Parallel 3D Adaptive Mesh Refinement in Titanium

Geoff Pike
Luigi Semenzato
Phillip Colella
Paul N. Hilfinger

UC Berkeley EECS
VMware, Inc.
NERSC, LBNL
UC Berkeley EECS
AMR Poisson

- Goal is to solve Poisson’s Equation
- Similar to Multigrid, more efficient
- Patch-based algorithm
Titanium

- based on Java
- extensions for scientific computing
- SPMD parallelism
- shared address space
- efficient
- portable
foreach (p in rhs.domain() / [2, 2, 2]) {
    field[p * 2] += factor[p * 2] * (field[p * 2 + [0, 0, 1]] + field[p * 2 + [0, 0, -1]] + field[p * 2 + [0, 1, 0]] + field[p * 2 + [0, -1, 0]] + field[p * 2 + [1, 0, 0]] + field[p * 2 + [-1, 0, 0]] - 6 * field[p * 2] - h * h * rhs[p * 2]);
}
Preliminary Results

sequential 3D multigrid:
(64x64x64 problem size)
within about 20% of FORTRAN

parallel 3D AMR:
(on a fixed well-balanced problem
8 patches of size 72x72x72 at finest level)
5x to 5.5x speedup on 8 procs
Timeline with Barriers

proc 0  proc 1  proc 2  proc 3

10.0%  3.6%  3.3%  3.9%
Conclusions

What makes it work:

- Titanium Programming Model
- amrvis (LBNL/CCSE) for visualization
- Profiling tools beyond gprof

Future Work:

- Adaptivity
- Combine with larger application (e.g., self-gravitating compressible flow)