Berkeley UPC on the BlueGene/P

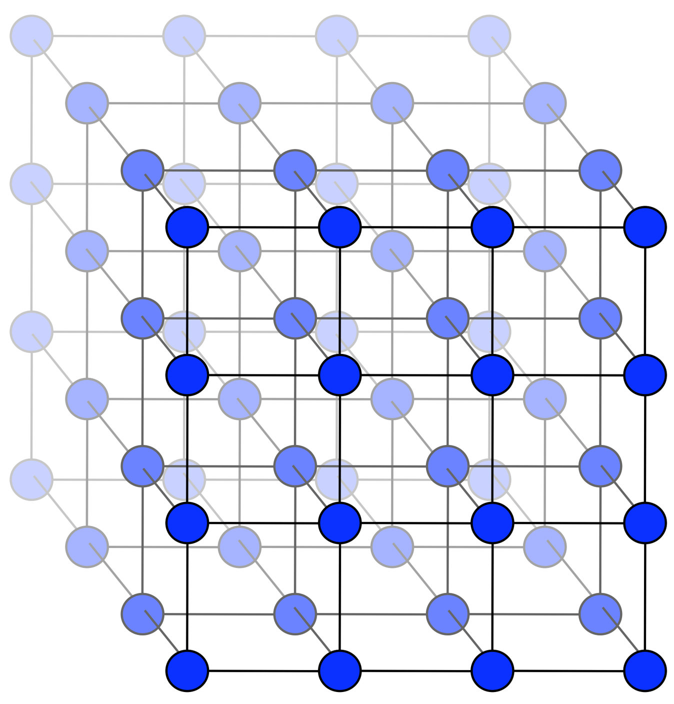
by Rajesh Nishtala, Paul Hargrove, and Dan Bonachea

Open Research Questions

- How well does the PGAS programming model scale to thousands and hundreds of thousands of nodes
- What new techniques must be employed to create scalable runtime systems for PGAS languages
- What is the effectiveness of non-blocking communication and overlap at large scale?

BlueGene/P Overview

- Each compute node has 4 cores running at 850 MHz and 2 GB memory.
- Peak performance per node:13.6 GFlop/s
- Peak Memory bandwidth: 13.6 GB/s
- Compute Nodes interconnected by many networks
  - Fast Collective Network
  - Fast Barrier Network
  - 3D Torus for general communication
    - 6 full-duplex links @ 425 MB/s per link

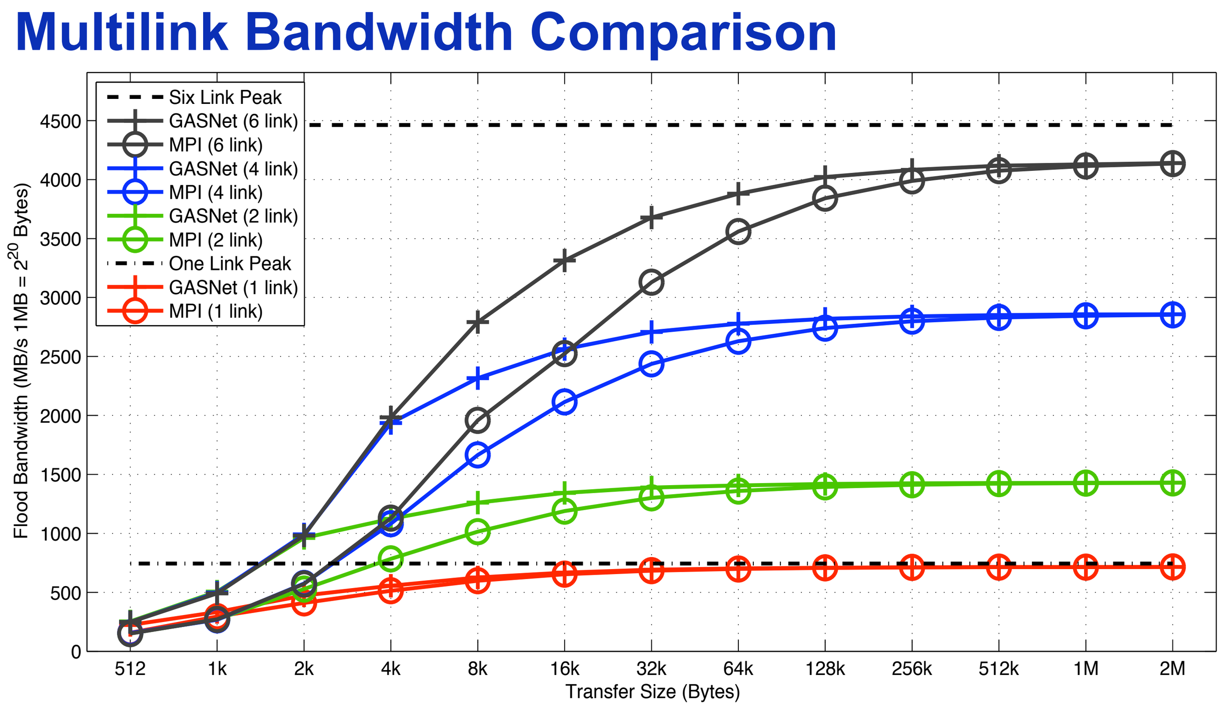


GASNet on BlueGene/P

GASNet is the portable high performance runtime layer for PGAS languages

- Currently used in Berkeley UPC, GCCUPC Titanium, Co-Array FORTRAN, and Chapel
- Provides high performance point-to-point communication primitives such as put/get
- Provides common collective operations that are designed for one-sided communication
- Often a better semantic match to modern network hardware and thus can realize better performance than MPI

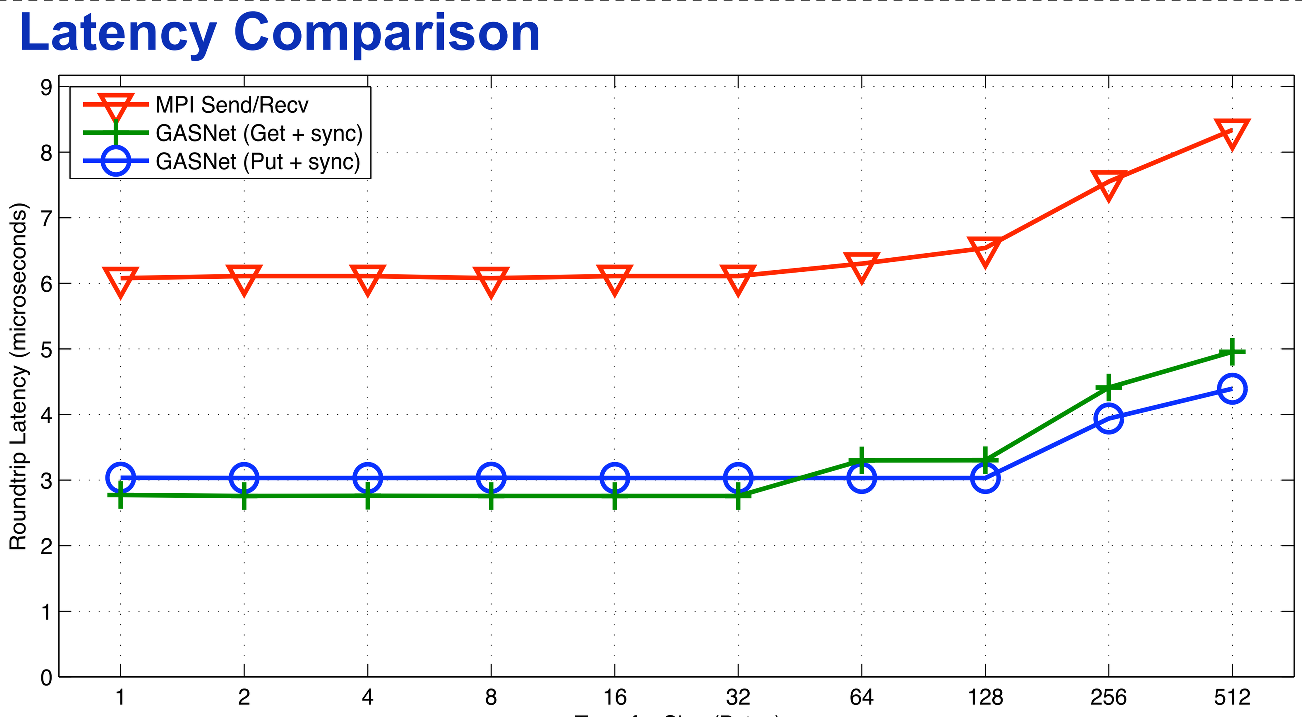
Multilink Bandwidth Comparison



Measures Bidirectional flood bandwidth across a varying number of links in the torus

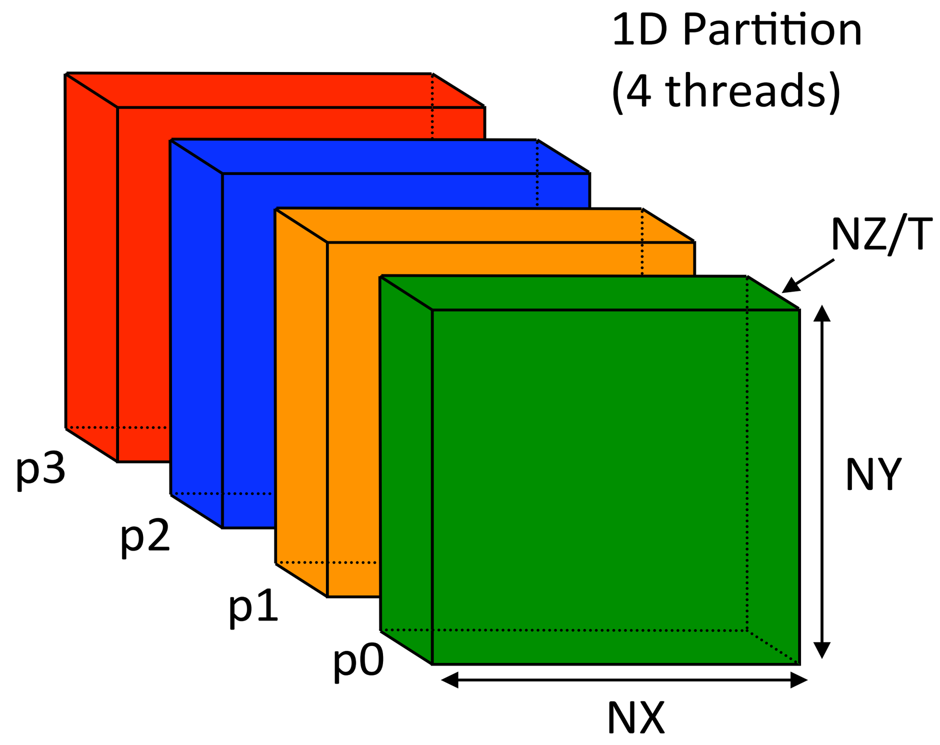
- Exploits communication/communication overlap
- GASNet outperforms MPI in midrange message sizes (512-64kBytes)

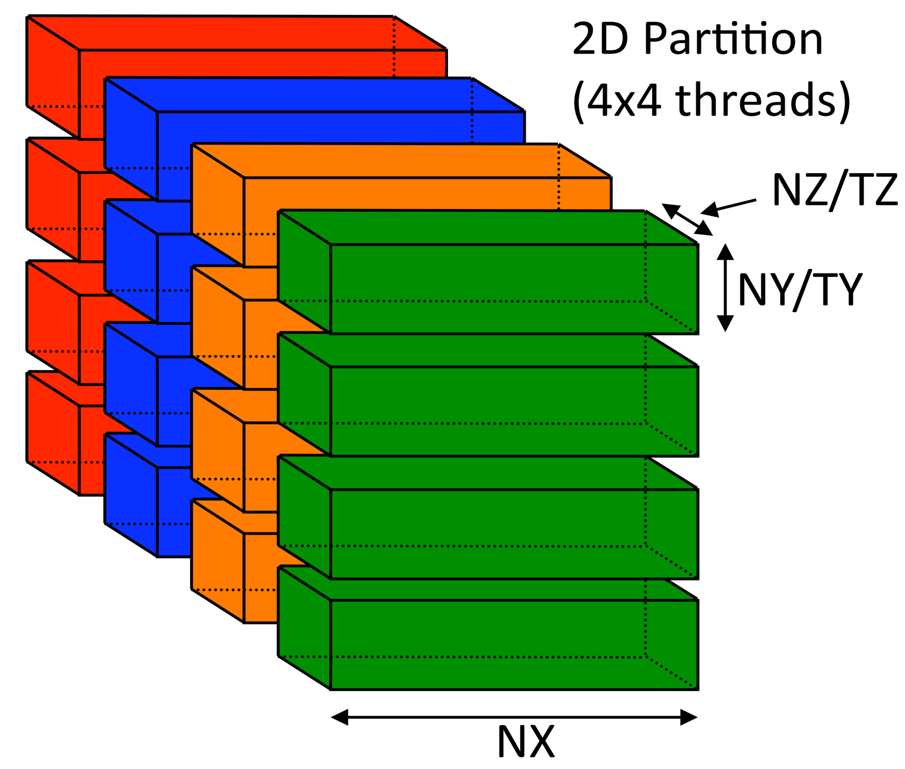
Latency Comparison



- Measure Ping-Ack latency
  - For MPI time initiator to send a message and respond with a 0 byte ack
  - For GASNet time to issue a put (w/ remote completion notification) or a get
- GASNet latency is about half that of MPI
  - Has implications for lower software overhead and thus better overlap potential

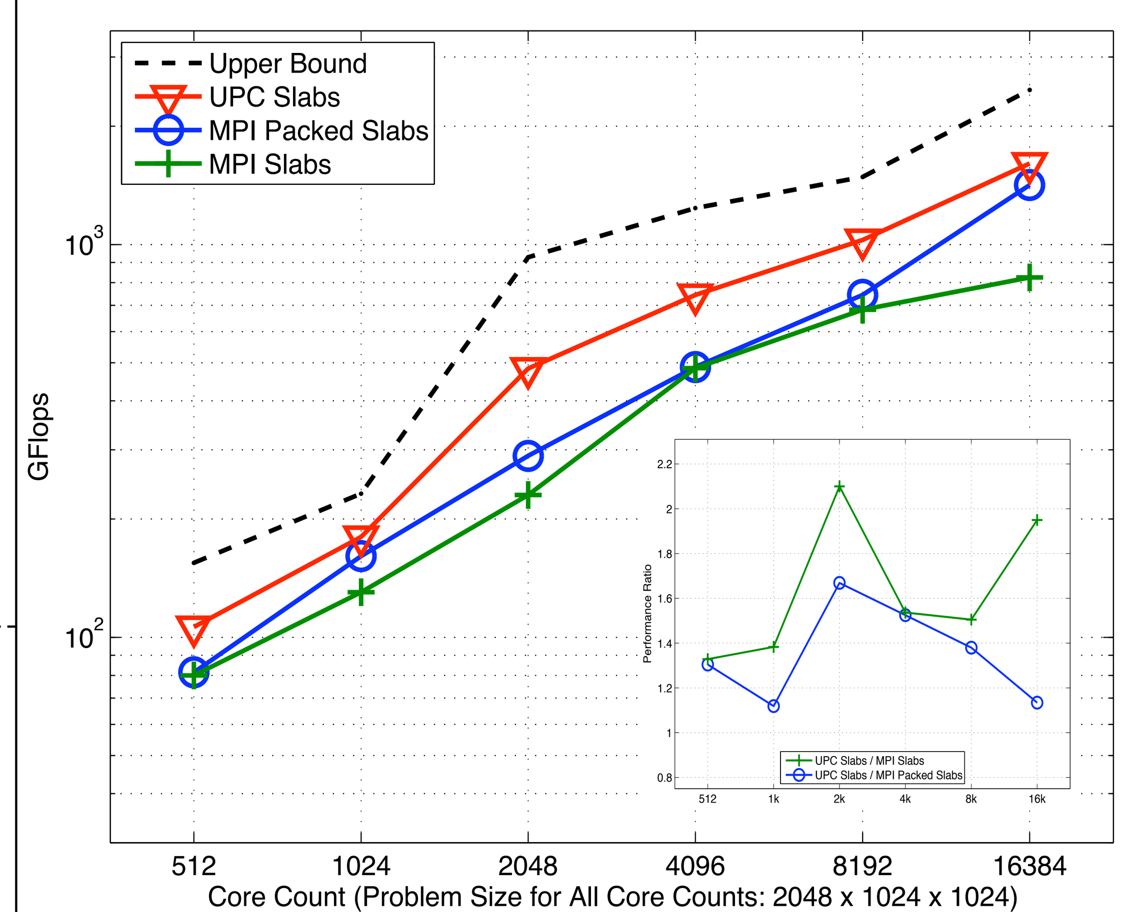
NAS FT Benchmark Results





- Benchmark computes a large 3D FFT
- Requires a large All-to-all transpose communication operation.
  - Communication intensive benchmark limited by the bisection bandwidth of the network
  - Our previous work demonstrated that nonblocking communication can lead to significant performance improvements
  - We explore how these techniques scale to thousands of processors on the BlueGene/P
- We consider two algorithms
  - Packed Slabs:
    - Separates computation and communication into two distinct phases
    - Pack the data to allow larger messages and thus better bandwidth
    - Keeps either computation or communication system idle
  - Slabs:
    - Initiate communication earlier and overlap transposes with the computation
    - Reduced message size could adversely affect communication performance

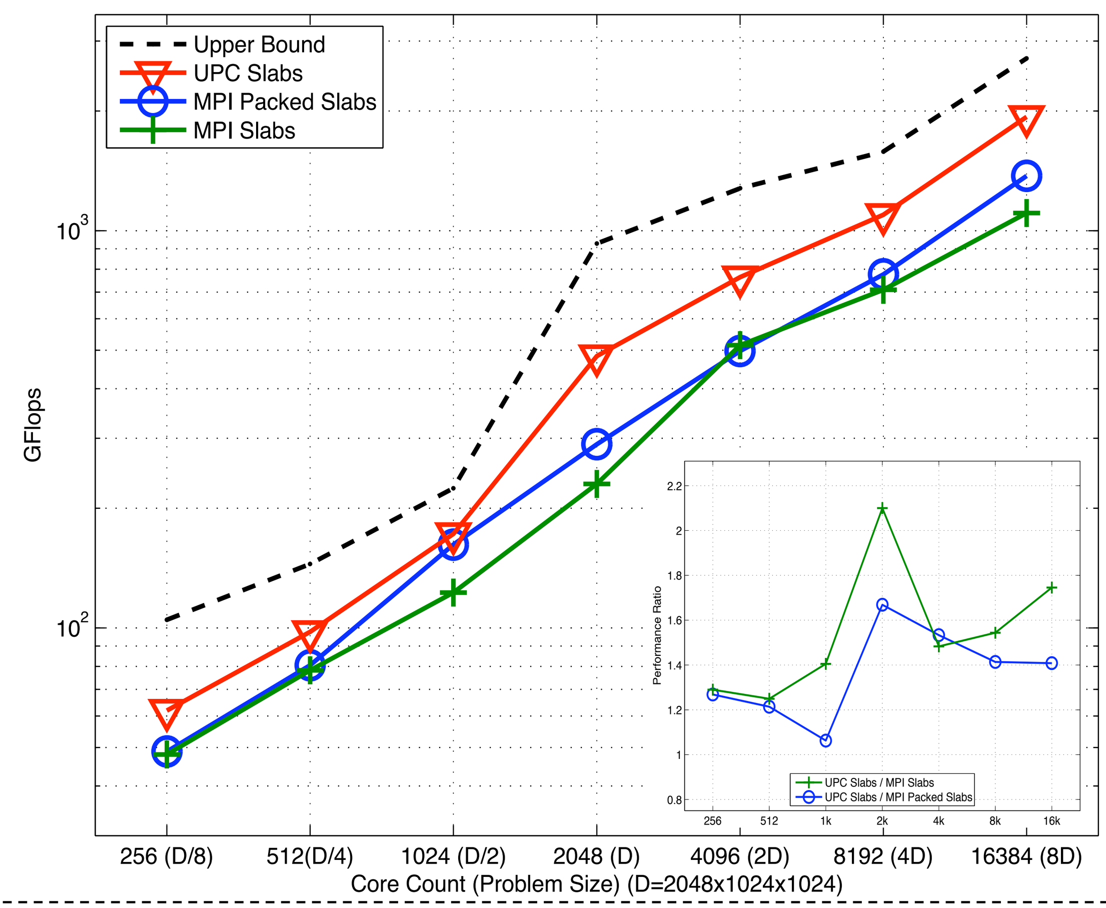
Strong Scaling Performance Results



- Keep the problem size fixed and vary the number of processors
- Overhead associated with overlapping communication and computation outweighs benefits with MPI
  - As core count grows message sizes become too small to effectively overlap communication
- UPC Slabs outperforms MPI Slabs due to GASNet's lower overheads and higher efficiency at mid-range message sizes
- UPC Slabs also outperforms MPI Packed Slabs by 13% @ 16k cores

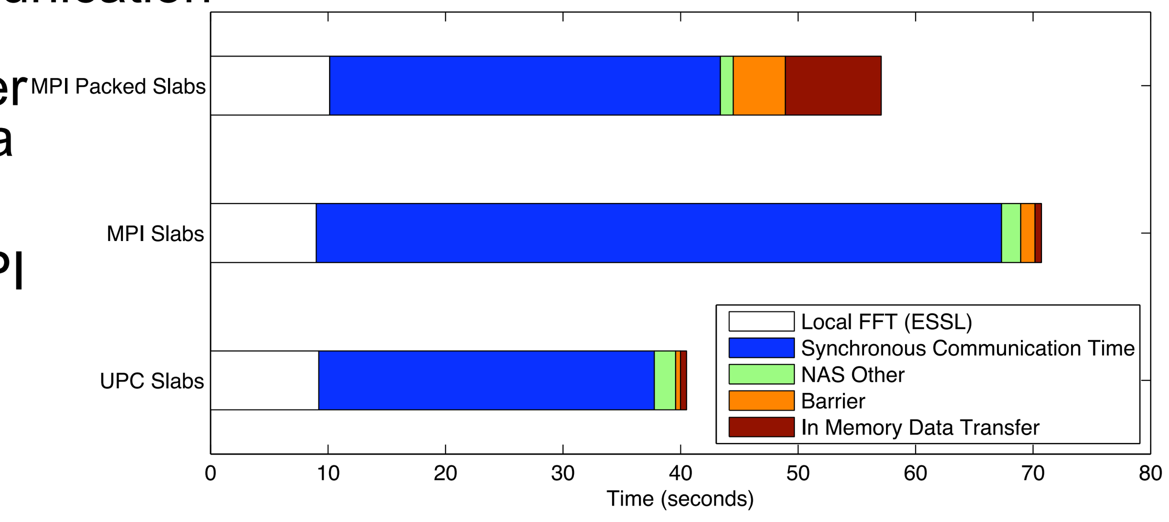
Weak Scaling Performance Results

- Scale problem size with core count
- Message sizes vary less with core count allowing consistently better performance
- UPC can better overlap communication compared to MPI as shown by MPI Slabs v. UPC Slabs.
- UPC Slabs outperforms MPI Packed Slabs to yield a 40% improvement in overall application performance



Performance Breakdown @ 16k cores (weak scaling)

- Performance is dominated by communication
- Packed Slabs algorithm incurs higher costs associated with in memory data movement for packing
- Performance difference between MPI Slabs and UPC Slabs illustrates performance advantages of UPC



Future Work

- Test at larger scale and other applications
- Leverage BlueGene hardware collectives in GASNet/UPC
- Explore techniques to better schedule communication for the 3D torus