Framework support for Accelerators

Sami Kama

Introduction

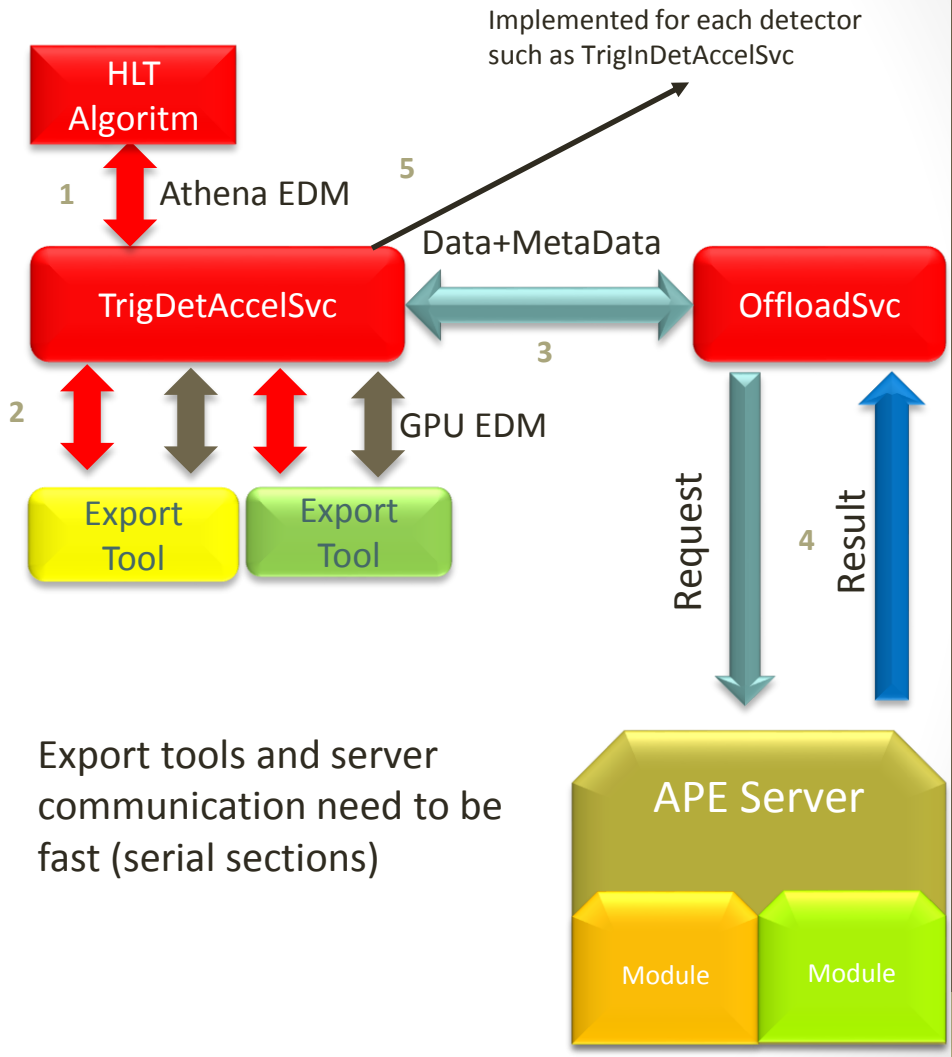
- Current Status
- Future Accelerator use modes
  - Symmetric resource
  - Asymmetric resource

Offloading now

- Accelerators are exposed to framework through use of OffloadSvc and APE server
- Athena algorithms prepare necessary data and do an offload request to OffloadSvc
- OffloadSvc sends request to APE server.
- Server executes the request on data by selecting appropriate module, and returns the results to OffloadSvc
- OffloadSvc passes response to algorithm, which convert results back to Athena EDM

Example HLT Demonstrator

1. HLT Algorithm asks for offload to TrigDetAccelSvc
2. TrigDetAccelSvc converts C++ classes for raw and reconstructed quantities from the Athena Event Data Model(EDM) to GPU optimized EDM through Data Export Tools
3. Then it adds metadata and requests offload through OffloadSvc
4. OffloadSvc manages multiple requests and communication with APE server
5. Results are converted back to Athena EDM by TrigDetAccelSvc and handed to requesting algorithm



Export tools and server communication need to be fast (serial sections)

Current status

- Multiple Athena's and AthenaMP are supported
  - Some glitches after forking, possible race condition in Yampl
- OffloadSvc can dump offload requests and responses for re-players.
  - Needed for quick development and validation of accelerator algorithm
- Works fine for multi-process frameworks!

Offloading in future

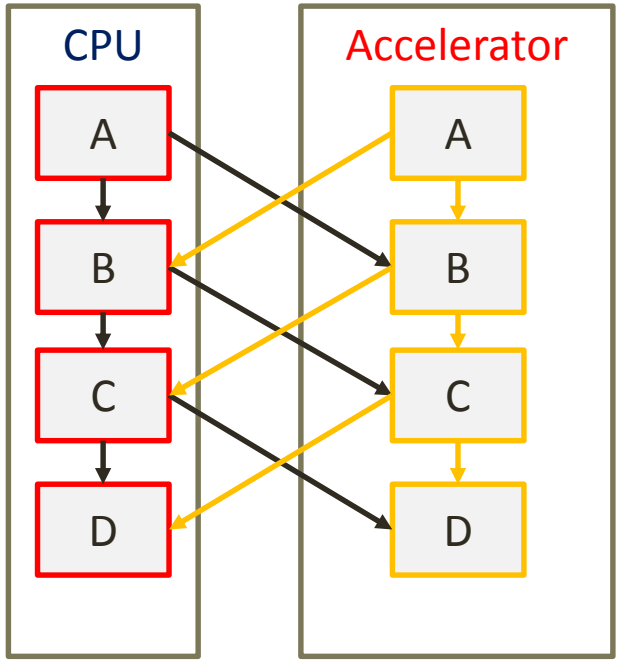
- Client-Server approach is still valid if
  - Accelerator is remote
  - Framework can not include or work with
    - Libraries
    - Languages
    - Compilers
    - Programming paradigms needed by accelerator

Future Accelerator use modes

- Offloading should be integrated in scheduler.
  - Could be a scheduler extension or plugin to modify graphs.
- Two different approaches depending on accelerator
  - Accelerator is a symmetric resource
  - Accelerator is an asymmetric resource

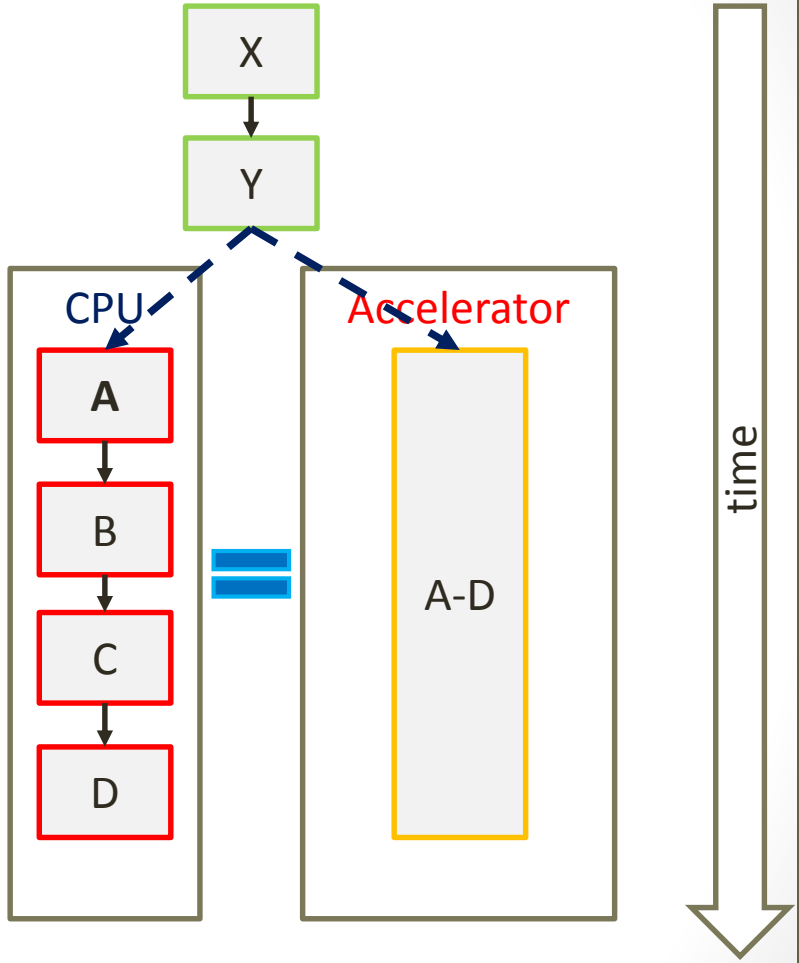
Symmetric Resource

- Accelerator algorithms generate same output containers so next algorithm can execute in either CPU or accelerator.
  - Easy to handle in current design
  - May need differentiation of algorithms if different implementations are required (eg. CPU-GPU vs CPU-KNL)



Asymmetric resource

- Accelerator algorithms execute several steps internally and create final containers
  - There is no support for this mode currently
  - Probably most efficient use of accelerator
  - Scheduler has to have a separate graph segment for accelerator paths
  - Event processing graph changes for depending on the availability of the resource
  - May need data conversions steps

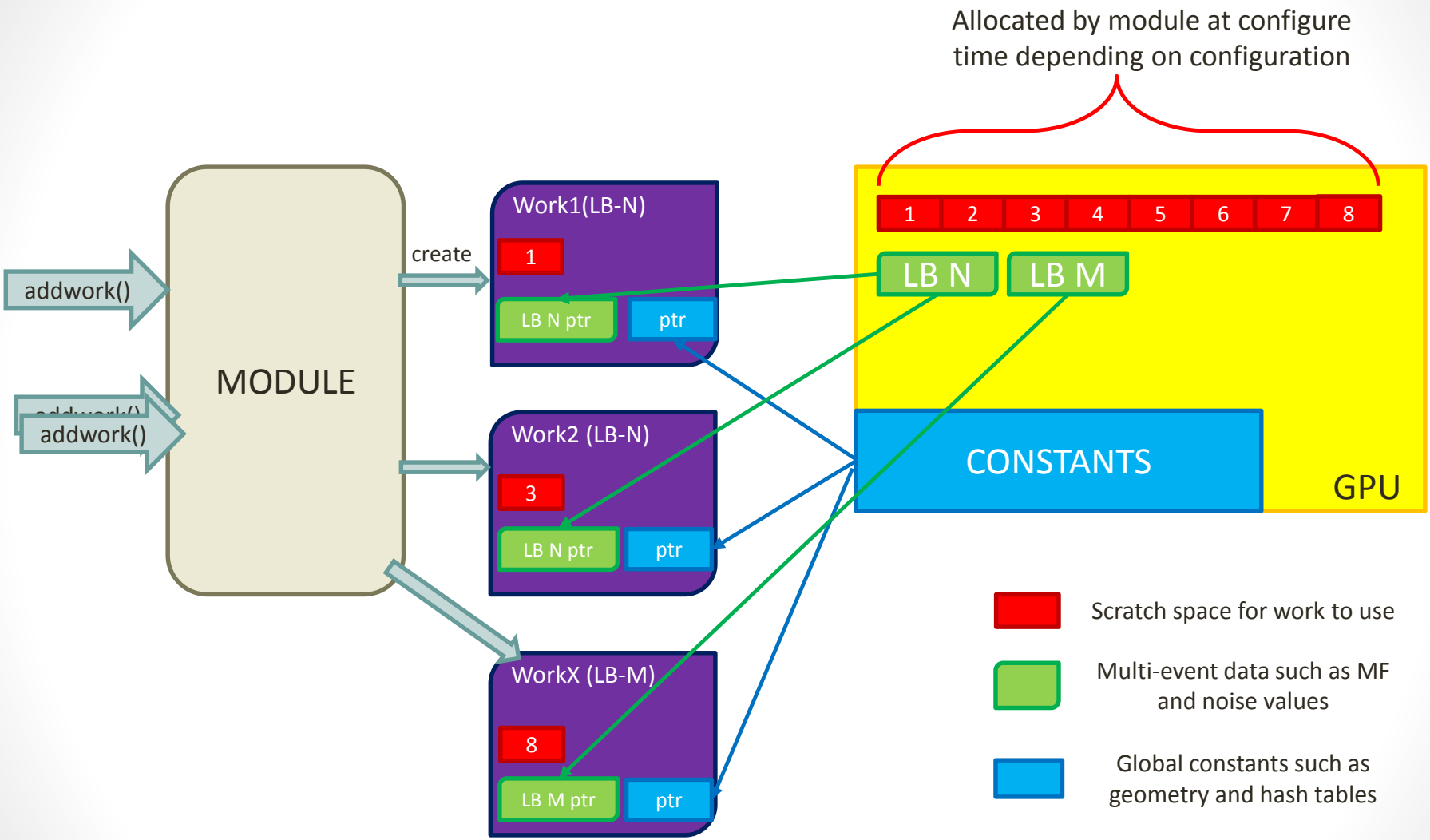


Summary

- The choice of offloading model highly depends on availability of algorithms and accelerators
- Probably it will happen in an evolutionary way
  - Plugin system can help
- Athena-APE will still work in new framework

BACKUP

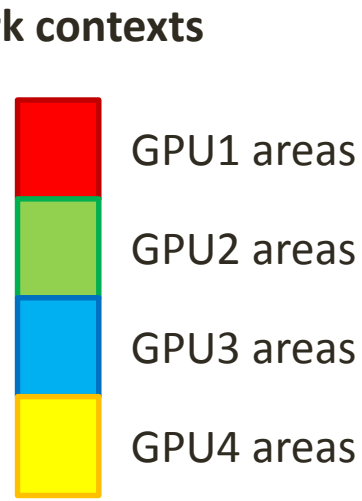
Multi-Work resource management



Works are processed and deleted returning resources to Module

MutliGPU utilization

- Module initializes multiple workspace on each GPU
- Queues them in cyclic manner
- Assigns front of the queue to new Works
- Workspace returns back to queue when job is deleted



WORK