Active Global Address Space (AGAS): Global Virtual Memory for Dynamic Adaptive Many-Tasking (AMT) Runtimes

Dissertation Research Showcase Presentation

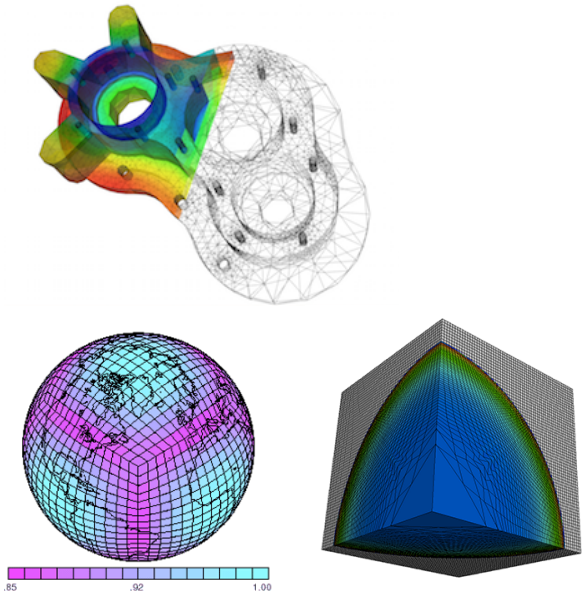
Abhishek Kulkarni

Center for Research in Extreme Scale Technologies

Indiana University

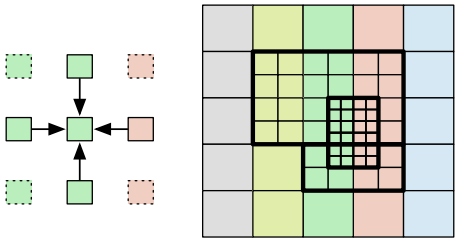
November 19, 2015

HPC Programming / Execution models



✓ Message Passing

- BSP: Compute-Communicate paradigm
- Data divided statically among available processors



✓ Asynchronous Many-tasking (AMT)

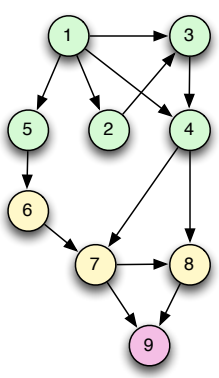
- Lightweight tasks concurrently operating on global data
- Well-suited for dynamic and adaptive execution

INDIANA UNIVERSITY
SCHOOL OF INFORMATICS AND COMPUTING

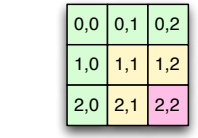
Asynchronous Many-Tasking (AMT) models

Task-parallel runtimes

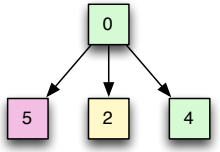
- Static/Dynamic task and dataflow graph
- Lightweight threads
- Inter-thread synchronization



Task Graph



Logical Representation



Data Distribution

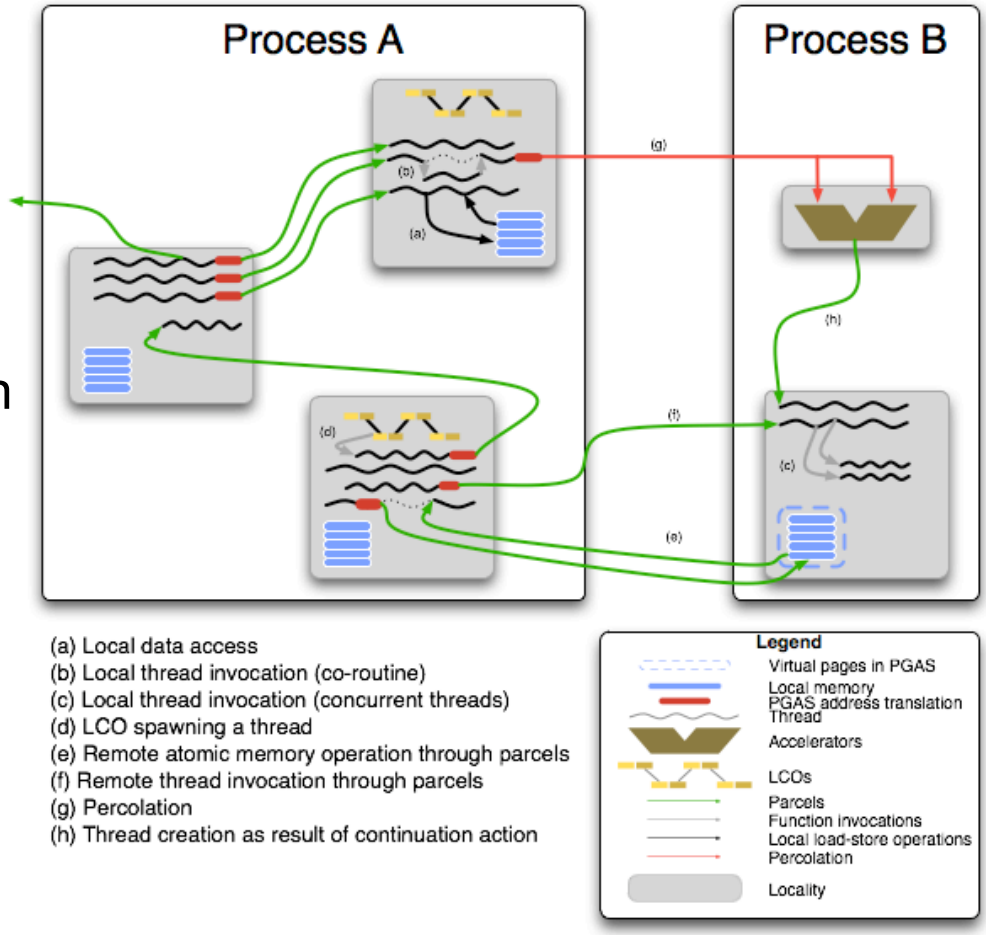
Distributed execution

- Active Messages
- Global Address Space

INDIANA UNIVERSITY
SCHOOL OF INFORMATICS AND COMPUTING
Bloomington

ParalleX Execution Model

- Lightweight multi-threading
  - Divides work into smaller tasks
  - Increases concurrency
- Message-driven computation
  - Move work to data
  - Keeps work local, stops blocking
- Constraint-based synchronization
  - Declarative criteria for work
  - Event driven
  - Eliminates global barriers
- Data-directed execution
  - Merger of flow control and data structure
- Shared name space
  - Global address space
  - Simplifies random gathers

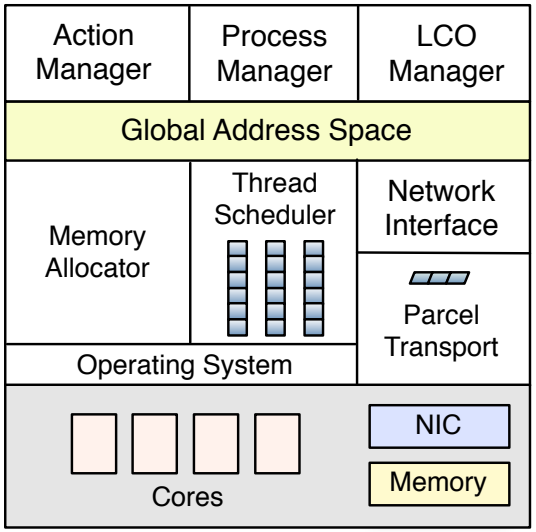


INDIANA UNIVERSITY
SCHOOL OF INFORMATICS AND COMPUTING
Bloomington

High Performance ParalleX

HPX-5

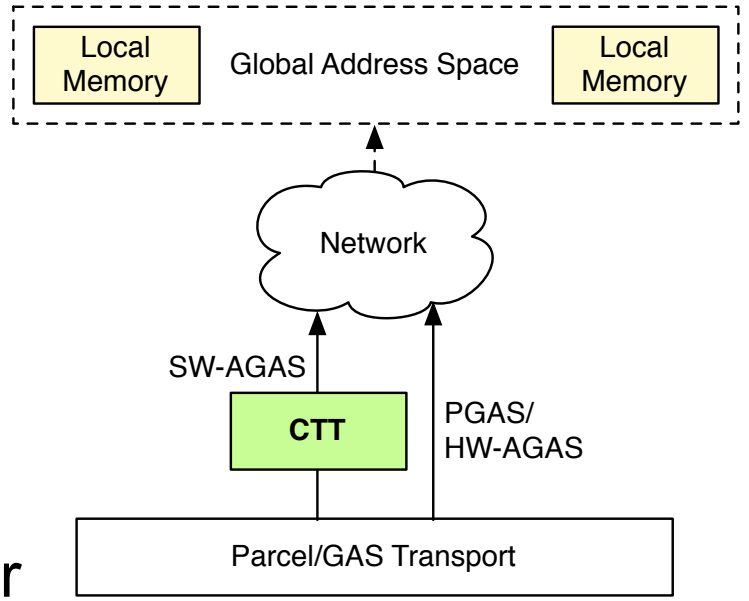
- The HPX runtime system reifies the ParalleX execution model:
  - Localities
  - Lightweight Threads
  - Processes
  - Local Control Objects (LCOs)
  - Parcels (Active Messages)
  - Active Global Address Space (AGAS)
- Fine-grained concurrency
  - Tasks, Interrupts, Lightweight Threads
- Lightweight, globally-addressable synchronization objects
  - Eg: Futures (IVars), Monotonic counters (Gate), Reductions



http://hpx.crest.iu.edu/

Global Address Space in HPX-5

- Global memory is globally addressable
- Supports local, cyclic, blocked and user-defined distributions
- Two modes of operations:
  - Static (PGAS)
  - Active (AGAS)
- Asynchronous access to global memory
- Memory must be pinned locally prior to access
- GAS attributes, atomics, collectives



Partitioned Global Address Space

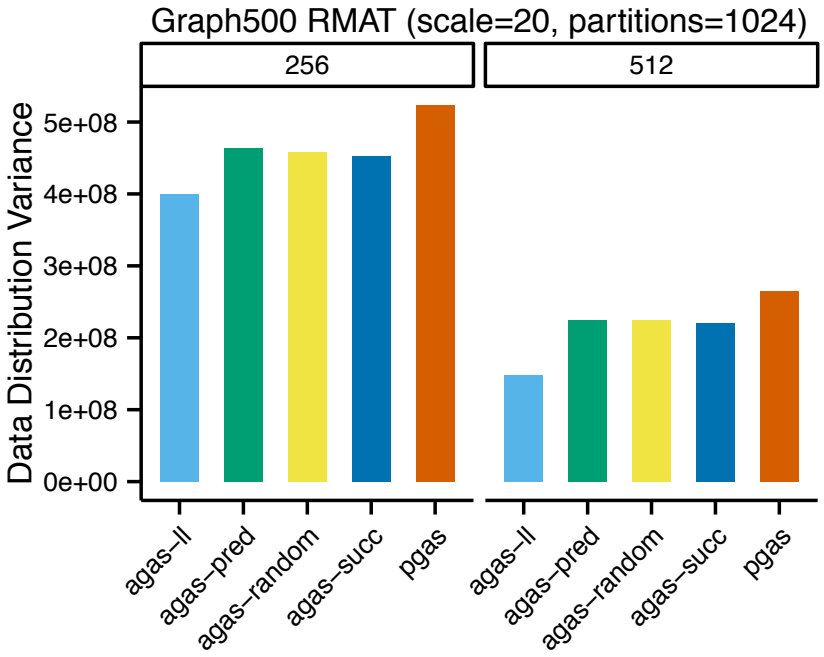
- Global Physical Memory
- Physical location encoded in the address
- Memory space same as address space
- Faster address translation
- Traditional PGAS supports put/get/atomics
- Static

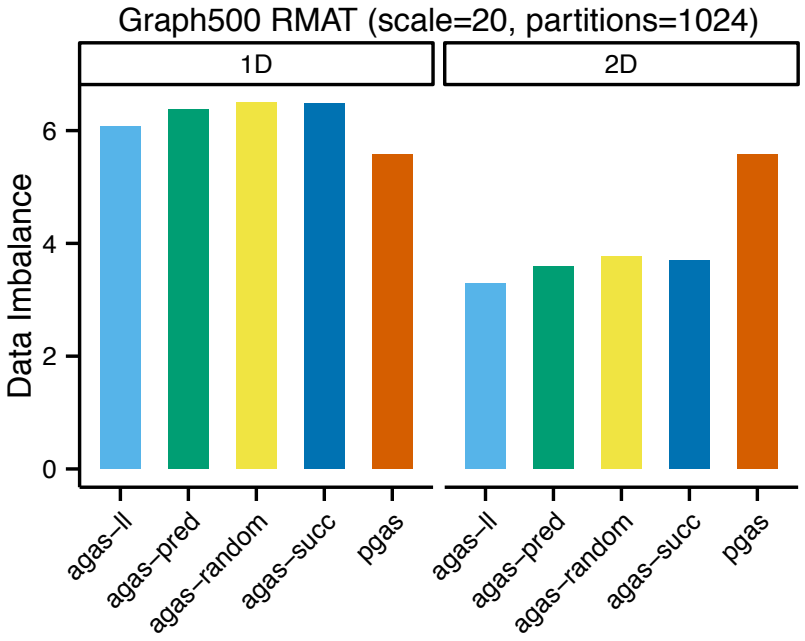
Active Global Address Space

- Global Virtual Memory
- Physical location maintained separately
- Memory space distinct from address space
- Address translation potentially expensive
- Support arbitrary AMs
- Dynamic (allows migration)

INDIANA UNIVERSITY
SCHOOL OF INFORMATICS AND COMPUTING
Bloomington

Load and Data Imbalance in Graph Algorithms





Move the smallest partition to:

- Successor (agas-succ)
- Predecessor (agas-pred)
- Random (agas-random)
- Least Loaded (agas-ll)

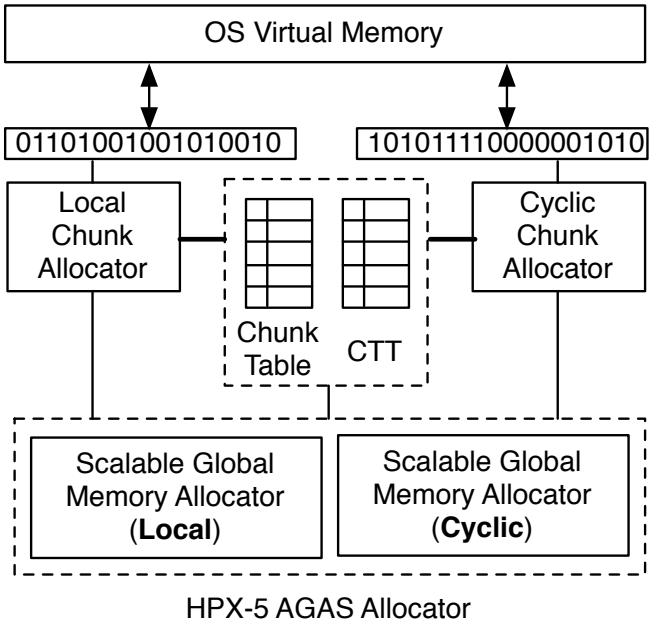
INDIANA UNIVERSITY
SCHOOL OF INFORMATICS AND COMPUTING

AGAS Features

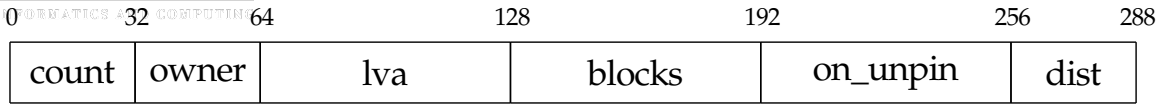
- User-defined data distributions
  - Existing PGAS models support cyclic, block-cyclic
  - Similar to Chapel, ZPL
  - Higher-order and composable
- Data Migration
  - Explicit moving of individual chunks
  - Rebalancing of global data
  - Implicit runtime-managed remapping based on introspection
- Co-data Actions
  - Relational dependencies between tasks and data
  - Runtime chooses between a) moving data to computation or b) moving computation to data
- Caching, Data stealing, speculative migration

Software AGAS

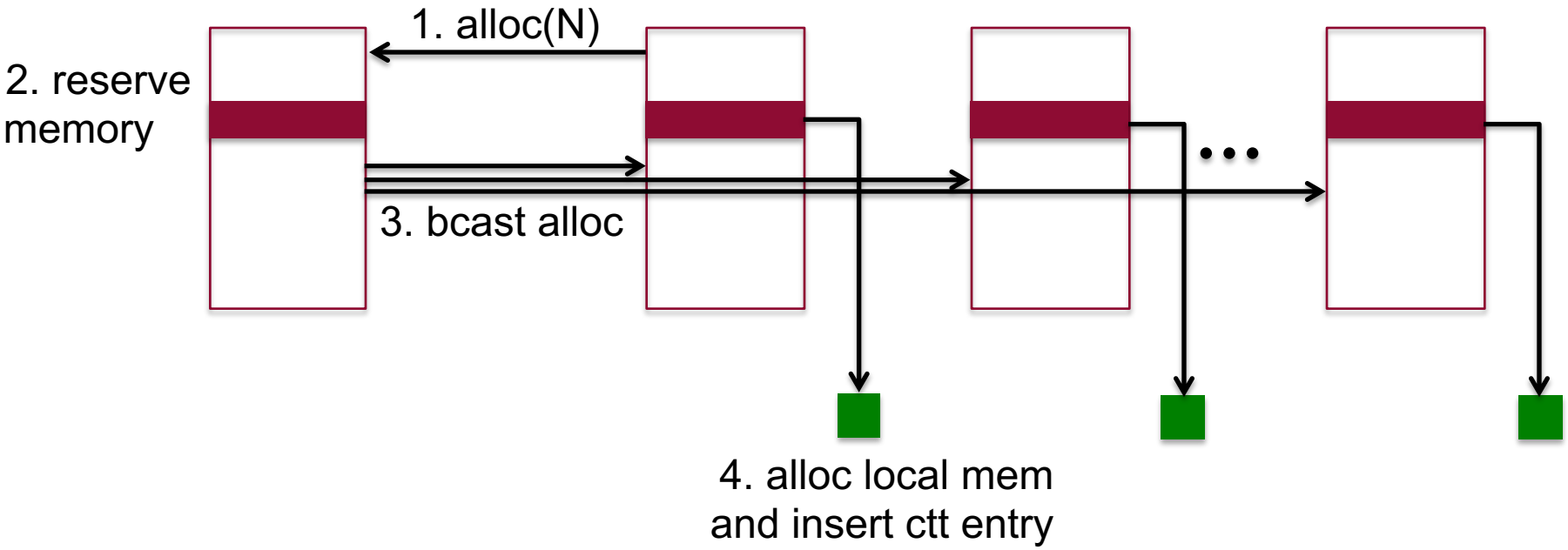
- Flat, byte-addressable global address space
- Chunk size limited to 4TB, Nodes limited to 64K
- Scalable memory allocators handle local/cyclic segments
  - jemalloc, TBBMalloc
- Chunk table maps LVA → GVA
- CTT (Chunk Translation Table) maps GVA → LVA
- CTT entry:



HPX-5 AGAS Allocator



Cyclic Allocation

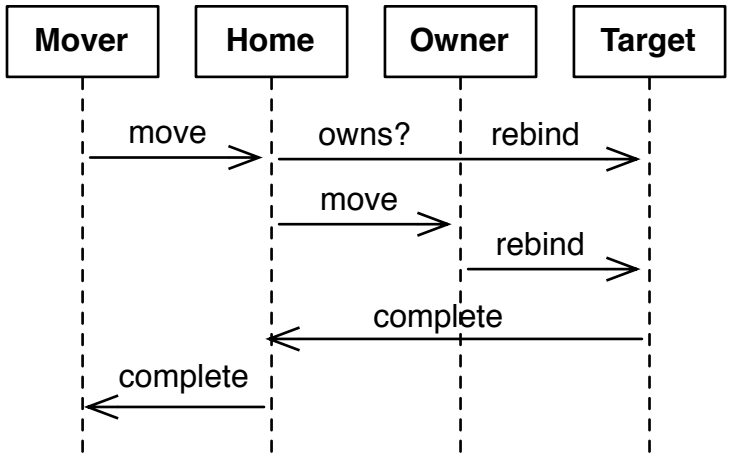


- Symmetric cyclic segment/heap
- Can be segmented to speed up cyclic allocation
  - At the expense of limiting allocation size

INDIANA UNIVERSITY
SCHOOL OF INFORMATICS AND COMPUTING

Move

- When a chunk moves, its *owner* is updated
- Move protocol
  - Overlaps routing updates and data transfer

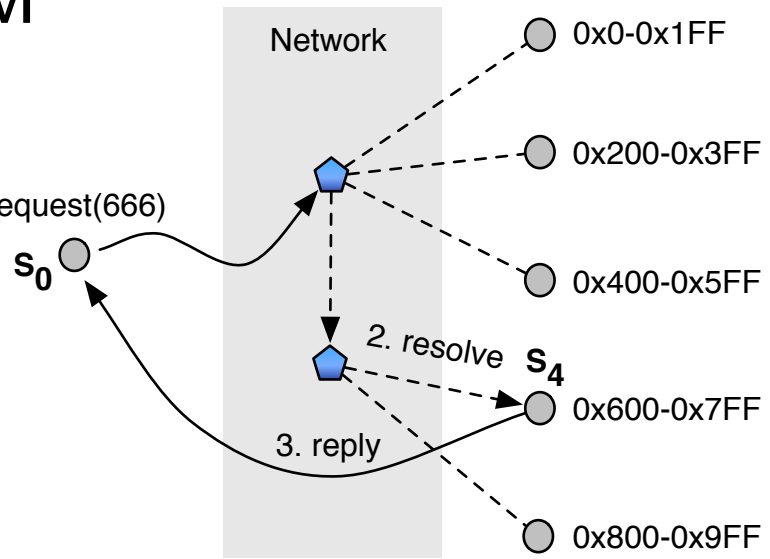


- Home always "knows" where a chunk is
- Current implementation uses a two-sided transport
  - Relies on being able to "resend" a parcel
- Complex interactions between move, unpin and delete

INDIANA UNIVERSITY
SCHOOL OF INFORMATICS AND COMPUTING

Hardware AGAS

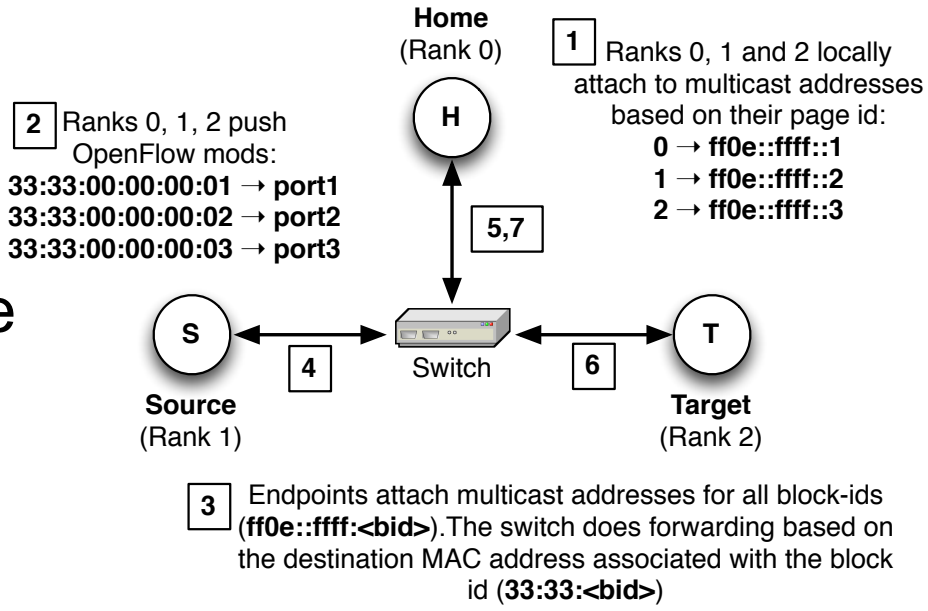
- Network-managed global address space
- CTT implemented by TCAM in hardware
- The network "knows" the location of chunks
- Pros:
  - Message optimal in terms of hops required for routing
- Cons:
  - Centralized directory
  - Existing hardware limitations

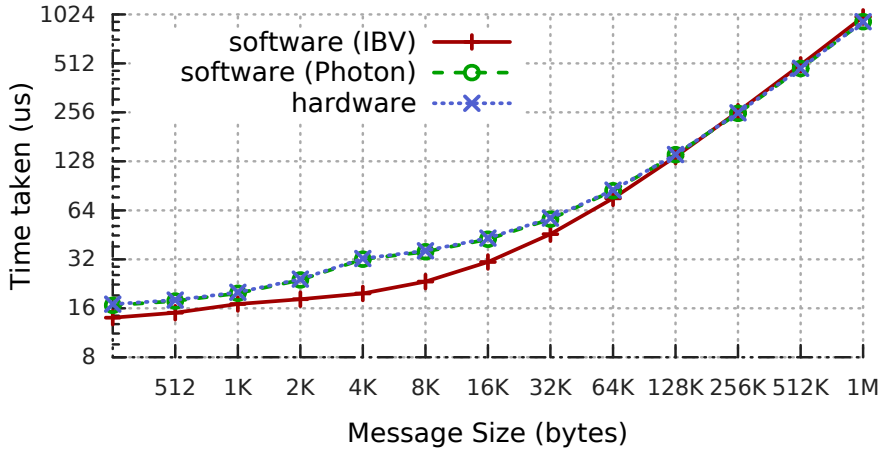


INDIANA UNIVERSITY
SCHOOL OF INFORMATICS AND COMPUTING

Hardware AGAS

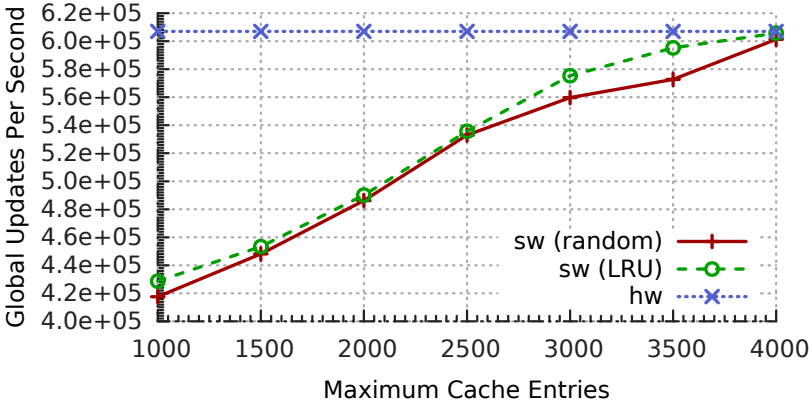
- Proof of concept based on the GASNet runtime system
- Uses SDN for routing table management
- Uses IB RD multicast
- IB multicast GIDs are mapped to multicast Ethernet MAC addresses
- Put performance:
  - High switching latency
  - Photon conduit overheads

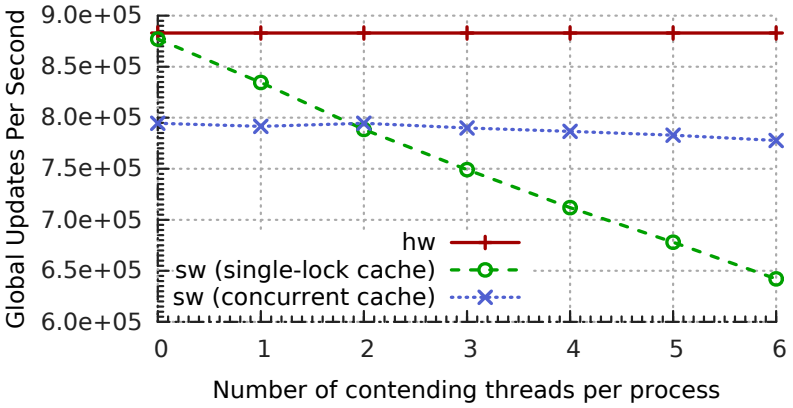




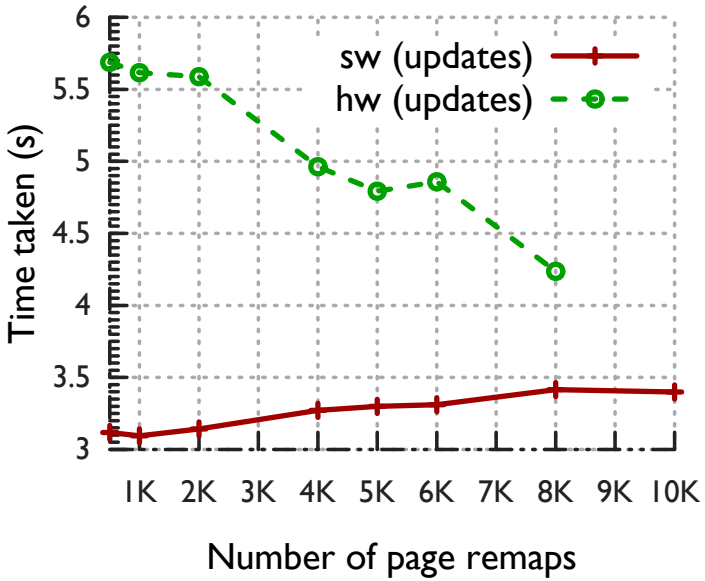
Comparing Hardware and Software AGAS

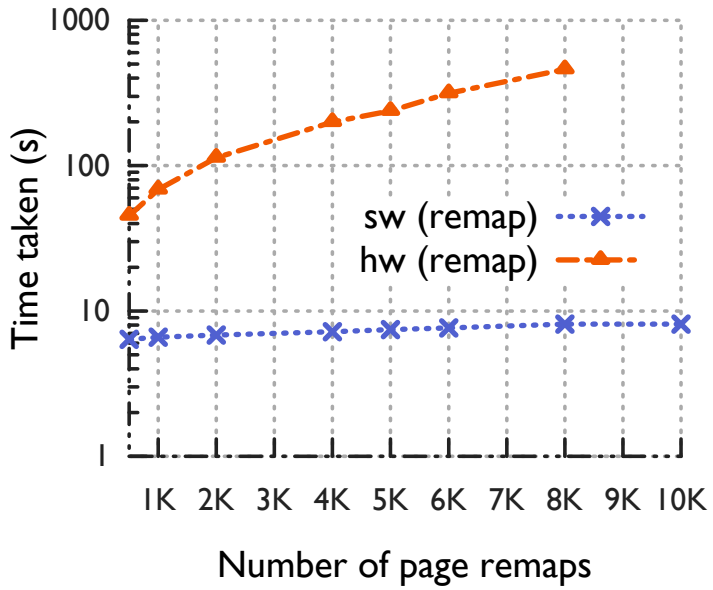
- Bounded software cache
  - Cache-replacement policies (random, LRU)
  - Evictions cause cache misses and degrade GAS perfomance
- Thread contention in concurrent cache
  - Concurrent cache is faster than a single-threaded cache but 25% slower than the hardware implementation





Remap performance in HW AGAS





- GUPS microbenchmark
  - table size 2048 words, 512 words/page, 16 nodes (192 cores)
  - 4 million random updates were performed
- As page movement frequency increases, SW becomes more expensive than HW

Conclusions

- The HPX-5 runtime system provides dynamic, adaptive features necessary for execution of large-scale irregular applications.
- Global virtual memory using an active global address space (AGAS) enables the runtime to manage both locality and load-imbalance concerns.
- Network-managed AGAS shows promise at smaller scales but requires support from vendors and supercomputing centers to be viable at larger scales.

Questions?


INDIANA UNIVERSITY
SCHOOL OF INFORMATICS AND COMPUTING
Bloomington