

Endterm Report: Distributed LSTM Training & System Monitoring

by Anel Madina Zhankozy Zhalgas

Introduction

This report outlines the step-by-step process of integrating Prometheus and Grafana to monitor an Apache Spark distributed data processing system. The setup collects and visualizes key metrics, including CPU load, memory usage, task execution time, and processed data volume. The report includes detailed explanations of each step, with placeholders for screenshots.

- Preprocessing the ETTh1 dataset (Electricity Transformer Data)
- Distributing LSTM training across multiple Spark workers
- Monitoring system performance using Prometheus & Grafana

Setting Up Prometheus

Install Prometheus

Download Prometheus from prometheus.io.

Open `C:\Users\zhanik\Downloads\prometheus\prometheus.yml` and add the following:

```
scrape_configs:  
  - job_name: 'spark'  
    static_configs:  
      - targets: ['192.168.224.224:8078', '192.168.224.224:8079']
```

start prometheus

```
PS C:\Users\zhanik\Downloads\prometheus-3.2.0-rc.1.windows-amd64> ./prometheus --config.file=prometheus.yml
```

Configuring Spark for Prometheus Metrics

modify the `metrics.properties` file:

```

# Prometheus Sink for Spark Metrics
*.sink.prometheus.class=org.apache.spark.metrics.sink.PrometheusServlet
*.sink.prometheus.path=/metrics/prometheus
*.sink.prometheus.period=10

#JMX Monitoring for Spark Metrics
*.sink.jmx.class=org.apache.spark.metrics.sink.JmxSink
executor.source.jvm.class=org.apache.spark.metrics.source.JvmSource
executor.sink.jmx.class=org.apache.spark.metrics.sink.JmxSink
executor.sink.prometheusServlet.class=org.apache.spark.metrics.sink.PrometheusServlet
executor.sink.prometheusServlet.path=/metrics/executor
executor.sink.prometheusServlet.port=8082










# JVM Metrics
master.source.jvm.class=org.apache.spark.metrics.source.JvmSource
worker.source.jvm.class=org.apache.spark.metrics.source.JvmSource
driver.source.jvm.class=org.apache.spark.metrics.source.JvmSource
executor.source.jvm.class=org.apache.spark.metrics.source.JvmSource

# Task & Execution Metrics
executor.source.task.class=org.apache.spark.metrics.source.TaskMetricsSource
driver.source.task.class=org.apache.spark.metrics.source.TaskMetricsSource
application.source.task.class=org.apache.spark.metrics.source.TaskMetricsSource

# Input/Output & Block Manager Metrics
executor.source.blockManager.class=org.apache.spark.metrics.source.BlockManagerSource
driver.source.blockManager.class=org.apache.spark.metrics.source.BlockManagerSource
application.source.blockManager.class=org.apache.spark.metrics.source.BlockManagerSource

```

prometheus

spark		2 / 2 up 	
Endpoint	Labels	Last scrape	State
http://192.168.224.224:8078/metrics	instance="192.168.224.224:8078" job="spark" 	 3.043s ago  71ms	
http://192.168.224.224:8079/metrics	instance="192.168.224.224:8079" job="spark" 	 47.811s ago  138ms	

Enable JMX Exporter for Spark

Download `jmx_prometheus_javaagent-1.1.0.jar` from [GitHub](#).

Create a `jmx_config.yaml` file:

```
rules:
  - pattern: "org.apache.spark<name=.*><>.*"
  - pattern: ".*"
```

Start Spark with Prometheus and JMX Exporter

run the command

```
PS C:\Users\zhanik> spark-submit --master spark://192.168.224.224:7077 `
>> --conf "spark.executor.instances=4" `
>> --conf "spark.executor.cores=1" `
>> --conf "spark.executor.memory=500m" `
>> --conf "spark.scheduler.mode=FAIR" `
>> --conf "spark.eventLog.enabled=true" `
>> --conf "spark.eventLog.dir=file:///C:/Users/zhanik/Downloads/spark-3.5.4-bin-hadoop3/spark-3.5.4-bin-hadoop3/logs" `
>> --conf "spark.history.fs.logDirectory=file:///C:/Users/zhanik/Downloads/spark-3.5.4-bin-hadoop3/spark-3.5.4-bin-hadoop3/logs" `
>> --conf "spark.ui.retainedJobs=50" `
>> --conf "spark.ui.retainedStages=50" `
>> --conf "spark.ui.retainedTasks=50" `
>> --conf "spark.sql.shuffle.partitions=2" `
>> --conf "spark.default.parallelism=4" `
>> --conf "spark.metrics.conf=conf/metrics.properties" `
>> --conf "spark.executor.processTreeMetrics.enabled=true" `
>> --conf "spark.driver.extraJavaOptions=-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=7078 -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false -javaagent:C:/Users/zhanik/Downloads/spark-3.5.4-bin-hadoop3/spark-3.5.4-bin-hadoop3/jmx_prometheus_javaagent-1.1.0.jar=8078:C:/Users/zhanik/Downloads/spark-3.5.4-bin-hadoop3/spark-3.5.4-bin-hadoop3/jmx_config.yaml" `
>> --conf "spark.executor.extraJavaOptions=-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=7079 -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false -javaagent:C:/Users/zhanik/Downloads/spark-3.5.4-bin-hadoop3/spark-3.5.4-bin-hadoop3/jmx_prometheus_javaagent-1.1.0.jar=8079:C:/Users/zhanik/Downloads/spark-3.5.4-bin-hadoop3/spark-3.5.4-bin-hadoop3/jmx_config.yaml" `
>> file:///C:/Users/zhanik/code1.py
```

Setting Up Grafana

connect with prometheus

Data sources

View and manage your connected data source connections

🔍 Search by name or type

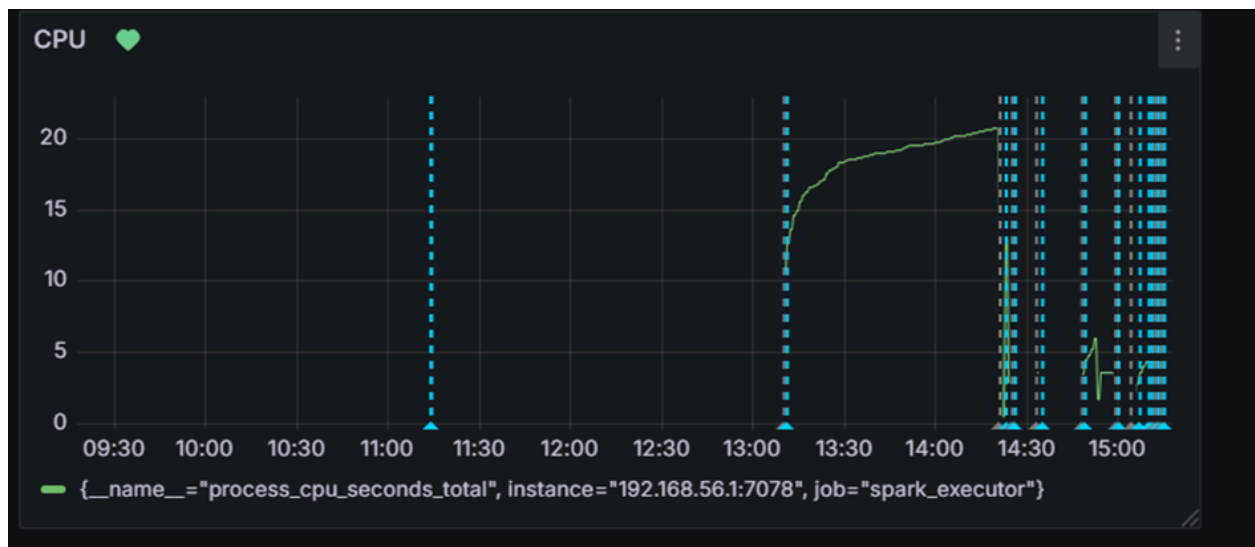


prometheus

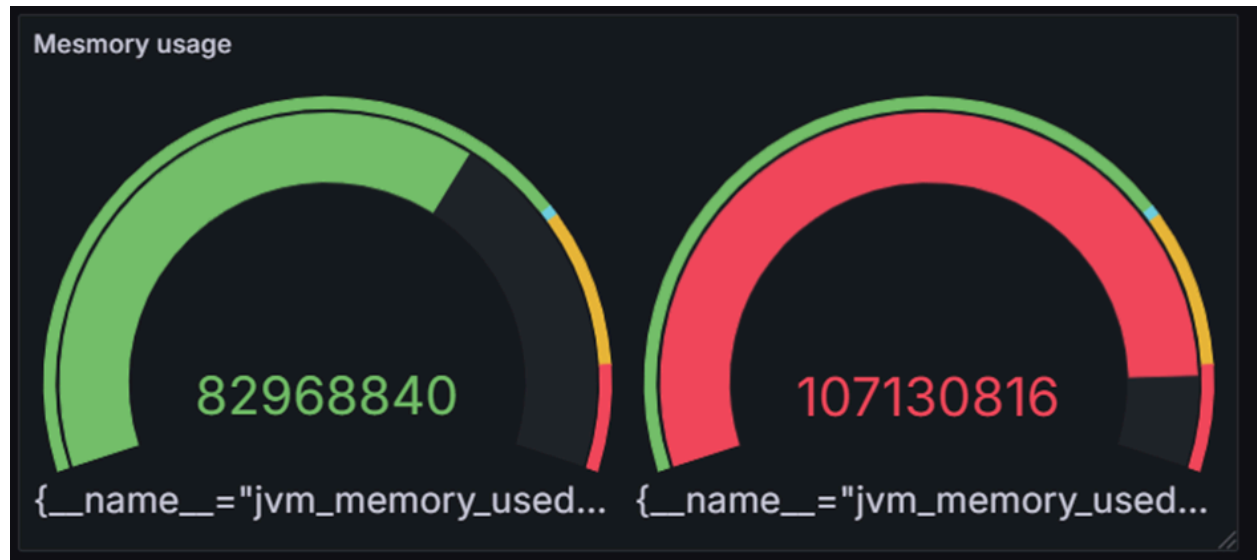
Prometheus | <http://localhost:9090> | **default**

create panels such as

CPU Load Panel



Memory Usage Panel



Code

This code loads time-series data, preprocesses it, and trains a distributed LSTM neural network using TensorFlow and Apache Spark to predict future values, then saves the trained model.

```

# -*- coding: utf-8 -*-
"""code2.ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1GgWScTLtJ3BA_NTcnIuZMnjoRbwkmQGh
"""

import os
import pandas as pd
import numpy as np
import tensorflow as tf
from pyspark.sql import SparkSession
from sklearn.preprocessing import MinMaxScaler

os.environ["JAVA_HOME"] = "C:\Program Files\Java\jdk-17"
os.environ["SPARK_HOME"] = "C:\\Users\\zhanik\\Downloads\\spark-3.5.4-bin-hadoop3\\spark-3.5.4-bin-hadoop3"
spark = SparkSession.builder \
    .appName("Distributed LSTM Training v2") \
    .master("spark://192.168.224.224:7077") \
    .config("spark.executor.cores", "1") \
    .config("spark.executor.memory", "500m") \
    .config("spark.metrics.conf", "metrics.properties") \
    .getOrCreate()

df = spark.read.csv(r"C:\Users\zhanik\ETTh1.csv", header=True, inferSchema=True)

df_pandas = df.select("date", "OT").toPandas()
scaler = MinMaxScaler()
df_pandas["OT"] = scaler.fit_transform(df_pandas["OT"].values.reshape(-1, 1))

def create_sequences(data, seq_length):
    sequences, labels = [], []
    for i in range(len(data) - seq_length):
        sequences.append(data[i:i+seq_length])
        labels.append(data[i+seq_length])
    return np.array(sequences), np.array(labels)

seq_length = 24 # Predict next hour using 24 past hours
X, y = create_sequences(df_pandas["OT"].values, seq_length)

df_train = spark.createDataFrame([(x.tolist(), float(y)) for x, y in zip(X, y)], ["sequence", "label"])

num_partitions = 4
df_train = df_train.repartition(num_partitions)

strategy = tf.distribute.MultiWorkerMirroredStrategy()

with strategy.scope():
    model = tf.keras.Sequential([
        tf.keras.layers.LSTM(64, return_sequences=True, input_shape=(seq_length, 1)),
        tf.keras.layers.LSTM(64),
        tf.keras.layers.Dense(32, activation='relu'),
        tf.keras.layers.Dropout(0.5),
        tf.keras.layers.Dense(1) # Regression output
    ])

    model.compile(optimizer='adam', loss='mse', metrics=['mae'])

def to_numpy_array(df, column):
    return np.array(df.select(column).rdd.flatMap(lambda x: x).collect())

train_data = np.array(to_numpy_array(df_train, "sequence"))
train_labels = np.array(to_numpy_array(df_train, "label"))

train_data = train_data.reshape(train_data.shape[0], train_data.shape[1], 1)


model.fit(train_data, train_labels, epochs=10, batch_size=64)

model.save("/content/distributed_etth1_model.keras")

print("✅ Distributed LSTM training complete. Model saved.")

```

spark master

 Spark Master at spark://192.168.224.224:7077

URL: spark://192.168.224.224:7077
Alive Workers: 3
Cores in use: 3 Total, 3 Used
Memory in use: 1536.0 MIB Total, 1500.0 MIB Used
Resources in use:
Applications: 1 Running, 1 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (5)

Worker Id	Address	State	Cores	Memory	Resources
worker-20250214154809-192.168.224.224-52007	192.168.224.224-52007	DEAD	1 (0 Used)	512.0 MIB (0.0 B Used)	
worker-20250214154835-127.0.0.1-49341	127.0.0.1-49341	DEAD	1 (0 Used)	512.0 MIB (0.0 B Used)	
worker-20250214154851-192.168.224.192-49765	192.168.224.192-49765	ALIVE	1 (1 Used)	512.0 MIB (500.0 MIB Used)	
worker-20250214154918-192.168.224.224-52142	192.168.224.224-52142	ALIVE	1 (1 Used)	512.0 MIB (500.0 MIB Used)	
worker-20250214155026-192.168.56.1-53140	192.168.56.1-53140	ALIVE	1 (1 Used)	512.0 MIB (500.0 MIB Used)	

Running Applications (1)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20250214154930-0001	(kill) Distributed IMDB Training	3	500.0 MIB		2025/02/14 15:49:30	zhanik	RUNNING	56 s

Completed Applications (1)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20250214154812-0000	Distributed IMDB Training	1	500.0 MIB		2025/02/14 15:48:12	zhanik	FINISHED	29 s

spark application dashboard

Spark Jobs (?)

User: zhanik
Total Uptime: 1.6 min
Scheduling Mode: FAIR
Completed Jobs: 4

Event Timeline

Completed Jobs (4)

Page: 1

1 Pages, Jump to 1, Show 100 items in a page, Go

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
3	collect at C:\Users\zhanik\code.py:50 collect at C:\Users\zhanik\code.py:50	2025/02/14 15:49:55	1 s	1/1 (1 skipped)	1/1 (2 skipped)
2	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2025/02/14 15:49:51	4 s	1/1 (1 skipped)	1/1 (2 skipped)
1	showString at NativeMethodAccessorImpl.java:0 showString at NativeMethodAccessorImpl.java:0	2025/02/14 15:49:45	4 s	1/1 (1 skipped)	4/4 (2 skipped)
0	runJob at PythonRDD.scala:181 runJob at PythonRDD.scala:181	2025/02/14 15:49:38	4 s	2/2	3/3

Page: 1

1 Pages, Jump to 1, Show 100 items in a page, Go

