IEEE CCGRID 2025
The 25th IEEE International Symposium on Cluster, Cloud and Internet Computing
Tromsø, Norway

ActorISx: Exploiting Asynchrony for Scalable High-Performance Integer Sort

Youssef Elmougy, Shubhendra Singhal, **Akihiro Hayashi***, and Vivek Sarkar

Habanero Extreme Scale Software Research Lab
Georgia Institute of **Tech**nology

* Presenting Author: ahayashi@gatech.edu

Georgia Tech

Integer and Data Movement Performance in HPC

- Integer sorting remains a critical operation in HPC, playing a vital role in numerous applications
  - Graph Analytics
  - Distributed Sparse Linear Algebra
  - Scientific Simulations
  - and more!
- The Integer Sort (IS) kernel from the NAS Parallel Benchmark (NPB) has served as a standard for assessing integer and data movement performance in HPC systems since 90s
  - Two-sided MPI-based Bucket Sort Algorithm with `MPI_Alltoallv`

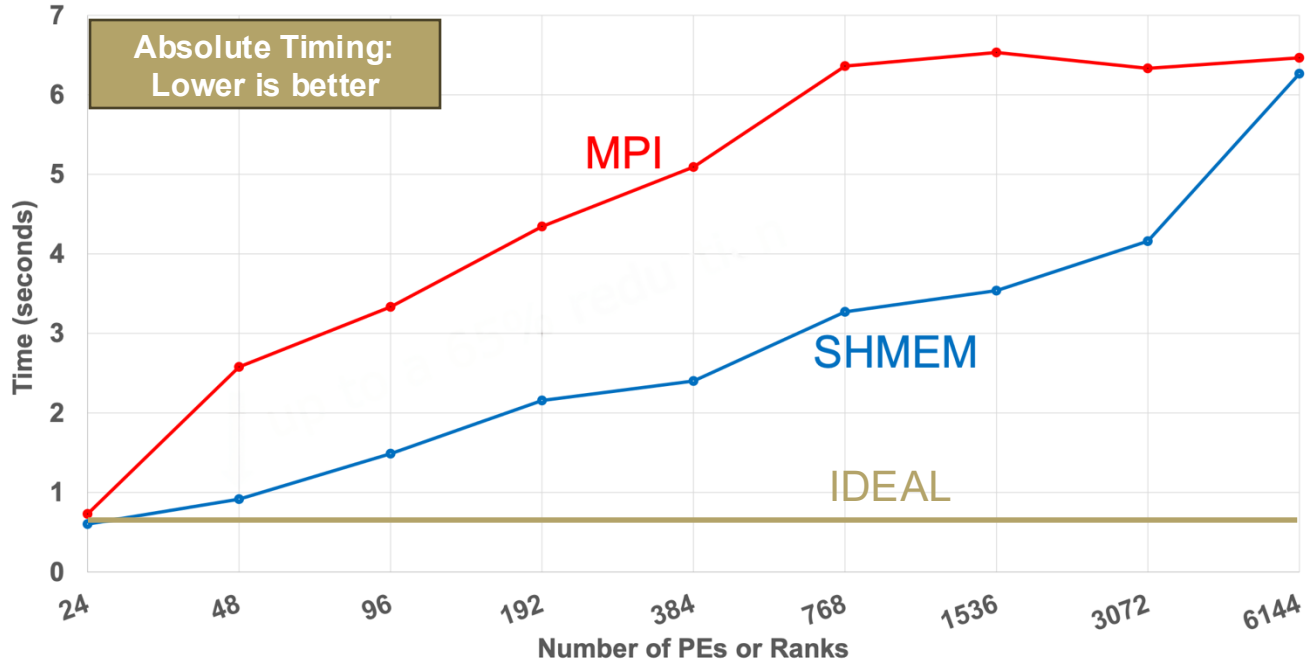
> IS — The NPB 2 implementation of the IS kernel benchmark is based on a bucket sort. The number of keys ranked, number of processors used, and number of buckets employed are all presumed to be powers of two. This simplifies the coding effort and leads to a compact program. The number of buckets is a tuning parameter. On the systems tested, best performance was obtained when the number of buckets was half that which gives best load balancing. Communication costs are dominated by an MPI_Alltoallv, wherein each processor sends to all others those keys which fall in the key range of the recipient.

Georgia Tech

ISx: Addressing limitations of the original NPB IS

- The development of **ISx** addressed several limitations of the original NPB IS benchmark by introducing OpenSHMEM's one-sided communication

Weak-Scaling Performance of the Key Exchange part of ISx (2^{27} keys per PE)



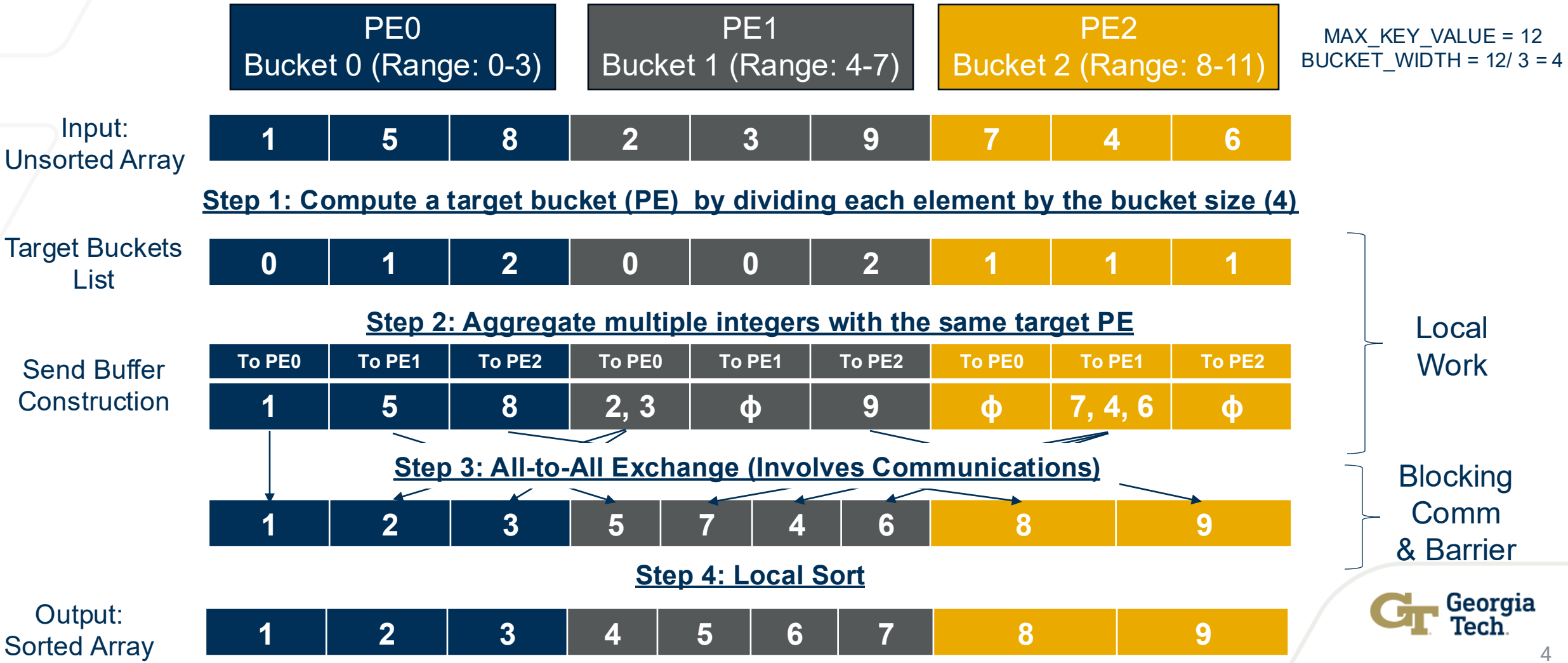
Research Question: Can we come closer to the IDEAL performance on modern distributed HPC systems?

Platform: Edison @ NERSC (decommissioned)
The image has been altered from its original source for clarity and simplicity in presentation: https://www.github.com/ParRes/ISx

Georgia Tech

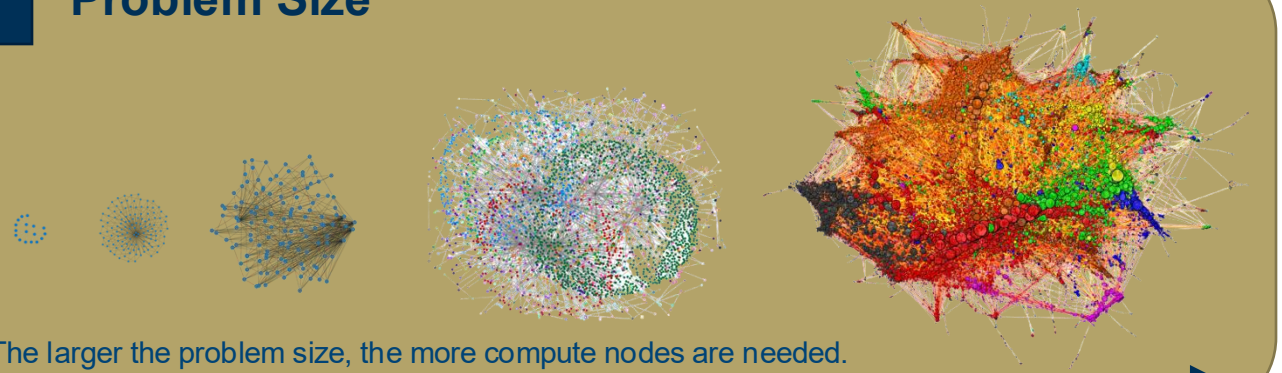
ISx: High-Level Overview

Goal: Sort a distributed array on 3 PEs (Ranks) using the Bucket Sort Algorithm
Input: an unsorted array, Output: a sorted array



The 3 Challenges in Scalable Integer Sort:

1 Problem Size

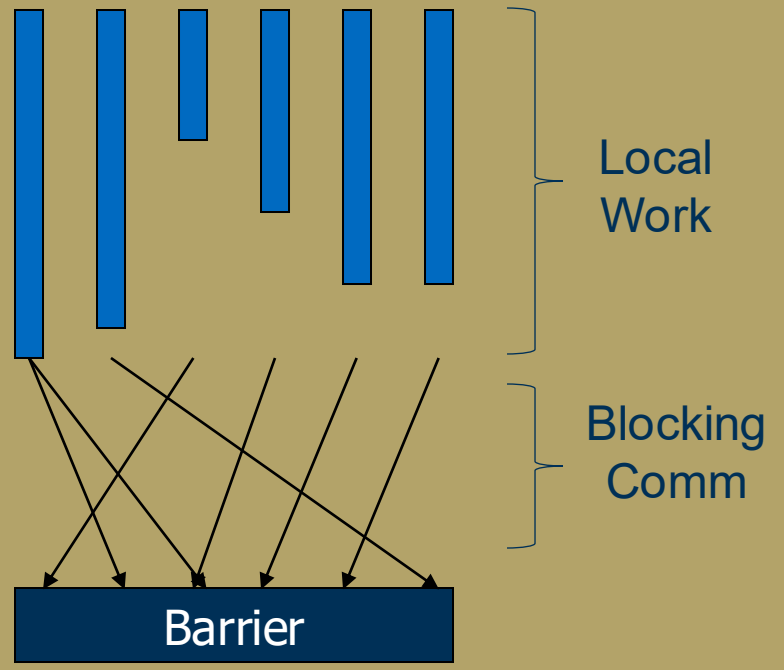


The larger the problem size, the more compute nodes are needed.

2 Irregular Messaging for Arbitrary Destinations

From	To PE0	To PE1	To PE2	…
PE0	1	5	8	
PE1	2, 3	ϕ	9	
PE2	ϕ	7, 4, 6	ϕ	
…				

3 Bulk Synchronous Parallelism (BSP)

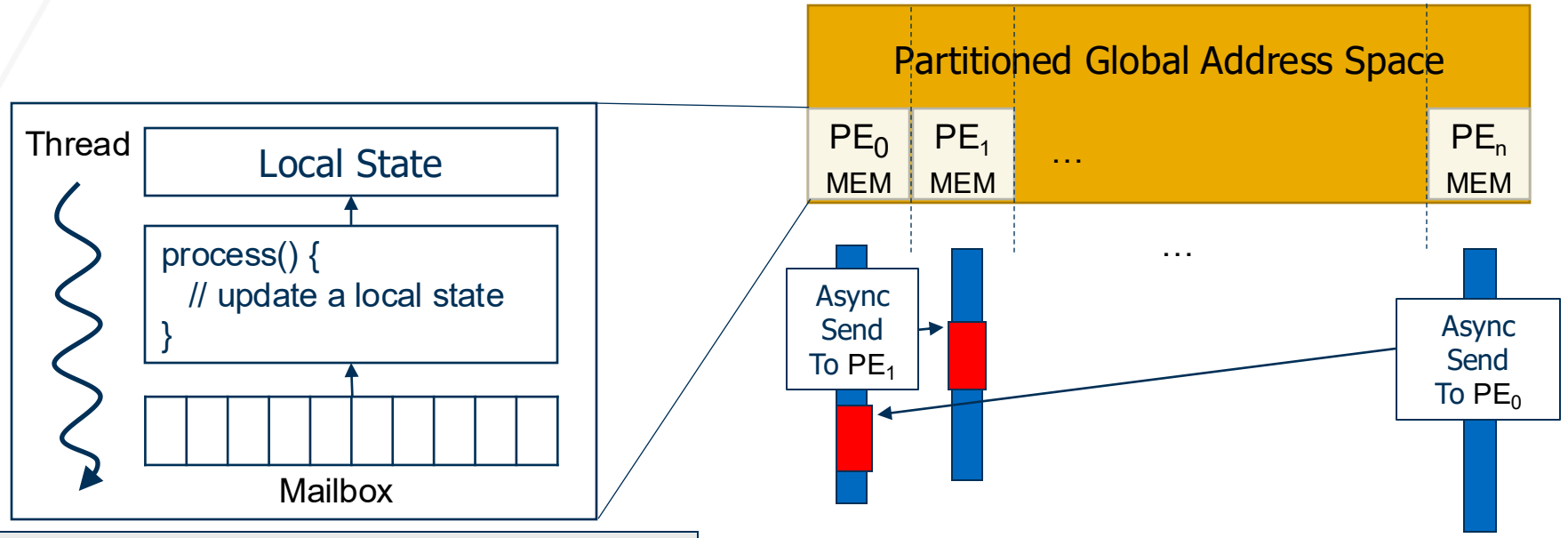


This paper studies the scalability of our actor-based approach to overcome the inherent challenges of traditional programming models by performing large-scale integer sorting.

Georgia Tech

Our Vision: Fine-grained-Asynchronous Bulk-Synchronous Model (FA-BSP)

- FA-BSP = PGAS + Actor-based asynchronous messaging + BSP
  - Actor = PE = a thread/rank that owns a slice of the global address space
  - Communications between PEs are achieved via asynchronous active messages
  - The extended barrier waits for all active messages to complete in a superstep



Local Work

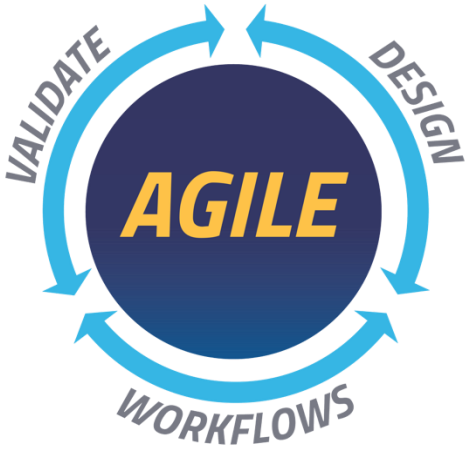
Async Active Messages

Extended Barrier (waits for all threads & all active messages to complete)

"A Fine-grained Asynchronous Bulk Synchronous parallelism model for PGAS applications." Sri Raj Paul, Akihiro Hayashi, Kun Chen, Youssef Elmougy, Vivek Sarkar. JCS 2023

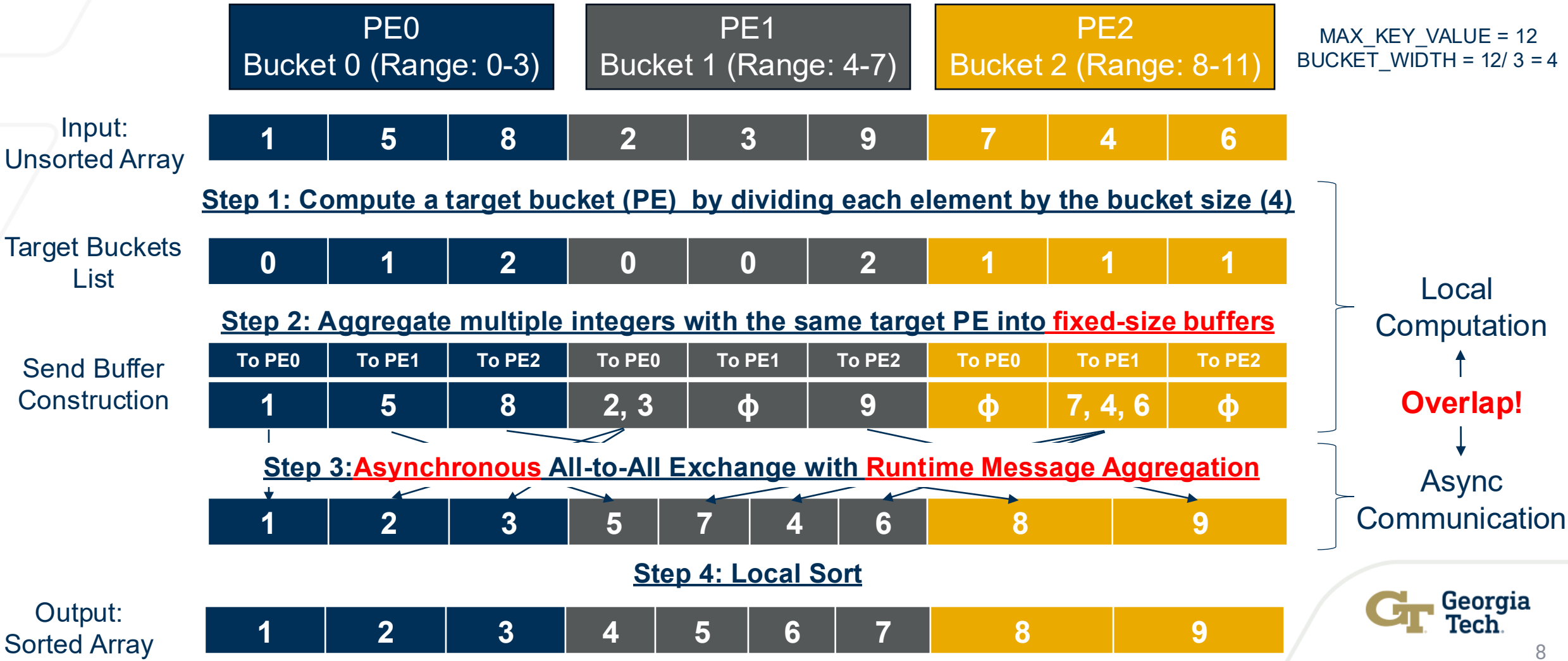
Many distributed graph/non-graph algorithms can be implemented using our FA-BSP model

- Our FA-BSP model is well-suited for graph and non-graph algorithms
- Bale Kernels (JCS'23)
  - Histogram
  - Index Gather
  - Permute Matrix
  - Random Permutation
  - Transpose Matrix
  - Triangle Counting
  - Toposort
- Other Graph kernels
  - Triangle Centrality (SCALE Challenge at CCGRID'24)
  - Page Rank (SCALE Challenge at CCGRID'23)
  - Jaccard Coefficients (ISC'24)
  - Triangle Counting (SC'23 Poster)
  - BFS
- IARPA AGILE Workflows
  - Graph Neural Networks
  - Pattern Matching (IPDPSW'25)
  - K-mer Counting (IPDPS'25)
  - Influence Maximization (SC'24)



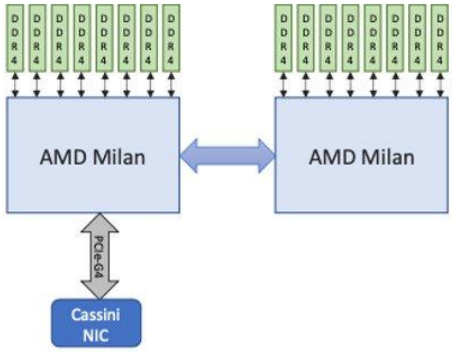
ActorISx: Exploiting Asynchrony for Scalable High-Performance Integer Sort

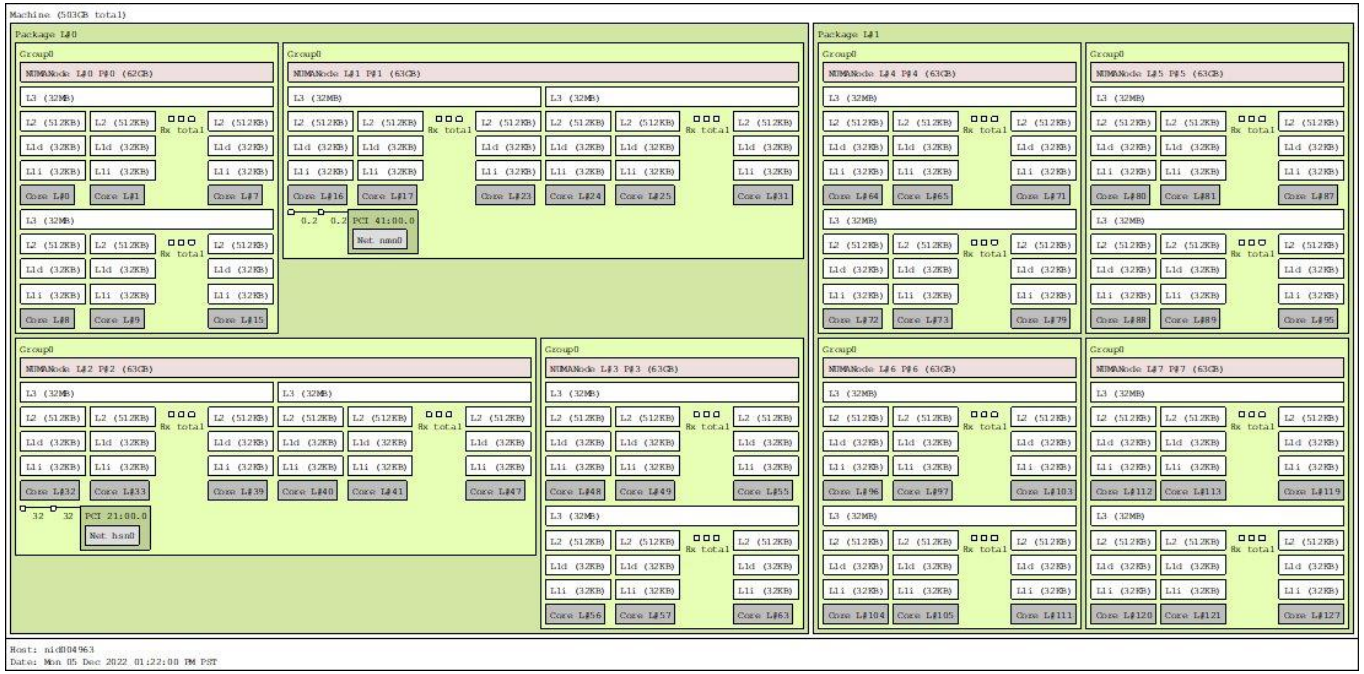
ActorISx enables asynchronous point-to-point communication with application-level and runtime-level message aggregation



Experimental Setup and Architecture

- Experiments conducted on the CPU nodes of the **Perlmutter supercomputer** at the National Energy Research Scientific Computing Center (NERSC)
  - 2x AMD EPYC 7763 (Milan) CPUs
  - 64 physical cores per CPU
  - 512 GB memory
  - 1x HPE Cray Slingshot Interconnect
- Results for **different dimensions of scalability** are presented





Picture borrowed from: https://docs.nersc.gov/systems/perlmutter/architecture/

Dimensions of Scalability

The paper studies two weak-scaling scenarios from the original ISx:

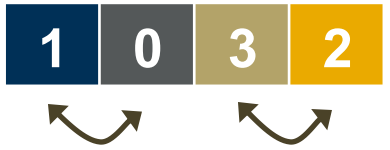
	WEAK SCALING	ISO WEAK SCALING
# of cores	Variable (2^6 - 2^{14})	
Keys per core	Constant (2^{27})	
MAX_KEY_VALUE	Constant (2^{28})	Variable (2^{19} - 2^{27})
BUCKET_WIDTH = MAX_KEY_VALUE / # of cores	Variable (2^{22} to 2^{14})	Constant (2^{13})
Stability of Communication	**Unstable**	**Stable**

Example (**Ultimate Case**)

MAX_KEY_VALUE = 4
BUCKET_WIDTH = 4 / # of cores

2 cores 

No communication

4 cores 

Unstable (depends on # of cores)

MAX_KEY_VALUE = 4 - 8
BUCKET_WIDTH = 2

2 cores 

No communication

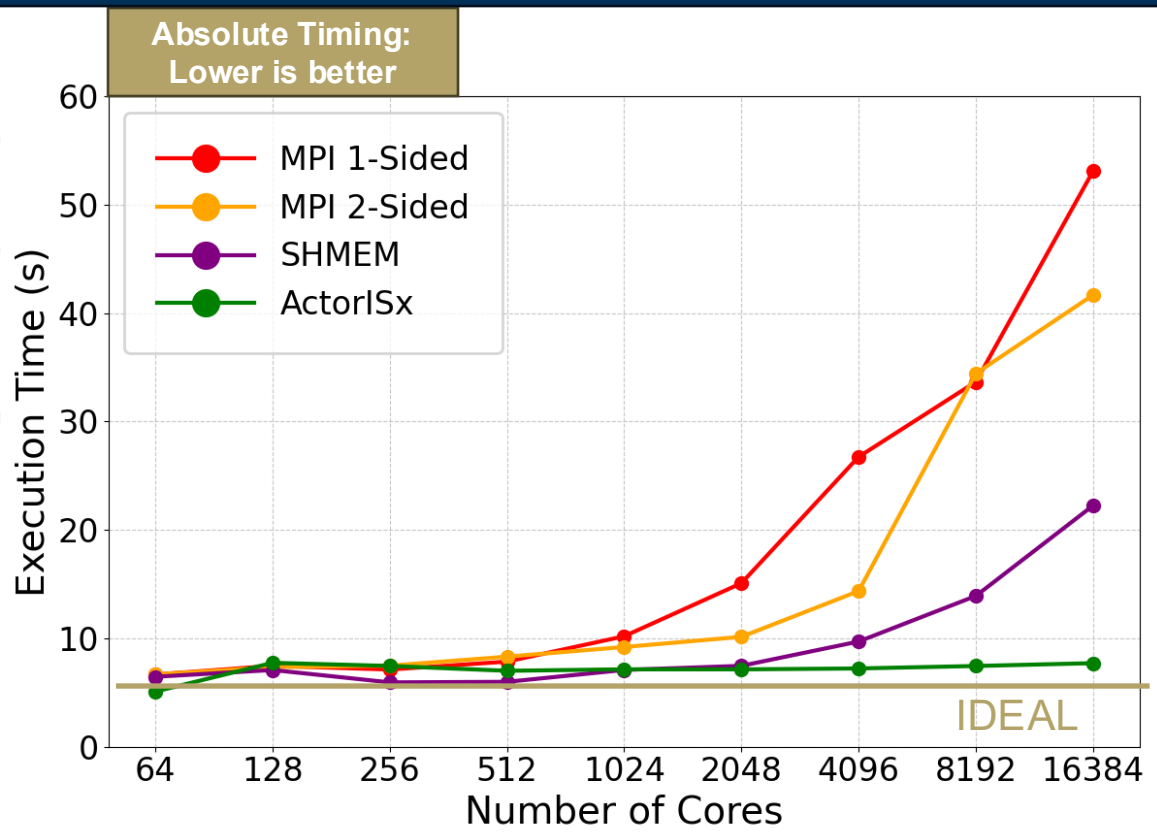
4 cores 

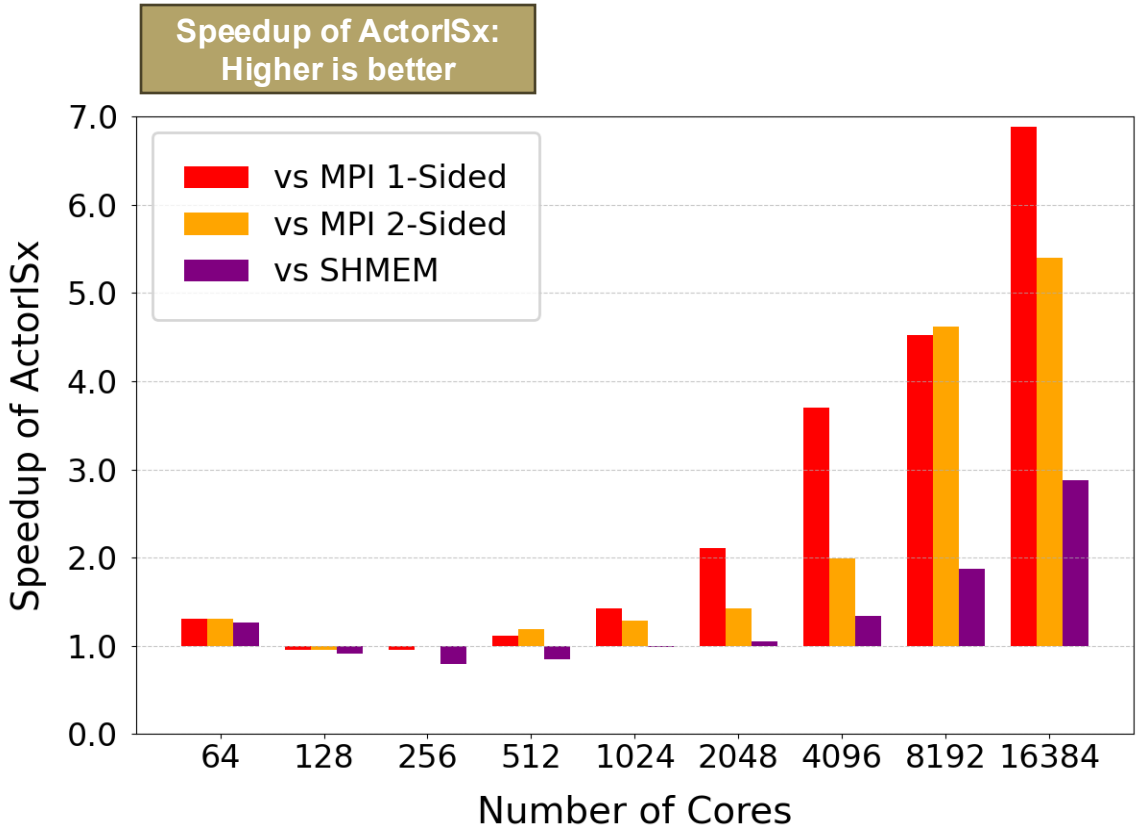
No communication

Stable (independent of core count)

Weak Scaling Results on Perlmutter (up to 16k cores): CONSTANT max key (2^{28}): Unstable communication

2^{27} keys per core => 2^{33} keys (64 cores), 2^{41} keys (16k cores)



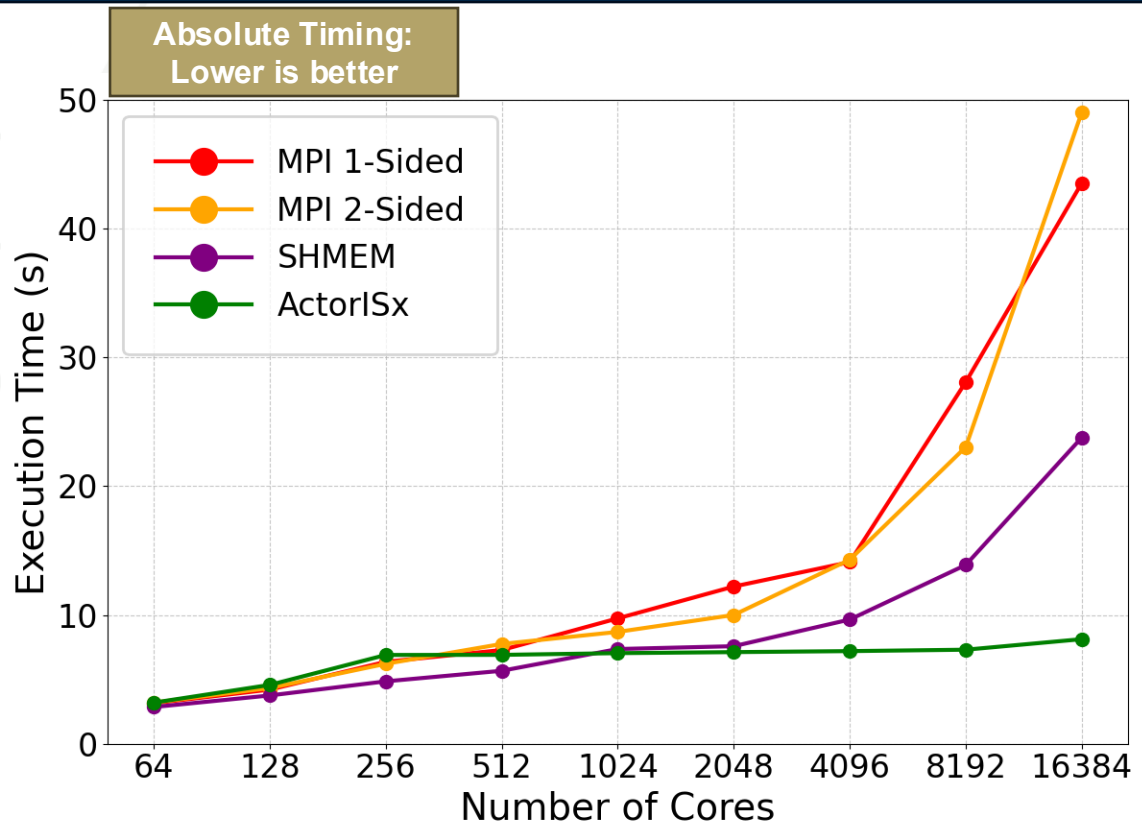


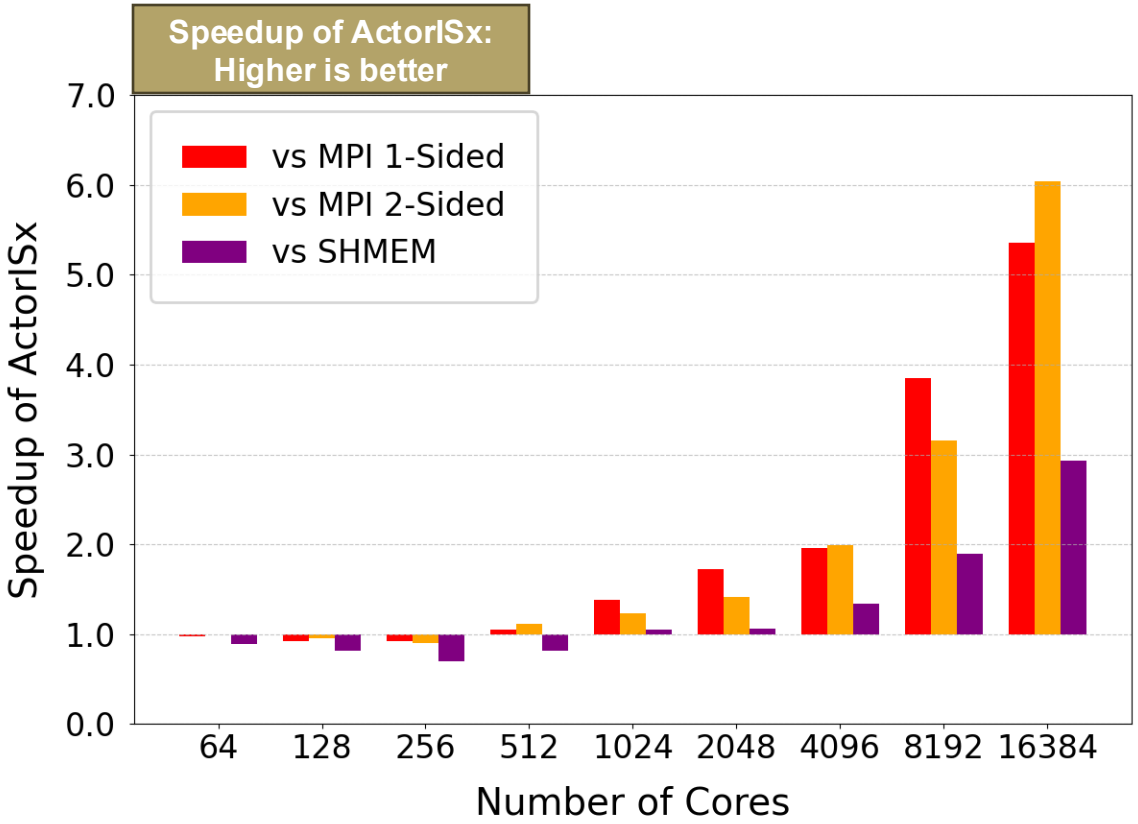
- **ActorISx** achieves almost an ideal weak-scaling result thanks to
  - Asynchronous communication
  - Two-level message aggregation
  - Multi-hop routing

Georgia Tech

ISO Weak Scaling Results on Perlmutter (up to 16k cores): INCREASE max key (2^{19} - 2^{27}): Stable communication

2^{27} keys per core => 2^{33} keys (64 cores), 2^{41} keys (16k cores)





- **ActorISx** achieves almost an ideal weak-scaling result thanks to
  - Asynchronous communication
  - Two-level message aggregation
  - Multi-hop routing

Georgia Tech

Impact of the Solution

- Our algorithm has shown efficient **scalability** and **performance**
- The extensibility of this algorithm has four major impacts:

1. **By leveraging the actor model, we have addressed the fundamental limitations of NPB IS while maintaining high performance and scalability**
2. **Our algorithm can be applied to other data-intensive and communication-intensive applications**
3. **Our actor-based approach is a viable alternative to traditional MPI-based and SHMEM-based approaches**
4. **Our solution shows that actor-based approaches will play an increasingly important role in future HPC systems**

Georgia Tech

DEMO

ActorISx: Exploiting Asynchrony for Scalable High-Performance Integer Sort

You can try this at home... Just visit hclib-actor.com !

Bulk Synchronous Parallel - HC

hclib-actor.com/background/bsp/

HClib-Actor Documentation | Search | hclib_actor ☆0 ⑂1

HClib-Actor Documentation
Home
Background
Theory
Bulk Synchronous Parallel
Partitioned Global Address Space
Actor Model
Practice
OpenSHMEM
Bale
Summary
spmat
libgetput
Habanero-C Library (HClib)
Getting Started
Containers
Docker
Singularity
Clusters/Supercomputers
NERSC/ORNL/PACE
Writing HClib-Actor Programs

Bulk Synchronous Parallel

What is the bulk synchronous parallel model?

The Bulk Synchronous Parallel (BSP) model is one of the most popular parallel computation models.

The model consists of:

- A set of processor-memory pairs.
- A communication network that delivers messages in a point-to-point manner.
- Efficient barrier synchronization for all or a subset of the processes.

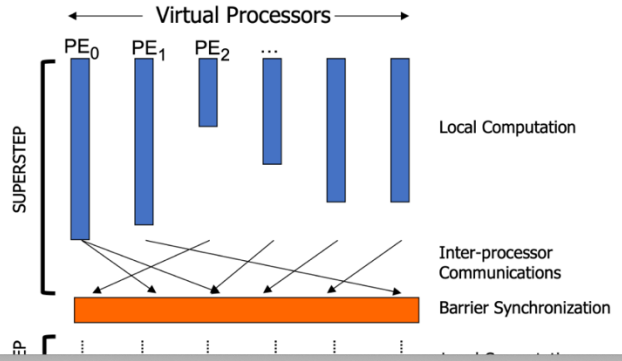


Table of contents
What is the bulk synchronous parallel model?
Single Program Multiple Data (SPMD) Programming
Further Readings

Georgia Tech

ACKNOWLEDGEMENT

This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), through the Advanced Graphical Intelligence Logical Computing Environment (AGILE) research program, under Army Research Office (ARO) contract number W911NF22C0083. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government.

Georgia Tech

IEEE/ACM CCGRID 2025
The 25th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing
Tromsø, Norway

ActorISx: Exploiting Asynchrony for Scalable High-Performance Integer Sort

Youssef Elmougy, Shubhendra Singhal, **Akihiro Hayashi**, and Vivek Sarkar

Habanero Extreme Scale Software Research Lab
Georgia Institute of Technology

Thank you for your attention!

Georgia Tech