本文红字标明部分是自己不明白的地方.要是有人知道,请告知下, o(∩\_∩)o

**Innotop 详解**

**Part 1:安装**

**1:**在ubuntu下面可以直接apt下来,不需要自己再去折腾安装

**2:**软件包安装:

下载:wget http://nchc.dl.sourceforge.net/sourceforge/innotop/innotop-1.6.0.tar.gz

http://code.google.com/p/innotop/downloads/list

采用CPAN来安装:  
perl -MCPAN -eshell  
install Term::ReadKey  
install DBI  
install DBD::mysql  
然后  
# perl Makefile.PL  
# make   
# make install

官网: http://www.xaprb.com/blog/2006/07/02/innotop-mysql-innodb-monitor/

**3:**说明:目前innotop 有2个版本.一个是1.6和1.8

我apt下来的是1.6,软件包安装的是1.8,使用下来他们最大的区别在:

1.8可以远程监控其他服务器的状态,而1.6则不行,(本人使用下来需要**修改/root/.innotp 的配置文件**)不知道正确不.他们的界面是一样的,1.6的apt结束之后需要:

You have not defined any database connections.

Choose a name for the connection. It cannot contain whitespace, colons or

semicolons.

Enter a name: **192.168.1.171**

Typical DSN strings look like

DBI:mysql:;host=hostname;port=port

The db and port are optional and can usually be omitted.

If you specify 'mysql\_read\_default\_group=mysql' many options can be read

from your mysql options files (~/.my.cnf, /etc/my.cnf).

Enter a DSN string: **DBI:mysql:test;host=192.168.1.171;port=3306**

Optional: enter a table (must not exist) to use when resetting InnoDB deadlock information:

Do you want to specify a username for 192.168.1.171?: **y**

Do you want to specify a password for 192.168.1.171?: **y**

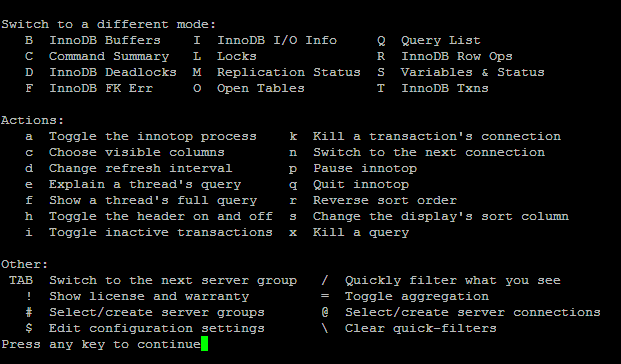
Enter username for 192.168.1.171: **root**

Enter password for 'root' on 192.168.1.171:**…..**

Save password in plain text in the config file?: **y**

配置好后直接可以用innotop 进入监控状态了,而1.8 则需要innotop -ummm -p123456 -h192.168.1.8 进入监控状态

**4:**说明:



**Part 2:**

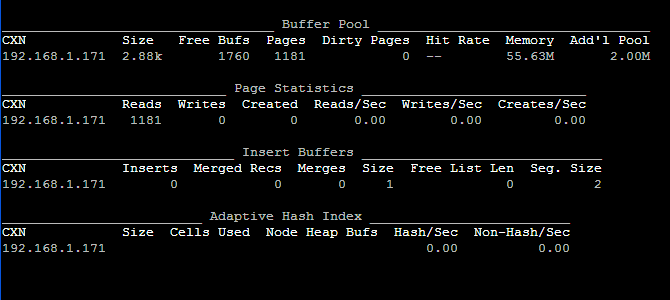
**Switch to a different mode:**

各种模式的切换(注意大写键) Alt+大写字母

**B: InnoDB Buffers**

该模式显示有关InnoDB Buffer Pool，页面统计，插入缓冲，自适应哈希索引。这些数据来自show innodb status;

该模型包含了“buffer\_pool“，”页面统计“，”插入缓冲器“和”adaptive\_hash\_index“的默认表。



**Buffer Pool:**

Size : 某sql使用Buffer Pool的大小

Free Bufs ：Innodb\_buffer\_pool\_pages\_free的值,空页数;

Pages: Innodb\_buffer\_pool\_pages\_data的值, 包含数据的页数(脏或干净)。Dirty Pages ：Innodb\_buffer\_pool\_pages\_dirty的值,当前的脏页数

Hit Rate: 命中率

Memory : Innodb\_buffer\_pool\_size 的值.

Add'l Pool : innodb\_additional\_mem\_pool\_size的值

**Page Statistics (页面统计)**

Reads：Innodb\_pages\_read 的值,读取的页数

Writes：Innodb\_pages\_written 的值,写入的页数

Created： Innodb\_pages\_created 的值,创建的页数

Reads/Sec ：每秒读取的页数

Writes/Sec : 每秒写入的页数

Creates/Sec ： 每秒创建的页数

**Insert Buffers(插入缓冲)**

Inserts ：表示执行insert 次数 （这个是个人猜想的）

Merged Recs : 表示执行 insert 索引树合并的次数

Merges： insert 语句合并的次数

Size Free List Len ：不详

Seg. Size：　不详

**Adaptive Hash Index(**自适应哈希索引**)**

Size:

Cells Used:

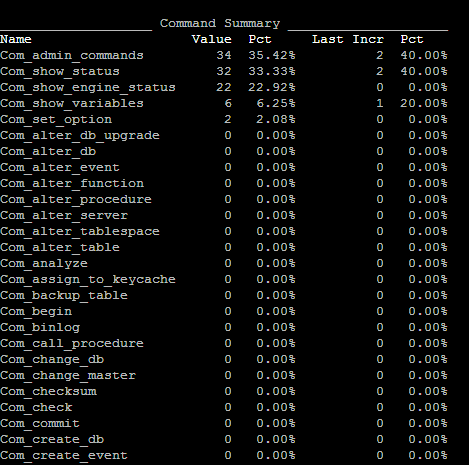
Node Heap Bufs:

Hash/Sec:

Non-Hash/Sec:

**C:** **Command Summary (show Global status)**

命令摘要,显示了cmd\_summary,从变量中提取建立STATUS\_VARIABLES,变量必须是数字, 按降序排序和比较,查看相关的参数的话,先按s 再输入相关的参数的前缀名;

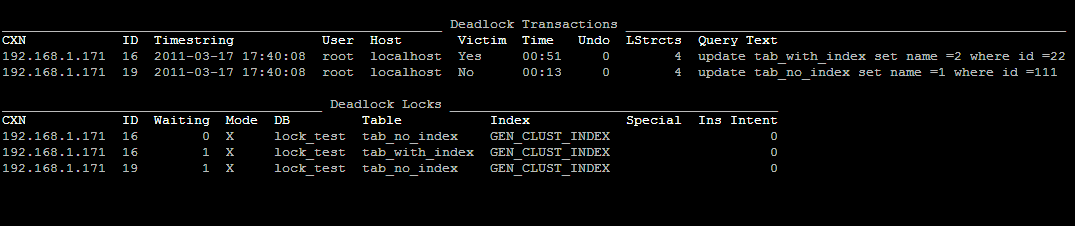


按s键:输入了innodb则:



**D: InnoDB Deadlocks**

显示innodb产生的死锁,和产生死锁的语句



**Deadlock Transactions:**

ID:连接线程ID号

Timestring : 死锁发生的时间

User: 用户名

Host:主机

Victim:Yse表示这个死锁SQL被强行终止了.No表示这个SQL正在执行

Time : 这条SQL的运行时间

Undo ： 回滚

LStrcts： 不详

Query Text : 死锁执行的SQL

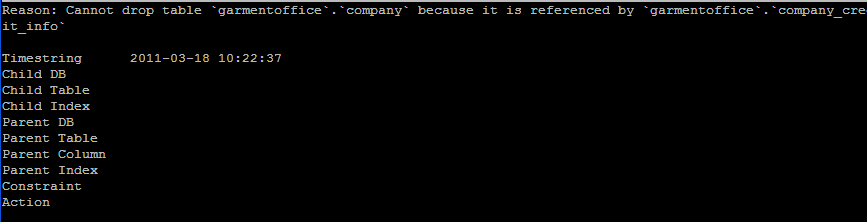
**F: InnoDB Foreign Key Errors**

此模式显示最后InnoDB的外键的错误信息

正常状态:

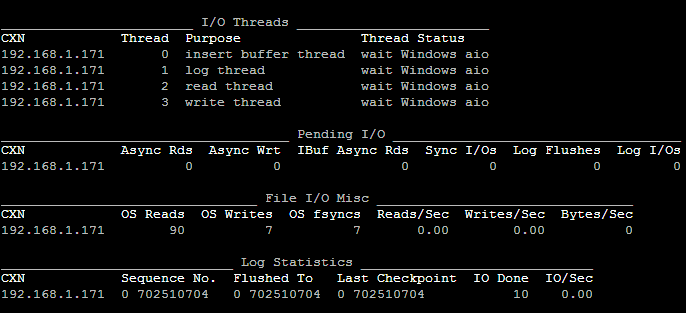


出现问题状态:



**I: InnoDB I/O Info**

这种模式显示InnoDB的I / O统计，包括I / O线程，挂起的I / O，文件I / O，日志统计



**I/O Threads(io线程)**

Thread :线程ID

Purpose:作用说明

Thread status 线程状态

**Pending I/O(挂起的IO)**

Async Rds : Asynchronous Reads 异步读

Async Wrt: : Asynchronous Write 异步写

IBuf Async Rds: : Innodb Buffer Asynchronous Reads 缓冲池的异步读

Sync I/Os: : Asynchronous Input/Output:的异步

Log Flushes : log Flushes

Log I/Os : Input/Output 异步写入/读出日志

**File I/O Misc(文件IO)**

OS Reads: 系统I./O的读

OS Writes: 系统.I/O的写

OS fsyncs: 不详

Reads/Sec: 每秒多少次读

Writes/Sec: 每秒多少次写

Bytes/Sec: 每秒多少字节的输入输出

**Log Statistics(日志统计)**

Sequence No: Innodb buffer 里面的 LSN 号

Flushed To: 刷新到事务日志的LSN号

Last Checkpoint:: LSN的检查点

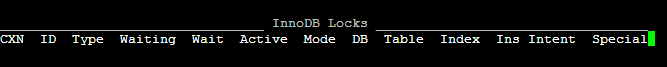
IO Done: 多少I/O已经完成

IO/Sec: 每秒I/O次数

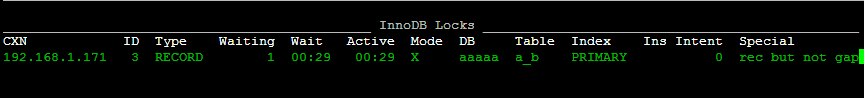
**L: Locks**

这种模式显示了有关当前锁的信息。此刻只对InnoDB锁的支持，并在默认情况下你只看到  
将锁定哪些事务正在等待。此信息来自从InnoDB的状态部分。有一个非常繁忙的服务器，你可能经常锁等待，它可以帮助能够看到哪些表和索引的“热点“的锁。

正常状态:



异常状态:



注意: 等到锁释放之后,又会回到正常状态的图

**InnoDB Locks:**

ID : 线程ID号

Type : 表示当前正在等待的状态

Waiting : 等待线程的个数

Wait : 这个线程等待了多长时间

Active ：这个线程活跃了多长时间

Mode : 锁模式,X:排他锁

DB ：当前数据库名

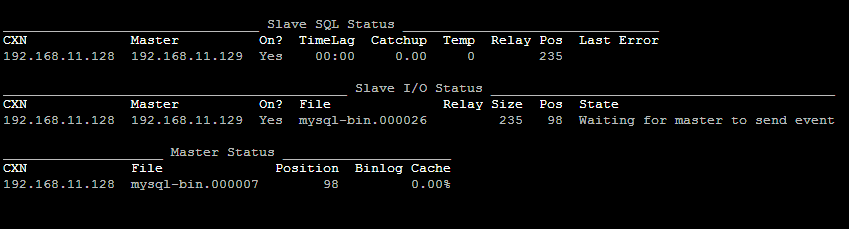
Table : 当前表名

Index : 当前索引名

Ins Intent Specia: rec but not gap 不是间隙的记录;要是空的话 就表示是间隙锁;

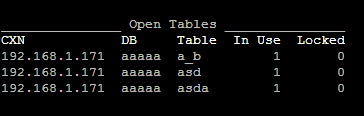
**M: Master/Slave Replication Status**

显示了SHOW SLAVE STATUS 和 SHOW MASTERSTATUS 的一些信息,

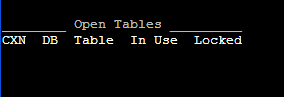


**O: Open Tables**

对小的查询不太有效果,对大的查询和X锁,很有效果(个人看法)



事件结束后,有恢复:



**Open Tables:**

DB:指当前数据库

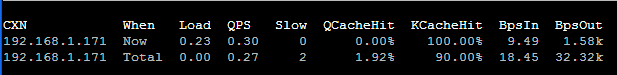
Table:指当前表

In Use : 1:在使用 （指这个表在当前时间有多少用户在访问这张表）

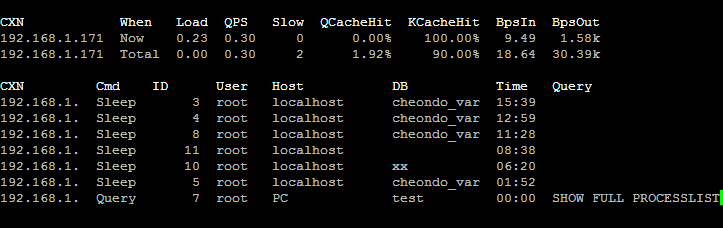
Locked : 不详

**Q: Query List**

信息:



显示完整PROCESSLIST的输出(按a 和i 键),h:隐藏



其下面有这些按键:

e键 : 按e并输入thread ID,就能显示改SQL的执行计划

f键 : 显示当前正在运行的完整SQL语句

o键 : 显示当前SQL被mysql内部优化后的SQL

**字段的解释:**

Load: 系统平均负载(特定时间间隔内运行队列中的平均进程数)

QPS : Query per Second 每秒查询的次数

Slow : 慢查询的个数 (在my.cnf里配置慢查询的秒数.只要某个select查询时间大于这个秒数,都将把这个SQL记录到慢日志文件里面)

Se/In/Up/De% ：增,删,改,查的比率.

QCacheHit : 查询缓存的命中率.

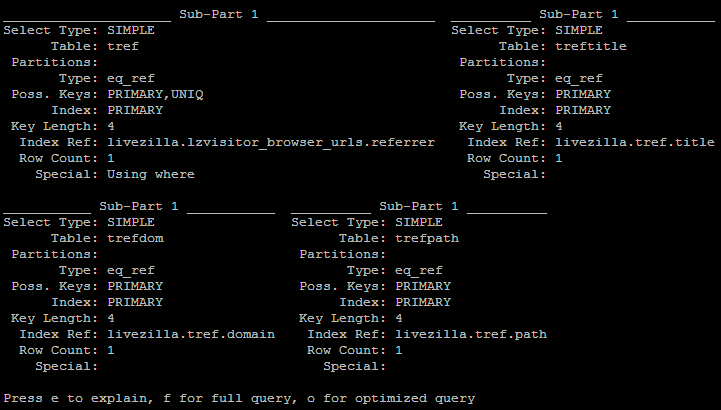
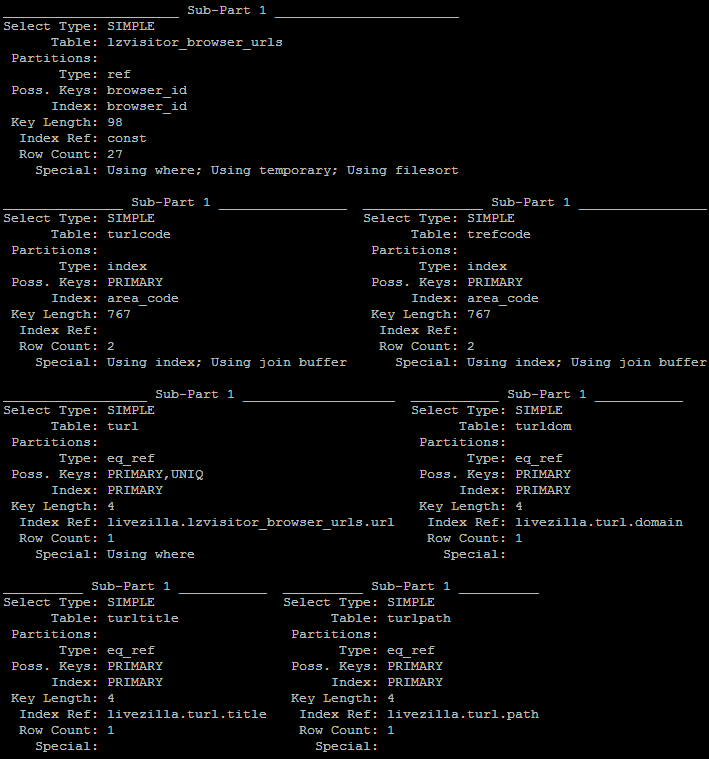
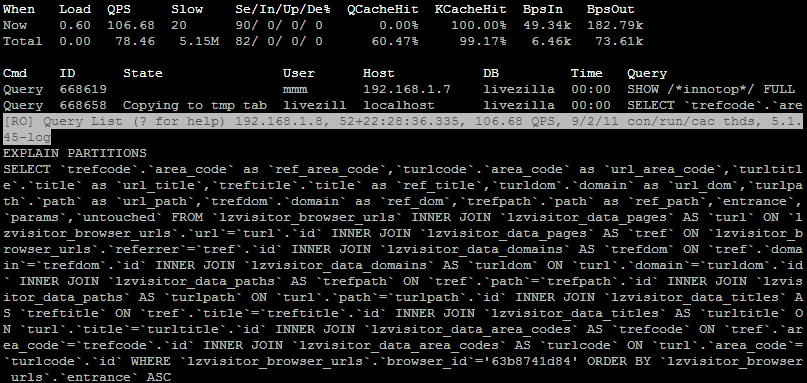
KCacheHit : 索引的命中率.

BpsIn : Bits per second 每秒写入的字节数

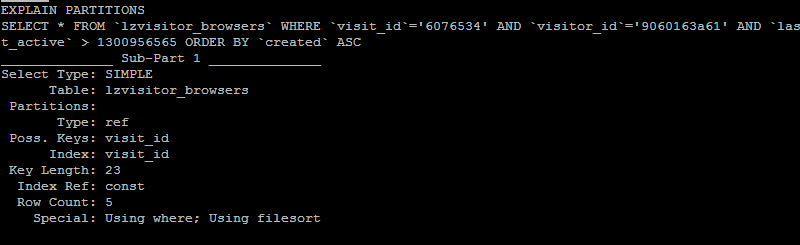
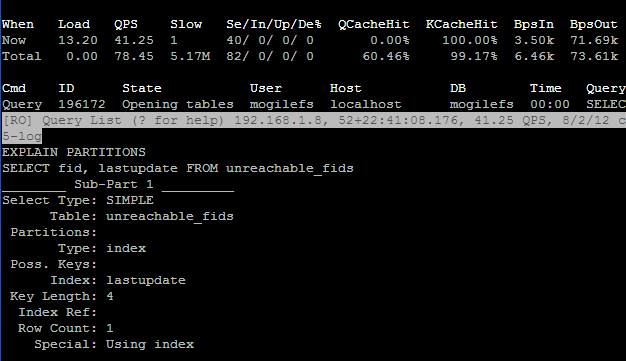
BpsOut : Bits per second 每秒读取的字节数

**如:e键:**

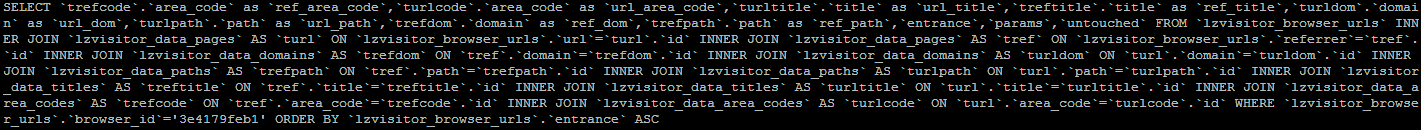
复杂的:



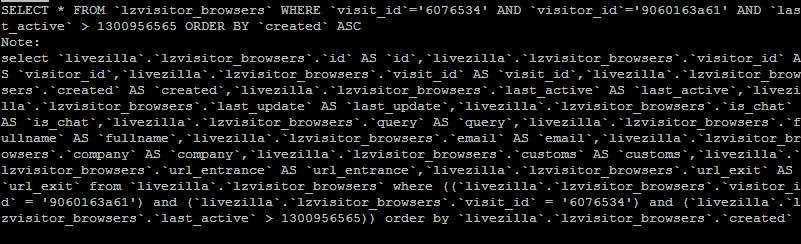
简单的:



**如:f键**

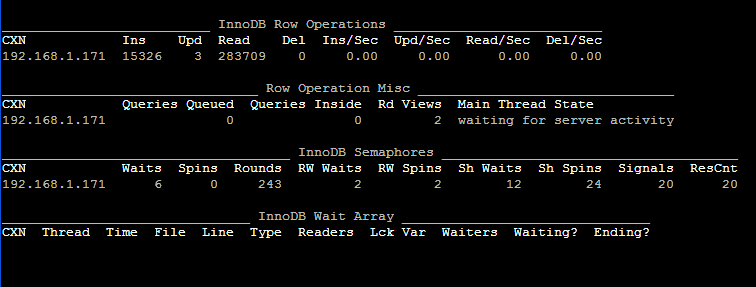


**O键:**



**R: InnoDB Row Operations and Semaphores**

这种模式显示InnoDB的行操作



**InnoDB Row Operations(InnoDB对行操作信息)**

Ins :innodb 插入的行数

upd: innodb 更新的行数

read : innodb 查询的行数

del : innodb 删除的行数

Ins/Sec：每秒输入的行数

Upd/Sec : 每秒更新的行数

Read/Sec :每秒读取数据的行数

Del/Sec: 每秒删除的行数

**Row Operation Misc(InnoDB对行操作的类别)**

Queries Queued : 查询队列

Queries Inside ： 同时有多少个SQL查询

Rd Views： 读取的视图个数

Main Thread State ：主线程的状态

**InnoDB Semaphores(InnoDB的信号)**

Waits

Spins

Rounds

RW Waits

RW Spins

Sh Waits

Sh Spins

Signals

ResCnt

**InnoDB Wait Array(Innodb 等待列表)**

Thread

Time

File

Line

Type

Readers

Lck Var

Waiters

Waiting?

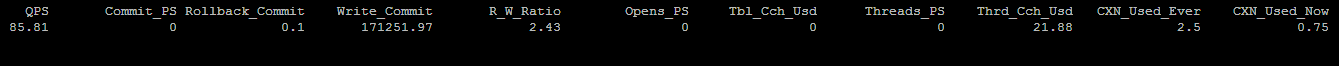
Ending?

**S: Variables & Status**

这个模块是监控mysql的变量值跟状态值的. s,g,v键分别用不同的样式来显示变量值和状态值

c键可以选择预定好的变量值或状态值的集合

e键是用来改变当前预定义好的变量值或状态值的集合

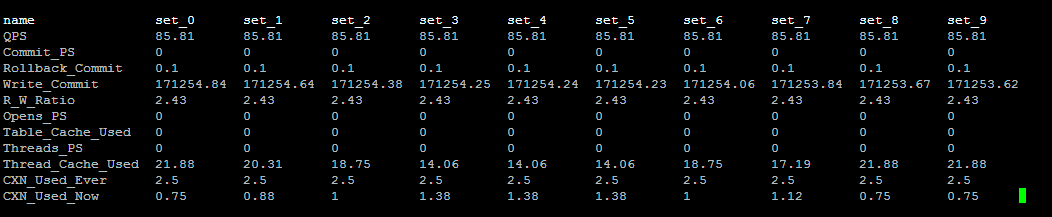


按s,g, v 切换样式默认是s上图

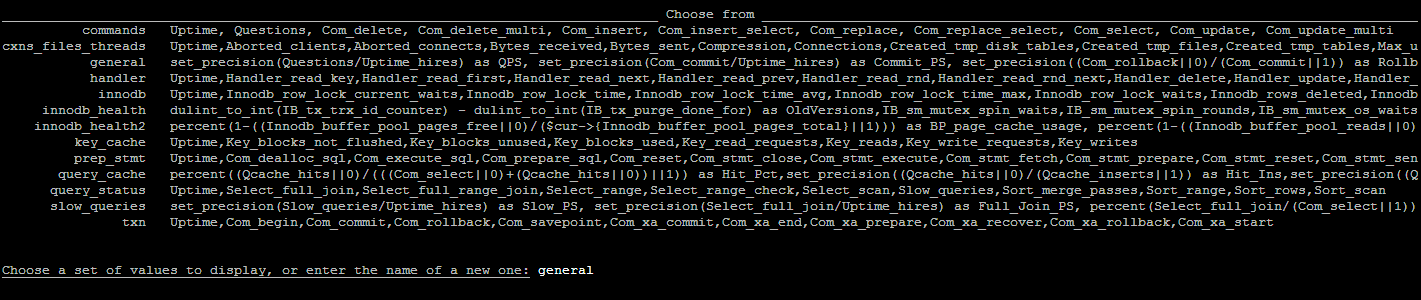
g为:



v为:

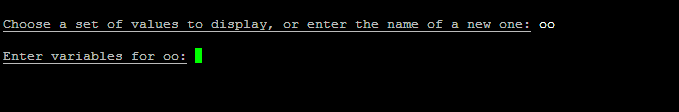


按c如下图:

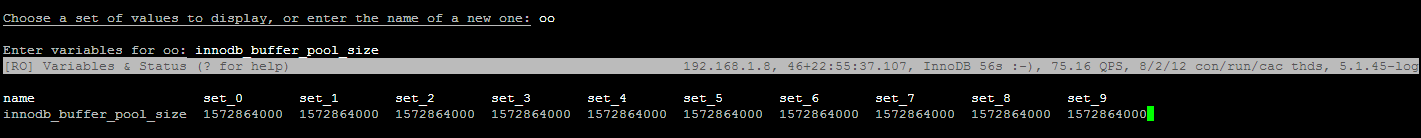


按照上面的提示选择需要查看的变量值

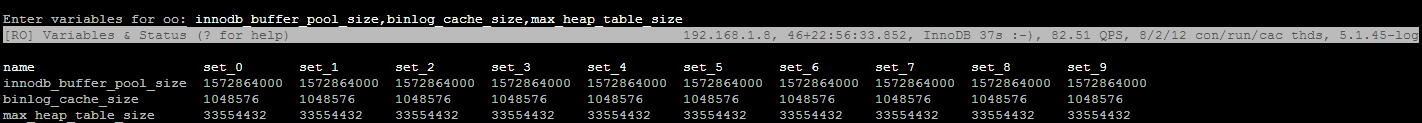
或则 自己定义一个:oo



输入需要查看的参数名称:innodb\_buffer\_pool\_size



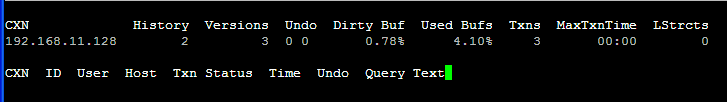
按e 继续给自己的oo添加参数



**T: InnoDB Transactions**

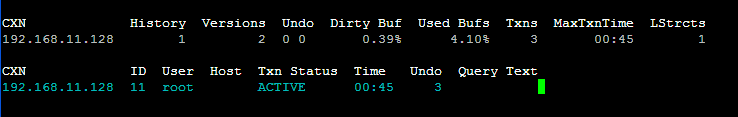
Innodb 事务信息(show innodb status)

没有事务的状态下:



有事务的状态: (手动开启了start transaction)

要是事务时间执行久的话,可以自己手动kill掉,



键k:杀死;键x:查询

可以使用a,i,e,f,o这些键来使用,和上面的意思一样

History ：历史记录?

Versions： 事务版本号

Undo：回滚

Dirty Buf ：脏数据页的使用率

Used Bufs: innodb buffer使用率

Txns : 同时运行的最大事务数

MaxTxnTime ：单个事务运行的最长时间

LStrcts:不详

**Part 3:** **Actions说明**

**a** Toggle the innotop process(切换innotop进程)

**k**  Kill a query's connection(杀死一查询的连接)  
   **c**  Choose visible columns    (可见列选择)

**n**  Switch to the next connection(切换到下一个连接)  
   **d** Change refresh interval   (更改刷新间隔)

**p**  Pause innotop (暂停innotop)  
   **e**  Explain a thread's query  (说明线程的查询)

**q**  Quit innotop  (退出)  
   **f** Show a thread's full query(显示线程的完整查询)

**r**  Reverse sort order (反向排序)  
   **h**  Toggle the header on and off(头切换和关闭)

**s**  Change the display's sort column(更改显示的排序列)  
   **i**  Toggle idle processes       (切换空闲进程)

**x**  Kill a query  (杀死一查询)

**Part 4: Other**

TAB Switch to the next server group(切换到下一个服务器组)

/ Quickly filter what you see 快速筛选

! Show license and warranty 显示许可和担保

= Toggle aggregation 切换

# Select/create server groups 选择/创建服务器组

@ Select/create server connections选择/创建服务器连接

$ Edit configuration settings 配置设置

\ Clear quick-filters 清除快速过滤器