MHT: A Light-weight Scalable Zero-hop MPI Enabled Distributed Key-Value Store

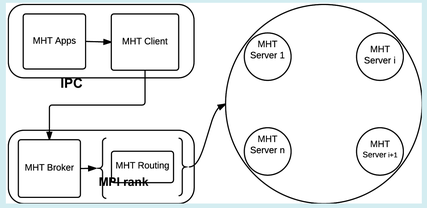
Xiaobing Zhou[1], Tonglin Li[2], Ke Wang[4], Dongfang Zhao[2], Iman Sadooghi[2], Ioan Raicu[2,3]
[1]Hortonworks, [2]Illinois Institute of Technology, [3]Argonne National Laboratory, [4]Intel

Abstract
NoSQL databases, such as key-value stores, are known for their ease of use and excellent scalability. However supercomputers and HPC applications are not able to enjoy the benefits of distributed key-value stores due to their customized OS and communication stack. In this paper, we propose and implement a key-value store that supports MPI while allowing application access at any time without having to declaring in the same MPI communication world. This feature may significantly simplify the application design and allow programmers leverage the power of key-value store in an intuitive way. In our preliminary experiment results captured from a supercomputer at Los Alamos National Laboratory, our prototype shows linear scalability at up to 256 nodes.

Motivation
- Distributed Key-value store for HPC
- Simplifying large system service design
- Simplifying HPC application design

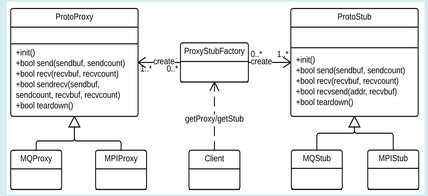
Contributions
- MHT: a MPI enabled key-value store
- Support dynamic MPI process join
- Real system evaluation up to 256-nodes

Architecture
- Broker based 3-tier architecture
  - Clients: MPI or regular processes
  - MHT brokers w/ MPI rank
  - MHT servers w/ MPI rank
- MHT brokers and servers in same MPI_COMM_WORLD
- Clients in different one
- Message Queue IPC



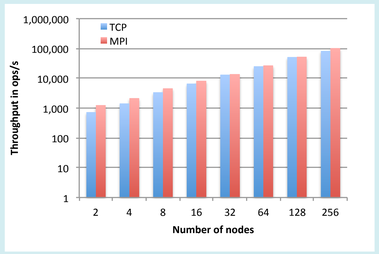
MHT architecture

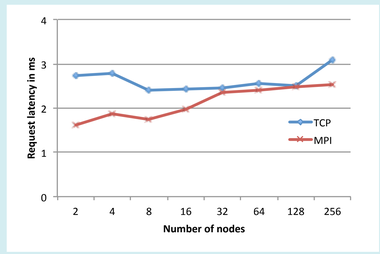
Implementation
- Abstracting protocols
- Client proto proxy and server proto stub
- Support MPI/TCP/UDP



Structure of protocol abstraction

Evaluation
- Test bed: PROBE at LANL, 1024 nodes, two 2.6GHz 64-bit AMD Opteron, 8GB RAM per node





Challenges
- Predefined MPI communication world
- Unified API for various protocols
- Dynamic MPI process join not available in many supercomputers.
- Applications and MPI layer share the same failure domain

Proposed Solution
- Use inter process communication (IPC) between clients and brokers
- Abstracting network communication protocol
- Client-side: proto proxy
- Server-side: proto stub
- Sync/async message send and receive
- Separating failure domain of application and MPI layer

Future work
- Enhanced fault tolerance features
- Exa-scale system services with MHT
- Utilizing MHT to boost HPC application performance and scalability

Acknowledgement
This work was supported in part by the National Science Foundation grant NSF-1054974. This work used Kodiak supercomputer, a Parallel Reconfigurable Observational Environment (PROBE) deployed at Los Alamos National Laboratory (LANL).

Reference
- Tonglin Li, Xiaobing Zhou, Xiaobing Zhou, Ke Wang, Dongfang Zhao, Iman Sadooghi, Zhao Zhang, Ioan Raicu, etc., A Convergence of Distributed Key-Value Storage in Cloud Computing and Supercomputing, Journal of Concurrency and Computation Practice and Experience (CCPE), 2015.
- Tonglin Li, Xiaobing Zhou, Kevin Brandstatter, Dongfang Zhao, Ke Wang, Anupam Rajendran, Zhao Zhang, and Ioan Raicu. ZHT: A light-weight reliable persistent dynamic scalable zero-hop distributed hash table. IPDPS ' 13.
- Tonglin Li, Raman Verma, Xi Duan, Hui Jin, and Ioan Raicu. Exploring distributed hash tables in highend computing. *SIGMETRICS Performance Evaluation Review*, 2011.
- J.M. Wozniak, B. Jacobs, R. Latham, S.W. Son S. Lang, and R. Ross. C-mpi: A DHT implementation for grid and HPC environments. 2010.