TonY: An Orchestrator for Distributed Machine Learning Jobs

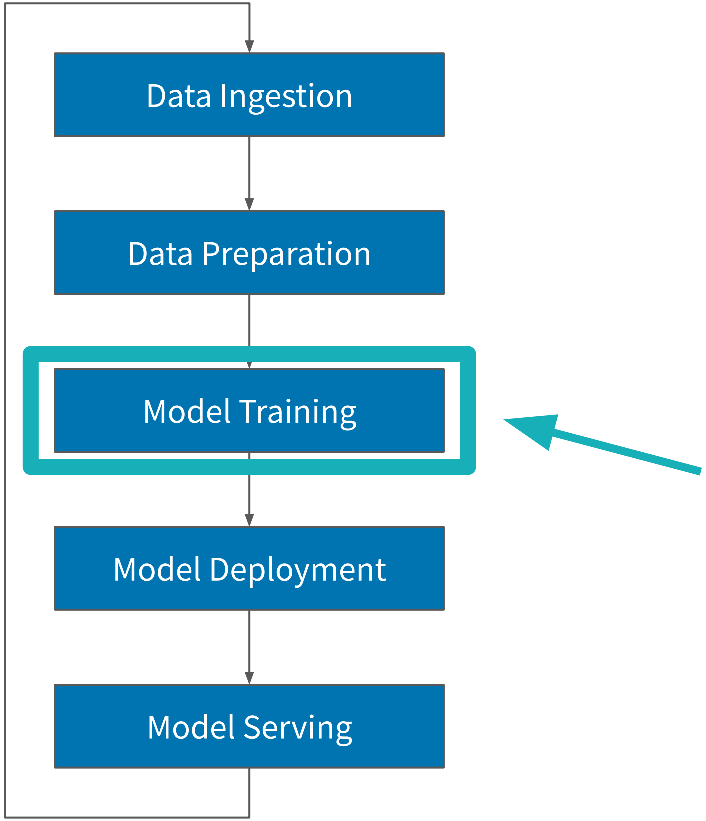
Jonathan Hung

LinkedIn

Agenda

- Background: TensorFlow and YARN
- What is TonY?
- Why use TonY for distributed training?
- Next steps

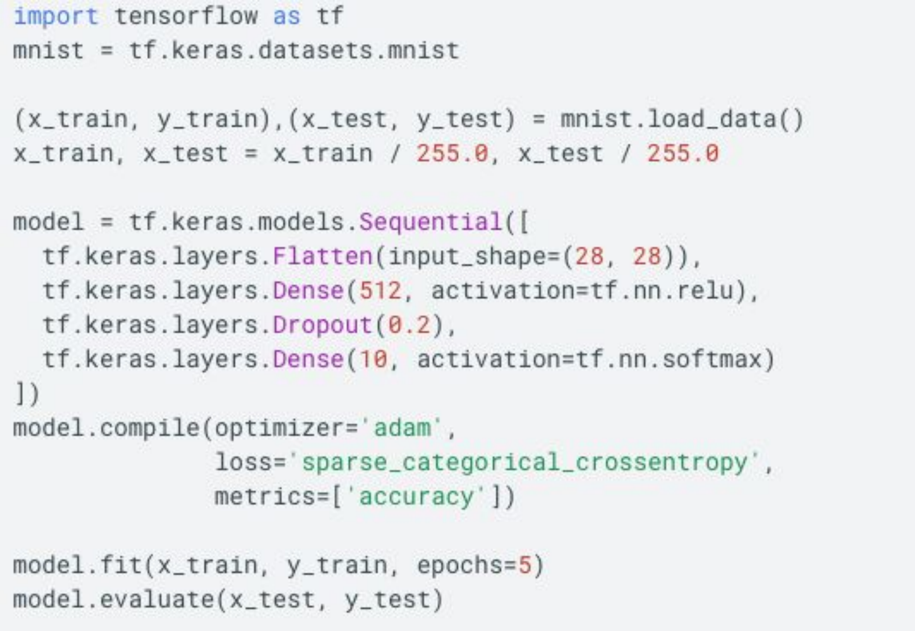
Machine Learning process



- Productionizing machine learning requires many steps
- The focus of this talk will be model training

Background

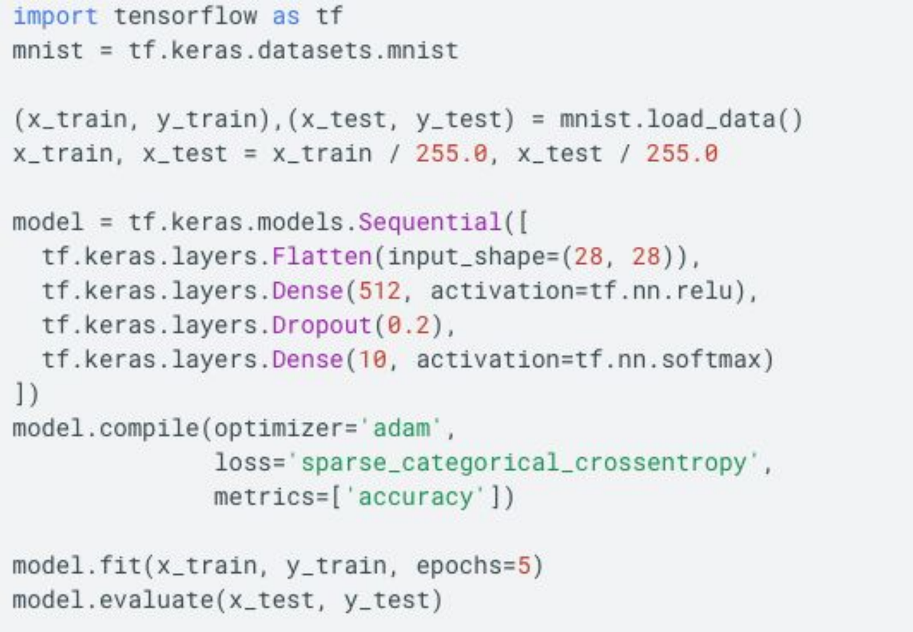
What is TensorFlow?



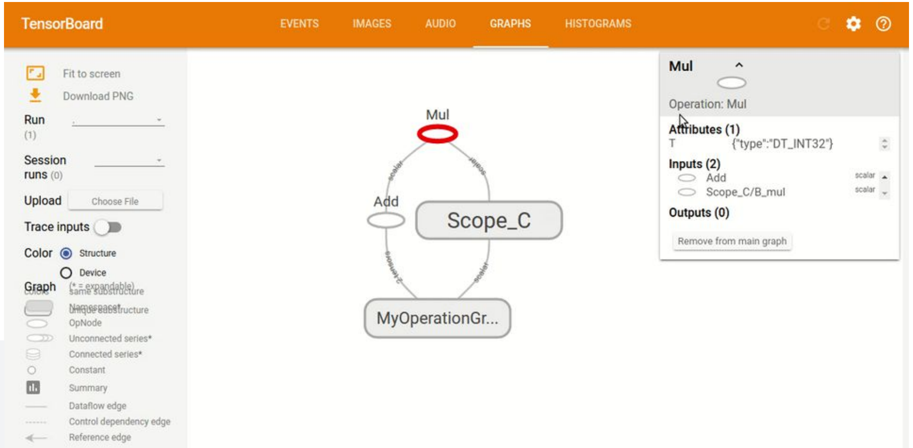
```python
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train),(x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(512, activation=tf.nn.relu),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```

Background

What is TensorFlow?



```
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train),(x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(512, activation=tf.nn.relu),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```



Run (1)

Session runs (0)

Upload

Trace inputs

Color ● Structure ○ Device

Graph (* = expandable)

Namespace*

OpNode

Unconnected series*

Connected series*

Constant

Summary

Dataflow edge

Control dependency edge

Reference edge

Mul

Operation: Mul

Attributes (1)

T {"type":"DT_INT32"}

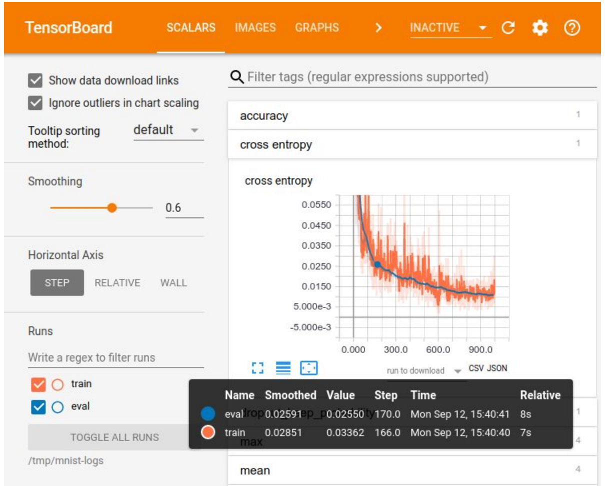
Inputs (2)

Add scalar

Scope_C/B_mul scalar

Outputs (0)

Visualisation with TensorBoard
https://learningtensorflow.com/Visualisation/



☑ Show data download links

☑ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0.6

Horizontal Axis STEP RELATIVE WALL

Runs

Write a regex to filter runs

☑ ○ train

☑ ○ eval

/tmp/mnist-logs

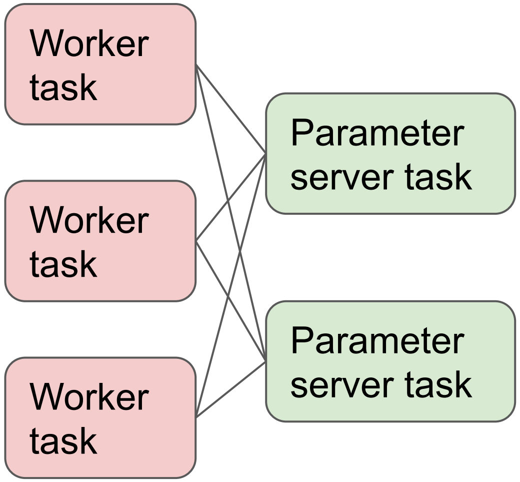
Filter tags (regular expressions supported)

	Name	Smoothed	Value	Step	Time	Relative
●	eval	0.02591	0.02550	170.0	Mon Sep 12, 15:40:41	8s
●	train	0.02851	0.03362	166.0	Mon Sep 12, 15:40:40	7s

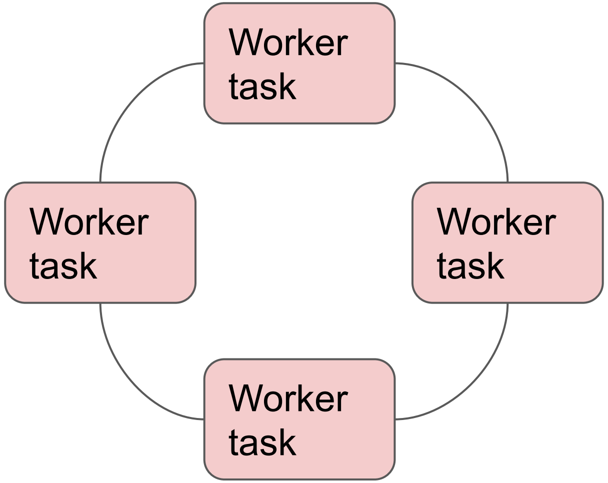
Background

What is distributed TensorFlow?

Worker + Parameter Server Model



Ring All-Reduce Model



How to run distributed TensorFlow?

- Distribute code/data artifacts across multiple machines in distributed job
- Allow tasks in the same distributed job to talk to each other (e.g. tell each worker where all other worker/parameter servers are)
- Ensure your task compute requirements are met before starting distributed job
- Or, have a framework do all of the above for you (Hadoop!)

Background

What is Hadoop?

Background

What is Hadoop?

Distributed File System

Background

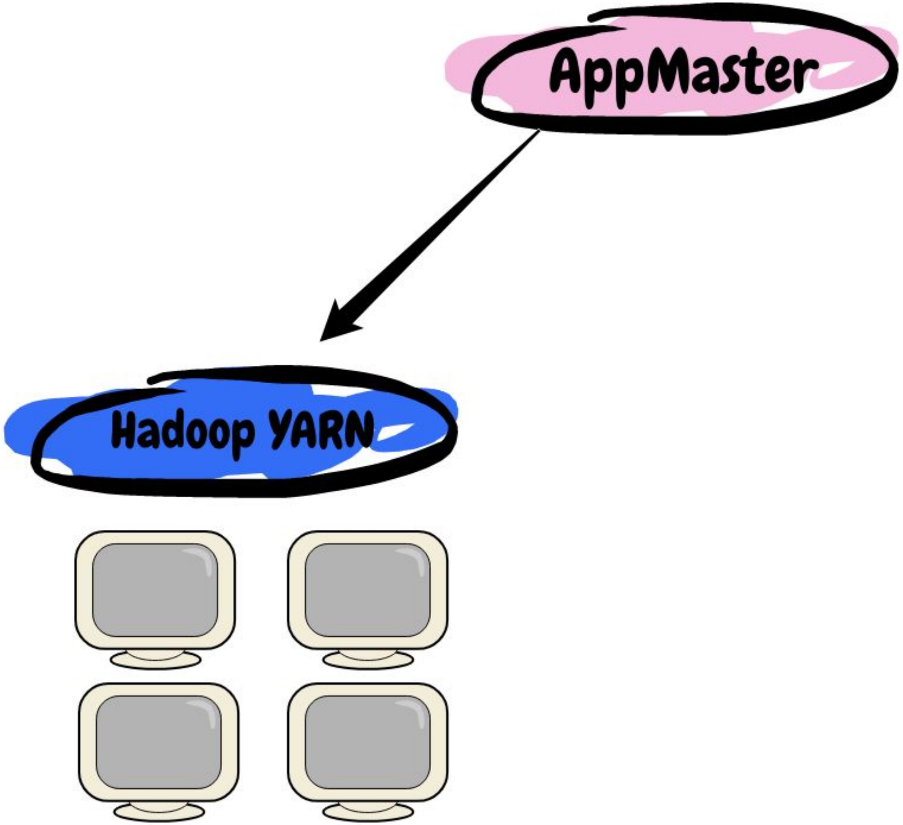
What is Hadoop?

Yet Another Resource Negotiator

Distributed File System

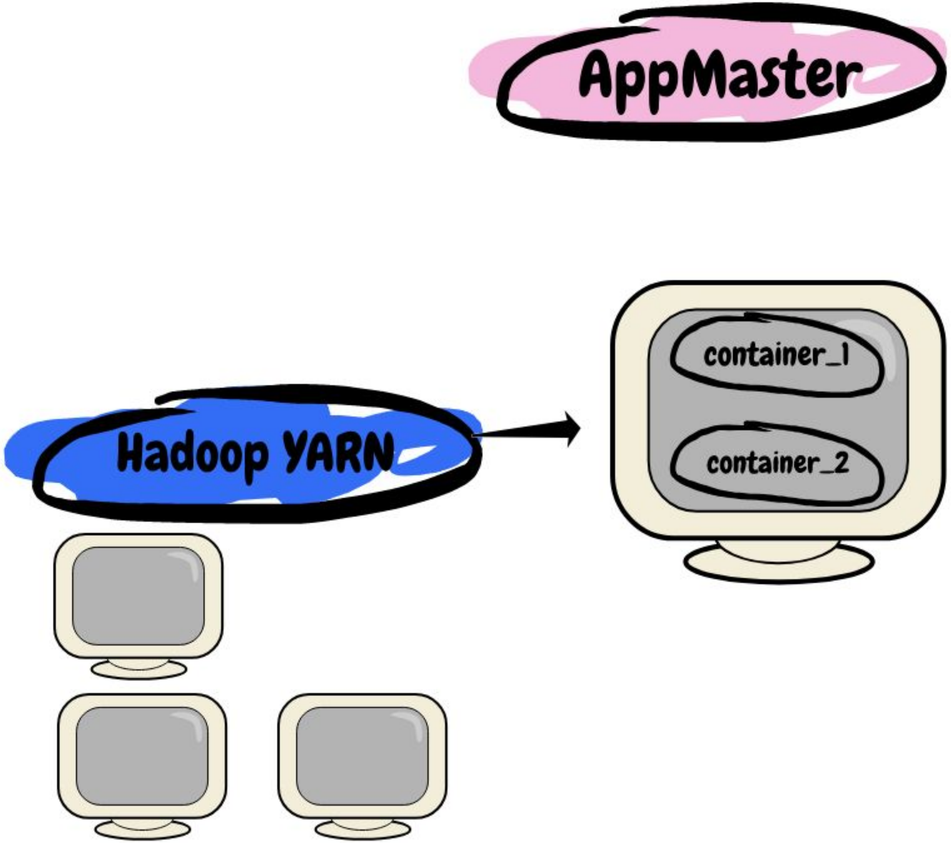
Background

How to work with YARN?



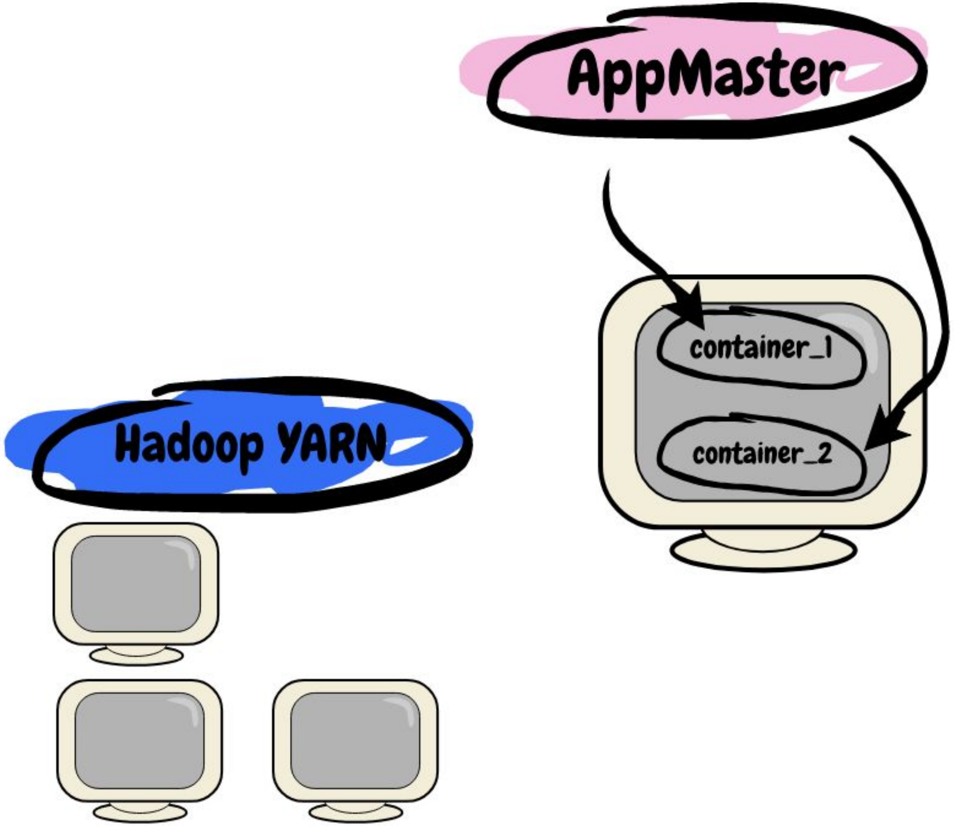
Background

How to work with YARN?



Background

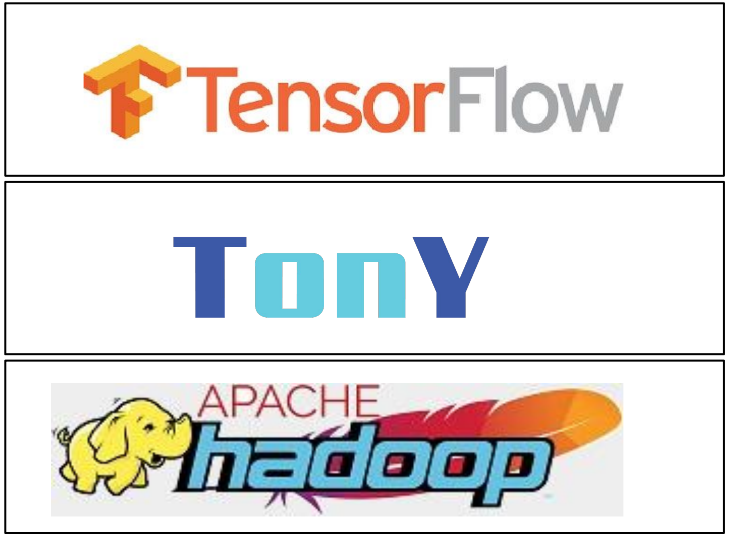
How to work with YARN?



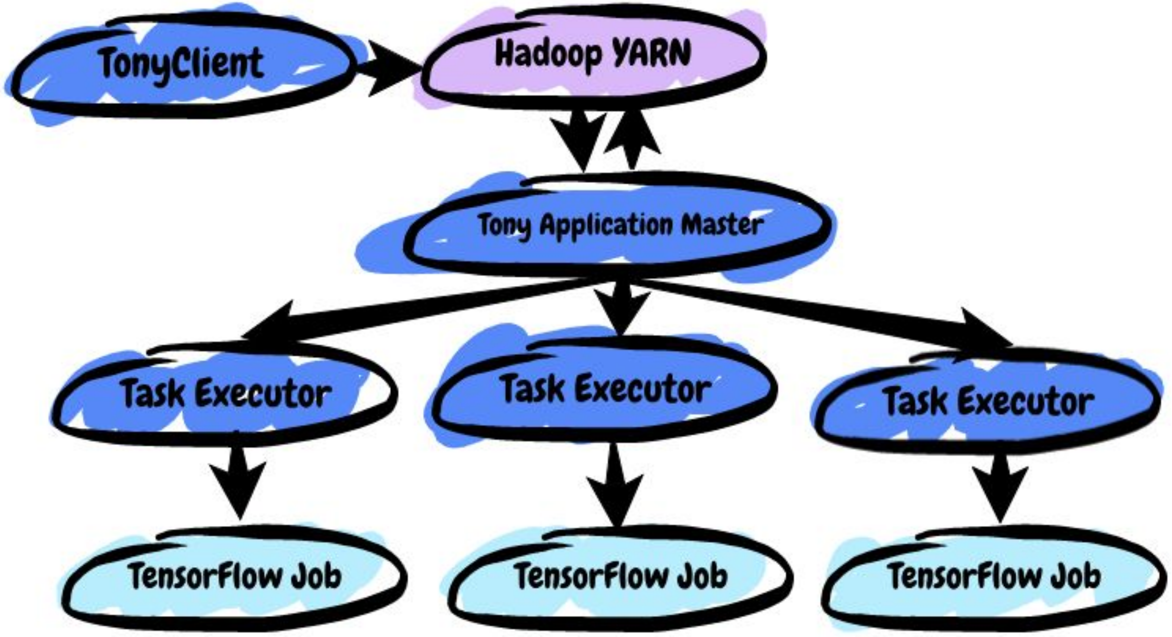
What is TonY?

What is TonY?

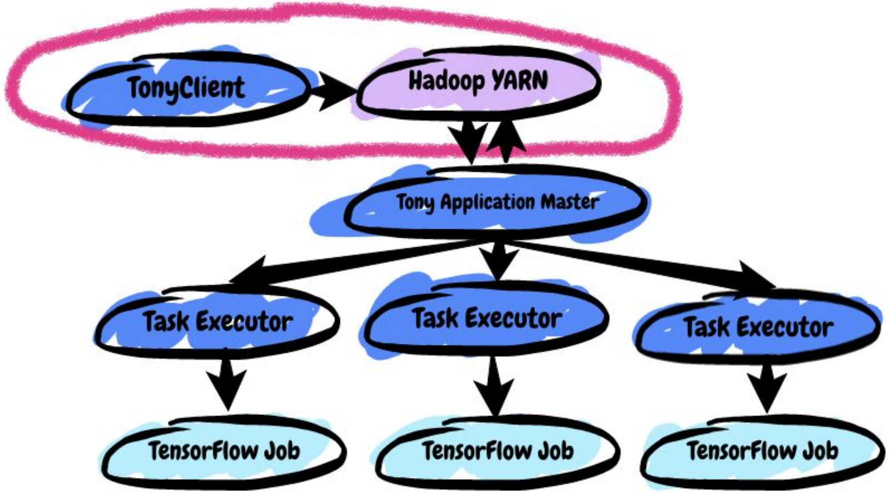
- Orchestrates running distributed TensorFlow on Hadoop
- Acquires compute resources from Hadoop (memory, CPU, GPU)
- Sets up and launches distributed TensorFlow jobs on Hadoop clusters
- Manages application lifecycle
  - Fault tolerance
  - Job monitoring



TonY Architecture

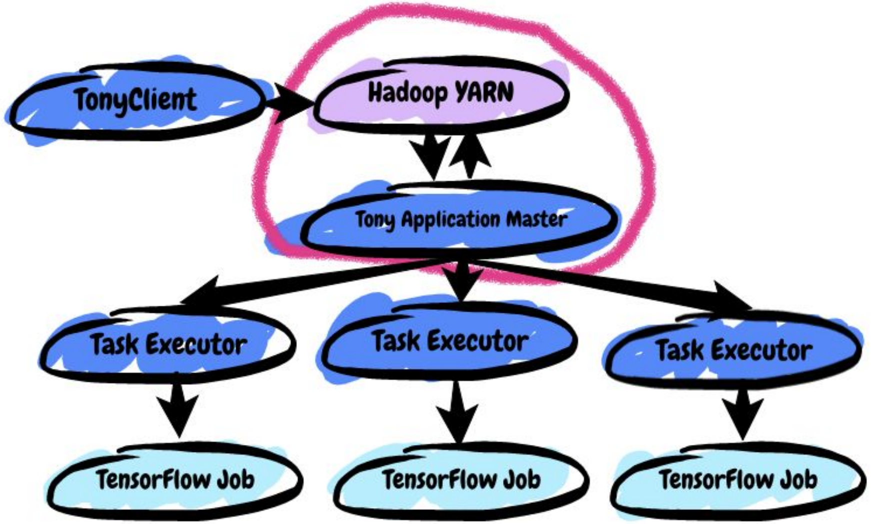


TonY Architecture



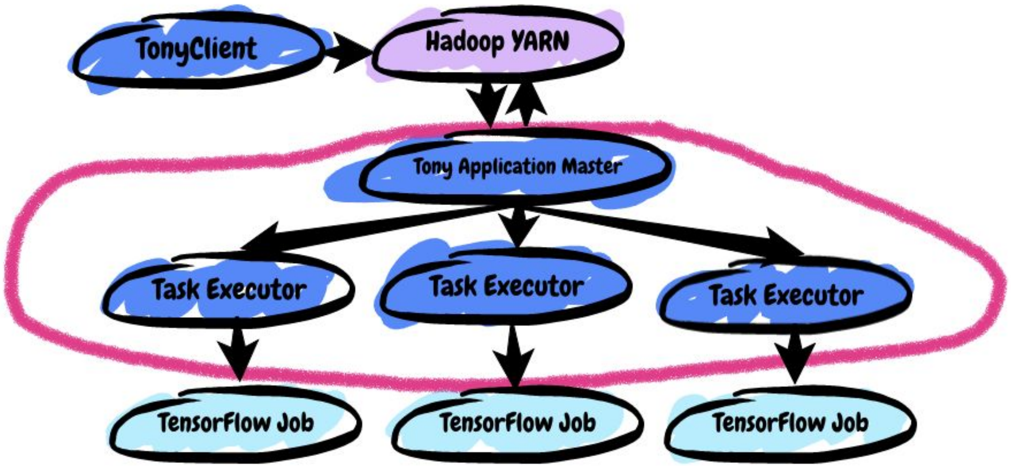
- Entry point for TonY jobs
- Package user's configurations, user's model code and submit as YARN application

TonY Architecture



- Job setup and lifecycle management
- Negotiates compute resources from Hadoop
- Sets up container environment
- Launches and monitors containers

TonY Architecture

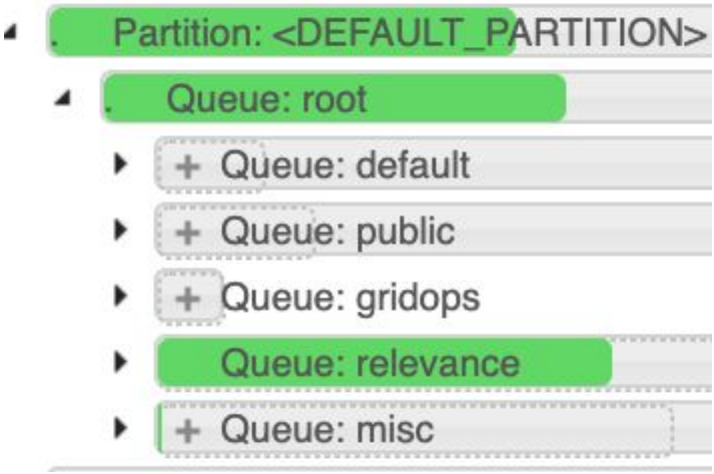


- Container = Task Executor
- Launches user’s provided python script
- Heartbeats to Application Master for liveness

Why use TonY for distributed training?

Scaling distributed TensorFlow on Hadoop

- Leverage YARN’s fine-grained resource management and multi-tenancy
  - Logical resource isolation via queues
  - Hardware-based physical resource partitioning (CPU, K80, V100)
  - User-based resource limits



Partition: <DEFAULT_PARTITION>

Queue: root

+ Queue: default

+ Queue: public

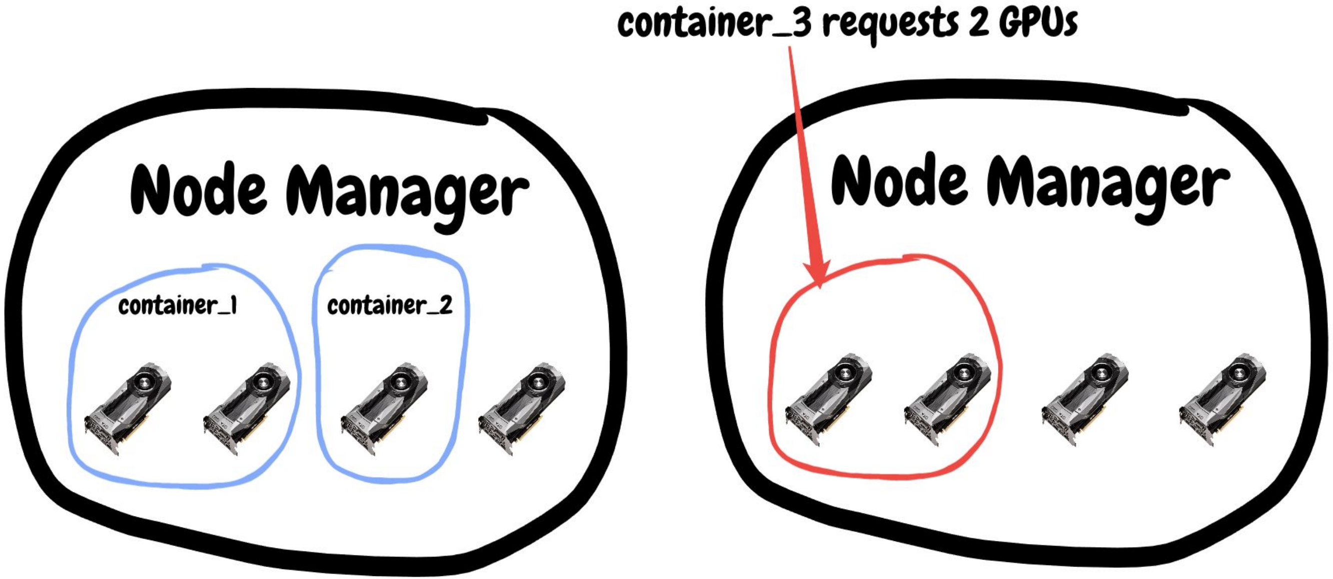
+ Queue: gridops

Queue: relevance

+ Queue: misc

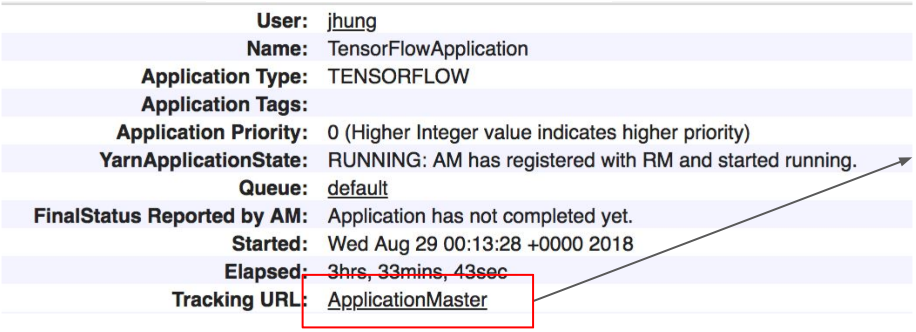
Scaling distributed TensorFlow on Hadoop

- Native GPU resource awareness
- Ensures GPU resource isolation and scheduling



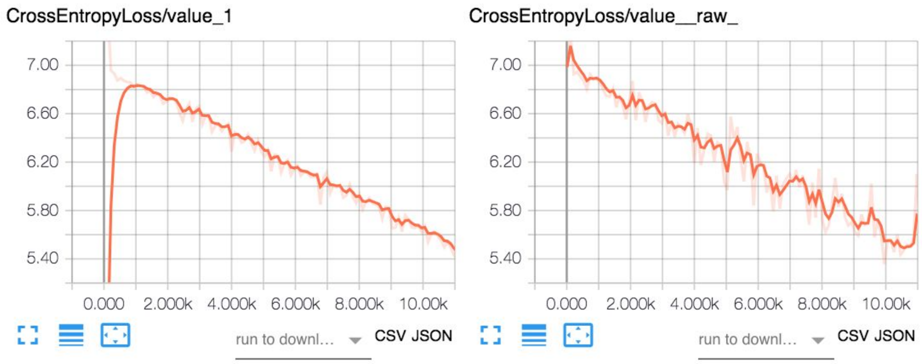
Scaling distributed TensorFlow on Hadoop

- One-click TensorBoard access for monitoring training progress



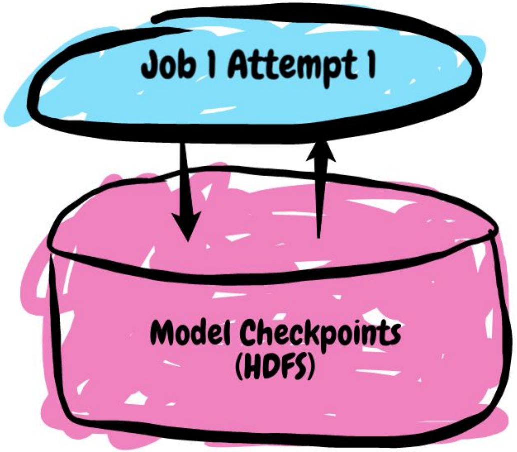
User:	jhung
Name:	TensorFlowApplication
Application Type:	TENSORFLOW
Application Tags:	
Application Priority:	0 (Higher Integer value indicates higher priority)
YarnApplicationState:	RUNNING: AM has registered with RM and started running.
Queue:	default
FinalStatus Reported by AM:	Application has not completed yet.
Started:	Wed Aug 29 00:13:28 +0000 2018
Elapsed:	3hrs, 33mins, 43sec
Tracking URL:	ApplicationMaster

TensorBoard



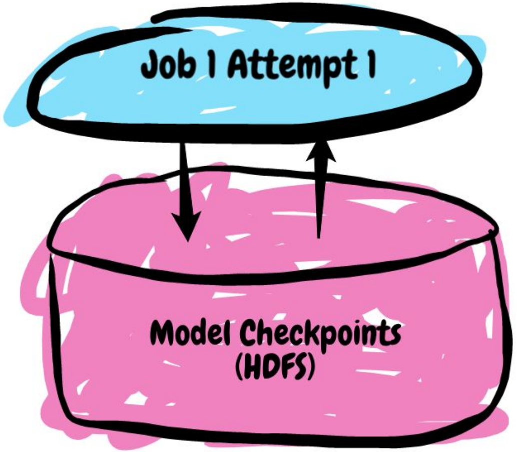
Scaling distributed TensorFlow on Hadoop

- Fault tolerance
- More workers = more failures



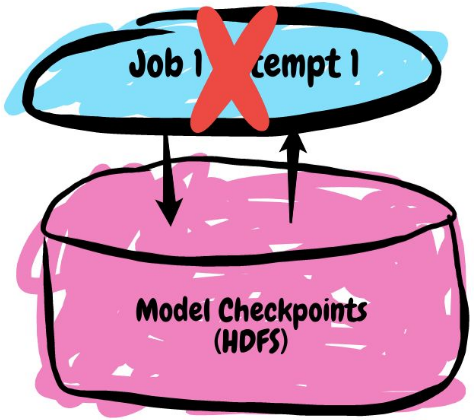
Scaling distributed TensorFlow on Hadoop

- Fault tolerance
- More workers = more failures
- First attempt periodically saves model checkpoints to HDFS



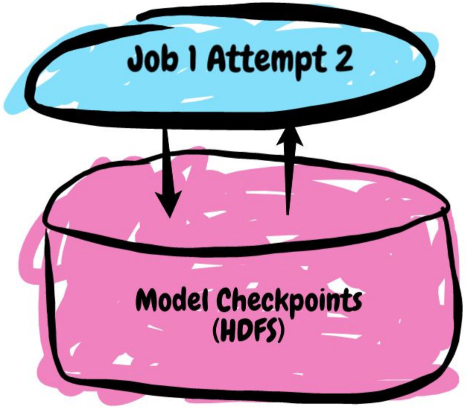
Scaling distributed TensorFlow on Hadoop

- Fault tolerance
- More workers = more failures
- First attempt periodically saves model checkpoints to HDFS
- Worker failure -> tear down and restart application



Scaling distributed TensorFlow on Hadoop

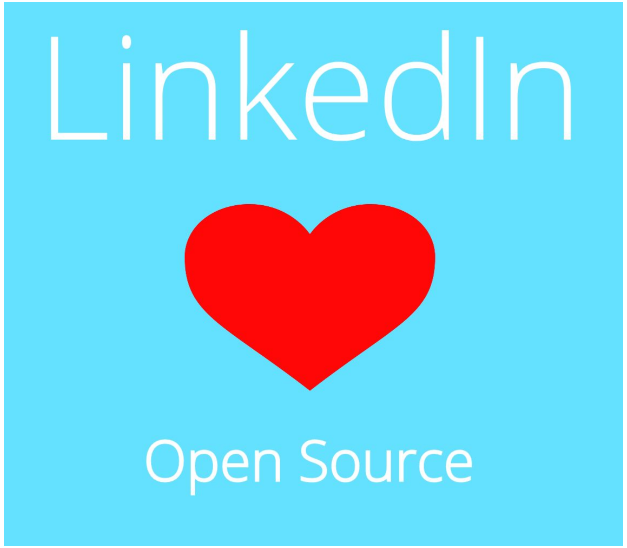
- Fault tolerance
- More workers = more failures
- First attempt periodically saves model checkpoints to HDFS
- Worker failure -> tear down and restart application
- Read checkpoints from HDFS, resume from where previous attempt left off



Open Sourced!

- https://github.com/linkedin/TonY
- Engineering blog post: https://bit.ly/2O6L5WD

Contributions Welcome!



Next steps

- Dr. Elephant integration
- TonY portal for notebook, job history, cross-execution monitoring

Q & A