Partnership For Advanced Computing In Europe

1st Implementation Phase
Work Package 9

Giannos Stylianou
Research Assistant

The Cyprus Institute

CaSToRC

Activities Intro

- **SLURM Resource Manager**
  - Build & test on a small cluster
  - Configuration & installation on Prometheus cluster

- **High Performance Linpack**
  - Re-run benchmark
  - Update power efficiency results (upcoming deliverable)

SLURM Resource Manager (1/2)

- **Getting Familiar**
  - Build a testing cluster (2 compute nodes with GPU)
  - Install SLURM and run tests
  - Observe the way resources are managed

- **Working on Prometheus (specs later)**
  - Replace Torque & Maui with SLURM
  - Configuration file according to PROMETHEUS specs
  - GRES configuration file for GR scheduling (GPUs)
  - PAM configuration file to prevent locked memory limit propagation

SLURM Resource Manager (2/2)

- **More Configuration**
  - MUNGE authentication service is required
  - Users constraint: necessarily use --gres parameter to assign Generic Resources to jobs
  - Restrict SSH connections to compute nodes
  - Achieve resource management and accounting using MySQL database
  - Set SLURM as a module which is loaded/unloaded on usage
  - Run SLURM tests suite and check their correctness
  - Re-run HPL jobs with SLURM to update Power Efficiency Results

  - Issue: Investigating ways to avoid users from logging into unassigned nodes, but allow SSH to assigned ones (PAM module)

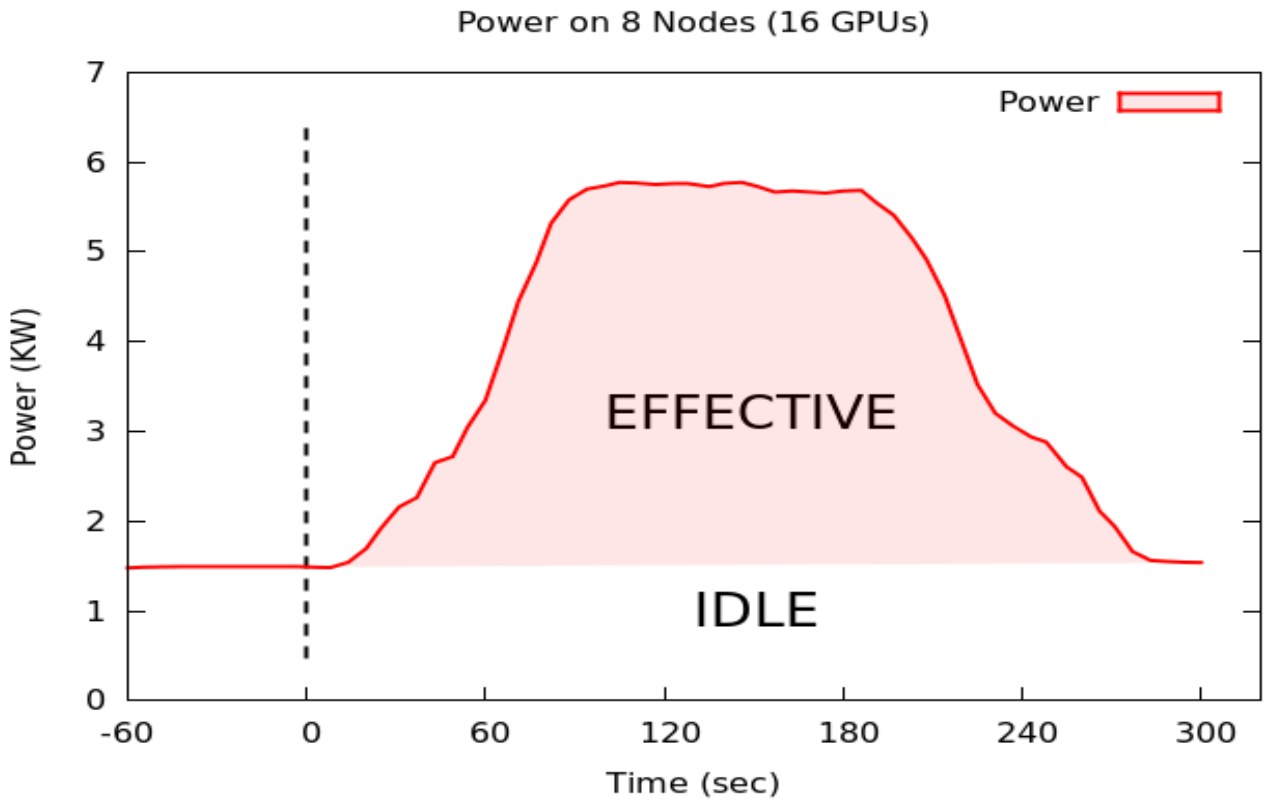
Prometheus Cluster Specifications

- **Cluster of 8x gpu-compute nodes**
- **Model:** IBM dx360 M3
- **CPU:** 2x Xeon X5650 @ 2.67GHz per node
- **GPU:** 2x Nvidia M2070 (Tesla) per node
- **RAM:** 24GB DDR3 @1333MHz per node
- **Disk:** 146GB
- **Interconnection:** Mellanox 4xQDR Infiniband (40Gbps)
- **Storage:** 1TB disk on Management node
- (purchased under PRACE)

HPL Execution

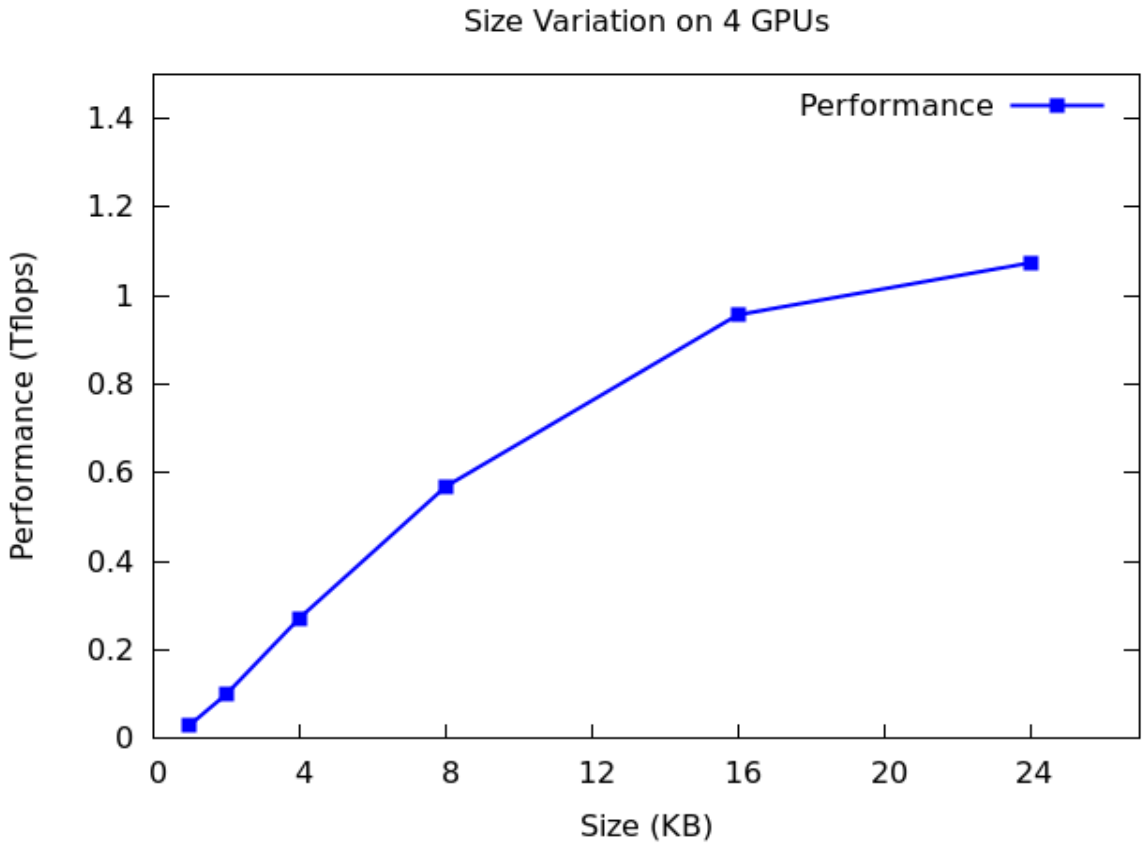
- **Experimental Setup**
  - Use SLURM to run HPL jobs
  - Run testing scenarios to find the optimal local size (N) and block size (NB)
  - Use **xCat rvitals** tool to monitor wattage on each node every 5 seconds
  - 60” sleep precedes every HPL run
  - Write power details in a monitoring file with timestamps
  - Write HPL results in an output file with timestamps

HPL Results – Power



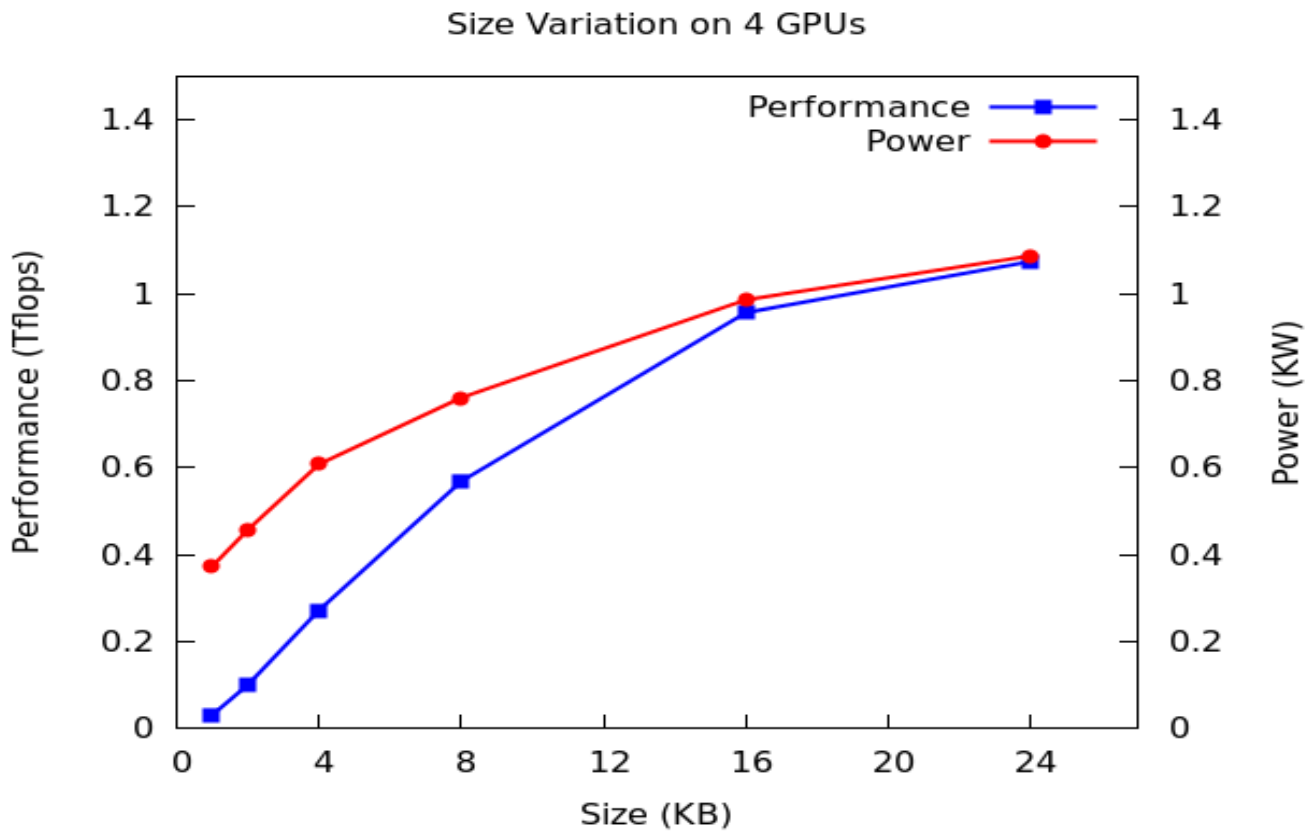
- All cluster resources fully used
- Similar chart for every scenario

HPL Results – Size Variation (1/3)



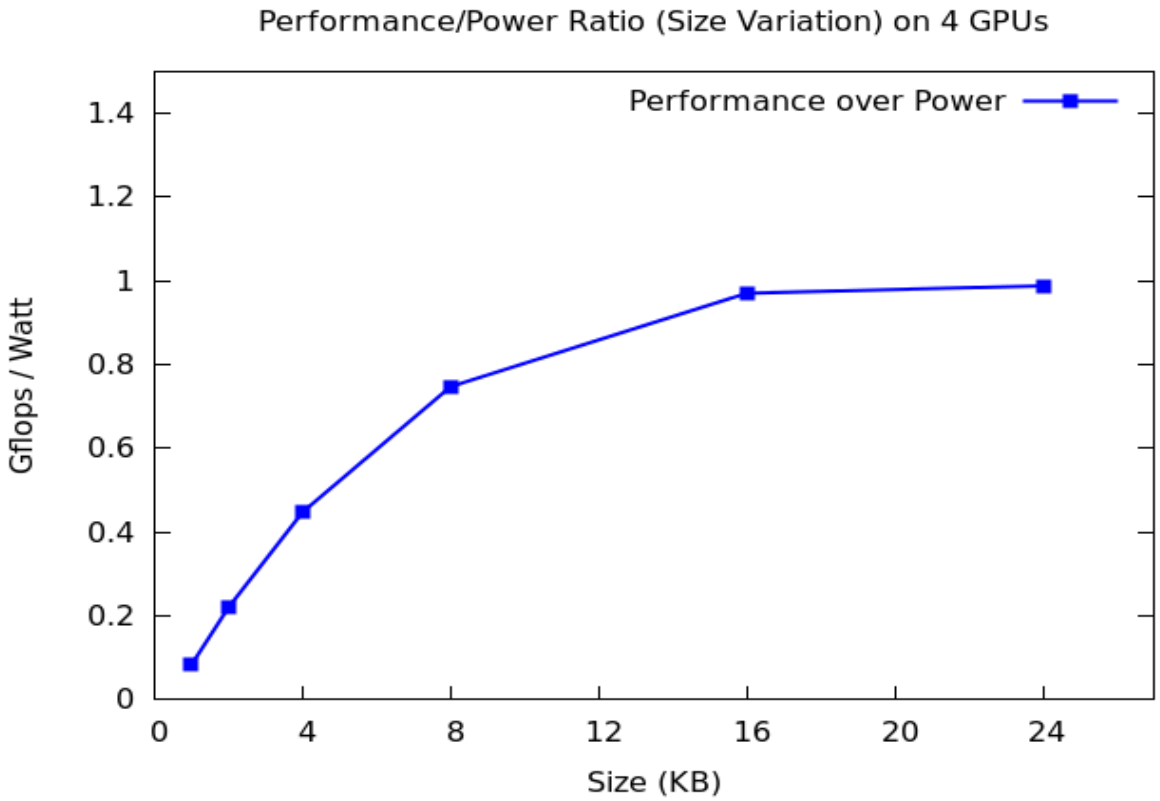
- **Selecting the optimal size N=24K**

HPL Results – Size Variation (2/3)



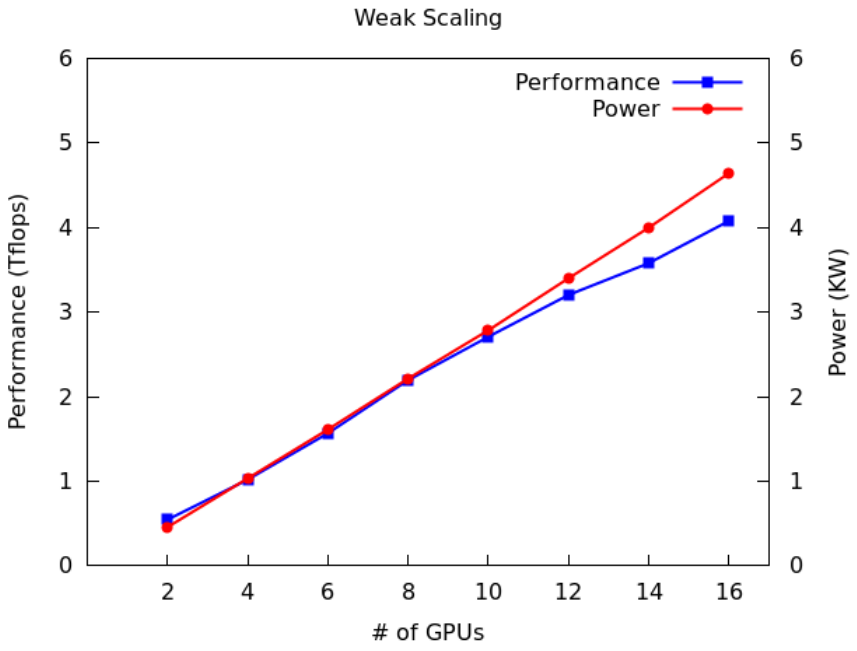
- **Selecting the optimal size N=24KB**

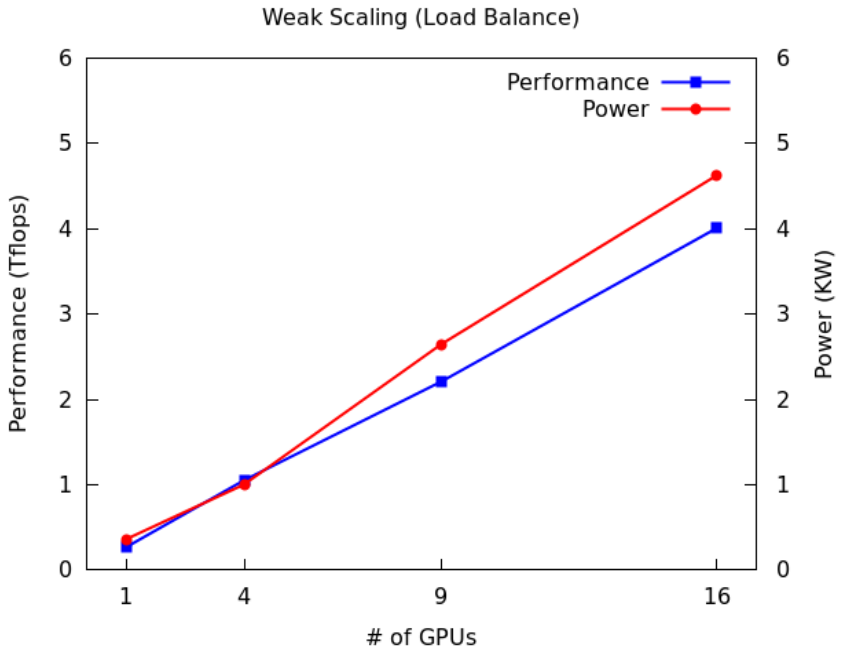
HPL Results – Size Variation (3/3)



- **Efficiency: optimal flops/watt**
- **Stable to 1 Gflops/Watt**

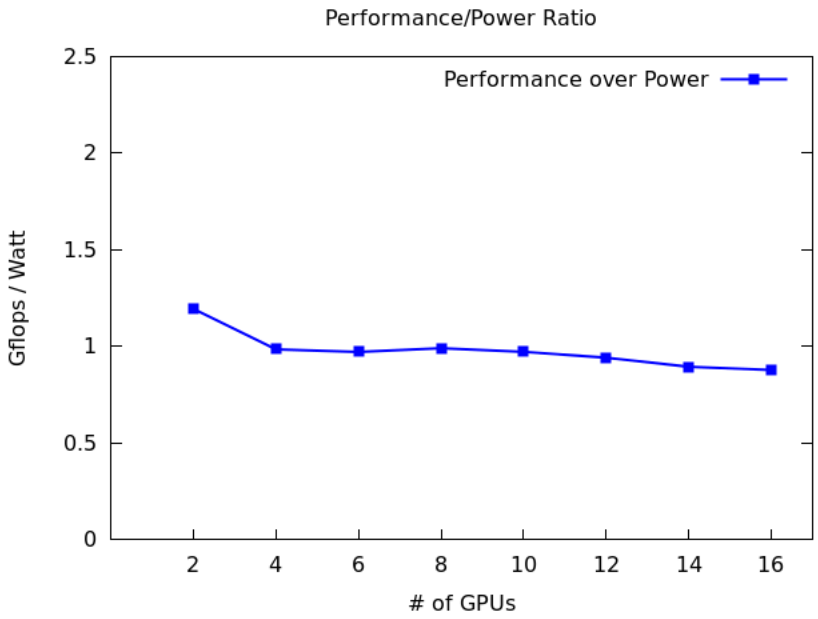
HPL Results – Weak Scaling

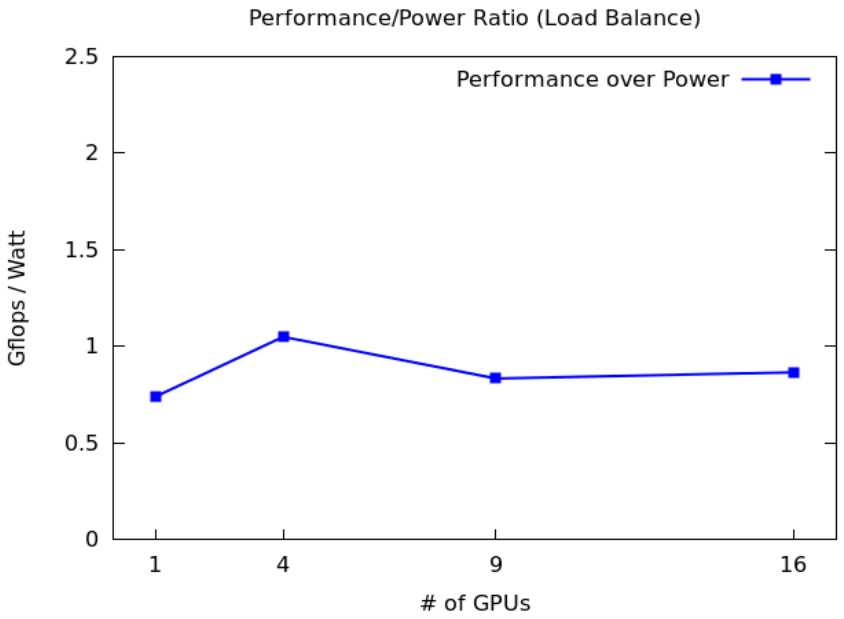




- Linear scaling of power and performance
- Try a load balancing scenario to see any difference

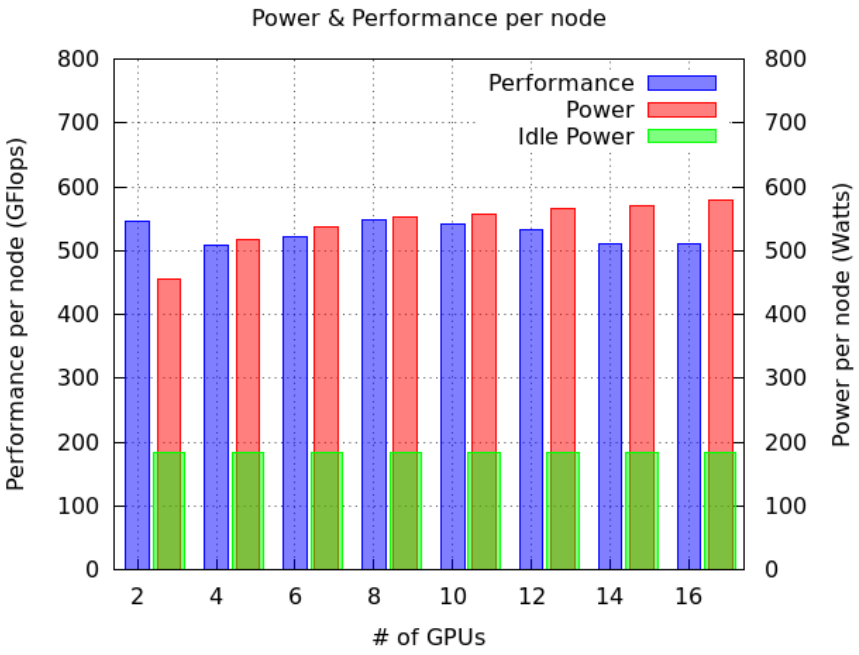
HPL Results - Ratio

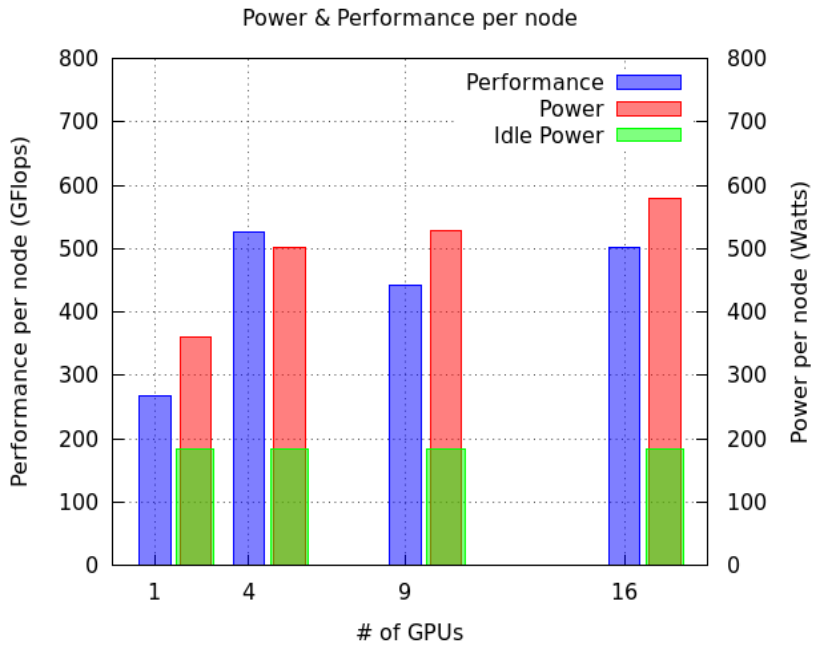




- Effectively constant to 1 Gflops/Watt
- Small, not significant, decrease while increasing #GPUs
- Idle power is included

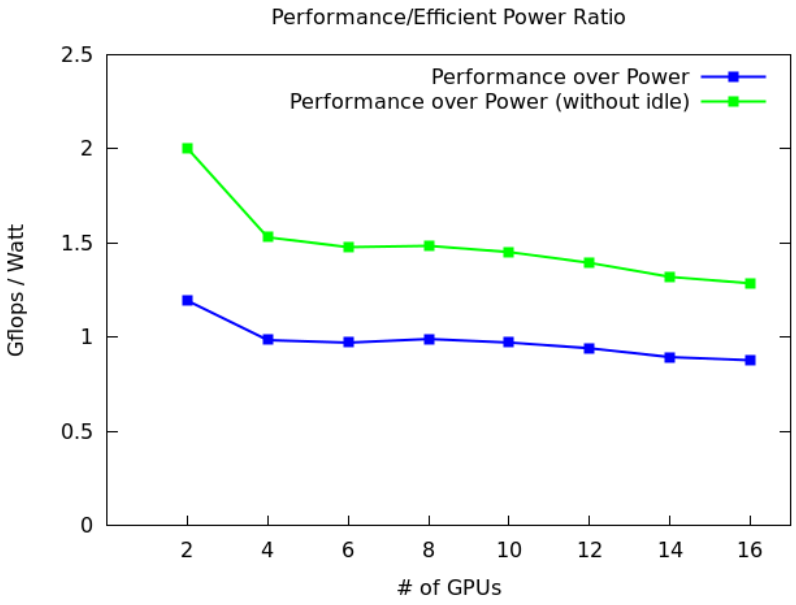
HPL Results – Histograms

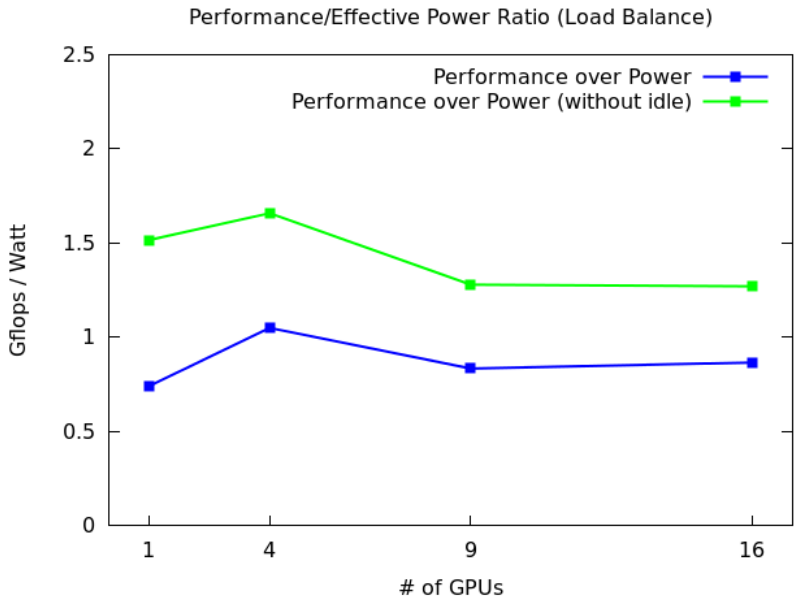




- Power consumed in node idle state: 184 Watts
- Total power consumed per node: 500 Watts
- Max performance per node: 500 GFlops

HPL Results – Ratio No2





- Effective Power Efficiency
- Idle power is not included
- Close to 1.5 Gflops/Watt

Summary

- SLURM is now on steady state - main scheduler
- Good HPL weak scaling
- Updated HPL Results for the upcoming deliverable
- HPL Data can be prepared and delivered with the right format
- Able to also re-run other benchmarks for the deliverables and collect better results

Thank you

www.cyi.ac.cy/castorc

CaSToRC