SCALING AND OPTIMIZING PERFORMANCE AND COST OF MACHINE LEARNING INGESTION ON UNSTRUCTURED DATA FOR SUBSURFACE APPLICATIONS

L.C.I. Panganiban[1], F. Baillard[1], N.M. Hernandez[1]

[1] Iraya Energies

Summary

In recent years, the energy industry has shifted their attention into extracting additional values from their in-house legacy datasets for shorter project turnaround and better decision making. Internal digital transformation initiatives and access to new technology such as cloud computing, machine learning and microservices made it possible to shift towards a scalable ingestion platform.
A modern scalable ingestion platform often includes 1) automated machine learning (ML) components articulated around pipelines to parse and go through the data and extract the needed information 2) storage component such Database, Datawarehouse and Datalake defining what data to be stored and how.
In this paper, we will provide a description of these different components, their modern implementation into a cloud environment using microservices and some performance benchmark based on real world data examples.

Scaling and optimizing performance and cost of machine learning ingestion on unstructured data for subsurface applications

Introduction

In recent years, the energy industry has shifted their attention into extracting additional values from their in-house legacy datasets for shorter project turnaround and better decision making. Internal digital transformation initiatives and access to new technology such as cloud computing, machine learning and microservices made it possible to shift towards a scalable ingestion platform.
A modern scalable ingestion platform often includes 1) automated machine learning (ML) components articulated around pipelines to parse and go through the data and extract the needed information 2) storage component such Database, Datawarehouse and Datalake defining what data to be stored and how.
In this paper, we will provide a description of these different components, their modern implementation into a cloud environment using microservices and some performance benchmark based on real world data examples.

Scalable ingestion platform architecture

The vast majority of experimentation and testing of new ML application are performed on notebooks on local machines leveraging on local hardware with the data and the code being stored locally-. Such a setup has the advantage of being lightweight allowing a fast turnaround between two ML iterations. However the lack of traceability, compatibility and hardware limitation makes it challenging to scale such application, hence the requirement of a more scalable solution.
In comparison, a scalable ingestion platform is a type of system native able to handle current and future workloads regardless of the amount of data ingested or users connecting to it, considering both scaling in (shrinking resources) and scaling out (expanding resources). An example of such system can be seen on Figure 1. The architecture is made of of four components namely pipeline, storage, compute and interfaces.

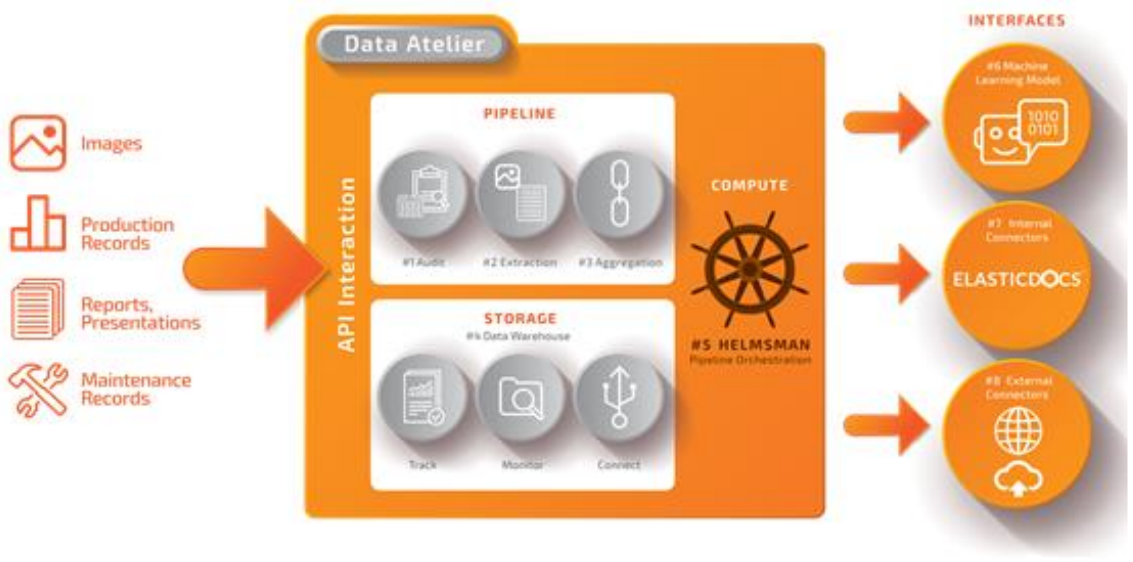

Figure 1: Ingestion Platform Architecture

Pipeline or data pipeline is the main driver in extracting and transforming data into a unified format. The typical ingestion workflow is based on an Extract-Transform-Load process with machine learning workflows able to accommodate the variety in sources and forms present in unstructured data (Hernandez et al., 2019).

The storage component is where the data resides. Storage is often the least thought about in development and architecture strategies, but it is one of the core contributors in terms of cost to the organization. The compute node component schedules and orchestrates the data pipelines, logic flows, and algorithms. The interface or the dashboard component provides the monitoring and observability capabilities. This presents the events, statuses, logs, and other operational insights that can be used for decision making. These components have different ways to be scaled depending of 1) the environment of installation: on-premise vs. cloud 2) the volume and type of data being processed 3) the early development decision: all out of the box cloud providers solution, opensource self maintained solution or hybrid.

Component's optimization

The first metrics to consider for optimization is processing time and should be independent from the volume of data to be processed by the pipeline. This optimization is achieved through parallelization, distributed computing and orchestration by scaling up or down the usage based on demand. Applications and data pipelines are packaged into containers and then deployed into a Kubernetes cluster in which it handles the scheduling, distribution, and allocation across different machines. Workflow managers (WM) provide the platform to execute and orchestrate the jobs and tasks that are contained in a Pod/Pool (Figure 2). Workflow managers also implement orchestration and scheduling though it handles the tasks in the application layer where this determines if the data execution is successful or not.

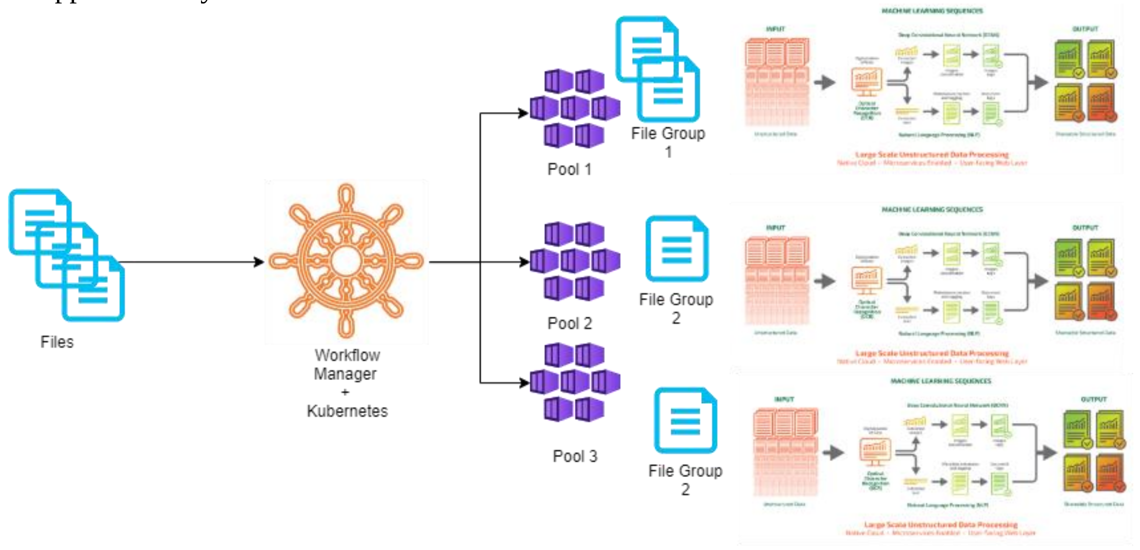
Files
Workflow Manager + Kubernetes
Pool 1
Pool 2
Pool 3
File Group 1
File Group 2
File Group 2
MACHINE LEARNING SEQUENCES
INPUT
OUTPUT
Large Scale Unstructured Data Processing

Figure 2: Unstructured data pipeline with orchestration in place

The second metric to consider is the extensibility metric. A single service or component can be used beyond what is originally intended. Thanks to the microservices, exposing your data via application programming interfaces or APIs is a good practise (Figure 3). It provides a common language across teams where implementation is unified across different data sources. An additional advantage of the use of APIs is the additional level of abstraction for the storage of the data. Data are now accessible from various mounting locations through a single call, allowing democratization and versioning of the data.

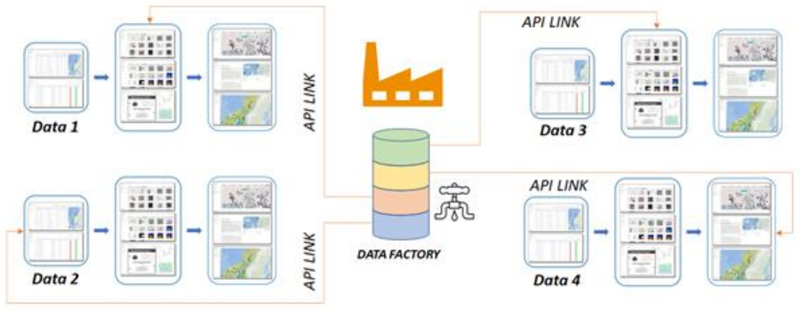

Figure 3: Connecting applications using API links

As an example, a full-text search endpoint originally used to do searches, can also be used to create aggregations models like heatmap and knowledge graphs (Baillard et Al., 2021) as seen on Figure 4.

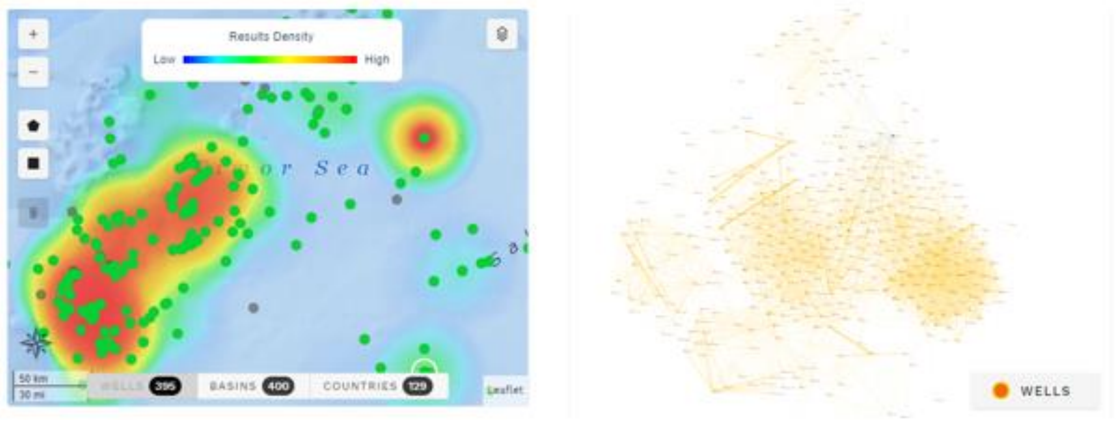

Figure 4: Heatmap (left) and Knowledge Graph (right)

The third metric to consider is the UI/UX responsivness and define the optimal data representation for the user or the operator to QC the data and for the service to have still an acceptable response time. As seen on Figure 4, the heat map and the knowledge graph are built on-top of 100,000 data points (Mamador et al., 2021). Interface and visualization scaling requires removing the dependency on the volume of data points. The amount of time visualizing 100 points should be the same as visualizing 100,000 points – in which development of aggregation and interpolation workflows must be performed in the backend and frontend to overcome this volume challenge.

Benchmarks and testing

To see the performance of this architecture we can consider the following data. We have 3 buckets of data for this assessment. It is composed of geoscience documents that ranges from final well reports, geological report, regional studies, interpretations coming from various file formats, countries, and languages. The data also includes images like thin sections, core, well logs, seismic. All these data are audited and stored into an object storage.

The following are the system that was used for this assessment:

- <u>Base case:</u> 1-node with 4 cores and 16 GB of RAM from cloud provider without WM
- <u>Optimized case</u>: 5-nodes with 4 cores and 16 GB of RAM per node from cloud provider with WM

Bucket	Number of Pages	Duration (hours) Single node	Failure Rate % (Single Node)	Duration (hours) Cluster	Failure Rate % (Cluster)	Language	Region
1	4,403	18	15	4	5	Mixed	Europe
2	40,565	170.25	11	38.5	0.5	English	Oceania
3	62,531	250	12	60	0.01	Mixed	South America

Table 1: Extraction Performance with 4-cores and 16 GB of RAM per node

Table 1 shows the extraction performance. The extraction workload encompasses the data loading, processing, and uploading. It was tested on a variety of datasets from different regions and languages. For a single node execution, the average number of pages per hour is around 244 pages and for a 5-node cluster, the average number of pages per hour is 1000. Since workflow manager provide an auto-retry function, this reduces the failure rate by a significant amount. This is due to tasks that are failing due to network or infrastructure issues to be retried or re-executed. The failure rate is now mostly on the data since we haveve removed the infrastructure constraint.

As one of the metrics that we have set earlier, we need to consider the cost and infrastructure usage. Using the same workloads, we can assess the cost and infrastructure optimization in Table 2.

Bucket	**Number of Pages**	**% Hardware usage Avg. Single node**	**% Hardware usage - Avg. Cluster**
1	4,403	42%	84%
2	40,565	40%	83%
3	62,531	41%	83%

Table 2: Extraction Infrastructure Utilization with 4-cores and 16GB of RAM

Table 2 shows that we have a higher utilization in the cluster hence we are fully utilizing the server. In the case of single node, only 41% of the resources is utilized vs 83% in the cluster approach. Both the reduced failure rates and the hardware utilization illustratres the gain of efficency in deploying a scalable ingestion platform

Conclusions

In this paper, we have seen how to scale and optimize the ML ingestion pipeline for subsurface applications. We have shown that scaling and optimizing ML ingestion pipelines can lead to improvements in terms of time and cost. We have also demonstrated that the ingestion platform is scalable to handle data from a variety of regions and languages. Finally, we've seen the power of scaling visualizations and exposing the data via API links in which we can get more insight and enhance the ability of the team to extract knowledge.

References

Baillard F. and Hernandez N.: A Case Study of Understanding Bonaparte Basin using Unstructured Data Analysis with Machine Learning Techniques. 82nd EAGE Conference & Exhibition, 18-21 October 2021, Amsterdam.

Mamador C., Hernandez N., Baillard F.: Production-scale processing of EAGE's EarthDoc data to stimulate new insights in CO2 and new energy management. 82nd EAGE Conference & Exhibition worskhop on ML solutions at scale, 22 October 2021, Amsterdam.

Hernandez N., Lucañas P., Graciosa J.C., Mamador C., and Panganiban L. C. I., 2019: Automated information retrieval from unstructured documents utilizing a sequence of smart machine learning methods within a hybrid cloud container. EAGE Workshop on Big Data and Machine Learning for E&P Efficiency 25 - 27 February.