KNL Performance Comparison: SENGA2

March 2017

1. Compilation, Setup and Input

Compilation

SENGA2 v2.1 is a Fortran/MPI Direct Numerical Simulation (DNS) code for simulating turbulent combustion with detailed chemistry (Prof. Stewart Cant, University of Cambridge). SENGA2 was compiled on both the ARCHER Xeon Phi (KNL) system and the ARCHER Xeon (Ivybridge) system. For both systems, the code was compiled using bash script *Allmake*. This script (and all dependent Makefiles) must be modified with the correct compiler and compiler flags before it is run. For access to the source code and compilation instructions please contact Neelofer Banglawala (EPCC).

The compiler versions and flags used were:

Compiler	ARCHER Xeon	ARCHER KNL
Intel Compiler	15.0.2.164	17.0.0.098
Compiler flags	-O2	-O2

Setup

- The ARCHER KNL nodes were used in "quad_100" configuration with all the MCDRAM configured as an additional cache level.
- In all cases, jobs were run on fully-populated nodes (at least as many MPI processes as there are physical cores per node) with Intel HyperThreading enabled per core.

Benchmark Case

The benchmark used was the default 1D laminar methane-flame test case that comes with the SENGA2 source code, but with the following parameter values (see files *com_senga2*.h and cont*.dat*):

- Global grid size (NXGLBL x NYGLBL x NZGLBL): 786432 x 1 x 1
- Halo size (NHALOX x NHALOY x NHALOZ) : 5 x 5 x 5
- Start step: 1
- Number of steps: 1
- Cold start switch 0

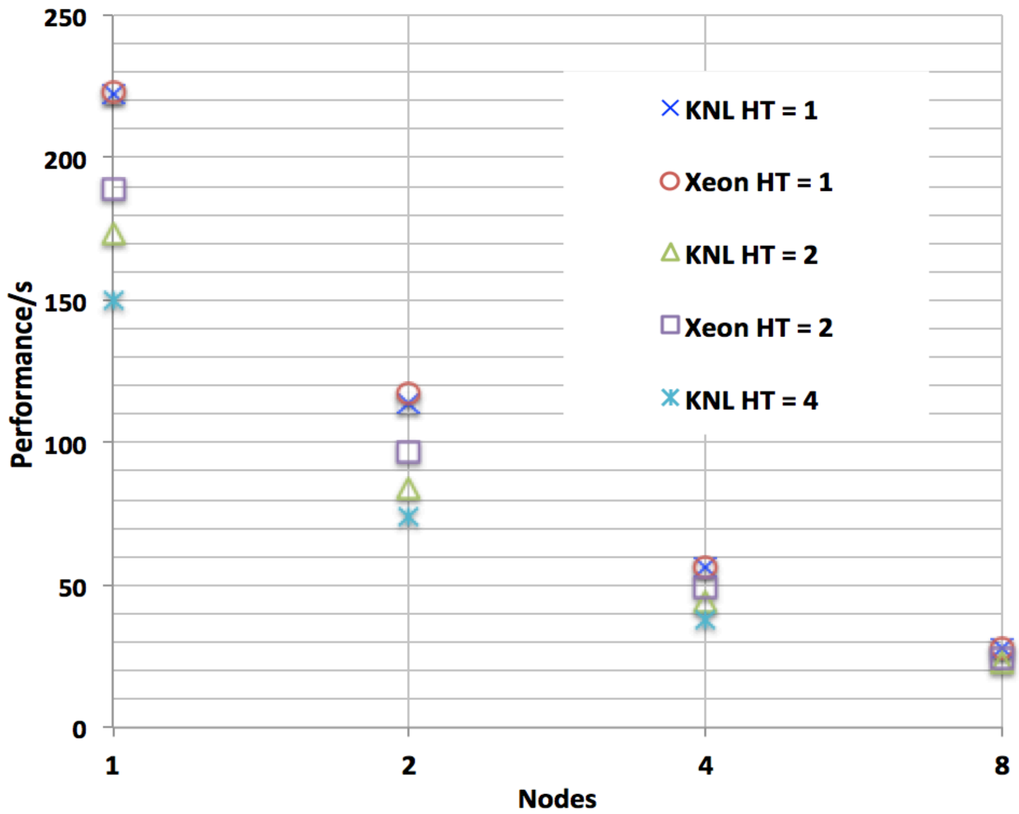
The following parameters were also varied as the number of MPI processes was varied (see files *com_senga2*.h and cont*.dat*):

- Number of processes (NXPROC, NYPROC, NZPROC / nx, ny, nz)
- Local grid size (NXSIZE, NYSIZE, NZSIZE)
- Size of parallel transfer array (NPARAY)

Any changes to files *cont.dat* and/or *com_senga2.h* requires recompilation before the code can be run. The modified input files used to obtain the performance data for this benchmark can be obtained from Neelofer Banglawala (EPCC).

2. Performance Results

The plot below shows a performance comparison between the KNL and Xeon systems for 1, 2, 4 and 8 nodes on each system with 1, 2 and 4 hyperthreads (HTs) per core. Each KNL node (with a single KNL processor) has 64 physical cores and each Xeon node (with 2 Xeon processors) has 24 physical cores. All nodes were fully-populated, meaning 1 MPI process per compute core (i.e. both physical and logical cores).



For default hyperthreading (1 HT per core), the KNL node provides similar performance to a Xeon node. For more than 1 HT per core, the KNL system is generally more performant than the Xeon system. For both systems, the best (absolute) performance is on 8 nodes with 2 HTs per core, with relatively little difference between the two (4% speedup on the KNL) . The largest performance gain on the KNL, relative to the Xeon system, is on 2 KNL nodes with 4 HTs per core (maximum hyperthreading), giving a ~24% speedup. Note that SENGA2 did not run correctly on 8 KNL nodes with full hyperthreading enabled. It is unclear why this was.

The table below compares the KNL to Xeon performance as a function of the number of nodes with 1, 2, and 4 HTs per core.

HT = 1			
Nodes	**KNL/s**	**Xeon/s**	**KNL speedup %**
1	222	223	0.45
2	114	117	2.56
4	56	56	0.00
8	28	28	0.00

HT = 2			
Nodes	**KNL/s**	**Xeon/s**	**KNL speedup %**
1	173	189	8.47
2	84	97	13.40
4	44	49	10.20
8	23	24	4.17

HT = 4				
Nodes	**KNL/s**	**Xeon/s**	**KNL speedup % ***	**KNL speedup %****
1	150	N/A	20.63	13.29
2	74	N/A	23.71	11.90
4	38	N/A	22.45	13.64
8	Error	N/A	Error	Error
*** relateive to Xeon HT = 2** **** relative to KNL HT = 2**				

3. Summary and Conclusions

- The KNL system provides the best overall performance with 8 fully-populated nodes and 2 hyperthreads per core. This gives a marginal ~4% performance improvement over 8 Xeon nodes with maximum hyperthreading (2 hyperthreads per core).
- On the Xeon system, the best performance is achieved with 8 fully-populated nodes and maximum hyperthreading.
- For both the KNL and Xeon systems, this benchmark should be run with fully-populated nodes i.e. one MPI process per compute core, and hyperthreading enabled. On the KNL system, the difference between 2 and 4 hyperthreads is noticeable, giving ~14% speedup when the number of hyperthreads is doubled.
- This benchmark does not currently run on 8 KNL nodes with maximum hyperthreading. This should be further investigated. Benchmarking on larger KNL systems is required to gain a better understanding of how performance varies across KNL nodes at scale.