BDAS: Comparison of Big Data Analytics Systems: Redshift & Hive

Prashant Ratnaparkhi, Insight Data Engineering Fellowship, New York

Hive setup: 2 AWS clusters with:

r3.Large (2vCPU, 6.5 ECU, 15 GiB memory) 6 instances

1 Name & 5 Data nodes

50 GB

r3.Large (2vCPU, 6.5 ECU, 15 GiB memory) 6 instances

1 Name & 5 Data nodes

500 GB

Redshift setup: One Amazon Redshift cluster with

dc1.Large (2vCPU, 7 ECU, 15 GiB memory) 6 instances

1 Leader & 5 Compute nodes

50 GB

500 GB

Schema and table size summary

- Schema based on TPC Big data Benchmark (TPC-BB)
- 23 tables representing a data warehouse of a large retailer

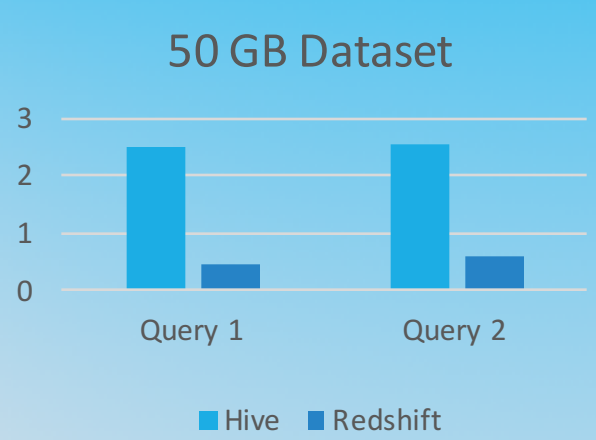
Table	# Records in millions [Total dataset 50 GB]	# Records in millions [Total dataset 500 GB]
customer	0.70	2.2
item	0.12	0.40
web_sales	50.1	595.6
store_sales	50.1	595.6
web_clickstreams	512.9	6035.5
…		

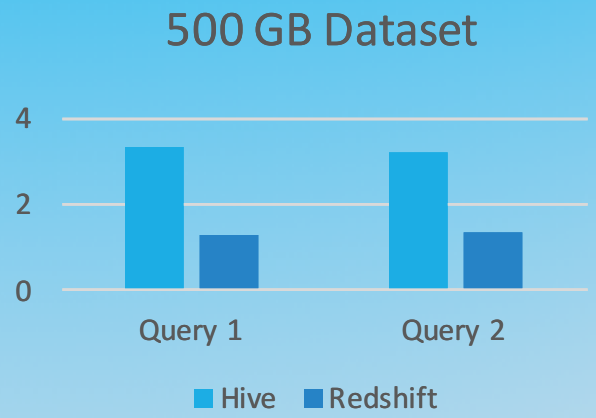
Two Sample Queries

#	Query	Operations
1	Customer-purchase behavior: List customers, who view online, followed by an in-store purchase 	• JOIN(web_clickstreams, item) → Web • JOIN(store, items)-> Store • JOIN(Web, Store) • ORDER BY • DISTINCT
2	Market basket analysis: List the items sold frequently together, in certain stores, categories 	• JOIN (store_sales, store_sales) -> Pairs • JOIN (Pairs, item) • COUNT • GROUP BY, HAVING • ORDER BY

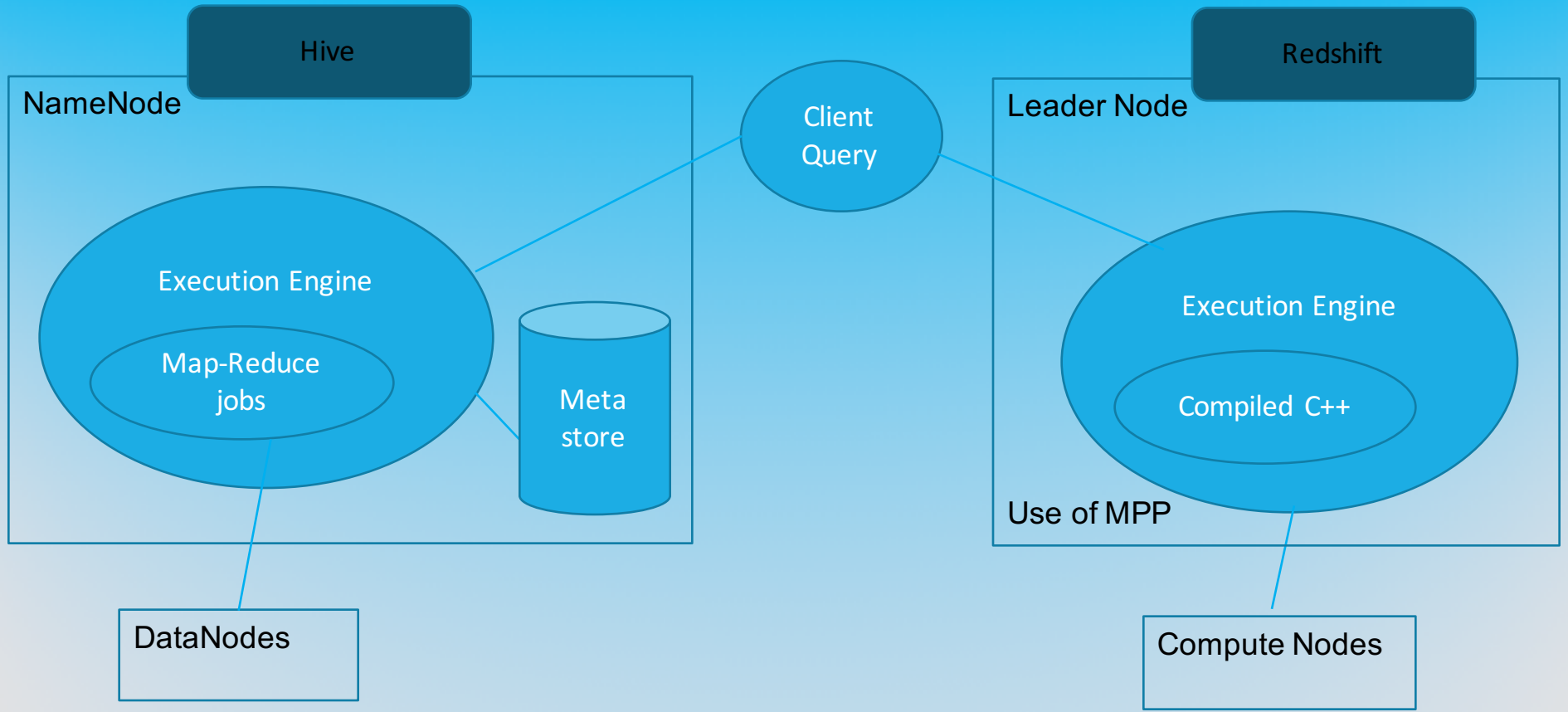
Summary of Results of selected queries

Execution time in seconds on log scale, for out-of-box setup:





Query Processing in a nutshell

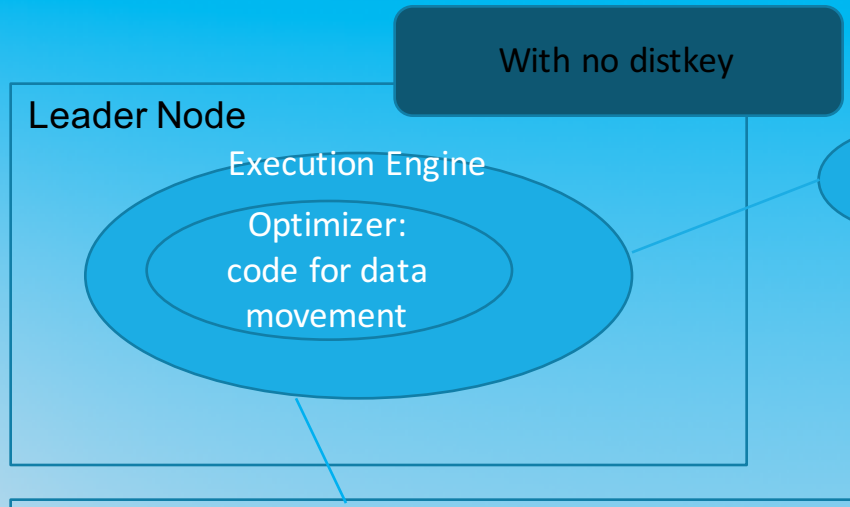


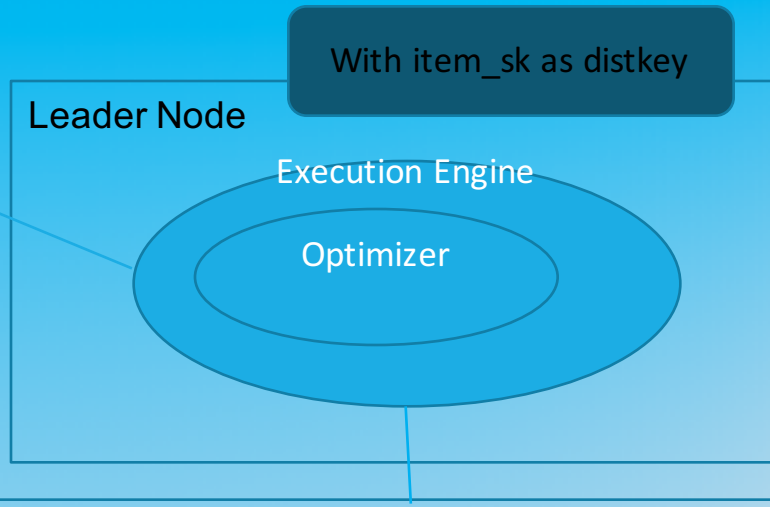
Redshift Performance Improvement

- 'Explain 'sql-statement' – gives query plan
- Look for DS_BCAST_*, DS_DIST_* steps, to identify data movement
- Use distkey while table creation, to indicate data co-location [Only one distkey per table allowed]

Table store_sales	Table Item	Table web_clickstreams
ss_item_sk (distkey)	i_item_sk (distkey, sortkey)	wcs_item_sk (distkey)
ss_customer_sk	i_item_id	wcs_sales_sk
ss_ticket_number (sortkey)	i_brand	wcs_click_date_sk (sortkey)
….	….	…

Redshift distkey illustration





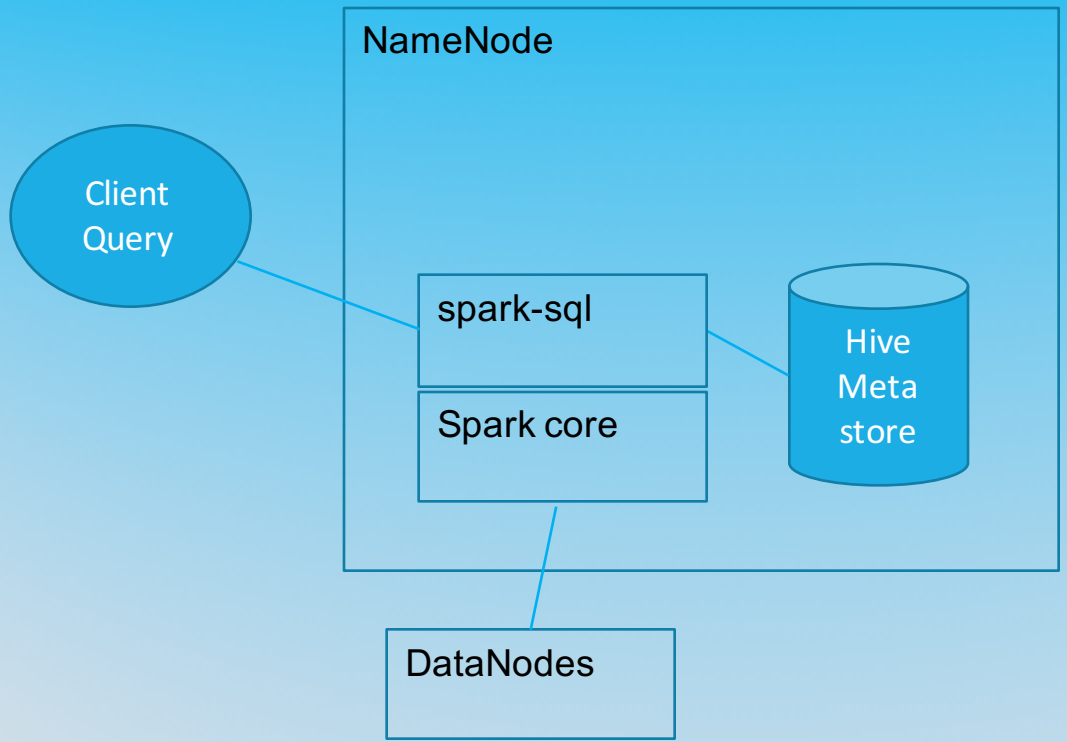
Compute Nodes: Data distribution EVEN (default), round robin during load

At run time:
Data movement on join of store_sales, item
Data movement on join of web_clickstreams, item

Compute Nodes: Data distribution KEY

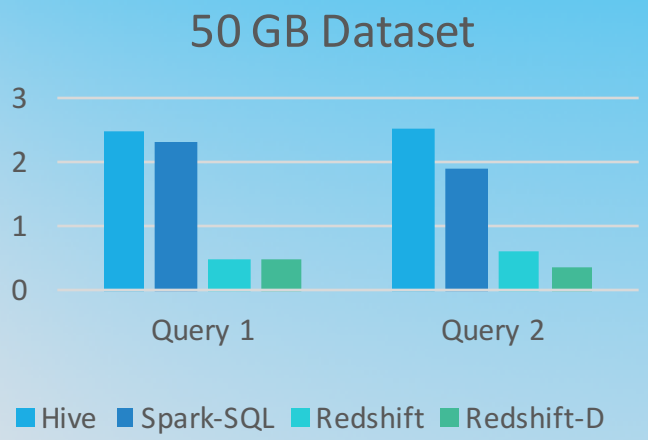
{store_sales.ss_item_sk=item.i_item_sk}
{web_clickstreams.wss_item_sk=item.i_item_sk}
Co-located together during load
At run time:
No data movement on join of store_sales, item
No data movement on join of web_clickstreams, item

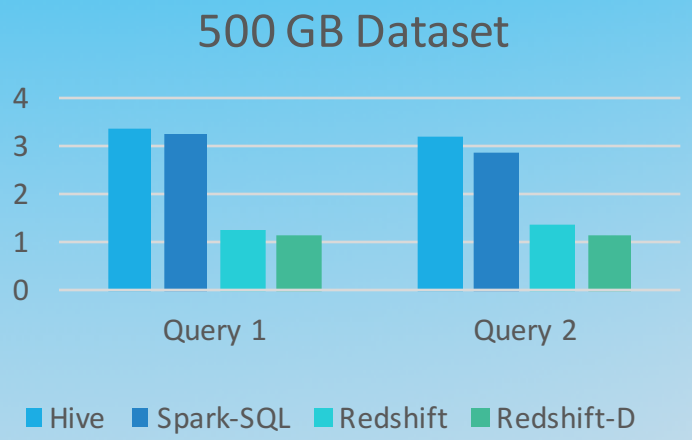
Hive Performance Improvement: spark-sql



Summary of Results of selected queries

Execution time in seconds on log scale, for out-of-box setup & after improvements:





Cost of this benchmark execution

#	Item	Duration in hours	Price node per hour in USD	Cost in USD
1	Redshift cluster: 6 dc1.large nodes (1 leader, 5 compute)	408	$0.250	$102.00
2	Hive Cluster: 12 m3.large nodes (2 master, 6 data)	384	$0.133	$51.07
3	Hive Cluster: 12 r3.large nodes (2 master, 6 data)	96	$0.166	$15.94
Total cost of Hive clusters				$67.01
Total cost of Redshift cluster				$102.00
Grand Total				$169.01

Redshift & Hive Use cases

If…	Then use	Examples
Requirements: • ease of use • high performance out-of-box • scalability in TB range good price-performance • low maintenance overhead	Redshift	• Companies with SaaS offerings in Analytics space • Most startups
Requirements: • replacement of traditional DW platforms • replacement of existing infrastructure with cloud • good price-performance in TB range	Redshift	• Large enterprises – Finra, NTT Docomo
Requirements: • massive amount of data – in PB range • customization/flexibility • no vendor lock-in • in-house data due to regulatory reasons	Hive	• Facebook • Sears • companies with existing big data infrastructure, staff

Prashant Ratnaparkhi: Technologist with experience in software product development

- Masters (Systems & Control Engineering), IIT, Mumbai
- Data Scientist Certification from Johns Hopkins/Coursera
- Enjoy running, skiing, swimming, biking, sudoku & learning

Appendix: Supporting slides

Summary of Results of selected queries

Task execution time, for out-of-the-box setup, is listed in the table below

Task	Hive (seconds)	RedShift (seconds)	Dataset (GB)
Query1	301	3.00	50 GB
Query1	2215	19.18	500 GB
Query2	346	4.15	50 GB
Query2	1639	22.52	500 GB

Results of selected queries after tuning

Task execution time is listed in the table below

Task	Hive (seconds)	RedShift (seconds)	Dataset (GB)	Spark-sql (seconds)	Redshift (with distkey, sortkey) seconds
Query1	301	3.00	50 GB	203 (down 33%)	3.00 (no change)
Query1	2215	19.18	500 GB	1901 (down 14%)	14.86 (down 22%)
Query2	346	4.15	50 GB	79 (down 77%)	2.25 (down 46%)
Query2	1639	22.52	500 GB	729 (down 56%)	13.95 (down 38%)

Summary of Results

Task execution time in seconds, for out-of-the-box setup, is listed in the table below (except for 'Load Data')

Task	Hive	RedShift	Dataset
Load Data*	34 Min: 44 Sec	41 Min: 35 Sec **	50 GB
Load Data*	3 Hr: 33 Min: 14 Sec	8Hr:25 Min **	500 GB
Query1	691	4.69	50 GB
Query1	4701	4.71	500 GB
Query2	301	3.00	50 GB
Query2	2215	19.18	500 GB
Query3	659	9.01	50 GB
Query3	4285	14.52	500 GB
Query4	612	3.66	50 GB
Query4	4563	8.12	500 GB
Query5	346	4.15	50 GB
Query5	1639	22.52	500 GB

*Load Data means populating metastore for Hive & copying data from S3 in case of RedShift.

** With three tables (store_sales, item, web_clickstream using distkey & sortkey, the 'Load Data' time increased to approx. 50 Min & 11 Hours respectively for Redshift.

Query 2

- Q2 -- Find all customers who viewed items of a given category on the web-- in a given month and year that was followed by an in-store purchase of an item from the same category in the three-- consecutive months.
- SELECT DISTINCT wcs_user_sk -- Find all customers
- FROM (-- web_clicks viewed items in date range with items from specified categories
- SELECT wcs_user_sk, wcs_click_date_sk
- FROM web_clickstreams, item
- WHERE wcs_click_date_sk BETWEEN 37134 AND (37134 + 30) -- in a given month and year
- AND i_category IN ('Books', 'Electronics') -- filter given category
- AND wcs_item_sk = i_item_sk AND wcs_user_sk IS NOT NULL AND wcs_sales_sk IS NULL --only views, not purchases
-) webInRange,(-- store sales in date range with items from specified categories
- SELECT ss_customer_sk, ss_sold_date_sk
- FROM store_sales, item
- WHERE ss_sold_date_sk BETWEEN 37134 AND (37134 + 90) -- in the three consecutive months.
- AND i_category IN ('Books', 'Electronics') -- filter given category
- AND ss_item_sk = i_item_sk AND ss_customer_sk IS NOT NULL
-) storeInRange -- join web and store
- WHERE wcs_user_sk = ss_customer_sk AND wcs_click_date_sk < ss_sold_date_sk -- buy AFTER viewed on website
- ORDER BY wcs_user_sk;

Query 5

```
-- Q5 Items sold together (frequently i.e. more that 50 times), in certain store,-- with specified categories & limit to 100.
select pairs.itm1, pairs.itm2, count(*) as cnt
from
 (
  select soldTogether.itm1, soldTogether.itm2
  from
   (
    select ss1.ss_item_sk as itm1, ss2.ss_item_sk as itm2
    from store_sales ss1, store_sales ss2
    where (ss1.ss_ticket_number = ss2.ss_ticket_number and ss1.ss_item_sk < ss2.ss_item_sk
            AND ss1.ss_store_sk IN (10, 20, 33, 40, 50))
   ) as soldTogether, item i
where soldTogether.itm1 = i.i_item_sk AND i.i_category_id IN (1, 2, 3)) as pairs
group by pairs.itm1, pairs.itm2
having cnt > 50
order by cnt desc, pairs.itm1, pairs.itm2
limit 100;
```

Description of Schema & Queries

- Schema & Queries are based on TPC Bigdata Benchmark (TPC-BB)
- Schema consists of 23 tables representing a data warehouse of a large retailer with multiple stores and large online sales presence, with both structured & semi-unstructured data such as web_clickstreams
- Find all customers who viewed items of a given category on the web-- in a given month and year that was followed by an in-store purchase of an item from the same category in the three -- consecutive months (Q2).
- List items sold together (frequently i.e. more that 50 times), in certain stores (with specified categories) & limit to 100 (Q5)

Redshift & Hive Features comparison

- Coming up