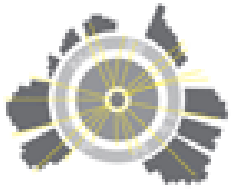
Technical Performance in ATLAS workflows

Antonio Limosani

University of Sydney / ARC CoEPP / CERN

ATLAS SW Technical Interchange Meeting
08.06.2016

THE UNIVERSITY OF SYDNEY



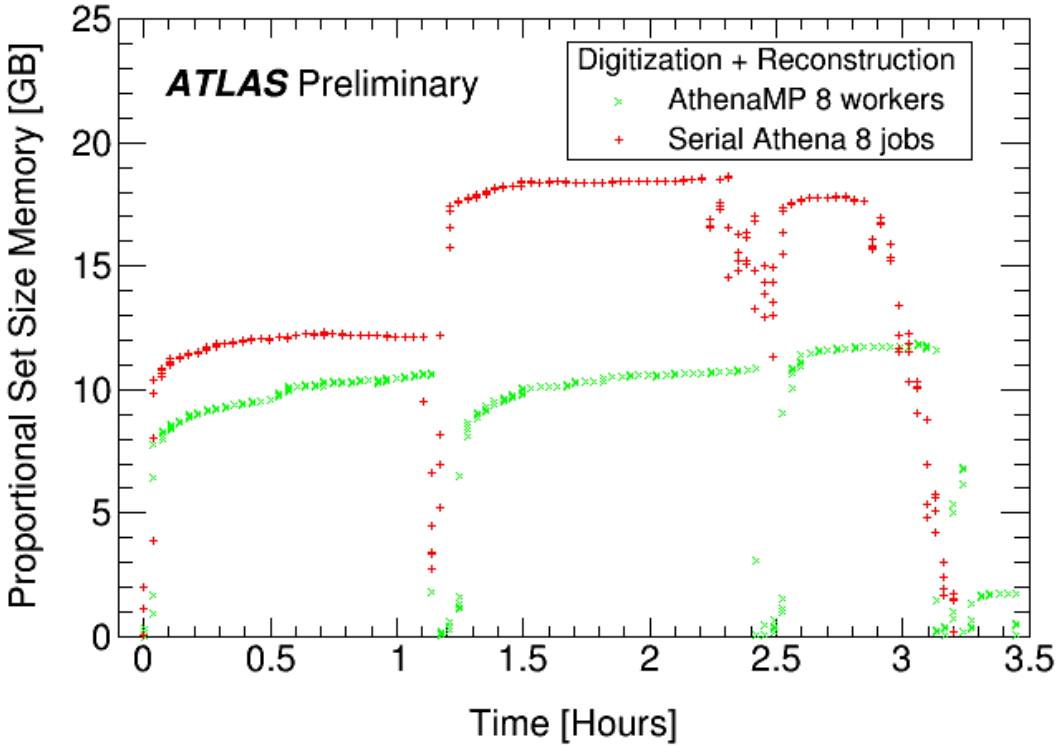
CoEPP
ARC Centre of Excellence for Particle Physics at the Terascale

Workflows : AthenaMP

Step	Time per event (sec)	PSS/RSS(GB)	Release
EVNT to HITS	180	1.7	19.2
HITS to RDO (4-core)	32	6.1/8.4	20.7
RDO to RDOTrigger (4-core)	10	6.7/12.8	20.7
RAW to ESD (4-core)	13	7.2/11.9	20.7
ESD to AOD (4-core)	0.3	5.1/7.9	20.7
AOD to DxAOD MC15ExotJetM	0.4	2.4	20.7
AOD to DxAOD Data15ExotJetM	0.15	1.7	20.7
Physics Main $\langle\mu\rangle \approx 30$ (4-core)			
BS to ESD (4-core)	11	11.0/18.3	20.7
ESD to AOD	0.9	7.7/11.7	20.7

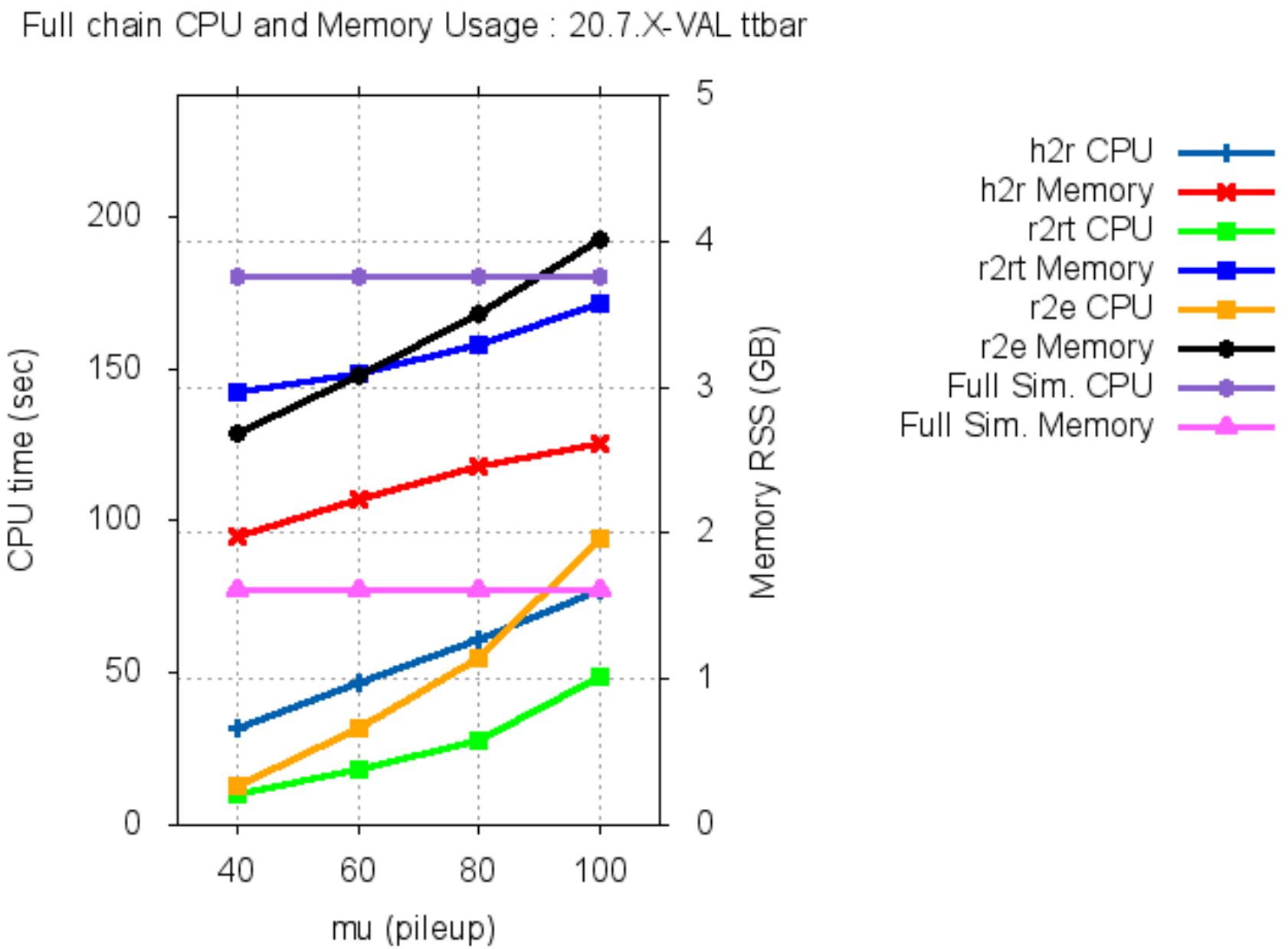
- grid resources have 2.0GB per code / Special high memory queues essential resource
- Switch from 20.1/Root 5 to 20.7/Root 6 added around 300-400MB due to auto-parsing of headers (ATEAM-259)
- Monitoring memory has been reduced to handle RAWtoALL workflow, which will improve CPU efficiency (event throughput)
- Forking memory after the 1st event can deliver more memory savings from AthenaMP (of order 200 MB/core reduction)
- RunTier0Tests.py has acted as an effective gate keeper on tags on Tier0 Release ... users are using it development releases (better to have a lightweight version). Raised the awareness of resource limits.

AthenaMP vs AthenaSerial

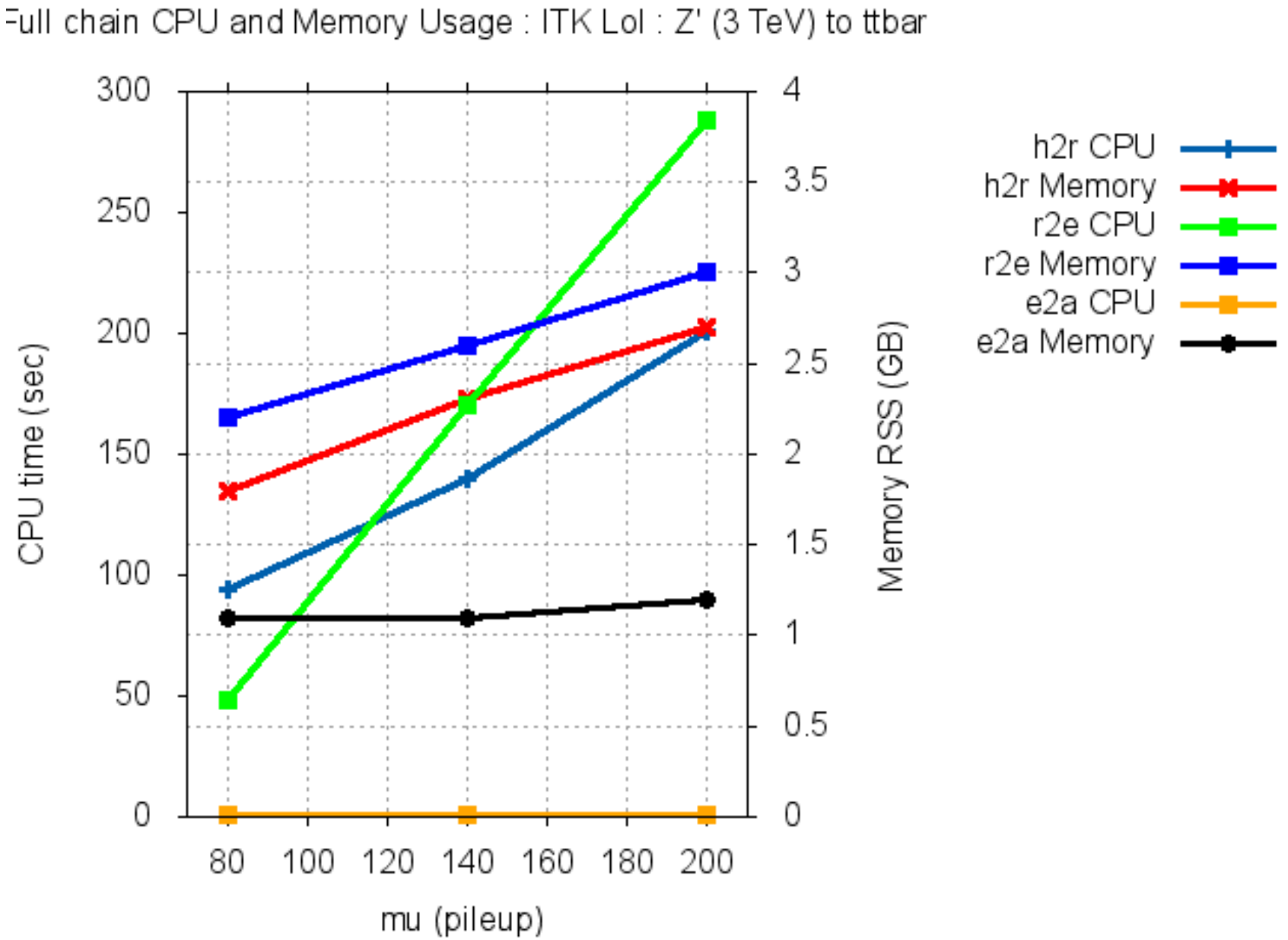


- 2000 event job using 8 cores from HITS to AOD (WW_lvqq)
- Used MemoryMonitor tool (R. Seuster and N. Rauschmayr (CERN-IT))
- Event throughout the same with much reduced memory overhead
- With ATLAS distributed computing, CERN IT "Understanding Performance" team developing "IOMonitorTool" metric (ATEAM-293) to understand I/O loads of our jobs and to help sites understand requirements.

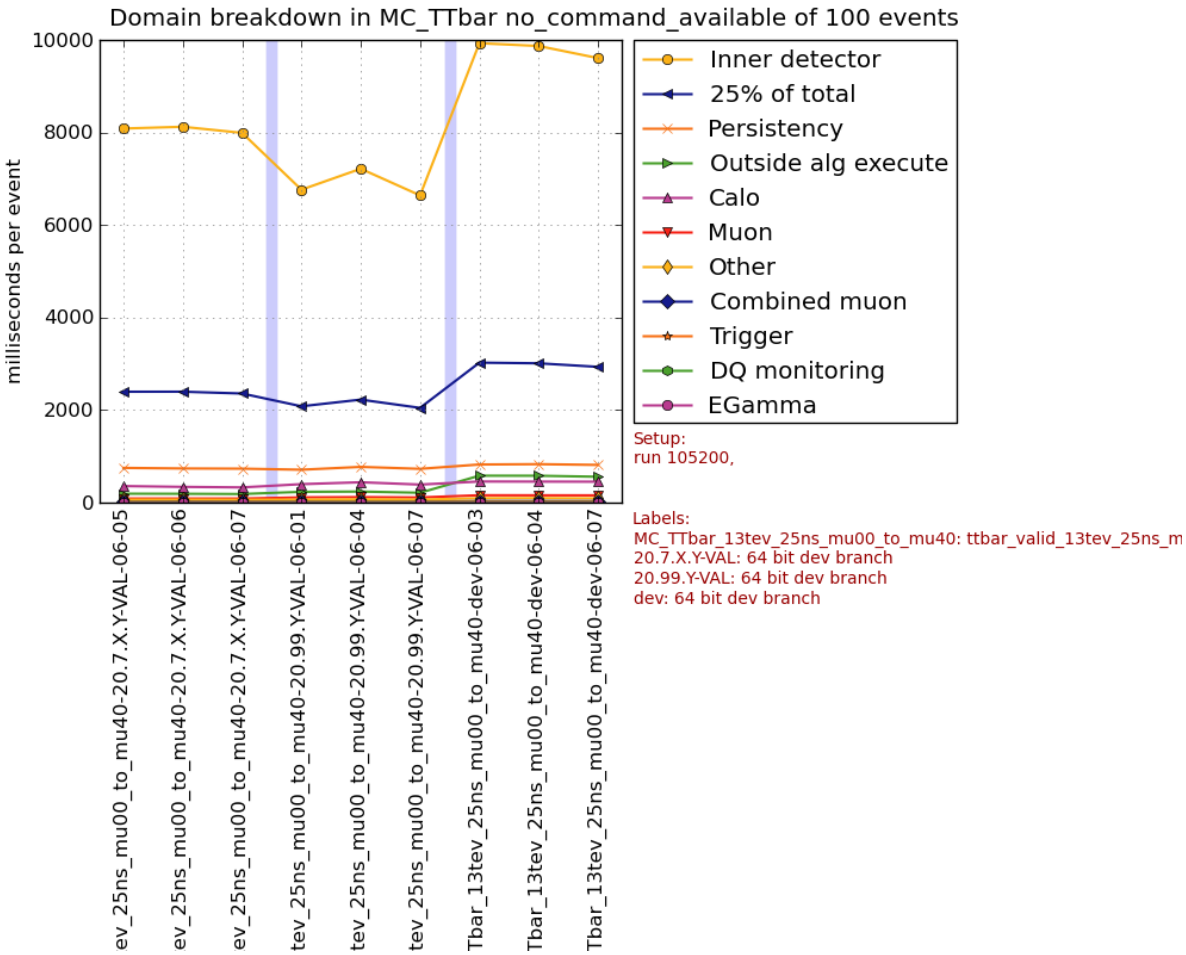
ATLAS workflow : Today



ATLAS workflow ITK



Workflows : CMT vs cmake (System Reconstruction ID/Calo/MS)



- Our cmake build is running slower in benchmark tests
- 20.99.Y-VAL (cmt) vs 21.X.Y (cmake) : Same tags. Different compilation options? external versions?
- (ATLASRECTS-3212)

Reciprocal-math compiler option

- Scott Snyder in ATLAS SW Core meeting talk discussed possible saving using "-funsafe-math-optimizations"
- See talk
- Test a sub-component flag "-freciprocal-math"
- PMB tests show that that the ave. cpu time per event in the RAW to ESD step reduces by 0.5 sec from 19 to 18.5 sec, or by 2.6%.
- Save 0.25 sec in the leading InDetSiSpTrackFinder algorithm, 6.9 to 6.6 sec.
- Save 40 ms in the 2nd leading InDetAmbiguitySolver algorithm, 1.93 to 1.89 sec.

Profiling tools

- PerfMon within Athena - Single-core PerfMonSD
- Heap/Memory monitor - Multi-core (N. Rauschmayr and R. Seuster) (CERN-IT-Profiling
- Finding Obsolete memory (N. Rauschmayr and S. Kami) (FOMTool)
- Large scale memory allocation studies (N. Rauschmayr and S. Kama) SCC-talk)
- COWTools within Athena (S. Kama)
- IgProf (developed by CMS)
- Using Valgrind at ATLAS
- gperftools Profiling athena with gperftools
- Intel VTune at CERN
- Challenge is to develop and maintain effective monitoring for AthenaMP/MT

Backup

xAOD Size

Default μ	40	60	80	100
xAODSize (MB/event)	0.65	0.97	1.30	1.74

ITK μ	80	140	200
xAODSize (MB/event)	0.98	1.43	1.82