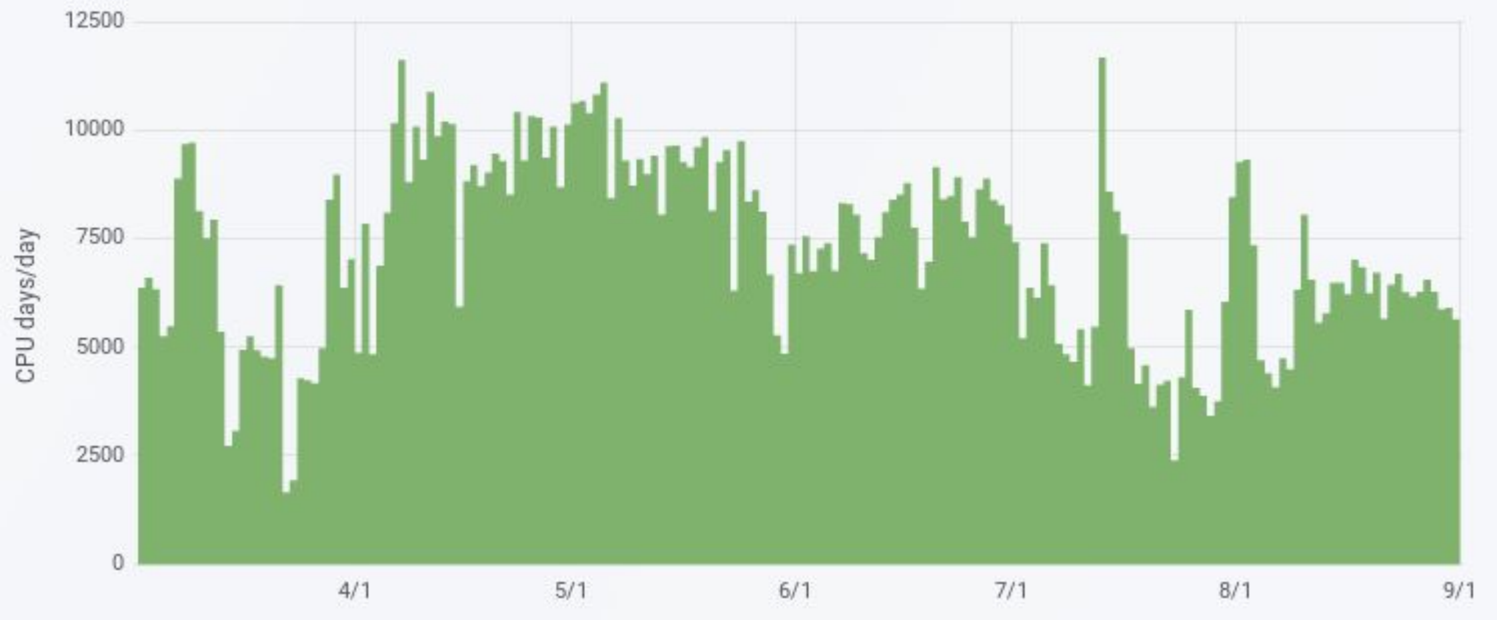
BOINC: Present and Future Resources

David Cameron
ADC mini-TIM, 1 October 2018

Measuring the performance

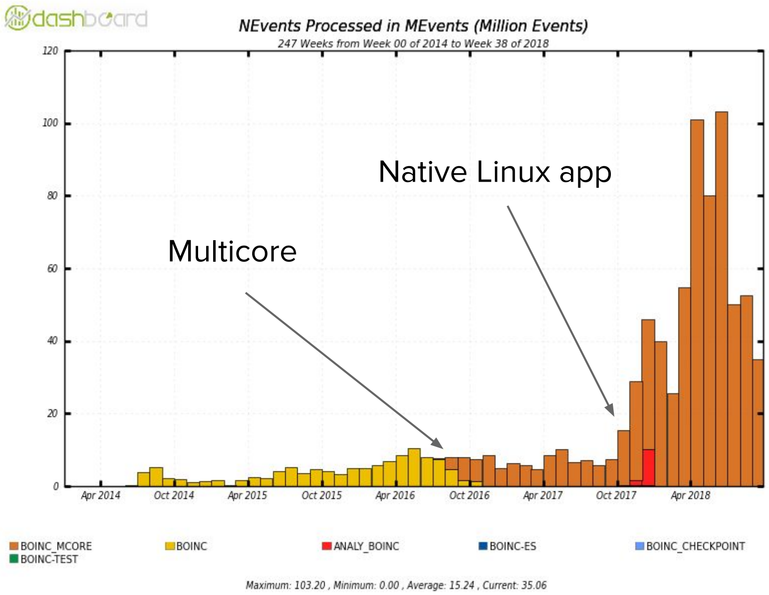
- Wallclock is not so accurate since jobs can be suspended/resumed
- CPU time is measured accurately, but power of processors varies a lot
- Average efficiency (cpu/wall) is 73% compared to 84% for all simulation
- We calculated average HEPSPEC06 to be 10
    - Based on event processing time for one task on a host with known HEPSPEC and average time over all BOINC
- Running slots is a “best guess” since we never know if jobs are actually running once they are sent out
    - Averages ~10k recently

Recent performance - CPU Consumption



CPU days/day March - September 2018
On average 2.5% of entire ATLAS CPU

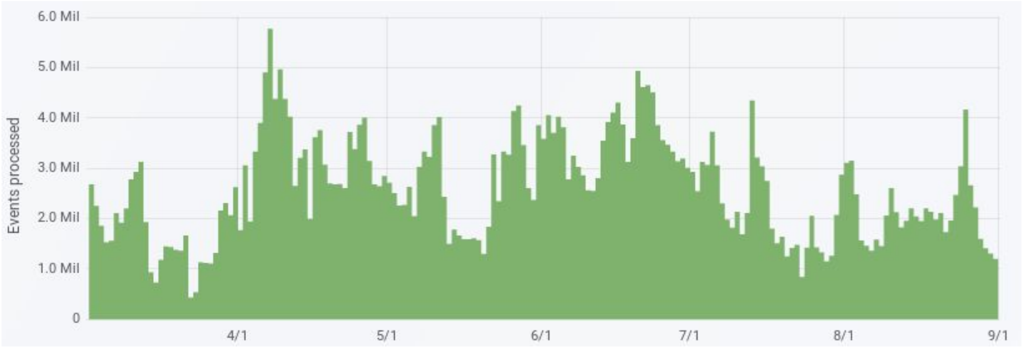
Recent performance - Events Processed



Native Linux app

Multicore

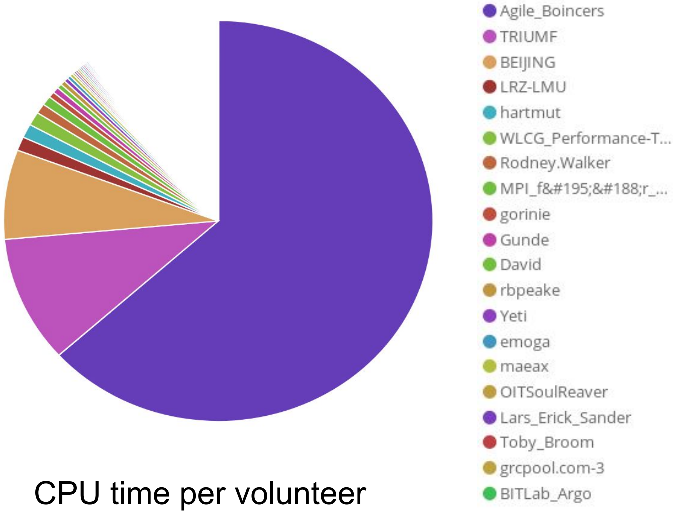
Events processed per month 2014 - now



Events processed per day March - September 2018
On average 4.5% of all simulation
627M events Oct 17 - Oct 18

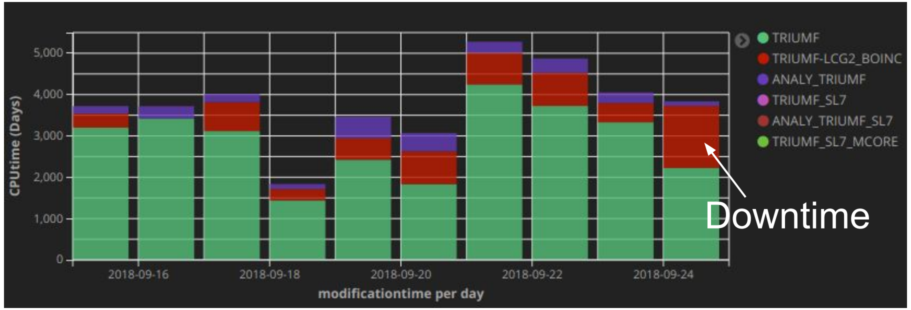
Future prospects

- Traditional volunteer base (@home) is stable but not significantly increasing
- Recent increase in resources is mainly from spare CERN machines and grid backfilling



CPU time per volunteer

- Grid backfilling can increase CPU consumption of a site up to 25%
- For dedicated ATLAS sites with memory to spare should be a no-brainer to install
- ADC service machines? (5000 cores)



Downtime

TRIUMF CPU time for Grid and BOINC

Issues to work on

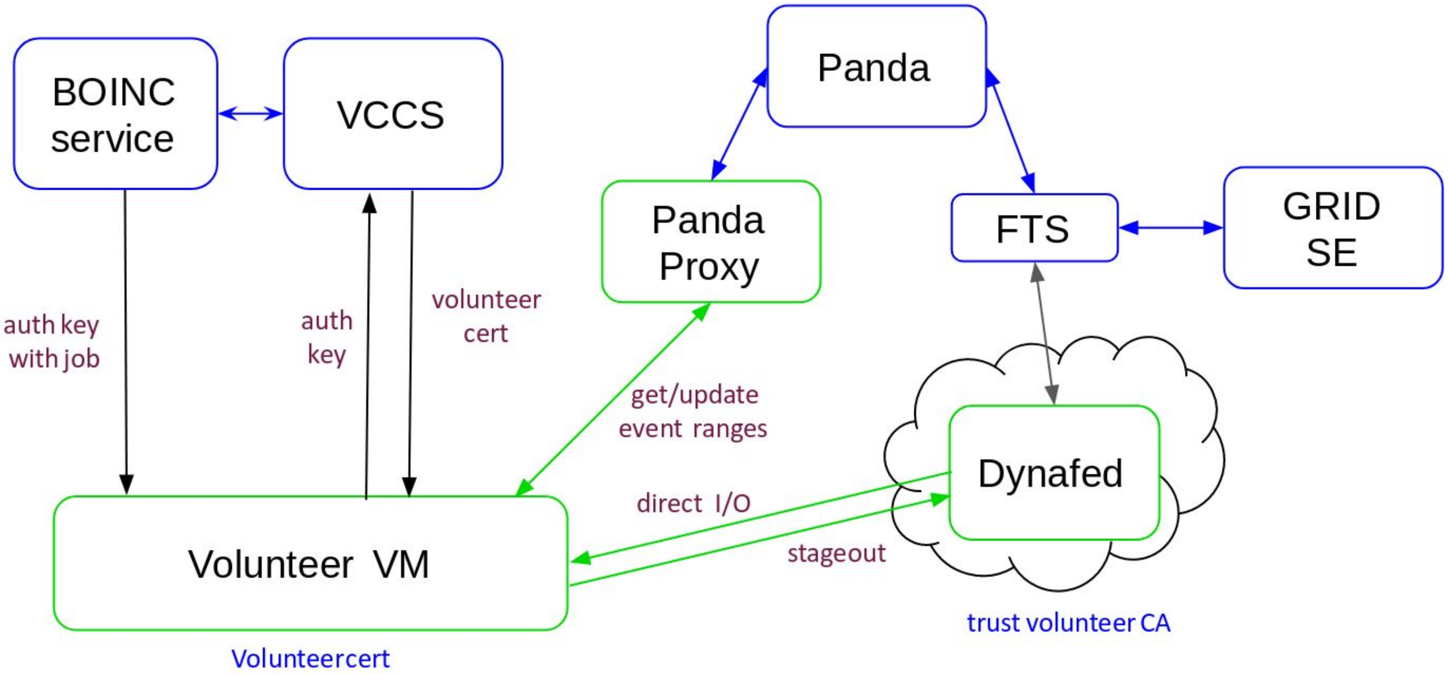
- Accounting for Grid backfilling
- Non-dedicated tasks
    - BOINC is manually assigned tasks (thanks to David South for keeping it continuously full!)
    - Only difference from normal grid jobs is we would like fewer events per job (currently 200)
    - If panda supported dynamic job sizes based on desired no of events or walltime we could run any simulation
- Running normal simul jobs (1000-event, non-dedicated) on dedicated resources, eg CERN spare machines
- Job types
    - Simulation is good for volunteers and backfilling
    - Evgen could be nice for volunteers too (less demanding on disk and memory)
    - Maybe other workloads could run on dedicated machines
- Event service
    - A prototype implementation was designed and tested, built around not giving credentials to volunteers
    - Still too many bleeding-edge technologies to deploy in production
        - Dynafed, pandaproxy, event service itself
    - No real gain expected in terms of throughput - BOINC itself handles preemption well
    - But maybe “normal” ES jobs could run on trusted resources like grid backfill

Backup

References

- https://lhcathome.cern.ch/lhcathome/
- Installing BOINC on grid nodes: https://twiki.cern.ch/twiki/bin/view/AtlasComputing/RunningATLASAtHomeOnACluster
- CHEP talk on grid backfilling: https://indico.cern.ch/event/587955/contributions/2937192/
- Kibana dashboard for grid sites (requires login)

Event service implementation in ATLAS@Home


BOINC service
VCCS
Panda
Panda Proxy
FTS
GRID SE
auth key with job
auth key
volunteer cert
get/update event ranges
Dynafed
direct I/O
stageout
Volunteer VM
Volunteercert
trust volunteer CA