## BenchmarkSQL - TPCC

### 0. 环境配置

硬件：

华为云服务器  
操作系统：CentOS 7.6  
都是 8核32G，500G磁盘  
  
ecs-faaa-0001  
8vCPUs | 32GiB | c7.2xlarge.4  
CentOS 7.6 64bit  
122.9.158.249(弹性公网)10 Mbit/s  
192.168.0.30(私有)   
  
ecs-faaa-0002  
8vCPUs | 32GiB | c7.2xlarge.4  
CentOS 7.6 64bit  
122.9.163.166(弹性公网)10 Mbit/s  
192.168.0.163(私有)   
  
ecs-faaa-0003  
8vCPUs | 32GiB | c7.2xlarge.4  
CentOS 7.6 64bit  
140.210.198.118(弹性公网)10 Mbit/s  
192.168.0.220(私有)

软件环境：

Klustron 1.2.1

BenchmarkSQL 5.0

### 1. 添加配置文件

配置props文件(配置文件切忌多余空格，否则会出现各种错误) /root/benchmark/benchmarksql-5.0/run/props.pg ：

500 warehouse

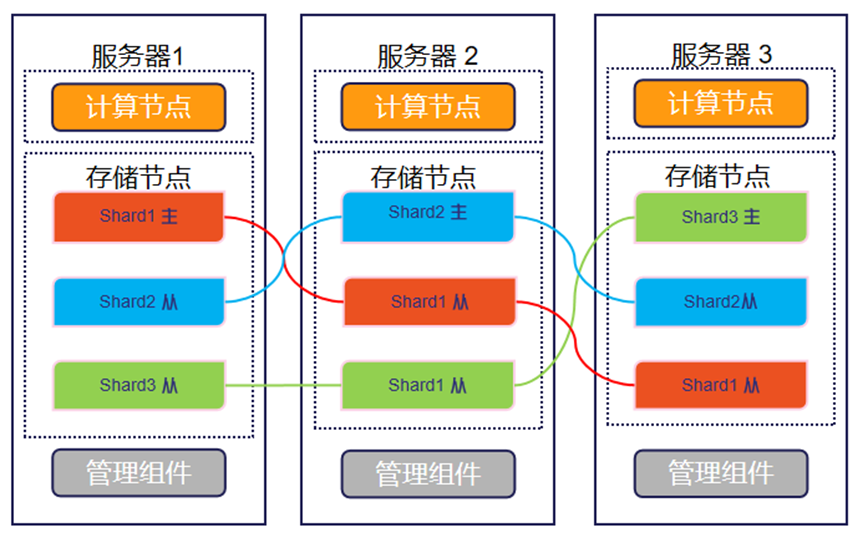
props.pg 关键配置文件：

#warehouses=1000  
warehouses=500  
loadWorkers=32  
  
#parallel numbers  
terminals=600  
//To run specified transactions per terminal- runMins must equal zero  
runTxnsPerTerminal=0  
//To run for specified minutes- runTxnsPerTerminal must equal zero  
runMins=5  
//Number of total transactions per minute  
limitTxnsPerMin=0  
  
//Set to true to run in 4.x compatible mode. Set to false to use the  
//entire configured database evenly.  
terminalWarehouseFixed=true  
  
//The following five values must add up to 100  
newOrderWeight=45  
paymentWeight=43  
orderStatusWeight=4  
deliveryWeight=4  
stockLevelWeight=4  
  
// Directory name to create for collecting detailed result data.  
// Comment this out to suppress.  
//resultDirectory=my\_result\_%tY-%tm-%td\_%tH%tM%tS  
//osCollectorScript=./misc/os\_collector\_linux.py  
//osCollectorInterval=1  
//osCollectorSSHAddr=user@dbhost  
//osCollectorDevices=net\_enp133s0 blk\_nvme0n1

### 2. 部署方式-对等部署

Klustron 集群的物理服务器数量没有限制，对等部署架构的特点是计算节点与存储节点并存于同一台物理服务器内，每台服务器既运行计算节点，同时也运行存储节点，在实际环境部署过程中，将计算引擎和存储引擎部署在同一台服务器上，以达到节约成本和获得资源的有效利用，同时通过不同节点间的组件冗余来实现高可靠性。

如下图所示：



建立分区表

表按需增加表分区；通过定义合理的分区方式可以最优化系统的数据读写性能和事务处理性，表结构如下：

create table bmsql\_config (

cfg\_name varchar(30) primary key,

cfg\_value varchar(50)

);

create table bmsql\_warehouse (

w\_id integer not null,

w\_ytd decimal(12,2),

w\_tax decimal(4,4),

w\_name varchar(10),

w\_street\_1 varchar(20),

w\_street\_2 varchar(20),

w\_city varchar(20),

w\_state char(2),

w\_zip char(9)

) partition by hash(w\_id);

create table bmsql\_warehouse\_0 partition of bmsql\_warehouse for values with (MODULUS 9, REMAINDER 0) with (shard = 11);

create table bmsql\_warehouse\_1 partition of bmsql\_warehouse for values with (MODULUS 9, REMAINDER 1) with (shard = 11);

create table bmsql\_warehouse\_2 partition of bmsql\_warehouse for values with (MODULUS 9, REMAINDER 2) with (shard = 11);

create table bmsql\_warehouse\_3 partition of bmsql\_warehouse for values with (MODULUS 9, REMAINDER 3) with (shard = 12);

create table bmsql\_warehouse\_4 partition of bmsql\_warehouse for values with (MODULUS 9, REMAINDER 4) with (shard = 12);

create table bmsql\_warehouse\_5 partition of bmsql\_warehouse for values with (MODULUS 9, REMAINDER 5) with (shard = 12);

create table bmsql\_warehouse\_6 partition of bmsql\_warehouse for values with (MODULUS 9, REMAINDER 6) with (shard = 13);

create table bmsql\_warehouse\_7 partition of bmsql\_warehouse for values with (MODULUS 9, REMAINDER 7) with (shard = 13);

create table bmsql\_warehouse\_8 partition of bmsql\_warehouse for values with (MODULUS 9, REMAINDER 8) with (shard = 13);

create table bmsql\_district (

d\_w\_id integer not null,

d\_id integer not null,

d\_ytd decimal(12,2),

d\_tax decimal(4,4),

d\_next\_o\_id integer,

d\_name varchar(10),

d\_street\_1 varchar(20),

d\_street\_2 varchar(20),

d\_city varchar(20),

d\_state char(2),

d\_zip char(9)

) partition by hash(d\_w\_id);

create table bmsql\_district\_0 partition of bmsql\_district for values with (MODULUS 9, REMAINDER 0) with (shard = 11);

create table bmsql\_district\_1 partition of bmsql\_district for values with (MODULUS 9, REMAINDER 1) with (shard = 11);

create table bmsql\_district\_2 partition of bmsql\_district for values with (MODULUS 9, REMAINDER 2) with (shard = 11);

create table bmsql\_district\_3 partition of bmsql\_district for values with (MODULUS 9, REMAINDER 3) with (shard = 12);

create table bmsql\_district\_4 partition of bmsql\_district for values with (MODULUS 9, REMAINDER 4) with (shard = 12);

create table bmsql\_district\_5 partition of bmsql\_district for values with (MODULUS 9, REMAINDER 5) with (shard = 12);

create table bmsql\_district\_6 partition of bmsql\_district for values with (MODULUS 9, REMAINDER 6) with (shard = 13);

create table bmsql\_district\_7 partition of bmsql\_district for values with (MODULUS 9, REMAINDER 7) with (shard = 13);

create table bmsql\_district\_8 partition of bmsql\_district for values with (MODULUS 9, REMAINDER 8) with (shard = 13);

create table bmsql\_customer (

c\_w\_id integer not null,

c\_d\_id integer not null,

c\_id integer not null,

c\_discount decimal(4,4),

c\_credit char(2),

c\_last varchar(16),

c\_first varchar(16),

c\_credit\_lim decimal(12,2),

c\_balance decimal(12,2),

c\_ytd\_payment decimal(12,2),

c\_payment\_cnt integer,

c\_delivery\_cnt integer,

c\_street\_1 varchar(20),

c\_street\_2 varchar(20),

c\_city varchar(20),

c\_state char(2),

c\_zip char(9),

c\_phone char(16),

c\_since timestamp,

c\_middle char(2),

c\_data varchar(500)

) partition by hash(c\_w\_id);

create table bmsql\_customer\_0 partition of bmsql\_customer for values with (MODULUS 9, REMAINDER 0) with (shard = 11);

create table bmsql\_customer\_1 partition of bmsql\_customer for values with (MODULUS 9, REMAINDER 1) with (shard = 11);

create table bmsql\_customer\_2 partition of bmsql\_customer for values with (MODULUS 9, REMAINDER 2) with (shard = 11);

create table bmsql\_customer\_3 partition of bmsql\_customer for values with (MODULUS 9, REMAINDER 3) with (shard = 12);

create table bmsql\_customer\_4 partition of bmsql\_customer for values with (MODULUS 9, REMAINDER 4) with (shard = 12);

create table bmsql\_customer\_5 partition of bmsql\_customer for values with (MODULUS 9, REMAINDER 5) with (shard = 12);

create table bmsql\_customer\_6 partition of bmsql\_customer for values with (MODULUS 9, REMAINDER 6) with (shard = 13);

create table bmsql\_customer\_7 partition of bmsql\_customer for values with (MODULUS 9, REMAINDER 7) with (shard = 13);

create table bmsql\_customer\_8 partition of bmsql\_customer for values with (MODULUS 9, REMAINDER 8) with (shard = 13);

create sequence bmsql\_hist\_id\_seq;

create table bmsql\_history (

hist\_id integer,

h\_c\_id integer,

h\_c\_d\_id integer,

h\_c\_w\_id integer,

h\_d\_id integer,

h\_w\_id integer,

h\_date timestamp,

h\_amount decimal(6,2),

h\_data varchar(24)

) partition by hash(h\_c\_w\_id);

create table bmsql\_history\_0 partition of bmsql\_history for values with (MODULUS 9, REMAINDER 0) with (shard = 11);

create table bmsql\_history\_1 partition of bmsql\_history for values with (MODULUS 9, REMAINDER 1) with (shard = 11);

create table bmsql\_history\_2 partition of bmsql\_history for values with (MODULUS 9, REMAINDER 2) with (shard = 11);

create table bmsql\_history\_3 partition of bmsql\_history for values with (MODULUS 9, REMAINDER 3) with (shard = 12);

create table bmsql\_history\_4 partition of bmsql\_history for values with (MODULUS 9, REMAINDER 4) with (shard = 12);

create table bmsql\_history\_5 partition of bmsql\_history for values with (MODULUS 9, REMAINDER 5) with (shard = 12);

create table bmsql\_history\_6 partition of bmsql\_history for values with (MODULUS 9, REMAINDER 6) with (shard = 13);

create table bmsql\_history\_7 partition of bmsql\_history for values with (MODULUS 9, REMAINDER 7) with (shard = 13);

create table bmsql\_history\_8 partition of bmsql\_history for values with (MODULUS 9, REMAINDER 8) with (shard = 13);

create table bmsql\_new\_order (

no\_w\_id integer not null,

no\_d\_id integer not null,

no\_o\_id integer not null

) partition by hash(no\_w\_id);

create table bmsql\_new\_order\_0 partition of bmsql\_new\_order for values with (MODULUS 9, REMAINDER 0) with (shard = 11);

create table bmsql\_new\_order\_1 partition of bmsql\_new\_order for values with (MODULUS 9, REMAINDER 1) with (shard = 11);

create table bmsql\_new\_order\_2 partition of bmsql\_new\_order for values with (MODULUS 9, REMAINDER 2) with (shard = 11);

create table bmsql\_new\_order\_3 partition of bmsql\_new\_order for values with (MODULUS 9, REMAINDER 3) with (shard = 12);

create table bmsql\_new\_order\_4 partition of bmsql\_new\_order for values with (MODULUS 9, REMAINDER 4) with (shard = 12);

create table bmsql\_new\_order\_5 partition of bmsql\_new\_order for values with (MODULUS 9, REMAINDER 5) with (shard = 12);

create table bmsql\_new\_order\_6 partition of bmsql\_new\_order for values with (MODULUS 9, REMAINDER 6) with (shard = 13);

create table bmsql\_new\_order\_7 partition of bmsql\_new\_order for values with (MODULUS 9, REMAINDER 7) with (shard = 13);

create table bmsql\_new\_order\_8 partition of bmsql\_new\_order for values with (MODULUS 9, REMAINDER 8) with (shard = 13);

create table bmsql\_oorder (

o\_w\_id integer not null,

o\_d\_id integer not null,

o\_id integer not null,

o\_c\_id integer,

o\_carrier\_id integer,

o\_ol\_cnt integer,

o\_all\_local integer,

o\_entry\_d timestamp

) partition by hash(o\_w\_id);

create table bmsql\_oorder\_0 partition of bmsql\_oorder for values with (MODULUS 9, REMAINDER 0) with (shard = 11);

create table bmsql\_oorder\_1 partition of bmsql\_oorder for values with (MODULUS 9, REMAINDER 1) with (shard = 11);

create table bmsql\_oorder\_2 partition of bmsql\_oorder for values with (MODULUS 9, REMAINDER 2) with (shard = 11);

create table bmsql\_oorder\_3 partition of bmsql\_oorder for values with (MODULUS 9, REMAINDER 3) with (shard = 12);

create table bmsql\_oorder\_4 partition of bmsql\_oorder for values with (MODULUS 9, REMAINDER 4) with (shard = 12);

create table bmsql\_oorder\_5 partition of bmsql\_oorder for values with (MODULUS 9, REMAINDER 5) with (shard = 12);

create table bmsql\_oorder\_6 partition of bmsql\_oorder for values with (MODULUS 9, REMAINDER 6) with (shard = 13);

create table bmsql\_oorder\_7 partition of bmsql\_oorder for values with (MODULUS 9, REMAINDER 7) with (shard = 13);

create table bmsql\_oorder\_8 partition of bmsql\_oorder for values with (MODULUS 9, REMAINDER 8) with (shard = 13);

create table bmsql\_order\_line (

ol\_w\_id integer not null,

ol\_d\_id integer not null,

ol\_o\_id integer not null,

ol\_number integer not null,

ol\_i\_id integer not null,

ol\_delivery\_d timestamp,

ol\_amount decimal(6,2),

ol\_supply\_w\_id integer,

ol\_quantity integer,

ol\_dist\_info char(24)

) partition by hash(ol\_w\_id);

create table bmsql\_order\_line\_0 partition of bmsql\_order\_line for values with (MODULUS 9, REMAINDER 0) with (shard = 11);

create table bmsql\_order\_line\_1 partition of bmsql\_order\_line for values with (MODULUS 9, REMAINDER 1) with (shard = 11);

create table bmsql\_order\_line\_2 partition of bmsql\_order\_line for values with (MODULUS 9, REMAINDER 2) with (shard = 11);

create table bmsql\_order\_line\_3 partition of bmsql\_order\_line for values with (MODULUS 9, REMAINDER 3) with (shard = 12);

create table bmsql\_order\_line\_4 partition of bmsql\_order\_line for values with (MODULUS 9, REMAINDER 4) with (shard = 12);

create table bmsql\_order\_line\_5 partition of bmsql\_order\_line for values with (MODULUS 9, REMAINDER 5) with (shard = 12);

create table bmsql\_order\_line\_6 partition of bmsql\_order\_line for values with (MODULUS 9, REMAINDER 6) with (shard = 13);

create table bmsql\_order\_line\_7 partition of bmsql\_order\_line for values with (MODULUS 9, REMAINDER 7) with (shard = 13);

create table bmsql\_order\_line\_8 partition of bmsql\_order\_line for values with (MODULUS 9, REMAINDER 8) with (shard = 13);

create table bmsql\_item (

i\_id integer not null,

i\_name varchar(24),

i\_price decimal(5,2),

i\_data varchar(50),

i\_im\_id integer

) with (shard = all);

create table bmsql\_stock (

s\_w\_id integer not null,

s\_i\_id integer not null,

s\_quantity integer,

s\_ytd integer,

s\_order\_cnt integer,

s\_remote\_cnt integer,

s\_data varchar(50),

s\_dist\_01 char(24),

s\_dist\_02 char(24),

s\_dist\_03 char(24),

s\_dist\_04 char(24),

s\_dist\_05 char(24),

s\_dist\_06 char(24),

s\_dist\_07 char(24),

s\_dist\_08 char(24),

s\_dist\_09 char(24),

s\_dist\_10 char(24)

) partition by hash(s\_w\_id);

create table bmsql\_stock\_0 partition of bmsql\_stock for values with (MODULUS 9, REMAINDER 0) with (shard = 11);

create table bmsql\_stock\_1 partition of bmsql\_stock for values with (MODULUS 9, REMAINDER 1) with (shard = 11);

create table bmsql\_stock\_2 partition of bmsql\_stock for values with (MODULUS 9, REMAINDER 2) with (shard = 11);

create table bmsql\_stock\_3 partition of bmsql\_stock for values with (MODULUS 9, REMAINDER 3) with (shard = 12);

create table bmsql\_stock\_4 partition of bmsql\_stock for values with (MODULUS 9, REMAINDER 4) with (shard = 12);

create table bmsql\_stock\_5 partition of bmsql\_stock for values with (MODULUS 9, REMAINDER 5) with (shard = 12);

create table bmsql\_stock\_6 partition of bmsql\_stock for values with (MODULUS 9, REMAINDER 6) with (shard = 13);

create table bmsql\_stock\_7 partition of bmsql\_stock for values with (MODULUS 9, REMAINDER 7) with (shard = 13);

create table bmsql\_stock\_8 partition of bmsql\_stock for values with (MODULUS 9, REMAINDER 8) with (shard = 13);

### 3. 灌数

由以上配置可知 loadWorkers 等于 12，即启用12个线程灌数：

bash ./runDatabaseBuild.sh props.pg

### 4. 压测

执行指令：

./runBenchmark.sh props.pg

### 5. 测试详细数据

benchmarksql 各并发情况的输出汇总

### 6. 测试结果

|  |  |  |
| --- | --- | --- |
| 并发数 | klustron tpmC | klustron tpmTOTAL |
| 100 | 31228.21 | 69511.49 |
|  | | |
| 200 | 30496.0 | 67647.7 |
|  | | |
| 300 | 29453.5 | 65057.59 |
|  | | |
| 400 | 21099.42 | 46817.33 |
|  | | |
| 500 | 21394.99 | 47496.24 |
|  | | |
| 600 | 21328.02 | 47413.76 |
|  | | |
| 700 |  |  |
|  | | |
| 800 |  |  |
|  | | |

### 7. 问题说明

服务器配置：

华为云服务器8C32G

1. 并发数700-800时候出现瓶颈，原因为MySQL社区版中一个bug（如下图所示），目前泽拓科技（深圳）有限责任公司正在修复中，待修复后提供100-800并发连接数的压力测试数据

