

Links:	2
January 31, 2020	2
January 10, 2020	3
January 3, 2020	4
November 22, 2019	4
November 1, 2019	5
October 18, 2019	6
August 23, 2019	6
August 9, 2019	7
July 15, 2019	7
June 21, 2019	7
May 31, 2019 (FastCaloSim Technical Discussion)	8
May 17, 2019 (FastCaloSim Technical Discussion)	8
May 3, 2019 (FastCaloSim Technical Discussion)	9
April 26, 2019	9
April 12, 2019	10
April 5, 2019	10
March 29, 2019	11
March 8, 2019	12
March 1, 2019	13
February 22, 2019 (FastCaloSim technical discussion)	14
February 15, 2019	15
February 1, 2019	16
December 7, 2018	16
November 20, 2018	16
November 9, 2018	17

August 16th, 2018	18
How to run FastCaloSim standalone	19

Links:

ATLAS-CSI Workshop: <https://indico.cern.ch/event/742090/>

4th US ATLAS HPC Meeting: <https://indico.physics.lbl.gov/indico/event/955/>

Live notes:

https://docs.google.com/document/d/1o34L_N83kSRghM10b8Co7D6tBJAI22RzfZODmwwljv0/e/dit#heading=h.9r9mqqa1ymzd

January 31, 2020

Attendees:

- Meifeng Lin
- Kwangmin Yu
- Charles Leggett
- Vince
- Walter Hopkins
- Tulay
- Zhihua Dong
- Doug Benjamin
- Paolo
- Weitao
-

Notes:

- Zhihua Dong - FCS CUDA port
 - Analyzed the performance of current CUDA implementation (multiple event, multiple particle, speedup not ideal).
 - Large overhead with cudaMalloc in LoadFH. Can allocate reusable GPU memory -> already did this in other parts of the code
 - Tulay: ALICE and LHCb are working on something similar (<https://indico.cern.ch/event/867891/>)
 - ATLAS is working to reduce the size of parameterization
 - Would be good to test the code with different simulation data, in particular those consistent with Run 3

- Do we have large data files? Current data file only runs for less than a minute. => Could duplicate the data files
- Vince - SyCL port
 - DPC++ convert tools doesn't seem to be the best strategy
 - Best to look at the original CUDA code, and convert each piece at a time
 - Using Codeplay's ComputeCPP implementation of SyCL
 - Fewer dependencies than DPC++
 - Can target Intel and NVidia (w/ PTX)
 - Easier(?) to get into ATLAS software framework than DPC++

January 10, 2020

Attendees:

- Meifeng Lin
- Kwangmin Yu
- Tulay Donszelmann
- Vincent R. Pascuzzi
- Walter Hopkins
- Doug Benjamin
- Zhihua Dong
- Michael (ATLAS, CERN, FCS expert)
- Charles Leggett

Notes:

- Meifeng: Background introduction for Michael
- Zhihua: ShapeValidation and simulation, what is needed in real simulations?
- Michael: in real case (within Athena framework), will also run ShapeValidation
 - In the new model, the number of hits is not fixed.
 - Major changes one or two months ago.
 - Hits can be made predictable.
 - There are standalone versions for the new FCS.
- Zhihua: question about multiple-particle data files provided by Tadej
 - Michael: hitchain is the most (only?) sensible part to put on GPUs
 - Optimization of Histogram class for HitChain
 - 5000 events in typical t-tbar simulations
 - ATLAS is working on multi-threaded version on CPUs
 - Each event needs 0.5MB, regardless how many particles
- Vince SyCL's progress
 - Use Intel conversion tool
 - Curand is not migratable
 - Printf not supported in SyCL kernels

January 3, 2020

Attendees:

- Doug
- Charles
- Meifeng
- Kwangmin
- Walter
- Zhihua
- Paolo

Notes:

- Doug - Tadej's dataset coming with the standalone FCS code is t bar event samples, one of the highest occupancy samples of Run 2, Standard luminosity
- Paolo - Run 4 simulations will have more computational complexity
- Zhihua - Presented progress in parallelizing event loop
 - half of time is spent in data transfer
 - not much gain compared to parallelization over nhits/multiple particles
 - std::map takes 3x longer than GPU simulations; need to change the data structures if we want to change the implementation
 - Will do some performance profiling to see if performance can be further improved
- Validation -
 - Charles-can ask physics validation people to see what they would suggest
 - May ask Vince
 - There are very specific procedures for benchmarking
 - What histograms to look at that need to be validated
- FastCaloSim - portability, how to coordinate for the new CCE project
 - Doug - Is FCS too specific for ATLAS?
 - Charles - Doesn't matter, as we are investigating the portable solutions. FCS is farther along than ACTS
 - Vince is using SyCl to CUDA conversion tools for the FCS-GPU code

November 22, 2019

Attendees:

- Meifeng
- Zhihua
- Kwangmin
- Vince
- Charles
- Doug

- Paolo

Agenda:

- FastCaloSim-GPU status - Zhihua Dong
 - Kept working on kernel fusion/reducing number of kernels
 - Random numbers on GPU - over-generate RNs at the beginning to reduce kernel launch overhead (a few percent improvement)
 - Plan to parallelize the particle loop may reduce the number of kernels - just need to make sure all the data are available on GPUs
 - Got synthetic data from Tadej. Would be nice to have real simulation data.
 - Doug offered to find the data (typical physics events with different multiplicity) needed
 - Doug: need to work on agreement with BNL to share ATLAS simulation data with CSI for the project
- Scaling studies on Cori-GPU - Charles Leggett
 - Cori-GPU: <https://docs-dev.nersc.gov/cgpu/hardware/>
 - Timing data: (10,000 events) <https://docs.google.com/spreadsheets/d/1dvkSwjHtOfh1m239nrokbN8FswJ98nwoXV7SXAKuAw/edit#gid=1061875772>
 - Event loop - time spent for simulating 10K events
 - Host Chain A - CPU simulation time
 - GPU Chain B - GPU simulation time
 - Total time - overall run time including I/O

November 1, 2019

Attendees:

- Meifeng
- Zhihua
- Kwangmin
- Vince
- Charles

Agenda:

- FastCaloSim CUDA port:
 - Zhihua is working on improving the CUDA kernel performance
 - Needs multi-particle input files - to follow up with Tadej
 - Needs algorithm improvement
- SyCL experimentation - Vince
 - Set up the code to run successfully on laptop and BNL IC
- Benchmarks for the paper (planned for PASC2020, deadline December 15, 2019)
 - Has access to Cori GPUs

- Need to sort out Summit access

October 18, 2019

Attendees:

- Meifeng
- Zhihua
- Kwangmin
- Vince Pascuzzi (LBNL)
- Walter Hopkins

Agenda:

- FastCaloSim CUDA port:
 - Zhihua further improved the performance of CUDA kernel since the GPU Hackathon (1000 hits to get GPU advantage over CPU -> 800 hits to get GPU advantage over CPU); working on kernel merge and outer loop parallelization (multiple particles, and/or multiple events)
 - Different events may have different run times, which could affect load balancing and cause thread idling
 - Kwangmin: could use a scheduler to fetch events from a pool of events
 - Not using cuda stream yet; Could be used for the event-level parallelization
 - Next steps:
 - Reduce kernel launch overhead
 - Work on multiple event parallelization
- Will look into compile error with the latest git commit
- Vince: LBNL has AMD and Intel GPUs, will look into syCL, hipSyCL and other variants

August 23, 2019

Attendees:

- Meifeng
 - Kwangmin
 - Charles
 - Doug
 - Zhihua
1. Doug presented his slides on his experience with FastCaloSim-GPU
 - a. General comment: Need to switch to a more modern compiler; Compilation failure with newer gcc versions
 - b. Could reproduce the performance results CPU vs GPU (5X speedup)
 - c. Results consistent within errors
 - d. Next step: create a matrix of compilation
 2. Charles: could make the code with gcc 8 with Doug's patches; still need the input files to run. (Doug to give instructions)

3. Zhihua: Histogram on GPUs? Non-trivial as it is a reduction. Doug suggested checking to see if ROOT already has GPU histogram functions.
4. GPU Hackathon in September: will focus on performance improvement of the CUDA port. As an aside, will also look at portable solutions. Charles suggested starting with Kokkos.

August 9, 2019

Attendees:

- Meifeng
- Kwangmin
- Zhihua
- Charles
- Tadej

1. Kwangmin Yu presented his progress with porting one of the FastCaloSim functions to GPUs. Code has been merged into the FastCaloSim-GPU github repo
2. Zhihua Dong presented his performance and scalability results on the BNL IC.
3. Tadej is working on a script to facilitate validation and verification of FCS. Will send code when ready.

July 15, 2019

Attendees:

- Meifeng
- Kwangmin
- Zhihua
- Doug
- Charles

- 1.

June 21, 2019

Attendees:

- Meifeng
- Zhihua
- Kwangmin
- Charles
- Torre

- Sergey Panitkin

FastCaloSim GPU progress:

1. Kwangmin: working on TFCSValidationHitSpy
2. Zhihua Dong: Already got three major functions to run on GPUs; Even without HitSpy GPU code, got about 5X speedup on P100 (10000 events, wallclock time decreased from 7 mins to less than 2 mins)

May 31, 2019 (FastCaloSim Technical Discussion)

Attendees:

- Meifeng
- Zhihua
- Kwangmin
- Walter Hopkins
- Charles

Agenda:

- GPU porting status and issues
 - Dong: need to figure out the best strategy to handle which data to load to GPUs
- Git repository
 - We can't have the git repository since we only have the CERN lightweight
 - Charles to look into getting the repos from CERN directly

May 17, 2019 (FastCaloSim Technical Discussion)

Attendees:

- Meifeng
- Zhihua
- Walter Hopkins (ANL)
- Charles Legget

Agenda:

- GPU porting status
 - Zhihua Dong: Questions about random numbers for the simulations - RNG to coordinate mapping; Will need people who are familiar with the algorithms in the code.
 - Charles will help look into it, after Zhihua sends him the code.
- Git access
 - Need full git repo of the standalone CPU version of the code
 - Meifeng followed up with Heather, Hasib and Tadej to see if we can get the full git repo
- Frontier architecture and implications for our next steps

- Not just Frontier, Aurora has the same issue
- Intel may also provide translation tool (AMD has HIPify)
- Sensible to focus on just one technology (CUDA) right now
- FCS could be a good testbed, to see how we can translate CUDA code into other codes
- Will start informational presentation series focusing on ECP architectures and programming models and tools.

May 3, 2019 (FastCaloSim Technical Discussion)

Attendees:

- Meifeng
- Zhihua
- Kwangmin
- Charles
- Doug
- Walter

Minutes:

- Doug is applying for ADSP time at ALCF (Cooley), also would like to run FCS-gpu through PANDA.
- Idea is to see if PANDA can handle the GPU code (compiling, running, error handling, etc.)
- No push access to the CERN repo with lightweight accounts; Temporary solution is to create a github private repository to do a one-way merge with the main repo.
- Should also create a wiki page on the github repo with compiling and running instructions

April 26, 2019

Zhihua presented his current progress on FastCaloSim gpuize.

Q1: Concern about increasing number of hits in reduction

Zhihua: The GPU code will manage the increase of hit. If the hit number exceed the shared memory capacity, then multiple blocks will be used.

Kwangmin: The reduction algorithm has $O(\text{long } N)$ complexity, it can handle easily the increase of hits

Q2 : GPU memory concern

Zhihua: The memory used in the current example is not huge. So the GPU memory deficiency is not concern at this moment

Kwangmin: We have plan to support multiple GPUs in a compute node. So this will be mitigate the memory deficiency when the problem size increase.

April 12, 2019

Attendees:

1. Meifeng Lin
2. Zhihua Dong
3. Kwangmin Yu
4. Doug Benjamin
5. Charles Legget

Agenda

1. FastCaloSim GPU Porting Progress
 - Zhihua has finished implementing the data transfer (20,000 calo cells to GPUs, took a few seconds). Still need to verify the correctness of the region data.
 - Data structure not completely flattened. Changed pointers into arrays so that they are consecutive in memory.
 - Next step is to write CUDA kernels for the actual simulations (after the correctness of the data copy is verified, it should be straightforward)
 - How many random numbers are needed? Still need to be figured out.
 - Discussed a bit more about validation between CPU and GPU, and between GPU runs (probably only need statistical reproducibility)
 - 36 ms / event on CPU for the current workload; small test to verify the code.
 - Doug - ATLAS has 6 GPU nodes (full reservation) at BNL. Can be used for testing.
2. Management of two development branches (CPU and GPU); access to CERN repo
 - a. Doug: will look into possibly a lightweight account at gitlab.

April 5, 2019

Attendees:

1. Meifeng
2. Zhihua
3. Charles Leggett
4. Kwangmin
5. Tadej
6. Doug Benjamin

Agenda:

- Changes of the new FastCaloSim code - Tadej

Tadej : Order of the hits needs to be changed. Don't expect it to change further in the future.

Doug: FastCaloSim is improving accuracy of the simulations, so the code is likely to change. GPU work needs to follow the change to make sure physics results are consistent.

Tadej: impossible to give external access to CERN repository; may make the repository public soon.

Doug: what about collaborator accounts? May not have full access

- Update on GPU porting - Zhihua

Working on the data transfer to the GPUs (either cell by cell, or the whole geometry)

Validation: results will only be statistically valid (Doug: that's what is being done anyway).

What about run-to-run comparison? May need to copy all the

- All other matters

March 29, 2019

Attendees:

1. Meifeng
2. Kwangmin
3. Zhihua
4. Charles Leggett (LBL)
5. Torre
6. Tadej

Charles - discussed needs for accelerated code. Big push to use Kokkos. Kokkos can come to us if we ask but we need to have a kernel.

Torre - well said. need to document this.

Meifeng (Post-meeting) - Charles email summarizing the motivation:

I am very interested in following this project on several levels. First of all it is one of the few uses of accelerators in ATLAS, and we need to find some good testbeds to experiment with on the new line of HPCs like Summit and its successors. I will be very interested to see how challenging it is to port something like this to CUDA, and what sort of algorithmic changes need to be made. I also understand that it is a standalone project (please correct me if I'm wrong!), and as such makes an excellent candidate as a so called "kernel" that we can bring to various hardware (and software) vendors so that they can help us optimize it for their hardware/software. Right now there seems to be a big push in the DOE labs to adopt Kokkos, and we've been told that they would be happy to offer us one on one tutorials to help migrate code. Also, NVidia has just signed a big contract with lbl to improve support of their compiler

and openMP on Perlmutter, so we can leverage that as well if we have a nice piece of self-contained code.

Kwangmin - presented "FastCaloSim Profile" (slides) which was sent to the mailing list

-Will you use nvcc for the whole code? No, will compile kernel and link due to complexity of the code.

- How much data is needed on the GPUs? Not much, just the geometry (~25 MB)

Zhihua - presented GPU porting strategies (slides)

Flattening of some data structures

Tadej - new code uploaded to BNL IC. new version seems stable.

Meifeng - need a git repository to track changes and merge new updates; could host locally at BNL.

Charles - may be able to let CSI people access CERN repository. Will look into it.

Charles - how to do validation?

Tadej - same as validation GEANT4 vs. FastCaloSim.

Zhihua - what about random numbers? May need to use curand.

March 8, 2019

Attendees:

1. Meifeng
2. Nick
3. Zhihua
4. Tadej
5. Sergey Padolski

Agenda:

- Update from Tadej on standalone FastCaloSim

Changed the code to make it easier to run (file path etc.). Now has access to BNL IC. Can copy code to BNL.

From Tadej's email update on 3/7/19:

I updated the tarball on my AFS. I now try to better handle missing files. Also you can use

-DINPUT_PATH=/path/to/files where path is the base path containing folders like

BigParamFiles CaloGeometry InputSamplesProdsysProduction ParametrizationProductionVer06

Note that we currently have some problems with the code and I can not update to latest Athena (locally it worked for me). I will delay the creation of Docker image until the "master" version of the code is working as expected, although most of things are now in place.

This new codebase should be much easier for you to use (no need to manually change paths).

Also, I now have a BNL account so I can maybe start copying the code directly to some defined place (but you will have to show me how).

Zhihua will try to create a shared directory on BNL IC. Already has a directory with all the input files.

Tadej: can create the correct directory structure for the input files.

- Update from Zhihua on performance profiling and initial porting effort

Still analyzing the performance breakdown of the code.

Discussed more about how to approach the GPU porting. OpenACC may be an option.

Discussed PGI compiler and possible issues with ROOT.

Meifeng - may not be an issue. Success story from last year's GPU hackathon team - SBNfit. Will try to make contact with them.

- Update from Kwangmin Yu

From Kwangmin's email: "I succeeded in compiling and running the code. As we discussed, I am taking "TFCSValidationHitSpy::simulate_hit" function.

I am analyzing it. The function has 193 lines."

- Compute resources for this project

This project has an allocation on BNL IC now - One node for 3 months. New nodes available on IC, not as busy as before. "Debug" queue doesn't count against allocation.

- All other matters

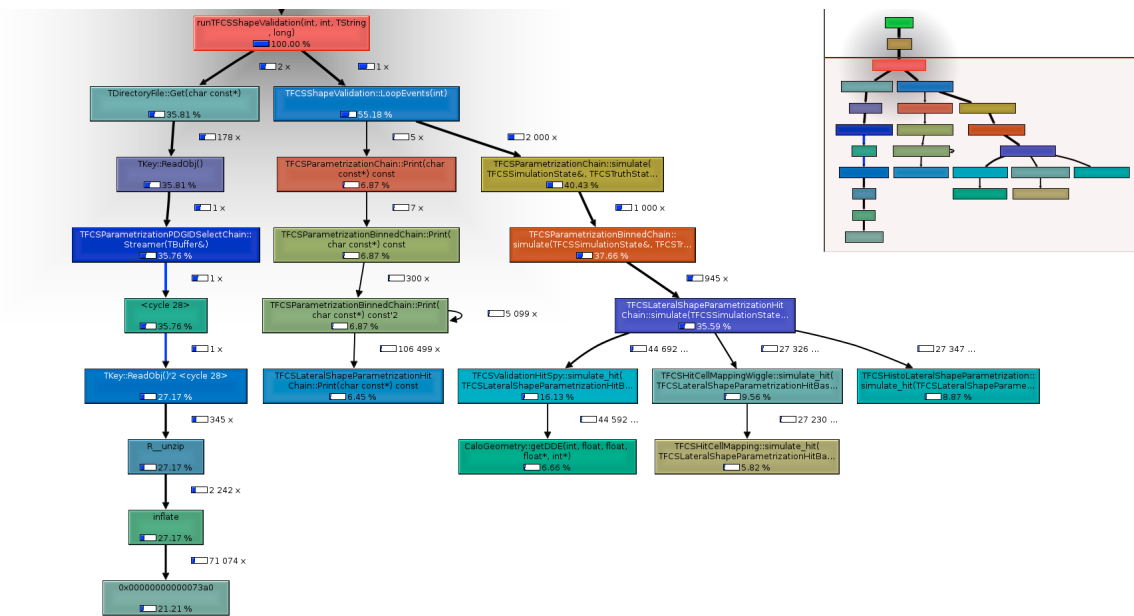
March 1, 2019

Attendees:

1. Meifeng Lin
2. Kwangmin Yu
3. Sergey Padolski
4. Zhihua Dong
5. Nick D'Imperio

Agenda:

- Update on FastCaloSim performance analysis (Zhihua)



Manual timing consistent with valgrind callgraph

1/3 time spent on I/O; simulation part loops over events; every event 36 ms, 10,000 events would be 6 minutes; data dependencies are nested; not clear if these events are independent.

C++ code, deeply nested, porting to GPUs may be difficult.

Try OpenACC? First try to compile the code with the pgc compiler without any OpenACC pragmas

Would dependence on ROOT cause any problems?

- Update on FastCaloSim container (Tadej)

May be ready next week

- Next steps

- Understand the data structure and data movement in the code
- Get a better understanding of the algorithms used in the code; Meifeng to send related literature
- Try out OpenACC

February 22, 2019 (FastCaloSim technical discussion)

Attendees:

1. Meifeng
2. Hasib
3. Zhihua
4. Tadej

1. Standalone compiled FastCaloSim code finally runs successfully on the BNL IC, with the code from Tadej and input files from Hasib.
2. Zhihua has produced a call graph (using valgrind callgraph) for the sample run (energy 65 GeV, 50,000 hits); 30% IO; 50% simulate.
3. Meifeng: what programming model to use to port/optimize the code? Ideally should be performance portable.
4. Tadej: parallelization across different algorithms in Athena (each algorithm is serial); FCS should be thread safe.
5. Meifeng: how would a GPU version of FCS be integrated into Athena? Tadej: is being experimented. May not be important how the FCS code is integrated. Should just use it as a prototype.
6. Tadej: almost have a docker image ready. FCS will be open source soon, as Athena is now open source.

February 15, 2019

Attendees:

1. Meifeng Lin
2. Kwangmin Yu
3. Zhihua Dong
4. Doug Benjamin
5. Nick D'Imperio
6. Walter
7. Tadej

Agenda:

1. Issues with the standalone FastCaloSim code

Zhihua still got Segfault with Tadej's new code, possibly related to missing input files.

2. Next steps? Should we revisit the possibility of doing the performance profiling/porting work within the Athena framework?

Tadej has a new tarball, and asked Zhihua for log files with the error. Zhihua will try the new tarball.

Doug suggested having a docker image, then converting it to singularity for BNL IC. Linking to external profiling tools or building the profiling tools in the image is also possible. Usually <10G docker images on DockerHub. Larger docker images are shared through CERN's distribution system.

May take 2 days to build the docker image with necessary input files.

Tadej said they run the code in CI, so all the dependencies are fairly self contained.

Doug commented that working within the Athena framework is still not preferable due to the large overheads not related to FastCaloSim. Once we figure out a GPU port within the standalone FastCaloSim, it can be absorbed into Athena.

We also agreed to have more frequent technical discussions to go over immediate issues. Tentatively the time has been set at 10am ET/4pm GVA on the Fridays when we don't have a WG meeting.

3. We should think about having a deadline for deliverables. Can we aim for a SC paper, due early April?

SC19 may be too ambitious. We should aim for CHEP in November 2019. We can also try to have an SC19 poster, which is due in July.

4. Computer access at BNL/CERN. It would be probably useful to create BNL computer accounts for the ATLAS people who are most actively involved in this project.

Tadej has been given instructions to apply for a BNL Guest ID. Once that step is complete, he can request an account on the BNL IC.

February 1, 2019

Hands-on session between Hasib and Zhihua to debug FCS on BNL IC. With Hasib's help, the previous errors Zhihua Dong saw went away. The code appeared to run (only to exit with another error)

December 7, 2018

We went over the presentations by Tadej and Serhan.

<https://indico.cern.ch/event/779677/>

November 20, 2018

Attendees:

1. Meifeng Lin
2. Kwangmin Yu
3. Zhihua Dong

Short meeting. Discussed whether it makes sense or is plausible to try to do some profiling with the interpreted version of FastCaloSim. Conclusion: might be too cumbersome without spending

a lot of time trying to understand the code. Will wait for the compiled version while looking into Athena.

November 9, 2018

Attendees:

1. Meifeng
2. Kwangmin
3. Hasib
4. Heather
5. Marilena
6. Sergei
7. Torre
8. Walter
9. Zhihua
10. Doug

Agenda:

1. Status of FastCaloSim compiled version

Hasib: problem was with some complicated interdependencies on Athena; getting close to having a compiled version. May take a week or two to finish one macro. A student (Tadej Novak) is working on it. FastCaloSim group has 10-12 people. Hasib is the convener.

Atlas is doing first-round of FastCaloSim validation. FastCaloSim has been developed for many years. Need to do physics validation before using it in production. AF2 is the old version.

Meifeng: Would be nice to have some input/output data to validate the code once we start porting and optimizing it.

Hasib: do standard validation; compare it with Geant4; shower shape/distribution etc...

Heather: can discuss it at the next meeting

Doug Benjamin: can get time on BNL IC through the ATLAS project. Also have ALCC allocation on Titan (80M hours); CSI to send estimated resource needed to Doug.

2. **Should we discuss the alternatives if a compiled version turns out to be too cumbersome? Would it be sensible for us to work directly within the Athena framework?**

Would be a lot of work. Athena does a lot more than FastCaloSim. Can revisit this approach in a month if the standalone version doesn't come through.

Heather to follow with Serhan about the athena-based profiling.

3. **Any work we could carry out to enable Fast Simulation on the new NERSC supercomputer Perlmutter?**

Could include CSI people as experts if we wrote a proposal

4. **For physics analysis for Run II papers, we need QCD background with statistics x1000 larger than what we have now (JZW samples). ATLAS fast simulation may not be sufficient to achieve this. Can we discuss alternative fastsim approaches (Sergei/Doug/Walter)?**

Will discuss in ATLAS simulation meeting.

Friday November 23 is a US holiday. Will try to find another day in that week for next meeting.
Meifeng to send follow-up emails.

August 16th, 2018

Attendees:

1. Heather
2. Kwangmin
3. Abid
4. Meifeng
5. Nick

Agenda:

1. **Aim to identify concrete objectives by the end of September**
2. Deliverable: CPU breakdown, what fraction is FCS, does offloading FCS make sense. What can GPUs win us?
3. TimeScale:
 - a. concrete objectives by end September
 - b. deliverable by end of December
4. Effort: ?

Discussion:

1. GPU service in plain athena called APE -- work is ongoing for multithreaded. Start simple with multithreaded.
2. Check how the standalone running of FCS compares to running it within athena
3. Scaling could be interesting to study: small and big scaling.
4. Can we run this algorithm at all on GPU? Substantial editing would be required.
5. Use OpenACC
6. Discussion about large amount of data needed for GPU. If it's static, then it's not necessarily a problem. 32 GB of GPU memory.

Action Items:

1. Investigate the effort from ATLAS side
2. Access to the code -> fork the repository
3. Know how to compile the code [Charles, Illya]. Configuration how to run the code: one that's small, one that's big
4. Profile the code
5. Technical assistance
6. Deadline first week of September: goal is to have compiled and profiled the code.

How to run FastCaloSim standalone

We have put a full copy of athena on afs at BNL:

/afs/usatlas.bnl.gov/software/downloads/gitrel/21.0. This is the full code base, so please treat it confidentially.

1. checkout the following Athena packages:

1) ISF_FastCaloSimEvent

2) ISF_FastCaloSimParametrization

from: https://gitlab.cern.ch/atlas/athena/tree/21.0/Simulation/ISF/ISF_FastCaloSim

(At BNL, instead you will need to copy the packages from here:

/afs/usatlas.bnl.gov/software/downloads/gitrel/21.0/athena/Simulation/ISF/ISF_FastCaloSim)

2. Make the following directory structure <workdir>/athena/Simulation/ISF/ISF_FastCaloSim/ (the following packages needs to be checked out in this location). <workdir> can be pretty much any location, but the relative path locations are important.

> checkout <https://gitlab.cern.ch/atlas-simulation-fastcalosim/FastCaloSimAnalyzer>

> checkout <https://gitlab.cern.ch/atlas-simulation-fastcalosim/EnergyParametrization>

> cd FastCaloSimAnalyzer/

> source setup.sh

```
> cd macro/  
> root  
  >> .x initTFCSShapeValidation.C  
  >> .x runTFCSShapeValidation.cxx
```

The runTFCSShapeValidation() is an example to run the simulation infrastructure:

http://atlas-project-fastcalosim.web.cern.ch/atlas-project-fastcalosim/TFCSDoxygen/de/ddc/run_t_f_c_s_shape_validation_8cxx.html#aed0f5ef643da93f74b8f032779ff37cf

The Base class for the infrastructure:

http://atlas-project-fastcalosim.web.cern.ch/atlas-project-fastcalosim/TFCSDoxygen/d7/da1/classes_t_f_c_s_parametrization_base.html