Berkeley UPC

http://upc.lbl.gov

PGAS

Overview

- A portable and high-performance UPC implementation, compliant with UPC 1.2 spec
- Features:
  - High performance UPC Collectives
  - Extensions for performance and programmability
    - Non-blocking memcpy functions
    - Semaphores and signaling put
    - Value-based collectives
    - Atomic memory operations
    - Hierarchical layout query
    - Localization (castability) queries
    - Collective deallocation functions
  - Compiler and runtime optimizations for application scalability
- Open Source Software (Windows/Mac/UNIX), installation DVD available at PGAS booth (#124)

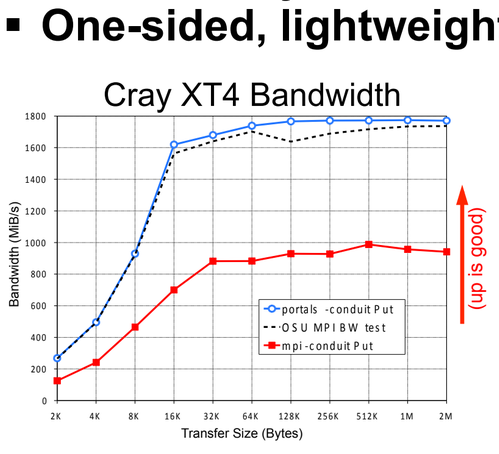
Portable Design

- Layered design, platform-independent code gen
- Supports wide range of SMPs, clusters and MPPs
  - x86, Itanium, Opteron, Alpha, PPC, MIPS, ARM, PA-RISC, SPARC, X1, SX-6, Cray XT and XE, IBM Blue Gene, …
  - Linux, {Free,Net,Open}BSD, Tru64, AIX, IRIX, HPUX, Solaris, MS Windows, Mac OS X, Unicos, SuperUX, …
  - Pthreads, Myrinet, Quadrics Elan 3/4, InfiniBand, IBM LAPI, Dolphin SCI, MPI, Ethernet, Cray X1 / SGI Altix shmem, Cray XT Portals, Cray XE Gemini (beta), IBM BG/P DCMF

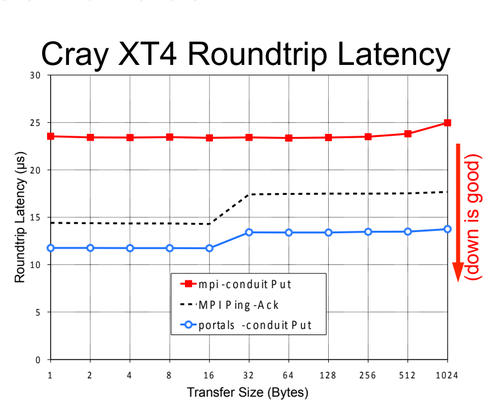
BUPC Runtime + GASNet

- Well-documented runtime interface, multiple UPC compilers (Berkeley UPC and Intrepid GCC/UPC)
- Debugging and tracing support
  - Performance Instrumentation Support (GASP)
  - Supports Parallel Performance Wizard (PPW)
  - Detailed communication tracing support
  - TotalView debugger support
- Interoperability with other programming env:
  - UPC calls to/from C, C++, Fortran, MPI
- Berkeley GASNet used for communication:
  - Performance from inline functions, macros, and network-specific implementations
- Optimized Collective ops
- High-performance communication
  - Consistently matches or outperforms MPI
  - One-sided, lightweight semantics

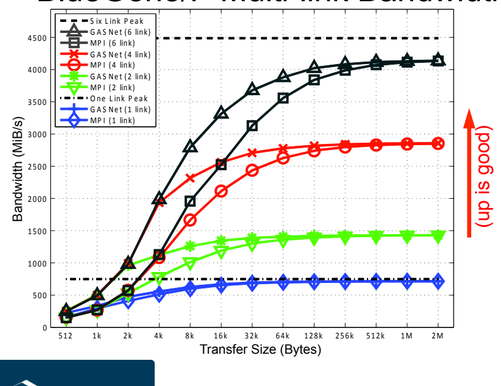
Cray XT4 Bandwidth



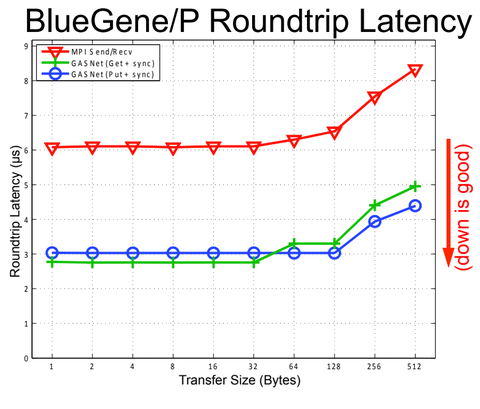
Cray XT4 Roundtrip Latency



BlueGene/P Multi-link Bandwidth



BlueGene/P Roundtrip Latency



UPC-to-C Translator

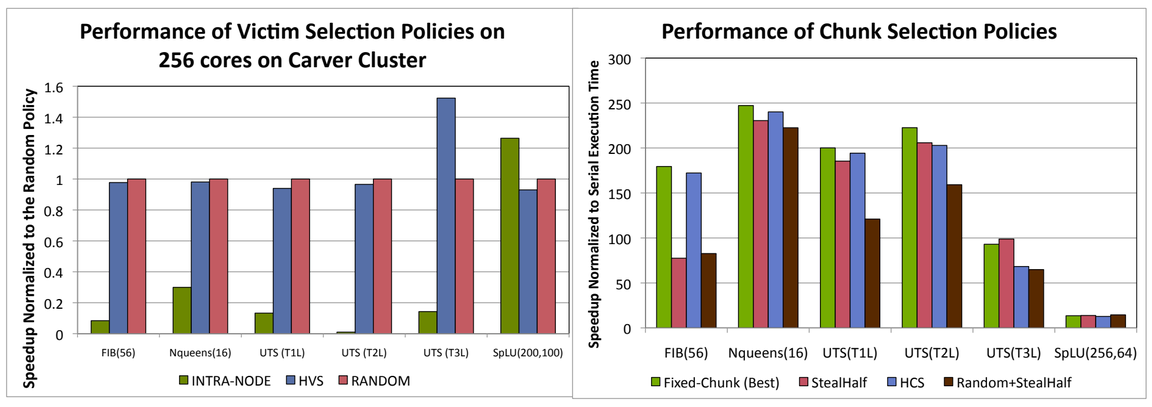
- Source-to-source translator, based on Open64
- Enhances programmer productivity through static and dynamic optimizations: compiler, runtime, communication libraries

Performance Portability: System, Scale, Load

- Compile time message vectorization and strip-mining
- Runtime Analysis: communication instantiated at runtime based on system specific performance models
- Performance models designed to take system scale and load into account

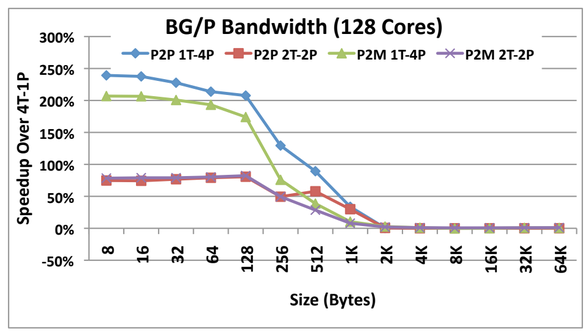
Dynamic Tasking in UPC

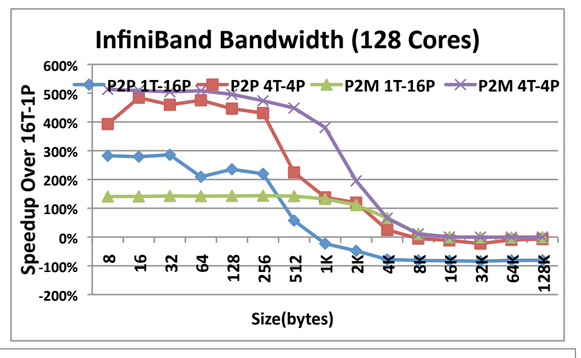
- Newly released UPC task library
- Task is defined as a task function + pointers to in/out
  - void *task_func* (void *in, void *out) { /* *task body* */ }
  - Tasks are stored in the global task queue
- Task programming models
  - Parallel-for parallelism and Fork-Join parallelism
  - Dependent task graph with task synchronization
- Performance improvement by dynamic load balancing
- Load balancing exploits locality
  - Hierarchical chunk selection
  - Hierarchical victim selection
- For more information – http://upc.lbl.gov/task.shtml



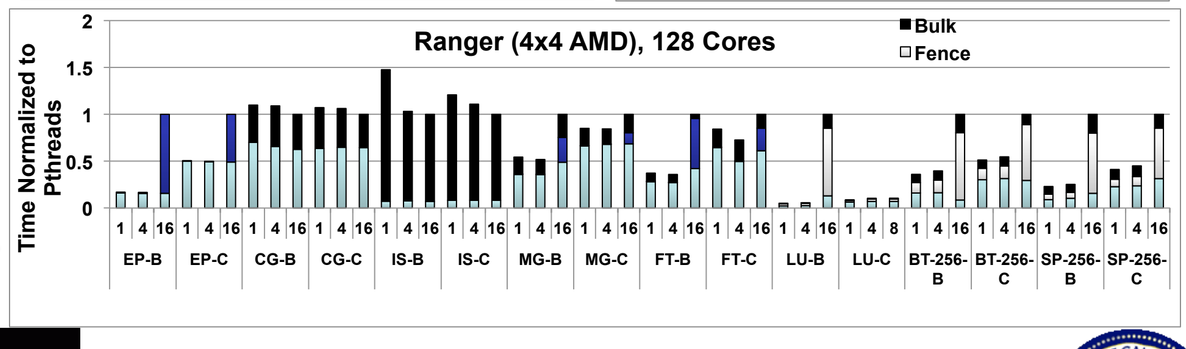
Hybrid "Task" Parallelism

- BUPC allows programs to use arbitrary combinations of Pthreads and Processes with shared memory
- Mixing Pthreads and Processes is required for:
  - Interoperability with MPI and non thread-safe libraries
  - Hybrid/hierarchical parallelism (for best performance)
- PSHM layer – Process SHared Memory
  - Shared memory comms through POSIX, SYSV or mmap()
  - Shared memory "network" for Active Messages support
  - Hybrid processes/pthreads execution
- Hybrid always performs best





Lower is Better



BERKELEY LAB
Lawrence Berkeley National Laboratory

© 2011, Lawrence Berkeley National Laboratory

BERKELEY UPC
UNIFIED PARALLEL C

THE UNIVERSITY OF CALIFORNIA BERKELEY 1868