



Lightweight computing resources

WLCG workshop, 19th June 2017

Maarten Litmaath, CERN

v1.0

How to enable computing

- Services that currently are or may be needed to enable computing at a grid site:
 - Computing Element
 - Batch system
 - Cloud setups
 - Authorization system
 - Info system
 - Accounting
 - CVMFS Squid
 - Monitoring

T2 vs. T3 sites

- T3 sites have not signed the WLCG MoU
 - Typically dedicated to a single experiment → can take advantage of shortcuts
- T2 sites have rules that apply
 - Availability / Reliability targets
 - Accounting into EGI / OSG / WLCG repository
 - EGI: presence in the info system for the Ops VO
 - Security regulations
 - Mandatory OS and MW updates and upgrades
 - Isolation
 - Traceability
 - Security tests and challenges
- Evolution is possible
 - Some rules could be adjusted
 - The infrastructure machinery can evolve

Lightweight sites – classic view

- How might sites provide resources with less effort?
 - Storage → see Data Management session on Tue
 - Computing → followup in Ops Coordination
- Here we are mostly concerned with EGI sites
 - US-ATLAS and US-CMS projects: see CHEP talk
- Site responses to a questionnaire show the potential benefits of shared repositories
 - OpenStack images
 - Pre-built services, pre-configured where possible
 - Docker containers
 - Ditto
 - Puppet modules
 - For site-specific configuration

Lightweight sites – alternative view

- CE + batch system *not strictly needed*
- Cloud VMs or containers could be sufficient
- They can be managed e.g. with Vac or Vcycle
 - Several GridPP sites are doing that already
 - All 4 experiments are covered
 - The resources are properly accounted
- They can directly receive work from an experiment's central task queue
- Or they can rather join a regional or global *HTCondor pool* to which an experiment submits work
 - Proof of concept used by GridPP sites for ALICE
 - Cf. the CMS global GlideinWMS pool → scalable to O(100k)

Distributed site operations model

- A site needs to provide resources at an agreed QoS level
- HW needs to be administered by the site
- Other admin operations could be done by a remote, possibly distributed team of experts
- Site resources within a region could be amalgamated (and hidden) by a regional *virtual* HTCondor batch system
 - VMs/containers of willing sites may join the pool directly
 - CEs and batch systems of other sites can be addressed through Condor-G
 - The virtual site exposes an HTCondor CE interface through which customers submit jobs *to the region*
 - HTCondor then routes the jobs according to fair-share etc.

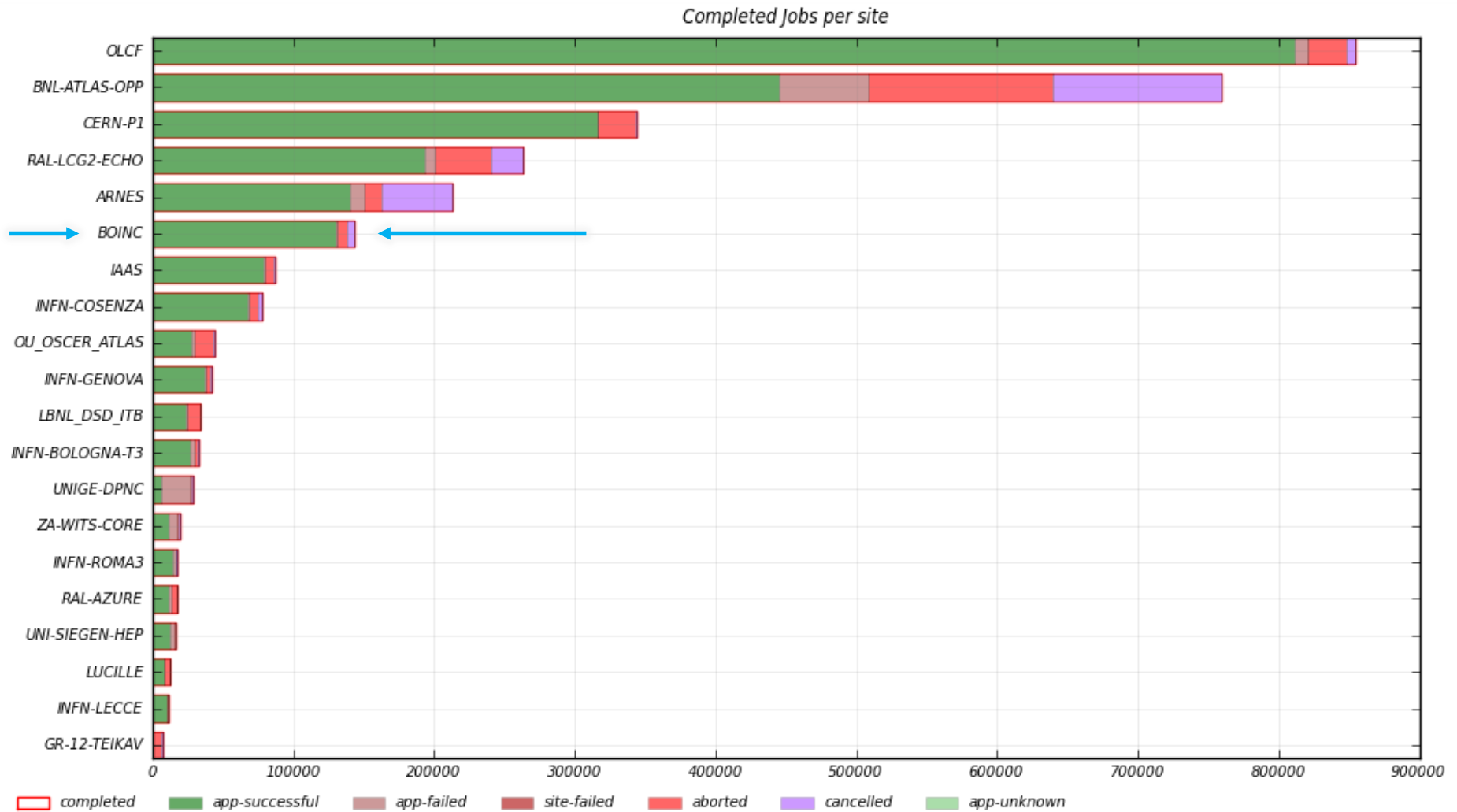
Volunteer computing ...

- The LHC@home project coordinates volunteer computing activities across the experiments
- ATLAS have benefited from 1-2% extra resources for simulation workloads
- See this recent talk by David Cameron
- It could become a way for a *computing-only lightweight site* to provide its resources
 - The central infrastructure can scale at least for simulation jobs
 - The resources can be properly accounted in APEL

... and lightweight sites

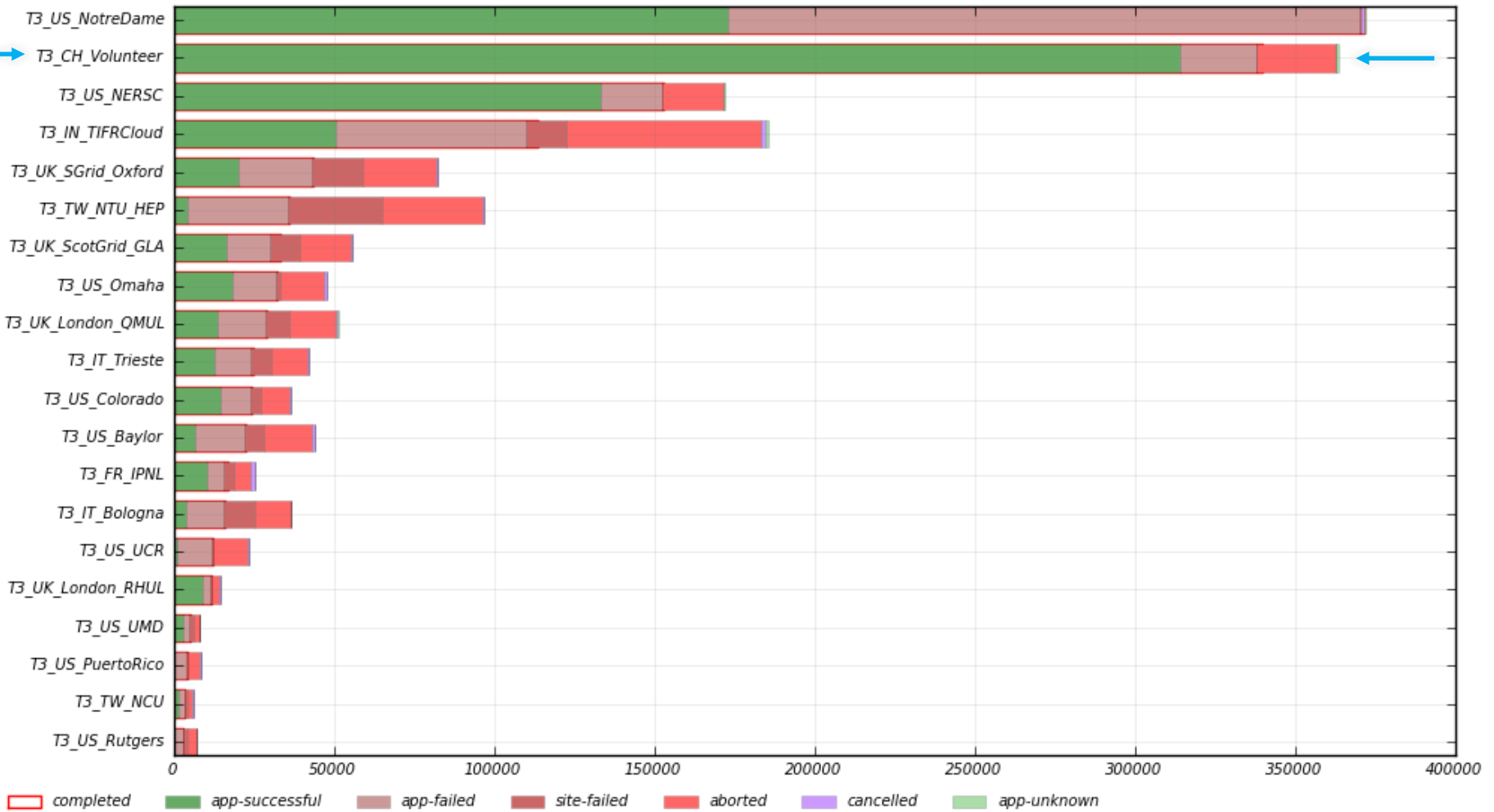
- Real sites can be trusted
 - No need for volunteer CA or data bridge
 - A separate, easier infrastructure would be set up
- BOINC might even coexist with a batch system on the same WN
 - Some ideas to explore...
- Also here HTCondor is used under the hood
 - Standard for experiments and service managers

Recent ATLAS T3 stats

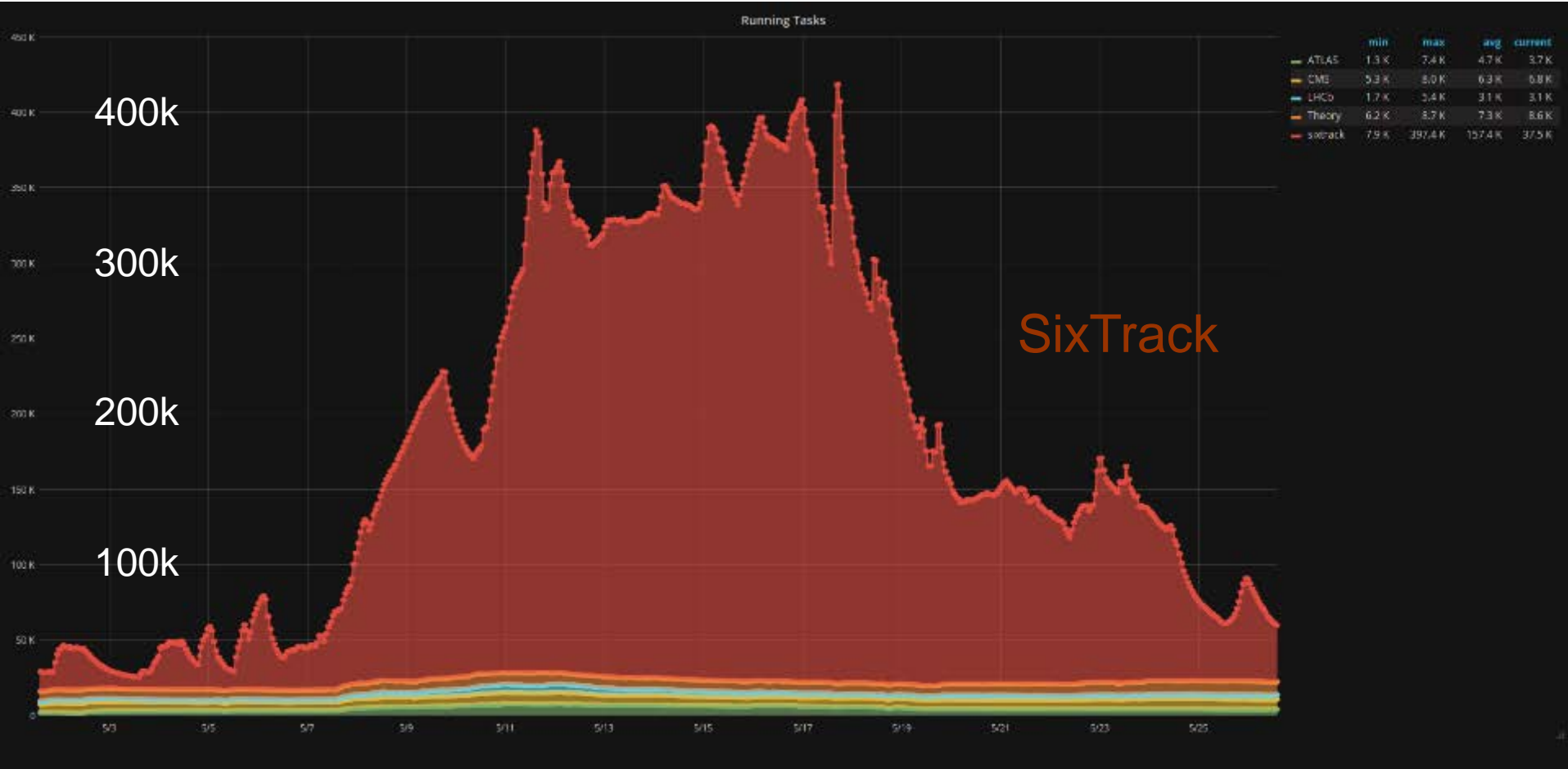


Recent CMS T3 stats

Completed Jobs per site



Volunteer potential



8th BOINC Pentathlon 2017

Computing resource SLAs

- The resources themselves can also be “lightweight”
- Please see this recent talk by Gavin McCance
- Extra computing resources could be made available at a lower QoS than usual
 - Disk server CPU cycles, spot market, HPC backfill, intervention draining, ...
 - Jobs might e.g. get lower IOPS and would typically be pre-emptible
 - Machine-Job Features (MJF) functionality can help smooth the use
- They would have an SLA *between* those of *standard* and *volunteer* resources → a *mid-SLA*