Event-driven simulation

Andrew Brown
Southampton
adb@ecs.soton.ac.uk

Partners

UNIVERSITY OF CAMBRIDGE

ARM

CCFE
CULHAM CENTRE FOR FUSION ENERGY

Imperial College London

e-Therapeutics plc
The Network Pharmacology Company

CU Medicine
HONG KONG

Newcastle University

Imagination

UNIVERSITY OF Southampton

nmi
Semiconductors to Systems
CONNECTED COMMUNITIES

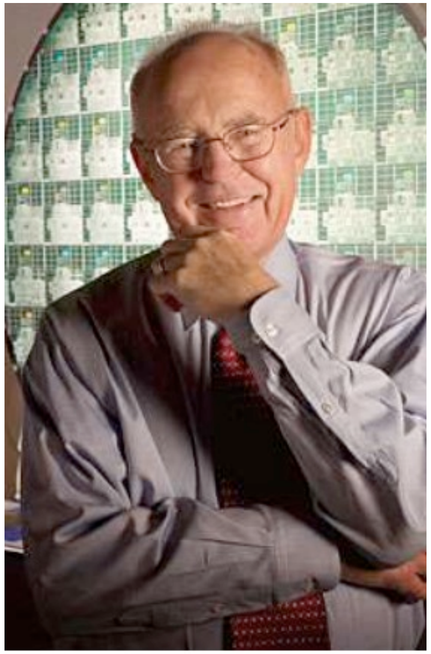
nag®

Outline

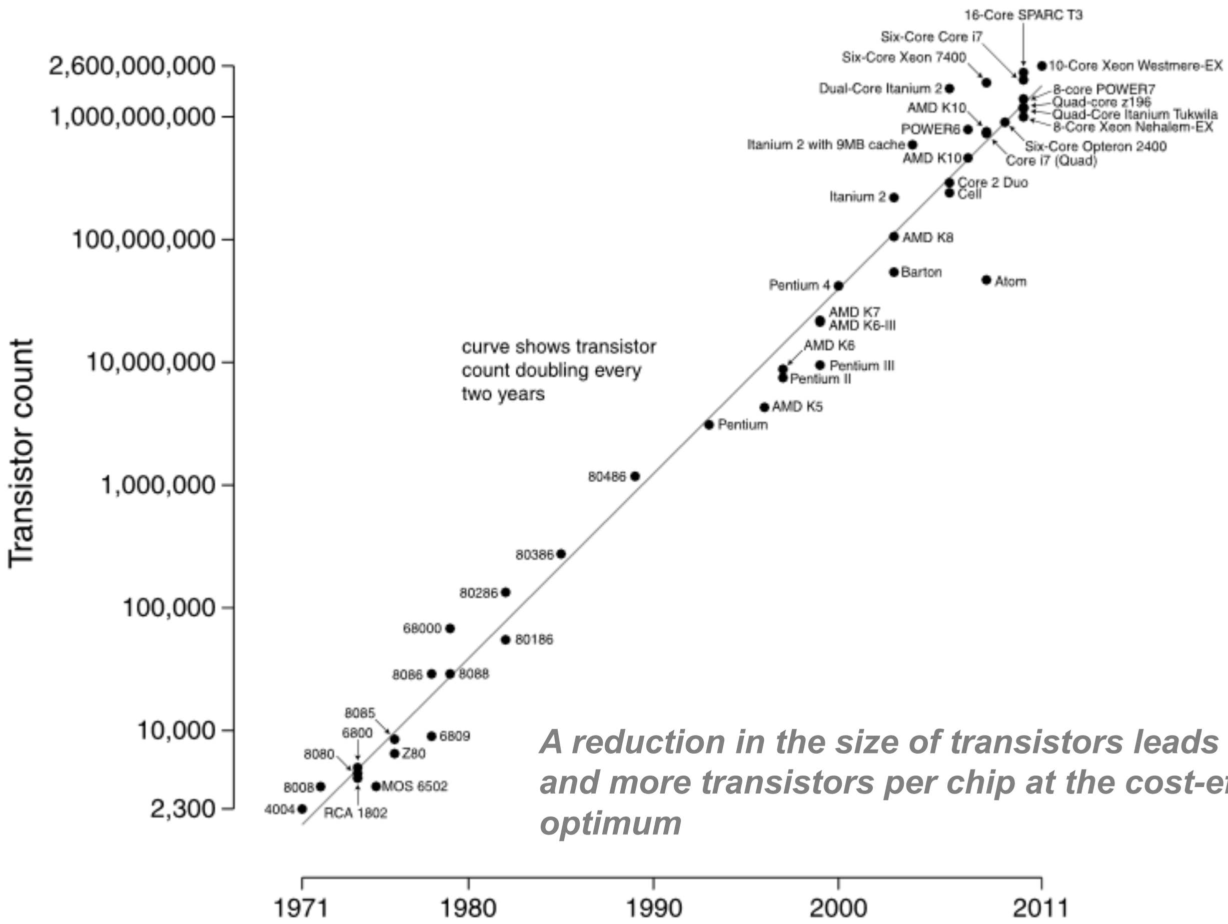
- The rise and asymptote....
- Simulation - there's a lot of it about
- Event-based simulation
- What is the user base?
- Where does all the time go?
- Down amongst the Hard Sums
- A brief meander into reliability
- Some pictures of hardware

Moore's Law

POETS



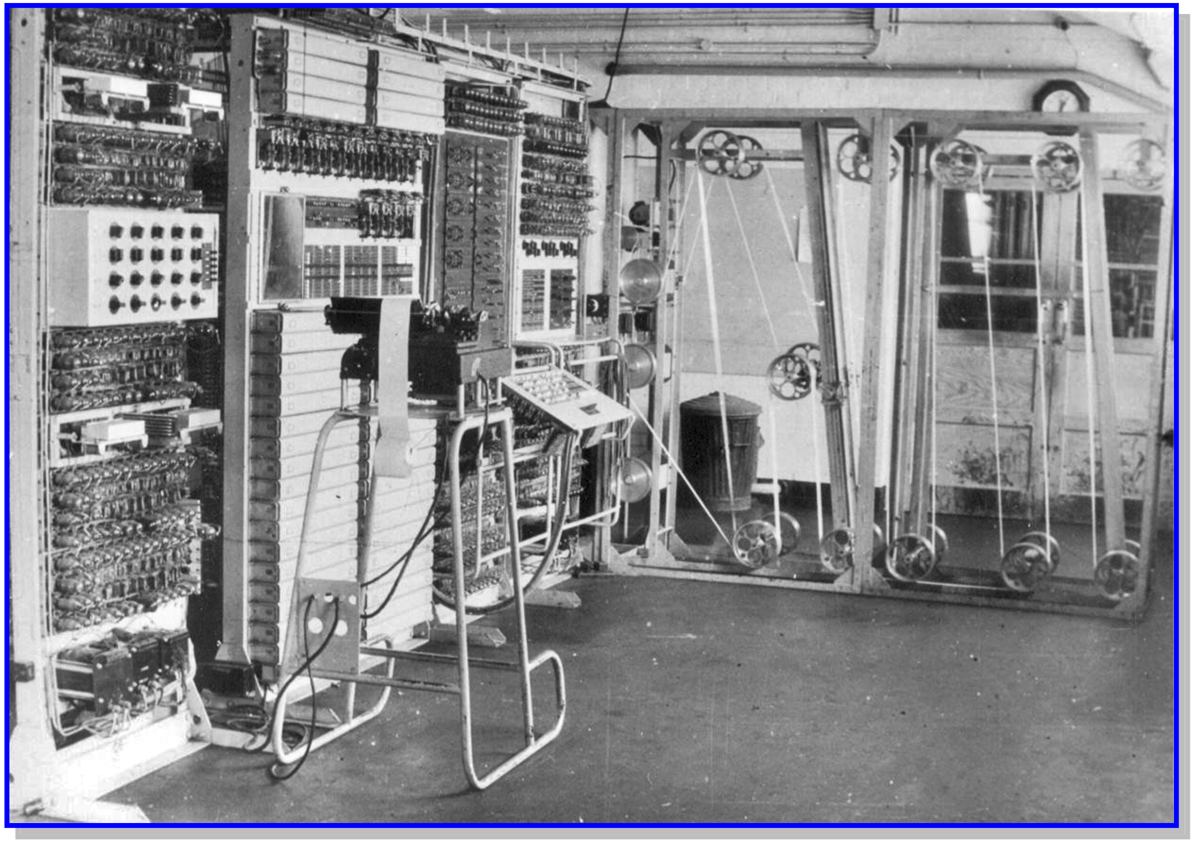
Microprocessor Transistor Counts 1971-2011 & Moore's Law



***"Transistor density of semiconductor chips will double every 18 months"** – Gordon Moore, co-founder of Intel, 1965*

A reduction in the size of transistors leads to more and more transistors per chip at the cost-effective optimum

Colossus (1945)



- Not a stored program machine
- State space search engine
  - Very coarse grained conditionals:

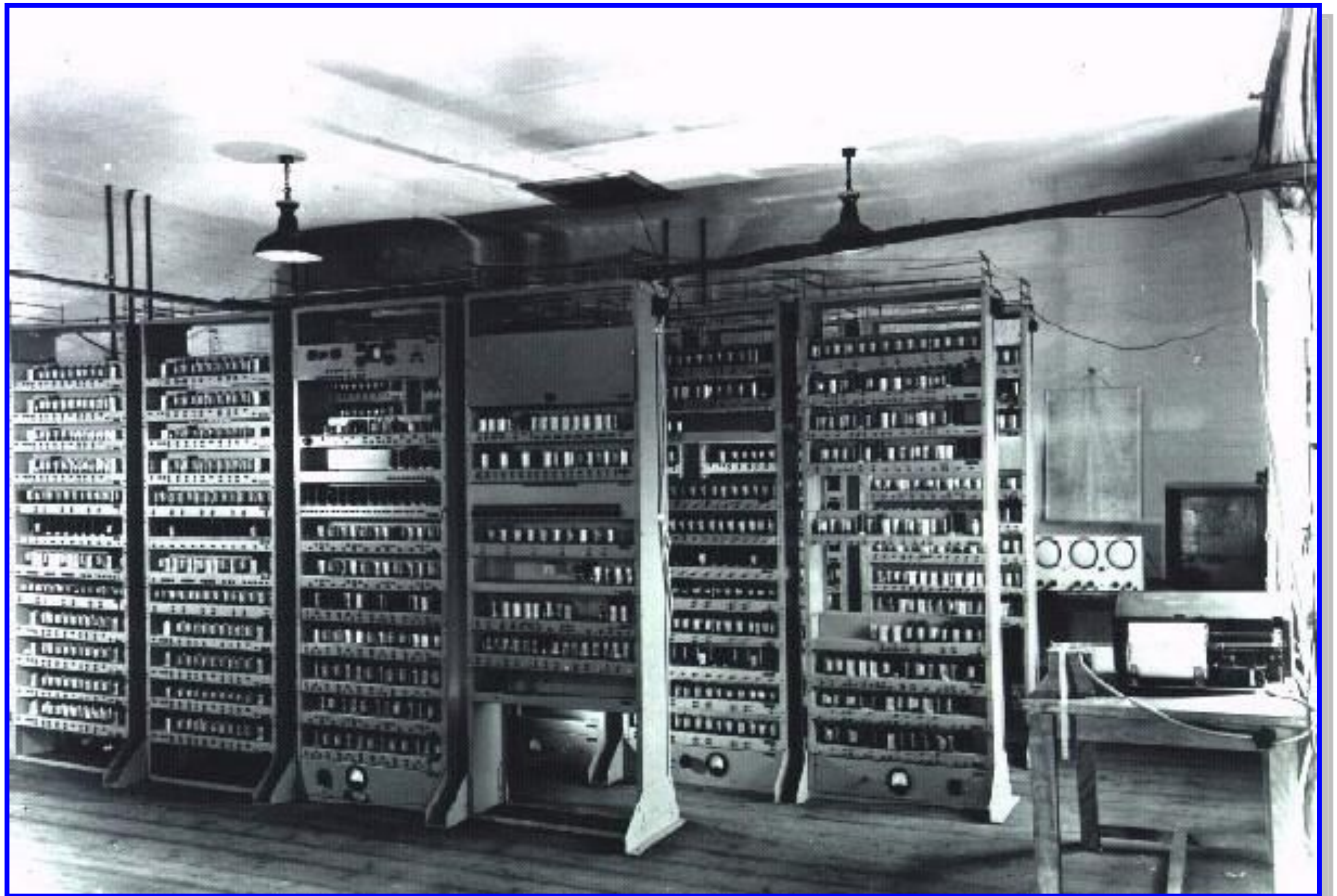
    if *this* **then** *stop|ring bell|print*
  - Not:

    if *this* **then** *change search path*

Baby (1948)

- Never intended as a practical computer
- Test harness for the Williams tube
    - First random-access memory
    - 32 words of 32 bits
    - 3.5 kW
    - 700 instr/sec
    - HW instruction set: subtract, negate

EDSAC (1949)

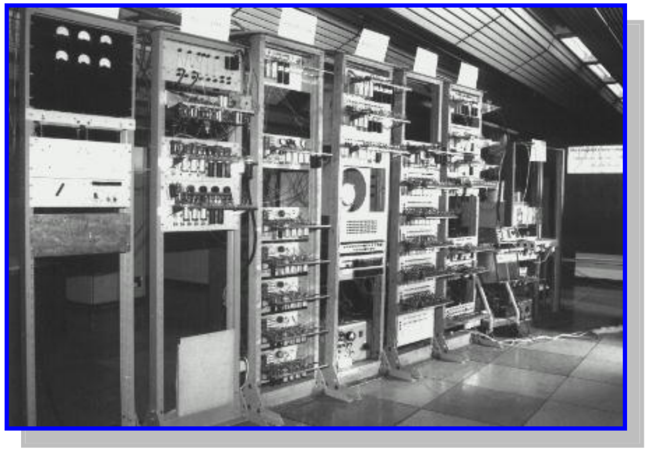


- First "real" computer:
- Instruction set contained control *and* arithmetic operations
- 512 18-bit words
- Mercury delay line memory
- 11 kW
- 1MHz clock
- 650 instr/sec
- (150 mult/sec)

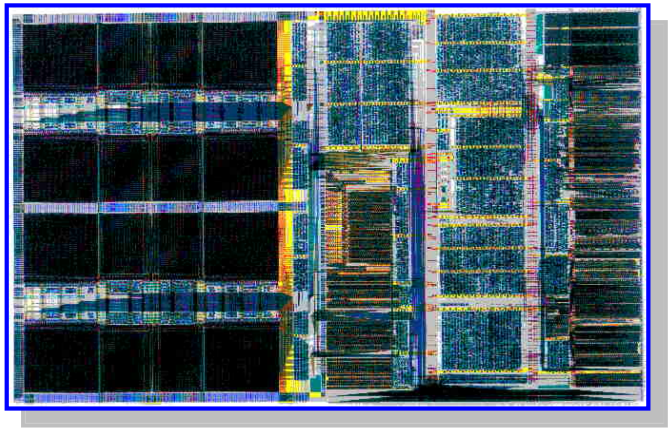
70 years of progress

POETS

- Baby:
  - Filled a medium-sized room
  - Used 3.5 kW of electrical power
  - Executed 700 instructions per second
  - ~ 5J/instr



- ARM968:
  - Fills 0.4mm^2 of silicon (130nm)
  - Uses 20 mW of electrical power
  - Executes 200M instructions per second
  - ~ 100 pJ/instr

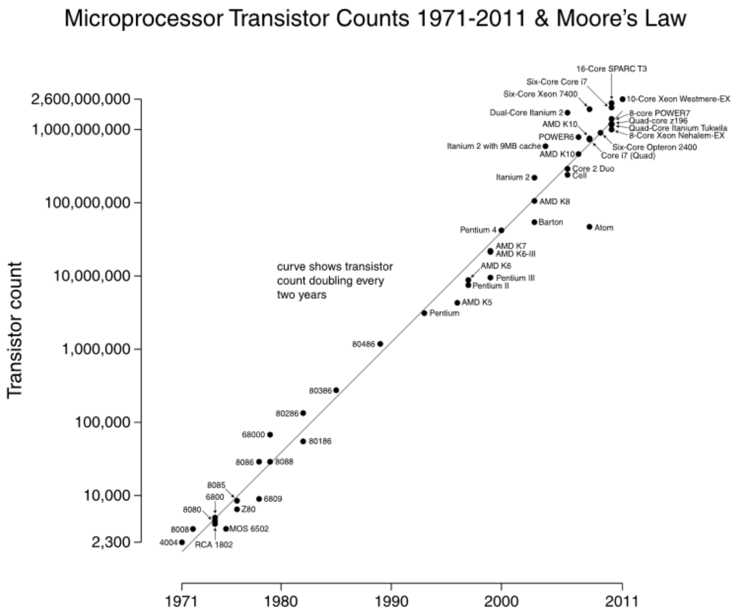


200000000 years of progress

- Brains demonstrate
  - Massive parallelism (10^{11} neurons)
  - Massive connectivity (10^{15} synapses)
  - Uses ~20W
  - Processes ~ 10^{15} events/sec
  - ~ 10^{-14} J/event
  - Low-performance components (~ 100 Hz)
  - Low-speed communication (~ metres/sec)
  - Adaptivity – tolerant of component failure
  - Autonomous learning

Moore's Law

POETS

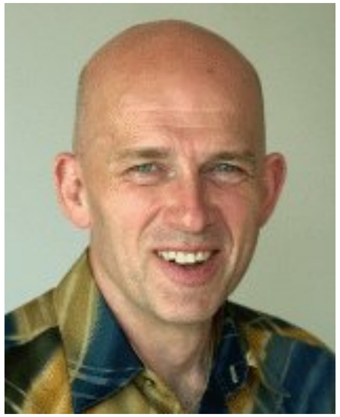


- Is *not* a fundamental law
- Is *not* a verifiable hypothesis
- Is *not* an extrapolated prediction
  - » Well, it sort of is, actually...
- Is *not* a philosophy

- It *is* a business model
  - Arbitrarily created
  - Trivially discarded

May's Law

POETS



David May, lead architect of the Transputer (Inmos), designer of Occam, founder of Xmos

- Software efficiency halves every 18 months, compensating Moore's Law
  - A mixture of
    - Shortage of skills
    - Adding too many features
    - Copy-paste programming
    - *Massive* overuse of windows and mouse-clicks
    - Reliance on Moore's law to solve inefficiency problems

Reality....

- No exponent is sustainable indefinitely in nature

- The rate of change indicated by Moore's Law indicates that in ~ 150 years there will be more memory cells / cm^2 of silicon than there are atoms in the universe

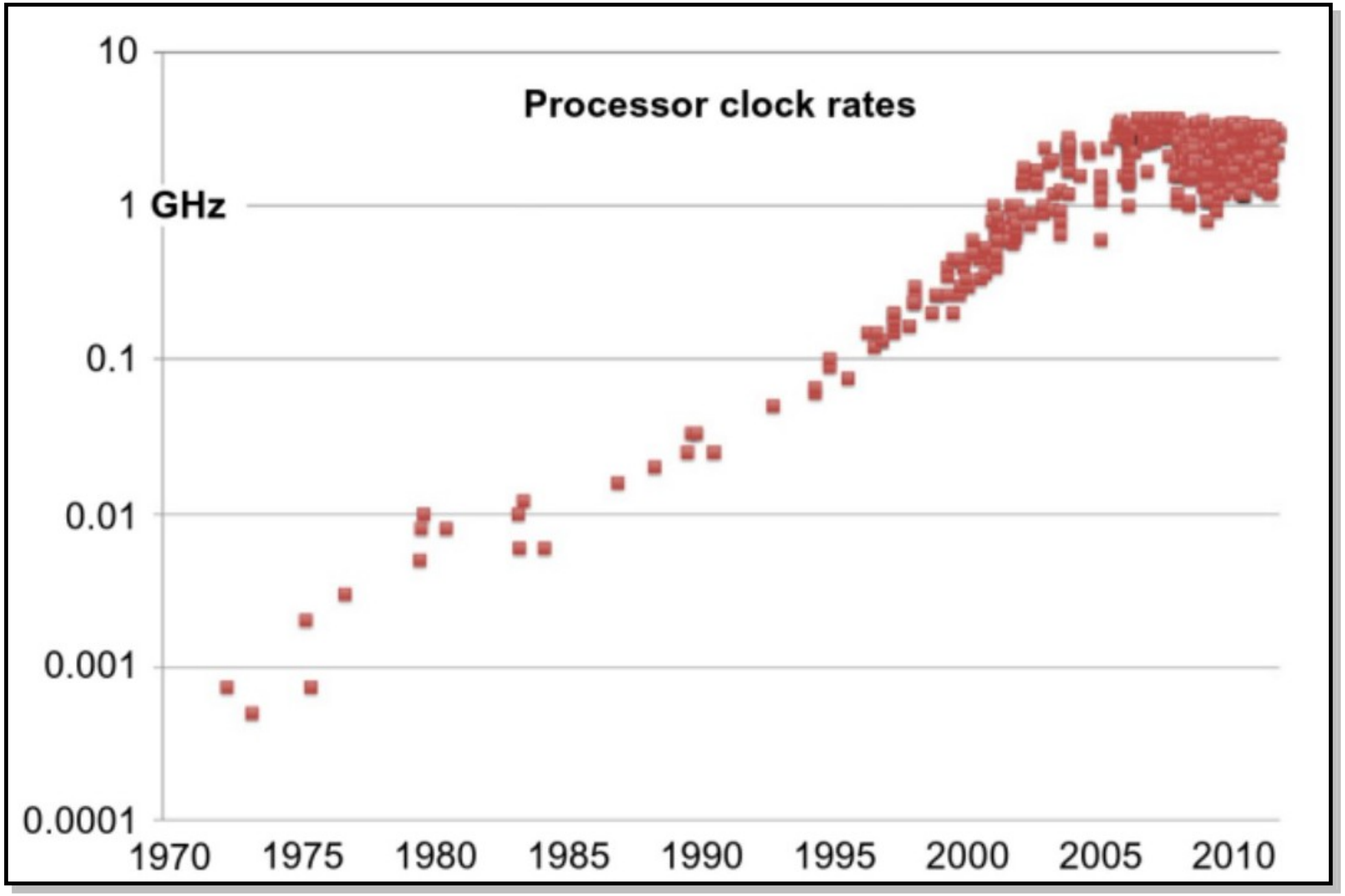
- Something is going to break
  - What?

Roadblocks

POETS

- Everyone has their favourite:
    - Device physics
    - Process spread
    - Interconnect
    - Economics
    - Design
    - Power dissipation
- You can run, but you can't hide
- *It just isn't worth it*

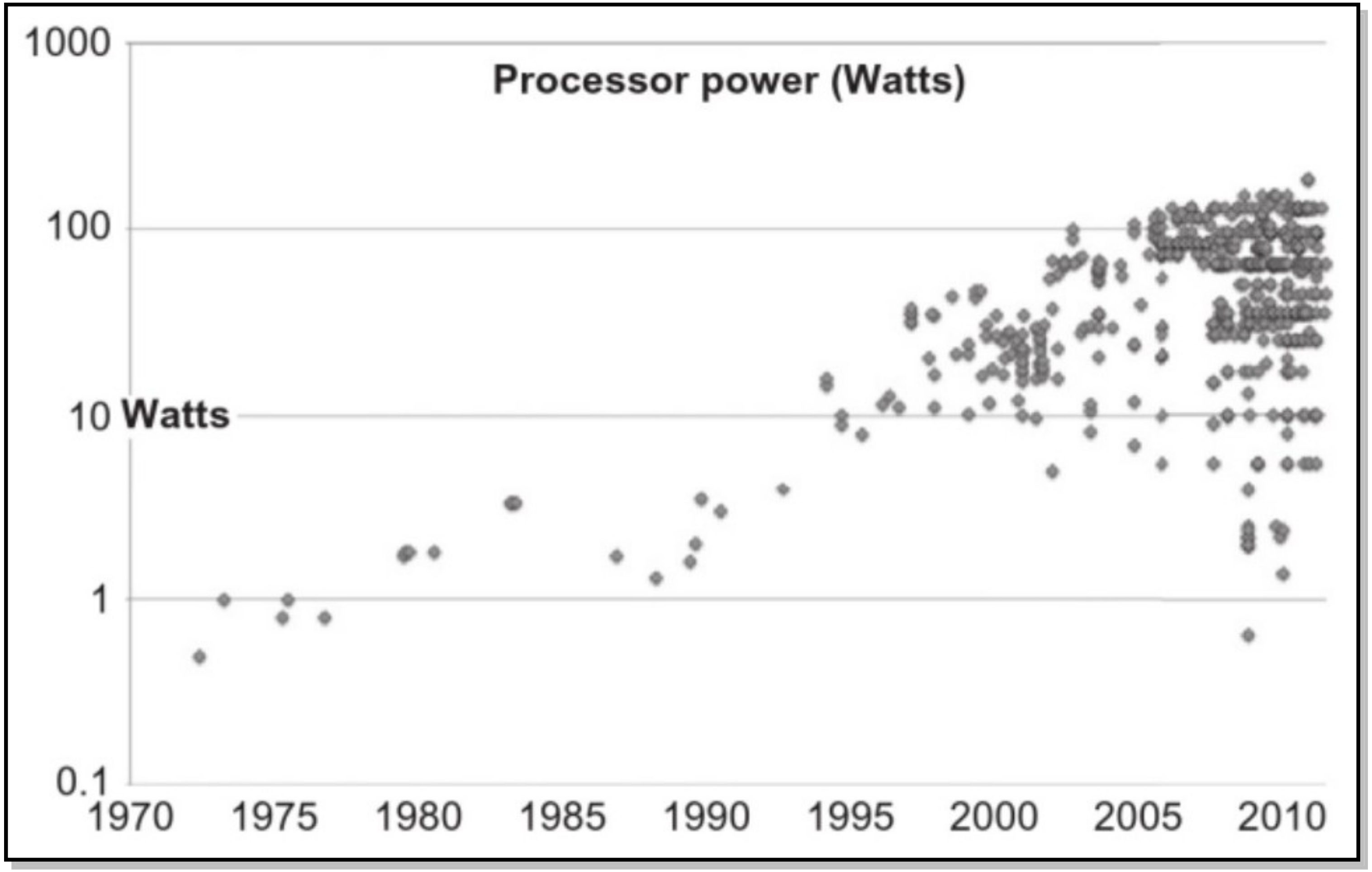
Relativity



(Structured parallel programming, McCool, Robison, Reinders)

Thermodynamics

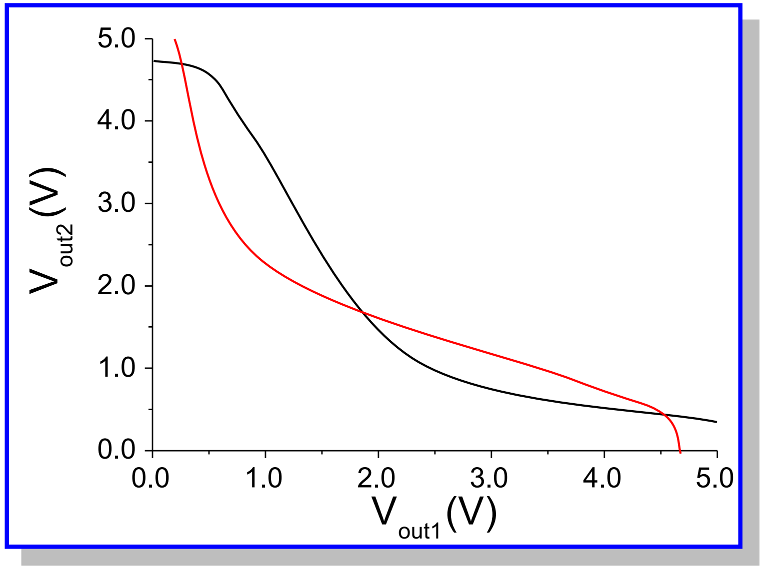
POETS



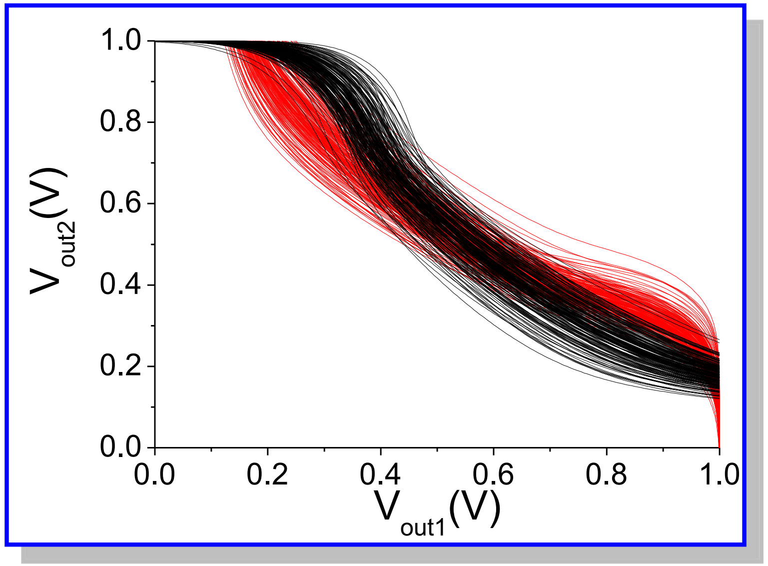
(ibid.)

Device variability

POETS



...becomes...



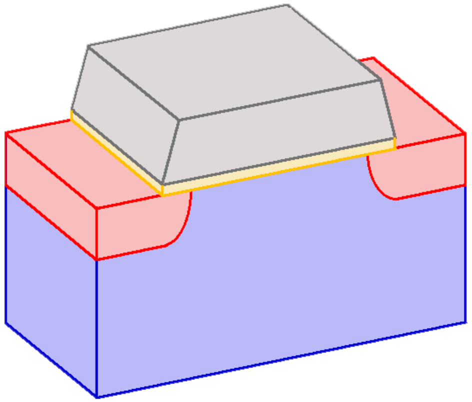
DEVICE MODELLING GROUP

UNIVERSITY of GLASGOW

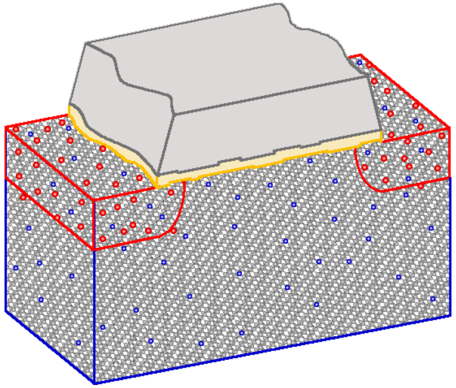
Simulation models

POETS

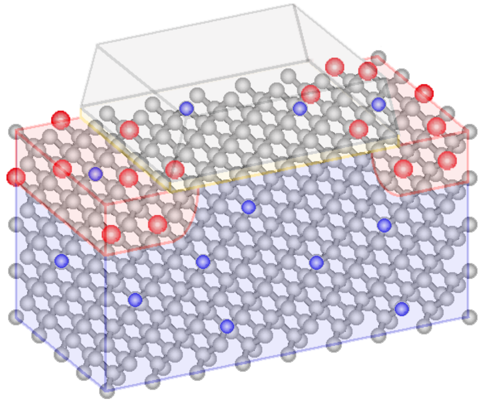
...become discrete...



Continuous simulation model



22 nm MOSFET 2008



4.2 nm MOSFET 2023

DEVICE MODELLING GROUP

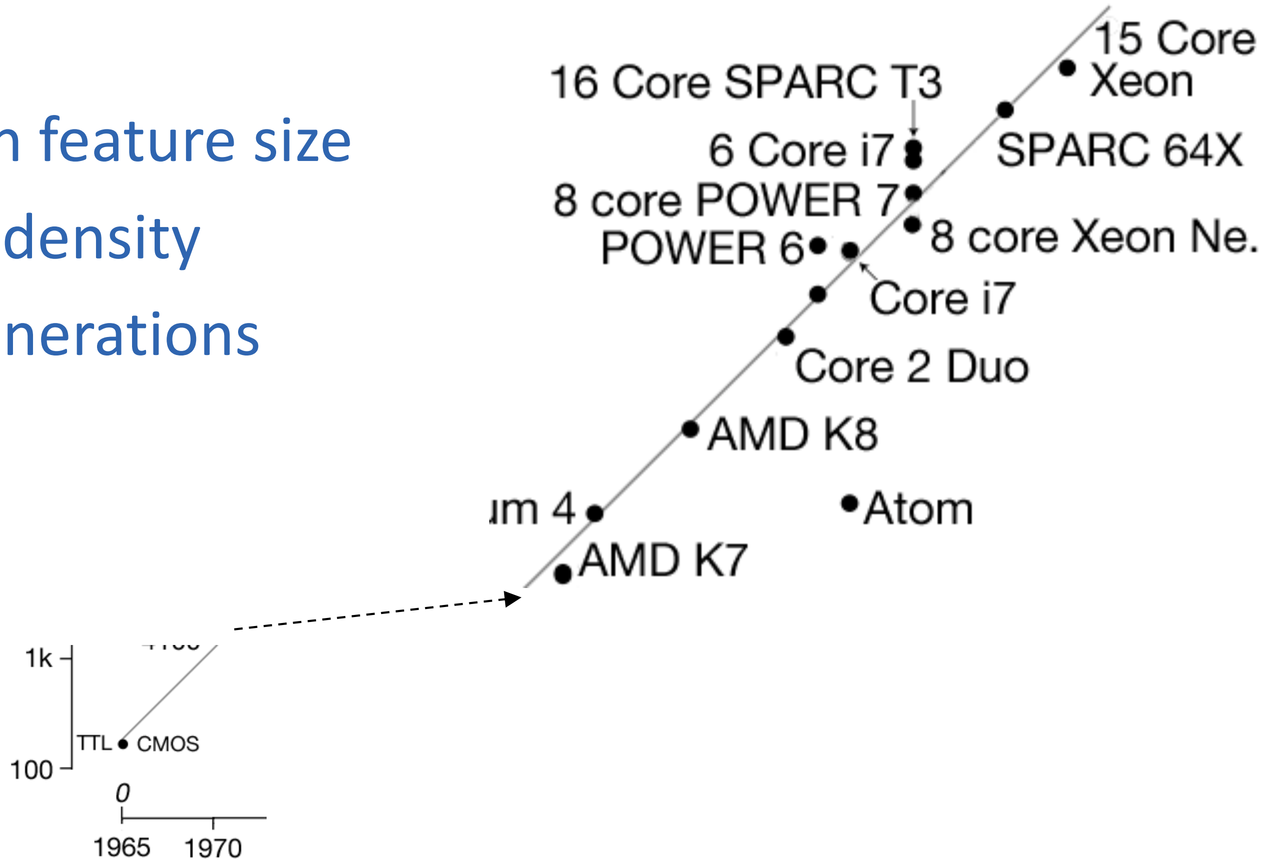
UNIVERSITY of GLASGOW

Cores are the new transistor

POETS

–Silicon

- 10um .. 10 nm
- 2^{10} reduction in feature size
- 2^{20} increase in density
- ~20 process generations



The rise of the many-core exponent is just beginning

(Elektor 26 May 2015)

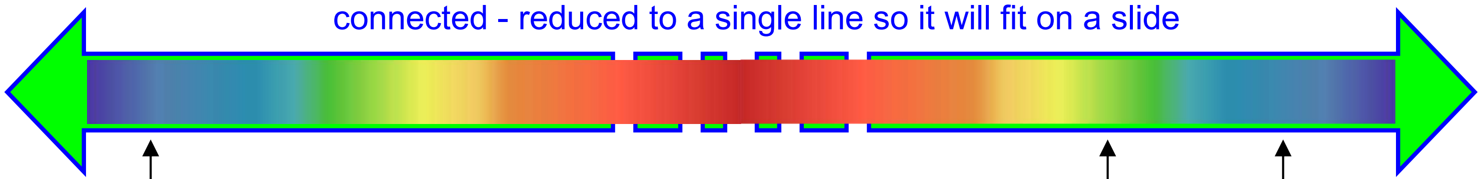
Computers do everything badly

- ...do everything badly

They are general purpose machines....

...designed in the absence of any knowledge of their intended application domains

In simulation, the architecture almost never matches the application dataflow



Why is that, then?

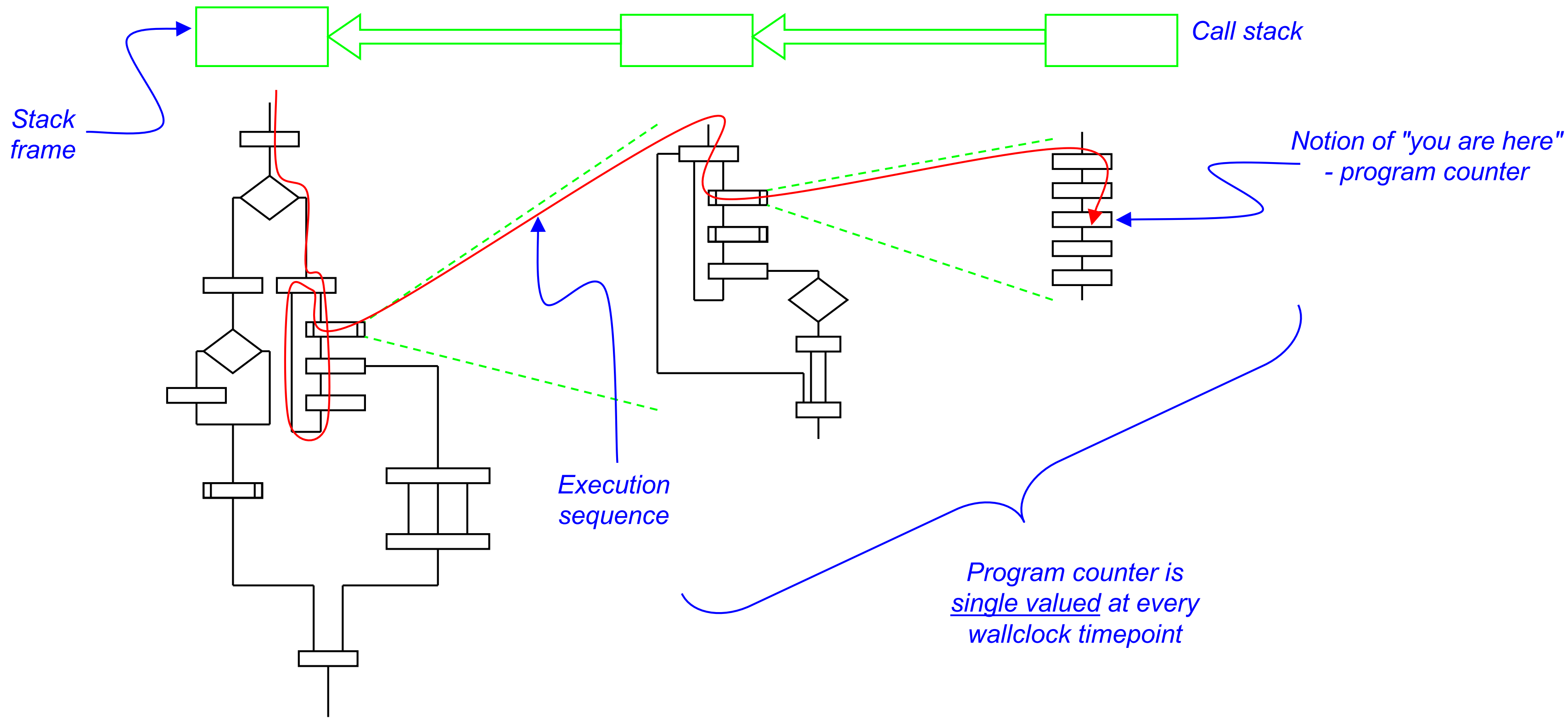
POETS

- *Unnatural* serialisation:
  - Everything gets shoved through the ALU bottleneck
  - Instruction ordering is deformed...
    - Out-of-order
    - Speculative execution
    - Code transformation
    - Multiple layers of cache
  - ...to the point of (but not past) breaking semantics
  - Nature doesn't do it this way

A complex sequence....

POETS

....of operations and instructions



- *Most* of the time, *most* of the code is doing nothing
  - Waiting for control to arrive

OK, let's go parallel

POETS

- Programming 101: Communicating sequential processes
  - Message passing:

1. *Avoid* passing messages
2. If you *must* pass messages

   ...make them big

   ...don't choreograph the code so that half the processes spend most of their time waiting for the other half to send them messages
3. From the above

   ...choreography is your problem

   ...it's hard

Is there a better way?

- Sequential instruction execution is not the only model of computation
  - Abandon it
  - Alternatives:
    - Massive parallel computing resource on a FPGA
    - Vast complex of biological neurons inside brains
    - The next exponent: *cores are free*
- Minor difficulty:
  - We do not (yet) have any general theory of computing on huge, distributed, networked systems
  - That's what research is for

There <u>is</u> a better way

POETS

- Sequential computing is not
  - Natural
  - Efficient

- The *only* thing it has going for it
  - It's easy

Outline

- The rise and asymptote....
- Simulation - there's a lot of it about
- Event-based simulation
- What is the user base?
- Where does all the time go?
- Down amongst the Hard Sums
- A brief meander into reliability
- Some pictures of hardware

Simulation

- The **Art** of simulation: to predict reality

1. Construct a computational model
    - Mathematical abstraction
    - Babies and bathwater

2. Solve the model
    - Finite word length machines
    - Numerical methods

What is a simulation?

POETS

- It is *not*
  - The analytical solution of a set of equations
- It *is*
  - A computer experiment of the behaviour of a physical system in which reality is replaced by mathematical constructs that interact in ways that mimic the interactions of the physical system, and where the evolution of the model mimics states equivalent to those found in reality

It's a miracle it ever works...

POETS

- We do not simulate real systems, we simulate *models* of them

1. We construct the model
   - And we leave things out / get things wrong
     - Because we don't know what's important
2. We adapt it for calculation in a machine
   - We model real values with non-linearly discrete approximations

Why simulate?

- Experiments are
  - Expensive
  - Dangerous
  - Time-consuming
  - Hard to instrument
  - Hard to control
  - Difficult to reproduce
  - Constrained by reality
  - Easy to misinterpret
  - Sometimes unethical

- Simulations are
  - Cheap(er)
  - Safe
  - Fast(er)
  - Easy to instrument
  - Easy to control
  - Easy to reproduce
  - Constrained by platform
  - Easy to misinterpret
  - Seductive

Caveat emptor

POETS

- The simulation says....
- The simulation is reproducible....

Does not mean

- The simulation is correct

And

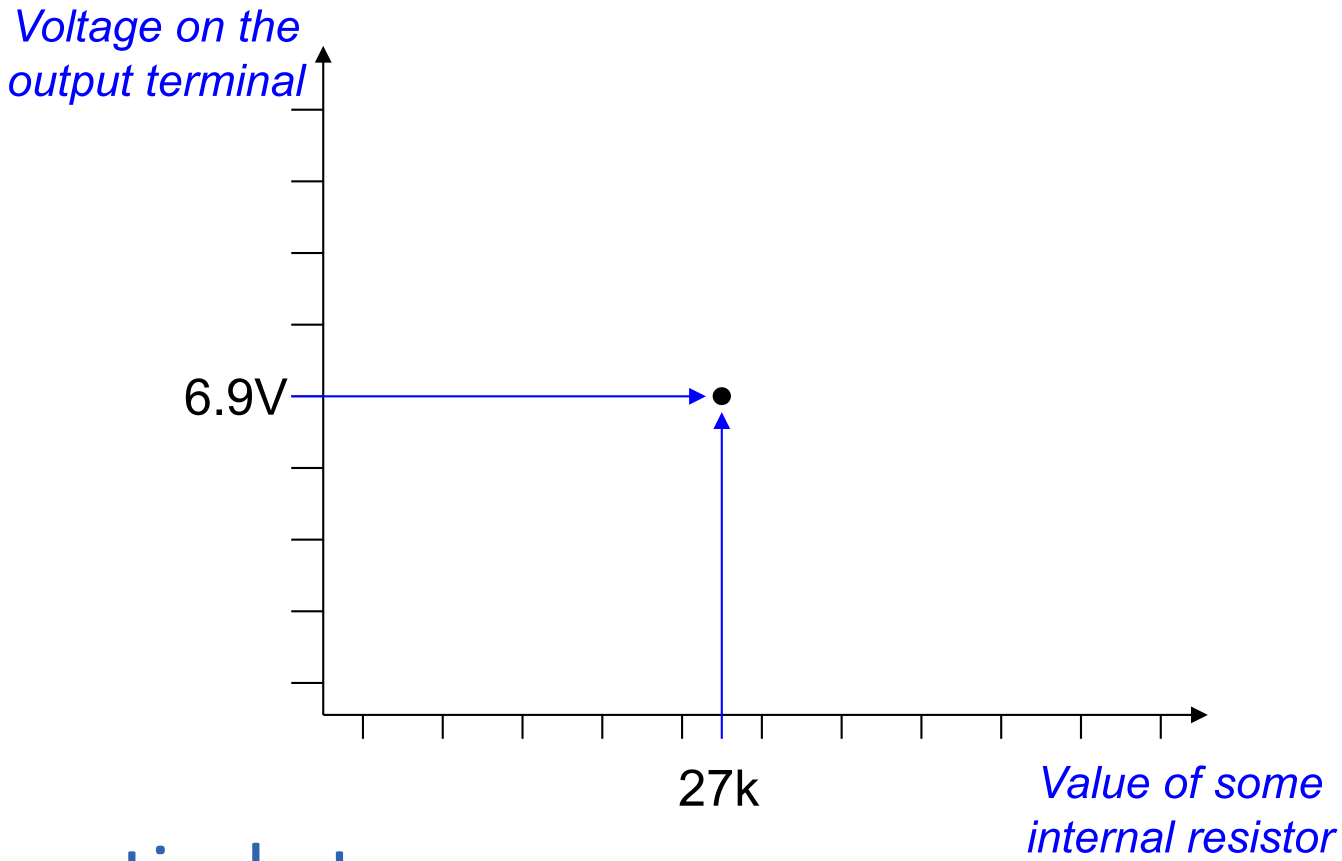
Does not mean

- The simulation is useful

Sensitivity

POETS

- What the simulator tells you:

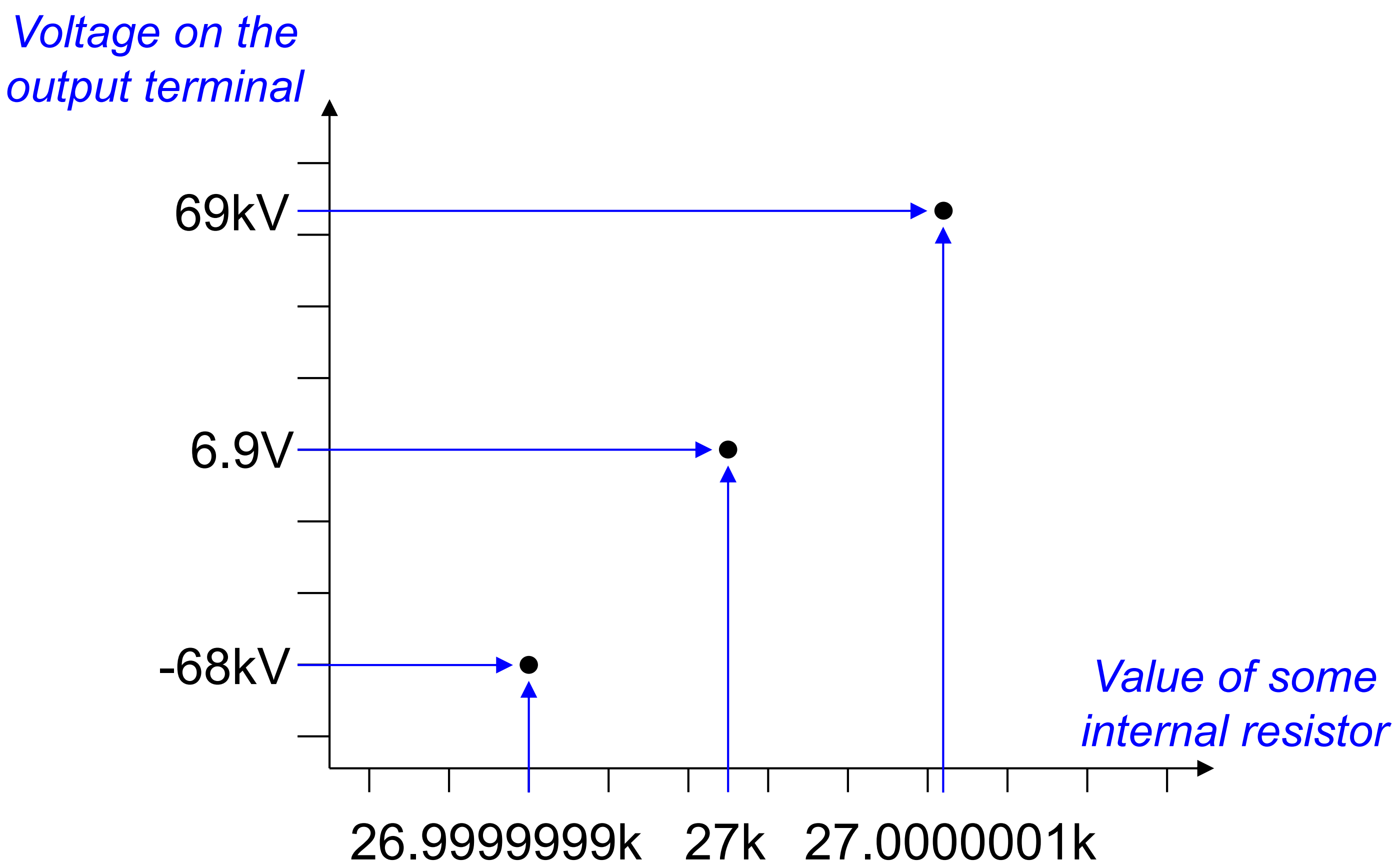


- May well be entirely true

Sensitivity

POETS

- What you never asked:



- May well kill you

Errors

POETS

- ***All*** simulators introduce noise and inaccuracy into their results

- OK, as long as this is
  - (Reasonably) small
  - (Reasonably) predictable
    - At least statistically
  - (Reasonably) uniform
  - (Broadly) understood by the user

But

- Any simulation...

    (Recall a simulation is an *experiment* run by a human using a *tool* that is not in itself intelligent)

...that does not provide quantitative indications of errors and sensitivities due to finite system size, word lengths, initial state, boundary conditions....

...**is at best useless, at worst downright dangerous**

Units and reality

- Simulators running on computers know nothing of
  - Reality
  - Units
- All quantities are dimensionless
- All equations are discrete even when they're not
- All numbers are integers even when they're not
- The same program on different machines *will* produce different answers

Units and reality

POETS

- A simulation of a mass falling in a uniform gravitational field will produce trans-light speeds very quickly
    - Unless *you** see it coming and make the model more complex

- A simulation of a 100V source driving a 1μohm resistor will happily predict a current of 10^8 amps
    - Unless *you** see it coming and make the model more complex

** The simulator isn't going to*

Simulation

POETS

- The **Art** of simulation: to predict reality
- How does nature do reality?

- *Unnatural* serialisation:
  - Everything gets shoved through the ALU bottleneck
  - Instruction ordering is deformed...
    - Out-of-order
    - Speculative execution
    - Code transformation
    - Multiple layers of cache
  - ...to the point of (but not past) breaking semantics
  - Nature doesn't do it this way

Outline

POETS

- The rise and asymptote....
- Simulation - there's a lot of it about
- Event-based simulation
- What is the user base?
- Where does all the time go?
- Down amongst the Hard Sums
- A brief meander into reliability
- Some pictures of hardware

The Big Picture

POETS

- POETS
  - Partial Ordered Event Triggered System

is a ***simulation acceleration*** technology

For <u>certain classes</u> of problem
- that can be broken down into large, discrete meshes/graphs,
with simple interconnect topology....

.....POETS can deliver orders of magnitude wallclock simulation speedup

$O(n^{?}) \rightarrow O(1)$

Fabrication technology...

...has bought about massive changes:

- Hardware increasingly statically and dynamically unreliable
- Cores are free
- Communication costs ~ *1000 x* compute costs

Software design...

...needs to change in response to this plethora of cores:

- Software must cope with
  - Unknown and changing physical core topologies
  - Non-deterministic message passing
  - Localised data
- Dramatic performance improvements
  - For certain classes of problems
- Underlying *mathematics* still valid
  - Algorithmic embodiment different

Simulation: The old way

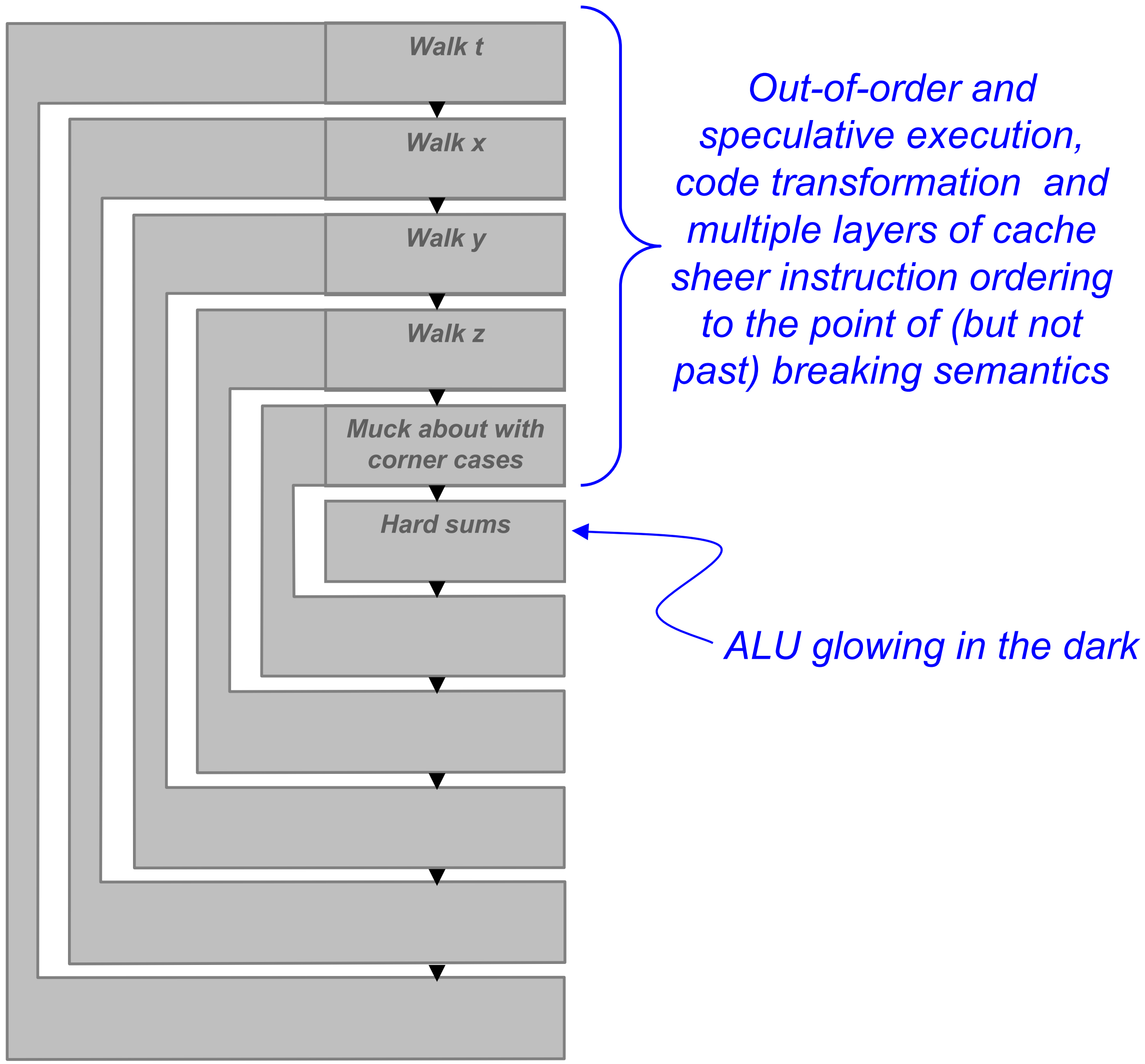
POETS

- Build a model:
  - Probably some sort of graph
  - Local interactions
- Process the model:
  - Linearise and shove it through some (small) set of processors
- Big problems:
  - A lot of fetching
  - A lot of shoving
  - A lot of writing back

Where does all the time go?

POETS

- *Unnatural* serialisation:
  - We're simulating (usually) nature
  - Nature doesn't do it this way



The POETS way

POETS

- Build a model:
  - Probably some sort of graph
  - Local interactions
- Process the model:
  - Distribute processors over the model
- Big problems:
  - No fetching
    - Wherever there's data, there's a processor
  - Little shoving
    - Things interact with their neighbours in parallel
  - No writing back
    - Data is stored locally to the generation site

Causality is not synchronisation

Interaction via small, fast messages

Imagine a system…

POETS

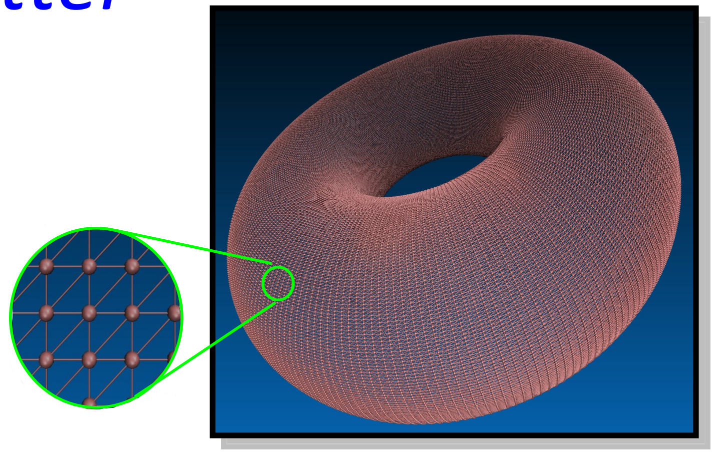
- Arbitrarily large number of cores
  - Cores are *free*
- Core-core communications *cheap*
  - We designed it that way

- *What* might we do with it?
- *How* might we do *anything* with it?

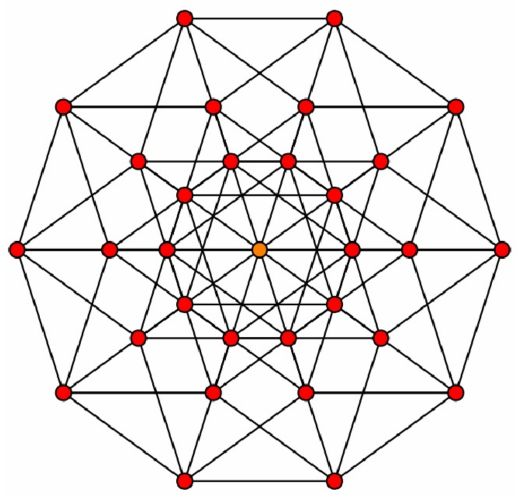
What might it look like?

POETS

- *It doesn't really matter*



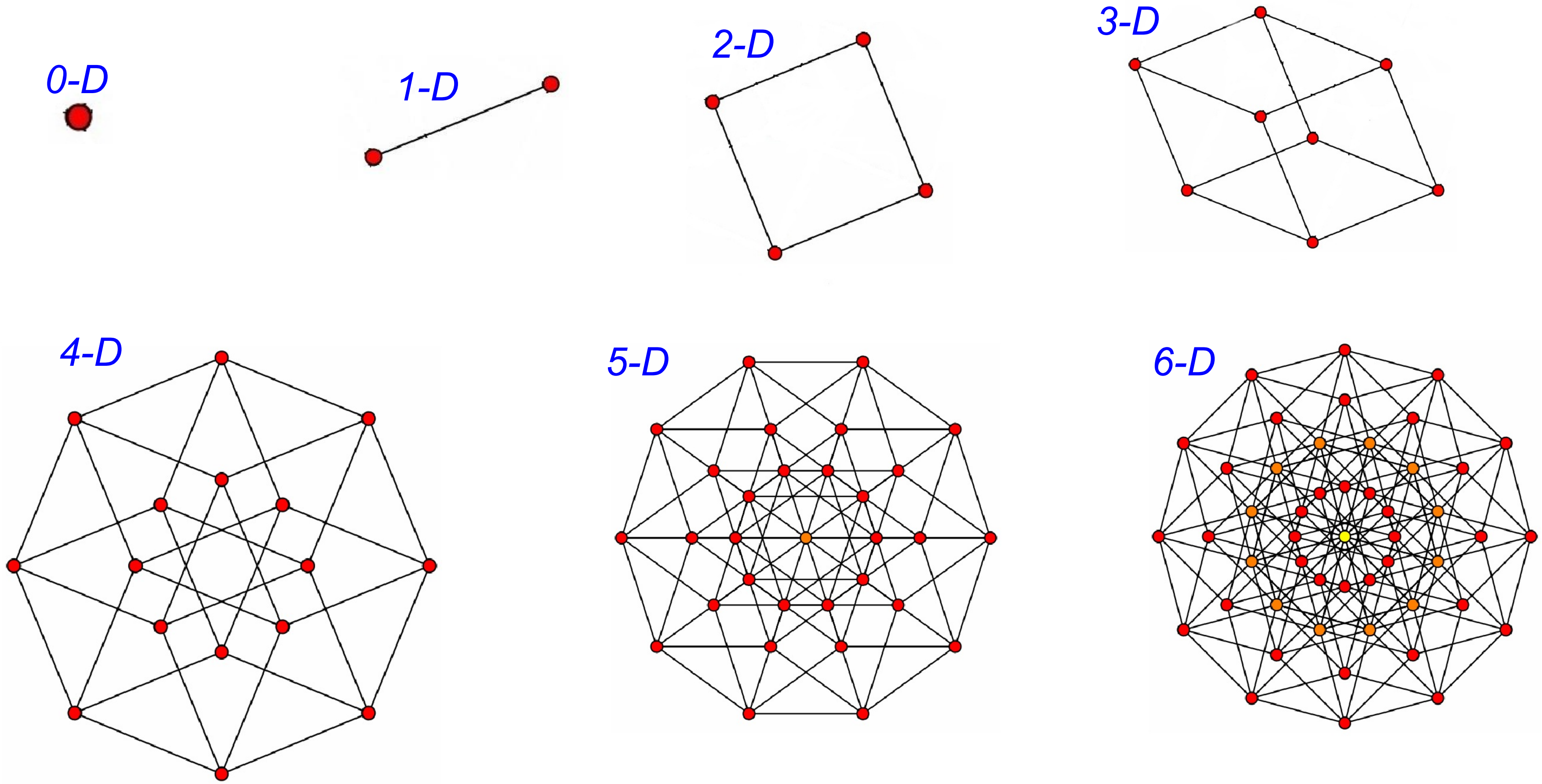
- SpiNNaker :
  - Nominal torus
- POETS :
  - Nominal 5D hypercube



Hypercubes….?

POETS

- Unit cubes, tiling space :



Why doesn't it matter?

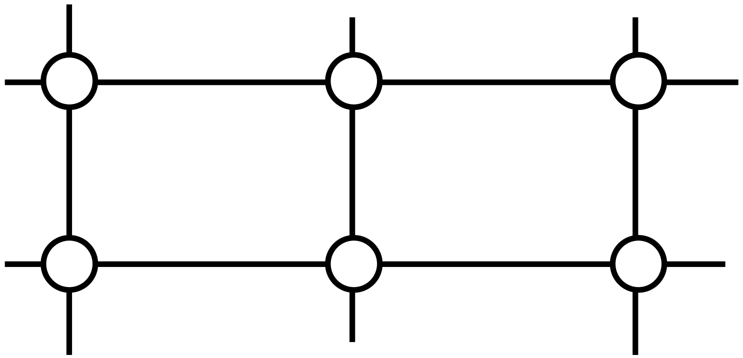
POETS

- Each node in the previous pictures =
  - Some small cluster of compute
  - Router
- Routers possess 'straight through' capabilities
- Can make the *physical* geometry of the hardware represent any *arbitrary* virtual topology

The story so far...

POETS

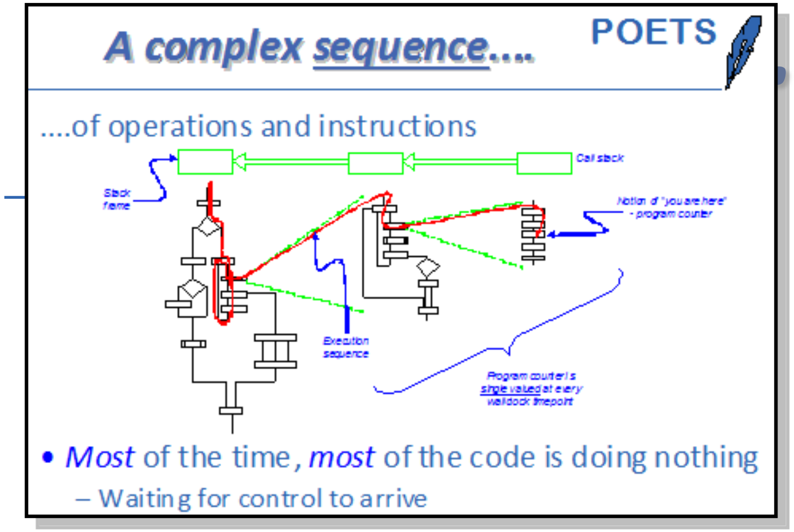
- Huge (O(10^6)) number of small processors
- Embedded in some regular (because it's easy), asynchronous hardware communication infrastructure
  - Massively parallel



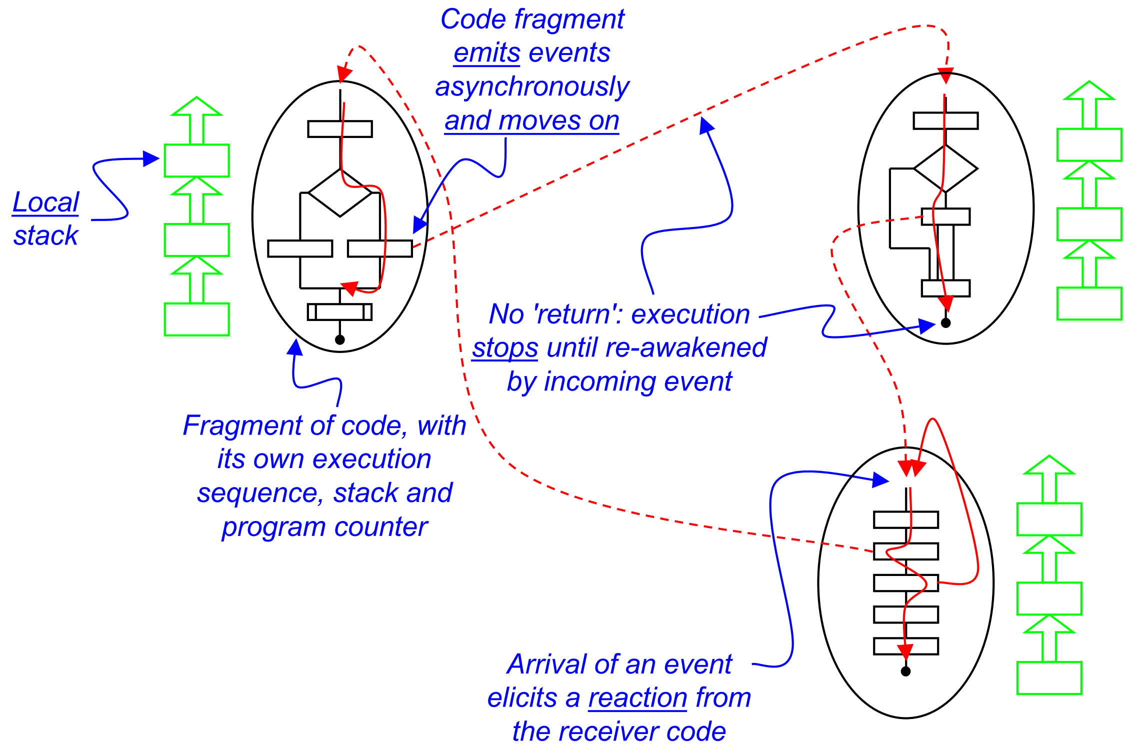
How does this work?

POETS

- The problem must be formulated as a graph of *vertices* interacting via *edges*
  - No restriction on connectivity
  - Graph may represent geometry or topology
    - Or both
- Vertices ***react*** to incoming messages
  - May send out messages of their own
  - No overall choreography of packet traffic
- Problem graph is mapped to physical core graph



...becomes a complex set of sequences...



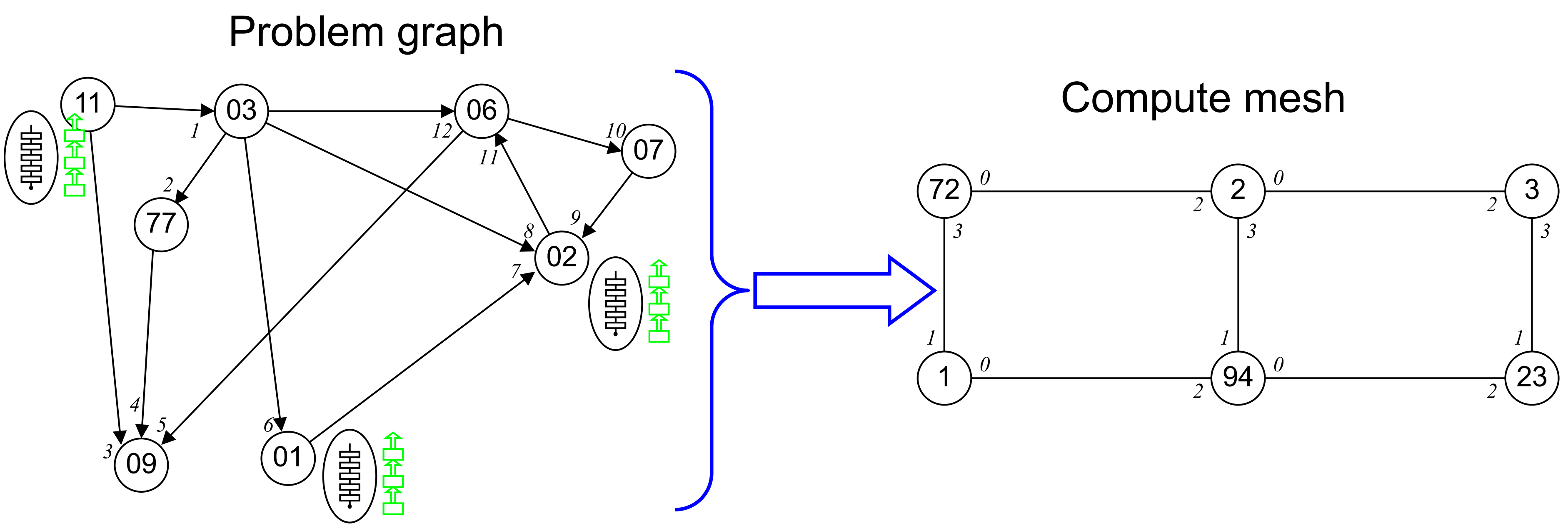
....of operations and instructions

- Activity in the simulation mirrors reality
  - If the code is idle, it's because *reality* is idle

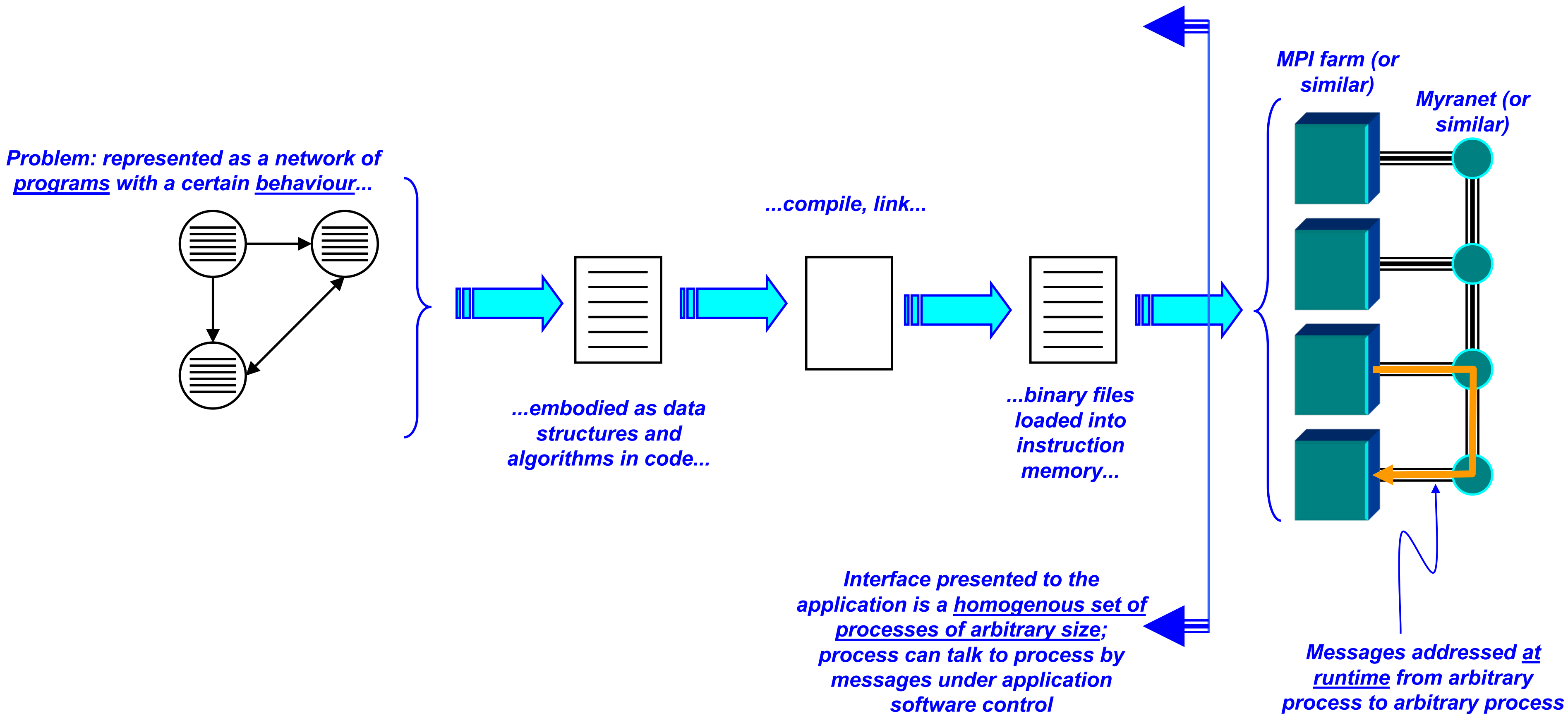
Mapping one to the other

POETS

- Anything that can be naturally represented as a discrete graph:

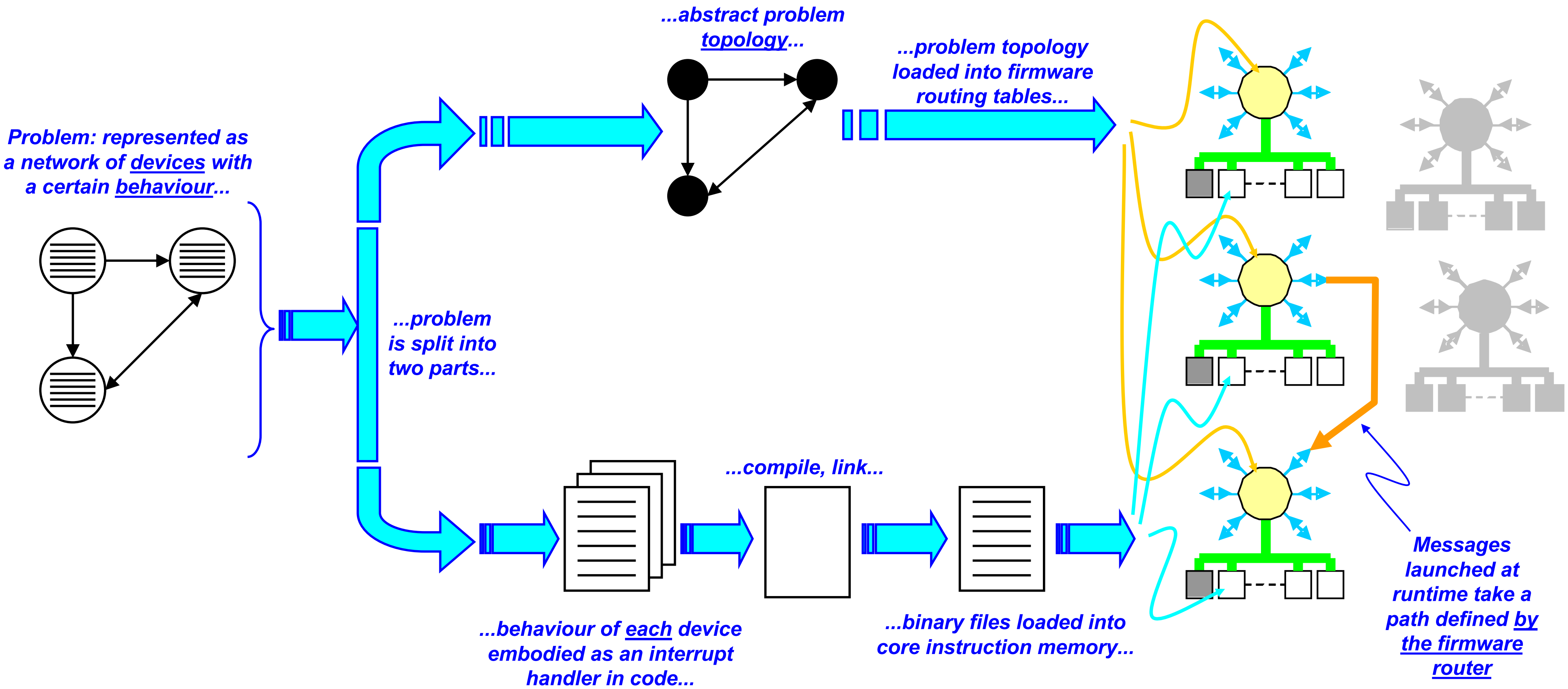


A conventional multi-processor program:



Event-based problem flow

POETS



The code says "send message" but has no control where the output message goes - the route tables in each node decide

Outline

POETS

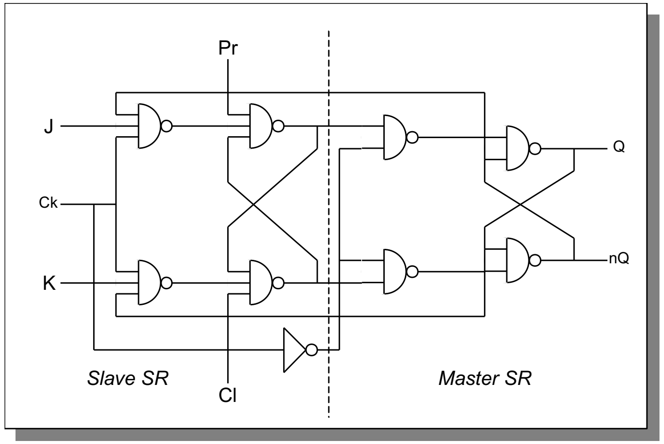
- The rise and asymptote....
- Simulation - there's a lot of it about
- Event-based simulation
- What is the user base?
- Where does all the time go?
- Down amongst the Hard Sums
- A brief meander into reliability
- Some pictures of hardware

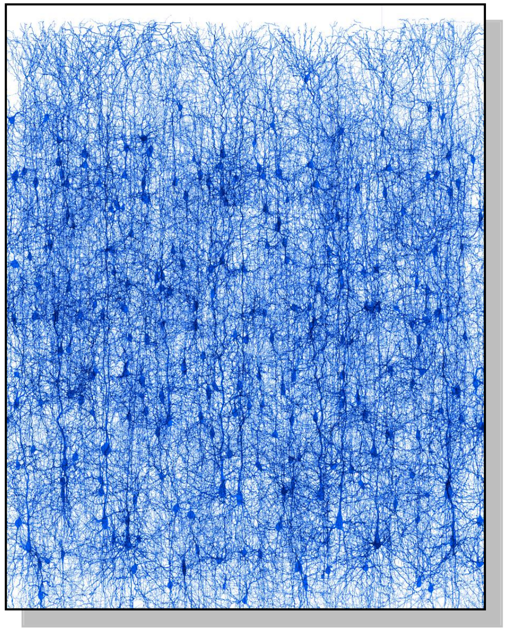
What sort of problem?

POETS

- Anything that can be naturally represented as a discrete graph:

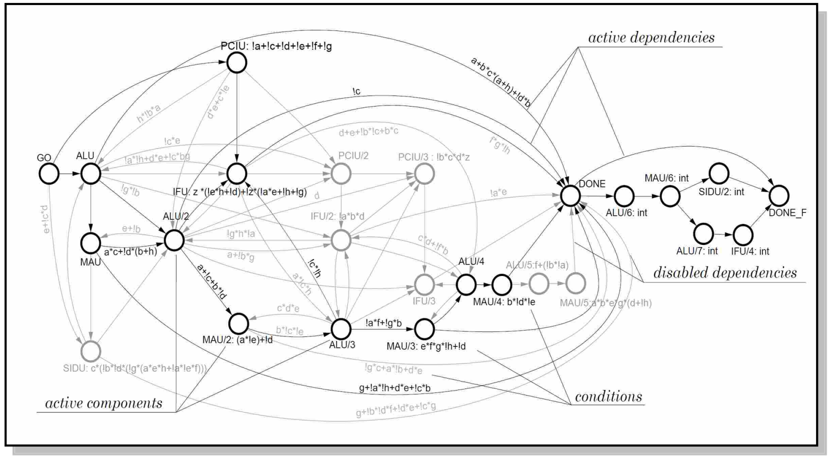
Electronics





Neural circuitry

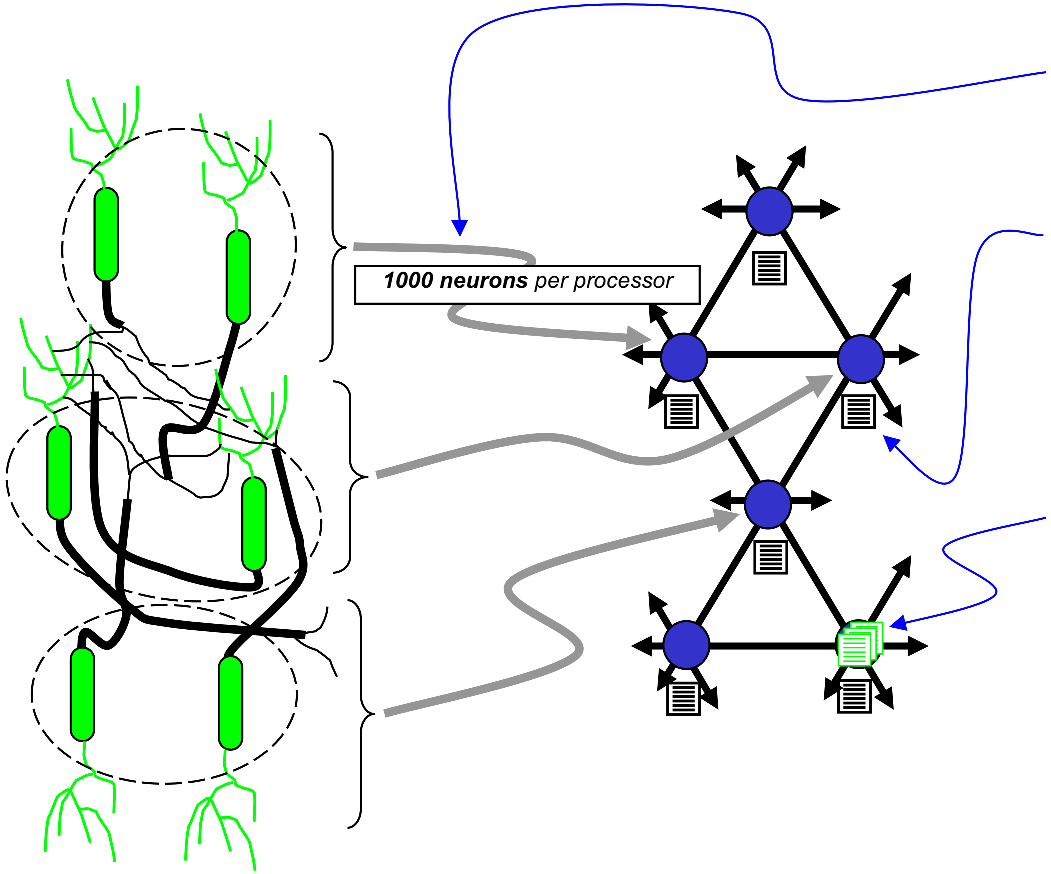
Discrete systems



Mapping

- Map problem nodes to compute nodes:

Offline configuration software maps neurons:cores (~1000:1)



Maps each individual *neuron* to a SpiNNaker *core*

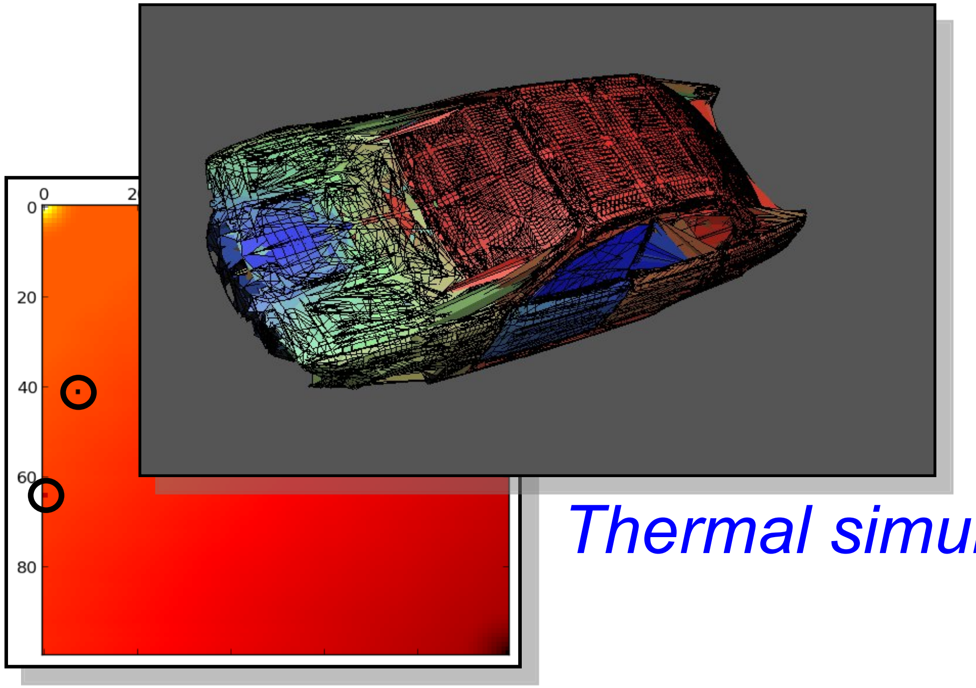
Defines the router tables for each *node*
Connectivity of neural topology is *distributed* throughout the system in the routing tables

Defines the index structures necessary in each core to allow fast retrieval of neuron and synapse state
Defines the packet handling *code* *(interrupt handlers)*

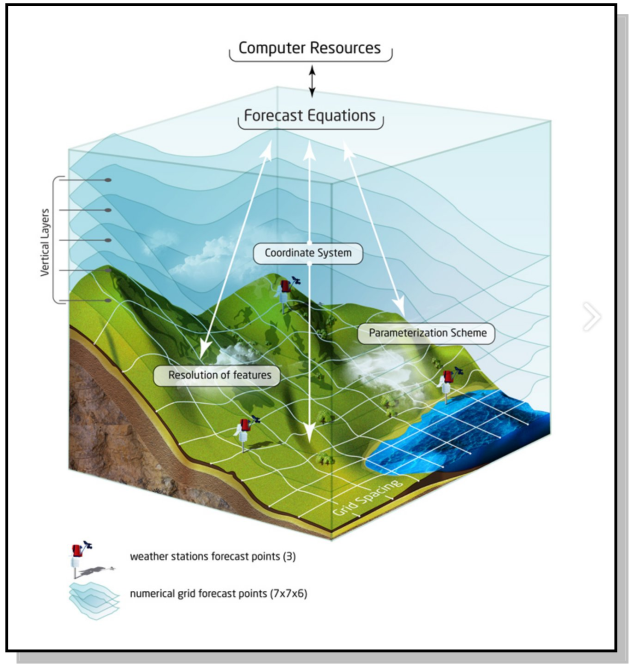
What sort of problem?

POETS

- (Some) physical systems can be naturally mapped to a continuous (not necessarily regular) tiling of 3-space:

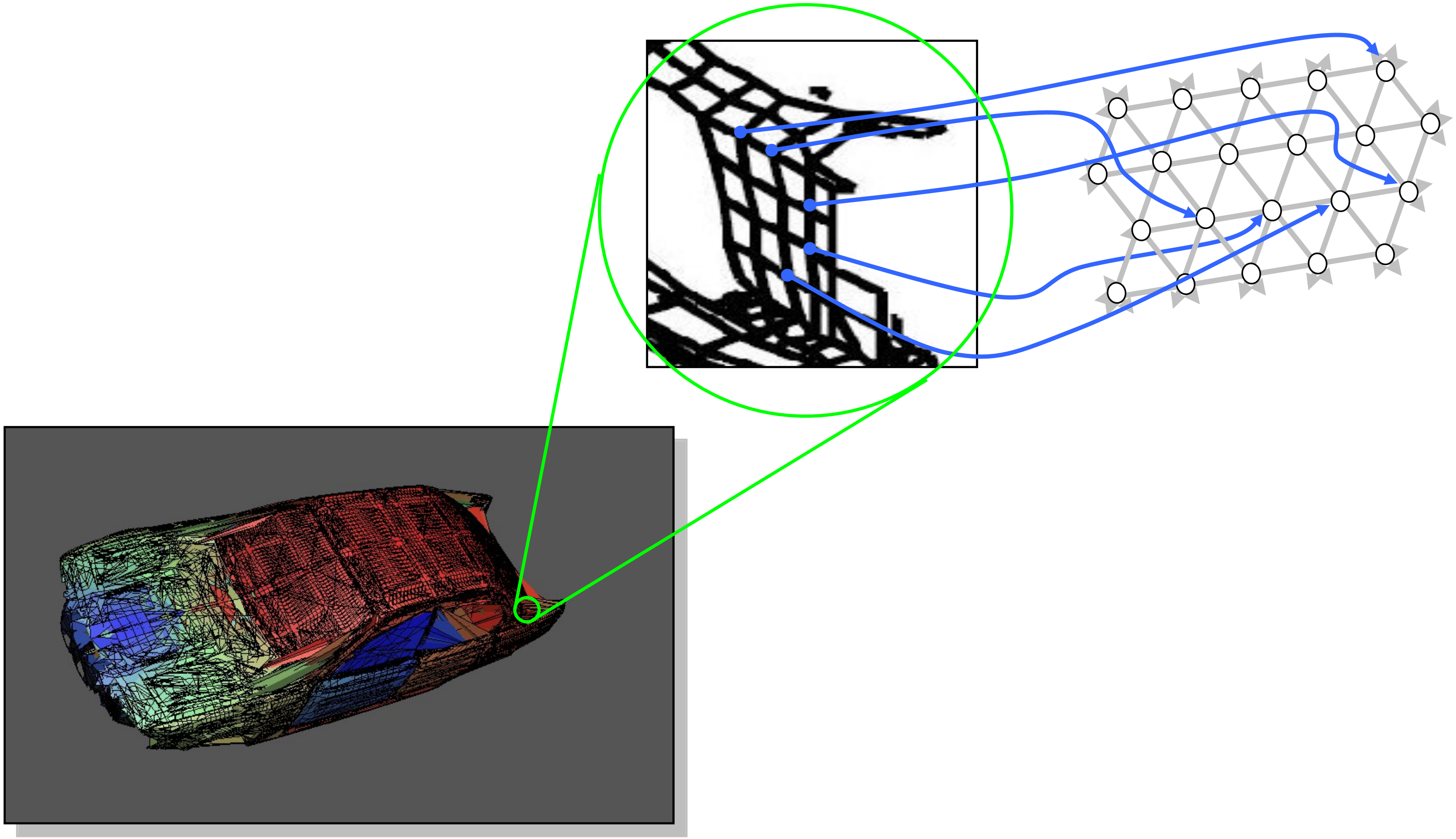


Thermal simulation



Weather modelling

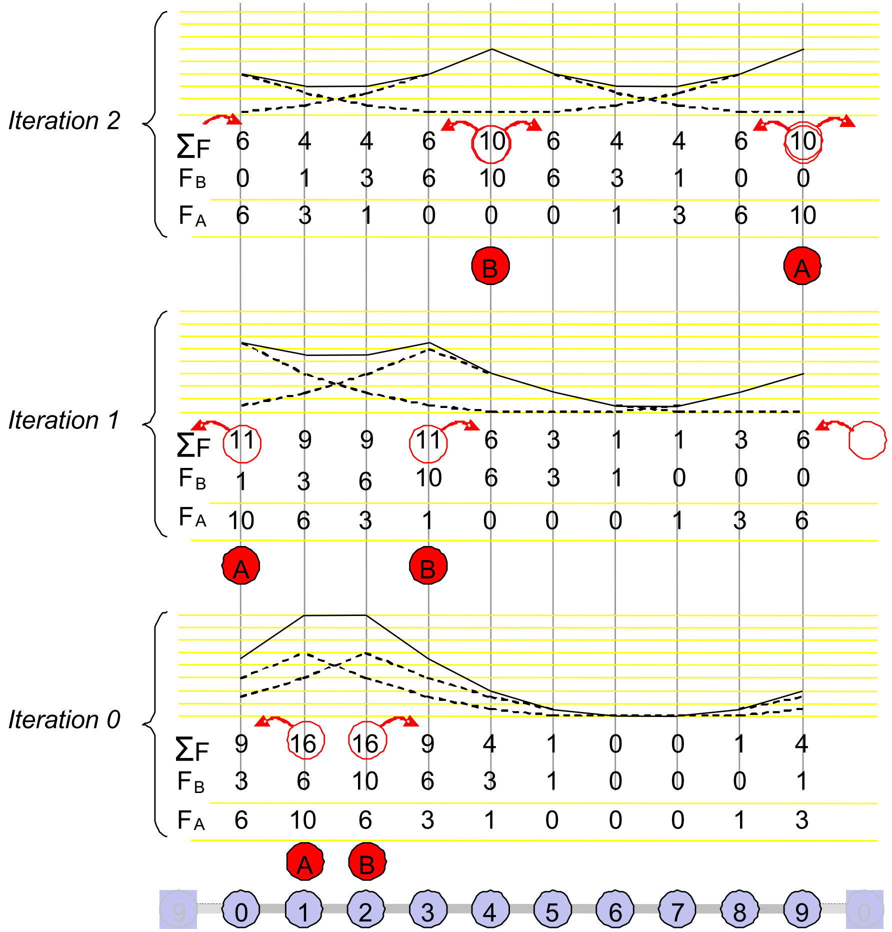
Mapping



What sort of problem?

- Continuous field problems:

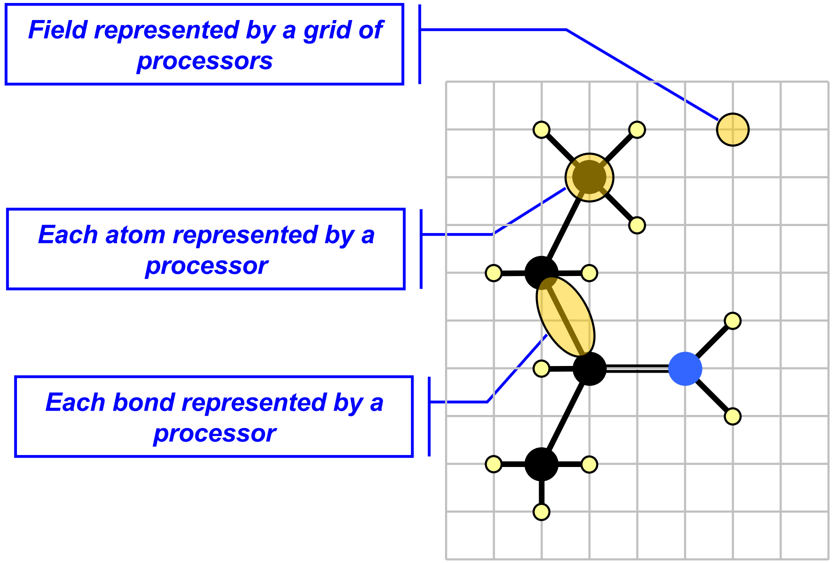
- Each device knows its physical position
  - Broadcast to the Universe via flood fill
- Each device knows the functional form of (field, distance)
  - Can derive the contribution of every other device to the local field at its location
  - Hence the local force
  - Hence *dr*
- 'Knowledge horizon' of each node is *immediate neighbours only*
- Notion of field *slope* can be a bit tricky in discrete systems
  - Asymptotic solutions can oscillate
  - Detection of this is non-trivial



What sort of problem?

POETS

- Hybrids:



A processor for each

Point in space

- Electric fields, Van der Waals forces

Atom

Bond

- Chaperone molecules

> *Particles tell space how to curve*
> *The curvature of space tells particles how to move*

  – (We thought) too big even for a 10^6 core machine
  – But.... the *computational chemists* say no, it's good for Monte Carlo, molecular dynamics and biological membranes

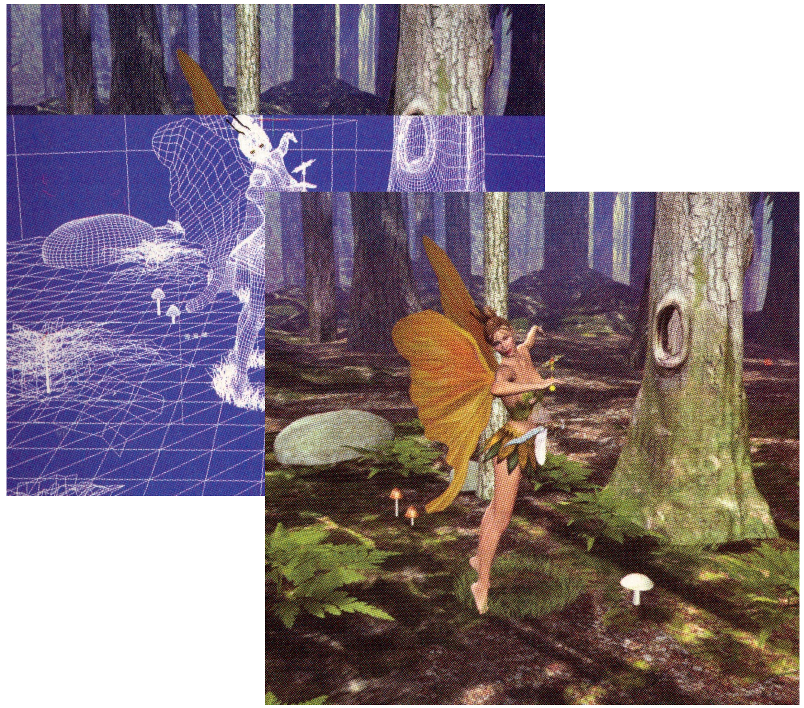
What sort of problem?

POETS

- Embarrassingly parallel:

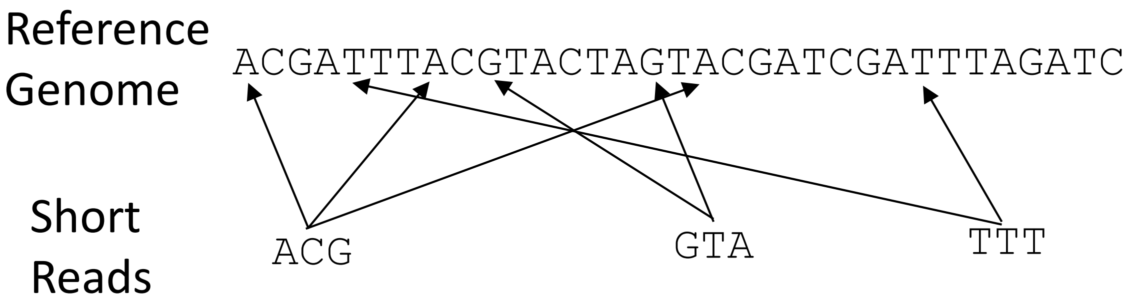
Ray-tracing:

One processor *per view plane* pixel



Genome searching:

Data structures *massive*



How big is "massive"?

Human genome ~ 750 Mbytes

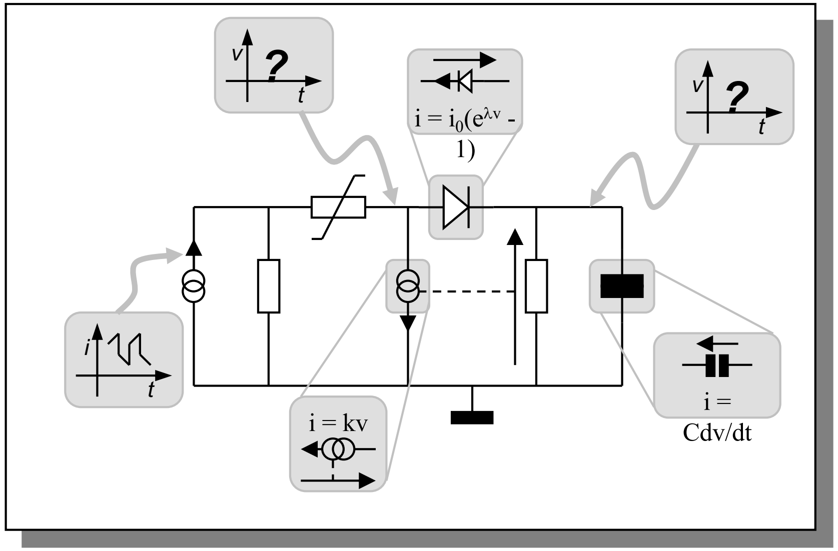
Substring matching problem

Not cryptographically massive: $O(10^{100})$

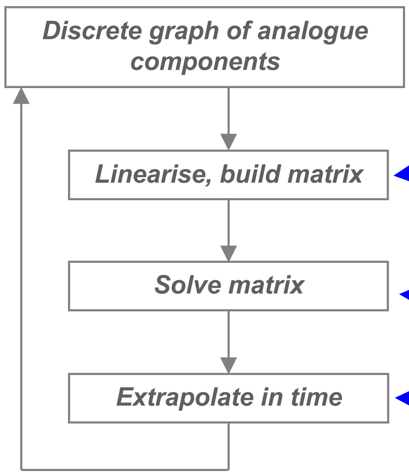
What sort of problem?

POETS

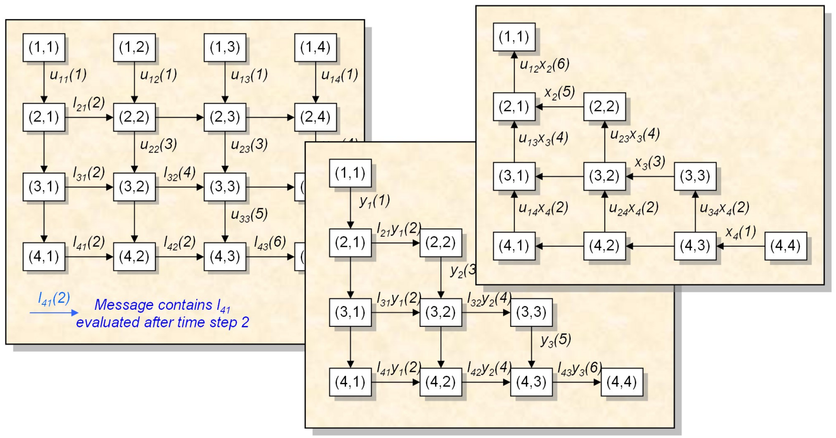
- Indirect problems:
- Cannot use POETS to directly attack the physics
  - Use it instead in the intermediate mathematics



Embarrassingly parallel:
1 core / problem node



Matrix manipulation:
1 core / element

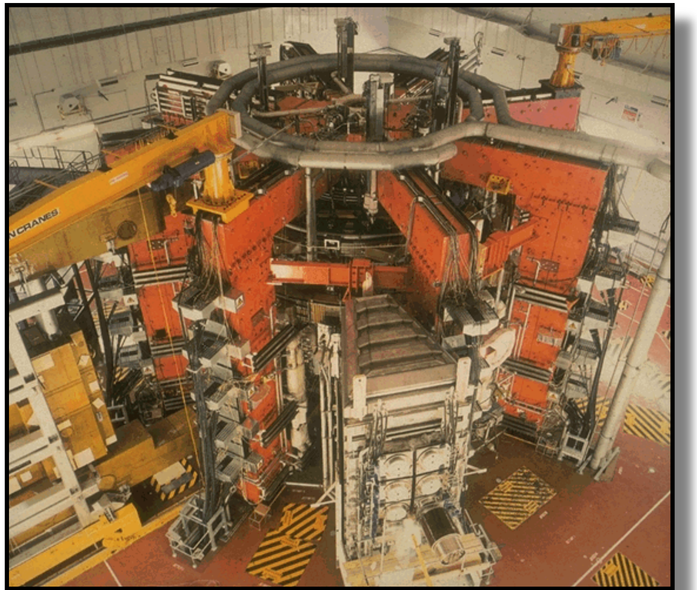


Embarrassingly parallel:
1 core / problem node

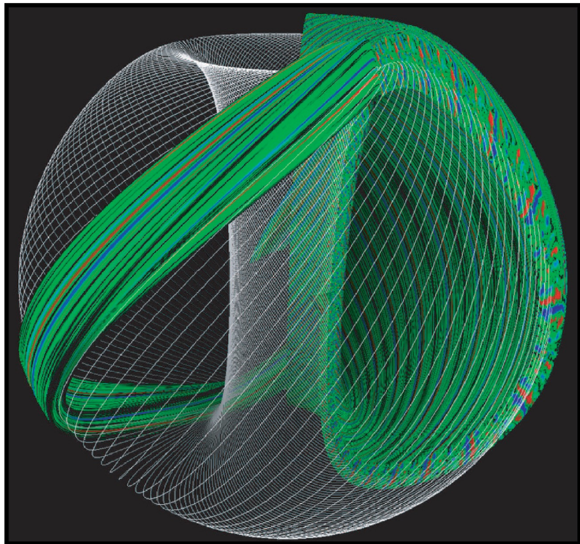
What sort of problem?

POETS

- Gyrokinetic plasma:



  - (We thought) too big even for a 10^6 core machine



  - But.... the *gyrokinetic plasma physicists* say no, it might be useful for plasma turbulence

Rob Akers, Culham Science Centre

What's the downside?

POETS

- POETS is not a general-purpose machine
  - Underlying mathematics of a simulation problem must be re-cast to POETS-speak
  - Algorithmic embodiment is different
    - Programmer has no control over or knowledge of the solution trajectory

- But we are continually amazed at what can be coerced into a POETS shape

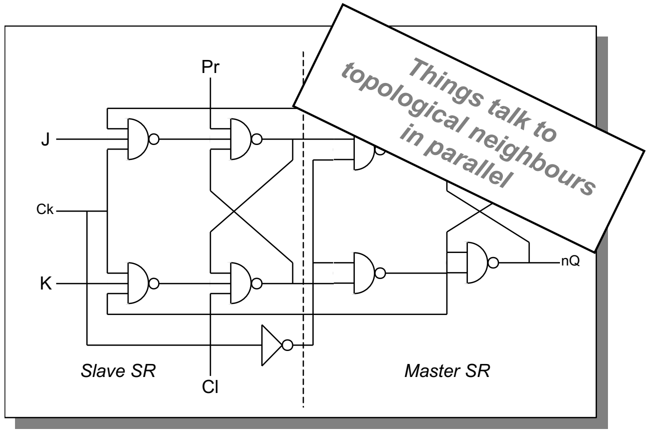
Outline

POETS

- The rise and asymptote....
- Simulation - there's a lot of it about
- Event-based simulation
- What is the user base?
- Where does all the time go?
- Down amongst the Hard Sums
- A brief meander into reliability
- Some pictures of hardware

Natural traffic patterns

POETS

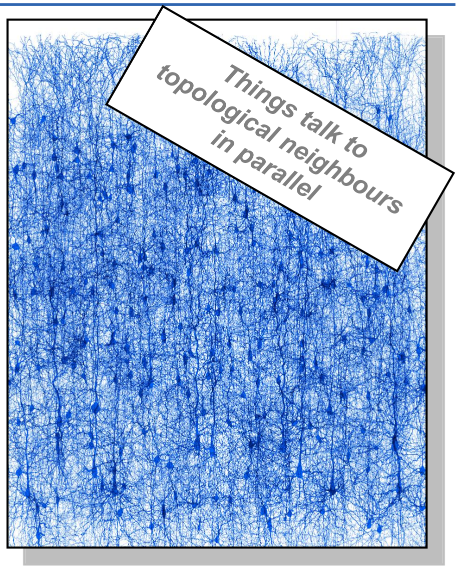


{value, net, time}

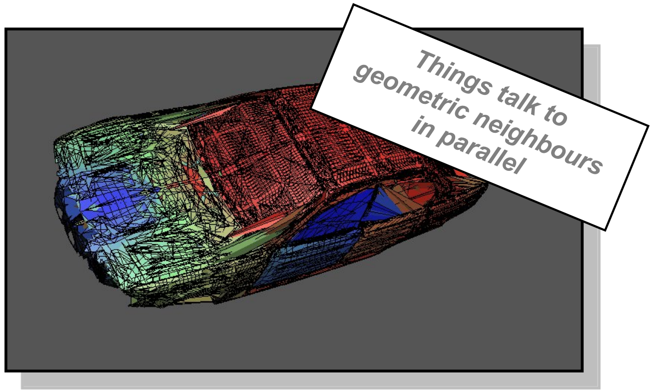
Stuff happens in parallel

Final solution independent of mathematical trajectory

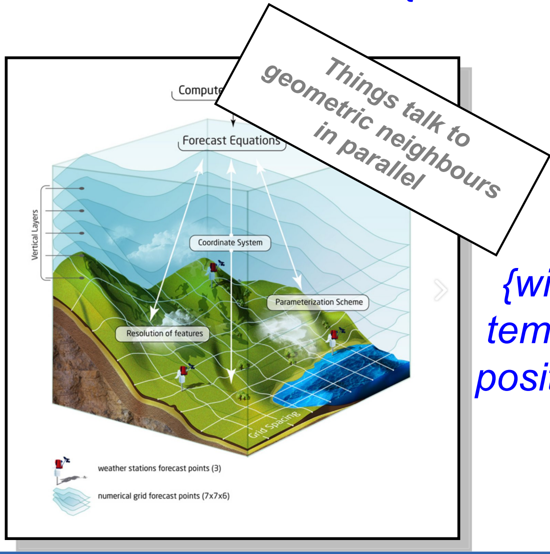
Causality is not the same as synchronisation



{neuron, time}



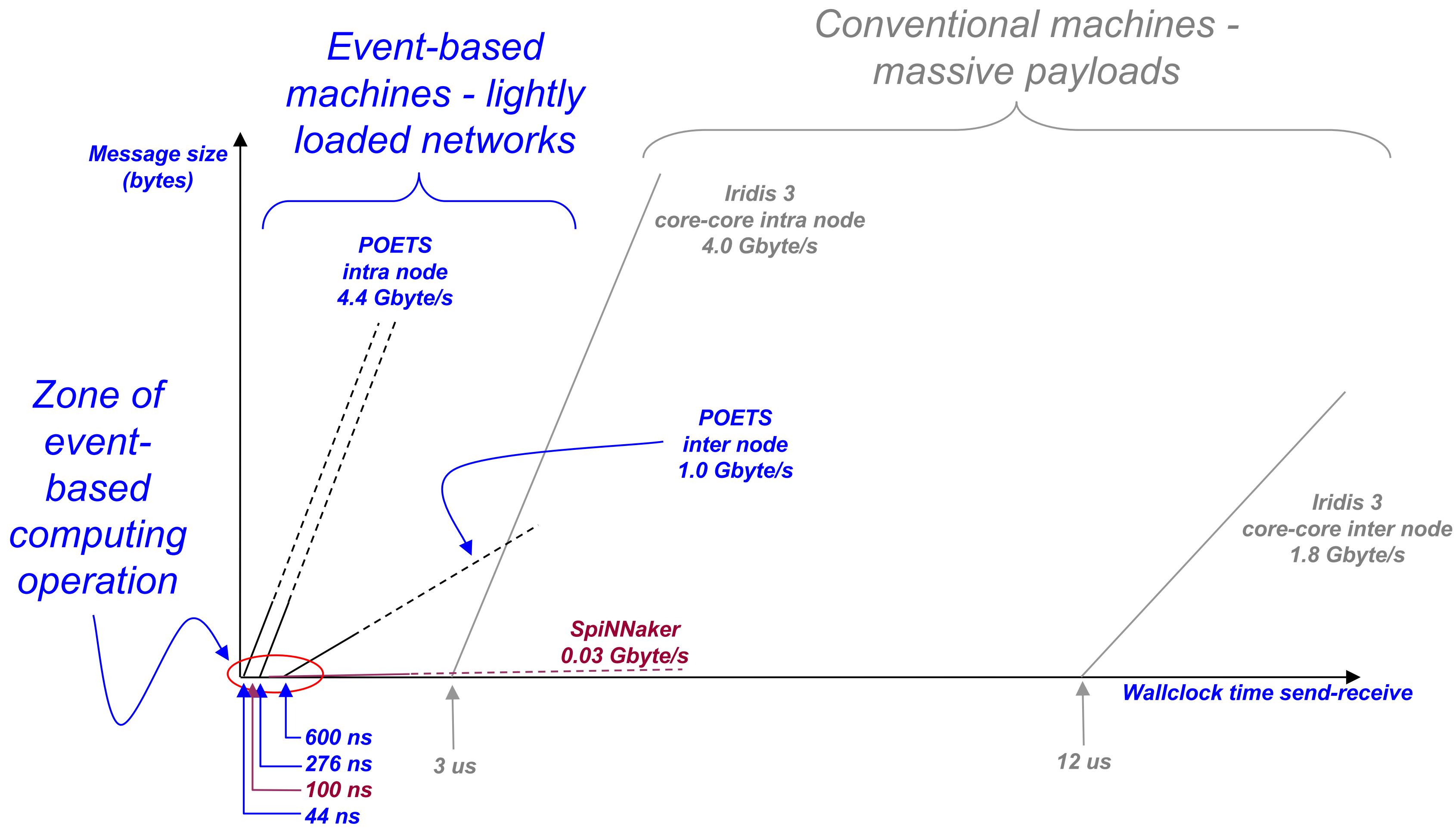
{temperature, position, time}



{wind, rain, temperature, position, time}

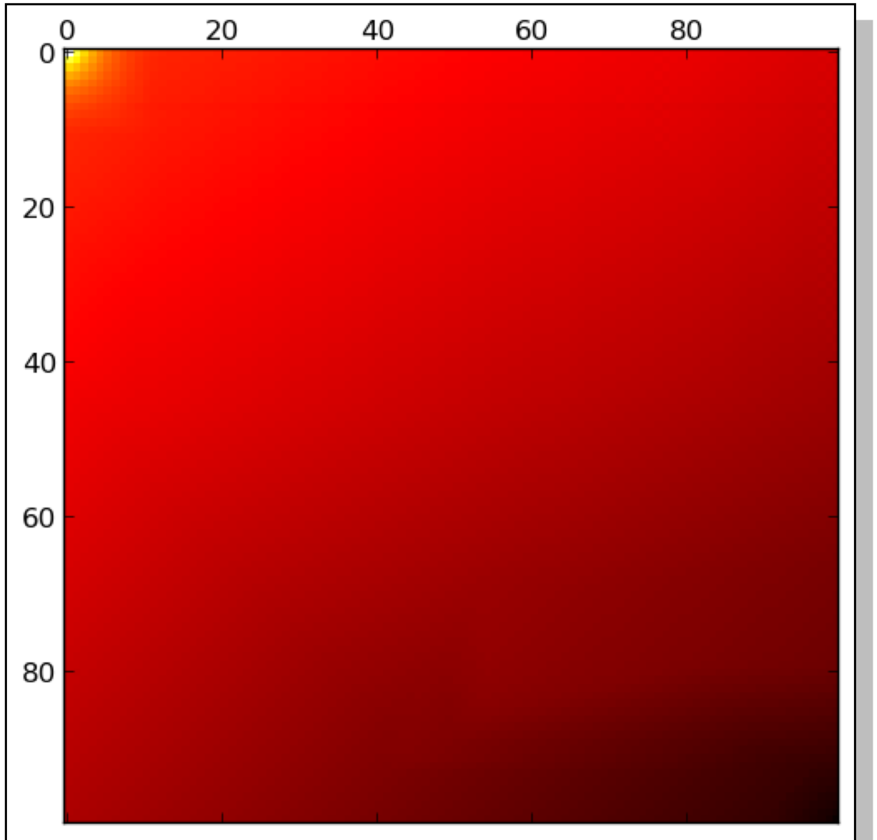
Parallelise our software...

POETS

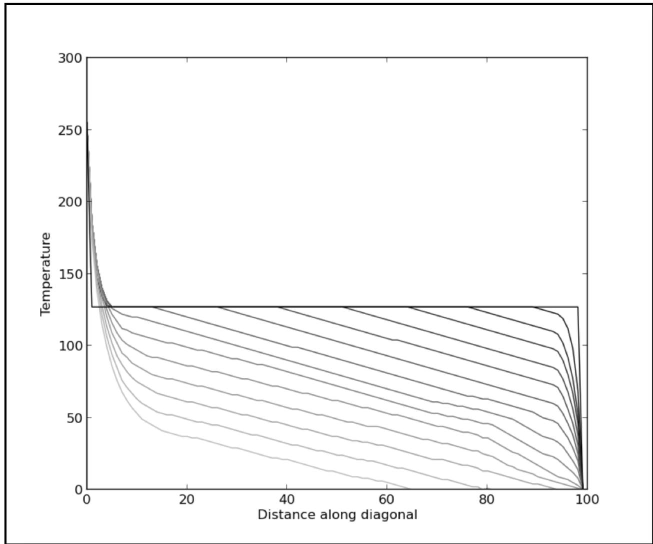


Using event-based simulation

POETS



Finite difference heat diffusion canonical 2D square grid:



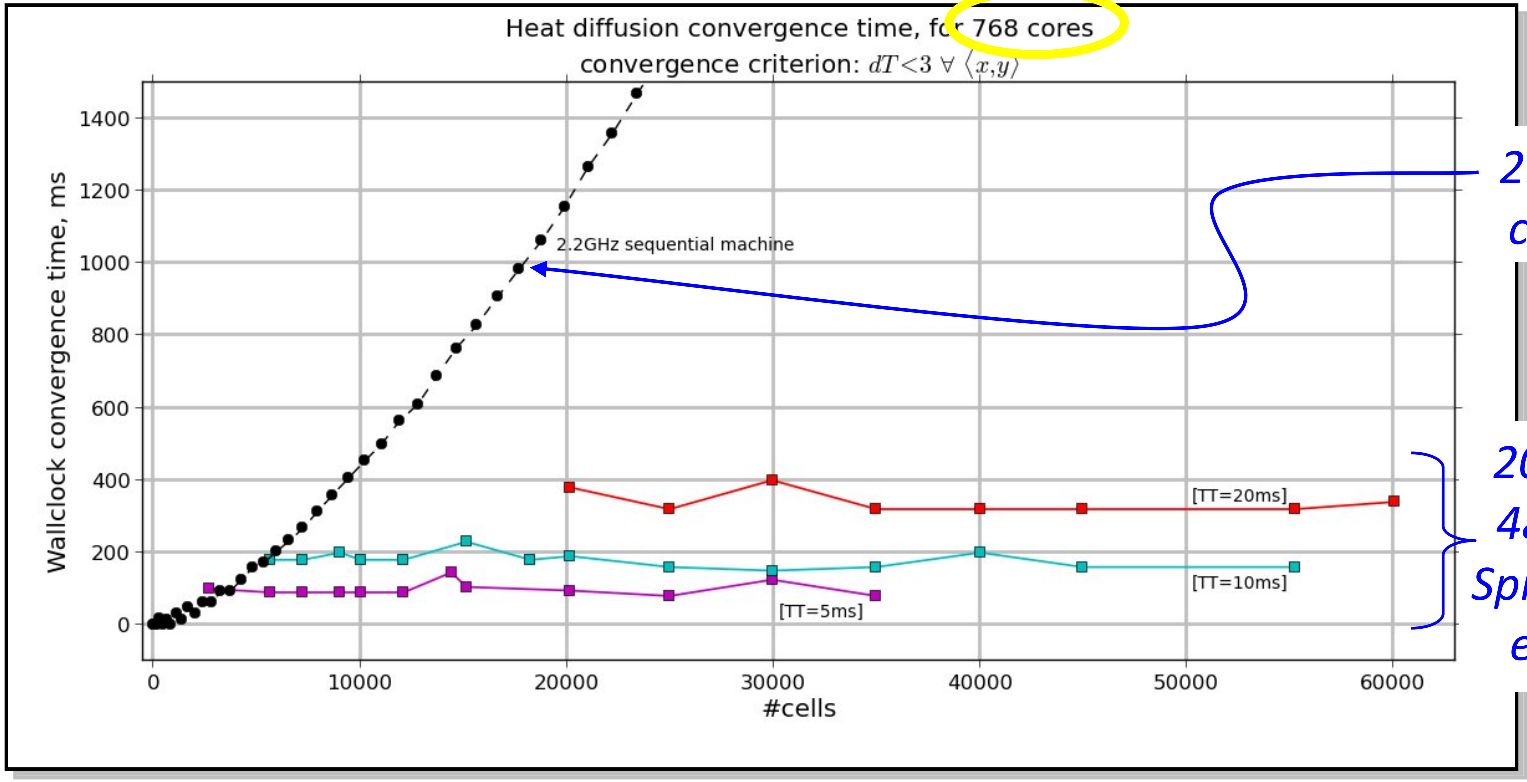
Diagonal temperature profile vs iteration

The reward...

POETS

- Dramatic performance improvements
  - For certain classes of problems

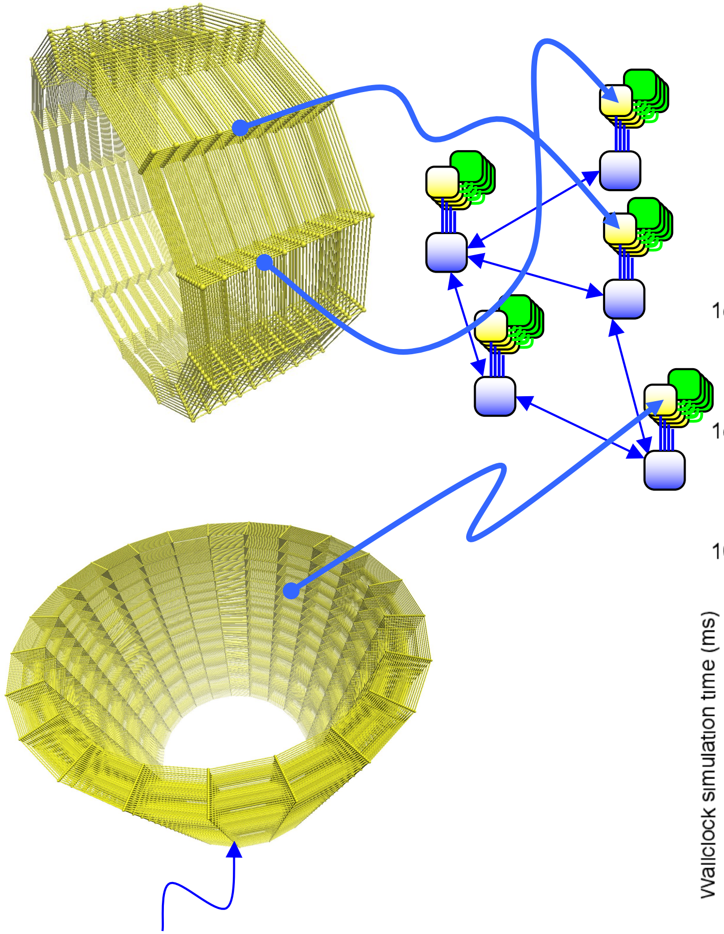
$O(n^?) \longrightarrow O(1)$



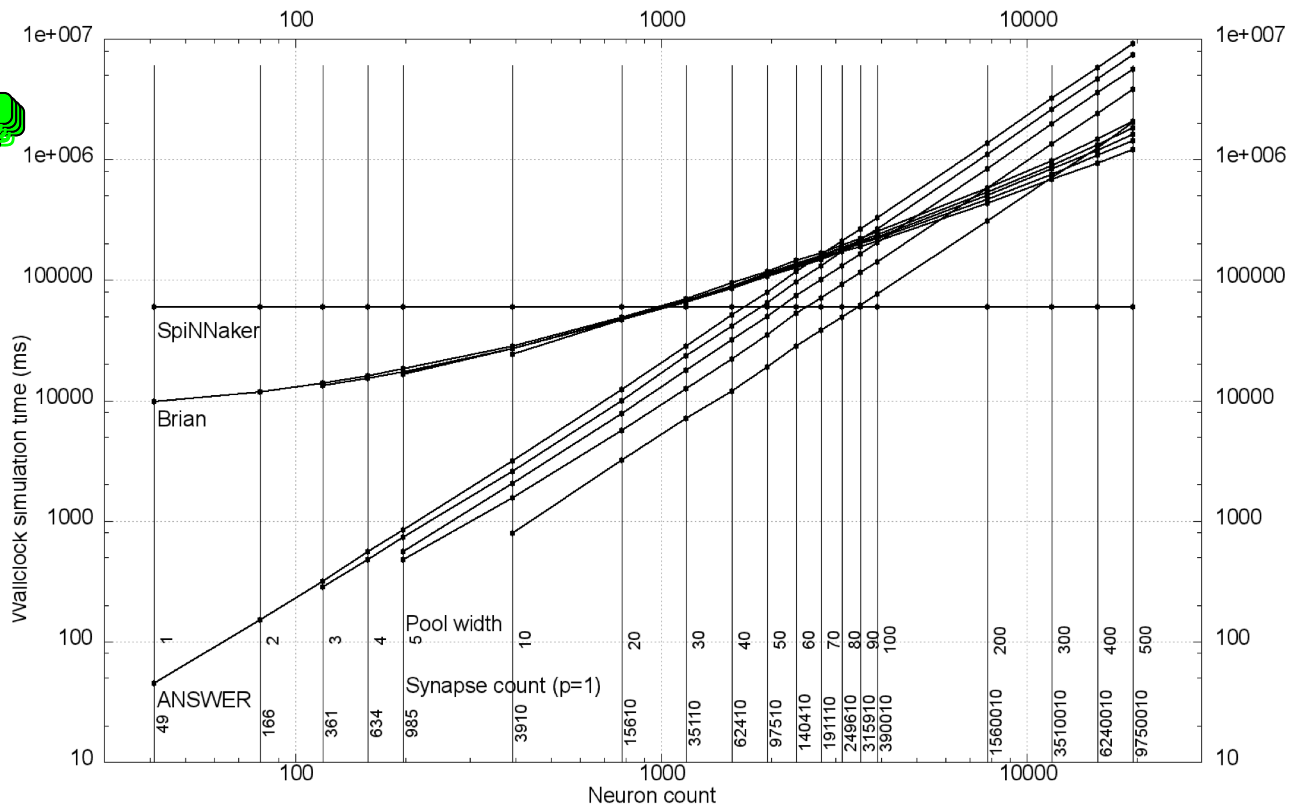
2.2GHz single core desktop

200MHz 48 node SpiNNaker engine

Synfire rings



Artificial "biological" neural structures



Outline

POETS

- The rise and asymptote....
- Simulation - there's a lot of it about
- Event-based simulation
- What is the user base?
- Where does all the time go?
- Down amongst the Hard Sums
- A brief meander into reliability
- Some pictures of hardware

Imagine a system…

POETS

- Arbitrarily large number of cores
  - Cores are *free*
- Core-core communications *cheap*
  - We designed it that way

- *What* might we do with it?
- *How* might we do *anything* with it?

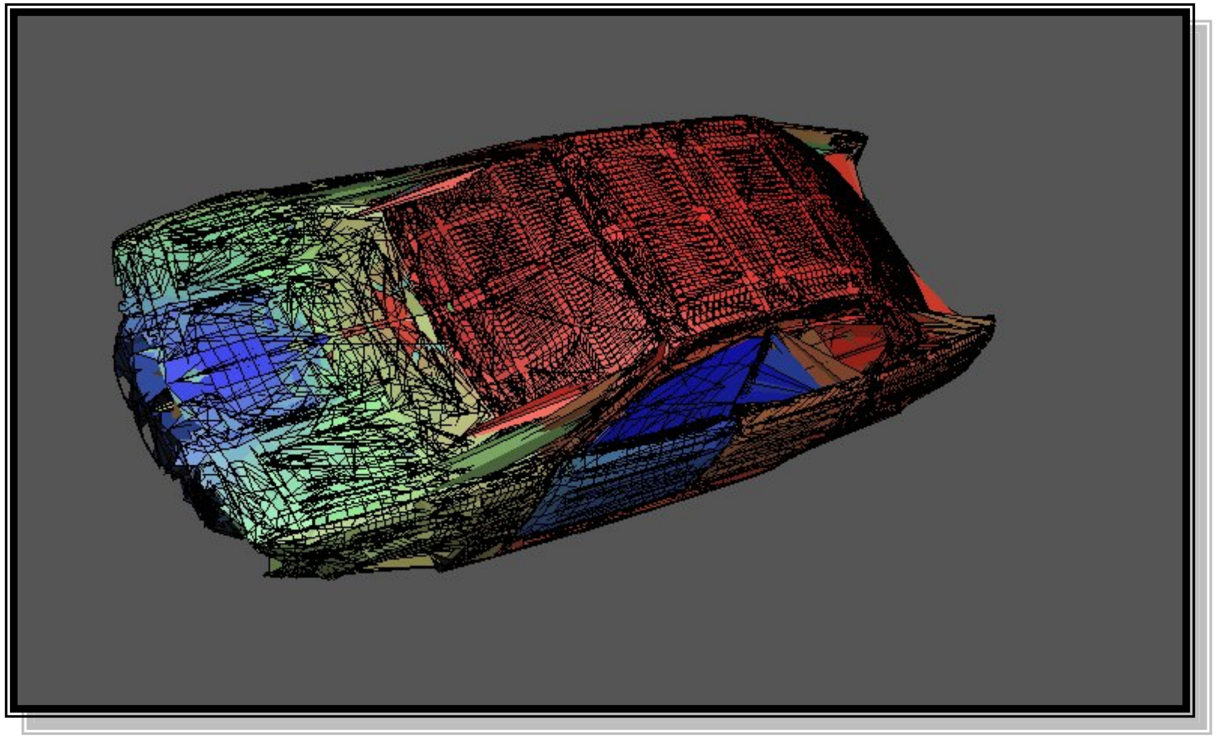
A direct real problem....

POETS

- Heat equation

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2}$$

  - becomes



n^{th} timestep

$$u_i^{(n+1)} = u_i^{(n)} + D\frac{\partial t}{\partial x_i^2}\left(u_{i-1}^{(n)} - 2u_i^{(n)} + u_{i+1}^{(n)}\right)$$

dx_i are different because the mesh is irregular

simulated time t = ndt

Numerical methods

POETS

Gauss-Seidl:

New value being calculated — *New values used as soon as they are available* — *Previous value - no update yet available*

$$\underline{x}^{(m+1)} = \underline{b} - \underline{\mathbf{L}}\underline{x}^{(m+1)} - \underline{\mathbf{U}}\underline{x}^{(m)}$$

Convergence fast: *newest* values used as soon as they become available

Predicate tree thin - not parallelisable

Jacobi:

New value being calculated — *Only old values used*

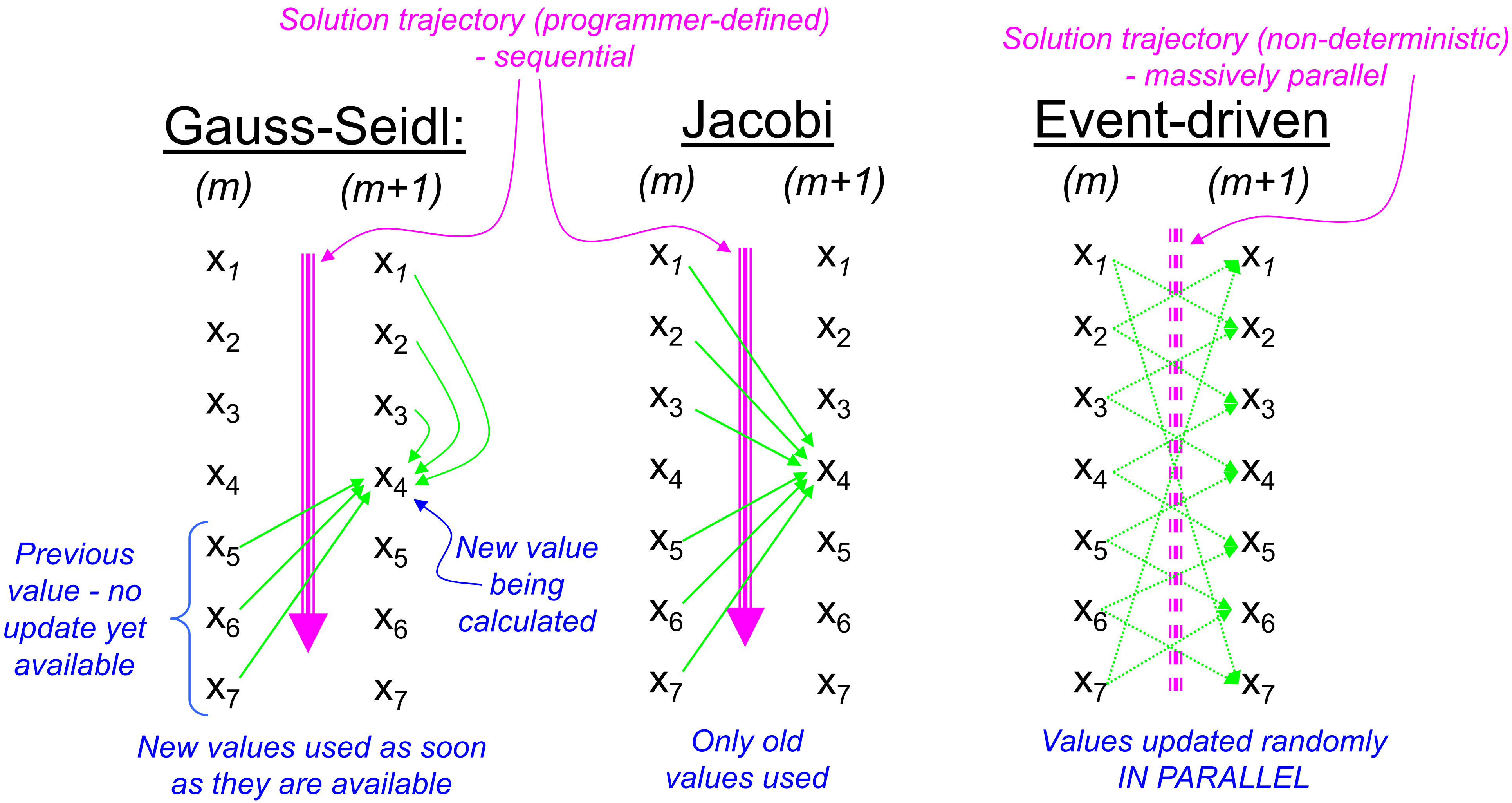
$$\underline{x}^{(m+1)} = \underline{b} + (\underline{\mathbf{I}} - \underline{\mathbf{A}})\underline{x}^{(m)}$$

Convergence slow: *entire* old value set used to generate *entire* new value set

Predicate tree wide - easily parallelisable

Event-driven method

POETS



An indirect real problem

POETS

- At their core, many problems resolve to

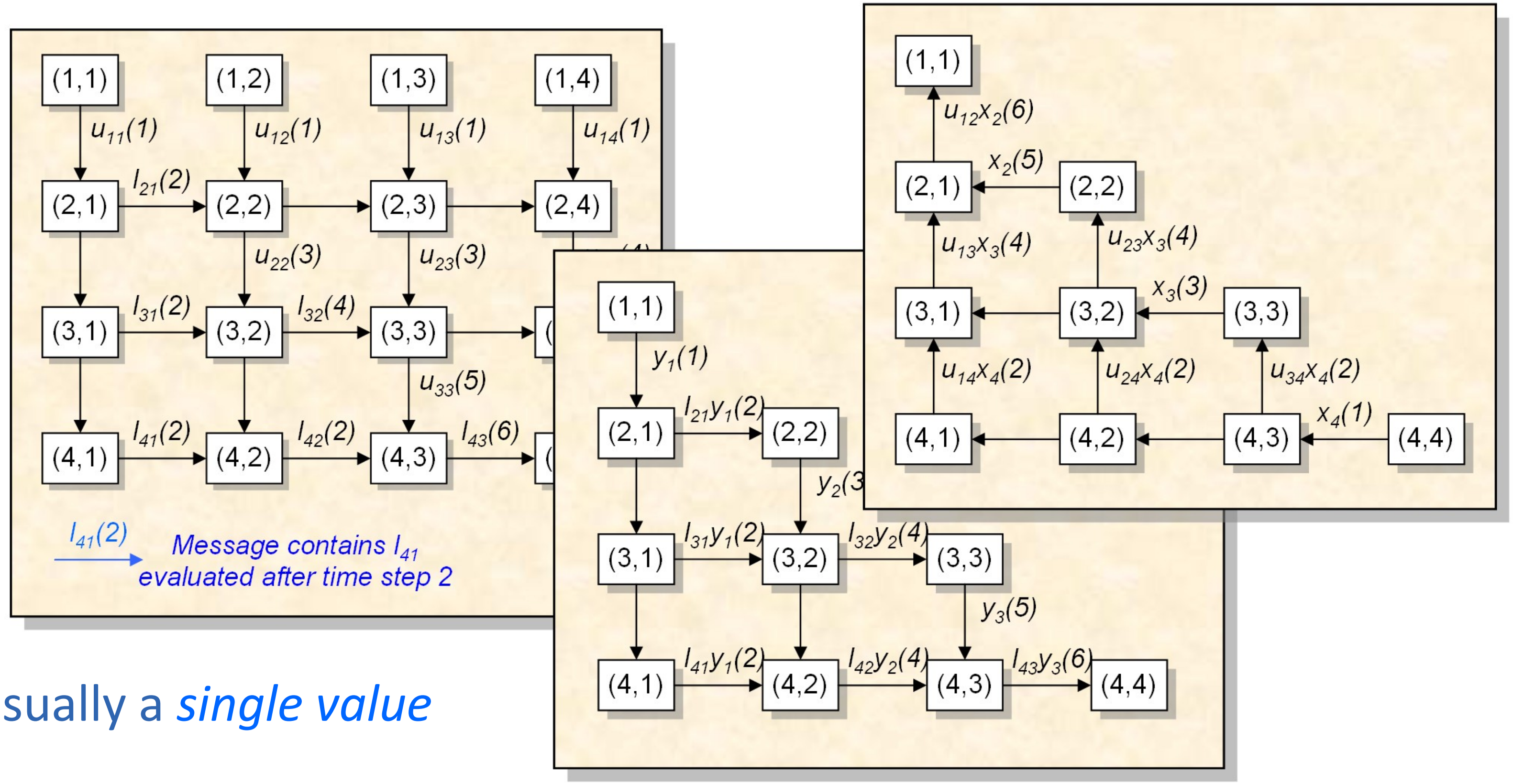
$$Ax = B$$

  - or similar
- Often sparse and ill-conditioned
- Numerically intensive
  - $O(n^3)$ for dense matrices
  - $O(n^{1.?})$ for sparse
  - Much ongoing research

$n \sim 10^{12}$: we have a problem

Matrix algebra

- Trades *cores* against *operations*
- LU factorisation becomes O(n)



- Messages usually a *single value*

What can we do.....

POETS

... when matrix algebra is an atomic operation?

Tomography

If you really want to know...

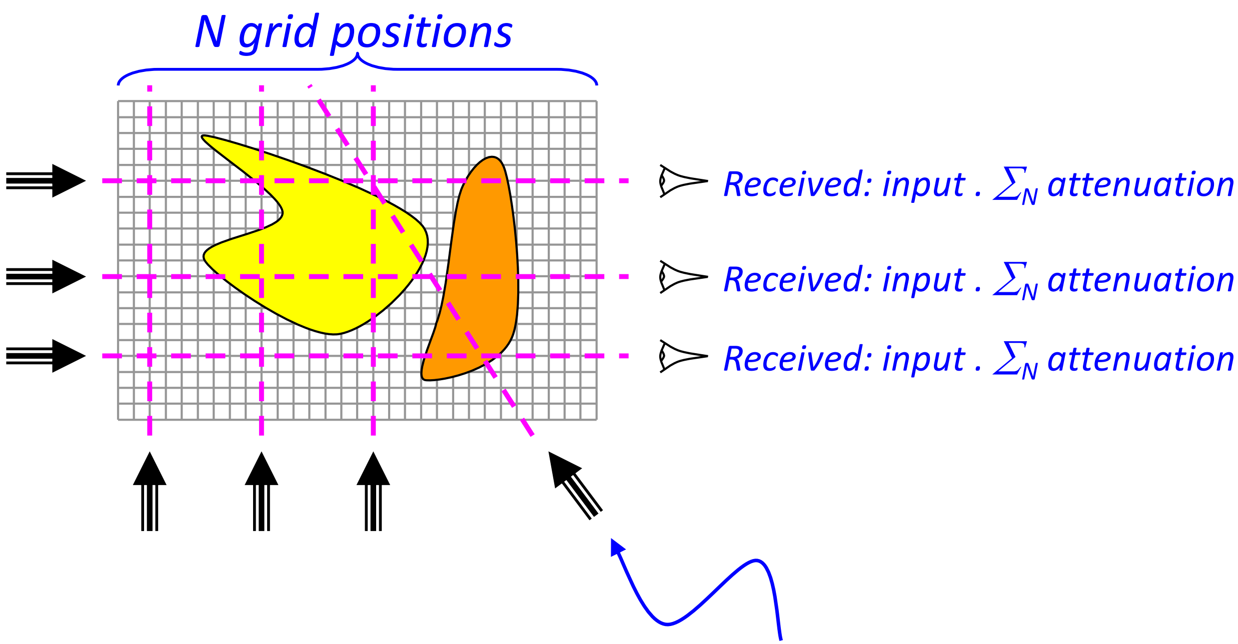
... how something works:

- Heave the cover off and take it to bits
- Unfortunately, this is usually
  - Dangerous
  - Expensive
  - Illegal
  - Amoral
  - Physically impossible
- So you have to use tomography (CAT)

CAT scans (in one slide)

POETS

- Non-invasive
- N.M unknowns
- K equations



- The trick is to get k =O(M.N)

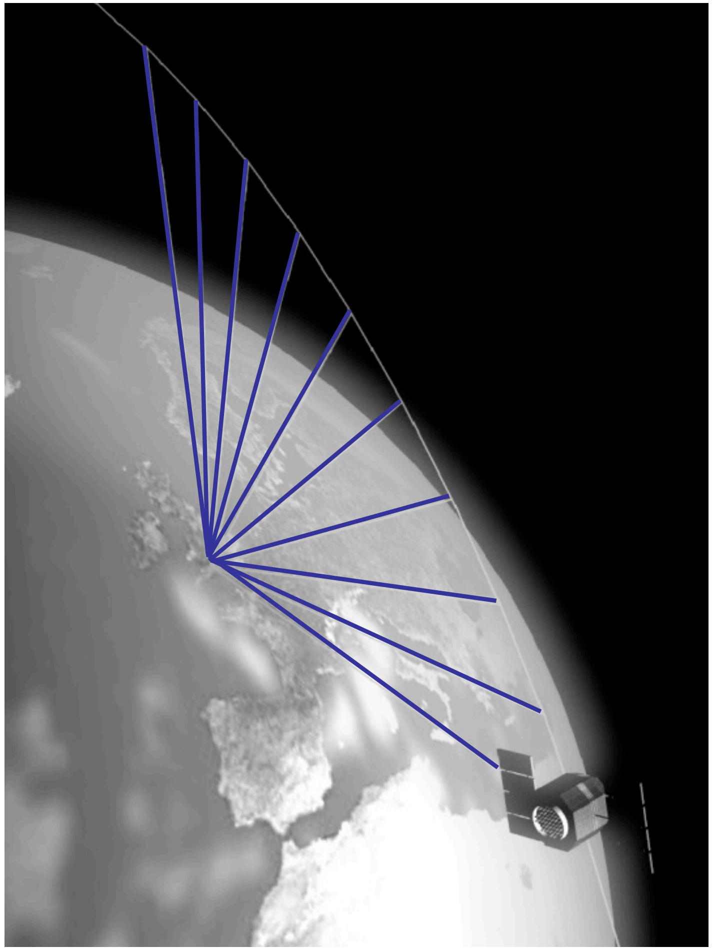
Arbitrary number of integral equations in arbitrary directions

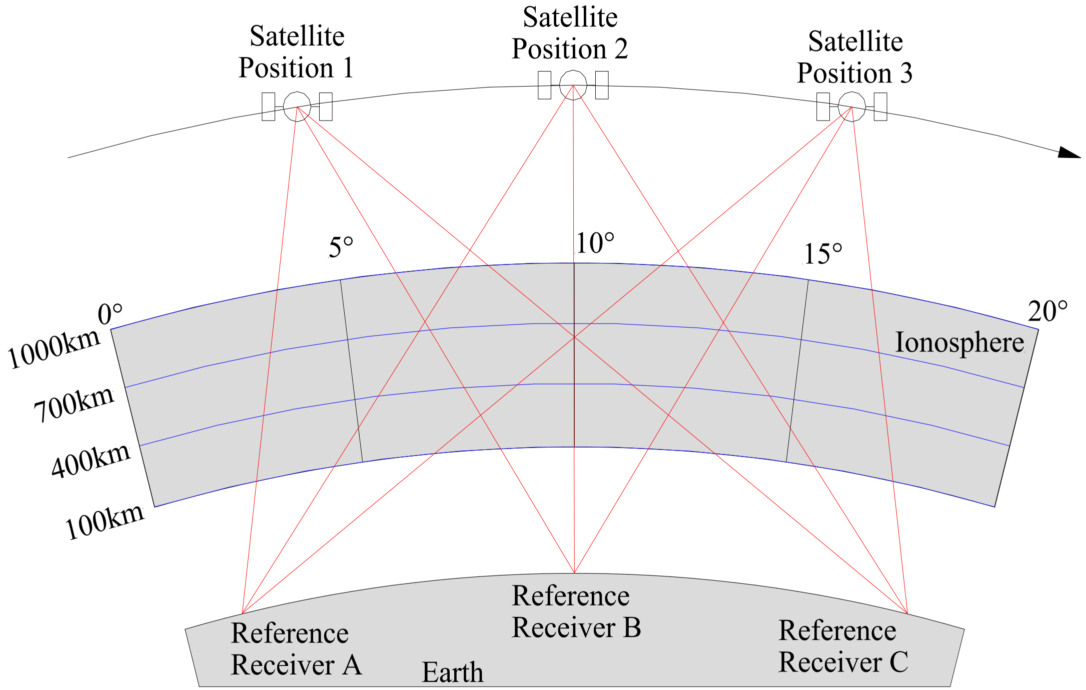
Invert equation set to get the attenuation at every internal point

Tomographic imaging

Ionospheric weather

POETS

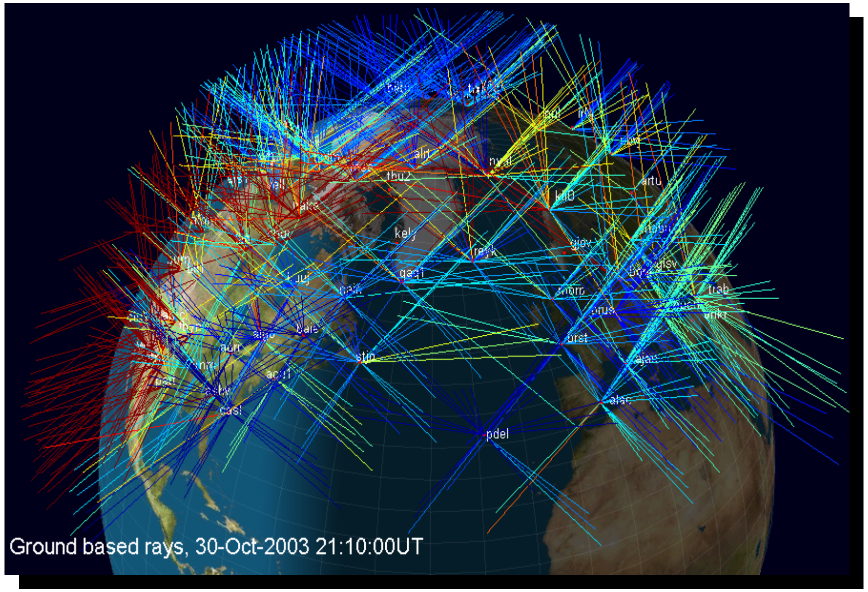




(C.N. Mitchell, University of Bath)

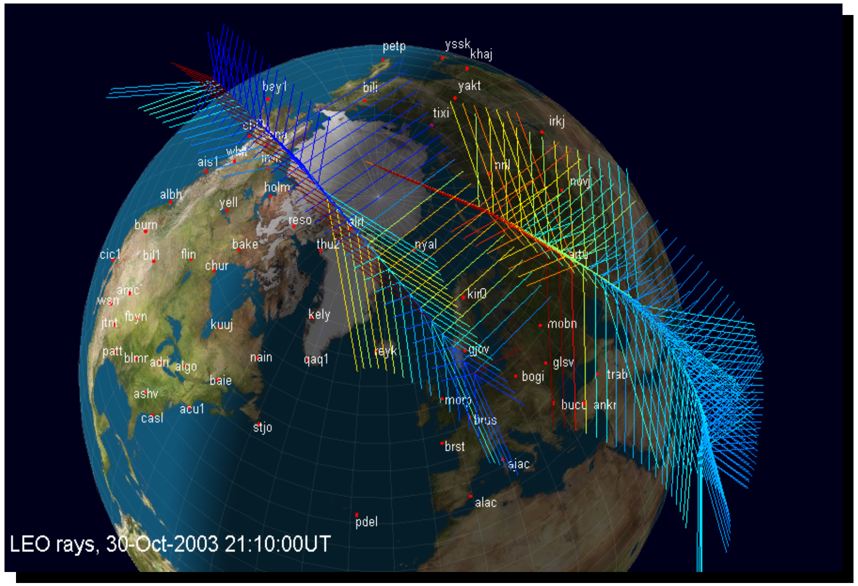
Equation set non-uniform

POETS



- Ground based

- Space based

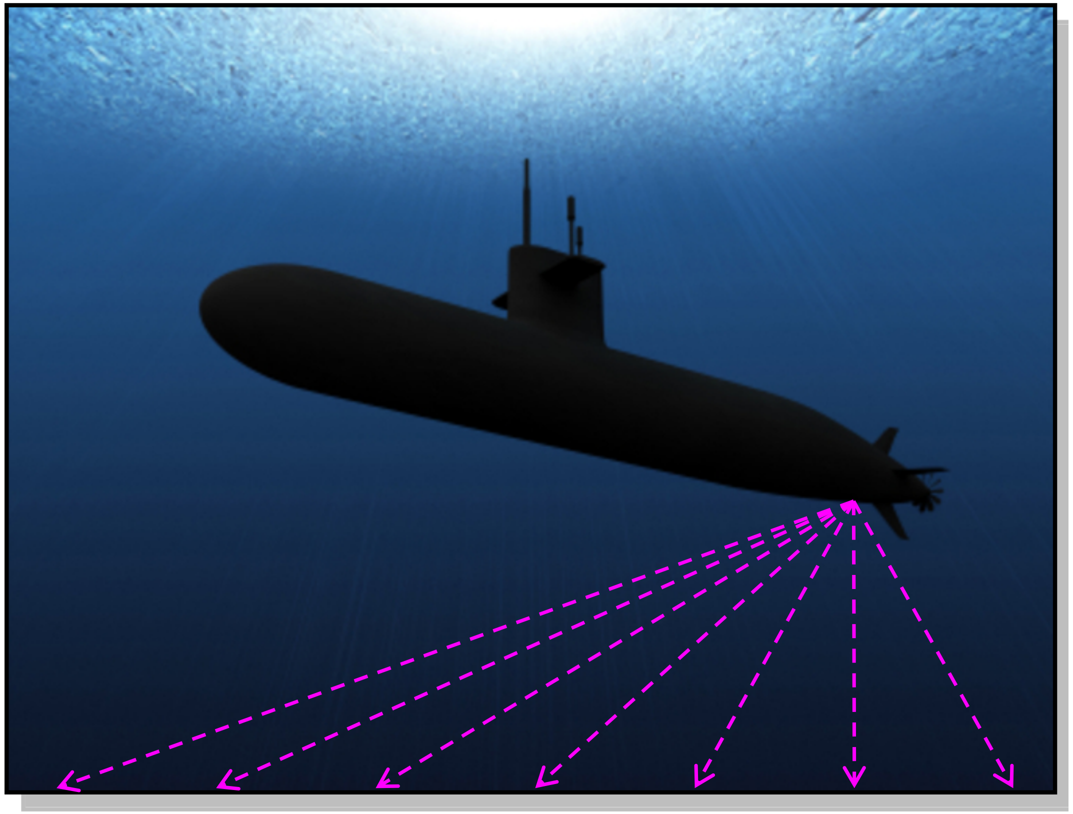


(ibid.)

Equation set secret

POETS

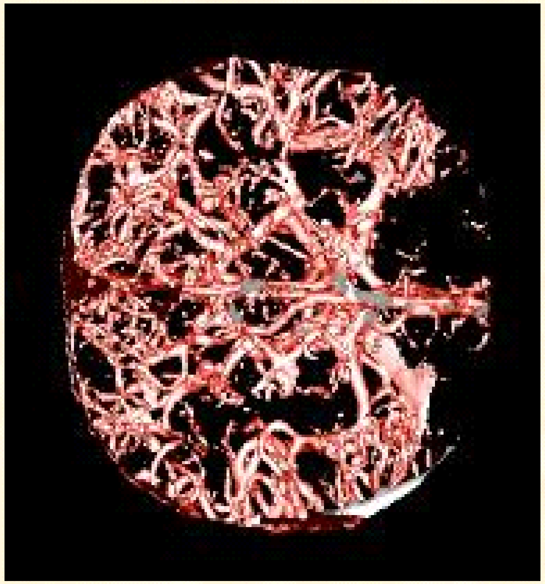
- Inverse field problems
  - Detection of submerged cylindrical magnetohydrodynamic anomalies



POETS

Medical

- Detailed non-invasive imaging of biological structures
  - Bones, brains, vascular systems

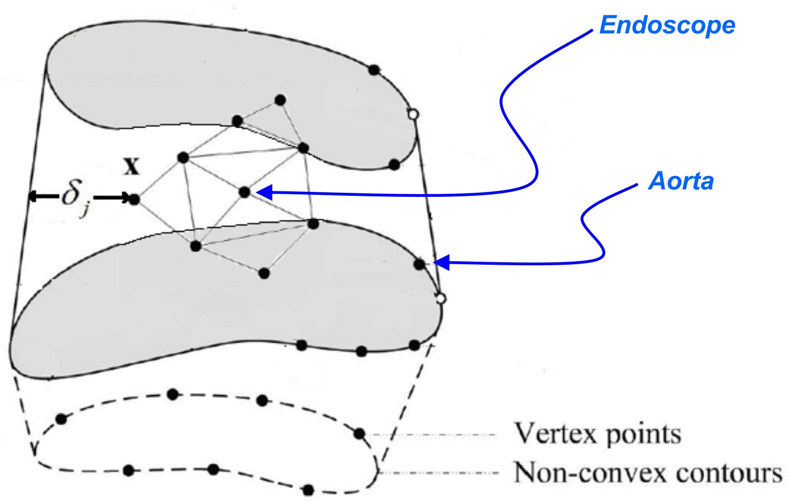


Cranial blood vessels imaged by X-ray tomography

Surgical

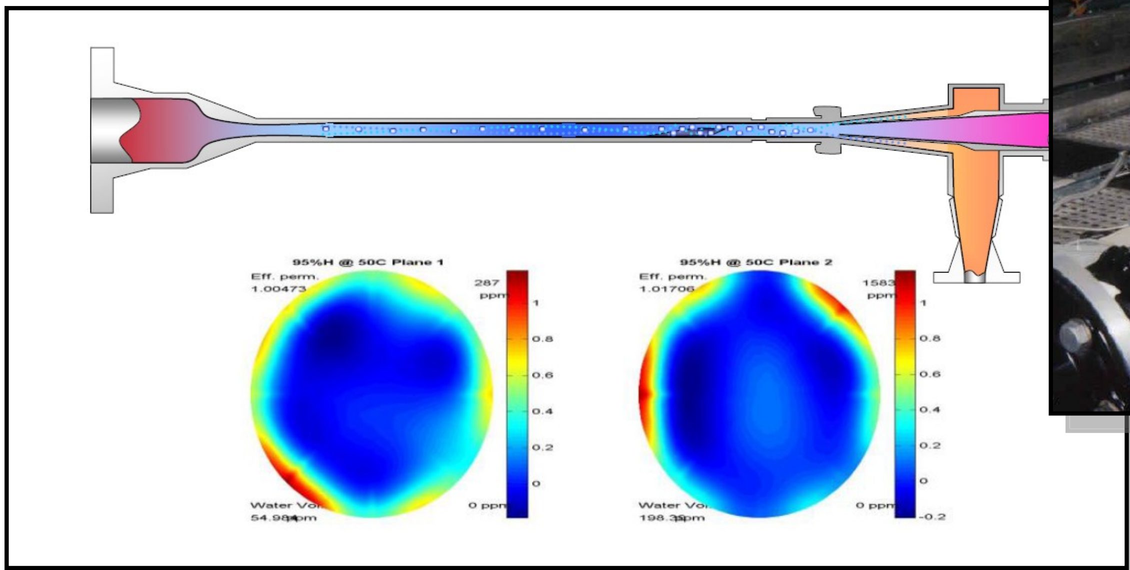
POETS

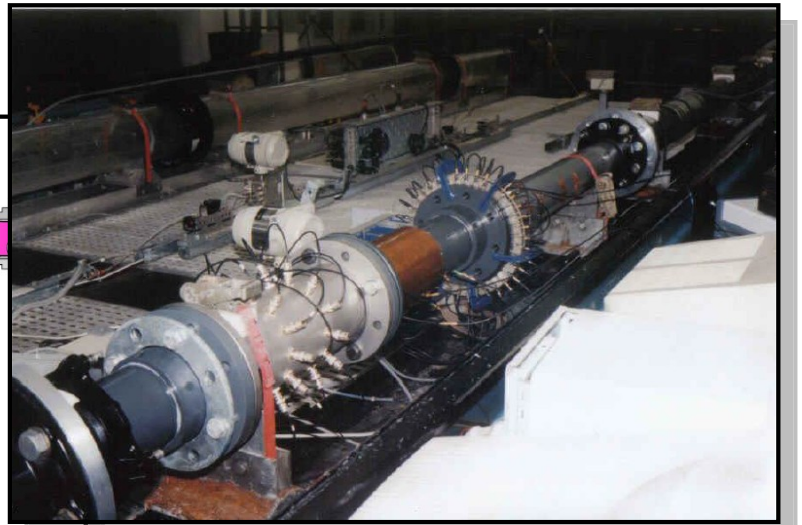
- Compute intersection/closest point-pair between 2 objects in 3D space
- Challenges: moving + accurate + fast
- Update $O(10^4)$ points at 1kHz: $O(10^6)$ points/s



Industrial

- Production line quality control
  - Mixing efficiency, void detection, structural integrity





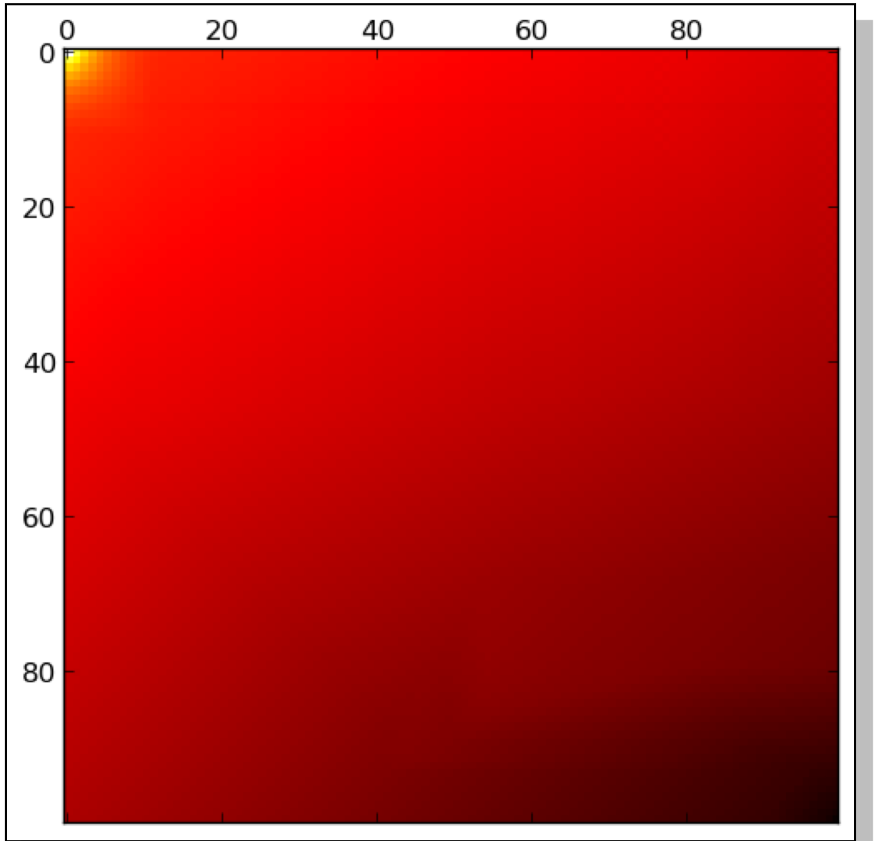
(*W. Yang, University of Manchester*)

POETS

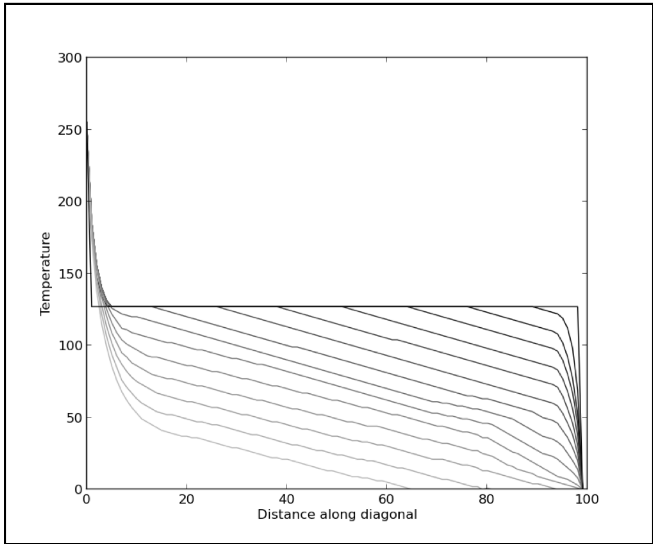
Outline

- The rise and asymptote....
- Simulation - there's a lot of it about
- Event-based simulation
- What is the user base?
- Where does all the time go?
- Down amongst the Hard Sums
- A brief meander into reliability
- Some pictures of hardware

Using event-based simulation



Finite difference heat diffusion canonical 2D square grid:

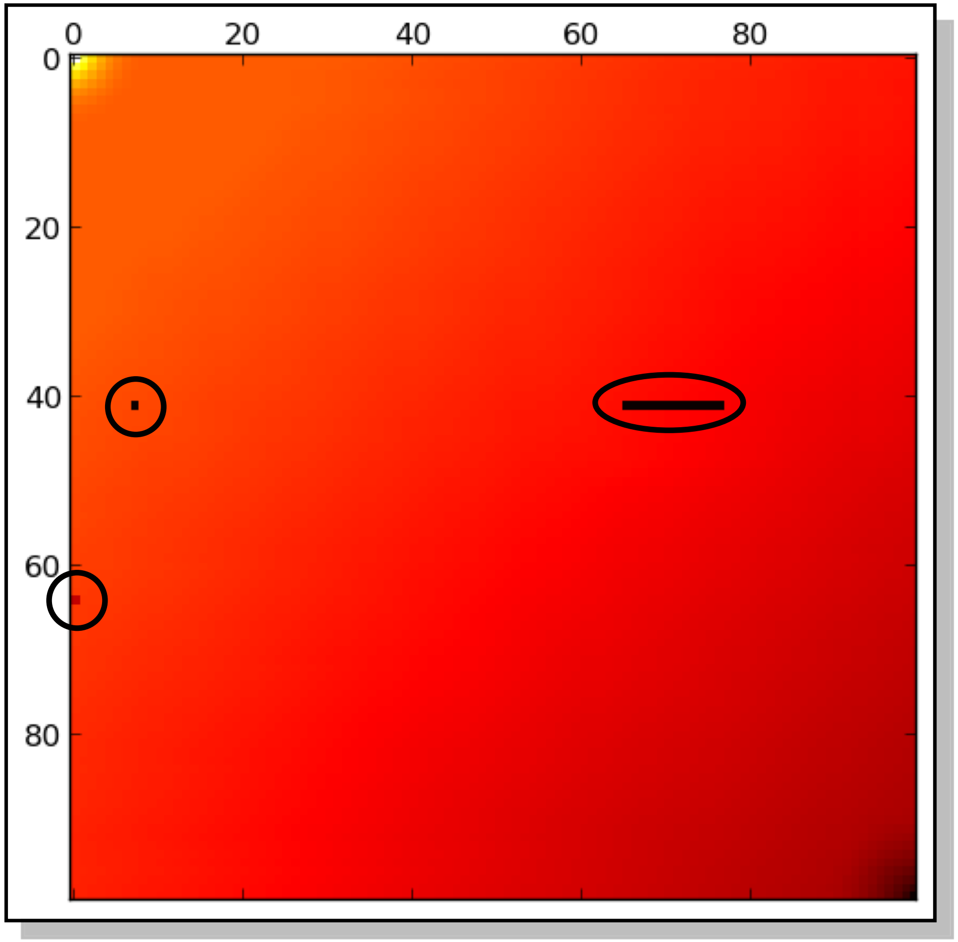


Diagonal temperature profile vs iteration

When things break:

POETS

- System with 1000000 cores
- Unrealistic to expect 100% uptime
- What happens when cores die?
- *Algorithm* 'self-heals' around unresponsive core



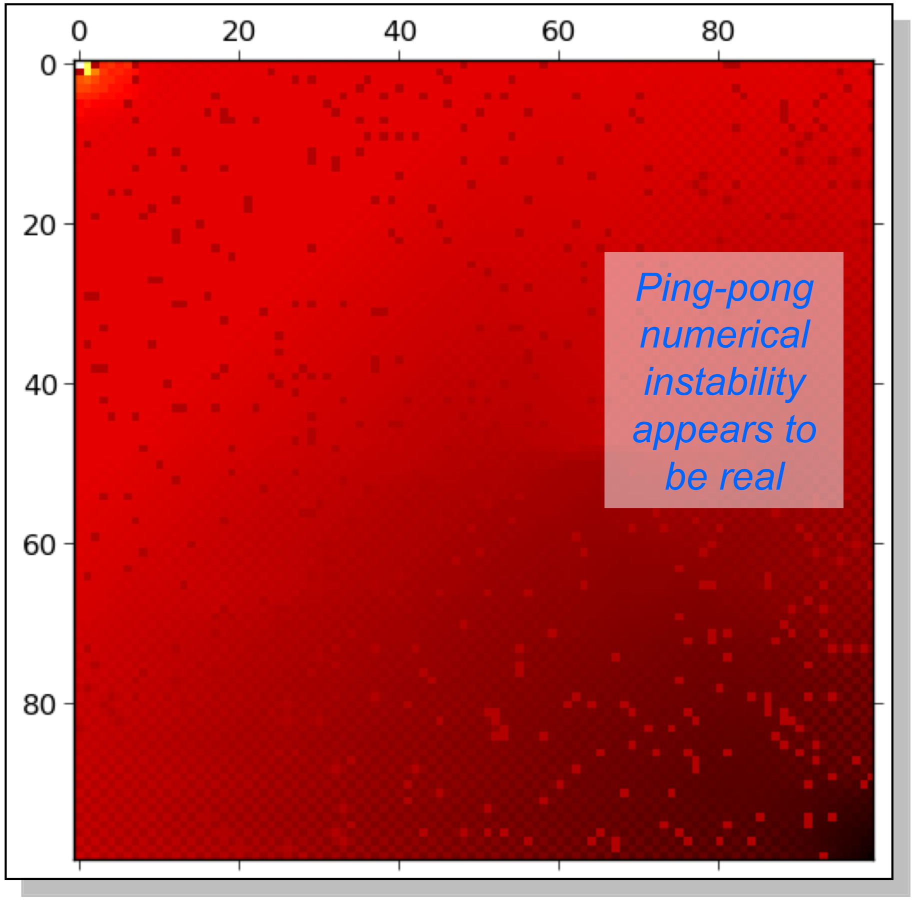
On the nature and consequence of failures

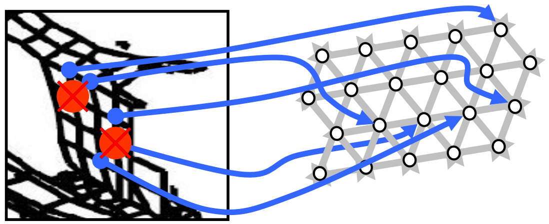
- Conventional electronic system:
  - Stuck-at, bridging, crosstalk....
  - Arbitrary effects propagate - usually disabling
- Event driven technology
  - Interconnect asynchronous
  - Core/node fails *silently*
  - Affected mesh sites disappear from *model*
  - Solution *algorithm* unaffected

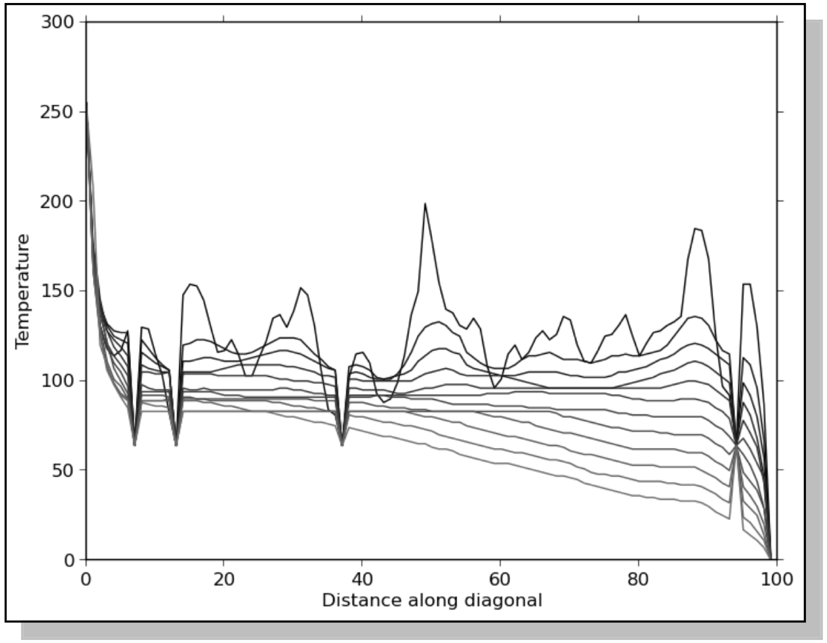
5% device death

POETS

Results degrade gracefully





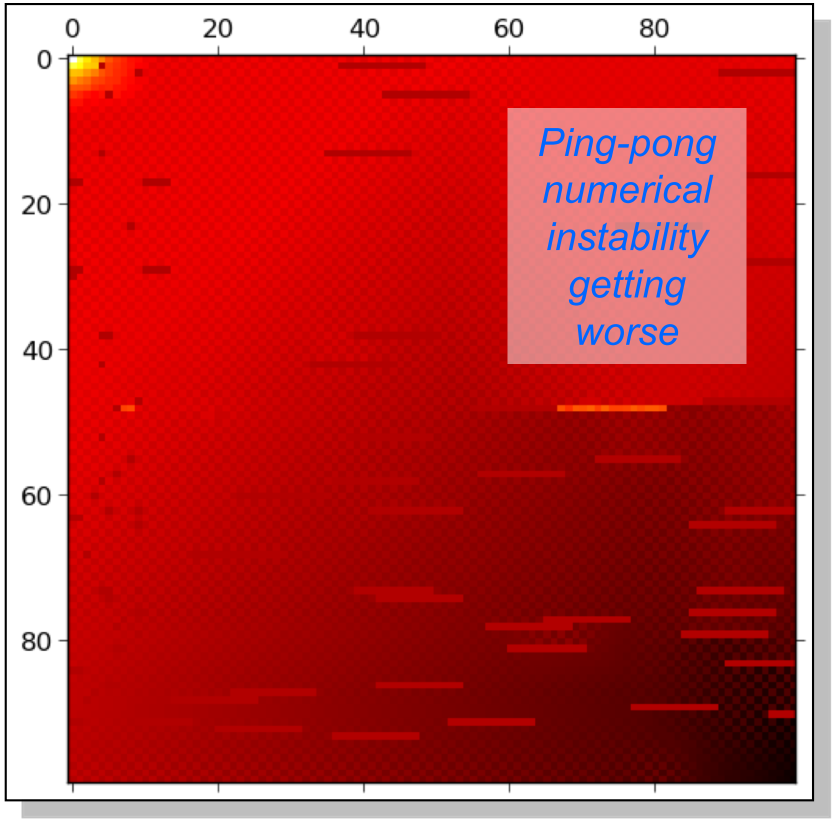


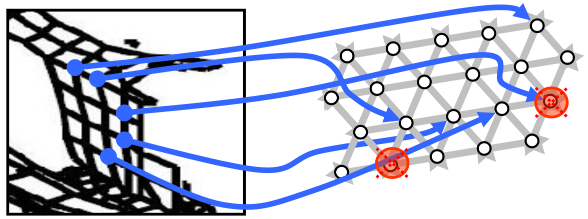
Diagonal temperature profile vs iteration

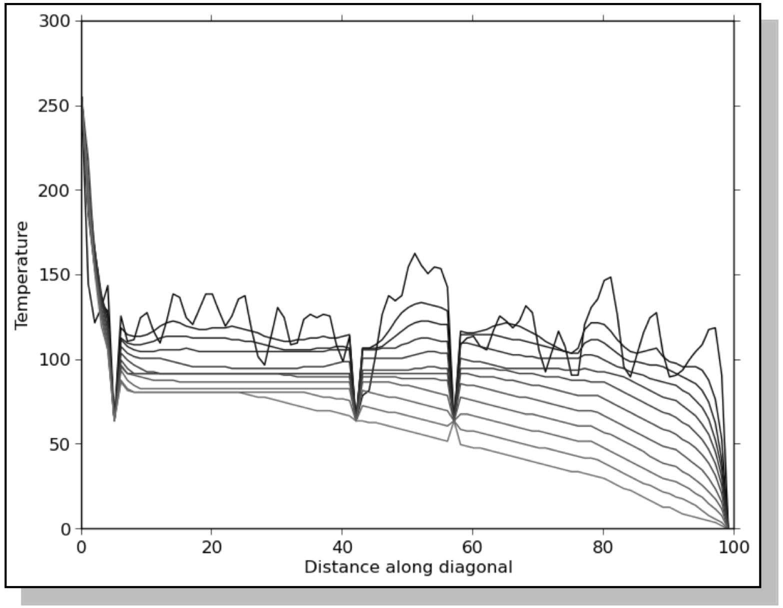
5% core death

POETS

Results degrade gracefully



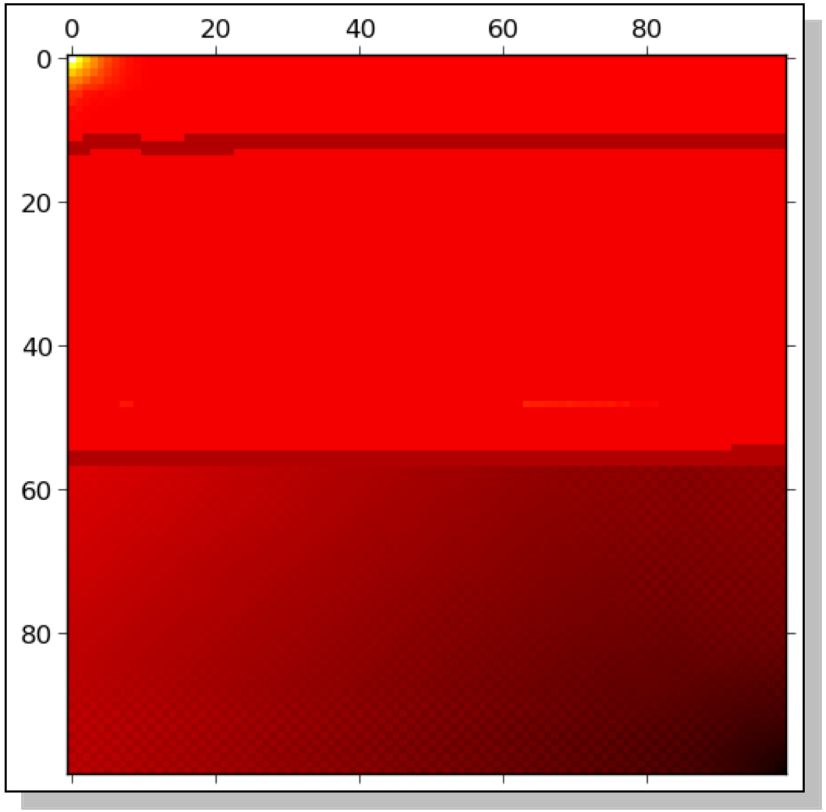


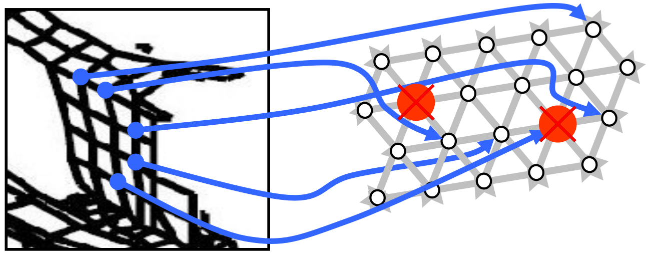


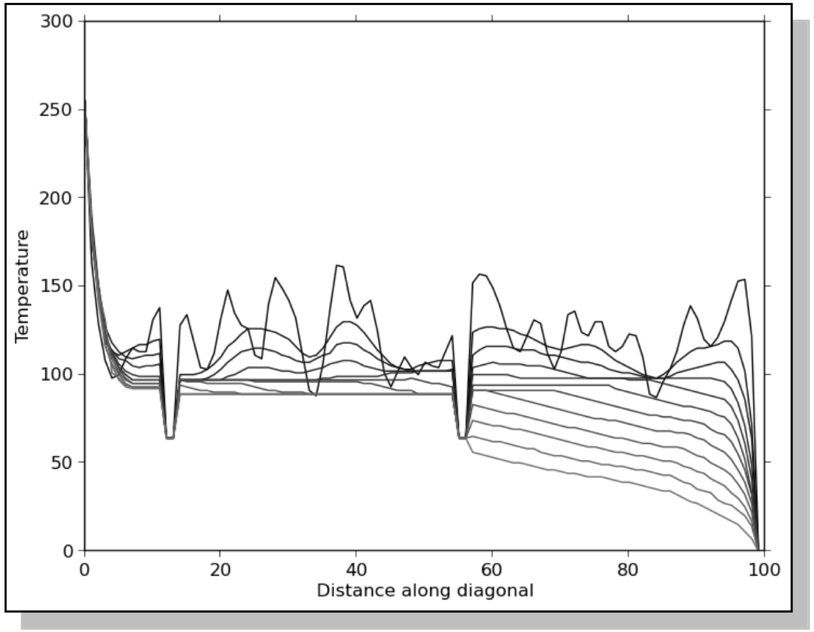
Diagonal temperature profile vs iteration

5% node death

Substrate damage has *severely perturbed* computational model







Diagonal temperature profile vs iteration

Outline

- The rise and asymptote....
- Simulation - there's a lot of it about
- Event-based simulation
- What is the user base?
- Where does all the time go?
- Down amongst the Hard Sums
- A brief meander into reliability
- Some pictures of hardware

SpiNNaker

105 machine

5 racks:

- 5 racks x
- 24 boards x
- 48 nodes x
- 18 cores
- = *103680* cores

106 machine

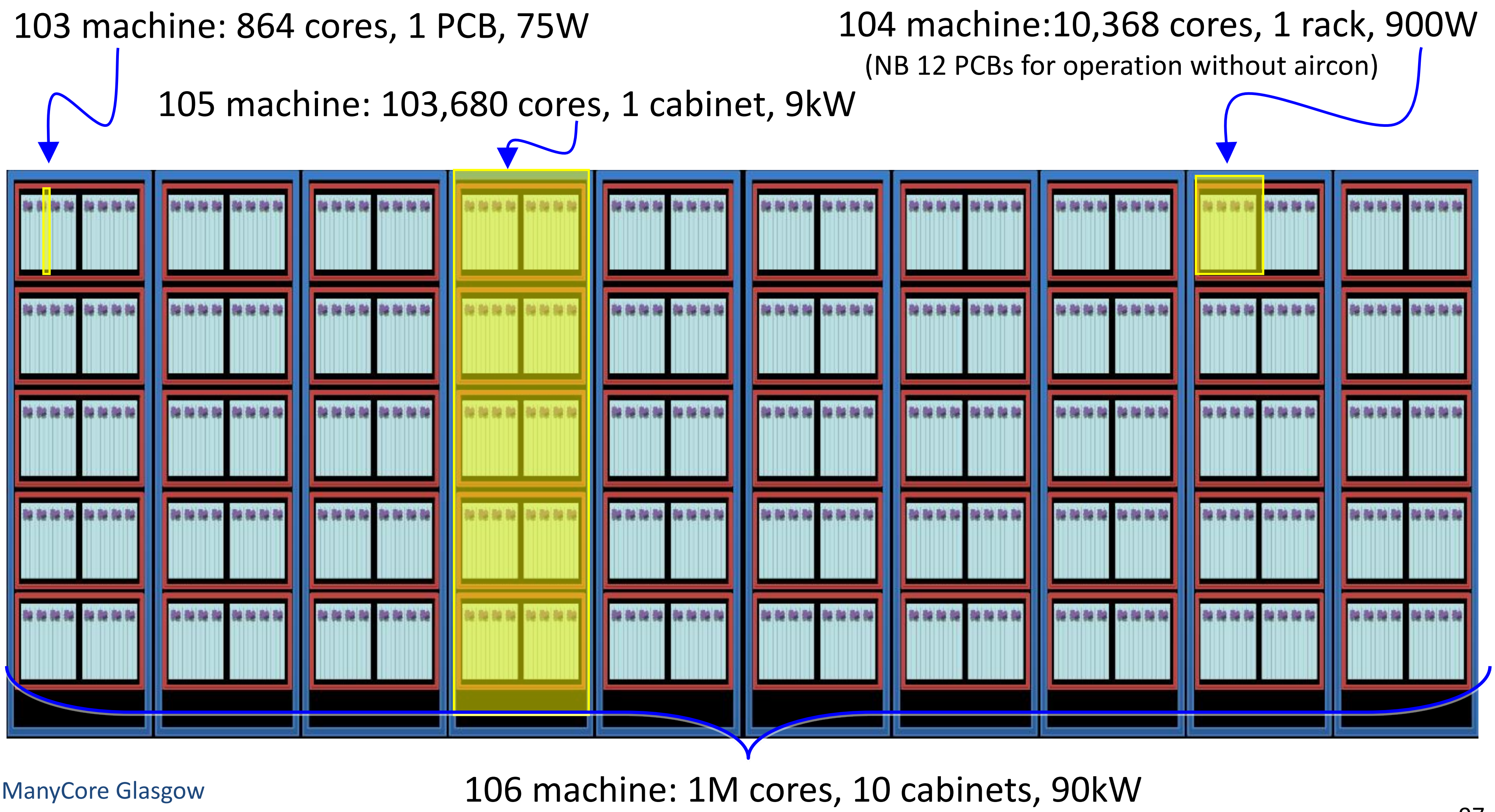
50 racks:

- 50 racks x
- 24 boards x
- 48 nodes x
- 18 cores
- = *1036800* cores

- Each core nominally hosts 1000 neurons

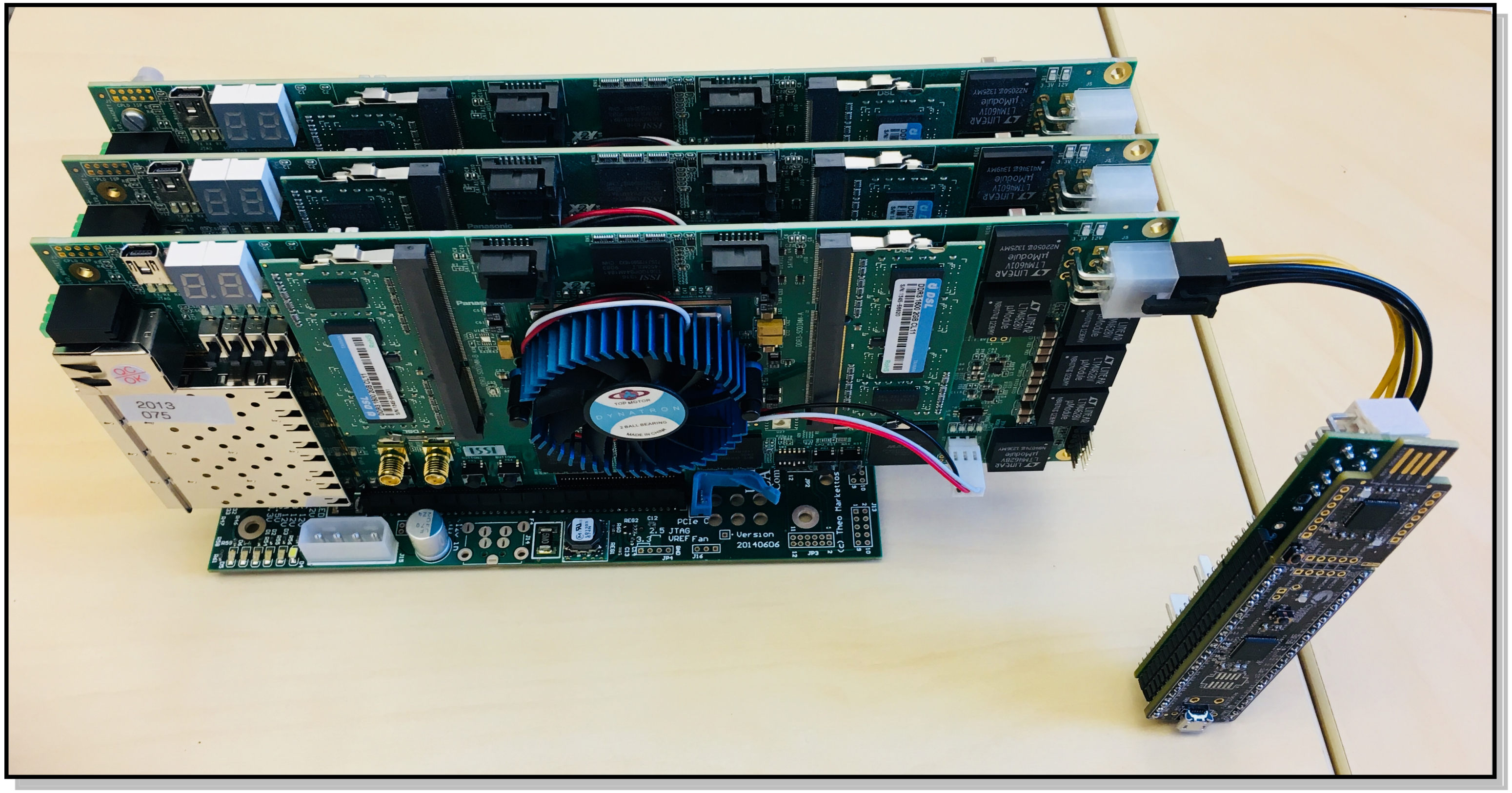
105 machine

...and the machine yet to be assembled:



POETS

POETS



Prototype: Tinsel

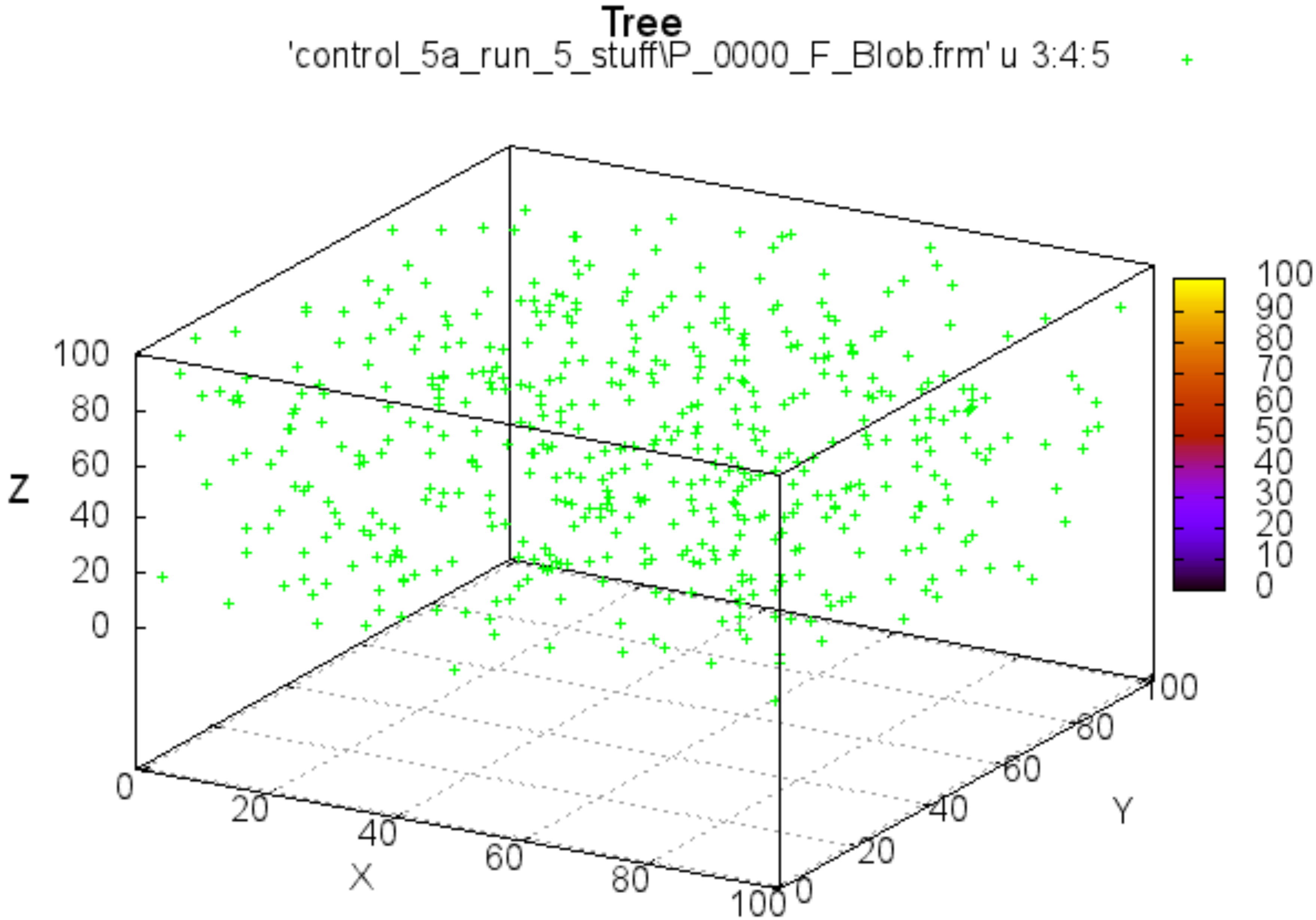
- Tinsel = multithreaded POETS processor
- 32-bit multithreaded RISC-V
- 64 cores x 16 threads/Tinsel = 1024 P-cores
- 16 caches, 16 mailboxes, 2 DDR3 controllers
- 40% of DE5-NET FPGA board
- Clocks at over ~300MHz (fast for a softcore)
- Each thread nominally hosts 1000 problem devices

By 2025...

A desktop machine ...

- a *personal* computer -

... will have 25000 cores


Tree
'control_5a_run_5_stuff\P_0000_F_Blob.frm' u 3:4:5
Z
X
Y
100
80
60
40
20
0
100
90
80
70
60
50
40
30
20
10
0