

MySQL性能优化概述

陈 慧
MySQL 工程师



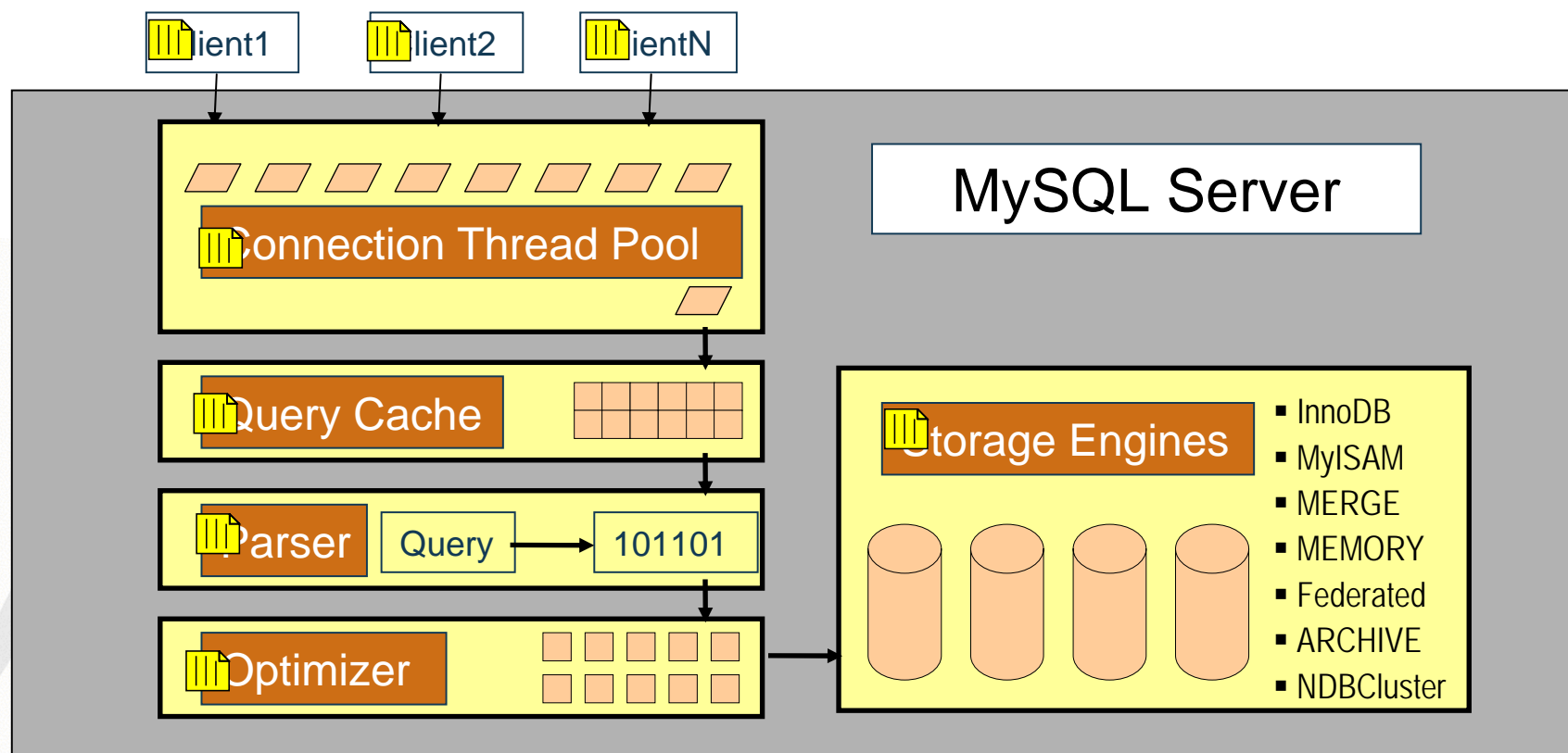
内 容

- MySQL调优概述
- 如何定位性能瓶颈
- 存储引擎和调优
- MySQL Cluster

MySQL调优概述

- 硬件 软件 网络环境
- 数据表结构
- 索引
- SQL语句
- 参数
- 存储引擎




MySQL调优概述



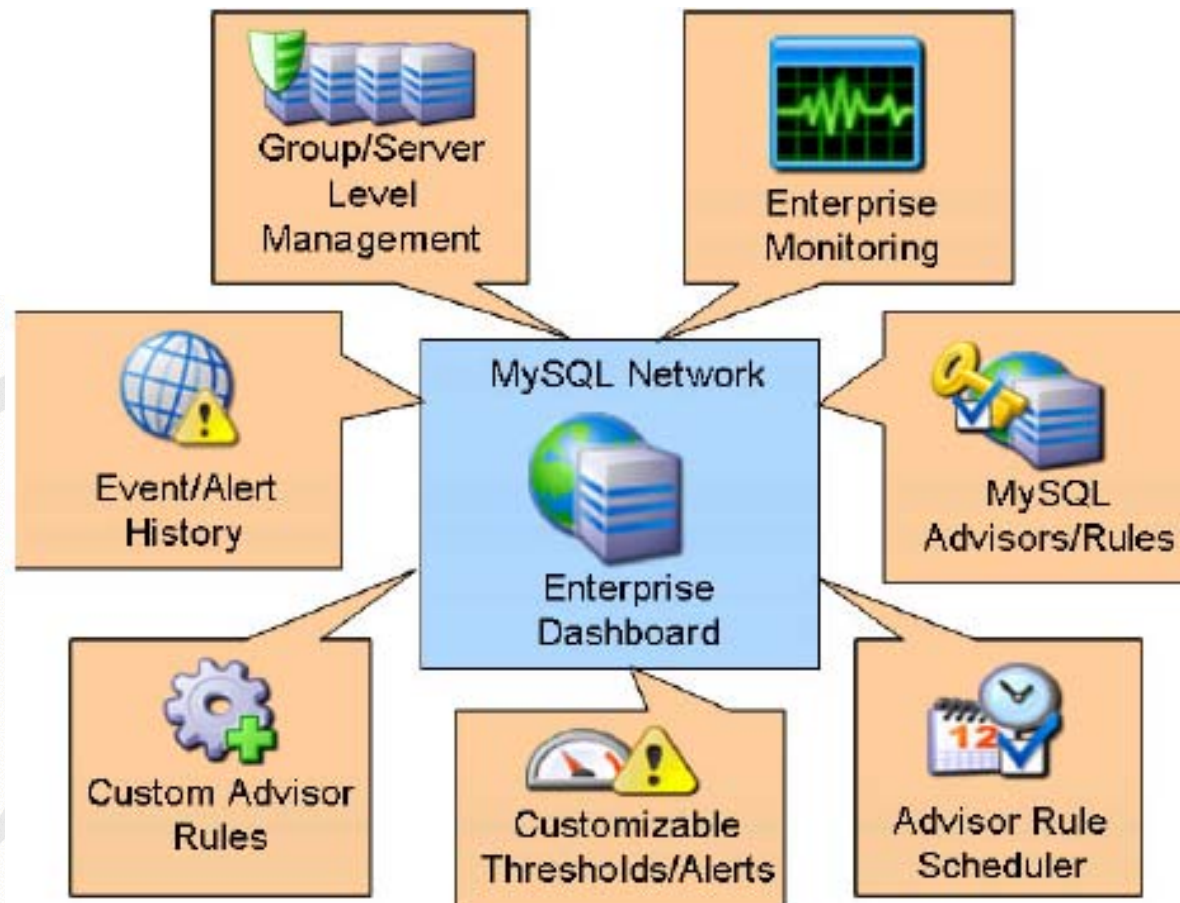
如何定位性能瓶颈



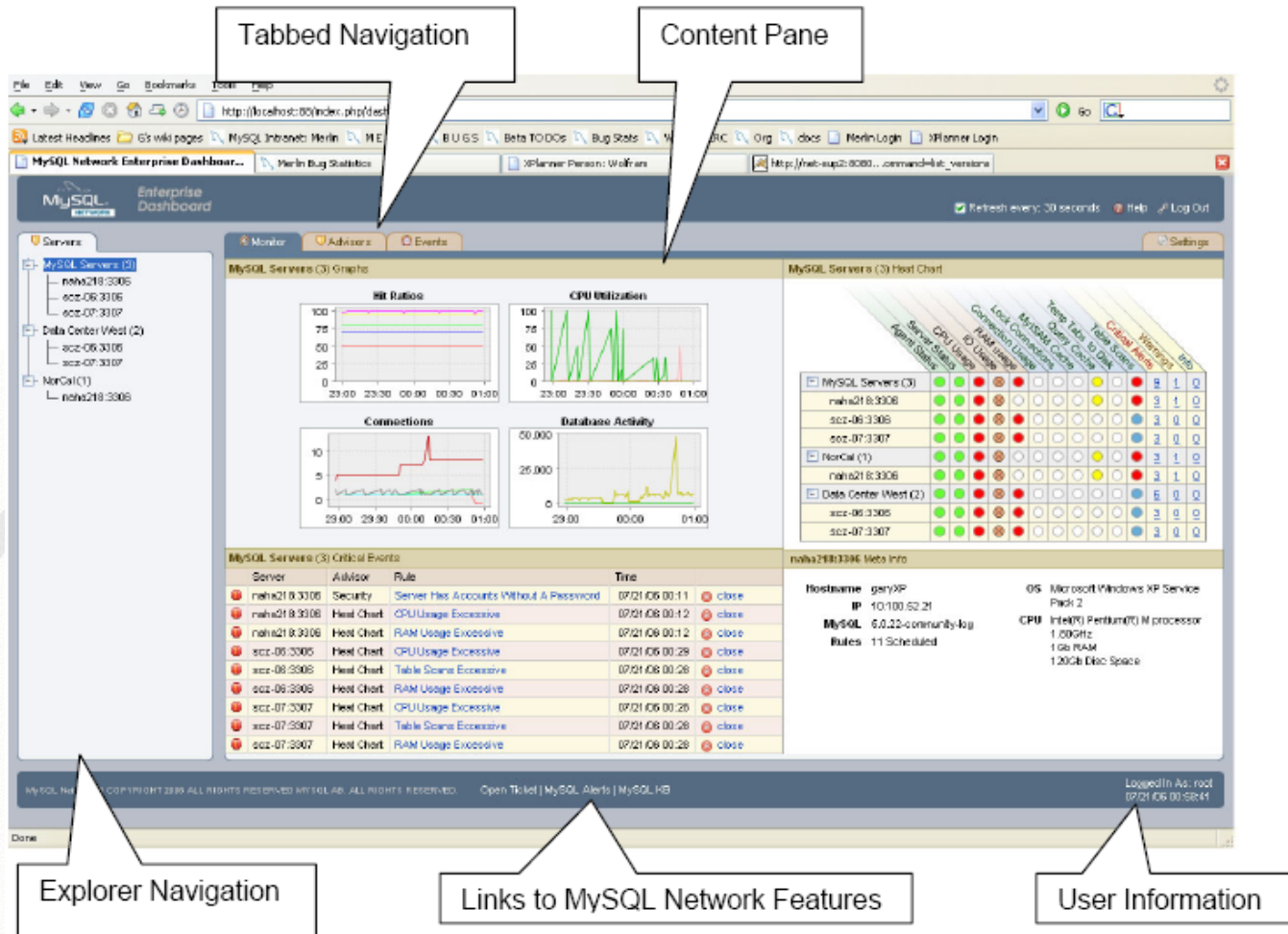
定位性能问题

- MySQL Monitor & Advisor 
- Explain 
- Profiling 

MySQL Monitor & Advisor



MySQL Monitor & Advisor



The screenshot displays the MySQL Monitor & Advisor web interface. Callouts point to the following features:

- Tabbed Navigation:** Located at the top of the interface, showing tabs for Monitor, Advisers, and Events.
- Content Pane:** The main area displaying monitoring data, including graphs for CPU Utilization, Connections, and Database Activity, as well as a table of critical events.
- Explorer Navigation:** A tree view on the left side showing the hierarchy of monitored servers.
- Links to MySQL Network Features:** A set of links at the bottom, including "Open Ticket", "MySQL Alerts", and "MySQL KB".
- User Information:** A box at the bottom right showing the current user (root) and login time.

The interface also includes a sidebar with a tree view of servers, a top navigation bar with links like "MySQL Enterprise Dashboard", "MySQL Network Enterprise Dashboard", and "MySQL Bug Statistics", and a bottom status bar with copyright information.

Query Execution Plan (EXPLAIN)

- **EXPLAIN** 模拟优化器执行查询,返回执行计划
- **EXPLAIN** tells you:
 - In which order the tables are read
 - What types of read operations that are made
 - Which indexes could have been used
 - Which indexes are used
 - How the tables refer to each other
 - How many rows the optimizer estimates to retrieve from each table

Query Execution Plan (EXPLAIN)

```
mysql> EXPLAIN SELECT Country.Name FROM Country JOIN CountryLanguage
```

```
-> ON Code=CountryCode WHERE Language = 'Russian' \G
```

```
***** 1. row *****
```

```
id: 1
```

```
select_type: SIMPLE
```

```
table: Country
```

```
type: ALL
```

```
possible_keys: PRIMARY
```

```
key: NULL
```

```
key_len: NULL
```

```
ref: NULL
```

```
rows: 239
```

```
Extra:
```

```
***** 2. row *****
```

```
id: 1
```

```
select_type: SIMPLE
```

```
table: CountryLanguage
```

```
type: eq_ref
```

```
possible_keys: PRIMARY
```

```
key: PRIMARY
```

```
key_len: 33
```

```
ref: world_test.Country.Code,const
```

```
rows: 1
```

```
Extra: Using where; Using index
```

```
2 rows in set (0.00 sec)
```

Query Execution Plan (EXPLAIN)

```
mysql> ALTER TABLE CountryLanguage ADD INDEX (Language);
```

```
Query OK, 984 rows affected (0.24 sec)
```

```
Records: 984 Duplicates: 0 Warnings: 0
```

```
mysql> EXPLAIN SELECT Country.Name FROM Country JOIN CountryLanguage
```

```
-> ON Code=CountryCode WHERE Language = 'Russian' \G
```

```
***** 1. row *****
```

```
id: 1
```

```
select_type: SIMPLE
```

```
table: CountryLanguage
```

```
type: ref
```

```
possible_keys: PRIMARY,Language
```

```
key: Language
```

```
key_len: 30
```

```
ref: const
```

```
rows: 15
```

```
Extra: Using where
```

```
***** 2. row *****
```

```
id: 1
```

```
select_type: SIMPLE
```

```
table: Country
```

```
type: eq_ref
```

```
possible_keys: PRIMARY
```

```
key: PRIMARY
```

```
key_len: 3
```

```
ref: world_test.CountryLanguage.CountryCode
```

```
rows: 1
```

```
Extra:
```

```
2 rows in set (0.00 sec)
```

Profiling

```
mysql> set @@profiling=1;
Query OK, 0 rows affected (0.00 sec)

mysql> select city.name, country.name from city, country where countrycode=code
and continent='Asia' order by city.name limit 5;
+-----+-----+
| name   | name   |
+-----+-----+
| Abadan  | Iran   |
| Abha    | Saudi Arabia |
| Abiko   | Japan  |
| Abohar  | India  |
| Abbottabad | Pakistan |
+-----+-----+
5 rows in set (0.00 sec)
```

Profiling

```
mysql> show profiles;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 1 | 0.00785400 | select city.name, country.name from city, country where countrycode=code and continent='Asia' order by city.name limit 5 |
+-----+-----+-----+
```

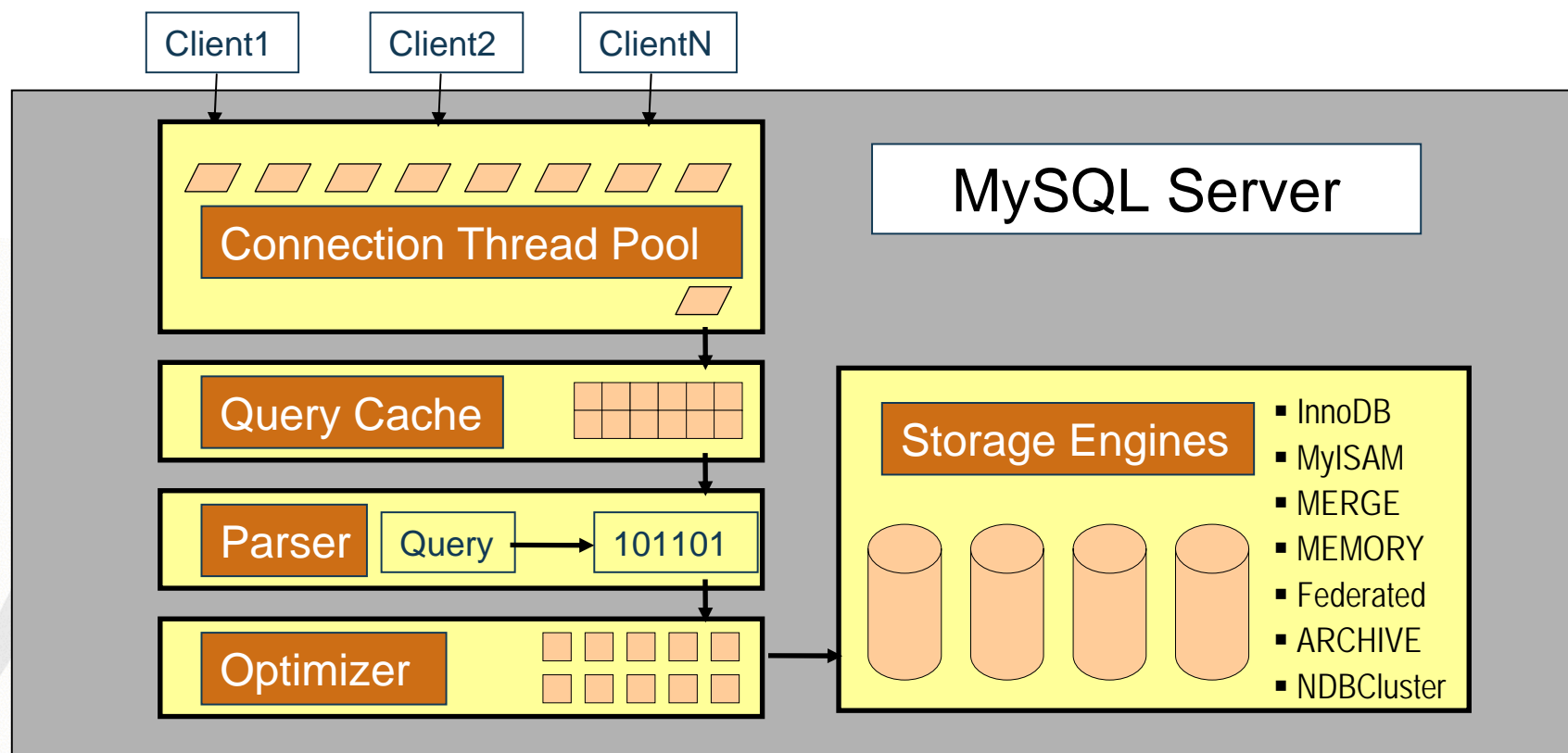

Profiling

```
mysql> show profile for query 1;
+-----+-----+
| Status                | Duration |
+-----+-----+
| (initialization)      | 0.00010875 |
| Opening tables         | 0.00001975 |
| System lock           | 0.000004675 |
| Table lock            | 0.000001450 |
| init                  | 0.000004500 |
| optimizing             | 0.000002175 |
| statistics            | 0.000009025 |
| preparing             | 0.000003200 |
| executing              | 0.000000500 |
| Sorting result        | 0.000000475 |
| Sending data          | 0.00045700 |
| end                   | 0.000000700 |
| query end             | 0.000000600 |
| freeing items         | 0.00697600 |
| closing tables        | 0.000001550 |
| logging slow query    | 0.000000400 |
+-----+-----+
16 rows in set (0.00 sec)
```

存储引擎和调优



MySQL的存储引擎



可插存储引擎结构

- MySQL 支持许多存储引擎，它们处理不同类型的表
- 通过选择，创建或扩展存储引擎来更好的适应应用的特殊要求
- 对你来说最重要的是什么？
 - 集中读
 - OLTP（联机事务处理）
 - 事务处理
 - 性能
 - 可伸缩性
 - 并发级别
 - 索引类型
 - 存储利用
 - 高可靠性
 - 复制
 - 在线备份
 - 数据仓库
 - 不相关关键字
 - 占用空间小
 - 行级别锁
 - 嵌入式
 - 表级别锁
 - 集群

存储引擎特点

- **MyISAM**

- 特点: 非常有效率的存储, 易于处理高速数据加载
- 适用于: 高流量网站 数据仓库

- **InnoDB**

- 特点: 提供ACID 事务处理支持 
- 适用于: 在线事务处理应用

- **Archive**

- 特点: 自动数据压缩
- 适用于: 历史数据仓库, 数据存档, 数据审计

- **NDB**

- 特点: 支持事务处理 提供高可靠机制
- 适用于: 高可靠 不停顿业务, 快速目录/关键字查找应用




存储引擎的特点

- 锁的机制
- 索引
- 对事务的支持
- 参数
- 备份机制



锁

- **InnoDB 支持**

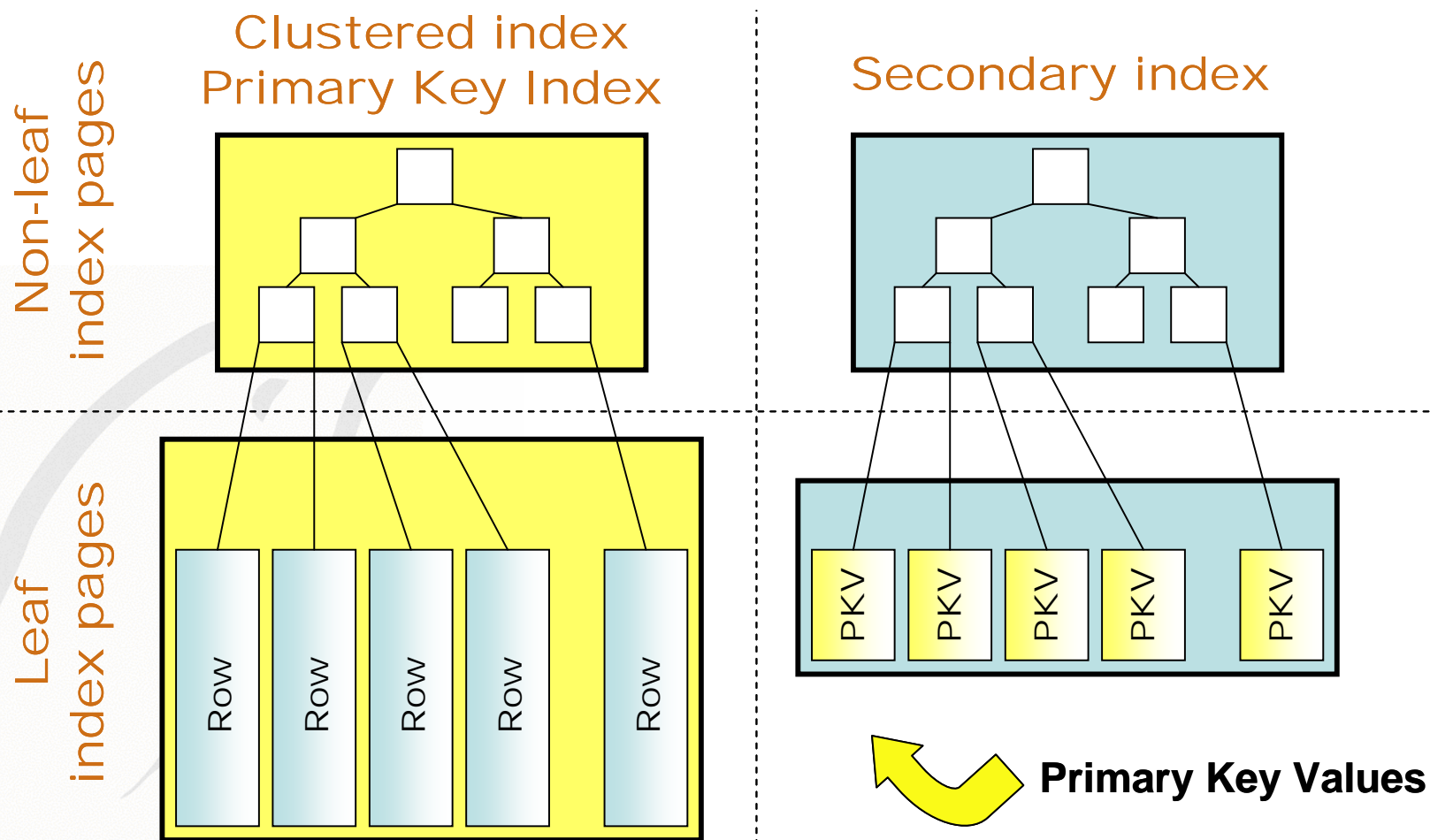
- 行级锁

- Finest level of lock granularity 
 - Only the row that is read or updated is locked 
 - Allows other concurrent transactions to access other rows on the same page 

- 表级锁

- Only used when there are changes to the table structure itself (as is the case with Alter table) 
 - Ensures the utmost integrity of the data it contains when doing direct changes to the table structure itself 
 - Controlled by the --InnoDB-table-locks server setting

索引



参数

[mysqld]

You can write your other MySQL server options here

...

Data files must be able to hold your data and indexes.

Make sure that you have enough free disk space.

innodb_data_file_path = ibdata1:10M:autoextend

#

Set buffer pool size to 50-80% of your computer's memory

innodb_buffer_pool_size=1024M

innodb_additional_mem_pool_size=10M

#

Set the log file size to about 25% of the buffer pool size

innodb_log_file_size=250M

innodb_log_buffer_size=8M

#

innodb_flush_log_at_trx_commit=1

InnoDB调优窍门

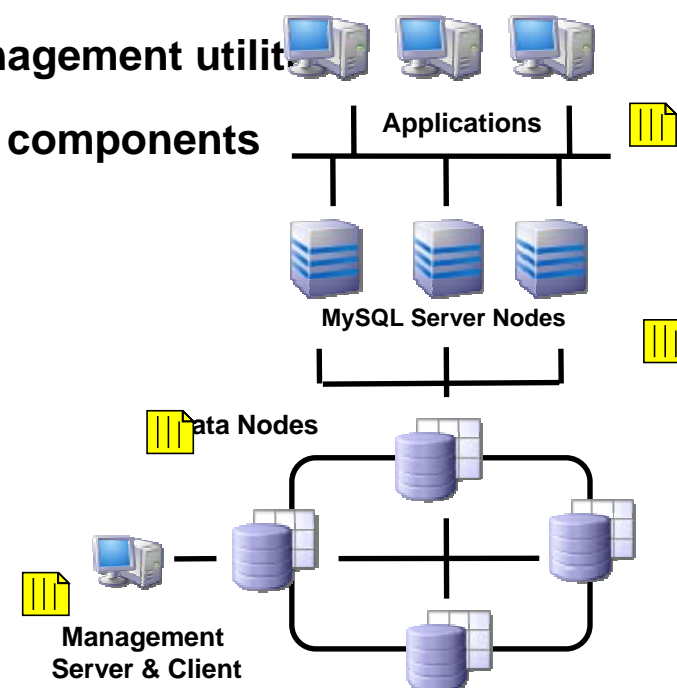
- 尽量使用短的,整型主键
- **Load/Insert** 数据时尽量用主键的顺序
- 增加日志文件大小
- 避免大的事务回滚
- 避免大量插入
- 尽量使用前缀索引

MySQL Cluster




高可靠性解决方案: 集群

- 成本
 - Use commodity hardware to accommodate the growth of users, traffic, and data
- 容错
 - No single point of failure
- 高可靠性
 - Data is replicated across nodes and always available
 - Automatic fail-over
- 可扩展性
 - Distributes large workloads
 - Replicas for Read
 - Partitions for Write
 - Supports “Scale Out”
- 高性能
 - Load balanced
 - Memory based storage engine
 - Designed to handle thousands of requests per second
- 简化管理
 - Cluster management utility
 - Commodity components

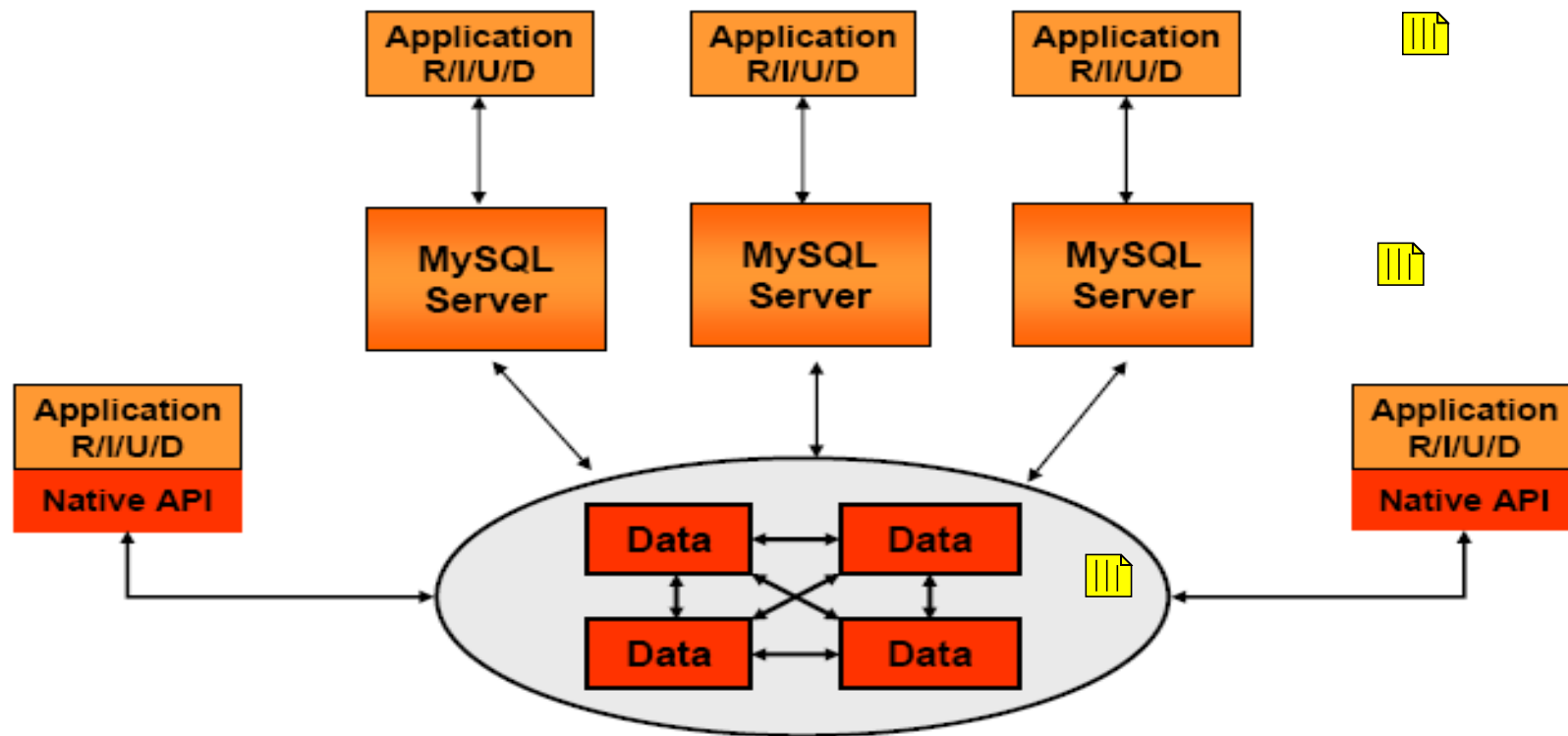


MySQL Cluster 概述



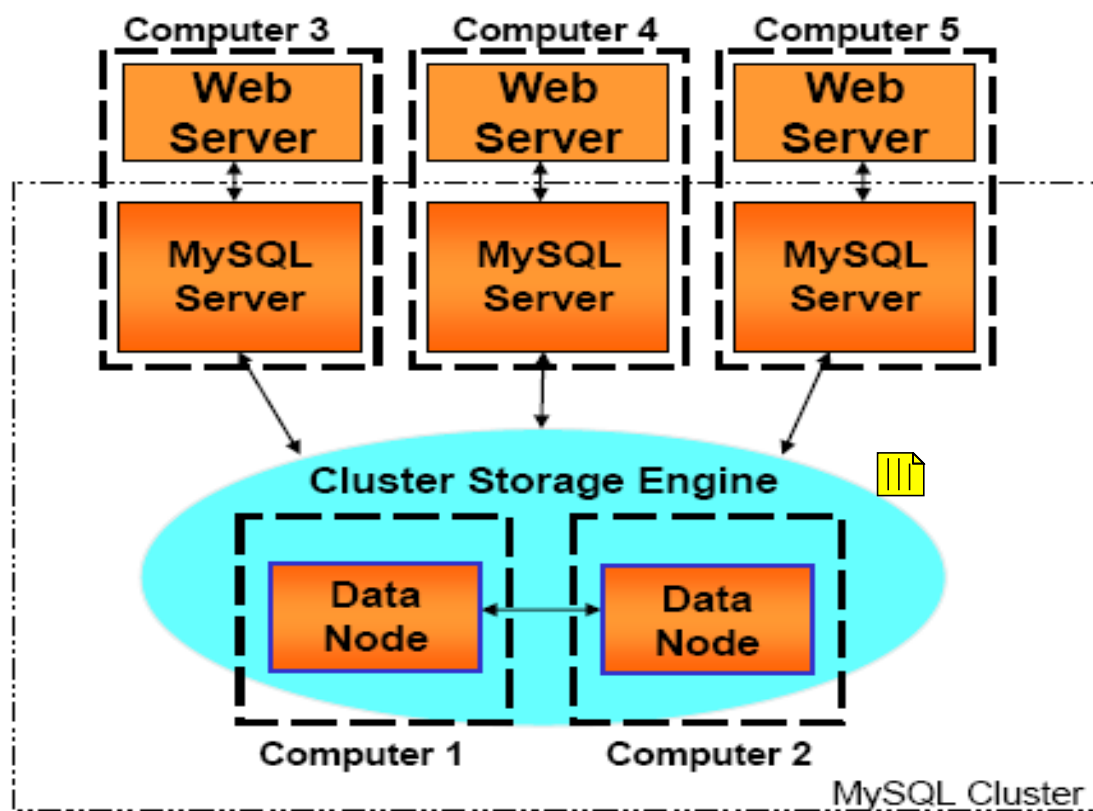
- 并行数据库
 - Shared-nothing architecture 
- 数据在节点上分散存储
 - In a RAID10 fashion
- 冗余
 - Synchronous replication 
 - Automatic fail over and recovery 
- 性能
 - Ability to mix and match disk and memory tables
 - Different access methods (SQL or native NDB API)

高可靠性解决方案: MySQL 集群





高性能配置



内 容

- MySQL调优概述
- 如何定位性能瓶颈
- 存储引擎和调优
- MySQL Cluster

谢谢大家!

欢迎访问我们的网站

<http://www.greatopensource.com>