Adaptive Mesh Refinement in Titanium

Tong Wen & Phillip Colella

http://seesar.lbl.gov/anag

Lawrence Berkeley National Laboratory

April 7, 2005

Overview

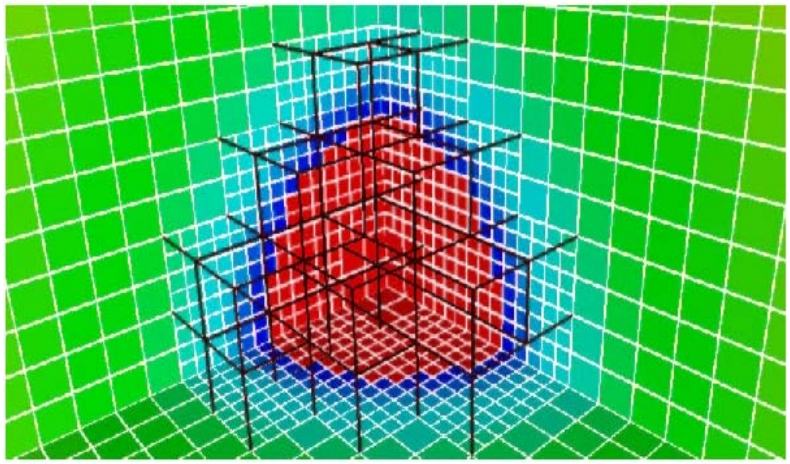
- **Motivations:**
  - Build the infrastructure in Titanium for applications of adaptive mesh refinement (AMR)
  - Provide a nontrivial case study of Titanium's usability (development and execution productivity)

The Titanium Language: a Java Dialect for Scientific computing

- **A high-level language designed to simplify parallel programming, meanwhile to provide high performance**
- **Features:**
  - Global address space and explicit SPMD execution model
  - Multidimensional rectangular arrays
  - One-sided communication
  - Templates
  - immutable (value) classes
  - Zone-based memory management
- **Titanium programs run on both shared-memory and distributed-memory architectures**

AMR for Partial Differential Equations (PDEs)

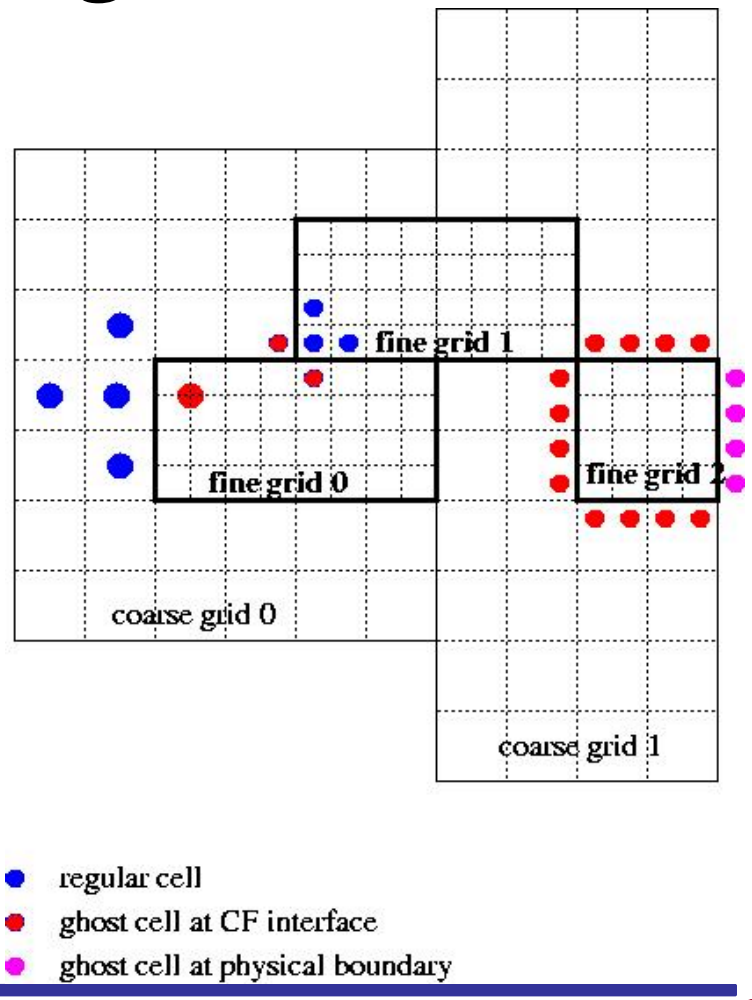
- **A variety of physical problems exhibit multiscale behavior, in the form of localized large gradients separated by large regions where the solution is relatively smoother**



- **In adaptive methods, one adjusts the computational effort locally to maintain a uniform level of accuracy throughout the problem domain**
- **The goal of local refinement is to save computational resources**

Implementing Block-Structured AMR Algorithms is Challenging

- **Simplicity is traded for computational resources in AMR**
  - Mixture of regular and irregular data access and computation
  - Dealing with all kinds of boundaries is the source of irregular operations
  - Once the ghost values are determined, evaluating any finite difference scheme on each grid is a local operation



Implementing Block-Structured AMR Algorithms is Challenging

- **Also complicated are the control structures and interactions between levels of refinement**

- **In real applications, grid configuration is not known until run time, and it may change from time to time**

A Prototype of AMR in Titanium

Chombo

- **Chombo is a widely used AMR package written in C++/Fortran with MPI:**
  - C++: complicated data structures and irregular computations
  - Fortran: evaluation of operations on rectangular arrays
- **Bulk-synchronous communication:**
  - Communicate boundary data for all grids at a level
  - Perform local calculation on each grid in parallel

Titanium AMR

- **Follow the design of Chombo with modifications to suit Titanium**
  - Basic AMR data structures and operations
  - A solver for elliptic PDEs
- **Fully written in Titanium (no Fortran/C, no MPI)**
- **Our implementation has covered almost all Titanium's features**

A Prototype of AMR in Titanium

- **Basic data structures for AMR applications:**
  - The metadata class and the data class
- **Basic AMR operations implemented as methods of classes:**
  - Exchange values along the grid boundaries at the same refinement level (Exchange)
  - Quadratic interpolation of the boundary values at the coarse-fine interface (CFInterp1&2)
- **A solver for elliptic PDEs is built on the above infrastructure**
  - Point relaxation scheme (GSRB)

Titanium VS. C++/Fortran/MPI: Lines of Code

- **Numbers of lines of code:**

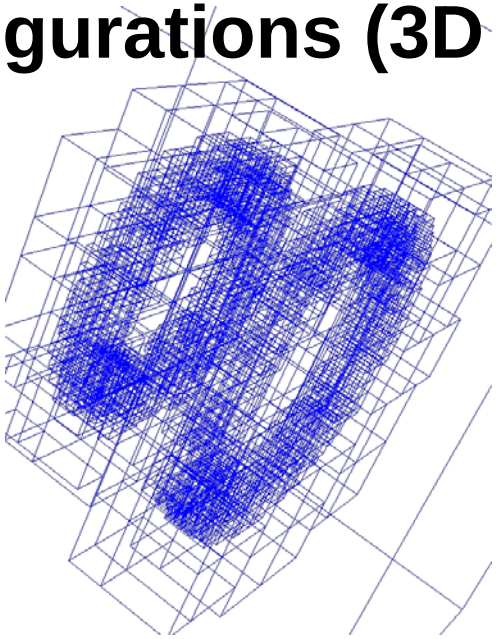
	Titanium	C++/Fortran/MPI
AMR data structures	2000	35000
AMR operations	1200	6500
elliptic PDE solver	1500	4200*

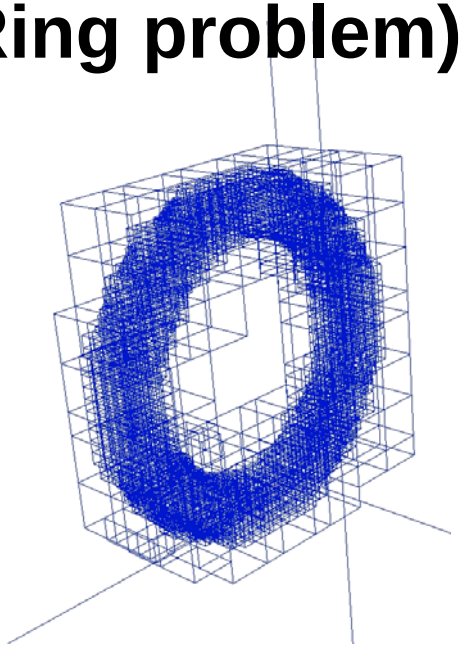
*more functionalities are implemented in Chombo

- **Why are numbers of lines smaller for Titanium?**
  - Functionality that has to be implemented as libraries in Chombo is supported at language level in Titanium

Two Test Problems

- **Solving Poisson's equation with two grid configurations (3D Vortex Ring problem):**





the small configuration			the large configuration		
level	# of grids	# of cells	level	# of grids	# of cells
0	1	33K	0	64	2M
1	106	280K	1	129	3M
2	1449	3M	2	3159	62M

Serial Performance

- **The same version of code is run on two platforms:**
  1. An Intel Pentium 4 workstation
  2. Seaborg: the 21st in Top500 list
- **The Titanium compiler we used is version 2.573**

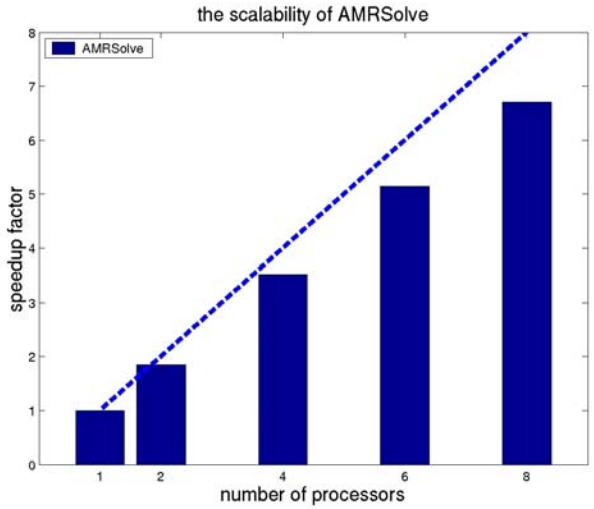
- **On the Intel Pentium 4 workstation, the small test problem:**

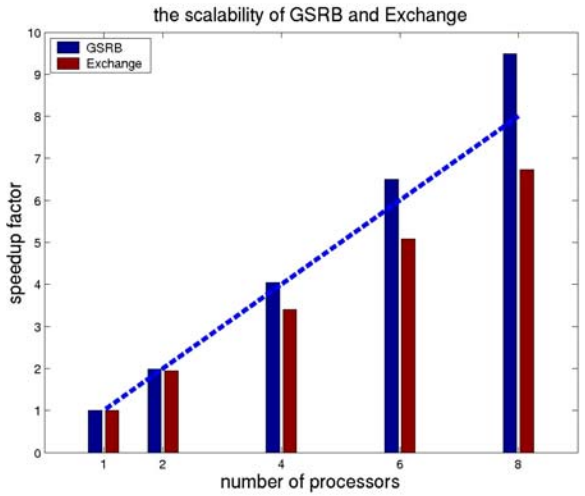
(the timing results are in seconds)

	Titanium AMR	Chombo
AMRSolve	**52.15**	**57.47**
GSRB	12.98	11.64
Exchange	11.25	17.31
CFInterp1	5.91	4.19
CFInterp2	4.97	4.31
other	17.04	20.02

Parallel Performance

The scalability of the small test problem on Seaborg (32 bit):





(the timing results are in seconds)

# of processors	1	2	4	6	8
AMRSolve	**138.4**	**74.63**	**39.31**	**26.90**	**20.62**
GSRB	33.85	17.11	8.38	5.21	3.57
Exchange	27.66	14.26	8.12	5.45	4.11
CFInterp1	14.33	7.13	2.88	1.51	1.00
CFInterp2	15.27	9.33	4.90	3.41	2.89
Other	47.29	26.80	15.03	11.32	9.05

Parallel Performance

- **Scalability of the large test problem on Seaborg (64-bit):**

(the timing results are in seconds)

# of processors	14	28	42
AMRSolve	**204.7**	**134.6**	**116.7**
GSRB	58.39	29.46	19.18
Exchange	42.60	41.32	46.23
CFInterp1	10.05	5.20	3.78
CFInterp2	12.53	10.03	9.87
other	81.13	48.59	37.64

- **Note that the source and destination regions of Exchange operation are non-contiguous in linear storage**
- **Possible improvements to reduce the communication cost:**
  - Packing at application level
  - More efficient packing in the GASNet communication system

Parallel Performance

- **Titanium vs. C++/Fortran/MPI on the large test problem on Seaborg, where two nodes (28 processors) are used.**

(the timing results are in seconds)

	Titanium AMR	Chombo
AMRSolve	**130.0**	**113.3**
GSRB	25.34	22.71
Exchange	40.56	37.12
CFInterp1	5.15	6.17
CFInterp2	10.10	7.97
other	48.85	40.33

Conclusion and Future Work

- **Titanium's strength:**
  - A high-level language that is easy to learn and easy to use
  - Writing AMR applications in Titanium requires much less programming effort
  - Potential to provide high performance
- **Continuing improvements to Titanium are motivated by this project:**
  - A recent change in Titanium compiler has provided an average of 10% speedup of our test code
- **Future work:**
  - Improve the performance of AMR Exchange
  - A performance model of Titanium AMR would be interesting
  - New AMR development: ocean modeling (solvers for large aspect ratio grids)

Acknowledgements

- **This project is supported by Lawrence Berkeley National Laboratory**
- **Our thanks go to**
  - Titanium group at University of California, Berkeley
    Dan Bonachea, Jimmy Su, and Amir Kamil
  - ANAG group at Lawrence Berkeley National Laboratory
    Dan Martin and Noel Keen

- **Our emails:**

  twen@lbl.gov and pcolella@lbl.gov
- **URLs:**
  1. http://seesar.lbl.gov/anag/staff/wen/download.html
  2. http://seesar.lbl.gov/ANAG/software.html

Appendix

- **The infinity norms of the residuals from the two test problems:**

the small test problem		
iteration	Titanium AMR	Chombo
initial	6144.0	6144.0
1	0.2727	0.2728
2	0.2538	0.2538
3	0.0091	0.0092
3	3.706E-04	3.580E-04
5	5.093E-06	4.748E-06
6	2.090E-07	1.570E-07

the large test problem		
Iteration	Titanium AMR	Chombo
Initial	9.830E04	9.830E04
1	4.169	4.169
2	1.290	1.277
3	0.0222	0.0219
4	1.046E-03	1.039E-03
5	1.761E-05	2.218E-05
6	7.839E-07	7.367E-07