站在POSTGRESQL的角度看MYSQL 8

由Kaarel Moppel于2019年2月发表

首先，我并不想要在这里引起争议，只是试图开拓大家的视野，并收集一些想法（也许我错过了很酷的东西，但MySQL毕竟是最常用的开源RDBMS），尽管很难正确、客观的来对两者进行比较。此外，我并没有对两者的性能进行比较，而是在可用功能，一般查询体验和文档清晰度等方面进行了比较，因为这些对于初学者来说是最重要的。我只是列出了一些要点，并没有按照特定的顺序来阐述。

声明：我上次使用MySQL是在10年前的个人项目，所以基本上我是从零开始花了一天半的时间来了解它的—因此，如果你看到博客中有惊人的错误请在评论区里指出，我会进行修改。另外，我在这篇文章中可能更倾向于支持Postgres ...但我很确定一位对MySQL优缺点非常了解的MySQL资深人士也可以写出关于Postgres的类似文章，所以我希望你能把这种偏见放在一边来学习关于任一系统的优缺点。

我使用官方的Docker镜像来运行MySQL 8.0.14。对于MySQL，默认的InnoDB引擎是有意义的。

docker run --rm -p 3306:3306 -e MYSQL\_ROOT\_PASSWORD=root mysql:8

“Mysql”CLI（与“psql”相同）和一般查询体验

\*当服务器需要密码时为什么不直接询问密码呢？

mysql -h 0.0.0.0 -u root   # adding '-p' will fix the error

ERROR 1045 (28000): Access denied for user 'root'@'172.17.0.1' (using password: NO)

\*与“psql”相比，tab补全非常糟糕。使用“mycli”则更有帮助。我99％的时间都花在 CLI-s,上，所以这是必不可少的。

\*用于列出表格，视图，函数等的快捷方式助手很少...

\*不能永久设置为“扩展输出”（列、行），仅可设置为“自动”和“每个查询”。

\*连接时不需要指定要连接的数据库—我认为它实际上是值得肯定的，因为很容易忘记这些数据库名称，一旦进入数据库，就可以调用“show databases”来显示数据库名称。

\*没有“generate\_series”函数...这看起来似乎是一件小事...但是在尝试生成一些测试数据时会产生相当大的代价（在时间意义上）。似乎在github上有一个替代函数，但你需要先创建一个表，所以这与“generate\_series”函数不同。

\* CLI帮助包含网络链接，例如 “help select;”在文字描述的最后会显示网址：[http://dev.mysql.com/doc/refman/8.0/en/select.html”](http://dev.mysql.com/doc/refman/8.0/en/select.html”。这很好。)  这很好。

\*如果SQL脚本中有错误，“mysql”立即停止执行，但是“psql”将会继续执行除非设置“-v ON\_ERROR\_STOP=1”。我认为“mysql”的默认停止在这里更好。

\*没有SQL标准“TABLE”语法支持。这是一个很好的快捷方式，在Postgres中我会在测试功能/查看配置或“system stats”表时使用它。

\* MySQL有索引/优化器提示，这有利于指导用户的查询语句。Postgres决定不实现此功能，因为当数据量变化或添加新的/更好的索引时，如果查询未更新，也会导致问题。Postgres有一个扩展（通常）。

\*缺失一些速记型转换（如Postgres中的“::”）。这确实是一件小事，但是很多小事会积累成一件大事。

\*缺少与“pgbench”等同的工具。这是一个我个人非常欣赏的Postgres的工具，小而简单，给快速测量服务器性能和重负载下的操作系统行为带来便利。

MySQL的积极发现

\*更多配置选项（548 vs 282），允许获得可能的更好的性能或特定操作。虽然这是一把双刃剑。

\*线程实现，为非常大量（数以百计）的并发用户提供更好的总体性能。

\*良好的JSON处理功能，例如数组范围索引器：“$[1 to 10]”和JSON Path。

\*“performance\_schema”中的更多的性能指标视图/表。尽管不知道里面的信息有多么大用处。

\*有一个官方集群产品选项（商业）。

\*内置表空间和WAL加密支持（Postgres需要第三方支持）。

\* MySQL工作台，一个用于查询和数据库设计的GUI工具，比“pgadmin3 / 4”更好（并且在视觉上更好）。还有一个具有更多功能的商业版本（备份自动化，审计）。

MySQL的缺点

\*对于初学者来说，MYSQL似乎比Postgres更复杂—对于像MyISAM这样的不同存储引擎来说，有很多选项和扩展。 这些选项也不是坏事，但请记住在这里是对于初学者来说。

\*文档一次提供太多细节，很难阅读—将一些角落的东西（旧版本的异常等）移到单独的页面上会好得多。也许，从好的方面来说：文档实际的数量几乎增加了2倍，因此，如果遇到一些奇怪的问题，你有更高的机会找到相关的解释。

\*从文档中看来，除了错误修正之外，还经常将一些功能添加到MySQL的小版本中...这使得Postgres用户会感到困惑。

\*不太符合SQL标准。基于我谷歌搜索的来源至少有：1,2,3。

\*导入和导出数据。有一些等同于COPY但更复杂的东西（一些特定的授权和配置设置涉及加载位于数据库服务器上的文件），用于导入数据的单独工具，称为“mysqlimport”。我在文档中还发现了一个有趣的地方，是关于事务行为中的一些隐式更改的，这取决于加载数据的方式：

使用LOAD DATA LOCAL INFILE，数据解释和重复键错误将成为警告，操作将继续，因为服务器无法在操作过程中停止传输文件。

\*在尝试了解查询速度慢的原因时，EXPLAIN提供的信息价值比较低。也没有EXPLAIN ANALYZE —这有点令人失望，因为“trace”的解决方法已经有点神秘了。 “EXPLAIN FORMAT = JSON”提供了更多细节来估算成本。

\*全文搜索有点不够成熟。内置配置似乎只适用于英语，并且没有词干（Postgres提供15种使用最广的西方语言开箱即用）。

\*某些大小限制似乎是随意的（表空间大小为64TB，InnoDB日志文件为512GB [我假设为WAL]）。Postgres将这些留给OS / FS（虽然单个表/分区大小限制为32TB）。

PostgreSQL架构/概念/平台优势

\* 100％全部ACID，无例外。 MySQL8在这方面已经有了很大的改进，但还不是很好，比如DDL。

\*更高级的扩展系统。MySQL也有一个插件系统，但不是通用的，例如在Python中启用存储过程。

\*更多不同的索引类型可用（6 VS 3）—例如，可以为正则表达式搜索索引字符串，并且有大数据的有损索引。MySQL似乎也不支持部分索引。

\*更简单的备用副本构建/管理。 从PG10 +开始，它是复制主机端的单个命令，没有特殊的配置组设置。

\*同步复制支持。

\*在功能、SQL标准兼容性和存储过程方面更接近Oracle。还有一些扩展添加了一些Oracle的字符串/日期处理函数等。

\*开箱即用的几种认证选项。尽管MySQL也有LDAP和可插拔的身份验证。

\*更高级的并行查询执行。从Postgres，Postgres在并行查询开发方面已经有几年了，MySQL刚刚从最新的8.0.14提供基本的（select count(\*) from tbl) ）支持。

\* JIT（即时）编译，例如用于元组提取和过滤的“定制机器代码”。大量节省数据仓库和其他行密集型查询。

MySQL架构/概念/平台优势

\*多个存储引擎。Postgres中也有类似的机制。

\*由于使用基于“UNDO”的行版本控制模型而减少了膨胀。然而，Postgres正在实现类似功能。

\*与进程相比线程在高会话数量时会提升性能。

\*内置支持多主（多主模式）复制。仍有一些警告（CAP定理仍然存在），并且很少有人真正需要这个功能，但绝对让人放心的是它在“内核”中—对于Postgres而言，是由第三方扩展提供类似功能，但据我所知有计划将其纳入“内核”。

\*内置“事件调度”。Postgres再次需要第三方扩展或自定义C代码来进行调度。

\*“REPEATABLE READ”是默认的事务模型，在开箱即用的整个事务中提供连续的读取，从而节省了新手RDBMS开发人员的负担。

我在MySQL中发现的奇怪的地方

\*聚合中的表别名可能会破坏查询：

mysql> select count(d.\*) from dept\_emp de join departments d on d.dept\_no = de.dept\_no  where emp\_no < 10011;

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '\*) from dept\_emp de join departments d on d.dept\_no = de.dept\_no  where emp\_no <' at line 1

\*无法从系统目录中找到内置例程列表。经过一些搜索后发现：

为了便于维护，更倾向于只在MySQL手册中记录。 好吧，这说的过去，但为什么不创建一些目录视图，至少可以记录函数名称并执行类似“\ df \* terminate \*”的操作。这在Postgres非常方便。

\*始终需要指定索引名称！我个人会把命名交给Postgres，因为生活证明，执行命名策略是非常困难的，即使团队由一个开发人员组成（是的，我是在说我自己）。

\* TIMESTAMP的最小值为'1970-01-01 00：00：01.000000'，更慷慨的DATETIME以'1000-01-01 00:00:00'开头，但不知道具体时区...

\* VARCHAR的有效最大长度取决于最大行大小（65,535字节，在所有VARCHAR列之间共享）。

\*MySQL不支持设定NULL -s，这有些奇怪...因为15年前的SQL标准2003就支持这种设定了。顺便说一句，默认的为NULL值升序模式与Postgres相反，Postgres支持NULLS LAST。这与SQL标准中未指定的那部分有关。

\*没有FULL OUTER JOIN。当然，这个子句很少使用，但大多数“竞争对手”都有，如果有LEFT JOIN等，这个应该不难实现。

\*只有“嵌套循环”JOIN，并且有一些变化。Postgres还有“哈希”和“合并”JOIN，这在加入数百万行时有很大帮助。

我在MySQL中发现的非常奇怪的地方

\* CAST（）函数不支持所有数据类型。例如，在声明表时“int”类型可用，但是：

mysql> select cast('1' as int) x;   -- will work when cast to 'unsigned'

ERROR 1064 (42000): You have an error in your SQL syntax

\*某些DDL（例如删除表）不是事务化的！当从尚未提交的事务声明表时，新表立即对于其他事务可见（尽管为空）。MySQL不足够ACID。

\*可以声明CHECK约束但是会被默默忽略！

\*使用较短的REFERENCES语句声明的FOREIGN KEY-s（在列定义的末尾）不会被强制执行，并且当引用的表/列丢失时甚至没有报错！需要使用更长的FOREIGN KEY + REFERENCES语句。

\*“无论在何种SQL模式下，截断TEXT类型数据末尾的多余空格总会产生警告。”即使是在默认的“严格模式”下，数据也会被默默截断。索引是可以截断数据的（Postgres中也可以这样做），但是数据不可以。

我希望Postgres可以引进的MySQL功能

\*隐式会话变量。PG中是使用“set”/ set\_config（）+ current\_setting（）函数这种繁琐的方式实现的。Postgres在这方面有一个补丁，但是没把它加到内核里。

"select @a := 42; select @a;"

\*内置“空间”支持。 尽管MySQL GIS功能虽然没有Postgres的PostGIS功能强大，但是可以在配置文件中配置，并且有官方的支持，这会让更多人了解它并且为潜在的需要GIS平台的开发人员提供保障，从而帮助整个Postgres项目。

\*生成列。目前在Postgres中，需要使用视图，但有些工作已经在做了。

\*用于优先处理/限制实例中的某些工作负载（用户）的资源组。目前只能管理CPU。

\*“X Protocol”插件。一个相对较新的东西，它允许来自单个会话的异步调用！

\*当行改变时，自动更新TIMESTAMP 列（使用ON UPDATE CURRENT\_TIMESTAMP子句）。 在Postgres中，类似的功能仅适用于初始化INSERT，否则需要使用触发器。

\*单个“SHOW STATUS”SQL命令，可以很好地概述服务器事件和常规查询操作的全局服务器状态，适用于连接，Aborted\_connects，Innodb\_num\_open\_files，Bytes\_received / sent，“admin commands”计数器，对象创建/删除计数器，pages\_read /写入，锁等。对于Postgres，它只能用于连续的pg\_stat \*监控和/或连续日志文件解析。

\* RESTART（也是SHUTDOWN）— 一个停止并重启MySQL服务器的SQL命令。它需要SHUTDOWN权限。

\*真正的集群化（索引组织的）的表（主键实现）。在Postgres中，集群化仅在很短的时间内起作用。

\*包含一个非常简单的工具，可以为服务器和客户端自动生成SSL证书。

\* Fres 8.0.14版本允许帐户具有双密码，指定为主密码和辅助密码。这样可以顺利实现密码淘汰。

我对MySQL 8 vs PostgreSQL 11的看法

首先，这篇文章的想法不是打击MySQL— 最新的MySQL 8版本中展现了很多进步。从发布说明来看，解决了很多问题，并且新增了很多很酷的功能（例如CTE-s，Window 函数），使其更适合企业级。与Postgres（根据www.openhub.net）相比，源代码的修改更为活跃，即使对于PostgreSQL咨询人员来说有点难以承认—它的使用量更大，并且具有非常好的前景，由于稳固的财务支持，这对Postgres来说是个问题，因为它并非真正归任何公司所有（这在其他方面是一件好事）。

但总而言之 - 目前（当然还需要了解很多PG知识）我仍然会推荐Postgres给99％需要关系型数据库的用户。由于本机多支持，剩余的1％用户是可能需要全局启动扩展的情况。在其他方面，PostgreSQL更轻巧，更易于理解（是的，这意味着选择更少），更重要的是保持数据完整性：如果设置约束（检查约束，外键约束）很少会发生数据丢失或被破坏的情况。 使用MySQL，需要开发人员始终保持警惕...但是正如我们所知，人们会在时间压力下遗忘，忙碌，采取捷径—这些事情可能会在快捷方式实现之前困扰您几年。Postgres还具有更先进的扩展功能，例如100多个外部数据包装器，可满足最苛刻的数据集成需求。

Kaarel Moppel：过去11年来，我一直对数据库感兴趣，过去7年专门研究PostgreSQL。 我仍然对全球友好社区的强大功能和快速发展感到惊讶。在业余时间，我喜欢踢足球和旅行。