EUROPEAN MIDDLEWARE INITIATIVE

WMS status and plans

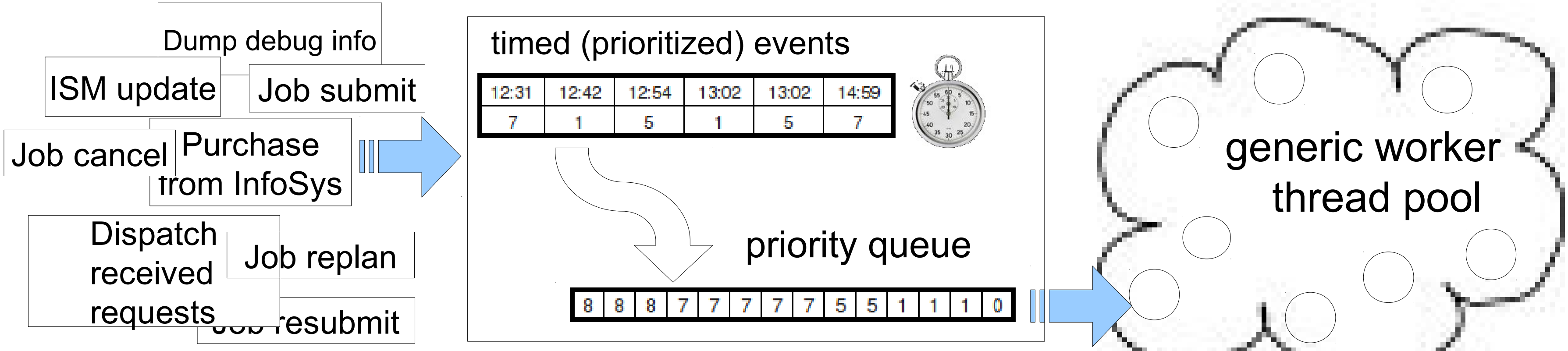
Marco Cecchi – INFN CNAF

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



12:31	12:42	12:54	13:02	13:02	14:59
7	1	5	1	5	7

8	8	8	7	7	7	7	7	5	5	1	1	1	0

- **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
    - https://wmsmonitor.cnaf.infn.it:8443/wmsmon/main/main.php
    - 2010 Jan 1st – 2010 Dec31th

VO	SUBMITTED
cms	28543161
atlas	6111813
ops	1141799
lhcb	1160437
biomed	639114
theophys	700223
argo	454453
compchem	369007
superbvo.org	155887
glast.org	83178
pamela	95239
gridit	132960
bio	51737

VO	SUBMITTED
dteam	106390
infngrid	4276
virgo	8061
enmr.eu	5832
alice	150519
…	
	39917149
CNAF	**~40M**

- **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

VO	avg. jobs/day per instance as of May '11	avg. jobs/day	avg. jobs/year
SAM	11k+4k+11k+10k+3k+3k+3k	48k	17.5M
CMS	15k+15k+15k+5k	50k	18.25M
Alice	5k+5k+5k	15k	5.5M
Atlas	2k+1.5k+1.5k	5k	1.8M
LHCb	2.5k+2.5k	5k	1.8M
			~45M

- **WMS/glite 3.1**
  - Utilization in production over Y2010

Y2010	submitted jobs		
	CERN	45Mjobs	
	CNAF	40Mjobs	
	TOTAL	**85Mjobs**	
	HEP	*83Mjobs*	

WMS/glite 3.1

monthly production sampled every sixth month

VO	Jan 2009	Jun 2009	Jan 2010	Jun 2010	Jan 2011	Apr 2011
alice	**143718**	**49106**	**133938**	**0**	**0**	**2**
argo	0	22009	20	1517	3904	51026
atlas	**225376**	**214731**	**213023**	**191775**	**586148**	**657552**
babar	0	1797	0	0	0	0
bio	0	4496	24947	5624	0	8908
biomed	0	10335	17057	83071	46780	101155
cdf	0	9876	0	0	0	0
cms	**848158**	**1610374**	**2346211**	**2570718**	**2084066**	**2264921**
compchem	0	5675	44403	1903	48778	97116
comput-er.it	0	0	0	0	36	1
cyclops	0	46	0	8	0	0
enmr.eu	0	873	1	0	0	2872
esr	0	3642	211	0	0	0
glast.org	0	40129	15483	10579	285	36365
gridit	0	691	9543	1163	2012	2015
infngrid	0	15	17	38	88	4472
lhcb	**41788**	**71030**	**114752**	**48743**	**105618**	**66078**
magic	0	0	0	0	0	2588
ops	0	21755	23187	90709	141547	137517
pamela	0	0	273	6287	1229	3687
superbvo.org	0	0	302	579	469	184
theophys	0	34370	62306	41343	19827	58140
virgo	0	708	754	0	0	3528

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

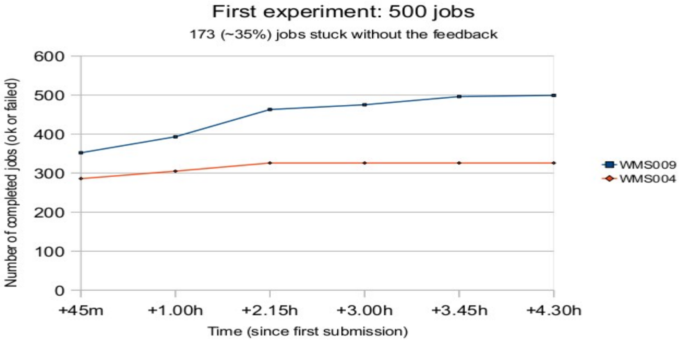
- Portability
  - full distribution
    - SL6 x86_64
    - Debian 6 x86_64
  - clients
    - SL5/32
    - latest Ubuntu
- GLUE2-aware MatchMaker, ISM, JDL
- Improve interactive access? (in WMS it is called perusal)
- Remove GSI – implement EMI delegation
- Do cloudy things :)
- Support all the EMI-blessed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

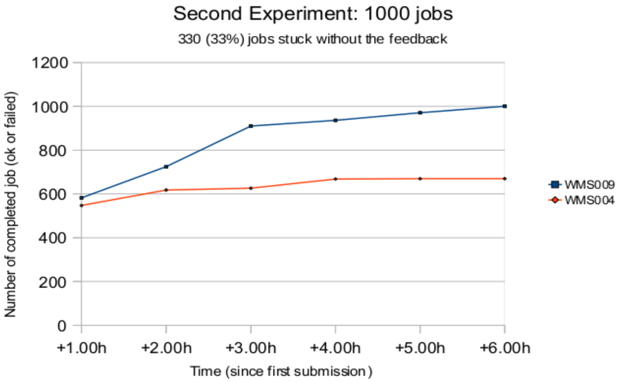
Preview of compute area components work plan in EMI

- 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

EUROPEAN MIDDLEWARE INITIATIVE

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