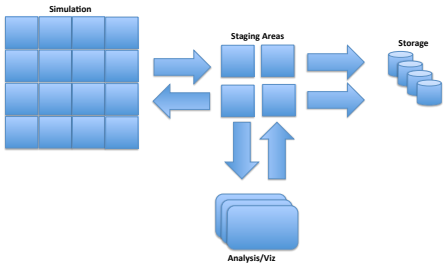
D^{2}T: Doubly Distributed Transactions for High Performance and Distributed Computing

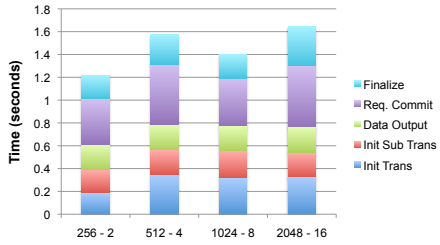
Jai Dayal, Jay Lofstead, Karsten Schwan, Ron Oldfield
jdayal3@gatech.edu, gflofst@sandia.gov, schwan@cc.gatech.edu raoldfi@sandia.gov

Motivation

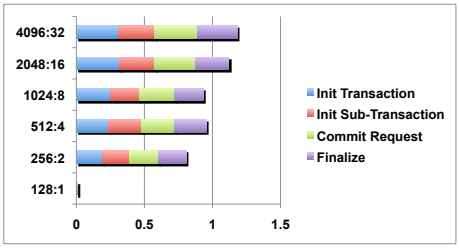
- IO limitations projected for exascale encouraging moving to online scientific workflows.
- Code coupling and system reconfiguration for load balancing and resilience need action validation.
- Transactions offer a model for consistency including ACID properties
- MxN environment demands new look at how to offer ACID properties.



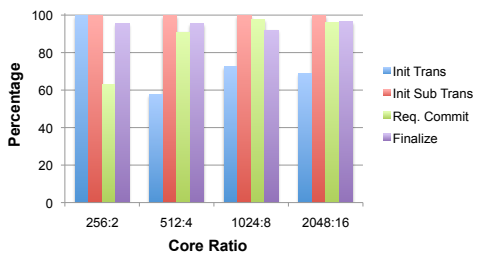
Example Architecture



Data Movement Overhead (1 MiB)



System Reconfiguration Overhead



Multi-Protocol Overhead

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Project Goals

- Bring ACID style guarantees to online data movement
- Offer mechanism to control visibility of system reconfiguration operations until they are complete and correct
- Offer mechanism to support dynamic load balancing without prematurely exposing or hiding resources

Solution

- Distributed MxN transactions
  - Inspired by current distributed transaction (1xN) semantics
  - Handle single operation with many coordinated clients (M) and many coordinated servers (N)
- Must be scalable
  - Large number of clients and servers leads to high message volumes and aggregate size (MxN)
  - Too much overhead will reduce the gains associated with using data staging

Initial Implementation

- Dual Coordinators
  - Reduces problem to 1 to 1 coordination and thus reduces the volume of messages by avoiding all-to-all communication
  - Improves scalability
  - But, localized bottlenecks that may not scale
  - Message count and aggregate size likely too big for a single node
- 3 stages in a given transaction
  - **Init Phase:** client side initializes transactions and sub-transactions with servers
  - **Operation Phase:** Clients perform op(s)
  - **Commit Phase:** Clients and servers validate success of operations
- Transactions and Sub-Transactions
  - **Transaction:** Groups multiple operations into an atomic action
  - **Sub-transaction:** represents one operation (or variable) in the overall transaction

Benefits

- Atomicity, Consistency, Isolation
  - Hides operations from other users until they are completed and correct
  - Provides guarantee that all operations have completed (atomic = all or none)
  - Correctness can be ensured by adding hashes (SHA-1, MD5, etc) to data
  - Applications are shielded from incomplete or erroneous data sets
- Durability (future work)
  - Identified approaches
    - Store data on local SSD
    - Replicate data to other node RAM or SSD
    - Save to centralized storage
  - Challenges
    - Find it later
    - Time/Space costs for storage

Challenges

- MxN at extreme scale is hard
  - 10 million clients to 1000 servers
  - message counts and aggregate sizes exceed per node limits
- Data staging systems hold data in volatile memory
  - Any crashes can lead to permanent loss of data and incomplete data sets
  - Processing should not commence until data is complete
  - Data should not be removed from a work queue until fully processed and stored
- System reconfiguration/load balancing need to manage access to resources
  - Fully start new replicas before advertised for use
  - Safely remove resources from access making the change permanent only when proper shutdown and current processing is complete
  - Manage redeployment as an atomic action that safely shuts down old, starts new, and commits new configuration to services directory

Logical Protocol