注：对sys"初相识"重要的是需要认识到：performance\_schema是sys的数据来源。不过：描述无细节则结论无意义。我们是否能够给出一些实际例子说明两者关系？。这一问题又将涉及如何解读从sys中获得的性能数据。因此，在本次实训我们通过自行构造实际例子，并对之进行详细具体解释和说明，达到真正初步地认识sys及其与performance\_schema关系的目的。

## 第7章　sys系统库初相识

## 7.1 sys系统库使用基础环境

总结一些要点：

0. sys库是通过**视图的形式把information\_schema 和performance\_schema结合起来**，查询出令人更容易理解的数据。

注：视图不是数据表，可简单理解为一个复杂但可读性更强的select语句结果。早在学习sysbench压测时，已经接触过sys视图。

文本

描述已自动生成

1.要启用performance\_schema：

图片包含 图形用户界面

描述已自动生成

“sys系统库的大部分功能才能正常使用”

2. 使用sys系统库本身来启用所有需要的instruments和consumers

3. 使用存储过程CALL sys.ps\_setup\_reset\_to\_default(TRUE)可快速恢复performance\_schema的默认配置

4. 查询或多或少都会对性能开销有一定的影响

## 7.2　sys系统库初体验

注：本节“初体验”用于了解sys的基本作用。书中举了一些例子，包括系统版本、IO性能、sys库视图、函数、存储过程的整体信息。

#查看**sys 系统库**和MySQL Server的**版本号**(注：视图、函数和存储过程可理解为内置的脚本)。直接使用sys系统库下的视图进行查询，例如使用version视图可以：

mysql> USE sys;

mysql> SELECT \* FROM version;

表格

描述已自动生成

对比：

图形用户界面, 文本

描述已自动生成

#在sys系统库下包含很多视图，它们以各种方式对performance\_schema表进行聚合计算展示。这些视图大部分是**成对出现**的，两个视图名称相同，但有一个视图是带“x$”前缀的，例如：host\_summary\_by\_file\_io(注：host\_summary\_by\_file\_io是按照主机进行汇总统计的文件I/O性能数据)和x$host\_summary\_by\_file\_io，代表按照主机进行汇总统计的文件I/O性能数据，两个视图访问的数据源是相同的，但是在创建视图的语句中，不带“x$”前缀的视图显示的是相关数值**经过单位换算后的数据**（单位是毫秒、秒、分钟、小时、天等），带“x$”前缀的视图显示的是原始的数据（单位是皮秒）。

# 使用x$host\_summary\_by\_file\_io视图汇总数据，显示未格式化的延迟时间（单位是皮秒），没有“x$”前缀的视图输出的信息**经过单位换算可读性更好**。

mysql> SELECT \* FROM host\_summary\_by\_file\_io;

图形用户界面, 应用程序

描述已自动生成

# **带“x$”前缀**的视图显示原始的数据（单位是皮秒），**对于程序或工具的获取及使用更易于数据处理**

mysql> SELECT \* FROM **x$**host\_summary\_by\_file\_io;

表格

中度可信度描述已自动生成

#要查看sys系统库对象定义语句，可以使用适当的SHOW语句或INFORMATION\_SCHEMA库查询。例如，要查看session视图和format\_bytes（）函数的定义，可以使用如下语句：

mysql> SHOW CREATE VIEW session;

mysql> SHOW CREATE FUNCTION format\_bytes;

表格

描述已自动生成

注：语句文本是经过格式化的，可读性比较差。

#使用mysqldump和mysqlpump工具**导出sys系统库**

注：mysqldump和mysqlpump用**于**迁移和备份数据库，后者更快。关于mysqldump的详细介绍，可参考本书第50章(MySQL主流备份工具之mydumper详解)。

[root@localhost～]# mysqldump --databases --routines sys> sys\_dump.sql

文本

描述已自动生成

注：原命令没有考虑root设置了密码。如果只是部分备份，因此会警告备份涉及了全局事务ID（GTIDs）,可通过设置--set-gtid-purged=OFF避免。留意mysqlpump对该选项的使用。

[root@localhost～]# mysqlpump **sys**> sys\_dump.sql

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

然后用编辑器打开：

文本

描述已自动生成

#重新导入sys系统库(注：此步过程略)

[root@localhost～]# mysql < sys\_dump.sql

注：如书中指出，这里只是希望通过mysqldump和mysqlpump提取sys系统库里面的信息。但实际内容不全。

可以访问sys系统库开发网站：https://github.com/mysql/mysql-sys上的各个．sql文件。

例(注：分别为两次数据的整合)：以视图sys.io\_by\_thread\_by\_latency为例，其参考手册(5.7)网址如下：

https://dev.mysql.com/doc/refman/5.7/en/sys-io-by-thread-by-latency.html

其中一些关键字段(即字段名未能完全表达其含义)的信息：

user: 对于前台线程为对应的账号名称，后台线程则为线程名称。

total: 关于该线程的I/O事件总次数

min\_latency(max\_latency，avg\_latency)：线程的最小(最大, 平均)的I/O事件等待时间。

根据以上解释，结合如下实际数据，

表格

描述已自动生成

可知：

1. 其中有一个root用户的I/O需求明显比另一root用户的I/O需求大。原因在于之前以此账号执行过I/O压力测试。

2. buf\_dump\_thread占用了较多的I/O，因此总延时较大。

buf\_dump\_thread介绍：

首先解释“BUFFER POOL”这个概念，回顾innodb架构：

图示

描述已自动生成

用“SHOW ENGINE INNODB STATUS\G”命令查看：

图片包含 文本

描述已自动生成

可以看到它的状态：

文本

描述已自动生成

作为缓冲，自然可知内存中的缓存池buffer pool能有效提高数据读写性能。buf\_dump\_thread的作用是把整个缓存池的内容转储(dump)到硬盘上，一般会选择关闭系统时进行转储（innodb\_buffer\_pool\_dump\_at\_shutdown | ON），然后启动时又重新装载(innodb\_buffer\_pool\_load\_at\_startup | ON)：

图形用户界面, 文本, 应用程序

描述已自动生成

而负责转储的线程正是buf\_dump\_thread，根据设置(innodb\_buffer\_pool\_filename| ib\_buffer\_pool)，可以查到转储文件的大小和更新信息：

文本

描述已自动生成

以上是对视图sys.io\_by\_thread\_by\_latency的内容进行的一种解读。另一方面，可从备份文件中获取sys.io\_by\_thread\_by\_latency的定义如下：

图片包含 QR 代码

描述已自动生成

显然如此复杂的SQL查询需要保存起来以便反复执行和持续优化。从内容可知实际上视图io\_by\_thread\_by\_latency来源于对performance\_schema中的表threads, events\_waits\_summary\_by\_thread\_by\_event\_name的数据整合。

比如，user字段首先将判断threads表中每个线程是前台还是后台线程。

图片包含 QR 代码

描述已自动生成

视图中这里实际调用"isnull"判定“PROCESSLIST\_ID”是否为NULL，并以此区分前台与后台线程，这里以1号线程(后台)和61号线程(前台)为例：

文本

描述已自动生成

文本

描述已自动生成

又比如，表events\_waits\_summary\_by\_thread\_by\_event\_name里面包含了许多事件总等待时长(SUM\_TIMER\_WAIT)为0，以及许多与I/O无关的事件，因此视图需要过滤这些无关内容：

图片包含 QR 代码

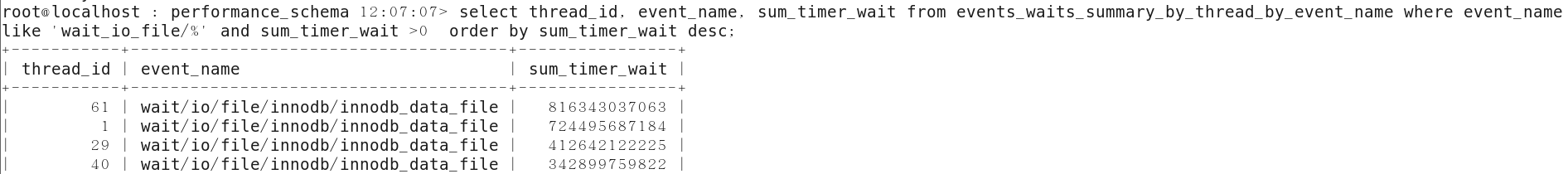
描述已自动生成

并且按总等待时间排序：

图片包含 QR 代码

描述已自动生成

我们可以直接对表events\_waits\_summary\_by\_thread\_by\_event\_name进行简单而原始的查询：



可知61号前台线程I/O量较大。不过，显然sys中的视图能更好地满足了我们对性能问题排查的实际需求。

注：以上内容参考了如下材料

1.https://dev.mysql.com/blog-archive/mysql-dumping-and-reloading-the-innodb-buffer-pool/ (MySQL Dumping and Reloading the InnoDB Buffer Pool)

2.https://cloud.tencent.com/developer/article/1828772 （详解MySQL中的Buffer Pool）

3.https://zhuanlan.zhihu.com/p/65811829 (InnoDB Buffer Pool详解)

4.https://dev.mysql.com/doc/dev/mysql-server/latest/buf0dump\_8h.html#a8c13e3f568429605d4dbe86cf2d73e32 (buf0dump.h File Reference, 函数buf\_dump\_thread说明)

要求(必做)：请参考上例的格式和方式，自行选择除上例之外的其他视图，自行补充总结最好在前面多次实验的环境下有真实数据统计结果。

## 7.3　sys系统库的进度报告功能

注：该部分内容也参考了6.4.2。此处内容说明，我们可以利用sys获取一些不能单纯从performance\_schema得到的信息。

#从MySQL 5.7.9开始，sys系统库视图提供了查看长时间运行的事务的进度报告，通过processlist和session以及带“x$”前缀的视图进行查看，其中processlist包含了后台线程和前台线程当前的事件信息；session不包含后台线程和command为Daemon的线程。

processlist

session

x$processlist

x$session

注：前面的实验我们都是用show processlist查看会话信息，

表格

描述已自动生成

信息都是从sys.processlist中筛选，其中书中所说的” command为Daemon的线程” (比如压缩gtid\_table线程，见实训2)就不会显示：

表格

描述已自动生成

此部分实验需要一些慢卡情景，比如sysbench压测之类，我们可以自由地在sysbench库中执行一些SQL语句。不过前提是这些语句应该有一定耗时。所以我们最好把原来压测后的表(sbtest1-5，压测后会有几百行)删掉：

图片包含 文本

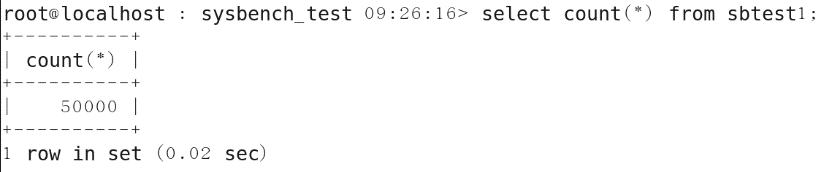
描述已自动生成

然后重新初始化为较大的表：

文本

描述已自动生成

我们在mysql统计一下sbtest1的行数：



然后对应地在sys.session中(也包括sys.processlist等)会有相关记录，但进度信息是空的：

表格

低可信度描述已自动生成

#**启用需要查看与进度相关的instruments**

stage/sql/Copying to tmp table

stage/innodb/alter table (end)

stage/innodb/alter table (flush)

stage/innodb/alter table (insert)

stage/innodb/alter table (log apply index)

stage/innodb/alter table (log apply table)

stage/innodb/alter table (merge sort)

stage/innodb/alter table (read PK and internal sort)

stage/innodb/buffer pool load

#对于不支持进度的stage事件，或者未启用所需的instruments或consumers的stage事件，则对应的进度信息字段值为NULL

注：根据以上要求执行如下语句：

UPDATE setup\_instruments SET ENABLED='YES', TIMED='YES' where name like 'stage/innodb%' or name like 'stage/sql/Copying to tmp table';

图片包含 表格

描述已自动生成

还需要启用consumers的stage事件，执行如下语句：

update setup\_consumers set enabled='yes' where name like '%stage%';

文本

中度可信度描述已自动生成

执行测试语句：

mysql>alter table sbtest1 add index i\_c(c);

在语句没有执行完毕之前，通过session查看语句的进度：

表格

描述已自动生成

中间结果略，能看到刚才提交的语句的进度：

表格

描述已自动生成

等语句执行完毕后，progress字段会又变为空：

表格

描述已自动生成