NORDUGRID

NORDUGRID-XXXXXXX-NN

19/3/2009

Administrator's and user's manual of the ARC storage system

Zsombor Nagy[*]

Jon Nilsen[†]

Salman Zubair Toor [‡]

[*]zsombor@niif.hu

[†]j.k.nilsen@usit.uio.no

[‡]salman.toor@it.uu.se

Contents

Chapter 1

Administrator's manual

1.1 Quick start guide

In this section we will follow the installation of the ARC storage system on a freshly installed debian lenny machine.

Get the latest ARC code from the subversion (`http://svn.nordugrid.org/`):

```
$ svn co http://svn.nordugrid.org/repos/nordugrid/arc1/trunk arc1
```

Let's check the README file for all the dependencies, and install them:

```
$ sudo aptitude install build-essential
$ sudo aptitude install autoconf
$ sudo aptitude install gettext
$ sudo aptitude install cvs
$ sudo aptitude install libtool
$ sudo aptitude install pkg-config
$ sudo aptitude install libglibmm-2.4-dev
$ sudo aptitude install python-dev
$ sudo aptitude install swig
$ sudo aptitude install libxml2-dev
$ sudo aptitude install libssl-dev
```

Run the `autogen` and `configure` scripts (you may want to specify target directories for the installation with the `--prefix` option):

```
$ cd arc1
$ ./autogen.sh
$ ./configure --disable-java --disable-a-rex-service --disable-isi-service \
    --disable-charon-service --disable-compiler-service \
    --disable-paul-service --disable-sched-service

Unit testing:       yes
Java binding:       no
Python binding:     yes (2.5)

Available third-party features:

RLS:                no
GridFTP:            no
LFC:                no
```

```
RSL:                no
SAML:               no
MYSQL CLIENT LIB:   no
gSOAP:              no

Included components:
A-Rex service:      no
ISI service:        no
CHARON service:     no
HOPI service:       yes
SCHED service:      no
STORAGE service:    yes
PAUL service:       no
SRM client (DMC):   no
GSI channel (MCC):  no
```

We can start to compile it:

```
$ make
```

And then install it (you need to be root if you want to install it to the default location):

```
$ sudo make install
```

Maybe we should run `ldconfig` to refresh the list of libraries:

```
$ ldconfig
```

We can check if the python packages are installed correctly:

```
$ python
Python 2.5.2 (r252:60911, Jan  4 2009, 17:40:26)
[GCC 4.3.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import arc
>>> import storage
>>> import arcom
>>>
```

If there is no error on importing any of the packages, we are good to go.

It is important to syncronize the time of the machine - the storage system will make decisions based on differences between timestamps, when we deploy it on multiple machines, different times would cause problems.

We need host (and user) certificates to run the system. For testing purposes we can use the NorduGrid InstantCA solution which is capable of creating a demo-CA with short lifetime. The URL of the InstantCA service is `https://vls.grid.upjs.sk/CA/instantCA`, let's create 3 user and 3 host certificates. (e.g. we can set the Organization Name to 'knowarc', the Common name of the CA to 'storage-test', the name of the users could be 'penny', 'billy' and 'hammer', the name of the hosts could be 'upsilon', 'omicron' and 'theta', and we need a password for the CA and passwords for the user certificates, both should be at least 4 characters long).

Let's download the generated certificates and untar the archive file to see its contents:

```
$ ls -lR
.:
```

```
total 28
drwxr-xr-x 2 zsombor zsombor 4096 2009-03-19 12:11 CA
-rw-r--r-- 1 zsombor zsombor  963 2009-03-19 12:11 ca.key
-rw-r--r-- 1 zsombor zsombor  786 2009-03-19 12:11 ca.pem
drwxr-xr-x 2 zsombor zsombor 4096 2009-03-19 12:11 hostCerts
-rw-r--r-- 1 zsombor zsombor 1044 2009-03-19 12:11 readme
-rw-r--r-- 1 zsombor zsombor    3 2009-03-19 12:11 serial.srl
drwxr-xr-x 2 zsombor zsombor 4096 2009-03-19 12:11 userCerts

./CA:
total 8
-rw-r--r-- 1 zsombor zsombor 786 2009-03-19 12:11 66fe707c.0
-rw-r--r-- 1 zsombor zsombor 139 2009-03-19 12:11 66fe707c.signing_policy

./hostCerts:
total 24
-rw-r--r-- 1 zsombor zsombor 786 2009-03-19 12:11 hostcert-omicron.pem
-rw-r--r-- 1 zsombor zsombor 782 2009-03-19 12:11 hostcert-theta.pem
-rw-r--r-- 1 zsombor zsombor 786 2009-03-19 12:11 hostcert-upsilon.pem
-rw-r--r-- 1 zsombor zsombor 887 2009-03-19 12:11 hostkey-omicron.pem
-rw-r--r-- 1 zsombor zsombor 891 2009-03-19 12:11 hostkey-theta.pem
-rw-r--r-- 1 zsombor zsombor 887 2009-03-19 12:11 hostkey-upsilon.pem

./userCerts:
total 24
-rw-r--r-- 1 zsombor zsombor 778 2009-03-19 12:11 usercert-billy.pem
-rw-r--r-- 1 zsombor zsombor 778 2009-03-19 12:11 usercert-hammer.pem
-rw-r--r-- 1 zsombor zsombor 778 2009-03-19 12:11 usercert-penny.pem
-rw-r--r-- 1 zsombor zsombor 963 2009-03-19 12:11 userkey-billy.pem
-rw-r--r-- 1 zsombor zsombor 963 2009-03-19 12:11 userkey-hammer.pem
-rw-r--r-- 1 zsombor zsombor 951 2009-03-19 12:11 userkey-penny.pem
```

We have certificate and key files for all the hosts and users, and we have the CA file with the proper hashed name. There is no rule where to put these certificate, but now we will put them in `/etc/grid-security`. Let's create this directory, and a subdirectory called `certificates`. We will use the 'theta' host certificates for this machine, let's copy it there:

```
$ sudo cp hostCerts/hostcert-theta.pem /etc/grid-security/hostcert.pem
$ sudo cp hostCerts/hostkey-theta.pem /etc/grid-security/hostkey.pem
$ sudo cp CA/* /etc/grid-security/certificates/
$ ls -lR /etc/grid-security/
/etc/grid-security/:
total 12
drwxr-xr-x 2 root root 4096 2009-03-19 12:36 certificates
-rw-r--r-- 1 root root  782 2009-03-19 12:35 hostcert.pem
-rw-r--r-- 1 root root  891 2009-03-19 12:36 hostkey.pem

/etc/grid-security/certificates:
total 8
-rw-r--r-- 1 root root 786 2009-03-19 12:36 66fe707c.0
-rw-r--r-- 1 root root 139 2009-03-19 12:36 66fe707c.signing_policy
```

And I choose a user certificate and put it into the `~/.arc` directory, conveniently removing the password from the key file:

```
$ mkdir ~/.arc
$ cp userCerts/usercert-billy.pem ~/.arc/usercert.pem
$ openssl rsa -in userCerts/userkey-billy.pem -out ~/.arc/userkey.pem
$ chmod 600 ~/.arc/userkey.pem
$ ls -l ~/.arc
total 8
-rw-r--r-- 1 zsombor zsombor 778 2009-03-19 13:06 usercert.pem
-rw------- 1 zsombor zsombor 891 2009-03-19 13:09 userkey.pem
```

The storage system runs withen the `arched` hosting environment daemon, which needs a configuration file which describes which services we want to run, and on which ports do we want to listen, etc. Let's copy the template configuration files to the `/etc/arc` directory:

```
$ sudo mkdir /etc/arc
$ sudo cp arc1/src/services/storage/storage_service.xml.example /etc/arc/storage_service.xml
$ sudo cp arc1/src/services/storage/clientsslconfig.xml /etc/arc
$ ls -l /etc/arc
total 12
-rw-r--r-- 1 root root  259 2009-03-19 13:26 clientsslconfig.xml
-rw-r--r-- 1 root root 4606 2009-03-19 13:25 storage_service.xml
```

The template configuration file specifies several directories where the services can store their data. We will create these directories at `/var/spool/arc`, but they can be created anywhere else.

```
$ sudo mkdir /var/spool/arc/ahash_data
$ sudo mkdir /var/spool/arc/shepherd_data
$ sudo mkdir /var/spool/arc/shepherd_store
$ sudo mkdir /var/spool/arc/shepherd_transfer
$ ls -l /var/spool/arc
total 16
drwxr-xr-x 2 root root 4096 2009-03-19 13:42 ahash_data
drwxr-xr-x 2 root root 4096 2009-03-19 13:42 shepherd_data
drwxr-xr-x 2 root root 4096 2009-03-19 13:42 shepherd_store
drwxr-xr-x 2 root root 4096 2009-03-19 13:42 shepherd_transfer
```

In this deployment we plan to run the daemon as root, so it is OK that the root owns these directories. If we create these directories somewhere else, we would need to modify the paths in the `storage_service.xml` file.

If we put the host certificates and the CA certificate somewhere else than `/etc/grid-security` then we would need to modify the `storage_service.xml` (look for `KeyPath`, `CertificatePath` and `CACertificatesDir`), and we would need to modify `/etc/arc/clientsslconfig.xml` as well. This file contains the credentials which is used by the services when they connect to an other service. If we put `clientsslconfig.xml` somewhere else than `/etc/arc` then we would need to modify all the `ClientSSLConfig` tags in the `storage_service.xml`.

Let's run the `arched` daemon with this config (specified by the `-c` option) first in the foreground to see immediately if everything is right:

```
$ sudo /usr/local/sbin/arched -c /etc/arc/storage_service.xml -f
```

The template configuration file specifies a loglevel which only prints error messages, so nothing should be written on the screen now. The `arched` daemon should now listen on the ports: 60001 for our data transfer service called Hopi, and 60000 for all the other services:

```
$ netstat -at | grep LISTEN
tcp        0      0 *:60000                 *:*                     LISTEN
tcp        0      0 *:60001                 *:*                     LISTEN
```

Now it is time to set up or prototype client tool to access the system. It is currently called `arc_storage_cli` and it is by default installed to `/usr/local/bin`. We have to tell this CLI tool where it can find our user credentials, and a URL of a Bartender service. The Bartender service is the front-end of the storage system. We can specify these things by environment variables, or by a configuration file called `~/.arc/client.xml`. Let's create this `client.xml` file:

```
$ cat ~/.arc/client.xml
<ArcConfig>
  <KeyPath>~/.arc/userkey.pem</KeyPath>
  <CertificatePath>~/.arc/usercert.pem</CertificatePath>
  <CACertificatesDir>/etc/grid-security/certificates</CACertificatesDir>
  <BartenderURL>https://localhost:60000/Bartender</BartenderURL>
</ArcConfig>
```

Now we can use the `arc_storage_cli` to access our local deployment. Let's do a 'list' on the root collection ('/'):

```
$ arc_storage_cli
Usage:
  arc_storage_cli <method> [<arguments>]
Supported methods: stat, make[Collection], unmake[Collection], list, move, put[File], get[File], del
Without arguments, each method prints its own help.

$ arc_storage_cli list
- The URL of the Bartender: https://localhost:60000/Bartender
- The key file: ~/.arc/userkey.pem
- The cert file: ~/.arc/usercert.pem
- The CA dir: /etc/grid-security/certificates
Usage: list <LN> [<LN> ...]

$ arc_storage_cli list /
- The URL of the Bartender: https://localhost:60000/Bartender
- The key file: ~/.arc/userkey.pem
- The cert file: ~/.arc/usercert.pem
- The CA dir: /etc/grid-security/certificates
- Calling the Bartender's list method...
[2009-03-19 18:37:12] [Arc.Loader] [ERROR] [12491/152991544] Component tcp.client(tcp) could not be
[2009-03-19 18:37:12] [Arc.Loader] [ERROR] [12491/152991544] Component tls.client(tls) could not be
[2009-03-19 18:37:12] [Arc.Loader] [ERROR] [12491/152991544] Component http.client(http) could not b
[2009-03-19 18:37:12] [Arc.Loader] [ERROR] [12491/152991544] Component soap.client(soap) could not b
ERROR: Wrong status from server.
Maybe wrong URL: 'https://localhost:60000/Bartender'
```

Chapter 2

User's manual