BigDataBench Tutorial

Jianfeng Zhan, Zhen Jia, and Gang Lu

http://prof.ict.ac.cn/BigDataBench

Institute of Computing Technology, Chinese Academy of Sciences and University of Chinese Academy of Sciences

BigDataBench Tutorial
MICRO 2014 Cambridge, UK
Acknowledgements

BigDataBench contributors

Lei Wang, Chunjie Luo, Zhen Jia, Wanling Gao, Dr. Rui Han, Dr. Yuqing Zhu, Qiang Yang, Xinlong Lin, Jingwei Li, Wei Zhu

Shujie Zhang, Dr. Chuliang Weng

Dr. Yongqiang He

Xiaona Li

Bizhu Qiu

Kent Zhan, Zijian Ming
Acknowledgements (Cont’)

- Great thanks for Prof. Jason Mars to invite us to give this tutorial at Micro’14.
BigDataBench Tutorial Program (1)

- 9:00-9:40 Jianfeng Zhan
  - What is BigDataBench?
  - BigDataBench benchmarking methodology
- 9:40-10:00 Gang Lu
  - BigDataBench data sets and workloads
- 10:00-10:30 Coffee break
BigDataBench Tutorial Program (2)

- 10:30-11:00  Gang Lu
  - How to use BigDataBench data sets and workloads?
  - How to generate Large-scale data sets?
  - Multi-tenancy version of BigDataBench

- 11:00-12:00  Zhen Jia
  - BigDataBench subsetting
  - How to use the simulator versions of BigDataBench?
Handbook of BigDataBench

- Please feel free to download and distribute this handbook.
  - http://prof.ict.ac.cn/BigDataBench_micro_14/
  - Draft version
  - 93 pages
BigDataBench handbook: $1^{st}$ part

- Section 1 Summary of BigDataBench 3.1
- Section 2 Benchmarking methodology
- Section 3 BigDataBench specification
- Section 4 BigDataBench implementations
- Section 5 BigDataBench subsetting
BigDataBench handbook: 2th part

- Section 6 BigDataBench simulator version
- Section 7 Multi-tenancy version of BigDataBench
- Section 8 User manual
- Section 9 BigDataBench users
- Section 10 Q&A
Outline

- What is BigDataBench?
- BigDataBench benchmarking methodology
Why Big Data Benchmarking?

Measuring big data systems and architectures quantitatively
What is *BigDataBench*?

- An open source big data benchmarking project
  - [http://prof.ict.ac.cn/BigDataBench](http://prof.ict.ac.cn/BigDataBench)
  - Search Google using “*BigDataBench*”
What is BigDataBench (cont’)?

BDGS (Big Data Generator Suite) for scalable data

<table>
<thead>
<tr>
<th>Data Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wikipedia Entries</td>
</tr>
<tr>
<td>Amazon Movie Reviews</td>
</tr>
<tr>
<td>Google Web Graph</td>
</tr>
<tr>
<td>Facebook Social Network</td>
</tr>
<tr>
<td>E-commerce Transaction</td>
</tr>
<tr>
<td>ProfSearch Resumes</td>
</tr>
<tr>
<td>ImageNet</td>
</tr>
<tr>
<td>English broadcasting audio</td>
</tr>
<tr>
<td>DVD Input Streams</td>
</tr>
<tr>
<td>Image scene</td>
</tr>
<tr>
<td>Genome sequence data</td>
</tr>
<tr>
<td>Assembly of the human genome</td>
</tr>
<tr>
<td>SoGou Data</td>
</tr>
<tr>
<td>MNIST</td>
</tr>
</tbody>
</table>

14 Real-world Data Sets

33 Workloads

Search Engine

Social Network

E-commerce

Multimedia

Bioinformatics

Impala

Shark

Hadoop RDMA

NoSql

MVAPICH

Oracle

MySQL

MPI

DataMPI

Software Stacks
BigDataBench evolution

- **BigDataBench 1.0**: 2013.7
  - Search engine
  - 6 workloads
  - BigDataBench 1.0

- **BigDataBench 2.0**: 2013.12
  - Typical Internet service domains
  - An architectural perspective
  - 19 workloads & data generation tools
  - BigDataBench 2.0

- **BigDataBench 3.0**: 2014.4
  - Multidisciplinary effort
  - 32 workloads: diverse implementations
  - BigDataBench 3.0

- **BigDataBench 3.1**: 2014.12
  - 5 application domains: 14 data sets and 33 workloads
  - Same specifications: diverse implementations
  - Multi-tenancy version
  - BigDataBench subset and simulator version

- **CloudRank 1.0**: 2014
  - Mixed data analytics workloads
  - CloudRank 1.0

- **DCBench 1.0**: 2014
  - 11 data analytics workloads
  - DCBench 1.0

**Typical Internet service domains**

**Mixed data analytics workloads**

**An architectural perspective**

**5 application domains**

**14 data sets** and **33 workloads**

**Same specifications**: diverse implementations

**Multi-tenancy version**

**BigDataBench subset and simulator version**
Why BigDataBench?

<table>
<thead>
<tr>
<th></th>
<th>Specification</th>
<th>Application domains</th>
<th>Workload Types</th>
<th>Workloads</th>
<th>Scalable data sets (from real data)</th>
<th>Multiple implementations</th>
<th>Multitenancy</th>
<th>Subsets</th>
<th>Simulation or version</th>
</tr>
</thead>
<tbody>
<tr>
<td>BigDataBench</td>
<td>Y</td>
<td>Five</td>
<td>Four</td>
<td>33</td>
<td>8</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>BigBench</td>
<td>Y</td>
<td>One</td>
<td>Three</td>
<td>10</td>
<td>3</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>CloudSuite</td>
<td>N</td>
<td>N/A</td>
<td>Two</td>
<td>8</td>
<td>3</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>HiBench</td>
<td>N</td>
<td>N/A</td>
<td>Two</td>
<td>10</td>
<td>3</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>CALDA</td>
<td>Y</td>
<td>N/A</td>
<td>One</td>
<td>5</td>
<td>1</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>YCSB</td>
<td>Y</td>
<td>N/A</td>
<td>One</td>
<td>6</td>
<td>N/A</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>LinkBench</td>
<td>Y</td>
<td>One</td>
<td>One</td>
<td>10</td>
<td>1</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>AMP Benchmarks</td>
<td>N</td>
<td>N/A</td>
<td>One</td>
<td>4</td>
<td>1</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>
Observations from CloudSuite

The characteristic of big data workloads

- High L1I miss
  - Frontend inefficiencies
- Low ILP
  - Core inefficiencies
- LLC is good but overprovision
  - Data-access inefficiencies
- Less off-chip bandwidth and MLP
  - Bandwidth inefficiencies

*Clearing the Clouds, ASPLOS 2012 Best paper*
System Behaviors of BigDataBench

- Diversified system level behaviors:

![Graph showing CPU utilization and I/O wait ratio for different BigDataBench benchmarks.](image)

- Weighted disk I/O time ratio

![Graph showing weighted disk I/O time ratio for different BigDataBench benchmarks.](image)
System Behaviors of BigDataBench

- Diversified system level behaviors:
  - High CPU utilization & less I/O time
System Behaviors of BigDataBench

- Diversified system level behaviors:
  - High CPU utilization & less I/O time
  - Low CPU utilization relatively and lots of I/O time
System Behaviors Of BigDataBench

- Diversified system level behaviors:
  - High CPU utilization & less I/O time
  - Low CPU utilization relatively and lots of I/O time
  - Medium CPU utilization and I/O
Workloads Classification Of BigDataBench

- Finding from system behaviors
  - System behaviors vary across different workloads
  - Workloads can be divided into 3 categories:

<table>
<thead>
<tr>
<th>Type</th>
<th>Workloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O Intensive</td>
<td>H-Read, H-Difference, I-SelectQuery, S-WordCount, S-Project, S-OrderBy, M-Grep and S-Grep</td>
</tr>
<tr>
<td>Hybrid</td>
<td>H-TPC-DS-query3, I-OrderBy, S-TPC-DS-query10, S-TPC-DS-query8, S-Sort, M-WordCount and M-Sort</td>
</tr>
</tbody>
</table>
The average IPC of the big data workloads are larger than CloudSuite, SPECFP and SPECINT, same as PARSEC and slightly smaller than HPCC

The average IPC of BigDataBench is 1.3 times of that of CloudSuite

Some workloads have high IPC (M_Kmeans, S-TPC-DS-Query8)
Instructions Mix of BigDataBench vs. other benchmarks

For big data workloads:
- More branch instructions
  The percentage is 20% (1.5 times larger than others), except for TPC-C
- The ratio of integer instructions to FP instructions is very high
  The average is 73
Cache Behaviors of BigDataBench vs. other benchmarks

- **L1I MPKI**
  - Larger than traditional benchmarks, but lower than that of CloudSuite (12 Vs. 31)
  - Different among big data workloads
    - CPU-intensive(8), I/O intensive(22), and hybrid workloads(9)
  - MPI workloads have less instruction cache miss
    - Only 3.4 on the average
Cache Behaviors of BigDataBench

- **L2 Cache:**
  - The IO-intensive workloads undergo more L2 MPKI

- **L3 Cache:**
  - The average L3 MPKI of the big data workloads is smaller than all of the other workloads

- The underlying software stacks impact data locality
  - MPI workloads have better data locality and less cache misses
Our observation from BigDataBench

- Unique characteristic
  - More branch instructions & Higher ratio of integer to FP instructions

- Different behaviors between Big Data workloads
  - Disparity of ILP and memory access behaviors
    - Several workloads can achieve higher IPC
    - Several workloads can achieve higher Off-chip bandwidth
  - CloudSuite is a subclass of Big Data

- High front-end stall is not the unique characteristics of big data workloads
  - Related with software stacks
BigDataBench Publications


- **Characterizing and Subsetting Big Data Workloads.** Zhen Jia, Jianfeng Zhan, Wang Lei, Rui Han, Sally A. McKee, Qiang Yang, Chunjie Luo, and Jingwei Li. In 2014 IEEE International Symposium on Workload Characterization (IISWC). IEEE, 2014.

- Characterizing data analysis workloads in data centers. 2013 IEEE International Symposium on Workload Characterization (IISWC 2013) (Best paper award)

- **BigOP: generating comprehensive big data workloads as a benchmarking framework.** 19th International Conference on Database Systems for Advanced Applications (DASFAA 2014)

- **BDGS: A Scalable Big Data Generator Suite in Big Data Benchmarking.** The Fourth workshop on big data benchmarking (WBDB 2014)
BigDataBench users

- More than 20 groups have published papers using BigDataBench

- http://prof.ict.ac.cn/BigDataBench/users/
Outline

- What is BigDataBench?

- *BigDataBench benchmarking methodology*
Five Steps

1. Investigate important application domain
2. Typical workloads and data set
3. Big data benchmark specification
4. Diverse implementation
5. Multi-tenancy or subset
BigDataBench Methodology

- Application Domain 1
- Application Domain ...
- Application Domain N

- Data models of different types & semantics
- Data operations & workload patterns

- Benchmark specification 1
- Benchmark specification ...
- Benchmark specification N

- Real-world data sets
- Data generation tools
- Workloads with diverse implementations

- BigDataBench subset
- Multi-tenancy version

- Mix with different percentages
- Reduce benchmarking cost
Investigate Application Domains

Internet Services

What application domains should we pay attention to?
Internet Services

Taking up 80% of internet services according to page views and daily visitors

Top 20 websites

http://www.alexa.com/topsites/global;0
Multimedia Data

Candidates for Big Data

- Surveillance
- Embedded and Medical
- Data Processing
- Entertainment and Social Media
- Consumer Images and Voice

(Share of Data That is Useful If Tagged and Analyzed)

<table>
<thead>
<tr>
<th>Multimedia Data</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>New VIDEOS on YouTube</td>
<td>600+</td>
</tr>
<tr>
<td>New PHOTOS on FLICKR</td>
<td>6600+</td>
</tr>
<tr>
<td>Music streaming on PANDORA</td>
<td>13000+</td>
</tr>
<tr>
<td>Video feeds from surveillance cameras</td>
<td>100’s</td>
</tr>
<tr>
<td>VOICE calls on Skype</td>
<td>370000+</td>
</tr>
<tr>
<td>Data growth</td>
<td>80%</td>
</tr>
</tbody>
</table>

The Explosive Growth of Human Genome Data

Figure 1. Approximate growth of different data populations. Note that the ‘short read archive’ (SRA) has grown faster than what would be predicted by Moore’s law. NGS began with a ‘bang’ in mid-2005 when Jonathan Rothberg’s company, 454, assembled a group of cutting-edge technologies and commercialized the first next-generation sequencer. Curves as indicated by colored text above the graph and as annotated, forecasts in growth indicated by the shaded region of the graph.

http://www.osehra.org/sites/default/files/genomics_and_big_data_1_0.pdf
Application Domain We Choose

- Internet Services
- Search Engine
- Social Network
- E-commerce
- Multimedia
- Bioinformatics

Emerging but Important

**Important**
BigDataBench Methodology

- Application Domain 1
- Data models of different types & semantics
- Application Domain ...
- Data operations & workload patterns
- Application Domain N
Success story: Relational model of data


- Set concept: general mathematical meaning
  - General representation of data
  - Basis of relational algebra (theoretical foundation of database)
  - 5 basic operations
    - Select, Project, Product, Union, Difference
Success story: parallel computing

The Landscape of Parallel Computing Research: A View from Berkeley

By a multidisciplinary group of well-known researchers

e.g.:
Jim Gray,
Michael Jordan
David A. Patterson

■ Operations & Patterns
  ■ Abstracted from 13 representative parallel computation patterns
  ■ Parallel computation → inherent demand for big data processing (volume & complexity)
Primitive Operations & Patterns in Big Data

- **3 Categories of Operations**
  - **11 basic operations**

<table>
<thead>
<tr>
<th>Categories</th>
<th>Typical Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element Operation</td>
<td>put, get, delete, transform, filter</td>
</tr>
<tr>
<td>Single-Set Operation</td>
<td>project, order by aggregation(min, max, sum, median, average)</td>
</tr>
<tr>
<td>Double-Set Operation</td>
<td>union, difference, cross product</td>
</tr>
</tbody>
</table>

- **3 Processing Patterns**

<table>
<thead>
<tr>
<th>Patterns</th>
<th>Example Workloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-Operation Processing</td>
<td>any abstracted operation</td>
</tr>
<tr>
<td>Multi-Operation Processing</td>
<td>operation combinations, SQL queries</td>
</tr>
<tr>
<td>Iterative Processing</td>
<td>graph traversal, finite state machines</td>
</tr>
</tbody>
</table>

BigOP: generating comprehensive big data workloads as a benchmarking framework. DASFAA 2014
Some Examples (not exhausted)

<table>
<thead>
<tr>
<th></th>
<th>Fast Storage</th>
<th>Log Monitoring</th>
<th>PageRank Computation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operations</td>
<td>put, get, delete union</td>
<td>put, get, filter, aggregation</td>
<td>get, transform, filter, order by</td>
</tr>
<tr>
<td>Patterns</td>
<td>single-operation</td>
<td>single- and multi-operation</td>
<td>all patterns</td>
</tr>
<tr>
<td>Data Set</td>
<td>randomly generated structured data</td>
<td>real server logs</td>
<td>randomly generated directed graph</td>
</tr>
<tr>
<td>Metrics</td>
<td>throughput</td>
<td>request latency statistics</td>
<td>test duration</td>
</tr>
</tbody>
</table>

**BigOP: generating comprehensive big data workloads as a benchmarking framework.**  
19th International Conference on Database Systems for Advanced Applications (DASFAA 2014)
Image Search

- Perceptual Hash Algorithm

Input Image → sampling → Basic information of image → SIFT → Image features → Hash → Image fingerprint → Database → Set Operation → Similar images → Sort → Output

BigDataBench | MICRO 2014
Feature Extraction--SIFT

- Gaussian Filter
- Input Image
- Convolution
- Image Scale Space
- Sampling
- Image Pyramid
- Matrix Subtraction
- DOG Image
- Sort
- Output
- Feature Vectors
- Count
- Gaussian Window
- Sampling
- Key Point Of Image
Workload & Data Set in BigDataBench

Data Model
- Structured
- Semi-Structured
- Unstructured

Semantics
- Text
- Graph
- Table
- Multimedia

Data Operations
- Unit of computation

Workload Patterns
- Different combination of units of computation
BigDataBench Methodology

Application Domain 1

Data models of different types & semantics

Application Domain ...

Data operations & workload patterns

Application Domain N

Benchmark specification 1

Benchmark specification ...

Benchmark specification N
Data management’s tradition

- Specification First.
- Functions of abstraction are units of computation that appear frequently in the application domain being benchmarked.
- They are expressed in a generic form that is independent of the underlying system implementation.
TPC-C examples

User-initiated operation
Database Transaction T1
Read row from table A
Update row in table B
Commit transaction
Database Transaction T2
Update row in table A
Insert row in table C
Commit transaction
Database Transaction T3
Read row from table C
Update row in table B
Commit transaction
BigDataBench Methodology

Application Domain 1

Application Domain ...

Application Domain N

Data models of different types & semantics

Data operations & workload patterns

Benchmark specification 1

Benchmark specification ...

Benchmark specification N

Real-world data sets

Data generation tools

Workloads with diverse implementations

BigDataBench MICRO 2014
# Real-World Data sets

## 14 real-world data sets

<table>
<thead>
<tr>
<th>No.</th>
<th>data sets</th>
<th>data set description</th>
<th>scalable data set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Wikipedia Entries [18]</td>
<td>4,300,000 English articles (unstructured text)</td>
<td>Text Generator of BDGS</td>
</tr>
<tr>
<td>2</td>
<td>Amazon Movie Reviews [8]</td>
<td>7,911,684 reviews (semi-structured text)</td>
<td>Text Generator of BDGS</td>
</tr>
<tr>
<td>3</td>
<td>Google Web Graph [11]</td>
<td>875713 nodes, 5105039 edges (unstructured graph)</td>
<td>Graph Generator of BDGS</td>
</tr>
<tr>
<td>4</td>
<td>Facebook Social Network [10]</td>
<td>4039 nodes, 88234 edges (unstructured graph)</td>
<td>Graph Generator of BDGS</td>
</tr>
<tr>
<td>5</td>
<td>E-commerce Transaction Data</td>
<td>Table 1: 4 columns, 38658 rows. Table 2: 6 columns, 242735 rows (structured table)</td>
<td>Table Generator of BDGS</td>
</tr>
<tr>
<td>6</td>
<td>ProfSearch Person Resumés</td>
<td>278956 resumés (semi-structured table)</td>
<td>Table Generator of BDGS</td>
</tr>
<tr>
<td>7</td>
<td>ImageNet [25]</td>
<td>ILSVRC2014 DET image dataset (unstructured image)</td>
<td>ongoing development</td>
</tr>
<tr>
<td>8</td>
<td>English broadcasting audio files [1]</td>
<td>Sampled at 16 kHz, 16-bit linear sampling (unstructured audio)</td>
<td>ongoing development</td>
</tr>
<tr>
<td>9</td>
<td>DVD Input Streams [2]</td>
<td>110 input streams, resolution:704*480 (unstructured video)</td>
<td>ongoing development</td>
</tr>
<tr>
<td>10</td>
<td>Image scene [3]</td>
<td>39 image scene description files (unstructured text)</td>
<td>ongoing development</td>
</tr>
<tr>
<td>11</td>
<td>Genome sequence data [4]</td>
<td>cfa data format (unstructured text)</td>
<td>4 volumes of data sets</td>
</tr>
<tr>
<td>12</td>
<td>Assembly of the human genome [5]</td>
<td>fa data format (unstructured text)</td>
<td>4 volumes of data sets</td>
</tr>
<tr>
<td>13</td>
<td>SoGou Data [16]</td>
<td>the corpus and search query data from SoGou Labs (unstructured text)</td>
<td>ongoing development</td>
</tr>
<tr>
<td>14</td>
<td>MNIST [13]</td>
<td>handwritten digits database which has 60,000 training examples and 10,000 test examples (unstructured image)</td>
<td>ongoing development</td>
</tr>
</tbody>
</table>
Big Data Generation Tool--BDGS

- Provide scalable data set extracted from real-world data sets
Naïve Text generator

select word randomly

words following multinomial distribution

- Only modeling on word level;
- Words are selected according to the same distribution
Improved Text generator

- Modeling on topic and word level
- Words are drawn from distribution under particular topic
- Topics are drawn from the same distribution, as a result, each document has the same topic proportion

topics: following multinomial distribution

words: following multinomial distribution under topic 2

select topic randomly

select word randomly
Optimized Text Generator

- Modeling on topic and word level
- Words are drew from distribution under particular topic
- Topics are selected from different distribution with parameters following a dirichlet distribution
Workloads With Diverse Implementations

Software Stack

MapReduce
MPI
DataMPI
Spark
BigDataBench Methodology

- Application
  - Domain 1
  - Domain ...
  - Domain N

- Data models of different types & semantics
- Data operations & workload patterns

- Benchmark
  - specification 1
  - specification ...
  - specification N

- Real-world data sets
- Data generation tools
- Workloads with diverse implementations

- Multi-tenancy version
- Mix with different percentages
- Reduce benchmarking cost
- BigDataBench subset
Multi-tenancy version of BigDataBench

- Scenarios of multiple tenants running heterogeneous applications in cloud datacenters
  - Latency-critical online services
  - Latency-insensitive offline batch applications

- Mining real-world Workload traces (Google and Facebook)
- Profiling Real-world Workload traces
- Workload matching using Machine learning techniques
- Parametric workload generation tool
- Benchmarking scenarios
  - Mixed workloads in public clouds
  - Data analytical workloads in private clouds

BigDataBench | MICRO 2014

INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES
BigDataBench Subset

Motivation

- Expensive to run all the benchmarks for system and architecture researches
  - multiplied by different implementations
  - BigDataBench 3.0 provides about 77 workloads
Subsetting Methodology

- Identify a comprehensive set of workload characteristics from a specific perspective
- Eliminate the correlation data in those metrics
- Map the high dimension metrics to a low dimension
- Use the clustering method to classify
- Choose representative workloads from each category
QUESTIONS And Answers