Scaling Glidein WMS to manage more jobs on more heterogeneous resources

Marco Mambelli
HTCondor Week
21 May 2015
Trends and growing needs

• Less structured resources and infrastructure
  – Traditionally OSG had Compute Elements, all resources and users had x509 certificates
  – Campus resources not in the Grid
  – Certificate-less authentication

• Need for more resources
  – Scale to more jobs
  – Access more resources
  – Simplify the management
**Glidein based Workload Management System**

- Factory submits Glideins to resources (entries) as needed
- Frontend helps understanding which resources are needed and triggers the Factory
- Glideins start and become available job slots for the users
  - They also run tests on the resource and prepare a more uniform environment
- Glideins appear to the user as a single pool of resources, User Pool

---

Graphic by Parag Mhashilkar
Supporting new resources

- Direct batch submission using BOSCO (leadership clusters, campus clusters)
- EC2 compliant clouds
  - Amazon
  - OpenStack
- HTCondor-CE
Cloud

• Initial support in 2012
• Glidein WMS team contributed to OpenStack
• Work in collaboration with CMS

• Better provisioning (burst ramp-up)
• Need to support more native APIs (OpenStack, Google CE, Microsoft Azure)
  – Better control
  – Access to more resources (sustain 50K VMs on cloud)

• More information in Tony Tiradani’s talk “CMS Experience Provisioning Cloud Resources with GlideinWMS”
HTCondor-CE

- OSG Compute Elements stating to move to HTCondor-CE
- Gatekeeper
  - HTCondor with some special configuration
  - BLAHP translating to Local Resource Manager
- HTCondor to HTCondor submission
- Support for adding any HTCondor attributes to the submit file that the Factory uses to submit Glideins (memory requirements, number of cores, …)
Direct batch: BLAHP and BOSCO

• Contributed HTCondor software
• Batch Local ASCII Helper Protocol [1] translates HTCondor commands into commands of other Local Resource Managers like PBS, SLURM, (S)GE, LSF
  – Used in HTCondor-CE, BOSCO
  – Worked with INFN (maintainer) and HTCondor team
• BOSCO (Blahp Over Ssh htCondor Overlay) provides a personal HTCondor pool that can submit to multiple heterogeneous resources
  – Installs BLAHP on remote resources and interacts via SSH
  – Integrated with other software, e.g. BOSCO-R, or used directly
  – Contributed by OSG, now partly integrated in HTCondor
  – Worked with HTCondor team
Direct batch submission in GlideinWMS

- Added a new entry type to the Glidein WMS system, batch
- BLAHP and some HTCondor components installed via BOSCO tools
- Using BOSCO ssh tunneling and BLAHP to submit Glideins to the remote Local Resource Manager (PBS, SLURM, (S)GE, LSF, HTCondor) via its submit host
- Authenticated via SSH key pair credentials managed and forwarded by the VO Frontend
- Completely transparent after the initial setup
- Going through firewalls
Glidein WMS architecture diagram: BOSCO submission
Implementing direct batch submission

• Added handling of the credentials:
  – The Frontend stores the ssh keys and forwards them to the Factory for Glidein submission
  – The proxy, used for Glidein authentication is transferred encrypted
• Thank you to the HTCondor team for being responsive and making the authentication more flexible (remote_gahp, bosco_ssh_start)
• Non structured sites:
  – Code alternatives for when the Grid software is not available
• Parameter passing needed to be tuned
• Worked with CMS Opportunistic Workflow effort to run CMS jobs on Gordon (SDSC) and Carver (NERSC)
Improving direct batch submission

• BOSCO is designed for single user
  – Relying on fixed paths in home directory
  – Single key pair
  – Username paired with the host
• Cumbersome to manage different resources with different credentials: remove the passphrase from the key, copy the single key generated by BOSCO or install one manually
• No option to manage HTCondor version installed at the BOSCO resource
  – Utilities pointing to repository where BOSCO tar ball is not released regularly
• Most changes are in BOSCO which is not in active development
Ready to take advantage of new resources

- Average of 40,000 CPU/hours on OSG made available to production jobs by Glideins in the last year
- Scaling to O(100k) sustained jobs

Sustaining 200K Glideins – by Edgar Fajardo

CMS jobs on OSG using Glidein WMS

NON CMS jobs on OSG using Glidein WMS
Conclusions

• Glidein WMS can scale to $O(100k)$ jobs (see Edgar Fajardo and Dave Mason talks)
• Glideins can be submitted beyond classic Grid sites:
  – HTCondor-CE
  – Clouds
  – Direct batch systems
• Transparent for the users submitting jobs
• Still complex to add and setup direct batch resources
• Cloud resource management is rigid (ramp-up/down)
• We need to reach more resources (other Cloud API)