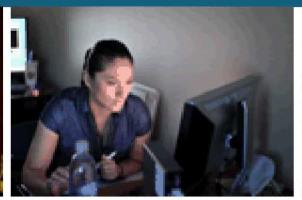
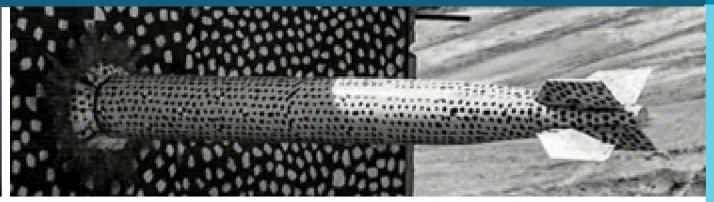
UNCLASSIFIED

SAND2018-10636PE

Investigating Factors Impacting Exascale Performance of ASC Codes: A Co-design Effort





PRESENTED BY

Principal Investigator/Lab: Stephen Olivier / SNL

Platform/Campaign ID: Trinity/atcc5-249[k], Sequoia/atcc5-122
Code Name: Tri-lab mini-apps and related ASC codes
Program: ASC
Usage (with Lin/Wilke): 0.76 (Trinity KNL), 0.86 (Trinity HSW), 0.11 (Sequoia)

SAND #

UNLIMITED RELEASE

UNCLASSIFIED

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Investigating Factors Impacting Exascale Performance of ASC Codes compared the MPI message characteristics of miniapps to their corresponding applications

Background Description:

- Miniapps are designed to model actual applications for codesign, but we need to understand how representative they are.

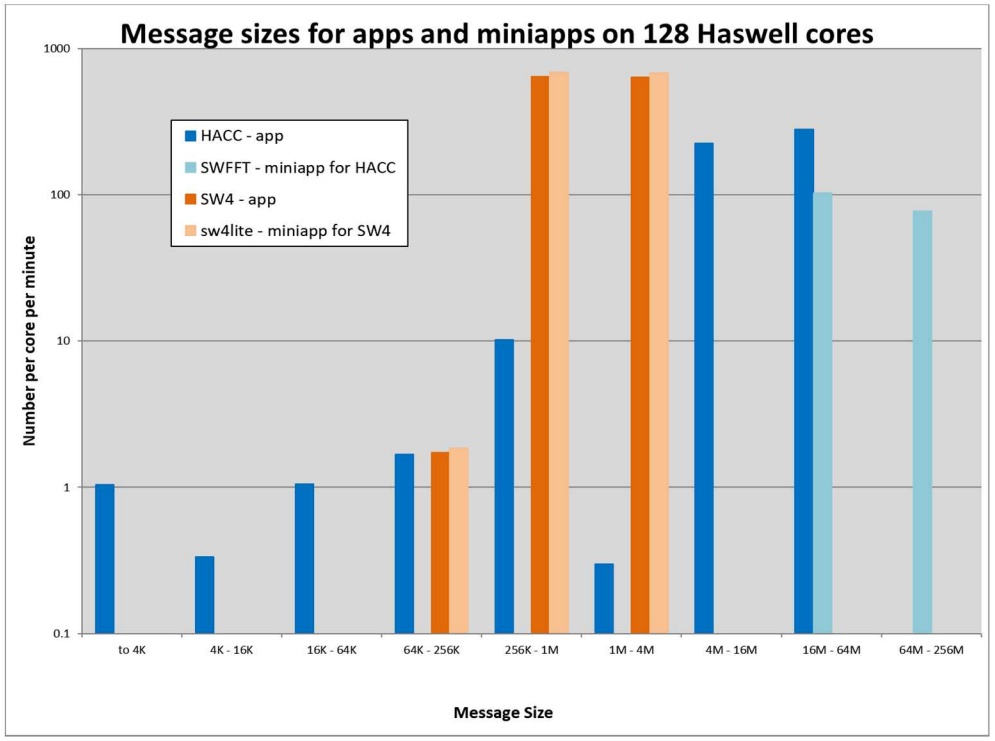
Potential Consequences/Issues:

- Codesign using miniapps may be less impactful if focused on characteristics in which miniapps are not representative of the actual applications.

Resolution/Impact:

- We compared MPI message statistics of applications and corresponding miniapps, and found some more representative than others.

Paper extending this work submitted to an ACM/IEEE Supercomputing workshop



Comparison of MPI message rate in two miniapps and the applications from which they were derived. The x-axis shows bins for different message sizes and the y-axis shows the rate at which messages of that size occur. The bars for the SW4 app and the SW4lite miniapp match closely, indicating similar message passing behavior in both. The bars for the HACC app and the SWFFT miniapp do not match very closely, indicating dissimilar message passing behavior in the app and miniapp.

Take Home Message: Some mini-apps model MPI messaging of their corresponding apps better than others

Investigating Factors Impacting Exascale Performance of ASC Codes demonstrated increased performance on sorting workloads using a new algorithm for KNL "near" memory

Background Description:

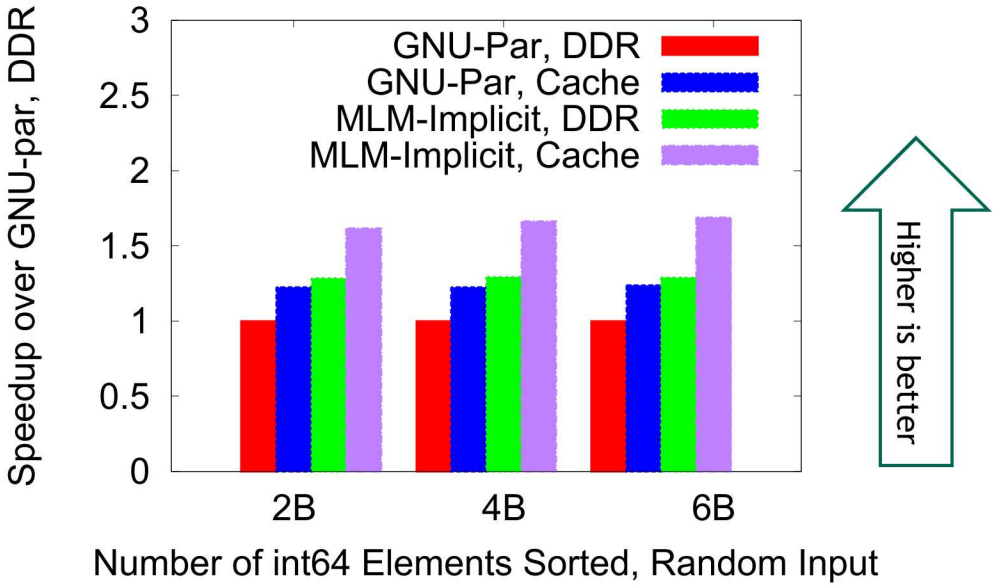
- Customizing entire ASC applications to use the "flat" (explicit data transfer) mode of KNL processors is impractical.

Potential Consequences/Issues:

- High bandwidth "near" memory of KNL is not put to optimal use in our applications, since the "cache" mode is a naïve mechanism.

Resolution/Impact:

- We designed a KNL-optimized sort algorithm to demonstrate that key kernels can better leverage "near" memory even in cache mode.



Speedup of our sort algorithm (MLM-implicit) versus the current state-of-the-art GNU parallel sort (GNU-Par) on the Knights Landing (KNL) processor for 2-6 billion integer arrays. Results are shown both for DDR memory use only and in cache mode. Even using only DDR memory, our algorithm outperforms GNU. It improves even more in cache mode.

Neil Butcher, Stephen L. Olivier, Jonathan Berry, Simon D. Hammond, and Peter M. Kogge. "Optimizing for KNL Usage Modes When Data Doesn't Fit in MCDRAM." In *Proceedings of International Conference on Parallel Processing (ICPP 2018)*. ACM, New York, NY, USA.

Take Home Message: Algorithms for "near" memory can increase performance even in cache mode