

Capacity Planning With MySQL

Monday, August 14th, 2007
Boston MySQL User Group

Sheeri Kritzer
awfief@gmail.com
<http://www.sheeri.com>

What is Capacity Planning?

- Business logic metrics
- Everyone wants to do it
- Nobody seems to be doing it

You Can Start Now!

- Do it step-by-step
- Start gathering metrics now
- Figure out what to do with them later

What Is a Metric?

- “standard of measurement”
- What are your company's standards?
- How will you get that information?

Commands That Output Metrics

- OS:

vmstat
netstat

df
ps
cat /proc

iostat
uptime

- MySQL:

SHOW STATUS

SHOW TABLE STATUS

\s

- Other:

wget

Commands vs. /proc

- Load:

```
/usr/bin/uptime | /bin/awk '{print $10}' |  
  /bin/cut -f1 -d,
```

or

```
cat /proc/loadavg | /bin/awk '{print $1}'
```

especially since you cannot guarantee how many commas there will be!

Commands that Output Metrics (con't)

- Scripts you write
- Try to measure user experience
- Do not forget: server name, current time

Gathering Metrics Affects the System

- Metrics record all effects anyway
- How often will you run them?
- Be smart with your scripts!

Skew the Data as Little as Possible

```
/usr/bin/mysql -u $USER -p$PASS $DB -e  
"SHOW STATUS LIKE Questions"
```

```
/usr/bin/mysql -u $USER -p$PASS $DB -e  
"SHOW STATUS LIKE Connections"
```

```
/usr/bin/mysql -u $USER -p$PASS $DB -e  
"SHOW STATUS LIKE Slow_queries"
```

versus:

```
/usr/bin/mysql -u $USER -p$PASS $DB -e  
"show status" | /bin/awk '/Connections\W|  
Questions\W|Slow_queries\W/ {printf $2 ", "}'
```

What Are You Waiting For?

- Write to a file
- Formatting
- Database storage

Permissions

- Most systems allow any user to get info with no special permissions
- MySQL allows any user to SHOW STATUS
- Grant permissions as needed

Import Data

- How to import without losing data?
- OK to have a manual process

Schema

- 1 table with all fields
- Joining table for easy retrieval
- Not the only way!

Update 'weeks' Table

```
REPLACE INTO weeks (serverName,statsTime)  
SELECT serverName,MAX(statsTime) FROM  
dbstats GROUP BY serverName;
```

```
UPDATE weeks INNER JOIN dbstats USING  
(serverName,statsTime) SET  
weeks.id=dbstats.id WHERE weeks.id IS  
NULL;
```

Now What?

- Not all metrics created equal
- Some can be compared simply: free space
- MySQL status resets on db restart

Stored Procedures

- `diffStats(date,serverName,field)`
serverName can be partial or ''
- `getAllStats(serverName,field)`
runs diffStats for all dates in 'weeks'

Stored Procedures (continued)

- `maxDiff(date,serverName,field)`
more complex subtraction
- `getAllDiff(serverName,field)`
- Still need to compare point-in-time data
processes, load, free & cache memory,
minModifiedTime, qps, tps, "Sessions"

Overall Report

- Not wonderful right now
- Better than nothing!
- Shows how much is happening

For the Future

- maxDiff should find all local maxima for calculations
- Flexible dates
- Prettier report

One Last Note

- This is not a monitoring system
- Monitoring is in real-time
- You need both!

Questions, Comments, Suggestions?

Thanks!