Lightweight sites: BOINC Backfilling

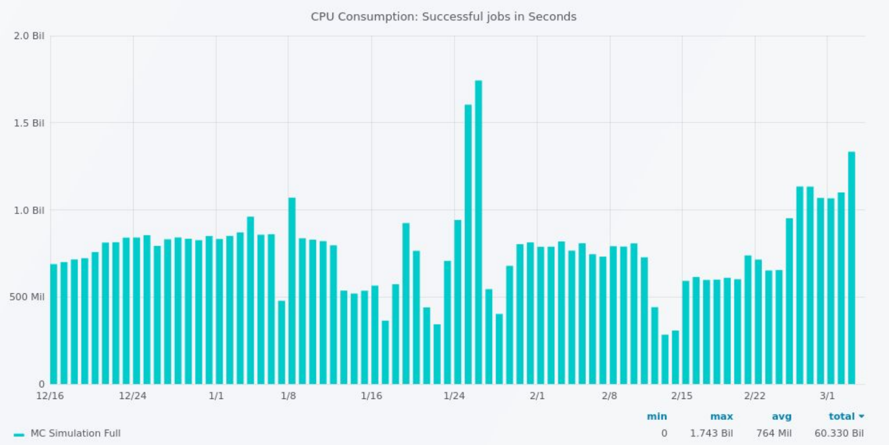
David Cameron (University of Oslo)

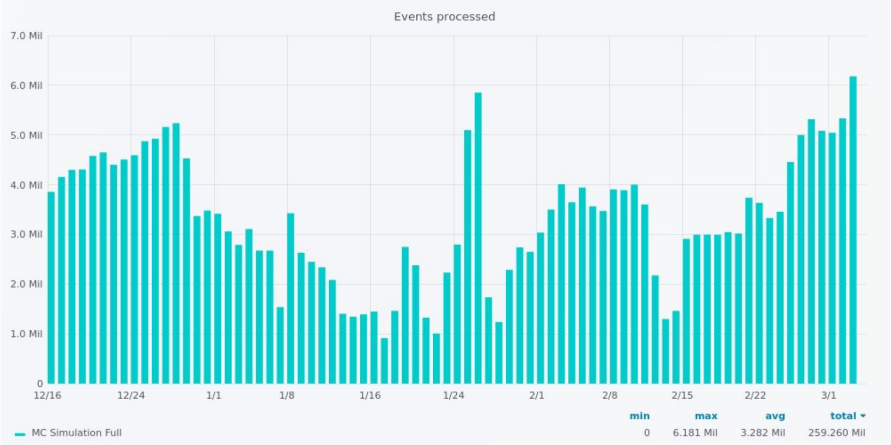
Shaojun Sun (University of Wisconsin-Madison)

ATLAS Sites Jamboree and HPC strategy, CERN, March 7, 2019

BOINC Status

- New Boinc monitoring appears reliably since 16 Dec, 2018
- https://monit-grafana.cern.ch/d/000000696/job-accounting-historical-data -> resource type "cloud_special"





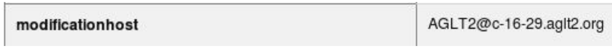
Average 8800 CPU days/day - equivalent of ~10k core grid cluster

Average 3.3M simulation events/day processed

BOINC Site Monitoring

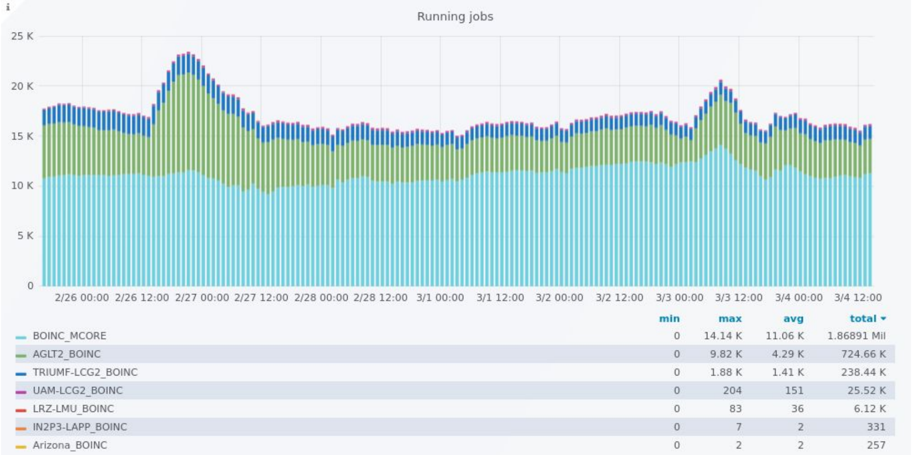
2 weeks ago per-site monitoring was finally implemented

- Site admin creates BOINC account with same name as ATLAS site (eg. AGLT2)
- BOINC wrapper sets job modification host to username@hostname
- MONIT monitoring maps these jobs to site_BOINC



modificationhost	AGLT2@c-16-29.aglt2.org

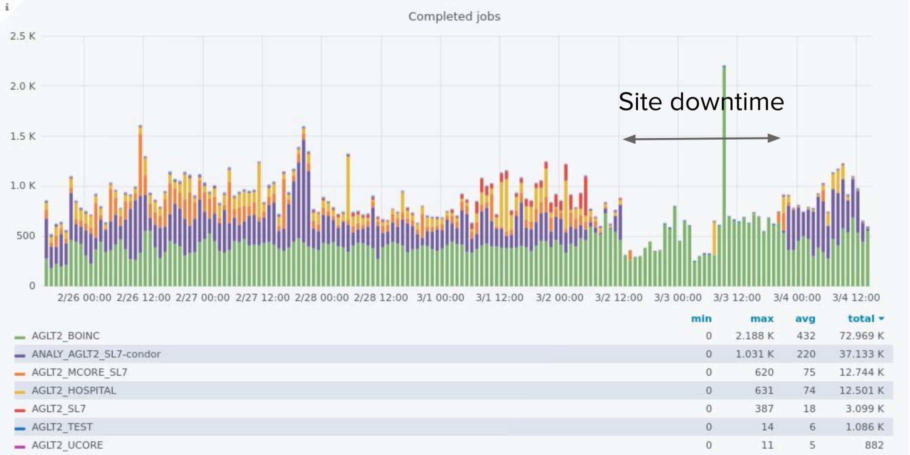
All running BOINC jobs per site



	min	max	avg	total ▾
BOINC_MCORE	0	14.14 K	11.06 K	1.86891 Mil
AGLT2_BOINC	0	9.82 K	4.29 K	724.66 K
TRIUMF-LCG2_BOINC	0	1.88 K	1.41 K	238.44 K
UAM-LCG2_BOINC	0	204	151	25.52 K
LRZ-LMU_BOINC	0	83	36	6.12 K
IN2P3-LAPP_BOINC	0	7	2	331
Arizona_BOINC	0	2	2	257

Link: Select resource type "cloud_special" and group by "**computingsite**"

Completed jobs per queue for AGLT2

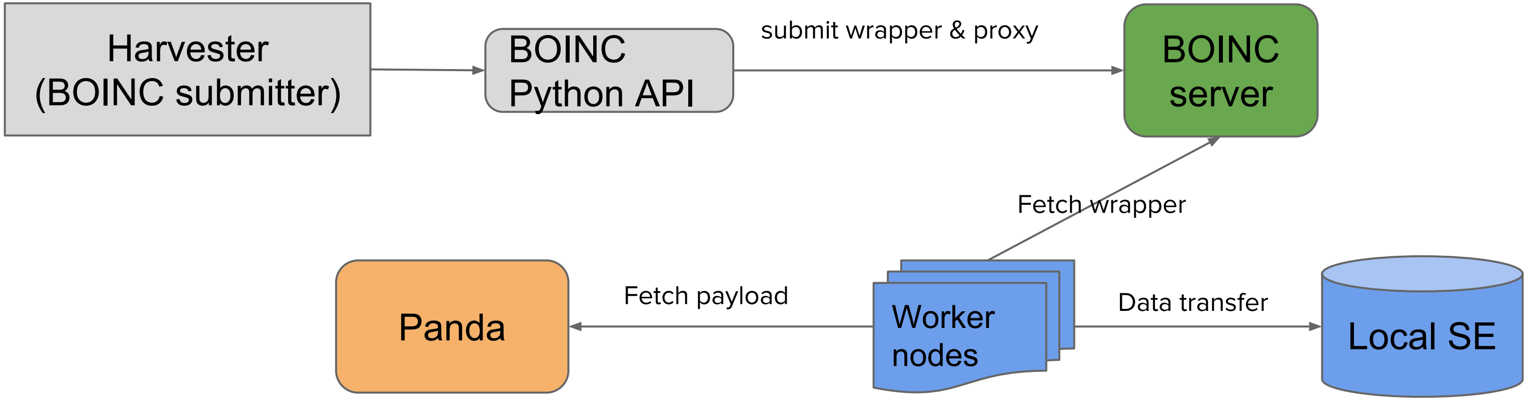


	min	max	avg	total ▾
AGLT2_BOINC	0	2.188 K	432	72.969 K
ANALY_AGLT2_SL7-condor	0	1.031 K	220	37.133 K
AGLT2_MCORE_SL7	0	620	75	12.744 K
AGLT2_HOSPITAL	0	631	74	12.501 K
AGLT2_SL7	0	387	18	3.099 K
AGLT2_TEST	0	14	6	1.086 K
AGLT2_UCORE	0	11	5	882

Link: Select one "Site", all resource types and group by "**computingsite**"

Harvester-BOINC Architecture for Trusted Resources

- Motivations:
  - Current architecture based on ARC-CE was designed around never giving credentials to untrusted resources
  - Allowing trusted resources to directly access grid storage addresses potential data management scaling issues
- Run a private BOINC server for trusted resources
- Harvester submits jobs (pilot wrapper & proxy) to BOINC server via Boinc Python API
- BOINC clients on WN pick up pilot wrappers which pull real payload from Panda server and interact with grid storage



Accomplished tasks

- Set up a PanDA queue called BOINC-BACKFILL.
- Set up a Harvester instance.
- Tested Boinc Remote submission API.
- Implemented Harvester-Boinc submitter, monitor and sweeper.
- Defined a few ES tasks for testing (e.g. 17248200, 17270723).
- Finished some ES jobs.

Harvester Instance Configuration

- Harvester supports various workflows: 1-to-1 and pull, 1-to-1 and push, 1-to-many and push, to name a few.
- We use NoJob & Pull workflow: Harvester submits jobs (pilot wrapper & proxy) without fetching/updating panda jobs; Pilot get jobs from PanDA server
- We are currently using multi-threading mode; can be switched to multi-processing mode with Apache+UWSGI
- We are using SQLite database; can be switched to other database backends, for example, MySQL.
- Thanks to Fa-Hui.

Next steps

- Commission harvester-boinc plugins into production harvester
- Run some real tasks
  - Currently BOINC has manual task submission (short jobs and fixed release)
  - In this new mode any simulation task could run, but event service or short jobs are preferable for backfilling
- Currently BOINC_BACKFILL is attached to CERN DDM endpoints
  - Ideally a job could dynamically change its own configuration to read from any endpoint and write to local storage or nucleus (how to get input data, which is not local?)
  - i.e. BOINC wrapper reassigns an activated job to a different panda queue (possible in panda?)
  - Or, run only event service and upload to CERN OS

How to connect your site

- Twiki entry page: https://twiki.cern.ch/twiki/bin/view/AtlasComputing/VolunteerComputing
- Create a BOINC account with the same name as your ATLAS site (as defined in AGIS)
- Required software on worker node:
  - CVMFS
  - Singularity (if not SLC6)
  - BOINC client ← this should be the only thing you don't already have
- Either install BOINC from eg EPEL or use simple install script (no root required)

  -
    ```
    wget http://atlasathome.cern.ch/Atlas-test/scripts/install_atlasathome.sh
    sh install_atlasathome.sh --ProDir /opt/BOINC --uid your_boinc_authenticator
    ```
  - ... and jobs start running
- For Backfill mode, once things are ready you will need to ask for an account on the private BOINC server

BACKUP

Experienced Issues

- StageIn error: BOINC_BACKFILL queue set to TEST
- StageOut error: failed to access Object Store
- Pilot wrapper could not get PanDA jobs due to not enough activated jobs.
    - added skip_scouting in task parameter for testing tasks
    - reduced the numbers of maxWorkers and nQueueLimitWorker