# MySQL Backup & Restore Benchmark Report

## System Configuration

* **OS:** Ubuntu 20.04.6 LTS
* **Kernel:** 5.4.0-208-generic
* **CPU:** Intel® Core™ i7-7700 @ 3.60GHz
* **Memory:** 3.8GB RAM
* **Storage:** 48GB (21GB free)
* **MySQL Version:** 8.0.27
* **Threads Used:** 2 (for mysqlpump and mydumper)

## 1. Backup & Restore Performance Comparison

| Tool | Backup Time (real) | Restore Time (real) |  | Live Progress |
| --- | --- | --- | --- | --- |
| **mysqldump** | 2m51.053s | 24m27.044s |  | No |
| **mysqlpump** | 2m26.788s | Same as mysqldump |  | Yes |
| **mydumper** | 1m42.770s | 24m25.213s |  | No |

## 2. Backup Performance Analysis

* mydumper was the **fastest** (1m42s), significantly faster than mysqldump (2m51s) and mysqlpump (2m26s).
* mysqlpump was **slightly faster** than mysqldump, benefiting from parallelism but still slower than mydumper.

## 3. Restore Performance Analysis

* Both mysqldump and mydumper took around **24m25s** for restoration.
* mysqlpump restores using the **same method as mysqldump**, meaning its restore time is also **24m27s**.

## 4. Additional Features

| Feature | mysqldump | mysqlpump | mydumper |
| --- | --- | --- | --- |
| Parallelism | No | Yes (2 threads) | Yes (2 threads) |
| Live Progress | No | Yes (Dump progress: X/Y tables, N/M rows) | No |
| Compressed Output | No | No | Yes |
| Binary Log Compatible | Yes | Yes | Yes |

## 5. Key Takeaways

1. **Best for Speed:** mydumper
   * Fastest backup speed (1m42s)
   * Restore speed similar to mysqldump (24m25s)
   * Supports parallelism
2. **Best for Monitoring:** mysqlpump
   * Slightly faster than mysqldump
   * Shows real-time progress
   * Parallelism can improve performance
3. **Most Common & Reliable:** mysqldump
   * No parallelism, but widely used and stable
   * Slowest backup speed

### Useful mysqlpump Commands That Offer Advantages Over mysqldump

mysqlpump is a **parallel backup tool** with **extra features** not available in mysqldump. Below are some key **commands and options** that make mysqlpump superior in certain scenarios:

## 1. Enable Parallelism for Faster Backups

Unlike mysqldump, mysqlpump supports **multi-threaded backup** to speed up the process.

### Command:

mysqlpump -u root -p --default-parallelism=4 > backup.sql

**What It Does:**

* Uses **4 threads** to dump tables in parallel.
* **Faster than mysqldump**, especially for large databases.

## 2. Show Live Progress During Backup

mysqlpump provides **real-time progress updates**, unlike mysqldump, which runs silently.

### Command:

mysqlpump -u root -p --default-parallelism=4 --verbose > backup.sql

**What It Does:**

* Shows which **tables** are being dumped.
* Displays **row count progress** during the backup.

## 3. Exclude Specific Databases or Tables

Unlike mysqldump, mysqlpump can **exclude** databases or tables **without listing each one manually**.

### Command:

mysqlpump -u root -p --exclude-databases=test\_db,logs\_db > backup.sql

**What It Does:**

* Excludes **test\_db** and **logs\_db** from the backup.
* mysqldump requires **explicit table lists**, making exclusion harder.

## 4. Backup Specific Schema Objects (Triggers, Views, Routines)

With mysqldump, you need separate commands, but mysqlpump allows **selective schema backup** easily.

### Command:

mysqlpump -u root -p --routines --triggers --events --users > schema\_backup.sql

**What It Does:**

* Dumps **stored procedures, triggers, events, and user accounts**.
* mysqldump requires multiple commands to achieve this.

## 5. Backup Only Table Structures Without Data

You can dump only **table structures** (DDL) without dumping data.

### Command:

mysqlpump -u root -p --no-data > schema\_only.sql

**What It Does:**

* Saves just the **table structure**, useful for schema migrations.
* **Equivalent to mysqldump --no-data** but faster with parallelism.

## 6. Backup Only Data Without Table Structures

You can dump only **table data** (DML) without schema definitions.

### Command:

mysqlpump -u root -p --no-create-db --no-create-info > data\_only.sql

**What It Does:**

* Exports only **INSERT statements**, skipping schema definitions.

## 7. Export Data with Column Names in INSERT Statements

When restoring, this prevents issues with column order changes.

### Command:

mysqlpump -u root -p --insert=both > backup\_with\_columns.sql

**What It Does:**

* Generates INSERT INTO table\_name (col1, col2) VALUES (...) instead of plain INSERT VALUES (...).
* Helps avoid errors if column order changes in the future.

## 8. Compress the Backup File in gzip Format

Unlike mysqldump, mysqlpump can **compress** the output **on-the-fly**.

### Command:

mysqlpump -u root -p | gzip > backup.sql.gz

**What It Does:**

* Creates a **compressed** backup file (.gz).
* Saves **disk space** compared to raw .sql dumps.

## 9. Backup & Restore User Accounts Separately

With mysqldump, you have to **manually export users**, but mysqlpump can handle it automatically.

### Backup Users & Privileges:

mysqlpump -u root -p --users > users\_backup.sql

### Restore Users:

mysql -u root -p < users\_backup.sql

**What It Does:**

* Exports **MySQL user accounts and grants**, making it easier to migrate users between servers.

## 10. Use Table Filtering with Wildcards (Regex Matching)

mysqlpump allows **wildcard filtering** for tables, unlike mysqldump.

### Example: Backup Only Tables Matching sales\_\*

mysqlpump -u root -p --include-tables='sales\_\*' > sales\_backup.sql

**What It Does:**

* Backs up **only** tables **starting with sales\_**.