ATLAS simulations performance on Summit

Sergey Panitkin
BNL

Introduction

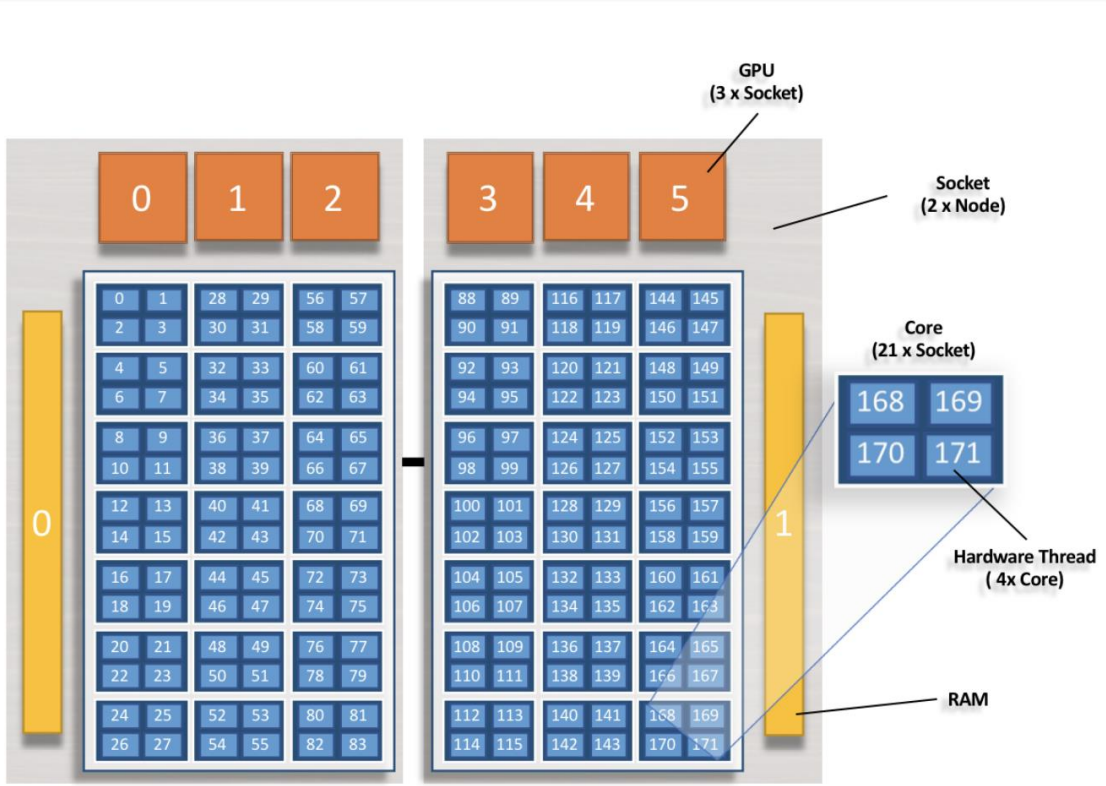
- In February AthSimulation release 21.0.34 was built by Alex Undrus on Summit, natively for Power9 CPU
- I started testing ATLAS detector simulation on Summit right away. First results were already shown in March at the ATLAS site jamboree meeting at CERN
- Goals:
  - Prepare for running actual production release (to be determined and build)
  - Learn how to use job submission interface, IBM Spectrum LSF and jsrun utility
  - Study performance, scalability, effects of multi-threading on Power9 CPUs, etc
- Work in progress.
  - Scalability tests are not completed since we ran out of allocaton

Summit at OLCF. Node structure

~4600 IBM Power System AC922 nodes each with 2 Power9 CPU and 6 nVidia Volta V100 GPUs, 512GB DDR4 RAM + 96GB HBM2

~193K CPU cores in 4600 nodes

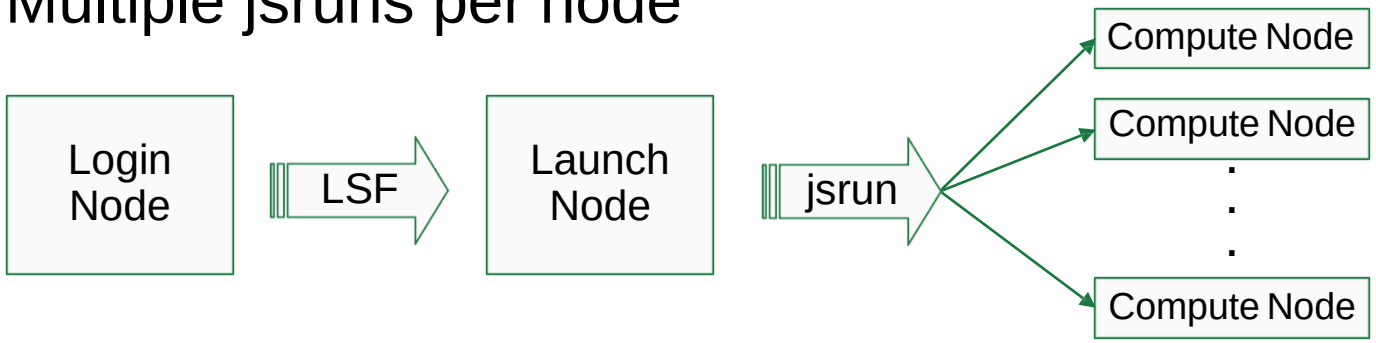
- 1 node
- 2 sockets (grey)
- 42 physical cores* (dark blue)
- 168 hardware cores (light blue)
- 6 GPUs (orange)
- 2 Memory blocks (yellow)



jsrun utility allows for fine grained allocation of CPU and GPU per MPI rank, as well as choice of SMT level. Large parameter space!

jsrun Introduction

- Launch job on compute resources
- Similar functionality to aprun and mpirun on Titan
- Still in development
- Launch nodes
  - Similar to Titan
  - Non-jsrun commands executed on launch node
  - Shared resource
- Multiple jsruns per node



From Chris Fuson's talk at OLCF user meeting

Basic jsrun Examples

Description	Jsrun command	Layout notes
64 MPI tasks, no GPUs	***jsrun –n 64 ./a.out***	2 nodes: 42 tasks node1, 22 tasks on node2
12 MPI tasks each with access to 1 GPU	***jsrun –n 12 –a 1 –c 1 –g1 ./a.out***	2 nodes, 3 tasks per socket
12 MPI tasks each with 4 threads and 1 GPU	***jsrun –n 12 –a 1 –c 4 –g1 –bpacked:4 ./a.out***	2 nodes, 3 tasks per socket
24 MPI tasks two tasks per GPU	***jsrun –n 12 –a 2 –c 2 –g1 ./a.out***	2 nodes, 6 tasks per socket
4 MPI tasks each with 3 GPUs	***jsrun -n 4 –a 1 –c 1 –g 3 ./a.out***	2 nodes: 1 task per socket

From Chris Fuson talk at OLCF user meeting

Hardware Threads: Multiple Threads per Core

jsrun –n 12 –a 1 –c 2 –g 1 –b packed:2 –d packed ./a.out

- 12 resource sets
- x
- 1 task
- 2 physical cores
- 1 GPU
- bind tasks to 2 cores in resource set
- assign tasks sequentially filling RS first

User should set OMP_NUM_THREADS = 4

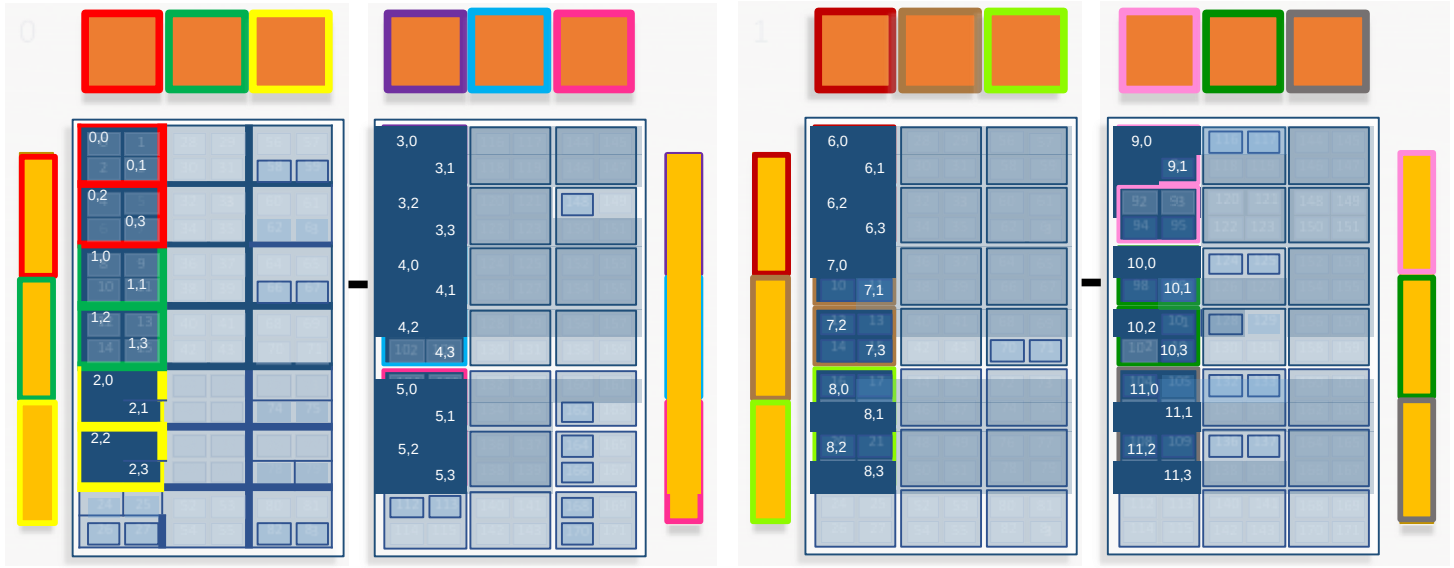
#BSUB –alloc_flags smt2

For rank 0 jsrun will set

OMP_PLACES {0:2},{4:2}

First RS (red) contains

- 1 task (0)
- 4 threads (0-3)
- 2 cores (0,4)
- 1 GPU (0)



Mixing of CPU and GPU payloads is very important. NGE!

From Chris Fuson talk at OLCF user meeting

First performance tests on Summit

ATLAS rel. 21.0.34 compiled on Summit. AthenaMP Geant 4 detector simulation. Single node.

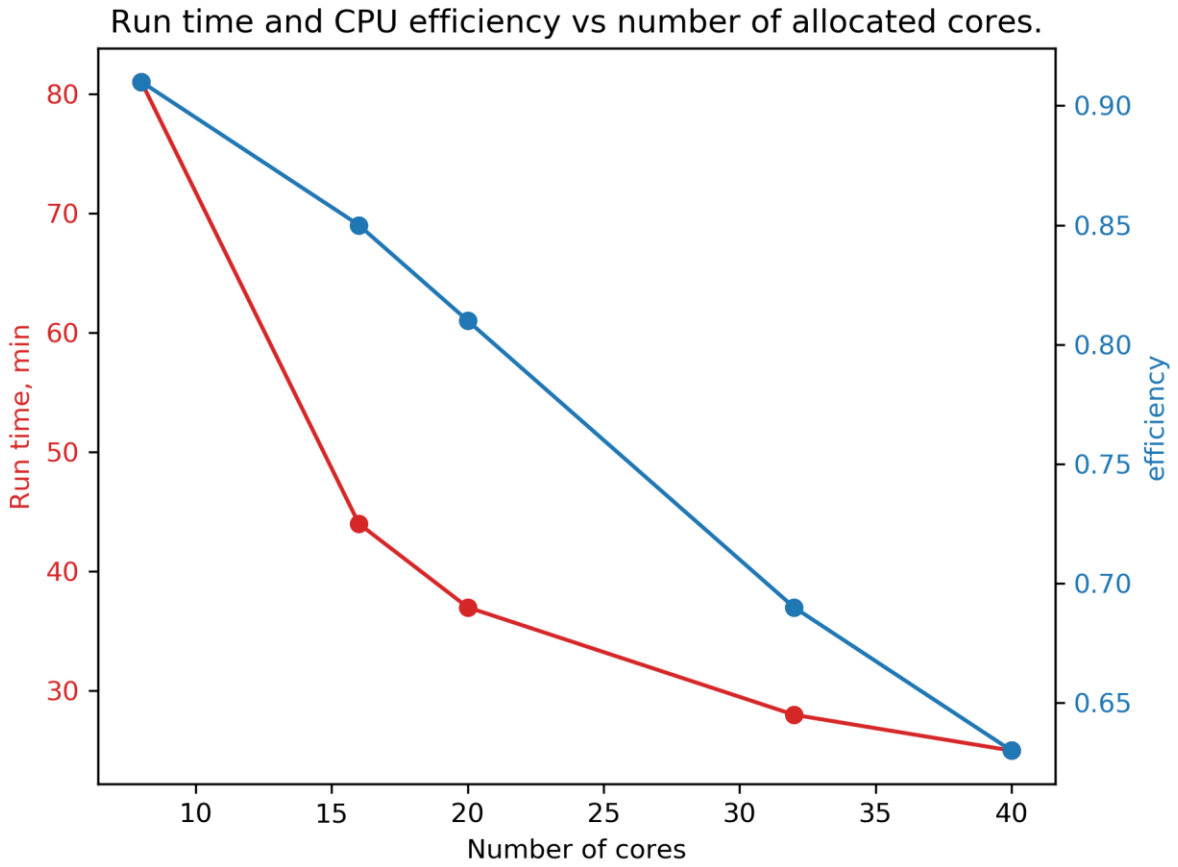
Sim_tf.py --inputEVNTFile=/ccs/home/panitkin/EVNT.06820177._000107.pool.root.1 --maxEvents=100 --preExec "EVNTtoHITS:simFlags.SimBarcodeOffset.set_Value_and_Lock(200000)" "EVNTtoHITS:simFlags.TRTRangeCut=30.0;simFlags.TightMuonStepping=True" –preInclude "EVNTtoHITS:SimulationJobOptions/preInclude.BeamPipeKill.py,SimulationJobOptions/preInclude.FrozenShowersFCalOnly.py" --skipEvents=0 --firstEvent=165001 --outputHITSFile=HITS.10974383._000371.pool.root.1 --physicsList=FTFP_BERT_ATL_VALIDATION --randomSeed=189 --conditionsTag "default:OFLCOND-MC16-SDR-14" --geometryVersion="default:ATLAS-R2-2016-01-00-01_VALIDATION" --runNumber=301053 --AMITag=s3126 --DataRunNumber=284500 --simulator=FullG4 --truthStrategy=MC15aPlus

Cores	SMT	workers	events	Run time, min	CPU efficiency
8	1	8	100	81	0.91
16	1	16	100	44	0.85
20	1	20	100	37	0.81
32	1	32	100	28	0.69
40	1	40	100	25	0.63

preliminary

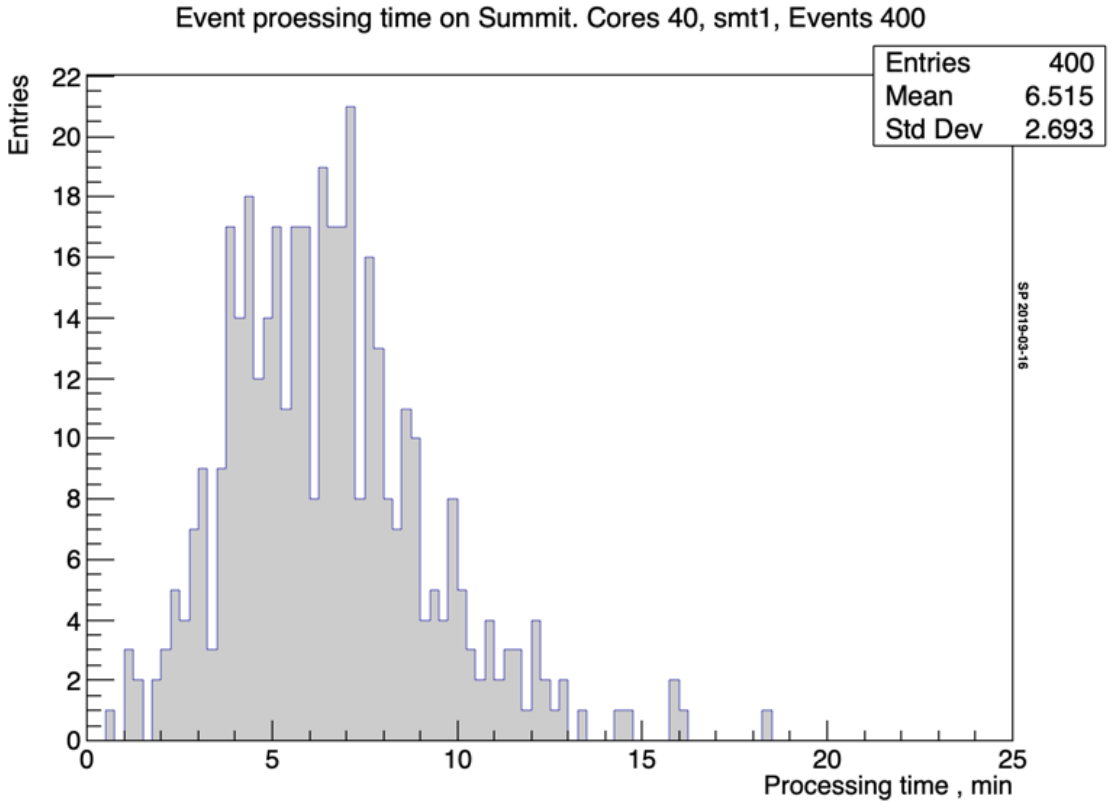
For comparison: same run with 100 events on Titan's 16 cores, with 16 AthenaMP workers took ~62 minutes. Summit cores are ~40% faster.

Run time and CPU efficiency vs number of allocated cores



Single Summit worker node. AthenaMP, 100 events, SMT1

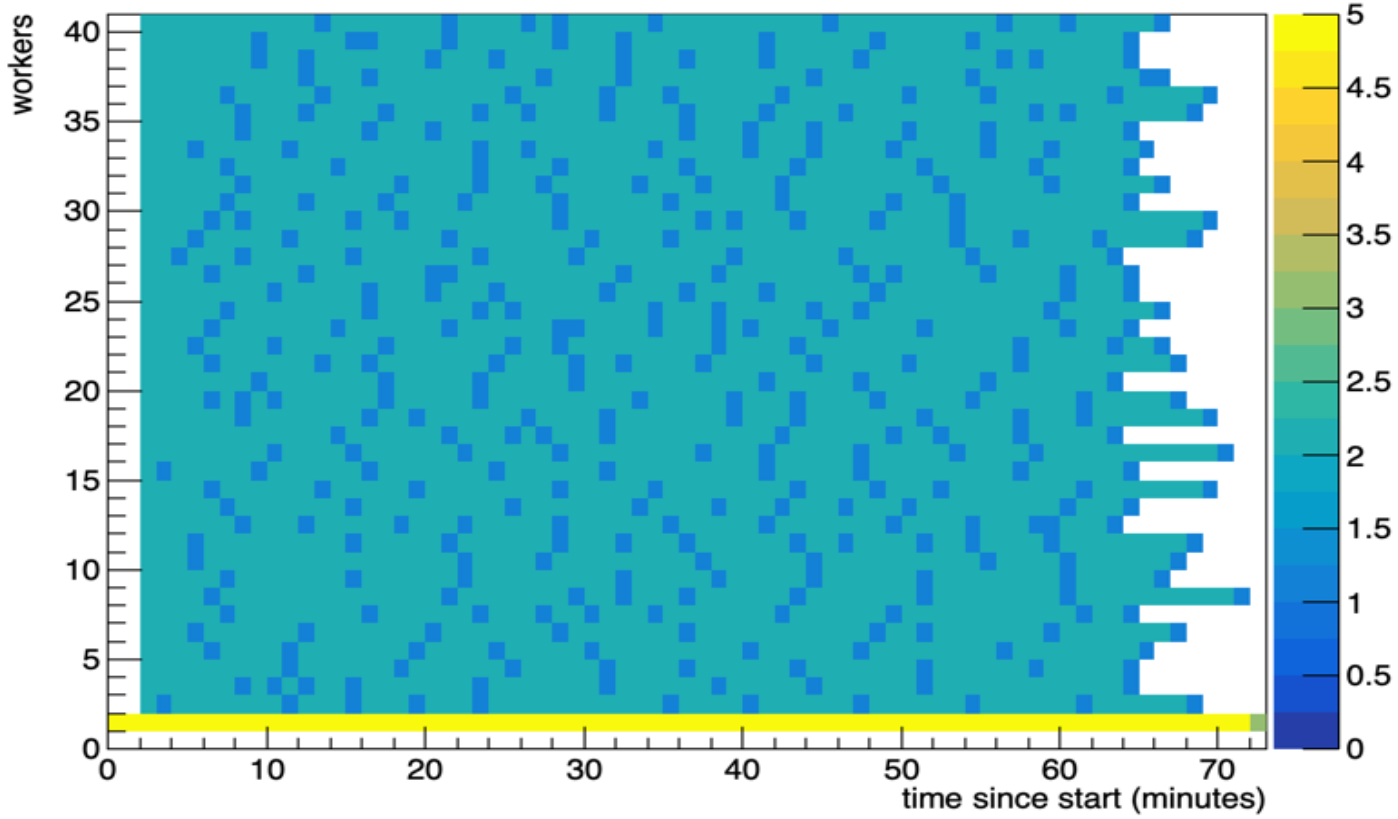
ATLAS simulation event processing time



1 node, 40 cores per job, 40 AthenaMP workers, 400 events

AthenaMP worker occupancy on Summit

1 node, 40 cores per job, 40 AthenaMP workers, SMT1, 400 events



Main contributors to inefficiency are idle cores at the end of job run.
AthenaMP initialization and configuration time is smaller @ ~2min

First performance tests on Summit. Effect of SMT

Summit's CPUs support Simultaneous Multi-Threading (SMT), each physical core supports up to 4 hardware threads.

ATLAS rel. 21.0.34 compiled on Summit. AthenaMP Geant 4 detector simulation. Single node.

Cores	SMT	workers	events	Run time, min	CPU efficiency
8	1	8	100	81	0.91
8	2	16	100	69	0.89
8	4	32	100	69	0.85
10	1	10	100	69	0.89
10	2	20	100	59	0.86
10	4	40	100	55	0.83
20	2	40	400	110	0.92
20	4	80	400	107	0.88

Use of SMT2 helps to improve run time, SMT4 shows diminishing returns

preliminary

Summary

- AthSimulation release 21.0.34 was built on Summit
- Testing of ATLAS detector simulation on Summit started.
- First results showed that per core Summit is ~40% faster than Titan
  - 40 cores per node and SMT leads to significant boost in per node performance compared to Titan
- Enabling SMT2 shows noticeable increase in performance, SMT4 does not show large effect
- In order to maintain job efficiency it is important to balance number of cores per job, SMT level with number of events per job and Summit policies constrains
- Scalability studies are in progress, pending project extension

The End

Summit batch queue polices

Bin	Min Nodes	Max Nodes	Max Walltime (Hours)	Aging Boost (Days)
1	2,765	4,608	24.0	15
2	922	2,764	24.0	10
3	92	921	12.0	0
4	46	91	6.0	0
5	1	45	2.0	0

Limit of 2 *eligible-to-run* jobs per user
No more than 100 jobs in any state at any time per user

Current CSC343 allocation status. 04/24/19

summit usage in Node-hours:

Project Totals

Project	Allocation	Usage	Remaining
csc343	5000	6429	-1429

Individual Usage

UserID	Usage	% of Total
amalik	3802	59.15%
panitkin	2626	40.85%
psvirin	0	0.00%