# 概述

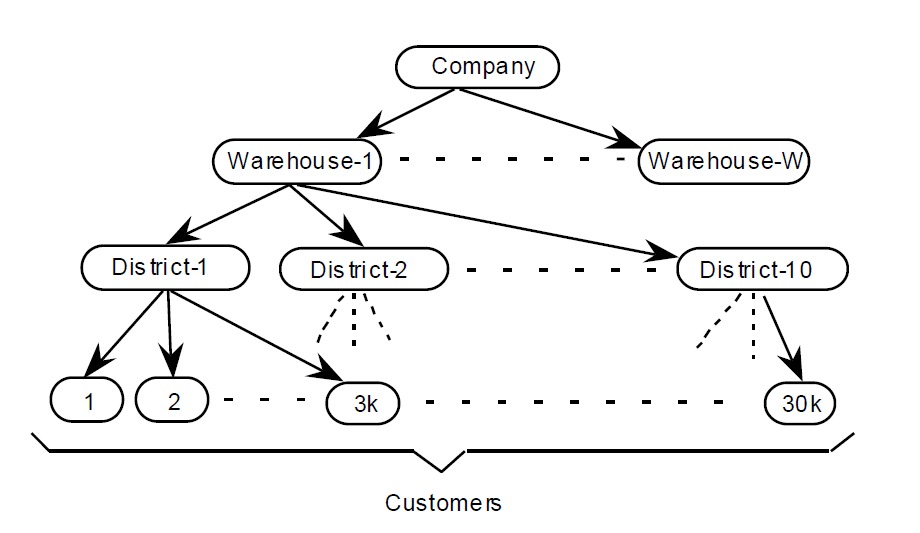
TPC-C是专门针对联机交易处理系统（OLTP系统）的规范，一般情况下我们也把这类系统称为业务处理系统。

tpcc-mysql是percona基于TPC-C(下面简写成TPCC)衍生出来的产品，专用于MySQL基准测试（事务处理能力、ACID验证）。其源码放在launchpad上，用bazaar管理。

项目地址：<https://code.launchpad.net/~percona-dev/perconatools/tpcc-mysql>

**TPC-C业务数据模型：**

tpc-c模拟的是一个在线零售公司，加入以一个仓库为例，仓库对10W种商品进行销售，具备针对用户进行水平扩展的能力，即建立更多的仓库。每个仓库负责10个区域，每个区域有单独的订单系统，每个区域管理3000个顾客，因此一个仓库负责3W个客户。树状图如右图：



TPC-C业务涉及到的9张表以及ER图介绍：

1）ITEM商品信息表：10w条商品信息，保持不变

2）warehouse仓库表：按需库容，比如上图表示有W个仓库，则有W条记录

3) Stock库存表:每个仓库有10W条商品的库存信息，因此总数目为W\*10w

4）district区域表: 每个仓库管理10个区域，因此有W\*10条记录

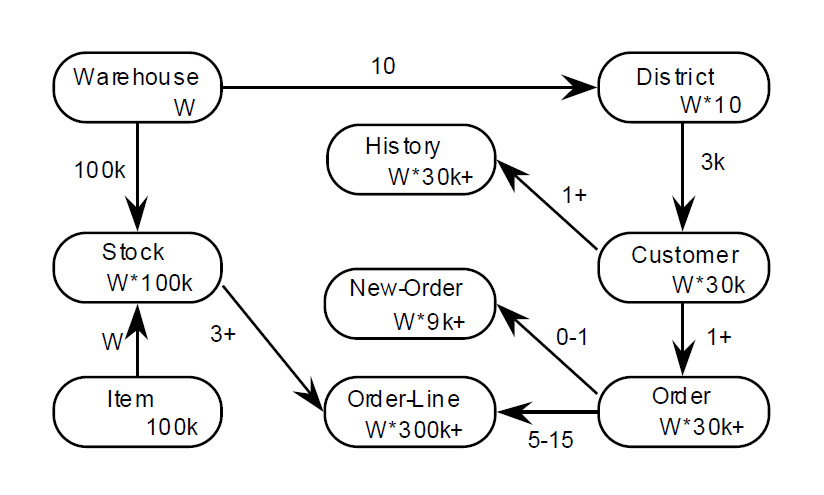
5）custoer客户表：每个仓库负责10个区域，每个区域管理3000个客户，因此客户数为W\*3w

6 ) Order订单表：每次客户下单会生成一条记录，会持续增长，不删除，初始化为每个客户一条订单，因此初始值为W\*3w

7）New-Order新订单表：没有发货的订单，发货后即删除，初始值为每个仓库9000条记录，因此为W\*9000

8) order-line订单明细表：每个订单会购买5-15件商品（平均为10），对于每件商品都要记录到这里，因此它的数目约为Order的10倍，会持续增长，不删除，初始值为W\*30w

9) history表：历史信息表，没有主键，不需要查询，每次支付的时候生成一条记录，初始值为W\*3w条



# 操作步骤

# 应用

## TDSQL

## GoldenDB

### 操作系统

#### 开启内存巨页

echo 20 >/proc/sys/vm/nr\_hugepages

echo 4194304>/proc/sys/kernel/shmall

#### 关闭swap

1、fdisk -l | grep swap #查找swap分区

2、swapoff /dev/mapper/rhel-swap

3、vim /etc/fstab注释下面的行防止重启后swap开启

/dev/mapper/rhel-swap swap swap defaults 0 0

#### 进程绑核

尽量保证在一个NUMA节点上：

Proxy绑核：

1. 通过numactl --hardware查看
2. 停止进程：dbmoni -stop
3. 绑核：numactl --physcpubind=64-96 dbstart

DB绑核：

1. dbmoni -stop
2. numactl --physcpubind=0-63 mysql.server start --log-slave-updates=OFF
3. dbstart启动dbagent

### 组件

#### Proxy

Proxy配置项修改：

log\_level=1

route\_instance\_number=10

update\_grade=1

select\_grade=1

judge\_active\_trans\_num=500

exec\_thread\_num=40

netstat\_checkperiod=1800000

shat\_report\_interval=0

stat\_report\_interval=0

stat\_proxy\_topnsql\_number=0

enable\_stmt\_flag=1

thread\_poll\_size=5

connect\_timeout=5000

read\_timeout=5000

#### DB

数据库配置文件my.cnf：

enable\_binlog\_gtmgtid\_index=OFF

performance\_schema=OFF

innodb\_buffer\_pool\_size=128G

innodb\_buffer\_pool\_instances=64

innodb\_io\_capacity=10000

innodb\_io\_capacity\_max=100000

innodb\_lock\_wait\_timeout=30

innodb\_flush\_neighbors=0

max\_binlog\_size=1048576000

thread\_pool\_size=64

thread\_pool\_stall\_limit=50

slave\_preserve\_commit\_order=0

lock\_wait\_timeout=60

#### DBAgent

binlog\_backup\_interval=0

#### GTM

WriteIncFileMode=2

## OceanBase