WMS
status and plans

Marco Cecchi – INFN CNAF
On behalf of the gLite job management Product Team

CCR Workshop, Isola d'Elba, May 19th 2011
- Introduction
- Utilization in production
- A few words on EMI
  - preview of compute area services work plan in EMI
- Roadmap for WMS
- Conclusions
• **Workload Management System**
  – Grid meta-scheduler, push-model
    • ‘canonical’ Grid paradigm
      – interoperates with security, information, data
  – Complex design
    • Timed/prioritized request queues
    • General purpose thread pool
    • Lock-free mechanism for match-making
    • Rich JDL, pluggable
    • Closed-loop mechanisms for safe job replanning
      – L&B Statistics on queue performance
    • DoS prevention based on both system and application parameters
    • Handle complex jobs, with dependencies
    • Fully-fledged monitoring tool (web interface, statistics)
• **WMS release history**

• **WMS/gLite 3.1**
  – SL4 ia32
    • more then 2 years in production
    • certified/released in EGEE-III
  – Very stable service
    • Improved over time in EGEE-III
    • Little need for baby-sitting
      – ‘manual’ maintenance basically only for disk handling

• **WMS/gLite 3.2**
  – SL5 x86_64
    • Released in the EMI-1 distribution (Kebnekaise)
    • To be tested in production yet
      – Requires some hardening (LB PR/bkserver)
### WMS/glite 3.1

- CNAF+BA: 24 instances in load balancing
  - 2010 Jan 1<sup>st</sup> – 2010 Dec 31<sup>th</sup>

<table>
<thead>
<tr>
<th>VO</th>
<th>SUBMITTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>cms</td>
<td>28543161</td>
</tr>
<tr>
<td>atlas</td>
<td>6111813</td>
</tr>
<tr>
<td>ops</td>
<td>1141799</td>
</tr>
<tr>
<td>lhcb</td>
<td>1160437</td>
</tr>
<tr>
<td>biomed</td>
<td>639114</td>
</tr>
<tr>
<td>theophys</td>
<td>700223</td>
</tr>
<tr>
<td>argo</td>
<td>454453</td>
</tr>
<tr>
<td>compchem</td>
<td>369007</td>
</tr>
<tr>
<td>superbvo.org</td>
<td>155887</td>
</tr>
<tr>
<td>glast.org</td>
<td>83178</td>
</tr>
<tr>
<td>pamela</td>
<td>95239</td>
</tr>
<tr>
<td>gridit</td>
<td>132960</td>
</tr>
<tr>
<td>bio</td>
<td>51737</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VO</th>
<th>SUBMITTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>dteam</td>
<td>106390</td>
</tr>
<tr>
<td>infnggrid</td>
<td>4276</td>
</tr>
<tr>
<td>virgo</td>
<td>8061</td>
</tr>
<tr>
<td>enmr.eu</td>
<td>5832</td>
</tr>
<tr>
<td>alice</td>
<td>150519</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>CNAF</td>
<td>~40M</td>
</tr>
</tbody>
</table>
**WMS/glite 3.1**

- WMS node-types deployed at CERN and CNAF, mostly
  - Other italian nodes: PD, CT, BA

- CERN: 25 nodes
  - [http://wmsmon.cern.ch/monitoring/monitoring.html](http://wmsmon.cern.ch/monitoring/monitoring.html)

<table>
<thead>
<tr>
<th>VO</th>
<th>avg. jobs/day per instance as of May ‘11</th>
<th>avg. jobs/day</th>
<th>avg. jobs/year</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAM</td>
<td>11k+4k+11k+10k+3k+3k+3k</td>
<td>48k</td>
<td>17.5M</td>
</tr>
<tr>
<td>CMS</td>
<td>15k+15k+15k+5k</td>
<td>50k</td>
<td>18.25M</td>
</tr>
<tr>
<td>Alice</td>
<td>5k+5k+5k</td>
<td>15k</td>
<td>5.5M</td>
</tr>
<tr>
<td>Atlas</td>
<td>2k+1.5k+1.5k</td>
<td>5k</td>
<td>1.8M</td>
</tr>
<tr>
<td>LHCb</td>
<td>2.5k+2.5k</td>
<td>5k</td>
<td>1.8M</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>~45M</td>
</tr>
</tbody>
</table>
**WMS/glite 3.1**

- Utilization in production over Y2010

<table>
<thead>
<tr>
<th>Y2010</th>
<th>submitted jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>CERN</td>
<td>45Mjobs</td>
</tr>
<tr>
<td>CNAF</td>
<td>40Mjobs</td>
</tr>
<tr>
<td>TOTAL</td>
<td>85Mjobs</td>
</tr>
<tr>
<td>HEP</td>
<td>83Mjobs</td>
</tr>
</tbody>
</table>
WMS/glite 3.1

monthly production sampled every sixth month

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>alice</td>
<td>143718</td>
<td>49106</td>
<td>133938</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>argo</td>
<td>0</td>
<td>22009</td>
<td>20</td>
<td>1517</td>
<td>3904</td>
<td>51026</td>
</tr>
<tr>
<td>atlas</td>
<td>225376</td>
<td>214731</td>
<td>213023</td>
<td>191775</td>
<td>586148</td>
<td>657552</td>
</tr>
<tr>
<td>babar</td>
<td>0</td>
<td>1797</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>bio</td>
<td>0</td>
<td>4496</td>
<td>24947</td>
<td>5624</td>
<td>0</td>
<td>8908</td>
</tr>
<tr>
<td>biomed</td>
<td>0</td>
<td>10335</td>
<td>17057</td>
<td>83071</td>
<td>46780</td>
<td>101155</td>
</tr>
<tr>
<td>cdf</td>
<td>0</td>
<td>9876</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>cms</td>
<td>848158</td>
<td>1610374</td>
<td>2346211</td>
<td>2570718</td>
<td>2084066</td>
<td>2264921</td>
</tr>
<tr>
<td>compchem</td>
<td>0</td>
<td>5675</td>
<td>44403</td>
<td>1903</td>
<td>48778</td>
<td>97116</td>
</tr>
<tr>
<td>comput-er.it</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>36</td>
<td>1</td>
</tr>
<tr>
<td>cyclops</td>
<td>0</td>
<td>46</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>enmr.eu</td>
<td>0</td>
<td>873</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2872</td>
</tr>
<tr>
<td>esr</td>
<td>0</td>
<td>3642</td>
<td>211</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>glast.org</td>
<td>0</td>
<td>40129</td>
<td>15483</td>
<td>10579</td>
<td>285</td>
<td>36365</td>
</tr>
<tr>
<td>gridit</td>
<td>0</td>
<td>691</td>
<td>9543</td>
<td>1163</td>
<td>2012</td>
<td>2015</td>
</tr>
<tr>
<td>infngrid</td>
<td>0</td>
<td>15</td>
<td>17</td>
<td>38</td>
<td>88</td>
<td>4472</td>
</tr>
<tr>
<td>lhcb</td>
<td>41788</td>
<td>71030</td>
<td>114752</td>
<td>48743</td>
<td>105618</td>
<td>66078</td>
</tr>
<tr>
<td>magic</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2588</td>
</tr>
<tr>
<td>ops</td>
<td>0</td>
<td>21755</td>
<td>23187</td>
<td>90709</td>
<td>141547</td>
<td>137517</td>
</tr>
<tr>
<td>pamelal</td>
<td>0</td>
<td>0</td>
<td>273</td>
<td>6287</td>
<td>1229</td>
<td>3687</td>
</tr>
<tr>
<td>superbvo.org</td>
<td>0</td>
<td>0</td>
<td>302</td>
<td>579</td>
<td>469</td>
<td>184</td>
</tr>
<tr>
<td>thephys</td>
<td>0</td>
<td>34370</td>
<td>62306</td>
<td>41343</td>
<td>19827</td>
<td>58140</td>
</tr>
<tr>
<td>virgo</td>
<td>0</td>
<td>708</td>
<td>754</td>
<td>0</td>
<td>0</td>
<td>3528</td>
</tr>
</tbody>
</table>
A few words on EMI

- The European Middleware Initiative is a close collaboration of the three major middleware providers, ARC, gLite and UNICORE
  
  • Develop middleware that strengthens European presence by **consolidating and enhancing** the existing distributed computing infrastructures
    - Gluing together european MW stacks
    - Reducing more than increasing code
  
  • Simplify and organize the different middleware services implementations by delivering a **streamlined, coherent, tested and standard compliant distribution**
  
  • Focus on **usability, compatibility, manageability, interoperability, sustainability**
EMI WMS 3.3 highlights

- Job feedback
  - Replanning jobs stuck in blocking queues
  - L&B statistics on queue performance
    - Avg and std dev over a given period of time
- Sandbox transfer tracking
  - sandbox transfer is a major cause of faults (proxy mixup, wrong FQAN/uid-gid mapping, network outages etc.)
    - LB 2.1 introduces the ability to log sandbox transfer progress, as a separate specific job type linked to the user job
- More performant WMS+LB co-location deployment
- GridSite 1.5
  - Delegation-2
  - RFC-style VOMS proxies
- Fixed bugs/ enhancements
  - When a collection is aborted the "Abort" event is now logged for all the sub-nodes as well (wmproxy side)
  - Retry policies for ISB download and OSB upload are now separated.
  - All attributes of a SA/SE can now be used in gangmatching
Preview of compute area components work plan in EMI

- Portability
  - full distribution
    - SL6 x86_64
    - Debian 6 x86_64
  - clients
    - SL5/32
    - latest Ubuntu
- GLUE2-aware MatchMaker, ISM, JDL
- Improve interactive access? (in WMS it is called perusal)
- Remove GSI – implement EMI delegation
- Do cloudy things :)
- Support all the EMI-blessed Batch Systems
- Provide a common framework for MPI
Preview of compute area components work plan in EMI

- Use Argus for AuthZ throughout
- Use common AuthN libraries
- Provide DoS protection mechanisms (done for WMS)
- Document performance/stability
  - Easy for the WMS thanks to WMSMon
WMS roadmap

- Access control with Argus
- DAGs without Condor DagMan
  - DAG processing engine in the WM
    - Job status taken by L&B
    - Reduced a number of helpers/components
  - Support for DAGs in CREAM!
    - Support workflows
    - WM Memory footprint problems
      - Somehow related to ISM restructuring for GLUE2.0
    - Test feedback
Conclusions

- WMS has an expanding use base
  - Especially non-HEP VOs
  - Aim at achieving a strategic position in EGI/IGI
- Implement workflows
  - CompChem has interesting use-cases
- Use EMI at our best for...
  - Make the distribution standard
  - Easily build from source
  - Try to do something interesting with HPC and clouds
Thank you!

EMI is partially funded by the European Commission under Grant Agreement RI-261611
Submission feedback: a simple example with the JDL

```
[ EnableWmsFeedback = true; // job will be replanned when (believed to be) stuck...
  ReplanTimeout = 3600; // ...for more than one hour, explicitly. this grace period can
  // also be automatically calculated by the WMS, according to its
  // statistics, feature enabled if the attribute is missing

  JobType="normal";
  Executable = "/bin/ls";
  InputSandbox = {};
  OutputSandbox = {"out.log", "err.log"};
  // let's create some aliases
  ERT = other.GLUECEEstimatedResponseTime;
  LastTwoHours = 7200;
  rt_compute = [mean = 0; weighted_mean = 1; std_dev = 2;];
  ART = MeasuredResponseTime(rt_compute.mean, other.GLUECEUniqueId, LastTwoHours);
  RT_StdDev = MeasuredResponseTime(rt_compute.std_dev, other.GLUECEUniqueId, LastTwoHours);

  WorkloadRequirements =
    ART == -1 // unknown resource/not significant sample, keep service discovery active
    ||
    (ERT >= ART - 2 * RT_StdDev) && (ERT <= ART + 2 * RT_StdDev);
  // if the RT estimate is exceedingly optimistic or pessimistic,
  // the site must be excluded, being not dependable
  // in this case, we assess the reliability of ERT with probability ~0.95
  // even if it is the condition ERT >= ART - 2 * RT_StdDev which is more important to us

UserRequirements = true; // requirements in terms of CPU, memory, network, software, etc.
Requirements = UserRequirements && WorkloadRequirements;
ERT_ART_tradeoff=.5; // gives the same importance to ART and ERT
Rank = -(ERT_ART_tradeoff * ERT + (1 - ERT_ART_tradeoff) * (ART > 0 ? ART : ERT));
]

WorkloadRequirements can also be expressed server-side, it will appended in && to
UserRequirements at MM time. This will simplify the user's JDL.
```
This design has some pros...

• By design scalability and no ‘single point of failure’
  – performance scales up with the number of instances with no fragmentation on the Grid resources
  – mechanisms to prevent overall congestion
• Stable service, requires little maintenance
• No ‘pressure’ on the Grid
  – only ‘real’ jobs are sent to sites
  – no waste of CPU cycles
• Information system is devoted to gathering throughout
  – no need to retrieve information in other ways
• Operations performed on behalf on the user
  – avoid security implications with identity switching
  – accounting is easier
• 'Static' matchmaking is still vital even with late-binding paradigms
  – Among the other things, it is primary that a user sends payloads only where they have enough time and processing power
    • this cannot be decided when the job is already running, lest having submitted and waited in queue for nothing
...and some cons as well

- Information system, plays a key role in the described architecture, however:
  - "The map is not the territory"
    - GLUE is an abstraction which sometimes cannot grasp the actual resource layout/distribution
      - cluster/subcluster
      - ...
  - Consumer services have to deal with several latencies
    - update rates
    - information caching at each involved level
  - It cannot be blindly trusted, especially for live parameters
    - freecpus, ERT, etc.
  - ‘On-the-fly’ reprioritization is hard to achieve with this model
    - once the job-queue binding is created, it cannot be changed
      - given the intrinsic/extrinsic weaknesses of the Information System, this might become a problem (otherwise the WMS is not supposed to have too many jobs waiting in queue)
  - Complex system, sometimes difficult to debug
Towards a mixed-paradigm (II)

• This feature will implement a feedback mechanism:
  – 1) to learn about the overall status from the previous jobs’s history and not only from the Information System
  – 2) to be able to migrate stuck jobs

• After a given, dynamic timeout, a resubmission will be triggered if the job is still queued at the LRMS
  • this is done via a mechanism which does not need to wait for the LRMS to actually perform the cancellation (done via job’s token removal)
    – each job instance has a unique token identifier
    – the WMS performs a new MM and atomically renames the token upon each reschedule
  – State transitions statistics will be produced by LB server >=2.1 and made available by the WMS to the user, via JDL extensions (classad plugins) and CLI
    – MeasuredResponseTime() available from the user JDL, to be evaluated by the WMS at each MM. It can return either average or standard deviation
    – env GLITE_WMS_QUERY_SERVER=lbserver.ics.muni.cz:9400
glite-lb-stats-duration-fromto ALL 4 5 # returns scheduled->running average time and std dev grouped by queue, for all users