5 Proactive Measures to Minimize MySQL Downtime

Fernando Ipar

MySQL NoSQL Cloud 2013

Buenos Aires, Argentina

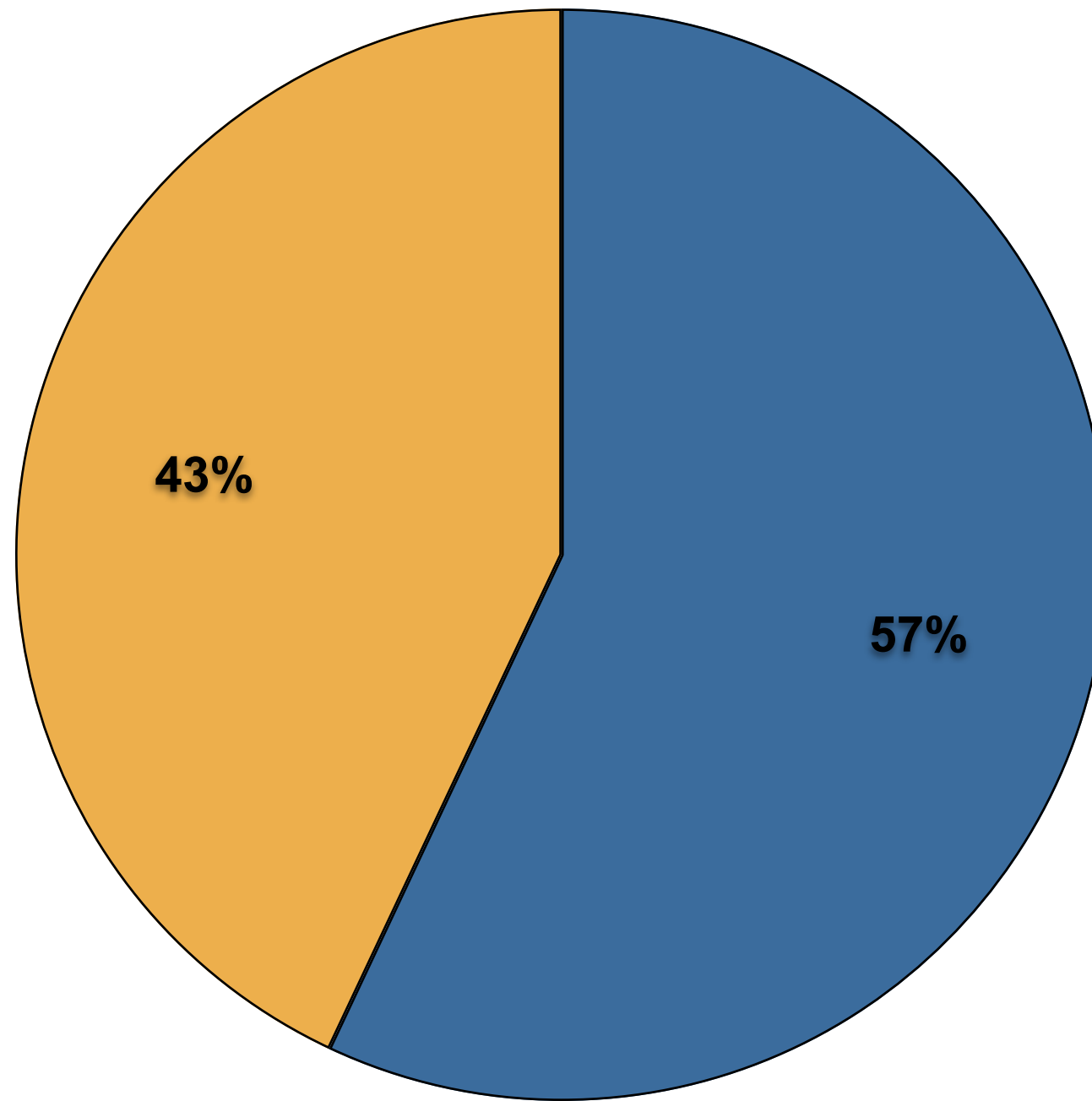# About me...

About this talk...

- 100% based on real world experience
- Involves failures, problems, <u>disaster</u>.
- How to avoid them or minimize their impact.

What about disaster?

It's bound to happen!

(total of 556 analyzed cases)

Avoidable or Mitigated   Unavoidable

43%

57%

The measures

1. Keep tested backups

Things that are not a backup:
- Replica

- RAID

- DRBD/SAN

- any faithful and live copy of your
  data.

Things that are a backup:
- mysqldump
- xtrabackup
- FS snapshot
- any faithful and consistent copy of your data at one point in time.

Things to consider
- Production impact
- Purpose
- Data set size
- Backup/Restore time

Testing backups

- If it's not tested, it's not a backup

- Restore to dev/stage/etc.

- Restore to a replica, then pt-table-checksum

- Big Data considerations

2. Build redundant components

Things that are not a backup but that provide you redundancy:

- Replica

- RAID

- DRBD/SAN

- any faithful and live copy of your data.

If possible, support a degraded mode.
- fallback to static content
- disable writes
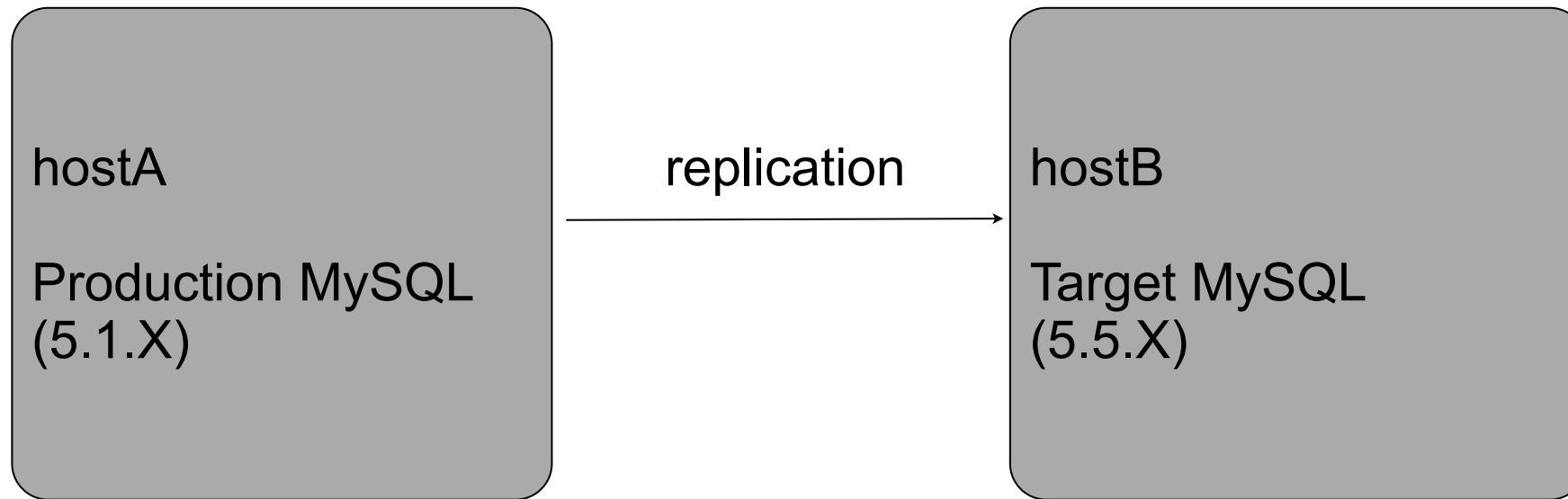- modularity and configurability are important for this

3. Test changes

- Test upgrades
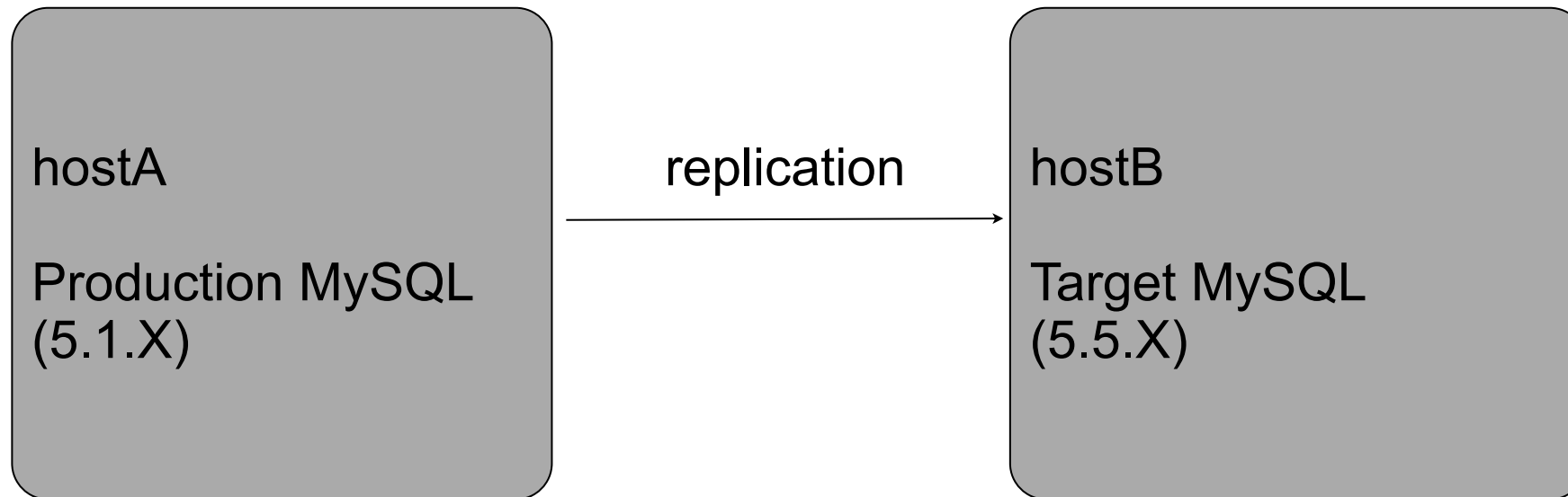   - pt-upgrade / pt-table-checksum

Testing upgrades

hostA

Production MySQL
(5.1.X)

Testing upgrades

hostA

Production MySQL
(5.1.X)

replication →

hostB

Target MySQL
(5.5.X)

Testing upgrades


```
hostA                          replication          hostB

Production MySQL          ───────────────▶     Target MySQL
(5.1.X)                                         (5.5.X)
```

pt-table-checksum --replicate percona.checksums h=hostA

pt-upgrade queries.txt h=hostA h=hostB
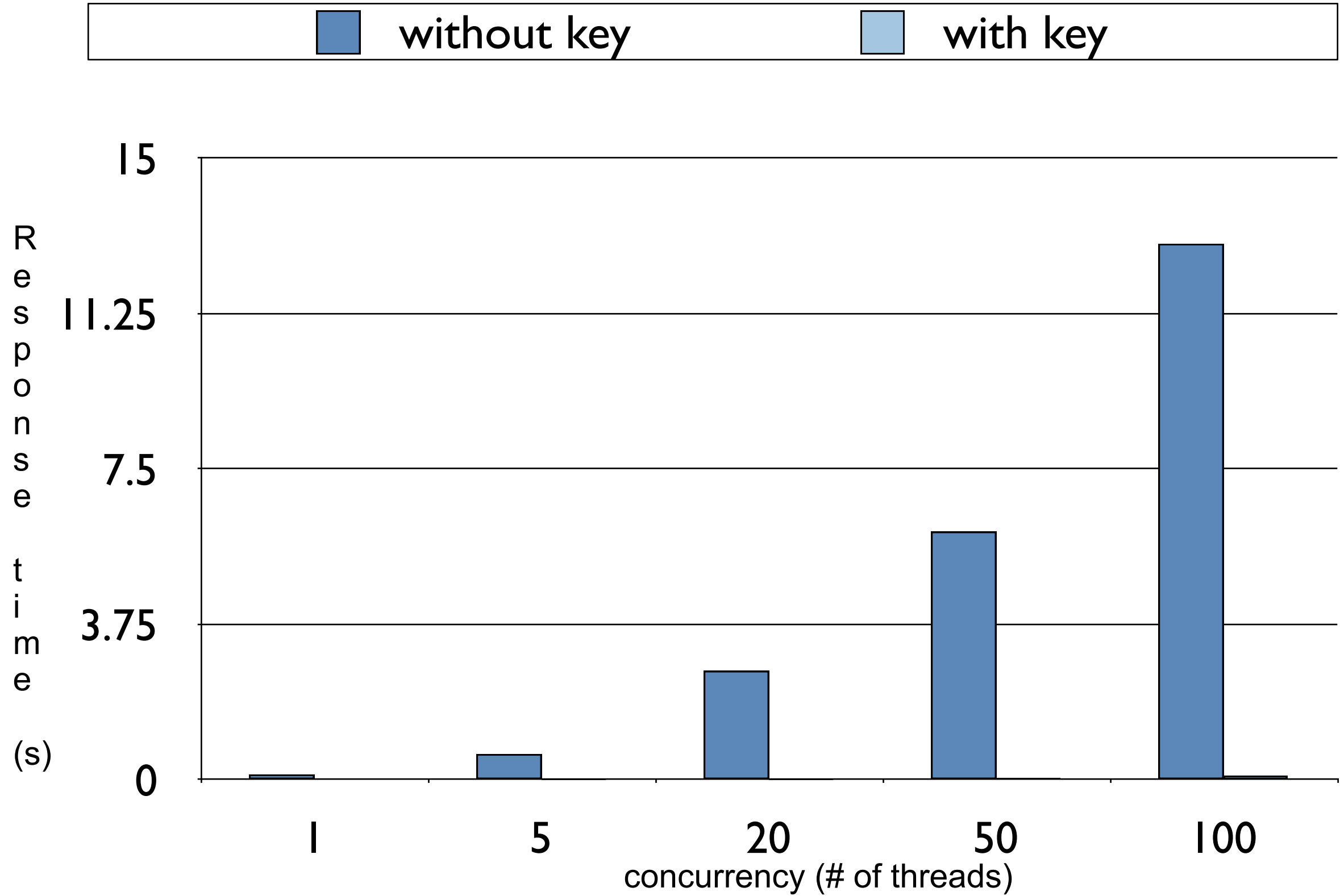
- Test upgrades
  - pt-upgrade / pt-table-checksum
- Test app changes
  - Jmeter, sysbench, mysqlslap, etc.
  - Test realistic workloads and data set sizes.

Testing app changes

```sql
CREATE TABLE `bad_scaling_test` (
  `id` int NOT NULL AUTO_INCREMENT,
  `payload` text,
  `rank` int unsigned DEFAULT '0',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB
```

```sql
select * from test.bad_scaling_test order by rank desc limit 10
```

Testing app changes

Legend: without key (dark blue), with key (light blue)

Response time (s) vs concurrency (# of threads)

Y-axis: 0, 3.75, 7.5, 11.25, 15

X-axis: 1, 5, 20, 50, 100

http://bugs.mysql.com/bug.php?id=61410

Bug #61410	an unsigned column can't subtract a bigger value than before	
Submitted:	5 Jun 2011 6:47	
Reporter:	甲冑 未名	
Status:	Not a Bug	
Category:	Server: DML	
Version:	5.5.11	

- Test upgrades
  - pt-upgrade / pt-table-checksum
- Test app changes
  - Jmeter, sysbench, mysqlslap, etc.
  - Test realistic workloads and data set sizes.
- Have a rollback plan
- Link queries to code paths

SELECT ... /* /app/cart/get */

4. Practice safe replication

- Prefer RBR when possible
  - Make sure you have PK/UK

- Prefer RBR when possible
  - Make sure you have PK/UK
- Monitor lag (pt-heartbeat)

- Prefer RBR when possible
  - Make sure you have PK/UK
- Monitor lag (pt-heartbeat)
- Avoid huge trx on master

- Prefer RBR when possible
  - Make sure you have PK/UK
- Monitor lag (pt-heartbeat)
- Avoid huge trx on master
- Set passive nodes @@read_only
- Reserve SUPER for admin accounts

- Prefer RBR when possible
  - Make sure you have PK/UK
- Monitor lag (pt-heartbeat)
- Avoid huge trx on master
- Set passive nodes @@read_only
- Reserve SUPER for admin accounts
- Periodic pt-table-checksum

5. Monitor, Trend

- http://www.percona.com/software/percona-monitoring-plugins
  - Nagios
  - Cacti
- Zabbix, Munin, Zenoss, etc.

- Avoids or minimizes impact of:
    - running out of disk space
    - degraded HW
    - reaching max_connections
    - workload changes
    - etc.
- Provides 'Normal' baseline when troubleshooting

The take home message

fernando.ipar@percona.com