8K Miles
Cloud | Identity | Big Data | Mobile

BIG DATA SOLUTION BENCHMARK

Introduction

In the last few years, Big Data Analytics have gained a very fair amount of success. The trend is expected to grow rapidly with further advancement in the coming years. Today, there is a plethora of diversified Big Data solutions featuring new-age technologies. As new solutions evolve at a rapid pace, there is need for an objective method to compare the performance, scalability and cost of different solutions.

This paper addresses the objective by benchmarking leading Big Data solutions. In this benchmarking, we have handpicked some of leading Big Data solutions; Amazon Redshift, Google BigQuery, Microsoft Azure SQL Data Warehouse, Cloudera Impala, Presto, Hive and Spark. These contenders are evaluated, discussed and presented as a benchmarking report.

This paper discusses the benchmarking objectives, methodology, infrastructure, data sets, setting up procedures, and benchmark tests with partial results.

Benchmark Objectives

Leading Big Data solutions viz. Amazon Redshift, Google BigQuery, Microsoft Azure SQL Data Warehouse were chosen for benchmarking. In addition, Cloudera Impala, Presto, Hive and Spark were included for the tests.

Following were set as the objectives for benchmarking above products.

- Define a monthly cost for setting up infrastructure for the identified Big Data solutions.
- Arrive at a benchmark approach and varied type of tests. In this exercise, we execute tests like Power run, Concurrent run and Throughput run.
- Find query response time for queries described in the TPC-H benchmark specification with different dataset sizes.
- Find query response time for queries described in the TPC-H benchmark specification with concurrent threads.
- Find the maximum queries that can be executed for a given period of time.

Methodology

In this section, we walkthrough the pre-benchmark activities.

Benchmark Tests

Effective benchmarking process can provide accurate metrics measuring of Big Data systems. In this exercise, we are performing the following tests.

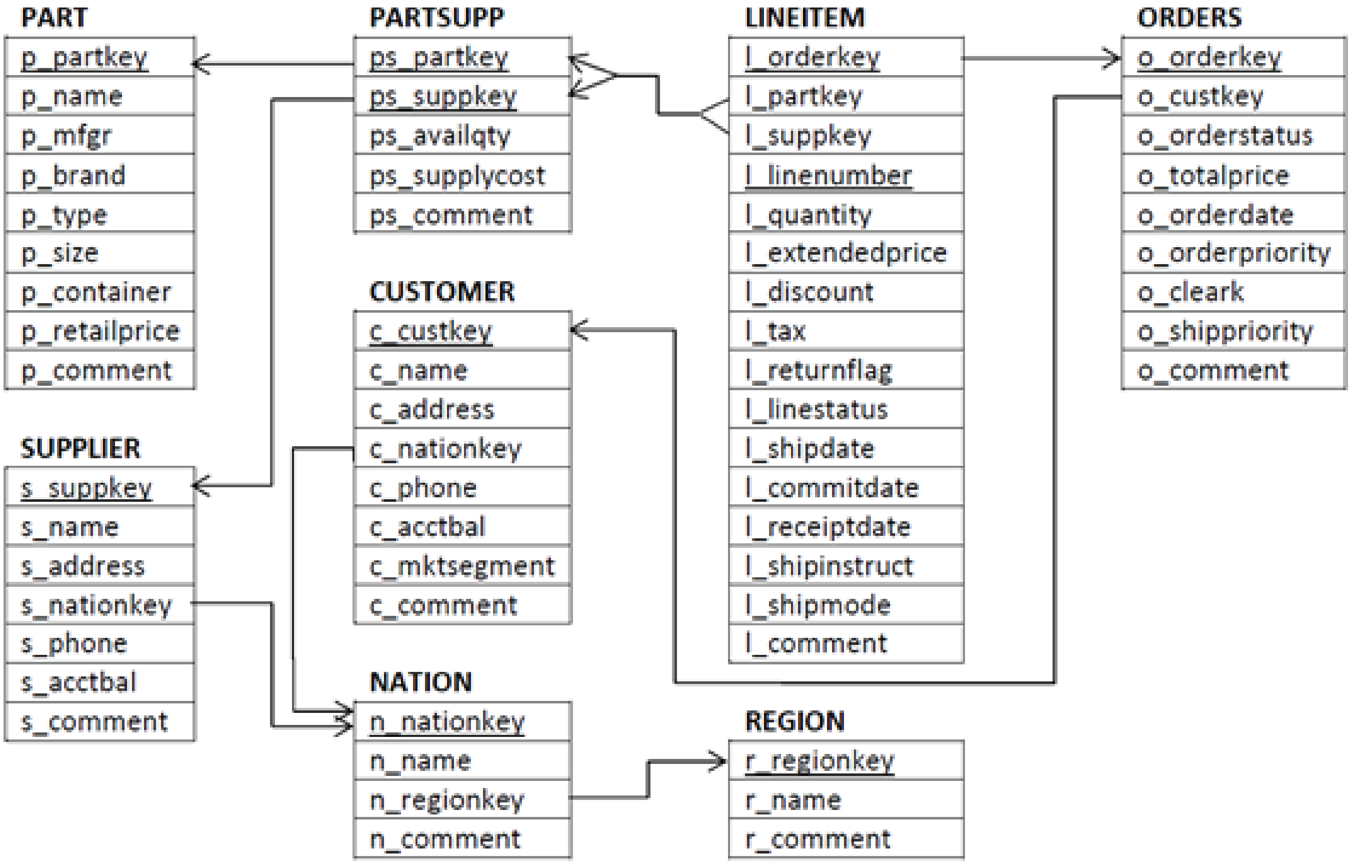
- Power Run: The power run aims at measuring the raw query execution power of the system with a single active session. This is achieved by sequentially running each and every identified query.
- Concurrency Run: The concurrency run is similar to power run but executes the queries with concurrent threads. The threads are increased till the performance of Big Data solution hits checkpoint.
- Throughput Run: The throughput run aims at measuring the ability of the system to process the most queries in the least amount of time, possibly taking advantage of I/O and CPU parallelism.

Along with above tests, we note down subjective experience, important observations and features present in each Big Data system.

Schema and Dataset

The benchmarking exercise adopts TPC-H standard of Transaction Processing Performance Council (TPC) for data schema, data generation and queries.

The standard defined schema consists of eight separate and individual tables. The relationships between columns of these tables are illustrated below.



Data Generation

The required data sets were generated using TPC-H DB-gen tool. The scale factors (SF) in DB-gen tool specifies how many GB of data will be generated. For example, for a SF of 100, 100GB of data will be generated. In our different tests, we benchmark 100 GB, 1 TB and 10 TB datasets. The below table depicts number of rows against each table and dataset.

Table	100 GB No. of Records	1 TB No. of Records	10 TB No. of Records
customer	15,000,000	150,000,000	1,499,999,439
orders	150,000,000	1,500,000,000	15,000,000,000
lineitem	600,037,902	5,999,989,709	59,999,994,267
nation	25	25	25
region	5	5	5
supplier	1,000,000	10,000,000	100,000,000
part	20,000,000	200,000,000	2,000,000,000
partsupp	80,000,000	800,000,000	8,000,000,000

Cloud Services and Infrastructure

Amazon Web Services (AWS), Google Compute Platform (GCP) and Microsoft Azure were chosen as Cloud hosting services. The primary contenders BigQuery, Redshift and Azure SQL Data Warehouse are Big Data services offered by GCP, AWS and Microsoft Azure respectively. In terms of Cloud computing features, all the above Cloud providers offer similar features, in our case high performance and scalable compute power for large datasets.

We then identify infrastructure capacity options (CPU, Memory, IO, Network bandwidth requirements) and supporting services required for this benchmarking.

#	Big Data	Cloud Hosting	Services Used
1	BigQuery	Google Compute Platform	BigQuery, Compute Engine, Cloud Storage
2	AWS Redshift	Amazon Web Services	AWS Redshift, EC2, S3
3	Impala	Google Compute Platform	Compute Engine, Cloud Storage
4	Presto	Google Compute Platform	Compute Engine, Cloud Storage
5	Spark	Google Compute Platform	Compute Engine, Cloud Storage
6	Hive	Google Compute Platform	Compute Engine, Cloud Storage
7	Azure SQL DWH	Microsoft Azure	Azure SQL Data Warehouse, Azure Storage - Blob

* The infrastructure details is explained in the Infrastructure specification section.

Setup, Configuration, Warm-up Tests

- Configure and setup individual environment for Amazon Redshift, Google BigQuery, Impala, Presto, Hive, Spark and Azure SQL Data Warehouse.
- The infrastructure cost setup for the environment will match the defined monthly cost. In this benchmarking exercise, the monthly infrastructure cost for each environment is 40K USD. In the case of Microsoft Azure, the pricing options did not match the defined monthly cost. Hence the tests were executed on both 3000 DWU & 6000 DWU which were priced about 27K and 54K per month respectively.
- Validate the infrastructure for internal & external connectivity post infrastructure setup.
- Identify or develop a benchmark client that would run assorted (Power run, Concurrent run, Throughput run) tests on Big Data solutions.
- Configure and setup benchmark client in the cloud hosting environment (AWS, Microsoft Azure and GCP).
- Run warm up tests using a small dataset on separate BigQuery, Redshift, Azure SQL Data Warehouse, Impala, Presto, Hive and Spark environments.
- Run multiple iterations with varied dataset size on the above configured environments. Document the response time and other metrics if available. Note down all the observations and search service behavior from a developer perspective.

Infrastructure Specification

This section summarizes the infrastructure setup for the chosen Big Data solutions.

100 GB, 1 TB, 10 TB datasets	
AWS Redshift	✓ Hosted : Amazon Web Services ✓ Region and zone : US West (Oregon) ✓ AWS Redshift dc1.8xlarge (32 vCPU, 244 RAM, 104 ECU, 2.56TB SSD IO 3.70GB/s) ✓ Number of nodes - 11 Nodes
Google BigQuery	
Azure SQL Data Warehouse	✓ Hosted : Microsoft Azure ✓ Region and zone : West Central US ✓ Azure SQL Data Warehouse - ✓ 3000 DWU (30 nodes, 2 databases per node, 6 GB memory) ✓ 6000 DWU (60 nodes, 1 database per node, 6 GB memory)
Cloudera Manager Impala/ Presto/Spark and Hive	✓ Hosting : Google Compute Platform ✓ Region and zone : Central US us-central1-b ✓ n1-highmem-32 (32 vCPUs, 208 GB memory) ✓ Boot disk 25 GB (SSD persistent disk), Additional Disk – 1.5TB, ✓ Number of nodes - 27 Nodes ✓ or ✓ n1- highmem-16 (16 vCPUs, 104 GB memory) ✓ Boot disk 25 GB (SSD persistent disk), Additional Disk – 1.5TB, ✓ Number of nodes - 53 Nodes ✓ Operating system : Centos 6.6

Please refer following sites for more information.
https://aws.amazon.com/ec2/instance-types/
https://cloud.google.com/compute/docs/machine-types
https://azure.microsoft.com/en-us/documentation/articles/virtual-machines-windows-sizes/

Instance Type Vs. Dataset

The choice of instance type (GCE, Microsoft Azure and AWS) is based on dataset size and type of benchmark test. The below table details the instance type and number of nodes chosen for each dataset and type of test.

#	Big Data	Dataset size	No. of Nodes / Instance type
1	Big Query	100 GB 1 TB	Benchmark client – 1 n1- highmem-16 (16 vCPUs, 104 GB memory) Infra Spec for BQ is abstracted for end users
2	Big Query	10 TB	Benchmark client – 1 n1- highmem-16 (16 vCPUs, 104 GB memory) Infra Spec for BQ is abstracted for end users
3	Redshift	100 GB 1 TB	Benchmark client – 1 r3.4xlarge (16 vCPU, 122 RAM, 52 ECU) 30 GB EBS GP2 SSD disk AWS Redshift - dc1.8xlarge (32 vCPU, 244 RAM, 104 ECU, 2.56TB SSD IO 3.70GB/s) – 11 Nodes
4	Redshift	10 TB	Benchmark client – 1 r3.8xlarge (32 vCPU, 244 RAM, 104 ECU) 30 GB EBS GP2 SSD disk AWS Redshift - dc1.8xlarge (32 vCPU, 244 RAM, 104 ECU, 2.56TB SSD IO 3.70GB/s) – 11 Nodes
5	Impala Presto Spark Hive	100 GB	Benchmark client – 1 n1- highmem-16 (16 vCPUs, 104 GB memory) Cloudera Setup - n1- highmem-16 (16 vCPUs, 104 GB memory) – 53 Nodes
6	Impala Presto Spark Hive	1 TB 10 TB	Benchmark client – 1 n1-highmem-32 (32 vCPUs, 208 GB memory) Cloudera Setup - n1-highmem-32 (32 vCPUs, 208 GB memory) – 27 Nodes
7	Azure SQL Data Ware- house	100 GB 1 TB 10 TB	Benchmark client - Standard DS14 v2 (16 cores, 112 GB memory) Local Disk: 224 GB (Local SSD) Operating system: Ubuntu Linux Azure SQL Data Warehouse – 3000 DWU & 6000 DWU

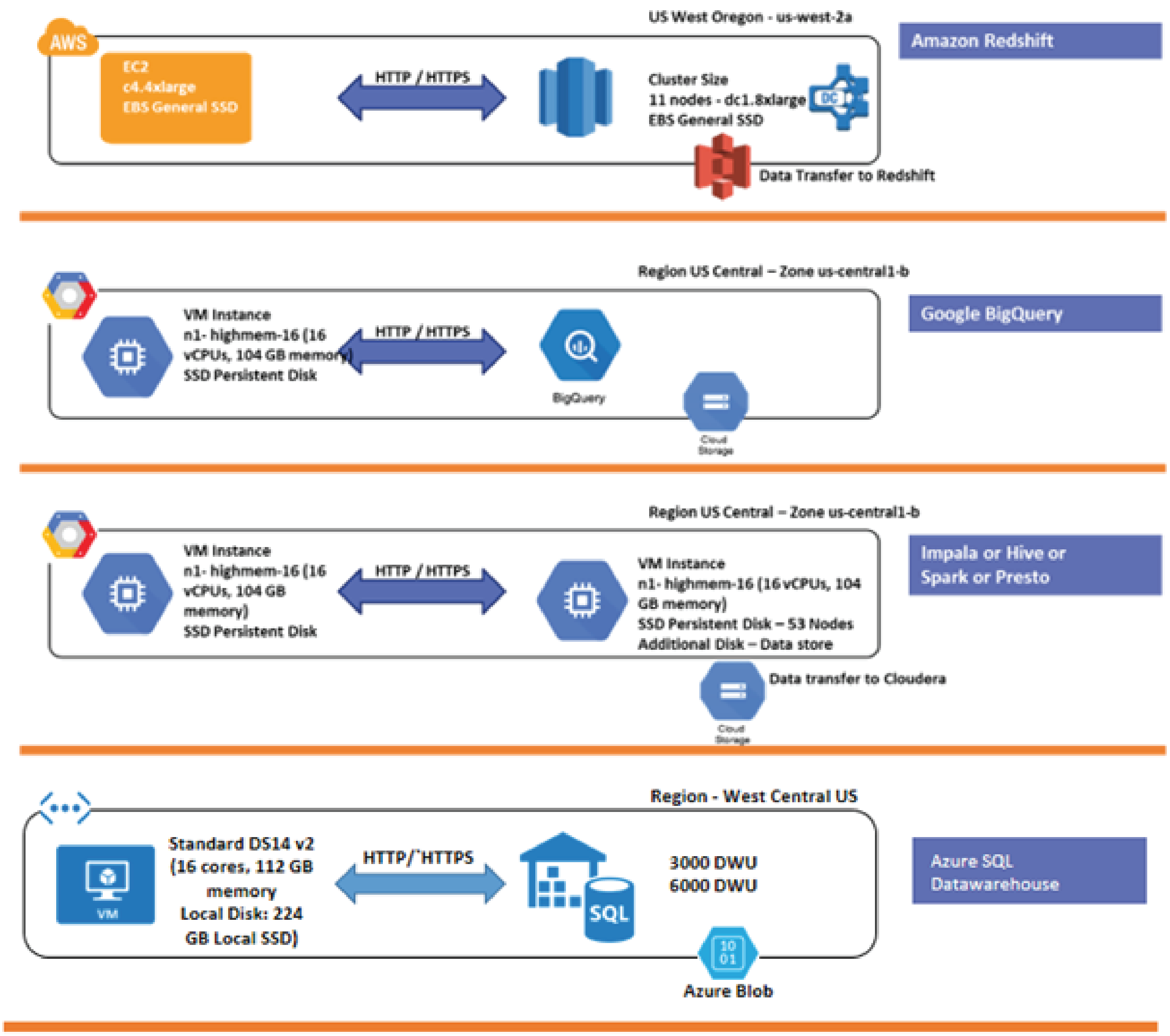
Cost

In this benchmarking exercise, the monthly infrastructure cost for each environment is fixed. One custom objective in this benchmarking exercise is to setup an environment for every Big Data solution that would cost the fixed amount per month.

However, for Azure SQL Data Warehouse, the cost could not be fixed and so test were executed for 3000 DWU & the 6000 DWU configurations.

Architecture Statement

1. The choice of instance type (GCE, Microsoft Azure and AWS) is based on dataset size and type of benchmark test. The below table details the instance type and number of nodes chosen for each dataset and type of test.


AWS
US West Oregon - us-west-2a
EC2
c4.4xlarge
EBS General SSD
HTTP / HTTPS
Cluster Size
11 nodes - dc1.8xlarge
EBS General SSD
DC
Data Transfer to Redshift
Amazon Redshift
Region US Central – Zone us-central1-b
VM Instance
n1- highmem-16 (16 vCPUs, 104 GB memory)
SSD Persistent Disk
HTTP / HTTPS
BigQuery
Cloud Storage
Google BigQuery
Region US Central – Zone us-central1-b
VM Instance
n1- highmem-16 (16 vCPUs, 104 GB memory)
SSD Persistent Disk
HTTP / HTTPS
VM Instance
n1- highmem-16 (16 vCPUs, 104 GB memory)
SSD Persistent Disk – 53 Nodes
Additional Disk – Data store
Data transfer to Cloudera
Cloud Storage
Impala or Hive or Spark or Presto
Region - West Central US
VM
Standard DS14 v2 (16 cores, 112 GB memory Local Disk: 224 GB Local SSD)
HTTP/'HTTPS
SQL
3000 DWU
6000 DWU
10
01
Azure Blob
Azure SQL Datawarehouse

SETUP OVERVIEW

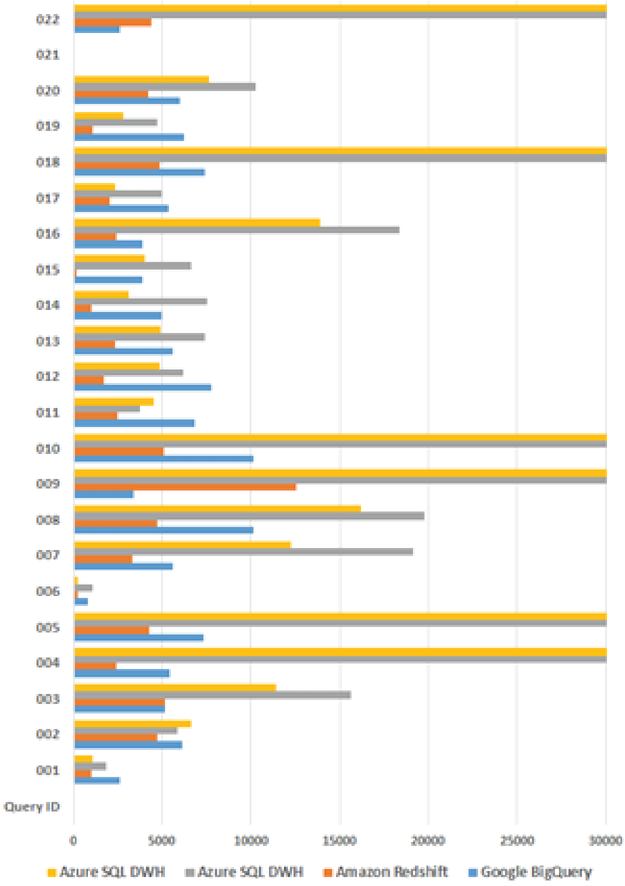
2. The Big Data solution, benchmark client infrastructure and dependent services are setup individually for the respective search provider at their hosting.
   Example 1: Amazon Redshift and its benchmark client are hosted at Amazon Web Services.
   Example 2: Cloudera Impala and its load client are hosted at Google Compute Platform.
   Example 3: Azure SQL Data Warehouse and its load client are hosted at Microsoft Azure platform.
3. The infrastructure setup of Big Data solution and benchmark client placed in the same region and same availability zone (based on feasibility). This is to ensure the regional and availability zone latencies are avoided.
   a. BigQuery: BigQuery dataset and its tables are configured in US region whereas the benchmark client is setup at region - US Central us-central1-f.
   b. Redshift: Amazon Redshift nodes and the benchmark client are setup in the same region – US West Oregon us-west-2a.
   c. Azure SQL Data Warehouse: Azure SQL Data Warehouse nodes and the benchmark client are setup in the same region – West Central US.
   d. Cloudera: Impala, Presto, Spark and Hive are setup -using Cloudera Manager (CM). The Cloudera manger is configured to use either US Central us-central1-f or US Central us-central1-b or US Central us-central1-c. During the installation the CM will ensure the nodes are spawned within same region and availability zone. To have speedy process and hassle free, we had setup 4 Cloudera setup in 4 availability zones that would cater Impala, Presto, Spark and Hive respectively. The benchmark client for each individual environment is setup at its respective availability zone.
4. For storage, boot volume and additional disks are SSD based. For Microsoft Azure & GCP, disk type is SSD persistent disk and for AWS it is General purpose SSD (GP2).
5. The test runs executed by the benchmark client are primarily heavy query scripts. So, it is ideal to have the benchmark client powered with good amount of compute and memory to execute these test runs in a multithreaded model. Hence the infrastructure specification for the benchmark client is chosen with high CPU and memory irrespective of the cloud providers (Microsoft Azure, AWS and GCP).

Benchmark Results

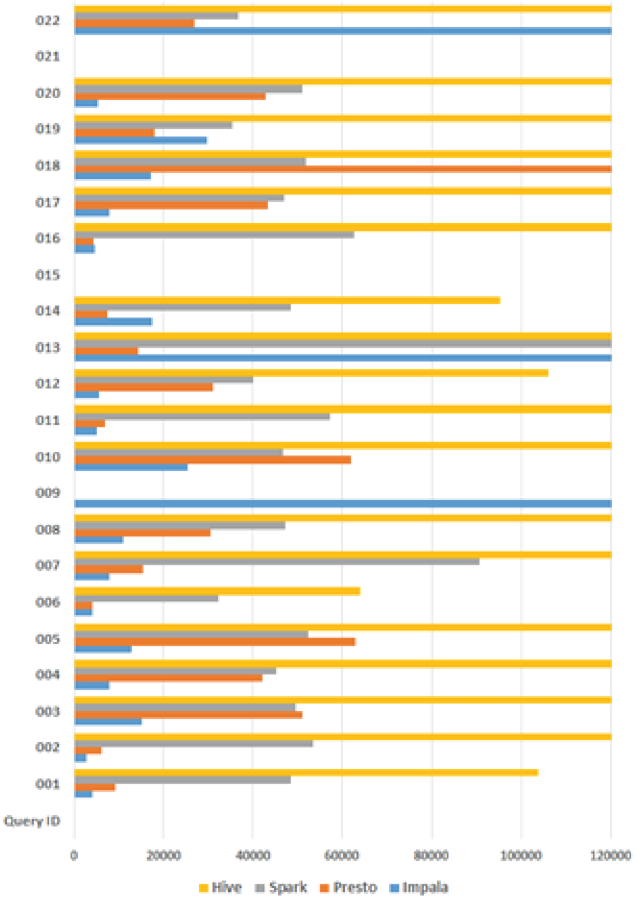
A partial list of results from different benchmarking tests are presented here.

Power Run – 100 Gb

	Query Response Time (milli seconds)			
	Google BigQuery	Amazon Redshift	Azure SQL DWH	Azure SQL DWH
Query ID	- Default Options	- Default Options - 11 nodes	- Default Options - 3000 DWU - smallrc instance	- Default Options - 6000 DWU - smallrc instance
001	2,641	1,023	1,859	1,102
002	6,148	4,737	5,839	6,655
003	5,144	5,176	15,629	11,399
004	5,410	2,393	169,176	140,479
005	7,324	4,278	35,567	39,416
006	803	247	1,080	232
007	5,637	3,322	19,152	12,235
008	10,131	4,749	19,768	16,235
009	3,353	12,588	110,979	80,534
010	10,168	5,093	91,142	132,404
011	6,820	2,474	3,732	4,552
012	7,805	1,697	6,180	4,814
013	5,641	2,356	7,426	4,888
014	4,974	1,038	7,541	3,113
015	3,861	175	6,622	4,024
016	3,913	2,399	18,353	13,886
017	5,353	2,062	4,945	2,372
018	7,388	4,853	53,191	35,027
019	6,271	1,111	4,734	2,807
020	5,985	4,218	10,293	7,653
021				
022	2,596	4,391	60,322	48,338
Geometric Mean	4,986	2,341	13,460	9,532

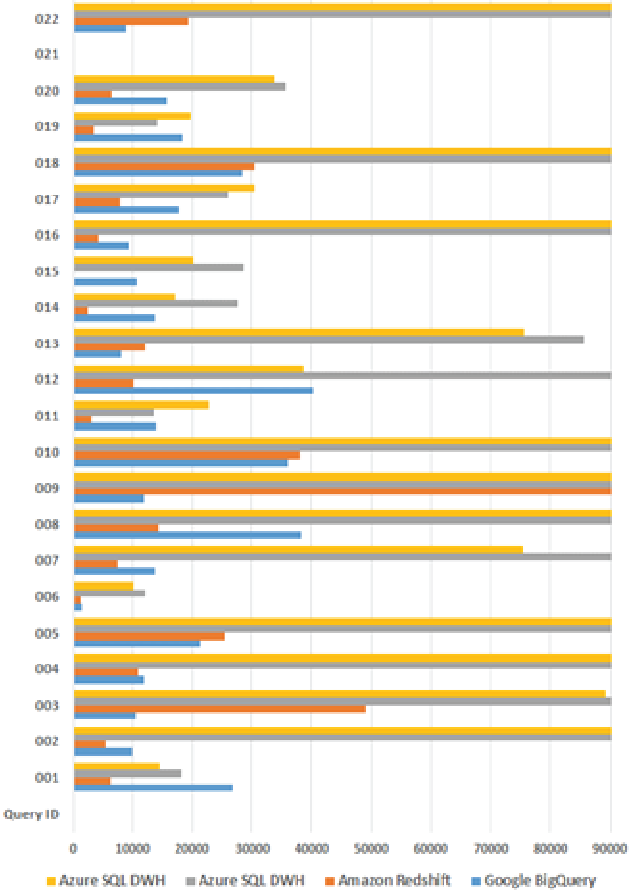


	Query Response Time (milli seconds)			
	Impala	Presto	Spark	Hive
Query ID	- Default Options	- Default Options	- Default Options	- Default Options
001	4,214	9,353	48,488	103,882
002	2,869	6,254	53,371	294,384
003	15,099	51,030	49,482	208,096
004	7,912	42,168	45,214	189,256
005	12,898	63,005	52,338	303,489
006	4,165	4,022	32,482	63,882
007	7,855	15,358	90,755	211,194
008	11,064	30,611	47,292	315,346
009	628,921			
010	25,318	61,955	46,862	198,465
011	5,194	6,897	57,356	188,927
012	5,731	30,957	39,995	106,070
013	147,786	14,485	172,194	381,333
014	17,436	7,360	48,663	95,184
015				
016	4,581	4,222	62,745	179,056
017	8,004	43,452	46,899	237,378
018	17,118	121,826	51,895	274,827
019	29,879	17,968	35,464	138,925
020	5,388	42,887	51,207	217,901
021				
022	120,554	26,843	36,714	137,710
Geometric Mean	14,128	20,667	51,900	184,159

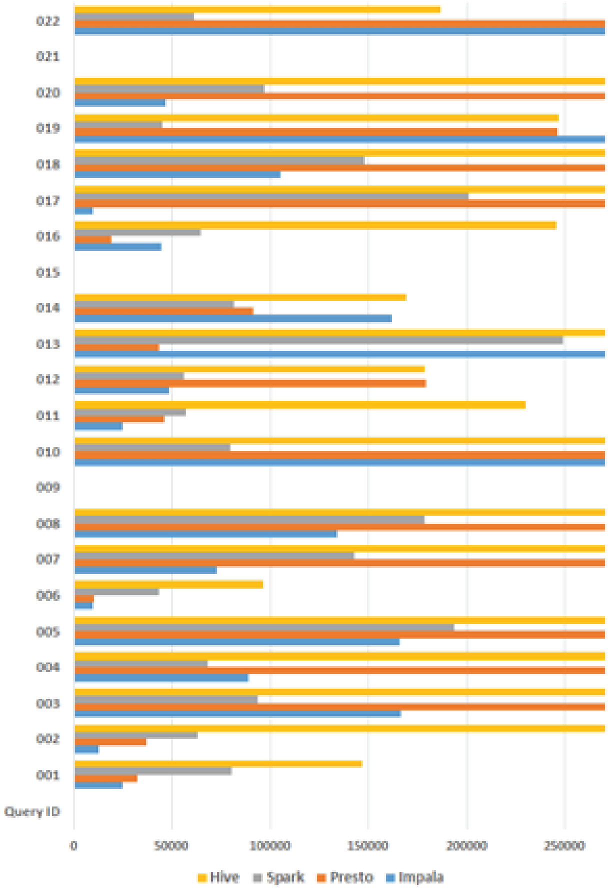


Power Run – 1 TB

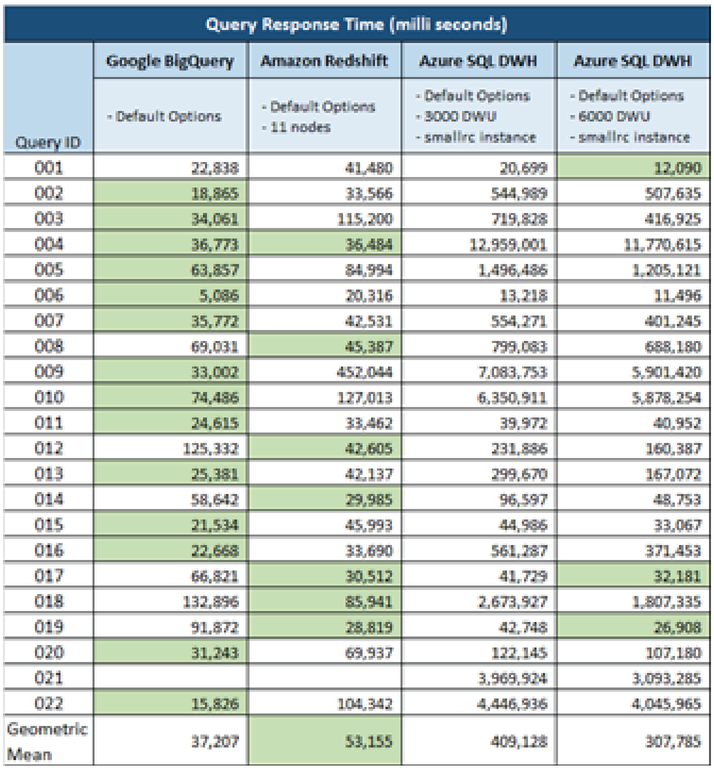
Query Response Time (milli seconds)				
	Google BigQuery	Amazon Redshift	Azure SQL DWH	Azure SQL DWH
Query ID	- Default Options	- Default Options - 11 nodes	- Default Options - 3000 DWU - smallrc instance	- Default Options - 6000 DWU - smallrc instance
001	26,788	6,291	18,260	14,551
002	9,918	5,538	102,672	90,839
003	10,513	48,911	243,674	89,297
004	11,871	10,988	1,585,263	1,491,596
005	21,164	25,464	258,005	233,518
006	1,509	1,336	12,130	10,204
007	13,789	7,434	107,726	75,395
008	38,282	14,329	116,840	97,381
009	11,790	149,844	1,242,229	1,193,395
010	35,917	38,125	809,735	773,836
011	13,870	3,137	13,637	22,851
012	40,228	10,046	173,099	38,622
013	7,989	11,957	85,593	75,677
014	13,744	2,489	27,540	17,043
015	10,691	162	28,533	20,117
016	9,276	4,200	125,048	91,248
017	17,815	7,838	26,094	30,399
018	28,267	30,449	404,178	310,886
019	18,293	3,410	14,152	19,779
020	15,709	6,551	35,595	33,741
021				
022	8,829	19,280	511,657	458,530
Geometric Mean	14,303	8,396	102,104	82,407



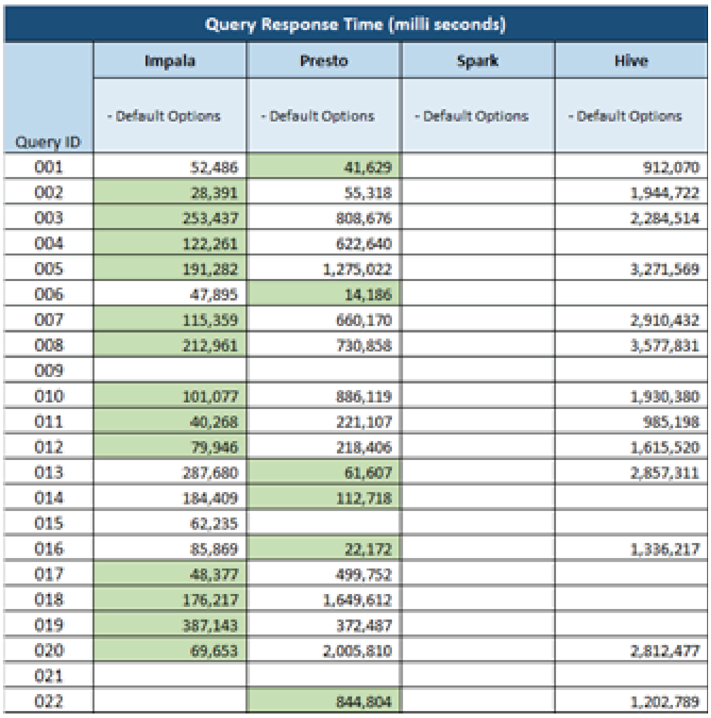
Query Response Time (milli seconds)				
	Impala	Presto	Spark	Hive
Query ID	- Default Options	- Default Options	- Default Options	- Default Options
001	24,753	32,081	80,388	146,823
002	12,747	36,868	63,262	345,163
003	166,633	375,877	93,641	310,257
004	89,020	357,883	68,323	365,016
005	166,042	966,321	193,887	497,449
006	9,662	10,617	43,558	96,363
007	72,895	411,017	142,679	534,125
008	134,255	526,085	178,868	526,076
009				
010	360,845	357,391	80,024	306,393
011	24,724	46,395	57,081	230,013
012	48,359	179,087	56,318	178,773
013	3,038,655	43,390	249,131	573,214
014	161,609	91,038	81,451	169,129
015				
016	44,202	19,117	64,589	245,718
017	9,544	342,892	201,189	423,872
018	105,074	1,276,116	148,133	562,754
019	473,570	246,312	44,853	246,633
020	46,935	770,261	96,974	363,435
021				
022	1,491,853	377,760	61,338	186,810
Geometric Mean	92,532	170,336	91,579	296,445



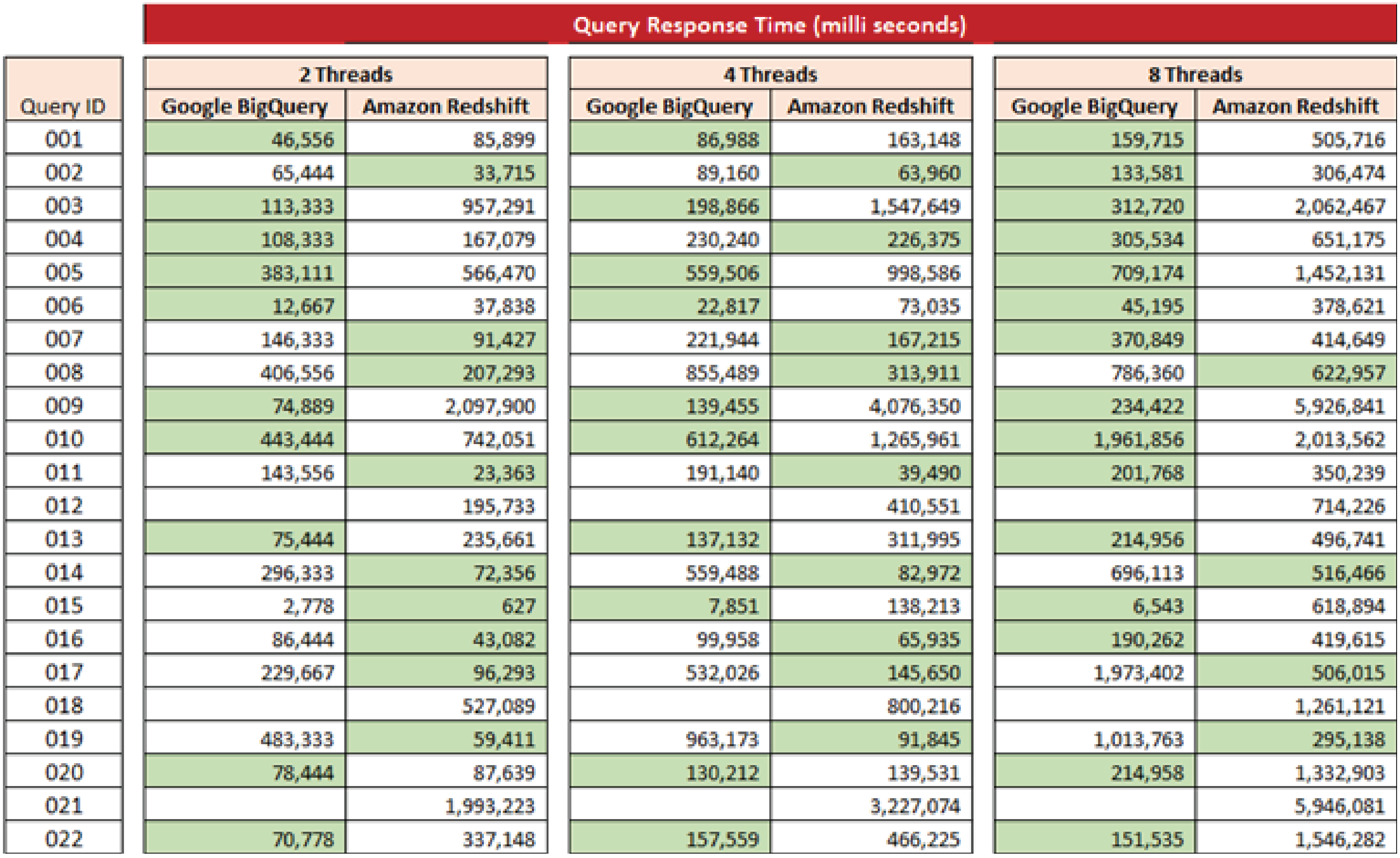
Concurrent Run – 10 Tb - 2/4/8 Threads



Query ID	Query Response Time (milli seconds)			
	Google BigQuery	Amazon Redshift	Azure SQL DWH	Azure SQL DWH
	- Default Options	- Default Options - 11 nodes	- Default Options - 3000 DWU - smallrc instance	- Default Options - 6000 DWU - smallrc instance
001	22,838	41,480	20,699	12,090
002	18,865	33,566	544,989	507,635
003	34,061	115,200	719,828	416,925
004	36,773	36,484	12,959,001	11,770,615
005	63,857	84,994	1,496,486	1,205,121
006	5,086	20,316	13,218	11,496
007	35,772	42,531	554,271	401,245
008	69,031	45,387	799,083	688,180
009	33,002	452,044	7,083,753	5,901,420
010	74,486	127,013	6,350,911	5,878,254
011	24,615	33,462	39,972	40,952
012	125,332	42,605	231,886	160,387
013	25,381	42,137	299,670	167,072
014	58,642	29,985	96,597	48,753
015	21,534	45,993	44,986	33,067
016	22,668	33,690	561,287	371,453
017	66,821	30,512	41,729	32,181
018	132,896	85,941	2,673,927	1,807,335
019	91,872	28,819	42,748	26,908
020	31,243	69,937	122,145	107,180
021			3,969,924	3,093,285
022	15,826	104,342	4,446,936	4,045,965
Geometric Mean	37,207	53,155	409,128	307,785

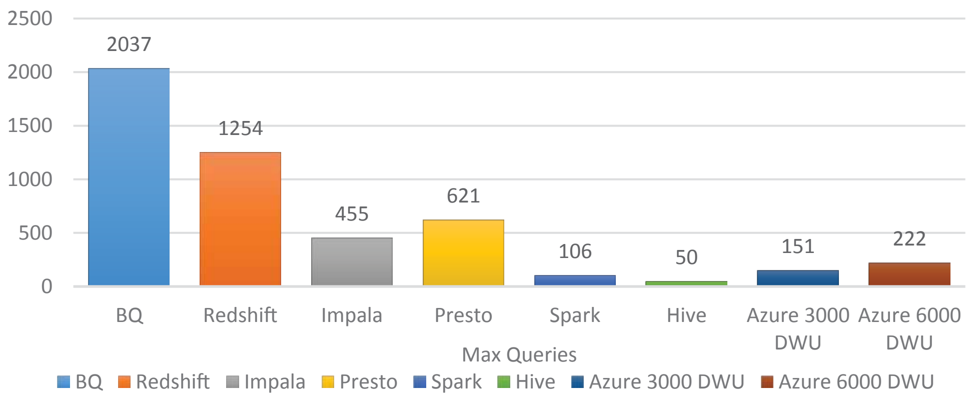


Query ID	Query Response Time (milli seconds)			
	Impala	Presto	Spark	Hive
	- Default Options	- Default Options	- Default Options	- Default Options
001	52,486	41,629		912,070
002	28,391	55,318		1,944,722
003	253,437	808,676		2,284,514
004	122,261	622,640		
005	191,282	1,275,022		3,271,569
006	47,895	14,186		
007	115,359	660,170		2,910,432
008	212,961	730,858		3,577,831
009				
010	101,077	886,119		1,930,380
011	40,268	221,107		985,198
012	79,946	218,406		1,615,520
013	287,680	61,607		2,857,311
014	184,409	112,718		
015	62,235			
016	85,869	22,172		1,336,217
017	48,377	499,752		
018	176,217	1,649,612		
019	387,143	372,487		
020	69,653	2,005,810		2,812,477
021				
022		844,804		1,202,789



	Query Response Time (milli seconds)					
	2 Threads		4 Threads		8 Threads	
Query ID	Google BigQuery	Amazon Redshift	Google BigQuery	Amazon Redshift	Google BigQuery	Amazon Redshift
001	46,556	85,899	86,988	163,148	159,715	505,716
002	65,444	33,715	89,160	63,960	133,581	306,474
003	113,333	957,291	198,866	1,547,649	312,720	2,062,467
004	108,333	167,079	230,240	226,375	305,534	651,175
005	383,111	566,470	559,506	998,586	709,174	1,452,131
006	12,667	37,838	22,817	73,035	45,195	378,621
007	146,333	91,427	221,944	167,215	370,849	414,649
008	406,556	207,293	855,489	313,911	786,360	622,957
009	74,889	2,097,900	139,455	4,076,350	234,422	5,926,841
010	443,444	742,051	612,264	1,265,961	1,961,856	2,013,562
011	143,556	23,363	191,140	39,490	201,768	350,239
012		195,733		410,551		714,226
013	75,444	235,661	137,132	311,995	214,956	496,741
014	296,333	72,356	559,488	82,972	696,113	516,466
015	2,778	627	7,851	138,213	6,543	618,894
016	86,444	43,082	99,958	65,935	190,262	419,615
017	229,667	96,293	532,026	145,650	1,973,402	506,015
018		527,089		800,216		1,261,121
019	483,333	59,411	963,173	91,845	1,013,763	295,138
020	78,444	87,639	130,212	139,531	214,958	1,332,903
021		1,993,223		3,227,074		5,946,081
022	70,778	337,148	157,559	466,225	151,535	1,546,282

Throughput Run – 100 GB



Key Findings

1. For the given benchmark queries, tests cases and dataset, Hive is slowest performance followed by Spark.
2. Presto and Impala had its equal share in Power run and Concurrent test runs. However Impala was better on throughput run.
3. On larger dataset 10 TB and 1 TB to some extent, Hive and Spark did not perform well. In some test runs, too many memory errors diluted the entire test run which lead to ignore the entire run.
4. The 10 TB Concurrent run was successful only for BigQuery and Redshift. Rest other contenders either had too many errors or took too much time to complete the test. Azure SQL Data Warehouse, Impala, Presto, Spark and Hive are ignored in Concurrent run due to failures.
5. For 10 TB tests, a single power run was not successful for Azure SQL Data Warehouse. Microsoft suggested to update the statistics on the tables. For queries which could be split, Microsoft suggested to split the queries and execute manually.
6. The primary contenders BigQuery and Redshift had some hard actions between them, however on a large dataset (10 TB), BigQuery fared better in terms of number of queries performance. In throughput test, BigQuery was ahead on Redshift and other contenders.
7. The order in which developer had good experience with the solutions is given below.

Developer experience of the Big Data solutions	
Ease of usability and Management console	BigQuery Redshift, Azure SQL Data Warehouse Cloudera (Impala, Presto, Spark, Hive)
Customization	Cloudera (Impala, Presto, Spark, Hive) Redshift, Azure SQL Data Warehouse BigQuery
Documentation	Cloudera (Impala, Presto, Spark, Hive) Redshift BigQuery, Azure SQL Data Warehouse
Monitoring	Cloudera (Impala, Presto, Spark, Hive) Redshift, Azure SQL Data Warehouse BigQuery
API access	Redshift, BigQuery, Azure SQL Data Warehouse Cloudera (Impala, Presto, Spark, Hive)
Data Ingestion Reliability	BigQuery, Redshift, Azure SQL Data Warehouse Cloudera (Impala, Presto, Spark, Hive)

REFERENCES

1. TPC
   a. http://www.tpc.org/
   b. https://github.com/electrum/tpch-dbgen
2. Amazon Redshift
   a. https://aws.amazon.com/redshift/
   b. https://aws.amazon.com/redshift/pricing/
3. Google BigQuery
   a. https://developers.google.com/bigquery/
   b. https://cloud.google.com/compute/pricing
4. Cloudera
   a. http://www.cloudera.com/documentation/other/reference-architecture/PDF/cloudera_ref_arch_gcp.pdf
5. Azure SQL Data Warehouse
   a. https://azure.microsoft.com/en-in/documentation/services/sql-data-warehouse/
   b. https://azure.microsoft.com/en-us/pricing/details/sql-data-warehouse/
   c. https://azure.microsoft.com/en-in/documentation/articles/sql-data-warehouse-tables-statistics/