ATLAS Global Shares implementation in PanDA

Fernando Barreiro Megino[1], Kaushik De, Alessandro Di Girolamo, Tadashi Maeno, Rodney Walker for the ATLAS collaboration
[1]*barreiro [at] uta.edu*

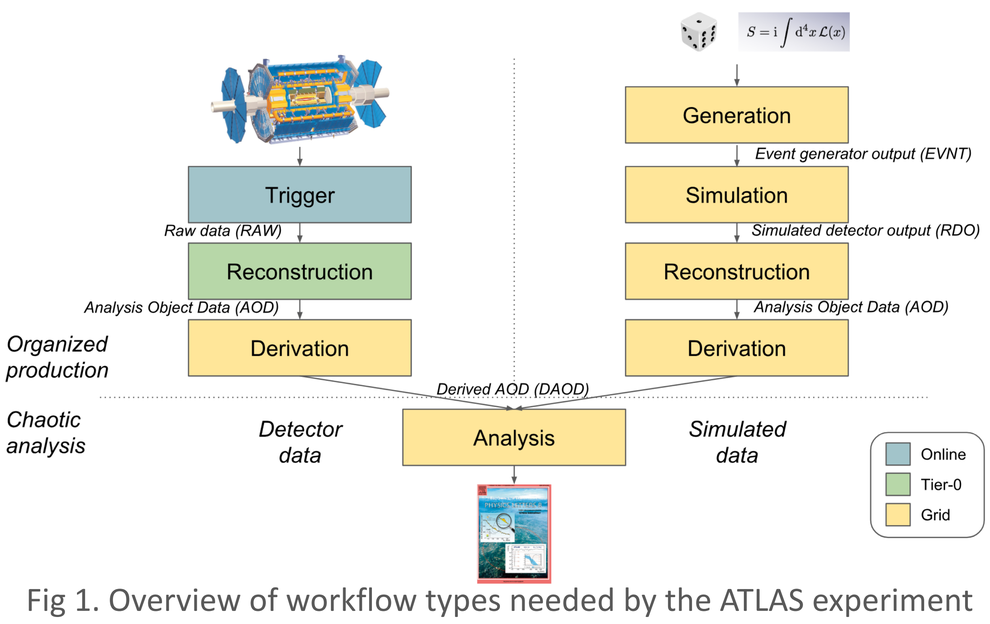

Fig 1. Overview of workflow types needed by the ATLAS experiment

Motivation

- ATLAS Grid resources are shared between different activities with different HW requirements
  - Some workflows are CPU intensive (e.g. MC Generation and Simulation)
  - Other workflows have much higher I/O and memory requirements (e.g. Reconstruction)
- Production is coordinated and planned centrally: campaign based reflecting software versions
- Analysis runs throughout the year, peaking before major conferences
- Physics coordination wants to have control over amount of resources dedicated to each activity for planned delivery of physics results
- The Global Shares project implements such control centrally **in PanDA, the Workflow Management System used in ATLAS**

L1 Share	L2 Share	L3 Share	Actual HS06	Target HS06	HS06 ratio	Queued HS06	Actual share	Target share
Analysis [20.0%] +			196,095.06	923,209.24	21.24 %	350,902.74	4.25 %	20.00%
Express [3.0%] +			24,374.67	138,481.39	17.60 %	7,135.35	0.53 %	3.00%
Production [75.0%] +			4,391,315.29	3,462,034.64	126.84 %	29,524,645.25	95.13 %	75.00%
	Derivations [20.0%]		677,411.29	685,551.41	98.81 %	2,094,590.11	14.68 %	14.85%
		Data Derivations [50.0%] +	90,364.67	342,775.71	26.36 %	84,352.62	1.96 %	7.43%
		MC Derivations [50.0%] +	587,046.63	342,775.71	171.26 %	2,010,237.50	12.72 %	7.43%
	Event Index [1.0%] +		103.04	34,277.57	0.30 %	418.11	0.00 %	0.74%
	HLT Reprocessing [3.0%] +		59,615.16	102,832.71	57.97 %	292,533.13	1.29 %	2.23%
	MC evgen [17.0%]		1,024,243.81	582,718.70	175.77 %	1,619,422.82	22.19 %	12.62%
		MC 16 evgen [80.0%] +	899,553.12	466,174.96	192.96 %	1,595,376.64	19.49 %	10.10%
		MC Other evgen [20.0%] +	124,690.69	116,543.74	106.99 %	24,046.18	2.70 %	2.52%
	MC root [6.0%]		342,600.21	205,665.42	166.58 %	1,479,252.33	7.42 %	4.46%
		MC 16 [80.0%] +	341,539.37	164,532.34	207.58 %	1,476,849.94	7.40 %	3.56%
		MC Other [20.0%] +	1,060.85	41,133.08	2.58 %	2,402.39	0.02 %	0.89%
	MC simul [36.0%]		2,266,699.26	1,233,992.55	183.69 %	9,280,848.58	49.10 %	26.73%
		MC 16 simul [80.0%] +	2,266,699.26	987,194.04	229.61 %	9,280,848.58	49.10 %	21.39%
		MC Other simul [20.0%]	0.00	246,798.51	---	0.00	---	5.35%
	Overlay [2.0%]		0.00	68,555.14	---	0.00	---	1.49%
	Reprocessing [10.0%]		368.31	342,775.71	0.11 %	202,202.73	0.01 %	7.43%
		Heavy Ion [20.0%]	0.00	68,555.14	---	0.00	---	1.49%
		Reprocessing default [80.0%] +	368.31	274,220.57	0.13 %	202,202.73	0.01 %	5.94%

Fig. 2.Snapshot of currently defined shares and their actual, target and queued computing power volumes

Global Shares definitions and basic implementation in PanDA

- **Nestable** shares applied **globally** (i.e. across all Grid sites)
  - Unoccupied shares are split between siblings
- Shares are measured in **currently** running HEPSpec06
- A mapping table assigns tasks to their global share at creation time
- PanDA will sort the jobs queued at each site by Global Share needs and assign the highest priority job to the next available slot
- This requires having jobs of all shares available: they need to be generated by parallel Workqueues

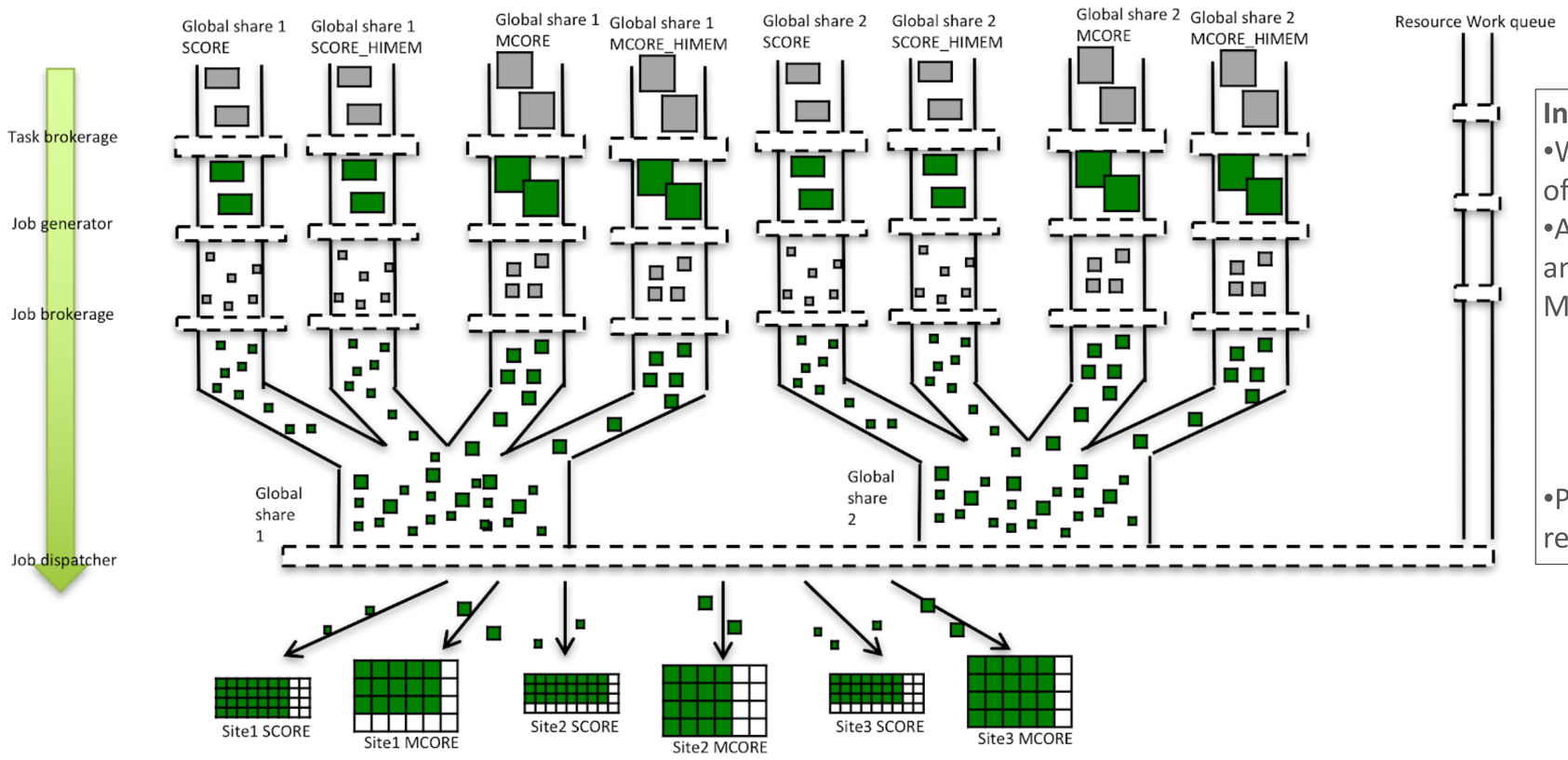

Fig. 3. Schematic showing the dynamic generation of Workqueues

Internal restructuring of Workqueues

- Workqueues are parallel, internal pipes for processing of tasks and jobs (brokerage, job generation, etc.)
- A Workqueue is generated dynamically for each share and resource type (e.g. single SCORE or multicore MCORE slots)
  - Guarantee they don't block each other and jobs are available for each share
  - Workqueues can be tuned individually, e.g. throttle and boost limits
- Possibility of adding Workqueues manually for special resources, e.g. for a large HPC center

Unified queues (being rolled out)

- Previous computing models established separate job queues for single core and multi core slot jobs at each site
  - These queues would compete against each other for resources
  - Each site could establish its own policies and partitions –static or dynamic- on how to divide their batch into the different types
- Nowadays central orchestrators are very well capable of managing the Grid globally
  - Global shares and local partitions don't play along very well
- For sites without local partitions, Unified queues mix jobs of different resource types. Requests to the batch system are then made by Global Shares priorities (push with ARC Control Tower, pull with Harvester), trying to dynamically influence the single to multi core ratio

Conclusions

- One of the major wishes for Physics Coordination is the ability to decide the amount of resources assigned to each activity
- Under the Global Shares project, internal components of the PanDA architecture have been re-engineered and have improved significantly the control over the resources
- We are rolling out Unified queues for influencing dynamically the single to multi core ratio at sites
- We have focused on the control of production jobs so far, analysis yet to be included
- There are corner cases where brokerage does not play along