

Overview

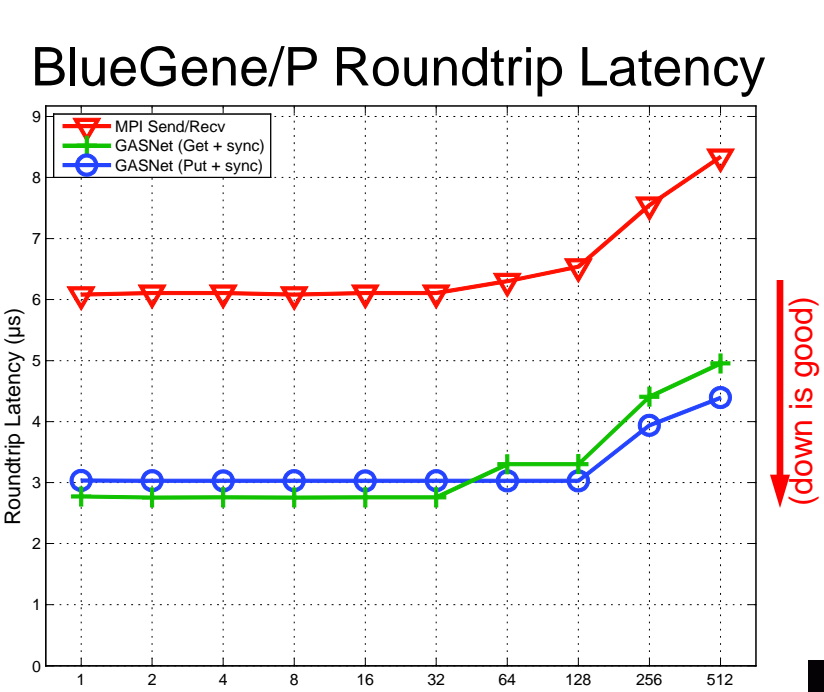
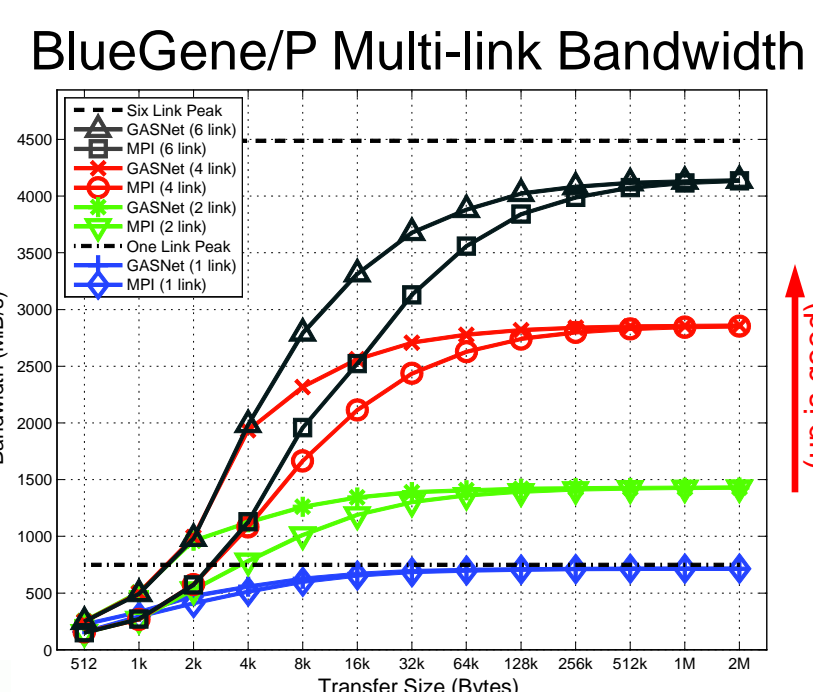
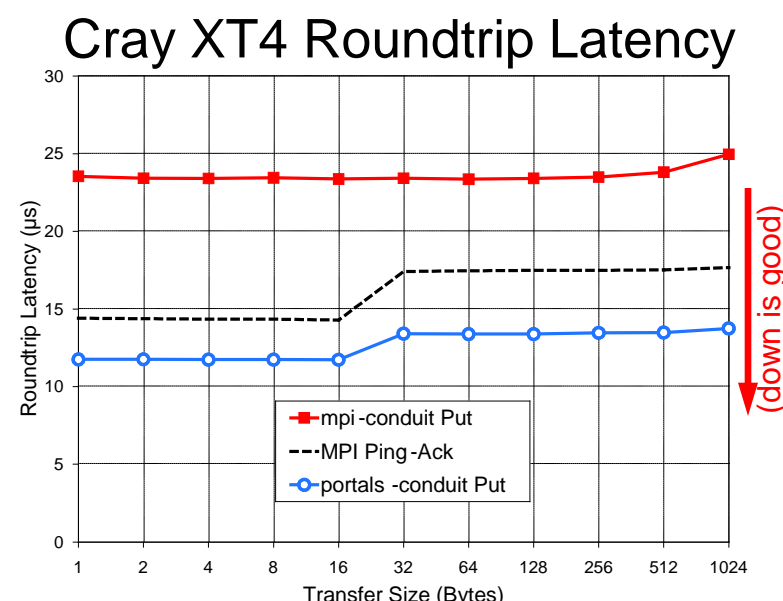
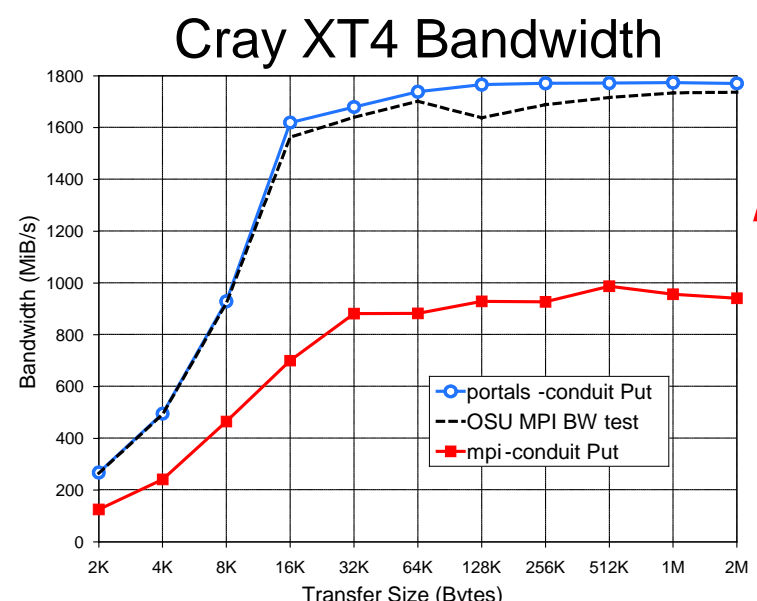
- A portable and high-performance UPC implementation, compliant with UPC 1.2 spec
- Features:
 - High performance UPC Collectives
 - Extensions for performance and programmability
 - Non-blocking memcpy functions
 - Semaphores and signaling put
 - Value-based collectives
 - Atomic memory operations
 - Hierarchical layout query
 - Localization (castability) queries
 - Compiler and runtime optimizations for application scalability
- Open Source Software (Windows/Mac/UNIX), installation DVD available at PGAS booth (#136)

Portable Design

- Layered design, platform-independent code gen
- Supports wide range of SMPs, clusters and MPPs
 - x86, Itanium, Opteron, Athlon, Alpha, PowerPC, MIPS, T3E, PA-RISC, SPARC, X1, SX-6, Cray XT, IBM Blue Gene, ...
 - Linux, FreeBSD, NetBSD, 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, 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

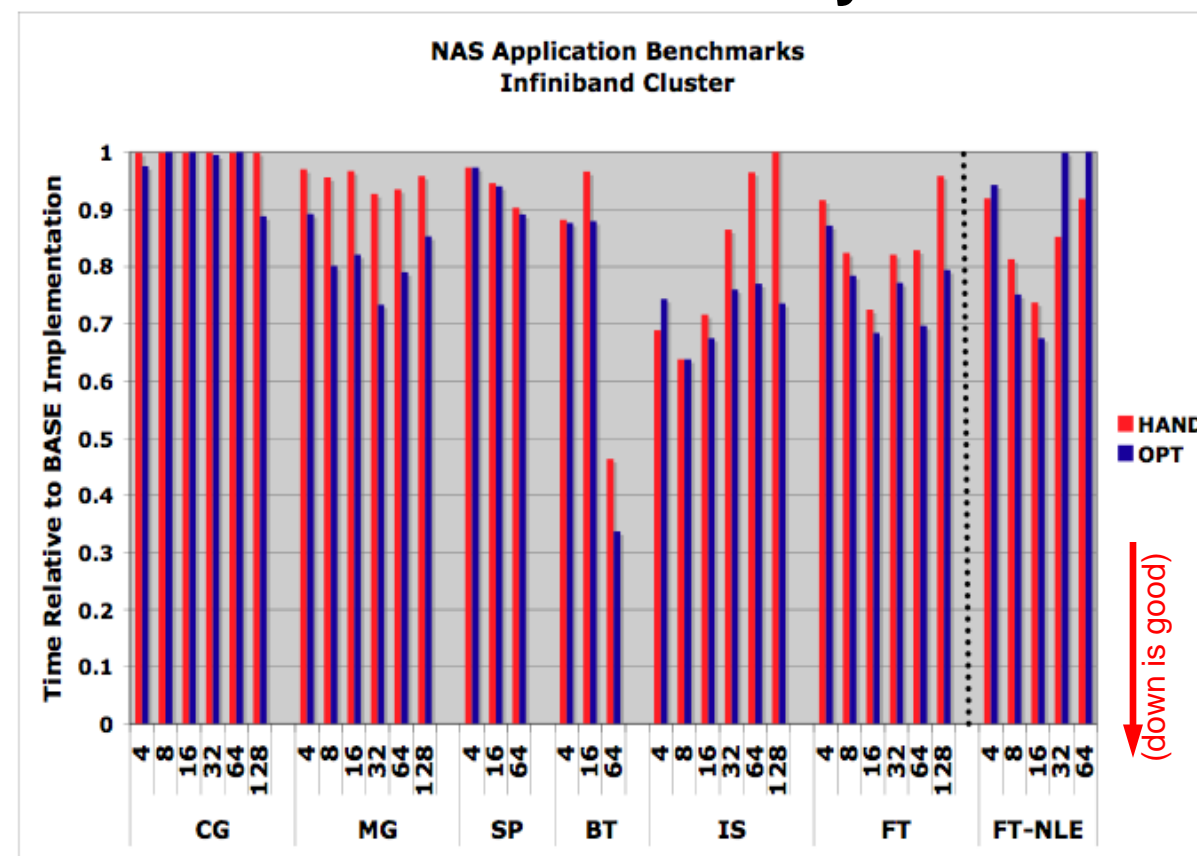


UPC-to-C Translator

- 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



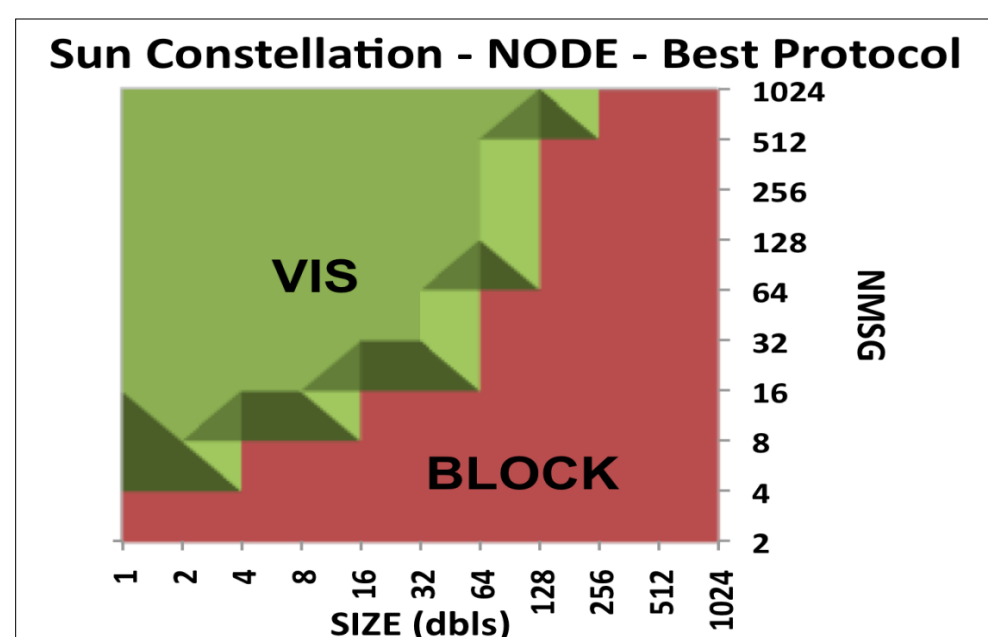
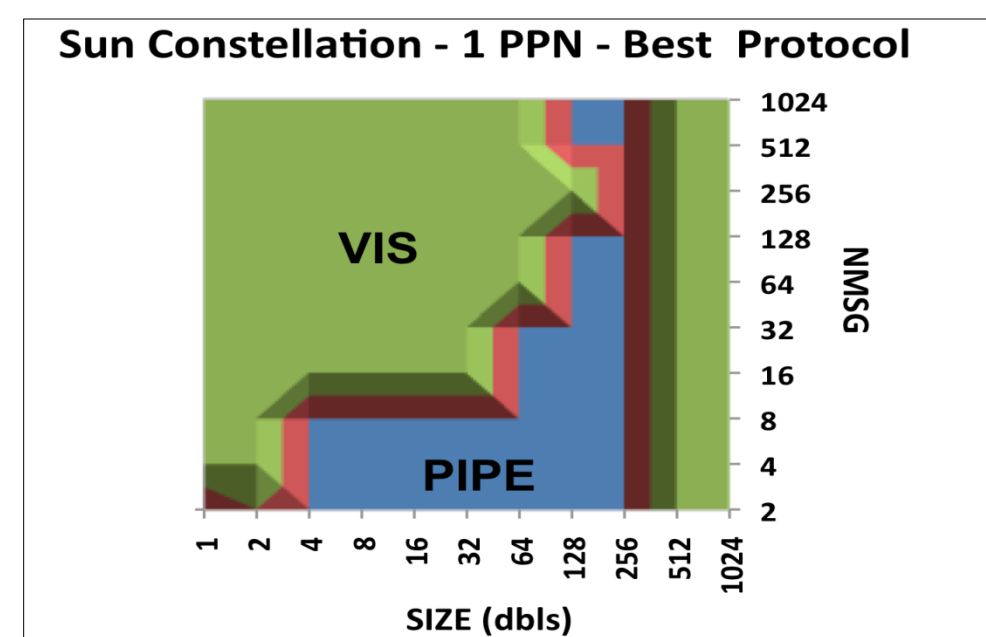
Communication dynamically instantiated using either Put/Get or VIS calls

Overall improved application scalability and programmer productivity

Runtime Optimization of Vector Operations

Scatter-gather operations are widely used in applications (boundary exchange)

- Multiple implementations
 - blocking, pipeline, AMs, progress threads
- Modern NICs are under-provisioned for multicore
- Dynamic approach required
 - Performance models for communication at scale pose a research challenge
 - Qualitative models: track derivatives instead of absolute performance
 - Determine first order performance parameters
 - Methodology for parameter space exploration at scale
 - Preserve performance ordering of implementations
 - Account for errors and partial exploration of parameter space



- Dynamic optimization approach implemented in the Berkeley UPC compiler
- Demonstrated improved performance and productivity

