The Performance and Productivity Benefits of Global Address Space Languages

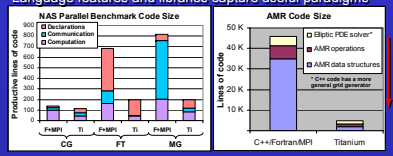
Christian Bell, Dan Bonachea, Kaushik Datta, Rajesh Nishtala, Paul Hargrove, Parry Husbands, Kathy Yelick

Titanium

- Titanium: GAS dialect of Java
  - Compiled to native executable:
    - No JVM or JIT
  - Highly portable & high performance
- High productivity language
  - All the productivity features of Java
    - Automatic memory management
    - Object oriented programming, etc
  - Built-in support for scientific computing:
    - n-D points, rectangles, domains
    - n-D domain calculus operators
    - Flexible & efficient multi-D arrays
  - High-performance templates
  - User-defined immutable (value) classes
  - Explicitly unordered n-D loop iteration
  - Operator overloading
  - Efficient cross-language support
- Allows for elegant and concise programs

Productivity

- Titanium reduces code size and development time
- Communication more concise than MPI, less error-prone
- Language features and libraries capture useful paradigms

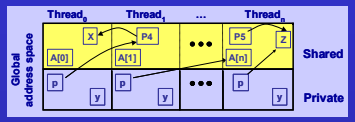


2-d Multigrid stencil code in Titanium

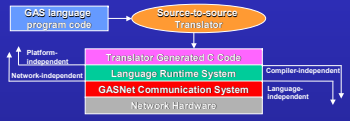
```
final Point<2> NORTH = [0,1], SOUTH = [0,-1],
               EAST = [1,0],  WEST = [-1,0];

foreach (p in gridA.domain())
  gridB[p] = S0 * gridA[p] +
             S1 * ( gridA[p+NORTH] + gridA[p+SOUTH] +
                    gridA[p+EAST] +  gridA[p+WEST] );
```

Global Address Space Languages

- Global Address Space languages (GAS) support:
  - Global pointers and distributed arrays
  - User controlled layout of data across nodes
  - Communicate using implicit reads & writes of remote memory
- Languages: UPC, Titanium, Co-Array Fortran
  - Productivity benefits of shared-memory programming
  - Competitive performance on distributed memory
- Use Single Program Multiple Data (SPMD) control
  - Fixed number of compute threads
  - Global synchronization, barriers, collectives
- Exploit fast one-sided communication
  - Individual accesses and bulk copies
  - Berkeley implementations use GASNet



- Global Address Space logically partitioned
  - Shared data access from any thread
  - Private data local to each thread
- Pointers are either local or global (possibly remote)



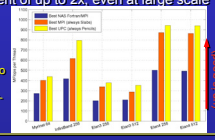
Unified Parallel C

- UPC: a GAS extension to ISO C99
  - shared types & pointers
- Libraries for parallel ops
  - Synchronization
  - Collective data movement
  - File I/O
- Many open source & vendor implementations
  - Berkeley UPC, MuPC
  - HP UPC, Cray UPC
- See UPC Booth #137
- NAS Benchmarks written from scratch
  - FT, CG, MG, LU - all NAS-compliant
  - Use some Berkeley UPC extensions
  - All competitive with best MPI versions, even on cluster networks

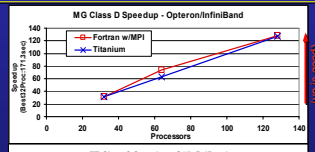
Fourier Transform

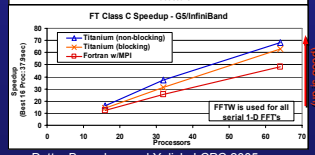
- 3-D FFT with 1-D partitioning across processors
  - Computation is 1-D FFTs using FFTW library
  - Communication is all-to-all transpose
  - Traditionally a bandwidth-limited problem
- Optimization approach in GAS languages
  - Aggressively overlap transpose with 2nd FFT
  - Send more, smaller msgs to maximize overlap
    - Per-target slabs or individual 1-D pencils
  - Use low-overhead one-sided communication
- Consistently outperform MPI-based implementations
  - Improvement of up to 2x, even at large scale

Fine-grained communication overlap less effective in MPI due to increased messaging overheads and higher per-message costs



Performance





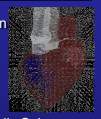
Datta, Bonachea and Yelick, LCPC 2005

AMR Solver	Titanium AMR	Chombo (C++/F/MPI)
Serial (small)	47.6 sec	57.5 sec
Parallel (large)	121.9 sec	113.3 sec

Wen and Collella, IPDPS 2005

Applications

- Immersed Boundary Method Simulation
  - Human Heart
  - Cochlea



- Adaptive Mesh Refinement (AMR)
  - AMR Poisson Multigrid Solvers
  - 3D AMR Gas Dynamics Hyperbolic Solver
  - AMR with Line Relaxation (low aspect ratio)
- Bioinformatics: Microarray oligonucleotide selection
- Finite Element Benchmarks
- Tree-structured n-body kernels
- Dense Linear Algebra: LU, MatMul

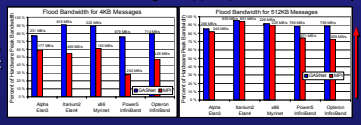
Mach-10 shock hitting solid surface at oblique angle

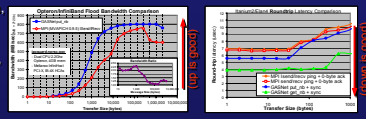
GASNet Portability

- Native network hardware support:
  - Quadrics QsNet I/II (Elan3/Elan4)
  - Cray X1 - Cray shmem
  - SGI Altix - SGI shmem
  - Dolphin - SCI
  - InfiniBand - Mellanox VAPI
  - Myricom Myrinet - GM-1 and GM-2
  - IBM Colony and Federation - LAPI
- Portable network support:
  - Ethernet - UDP: works with any TCP/IP
  - MPI 1: portable impl. for other HPC systems
- Berkeley UPC, Titanium & GASNet highly portable:
  - Runtimes and generated code all ANSI C
    - New platform ports in 2-3 days
    - New network hardware 2-3 weeks
  - CPU's: x86, Itanium, Opteron, Athlon, Alpha, PowerPC, MIPS, PA-RISC, SPARC, T3E, X-1, SX-6, …
  - OS's: Linux, FreeBSD, NetBSD, Tru64, AIX, IRIX, HPUX, Solaris, MS-Windows/Cygwin, Mac OSX, Unicos, SuperUX, Catamount, BlueGene, …

High Performance

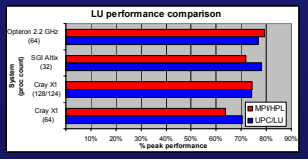
- Communication micro-benchmarks show GASNet consistently matches or outperforms MPI
- One-sided, lightweight communication semantics
  - eliminates matching & rendezvous overheads
  - better match to hardware capabilities
  - leverages widely-available support for remote direct memory access
  - Active Messages support provides extensibility





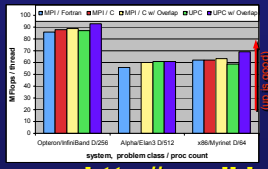
LU Decomposition

- High ratio of computation to communication
  - scales very well to large machines: Top500
- Non-trivial dependence patterns
  - HPL code uses two-sided MPI messaging, and is very difficult to tune
- UPC implementation of LU uses:
  - one-sided communication in GAS model
  - lightweight multithreading atop SPMD
  - memory-constrained lookahead to manage amount of concurrency at each processor
  - highly adaptable to problem/machine sizes
  - natural latency tolerance
- Performance is comparable to HPL / MPI
  - UPC code less than half the size



Conjugate Gradient

- Solves Ax = b for x, where A is sparse 2-D array
- Computation dominated by Sparse Matrix-Vector Multiplications (SPMV)
- Key communication for 2-D (NAS) decomposition:
  - team-sum-reduce-to-all (vector&scalar)
  - point-to-point exchange
- Need explicit synchronization in one-sided model
  - Uses tree-based team reduction library
  - Tunable based on network characteristics
- Overlap the vector reductions on the rows of the matrix with the computation of the SPMV
- Outperform MPI implementation by up to 10%



http://titanium.cs.berkeley.edu

http://gasnet.cs.berkeley.edu

http://upc.lbl.gov