XSEDE Genesis II Client/Container Installation

Andrew Grimshaw and Karolina Sarnowska-Upton

XSEDE
Extreme Science and Engineering Discovery Environment
Goals and Prerequisites

• Goals
  – Learn how to install an XSEDE Genesis II client & container

• Prerequisites
  – A Grid Interface Unit (GIU) - a Linux, Windows, or MacOS machine* with access to the internet
Agenda

• What are we installing?
• Questions you’ll be asked during installation
• The installation process
• Post-installation notes
What are we installing?

• Genesis II software stack – implements
  – Campus bridging component
  – GFFS
  – Meta-scheduler
  – BES implementation for campuses

• Installer installs client software and/or container
  – Client software includes CLI, API, GUI, and GFFS FUSE driver
  – Container is a server that implements XES services
Why Install A Container?

• To share local resources within XSEDE:
  – *compute resources* such as desktops or clusters
  – *data resources* such as files and directory trees visible from a local machine, zip files, or CIFS mount
  – *local identity resources*
Simple Grid Interface Unit

Figure 24. A simple Grid Interface Unit consisting of a host with a Web Service container. XSEDE services in the container interact with the local queuing systems, the local hard disk to store their state in a RDBMS, and the local distributed file systems.
Grid Interface Unit Minimums

• **Linux Node Requirements**
  – 4 GB Memory – preferred 24 GB
  – 1 core
  – 50 GB direct attached storage – speed matters, persistent state kept in on-disk database
  – Gigabit connection to site wide file system and external network – prefer 10 gigabit
  – Prefer UPS power that outlasts transients and lasts as long as network still functioning
  – Access to file system shared by compute nodes

• **Environment Configuration Requirements**
  – File system support for extended access control lists
  – Local account that a grid administrator can ssh onto
  – Ability to submit, stat, and delete jobs from the resource queues
  – Require local queue configuration for the queues, e.g., queuing system used (PBS, SGE, etc), queue names, wallclock limits, properties of queues, special flags, etc.
  – Require local HPC node configuration information (number of cores, memory per node, number of nodes, basic stuff)
A Note on the UNICORE 6 Container

• XSEDE is installing the UNICORE 6 container to implement BES services at the centers
  – They are interface compatible with the Genesis II BES containers – mixed systems are fine

• This tutorial does not cover installing a UNICORE 6 tutorial
  – There will be a workshop for SP system administrators on installing the UNICORE 6 container in late September or early October
Things You’ll Need Before Starting a Container on Your Home Resource

- Select a Grid Interface Unit (GIU), a Windows, Linux, or MacOS host, which can access resources you want to share
- Ensure that GIU has a public IP address
  - Use of GIUs behind a NAT is possible, but setup is more complex and will not be discussed here
- Choose open port number (default is 18443)
  - Update firewall rules as necessary to allow incoming TCP traffic
- Ensure there is sufficient storage on a locally attached disk
  - Container uses an on-disk transactional database
  - A slow remote disk will significantly reduce performance
  - For a container managing a moderate number of resources 5GB is adequate
- If you are going to share cluster or supercomputer resources, the GIU must share a file system with the compute nodes and it must have sufficient space for input and output files as well as temporary files used by jobs
- Chose a local account that will run the grid container
  - Account does not require root privilege unless running on Windows host
- If you are going to share local user directories, it is best if extended access control lists are turned on in your local environment
- A grid account for the user/group that will “own” container
Download Container Installation Bundle

• Installers are specific to target architecture/operating system and grid deployment
  – Reduces configuration burden to deploy new containers
  – Helps avoid deployment mistakes and mismatches between different containers within grid system

The XSEDE container installer is available for Linux. It is available in the Increment 1 TRR resources.
Container Installation

• Turns on **HTTPS/SSL communication**
• Deploys and configures **certificates** and **trust stores** to allow grid containers within one grid system to authenticate and/or trust each other
• Downloads information about **global namespace root** (i.e. “/”) for target grid system and store sit in local state directory
  – This allows clients and container to communicate with proper grid system without manual user intervention
• Executes code to generate new unique grid container certificate.
The Installation Process Questions

• OK to install?
  – License follows Apache license agreement
The Installation Process Questions

• Installation directory path?
  – where code and configuration files will be placed

**Directory to store container state will be created at ~/.genesisII-2.0**
The Installation Process Questions

• Deployment Type
  – XSEDE connects to XSEDE Test Grid.
  – XCG connects to the Cross Campus Grid.
Installation Progress...

- Container and client files are installed.
The Installation Process Questions

- Port for container (default is 18443)
- Host Name is the public DNS name of Container host.
The Installation Process Questions

• Owner information?
  - *Username* refers to just user id portion of the grid user resource (i.e. jfk3w)

  ![Screen shot of Owner Information dialog box]

  - New grid container can only be created by someone who has access to a grid user account within target grid system
The Installation Process Questions

• Use grid service for key generation?

- Checking this will rely on grid services for signing the container.
- Leaving unchecked will allow you to provide your own certificate.
The Installation Process Questions

• Using the Grid Keypair Generating Service

- Path to Service should be the RNS path of the Genesis II certificate signing service you would like to use.
- The Grid User Name must be an existing grid user with permission to use the service.
- Choose a keystore password and alias for your local container.
The Installation Process Questions

- Providing own certificate to sign container
The Installation Process Questions

- Start container running now?

Can also start container later with command:

- XCGContainer start
The Installation Process Questions

- Finished!
Post-Install: Adding Container to the Grid

- Your container has now been installed, but is not yet linked in to a namespace.
- grid ln --service-url={service path} {Destination RNS Path}
- For service path, a file called service-url.txt is generated in the install directory during installation.
Post-Install: A Note on Persistence

• Container runs as a service
• If host or container fails, container should be automatically restarted
  – In Linux, the “./XCGContainer start” command must be in .initrc
  – In Windows, the installer attempts to set this up. You must be an admin user for this to work.
Note on access control

• There are a set of scripts for setting up different access control policies in /bin/access-control that may be tailored to your specific needs.
Post-Install Container Testing

• Check that container is linked into target grid system name space
  – Located at /uninitialized-containers under IP or hostname

• We have a script you can use to test - /bin/testing

• Make sure container services are working
  – List services directory
    • Located at /uninitialized-containers/<container>/Services
  – Create directory and file on new container
    • mkdir –rns-service=<path-to-container>/Services/EnhancedRNSPortType <new-dir>
    • cp local:<path-to-local-file> <new-dir>
Post-Install: Resulting Directory Structure

**Genesis II Installation Directory**

**Location specified during install**
Post-Install: State Directory

**Container State Directory**

- container-id.dat
- derby-db
- rbyteio-data
- user-config.xml
- sbyteio-forks
- user-context.xml

- Derby database files
- Contents of Random Byte IO Files
- Streamable Byte IO Data

**Default location is at ~/.genesisII-2.0**