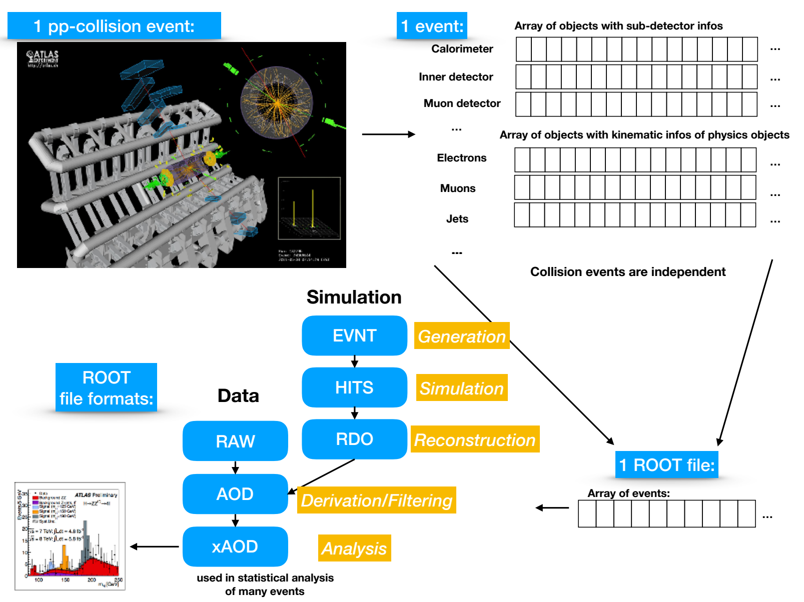
ATLAS Grid Workflow Performance Optimization

Johannes Elmsheuser with A. Di Girolamo, A. Filipcic, A. Limosani, M. Schulz, D. Smith, A. Sciaba, A. Valassi on behalf of the ATLAS collaboration and the workflow performance group

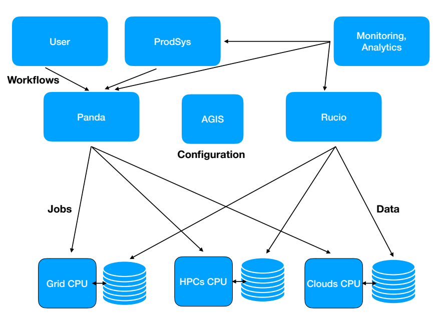
10 July 2018
CHEP 2018, Sofia

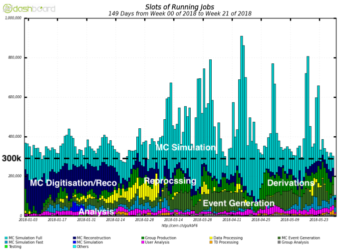

1 pp-collision event:
1 event:
Array of objects with sub-detector infos
Calorimeter
Inner detector
Muon detector
...
Array of objects with kinematic infos of physics objects
Electrons
Muons
Jets
...
Collision events are independent
Simulation
EVNT
Generation
HITS
Simulation
RDO
Reconstruction
ROOT
file formats:
Data
RAW
AOD
Derivation/Filtering
xAOD
Analysis
used in statistical analysis
of many events
1 ROOT file:
Array of events:
ATLAS Preliminary

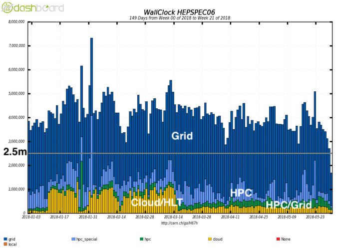
ATLAS DISTRIBUTED COMPUTING OVERVIEW

The ATLAS distributed computing system is centered around:

- Workflow management system PanDA and data management system Rucio
- Diverse resources: WLCG grid sites, Tier0, HPCs, Boinc, Cloud
- Many workflows, users, running jobs, 350 PB data on disk and tape







Workflow performance group

- About 2 years ago, started informal regular meetings of ATLAS computing, software and CERN IT experts
- Discussions about how to optimise different workflows w.r.t resource usage and data throughput
- First improve process monitoring tools
- Build on and contribute to emerging ATLAS analytics platform for high statistics analysis
- Selected workflow performance studies over time

Workflow examples, metrics and typical costs

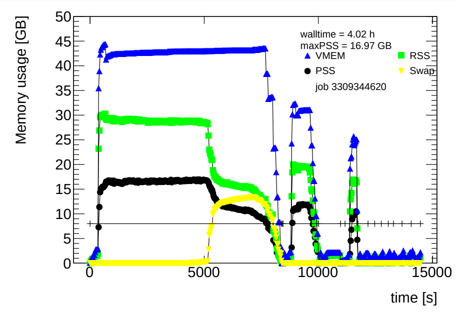
- Ideally: fast turn-around with high events/s - but every workflow has some **cost**
- Costs of workflows: **CPU, memory, disk** and **network I/O**
- Examples loosely categorised by:

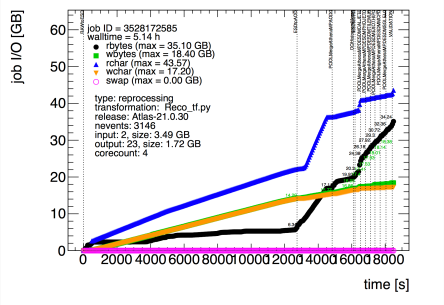
Category	Workflow	Time/evt. [s]	Evt. size [MB]	CPU/Walltime [%]
CPU heavy	MC simulation	30-600	1	80-95
CPU + I/O	MC digitisation/reco	10-40	0.1-0.5	50-80
	data reco			
I/O heavy	derivations	0.1-10	0.1-0.5	30-80
	analysis			

- **Memory:** Fit into $\approx$ 2 GB/CPU core grid slot (can vary)
- **Network:** usually not directly specified, since input files replicated to sites in advance and then locally read, only remote conditions DB access

$\rightarrow$ Optimisation dependent on workflow type

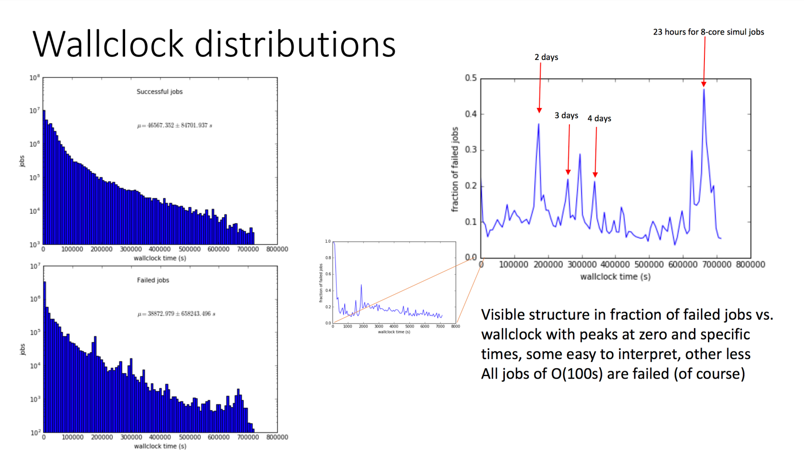
Tools: MemoryMonitor - Profile Memory and I/O ATLAS jobs



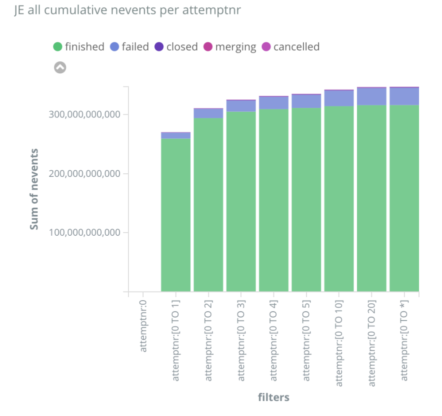


- Every ATLAS PanDA job is instrumented with MemoryMonitor - Forked version available as prmon (Process Monitor)
- Detailed info in job log tarball - summary information collected to PanDA monitor and for **Analytics** in elasticsearch/kibana
- Process Network I/O more difficult: prototype available

- Detailed study of Panda job error codes from last year
- All production payloads show similar wallclock inefficiencies



Workflow job attempts

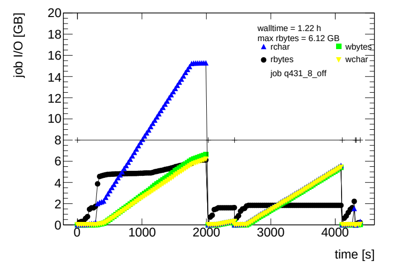


- All PanDA production jobs in Feb-April 2018 - cumulative time/events spent in job attempts
- All PanDA jobs are retried in case of job failures
- Different `processingtype` (Simulation, Reconstruction, etc.) have different maximum number of retry attempts
- → Smart and reliable error detection necessary or efficient retries

Avoid extra merging step: SharedWriter in AthenaMP

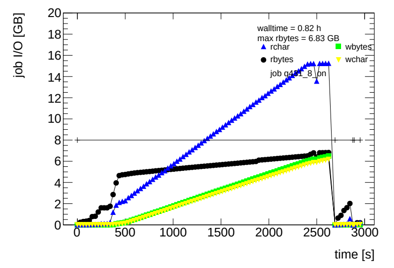
8-core AthenaMP data reconstruction

separate output file merging



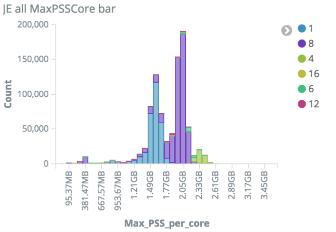
8-core AthenaMP data reconstruction

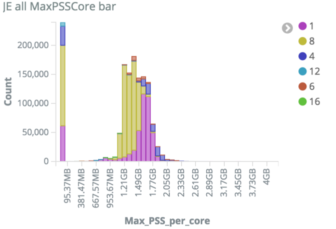
including SharedWriter output merging

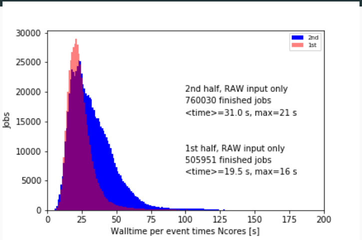


- AthenaMP requires output file merging in separate process or even separate job where output files need to be shipped around
- SharedWriter uses a shared memory process to merge the outputs of the separate AthenaMP processes in one output
- Memory savings, shorter overall walltime, less errors
- In default usage in derivation production to avoid error prone merging

Data17 reprocessing, AthenaMP fork-after-first-event



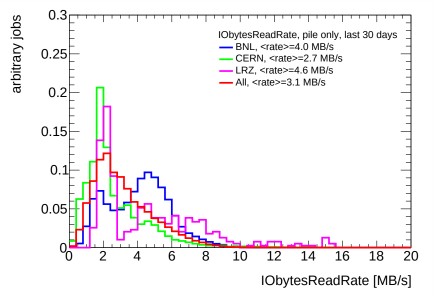


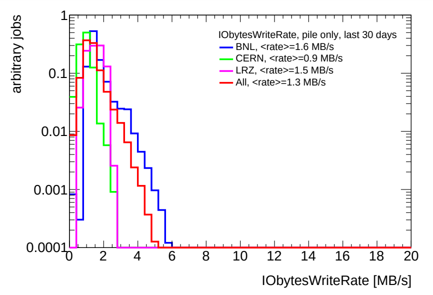


- Lots of conditions and geometry info loaded in first event
  → large memory savings due to memory sharing when forking into AthenaMP subprocesses after instead before first event
- Jobs don’t have to be forced into 2 GB/core slots
  → significant lower job failure rates and overall walltime saving
- Higher walltime/event*cores for 2nd half due to higher $< \mu >$

	10^9 events good	10^9 events good&fail	walltime good [a]	walltime good&fail [a]	walltime % failed
1st, RAW	2.180	2.580	1667	2019	17.4
2nd, RAW	3.047	3.208	3094	3325	6.4

Workflow disk I/O





- MC Digitisation+/Reconstruction : broad site dependent distributions for read rates 1-10 MB/s
- Write rates 1 MB/s
- Using I/O monitoring info for job brokering to avoid ”weaker” sites or to identify software bugs

Further improvements under development

In the works, under testing or R&D:

- Process network monitoring - promising prototype, but need to be properly integrated into jobs
- Presented at last ACAT: AthenaMP CheckPoint-and-Restart: Prototype for Boinc/ATLAS@Home or HPCs, but requires many checkpoint images for different releases
- Build one big static library for Geant4
  $\rightarrow$ demostrated to work in older release, work in progress newer CMake based build infrastructure
- AutoFDO in simulation for: execution speed improvements
- Pile-up pre-mixing in MC

Conclusions

- Established ≈2 years ago working group for ATLAS workflow performance
- Essential: all PanDA jobs are instrumented lightweight process monitoring
- PanDA job information collected in powerful analytics platform
- Identified several bottlenecks and addresses with improved workflows
- Valuable input for the WLCG cost performance working group

BACKUP

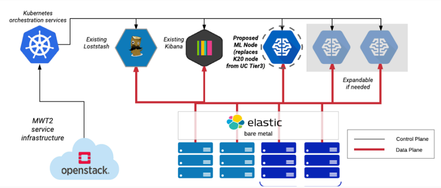
Tools: Analytics

CERN

- Big data processing tools
- Visualisation tools
- Data Sources, including:
  - file transfer data, dataset usage (Rucio)
  - job information (PanDA)
  - xAOD access information
- CERN IT provides the infrastructure for monitoring & analytics by the ATLAS distributed computing team (ADC)

Analytics Platform (ATLAS Midwest Tier2 Center @ University of Chicago):

- Additional data sources (network data from WLCG/OSG, CPU benchmarks, ?)
- Elasticsearch cluster indexed for analytics data
- Jupyter for analytics and GPUs for ML



Following slides show different studies using the elasticsearch/kibana PanDA job informations to find workflow optimizations