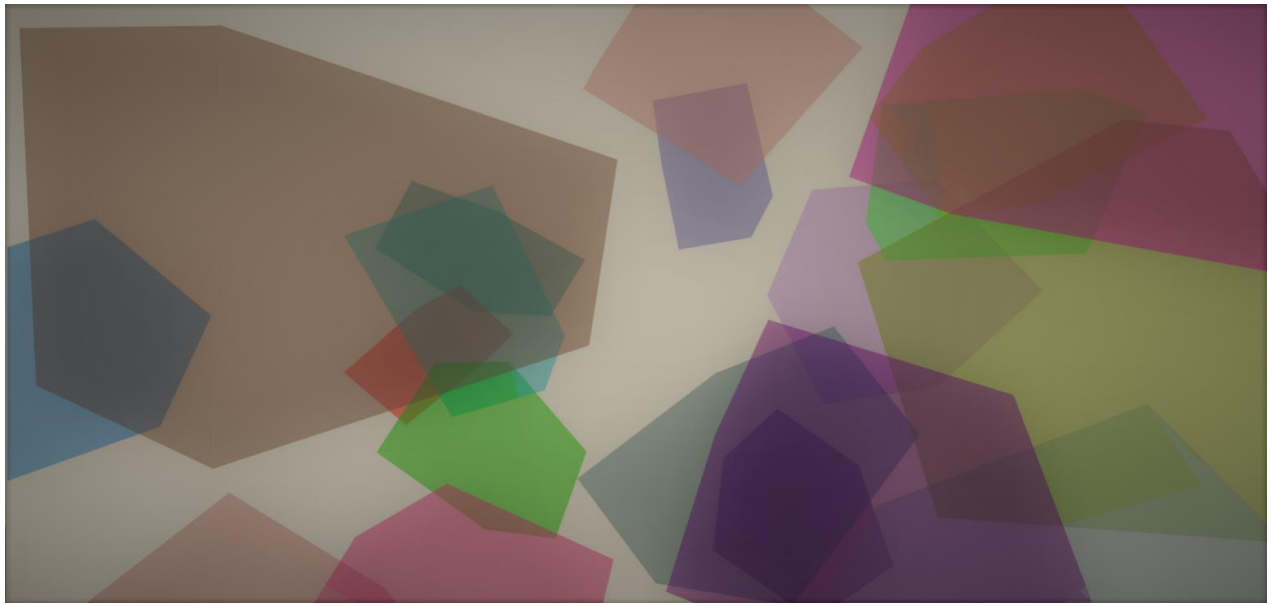


Graphical processor unit optimization of CreditMetrics model

Dr. Anatoliy Antonov

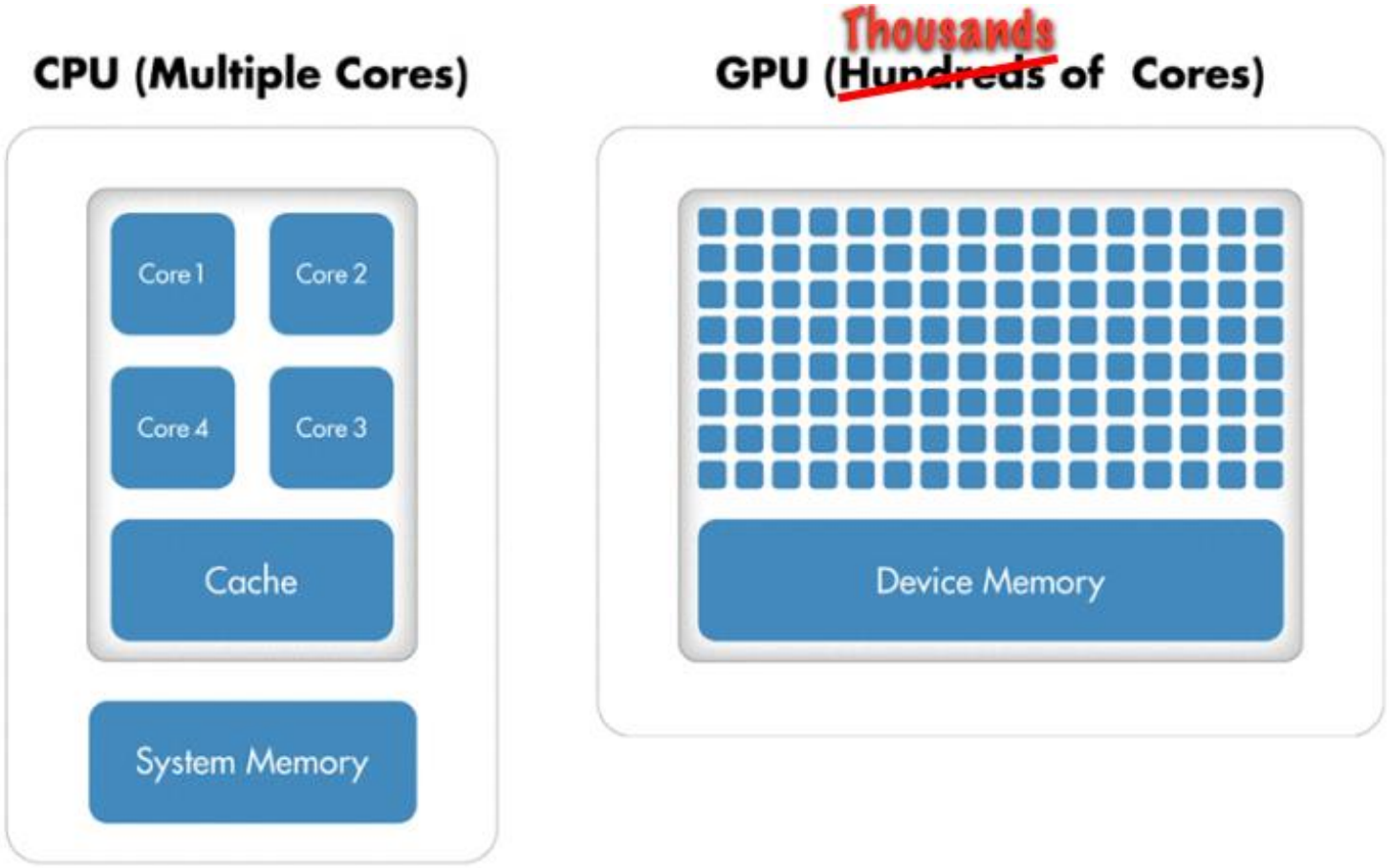
Nikola Vasilev

Agenda

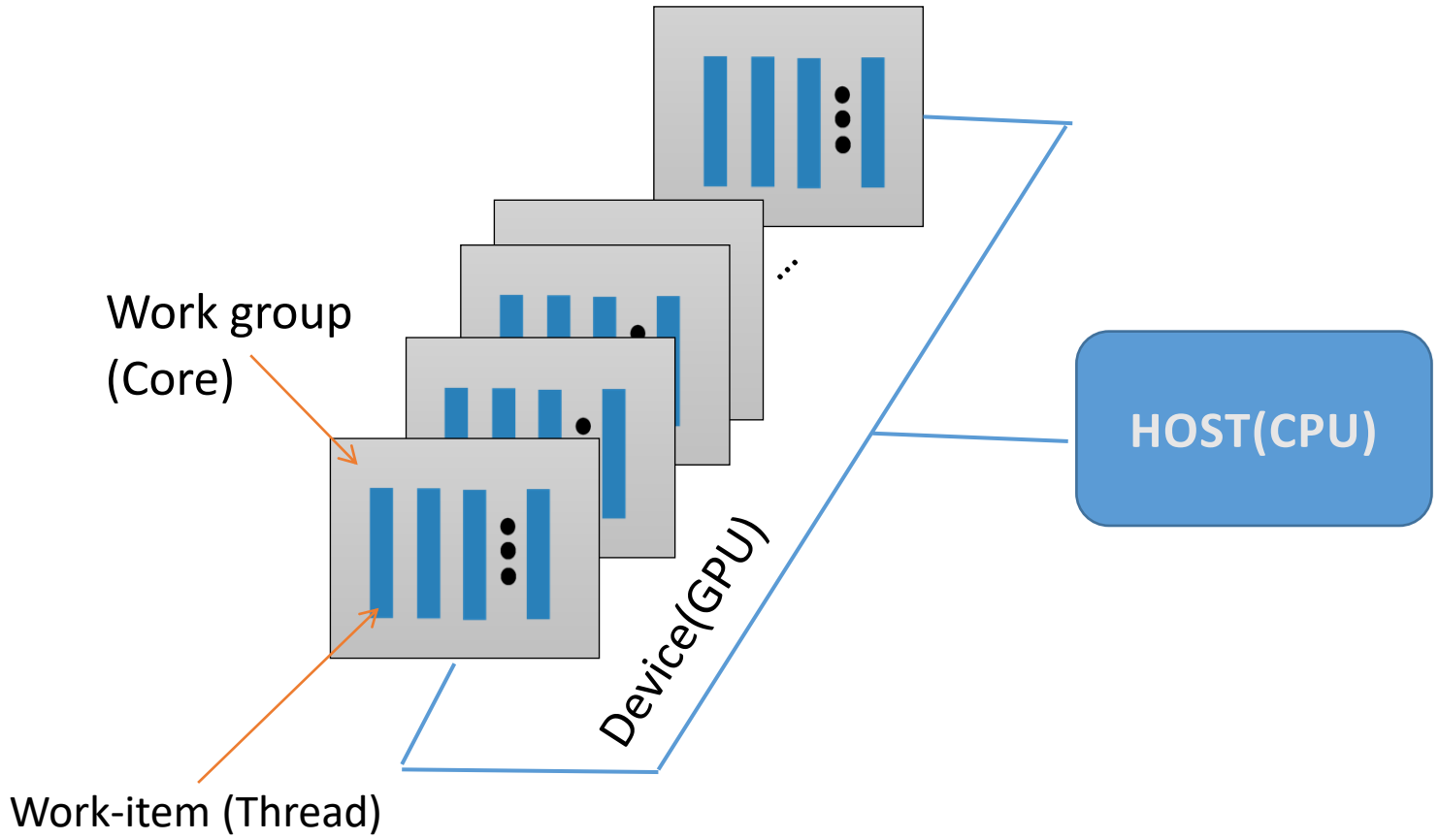
- What is GPU?
- What is OpenCL?
- MonteCarlo CreditMetrics Model
- Approach to integration of GPU computation
- Algorithm design
- Time benchmarks

Graphical Processor Units(GPU)

- Single-chip processor with integrated transform, lighting, triangle setup/clipping, and rendering engines
- faster pricing gains more revenue
- more modeling gains less risk and maximizing resources gives more efficiency
- architecture allows to run more simulations that would increase the quality of your results

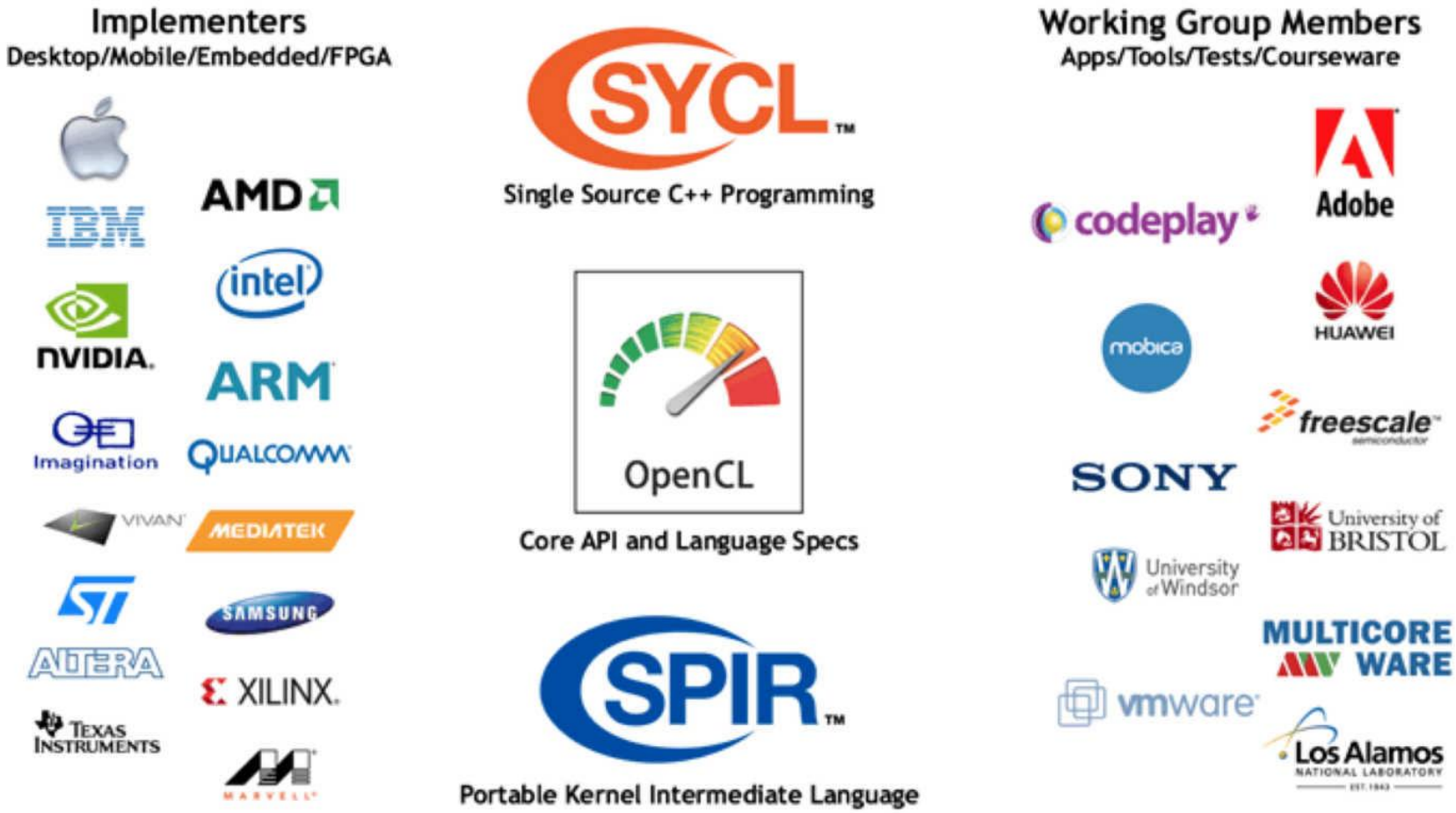


- At the top-level, a PCIe graphics card with a many-core GPU and high-speed graphics “device” memory sits inside a standard PC/server with one or two multicore CPUs
- GPU is designed to calculate massive parallelism
- Suitable for simulation models



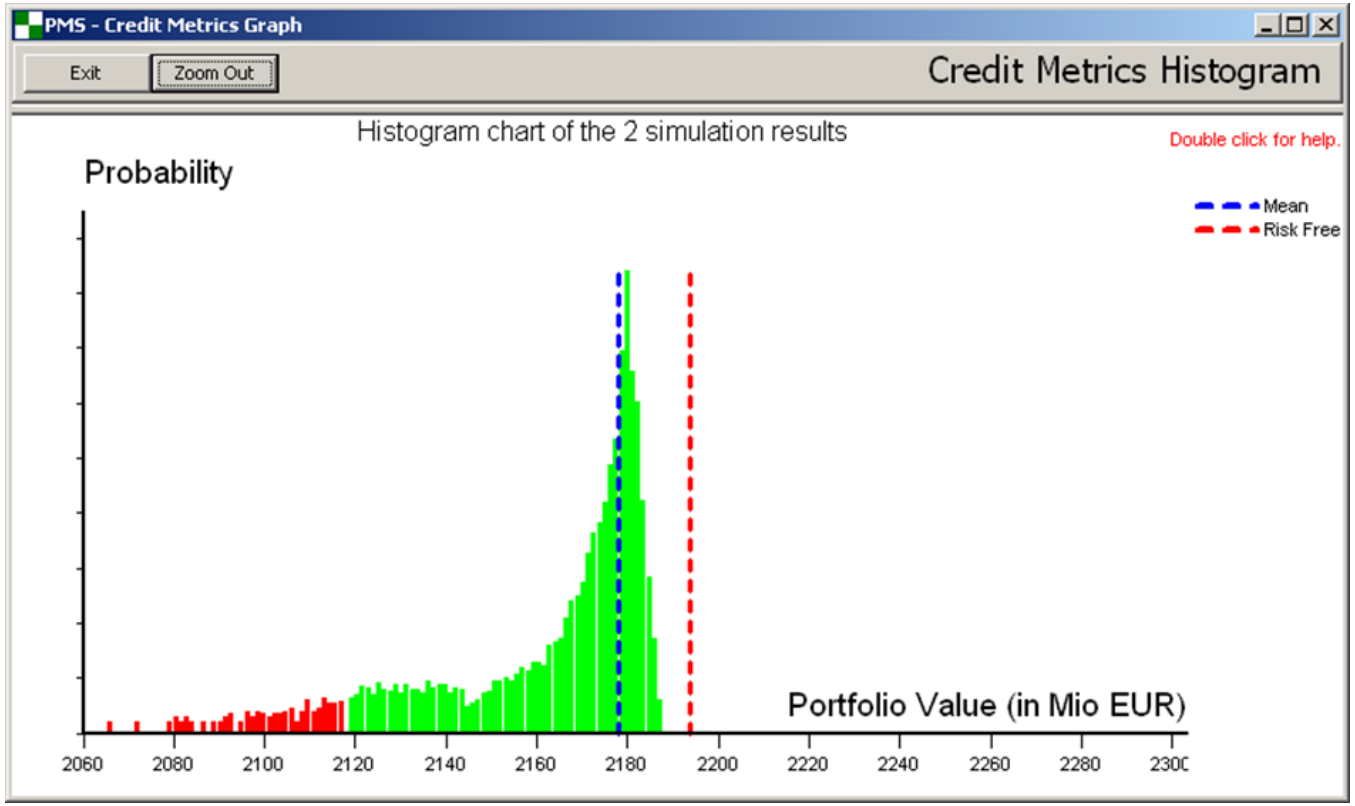
OpenCL™

- Open Computing language
- royalty-free standard for cross-platform, parallel programming of diverse processors
- users can launch computation on high-performance devices – mostly Graphical Processor Units(GPU)



CreditMetrics Analysis Model

- Quantifies the loss caused by a change in the obligor's creditworthiness
- Main goals:
  - Computation of credit risk exposure
  - Showing the notional amount that could be lost
- The metrics suite is designed to measure different aspects of exposure, both now and in the future



Portfolio histogram and distribution

What is Monte Carlo?

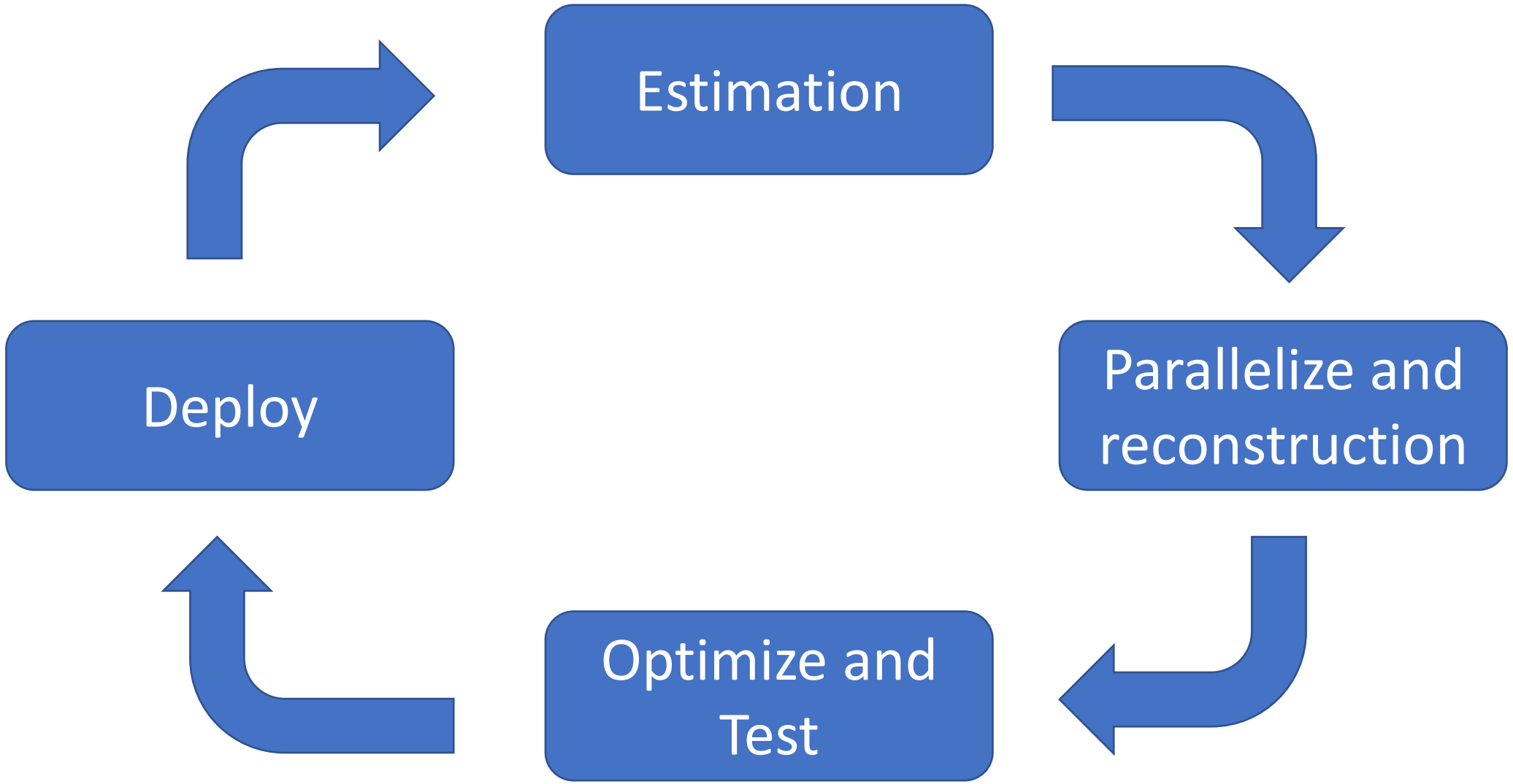
- Monte Carlo is a nimerical procees based on repeated random sampling to obtain a numerical result
- The samples represent realizations of random variableas and are used to create possible outcomes
- Implementing a Monte Carlo simulation in CreditMetrics generaties distibution of portfolio values:
  - Estimation of correlation between obligors asset returns
  - Asset return scenarios
  - Value at Risk(VaR)
  - Credit exposure

CreditMetrics calculation stages

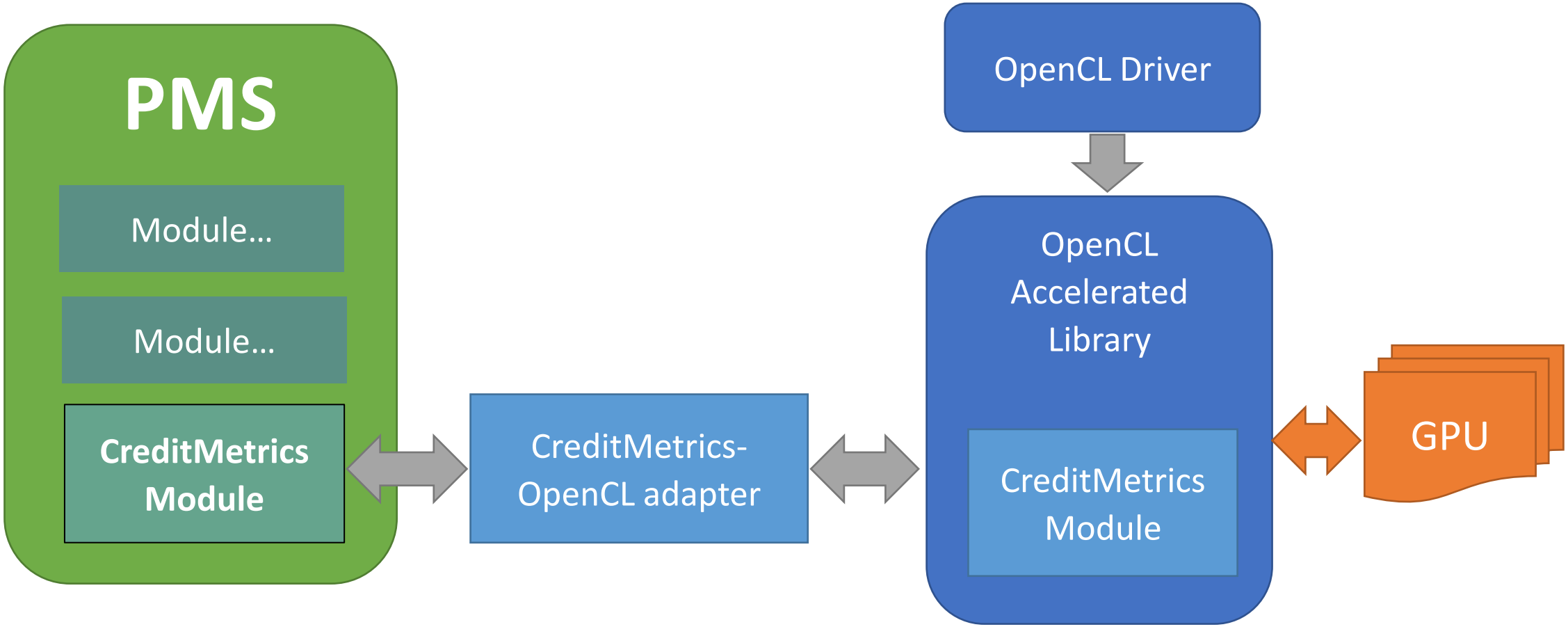
1. Portfolio Load
2. Portfolio Calculation
3. Start CreditMetrics, Exposure calculation
4. **Generation and correction of correlated simultanious numbers**
5. **MonteCarlo simulation produces credit VaR distribution**
6. **Evaluation of the results and statistics on the credit VaR distribution**

GPU accelerated for large number of runs

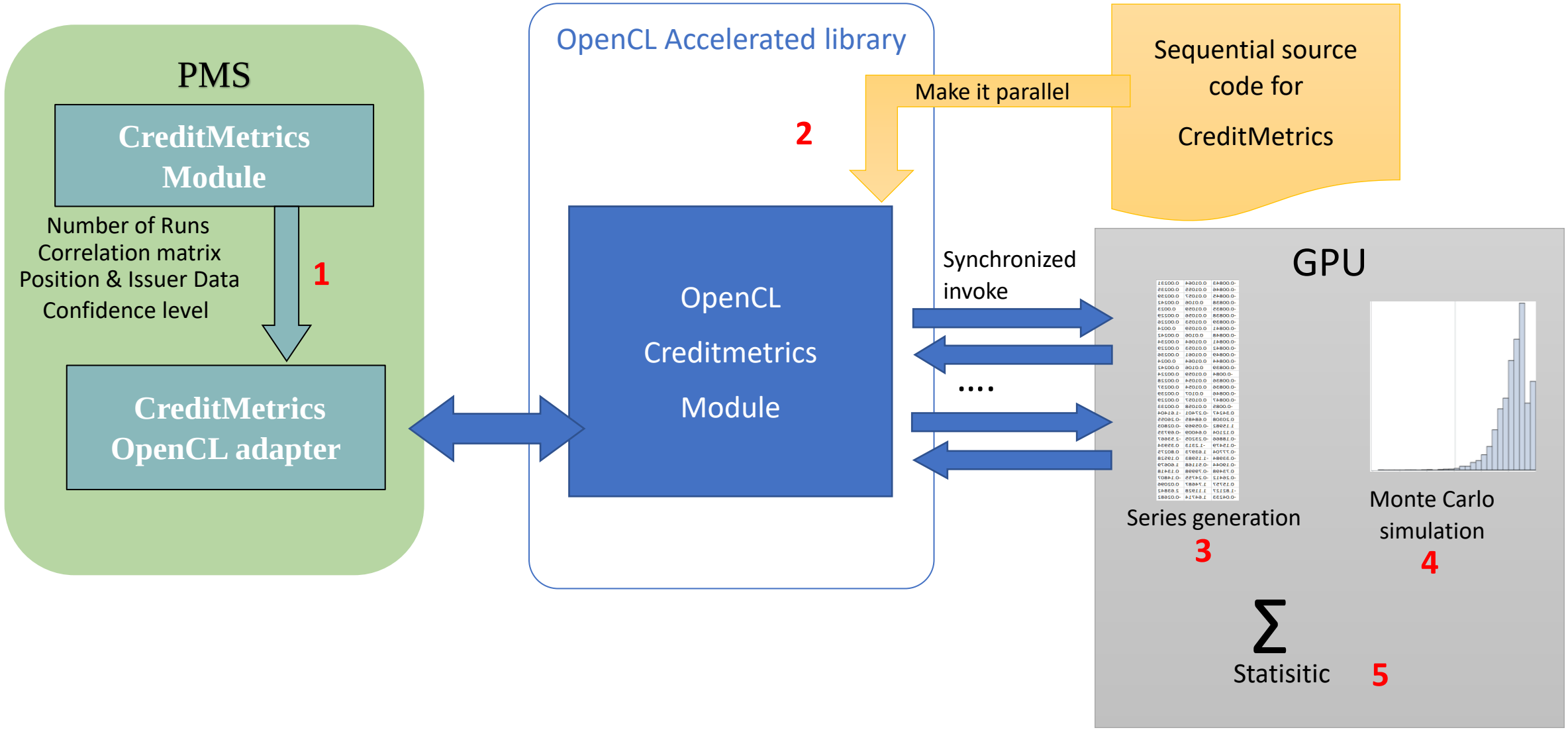
Basic steps of GPU acceleration project



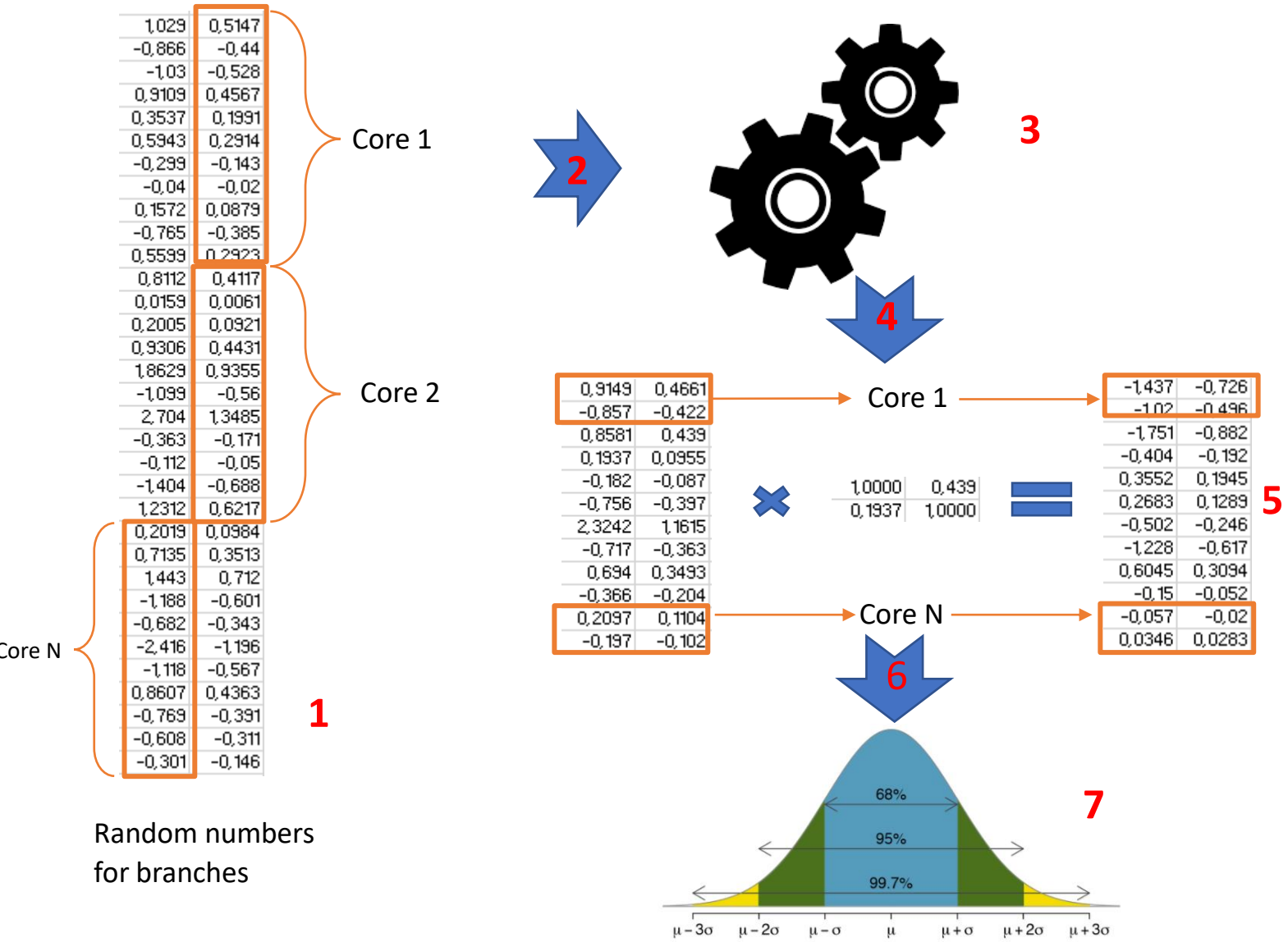
OpenCL™ approach in PMS



CreditMetrics GPU usage scheme

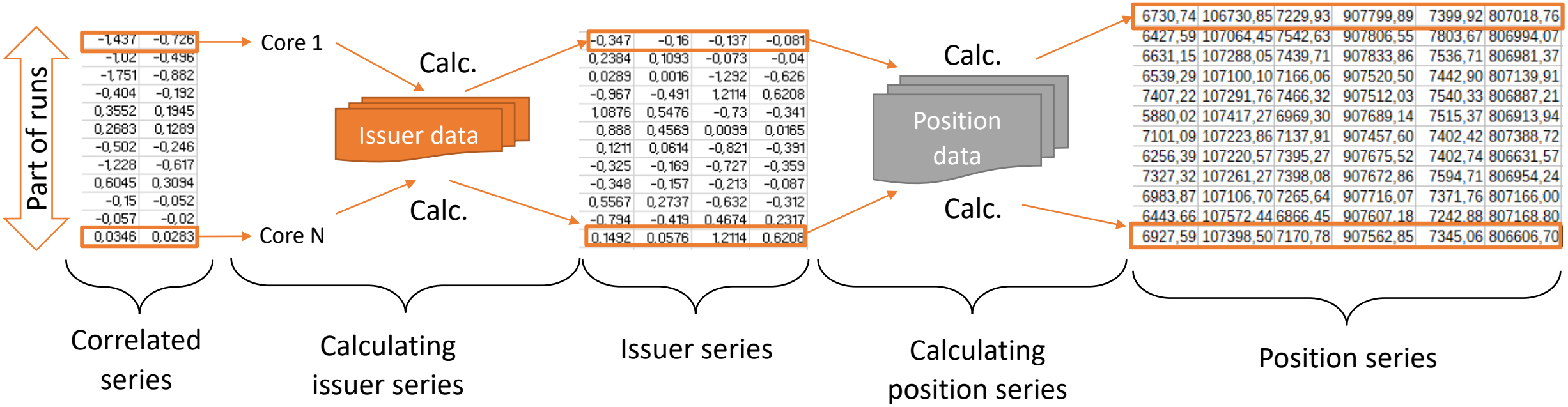


Parallel generation and correction of randoms on GPU



1. Each core generates several random distributed numbers successively
2. Synchronize all cores
3. Zero-correlation of initial randoms
4. Syncronize all cores
5. Apply branch index correlations – each core will multiply several rows of series matrix
6. Syncronize all cores
7. Correcrtion for normal distribution ND(0,1). The same approach to distribute the work on the GPU cores as for 1. is used.

Monte Carlo calculation stage on GPU



Monte carlo calculation stage is based on 2 parts: calculate volatility series per issuer and calculate monte carlo position series. Parallization on level run – each core calculates one run (all position values).

The calcutation proces is repeated until all runs are calculated - this builds the credit VaR distribution.

The satege is designed on separate parallel steps to minimize memory usage.

Distribution statistic stage on GPU (example for standard deviation)

Calculated distribution statistics:

1. Average
2. Min
3. Max
4. Standart Deviation
5. Kurtosis
6. Skewness
7. Value at Risk

$$\sqrt{\frac{\sum_{i=0}^{n}(x_i - \bar{x})^2}{(n-1)}}$$

Standard deviation formula

$$\sum_{i=0}^{n}(x_i - \bar{x})^2$$

GPU cores iterate on parts of the difference sequences

Sub-sum

$$\sum_{i=core\ index}^{runs\ per\ core}(x_i - \bar{x})^2$$

$$\sum_{i=core\ index}^{runs\ per\ core}(x_i - \bar{x})^2 \quad \bullet\bullet\bullet \quad \sum_{i=core\ index}^{runs\ per\ core}(x_i - \bar{x})^2$$

0,9149	0,4661
-0,857	-0,422
0,8581	0,439
0,1937	0,0955
-0,182	-0,087
-0,756	-0,397
2,3242	1,1615
-0,717	-0,363
0,694	0,3493
-0,366	-0,204
0,2097	0,1104
-0,197	-0,102

Each core aggregates for a part oft he position runs

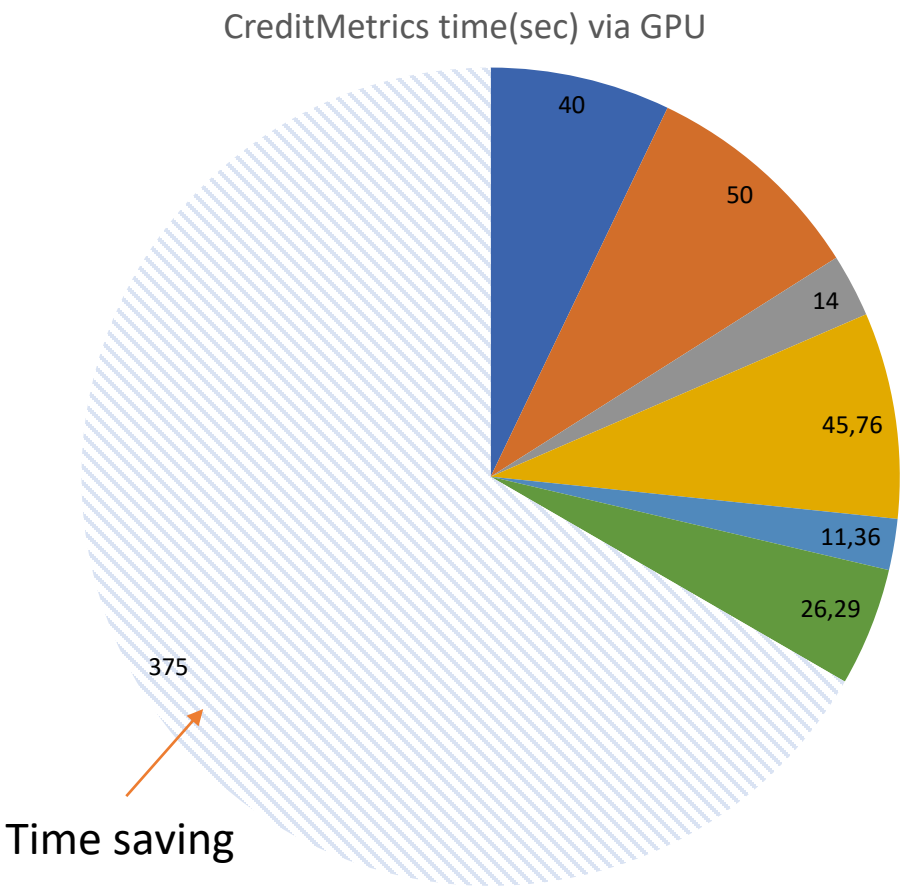
Sub-sums: each core aggregates several position values

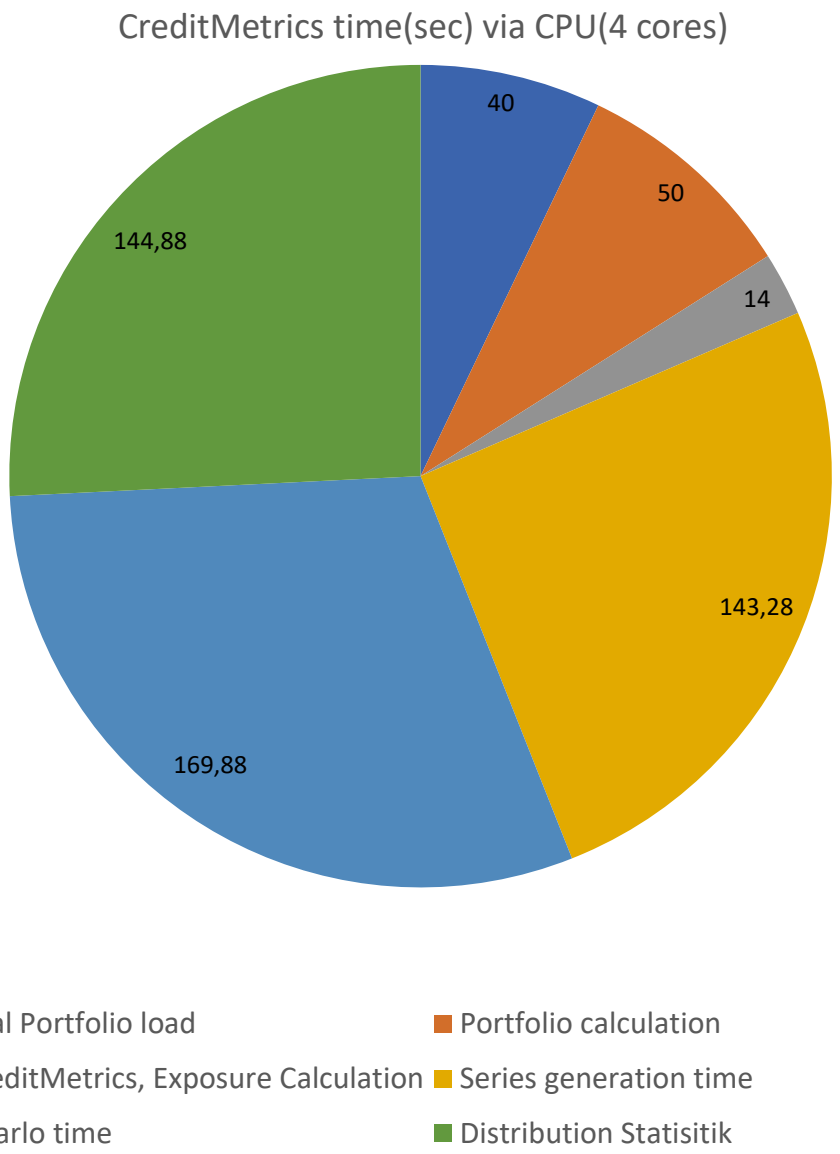
Synchronize

$$\sqrt{\frac{\sum_{i=0}^{cores}(SubSum_i)^2}{(n-1)}}$$

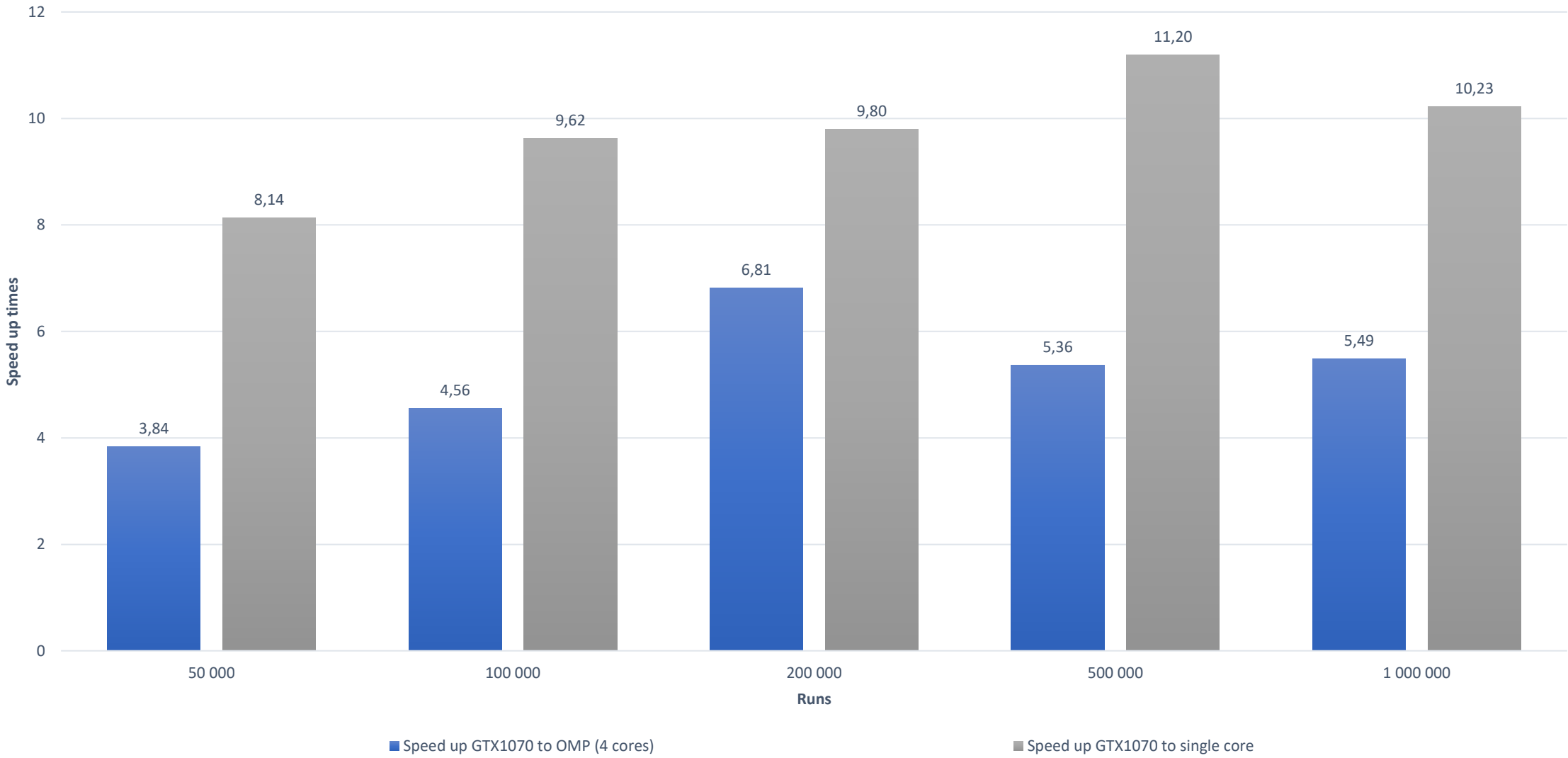
CPU aggregates the final result for the standard deviation

Complete Credit metrics time improvement

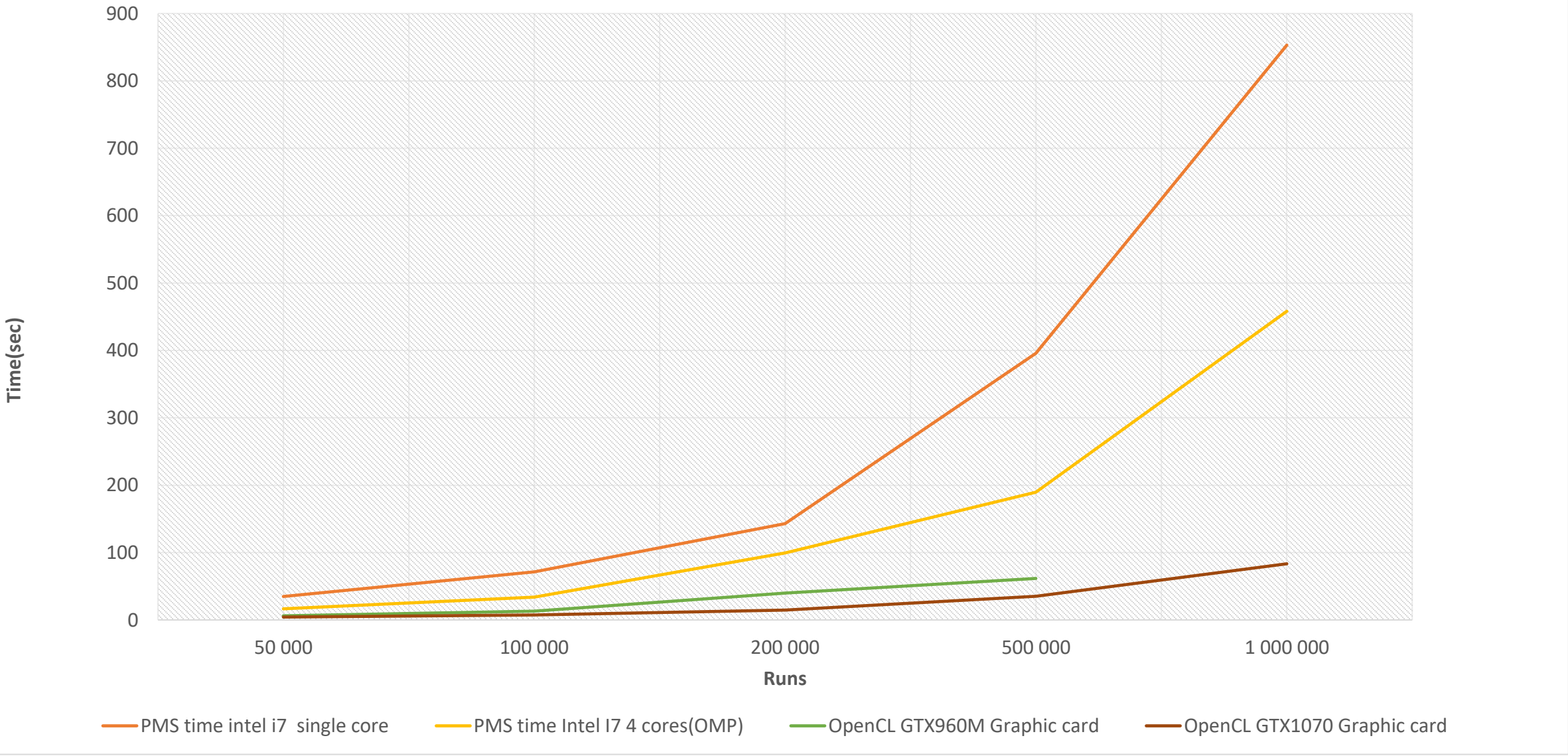




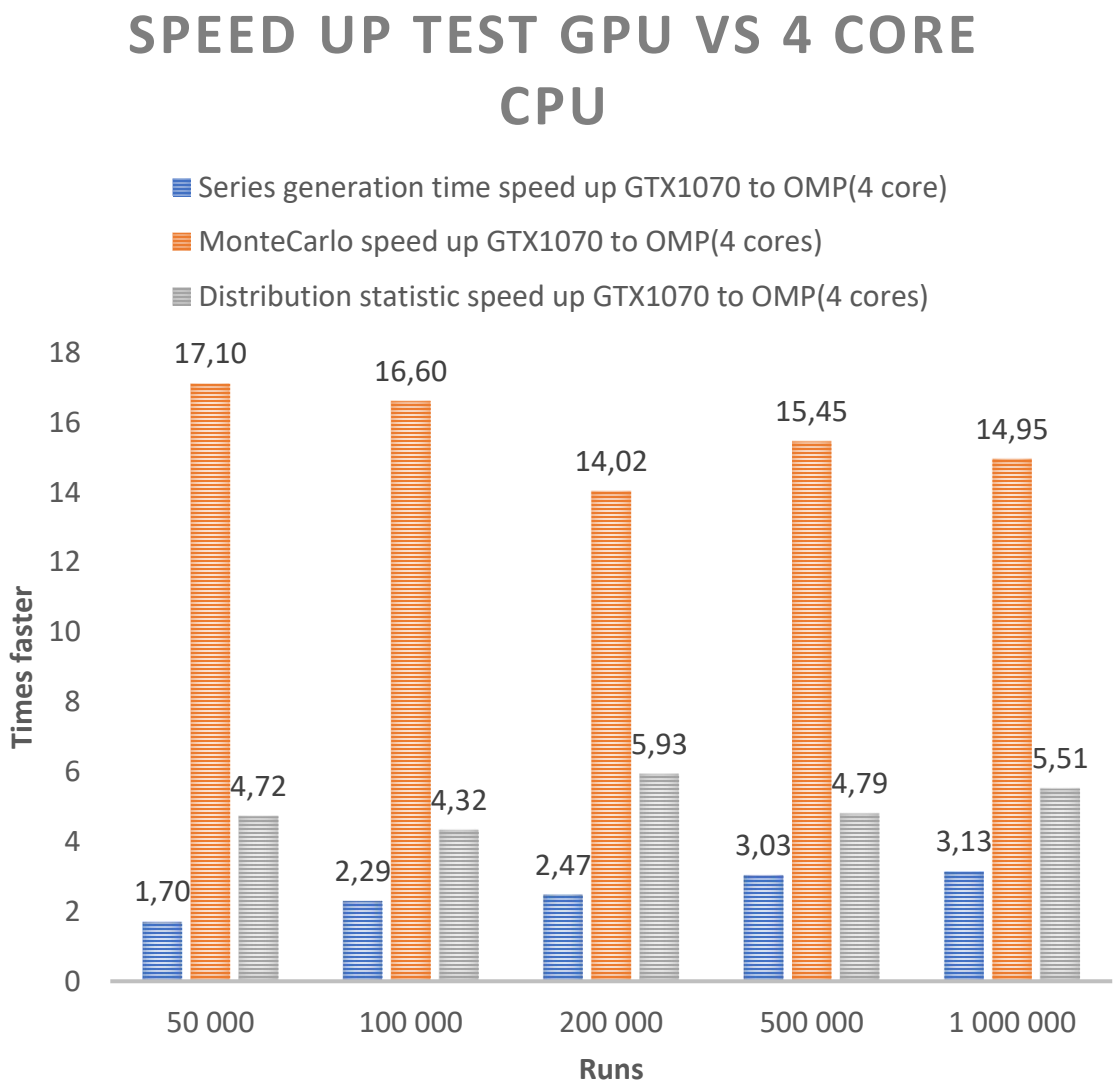
Overall speed up comparasion

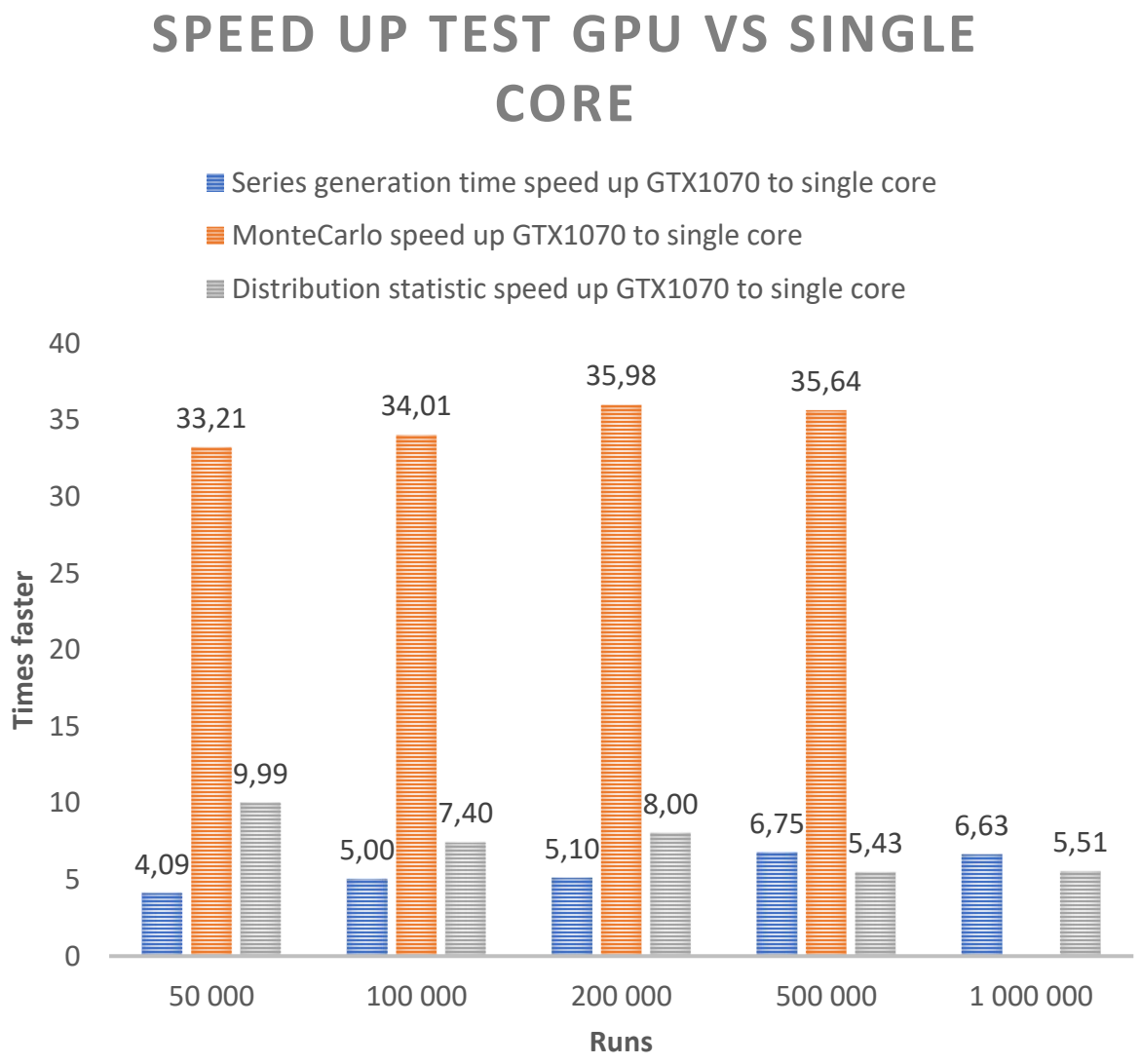


Calculation Time (sec) vs. Number of Runs chart



Speed up tests charts





Speed up tests table

		Time(sec)				Speed up		
runs	Stage	PMS time intel i7 single core	PMS time Intel I7 4 cores(OMP)	OpenCL GTX960M Graphic card	OpenCL GTX1070 Graphic card	Speed up GTX960M to OMP (4 cores)	Speed up GTX1070 to OMP (4 cores)	Speed up GTX1070 to single core
50.000	Series generation time	14,87	6,18	4,48	3,64	1,38	1,70	4,09
	MonteCarlo time	19,26	9,92	1,69	0,58	5,88	17,10	33,21
	Distribution Statisitik	0,93	0,44	0,21	0,09	2,08	4,72	9,99
	TOTAL	35,06	16,54	6,38	4,31	2,59	3,84	8,14
100.000	Series generation time	29,92	13,69	9,27	5,98	1,48	2,29	5,00
	MonteCarlo time	39,22	18,81	3,31	1,13	5,68	16,60	34,62
	Distribution Statisitik	2,42	1,41	0,55	0,33	2,56	4,32	7,40
	TOTAL	71,57	33,92	13,13	7,44	2,58	4,56	9,62
200.000	Series generation time	58,22	28,24	18,66	11,43	1,51	2,47	5,10
	MonteCarlo time	77,75	32,06	6,66	2,29	4,82	14,02	34,01
	Distribution Statisitik	7,03	5,22	1,60	0,88	3,25	5,93	8,00
	TOTAL	143,01	99,43	40,05	14,59	2,48	6,81	9,80
500.000	Series generation time	154,81	69,58	35,84	22,95	1,94	3,03	6,75
	MonteCarlo time	204,60	87,84	16,56	5,69	5,31	15,45	35,98
	Distribution Statisitik	36,49	32,22	9,22	6,72	3,50	4,79	5,43
	TOTAL	395,91	189,63	61,61	35,35	3,08	5,36	11,20
1.000.000	Series generation time	303,30	143,28	NO Memory	45,76	NO Memory	3,13	6,63
	MonteCarlo time	404,99	169,88		11,36		14,95	35,64
	Distribution Statisitik	144,76	144,88		26,29		5,51	5,51
	TOTAL	853,05	458,03		83,42		5,49	10,23

1 Core CPU

4 Core CPU

640 Core GPU

1920 Core GPU

Technical Specifications of the test hardware

Type	Brand	Model	Clock Rate(GHz)	Cores	Threads per core
CPU	Intel	i7 6700HQ	2,6	4	2
GPU	NVIDIA	GTX960M	1,097	640	32
GPU	NVIDIA	GTX1070	1,683	1920	32

Type	Brand	Model	Price in Euro	**Performance to price**
CPU	Intel	i7 6700HQ	300	**1**
GPU	NVIDIA	GTX960M	200	**3,08**
GPU	NVIDIA	GTX1070	550	**9,83**

Time of simulation divided by price and assimilated to CPU prices (higher is better)