ATLAS WORLD-cloud and networking in PanDA

Fernando H. Barreiro Megino[1], *Kaushik De*[1], *Alessandro Di Girolamo*[2], *Tadashi Maeno*[3], *Rodney Walker*[4] *on behalf of the ATLAS Collaboration*

[1]*University of Texas at Arlington*, [2]European Organization for Nuclear Research, [3]*Brookhaven National Laboratory*, [4]*Ludwig-Maximilians-Universität München*

CHEP 2016 San Francisco

UNIVERSITY OF TEXAS ARLINGTON

PanDA

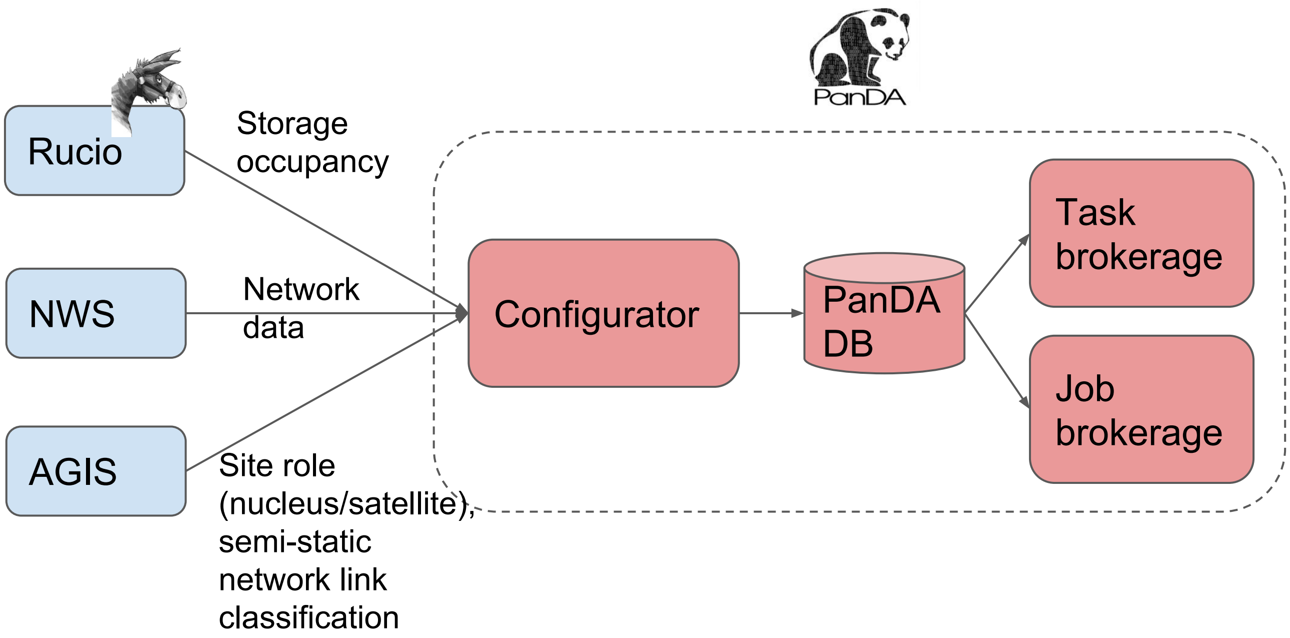
Rationale

- Original ATLAS Computing Model was designed as static clouds (=mostly national or geographical groupings of sites), setting data transfer perimeters
- Hierarchic model with clear distinctions in Tier 1-2-3 level
- The particular policies enforced in the workload management system
  - Output of tasks (=set of jobs) had to be aggregated in the O(10) Tier 1s
  - Tasks have to be inflexibly executed within a static cloud
- This model works, but is getting outdated and has a series of disadvantages
  - WLCG networks have evolved significantly in the last two decades and bandwidth has increased O(1000)
    - Limiting transfers within a cloud is no longer needed
  - Usage of sites is uneven. In particular Tier 2 storage was not optimally exploited
  - Small clouds stuck with large, high priority tasks
  - First try was to allow sites to below to multiple clouds, now we are completely breaking the boundaries

WORLD cloud

- Dynamic model, where tasks are not confined to a cloud. A grouping of processing sites is defined dynamically for each task
- **Task nucleus:**
  - Task brokerage will choose a nucleus for each task based on data locality, queued work and available storage
  - The capability of a site to be a nucleus is defined manually in AGIS (ATLAS Grid Information System). Tier 1s and the bigger Tier 2s are defined as nuclei
  - The output will be aggregated in the task nucleus
- **Task satellites:**
  - Run jobs and ship the output to the nucleus
  - Job brokerage selects satellites for each task, based on usual criteria (e.g. number of jobs and data availability)
  - Satellites are selected across the globe: a network weight will match well connected nuclei and satellites

Configurator



Chooses the nucleus for a task

Chooses satellite sites for the task, where the jobs will run

Network data

- Rucio, FAX, PerfSonar events are collected in the ATLAS analytics platforms [1]
- The Network Weather Service[2] (DDM team) aggregates information from the platform. Per source-destination pair:
    - #files transferred in last hour
    - #files queued
    - Throughput according to FTS, based on 1 week data
    - Throughput according to FAX
    - PerfSonar metrics (latency, packet loss, throughput)
- AGIS also provides semi-static link classification to be used as a backup
- Configurator agent downloads and processes this information every 30 min. Data is cached in a key-value table in PanDA DB
    - Table is extensible for any new metric without modifying the structure

Presentations in this conference:

[1] Ilija Vukotic et al.,"Big Data Analytics Tools as Applied to ATLAS Event Data"

[2] Mario Lassnig et al., "Using machine learning algorithms to forecast network and system load metrics for ATLAS Distributed Computing"

Task brokerage: nucleus selection

- One nucleus is chosen for each task
- Nuclei must fulfil all conditions:
  - Be in active state and be able to execute the workload
  - Have over 5TB free space - the free space includes an estimation of the space to be filled by pending tasks
- The number of output files transferring to the nucleus must be below 2k
- Nuclei compete through a combined weight, based on data locality, total RW (remaining work) and available storage size in the nucleus.

$$RW = (nEvents - nEventsUsed) \times cpuTime$$

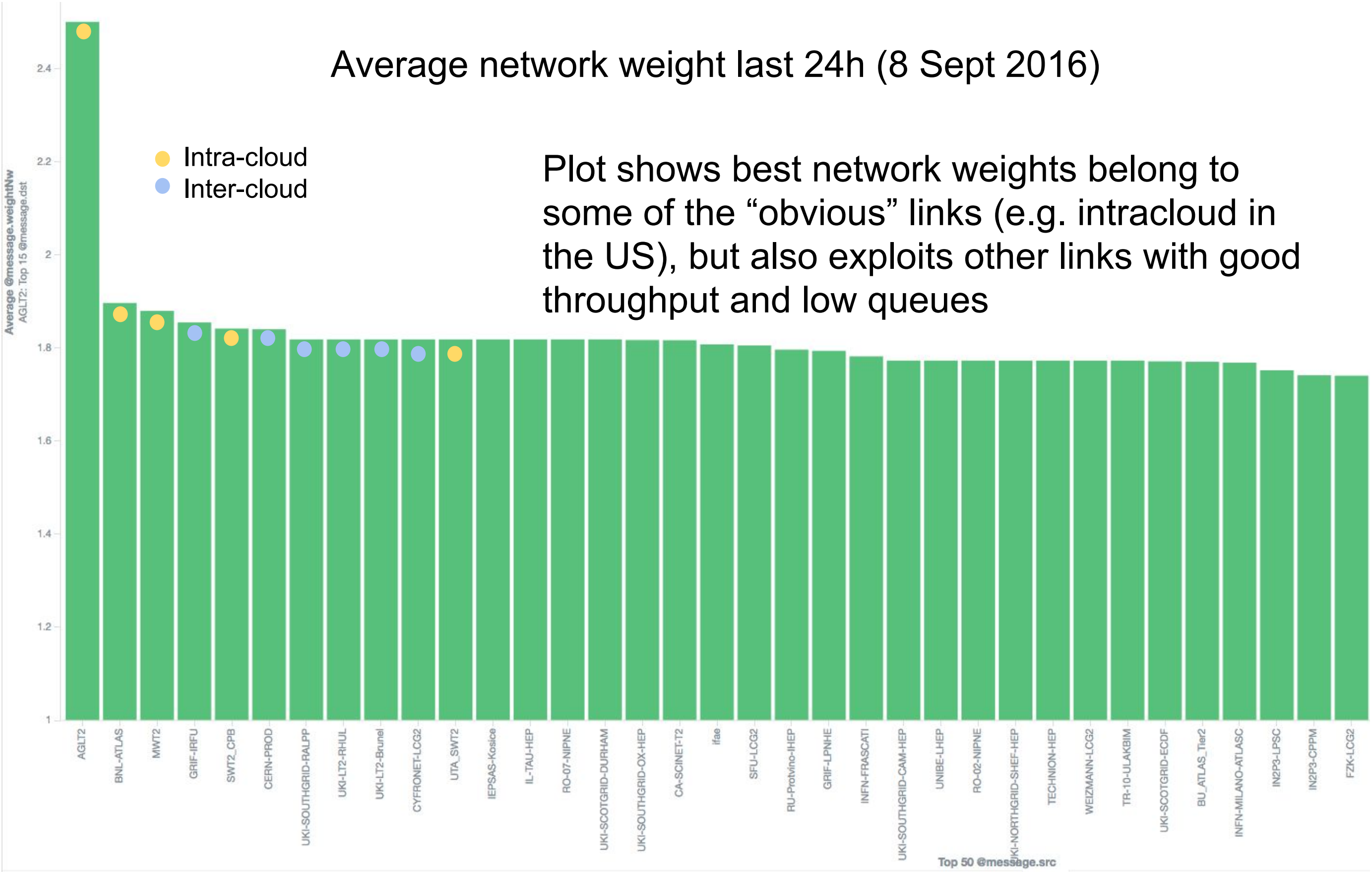
$$weight = \frac{1}{totalRW} \times \frac{availableInputSize}{totalInputSize} \times tapeWeight \times \frac{freeSpace}{totalSpace}$$

Job brokerage: satellite selection

- Up to 10 satellites are selected to execute the task
- The satellites must be able to run the jobs (RAM, walltime, core count, queues, SW releases and other settings)
- Sites must have <150 files in the transfer queue to the nucleus
- Eligible sites compete on a weight basis (#jobs, available & missing data)
- We have added a new network weight, looking for balance between good throughput and queue length of output files
  - Throughput calculated from FTS transfers in the last hour
  - Queued files in Rucio/FTS. In practice the number of files queued for a link is very low; having O(100) files starts to indicate the channel is not keeping up

Example: Top connected sites to Nucleus **AGLT2** (Michigan)

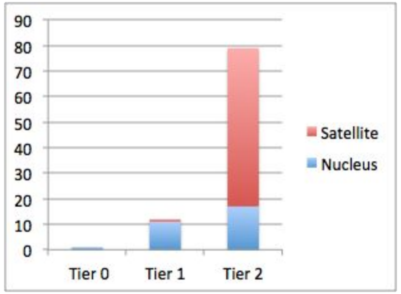
Average network weight last 24h (8 Sept 2016)

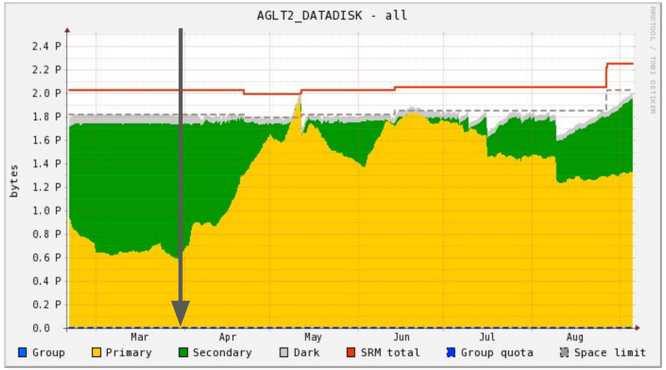


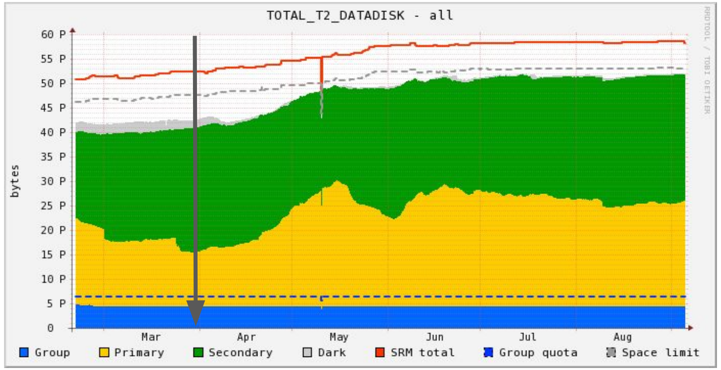
Plot shows best network weights belong to some of the “obvious” links (e.g. intracloud in the US), but also exploits other links with good throughput and low queues

Status: Impact on T2 disk space usage

- WORLD cloud was fully activated end March 2016
- Nuclei being added progressively
  - Currently T1s and ~20% of T2s (tentatively more T2s will be added)
- Extending task output recipients to T2s is starting to have a positive impact on the overall disk usage (more primary data on T2s)

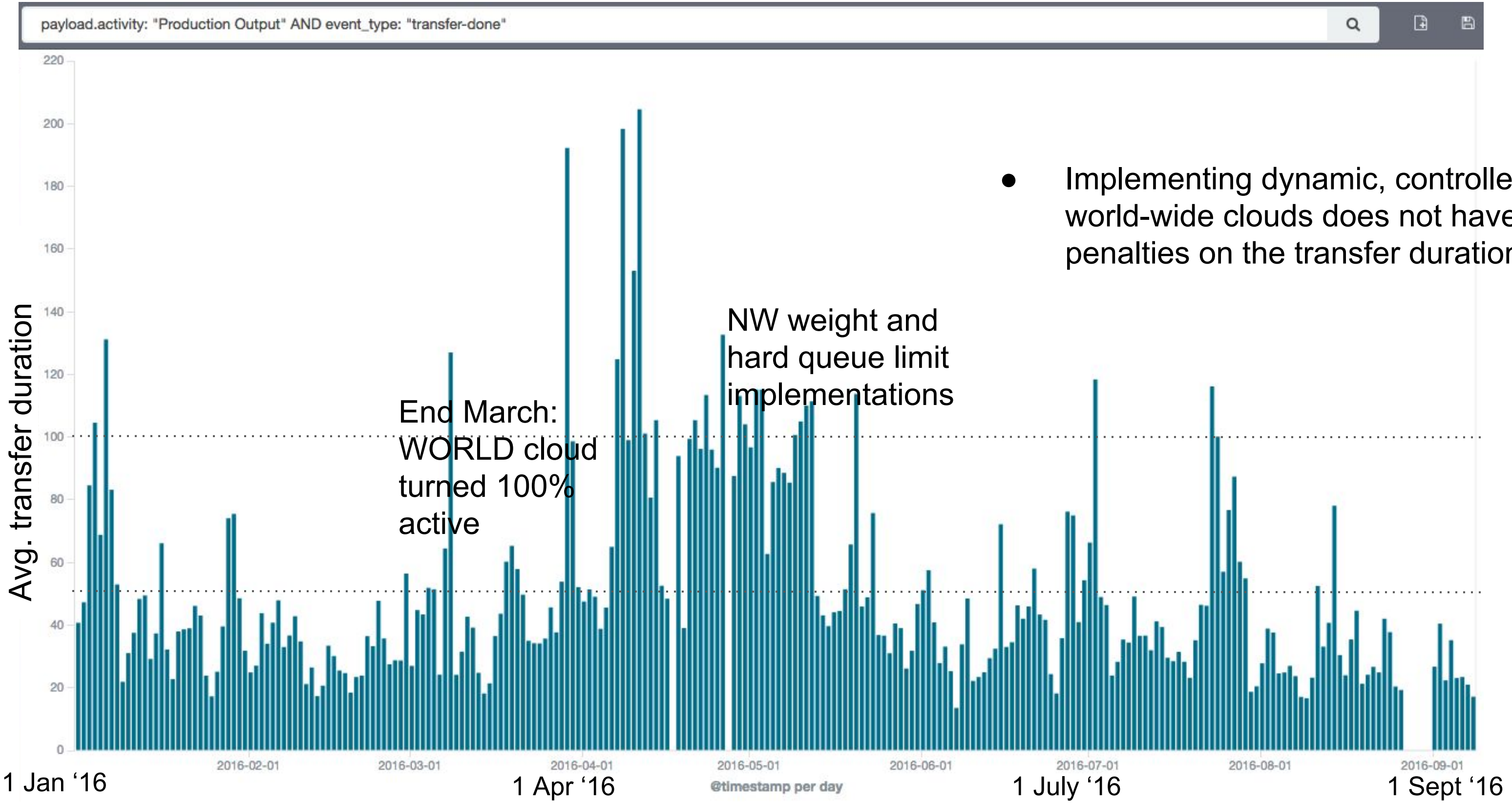






Storage plots from David Cameron's space monitoring: http://adc-ddm-mon.cern.ch/ddmusr01/plots/

Impact on output file transfer duration



payload.activity: "Production Output" AND event_type: "transfer-done"

End March: WORLD cloud turned 100% active

NW weight and hard queue limit implementations

- Implementing dynamic, controlled world-wide clouds does not have penalties on the transfer durations

Observations and future work

- Some sites suffered initially under transfer load during heavy campaigns
  - Not because of inter-cloud transfers, but because they had not enough bandwidth for the Nucleus role
- Hard limit queue controls worked fine to alleviate the issue and deviate the traffic from blocked sites
  - Limits trigger also during unrelated, “accidental” massive transfers
- Further downstream controls could be implemented to e.g. avoid already assigned jobs to run while their nucleus is stuck
  - Pause overloaded nucleus in job brokerage (using a higher queued file threshold to avoid waves)
- Reduced operational effort to rebroker tasks with more resources
- Need to improve analytics data for gridwise analysis
- We have tried to optimize output file transfer, but still need to include some optimization for input file transfers
  - This case needs to be solved together with the DDM team, since it involves further uncertainties (multiple copies, tape staging, etc)