Seismic Inversion on Federated Kubernetes Clusters

M. Griffiths[1], R. Abdul Alim[1], J. Ramos-Martinez[1], N. Chemingui[1], R.G. Clapp[2]

[1] PGS; [2] Stanford University

Summary

We describe a cloud-native, fully fault-tolerant design for High-Performance Compute (HPC) seismic imaging algorithms, specifically optimized for execution across multiple Kubernetes clusters in the cloud. We use the simultaneous inversion of velocity and pre-stack reflectivity software as an illustration and demonstrate the compute and IO scalability of a high-frequency seismic inversion within a compressed timeframe. Our cloud-native strategy paves the way for advanced seismic imaging and inversion that deliver high-quality earth models in a cost-effective manner while reducing the cycle time of production projects.

Seismic Inversion on Federated Kubernetes Clusters

Introduction

Seismic High-Performance Computing (HPC) applications are often embarrassingly parallel. Trace-based or ensemble-based algorithms can be distributed horizontally across multiple compute instances or vertically across multiple CPU threads to decrease computational elapsed time. Techniques employing OpenMP and MPI have been implemented since the mid 1990s, as seen in works like Amundsen *et al*. (1996) and Ober *et al*. (1997). The horizontal scaling of applications has historically faced limitations due to the availability of compute resources in local dedicated centers and performance constraints of I/O subsystems. In recent years, Cloud based HPC has established itself as an alternative to traditional, privately owned HPC clusters with major cloud service companies offering Infrastructure as a Service (IaaS) as a part of their business model. Correctly leveraged, IaaS can offer an affordable alternative to the high costs of building, maintaining, and upgrading private clusters, while facilitating on-demand HPC growth and automatic rescaling of corporate HPC footprints. Work by Okita *et al.* (2019), Witte *et al.* (2019), and Debens *et al.* (2022), highlight previous attempts to build and cost optimize seismic processing algorithms across various cloud-based platforms.
In this paper we describe how we have engineered our HPC algorithms to utilize cloud-based computing resources using Kubernetes as the cloud-native approach to distributed processing. We show a successful example of performing seismic inversion in the cloud, highlighting the intense computational needs of the process and the sophisticated architecting of the algorithm.

Kubernetes as a platform

To implement HPC applications on the cloud, Kubernetes provides a streamlined and scalable solution. Serving as a robust orchestration platform, Kubernetes facilitates the deployment, scaling, and management of parallel applications within containerized environments. It enables automatic resource provisioning, dynamic horizontal scaling, and load balancing through concurrent pod execution on single virtual machines (VMs). Additionally, Kubernetes ensures automatic self-healing in response to errors, preemption events, or resource pressures. The highly dynamic nature of Kubernetes clusters contrasts with the strictly static nature of traditional MPI clusters, making direct application ports and seamless migration between the two environments a non-trivial challenge. Additionally, the cloud offers massively parallel object storage for IO, event driven computation, and resource flexibility. For cost-efficient implementation it is necessary to incorporate cloud-native IO formats, to eliminate any inter-node communication dependencies, and to implement fault tolerance techniques for full preemption support.

HPC application - Simultaneous inversion of velocity and pre-stack reflectivity

Simultaneous inversion, described by the workflow in Figure 1, is a novel inversion solution for joint estimation of velocity and angle-dependent reflectivity. It is based on two fundamental elements: a wave equation parameterized in terms of velocity and vector reflectivity as a modeling engine, and an efficient scale separation of the inversion gradient via inverse scattering theory (Yang *et al*., 2021). The scale separation is key to minimize the crosstalk of the velocity and the vector reflectivity during the inversion. Vector reflectivity parameterization of the wave equation enables a fully data driven inversion not requiring density information. Most importantly, it provides insightful information to compute pre-stack image gathers for angle-dependent reflectivity. This is explained in detail by Chemingui *et al*. (2023) where the scattering angle can be computed from the gradient of the forward pressure wavefield and the vector reflectivity which provides information about the dip of the reflectors. The inversion workflow is accomplished by implementing individual shot forward modelling and adjoint modeling tasks that are computationally scaled horizontally, coupled with an accumulation and update stage computationally scaled vertically.

Implementation of Simultaneous inversion on Kubernetes Clusters

Seismic inversion algorithms can be divided into a series of sequential stages. For simultaneous inversion these stages consist of shot-based forward modelling, residual calculation, adjoint modelling or backward propagation of the residuals, accumulation of the individual shot gradients, and the updating step for each iteration. Broadly, this equates to three distinct phases with vastly different compute requirements.

Partitioning monolithic applications like simultaneous inversion into smaller compute elements and staging the workflow allows for the application to be executed across one or many task-specific node configurations. This partitioning provides a natural means of performing roll-back recovery in the event of pod eviction or preemption. Figure 2 shows a schematic of the cluster configuration and task partitioning used in the restructured application.

A key component of the redesigned HPC algorithms is the job orchestration layer. This is a low compute, low memory, application responsible for monitoring, distribution, and scheduling the main application's compute elements. A separate Redis server serves as an intermediary between the orchestration layer and the Kubernetes scheduler with Kubernetes being responsible for the compute node provisioning and the task execution.

Limitations in the VPC address space, capacity restrictions, and spot instance availability place an upper most limit on the eventual Kubernetes cluster size in terms of total core count. For very large production jobs involving tens to hundreds of thousands of individual compute tasks it becomes beneficial to distribute the work across multiple Kubernetes clusters.

The automatic self-healing, restarting, and load balancing functionality of Kubernetes, even in the absence of preemption, necessitates the addition of fault tolerance support at all stages of the application including the Redis server and the orchestration layer. The compute intensive tasks require fault tolerance for preemption support.

Application metadata, which includes information on the runtime status of the job and/or individual tasks, as well as additional application checkpointing, primarily any non-persistent arrays and output buffers, is saved to persistent storage. The rate and complexity of the checkpointing is balanced between the recompute costs and the associated costs of the preemption support. This checkpointing places an enormous burden on the IO layer and associated IO components: a situation that would be untenable in traditional private data centers. It does, however, open many possibilities for data analytics and enhanced data processing of intermediate products.

Application to production project

We show seismic inversion results from a field data example acquired in the Cape Anguille/Orphan Basin region offshore Canada. This is a narrow azimuth, multi-sensor dataset consisting of 16, 8km cables with a 1.6km cross spread. Simultaneous inversion of velocity and pre-stack reflectivity assists in the identification and interpretation of potential hydrocarbon bearing reservoirs within the Tertiary and Cretaceous sediments. Initial analysis of the inversion parameters determined the optimum system configuration for the forward and adjoint modelling tasks to be 32vCpus and 128Gb of memory. The accumulation and update phases used a secondary resource pool consisting of 64vCpu and 512Gb VMs. These node pools were provisioned across a federation of 4 separate GKE clusters. Figures 3a to 3d show the VM usage (running and pending) for each cluster. The blue shaded areas highlight periods where individual clusters are starved for capacity.

During the initial shot and adjoint modelling phase all 4 clusters were fully utilized. Current limits in the VPC address space coupled with quota limits for the cluster core count puts the maximum size of each cluster at around 8000 VMs. Under full utilization around 1 to 1.1million vCpu's were active. The Kubernetes scheduler automatically removes idle instances resulting in the characteristic rapid upscaling and gradual downscaling pattern seen in the utilization graphs. This ensures that costs are only incurred while the VM's are active. The IO generated for each shot modelling task was approximately 3.2TBytes of mostly temporary data. Figures 3e and 3f show the cumulative throughput for all clusters with peaks above 1.4TBytes/sec. Each iteration involved the generation of over 100 Petabytes of temporary data written to cloud storage. Figure 4a shows the inverted angle dependent reflectivity (0 to 58°) along an inline. The maximum frequency of inversion is 40 Hz. Notice the overall flatness of the gathers that are used for quantitative interpretation. Figures 4b and 4c show the inverted high-resolution velocity model and full stack reflectivity at a depth of 3900 m.

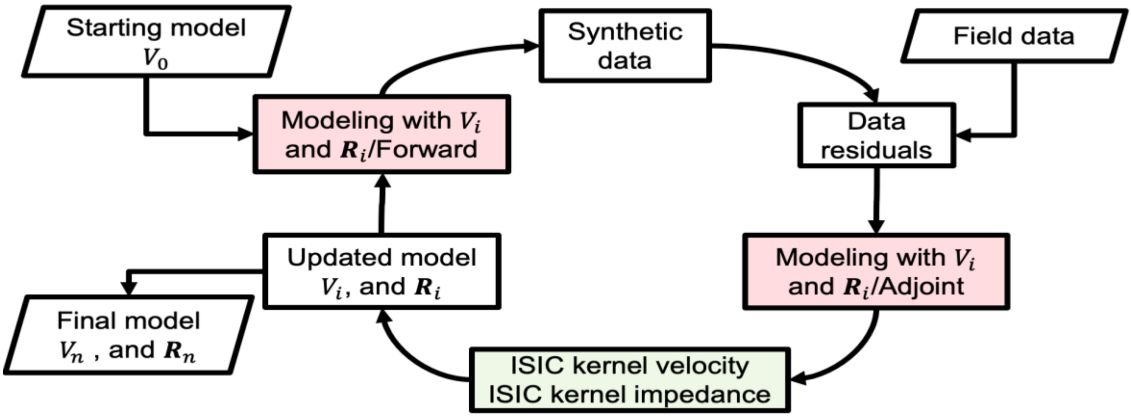

***Figure 1**. Workflow for simultaneous inversion for velocity and pre-stack reflectivity.*

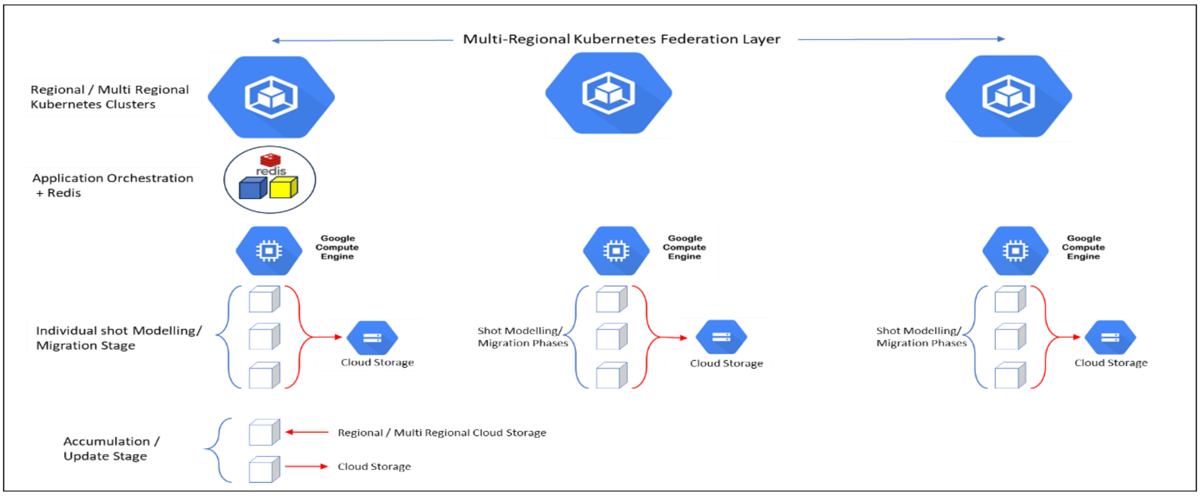

***Figure 2**. Kubernetes cluster configuration, application partitioning, and fault tolerance aspects of the designed inversion application in the cloud.*

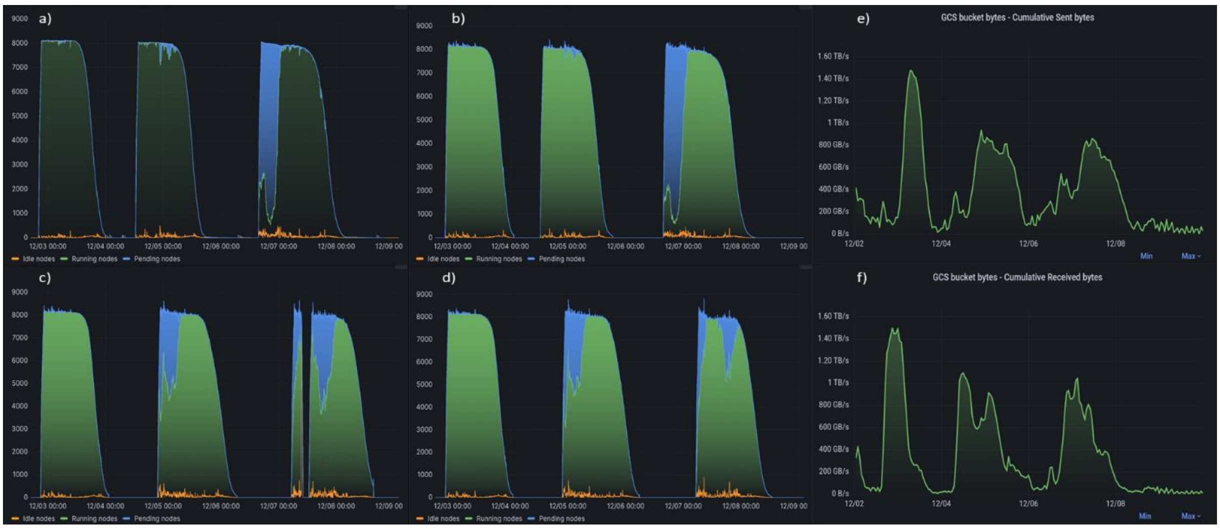

***Figure 3**. VM utilization for each of the 4 federated Kubernetes clusters during 3 iterations of simultaneous inversion, a)US-West-1, b)US-West-2, c)US-West-3, and d)US-West-4. Each VM contains 32vCpus. e) Cumulative write throughput and f) cumulative read throughput illustrating the performance strength of cloud storage*

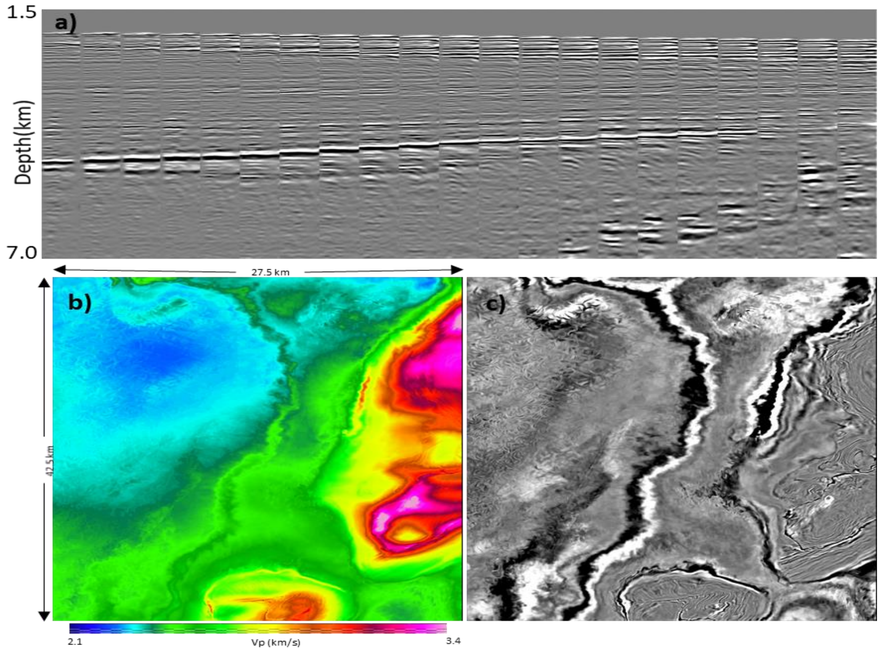

__Figure 4.__ Cape Anguille field data. a) Inverted angle-dependent reflectivity along an inline; b) velocity model and c) full stack reflectivity for a depth slice at 3900. The maximum frequency of inversion is 40 Hz.

Conclusions

We demonstrate a cost-effective, fully fault-tolerant implementation of seismic inversion in the cloud by applying simultaneous inversion of velocity and angle-dependent reflectivity to a production field project. We illustrate the CPU and IO requirements of high frequency pre-stack inversions and demonstrate their fulfillment through a combination of horizontally scaled Kubernetes clusters and massively parallel cloud storage. Future work will address leveraging the large volumes of intermediate products generated on the cloud object storage for enhanced imaging and inversion.

Acknowledgements

We thank PGS for permission to show this work, PGS MultiClient for the field data, and the Google support teams for their ongoing assistance and feedback.

References

Amundsen, J., Arntsen, B., Sollid, A. Buland, A., Sollie, R. [1996]. Parallelization of 2-D and 3-D depth migration. SEG Technical program Expanded Abstracts 1996.

Chemingui, N., Yang, Y., Ramos-Martinez, J., Huang, G., Whitmore, D., Crawley, S., Klochikhina, E. and S. Arasanipalai, [2023]. Simultaneous inversion of velocity and angle-dependent reflectivity, Third International Meeting of Applied Geoscience and Energy, Expanded Abstracts.

Debens, H., Knodel, D., Mancini, F., Harris, D., Warner, M. Umpleby, A. [2022]. Semi-global multi-parameter FWI using public cloud HPC. Second International Meeting for Applied Geoscience & Energy.

Ober, C., Oldfield, R., Wombel, D., Mosher, C. [1997]. Seismic imaging on massively parallel computers. SEG Technical program Expanded Abstracts 1997.

Okita, N., Coimbra, T., Tygel, M., Borin, E. [2019]. A heuristic to optimize the execution cost of distributed seismic processing programs on the cloud. SEG International Exposition and 89th Annual Meeting.

Witte, P, Louboutin, M., Jones, C., Herrmann, F. [2019]. Serverless seismic imaging in the cloud. Rice Oil and Gass Conference 2020.

Yang, Y., Ramos-Martinez, J., Whitmore, D., Guanghui, H. and Chemingui, N. [2021] Simultaneous inversion of velocity and reflectivity. First Break, 39(12), 55-59.