gLExec & CE Roadmap

John Hover

Group Leader

Experiment Services (Grid Group)

RACF, BNL

Outline

- gLExec: What, Why, How?
- gLExec: Deployment issues and status
- ATLAS Tier 2/3s: gLExec-related survey
- OSG CE Roadmap
- CREAM
- GRAM5
- ATLAS Tier 2/3s: CE-related survey

gLExec: What?

- Job re-authentication for pilot-based systems.
- Developed in Netherlands (NIKHEF) for EGEE. Initially hooked into LCAS/LCMAPS. Now also allows GUMS as a plugin.
- Based on Apache suexec. Takes an executable input and a credential (grid or VOMS proxy file), re-authenticates, and executes the input as the new user (i.e. switches UID).
- Every worker node is now a gatekeeper.

gLExec: Why?

- In pilot-based systems, the pilots are submitted under a pilot credential, often privileged in some way.
- Without gLExec,
  - Any user payload may read the pilot credential and use it for bad purposes. I.e. any compromised user proxy gets you the production proxy.
  - Activity of end users is “invisible” to site/grid accounting.
- In ATLAS, the pilot credential has many privileges (it is the production proxy), and user payloads can be arbitrary (e.g. with *prun*). So gLExec is rather important.

gLEexec: How?

- Can be run in 4 modes:
  - Full auth. Executable is suid root and switches user.
  - Partial isolation. Executable is suid to a generic account.
  - Logging only. No UID switching, but GUMS call made.
  - No-op. Mainly for compatibility.
- Currently being tested at BNL.
- Panda team is developing pilot functionality to use gLExec.

gLExec: How? (2)

- Already included in VDT as add-on to worker node client.
- Does not require new UNIX accounts-- payloads can map to whatever account they would map to now (if they submitted directly).
- May require additional groups (one for each core on a WN).

GLExec: Deployment Issues

- Justifiably careful about:
  - File locations: Does not trust files on network file systems (NFS, AFS).
  - File permissions: Does not trust group-writable files.
  - Job environment: su'd job does not inherit full environment.
- Therefore: MUST BE INSTALLED LOCALLY.
- Pilot must be carefully implemented.
- Requires global or site-specific info: GUMS host, VO-specific “allowed invokers” list, tracking GUIDs.
- Requires host cert on each WN.

GLExec: Deployment status

- GLExec native packages (RPM, DEB) are a VDT/OSG high priority, because of NFS restrictions.
- Because of configuration issues, gLExec embodies the most difficult job for VDT native packaging: must be installed locally but requires site-global information (GUMS server hostname, allowed invokers).
- First VDT RPMs will probably leave site/global info un-configured. Sites may need to configure out of band.

ATLAS Tier 2/3 Survey *

- Where is your worker node client installed? NFS? Local?
- Do all worker nodes already have a host cert?
- ATLAS Tier 2s: What kind of configuration management, if any?
- If currently network filesystem-based, how do you think you will configure WN-based local software?
- Will Tier 3s run analysis for users from other sites?
- What if VDT provided a site-customizable gLExec RPM-builder?

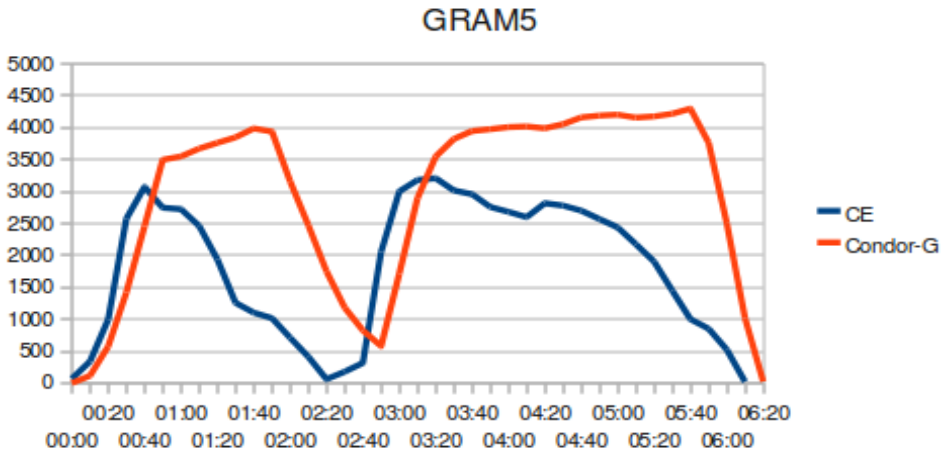
OSG CE Roadmap

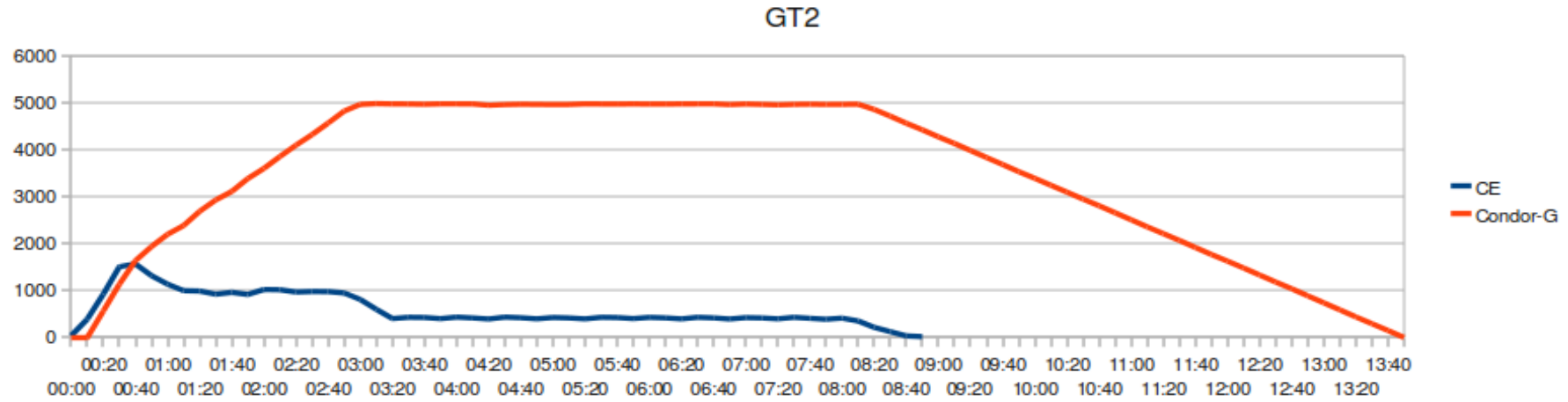
- ATLAS' adoption of a Condor-G, pilot-based workload management system makes CE scalability and function important.
- GT2 has shortcomings in performance, manageability.
- OSG currently developing a roadmap for CE options.
- GRAM 5 and CREAM testing under way (Alain Roy and Igor Sfigoli).
- Other approaches? OSG open to ideas.

GRAM 5

- Enhanced version of Globus GT2 GRAM.
- Evolutionary. Improved performance. (Nearly) backwards compatible at protocol level.
- Enforces one jobmanager per user.
- But jobmanagers have been observed with ~1GB memory footprint.

GRAM 5 vs. GT2 Test



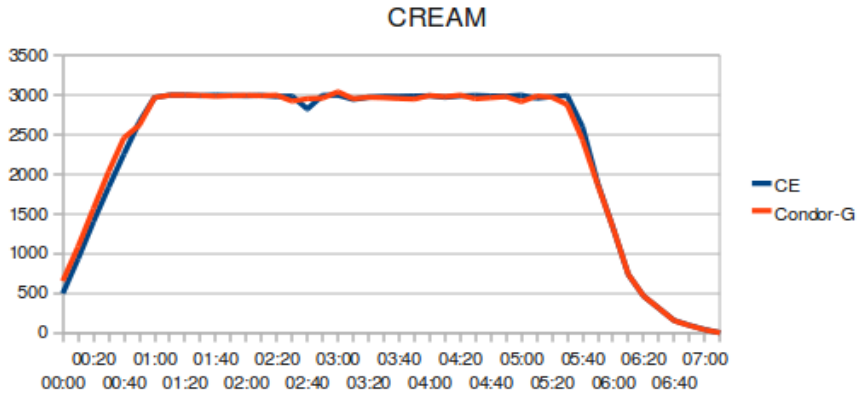


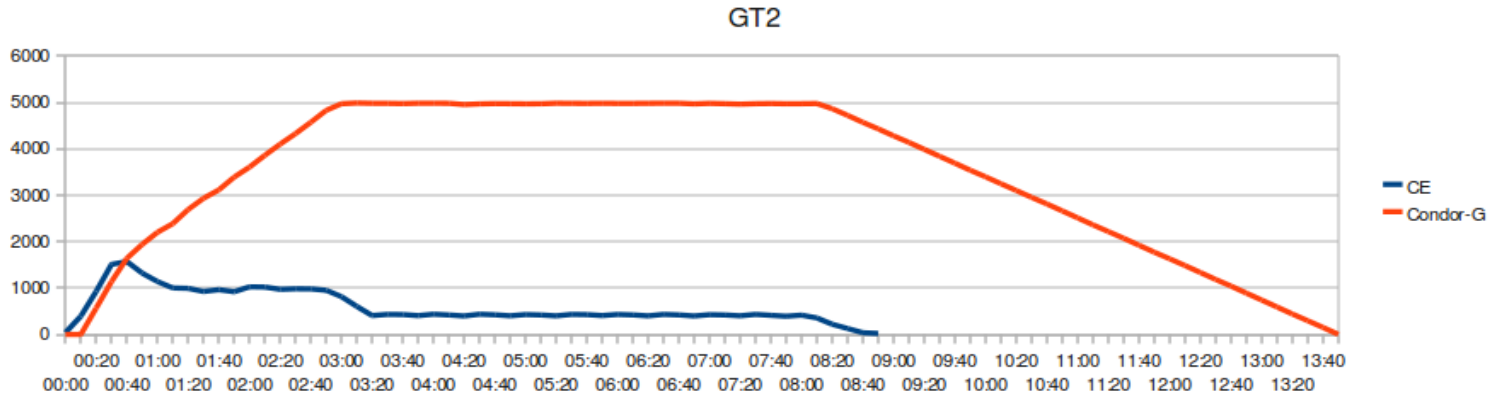
10K jobs @ 30 min each. GRAM 5: 6 hrs vs. GT2: 9 hrs.

CREAM

- Java J2EE Tomcat-based CE. Database back-end. Therefore, in principle “cluster-able” (aside from batch system statefulness).
- Revolutionary rather than evolutionary.
- But, currently requires GridFTP server on the client.
- Significant integration work to be done before inclusion in VDT/OSG. Only EGEE-specific deployment now (gLite RPMs, *yaim*). In (semi- ?) production use in EGEE.
- Still lots of questions for OSG usage.

CREAM vs. GT2 Test





10K jobs @30 min each. CREAM: 7 hrs vs. GT2: 9 hrs

ATLAS Tier 2/3 Survey: CE

- What scaling/performance issues have Tier 2s seen with GT2? Any? Major? Minor? Other CE-specific annoyances/ shortcomings?
- For the future, what would you prefer OSG to focus on for the CE component? I'm interested in clustering, but maybe this isn't really needed.
  - Scalability?
  - Single node performance?
  - Configurability/Flexibility?
  - Simplicity of deployment/configuration?
  - Reliability?

Acknowledgments

- GT2/GRAM5/CREAM Testing graphs by Igor Sfigoli and Alain Roy.