Benchmarking Parallel Performance on Many-Core Processors

CHREC
NSF Center for High-Performance Reconfigurable Computing

OpenSHMEM Workshop 2014

Bryant C. Lam (speaker)
Ajay Barboza
Ravi Agrawal
Dr. Alan D. George
Dr. Herman Lam

NSF Center for High-Performance Reconfigurable Computing (CHREC), University of Florida

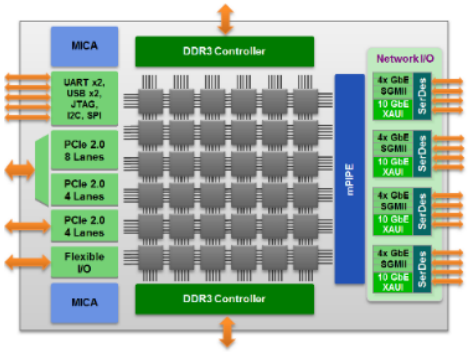
March 4-6, 2014

Motivation and Approach

- Motivation
  - Emergent many-core processors in HPC and scientific computing require performance profiling with existing parallelization tools, libraries, and models
    - HPC typically distributed cluster systems of multi-core devices
      - New shifts toward heterogeneous computing for better power utilization
    - Can many-core processors replace several servers?
    - Can computationally dense servers of many-core devices scale?
    - Can many-core replace other accelerators (e.g., GPU) in heterogeneous systems?
- Approach
  - Evaluate architectural strengths of two current-generation many-core processors
    - Tilera TILE-Gx8036 and Intel Xeon Phi 5110P
  - Evaluate many-core app performance and scalability with ***SHMEM and OpenMP*** on these many-core processors

Overview

- Devices
  - Tilera TILE-Gx
  - Intel Xeon Phi

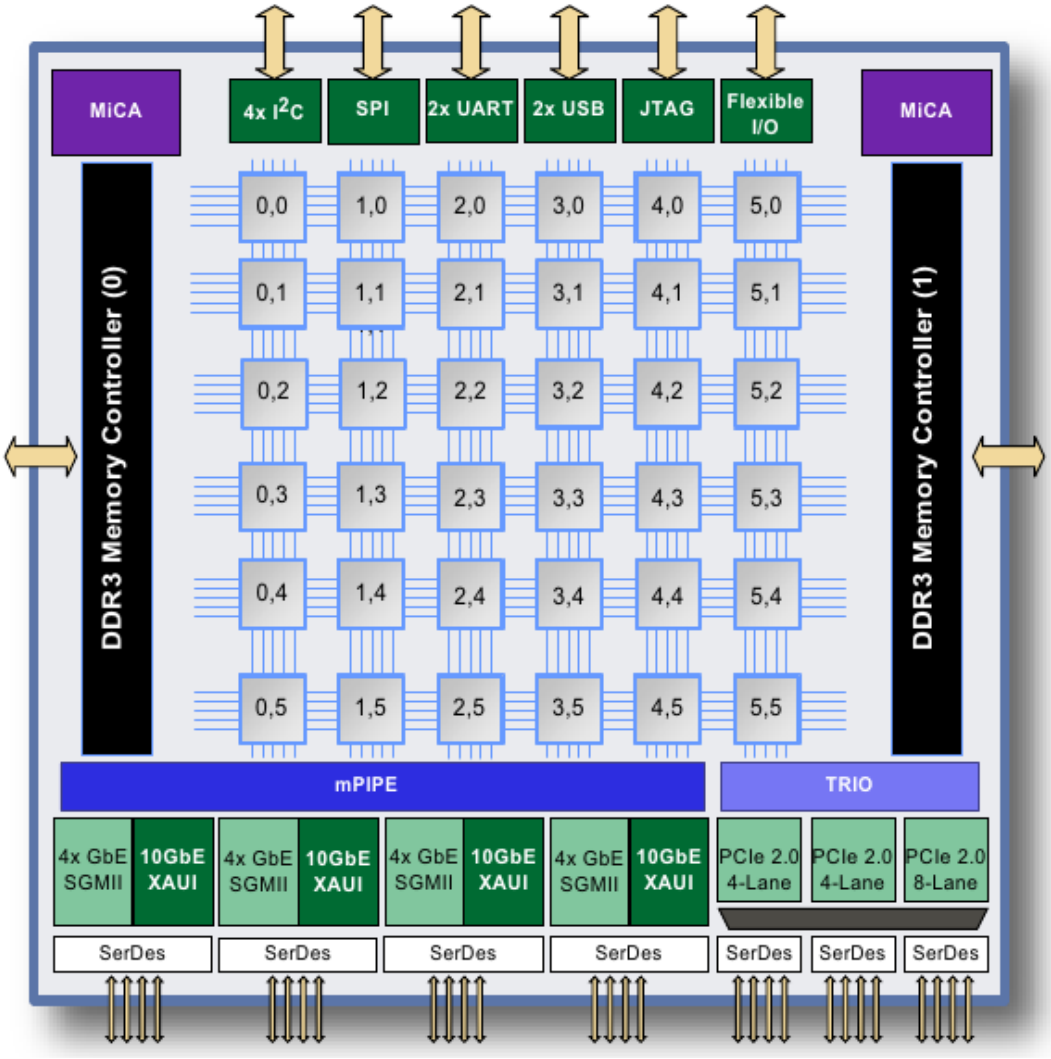


- Benchmarking Parallel Applications
  - SHMEM and OpenMP applications
  - SHMEM-only applications

- Conclusions

Tilera TILE-Gx8036 Architecture

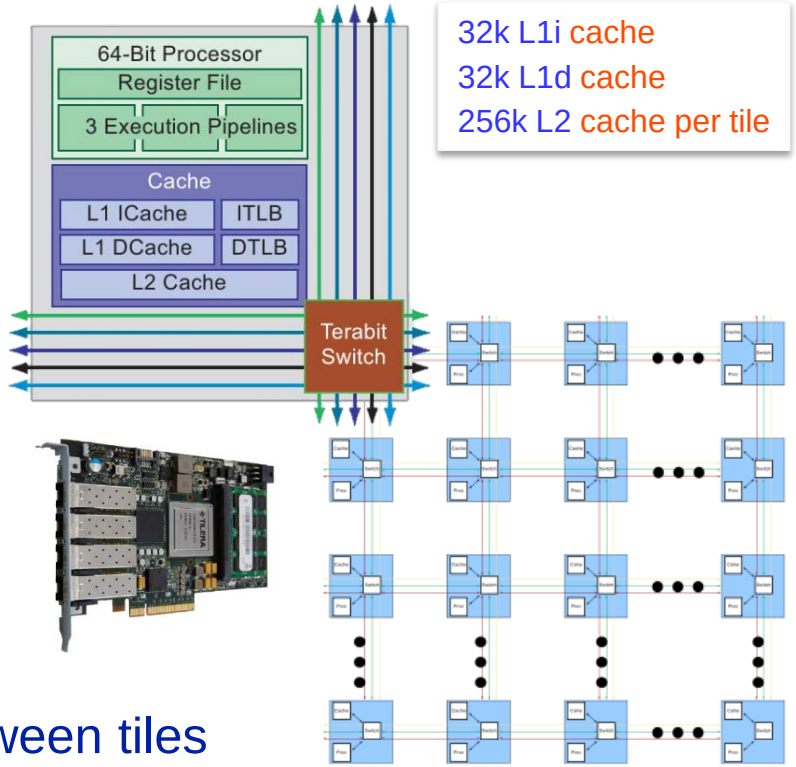
- 64-bit VLIW processors
- 32k L1i cache, 32k L1d cache
- 256k L2 cache per tile
- Up to 750 billion operations per second
- Up to 60 Tbps of on-chip mesh interconnect
- Over 500 Gbps memory bandwidth
- 1 to 1.5 GHz operating frequency
- Power consumption: 10 to 55W; 22W for typical applications
- 2-4 DDR3 memory controllers
- mPIPE delivers wire-speed packet classification, processing, distribution
- MiCA for cryptographic acceleration
- Available in stand-alone server or PCIe



VLIW: very long instruction word
mPIPE: multicore Programmable Intelligent Packet Engine
MiCA: Multicore iMesh Coprocessing Accelerator

Tilera iMesh Interconnect Fabric

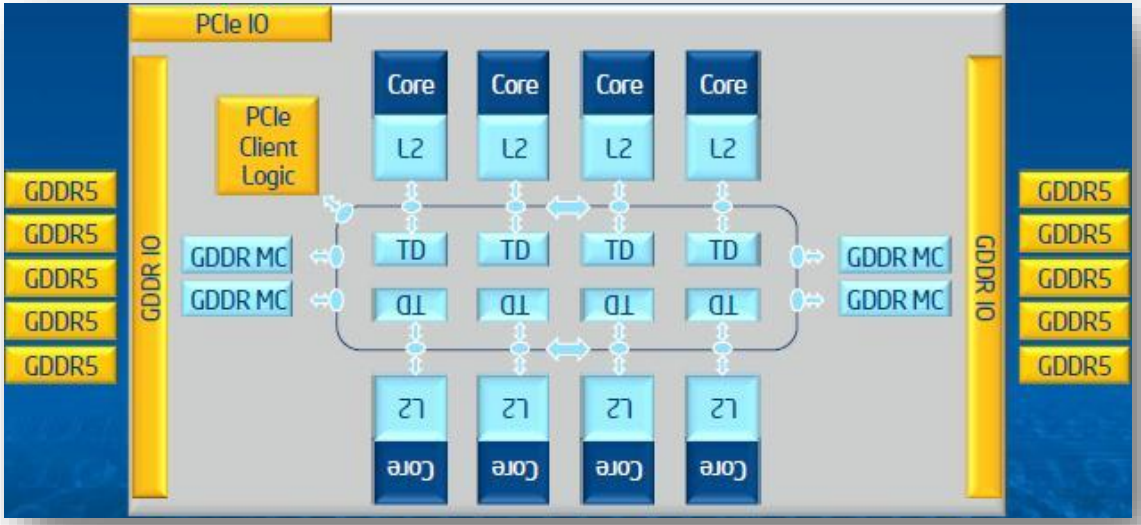
- ***Tiles*** consist of processing core, cache, and switch
- Switches attach to mesh fabric via ***five networks***
  - QDN (reQuest)
    RDN (Response)
    - Two networks for improved memory-access bandwidth
  - SDN (Share)
    - Cache access and coherence
  - IDN (Internal)
    - Communication with external I/O
  - UDN (User)
    - User-accessible dynamic network for low-latency packet transfers between tiles

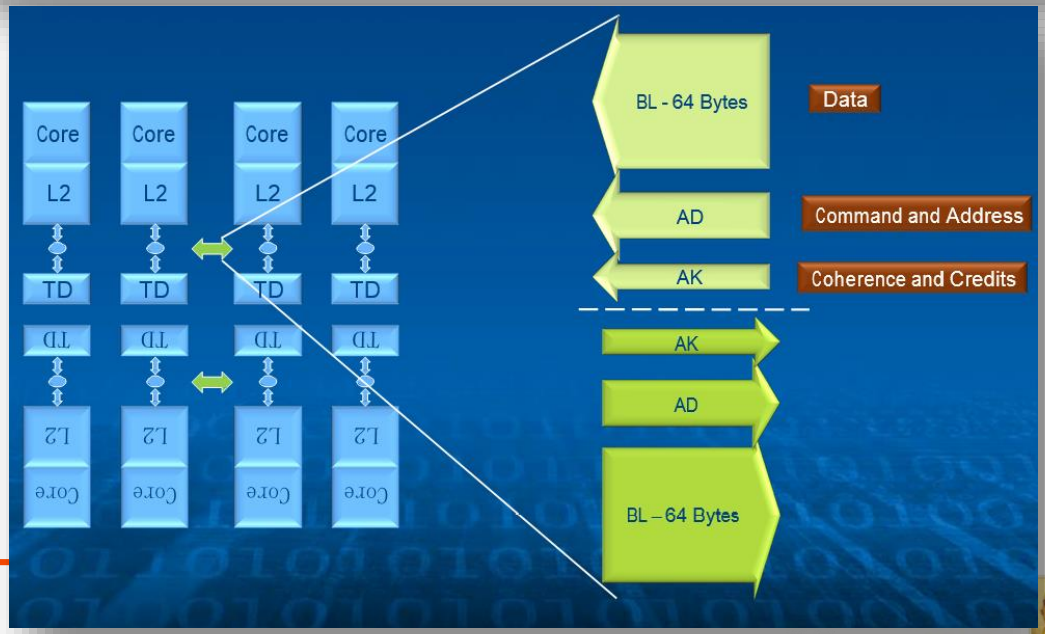


http://software.intel.com/en-us/articles/intel-xeon-phi-coprocessor-codename-knights-corner

Intel Xeon Phi 5110P Architecture

- 60 cores up to 240 threads
- 32k L1i cache, 32k L1d cache
- 512k L2 cache per core
- 8 GB GDDR5 memory
- 320 GB/s bandwidth
- PCIe x16 form factor, passively cooled
- 512-bit SIMD instructions
- 1.053 GHz
- 225W TDP
- Native and Offload compute models
- Intel compiler and library support
  - Math Kernel Library
  - OpenMP
  - Intel MPI, Intel Clik

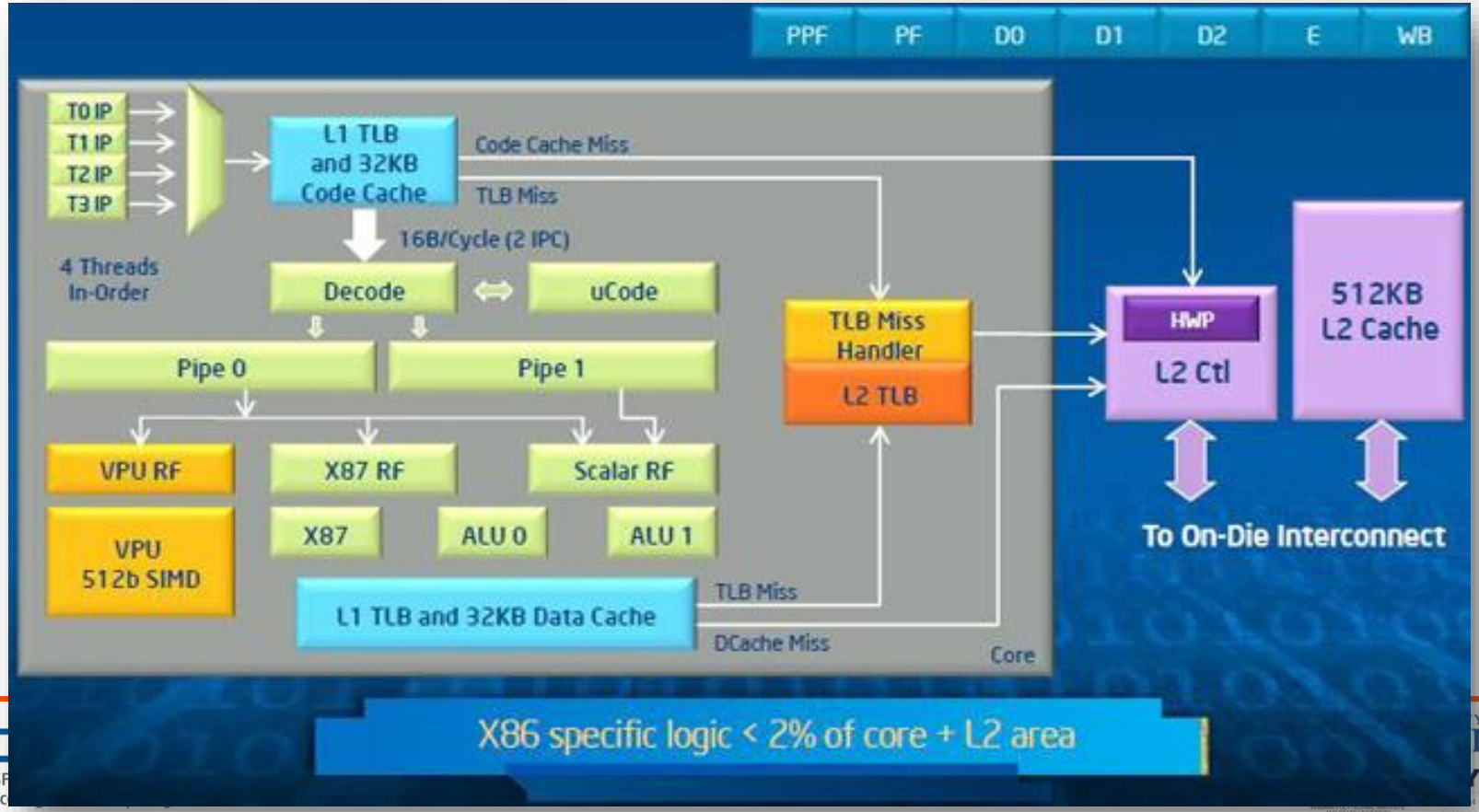




http://software.intel.com/en-us/articles/intel-xeon-phi-coprocessor-codename-knights-corner

Xeon Phi Coprocessor Core

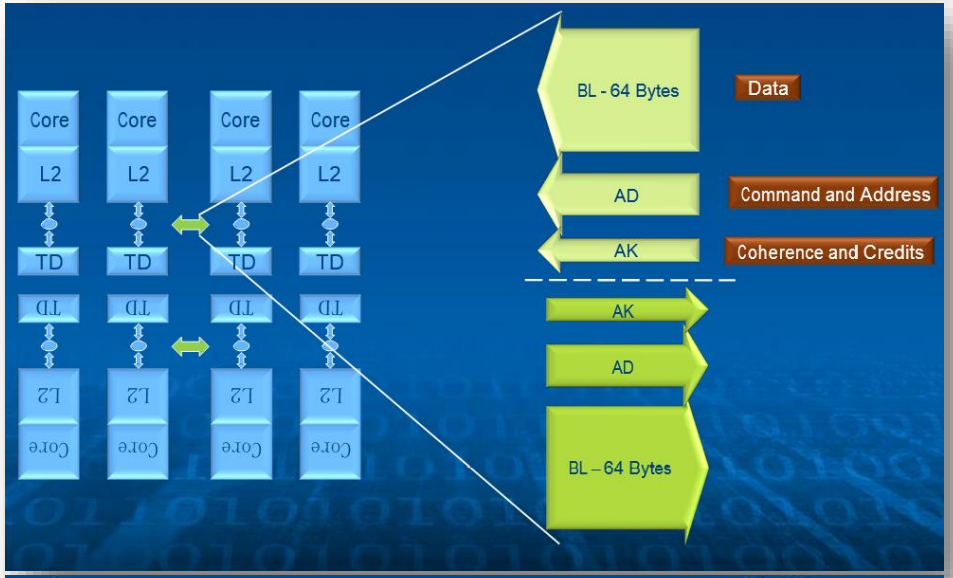
- Coprocessor core features x86-like instruction set
  - 4-way Simultaneous Multithreading (SMT)
  - In-order instruction processing
  - Wide 512-bit SIMD vector units

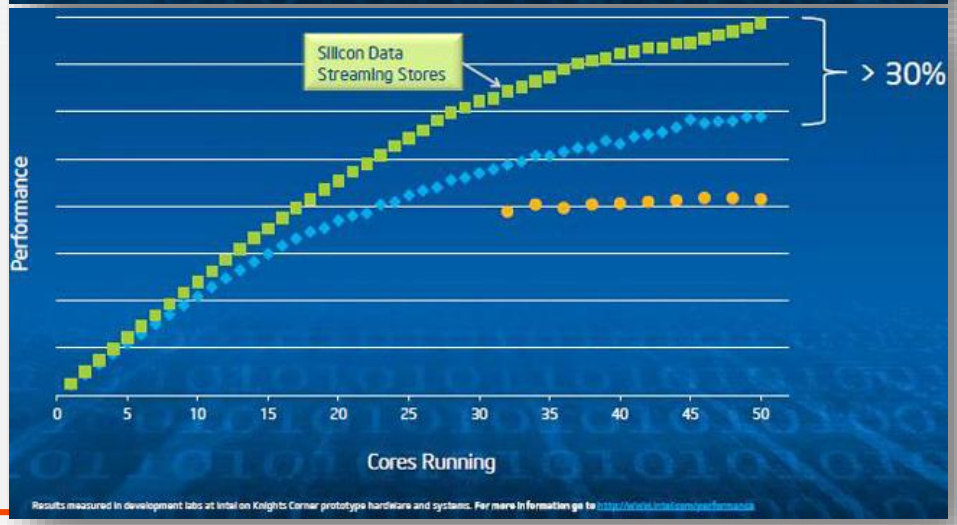


http://software.intel.com/en-us/articles/intel-xeon-phi-coprocessor-codename-knights-corner

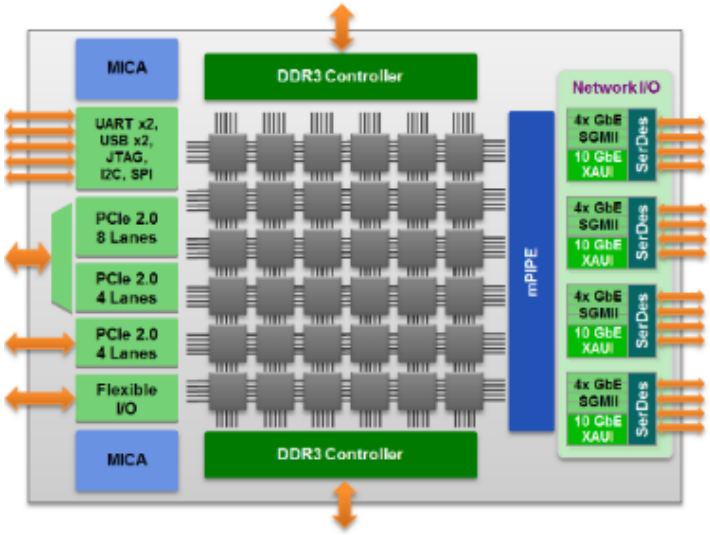
Xeon Phi Memory Accesses

- Ring Interconnect
  - High-bandwidth wide interconnect feeds cores from other caches and GDDR5 memory
- Streaming Stores
  - for (i = 0; i < HUGE; i++)
    A[i] = k*B[i] + C[i];
  - Historically, A is read from cache, then written
  - With streaing stores: write cache line of A without unnecessary read
    - Boosts memory bandwidth





Device Comparison



Tilera TILE-Gx8036

36 tiles of 64-bit VLIW processors

32k L1i, 32k L1d, 256k L2 cache per tile, 9MB shared L2

Up to 750 billion operations per second

60 Tbps of on-chip mesh interconnect

Over 500 Gbps memory bandwidth

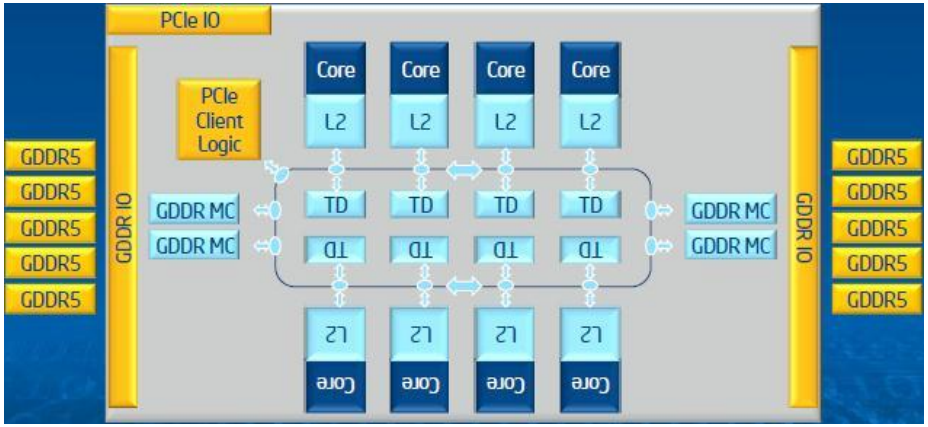
1.0 to 1.5 GHz operating frequency

10 to 55W (**22W typical**)

2 DDR3 memory controllers

mPIPE for 10×4 Gbps wire-speed packet processing

MiCA for crypto and compression



Intel Xeon Phi 5110P

60 cores with up to 240 threads

32k L1i, 32k L1d, 512k L2 cache per core, 30 MB shared L2

8 GB GDDR5 memory

320 GB/s bandwidth

PCIe x16 form factor, passively cooled

512-bit SIMD instructions

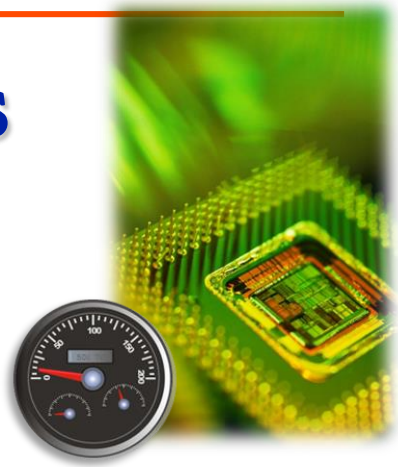
1.053 GHz

225W TDP

Native and Offload compute models

Intel compiler and library support

Benchmarking Parallel Apps



- Evaluate ***scalability of parallel apps*** on TILE-Gx and Xeon Phi

- Leverage SHMEM and OpenMP for parallelization
  - TILE-Gx
    - OpenMP
    - OpenSHMEM reference implementation atop GASNet
    - CHREC ongoing project: ***TSHMEM for TILE-Gx***
  - Xeon Phi
    - OpenMP
    - OpenSHMEM reference implementation atop GASNet

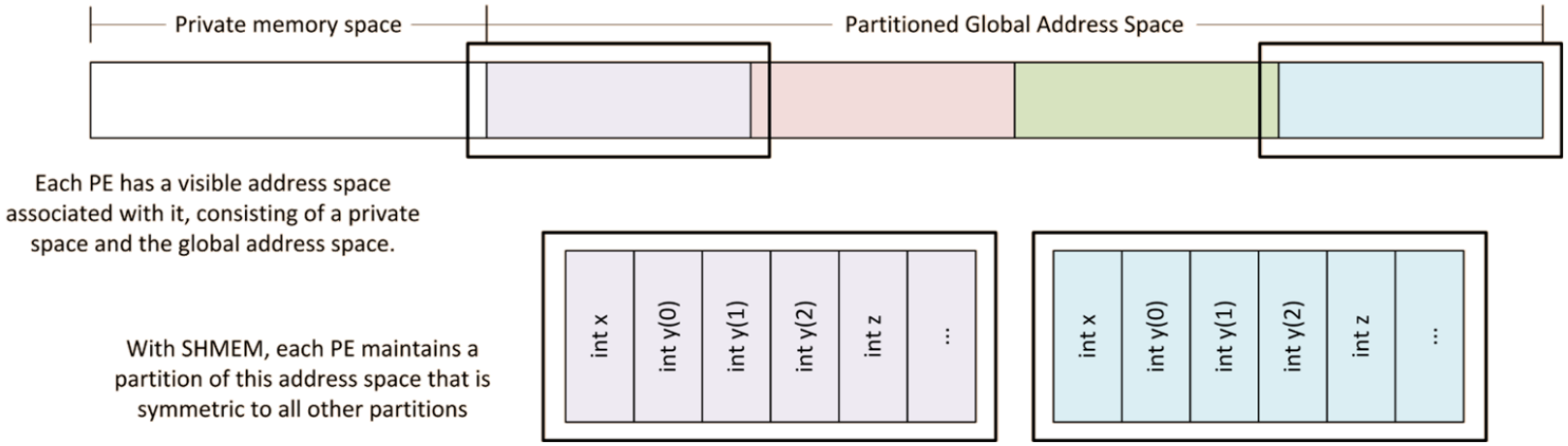
OpenMP

- OpenMP first standardized in 1997
  - ***Thread parallelization*** with fork/join model
  - Programming via ***compiler directives*** in C/C++ or Fortran
    - Example: #pragma omp parallel for

  - ✓ Incremental parallelization of sequential applications
  - ✓ Commonly available via compilers
  - Potential difficulties with race conditions, false sharing, and non-optimal serialization in parallel sections
  - Limited to local SMP devices; Little to non-existent support for programming across multiple devices

Introduction to PGAS

- PGAS = partitioned global address space
  - Share data by dividing the global address space into local and remote partitions
    - Hardware and software infrastructure handle transmission of data when not local
  - Gives a ***shared-memory view*** of distributed-memory cluster systems
  - PGAS languages and libraries
    - SHMEM, UPC, Cray Chapel, IBM X10, Co-Array Fortran, Titanium



SHMEM

- SHMEM developed originally in 1993 for Cray T3D distributed-memory supercomputer
  - Standardized via OpenSHMEM community (January 2012)
  - ***Process parallelization*** with single-program, multiple-data process replication (e.g., mpirun)
  - Programming via ***library functions*** in C/C++ or Fortran
    - Features one-sided point-to-point and collective operations
    - Example: *shmem_int_put()*, *shmem_broadcast32()*

✓ Potential to replace MPI+OpenMP model common in HPC with ability to handle distributed systems of SMP devices

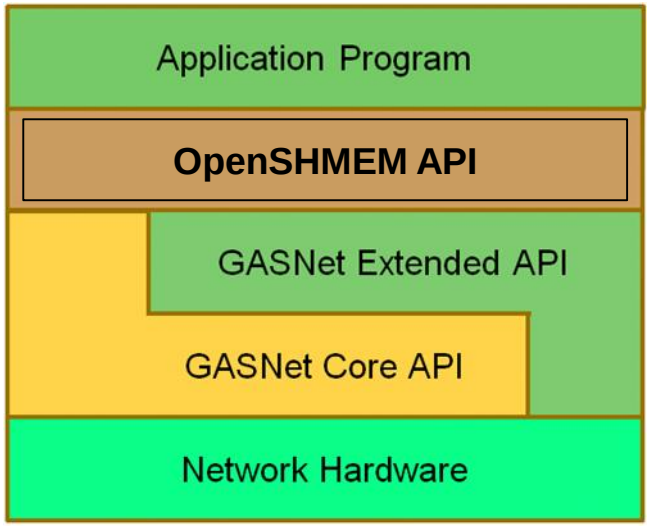
✓ PGAS via library functions; No need for compiler support

- Large, shared, data structures difficult to handle if data is not amenable for symmetric distribution to participating PEs

OpenSHMEM

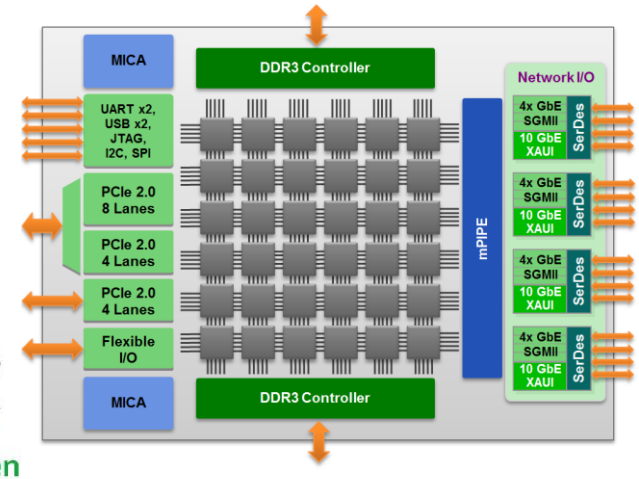
- OpenSHMEM community now manages development of future SHMEM specifications
  - Reference implementation and test suite also provided

- OpenSHMEM reference impl. built atop ***GASNet***
  - GASNet is a network and communication abstraction library
    - Supports hardware networking technologies (e.g., InfiniBand)
    - Enables portability between networking hardware if GASNet API used in your application or library
    - Enables support for vast majority of cluster-based systems
  - We use GASNet's SMP abstraction with OpenSHMEM for TILE-Gx and Xeon Phi benchmarking



TSHMEM Overview

HPC acceleration with SHMEM on many-core processors



SHMEM reference design directly over *Tilera TILE-Gx* architecture and libraries

- Stay true to SHMEM principles
  - High performance with low overhead
  - Portability with ***easy programmability***
- Maximize architectural benefits
  - Tile interconnect, mPIPE, MiCA
- Extend design to multi-device systems
  - Evaluate interconnect capabilities
- Explore design on other many-core devices including Intel Xeon Phi

Open SHMEM

OpenSHMEM and TSHMEM

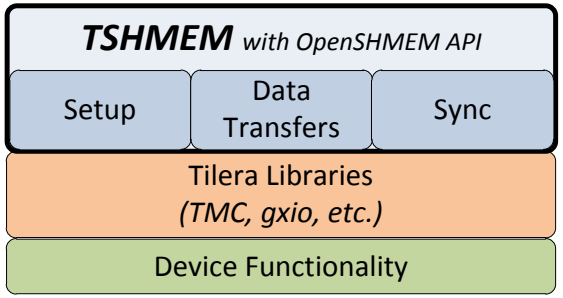
Achieved

- Dynamic symmetric heap management
- Point-to-point data transfer
- Point-to-point synchronization
- Barrier synchronization
- Broadcast, Collection, Reduction
- Atomics for dynamic variables
- Extension to multiple many-core devices

Ongoing

- Port of TSHMEM to Intel Xeon Phi
- Exploration of new SHMEM extensions

TILERA



Modular design utilizing vendor libraries

TSHMEM reference design on TILE-Gx36

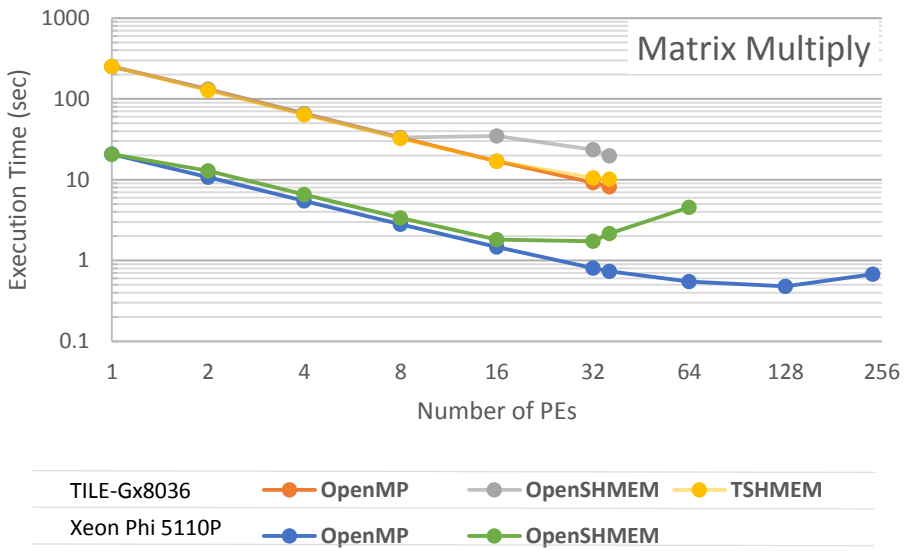
Applications Benchmarking

- Three apps focused on SHMEM vs. OpenMP performance with same computation and communication patterns
  - Matrix multiply
  - Linear curve fitting
  - Exponential curve fitting
- Four apps focused on SHMEM-only performance between TILE-Gx and Xeon Phi
  - OSH matrix multiply
  - OSH heat image
  - Huge async radix sort
  - FFTW with SHMEM

Apps – Matrix Multiplication

- Matrix multiply common in many applications ($C = A \times B$)
  - For OpenMP version, all three matrices are shared and accessible from any thread
  - For SHMEM version, *A* and *C* matrices distributed; *B* matrix is private copy
    - Large memory consumption in SHMEM; OpenMP uses *shared* compiler directive
    - SHMEM version should use a more distributed implementation, but the chosen implementation preserves computation pattern with OpenMP version
    - OpenMP has advantage; SHMEM not amenable for large non-distributed data structures (in this case, matrix B)

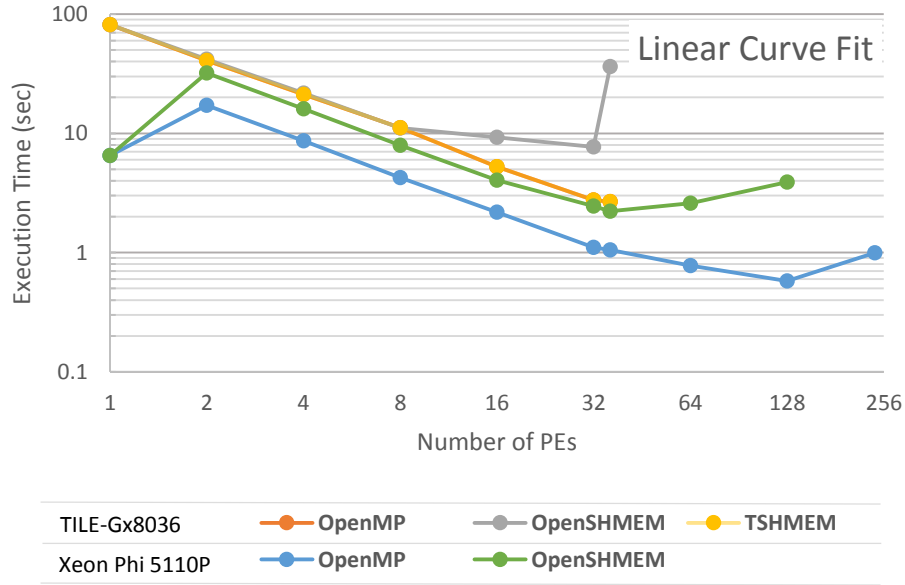
- TILE-Gx
  - TSHMEM and OpenMP execution times are very similar
  - OpenSHMEM scalability concerns with more than 8 PEs (1/4 of device)
- Xeon Phi
  - OpenMP scales well up to 128 PEs
  - OpenSHMEM scalability concerns with more than 16 PEs (7% of possible threads)



Apps – Linear Curve Fitting

- Linear curve fitting performs linear least-squares approximation on a set of points
  - Calculates minimum least-squares deviation from a set of points and approximates a linear-regression line

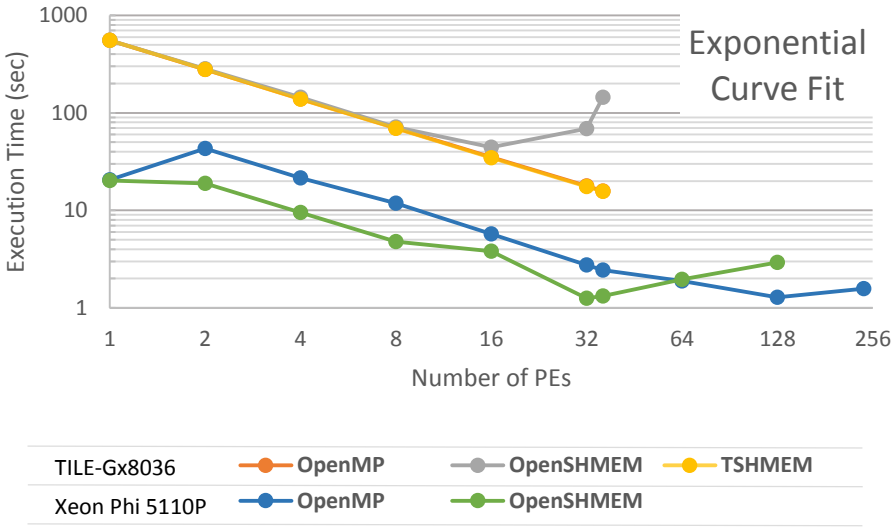
- TILE-Gx
  - TSHMEM and OpenMP performance approximately equivalent
  - OpenSHMEM shows scalability concerns after 8 PEs
- Xeon Phi
  - OpenMP outperforms OpenSHMEM
  - Similar performance between TSHMEM on TILE-Gx and OpenSHMEM on Xeon Phi at 32 and 36 PEs



Apps – Exponential Curve Fitting

- Exponential curve fitting can leverage linear curve fitting algorithm by transforming exponential equation into linear equation with logarithms

- TILE-Gx
  - Similar performance with TSHMEM and OpenMP
  - OpenSHMEM shows scaling issues beyond 16 PEs
- Xeon Phi
  - OpenSHMEM is faster than OpenMP with parity performance at 64 PEs
  - Bottleneck in OpenMP is parallel reduction operation with logarithms in loop body and subsequent sync; SHMEM version avoids this

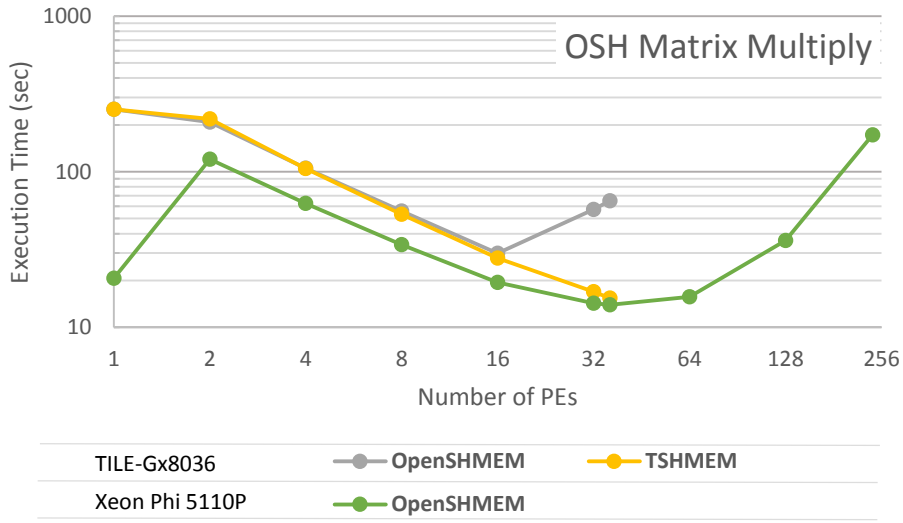


Apps – OSH Matrix Multiplication

- Next four applications are SHMEM-only applications to compare optimal SHMEM performance between ***TILE-Gx*** and ***Xeon Phi***
  - Emphasis on performance comparison between SHMEM implementations

- This matrix multiplication is provided by OpenSHMEM test suite
- Same serial baseline as previous matrix multiply, but this app has different data structure arrangement
  - *A*, *B*, and *C* are distributed data structures across all SHMEM partitions
  - More communication with *B* matrix

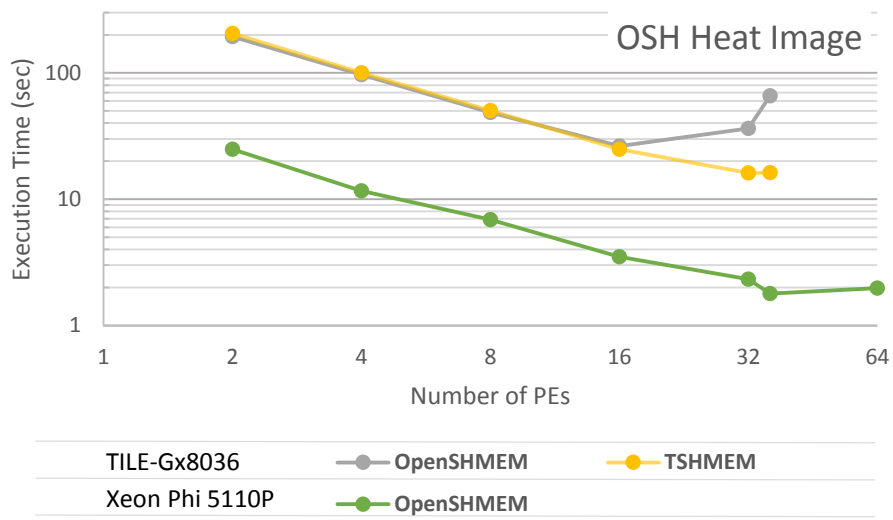
- TILE-Gx
  - OpenSHMEM scalability issues after 16 PEs
- Xeon Phi
  - Due to more optimal serial baseline, OpenSHMEM performance heavily suffers
  - TILE-Gx TSHMEM performance very comparable to Xeon-Phi OpenSHMEM performance



Apps – OSH Heat Image

- Also from OpenSHMEM test suite, this app does heat-convection modeling and outputs an image

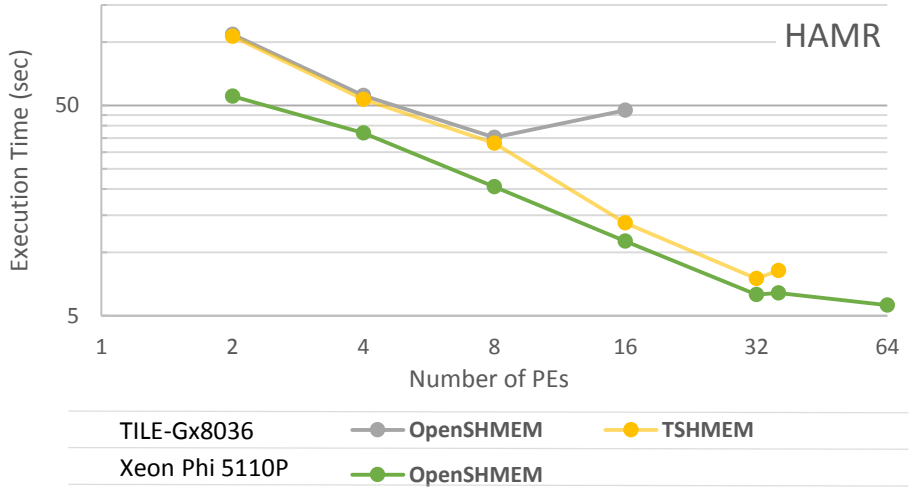
- TILE-Gx
  - Similar performance between OpenSHMEM and TSHMEM until 16 PEs
- Xeon Phi
  - OpenSHMEM scales well
  - Application runtime checks prevent execution for PE counts > 64



Apps – Huge Async Radix Sort

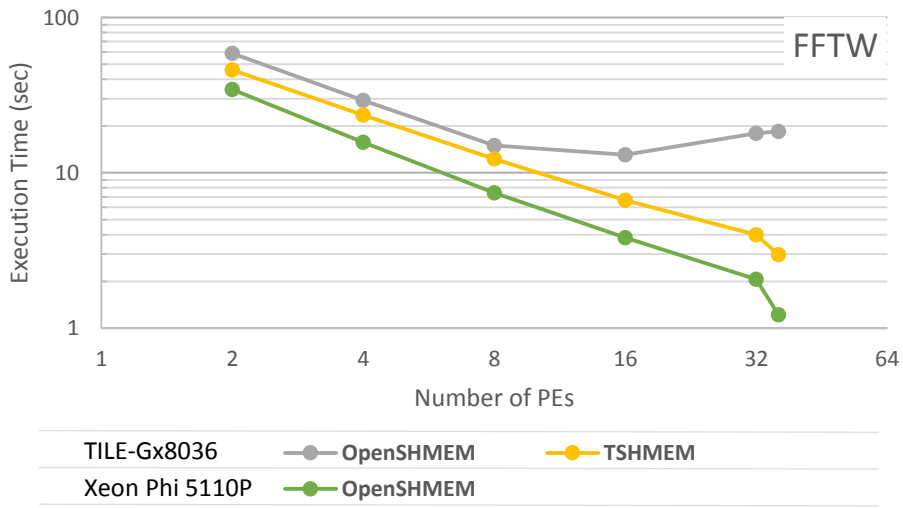
- Pseudo-application which performs large, asynchronous radix sort
  - Randomly generated values to fill requested memory
  - All-to-all communication for key-exchange phase
    - Majority of application's runtime
  - Very quick integer sorting after key exchange

- TILE-Gx
  - OpenSHMEM scalability concerns beyond 8 PEs
  - This application is very amenable for TILE-Gx due to integer comparisons
- Xeon Phi
  - OpenSHMEM scales well, but scaling considerably decreases at higher PE counts
    - Very favorable performance with TSHMEM on TILE-Gx at 32 PEs, especially after normalization of device power consumption

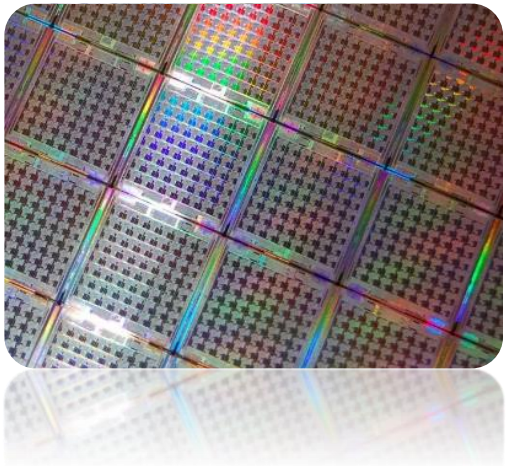


Apps – FFTW with SHMEM

- FFTW is popular threaded library for very fast DFT operations
  - Process-level parallelization of threaded FFTW library using SHMEM
- TILE-Gx
  - Unlike previous applications that were similar in performance, TSHMEM is 20% faster than OpenSHMEM
  - OpenSHMEM scalability issues beyond 8 PEs
- Xeon Phi
  - OpenSHMEM scales well
  - TSHMEM on TILE-Gx has comparable performance to OpenSHMEM on Xeon Phi
    - Surprising result given strength of floating-point performance on Xeon Phi



Conclusions



- **SHMEM and OpenMP applications**
  - TSHMEM and OpenMP exhibit similar performance on TILE-Gx
  - OpenSHMEM exhibits scalability concerns
  - Faster execution times with Xeon Phi over TILE-Gx
    - Performance per watt, however, still requires further exploration
- **Independently developed SHMEM-only applications**
  - TSHMEM outperforms OpenSHMEM for all SHMEM applications on TILE-Gx
    - Justifies TSHMEM approach of bare-metal library design for many-core performance
    - OpenSHMEM reference library designed for portability to distributed cluster systems
    - At this moment, TSHMEM is only for Tilera devices
      - Work underway for portability to Xeon Phi
- **Questions?**
  - Contact me at ***blam@chrec.org***
  - More about CHREC at http://www.chrec.org/