

HEC FSIO 2010 Workshop

Breakout 1.2: Combining Analytics and File Systems, Leveraging Google, etc.

Moderators: Marti Bancroft, John Bent, and Rob Ross

Highlights of discussion

- How much can we analyze, and how will we perform that analysis?
- How do we choose what data to keep for analysis?
- Changes to I/O architectures
- Data models for computational science
- Dynamic and composable systems
- Scheduling and co-scheduling
- (Things I was surprised *not* to hear)



How much can we analyze, and how will we perform that analysis?

- Trend toward increasing role of analysis in HPC I/O workloads
- At the same time, discussion that perhaps 95% of data that is produced is never analyzed. Can we do better than this, given upcoming “tsunami” of data?
- Is our current model of analysis (dump and post-process) adequate to our analysis needs?
 - If we want a more “active storage” etc. approach, do we have enough knowledge at the low levels to accomplish that?
 - In situ analysis (i.e., analysis of data while in memory during simulation) provides some help, but not all desired analysis is known beforehand or can be performed in situ (e.g., analysis that requires multiple timesteps).
 - What about running analysis on the HPC system?



How do we choose what to keep for analysis?

- We want to keep everything we can.
- How do we prioritize what to throw away?
 - Can we quantify the cost of recomputing data vs. storing and retrieving data?
 - How do you know the value of data that you haven't analyzed?
 - Is the value related to how many people will extract knowledge from it?
- What about real-time analysis or simulations informed by real-time input data?
 - Some emergency-management related real-time stuff exists.
 - UQ and feedback of analysis into what to simulate next falls in this category.
- Will some application teams will hit the point where they simply cannot save enough to do their science (e.g., molecular dynamics, cross correlations on trajectories)?



Changes to I/O architectures

- Can we justify including analysis in the FS, or should we just put things in middleware?
- Combining data such as indices with traditional datasets confuses prefetching.
- Primitives in current PFSes aren't right (PLFS results as one indicator)
 - Rather than starting with a clean slate, start with the object model from current PFSes and use middleware for the rest?
 - Provide control over buffering, prefetching, etc. so that the middleware can control it.
- How do we, or do we, separate archive from online?
- Smart compression might be a method to save space and bandwidth. Where is it best applied?



Data models for computational science

- There is a need for more complex data models than those supported in tools like HDF5, netCDF, and ADIOS today.
- Storage system needs to expose interface for layout control and inspection.
- Data model needs to map data into storage containers (e.g., files, objects) so that locality may be exploited during analysis.
- Data model support can exist in a library.
- As an alternative to explicitly storing data in a particular format, we could capture data as it resides in memory so that it may be restored quickly.
- We will want operators on the data model.



Dynamic and composable systems

- How is I/O support different in dynamic/composable systems?
- Will we always forklift upgrades, or will we be adding/removing components over time?
- How does the purposeful addition/removal of components compare to how we manage fault tolerance (unexpected removal of components)?



Scheduling and Co-Scheduling

- Awareness of data location during scheduling is seen as an advantage of the MapReduce approach.
- There is concern about contention between simulation output (checkpoint) and analysis, whether they are occurring on the same machine, or in an active storage environment.
 - HEC FSIO QoS work should provide some (partial?) solutions.
- There is need for integration between scheduling of HPC jobs and analysis jobs.



Things I was surprised not to hear (and will bring up now)

- Indexing as a tool for reducing I/O demands of analysis.
- Possible roles of heterogeneous storage in analysis I/O (e.g., shifting data onto SSD from HDD prior to analysis, as per Reddy's second scenario).
- Leaving data on compute node SSD and scheduling analysis back on the same nodes.

