



# Demo - Smart City Use-case

Using ODPi Hadoop, Spark, H2O and Sparkling water

Ganesh Raju



# ODPi

- Simplify & standardize big data ecosystem with a common reference specification and test suites.
- Reduces cost and complexity and accelerates the development of Big Data solutions.
- Cross-compatibility between different distributions of Hadoop and big data technologies
- Has two stacks: Runtime and Operations
- V2.0 alpha release coming soon
- Linaro is a member of ODPi

# Spark

- Distributed and fast in-memory data processing engine
- Provides development APIs to efficiently execute iterative streaming, machine learning or SQL workloads
- Spark was developed as an alternative approach to Map Reduce with ease of use in mind.
- Code in Java, Scala, or Python.



# H2O Sparkling Water

- H2O is a in-memory user friendly machine learning API
- Compatible with Hadoop and Spark
- Spark + H2O is Sparkling Water
- Sparkling Water allows to combine fast & scalable machine learning algorithms of H2O with high performance distributed processing capabilities of Spark engine.
- Spark's RDD and DataFrame and H2O's H2OFrame are interoperable
- Users can utilize H2O Flow UI to drive Scala / R / Python computation from Spark



# Demo

- Utilizing ODPi v1 based Native Hadoop, Spark, H2O Sparkling Water, H2O flow.
- All Compiled on ARM - ODPi Hadoop 2.7, Spark 1.6 with Scala 2.10 (Scala 2.11 is not supported with SparklingWater)
- 3 node cluster running on Linaro Developer Cloud - HP MoonShot machines
- Dataset files stored in HDFS.
- Spark utilizing Yarn for Resource manager.
- H2O Sparkling water utilizing Spark as execution Engine.
- H2O Flow utilizing Spark SQL API and scala code
- .csv data -> HDFS -> Spark RDD -> H2O H2OFrame

<https://wiki.linaro.org/LEG/Engineering/BigData>



ENGINEERS AND DEVICES  
WORKING TOGETHER



# Benchmarking Big Data

Ganesh Raju and Naresh Bhat



# Abstract

- Various Benchmarking Tools
- Types of Benchmarks and standards
- Challenges of BigData benchmarking on ARM
- Some of the tools that we will be covering are TPC (Transaction Processing Performance Council) based TPCx-HS, TPC-DS, TPC-H benchmark, HiBench (TestDFSIO), Spark-Bench for Apache Spark, MRBench for Mapreduce, NNBench for HDFS...etc



# Why Benchmarking ..?

- Measure performance and scale
- Simulate higher load
  - Find bottlenecks/limits
- Evaluate different hardware/software
  - OS, Java, VM.
  - Hadoop, Spark, Pig, Hive..
- Validate reliability
- Validate assumptions / Configurations
- Compare two different deployments
- Performance tuning





# Challenges of BigData benchmarking

- System Diversity
  - Variety of Solutions - Data Read, I/O, Streaming, Data warehousing, Machine Learning
- Rapid Data Evolution - Velocity.
- System and Data Scale
- System Complexity
  - Multiple pipelines (layers of Transformations)



# Types of benchmarks and standards

- Micro benchmarks: To evaluate specific lower-level, system operations
  - E.g. Hadoop Workload Examples (sort, grep, wordcount and Terasort, Gridmix, Pigmix), HiBench, HDFS DFSIO, AMP Lab Big Data Benchmark
- Functional/Component benchmarks: Specific to low level function
  - E.g. Basic SQL queries (select, join, etc.,)
  - Synthetic benchmarks
- Application level
  - Bigbench
  - Spark bench



Benchmark Efforts - Microbenchmarks	Workloads	Software Stacks	Metrics
HiBench	Sort, WordCount, TeraSort, PageRank, K-means, Bayes classification, Index	Hadoop and Hive	Execution Time, Throughput, resource utilization
DFSIO	Generate, read, write, append, and remove data for MapReduce jobs	Hadoop	Execution Time, Throughput
AMPLab benchmark	Part of CALDA workloads (scan, aggregate and join) and PageRank	Hive, Tez	Execution Time



Benchmark Efforts - TPC	Workloads	Software Stacks	Metrics
TPCx-HS	HSGen, HSData, Check, HSSort and HSValidate	Hadoop	Performance, price and energy
TPC-H	Datawarehousing operations	Hive, Pig	Execution Time, Throughput
TPC-DS	Decision support benchmark Data loading, queries and maintenance	Hive, Pig	Execution Time, Throughput



Benchmark Efforts - Synthetic	Workloads	Software Stacks	Metrics
SWIM	Synthetic user generated MapReduce jobs of reading, writing, shuffling and sorting	Hadoop	Multiple metrics
GridMix	Synthetic and basic operations to stress test job scheduler and compression and decompression	Hadoop	Memory, Execution Time, Throughput
PigMix	17 Pig specific queries	Hadoop, Pig	Execution Time
MRBench	MapReduce benchmark as a complementary to TeraSort - Datawarehouse operations with 22 TPC-H queries	Hadoop	Execution Time
NNBench and NNBenchWithOutMR	Load testing namenode and HDFS I/O with small payloads	Hadoop	I/O
SparkBench	CPU, memory and shuffle and IO intensive workloads. Machine Learning, Streaming, Graph Computation and SQL Workloads	Spark	Execution Time, Data process rate
BigBench	Interactive-based queries based on synthetic data	Hadoop, Spark	Execution Time

Benchmark Efforts	Workloads	Software Stacks	Metrics
BigDataBench	<ol style="list-style-type: none"> <li>1. Micro Benchmarks (sort, grep, WordCount);</li> <li>2. Search engine workloads (index, PageRank);</li> <li>3. Social network workloads (connected components (CC), K-means and BFS);</li> <li>4. E-commerce site workloads (Relational database queries (select, aggregate and join), collaborative filtering (CF) and Naive Bayes);</li> <li>5. Multimedia analytics workloads (Speech Recognition, Ray Tracing, Image Segmentation, Face Detection);</li> <li>6. Bioinformatics workloads</li> </ol>	Hadoop, DBMSs, NoSQL systems, Hive, Impala, Hbase, MPI, Libc, and other real-time analytics systems	Throughput, Memory, CPU (MIPS, MPKI - Misses per instruction)



# Hadoop benchmark and Test tool

- Hadoop distribution comes with a number of benchmarks
- TestDFSIO, nnbench, mrbench are in `hadoop-*test*.jar`
- TeraGen, TeraSort, TeraValidate are in `hadoop-*examples*.jar`
- You can check it using the command

```
$ cd /usr/local/hadoop
$ bin/hadoop jar hadoop-*test*.jar
$ bin/hadoop jar hadoop-*examples*.jar
```

- While running the benchmarks you might want to use `time` command which measure the elapsed time. This saves you the hassle of navigating to the Hadoop JobTracker interface. The relevant metric is real value in the first row.

```
$ time hadoop jar hadoop-*examples*.jar ...
[...]
```

real	9m15.510s
user	0m7.075s
sys	0m0.584s



# TeraGen, TeraSort and TeraValidate

- This is a most well known Hadoop benchmark
- The TeraSort is to sort the data as fast as possible
- This test suite combines HDFS and mapreduce layers of a hadoop cluster
- TeraSort benchmark consists of 3 steps
  - Generate input via **TeraGen**
  - Run **TeraSort** on input data
  - Validate sorted output data via **TeraValidate**

<https://wiki.linaro.org/LEG/Engineering/BigData/HadoopBuildInstallAndRunGuide>





# HiBench

- Contains 9 typical Hadoop and Spark workloads (including micro benchmarks, HDFS benchmarks, web search benchmarks, machine learning benchmarks using Mahout, and data analytics benchmarks)
- Sort, WordCount, TeraSort, TestDFSIO, Nutch indexing (search indexing using Nutch engine), PageRank (An implementation of Google's Web page ranking algorithm), hivebench
- Uses zlib compression for input and output
- Metrics: Time (sec) & Throughput (Bytes/Sec), Memory partitions, parallelism,
- Cons: Lack of AARCH bits, Lack of documentations

<https://wiki.linaro.org/LEG/Engineering/BigData/HiBench>



ENGINEERS AND DEVICES  
WORKING TOGETHER

# TestDFSIO

- It is part of *hadoop-mapreduce-client-jobclient.jar*
- Stress test I/O performance (throughput and latency) on a clustered setup.
- This test will shake out the hardware, OS and Hadoop setup on your cluster machines (NameNode/DataNode)
- The tests are run as a MapReduce job using 1:1 mapping (1 map / file)
- Helpful to discover performance bottlenecks in your network
- Benchmark write test followed up with read test
- Use `-write` for write tests and `-read` for read tests.
- The results stored in `TestDFSIO_results.log`. Use `-resFile` to choose different file name



# Hive Testbench

- Based on TPC-H and TPC-DS benchmarks
- Experiment Apache Hive at any data scale
- Contains data generator and set of queries
- Test the basic Hive performance on large data sets

<https://wiki.linaro.org/LEG/Engineering/BigData/HiveTestBench>



# MR(Map Reduce) Benchmark for MR

- Loops a small job number of times
- Checks whether small job runs are responsive and running efficiently on your cluster
- Puts focus on MapReduce layer as its impact on the HDFS layer is very limited
- The multiple parallel MRBench issue is resolved. Hence you can run it from different boxes
- Test command to run 50 small test jobs
- Exemplary output, which means in 31 sec the job finished

```
$ hadoop jar hadoop-*test*.jar mrbench -numRuns 50
```

DataLines	Maps	Reduces	AvgTime (milliseconds)
1	2	1	31414

# NNBench and NNBenchmarkWithoutMR

- Load testing NameNode through continuous read, write, rename and delete operations on small files
- Stress tests HDFS (I/O)
- To increase stress, multiple instances of NNBenchmarkWithoutMR can be run simultaneously from several machines or increase map tasks for NNBench
- All write tests are run then followed by read tests
- The test command: The below command will run a NameNode benchmark that creates 1000 files using 12 maps and 6 reducers.

```
$ hadoop jar hadoop-*test*.jar nnbench -operation create_write \  
-maps 12 -reducers 6 -blockSize 1 -bytesToWrite 0 -numberOfFiles 1000 \  
-replicationFactorPerFile 3 -readFileAfterOpen true \  
-baseDir /benchmarks/NNBench-`hostname -s`
```



# TPC Benchmark

- TPCx-HS - <https://wiki.linaro.org/LEG/Engineering/BigData/TPCxHS>
  - Currently facing problems with cluster shell configuration
- TPC-H
  - TPC-H benchmark focuses on ad-hoc queries
- TPC-DS
  - “the” standard benchmark for decision support
- TPC-C
  - Is an on-line transaction processing (OLTP) benchmark



# TPCx-HS Benchmark

X: Express, H: Hadoop, S: Sort

The TPCx-HS kit contains

- TPCx-HS specification documentation
- TPCx-HS User's guide documentation
- Scripts to run benchmarks
- Java code to execute the benchmark load

## TPCx-HS Execution

- A valid run consists of 5 separate phases run sequentially with overlap in their execution
- The benchmark test consists of 2 runs (Run with lower and higher TPCx-HS Performance Metric)
- No configuration or tuning changes or reboot are allowed between the two runs



# TPC vs SPEC models

## TPC model

- Specification based
- Performance, Price, energy in one benchmark
- End-to-End
- Multiple tests (ACID, Load)
- Independent Review
- Full disclosure
- TPC Technology conference

## SPEC model

- Kit based
- Performance and energy in separate benchmarks
- Server centric
- Single test
- Summary disclosure
- SPEC research group ICPE





# BigBench

- BigBench is a joint effort with partners in industry and academia on creating a comprehensive and standardized BigData benchmark.
- BigBench builds upon and borrows elements from existing benchmarking efforts (such as [TPC-xHS](#), [GridMix](#), [PigMix](#), [HiBench](#), [Big Data Benchmark](#), [YCSB](#) and [TPC-DS](#)).
- BigBench is a specification-based benchmark with an open-source reference implementation kit.
- As a specification-based benchmark, it would be technology-agnostic and provide the necessary formalism and flexibility to support multiple implementations.
- Focused around execution time calculation
- Consists of 30 queries/workloads (10 of them are from TPC)
- Drawback - it is structured-data-intensive



# Spark Bench for Apache Spark

- Build on ARM works
- FAIL: When spark bench examples are run, a KILL signal is observed which terminates all workers.
- This is still under investigation as there are no useful logs to debug. No proper error description and lack of documentation is a challenge.
- A ticket is already filed on spark bench git which is unresolved.
- Con: Lack of documentation.



# GridMix

- Mix of Synthetic Mapreduce jobs (sorting text data and SequenceFiles)
- Evaluate MapReduce and HDFS performance
- The input file needs to be in JSON format
- Jobs can be either LOADJOB (trace of history logs using Rumen) or SLEEPJOB (A synthetic job where each task does \*nothing\* but sleep for a certain duration)
- Jobs can be run in STRESS, REPLAY or SERIAL mode
- You can emulate number of users, number of job queries and resource usage (CPU, memory, JVM heap)
- Basic command line usage: (Provided as part of hadoop command)  

```
$ hadoop gridmix [-generate <size>] [-users <users-list>] <iopath> <trace>
```
- Con: Challenging to explore the performance impact of combining or separating workloads, e.g., through consolidating from many clusters.



# PigMix

- PigMix is a set of queries used test Apache Pig performance
- There are queries that test latency (How long it takes to run this query ?)
- Queries that test scalability (How many fields or records can pig handle before it fails ?)
- Usage: Run the below commands from pig home

```
ant -Dharness.hadoop.home=$HADOOP_HOME pigmix-deploy (generate test dataset)
```

```
ant -Dharness.hadoop.home=$HADOOP_HOME pigmix (run the PigMix benchmark)
```



# SWIM(Statistical Workload Injector for MapReduce)

- Enables rigorous performance measurement of MapReduce systems
- Contains suites of workloads of thousands of jobs, with complex data, arrival, and computation patterns
- Informs both highly targeted, workload specific optimizations
- Highly recommended for MapReduce operators
- Performance measurement

<https://github.com/SWIMProjectUCB/SWIM/wiki/Performance-measurement-by-executing-synthetic-or-historical-workloads>



# AmpLab

- The Big Data Benchmark from AMPLab, UC Berkeley provides quantitative and qualitative comparisons of five systems
  - Redshift – a hosted MPP database offered by Amazon.com based on the ParA warehouse
  - Hive – a Hadoop-based data warehousing system
  - Shark – a Hive-compatible SQL engine which runs on top of the Spark computing framework
  - Impala – a Hive-compatible\* SQL engine with its own MPP-like execution engine
  - Stinger/Tez – Tez is a next generation Hadoop execution engine used in Spark
- This benchmark measures response time on a handful of relational queries: scans, aggregations, joins, and UDF's, across different data sizes.



# BigDataBench

BigDataBench is a benchmark suite for scale-out workloads, different from SPEC CPU (sequential workloads), and PARSEC (multithreaded workloads). Currently, it simulates five typical and important big data applications: search engine, social network, e-commerce, multimedia data analytics, and bioinformatics.

Includes 15 real-world data sets, and 34 big data workloads.





**Linaro  
connect**

Las Vegas 2016

ENGINEERS  
AND DEVICES  
WORKING  
TOGETHER

# References

<https://www2.eecs.berkeley.edu/Pubs/TechRpts/2011/EECS-2011-21.pdf>

**Terasort, TestDFSIO, NNBench, MRBench**

<https://wiki.linaro.org/LEG/Engineering/BigData>

<https://wiki.linaro.org/LEG/Engineering/BigData/HadoopTuningGuide>

<https://wiki.linaro.org/LEG/Engineering/BigData/HadoopBuildInstallAndRunGuide>

<http://www.michael-noll.com/blog/2011/04/09/benchmarking-and-stress-testing-an-hadoop-cluster-with-terasort-testdfsio-nnbench-mrbench/>

**GridMix3, PigMix, HiBench, TPCx-HS, SWIM, AMPLab, BigBench**

<https://hadoop.apache.org/docs/current/hadoop-gridmix/GridMix.html>

<https://cwiki.apache.org/confluence/display/PIG/PigMix>

<https://wiki.linaro.org/LEG/Engineering/BigData/HiBench>

<https://wiki.linaro.org/LEG/Engineering/BigData/TPCxHS>

<https://github.com/SWIMProjectUCB/SWIM/wiki>

<https://github.com/amplab>

<https://github.com/intel-hadoop/Big-Data-Benchmark-for-Big-Bench>





# Thank you

[ganesh.raju@linaro.org](mailto:ganesh.raju@linaro.org)

[naresh.bhat@linaro.org](mailto:naresh.bhat@linaro.org)

#LAS16

For further information: [www.linaro.org](http://www.linaro.org)

LAS16 keynotes and videos on: [connect.linaro.org](http://connect.linaro.org)

