ML pipeline performance comparison

Ben Stuart
School of Enterprise Computing and Digital Transformation, TU Dublin, Ireland
X00193209@myTUDublin.ie

Introduction

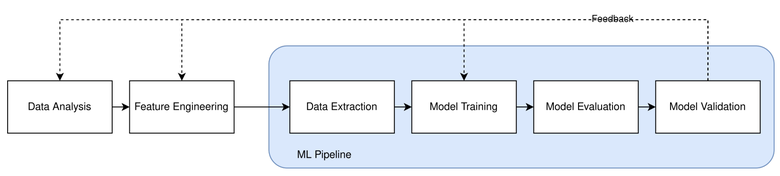
This project aimed to analyse and compare machine learning pipeline architecture performance to identify performance differences between architectures. Interest in operating machine learning has become a growing topic in the data science community, which has brought an increased focus on MLOps. Little research has been done on MLOps to date, with most works focusing on foundational information collection through literature reviews, interview studies and proof of concept architectures. Progress has been achieved in establishing a high-level state-of-the-art, but much more research is required to identify future work and deepen collective knowledge. Machine learning pipelines are often used for continuous training and machine-learning tasks in an MLOps context to automate and orchestrate model training, delivery, and other tasks. This work compares Metaflow, Apache Airflow and SageMaker pipeline frameworks deployed to AWS-based infrastructure regarding resource requirements for different training and inference workloads and runtime environments (Kubernetes cluster, SageMaker jobs and AWS Batch).

Research Question

This work aimed to discover how distributed machine learning pipeline architectures compare resource utilisation and time taken to orchestrate tasks with equivalent workloads and resource allocation.

What is an ML Pipeline?

Continuous Training pipelines are often called Machine Learning pipelines (MLP), sometimes referred to as *ML workflow pipelines*. These pipelines are written as discrete interdependent steps, forming a directed acyclical graph (DAG). Writing workflows in this way allows these pipelines to be orchestrated on distributed systems, re-run steps independently, and utilise available compute resources efficiently. Notable works and interview studies have shown writing ML workloads as MLPs avoids known pitfalls in notebooks such as scalability and low code quality.



Experiment design

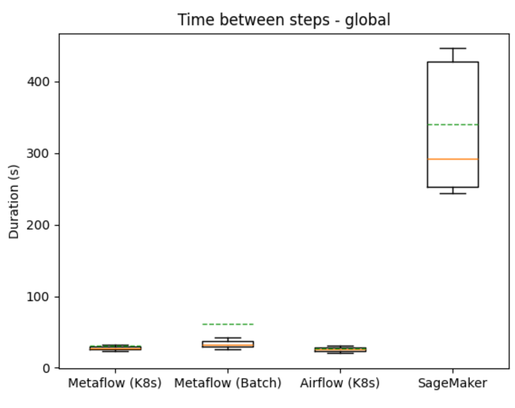
1. Workloads:

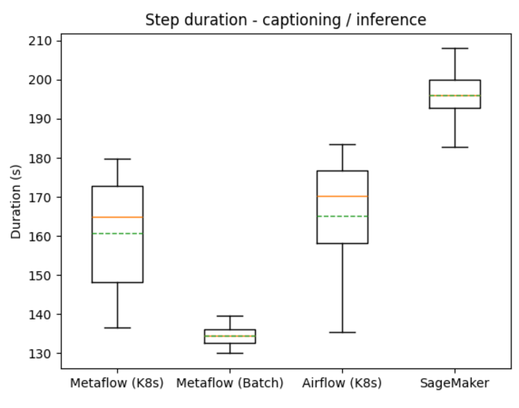
Two separate workloads have been created to test the four tools. The first is a batch inference workload to caption a set of images, and the second is a model training workload which trains a neural network. These workloads have been chosen as they represent everyday use cases for MLP tools and two critical stages of the model life-cycle, training and inference.

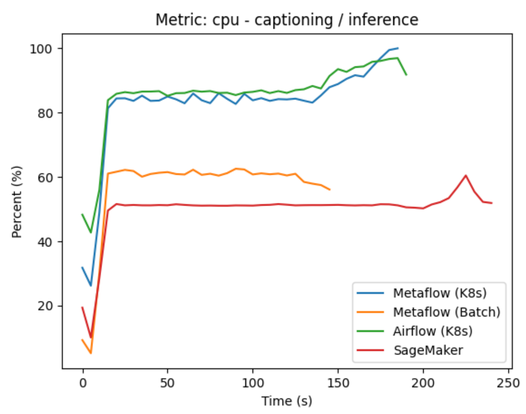
2. Architecture configuration:

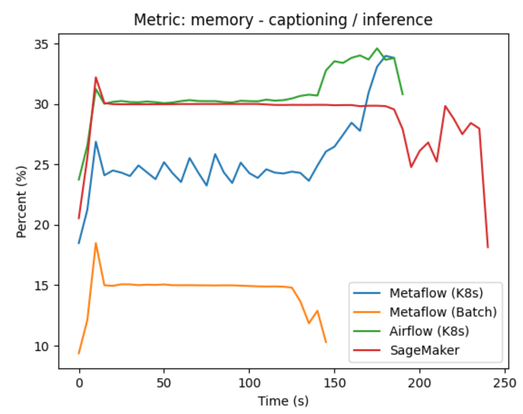
This study focused on four architecture configurations: the Metaflow pipeline framework, backed by Kubernetes cluster for compute resources; Metaflow us- ing the AWS managed service Batch for compute resources; Apache Airflow, also supported by a Kubernetes cluster and AWS's managed service SageMaker. These configurations have been selected to represent a mixture of open-source and self-hosted solutions and proprietary services and hybrids of the two.
Metaflow is an open-source MLP framework initially developed at Netflix; container-based service that can be backed by various orchestration and computing resources such as Kubernetes, AWS Batch, and Airflow. In this study, both Kubernetes and AWS Batch based architectures were used.
Apache Airflow is an open-source workflow orchestration platform. Like Metaflow, Airflow has a Python library for describing workloads, known as DAGs (directed acyclic graph), and a container-based service that can schedule tasks on multiple backend compute resources, including Kubernetes used in this study.
SageMaker is a fully managed service from AWS that offers many sub-services, including scheduling various job types using SageMaker pipelines. SageMaker pipelines comes with an SDK for describing and executing pipelines. In this study, SageMaker Processing or Training jobs were used throughout to execute tasks.
All data was collected using OpenTelemetry exporters and psutil functions.

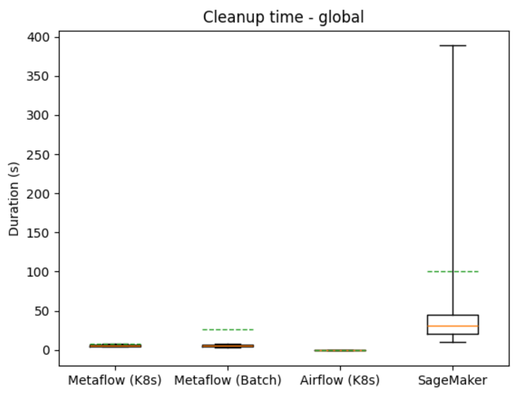
Results

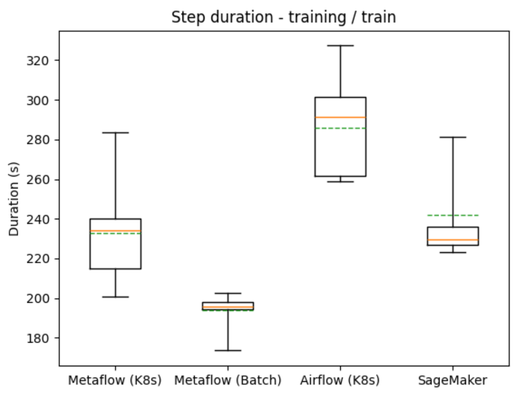


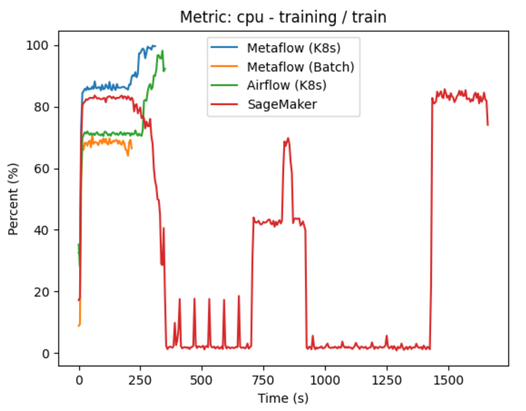


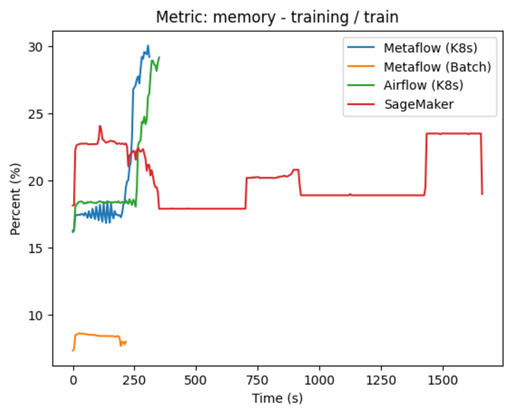












Conclusions and Future Work

The results of this work have shown that behaviours across architectures can differ vastly, and some may be beneficial to different use cases. It has also been shown that while framework choice can contribute to performance; it is most impacted by good, well-architected compute and orchestration resources. Future work is required to understand deeply how data scientists use machine learning pipeline frameworks, what features are most desirable, and the pain points with these tools.

QR Code for Recording