Kubebench: Benchmarking ML Workloads on Kubernetes

Xinyuan Huang (Cisco)  Ce Gao (Caicloud)
Why Kubebench?

• Understanding system performance is essential for moving ML from lab to production.

• Benchmarking and analyzing ML workloads on Kubernetes is not an easy job today.

• Many requirements for a good benchmark: compliance, consistency, reproducibility, …
What is Kubebench?

Kubebench is a harness for benchmarking and analyzing Machine Learning workloads on Kubernetes.
Goals of Kubebench

• Support benchmarking in various circumstances
  • Multi-cloud and various infrastructure
  • Different ML frameworks
  • Distributed workloads
  • …

• Make it easier to manage benchmarks
  • Consistent workloads
  • Reproducible results
  • Integrable with the rest of ML lifecycle
  • …
Tech Stack

**Kubebench**
Benchmark config/result management; Benchmark workflow deployment

**Kubeflow**
ML job deployment / lifecycle management

**kubernetes**
Production grade container orchestration

**Infrastructure**
Cloud/On-premise infrastructure environment
Architecture

**Workflow (Argo)**
- Configurator
- Job Deployer/Manager
- Reporter

**Workload**
- Pre-process Job
- Kubeflow Job (TFJob/PyTorchJob/etc.)
- Post-process Job

**Monitoring**
- Metrics Visualization (Grafana)
- Monitoring Service (Prometheus)

**Interface**
- API (CRD)

**Dashboard**
- Manage
- Read
- Deploy
- Monitor
- Read/Write

**Storage**
- Configs
- Data
- Experiment Records

**User-defined**
- Kubebench-provided
User’s Perspective

Job Developer

- Pre-job
- Main job
- Post job

Shared storage (auto mounted)

Kubebench workflow

Experiment Runner

- Job template (Ksonnet)
- Job params (.yaml)

Kubebench workflow

Benchmark results
Where we are

Current release (V0.3):

• Support local/distributed training workloads
• Support multiple frameworks
  • TFJob
  • PyTorchJob
  • (more planned)
• Support result aggregation for multi-experiments
  • Stored in filesystem
  • (Remote/Cloud DB planned)
• Quick starter package
  • Parameter-less e2e example for quick start
  • Example workloads (TF-CNN)

Upcoming and Future releases:

• UI/UX
  • Dashboard
  • Results/metrics visualizations
• API
  • Kubebench CRD
• More benchmarking scenarios
  • Serving/inference benchmarks
  • Mixed/scaled workloads
• …
Case Study

Ce Gao
Local Training Benchmark

TensorFlow CNN Benchmark
Dataset: imagenet (synthetic)
Mode: forward-only
SingleSess: False
Num batches: 100
Num epochs: 0.00
Data format: NCHW
Optimizer: sgd
Variables: parameter_server

Training performance among different GPU numbers, batch sizes, and platforms
Distributed Training Benchmark

Dataset: imagenet (synthetic)
Mode: forward-only
SingleSess: False
Num batches: 100
Num epochs: 0.00
Data format: NCHW
Optimizer: sgd
Variables: parameter_server

1 PS
2 workers (2 GPU per worker)
<table>
<thead>
<tr>
<th><strong>Dependency</strong></th>
<th><strong>Version</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cuda</td>
<td>9.0</td>
</tr>
<tr>
<td>CuDNN</td>
<td>7.1</td>
</tr>
<tr>
<td>GPU</td>
<td>GTX 1080ti</td>
</tr>
<tr>
<td>TensorFlow</td>
<td>1.10</td>
</tr>
<tr>
<td>Kernel Version</td>
<td>3.10.0-862.11.6.el7.x86_64</td>
</tr>
<tr>
<td>OS Image</td>
<td>CentOS Linux 7 (Core)</td>
</tr>
<tr>
<td>Operating System</td>
<td>linux</td>
</tr>
<tr>
<td>Architecture</td>
<td>amd64</td>
</tr>
<tr>
<td>Container Runtime Version</td>
<td>Docker 18.03.0-ce</td>
</tr>
<tr>
<td>Kubernetes Version</td>
<td>v1.10.1</td>
</tr>
</tbody>
</table>
Thanks!

Contributors & Advisors
(alphabetical order)

Adhita Selvaraj
Amit Kumar Saha
Andrey Velichkevich
Ce Gao
David Aronchick
Debo Dutta
Jeremy Lewi
Johnu George
Kirill Prosvirov
Ramdoot Kumar Pydipaty
Xinyuan Huang

...