Experiences Developing Regular and Irregular Applications on OpenSHMEM

- PGAS offers performance and programmability benefits for large-scale applications, more natural extension of shared-memory programming
- OpenSHMEM's symmetric heap offers high-level abstraction and efficient RDMA
- Evaluate effectiveness on a range of regular and irregular benchmarks, explore integrating its use with OpenMP and **AsyncSHMEM** for hybrid parallelism
- See poster titled "Scaling to Exascale: Intra- and Inter-node Asynchronous Tasking in OpenSHMEM" for motivation and implementation of AsyncSHMEM, a framework for combining task parallel programming with OpenSHMEM

AsyncSHMEM Fork-Join vs. Offload Runtime

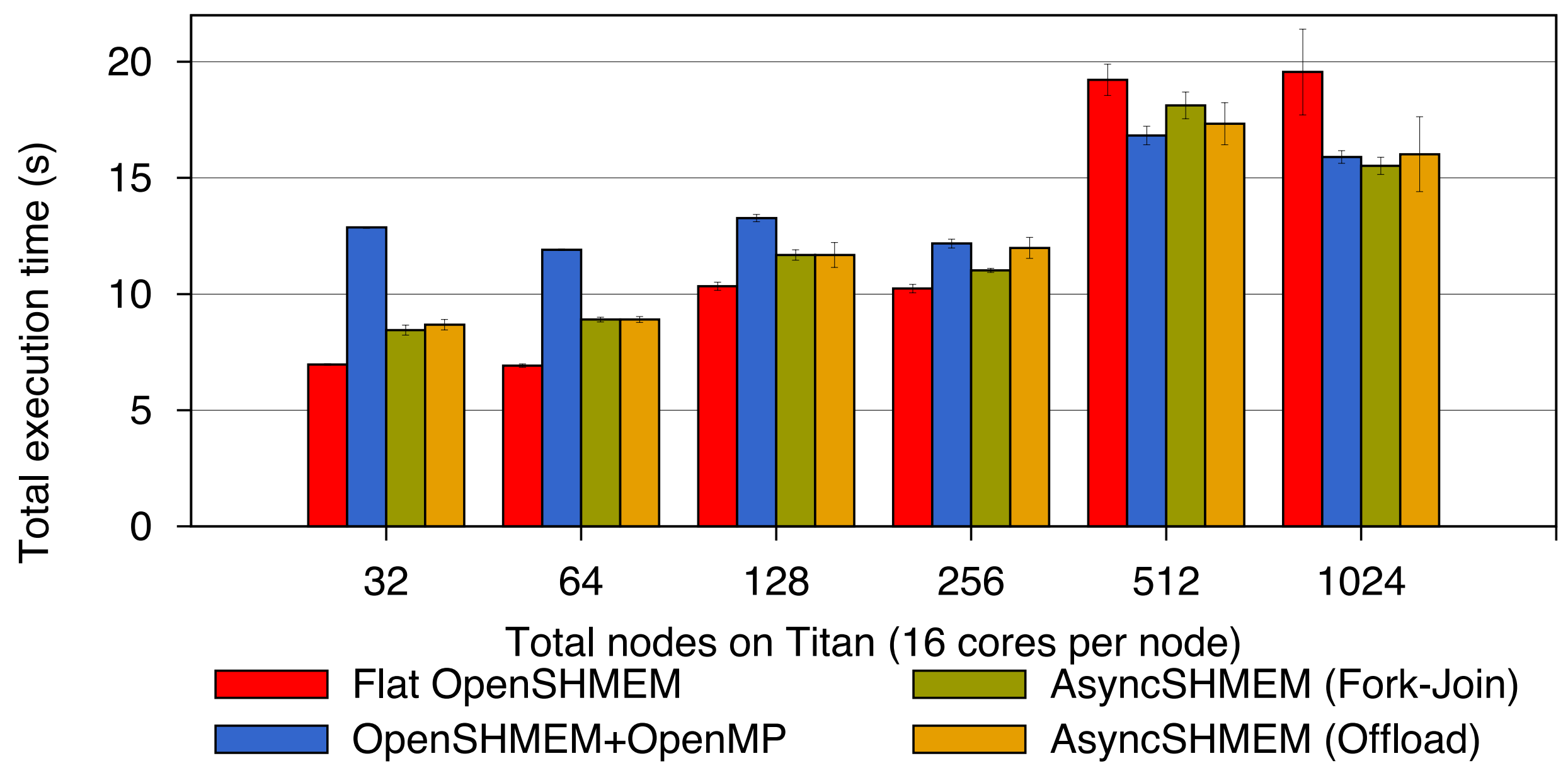
- **Fork-Join**: simple integration of task-parallel runtime w/ OpenSHMEM, includes tasking API extensions, no communication permitted in parallel regions
- **Offload**: offload communication work to communication worker thread, more general combination of OpenSHMEM and parallelism

Evaluation Platform

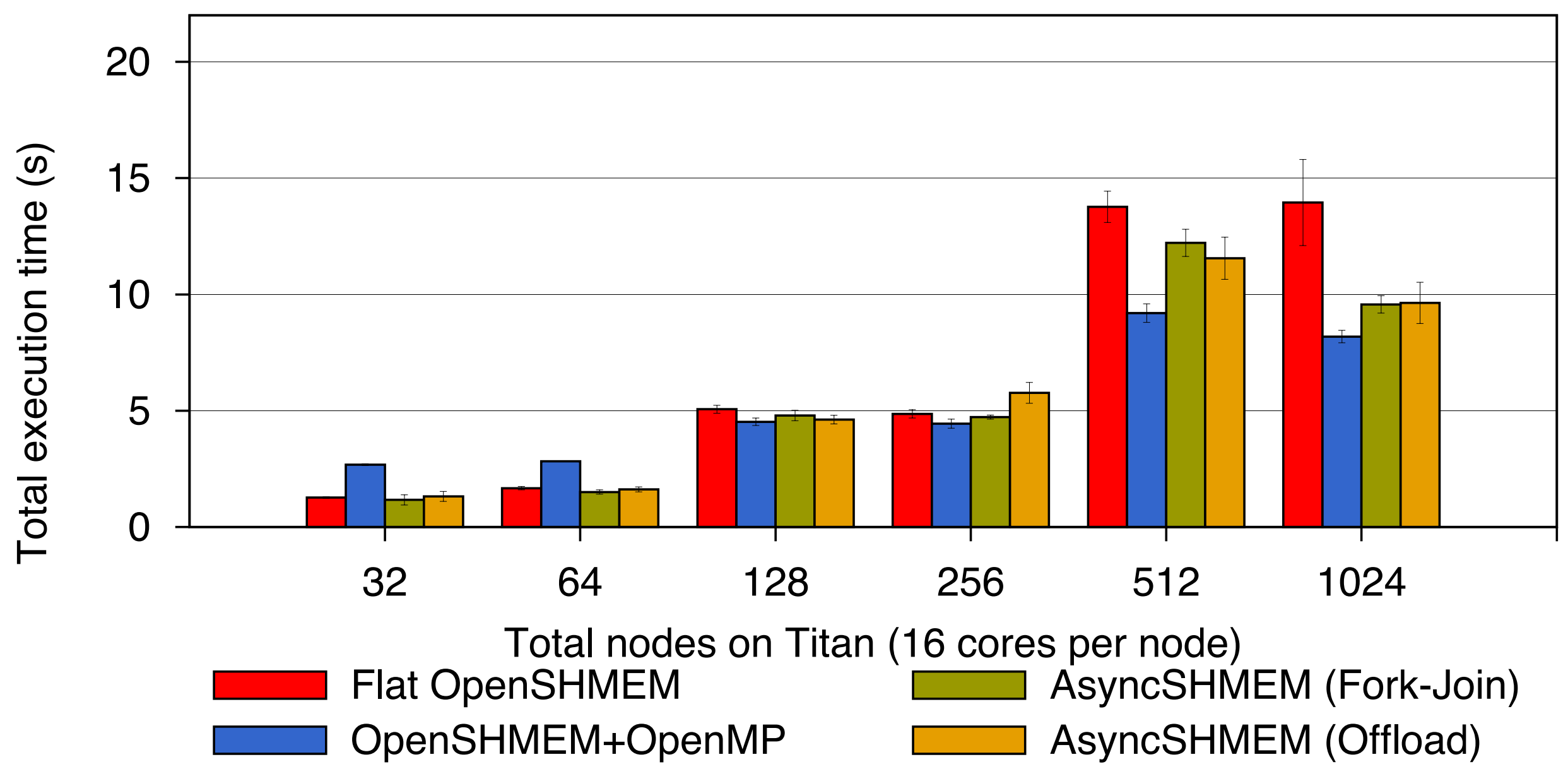
Titan @ ORNL : 16-core AMD, 32 GB DRAM. No experiments with GPUs.

ISx

- Distributed Integer sort, regular parallelism, weak scaling with 2^{29} keys/node
- All-to-all to distribute keys among nodes, followed by local sort
- Demonstrate value added by hybrid parallelism on regular applications



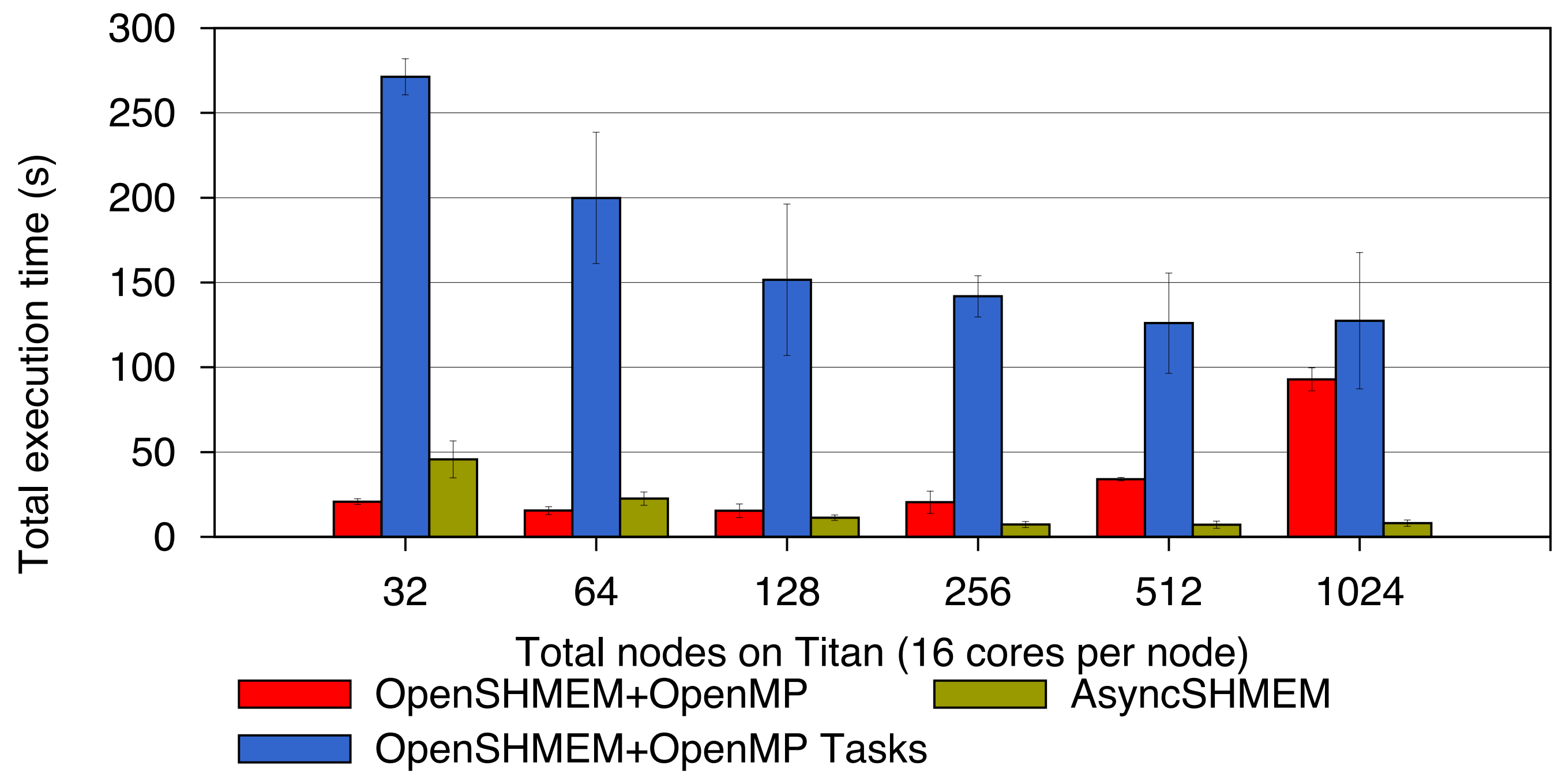
Total Execution Time



Total All-to-all Time

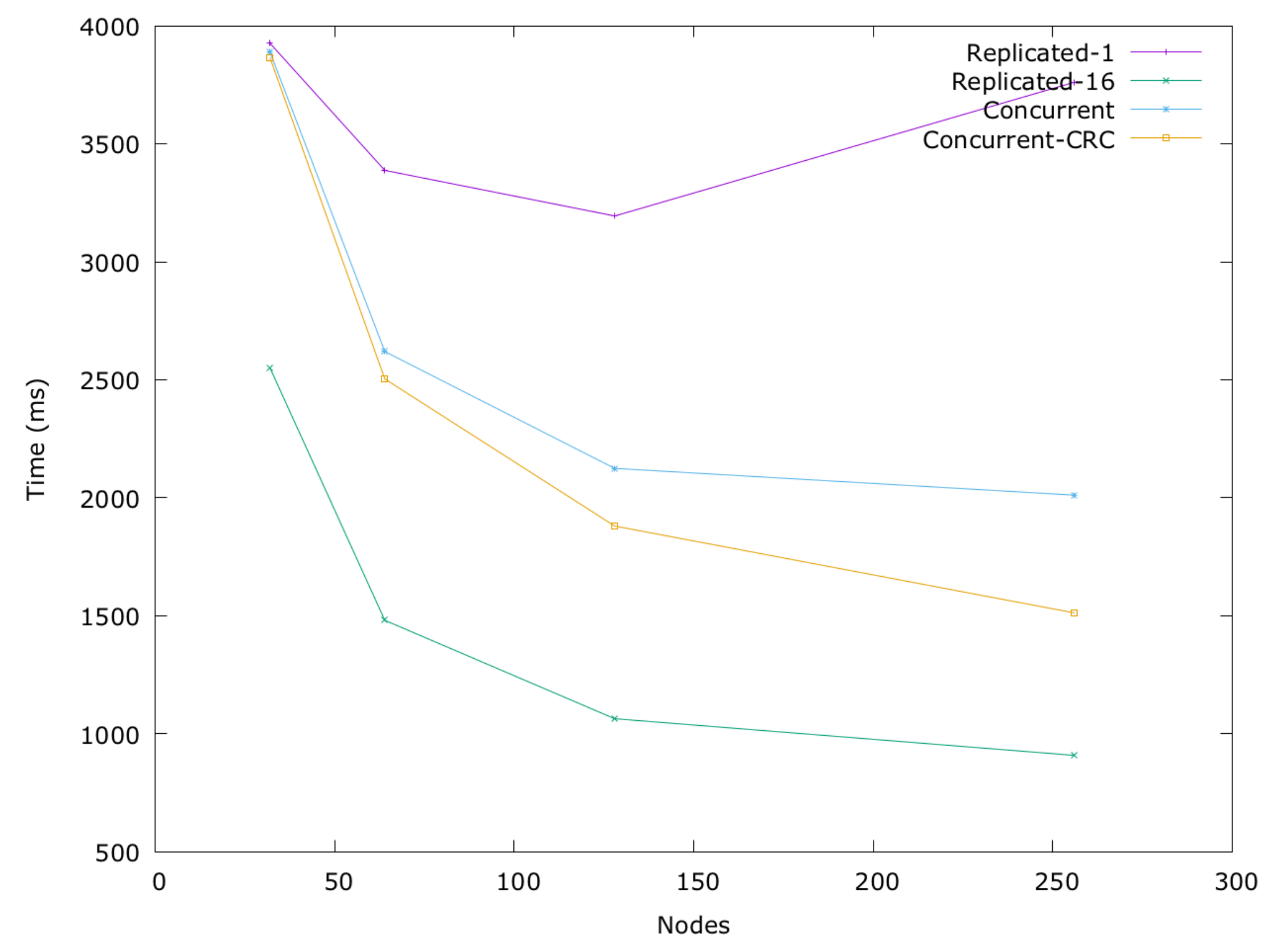
UTS

- Unabalanced tree sort, irregular parallelism, strong scaling on T1XXL
- Distributed and shared-memory dynamic load balancing
- Task-parallel programming with hand-coded scalability



Graph 500

- Distributed breadth first search, generally wavefront-based
- OpenSHMEM-based implementation (Concurrent-CRC) demonstrates highest sclability of flat implementations



Max Grossman (jmg3@rice.edu), Vivek Kumar,
Zoran Budimlic, Howard Pritchard, Jeff Kuehn, Vivek Sarkar