EvaDB in Amazon SageMaker

GitHub: https://github.gatech.edu/ssathish6/EvaDB-SageMaker-Container

DockerHub: https://hub.docker.com/r/ssathish6/evadb-sagemaker-container (CPU-only)
https://hub.docker.com/r/ssathish6/evadb-sagemaker-container-cuda (CUDA GPU)

Introduction

Amazon SageMaker is a cloud machine learning platform which provides developers with tools to create, train, and deploy ML models. EvaDB aligns with SageMaker's goals but with a very different execution, abstracting many of the details of creating ML models to make model creating simpler. With this point, EvaDB could be used by SageMaker users in assisting the execution of steps in the ML model pipeline.

Project 1 Overview

To be able to use EvaDB as a tool with SageMaker, first, EvaDB must be exist within a containerized environment and inserted into a SageMaker instance hosted on Amazon Elastic Container Service (ECS). After this, to effectively use EvaDB in SageMaker's tools, a train and serve execution mode will have to be implemented. As there are two assignments for the project, and there is a logical split for the implementation, this first part of the project implements Docker containers for different hardware and deployment scripts for ECS.

Implementation Details

Consider the project directory listed below.

```
./
├── Dockerfile              – Dockerfile for usage without a CUDA GPU
├── Dockerfile.CUDA         – Dockerfile for usage with a CUDA GPU
├── build_and_push.sh       – Shell script to push to ECS and build
├── serve_local.sh          – Shell script for testing serve execution
└── evadb_instance/         – EvaDB instance to be used
    ├── algorithm.py        – EvaDB application for usage built with Flask
    ├── nginx.py            – Application configuration
    ├── serve               – Serve execution mode
    └── wsgi.py             – Wrapper for Gunicorn
```

The directory tree above provides a high-level overview of the project so far. At the moment, a Dockerfile exists to be used to create and image on ECS to later be incorporated in execution modes of SageMaker. The normal Dockerfile is to only be used when a CUDA-supported GPU isn't available.

The EvaDB instance itself is built as a Flask application using Gunicorn and NGINX for server functionality. Users will be able to edit how the application works thorough the algorithm.py file regarding training and serving. In the future with project 2, serving and training will be fully support, hopefully, along with custom model creation. Direct instruction to use the container are hosted on the GitHub link to the repository.

Metrics

Consider some of the following interesting points:

- The Docker image to be hosted on ECS is currently about 2.23 GB before implementation of training and serving handlers
- On the other hand, the Docker image to be primarily used for local testing and development is currently about 314 MB in the same state with the handlers
- Hosting options vary in pricing largely
    - For the more common data science workflow, training will about $0.96 per hour with an ml.m4.4xlarge for 30 minutes per training run with Amazon SageMaker Debugger enabled using 2 built-in rules and 1 custom rule
    - For larger, more customized workloads, the price still stays many times under a dollar per hour, but as the time requirements increase so does the total cost
    - The increase in cost the use of EvaDB within the container is yet to be experimented on; following project 2 would be a reasonable time to experiment

Challenges & Lessons

The MindsDB SageMaker container repository used to create this EvaDB SageMaker container is executed quite well with some good documentation, but the difference in usage for EvaDB compared to MindsDB is notable – it's not a direct translation.

I have not really done much with Python outside of ML and DL courses and algorithms problems, so building a Flask application with different server libraries was new and challenging. I'll have to spend a lot more time these utilities to do the rest of the project in part 2.

Outputs

The container uses EvaDB with all the libraries required. The environment is fitted that any EvaDB application should work as expected. I've tested the test/example programs on the docs, and the output is the same. For example, on the speech to test, the output on the Ukraine video was:

```
+-------------------------------------------------------------------------------------------------------------------------+
|                                                    text_summary.text                                                    |
+-------------------------------------------------------------------------------------------------------------------------+
| The war in Ukraine has been on for 415 days. Who is winning it? Not Russia. Certainly not Ukraine. It is the US oil ... |
+-------------------------------------------------------------------------------------------------------------------------+
```

This is what's expected.

References

The only sources I used for this project was the MindsDB implementation provided, Docker docs, and Amazon SageMaker docs.

- https://github.com/mindsdb/mindsdb-sagemaker-container
- https://docs.docker.com/
- https://docs.aws.amazon.com/sagemaker/
- https://sagemaker.readthedocs.io/en/stable/