

MySQL 性能优化最佳实践

About me

简朝阳(sky000)

Oracle ACE(Expertise: MySQL)

技术保障部 @麦包包

Blog: <http://isky000.com>

Twitter: [@sky000](#)

Weibo: [@简朝阳](#)



DTCC2012

MySQL 性能优化最佳实践

优化过程



DTCC2012

MySQL 性能优化最佳实践

找出瓶颈



DTCC2012

MySQL 性能优化最佳实践

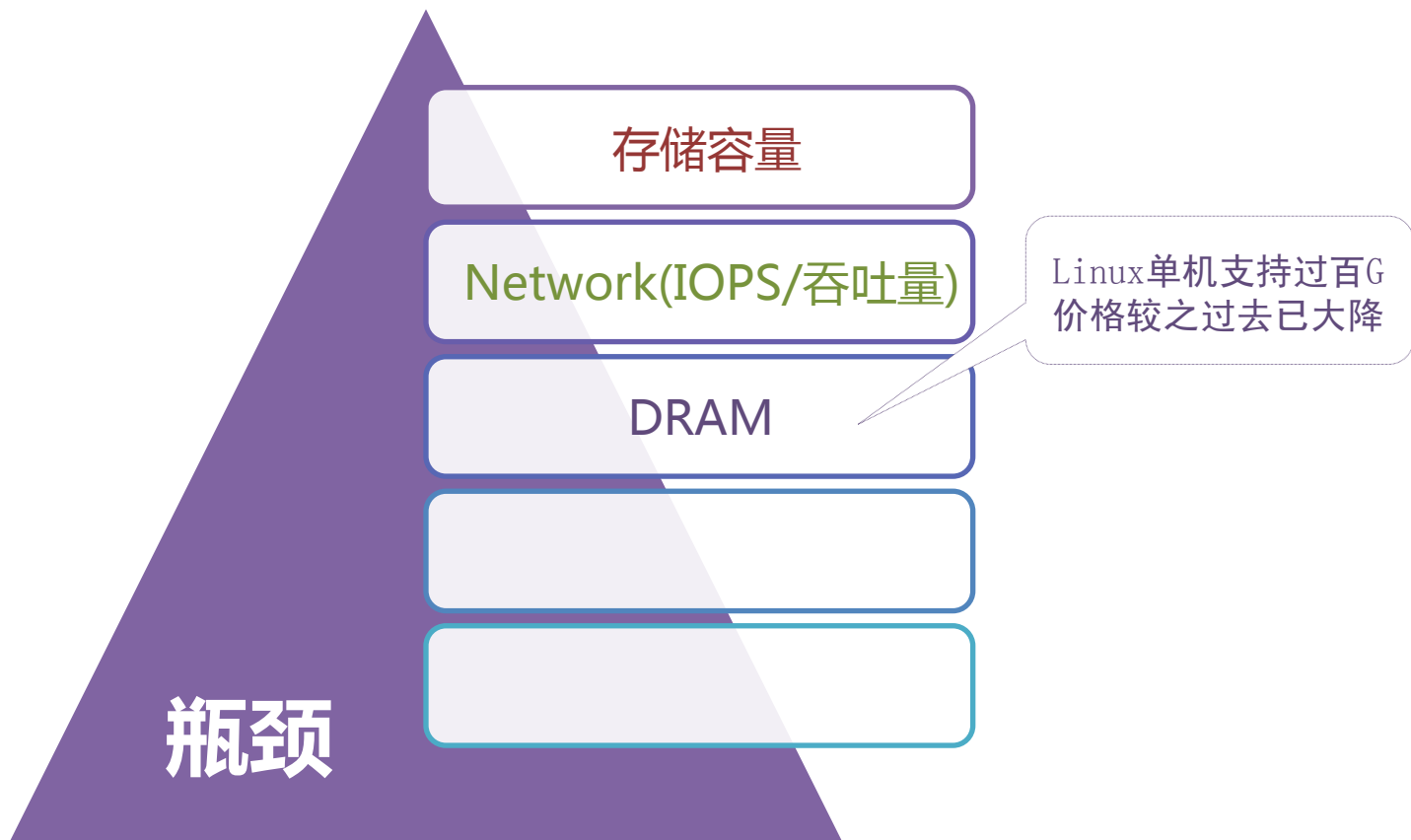
找出瓶颈



DTCC2012

MySQL 性能优化最佳实践

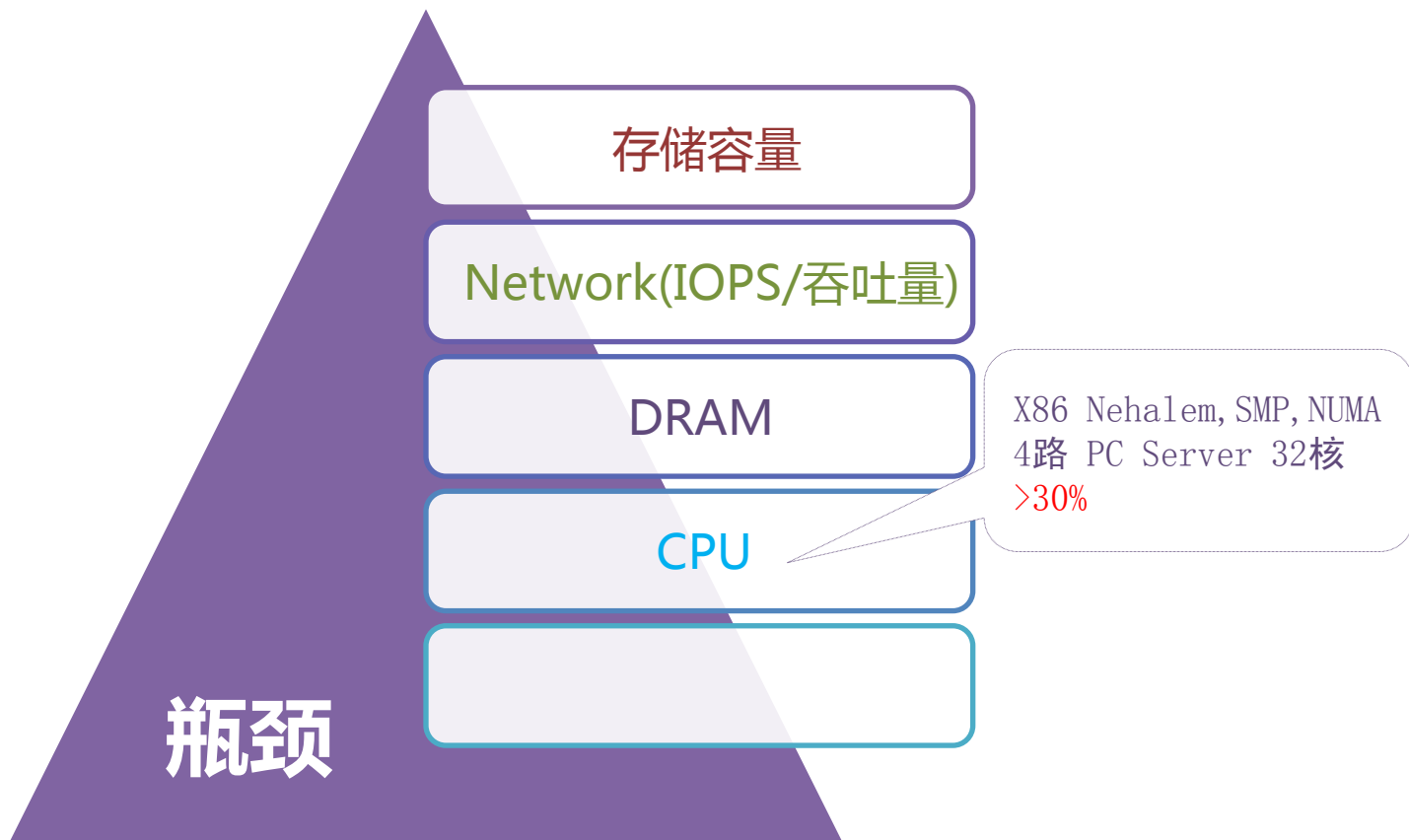
找出瓶颈



DTCC2012

MySQL 性能优化最佳实践

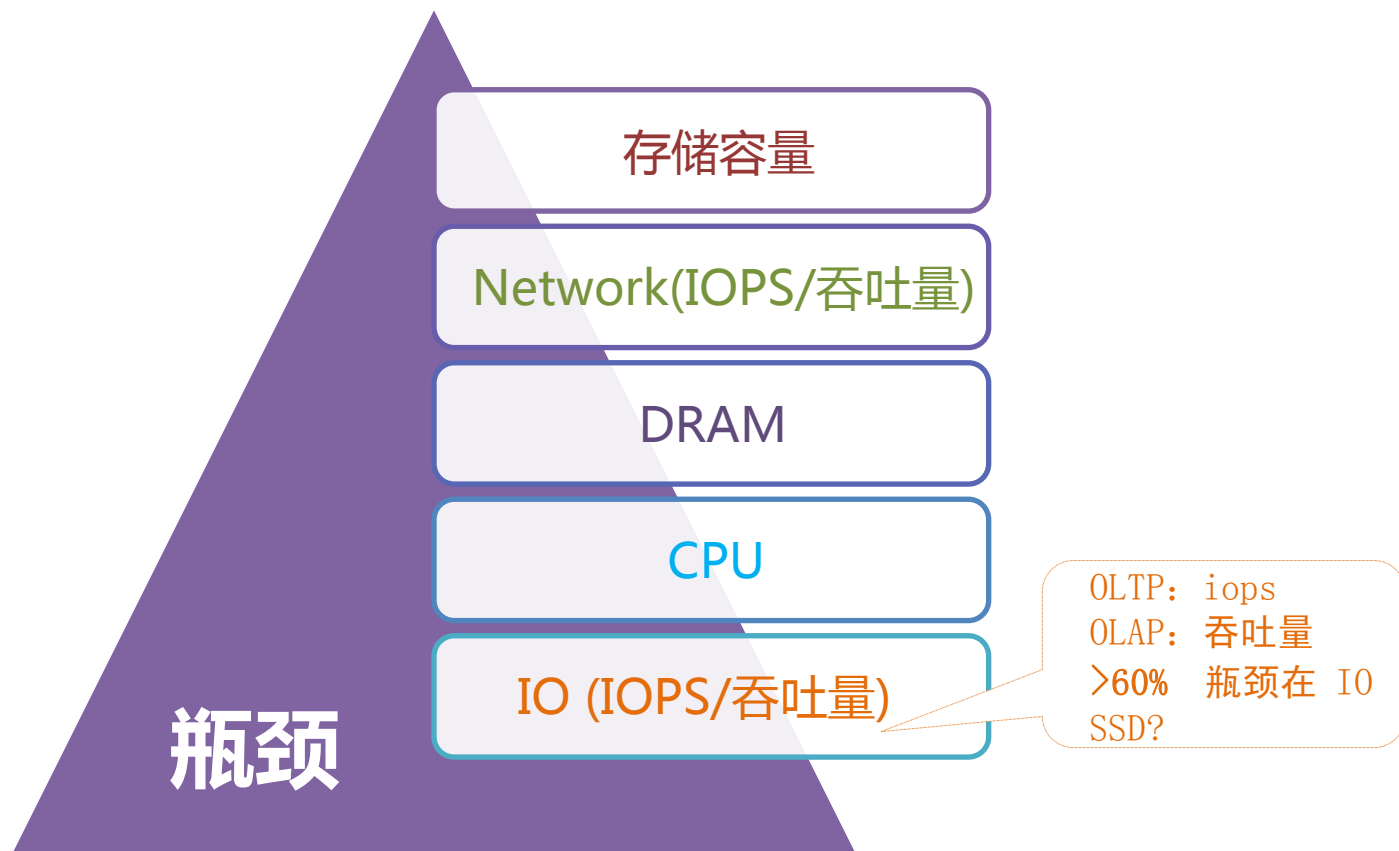
找出瓶颈



DTCC2012

MySQL 性能优化最佳实践

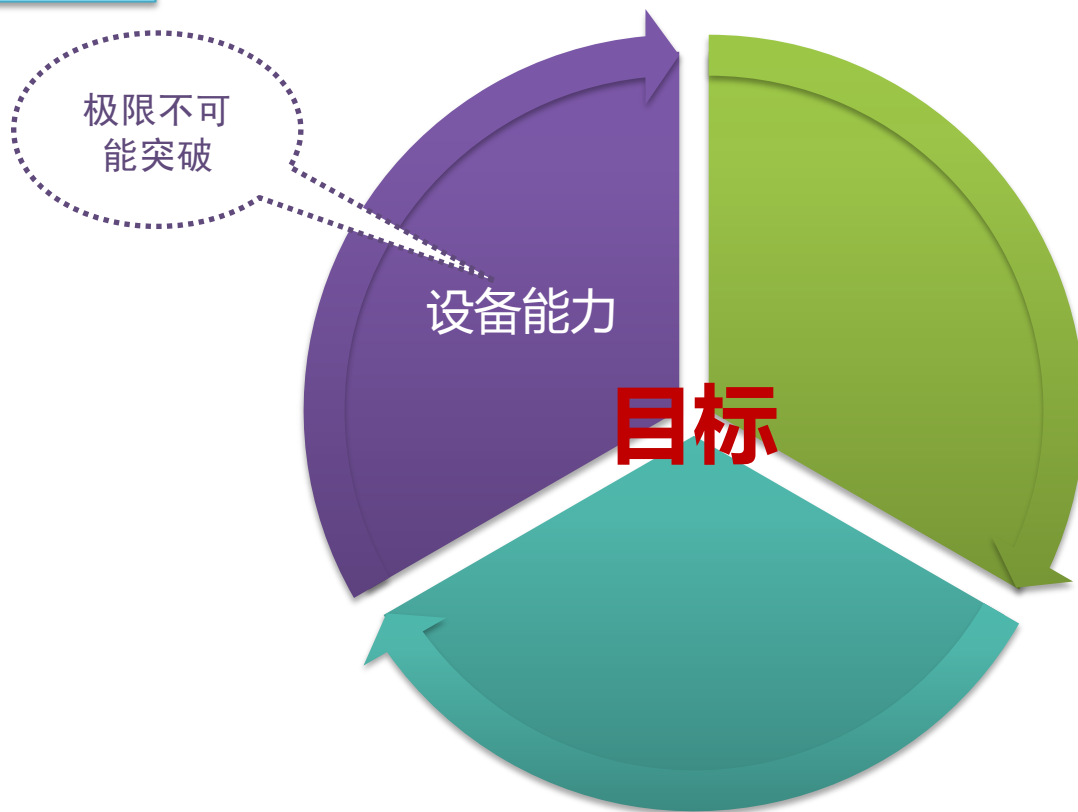
找出瓶颈



DTCC2012

MySQL 性能优化最佳实践

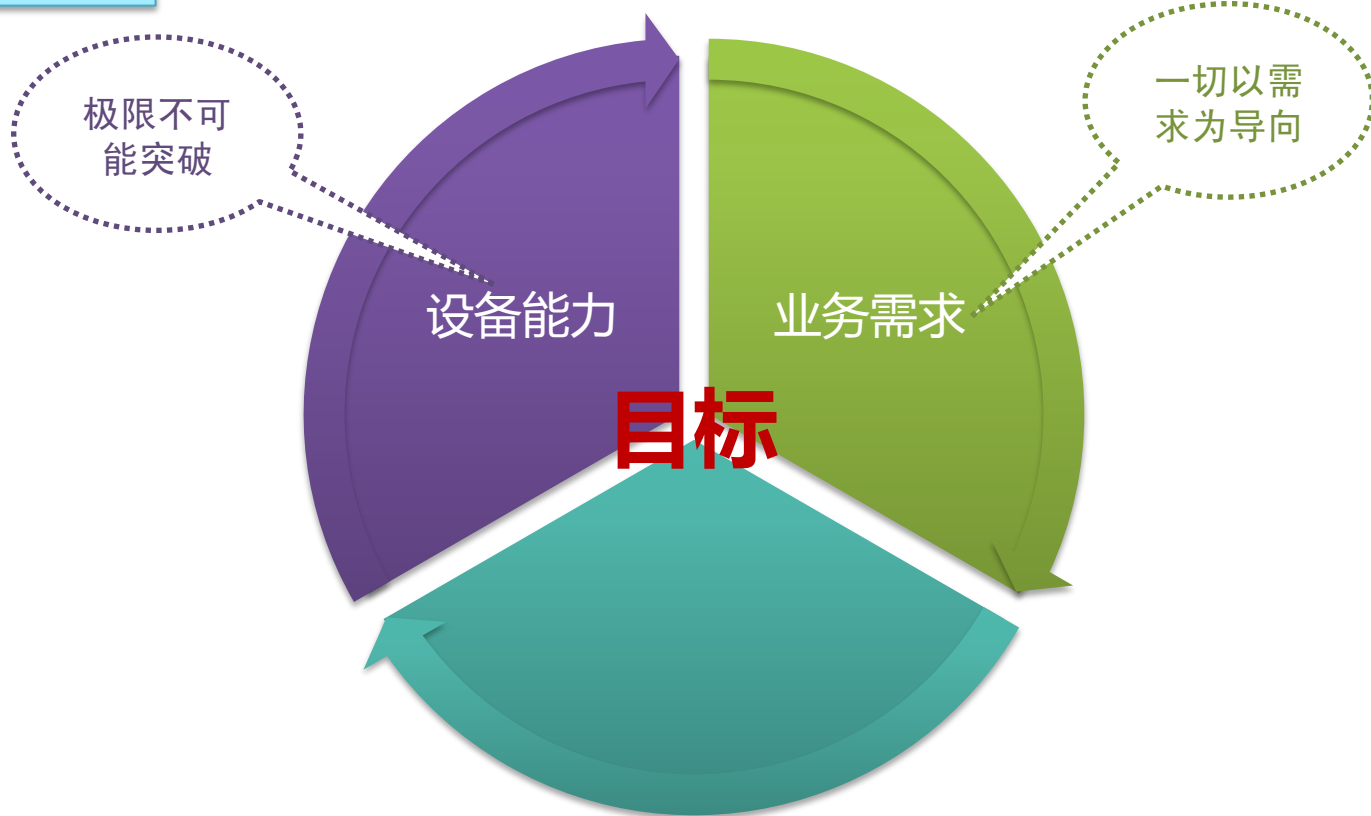
设定目标



DTCC2012

MySQL 性能优化最佳实践

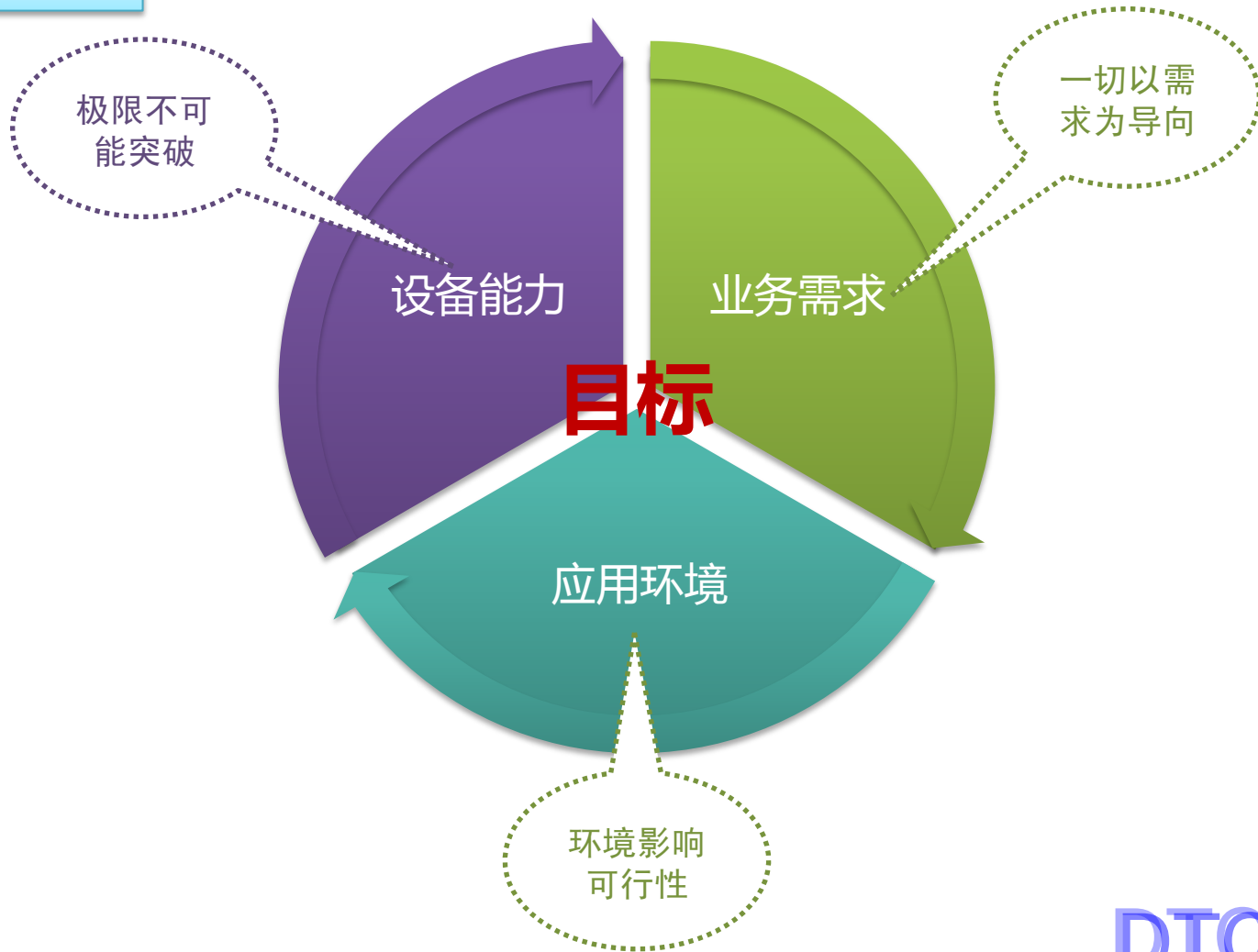
设定目标



DTCC2012

MySQL 性能优化最佳实践

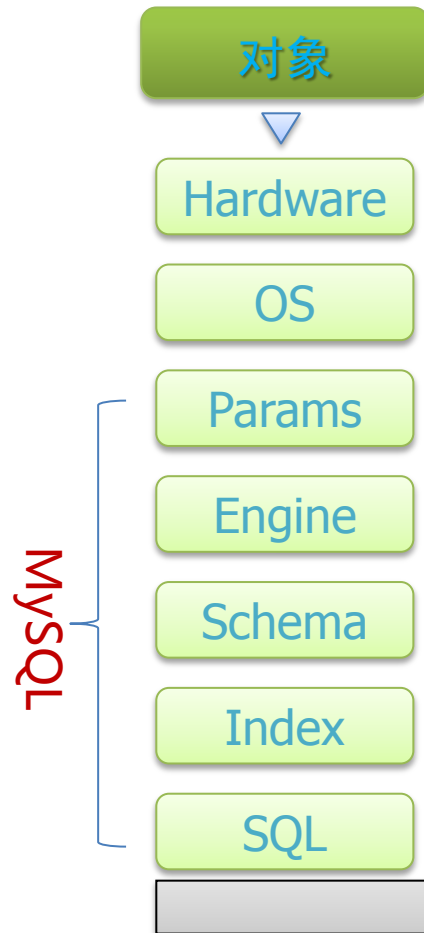
设定目标



DTCC2012

MySQL 性能优化最佳实践

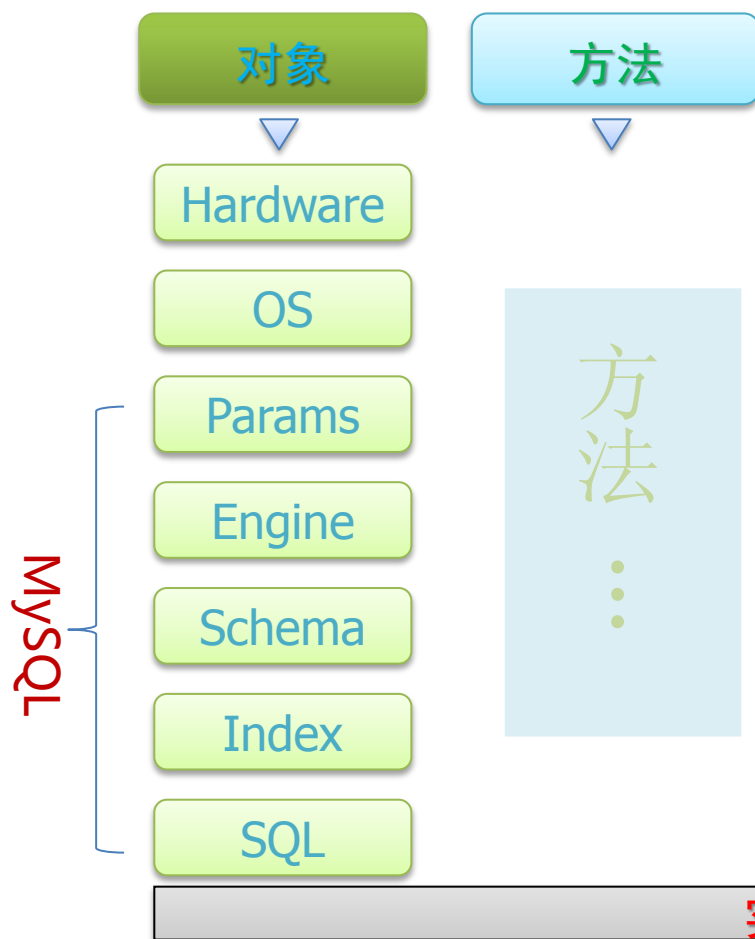
实施优化



DTCC2012

MySQL 性能优化最佳实践

实施优化



DTCC2012

MySQL 性能优化最佳实践

实施优化



MySQL 性能优化最佳实践

实施优化



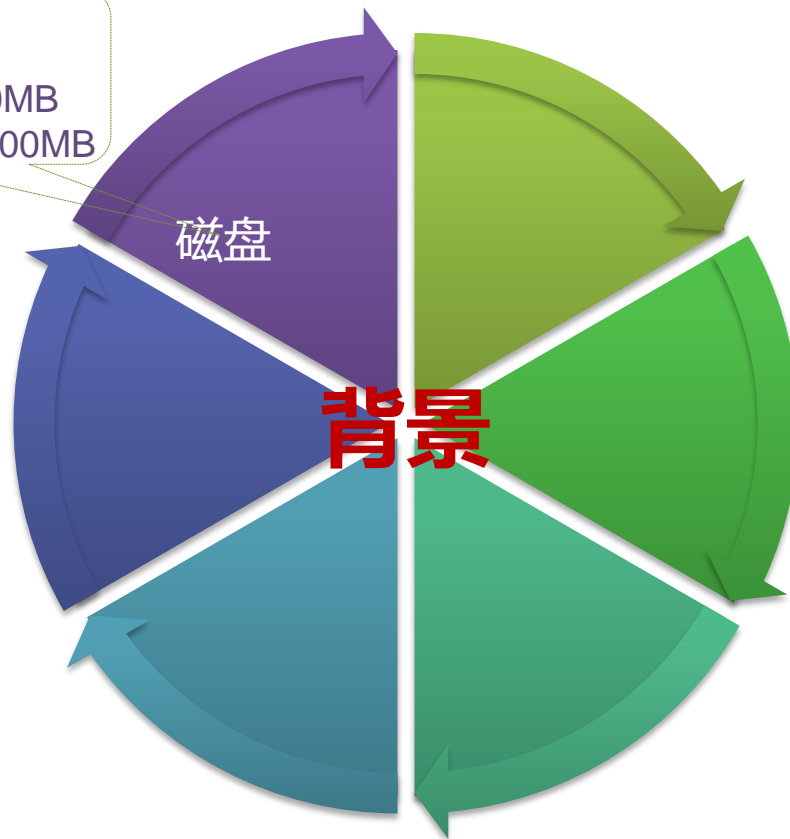
MySQL 性能优化最佳实践

实施优化

转速，容量，接口

HDD: ~150 iops, < 200MB

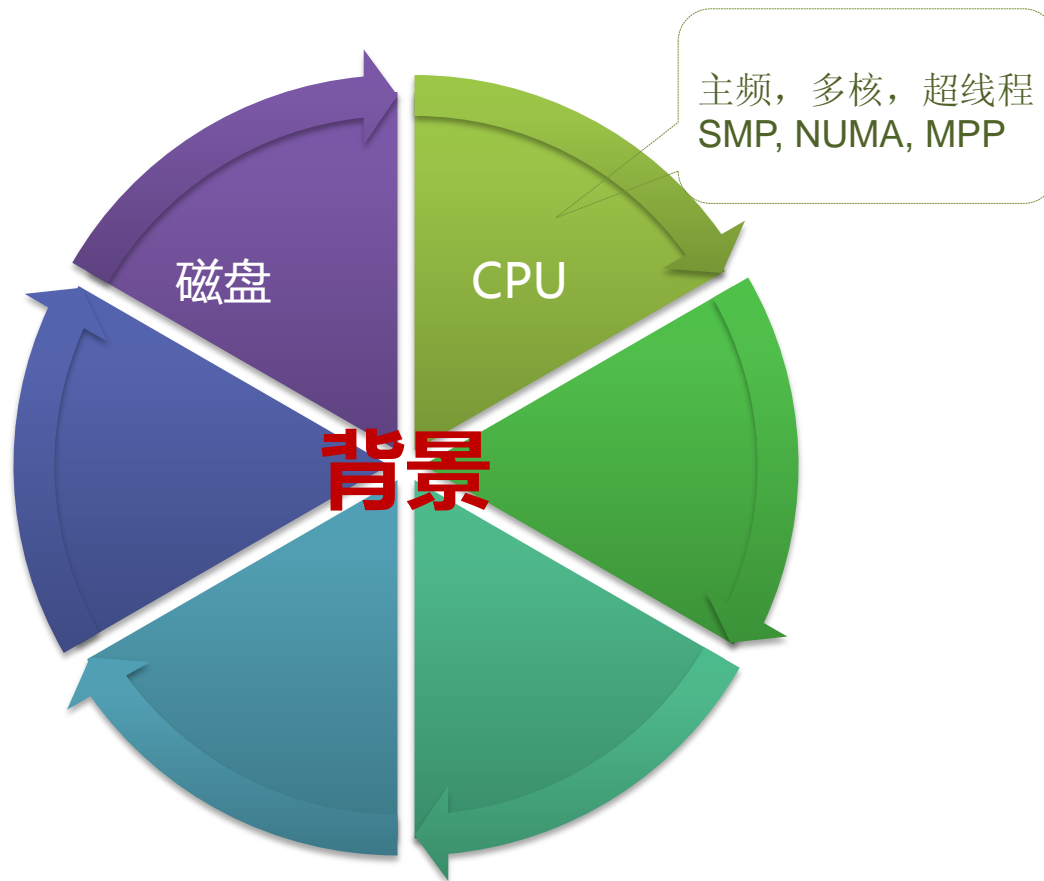
SSD: 10x ~ 1000x, < 400MB



DTCC2012

MySQL 性能优化最佳实践

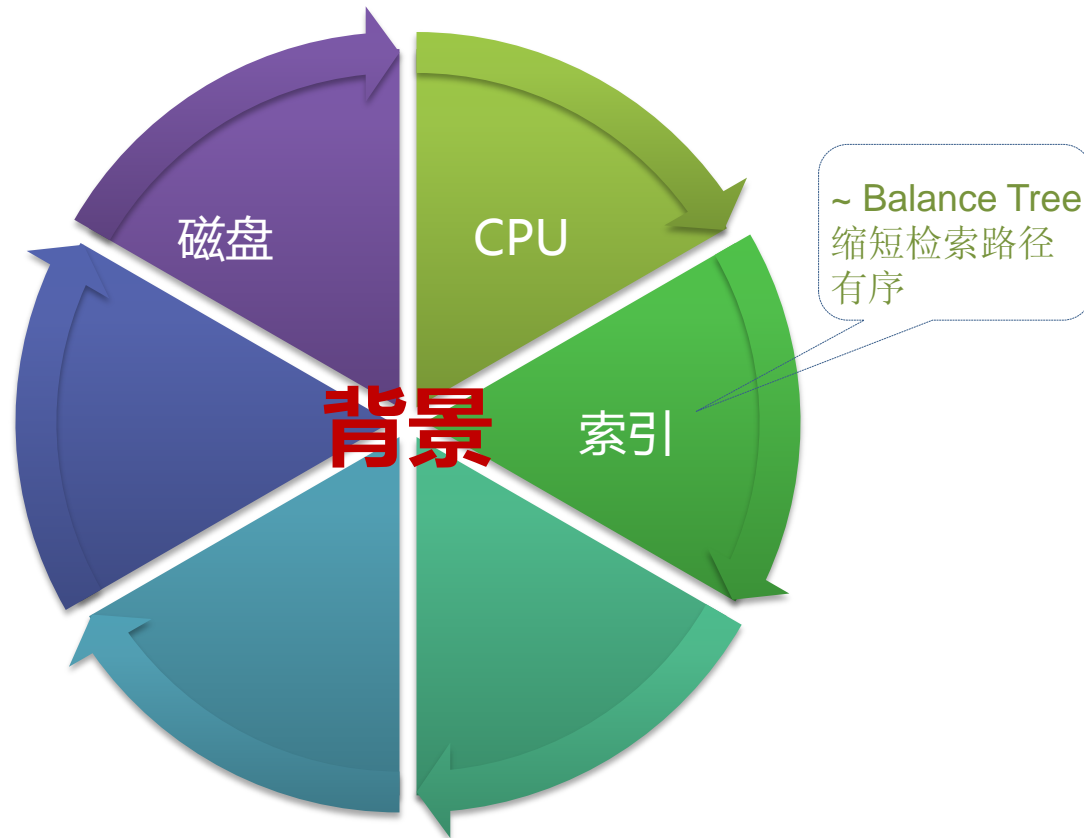
实施优化



DTCC2012

MySQL 性能优化最佳实践

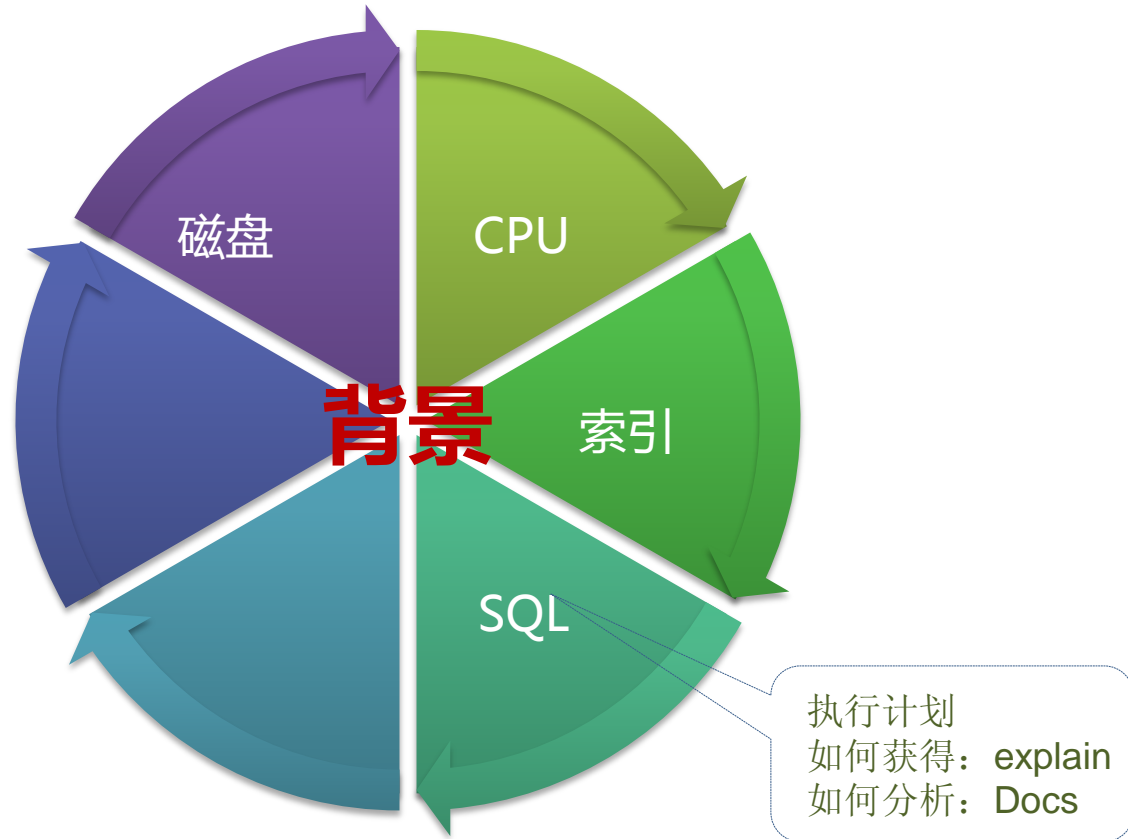
实施优化



DTCC2012

MySQL 性能优化最佳实践

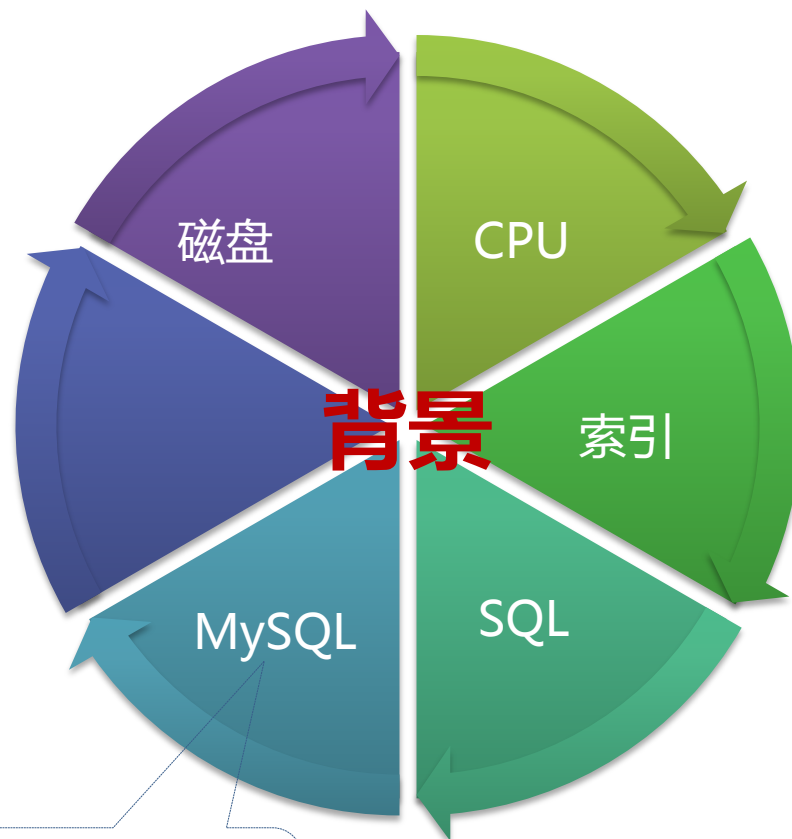
实施优化



DTCC2012

MySQL 性能优化最佳实践

实施优化

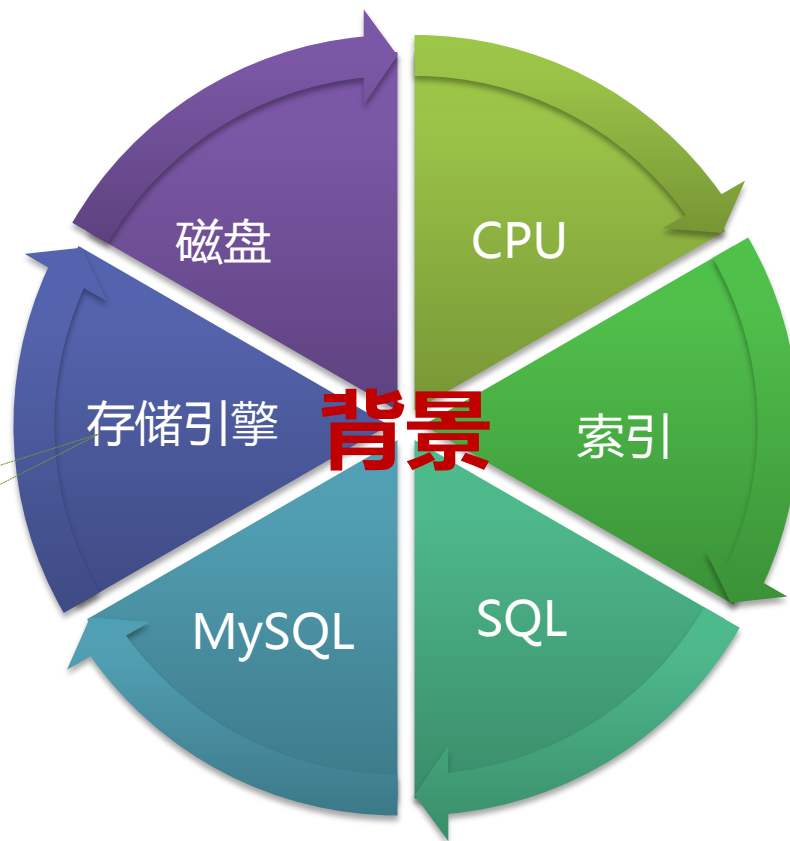


简单，轻型，开放
多线程，插件式
SQL+Storage Engine ...

DTCC2012

MySQL 性能优化最佳实践

实施优化



插件式，可自由更换
开放型，可自行开发
多样性，特性不一
并存性，可并存使用

DTCC2012

MySQL 性能优化最佳实践

方法



DTCC2012

MySQL 性能优化最佳实践

误区



DTCC2012

MySQL 性能优化最佳实践

最佳实践



DTCC2012

MySQL 性能优化最佳实践

方法



DTCC2012

MySQL 性能优化最佳实践

误区

Hardware

Disk

- OLTP: 小容量”高”转速
- OLAP: 大容量”低”转速(钱多可以高转速)
- 磁盘数量尽可能多
- 有钱可以上 SSD(IO 瓶颈场景下)

Raid Card

- 读写都是用 Cache 提升效率?
- Raid10一定比Raid5快?
- 带电池的 Raid 卡数据一定安全?

CPU

⋮

DTCC2012

MySQL 性能优化最佳实践

最佳实践



DTCC2012

MySQL 性能优化最佳实践

方法



DTCC2012

MySQL 性能优化最佳实践

误区

Hardware

Disk

- OLTP: 小容量”高”转速
- OLAP: 大容量”低”转速(钱多可以高转速)
- 磁盘数量尽可能多
- 有钱可以上 SSD(IO 瓶颈场景下)

Raid Card

- Cache 只供写使用, Direct 读取
- OLTP Raid10, Strip Size 参考DB
- OLAP Raid5
- 关注 Raid 卡充放电带来的 Cache 失效
- 预读只对连续读有效, OLTP 关闭预读

CPU

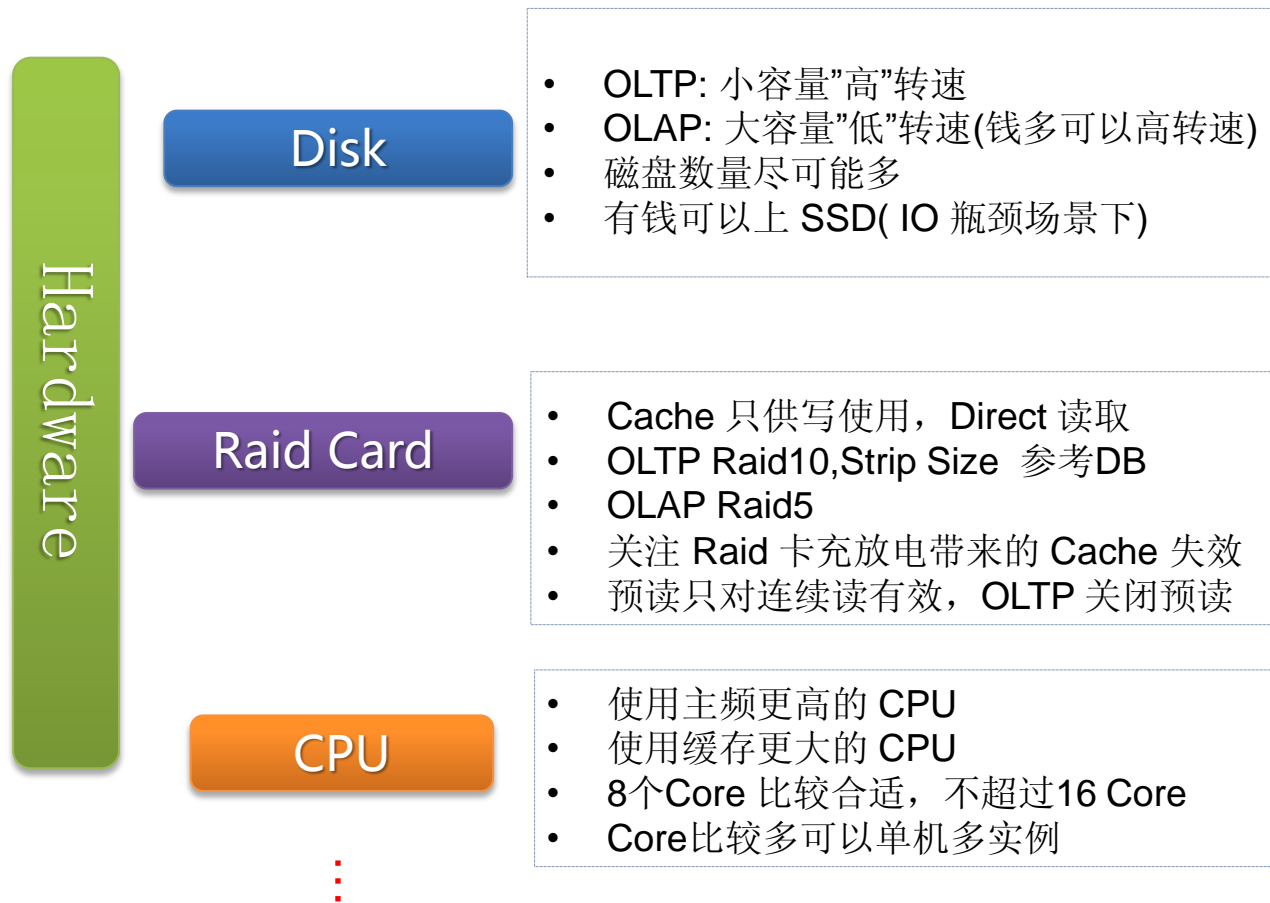
- CPU 越多越好?
- Core 越多越好?

⋮

DTCC2012

MySQL 性能优化最佳实践

最佳实践



DTCC2012

MySQL 性能优化最佳实践

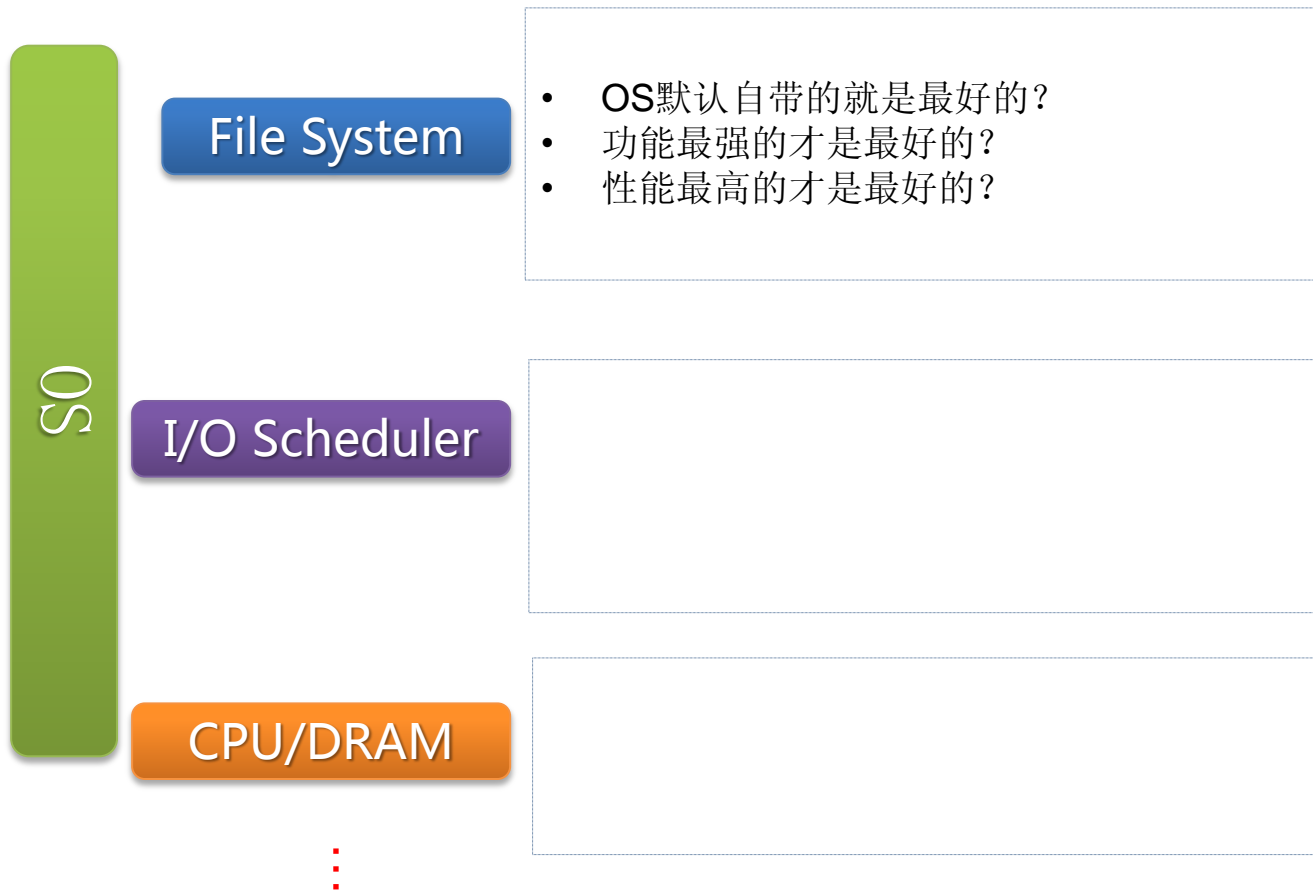
方法



DTCC2012

MySQL 性能优化最佳实践

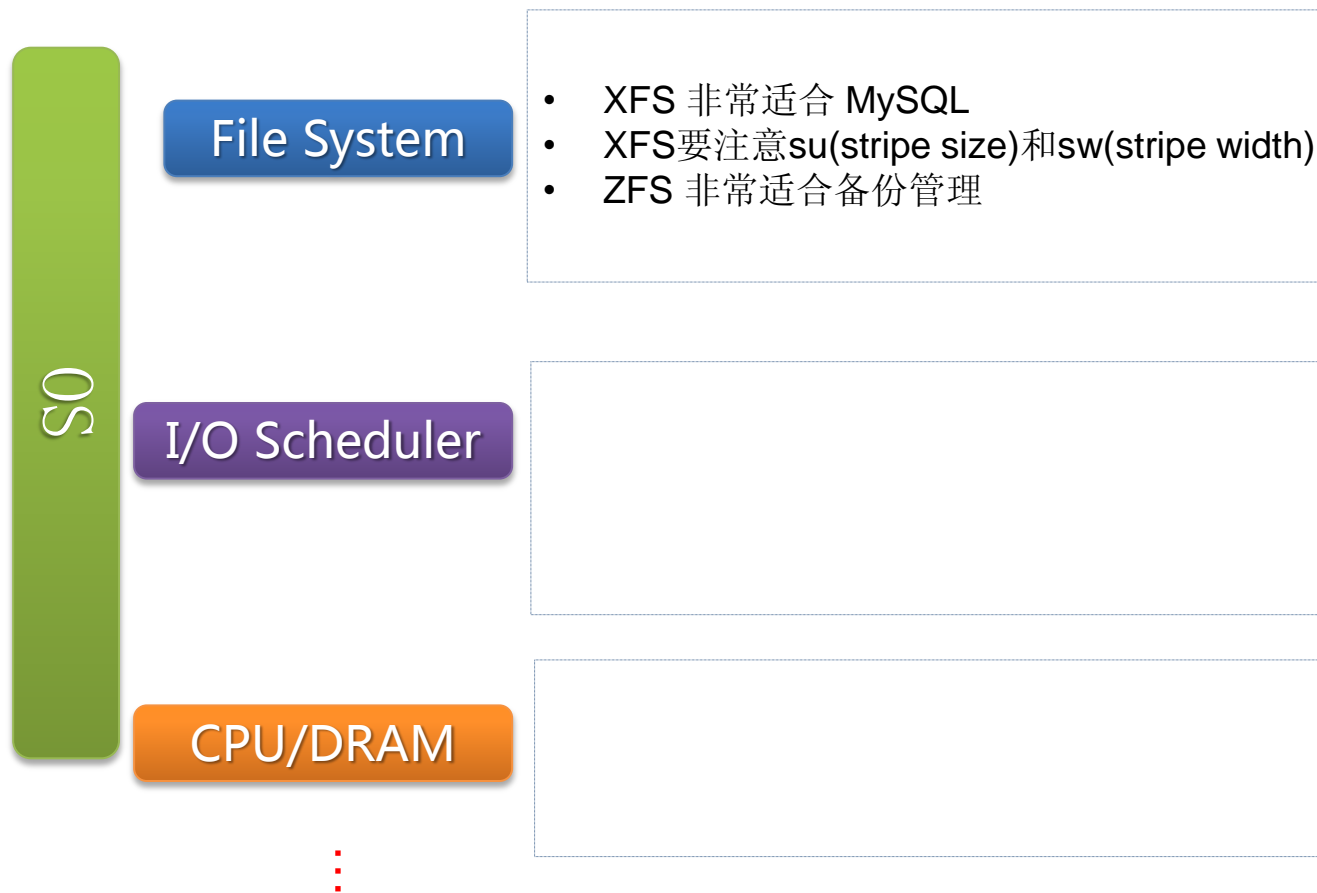
误区



DTCC2012

MySQL 性能优化最佳实践

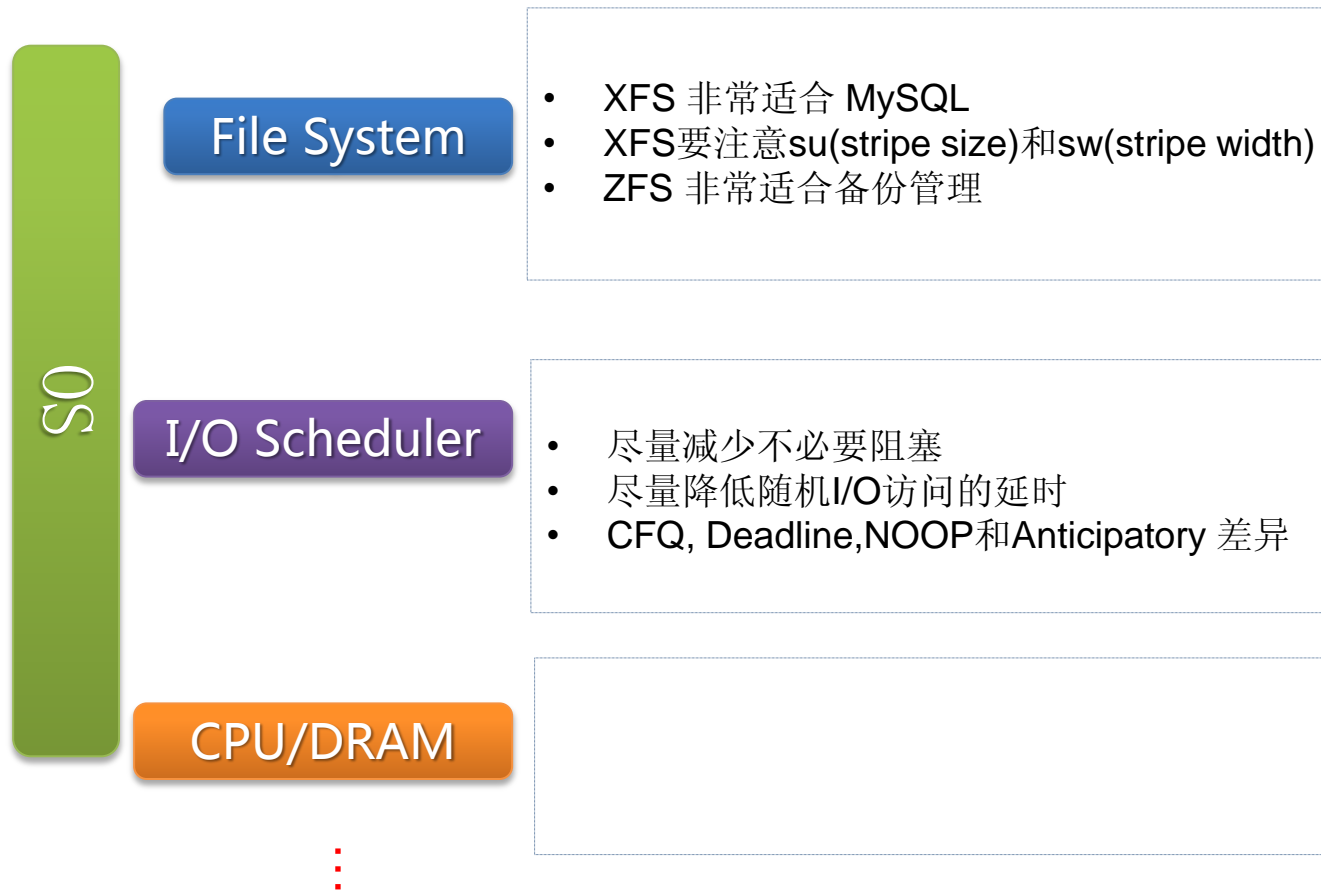
最佳实践



DTCC2012

MySQL 性能优化最佳实践

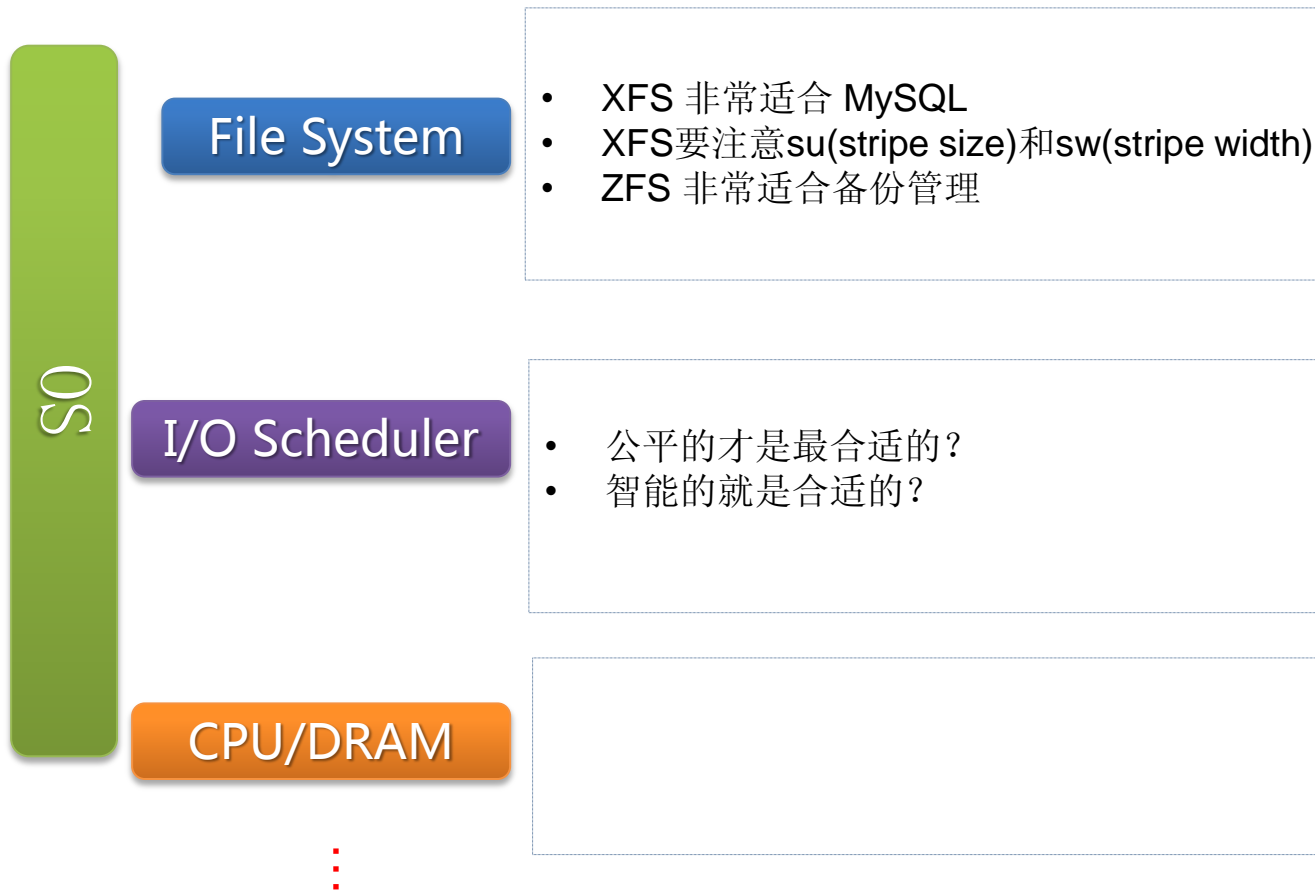
方法



DTCC2012

MySQL 性能优化最佳实践

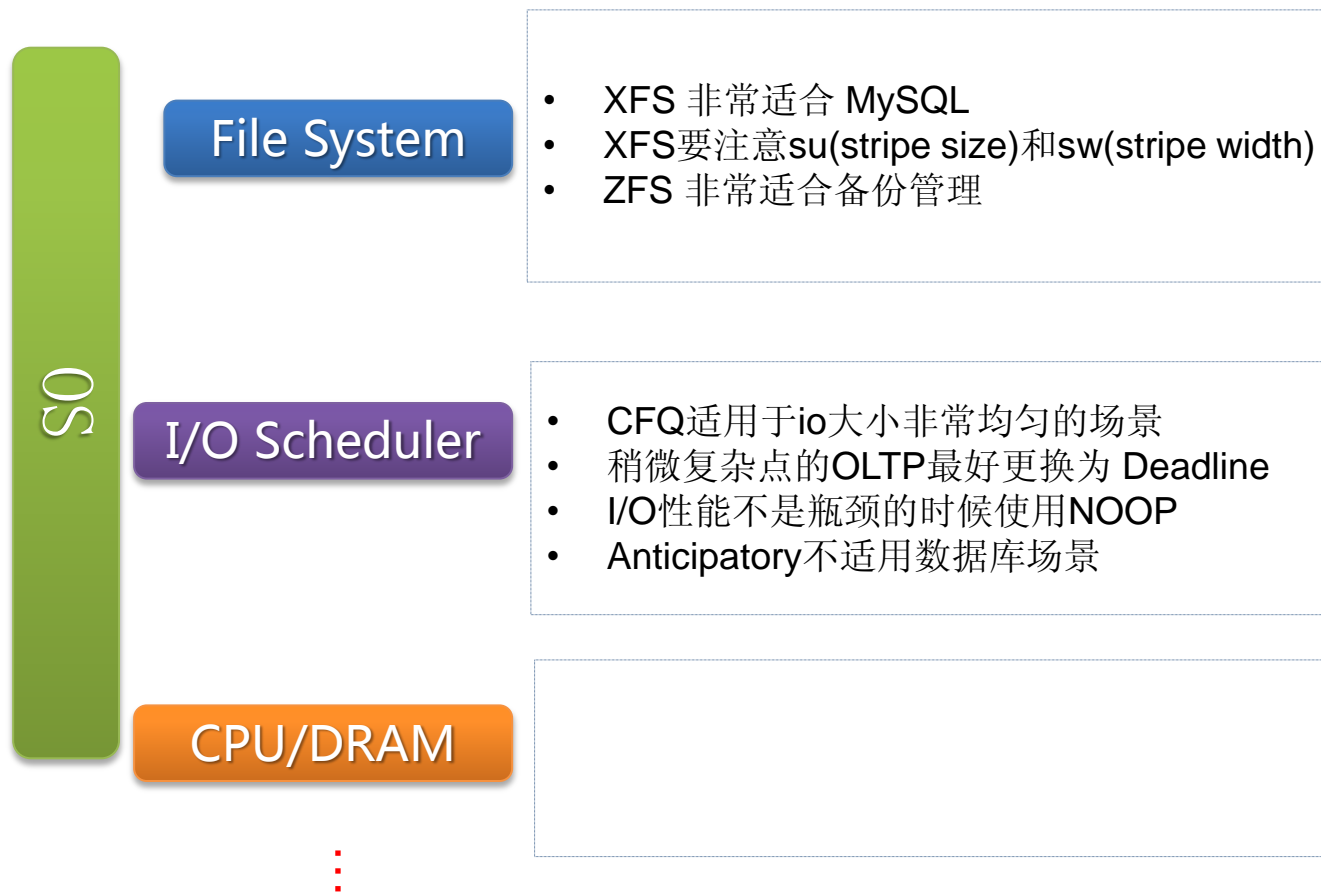
误区



DTCC2012

MySQL 性能优化最佳实践

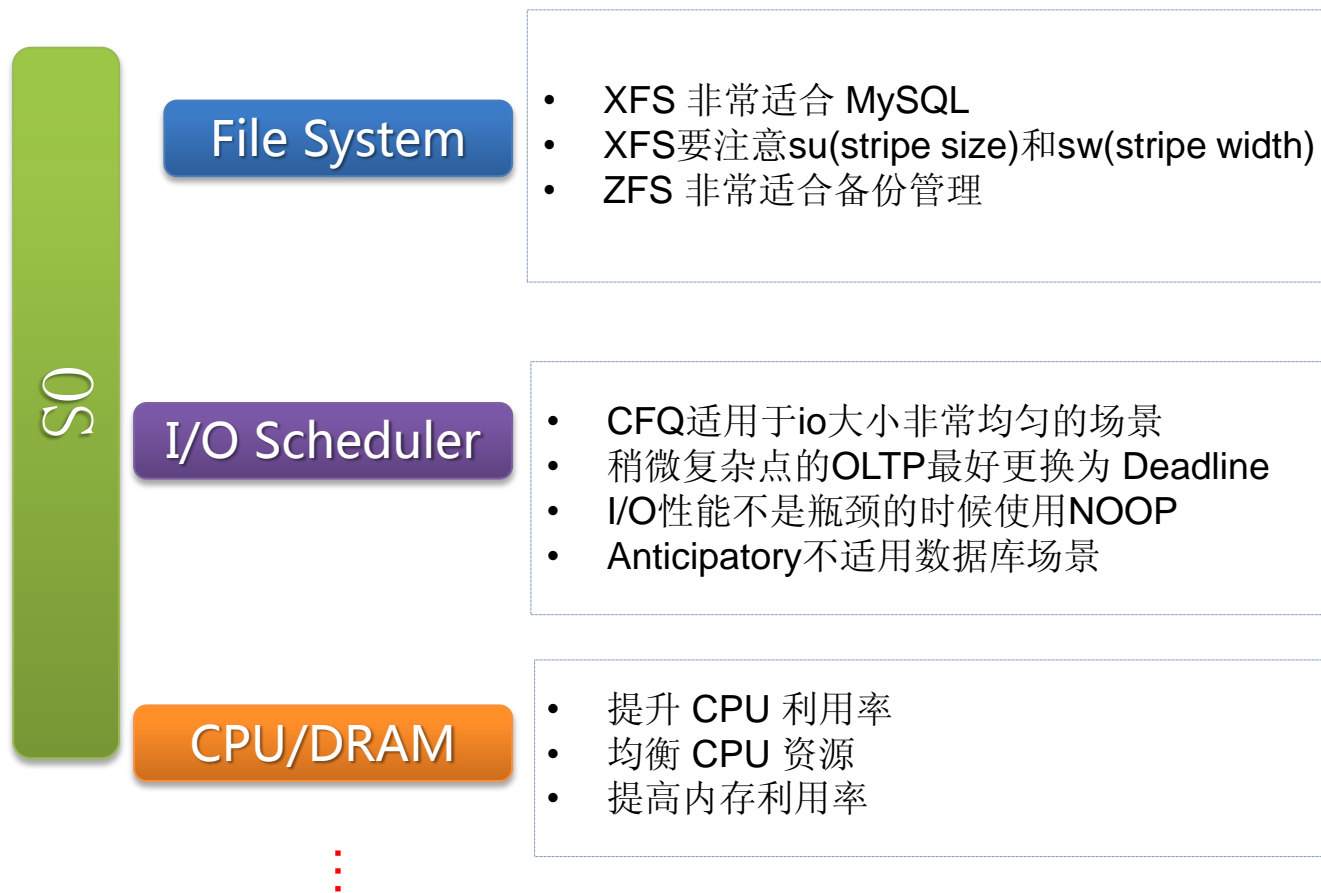
最佳实践



DTCC2012

MySQL 性能优化最佳实践

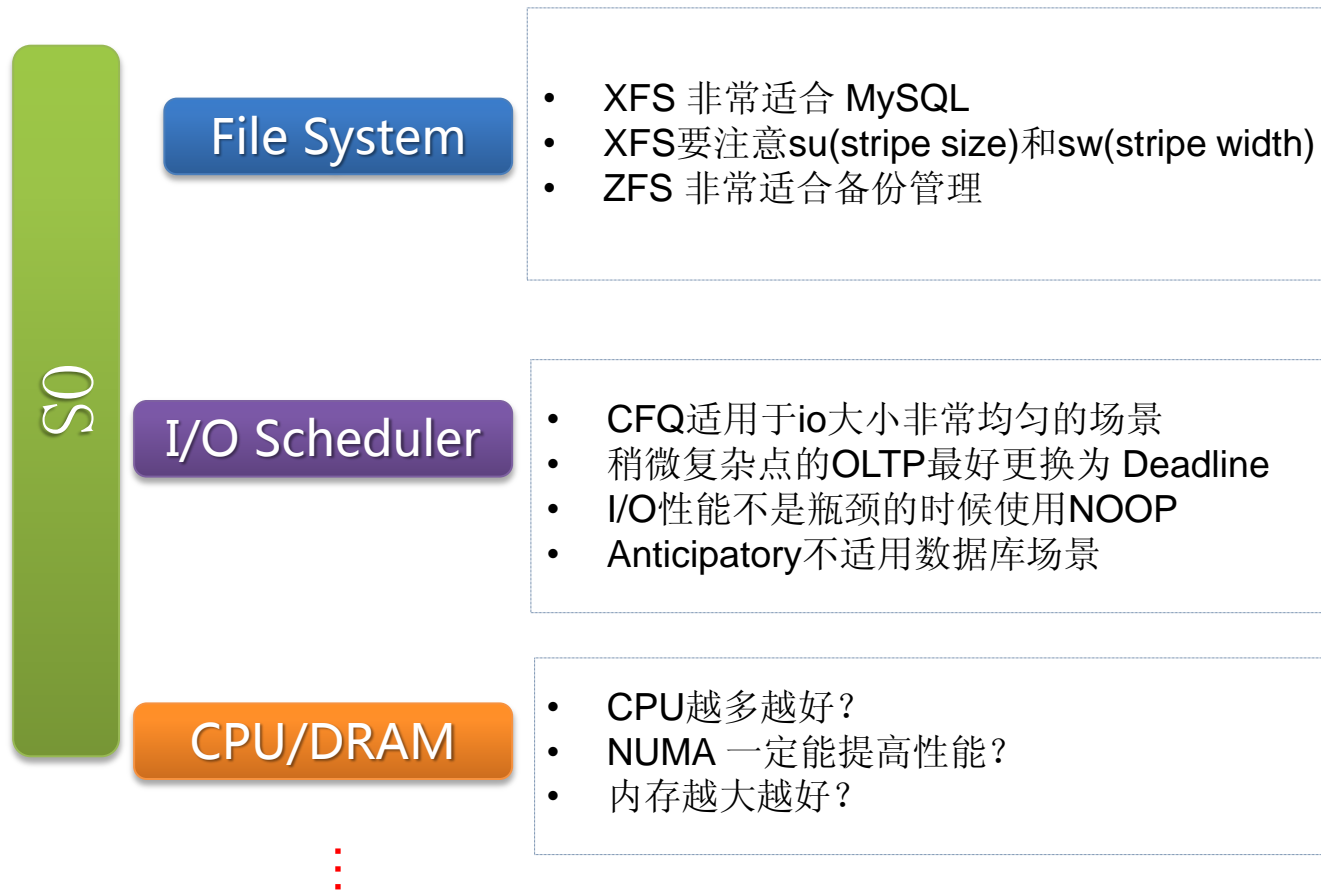
方法



DTCC2012

MySQL 性能优化最佳实践

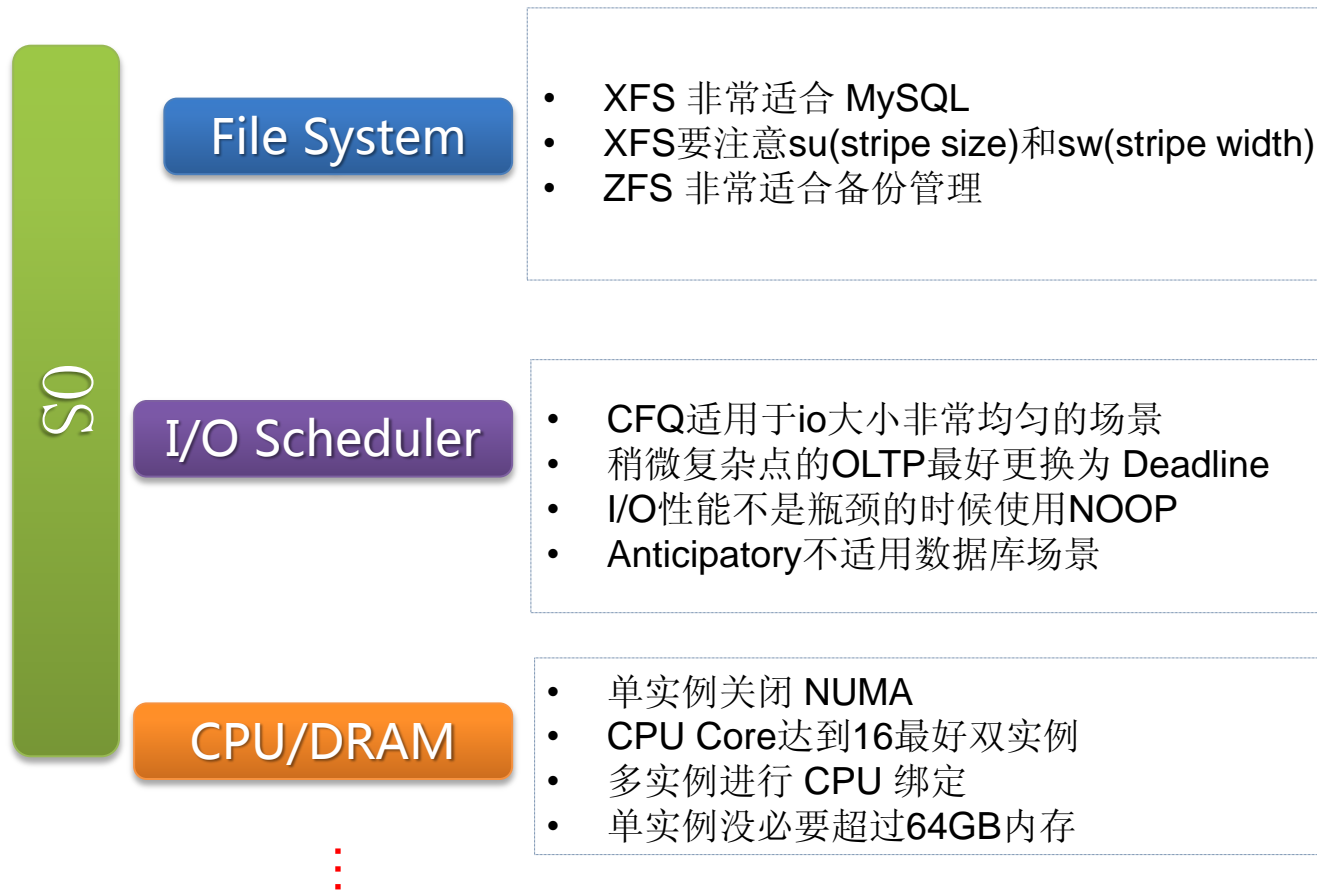
误区



DTCC2012

MySQL 性能优化最佳实践

最佳实践



DTCC2012

MySQL 性能优化最佳实践

方法

Cache/Buffer

Connction

IO

⋮

- **query_cache**: 缓存结果集, 极高效, 与SQL语句一一对应
- **binlog_cache_size**: 缓存binlog数据, 影响所有写入操作的性能
- **table_cache**: 缓存打开的表信息, MyISAM会占用较多, 表多的需注意
- **thread_cache**: 缓存连接线程, 影响连接建立效率, 对短连接影响较大
- **key_buffer_size**: 缓存MyISAM索引, 对MyISAM表性能影响极大
- **innodb_db_buffer_pool_size**: 对InnoDB极大影响, 缓存索引及数据
- **innodb_log_buff_size**: 缓存InnoDB写入日志, 影响写入效率
- **innodb_max_dirty_pages_pct**: 设置InnoDB Buffer中脏页占比

延伸阅读: <http://isky000.com/database/mysql-performamce-tuning-cache-parameter>

DTC2012

MySQL 性能优化最佳实践

误区

Cache/Buffer

- `query_cache`: 一定要有? 越大越好?
- `binlog_cache_size`: 越大越好?
- `table_cache`: 越多越好?
- `thread_cache`: 越多越好?
- `key_buffer_size`: 缓存数据? 越大越好?
- `innodb_db_buffer_pool_size`: 越大越好?
- `innodb_log_buff_size`: 越大越好?
- `innodb_max_dirty_pages_pct`: 脏页占比越多越快?

Connction

IO

⋮

延伸阅读: <http://isky000.com/database/mysql-performamce-tuning-cache-parameter>

DTC2012

MySQL 性能优化最佳实践

最佳实践

Params

Cache/Buffer

Connction

IO

⋮

- `query_cache`: 不超过256MB, 除非基本静态, InnoDB无效
- `binlog_cache_size`: 2MB~4MB, < 32MB
- `table_cache`: 1024, 具体需要根据实际环境调整
- `thread_cache`: 1024, < max_connectios
- `key_buffer_size`: 无MyISAM 16MB, 否则所有MYI大小之内尽可能大
- `innodb_db_buffer_pool_size`: 仅InnoDB, 所有文件大小之内尽可能大
- `innodb_log_buff_size`: 4MB~8MB, < 32MB
- `innodb_max_dirty_pages_pct`: $< 1G / \text{innodb_db_buffer_pool_size}(G) * 100$

延伸阅读: <http://isky000.com/database/mysql-performamce-tuning-cache-parameter>

DTC2012

MySQL 性能优化最佳实践

方法

Cache/Buffer

- **query_cache**: 不超过256MB, 除非基本静态, InnoDB无效
- **binlog_cache_size**: 2MB~4MB, < 32MB
- **table_cache**: 1024, 具体需要根据实际环境调整
- **thread_cache**: 1024, < max_connections
- **key_buffer_size**: 无MyISAM 16MB, 否则所有MYI大小之内尽可能大
- **innodb_db_buffer_pool_size**: 仅InnoDB, 所有文件大小之内尽可能大
- **innodb_log_buff_size**: 4MB~8MB, < 32MB
- **innodb_max_dirty_pages_pct**: $< 1G / \text{innodb_db_buffer_pool_size}(G) * 100$

Connction

- **max_connections**: 影响能够保持的最大客户端连接数, 属于自我保护类
- **max_connect_errors**: 某个用户允许最大登录失败次数, 类似于防破解
- **back_log**: 影响突发连接暴增场景, 比如服务器重启后瞬间
- **skip-name-resolve**: 取消对客户端的 DNS 反解, 影响连接和授权
- **interactive_timeout**和**wait_timeout**: 影响空闲连接最大可空闲时间

IO

⋮

延伸阅读: <http://isky000.com/database/mysql-performance-tuning-cache-parameter>

DTC2012

MySQL 性能优化最佳实践

误区

Cache/Buffer

Connction

IO

⋮

- **query_cache**: 不超过256MB, 除非基本静态, InnoDB无效
- **binlog_cache_size**: 2MB~4MB, < 32MB
- **table_cache**: 1024, 具体需要根据实际环境调整
- **thread_cache**: 1024, < max_connections
- **key_buffer_size**: 无MyISAM 16MB, 否则所有MYI大小之内尽可能大
- **innodb_db_buffer_pool_size**: 仅InnoDB, 所有文件大小之内尽可能大
- **innodb_log_buff_size**: 4MB~8MB, < 32MB
- **innodb_max_dirty_pages_pct**: $< 1G / \text{innodb_db_buffer_pool_size}(G) * 100$

- **max_connections**: 最大连接数越大越好?
- **max_connect_errors**: 最大错误数越小越好?
- **back_log**: back log队列越长越好吗?
- **skip-name-resolve**: 一定要忽略 DNS 反解吗?
- **interactive_timeout**和**wait_timeout**: 空闲时间越长越好吗?

延伸阅读: <http://isky000.com/database/mysql-performance-tuning-cache-parameter>

DTC2012

MySQL 性能优化最佳实践

最佳实践

Params

Cache/Buffer

- **query_cache**: 不超过256MB, 除非基本静态, InnoDB无效
- **binlog_cache_size**: 2MB~4MB, < 32MB
- **table_cache**: 1024, 具体需要根据实际环境调整
- **thread_cache**: 1024, < max_connections
- **key_buffer_size**: 无MyISAM 16MB, 否则所有MYI大小之内尽可能大
- **innodb_db_buffer_pool_size**: 仅InnoDB, 所有文件大小之内尽可能大
- **innodb_log_buff_size**: 4MB~8MB, < 32MB
- **innodb_max_dirty_pages_pct**: $< 1G / \text{innodb_db_buffer_pool_size}(G) * 100$

Connction

- **max_connections**: 1000~2000, < 10000
- **max_connect_errors**: >1000, 尽量大一点吧
- **back_log**: 100, < OS网络层设置
- **skip-name-resolve**: 建议启用, 确保授权都是用IP
- **interactive_timeout**和**wait_timeout**: 86400, 24小时基本足矣

IO

⋮

延伸阅读: <http://isky000.com/database/mysql-performamce-tuning-cache-parameter>

DTC2012

MySQL 性能优化最佳实践

方法

Params

Cache/Buffer

- **query_cache**: 不超过256MB, 除非基本静态, InnoDB无效
- **binlog_cache_size**: 2MB~4MB, < 32MB
- **table_cache**: 1024, 具体需要根据实际环境调整
- **thread_cache**: 1024, < max_connections
- **key_buffer_size**: 无MyISAM 16MB, 否则所有MYI大小之内尽可能大
- **innodb_db_buffer_pool_size**: 仅InnoDB, 所有文件大小之内尽可能大
- **innodb_log_buff_size**: 4MB~8MB, < 32MB
- **innodb_max_dirty_pages_pct**: $< 1G / \text{innodb_db_buffer_pool_size}(G) * 100$

Connction

- **max_connections**: 1000~2000, < 10000
- **max_connect_errors**: >1000, 尽量大一点吧
- **back_log**: 100, < OS网络层设置
- **skip-name-resolve**: 建议启用, 确保授权都是用IP
- **interactive_timeout**和**wait_timeout**: 86400, 24小时基本足矣

IO

- **innodb_flush_method**: innodb文件打开方式, linux下文件系统影响较大
- **innodb_flush_log_at_trx_commit**: 影响innodb日志事务刷新机制
- **innodb_file_per_table**: 影响表存储方式, 文件过大会影响性能
- **sync_binlog**: 影响binlog日志刷新到磁盘的机制

⋮

延伸阅读: <http://isky000.com/database/mysql-performamce-tuning-cache-parameter>

DTC2012

MySQL 性能优化最佳实践

误区

Params

Cache/Buffer

Connction

IO

⋮

- **query_cache**: 不超过256MB, 除非基本静态, InnoDB无效
 - **binlog_cache_size**: 2MB~4MB, < 32MB
 - **table_cache**: 1024, 具体需要根据实际环境调整
 - **thread_cache**: 1024, < max_connections
 - **key_buffer_size**: 无MyISAM 16MB, 否则所有MYI大小之内尽可能大
 - **innodb_db_buffer_pool_size**: 仅InnoDB, 所有文件大小之内尽可能大
 - **innodb_log_buff_size**: 4MB~8MB, < 32MB
 - **innodb_max_dirty_pages_pct**: $< 1G / \text{innodb_db_buffer_pool_size}(G) * 100$
-
- **max_connections**: 1000~2000, < 10000
 - **max_connect_errors**: >1000, 尽量大一点吧
 - **back_log**: 100, < OS网络层设置
 - **skip-name-resolve**: 建议启用, 确保授权都是用IP
 - **interactive_timeout**和**wait_timeout**: 86400, 24小时基本足矣
-
- **innodb_flush_method**: 注意系统之间的差异及文件系统差异
 - **innodb_flush_log_at_trx_commit**: 设为1最好吗?
 - **innodb_file_per_table**: 独享表空间更好吗?
 - **sync_binlog**: 刷新越频繁越好吗?

延伸阅读: <http://isky000.com/database/mysql-performance-tuning-cache-parameter>

DTC2012

MySQL 性能优化最佳实践

最佳实践

Params

Cache/Buffer

- **query_cache**: 不超过256MB, 除非基本静态, InnoDB无效
- **binlog_cache_size**: 2MB~4MB, < 32MB
- **table_cache**: 1024, 具体需要根据实际环境调整
- **thread_cache**: 1024, < max_connections
- **key_buffer_size**: 无MyISAM 16MB, 否则所有MYI大小之内尽可能大
- **innodb_db_buffer_pool_size**: 仅InnoDB, 所有文件大小之内尽可能大
- **innodb_log_buff_size**: 4MB~8MB, < 32MB
- **innodb_max_dirty_pages_pct**: $< 1G / \text{innodb_db_buffer_pool_size}(G) * 100$

Connction

- **max_connections**: 1000~2000, < 10000
- **max_connect_errors**: >1000, 尽量大一点吧
- **back_log**: 100, < OS网络层设置
- **skip-name-resolve**: 建议启用, 确保授权都是用IP
- **interactive_timeout**和**wait_timeout**: 86400, 24小时基本足矣

IO

- **innodb_flush_method**: O_DIRECT(Linux)
- **innodb_flush_log_at_trx_commit**: 2, 特别重要的设置为1, 不建议0
- **innodb_file_per_table**: 一般建议开启
- **sync_binlog**: 4~8, 非常频繁的系统可适当增大, 但不建议0

⋮

延伸阅读: <http://isky000.com/database/mysql-performamce-tuning-cache-parameter>

DTC2012

MySQL 性能优化最佳实践

方法

Storage Engine

MyISAM

InnoDB

⋮

- 尽量索引，MyISAM只缓存索引不缓存数据
- 调整读写优先级，根据实际需求，调整读写优先级
- 延迟插入，使用 `insert delay`，减少和 `select` 竞争
- 数据顺序操作，让 `insert` 全部到尾部，减少和 `select` 竞争
- 分解大操作，将大操作分解成多步小操作，防止长时间锁定
- 降低并发数，表锁会导致竞争激烈，通过排队机制提高效率
- 充分利用 Query Cache：对于静态数据，尽量使用 Query Cache
- ...

DTCC2012

MySQL 性能优化最佳实践

误区

Storage Engine

MyISAM

- key_buffer 会缓存所有 MyISAM 的数据和索引?
- MyISAM 读写一定是互斥的?
- MyISAM 读效率一定高于 InnoDB?
- 在 MyISAM 中所有的 count 都高效?
- ...

InnoDB

⋮

DTCC2012

MySQL 性能优化最佳实践

最佳实践

Storage Engine

MyISAM

- 不需要事务支持
- 并发相对较低
- 数据修改相对较少
- 以读为主
- 数据一致性要求较低
- ...

InnoDB

⋮

DTCC2012

MySQL 性能优化最佳实践

方法

Storage Engine

MyISAM

- 不需要事务支持
- 并发相对较低
- 数据修改相对较少
- 以读为主
- 数据一致性要求较低
- ...

InnoDB

- 主键尽可能小：所有非主键索引都需要存储主键
- 索引整合，减少冗余索引，降低数据量
- 避免全表扫描，因为会导致表锁
- 尽量自己控制事务，关闭 `autocommit`
- 尽量缓存所有数据和索引
- 合理设置 `innodb_flush_log_at_trx_commit`
- 充分利用索引避开表锁
- 避免主键更新
- ...

⋮

DTCC2012

MySQL 性能优化最佳实践

误区

Storage Engine

MyISAM

- 不需要事务支持
- 并发相对较低
- 数据修改相对较少
- 以读为主
- 数据一致性要求较低
- ...

InnoDB

- InnoDB_buffer_pool 只缓存索引?
- 任何情况下都是行锁?
- 事务越小越好?
- 日志刷新越快越好?
- ...

⋮

DTCC2012

MySQL 性能优化最佳实践

最佳实践

Storage Engine

MyISAM

- 不需要事务支持
- 并发相对较低
- 数据修改相对较少
- 以读为主
- 数据一致性要求较低
- ...

InnoDB

- 需要事务支持
- 并发较大
- 数据变更比较频繁
- 数据一致性要求较高
- 硬件设备内存较大，远大于索引数据量
- ...

⋮

DTCC2012

MySQL 性能优化最佳实践

方法

Schema

优化数据类型

- 合理设置长度
- 尽量避免使用lob字段
- 尽量使用更小的数据类型
- ...

调整字符编码

适当拆分

适度冗余

⋮

延伸阅读: <http://isky000.com/database/mysql-performamce-tuning-schema>

DTCC2012

MySQL 性能优化最佳实践

误区

Schema

优化数据类型

- 预留越长越好？
- INT(1) 代表存放1位长度的整数值？
- MySQL能够高效处理各种数据类型？
- ...

调整字符编码

适当拆分

适度冗余

⋮

延伸阅读: <http://isky000.com/database/mysql-performamce-tuning-schema>

DTCC2012

MySQL 性能优化最佳实践

最佳实践

Schema

优化数据类型

- 避免DOUBLE，区分开 TINYINT / INT / BIGINT
- 尽量避免TEXT，VARCHAR不要留过大缓冲
- 尽量TIMESTAMP，能用DATE不用DATETIME
- 拒绝 LOB类型，可尝试 ENUM & SET

调整字符编码

适当拆分

适度冗余

⋮

延伸阅读: <http://isky000.com/database/mysql-performamce-tuning-schema>

DTCC2012

MySQL 性能优化最佳实践

方法

Schema

优化数据类型

- 避免DOUBLE，区分开 TINYINT / INT / BIGINT
- 尽量避免TEXT，VARCHAR不要留过大缓冲
- 尽量TIMESTAMP，能用DATE不用DATETIME
- 拒绝 LOB类型，可尝试 ENUM & SET

调整字符编码

- 够用就可以，选择更小的字符集
- 保证语言环境能够支持覆盖
- ...

适当拆分

适度冗余

⋮

延伸阅读: <http://isky000.com/database/mysql-performamce-tuning-schema>

DTCC2012

MySQL 性能优化最佳实践

误区

Schema

优化数据类型

- 避免DOUBLE，区分开 TINYINT / INT / BIGINT
- 尽量避免TEXT，VARCHAR不要留过大缓冲
- 尽量TIMESTAMP，能用DATE不用DATETIME
- 拒绝 LOB类型，可尝试 ENUM & SET

调整字符编码

- 一定要整个 Server 统一？
- 一定要全库统一？
- 一定要全表统一？

适当拆分

适度冗余

⋮

延伸阅读: <http://isky000.com/database/mysql-performamce-tuning-schema>

DTCC2012

MySQL 性能优化最佳实践

最佳实践

Schema

优化数据类型

- 避免DOUBLE，区分开 TINYINT / INT / BIGINT
- 尽量避免TEXT，VARCHAR不要留过大缓冲
- 尽量TIMESTAMP，能用DATE不用DATETIME
- 拒绝 LOB类型，可尝试 ENUM & SET

调整字符编码

- 纯拉丁字符能表示的内容，没必要选择 latin1
- 数据类型可精确到字段，极端情况下单独设置
- 确定不需要多语言，就没必要UNICODE类型

适当拆分

适度冗余

⋮

延伸阅读: <http://isky000.com/database/mysql-performamce-tuning-schema>

DTCC2012

MySQL 性能优化最佳实践

方法

Schema

优化数据类型

- 避免DOUBLE，区分开 TINYINT / INT / BIGINT
- 尽量避免TEXT，VARCHAR不要留过大缓冲
- 尽量TIMESTAMP，能用DATE不用DATETIME
- 拒绝 LOB类型，可尝试 ENUM & SET

调整字符编码

- 纯拉丁字符能表示的内容，没必要选择 latin1
- 数据类型可精确到字段，极端情况下单独设置
- 确定不需要多语言，就没必要UNICODE类型

适当拆分

- 降低单条记录长度，使单个数据块中存放尽可能多的纪录
- ...

适度冗余

⋮

延伸阅读: <http://isky000.com/database/mysql-performamce-tuning-schema>

DTCC2012

MySQL 性能优化最佳实践

误区

Schema

优化数据类型

- 避免DOUBLE，区分开 TINYINT / INT / BIGINT
- 尽量避免TEXT，VARCHAR不要留过大缓冲
- 尽量TIMESTAMP，能用DATE不用DATETIME
- 拒绝 LOB类型，可尝试 ENUM & SET

调整字符编码

- 纯拉丁字符能表示的内容，没必要选择 latin1
- 数据类型可精确到字段，极端情况下单独设置
- 确定不需要多语言，就没必要UNICODE类型

适当拆分

- 数据表一定要和程序对象对应才叫合理的设计？
- 只要不在 select 子句中的字段就不会被访问？
- ...

适度冗余

⋮

延伸阅读: <http://isky000.com/database/mysql-performamce-tuning-schema>

DTCC2012

MySQL 性能优化最佳实践

最佳实践

Schema

优化数据类型

- 避免DOUBLE，区分开 TINYINT / INT / BIGINT
- 尽量避免TEXT，VARCHAR不要留过大缓冲
- 尽量TIMESTAMP，能用DATE不用DATETIME
- 拒绝 LOB类型，可尝试 ENUM & SET

调整字符编码

- 纯拉丁字符能表示的内容，没必要选择 latin1
- 数据类型可精确到字段，极端情况下单独设置
- 确定不需要多语言，就没必要UNICODE类型

适当拆分

- 将不常使用的字段以及大字段拆分到独立附属表中
- ...

适度冗余

⋮

延伸阅读: <http://isky000.com/database/mysql-performamce-tuning-schema>

DTCC2012

MySQL 性能优化最佳实践

方法

Schema

优化数据类型

- 避免DOUBLE，区分开 TINYINT / INT / BIGINT
- 尽量避免TEXT，VARCHAR不要留过大缓冲
- 尽量TIMESTAMP，能用DATE不用DATETIME
- 拒绝 LOB类型，可尝试 ENUM & SET

调整字符编码

- 纯拉丁字符能表示的内容，没必要选择 latin1
- 数据类型可精确到字段，极端情况下单独设置
- 确定不需要多语言，就没必要UNICODE类型

适当拆分

- 将不常使用的字段以及大字段拆分到独立附属表中
- ...

适度冗余

- 冗余常用字段，减少关联查询
- ...

⋮

延伸阅读: <http://isky000.com/database/mysql-performamce-tuning-schema>

DTCC2012

MySQL 性能优化最佳实践

误区

Schema

优化数据类型

- 避免DOUBLE，区分开 TINYINT / INT / BIGINT
- 尽量避免TEXT，VARCHAR不要留过大缓冲
- 尽量TIMESTAMP，能用DATE不用DATETIME
- 拒绝 LOB类型，可尝试 ENUM & SET

调整字符编码

- 纯拉丁字符能表示的内容，没必要选择 latin1
- 数据类型可精确到字段，极端情况下单独设置
- 确定不需要多语言，就没必要UNICODE类型

适当拆分

- 将不常使用的字段以及大字段拆分到独立附属表中
- ...

适度冗余

- 严格遵循第三范式的设计才是最高效的设计？
- ...

⋮

延伸阅读: <http://isky000.com/database/mysql-performamce-tuning-schema>

DTCC2012

MySQL 性能优化最佳实践

最佳实践

Schema

优化数据类型

- 避免DOUBLE，区分开 TINYINT / INT / BIGINT
- 尽量避免TEXT，VARCHAR不要留过大缓冲
- 尽量TIMESTAMP，能用DATE不用DATETIME
- 拒绝 LOB类型，可尝试 ENUM & SET

调整字符编码

- 纯拉丁字符能表示的内容，没必要选择 latin1
- 数据类型可精确到字段，极端情况下单独设置
- 确定不需要多语言，就没必要UNICODE类型

适当拆分

- 将不常使用的字段以及大字段拆分到独立附属表中
- ...

适度冗余

- 被频繁引用且只能通过 Join 2张（或者更多）表的方式才能得到的独立小字段，建议冗余
- ...

⋮

延伸阅读: <http://isky000.com/database/mysql-performamce-tuning-schema>

DTCC2012

MySQL 性能优化最佳实践

方法

Index

合适的字段

- 提高过滤性
- 降低索引的更新分裂
- 避免无效索引
- 非不得已不用外键

合适的顺序

合适的比例

合理的维护

⋮

延伸阅读: <http://isky000.com/database/mysql-performance-tuning-index>

DTCC2012

MySQL 性能优化最佳实践

误区

Index

合适的字段

- 只要在 **Where** 条件中就应该创建索引？
- 只要创建了索引，就能被 **SQL** 使用？
- 使用索引一定比不使用索引快？

合适的顺序

合适的比例

合理的维护

⋮

延伸阅读: <http://isky000.com/database/mysql-performance-tuning-index>

DTCC2012

MySQL 性能优化最佳实践

最佳实践

合适的字段

- 给索引的字段设置默认值
- 不要让含NULL的字段进入组合索引
- 删除过滤性低的字段的索引，可能性能更差
- 不能在索引字段上做运算，会失效
- 避免频繁更新的字段进入索引，增加IO负担
- 尽量覆盖索引(MySQL排序效率不高)

合适的顺序

合适的比例

合理的维护

⋮

延伸阅读: <http://isky000.com/database/mysql-performance-tuning-index>

DTCC2012

MySQL 性能优化最佳实践

方法

Index

合适的字段

- 给索引的字段设置默认值
- 不要让含NULL的字段进入组合索引
- 删除过滤性低的字段的索引，可能性能更差
- 不能在索引字段上做运算，会失效
- 避免频繁更新的字段进入索引，增加IO负担
- 尽量覆盖索引(MySQL排序效率不高)

合适的顺序

- 提早过滤
- 减少排序

合适的比例

合理的维护

⋮

延伸阅读: <http://isky000.com/database/mysql-performance-tuning-index>

DTCC2012

MySQL 性能优化最佳实践

误区

- 给索引的字段设置默认值
- 不要让含NULL的字段进入组合索引
- 删除过滤性低的字段的索引，可能性能更差
- 不能在索引字段上做运算，会失效
- 避免频繁更新的字段进入索引，增加IO负担
- 尽量覆盖索引(MySQL排序效率不高)

合适的字段

合适的顺序

- 只要将where条件中的字段全部放在索引中就可以了？
- 索引的顺序对 SQL 访问没有影响？

合适的比例

合理的维护

⋮

延伸阅读: <http://isky000.com/database/mysql-performance-tuning-index>

DTCC2012

MySQL 性能优化最佳实践

最佳实践

Index

合适的字段

- 给索引的字段设置默认值
- 不要让含**NULL**的字段进入组合索引
- 删除过滤性低的字段的索引，可能性能更差
- 不能在索引字段上做运算，会失效
- 避免频繁更新的字段进入索引，增加IO负担
- 尽量覆盖索引(MySQL排序效率不高)

合适的顺序

- 过滤性越高的字段需要越靠前
- 核心SQL覆盖索引，确保尽可能高效
- 不干扰过滤前提下，排序字段进入索引
- 多 SQL 综合考虑，重复利用索引

合适的比例

合理的维护

⋮

延伸阅读: <http://isky000.com/database/mysql-performance-tuning-index>

DTCC2012

MySQL 性能优化最佳实践

方法

Index

合适的字段

- 给索引的字段设置默认值
- 不要让含**NULL**的字段进入组合索引
- 删除过滤性低的字段的索引，可能性能更差
- 不能在索引字段上做运算，会失效
- 避免频繁更新的字段进入索引，增加IO负担
- 尽量覆盖索引(MySQL排序效率不高)

合适的顺序

- 过滤性越高的字段需要越靠前
- 核心SQL覆盖索引，确保尽可能高效
- 不干扰过滤前提下，排序字段进入索引
- 多 SQL 综合考虑，重复利用索引

合适的比例

- 控制索引长度，尤其是较长的字符串字段

合理的维护

⋮

延伸阅读: <http://isky000.com/database/mysql-performance-tuning-index>

DTCC2012

MySQL 性能优化最佳实践

误区

合适的字段

- 给索引的字段设置默认值
- 不要让含**NULL**的字段进入组合索引
- 删除过滤性低的字段的索引，可能性能更差
- 不能在索引字段上做运算，会失效
- 避免频繁更新的字段进入索引，增加IO负担
- 尽量覆盖索引(MySQL排序效率不高)

合适的顺序

- 过滤性越高的字段需要越靠前
- 核心**SQL**覆盖索引，确保尽可能高效
- 不干扰过滤前提下，排序字段进入索引
- 多 **SQL** 综合考虑，重复利用索引

合适的比例

- 索引可以无限大？
- 索引只能使用整个字段？

合理的维护

⋮

延伸阅读: <http://isky000.com/database/mysql-performance-tuning-index>

DTCC2012

MySQL 性能优化最佳实践

最佳实践

Index

合适的字段

- 给索引的字段设置默认值
- 不要让含**NULL**的字段进入组合索引
- 删除过滤性低的字段的索引，可能性能更差
- 不能在索引字段上做运算，会失效
- 避免频繁更新的字段进入索引，增加IO负担
- 尽量覆盖索引(MySQL排序效率不高)

合适的顺序

- 过滤性越高的字段需要越靠前
- 核心**SQL**覆盖索引，确保尽可能高效
- 不干扰过滤前提下，排序字段进入索引
- 多 **SQL** 综合考虑，重复利用索引

合适的比例

- 必须回表取数据时，字符字段前缀索引(8)
- 不用回表取数据时，建议整个字段

合理的维护

⋮

延伸阅读: <http://isky000.com/database/mysql-performance-tuning-index>

DTCC2012

MySQL 性能优化最佳实践

方法

Index

合适的字段

- 给索引的字段设置默认值
- 不要让含**NULL**的字段进入组合索引
- 删除过滤性低的字段的索引，可能性能更差
- 不能在索引字段上做运算，会失效
- 避免频繁更新的字段进入索引，增加IO负担
- 尽量覆盖索引(MySQL排序效率不高)

合适的顺序

- 过滤性越高的字段需要越靠前
- 核心SQL覆盖索引，确保尽可能高效
- 不干扰过滤前提下，排序字段进入索引
- 多 SQL 综合考虑，重复利用索引

合适的比例

- 必须回表取数据时，字符字段前缀索引(8)
- 不用回表取数据时，建议整个字段

合理的维护

- 定期维护存在频繁增删改字段的索引

⋮

延伸阅读: <http://isky000.com/database/mysql-performance-tuning-index>

DTCC2012

MySQL 性能优化最佳实践

误区

- 给索引的字段设置默认值
- 不要让含NULL的字段进入组合索引
- 删除过滤性低的字段的索引，可能性能更差
- 不能在索引字段上做运算，会失效
- 避免频繁更新的字段进入索引，增加IO负担
- 尽量覆盖索引(MySQL排序效率不高)

合适的字段

合适的顺序

- 过滤性越高的字段需要越靠前
- 核心SQL覆盖索引，确保尽可能高效
- 不干扰过滤前提下，排序字段进入索引
- 多 SQL 综合考虑，重复利用索引

合适的比例

- 必须回表取数据时，字符字段前缀索引(8)
- 不用回表取数据时，建议整个字段

合理的维护

- 索引不会出现碎片？
- 索引会自动维护？

⋮

延伸阅读: <http://isky000.com/database/mysql-performance-tuning-index>

DTCC2012

MySQL 性能优化最佳实践

最佳实践

Index

合适的字段

- 给索引的字段设置默认值
- 不要让含**NULL**的字段进入组合索引
- 删除过滤性低的字段的索引，可能性能更差
- 不能在索引字段上做运算，会失效
- 避免频繁更新的字段进入索引，增加IO负担
- 尽量覆盖索引(MySQL排序效率不高)

合适的顺序

- 过滤性越高的字段需要越靠前
- 核心SQL覆盖索引，确保尽可能高效
- 不干扰过滤前提下，排序字段进入索引
- 多 SQL 综合考虑，重复利用索引

合适的比例

- 必须回表取数据时，字符字段前缀索引(8)
- 不用回表取数据时，建议整个字段

合理的维护

- 每月维护（重建）非核心表上的索引(可以的前提)
- 每季/年维护核心表上的索引(可以的前提)

⋮

延伸阅读: <http://isky000.com/database/mysql-performance-tuning-index>

DTCC2012

MySQL 性能优化最佳实践

方法

SQL

调整执行计划

- 缩短访问的路径
- 尽早过滤数据
- 尽可能减少排序
- 降低 SQL 复杂度
- 避开 MySQL 优化器 Bug，比如子查询
- ...

延伸阅读: <http://isky000.com/database/mysql-performance-tuning-sql>

DTCC2012

MySQL 性能优化最佳实践

误区

TSQL

调整执行计划

- `count(1)`和`count(primary_key)` 优于 `count(*)`
- `count(column)` 和 `count(*)` 一样
- `select a,b from ...` 比 `select a,b,c from ...` 可以让数据库访问更少的数据量
- `order by` 一定需要排序操作
- 执行计划中有 `filesort` 就会进行磁盘文件排序
- ...

延伸阅读: <http://isky000.com/database/mysql-performance-tuning-sql>

DTCC2012

MySQL 性能优化最佳实践

优化原则

SQL

调整执行计划

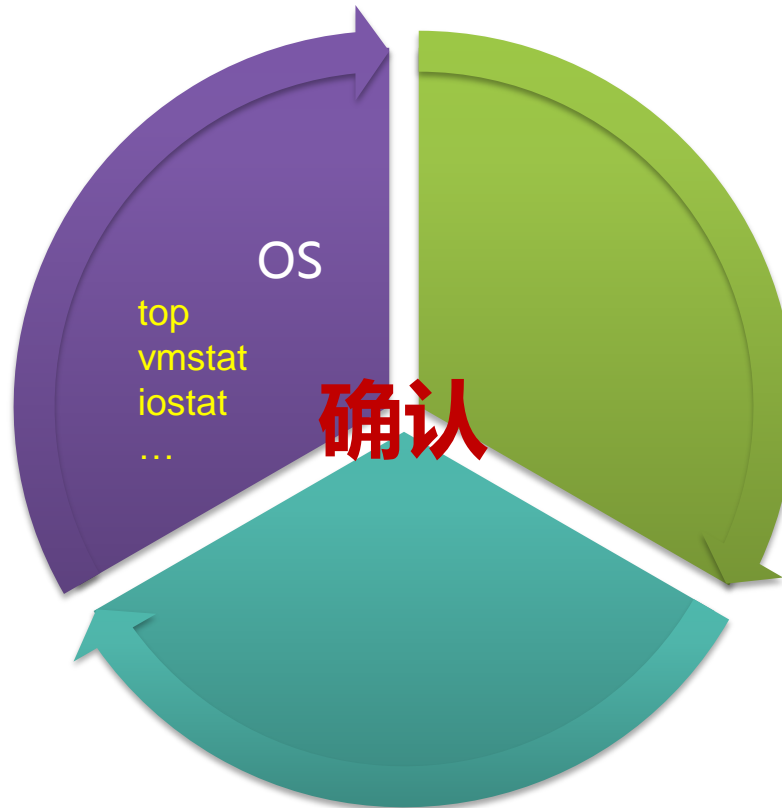
- 减少表连接，减少复杂 SQL，拆分成简单SQL
- 减少排序：非必要不排序，利用索引排序，减少参与排序的记录数
- 尽量避免 `select *`
- 尽量用 `join` 代替子查询
- 尽量少使用 `or`，使用 `in` 或者 `union(union all)` 代替
- 尽量用 `union all` 代替 `union`
- 尽量早的将无用数据过滤：选择更优的索引，先分页再Join...
- 避免类型转换：索引失效
- 优先优化高并发的 SQL，而不是执行频率低某些“大”SQL
- 从全局出发优化，而不是片面调整
- 尽可能对每一条SQL进行 `explain`
- ...

延伸阅读：<http://isky000.com/database/mysql-performance-tuning-sql>

DTCC2012

MySQL 性能优化最佳实践

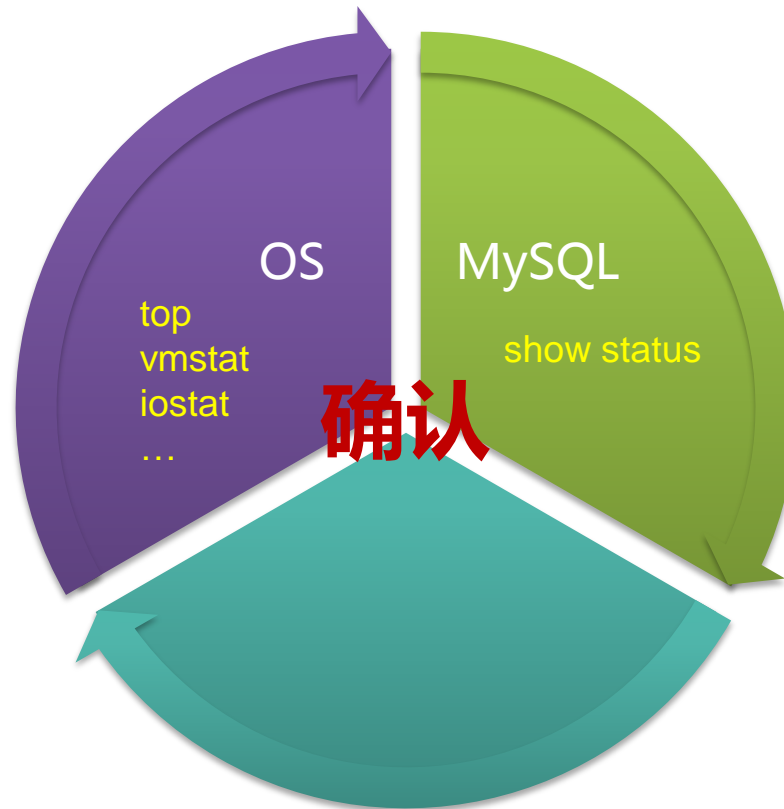
确认结果



DTCC2012

MySQL 性能优化最佳实践

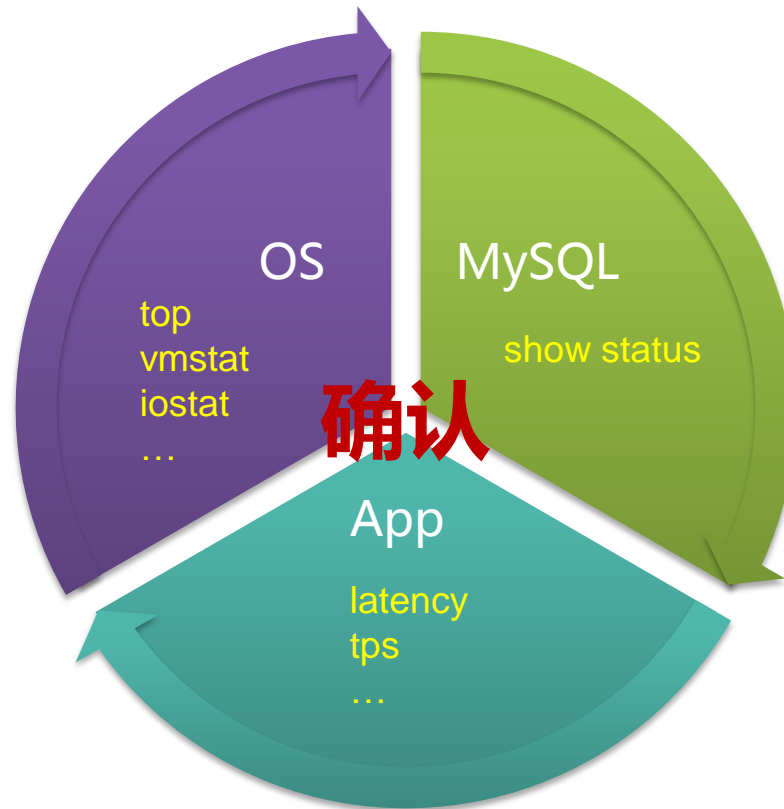
确认结果



DTCC2012

MySQL 性能优化最佳实践

确认结果



DTCC2012

MySQL 性能优化最佳实践

Thanks, Q & A

<http://isky000.com>

[Twitter: @sky000](#)

[Weibo: @简朝阳](#)

DTCC2012