KNL Performance Comparison: CASTEP

March 2017

1. Compilation, Setup and Input

Compilation

CASTEP 16.1.2 was compiled on both the ARCHER KNL and Xeon systems according to the instructions found here:
https://github.com/ARCHER-CSE/build-instructions/blob/master/CASTEP/build_castep_16.1.2_intel16_ivybrg.md

The only difference from the above instructions is that for the KNL system the "linux_x86_64_ifort16-XT" make configuration file was copied to "linux_x86_64_ifort17-XT" in the obj/platforms directory of the CASTEP source tree, so that the build system had instructions to build with the Intel 17 Compilers.

Compilers and library versions used were:

Compiler/Library	ARCHER Xeon	ARCHER KNL
Intel Compiler	16.0.2.181	17.0.0.098
FFTW	3.3.4.10	3.3.4.10

Setup

- The ARCHER KNL nodes were used in the “quad_100” configuration with all the MCDRAM configured as an additional cache level.

- Each (KNL and Xeon) node was fully populated by either MPI processes or a combination of MPI processes and OpenMP threads. Hyperthreads were not enabled.

Input

The “medium”-sized Al3x3 benchmark was used. The files can be found at this link:
http://www.castep.org/CASTEP/Al3x3

The al3x3 simulation cell comprises a 270-atom sapphire surface, with a vacuum gap.

2. Performance Data

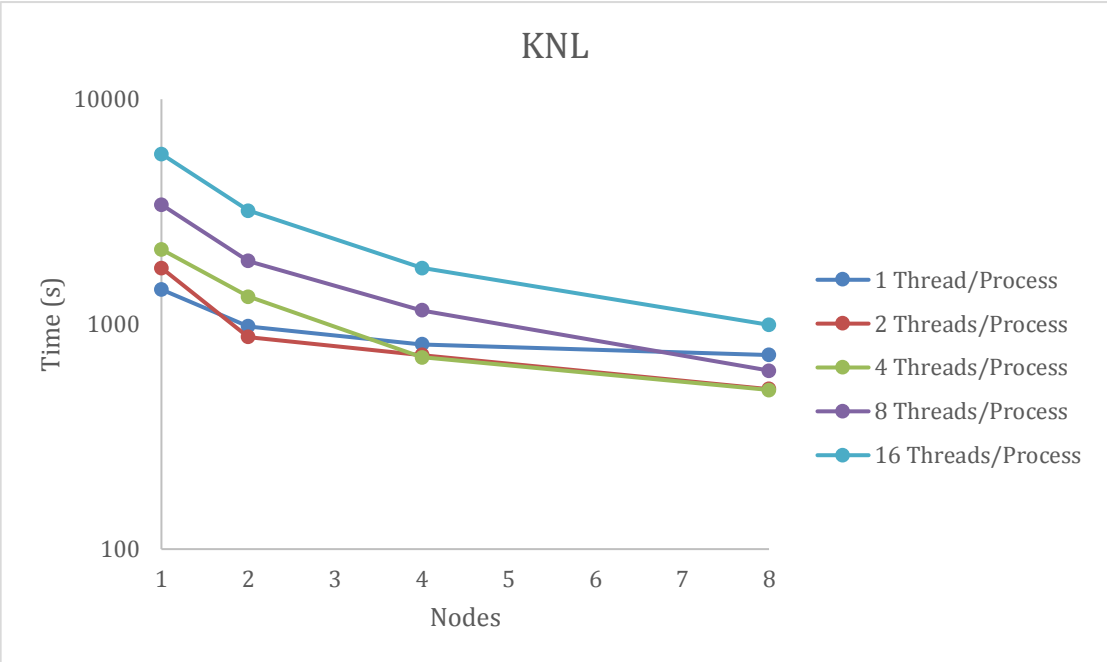
For the KNL and Xeon systems, the benchmarks were run on 1, 2, 4 and 8 nodes, for the following combination of MPI processes/threads per node:

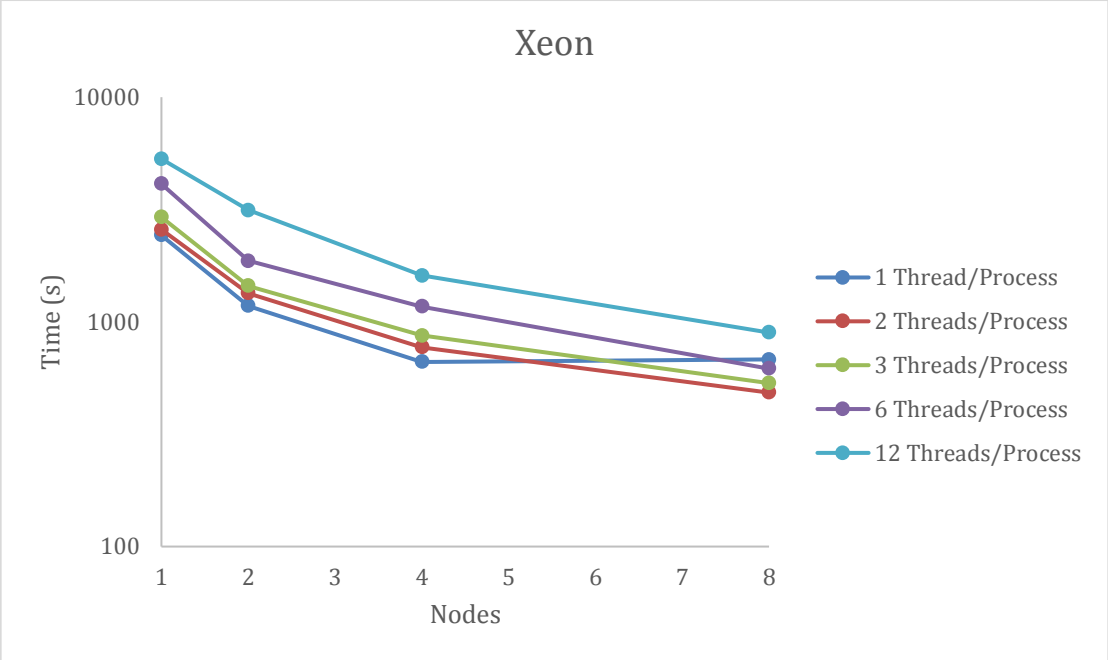
	Processes per node (OpenMP threads per process)				
KNL	64 (1)	32 (2)	16 (4)	8 (8)	4 (16)
Xeon	24 (1)	12 (2)	8 (3)	4 (6)	2 (12)

All numeric results can be found in the appendix.

2.1 MPI vs OpenMP

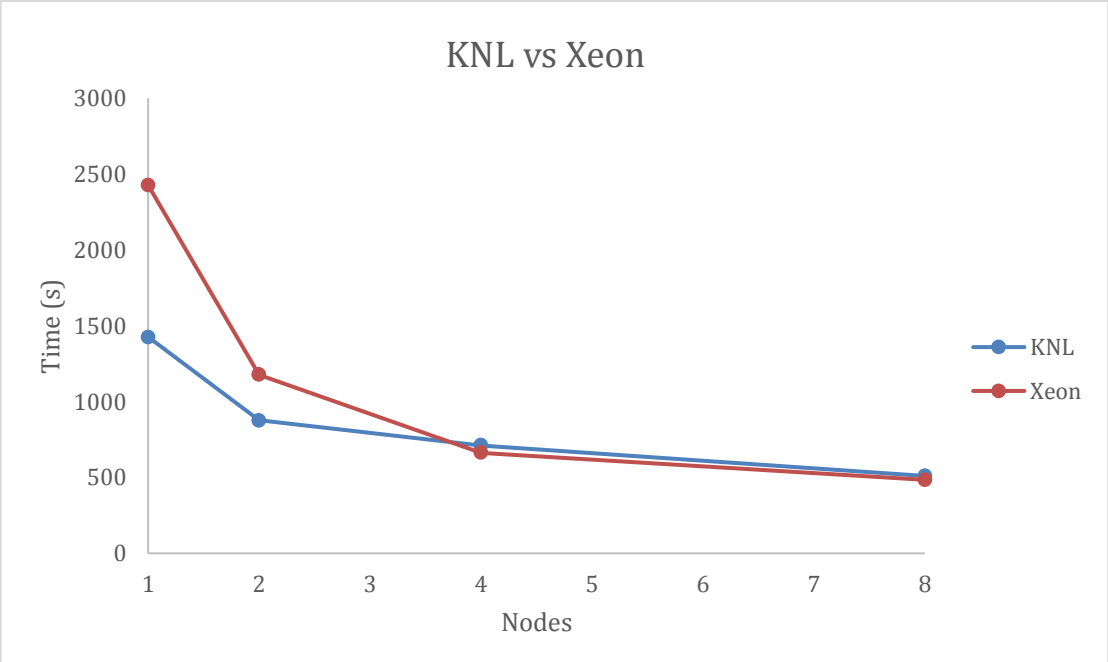
For both systems, with a single node, performance is best using pure MPI, with no OpenMP. For the KNLs, with more than one node the best performance is found using two - or for larger node-counts - four OpenMP threads per MPI process. On the Xeon system, except for the case of eight nodes, pure MPI performs best.





2.2 KNL vs Xeon

Taking the best combination of MPI and OpenMP each number of nodes, it is found that for one and two nodes, the KNL system performs better than the Xeon system, but for a greater number of nodes the Xeon system performs best, although not by a significant amount.



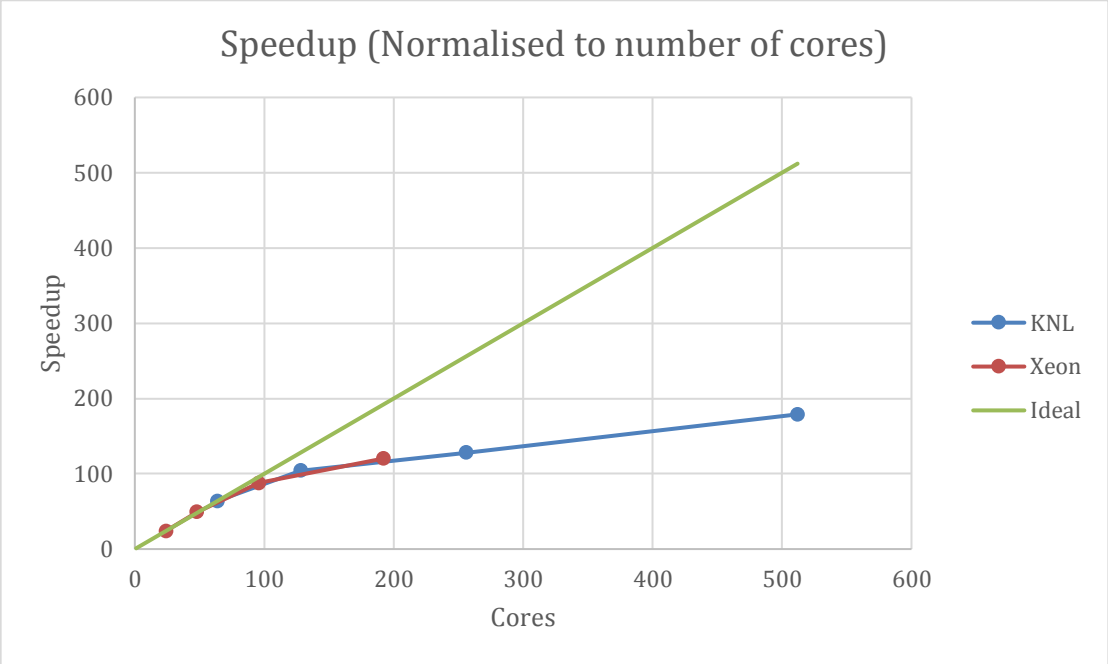
2.3 Speedup analysis

In order to directly compare the speedups between the KNL and Xeon systems, we calculate the speedup relative to one node, then multiply this by the number of cores per node (24 in the case of the Xeon, and 64 in the case of the KNL). The speedup graph (below) shows that for both

systems, the speedup begins to deviate significantly from ideal at around 100-150 cores, implying the problem does not scale well beyond this core-count.

The poor scaling for high core-counts could explain why the Xeon system has better performance than the KNL for larger node counts, since the KNL nodes have more cores per node, so reach the point where the problem no longer scales at a lower node count (2-3 nodes), whilst this is not achieved until around 6 nodes in the case of the Xeon nodes.

The poor scaling of the problem could also contribute to the findings that on the KNL performance was best with two or four OpenMP threads per MPI process.



3. Summary and Conclusions

In general, it was found that the KNLs performed better than the Xeon nodes for small (1-2) node-counts, although the Xeon nodes performed better for larger node-counts.

On the Xeon system (except for 8 nodes) best performance was attained using pure MPI, however for the KNL system performance was best using 2 OpenMP threads per MPI process for greater than one node.

Considering the speedup results, it is found that the benchmark problem only scales up to 100-150 cores. Much of the above findings about KNL vs Xeon must therefore be considered cautiously, as two KNL nodes have 128 cores, and so have reached the point where the problem no longer scales. Ideally a larger benchmark would be used, but the only available benchmarks that are larger are too big to be run on eight KNL or Xeon nodes.

Appendix – Numerical Results

KNL nodes:

Nodes	Threads					Best	Speedup	Speedup (cores)
	1	2	4	8	16			
1	1426	1778	2150	3399	5709	1426	1.00	64
2	976	876	1323	1908	3195	876	1.62	104
4	813	729	712	1153	1775	712	2.00	128
8	729	514	510	622	993	510	2.79	178

Xeon Nodes:

Nodes	Threads					Best	Speedup	Speedup (cores)
	1	2	3	6	12			
1	2426	2576	2923	4108	5308	2426	1.00	24
2	1177	1342	1446	1871	3136	1177	2.06	49
4	662	771	867	1169	1608	662	3.65	88
8	680	484	533	619	896	484	5.00	120