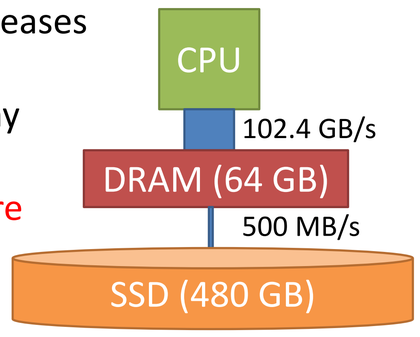
vGASNet: A PGAS Communication Library Supporting Out-of-Core Processing

Ryo Matsumiya and Toshio Endo
Tokyo Institute of Technology

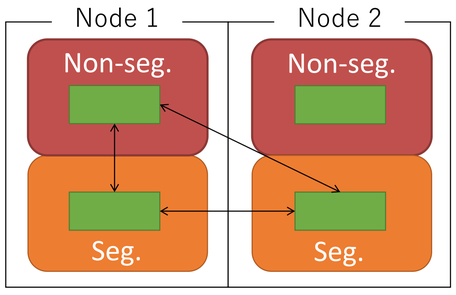
Background

- Partitioned Global Address Space (PGAS) eases distributed programming.
- Out-of-core processing is required in many fields.
- Few PGAS frameworks support out-of-core processing.
- Node-local SSDs are available for out-of-core processing.



vGASNet overview

- Remote memory access based communication library.
- The interface is similar to GASNet [1].
- Two memory regions are available.
  - Segmented memory region
    - Can be accessed by other nodes.
    - Allocated in node-local SSDs with vgasnet_allocate().
  - Non-segmented memory region
    - Local access only.
    - Allocated in DRAM with general memory allocate functions (e.g. malloc()).



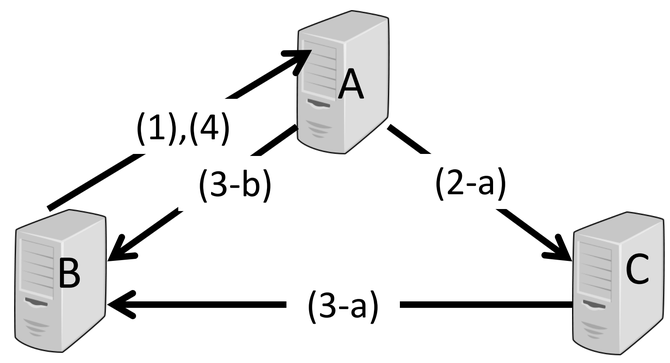
Cache mechanism overview

- SSD is much slower than DRAM.
- Under vGASNet, each node has their own cache pool on DRAM.
  - Page-based cache
- Reducing the amount of accesses to the SSDs, vGASNet adopts cooperative caching mechanism.
  - Using not only local caches but also remote caches.
  - Firstly implemented in a distributed file system [2].

Cooperative caching

- Each page is originally stored in the SSD of its *owner*.
- Each node has a *cache table*, which assigns its own pages with the node whose DRAM stores the cache.
- The rough flow of forwarding cache is below.
  - In this example, node B is to receive a page of node A.

(1) Node B requests node A.
(2) Node A refers its own cache table.
(2-a) When node C has a cache of the page, node A forwards the request to node C.
(3-a) Node C sends the page cache to node B.
(2-b) When no node has any caches of the page, node A reads the original page from its SSD to the buffer.
(3-b) Node A sends the buffer to node B.
(4) When node B has received the cache, node B requests node A to register the cache with node A's cache table

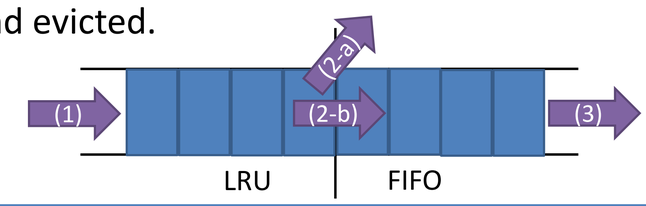


- Guaranteeing cache consistency is challenging.
- In vGASNet, MOESIF protocol is implemented as a cache coherence protocol.
- MOESIF protocol is based on two practical protocol.
  - MOESI protocol [3]
    - Used in AMD 64-bits multicore processors.
    - Dirty caches are not evicted if the same page is cached in another node.
  - MOSIF protocol [4]
    - Used in Intel multicore processors.
    - The node whose cache is used to forward the cache is specified per cache line.

Cache replacement policy

- LRU-based policy
- Consists of pure a LRU queue and a FIFO queue.
  - The size of the FIFO queue is half of the cache pool.
  - The size of the LRU queue is the rest of the cache pool.

(1) All stored caches are enqueued to the LRU queue firstly.
(2) When the cache pool is filled, a cache is dequeued from LRU queue.
(2-a) If any other nodes have the same cache line, the cache is evicted.
(2-b) Otherwise, the cache is enqueued to the FIFO queue.
(3)When the FIFO queue is filled, the bottom of the FIFO queue is dequeued and evicted.

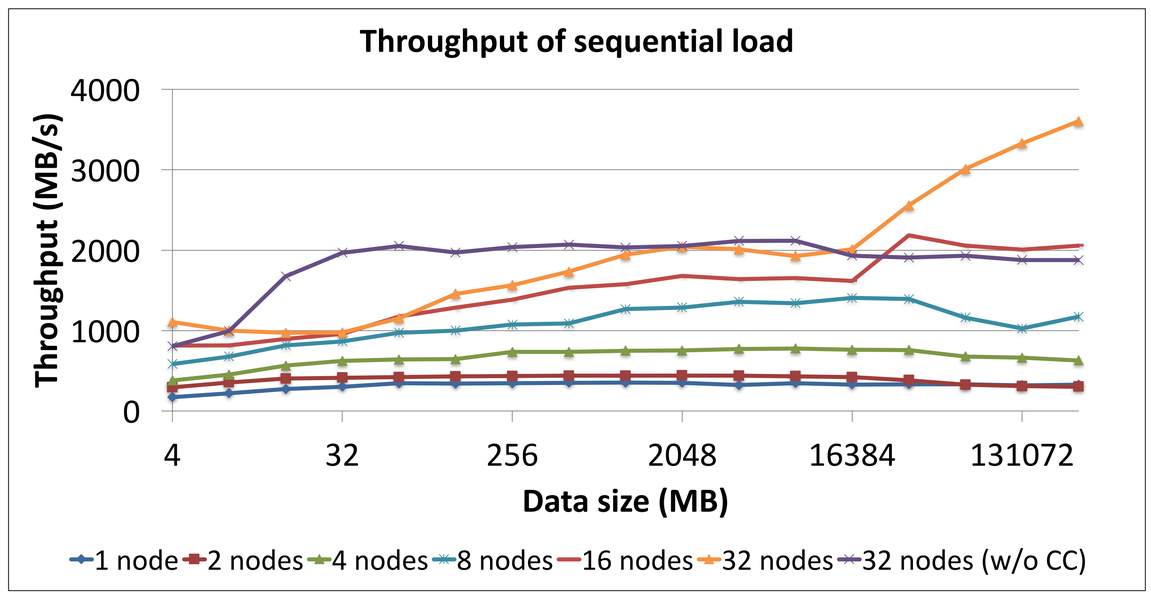


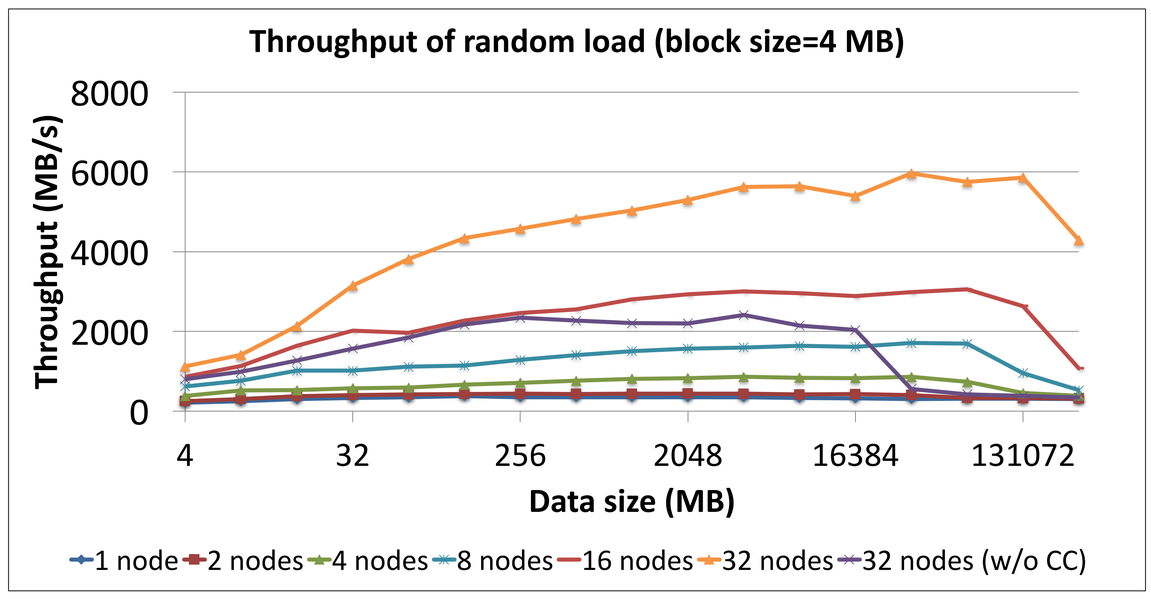
Performance evaluation

- Conducted with TSUBAME-KFC/DL [5].
- Page size is 4 MB.
- The cache pool size of each node is about 16 GB.

CPU	Xeon E5-2620v2 × 2	Network	Infiniband 4x FDR
DRAM	DDR3-1600 64 GB	OS	CentOS 7.3
SSD	SATA3 480 GB	Filesystem	XFS

- The performance is evaluated by loading the data stored in one node.
- (w/o CC) means cooperative caching is not used.





Future work

- Implementing a practical PGAS runtimes.
  - Undergoing: UPC++
- Performance evaluation using practical applications.
  - Numerical solver, machine learning, genetic analysis, etc.

Related work

- ComEx-PM [6]
  - A PGAS communication library supporting out-of-core processing.
  - The cache mechanism depends on Linux VFS cache
- HHRT [7], Papyrus [8]
  - Ease MPI programs of supporting out-of-core processing.
  - Unlike vGASNet, their interface is not based on remote memory access.

[1] Chan and Igual, Runtime Data Flow Graph Scheduling of Matrix Computations with Multiple Hardware Accelerators, FW Note, 50 (1996)
[2] Dahlin et al., Cooperative Caching: Using Remote Client Memory to Improve File System Performance, in Proc. of OSDI '94
[3] AMD, AMD64 Architecture Programmer's Manual Vol.2 System Programming.
[4] Kanter, The Common System Interface: Intel's Future Interconnect, Real World Tech 5.
[5] Endo et al., TSUBAME-KFC: A Modern Liquid Submersion Cooling Prototype Towards Exascale Becoming the Greenest Supercomputer in the World, in Proc. of ICPADS '14
[6] Matsumiya and Endo, PGAS Communication Runtime for Extreme Large Data Computation, in Proc. of ESPM2 '16
[7] Endo, Realizing Out-of-Core Stencil Computations using Multi-Tier Memory Hierarchy on GPGPU Clusters, in Proc. of Cluster '16
[8] Kim et al., Design and Implementation of Papyrus: Parallel Aggregate Persistent Storage, in Proc. of IPDPS '17

This work is supported by JST-CREST, "Software Technology that Deals with Deeper Memory Hierarchy in Post-petascale era".