Software Improvements related to ADC Performance

Johannes Elmsheuser[1]

22 Sep 2017

ADC TIM

[1] Brookhaven National Laboratory

Outline

Introduction

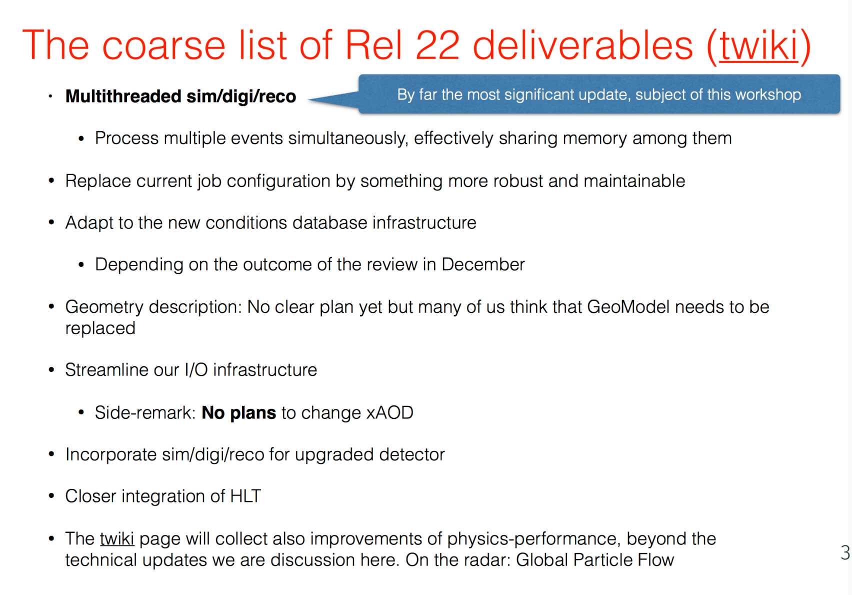
SharedWriter

Check-pointing

ART

Conclusions

Silde from W.Lampl from AthenaMT workshop on Monday - timeline: Run3



The coarse list of Rel 22 deliverables (twiki)

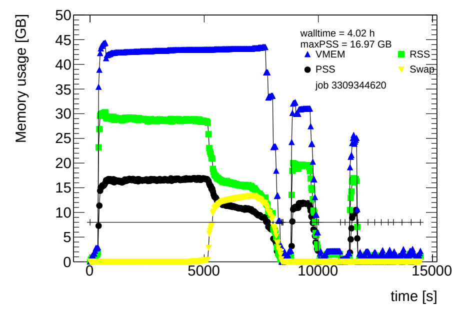
- **Multithreaded sim/digi/reco**
  - Process multiple events simultaneously, effectively sharing memory among them
- Replace current job configuration by something more robust and maintainable
- Adapt to the new conditions database infrastructure
  - Depending on the outcome of the review in December
- Geometry description: No clear plan yet but many of us think that GeoModel needs to be replaced
- Streamline our I/O infrastructure
  - Side-remark: **No plans** to change xAOD
- Incorporate sim/digi/reco for upgraded detector
- Closer integration of HLT
- The twiki page will collect also improvements of physics-performance, beyond the technical updates we are discussion here. On the radar: Global Particle Flow

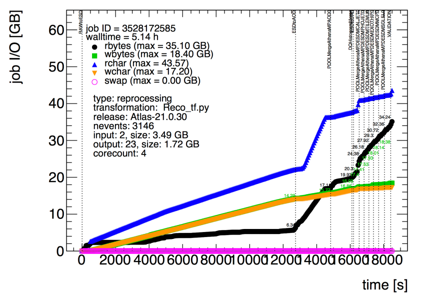
By far the most significant update, subject of this workshop

More collected topics

- Licence: Copyright added to all offline source code, open licence under discussions, then open the code
- Other architectures: x86 is basis, some R&D on ARM
- Containers: user documentation in place for Linux and MAC, with new release build system (RPMs) also easier way to create dedicated containers for developers

Optimize today's throughput - Profile of example jobs





- Marc de Beurs validated I/O additions to MemoryMonitor
- Very detailed study of I/O of different example workflows on Panda - see Presentation link
- rbytes info also used now in job brokering
- Would be nice to have network IO info per process similar to `nethogs` (probably too intrusive) - ideas, volunteers ?

Further SW improvements under development or wishes I

In the works, under testing or R&D:

- SharedWriter, SharedReader for AthenaMP outputs
  → see later
- CheckPointing vs. athena configuration rework
  → see later
- I/O additions to MemoryMonitor tool
  → under validation, see before
- Build one big static library for Geant4
  → Demostrated in 20.7, now work in progress for 21.0 (ATLASSIM-3150)
- Pile-up pre-mixing in MC - status discussed on Monday
- Try AutoFDO in simulation: execution speed improvements (link)
  → some technical hurdles on the way
- ART (ATLAS release tester) on the grid - replacement of RTT (twiki)
  → see later
- Output file compression with LZMA instead zlib (ATEAM-420, presentation link)
  → can safe ∼ 10 % in xAOD size, but increases write times from 10 to 60-150s and doubles reading times from 8 to 17s

Further SW improvements under development or wishes II

Further wishes:

- Fork after first event in AthenaMP (ATLASDQ-405)
- Full remote I/O (reading) root/https/metalink support in Athena/ROOT/Rucio
  → most pieces in place - needs coherent testing
- Open new output file with ending _NNN at some size/event limit (ATEAM-335)
- Panda job monitoring:
  → MemoryMonitorIO add network IO per process similar to nethogs
- Inputfilepeeker improvements/rewrite
  → Rewrite rather heavy procedure in a more lightweight mode, input files are open multiple times right now, project started a while ago, but no recent progress

Outline

Introduction

SharedWriter

Check-pointing

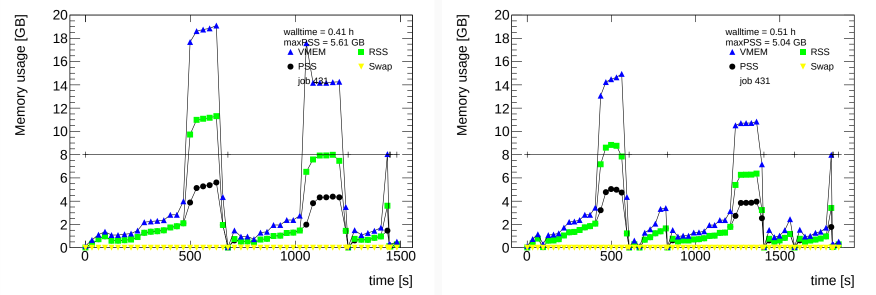
ART

Conclusions

SharedWriter and SharedReader

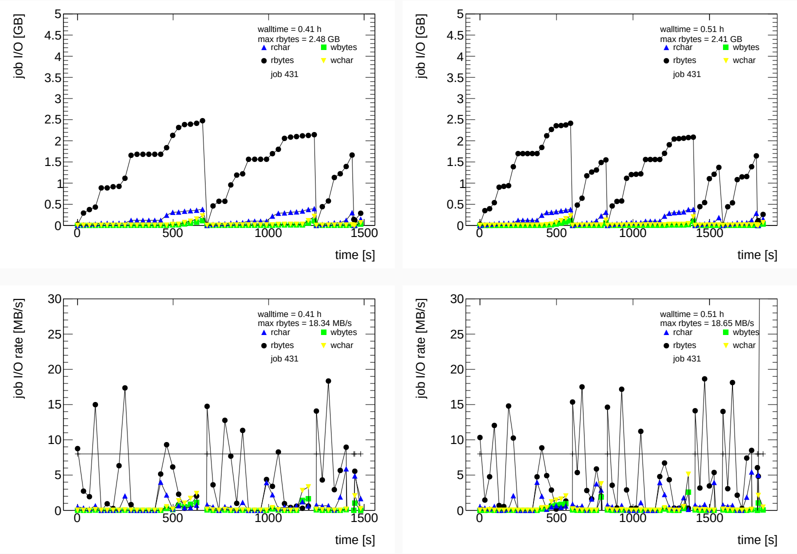
- Workflow developed by Peter van Gemmeren for AthenaMP input and output file access - presented in previous ATLAS S&C weeks
- Idea: use shared memory process to pass serialized objects from AthenaMP subprocesses to write single output file
- Developed in master git branch and now ported back to 21.0/21.2
- Extensive testing and debugging in the past weeks - details see ADCDPA-50
- SharedWriter can save the extra output file merging step - esp. DAOD_Merge step in derivations - at the cost of slightly more memory

Memory: 1 SharedWriter (l) vs. Regular (r) - q431 RAWtoESD



- q431 test in `Athena,master,r29` + a few patches
- 1 SharedWriter (left) vs. regular AthenaMP (right)
- SharedWriter uses 0.57 GB more maxPSS but overall uses $\sim$ 6 min (20 %) less walltime for 25 events

IO total (top), rate (bottom): SharedWriter (l) vs. Regular (r) - q431 RAWtoESD



Walltimes and Memory: SharedWriter vs. Regular - NCore=2

	SharedWriter	Regular	diff
Derivation train [s]	922	808 (w/o DAOD_Merge)	
only RAWtoALL [s]	864	753	+13%
only RAWtoESD [s]	658	571	+15%
Full RAWtoALL TRF [s]	1194	1451	-18 %
Full RAWtoESD/ESDtoAOD etc. TRF [s]	1505	1870	-20%

	SharedWriter	Regular	diff
Derivation train [GB]	5.38	5.60	+4%
only RAWtoALL [s]	6.56	7.01	+7%
only RAWtoESD [s]	5.04	5.61	+11%

- The single RAWtoSomething process takes about 10-15% more wall time with SharedWriter
- But overall the full TRF is 18-20% faster since there are no merging steps necessary at the end of the TRF

Open issues

- Instead of 1, there are as many MetaData containers in output files as AthenaMP processes
- DAODMerge files sizes smaller than SharedWriter output files due to different ROOT split level etc. settings
- Latest reco SharedWriter fixes will be in next AthDerivation,21.2.2.0 ad Athena,21.0.38 releases
- For derivations everything is in AthDerivation,21.2.1.0 already - production could benefit from re-shuffling trains/carriages
- Physics validation planned together with validation of new DAOD_PHYSVAL

Outline

Introduction

SharedWriter

Check-pointing

ART

Conclusions

Check-pointing prototype for Sim_tf (I)

- See Vakho's comprehensive ACAT talk: link
- Idea: reduce the job start-up time by check-point before AthenaMP fork and restart in subsequent jobs
- Technically challenging
- DMTCP + prototype code for check-pointing AthenaMP+Sim_tf in 21.0.27 (still some issues with e.g. run number)
- Need homogenous OS environment for check-pointing, like e.g. VirtualBox+CernVM (ATLAS@Home) or a large multi-core node on HPC
- ATLAS@Home: restart from the checkpoint image in 15-20 sec vs. regular initializations 4 min (fast conditions DB/Frontier connections) to 10-15 min (slow conditions DB/Frontier connection)

CHECK-POINTING PROTOTYPE FOR SIM_TF (II)

- AthenaMP Simulation Startup Times on Cori KNL Nodes - 300 jobs:

	Image size	Startup time (sec)	Startup speedup
Conventional AthenaMP	N/A	663.1 ± 22.8	1
Compressed image	550MB	50 ± 9.7	13.3x
Uncompressed image	1.8GB	20.8 ± 9.1	31.5x

- Several open iitems:
  - Fix run number change and possibly other variables
  - Physics output validation
  - Needs easy and automated check-point image creation and distribution

Outline

Introduction

SharedWriter

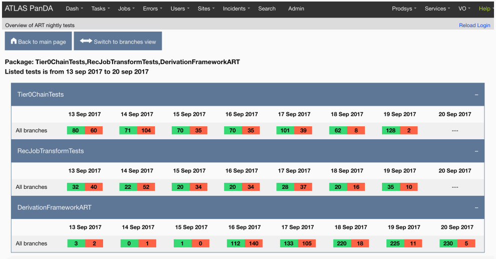
Check-pointing

ART

Conclusions

ART (ATLAS release tester) on the grid

- Move lxbatch based RTT test of nightly releases to ART on Panda
- Uses automated pathena/prun submission of predefined test jobs with artprod cert
- BigPanda monitoring overview on top regular job monitor (https://bigpanda.cern.ch/art/):



Overview of ART nightly tests

Package: Tier0ChainTests,RecJobTransformTests,DerivationFrameworkART
Listed tests is from 13 sep 2017 to 20 sep 2017

Tier0ChainTests

	13 Sep 2017	14 Sep 2017	15 Sep 2017	16 Sep 2017	17 Sep 2017	18 Sep 2017	19 Sep 2017	20 Sep 2017
All branches	🟩 80 🟥 60	🟩 71 🟥 104	🟩 70 🟥 35	🟩 70 🟥 35	🟩 101 🟥 39	🟩 62 🟥 8	🟩 128 🟥 2	---

RecJobTransformTests

	13 Sep 2017	14 Sep 2017	15 Sep 2017	16 Sep 2017	17 Sep 2017	18 Sep 2017	19 Sep 2017	20 Sep 2017
All branches	🟩 32 🟥 40	🟩 22 🟥 52	🟩 20 🟥 34	🟩 20 🟥 34	🟩 28 🟥 37	🟩 20 🟥 16	🟩 35 🟥 10	---

DerivationFrameworkART

	13 Sep 2017	14 Sep 2017	15 Sep 2017	16 Sep 2017	17 Sep 2017	18 Sep 2017	19 Sep 2017	20 Sep 2017
All branches	🟩 3 🟥 2	🟩 0 🟥 1	🟩 1 🟥 0	🟩 112 🟥 140	🟩 133 🟥 105	🟩 220 🟥 18	🟩 225 🟥 11	🟩 230 🟥 5

- Comparisons within job with results from previous day(s)
- Tests are gradually being migrated and improvements underway

Outline

Introduction

SharedWriter

Check-pointing

ART

Conclusions

Conclusions

- AthenaMT migration is underway for Run3, but algorithmic code migration will take time
- Some workflow improvements in the works which will (hopefully) improve the resource utilisation earlier than Run3

BACKUP

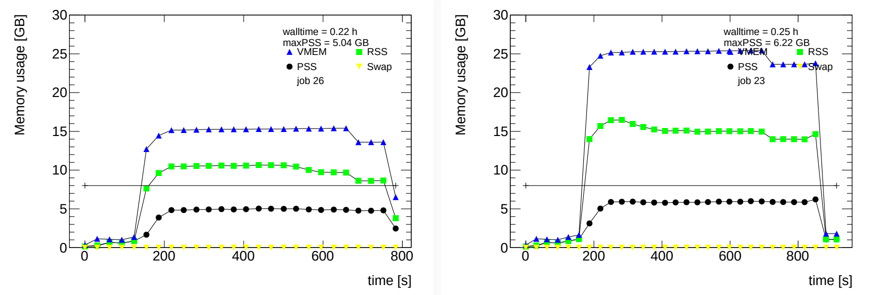
Merged file sizes: 1 SharedWriter vs. Regular

- Input: data16_13TeV:AOD.11078889._000001.pool.root.1
- 500 events processed

	JETM1	JETM2	JETM3	JETM4	JETM5	JETM6
SharedWriter (initial version) [MB]	5.3	7.5	2.3	3.4	1.5	11
SharedWriter (fixed auxstore) [MB]	3.6	6.2	2.1	3.0	1.5	7.5
DAODMerge [MB]	2.7	5.2	0.9	1.9	0.38	6.1
SharedWriter (22 Aug) [MB]	3.6	6.2	2.1	3.0	1.5	7.5
DAODMerge (22 Aug) [MB]	2.5	5.2	0.9	2.0	0.39	6.4
DAODMerge on SharedWriter single file (22 Aug) [MB]	2.5	5.2	0.9	2.0	-	6.4
No. of events	50	101	2	17	0	48

- SharedWriter inital version accidentally wrote all auxstore variables, fixed in subsequent version
- Number of events in corresponding files identical
- SharedWriter files contain: 4 container DataHeader/MetaData and DataHeaderForm [MetaData] vs. 1 container in DAODMerge files
- When running DAODMerge with a single inputfile from the SharedWriter output, the size is reduced and only 1 MetaData instead of 4 container remains

Memory: 1 SharedWriter (l) vs. 6 SharedWriter (r)



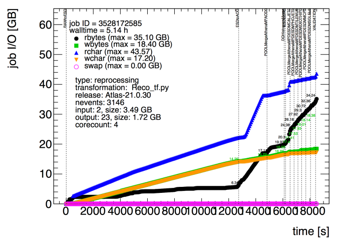
- 10000 events processed, `AthDerivation,21.2,r01`
- 1 SharedWriter (l) and 6 SharedWriter (r), 6 athena.py vs. 11 processes after the fork
- Walltime: 820s (1 SharedWriter), 967s (6 SharedWriter)
- maxPSS: 5.04 GB (1 SharedWriter) vs. 6.22 GB (6 SharedWriter)

Outline

MemoryMonitor IO and Network

MemoryMonitor I/O and Network

- Marc de Beurs has validated the I/O additions to the MemoryMonitor
- Very detailed study of I/O of different example workflows on Panda - see Presentation link
- rbytes info also used now in job brokering



- Would be nice to have network IO info per process similar to `nethogs` - ideas, volunteers ?