SharP: Towards Programming Extreme-Scale Systems with Hierarchical Heterogeneous Memory

Achievement: Designed, implemented, and evaluated a novel, data-centric programming abstraction for scientific applications, which (1) abstracts the hierarchical and heterogeneous memories of extreme-scale systems, (2) provides users with simple, straight-forward interfaces for data structure creation and management across various memories with *usage hints* and *constraints* allowing for data-locality, data-affinity, data-resilience, and data-sharing, and (3) allows for interoperability with modern programming models such as MPI, OpenSHMEM, and task-based models reducing the difficulty of any porting effort to SharP.

Significance and Impact: This work provides a novel, data-centric programming abstraction for Users of extreme-scale systems working on Big-Compute and Big-Data applications, which allows users to easily leverage the various architectural components of the system while providing data-locality, data-affinity, data-resilience, and data-sharing as well as reducing data movement.

Research Details:
- Designed the abstractions for SharP to abstract the various memories and components of a system as well as the underlying communication library to support features such as globally shared data-structures across various memories, data-locality and data-affinity to various components in the system, a reduction in data-movement between components, and interoperability with various popular programming models.
- Implemented the SharP abstraction as a linkable library that leverages various libraries such as hwloc, memkind, PMEM, and UCX to ensure continued high-performance deployment of SharP-based applications regardless of future architectural changes.
- Evaluated the SharP library on ORNL’s Titan and Rhea systems by porting applications and ecosystems from both Big-Compute (i.e., QMCPack) and Big-Data (i.e., Memcached) and compared these ports to their vanilla versions.

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:
The pre-exascale systems are expected to have a significant amount of hierarchical and heterogeneous on-node memory, and this trend of system architecture in extreme-scale systems is expected to continue into the exascale era. Along with hierarchical-heterogeneous memory, the system typically has a high-performing network and a compute accelerator. This system architecture is not only effective for running traditional High Performance Computing (HPC) applications (Big-Compute), but also running data-intensive HPC applications and Big-Data applications. As a consequence, there is a growing desire to have a single system serve the needs of both Big-Compute and Big-Data applications.

Though the system architecture supports the convergence of the Big-Compute and Big-Data, the programming models have yet to evolve to support either hierarchical-heterogeneous memory systems or the convergence. In this work, we propose and develop the programming abstraction called SHARed data-structure centric Programming abstraction (SharP) to address both of these goals, i.e., provide (1) a simple, usable, and portable abstraction for hierarchical-heterogeneous memory and (2) a unified programming abstraction for Big-Compute and Big-Data applications.

To evaluate SharP, we implement a Stencil benchmark using SharP, port QMCPack, a petascale-capable application, and adapt the Memcached ecosystem, a popular Big-Data framework, to use SharP, and quantify the performance and productivity advantages. Additionally, we demonstrate the simplicity of using SharP on different memories including DRAM, High-bandwidth Memory (HBM), and non-volatile random access memory (NVRAM).