HPCG
Performance Benchmark and Profiling

April 2017

Note

- **The following research was performed under the HPC Advisory Council activities**
  - Compute resource - HPC Advisory Council Cluster Center

- **The following was done to provide best practices**
  - HPCG performance overview
  - Understanding HPCG communication patterns
  - Ways to increase HPCG productivity

- **For more info please refer to**
  - https://gitlab.com/edbennett/HPCG

HPCG

- **HPCG Benchmark project**
  - An effort to create a more relevant metric for ranking HPC systems
  - Potential replacement for the High Performance LINPACK (HPL) benchmark
  - Currently HPL is used by the TOP500 benchmark

- **HPCG**
  - High Performance Conjugate Gradient
  - Stand-alone code that measures the performance of basic operations
    - Sparse matrix-vector multiplication
    - Sparse triangular solve
    - Vector updates
    - Global dot products
    - Local symmetric Gauss-Seidel smoother
  - Driven by multigrid preconditioned CG algorithm that exercises the key kernels on a nested set of coarse grids
    - Reference implementation is written in C++ with MPI and OpenMP support

Objectives

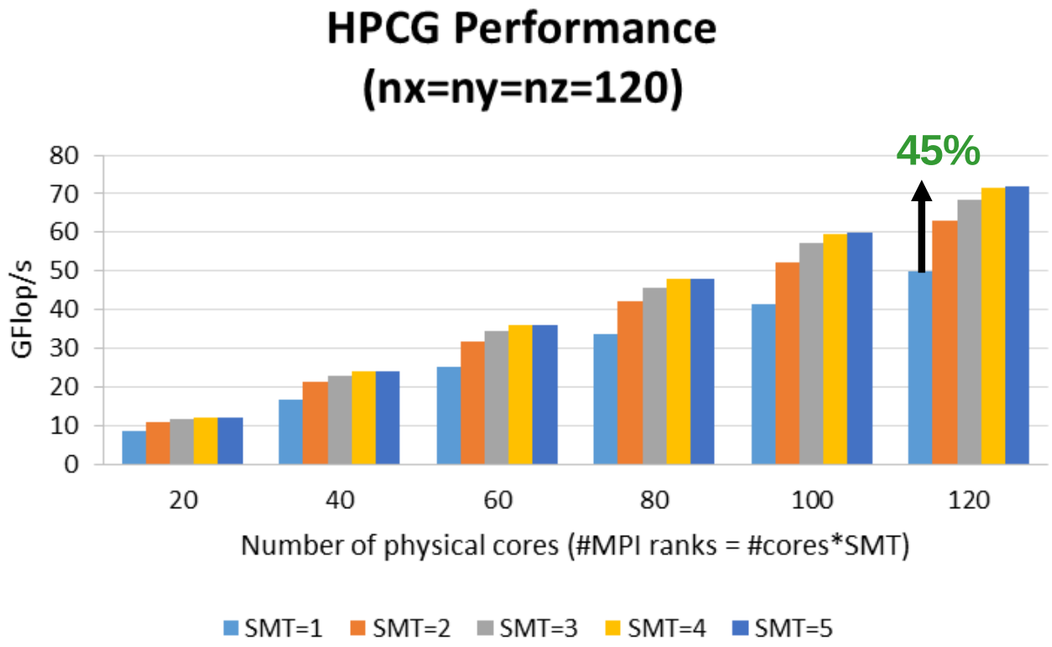
- **The presented research was done to provide best practices**
  - HPCG performance benchmarking
    - Hardware comparison
    - Optimization tuning
- **The presented results will demonstrate**
  - The scalability of the compute environment/application
  - Considerations for higher productivity and efficiency

Test Cluster Configuration

- **IBM OperPOWER 8-node “Telesto” cluster**
- **IBM Power System S822LC (8335-GTA)**
  - IBM: Dual-Socket 10-Core @ 3.491 GHz CPUs, Memory: 256GB memory, DDR3 PC3-14900 MHz
- **Wistron OpenPOWER servers**
  - Wistron: Dual-Socket 8-Core @ 3.867 GHz CPUs. Memory: 224GB memory, DDR3 PC3-14900 MHz
- **OS: RHEL 7.2, MLNX_OFED_LINUX-3.4-1.0.0.0 InfiniBand SW stack**
- **Mellanox ConnectX-4 EDR 100Gb/s InfiniBand Adapters**
- **Mellanox Switch-IB SB7800 36-port EDR 100Gb/s InfiniBand Switch**
- **Compilers: GNU compilers 4.8.5, IBM XL Compilers 13.1.3**
- **MPI: Open MPI 2.0.2, IBM Spectrum MPI 10.1.0.2**
- **Application: HPCG Version 1.0**
- **MPI Profiler: IPM**

HPCG Performance – SMT

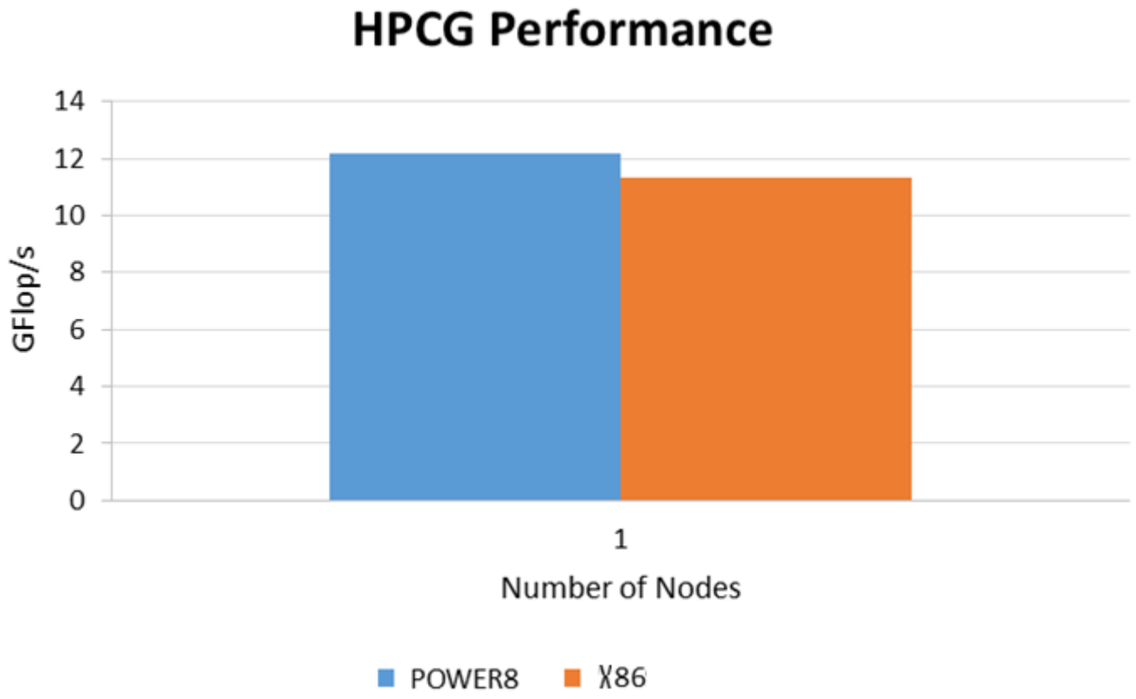
- **Simultaneous Multithreading (SMT) allows additional hardware threads for compute**
- **Additional performance gain is seen with SMT enabled**
  - Up to 45% of performance gain is seen between no SMT versus 5 SMT threads are used
  - As more MPI ranks being used for SMT cores, but would need more memory to run
  - Memory bandwidth saturation appears to be around at around 5 SMT thread



Higher is better

HPCG Performance – System Architecture

- **Power CPU demonstrates 8% higher performance compared to x86**
  - Performance gain on a single node is approximately 8% for Power8
  - 32 cores per node used for x86, versus 20 physical cores used per node for Power

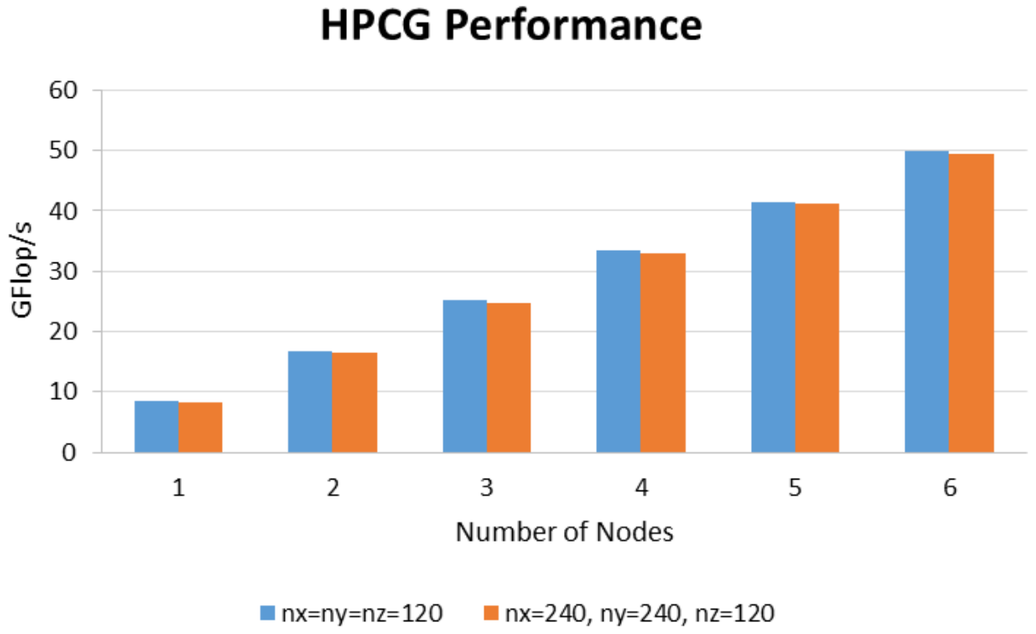


Higher is better

SMT=5 for IBM POWER8

HPCG Performance – Matrix Size

- **The sparse matrix size specified determines the amount of memory consumed**
  - The amount of memory for sparse matrix computation is bounded by matrix size specified
  - Performance achieved by using slightly lower matrix size appeared to have no effect
- **Shorter time duration appeared to have no effect on the performance**
  - The standard runtime for HPCG is 30 minutes; running shorter appears to perform the same

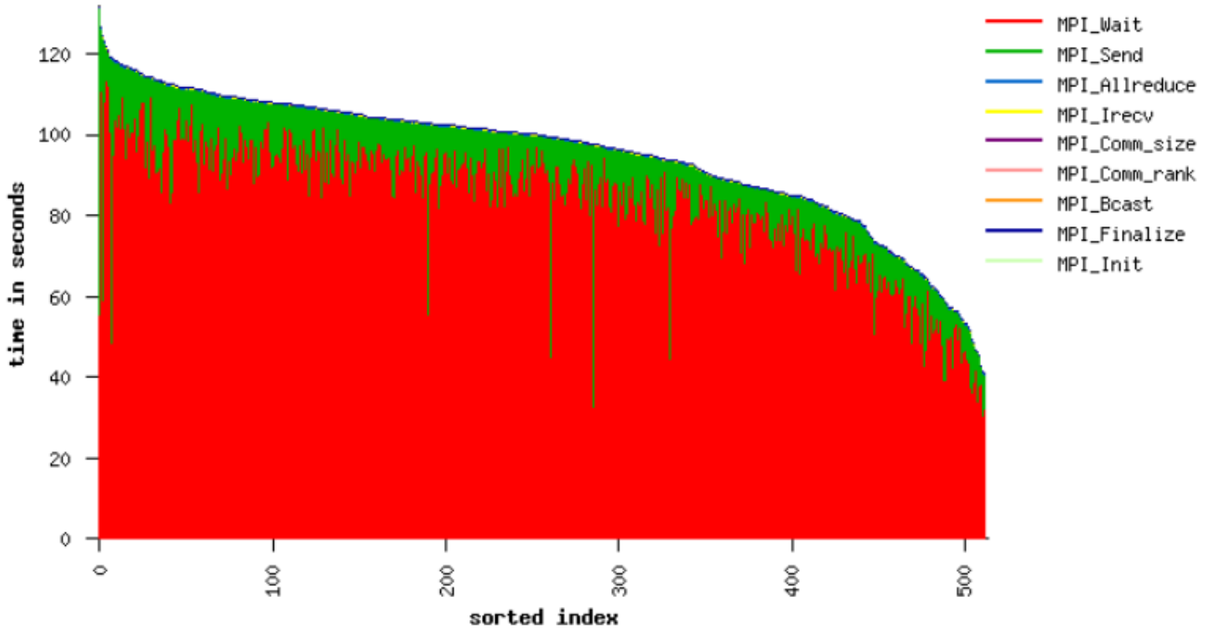


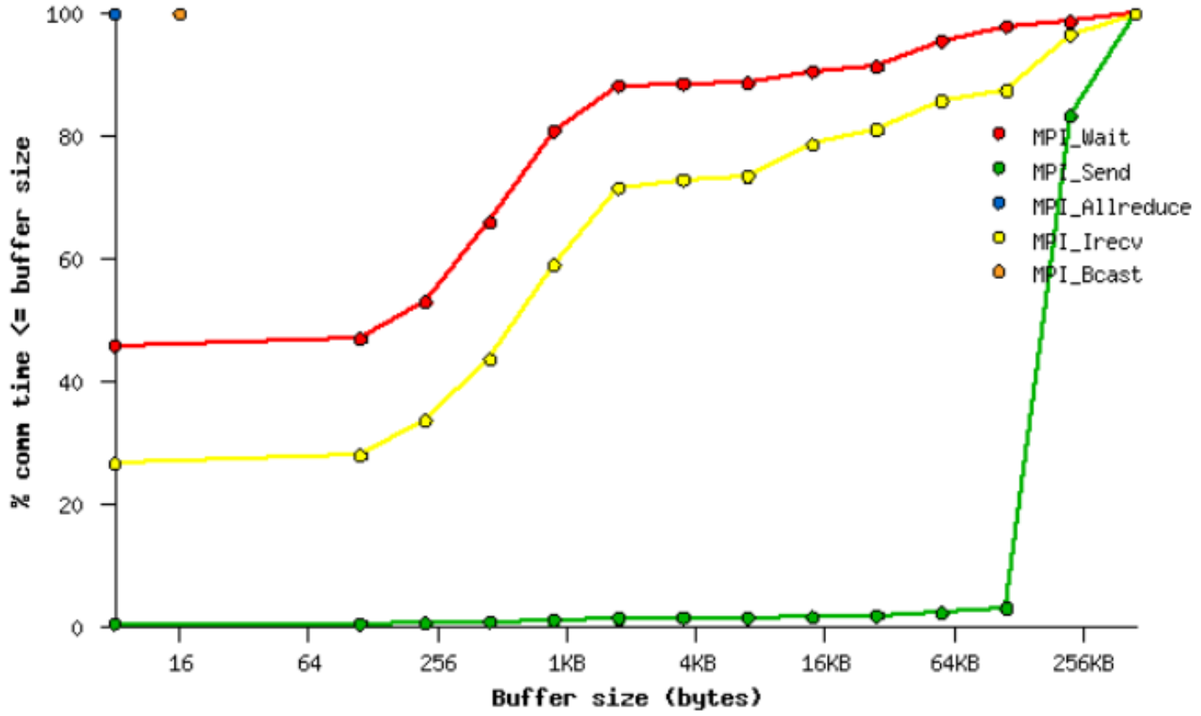
Higher is better

SMT=1 for IBM POWER8

HPCG Profiling – % of MPI Calls

- **For the most time consuming MPI calls (as % of wall time):**
  - MPI_Wait (1.5%), MPI_Send (%), MPI_Waitall (0.5%)
- **The percentage time spent on communication is limited**





HPCG Summary

- **HPCG Project**
  - Potential replacement for the High Performance LINPACK (HPL) benchmark
- **Simultaneous Multithreading (SMT) provides additional benefit for compute**
  - Up to 45% of performance gain is seen between no SMT versus 5 SMT threads are used
- **Power CPU showcase 8% higher performance vs x86**
  - 32 cores per node used for x86, versus 20 cores used per node for Power
- **The sparse matrix size specified determines the amount of memory consumed**
  - The amount of memory for sparse matrix computation is bounded by matrix size specified
  - Performance achieved by using slightly lower matrix size appeared to have no effect
  - Shorter time duration appeared to have no effect on the performance
  - The standard runtime for HPCG is 30 minutes; running shorter appears to perform the same
- **The percentage time spent on communication is limited**
  - For the most time consuming MPI calls (as % of wall time):
  - MPI_Wait (1.5%), MPI_Send (%), MPI_Waitall (0.5%)

Thank You