

# **EA High-Performance Reservoir Simulations on Modern CPU-GPU Computational Platforms\***

**Kirill Bogachev<sup>1</sup>, Sergey Milyutin<sup>1</sup>, Alexey Telishev<sup>1</sup>, Vladimir Nazarov<sup>1</sup>, Vasilii Shelkov<sup>1</sup>, Dmitry Eydinov<sup>1</sup>, Olivier Malinur<sup>2</sup>, and Simon Junior Hinneh<sup>2</sup>**

Search and Discovery Article #70385 (2019)\*\*

Posted March 11, 2019

\*Adapted from extended abstract prepared in conjunction with poster presentation given at 2018 International Conference and Exhibition, Cape Town, South Africa, November 4-7, 2018

\*\*Datapages © 2019 Serial rights given by author. For all other rights contact author directly. DOI:10.1306/70385Bogachev2019

<sup>1</sup>CTO, Rock Flow Dynamics, Moscow, Russian Federation

<sup>2</sup>Consultant, WellCome Energy Services, Accra, Ghana ([hinnehsimon@outlook.com](mailto:hinnehsimon@outlook.com))

## **Abstract**

With growing complexity and grid resolution of the numerical reservoir models, computational performance of the simulators becomes essential for field development planning. To achieve optimal performance reservoir modeling software must be aligned with the modern trends in high-performance computing and support the new architectural hardware features that have become available over the last decade. A combination of several factors, such as more economical parallel computer platforms available on the hardware market and software that efficiently utilizes it to provide optimal parallel performance helps to reduce the computation time of the reservoir models significantly. Thus, we can get rid of the perennial problem with lack of time and resources to run required number of cases. That, in turn, makes the decision about field development more reliable and balanced.

In the past, the advances in simulation performance were limited by memory bandwidth of the CPU based computer systems. During the last several years, a new generation of graphical processing units (GPU) became available for high-performance computing with floating point operations. That made them a very suitable solution for the reservoir simulations. The GPU's currently available on the market offer significantly higher memory bandwidth compared to CPU's and thousands of computational cores that can be efficiently utilized for simulations.

In this paper, we present results of running full physics reservoir simulator on CPU+GPU platform and provide detailed benchmark results based on hundreds real field models. The technology proposed in this paper demonstrates significant speed up for models, especially with large number of active grid blocks. In addition to the benchmark results we also provide analysis of the specific hardware options available on the market with the pros and cons and some recommendations on the value for money of various options applied to the reservoir simulations.

## Introduction

In 1965 Gordon Moore predicted that the number of transistors in a dense integrated circuit has to double every two years (Moore, 1965). This is referred to as the Moore's law and it has proved to be true over the last 50 years. Over the last several years it has been mostly achieved by growth in the number of CPU cores integrated within one shared memory chip. The number of cores available on the CPU's has grown from 1-2 in 2005 to 20+ and keeps growing every year (see <https://www.intel.com/content/www/us/en/products/processors/xeon.html>). In the reality multicore supercomputers are not exclusive anymore and have become available for everyone in shape of laptops or desktop workstations with large amount of memory and high-performance computing capabilities.

There has also been a significant improvement in the cluster computing options that were very expensive and required special infrastructure, costly cooling systems, significant efforts to support, etc. only about 10-15 years ago. These days the clusters have also become economical and easy-to-use machines. It has been demonstrated that the reservoir simulation time can be efficiently scaled on the modern CPU-based workstations and clusters if the simulation software is implemented properly to support the features of modern hardware architecture (Dzyuba et al., 2012 and Tolstolytkin et al., 2014). It is fair to conclude that the simulation time is not a principal bound for the projects anymore, as it can always be reduced by adding extra computational power. In addition to in-house hardware one can consider simulation resources available on the clouds based on pay-per-use model (see <https://aws.amazon.com/>). This makes the reservoir modeling solutions much more scalable in terms of time, money, human and computational resources. Reservoir engineers from large and small companies can have access to nearly unlimited simulation resources and reduce the simulation time to minimum whenever it is required.

There exists another hardware technology that progresses rapidly - graphical processing units (GPU). The modern GPU's have thousands of cores that are suitable for high-performance parallel simulations. However, it is not even the large number of cores that makes GPU's particularly attractive for this purpose. One of the most critical parameters for parallel simulations is memory bandwidth which is effectively the speed of communication between the computational cores. [Figure 1](#) shows a typical example of simulation time spent on computation of a reservoir model. A significant part of it is the linear solver (60% in this case), and this task is the most demanding with respect to the memory bandwidth due to the large volume of communications. That is why reservoir simulations is often considered to be a problem of the so-called 'memory bound' type.

If we compare the memory bandwidth characteristics of the CPU and GPU platforms available today ([Figure 2](#)), we can see that the difference in parameter is about one order of magnitude, i.e. from 100GB/s to 1000GB/s respectively. Moreover, the progress of the GPU's has been very rapid over the last several years, and the trend clearly shows that the gap in memory bandwidth parameter keeps growing (NCAR 2017). This means that the GPU platforms will become more and more efficient for parallel computing and reservoir simulations in particular.

## Hybrid CPU-GPU Approach

There have been a few recent developments applying GPU for the reservoir simulations, e.g. (Yu et al., 2012; Khait and Voskov, 2017; and Maea and Tchelep, 2017). Typical approach is to adopt the GPU for the whole simulation process. In this work, we utilize the hybrid approach where the workload is distributed between CPU and GPU in an efficient manner. This method was first introduced in Telishev et al., 2017 and

this paper is an extension of that work. The idea of the proposed technology is to delegate the most computationally intensive parts of the job to the GPU and leave everything else on the CPU side.

In Telishev et al., 2017 the part of the jobs assigned to the GPU was limited to the linear solver. It worked very well for large and mid-size problems (say, 1 million active blocks or higher) and provides many-fold acceleration compared to the conventional CPU-based simulators. In many cases acceleration 2-3 times was observed. In specific cases like SPE 10 model (Christie and Blunt, 2001) where the time spent on linear solver is nearly 90% due to extreme contrast in the reservoir property  $v$ , the speed-up due to including GPU was almost 6 times (Figure 3).

In this work we extend the approach for compositional models and also include EOS flash calculations in the part of the calculations that are run on the GPU.

### **Flash Calculations for Compositional Models**

For compositional models, it is required to identify the parameters of phase equilibrium of the hydrocarbon mixture. This part can take as much as 10-20% of the total modeling time for real cases. Opposed to linear equations with sparse matrices, the standard solution methods for flash calculations (Michelsen, 1982) are primarily bound by the double-precision calculation speed, and not the memory bandwidth. In addition, the algorithm for the phase equilibrium calculations can be naturally vectorized as a sequence of iteration series for certain sets of grid blocks, where the calculations for each set are independent from the other grid blocks. To achieve maximum computation efficiency on CPU's we can change the approach from single block calculations to using embedded CPU instructions SSE/AVX (Reinders, 2013) that allow to work with vector of parameters for multiple grid blocks at the same time.

Flash calculations in the fluid flow models can also be efficiently carried out on the graphical cards, especially for the cases with large areas with oil and gas phases present. Comparison of the simulation performance for this method is discussed in the results section.

The workload distribution between the CPU's and GPU's can be summarize as follows (Figure 4). The GPU is assigned to run the most computationally intensive parts of the simulations, such as the linear solver and the flash calculations, and everything else remains on the CPU side.

The communication between CPU's and GPU's is going via PCI Express slots. In this example a dual socket CPU system is equipped with multiple graphical cards and each CPU communicates with the corresponding GPU's. The multiple graphical cards communicate directly via fast interconnect channels. This is the most general scheme. Practically in most cases used for reservoir simulations the system consists of one or two GPU's connected to one or two PCIE respectively. In this paper, we only consider examples with single or dual GPU systems and single GPU per PCIE slot.

### **Hardware Options Available**

Table 1 presents some hardware alternatives currently available on the market. CPU models are shown in the upper part of the table, and the

key GPU models are listed in the bottom. We also provide some pricing estimate for the readers information. Note, the prices can vary region by region and also change in time. So, the information here is a rough approximation, but gives fair understanding of the cost level and price ratio between different options.

There are several parameters that are fundamentally different between CPU's and GPU's:

- **Memory Bandwidth.** As discussed above, this parameter is one of the most critical for parallel computation performance in reservoir simulations. As one can see from the table, graphical processors offer significantly better solution here, including the basic models (GTX) for mass markets that can be purchased at a very reasonable cost. More advanced and expensive Tesla cards have higher bandwidth. However, as it follows from our observations, in many cases GTX processors work very well and achieve almost the same level of computation performance as the more expensive alternatives. It is also worth to emphasize new EPYC processors by AMD that offer higher memory bandwidth parameters and lower price than Intel Xeons that have been dominating the simulation hardware market during the last years. This is another interesting and promising solution that can be expected to grow in the future and should be considered for the engineers running reservoir simulations.
- **Number of Cores.** The number of cores on the GPU's is dramatically higher than on the CPU's. However, it has to be emphasized that each GPU core is much weaker compared to the ones in the CPU's, and do not provide equal calculation speed. From the practical point of view though the single core performance is not so important, as in all cases it is natural to utilize all available machine resources. For both CPU and GPU, the communication between shared memory cores is fast and efficient, and opposed to the distributed memory does not require model grid domain decomposition as would be required for HPC clusters (Dzyuba et al., 2012).
- **Memory Size.** This parameter is critical for the size of models that can be handled by the corresponding hardware. In this aspect CPU's look more advanced. However, over the last years GPU's have been also progressing rapidly, and many graphical cards will be found suitable for most of the real field models. In case the model does not fit in the memory available, systems with multiple graphical cards need to be considered.

## Results

In this section, we present results for the simulation time of models with various grid dimensions and physics on various computational platforms. The results give a fair overview of what level of simulation performance can be expected from the hardware platforms considered.

### EOS Flash Calculations

In this experiment we consider a real field simulation model with 450 thousand active grid blocks, 13 wells and about 1000 well completions, 9 hydrocarbon components + water. The computations were carried out on a PC with dual Intel Xeon CPU E5-2680v4@2.40GHz (28 cores) and GPU Nvidia Tesla P100-SXM2-16GB (3584 Cuda cores). Three cases were considered: regular with single block operations in flash

calculations (I), AVX (II), and hybrid CPU-GPU (III). The results are presented on [Figure 5](#). Up to 2018 the simulation time is nearly identical, because in this case the fraction of grid blocks with two hydrocarbon phases is very small (about 0.001%). However, after 2018 the fraction of blocks with mixed phases is around 15%, and in that case the speed-up with the GPU version is significant, around 12%.

In the next experiment we consider another compositional model with 770 thousand active grid cells, 10 components and 90 wells. [Figure 6](#) shows the total calculation time on a laptop Quad Core i7 6700K and GPU Nvidia GTX 1080. We compare 4 simulation cases: conventional calculations limited to the CPU; GPU is enabled for the linear solver; CPU AVX for flash calculations – linear solver on GPU; flash and solver on GPU. As you can see from the plots the most efficient way is to run both flash and linear solver calculations on the GPU side. It is worth mentioning though that the AVX CPU technology also improves the simulation time and can in general be considered as an alternative. The biggest improvement in this case, as expected, comes from the linear solver delegated to the GPU which gives around 30% saving on the simulation time.

### **Large Grid Simulations**

In this experiment we consider a large black oil model with dual porosity and 18 million active grid blocks. The model has 10 wells with multistage fractures and local grid refinement around them. The benchmark was run on a computer with dual AMD EPYC 7601 CPU's and dual Tesla V100 video cards. The comparison includes 3 cases: CPU only, +1 video card, and +2 video cards enabled. In this case, we observe a very interesting behavior for the model simulation performance. Adding one GPU for the linear solver calculation slows this simulation down, and the second card eventually improves it ([Figure 7](#)).

This can be explained by slow communication via single PCIE slot. The matrix generated by the CPU for the linear solver on the GPU is quite large. At each iteration the large matrix has to be transferred from CPU to GPU via PCIE slot which is slower than both CPU and GPU access to the respective local memory. In this case, the PCIE communication becomes the bottleneck and takes more time than solving the system directly on the powerful EPYC CPU's without getting the GPU involved. However, when the second card is included and two PCIE are used in parallel, the data exchange between CPU's and GPU's does not dominate anymore, and the total simulation time improves.

This example shows how non-linear solver behavior can be. In general, all simulation models have individual features and to draw more general conclusions larger model sets have to be considered. This is illustrated in the next examples.

### **Multiple Cases Analysis**

In this section we compare computational speed of several variations of CPU/GPU hardware platforms on multiple model cases and aim to investigate the performance indicators with statistical analysis to generalize conclusions about the optimal configurations.

[Figure 8](#) shows simulation time results on 5 real field black oil simulation models of average size (all around 1 million active grid cells) on three different machines with and without GPU's enabled: 2x Intel Xeon E5-V2 with Tesla Quadro P6000; 2x Intel Xeon E5-V4 with GTX 1080; and 2x AMD EPYC with GTX 1080. Oldest CPU E5-V2 is the weakest compared to the others and significant improvement of the

simulation time can be observed when the GPU is enabled on that platform. This result can be generalized for the other CPU models as well. Adding GPU, more powerful Tesla cards, or more economical GTX models, will in most cases help to achieve significant improvement in the simulation speed. Technically the installation of a graphical card is straightforward, it just needs to be inserted in the PCIE slot. Thus, this option can be very useful for all reservoir engineers who use less powerful computers to improve the simulation performance. As illustrated above with the SPE 10 case, the simulation time improvement will in many cases be many-fold, which is a great gain for several hundred dollars upgrade (here we mean the GTX card family).

For the same models run on the most powerful Dual AMD EPYC platform the difference in the simulation time is not so significant. In that case, the CPU platform is already powerful enough to handle models of this size and adding GPU does not show dramatic improvement to the simulation performance. Some improvement is still obtained, but it is only 10-20% of the total time. For larger models, the difference can be higher assuming the amount of memory on the GPU side is enough for the model in hand, but as has been shown above with the large dual porosity model, it is not always the case. In general, we recommend testing the specific models a machine with the GPU disabled and enabled and compare two cases before running multiple simulations for the project in hand. Pay-per-use cloud solutions offer a wide range of the hardware options, and the tests can be conducted there for a very low price before choosing a specific platform for the modeling project.

[Figure 9](#), [Figure 10](#), [Figure 11](#), [Figure 12](#), and [Figure 13](#) show benchmark series of runs for several platforms on a large number of black oil and compositional models with different grid sizes. Each blue dot represents a single model and the value on the Y-axis represents the ratio between simulation time with the latter platform over the corresponding time on the first one. If the value is greater than 1 the latter option shows higher performance, and vice versa. For the practical problems the computational performance of each platform has to be considered along with the costs. For that purpose, we also indicate a rough estimate of the hardware costs ratio in the figure captions, which helps to understand the value for money aspects in each case.

The first sample does not include any GPU runs. Here we only compare two different generations of Intel Xeon CPU's. As expected, the performance of the more powerful V4 is higher than V2 in the absolute majority of cases.

In the next example ([Figure 10](#)) we compare model samples run on CPU only and on CPU-GPU hybrid with Tesla P100 card. The result is not so straightforward here, but we can clearly observe a trend that the speed-up factor grows with the number of active grid cells. For large cases (> 1 million active cells) almost all models demonstrate improvement with the GPU enabled. For models with less than 100 thousand active cells most cases slow down, because for small models the fraction of time spent on the linear solver is very small, and the gain from adopting the GPU for them is smaller than the loss on CPU-GPU communications. For the range between 100 thousand and 1 million active cells the results are mixed with the clear trend of speed-up growing with growing model dimensions.

In the next experiment ([Figure 11](#)) we investigate the effect of adding a second GPU for the calculations. For most of the cases the difference is not very significant. There are several models that demonstrate speed-up factor close to 2 likely due to higher bandwidth between CPU and GPU due to two PCIE slots. However, they look more like exceptions, and for most of the models the difference remains within 25% of the total simulation time.

Figure 12 shows comparison of two graphical cards of the Tesla family, i.e. P100 and V100. The latter expectedly shows better results, which is a natural consequence of the key characteristics: greater number of cores and higher memory bandwidth. The time spent on the linear solver and flash calculations in the models is reduced further due to the use of a more powerful graphical processor.

In the last example (Figure 13) we compare two CPU platforms: dual Intel Xeon (28 cores) and dual AMD EPYC (56 cores). In both cases all CPU cores are used for simulations. Naturally for medium and large models EPYC shows better performance and the trend clearly shows that the difference grows for larger cases. When it comes to small models, the effect is the opposite. This comes from the ratio of active grid blocks per core. In case model with about 50 thousand grid blocks is run on 56 EPYC cores, there are only about 1 thousand active grid blocks per core. This is far too small for efficient parallelization, the matrix blocks in the linear solver become too small and the communications between the cores start dominating over the gain from parallel calculations. However, for practical models with resolution starting from 200-300 thousand active blocks AMD EPYC turns out to be clearly a more efficient option.

### **Discussion and Conclusions**

It has been demonstrated that the hybrid CPU+GPU approach with proper balancing of the computational workload can deliver good performance for reservoir simulations. Wider memory bandwidth is one of the most critical parameters here, and the modern hardware solutions offer significant improvement in that aspect.

The advantages of adopting GPU's for simulations are mostly observed on models of large and medium size. For reservoir models with a small grid, where the linear solver takes relatively short time, the hybrid approach may not be the most efficient, and regular CPU runs often turn out to be faster. The results turn out to be model sensitive and it is recommended to test each particular case to choose whether the GPU has to be enabled for calculations.

A new wave of competition between hardware vendors stimulates more rapid progress in the industry. New exciting solutions are expected to come from CPU's, GPU's, FPGA (Field-Programmable Gate Array) technologies. At this point it is not clear where we are going to be in the next 2-3 years, but the GPU appears to be a good choice providing high performance for reasonable price. In any case, the Hybrid CPU + GPU approach in reservoir simulation software development appears to be a good bet generating great performance improvements and allowing to keep up with all the emerging variety of hardware.

### **References Cited**

Bogachev, K., V. Shelkov, Y. Zhabitskiy, D. Eydinov, and T. Robinson, 2011, A New Approach to Numerical Simulation of Fluid Flow in Fractured Shale Gas Reservoirs: Canadian Unconventional Resources Conference, 15-17 November, Calgary, Alberta, Canada, SPE-147021-MS, 6 p. doi:10.2118/147021-MS

Christie, M.A., and M.J. Blunt, 2001, Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques: SPE Reservoir Evaluation and Engineering, v. 4/04, SPE-72469-PA, 10 p. doi:10.2118/72469-PA



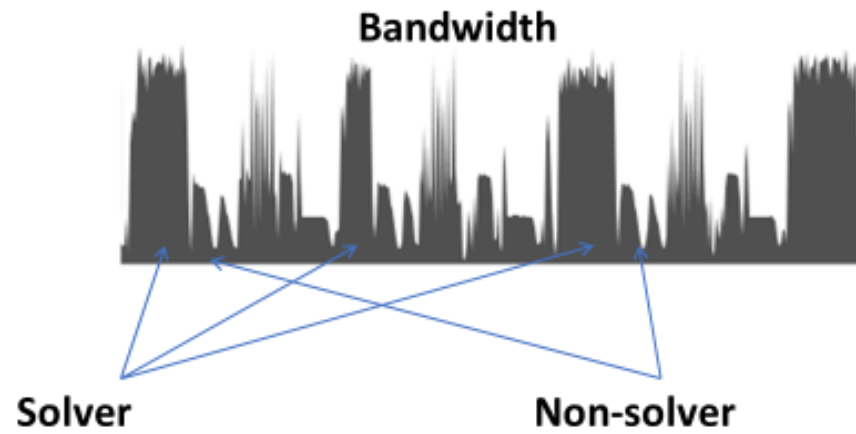
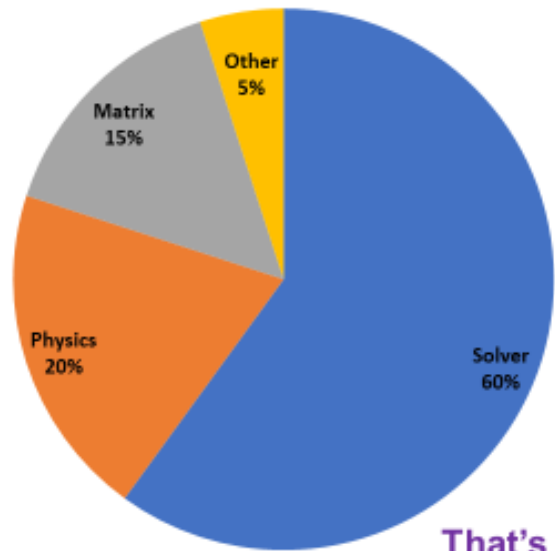
- Dzyuba, V.I., K.Y. Bogachev, A.S. Bogaty, A.R. Lyapin, A.R. Mirgasimov, and A.E. Semenko, 2012, Advances in Modeling of Giant Reservoirs: SPE Russian Petroleum Technology Conference, 16-18 October, Moscow, Russia, SPE-187797-MS, 8 p. doi:10.2118/163090-MS
- Khait, M., and D. Voskov, 2017, GPU-Offloaded General Purpose Simulator for Multiphase Flow in Porous Media: SPE Reservoir Simulation Conference, 20-22 February, Montgomery, Texas, USA, SPE-182663-MS, 10 p. doi:10.2118/182663-MS
- Manea, A.M., and H.A. Tchelepi, 2017, A Massively Parallel Semicoarsening Multigrid Linear Solver on Multi-Core and Multi-GPU Architectures: SPE Reservoir Simulation Conference, 20-22 February, Montgomery, Texas, USA, SPE-182718-MS, 45 p. doi:10.2118/182718-MS
- Michelsen, M.L., 1982, The Isothermal Flash Problem. Part I. Stability: Fluid Phase Equilibria, v. 9, p. 1-19.
- Michelsen, M.L., 1982, The Isothermal Flash problem. Part II. Phase-split Calculation: Fluid Phase Equilibria, v. 9, p. 21-40.
- Moore, G.E., 1965, Cramming More Components onto Integrated Circuits: Electronics Magazine, p. 4.
- NCAR, 2017, GPU-Accelerated Microscale Modeling: NCAR Research Application Laboratory, 2017 Annual Report.
- Reinders, J., 2013, AVX-512 Instructions: Intel. <https://software.intel.com/en-us/blogs/2013/avx-512-instructions>. Website accessed February 2019.
- Telishev, A., K. Bogachev, V. Shelkov, D. Eydinov, and H. Tran, 2017, Hybrid Approach to Reservoir Modeling Based on Modern CPU and GPU Computational Platforms: SPE Russian Petroleum Technology Conference, 16-18 October, Moscow, Russia, SPE-187797-MS, 8 p. doi:10.2118/187797-MS
- Tolstolytkin, D.V., E.V. Borovkov, I.A. Rzaev, and K.Y. Bogachev, 2014, Dynamic Modeling of Samotlor Field Using High Resolution Model Grids: SPE Russian Petroleum Technology Conference, 16-18 October, Moscow, Russia, SPE-187797-RU, 13 p. doi:10.2118/171225-MS
- Yu, S., H. Liu, Z.J. Chen, B. Hsieh, and L. Shao, 2012, GPU-Based Parallel Reservoir Simulation for Large-scale Simulation Problems: SPE Europe/EAGE Annual Conference, 4-7 June, Copenhagen, Denmark, SPE-152271-MS, 5 p. doi:10.2118/152271-MS

### Websites Cited

<https://www.intel.com/content/www/us/en/products/processors/xeon.html>. Website accessed February 2019.

<https://aws.amazon.com/>. Website accessed February 2019.





That's why reservoir simulations are often called "memory bound"!

Figure 1. Memory bound problems in reservoir simulations.

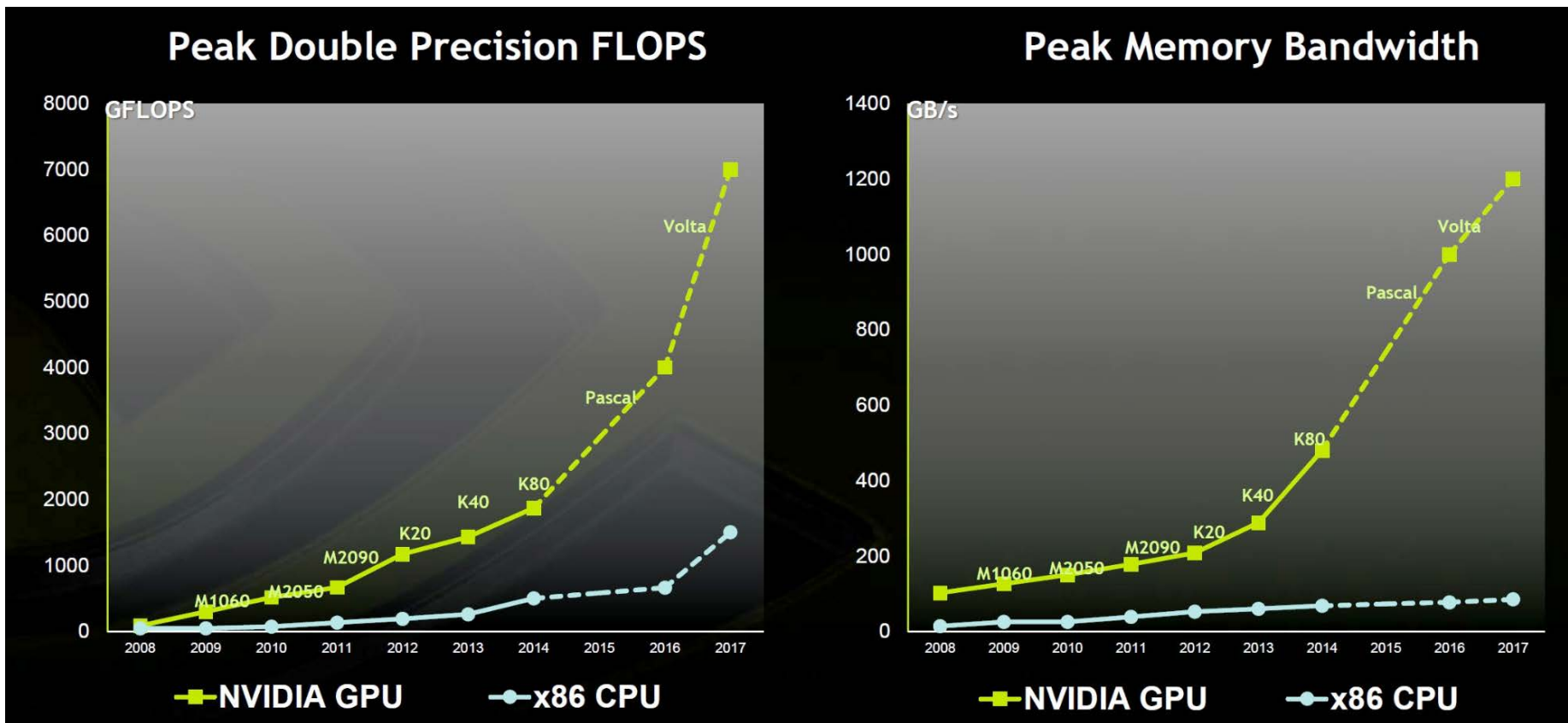


Figure 2. Floating point operations and memory bandwidth progress GPU vs. CPU.

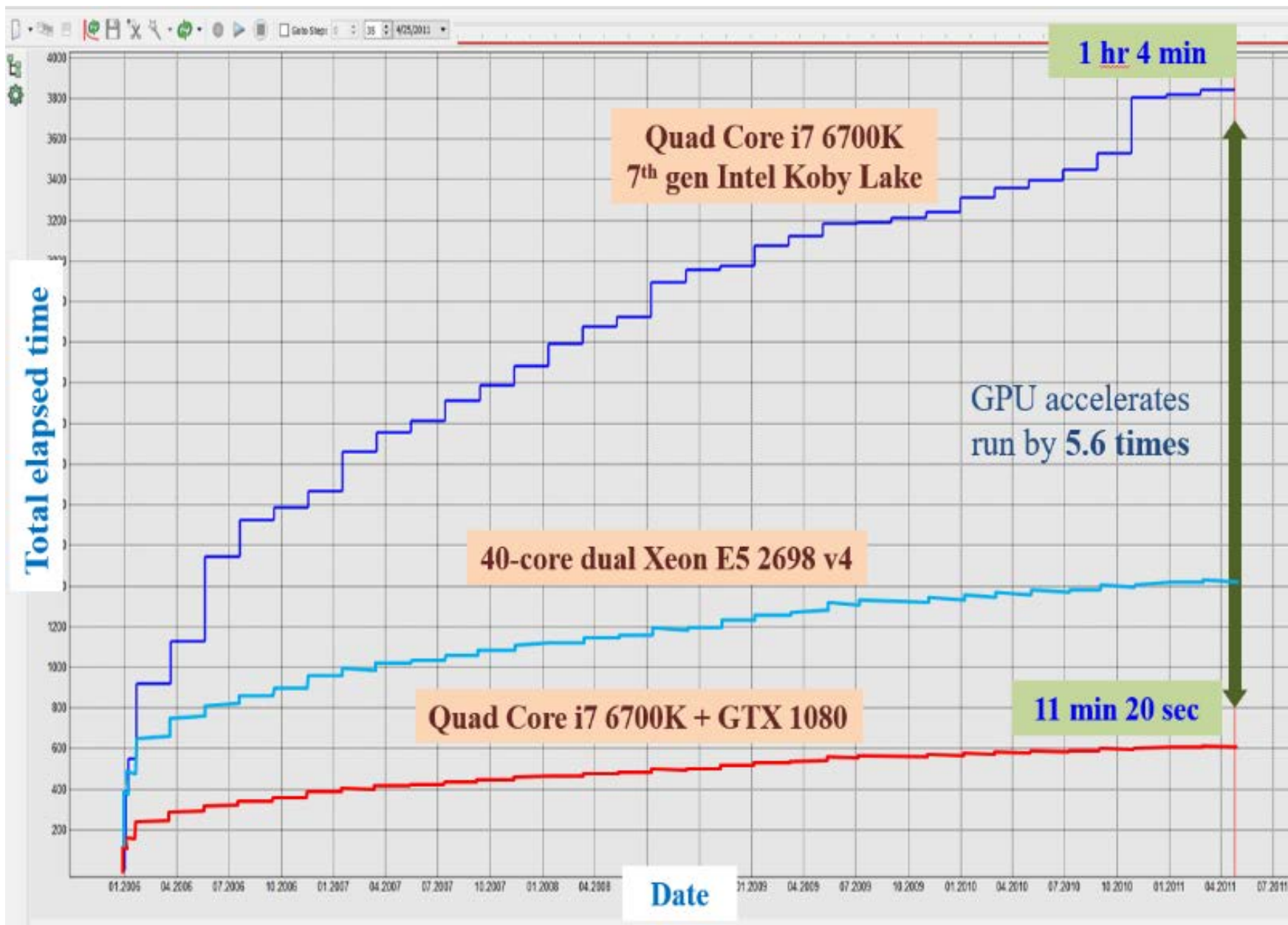


Figure 3. SPE 10 model simulation performance on a laptop with and without GPU and dual CPU workstation commonly used for reservoir simulations.

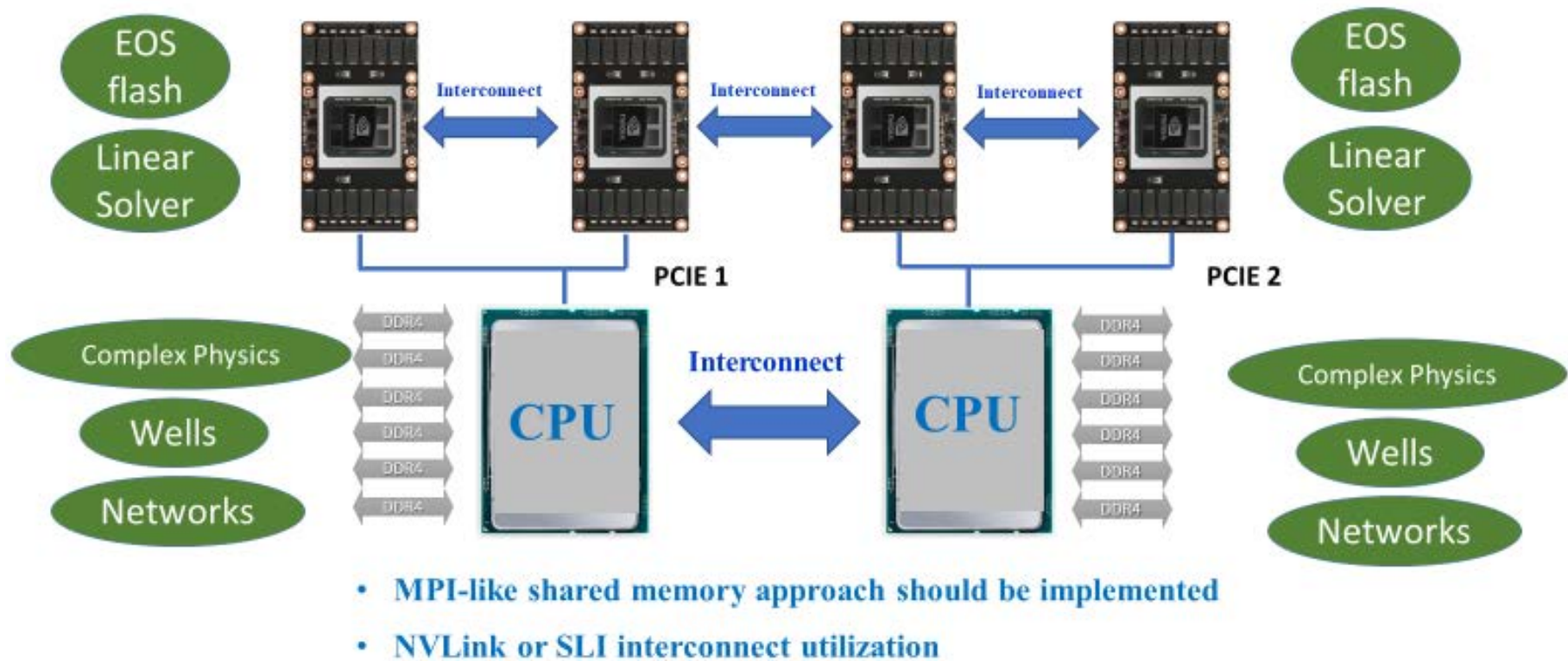


Figure 4. CPU-GPU hybrid algorithm for reservoir simulations.

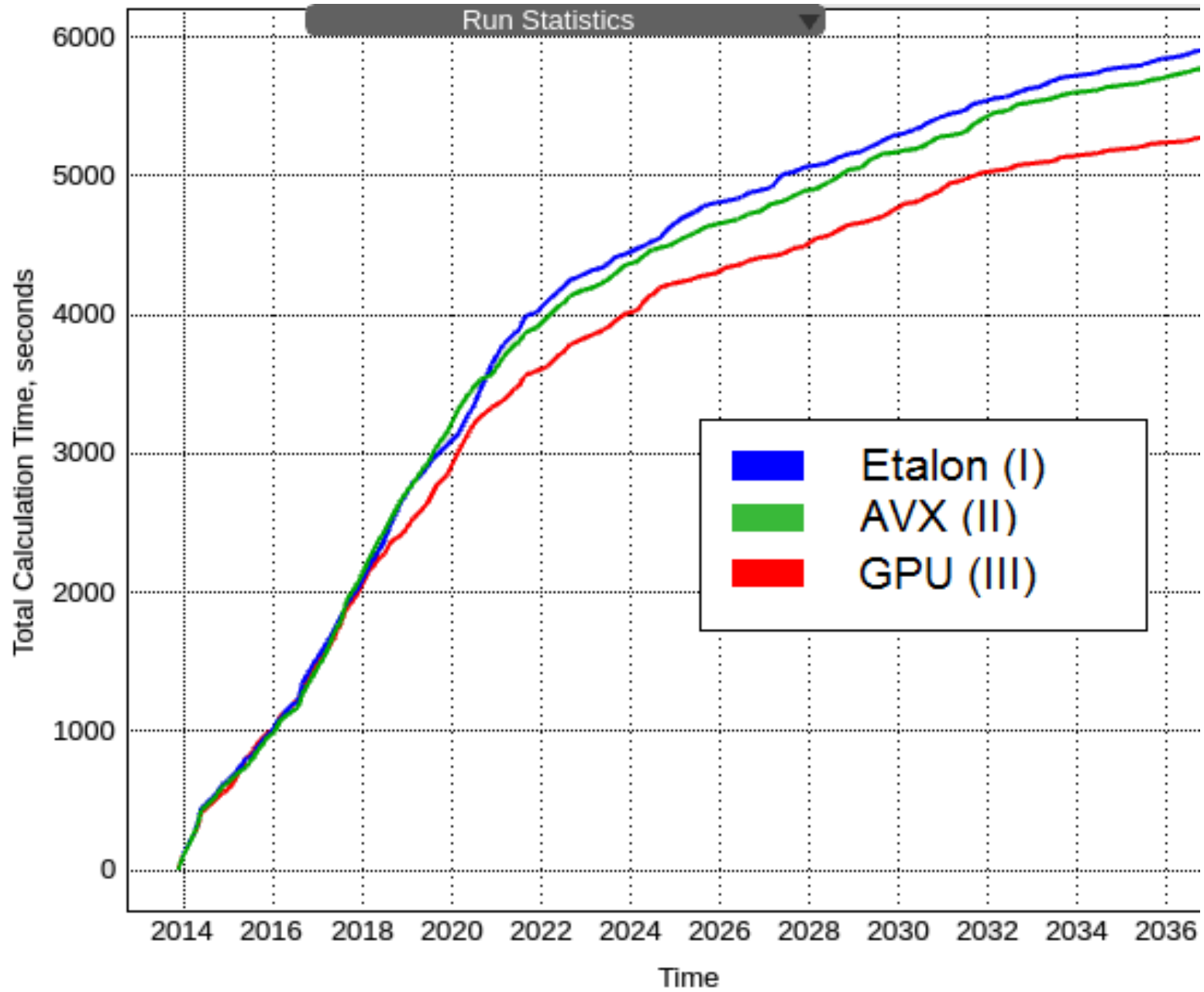


Figure 5. Total calculation time (sec.) for regular CPU approach (I), AVX with vector instructions (II), and hybrid scheme with GPU (III).

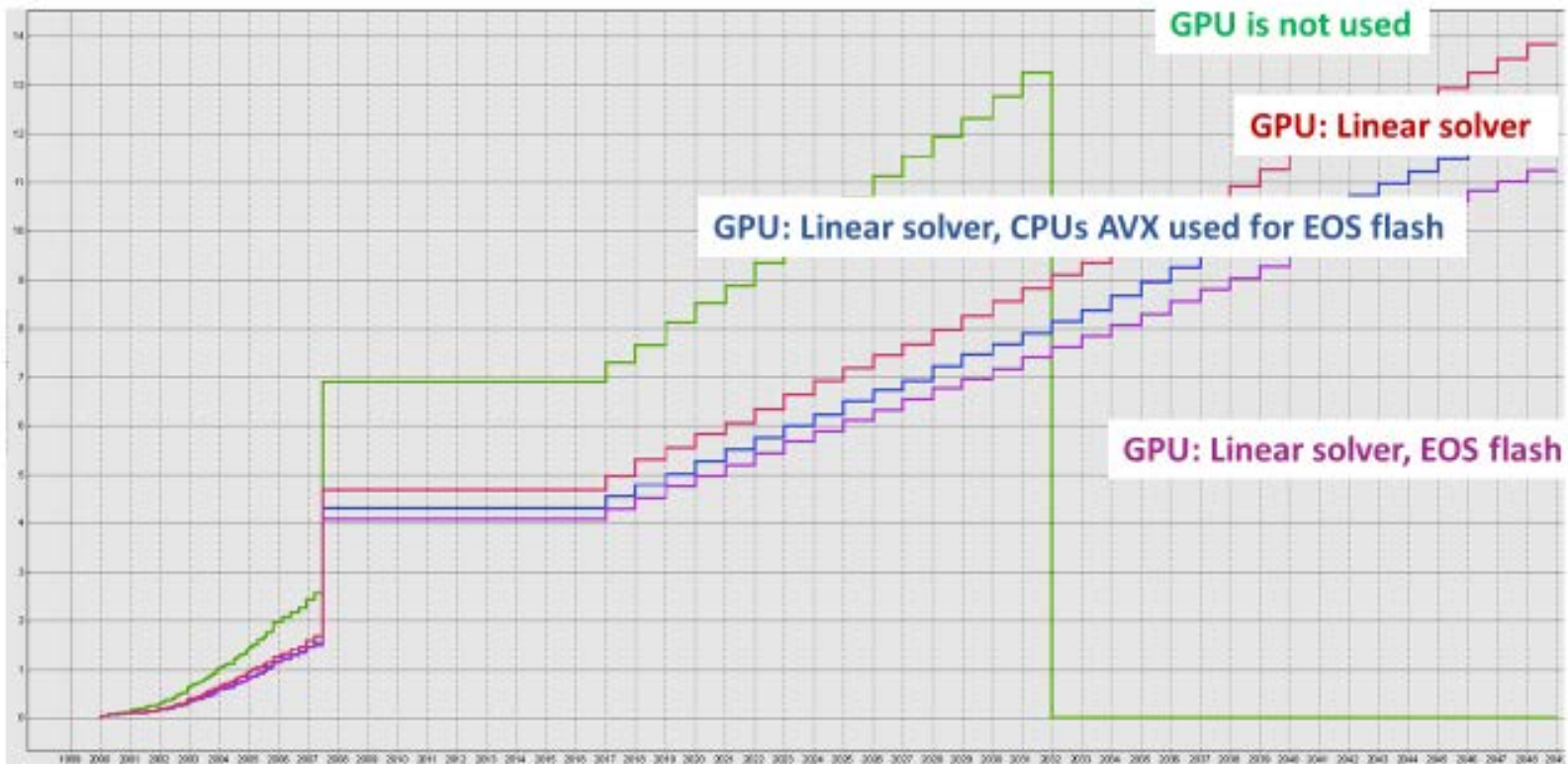


Figure 6. Compositional model 770 thousand active blocks, 10 components, 90 wells, 50 years forecast. Total calculation time.





Figure 7. 18 million active blocks, dual porosity black oil model. 10 wells with multistage fractures and LGR's, 50 years production forecast. Total calculation time.



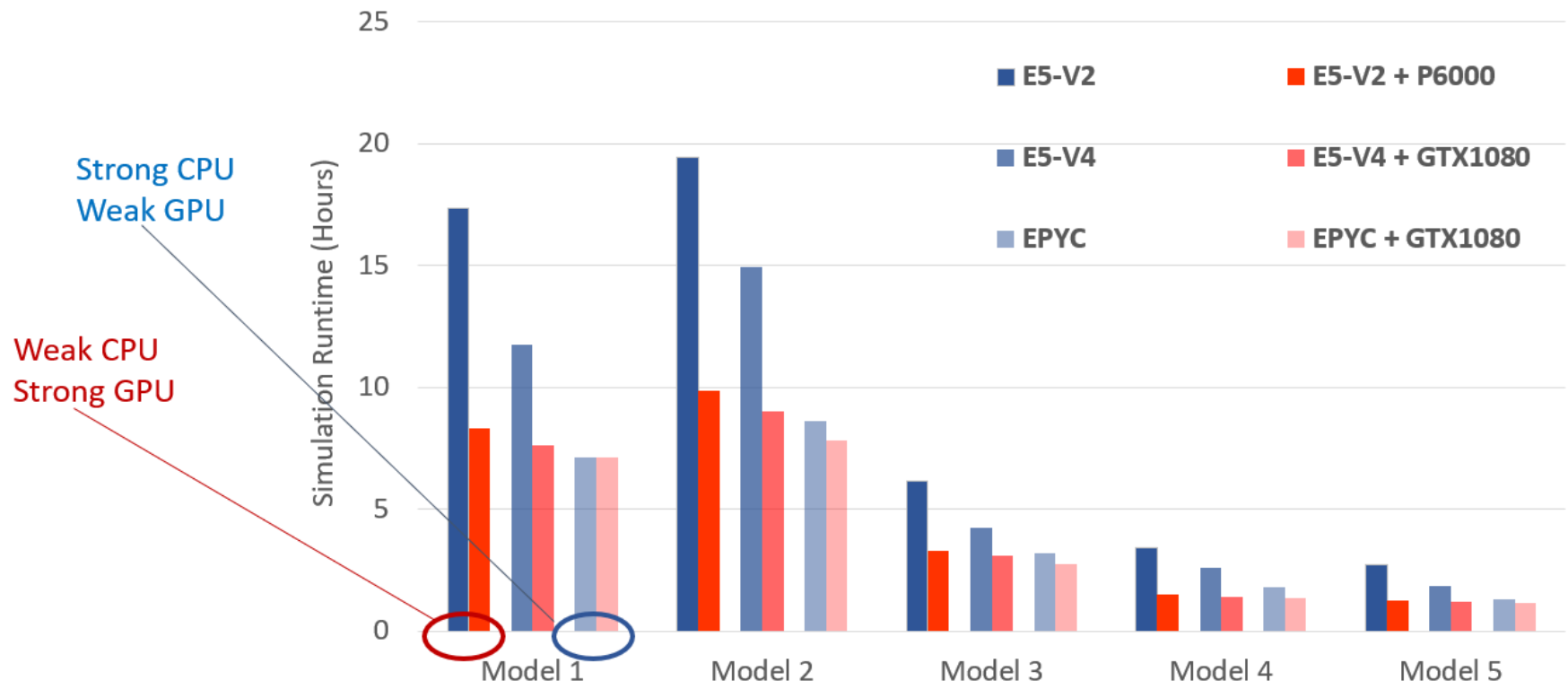
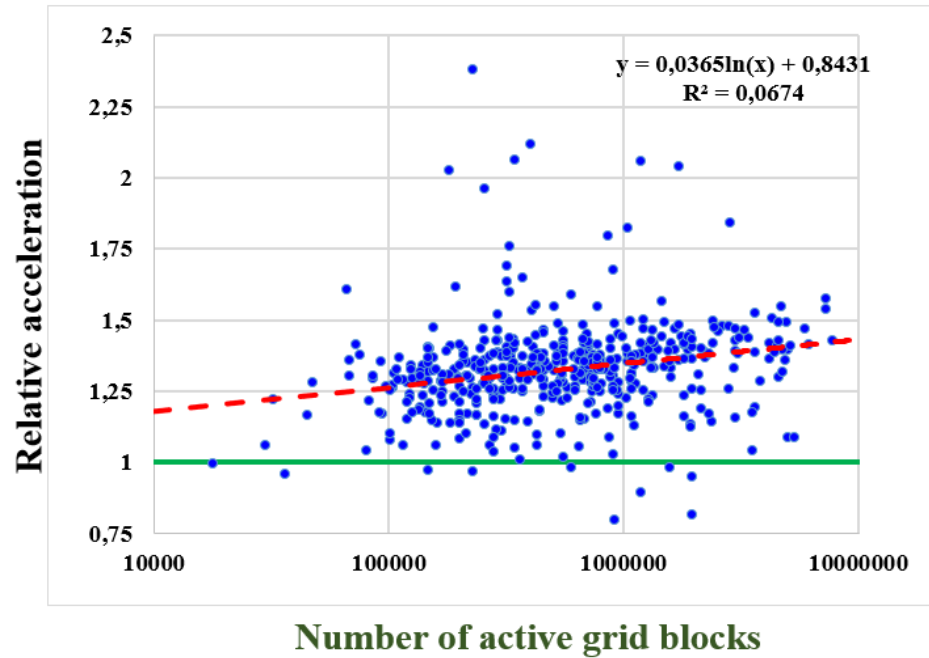


Figure 8. How GPU helps different CPU platforms.

### 506 black oil real field models



### 275 compositional real field models

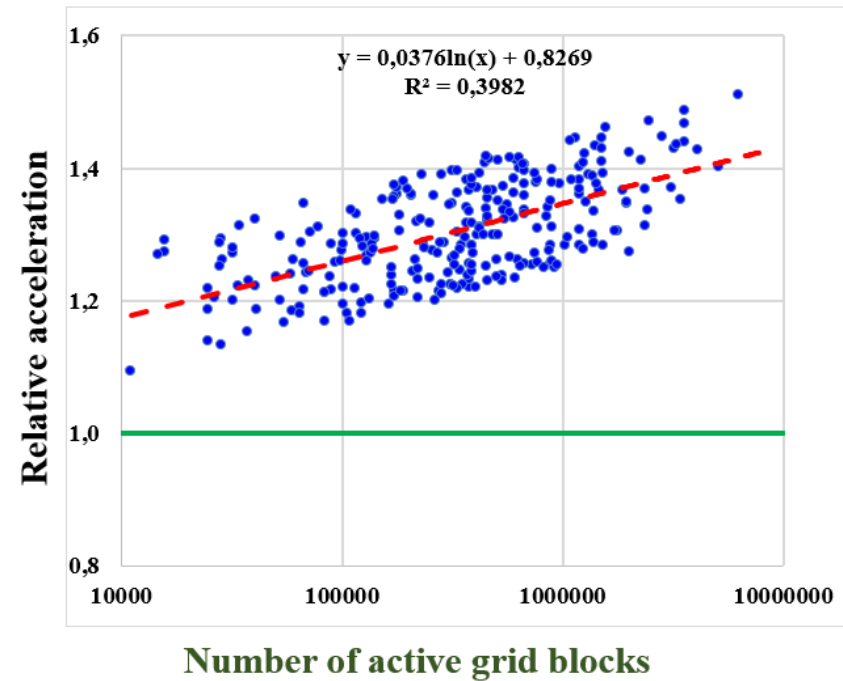
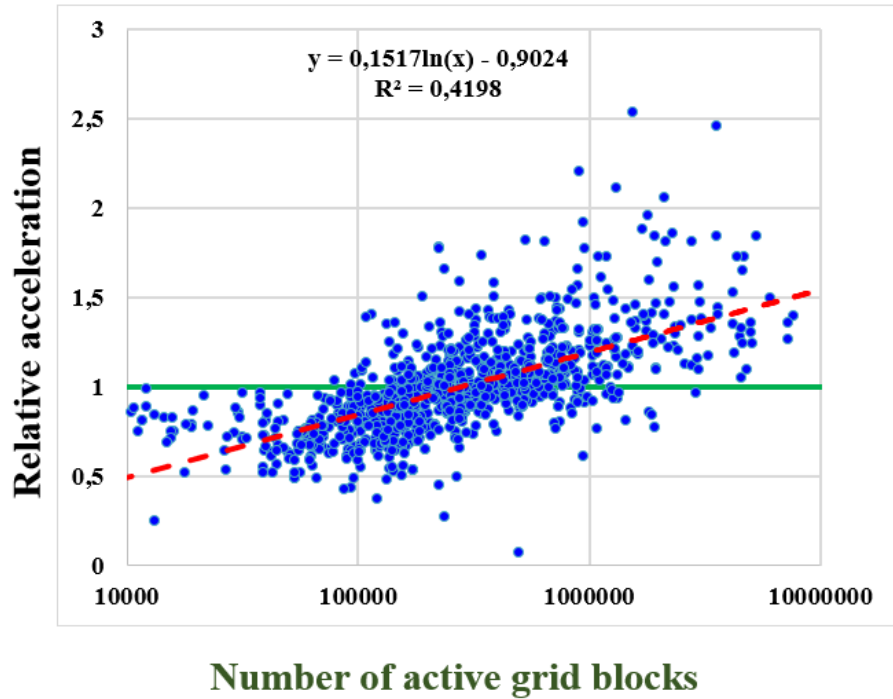


Figure 9. Dual Xeon 2680v2 -vs- Dual Xeon 2680v4. Price ratio (at the release date) ~ 1:1 if Y-value is higher than 1, then the latter platform is faster than the first.

### 937 black oil real field models



### 147 compositional real field models

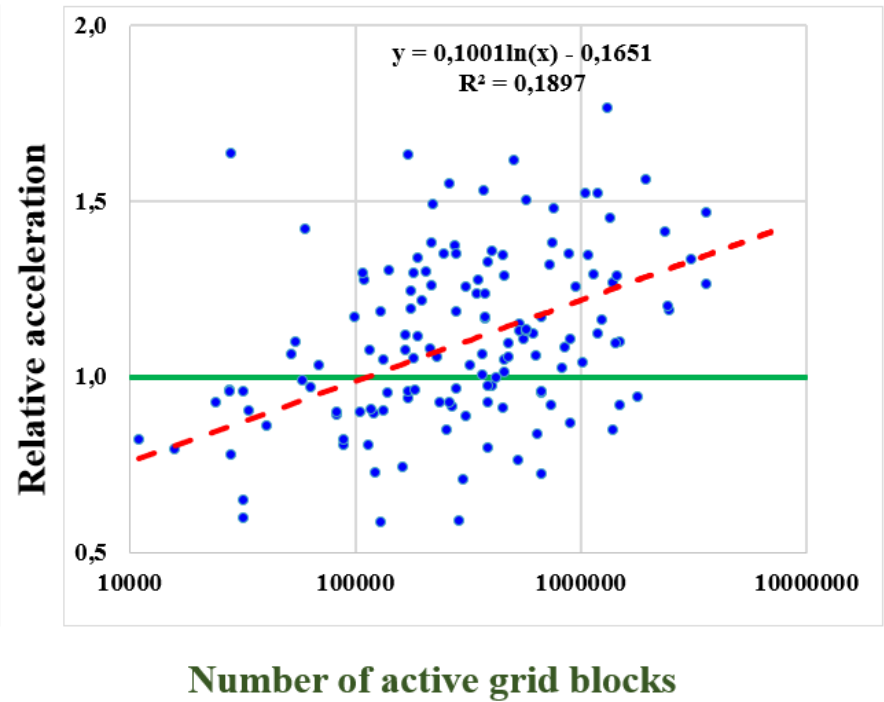
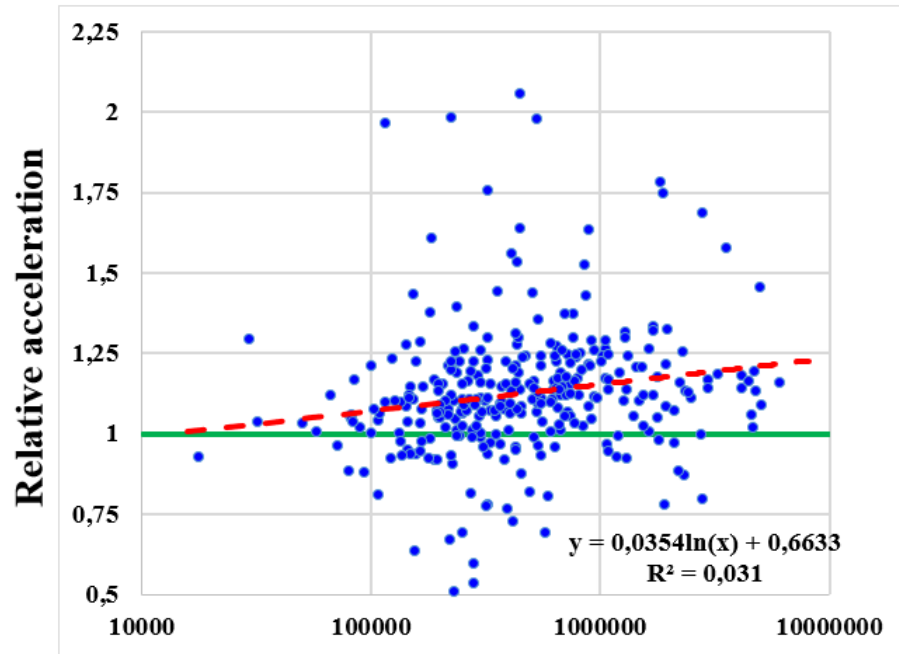


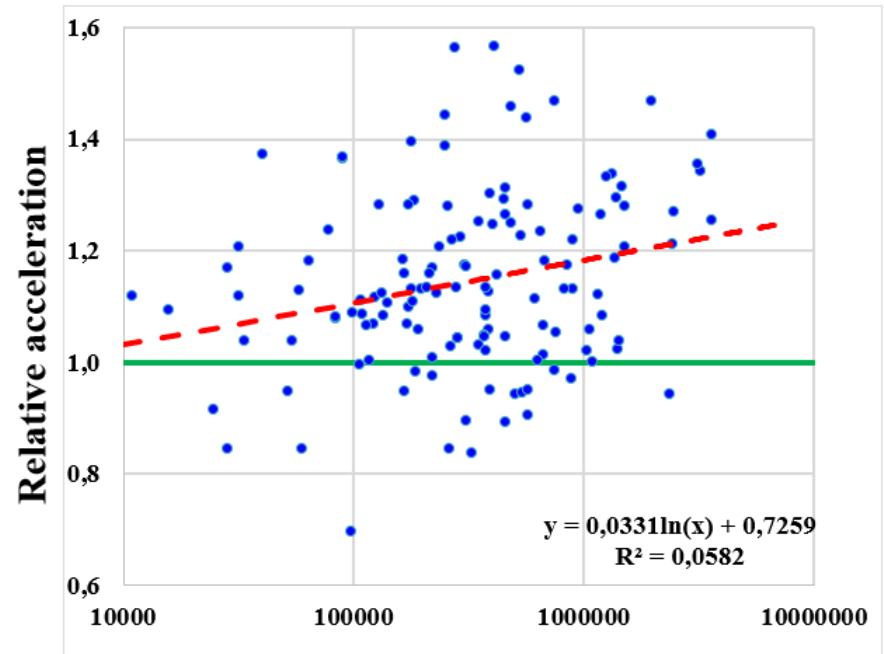
Figure 10. Dual Xeon 2680v4 -vs- Dual Xeon 2680v4 + Tesla P100. Price ratio (at the release date) ~ 1:2 if Y-value is higher than 1, then the latter platform is faster than the first.

### 316 black oil real field models



Number of active grid blocks

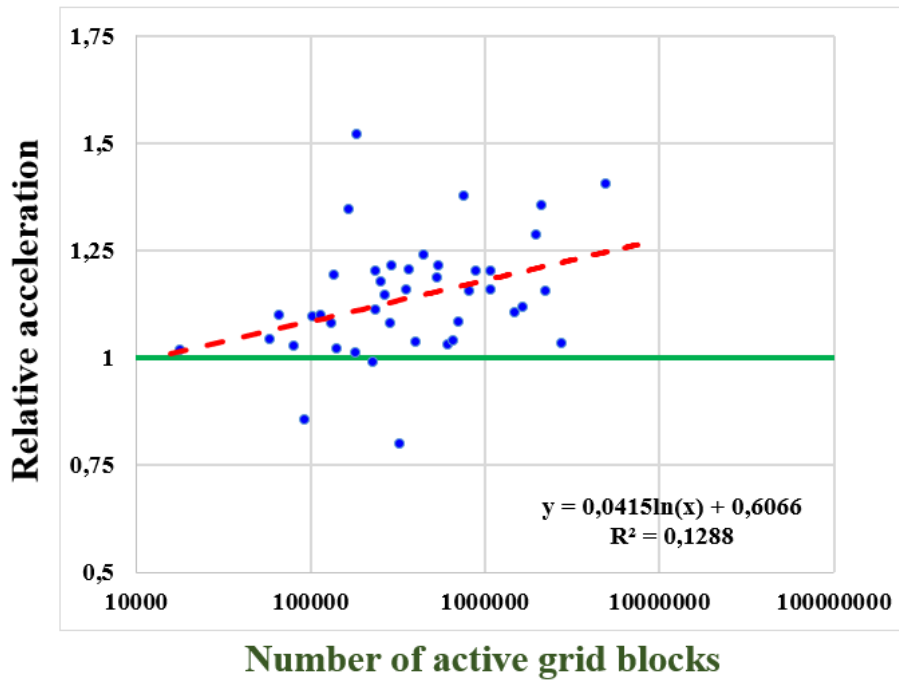
### 135 compositional real field models



Number of active grid blocks

Figure 11. Dual CPUs + Tesla P100 -vs- Dual CPUs + Dual Tesla P100. Price ratio (at the release date) ~ 1:1.5 if Y-value is higher than 1, then the latter platform is faster than the first.

### 42 black oil real field models



### 133 compositional real field models

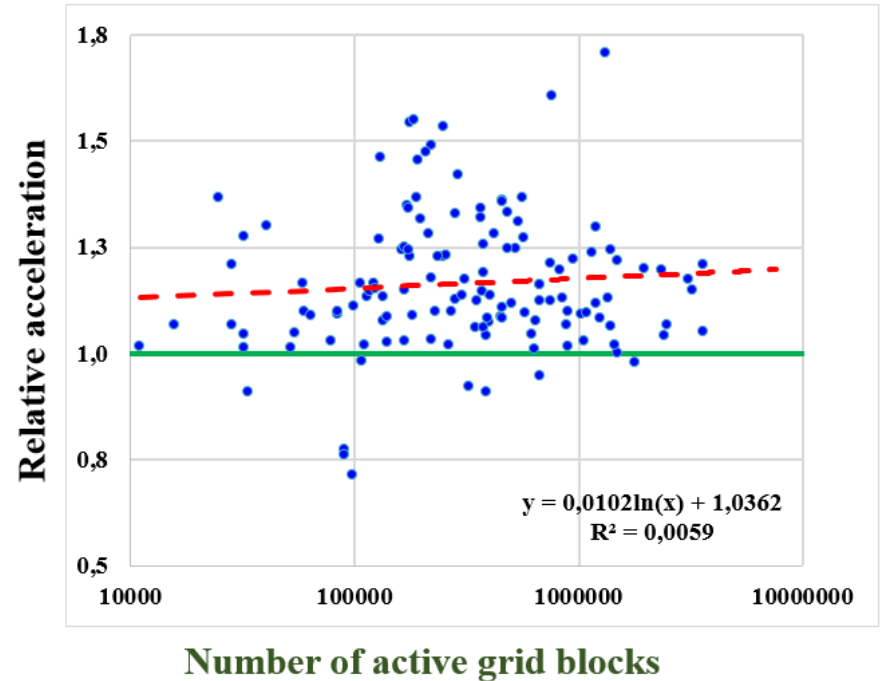
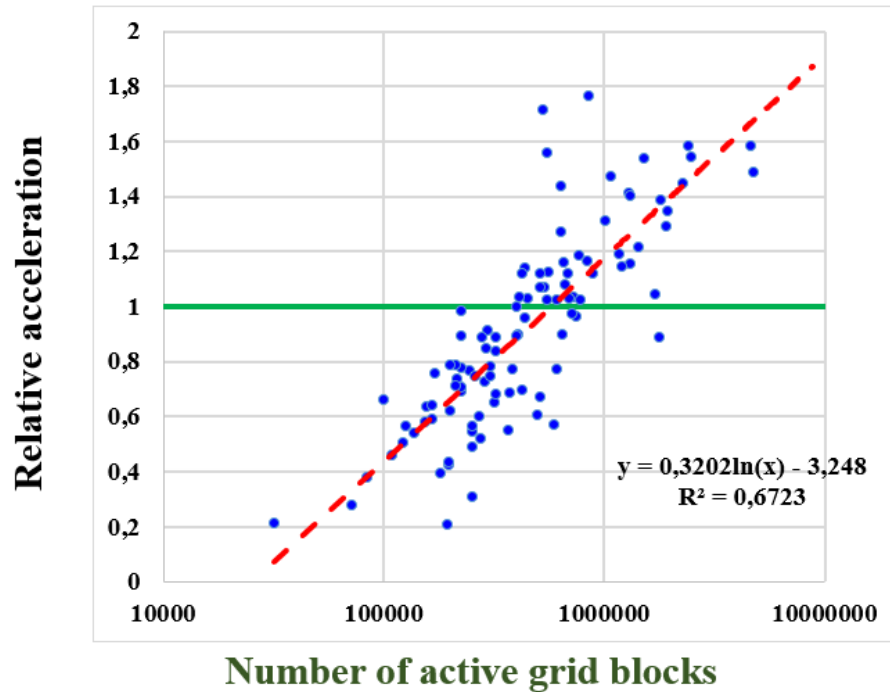


Figure 12. Tesla P100 -vs- Tesla V100. Price ratio (at the release date) ~ 1:1 if Y-vale is higher than 1, then the latter platform is faster than the first.

### 144 black oil real field models



### 103 compositional real field models

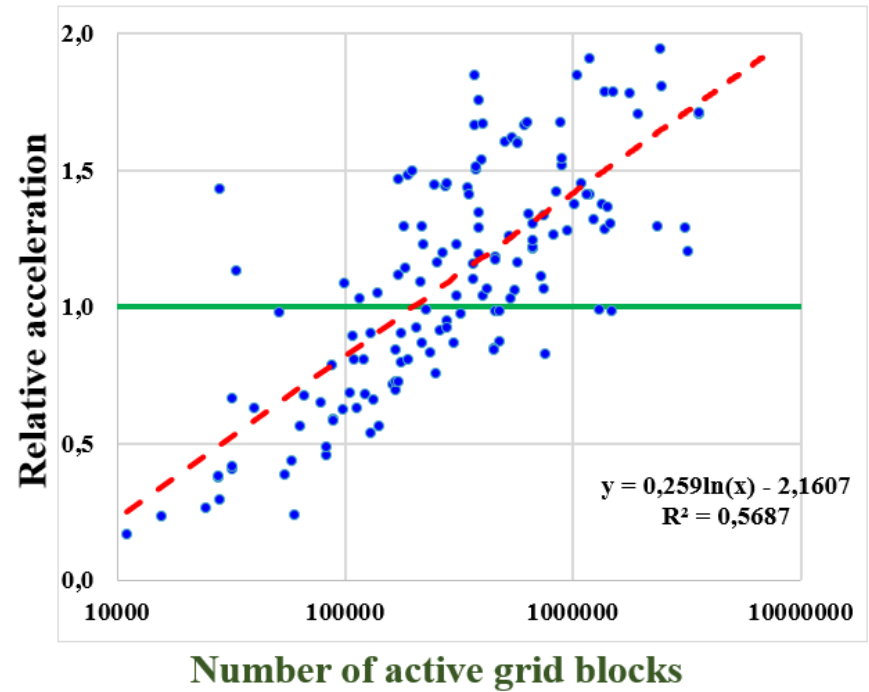


Figure 13. Dual Xeon 2680v4 -vs- Dual AMD EPYC 7601. Price ratio (at the release date) ~ 1:2 if Y-value is higher than 1, then the latter platform is faster than the first.

Device	Cores	Clock	Memory	Size	Clock	Bandwidth	Price
2x Xeon E5 2630 v3	16	2400MHz	DDR4	768GB	1866MHz	2x 56GB/s	\$1340
2x Xeon E5 2680 v4	28	2600MHz	DDR4	1.54TB	2400MHz	2x 59GB/s	\$3500
2x Xeon Platinum 8170 (Skylake)	52	2100MHz	DDR4	2.31TB	2666MHz	2x 119GB/s	\$15000
Xeon Phi 7210	64	1300MHz	HBM+ DDR4	16GB+ 384GB	1300MHz 2133MHz	450GB/s 102GB/s	\$2500
2x EPYC 7601	64	2200MHz	DDR4	4TB	2400MHz	2x 170GB/s	\$11000
GTX-1080	2560	1607MHz	GDDR5X	8GB	10000MHz	320GB/s	\$500
GTX-1080 TI	3584	1582MHz	GDDR5X	11GB	11000MHz	484GB/s	\$700
Tesla Quadro P6000	3840	1560MHz	GDDR5X	24GB	9000MHz	432GB/s	\$5500
Tesla P100	3584	1480MHz	HBM2	16GB	1400MHz	720GB/s	\$8900
Tesla V100	5120	1455MHz	HBM2	32GB	1750MHz	900GB/s	\$10000
Titan V	5120	1455MHz	HBM2	12GB	1700MHz	652GB/s	\$3000

Table 1. Modern CPU and GPU solutions for reservoir simulations.