Scalable Simulation for Systems Medicine

Adam Sampson
CoSMoS, CRISP, IAMG
University of Abertay Dundee
The starting point

• Models of cell behaviour
  – e.g. our cancer cell models: 60+ differential equations, calibrated against real data
• We know how to run simulations of these models
  – … but they only tell us about a single cell
• We want to experiment at the tissue scale
The bulk approach

• Build a new, simplified model of the larger system
• … but that can only include the high-level behaviours that we already expect to find
• … and often it assumes that all cells develop at the same rate, have the same properties, etc.
• It probably won't tell us anything surprising
Combining individuals

- Instead, build larger simulations from lots of independent single-cell simulations
- Model the **interactions** between cells
  - Spatial occupancy, chemical environment...
- Each cell can develop differently – not soup
- Now higher-level behaviours can **emerge**
Scaling up

• But we need to simulate **enough** cells to be able to reproduce the behaviours we're interested in...

• How many is enough?
  - As big as the real system we're measuring?
  - Just big enough to reproduce the measurable behaviours?
  - Use CoSMoS validation techniques

• How do we harness sufficient computing power?
Concurrent Programming

• My background is in **concurrent programming**
  - Building software that has many activities going on and interacting with each other at the same time
  - e.g. network servers, robotic control systems, multiplayer games, media processing...

• Lots of current work on concurrent techniques, languages and runtime systems
  - e.g. Erlang, Go, TBB, GCD
  - ... and much more that's still being developed
Why go concurrent?

- Because complex systems are **naturally concurrent**
  - They consist of many interacting entities
  - ... where each entity has its own behaviour and developmental state

- These systems can be **modelled** conveniently using concurrent approaches
  - Each entity has its own flow of control
  - Interactions are made explicit
  - No artificial ordering of events
Why go concurrent?

- Because modern computers have **parallel** processors: they can work on many tasks at the same time
  - Multicore, hyperthreading, SIMD, GPUs
- “Parallel programming is hard”...
- .. but using concurrent design techniques **exposes** the bits that can be done in parallel
- Concurrent **runtime systems** can then work out automatically how best to execute your program using multiple CPU cores
Distributed computing

- Sometimes one computer isn't enough
- We need to use a cluster of networked computers, or rent time on a cloud system
- How do we break up our simulator into multiple parts that communicate over a network?
  - ... without slowing it down too much
**Message-passing**

- A common concurrent programming approach: activities interact by sending and receiving messages
  - … which works just as well over a network as it does between cores on a single computer
- The network has **much** higher latency...
  - … but a smart runtime system can run other activities within the simulation while it's waiting
Scalability

- **Scalability** is more important than straight performance
- It's OK to be a bit slower on a single core if it lets you use 8 cores – overall it's much faster!
- Get the scalability right, then optimise out the constant factors...
Simulating space

• Efficiently simulating **spatial interaction** has been a particular concern within CoSMoS

• Adapting techniques used in games, we model regions of space as concurrent entities
  
  – Near-linear scalability: just add more machines
  
  – Works for both real and abstract ideas of space
measure, model → cell
cell measure, model simplify tissue
measure, model

simplify

parallelise

tissue

cell

cell  cell  cell

cell  cell  cell

cell  cell  cell
cell
measure, model

simplify

parallelise

validate

cell
cell
cell
cell
cell
cell
cell
cell
tissue
parallelise
Where next?

- This summer:
  - Using CoSMoS space techniques to simulate physical interactions between cancer cells
  - Combining existing models of blood flow and platelet signalling to study drug effects on clotting
- Multi-scale interaction and visualisation
- “Sloppy time” to improve simulation performance
- Interactive simulation using cloud resources
- Any questions?