Introductory Tutorial in Parallel Programming

Marcel Meyer, Markus Gerstel, Stefan Wallentowitz

Seminar 20 – Advanced Computing
Technische Universität München

September 25, 2009

Agenda

1. Parallel Programming (Markus)
2. Parallel Architectures (Stefan)
3. OpenMP (Markus)
4. MPI (Marcel)
5. Hardware Support for Parallel Programming (Stefan)

Why Parallel Programming - Small Scale View

- Single core performance plateauing
- Further increase possible but increasingly difficult
- Especially power limitations
- $\Rightarrow$ **Cost efficiency**

Why Parallel Programming - Small Scale View

- Single core performance plateauing
- Further increase possible but increasingly difficult
- Especially power limitations
- $\Rightarrow$ **Cost efficiency**

Why Parallel Programming - Large Scale View

- Intel Core i7, 3.2GHz: $\approx$ 0.0000512 petaflops
- Earth Simulator: $\approx$ 0.03586 petaflops
- Altix/HLRBII: $\approx$ 0.0623 petaflops
- Bluegene/L: $\approx$ 0.280 petaflops
- Roadrunner: $\approx$ 1.7 petaflops
- BOINC: currently $\approx$ 1.9 petaflops

- $\Rightarrow$ Allows large problemsets

Why Parallel Programming - Large Scale View

- Intel Core i7, 3.2GHz: $\approx$ 0.0000512 petaflops
- Earth Simulator: $\approx$ 0.03586 petaflops
- Altix/HLRBII: $\approx$ 0.0623 petaflops
- Bluegene/L: $\approx$ 0.280 petaflops
- Roadrunner: $\approx$ 1.7 petaflops
- BOINC: currently $\approx$ 1.9 petaflops

- $\Rightarrow$ Allows large problemsets

Why Parallel Programming - Large Scale View

- Intel Core i7, 3.2GHz: $\approx$ 0.0000512 petaflops
- Earth Simulator: $\approx$ 0.03586 petaflops
- Altix/HLRBII: $\approx$ 0.0623 petaflops
- Bluegene/L: $\approx$ 0.280 petaflops
- Roadrunner: $\approx$ 1.7 petaflops
- BOINC: currently $\approx$ 1.9 petaflops

- $\Rightarrow$ Allows large problemsets

Why Parallel Programming - Large Scale View

- Intel Core i7, 3.2GHz: $\approx$ 0.0000512 petaflops
- Earth Simulator: $\approx$ 0.03586 petaflops
- Altix/HLRBII: $\approx$ 0.0623 petaflops
- Bluegene/L: $\approx$ 0.280 petaflops
- Roadrunner: $\approx$ 1.7 petaflops
- BOINC: currently $\approx$ 1.9 petaflops

- $\Rightarrow$ Allows large problemsets

Why Parallel Programming - Large Scale View

- Intel Core i7, 3.2GHz: $\approx$ 0.0000512 petaflops
- Earth Simulator: $\approx$ 0.03586 petaflops
- Altix/HLRBII: $\approx$ 0.0623 petaflops
- Bluegene/L: $\approx$ 0.280 petaflops
- Roadrunner: $\approx$ 1.7 petaflops
- BOINC: currently $\approx$ 1.9 petaflops

- $\Rightarrow$ Allows large problemsets

Why Parallel Programming - Large Scale View

- Intel Core i7, 3.2GHz: $\approx$ 0.0000512 petaflops
- Earth Simulator: $\approx$ 0.03586 petaflops
- Altix/HLRBII: $\approx$ 0.0623 petaflops
- Bluegene/L: $\approx$ 0.280 petaflops
- Roadrunner: $\approx$ 1.7 petaflops
- BOINC: currently $\approx$ 1.9 petaflops

- $\Rightarrow$ Allows large problemsets

Parallel Programming - the Cure for Cancer?

So parallel programming is perfect:
Everything gets faster, better, tastier, brighter!

Except that it's not. Drawbacks?

race conditions, synchronisation,

Parallel Programming - the Cure for Cancer?

So parallel programming is perfect:
Everything gets faster, better, tastier, brighter!

Except that it's not. Drawbacks?

race conditions, synchronisation, parallel slowdown,

Parallel Programming - the Cure for Cancer?

So parallel programming is perfect:
Everything gets faster, better, tastier, brighter!

Except that it's not. Drawbacks?

race conditions, synchronisation, parallel slowdown, deadlocks, ...

Parallel Programming - the Cure for Cancer?

So parallel programming is perfect:
Everything gets faster, better, tastier, brighter!

Except that it's not. Drawbacks?

race conditions, synchronisation, parallel slowdown, deadlocks, debugging

Parallel Programming - the Cure for Cancer?

So parallel programming is perfect:
Everything gets faster, better, tastier, brighter!

Except that it's not. Drawbacks?

race conditions, synchronisation, parallel slowdown, deadlocks, debugging

Parallel Programming - the Cure for Cancer?

So parallel programming is perfect:
Everything gets faster, better, tastier, brighter!

Except that it's not. Drawbacks?

race conditions, synchronisation, parallel slowdown, deadlocks, debugging

Parallel Programming - the Cure for Cancer?

So parallel programming is perfect:
Everything gets faster, better, tastier, brighter!

Except that it's not. Drawbacks?

race conditions, synchronisation, parallel slowdown, deadlocks, debugging

Parallel Programming - the Cure for Cancer?

So parallel programming is perfect:
Everything gets faster, better, tastier, brighter!

Except that it's not. Drawbacks?

race conditions, synchronisation, parallel slowdown,
deadlocks, debugging

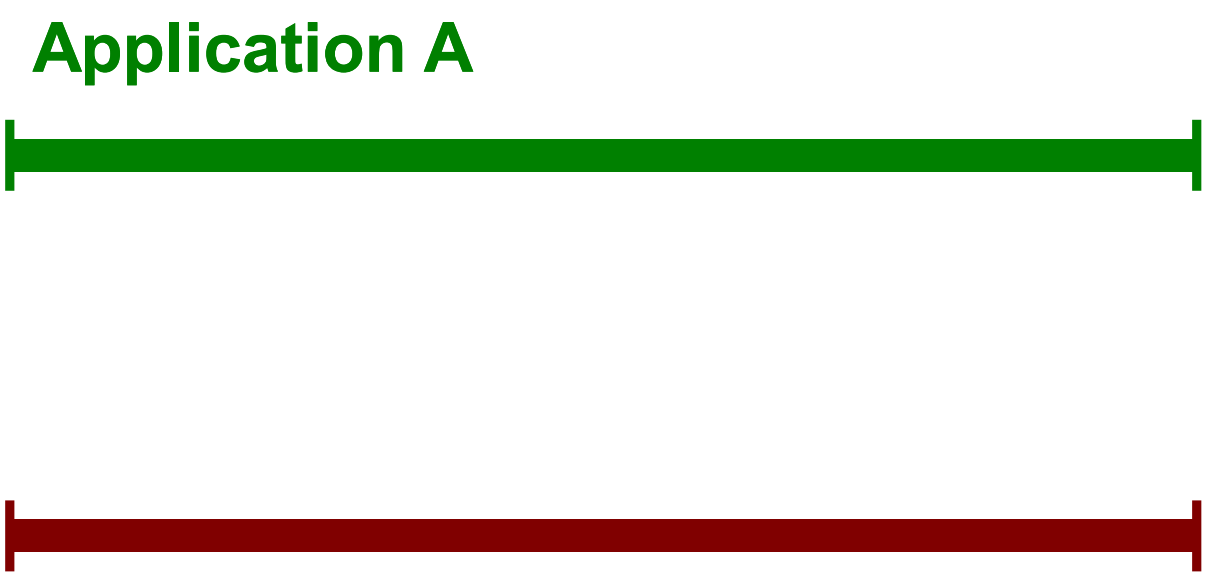
Parallel Programming - the Cure for Cancer?

So parallel programming is perfect:
Everything gets faster, better, tastier, brighter!

Except that it's not. Drawbacks?

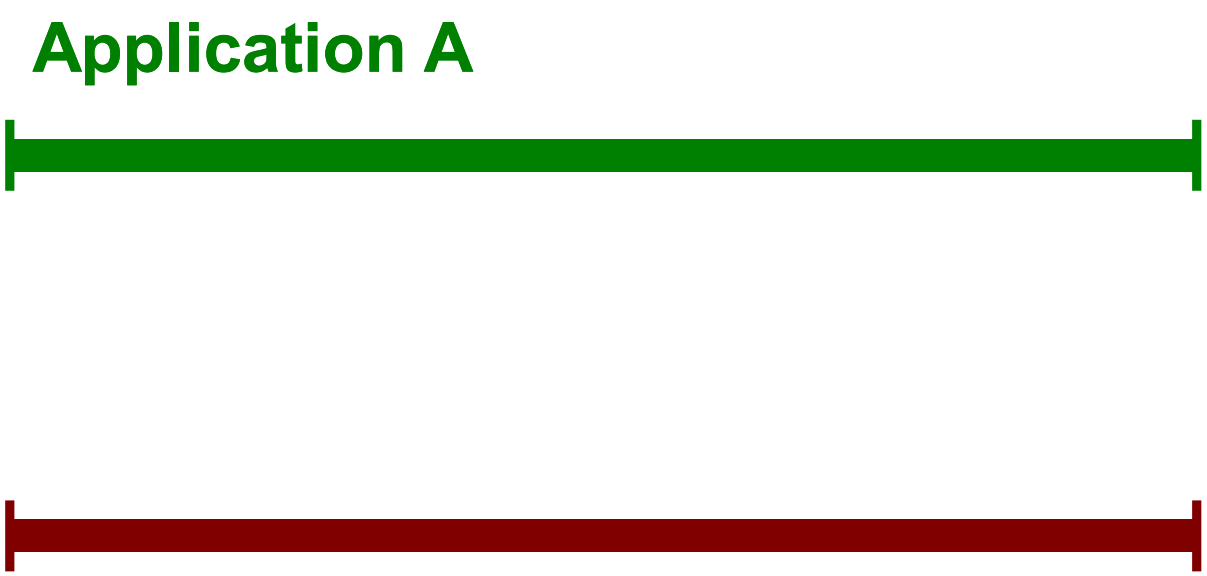
race conditions, synchronisation, parallel slowdown,
deadlocks, debugging

Data Dependency I



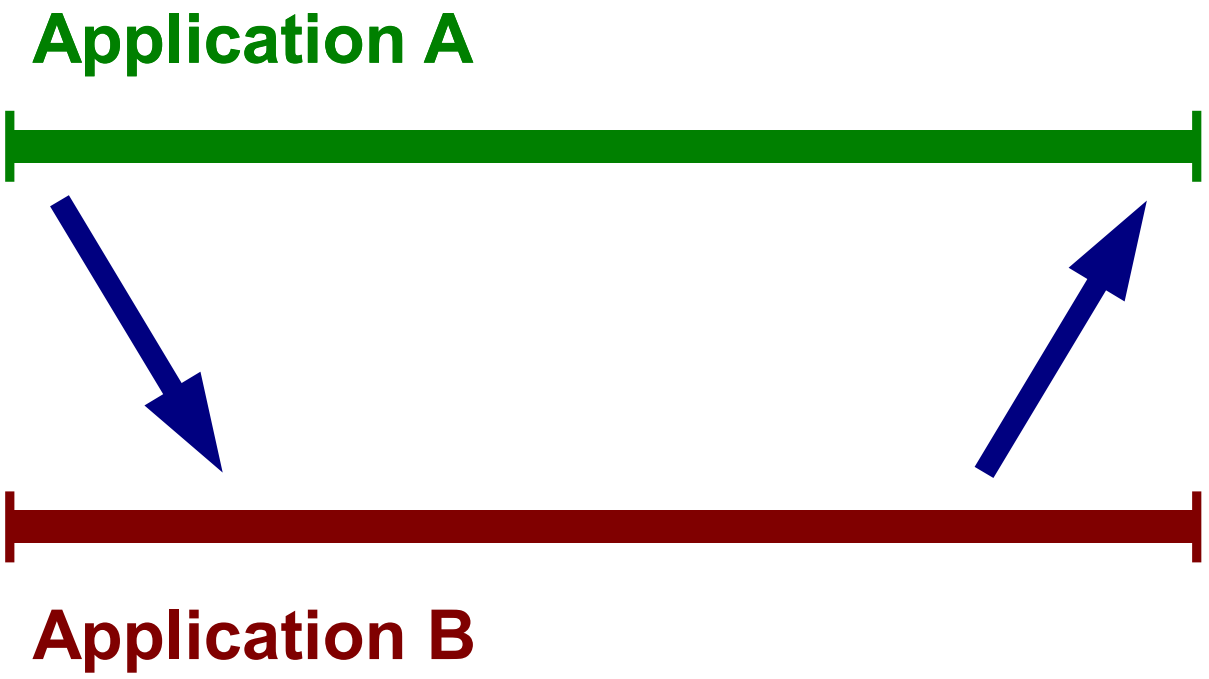
Virtually no data dependency, embarrassingly parallel
Think of: Independent jobs, Distributed networking,...

Data Dependency I



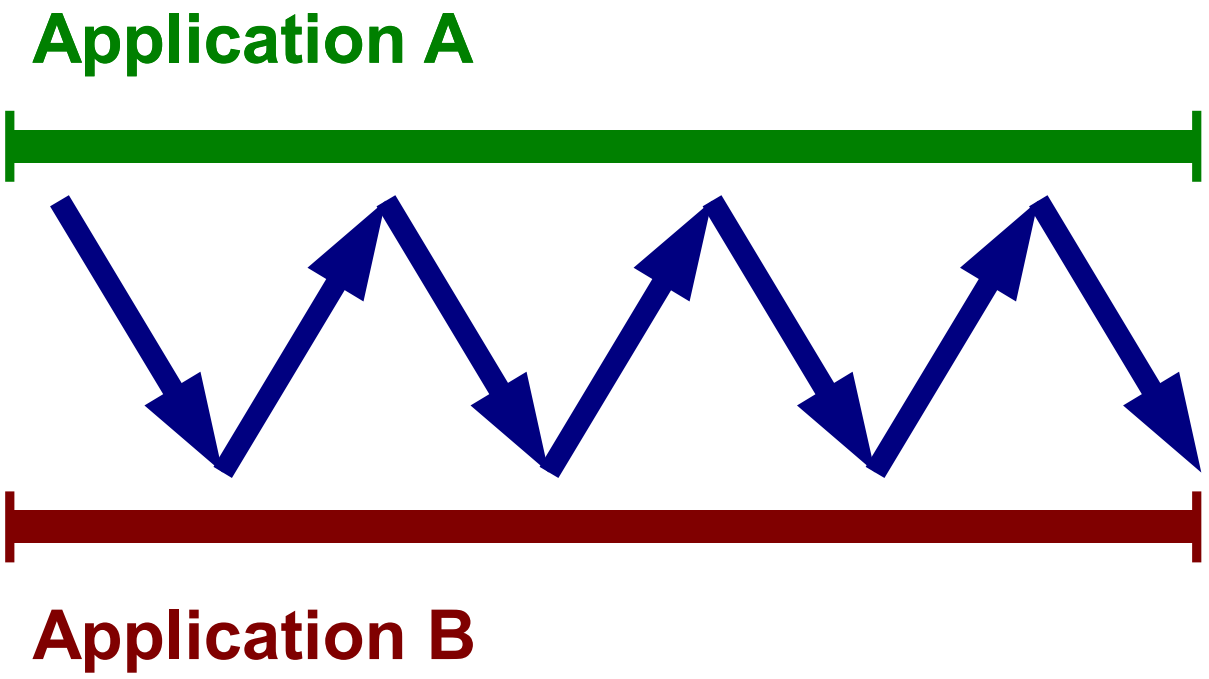
Virtually no data dependency, embarrassingly parallel
Think of: ..., SETI@Home, Brute force algorithms

Data Dependency II



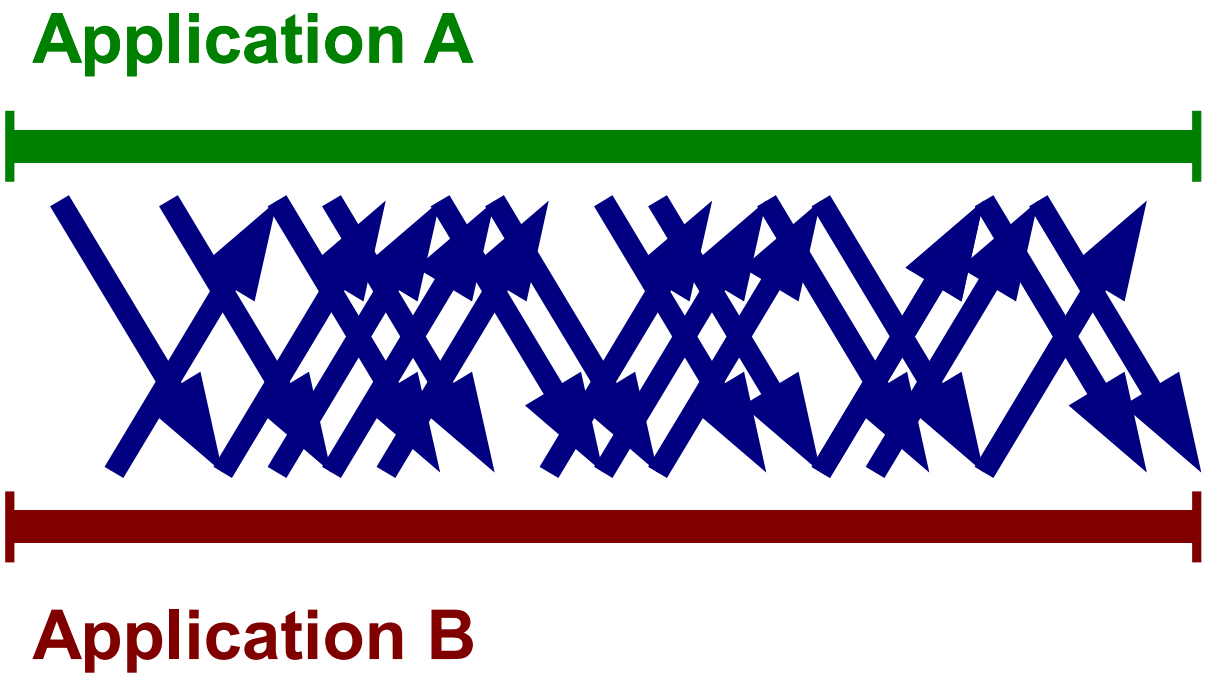
Low data dependency, coarse-grained parallelism
Think of: Rendering Jobs

Data Dependency III



Low data dependency, coarse-grained parallelism
Think of: Relaxation on 2d-grids with small stencil

Data Dependency IIII



High data dependency, fine-grained parallelism
Think of: Relaxation with large stencils