SAND2006-6921C

Presentation at INFORMS Fall 2006, Pittsburgh

# Opportunities for Parallelism in Optimization Algorithms

**David M. Gay**

Optimization and Uncertainty Estimation

*dmgay@sandia.gov*

+1-505-284-1456

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.
Released as SAND2006-4717C.

Sandia National Laboratories

# Outline

- Motivation: do hard things faster
- Kinds of parallelism
- Some forthcoming(?) systems
- Programming models
- Algorithmic opportunities
- Modeling-language opportunities

Sandia National Laboratories

# Need for Speed

- Better models (more physics)
- Hard problems (global opt.)
- Dealing with uncertainty
  - UQ — quantifying affects
  - decisions/control — stochastic programming

# Single-Processor Parallelism

- Vector
- Instruction-level
- Speculation
- Threading

Sandia National Laboratories

# Multiprocessor Parallelism

- SIMD
  - Vector (Cray; attached proc.)
  - Connection Machine; IBM Cell
- MIMD
  - Shared memory
  - Distributed memory

## Some Forthcoming? Systems — Intel

Intel (& AMD...) multicore:

- Today dual core
- Soon quad core
- Within years, 10s or 100s of cores
  - perhaps simpler than x86
  - lower power consumption

## Some Forthcoming? Systems — IBM Cell

IBM Cell:

- PPU + 8 (later more) SPUs
- Soon in Playsta. 3, X Box, ...
- Balanced, heterogeneous arch.:
  SPUs with 3-level memory
  - register files; local & main store
  - DMA local $\Longleftrightarrow$ main store

# Some Forthcoming? Systems — Eldorado

## Cray (nee Tera) Eldorado

- MTA $\Longrightarrow$ MTA-2 $\Longrightarrow$ Eldorado
- Many threads (128/processor)
- Custom CPU, "Commodity" boards (Red Storm, Cray XT3)
- 4 extra tag bits / 64-bit word

Sandia National Laboratories

# Programming Models

- Automatic
  - CPU
  - Compiler
- Added directives
  - Special comments; #`pragmas`
- Intrusive changes

# How Programming Models Change

Expectation for Cell seems general:

- Initially do custom programming
- Then call library routines
  - e.g., BLAS, LINPACK, LAPACK
- Tools (compiler, linker, ...) automate some things

Sandia National Laboratories

# Opportunities for (Opt.) Algorithms

- Parallelize existing algorithms
  - lib. lin. algebra may win big
- Alt. algorithms may be attractive
  - decomposition
  - speculation
  - polyalgs. (hetereogeneous search)

Sandia National Laboratories

# More Opportunity for (Opt.) Algorithms

More computing resources sometimes make new algorithms feasible...

- Direct rather than iterative methods
- All-at-once (SAND) rather than block-by-block
- E.g., replace EM alg. by estimating missing data & model pars together

Sandia National Laboratories

# When Performance Matters

Only resort to parallelism when serial is too slow, then follow standard advice...

- First get it right (& simple) in serial
- Gradually adapt
- Profile to find hot spots
- Many kinds of tools can help.

Sandia National Laboratories

# Modeling-Language Opportunities

Specifically, AMPL...

- Could instantiate in parallel.
- Higher priority = 2-way talk with solvers, for fast problem updates.
- For decomposition algs,

```
parfor{i in S} solve Prob[i];
```

Sandia National Laboratories

# Closing Remarks

- These are "interesting" times.
- Speedups can change how we work.
- Many ways to exploit parallelism.
- Reversible computing promises more speed, less power — more new computing.

Sandia National Laboratories