Tune Your Microservices by Learning from Traces

Zhang Wentao, Yang Yang
Agenda

- About us
- Background story
- Brief about Distributed Tracing & Kubeflow
- Architecture Overview
- Training & Modeling
- Tune microservices based on result
About Us

张文涛
zwitzhang@cn.ibm.com

Zhang WenTao is advisory software engineer in IBM. He is experienced in system/Cloud monitoring, DevOps, big data and kubernetes. He is interested in container orchestration in clusters, Service Mesh and AI.

杨洋
bijyangyy@cn.ibm.com

Yang Yang is advisory software engineer in IBM. She's been working on monitoring for cloud platform over 5 years, and has a lot experience on large scale and dynamic environments. Besides cloud related, she is also very interested in front-end technologies.
Background Story…

• How to track down problems in cloud world easily?

• Traces are very helpful, but **one** request result in **tens of traces** — how to work with them efficiently?

• Is the **pre-set static threshold** can really identify abnormal in a **constantly changing** cloud environment?

• **What we’re trying to do:**
  • Leverage ML to help us understand the huge amount of traces
  • Use the model to help us tune and refine services:
    • Anomaly detection
    • Scaling guidance
Distributed tracing is a **super powerful tool** to help with trouble shooting and performance analysis in real world operation.

Jaeger is inspired by Dapper and Zipkin, initiated by Uber.

Implemented by following OpenTracing [https://opentracing.io/docs/overview/](https://opentracing.io/docs/overview/)

We use **Istio** to help us gather traces. In Istio, spans will be generated by each envoy sidecar and send to Jaeger by default.
Distributed tracing is a **super powerful tool** to help with trouble shooting and performance analysis in real world operation.

Jaeger is inspired by Dapper and Zipkin, initiated by Uber.

Implemented by following OpenTracing [https://opentracing.io/docs/overview/](https://opentracing.io/docs/overview/)

We use **Istio** to help us gather traces. In Istio, spans will be generated by each envoy sidecar and send to Jaeger by default.

- **Data is well formatted & aligned by Istio**
- **Time series data**
- **Huge amount of data within short time period**
• Making deployments of machine learning workflows on Kubernetes simple, portable and scalable

• Support multiple ML frameworks
  • TensorFlow
  • Pytorch
  • Caffe

• Distributed training

• Models Serving

• **How we’re using it:**
  • Kubernetes cluster of 40 nodes
  • TensorFlow as backend
  • ~700,000 traces collected per day

From *Hidden Technical Debt in Machine Learning Systems*
- EDA (Exploratory data analysis)

- slice in minute

- last 30 minutes

- last 12 hours

- last 7 days

- requests in the last 7 days
• Duration anomaly detection
• Problem type
  • data is time series
  • there is no label
  • predict time series in the future
• Using LSTM to generate new sequence
• Find anomaly point based on prediction

Legend:
- Orange: Layer
- Yellow: Pointwise op
- Green: Copy

from WikiPedia Long short-term memory
Training & Modeling

Training dataset

Test dataset
- 990 data points in test data set
- 169 data points are anomaly
Training & Modeling

Model with data in 30 minutes

Model with data in 24 hours
Tuning microservices based on result

my-app

Scaling

traces

24h

Model 2

Scaling

30min

Model 1

Anomaly Detection

Responde & Fix

24h

Model 2

Anomaly Detection

Responde & Fix
Thank YOU !