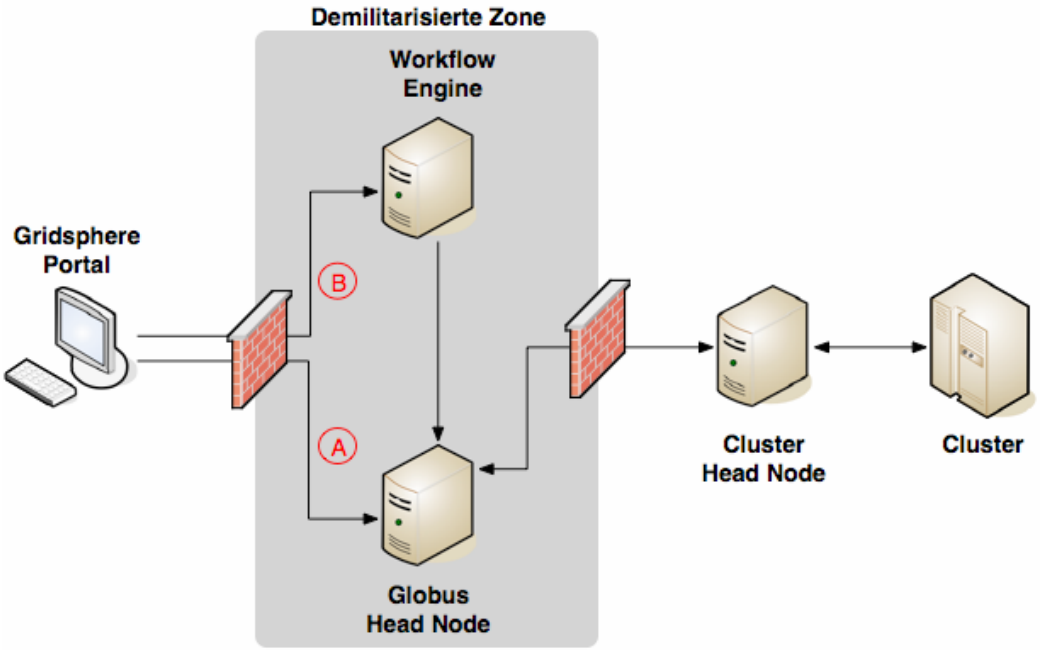

Abbildung 1 – Darstellung des erstellten Gesamtsystems

Die Workflow Engine dient zum Ausführen eines Workflows, der mit Hilfe des entsprechenden Editors erzeugt wurde. Der Benutzer kann Aktionen definieren, die sukzessive abgearbeitet werden. Jede Aktion besteht entweder aus einem einzelnen Webservice-Aufruf, oder aus mehreren, die parallel ablaufen. Meistens handelt es sich dabei um GRAM-Aufrufe. Der Editor ist an das System gekoppelt und kann so den Fortschritt live visualisieren.

Das Portal sorgt für den Transfer der Job-Daten via GridFTP und ruft danach den an der Universität Siegen erstellten Casts-Service auf dem Globus-Rechner auf. Der Service erzeugt für die für die Berechnung notwendigen Dateien und ruft den GRAM-Service auf. GRAM verwendet den Job-Manager aus der *Fence Suite,* um mit Hilfe der Dienste die Daten zum Cluster Head Node zu übertragen. Nachdem der Job an der Xen Grid Engine angekommen ist, wird er bearbeitet. Die Resultate werden nach erfolgreicher Berechnung zurück zum Globus-Rechner kopiert, von dem sie der Benutzer abholen kann.

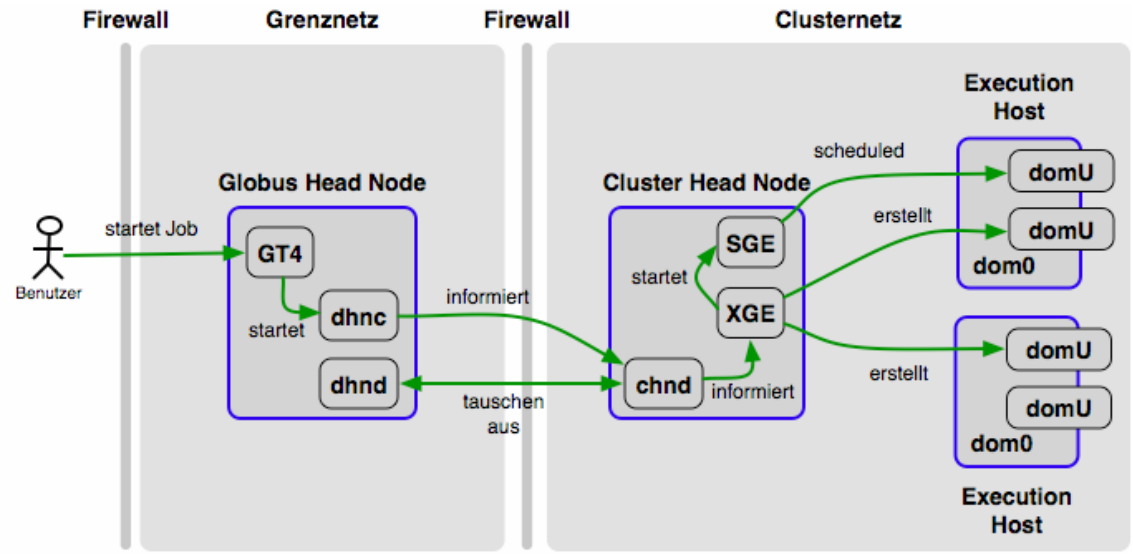

Abbildung 2

Während die Sicherheit bei der Übertragung der sensiblen Daten (Modelle, Modellierungswerte, ...) vom Gridsphere-Portal zum Grid Head Node durch die Globus Middleware gewährleistet ist, muss die sichere Handhabung auch nach der Übergabe vom Grid- an das Cluster-System gewährleistet werden.

Es muss sichergestellt werden, dass die auf den Clusterknoten laufenden Jobs keine sensitive Daten über andere Jobs Prozesse sammeln können, die ebenfalls dem Cluster-(Knoten) zugeordnet wurden. Um dieses zu verhindern, muss auf den Knoten eine sichere Umgebung geschaffen werden, die solche Angriffe von vornherein ausschließt. Dafür werden für jeden Grid Job virtuelle Maschinen im Cluster erzeugt und die Berechnung der Jobs in diesen ausgeführt. Während der Berechnung läuft der Job in einer abgeschotteten Umgebung und ist von anderen zeitgleich laufenden Tasks durch die Virtualisierungstechnologie abgeschottet.

Zusätzlich zu dieser Sicherheitskomponente bieten virtuelle Maschinen den Vorteil einer speziell auf diesen Job abgepassten Ausführungsumgebung, die frei ist von Seiteneffekten durch andere Softwareinstallationen.

Jeder Job, der am Grid Head Node ankommt, wird mittels *Fence* an die XGE auf dem Cluster Head Node weitergereicht. Diese ermittelt für den Job eine virtuelle Maschine, erzeugt diese aus einem Template und startet sie auf den Execution Hosts. Anschließend wird der Job über die Sun Grid Engine in die zugehörige(n) virtuelle(n) Maschine(n) geschickt und dort berechnet. (siehe Abbildung 2)

Dokumentation

Installation der Xen Grid Engine

Die Xen Grid Engine (XGE) besteht aus zwei Komponenten:

- Kernkomponente, die auf dem Cluster Head Node läuft
- Steuermodule auf den Execution Hosts

1.1.1 Installation der Kernkomponente auf dem Head Node

Die Kernkomponente ist in Java implementiert und auf jedem unixoiden Betriebssystem (getestet mit Sun Solaris und Linux) lauffähig, das folgende Voraussetzungen erfüllt:

- Java Development Kit, Version 1.5 oder neuer
- Funktionsfähige SSH-Installation
- Funktionsfähige Installation der Sun Grid Engine

Als erstes wird das Archiv der Kernkomponente auf dem Cluster Head Node entpackt. Es muss eine systemweite Variable `$XGE_HOME` existieren, die auf dieses Verzeichnis zeigt. Anschließend kann man die Software kompilieren:

```
# cd $XGE_HOME/*
# javac -d bin -sourcepath src/ src/xge/XGE.java
# javac -d bin \
    -cp "$XGE_HOME/lib/jdom.jar:$XGE_HOME/lib/drmaa.jar:$XGE_HOME/bin" \
    -sourcepath src/ src/xged/XGEd.java
```

Die zweite Zeile kompiliert die Kommandozeilentools der *XGE*.
Die letzte Zeile schließlich kompiliert *XGEd*, die Kernkomponente der Xen Grid Engine. Nach dem Kompilieren kann man diese Kernkomponente starten:

```
# cd $XGEHOME/bin
# java \
    -cp "$XGE_HOME/lib/jdom.jar:$XGE_HOME/lib/drmaa.jar:$XGE_HOME/bin" \
    xged.XGEd
```

Im Verzeichnis `$XGE_HOME/tools/` findet sich das Programm `xge`. Dieses kann man an eine geeignete Stelle im System (z.B. `/sbin`) kopieren, um die Software einfach zu bedienen.

Die Kommunikation mit den Execution Hosts findet über SSH statt. Weil bei der Verarbeitung kein Passwort abgefragt werden soll, wird das Public Key Verfahren zur Authentifizierung verwendet. Auf dem Cluster Head Node muss daher ein öffentlicher Schlüssel erzeugt werden.

```
# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
55:73:a3:13:d2:85:de:23:1d:fb:70:f9:43:cf:85:21 root@headnode
```

Schließlich muss sichergestellt werden, dass das Verzeichnis, in dem *Fence* die Dateien der Grid Jobs abgelegt, angelegt ist. Standardmäßig ist `/opt/xge/xen/data` dafür vorgesehen.

```
# mkdir –p /opt/xge/xen/data
```

Dieses Verzeichnis kann beliebig gewählt werden, vorausgesetzt die Konfigurationsdatei der XGE wird angepasst.

Konfiguration der Hauptkomponente

Die Konfigurationsdatei der Hauptkomponente findet sich in `$XGE_HOME/etc/rc.xml`.

```
<?xml version='1.0'?>
<configuration>
        <controld>
                <enabled>true</enabled>
                <port>2322</port>
                <bind_to>*</bind_to>
        </controld>

        <grid>
                <watchdog>
                        <jobdir>/opt/xge/xen/data</jobdir>
                        <check_period>15</check_period>
                        <threshold>7200</threshold>
                </watchdog>
        </grid>

        <cluster>
                <waiting_queue>parking</waiting_queue>
                <watchdog>
                        <check_period>15</check_period>
                </watchdog>
                <sge>
                        <bin_dir>/opt/sge/bin/sol-x86/</bin_dir>
                </sge>
        </cluster>

        <xen>
                <path>/opt/xen</path>
                <image_path>/opt/xge/xen/image</image_path>
                <default_image>vmdefault</default_image>
                <config_path>/opt/xge/xen/config</config_path>
        </xen>

        <files>
                <hostlist>/etc/hostlist.xml</hostlist>
                <usermap>/etc/users.xml</usermap>
                <groupmap>/etc/groups.xml</groupmap>
        </files>

        <nfs>
                <server>10.0.1.4</server>
                <path>/media/nfs</path>
        </nfs>
</configuration>
```

`controld` ist die Telnet-Schnittstelle zu den Userland-Tools. Diese sollte man an eine Adresse binden, in der Regel an `localhost` bzw. `127.0.0.1`.
Im Abschnitt `grid` konfiguriert man den Watchdog, der nach neuen Jobs sucht. Dabei lässt sich das Verzeichnis angeben, in dem neue Jobs von *Fence* gespeichert werden, sowie den Abstand zwischen den Suchvorgängen.
Im Abschnitt `cluster` wird die Queue konfiguriert, in der neue Jobs standardmäßig submitted werden. Außerdem kann angegeben werden, in welchen Zeitabständen der Cluster Watchdog ausgeführt wird.

Im Abschnitt `xen` wird der Xen-spezifische Teil der Software konfiguriert. Neben dem Pfad zu den Xen-Binaries muss auch der Pfad zu den Xen-Images und den Image-Konfigurationsdateien angegeben werden. Hier findet sich auch die Angabe eines Standard-Images, in dem alle Jobs ausgeführt werden, für die kein eigenes Image für eine virtuelle Maschine konfiguriert ist.

Im `files`-Bereich finden sich die Verweise auf Dateien, in denen sich die Benutzer der XGE und deren Gruppenzuordnungen finden. Außerdem enthält es die Liste aller Execution Hosts, die er Kontroller der XGE-Software unterstehen.

Im Abschnitt `nfs` wird schließlich der clusterweite NFS-Server konfiguriert, auf dem die Images für die virtuellen Maschinen erzeugt und den DomUs zur Verfügung gestellt werden.

1.1.2 Userland-Tools

Um die XGE zu bedienen, dient das Programm `xge` aus dem Ordner `$XGE_HOME/tools/`. Dieses verbindet sich auf die Telnet-Schnittstelle der XGE-Software, welche die Useranfragen entgegennimmt und auswertet. Es verfügt über eine Online-Hilfe, die der User bei Bedarf aufrufen kann mit

```
# xge help
```

1.1.3 Installation der Komponenten auf den Execution Hosts

Damit XGE mit den Execution Hosts (getestet mit Linux) zusammenarbeiten kann, müssen auf diesen Computern folgende Bedingungen erfüllt sein:

- Funktionsfähige Xen-Installation

Das Archiv für die Execution Host Software sollte auf dem Cluster Head Node in das Verzeichnis /opt/xge entpackt und von diesem über NFS exportiert werden. Auf den Execution Hosts sollte dieser Export ebenfalls am Mountpoint /opt/xge/ eingehängt werden. In diesem Verzeichnis finden sich verschiedene Programme für die Handhabung virtueller Maschinen und die Image-Dateien für die virtuellen Maschinen.

Die Xen Grid Engine muss jeden Execution Host mit SSH erreichen können. Damit dies ohne Passworteingabe geschehen kann, muss die Public Key Authentifizierung eingerichtet werden. Dazu kopiert man den auf dem Head Node erzeugen Public Key nach `/root/.ssh/authorized_keys` auf den Execution Hosts.

Installation von Fence

Fence besteht aus mehreren Komponenten, die entweder auf dem Globus Head Node oder dem Cluster Head Node installiert werden. Folgende Voraussetzungen müssen auf dem Globus Head Node erfüllt sein:

• Unix-kompatibles Betriebssystem (getestet mit Linux, FreeBSD und Solaris)
• Globus Toolkit Version ≥ 4.0.3
• C-Entwicklungsumgebung (getestet mit GCC und GNU Make)

Folgende Voraussetzungen müssen auf dem Cluster Head Node erfüllt sein:

• Unix-kompatibles Betriebssystem (getestet mit Linux, FreeBSD und Solaris)
• Xen Grid Engine
• C-Entwicklungsumgebung (getestet mit GCC und GNU Make)

Der Übersichtlichkeit halber werden einige Ausgaben auf der Konsole, die nur informellen Zwecken dienen, gekürzt dargestellt. Weiterhin ist die Reihenfolge der Installation zwingend, da einige Schritte aufeinander aufbauen.

1.1.4 DMZ Head Node Client

Als erstes wird das Archiv ausgepackt und alle Komponenten übersetzt. Der Einfacheit halber werden alle Schritte als User *root* ausgeführt.

```
# tar xfz daemons.tgz
# cd code
# make
make --no-print-directory -C common
...
```

Für den Betrieb beider Dienste auf dem Globus Head Node ist ein eigener Benutzer inklusive Gruppe und ein spezielles Verzeichnis nötig. Je nach verwendetem Betriebssystem bedürfen die `add*`-Befehle möglicherweise einer Anpassung.

```
# addgroup --gid 2048 dhnd
# adduser --system --home /usr/dhnd --disabled-password \
--disabled-login --gid 2048 dhnd
# mkdir -p /usr/dhnd/data
# chown -R dhnd:dhnd /usr/dhnd
```

Abschließend wird der dhnc inklusive Manpage installiert:

```
# make install
install -m 755 -o dhnd -g dhnd ../bin/dhnc /usr/bin/
install -m 644 dhnc.1 /usr/share/man/man1/
```

1.1.5 DMZ Head Node Daemon

Da der benötigte Benutzer schon angelegt ist und der dhnd bereits im letzen Abschnitt übersetzt wurde, muss er installiert werden:

```
# cd ../dhnd/
# make install
install -m 755 ../bin/dhnd /usr/bin/
install -m 644 dhnd.8 /usr/share/man/man8/
```

Jetzt kann der Daemon gestartet werden und ist einsatzbereit. Nach erfolgreichem Start meldet sich der Daemon im lokalen syslog.

```
# ./dhnd.sh start
Start dhnd
# tail /var/log/syslog | grep dhnd
dhnd[15518]: Starting...
dhnd[15520]: Chroot to /usr/dhnd and user dhnd successfull
dhnd[15521]: Chroot to /usr/dhnd and user dhnd successfull
```

1.1.6 Job-Manager

Zur Integration des Job-Managers muss das Job-Manager-Skript und ein Setup-Skript installiert werden. Beide Skripte liegen mit anderen für die Installation notwendigen Dateien in je einem Verzeichnis. Die Verzeichnisse müssen in Archive gepackt werden, damit diese in das Paketmanagement von Globus aufgenommen werden können. Alle Schritte in diesen Abschnitt müssen als Globus-Benutzer ausgeführt werden.

```
$ cd ../job-manager/
$ tar cfz globus_gram_job_manager_setup_sge-1.1.tar.gz \
globus_gram_job_manager_setup_sge-1.1/
$ tar cfz globus_scheduler_event_generator_sge_setup-1.1.tar.gz \
globus_scheduler_event_generator_sge_setup-1.1/
$ cd ..
```

Die entstandenen Archive und die Archive im contrib-Verzeichnis werden nun mit gpt-build installiert. *flavor* muss hierbei durch die installierte Globus-Architektur ersetzt werden, in den meisten Fällen ist dies gcc32dbg. Die Dateien im contrib Verzeichnis sind für die komplette Installation notwendig, wurde aber nicht im Rahmen dieser Arbeit geschrieben. Sie stammen von David McBride vom London e-Science Centre und sind unter der GNU General Public License als Open Source freigegeben.

```
$ gpt-build job-manager/globus_gram_job_manager_setup_sge-1.1.tar.gz
$ gpt-build job-manager/globus_scheduler_event_generator_sge_setup-1.1.tar.gz
$ gpt-build contrib/globus_scheduler_event_generator_sge-1.1.tar.gz flavor
$ gpt-build contrib/globus_wsrf_gram_service_java_setup_sge-1.1.tar.gz
```

Abschließend müssen die frisch installierten Komponenten noch durch Globus initialisiert werden.

```
$ gpt-postinstall
```

1.1.7 Cluster Head Node Daemon

Zur Installation des chnd muss das Fence-Archiv auf den Cluster Head Node kopiert und dort ausgepackt werden. Danach werden alle Dienste übersetzt. Der Einfacheit halber werden alle Schritte als User *root* ausgeführt.

```
# tar xfz daemons.tgz
# cd code
# make
make --no-print-directory -C common
...
```

Genau wie für den dhnd ist auch hier ein eigener Benutzer inklusive Gruppe für den Betrieb notwendig. Das Home-Verzeichnis des Benutzers entspricht dem Basis-Verzeichnis der Xen Grid Engine, das vom Administrator erfragt werden muss und hier mit `$xge_home` benannt ist.

```
# cd chnd/
# addgroup --gid 2048 chnd
# adduser --system --home $xge_home --disabled-password \
--disabled-login --gid 2048 chnd
```

Jetzt kann der Daemon installiert und gestartet werden. Nach erfolgreichem Start meldet sich der chnd über syslog :

```
# make install
install -m 755 ../bin/chnd /usr/bin/
install -m 644 chnd.8 /usr/share/man/man8/
# ./chnd.sh start
Start chnd
# tail /var/log/syslog | grep chnd
chnd[15599]: Starting...
chnd[15602]: Privilege drop to chnd successfull
chnd[15601]: Privilege drop to chnd successfull
```

1.1.8 Technische Details

Der dhnd bindet sich standardmäßig an Port 2324. Das Basisverzeichnis ist `/usr/dhnd` und der Benutzer dhnd. Alle Werte können im Quelltext geändert werden.
Der chnd bindet sich standardmäßig an Port 2323 und arbeit im Verzeichnis `/opt/xge/xen/data`, dass das Basisverzeichnis der Xen Grid Engine ist. Die lokale Verbindung zur Xen Grid Engine erwartet der chnd auf Port 2322. Alle diese Werte können im Quelltext geändert werden.

1.1.9 Handhabung

Nach erfolgreicher Installation können neue Jobs über den Job-Manager verschickt und mit Hilfe der Xen Grid Engine in einer virtualisierten Umgebung laufen. Um den Job-Manager zu verwenden, muss SGE als Factory Type angegeben werden. Im folgenden Beispiel wird ein Job gestartet, der `/bin/false` ausführt:

```
$ globusrun-ws -submit -factory pc12714 -Ft SGE -c -- /bin/false
Submitting job...Done.
Job ID: uuid:7fef6276-d320-11db-9dbc-0011118432c8
...
```

Wurde der Job erfolgreich an die Xen Grid Engine übergeben, sind auf dem Globus und Cluster Head Node entsprechende Einträge der Daemons im syslog zu sehen:

```
dhnd[31324]: GET request from 192.168.121.51
dhnd[31324]: Write 114 bytes
dhnd[31323]: GET request from 192.168.121.51
dhnd[31323]: Write 83 bytes
...
chnd[12654]: 137.248.121.151 requested
7fef6276-d320-11db-9dbc-0011118432c8#job.sh
chnd[12654]: Finished writing file
/opt/xge/xen/data/7fef6276-d320-11db-9dbc-0011118432c8/job.sh with 114 bytes
...
```

Der Status eines Jobs kann mit Hilfe des dhnc -Kommandozeilenmodus abgefragt werden:

```
$ dhnc -s 7fef6276-d320-11db-9dbc-0011118432c8
Job: 7fef6276-d320-11db-9dbc-0011118432c8 IP: 192.168.121.51
16
Job pending
```

Der dhnc kann ebenfalls dazu genutzt werden, einen laufenden Job über die Xen Grid Engine aus dem Cluster zu löschen:

```
$ dhnc -d 7fef6276-d320-11db-9dbc-0011118432c8
Job: 7fef6276-d320-11db-9dbc-0011118432c8 IP: 192.168.121.51
scheduled job for deletion: true
```