饿了么

TensorFlow 深度学习平台

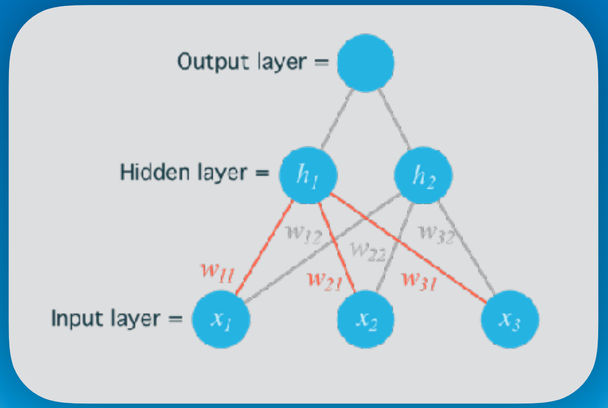
elearn

饿了么资深后端工程师　江骏

AI

Machine Learning

Deep Learning

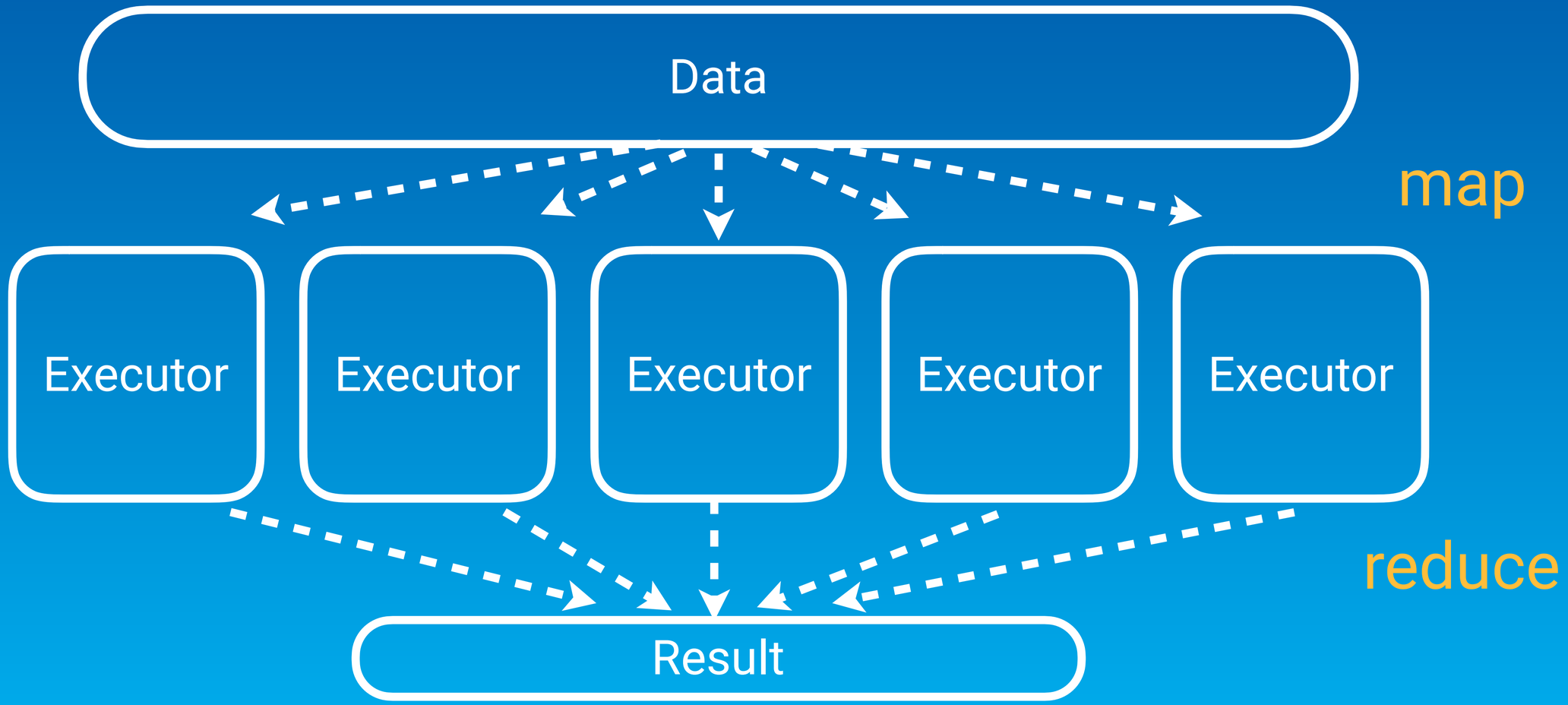


Hadoop?

Spark?

MapReduce

Hadoop & Spark



MapReduce

Hadoop & Spark

“In parallel computing, an **embarrassingly parallel** workload or problem (also called **perfectly parallel** or **pleasingly parallel**) is one where little or no effort is needed to separate the problem into a number of parallel tasks.”

—— Wikipedia

https://en.wikipedia.org/wiki/Embarrassingly_parallel

Spark MLlib

org.apache.spark.mllib.classification hide focus

- ClassificationModel
- LogisticRegressionModel
- LogisticRegressionWithLBFGS
- LogisticRegressionWithSGD
- NaiveBayes
- NaiveBayesModel
- StreamingLogisticRegressionWithSGD
- SVMModel
- SVMWithSGD

org.apache.spark.mllib.clustering hide focus

- BisectingKMeans
- BisectingKMeansModel
- DistributedLDAModel
- EMLDAOptimizer
- GaussianMixture
- GaussianMixtureModel
- KMeans
- KMeansModel
- LDA
- LDAModel
- LDAOptimizer
- LocalLDAModel
- OnlineLDAOptimizer
- PowerIterationClustering
- PowerIterationClusteringModel
- StreamingKMeans
- StreamingKMeansModel

org.apache.spark.mllib.tree hide focus

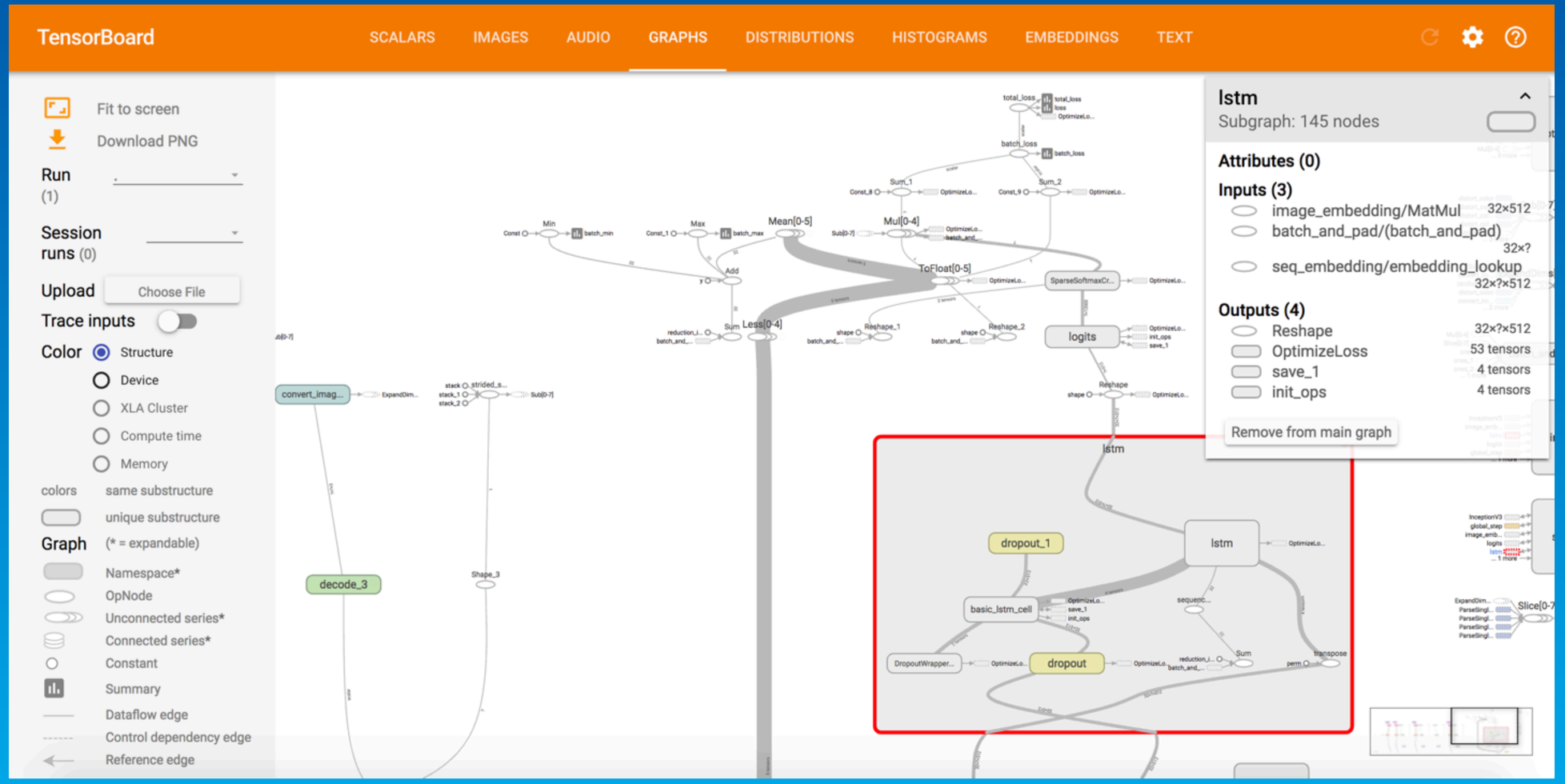
- DecisionTree
- GradientBoostedTrees
- RandomForest

org.apache.spark.mllib.regression hide focus

- GeneralizedLinearAlgorithm
- GeneralizedLinearModel
- IsotonicRegression
- IsotonicRegressionModel
- LabeledPoint
- LassoModel
- LassoWithSGD
- LinearRegressionModel
- LinearRegressionWithSGD
- RegressionModel
- RidgeRegressionModel
- RidgeRegressionWithSGD
- StreamingLinearAlgorithm
- StreamingLinearRegressionWithSGD

What about TensorFlow?

TensorFlow Graph



Training Data

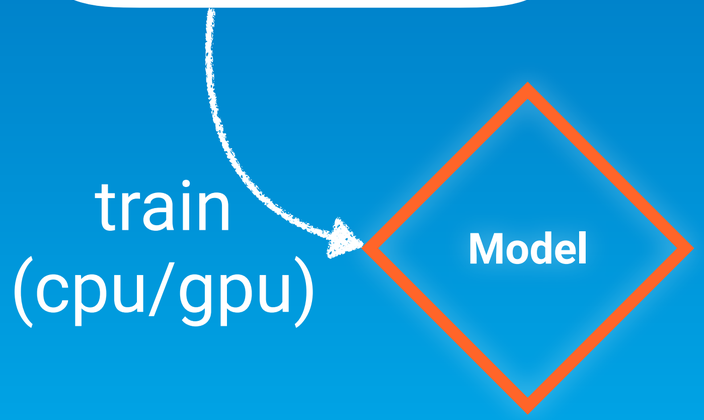
Validation Data

Test Data

Training Data

Validation Data

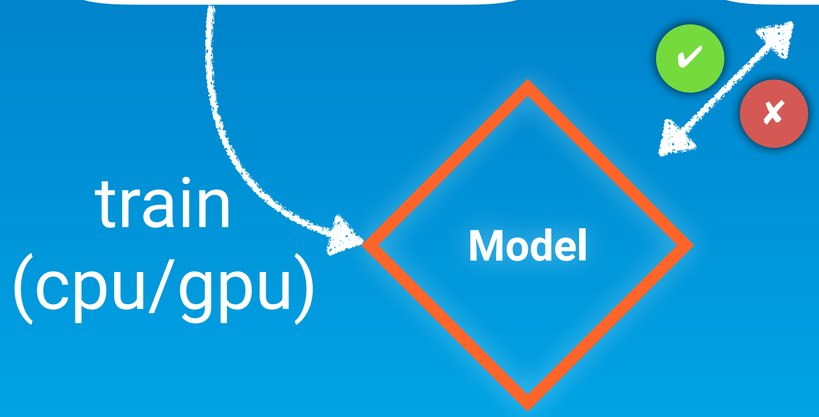
Test Data

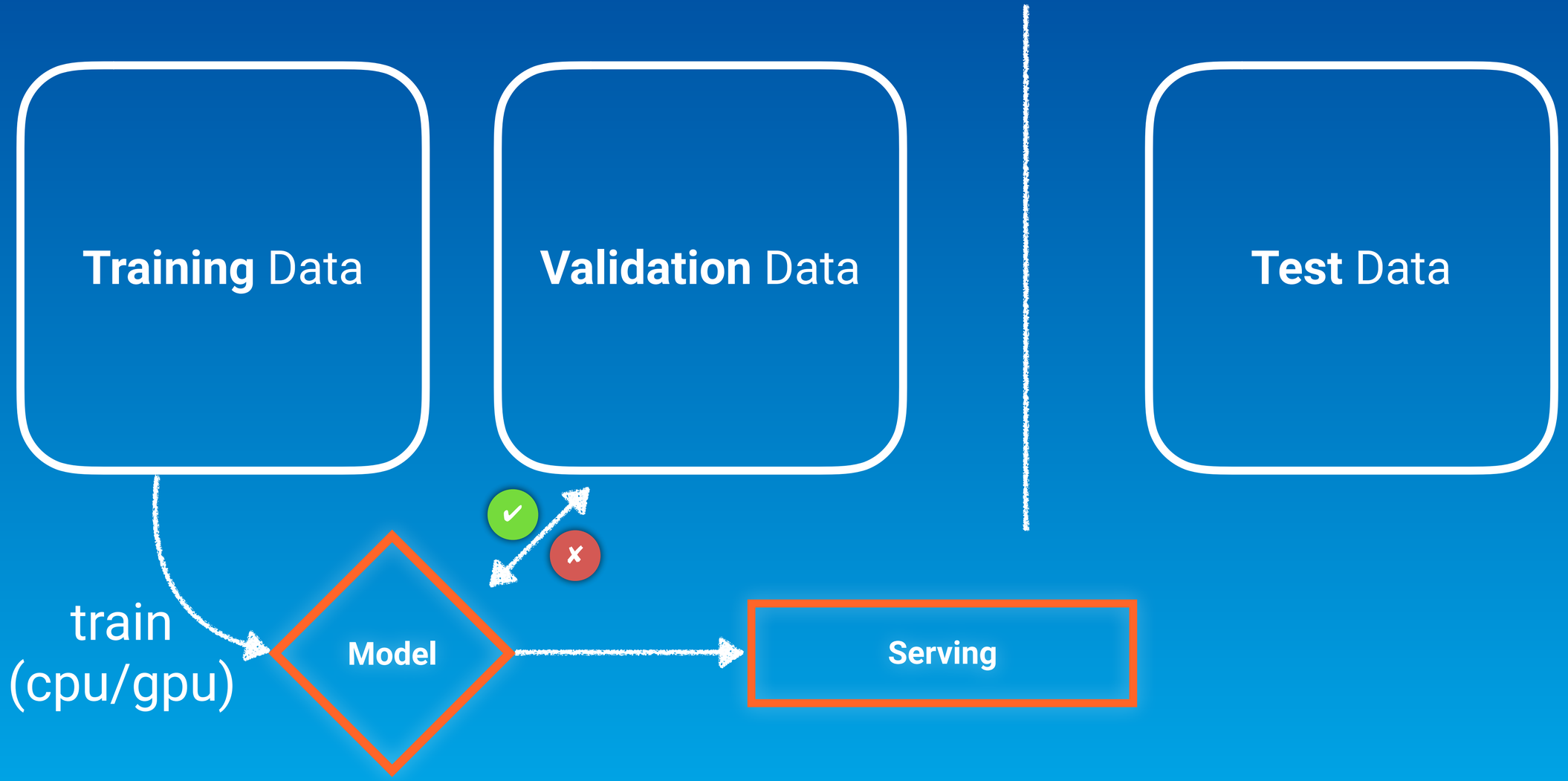


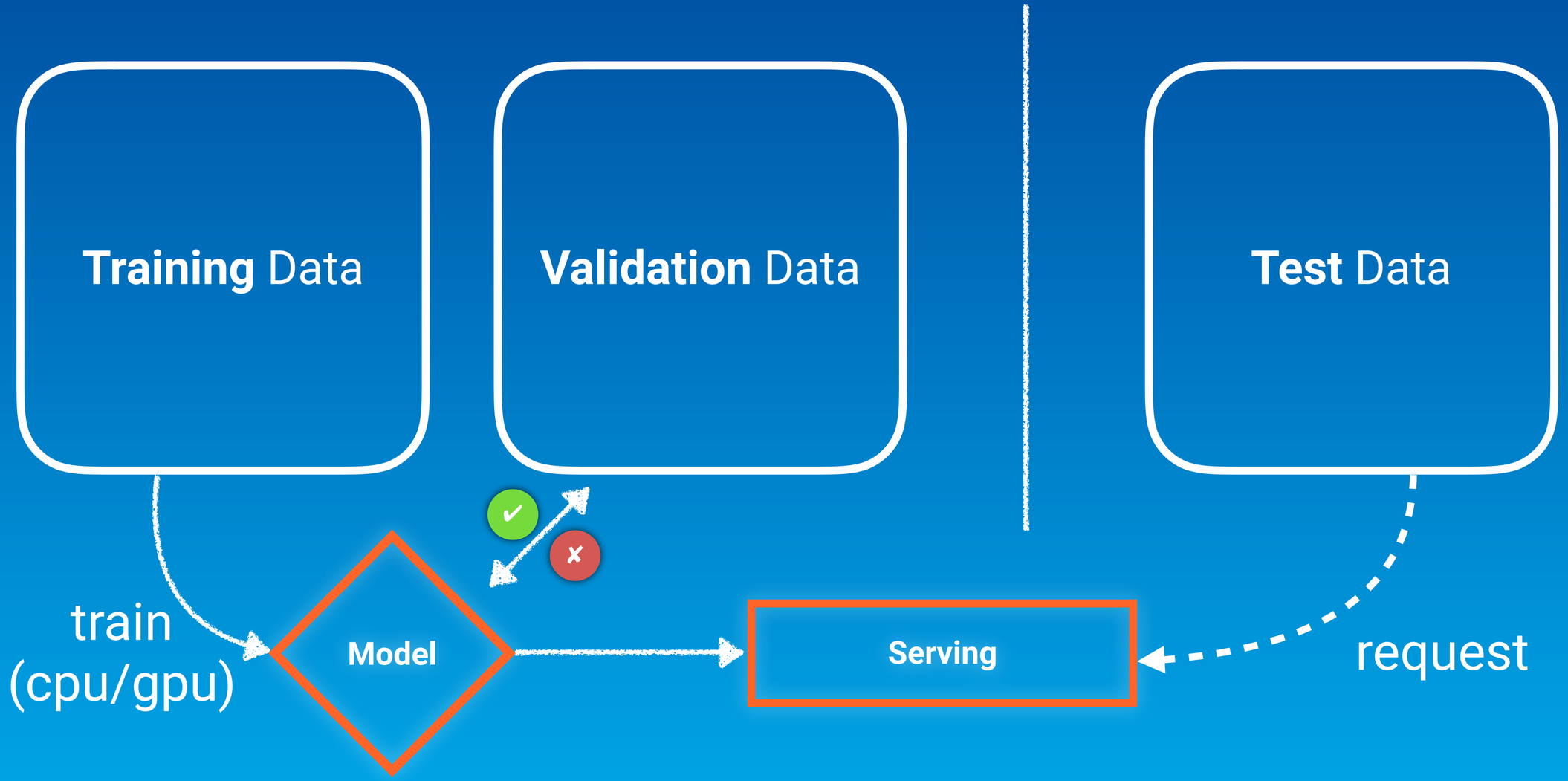
Training Data

Validation Data

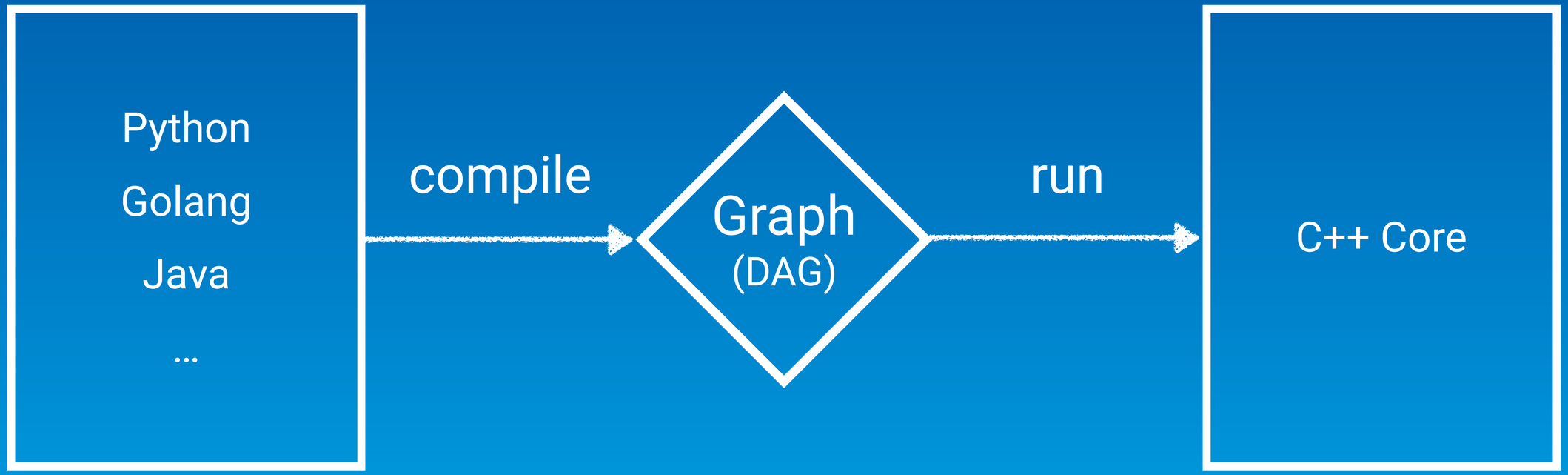
Test Data



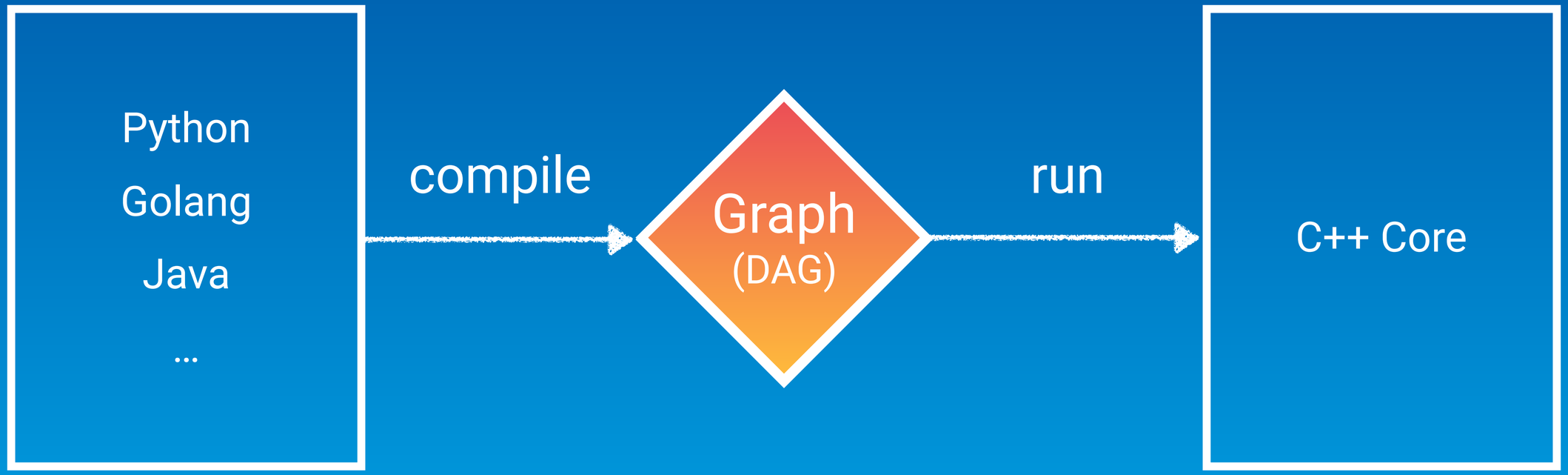




How does TensorFlow work

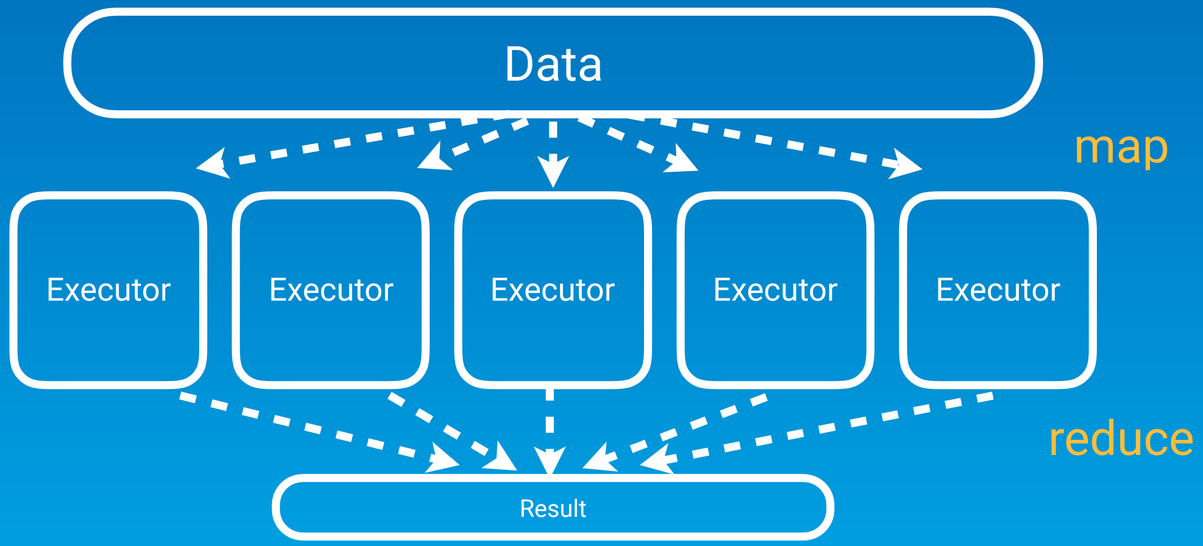


How does TensorFlow work



Preprocess

Hive, Spark, Storm, ...

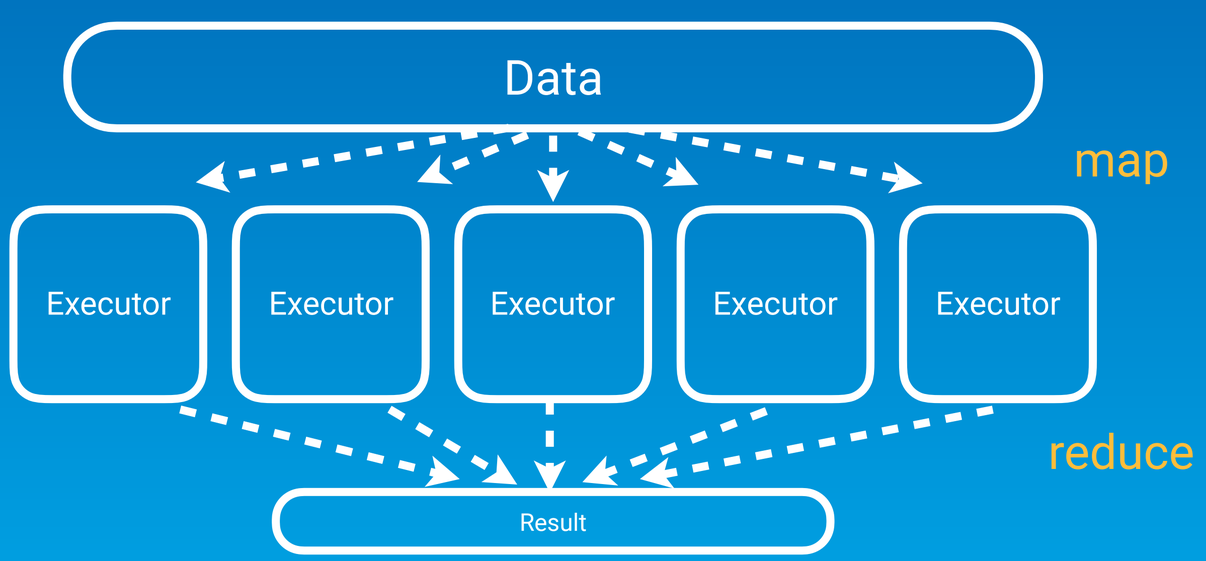


Preprocess

Hive, Spark, Storm, ...

Distributed Storage

NFS, HDFS, S3, ...



Preprocess

Hive, Spark, Storm, ...

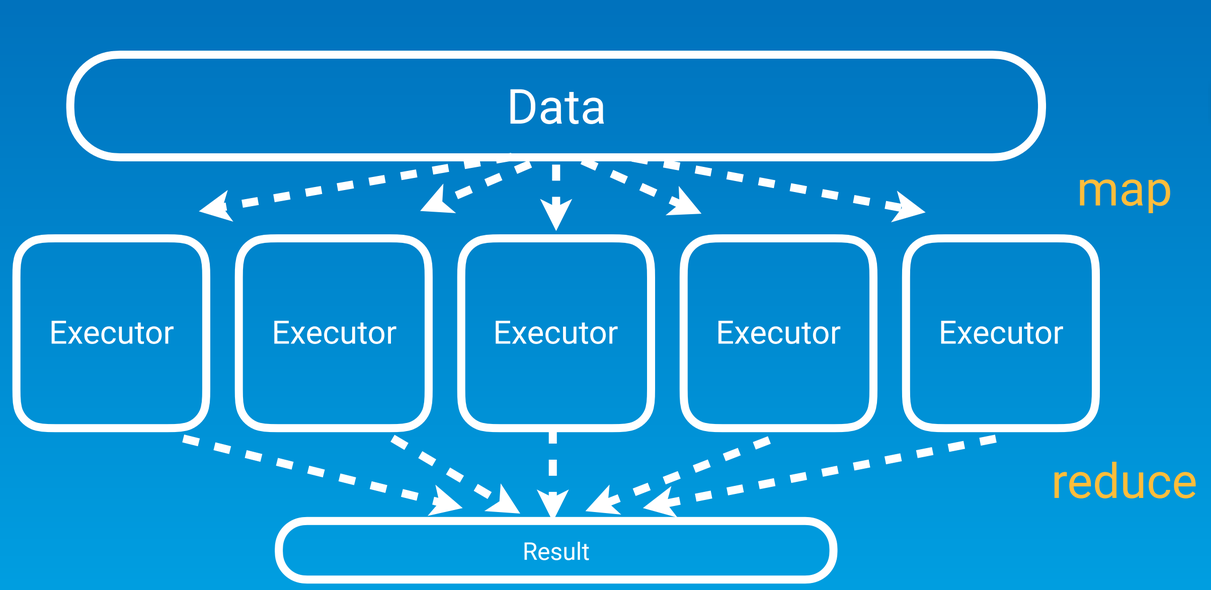
Distributed Storage

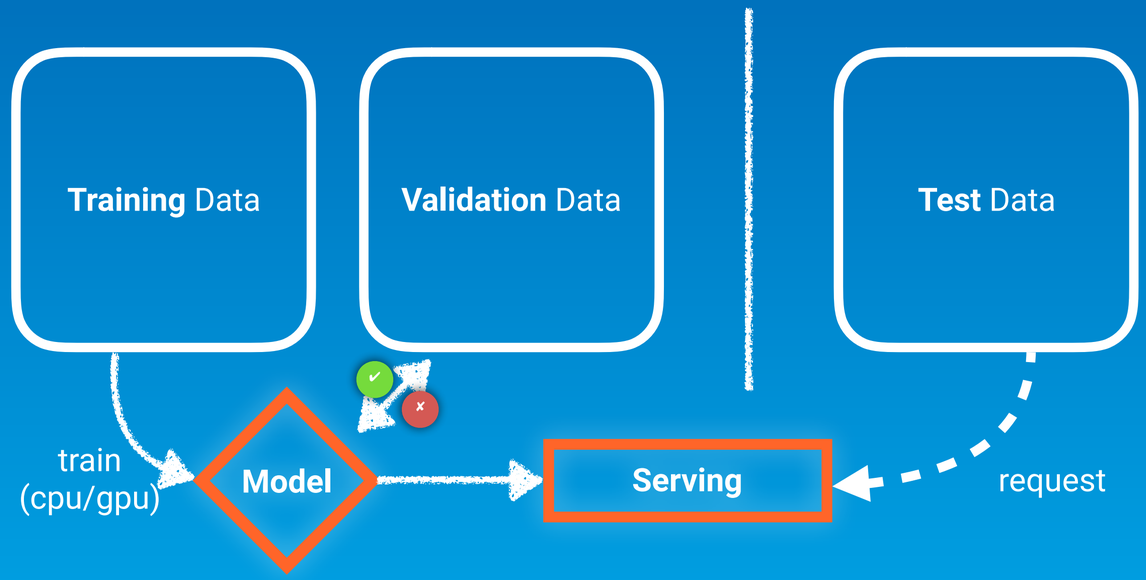
NFS, HDFS, S3, ...

Training

TensorFlow

(Distributed Training)





elearn

caicloud 才云 cloud acceleration intelligence.

IT大咖说 知识分享平台

Preprocess

Hive, Spark, Storm, ...

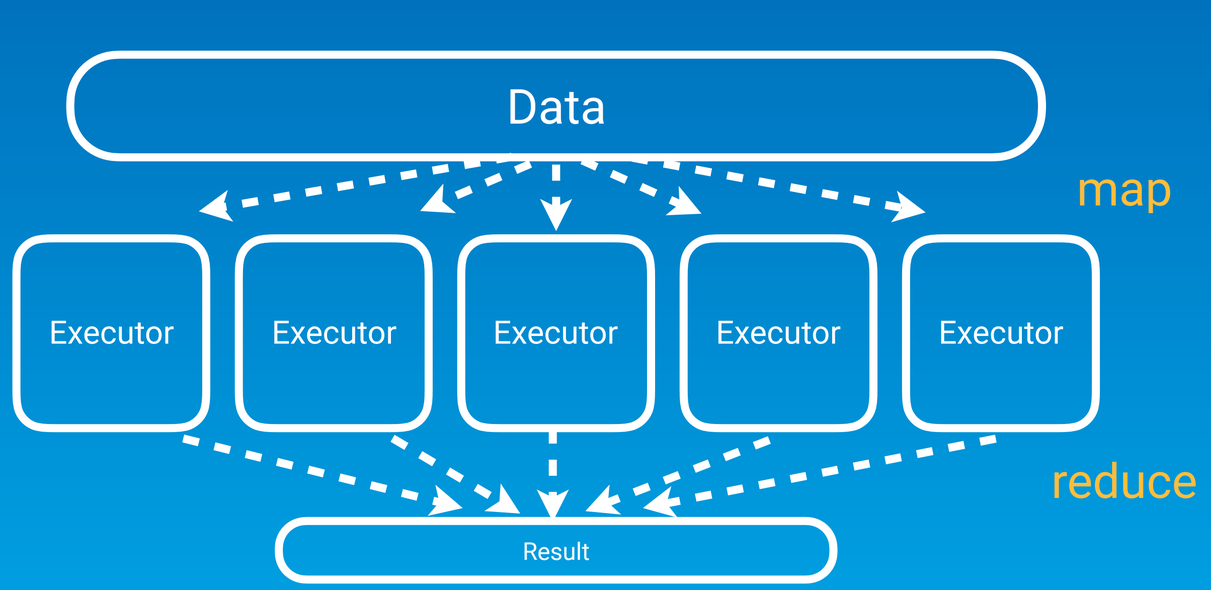
Distributed Storage

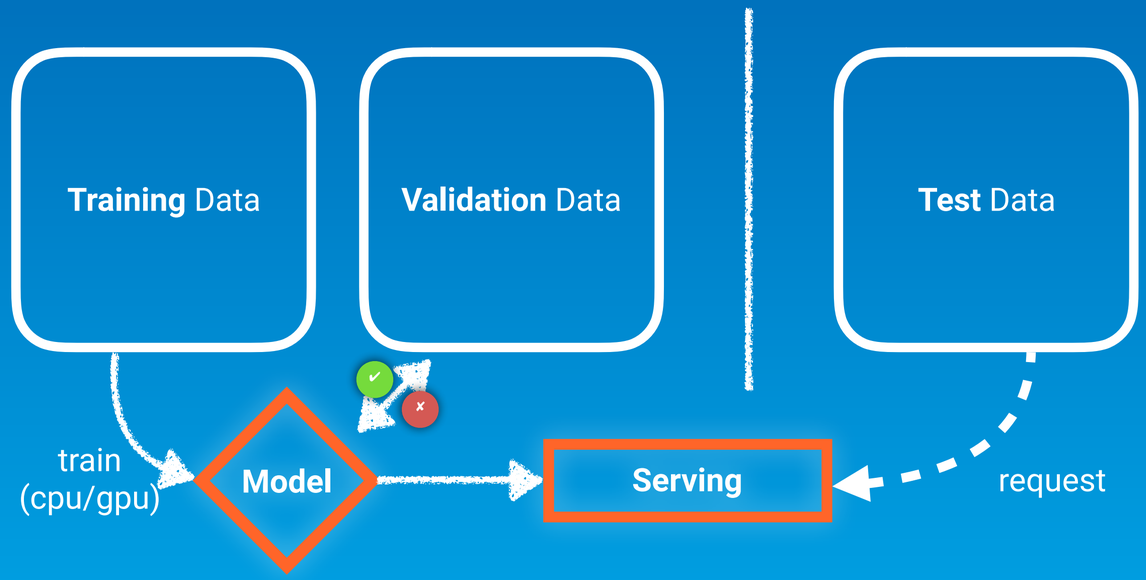
NFS, HDFS, S3, ...

Training

TensorFlow

(Distributed Training)





elearn

- **TaaS** (TensorFlow as a Service)
- 开始于 2016 年 8 月底
- 受到 Google Cloud 的 CloudML 产品启发
- 让算法工程师可以专注于算法，其它的事情交给 elearn

elearn

- **TaaS** (TensorFlow as a Service)
- 开始于 2016 年 8 月底
- 受到 Google Cloud 的 CloudML 产品启发
- 让算法工程师可以专注于算法，其它的事情交给 elearn

CPU 弹性需求　service 的 IP、Port 管理

分布式存储　大量 container 的生命周期　→　API

GPU

TensorBoard 等配套服务

Distributed
Storage

Datastore

- Abstraction of data
- Compatible with many types of distributed storage
- Decoupling computing and storage
- Bring your own storage

Datastore

	Client	缺点	应用场景
NFS	nfs-utils	性能问题、权限管理	存放训练数据 模型训练
S3 compatible	s3fs-fuse	无法创建空目录 mkdir	存放训练数据
	MinFS	无法创建 symlink	存放训练数据
GlusterFS	-	-	-

Datastore Example

```
{
  "kind": "datastore",
  "metadata": {
    "name": "demo-nfs-output"
  },
  "spec": {
    "nfs": {
        "server": "[redacted]",
        "path": "/d[redacted]0001/output"
    }
  }
}
```

NFS

```
{
  "kind": "datastore",
  "metadata": {
    "name": "demo-s3"
  },
  "spec": {
    "s3": {
        "server": "http://10.0.[redacted]",
        "access_key": "DGKKUVCZ[redacted]",
        "secret_key": "oPdB3X53[redacted]xzoTt8AQz0rB+fIfDX1t",
        "bucket": "tmp"
    }
  }
}
```

S3

GPU
in container

```
import os
os.environ["CUDA_DEVICE_ORDER"]="PCI_BUS_ID"
os.environ["CUDA_VISIBLE_DEVICES"]="0"
```

```
Processes:                                                       GPU Memo
 GPU       PID  Type  Process name                               Usage
=============================================================================
   0      5552    C   python                                     11657MiB |
   1      5552    C   python                                     11655MiB |
   2      5552    C   python                                     11587MiB |
   3      5552    C   python                                     11587MiB |
   4      5552    C   python                                     11587MiB |
   5      5552    C   python                                     11587MiB |
   6      5552    C   python                                     11587MiB |
   7      5552    C   python                                     11587MiB |
   8      5552    C   python                                     11587MiB |
-----------------------------------------------------------------------------+
```

有人把GPU全部跑满了~~

我去看看

我只用一个GPU就好啦~

比较坑爹是，找不出是谁

我在找

```
[root@wg-bdi-vision-1 jun.jiang02]# nvidia-smi
Mon Jun  5 16:12:43 2017
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 375.66                 Driver Version: 375.66                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  TITAN X (Pascal)    Off  | 0000:04:00.0     Off |                  N/A |
| 23%   31C    P0    56W / 250W |      0MiB / 12189MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  TITAN X (Pascal)    Off  | 0000:05:00.0     Off |                  N/A |
| 23%   35C    P0    59W / 250W |      0MiB / 12189MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   2  TITAN X (Pascal)    Off  | 0000:06:00.0     Off |                  N/A |
| 23%   34C    P0    57W / 250W |      0MiB / 12189MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   3  TITAN X (Pascal)    Off  | 0000:07:00.0     Off |                  N/A |
| 23%   36C    P0    59W / 250W |      0MiB / 12189MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   4  TITAN X (Pascal)    Off  | 0000:08:00.0     Off |                  N/A |
| 23%   34C    P0    59W / 250W |      0MiB / 12189MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   5  TITAN X (Pascal)    Off  | 0000:0B:00.0     Off |                  N/A |
| 23%   35C    P0    56W / 250W |      0MiB / 12189MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   6  TITAN X (Pascal)    Off  | 0000:0C:00.0     Off |                  N/A |
| 23%   35C    P0    57W / 250W |      0MiB / 12189MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   7  TITAN X (Pascal)    Off  | 0000:0D:00.0     Off |                  N/A |
| 23%   36C    P0    57W / 250W |      0MiB / 12189MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   8  TITAN X (Pascal)    Off  | 0000:0E:00.0     Off |                  N/A |
| 23%   39C    P0    57W / 250W |      0MiB / 12189MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   9  TITAN X (Pascal)    Off  | 0000:0F:00.0     Off |                  N/A |
| 23%   37C    P0    57W / 250W |      0MiB / 12189MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID  Type  Process name                               Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

host GPU

```
[root@wg-bdi-vision-1 jun.jiang02]# dt ssh b2ebd5eaaefe
Enter container 'b2ebd5eaaefe', using /bin/bash
root@gpu-pod:/notebooks# nvidia-smi
Mon Jun  5 08:13:18 2017
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 375.66                 Driver Version: 375.66                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  TITAN X (Pascal)    Off  | 0000:0D:00.0     Off |                  N/A |
|  0%   36C    P0    53W / 250W |      0MiB / 12189MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID  Type  Process name                               Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

container GPU

GPU with Docker

- GPU 内存使用有讲究
  https://www.tensorflow.org/tutorials/using_gpu
- GPU Docker images
  https://hub.docker.com/r/nvidia/cuda/
- nvidia-docker 不是必须
  CUDA_SO, NVIDIA_SO, DEVICES, ...

GPU with Kubernetes

```
--feature-gates="Accelerators=true"
```

```yaml
spec:
  containers:
    -
      name: gpu-container-1
      resources:
        limits:
          alpha.kubernetes.io/nvidia-gpu: 2 # requesting 2 GPUs
```

Kubernetes 只是把 GPU 选好，放进 container

要真正用起来，elearn 接管了剩下的

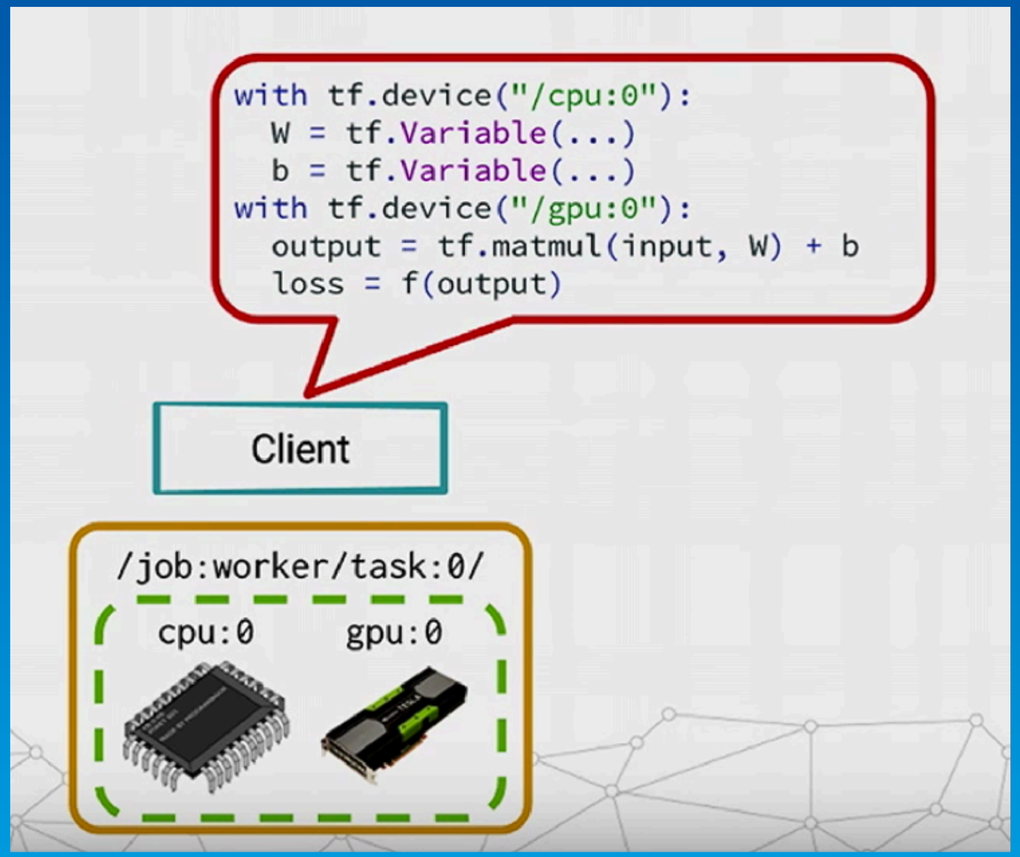
Distributed **Training**

TensorFlow 的上一代产品

DistBelief

- 解决了数据量 > GPU 最大内存的问题
- paper 中提到最大的 model，达到了
  1.7 billion parameters, utilizing 81 machines, delivering a 12x speedup.
- 它的缺点：
  - 擅长图像识别，但对其它机器学习 model 适用性差，而且不支持 mobile
  - 维护大规模系统成为负担，缺乏抽象。

https://research.google.com/pubs/pub40565.html

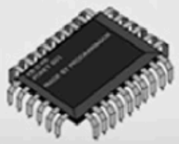


```
with tf.device("/cpu:0"):
  W = tf.Variable(...)
  b = tf.Variable(...)
with tf.device("/gpu:0"):
  output = tf.matmul(input, W) + b
  loss = f(output)
```

Client

/job:worker/task:0/

cpu:0 gpu:0



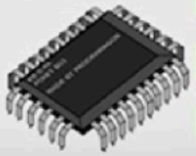

Distributed TensorFlow

```
with tf.device("/job:ps/task:0/cpu:0"):
  W = tf.Variable(...)
  b = tf.Variable(...)
with tf.device("/job:worker/task:0/gpu:0"):
  output = tf.matmul(input, W) + b
  loss = f(output)
```

Client

/job:worker/task:0/

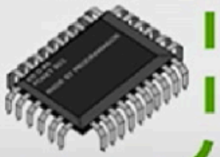
cpu:0 gpu:0




gRPC

/job:ps/task:0/

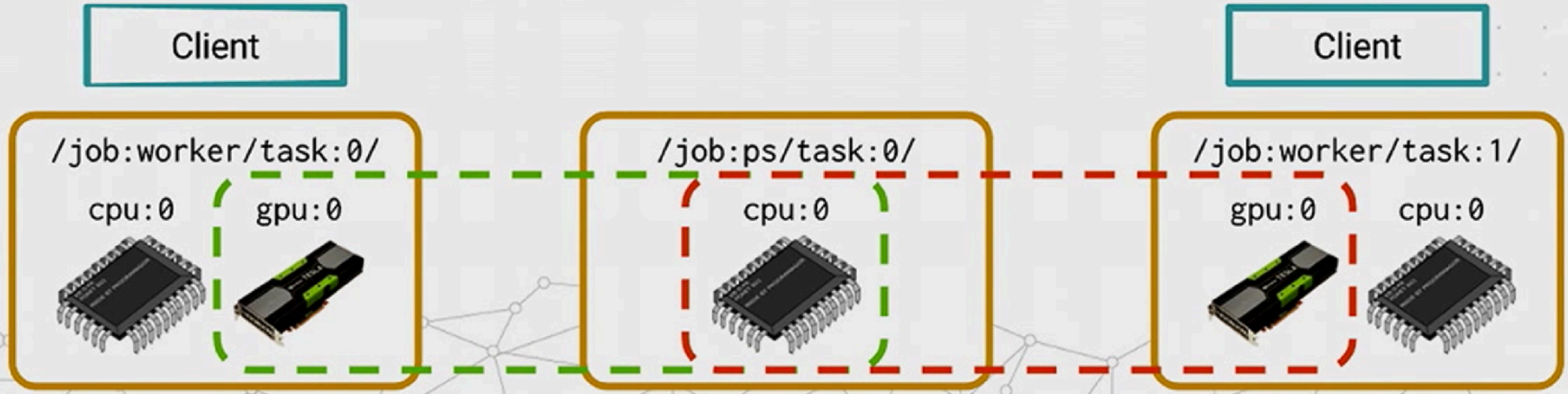
cpu:0



Between-graph replication

```
with tf.device("/job:ps/task:0/cpu:0"):
  W = tf.Variable(...)
  b = tf.Variable(...)
with tf.device("/job:worker/task:0/gpu:0"):
  output = tf.matmul(input, W) + b
  loss = f(output)
```

```
with tf.device("/job:ps/task:0/cpu:0"):
  W = tf.Variable(...)
  b = tf.Variable(...)
with tf.device("/job:worker/task:1/gpu:0"):
  output = tf.matmul(input, W) + b
  loss = f(output)
```



👋手动管理的资源

- IP or DNS
- port
- task ID
- CPU/GPU ID

😫抓狂的节奏

登录 10 台服务器

mount 10 遍共享存储

小心翼翼安排 GPU 资源

仔细设计 10 条启动命令

敲 10 遍这不一样的启动命令

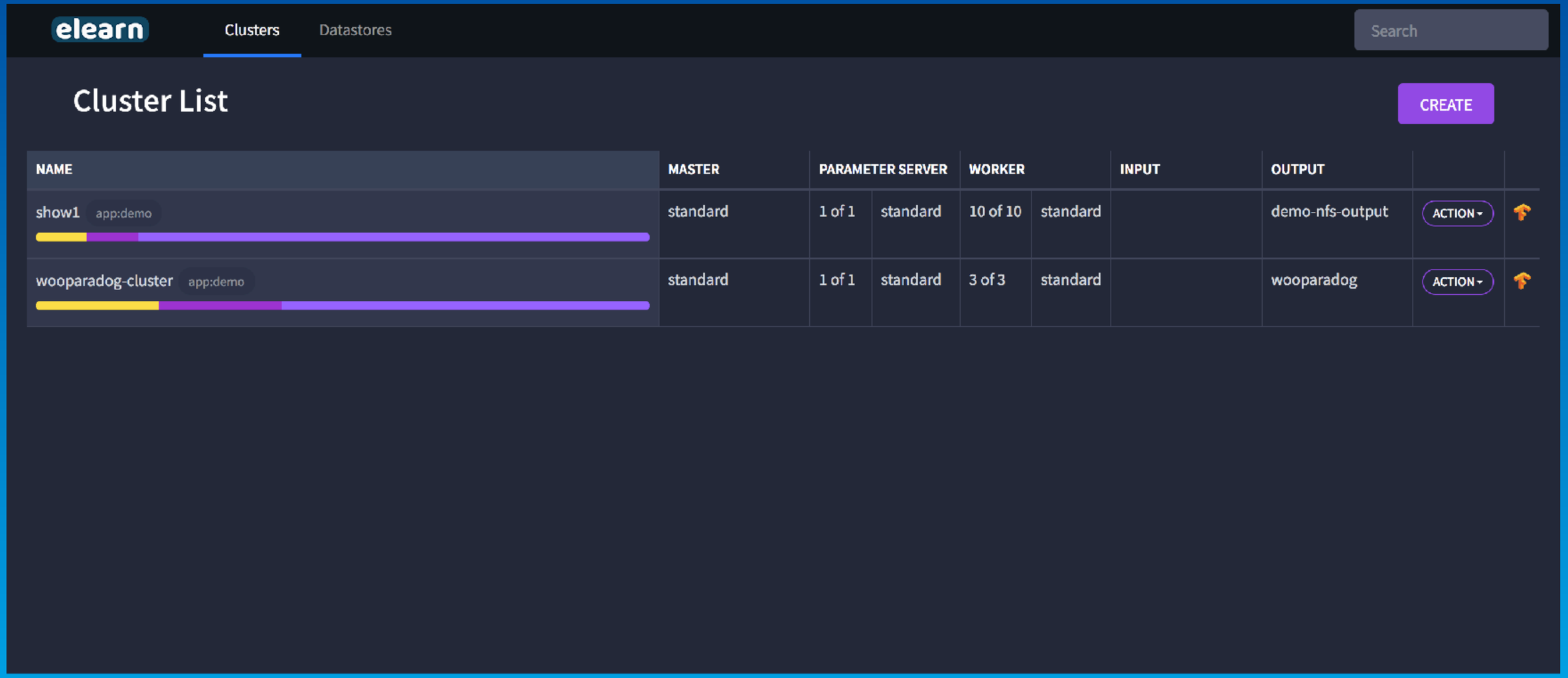
手动启动一个 TensorBoard

手动创建目录

mkdir + cp 手动管理模型版本

……

天呐！这才训练一次
以后可怎么每周训练、天天训练啊？！



New Cluster

NAME 1 这个训练的名字

my-demo

IMAGE 2 你的 Docker image

docker.elenet.me/jun.jiang02/tf-demo:20170602_182807-gpu

INPUT 3 input 填 Datastore 的名字。
可以不填，因为有些场景会需要实时获取数据。

Input will be mounted at `/elearn/input`

OUTPUT 4 output 填 Datastore 的名字。
必填。

demo-output

Output will be mounted at `/elearn/output`

COMMAND 5 启动命令完全自定义

python -m trainer.task --train_data_paths /demo/data/train.tfr.gz --eval_data_paths /demo/data/eval.tfr.gz --output_path /elearn/out

When running the image, this command will be appended after "bash -c"

ENV 6 环境变量

PYTHONPATH=/demo

Multiple envs are divided by "|" e.g. PYTHONPATH=/demo | DATA_DIR=/elearn/output

MASTER

Middle + 1 GPU

PARAMETER SERVER

Middle + 1 GPU

1

WORKER

Middle + 1 GPU

1

7 如果把 parameter server & worker 的数量
都设为 0，就是“单机版”。

CANCEL

CREATE CLUSTER

TensorFlow (GPU/CPU) cluster benchmar


```
INFO:root:Train [master/0], step 718 (73.296 sec) 11.5 global steps/s, 6.2 local steps/s
INFO:root:Train [master/0], step 731 (74.424 sec) 11.5 global steps/s, 6.2 local steps/s
INFO:root:Train [master/0], step 743 (75.439 sec) 11.8 global steps/s, 6.9 local steps/s
INFO:root:Train [master/0], step 754 (76.442 sec) 11.0 global steps/s, 6.0 local steps/s
INFO:root:Train [master/0], step 767 (77.495 sec) 12.3 global steps/s, 6.6 local steps/s
INFO:root:Train [master/0], step 781 (78.643 sec) 12.2 global steps/s, 7.0 local steps/s
INFO:root:Train [master/0], step 794 (79.765 sec) 11.6 global steps/s, 6.2 local steps/s
INFO:root:Train [master/0], step 806 (80.821 sec) 11.4 global steps/s, 6.6 local steps/s
INFO:root:Train [master/0], step 820 (81.973 sec) 12.2 global steps/s, 6.9 local steps/s
INFO:root:Train [master/0], step 824 (82.267 sec) 13.6 global steps/s, 6.8 local steps/s
```

Total: 3 GPUs

11
global steps/s

master	worker
2 GPUs x 1	1 GPU x 1

```
INFO:root:Train [master/0], step 10243 (385.718 sec) 27.0 global steps/s, 6.4 local steps/s
INFO:root:Train [master/0], step 10275 (386.849 sec) 28.3 global steps/s, 6.2 local steps/s
INFO:root:Train [master/0], step 10309 (388.070 sec) 27.9 global steps/s, 6.6 local steps/s
INFO:root:Train [master/0], step 10342 (389.215 sec) 28.8 global steps/s, 7.0 local steps/s
INFO:root:Train [master/0], step 10373 (390.361 sec) 27.1 global steps/s, 6.1 local steps/s
INFO:root:Train [master/0], step 10403 (391.413 sec) 28.5 global steps/s, 6.7 local steps/s
INFO:root:Train [master/0], step 10431 (392.424 sec) 27.7 global steps/s, 6.9 local steps/s
INFO:root:Train [master/0], step 10464 (393.550 sec) 29.3 global steps/s, 6.2 local steps/s
INFO:root:Train [master/0], step 10496 (394.768 sec) 26.3 global steps/s, 6.6 local steps/s
```

Total: 6 GPUs

28
global steps/s

master	worker
2 GPUs x 1	1 GPU x 4

```
INFO:root:Train [master/0], step 2657 (84.972 sec) 43.1 global steps/s, 5.7 local steps/s
INFO:root:Train [master/0], step 2704 (85.999 sec) 45.8 global steps/s, 6.8 local steps/s
INFO:root:Train [master/0], step 2749 (87.026 sec) 43.8 global steps/s, 5.8 local steps/s
INFO:root:Train [master/0], step 2797 (88.074 sec) 45.8 global steps/s, 6.7 local steps/s
INFO:root:Train [master/0], step 2845 (89.198 sec) 42.7 global steps/s, 6.2 local steps/s
INFO:root:Train [master/0], step 2893 (90.199 sec) 47.9 global steps/s, 6.0 local steps/s
INFO:root:Train [master/0], step 2940 (91.223 sec) 45.9 global steps/s, 6.8 local steps/s
INFO:root:Train [master/0], step 2984 (92.236 sec) 43.4 global steps/s, 5.9 local steps/s
INFO:root:Train [master/0], step 3030 (93.277 sec) 44.2 global steps/s, 6.7 local steps/s
INFO:root:Train [master/0], step 3060 (93.968 sec) 43.4 global steps/s, 5.8 local steps/s
```

Total: 9 GPUs

45
global steps/s

master	worker
2 GPUs x 1	1 GPU x 7

```
INFO:tensorflow:global_step/sec: 5.3238
INFO:tensorflow:global_step/sec: 5.3238
```

Total: 9 CPUs

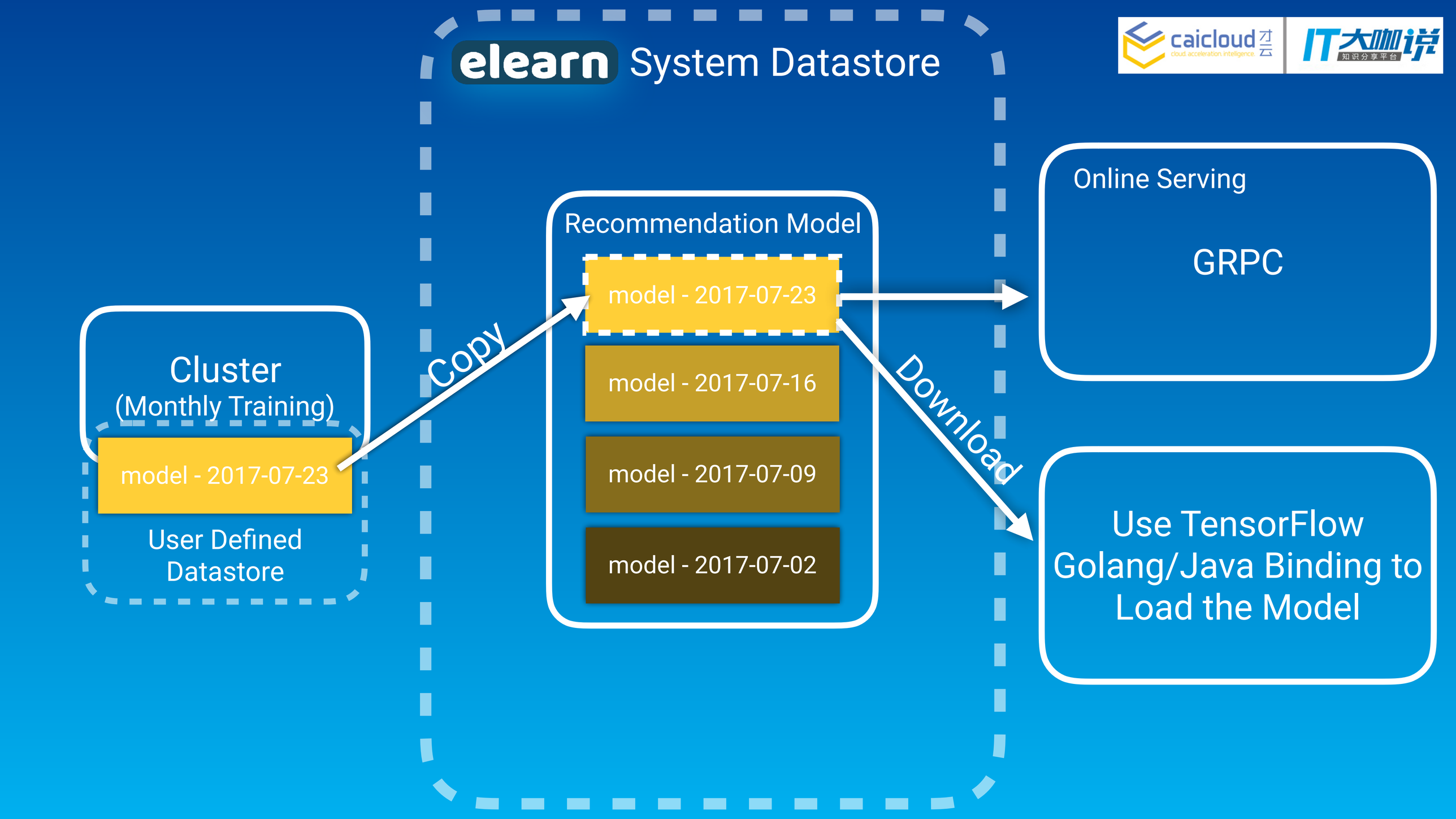
5
global steps/s

master	worker
2 CPUs x 1	1 CPU x 7

Model
Management

+

Serving

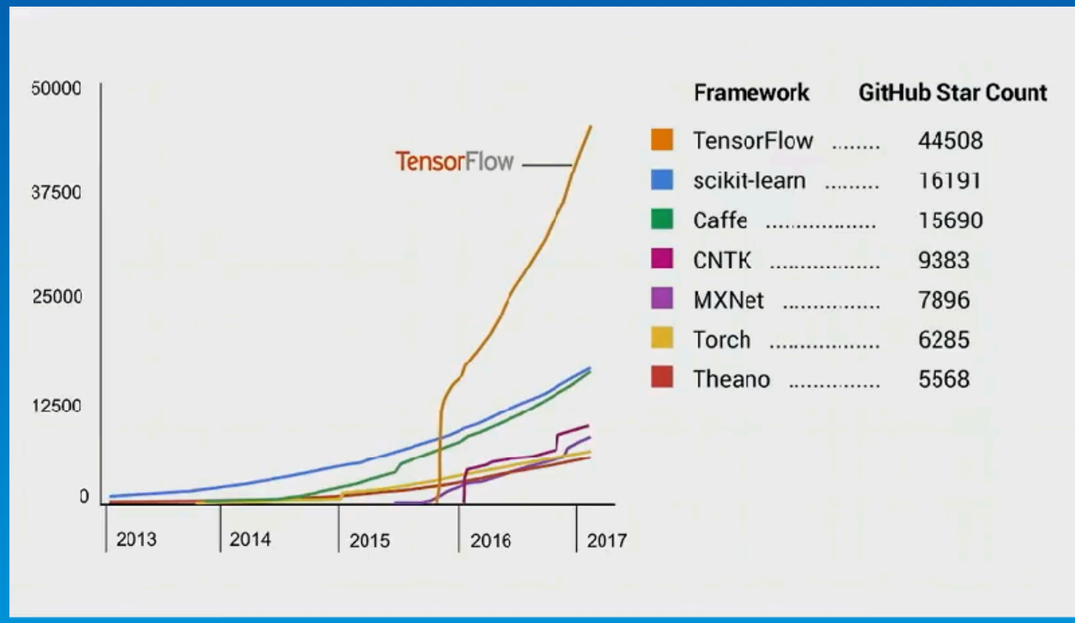

elearn System Datastore
caicloud 才云 cloud. acceleration. intelligence.
IT大咖说 知识分享平台
Online Serving
GRPC
Recommendation Model
model - 2017-07-23
model - 2017-07-16
model - 2017-07-09
model - 2017-07-02
Cluster
(Monthly Training)
model - 2017-07-23
User Defined
Datastore
Copy
Download
Use TensorFlow
Golang/Java Binding to
Load the Model

Demo

Thinking of elearn

为 TensorFlow 量身打造

如果让 elearn 支持其它 deep learning 框架，太容易了，只要写 driver 就可以了，但是我不会去这样做的。



尽量发挥 Kubernetes 的特色功能和编排能力

Deployment, Job, StatefulSet, ConfigMap, Nginx Ingress, ...

如果让 elearn 支持其它 Cloud，太容易了，只要写 interface 够通用，

然后写 driver 就可以了，但会失去 Kubernetes 本身的意义，

所以 elearn cloud interfaces 的设计是面向功能的 。

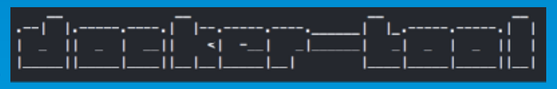
让 Kubernetes 变成 OS

所有小的操作都以 Job 形式下发

比如训练完成后的 post run、保存 modelVersion，这些任务都是 Kubernetes Job，并不在 elearn server 上运行。

I'm 江骏 / ohmystack
@饿了么

推荐项目（点击链接前往）

dt (docker-tool) 

最爽功能：dt ssh

gotool

管理 Golang 开发环境 GOPATH 的利器