Architecture-aware Themes in FASTMath Algorithms and Software

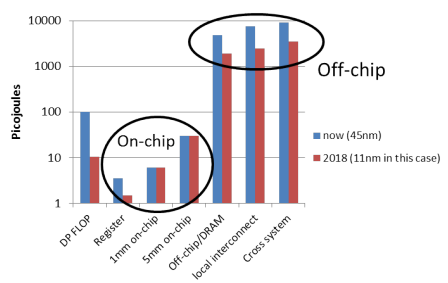
These activities involve all members of FASTMath.

Awareness of architecture features has become increasingly important in the design and implementation of efficient mathematical algorithms for HPC. This awareness guides all the algorithm and software development within FASTMath

Architectural Issues

- Synchronization
  - Must eliminate or limit nonblocking MPI_Allreduce().
- Limited memory/network bandwidth and high latencies
  - Promote data reuse in registers and caches
  - Use cooperative prefetch to prevent saturating the bus
  - Use communication avoiding algorithms
- Data locality
  - Control locations of memory allocations (first touch)
  - Locality versus granularity trade-offs (management of ghost point regions – per core vs per node)
- Load balance (including from machine jitter)
  - Dynamic versus static
  - Interleaved versus locality-enforced memory
- Portable multi-library multicore programming models
  - Library friendly scoping
  - Shared thread pools

Communication Costs Energy and Time

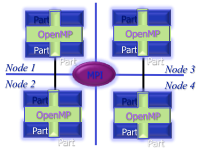


Source: John Shalf, LBL

Architecture Aware Hierarchical Partitioning

Use topology aware MPI and knowledge of node level layout of cores and memory sockets (via hwloc) to simultaneously reduce inter-node and intra-node dependencies and communication volume.

Strategy demonstrated in Zoltan for both graph-based and geometric partitions.



See the partitioning poster for more details.

Lead: Karen Devine http://www.cs.sandia.gov/Zoltan/

Unstructured Adaptive Meshes

Unstructured mesh adaptation characteristics

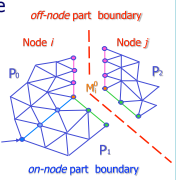
- Flexible data structure required for mesh modifications
- Parallelization of complex mesh modification logic has been done at a partition level
- Continuing parallel control at a partition level, but with on-node threading, appears to be desirable

Approach – Two level partition model

- Architecture-aware mesh partitioning
  - first-level to node level - MPI process
  - second-level partitioning on thread level: cores on a node, each part is mapped to a thread
- Only high-level mesh modification APIs are exposed to the application

Status of Development in FMDB

- Portable Hardware Locality (hwloc) library used to detect needed architecture details
- Two-level mesh partitioning – on-going
- Part handle to thread in shared memory – on-going



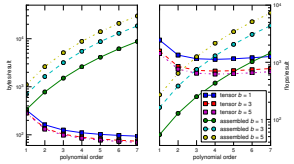
Lead: Seegyoung Seol http://www.scorec.rpi.edu/FMDB

Communication Avoiding Algorithms

- Krylov Subspace Methods (KSMs)
  - Bottleneck: Sparse-matrix-vector multiplication (SpMV)
    - Need to reorganize KSM to avoid communication
  - Have reorganized many KSMs to attain lower bound
    - GMRES
    - Conjugate Gradients (CG), Biconjugate Gradients (BiCG), CG Squared (CGS), BiCG Stabilized (BiCGStab), Arnoldi, Lanczos
    - Needs CA "matrix powers kernel" $[Ax, A^2x, \ldots, A^kx]$, other dense CA kernels
  - Obstacle – choosing numerically stable reorganization
    - May choose more stable polynomial basis $[p_1(A)x, p_2(A)x, \ldots, p_k(A)x]$ where $p_k(A)$ is a degree-k polynomial in A
    - New Result – extension of "residual replacement strategy" to CA-KSMs
      - Monitors divergence of true and implicit residual, restarts when needed (rarely)

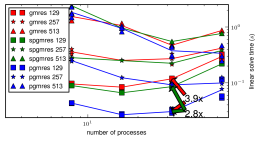
Lead: Jim Demmel http://bebop.cs.berkeley.edu

Algorithmic Adaptation



- Matrix-free methods relieve memory bandwidth bottleneck
- Unassembled Jacobian stored as coefficients at quadrature points
- Polynomial/multistage smoothers
- Dual-order defect correction for h-ellipticity
- 70% of peak FPU for third order tensor product elements

- High order and multiscale discretization: more flops and increased arithmetic intensity in exchange for better accuracy
- Nonlinear solvers avoiding linearization can reduce number of communicating iterations
- Pipelined Krylov methods to hide latency of reductions, see linear algebra poster



Future Plans

- Thread-level nonblocking communication with few or no expensive atomic instructions
- Vectorization and cooperative threading of library kernels
- Algorithmic advances in robustness of matrix-free methods
- More complete and transparent use of accelerators (GPUs and MIC)
- Topology-aware partitioning and load balancing that respects memory locality

More Information: http://www.fastmath-scidac.org or contact Lori Diachin, LLNL, diachin2@llnl.gov, 925-422-7130

SciDAC Scientific Discovery through Advanced Computing