Accelerating HPL on Heterogeneous GPU Clusters

Presentation at GTC 2014

by

Dhabaleswar K. (DK) Panda

The Ohio State University

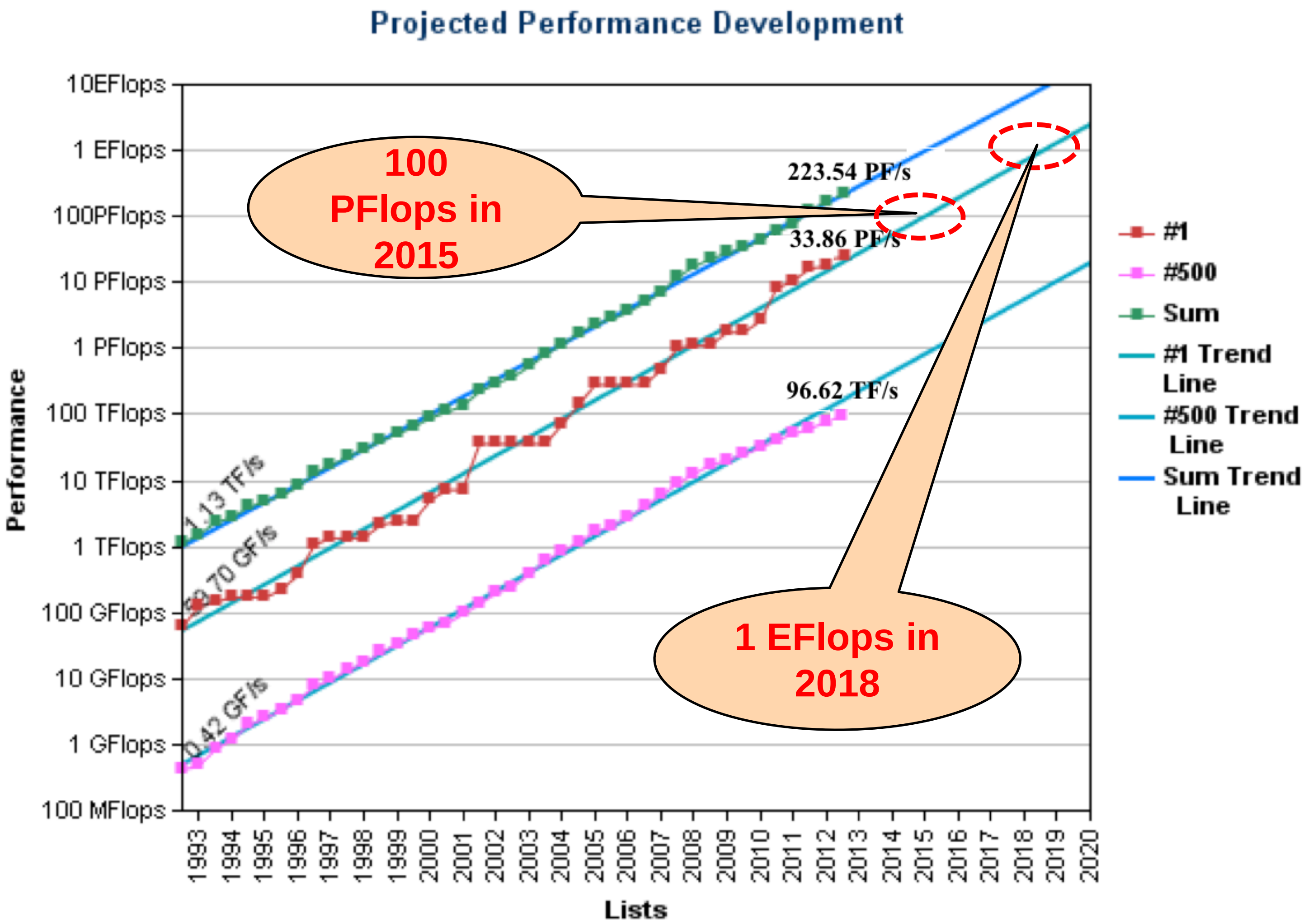
E-mail: panda@cse.ohio-state.edu

http://www.cse.ohio-state.edu/~panda

Outline

- Introduction and Motivation
- Proposed Design for Hybrid HPL
- Performance Evaluation
- Conclusion and Future Work

High-End Computing (HEC): PetaFlop to ExaFlop



Expected to have an ExaFlop system in 2019 -2022!

Drivers of Heterogeneous HPC Cluster

Multi-core Processors

Accelerators / Coprocessors
high compute density, high performance/watt
>1 TFlop DP on a chip

- Multi-core processors are ubiquitous
- High Performance Linpack (HPL) is used to measure the peak performance
- NVIDIA GPUs are becoming common in high-end systems
- Pushing the envelope for heterogeneous computing

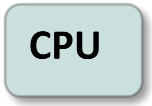
Tianhe – 1A

Stampede

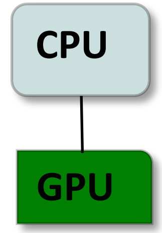
Oakley (OSC)

Blue Waters (NCSA)

GPU Cluster configurations

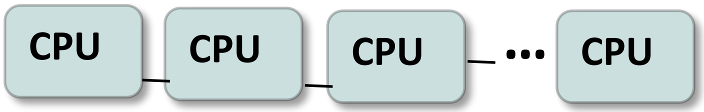


CPU Node

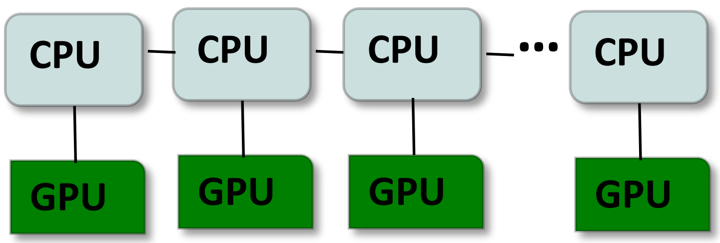


GPU Node

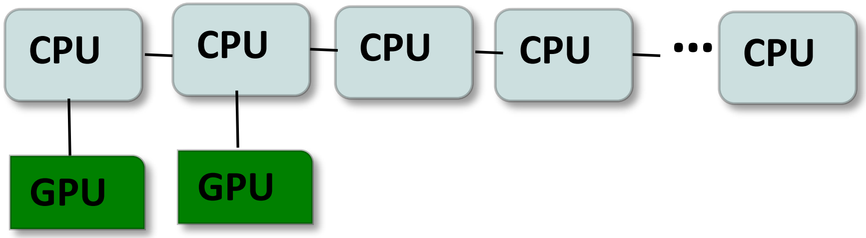
- Homogeneous CPU Clusters
  - No GPU accelerators (ex: BlueGene)



- Homogeneous GPU Clusters
  - All nodes have the same configuration
  - Titan, Keeneland, Wilkes



- Heterogeneous GPU Clusters
  - CPU nodes + GPU nodes
  - Ratio CPU/GPU > 1
  - Oakley@OSC: 634 CPU + 64 GPU
  - BlueWaters@NCSA: 22,500 XE + 4200 XK



HPL - High Performance Linpack

- Benchmark

  Performance measure for ranking supercomputers in the top500 list

- Time Complexity: $\frac{2}{3}N^3 + 2N^2 + O(N)$ N is the problem size

  $O(N^3)$: LU Decomposition

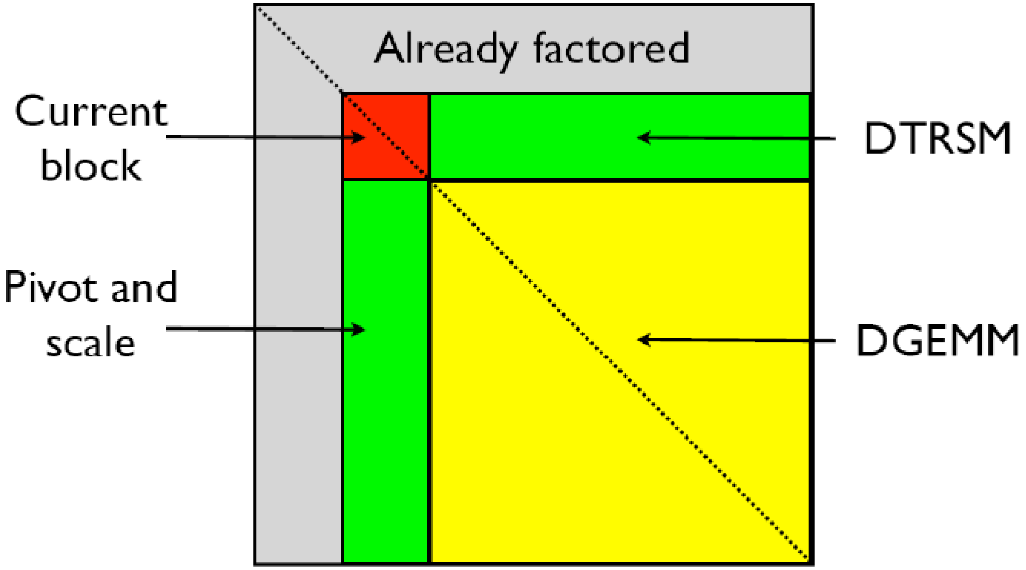
  $O(N^2)$: Backward Substitution

- Iterative Procedure of LU

  Factorize the current block

  Broadcast and update the green parts

  Update the yellow parts



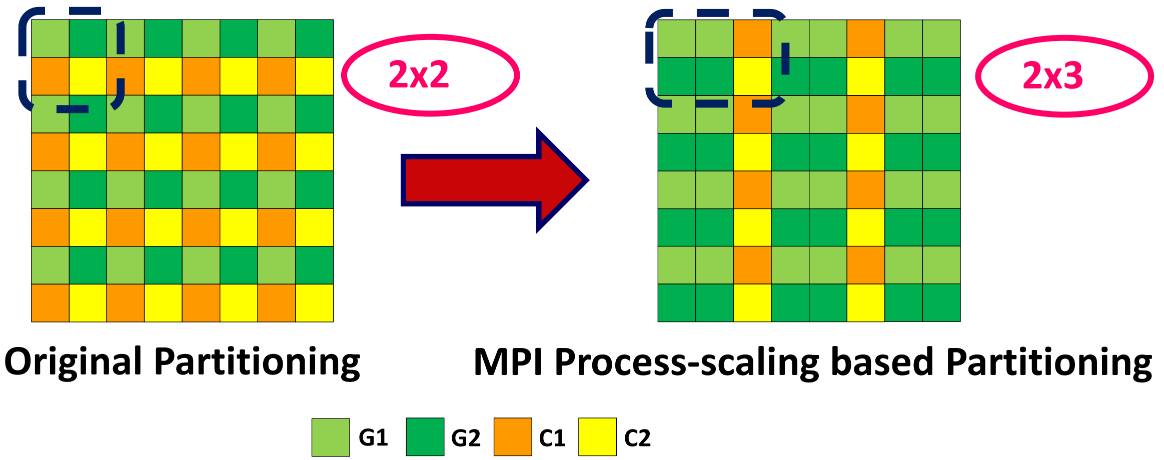
Current Execution of HPL on Heterogeneous GPU Clusters

- Current HPL support for GPU Clusters
  - Heterogeneity inside a node CPU+GPU
  - Homogeneity across nodes
- Current HPL execution on heterogeneous GPU Clusters
  - Only CPU nodes (using all the CPU cores)
  - Only GPU nodes (using CPU+GPU on only GPU nodes)
  - As the ratio CPU/GPU is higher => report the "Only CPU" runs
- Need for HPL which supports heterogeneous systems
  - Heterogeneity inside a node (CPU+GPU)
  - Heterogeneity across nodes (nodes w/o GPUs)

Outline

- Introduction and Motivation
- Proposed Design for Hybrid HPL
  - Two-level Workload Partitioning (Inter-node and Intra-node)
  - Process-grid Reordering
- Performance Evaluation
- Conclusion and Future Work

Two Level Workload Partitioning: Inter-node



- **Inter-node Static Partitioning**

  Original design: uniform distribution, bottleneck on CPU nodes

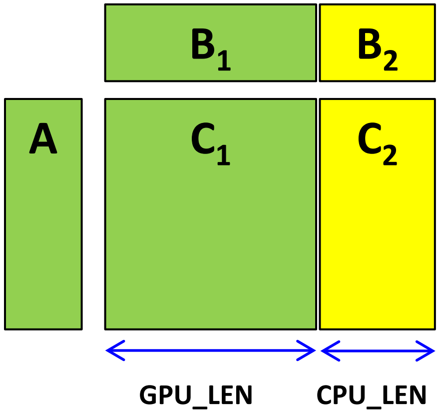
  New design: identical block size, schedules more MPI processes on GPU nodes

  MPI_GPU = ACTUAL_PEAK_GPU / ACTUAL_PEAK_CPU + β

  (NUM_CPU_CORES mod MPI_GPU = 0)

  Evenly split the cores

Two Level Workload Partitioning: Intra-node



- Intra-node Dynamic Partitioning
  - MPI-to-Device Mapping

    Original design: 1:1

    New design: M: N (M > N), N= number of GPUs/Node, M= number of MPI processes

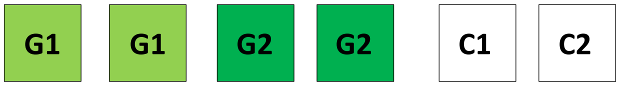
  - Initial Split Ratio Tuning: alpha = GPU_LEN / (GPU_LEN + CPU_LEN)

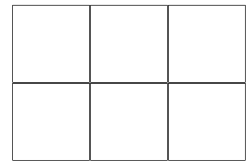
    Fewer CPU cores per MPI processes

    Overhead caused by scheduling multiple MPI processes on GPU nodes

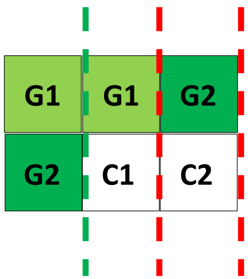
Process Grid Reordering

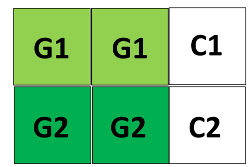
- **Default (blocking) Process Grid with Multiple MPI processes/GPU nodes**





2 x 3 Grid





- **Synchronization overhead of Panel Broadcast**

- Default Design

G1 → G1 → G2
G2 → C1 → C2

- New Design

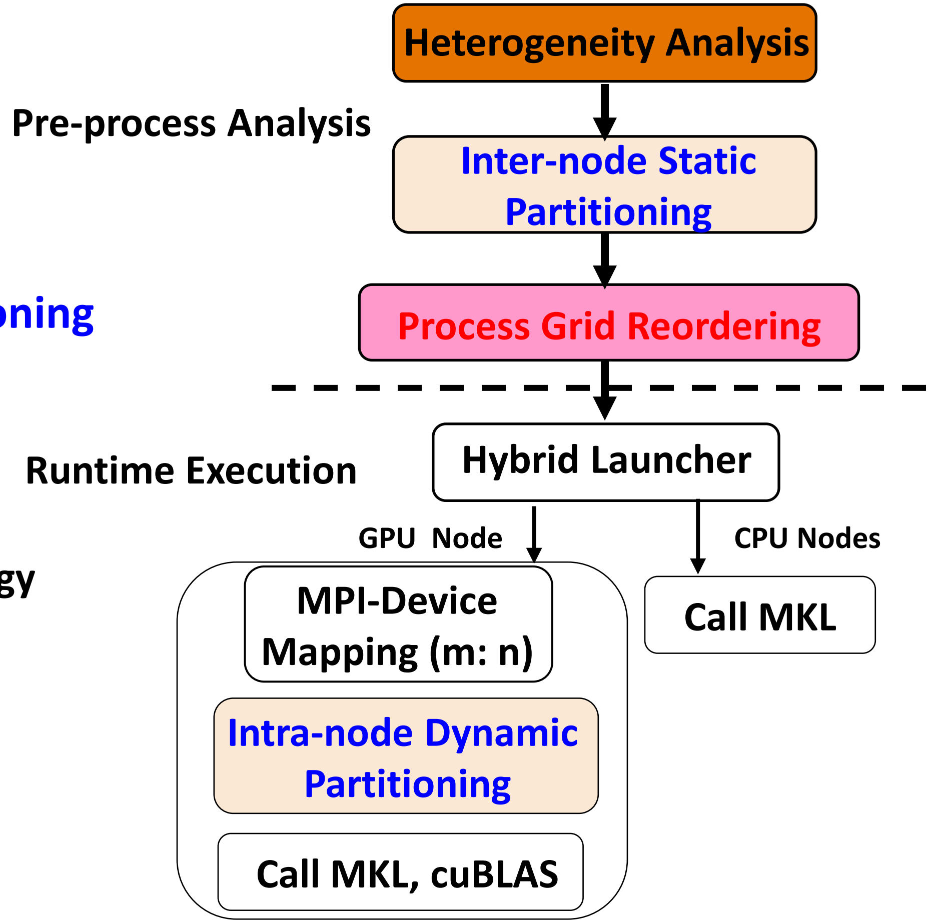
G1 → G1 → C1
G2 → G2 → C2

- **Unbalanced Workload**
  G1 might get more blocks than G2
  C1 might get more blocks than C2

- **Balanced Workload**
  All the parties have the adequate workload

Overview of Hybrid HPL Design

- **Heterogeneity Analysis**
  Pure CPU nodes
  Pure GPU nodes
  Hybrid CPU+GPU nodes
- **Two-level Workload Partitioning**
  Inter-node Static
  Intra-node Dynamic
- **Process Grid Reordering**
  Generate efficient node topology
- **Hybrid Launcher**
  GPU nodes
  Asynchronous Memory Copy
  MPI-Device Mapping
  Adaptive Split Ratio Tuning
  CPU nodes



Outline

- Introduction and Motivation
- Proposed Design for Hybrid HPL
- Performance Evaluation
- Conclusion and Future Work

Experimental Setup

- Experiment Environment

Specifications	Cluster A	Oakley Cluster
CPU Processor Type	Intel Xeon E5630	Intel Xeon X5650
CPU Clock	2.53GHz	2.66GHz
Node Type	two quad-core sockets	two 6-core sockets
CPU Memory	11.6 GB	46 GB
CPU Theo.peak (double)	80.96 Gflops	127.68 Gflops
GPU Processor Type	NVIDIA Tesla C2050	NVIDIA Tesla M2070
GPU Theo.peak (double)	515 Gflops/GPU	515 Gflops/GPU
BLAS Lib	MKL 10.3/cuBLAS	MKL 10.3/cuBLAS
Compilers	Intel Compilers 11.1	Intel Compiler 11.1
MPI Lib	MVAPICH2 1.9	MVAPICH2 1.9
OS	RHEL 6.1	RHEL 6.3
Interconnect	Mellanox IB QDR	Mellanox IB QDR

- MPI Library: MVAPICH2
  - High Performance open-source MPI Library for InfiniBand, 10Gig/iWARP, and RDMA over Converged Enhanced Ethernet (RoCE)
  - Support for heterogeneous GPU Clusters
  - Used by more than 2,150 organizations (HPC Centers, Industry and Universities) in 72 countries

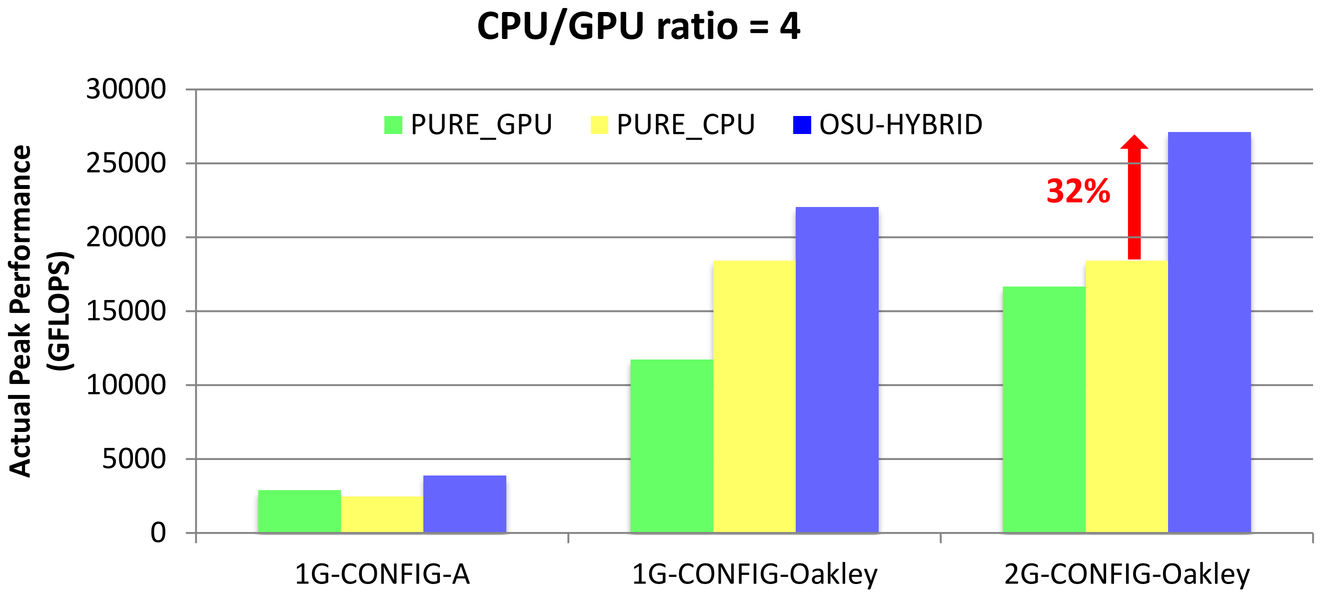
  http://mvapich.cse.ohio-state.edu/

Actual Peak Performance

1G-CONFIG-A: 8 GPU nodes (1 GPU accelerators) + 32 CPU nodes

1G-CONFIG-Oakley: 32 GPU nodes (1 GPU accelerators) + 128 CPU nodes

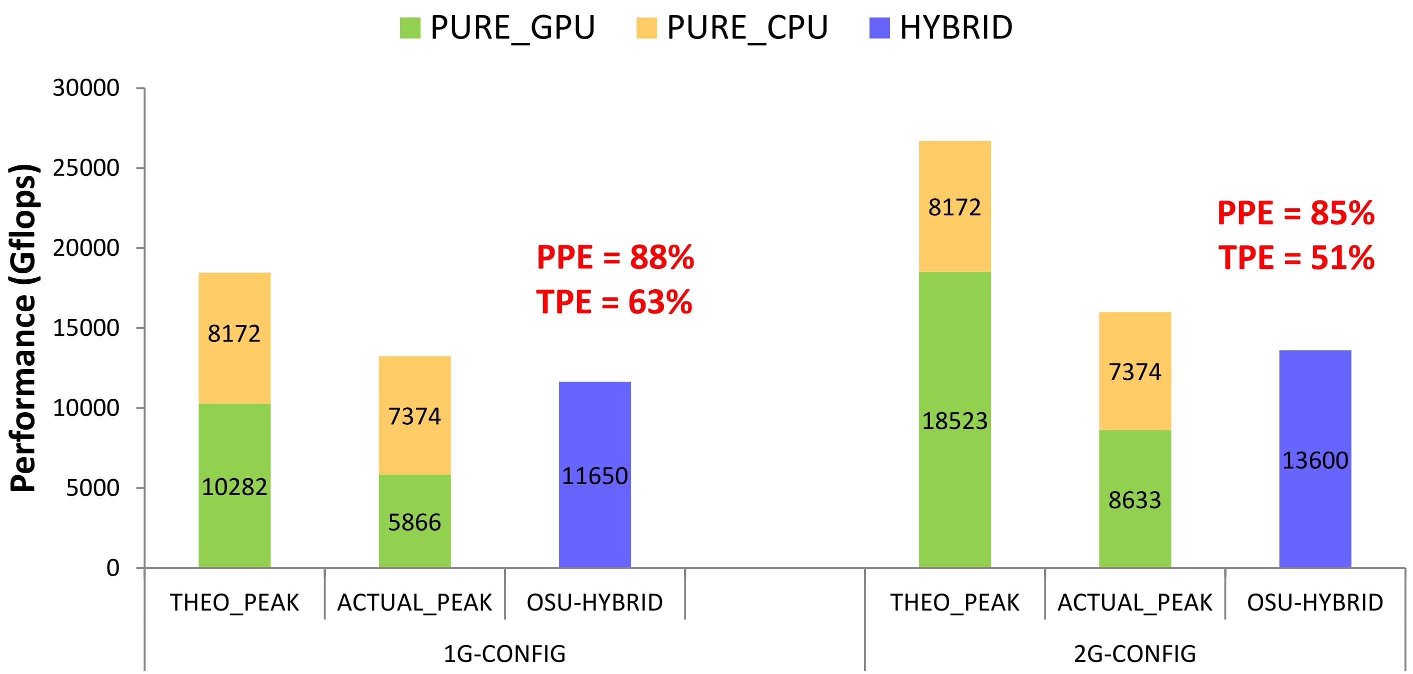
2G-CONFIG-Oakley: 32 GPU nodes (2 GPU accelerators) + 128 CPU nodes



- Hybrid designs outperforms the PURE_CPU by 32%

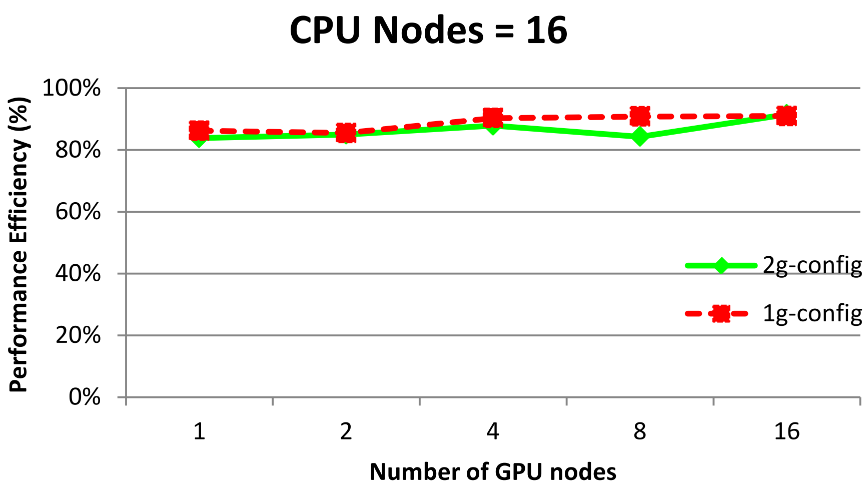
R. Shi, S. Potluri, K. Hamidouche, X. Lu, K. Tomko and D. K. Panda, A Scalable and Portable Approach to Accelerate Hybrid HPL on Heterogeneous CPU-GPU Clusters, IEEE Cluster (Cluster '13), Best Student Paper Award

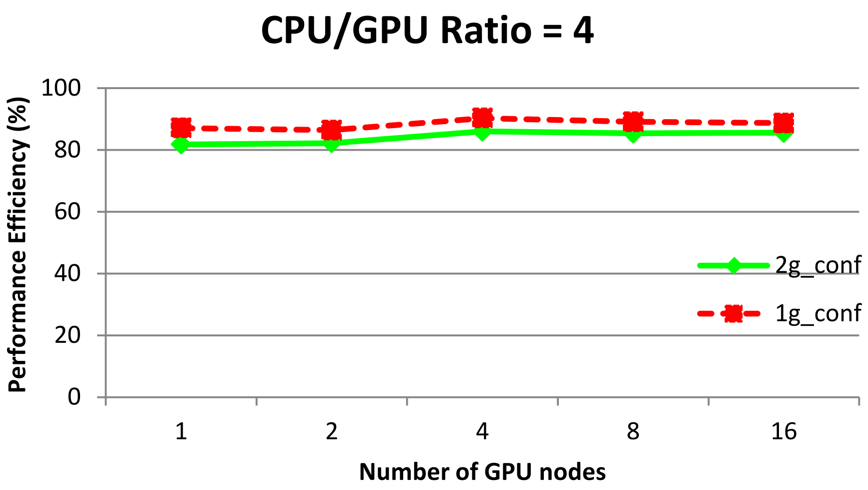
Efficiency

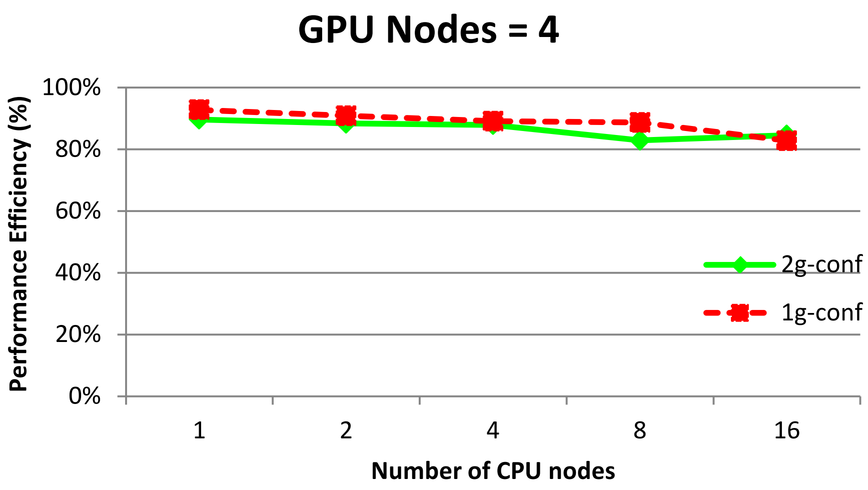


- PPE: Peak Performance Efficiency
- TPE: Theoretical Performance Efficiency

Peak Performance Efficiency (PPE) Scalability







- **Constant PPE for fixed CPUs, fixed GPUs and fixed ratio**

Outline

- Introduction and Motivation
- Proposed Design for Hybrid HPL
- Performance Evaluation
- Conclusion and Future Work

Conclusion

- Propose a novel approach of HPL on GPU clusters with heterogeneity on intra- and inter-nodes configuration.
- Achieve **80%** of the combined actual peak performance of pure CPU and pure GPU nodes
- Studying the impact of such designs for other benchmarks and applications.

Thank You!

panda@cse.ohio-state.edu

Network-Based Computing Laboratory

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

MVAPICH Web Page

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