Configuration Management

I. Problem Statement

Main Point: Configuration in the OSG Software Stack is a mess, but we have a plan to fix it.

The problems:

1. It's too hard to configure a single OSG installation (a single computer)
    - We have multiple methods of configuration, all must be used.
    - Our best method is confusing (too much information, unclear defaults, easy to corrupt, hard to update…) and only works for a CE installation.
2. It's too hard to configure a single OSG site
    - Site admins maintain consistency between computers
    - There is duplication of effort and settings in each installation
3. As we add native packaging, we wish to keep consistency between configuration for Pacman and native packaging. We don't yet know how to do this.

The opportunity

As we add native packaging, we have to do something for configuration, and this is a great chance to think about configuration for both native packages and Pacman packages and improve what we do for the benefit of all site administrators.

II. What we have today

configure-osg: A great idea, with need for improvement.

This program takes a single, human-readable/editable file that fully specifies the configuration for a site. It does whatever is needed to edit the underlying software's configuration.

Most of its work is to translate the configuration options into calls to configuration scripts for individual software components. For example, it will push the local GUMS host into the right configuration files, using the VDT's configure_prima and configure_prima_gt4 scripts.

People have to use configure_osg, these configuration scripts, and hand-edit individual components' configuration files.

III. Sketch of a solution

For single-installation configuration: We improve configure-osg so that:

- It works with all installation types.
- Add metadata to support defaults, better updates, and better usabilit.
- The human-editable configuration file is separate from the configuration repository and is not the definitive configuration. (Just the input of how to change the configuration)

For site-configuration: People can integrate configure-osg with their site configuration mechanisms (cfengine, etc…). All they have to do is place the human-editable configuration file on the computer before they install or configure an installation. This is well-handled by existing site configuration systems.

Reference Examples (Alain's notes, probably not distributed)

(Not exhaustive)

Settings that are duplicated across computers:

- GUMS hosts (Needed on CE and on worker nodes for glexec)
- site details (site name, admin email, location)
- Gratia collector used (whether local or an OSG (production/ITB) one.
- Squid server
- Information about SE and CE storage (OSG_APP_DATA)

Defaults that we've changed over the years:

- The Gratia collector
- The RSV collector
- CEMon Collectors (aka Client)

Configuration settings that are dynamic and not set by users:

- Passwords for MySQL application databases (GUMS, VOMS Admin…)
- MySQL connection information (especially the port)
- Select user (such as 'globus' if it exists, otherwise 'daemon')

Some Statistics:

- We have 42 configuration scripts today
- About 17,000 lines in those scripts, plus another 1700 in a common library.

Places where end-users have to call VDT configuration scripts (not configure-osg):

- Bestman
- Xrootd
- LFC
- VOMS

Places where end-users have to hand-edit software component's configuration files:

- Squid
- RSV (but this is getting better)
- Glexec