Pagoda: UPC++/GASNet

A computation being performed on one part of a large system often needs to access or provide data to another part of the system in order to complete a scientific simulation. The Partitioned Global Address Space (PGAS) model provides the appearance of shared memory accessible to all the compute nodes while implementing this shared memory behind the scenes using physical memory local to the nodes and primitives, such as remote direct memory access. The Pagoda project is developing a performant PGAS programming system to be deployed on exascale systems.

The Pagoda project is developing a programming system to support exascale application development using the PGAS model, with a focus on supporting irregular applications and data structures. There are two components to Pagoda: (1) a portable, high-performance, global-address-space communication library and (2) a template library that provides convenient methods for access and using the global address space. Together, these components enable the agile, lightweight communications that occur in applications, libraries, and frameworks running on exascale systems.

Pagoda enables effective scaling by minimizing the work funneled to heavyweight cores, avoiding the overhead of long, branchy serial code paths and supporting efficient fine-grained communication for both single- and multi-threaded environments. The importance of these properties is exacerbated by application trends; many applications in the ECP require the use of adaptive meshes, sparse matrices, dynamic load balancing, or similar techniques. Pagoda's low-overhead communication mechanisms can maximize the injection rate and network utilization, tolerate latency through overlap, streamline unpredictable communication events, minimize synchronization, and efficiently support small- to medium-sized messages arising in such applications. Pagoda complements other programming models, enabling developers to focus their efforts on optimizing performance-critical communications.

The Pagoda team is focusing on developing new features that will support application and library requirements unique to the ECP and performance improvements that will enable the ECP software stack to exploit the best-available communication mechanisms, including novel features being developed by vendors, such as remote direct memory access mechanisms offered by network hardware and on-chip communication between distinct address spaces.

Progress to date

- The Pagoda team provides regular releases of their communication and template library that typically include new features and performance improvements.
- The team delivered new features in the template library, including support for subset teams and collectives and features for expressing data movement between processors and accelerators.
- The communication library developed by the team supports remote direct memory access, remote procedure calls, future-based contracts, and remote atomics with offload to network hardware.

PI: Erich Strohmaier, Lawrence Berkeley National Laboratory

Collaborators: Lawrence Berkeley National Laboratory