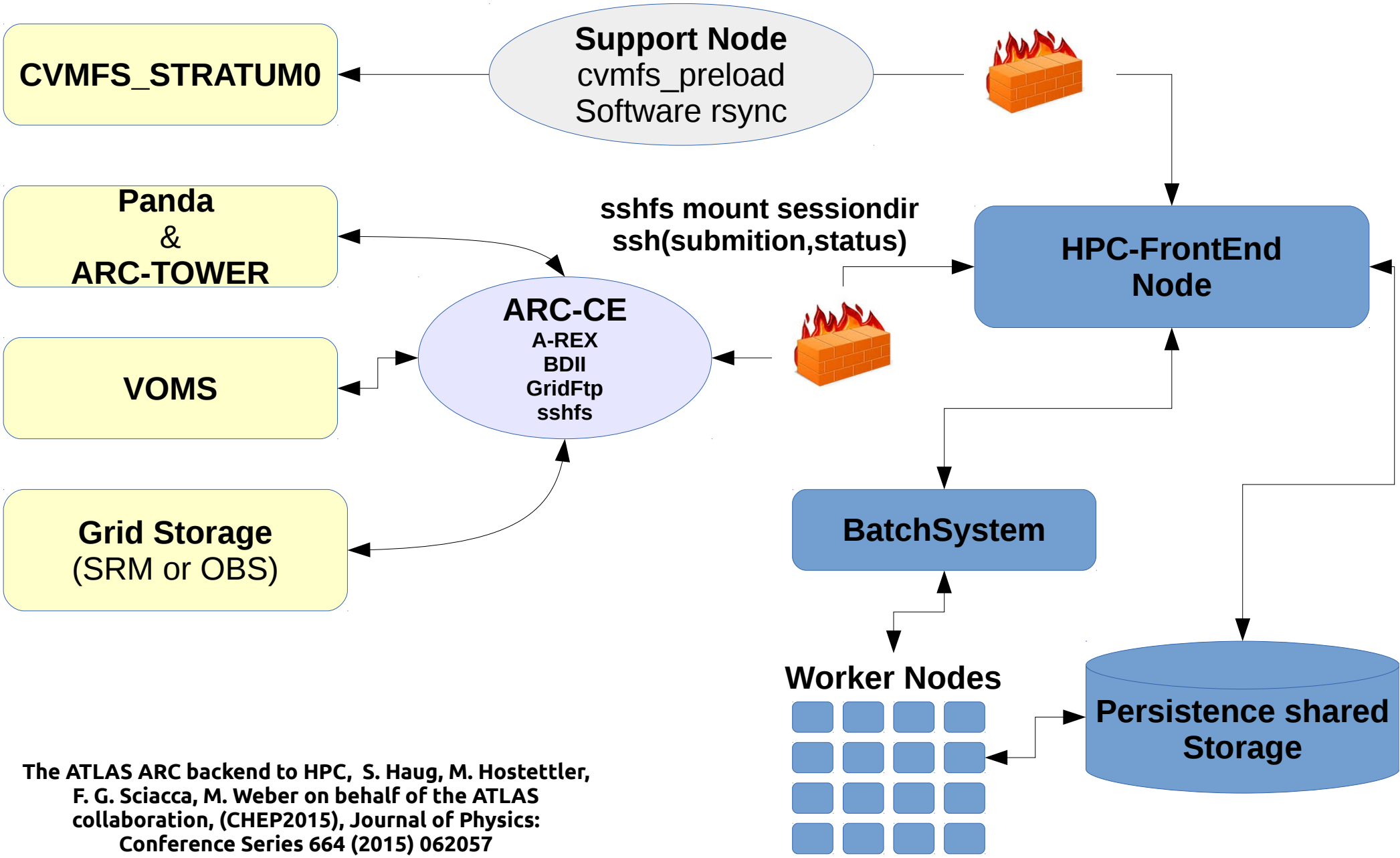
Interface CC-IDRIS pour les jobs Grille - Cas d'utilisation d' Atlas

Vamvakopoulos Emmanouil

IN2P3-CC, LYON

Context Diagram

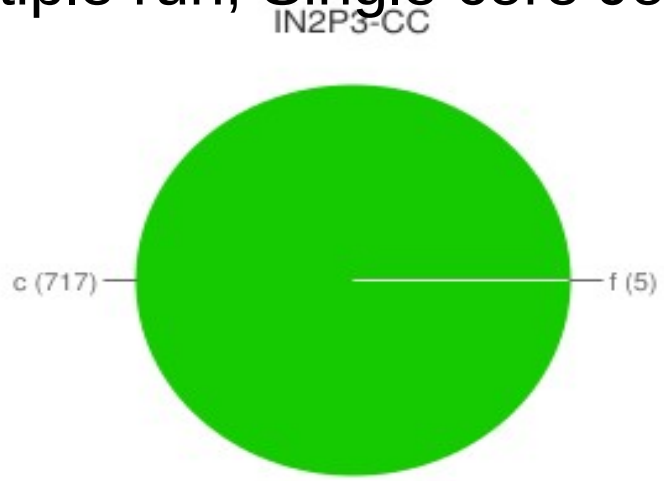


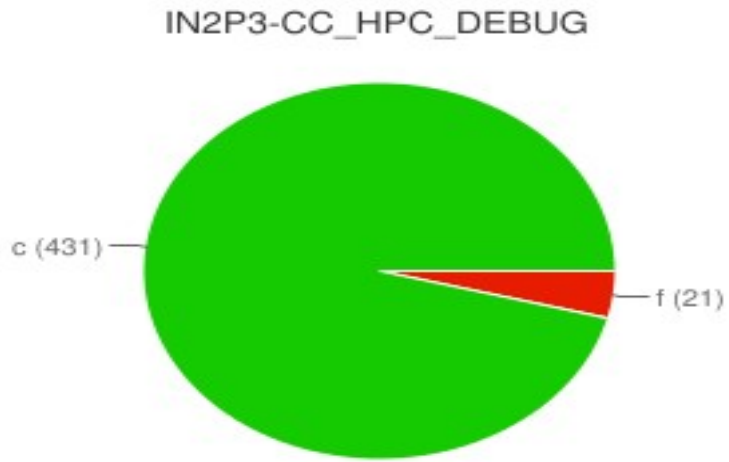
The ATLAS ARC backend to HPC, S. Haug, M. Hostettler, F. G. Sciacca, M. Weber on behalf of the ATLAS collaboration, (CHEP2015), Journal of Physics: Conference Series 664 (2015) 062057

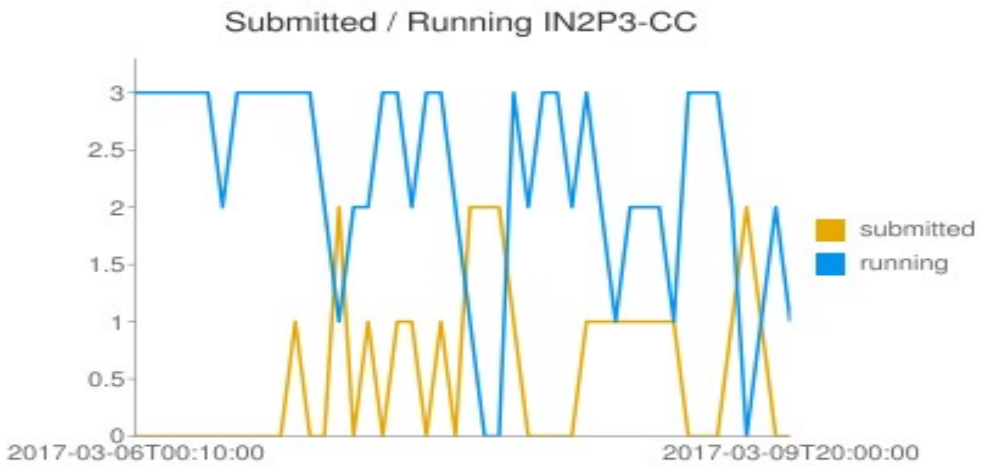
HammerCloud tests

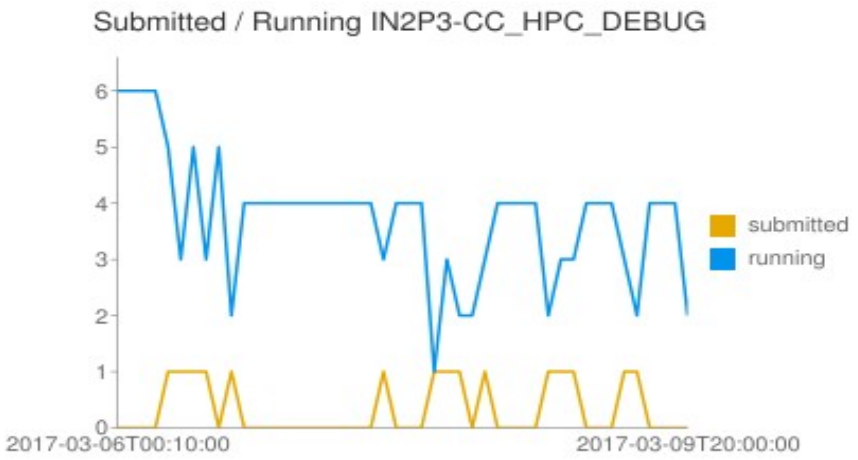
HammerCloud Template: 914 ~4days,

Multiple run, Single core Job , only 2 event, 30min wallclock per job





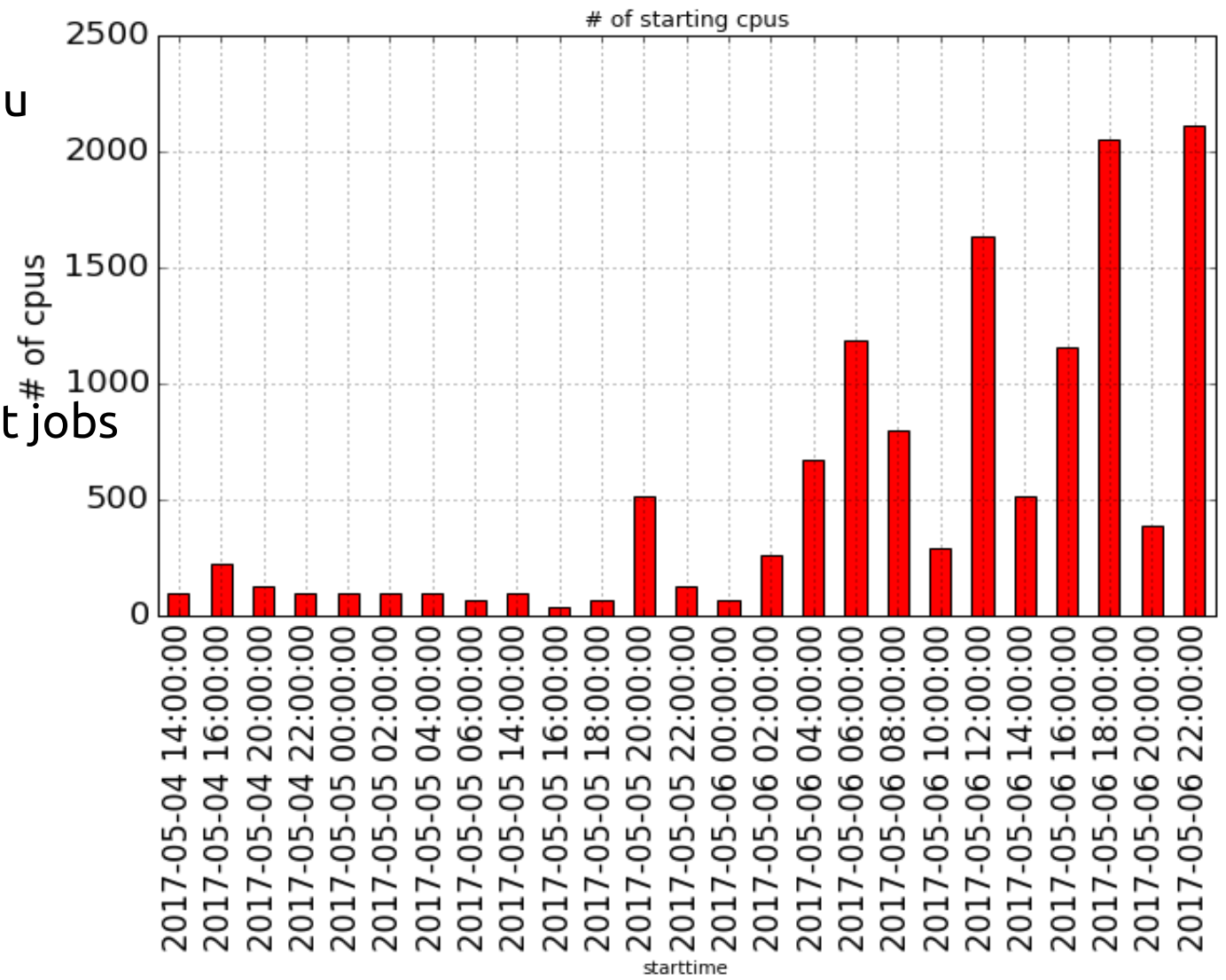




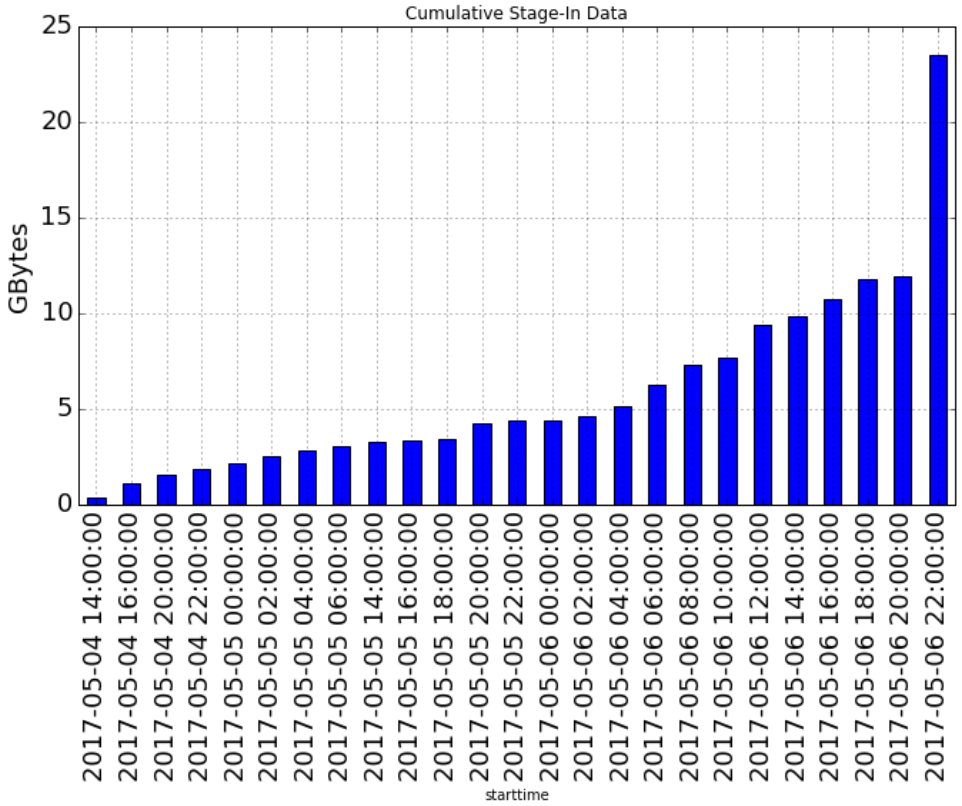
http://hammercloud.cern.ch/hc/app/atlas/test/20103409/

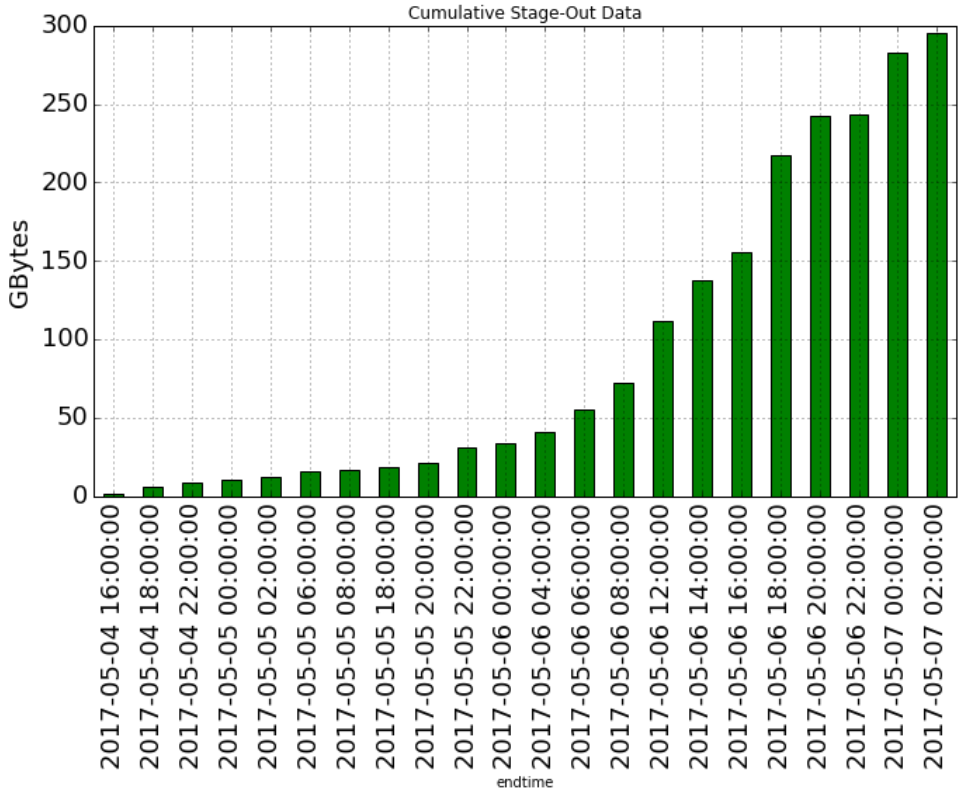
Atlas MC real jobs

- AtlasOffine 21.0.15
  - 400 jobs with 32 cpu
  - Dedicated node
  - ~5Hours
  - Up to 2500 cpu
  - Up to 72 concurrent jobs
  - 4 to 6 Mai 2017
  - ~3 days



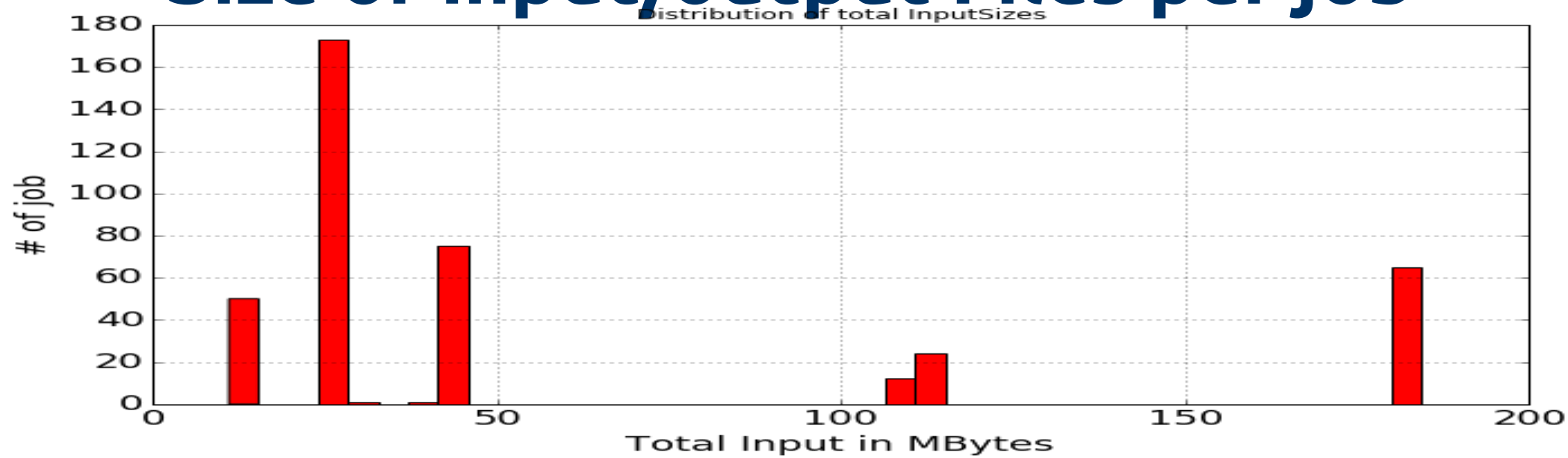
Data stage-in/out

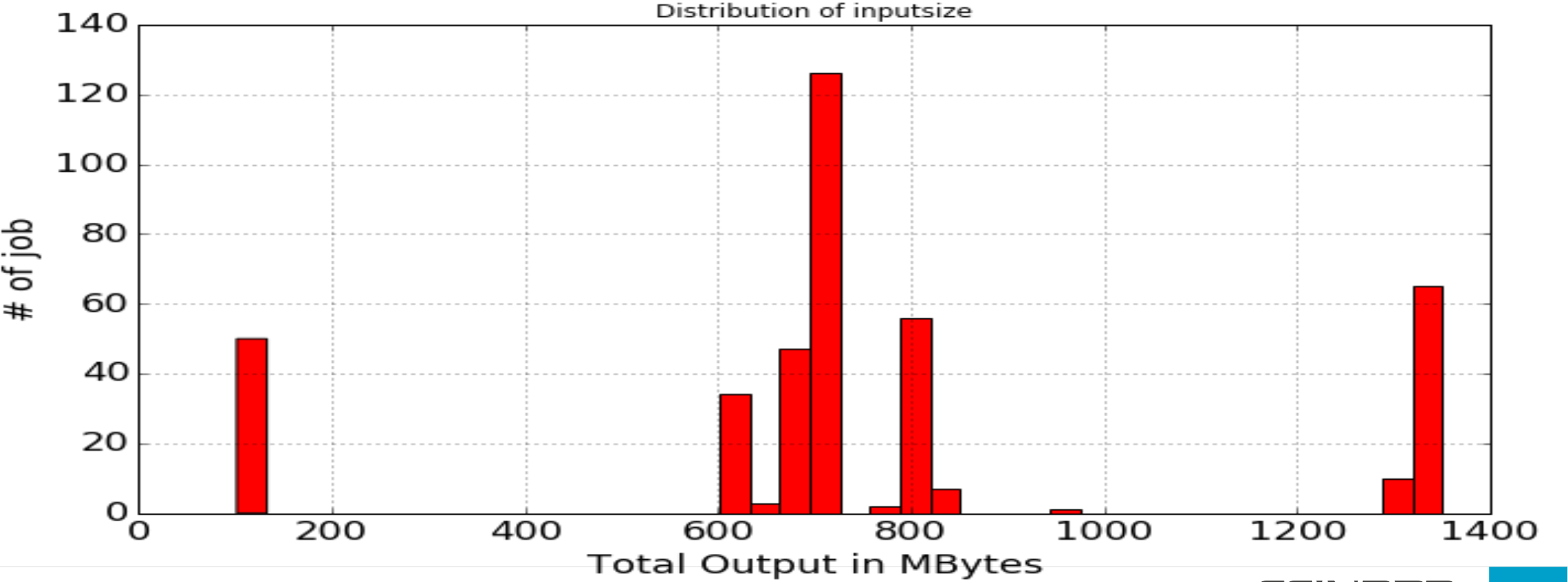




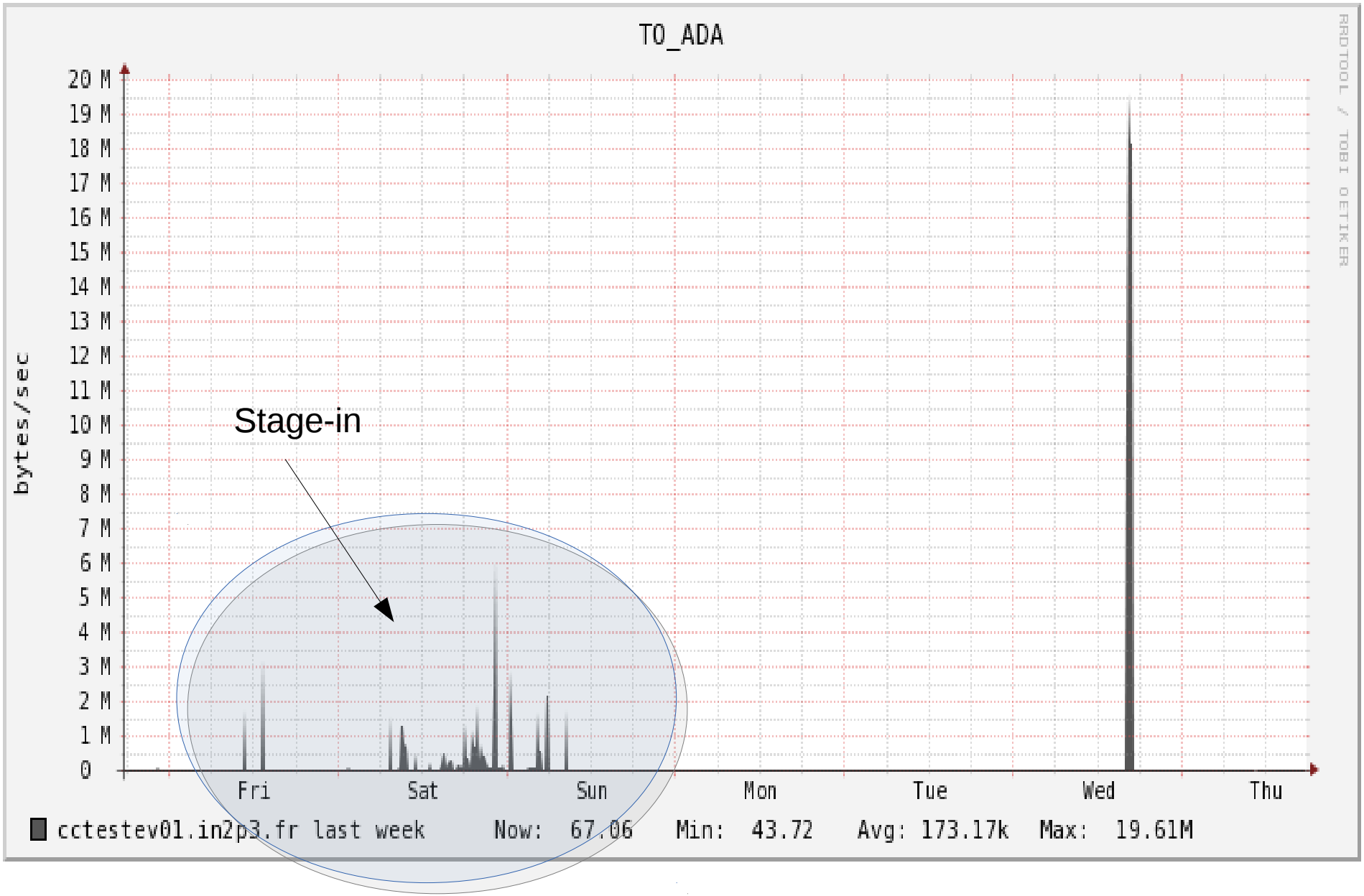
Total 25G stage-in
Total 300GB stage-out

Size of input/output Files per job

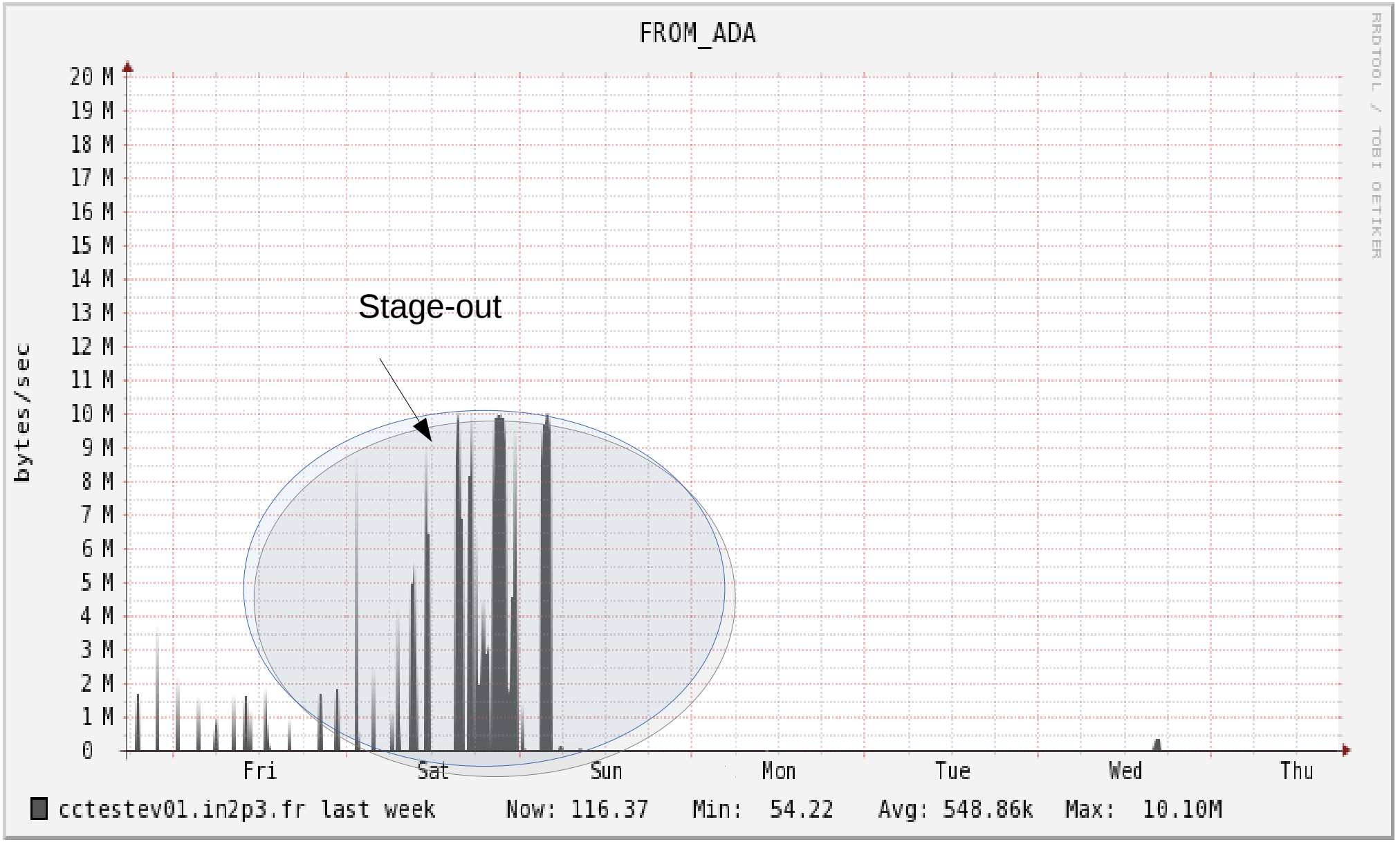




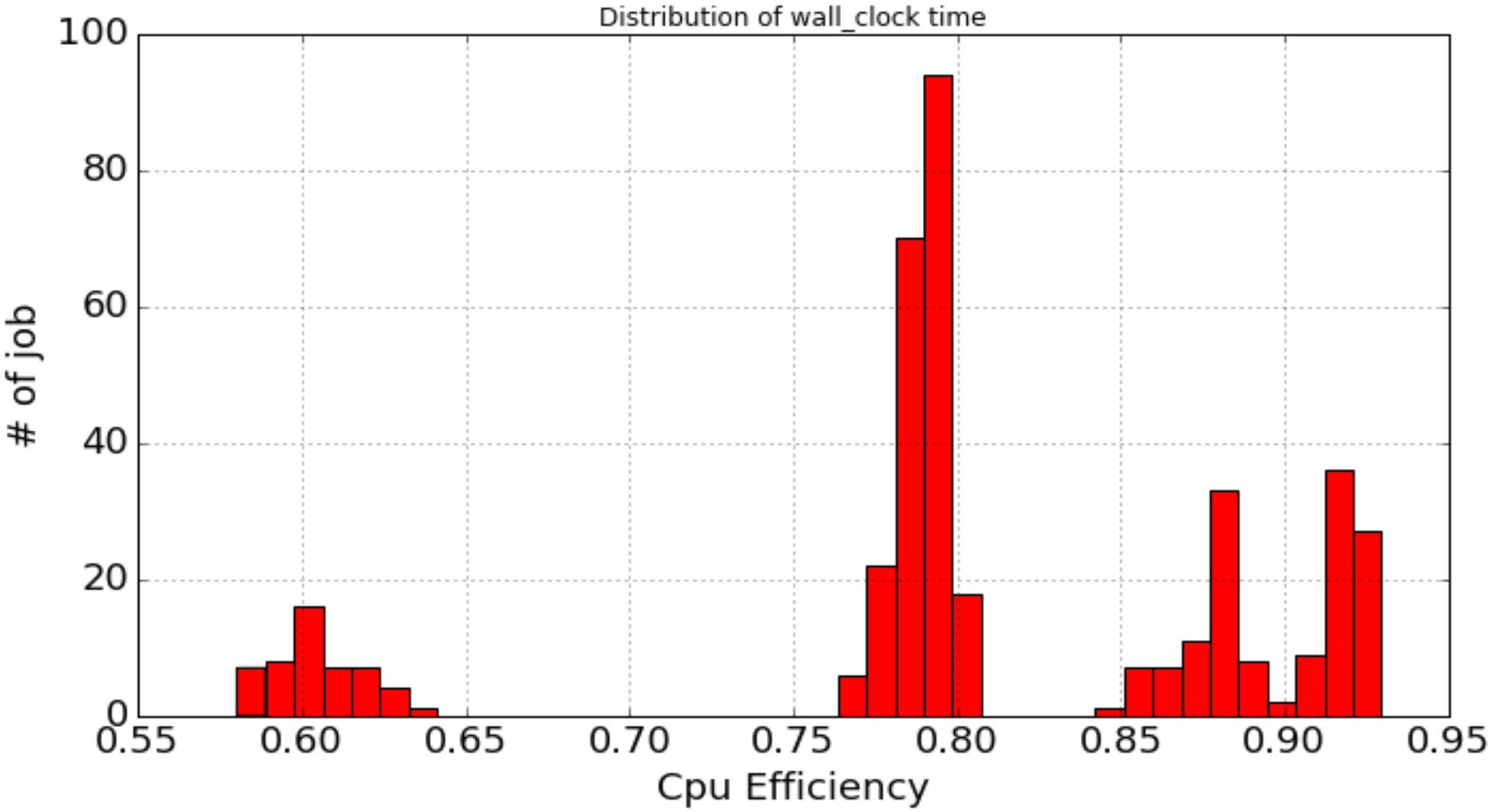
Traffic to_Ada frontend



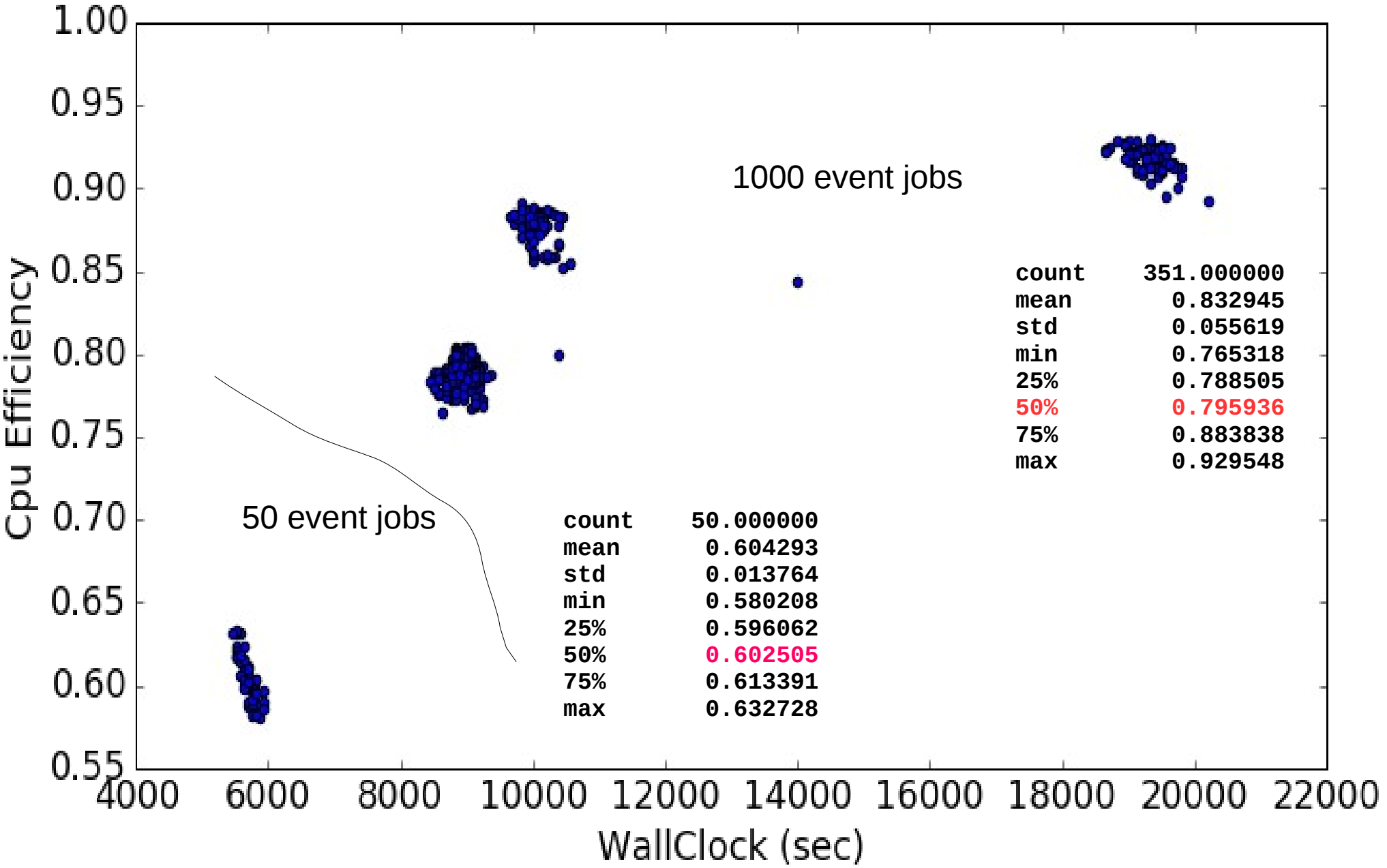
Traffic From_Ada frontend



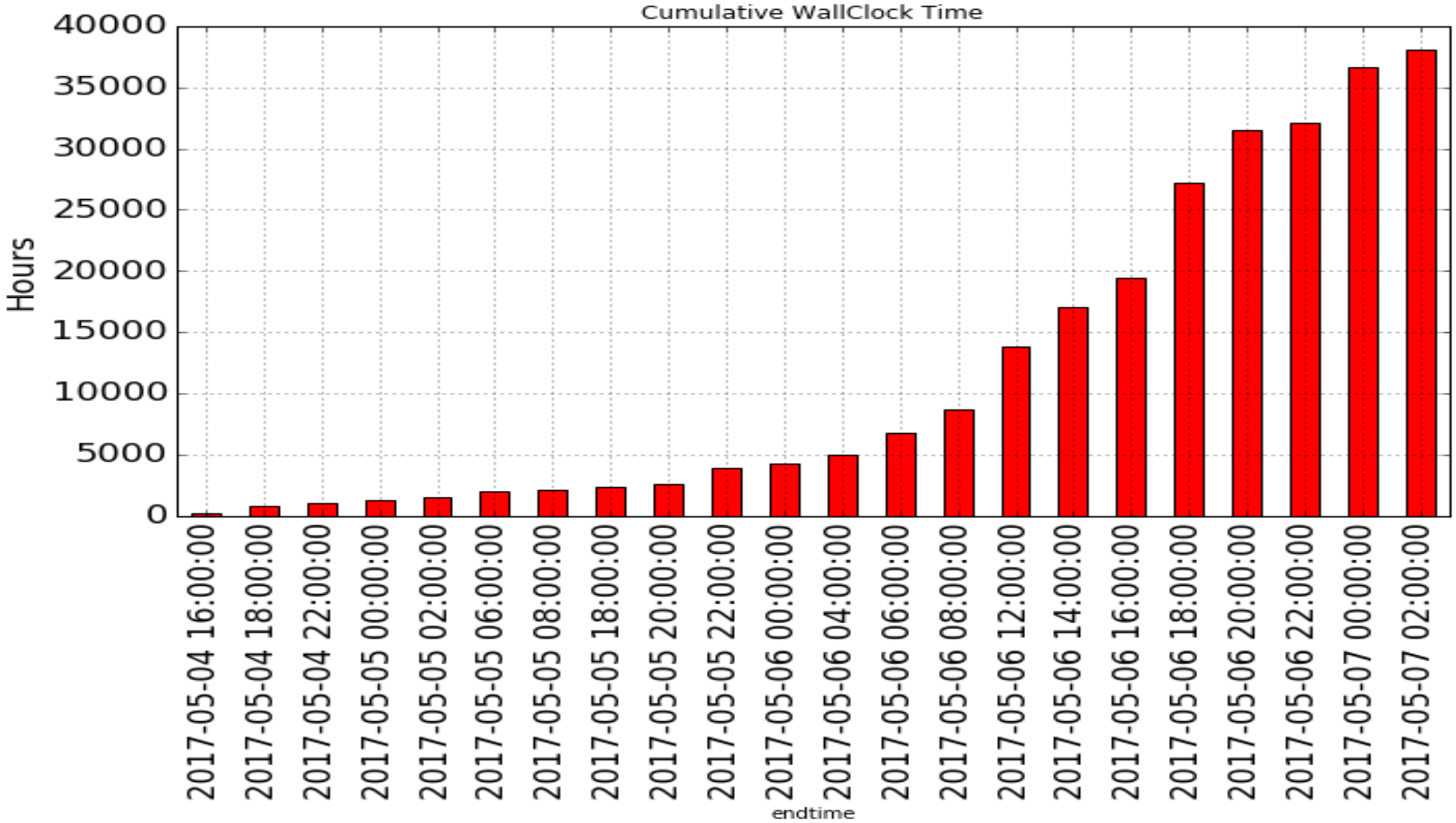
CPU efficiency of 32 core jobs in ADA



Cpu eff vs wallclock



Total Consumed Wallclock time



Conclusion

- ARC-CE stage-in and stage-out to CC SRM storage via sshFS looks that it works
- Ssh wrapper and sshfs looks stable
- The total amount of stage-in and stage-out data for 400 jobs →is too low
- CPU eff. of the job is quite acceptable >80 % for job with wallclock >3-4 Hours
- 1.5 hours wallclock is too low: all job failed in atmt32 class

Other point

- Solved issue with bad report of wall-clock and cpu efficiency
- Solved the issue with issued llq command
  - One general llq + one for the class every 10min
  - llclass and llstatus run ones and cached
- rsync 17k files, 57Gbytes , 2.5 hours , trafic peak to 25 M/sec

LL batchsystem

- **atmt32 is too short** , could we increase to **6h,** from our side we can tune and try to keep the wall-clock time at minimum level...(need.to.be.test) and try to avoid abuse of the resource
- Which is the maximum number of CPUs that we can use ?

- Can we have the accounting record for atlas jobs from LL accounting → Full record + exploitation of the fields
- LL did not report the total # of CPU if we run multi thread jobs ? ...
- Which is the future of ADA x86_64 ?

Next Point

- Clarify the max wallclock and max # of CPU in Ada ?
- Create a kind of «limit » in arc-ce in order to avoid abuse of Ada resources, max number of cpu
- Check the jobs cpu efficiency vs the wallclock time
- Check the jobs cpu between cvmfs and local stored software
- Prepare the EventService mode , check cpu_eff ?
- Check arc-ce cached options

Références

- http://cvmfs.readthedocs.io/en/stable/cpt-hpc.html?highlight=parrot
- http://www.nordugrid.org/arc/
- https://github.com/libfuse/sshfs
- http://ccl.cse.nd.edu/software/manuals/parrot.html
- The ATLAS ARC backend to HPC, S. Haug, M. Hostettler, F. G. Sciacca, M. Weber on behalf of the ATLAS collaboration, (CHEP2015),Journal of Physics: Conference Series 664 (2015) 062057,Link
- Bringing ATLAS production to HPC resources - use case with the Hydra supercomputer of the Max Planck Society, A Kennedy, S Kluth, L Mazzaferro and Rodney Walker on behalf of the ATLAS Collaboration, (CHEP2015), Journal of Physics: Conference Series 664 (2015) 092019, Link
- ARC Control Tower: A flexible generic distributed job management framework, J.K. Nilsen1, D. Cameron and A. Filipčič, (Chep 2015), Journal of Physics: Conference Series 664 (2015) 062042,Link

BACKUP

StakeHolders