

MySQL DBA Tools and Tricks

ggurov@theladders.com

Me

- MySQL Production support DBA
- TheLadders.com – 100k+ Jobs for
- Previously: Hearst Digital, Vonage.
- On call 24/7 no matter what
- One of the first people to be called/woken up/tracked down.

Unreasonable demands

- Demand root
- Monitoring server/jumpbox in case of prod/dev separation to keep all the scripts on
- Maintain your own servers
- Trust few people
- The app always sucks
- Keep the same version on the servers and know the bugs
- The more things you automate, the more sleep you will get
- Simple tools help at 4am when something breaks, fixing it lets you get back to sleep quicker.

DB Server File Layout

- Everything in /dbdata[1-9] /dblogs and /prod
- /dbdataN and /dblogs are symlinks into real filesystem:

```
/dev/sdb1 4.00 131 3.70 2% /var/zeroin
/dev/mapper/DBSnap2-dbddata 1.8T 875G 947G 49% /dbdata_raw
[ggurov@db2 ~]$ ls -lad db*
lrwxrwxrwx 1 root root 19 May 3 2009 dbdata1 -> /dbdata_raw/dbdata1
drwxr-xr-x 6 root root 4096 Jun 22 2010 dbdata_raw
lrwxrwxrwx 1 root root 18 May 3 2009 dblogs -> /dbdata_raw/dblogs
[ggurov@db2 ~]$
```

DB Server File Layout Data

- /dbdata1/<instance name>/ houses the db files
- Predictable file paths, scriptable, etc
- Easily spin up a parallel instance/copy config files by running them through sed to replace instance name and server

```
dbdata1/  
dbdata1/db2  
dbdata1/db2/falcon_published  
dbdata1/db2/maria_log_control  
dbdata1/db2/maria_log.000000001  
dbdata1/db2/mysql  
dbdata1/db2/test  
dbdata1/db2/falcon  
dbdata1/db2/master.info  
dbdata1/db2/innodata1  
dbdata1/db2/relay-log.info  
dbdata1/db2/ladders
```

Master.info

- Replication information, log file, log position, user, password, port, timeout, etc. If you ever forget replication password, it is there. Can edit this file with the mysqld process shut down. Just know what you're doing. Log position here is


```
[root@db2 db2]# cat master.info
15
db1-bin.006576
103825018
10.0.1.76
sy_repl
Roach 
3306
60
0

0
[root@db2 db2]#
```

relay-log.info

- Exec position is in relay-log.info
- Flush tables with read lock keeps grabbing binary logs from master, can be misleading when trying to recover a DB from shambles.

```
[root@db2 db2]# cat relay-log.info
/dblogs/db2/relay/db2-relay.003218
2132584
db1-bin.006577
2132441
374
[root@db2 db2]#
```



DB Server Layout Logs

- Logs are split into 4 dirs: arch, bin, relay, tmp
- /dblogs/<instance name>/ is the base

```
total 1308072
drwx----- 6 mysql mysql      4096 Feb  9 11:27 .
drwx----- 3 mysql mysql      4096 Jul 19 2010 ..
drwx----- 2 mysql mysql      4096 Jun 21 2010 arch
drwx----- 2 mysql mysql    143360 May 25 09:15 bin
-rw-rw---- 1 mysql mysql     26688 May  4 16:22 mysqld.err
-rw-rw---- 1 mysql mysql     65932 Feb  2 16:12 mysqld.err-old
drwx----- 2 mysql mysql     36864 May 25 09:10 relay
-rw----- 1 mysql mysql 1337847874 May 19 23:18 slow.log
drwx----- 2 mysql mysql      4096 May 20 13:24 tmp
[root@db2 db2]#
```

DB Server Layout PID and socket

- /prod/<instance name>/mysqld.pid
- /prod/<instance name>/mysqld.sock
- Predictable

```
root@db2:/dblogs/db2
[root@db2 db2]# egrep -i "(pid|sock)" /etc/my.cnf | grep -v "spider" | uniq
pid-file           = /prod/db2/mysqld.pid
socket             = /prod/db2/mysqld.sock
[root@db2 db2]#
```

DB Server Layout Config

- Specify majority of the paths in the config file to know exactly where files will end up

```
root@db2:/dblogs/db2
[root@db2 db2]# egrep "(dbdata|dblogs|prod)" /etc/my.cnf | sort -k 3 | uniq
datadir                = /dbdata1/db2
innodb_data_file_path  = /dbdata1/db2/innodata1:128M:autoextend
innodb_log_group_home_dir = /dblogs/db2/arch
log_bin                = /dblogs/db2/bin/db2-bin
err-log                = /dblogs/db2/mysqld.err
log-error              = /dblogs/db2/mysqld.err
#log                  = /dblogs/db2/query.log
relay-log              = /dblogs/db2/relay/db2-relay
log-slow-queries       = /dblogs/db2/slow.log
slow_query_log_file    = /dblogs/db2/slow.log
tmpdir                 = /dblogs/db2/tmp
#update-log           = /dblogs/db2/update.log
pid-file               = /prod/db2/mysqld.pid
socket                 = /prod/db2/mysqld.sock
[root@db2 db2]#
```

MySQL Prompt with time

- Puts time into the prompt along with hostname to avoid performing operations on the wrong DB (at 3am).

```
[ggurov@db1 ~]$ grep -B 1 prompt /etc/my.cnf
[mysql]
prompt="db1 \\\R:\\\m:\\\s> "
[ggurov@db1 ~]$ mysql -u root

db1 21:48:30> █
```

Storage Tricks – LVM

- Quickly make a copy of a running database, no matter the size to copy to another host or simply bring up an extra instance
- Useful for DEV sanboxes, migration script testing

```
qadb8> flush tables;
Query OK, 0 rows affected (1.25 sec)

qadb8> flush tables with read lock;
Query OK, 0 rows affected (0.00 sec)
```

```
10:43 0:00 sshd: ggurov [priv]
[root@qadb8 ~]# lvcreate -s -n SNAP1 -L 50G /dev/QAUKDatabase/dbdata1
Logical volume "SNAP1" created
```

```
qadb8> unlock tables;
Query OK, 0 rows affected (0.00 sec)
```

Storage Tricks - LVM

- After snapshot, one can see how much snaps space is used up, and grow accordingly to extend life of the snapshot.

• The snapshot can be used to reync from

```
[root@qadb8 ~]# lvs
```

LV	VG	Attr	LSize	Origin	Snap%
SNAP1	QAUKDatabase	swi-ao	50.00G	dbdata1	0.00
dbdata1	QAUKDatabase	owi-ao	199.00G		

```
[root@qadb8 ~]# lvextend -L +25G /dev/QAUKDatabase/SNAP1
```

Extending logical volume SNAP1 to 75.00 GB
Logical volume SNAP1 successfully resized

```
[root@qadb8 ~]# lvs
```

LV	VG	Attr	LSize	Origin	Snap%
SNAP1	QAUKDatabase	swi-ao	75.00G	dbdata1	0.00
dbdata1	QAUKDatabase	owi-ao	199.00G		

Storage – LVM

- Combine multiple devices into one logical volume
- Easily add/remove disks if one is damaged or needs to be removed/restructured. Could expand the filesystem with `xfs_growfs` or `resize2fs`

```
[root@timemachinedb1 ~]# pvs
PV          VG          Fmt  Attr  PSize  PFree
/dev/sda2   VolGroup00  lvm2 a-    63.88G 12.94G
/dev/sdc1   VolGroup01  lvm2 a-    1.90T  96.99G
/dev/sdd1   VolGroup01  lvm2 a-    1.90T 660.94G
/dev/sde1   VolGroup01  lvm2 a-    1.90T  0
/dev/sdf1   DBSnap1     lvm2 a-    1.90T 106.79G

[root@timemachinedb1 ~]# vgs
VG          #PV #LV #SN Attr   VSize  VFree
DBSnap1     1  1  0 wz--n- 1.90T 106.79G
VolGroup00  1  8  0 wz--n- 63.88G 12.94G
VolGroup01  3 43 40 wz--n- 5.71T 757.93G

[root@timemachinedb1 ~]#
```

Storage tricks – SAN

- Slow, but provides useful facility to snapshot drives in the backend
- Lets create new slaves very quickly, there is not the initial copy time
- Saves lots of very expensive space in the backend when trying to scale reads with replicated slaves.
- Slow means more memory is required to offset the performance.
- Can be used for staging/dev environment that refreshes on a schedule.

Network

- Management on eth0
- Service on aliases (eth0:0 etc)
- Never give out management ip/name to the app
- Creates a flexible infrastructure – Can quickly reorganize the database split without reconfiguring the app, a lot less confusion

Big ON/OFF switch that DBA controls (assuming root)

```
[root@db2 db2]# /sbin/ifconfig | grep inet
    inet addr:10.0.1.77
    inet addr:10.0.1.177
    inet addr:127.0.0.1
[root@db2 db2]# host 10.0.1.77
77.1.0.10.in-addr.arpa domain name pointer db2-mgmt.backend.
[root@db2 db2]# host 10.0.1.177
177.1.0.10.in-addr.arpa domain name pointer db2.backend.
[root@db2 db2]#
```

Network

- One replica can serve many apps configured with different hostnames
- This server is a replicated slave that is also serving as a search and index instance. Should this machine become damaged, fall behind, service could be transferred to another slave without losing access (-mgmt is still up)
- Generally, The service ips do not come up on boot as one would want to figure out why the machine rebooted before putting it back into service.

```
[ggurov@dbslave5 ~]$ /sbin/ifconfig | egrep -i "(eth|inet)"
eth0      Link encap:Ethernet  HWaddr 00:50:56:91:39:2F
          inet addr:10.0.1.87  Bcast:10.0.1.255  Mask:255.255.254.0
eth0:0    Link encap:Ethernet  HWaddr 00:50:56:91:39:2F
          inet addr:10.0.1.190  Bcast:10.255.255.255  Mask:255.0.0.0
eth0:1    Link encap:Ethernet  HWaddr 00:50:56:91:39:2F
          inet addr:10.0.1.191  Bcast:10.255.255.255  Mask:255.0.0.0
          inet addr:127.0.0.1  Mask:255.0.0.0
[ggurov@dbslave5 ~]$ host 10.0.1.87
87.1.0.10.in-addr.arpa domain name pointer dbslave5-mgmt.backend.
[ggurov@dbslave5 ~]$ host 10.0.1.190
190.1.0.10.in-addr.arpa domain name pointer searchdb1.backend.idc.theladders.com.
[ggurov@dbslave5 ~]$ host 10.0.1.191
191.1.0.10.in-addr.arpa domain name pointer indexdb1.backend.idc.theladders.com.
[ggurov@dbslave5 ~]$
```

SSH

- Use keys, makes life a lot easier:

```
[ggurov@dbadmin1 ~]$ cat sshize
cat ~/.ssh/id_rsa.pub | ssh $* "mkdir .ssh; cat >> .ssh/authorized_keys;chmod -R
700 .ssh"
```

```
[ggurov@dbadmin1 ~]$ ./sshize vermini
mkdir: cannot create directory `'.ssh': File exists
```

```
[ggurov@dbadmin1 ~]$ ssh vermini
Last login: Tue May 24 10:56:56 2011 from dbadmin1.backend
[ggurov@vermini ~]$
```

SSH and Network

- SSH can be used to forward connections
- Might not be the fastest thing, but in pinch it could be used to restore service if there's no sysadmins around after a crash
- Can be used to perform maintenance on a slave with connections redirected to another slave

```
[ggurov@dbadmin1 ~]$ ssh -g -L 8888:localhost:3306 -N localhost  
█
```

```
[ggurov@dbadmin1 ~]$ telnet localhost 8888  
Trying 127.0.0.1...  
Connected to localhost.  
Escape character is '^]'.  
G  
5.0.40-enterprise-gpl-log_08eEfUU,yA, tIM'vB&4re#(^[[2~█
```

SSH and Network

- screen this to make it last a bit longer

```
[ggurov@dbadmin1 ~]$ while true; do ssh -g -L 8888:localhost:3306 -N localhost; sleep 1; done
```

- SSH can be used to temporarily create paths for replication, mysqldump, copying files, etc from either side

UCSPI – TCP

- Easily create network-bound services, interact with client via stdin/stdout

```
root@gurov-ssd:~/tcpserver# tcpserver 0 5559 ./runme.sh
^C
root@gurov-ssd:~/tcpserver# cat runme.sh
#!/bin/sh

echo "hi"
echo "current time is `date`"
root@gurov-ssd:~/tcpserver#
```

```
gurov@gurov-ssd:~$ telnet localhost 5559
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
hi
current time is Thu May 26 08:51:13 EDT 2011
Connection closed by foreign host.
gurov@gurov-ssd:~$ tcpcat 0 5559
hi
current time is Thu May 26 08:51:20 EDT 2011
```

UCSPI – TCP

- Binds stdin/stdout to a TCP socket
- <http://cr.yp.to/ucspi-tcp.html>

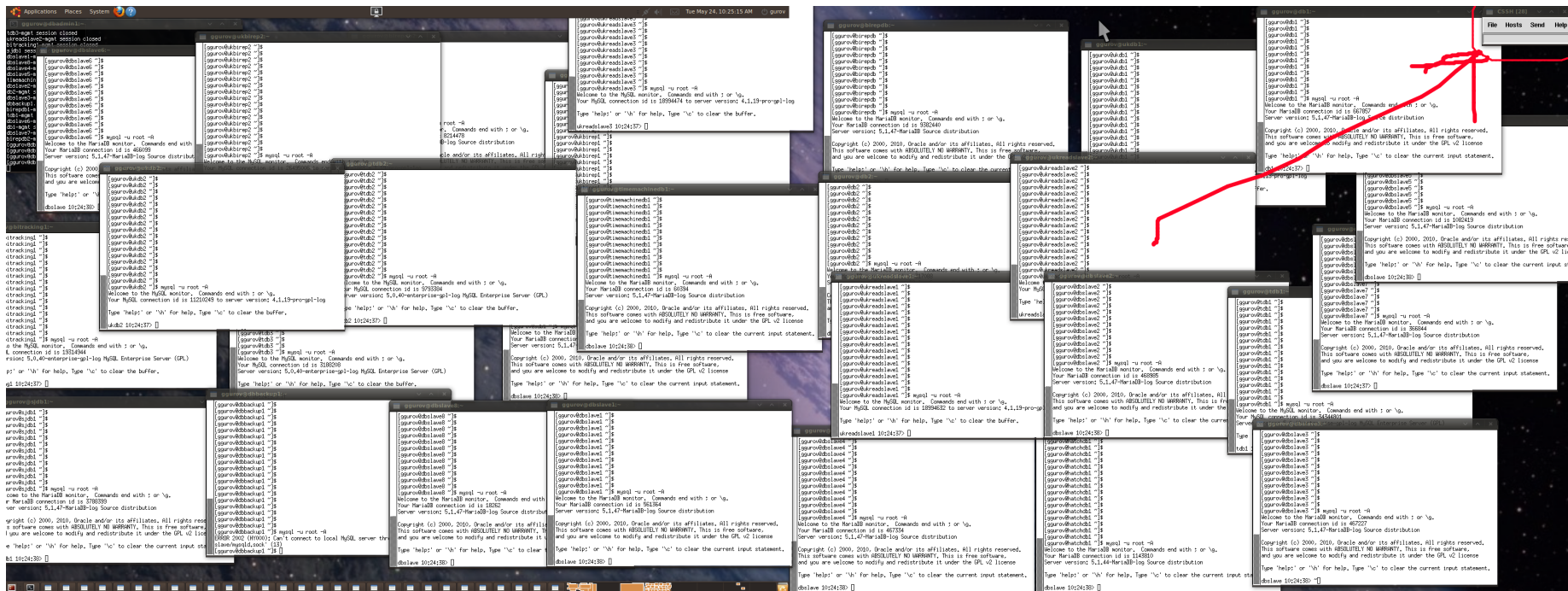
```
root@gurov-ssd:~/tcpserver# cat runme.sh
#!/bin/sh

mysql -N -u ap_dbtools -p`cat ~/.ap_dbtools_pass` -e "select publication_id, name
  from ladders.publication where premium = 1" ladders -h varick | cat
root@gurov-ssd:~/tcpserver# tcpserver 0 5559 ./runme.sh
```

```
gurov@gurov-ssd:~$ tcpcat 127.0.0.1 5559
2      SalesLadder Premium
5      MktgLadder Premium
6      FinanceLadder Premium
8      UpLadder Premium
12     OpsLadder Premium
14     TechnologyLadder Premium
16     LawLadder Premium
18     HRLadder Premium
gurov@gurov-ssd:~$ █
```

Cluster SSH

- Lots of similar machines ? No problem.
- Must have ssh X forwarding enabled (ssh -Y), have xauth installed, etc



ClusterSSH and use

- Example: check in every server with admin host via clusterssh

```
[root@tdb2 ~]# wget "http://dbadmin1/whoami/`hostname`/`whoami`"  
--10:28:28-- http://dbadmin1/whoami/tdb2.idc.theladders.com/root  
Resolving dbadmin1... 10.0.1.88  
Connecting to dbadmin1|10.0.1.88|:80... connected.  
HTTP request sent, awaiting response... 404 Not Found  
10:28:28 ERROR 404: Not Found.
```

```
[root@dbadmin1 httpd]# tail -2000 access_log | grep whoami  
10.0.1.104 - - [24/May/2011:10:28:28 -0400] "GET /whoami/ukreadslave1.idc.theladders.com/root HTTP/1.0" 200 100  
10.0.1.102 - - [24/May/2011:10:28:28 -0400] "GET /whoami/ukdb1.idc.theladders.com/root HTTP/1.0" 200 100  
10.0.1.105 - - [24/May/2011:10:28:28 -0400] "GET /whoami/ukreadslave2.idc.theladders.com/root HTTP/1.0" 200 100  
10.0.1.107 - - [24/May/2011:10:28:28 -0400] "GET /whoami/ukbirep1.idc.theladders.com/root HTTP/1.0" 200 100  
10.0.1.106 - - [24/May/2011:10:28:28 -0400] "GET /whoami/ukreadslave3.idc.theladders.com/root HTTP/1.0" 200 100  
10.0.1.103 - - [24/May/2011:10:28:28 -0400] "GET /whoami/ukdb2.idc.theladders.com/root HTTP/1.0" 200 100  
10.0.1.80 - - [24/May/2011:10:28:28 -0400] "GET /whoami/tdb1.idc.theladders.com/root HTTP/1.0" 404 0  
10.0.1.83 - - [24/May/2011:10:28:28 -0400] "GET /whoami/dbslave1.idc.theladders.com/root HTTP/1.0" 200 100  
10.0.1.85 - - [24/May/2011:10:28:28 -0400] "GET /whoami/dbslave3.idc.theladders.com/root HTTP/1.0" 200 100  
10.0.1.88 - - [24/May/2011:10:28:28 -0400] "GET /whoami/tdb2.idc.theladders.com/root HTTP/1.0" 404 0
```

Shell tricks

- Readily available on all machines
- Can make things a lot easier, less typing, etc.

```
[root@dbadmin1 httpd]# tail -2000 access_log | grep whoami | awk '{print $1" "$7}' | sed "s/\\/ /g" | awk '{print $1" "$3}' | sort
10.0.1.102 ukdb1.idc.theladders.com
10.0.1.103 ukdb2.idc.theladders.com
10.0.1.104 ukreadslave1.idc.theladders.com
10.0.1.105 ukreadslave2.idc.theladders.com
10.0.1.106 ukreadslave3.idc.theladders.com
10.0.1.107 ukbirep1.idc.theladders.com
10.0.1.108 ukbirep2.idc.theladders.com
10.0.1.109 tdb3.idc.theladders.com
10.0.1.129 timemachinedb1.idc.theladders.com
10.0.1.178 sdb1.idc.theladders.com
10.0.1.45 sjdb1.idc.theladders.com
10.0.1.76 db1.idc.theladders.com
```

```
[ggurov@db1 ~]$ mysql -u root -e "show grants for ap_lw_rw" -N | sed "s/;/;"
GRANT USAGE ON *.* TO 'ap_lw_rw'@'%' IDENTIFIED BY PASSWORD '*6E E663';
GRANT SELECT, INSERT, UPDATE, DELETE ON `jackrabbit`.* TO 'ap_lw_rw'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON `ladders`.* TO 'ap_lw_rw'@'%';
GRANT ALL PRIVILEGES ON `ladders`.`auto_complete_location_temp` TO 'ap_lw_rw'@'%';
GRANT ALL PRIVILEGES ON `ladders`.`auto_complete_company_name_temp` TO 'ap_lw_rw'@'%';
GRANT ALL PRIVILEGES ON `ladders`.`temp_suggested_jobs_old` TO 'ap_lw_rw'@'%';
GRANT ALL PRIVILEGES ON `ladders`.`temp_suggested_jobs_in_progress` TO 'ap_lw_rw'@'%';
GRANT ALL PRIVILEGES ON `ladders`.`auto_complete_study_field_temp` TO 'ap_lw_rw'@'%';
GRANT ALL PRIVILEGES ON `ladders`.`auto_complete_title` TO 'ap_lw_rw'@'%';
```

Shell tricks

- Count rows in tables – can be used find differences in tables, do on both hosts, diff the output.

```
[ggurov@sdb1 ~]$ for i in `mysql -u root -e "show tables like 'jobseeker_profile_history\_%'" ladders -N` ; do echo "select '$i', count(1) from $i;" | mysql -u root ladders -N ;done
jobseeker_profile_history_20051217      7323920
jobseeker_profile_history_20070720      6186458
jobseeker_profile_history_20080219      6453922
jobseeker_profile_history_20080612      3077408
jobseeker_profile_history_20080810      6434360
```

- Kill a bunch of connections in a pinch (barrage, massive cache expiration)

```
[ggurov@sdb1 ~]$ for i in `mysql -N -u root -e "show processlist" | grep -v "Sleep" | grep -v "sys_r
ep1" | awk '{print $1}'` ; do echo "kill $i;";done
kill 4472163:
kill 4730850:
kill 5633397:
```

```
[ggurov@sdb1 ~]$ for i in `mysql -u root -e "select user from user where host = '%'" -N mysql` ; do mysql -N -
u root -e "show grants for $i" | sed "s/;/;/";done | sed "s/IDENTIFIED BY,./;/";
GRANT USAGE ON *.* TO 'ap_bi_etl_ro'@'%';
GRANT SELECT ON `ladders`.* TO 'ap_bi_etl_ro'@'%';
GRANT SELECT, PROCESS, SUPER, LOCK TABLES, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'ap_dbtools'@'%';
GRANT INSERT ON `ladders`.`sys_read_ratio` TO 'ap_dbtools'@'%';
GRANT UPDATE ON `ladders`.`replication_timing_test` TO 'ap_dbtools'@'%';
GRANT DELETE ON `ladders`.`visitor_verification_code` TO 'ap_dbtools'@'%';
GRANT USAGE ON *.* TO 'ap_fr2_ro'@'%';
GRANT SELECT ON `ladders`.* TO 'ap_fr2_ro'@'%';
GRANT SELECT ON `ladders_franklin`.* TO 'ap_fr2_ro'@'%';
GRANT USAGE ON *.* TO 'ap_fr2_rw'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON `ladders`.* TO 'ap_fr2_rw'@'%';
```

Shell tricks

- Change every table to innodb (rebuilding innodb tables in the process - create an LVM snapshot in DEV first, do this, see if anything breaks after the app starts up.

```
[ggurov@sdb1 ~]$ for i in `mysql -u root -e "show tables" -N falcon`; do echo "alter table $i engine=innodb;"; done
alter table acquisition_type engine=innodb;
alter table audit_trail engine=innodb;
alter table cell engine=innodb;
alter table component engine=innodb;
alter table component_property engine=innodb;
alter table component_property_map engine=innodb;
alter table component_section_map engine=innodb;
alter table component_type engine=innodb;
```

- Get count of engines used. Information_schema can also be used, but 4.1 doesn't have it.

```
[ggurov@sdb1 ~]$ mysql -N -u root -e "show table status" ladders | awk '{print $2}' | sort | uniq -c | sort -rn
586 InnoDB
423 MyISAM
9 MRG_MYISAM
3 MEMORY
1 MARIA
```

Admin file database

- Nothing to go down, break, just files. Administrative tasks less likely to be affected.

```
[ggurov@dbadmin1 ~]$ ls hosts
birepdb1      dbslave2     dbslave8     tdb3         ukreadslave1
birepdb2      dbslave3     hatchdb1     timemachinedb1 ukreadslave2
bitracking1   dbslave4     sdb1         ukbirep1     ukreadslave3
db1           dbslave5     sjdb1        ukbirep2
db2           dbslave6     tdb1         ukdb1
dbslave1     dbslave7     tdb2         ukdb2
[ggurov@dbadmin1 ~]$
```

- Foreach the hosts:

```
[ggurov@dbadmin1 ~]$ for i in hosts/*; do mysql -h `basename $i` -mgmt -uap_dbtools -p`cat ~/.ap_dbtools_pass` -N -e "
show global variables like 'hostname'" | cat; done
hostname      birepdb.idc.theladders.com
hostname      birepdb2.idc.theladders.com
hostname      bitracking1.idc.theladders.com
hostname      db1.idc.theladders.com
hostname      db2.idc.theladders.com
```

Admin file database

- check_running, single command returns number of Threads_running across all hosts:

```
#!/bin/sh

app_path=`dirname $0`
cd $app_path

DBPASS=`cat ~/.ap_dbtools_pass`

for i in ~/hosts/*;
do
    HOST=`basename $i`
    MGMT=`basename $i`-mgmt

    echo -n "$HOST `cat $i` : ";
    echo `mysql -f -u ap_dbtools -h $MGMT -p$DBPASS -e "show status like 'Threads
_running'\G" 2> /dev/null | grep "Value:" | awk -F ':' '{print $2}'`
done
```

- Quick “health check” when there is panic, quickly identify server with backed up queries.

```
[ggurov@dbadmin1 scripts]$ ./check_running
birepdb1          : 1
birepdb2          : 1
bitracking1       : 1
db1               : 16
db2               : 5
```

Monitoring

- Always watch, basic stats. Replication status, number of connections, threads_running, qcache state, queries per second, out of pattern things demand

host	LAG	CONN	RUN	QCACHE	QPS
birepdb1		16		624	
birepdb2		3		30	
bitracking1		5		33	6
clx					
db1		640	16	21 371	593
db2		7	5	3 640	
dbbackup1		3			
dbbackup1:3307	22 799	3			
dbbackup1:8306		3			
dbslave1		108	2	24 993	81

- Get this emailed every hour, and look at it. You will start to notice pattern, out of pattern means something broke or changed

From: ggurov@dbadmin1.idc.theladders.com
To: Gennady Gurov
Cc:
Subject: cluster state

host	LAG	CONN	RUN	QCACHE	QPS
birepdb1		16		633	
birepdb2		7		39	
bitracking1		5		33	
clx					
db1		643	15	20 947	
db2		7	5	3 640	

Monitoring

- Watch disk usage, make the servers complain repeatedly until fixed

```
TMP="`mktemp`"
THRESHHOLD=75

for i in `grep -v 'Mounted on' | grep -v '/boot' | sed "s/%%/g" | awk '{print $6"|"$5"|"$4"|"$2}'`; do
    FS=`echo $i | awk -F'|' '{print $1}'`
    SZ=`echo $i | awk -F'|' '{print $2}'`
    FREE=`echo $i | awk -F'|' '{print $3}'`
    TOTAL=`echo $i | awk -F'|' '{print $4}'`

    CALC=`echo "$SZ > $THRESHHOLD" | bc`

    if [ $CALC == "1" ]; then
        echo "$FS is over $THRESHHOLD% (free $FREE/$TOTAL = $SZ%)";
    fi;
done > "$TMP"

MSG="`cat $TMP`"

if [ ! -z "$MSG" ]; then
    echo "$MSG" | mail ggurov@theladders.com -s "FIXME - $1 filesystem alert - FIXME";
fi;

rm -f "$TMP"
```

```
From: ggurov@dbadmin1.idc.theladders.com
To: Gennady Gurov
Cc:
Subject: FIXME - sjdb1 filesystem alert - FIXME
```

```
/dbdata is over 75% (free 30G/192G = 85%)
```

- Make it obvious from subject line that something could be broken.

Monitoring

- Check filesystem every hour from cron. Make sure ssh keys are set

```
[ggurov@dbadmin1 scripts]$ crontab -l | grep check_fs
0 * * * * /home/ggurov/scripts/check_fs_cmd
[ggurov@dbadmin1 scripts]$ cat /home/ggurov/scripts/check_fs_cmd
app_path=`dirname $0`
cd $app_path

for i in ~/hosts/*;
do
    HOST=`basename $i`
    MGMT=`basename $i`-mgmt
    ssh $MGMT " df -Ph " | ./check_filesystem $HOST
done
[ggurov@dbadmin1 scripts]$
```

Error log reporting

Subject: *ERROR LOG* timemachinedb1.idc.theladders.com

```
110524 12:00:04 [Note] Slave I/O thread killed while reading event
110524 12:00:04 [Note] Slave I/O thread exiting, read up to log 'db1-bin.006559', position 13423624
110524 12:00:04 [Note] Slave SQL thread exiting, replication stopped in log 'db1-bin.006559' at position 13422166
110524 12:00:12 [Note] Slave SQL thread initialized, starting replication in log 'db1-bin.006559' at position 13422166, relay log
'/dblogs/dbslave/relay/dbslave-relay.001402' position: 13422309
110524 12:00:12 [Note] Slave I/O thread: connected to master 'sy_repl@10.0.1.76:3306', replication started in log 'db1-bin.006559' at position 13423624
110524 12:00:04 [ERROR] Slave I/O thread killed while reading event
110524 12:00:04 [ERROR] Slave I/O thread exiting, read up to log 'ukdb1-bin.001604', position 69237334
110524 12:00:04 [ERROR] Error reading relay log event: slave SQL thread was killed
```

```
[ggurov@timemachinedb1 ~]$ cat /usr/local/bin/rtail.sh
#!/bin/sh

FILE=$1
STATEDIR=~/.rtail_state
STATEFILE_FRAG=`echo $FILE | sed "s/\/\//-/g"`
STATEFILE=$STATEDIR/$STATEFILE_FRAG

if [ ! -f "$FILE" ]; then
    echo "file not found";
    exit;
fi;

if [ ! -d "$STATEDIR" ]; then
    mkdir "$STATEDIR";
fi;

CURSIZE=`ls -la "$FILE" | awk '{print $5}'`

if [ ! -f "$STATEFILE" ]; then
    LASTSIZE=0;
else
    LASTSIZE=`cat "$STATEFILE"`;
fi;

if [ $LASTSIZE -gt $CURSIZE ]; then
    LASTSIZE=0
fi;

echo $CURSIZE > "$STATEFILE"

TOGET=$(( $CURSIZE - $LASTSIZE ))

tail -c $TOGET "$FILE"
```

- Rtail.sh

Error log reporting

- Schedule to run on every machine every minute. Anything goes into the error log, you will know about it before rest of the monitoring kicks in or the application starts breaking

```
[root@timemachinedb1 ~]# crontab -l | grep err
* * * * * /usr/local/dbscripts/err_report 2> /dev/null
[root@timemachinedb1 ~]# cat /usr/local/dbscripts/err_report
#!/bin/sh
TFILE=`mktemp`

for i in /dblogs/*/mysqld.err; do
    /usr/local/bin/rtail.sh $i >> $TFILE
done

if [ -s "$TFILE" ]; then
    cat $TFILE | /bin/mail -s "*ERROR LOG* `hostname`" database.team
fi;

rm -f $TFILE
```

Moderately-sized data

- Merge tables

```
tdb2 18:00:45> show tables like 'user_tracking%';
+-----+
| Tables_in_ladders_tracking (user_tracking%) |
+-----+
| user_tracking                               |
| user_tracking_20100504                     |
| user_tracking_20100509                     |
| user_tracking_20100512                     |
| user_tracking_20100517                     |
| user_tracking_20100520                     |
| user_tracking_20100524                     |
| user_tracking_20100528                     |
| user_tracking_20100602                     |
```

```
[root@tdb2 ladders_tracking]# du -ch user_tracking* | grep total
306G    total
```

Merge tables

- Concatenates a bunch of MyISAM tables together under a common name
- Auto rotates every few days when it gets to 4-ish gigs (script does this)
- Makes alter_table possible one segment at a time. Can be done “online” 4 gig segment at a time, possibly without replication delay. Segments are disconnected, altered and recombined with new schema in the end.
- Not the fastest thing out there, but already included in mysql and easy to swap segments in and out

```
mysql> show create table user_tracking \G
***** 1. row *****
      Table: user_tracking
Create Table: CREATE TABLE `user_tracking` (
  `user_tracking_id` bigint(20) NOT NULL auto_increment,
  `visitor_tracking_id` bigint(20) unsigned NOT NULL default '0',
  `session_id` varchar(32) NOT NULL default '',
  `visitor_id` varchar(48) NOT NULL default '',
  `ts` datetime NOT NULL default '0000-00-00 00:00:00',
  `subscription_id` int(11) NOT NULL default '0',
  `ip_address` varchar(15) NOT NULL default '',
  `referer` text,
  `url` text,
  `basket` varchar(4) NOT NULL default '',
  `egg` varchar(16) NOT NULL default '',
  `status` char(1) NOT NULL default '',
  `carton` varchar(16) NOT NULL default '',
  `parameter` text,
  `web_node` char(3) NOT NULL default '',
  `response_time` int(7) NOT NULL default '0',
  `web_page_id` int(11) NOT NULL default '0',
  `dyn_cell_id` int(11) NOT NULL default '0',
  `response_code` int(3) NOT NULL default '0',
  PRIMARY KEY (`user_tracking_id`),
  KEY `user_tracking_subscription_id_i` (`subscription_id`)
) ENGINE=MRG_MyISAM DEFAULT CHARSET=latin1 INSERT_METHOD=LAST UNION=(`user_tracking_20100524`,`user_tracking_20100526`,`user_tracking_20100602`,`user_tracking_20100628`,`user_tracking_20100702`,`user_tracking_20100707`,`user_tracking_20100804`,`user_tracking_20100810`,`user_tracking_20100815`,`user_tracking_20100916`,`user_tracking_20100922`,`user_tracking_20100927`,`user_tracking_20101031`,`user_tracking_20101105`,`user_tracking_20101112`,`user_tracking_20101220`,`user_tracking_20101229`,`user_tracking_20110110`,`user_tracking_20110202`,`user_tracking_20110207`,`user_tracking_20110307`,`user_tracking_20110310`,`user_tracking_20110315`,`user_tracking_20110407`,`user_tracking_20110411`,`user_tracking_20110414`,`user_tracking_20110505`,`user_tracking_20110510`,`user_tracking_20110515`,`user_tracking_20110518`)
1 row in set (0.00 sec)
```

timemachine

- **LVM-based**
- **Not a backup replacement, but quite useful, mine snapshots every 12 hours, keeps 40 copies of 1tb data in 3-4 TB**

```
[root@timemachinedb1 ~]# /usr/local/timemachine/time_controller/time_stop.php
```

```
=====
```

```
Tue May 24 11:49:39 EDT 2011
```

```
time=1306252179
```

```
-----
```

```
Total snapshots: 40.
```

```
Newest snapshot: 11 hours 49 minutes 28 seconds.
```

```
Oldest snapshot: 2 weeks 5 days 23 hours 49 minutes 33 seconds.
```

```
-----
```

```
Current states
```

TIME_10	125.00G	54.54	1304524806	Wed, 04 May 2011 12:00:06	-0400
TIME_11	95.00G	70.58	1304568004	Thu, 05 May 2011 00:00:04	-0400
TIME_12	95.00G	69.78	1304611206	Thu, 05 May 2011 12:00:06	-0400
TIME_13	95.00G	68.86	1304654405	Fri, 06 May 2011 00:00:05	-0400
TIME_14	95.00G	68.08	1304697605	Fri, 06 May 2011 12:00:05	-0400
TIME_15	95.00G	67.23	1304740806	Sat, 07 May 2011 00:00:06	-0400

TIME_11	VolGroup01	swi-a-	95.00G	lv.dbdata	70.58
TIME_12	VolGroup01	swi-a-	95.00G	lv.dbdata	69.78
TIME_13	VolGroup01	swi-a-	95.00G	lv.dbdata	68.85
TIME_14	VolGroup01	swi-a-	95.00G	lv.dbdata	68.08
TIME_15	VolGroup01	swi-a-	95.00G	lv.dbdata	67.23
TIME_16	VolGroup01	swi-a-	95.00G	lv.dbdata	65.80
TIME_17	VolGroup01	swi-a-	95.00G	lv.dbdata	65.41
TIME_18	VolGroup01	swi-a-	95.00G	lv.dbdata	65.08

timemachine

- Each instance can be brought up fairly quickly by mounting and starting `mysqld_safe --defaults-file=/etc/my-BK.cnf`
- Snapshot creation stops slave on two instances, waits for a bit, creates lvm snapshot
- Relay logs live on separate partition to avoid blowing through snap space too quickly
- Straightforward to implement but very useful in case someone runs something and it's not noticed for some time
- Not a replacement for proper backups but can come close to it for short term recovery.

Row level binary log parsing

- Can see rows/second to compare previous days with current. These are raw rows, not queries, but those can also be retrieved to see if any query should be batched to eliminate queries.

```
Subject:      db1 insert traffic US
-----
time: (Tue, 24 May 2011 08:19:41 -0400 - Tue, 24 May 2011 10:37:01 -0400) elapsed: 8241 seconds

125309 15.21/sec ladders.jp_search_industry
57000  6.92/sec ladders.jobclick
54000  6.55/sec ladders.beacon
16568  2.01/sec ladders.jp_search_job_function
13378  1.62/sec ladders.job_tracking
```

Row level binary log parsing

- Insert can be substituted for update or delete to see how many rows were affected in that log file. Logfile also contains markers for start and end time.

```
[ggurov@dbadmin1 ~]# /usr/local/bin/mysqlbinlog --verbose -R -u ap_dbtools -h db1 -p`cat ~/.ap_dbtools_pass` db1-bin.006575 | g
rep "### INSERT INTO ladders." | sed "s/### INSERT INTO//" | sort | uniq -c | sort -rn
169784 ladders.jp_search_industry
47000 ladders.beacon
32000 ladders.jobclick
21531 ladders.profile_hit_count
20834 ladders.email_parameter
20021 ladders.jp_search_job_function
16557 ladders.recruiter_broadcast
```

Binary log following

- Can be used to implement 0-cost custom triggers that connects as a mysql slave and follows the binary log stream.

```
$cmd = "/usr/local/src/labs/mysql-5.6.2-labs-innodb-memcache  
d-linux2.6-x86_64/bin/mysqlbinlog --verbose --stop-never --s  
top-never-slave-server-id=8344 -R -h {db_host} -u {db_user  
} -p{db_pass} --start-position=$last_pos $binlog_file";  
$r = popen($cmd, 'r');
```

Table growth

- Monitor table growth to ensure nothing runs away. Shows current sizes of tables that changed in size. This is raw size in bytes.

```
20,717,764,608 jobclick InnoDB
13,748,928,512 job_feed_job_history InnoDB
13,571,719,168 jobseeker_experience InnoDB
9,290,383,360 jp_search_industry InnoDB
7,966,031,872 job InnoDB
7,013,924,864 subscriber_parameter InnoDB
3,931,062,272 jobseeker_profile InnoDB
3,672,113,152 job_tracking InnoDB
3,468,740,268 job_tracking_history MyISAM
3,122,407,824 job_history_20110105 MyISAM
```

Questions/Contact

- email: ggurov@gmail.com
- Aim: ggurov