MVAPICH

MPI, PGAS and Hybrid MPI+PGAS Library

NETWORK-BASED COMPUTING LABORATORY

Exploiting and Evaluating OpenSHMEM on KNL Architecture

Jahanzeb M. Hashmi, Mingzhe Li, Hari Subramoni, and Dhabaleswar K. (DK) Panda

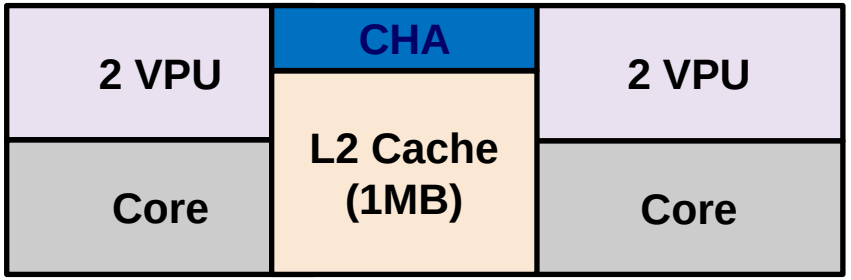
Network-Based Computing Laboratory
Department of Computer Science and Engineering
The Ohio State University, Columbus, OH, USA

Outline

- **Introduction**
- Motivation
- Contributions
- Evaluation Methodology
- Results and Discussion
- Conclusion

Intel Knights Landing Processor – Overview

- Multi-threaded cores
  - Up to 72 cores (model 7290)
- All cores divided into 36 Tiles
- Each tile contains two core
  - 2 VPU per core
  - 1MB shared L2 cache
- 512-bit wide vector registers
  - AVX-512 extensions

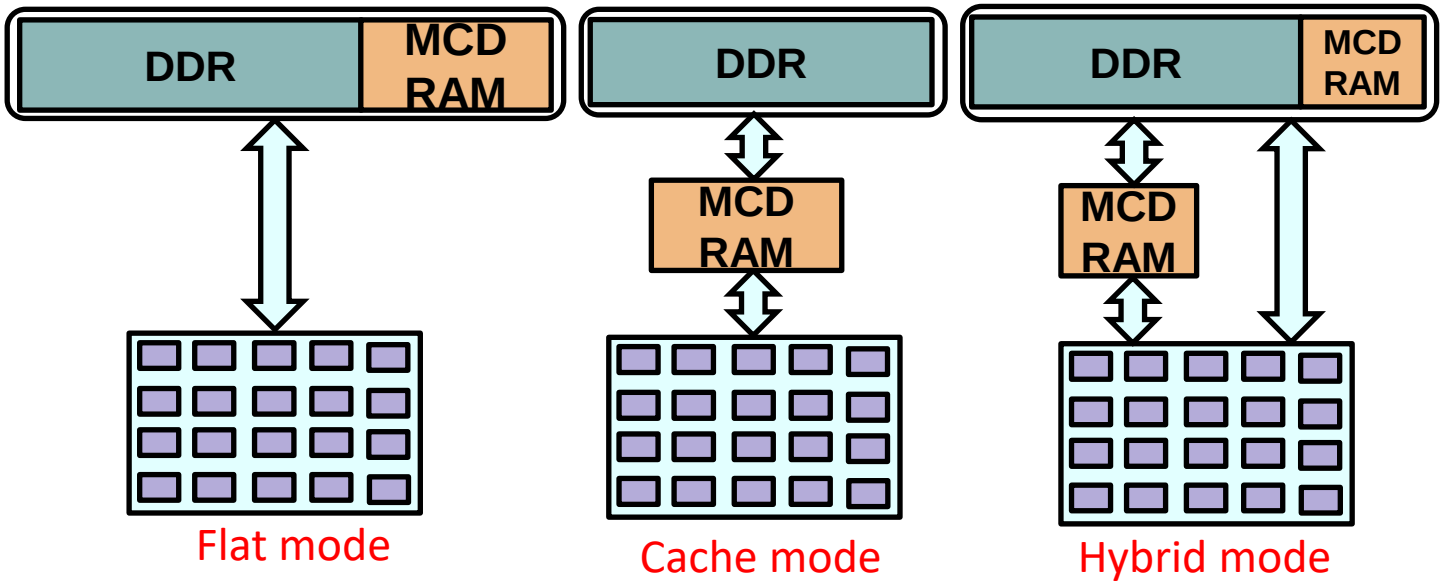


A single Tile of KNL

Intel Knights Landing Processor – Overview

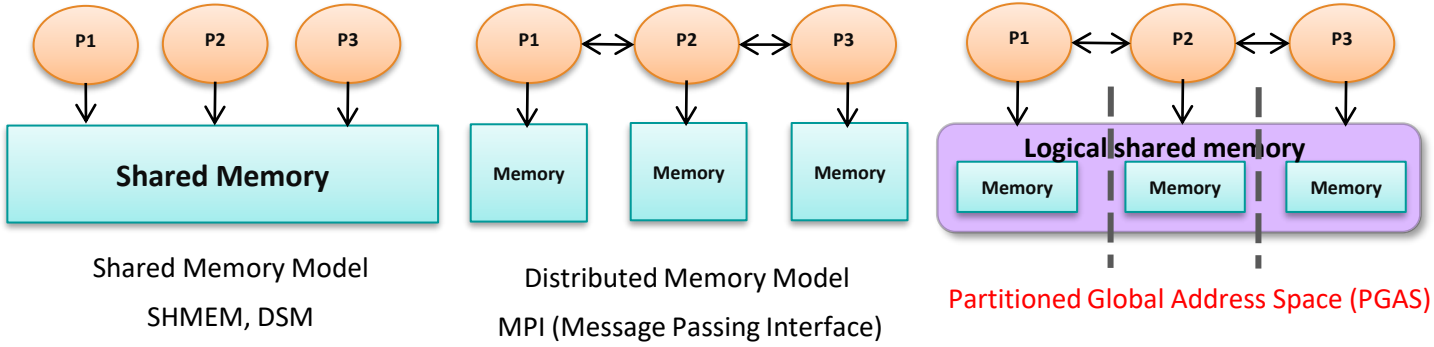
- 4 *threads* per core with Simultaneous Multi-threading
  - Shared and dynamically partitioned resources
  - Out of order execution
- Configurable mesh interconnect
  - *AlltoAll*: addresses are uniformly distributed across tag directories
  - *Quadrant*: memory appears as single NUMA domain
  - *SNC*: memory appears as distinct NUMA domains

Intel Knights Landing Processor – MCDRAM



- On-package Multi Channel DRAM (MCDRAM)
  - 450 GB/s of theoretical bandwidth (4x of DDR)
  - Configurable in Flat, Cache, and Hybrid modes

Partitioned Global Address Space (PGAS) Models



- Key abstraction
  - Shared memory abstraction over distributed system images
- Library-level solutions
  - OpenSHMEM
  - Global Arrays
  - UPC++
- Language-level solutions
  - UPC
  - Coarray Fortran (CAF)
  - …

Outline

- Introduction
- **Motivation**
- Contributions
- Evaluation Methodology
- Results and Discussion
- Conclusion

Motivation

- Optimizing HPC runtimes and applications on emerging manycores is of great research interest
- Exploring benefits of the architectural features of KNL for OpenSHMEM model and application
  - Impact of vectorization on application kernels
  - MCDRAM vs. DDR performance
  - Exploiting hardware multi-threading

Problem Statements

Can PGAS models, specifically OpenSHMEM, benefit from the architectural features of KNL processor at micro-benchmarks as well as application level?

Can we identify the optimizations that could help achieve better performance on KNL?

Outline

- Introduction
- Motivation
- **Contributions**
- Evaluation Methodology
- Results and Discussion
- Conclusion

Contributions

- Evaluate OpenSHMEM point-to-point, collective, and atomic operations on KNL architecture
- Evaluate OpenSHMEM application kernels
  - Discuss the impact of MCDRAM and vectorization
  - Comparison of KNL against Broadwell on core-by-core and node-by-node performance
- Discuss potential optimization for KNL architecture

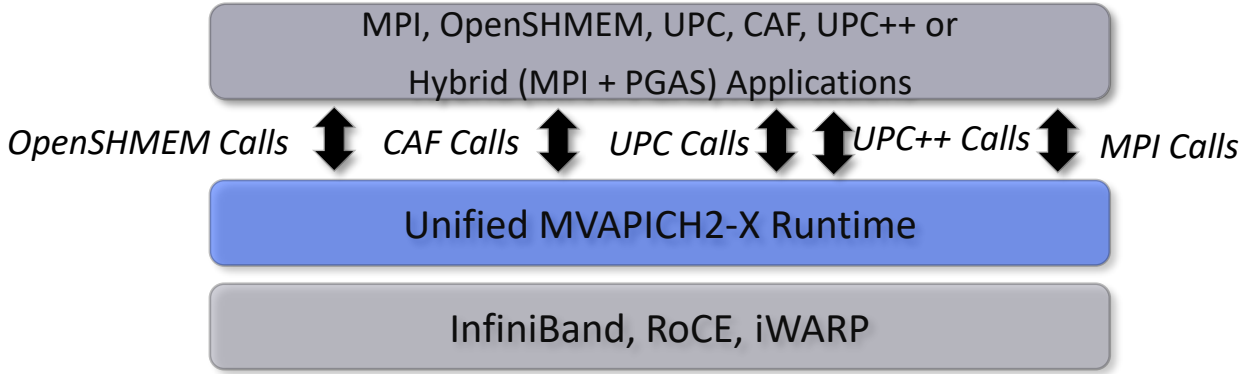
Outline

- Introduction
- Motivation
- Contributions
- **Evaluation Methodology**
- Results and Discussion
- Conclusion

Overview of the MVAPICH2 Project

- High Performance open-source MPI Library for InfiniBand, Omni-Path, Ethernet/iWARP, and RDMA over Converged Ethernet (RoCE)
  - MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.0), Started in 2001, First version available in 2002
  - MVAPICH2-X (MPI + PGAS), Available since 2011
  - Support for GPGPUs (MVAPICH2-GDR) and MIC (MVAPICH2-MIC), Available since 2014
  - Support for Virtualization (MVAPICH2-Virt), Available since 2015
  - Support for Energy-Awareness (MVAPICH2-EA), Available since 2015
  - Support for InfiniBand Network Analysis and Monitoring (OSU INAM) since 2015
  - **Used by more than 2,750 organizations in 84 countries**
  - **More than 416,000 (> 0.4 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (Nov '16 ranking)
    - **1st, 10,649,600-core (Sunway TaihuLight) at National Supercomputing Center in Wuxi, China**
    - 13th, 241,108-core (Pleiades) at NASA
    - 17th, 462,462-core (Stampede) at TACC
    - 40th, 74,520-core (Tsubame 2.5) at Tokyo Institute of Technology
  - Available with software stacks of many vendors and Linux Distros (RedHat and SuSE)
  - **http://mvapich.cse.ohio-state.edu**
- Empowering Top500 systems for over a decade
  - System-X from Virginia Tech (3rd in Nov 2003, 2,200 processors, 12.25 TFlops) ->
  - Sunway TaihuLight (1st in Jun'16, 10M cores, 100 PFlops)

MVAPICH2-X for Advanced MPI and Hybrid MPI + PGAS Applications



- Unified communication runtime for MPI, UPC, OpenSHMEM, CAF available with MVAPICH2-X 1.9 (2012) onwards!
- UPC++ support available since MVAPICH2-X 2.2RC1
- Feature Highlights
  - Supports MPI(+OpenMP), OpenSHMEM, UPC, CAF, UPC++, MPI(+OpenMP) + OpenSHMEM, MPI(+OpenMP) + UPC + CAF
  - MPI-3 compliant, OpenSHMEM v1.0 standard compliant, UPC v1.2 standard compliant (with initial support for UPC 1.3), CAF 2008 standard (OpenUH), UPC++ 1.0
  - Scalable Inter-node and intra-node communication – point-to-point and collectives

Benchmarks and Configurations

- OpenSHMEM Microbenchmark Evaluation
  - Point-to-point (Put/Get), collectives (broadcast, reduce), and atomics
  - OSU Microbenchmark v5.3.2
- OpenSHMEM NAS Parallel Benchmark Kernels
  - MG, BT, EP, SP
- University of Houston OpenSHMEM test suite
  - Five application Kernels
  - 2D-Heat, Heat-Image, Matrix Multiplication, DAXPY, and ISx
- All experiments evaluate KNL 7250 against Broadwell
- Application evaluation on KNL also discuss MCDRAM and AVX-512 benefits

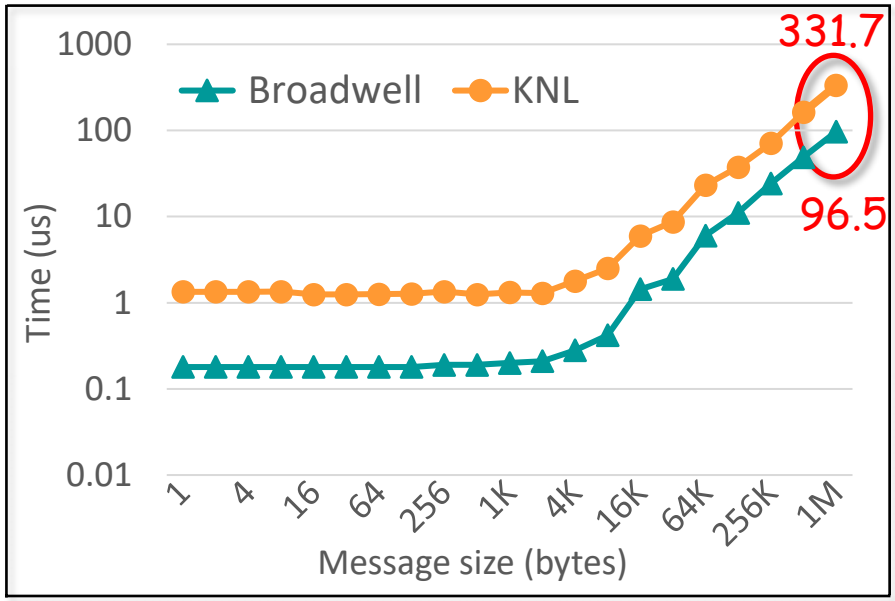
Experiment Setup

- Intel Xeon Phi KNL Cluster @ CSE, OSU (4 nodes)
  - KNL 7250 (1.4GHz) with 16GB MCDRAM and 96GB DDR
  - Mellanox EDR Connect-X HCAs (100 Gbps data rate)
- RI2 Cluster @ CSE, OSU (40 nodes)
  - Xeon E5-2680 v4 (2.40 GHz) with 128 GB DDR
  - Mellanox EDR Connect-X HCAs (100 Gbps data rate)
- Software stack
  - RHEL 6.3 with Mellanox OFED v2.2-1.0.0
  - MVAPICH2-X 2.2 with GCC v5.4.0

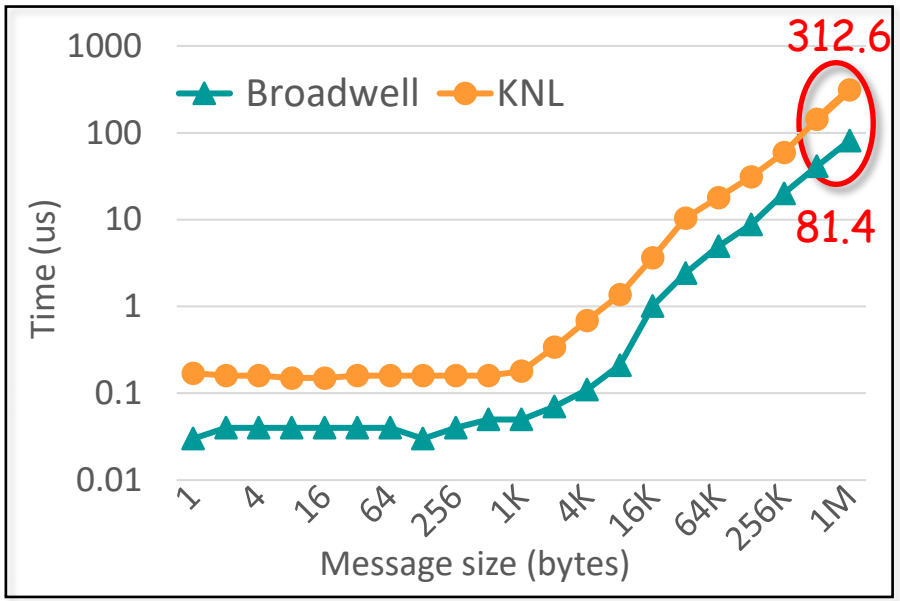
Outline

- Introduction
- Motivation
- Contributions
- Evaluation Methodology
- **Results and Discussion**
- Conclusion

Microbenchmark Evaluations (Intra-node Put/Get)



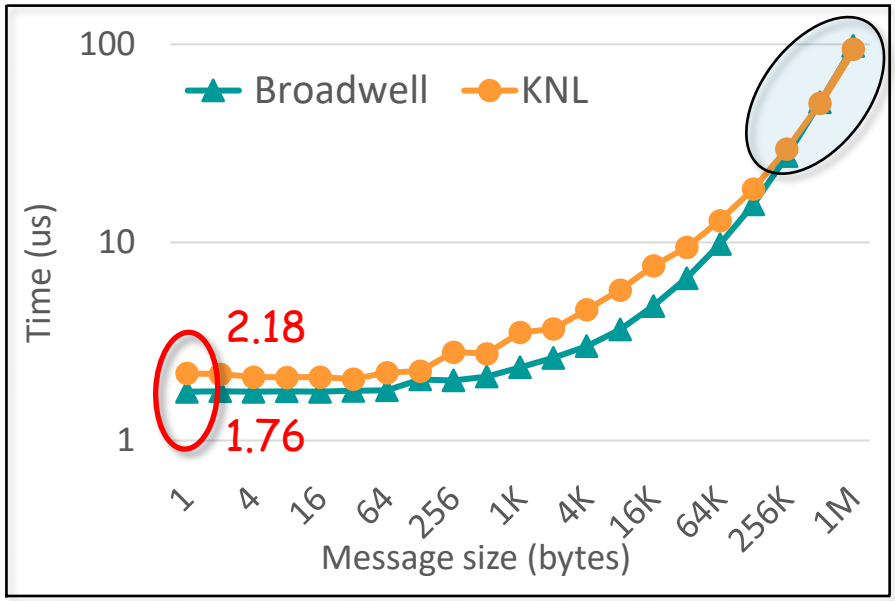
Shmem_putmem



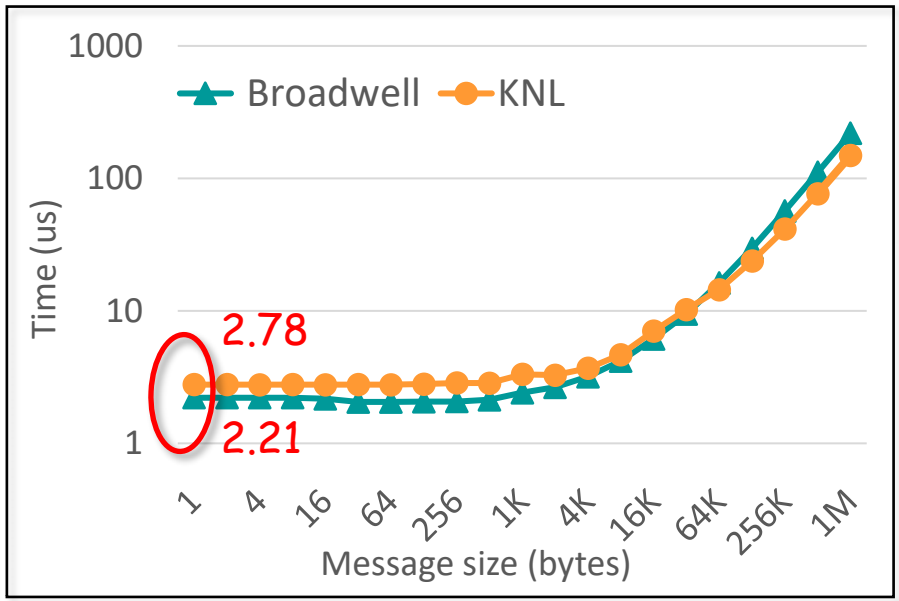
Shmem_getmem

- Broadwell shows about 3X better performance than KNL on large message
- Muti-threaded memcpy routines on KNL could offset the degradation caused by the slower core on basic Put/Get operations

Microbenchmark Evaluations (Inter-node Put/Get)



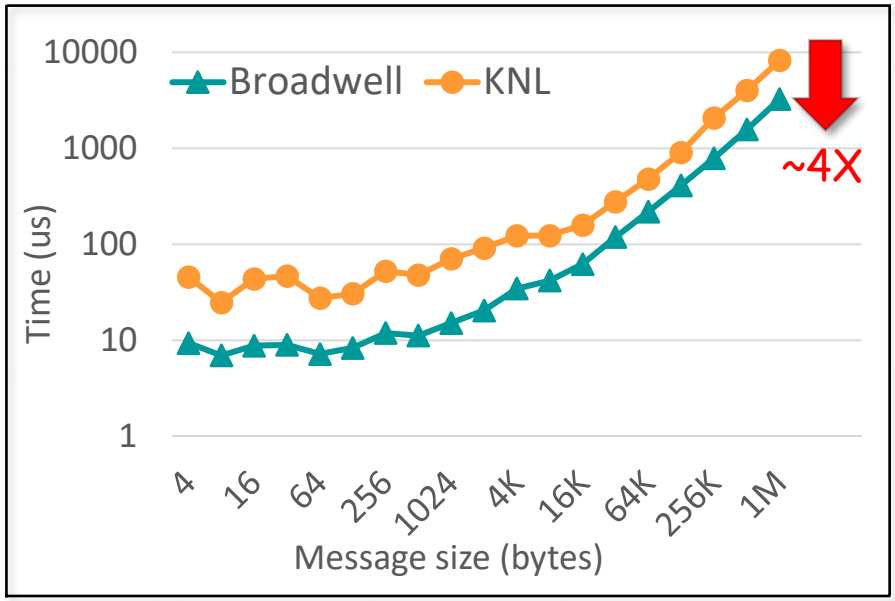
Shmem_putmem



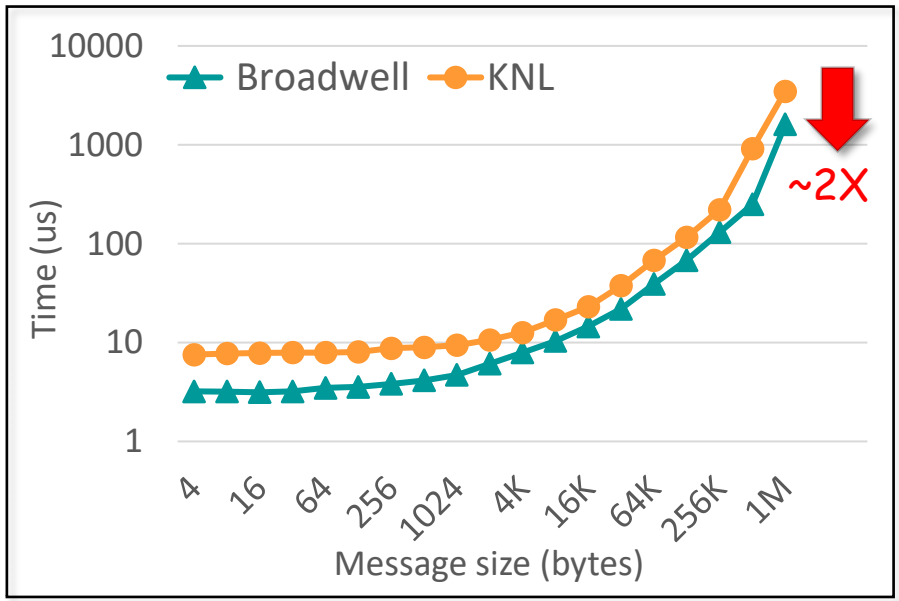
Shmem_getmem

- Inter-node small message latency is only 2X worse on KNL. While large message performance is almost similar on both KNL and Broadwell.

Microbenchmark Evaluations (Collectives)



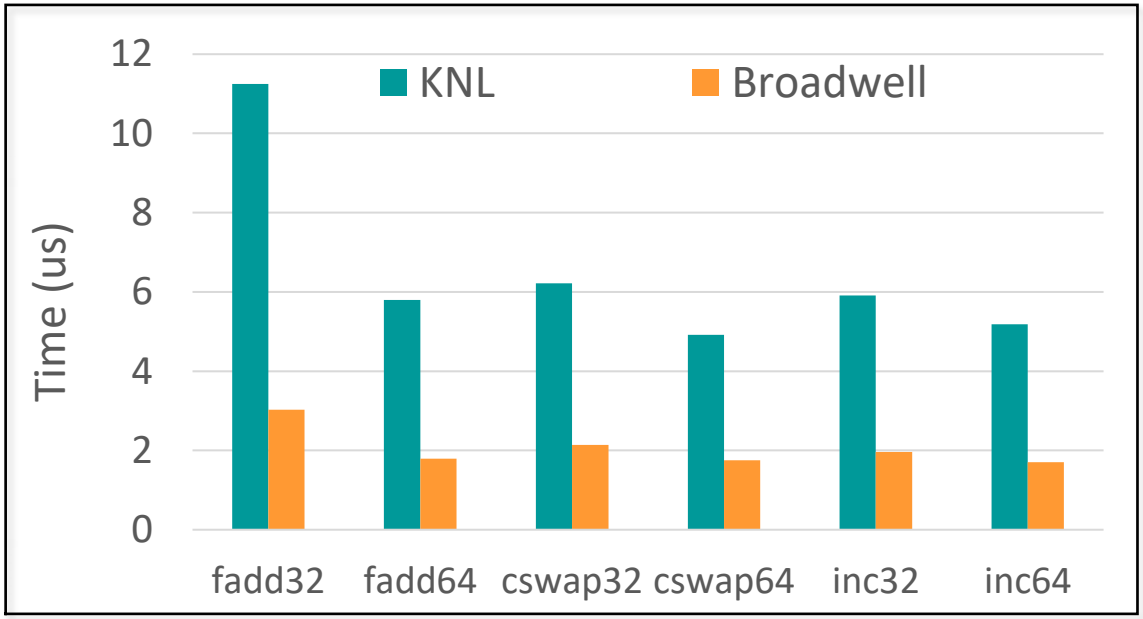
int_sum_to_all on 128 PEs



broadcast on 128 PEs

- 2 KNL nodes (64 ppn) and 8 Broadwell nodes (16 ppn).
- 4X degradation is observed on KNL using collective benchmarks.
- Basic point-to-point performance difference is reflected in collectives as well

Microbenchmark Evaluations (Atomics)

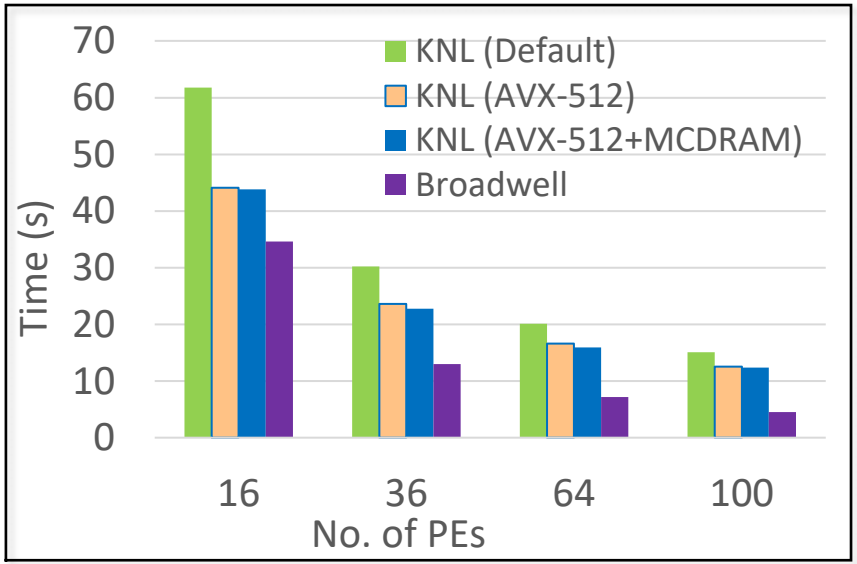


OpenSHMEM atomics on 128 PEs

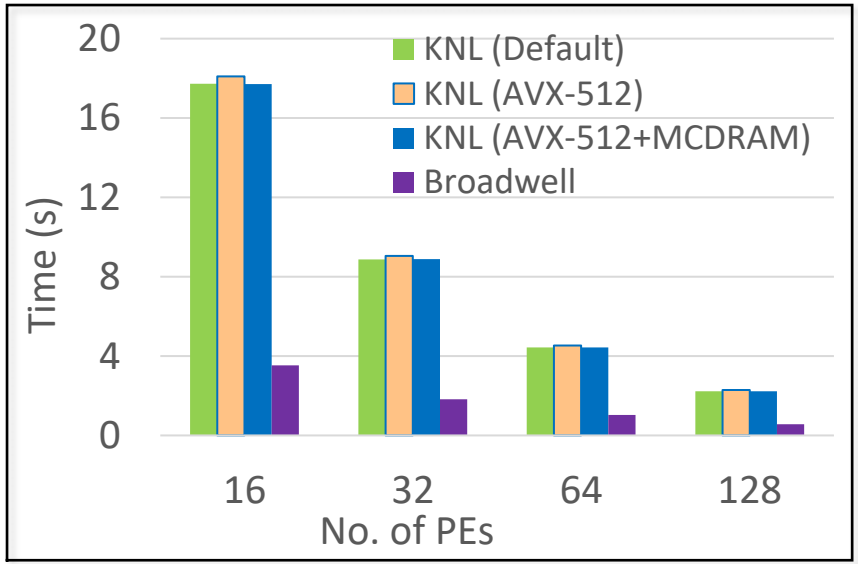
- Using multiple nodes of KNL, atomic operations showed about 2.5X degradation on compare-swap, and Inc atomics
- Fetch-and-add (32-bit) showed up to 4X degradation on KNL

NAS Parallel Benchmark Evaluation

NAS-BT (PDE solver), CLASS=B



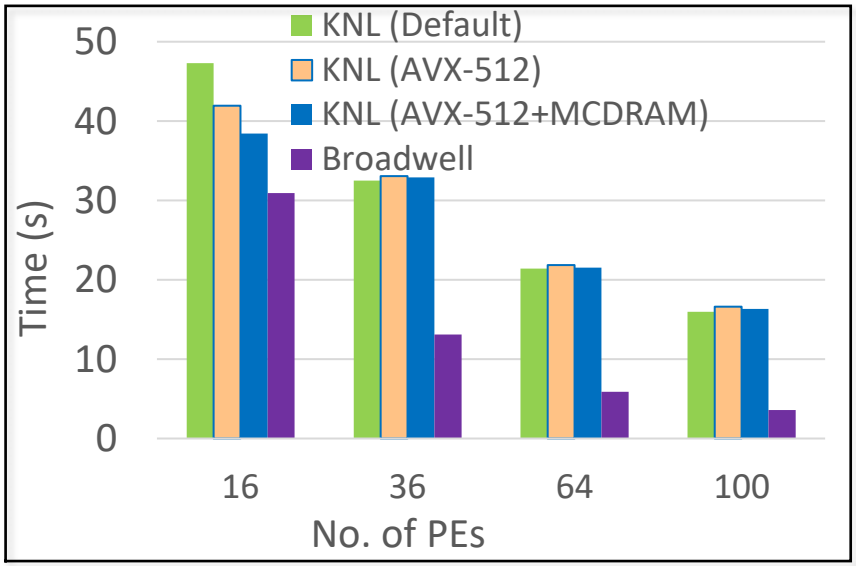
NAS-EP (RNG), CLASS=B



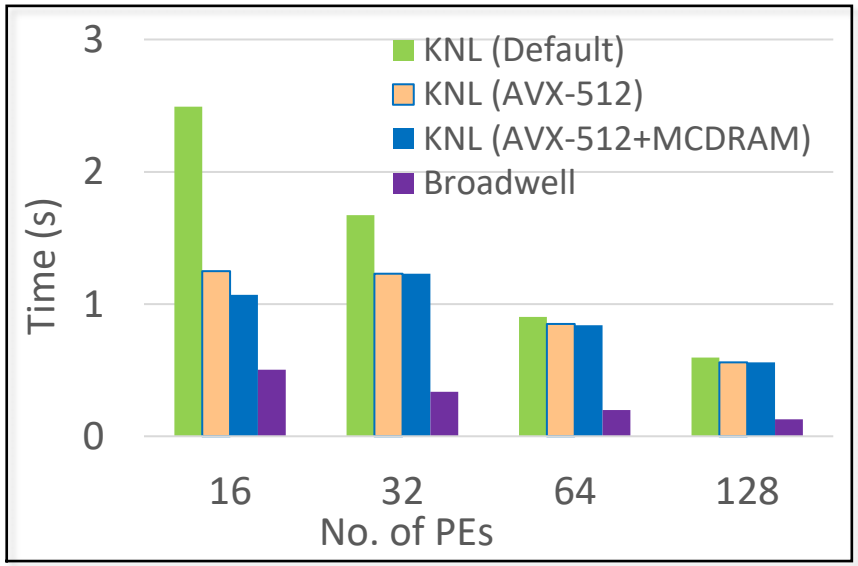
- AVX-512 vectorized execution of BT kernel on KNL showed 30% improvement over default execution while EP kernel didn't show any improvement
- Broadwell showed 20% improvement over optimized KNL on BT and 4X improvement over all KNL executions on EP kernel (random number generation).

NAS Parallel Benchmark Evaluation (contd.)

NAS-SP (non-linear PDE), CLASS=B



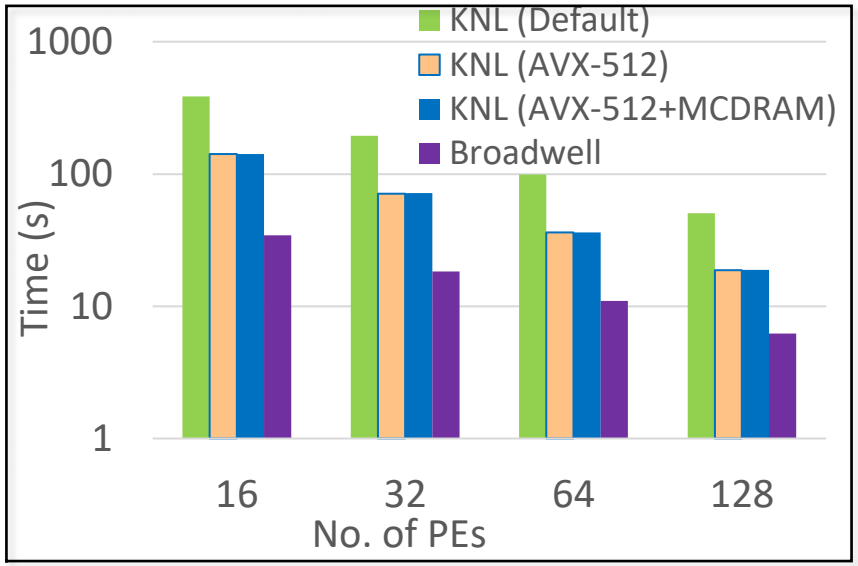
NAS-MG (MultiGrid solver), CLASS=B



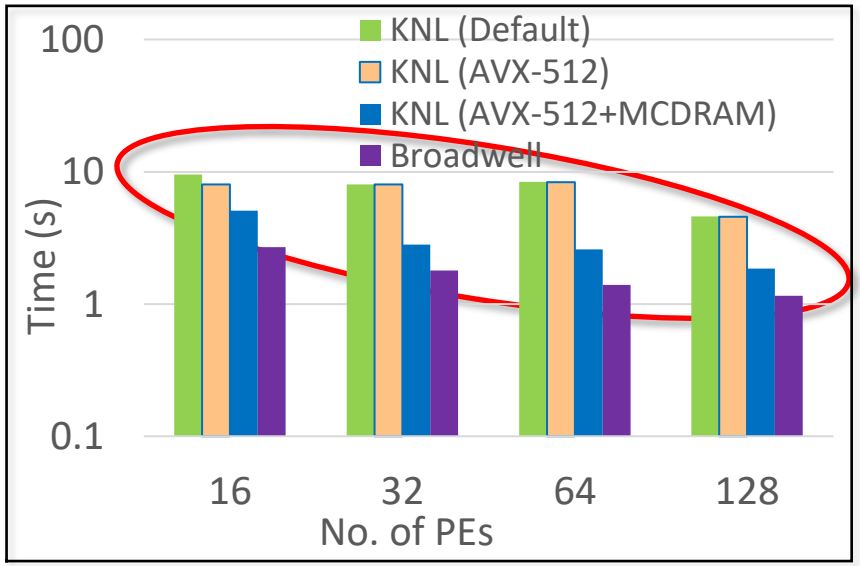
- Similar performance trends are observed on BT and MG kernels as well
- On SP kernel, MCDRAM based execution showed up to 20% improvement over default at 16 processes.

Application Kernels Evaluation

Heat-2D Kernel using Jacobi method



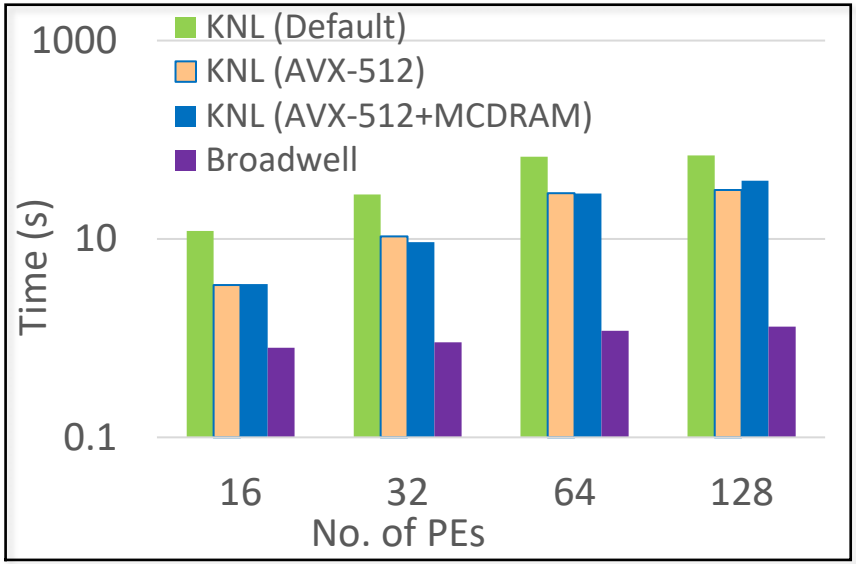
Heat Image Kernel



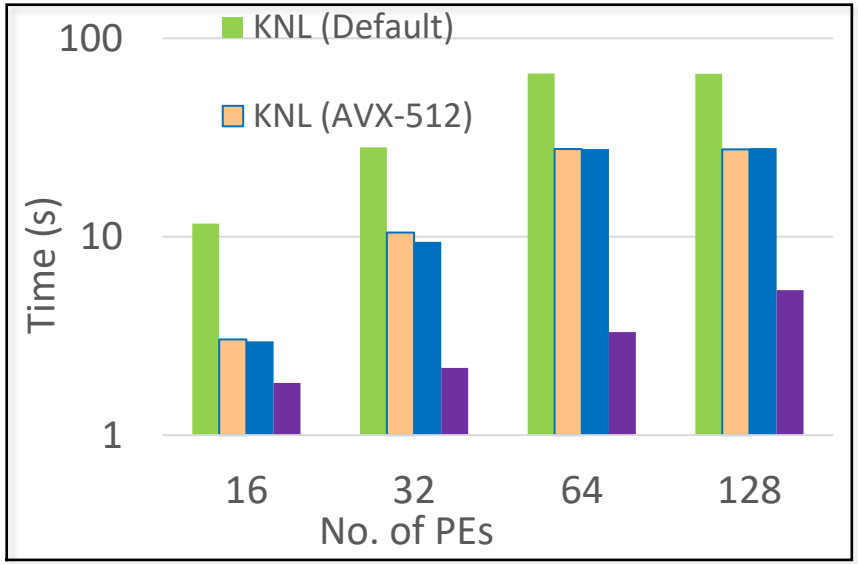
- On heat diffusion based kernels AVX-512 vectorization showed better performance
- MCDRAM showed significant benefits on Heat-Image kernel for all process counts. Combined with AVX-512 vectorization, it showed up to 4X improved performance

Application Kernels Evaluation (contd.)

Matrix Multiplication kernel



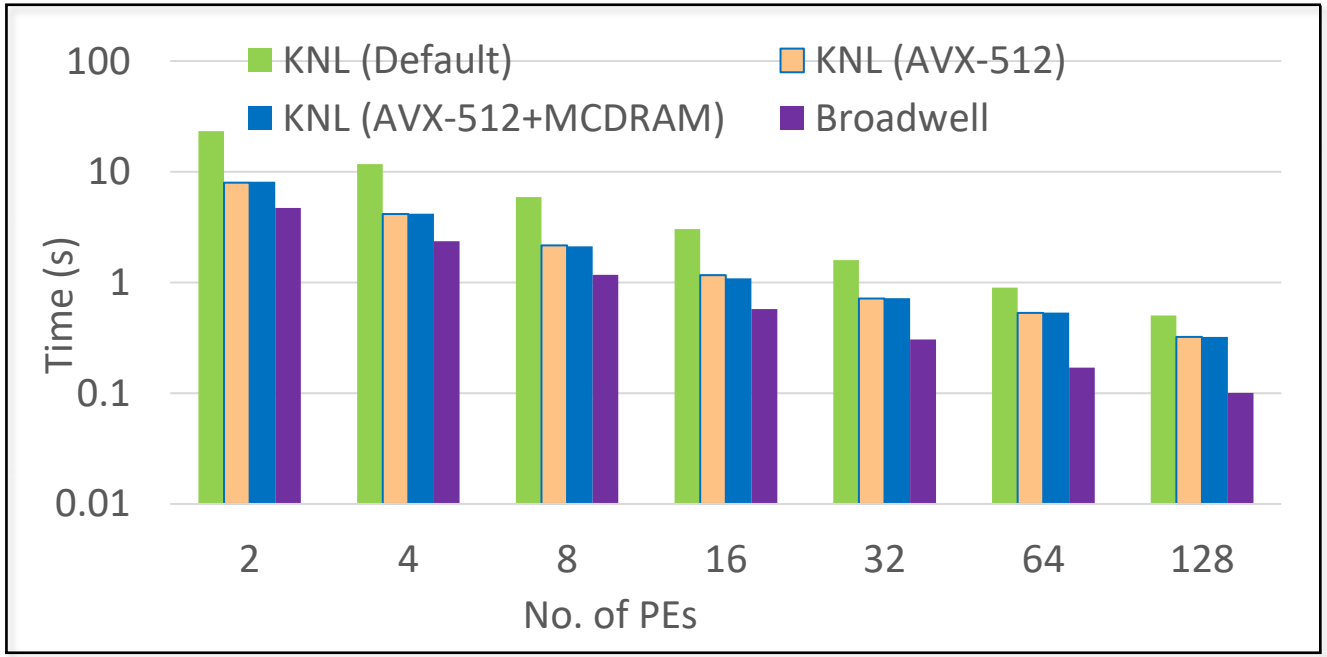
DAXPY kernel



- Vectorization helps in matrix multiplication and vector operations.
- Due to heavily compute bound nature of these kernels, MCDRAM didn't show any significant performance improvement.

Application Kernels Evaluation (contd.)

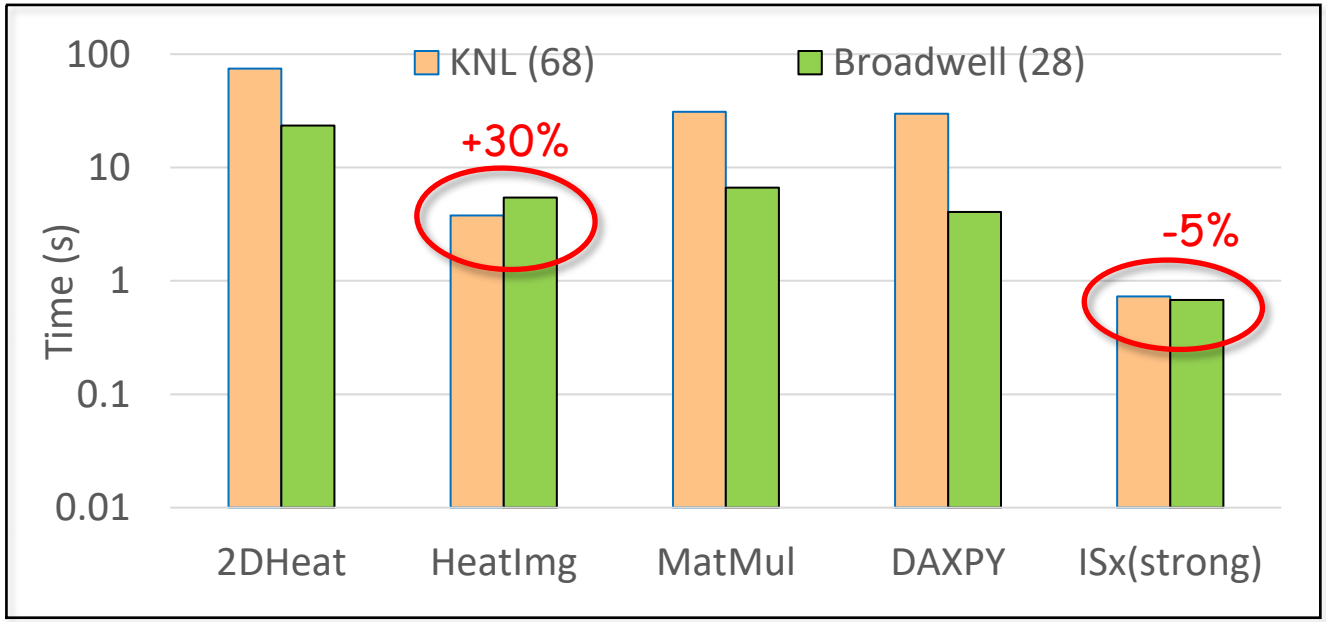
Scalable Integer Sort Kernel (ISx)



- Up to 3X improvement on un-optimized execution is observed on KNL
- Broadwell showed up to 2X better performance for core-by-core comparison

Node-by-node Evaluation using Application Kernels

Application Kernels on a single KNL vs. single Broadwell node



- A single node of KNL is evaluated against a single node of Broadwell using all the available physical cores
- HeatImage and ISx kernels, showed better performance than Intel Xeon

Outline

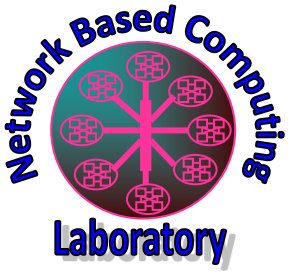
- Introduction
- Motivation
- Contributions
- Evaluation Methodology
- Results and Discussion
- **Conclusion**

Conclusion

- Comprehensive performance evaluation of MVAPICH2-X based OpenSHMEM over the KNL architecture
  - Intra- and inter-node comparison with Broadwell
  - Microbenchmarks and application kernels
- Observed significant performance gains on application kernels when using AVX-512 vectorization
  - 2.5x performance benefits in terms of execution time
- MCDRAM benefits are not prominent on most of the application kernels
  - Lack of memory bound operations
- KNL showed up to 3X worse performance than Broadwell for core-by-core evaluation
- KNL showed better or on-par performance than Broadwell on Heat-Image and ISx kernels for Node-by-Node evaluation
- The runtime implementations need to take advantage of the concurrency of KNL
  - Multi-threaded OpenSHMEM runtimes

Thank You!

{hashmi.29, li.2192, subramoni.1, panda.2}@osu.edu



MVAPICH

MPI, PGAS and Hybrid MPI+PGAS Library

Network-Based Computing Laboratory

http://nowlab.cse.ohio-state.edu/

MVAPICH Web Page

http://mvapich.cse.ohio-state.edu/