WeWIPIC02

Full-GPU Reservoir Simulation Delivers on its Promise for Giant Carbonate Fields

A. Vidyasagar[1]*, L. Patacchini[1], P. Panfili[2], F. Caresani[2], A. Cominelli[2], R. Gandham[1], K. Mukundakrishnan[1]

[1] Stone Ridge Technology; [2] Eni

Summary

Simulation of carbonate fields presents challenges due to the underlying multi-scale heterogeneities and consequent stiff nature of the flow equations. This paper highlights the principles of a full-GPU (Graphics Processing Unit) reservoir simulator, currently approaching feature parity with traditional CPU-based codes. The approach exhibits fine-grained parallelism beyond that of CPU-based and hybrid CPU-GPU solutions; consequent performance improvements enable modeling of giant carbonate fields with limited computing resources. Additionally, large black-oil models are memory-bound, and GPU bandwidth has shown significant progress with every generational release of new hardware. Performance will keep improving without changes in the code base, which has not been observed with CPU codes in almost two decades.

Computational performance of a full-GPU black-oil reservoir simulator is benchmarked against legacy and modern parallel CPU simulators, for two giant gas and oil carbonate reservoirs. Results for the gas reservoir indicate a ~7.3x chip-to-chip speed improvement (one GPU vs. to 16 CPU cores), and ~5.5x for the oil reservoir, both against the fastest reference simulator. These results suggest that full-GPU codes are ready to simulate complex carbonate models of commercial grade, with exceptional performance, which should encourage the industry to pursue research and development efforts geared towards this approach.

Introduction

The fundamental principles of reservoir simulation and their attending numerical methodologies are geology-agnostic. Simulation of carbonate fields is nevertheless particularly challenging for two reasons. First, they typically exhibit strong heterogeneities at different scales, in addition to often being fractured; as a consequence, flow equations are stiff and require sophisticated solution methods. Second, they are difficult to characterize, both in terms of distributing rock properties and developing saturation-height models; as a consequence, carbonate models are particularly difficult to history-match, and uncertainties are best managed through compute-heavy approaches involving ensembles.

Improvement of reservoir simulator performance to tackle the above challenges can be achieved through two avenues. The first is to develop more efficient numerical formulations or solution algorithms; as an example, one can think of the adoption of AMG (Algebraic Multi Grid) as the pressure preconditioner, which has revolutionized the simulation of heterogeneous models (Cao *et al.*, 2005; Esler *et al.*, 2012). The second is to ride the wave of increasing hardware performance with limited code modifications; this served the discipline well until the early 2000s, after which CPU clock-speed started to stagnate, and reservoir simulators transitioned to multi-core CPUs and multi-node architectures. So-called 'next generation' simulators were born in the first half of that decade with native support for coarse parallelism, essentially based on domain decomposition.

Reservoir simulation, in particular for black-oil models, is memory bandwidth limited; this means that the bottleneck is not so much the speed of computational units, but how fast they can be fed with data. While the bandwidth of multi-core CPUs showed modest progress over the past decade (reaching ~100 GB/s today), GPU (Graphical Processing Unit) bandwidth keeps improving with each new generation of hardware (reaching ~900 GB/s for the NVIDIA V100). In the vast majority of published results, GPUs are introduced to reservoir simulation in the form of hybrid CPU-GPU codes, where only specific parts of the computation are ported to GPUs (typically the linear solver), while the rest (property calculations, Jacobian assembly) stays on the CPUs.

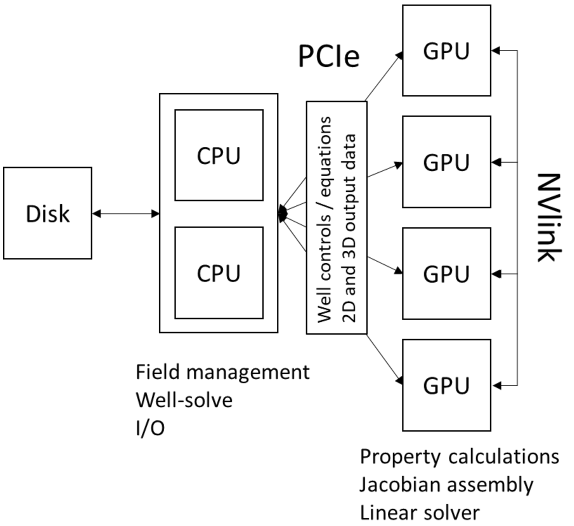
Esler *et al.* (2014) proposed an approach where all reservoir calculations are performed on the GPU, overcoming the limitations of hybrid approaches. Performance benchmarks with respect to a reference 'legacy' reservoir simulator were provided, suggesting speed-ups in the order of 10x to 100x (16 CPU cores vs. a modern GPU) on models with simple field management functionalities.

This paper demonstrates that full-GPU simulators can incorporate all the physical features available in modern CPU simulators, enabling simulation of giant carbonate fields. Performance benchmarks against a reference 'next generation' CPU simulator shows speed-ups in the order of more than 5x on a socket to socket basis (modern 16-core CPU vs. modern GPU), consistent with the bandwidth gap between CPUs and GPUs. Our objective is therefore to encourage the community to pursue research and development efforts geared towards the full-GPU solution (simply referred to as 'GPU' solution hereafter).

GPU vs. hybrid CPU-GPU solution

The principles of a GPU simulator are illustrated in Figure 1. All reservoir-related calculations are performed on the GPUs, while CPUs are only used for I/O, field management and well solves. This approach avoids lengthy data transfers between CPU and GPUs required when only the linear solver is ported to the GPU. The often-heard statement that 'some simulator components may not be suitable for porting to the GPGPU' (Fung *et al.*, 2014) holds, in our view, for the well-solve only.

The GPU solution has three key advantages. First, it is faster because memory bandwidth is faster on GPU than on CPU. It is more scalable because GPU domains are typically one to two orders of magnitude larger than those used by multi-core codes; this keeps more of the calculation on a single chip and minimizes costly inter-domain communication. Finally, it requires very compact hardware; a single GPU node has an aggregate bandwidth corresponding to that of 40 standard CPU nodes.



Element	Approx. bandwidth (GB/s)
GPU (V100)	900
CPU (modern multicore)	100
NVlink	300
PCIe3	15

Figure 1 *(left) Repartition of roles between CPUs and GPUs in a full-GPU reservoir simulator, for a single node with 2 CPUs and 4 GPUs. The approach can scale up to multiple nodes using MPI, trough network connection. (right) Memory bandwidth of different elements.*

Figure 2 shows that the performance of a GPU simulator is linearly correlated to the available memory bandwidth. Without changing the code, significant speed-up is observed every two to three years as a new GPU generation becomes available.

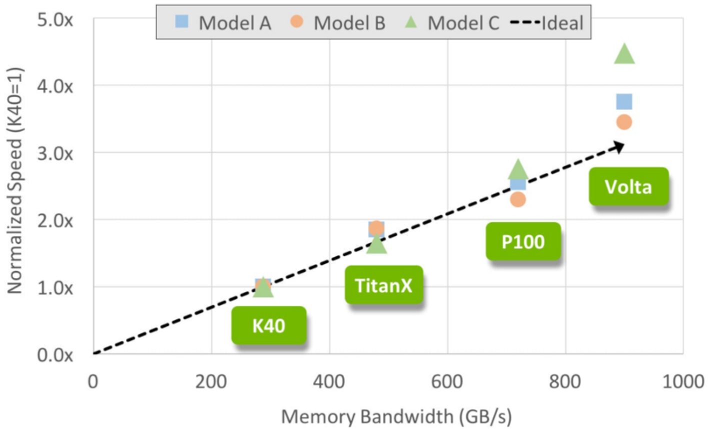


Figure 2 *Running three real models with a full-GPU black-oil simulator on different generations of hardware indicate that performance is proportional to the GPU bandwidth. (The slight supralinear performance increase observed with the Volta is due to a better memory controller and improved instruction sets which allow to operate closer to peak bandwidth.)*

The hybrid CPU-GPU solution has fundamental limitations. First, the maximum achievable speedup is dictated by the parts of a code that are not accelerated (Amdhal's law). For example, if the linear solver takes 50% of simulation time, and porting it to GPU hypothetically reduces its cost to zero, the overall speed up is only 2x; far from the almost 10x bandwidth advantage offered by GPUs (the curve in Figure 2 would bend downwards). Second, it does not allow for reduction in hardware footprint and cost; all the CPUs required by the CPU-only solution are still needed.

While the hybrid approach has been extensively studied academically and implemented in proprietary simulators (Fung *et al.*, 2014), to the best of our knowledge, there exists only one commercial simulator adopting this strategy (Telishev *et al.*, 2017; Bogachev *et al.*, 2018). Unfortunately, because an AMG pressure solver is not used, it is difficult to clearly assess how much of the performance benefit from the proposed GPU solution comes from the solver, in particular for heterogeneous models. For example, the authors report an elapsed time of 11min. 20s. for the SPE10 benchmark problem on a laptop with quad-core CPU and NVIDIA GTX 1080, while an elapsed time of 1min. 12s (~10x speed-up) can be achieved with the proposed GPU solution on a similar laptop with an older GPU (GTX 1060) and a timing of 23s is achievable on two NVIDIA V100. The proposed GPU approach is therefore benchmarked versus either a reference 'modern' or reference 'legacy' CPU-only simulator.

Example #1: Giant gas field

The first application example is a Mediterranean sea, dry gas deep-water carbonate reservoir, with 3.5M active cells. The simulation runs for 43 years (history+forecast), with a total of 25 producing wells (Figure 3).

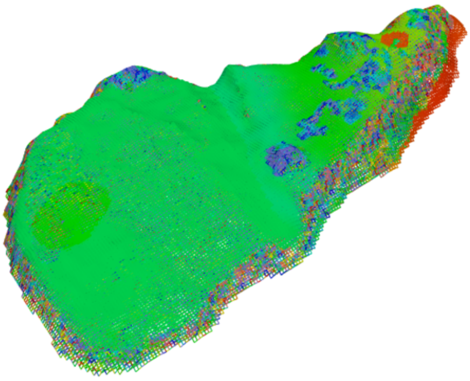


Figure 3 *Permeability field of the giant gas field model example.*

Performance is compared with respect to a 'modern' CPU-based simulator in Table 1. On a chip-to-chip basis, the GPU solution is about 7x faster, and requires a comparable amount of nonlinear and linear iterations. Scalability to multiple GPUs is in the order of 1.2x to 1.3x, which in this case is not as high as observed in other test cases (usually in the order of 1.5x); this is due to using a specific linear solver preconditioner adapted to high vertical transmissibility models, on which research is still ongoing.

	#Cores	#GPUs	Elapsed Time (min)	Nonlinear Iterations	Linear Iterations	Speedup	Scalability
Modern CPU-based	16	-	219	4,615	19,660	-	-
GPU	-	1	30	3,276	19,419	7.3x	-
GPU	-	2	23	3,545	23,388	9.5x	1.3x
GPU	-	4	19	3,577	26,591	11.5x	1.2x

Table 1 *Performance benchmark of the full-GPU simulator wrt. a modern CPU-based simulator on the Giant gas field case. Equivalent convergence criteria have been set for both simulators.*

Example #2: Giant oil field

The second application example is a Middle-Eastern carbonate oil reservoir, with 6.2M active cells, ~80 producers and ~60 water injectors, ~50 years of history and ~50 years of forecast.

Performance is compared with respect to a 'legacy' CPU-based simulator in Table 2, because in this case it was faster than the reference modern simulator (it completes the simulation in less nonlinear iterations than both the modern CPU-based and the GPU simulators). Work is ongoing to improve nonlinear convergence for this model; nevertheless, speedup on a chip-to-chip basis is more than 5x, and scalability to multiple GPUs is in the order of 1.5x.

	#Cores	#GPUs	Elapsed Time (min)	Nonlinear Iterations	Linear Iterations	Speedup	Scalability
Legacy CPU-based	16	-	938	7,000	35,611	-	-
GPU	-	1	169	17,822	83,722	5.5x	-
GPU	-	2	106	18,154	87,164	8.8x	1.6x
GPU	-	4	70	17,389	84,311	13.4x	1.5x

Figure 2 *Performance benchmark of the GPU simulator wrt. a legacy CPU-based simulator on the Giant oil field case (here, the legacy simulator was faster than the modern CPU-based simulator).*

Conclusions

Using two giant carbonate fields as examples, it has been shown that the breadth of options available in CPU-based reservoir simulators, needed to run models of practical commercial interest, can be incorporated in a full-GPU-simulator. The models show a speed-up of more than 5x over a modern CPU-based simulator, which is significant enough to encourage the community to pursue research and development efforts geared towards the GPU solution.

The results shown here indicate that there is a margin to improve performance further through both algorithmic and hardware advances. The former requires improved nonlinear update heuristics and additional fine tuning strategies; the latter will come with each new generation of GPU hardware.

Acknowledgements

The authors would like to thank Eni for permission to publish this work.

References

Bogachev, K., Milyutin, S., Telishev, A., Nazarov, V., Shelkov, V., Eydinov, D., Malinur, O., Hinneh, S. [2018] High-Performance Reservoir Simulations on Modern CPU-GPU Computational Platforms. ICE 2018.

Cao, H., Tchelepi, H.A., Wallis, J. [2005] Parallel Scalable Unstructured CPR-Type Linear Solver for Reservoir Simulation. SPE-96809-MS

Christie, M.A. and Blunt, M.J. [2001] Tenth SPE comparative solution project: A comparison of upscaling techniques. SPE-66599-MS.

Esler, K., Natoli, V., Samardzic, A. [2012] GAMPACK (GPU Accelerated Algebraic Multigrid Package). ECMOR XIII B37.

Esler, K., Mukundakrishnan, K., Natoli, V., Shumway, J., Zhang, Y., Gilman, J. [2014] Realizing the Potential of GPUs for Reservoir Simulation. ECMOR XIV, Mo A05.

Fung, L.S.K., Sindi, M.O. and Dogru, A.H. [2014] Multi-paradigm parallel acceleration for reservoir simulation. SPEJ-163591.

Telishev, A., Bogachev, K., Shelkov, V., Eydinov, D., Tran H. [2017] Hybrid Approach to Reservoir Modeling Based on Modern CPU and GPU Computational Platforms. SPE-187797-MS.