Improving QoS on EGEE with application-level site pre-selection

Wen-Jun Tan[1,2], Sorina Camarasu-Pop[1], Tristan Glatard[1]
[1]Creatis, CNRS, INSERM, Université de Lyon, France
[2]Nanyang Technological University, Singapore

EGEE'09 – Barcelona – September 21st 2009

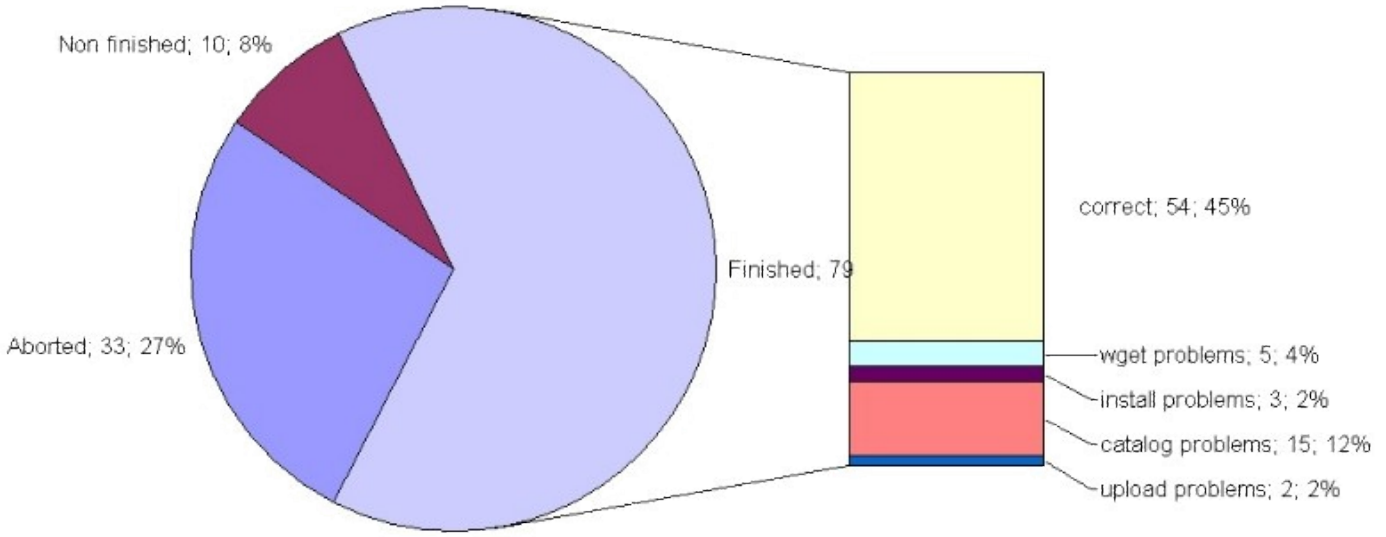
Job errors in biomed VO

- **Sequential jobs** [Jacq *et al*, JGC 2007]

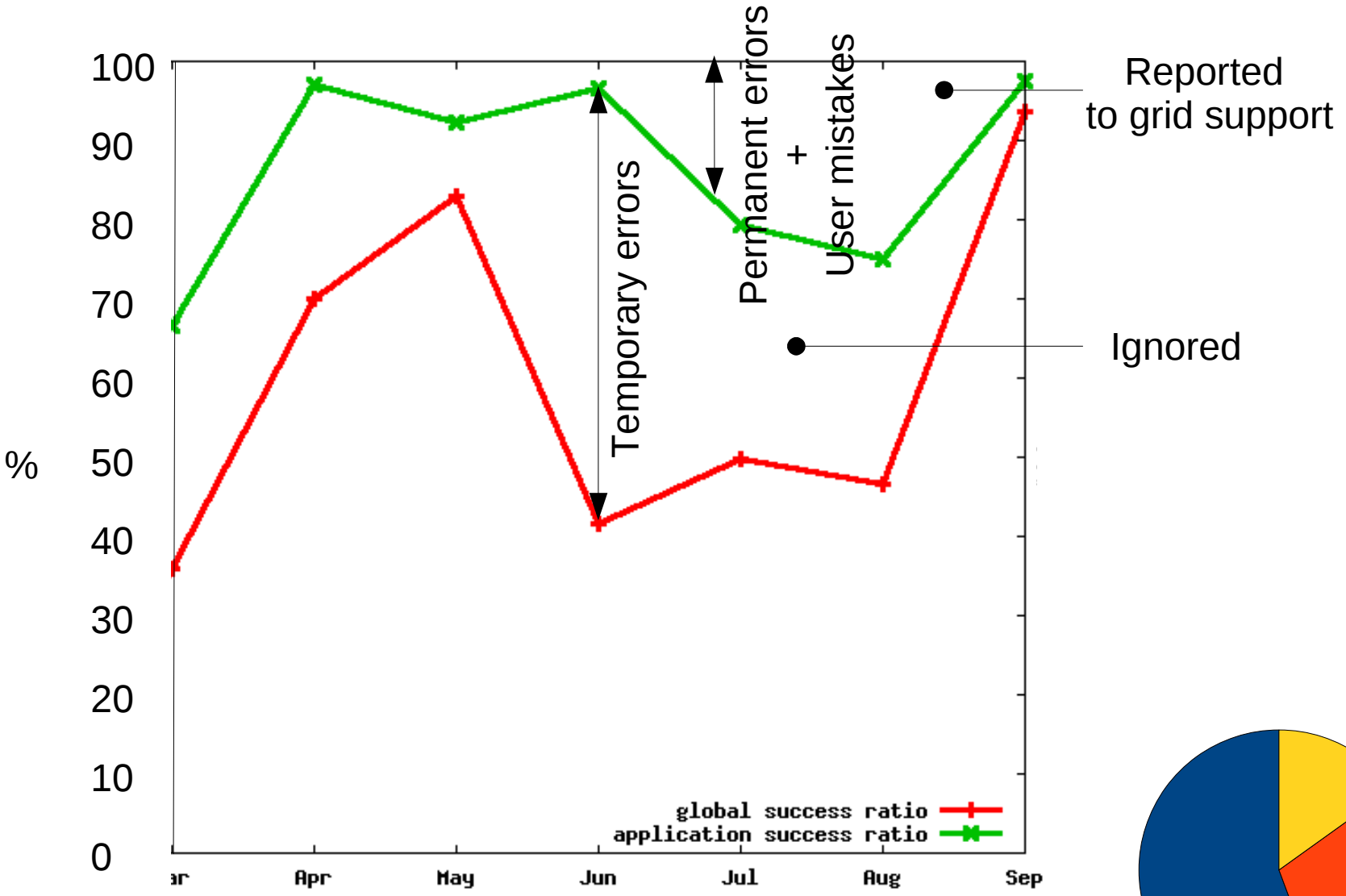
Metrics	Total	FlexX phase	Autodock phase
EGEE success rate	77 %	80,4 %	71,9 %
Success rate after checking output data	46,2 %	33,8 %	64,4 %
Success rate after checking output data and subtracting WISDOM and server license failures	63 %	61,6 %	65 %

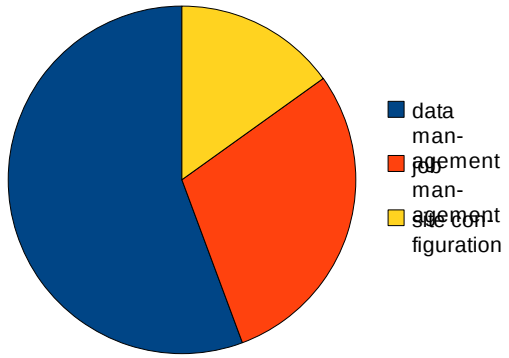
Table 4: Efficiency metrics of the WISDOM deployment.

- **MPI jobs** [Aparicio *et al*, HealthGrid'08]



Errors in the vlemed Dutch VO

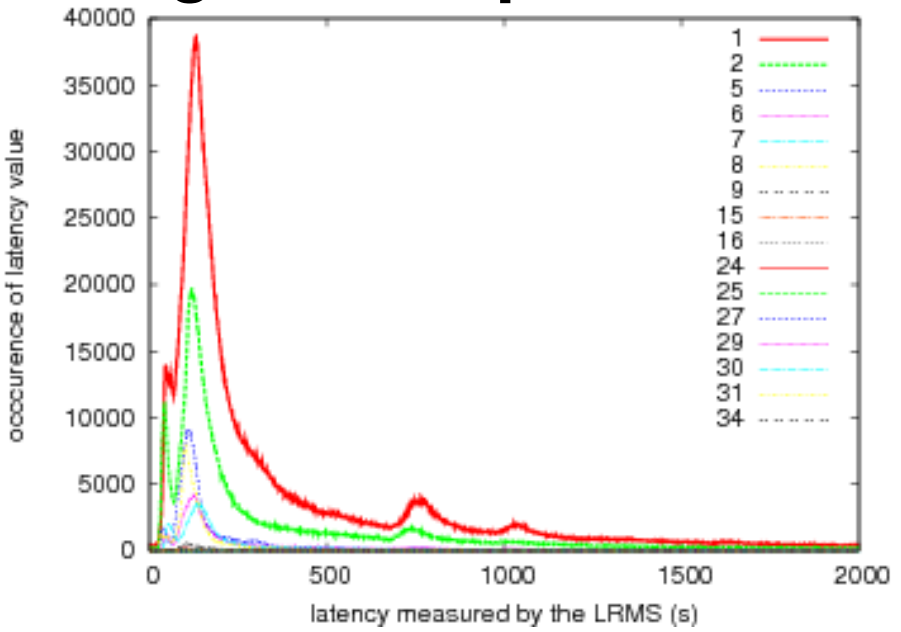




Error repartition (number of emails)

Overhead / latencies

- **Jobs face heterogeneous performance** [Lingrand *et al*, JSSPP'09]



RDRC – Job term. succ.
RDR – Job term. succ.
RDRC -
RDRC -
RDR - warnings
RDRC - warnings

- **It has serious impact on the applications** [Jacq *et al*, JGC 2007]

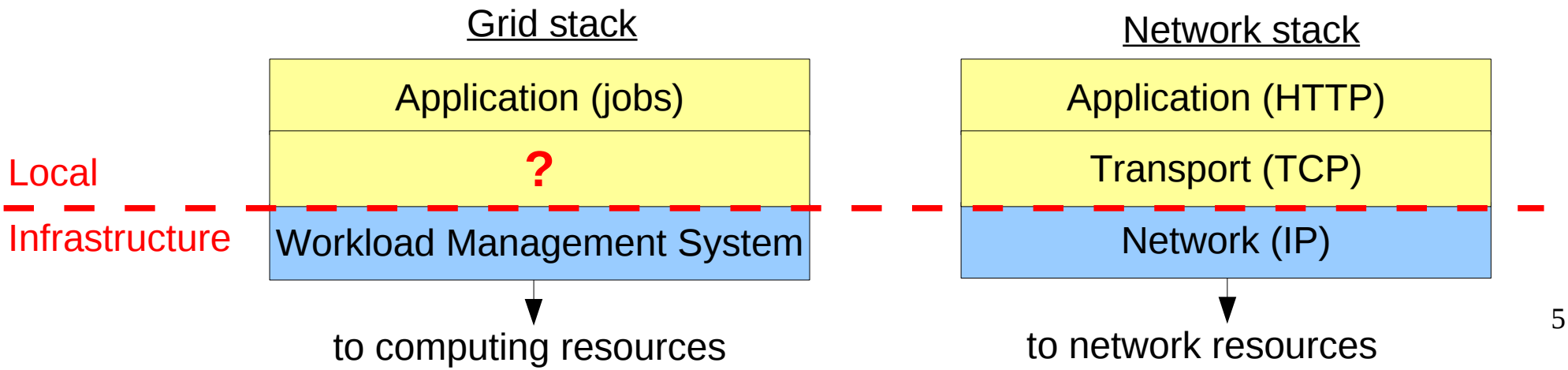
Metrics	Total	FlexX phase	Autodock phase
Total CPU time	80 years	29,5 years	50,5 years
Effective CPU time used by success jobs	67,2 years	24,8 years	42,4 years
Overhead time	77,1 years	25,9 years	51,2 years

Table 3: Performance metrics of the WISDOM deployment.

Application-level Quality of Service

- **Who should be handling errors ?**
  - People: users + developers + site admins (GGUS) ?
  - Middleware (e.g. WMS) ?
  - Dedicated agents ?
  - Application-level software ? ← *What we are looking into (for now)*

- **Assumption: infrastructure only provides best-effort**
  - Analogy with TCP/IP model [Meng *et al* IPDPS'09]



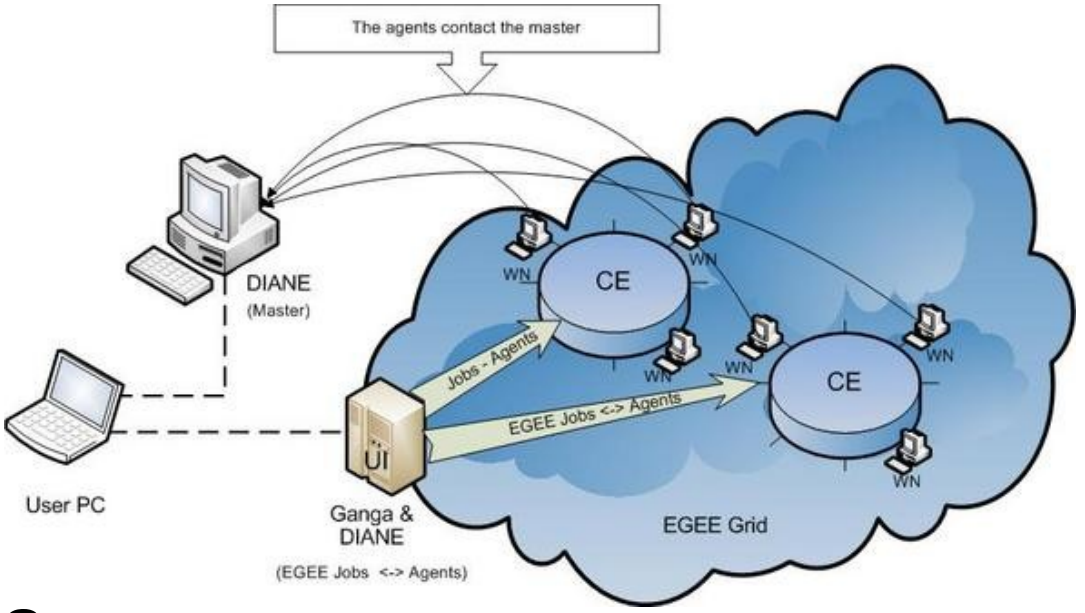
Application-level QoS: solutions

- **Job resubmission**
  - In case of errors
  - After timeout [Lingrand *et al*, HPDC'09]
- **Site pre-selection**
  - Based on site performance
  - Application specific
- **Pilot jobs**

Need a Grid Observatory !

Pilot jobs

- **“Pull” model (vs gLite “push)**

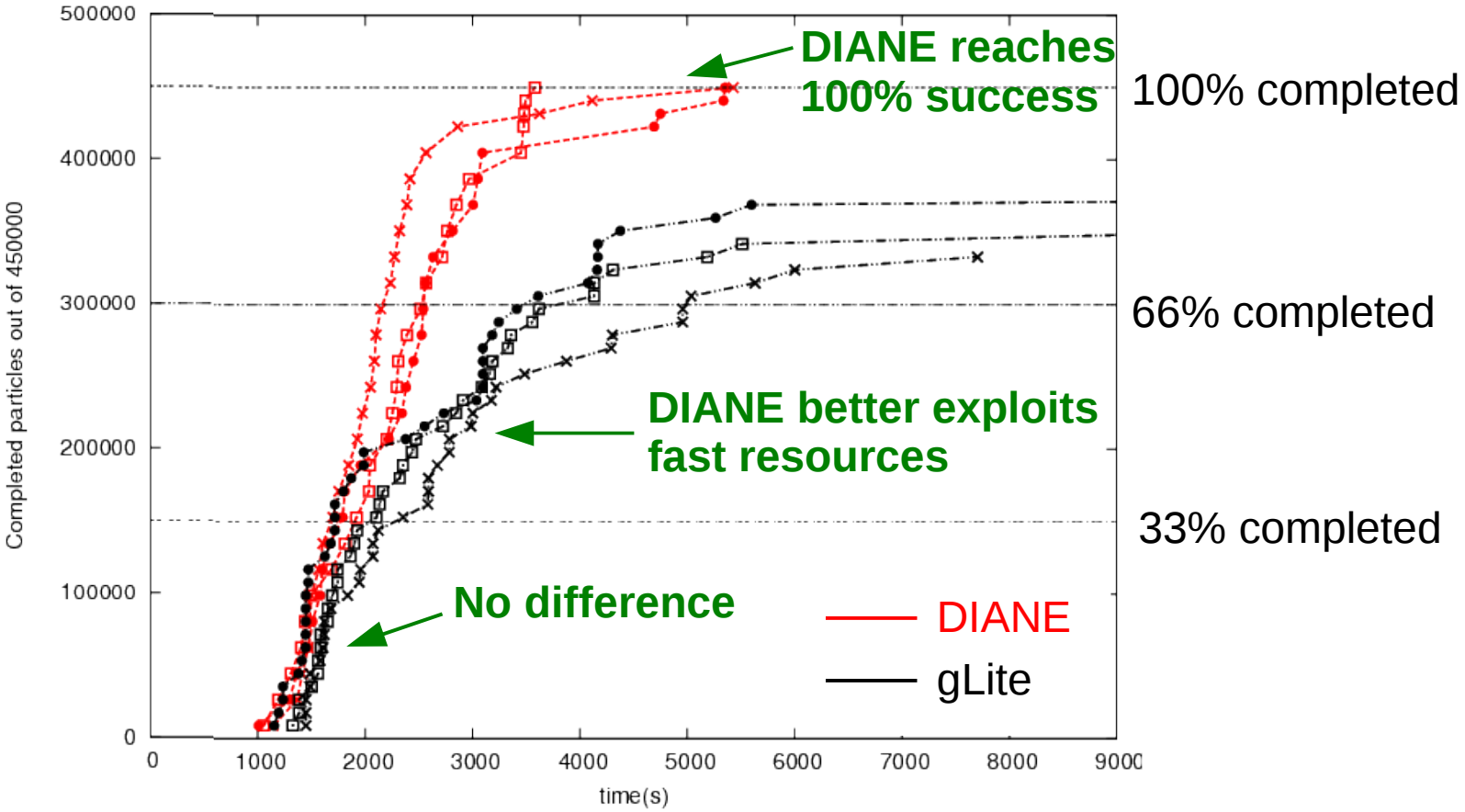


- **Advantages**
  - Fault-tolerance (faulty pilots are removed and others take over)
  - Heterogeneity handling (fast resources are privileged)
  - Latency/overhead reduction (bypass gLite WMS)

- **Very successful model**
  - DIANE, Funicolare, WISDOM-II, ToPOS, DIRAC, PanDA, Alien, etc.

Example: DIANE vs gLite-only

- **75 glite jobs VS 75 DIANE pilots and 75 tasks**
  - No job resubmission



Some limits of pilot jobs

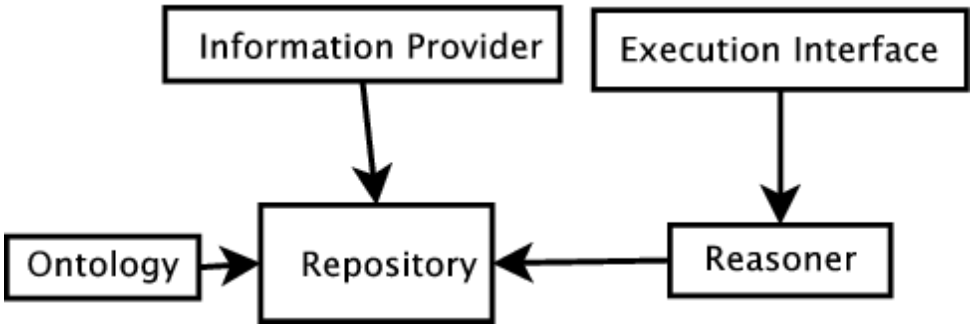
- **Task requirements**
  - Tasks are not gLite jobs (i.e. no GLUE requirements)
  - Master do not know about pilot (GLUE) requirements
- **Introduces (yet) another layer in execution**
  - Control of agent submission
  - Handle ports openings
- **Interoperability with other middleware**
  - e.g. with ARC (data transfer by CE VS late task-to-resource binding)
- **Security**
  - Pilot GSI authentication to master
  - Risk of compromised master

Grid Application-Level QoS Service (GALQS)

- **Monitoring**
  - One probe per application and per CE
  - Probes submitted once per day (at 1am)
  - Probes check application execution time/correctness
  - Maintain database of error ratios and latencies per CE

- **Site pre-selection**
  - Select only CEs with 100% reliability
  - Select x% fastest CEs
  - Performed before submission to WMS -> JDL requirements adaptation

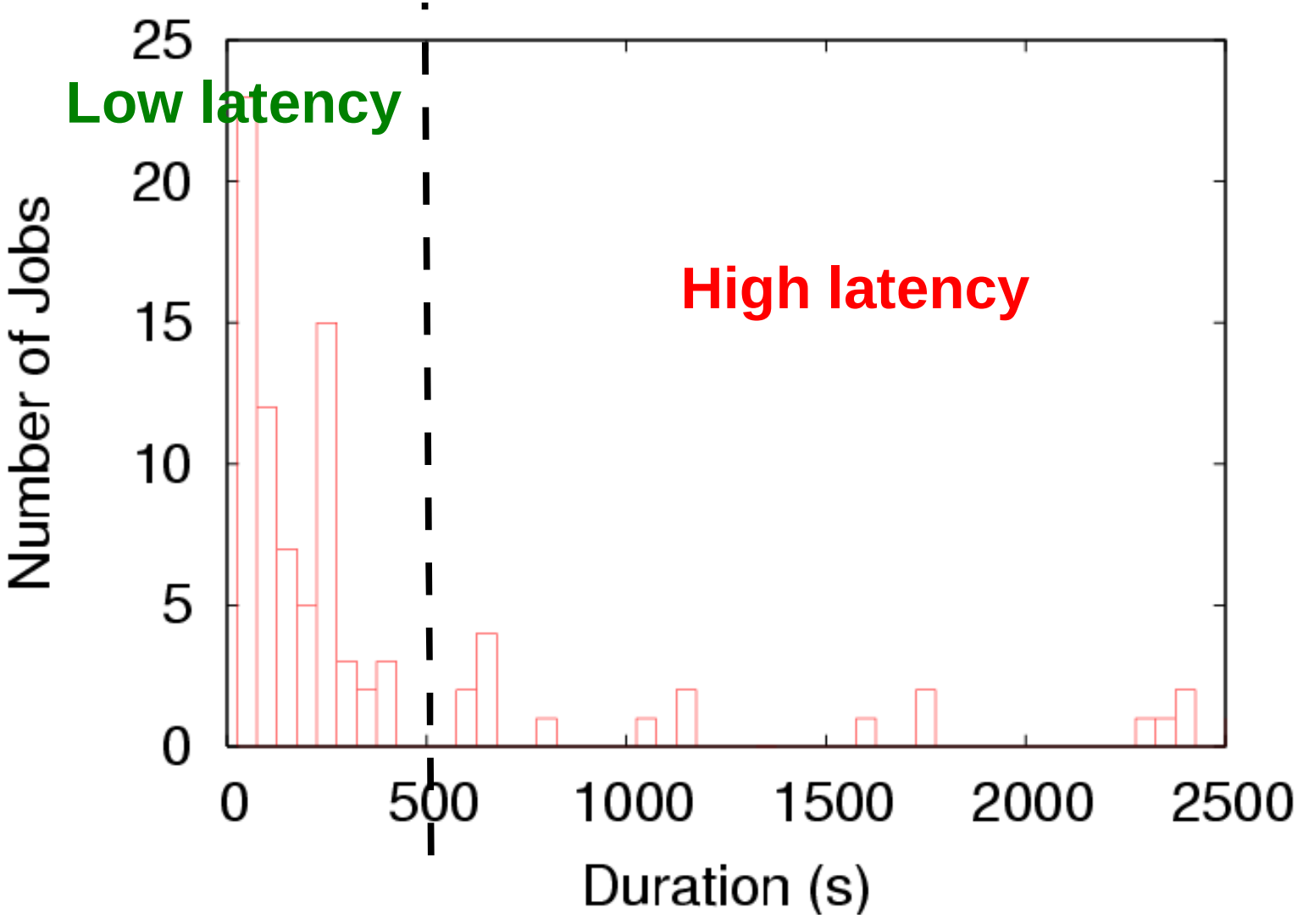
System architecture



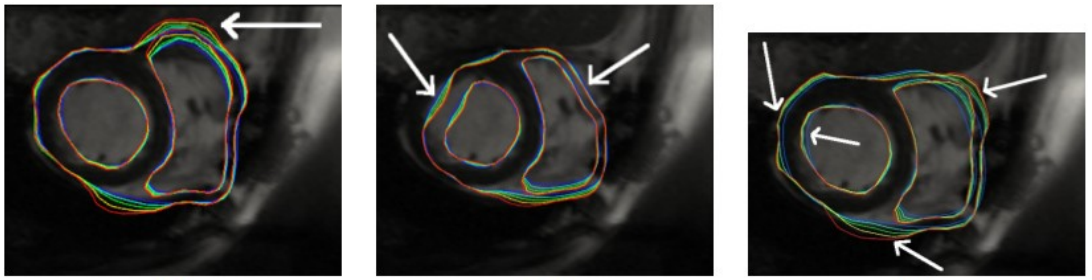
- **RDF repository (Sesame)**
- **Information provider**
  - Application-specific probe jobs
  - Periodically submitted
- **Ontology**
  - Probe type (MPI, pilot, application, dummy, data transfer, etc)
  - Resources (CEs, SEs, WMS)
  - Measured values (latency, success ratio)
- **Reasoner**
  - Implement RDF queries
  - Select CEs with **100% of reliability** for application X
  - Select CEs with 100% of reliability **and low latency** for application X

CE selection based on latency

- **A CE has low latency iif it belongs to the first mode of the latency distribution**

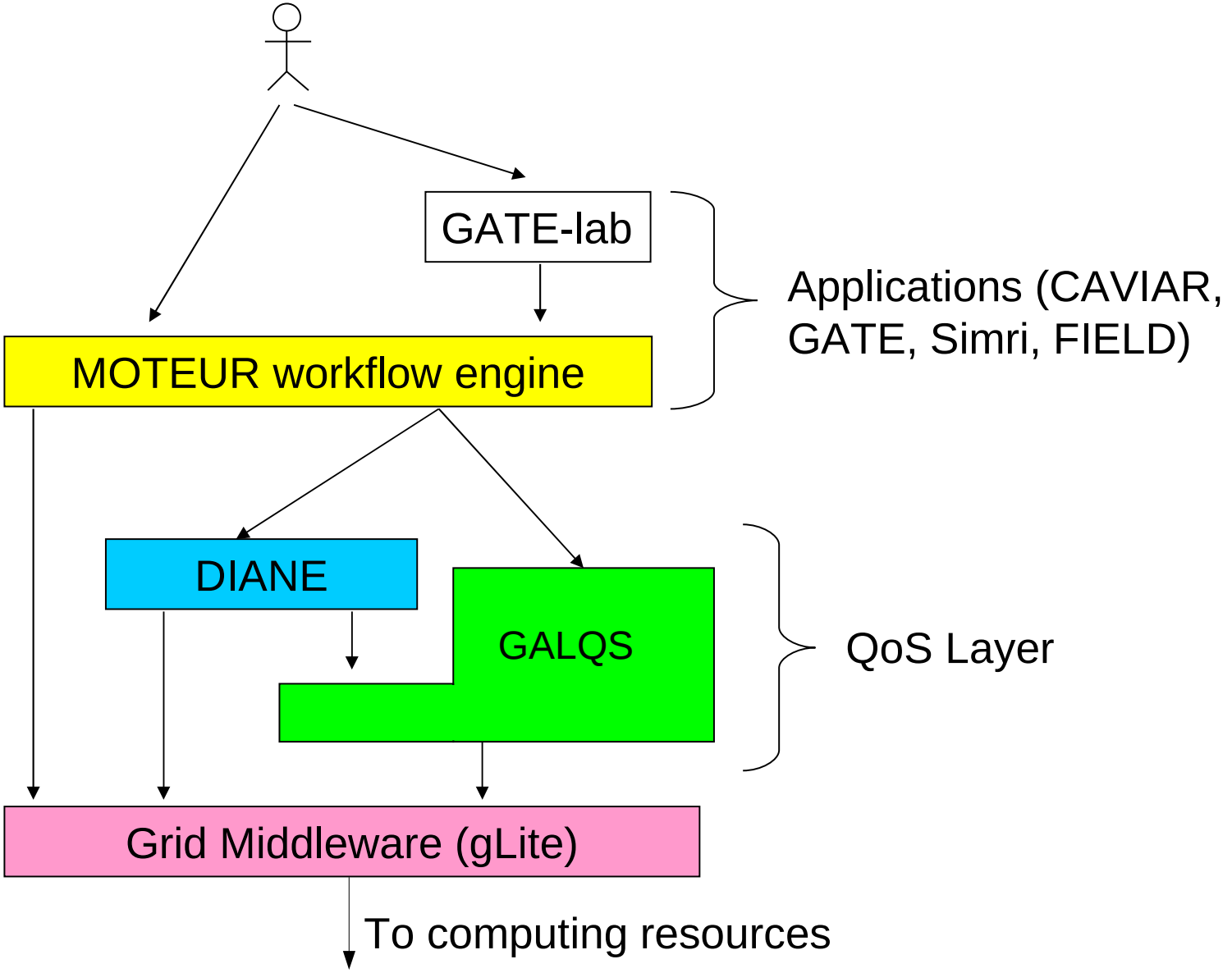


Experiment

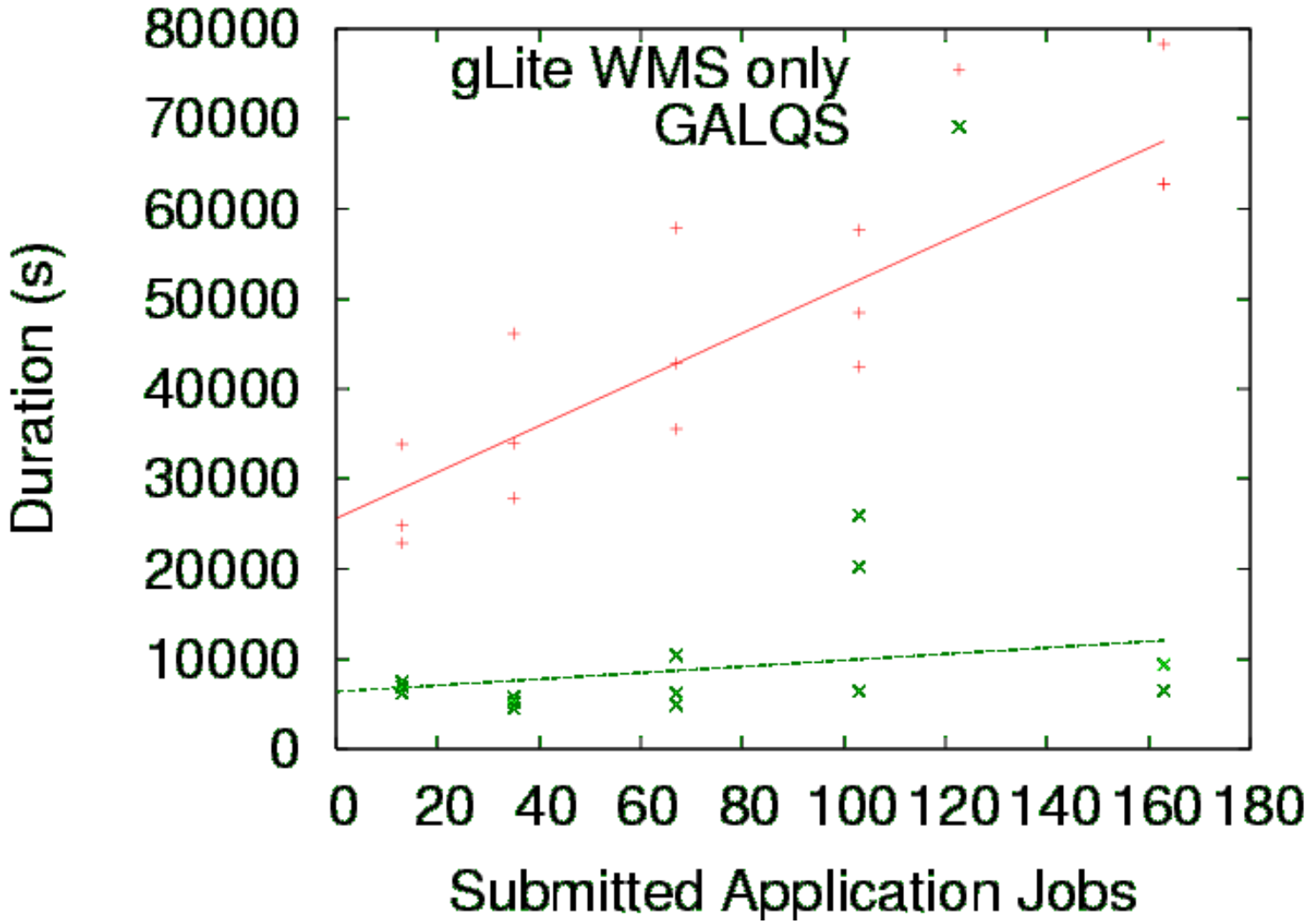


- **Application**
  - “CAVIAR” cardiac 3D segmentation
- **Probes Parameters**
  - submit at 1:00am
  - Time out: 6 hours
- **Comparisons**
  - gLite WMS vs GALQS
  - GALQS vs DIANE
  - DIANE vs DIANE + GALQS
- **Evaluation metric: time to reach 100% success**
  - High retry count: 100
  - Time out: 2 hours

Application Setup

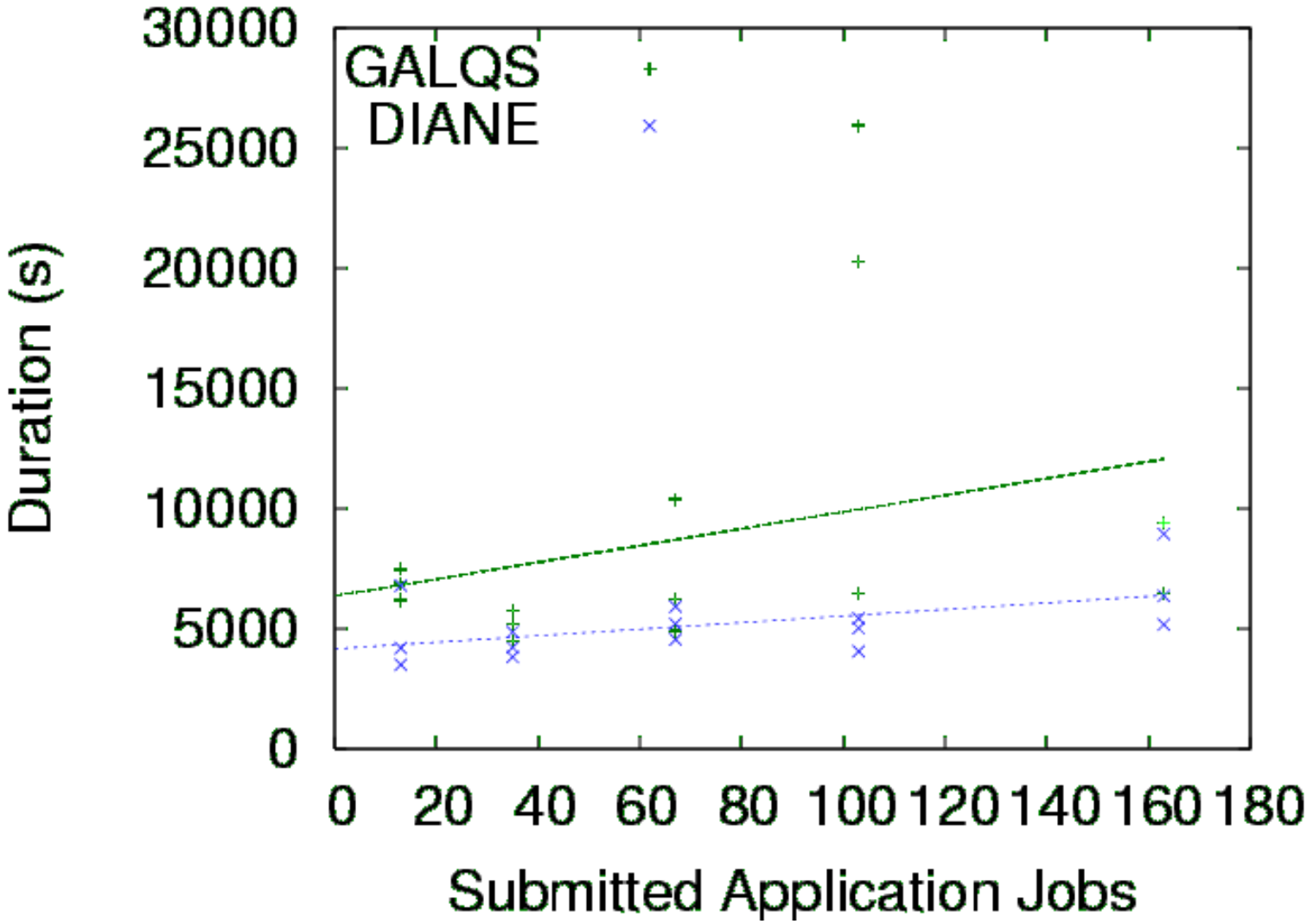


gLite WMS vs. GALQS



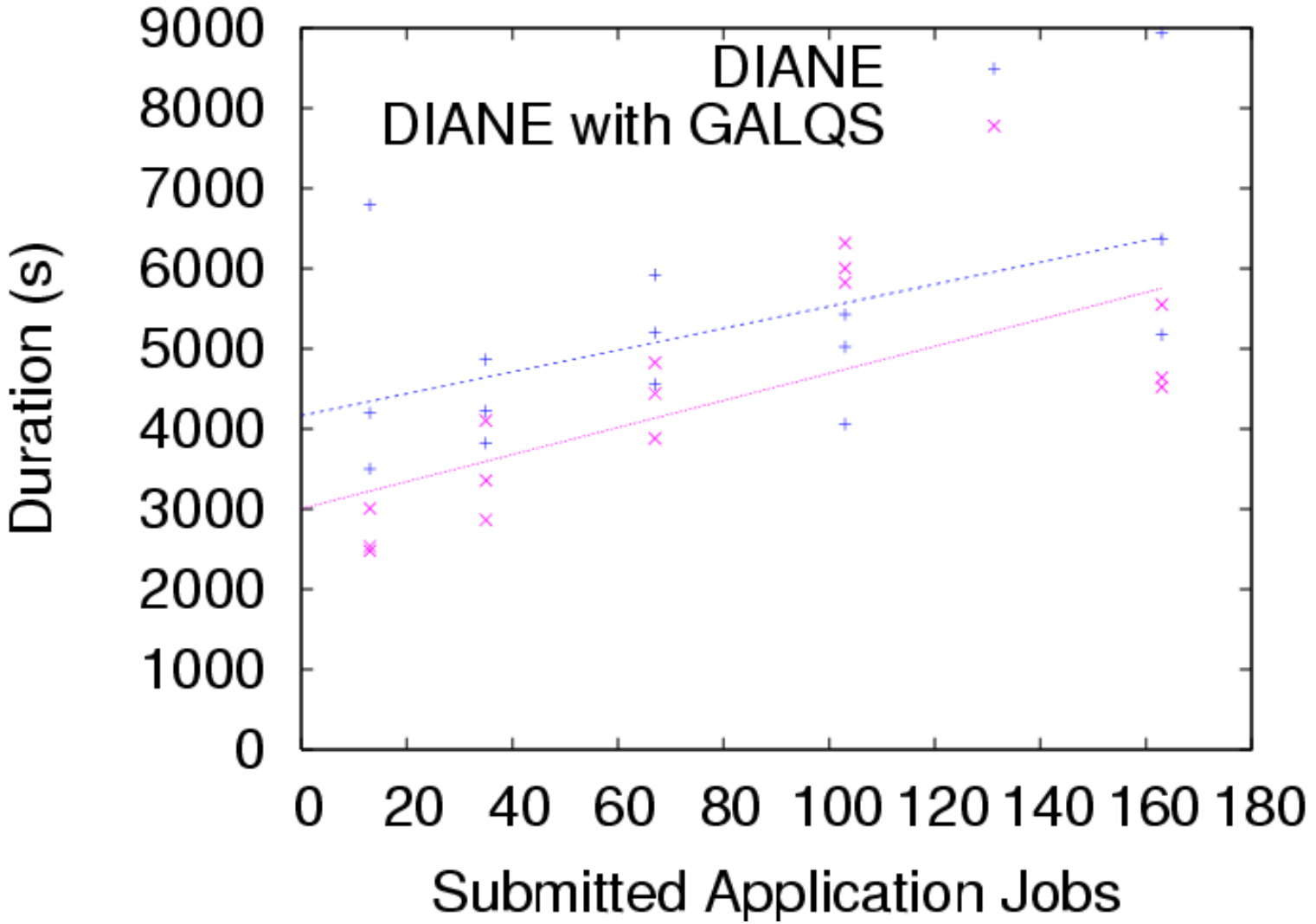
- **Improves throughput by a factor 5**

GALQS vs. DIANE

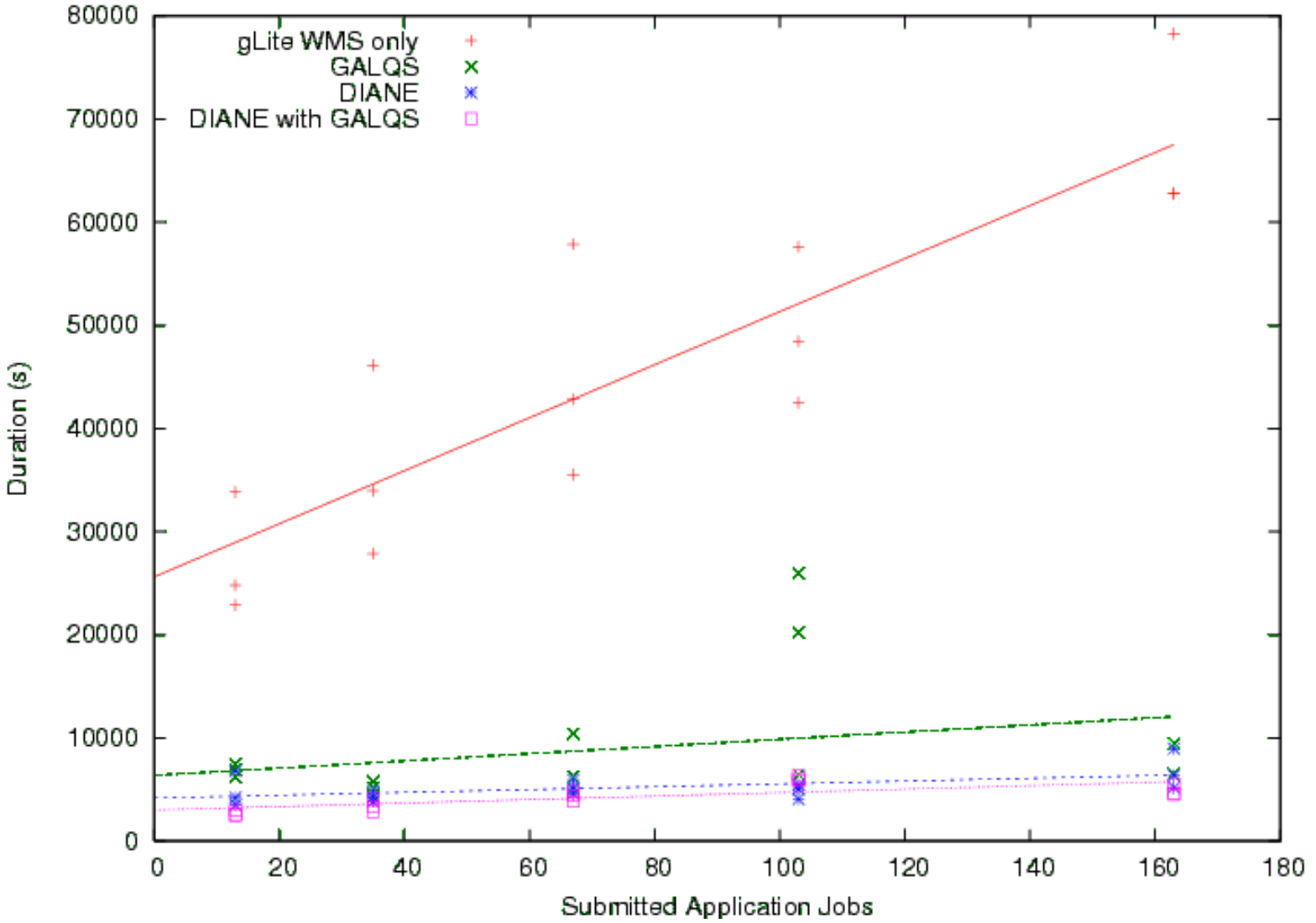


- **Equivalent in some cases**

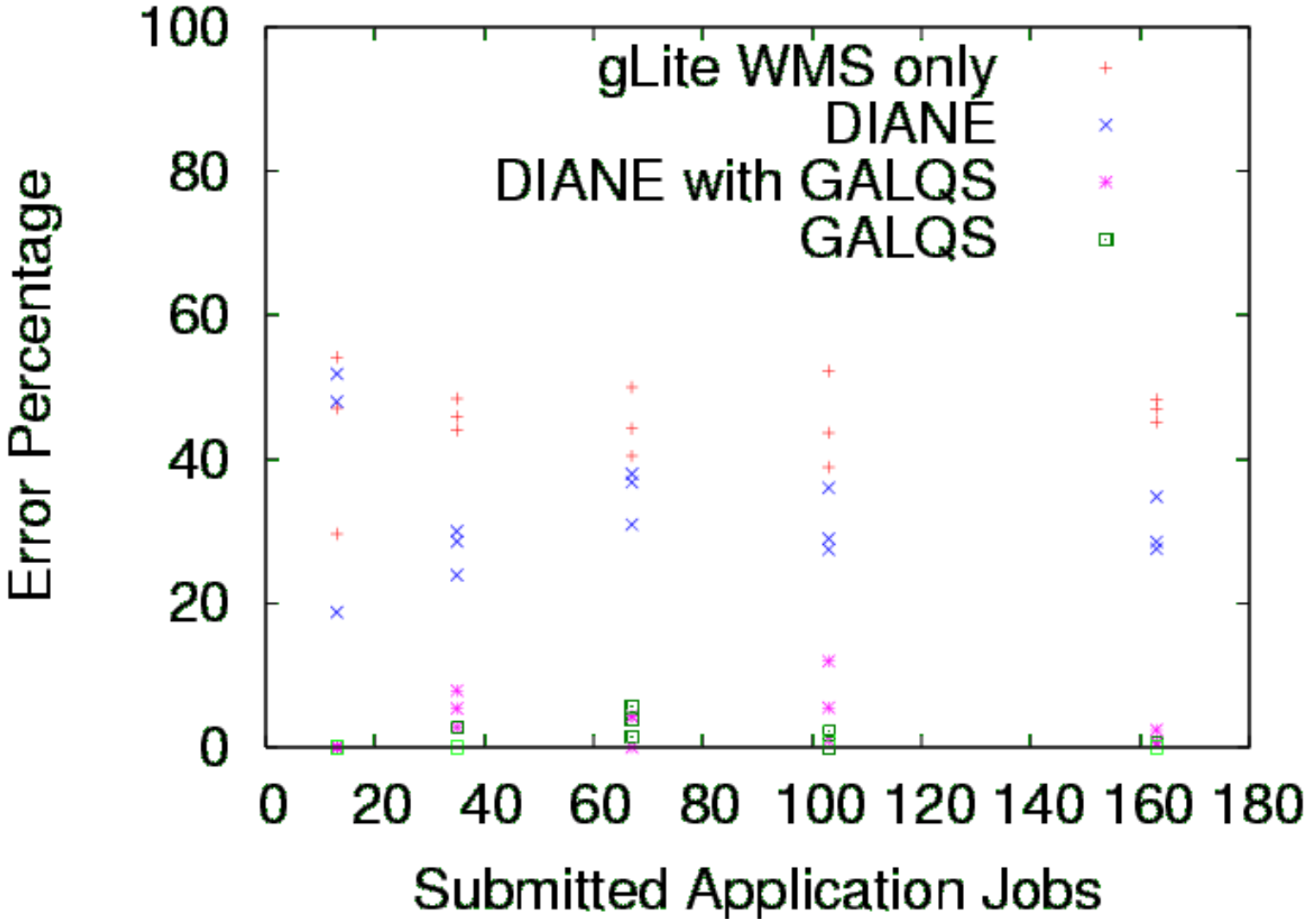
DIANE vs. DIANE + GALQS



Overall comparison



Error Percentages



- **Number of submitted jobs is reduced by 40%**

Conclusions

- **Lessons learned**
  - Application-specific site pre-selection improves throughput by a factor 5 compared to regular gLite WMS submission
  - Equivalent to pilot jobs in some cases
  - Reduces the number of submitted jobs by 40%
- **Future work**
  - Include application feedback
  - Merge with IN2P3 Java Job Submission (JJS)

http://cc.in2p3.fr/docenligne/269

Vacancies...

- **At Creatis, Lyon, France**
  - 30-month software engineer position
  - 1 master student on grids for radiotherapy simulation
  - 1 master student on grids for cardiac image analysis

glatard@creatis.insa-lyon.fr

- **At the University of Amsterdam, NL**
  - Software engineer
  - PhD student
  - Post-doc

silvia@science.uva.nl