Marriott Sinclair LLP
Financial Engineering

A presentation to EMiT
2015

Overview

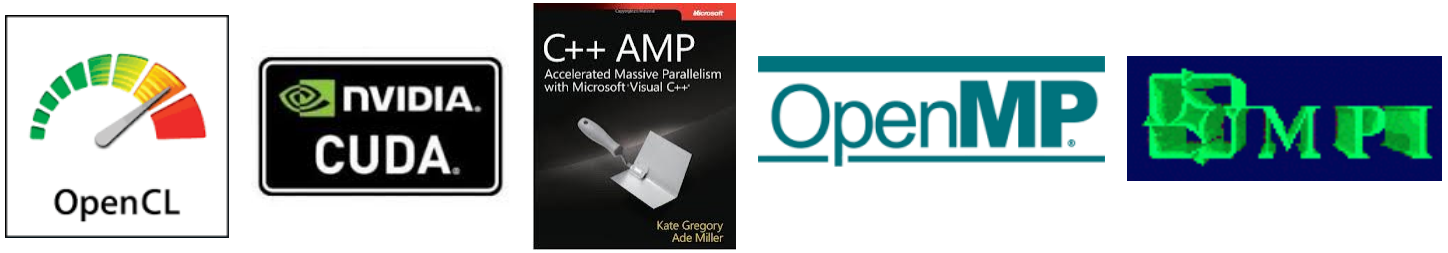
- The Problem: Technology
- The Problem: Computational Finance
- The Solution
- Tyche Design
- Future-Proofing and Modularity
- Performance
- Ease of Use
- Processing Stages
- Efficiency

The Problem: Technology

- **Moore's Law** has **ended**, at least in terms of increasing clock speed. As Herb Sutter has said, the **free lunch** is **definitely over**.

- Instead of a single CPU, we now have a **zoo** of **different architectures** to write code for: multi-core, vector processors, GPUs, FPGAs, accelerators like Intel Xeon Phi…

- Likewise, there is a **forest** of **different languages** to target these different pieces of hardware: OpenCL, CUDA, AMP, OpenMP, MPI, VHDL…



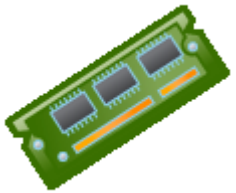
- Programming in these languages, and for this hardware, can require much expertise. It can also take a long time!

The Problem: Computational Finance

- We supply **risk management** software to insurers, banks and hedge funds.

- Increasing regulation (Solvency II, Basel II/III) means that financial institutions must maintain detailed models of the material risks to which they are exposed, and use these to manage their business as well as determining **capital requirements**.

- Models are also **large** and **complex**. Peak working set requirements can be **hundreds of GB**. Some current models can approach a working set of 1 TB. The number of Monte Carlo scenarios required can be up to 1e6.



- These models, particularly for insurers, take a long time to run! **Overnight** or **whole-week** runs are common. They also can take many man-years to build.

- SIMD is a possible solution to model runtimes as models are typically Monte Carlo, and hence **embarrassingly parallel**. However, current software, particularly in insurance, does not take advantage of the latest hardware advances. Re-coding models directly in CUDA or similar will take many years.

The Solution

- We needed a platform that was **intuitive to use**, and didn't require low-level expertise in new languages. Finance professionals often don't have the time or inclination to learn these new languages.

- We also needed a platform that **could take advantage** of the **latest advances** in **hardware**, and continue to do so in the future.

- Finally, this platform needed to be sufficiently **flexible** that it could be turned to any computational finance problem of interest:
  - Pricing portfolios of (potentially exotic) assets – e.g. for a Basel II PFE calculation.
  - Determining capital
  - Strategy optimisation – e.g. portfolio optimisation, capital optimisation, reinsurance optimisation
  - Calculating insurance premiums
  - Projecting economic variables

- In summary "**supercomputer power in a package as easy to use as a spreadsheet**."

tyche™ : Design

- tyche™ design has been based on the following pillars:

  - **Future-Proofing**. We have abstracted away the low-level SIMD languages like CUDA and MPI behind a common interface layer. As new hardware and languages become available, we can insert those beneath this layer.
  - **Modularity**. tyche™ technology and codebase has been designed in a modular fashion so that it can be used in a broad range of environments. For instance, the tyche™ Engine can be used independently of the tyche™ GUI.
  - **Performance**. Monte Carlo calculations are often **embarrassingly parallel**. This means that they can be completely parallelised by scenario. tyche™ leverages the latest SIMD technologies to take advantage of this.
  - **Ease-of-use**. tyche™ intended end-users include non-programmers (e.g. actuaries). An intuitive interface that allows fast model construction is thus essential. At the same time, more tech-savvy users (e.g. quants & scientists) can leverage their C# and C++ knowledge if they choose to do so.

tyche™ : Future-Proofing & Modularity

>1 million lines of code and counting.....

Tyche GUI

Model design via **flowcharts**. Intuitive **scripting** language.

Tyche Engine

Construction and execution of a Directed Acyclic Graph of operations (**DAG**).

SIMD Engine

Common interface of ~300 core vector operations

CUDA Engine	AMP Engine	Intel Engine	OpenCL Engine	SystemVerilog Engine*

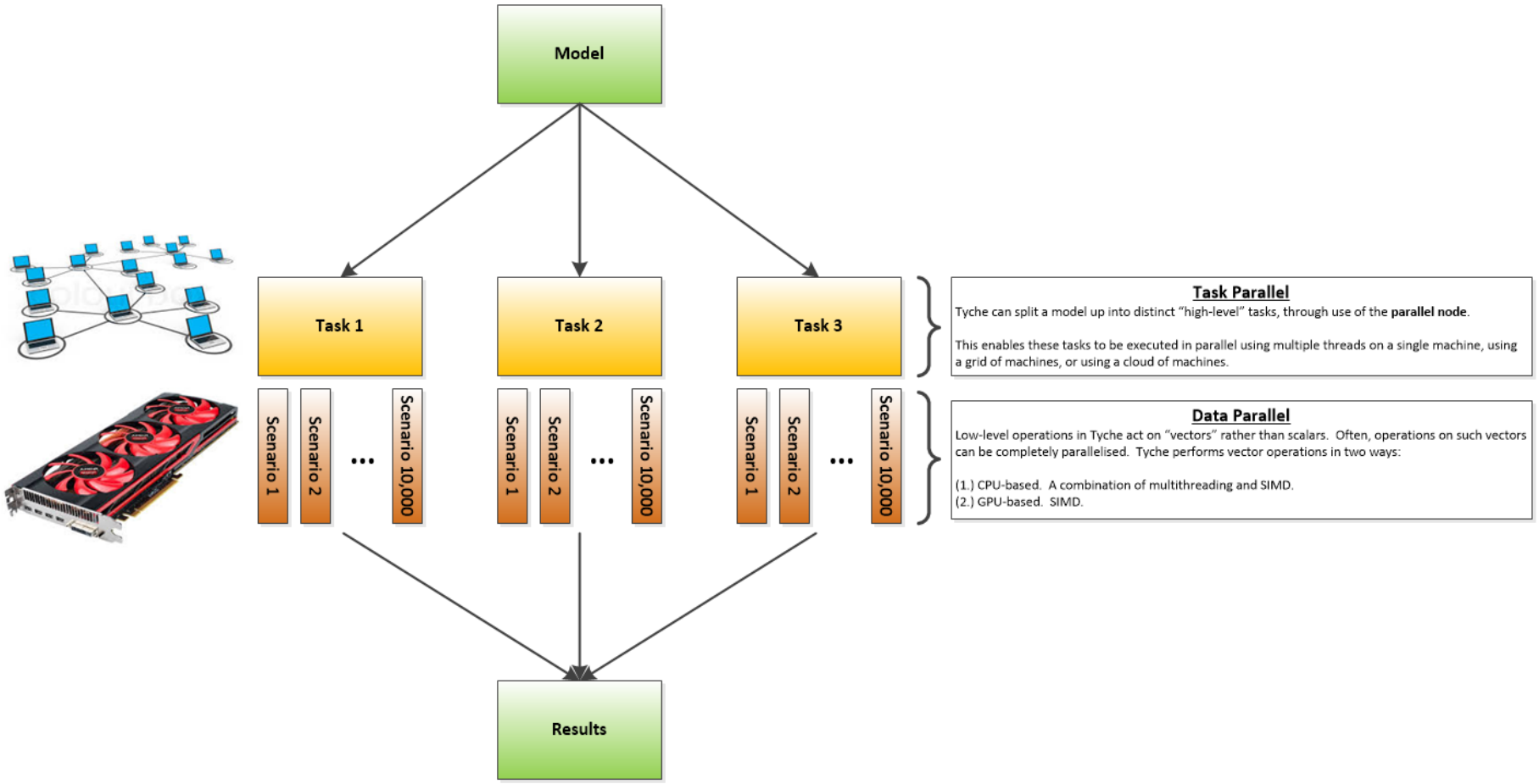
Each DAG node is executed as an **atomic operation**.

Hardware memory transfers (where relevant) are minimised.

Kernel size is minimised*.

*in progress

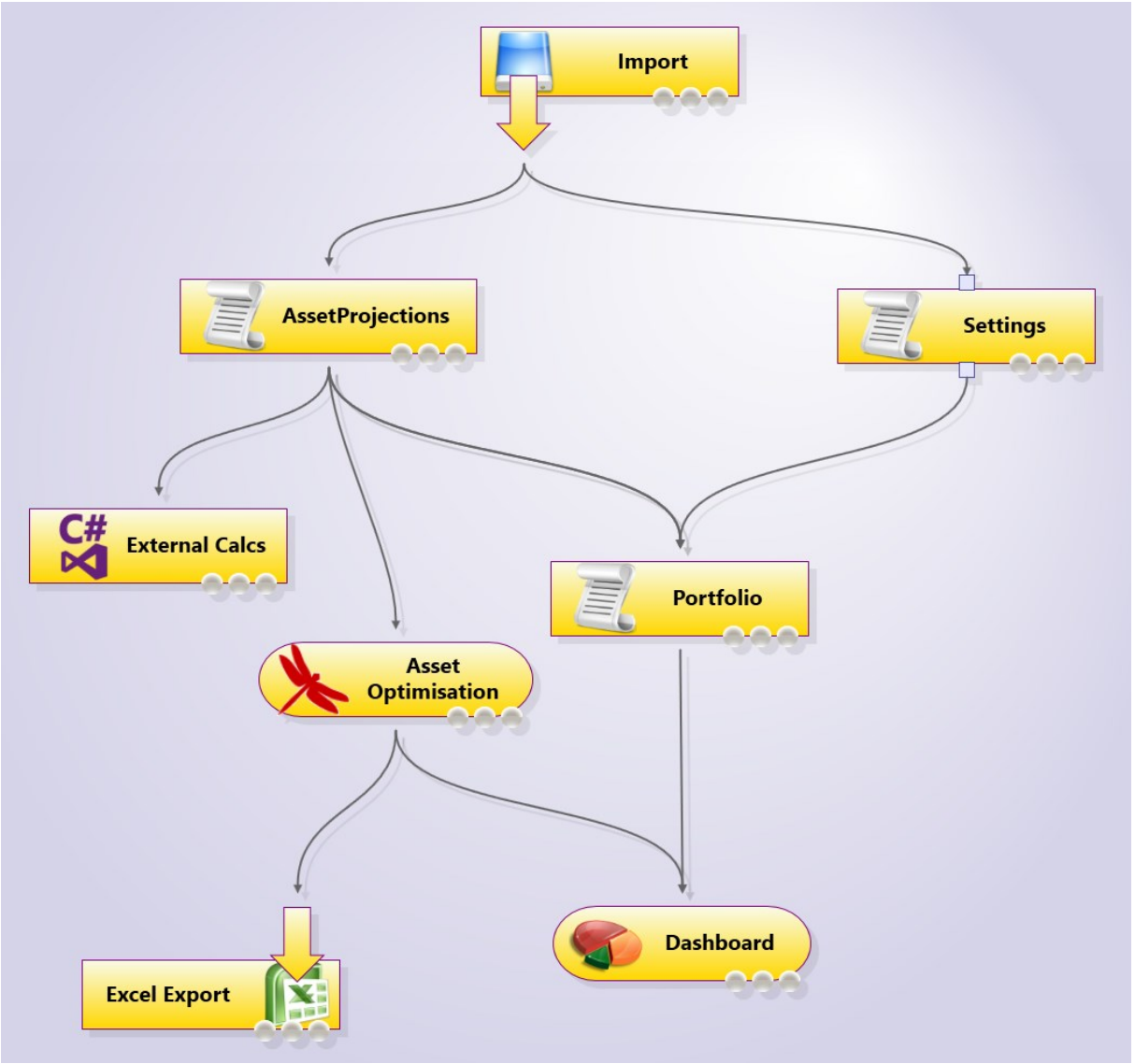
tyche™ : Performance



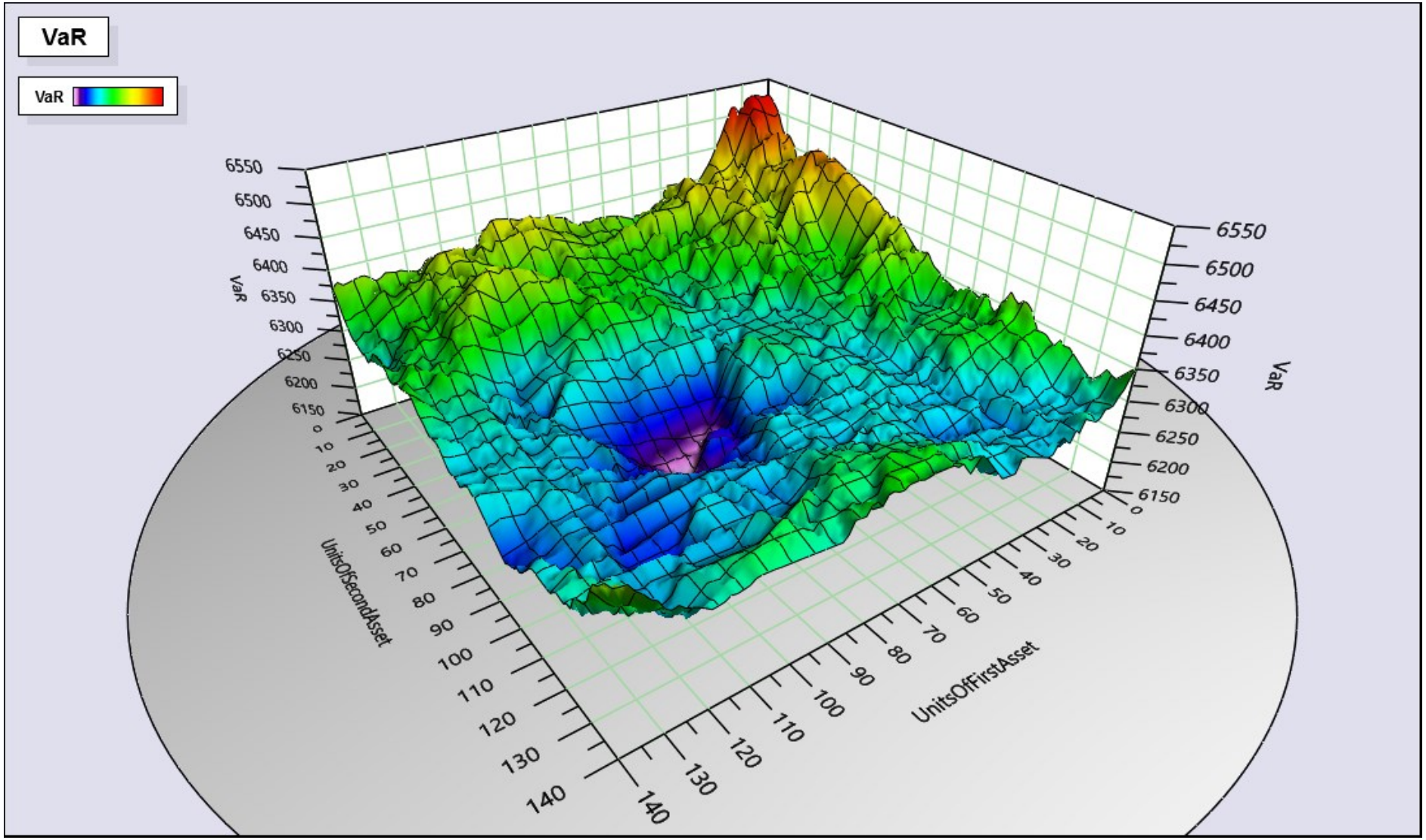
- tyche™ is designed from the ground up to leverage the latest parallel computing technologies.
- Models can be distributed over a grid of machines, and on each machine the GPU may be used to further accelerate computation.
- Models with a large "working set" of memory that are too big for the GPU can make use of SIMD on the CPU.

tyche™ : Ease-of-Use and the Flowgram

- The tyche™ user interface is based around the **flowgram™**.
- This allows flowchart-based models to be constructed quickly and intuitively.
- A **flowgram™** allows different technologies to work together in the same model, such as:
  - T#
  - C#
  - C++
  - Visual Basic
  - Numerical optimisers & rootfinders
  - Excel interaction: import, export & calculation
  - Grid- and SIMD-based computing
- Flowgrams™ may be constructed from the "core" tyche™ node types, or by making use of higher-level functionality in the tyche™ **Toolkit**.

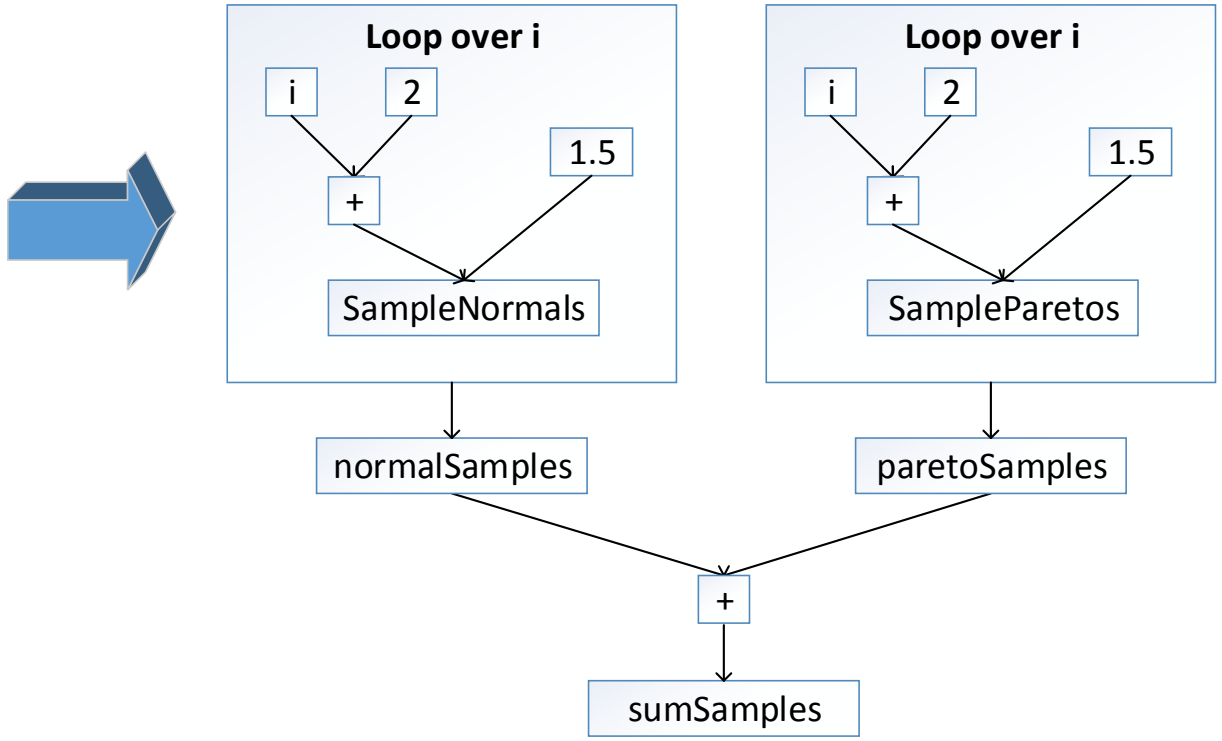


Quick Demo

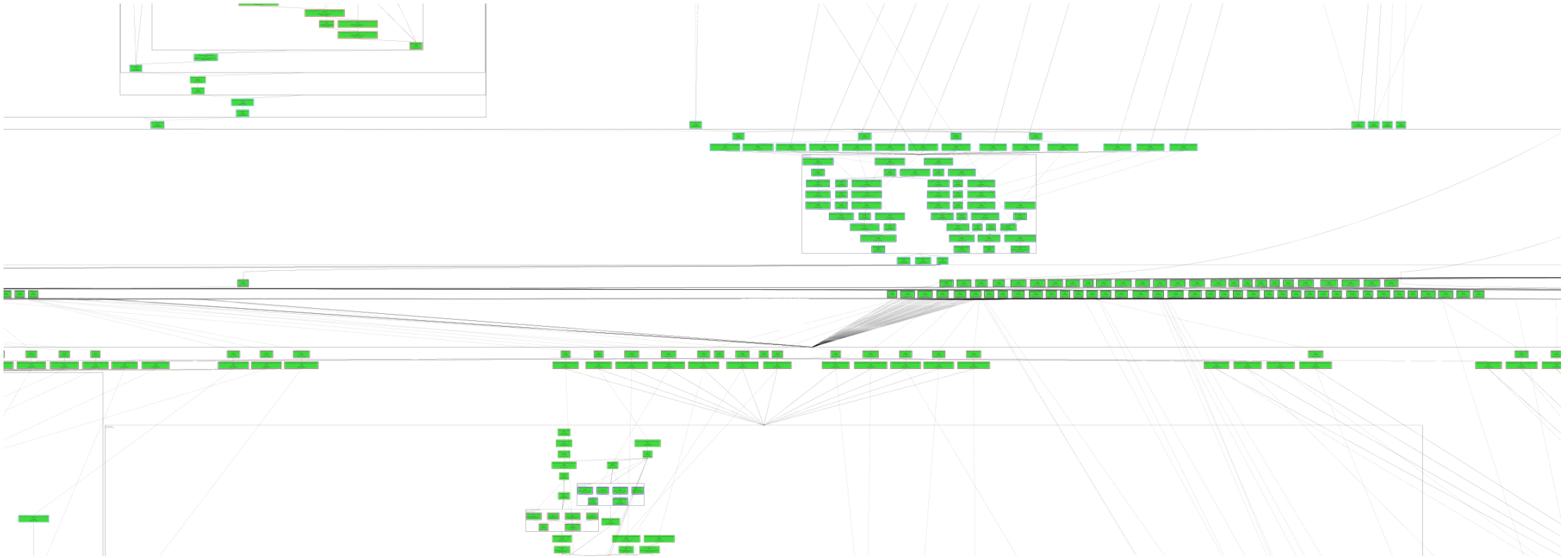


Processing Stages

```
1 // 10 groups of 10,000,000 vectorised Normal samples in total.
2 normalSamples[i][10] = Normal(2 + i,1.5);
3
4 // 10 groups of 10,000,000 vectorised Pareto samples in total.
5 paretoSamples[i][10] = Pareto(1 + i, 0.7);
6
7 // Simple vectorised sum over all samples.
8 sumSamples = normalSamples + paretoSamples;
```

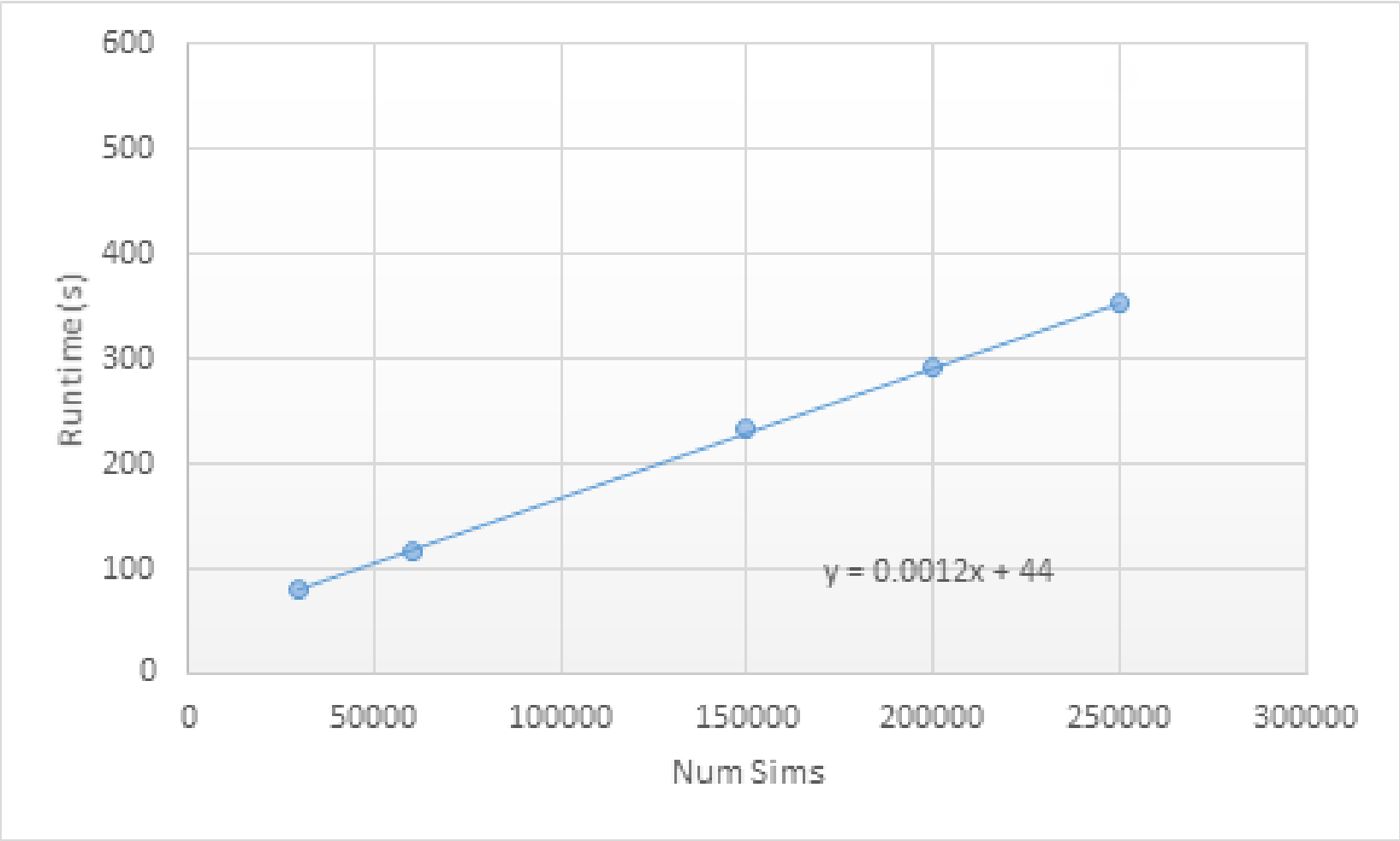


Internal Model Example



- This internal model originally took 3 years to build, and 6 hours to run on previous "state of the art" software.
- Recoding in Tyche took one month, and ran in 2 minutes.

Efficiency



Summary

- tyche™ is a SIMD Monte Carlo engine.
- Low-level SIMD languages and hardware are **abstracted away**.
- A **simple flowchart-based interface** is presented to the end-user, together with an **intuitive scripting language**.
- The **performance cost** for this layer of abstraction is **minimal**.
- The **reduction in model construction time** is **huge**.
- tyche™ is currently targeted at **computational finance**, but we anticipate more general uses in the future: the software is completely **usage-agnostic**, just like Excel.

Questions?

Legal Notices

This presentation is confidential and is intended for clients and professional contacts of Marriott Sinclair LLP.

No distribution, publication, copying or dissemination (in any form) is permitted without the prior written consent of an authorised signatory of Marriott Sinclair LLP.

The information and opinions contained in this presentation are for general information purposes only and do not constitute professional advice, and should not be relied on or treated as a substitute for specific advice relevant to particular circumstances.

Marriott Sinclair LLP does not accept or assume any liability, responsibility or duty of care for any loss which may arise from reliance on information or opinions published in this publication or for any decision based on it.

We would be pleased to discuss how the content of the presentation may tailored to your specific circumstances.