Testing SLURM batch system for a grid farm: functionalities, scalability, performance and how it works with Cream-CE

DONVITO GIACINTO (INFN)

ZANGRANDO, LUIGI (INFN)

SGARAVATTO, MASSIMO (INFN)

REBATTO, DAVID (INFN)

MEZZADRI, MASSIMO (INFN)

FRIZZIERO, ERIC (INFN)

DORIGO, ALVISE (INFN)

BERTOCCO, SARA (INFN)

ANDREETTO, PAOLO (INFN)

PRELZ, FRANCESCO (INFN)

Outline

- Why we need a "new" batch system
  - INFN-Bari use case
- What do we want from a batch system?
- SLURM short overview
- SLURM functionalities test
  - … fail-tolerance considerations
  - … pros & cons
- SLURM performance test
- CREAM support to SLURM
- Future Works
- Conclusions

Why we need a “new” batch system

- Multi-Core CPU are putting pressure on batch system as it is becoming quite common to have computing farms with O(1000) CPU/cores
- Torque/MAUI is a common and easy-to-use solution for small farms
  - It is open source and free
  - Good documentation
  - and wide user base
- …but it could start suffering as soon as the farm becomes larger
  - in terms of Cores
  - and of WN
  - … but especially in terms of users

Why we need a "new" batch system:

INFN-Bari use case

- We started with few WN in 2004 and constantly growing
  - we now have about:
    - 4000 CORES
    - 250 WNs
- We have Torque 2.5.x + MAUI:
  - We see a few problem with this setup:
    - "Standard" MAUI supports up-to ~4000 queued jobs
      - All the "others" jobs are not considered in the scheduling
    - We modified the MAUI code to support up to 18000 queued jobs and now it works
      - … but it often saturates the CPU where it is running and soon it becomes un-responsive to client interaction

Why we need a "new" batch system:

INFN-Bari use case (2)

- Torque is suffering from memory leak:
  - It usually use ~2GB of memory under stress condition
  - We need to restart it from time to time
- Network connectivity problems to a few nodes could affect the whole Torque cluster

- We need a more reliable and scalable batch system and (possibly) … open source and free of charge

What we need from a batch system

- Scalability:
  - How it deals with the increasing number of Cores and jobs submitted
- Reliability and Fault-tolerance
  - HighAvailability features, client behavior in case of service failures
- Scheduling functionalities:
  - The INFN-Bari site is a mixed site, both grid and local users share the same resources
    - We need complex scheduling rules and full set of scheduling capabilities
- TCO
- Grid enabled

SLURM short overview

- OpenSource (https://computing.llnl.gov/linux/slurm/)
- Used by many of the TOP500 super-computing centers
- Documentation states that:
  - It supports up to 65’000 WNs
  - 120’000 jobs/hour sustained
  - High Availability features
  - Accounting on Relational DataBase
  - Powerful scheduling functionalities
  - Lightweight
  - It is possible to use MAUI/MOAB or LSF as scheduler on top of SLURM

SLURM functionalities test

- Functionalities tested:
  - QoS
  - Hierarchical Fair-share
  - Priorities on users/queue/group etc.
  - Different pre-emption policies
  - Client resilience on temporary failures
    - The client catchs the error and retries after a while automatically
  - The server could be configured with HighAvailability configuration
    - This is not so easy to configure
    - It is based on "events"
  - The accounting information stored on MySQL/PostgreSQL DB
    - This is also the only way to configure the Fair-Share

SLURM functionalities test (2)

- Functionalities tested:
  - Age based priority
  - Support for Cgroup for limiting the usage of resources on the WN
  - Support for basic “consumable resources” scheduling
  - “Network topology” aware scheduling
  - Job suspend and resume
  - Different kind of jobs tested:
    - Interactive jobs
    - MPI jobs
    - “Whole node” jobs
    - Multi-threaded jobs
  - Limits on amount of resources usable at a given time for:
    - Users, groups, etc.
    - It is possible to limit also the number of submitted jobs (Queued)

SLURM functionalities test (3)

- Functionalities tested:
  - Computing resources could be associated to:
    - Users, group, queue, etc
  - ACL on queues, or on each of the associated nodes
  - Job Size scheduling (Large MPI Jobs first or small jobs first)
  - It is possible to submit executable directly from CLI instead of writing a script and submitting it
  - The jobs lands on the WN exactly in the same directory where the user was when it is submitting the jobs
  - Triggers on events
  - Any batch job running on a failed node will be requeued for execution on different nodes

SLURM results: pros & cons

- The scheduling functionalities is powerful but can be enriched by means of using MOAB or LSF scheduler
- Security is managed using "*munge*" as with the latest version of Torque
- There is no RPM available for installing it but it is quite easy to compile from the source code
- There is no way to transfer the output files from the WN to the submission host
  - The system is built assuming that the working file system is shared
- Configuring complex scheduling policy is quite complex and requires a good knowledge of the system
  - Documentation could be improved with more advanced and complete examples
  - There are only few source of information apart from the official site

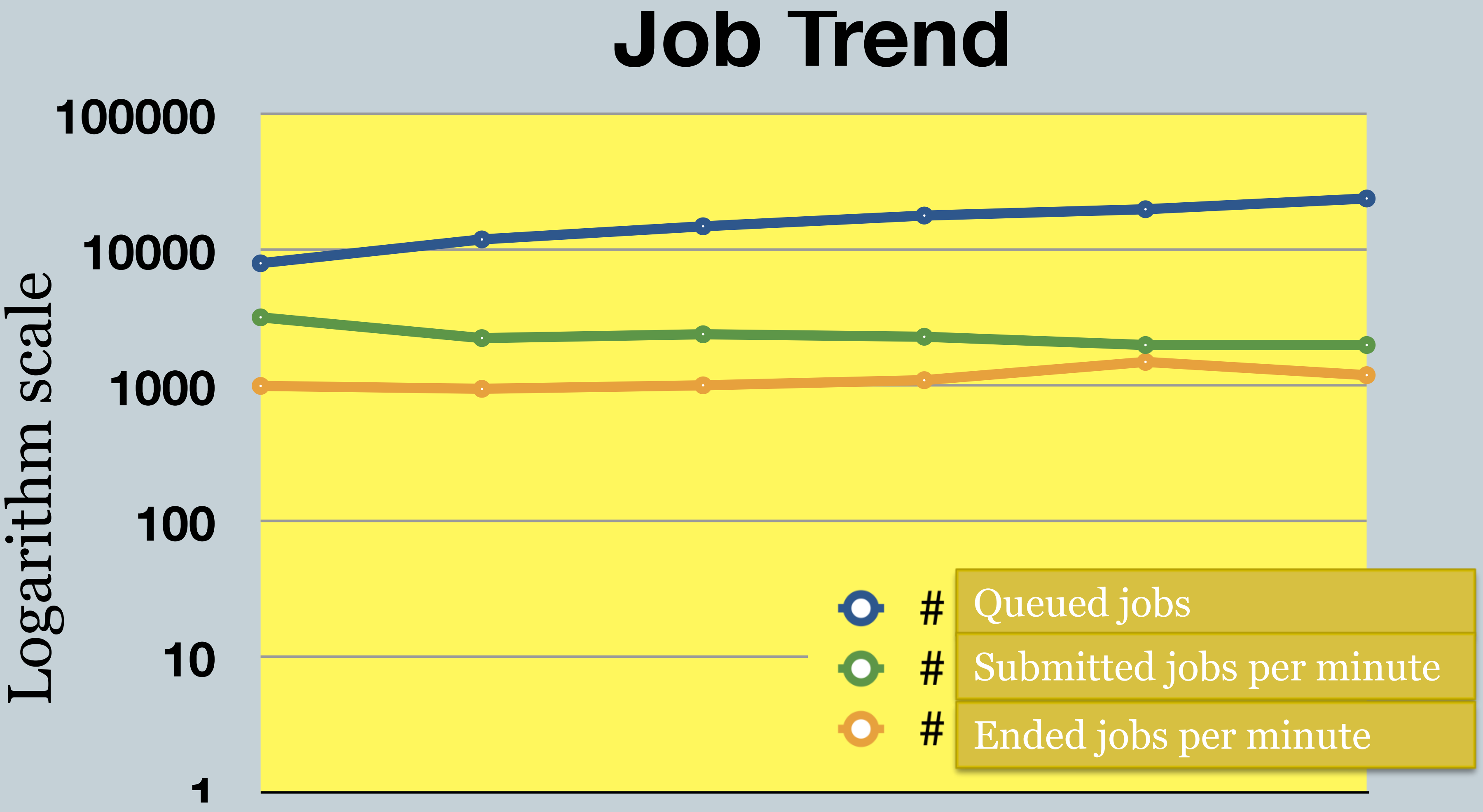
Performance test: description

- We have tested the SLURM batch system in different stressing conditions:
  - High amount of jobs in queue
  - Fairly high number of WNs
  - High number of concurrent submitting users
  - Huge amount of jobs submitted in a small time interval
  - The accounting on the MySQL databases is always enabled

Performance test: description (2)

- High number of jobs in the queue:
  - One single client is constantly submitting jobs to the server for more than 24 hours
  - The jobs are fairly long…
  - … so the number of jobs in the queue are increasing constantly
  - We measured:
    - the number of queued jobs
    - the number of submitted job per minutes
    - the number of ended jobs per minutes
- The goal is to prove:
  - the reliability of the system under high load
  - the ability to cope with the huge amount of jobs in the queue keeping the number of executed and submitted job as constant as possible

Performance test: results (1)



Performance test: results (2)

- The test was measured up to 25kjobs in queue
- No problems registered
  - The server was always responsive and the memory usage is as low as ~200MB
  - The submission rate is decreasing slowly and gracefully
  - … the number of executed jobs is not decreasing
    - This means that the jobs scheduling on the nodes is not suffering
  - We were able to keep a scheduling period of 20 seconds without any problem
  - The loadaverage on the machine is stable at ~1
- TEST PASSED ☺

Performance test: description (3)

- High amount of WNs
- High number of concurrent clients submitting jobs:
- Huge number of jobs to processed a short period of time:
  - 250 WNs
    - ~6000 Cores
  - 10 concurrent client …
  - … each submitting 10’000 jobs
  - Up to 100’000 job to be processed

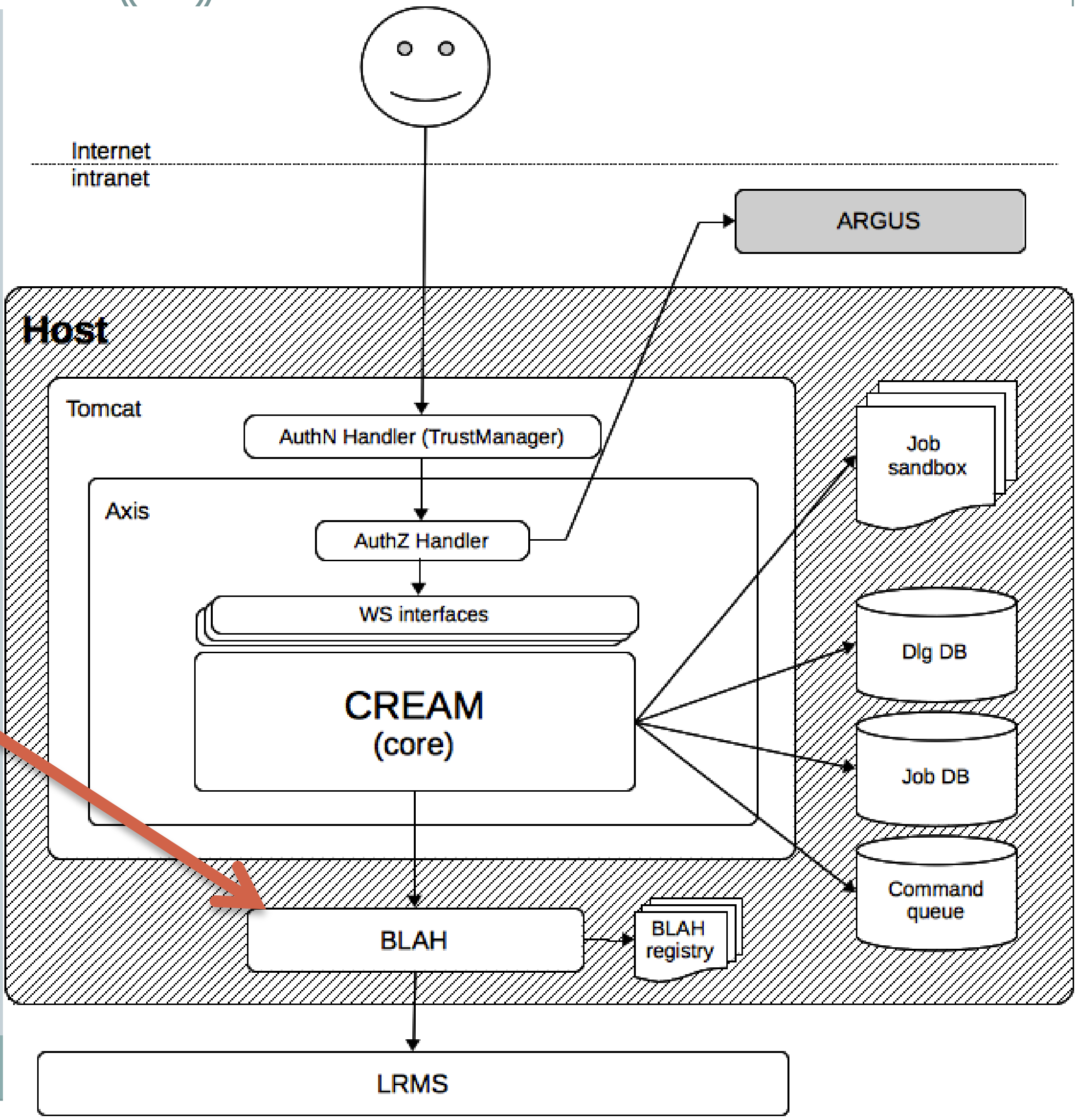
- The goal is to prove:
  - the reliability of the system under high load from the clients
  - The ability to deal with a huge pick of job submission
  - Managing a quite large farm

Performance test: results (3)

- The test was executed in about 3.5 hours
- No problems registered
  - The submission do not experienced problems
  - the memory used on the server always less than 500MB
  - The loadaverage on the machine is stable at ~1.20
  - At the beginning of the test the submission/execution rate is 5,5kjob per minute
  - During the pick of the load:
    - the rate of submission/execution is about 350 job/minute
  - It was evident that the bottleneck is on the single CPU/Core computing power
- TEST PASSED ☺

CREAM CE & SLURM

- Interaction with the underlying resource management system implemented via BLAH
- Already supported batch systems: LSF, Torque/PBS, Condor, SGE, BQS



CREAM & SLURM

- The testbed in INFN-Bari was originally used to develop and test the submission scripts by the CREAM team
  - Those scripts takes care also of the file transfers among WN and CE
  - The basic idea is to provide the same functionalities on all the supported batch systems
- CREAM status:
  - BLAH script => OK ☺
    - Already tested from a site in Poland
  - Infoprovider => OK ☺
  - APEL Sensors => Work-in-progress 😐
- If you are interested in testing/provide feedback or develop some missing piece, please contact us!

Future Works

- We will go on testing additional features and configuration:
  - pre/post exec files
  - Mixed configuration (SLURM+MAUI or SLURM+LSF)
  - More on “triggers”
- We will test the possibility to exploit SLURM as batch system for the EMI WNoDeS cloud and grid virtualization framework

Conclusions

- The test on SLURM carried on at INFN-Bari highlight the optimal performance and functionalities of this batch system
- Looks quite promising for medium-large farms that do not want to use proprietary batch systems.
- There is a need for improving test/documentation/best practice/how-to etc.
  - We need volunteers to set-up a common repository of documentation and other useful materials