Potential HPC usage for generators and simulation

- Rod Walker, LMU 15th Jan 2014

- Motivation and available resources
- Current usage for ATLAS
- CPU time allocation roadblock
- Can HEP efficiently use HPC capabilities
- Wishlist

High Performance Computing

- Supercomputers to run massively parallel applications
- Historically used special architectures & OS's
  - e.g. BlueGene, Cray, SGI
- Latest competitive supercomputers are x86 based
  - familiar linux cluster, with fast interconnect(InfiniBand)
    - IB makes it more expensive (~20%) - partially offset by bulk deal
  - OS's are RH-variants, SUSE
    - can run unmodified ATLAS binaries
- Examples of both BlueGene and x86 resources
  - *potentially* accessible to ATLAS

DoE funded Supercomputers

Machine	Cores	Location	Notes
Intrepid	164K	Argonne	BlueGene/P 850 MHz PPC 450
Mira	768K	Argonne	BlueGene/Q 1.6 GHz PPC A2
Hopper	153K	NERSC	AMD Opteron
Edison	10K → 83K	NERSC	Intel Sandy Bridge
Carver	10K	NERSC	Intel Xeon X5550
Titan	299K (+299K GPUs)	Oak Ridge	AMD Opteron +Nvidia Tesla

Works out to about 7 billion 2GHz x86-equivalent CPU-hours/year

~ factor 7 of ATLAS capacity

x86 Resource Examples

- SuperMUC, Munich
  - 155,000 Sandy Bridge cores, 2.8M HS06
    - ATLAS 2013 T1/2 pledges ~ 730K HS06
  - Suse Enterprise Linux 11, 2GB/core
  - warm water cooling
    - 40ºC inlet. 70ºC outlet used to heat building
- Hydra, MPP, Munich
  - 75k cores, sand and ivy bridge

Motivation

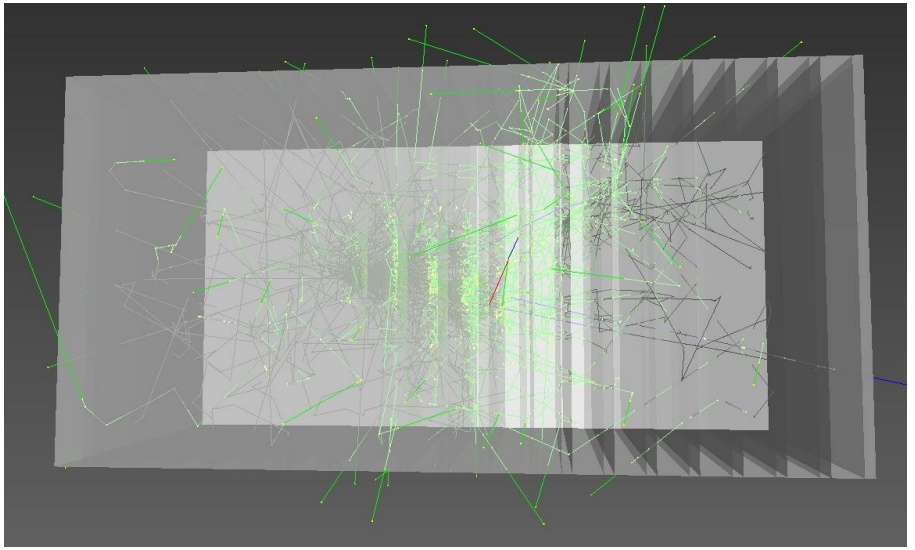
- Funding profile flat for Run II, but computing requirements increase
  - lumi, high pile-up, generators
- HPC resources typically CS toys with general science customers
  - HEP has good science case → allocation
- 10% cpu used for Generators
  - offload to HPC to leave Grid for data-intensive work

Current Usage

- HPC used for ATLAS production in NorduGrid
  - running single core jobs via ProdSys
- Semi-manual event generation on BlueGene
  - US leadership facilities
- Production integrated Geant4 on x86 based machines outside NorduGrid
  - identical method
    - administrative success rather than technical

Application porting/rebuild

- Tom rebuilt some stand-alone packages on BlueGene
  - Geant4
    - no code changes.
    - difficult build environment
    - ran N03 example
  - Root
    - no code changes, but Reflex not built
    - ran some Toy MC
  - Alpgen
    - minimal code changes required
    - 80M events(4-vectors) delivered to ATLAS
      - 30mins on 2048 cores
    - would need athena format EVNTS for validation
      - cannot be done on BlueGene
      - revert to old method of pushing 4-vectors though athena



Full integration for easier x86 case

- No rebuild required
  - Athena integrated generators fine
- only need shared-FS and an edge service
  - some inbound and outbound ports open
- ARC CE gets real payload, and stages data to shared-FS
  - no grid-client, or IP needed on WNs
  - submit serial, mcore, or mpi jobs to batch(Loadleveler,slurm,..)
  - exactly how NorduGrid HPC integrated for years

Hydra @ MPP, Munich

- Allowed ARC CE, batch and shared FS access
  - so integrated into production system
  - running MCORE AthenaMP sim
- ATLAS invited to backfill
- potential allocation in future

CPU allocation on HPC

- Application judged on
  - scientific merit – fine.
  - need for HPC capabilities, parallel scalability
    - thus far, completely trumps scientific merit
- Three possible usage modes
  - 1.just another linux cluster – run serial or whole-node
  - 2.fake parallel job
    - MPI doing nothing , eg. athenaMT
    - sub-scheduler to fill serial jobs on assigned nodes
  - 3.genuine parallel workload
    - preferably with advantage over serial equivalent
    - use of co-processors: GPU,MIC,Phi
- (2) more acceptable, to HPC people(computer scientists), but worse than (1)
  - all nodes occupied until last thread finishes
- With (3) can make an honest application
- (1) perfect for backfill, but needs attitude change from some HPC people

(1)Athena MP

- Use the whole-node, but no more
  - no MPI, multi-node parallel job
- Fork after initialization to share RAM
  - forked processes split events between them
- Keep them short to take advantage of backfill
- Slightly worse throughput than serial
  - but advantages
    - scaling for multi-core cpus: sw access, in/out files
    - fewer files(for DDM), less merging
    - minimum requirement for HPC

(2)GEANT MT

- Multi-threaded Geant processes single event in multiple threads
  - can use co-processor threads
- Possible to run with mpi
  - fake – no communication between nodes
- Speed comparison with AthenaMP?
  - only wins with co-processor?

(3)Genuine parallel HPC apps

- Sherpa MPI
  - current production serial jobs loads manually created integration result
  - MPI parallelizes integration step
    - some minimal MPI really needed
- On HPC, could also generate events across same nodes
  - would not need MPI, but increases cpu shifted to HPC
    - can sell this to HPC people
  - not clear #nodes useful for integration is same as needed for generation

Automate Integration step

- Understand from Frank that manual step is bottleneck
- Propose to put this into ProdSys
  - task request, trf, store integration result
  - support whole node and MPI parallel
    - # nodes depends on process
  - some ProdSys work to allow mpi jobs
- Possibility to ssh to HPC and run script (created by ProdSys
  - some HPC may not allow gatekeeper(ARC CE)
  - manual but
    - scripted
    - not Frank

(3)Fantasy G4

- Could we have genuinely parallel G4 with advantage over serial?
  - single thread responsible for one or more volumes
  - MPI passes 4-vectors between threads, as particles traverse volumes
  - supercomputer mimics detector, process whole event in parallel
- Any benefit over MP/MT?
  - depends if initializing a volume uses significant resources
  - maybe some volumes handled on co-processors
- Maybe some completely different way to take advantage of HPC capabilities for G4

(3)Fantasy G4

- G4 contact confirms it is fantasy BTW!
  - Geant V, for vectorization, is current dev path
  - leave slide in as example. There may be some other crazy scheme to take advantage of HPC capabilities.

Wishlist for any HPC app

- Benefits from HPC capabilities
  - MPI, GPU, Phi, MIC
- Flexible job shape
  - number of nodes vs. time
    - short fat job or long thin, might fit scheduling hole best
    - assumes predictable run time
- High-level checkpointing
  - batch can stop and restart task
- Preemptable with minimal loss
  - event server pull and process single event, store output

Conclusions

- Most scientific computing in HPC resources
  - can technically integrate into ProdSys(for x86)
  - HEP must win political fight for their share
    - whole-node backfill and genuine parallel apps
- Generator integration step best candidate
  - can tag on very serial generation part
- Proposed work to automate Sherpa-MPI
  - integrate into ProdSys
  - pursue this as part of (newish)ATLAS HPC working group