Hadoop Data Integration Benchmark

*Product Profile and Evaluation:
RedPoint Data Management for Hadoop*

By William McKnight and Jake Dolezal
August 2016

Sponsored by RedPoint Global Inc.
Table of Contents

EXECUTIVE SUMMARY 3

HADOOP IN THE ENTERPRISE 4

THE EVOLUTION OF HADOOP DATA INTEGRATION 5

REDPOINT PRODUCT PROFILE 6

COMPANY PROFILE 6

BENCHMARK OVERVIEW 7

BENCHMARK SETUP 8

VIRTUAL SERVER ENVIRONMENT 8

REDPOINT INSTANCES 9

RELATIONAL DATABASE INSTANCE 9

HADOOP CLUSTER 9

SOURCE DATA 9

RELATIONAL DATA SOURCE 9

WEB-CLICK LOG 10

COUPON LOG 10

NAME AND ADDRESS CSV FILE 11

DATA VOLUME 11

DATA MANAGEMENT JOBS 12

WEB-COUPON LOG ON HADOOP JOIN WITH ORDERS JOB DESIGN 12

ADDRESS STANDARDIZATION JOB DESIGN 13

NAME MATCHING JOB DESIGN 14

BENCHMARK RESULTS 16

USE CASE 1: WEB-COUPON LOG ON HADOOP JOIN WITH ORDERS 16

EXECUTION TIME AND ACTUAL-VERSUS-EXPECTED RESULTS 16

VENDOR COMPARISON 17

USE CASES 2 AND 3: ADDRESS STANDARDIZATION AND NAME MATCHING 18

EXECUTION TIME AND ACTUAL-VERSUS-EXPECTED RESULTS 18

PERCEIVED USABILITY ASSESSMENT 18

CONCLUSION 19

ABOUT MCG GLOBAL SERVICES 20

ABOUT REDPOINT GLOBAL 21
Executive Summary

This benchmark is part of research into the performance of loads on Hadoop clusters — an increasingly important platform for storing data-powering corporate strategies.

The intent of the benchmark’s design is to simulate a set of basic-load scenarios to answer some fundamental business questions that organizations from nearly any industry might encounter and ask.

For a growing industry, there are a surprising variety of approaches and vendor architectures for Hadoop-loading products (such as: MapReduce, Spark, Spark through Hive, YARN, nifi, Sqoop, Sqoop interfaces, Flume interfaces, and interfaces to command line HDFS). Based on the differences in the results we’ve found, this architecture foundation greatly influences performance.

RedPoint Data Management for Hadoop is based on YARN, a resource negotiator a.k.a. operating system, which is the foundation of Hadoop 2.0.

In the case of our queries, RedPoint was able to complete workloads in a very short time frame, well within enterprise requirements and faster than what we imagined possible. Compared to a previous benchmark, one workload ran 550% faster than a product using Spark and 1900% faster than a product using MapReduce. RedPoint’s platform, continually fine-tuned for over a decade, has achieved unparalleled high performance in utilizing YARN without the overhead of other Hadoop components.

This paper further explores and investigates these results.
Hadoop in the Enterprise

Companies are clamoring to capture as much data as possible and harness that data as meaningful information to drive their businesses. Today, this information, or “big data,” would include all data generated by a company’s digital strategy. It would also include all data that past technologies were unable to record and analyze for business use. Big data is not only controllable today, but its implementation is also essential in conducting business.

Machines are primarily responsible for big data. Machine data contains critical insights; it allows us to conduct unprecedented triangulation of physical objects. Unlike traditional structured data (for example, data stored in a traditional relational database for batch reporting) machine data is non-standard, highly diverse, dynamic, and high-volume.

We can build a comprehensive picture of activity when we correlate and visualize the related events across disparate sources. The challenge is in bringing the data together. Companies that can capture and harness this data will benefit accordingly. In other words, the more companies store and process data, the more success they can tap into. Businesses across industries show clear, upward trends in spending on big data, and it is projected to be the top budget item in many sectors.

Hadoop is a technology that was formed in 2006 to meet the needs of the Silicon Valley data elite. Previously, these companies had data needs that far surpassed budgets for the database management systems (DBMS) out there. The scale they were using was another order of magnitude away from the target for the DBMS. And the timing of the scale was not certain, given the variability of the data.

Hadoop is quickly being adopted by businesses from start-up companies to the Fortune 1000 because it scales very well and relatively cheaply. This means you do not have to accurately predict the data size at the outset.

Hadoop is a great fit for many types of data in an organization. Sensor data, clickstream data, social data, server logs, smart grid data, electronic medical records, video and pictures, unstructured text, geolocation data, high-volume data, and “cold” enterprise data are all a great fit in the Hadoop open-source software framework for storing data on clusters of commodity hardware.

Scale-out file systems that may be lacking in functionality, but can handle modern levels of complex data are here to stay. Hadoop is the epitome of that idea and an ecosystem is building up around it.

While there used to be little overlap between reasonable selection of Hadoop and reasonable selection of a DBMS, that has changed. Hadoop has withstood the test of time and has grown
to the point where quite a few applications architected on a DBMS will be moved to Hadoop. The cost savings, combined with the ability to execute the complete application will be persuasive. It is especially useful as a collection point for post-operational data across the enterprise, not all of which may be destined for a relational data warehouse. This “data lake” can be left at low refinement, which is just fine for the emerging class of data scientists and others in need of deep insight.

Traditionally, data preparation has consumed an estimated 80 percent of legacy data development efforts. Loading Hadoop clusters will continue this tradition as a top job at a range of companies. Luckily, it is possible to lessen the cost and risk of this work with a robust data integration tool.

The Evolution of Hadoop Data Integration

In the early days, low-performing, open source vendor architectures like Sqoop, Flume, command line HDFS and Hive were limiting. Since then, numerous approaches and tools have arisen to meet the Hadoop data integration challenge.

MapReduce was the original [and in Hadoop 1.0, the only] data-processing engine for Hadoop. However, it has proved unwieldy and unable to meet increasingly complex workloads, suffering from issues such as an inability to scale index-based lookups.

Spark emerged as a replacement for MapReduce. By utilizing a pool of persistent "executor services" it can nearly eliminate inter-stage startup costs — one of MapReduce's big weakness. In addition, Spark uses Resilient Distributed Datasets (RDDs) for inter-stage storage. RDDs are a form of HDFS-backed memory images that combine the fast access of memory with the fault-tolerance of HDFS. Spark can be used to achieve very fast throughput for certain workloads.

Spark is also being leveraged to improve the performance of Hive processing, specifically HQL queries. So-called "Hive on Spark" has the ability to accelerate Hive itself, but doesn't serve as a general data-integration platform.

But even Spark has its limitations. The amount of memory required to process a dataset can be an order of magnitude larger than the input dataset size. If less memory is available due to various factors (such as cluster load, node downtime, or unexpected data scale), Spark's performance degradation curve can be severely non-linear, even becoming a “cliff” beyond which jobs simply fail. It is increasingly impossible to expect a “reserved” cluster for Hadoop activity, which means a cluster’s memory resources are increasingly limited and unpredictable.

Still, Spark would be the number one choice for most workloads if these were your only options.
However, by applying engineering to the cluster to achieve higher performing results with true commodity nodes — without the added memory — some have improved upon preceding models. For example, RedPoint uses a native engine on top of YARN, a resource negotiator and operating system, which is the foundation of Hadoop 2.0. It is the layer that integrates and manages resources, including storage resources, CPU, I/O and memory.

RedPoint is based around YARN, which runs in the cluster. By leveraging YARN, it can run in massive parallelism without the assumption that all the data must fit into memory. Workload performance is more predictable as well, given its lack of dependency on memory. Additionally, the degradation curve when faced with limited resources is more gentle.

RedPoint Data Management™ for Hadoop leverages RedPoint’s 10-year legacy with the high-performance RedPoint Data Management data integration tool. It uses a visual-design dataflow model, allowing non-programmers to create complex data transformations. Organizations with existing data staff should find this technology to have a faster and more affordable adoption curve than when hiring for Spark.

## RedPoint Product Profile

<table>
<thead>
<tr>
<th>Product Name</th>
<th>RedPoint Data Management for Hadoop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Launch</td>
<td>2013</td>
</tr>
<tr>
<td>Current Release and Date</td>
<td>7.3.1, June 2016</td>
</tr>
<tr>
<td>Key Features</td>
<td>Based on YARN; Company with 10-year legacy with the high-performance RedPoint Data Management data integration and data quality tool; Predictable high performance</td>
</tr>
<tr>
<td>Hadoop DI Competitors</td>
<td>Informatica, Pentaho, Syncsort, Talend</td>
</tr>
</tbody>
</table>

## Company Profile

<table>
<thead>
<tr>
<th>Company Founded</th>
<th>2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus</td>
<td>Empower data-driven organizations by unlocking the full value of their data to drive consumer engagement and profitable, sustained growth.</td>
</tr>
</tbody>
</table>
Benchmark Overview

The intent of the benchmark’s design is to simulate a set of basic scenarios to address some fundamental business problems that an organization from nearly any industry sector might encounter and ask. These common business questions formulated for the benchmark and from our experience working with a range of clients over the past decade are:

- What impact does customers’ views of pages and products on our website have on sales? What is the average number of page views before customers make a purchase decision (online or in-store)?
- How do our coupon promotional campaigns impact our product sales or service utilization? Do our customers who view or receive our coupon promotions come to our website and buy more or additional products than they may have otherwise purchased?
- How can we identify and remove potential duplicates from a customer data source of questionable data quality?
- How can we standardize customer mailing addresses to improve the quality of our geographic data for same-household recognition and for the efficacy of our mail-marketing campaigns?

The benchmark was designed to demonstrate how a company might approach addressing these business problems by bringing different sources of information into play. We also have taken the opportunity to show how Hadoop can be leveraged, because some of the data of interest in these data management cases are likely of a large volume and non-relational or semi- to unstructured in nature. In these cases, using Hadoop would be the best course of action for clients seeking to answer these questions.

Since it is highly probable that the data required resides in different sources, the benchmark was also setup for data integration. Some of these sources are also probably not being consumed and aggregated into an enterprise data warehouse due to their high volume and the difficulty in integrating voluminous amounts of semi-structured data into a traditional data warehouse.

Thus, the benchmark was designed to mimic common scenarios and the challenges faced by organizations seeking to integrate data to address these and similar business problems.
Benchmark Setup

The benchmark was executed using the following setup, environment, standards, and configurations.

**Virtual Server Environment**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hadoop Distribution</td>
<td>Hortonworks Data Platform 2.4.2 (HDFS, MapReduce2, YARN, Tez, Hive, Pig, ZooKeeper, and Ambari installed)</td>
</tr>
<tr>
<td>EC2 Instance</td>
<td>Memory optimized m3.xlarge (4 vCPUs, 16 GB Memory)</td>
</tr>
<tr>
<td>OS</td>
<td>CentOS 6.7</td>
</tr>
<tr>
<td>Source Data Types</td>
<td>Text-based log files, a relational database, and comma-separated value (CSV) files</td>
</tr>
<tr>
<td>Data Volume</td>
<td>20GB (Log files); 7,500,000 rows (RDBMS); and 10,000,000 lines (CSV)</td>
</tr>
<tr>
<td>TPC-H Scale Factor</td>
<td>1x</td>
</tr>
<tr>
<td>RDBMS</td>
<td>PostgreSQL 9.4</td>
</tr>
<tr>
<td>Java Version</td>
<td>1.8.0_91</td>
</tr>
</tbody>
</table>

*Figure 1 and Table 1: Server Environment and Setup*
The benchmark was setup using Amazon Web Services (AWS) EC2 instances deployed into an AWS Virtual Private Cloud (VPC) within the same Placement Group. According to Amazon, all instances launched within a Placement Group have low latency, full bisection and 10 Gigabits per second bandwidth between instances.

**RedPoint Instances**
The RedPoint Client EC2 instance was a general purpose t2.large with 2 vCPUs and 8GB of RAM running CentOS 6.7. This Windows instance ran Microsoft Server 2012. On this instance, we installed the RedPoint Data Management for Hadoop Client version 7.3.1.

The RedPoint Execution and Site Server EC2 instance was a general-purpose, m4.xlarge machine with 4 vCPUs and 16GB of RAM running CentOS 6.7. In this instance, we installed the RedPoint Data Management Execution and Site Servers version 7.3.1.

**Relational Database Instance**
The relational source for the benchmark was a m4.xlarge EC2 instance running CentOS 6.7. We installed PostgreSQL 9.4 on this server.

**Hadoop Cluster**
The Hadoop cluster for the benchmark consisting of 3 identical nodes, each a m4.xlarge EC2 instance running CentOS 6.7. We installed Hortonworks Data Platform Hadoop distribution. Using Ambari, we installed the following Hadoop services: HDFS, MapReduce2, YARN, Tez, Hive, Pig, and ZooKeeper.

This is a minimum viable product (MVP) setup.

**Source Data**
We created the data sources used in the benchmark to mimic real-life use cases:

- Relational data
- Web-click log
- Coupon log
- Customer names and addresses

**Relational Data Source**
The relational source for the benchmark (stored in PostgreSQL) was constructed using the Transaction Processing Performance Council TPC Benchmark H (TPC-H) Revision 2.17.1 Standard Specification. The TPC-H database was constructed to mimic a real-life point-of-sale system according to the entity-relationship diagram and the data type and scale specifications provided by TPC-H.
the TPC-H.

We populated the database with scripts that were seeded with random numbers to create the mock dataset. The TPC-H specifications have a scale factor by which the record count for each table is derived. For this benchmark, we selected a scale factor of 1. In this case, the TPC-H database contained 1.5 million records in the ORDERS table and 6 million records in the LINEITEM table.

**Web-Click Log**

A web-click log was generated using the same fashion as a standard Apache web server log file. The log file was generated using scripts to simulate two types of entries: 1. completely random page views (seeded by random numbers) and, 2. web-clicks that correspond to actual page views of ordered products (seeded by random records in the TPC-H ORDERS and LINEITEMS tables).

The “dummy” or “noise” web-log entries appeared in a variety of possibilities but followed the same format consistent with an Apache web-click log entry. All data were randomly selected.

For example:

```
```

The “signal” web log entries that corresponded to (and were seeded with) actual ORDERS and LINEITEM records had the same randomness as the “dummy” entries. Except actual LINEITEM.L_PARTKEY values and corresponding ORDERS.O_ORDERDATE values from the TPC-H database were selected to create records to represent a page view of an actual ordered item on the same day as the order. The segments below represent those that potentially correspond to actual orders:

```
```

The web-click log file contained 64,000,000 lines and was 5.4GB in size. There were randomly-inserted, web-click entries that corresponded to certain LINEITEM and ORDERS records. Approximately 1 in 1,000 of the web-click log entries corresponded to orders. The rest of the entries were random.

**Coupon Log**

A coupon log was generated using the same fashion as a customized Apache web server log file. The coupon log was designed to mimic a special case log file generated whenever a potential customer viewed an item because of a click-through from a coupon ad campaign. Again, the log file was generated using scripts to simulate two types of entries: 1. completely random page views (seeded by random numbers) and, 2. page views that correspond to actual page views of
ordered products by actual customers via the coupon ad campaign (seeded by random records in the TPC-H ORDERS and LINEITEMS tables).

The “dummy” or “noise” coupon log-entry data were randomly selected. The “signal” coupon log entries that corresponded to, and were seeded with, actual ORDERS and LINEITEM records had the same randomness as the “dummy” entries. Except actual LINEITEM.L_PARTKEY values and corresponding ORDERS.O_ORDERDATE values from the TPC-H database were selected to create records to represent a page view of an actual ordered item on the same day as the order. The segments below represent those that potentially correspond to actual orders:

```
```

The coupon log file contained 16,000,000 entries and was 14.3GB in size. There were randomly-inserted coupon entries that corresponded to certain LINEITEM and ORDERS records. Approximately 1 in 1,000 of the coupon log entries corresponded to orders. The rest of the entries were random.

**Name and Address CSV File**

The customer name and address data was in a comma-separated value file format and stored in the Hadoop Distributed File System on our cluster. The layout of the file is demonstrated by the first few lines of the 10 million rows:

```
"NAME", "ADDRESS", "CITY", "STATE", "ZIP", "PHONE", "ID"
CELESTE A ZIENUK, 125 MINOT AVE, EAST WAREHAM, MA, 02538, , 100000022
SEBASTIAO C BARBOSA, 15 HOOSAC ST, ADAMS, MA, 01220, , 100000064
GREG S STURGEON, 1640 ALVIN LN, BROOKFIELD, WI, 53045, , 100000075
RENAE BATTISTELLA, 15 COMMONWEALTH AVE, QUINCY, MA, 02169, , 100000080
```

The names were randomly generated from a generic name database. The addresses are real addresses. However, just over 2 million of the addresses were “dirty,” i.e., not up to USPS standards. Since RedPoint uses a CASS (Coding Accuracy Support System) standardization module validated by the United States Postal Service (USPS), it was necessary to correct and match US street addresses for these 2 million entries.

**Data Volume**

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Type</th>
<th>Location</th>
<th>Rows</th>
<th>Size on Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Log</td>
<td>Apache Log</td>
<td>HDFS</td>
<td>64,000,000</td>
<td>5.5GB</td>
</tr>
<tr>
<td>Coupon Log</td>
<td>Apache Log</td>
<td>HDFS</td>
<td>16,000,000</td>
<td>14.3GB</td>
</tr>
<tr>
<td>Orders</td>
<td>RDBMS</td>
<td>PostgreSQL</td>
<td>1,500,000</td>
<td>N/A</td>
</tr>
<tr>
<td>Line Items</td>
<td>RDBMS</td>
<td>PostgreSQL</td>
<td>6,000,000</td>
<td>N/A</td>
</tr>
<tr>
<td>Names and Addresses</td>
<td>CSV</td>
<td>HDFS</td>
<td>10,000,000</td>
<td>0.6GB</td>
</tr>
</tbody>
</table>
Table 2: Benchmark source data volumes

Each of the data sources (the TPC-H database, log files, and customer address CSV file) were also scaled to different scale factors, so that the integration routines (described in the next section) could be executed against data sources of various sizes.

Data Management Jobs

The use case of the benchmark was designed to demonstrate real-life data management scenarios where companies desire to integrate data from their transactional systems with unstructured and semi-structured data. The benchmark demonstrates this by executing routines that:

- Integrate the TPC-H relational source data with the individual log files
- Standardize customer addresses
- Identify duplicate customer records

The following data management and integration routines were created for the benchmark. In all cases, best practices were observed to optimize the performance of each job.

Web-Coupon Log on Hadoop Join with Orders Job Design

The purpose of the Web-Coupon Log on Hadoop Join with Orders was to test the capability of the vendor software to efficiently combine a variety of data from multiple sources, both on and off Hadoop. Figure 3 represents the job design that was created in the RedPoint Data Management Client.

RedPoint offers a Parallel Section tool with inputs that define all the splittable data available to the Parallel Section transforms. Splittable data is then divided up among a set of tasks to be processed in parallel. Input tools within the Parallel Section tool’s processing area read their entire input data in each task and are used to define and drive data parallelism.

Within the Hadoop Parallel Section, two CSV input sources were read: Web Log and Coupon Log.

The Number Records tool was used to generate a sequence of numeric identifiers for individual records in each CSV input row.

The Calculate tool was used to convert the string Apache log date to a date format with the RedPoint ScanDateTime function:

```
ScanDateTime(Trim(DATESTR, "[ "), "DD/Mmm/YYYY:HH:mm:ss")
```
The Select tool is similar to the SQL SELECT clause. We used this tool to select only a few, necessary fields from the log inputs. The selected set of fields was used for the join and the output table.

The Join tool accepts two inputs—Left and Right—and matches records from both inputs on a single key field or column. We used the Cartesian Join option to combine the matched Left (Web Log) and Right (Coupon Log) records into a single "wide" record containing all fields from both inputs. This function is similar to an SQL join.

<table>
<thead>
<tr>
<th>Web Log</th>
<th>Coupon Log</th>
<th>Join</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>IP</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PARTKEY</td>
<td>PARTKEY</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>COUPONID</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CUSTOMERID</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Fields selected from the Web and Coupon logs used for the Join and output

The resulting output completed the preceding Parallel Section within Hadoop. However, while these parallel tasks were processing, the RedPoint Execution Server was also processing the RDBMS input task.

We used the RDBMS Input tool to read data from the PostgreSQL TPC-H database and tables by executing the following query:

```sql
SELECT L_ORDERKEY, L_PARTKEY, O_CUSTKEY, O_ORDERDATE FROM LINEITEM
LEFT OUTER JOIN ORDERS ON L_ORDERKEY = O_ORDERKEY;
```

We attached a Data Viewer to the output of the final Join between the joined Web-Coupon log Hadoop output and the RDBMS to inspect the resultant data set. The resulting execution times and expected output are discussed in the next section.

**Address Standardization Job Design**

The purpose of the Address Standardization job was to assess the ability of the RedPoint platform to quickly and accurately detect and correct malformed US postal addresses in a single source of data on Hadoop. Figure 4 represents the job design that was created in RedPoint Data Management Client.
Again, the RedPoint Parallel Processing Container was used to take advantage of the multiple thread capacity of our Hadoop cluster.

The 10-milion-item customer name and address CSV file was used as the primary input. For this job, we set the workload to be split by partition and used the ZIP Code as the partition field. This made the standardization more efficient by organizing the records. We also set the Partition Mode to Segment, because a Segment partition is faster than one based on a sort, according to the vendor’s documentation.

We used the RedPoint AO Address Quality tool to provide the address correction, parsing, and standardization. You can enable geocode assignment with a single option. For this workload, we loaded the USPS CASS-certified compressed tar file (tgz) right onto HDFS, and the RedPoint Execution Server was able to bring it directly into the Parallel processing segment of the job. The tool went through the data set and standardized the CSV file.

Next, we used the Filter tool to select only those addresses that were standardized and changed.

Again, we attached a Data Viewer to the output of the parallel Hadoop process to inspect the resultant data set. The resulting execution times and actual-versus-expected output are discussed in the next section.

**Name Matching Job Design**

The purpose of the Name Matching job was to assess the ability of the platform to quickly and accurately detect potential duplicate customer records by name and address within a single source of data on Hadoop. Figure 5 represents the job design created in the RedPoint Data Management Client.

Once again, the RedPoint Parallel Processing Container was used to take advantage of the multiple thread capacity of our Hadoop cluster.

The 10-milion-item customer name and address CSV file (the same one used in the Address Standardization job) was used as the primary input. For this job, we set the workload
to be split by partition and used the ZIP Code as the partition field. Since the address is important to identifying matches, the ZIP was an efficient means of getting potential matches grouped closer together, instead of in random order. We also set the Partition Mode to Segment for performance purposes, just as we did in the Address Standardization job.

We used the AO Consumer Match macro to match individuals using name and address information — in this case, we set the segmentation to ZIP + address parts. The AO Consumer Match can also be used to match the individual (full name), the family (last name only) or by address (no name components). It even has additional parameters designed to match female individuals who may have changed their surnames. We used the default scores produced by the matching algorithm and did not fine-tune them in any way.

Next, we used the Filter tool to remove unmatched records out of the data output.

Then, we used the Calculate tool to offset the group identifier produced by the AO Consumer Match tool by task number. This made them globally unique.

As the final task in the Parallel Section, we sorted the dataset by the group identifier, so we could see matches adjacent to each other.

Finally, we attached a Data Viewer to the output of the parallel Hadoop process to inspect the resultant data set. The resulting execution times and actual-versus-expected output are discussed in the next section.
Benchmark Results

Use Case 1: Web-Coupon Log on Hadoop Join with Orders

The goal of the first use case for the benchmark was to prepare a data set that correlates products ordered with the page views and coupon campaign click-throughs on an e-commerce website. The integration job was written to map the page views and coupons to products ordered. Figure 6 is a conceptual mapping of this integration.

![Figure 6: Web-Coupon Log On Hadoop Join with Orders Mapping]

### Execution Time and Actual-Versus-Expected Results

Table 4 lists the median execution times of the Web-Coupon Log On Hadoop Join with Orders job.

<table>
<thead>
<tr>
<th>Job</th>
<th>Trials</th>
<th>Median Run Time</th>
<th>Output Rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web-Coupon Log On Hadoop Join with Orders</td>
<td>5</td>
<td>3m 47s</td>
<td>160,176</td>
</tr>
</tbody>
</table>

Table 4: Web-Coupon Log On Hadoop Join with Orders Benchmark Results
Vendor Comparison
As a comparison with the rest of the data management industry, the results of this benchmark were compared against a benchmark run by MCG Global Services in late 2015, comparing Talend and Informatica.¹

Hadoop MapReduce, Apache Spark, and YARN represent a critical architectural choice that many information management professionals must make. Thus, the results of the previous benchmark are valuable when evaluating RedPoint’s performance and capabilities.

The Web-Coupon Log On Hadoop Join with Orders job created in RedPoint used the same data volume and variety, a nearly identical job design, and comparable EC2 instances to the achieve the benchmark workload output as the previous benchmark.

<table>
<thead>
<tr>
<th>Vendor Platform</th>
<th>Execution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hadoop MapReduce</td>
<td>1h 11m 52m</td>
</tr>
<tr>
<td>Apache Spark</td>
<td>20m 43s</td>
</tr>
<tr>
<td>RedPoint on Hadoop (YARN only)</td>
<td>3m 47s</td>
</tr>
</tbody>
</table>

*Table 5: RedPoint performance compared to a previous benchmark*

RedPoint was able to complete the same workload 550% faster than Talend using Spark and 1900% faster than Informatica using Hadoop MapReduce. This demonstrates how RedPoint designed its platform and performance over the span of a decade. Moreover, it indicates how RedPoint achieved with their platform that has been continually tuned for over a decade and utilizes YARN.

Use Cases 2 and 3: Address Standardization and Name Matching

The goal of the second and third use cases for the benchmark was to prepare datasets of sanitized customer addresses and matching customer duplicates. The data quality jobs were written to make use of and assess RedPoint’s toolset.

Execution Time and Actual-Versus-Expected Results
Table 6 lists the median execution times of the Address Standardization and Name Matching jobs.

<table>
<thead>
<tr>
<th>Job</th>
<th>Trials</th>
<th>Median Run Time</th>
<th>Output Rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address Standardization</td>
<td>5</td>
<td>0:02:30</td>
<td>2,005,055</td>
</tr>
<tr>
<td>Name Matching</td>
<td>5</td>
<td>0:02:52</td>
<td>6,367,507</td>
</tr>
</tbody>
</table>

Table 6: Address Standardization and Name Matching Benchmark Results

The benchmark produced very satisfactory data quality output within a range we expected based on the original source data generated. What was impressive was RedPoint’s performance. While we have no other previous benchmark with which to compare these results, the Address Standardization workload processed 10 million records at a rate of 66,667 records per second, and the Name Matching was achieved at 58,140 records per second.

These results are a testament to the power of RedPoint’s ability to leverage the Hadoop cluster for parallel processing via YARN with minimal overhead.

Perceived Usability Assessment
Important, but often-overlooked, considerations when benchmarking and evaluating data management tools are product usability and maturity. In previous benchmarks and client engagements, we have seen tools that rank highly for how easy they are to install, configure, understand, and use. We have also seen some that are quite difficult to use. Additionally, we have evaluated RedPoint’s perceived ease-of-use. For this assessment, we used the rubric in Table 7 (which is based on an ISO/IEC 9126-4 approach to usability metrics) and evaluated the RedPoint Data Management tool accordingly.
Measure | Result
--- | ---
**Efficiency** — Ease of installation, setup, and configuration  
• Using the vendor’s documentation, how much effort (in-person hours) was required to install and setup the software once the target instance(s) were available?  
• How much effort (in person-hours) was required to configure the necessary Hadoop components to get the jobs to execute?  
  The installation and setup of RedPoint Data Management Site and Execution Servers and Client tool took less than 1.5 person-hours. The configuration of Hadoop tools took less than 0.5 person-hours.

**Effectiveness** — Job execution completion rate  
• Once a data management/integration job is created and runs successfully on a test set of data, how many benchmark jobs failed to complete due to problems with the vendor software or Hadoop?  
  No failures. RedPoint Data Management successfully completed every benchmark test after we confirmed the job was properly formed by running a test data set.

**Satisfaction** — User Interface  
• On a scale from very difficult to very easy, how did we find our experience building the data integration/management jobs?  
  Very easy. The user interface is intuitive. Data integration/management components are clearly identified and configuration options were easy to set. We only referred to the documentation and in-tool help content (which was very thorough) to confirm our usage and settings of components. In our experience, most other vendor tools rate from easy to moderately difficult.

Table 7: RedPoint’s perceived usability tests

**Conclusion**

There are multiple ways to integrate data into Hadoop. There are vast differences in the architectures of the vendors, wrapping open source tools like MapReduce and Spark.

You cannot be satisfied with the functionality of a Hadoop load; you must also be concerned with performance. Ensure the wind is in your sails with your tool selection by leaving yourself room for experimentation, error, and growth. Performance will be there for the vast cycles of development, testing, quality assurance and, of course, production. Ultimately, the proof is in the testing outcomes. Our benchmark results were beyond what we thought possible.

Vendor architecture is important in integrating data with Hadoop, yet the differences are vast. RedPoint is based on a foundation of YARN, which has proven to be a good choice.
About MCG Global Services

William Mc Knight is President of McKnight Consulting Group (MCG) Global Services (http://www.mcknightcg.com). He is an internationally recognized authority in information management. His consulting work has included many of the Global 2000 and numerous midmarket companies, and his teams have won several best practice competitions for their implementations and many of his clients have gone public with their success stories. McKnight’s strategies form the information management plan for leading companies in various industries.

Jake Dolezal has over 17 years of experience in the Information Management field with expertise in business intelligence, analytics, data warehousing, statistics, data modeling and integration, data visualization, master data management, and data quality. Dolezal has experience across a broad array of industries, including: healthcare, education, government, manufacturing, engineering, hospitality, and gaming.

With an A-list of clients representing complex and highly-successful information management, MCG has broad catalogue of experience. Our advice is a combination of the latest best practices with our personal experience and expertise. It is practical, not theoretical.

- We take a keen focus on business justification.
- We take a programatic, not a project-based, approach.
- We believe in integrating with client staff and prioritize knowledge transfer.
- We engineer client workforces and processes to carry you forward.
- We’re vendor neutral so you can rest assured that our advice is completely client oriented.
- We know, define, judge, and promote best practices.
- We have encountered and overcome most conceivable information management challenges.
- We ensure business results are delivered early and often.

We anticipate our customer’s needs well into the future with our full lifecycle approach. Our focused, experienced teams generate efficient, economic, timely, and sustainable results for our clients.
About RedPoint Global

RedPoint Global offers a comprehensive set of world-class ETL, data quality, and data integration applications that operate in and across both traditional and Hadoop 2.0/YARN environments. The company also offers data-driven customer engagement solutions that help companies derive insights from customer behaviors and create consistent, relevant, and precise messaging across any and all channels. All RedPoint applications offer a unique visual user interface that eliminates the need for programming skills. This allows enterprises to utilize all data to achieve their strategic business goals. For more information, visit www.redpoint.net or email: contact.us@redpoint.net.