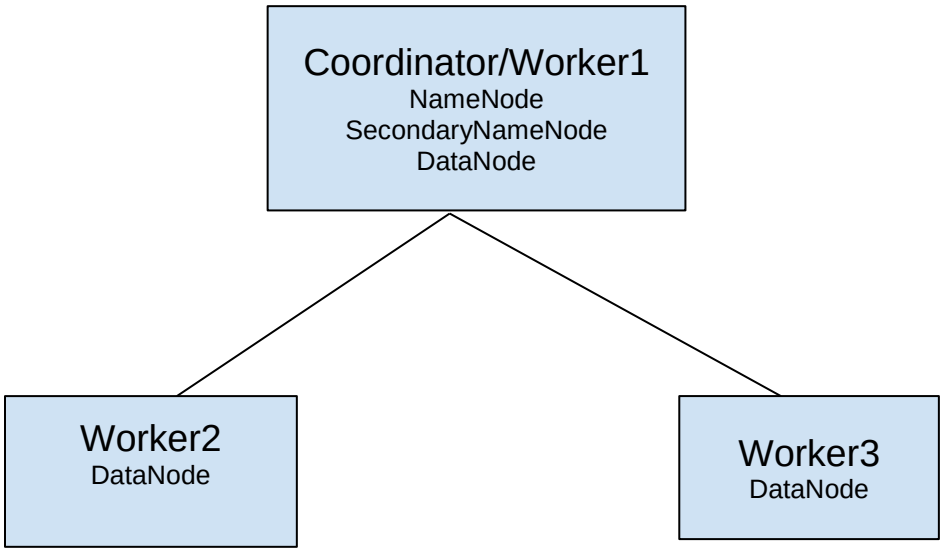
Background

This document explains the TPCH report of CarbonData (1.5.2 version) and ORC on Presto 2.10 execution engine.

Hardware

CPU: Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz - 48 CPU
Memory: 378 GB DDR4 RAM
Hard Disk: 11 x 4 TB SATA 7200 RPM HDD



Configurations

Carbon Properties

```
connector.name=carbondata
enable.unsafe.in.query.processing=false
enable.unsafe.sort=false
enable.unsafe.columnpage=false
carbon.unsafe.working.memory.in.mb=5120
hive.metastore.uri=thrift://10.19.89.43:9083
hive.config.resources=/srv/spark2.2Bigdata/install/hadoop/datanode/etc/hadoop/hdfs-site.xml,/srv/spark2.2Bigdata/install/hadoop/datanode/etc/hadoop/core-site.xml,/srv/spark2.2Bigdata/install/hadoop/datanode/etc/hadoop/hive-site.xml
```

Presto Configurations for Query – Coordinator

```
coordinator=true
node-scheduler.include-coordinator=true
http-server.http.port=8086
discovery-server.enabled=true
discovery.uri=http://172.168.100.196:8086
http-server.http.port=8086
query.max-memory=600GB
query.max-memory-per-node=190GB
query.max-total-memory-per-node=195GB
task.max-partial-aggregation-memory=16MB
#Max size of partial aggregation result (if it is split able). High value may cause a drop in performance in unstable cluster condition.(before it was 32MB and decreased to 16MB)
task.max-worker-threads=96
#Sets the number of threads used by workers to process splits (Default value: Node CPUs * 2)
task.min-drivers = 192
#This describes how many drivers are kept on a worker at any time (Default value: Node CPUs * 4)
task.http-timeout-threads=3
#Presto server sends update of query status whenever it is different than the one that client knows about.(Before it was 10 and kept default value of 3)
task.http-response-threads=100
#Threads are created on demand and they end when there is no response to be sent.
task.info-update-interval=200ms
#Controls staleness of task information which is used in scheduling.(Before it was 100ms and kept default value)
query.execution-policy = phased
#Setting this value to phased will allow the query scheduler to split a single query execution between different time slots.(Here we have 2 types of strings all-at-once or phased)
node-scheduler.network-topology = flat
#Sets the network topology to use when scheduling splits. Legacy will ignore the topology when scheduling splits. Flat will try to schedule splits on the host where the data is located by reserving 50% of the work queue for local splits.
node-scheduler.max-splits-per-node=400
#This property describes how many splits can be queued to each worker node. Having this value higher will allow more jobs to be queued but will cause resources to be used for that
```

Presto Configurations for Query - Worker

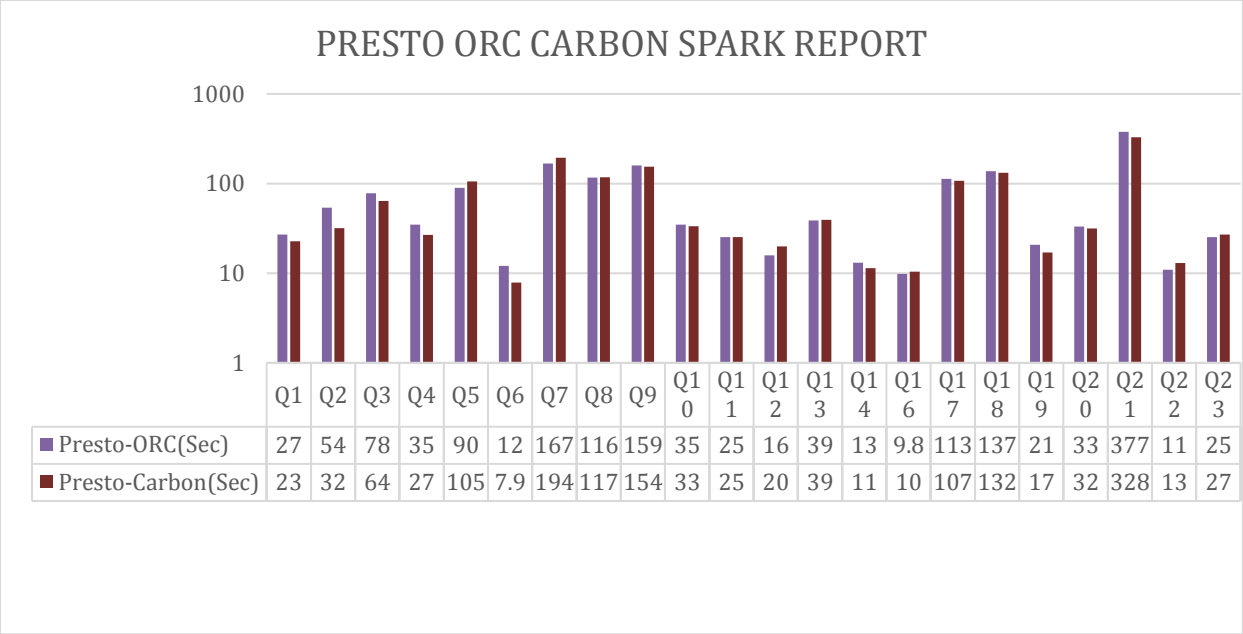
```
coordinator=false
node-scheduler.include-coordinator=false
#rest of the configuration are same as Coordinator.
```

Query Performance

The following chart depicts the performance of Presto Carbon and Presto ORC.

How it is tested

Executed each query three times and taken best out of it in both Presto Carbon and Presto ORC.



Queries	Presto-ORC	Presto-Carbon
Q1	26.972	22.791
Q2	53.894	31.809
Q3	78	64.129
Q4	34.7	26.819
Q5	89.95	105.394
Q6	12.082	7.863
Q7	166.982	193.506
Q8	116.437	117.341
Q9	158.817	153.885
Q10	34.697	33.314
Q11	25.206	25.26
Q12	15.877	19.905
Q13	38.659	39.409
Q14	13.141	11.431
Q16	9.792	10.366
Q17	113.124	107.21
Q18	137.062	132.01
Q19	20.753	17.059
Q20	33.175	31.564
Q21	376.691	327.888
Q22	10.911	12.965
Q23(Full Scan Query)	25.35	27.023

Scripts and data

Data Size : 500 GB (Generated using https://github.com/electrum/tpch-dbgen)

Number of Presto-Carbon Files: 1492
Number of Presto-ORC Files: 1485

Table Size of Presto-Carbon: 171.1 GB
Table Size of Presto-ORC: 163.8 GB

Note: Create and Load is done in Spark-Carbon and TPCH queries will be executed in Presto.

TPCH Queries

```
select l_returnflag, l_linestatus, sum(l_quantity) as sum_qty, sum(l_extendedprice) as sum_base_price, sum(l_extendedprice*(1-l_discount)) as sum_disc_price, sum(l_extendedprice*(1-l_discount)*(1+l_tax)) as sum_charge, avg(l_quantity) as avg_qty, avg(l_extendedprice) as avg_price, avg(l_discount) as avg_disc, count(*) as count_order from lineitem where l_shipdate <=date( '1998-09-02') group by l_returnflag, l_linestatus order by l_returnflag, l_linestatus;

select s_acctbal, s_name, n_name, p_partkey, p_mfgr, s_address, s_phone, s_comment from part, supplier, partsupp, nation, region where p_partkey = ps_partkey and s_suppkey = ps_suppkey and p_size = 15 and p_type like '%BRASS' and s_nationkey = n_nationkey and n_regionkey = r_regionkey and r_name = 'EUROPE' and ps_supplycost = ( select min(ps_supplycost) from partsupp, supplier,nation, region where p_partkey = ps_partkey and s_suppkey = ps_suppkey and s_nationkey = n_nationkey and n_regionkey = r_regionkey and r_name = 'EUROPE' ) order by s_acctbal desc, n_name, s_name, p_partkey limit 100;

select l_orderkey, sum(l_extendedprice * (1 - l_discount)) as revenue, o_orderdate, o_shippriority from customer, orders, lineitem where c_mktsegment = 'BUILDING' and c_custkey = o_custkey and l_orderkey = o_orderkey and o_orderdate < date('1995-03-15') and l_shipdate > date('1995-03-15') group by l_orderkey, o_orderdate, o_shippriority order by revenue desc, o_orderdate limit 10;

select o_orderpriority, count(*) as order_count from orders where o_orderdate >= date('1993-07-01') and o_orderdate < date('1993-10-01') and exists ( select * from lineitem where l_orderkey = o_orderkey and l_commitdate < l_receiptdate ) group by o_orderpriority order by o_orderpriority;

select n_name, sum(l_extendedprice * (1 - l_discount)) as revenue from customer, orders, lineitem, supplier, nation, region where c_custkey = o_custkey and l_orderkey = o_orderkey and l_suppkey = s_suppkey and c_nationkey = s_nationkey and s_nationkey = n_nationkey and n_regionkey = r_regionkey and r_name = 'ASIA' and o_orderdate >=date('1994-01-01') and o_orderdate < date('1995-01-01') group by n_name order by revenue desc;

select sum(l_extendedprice * l_discount) as revenue from lineitem where l_shipdate >= date('1994-01-01') and l_shipdate < date('1995-01-01') and l_discount between 0.05 and 0.07 and l_quantity < 24;

select supp_nation, cust_nation, l_year, sum(volume) as revenue from ( select n1.n_name as supp_nation, n2.n_name as cust_nation, year(l_shipdate) as l_year, l_extendedprice * (1 - l_discount) as volume from supplier,lineitem,orders,customer,nation n1,nation n2 where s_suppkey = l_suppkey and o_orderkey = l_orderkey and c_custkey = o_custkey and s_nationkey = n1.n_nationkey and c_nationkey = n2.n_nationkey and ( (n1.n_name = 'FRANCE' and n2.n_name = 'GERMANY') or (n1.n_name = 'GERMANY' and n2.n_name = 'FRANCE') ) and l_shipdate between date('1995-01-01') and date('1996-12-31') ) as shipping
```

```
group by supp_nation, cust_nation, l_year order by supp_nation, cust_nation, l_year;

select o_year, sum(case when nation = 'BRAZIL' then volume else 0 end) / sum(volume) as
mkt_share from (select year(o_orderdate) as o_year, l_extendedprice * (1-l_discount) as
volume, n2.n_name as nation from part,supplier,lineitem,orders,customer,nation n1,nation
n2,region where p_partkey = l_partkey and s_suppkey = l_suppkey and l_orderkey =
o_orderkey and o_custkey = c_custkey and c_nationkey = n1.n_nationkey and
n1.n_regionkey = r_regionkey and r_name = 'AMERICA' and s_nationkey = n2.n_nationkey
and o_orderdate between date('1995-01-01') and date('1996-12-31') and p_type =
'ECONOMY ANODIZED STEEL' ) as all_nations group by o_year order by o_year;

select nation, o_year, sum(amount) as sum_profit from ( select n_name as nation,
year(o_orderdate) as o_year, l_extendedprice * (1 - l_discount) - ps_supplycost * l_quantity as
amount from part, supplier, lineitem, partsupp, orders, nation where s_suppkey = l_suppkey
and ps_suppkey = l_suppkey and ps_partkey = l_partkey and p_partkey = l_partkey and
o_orderkey = l_orderkey and s_nationkey = n_nationkey and p_name like '%green%' ) as
profit group by nation, o_year order by nation, o_year desc;

select c_custkey, c_name, sum(l_extendedprice * (1 - l_discount)) as revenue, c_acctbal,
n_name, c_address, c_phone, c_comment from customer, orders, lineitem, nation where
c_custkey = o_custkey and l_orderkey = o_orderkey and o_orderdate >= date('1993-10-01')
and o_orderdate < date('1994-01-01') and l_returnflag = 'R' and c_nationkey = n_nationkey
group by c_custkey, c_name, c_acctbal, c_phone, n_name, c_address, c_comment order by
revenue desc limit 20;

select ps_partkey, sum(ps_supplycost * ps_availqty) as value from partsupp, supplier, nation
where ps_suppkey = s_suppkey and s_nationkey = n_nationkey and n_name = 'GERMANY'
group by ps_partkey having sum(ps_supplycost * ps_availqty) > ( select sum(ps_supplycost *
ps_availqty) * 0.0001000000 s from partsupp, supplier, nation where ps_suppkey =
s_suppkey and s_nationkey = n_nationkey and n_name = 'GERMANY' ) order by value desc;

select l_shipmode, sum(case when o_orderpriority = '1-URGENT' or o_orderpriority = '2-
HIGH' then 1 else 0 end) as high_line_count, sum(case when o_orderpriority <> '1-URGENT'
and o_orderpriority <> '2-HIGH' then 1 else 0 end) as low_line_count from orders, lineitem
where o_orderkey = l_orderkey and l_shipmode in ('MAIL', 'SHIP') and l_commitdate <
l_receiptdate and l_shipdate < l_commitdate and l_receiptdate >= date('1994-01-01') and
l_receiptdate < date('1995-01-01') group by l_shipmode order by l_shipmode;

select c_count, count(*) as custdist from (select c_custkey, count(o_orderkey) as c_count
from customer left outer join orders on ( c_custkey = o_custkey and o_comment not like
'%special%requests%' ) group by c_custkey ) as c_orders group by c_count order by custdist
desc, c_count desc;

select 100.00 * sum(case when p_type like 'PROMO%' then l_extendedprice * (1 - l_discount)
else 0 end) / sum(l_extendedprice * (1 - l_discount)) as promo_revenue from lineitem, part
where l_partkey = p_partkey and l_shipdate >= date('1995-09-01') and l_shipdate <
date('1995-10-01');

select p_brand, p_type, p_size, count(distinct ps_suppkey) as supplier_cnt from partsupp,
part where p_partkey = ps_partkey and p_brand <> 'Brand#45' and p_type not like 'MEDIUM
POLISHED%' and p_size in (49, 14, 23, 45, 19, 3, 36, 9) and ps_suppkey not in ( select
s_suppkey from supplier where s_comment like '%Customer%Complaints%' ) group by
p_brand, p_type, p_size order by supplier_cnt desc, p_brand, p_type, p_size;
select sum(l_extendedprice) / 7.0 as avg_yearly from lineitem,part where p_partkey =
l_partkey and p_brand = 'Brand#23' and p_container = 'MED BOX' and l_quantity < ( select
0.2 * avg(l_quantity) from lineitem where l_partkey = p_partkey );

select c_name, c_custkey, o_orderkey, o_orderdate, o_totalprice, sum(l_quantity) from
```

```
customer, orders, lineitem where o_orderkey in ( select l_orderkey from lineitem group by l_orderkey having sum(l_quantity) > 300 ) and c_custkey = o_custkey and o_orderkey = l_orderkey group by c_name, c_custkey, o_orderkey, o_orderdate, o_totalprice order by o_totalprice desc, o_orderdate;

select sum(l_extendedprice* (1 - l_discount)) as revenue from lineitem, part where ( p_partkey = l_partkey and p_brand = 'Brand#12' and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG') and l_quantity >= 1 and l_quantity <= 1 + 10 and p_size between 1 and 5 and l_shipmode in ('AIR', 'AIR REG') and l_shipinstruct = 'DELIVER IN PERSON' ) or ( p_partkey = l_partkey and p_brand = 'Brand#23' and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK') and l_quantity >= 10 and l_quantity <= 10 + 10 and p_size between 1 and 10 and l_shipmode in ('AIR', 'AIR REG') and l_shipinstruct = 'DELIVER IN PERSON' ) or ( p_partkey = l_partkey and p_brand = 'Brand#34' and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG') and l_quantity >= 20 and l_quantity <= 20 + 10 and p_size between 1 and 15 and l_shipmode in ('AIR', 'AIR REG') and l_shipinstruct = 'DELIVER IN PERSON' );

select s_name, s_address from supplier, nation where s_suppkey in ( select ps_suppkey from partsupp where ps_partkey in ( select p_partkey from part where p_name like 'forest%' ) and ps_availqty > ( select 0.5 * sum(l_quantity) from lineitem where l_partkey = ps_partkey and l_suppkey = ps_suppkey and l_shipdate >= date('1994-01-01') and l_shipdate < date('1995-01-01') ) ) and s_nationkey = n_nationkey and n_name = 'CANADA' order by s_name;

select s_name, count(*) as numwait from supplier, lineitem l1, orders, nation where s_suppkey = l1.l_suppkey and o_orderkey = l1.l_orderkey and o_orderstatus = 'F' and l1.l_receiptdate > l1.l_commitdate and exists ( select * from lineitem l2 where l2.l_orderkey = l1.l_orderkey and l2.l_suppkey <> l1.l_suppkey ) and not exists ( select * from lineitem l3 where l3.l_orderkey = l1.l_orderkey and l3.l_suppkey <> l1.l_suppkey and l3.l_receiptdate > l3.l_commitdate ) and s_nationkey = n_nationkey and n_name = 'SAUDI ARABIA' group by s_name order by numwait desc, s_name;

select cntrycode, count(*) as numcust, sum(c_acctbal) as totacctbal from ( select substring(c_phone,1 ,2) as cntrycode, c_acctbal from customer where substring(c_phone ,1,2) in ('13','31','23','29','30','18','17') and c_acctbal > ( select avg(c_acctbal) from customer where c_acctbal > 0.00 and substring(c_phone,1,2) in ('13', '31', '23', '29', '30', '18', '17') ) and not exists ( select * from orders where o_custkey = c_custkey ) ) as custsale group by cntrycode order by cntrycode;

select count(l_shipdate), count(l_shipinstruct), count(l_orderkey), count(l_suppkey), count(l_quantity), count(l_partkey), count(l_receiptdate), count(l_commitdate), count(l_comment), count(l_discount), count(l_linenumber), count(L_RETURNFLAG), count(L_LINESTATUS), count(l_shipmode) from lineitem;
```