Speeding up your model: how to do it and the benefits

Sam Worthington

Model speed is the interdependent result of ...

Model architecture and data

Software platform    Hardware
Part 1
Model design patterns and the hardware to implement them

A capital model in memory

• 128GB
• 256GB
• 512GB
• 1TB
• 2TB
• 4TB
• 8TB
• 24TB

What is the peak memory for a large GI capital model?
(100k sims, 400 classes, 2 future underwriting years, 10 years of accounts)?

What is the maximum available RAM on a single server?
1. Single core

Example: Import data, simulate gross and losses for class

Node 1

Import

Simulate

Aggregate

Single core

Level 1: CPU runs tasks sequentially on a single core; clock speed is key factor to determine performance

Pros: Default approach: no need to change hardware or restructure model

Cons: Poor run time performance as no parallelism possible

2. Task farm

Node 1

Node 2

Node 3

Import

Simulate

Aggregate

Task Farm

Level 2: Head node runs serial sections of model and manages the data flow. Farms out tasks to an array of worker nodes using direct network communication of data

Pros: Good for 'embarrassingly parallel' problems which are computationally intensive;

Cons: Head node is a potential bottleneck limiting speed / scalability. Pinch point to aggregate large datasets
2.a Task farm – Single machine, multi-core CPU

**Intel i9 processor**
128GB RAM
10 cores

![Intel i9 processor image](image)

<table>
<thead>
<tr>
<th>Multi-core CPU</th>
<th>Description: CPU manages the data transfer and task allocation to the available cores</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pros</strong>: Make use of existing hardware for no extra work / cost assuming the software allows threading;</td>
<td></td>
</tr>
<tr>
<td><strong>Cons</strong>: Not scalable – can only use available cores and RAM; Not practically feasible to add more cores</td>
<td></td>
</tr>
</tbody>
</table>

Physical proximity matters in terms of speed

![Motherboard diagram](image)

GPU is physically located much further from the CPU than RAM is from the CPU, creating an overhead for data transfer.
2.b Task farm – Single machine using GPU

**Gigabyte Nvidia GeForce GTX 1050**

- 4GB RAM
- 768 CUDA cores

**GPU**

<table>
<thead>
<tr>
<th>Description</th>
<th>Machine’s CPU transfers data to the GPU, which distributes threads across GPU cores</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pros</strong></td>
<td>Very large number of cores available cheaply; Fast for complex calculations e.g. Asian Options</td>
</tr>
<tr>
<td><strong>Cons</strong></td>
<td>Low RAM per core so cannot hold many results in memory; Overhead to transfer data outweighs benefit to parallelise simple calculations</td>
</tr>
</tbody>
</table>

*GPU is a specialist chip designed for parallel, independent computationally-intensive calculations e.g. graphics display*

---

2.c Task farm – Grid (including Cloud)

**Grid / Cloud**

<table>
<thead>
<tr>
<th>Description</th>
<th>Grid head node manages the data and allocates tasks using direct network communication</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pros</strong></td>
<td>“Infinitely” scalable – keep adding machines; Better utilisation in a general purpose grid for other tasks</td>
</tr>
<tr>
<td><strong>Cons</strong></td>
<td>Hardware (and software) costs; Performance constrained by network transfer time; Bottleneck in parts of the model that cannot be parallelised</td>
</tr>
</tbody>
</table>
Speed is no longer just a case of shrinking transistors on a CPU

From 2010 chip speed has leveraged vector processing

- From 2010 Intel added an Advanced Vector eXtension ("AVX") co-processor to its CPUs
- AVX allows parallel processing of data – increased speed
- With each generation of AVX, Intel has doubled the length of vectors and hence parallel vector processing

2.d Task farm – Multi-core CPU + vector extensions

Vector extensions

Description: An instruction is performed on a vector of values, not just on a single value

Pros: Automatically included (and upgraded) in modern chipsets at no extra cost; Highly effective at processing large volumes of stochastic data;

Cons: Requires software to execute SIMD instructions; Not scalable – can only use available cores and RAM;

Vector extensions permit computers to use SIMD (Single instruction, multiple data) to process the same operation on multiple data points simultaneously.
2. Task farm

Task Farm

**Level 2**: Head node runs serial sections of model and manages the data flow. Farms out tasks to an array of worker nodes using direct network communication of data.

**Pros**: Good for ‘embarrassingly parallel’ problems which are computationally intensive;

**Cons**: Head node is a potential bottleneck limiting speed / scalability. 

**Pinch point to aggregate large datasets**

3. Symmetric parallelism

**Single core**

**Description**: Data is distributed symmetrically across a set of machines (e.g. by ‘simulation’). Each machine runs the complete workflow on its own data.

**Pros**: Potential for symmetric scalability and performance; Applicable for the largest of models

**Cons**: Requires development of software parallel algorithms;
Parallel aggregation functions

Node 1

Sim 1 2 3
Row
1 0.5 1.2 3.5
2 1.5 5.4 5.3

Sim 1 2 3 4 5 6
Row
1 0.5 1.2 3.5 1.5 2.1 5.3
2 5.1 4.5 3.5 1.5 5.4 5.3

Median
Row
1 1.65
2 4.0

Node 2

Sim 4 5 6
Row
1 1.5 2.1 5.3
2 5.1 4.5 3.5

Sim 1 2 3 4 5 6
Row
1 1.5 2.1 5.3 1.5 5.4 5.3
2 5.1 4.5 3.5 1.5 5.4 5.3

Median
Row
1 1.65
2 4.0

3. Symmetric parallelism – Grid

Single core

Description: Data is distributed symmetrically across a Grid of machines (e.g. by 'simulation'). Each machine runs the complete workflow on its own data.

Pros: Potential for symmetric scalability and performance; Can run on a grid of any size machines (large / low spec);

Cons: Requires software to be developed with parallel algorithms.
Part 2
Using speed to improve financial modelling

Financial models support GI business objectives ...

Increase Revenue

Reduce Costs

Optimise Capital and Risk
Rationally, the run time constraint used to result in...

- **Fit for purpose**
- **Calibrated to a basis**
- **Materiality**

Cost
- Justify and explain inconsistent results between Capital, Accumulation, RI pricing tools
- Run on 1 year / ultimate and GAAP / SII bases separately
- Outside model adjustments, increasing complexity and operational risk

---

Typical financial model setup

- **Internal Model**
- **Inwards Pricing**
- **Outwards RI Pricing**
- **Accumulation Management**

**Key Features**
- Silo Models → independently operated, and therefore silo teams
- Different platforms → impossible to share code across models
- Duplication of code → inconsistency of modelling, development and maintenance cost
Better financial model setups might look like …

Key Features
Model at greatest granularity and then re-use across multiple applications (potentially with separate UIs) … or deploy through a single integrated application

Real-time pricing of marginal net impact

Marginal net risk / capital impact at point of underwriting
Consistent results to capital model
**Reinsurance purchase after market turning event**

The need for modelling speed is most acute after a market turning event. E.g. Hurricane Katrina

- RI cover exhausted / unacceptably low
- Insurer capital position threatened
- RI terms / prices changing several times a day
- High RI prices so cost of reinstatement is material

---

**Real-time risk reporting**

- Investment – daily balance and 3month / 1year VaR
- Pension schemes – real-time funding ratio and risk
- Banking – overnight and 1year VaR
- Insurance – quarterly SCR?

To achieve real-time risk capital modelling it would need automation:

- update of Gross and RI paid and incurred claims, earned and unearned exposure
- Update of economic risk factors e.g interest rates
- Off-cycle calibration to review and update calibration
Faster financial models deliver benefits …

- **Increase Revenue**
  Faster quote response for CAT / large policies

- **Reduce Costs**
  Integrated model with single calibration; Reduced hardware

- **Optimise Capital and Risk**
  Optimise policy selection; portfolio management; RI purchase after MTE

... and a better life for the modelling teams

THANK YOU