

Serverless seismic imaging in the cloud

1 st Philipp A. Witte College of Computing Georgia Inst. of Technology Atlanta, USA pwwitte3@gatech.edu	2 nd Mathias Louboutin College of Computing Georgia Inst. of Technology Atlanta, USA mlouboutin3@gatech.edu	3 rd Charles Jones Osokey Ltd. Henley-on-Thames, UK charles@osokey.com	4 th Felix J. Herrmann College of Computing Georgia Inst. of Technology Atlanta, USA felix.herrmann@gatech.edu
--	--	--	---

Abstract—This talk presents a serverless approach to seismic imaging in the cloud based on high-throughput containerized batch processing, event-driven computations and a domain-specific language compiler for solving the underlying wave equations. A 3D case study on Azure demonstrates that this approach allows reducing the operating cost of up to a factor of 6, making the cloud a viable alternative to on-premise HPC clusters for seismic imaging.

Index Terms—reverse-time migration; cloud; serverless

I. AUDIENCE

The content of this abstract is intended as a technical talk for an audience with little to no prior knowledge about cloud computing, but basic knowledge about reverse-time migration and HPC. This talk is designed for people who are interested in how the cloud can be adapted for large-scale seismic imaging and inversion in both research and production environments.

II. INTRODUCTION

Seismic imaging and parameter estimation are among the most computationally challenging problems in scientific computing and thus require access to high-performance computing (HPC) clusters for working on relevant problem sizes as encountered in today’s oil and gas (O&G) industry. Some companies such as BP, PGS and Exxon Mobile operate private HPC clusters with maximum achievable performance in the order of petaflops [1], [2], while some companies are even moving towards exascale computing [3]. However, the high upfront and maintenance cost of on-premise HPC clusters make this option only financially viable in a production environment where computational resources constantly operate close to maximum capacity. Many small and medium sized O&G companies, academic institutions and service companies have a highly varying demand for access to compute and/or are financially not

in a position to purchase on-premise HPC resources. Furthermore, researchers in seismic inverse problems and machine learning oftentimes require access to a variety of application-dependent hardware, such as graphical processing units (GPUs) or memory optimized compute nodes for reverse-time migration (RTM).

Cloud computing thus offers a valuable alternative to on-premise HPC clusters, as it provides a large variety of (theoretically) unlimited computational resources without any upfront cost. Access to resources in the cloud is based on a pay-as-you-go pricing model, making it ideal for providing temporary access to compute or for supplementing on-premise HPC resources to meet short-term increases in computing demands. However, some fundamental differences regarding hardware and how computational resources are exposed to users exist between on-premise HPC clusters and the cloud. While cloud providers are increasingly investing in HPC technology, the majority of existing hardware is not HPC optimized and networks are conventionally based on Ethernet. Additionally, the pay-as-you-go pricing model is a usage-based system, which means users are constantly charged for running instances. This possibly results in very high operating costs if instances sit idle for extended amounts of time, which is common in standard RTM workflows based on a client-server model in which the master process distributes the workload to the parallel workers. In this work, we demonstrate a serverless approach to seismic imaging on Microsoft Azure, which does not rely on a cluster of permanently running virtual machines (VMs). Instead, expensive compute instances are automatically launched and scaled by the cloud environment, thus preventing instances from sitting idle. For solving the underlying forward and adjoint wave equations, we use a domain-specific language compiler called *Devito* [4], [5], which combines a symbolic user interface with automated performance optimization for generating fast and parallel C code using just-in-time

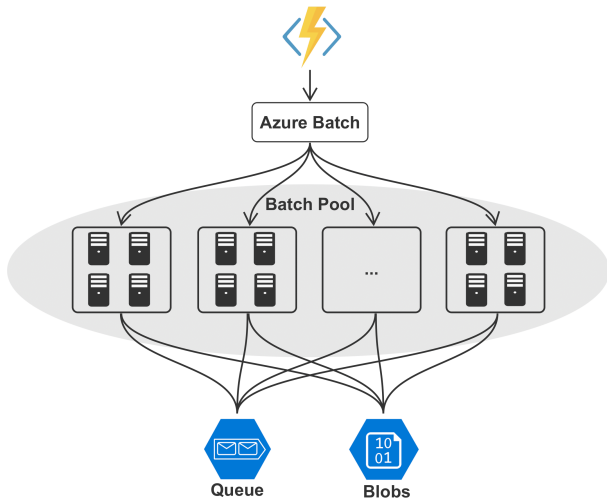


Fig. 1: Azure setup for computing an RTM image as a containerized embarrassingly parallel workload.

compilation. The separation of concerns between the wave equation solver and the serverless workflow implementation leads to a seismic imaging framework that scales to large-scale problem sizes and allows reducing the operating cost in the cloud up to a factor of 2 – 6, as demonstrated in our subsequent RTM case study.

III. CURRENT STATE OF THE ART

The cloud is increasingly adopted by O&G companies for general purpose computing, marketing, data storage and analysis [6], but utilizing the cloud for HPC applications such as (least-squares) RTM and full-waveform inversion (FWI) remains challenging. A wide range of performance benchmarks on various cloud platforms find that the cloud can generally not provide the same performance in terms of latency, bandwidth and resilience as conventional on-premise HPC clusters [7]–[9], or only at considerable cost. Recently, cloud providers such as Amazon Web Services (AWS) or Azure have increasingly extended their HPC capabilities and improved their networks [10], but HPC instances are oftentimes considerably more expensive than standard cloud VMs [10]. On the other hand, the cloud offers a range of novel technologies such as massively parallel objective storage, containerized batch computing and event-driven computations that allow addressing computational bottlenecks in novel ways. Adapting these technologies requires restructuring seismic inversion codes, rather than running legacy codes on a virtual cluster of permanently running cloud instances (*lift and shift*). Companies that have taken steps towards the development of cloud-native technology include S-Cube, whose FWI workflow for

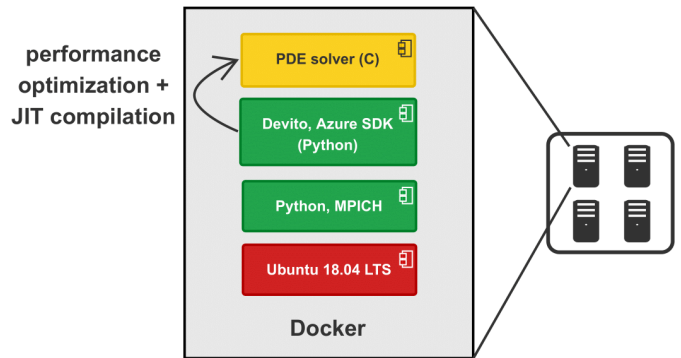


Fig. 2: Software stack of the Docker container for computing the RTM images.

AWS utilizes object storage, but is still based on a master-worker scheme [11]. Another example is Oskey [12], a company offering fully cloud-native and serverless software for seismic data visualization and interpretation. In a previous publication, we have adapted these concepts for seismic imaging and introduced a fully cloud-native workflow for serverless imaging on AWS [13]. Here, we describe the implementation of this approach on Azure and present a 3D imaging case study.

IV. METHODS AND KEY RESULTS

A. Workflow

The key contribution of this talk is a serverless implementation of an RTM workflow on Azure. The two main steps of a (generic) RTM workflow are the parallel computation of individual images for each source location and the subsequent summation of all components into a single image, which can be interpreted as an instance of a MapReduce program [14]. Rather than running RTM on a cluster of permanently running VMs, we utilize a combination a high-throughput batch processing and event-driven computations to compute images for separate source locations as an embarrassingly parallel workflow (Figure 1 and 2). The parallel computation of RTM images for separate source locations is implemented with Azure Batch, a service for scheduling and running containerized workloads. The image of each respective source location is processed by Azure Batch as a separate job, each of which can be executed on a single or multiple VMs (i.e. using MPI-based domain decomposition). Azure Batch accesses computational resources from a batch pool and automatically adds and removes VMs from the pool based on the number of pending jobs, thus mitigating idle instances. The software for solving the underlying forward and adjoint wave equations is

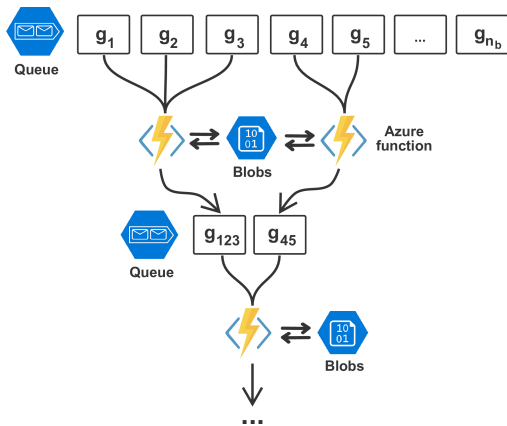


Fig. 3: Event-driven gradient summation on Azure, using Azure Functions and Queue Storage.

deployed to the batch workers through Docker containers and Devito’s compiler automatically performs a series of performance optimization to generate optimized C code for solving the PDEs (Figure 2).

As communication between individual jobs is not possible, we separately implement the reduce part of RTM (i.e. the summation of all images into a single data cube) using Azure functions. These event-driven functions are automatically invoked when a batch workers writes its computed image to the object storage system (blob storage) and sends the corresponding object identifier to a message queue, which collects the IDs of all results (Figure 3). As soon as object IDs are added to the queue, Azure functions that sum up to 10 images from the queue are automatically invoked by the cloud environment. Each function writes its summed image back to the storage and the process is repeated recursively until all images have been summed into a single volume. As such, the summation process is both asynchronous and parallel, as the summation is started as soon as the first images are available and multiple Azure functions can be invoked at the same time.

B. RTM Case study

For our 3D RTM case study on Azure, we use a synthetic velocity model derived from the 3D SEG Salt and Overthrust models, with dimensions of $3.325 \times 10 \times 10$ km. We discretize the model using a 12.5 m grid, which results in $267 \times 801 \times 801$ grid points. We generate data at 15 Hz peak frequency for a randomized seismic acquisition geometry, with data being recorded by 1,500 receivers that are randomly distributed along the ocean floor. The source vessel fires the seismic source on a dense regular grid, consisting of 799×799 source

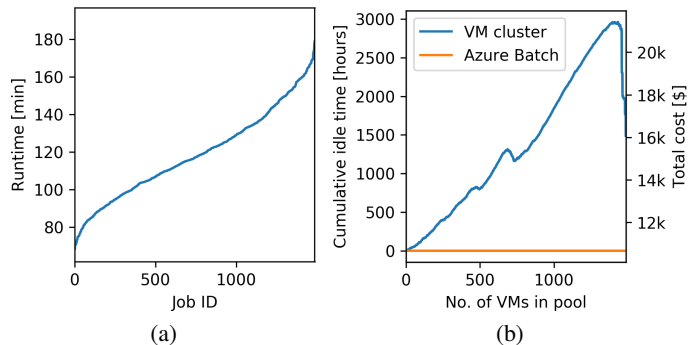


Fig. 4: (a) Sorted runtimes for computing the RTM image. Each job corresponds to the image of a single source location. (b) Cumulative idle time for computing this workload as a function of the number of parallel workers on either a fixed cluster of VMs or using Azure Batch. The right-hand y-axis shows the corresponding cost, which is proportional to the idle time.

locations (638,401 in total). For imaging, we assume source-receiver reciprocity, which means that sources and receivers are interchangeable and data can be sorted into 1,500 shot records with 638,401 receivers each. We model wave propagation for generating the seismic data with an anisotropic pseudo-acoustic TTI wave equation and implement discretized versions of the forward and adjoint (linearized) equations with Devito, as presented in [15].

For the computations, we use Azure’s memory optimized E64 and E64s VMs, which have 432 GB of memory, 64 vCPUs and a 2.3 GHz Intel Xeon E5–2673 processor [10]. To fit the forward wavefields in memory, we utilize two VMs per source and use MPI-based domain decomposition for solving the wave equations. The time-to-solution of each individual image as a function of the source location is plotted in Figure 4a, with the average container runtime being 119.28 minutes per image. The on-demand price of the E64/E64s instances is 3.629\$ per hour, which results in a cumulative cost of 10,750\$ for the full experiment and a total runtime of approximately 30 hours using 100 VMs. Figure 4b, shows the cumulative idle time for computing the workload from Figure 4a on a fixed cluster as a function of the number of parallel VMs. We model the idle time by assuming that all 1,500 jobs (one job per source location) are distributed to the parallel workers on a first-come-first-serve basis and that a master worker collects all results. The idle time using a fixed VM cluster results from the fact that all cluster workers have to wait for the final worker to finish its computations, while

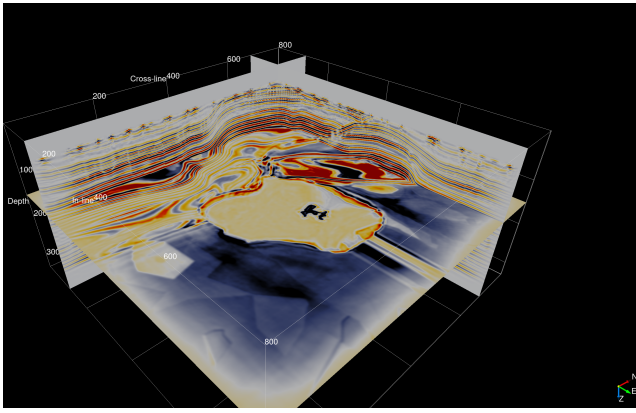


Fig. 5: Horizontal depth slice through the final 3D image cube at 1,500 m depth.

Azure Batch automatically scales down the cluster, thus preventing instances from sitting idle. While VM clusters on Azure in principle support auto-scaling as well, this is not possible if MPI is used to distribute the data/sources to the workers. Thus, performing RTM on a fixed cluster of VMs results in additional costs due to idle time up to a factor of $2\times$. By utilizing low-priority instances, it is possible to further reduce the operating cost by a factor of $2 - 3$ (i.e. up to a factor 6 in total).

V. CONCLUSION

Adapting the cloud using serverless workflows, in contrast to lift and shift, allows us to leverage cloud services such as batch computing and reduce the operating cost of RTM by a factor of $2 - 6$. This transition is made possible through abstract user interfaces and an automatic code generation framework, which is highly flexible, but provides the necessary performance to work on industry-scale problems.

VI. PRESENTER BIO

P. Witte, M. Louboutin and F. J. Herrmann are part of the core Devito development team and have closely collaborated with multiple cloud companies to develop serverless implementations of RTM. They believe that the computational cost and complexity of seismic inversion can only be managed through abstractions, automatic code generation and software based on a separation of concerns. C. Jones is the head of development at Osokey, a company specialized in cloud-native software for seismic data analysis.

ACKNOWLEDGEMENTS

We would like to acknowledge S. Brandsberg-Dahl, A. Morris, K. Umay, H. Shel, S. Roach and E. Burness (all

with Microsoft Azure) for collaborating with us on this project. Many thanks also to H. Modzelewski (University of British Columbia) and J. Selvage (Osokey). This research was funded by the Georgia Research Alliance.

REFERENCES

- [1] "Seismic processing and imaging," 2019. [Online]. Available: <https://www.pgs.com/imaging/services/processing-and-imaging/>
- [2] "A close-up look at the world's largest HPC system for commercial research," <https://www.hpcwire.com/2018/01/14/close-look-worlds-largest-hpc-system-commercial-research/>, 2019.
- [3] "DUG McCloud," <https://dug.com/dug-mccloud/>, 2019.
- [4] M. Louboutin, M. Lange, F. Luporini, N. Kukreja, P. A. Witte, F. J. Herrmann, P. Velesko, and G. J. Gorman, "Devito (v3.1.0): an embedded domain-specific language for finite differences and geophysical exploration," *Geoscientific Model Development*, vol. 12, no. 3, pp. 1165–1187, 2019.
- [5] F. Luporini, M. Lange, M. Louboutin, N. Kukreja, J. Hückelheim, C. Yount, P. A. Witte, P. H. J. Kelly, G. J. Gorman, and F. J. Herrmann, "Architecture and performance of Devito, a system for automated stencil computation," <https://arxiv.org/abs/1807.03032>, 2018, computing Research Repository (arXiv CoRR).
- [6] "AWS enterprise customer success stories," 2019. [Online]. Available: <https://aws.amazon.com/solutions/case-studies/enterprise>
- [7] L. Ramakrishnan, R. S. Canon, K. Muriki, I. Sakrejda, and N. J. Wright, "Evaluating interconnect and virtualization performance for high performance computing," *Association for Computing Machinery (ACM) SIGMETRICS Performance Evaluation Review*, vol. 40, no. 2, pp. 55–60, 2012.
- [8] S. Benedict, "Performance issues and performance analysis tools for HPC cloud applications: a survey," *Computing*, vol. 95, no. 2, pp. 89–108, 2013.
- [9] P. Mehrotra, J. Djomehri, S. Heistand, R. Hood, H. Jin, A. Lazanoff, S. Saini, and R. Biswas, "Performance evaluation of Amazon Elastic Compute Cloud for NASA high-performance computing applications," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 4, pp. 1041–1055, 2016.
- [10] "Linux virtual machines pricing," 2019. [Online]. Available: <https://azure.microsoft.com/en-us/pricing/details/virtual-machines/linux/>
- [11] "XWI on AWS: Revolutionary earth model building on the cloud," <https://www.s-cube.com/xwi-on-the-cloud/>, 2019.
- [12] S. Ashley, A. Gray, C. Jones, J. Nicholson, J. Selvage, M. So-hail, and K. Stewart, "Sharing learnings: The methodology, optimisation and benefits of moving subsurface data to the public cloud," in *81st Annual International Meeting, EAGE, Expanded Abstracts*, 2019.
- [13] P. A. Witte, M. Louboutin, H. Modzelewski, C. Jones, J. Selvage, and F. J. Herrmann, "Event-driven workflows for large-scale seismic imaging in the cloud," in *89th Annual International Meeting, SEG, Expanded Abstracts*, 2019, pp. 3984–3988.
- [14] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Communications of the Association for Computing Machinery (ACM)*, vol. 51, no. 1, pp. 107–113, 2008.
- [15] M. Louboutin, P. Witte, and F. J. Herrmann, "Effects of wrong adjoints for RTM in TTI media," in *88th Annual International Meeting, SEG, Expanded Abstracts*, 2018, pp. 331–335.