BenchCouncil AIBench
--- A Datacenter AI Benchmark Suite

Wanling Gao, Fei Tang, Jianfeng Zhan

http://www.benchcouncil.org/AIBench/index.html

BenchCouncil
Bench’19, Denver, Colorado, USA
Why Datacenter AI?

- **AI** is widely employed to augment **Internet services**
  - processing images, video, speech, and audio

- There is an urgent need for datacenter AI benchmarks
Challenge 1# Isolation

- Confidential issues of workloads and datasets
  - Isolation!

- There is **no publicly available** industry-scale Internet service benchmark
Challenge 2# Microservice based Architecture

- A collection of loosely coupled services
  - Various modules and complex execution path
  - Massive scale and complex hierarchy of infrastructure

End-to-end benchmark that models the critical paths and primary modules is needed
Challenge 3# Diversity of workloads and models

Challenge 4# Domain-specific metrics

- Time-to-accuracy
  - State-of-the-art accuracy

- Throughput

- Latency

- Tail latency
Challenge 5# More vs. Less

- Workload characterization
  - SPECCPU 2017 (43), PARSEC3.0 (30), TPC-DS (99)

- Performance ranking (Benchmarketing)
  - top500
Challenge 6# Inconsistency

- Inconsistent benchmarking requirements

Micro or Component?  
Portability  
Reality  
Application benchmark?

Earlier stage of architecture research  
Later stage of architecture research
Requirements

- **Industry-scale**
  - critical paths and primary modules of business AI scenario
    - Al-related and non Al-related components

- **A modular framework design**
  - Collectively as a whole end-to-end application
  - Individually as a micro or component benchmark

- **Representativeness and coverage**
  - Diverse AI problem domains and datasets are needed
Outline

- AlBench Overview
  - Tasks, Models, Datasets, Metrics
- How to use AlBench
- Preliminary Results
- Conclusion
BenchCouncil AlBench

- A Datacenter AI Benchmark Suite
  - Contributors: many companies and top universities
    - Alibaba, Microsoft, Paypal, Tencent, etc

http://www.benchcouncil.org/AlBench/index.html

**AI Bench Overview**

- **The First end-to-end** industry-standard AI benchmark suite
  - Industry-scale Internet services
    - critical paths and primary modules
    - AI-related and non AI-related

- A highly extensible, configurable, and flexible benchmark framework
  - 16 prominent AI problem domains
  - Multiple loosely coupled modules
    - Individually
      - Micro/Component benchmarks
    - Collectively
      - Application benchmarks
Sixteen AI Problem Domains

- **Text Processing (4)**
  - Text-to-Text translation, Text summarization, Learning to rank, Recommendation

- **Image Processing (8)**
  - Image classification, Image generation, Image-to-text, Image-to-Image, Face embedding, Object detection, Image compression, Spatial transformer

- **Audio Processing (1)**
  - Speech recognition

- **Video Processing (1)**
  - Video prediction

- **3D Data Processing (2)**
  - 3D face recognition, 3D object reconstruction
End-to-End: E-commerce Search

- **Query generator**: simulate concurrent users and send query requests
- **Online Module**: personalized searching and recommendations
- **Offline Module**: a training stage to generate a learning model
- **Data storage module**: data storage, e.g., user database, product database
## Component Benchmark (16)

<table>
<thead>
<tr>
<th>No.</th>
<th>Component Benchmark</th>
<th>Algorithm</th>
<th>Data Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC-AI-C1</td>
<td>Image classification</td>
<td>ResNet50</td>
<td>ImageNet</td>
</tr>
<tr>
<td>DC-AI-C2</td>
<td>Image generation</td>
<td>WassersteinGAN</td>
<td>LSUN</td>
</tr>
<tr>
<td>DC-AI-C3</td>
<td>Text-to-Text translation</td>
<td>Transformer</td>
<td>WMT English-German</td>
</tr>
<tr>
<td>DC-AI-C4</td>
<td>Image-to-Text</td>
<td>Neural Image Caption Model</td>
<td>Microsoft COCO</td>
</tr>
<tr>
<td>DC-AI-C5</td>
<td>Image-to-Image</td>
<td>CycleGAN</td>
<td>Cityscapes</td>
</tr>
<tr>
<td>DC-AI-C6</td>
<td>Speech recognition</td>
<td>DeepSpeech2</td>
<td>Librispeech</td>
</tr>
<tr>
<td>DC-AI-C7</td>
<td>Face embedding</td>
<td>Facenet</td>
<td>LFW, VGGFace2</td>
</tr>
<tr>
<td>DC-AI-C8</td>
<td>3D Face Recognition</td>
<td>3D face models</td>
<td>77,715 samples from 253 face IDs</td>
</tr>
<tr>
<td>DC-AI-C9</td>
<td>Object detection</td>
<td>Faster R-CNN</td>
<td>Microsoft COCO</td>
</tr>
<tr>
<td>DC-AI-C10</td>
<td>Recommendation</td>
<td>Neural collaborative filtering</td>
<td>MovieLens</td>
</tr>
<tr>
<td>DC-AI-C11</td>
<td>Video prediction</td>
<td>Motion-Focused predictive models</td>
<td>Robot pushing data set</td>
</tr>
<tr>
<td>DC-AI-C12</td>
<td>Image compression</td>
<td>Recurrent neural network</td>
<td>ImageNet</td>
</tr>
<tr>
<td>DC-AI-C13</td>
<td>3D object reconstruction</td>
<td>Convolutional encoder-decoder network</td>
<td>ShapeNet Data set</td>
</tr>
<tr>
<td>DC-AI-C14</td>
<td>Text summarization</td>
<td>Sequence-to-sequence model</td>
<td>Gigaword data set</td>
</tr>
<tr>
<td>DC-AI-C15</td>
<td>Spatial transformer</td>
<td>Spatial transformer networks</td>
<td>MNIST</td>
</tr>
<tr>
<td>DC-AI-C16</td>
<td>Learning to rank</td>
<td>Ranking distillation</td>
<td>Gowalla</td>
</tr>
</tbody>
</table>
BenchCouncil International Competitions

- DC-AI-C1 Image classification
- DC-AI-C8 3D face recognition
- DC-AI-C10 Recommendation

Competition papers are available soon!
Image Classification

- Extract different thematic classes within an image
  - a supervised learning problem to define a set of target classes and train a model to recognize
- ResNet neural network, Dataset: ImageNet2012, 100GB+
Image Generation

- Mimic the distribution of data and generate image data
  - Dataset: LSUN, about million labelled image data
  - Model: WGAN algorithm
Text-to-Text Translation

- Translates text from one language to another
  - Model: Transformer
  - Dataset: WMT English-German (4.5MB training text data)
Image-to-Image

- Converts an image from one representation of a specific scene to another scene or representation
  - Model: cycle- GAN algorithm
  - Datasets: Cityscapes from 50+ cities (300MB)
Speech-to-Text

- Recognizes and translates the spoken language to text
  - Model: deep speech 2
  - Dataset: LibriSpeech, 1000+ hours’ speech data
Object Detection

- Detects the objects within an image
  - Model: Faster R-CNN algorithm
  - Dataset: MSCOCO2014
    - 82783 training samples, 40504 Validation samples, 40775 test samples (20GB+)
Image-to-Text

- Generates the description of an image automatically
  - Model: Neural Image Caption model
  - Dataset: MSCOCO2014

A woman is throwing a frisbee in a park.
A dog is standing on a hardwood floor.
A stop sign is on a road with a mountain in the background.
A little girl sitting on a bed with a teddy bear.
A group of people sitting on a boat in the water.
A giraffe standing in a forest with trees in the background.
Face Embedding

- Transforms a facial image to a vector in embedding space
  - Model: FaceNet algorithm
  - Dataset: VGGFace2
    - 36GB training data, 1.9GB test data
3D Face Recognition

- Recognize the 3D facial information from an image
  - Model: 3D face models
  - Dataset: Intellifusion data set, 77,715 samples from 253 face IDs
Video Prediction

- Predicts the future video through predicting previous frames transformation
  - Model: motion-focused predictive models
  - Dataset: Robot pushing dataset
    - 59000 samples, 100GB+
Image Compression

- Full-resolution lossy image compression
  - Model: recurrent neural networks
  - Dataset: ImageNet2012, 100GB+
Recommendation

- Collaborative filtering-based movie recommendations
  - Model: Collaborative filtering algorithm
  - Dataset: MovieLens
    - 20,000,000 movie ratings data

More Like: The Heart of Christmas

![Movie posters](image-url)
3D Object Reconstruction

- Predicts and reconstructs 3D objects
  - Model: a convolutional encoder-decoder network
  - Dataset: ShapeNet
    - 51,300 different 3D data covering 5 object categories
Text Summarization

- Generate the text summary
  - Model: sequence-to-sequence model
  - Dataset: Gigaword
    - 10,000,000 text data, Four billion words
Spatial Transformer

- Performs spatial transformations
  - Model: spatial transformer networks
  - Dataset: MNIST
    - 60000 training samples, 10000 test samples
Learning to Rank

- Machine-learned ranking for recommender system
  - Model: ranking distillation
  - Dataset: Gowalla
    - Social network data: 196,591 nodes and 950,327 edges
## Micro Benchmark (12)

<table>
<thead>
<tr>
<th>No.</th>
<th>Micro Benchmark</th>
<th>Data Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC-AI-M1</td>
<td>Convolution</td>
<td>ImageNet, Cifar</td>
</tr>
<tr>
<td>DC-AI-M2</td>
<td>Fully Connected</td>
<td>ImageNet, Cifar</td>
</tr>
<tr>
<td>DC-AI-M3</td>
<td>Relu</td>
<td>ImageNet, Cifar</td>
</tr>
<tr>
<td>DC-AI-M4</td>
<td>Sigmoid</td>
<td>ImageNet, Cifar</td>
</tr>
<tr>
<td>DC-AI-M5</td>
<td>Tanh</td>
<td>ImageNet, Cifar</td>
</tr>
<tr>
<td>DC-AI-M6</td>
<td>MaxPooling</td>
<td>ImageNet, Cifar</td>
</tr>
<tr>
<td>DC-AI-M7</td>
<td>AvgPooling</td>
<td>ImageNet, Cifar</td>
</tr>
<tr>
<td>DC-AI-M8</td>
<td>CosineNorm</td>
<td>ImageNet, Cifar</td>
</tr>
<tr>
<td>DC-AI-M9</td>
<td>BatchNorm</td>
<td>ImageNet, Cifar</td>
</tr>
<tr>
<td>DC-AI-M10</td>
<td>Dropout</td>
<td>ImageNet, Cifar</td>
</tr>
<tr>
<td>DC-AI-M11</td>
<td>Element-wise multiply</td>
<td>ImageNet, Cifar</td>
</tr>
<tr>
<td>DC-AI-M12</td>
<td>Softmax</td>
<td>ImageNet, Cifar</td>
</tr>
</tbody>
</table>
AI Bench Inference Specification

- Inference System under Test

  - Query Generator
    - Concurrency
    - Arriving rate
    - Distribution
    - Thinking time

  - Datasets

  - System under Test

  - Monitoring Tools

  - Result Outputs
    - Accuracy
    - Latency
    - Tail Latency
    - Throughput

Datasets

Result Outputs
Inference Metrics

- Online Inference (*Accuracy-ensured*)
  - Latency, Tail latency
  - Latency-bounded throughput

- Offline Inference (*Accuracy-ensured*)
  - Throughput, Energy consumption

- Accuracy-ensured:
  - Accuracy deviation with target accuracy is within 2%
AIBench Training Specification

Training System under Test

- Datasets
- System under Test
- Monitoring Tools
- Result Outputs
  - Accuracy
  - Latency
  - Tail Latency
  - Throughput

Datasets
System under Test
Monitoring Tools
Result Outputs
Training Metrics

- Offline Training
  - Time-to-accuracy
  - Energy-to-accuracy
  - Throughput
    - Running 1000 epochs
    - Hyper parameter settings should be able to achieve target accuracy
Benchmark Guideline

- Online Server
  - Each component can be distributed deployed on a large cluster

<table>
<thead>
<tr>
<th>Module</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Generator</td>
<td>Jmeter 5.1.1</td>
</tr>
<tr>
<td>Search Planer</td>
<td>SpringBoot 2.1.3</td>
</tr>
<tr>
<td>Recommender</td>
<td>Category Predictor, Flask Web 1.1.1, Nginx 1.10.3</td>
</tr>
<tr>
<td></td>
<td>TensorFlow Serving, TensorFlow Serving 1.14.0</td>
</tr>
<tr>
<td>Searcher</td>
<td>Cluster (high popularity)</td>
</tr>
<tr>
<td></td>
<td>Cluster (medium popularity)</td>
</tr>
<tr>
<td></td>
<td>Cluster (low popularity)</td>
</tr>
<tr>
<td>Ranker</td>
<td>Elasticsearch 6.5.2</td>
</tr>
<tr>
<td>Data Storage</td>
<td>User database, Neo4j 3.5.8</td>
</tr>
<tr>
<td></td>
<td>Product database, Elasticsearch 6.5.2</td>
</tr>
</tbody>
</table>
Benchmark Guideline

- Offline Analytics
  - Single GPU, Multi GPUs, Distributed versions
    - TensorFlow implementation
    - PyTorch implementation
  - Example distributed training setting

```python
parser.add_argument('--world-size', default=-1, type=int,
                    help='number of nodes for distributed training')
parser.add_argument('--dist-url', default='tcp://224.66.41.62:23456', type=str,
                    help='url used to set up distributed training')
parser.add_argument('--dist-backend', default='nccl', type=str,
                    help='distributed backend')
parser.add_argument('--seed', default=None, type=int,
                    help='seed for initializing training.')
parser.add_argument('--gpu', default=None, type=int,
                    help='GPU id to use.')
parser.add_argument('--multiprocessing-distributed', action='store_true',
                    help='Use multi-processing distributed training to launch ' +
                         'N processes per node, which has N GPUs. This is the ' +
                         'fastest way to use PyTorch for either single node or ' +
                         'multi node data parallel training')
```
Outline

- AlBench Overview
  - Tasks, Models, Datasets, Metrics
- How to use AlBench
- Preliminary Results
- Conclusion
General Steps to Use AIBench

General steps to run the benchmarks
- Prepare the package of AIBench
- Prepare the environments of the selected software stack
- Prepare corresponding data set
- Run the scripts or commands (User Manual!)
  - Micro benchmarks
    - run-tensorflow.sh (TensorFlow), run-pthread.sh (Pthreads)
  - Component benchmarks
    - run_train_time.sh (Training stage), run_val_time.sh (Inference stage)
  - Application benchmarks
    - Start online and offline modules
      » neo4j, Elasticsearch, Recommender, Search-planer
How to Download AIBench

- [http://www.benchcouncil.org/AIBench/download.html](http://www.benchcouncil.org/AIBench/download.html)

- User Manual

Download AIBench from BenchHub:

- AIBench Framework:
- AIBench Application Benchmark (E-commerce Search):
- AIBench Application Benchmark (DCMix):
- AIBench Component Benchmark:
- AIBench Micro Benchmark:
Directory Structure

- **Micro Benchmark**
  - Pthreads: 12 benchmarks
  - TensorFlow: 12 benchmarks

- **Component Benchmark**
  - TensorFlow: 16 benchmarks
  - PyTorch: 16 benchmarks

- **Application Benchmark**
  - Offline Module: 10 benchmarks
  - Online Module: Online benchmarks
AlBench Framework


Name | Last commit | Last update  
--- | --- | ---  
DataInputModule | Initial commit | 3 weeks ago  
DeploymentToolModule | Add DeploymentTool Readme | 3 weeks ago  
OfflineModule | Initial commit | 3 weeks ago  
OnlineModule | Initial commit | 3 weeks ago  
.DS_Store | Initial commit | 3 weeks ago  
README.txt | Initial commit | 3 weeks ago  

Add license | 3 Commits | 1 Branch | 0 Tags | 1.5 GB Files

Add DeploymentTool Readme
aibench authored 3 weeks ago

Add README | Add CHANGELOG | Add CONTRIBUTING | Enable Auto DevOps

Add Kubernetes cluster | Set up CI/CD
Application Benchmarks

Component Benchmarks


DC_AIBench_Component
Project ID: 14

Add license  9 Commits  1 Branch  0 Tags  580.1 MB Files

DC AI Bench: Towards scalable and comprehensive datacenter AI benchmarking

Add DC_AIBench_Component
aibench authored 3 weeks ago
c968a842

README  Add CHANGELOG  Add CONTRIBUTING  Enable Auto DevOps
Add Kubernetes cluster  Set up CI/CD

<table>
<thead>
<tr>
<th>Name</th>
<th>Last commit</th>
<th>Last update</th>
</tr>
</thead>
<tbody>
<tr>
<td>PyTorch</td>
<td>Add DC_AIBench_Component</td>
<td>3 weeks ago</td>
</tr>
<tr>
<td>TensorFlow</td>
<td>Add DC_AIBench_Component</td>
<td>3 weeks ago</td>
</tr>
<tr>
<td>README.md</td>
<td>Update README.md</td>
<td>2 months ago</td>
</tr>
</tbody>
</table>
Micro Benchmarks


DC_AIBench_Micro
Project ID: 13

Add license  7 Commits  1 Branch  0 Tags  129.8 MB Files

DC AI Bench: Towards scalable and comprehensive datacenter AI benchmarking

master  DC_AIBench_Micro  History  Find file  Web IDE

Update README.md
Al Bench authored 3 weeks ago

- README
- Add CHANGELOG
- Add CONTRIBUTING
- Enable Auto DevOps
- Add Kubernetes cluster
- Set up CI/CD

<table>
<thead>
<tr>
<th>Name</th>
<th>Last commit</th>
<th>Last update</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pthread</td>
<td>Add DC_AIBench_Micro</td>
<td>4 months ago</td>
</tr>
<tr>
<td>TensorFlow</td>
<td>Add DC_AIBench_Micro</td>
<td>4 months ago</td>
</tr>
<tr>
<td>README.md</td>
<td>Update README.md</td>
<td>3 weeks ago</td>
</tr>
</tbody>
</table>
BenchCouncil Testbed

- [http://www.benchcouncil.org/testbed.html](http://www.benchcouncil.org/testbed.html)

- Provide container-based AI Bench images
  - Log in and apply for nodes!

- Provide pretrained AI models
Image-to-Text (TensorFlow)

- Generates the description of an image automatically
  - Model: Neural Image Caption model
  - Dataset: MSCOCO2014

Steps:

- Apply for nodes on testbed
  ```bash
cd DC_AIBench_Component/TensorFlow/Image_to_Text/tf-models/research/im2txt
```

- Training or inference
  ```bash
  ./run_train_time.sh
  ./run_val_time.sh
  ```
Image Classification (PyTorch)

- Extract different thematic classes within an image
  - ResNet neural network, Dataset: ImageNet2012, 100GB+

- Steps:
  - Apply for nodes on testbed
  ```
  cd DC_AIBench_Component/PyTorch/Image_classification
  ```
  - Training or inference
  ```
  ./run_train_time.sh
  ./run_val_time.sh
  ```
Text-to-Text (PyTorch)

- Translates text from one language to another
  - Model: Transformer
  - Dataset: WMT English-German

Steps:
- Apply for nodes on testbed
  ```bash
cd DC_AIBench_Component/PyTorch/Text_to_Text
```
- Training or inference
  ```bash
./run_train_time.sh
./run_val_time.sh
```
Outline

- AlBench Overview
  - Tasks, Models, Datasets, Metrics
- How to use AlBench
- Preliminary Results
- Conclusion
Latency of Online Server

- AI components change the critical path significantly
  - 34.29 vs. 49.07 milliseconds for average latency

- Model depth and size limit QoS
  - 99th percentile latency increasing from 149.12 to 5335.12 milliseconds when model increasing from 184 MB to 253 MB

- AI-related components suffer from higher cache misses
  - 61 vs. 37 for L2 cache misses per Kilo instructions
SM Efficiency

- Different models have different execution efficiency
- Learning_to_rank has the lowest efficiency
Runtime Breakdown

- Six categories
  - using nvprof to trace the running time breakdown and find the hotspot functions that occupy more than 80% of running time in total
## Hotspot Functions

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Function Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolution</td>
<td>maxwell.scudnn.128x128.stridedB.splitK.interior.nn</td>
</tr>
<tr>
<td></td>
<td>maxwell.scudnn.128x32.stridedB.splitK.interior.nn</td>
</tr>
<tr>
<td></td>
<td>maxwell.scudnn.winograd.128x128.ldg1.ldg4.tile148n.nt</td>
</tr>
<tr>
<td>GEMM</td>
<td>maxwell.sgemm.128x64.nt</td>
</tr>
<tr>
<td></td>
<td>maxwell.sgemm.128x64.nn</td>
</tr>
<tr>
<td></td>
<td>sgemm.32x32x32.NN.vec</td>
</tr>
<tr>
<td>BatchNorm</td>
<td>cudnn::detail::bn_fw_tr.IC11.kernel.NCHW</td>
</tr>
<tr>
<td></td>
<td>cudnn::detail::bn_bw.IC11.kernel.new</td>
</tr>
<tr>
<td></td>
<td>batch.norm.backward.kernel</td>
</tr>
<tr>
<td></td>
<td>at::native::batch.norm.backward.kernel</td>
</tr>
<tr>
<td>Relu</td>
<td>maxwell.scudnn.128x128.relu.small.nn</td>
</tr>
<tr>
<td></td>
<td>maxwell.scudnn.128x128.relu.small.nn</td>
</tr>
<tr>
<td></td>
<td>maxwell.scudnn.128x32.relu.interior.nn</td>
</tr>
<tr>
<td>Element-wise</td>
<td>element-wise add kernel</td>
</tr>
<tr>
<td></td>
<td>element-wise threshold kernel</td>
</tr>
<tr>
<td>Gradient</td>
<td>cudnn::detail::dgrad_engine</td>
</tr>
<tr>
<td></td>
<td>cudnn::detail::dgrad_alg1_engine</td>
</tr>
</tbody>
</table>
Stall Breakdown

- Top two stalls
  - memory dependency stalls, execution dependency stalls

![Chart showing stall breakdown](chart.png)
Outline

- AlBench Overview
  - Tasks, Models, Datasets, Metrics

- How to use AlBench

- Preliminary Results

- Conclusion
Conclusion

- **AI Bench Website:**
  http://www.benchcouncil.org/AIBench/index.html

- Please refer to user manual for more details!
Publications

- **Benchmarking**
  - Data Motifs: A Lens Towards Fully Understanding Big Data and AI Workloads. PACT’18.
  - BigDataBench: a Big Data Benchmark Suite from Internet Services. HPCA’14
  - Data Motif-based Proxy Benchmarks for Big Data and AI Workloads. IISWC 2018.
  - [Auto-tuning Spark Big Data Workloads on POWER8: Prediction-Based Dynamic SMT](https://example.com). PACT’16
  - CVR: Efficient Vectorization of SpMV on X86 Processors. CGO’18.
  - Characterizing data analysis workloads in data centers. IISWC 13 best paper award.
QUESTIONS
And
Answers