Submission code: MRVPB5Q0RD

See this and other abstracts on our website: https://home.sbgf.org.br/Pages/resumos.php

Energy HPC Orchestrator: A Cloud Native HPC Workflows Management Application

Kun Jiao (AWS), Cyril Lagrange, Dan Kahn, Hussein Shel, Max Liu, Weishan Han, Christos Mavropoulos, Marwan Wirianto, Kent Stevens

Energy HPC Orchestrator: A Cloud Native HPC Workflows Management Application

Abstract

In this abstract, we introduce the Energy HPC Orchestrator (EHO), an innovative framework designed to transform energy HPC applications. We propose an open marketplace for operators, software providers, and academia to deliver seismic processing and HPC algorithms. Through EHO, we cultivate competition that drives innovation. By leveraging cloud technologies, our solution enhances efficiency and flexibility to meet evolving industry demands.

Introduction

In the energy industry's digital transformation, there's a substantial increase in compute demands. This trend is amplified by advanced imaging methods like RTM and FWI, where doubling frequency causes a 16X compute demand increase. Additionally, AI/ML applications, model training and inference in the energy industry are growing exponentially. These developments hunger for flexible, scalable computing resources, with cloud emerging as a viable solution.

However, achieving optimal cloud performance and cost requires significant engineering to modernize existing HPC applications. The EHO is aiming to address these challenges. This open industry platform and marketplace ecosystem, developed by AWS, Shell, Occidental, EPAM Systems, Shearwater, S-Cube, SeisWave, SEIMAX, Devito Codes, NVIDIA, etc., enables interoperability between HPC modules for optimized scalability, flexibility, and economics. It provides pre-optimized cloud native HPC templates to reduce engineering efforts and creates an open HPC marketplace where players can focus on differentiating technology.

The design of EHO

The EHO comprises three components:

1) a system that allows:
    a. the orchestration of different HPC applications that use a common storage.
    b. enterprise functionalities such as user management, project management, and data management etc.

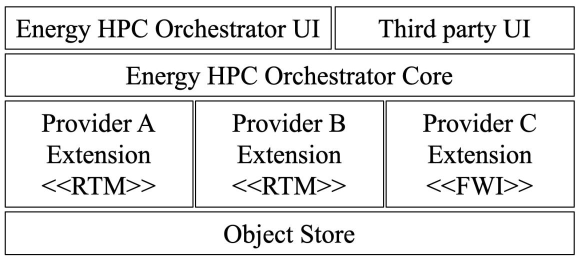

Figure 1: The logical components of the system

2) an ecosystem of applications that are EHO compatible and distributed via a marketplace.
3) a set of standards to allow the interoperability (data/parameter) between applications.

System architecture

Figure 1 depicts the logical components of the system.

The EHO core (core here after) is at the center and acts as a control plane to coordinate the other applications and orchestrate the execution of workflows. The EHO UI (web-ui here after) is a web application that allows the end user to organize their work in projects and groups, and inside those projects users can create workflows by chaining extensions.

The extension utilizes the compute component for its execution and communicates with other extensions through the storage component for data exchange.

Core

Figure 2 presents the core's architectural design, which is built on a serverless framework. At the foundation of this framework are serverless compute for processing, No SQL database for data storage, and an API Gateway, which facilitates the exposure of a REST API to user interfaces. Additionally, an event bus is employed for enabling asynchronous, event-driven communications with extensions.

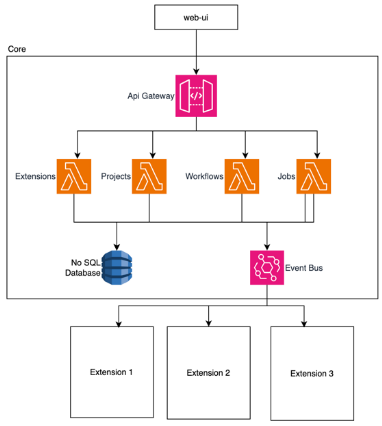

Figure 2: The architecture of the core

Web-UI

As shown on Figure 3 and 4, the web-ui is a React application with a serverless backend and a No SQL database.

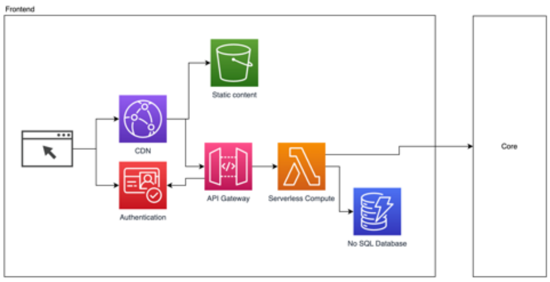

Figure 3: The architecture of the Web-UI

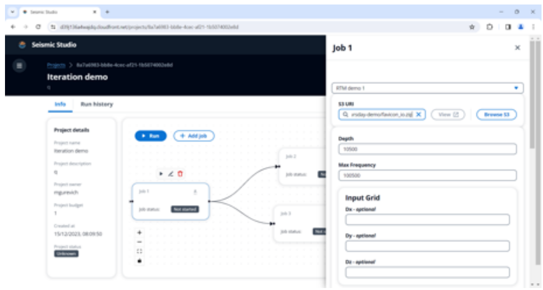

Figure 4: Workflow management Web-UI

Extensions and Template

Extensions are plug-ins to the system that provide the domain specific capabilities. For example, we can have different extensions providing algorithms such as RTM or FWI. Extensions are required to implement the event-based protocols defined by the core. Templates are reusable for a class of algorithms and encapsulate the best practices for running this class of algorithm on AWS. An example of RTM template is described below.

RTM Template

This template is designed to transform a traditional RTM application for seismic processing, into a modernized, cloud-native application. This design capitalizes on robust AWS services to boost the scalability, resilience, and operational efficiency of high-end imaging algorithms.

The execution of the RTM algorithm unfolds within the dynamic backend, segmented into four distinct, decoupled microservices, each one specialized for a task of the RTM:

Analysis Service: This service procures work items from the input queue, scans seismic file headers to determine shot count, and dispatches metadata to the migration queue.

Migration Service: This service pulls a work item details from the migration queue. Thereafter, it solves wave equation, and yields a 3D image that is then uploaded to the object store. After this, a corresponding reduction work item is pushed into the reduction queue.

Reduction Service: This service activates by retrieving two work items from reduction queue, stacks corresponding images, re-uploads them, and re-enqueues work items until a single composite image remains.

Converter Service: This service converts the final stacked image into the appropriate final format.

The indirect interaction between microservices via a queuing system not only enforces the system's resilience and fault tolerance but also enables the autonomous scaling of each service. Through an auto-scaling group, the services can dynamically adjust in scale, driven by the number of work items or 'shots' queued, thereby ensuring optimal resource allocation.

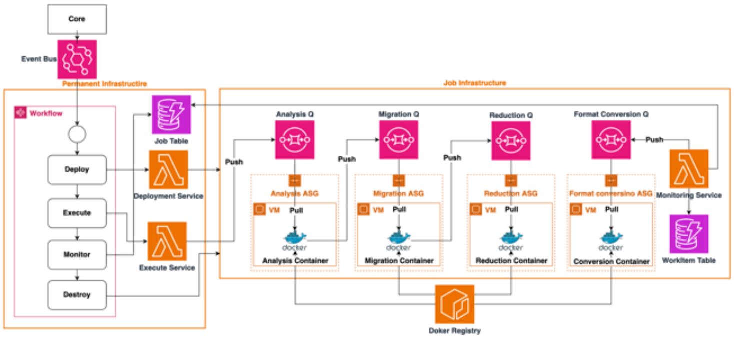

Figure 5: Cloud native RTM template

The asynchronous and resilient nature of this architecture also offers a significant edge in cost optimization through AWS Spot Instances which are unused AWS capacity with significant discounts. However, the instance can be reclaimed with a short notice. This architecture's stateless, decoupled design ensures that if an instance is interrupted, the task can be reinserted into the queue and resumed by another instance, therefore minimizing the disruption. Different Instance types can be used for various services to achieve further optimization. For example, Analysis Services can use general-purpose instances. Migration Services can utilize HPC instances, and the Reduction Service can use network-optimized instances.

Other Template

We can envisage other templates, such as conventional Slurm-based applications, or applications with have their own cloud native designs, as long as it has compatible event bus protocol.

Ecosystem and Distribution Model

Building an open ecosystem that includes HPC applications from a range of providers is crucial for the development of the EHO. We've engaged with several AWS partners and have integrated a variety of applications, UI and SDKs so far:

1) RTM of SeisWave (public extension in marketplace)
2) RTM of SEIMAX (public extension in marketplace)
3) FWI of S-Cube (public extension in marketplace)
4) RTM of Shell (private extension)
5) RTM of Occidental (private extension)
6) RTM of Devito Codes (SDK)
7) RTM of Nvidia Energy SDK (SDK)
8) Reveal of Shearwater (Alternative UI front-end)

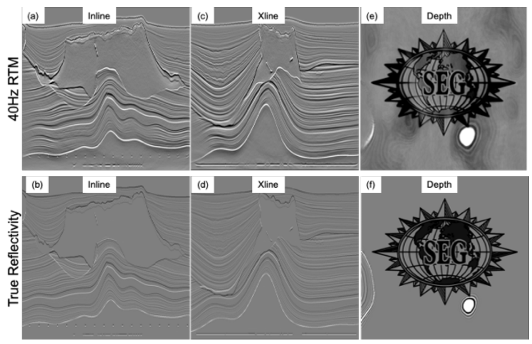

Figure 6: The result of SeisWave RTM

Partners have the option to offer applications through a public marketplace (e.g. AWS Marketplace), which includes various commercial and evaluation options. An open marketplace simplifies the discovery and acquisition of HPC applications.

Standardization and Interoperability

Standardization is important for the EHO, as it allows interoperability among extensions. It contains two sets of standards: 1) a standard for input and output data formats, promoting data exchange, and 2) a standard for parameterization, for instance, definitions for anisotropic models, frequency, aperture, etc... This ensures a consistent and effective framework for communication and operation across various extensions. Through enabling interoperability, our solution allows for creation of innovative workflows by combining best-of-breed of the industry.

Examples

In the first example, we showcase the execution of SeisWave's RTM via the EHO. The outcome, depicted in Figure 6, presents the migration results of the SEAM 3D TTI Subsalt model (Fehler and Keliher, 2012). The TTI RTM (Zhang et al., 2011) configuration included a 40Hz maximum frequency, utilization of 1476 Ocean Bottom Nodes (OBN), a 17.5km migration aperture in both directions, and a migration duration of 16 seconds. This computation was performed on AWS's Graviton3 processors, with spot instances. Impressively, the total computation cost is under $2000.

A real data FWI example is also shown with S-Cube’s advanced multi-cost function FWI framework, known as XWI, giving a superior well log match with AWI-RWI-FWI. The data is from northwest shelf of Australia with narrow azimuth acquisition. The inversion ran in S-Cube’s AWS infrastructure utilizing Graviton3 spot instances to ensure cost-effectiveness and scalability.

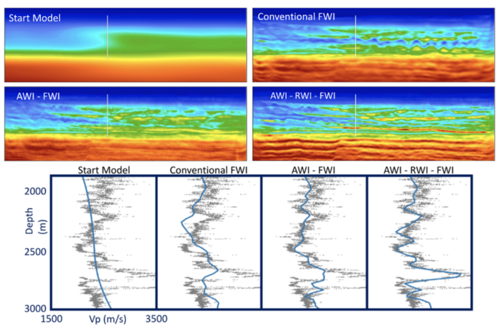

Figure 7: Result of S-Cube FWI

Conclusions

In the era of digital transformation, the energy industry faces significantly increasing computational demands, especially in HPC applications for advanced seismic imaging and AI/ML. The EHO, a collaborative innovation among industry presents as a key initiative to address those challenges. It not only enhances current capabilities but also offers flexibility to meet future requirements. Furthermore, this platform and ecosystems can also be extended beyond seismic workflows. This effort exemplifies the crucial role of technological advancement and collaboration among industry in advancing the computational frontiers for the energy industry.

Acknowledgments

We thank AWS, Shell, Occidental, EPAM Systems, Shearwater, S-Cube, SeisWave, SEIMAX, Devito Codes, NVIDIA, Wintershall Dea, and others for their contributions and permission to publish this abstract. Special thanks to Mik Isernia, Tim Roden, Francesco Menapace, Klaas Koster and Chao Wang for their enthusiasm, pivotal role in initiating this project, and support in guiding us this far.

References

Fehler, M., and P. J. Keliher, 2012, SEAM phase I: Challenges of subsalt imaging in Tertiary basins, with emphasis on deepwater Gulf of Mexico: SEG, http://dx.doi.org/10.1190/1.9781560802945.

Zhang, Y., Zhang, H. and Zhang, G., 2011: A stable TTI reverse time migration and its implementation, Geophysics, 76:3, WA3-WA11.