

**ORACLE®**

# **Datensicherung für MySQL - Möglichkeiten und Unterschiede**

Mario Beck – Principal Sales Consultant MySQL

# Agenda

- Database Backup Overview
- MySQL Enterprise Backup: Features & Benefits
- Database Backup Alternatives: Comparison
- MySQL Enterprise Backup: How it Works



# Database Backup Overview

# DBA Challenge

- Core responsibility for backup and recovery
- Challenges
  - Exponential growth
  - Shorter backup times
    - Impact to others
      - End Users
      - DBA Maintenance
- Increasing storage cost
- In bad times
  - Needs to work
  - Taking forever to recover



# Database Backup: Terms

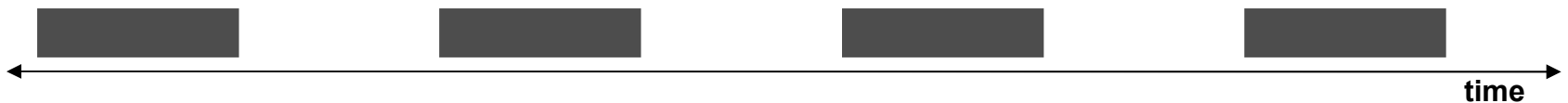
- Online Backup (aka “Hot” or “Online”)
  - Backup while database is running
  - Zero business interruption during backups
- Incremental Backup
  - Backup of data that has changed since the last full backup.
- Partial Backup
  - Backup of select tables
- Consistent Backup
  - Restoring all items to the same point in time.
- Point in Time Recovery (PITR)
  - Restore a database to any chosen state from the past.

# Define your Recovery Requirements

- Recovery Point Objective (RPO)
  - Defines tolerance for data loss
  - How frequently should backups be taken?
  - Is point-in-time recovery required?
- Recovery Time Objective (RTO)
  - Defines tolerance for downtime
  - Tiered RTO per level of granularity, e.g. database, table, row
- Determine backup retention policy
  - Onsite, offsite, long-term
- Determine acceptable impact during Backup

# Backup Method 1: Full

- Well Suited for:
  - Databases that can tolerate hours/days RPO
  - Application that require short RTO
  - Medium-High change between backups (e.g. over 30%)
  - Environments where disk can be allocated for 1x size of database
- Backup Strategy
  - Full backups with optional backup compression
  - Full backup archived to tape, as needed



## Backup Method 2: Full + Incremental

- Well suited for
  - Databases that can tolerate no more than a few hours RPO
  - Environments where disk can be allocated for 1x size of database
  - Applications that require a short backup window
- Backup strategy
  - Occasional Full backup, followed by more frequent incremental
  - To recover - apply Full and then applying 1 or more Incremental
  - Full backups archived to tape, as needed
  - Incremental Backups retained on-disk, as needed





## Backup Method 3: Full + Incremental + Log

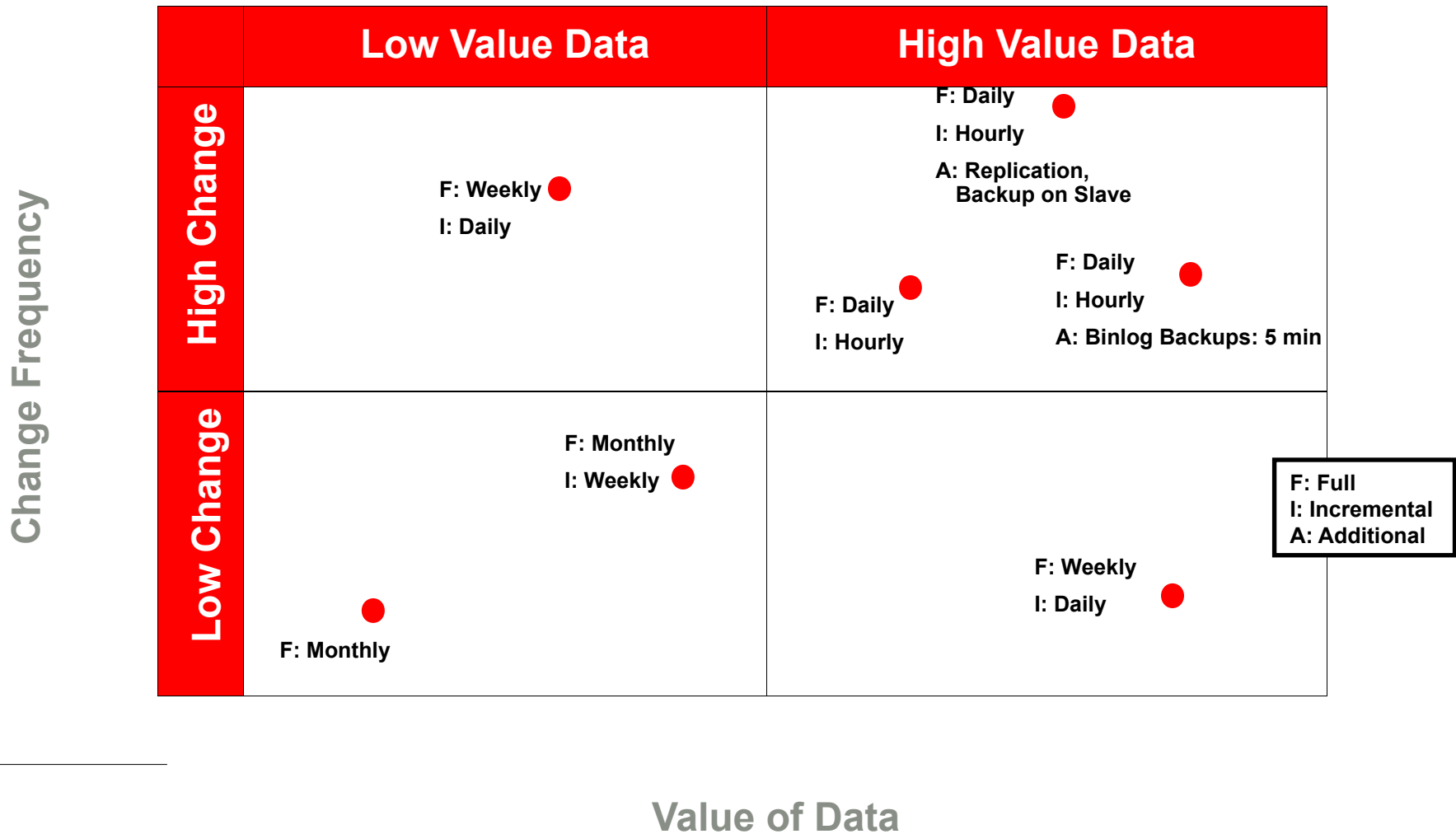
- Well suited for
  - Databases that can tolerate no more than a few minutes RPO
  - Environments where disk can be allocated for more than 1x size of database
- Backup strategy
  - Initial full backup, followed by incremental backups
  - Backup Transaction Logs (binlog)
  - To recover - apply Full and 1-n Incremental
  - Finally Roll Forward with Transaction Log to “minute” desired.



# Backup Method 4: Offload Backups to Slave (Replication)

- Well Suited for:
  - Databases that require no more than several minutes of recovery time, in event of failure
  - Environments that can preferably allocate symmetric hardware and storage for physical standby database
  - Environments whose backup storage infrastructure can be shared between master and slave database sites
- Backup Strategy
  - Setup Master / Slave replication
  - Slave acts as physical standby database
  - Run full and incremental backup on slave
  - Backup can be restored to master or slave database
  - Backups can be taken at each database for optimal protection

# Determining Backup Strategy



# Backup Strategies Comparison

Method	Backup Factors	Recovery Factors
Method 1: Full Backups	<ul style="list-style-type: none"> <li>• Longest Backup Times</li> <li>• Largest Storage Space</li> <li>• Save space with compression</li> </ul>	<ul style="list-style-type: none"> <li>• Easy to Recover</li> <li>• Fastest Restore Times</li> </ul>
Method 2: Full + Incremental Backup	<ul style="list-style-type: none"> <li>• Shortest Backup Time</li> <li>• Reduced Storage Requirements</li> <li>• Requires 1X production storage for copy</li> </ul>	<ul style="list-style-type: none"> <li>• Finer-grained Recovery</li> <li>• Slower Restore Times</li> <li>• First Restore Full Backup</li> <li>• Then Restore Incrementals</li> </ul>
Method 3: Full + Incremental + Log Backup	<ul style="list-style-type: none"> <li>• Added Storage Requirements</li> <li>• Requires more than 1X production storage for copy</li> </ul>	<ul style="list-style-type: none"> <li>• Finest-grained Recovery</li> <li>• Slowest Restore Times</li> <li>• First Restore Full Backup</li> <li>• Then Restore Incrementals</li> <li>• Then Apply Logs</li> </ul>
Method 4: Offload Backups Slave Replication	<ul style="list-style-type: none"> <li>• Used with 1 of the above</li> <li>• Frees Master for more workload</li> <li>• Requires 1X production hardware and storage for standby database</li> </ul>	<ul style="list-style-type: none"> <li>• Fast failover to standby</li> <li>• Backups are last resort, in event of double site failure or need to perform PITR</li> </ul>



# **Database Backup Types**

## **Advantages & Disadvantages**

# MySQL Backup Methods

- Hot Backup (online)
  - MySQL Enterprise Backup
- Export/Import (portable copies – a logical backup)
  - mysqldump
- Standby Copy (hot swap)
  - MySQL Replication
- Cold Backup (offline)
  - Simple File Copies when server is shutdown
- File System Volume Managers (snapshots)
  - LVM for example - create snapshot copy

# mysqldump

- Advantages
  - Good for small databases or tables
  - Good assurance that database files are not corrupt
  - Logical Backup – thus flexible and portable
- Disadvantages
  - Very slow restore times
  - Uses database processing cycles and resources
  - Not Online (requires Transaction or Locks on Tables in the database)
  - Not Incremental (requires a Full Backup every time)
  - Not Consistent (unless transaction is used)

# MySQL Replication

- Advantages
  - Rolling “snapshot”
  - Quick Recovery - via failover
  - Non-Blocking
  - Works well in conjunction with other backup options
- Disadvantages
  - Only latest “Point in Time” (point in time keeps moving forward)
  - Not historical
  - Not for archival purposes
  - Doesn’t protect from “oops”



# LVM Snapshots

- Advantages
  - Quick
  - Feature of Linux
  - Good to use in conjunction with backups
- Disadvantages
  - It's a snapshot
    - Still need to make a backup copy – which is “full” in size
  - Performance degrades with each concurrent snapshot
    - Snapshots need to be released
  - Cross File System Limitations

# MySQL Enterprise Backup

- Advantages
  - Physical Backup so Fast – esp. restores
  - Flexible - many options
  - Archival
  - Scalable
  - Consistent
  - Supported
- Disadvantages
  - Requires some planning

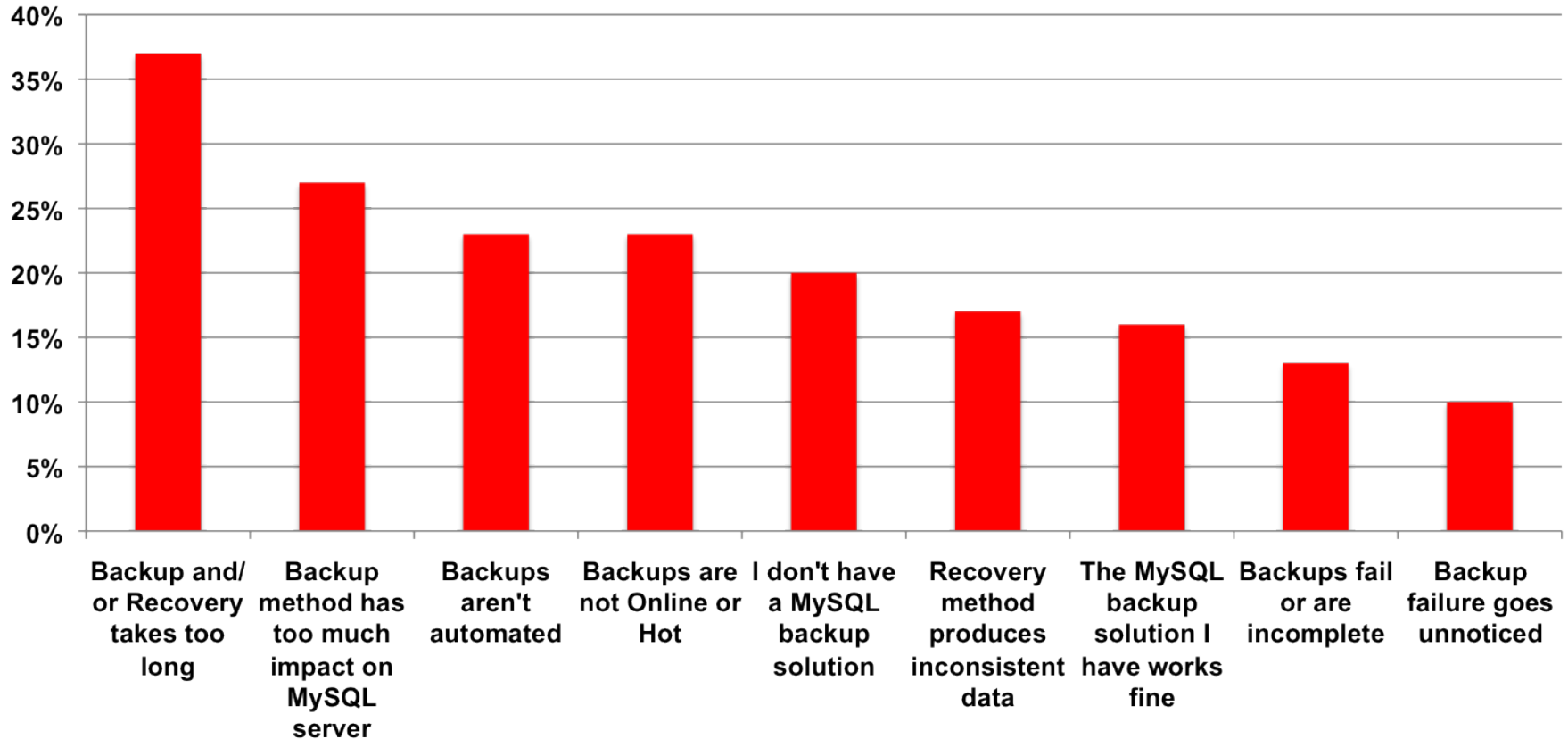


# **MySQL Enterprise Backup**

## **Features & Benefits**

# MySQL Backup Concerns

What concerns you most about your current MySQL backup solution?



- Backup & Recovery Performance is the #1 Concern

# MySQL Enterprise Backup Benefits

- Online “Hot” Backup (Non-blocking)
  - Reads and Writes to InnoDB
  - Reads for MyISAM tables
- High Performance
  - Backup: >3x faster than mysqldump (export)
  - Restore: >10x than mysqldump recovery
- Consistent Backups
  - Combined with Binlogs: Point in Time Recovery
- Compression
  - Multi-level compression
  - Save 70% or more of the storage required
- Incremental Backups
  - Saves backup time and storage

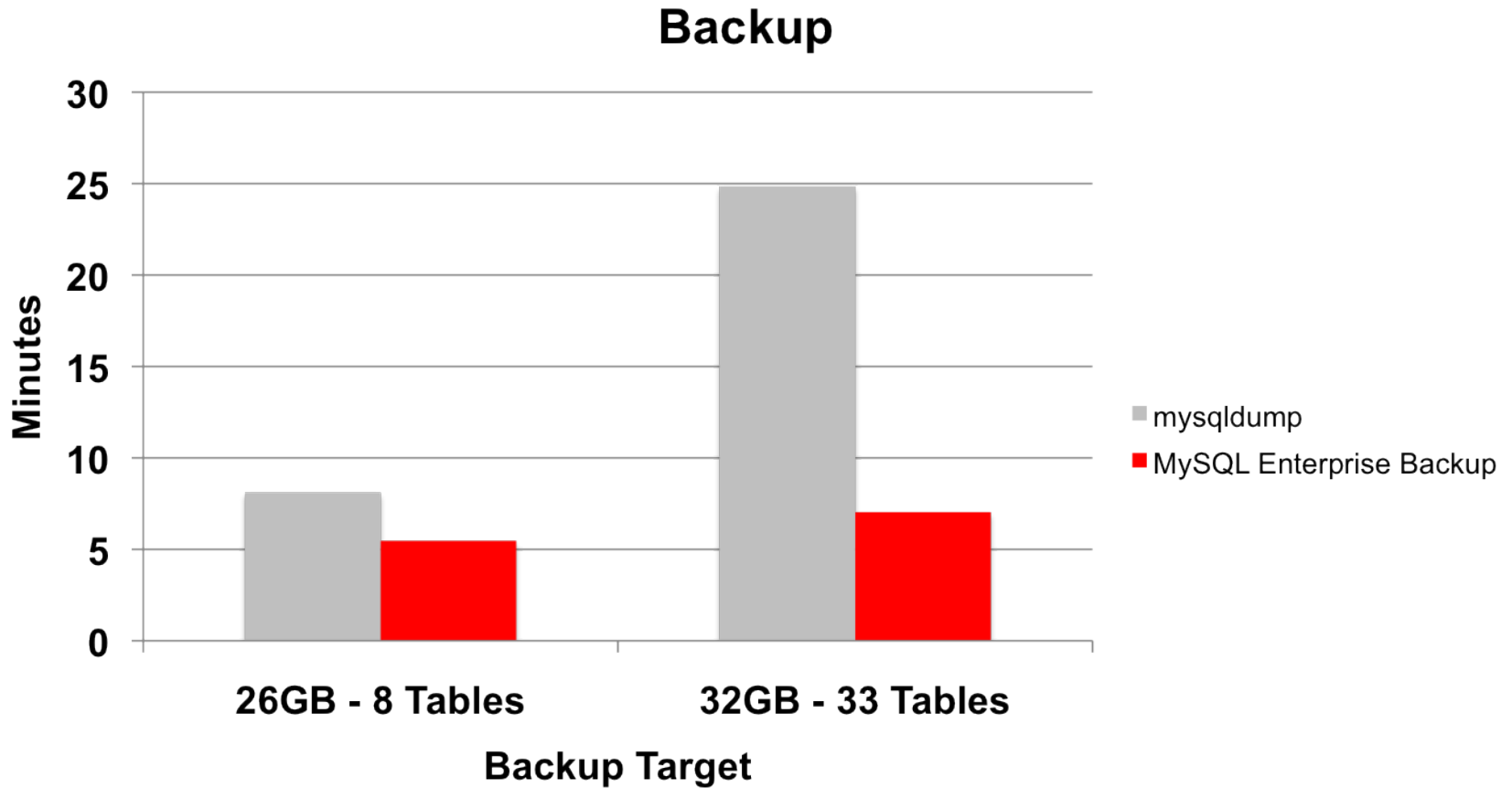
# Benefits

- Partial Backups
  - Copy only important data
- Cross Plattform
  - Windows, Linux, Unix
  - Very important for ISV
- Reliable
  - Proven for 7+ Years
- Scalable for Large Databases
  - No Database Size Limitations
- Easy to automate
  - Easily integrate within various scheduling systems
  - Examples: cron, OSB scheduler, others

# MySQL Enterprise Backup 3.5: New Features

- Incremental backup
- Support of InnoDB Barracuda file format
- Backup of compressed tables
- Backup of partition files
- Backup of in-memory database
  - with `--exec-when-locked` option
- Adds mysql system tables to keep backup status, progress, and history

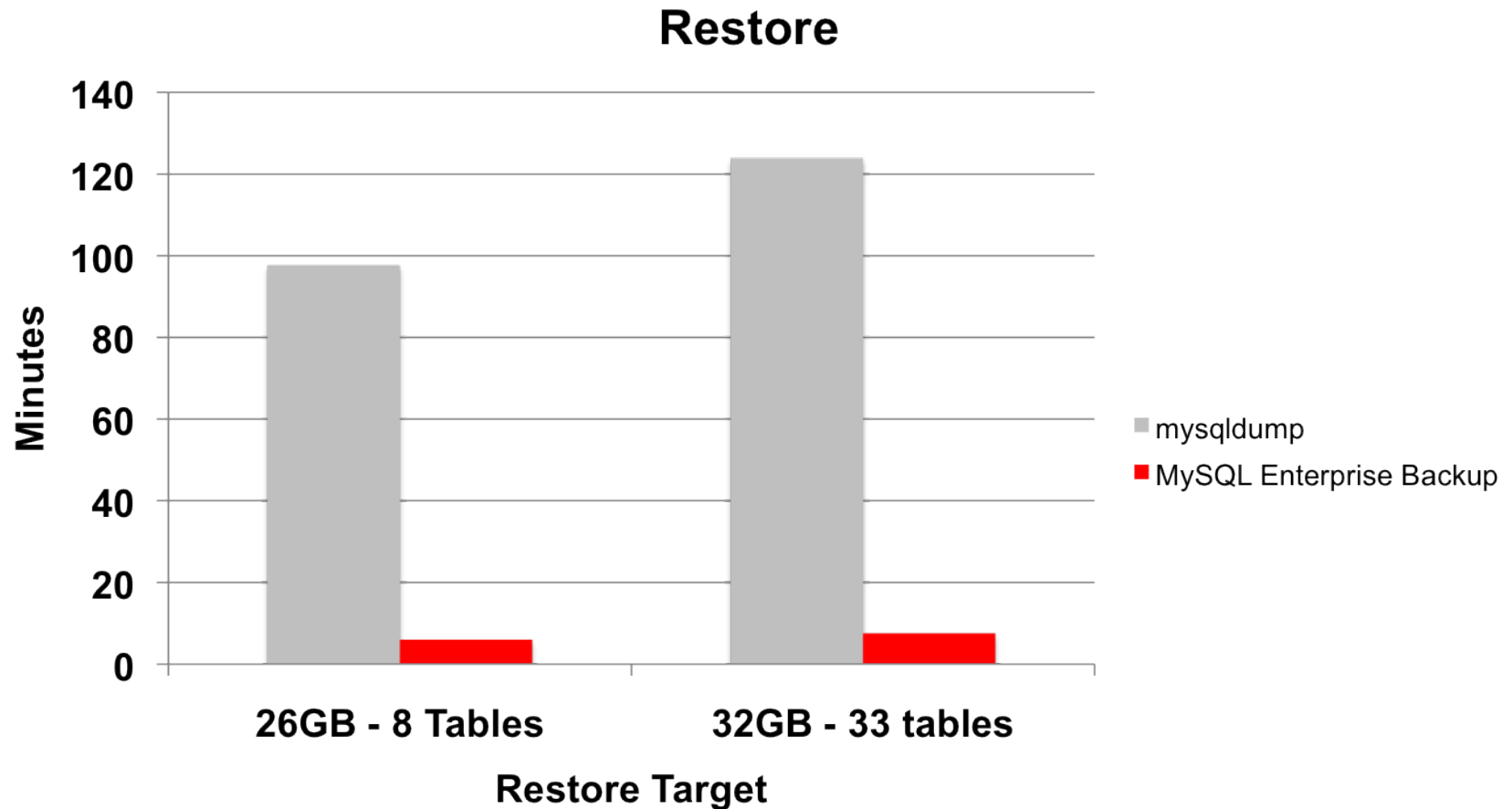
# High Performance Backups



Backups are up to 3.5x Faster than MySQL Dump

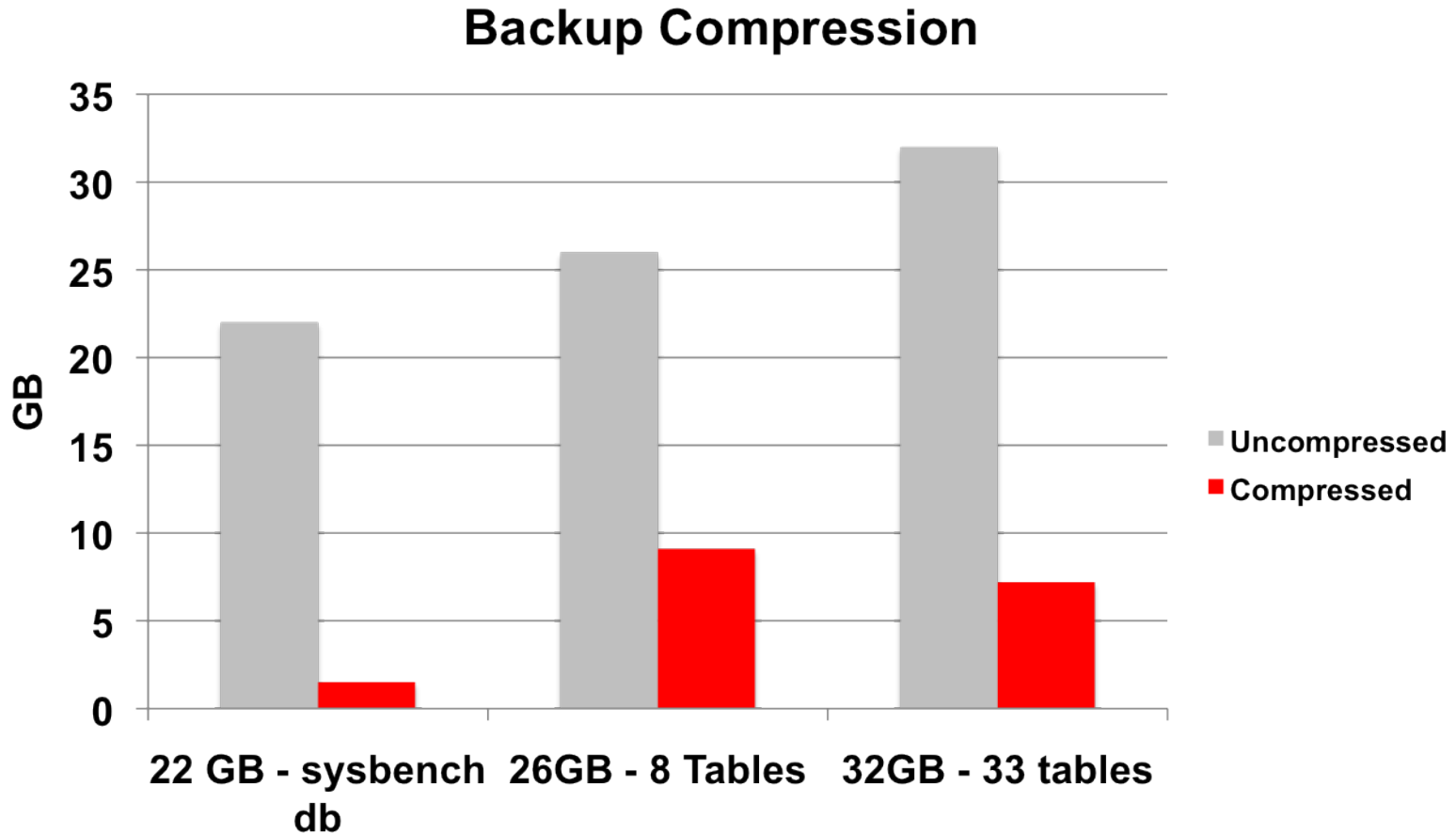


# High Performance Restore



Restore is up to 16x Faster than MySQL Dump

# Backup Compression Storage Savings



Backup size reduced from 65% up to 93%



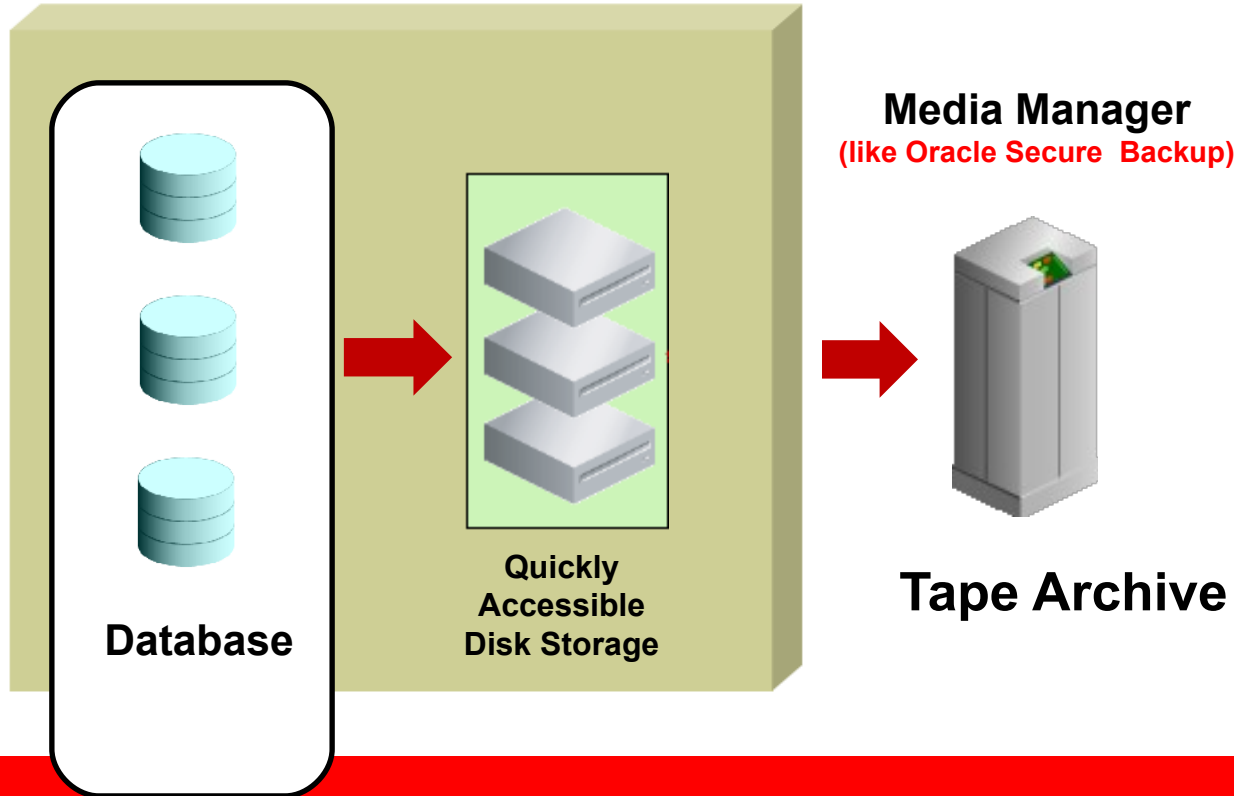
# **MySQL Enterprise Backup**

## **How it Works**

# MySQL Enterprise Backup



- MySQL Enterprise Backup CLI
- Oracle Secure Backup



Intrinsic knowledge of database file formats

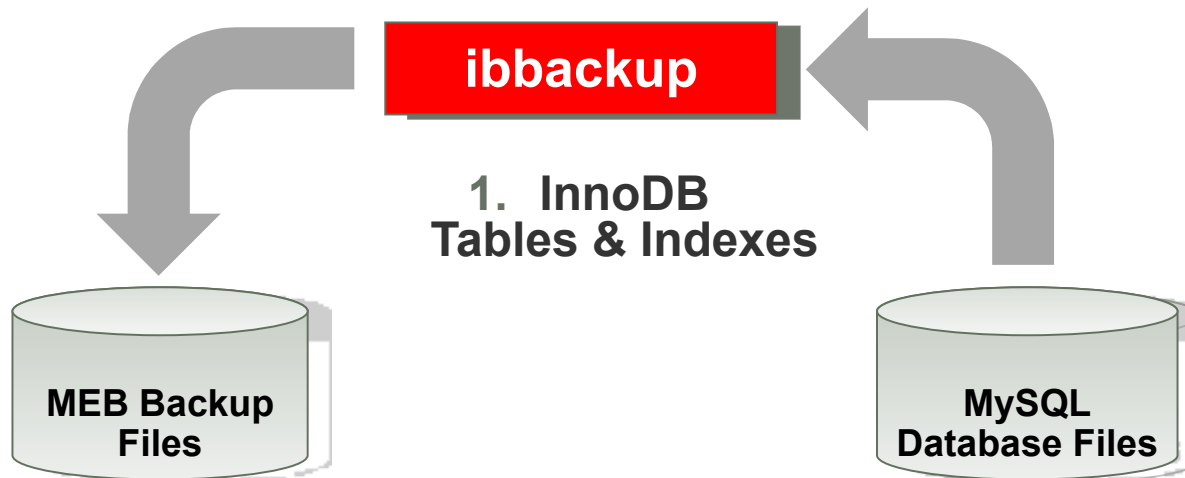
- Block Validation
- Tablespace/Data file recovery
- Unused Block Compression
- Consistent Recovery
- File Compression

# MySQL Enterprise Backup: Terms

- `mysqlbackup` : backup executable which includes InnoDB, MyISAM and other MySQL Data. `mysqlbackup` is a compatible replacement for the `innobackup` pre 3.5.1 and includes additional features and capabilities
- `ibbackup`: finer grained raw innodb backup executable for innodb files alone
- `binlog`: contains database changes – eg DDL and DML
- `LSN`: Log Sequence Number – the unique monotonically increasing id for each change in the binlog
- `Ibdata`: system tablespace files
- `.ibd`: single table space file

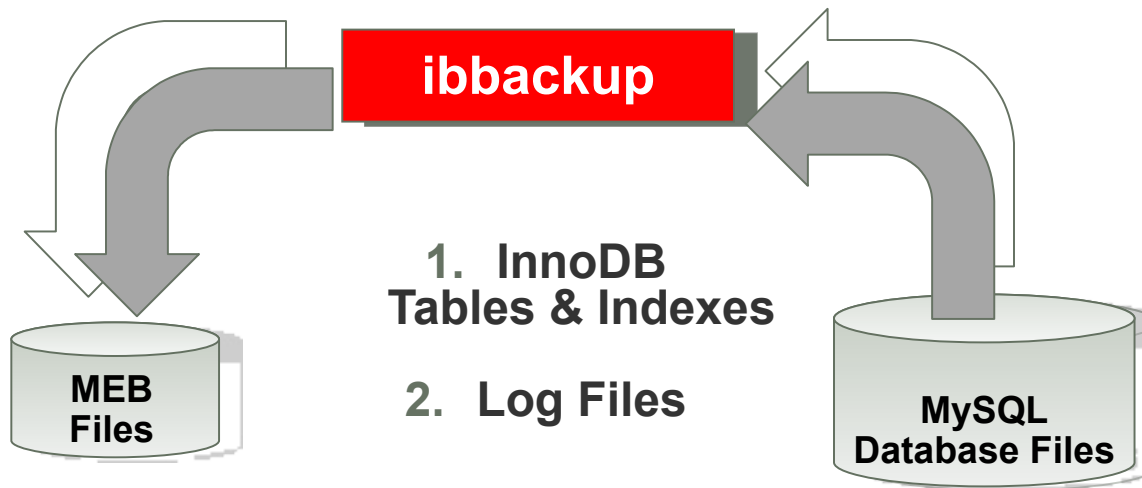
# How it Works: Backup for InnoDB

- Step 1: Backing Up InnoDB Data Files
  - Copies and compresses InnoDB data files
    - System Database (ibdata) & Single-table Tablespaces (.ibd)
  - Produces “Fuzzy Backup”
    - Backup of data files doesn’t correspond to any specific log sequence number (LSN)
    - Different database pages are copied at varying times



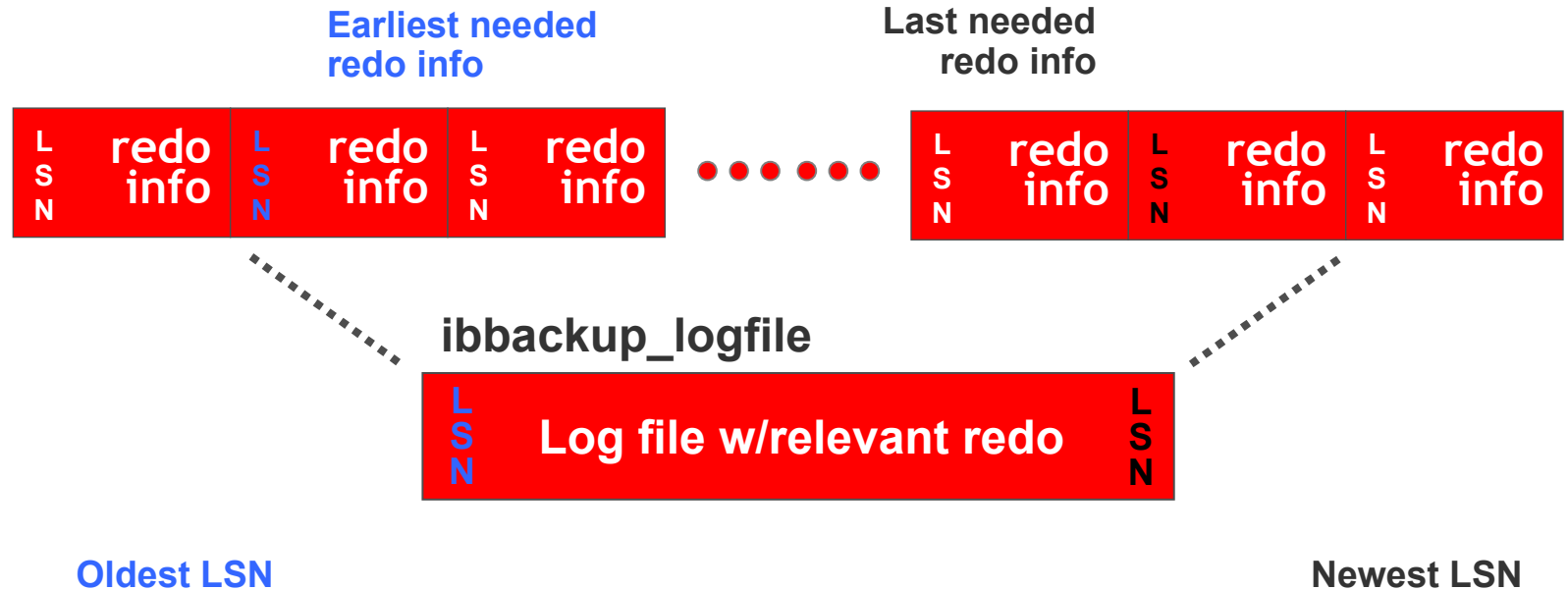
# How it Works: Backup for InnoDB

- Step 2: Backing up InnoDB Log Files
  - Copies Log Records accumulated during data file copy
  - All redo records with LSNs during data file copy
  - If needed compresses backup files



# How it Works: Backing up InnoDB Log Files

## Log File



- Copies portion of the log file that contains all required redo information
- Covers the time from beginning to end of data backup
- Recovers all data blocks modified after copied to compressed data file

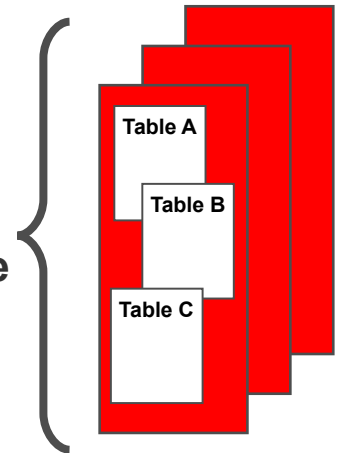


# Full & Partial Backups

- Backup contains all tables in system tablespace
  - Plus those separate tables that match the pattern
- When using “file per table”, you can backup a subset of InnoDB tables
  - Tables included in the backup are specified with regular expressions
  - Use the -- include option

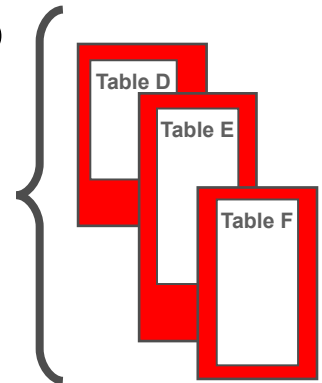
## Full Backup

Multiple tables & indexes in the system tablespace (ibdata files)



## Partial Backup

One table & indexes per file (.ibd files)



# mysqlbackup (innobackup) Examples

- Full Backup

```
mysqlbackup --user=dba --password=xyz --compress /etc/my.cnf /backups
```

- Incremental Backup

- The backup only contains changed data

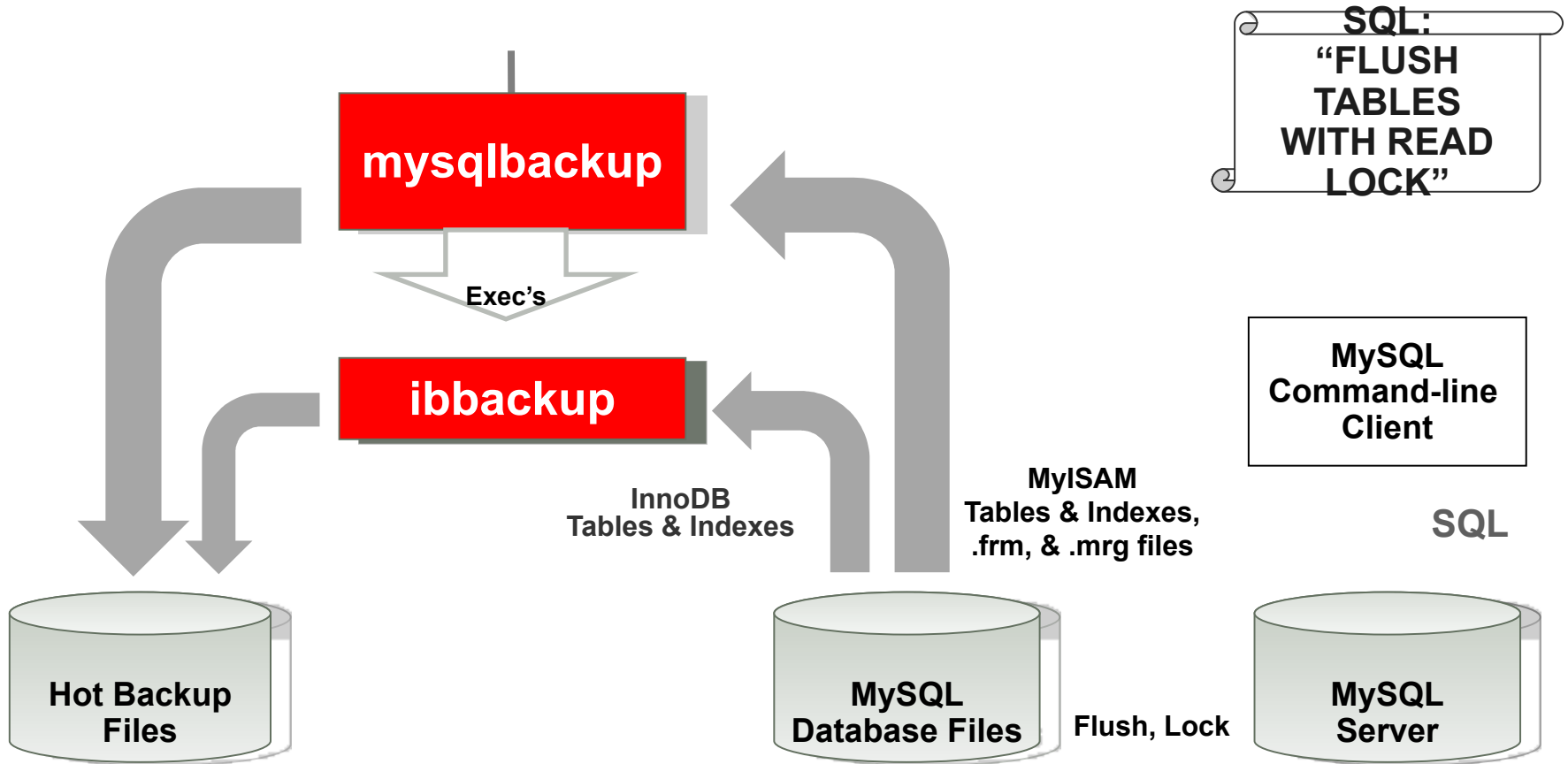
```
mysqlbackup --incremental --lsn 2261747124 /etc/my.cnf /incr-backup
```

- Partial

- The backup contains tables in test database that match the `.ib.*` regular expression.

```
mysqlbackup --include 'test\.ib.*' /etc/my.cnf /backups
```

# How mysqlbackup Works

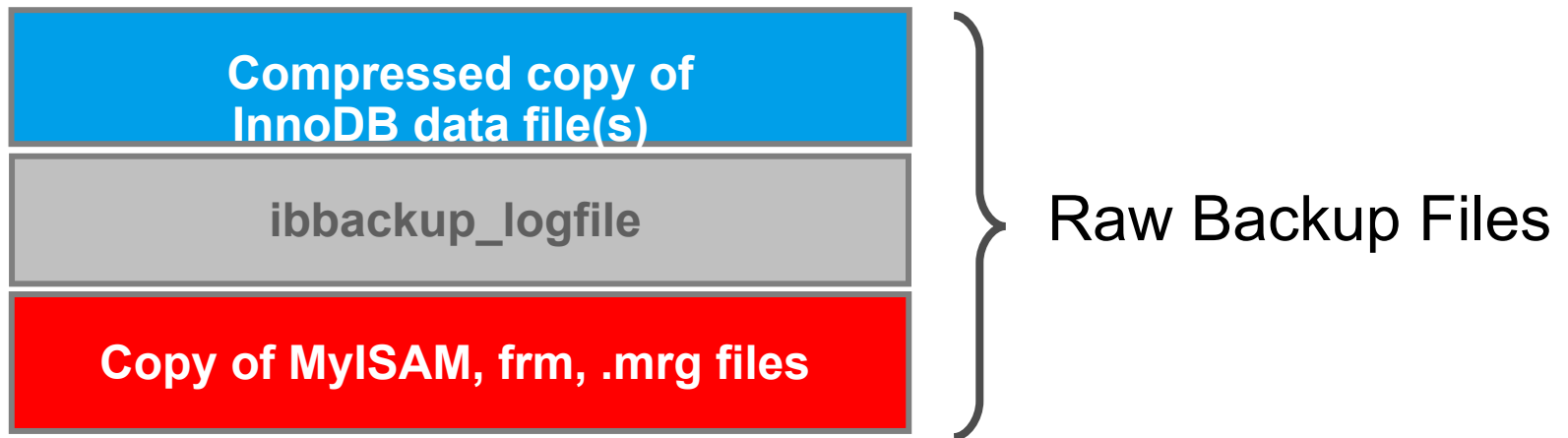


## Tips: InnoDB and MyISAM Backup

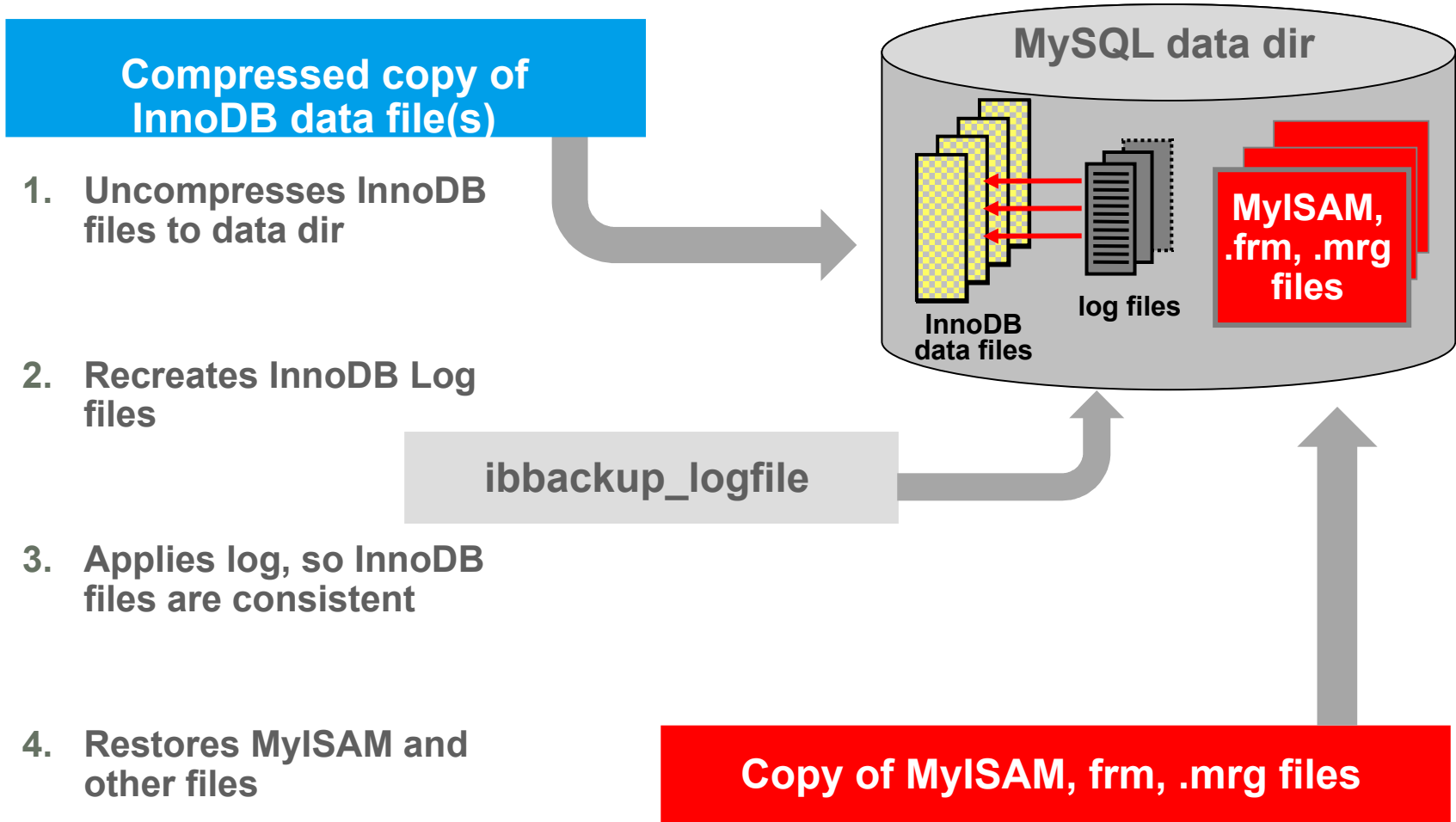
- InnoDB tables are fully accessible during backup
  - Insert, Update & Delete
- MyISAM tables cannot be updated during backup
  - Uses FLUSH TABLES WITH READ LOCK near the end of the backup
- Works best if ...
  - Wait for insert/update/delete transactions during MyISAM backup
  - Do not run long SELECT queries during the backup
  - MyISAM tables are small, thus copied quickly

## Tips: “Raw Backup” Files

- The “raw backup” files from backup phase cannot be directly consumed by MySQL
- These files can be copied to media
- The database must be “restored” first
- Use `mysqlbackup` to restore database before use



# How it Works: Restoring a Database



## Restoring a Database Con't...

- MEB restore rolls forward data files to a common point in time (the time at the end of backup)
- After restore, MEB Backup prints the location in the binlog for the next SQL operation that executed after the backup completed
- Note: the restore phase need not run on database server host
  - You can perform recovery on any machine, and copy recovered files to your database server host

## Summary MySQL Enterprise Backup

- Offers best performance for backup
- Offers best performance for restore
- Adds minimal load to MySQL server
- No impact on application (online backup)
- Easily integratable into you application



# MySQL Backup Types: Comparison

	mysqldump	LVM Snapshots	MySQL Replication	MySQL Enterprise Backup
Full Backup	✓	✓	✓	✓
Incremental Backups	✗	✓	✗	✓
Partial Backups	✓	✗	✗	✓
Compression Support	✗	✗	✗	✓
Allows updates	✗	✗	✓	✓
Point in Time - Consistent	✗	✓	✓	✓
Backup Speed	Poor	Good	Very Good	Very Good
Recovery Speed	Very Poor	Good	Very Good	Very Good
Partial Restore	✓	✗	✗	✓
Corruption Detection	✓	✗	✗	✓
Meets Regulatory Archive Req.	✓	✗	✗	✓
Supports DDL	✓	✗	✗	✓

## Additional Resources

- Product Information

<http://www.mysql.com/products/enterprise/backup.html>

- Documentation

<http://dev.mysql.com/doc/mysql-enterprise-backup/3.5/en/index.html>

- Backup Forum

<http://forums.mysql.com/list.php?28>

- Download (30 Day Trial)

<http://edelivery.oracle.com/>

**ORACLE®**