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

Jai Dayal[1], Jay Lofstead[2], Karsten Schwan[1], Ron Oldfield[2]
[1]CERCS, Georgia Institute of Technology, [2]CSRI, Sandia National Labs.

ABSTRACT

Current exascale computing projections suggest rather than a monolithic simulation running for the majority of the machine, a collection of components comprising the scientific discovery process will be employed. This move to an online workflow scenario requires knowledge that inter-step operations are completed and correct before the next phase begins. Further, dynamic load balancing or fault tolerance techniques may dynamically deploy or redeploy resources for optimal use of computing resources. These resources should only be used if they are successfully configured.

Our D^{2}T system offers a mechanism to support these kinds of operations by providing database-like transactions with distributed servers and clients. Ultimately, with adequate hardware support, full ACID compliance is possible for the transactions. To prove the viability of this approach, we show that the D^{2}T protocol has less than 1.2 seconds of overhead using 4096 clients and 32 servers with good scaling characteristics using this initial prototype implementation.

1. INTRODUCTION

As scientific applications scale towards exascale, they will incorporate more complex models that have previously only been run as separate applications. For example, in fusion science, simulation of the edge of the plasma [2] and the interior of the plasma [6] are currently separate simulations. To have a more complete, accurate model for a fusion reactor, these components will need to be tightly coupled to share the effects between the two models. The CESM climate model [5] is similar in that it incorporates atmosphere, ocean, land surface, sea ice, and land ice through a coupling engine to manage the interactions between each of these different systems yielding a more accurate model of global climate. In most cases, these and other scientific applications are part of larger offline workflows that process the output written to storage in phases that ultimately yields insights into the phenomena being studied. Current work to enable these coupling and workflow scenarios are all focused on the data issues to resolve resolution and mesh mismatches, time scale mismatches, and simply making data available through data staging techniques. In most of these cases, each of the components are run using a separate execution space for fault isolation and to aid in scalability.

The D^{2}T protocol aims to offer full ACID-style guarantees for the encapsulated operations such as data movement or system reconfiguration for fault tolerance and load balancing. While this is certainly possible with adequate hardware particularly to support the durability guarantee, this initial work shows that such a system can be built that supports most of the ACID-style guarantees with current hardware, what the performance impact will be, application coding implications, and begin to address the scalability challenges so that it is applicable for exascale-sized platforms. More specifically, D^{2}T can address both the code coupling/online workflow/data staging as well as the fault tolerance/system reconfiguration scenarios.

While it is true that for many high transaction volume environments, ACID-style transactions can lead to scalability problems catalyzing the explosive growth of NoSQL style data stores. The D^{2}T techniques are intended for a different environment than BASE properties can support. In our scenario, we are focused on supporting online workflows and other high performance and distributed computing operations. For these scenarios, eventual consistency is insufficient. Throughput in the online workflow is only possible with guarantees about data completeness and correctness. The BASE properties are insufficient for maintaining this throughput.

2. RELATED WORK

The concept of distributed transactions has been around for decades. We are extending this technology to address distributed clients working in concert. While this is not critical for the core database area, is crucial for HPC applications given the massively parallel nature of the modern HPC environment.

ZooKeeper [1] and other Paxos [3] implementations have a superficial similarity to our D^{2}T protocol in that they provide the consistency and synchronization mechanisms for messaging to a collection of servers from a distributed set of sources. Under the hood, Paxos is solely 1xN with an eventual consistency model. The inherent assumption that an update or insert originates from a single source limits the applicability of the protocol for this environment.

GridFTP and Sinfonia offer related functionality, but either do not offer the same levels of guarantees or is limited to a 1xN semantics inappropriate for the HPC environment.

3. THE D^{2}T PROTOCOL

The D^{2}T protocol provides the necessary extensions to traditional distributed transactions to afford extensions to distributed clients as well as distributed servers, hence the doubly distributed transactions. (Figure included in final poster.)

D^{2}T provides a two-level system where a master transaction represents the entire collection of operations comprising the atomic whole and the sub-transactions represent the individual component operations that should be handled as a unit. It is expected there will be a series of sub-transactions that must be handled as a single atomic operation. This affords the opportunity to isolate a single action and offers an opportunity to retry failures before giving up entirely.

4. PRELIMINARY RESULTS

These tests are performed on the RedSky Sun blade system at Sandia. Our protocol is implemented with two popular technologies, Sandia's NSSI RPC substrate, and Open MPI. We use NSSI to communicate across MPI application boundaries, such as moving data or operation messages between clients and servers, and we use MPI to handle communication within MPI boundaries.

To test the kind of overhead involved in online workflows, a payload size of 1 MiB is used. While larger payloads are typical for HPC applications, this gives a baseline that should be straightforward to extrapolate to larger payload sizes. An idea of how to scale the payload sizes for the NSSI protocol [4] on various platforms has been performed. The results are shown in Figure 1.

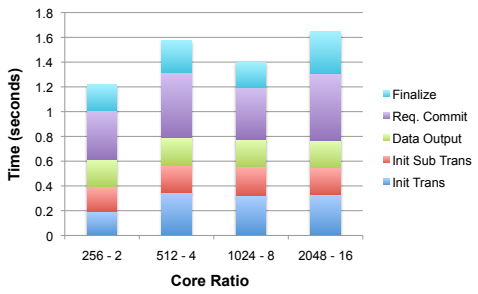

Figure 1: Online Workflow Overhead

For this experiment a ratio of 128:1 clients to servers ratio is maintained as it scales.

By isolating the NSSI from the MPI messaging and viewing from the server side, the root cause of the overhead becomes apparent. Overall, the MPI overhead shown for each phase of the messaging averages 75%+ of the time spent for each phase. (A figure detailing this will be included in the final poster)

Similar results were also obtained for system reconfiguration operations. With no significant data being moved for these operations, we demonstrate the overheads associated with these kinds of non-data movement transactions. (Figure included in final poster)

5. CONCLUSIONS AND FUTURE WORK

This is very early results showing the potential of incorporating MxN doubly distributed transactions as a way to enable online scientific application workflows or dynamic system reconfiguration operations. We are extending this current implementation to be able to read completed transactions out of the staging area requiring at least a simple metadata system. Existing metadata solutions will both inform and hopefully supply a usable system for this purpose.

Durability approaches under investigation focus on node-local and compute area solutions. Avoiding the centralized storage system is a primary goal to ensure performance and scalability of this system.

There are several opportunities to optimize D^{2}T by piggybacking certain messages and providing an optional optimistic and potentially implied success model, thus reducing the overall volume of messages. Some examples of similar optimizations were found in Sinfonia, where they introduce the concept of mini-transactions. One such example found in this work is piggy-backing the transmission of the data along with the commit/abort request.

6. ACKNOWLEDGEMENTS

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.

7. REFERENCES

[1] P. Hunt, M. Konar, F. P. Junqueira, and B. Reed. Zookeeper: wait-free coordination for internet-scale systems. In *Proceedings of the 2010 USENIX conference on USENIX annual technical conference*, USENIXATC'10, pages 11–11, Berkeley, CA, USA, 2010. USENIX Association. URL: `http://dl.acm.org/citation.cfm?id=1855840.1855851`.

[2] S. Ku, C. S. Chang, M. Adams, E. D. Azevedo, Y. Chen, P. Diamond, L. Greengard, T. S. Hahm, Z. Lin, S. Parker, H. Weitzner, P. Worley, and D. Zorin. Core and edge full-f ITG turbulence with self-consistent neoclassical and mean flow dynamics using a real geometry particle code XGC1. In *Proceedings of the 22th International Conference on Plasma Physics and Controlled Nuclear Fusion Research*, number IAEA-CN-165/TH/P8-40, Geneva, Switzerland, 2008.

[3] L. Lamport. The part-time parliament. *ACM Trans. Comput. Syst.*, 16:133–169, May 1998. URL: `http://doi.acm.org/10.1145/279227.279229`, `doi:http://doi.acm.org/10.1145/279227.279229`.

[4] J. Lofstead, R. Oldfield, T. Kordenbrock, and C. Reiss. Extending scalability of collective io through nessie and staging. In *In Proceedings of Petascale Data Storage Workshop 2011 at Supercomputing 2011*, 2011.

[5] NCAR and UCAR. Community earth system model. http://www.cesm.ucar.edu/models/cesm1.0, 2012.

[6] W. X. Wang, Z. Lin, W. M. Tang, W. W. Lee, S. Ethier, J. L. V. Lewandowski, G. Rewoldt, T. S. Hahm, and J. Manickam. Gyro-Kinetic simulation of global turbulent transport properties in tokamak experiments. *Physics of Plasmas*, 13(9):092505, 2006. URL: `http://link.aip.org/link/?PHP/13/092505/1`, `doi:10.1063/1.2338775`.