Experience with the gLite Workload Management System in ATLAS Monte Carlo Production on LCG

Simone Campana, CERN

David Rebatto, INFN

Andrea Sciabà, CERN

CHEP2007 Victoria (Canada)

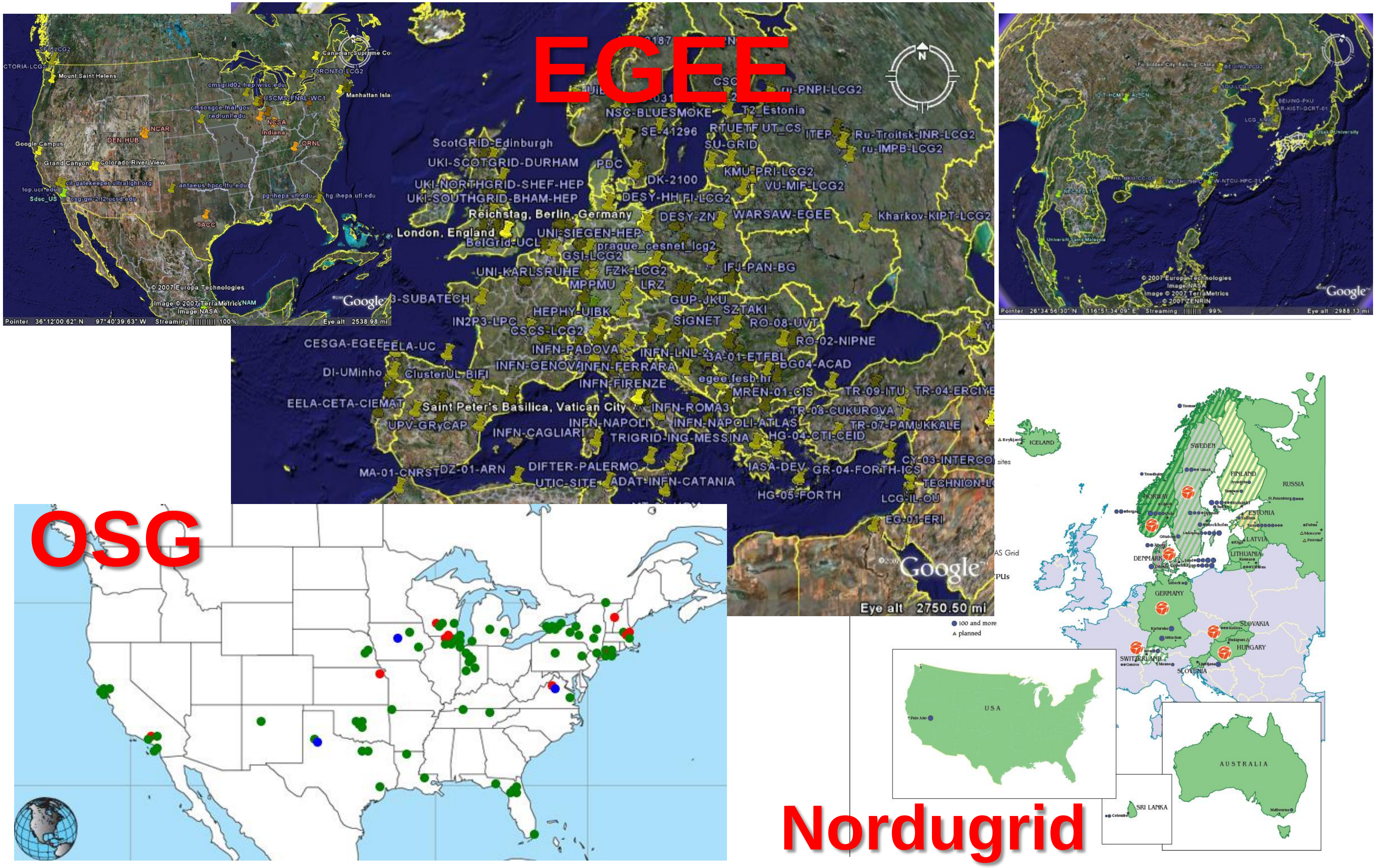
www.eu-egee.org

Information Society and Media

Monte Carlo production

- **LHC experiments need to generate huge amounts of simulated data**
  - Validate the computing and data model
  - Test the complete software suite
  - Develop physics data analysis
- **ATLAS currently benefits of ~10 KSI2K of CPU power and more than one PB of disk space distributed around more than 60 sites**
  - Decentralization and sharing of computing resources
  - Different computing facilities are organized in a hierarchical structure (T0, T1, T2)
    - distinct roles at different levels.
  - In ATLAS Computing Model, MC Production is run at T2s
    - … but currently also at T1s
- **Specific tools have been developed by each experiment to manage the production workflow**
  - For ATLAS this is the Production System (Prodsys)

ATLAS resource and Grids



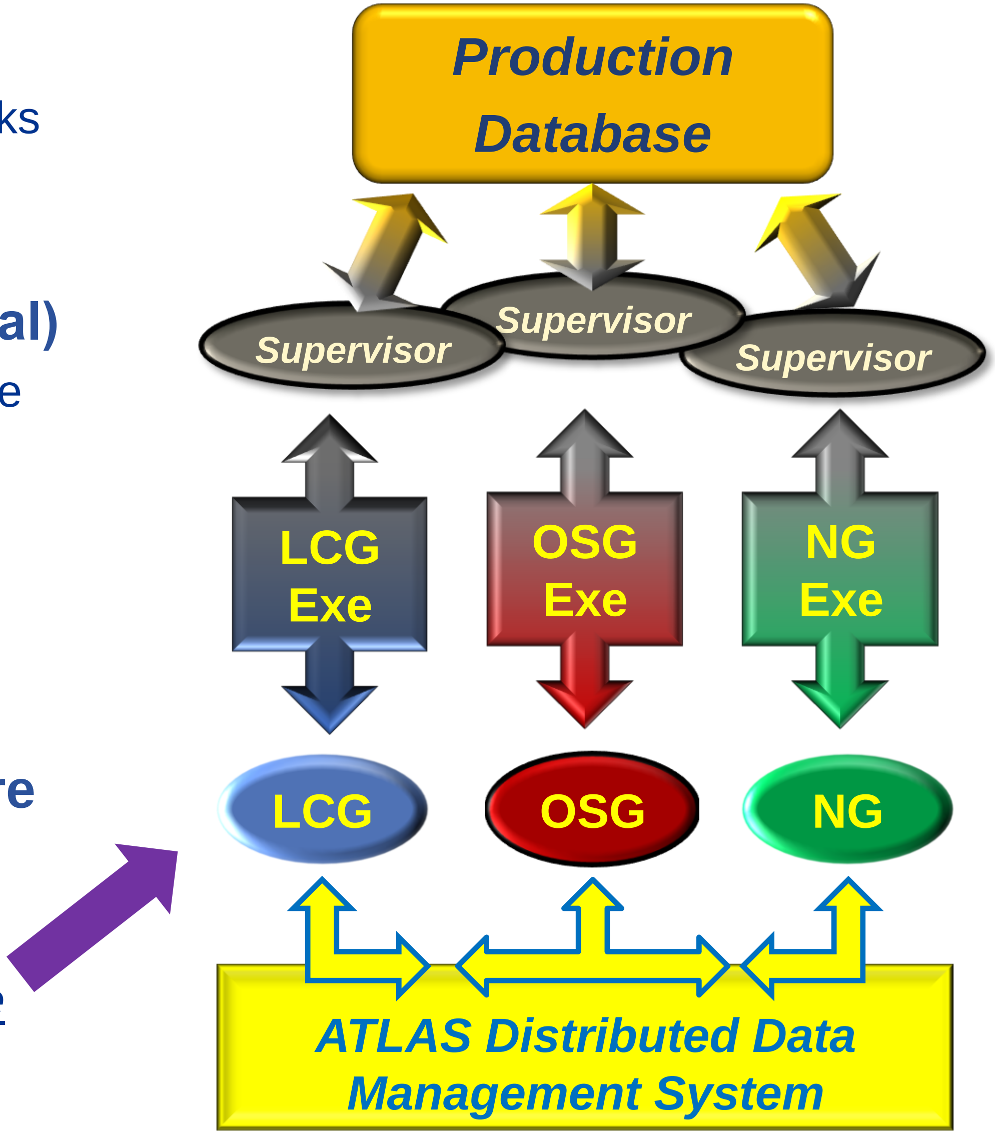
EGEE

OSG

Nordugrid

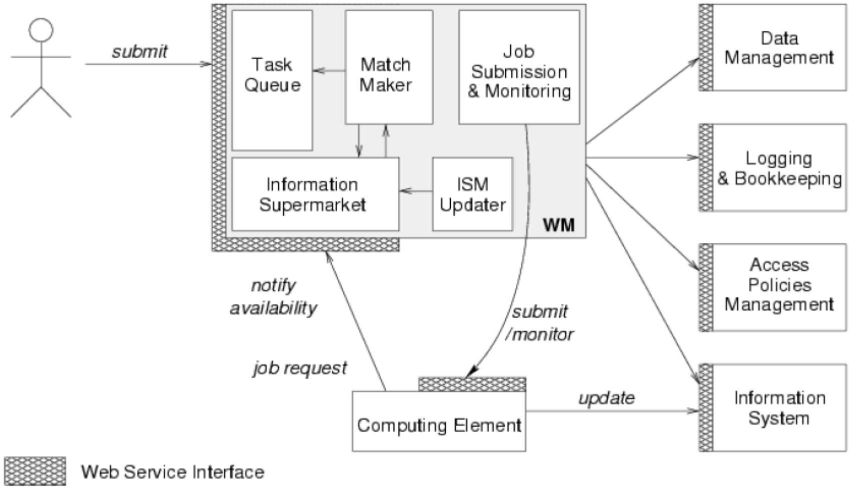
The ATLAS Production System

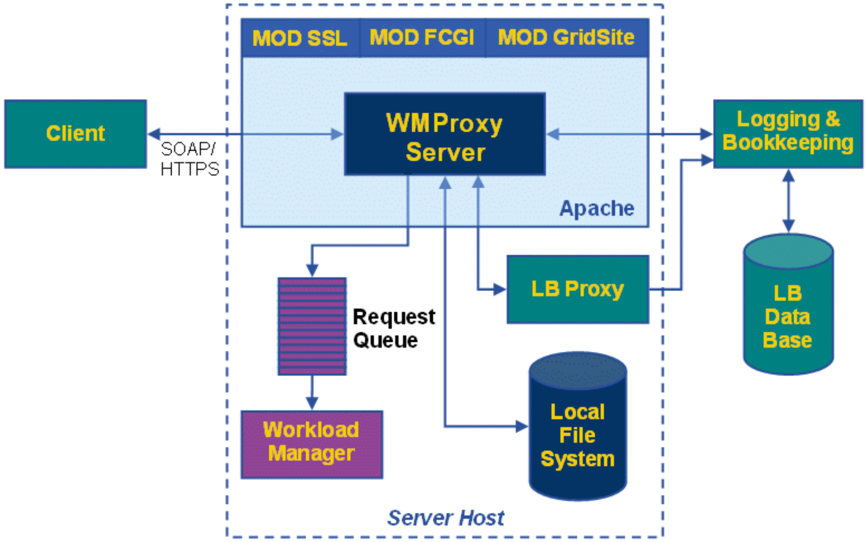
- **A central database**
  - holds grid neutral definitions of tasks and jobs, together with job states
- **A "supervisor"(also Grid neutral)**
  - pulls jobs from the central database
  - submits jobs to the Grid
  - monitors jobs and checks their outcome
- **An "executor" layer acting as interface to the Grid middleware**
  - EGEE/WLCG
    - Lexor using the gLite WMS
      - *Was the LCG-RB before*
    - Condor-G direct submission
    - CRONUS (Condor glide-ins)



The gLite WMS architecture

- **The service to submit and manage jobs**
  - **Task queue**: holds jobs not yet dispatched
  - **Information SuperMarket**: caches all information about Grid resources
  - **Match Maker**: selects the best resource for each job
  - **Job Submission & Monitoring**
  - Interacts with Data Management, Logging & Bookkeeping, etc.



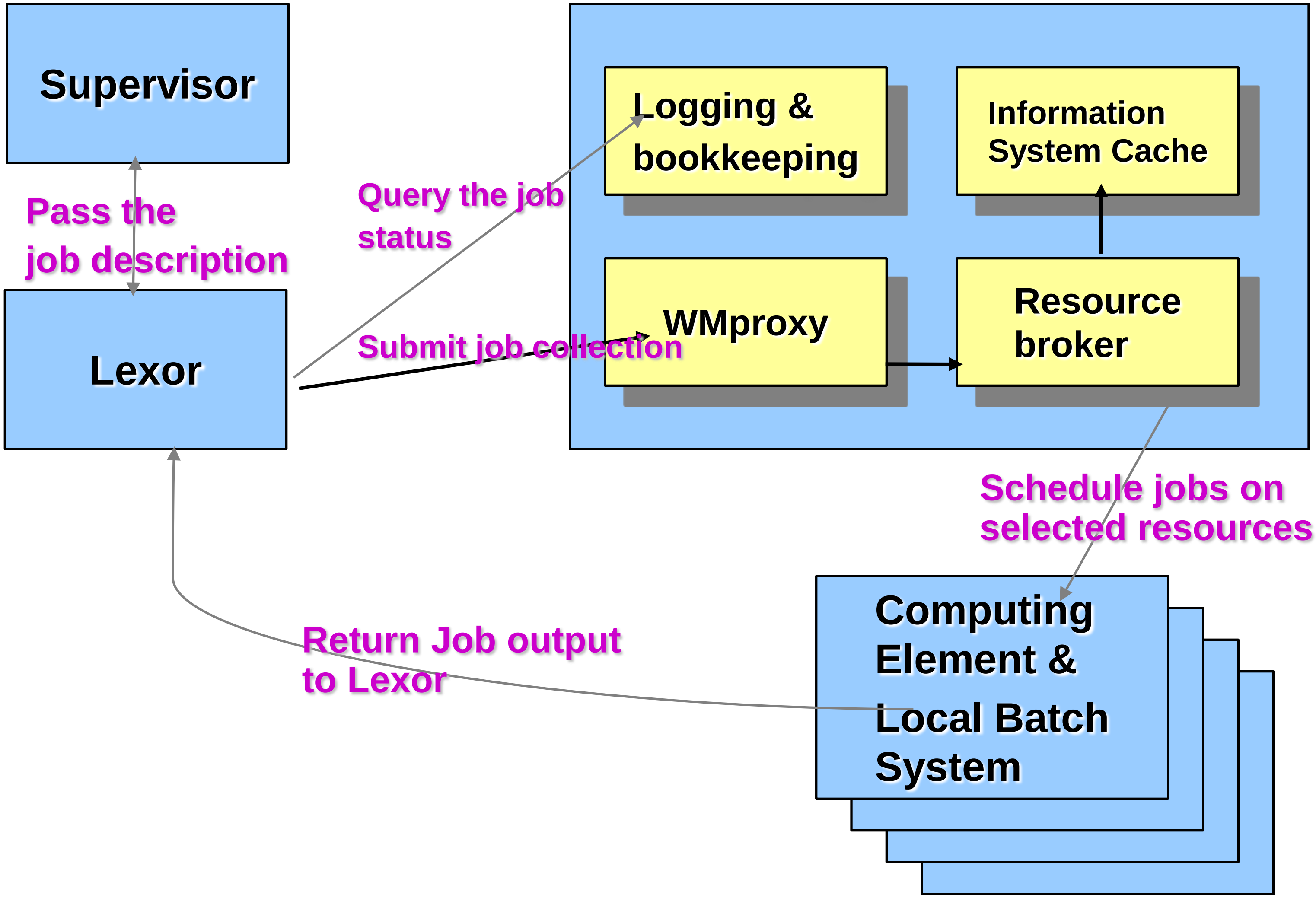


- **WMProxy service optimizes job management and stands between the user and the real WMS**
  - Service Oriented Architecture (SOA) compliant
    - Implemented as a SOAP Web service
  - Validates, converts and prepares jobs and sends them to the WM
  - Interacts with the L&B via LBProxy (a state storage of active jobs)
  - Implements most new features

Main functionality

- **The gLite WMS offers several advantages over the old LCG WMS**
  - Bulk submission
    - Collections: sets of independent jobs
    - New, much more reliable implementation as a compound job submission
  - Job sandboxes
    - Shared input sandboxes for a collection
    - Download/upload of sandboxes via GridFTP, http, https
  - Faster match-making
    - "bulk" matchmaking and ranking for collections
  - Internal task queue
    - If a job cannot match right away it is kept for some time until it matches
  - Resubmission of failed jobs
    - a job is resubmitted right away after a middleware/infrastructure-related failure
    - greatly improves the job success rates
  - A limiter mechanism which prevents submission of new jobs if the load exceeds a certain threshold
    - Leads to "artificial", but desired, limitations of the job submission rate
    - Improves the stability of the system
  - Last but not least, the gLite WMS is actively developed and maintained, while the LCG RB is "frozen"

Lexor


Supervisor
Pass the job description
Lexor
Query the job status
Submit job collection
Logging & bookkeeping
Information System Cache
WMproxy
Resource broker
Schedule jobs on selected resources
Computing Element & Local Batch System
Return Job output to Lexor

gLite WMS issues

- **The gLite WMS has been one source of inefficiency of Lexor in the past**

- **Lexor is using gLite WMS since Summer 2006**
  - gLite WMS not really ready for production
  - A lot of manual intervention was needed
    - Period September-November 2006 has been very critical
- **Main problems**
  - Bulk submission not completely reliable
    - jobs remaining “stale” (in the same state) forever: “zombie” jobs
  - Memory usage growth
    - linear it time under continuous job submission

- **New activity of testing and debugging of gLite WMS started in January 2007**
  - Starting from Experiment’ s requirements and acceptance criteria

Requirements for the gLite WMS

	CMS	ATLAS
Performance		
2007	50K jobs/day	20K production jobs/day + analysis load
2008	200K jobs/day (120K to EGEE, 80K to OSG) Using <10 WMS entry points	100K jobs/day through the WMS; Using <10 WMS entry points
Stability		
		<1 restart of WMS or LB every month under load

WLCG acceptance criteria

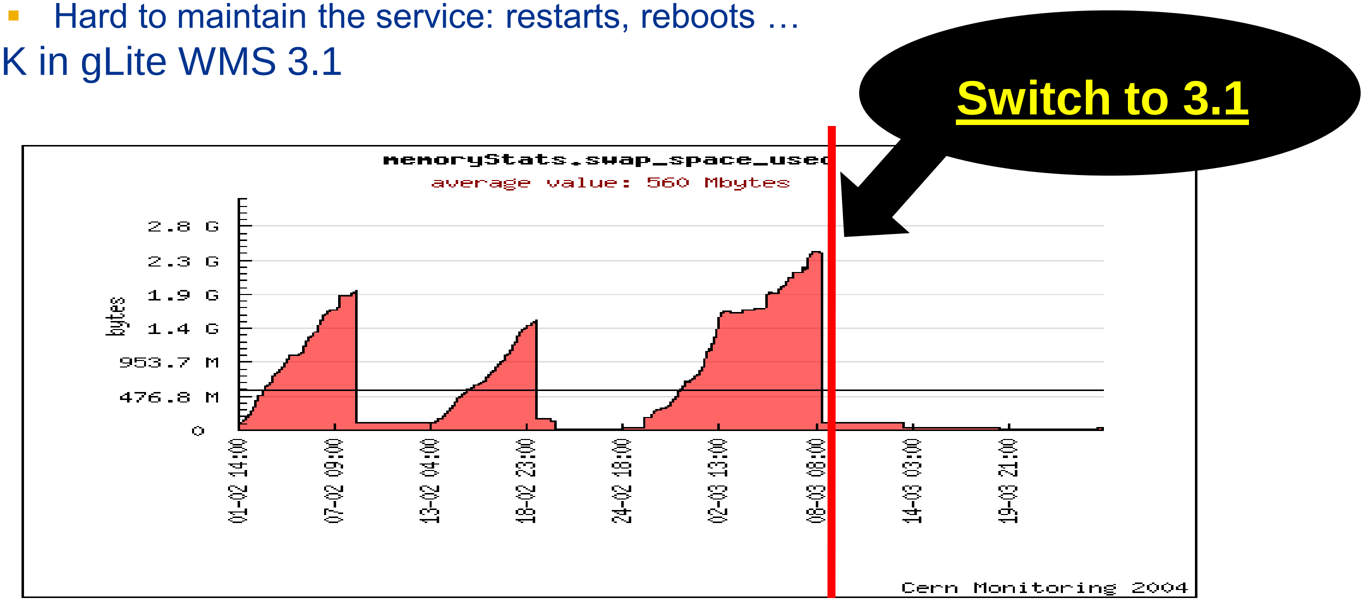
- **Based on the experiment requirements, some criteria have been defined to decide if the gLite WMS satisfies the requirements**
  - At least 10000 jobs/day submitted for at least five days
  - No service restart required for any WMS component
  - The WMS performance should not show any degradation during this period
  - The number of zombie jobs should be less than 0.5% of the total

Testing and improving gLite WMS

- **The testing of the gLite WMS is mainly done by the Experiment Integration and Support team of WLCG**
  - Collaboration between Experiment Integration Support Team, JRA1 (EGEE developers), SA1 (EGEE operations), SA3 (EGEE integration and testing)
  - Bugs discovered, fixed and patched bypassing normal certification procedures
    - WMSes continuously tested, patched and re-deployed
    - Pragmatic approach: very quick turnover
  - Huge improvements in stability and performance

Main improvements: Memory Usage and “Zombie” jobs

- **Memory usage**
  - Grows linearly in the gLite WMS 3.0
    - Hard to maintain the service: restarts, reboots …
  - OK in gLite WMS 3.1



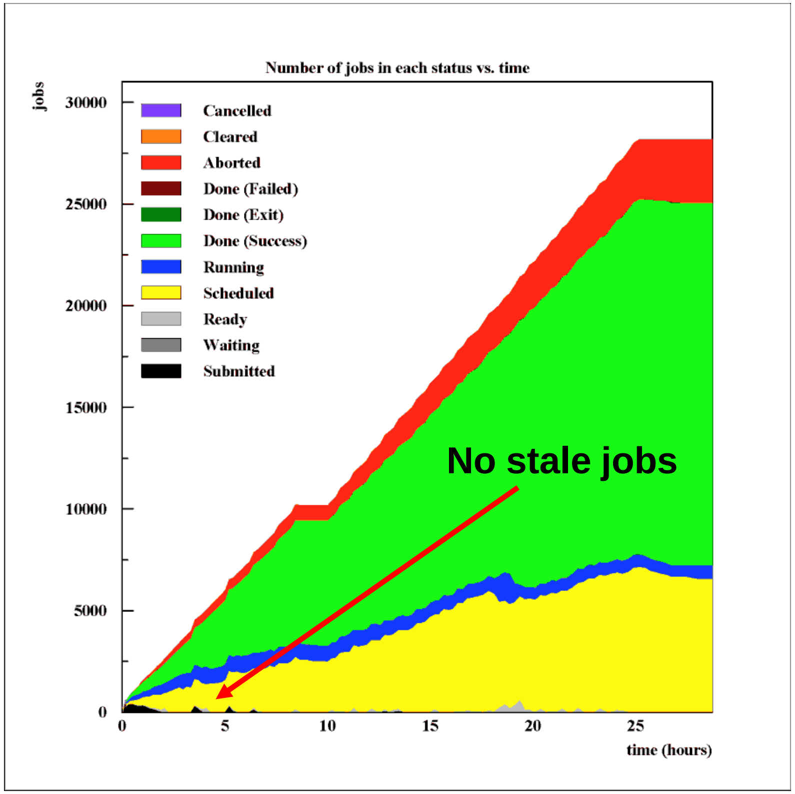
Switch to 3.1

- **The problem of “zombie” jobs have been identified**
  - Collection handling via Condor DAGMAN

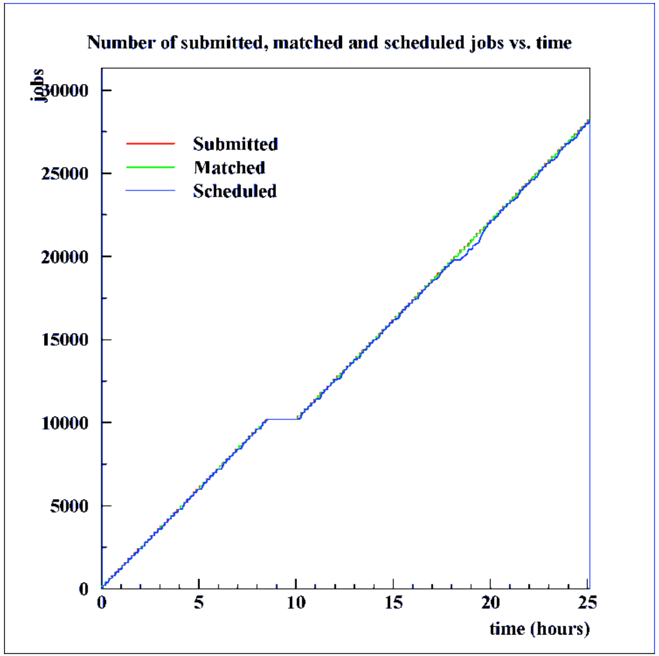
- **DAGMAN has been removed for collection handling**
  - Still there for Acyclic Job Diagrams
  - Collections are handled via a native mechanism
  - The latest test have shown NO jobs stale at all

Results of the acceptance test

- **115000 jobs submitted in 7 days**
  - ~16000 jobs/day well exceeding acceptance criteria
  - The "limiter" prevented submission when load was very high (>12)
- **All jobs were processed normally but for 320**
  - ~0.3% of jobs with problems, well below the required threshold
  - Recoverable using a proper command by the user

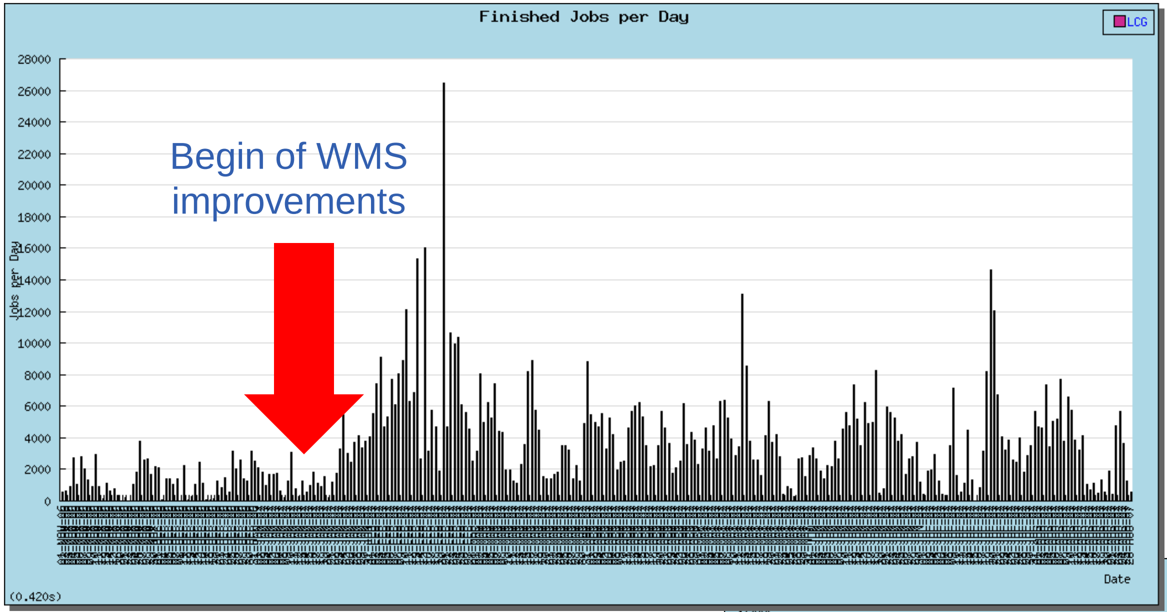


No stale jobs



- **The WMS dispatched jobs to computing elements with no noticeable delay**
- **Acceptance tests were passed**

Lexor Production Activity: Number of jobs

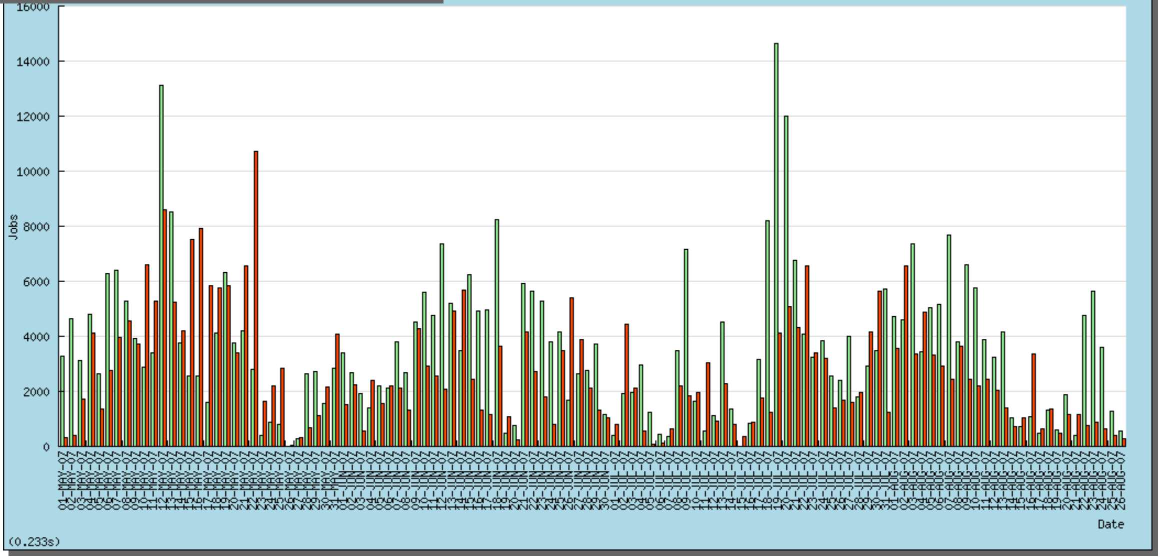


Begin of WMS improvements

Big ramp-up after WMS improvements

Reached 20000 jobs/day

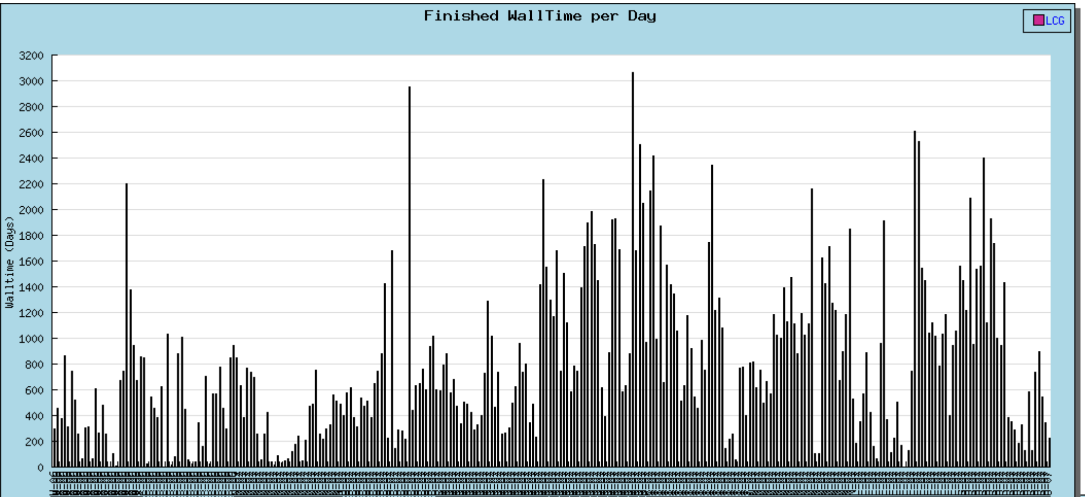
Oscillatory behavior
(also depending on the type of production activity being run)



Job efficiency:
Roughly 60%

See later slides for error categorization and impact

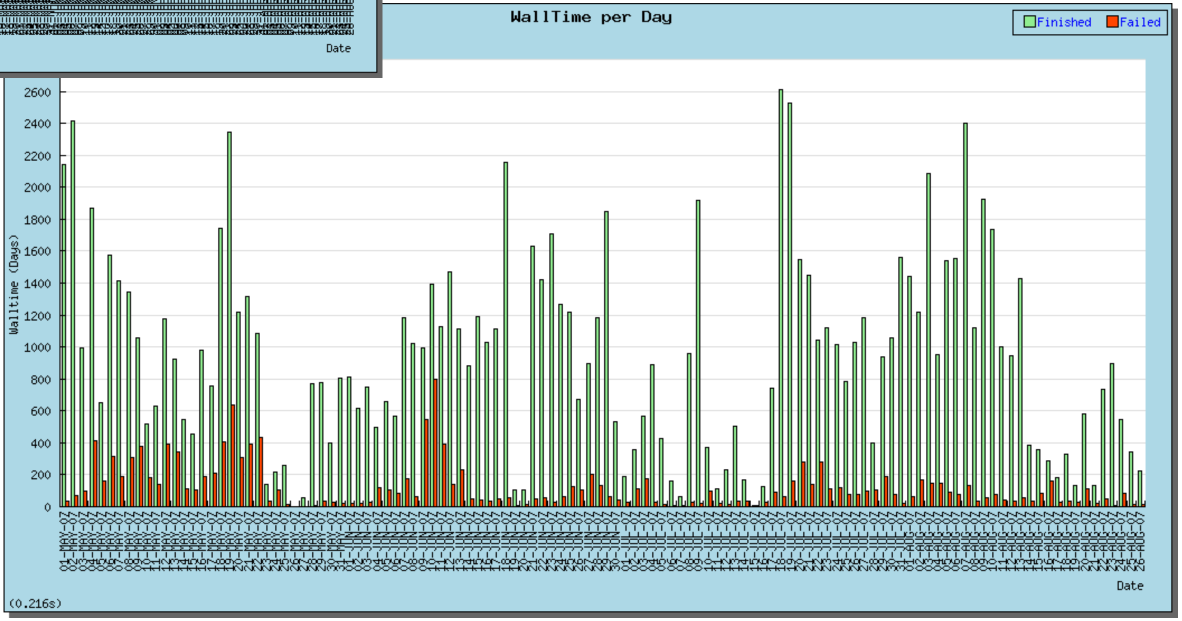
Lexor Production activity: WallClockTime



WCT per day:

Somehow different from Job/Day distribution

Different type of jobs in different periods

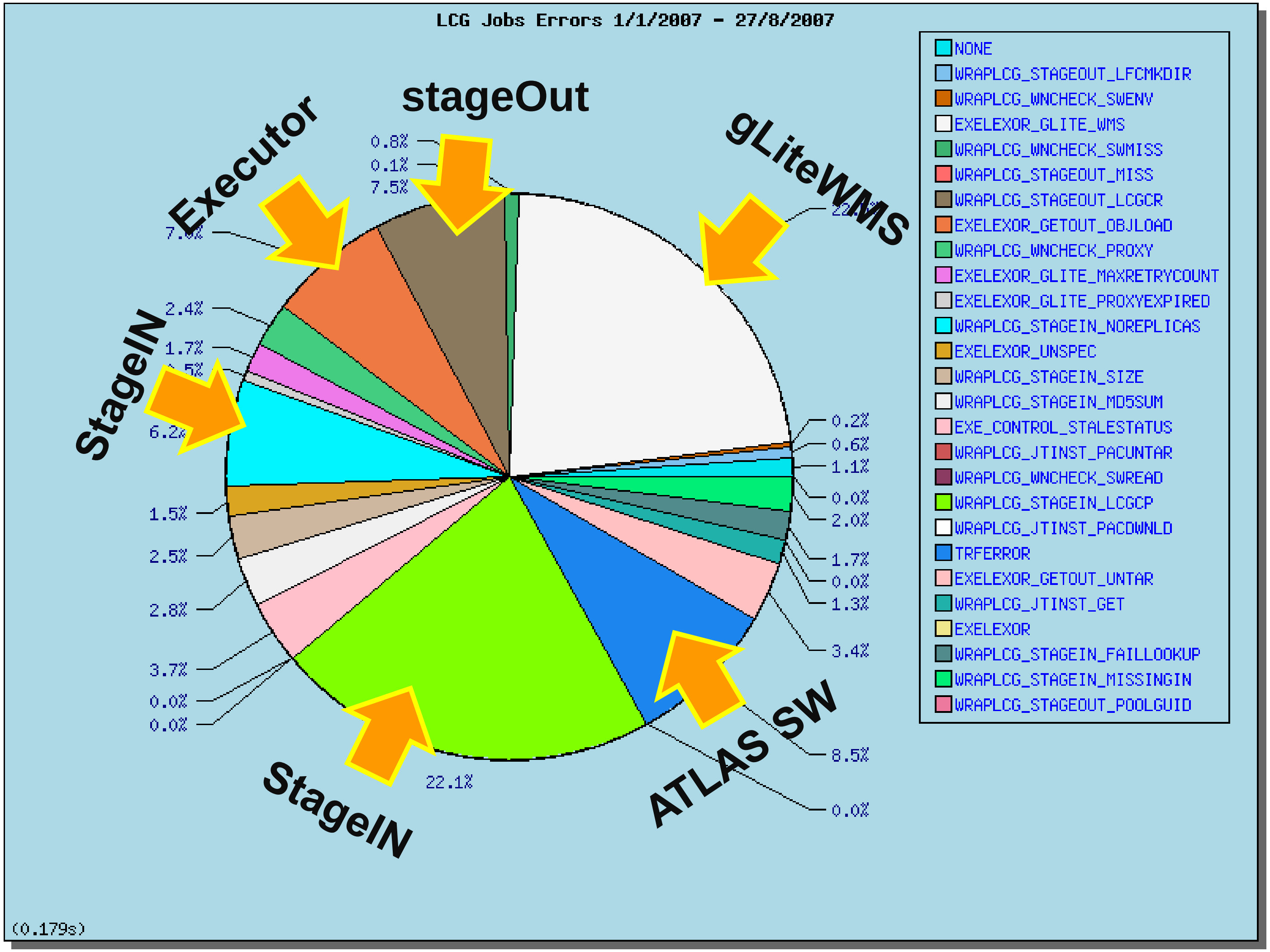


WCT efficiency:

Much higher than job efficiency

We focsed in reducing error with large waste of WCT

Number of Jobs Error Breakdown: January to August 2007

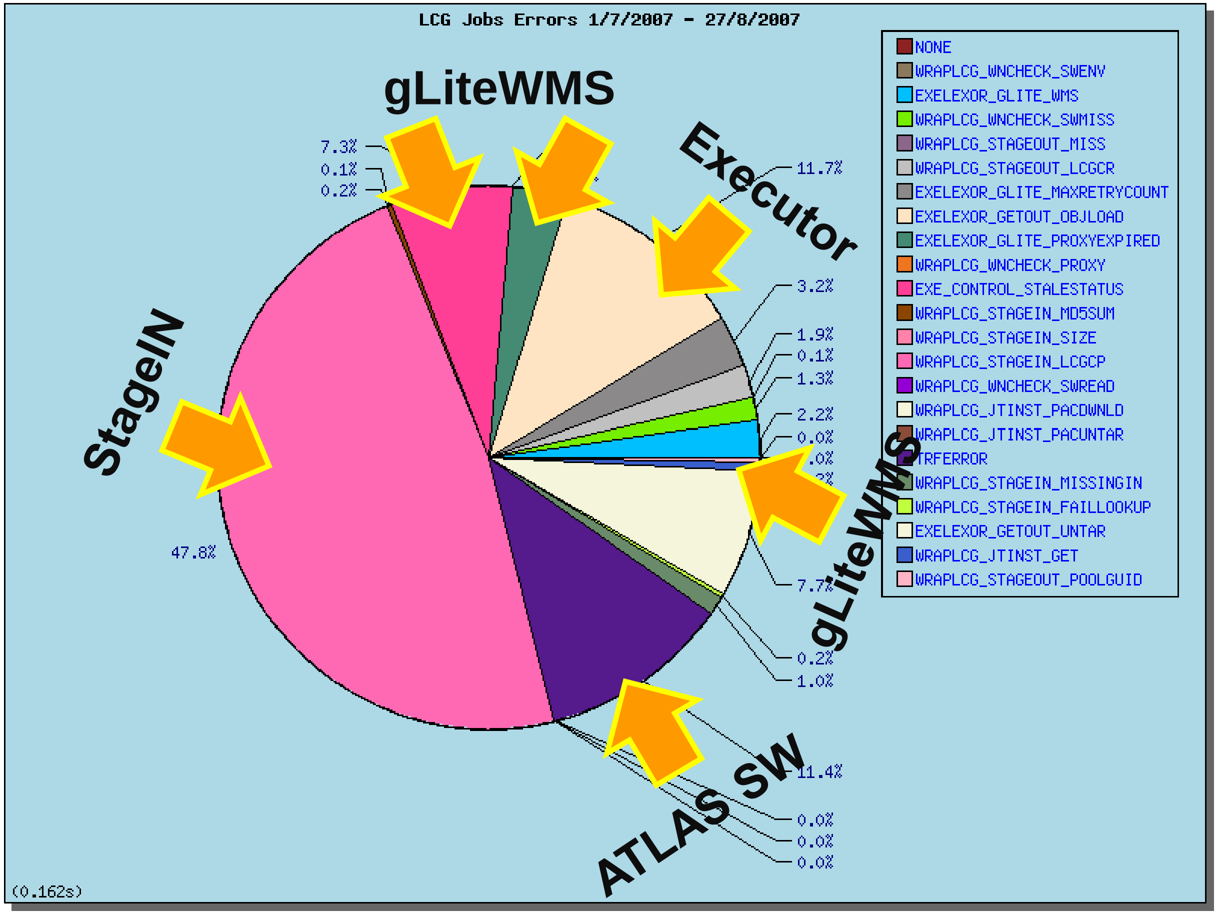


gLite WMS: ~22%

Data Management: 36%

ATLAS SW: 8%

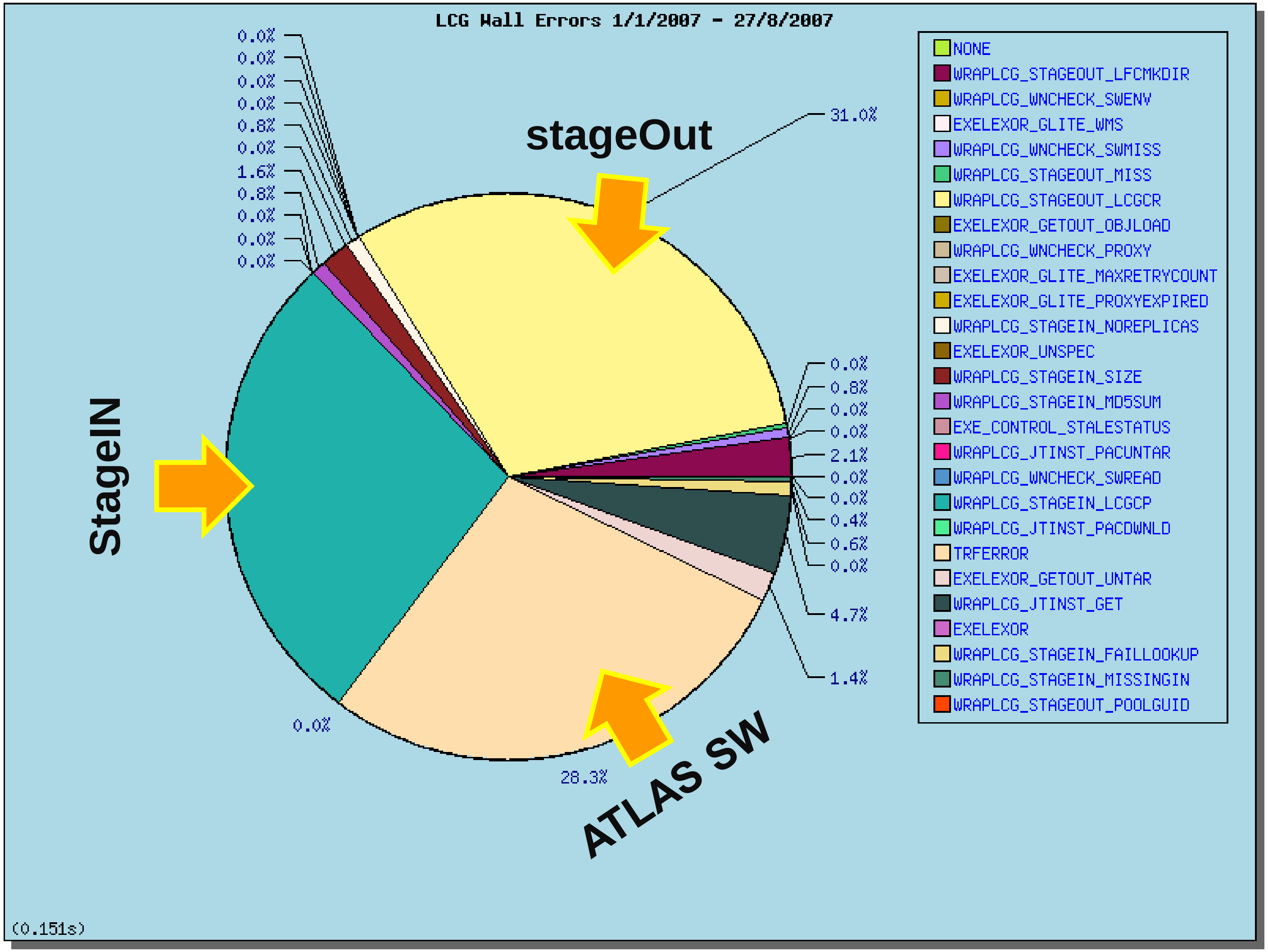
Number of Jobs Error Breakdown: July and August 2007



gLite WMS: ~13% **Data Management: 47%** **ATLAS SW: 11%**

gLite WMS category includes also site specific issues and problematic job distribution (with subsequent proxy expiration).

WallClockTime Error Breakdown: January to August 2007



gLite WMS: negligible **Data Management: ~60%** **ATLAS SW: 28%**

Statistics from Jan1st to August 27th

	Lexor	ATLAS Production (tot)
Finished Jobs	942439	4317243
Failed Jobs	746252	2208135
Job Efficiency	**55.8**	**66.2**
Finished WCT	17567932826	79018496695
Failed WCT	2550170383	11910469832
WCT Efficiency	**87.32**	**86.9**

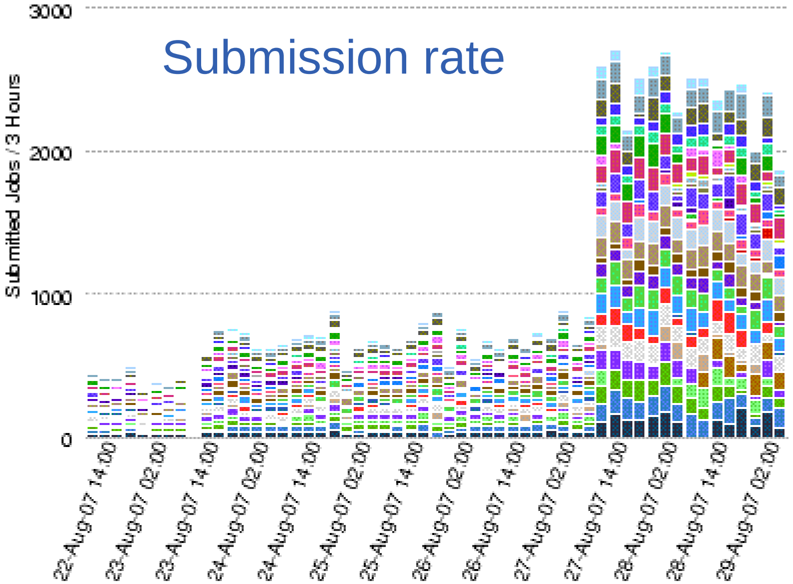
Lexor Job Efficiency is somewhat lower than ATLAS overall job efficiency

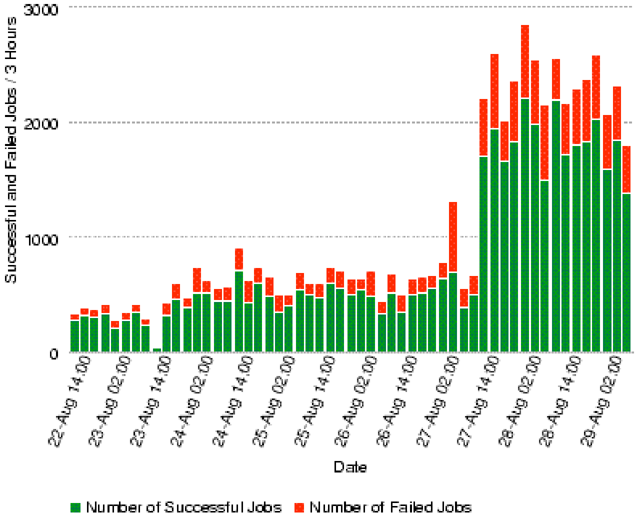
Automatic resubmission can cope with this

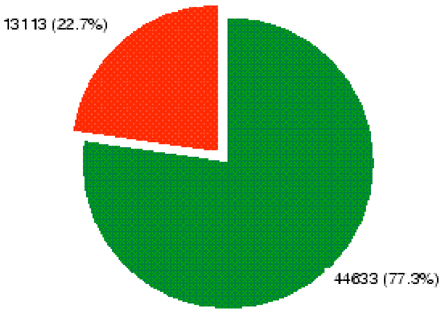
In terms of WCT (which means how much resources you are wasting for failed jobs) Lexor is quite efficient (more than ATLAS overall average)

The WMS in CMS data analysis

- **CMS supports submission of analysis jobs via WMS**
  - Using two WMS instances at CERN with the latest certified release
  - For CSA07 the goal is to submit at least 50000 jobs/day via WMS
  - The Job Robot (a load generator simulating analysis jobs) is successfully submitting more than 20000 jobs/day to two WMS







Success rate

Conclusions

- **Most reliability problems in gLite WMS are understood**
  - A few minor issues still being investigated
- **Several features of gLite WMS still to be considered**
  - Job Perusal: real time access to job stderr and stdout
  - Reputability Ranking: exclusion of resources causing large job failures
  - Job Provenance in Logging and Bookkeeping
- **Some improvements being discussed with developers**
  - e.g. stochastic ranking expression
- **The advantages compared to the LCG Resource Broker are very significant**
- **The achieved improvements had a big impact on many production activities**
  - e.g. for the ATLAS Monte Carlo production
- **All the LHC experiments are ready to use it**
  - Either they are already using it, or have finished the testing phase
- **Experimental services approach has shows to be extremely effective**
  - Adopted also for other components, i.e. CondorCE and CREAM