ISx: A Scalable Integer Sort Mini-Application

OpenSHMEM

PGAS

Jacob Hemstad (Intel and University of Minnesota), Ulf R. Hanebutte (Intel)

What is ISx?

- A new open source, parallel integer sort mini-application
- In memory bucket sort with built-in verification
- OpenSHMEM*, MPI* 1.0, and Chapel*^ implementations
- Allows for arbitrarily large problem sizes; demonstrated with ½ million ranks
- Supports both strong and weak scaling studies

Why create ISx?

- Parallel integer sort is an essential problem with many salient features:
  - An all-to-all with non-uniform message sizes that are not known *a priori*
  - Opportunity for communication and computation overlap
- Succeed NAS IS as the *de facto* parallel integer sort research tool at scale
- Provide a concise application for studies in OpenSHMEM and PGAS

ISx: OpenSHMEM vs. MPI 1.0 All-To-All Results

- By comparing an OpenSHMEM implementation of ISx to a traditional two-sided MPI implementation, we hope to better understand the following questions:
  - What kind of communication patterns benefit from a one-sided/PGAS implementation over a message passing implementation?
  - When there is a benefit, how much?
- Our results show up to a 65% reduction in execution time for the key exchange with the OpenSHMEM implementation vs. an MPI 1.0 alltoallv implementation
- For applications that feature an irregular all-to-all, a one-sided/PGAS implementation can have significant benefits over a two-sided collective
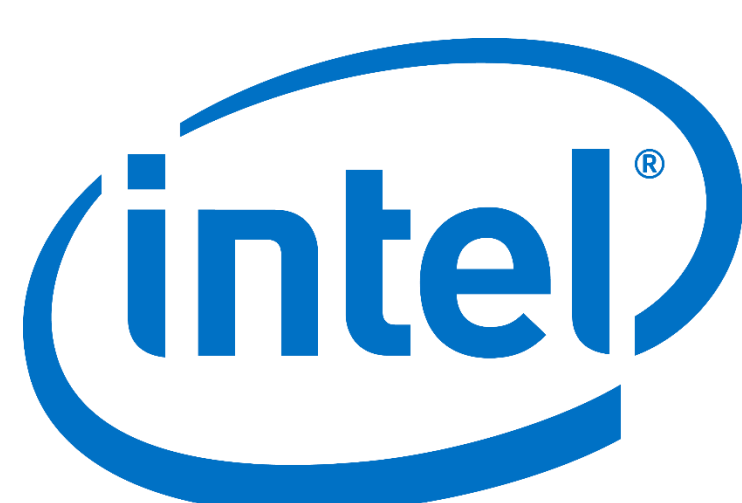
ISx: All-To-All Time
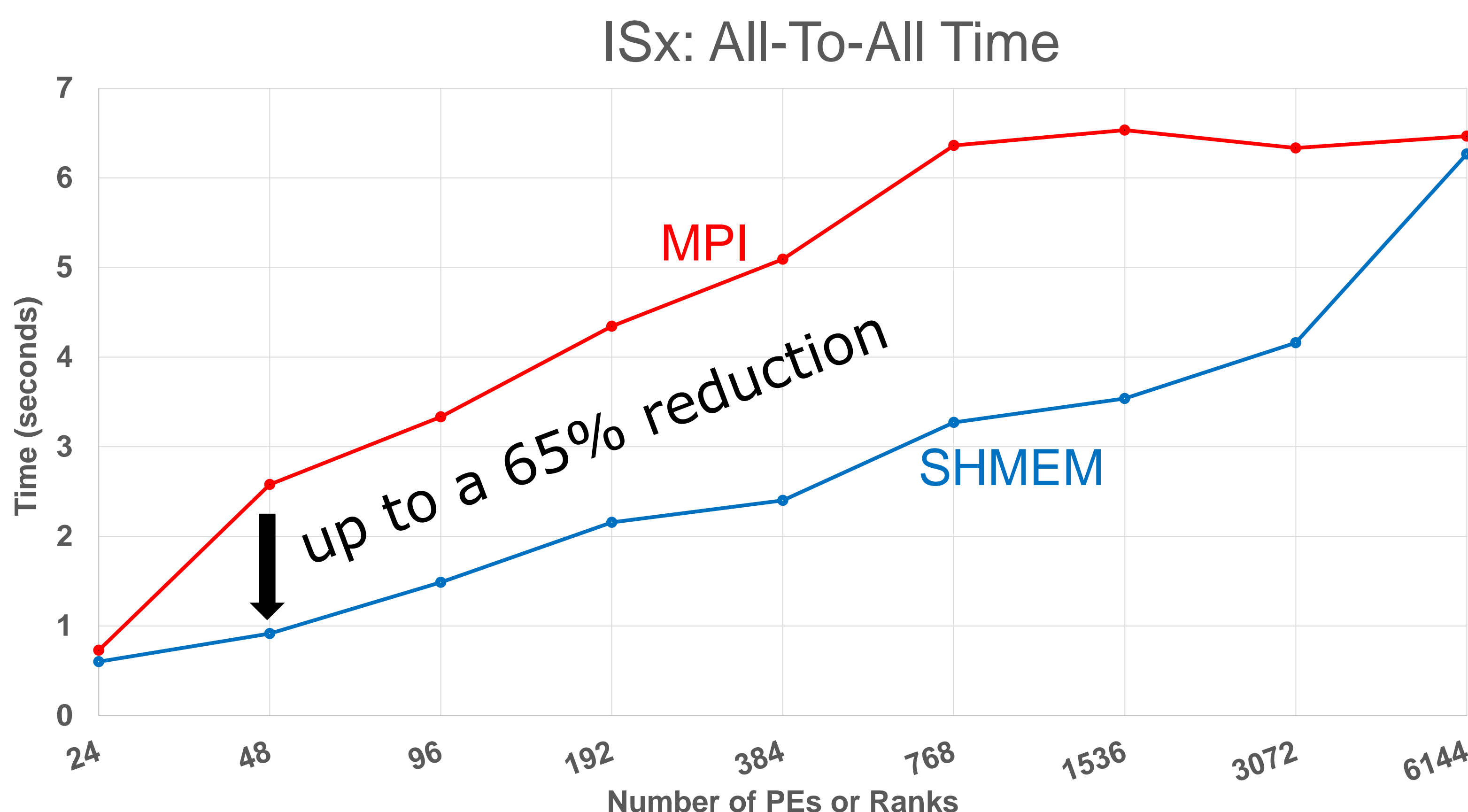
Experimental Details
- Weak Scaling with 2^{27} keys per rank
- For each number of ranks, we ran 10 unique jobs and are reporting the best performing of those 10 jobs
- Results collected on Edison @ NERSC
  - Cray® XC30 with 5,576 nodes
  - 2x 12-core Intel® Xeon® E5-2695 v2, 2.4GHz
  - Aries™ w/ Dragonfly topology

https://github.com/ParRes/ISx