

MySQL Expert Group

<http://meg.imysql.cn>

Linux下的MySQL数据库监控

吴诗展

MySQL专家组核心成员

<http://mysqlmeg.cn>

2008-11-1

Email:cnwsz#QQ.com

Blog: <http://www.dbaview.cn>

主要内容

- 概述
- 服务器层面
 - 被动监控
 - 健康监控
- 数据库层面
 - 被动监控
 - 健康监控
 - 同步监控
- 监控工具

概述

- 监控的重要性
 - 那在黑暗里行走的，不知道往何处去--《新约》
 - 监控是一个大而复杂的任务，不同的应用有不同的环境
 - 而且关注点不同，监控项也不同，所以会有很多不同的监控侧重点
 - 今天所讲侧重于故障监控和性能监控
- 监控思想
 - 工具会很快过时，思想远比工具重要
 - 知道花费了什么/获得了什么/投资回报率？
 - 用户体验被赋予最高优先级

服务器层面

- 被动监控
 - 树倒猢猻散
 - 如何监控?
 - 事后监控/sms/需立刻处理
- 健康监控
 - 预警型监控/如何敲定报警阈值
 - Memory:Free (关swap)
 - CPU:vmstat 1 ; load average
 - IO:iostat -x 1 (关注IOPS)

健康监控

Linux 2.6:iostat -x 1

```
avg-cpu:  %user   %nice   %sys %iowait  %idle
           2.31    0.00    0.82   3.48   93.39
```

<http://www.dbaview.cn>

```
Device:            rrqm/s  wrqm/s     r/s     w/s  rsec/s  wsec/s    rkB/s    kB/s  avgrq-sz  avgqu-sz   await  svctm   %util
cciss/c0d0         0.64    0.73   41.34  101.29   108.03   90.04    54.02   45.02    1.39     0.09    0.63   0.17   2.47
```

```
procs -----memory----- --swap--  -----io----- --system--  -----cpu-----
 r  b  swpd  free  buff  cache  si  so  bi  bo  in  cs  us  sy  id  wa
 0  0  1728  25256  44004  5751272  0  0  586  1024  0  0  2  1  93  3
 0  0  1728  25120  44004  5751340  0  0  0  284  1231  461  0  0  99  0
 0  0  1728  25120  44004  5751340  0  0  4  144  1123  234  0  0  100  0
 0  0  1728  25120  44004  5751340  0  0  4  172  1137  283  0  0  99  0
 0  0  1728  25056  44008  5751336  0  0  36  172  1132  272  0  0  99  1
 0  0  1728  25056  44008  5751336  0  0  12  304  1236  474  0  0  99  1
 0  0  1728  25056  44008  5751336  0  0  0  160  1125  256  0  0  100  0
 0  0  1728  25056  44008  5751336  0  0  0  212  1157  341  0  0  100  0
 0  0  1728  25056  44008  5751336  0  0  0  100  1150  200  0  0  100  0
```

Linux 2.6:vmstat 1

<http://www.dbaview.cn>

数据库层面

- 被动监控
 - 如何监控Mysql Crash?
 - 端口?语义? Mysql pid?

- 注:端口监控引起的连接出错问题?
 - 调大max_connect_errors=10000

数据库层面

- 数据库健康监控
 - 总体情况
 - 一些关注项
 - 同步监控

总体情况

- **mysql >SHOW /*!50000 GLOBAL */ VARIABLES;**
 - SHOW /*!50000 GLOBAL */ VARIABLES like “key%”
 - 为什么加/*!50000 GLOBAL */ ? MySQL4和MySQL5区别. 即MySQL4: SHOW VARIABLES;MySQL5 :SHOW GLOBAL VARIABLES;
 - 后继默认为MySQL 5
- **mysql> SHOW /*!50000 GLOBAL */ STATUS;**
 - j\$ mysqladmin extended (绝对值, 全局)
 - j\$ mysqladmin extended -i10 -r (相对值,每隔10秒)
- **mysql> SHOW FULL PROCESSLIST;**
 - j\$ mysqladmin processlist
 - Tcpdump 抓包
- **mysql> SHOW ENGINE INNODB STATUS;**
 - mysql> SHOW /*!50000 GLOBAL */ STATUS LIKE 'innodb%';
 - j\$ mysql -e "show innodb status"
- **mysql> SHOW MASTER STATUS; SHOW SLAVE STATUS;**

一些关注项:通用情况

- 每秒查询率**QPS: questions**增量/time

- 间隔时间参考: 10s,1h,24h

- show status like “Questions” :已经发送给服务器的查询个数(不单是select操作)

- **com_xxx** 计数器

- Com_delete/Com_insert/Com_update/Com_delete等

- show status like “Com_xxx”: 表示xxx 语句执行的次数。例如, Com_delete统计DELETE 语句执行的次数

- show **global** status like ‘com_insert’; & show status like ‘com_insert’; (Mysql5下区分)

- 事务:

- Com_rollback, Com_commit, Com_begin,等

- Com_rollback+Com_commit=?

- 读写比率:

- (select + qcache_hits) / (insert+update+delete+replace)

- QCACHE_hits:查询缓存被访问的次数。

一些关注项:通用情况

● 表缓存:**opened_tables**增量/time

- 时间间隔建议为1h或24h
- Opened_tables:已经打开的表的数量。如果Opened_tables较大, table_cache 值可能太小。
- 高峰期查看:variable:table_cache – status:open_tables

● 线程缓存:**threads_created**增量/time

- Threads_created:创建用来处理连接的线程数。如果Threads_created较大, 你可能要增加thread_cache_size值。
- 高峰期查看: variable:thread_cache_size – status:threads_cached

● 连接数: **Connections**

- variable:max_connections – status:max_used_connections
- variable:max_connections - status:threads_connected

一些关注项:查询缓存

- 查询缓存命中率

- $Qcache_hits / (Com_select + Qcache_hits)$

- `Qcache_hits`: 查询缓存命中的次数

- 查询缓存删除速率

- $Qcache_lowmem_prunes / time$

- `Qcache_lowmem_prunes`: 由于内存较少从缓存删除的查询数量。

- 增大查询缓存值 `query_cache_size` , 以减小 `lowmem` , 增加缓存命中率

一些关注项:存储引擎MyISAM

- **Key 缓存命中率**

- $1 - (\text{key_reads} / \text{key_read_requests})$
- `Key_reads`从硬盘读取键的数据块的次数
- `key_read_requests`从缓存读键的数据块的请求数
- 命中率低则调大`Key_buffer_size`

- **MyISAM 连接锁比率**

- `table_locks_waited / table_locks_immediate`
- <1% 佳, 1% 需引起注意, >3% 性能问题
- `table_locks_waited`:不能立即获得的表的锁的次数
- `Table_locks_immediate`:立即获得的表的锁的次数

一些关注项:存储引擎InnoDB

- **Cache命中率:**

- $1 - (\text{InnoDB_buffer_pool_reads} / \text{InnoDB_buffer_pool_read_requests})$
- `InnoDB_buffer_pool_reads`:不能满足InnoDB必须单页读取的缓冲池中的逻辑读数量。
- `InnoDB_buffer_pool_read_requests`:InnoDB已经完成的逻辑读请求数。
- 变量`innodb_buffer_pool_size`

- **InnoDB缓冲池**

- `InnoDB_buffer_pool_wait_free`:向InnoDB缓冲池写时, 如果需要读或创建页, 并且没有干净的页可用, 则需等待页面清空的次数。
- 该值应小, 否则需调大缓冲池`innodb_buffer_pool_size`大小。

- **日志**

- `InnoDB_log_waits`:因为日志缓冲区太小, 必须等待清空的时间
- 此值尽量小, 否则需调大`innodb_log_file_size`,

一些关注项:慢查询和临时表

- 慢查询率:**Slow_queries/time**

- Slow_queries:查询时间超过long_query_time秒的查询的个数
- select_full_join:没有使用索引的联接的数量。如果该值不为0,应仔细检查表的索引
- 配置项中开启慢查询日志以便优化: log-slow-queries = slow.log

- 临时表创建率

- created_tmp_tables/time

- 写硬盘的临时表:

- created_disk_tmp_tables / created_tmp_tables
- tmp_table_size是否足够大? 临时表目录考虑用内存替代? 查询blob/text多吗?

同步监控

- **Mysql > slave slave status\G**
- 是否工作:
 - Slave_IO_running: YES
 - Slave_SQL_running: YES
- 延迟情况:
 - Seconds_behind_master
- 更新速率:
 - Read_Master_Log_Pos增量/sec, Exec_Master_Log_Pos增量/sec
- **Binlog缓存:**
 - binlog_cache_disk_use / binlog_cache_use
 - 溢出后临时文件保存的事务数/二进制日志缓存的事务数
 - 变量: binlog_cache_size
- 注:Log_Pos会hang住?数据导入期的同步监控屏蔽?

监控工具

- 分为交互式监控工具和非交互式监控工具
- 交互监控工具特性
 - 实时的观察服务器状态
- 非交互式监控工具特性:
 - 很多非交互式监控都不是专门为MySQL而单独设计的，而是多用途的
 - 定期的检查系统状态值，对特定参数的衡量或预警，监控失败或是超过某安全值时报警或是触发某动作
 - 持续性的自动记录和保留数据库状态，监控并在web端画图
 - 在数据库越来越慢的时候，进行诊断，可以了解历史值以及以往的变化情况
 - 好的监控会在灾难来临前给出警示,并帮助你确定问题关键所在

监控工具

- 交互性监控工具
 - innotop:<http://innotop.sourceforge.net>
 - 《High Performance MySQL Second Edition》作者之一
 - Mysqlreport:<http://hackmysql.com/mysqlreport>

- mtop:<http://mtop.sourceforge.net>
- mytop:<http://jeremy.zawodny.com/mysql/mytop>

非交互式监控工具

- **Nagios:**<http://www.nagios.org>
 - 开源，定期检查并报警阈值
 - 可做plug-in运行任何程序，有现成的mysql监控的外挂
 - 缺点:有些复杂,不好管理
- **Zabbix:** <http://www.zabbix.co>
 - 配置文件和数据都在数据库中
 - 画图效果等都好于Nagios
 - 更灵活且更容易配置
- **MySQL企业版监控**
 - 企业版附带 不开源 收费
 - 优点是有很多预制的建议
 - <http://www.mysql.com/products/enterprise/advisors.html>

非交互式监控工具

- **Zenoss** :<http://www.zenoss.com>
- **Hyperic HQ** <http://www.hyperic.com>
- **OpenNMS**<http://www.opennms.org>
- **Groundwork Open Source** <http://www.groundworkopensource.com>
- **MONyog** <http://www.webyog.com>
- 基于**RRDTool**的监控系统
 - **MRTG**:<http://oss.oetiker.ch/mrtg/>
 - **Munin**:<http://munin.projects.linpro.no>
 - **Cacti**:<http://www.cacti.net>
 - **Cricket**:<http://cricket.sourceforge.net>
 - **Ganglia**:<http://ganglia.sourceforge.net>

Q&A?

<http://www.mysqlmeg.cn>

- 参考&推荐

- 《MySQL手册》
- 《深入浅出MySQL数据库开发、优化与管理维护》
- 《构建oracle高可用环境》
- 《SQL语言艺术》
- 《Uderstanding MySQL Internals》
- 《High Performance MySQL Second Edition 》
- 《Monitoring Mysql》
- 《Expert MySQL》

- 联系方式：吴诗展

Email:cnwsz#QQ.com

blog:www.dbaview.cn

MySQL专家组介绍

MySQL专家组:

英文名称MySQL Expert Group，简称MEG。MEG是由知名MySQL DBA发起，各大互联网公司DBA、开发人员组成的MySQL组织；成员具有丰富的MySQL数据库开发、管理和优化经验，具备优秀的LAMP构架设计和规划能力。

组织的宗旨:

促进MySQL技术的交流和分享，提供企业MySQL培训；推广MySQL数据库的广泛使用，致力于为不同环境下的应用提供贴身全面的解决方案。

特邀顾问:

陈吉平(Piner.chen)，淘宝网首席DBA，Oracle ACE，著有《构建Oracle高可用环境》一书。Piner在大型高可用构架设计有着丰富的经验，并且能一直保持新技术的跟进

Email:mysqlmeg@mysqlmeg.cn

网站:<http://mysqlmeg.cn>