
Fast… Faster… FASTER!
Ade Miller, Programmer

Caveat Emptor

This isn’t Microsoft

This is me

I’m not (much of) an expert

These are my crazy ideas

Your mileage may vary… a lot

“Some guy from Microsoft said…”

So What's this All About?

Real Problems Like This…

$N > 10^6$

First Some Physics…

For each particle calculate force and acceleration:

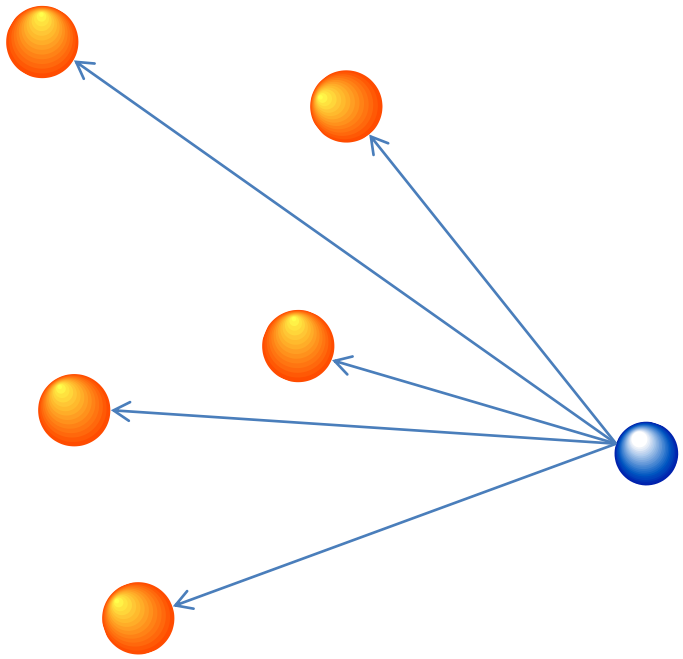
$$\vec{f}_{ij} = \frac{G m_i m_j}{r_{ij}^2} \hat{r}$$

$$a = f / m$$

At each time step update position and velocity:

$$r_{n+1} = r_n + v_n \times dt$$

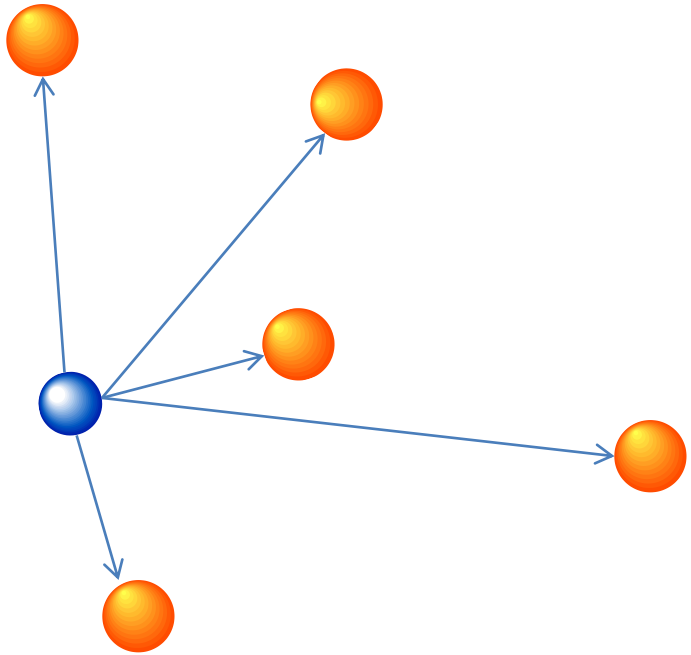
$$v_{n+1} = v_n + a_n \times dt$$



First Some Physics…

Do this for every particle:

Scales as: N^2



For real problems N is very large

So… N = 10,000 is **100 times slower** than 1,000

Also Applies To This…

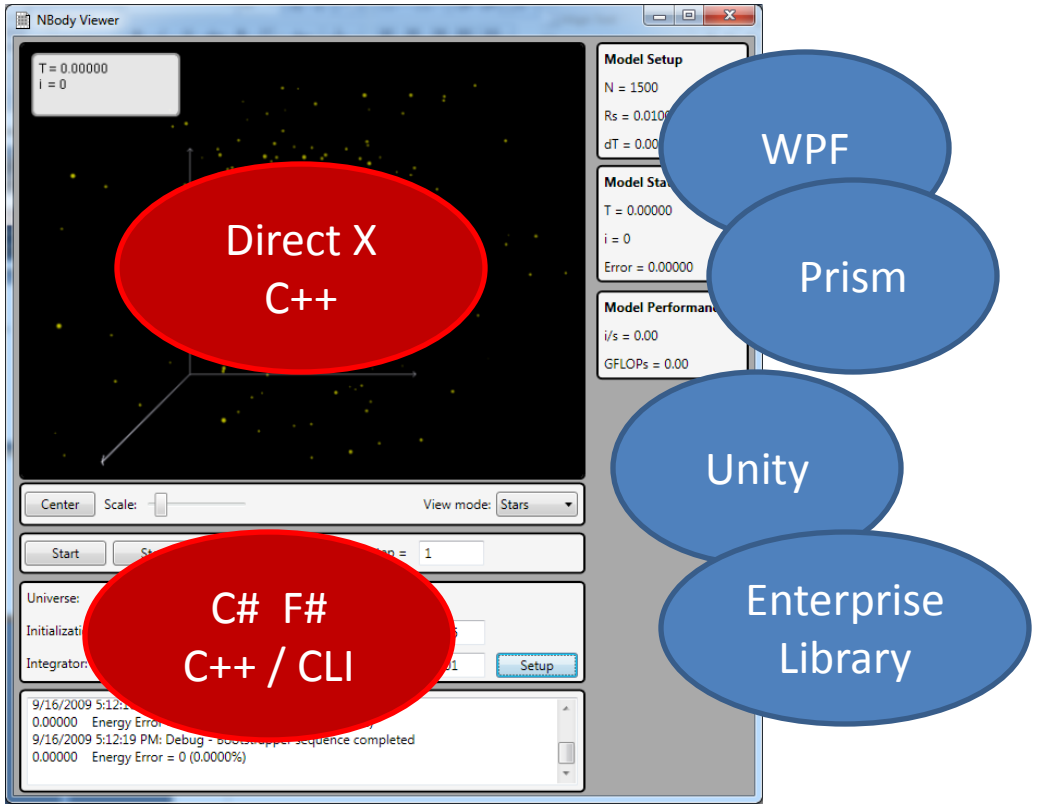
Familiar Requirements

- A rich UI with significant rendering requirements
  - Mixed WPF and DirectX UI
- Requires significant processor resources
  - Native code
  - Parallel (multi-core and GPGPU) code
- A diverse problem domain
  - Both OO and Functional implementations

The Problem

- I want the productivity and richness of .NET
  - Memory management
  - Rich tooling and frameworks
- But native code gives me:
  - Fine grained control
  - Performance
  - Access to specific hardware like SSE and GPUs

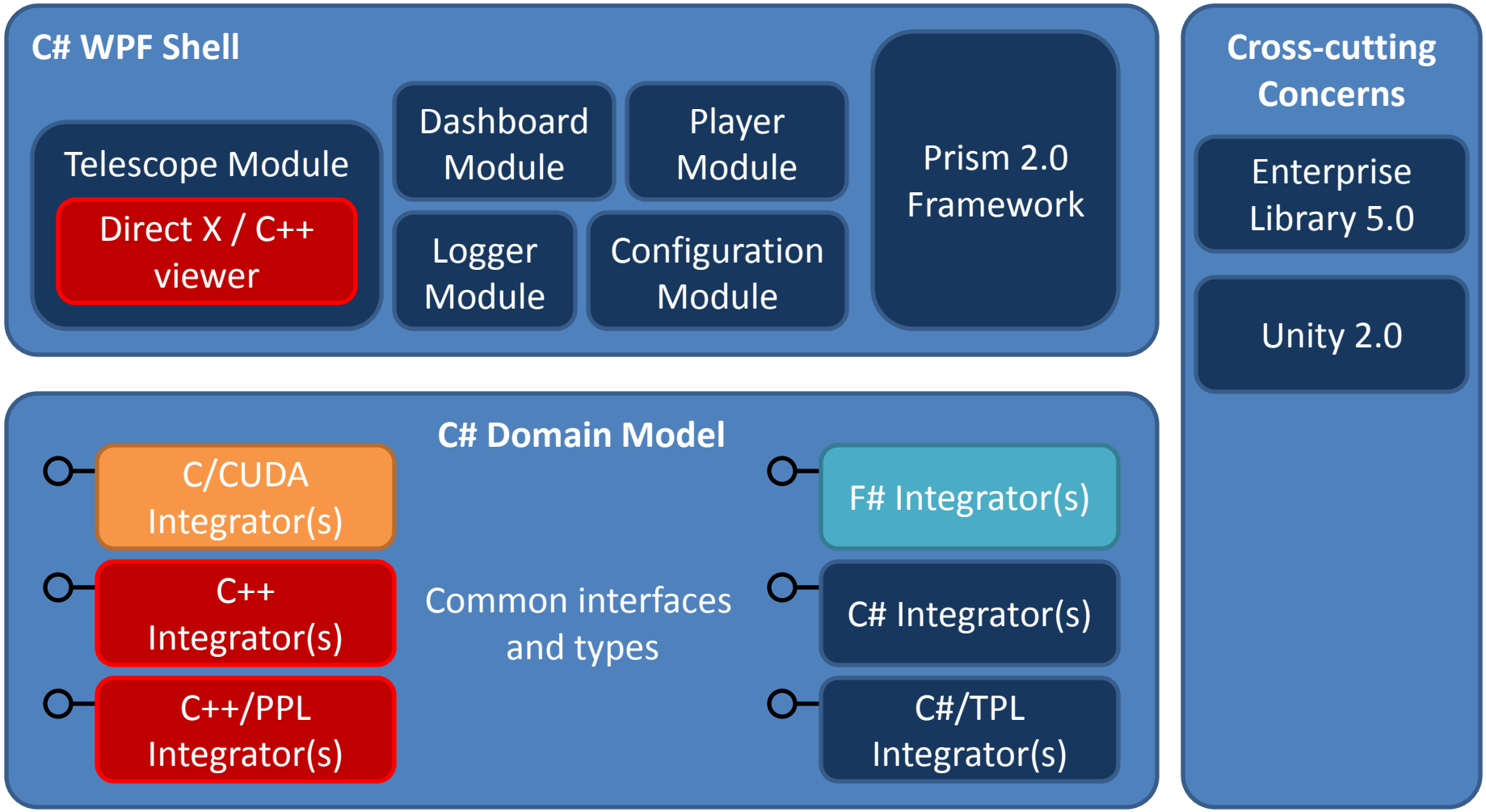
I'd like to have my cake and eat it



Demo…

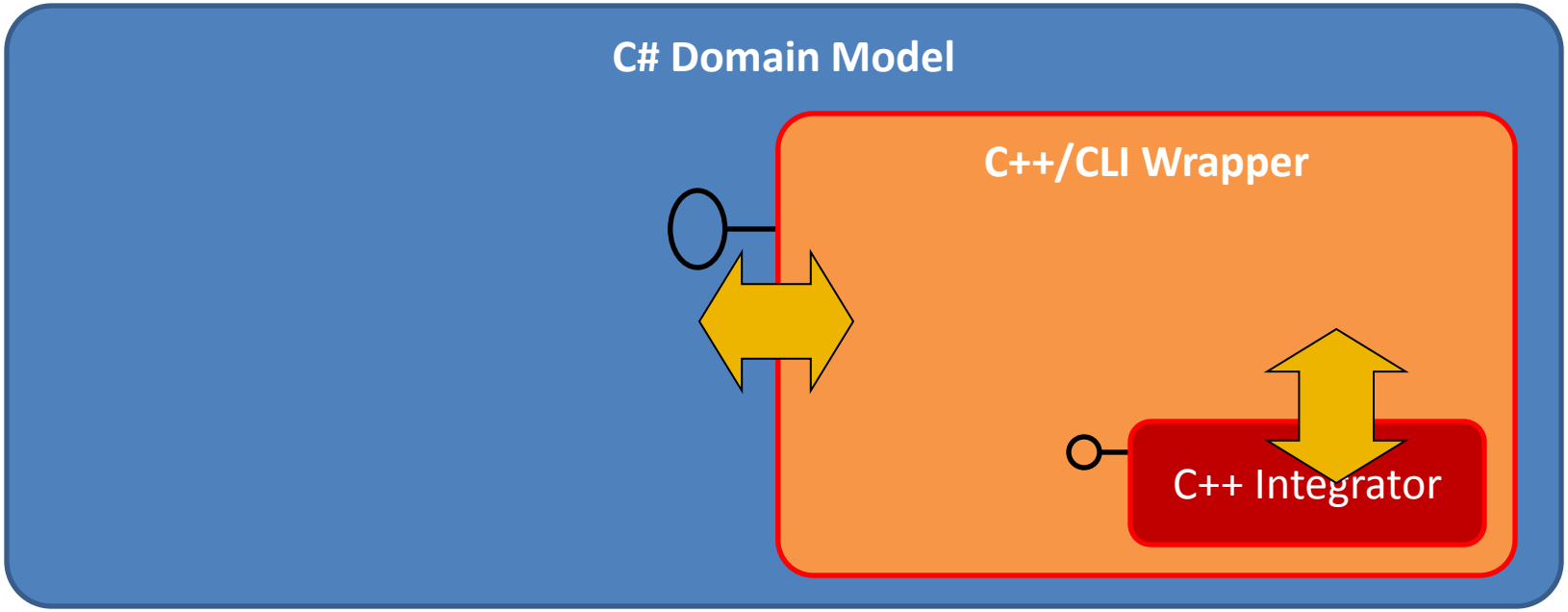
A QUICK APPLICATION TOUR

A Mixed Language Architecture



Deep Dive…

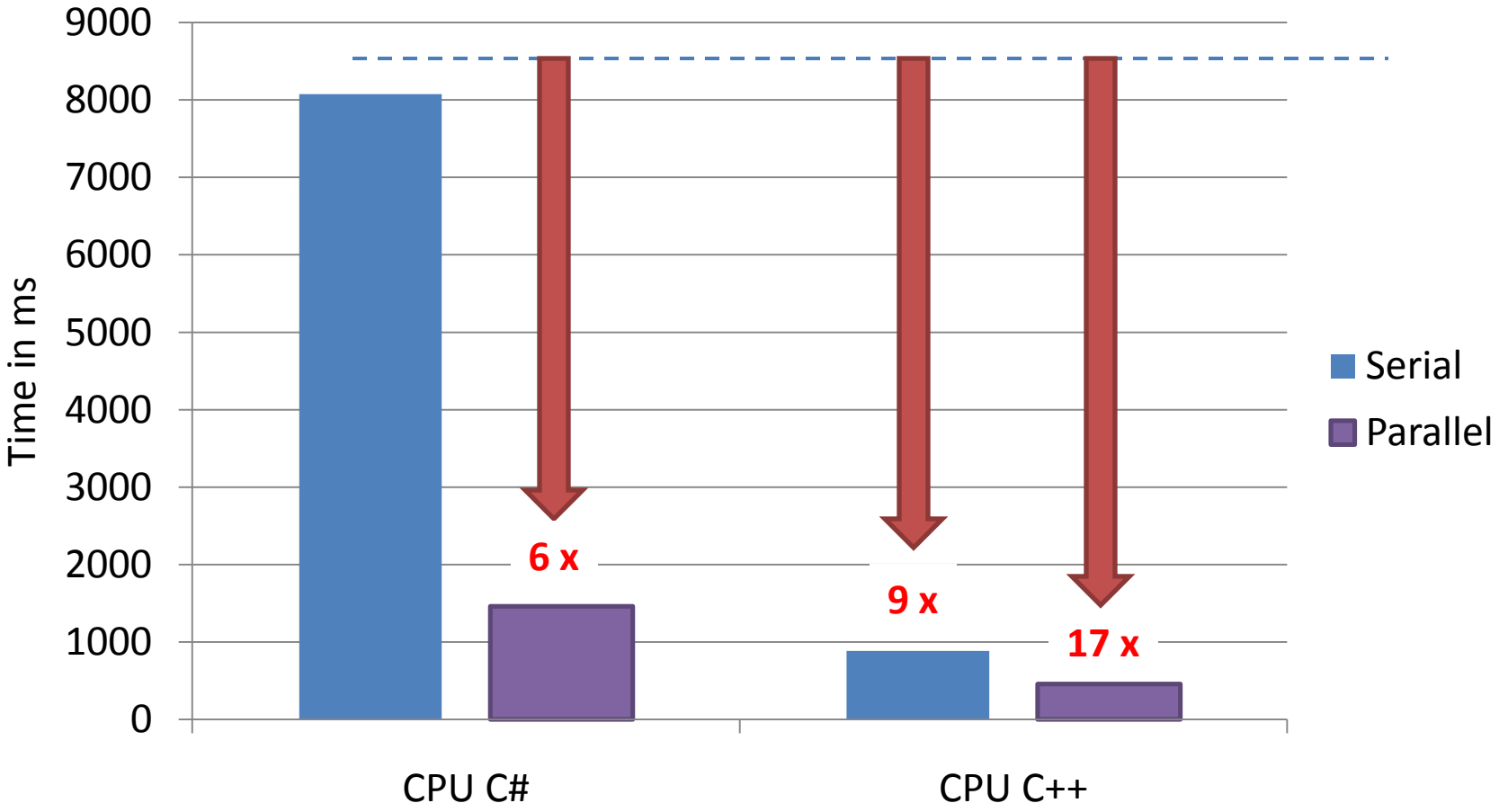
THE BASICS… PARALLEL C#



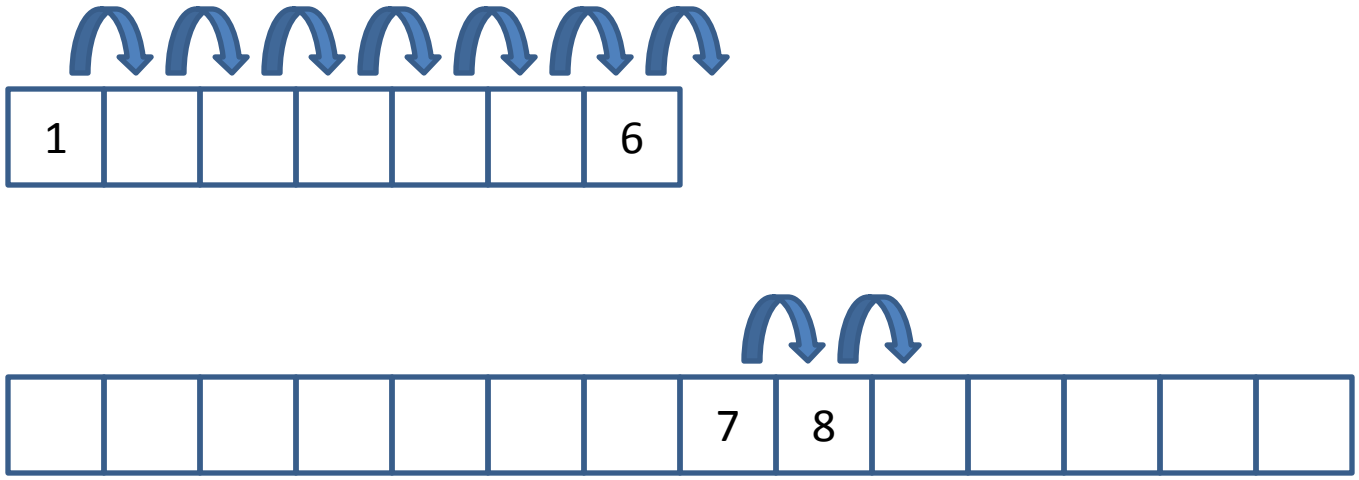
Deep Dive…

MIXING C# AND NATIVE C++

Performance For This Application



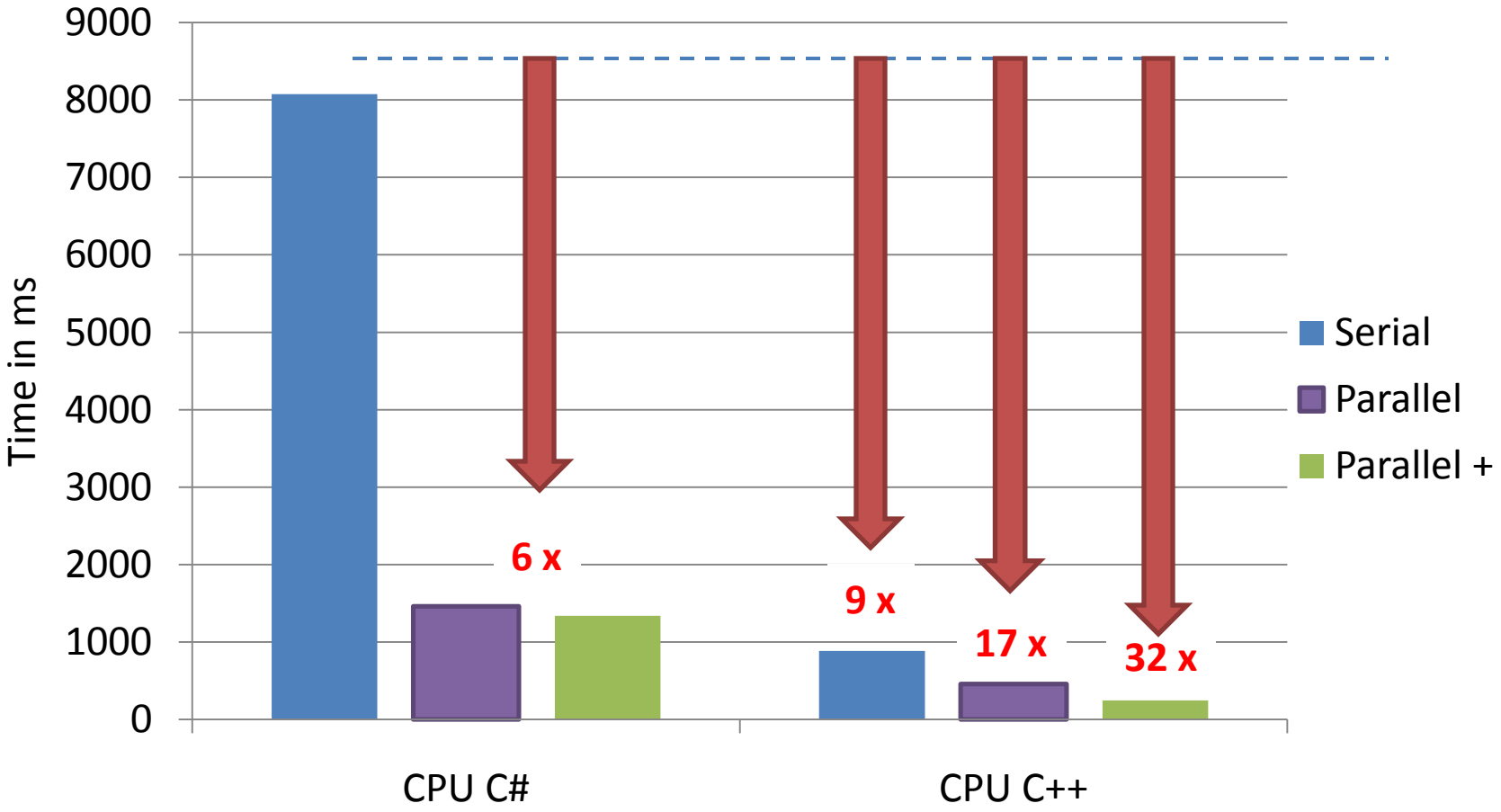
Average runtime for model with N = 3000 running for 16 iterations on a i7 920 or GTX 260



Deep Dive…

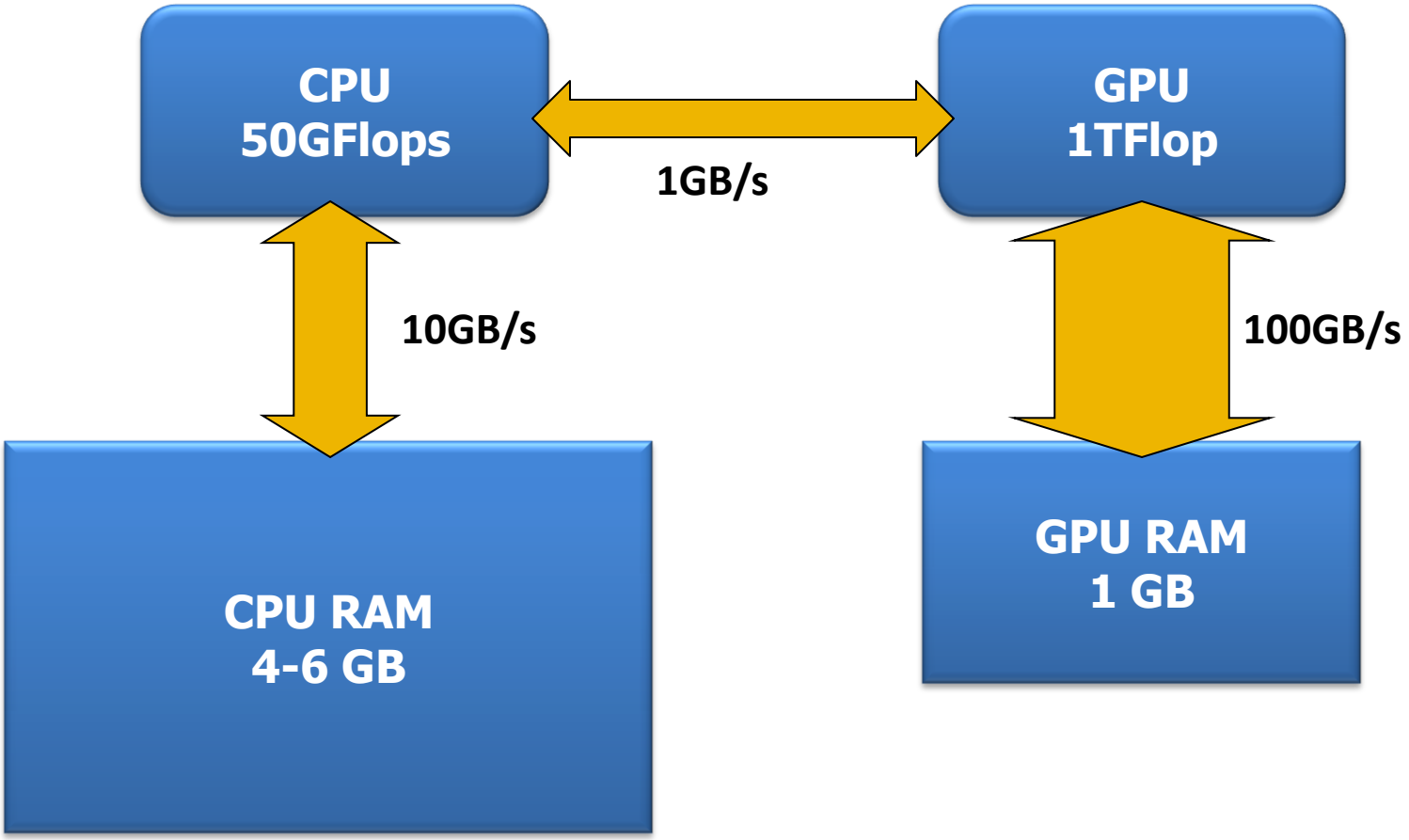
THINKING ABOUT CACHE

Performance For This Application

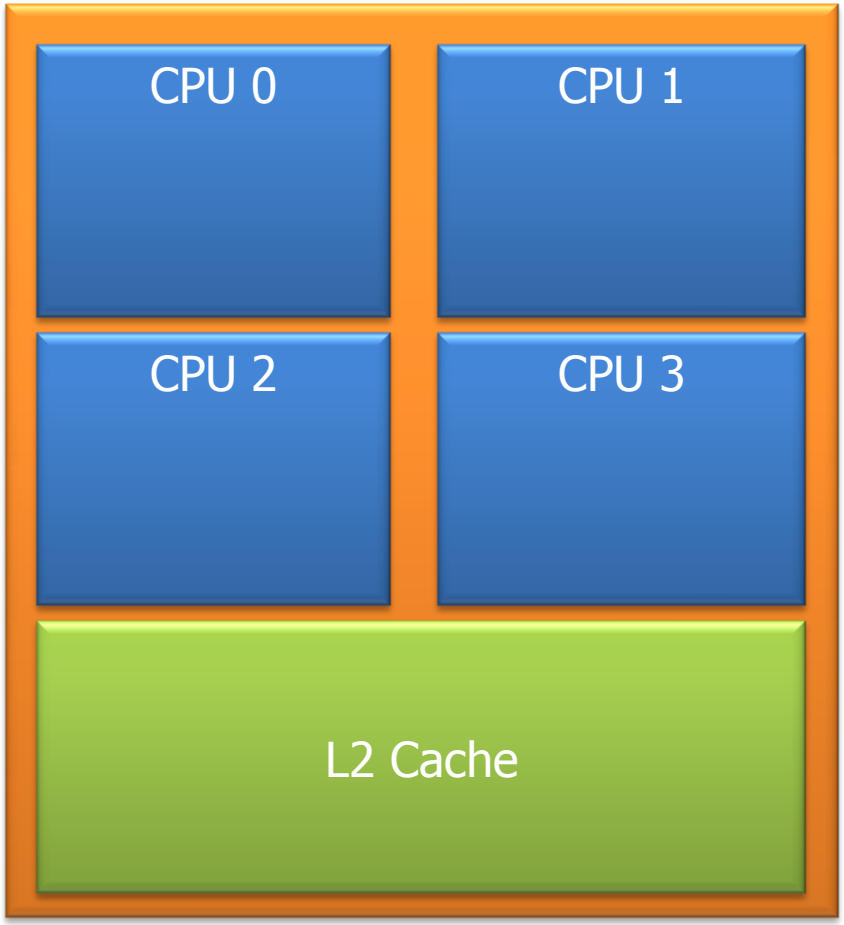


Average runtime for model with N = 3000 running for 16 iterations on a i7 920 or GTX 260

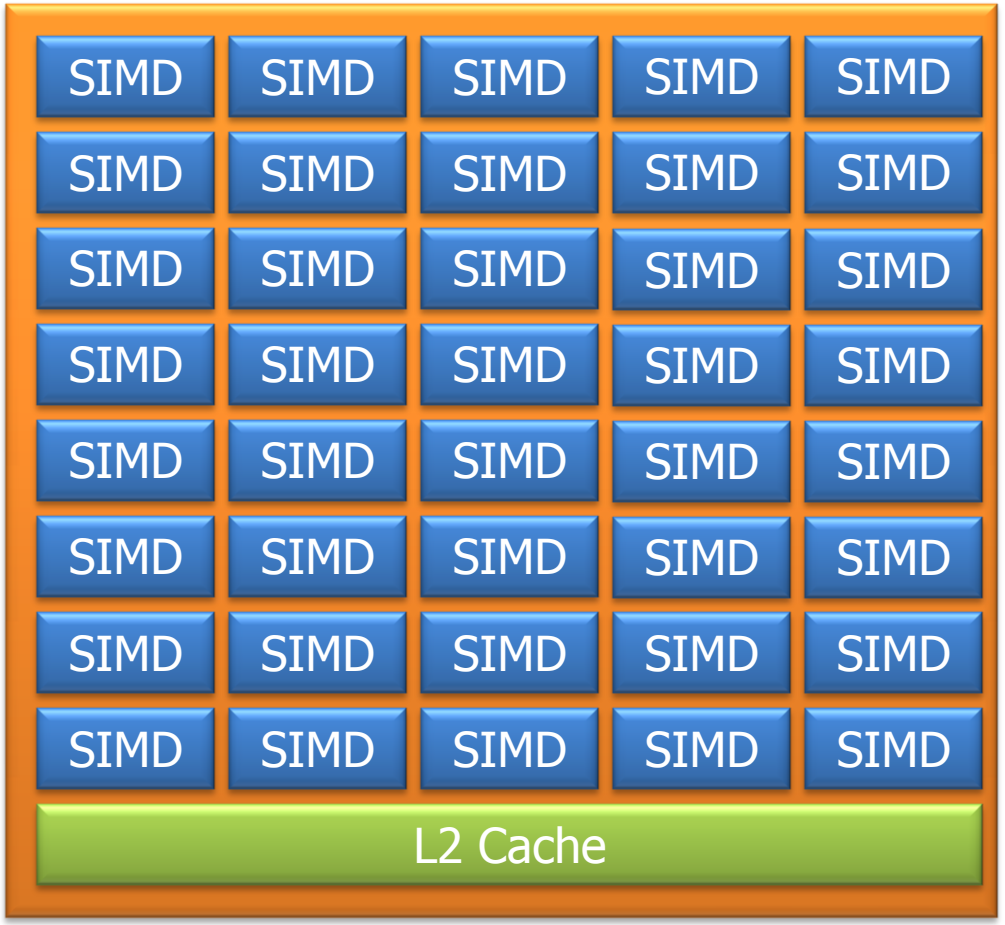
The Computer Under Your Desk



CPU vs. GPU



CPU

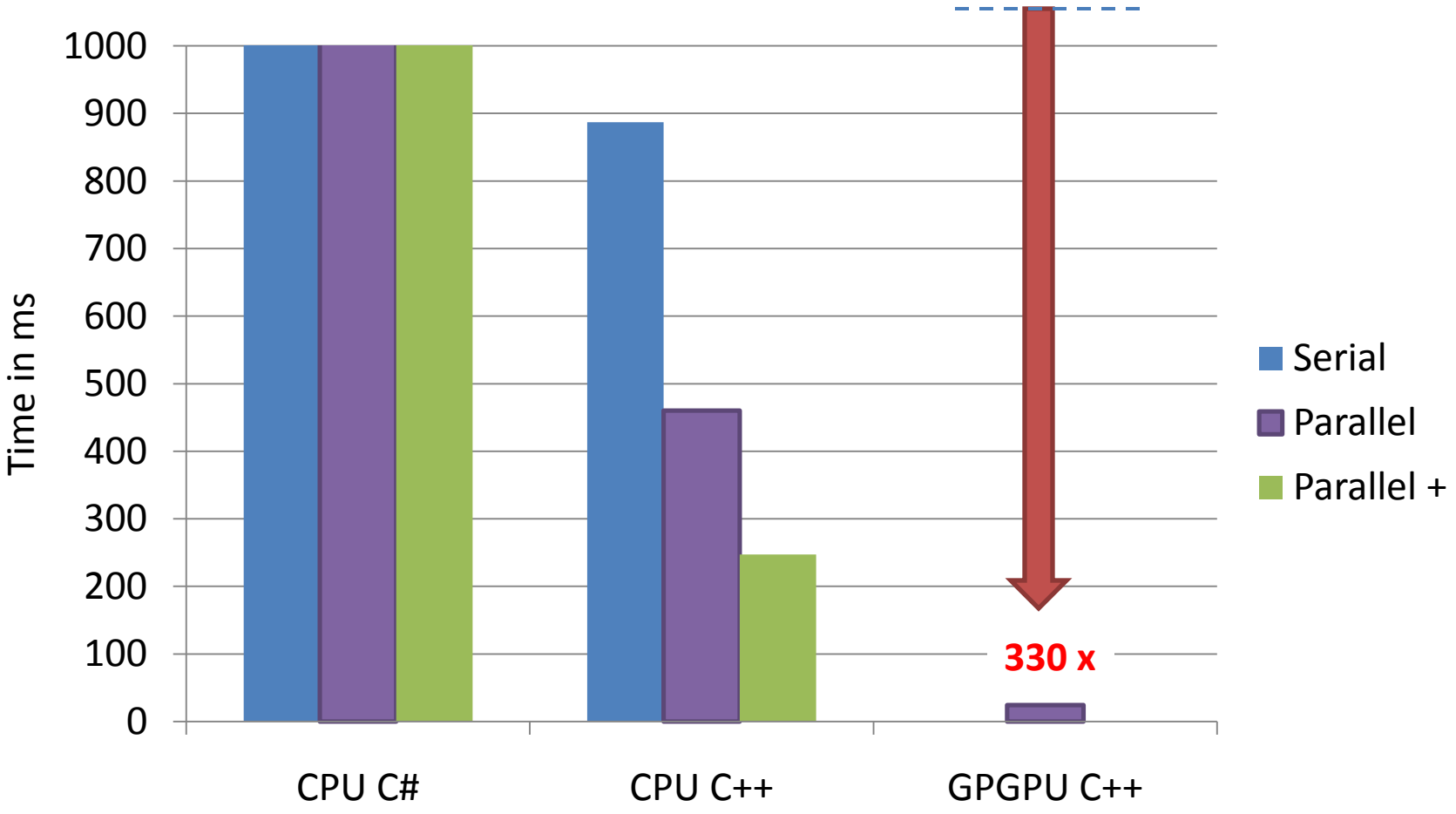


GPU

Deep Dive…

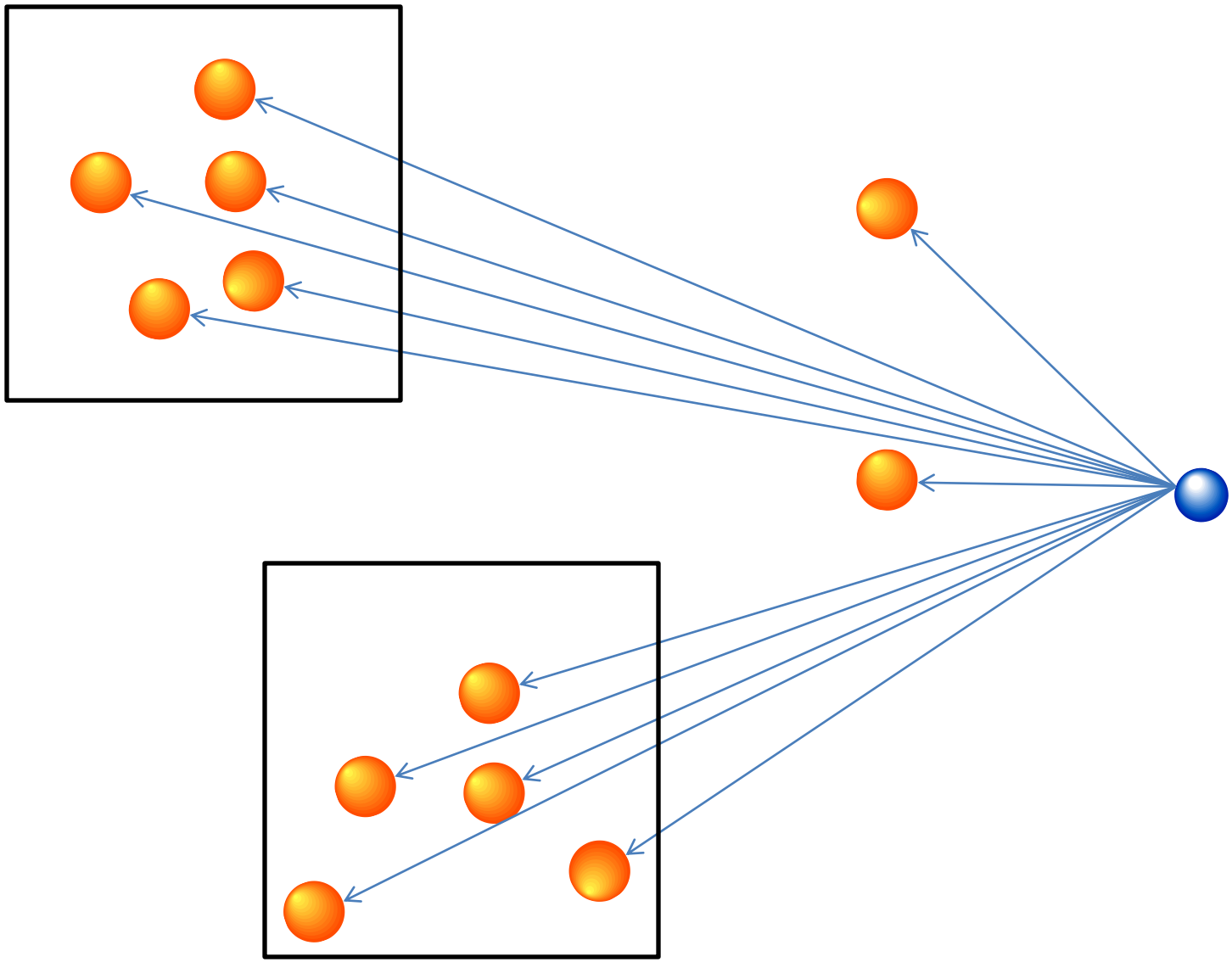
PROGRAMMING THE GPU WITH CUDA

Performance

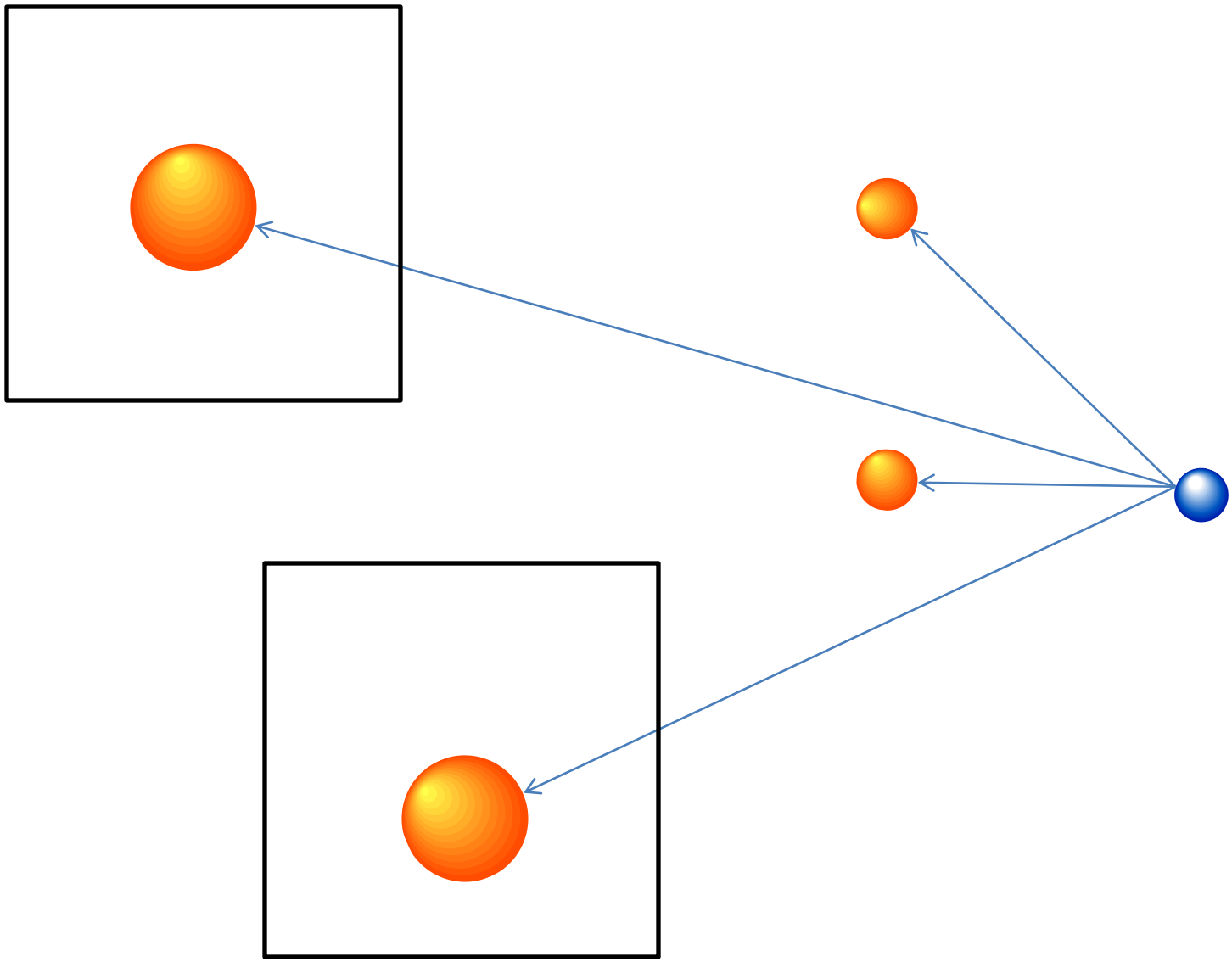


Average runtime for model with N = 3000 running for 16 iterations on a i7 920 or GTX 260

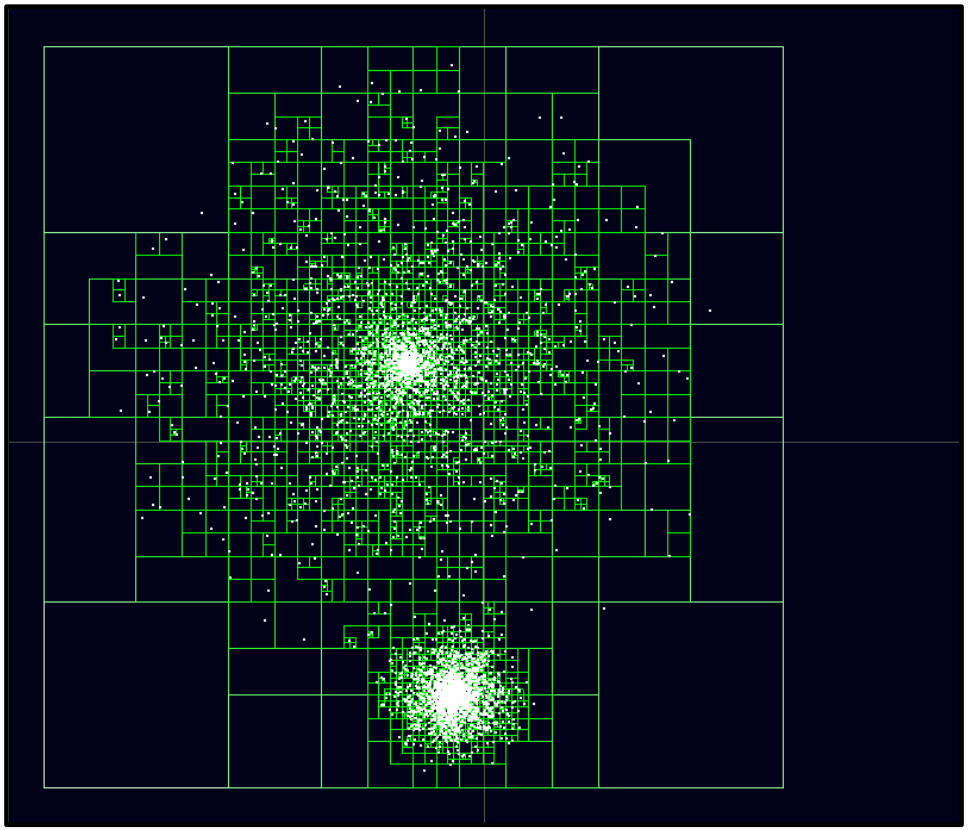
First Some More Physics…



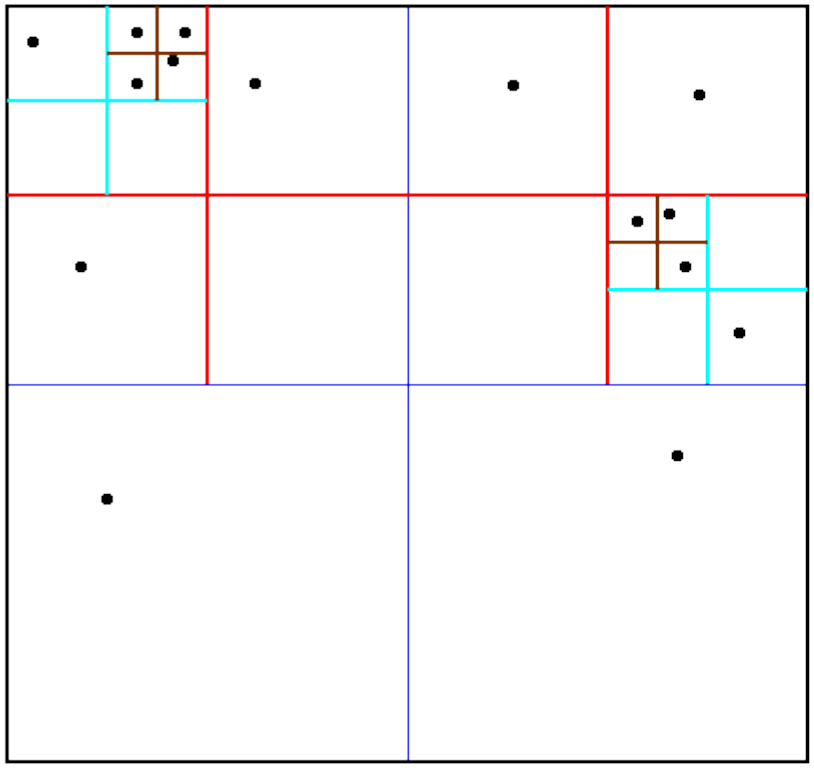
First Some More Physics…

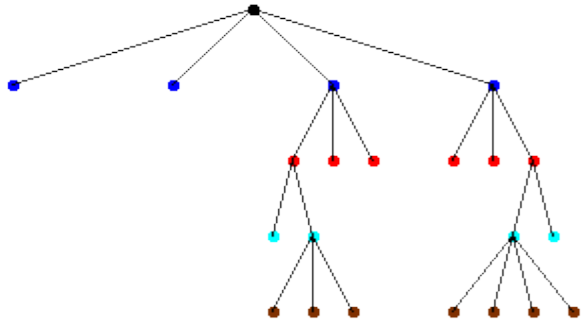


Barnes-Hut Model



Scales as: NLogN





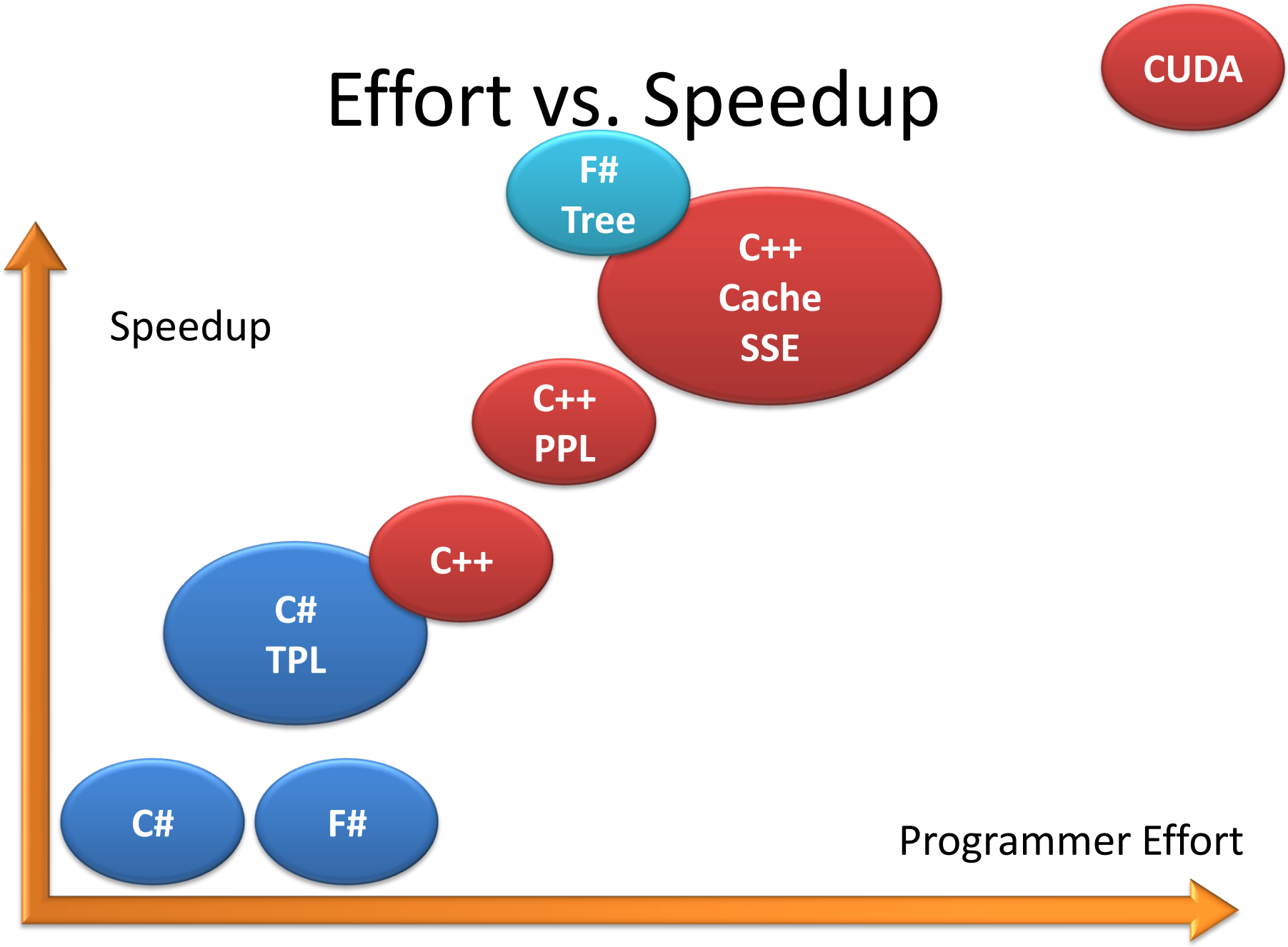
Deep Dive…

A PARALLEL TREE CODE IN F#

Conclusions

- Best tool for the job; C#, F#, C++
- Best hardware for the job:
  - CPUs good at branching
  - GPGPUs good at crunching (not branching)
- The TPL will do a lot for you but you still have to think

Effort vs. Speedup



Conclusions

"It's The Algorithm Stupid"

Good Hardware + Bad Code
= Bad Performance

Instructions are Cheap
Cache Misses are Expensive

That's All

Rats by… BANKSY

Resources

- The code – http://ademiller.com/nbody
  - Examples in C#, F# and C++
  - CUDA source soon (waiting for 2010 support)
  - GUI sometime soon (maybe)