

Core SW developments for the ES

Vakho Tsulaia

ATLAS Opportunistic Resources Mini-TIM

CERN, October 1, 2018

REMINDER

Today: AthenaMP and the ES

- AthenaMP in the Event Service uses a special configuration, which consists of **one Event Range Scatterer** process and **one or many Event Range Processor** processes
 - `ATHENA_PROC_NUMBER=N` means 1 Range Scatterer and N Range Processors
- **Event Range Scatterer**
 - Communication channel with the Pilot: receives event ranges; sends range completion reports and error messages
 - Communication channel with Range Processors: sends event ranges; receives requests for new ranges and range completion reports
- **Event Range Processor**
 - Processes event ranges one at a time
 - Upon successful completion of an event range writes the output to the disk, reports the output location to the Scatterer and requests next range

REMINDER

Specifics of the ES configuration for AthenaMP

- A failed worker process (Range Processor) does not bring down the entire instance of AthenaMP
 - When one of the workers dies, AthenaMP replaces it with a new one
- AthenaMP **sub-processes exchange messages** throughout the job
 - Unlike "regular" AthenaMP, which first fills a **Shared Event Queue** with event identifiers (event positions within the input file) and then lets its workers pull event IDs from the queue
- AthenaMP **exchanges messages with the Pilot** throughout the job
 - No communication between the Pilot and "regular" AthenaMP
- Event processors leverage the **Output File Sequencer** mechanism for writing event range outputs into separate files on the disk
 - Event Service today is perhaps the only user of the Output File Sequencer
 - Handling of in-file metadata requires special attention.

AthenaMP improvements

- The mechanism for handling failed AthenaMP workers needs to be rewritten
 - A relatively minor development
- If we want to use Shared Writer in the ES
 - Shared Writer will need to be able to work with the Output File Sequencer
 - This will result in combining outputs of multiple event ranges into a single file.
 - Event Range Scatterer and Pilot/JEDI need to be able to handle this
- Shared Reader can help us avoid reading the same input file from multiple processes on the compute node
 - The integration of Shared Reader into the ES is hopefully a relatively minor development too

MPI Shared Writer

- An MPI data collector can help us write output files only from one or few compute nodes
 - The majority of compute nodes within an MPI job will be sending their outputs to the collector over MPI
- The idea is to eventually use a new TMPIFile mechanism of ROOT. Currently under development
 - Taylor is leading this effort
 - Initial prototype implemented by a CCE student, who already left the project
 - Yunsong (NERSC) recently joined this activity
- TMPIFile is just one part of the whole picture. We will need to design and implement the data collector and integrate it with Yoda
 - Will also require changes in Yoda/Droid

ES with AthenaMT

- ES integration with AthenaMT requires a number of non-trivial code developments
 - Need to implement an MT equivalent of the Event Range Scatterer and integrate it with the Gaudi Scheduler
- Error handling/reporting will be a tricky part
 - If there is a segfault/crash in one of the threads, then the entire AthenaMT instance goes down and pilot/droid will have no idea which event triggered the crash
 - **Is this going to be a blocker?**
- One possible way to work around this problem is to not run more than one range in flight
 - Such that if there is a crash, we at least can be certain which range crashed
 - OTOH, by doing that we'll introduce visible CPU inefficiency: at the end of each range CPU cores will be idle waiting for the slowest event to finish
- Running multiple event ranges in flight seems to be more CPU-efficient, but then cutting outputs by event range boundaries becomes tricky

Summary

- Support of AthenaMP for the Event Service looks rather straightforward
- MPI data source and sink mechanisms look promising for running on HPC
 - The benefits will not come for free, though. Requires changes in Yoda/Droid
- The integration of AthenaMT with the Event Service will be tricky
 - ... and the benefits of running AES with AthenaMT are yet to be demonstrated