DolphinDB 与 Vertica 数据测试

测试环境

本次测试使用单物理节点集群部署

主机：DELL OptiPlex 7060（085A）

CPU ：Intel Core i7-8700（6 核 12 线程 3.20 GHz）

内存：32 GB （8GB × 4, 2666 MHz）

硬盘：2T HDD （222 MB/s 读取；210 MB/s 写入）

OS：CentOS 7.6.1810

测试使用的DolphinDB版本为Linux 0.97，最大内存设置为24G, ，数据副本设置为2，设置1个控制节点，1个代理节点，4个数据节点。 每个数据节点workernum 为8

测试使用的Vertica版本为Linux 9.2.1, 使用adminTools 创建database， cpu使用核数为12.

小数据集

小数据集使用device\_big\_readings.csv, 大小为4.2G。测试集包含3000个设备在2016年11月15日到2016年11月19日10000个时间间隔上的传感器时间，设备ID，电池，内存，CPU等时序统计信息。

分区方案

Dolphindb 和 vertica 都按每天的小时分区，15号到19号每小时做一个分区，总计84个分区。

Vertica导入测试

create table device(

time timestamp not null,

device\_id varchar primary key not null,

battery\_level int,

battery\_status varchar,

battery\_temperature float,

bassid varchar,

cpu\_avg\_1min float,

cpu\_avg\_5min float,

ti float,

mem\_free int,

mem\_used int,

rssi int,

ssid varchar)

Alter table device partition by extract(day from time)\*100 + extract(hour from time) REORGANIZE

COPY device from ‘/home/mhzhu/devices\_big\_readings.csv’ parser fcsvparser() DIRECT;

测试结果：

导入用时： 60s 289ms

导入后数据大小为1.16G

DolphinDB 导入测试

login("admin","123456")

FP\_DEVICES = 'dfs://device'

FP\_READINGS = '/home/mhzhu/devices\_big\_readings.csv'

FP\_INFO = '/home/mhzhu/devices\_big\_device\_info.csv'

FP\_DB = FP\_DEVICES + "/database";

COLS\_READINGS = `time`device\_id`battery\_level`battery\_status`battery\_temperature`bssid`cpu\_avg\_1min`cpu\_avg\_5min`cpu\_avg\_15min`mem\_free`mem\_used`rssi`ssid

TYPES\_READINGS = `DATETIME`SYMBOL`INT`SYMBOL`DOUBLE`SYMBOL`DOUBLE`DOUBLE`DOUBLE`LONG`LONG`SHORT`SYMBOL

schema\_readings = table(COLS\_READINGS, TYPES\_READINGS)

if (existsDatabase(FP\_DB)){

dropDatabase(FP\_DB);}

db = database(FP\_DB, RANGE, datehour(2016.11.15T00:00:00) + 0..95);

readings = loadTextEx(db,"readings",`time, FP\_READINGS, ,schema\_readings);

导入用时：30s 10ms

导入后数据大小为 1.3G

查询测试：每次查询之前需要清空缓存。Vertica 使用select clear\_caches, dolphindb 使用

clearAllCache()。两者都需使用 sudo sh -c "sync;echo 3 > /proc/sys/vm/drop\_caches " 清空系统缓存

查询语句

|  |  |  |
| --- | --- | --- |
|  | Vertica | DolphinDB |
| 1 | select \* from device where extract(day from time)\*100 + extract(hour from time) = 1507 | select \* from readings where time == 2016.11.15 07:00:00 |
| 2 | select \* from readings where device\_id = ‘demo000003’ | select \* from readings where device\_id = `demo000003 |
| 3 | select \* from device where extract(day from time)\*100 + extract(hour from time) between 1609 and 1611 and device\_id in ('demo000001', 'demo000010', 'demo000100', 'demo001000') | select \* from readings where time between 2016.11.16T09:00:00 : 2016.11.16T11:59:59,device\_id in ['demo000001', 'demo000010', 'demo000100', 'demo001000'] |
| 4 | select \* from device where time between ‘2016.11.15 07:00:01’ and ‘2016.11.15 07:59:59’ and device\_id = ‘demo000000’ | select \* from readings where time between 2016.11.15T07:00:01:2016.11.15T07:59:59, device\_id = `demo000000 |
| 5 | select \* from device where time between ‘2016.11.16 10:00:00’ and ‘2016.11.17 18:00:00’ and  device\_id in ('demo000002', 'demo000020', 'demo000200', 'demo002000') and battery\_level <= 80 and battery\_status = 'charging' | select \* from readings where time between 2016.11.16 10:00:00 : 2016.11.17 18:00:00,  device\_id in ['demo000002', 'demo000020', 'demo000200', 'demo002000'],battery\_level <= 80,battery\_status = 'charging' |
| 6 | select avg(battery\_temperature) from device group by device\_id | select avg(battery\_temperature) from readings group by device\_id |
| 7 | select min(battery\_temperature) from device group by device\_id and hour(time) | select min(battery\_temperature) from readings group by device\_id, hour(time) |
| 8 | select (max(mem\_free)-min(mem\_free)) from device group by hour(time) | select (max(mem\_free)-min(mem\_free)) from readings group by hour(time) |
| 9 | select max(mem\_used) - min(mem\_used) from device group by bar(time, 60 \* 5) | select max(mem\_used) - min(mem\_used) from readings group by bar(time, 60 \* 5) |
| 10 | 不支持pivot by | select avg(battery\_temperature) from readings  where  time between 2016.11.15 09:00:00 : 2016.11.16 06:00:00,  device\_id < 'demo000050'  pivot by time.hour(), device\_id |
| 11 | select max(cpu\_avg\_1min) as sum\_load from device where extract(day from time)\*100 + extract(hour from time) between 1512 and 1612 and device\_id in ('demo000001', 'demo000010', 'demo000100', 'demo001000') group by hour(time) order by sum\_load | select max(cpu\_avg\_1min) as sum\_load from readings where time between 2016.11.15 12:00:00 : 2016.11.16 12:00:00, device\_id in ['demo000001', 'demo000010', 'demo000100', 'demo001000'] group by hour(time) order by sum\_load |
| 12 | select max(mem\_free) as mem\_max from device where time <= ‘2016.11.18 06:00:00’ and battery\_level >= 100 and cpu\_avg\_5min > 100 group by hour(time), device\_id | select max(mem\_free) as mem\_max from readings where time <= 2016.11.18 06:00:00, battery\_level >= 100, cpu\_avg\_5min > 100 group by hour(time), device\_id |
| 13 | select \* from device join info on device.device\_id = info.id where time between '2016.11.16 09:00:00' and '2016.11.16 12:00:00' and device\_id='demo000060' and battery\_status='discharging'; | select \* from lj(readings,device\_info,`device\_id) where time between 2016.11.16T09:00:00:2016.11.16T12:00:00,device\_id==`demo000060,battery\_status==`discharging |
| 14 | select avg(battery\_temperature) as avg\_temperature,sum(cpu\_avg\_1min) as sum\_cpu\_avg\_15min from device join info on device.device\_id = info.id where device\_id between 'demo000200' and 'demo000300' and time between '2016.11.16 12:00:00' and '2016.11.16 19:00:00' group by hour(time),device\_id order by avg\_temperature; | select avg(battery\_temperature) as avg\_temperature,sum(cpu\_avg\_1min) as sum\_cpu\_avg\_15min from lj(readings,device\_info,`device\_id) where device\_id between `demo000200:`demo000300,time between 2016.11.16T12:00:00:2016.11.16T19:00:00 group by hour(time),device\_id order by avg\_temperature; |
|  |  |  |

结果

|  |  |  |
| --- | --- | --- |
|  | Vertica | DolphinDB |
| 1 | 110.752ms | 82.355ms |
| 2 | 809.377ms | 732.119ms |
| 3 | 108.589ms | 5.551ms |
| 4 | 53.187ms | 87.167ms |
| 5 | 275.545ms | 326.761ms |
| 6 | 701.962ms | 246.834ms |
| 7 | 960.543ms | 164ms |
| 8 | 957.273ms | 119.963ms |
| 9 | 619.744ms | 134.116ms |
| 10 |  | 44.694ms |
| 11 | 295.018ms | 44.049ms |
| 12 | 1247.685ms | 257.17ms |
| 13 | 66.072ma | 85.255ms |
| 14 | 53.617ms | 26.48ms |

大数据

大数据集使用20070910-20070914和20070917-20070921 十天中NYSE股票数据集。十个csv文件总大小为89.68G.

分区方案

因为vertica 中最大分区数为1024个，过多分区会降低其性能，所以依然采取小数据分区方案，按每天小时分区，一天17个分区，总共170个分区

DolphinDB 采用复合分区，按天分区后，再按symbol的range分区， 总计1002个分区

因为Vertica不支持按端口分配node, node必须是主机，所以无法进行并行导入。DolphinDB和Vertica都使用串行导入。

Ssd 空间不足，所以数据储存于hdd

Vertica 导入

create table taq(

symbol varchar not null,

date date not null,

time time not null,

bid float,

ofr float,

bidsiz int,

ofrsiz int,

mode int,

ex char,

mmid varchar)

Alter table device partition by extract(day from time)\*100 + extract(hour from time) REORGANIZE

COPY device from ‘/home/mhzhu/69d2fb258f5a038a.csv’，’/home/mhzhu/58d241f679648e15.csv’, ‘/home/mhzhu/55ea5003a3761c42.csv’, ‘/home/mhzhu/8b849aac0c818628.csv’, ‘/home/mhzhu/6ca6657c186703d5.csv’, ‘/home/mhzhu/3f05726b5433c223.csv’, ‘/home/mhzhu/0a0f485e8b27db7b.csv’, ‘/home/mhzhu/eff2bcc26f4c7a02.csv’, ‘/home/mhzhu/e30e8ce6d7019602.csv’, ‘/home/mhzhu/defe1dbca92b29ec.csv’ parser fcsvparser() DIRECT;

导入用时

22m3s714ms

导入大小

16G

DolphinDB 导入

login("admin","123456")

// ----------------- 路径配置

FP\_TAQ = 'dfs://TAQ'

FP\_SAMPLE\_TB = '/home/mhzhu/0a0f485e8b27db7b.csv'

FP\_DB = FP\_TAQ + '/databases'

FP\_READINGS1 = '/home/mhzhu/0a0f485e8b27db7b.csv'

FP\_READINGS2 = '/home/mhzhu/3f05726b5433c223.csv'

FP\_READINGS3 = '/home/mhzhu/55ea5003a3761c42.csv'

FP\_READINGS4 = '/home/mhzhu/58d241f679648e15.csv'

FP\_READINGS5 = '/home/mhzhu/69d2fb258f5a038a.csv'

FP\_READINGS6 = '/home/mhzhu/6ca6657c186703d5.csv'

FP\_READINGS7 = '/home/mhzhu/8b849aac0c818628.csv'

FP\_READINGS8 = '/home/mhzhu/e30e8ce6d7019602.csv'

FP\_READINGS9 = '/home/mhzhu/eff2bcc26f4c7a02.csv'

FP\_READINGS10 = '/home/mhzhu/defe1dbca92b29ec.csv'

//

// ----------------- 创建 schema

orig\_tb\_schema = extractTextSchema(FP\_SAMPLE\_TB)

// 查看 orig\_tb\_schema

// 将列名调整为小写避免与 DolphinDB 内置的 SYMBOL, DATE, TIME 等保留关键字产生冲突

cols = lower(orig\_tb\_schema.name)

schema = table(cols, orig\_tb\_schema.type)

// 加载 sample table, 用第一个 CSV 中的股票代码频率作为分区范围的依据

sample\_tb = ploadText(FP\_SAMPLE\_TB, , schema)

sample\_freq\_tb = select count(\*) from sample\_tb group by symbol

sample\_tb = NULL

// 8369 rows, [symbol, count], 分到 100 个 buckets

BIN\_NUM = 100

buckets = cutPoints(sample\_freq\_tb.symbol, BIN\_NUM, sample\_freq\_tb.count)

// [A, ABL, ACU, ..., ZZZ], 101 个边界

buckets[BIN\_NUM] = `ZZZZZZ // 调整最右边界

DATE\_RANGE = 2007.01.01..2008.01.01

date\_schema = database('', VALUE, DATE\_RANGE)

symbol\_schema = database('', RANGE, buckets)

cols = lower(orig\_tb\_schema.name)

schema = table(cols, orig\_tb\_schema.type)

select count(\*) from rpc(getControllerAlias(),getClusterChunksStatus) where file ilike '%/taq/%';

if (existsDatabase(FP\_DB)){

dropDatabase(FP\_DB);}

db = database(FP\_DB, COMPO, [date\_schema, symbol\_schema])

readings = loadTextEx(db,"readings",`date`symbol, FP\_READINGS1, ,schema);

readings = loadTextEx(db,"readings",`date`symbol, FP\_READINGS2, ,schema);

readings = loadTextEx(db,"readings",`date`symbol, FP\_READINGS3, ,schema);

readings = loadTextEx(db,"readings",`date`symbol, FP\_READINGS4, ,schema);

readings = loadTextEx(db,"readings",`date`symbol, FP\_READINGS5, ,schema);

readings = loadTextEx(db,"readings",`date`symbol, FP\_READINGS6, ,schema);

readings = loadTextEx(db,"readings",`date`symbol, FP\_READINGS7, ,schema);

readings = loadTextEx(db,"readings",`date`symbol, FP\_READINGS8, ,schema);

readings = loadTextEx(db,"readings",`date`symbol, FP\_READINGS9, ,schema);

readings = loadTextEx(db,"readings",`date`symbol, FP\_READINGS10, ,schema);

导入用时14m58s338ms

所占空间 16.4G

|  |  |  |
| --- | --- | --- |
|  | Vertica | DolphinDB |
| 1 | select \* from taq where symbol = 'IBM' and date == 2007.09.12 | select \* from readings where symbol = 'IBM', date == 2007.09.12 |
| 2 | select \* from taq where time == 09:25:20 and date == 2007.09.11 | select \* from readings where time == 09:25:20, date == 2007.09.11 |
| 3 | select \* from taq where symbol in ('IBM', 'MSFT', 'GOOG', 'YHOO') and  date between ‘2007.09.11’ and ‘2007.09.13’ and time between ‘10:00:00’ and ‘20:40:00’ and bid > 20 | select \* from readings where symbol in ('IBM', 'MSFT', 'GOOG', 'YHOO'),  date between 2007.09.11 : 2007.09.13,time between 10:00:00 :20:40:00,bid > 20 |
| 4 | select \* from taq where symbol in ('IBM', 'MSFT', 'GOOG', 'AMZN') and  date = ‘2007.09.14’ and time >= ‘10:00:00’  order by ofr desc | select \* from readings where symbol in ('IBM', 'MSFT', 'GOOG', 'AMZN'),  date == 2007.09.14,time >= 10:00:00  order by ofr desc |
| 5 | select max(bid) as max\_bid,max(ofr) as max\_ofr from taq where date = ‘2007.09.11’ and  symbol = 'IBM' group by hour(time) | select max(bid) as max\_bid,max(ofr) as max\_ofr from readings where date == 2007.09.11,  symbol == 'IBM' group by hour(time) |
| 6 | select std(bid) as std\_bid,sum(bidsiz) as sum\_bidsiz from taq where date = ‘2007.09.18’ and time between ‘10:00:00’ and ‘21:00:00’ and symbol in (‘IBM’,’MSFT’,’GOOG’,’YHOO’) and bid > 25 and ofr > 25 group by symbol, minute(time) order by symbol, minute\_time | select std(bid) as std\_bid,sum(bidsiz) as sum\_bidsiz from readings where date == 2007.09.18,time between 10:00:00 : 21:00:00,symbol in `IBM`MSFT`GOOG`YHOO,bid > 25,ofr > 25 group by symbol, minute(time) order by symbol, minute\_time |
| 7 | select symbol, time, bid, ofr from taq where symbol in ('IBM', 'MSFT', 'GOOG', 'YHOO') and date = ‘2007.09.14’ and time between ‘10:30:00’ and ‘18:30:00’ and bid > 10 and ofr >10 | select symbol, time, bid, ofr from readings where symbol in ('IBM', 'MSFT', 'GOOG', 'YHOO'), date = 2007.09.14, time between 10:30:00 : 18:30:00, bid > 10, ofr >10 |
| 8 | select std(bid) as std\_bid,avg(bid) as avg\_bid from taq where date = ‘2007.09.20’ and time between ‘11:30:00’ and ‘18:00:00’ and symbol in (`IBM,`MSFT,`GOOG,`YHOO) and bid>20 group by symbol, minute(time) | select std(bid) as std\_bid,avg(bid) as avg\_bid from readings where date ==2007.09.20,time between 11:30:00 : 18:00:00,symbol in (`IBM,`MSFT,`GOOG,`YHOO),bid>20 group by symbol, minute(time) |
| 9 | select max(ofr-bid) as max\_price from taq where date between ‘2007.09.16’ and ‘2017.09.20’ and time between ‘07:30:00’ and ‘10:00:00’ and symbol in ('GOOG',`SBW,'MSFT',`USBE,'YHOO') group by symbol order by max\_price desc | select max(ofr-bid) as max\_price from readings where date between 2007.09.16:2017.09.20,time between 07:30:00 : 10:00:00,symbol in ('GOOG',`SBW,'MSFT',`USBE,'YHOO') group by symbol order by max\_price desc |
| 10 | select avg(ofr) as avg\_ofr,avg(bid) as avg\_bid from taq where symbol = ‘IBM’ and date in (‘2007.09.14’, ‘2007.09.19’, ‘2007.09.20’) and time between ‘12:00:00’ and ‘16:00:00’ group by date, hour(time) | select avg(ofr) as avg\_ofr,avg(bid) as avg\_bid from readings where symbol = `IBM, date in [2007.09.14, 2007.09.19, 2007.09.20] ,time between 12:00:00 :16:00:00 group by date, hour(time) |
| 11 | select avg(ofr + bid) as avg\_price from taq where date between ‘2007.09.10’ and ‘2007.09.14’ and time between ‘13:00:00’ and ‘17:30:00’ and symbol in (‘IPB’,’SBW’,’IBM’,’USBE’,'YHOO') group by symbol, date | select avg(ofr + bid) as avg\_price from readings where date between 2007.09.10 : 2007.09.14,time between 13:00:00 : 17:30:00,symbol in (`IPB,`SBW,`IBM,`USBE,'YHOO') group by symbol, date |
| 12 | select sum(ofr) as total\_ofr, sum(bid) as total\_bid from taq where date between ‘2007.09.17’ and ‘2007.09.21’ and symbol in (`CCMP,`SBW,`IBM,`USBE,`YHOO,`XOMA) group by symbol order by total\_ofr desc, total\_bid desc | select sum(ofr) as total\_ofr, sum(bid) as total\_bid from readings where date between 2007.09.17 : 2007.09.21, symbol in (`CCMP,`SBW,`IBM,`USBE,`YHOO,`XOMA) group by symbol order by total\_ofr desc, total\_bid desc |
|  |  |  |
|  |  |  |
|  |  |  |

测试结果

|  |  |  |
| --- | --- | --- |
|  | Vertica | DolphinDB |
| 1 | 1153.218ms | 352.467ms |
| 2 | 2942.121ms | 7106.013ms |
| 3 | 7910.514ms | 1328.684ms |
| 4 | 5002.597ms | 536.12ms |
| 5 | 425.455ms | 11.814ms |
| 6 | 1920.204ms | 34.24ms |
| 7 | 1249.134ms | 396.105ms |
| 8 | 567.74ms | 23.173ms |
| 9 | 3629.865ms | 211.155ms |
| 10 | 663.261ms | 8.9ms |
| 11 | 5720.305ms | 1003.966ms |
| 12 | 14721.478ms | 204.443ms |