Milliman

Application of GPU in Actuarial Modeling

Chihong An
Managing Director of Milliman Korea

July, 2018

Agenda

1. Technology Trend in Actuarial Modeling (Korea)
2. Introduction to GPU
3. Limitations of GPU and Solutions
4. Case Study and Implications

Technology Trend in Actuarial Modeling (Korea)

Challenges of IFRS 17

Adapting to IFRS 17 in Korea implies a groundbreaking change to the traditional practice of financial reporting which would also significantly increase computing power requirements.

Item	Changes	Increase in computing requirements
Reserve Principle	Net Level Premium → Gross Premium	x50
Unique Product Characteristics	Complex Products with many interdependent benefits, policy-holder options and rider choices	Requires Seriatim Projection
Model Point	Clustered/Grouped (1%) → Seriatim (100%)	x100
Scenarios	Deterministic (1 Scenario BE or Worst Case) → Stochastic (average of 1000 scenarios)	x1000
Movement Analysis	Need to isolate the impact of one change from various changes (1 Run → 10 Runs)	x10
Total	**Significant increase in computing power requirements**	**x50,000,000**

Taxonomy of model efficiency

There are multiple ways of coping with the issue of increased computing requirements.

Actuarial and Modeling Techniques

- Scenario Design and Selection
- Mathematical and/or Model Design
- Model Data Building Techniques

Proxy Conceptual Model Design:
Some impact on model results

Technology Solutions

- Hardware Design
- Software Design

Model Implementation:
No impact on model results

Source: 2017 SOA Life and Annuity Symposium
Session 57 PD: Model Efficiency – Part 1

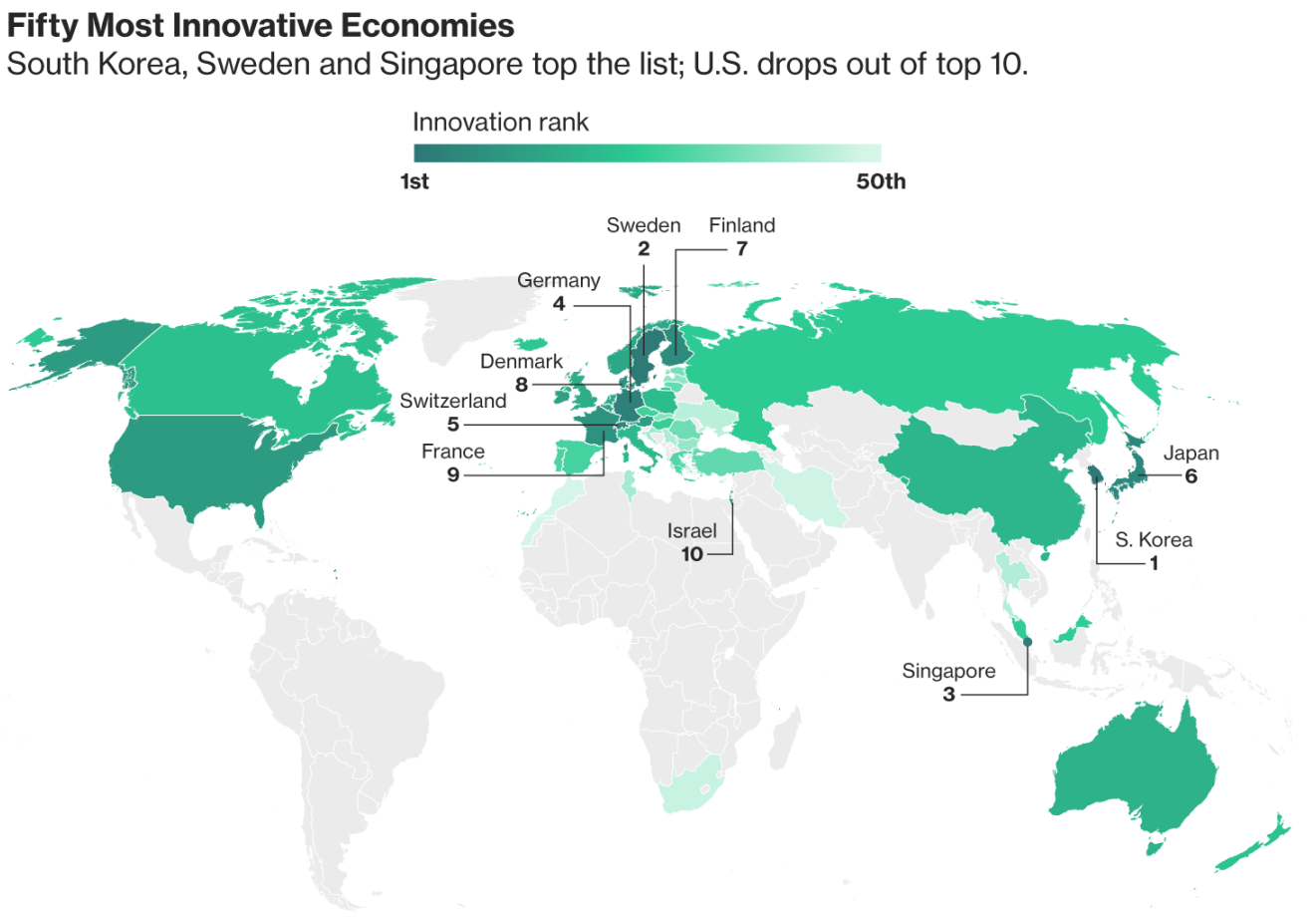
Modeling trends in Korea

Korean insurers are focusing more on technological solutions – especially GPU-computing.

Rank	Current SW	New SW (IFRS 17)	Use of GPU
1	Existing System	In-House	
2	Existing System	Existing System	
3	Existing System	New System (1)	**GPU**
4	Existing System	New System (2)	**GPU / CPU**
5	Existing System	New System (2)	**GPU / CPU**
6	Existing System	New System (2)	**GPU / CPU**
…	…	…	**…**
5 mid-small companies	Existing System	Industry Consortium	**GPU**

Why is Korea taking a different approach?

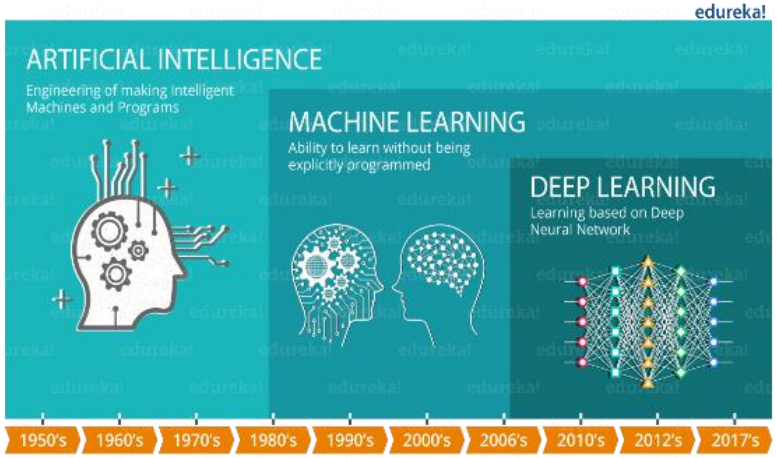
South Korea is the most innovative country in the world!



Introduction to the GPU

What is behind the recent IT innovations?

It is becoming increasingly common to apply GPU in areas which require computationally intensive calculations.



Artificial Intelligence

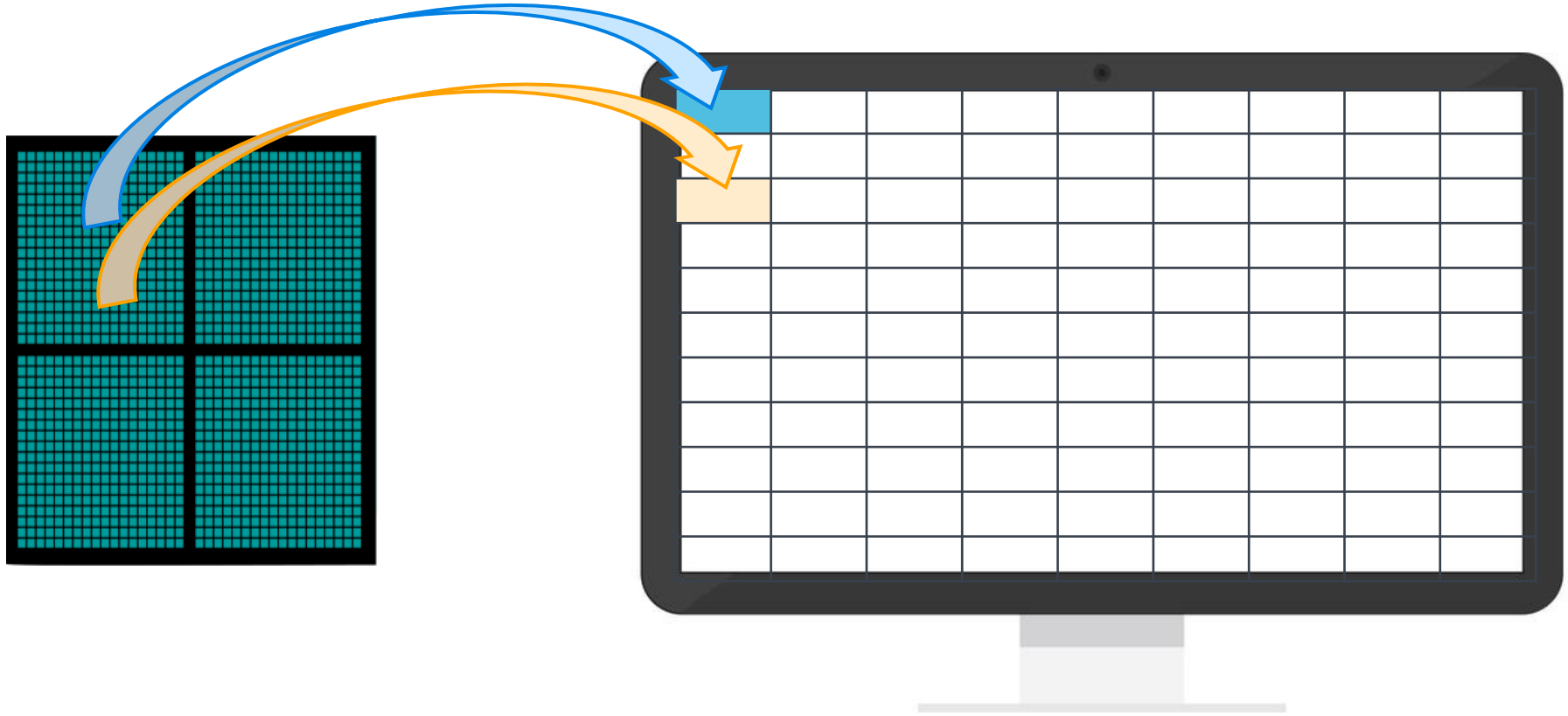
Big Data

Bitcoin / Blockchain

GPU (Graphic Processing Unit)

What is the GPU?

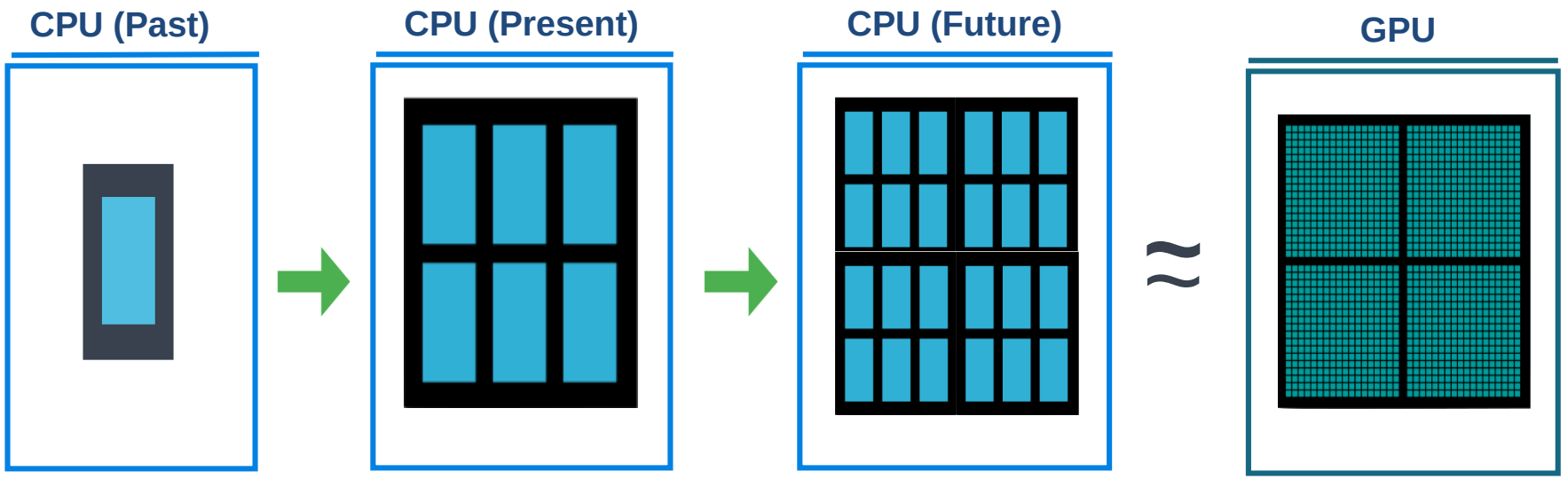
The Graphics Processing Unit (GPU) is a type of electronic circuit which is developed to control the pixels of display devices (i.e. rendering images to your screen) – so it's designed to have many processing units inside the chip.



What makes the GPU so powerful?

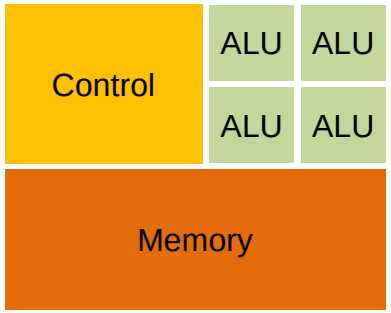
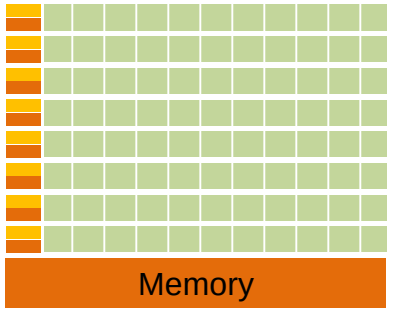
GPU's unique structure with many processing units make it superior for certain calculations.

- The capacity of a single-core has reached its limit for improving (the end of Moore's Law: the number of transistors in a dense, integrated circuit doubles about every two years)
- CPU developers are instead increasing the number of cores in a processor instead of increasing the transistors in a core - introducing multi-core technology



GPU vs CPU

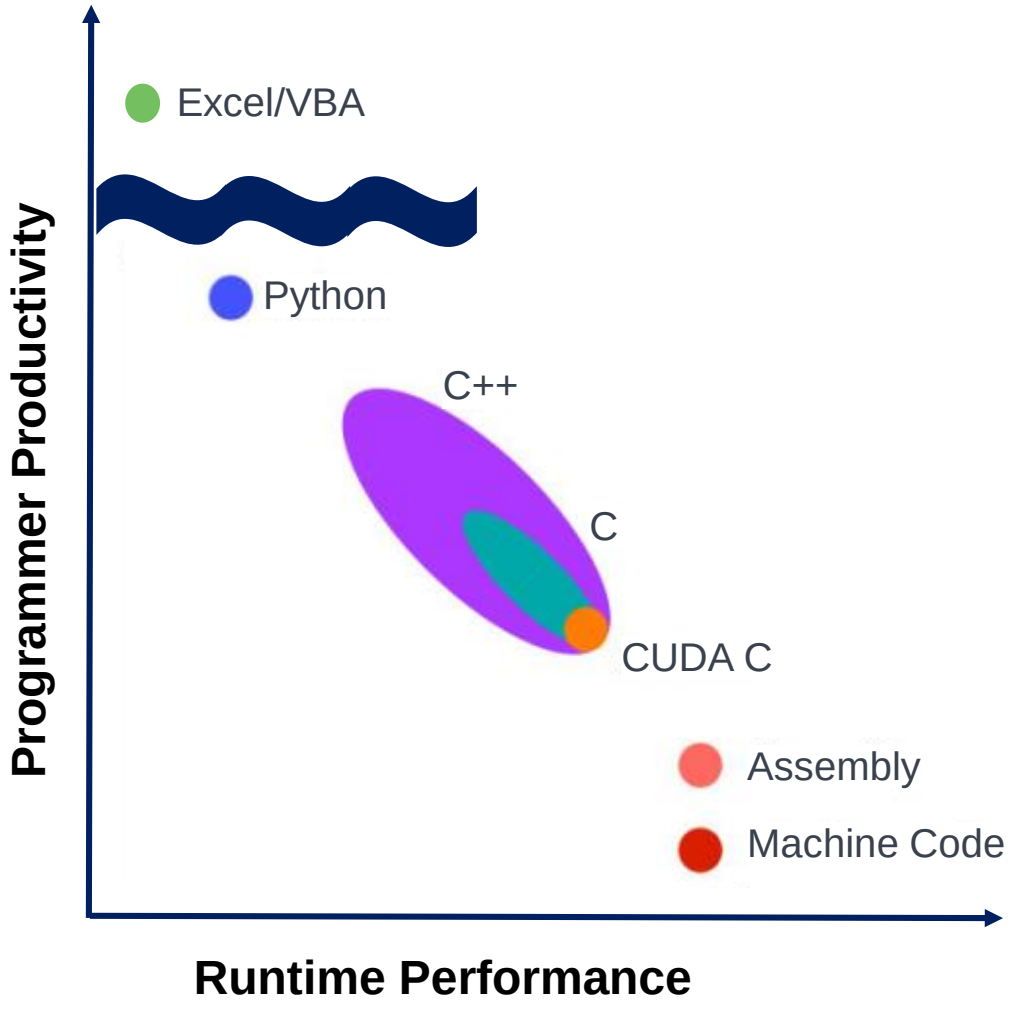
Both chips have a very different structure and hence have distinct pros and cons.

	CPU	GPU
Structure	• A few number (4-16) of high-performing cores 	• Many numbers (5000+) of low-performing ALUs 
Strength	• Serial calculations (small number of complex order-dependent calc.) • Relatively easy to program • Relatively large size of memory per cores (efficient to handle large inputs)	• Parallel calculations (many numbers of short independent calculations – TVOG/GMxB) • Less Expensive (Best per-Dollar Performance)
Weakness	• Relatively expensive to acquire thousands of cores	• Relatively difficult to use (because not developed for general purpose) • Relatively small size of memory per cores (not efficient to handle large input data)

Limitations of the GPU and Solutions

1. High Programming Difficulty

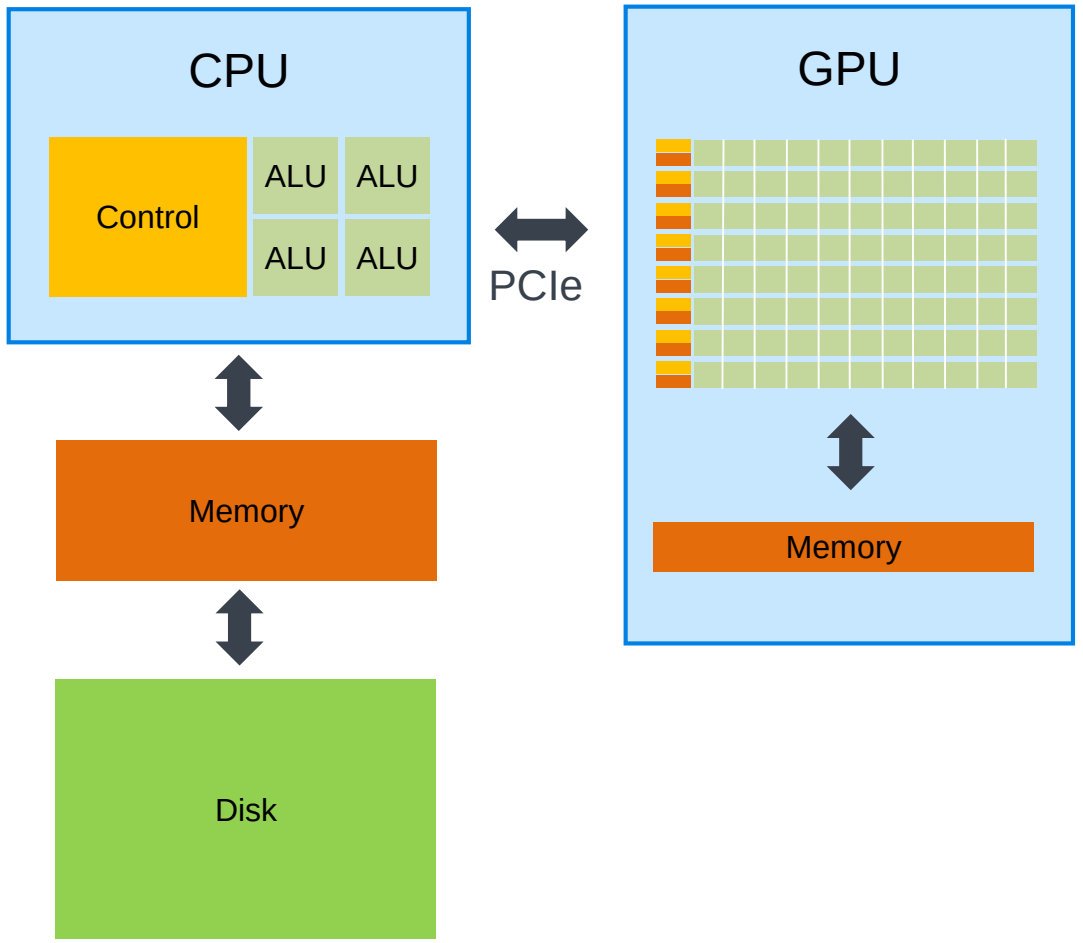
- Impractical for actuaries to learn CUDA C for daily modeling tasks. Both C and CUDA C are difficult languages for even veteran programmers.
- **Solution**: A GPU-based system should provide users (actuaries) with a simple language (like VBA) for model development but should be translated into CUDA C when code is executed.



2. Memory Issue - Limited Size and Slow Data Transfer

- Since GPU's memory has to be shared by 5000+ cores during parallel processing, the size of memory potentially allocated to each core is very limited.

- If the calculation of each core requires more computing power than they are allocated, GPU cores would have to take data I/O to CPU's memory through PCI-Express channel which is going to be a lot slower than GPU's internal memory I/O.



2. Memory Issue – Affects the Level of Parallelization

- In this respect, GPU is most efficient to process one model point's multi-(inner) scenarios in parallel. However, the level of parallelization is limited, also limiting the performance enhancement through the parallelization.

- To increase the level parallelization, the information for all or more model-points (for ALM's) would have to be held in memory. However, this would fall exceed the size of GPU's memory, causing extra data transfers.

Logic Flow for Typical Actuarial Projection

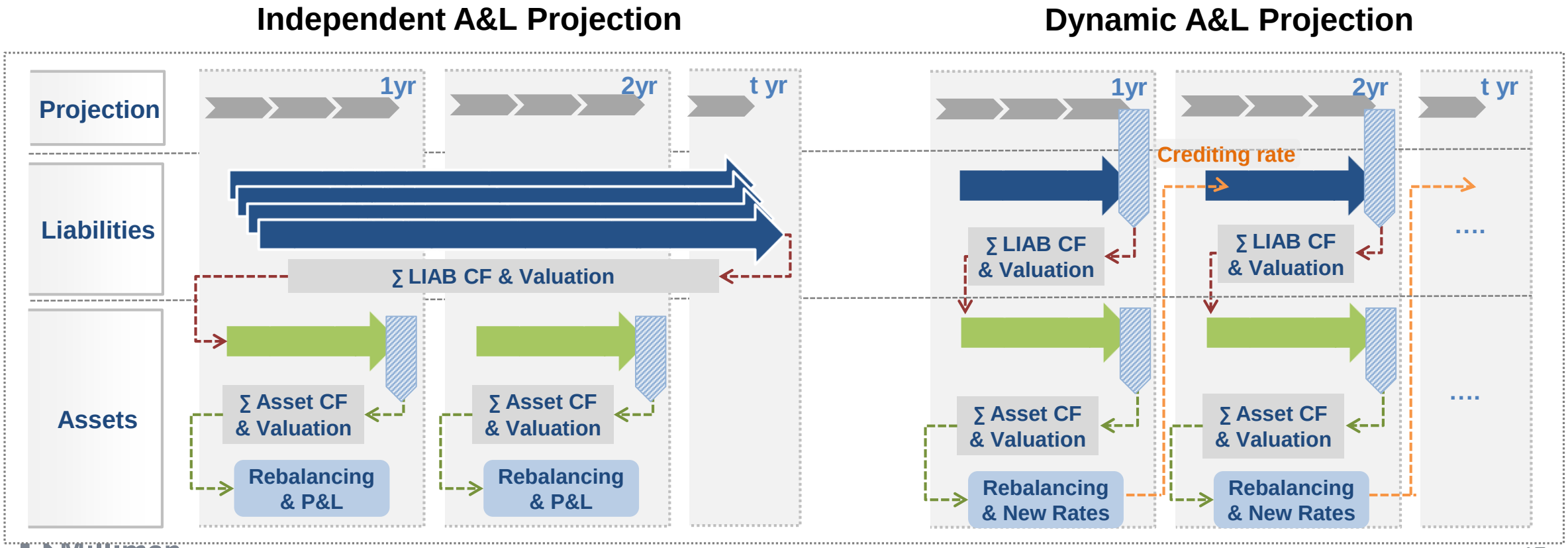
```
Public Sub Main()
    Call Import_Global_Inputs
    For Outer_Scenario(ALM) = 1 to 1000
        For Policy = 1 To N
            Call Import_ModelPoint_And_Inputs
            For Inner_Scenario (Val'n) = 1 To 1000
                Call Projection (Calculation)
            Next Inner_Scenario
        Next Policy
    Next Outer_Scenario
    Call Export_Results
End Sub
```

(Inner) Scen Loop has a limited parallelization but requires only small memory during calculation

Outer Scen Loop has a higher level of parallelization but requires much more memory

2. Memory Issue (Example) - Dynamic ALM Projection

- Dynamic ALM's period-by-period projections (unlike independent liability run's record-by-record projections) requires all model points' interim variables to be held in memory during throughout the whole projection. This requirement often incurs memory overflow in CPU-based computing and will incur even bigger problem with GPU.



3. Not Efficient with Complex Sequential Logics

- Complex actuarial logic may significantly slow down the performance of GPU.
- Due to the native structure with massive cores, all GPU cores have to process the same command at the same time and cannot process different logic like CPU cores.
- This does not fit well with typical actuarial calculation logic, having many “IF” and “ELSE IF” statements

1. Simple Case (CPU vs GPU)

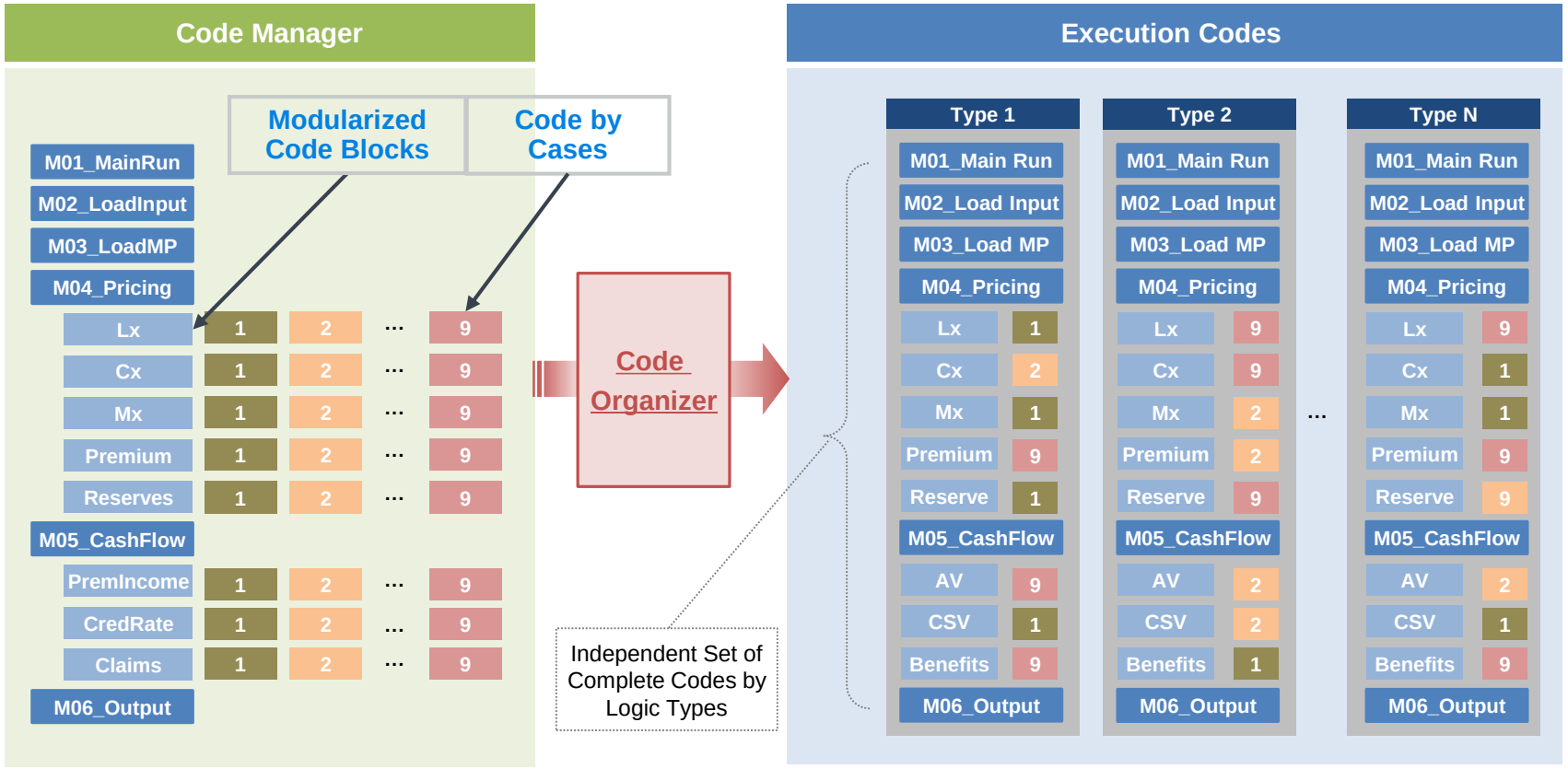
	CPU 1	…	CPU 4	GPU1	…	GPU1000
Do 1	Process		Process	Process		Process
Number of total processes	**1**	**1**	**1**	**1**	**1**	**1**

2. Typical Actuarial Calculation (Complex Condition Checks)

				CPU 1	…	CPU 4	GPU1	…	GPU1000
				A=TRUE		**A=FALSE**	**A=TRUE**		**A=FALSE**
				B=TRUE		**C=FALSE**	**B=TRUE**		**C=FALSE**
IF Condition A = True				Process		Process	Process		Process
	THEN	IF Condition B = TRUE		Process			Process		Wait
			THEN Do 1	Process			Process		Wait
			ELSE THEN Do 2				Wait		Wait
	ELSE THEN	IF Condition C = TRUE				Process	Wait		Process
			THEN Do 3				Wait		Wait
			ELSE THEN Do 4			Process	Wait		Process
Number of total processes				**3**	**3**	**3**	**7**	**7**	**7**

3. Not efficient with Complex Sequential Logics

- **Solution**: GPU-based solution should provide advanced modular code management features which can populate many sets of simple/efficient code (with less conditions) instead of a single set of complex code (with a lot of conditions)

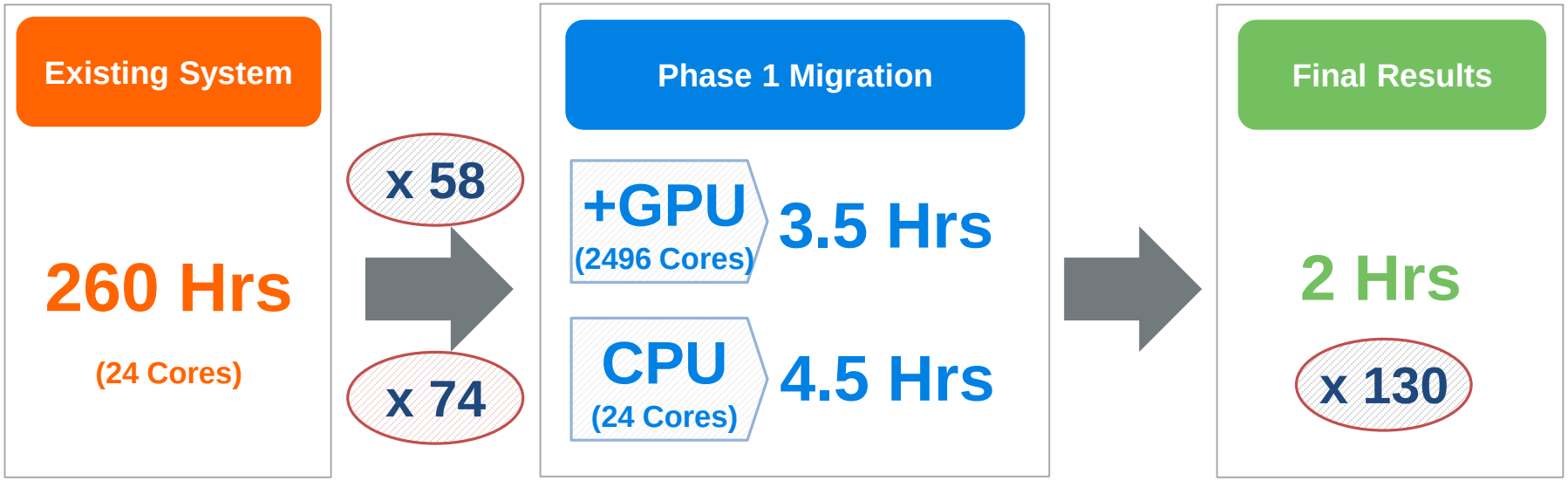


Case Study and Implications

Actual Implementation Results (GMxB Reserving)

GPU-accelerated computing can significantly boost up the speed of actuarial calculations but most of them are realized by improving the efficiency of the model logic itself (optimizing data I/O, calculation and etc).

- Considering the scalability of the hardware, GPU would still have best per-dollar performance.



Implication for the future of actuarial modeling

Actuaries need to keep on eye on various current IT technologies which are rapidly evolving

Hardware (CPU vs GPU)

- Both have very distinct pros and cons
  – the key is to use where appropriate
- Both will continue to challenge each other
  - GPU: larger memory and faster data I/O.
  - CPU: more cores

Modeling Technology

- Eventually move from proxy modeling techniques to full modeling approaches based on evolving SW/HW technologies.
- Seriatim nested-stochastic with dynamic ALM interaction modeling will be the norm someday.

Milliman

Thank you

Chihong An

Chihong.An@Milliman.com

Managing Director – Seoul

+82-10 3789 2193