# 恒辉hhbench说明文档

|  |  |
| --- | --- |
| 实施团队 | 北京恒辉信达技术有限公司研发团队 |

日期： 2017 年 1 月 11 日

# 目录

[恒辉hhbench说明文档 1](#_Toc20502)

[目录 2](#_Toc29127)

[一：描述 4](#_Toc6521)

[二：选项 6](#_Toc3778)

[初始化选项 6](#_Toc29108)

[基准测试选项 7](#_Toc10410)

[通用参数 9](#_Toc11224)

[三：使用方法举例 10](#_Toc5073)

[四：详细解说 11](#_Toc2633)

[事务处理日志 11](#_Toc21221)

[聚合的日志 12](#_Toc3959)

[每个语句延迟 13](#_Toc1860)

[客户化定制脚本 13](#_Toc1347)

# 一：描述

      hhbench是一个用于在恒辉数据库中运行基准测试的简单程序。恒辉在多个并发的数据库会话中反复运行一系列相同的SQL命令，并计算事务执行的平均速率（每秒执行的事务个数）。 恒辉默认测试的是一种基于TPC-B的松散的测试，即一个事务中包括5个SELECT,UPDATE和INSERT语句。同时允许基于开发者自己书写的事务脚本文件进行其他场景的测试。

      典型的hhbench输出如下：

transactiontype: TPC-B (sort of)  
scaling factor: 10  
query mode: simple  
number of clients: 10  
number of threads: 1  
number of transactions per client: 1000  
number of transactions actually processed: 10000/10000  
tps = 85.184871 (including connections establishing)  
tps = 85.296346 (excluding connections establishing)

      前六行显示了一些非常重要的参数。接下来显示的是已经完成的事务和预计应该执行的事务个数。这两个数值应该是相等的，如果不相等说明存在执行失败的情况（如果指定为**-T**模式，则只打印实际完成的事务数量）。最后两行显示的是每秒完成的事务数量（包含连接建立时间和不包含连接建立时间分别计算出的平均值）。

      默认的TPC-B-like事务测试需要预先设置特定的表。**hhbench**调用**-i**(初始化)选项来创建和填充这些表。在使用一个定制化脚本测试时，不需要调用-i选项初始化，但是需要对这个定制化脚本需要的表进行初始化。初始化过程类似如下：

**./hhbench.sh -i [ *other-options* ]**

**注意**

./hhbench.sh -i命令会创建四个表（pgbench\_accounts, pgbench\_branches, pgbench\_history, 和pgbench\_tellers），创建这几个表会执行对应表的删除动作。执行前应确认对应数据库中是否已经有这些表，是否会误删除。

  按照默认的比例因子1，这些表按如下行数初始化:

表名                            行数  
---------------------------------  
pgbench\_branches       1  
pgbench\_tellers            10  
pgbench\_accounts       100000  
pgbench\_history           0

可以通过指定**-s**（比例因子）参数来增加初始化时各表的行数。**-F**（填充因子）参数此时也可以同时使用。

      做了必要的设置后，就可以使用如下命令进行基准测试（不需要再使用**-i**参数）

**hhbench [ *options* ]**

      在几乎所有情况下,您将需要一些选项做出有用的测试。最重要的选项是**-c**（客户端数量）,**-t**（事务数量）,**-T**（时间限制）和**-f**（定制脚本）

      具体参数见下文的完整列表。

# 二：选项

      下面的选项分成三个不同类别，分别是使用在数据库初始化阶段，基准测试运行阶段或两个阶段都可以使用。

## 初始化选项

      hhbench接受以下命令行参数:

**-i**

调用初始化模式.

**-F *fillfactor***

使用给出的填充因子创建pgbench\_accounts，pgbench\_tellers，pgbench\_branches等表.默认值是100.

**-n**

初始化后不执行vacuum。

**-q**

日志模式切换为静默方式。即每五秒只打印一条进度信息，默认是每10万条记录打印一条日志，这样通常会有很多日志产生。

**-s *scale\_factor***

生成的行数乘以比例因子。比如，-s 100会在pgbench\_accounts表中将会生成1000万条记录。默认值是1。

**--fk**

在默认创建的表上创建外键。

**--it *index\_tablespace***

在指定的表空间上创建索引。

**--ts *tablespace***

在指定的表空间上创建默认表。

**--ut**

创建日志临时表

## 基准测试选项

      hhbench接受以下命令行基准测试参数参数:

**-c *clients***

模拟的客户数量,具体指并发运行的数据库会话数。默认值为1。

**-C**

每个事务都新创建一个连接来执行，而不是仅使用一个客户端连接。这个对于测试连接过载是有意义的。

**-d**

打印调试信息

**-D *varname*=*value***

定义一个客户定制脚本适用的变量。允许多个-D选项。

**-f *filename***

从指定文件中读取事务脚本。具体参见下面的详细介绍。-N, -S和-f是互斥的。

**-j *threads***

指定工作线程数量。在多核场景下使用多于一个线程是有意义的。由于每个线程会被分配相同数量的客户端数目，所以客户端数量必须设置为线程数目的整数倍。该参数默认值为1。

**-l**

记录每个线程的耗时到日志文件中。详细内容见下文。

**-n**

测试运行前不执行vacuum。如果运行的是一个不包含默认表（ pgbench\_accounts, pgbench\_branches, pgbench\_history和pgbench\_tellers.

）的定制测试时，这个选项是必要的。

**-N**

不更新表pgbench\_tellers和pgbench\_branches.设置后会避免更新着两个表。但是这样返会使该测试与TPC-B测试产生差异。

**-P *sec***

每秒显示一次进度报告。报告包括已运行时间，上次报告到当前的tps、事务延迟时间、标准误差。节流（-R）模式下，误差是从事务计划开始时间计算，而不是实际的事务开始时间，因此，这个时间还包含了平均执行计划延迟时间。

**-r**

基准测试运行完成后报告每个命令的平均延迟时间（从客户角度看到的执行时间）。

**-R *rate***

按执行速率执行事务，而不是尽可能快的运行（默认方式）。这个速率指定的是每秒的事务数。如果指定的速率比最大可能速率还大，则这个速率参数不会对结果产生影响。

**-s *scale\_factor***

在hhbench报告中输出定制的比例因子。内置测试不需要设置这个参数。通过计算在pgbench\_branches表中的行数，正确的比例因子将会被检测到。但是，如果通过-f运行客户定制的基准测试，比例因子会被报告为1，除非使用了这个参数。

**-S**

运行select-only事务测试而不是TPC-B-like测试。

**-t *transactions***

每个客户端运行的事务数量，默认值为10.

**-T *seconds***

运行测试这么多秒,而不是每个客户端运行固定数量的事务。-t - t是互相排斥的。

**-v**

运行测试前先对四张基准表做vacuum。如果不指定-n也不执行-v，默认对表pgbench\_tellers和pgbench\_branches做vacuum并将pgbench\_history表truncate掉。

**--ai *seconds***

汇总时间间隔长度。只有-l和这个参数同时使用才有效。日志包含每个间隔的概要（事务数量，最大最小时延和两个额外的用于方差估计的字段）

**--sr *rate***

抽样速率，将数据写入日志时使用，用于降低生产的日志数量。如果设置了这个参数，只有指定的事务部分才会被记录。1.0表示所有事务都被记录。0.05表示只有5%的事务会被记录。当处理日志文件时记得考虑采样率。例如，当计算tps值，你需要乘上对应的数量（比如0.01采样率，你实际只是获得了1/100的实际tps）。

## 通用参数

**hhbench**接受以下命令行常见参数：

**-h *hostname***

数据库服务器主机名称

**-p *port***

数据库服务器端口号

**-U database**

用来选择登陆的数据库名称

-V

**--version**

打印hhbench版本

**-?**

**--help**

显示hhbench命令帮助信息

# 三：结构说明

### 1：说明

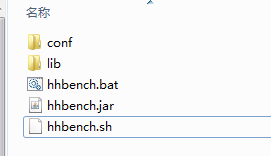
1）：conf文件夹连接数据库配置信息

2）：Lib文件是需要的库文件

3）：Hhbench.jar程序代码包

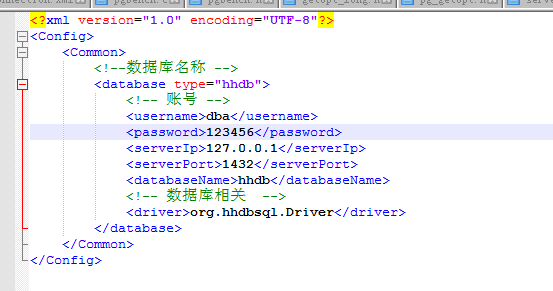
4）：Hhbench.bat window系统下运行此bat文件

5）：Hhbench.sh linux系统运行此sh文件



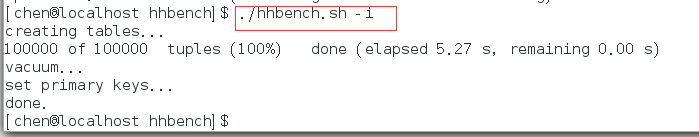
# 四：使用方法举例

测试之前在conf目录下寻找connection.xml文件配置数据库信息：

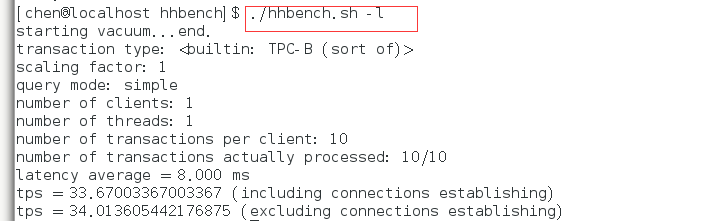


注意：使用方法举例都是在linux操作系统至上，如需要window操作请使用hhbench.bat。使用之前记得启动数据库

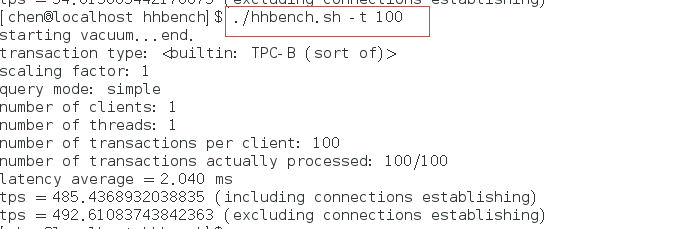
使用 -i 命令：



使用 -l 命令：

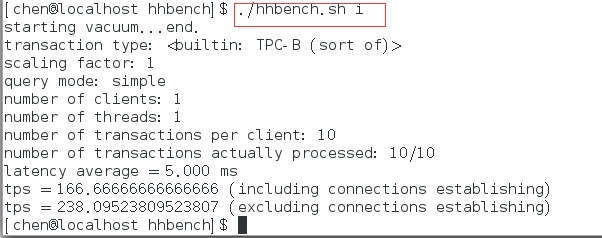


使用 -t 命令：



注意：如果使用不存在的命令则会被忽略。比如使用 i 命令，i是不存在的。

输入./hhbench.sh i 忽略 i ，等于输入./hhbench



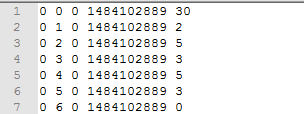
# 五：详细解说

## 事务处理日志

      使用**-l**参数但没有使用**-ai**，hhbench会将每个事务消耗的时间写入到一个日志文件中。这个日志文件会被命名为**hhbench.nnn**，nnn是hhbench进程的PID，如果通过**-j**指定的参数大于等于2，每个hhbench进程会有多个工作线程。每个工作线程都会有单独的日志文件。第一个工作线程的日志文件名称与进程的日志文件名称相同，其他的日志文件会被命名为**hhbench\_log.**nnn**.**mmm，mmm是一个从1开始的序号。

**log的格式如下：**

客户id+事务id+脚本id+事务完成的UNIX时间戳+毫秒偏移量

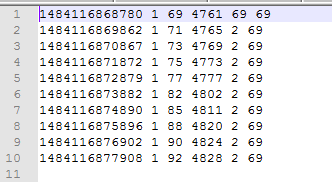


## 聚合的日志

      指定--ai参数，日志使用不同的格式：

**格式如下:**

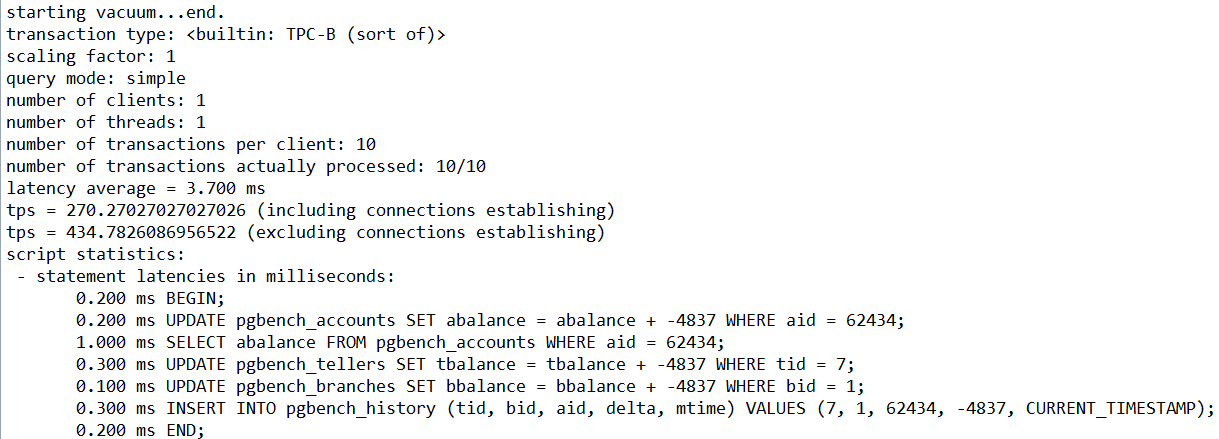
开始时间（UNIX时间戳）+执行的事务个数+时延总和+时延平方时间总和+事务的时间最小值+事务的时间最大值



## 每个语句延迟

 使用**-r**参数，hhbench会收集每个客户端的每条sql语句的执行时间。基准测试完成后，依赖于每个语句的时延，会输出这些统计项的平均值。

      对于默认的事务脚本，输出类似如下:



如果指定了多个事务脚本文件，会安装事务脚本分别输出上述统计项的平均值。

注意收集这些额外的时间信息，每个语句都会增加一些额外开销。这些会降低平均执行速率和命令的TPS。对速率和TPS的影响随平台和硬件的不同而不同。分别统计包含和不包含时延的报告是用来评估这些额外开销影响是否严重的好方法。

## 客户化定制脚本

      通过使用-f参数读取文件替换掉默认的事务脚本，hhbench可以实现客户化基准测试场景。在这种情况下,一个“事务”指的是一个SQL脚本文件。你甚至可以指定多个SQL脚本(多个-f参数),在这种情况下,每次客户端会话会随机选择一个脚本开始一个新事务。

      脚本文件格式是每一条SQL命令独占一行。一行多个SQL命令是不支持的。

      hhbench有一个简单的脚本文件的变量替换功能。变量可以使用上面提到的-D参数设置。除了通过-D参数设置的变量，还有一些变量是默认设置的。-D设置的变量优先于预设变量。一旦设置后，一个变量的值就可以通过**：变量名称**的方式插入到一个SQL命令中。当超过一个会话运行时，每个会话都可以有自己的一组变量。