

MySQL Monitoring

ChinaUnix网友技术交流

曹金成 **NetSeek**

MySQL Monitoring

- 一、如何有效监控?监控的目的
- 二、基本监控解决方案(基础数据收集)
- 三、高级解决方案(数据库健康状态, 优化)
- 四、**MySQL**性能相关交互式监控工具
- 五、相关资源参考

一、如何有效监控？

一、如何有效监控？监控的目的？

- 1、服务健康监测（存活，故障告警.....）
- 2、监控系统采集数据，依据数据进行调优

二、基本监控解决方案

二、基本监控解决方案(收集系统信息)

top 、 vmstat 、 iostat 、 mpstat 、 mytop 、
dstata 、 free 、 /proc/.... 、 mstat 、
mtop命令行工具分析系统资源使用状
况.

三、高级监控解决方案

三、高级监控解决方案

1、Nagios相关:(收集数据库信息及健康状态，对数据库调整优化)

check_mysql [推荐]

nagios-mysql-plugins-0.3 [适当选择]

check_mysql_health [重点介绍推荐]

* 由于时间关系本PPT只重点介绍一下Nagios相关的插件脚本。
重点介绍一下**check_mysql_health**监控数据库调优。

2、MySQL Activity Report

基于rrdtool

*<http://gert.sos.be/en/projects/mysqlar/>

*<http://gert.sos.be/demo/mysqlar/> 演示网站

三、高级监控解决方案

三、高级监控解决方案

.....

3、CACTI

4、RRD

参考资料:

* <http://vvv.k6p.de/rrd/>

* <http://oss.oetiker.ch/rrdtool/rrdworld/>

* <http://www.fi.muni.cz/~kas/mrtg-rrd/>

* <http://www.tnpi.biz/internet/manage/rrdutil/faq/mysql.shtml>

5、Munin

.....

Nagios check_mysql

```
#cd /usr/local/nagios/libexec/
```

```
# ./check_mysql --help //查看使用说明
```

-
- Usage: check_mysql [-d database] [-H host] [-P port] [-s socket]
[-u user] [-p password] [-S]
- Options:
- -h, --help
Print detailed help screen
- -V, --version
Print version information
- -H, --hostname=ADDRESS
Host name, IP Address, or unix socket (must be an absolute path)
- -P, --port=INTEGER
Port number (default: 3306)
- -s, --socket=STRING
Use the specified socket (has no effect if -H is used)
- -d, --database=STRING
Check database with indicated name
- -u, --username=STRING
Connect using the indicated username
- -p, --password=STRING
Use the indicated password to authenticate the connection
==> IMPORTANT: THIS FORM OF AUTHENTICATION IS NOT SECURE!!! <==
Your clear-text password could be visible as a process table entry
- -S, --check-slave //检测Slave状态.
Check if the slave thread is running properly.
- -w, --warning
Exit with WARNING status if slave server is more than INTEGER seconds
behind master
- -c, --critical
Exit with CRITICAL status if slave server is more than INTEGER seconds
behind master
-

Nagios check_mysql

- 示例:
- Nagios 监控服务器:192.168.169.138
- 如要监控DB服务器192.168.169.204和Slave DB服务器192.168.169.123
- 数据库授权: (登陆DB服务器, 进行授权用户名netseek,密码linuxtone)
- `mysql> grant all privileges`
- `-> on *.*`
- `-> to netseek@192.168.169.138 identified by 'linuxtone';`
- `Query OK, 0 rows affected (0.00 sec)`
- `mysql> flush privileges;`
- `Query OK, 0 rows affected (0.00 sec)`

Nagios check_mysql

- 示例:
 -
 - 在监控机上.
 - **#cd /usr/local/nagios/libexec/**
 - 连接204查看数据库状态:
 - **#!/check_mysql -H 192.168.169.204 -u netseek -p linuxtone**
 - Uptime: 2146510 Threads: 1 Questions: 61155591 Slow queries: 325 Opens: 1273
Flush tables: 1 Open tables: 767 Queries per second avg: 28.491
 - 连接数据库123查看
 - **./check_mysql -H 192.168.169.123 -u netseek -p linuxtone -S -w 60 -c 600**
 - Uptime: 35349 Threads: 1 Questions: 4022 Slow queries: 0 Opens: 38 Flush
tables: 1 Open tables: 32 Queries per second avg: 0.114 Slave IO: Yes Slave SQL:
Yes Seconds Behind Master: 0
- OK,在命令行下都能正常连接数据库:

Check_mysql Nagios配置

vi commands.cfg 添加如下:

```
#check_mysql
```

```
define command{
```

```
    command_name    check_mysql
```

```
    command_line    $USER1$/check_mysql -H $ARG1$ -P  
$ARG2$ -u $ARG3$ -p $ARG4$ //仔细看参数传递与上面的命令  
行对应.
```

```
}
```

```
#check_slave
```

```
define command{
```

```
    command_name    check_slave
```

```
    command_line    $USER1$/check_mysql -H $ARG1$ -P  
$ARG2$ -u $ARG3$ -p $ARG4$ -S -w $ARG5$ -c $ARG6$
```

```
}
```

Check_mysql Nagios配置

-
- **#vi sh-wt-www-db.cfg** 给上海网通两台DB服务器上mysql数据库检测:
- 在自己定的的配置文件里, 添加如下服务段.
- **#check_mysql**
- **define service{**
- **host_name dbss-master**
- **service_description check_mysql**
- **check_command check_mysql!192.168.169.122!3306!netseek!linuxtone**
-
- **}**

- **#check slave**
- **define service{**
- **host_name dbss-slave**
- **service_description check_slave**
- **check_command check_slave!192.168.169.123!3306!netseek!linuxtone!60!600**
-
- **}**

Nagios check_mysql

- check_mysql监控演示效果

Host ↑↓	Service ↑↓	Status ↑↓	Last Check ↑↓	Duration ↑↓	Attempt ↑↓	Status Information
dbss-master	check_mysql	OK	03-24-2009 13:22:30	0d 2h 13m 33s	1/5	Uptime: 49403 Threads: 2 Questions: 618990 Slow queries: 0 Opens: 122 Flush tables: 2 Op second avg: 12.529
dbss-slave	check_slave	OK	03-24-2009 13:22:34	0d 0h 7m 29s	1/5	Uptime: 35811 Threads: 1 Questions: 4057 Slow queries: 0 Opens: 38 Flush tables: 1 Open t avg: 0.113 Slave IO: Yes Slave SQL: Yes Seconds Behind Master: 0

模块可以很好的检测mysql replication slave的健康状态.

mysql>show slave stauts\G

- *是否工作
 - Slave_IO_running: YES
 - Slave_SQL_running: YES
- *延迟情况
 - Sencodes_behind_master

nagios-mysql-plugins

选择性的使用此插件来配合监控工作

```
# wget http://www.shinguz.ch/MySQL/nagios-mysql-plugins-0.3.tar.gz
# tar zxvf nagios-mysql-plugins-0.3.tar.gz
# cd nagios-mysql-plugins-0.3
# chmod a+x *
# mv * /usr/local/nagios/libexec
```

插件解释:

check_db_mysql.pl 检测mysql是否运行.

check_errorlog_mysql.pl 检测数据库下的错误日志

perf_mysql.pl 收集性能数据,类似后面要讲的check_mysql_health

replication相关.

check_repl_mysql_cnt_slave_hosts.pl //检测复制相关。

check_repl_mysql_hearbeat.pl

check_repl_mysql_io_thread.pl

check_repl_mysql_read_exec_pos.pl

check_repl_mysql_readonly.pl

check_repl_mysql_seconds_behind_master.pl

check_repl_mysql_sql_thread.pl

- -----

nagios-mysql-plugins

```
./check_db_mysql.pl -h 192.168.169.204 -u netseek -p linuxtone -port 3306
```

编写command

```
define command{  
    command_name    check_db_mysql  
    command_line    $USER1$/check_db_mysql.pl -h $ARG1$ -u $ARG2$ -p  
    $ARG3$ -port $ARG4$  
    ..  
}
```

服务端配置:

```
.....  
check_command      check_db_mysql!192.168.169.204!netseek!linuxtone!3306
```

```
.. check_db_mysql  OK 03-25-2009 14:53:01 1d 20h 47m 33s 1/5 Database seems up and running...
```

check_mysql_health

一、安装check_mysql_health

官方网站:<http://www.consol.de/opensource/nagios/check-mysql-health/>

```
# wget http://www.consol.de/fileadmin/opensource/Nagios/check\_mysql\_health-2.0.3.tar.gz
```

```
# tar zxvf check_mysql_health-2.0.3.tar.gz
```

```
# cd check_mysql_health-2.0.3
```

```
# ./configure --prefix=/usr/local/nagios --with-nagios-user=nagios \  
-- with-nagios-group=nagios --with-perl \  
--with-statefiles-dir=/tmp
```

```
# make && make install
```

注:**check_mysql_health**

(**check_mysql_perf**的替代方案, 官方不再支持 **check_mysql_perf**)

详细参见:<http://www.consol.com/opensource/nagios/check-mysql-perf>

check_mysql_health

二、check_mysql_health 插件使用说明

```
# cd /usr/local/nagios/libexec/  
# ./check_mysql_health --help
```

Check various parameters of MySQL databases

Usage:

```
check_mysql_health [-v] [-t <timeout>] [[--hostname <hostname>]  
  [--port <port> | --socket <socket>]  
  --username <username> --password <password>] --mode <mode>  
  [--method mysql]  
check_mysql_health [-h | --help]  
check_mysql_health [-V | --version]
```

.....

check_mysql_health

.....

Options:

- hostname
the database server's hostname
- port
the database's port. (default: 3306)
- socket
the database's unix socket.
- username
the mysql db user
- password
the mysql db user's password
- database
the database's name. (default: information_schema)
- warning
the warning range
- critical
the critical range

check_mysql_health

.....
--mode

the mode of the plugin. select one of the following keywords:

- | | |
|----------------------------|---|
| connection-time | (Time to connect to the server)
连接到服务器的时间. |
| uptime | (Time the server is running)
MySQL服务器运行的时间 |
| threads-connected | (Number of currently open connections)
数据库服务器当前打开的连接 |
| threadcache-hitrate | (Hit rate of the thread-cache) 线程缓存命中率 |

-----mysql replication 相关段-----

- | | | |
|--------------------------|--------------------------|---------------------|
| slave-lag | (Seconds behind master) | 判断slave落后于master多少秒 |
| slave-io-running | (Slave io running: Yes) | 表明Slave复制正常运行 |
| slave-sql-running | (Slave sql running: Yes) | 表明Slave复制正常运行 |

用check_mysql 模块来替代这些功能更,check_mysql是用C写的执行速度更快. --check-slave
./check_mysql -H 192.168.169.123 -u netseek -p linuxtone -S 来解决

check_mysql_health

.....

-----查询缓存相关-----

qcache-hitrate (Query cache hitrate)
查询命中率,这个比率越高则表明服务器的**SELECT** 查询性能就越好

qcache-lowmem-prunes (Query cache entries pruned because of low memory)
由于内存较小从缓存删除的查询数量
增大**query_cache_size**的值, 以减小**lowmem**,增加缓存命中率

keycache-hitrate (MyISAM key cache hitrate)n **key**缓存命中率
如果命中率低, 则调大**key_buffer_size**

-----InnoDB Cache命中率-----

bufferpool-hitrate (InnoDB buffer pool hitrate) **InnoDB** 缓冲池命中率
bufferpool-wait-free (InnoDB buffer pool waits for clean page available) **InnoDB**的缓冲池等待清理页.
log-waits (InnoDB log waits because of a too small log buffer) 因为太小**log**缓冲区导致
innodb_log等待.

check_mysql_health

tablecache-hitrate	(Table cache hitrate)	表缓存命中率
table-lock-contention	(Table lock contention)	连接锁表率
	table_locks_waited/table_locks_immediate	
	table_lock_waited:	不能立即获得的表的锁表次数
	table_lock_immediate:	立即获得的表的锁表次数.
	小于1%较优, 如果1%需要引起注意, >3% 性能问题.	
index-usage	(Usage of indices)	索引使用情况.
tmp-disk-tables	(Percent of temp tables created on disk)	临时表创建.
slow-queries	(Slow queries)	慢查询
long-running-procs	(long running processes)	长期运行的进程.
cluster-ndbd-running	(ndbd nodes are up and running)	ndbd 集群节点运行状况
sql	(any sql command returning a single number)	执行返回一个数字的任何SQL

o o o o o o o o o

告警值规则:

"10" means "Alert, if > 10" and

"90:" means "Alert, if < 90"

check_mysql_health

三、示例：

Nagios 监控服务器:192.168.169.138

如要监控**DB服务器192.168.169.204**和**Slave DB服务器192.168.169.123**

数据库授权：(登陆**DB服务器**，进行授权用户名**netseek**,密码**linuxtone**)

```
mysql> grant all privileges
```

```
-> on *.*
```

```
-> to netseek@192.168.169.138 identified by 'linuxtone';
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> flush privileges;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
OK - 2 client connection threads | threads_connected=2;10;20
```

check_mysql_health

三、示例：

在监控机上.

```
#cd /usr/local/nagios/libexec/
```

连接123查看数据库查询线程缓存命中率状态：

```
# ./check_mysql_health --hostname 192.168.169.123 --port 3306 --  
  username netseek --password linuxtone --mode threadcache-hitrate  
OK - thread cache hitrate 91.30% | thread_cache_hitrate=91.30%;90::80:  
  thread_cache_hitrate_now=91.30% connections_per_sec=0.00
```

90: 表示小于**90 warning**, **80** 表示小于**80 则critical**

查询命中率

```
# ./check_mysql_health --hostname 192.168.169.122 --port 3306 --  
  username netseek --password linuxtone --mode qcache-hitrate  
WARNING - query cache hitrate 88.19% | qcache_hitrate=88.19%;90::80:  
  qcache_hitrate_now=78.57% selects_per_sec=0.75
```

check_mysql_health

三、示例：

在监控机上。

.....
调整告警阈值：

```
# ./check_mysql_health --hostname 192.168.169.122 --port 3306 --username netseek --password  
linxstone -w 80: -c 70: --mode qcache-hitrate  
OK - query cache hitrate 88.19% | qcache_hitrate=88.19%;80::70: qcache_hitrate_now=85.63%  
selects_per_sec=0.52
```

连接123数据库查看锁表率。

```
# ./check_mysql_health --hostname 192.168.169.123 --port 3306 --username netseek --password  
linxstone --mode table-lock-contention  
OK - table lock contention 0.00% | tablelock_contention=0.00%;1;2  
tablelock_contention_now=0.00%
```

连接123数据库查看数据库服务器当前连接数量：

```
# ./check_mysql_health --hostname 192.168.169.123 --port 3306 --username netseek --password  
linxstone --mode threads-connected  
OK - 2 client connection threads | threads_connected=2;10;20
```

check_mysql_health

Nagios相关配置

vi commands.cfg 添加如下:

```
#check_health_mysql
```

```
define command{
```

```
    command_name    check_mysql_health
```

```
    command_line    $USER1$/check_mysql_health --hostname $ARG1$ --port  
$ARG2$ --username $ARG3$ --password $ARG4$ --mode $ARG5$
```

```
}
```

#vi sh-wt-www-db.cfg 给上海网通两台DB服务器上mysql数据库检测:

..... 在自己定的的配置文件里, 添加如下服务段.

```
define service{
```

```
    host_name        dbss-slave
```

```
    service_description    threads-connected
```

```
    check_command
```

```
check_mysql_health!192.168.169.123!3306!netseek!linuxtone!threads-  
connected
```


check_mysql_health

```
.....  
define service{  
    host_name          dbss-slave  
    service_description qcach-hitrate  
    check_command  
    check_mysql_health!192.168.169.123!3306!netseek!linuxtone!qcach-hitrate  
    .....
```

```
}  
  
define service{  
    host_name          dbss-slave  
    service_description keycach-hitrate  
    check_command  
    check_mysql_health!192.168.169.123!3306!netseek!linuxtone!keycach-hitrate  
    .....
```

```
}
```

check_mysql_health

- 演示效果

keycache-hitrate	CRITICAL	03-25-2009 11:31:56	0d 2h 5m 39s	5/5	CRITICAL - myisam keycache hitrate at 84.40%
load	OK	03-25-2009 11:29:54	204d 17h 53m 35s	1/5	OK - load average: 0.00, 0.00, 0.00
qcache-hitrate	CRITICAL	03-25-2009 11:32:22	0d 19h 20m 13s	5/5	CRITICAL - query cache hitrate 0.00%
table-lock-contention	OK	03-25-2009 11:29:46	0d 8h 5m 49s	1/5	OK - table lock contention 0.00% (uptime < 10800)
threads-connected	OK	03-25-2009 11:29:54	0d 19h 51m 4s	1/5	OK - 2 client connection threads

- 更详细请参照官方文档:

<http://www.consol.de/opensource/nagios/check-mysql-health>

四、MySQL性能交互式监控工具

1、MySQL性能调优命令行工具介绍

* <http://www.day32.com/MySQL/tuning-primer.sh>

* <http://hackmysql.com/mysqlreport>

* <http://tools.assembla.com/svn/mysqltuner/mysqltuner.pl>

操作文章参考: <http://bbs.linuxtone.org/thread-988-1-1.html>

* innotop: <http://innodb.sourceforge.net>

2、慢查询日志分析

* 如何记录慢查询日志

```
# vi /etc/my.cnf
```

```
log-queries-not-using-indexes
```

```
long_query_time = 10
```

```
log-slow-queries = /data/mysql/data/slow.log
```

3、慢查询日志分析工具

[mysqldumpslow](#) , [mysqsla](#), [myprofi](#), [mysql-explain-slow-log](#), [mysqllogfilter](#)

四、相关参考资源

- 相关参考资源

1. [monitoring mysql slides en](#)

2. [MySQL Slow Tools](#)

3. [check mysql.c](#)

4. [MySQL Monitoring](#)

Thanks!



- **LinuxTone Web:**
--<http://www.linuxtone.org>
- **Email:**
- **--cnseek@gmail.com**