Mo.net

Advanced Computing Techniques with Mo.net

January 2024
Revision 10

Background

Ever since actuaries & risk managers got their hands on desktop computers in the mid-1990s, they've had an almost continuous thirst for more power & performance to satisfy their increasingly demanding modelling needs. Fortunately, and until recently, Moore's Law – the doubling of processing power for about the same cost, as observed by Gordon Moore, one of Intel's founders in 1965 - has largely kept pace with these requirements.

While there has been a legitimate regulatory demand for more frequent modelling, with more scenarios and more data, to be performed in less time as reporting schedules are accelerated, some modelling requirements have exceeded the limits of available computing hardware. This has led many insurers into the realm of parallel processing & high-performance computing, either using on-premise or hosted grid computing clusters, or more recently similar technology running in the cloud. The basis of the approach is simply to use hundreds or thousands of separate single or multi-core CPUs running on multiple computers to solve a single problem / task by splitting the work up into manageable units, usually a policy, model point or scenario. This is the basis for Mo.net's own master / worker distribution functionality.

While HPC or cloud-based parallelism is a tried-and-tested, cost-effective solution to the evolving modelling needs of insurance actuaries, it is no longer the only solution on offer. This paper considers an alternative approach to traditional workload distribution and reviews its appropriateness for the life modelling requirements of today and tomorrow.

Flynn's Taxonomy

All forms of parallel computing inhabit a quadrant in Flynn's taxonomy, as proposed by the notable computer scientist Michael J. Flynn in 1966.

- SISD – single instruction, single data unit
- MISD – multiple instructions, single data unit

- SIMD – single instruction, multiple data units
- MIMD – multiple instructions, multiple data units

Traditional financial modelling HPC or workload distribution techniques occupy the MIMD quadrant since while the instructions may be almost the same, they are not identical due to the instruction path itself being dependent on the data unit. The instructions can therefore be considered as independent / different for the purposes of this discussion.

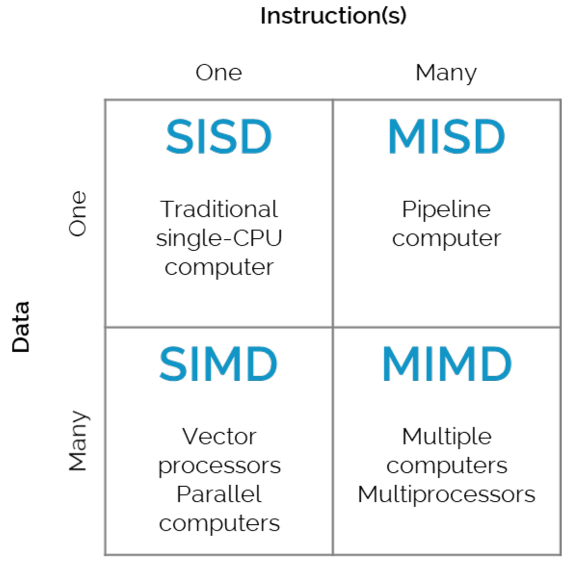

Figure 1 - Flynn's Taxonomy

Advances in Parallel Computing Approaches

Over the last decade there have been significant advances in the design and architecture of computers and CPUs. In part this has been fuelled by the growth of the games console and mobile phone industries, which have driven the need for specific processing units aligned to very specific workloads. Of particular interest here are the advances made in vector processing in the games industry, where the fast-moving graphics have demanded dedicated chips – so called GPUs (graphical processing units) – which use a variation of SIMD called SIMT (single instruction, multiple threads).

Unlocking the potential power of GPUs requires specially written logic for relatively specific problems. Attempting to apply similar approaches to financial modelling would require specific GPU-aware models to be developed and data (policy, market, assumptions, etc) to be structured in a form suitable for processing by the GPU logic. The significant knowledge of GPU architecture and the low-level programming skill required to accomplish this task coupled has so far presented a barrier too high for any but academics to challenge.

What's Our Vector, Victor?

An alternative to using GPUs is to utilise the vector-processing potential added to recent generations of multi-core CPUs fitted to consumer laptops and workstations.

Vector processing, which is a form of SIMD, simply allows the same instruction to be applied simultaneously to a number of data units held in memory. The number of data units to which the vector instruction is applied is limited only by available memory. The vector instruction is necessarily simple, so complex operations will need to be broken into a number of distinct steps.

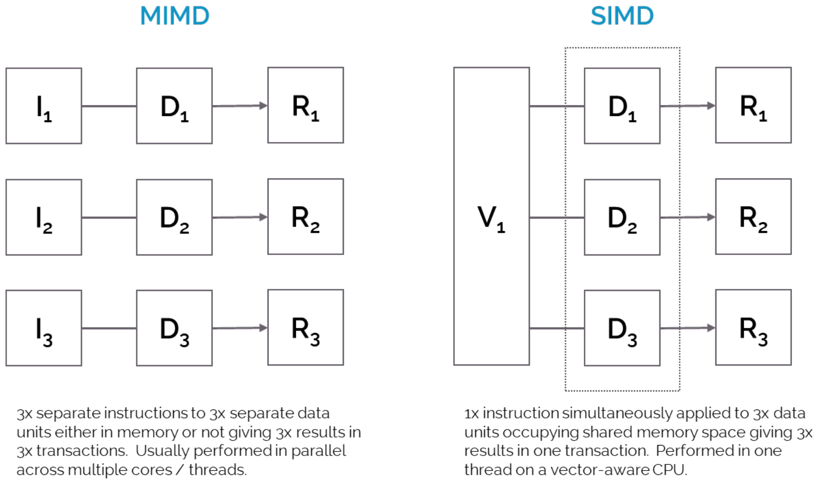

Figure 2 - Conceptual view of MIMD vs SIMD

Clearly, there is the potential for significant performance improvement if operations can be encapsulated into one or more vector instructions. The data units can be stored together in memory, and the problem is presented to a suitably equipped CPU. The exact performance improvement will depend on the number of operations to be performed, but simplistic problems might expect to see order of magnitude improvements compared to traditional methods.

Suitability for Life Modelling Applications

Obviously, any potential performance improvement of this order is of interest to the actuarial and financial modelling community, particularly in the life modelling universe where the complexities of products / models consume vast amounts of traditional computing resource.

However, it is the features of the life modelling universe that present a number of potential challenges when it comes to adopting a SIMD / vector processing solution.

- **Data**
  Life datasets are large – both in terms of width (number of factors) and depth (number of policies). The memory requirements of any environment used for SIMD life modelling would be significant and costly. Even with the most exotic hardware, datasets would most likely require splitting into groups and processing separately.
- **Instructions / Models**
  Life insurance modelling is tricky at best to break into small, distinct operations appropriate for vector processing. Even if traditional modelling functionality could be converted into a number of sequential vectorised steps, this would require wholesale model refactoring. While this itself isn't necessarily a showstopper, the effort would require a significant upside benefit.
- **Hardware / Instructure**
  SIMD / vector-ready CPUs are becoming commonplace in laptops and desktops where applications have been developed to take advantage of this capability, but server-based hardware with appropriate features are still relatively costly.

Implementing SIMD / Vector Solutions with Mo.net

To understand the potential of SIMD / vector processing solutions for life insurance modelling, and to demonstrate some of the implementation challenges, we have harnessed the integration potential and flexibility of the Mo.net platform to develop some simplistic models that use both traditional SISD / MIMD and SIMD approaches to solve the same problem. Our SIMD demonstration project uses an external SIMD / vector mathematics library now provided as part of Microsoft's .NET Framework. This allows us to easily replace traditional VB.NET operations with vector-based equivalents, rather than refactoring the calculation logic from scratch.

When running on appropriately equipped hardware, the performance gains of SIMD over SISD are obvious, with SIMD tasks performing almost an order of magnitude faster. Comparison of SIMD vs MIMD is, however, more subjective, since distribution of workload over a number of cores or CPUs, while perhaps not elegant can scale well beyond the benefits provided by SIMD and without any refactoring to use vector processing libraries.

Example Mo.net Project

The SIMD / vector processing project used for our research will be available as a sample project with Mo.net 7.5. Customers are encouraged to experiment with this project and understand the potential benefits and challenges for themselves.

Alternative Solutions

For now, we feel that SIMD / vector processing has limited application in the life modelling universe. While the potential performance benefits can be shown using relatively simplistic applications, the work to extend these to real-world modelling projects is significant, and even if the refactoring could be achieved there are some significant infrastructure issues that would need to be resolved.

Customers should perhaps also ask themselves "how fast is fast enough?". With easy access to flexible, powerful, and cost-effective compute resource either in hosted or cloud-based environments, it feels like traditional MIMD-type approaches still have the upper hand for now, both in terms of cost and simplicity.

As and when a killer life modelling application emerges that requires SIMD / vectoring processing, Mo.net will be ready to receive.

Contact Us

For more information regarding the Mo.net platform or the SIMD / vector processing sample application, please get in touch:

Software Alliance Limited
30 Stamford Street, London, SE1 9LQ
Tel: +44 (0) 20 3964 2755
www.softwarealliance.net

Author: Guy Shepherd