Exploring Trade-offs in Transactional Parallel Data Movement
Ivo Jimenez, Carlos Maltzahn (UCSC); Jai Dayal (Georgia Institute of Technology); Jay Lofstead (Sandia National Labs);

The Road to Exascale

Exascale systems that are slated for the end of this decade will include up to a million compute nodes running about a billion execution threads. In this scenario, traditional methods that ameliorate I/O bottlenecks do not work anymore. I/O Staging \cite{1, 2} proposes designing a portion of the nodes to manage I/O.

I/O stack requirements

In order to solve the multi-client scenario, recent work \cite{3, 4} proposes abstracting the storage with basic concurrency control capabilities and thus allow clients to manage isolation semantics. One way this can be achieved is by having storage servers that implement:

1. Multi-versioning concurrency control.
2. Object visibility control.

Consensus Protocols

Our goal is to explore the trade-offs across the transaction coordination spectrum, identifying precisely where overheads are at and thus provide a toolkit for scientists to allow them to pick the most appropriate alternative for their workloads.

Preliminary Evaluation

Related Work

- The DOE’s Fast Forward Storage and I/O project is implementing transactional features into a next-generation stack. The FastForward protocol used to implement transactions is similar to the NBTA protocol referenced here.
- Many proposals for fault-tolerance \cite{5} in HPC make use of consensus protocols to identify faulty processes. Our work is complementary to these efforts.