EffectiveMySQL.com

Its all about Performance and Scalability

MySQL Best Practices

for DBAs and Developers

Volume II

Ronald Bradford
http://ronaldbradford.com

2011.04

AGENDA

- Schema optimizations
- Instrumentation
- Monitoring
- Testing

PART 1 RECAP

- Essential MySQL configuration
  - Using SQL_MODE
- MySQL user security
  - Why SUPER is BAD
- Improving your SQL

See Part I for more tips

AUTHOR

- 2011 - All time top blog contributor to Planet MySQL
- 2010 - Published Author of Expert PHP & MySQL
- 2010 - Oracle ACE Director (first in MySQL)
- 2009 - MySQL community member of the year
- 22 years of RDBMS experience, 12 years with MySQL
  - MySQL Inc (2006-2008), Oracle Corp (96-99)
- Provide independent consulting/ Available NOW

Schema Design

DESIGN

- Optimal Data Types
  - Saving Disk Space
- Naming Standards

MYSQL DATA TYPES

- Numeric Data Types
  - Oracle has 1
  - MySQL has 9

```
TINYINT, SMALLINT, MEDIUMINT, INT,
BIGINT, FLOAT, DOUBLE, DECIMAL, BIT
```

AUTO INCREMENT

- Can reduce index space by 50+%
- Better memory usage, less I/O

- Don't use `BIGINT AUTO_INCREMENT`
- Use `INT UNSIGNED AUTO_INCREMENT`
- `BIGINT` is 8 bytes
- `INT` is 4 Bytes
- `INT UNSIGNED` stores 4.3 billion values

http://ronaldbradford.com//blog/bigint-v-int-is-there-a-big-deal-2008-07-18/

INTEGER

INT(1) is not what it looks like

- INT(1)
  - This is not 1 byte, it's 4 bytes
  - (1) is only for client display only
  - Client with 10+ flags using INT(1)
  - 40 bytes reduced to 10 bytes (per row)
    - or 2 bytes using bit operators

DATES

- MySQL supports
  - `DATE, DATETIME, TIMESTAMP, YEAR`
- `TIMESTAMP` for Epoch values
  - `TIMESTAMP` is 4 bytes
  - `DATETIME` is 8 bytes
  - Supports `DEFAULT CURRENT_TIMESTAMP`
- Neither store milliseconds

BEAUTY OF ENUM

- Values Check Constraint
- Ideal for static codes
- Compact - i.e. 1 byte for 'True', 'False'
- Human readable values
- 5.1 Non blocking ALTER

BEAUTY OF ENUM
EXAMPLE

```
CREATE TABLE enums (
  flag1  CHAR(1) NULL COMMENT 'T or F, Y or N',
  flag2  TINYINT NULL COMMENT '0 or 1',
  flag3  BIT  NULL COMMENT 'True or False',
  flag4  ENUM ('True','False') NULL,
  flag5  VARCHAR(5) NULL
);
INSERT INTO enums(flag4, flag1, flag2)
VALUES ('True', 'T', 1), ('False', 'F', 0);
SELECT flag1, flag2, flag4 FROM enums;
+-------+-------+-------+
| flag1 | flag2 | flag4 |
+-------+-------+-------+
| T     |     1 | True  |
| F     |     0 | False |
+-------+-------+-------+
```

- All are 1 Byte, which is more readable?

DISK FOOTPRINT

- Compress large text data
  - 90% saving on XML data
- Store static data in files
  - Avoids DB handling overhead

NAMING STANDARDS

- Primary Key/Foreign Key
  - e.g. user_id, not id & user_id
- Do not use reserved words
  - e.g. date
- Be consistent with object names

SCHEMA MGMT

- Always have current schema.sql
- Use patch/revert SQL for upgrades
- See http://schemasync.org

- Reproducibility
- Upgrade/Downgrade path

API

API

- One code path for business functionality
- Implied business documentation
- Enforced data exchange standard
- **Testability**

API

- Technology independence
- Business specification dependence (API)
- Stress testable

API

- Define inputs
- Define outputs

- Define throughput
- Define latency

IMPLEMENTATION

- Developers do not have access to DB
- Developers do not write SQL

http://oa-api.ronaldbradford.com/

Monitoring

If you don't have monitoring in place, make it your top priority

MONITORING

- Monitoring/Alerting
  - Graphical
  - Historical
  - Necessary
  - Generally missing/incomplete
  - Useless for real-time analysis

MONITORING

- Monitoring
- Alerting
- Dashboard
- Public Status

Successful Scalability

http://ronaldbradford.com/blog/successful-mysql-scalability-presentation-2010-09-17/

DASHBOARD

- Dashboard
  - The state of NOW
  - Sampling at 1s/3s/5s
    - e.g. 0.1% of throughput

EXAMPLE APP

- Etsy statsd

```
StatsD::increment("grue.dinners");
StatsD::timing("grue.dinners", microtime(true) - $start)
StatsD::increment("adventurer.heartbeat", 0.1);
```

https://github.com/etsy/statsd

http://codeascraft.etsy.com/2011/02/15/measure-anything-measure-everything/

Instrumentation

INSTRUMENTATION

- Creating one primary abstract DB call
- Enable logging of ALL SQL statements
- Enable embedded HTML output
  - Total Execution Time/Count
  - Individual SQL Execution Time/SQL

- Enable runtime analysis via browser
- No additional tools needed to gather

EffectiveMySQL.com - Its all about **Performance** and **Scalability**

Application Toolset

Memory Usage

5641240

17 queries logged

1. SELECT [illegible]
2. DELETE [illegible]
3. SELECT * FROM j15svn_session WHERE session_id = 'ffb66b9026667c28f956dda21870ce00'
4. INSERT INTO j15svn_session (`session_id`,`time`,`client_id`) VALUES ('ffb66b9026667c28f956dda21870ce00','1186590522',
5. SELECT id, name, folder, element, published, params FROM j15svn_plugins WHERE published >= 1 AND access <= 0 ORDER BY or
6. SELECT m.*, c.`option` AS component FROM j15svn_menu AS m LEFT JOIN j15svn_components AS c ON m.componentid = c.id WHERE
7. SELECT * FROM j15svn_components WHERE parent = 0
8. SELECT template, menuid FROM j15svn_templates_menu WHERE client_id = 0
9. SELECT a.id, a.title, a.title_alias, a.introtext, a.sectionid, a.state, a.catid, a.created, a.created_by, a.created_by_a
   a.checked_out_time, a.publish_up, a.publish_down, a.images, a.attribs, a.urls, a.ordering, a.metakey, a.metadesc, a.acce
   a.alias) ELSE a.id END AS slug, CHAR_LENGTH(a.`fulltext`) AS readmore, u.name AS author, u.usertype, g.name AS groups,
   j15svn_content_frontpage AS f ON f.content_id = a.id LEFT JOIN j15svn_categories AS cc ON cc.id = a.catid LEFT JOIN j15s
   ON a.access = g.id WHERE 1 AND a.access <= 0 AND a.state = 1 AND (publish_up = '0000-00-00 00:00:00' OR publish_up <= '
   OR publish_down >= '2007-08-08 16:28:42') ORDER BY f.ordering LIMIT 0, 9
10. SELECT a.id, a.title, a.title_alias, a.introtext, a.sectionid, a.state, a.catid, a.created, a.created_by, a.created_by_a
    a.checked_out_time, a.publish_up, a.publish_down, a.images, a.attribs, a.urls, a.ordering, a.metakey, a.metadesc, a.acce
    a.alias) ELSE a.id END AS slug, CHAR_LENGTH(a.`fulltext`) AS readmore, u.name AS author, u.usertype, g.name AS groups,
    j15svn_content_frontpage AS f ON f.content_id = a.id LEFT JOIN j15svn_categories AS cc ON cc.id = a.catid LEFT JOIN j15s
    ON a.access = g.id WHERE 1 AND a.access <= 0 AND a.state = 1 AND (publish_up = '0000-00-00 00:00:00' OR publish_up <= '
    OR publish_down >= '2007-08-08 16:28:42') ORDER BY f.ordering
11. SELECT id, title, module, position, content, showtitle, control, params FROM j15svn_modules AS m LEFT JOIN j15svn_module
    m.access <= 0 AND m.client_id = 0 AND (mm.menuid = 1 OR mm.menuid = 0) ORDER BY position, ordering
12. SELECT id, title FROM j15svn_polls WHERE id = 14 AND published = 1
13. SELECT id, text FROM j15svn_poll_data WHERE pollid = 14 AND text <> "" ORDER BY id
14. SELECT guest, usertype, client_id FROM j15svn_session WHERE client_id = 0
15. SELECT a.*, CASE WHEN CHAR_LENGTH(a.alias) THEN CONCAT_WS(":", a.id, a.alias) ELSE a.id END AS slug, CASE WHEN CHAR_LENG
    AS catslug FROM j15svn_content AS a LEFT JOIN j15svn_content_frontpage AS f ON f.content_id = a.id INNER JOIN j15svn_cat
    ON s.id = a.sectionid WHERE (a.state = 1 AND s.id > 0) AND (a.publish_up = "0000-00-00 00:00:00" OR a.publish_up <= "
    00:00:00" OR a.publish_down >= "2007-08-08 16:28:42") AND a.access <= 0 AND cc.access <= 0 AND s.access <= 0 AND s.publ
16. SELECT a.*, CASE WHEN CHAR_LENGTH(a.alias) THEN CONCAT_WS(":", a.id, a.alias) ELSE a.id END AS slug, CASE WHEN CHAR_LEN
    AS catslug FROM j15svn_content AS a INNER JOIN j15svn_categories AS cc O
    "0000-00-00 00:00:00" OR a.publish_up <= "2007-08-08 16:28:42") AND ([illegible]
    AND cc.access <= 0 AND s.access <= 0 AND s.published = 1 AND cc.publish
17. SELECT a.*, CASE WHEN CHAR_LENGTH(a.alias) THEN CONCAT_WS(":", a.id, a.
    catslug FROM j15svn_content AS a INNER JOIN j15svn_categories AS cc ON c
    cc.access <= 0 AND s.access <= 0 AND (a.publish_up = "0000-00-00 00:00:00" OR a.publish_up <= "2007-08-08 16:28:42") A
    "2007-08-08 16:28:42") AND cc.id = 3 AND cc.section = s.id AND cc.published = 1 AND s.published = 1 ORDER BY a.ordering

Language Files Loaded

- SQL Statements Example

111 ms	111 ms	1	/home/sitedirs/example/site/account/column_right.cfm
109 ms	109 ms	1	/home/example/public_html/includes/marketplace.cfm
15 ms	15 ms	1	/home/sitedirs/example/site/layout/leaderboard.cfm
14 ms	14 ms	1	/home/example/public_html/includes/layout/leaderboard.cfm
13 ms	13 ms	1	/home/example2/public_html/includes/scripts/ad.cfm
13 ms	13 ms	1	/home/sitedirs/example/site/account/column_body_left.cfm
12 ms	12 ms	1	/home/example/public_html/account/includes/main.cfm
8 ms	8 ms	1	/home/sitedirs/example/site/layout/searchbar.cfm
4 ms	4 ms	1	/home/example/public_html/Application.cfm
1 ms	1 ms	1	/home/example/public_html/OnRequestEnd.cfm
1 ms	1 ms	1	/home/sitedirs/example/site/layout/searchbar_form.cfm
0 ms	0 ms	1	/home/example/public_html/account/includes/newsletters.cfm
0 ms	0 ms	1	/home/example/public_html/includes/google_analytics.cfm
0 ms	0 ms	1	/home/sitedirs/example/site/Application.cfm
0 ms	0 ms	1	/home/sitedirs/example/site/includes/create_account.cfm
0 ms	0 ms	1	/home/sitedirs/example/site/includes/navmenu.cfm
0 ms	0 ms	1	/home/sitedirs/example/site/layout/footer.cfm
2 ms			*STARTUP, PARSING, COMPILING, LOADING, & SHUTDOWN*
162 ms			*TOTAL EXECUTION TIME*

red = over 250 ms average execution time

SQL Queries **table1** (Datasource=users, Time=0ms, Records=1, Cached Query) in /home/example/public_html/Application.cfm @ 15:42:56.056

```
select * from users.table1 where id='41'
```

check_rt (Datasource=users, Time=2ms, Records=1) in /home/sitedirs/example/site/sections/section_body.cfm @ 15:42:56.056

```
select rivertown from table1
where id='41'
```

ad_juggler (Datasource=users, Time=11ms, Records=1) in /home/example2/public_html/includes/scripts/ad.cfm @ 15:42:56.056

```
select *
from users.table1 where id='41'
```

searchbar_user (Datasource=users, Time=1ms, Records=1) in /home/sitedirs/example/site/layout/searchba[illegible].cfm @ 15:42:56.056

```
select first,last,email from users where users.id='178233'
```

- SQL Statements Example

BENEFITS

- End to End Timing
- Component Timing
  - (i.e. a series of SQL)
- Observe as desired
- Intelligent Activation
  - e.g. Page Load exceeds x ms

EffectiveMySQL.com - Its all about **Performance** and **Scalability**

Testing

Testing is not about what works; testing is about breaking your software

TESTING

- Enforce an API
- Simplify testing
- Separate responsibility

RESILIENCE

- What is your backup strategy?
- What is your recovery strategy?
- How long does it take?
- Have you actually tested it end to end?

Take the survey

http://rb42.com/mysql-backup-quiz

RESILIENCE

1. Do you have MySQL backups in place?
2. Do you backup ALL your MySQL data?
3. Do you have consistent MySQL backups?
4. Do you have backups that include both static snapshot and point in time transactions?
5. Do you review your backup logs EVERY SINGLE day or have tested backup monitoring in place?
6. Do you perform a test recovery of your static backup?
7. Do you perform a test recovery to point in time?
8. Do you time your backup and recovery process and review over time?
9. Do you have off-site copies of your backups?
10. Do you backup your primary binary logs?

http://rb42.com/mysql-backup-quiz

Monitoring Instrumentation & Testing

Number 1 problem!

Don't change a setting without evidence to prove/disprove the benefit

BAD CONFIGURATION

- [sort|join|read|read_rnd] _buffer_size
  - Defaults are 128K - 256K
  - Settings of 1M, 2M,16M, 128M
- Pre allocated per thread buffer
  - Larger buffer is slower to create > 256K
  - Wastes valuable process memory

THE BOTTLENECK?

- Is the database the problem?
- Front end performance
  - The website will always be too slow
  - Identify the true components
  - End to End time
    - Database may be only a small portion

Conclusion

CONCLUSION

- Identify poor development practices
- Recommended best practices
- MySQL development tips and tricks
- Standard & expected RDBMS practices