Berkeley TIM, November 9-11
Meeting Notes

Monday, November 9. **Core Software**

Conditions data access

How to substitute for SQLite access (important when no outbound access)? Andrea, can have a local server. Chris - thinks CMS implementation can read directly from SQLite. Marco - LHCb also interested and need SQLite access to work. Clearly it is desirable not to have to start a local server.

Format for payload? Under investigation. Data formats do need worked on to understand constraints and benefits (Shaun Roe investigating).

Open intervals may be risky. Need some mechanism for knowing the last interval does not go on forever (espc. when reading in-file metadata).

Efficiency loss at LB boundary will depend on number of threads, number of events, time to process events. It's not perfectly scalable. Chris strategy 2 is ok for CMS at the moment, but relaxing constraints.

Andrea - most conditions are lumi-block based. Sami - attach conditions to event? Problem is that conditions do not generally have event frequency granularity (putting them into event store is basically solution 1).

Graeme - like case 3. Simplify client code except where sub-lumi data is needed, then we use new conditions infrastructure with detector/client agreeing on how to interpret this (new Run3 conditions infrastructure).

CMS have no callbacks - have designed detector store (equivalents) to be very fast. Algorithmic code *always* asks the det store. FFReq report - alg code asks for *calibrated* conditions, backend triggers raw recovery and conditions preparation.

Tom - we have magnetic field conditions and other conditions that changes faster. We have

David - can be hard to know if you are at the end of lumi-block (e.g., event service). Need to watch out for this.

Chris - push for frontier to be thread safe if ATLAS want it too!

Stefan - do we have lumi information downstream? Yes, but fewer events per block. However, conditions validity is often much greater than an LB. Scott - should look at what conditions analysis jobs need to see how much of an issue this is.

Sami - in online do not know *apriori* when conditions will change. More checks are built into code to poll if conditions have changed. (Need to better understand online use!)

Vakho - let's take ½ day for conditions discussions in the hackathon.

Zach - Can we have a function call to load conditions early in the job (before the event loop) please?

Framework support for accelerators

Graeme - have the scheduler only say "A->D", but call the plugin deal with how to execute A->D (CPU or Accelerator). Then this is really independent of the main scheduler (BTW, plugins for scheduler are probably useful for HLT). Can make decision based on knowledge of state of CPU and accelerator. Sami - actually not too much of a complication to the scheduler. (No definite conclusion though.)

Conditions bound to events -> pass on a pointer to the conditions reference for each event (slide 12)? Charles - use callbacks to prepare calibrations. Chris - need to be careful of thrashing going back and forth across boundaries. Has implications for work 'packages' (algs or sets of algs). Pseudo-code examples would help!

Gaudi changes / StoreGate / *Handles

Migration to v27. We should do that as quickly as possible. The head version of Gaudi already has v27. AthBaseComps need no updates, there is no Hive-specific version of either SG or AthBaseComps.

What happens if two algorithms write to the same object? There is a lock, but right now bad things will happen. Two options:

1. Write two separate containers and the merge them
2. Define a sequence
3. Use aliases?

What happens if you have multiple algorithms taking the same input and modifying it? This can be solved by defining a sequence, such that the order is fixed. `ForwardSchedulerSvc.CheckDependencies` **should be** `True` **by default.**

Note, use class names not ID numbers in python. ClassID is only for C++ code to run fast!

Looking for **volunteers to start the migration to Data Handles**. They seem to be in good enough shape now.

Processing phases are used to do 'macro' scheduling in an event (run this set of algorithms first). This adds barriers inside the processing of an event, so should reserve this for cases where it's really, really needed!

Event Views

Chris - do you want a hierarchy of schedulers? CMS's mixing module can do something like this. Benedikt - this probably could be done with the hive (forward) scheduler. Relates to talking to GPUs, which also needs scheduling. What are the interface changes? Can they be made the same? **This should be investigated.**

Eric - is view creation trigger specific? No - you could use them offline as well (for ISF, high density tracking, etc.). Tracking demonstrator had problems at boundaries.

I/O

Validation - it's a good idea, but i/o **metadata experts need to help** integrate these tests into ATN and RTT (and even physics validation).

Is metadata access monitored? We're not sure - **should check this**.

ROOT can open the same file for *reading* twice. But it can't do it for writing. Can you read different branches from two threads? Not at the moment and technically tricky (TTreeCache). Scaling problems come from different sized branches (but tail problem).

File Manager is not thread-safe.

Long term for AthenaMT should have no dependence on data model - thus **drop EventInfo incidents passing back EDM objects**.

DataHeader is a very heavy object - how much do we really use? And how much might we use in AthenaMT?

Run micro-job through a profiler to see where the 20s runtime is going.

Not clear if metadata objects should be able to merge themselves - there are advantages, but it means the objects will be more complex (inherit from TObject?). Philippe - can actually write any object into a TDirectory, so don't need to use TObject. **Should discuss this in a splinter group**.

Should have both **validation** and **profiling** in place for merging workflow. **PMB**.

Problems in the past due to **incident sequencing** - so should **add this to validation of jobs**. Either teach the incident service what the correct order is or do post-facto validation vs expected order.

Event Service

Want to deliver data in ~15min units of work.

Philippe - isn't this just a remote read with TTreeCache? Yes, but use case is important - EVNT not important. Analysis is a real challenge. (TTC is not efficient for MP case.) Our "cluster" is 10 events - for simulation that's too long. Zipping and deserialisation would be bad to concentrate on a single node, could be a bottleneck.

Tom - we optimise for global throughput. Some jobs may stink and it's ok if they are a small fraction.

Metadata writers need to be changed to cope with this new processing model. Each fragment will have incomplete metadata, but the final merged output must have coherent set.

Beware of complexity - costs of problems can be considerable in (re)processing and bugs are more likely to happen. e.g., duplicated event issue.

Should develop with the ESS idea aiming to ultimately **reduce overall complexity**.

Tuesday, November 10. **Reconstruction**

Katie's talk: quite a lot of discussion about applicability for ATLAS. We have short loops (~4) which are not easily vectorizable.
TBB worked well for CMS (and works well for us, AFAWCS)

Andi's talk:

Pattern recognition - per module probably too small a loop to benefit from parallelism.
Paolo - Roberto showed that SP formation not vectorized.

John - trigger also doing GPU-based track finding, but implementation a bit different (less overlap issues) to Johannes's & they have some tests which should be looked at by offline.
Eric - you're showing the current design. Maybe we really need to rethink for a future parallelised reco.

Nick's Talk:
Feedback from tracking software to ITK design?
ITK taskforce had a study showing the time spent for a triplet of double layers, and averagely spaced layers. So looking at optimal spreading out of layers. Problem is we don't know if the tracking strategy is optimally extrapolated to high mu, so need to be cautious with conclusions.
How does non-ID software scale? Generally, not as bad as ID - easier scaling.
Digitisation has memory issues at high-mu.
Can we put a **SLHC job into PMB**? **Yes**, if it's useful for the community. Does not need to run every night (2 x week?).
Vertexing at very high mu is not as useful for physics as at low mu (it's an extra track constraint).

Ed's Talk:
Could use event views for special cases like punch through jets? Maybe better to deal with this 'internally' - event view is just overhead?
CPU costs in the trigger for muons are (relatively) higher. Could be a good test case for optimisation (although budget is per-event *total* cpu; parallelising doesn't reduce total cpu cost).

Tuesday, November 10. **Simulation**

Elmar's introduction to the framework
Q: Have we looked at the Virtual Monte Carlo package?
A: (Andi) We looked, but it didn't provide all the functionality we wanted.
Q: Could you have extended it?
A: (Zach) Yes, but it provides a whole lot of things that we don't want, so extending it is not the solution to this one.
Q: (Steve) Do you let particles change simulators part-way through?
A: No, we basically decided that we weren't going to support that
Q: Can we do event-level parallelism?
A: Not easily by passing the event on the stack, but the "simple" G4Hive approach might work (with some effort to ensure that the services are watching which thread they are in)
C: (Andrea D) The way we are considering sub-event parallelism is a good way to do this in Geant4 language
C: (Zach) Can we use a G4Run to demarcate each athena event as a set of G4Events?
Q: Does the splitting into sub-events actually improve throughput?
A: Not as such; there are other ways we should improve throughput first. It speeds up the synchronization points perhaps, and it could save us some memory.

Q: Does the split between signal and pileup digi allow clusters to merge between signal and pilupe?
A: No, not with this scheme. Not obvious fast digi will get that right. Also, it's still a pretty small effect (except for the TRT); it's high-energy jets where that's a big issue, and those jets are all coming from the same event.
Q: What sample are these numbers for?
A: This is mu=40, and we aren't totally sure :(

C: This is very impressive! :-)

Andrea's talk on Geant4 Updates

Q: What's a biasing scheme?
A: It's an artificial enhancement in some region of phase space
Q: Is the 10% in russian roulette at some cost in physics?
A: It is not clear from what CMS have provided us with.
C: We need to chat with physics and CP groups to see what samples (pi->mu decay? punch through?) might benefit from biasing -- this could give us a big advantage
Q: This is a much easier version of a rewritten physics list, right?
A: Yes, and rewriting the physics list could be quite tricky
C: This could be very helpful for our inner detector fast simulation setup!
Q: What's the memory overhead?
A: Don't know yet, but the algorithm is supposed to be minimally hungry, especially if it is done only selectively
Q: What were the major improvements that helped with single thread throughput?
A: In some cases we had to clean up the code; in some cases we had to improve the memory layout in ways that were quite beneficial.
Q: Is the profile guided compilation speed up general for other kinds of events?
A: We don't know yet, but this is a substantial gain!
C: We really need to look into ApplyCuts again :-(
Q: How fast is the doubling in these plots?
A: It's almost Moore's Law scaling
Q: What challenges did you have when trying to debug the multi-threaded environment for safety?
A: We have reproducibility tests (start from the same place with the same seed and get the same result, and seed specifically event #X with the required seed during an MT job and get the same result). There's no guarantee of the reproducibility of the result of merging.

Steve's talk on Making MT Happen

C: The lazy thread initialization is only valid in the case of a pre-defined (before the job) number of threads. You cannot add a thread part-way through a job without causing potential problems (similar to a vector re-allocation and move in memory, such that pointers would be invalidated). There does not seem to be a nice way around this.

Q: Could you move to a state-passing model?

A: This was the choice based on performance and ensuring lock-free code, but we have to look more into these. It is not a trivial change, and might require changes to more interfaces, which were kept untouched for most of this work.

C: (Makoto) We intend to work a bit more on allowing threads to join mid-job

C: CMS also wants to see the Workspace parallelism model for simulation

C: There is a limit to the number of libraries with thread-local storage that can be used

Q: This limit seems quite high… really?

A: Yes.

Q: How do you avoid the context specific initialization for each thread

A: We need to do this in order to avoid initializing G4 for all the threads, some of which might not be running simulation

Chaos ensues. We will follow the issue of Workspaces and TLS offline, and we will need to understand all the issues here.

Jason's talk on Upgrade Issues

Q: Do we understand correctly that the beamspot is two things, the boost and the time distributions?

A: Yep, those are the issues

Q: Can we treat them as 1D additions?

A: Except that time and z position *may* be linked, yes

Q: Do we store time in the LAr hits?

A: Yes, but it's not obvious that we are modeling it all correctly

C: FTK is not something we should push aside, because ITK is considering an "FTK++" now

Q: What code does the FTK simulation run?

A: It's in athena, but it is separate packages of course. The emulation is stand-alone still, but the input is from athena, and there is an athena merge step. It is done within prodsys though.

C: The fast simulation is truth-seeded

C: Super-computers do have 35 GB of memory on a compute node :-)

Zach should email Paolo an FTK simulation talk

C: We could try moving the 256 jobs through Yoda

A: It's not obvious that this is useful -- people seem to be happy with the current setup. But such a setup might be able to help protect us against the next version of this which could be even more resource hungry.

C: We should work on using truth-seeded tracking for the upgrade

Q: Is the 35GB on top of the normal job?

A: Yes

C: As a whole-node job, this would work; we could try this as an MP job, as the FTK patterns should not change and can be shared.

Zach's talk on Other Stuff

Send Peter a pileup file

For simulation build can kill ChkLib, AtlasShift, AtlasDCACHE at least! And probably others...

Can use TBB tasks for digitisation - don't need to worry about low level thread stuff.

Digitisation needs re-profiled and there could be a lot of benefit to making the data structures better suited to vectorisation.

Wednesday, November 11. **Analysis**

Introduction - Nils

RootCore is liked by users - need to provide the same advantages to users if there were to migrate to an athena-ish solution. Sociological considerations are not trivial!
Athena has been built ~monolithically and has many dependencies that make changing one single piece very difficult at the moment.
Some progress in decomposing - AthAnalysisBase and AthSimulation, but need to go alot further.
Can we make athena(MT) do *simple* things more easily?
David - how deep do we want to go? e.g., can we design Athena to run without StoreGate?

ROOT7 - Philippe

David Malon: When is ROOT 7 going to need to be ready?
Not clear yet. To be discussed… (2018-2019?)
Charles: Which C++XX version?
C++14 or C++17.
Zach: Interfaces from RootPy?
Philippe will look at it.
Additional libraries (RooFit, RooStats,...) will only be migrated later. With the help of their developers…
Interface design workshop? With invites from a number of areas?
Unit testing? Will have to introduce a lot of it for ROOT 7's new classes.

Analysis Frameworks - Steve

Dual-use implementation (switching out code with pre-processor flags). <-> Using good interfaces that allow us to switch out implementations.
Integration with PROOF? Not at all clear yet. Try to design the core of the framework such that external schedulers could be made use of.
Memory concerns? Multithreading? Not necessarily too worried about this.
Discussion about "the ASG analysis framework". Now up for review in ASG…

Release Building and Distribution - Attila

Various strategies for providing releases on OS X: click installers, brew, cvmfs, VMs, etc. Need to see what would be really useful for users and not an excessive load.

Wednesday, November 11. **Infrastructure, Tools and Education**

Static Checks - Scott

Does thread safety checking work with inheritance? A - it could do this.
Could classes that are going to be concurrent inherit from a different base (ConcurrentAlg)? Seems like a good idea and we should consider it.
Chris - what is the level of checking? C++ const or really, really thread safe…? CMS distinguish between thread friendly and thread safe.
Don't care about every method - only about ones that might be called concurrently. Marking these up individually might be quite tedious.
CMS use a 'thread guard' mark up to help with this.

Push this up to Gaudi? "**Yes, please!**" says Marco.

How do you enable this? Shared library with links to the compiler package, use with the "-f" option to gcc.

Build overhead not measured, but doesn't seem large (all linear scaling).

Will it work with distcc? If the sharable library is available to distcc.

Benedikt - this would be really good to push upstream!

Stefan - Should we enable it by default? Not really - for now better to focus on the areas we need to be thread friendly/safe.

Work on the inheritance of checking flags - this is the way to ensure that necessary code is checked.

Exceptions are necessary, but only for people who really know what they are doing!

We should enable the name convention checker in one devval build and see how many warnings we get. Start with these as minor warnings. For the moment don't worry about class naming.

Zach - how to get renaming as a high enough priority? Philippe -clion tool?

Dynamic Checkers - Stewart

Lightweight LCG builds would definitely help (also with analysis builds as noted earlier)!
This has been discussed with SFT.
Should LCG nightlies build with SAN switched on? Not sure if we can enable/disable the sanitiser easily?
Valgrind-Helgrind also gives many false positives for TBB (does not understand atomics).
Would be good to measure code coverage more systematically.
Scott - has almost all clang failures worked out.
SAS - static analysis suite runs in clang (developed by CMS). Would be useful to compare and contrast.
Our clang nightly uses a gcc LCG build.
We should have **more unit tests**!

Eric - How do we stop bad code from even getting into the release? At least have “check” builds available to developers before they commit their code. Code reviews? Note q-tests don’t necessarily run code path that people just developed (so “standard” checks are not sufficient).

Software Quality - Stefan

For subsystem workbook, recognise there are not that many new developers per year (<5…?).
Muon and simulation tutorials - a lot of effort and not successful in attracting new software development effort.
Teaching new people to do things is time consuming. Future benefits, but in a context of high current pressure it’s hard to find the time to invest.
Have a software development test before we allow people to commit code. Should not be too heavy, but underline this is important. (Marketing - “Make best use of your time and our time.”)
Organised (week) tutorials do also provide good reference materials.
Ben - encourage people to take good standard courses in programing.
Have made bad mistakes in the past - e.g, monitoring code (and we suffered for it!).
Get newcomers to write documentation.
Need an ***Education implementation task force****?*
Teams - Argonne, Orsay, Edinburgh, …
Could harvest much information about active developers and teams from SVN + SRLs, etc.
Beware of out of date information and generating *more* work.
Unit tests will be easier if objects are instantiated in a working state. **gmock**. Nils could help make this available in RootCore.

Version Control Systems - Graeme

Q: Need to tag a specific package might still be needed.
A: Can be avoided by branching

It’s possible to make diff on specific location between different revisions if one wants to check what changes for a specific package.
Q:How the repository should be set - shall we have a separate git repo per package ?
A: The whole release in one git repository
Q: How to deal with analysis releases
A: Just use a different branch

There is git subtree which can be considered next to submodules eventually.

Time plan: not moving in 2017 but this is something for LS2

Jenkins Evaluation - Alex

Hope to get more help from CERN IT for Jenkins in next few weeks. Need to make sure that priority is realistic and we help if necessary.
SFT, CMS, LHCb and ALICE all have experience with Jenkins - puppet templates.
Can Jenkins delegate responsibilities? Yes.
Marco - Jenkins works much better with a single code repository (*a la* CMS) than multiple ones.

HSF - Benedikt

Should we have HEP specific training? Yes, but after generic training.

Hackathon

Conditions Access Discussion

- Quantising access on LBs seems like a good idea
    - Conditions with sub-LB granularity should store this “internally”
    - Internal structure should be simple
        - Container of objects which are valid for each sub-slice of the LB
        - Should be trivial to map from timestamp(?) to sub-object, so that the conditions data service can generically send the correct data back to each client
- Event context holds all the information the CondSvc needs to identify the correct objects for a client.
- Should try to get rid of callbacks - conditions data service should know how to generate calibrated objects (configuration - see FFReq report fig. 8).
- As calibrated objects are returned, clients should not need to cache values
- Do we have a detector store per LB?

    - Or just one large one?
- Certainly try not to have a DS per finest granularity object (forces the highest cadence on all conditions!)
    - See above for an idea of how the structure could work
- Run a garbage collector on the DS - get rid of conditions over a watermark (LRU).

Open Question List

1. How to associate a sequence with object conversion
2. Measure the size of the detector store of a real job - get an idea of volume per event, LB. etc.
3. Basic information to add to EventContext that can be shared between Atlas and Gaudi

Tracking Challenge

- Conformal mappings don't work in ID particularly at Hi_LHC because of material/mag field effects on ideal trajectory
- Seeding and following
    - 2-3 hits used to build a road, search inside road
    - Alternative follow track layer by layer and do a linear search on layer
    - Progressive filtering
    - Deterministic annealing would be perfect but costs too much
    - Losing track to combinatorics in the core of a dense jet is much more harmful than in a minbias region
- Fast Track sim 100K tracks in 20secs
    - Isabelle: should we use real data using the reco as reference
- Not looking for a track fitter, only pattern reco. Scoring may be based on hit count. Weight more pixel than sct because of impact parameter resolution
- Completeness of a track (no holes) is very important. Assume no bad channels
    - Should we have holes coming from hits below threshold to teach ML to deal with innefficiency?
- Add hardware info to hit:
    - energy deposition and cluster widths.
- Magnetic field map as input, as well as particle truth/reco momentum for training.

- Metric: Clustering/coloring problem vs regression
    - coloring may need weights of hits found/missed
        - scale weights by pT
- Isabelle:
    - ML really bad at trigonometry. Must turn all potential useful derived quantities into a feature vector
    - Many prizes:
        - 200mu pileup identify minbias tracks, most important task
        - find needle in haystack. Use “sexy” channel
- Short term: Rebecca triplet classification project
- What to present @CTD2016
    - standalone FastSim application (Andy)
    - status of challenge

Friday 13

Conditions Redux

- what is the cost to schedule all the “conditions callback” algorithm?