Design and Implementation of OpenSHMEM using OFI on the Aries interconnect

Seager Kayla, Sung-Eun Choi (Cray), James Dinan, Howard Pritchard (LANL), Sayantan Sur

8/3/16

I. Motivation & background

II. Libfabric / Open Fabrics Interface (OFI)

III. OFI-uGNI

IV. SHMEM-OFI-uGNI

V. Results on CORI

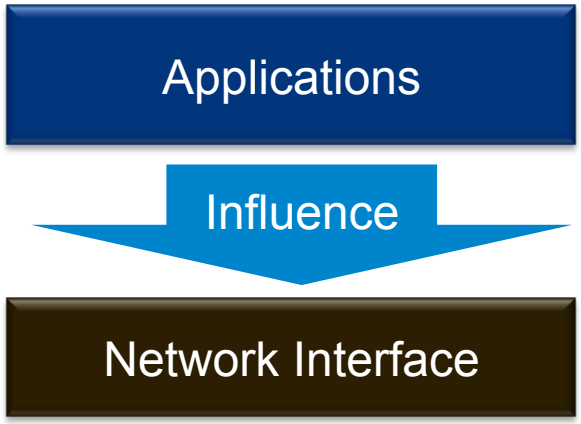
VI. Concluding remarks

Recap: OpenSHMEM-OFI ~ OFA driven

- OFA's OFI working group created *Libfabric*
- Top Down Designed (*app centric*)
  - users defined requirements
  - minimal software overhead

OPENFABRICS ALLIANCE

"Defines the high performance network interface standard"



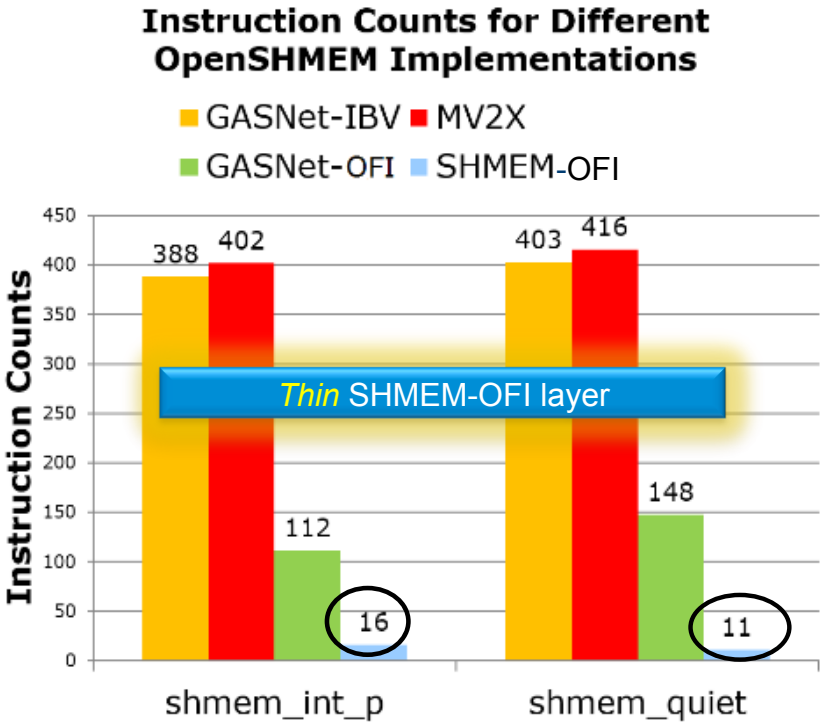
App Centric Design = improved Performance & Scalability

Recap: OpenSHMEM-OFI enabling

Evaluating and driving specifications

- OpenSHMEM 1.3 compatible
- Libfabric
  - Evaluated on latest OpenSHMEM spec
  - pacing Libfabric changes

Missing true evaluation on real hardware



Motivation for Cray* Aries

1. Contemporary high performance RDMA interconnect

2. Enable users with Aries/Cray interconnect to run SHMEM-OFI

3. Test against Cray-SHMEM

I. Motivation & background

II. Libfabric (OFI)

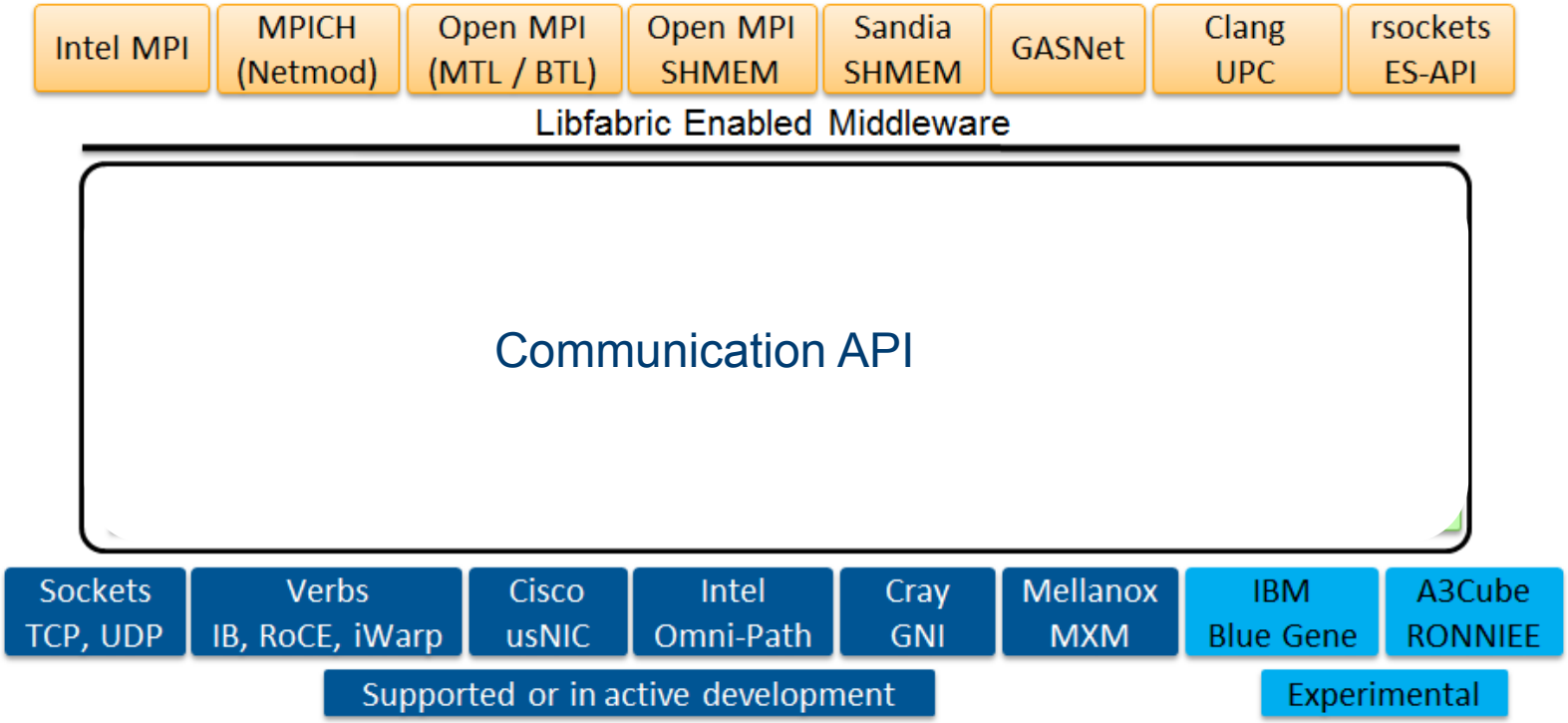
III. OFI-uGNI

IV. SHMEM-OFI-uGNI

V. Results on CORI

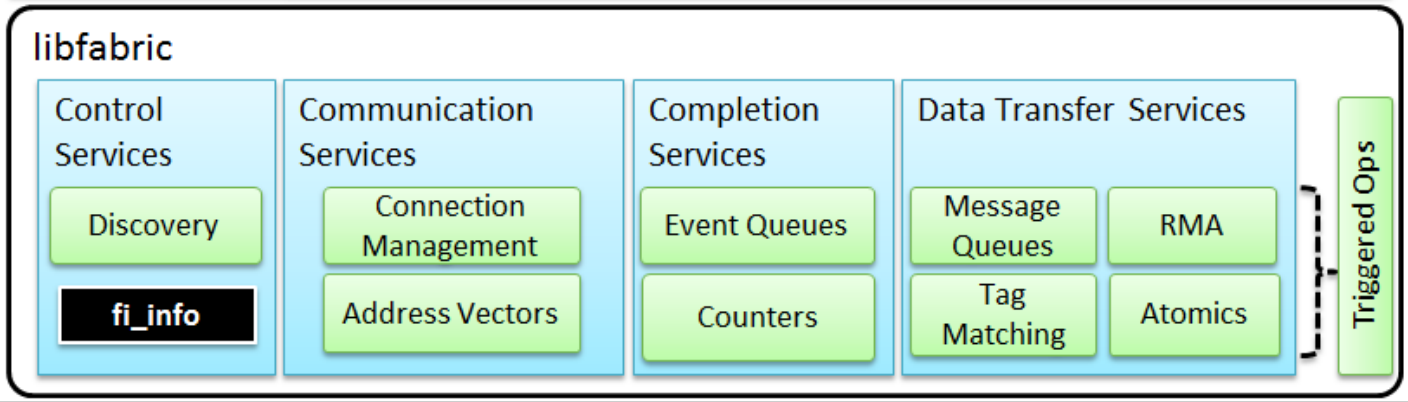
VI. Concluding remarks

Libfabric enabling Overview



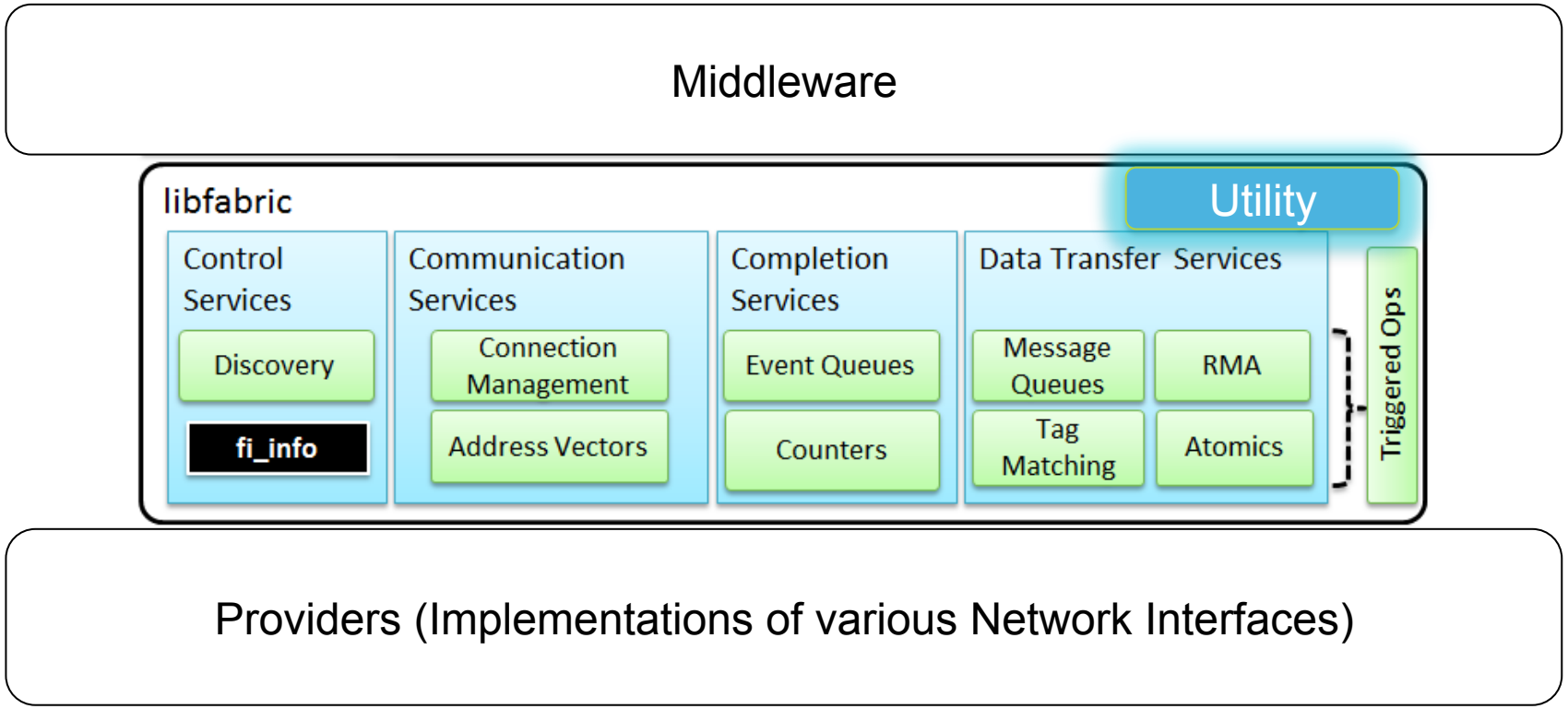
Libfabric API set

Middleware



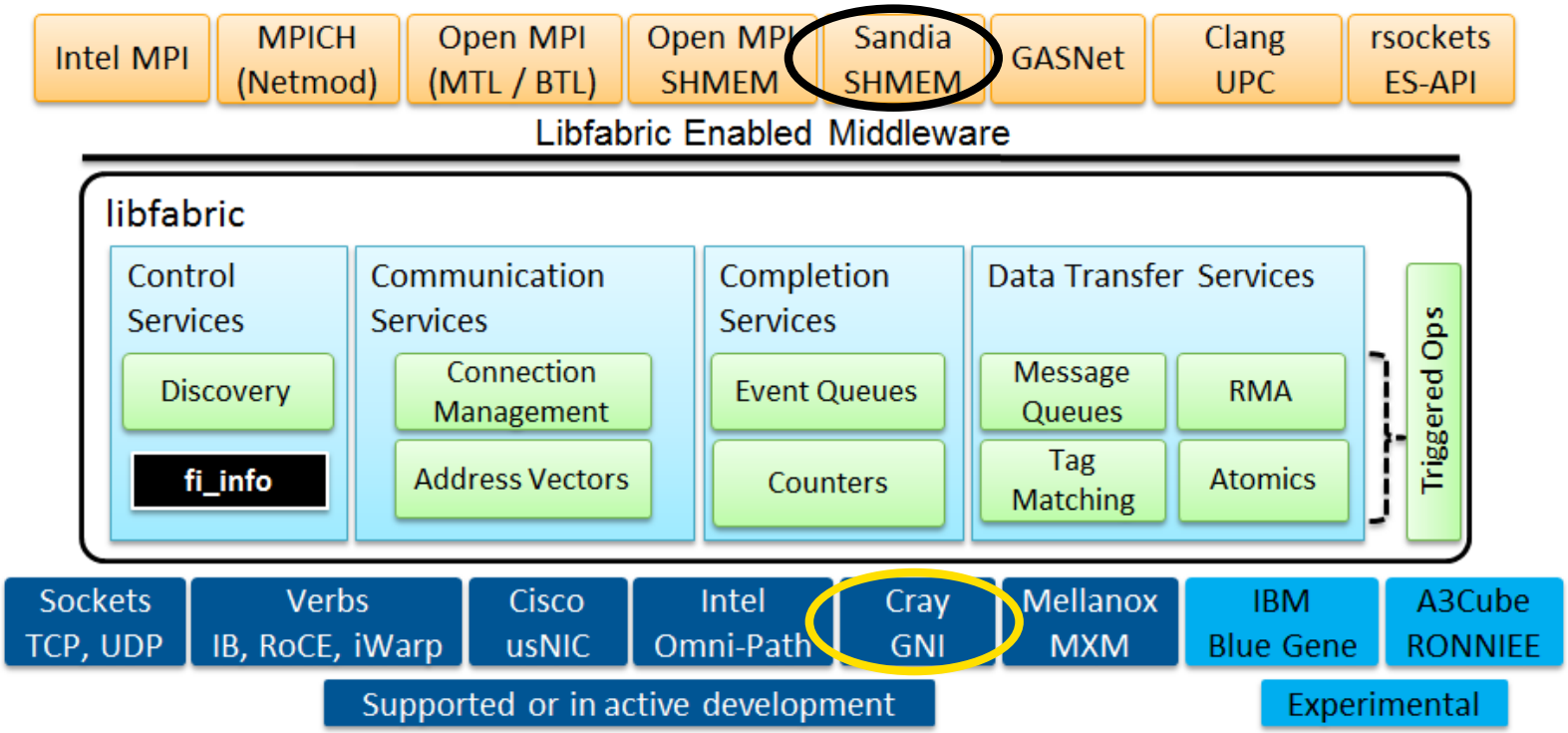
Providers (Implementations of various Network Interfaces)

Libfabric API set: “Utility Provider” NEW



Utility Provider: enhances some providers capability set

Libfabric: full picture



I. Motivation & background

II. Libfabric (OFI)

III. OFI-uGNI

IV. SHMEM-OFI-uGNI

V. Results on CORI

VI. Concluding remarks

OFI-uGNI Provider Capabilities: Part 1

- RMA & Atomic Messaging
  - Small messaging: FMA engine (fast memory access)
  - Large messaging: BTE (bulk transfer engine)
  - Hardware Atomic Support
    - 32/64-bit atomic operations
- Addressing:
  - Address Vector (AV) map & AV table support
    - utilizes hardware protection

OFI-uGNI Provider Capabilities : Part 2

- Memory usage (MR_BASIC)
  - Memory registration cache support
  - Provider sets MR key (cannot support MR scalable)
- Completion: Queues
  - Queues with callback
    - counter is emulated with Queue
- Connection-oriented: emulates connectionless
  - Internal connection management

I. Motivation & background

II. Libfabric (OFI)

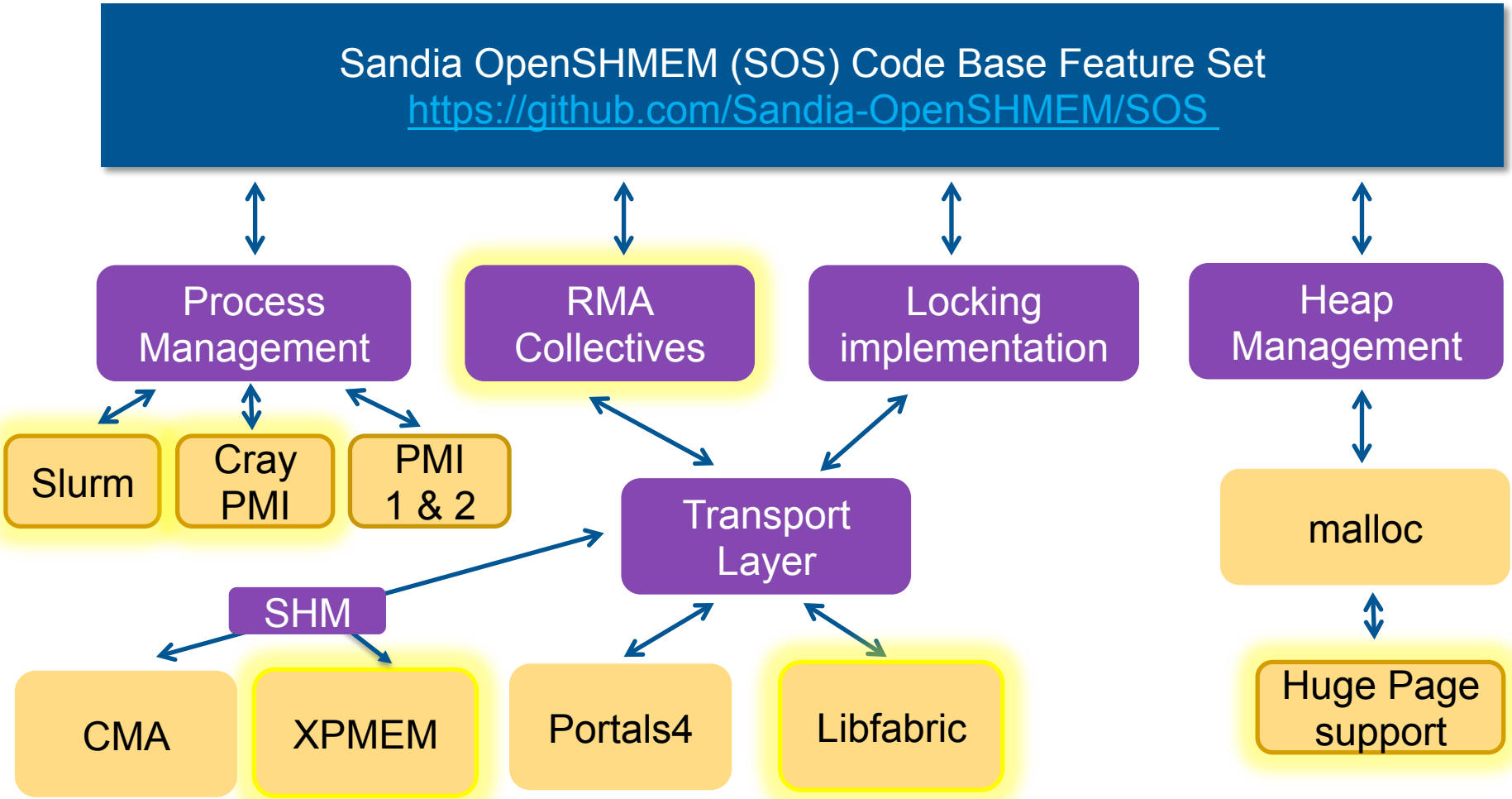
III. OFI-uGNI

IV. SHMEM-OFI-uGNI

V. Results on CORI

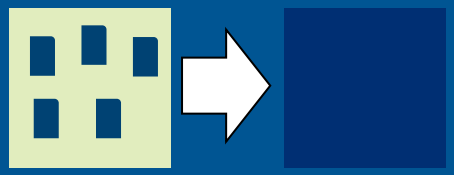
VI. Concluding remarks

Changes to codebase to enable OFI-uGNI



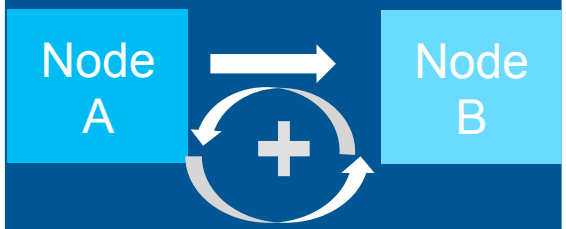
OFI capability set for SHMEM *required* from Provider

Scalable, memory usage model



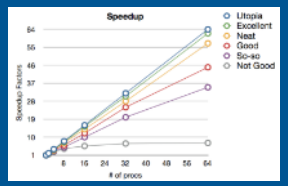
VA communication style, expose full VA range

Rich Atomic Set



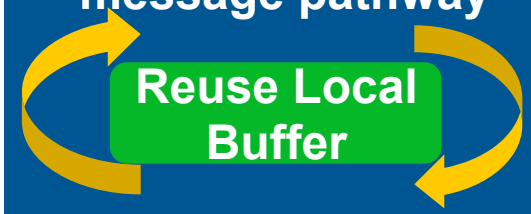
Full SHMEM converge with fine API granularity

Scalable Endpoint



Reliable connectionless endpoints with logical

Accelerated small message pathway



Immediate local completion: “fi_inject”

PGAS Implementation

Hardware Fabric

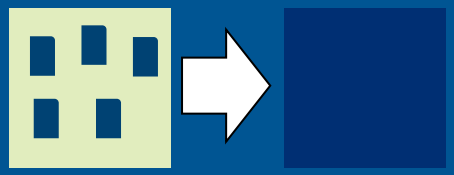
Efficient Remote Completion for Put/Get

counters for lightweight remote completion

OFI-uGNI Provider capability Support

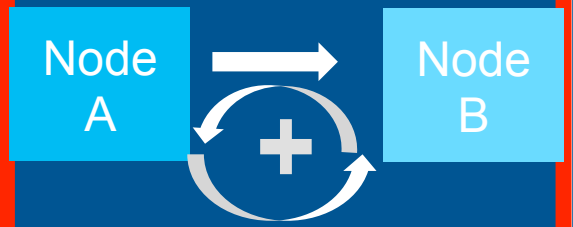
Not Supported

Scalable, memory usage model



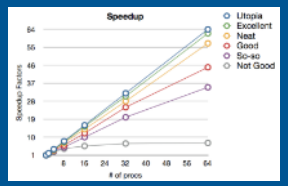
VA communication style, expose full VA range

Rich Atomic Set



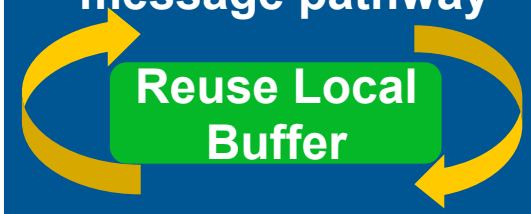
Full SHMEM converge with fine API granularity

Scalable Endpoint



Reliable connectionless endpoints with logical

Accelerated small message pathway



Immediate local completion: “fi_inject”

PGAS Implementation

Hardware Fabric

Efficient Remote Completion for Put/Get

counters for lightweight remote completion

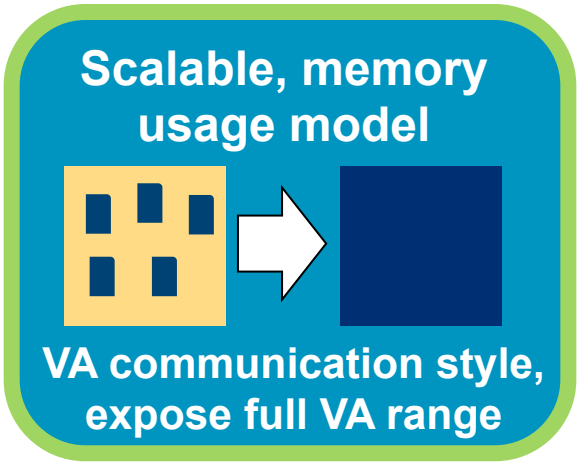
SHMEM-OFI: Memory Model Solution

SHMEM-OFI Support:

- Added MR_BASIC
  - Provider sets MR key
  - Track unique key per PE
  - Compile-time option
- *New huge page feature*

OFI-uGNI Support:

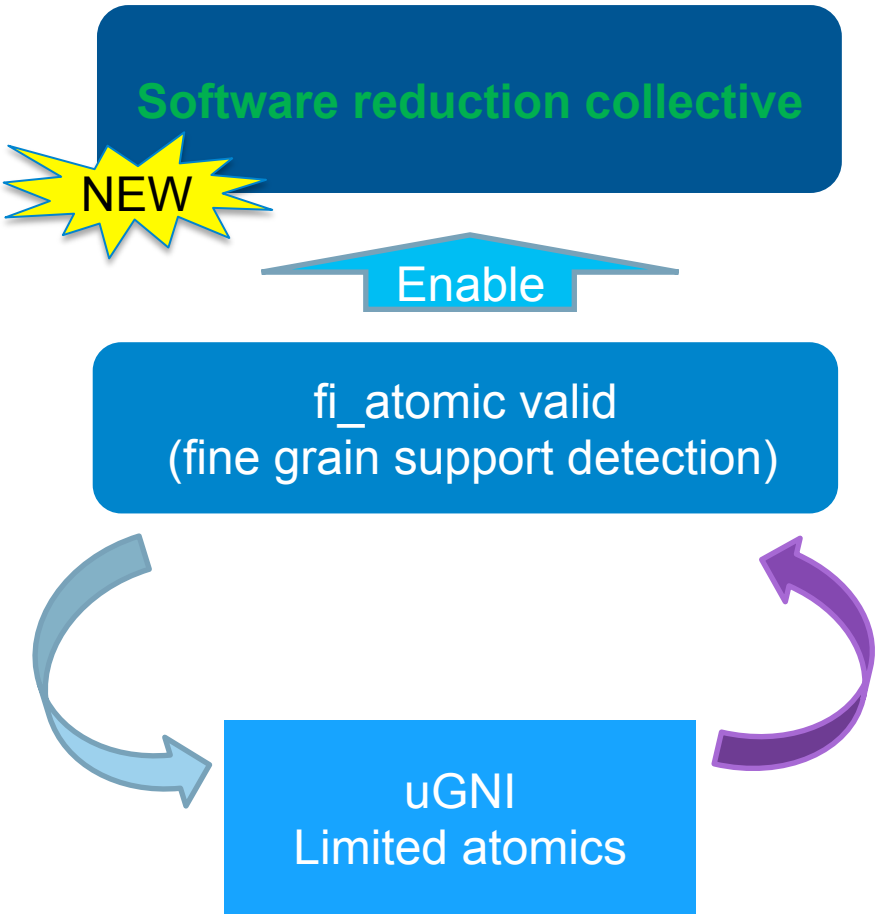
- Memory segment management
  - Emulates full VA range



Adding MR Basic enabled uGNI Support

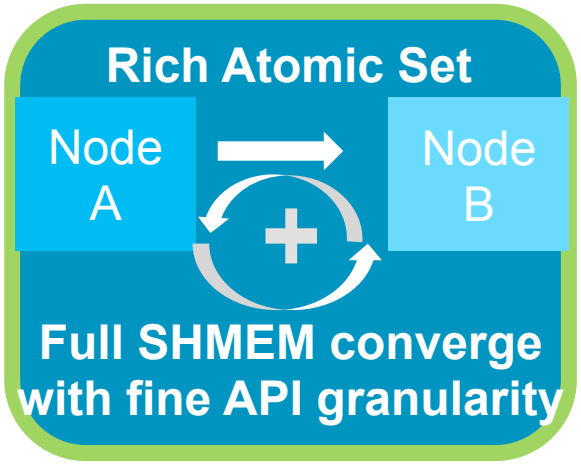
SHMEM-OFI: Atomic Support Solution

SHMEM-OFI Added Support:



OFI-uGNI support:

- Point-point Atomic limitations



I. Motivation & background

II. Libfabric (OFI)

III. OFI-uGNI

IV. SHMEM-OFI-uGNI

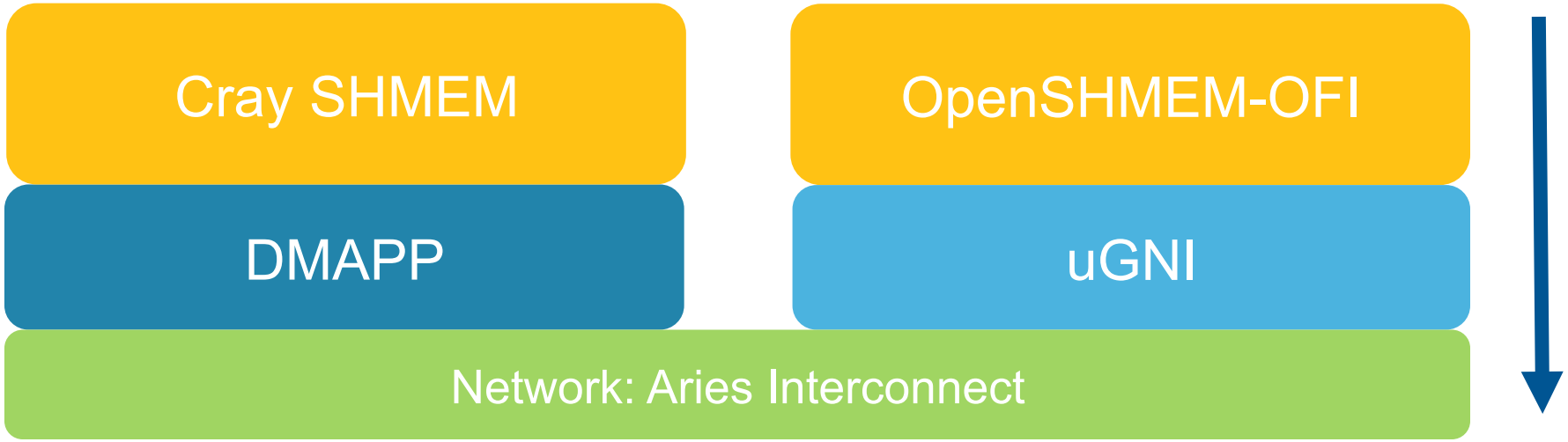
V. Results on CORI

VI. Concluding remarks

Results: Environment Used

- Cori
  - Cray* XC40
  - Cray* Aries high-speed interconnect with Dragonfly* topology
  - 1,630 computes nodes (Intel* Xeon “Haswell”)
    - 16 cores/socket
    - 128 GB memory per node
  - CLE (Cray Linux*) and Slurm*
  - No static builds were used
  - Utilized huge page support
  - https://www.nersc.gov/users/computational-systems/cori/configuration/

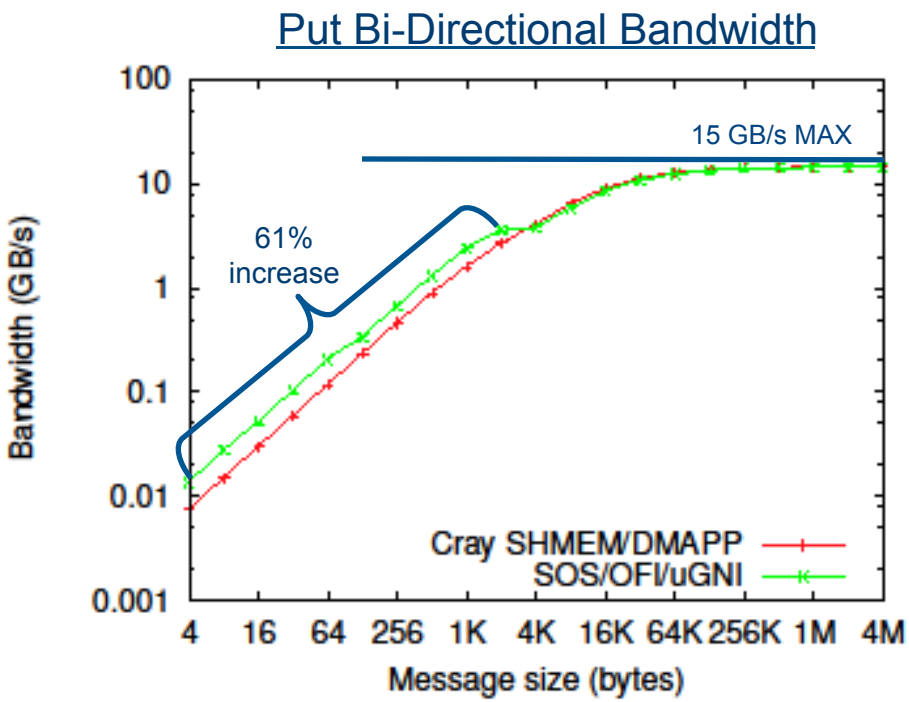
Cray SHMEM and SHMEM-OFI: Different network interfaces

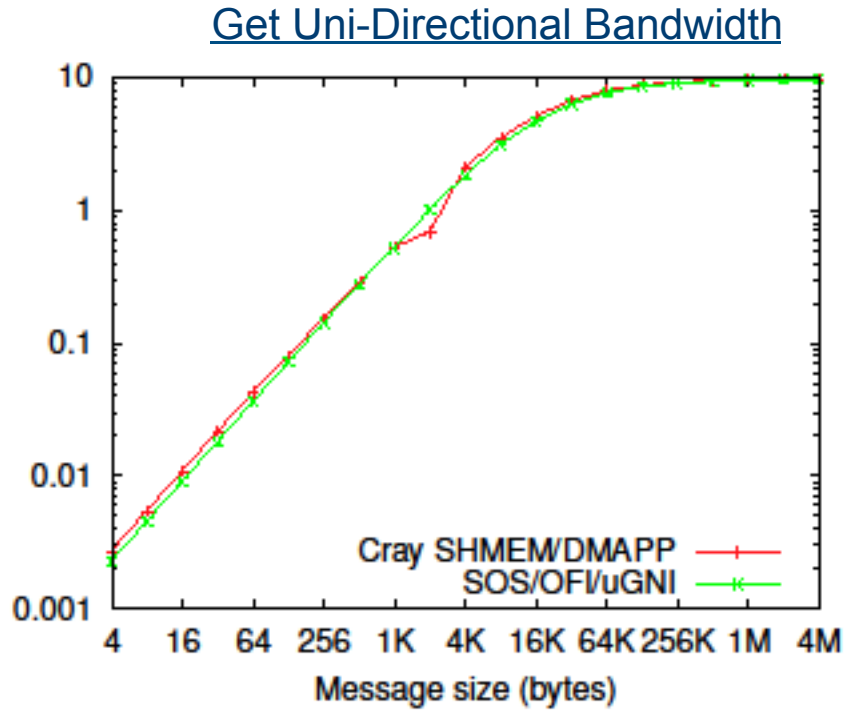


uGNI vs DMAPP: Different Application Focus

- uGNI
  - Designed to enable two-sided communication (MPI) *and* PGAS
    - Best choice for OFI
  - *Optimized for large messages*
- DMAPP
  - Designed for PGAS
  - *Optimized small message support*
  - *unique hardware mechanism for management PCI-e write credits (XC40)*

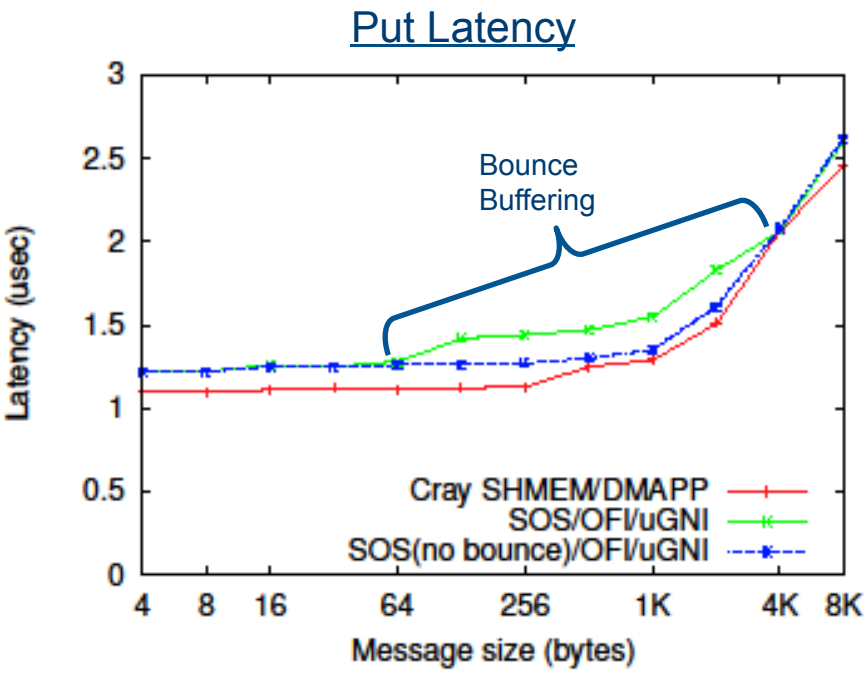
Bandwidth

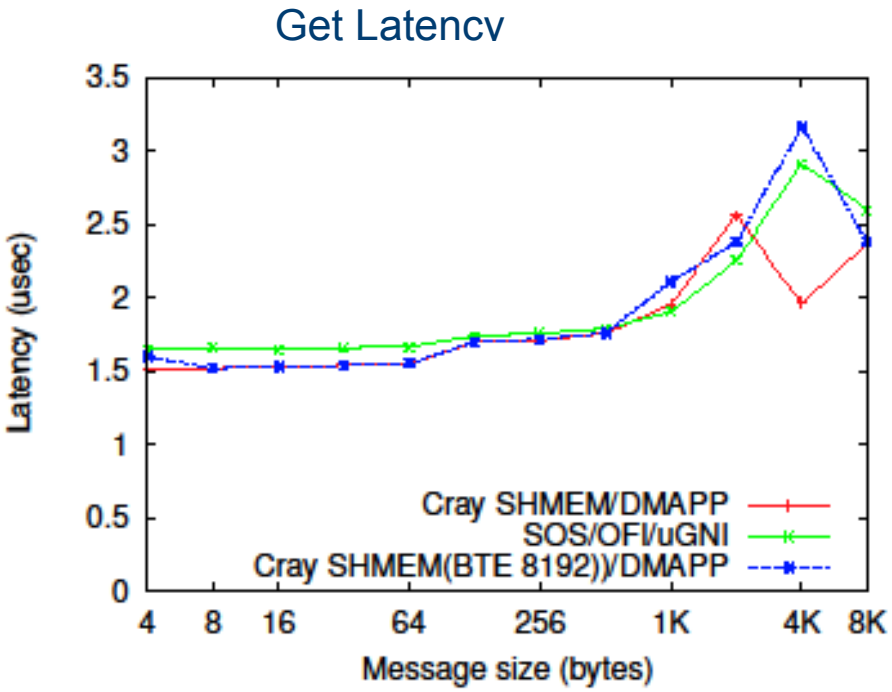




- Blocking Put Bi-Directional BW: 61%~ improvement
  - fi_inject (0-64 bytes), Bounce Buffering (up to 1KB)
- Blocking Get BW within 2%~

Latency



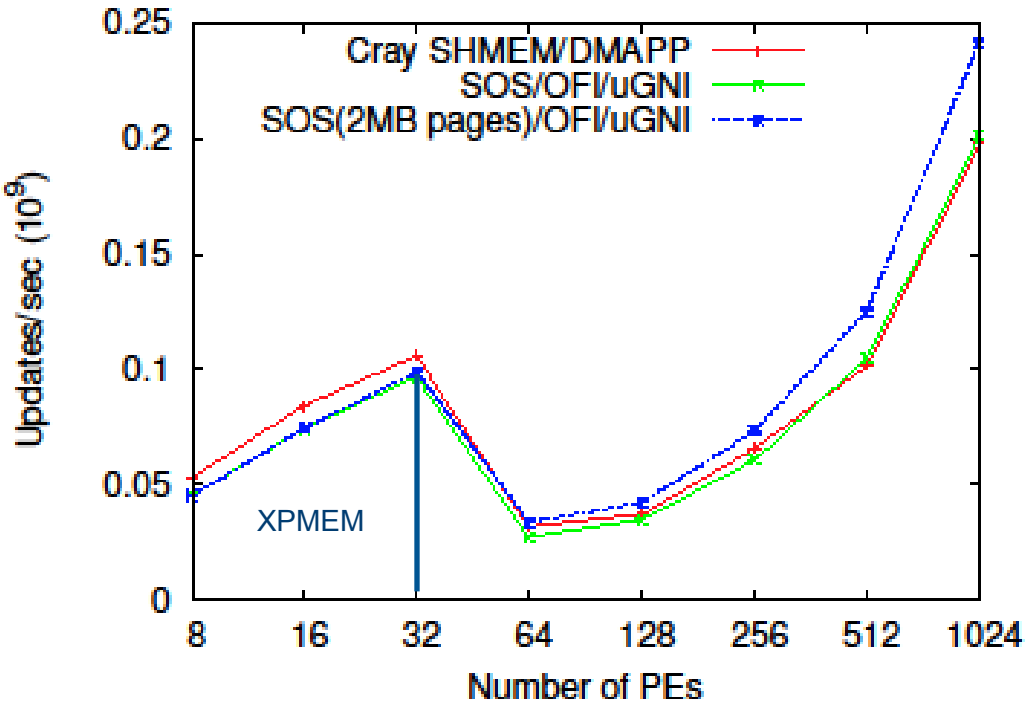


- Comparable Blocking Get/Put Latency
  - Sandia SHMEM-OFI-uGNI latency: 1.22 microseconds
  - at most 150ns~ delta (due to uGNI vs DMAPP)
- Bounce buffering adds small overhead

NOTE:
SHMEM-OFI-uGNI BTE Default **8KB**

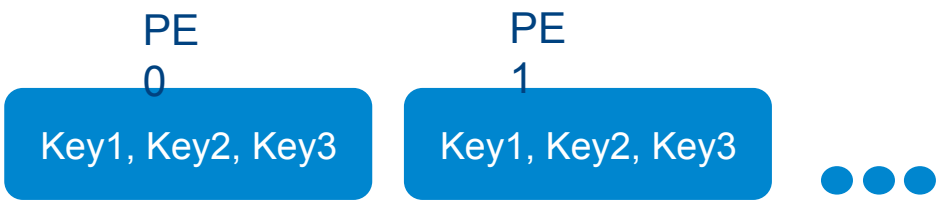
GUPS: Scaling to 1024 PEs

- Random access benchmark
  - Small message size
    - Shmem_longlong_g/p
- SOS 2MB pages
  - Cray Default
- SOS does better at scale
  - inject functionality
- *Coming soon to SOS Performance Suite*



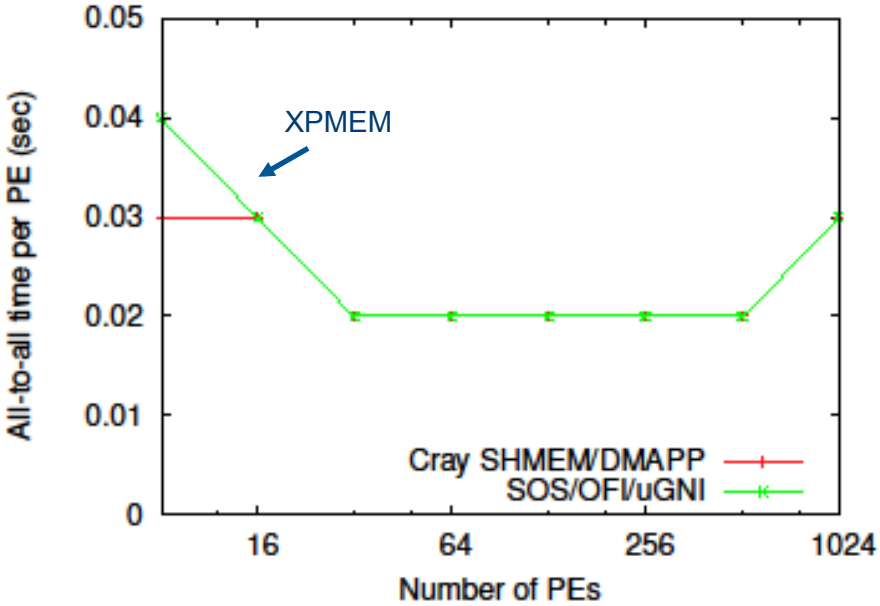
NAS ISx (Integer Sort)

- Benchmark characteristics
  - shmem_int_put(# of keys) ~ (all-to-all)
    - Large message sizes
  - Weak scaling:
    - fixed keys/PE
  - Strong scaling:
    - same number of keys *across PE set*
- https://github.com/ParRes/ISx

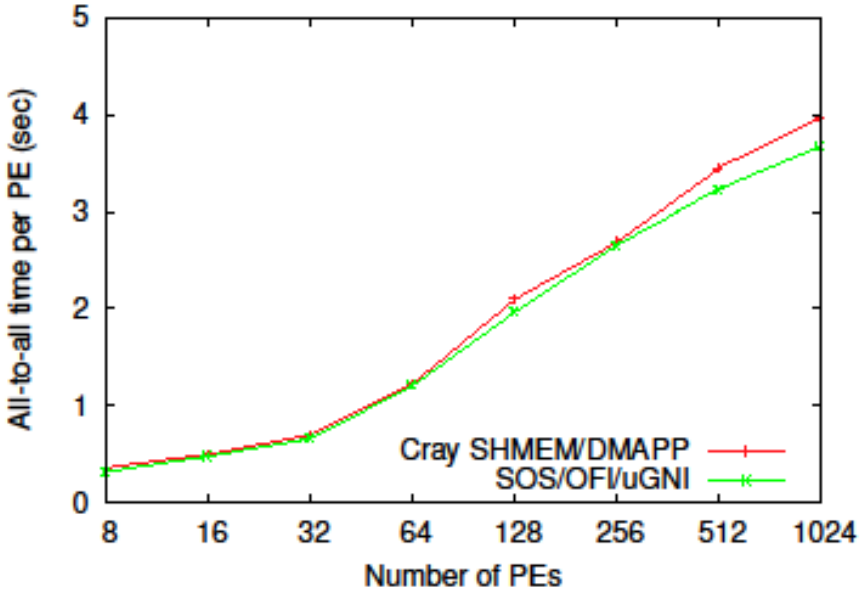


ISx: Scaling to 1024 PEs

Strong Scaling



Weak Scaling



- Strong: communication overhead causes plateau
- Weak: # messages increases relative to PE size
- Comparable results with small improvement in SOS

I. Motivation & background

II. Libfabric (OFI)

III. OFI-uGNI

IV. SHMEM-OFI-uGNI

V. Results on CORI

VI. Concluding remarks

Results Take-Away: SHMEM-OFI layer

- SHMEM-OFI and OFI "layer"
  - Low overhead
  - Improved performance

Thin SHMEM-OFI layer

- Inject
  - Improves BW and random access
- Bounce Buffer
  - increases medium message Bandwidth at a small latency cost
  - Environment variable exposed parameter

Summary

- Sandia SHMEM-OFI codebase:
  - tracking OFI and OpenSHMEM communities
- SHMEM-OFI model changes for uGNI
- Results on Aries
  - SHMEM-OFI competitive to Cray SHMEM
    - SHMEM-OFI ~ 61% Bandwidth improvement
    - Comparable latency and scaling (1024 PEs)
      - Inject improved GUPs performance
      - Latency differences from uGNI/DMAPP

Links to source

OFI-uGNI

https://github.com/ofiwg/libfabric/tree/master/prov/gni

Sandia-SHMEM

https://github.com/Sandia-OpenSHMEM/SOS/

SHMEM-OFI-uGNI: Build instructions

https://github.com/Sandia-OpenSHMEM/SOS/wiki/OFI-Build-Instructions-cray-xc

SHMEM-OFI Performance Test Suite

https://github.com/Sandia-OpenSHMEM/SOS/tree/master/test/performance/shmem_perf_suite

Legal Disclaimer & Optimization Notice

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Optimization Notice

Intel’s compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

intel®

experience
what's inside™