**Motivation**

- Traditional bulk-synchronous parallel routines do not excel at irregular computation patterns present in big-data and adaptive applications.
- Asynchronous many-tasking (AMT) models, characterized by lightweight tasks concurrently operating on global data, are dynamic and adaptive in nature. When work follows data, global load-balancing naturally takes the form of dynamically balancing global data.
- PGAS is less well-suited for event-driven and active-message execution where ephemeral computation is performed on global data. Static nature of PGAS restricts the system's ability to load balance computation and communication.
- Examples of applications demonstrating runtime imbalance: dynamic graph algorithms, adaptive mesh-refinement, etc.
- Scalability and performance depends on initial data distribution and dynamic runtime-supported load and data balancing.

- We propose global virtual memory for AMT runtimes using an active global address space (AGAS).
  - The active global address space is “active” in two senses. It is virtualized and allows memory to dynamically relocate; its usage is primarily through the use of active messages.

**Problem: Dynamic Load Imbalance**

- Adaptive Mesh Refinement (AMR) affected by dynamic runtime load imbalance.
- Sub-optimal initial partitioning of data can create communication and computation hotspots that cause performance degradation—ever after employing the common latency-hiding optimizations.

**High-Performance ParalleX (HPX-5)**

- HPX-5 is a library-based implementation of the ParalleX model in C.
- The HPX-5 runtime system features:
  - Cooperatively scheduled lightweight threads
  - Inter-thread synchronization (LCOs)
  - Active messages (via Pallets)
  - Global Address Space (AGAS)
- Programming interfaces for global address manipulation, translation, and allocation; parcels, lightweight threads, and LCOs.
- Parcel transport using flexible one-sided or two-sided networking interfaces.
- Parcels target global addresses, carry payload data, identify message handlers for execution, and specify continuation addresses.

**Network-Managed Global Virtual Memory**

- Network-assisted AGAS leverages the capabilities of the network fabric to manage addressing rather than software at the endpoint hosts.
- Uses a GASNet conduit with IB multicast over unreliable datagram (UD) which allows receivers to accept packets from any node in the network.
- Address-to-port mapping in the switch determines the current owner of the global address.
- Storing the page table in the network switch makes lookup operation challenging.

**Benefits of Network-assisted AGAS**

- Messaging overheads can be reduced through the use of a software translation cache.
- However, software caching incurs storage and synchronization overheads.
- Page table mappings have to be evicted as pages are remapped.
- Sequential direct-mapped cache extended with two cache replacement policies: random and LRU (least recently used).
- Bounding the cache size adds extra cache eviction logic and incurs capacity cache misses.

**Software AGAS uses a chunk translation table (CTT) to translate GVA to GPA. In the hardware implementation, this table is maintained in the switch.**

**Effect of bounded storage and cache replacement policies on the GUPS random access microbenchmark at 192 cores.**

The figure on the right shows LRU-cache statistics at rank 0 for 2^26 random accesses of a 2^21 words global table.

**Performance degradation of global updates due to thread contention.** The concurrent cache is 25% slower than the hardware directory (cache bypass) solution.

**Impact of dynamically moving pages (i.e., “remapping” blocks)**

- The GUPS microbenchmark with a global table consisting of 4 pages distributed across 192 cores.
- As page movement frequency increases, the software approach takes increasingly longer. In contrast, the distributed approaches offered by the hardware can maintain a constant, or improved, runtime.

**Conclusions**

- The HPX-5 runtime system provides the dynamic, adaptive features necessary for the efficient execution of large-scale irregular applications.
- To manage both locality and load concerns for the programmer, HPX-5 provides global virtual memory using an active global address space (AGAS) where data can be dynamically migrated around in the system.
- Network-assisted AGAS incurs reduced common-case overheads; however, hardware remapping remains expensive due to the cost of flow programming on present-day switches.

**Further Information**

Please contact the authors for a demo.