Performance of Popular HPC Applications on the Intel Knights Landing Platform

Antonio Gómez-Iglesias, Feng Chen, Lei Huang, Hang Liu, Si Liu, Antia Lamas-Linares, John Cazes, Carlos Rosales

Texas Advanced Computing Center

TEXAS ADVANCED COMPUTING CENTER

WWW.TACC.UTEXAS.EDU

Abstract

We present direct performance measurements for six popular HPC applications in the Knights Landing platform. Performance numbers for Sandy Bridge and Haswell processors are provided for contrast. The applications were selected from among the ten most used in the Stampede supercomputer at the Texas Advanced Computing Center. Current results are promising, with execution on a single KNL processor showing speedups up to 2.7X with respect to a dual socket Sandy Bridge and up to 1.7x with respect to a dual socket Haswell.

Introduction

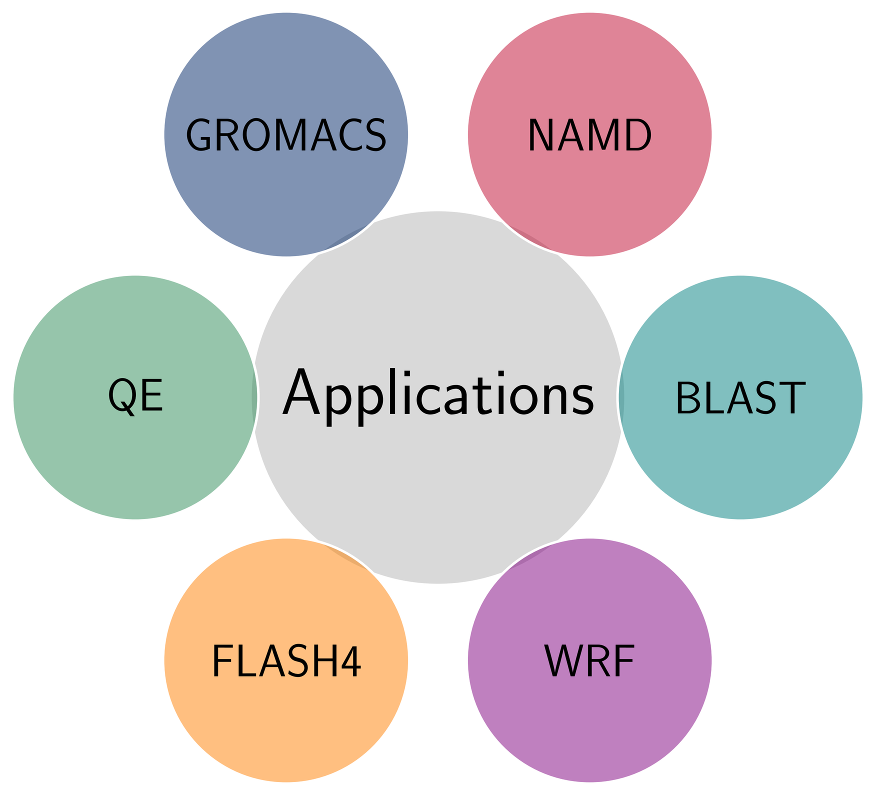
The relentless pursuit of increase in computational power has lead to a common trend in the World of HPC: increased core counts and more specialized CPU cores. Accelerators such as GPUs or the first generation Intel Xeon Phi (KNC) are good examples of this trend, providing high peak performance within a moderate power envelope. Accelerated systems are not without challenges though. The presence of a PCIe interface between host CPU and accelerator often creates a data motion bottleneck that reduces the achieved sustained performance of scientific applications. Minimizing this bottleneck in heterogeneous computing infrastructures requires complex implementations and increases both time of development and code maintenance cost.

The second generation of the Intel Xeon Phi processor, the Knights Landing (KNL) platform, attempts to address some of these issues by providing a many-core architecture on a socket. On paper the KNL architecture represents a qualitative leap over the KNC. It includes features that make programming simpler and more effective such as support for out-of-order and unaligned fast instructions, as well as a completely redesigned connectivity mesh across cores and attached high speed memory. The single most important change in KNL is the fact that it is a self-hosted processor running a regular Linux OS, making it amenable to homogeneous cluster configurations and eliminating the PCIe bottleneck present in heterogeneous accelerated solutions.

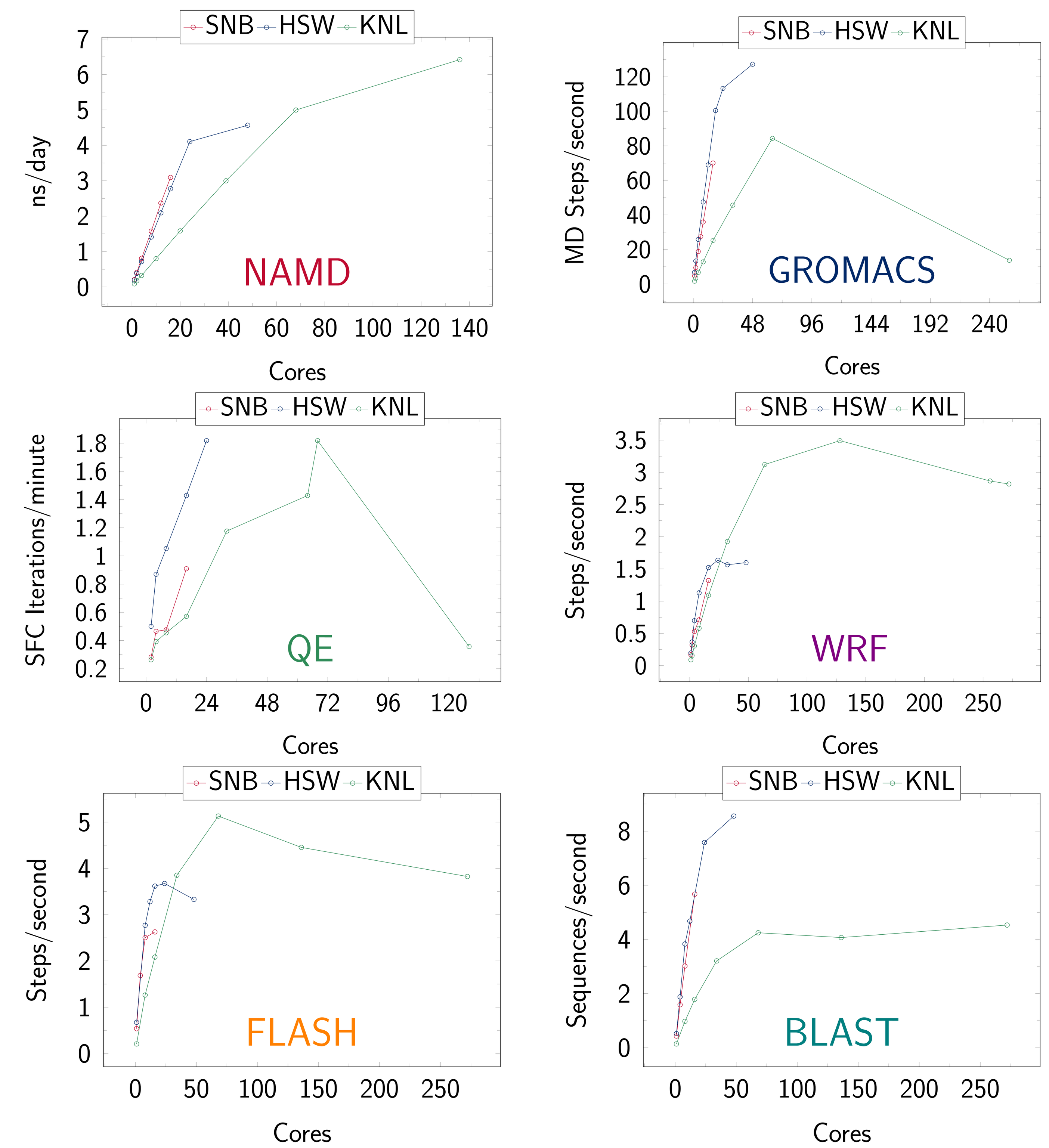
We present an evaluation of the performance of several well known HPC applications, which have been chosen based on the overall utilization of the Stampede supercomputer at the Texas Advanced Computing Center [1, 2, 3, 4, 5, 6]. Stampede hosts thousands of users from all backgrounds and research fields, and thus these applications should be representative of the wider range of workloads used in across the scientific computing community.

Case Studies

- ▶ Stampede: Sandy Bridge, dual socket Xeon E5-2680, 32 GB DDR3.
- ▶ Lonestar 5: Haswell, dual socket Xeon E5-2690v3, 64 GB DDR4.
- ▶ Stampede KNL Upgrade: Knights Landing 7250, 16GB MCDRAM + 96 GB DDR4, Cache-Quadrant mode [7].



Results



Normalized Results

The single node performance for each one of the applications on Sandy Bridge, Haswell and KNL are displayed in Fig. 1. **Normally, KNL either outperforms or, at least, provides similar results to those achieved when using Sandy Bridge**. Moreover, **only in the case of GROMACS and NCBI BLAST, Haswell significantly improves the performance achieved with KNL**. A new version of GROMACS already improves these results on KNL. Overall, it is possible to see how the many-core approach is comparable if not better than the used multi-core alternatives provided by Intel. This fact, combined with some of the characteristics of the KNL, makes this new processor an optimal alternative for some of the most popular HPC applications.

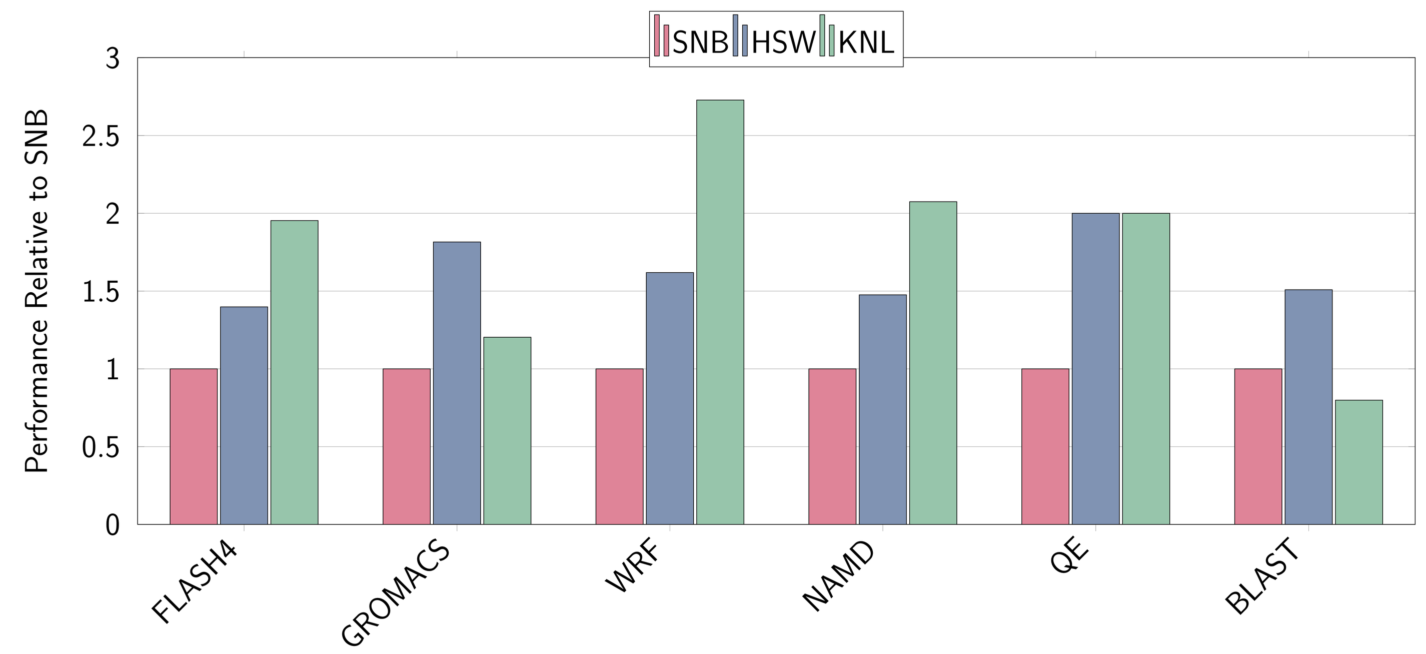


Figure 1: Normalized performance on Sandy Bridge, Haswell and KNL

No effort was put into optimizing any of these codes to target any of the capabilities of the processors analyzed. Because of the maturity of Sandy Bridge and Haswell processors, it is possible that some of these applications have been previously optimized for these targets, while no optimization has been yet added to support KNL.

References

[1] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403 – 410, 1990.

[2] H.J.C. Berendsen, D. van der Spoel, and R. van Drunen. GROMACS: A message-passing parallel molecular dynamics implementation. *Computer Physics Communications*, 91(1):43 – 56, 1995.

[3] Paolo Giannozzi et al. Quantum espresso: a modular and open-source software project for quantum simulations of materials. *Journal of Physics: Condensed Matter*, 21(39):395502, 2009.

[4] W. C. Skamarock et al. A description of the advanced research WRF version 3. Technical report, National Center for Atmospheric Research, 2008.

[5] James C. Phillips, Rosemary Braun, Wei Wang, James Gumbart, Emad Tajkhorshid, Elizabeth Villa, Christophe Chipot, Robert D. Skeel, Laxmikant Kalé, and Klaus Schulten. Scalable molecular dynamics with namd. *Journal of Computational Chemistry*, 26(16):1781–1802, 2005.

[6] David Radice, Sean M. Couch, and Christian D. Ott. Implicit large eddy simulations of anisotropic weakly compressible turbulence with application to core-collapse supernovae. *Computational Astrophysics and Cosmology*, 2(1):1–17, 2015.

[7] A. Sodani, R. Gramunt, J. Corbal, H. S. Kim, K. Vinod, S. Chinthamani, S. Hutsell, R. Agarwal, and Y. C. Liu. Knights landing: Second-generation intel xeon phi product. *IEEE Micro*, 36(2):34–46, Mar 2016.

{agomez,chenk,huang,hliu,siliu,alamas,cazes,carlos}@tacc.utexas.edu