# Asynchronous Many-Task Programming with UPC++

PGAS

- HUPC++ (HabaneroUPC++) is a compiler-free task-parallel and PGAS library
- Integrates the UPC++ PGAS library and HClib dynamic tasking libraries, with extensions to both
- Two approaches: SPMD Task Region and Dynamic Task Parallel

SPMD Task Regions

- Applications are segmented into code regions of asynchronous tasks and communication, separated by global computation & communication barriers
- Defers comm. to master thread when it reaches end of SPMD region

Distributed Load-Balancing

- asyncAny API expresses locality-flexible tasks, eligible for load-balancing across address spaces
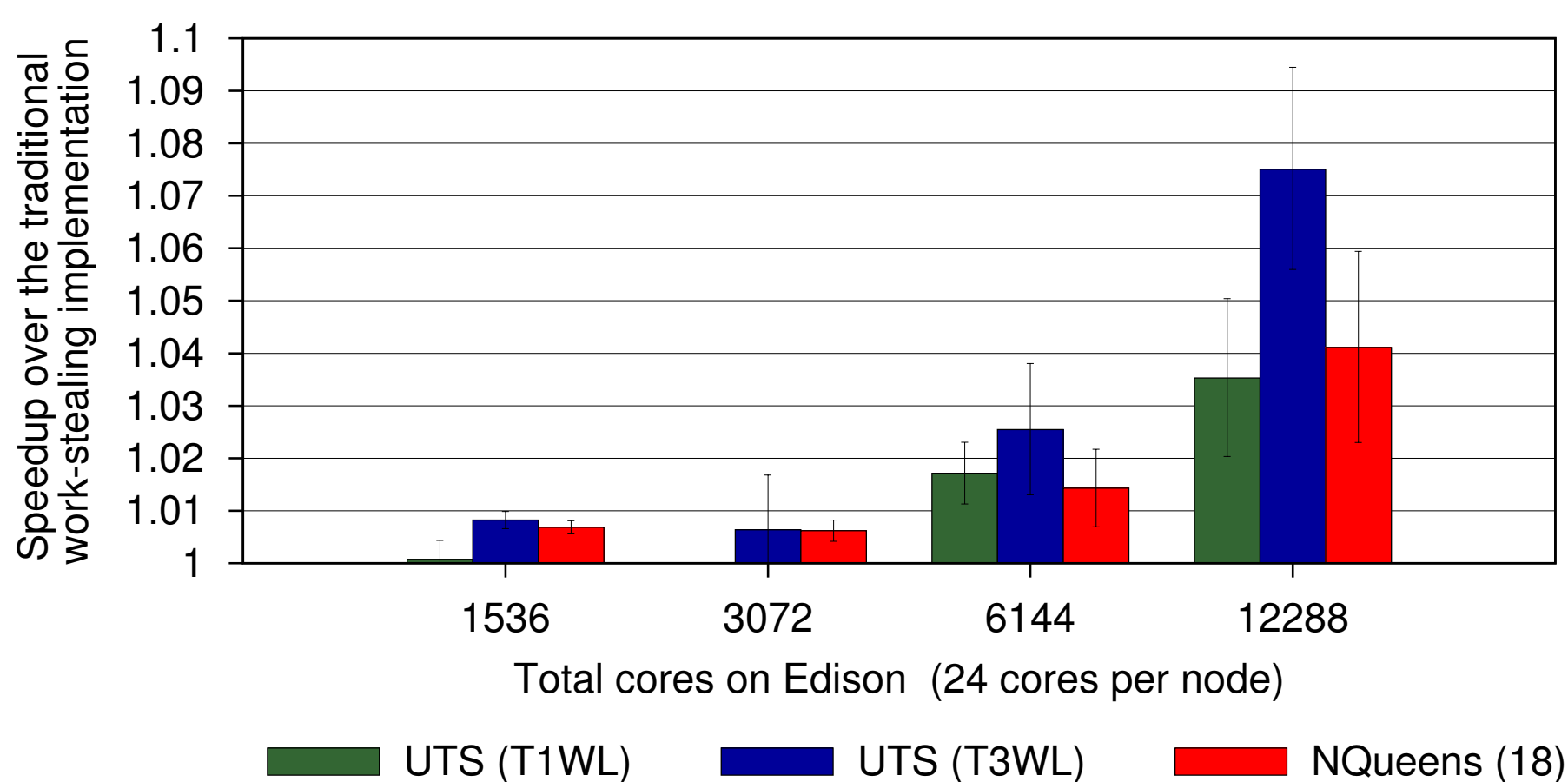- Task body is C++11 lambda
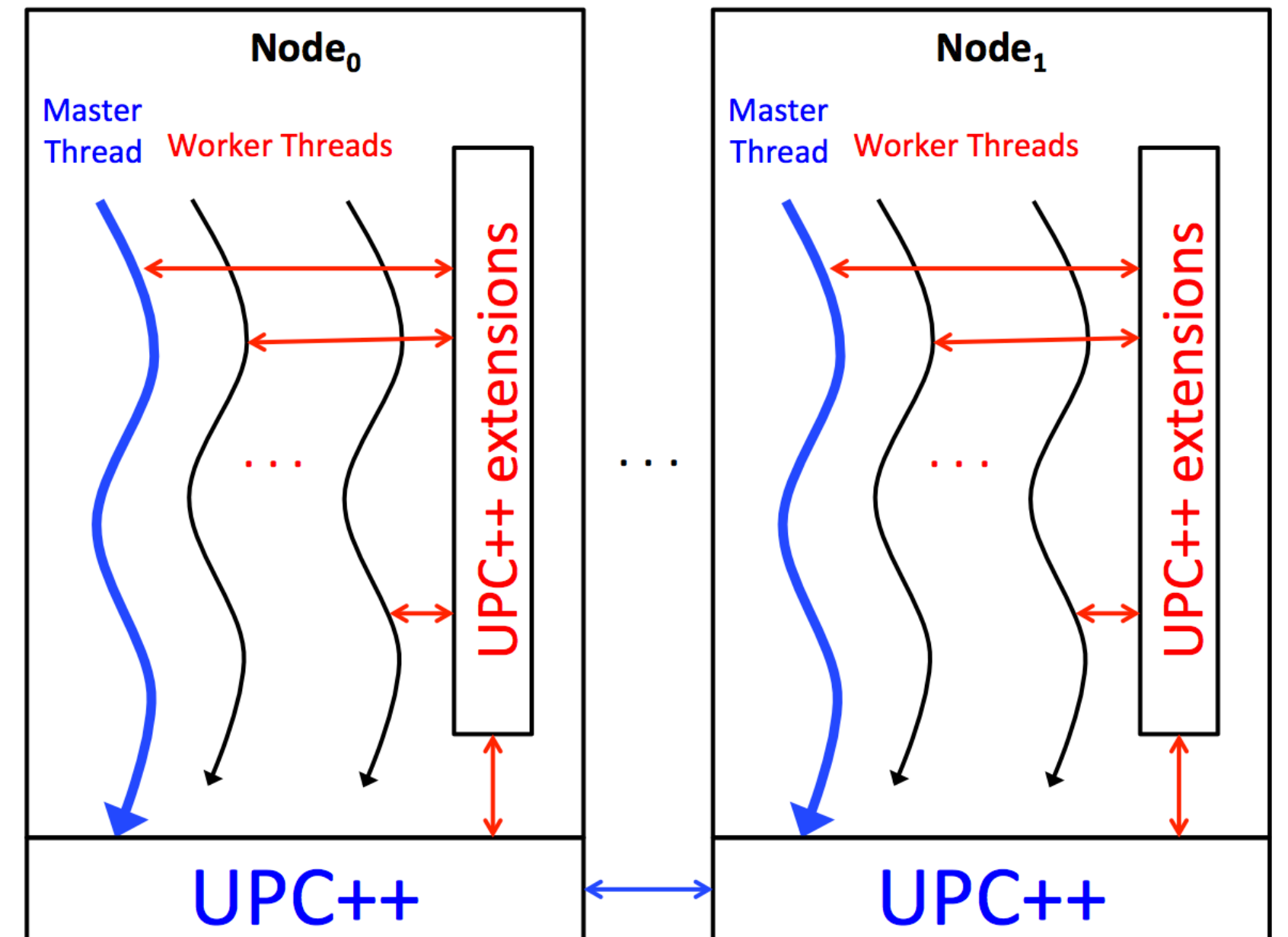
Dynamic Task Parallel

- Future-based = fewer implicit global barriers, more explicit dependency management
- Communication is funneled to single worker thread in tasking runtime
- Unified scheduling of heterogeneous workloads on single runtime (Tasking + MPI + CUDA + UPC++)
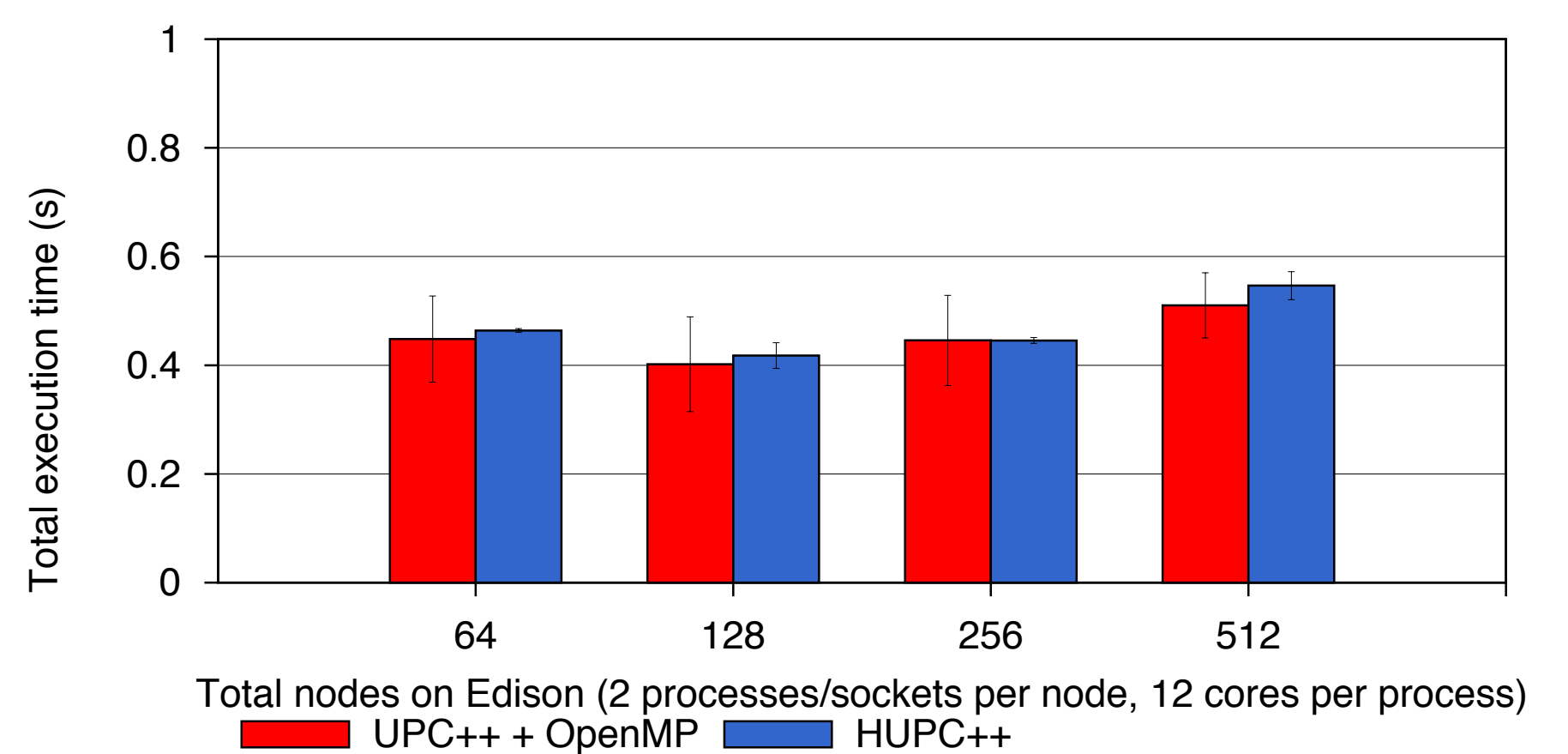
Performance Evaluation

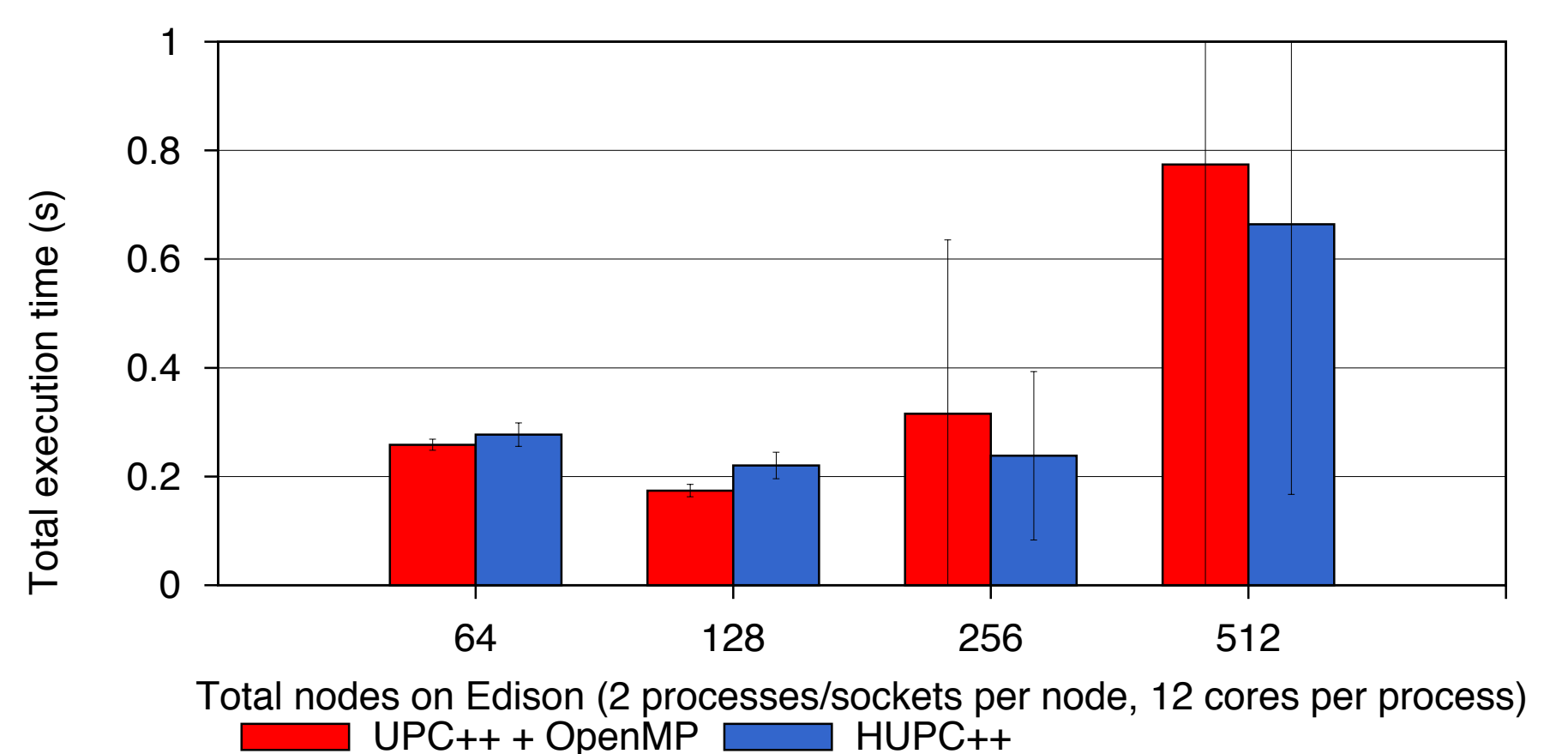Edison @ NERSC: Cray XC30, each node contains a 24 core Intel Ivy Bridge
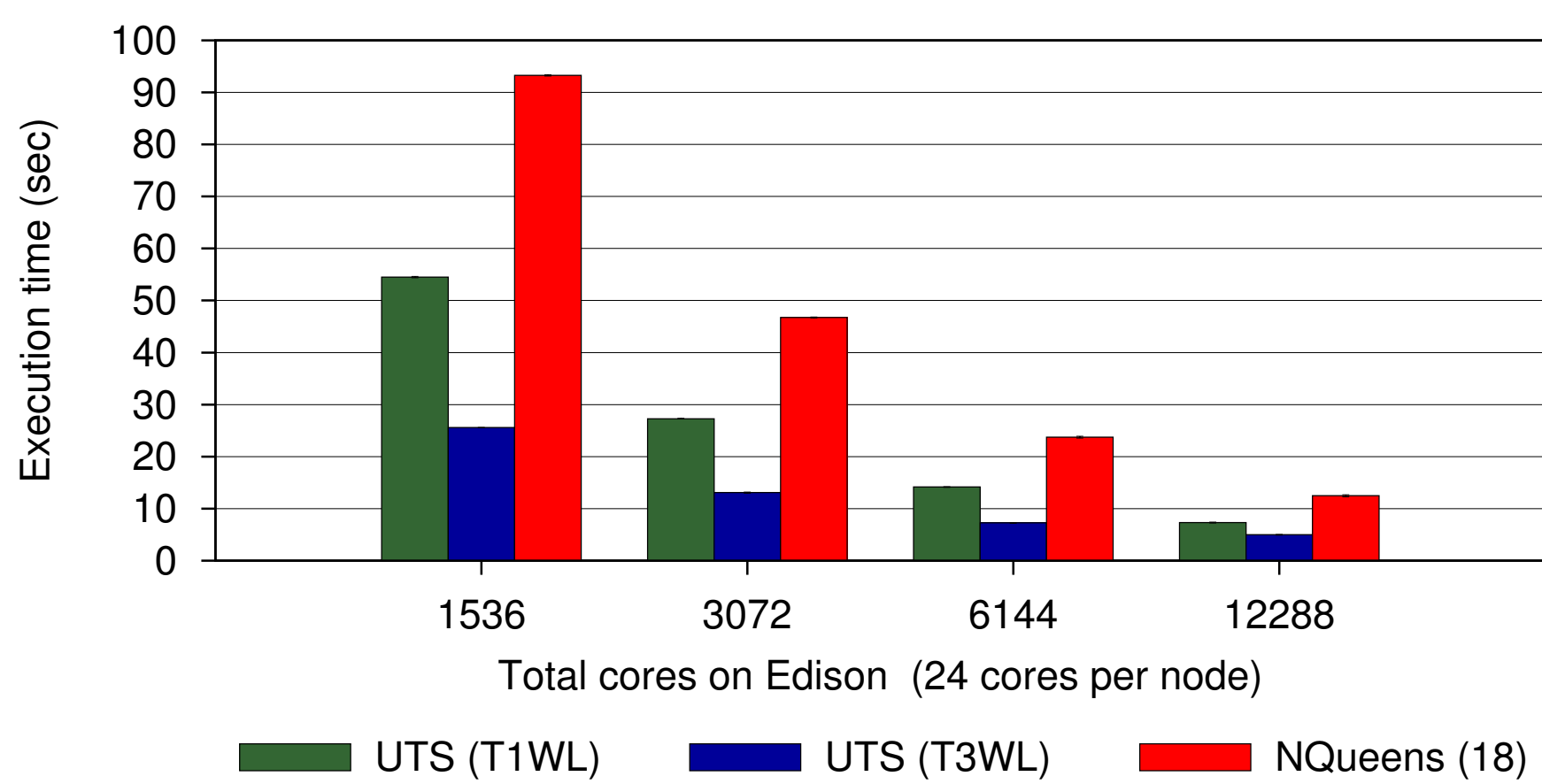
Scaling of traditional work-stealing implementations in HUPC++

Reduced variance for HPGMG-Solve on HUPC++ runtime

Speedup using novel distributed work-stealing implementation in HUPC++, elimination of inter-node failed steals

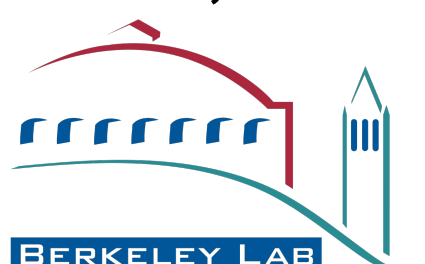Improved scalability for HPGMG-Build on HUPC++ runtime

Max Grossman (jmg3@rice.edu), Vivek Kumar, Zoran Budimlic, and Vivek Sarkar

Publications

1. V. Kumar, K. Murthy, V. Sarkar, and Y. Zheng, "Optimized Distributed Work-Stealing", IA^3 2016.
2. V. Kumar, Y. Zheng, V. Cave, Z. Budimlic, and V. Sarkar, "HabaneroUPC++: A Compiler-free PGAS Library", PGAS 2014.