

CS213 Project 3 Report

薛哲

SID: 12311012

1 引言

本项目旨在从性能与功能两方面出发，对 openGauss 与 PostgreSQL 进行对比研究。

2 性能比较

2.1 配置

2.1.1 硬件配置

本次测试中使用的华为云服务器配置为

- 规格名称: kc1.large.2
- vCPUs: 2vCPUs
- 内存: 4GiB
- CPU: Huawei Kunpeng 920 2.6GHz
- 基准/最大带宽: 0.8 / 3 Gbit/s
- 内网收发包: 30万PPS
- 系统: openEuler 20.03 64bit with ARM
- 系统盘: 通用型SSD 40GiB

若无特殊说明，下文中用到的服务器均为上述配置。

2.1.2 软件配置

openGauss 采用当前最新版本 6.0.0，安装过程参考 [单节点安装](#) | [openGauss文档](#) | [openGauss社区](#)。

PostgreSQL 采用当前最新版本 17，由于 openEuler 下暂无安装包，编译安装过程参考 [从源码编译安装 PostgreSQL 16.x](#) | [青蛙小白](#)。

为了确保 openGauss 与 PostgreSQL 互不影响，二者被分别装在两台服务器上。

2.2 测试标准

[TPC-C](#) 通过对处理仓库订单中的各个流程进行模拟，通过衡量每分钟处理订单数（tpmC），达到对数据库进行实际场景测试的效果。

这里使用 [BenchmarkSQL](#) 进行 TPC-C 测试。为了避免测试程序对数据库性能造成影响，同时更好地模拟数据库服务器独立的情况，这里额外使用服务器作为测试服务器，型号为 c6s.large.2，操作系统为 CentOS 8.2 64bit。配置过程主要参考了 [BenchmarkSQL性能测试\(openGauss\) - 知乎](#)，由于本次测试没有用到多块磁盘，跳过了其中调整表结构的步骤。

2.3 初步测试

测试使用的配置文件如下：

```
db=postgres
driver=org.postgresql.Driver

conn=jdbc:postgresql://<ip_address>:<port>/<database>

user=<username>
password=<password>

warehouses=20
loadWorkers=4
terminals=6
runTxnsPerTerminal=0
runMins=5
limitTxnsPerMin=0

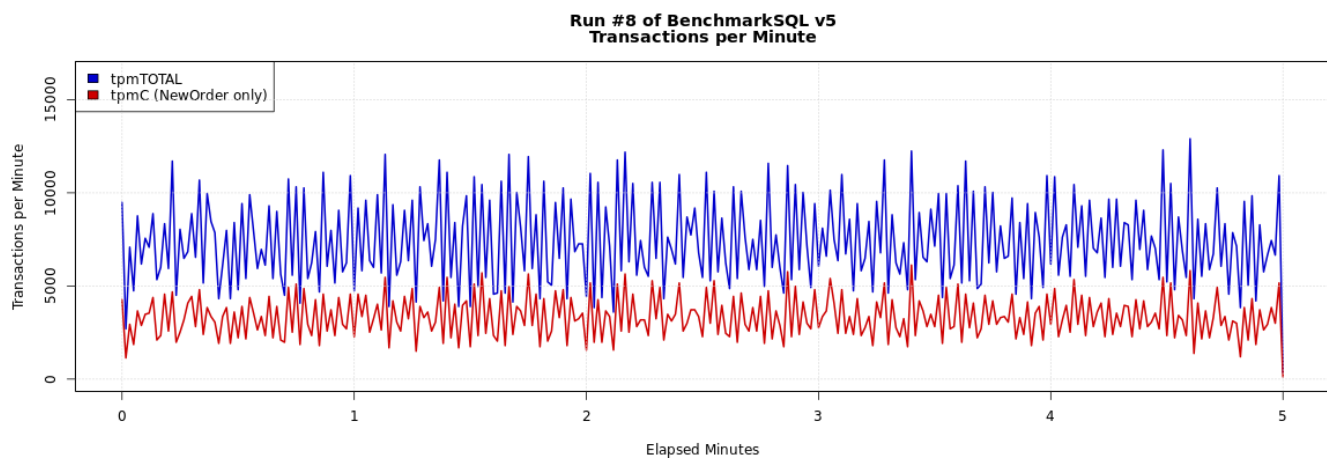
terminalWarehouseFixed=false

newOrderWeight=45
paymentWeight=43
orderStatusWeight=4
deliveryWeight=4
stockLevelWeight=4

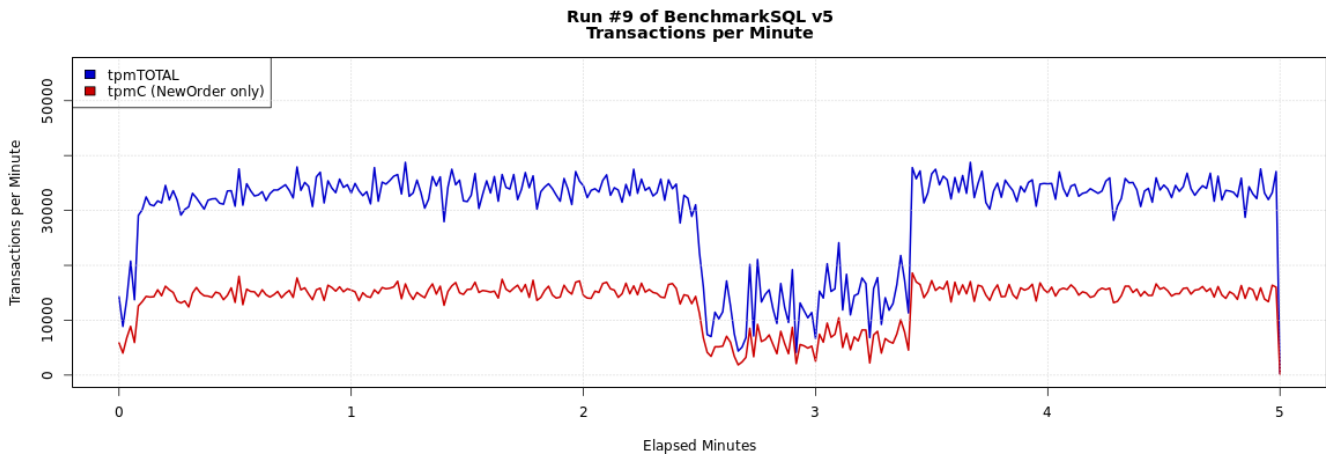
resultDirectory=my_result_%tY-%tm-%td_%tH%M%S
osCollectorScript=./misc/os_collector_linux.py
osCollectorInterval=1

osCollectorSSHAddr=<user>@<ip_address>
osCollectorDevices=net_eth0 blk_vda
```

openGauss 初步测试结果图像如下 ([详细结果](#))：



PostgreSQL 初步测试结果图像如下 ([详细结果](#))：



可以观察到 PostgreSQL 的速度是高于 openGauss 的，在本轮初步测试中 PostgreSQL 的 tpmC 约为 openGauss 的四倍。

但 PostgreSQL 在运行过程中会突然进行大规模的磁盘 IO，导致在一段时间内速度大幅度下降，通过重复测试可以发现这并非个例。我个人尝试调整缓存大小，但并无影响，故原因暂时未知。

2.4 正式测试

此次测试前，我对比了二者配置文件 `postgresql.conf`，并调整了以下参数：

- `max_connections`：最大连接数，二者均调整至 1000。
- `shared_buffers`：内存中的缓存大小，这里取二者默认设置的最大值 128MB。

并另外调整 BenchmarkSQL 运行的线程数量，对运行结果进行比较。

线程数量	tpmC (openGauss)	tpmC (PostgreSQL)	比值
1	2550.60	4160.20	1.63
10	5569.60	17476.60	3.14
100	2182.40	11621.80	5.33

可以注意到注意到二者的 tpmC 随线程数量均呈先上升后下降的趋势，且随线程数增加，PostgreSQL 相比 openGauss 的性能优势越发明显。

根据 [知乎回答](#)，openGauss 内核源于 PostgreSQL 9.2.4。而 PostgreSQL 9.2.4 本身在 2013 年 4 月 4 日发布，经过尝试无法在相同配置的服务器中成功运行，故未做额外测试。

3 功能比较

openGauss 官网网站中提到其一大优势：易运维——基于 AI 的智能参数调优。

openGauss 文档中的 [AI 特性](#) 一节对于 openGauss 与 AI 的融合进行了详细介绍。

我本打算进行 AI4DB 部分的测试，但即使我安装了 prometheus-client 包，仍然出现了报错提示：

```
FATAL: Require dependency prometheus-client. You should use pip to install it, following:
```

```
/opt/software/openGauss-DBMind/python/bin/python3 -m pip install -r  
/opt/software/openGauss-DBMind/dbmind/../requirements-x86.txt
```

DB4AI 部分则主要聚焦于直接在数据库内进行机器学习，使得模型能够更好地直接利用数据库中的数据，这里不做额外测试。

4 结论

根据上面的比较，同取当前最新版本，在 BenchmarkSQL 的测试下，openGauss 的性能低于 PostgreSQL，但测试过程中速度的稳定性较好。

同时，相比于 PostgreSQL，openGauss 将 AI 融合进数据库中，形成了其独特优势，但 AI4DB 部分目前疑似处于不可用状态。

附录 性能测试常见问题排查

- 为什么我在数据库服务器能够通过终端工具（psql/gsql）访问数据库，并在安全组设置/防火墙中开放了对应端口，但无法在远端通过客户端访问？

需要配置服务器的监听地址以及服务器与客户端的连接规则。该过程在上面提到的 PostgreSQL 编译安装教程中提到了，但在 openGauss 的安装教程中并未强调。

修改数据库数据目录下 `postgresql.conf` 文件，设置监听地址：

```
listen_addresses = '*'
```

除此之外，还需要在数据库数据目录下的 `pg_hba.conf` 文件末尾添加一行：

```
host <database> <user> <address> <auth_method>
```

其中 `<database>` 为允许客户端访问的数据库，`<user>` 为允许客户端使用的数据库用户，`<address>` 为允许的客户端网段，`<auth_method>` 为身份验证方式，具体可参考 [PostgreSQL 文档](#)。

例：

```
host tpcc1000 jack 192.168.0.219/32 trust
```

若使用外网机器访问，建议选择更加安全的身份验证方式。

修改后需要重启数据库服务。

- 为什么我在运行测试脚本时出现了 `Invalid SCRAM client initialization` 或 `Protocol error. Session setup failed.` 的错误提示？

openGauss 与最新版 PostgreSQL 的 JDBC 驱动并不通用。openGauss 的 JDBC 驱动可以在 <https://opengauss.org/zh/download/> 中找到，PostgreSQL 的 JDBC 驱动可以在 <https://jdbc.postgresql.org/> 找到。下载后覆盖 BenchmarkSQL 目录下 `lib/postgres/postgresql.jar` 即可。建议下载后在本地留存两个驱动的副本，以便重复进行测试。

- 我正确配置了 SSH 互信，为什么没有成功收集数据库服务器系统负载？

若你使用了其他品牌的服务器，请确保配置文件中 `osCollectorDevices` 项配置正确，格式为 `osCollectorDevices=net_<net_name> blk_<blk_name>`。其中 `<net_name>` 为网卡名称，可使用 `ifconfig` 命令查出；`<blk_name>` 为数据库数据目录所处磁盘分区，可使用 `df` 命令查出。

除此之外，一些不规范的 openGauss 安装教程可能有使用 Python 3 可执行文件链接覆盖 Python 2 的步骤，请务必保证数据库服务器中 `python` 命令对应的版本为 Python 2，可以通过 `python -V` 命令查看。