OPENFABRICS ALLIANCE

13th ANNUAL WORKSHOP 2017

ADVANCED PGAS CENTRIC USAGE OF THE OPENFABRICS INTERFACE

Erik Paulson, Kayla Seager, Sayantan Sur,
James Dinan, Dave Ozog: Intel Corporation
Collaborators: Howard Pritchard: Los Alamos National Laboratory
Sung-Eun Choi: Cray Inc

[March 30th, 2017]

LEGAL DISCLAIMER

OPTIMIZATION NOTICE

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

OVERVIEW

- An update on OFI implementations of OPENSHMEM and GASNET
- Performance Characteristics
- Changing landscape of PGAS languages
- Upcoming OFI features useful for PGAS
- Current and Future work

OPENFABRICS ALLIANCE

OPENSHMEM

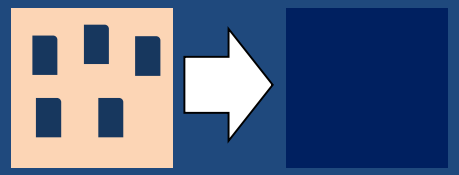
WHAT IS OPENSHMEM?

- **HPC Communication Programming Model API**
  - RMA & Atomic Pt-Pt
  - Distributed shared memory model (symmetric addressing)
  - Collectives
    - barrier, broadcast, reduce, all-to-all, strided all-to-all

OFI CAPABILITY SET FOR SHMEM *REQUIRED* FROM PROVIDER

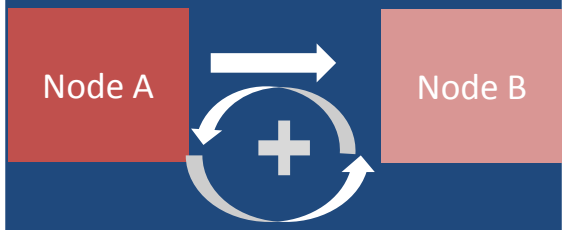
- **Prototyped, designed, implemented, and presented at OpenSHMEM Workshop 2016.**
  - https://rd.springer.com/chapter/10.1007/978-3-319-50995-2_7

Scalable, memory usage model



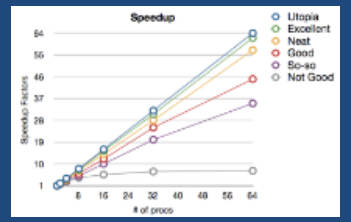
VA communication style, expose full VA range

Rich Atomic Set



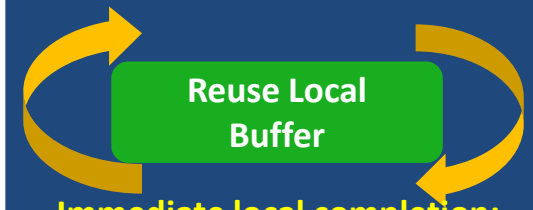
Full SHMEM converge with fine API granularity

Scalable Endpoint



Reliable connectionless endpoints with logical addressing

Accelerated small message pathway



Immediate local completion: "fi_inject"

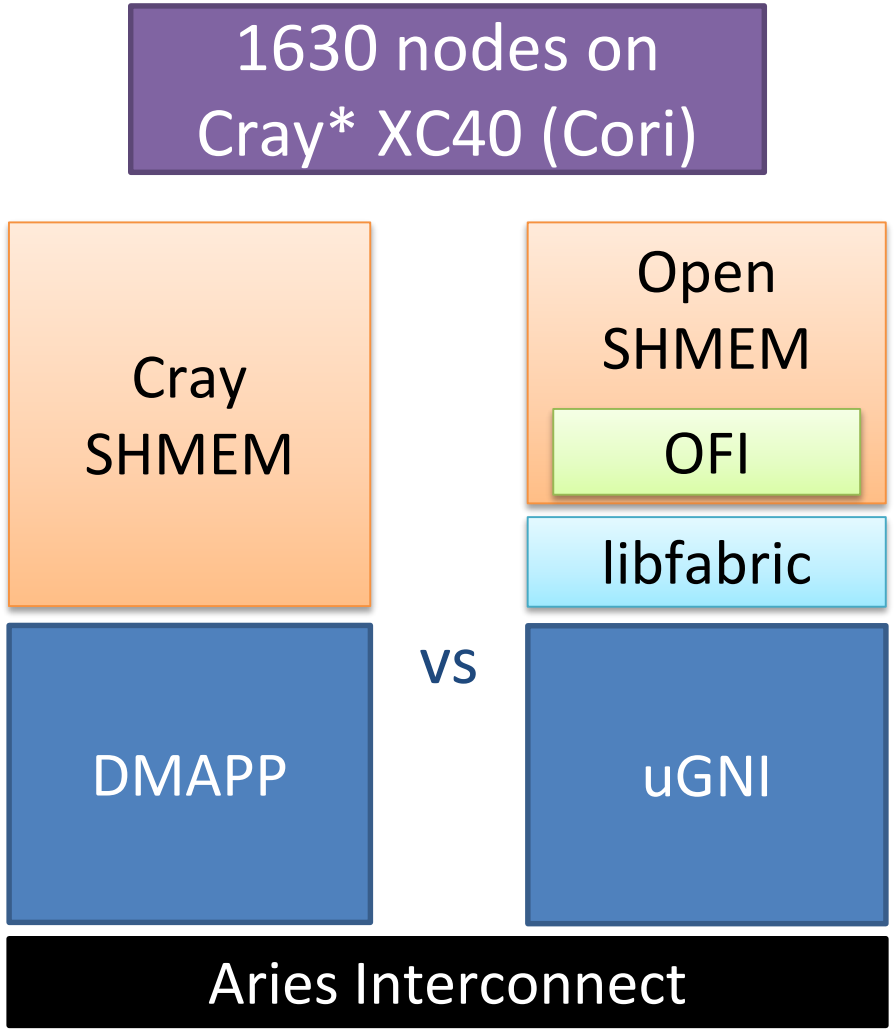
PGAS Implementation

Hardware Fabric

Efficient Remote Completion for Put/Get

counters for lightweight remote completion

SHMEM/OFI TESTING ENVIRONMENT

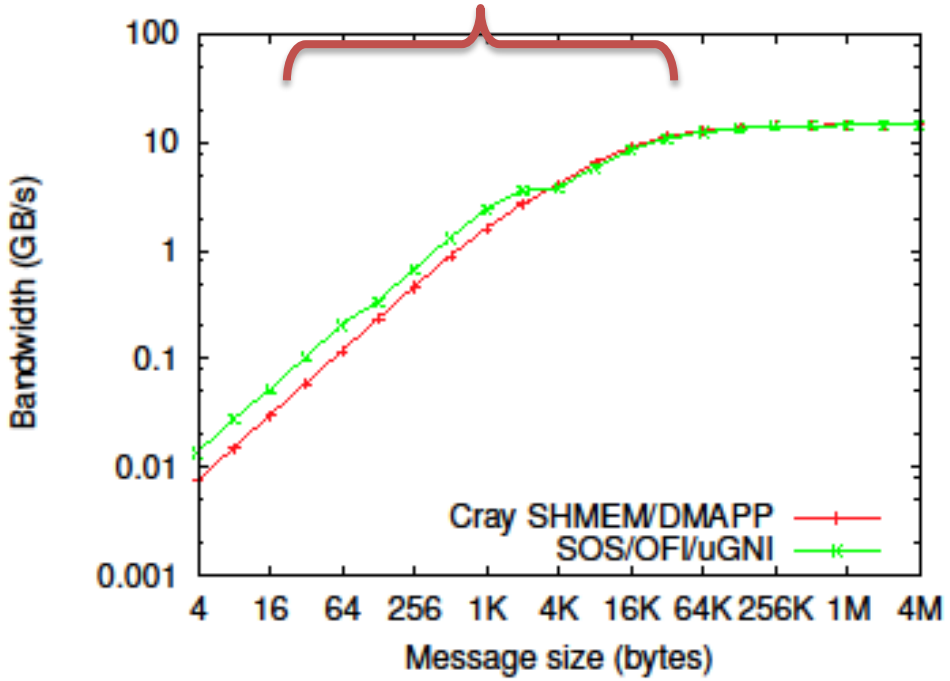


- **All tests run on CORI at NERSC**
- **Cray* SHMEM**
  - Cray* Aries, Dragonfly* topology
  - CLE (Cray* Linux*), SLURM*
  - DMAPP
    - Designed for PGAS
    - Optimized for small messages
- **Sandia* OpenSHMEM / libfabric**
  - uGNI
    - Designed for MPI and PGAS
    - Optimized for large messages

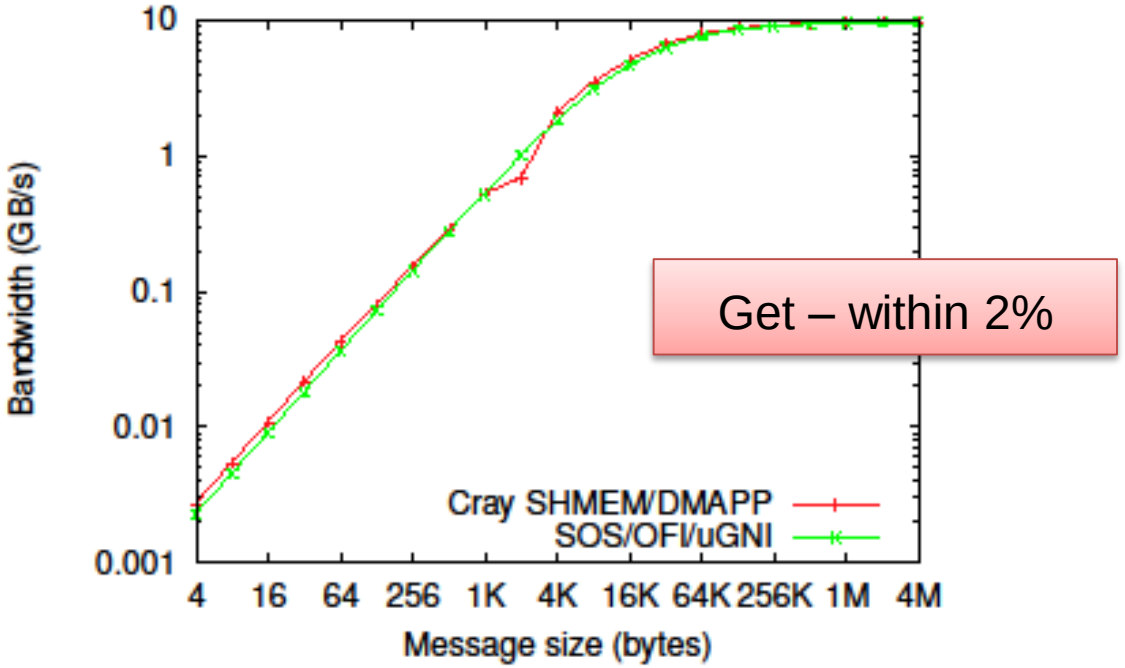
SHMEM PERFORMANCE

CRAY* XC40

Put – up to 61% improvement



Blocking Get/Put B/W



Get – within 2%

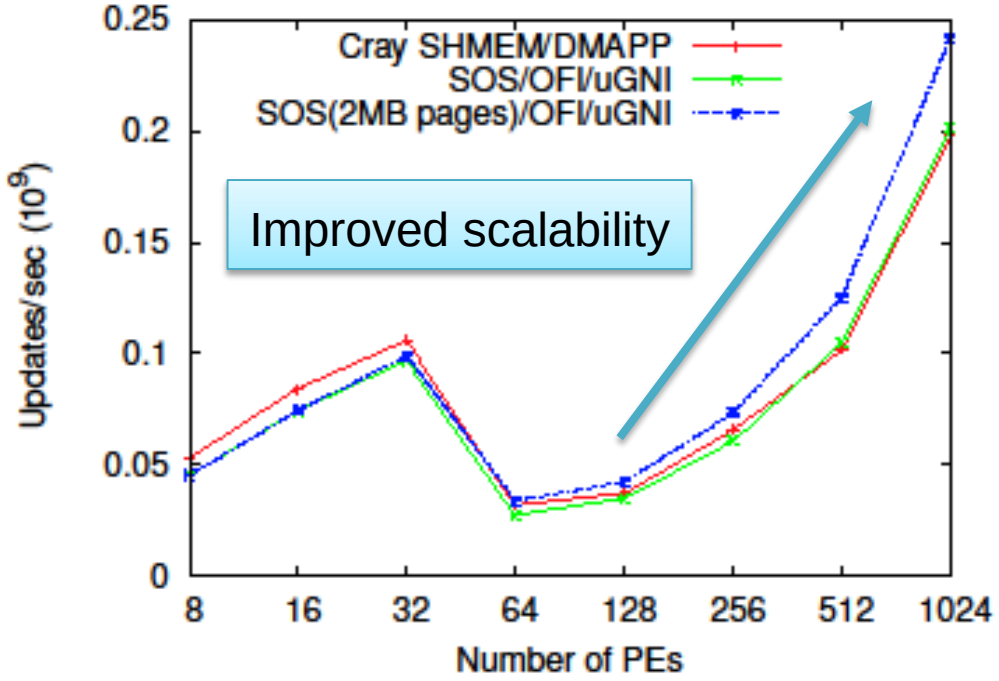
Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance. Configuration: CORI @ NERSC

* Other names and brands may be claimed as the property of others

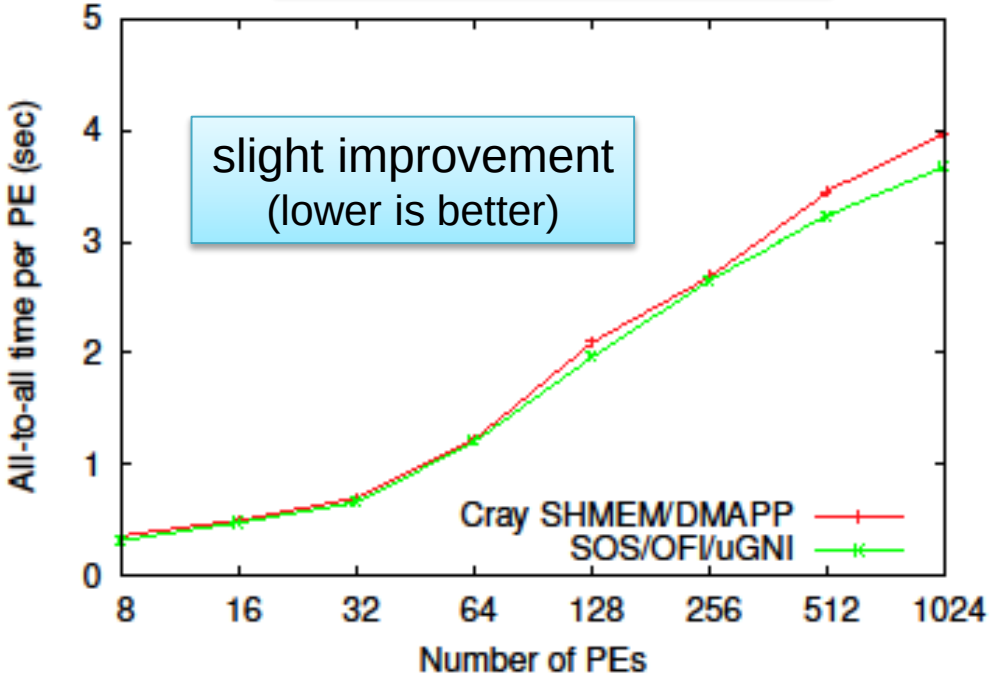
SHMEM PERFORMANCE

CRAY* XC40

GUPS Scaling



NAS ISx (Integer Sort)
weak scaling



Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance. Configuration: CORI @ NERSC

* Other names and brands may be claimed as the property of others

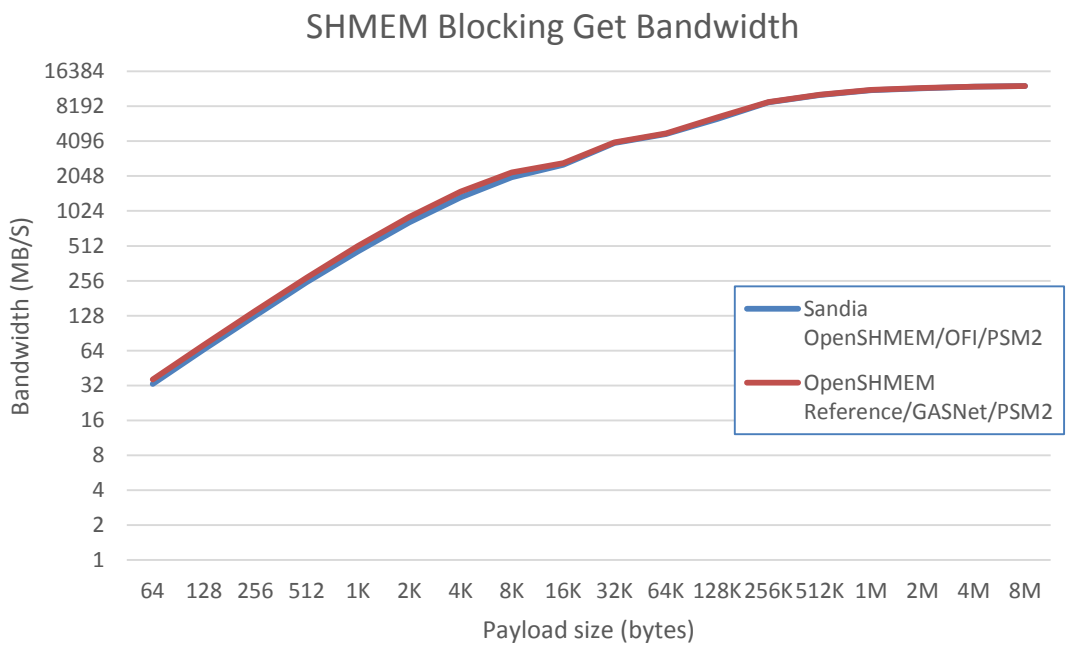
SHMEM PERFORMANCE

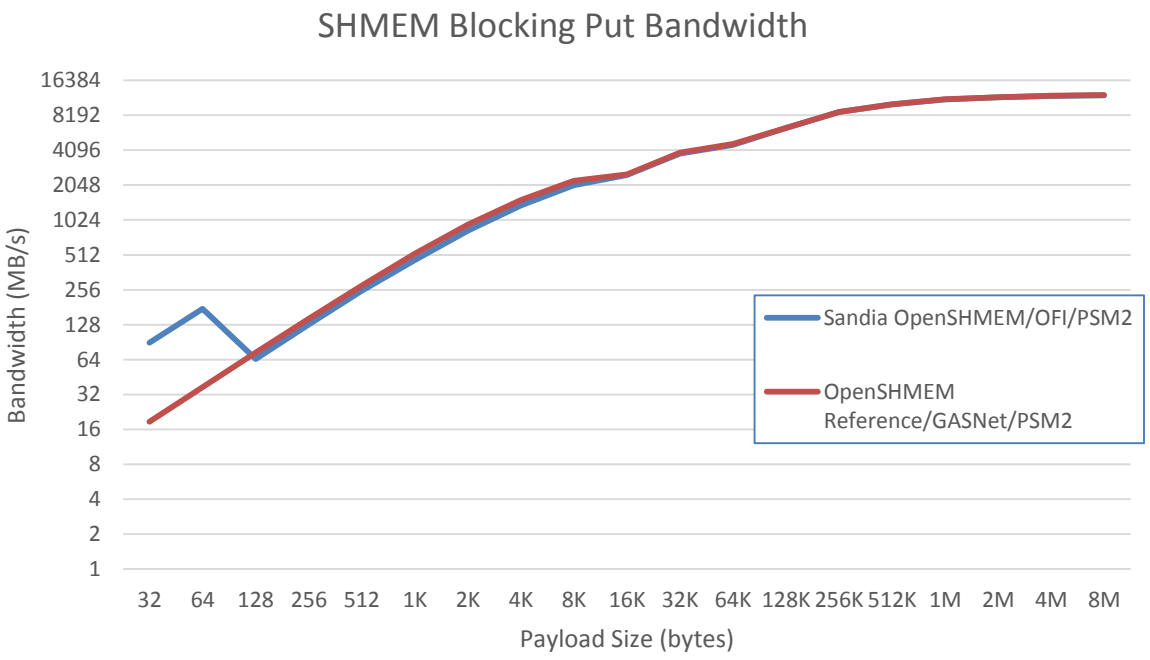
Intel® Omni-Path Architecture

- **For some additional insight, a comparison between SOS running over the PSM2 provider compared to the OpenSHMEM Reference Implementation over GASNet/PSM2**
  - Point-to-point blocking communication
  - Two nodes with Intel® Xeon™ processors
  - libfabric 1.4.1
  - psm2 10.2.63-1

SHMEM PERFORMANCE

Intel® Omni-Path Architecture

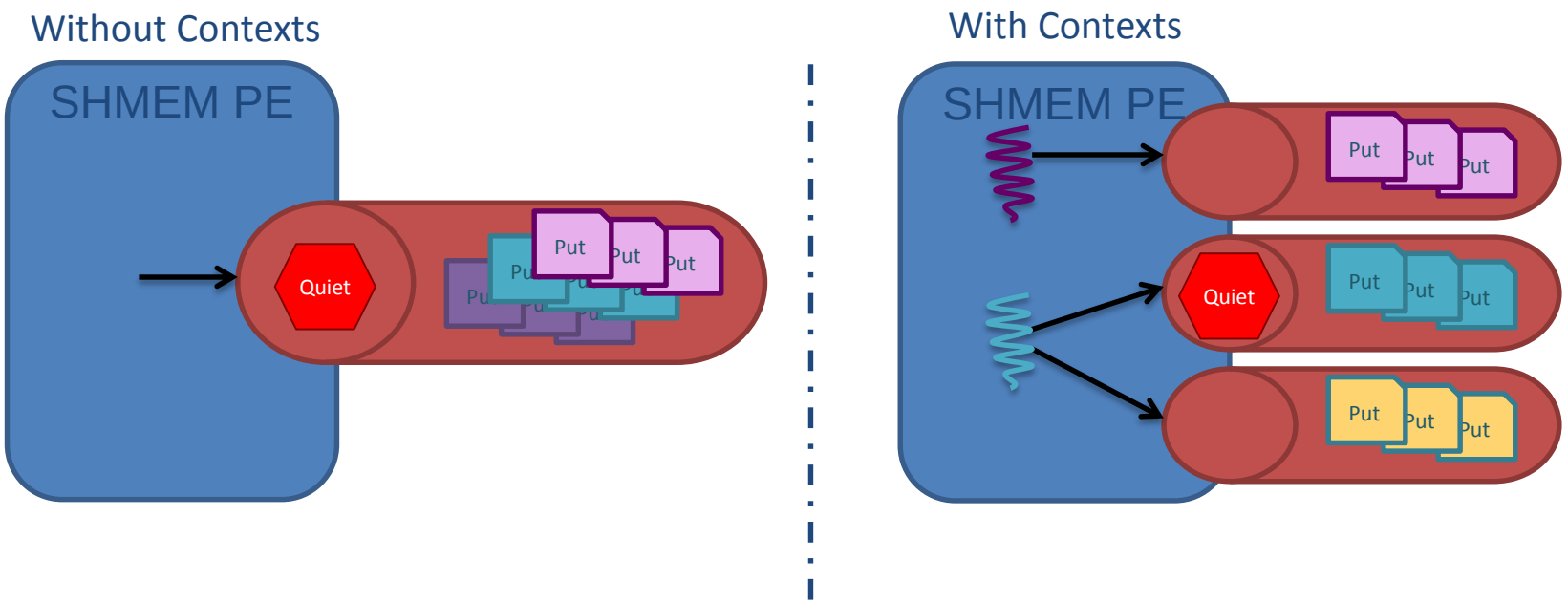




Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance. Configuration: Intel(R) Xeon(TM) CPU E5-2699 v3 @ 2.30 GHz , RHEL 7.3, libfabric 1.4.1, GASNet 1.24.2, libpsm2-10.2.63-1, OpenSHMEM Reference Implementation 1.3

FUTURE OPENSHMEM WORK: CONTEXTS

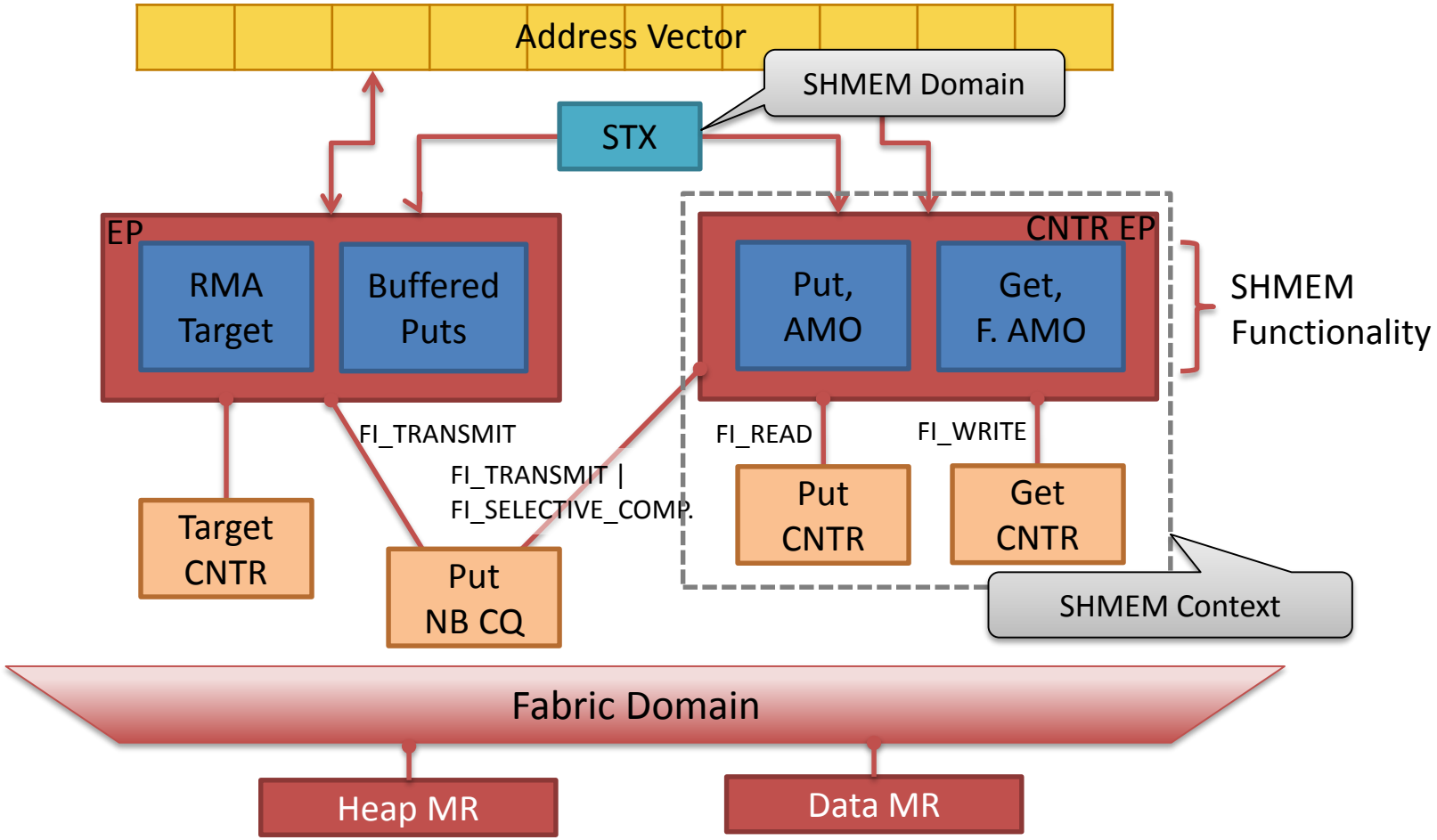
OPENSHMEM CONTEXTS: ISOLATION AND OVERLAP



- **Proposed SHMEM extension to enable threading support as well as communication overlap**
- **Adds context argument to communication routines**
- **Contexts define which operations are included in quiet (completion) and fence (ordering)**
- **Cleanly and conveniently maps to OFI features (shared transmit contexts)**

SANDIA* OPENSHMEM: CURRENT OFI TRANSPORT ARCHITECTURE

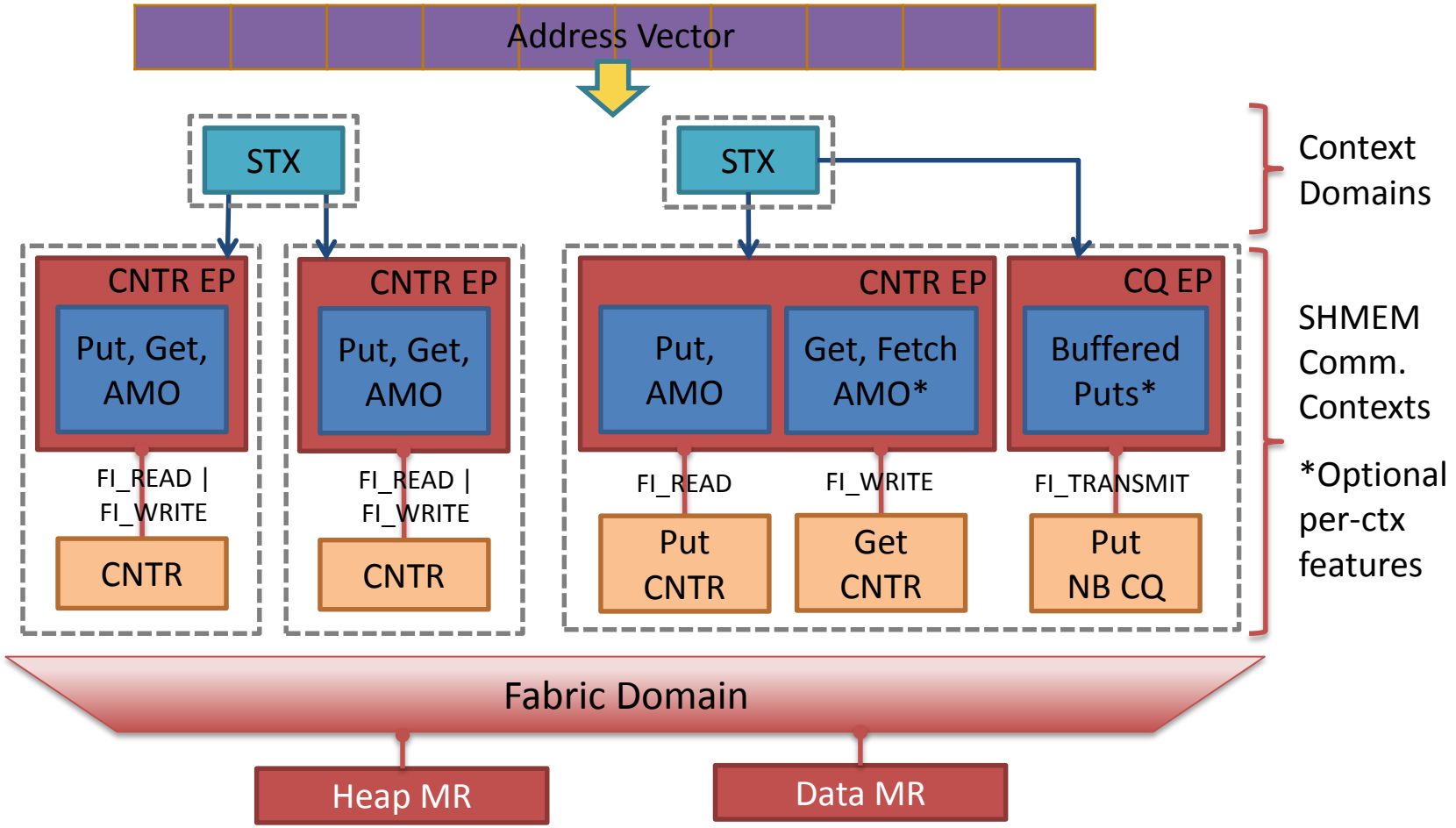
- **Current implementation tries to optimize for situations that the user would if they had contexts.**
  - e.g. Blocking gets don't wait for puts to complete
- **Ideally each thread would have its own STX.**



* Other names and brands may be claimed as the property of others

SANDIA* OPENSHMEM: MULTITHREADED OFI TRANSPORT ARCHITECTURE

- **If users want to separate completion of puts and gets, they can issue them on separate contexts.**
- **Exposed access to shared transmit contexts through OFI is crucial for this model**
  - No other networking environments provide this.

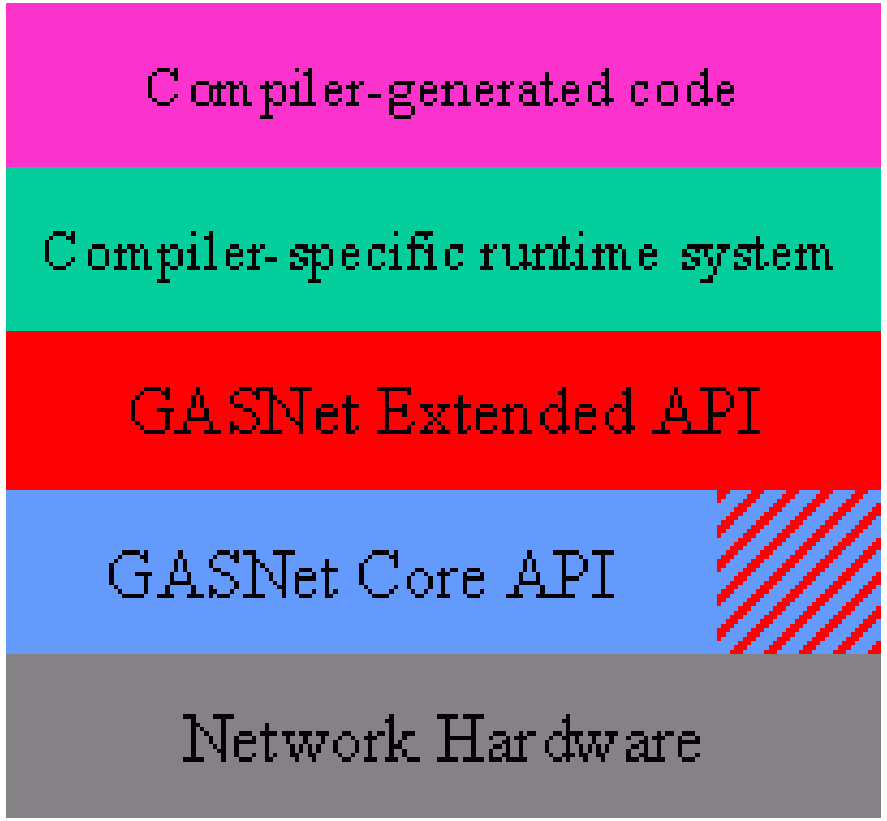


* Other names and brands may be claimed as the property of others

GASNET

WHAT IS GASNET?

- **What is GASNet?**
  - http://gasnet.lbl.gov
- **Low-level networking API meant to enable PGAS languages.**
  - NOT for end-users, but for people like us.
  - Developed by Lawrence Berkeley National Laboratory
- **Projects using GASNet: Berkeley UPC, Chapel, Legion, UPC++, Co-Array Fortran, OpenSHMEM**
- **Layered Approach**
- **Core API is required to be implemented**
  - Reference implementation of extended API in terms of the core API is provided.
- **Native support for most relevant networks**



STATE OF GASNET SUPPORT

- **Currently GASNet/OFI supports Intel® True Scale Architecture, Intel® Omni-Path Architecture, and TCP/IP.**
  - Experimental support for Cray* XC systems via GNI provider.
  - Blue Gene/Q provider supports the implementations requirements, but has not been tested yet.
- **Provider Requirements to support GASNet**
  - FI_EP_RDM
  - (Preferred) FI_MR_SCALABLE, FI_MR_BASIC
  - FI_MSG, FI_RMA
  - FI_MULTI_RECV

Will change to new mode bits in OFI 1.5

- **Support on platforms like verbs may be easily achieved through utility providers**

* Other names and brands may be claimed as the property of others

WORK DONE ON GASNET/OFI

- **Provider specific optimizations and detection**
- **FI_MR_BASIC support (enable gni provider)**
- **Threading improvements**
  - Moved from global lock to FI_THREAD_SAFE
- **Bounce buffering for non-blocking, non-bulk puts**
  - In this case, GASNet has stricter data-reuse requirements than OFI guarantees
  - Multi-faceted approach using FI_INJECT, bounce buffers, and simple blocking increases performance
- **Bug fixes and refactoring**
  - Improvements to operation progress
  - Receive buffer reference counting

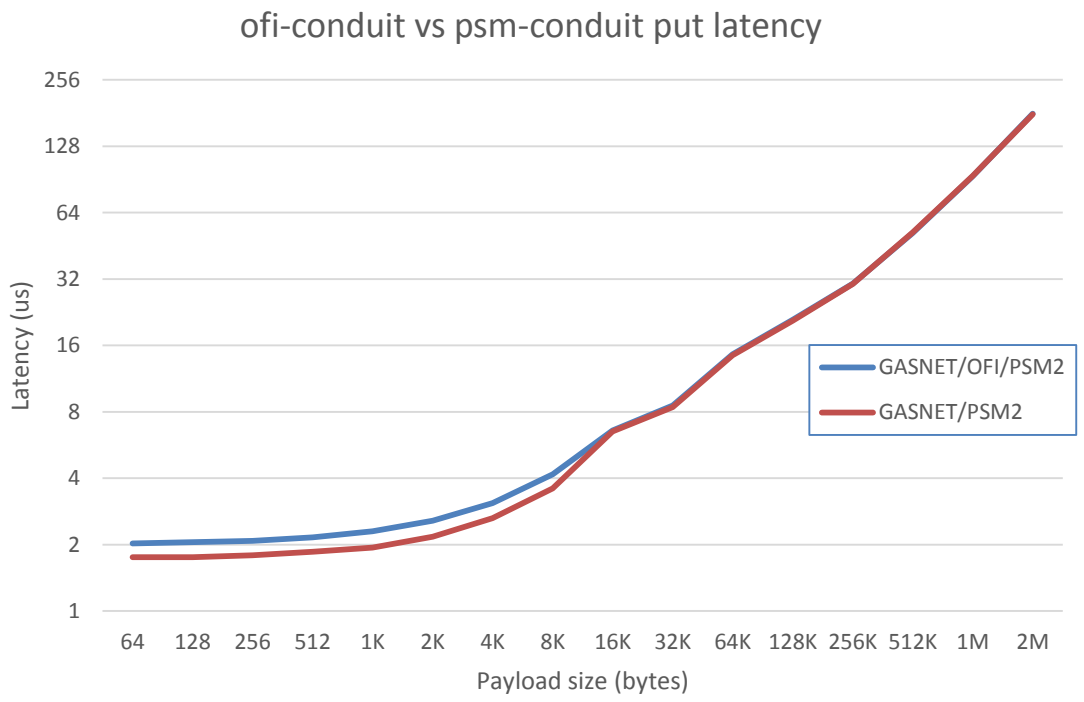
GASNET PERFORMANCE COMPARISIONS

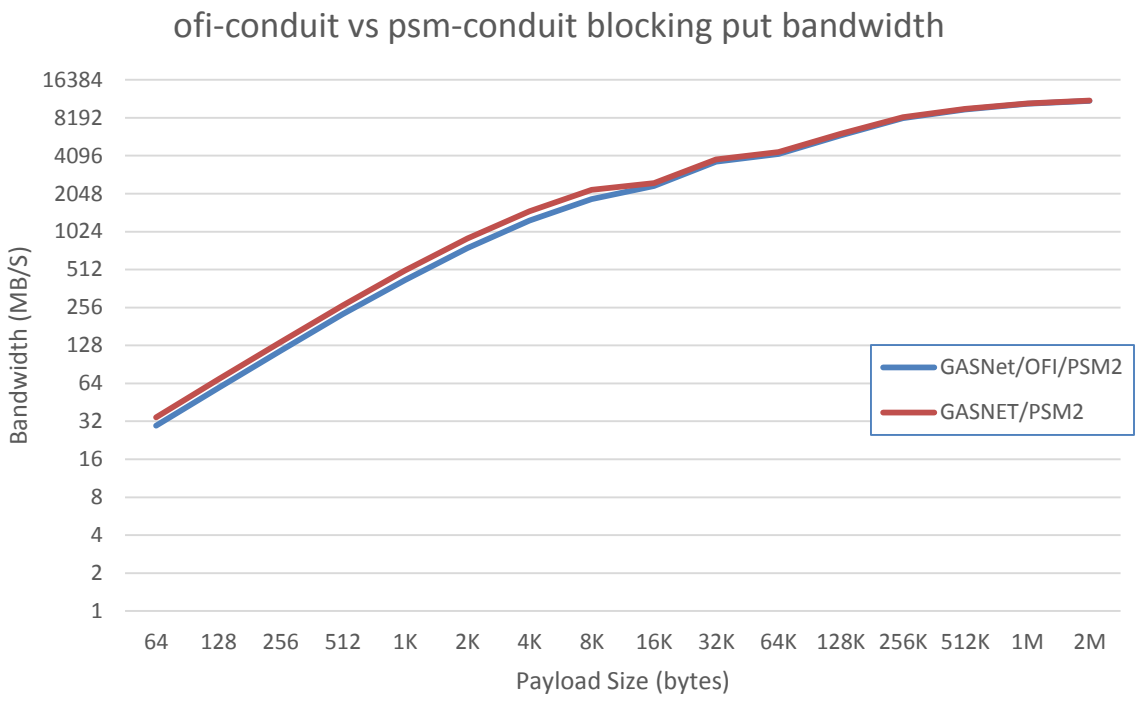
OFI vs Native PSM2

- **Point-to-point communication**
- **Two nodes with Intel® Xeon™ processors**
- **libfabric 1.4.0**
- **psm2 10.2.58-1**
- **GASNet testsmall (latency) and testlarge (bandwidth)**
  - Both blocking results

GASNET PERFORMANCE COMPARISONS

GASNet/OFI/PSM2 vs GASNET/PSM2





Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance. Configuration: Intel(R) Xeon(TM) CPU E5-2697 v3 @ 2.60GHz, RHEL 7.3, libfabric 1.4.0, GASNet 1.28.2, libpsm2-10.2.58-1

PERFORMANCE DISCUSSION

- **At large message sizes, performance is more or less the same**
- **For small message sizes there is a disparity**
- **Reason: native psm-conduit is using different completion mechanism**

```
psm2_error_t
psm2_am_request_short(psm2_epaddr_t epaddr, psm2_handler_t handler,
        psm2_amarg_t *args, int nargs, void *src,
        size_t len, int flags,
        psm2_am_completion_fn_t completion_fn,
        void *completion_ctxt);
```

- **Callback function executed when remote completion is finished**
  - Pros: Better latency, reduces overhead related to completion queue processing
  - Cons: Does not return error/success information
- **Native OPA provider could map better, as opposed to through PSM2**

CHANGING LANDSCAPE OF PGAS

GASNet-EX

- **LBNL is working on the next generation of GASNet**
  - Working towards exascale
- **Vectored-Indexed-Strided Operations**
  - Maps well to using a scatter-gather list to reduce number of calls into OFI.
  - May be useful to use a completion counter instead of CQ
- **Collectives/Dependent Operations**
  - Upcoming FI_TRIGGER improvements will lend to a more natural implementation
  - Deferred work queue concept
- **Multi-endpoint support**
  - OFI's connectionless, reliable endpoints are a natural fit
- **Multi-segment support**
  - Flexible memory registration semantics are a tight semantic match

FUTURE WORK FOR GASNET/OFI

- **Investigate scalable endpoints for GASNet and scalable communication in general**
  - Currently two endpoint addresses are registered for every node in the job, on every node.
  - Scalable endpoints could cut that in half
  - More scalable communication should be considered looking towards exascale.
  - FI_SHARED_AV in OFI-1.5 can further reduce per node memory usage.
- **Fine tune performance**
- **Support GASNet-Ex**
- **Improve active-messaging path**
- **Support more OFI providers**
  - Fully support gni provider and move out of experimental support

OPENFABRICS ALLIANCE

13th ANNUAL WORKSHOP 2017

THANK YOU

Erik Paulson

Intel Corporation

Special thanks to:

Paul Hargrove and Dan Bonachea, Lawrence Berkeley National Laboratory

OPENFABRICS ALLIANCE

QUESTIONS?