HighGo Parallel Foreign Scan Whitepaper

December 2020

HighGo Software Inc. ™
10318 Whalley Blvd #1 Surrey, BC, V3T 4H4, Canada
T: +1 (604) 781-6749 E: contact@highgo.ca W: www.highgo.ca

TABLE OF CONTENTS

1 Abstract

Sharding in database is the ability to horizontally partition data across one more database shards. It is the mechanism to partition a table across one or more foreign servers. While the declarative partitioning feature allows users to partition tables into multiple partitioned tables living on the same database server, sharding allows tables to be partitioned in a way that the partitions live on external foreign servers and the parent table lives on the primary node where the user is creating the distributed table. The built-in sharding feature in PostgreSQL will use a FDW-based approach. FDW's are based on the SQL/MED specification that defines how an external data source can be accessed. PostgreSQL provides a number of foreign data wrappers (FDW's) that are used for accessing external data sources. Using the FDW-based sharding, the data is partitioned to the shards in order to optimize the query for the sharded table. Various parts of the query e.g., aggregates, joins, are pushed down to the shards. This enables the heavy query processing to be done on the shards and only results of the query are sent back to the primary node.

2 The Problem

When a query is querying multiple foreign scans in a single query, all the foreign scans are being executed in a sequential manner, one after another. Parallel foreign scan functionality will allow executing multiple foreign scans in parallel. This feature is particularly important for the OLAP use cases, for example if you are running a query on large volume sharded table with multiple partitions residing on multiple foreign servers, the query will be sent to each foreign server sequentially and results from each server are sent to the parent node. The important point to note is that without parallel foreign scan feature, each sub-partition will be scanned sequentially and with this feature all the sub-partitions residing on different foreign servers will be scanned in parallel and result will be sent in parallel to parent node. The parent node will process the data from all the foreign server nodes and sent the results back to the client.

3 The Solution

HG-PGSQL 2.0 organizes sharding structure in the way of external partition tables. Before adding Parallel Foreign Scan function, append node is used in the code to maintain the structural relationship between multiple external partition tables. When executing append node, the child nodes of each append node will be processed in turn. Suppose partition table A has three foreign sub-partitions A1, A2 and A3. When querying Table A, the executor will scan A1 and return data first, then scan A2 and return data, and then scan A3 and return data.

This function changes the execution logic of the Append node, sends a scan request to the three foreign subpartitions in parallel and then polls to receive the data returned by the three nodes. This can greatly improve query performance with less data returned.

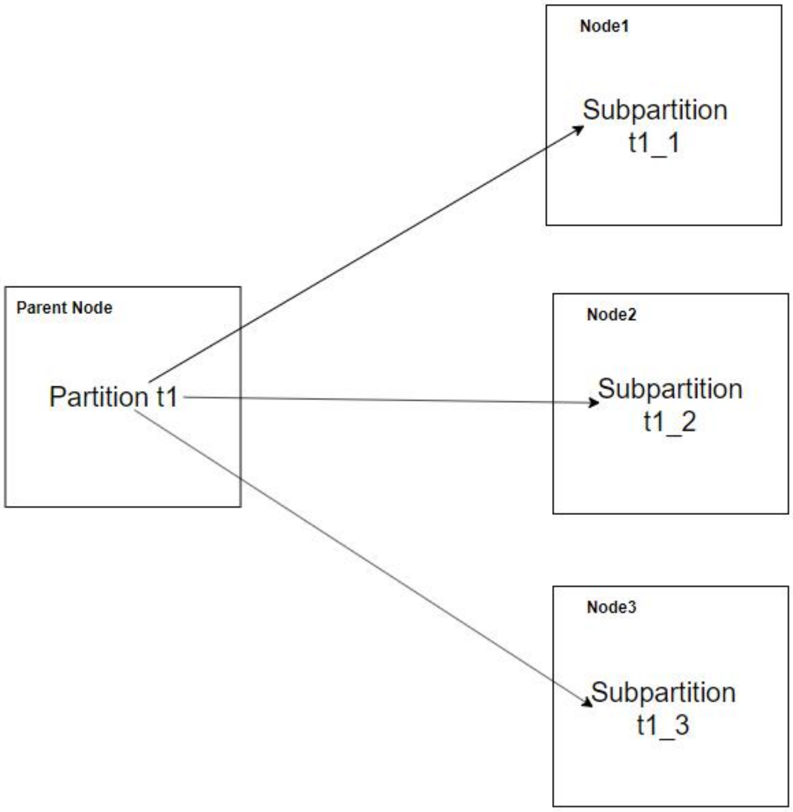
4 Benchmark Environment

4.1 Database Server

Instance Type	kails.xlarge.1

Volume Type	io1
Memory (GB)	4
Network Performance	Moderate
Architecture	X86_64, i386
vCPU #	4
IOP	16000
Storage (GB)	100

4.2 Test Topology:



We have partition t1 on Parent node, and subpartition t1_1 on node1, and subpartition t1_2 on node2, and subpartition t1_3 on node3. We will do query on t1 and get the time it spend to finish scan, to see if our feature gain performance.

5 Benchmark Result

We use a script to test HG-PSQL before and after the addition of parallel foreign scanning feature. We conducted multiple tests under the same data scale to get the average value and tested multiple data scales.

Result no Parallel Foreign Scan

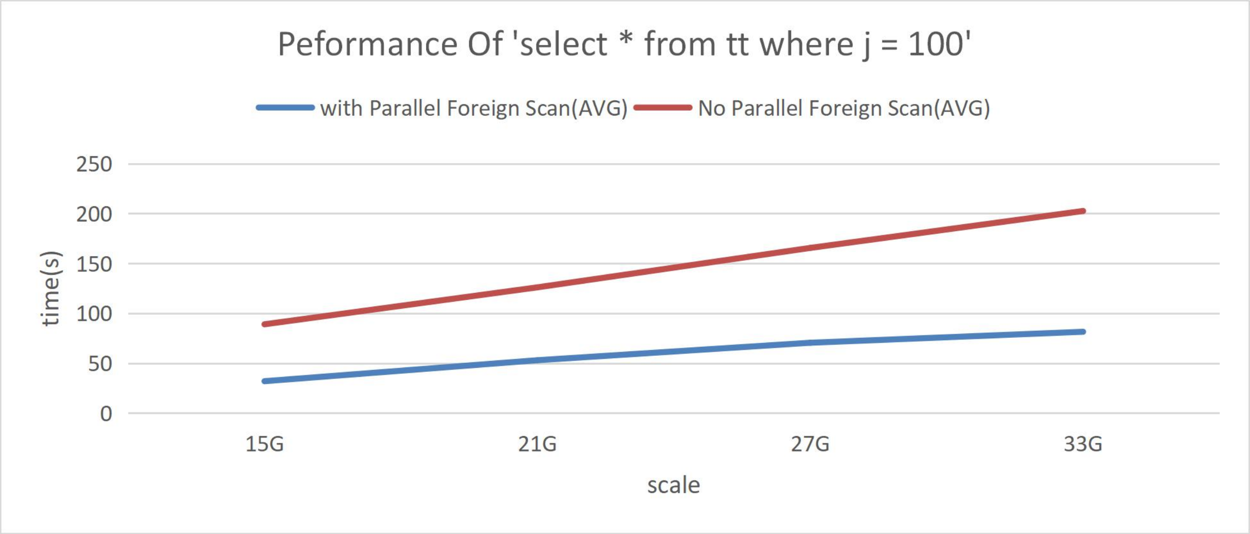
Scale	time(s)			AVG
15G	99	95	83	89
21G	142	126	126	126
27G	166	166	165	165.5
33G	202	202	203	202.5

Result with Parallel Foreign Scan

Scale	time(s)			AVG
15G	37	33	31	32
21G	50	48	58	53
27G	64	76	65	70.5
33G	84	91	72	81.5

Result Compare

Scale	With Parallel Foreign Scan(AVG)	No Parallel Foreign Scan(AVG)	Performance
15G	32	89	278%
21G	53	126	237%
27G	70.5	165.5	234%
33G	81.5	202.5	249%



We can see the test result, in all scales it spents less time with Parallel Foreign Scan feature, so we can say that we can gain performance in foreign partition scan with Parallel Foreign Scan feature.

6 Conclusion

This feature is particularly important for the OLAP use-case. The idea of having a large number of foreign servers containing partitions for a large partitioned table and aggregate operations running on all foreign servers in parallel is very powerful.

However, there is a limitation to this functionality. When most of the data in the table is returned, the performance pressure is shifted from the sub-partition node to the parent partition table node, in which case we get limited performance gains.

瀚高

HIGH GO