glideinWMS – Dynamic Glideins Across National Infrastructures

Mats Rynge - USC Information Sciences Institute

Gideon Juve - USC Information Sciences Institute

Bruce Berriman - Infrared Processing and Analysis Center, Caltech

Ewa Deelman - USC Information Sciences Institute

Krista Larson - Fermilab

Igor Sfiligoi - University of California San Diego
Motivation

• Show that a researcher can bring in and combine local and national infrastructures to her/his desktop computer
  • Local Condor pool
  • Open Science Grid
  • TeraGrid

• glideinWMS with the Corral frontend
Kepler / Periodograms

- Calculates the significance of different frequencies in time-series data to identify periodic signals.
  - Light curve -> Periodogram -> Event -> Event database
  - Mostly FFT
  - Three different algorithms

BLS periodogram for Kepler -4b, the smallest transiting exoplanet discovered by Kepler to date.
Desktop Machine

- Why desktop machine? Where the data is!
- Desktop is the submit host and central manager
  - GSI authentication
  - 10 Slave collectors

SEC_DEFAULT_AUTHENTICATION = REQUIRED
SEC_READ_AUTHENTICATION = OPTIONAL
SEC_WRITE_AUTHENTICATION = REQUIRED
SEC_CLIENT_AUTHENTICATION = OPTIONAL
SEC_DEFAULT_AUTHENTICATION_METHODS = FS,GSI
SEC_DEFAULT_INTEGRITY = REQUIRED
DENY_WRITE = anonymous@*
DENY_ADMINISTRATOR = anonymous@*
DENY_DAEMON = anonymous@*
DENY_NEGOTIATOR = anonymous@*
DENY_CLIENT = anonymous@*

GSI_DAEMON_NAME=/DC=doegrids/OU=Services/
            CN=host.isi.edu,
            /DC=doegrids/...
glideinWMS

- Local Condor Pool
- OSG
- TeraGrid
- Condor Central Master
  - Schedd
  - Negotiator
  - Collector
  - 10 slave collector
glideinWMS setup

• Corral frontend
  • Simpler than the VO frontend
  • No concept of VOs
  • Single users, personal grid proxy

• Corral monitors the Condor queue, if the demand exceeds available resources, asks the factory for more glideins
# Infrastructure Differences

<table>
<thead>
<tr>
<th>Local/Campus</th>
<th>OSG</th>
<th>TeraGrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small – but easily prioritizable</td>
<td>Opportunistic use</td>
<td>Allocations</td>
</tr>
<tr>
<td>Manually managed grid user mappings</td>
<td>Virtual Organization mapping (many VO users to one local UID)</td>
<td>Automatically mapped (one VO, individual accounts)</td>
</tr>
<tr>
<td>One glidein per core</td>
<td>One glidein per core</td>
<td>One glidein for many cores (chunking)</td>
</tr>
</tbody>
</table>

The glideins are submitting as Condor-G jobs (Globus GRAM)
Run in numbers

• Inputs
  • 210664 input light curves
  • 61 GB

• Jobs mapped into 11 dags,
  • Total jobs: 8264
  • Job restarts: 1384

• Outputs
  • 790 GB

We guessed the run would take 24 hours – it took approximate 10 hours!
Workflow Details

• Pegasus Workflow Manager
  • 11 dags, ~ 50000 tasks each
  • Wall time based job clustering
    Target: 1 hour
  • ~ 800 jobs per dag

• Wrapper scripts wrapping wrapper scripts, wrapping wrappers...
  • Glideins can only abstract to a certain level
Conclusion

• Running across national cyber infrastructures is getting easier!

• Data is a limiting factor for these kind of runs