

The Ideal Performance Architecture

For MySQL System Architects

Ronald Bradford

Principal - 42SQL

Percona Performance Conference

April 2009

Overview

Overview

- ❖ Technology
- ❖ Disk
- ❖ Memory
- ❖ Indexes
- ❖ SQL
- ❖ Data

Technology

Know Your Technology Tools

- ❖ Generics are inefficient
- ❖ Product expertise in a different RDBMS is not enough
- ❖ You have chosen MySQL
 - ❖ Maximize it's strengths
 - ❖ Minimize it's weaknesses

Disk

Know Your Disk Footprint

Disk = Memory = Performance

- ❖ Every single byte does count

Poor Design is

- ❖ INT(1)
- ❖ BIGINT AUTO_INCREMENT
- ❖ no UNSIGNED used
- ❖ DECIMAL(31,0)
- ❖ VARCHAR(255)
- ❖ All Nullable Columns

30% saving

```
CREATE TABLE `searchtools_raw_term_domain_referrals_temp` (  
  `id` bigint(20) unsigned NOT NULL auto_increment,  
  `term_id` bigint(20) unsigned NOT NULL,  
  `domain_id` bigint(20) unsigned NOT NULL,  
  `referrals` int(10) unsigned NOT NULL,  
  `time` int(10) unsigned NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `term_domain` (`term_id`,`domain_id`),  
  KEY `term_values` (`term_id`,`referrals`,`time`),  
  KEY `domain_values` (`domain_id`,`referrals`,`time`)  
) ENGINE=MyISAM AUTO_INCREMENT=38890778 DEFAULT CHARSET=latin1;
```

- ❖ Converted BIGINT to INT
- ❖ Before: 1.2GB Data + 3.2GB Indexes
- ❖ After: 0.7GB Data + 2.2GB Indexes
- ❖ 33% Reduction

60% saving

```
CREATE TABLE `dev_stats_0` (  
  `object` mediumint(3) unsigned NOT NULL default '0',  
  `date` date NOT NULL default '0000-00-00',  
  `time` time NOT NULL default '00:00:00',  
  `data` char(128) default NULL,  
  `ind` tinyint(5) unsigned NOT NULL default '0',  
  `normalized` tinyint(5) unsigned NOT NULL default '0',  
  PRIMARY KEY (`date`,`time`,`object`,`ind`),  
  KEY `normalized` (`normalized`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

- ❖ Converted CHAR(128) to CHAR(20)
- ❖ Before: 45GB After 18GB 60% Saving
- ❖ Schema had multiple occurrences
 - ❖ 600+ GB significantly reduced

84% saving

```
CREATE TABLE `client_table` (  
  `mid` decimal(31,0) NOT NULL default '0',  
  `sid` decimal(31,0) NOT NULL default '0',  
  `oid` decimal(31,0) NOT NULL default '0',  
  `nid` decimal(31,0) NOT NULL default '0',  
  `bid_status` varchar(30) default NULL,  
  UNIQUE KEY `pk_client_table` (`sid`,`nid`,`mid`,`oid`)  
) TYPE=InnoDB;
```

- ❖ Converted DECIMAL(31,0) to INT UNSIGNED
- ❖ Before: 5.2GB After 850MB - 84%
- ❖ Schema had 77 occurrences
- ❖ Converted 1 of 12 shards
 - ❖ Total: 7.8GB to 1.7GB (78% saving)

Data Types

- ❖ INT(1) - 1 does not mean 1 digit
 - ❖ (1) represents client output display format only
 - ❖ INT is 4 Bytes, TINYINT is 1 Byte
 - ❖ TINYINT UNSIGNED can store from 0 – 255

Data Types

- ❖ BIGINT is not needed for AUTO_INCREMENT
- ❖ INT UNSIGNED stores 4.3 billion values
 - ❖ You should be partitioning when at billions of rows
- ❖ BIGINT is applicable for some columns
 - ❖ e.g. summation of values

Memory

Don't waste memory

- ❖ `SELECT *`
- ❖ `utf8` for everything
- ❖ `TEXT/BLOB`
- ❖ large per session buffers

Character Sets

- ❖ Default in MySQL 5 is utf8 (3 bytes)
- ❖ Only define columns that need utf8
 - ❖ e.g. Not Codes, MD5 Value, web address
- ❖ MySQL internal buffers are fixed width
 - ❖ e.g. VARCHAR(255) utf8 is 765 bytes to store just 1 byte

Data Types

❖ VARCHAR(255) Is not a design practice

```
CREATE TABLE `XXX` (  
  `orderHandle` varchar(255) NOT NULL default '',  
  `personName` varchar(255) default NULL,  
  `addressLines` varchar(255) default NULL,  
  `city` varchar(255) default NULL,  
  `state` varchar(255) default NULL,  
  `postalCode` varchar(255) default NULL,  
  `countryCode` varchar(255) default NULL,  
  `phone` varchar(255) default NULL,  
  `email` varchar(255) default NULL,  
  `shipMethod` varchar(255) default NULL,  
  `shipTo` varchar(255) default NULL,  
  `receiveByDate` date default NULL,  
  `currency` varchar(3) default NULL,  
  `price` varchar(255) default NULL,  
  `flags` int(11) default '0',  
  `lastUpdateTime` timestamp NOT NULL default CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP,  
  `creationTime` timestamp NOT NULL default '0000-00-00 00:00:00',  
  PRIMARY KEY (`orderHandle`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8
```

Be Wary of TEXT/BLOB

- ❖ Using SELECT *
- ❖ MySQL Internal Temporary table will force Temp Disk Table
- ❖ Internal storage (e.g. Innodb)
 - ❖ Stores first 768 bytes, then a separate 16k data page per row per TEXT/BLOB field

Indexes

The Impact Of Indexes

❖ Good

- ❖ Dramatic performance improvements
- ❖ Improves memory usage
- ❖ Data Integrity

❖ Bad

- ❖ Slows performance for writes
- ❖ Wastes disk space for unused, duplicate or ineffective indexes
- ❖ In-effective usage of memory

Minimizing internal MySQL processing

- ❖ Correctly designed tables, indexes and SQL can eliminate
 - ❖ Using temporary table
 - ❖ Using filesort

Covering Indexes

- ❖ Understand how to use covering indexes
- ❖ Write efficient SQL

SQL

Know Every SQL Statement

- ❖ Developers don't write proper SQL statements
- ❖ SQL statements will directly affect your performance
- ❖ For Example
 - ❖ Repeating SQL statements for no benefit
 - ❖ 1000 very quick small unnecessary queries is worse than 1 slow query

Monitor Every SQL Statement

- ❖ Review Query Execution Plan (QEP)
 - ❖ EXPLAIN
- ❖ Time queries
- ❖ Row Count / Affected rows
- ❖ Result Set Size

Review over time, things change

Data

Sharding

- ❖ Plan and implement from Day 1
- ❖ Find a good partition key
 - ❖ Business Related
 - ❖ Programmatically e.g. MOD (id,1000)
- ❖ Design for rebalancing support
- ❖ Virtual Host / Physical Host

Data Availability

- ❖ Messaging system for data availability
 - ❖ Read/Write availability
 - ❖ Read Availability
- ❖ Support at table level
- ❖ Support at table partitions
 - ❖ e.g. by PK id range

Write Once Data

- ❖ Separate your online and archival data
- ❖ Reduces backup/recovery time
- ❖ Enables data synchronization
 - ❖ including MySQL to non-MySQL

Professional Help is Available

- ❖ Two decades IT expertise
- ❖ 10 years in MySQL
- ❖ System/Data Architecture
- ❖ Database Performance and Tuning
- ❖ High Availability and Scalability
- ❖ Education and Training

<http://ronaldbradford.com>