



What causes downtime in MySQL, and how can you prevent it?

Espen Braekken
Percona Live London 2011

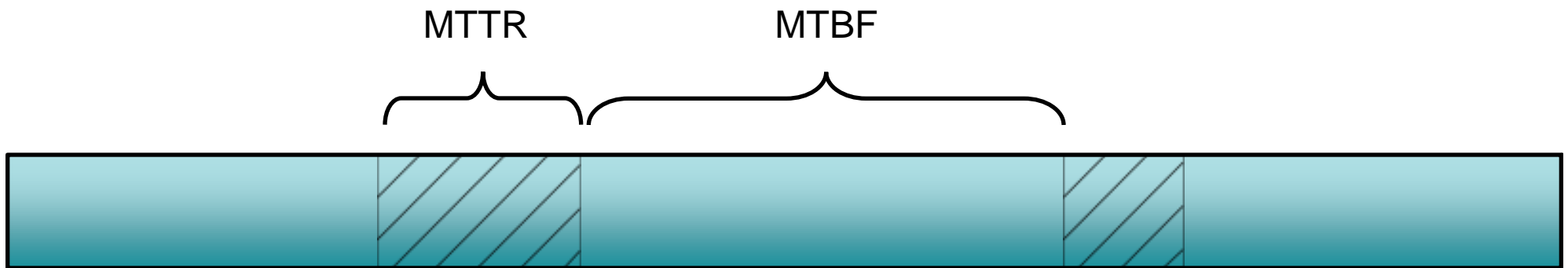
Agenda

- What is High Availability?
- What Causes Downtime in MySQL?
- How to Prevent Downtime
- Resources

Part I: High Availability

High Availability

- Absence of Downtime
- MTBF
- MTTR



Reducing MTTR

- Find out quickly (monitoring & alerting)
- Recover quickly (redundancy & failover)



Many people focus on technology; limited, reactive

Increasing MTBF

- Understand failures (research, post-mortems)
- Work to prevent or reduce failures



Boring, hard to justify—but proactive!

Goals of this Presentation

- Why does downtime happen?
 - Prerequisite to preventing it
- Which failures are most common?
 - Understand and prioritize risks
- What could have prevented the incidents?
 - Which preventions are effective?

Proactive

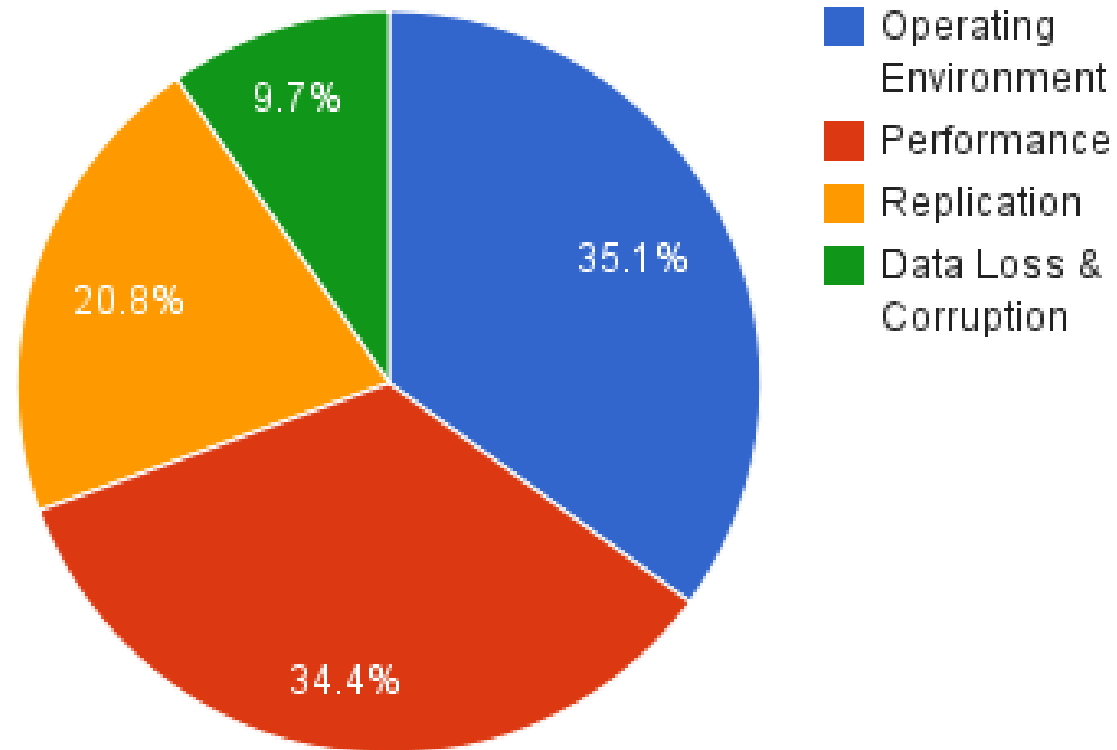
- “-adjective. Serving to prepare for, intervene in, or control an expected occurrence or situation, especially a negative or difficult one; anticipatory: proactive measures against crime.”
— dictionary.com

Part II: Understanding Downtime Incidents

Research Background

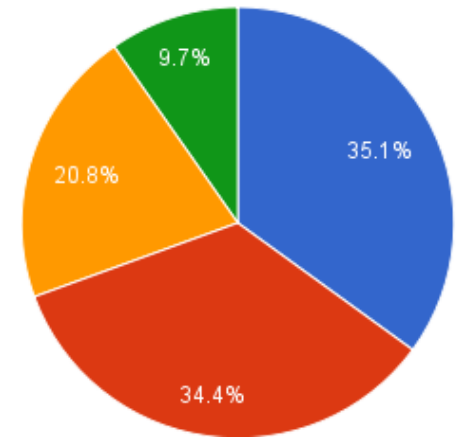
- Our credentials
 - We provide emergency services for MySQL users
- Source dataset
 - About 200 emergency issues; 154 selected
- Identify and categorize
 - Location, causes, preventions of failure
 - Rank these three by frequency

Where Incidents Occur



Top Ten Incident Types

“Cause”	Category	Count	Percent
SQL	Performance	20	12.9%
Data difference	Replication	14	9.1%
DROP TABLE	Data loss/corruption	9	5.8%
Disk full	Operating environment	9	5.8%
Network	Operating environment	9	5.8%
Operating system	Operating environment	8	5.2%
Schema/indexing	Performance	8	5.2%
InnoDB	Performance	8	5.2%
Configuration	Performance	7	4.5%
Configuration	Replication	6	3.9%



For much more detail, see the Resources at the end of this slide deck.

Root Cause Analysis

- Incidents have causes, but not “root” causes
- There is always a chain of failures
- A single intervention is a prevention

#1 Cause of Downtime

- Lack of Change Control
 - Often upgrade-related, but not always
 - Configuration changes
 - Schema/query changes; deployments
- Upgrades
 - Careless Upgrades
 - Query behavior changes, plan changes, bugs
 - Failure to upgrade
 - Bugs, bugs, bugs

Part III: Prevention and Proactivity

Proactivity

- Proactivity requires routine activity
- It's important to document
- Choose appropriate schedules for activities

Documentation

- Document the how (transcript) & what (result)



Queries causing errors captured and reviewed _____
Queries with unstable execution plans reviewed _____
Simple scalability modeling performed _____

Backup and Recovery Testing

Results

Restore operation tested _____
Point-in-time recovery tested _____
 Total time required to restore _____
 CPU utilization _____
 Network utilization _____
 Disk/storage utilization _____
 Memory utilization _____
 Disk space required _____
 Network transfer required _____

Scheduling

- Choose appropriate schedules for activities
 - One-time tasks
 - Monthly
 - Weekly
 - On-demand, irregular
- Following slides list some main points
 - For much greater detail, see Resources at end.

One-Time Tasks

- Inspect the server and application
 - External systems
 - Storage
 - Privileges
 - Basic configuration settings

One-Time Tasks Cont'd

- Monitor and alerting
 - Be frugal
 - Avoid false positives
 - Monitor for problems, not heuristics
- Metrics and trending
 - Capture everything
 - Keep as long as practical

Weekly Tasks

- Predict replication lag
- Predict performance problems
 - Use cheap & fast “black-box” analysis
- Analyze workload performance
 - Find schema, indexing, data distribution, and query problems

Weekly Tasks Cont'd

- Review new queries
- Review schema changes
- Compare my.cnf to SHOW VARIABLES
- Validate backups

Weekly Tasks Cont'd

- Check for corruption
- Review all logs, prove they work
- Verify that alerts get through
- Check if it's time to restart

Monthly Tasks

- Review backup & recovery procedures & policy
- Test restore and recovery
- Review privileges and security
- Review archiving/purging plan
- Clean up accumulated cruft

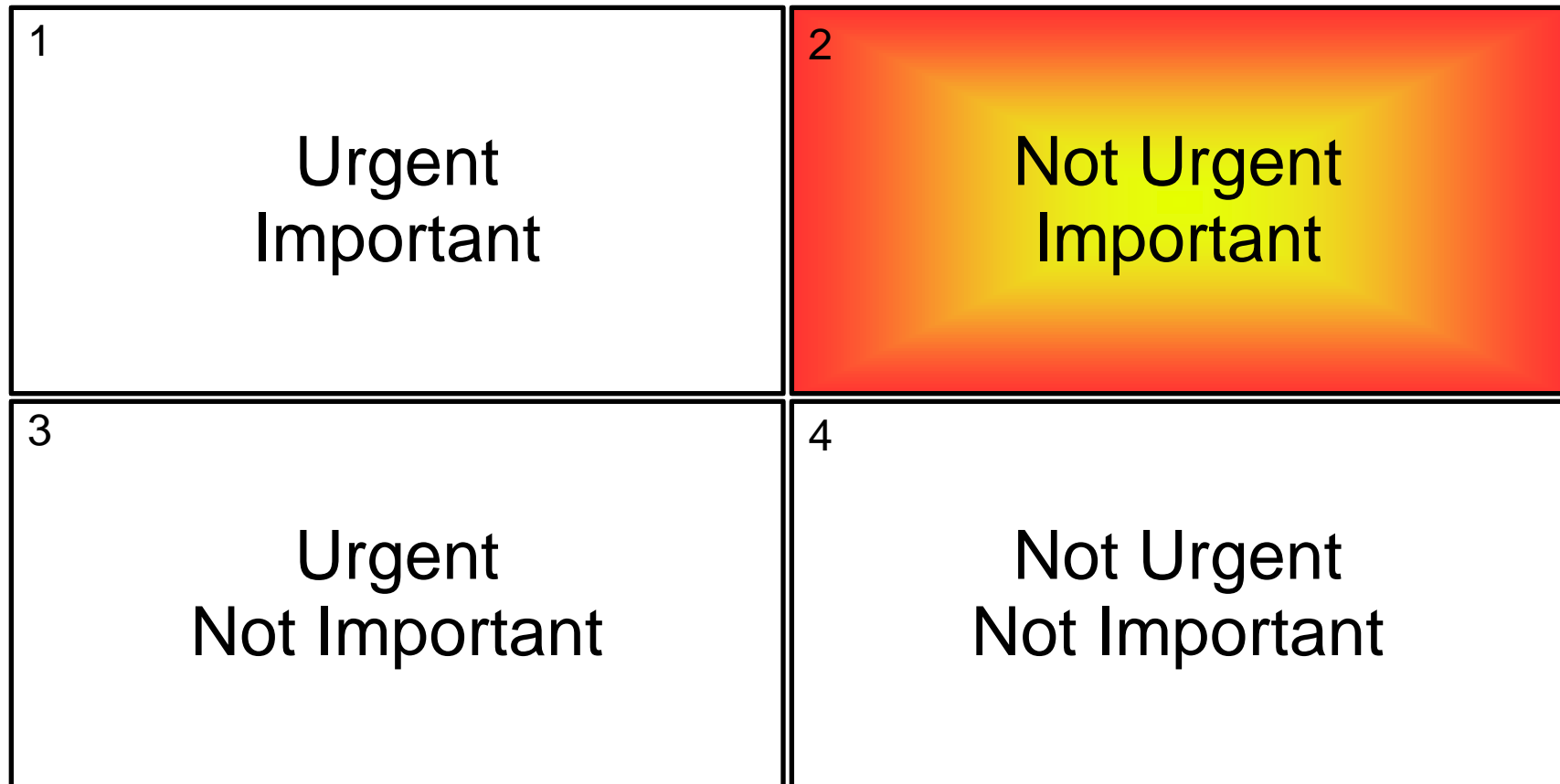
Irregular & On-Demand Tasks

- Check schema changes
- Review change logs for upgrades
- Restart systems
- Test upgrades
- Apply one-time tasks to new servers

Steven Covey's Four Quadrants

1 Urgent Important	2 Not Urgent Important
3 Urgent Not Important	4 Not Urgent Not Important

Steven Covey's Four Quadrants



Urgent; Not Important



The Second Quadrant

- Important; Not Urgent
 - Test restore & recovery
 - Look for early warnings
 - Manage and validate changes

Practice. Develop insight and understanding.

Conclusion

- Not sexy
- But, it works

Percona Live London Sponsors

Platinum Sponsor



Gold Sponsor



Silver Sponsors



Percona Live London Sponsors

Exhibitor Sponsors



Friends of Percona Sponsors

Couchbase



Monty Program



Tokutek



Media Sponsors



O'REILLY

Annual MySQL Users Conference

Presented by Percona Live

The Hyatt Regency Hotel, Santa Clara, CA

April 10th-12th, 2012

Featured Speakers

Mark Callaghan, Facebook

Jeremy Zawodny, Craigslist

Marten Mickos, Eucalyptus Systems

Sarah Novotny, Blue Gecko

Peter Zaitsev, Percona

Baron Schwartz, Percona

The Call for Papers is Now Open!

Visit www.percona.com/live/mysql-conference-2012/

Resources

- All research and results are available online:
 - percona.com/about-us/mysql-white-papers/
 - Causes of Downtime in Production MySQL Servers
 - Preventing MySQL Emergencies
 - Detailed activity lists and scheduling suggestions
 - Detailed advice on what to monitor



espen.braekken@percona.com

We're Hiring! www.percona.com/about-us/careers/



PERCONA
LIVE

www.percona.com/live