SharP Hash: A High-Performing Distributed Hash for Extreme-Scale Systems

Achievement: Designed, implemented, and evaluated a novel, RDMA-based distributed hash table for extreme-scale systems with support for hierarchical and heterogeneous memory and simple, straight-forward interfaces for data-structure creation and manipulation.

Significance and Impact: This work provides a high-performing, RDMA-based distributed hash table for both Big-Compute and Big-Data applications on extreme-scale systems with support for hierarchical and heterogeneous memories.

Research Details:
- Designed the interfaces and abstractions required to implement a high-performing distributed hash table through the adaptation of hopscotch hash table to RDMA-based networks.
- Implemented SharP Hash as a supported data-structure of SharP allowing the user great flexibility with respect to the memories used by both Big-Compute and Big-Data applications.
- Evaluated SharP Hash on ORNL’s Titan and Rhea systems with two applications: (1) a micro-benchmark capable of measuring the latency and throughput of SharP Hash through direct usage, and (2) the porting of the Memcached and memaslap ecosystem to SharP Hash with a comparison to the vanilla version.

Sponsor/Facility: Work was performed with support from ORNL

PI and affiliation: Manjunath Gorentla Venkata from CSMD – Oak Ridge National Laboratory

Team: Manjunath Gorentla Venkata and Ferrol Aderholdt – CSMD, Oak Ridge National Laboratory, and Zachary W. Parchman – Tennessee Tech University.


Overview:
A high-performing distributed hash is critical for achieving performance in many applications and system software using extreme-scale systems. It is also a central part of many Big-Data frameworks including Memcached, file systems, and job schedulers. However, there is a lack of high-performing distributed hash implementations. In this work, we propose, design, and implement, SharP Hash, a high-performing, RDMA-based distributed hash for extreme-scale
systems. SharP Hash’s high performance is obtained through the use of high-performing networks and one-sided semantics. We perform an evaluation of SharP Hash and demonstrate its performance characteristics with a synthetic micro-benchmark and implementation of a Key-Value (KV) store, Memcached.