Condor-G – A few lessons learned

by Igor Sfiligoi @ UCSD

How do I use Condor-G

- GlideinWMS
  - Part of the development team
  - Operate the OSG glidein factory
  - Condor-G is used for glidein submission
- OSG Scalability, Reliability and Usability area
  - Testing CE scalability using Condor-G
    - Both production and in-development CE software

Condor-G is old, nothing to learn

- This is what one would expect
- This is why we use Condor-G in glideinWMS
  - Solid reliable technology
  - Use-and-forget
  - etc.
- Turns out, it is not really like that

Condor-G is **not** old

- Just been around for a long time
  - since v6.2 – year 2001
- It is changing all the time
  - New Grid universes added with time
    - v6.8 adds Condor, GT3 and GT4
    - v7.0 adds nordugrid, unicore, pbs and lsf
    - v7.2 adds Amazon EC2
    - v7.4 adds CREAM and GT5
  - Existing protocols being tuned

Some benchmarks first

- As part of the OSG Scalability effort
- Tested
  - GT2 (still by far the most used Grid universe in OSG)
  - GRAM5
  - CREAM
- Using a mix of v7.4.X and v7.5.X

What about the other universes?

- GT4, the WS-based Globus Gatekeeper has never got wide acceptance in OSG
  - Plus Globus has deprecated it (not in Globus 5)
- I plan to test Nordugrid soon
  - Just did not have time yet
- Amazon is interesting but it is quite different
- Unicore, pbs, lsf and Condor-C not interesting for OSG right now

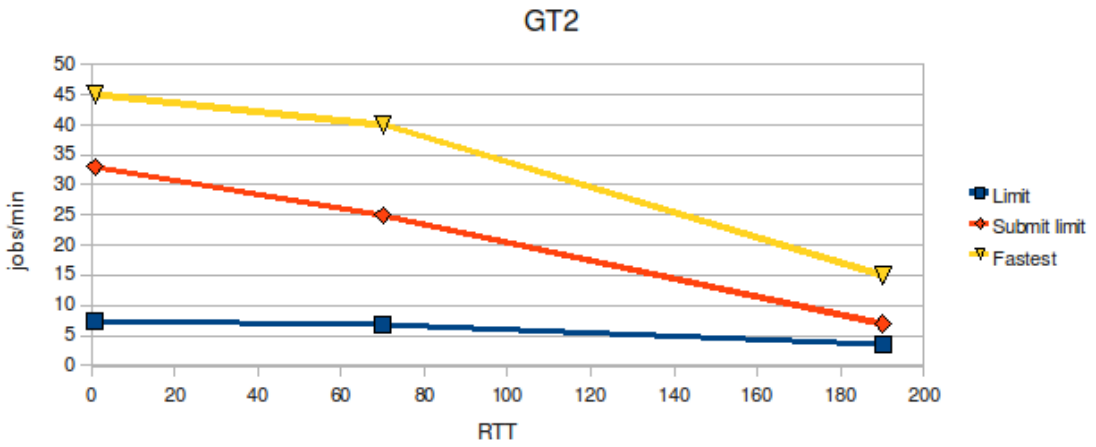
Do I need to mention GT3?

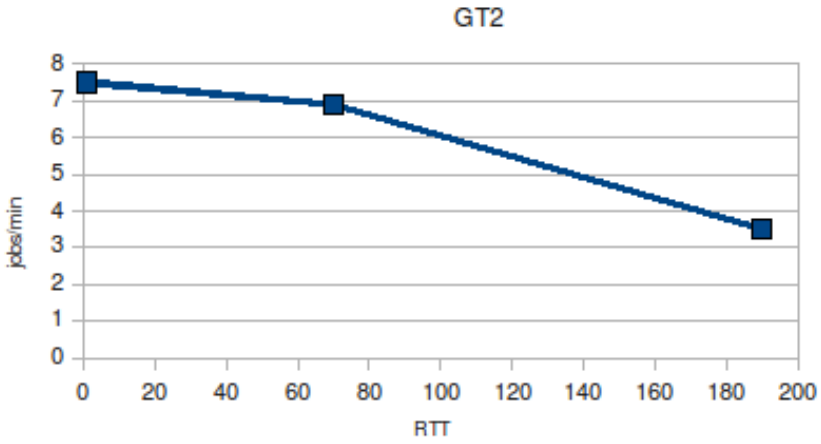
GT2 performance

- How fast can I submit GT2 jobs?
- It really depends what you want to measure
  (with 10 jobmanagers, single user)
  - Job submission alone – 45 jobs/min
  - Job submission while jobs terminating – 33 jobs/min
  - Processing job termination – 15 jobs/min
  - Processing job termination
    while jobs being submitted – 7.5 jobs/min

GT2 performance

- How fast can I submit GT2 jobs?
- It also depends how far away are you from the Grid gatekeeper
  - Submitting across the world will cut rates in half





GT2 performance

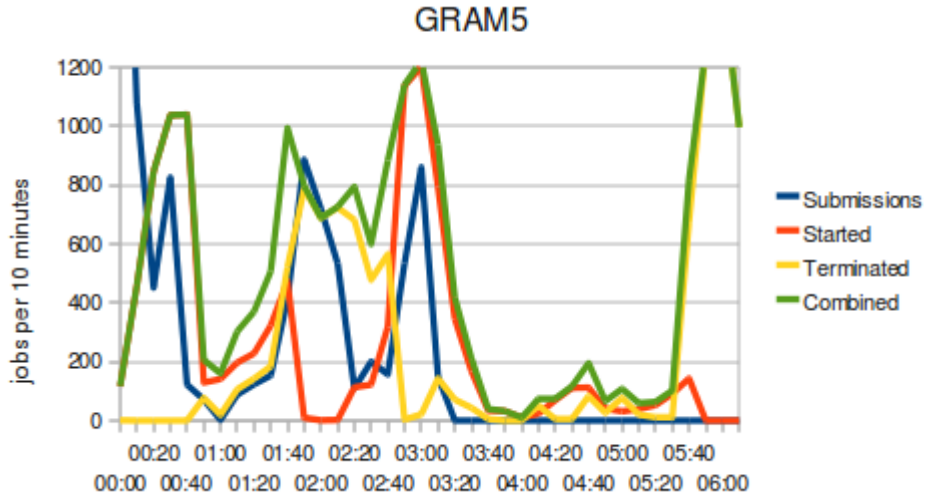
- How fast can I submit GT2 jobs?
- It also depends how many jobs you have in the queue
  - Try submitting 20k jobs at the same time and you may have to wait hours to get the first job started!
  - Use
    dagman -maxidle 1000
    if you have a lot of jobs

GRAM5

- What is GRAM5?
  - The new non-WS Gatekeeper coming with Globus 5
  - Major scalability improvement over GT2
- **Almost** backwards compatible with GT2
  - Removed just streaming support
  - Too bad Condor-G relies on this feature in GT2!
- Condor 7.4 adds explicit GT5 support

GRAM5 performance

- Is it really faster than GT2?
- Yes!
  - Both submit and termination rates exceeded 100 jobs/min
  - But a lot of variation in time
  - Average still over 2x of GT2
- Globus 5.0.0 still has bugs
  - Not ready for production yet, but looks promising!



CREAM

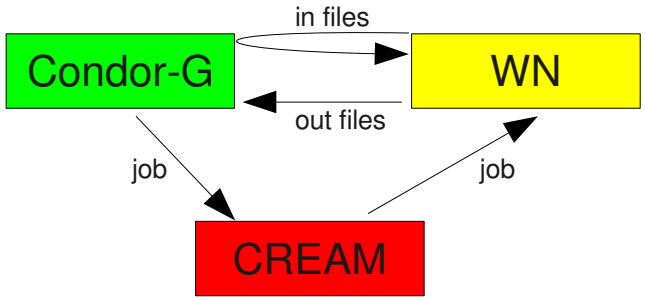
- What is CREAM?
  - A brand new Grid Gatekeeper coming from Italy
  - Web-based, runs inside Tomcat
  - State stored in a relational database
  - Support of multi-home deployments planned
- Uses its own protocol
  - Condor 7.4 adds the CREAM Grid universe

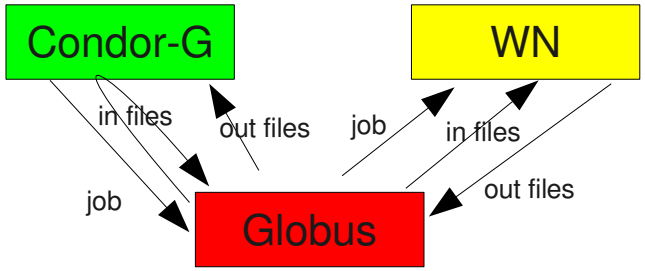
CREAM performance

- How it compares to GT2 and GT5?
- Much faster!
  - Consistent rates in the 50 jobs/minute range
  - About 5x faster than GT2, and 2x faster than GT5
- But the reliability is dismal
  - Out of 10k jobs, over 2k failed to run!
  - We managed to overload the Condor-G client!

How did CREAM overload the client?

- CREAM moves files directly to and from the worker nodes
  - No intermediate staging to the CE like in Globus





- Each worker node will independently contact the GridFTP server running on the Condor-G node
  - No coordination between WNs

GridFTP on the client?

- Wait! I have a GridFTP running on my client?
- That can access any file I own?
  - Can read /etc/passwd? My tax return?
  - Can overwrite ~/.bashrc? ~/.ssh/authorized_keys?
  - My proxy is of course needed,
    but I delegated the proxy to the Grid node!
- I am not sure I want to use CREAM anymore
  - But Globus uses an equivalent GASS Server

Condor-G basically insecure!

- It takes a lot of trust to use Condor-G
  - At least for GT2, GT5 and CREAM
- If any Grid admin wants to compromise your client node, it can
  - And there is no way to prevent that!

You have to log in, right?

- The good news is that vanilla Condor is much more secure
  - Condor team tells me remote admins can only access/modify files in the submit directory

What about portals?

- Portals can submit Grid jobs for multiple users
  - Using user-provided proxies
- I know of portals that use a single system account to host all the user files
  - No user run code on the portal node, so why not?
  - Makes administration easier
- Use of Condor-G gives every user they serve access to **all portal files**

The glideinWMS factory

- The glideinWMS factory used that philosophy
  - Had to find a solution!
- We now have one user account per user served
  - For storing user files and submitting user jobs
  - But no logins!
- Factory processes run as a dedicated user
  - Need a way to switch to the user account when needed
  - But do not want to run as root!

Condor privsep

- Since v7.0, Condor ships with a tool that allows account switching without becoming root
  - Sort of a lightweight **sudo**
  - But more limited in scope, thus
    - easier to configure
    - more secure, if you can live within the limits
- GlideinWMS v2.4 now uses Condor privsep to operate safely with Condor-G

Condor privsep

- Documentation was a problem
  - The switchboard configuration is actually well documented
  - But how to use it.... had to look through the code!
- It is relatively easy to use
  - But not by hand
  - Wrote a python module that made the use simple in glideinWMS

Summary

- Condor-G a wonderful tool to access the Grid
  - Easy to use, flexible, etc.
- But its security model can be a real problem
  - Not really Condor's fault
  - But users only know about Condor, so security implications should be clearly stated
- Condor privsep can help
  - Although this is currently not a supported use-case

Additional resources

Links

- OSG Scalability area results
  - http://hepuser.ucsd.edu/twiki2/bin/view/UCSDTier2
- OSG Scalability test tools
  - http://sourceforge.net/projects/osgscal/
- GlideinWMS home page
  - http://www.uscms.org/SoftwareComputing/Grid/WMS/glideinWMS/

Acknowledgements

- Many thanks to the whole Condor team for the continuous support
  - And in particular to Dan Bradley and Jaime Frey
- This work was partially supported by
  - DoE grant DE-FC02-06ER41436
  - NSF grants PHY-0533280, PHY-0612805 and OCI-0943725