HPCG Performance study with Intel KNL

By Ashish Kumar Singh. January 2017 (HPC Innovation Lab)

This blog presents an in-depth analysis of the High Performance Conjugate Gradient (HPCG) benchmark on the Intel Xeon Phi processor, which is based on Intel Xeon Phi architecture codenamed “Knights Landing”. The analysis has been performed on PowerEdge C6320p platform with the new Intel Xeon Phi 7230 processor.

Introduction to HPCG and Intel Xeon Phi 7230 processor

The **HPCG** benchmark constructs a logically global, physically distributed sparse linear system using a 27-point stencil at each grid point in 3D domain such that the equation at the point (I, j, k) depend on its values and 26 surrounding neighbors. The global domain computed by benchmark is (NRx * Nx) X (NRy*Ny) X (NRz*Nz), where Nx, Ny and Nz are dimensions of local subgrids, assigned to each MPI process and number of MPI ranks are NR = (NRx X NRy X NRz). These values can be defined in hpcg.dat file or passed in the command line arguments.

The HPCG benchmark is based on conjugate gradient solver, where the pre-conditioner is a three-level hierarchical **multi-grid (MG) method with Gauss-Seidel**. The algorithm starts with MG and contains Symmetric Gauss-Seidel (SymGS) and Sparse Matrix-vector multiplication (SPMV) routines for each level. Both SYMGS and SPMV require data from their neighbor as data is distributed across nodes which is provided by their predecessor, the Exchange Halos routine. The residual should be lower than 1^{-6} which is locally computed by Dot Product (DDOT), while MPI_Allreduce follows the DDOT and completes the global operation. WAXPBY only updates a vector with sum of two scaled vectors. Scaled vector addition is a simple operation that calculates the output vector by scaling the input vectors with a constant and performing an addition on the values of the same index. So, HPCG has four computational blocks **SPMV, SymGS, WAXPBY and DDOT**, while two communication blocks **MPI_Allreduce** and **Halos Exchange**.

Intel Xeon Phi Processor is a new generation of processors from the Intel Xeon Phi family. Previous generations of Intel Xeon Phi were available as a coprocessor, in a PCI card form factor and required an Intel Xeon processor. The Intel Xeon Phi 7230 contains 64 cores @ 1.3GHz of core frequency along

with the turbo speed of 1.5GHz and 32MB of L2 cache. It supports DDR4-2400MHz memory up to 384GB and instruction set of AVX512. Intel Xeon Phi processor also encloses 16GB of MCDRAM memory on socket **with a sustained memory bandwidth of up to ~480GB/s measured by the Stream benchmark. Intel Xeon Phi 7230 delivers up to ~1.8TFLOPS of double precision HPL performance.**

This blog showcases the performance of HPCG benchmark on the Intel KNL processor and compares the performance to that on the Intel Broadwell E5-2697 v4 processor. The Intel Xeon Phi cluster comprises of one head node which is PowerEdge R630 and 12 PowerEdge C6320p as compute nodes. While Intel Xeon processor cluster includes one PowerEdge R720 as head node and 12 PowerEdge R630 as compute nodes. All compute nodes are connected by Intel Omni-Path of 100GB/s. The cluster shares the storage of head node over NFS. The detailed information of the clusters are mentioned below in table1. All HPCG tests on Intel Xeon Phi has been performed with the BIOS settings of "quadrant" cluster mode and "Memory" memory mode.

Table1: Cluster Hardware and software details

Testbed configuration

Platform	PowerEdge C6320p	PowerEdge R630
Processor	1 x Intel Xeon Phi 7230	2 x Intel E5-2697 v4@2.3Ghz
Memory	96GB DDR4 @ 2400MHz	128GB DDR4@2400MHz
Nodes Interconnects	Intel Omni-Path 100GB/s	Intel Omni-Path 100GB/s
	Software and Firmware	
Operating System	RHEL 7.2 x86_64	RHEL 7.2 x86_64
Kernel	3.10.0-327.36.3.el7.x86_64	3.10.0-327.el7.x86_64
BIOS	Version 1.0.0	Version 2.2.5
XPPSL	XPPSL-1.4.2	---
Intel Compiler	Version 2017.0.098	Version 2017.0.098
	Benchmark	
HPCG	Intel HPCG 3.0	Intel HPCG 3.0

HPCG Performance analysis with Intel KNL

Choosing the right problem size for HPCG should follow the following rules. **The problem size should be large enough not to fit in the cache of the device. The problem size should be able to occupy the significant fraction of main memory, at least 1/4th of total**. For HPCG performance characterization, we have chosen the local domain dimension of 128^3, 160^3, and 192^3 with the

execution time of t=30 seconds. The local domain dimension defines the global domain dimension by (NR*Nx) x (NR*Ny) x (NR*Nz), where Nx=Ny=Nz=160 and NR is the number of MPI processes.

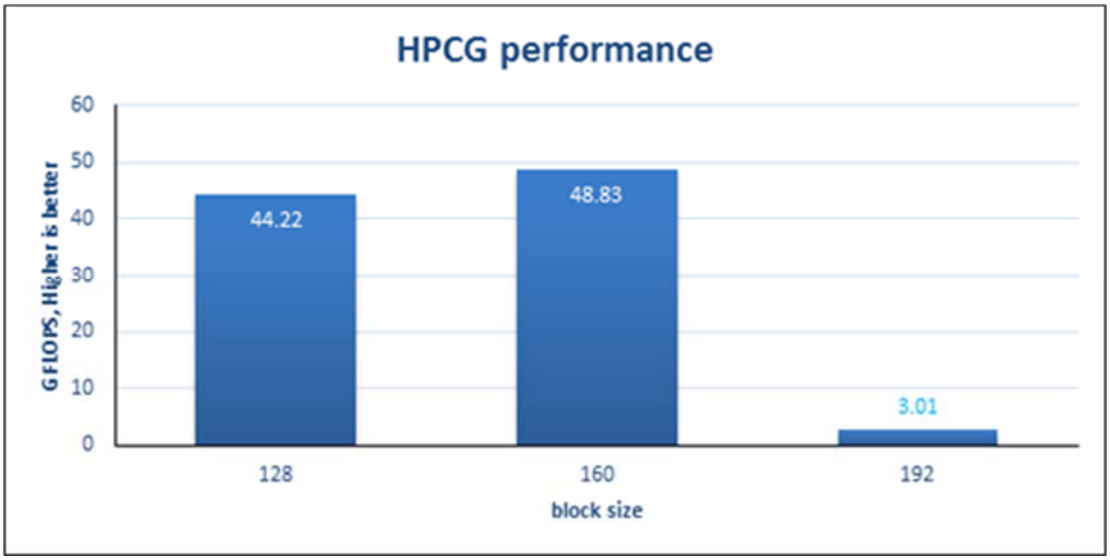

Figure 1: HPCG Performance comparison with multiple local dimension grid size

As shown in figure 1, the local dimension grid size of 160^3 gives the best performance of 48.83GFLOPS. The problem size bigger than 128^3 allows for more parallelism and it fits well inside the MCDRAM while 192^3 does not. All these tests have been carried out with 4 MPI processes and 32 OpenMP threads per MPI process on a single Intel KNL server.

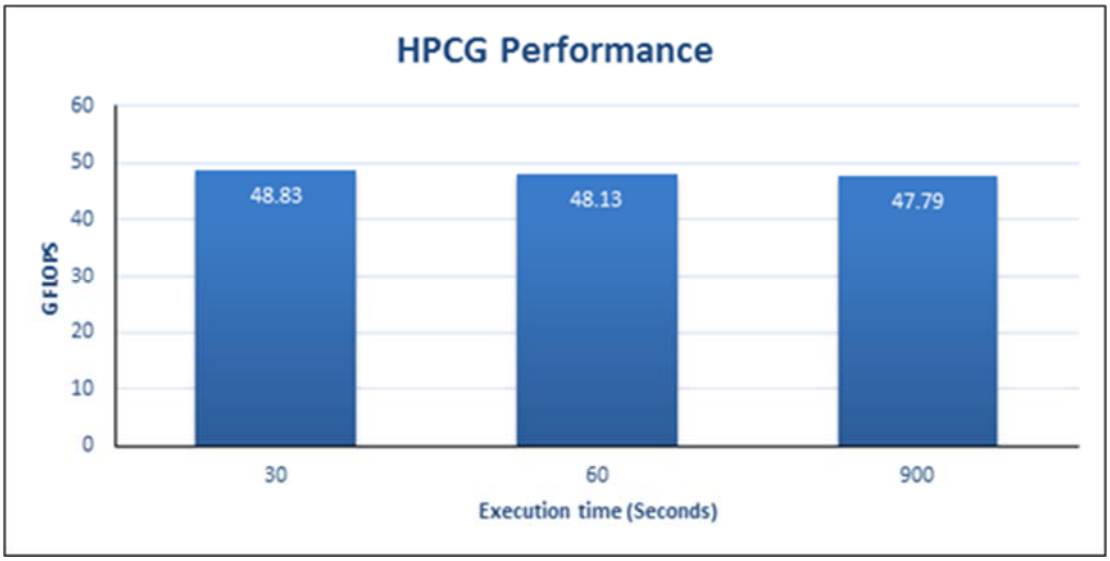

Figure 2: HPCG performance comparison with multiple execution time.

Figure 2 demonstrates HPCG performance with multiple execution times for grid size of 160^3 on a single Intel KNL server. As per the graph, HPCG performance doesn't change even by changing the execution time. It means execution time does not appear to be a factor for HPCG performance. So, we may not need to spend hours or days of time to benchmark large clusters, which in result, will save both time and power. Although, the **official execution time should be >=1800 seconds** as reported in the output file. If you decide to submit your results to TOP 500 ranking list, execution time should be not less than 1800seconds.

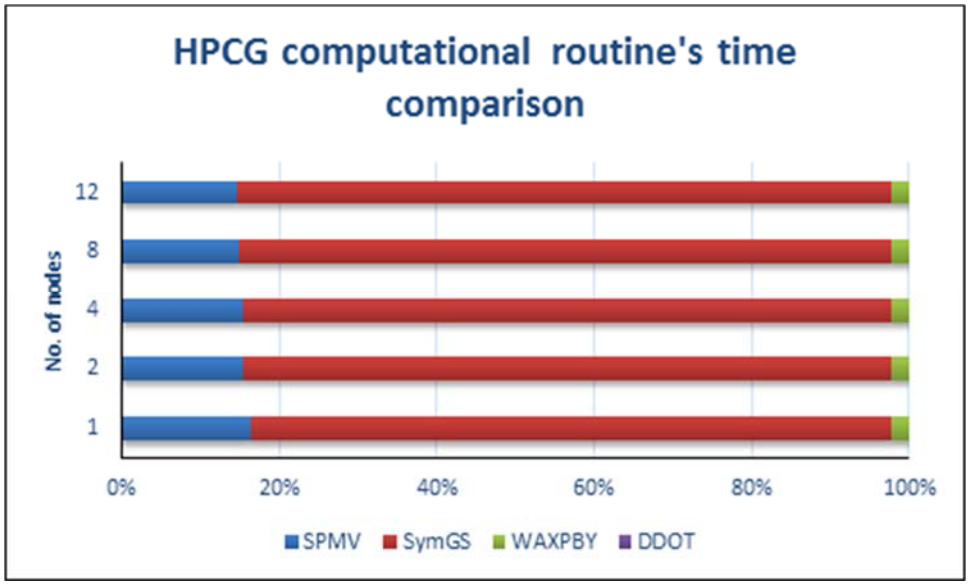

Figure 3: Time consumed by HPCG computational routines.

Figure 3 shows the time consumed by each computational routine from 1 to 12 KNL nodes. Time spent by each routine is mentioned in HPCG output file as shown in the figure 4. As per the above graph, HPCG spends its most of the time in the compute intensive pre-conditioning of SYMGS function and matrix vector multiplication of sparse matrix (SPMV). The vector update phase (WAXPBY) consumes very less time in comparison to SymGS and least time by residual calculation (DDOT) out of all four computation routines. As the local grid size is same across all multi-node runs, the time spent by all four compute kernels for each multi-node run are approximately same. The output file shown in figure 4, shows performance of all four computation routines. In which, MG consists both SymGS and SPMV.

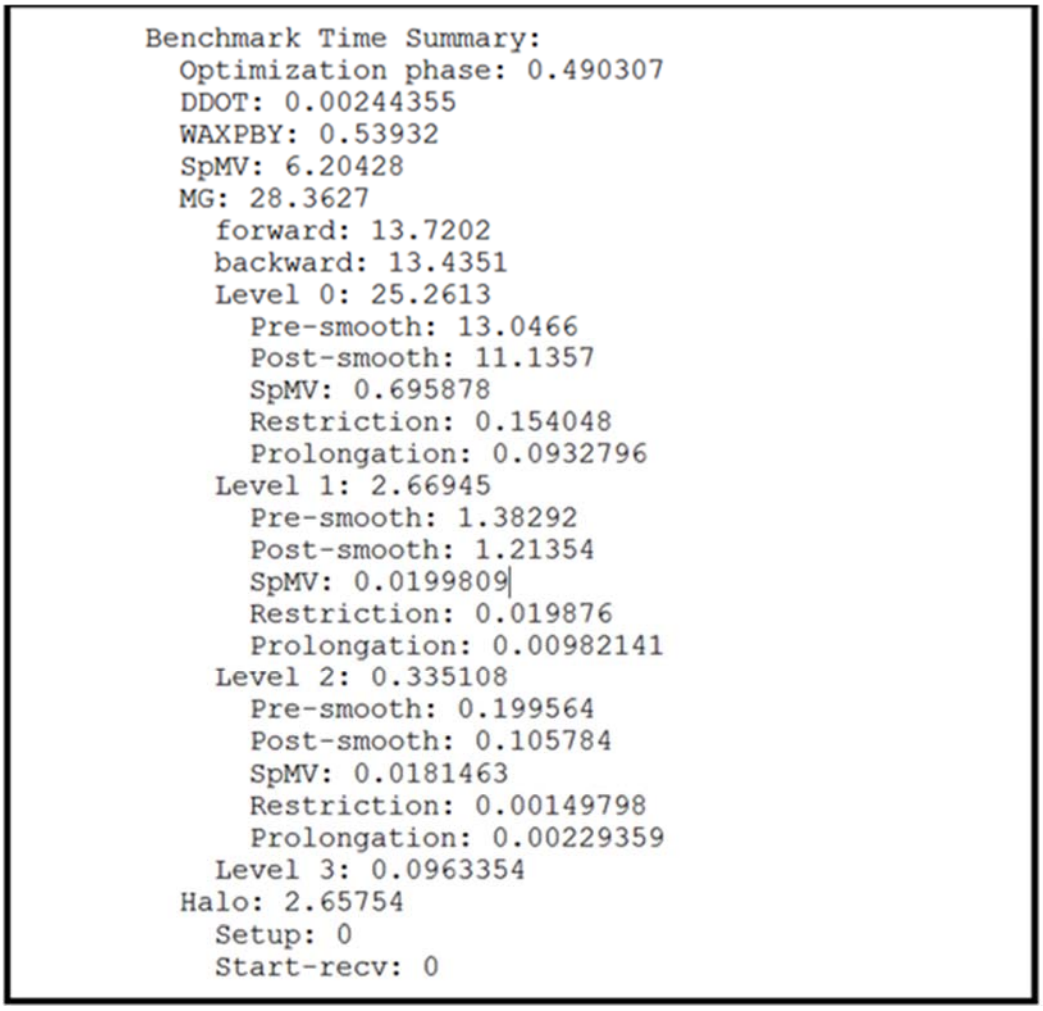

Figure 4: A slice of HPCG output file

Performance Comparison

Here is the HPCG multi-nodes performance comparison between Intel Xeon E5-2697 v4 @2.3GHz (Broadwell processor) and Intel KNL processor 7230 with Intel Omni-path interconnect.

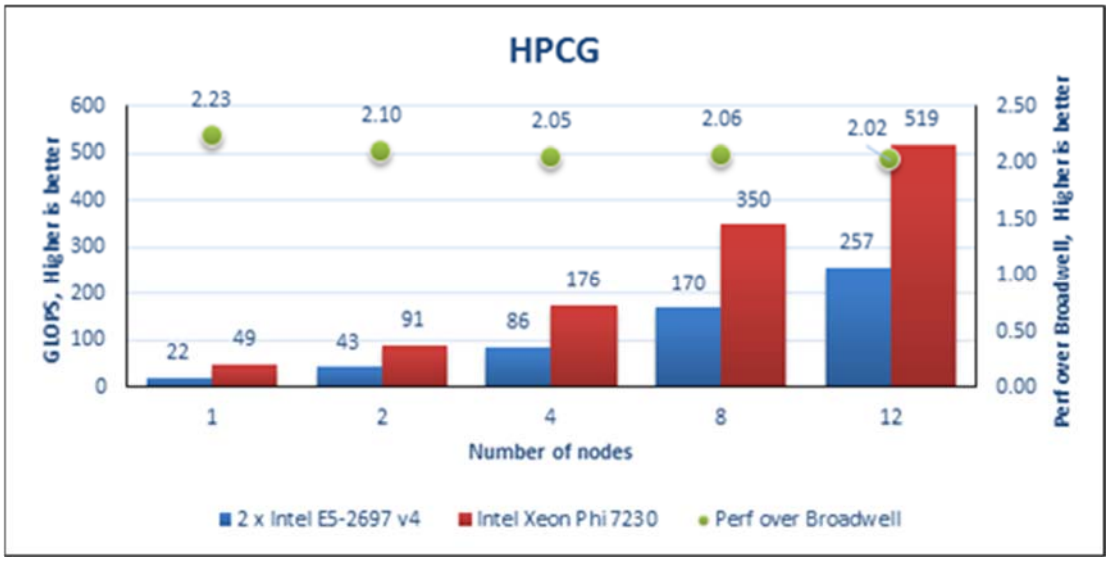

Figure 5: HPCG performance comparison between Intel Xeon Broadwell processor and Intel Xeon Phi processor

Figure 5 shows HPCG performance comparison between dual Intel Broadwell 18 cores processors and one Intel Xeon phi 64 cores processor. Dots in figure 5 show the performance acceleration of KNL servers over Broadwell dual socket servers. For single KNL node, HPCG performs 2.23X better than Intel Broadwell node. For Intel KNL multi-nodes also HPCG show more than 100% performance increase over Broadwell processor nodes. With 12 Intel KNL nodes, HPCG performance scales out well and shows performance up to ~520 GFLOPS.

Conclusion

Overall, HPCG shows ~2X higher performance with Intel KNL processor on PowerEdge C6320p over Intel Broadwell processor server. HPCG performance scales out well with more number of nodes. So, PowerEdge C6320p platform will be a prominent choice for HPC applications like HPCG.

Reference:

https://software.sandia.gov/hpcg/doc/HPCG-Specification.pdf

http://www.hpcg-benchmark.org/custom/index.html?lid=158&slid=281