



# MySQL最新开发动态

updated : 2019/05/25

Yitao Xu / 徐轶韬

MySQL Global Business Unit

MySQL Senior Solution Engineer

ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

# Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

**SQL & NoSQL**  
Unmatched Flexibility

**Continuous Delivery Model**  
Launchpad for Future

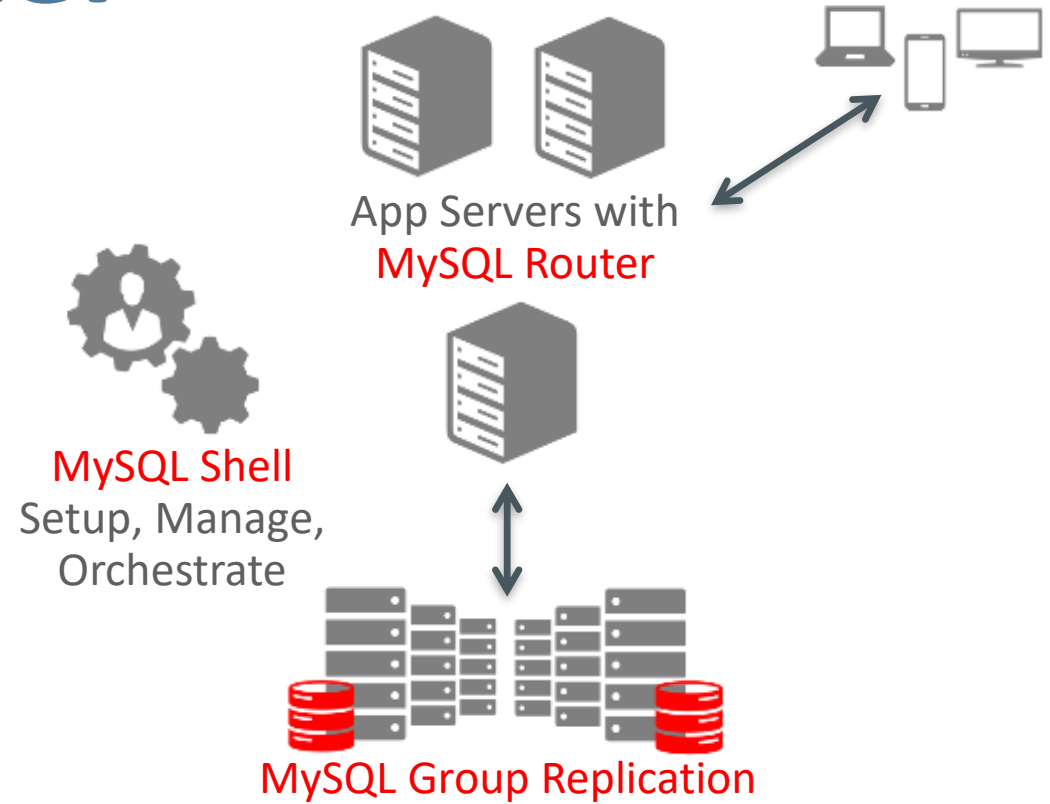
**New Architecture**  
Eliminating Legacy

**Open Source & Enterprise**  
Scalable & Robust

MySQL  
8.0

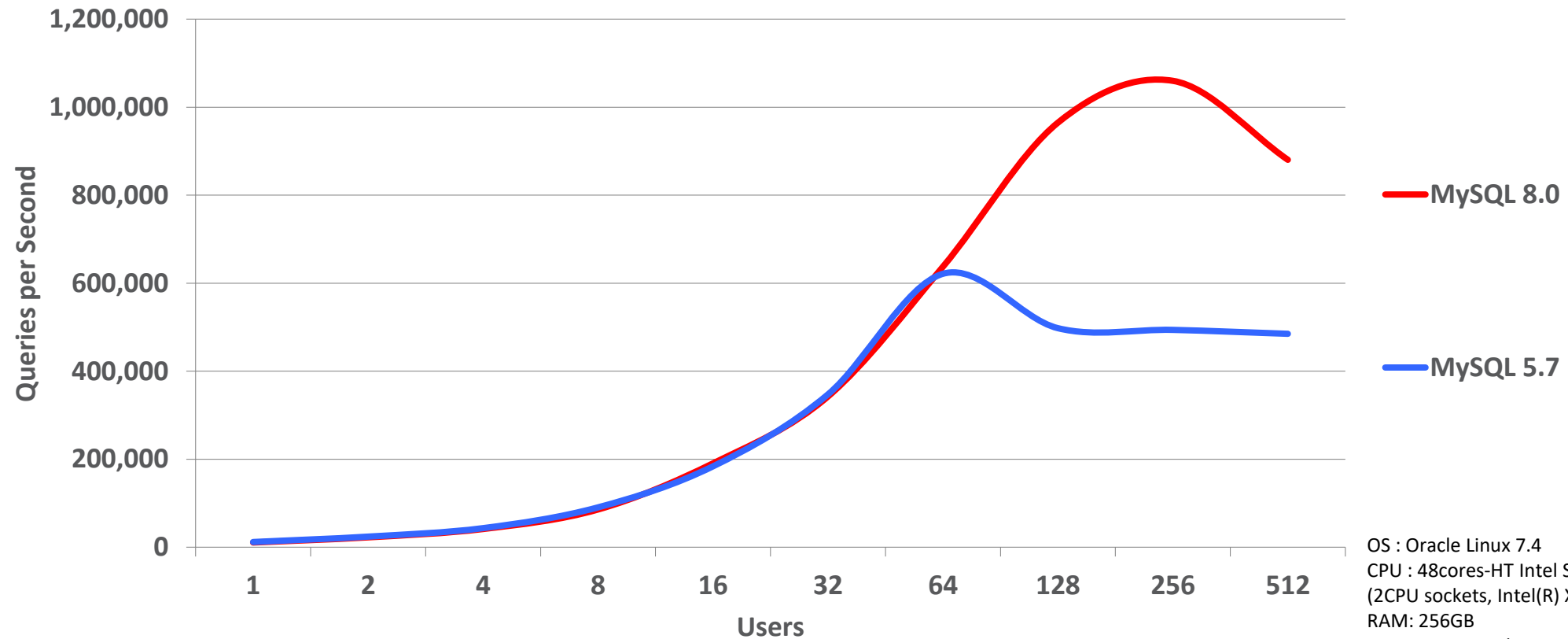
# MySQL InnoDB Cluster

“MySQL高可用性的核心功能！”



# MySQL 8.0: SysBench IO Bound 只读（使用主键查询）

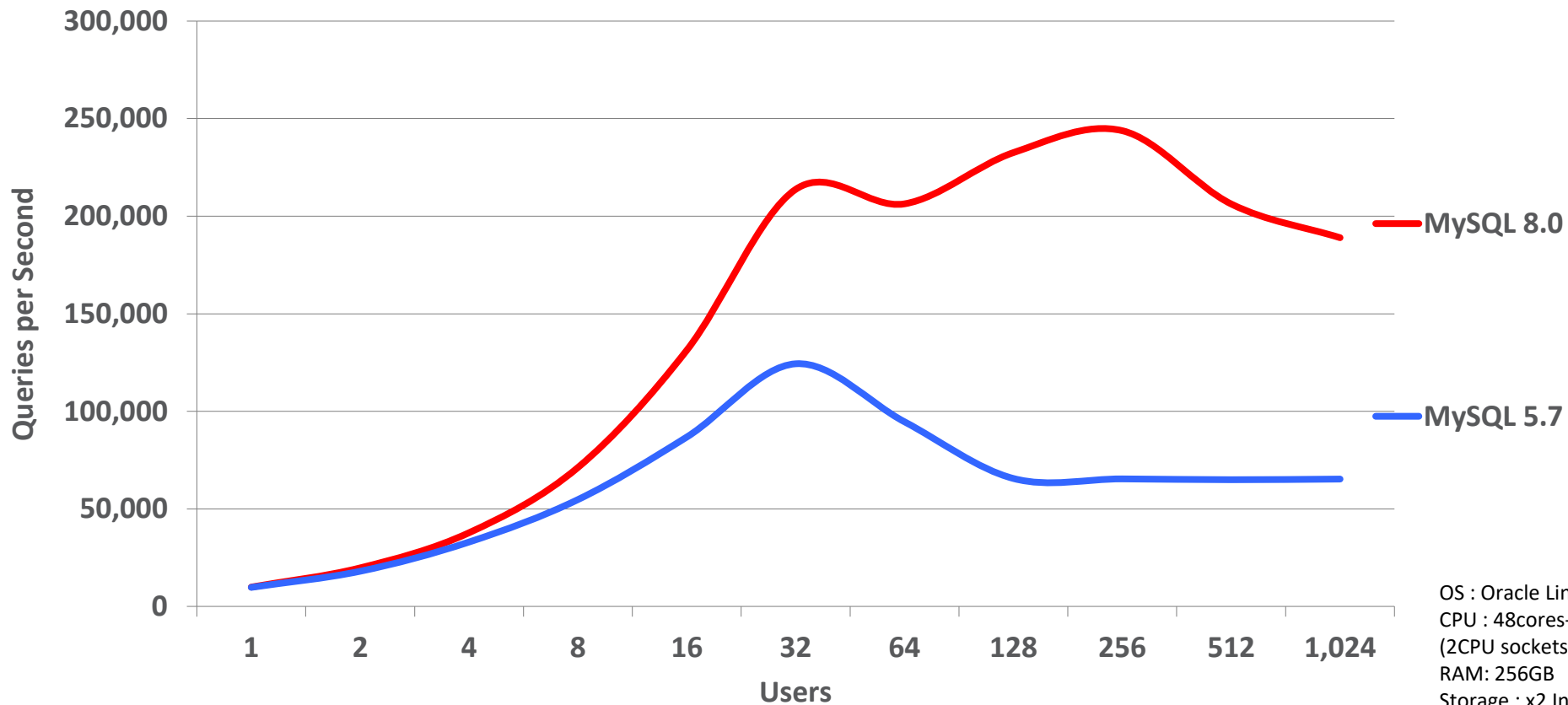
比MySQL 5.7快2倍



OS : Oracle Linux 7.4  
CPU : 48cores-HT Intel Skylake 2.7Ghz  
(2CPU sockets, Intel(R) Xeon(R) Platinum 8168 CPU)  
RAM: 256GB  
Storage : x2 Intel Optane flash devices  
(Intel (R) Optane (TM) SSD P4800X Series)

# MySQL 8.0: SysBench 读取/写入 (update不使用索引)

比MySQL 5.7快2倍



OS : Oracle Linux 7.4  
CPU : 48cores-HT Intel Skylake 2.7Ghz  
(2CPU sockets, Intel(R) Xeon(R) Platinum 8168 CPU)  
RAM: 256GB  
Storage : x2 Intel Optane flash devices  
(Intel (R) Optane (TM) SSD P4800X Series)

# MySQL 8.0 : 提高Web应用程序的开发效率



## Mobile Friendly

增强了基于位置信息提供服务的功能、并支持新的Unicode



## Developer First

混合型的数据模型及API，提高开发灵活性



## Data Driven

通过应用程序数据分析改善运营支持服务

**24x7  
at Scale**

## Scalable & Stable

改善集中访问处理、增强安全性和容错性

# MySQL 8.0 : 移动应用的亲和力



## 增强对GIS(空间图形信息)的支持

- 改善与基于位置信息的服务
- MySQL 5.7 里面增加了 Boost.Geometry 库
- MySQL 8.0 支持球面坐标和测地坐标系统(SRS)



## 默认字符集改为unicode

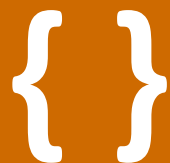
- **utf8mb4** 为默认的字符集 (包含emoji)
- 处理unicode文字列的能力提升高达16倍
- 支持Unicode 9.0
- 基于UCA(Unicode查询排序算法), 使用各种语言查询



# MySQL 8.0 : 提高应用程序的开发灵活性



数据类型



## JSON 数据类型

关系型表和非关系型数据相结合。MySQL 8.0 针对更新操作的性能进行了优化。

SQL 函数



## JSON 函数

实装了用于查询和更新 JSON 数据的各种 SQL 函数。MySQL 8.0 增加了使用 SQL 分析 JSON 数据的函数

混合API



## MySQL X DevAPI

适用于 SQL 和 CRUD 类型的 NoSQL 混合API



# MySQL 8.0 : 提高数据分析处理的效率

## 通用表表达式(CTEs)

- 替代子查询生成的表 (derived table)
- 使用WITH 语句
- 提高分析处理SQL句的可读性和处理性能，并且适用于递归数据

```
WITH tickets_filtered AS (  
  SELECT tickets.*, seats.doc  
  FROM tickets  
  INNER JOIN seats ON  
    tickets.seat_id = seats.id  
  WHERE tickets.event_id = 3  
)  
SELECT * FROM tickets_filtered  
WHERE doc->"$.section" = 201\G
```

## Window 函数

- 用于生成排名等分析用途，来自于广大用户的需求
- 将查询对象和周边数据的关联性，进行统计分析

```
SELECT name, dept_id, salary,  
  RANK() OVER w AS `rank`  
FROM employee  
  WINDOW w AS  
    (PARTITION BY dept_id  
     ORDER BY salary DESC);
```

# MySQL 8.0 : 提高应用程序的性能扩展性



## 改善集中访问

**SELECT FOR UPDATE** 语句使用  
**NOWAIT** 和 **SKIP LOCKED**  
减少锁等待

## 隐藏索引

索引对优化器隐藏。适用于2步删除和增加索引

## Performance Schema

扩展默认的性能统计信息项目，提高查询性能

## 直方图

为优化器提供没有索引列的统计信息

## 优化成本预算

优化器根据最新的存储技术，数据的缓存情况生成执行计划

## 事务调度

“Contention-Aware Transaction Scheduling”作为InnoDB的默认调度算法，大幅提升性能。

# MySQL 8.0 : 增强安全性



## 实装SQL角色

Easier to manage user and applications rights and SQL standard compliant

## 元数据更改原子性

New InnoDB based data dictionary enables ACL statements atomic and reliable

## 动态权限

Provides finer grained administrative level access controls for less use of root user

## 日志文件加密

AES 256 encryption of REDO, and UNDO Log in addition to tablespace files

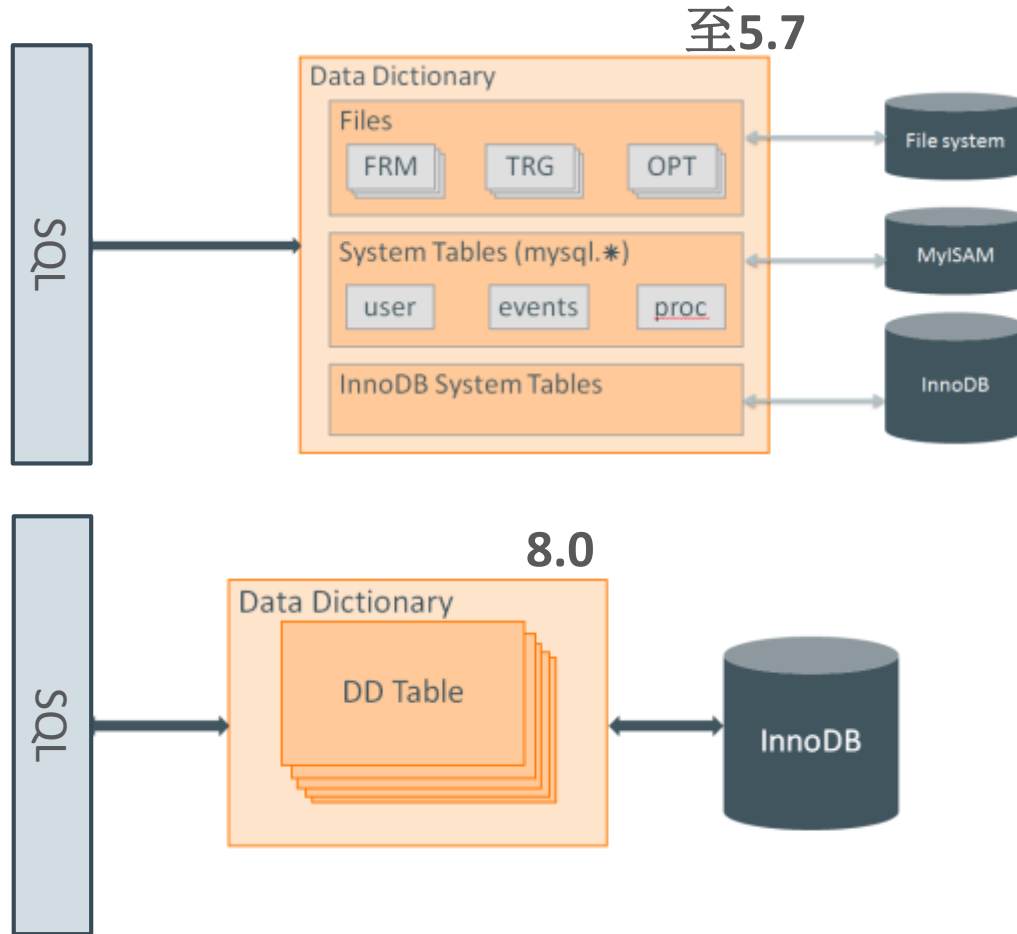
## 增强密码管理

Establish password-reuse policy with Password History, and faster with caching

## 使用OpenSSL动态链接

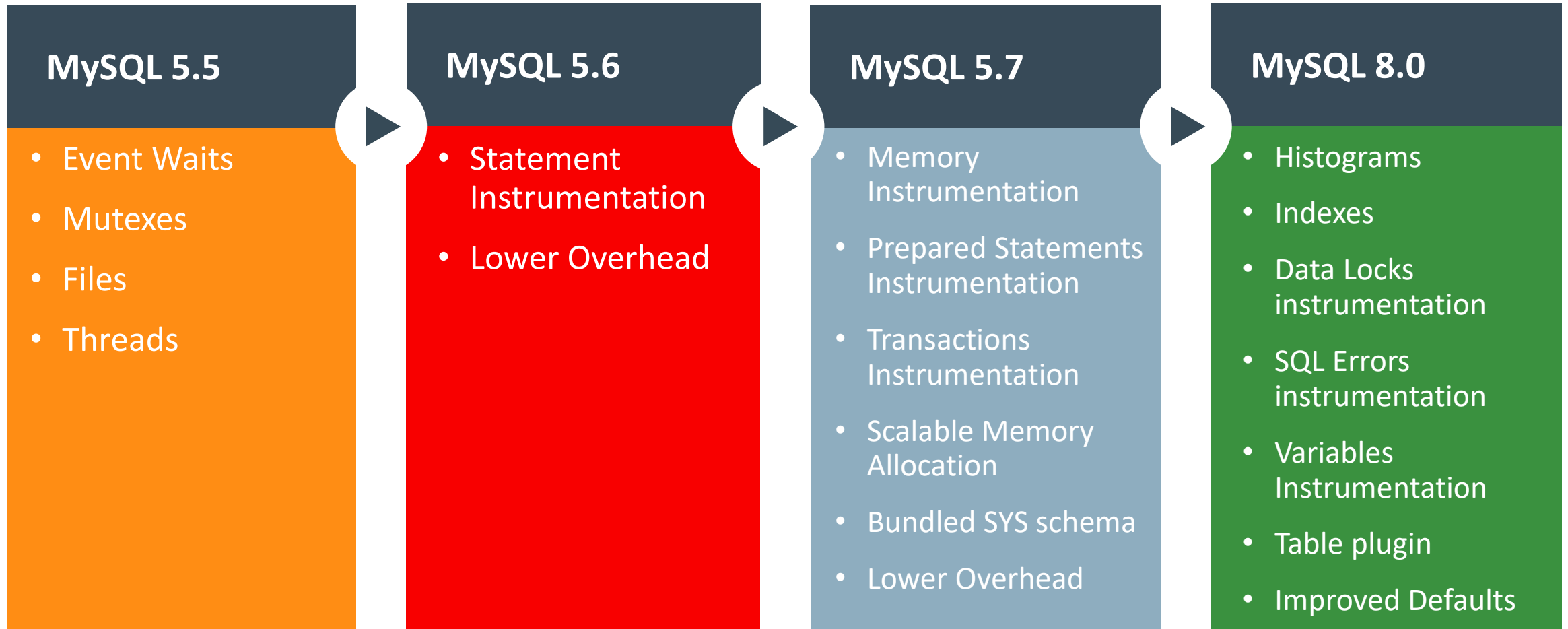
MySQL Community Edition to use OpenSSL, and all binaries are dynamically linked

# MySQL 8.0: Transactional Data Dictionary



- 元数据保存在InnoDB表内
- DDL操作原子性
- 废止使用文本文件和MyISAM表管理元数据
- 大幅改善可靠性、性能和可扩展性

# MySQL Performance Schema



# MySQL 8.0.12: Alter Table - Instant Add Column

- 来自腾讯游戏的贡献
  - Only a metadata change
  - No copying of data
  - Smaller final data size
  - Forward compatibility with old data file
- ALTER TABLE ... ADD COLUMN c, **ALGORITHM = INSTANT**
- Supports DYNAMIC/COMPACT/REDUNDANT row formats



2018-7-28  
Release

# MySQL Enterprise Data Masking

8.0.13

## De-identify, Anonymize Sensitive Data

- 数据屏蔽
  - 字符串屏蔽，字典替换
- 随机产生数据
  - 基于范围、支付卡、Email、SSN
- 满足监管要求
  - 适用于GDPR, HIPAA 和 PCI DSS
- 改进生产、开发、测试和分析环境
  - 保护机密数据

Employee Table

ID	Last	First	SSN
1111	Smith	John	555-12-5555
1112	Templeton	Richard	444-12-4444

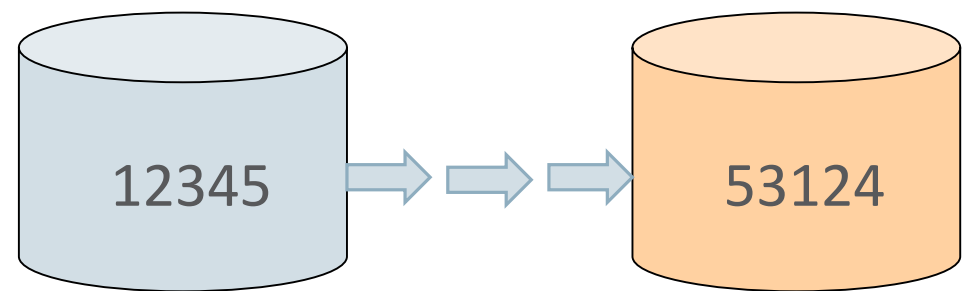
Masked View

ID	Last	First	SSN
2874	Smith	John	XXX-XX-5555
3281	Templeton	Richard	XXX-XX-4444

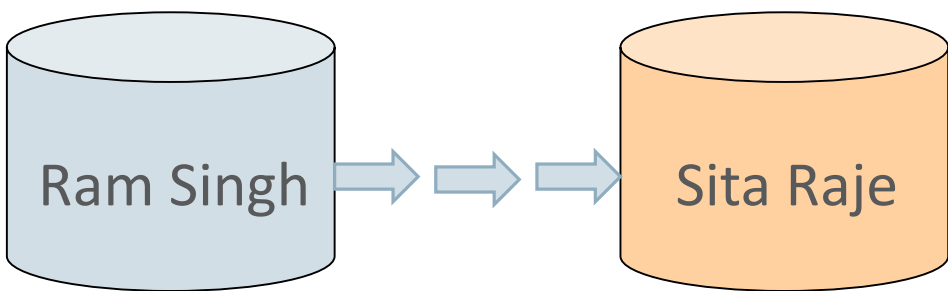


# 数据屏蔽技术

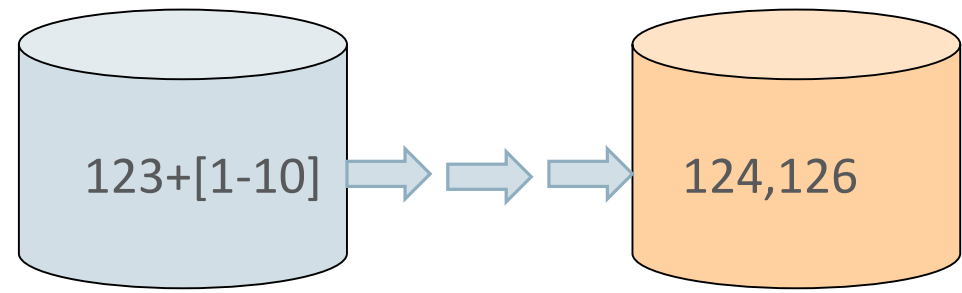
改变顺序



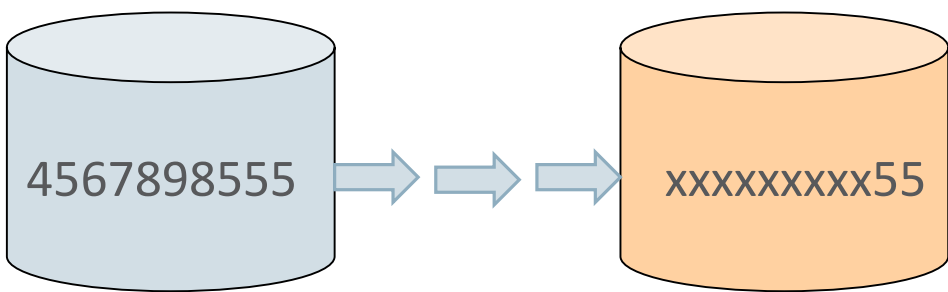
替换



随机



屏蔽

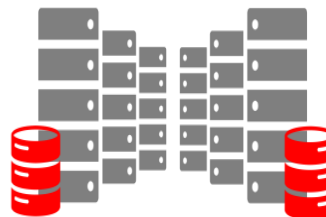


# MySQL 8.0.14 Release Note 概要



## 新功能

- LATERAL



## InnoDB

- Group Replication
- 表空间



## Security

- DB监察
- DB加密



## 管理

- 管理专用端口
- 密码管理

# MySQL 8.0.14 Release Notes (部分)

1	MySQL用户帐户可以有主密码和辅助密码	RETAIN CURRENT PASSWORD DISCARD OLD PASSWORD
2	TCP / IP端口可以设置为管理连接专用	admin_address admin_port
3	使用OVER语句时, JSON_ARRAYAGG() / JSON_OBJECTAGG()可以作为window函数	
4	慢日志结果里可以适用执行开始时间	log_slow_extra
5	对日志文件进行加密 (bin-log relay-log)	binlog_encryption
6	LDAP认证改为LDAP->LDAPS (LDAP over SSL)(Port No. 636 or 3269)	authentication_ldap_sasl_server_port authentication_ldap_simple_server_port
7	指定ST_Distance()函数返回值单位	INFORMATION_SCHEMA.ST_UNITS_OF_M EASURE
8	派生表可以使用关键字LATERAL, 允许在同一FROM语句里使用之前的表	

# MySQL 8.0.14 Release Notes（部分）

9	X Plugin包含错误处理类5位SQLSTATE错误代码	Bug #28735058
10	innodb_buffer_pool_in_core_file设置为无效，可以消减核心文件的大小（禁止写入核心文件）	innodb_buffer_pool_in_core_file
11	CREATE UNDO TABLESPACE可以指定UNDO表空间创建位置	innodb_undo_tablespaces
12	支持并行读取聚簇索引（InnoDB）	innodb_parallel_read_threads
13	CREATE TABLESPACE中的ADD DATAFILE子句更改为选项	
14	innodb_dedicated_server自动设置日志文件	innodb_buffer_pool_size innodb_log_file_size innodb_flush_method
15	Group Replication使用的通信系统(GCS)及通信引擎(XCom、Paxos的一种)支持IPv6	
16	MySQL Group Replication可以使用专门的输入通道来替代TCP进行通信	

# MySQL 8.0.14 Release Notes (部分)

---

17	添加了两个新的会话系统变量供复制内部使用	<code>original_server_version</code> <code>immediate_server_version</code>
18	更改了MySQL 5.7到MySQL 8.0转储导入的错误处理	<code>ER_WRONG_VALUE_FOR_VAR</code>
19	审核日志里面追加了独自的消息	<code>audit_api_message_emit_udf()</code>
20	CTE的外部可以参照派生表	
21	在群组复制中，增加了对主机切换的控制	<code>group_replication_consistency</code>
	其他，bugfix....	

---

## 2 设置管理连接专用的TCP/IP端口

- 面向管理者的专用TCP/IP端口
- 不受max\_connections限制
- 管理连接数量没有限制
- admin\_port (default : 33062 )
- 为admin\_address设置IP地址使其有效
- create\_admin\_listener\_thread 可以配置使用专用线程



### 3 JSON\_ARRAYAGG()

- select \* from employee;

emp_no	first_name	last_name	hire_date	dept_id
1	Samanyu	Kishore	2010-10-10	1
2	Nethra	Deepak	2003-08-08	1
3	Sushma	Sunil	2008-01-02	1
4	Kishore	Krishna	1999-06-01	2
5	Muni	Raju	2000-10-01	2
6	Keshav	Basappa	1999-08-01	2
7	Bhagath	Reddy	2000-01-01	2

- select dept\_id, hire\_date, json\_arrayagg(first\_name) over (partition by dept\_id order by hire\_date) as list from employee;

dept_id	hire_date	list
1	2003-08-08	["Nethra"]
1	2008-01-02	["Nethra", "Sushma"]
1	2010-10-10	["Nethra", "Sushma", "Samanyu"]
2	1999-06-01	["Kishore"]
2	1999-08-01	["Kishore", "Keshav"]
2	2000-01-01	["Kishore", "Keshav", "Bhagath"]
2	2000-10-01	["Kishore", "Keshav", "Bhagath", "Muni"]

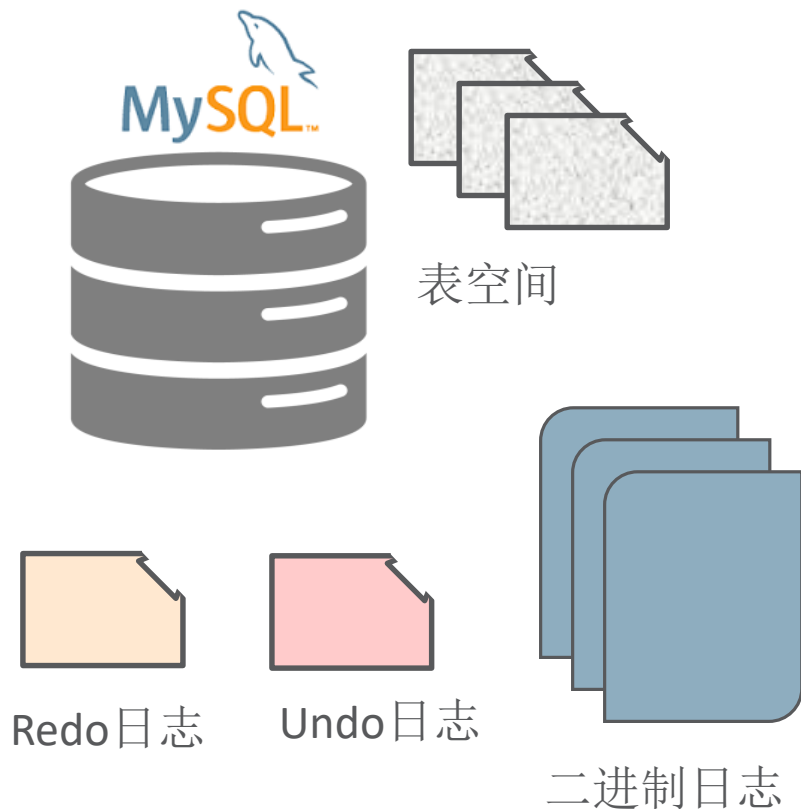
### 3 JSON\_OBJECTAGG()

- select dept\_id, json\_objectagg(first\_name, hire\_date) over (partition by dept\_id order by hire\_date) as line from employee;

dept_id	line
1	{"Nethra": "2003-08-08"}
1	{"Nethra": "2003-08-08", "Sushma": "2008-01-02"}
1	{"Nethra": "2003-08-08", "Sushma": "2008-01-02", "Samanyu": "2010-10-10"}
2	{"Kishore": "1999-06-01"}
2	{"Keshav": "1999-08-01", "Kishore": "1999-06-01"}
2	{"Keshav": "1999-08-01", "Bhagath": "2000-01-01", "Kishore": "1999-06-01"}
2	{"Muni": "2000-10-01", "Keshav": "1999-08-01", "Bhagath": "2000-01-01", "Kishore": "1999-06-01"}



## 5 二进制日志和中继日志加密

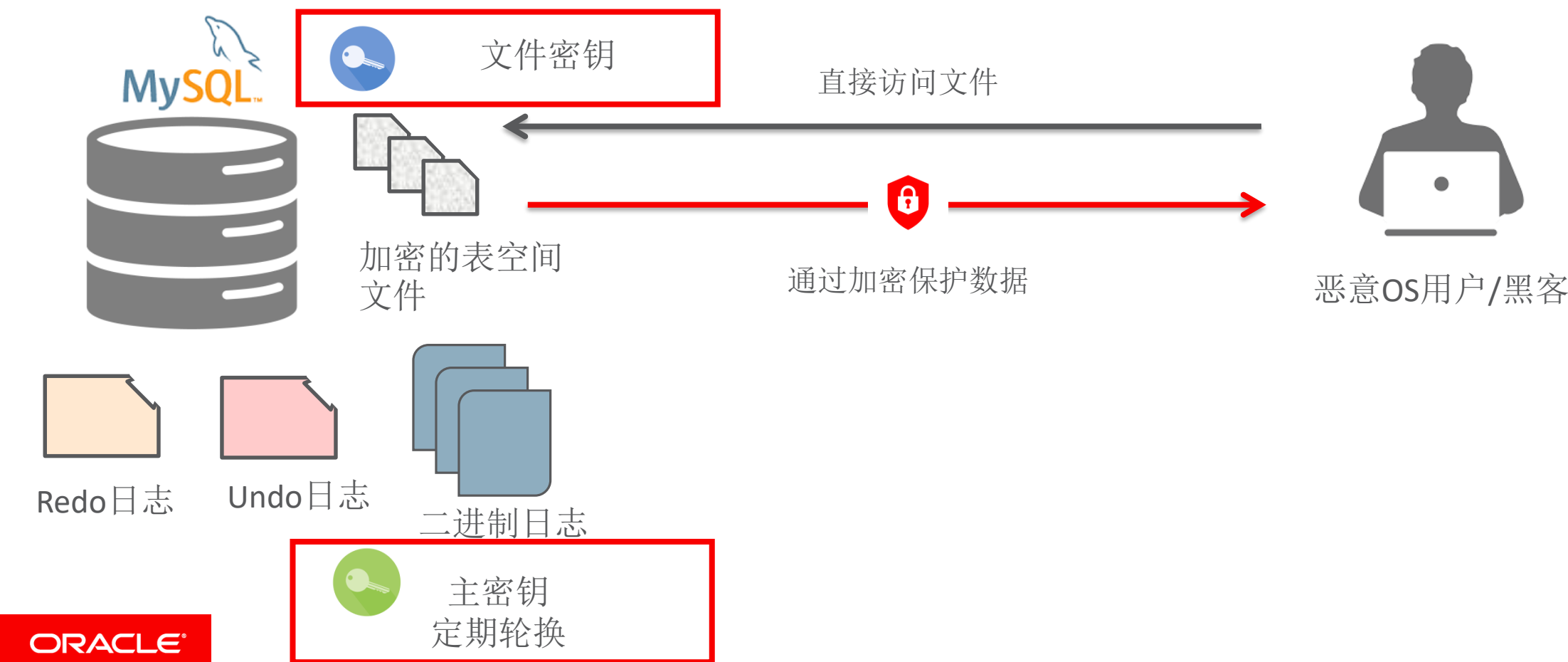


- 表空间
  - `ALTER TABLE xxxx ENCRYPTION='Y'`
- Redo日志/ Undo日志
  - `innodb_redo_log_encrypt=ON`
  - `innodb_undo_log_encrypt=ON`
- 二进制日志
  - `binlog_encryption=ON`

### 使用时需注意!!

使用keyring\_file及keyring\_encrypted插件不能满足某些合规要求。。需要额外对KEY进行管理。

# MySQL Enterprise Transparent Data Encryption



## 7 ST\_Distance()函数

- ST\_Distance(g1, g2 [, unit])
- INFORMATION\_SCHEMA.ST\_UNITS\_OF\_MEASURE

UNIT_NAME	UNIT_TYPE	CONVERSION_FACTOR	DESCRIPTION
metre	LINEAR	1	
millimetre	LINEAR	0.001	
centimetre	LINEAR	0.01	
German legal metre	LINEAR	1.0000135965	
foot	LINEAR	0.3048	
;	:	:	
Indian yard	LINEAR	0.9143985307444408	
kilometre	LINEAR	1000	
Clarke's chain	LINEAR	20.1166195164	
Clarke's link	LINEAR	0.201166195164	
British yard (Benoit 1895 A)	LINEAR	0.9143992	
British yard (Sears 1922)	LINEAR	0.9143984146160288	
British foot (Sears 1922)	LINEAR	0.3047994715386762	
Gold Coast foot	LINEAR	0.3047997101815088	
British chain (Sears 1922)	LINEAR	20.116765121552632	
yard	LINEAR	0.9144	
;	:	:	

## 8 LATERAL

- 实现From语句的相关子查询

例) <https://mysqlserverteam.com/support-for-lateral-derived-tables-added-to-mysql-8-0-14/>

- 随机生成 From / To
  - insert into edges(from\_id, to\_id)select origin\_nodes.id, target\_nodes.id from nodes as origin\_nodes,( **select id from nodes order by rand() limit 2**) as target\_nodes;
  - 适用于全部的随机结果
- order by rand()+0\***origin\_nodes.id** 利用前表的id作为随机系数

# 8 LATERAL

```
explain
select origin_nodes.id, target_nodes.id
from nodes as origin_nodes,
(
  select id from nodes order by rand() limit 2
) as target_nodes;
```

▶	1	PRIMARY	<derived2>	NULL	ALL	NULL	NULL	NULL	NULL	2	100.00	NULL
	1	PRIMARY	origin_nodes	NULL	ALL	NULL	NULL	NULL	NULL	20	100.00	Using join buffer (Block Nested Loop)
	2	DERIVED	nodes	NULL	ALL	NULL	NULL	NULL	NULL	20	100.00	Using temporary; Using filesort

```
explain
select origin_nodes.id, target_nodes.id
from nodes as origin_nodes,
  LATERAL (
    select id
    from nodes
    order by rand()+0*origin_nodes.id
    limit 2
  ) as target_nodes;
```

▶	1	PRIMARY	origin_nodes	NULL	ALL	NULL	NULL	NULL	NULL	20	100.00	Rematerialize (<derived2>)
	1	PRIMARY	<derived2>	NULL	ALL	NULL	NULL	NULL	NULL	2	100.00	NULL
	2	DEPENDENT DERIVED	nodes	NULL	ALL	NULL	NULL	NULL	NULL	20	100.00	Using temporary; Using filesort



# 18 MySQL 8.0转储导入时的错误处理

- ER\_WRONG\_VALUE\_FOR\_VAR

- 由SQL 模式不同引起

SQL Mode	Version 5.7	Version 8.0
NO_AUTO_CREATE_USER	有效	不支持

- 通过pseudo\_slave\_mode系统变量控制

系统变量	true	false
pseudo_slave_mode	忽略模式，输出警告	ER_UNSUPPORTED_SQL_MODE

## 21 group\_replication\_consistency

- 控制保证群组事务一致性
  - EVENTUAL (default)
  - BEFORE\_ON\_PRIMARY\_FAILOVER
    - 客户端处理暂停，直至全部的积压待处理事务完成:保持一致性
  - BEFORE
    - 等待RW事务之前的事务完成
    - 等待RO事务之前的事务完成
  - AFTER
    - RW事务等待更改应用于所有其他成员
  - BEFORE\_AND\_AFTER
    - 适用于 BEFORE 和 AFTER

<https://dev.mysql.com/doc/refman/8.0/en/group-replication-options.html>

# MySQL 8.0.16 Release Notes (部分)

1	MySQL用户帐户类别，根据他们是否具有SYSTEM_USER权限进行区分系统和普通用户	SYSTEM_USER
2	utf8mb4_zh_0900_as_cs适用于 utf8mb4的Unicode字符集。MySQL中第一个可用于Unicode的中文语言排序规则	utf8mb4_zh_0900_as_cs
3	将mysql_upgrade功能集成于MySQL服务器	mysql_upgrade
4	地理空间数据，ST_Length()函数增加可选的第二个参数，能够指定返回值的单位。	ST_Length()
5	群组复制通信的消息分段处理	group_replication_communication_max_message_size
6	检查群组现有成员公布的通信协议版本。UDF	group_replication_get_communication_protocol()
7	为群组成员定义自动重新加入	group_replication_autorejoin_tries



# 1.SYSTEM\_USER权限

- 可以保护用户被其他具有CREATE USER权限的用户修改。
- SYSTEM\_USER权限的约定：
  - 如果要修改具有SYSTEM\_USER权限的用户，除了修改用户所需的权限之外，还需要具有SYSTEM\_USER权限。
  - 具有SYSTEM\_USER和CREATE USER权限的用户可以修改具有SYSTEM\_USER权限的用户。

## 2. 字符集utf8mb4\_zh\_0900\_as\_cs

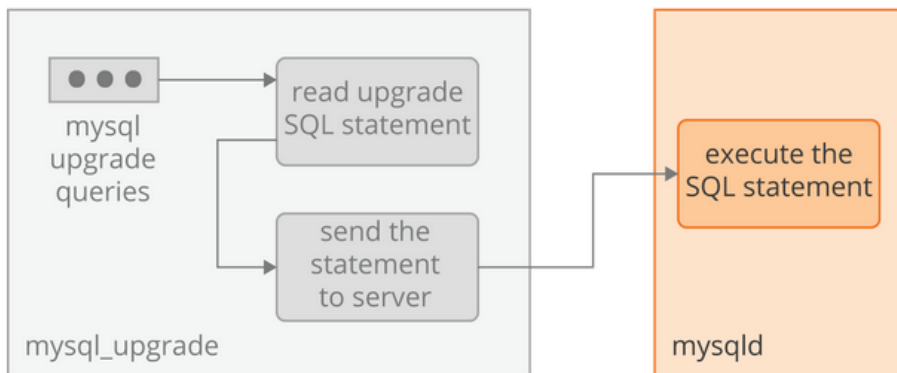
- **New!** Chinese Collation
- | utf8mb4 | utf8mb4\_zh\_0900\_as\_cs |
- | gb18030 | gb18030\_chinese\_ci |
- | gb2312 | gb2312\_chinese\_ci |
- | gbk | gbk\_chinese\_ci |



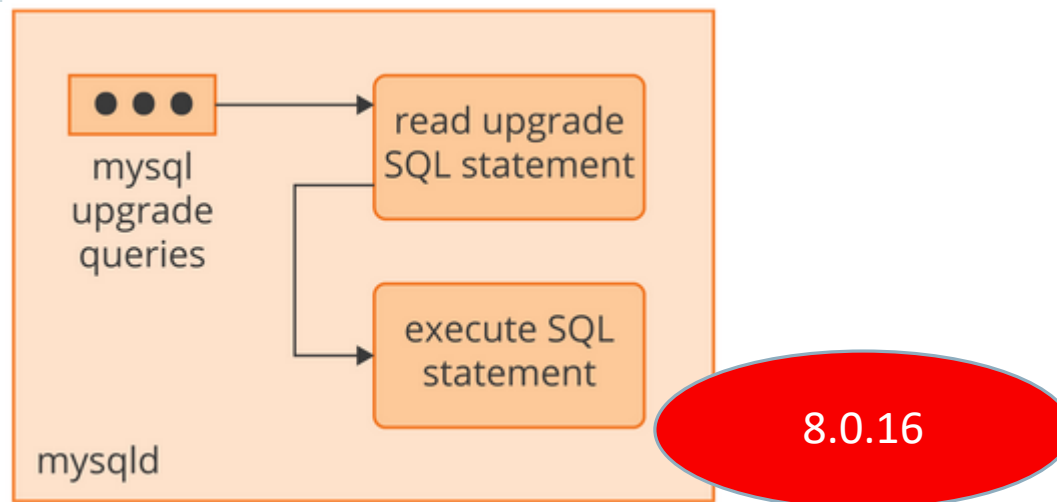
妈  
麻  
马  
骂

# 3.mysql\_upgrade功能集成于MySQL服务器

- 1.关闭服务器并安装新的二进制文件
- 2.启动服务器并让服务器升级数据字典表
- 3.运行mysql\_upgrade客户端以更新某些系统表，可能也会更新用户表
- 4.从Web下载帮助表的更新脚本
- 5.加载新的帮助表
- 6.重新启动服务器



- 1.关闭服务器并安装新的二进制文件
- 2.启动服务器并让服务器升级数据字典表和系统表。如果需要升级用户表，我们现在可以通过SQL来分析是否需要升级用户表。帮助表也会更新。



## 5. 群组消息片段化

- 将消息分段引入群组通信框架
  - Group Replication的Paxos引擎依赖于一个单线程消息传递内核（XCom）
  - 在某些情况下，线程可能忙于处理过大的消息，并且延迟参与故障检测相关任务，这可能导致故障检测器中的误报。
- `group_replication_communication_max_message_size`

## 7.为群组成员定义自动重新加入

- 当检测到集群成员无法正常工作时，Group Replication的故障检测机制会将其从集群中驱逐。
  - 某些场景，诸如慢速或脆弱的网络，瞬态故障率很高的网络中，产生不必要的成员驱逐。
- autorejoinRetries选项扩展了以下命令，用以定义重新加入群组次数：
  - dba.createCluster()
  - <Cluster>.addInstance()
  - <Cluster>.setOption()
  - <Cluster>.setInstanceOption()

# MySQL 升级的重点



# Oracle Lifetime Support for MySQL

支持概要		Premier (1-5年)	Extended (6-8年)	Sustain (9年以后)
24小时365日支持		•	•	•
支持事件数量无限制		•	•	•
知识库		•	•	•
版本维护、修改错误、提供补丁升级		•	•	仅限现存版本
MySQL 顾问支持		•	•	•

MySQL Version	GA(YYYY-MM)	Premier	Extended	Sustain
5.1	2008-12	x	x	○
5.5	2010-12	x	x	○
5.6	2013-02	x	2021-02	○
5.7	2015-10	2020-10(预定)	2023-10(预定)	○
8.0	2018-04	2023-04(预定)	2026-04(预定)	○



# 升级到MySQL 8.0



# MySQL 8.0: 升级检查程序

- 非常简单的MySQL Shell工具
  - JavaScript
  - Python
- 基于重要程度确定问题点
  - 无问题
  - 潜在的错误
  - 升级前必须需要修改的错误
- 推荐修改
  - Schema、配置
  - 服务器上的数据，等等

```
-js> util.checkForServerUpgrade("root@localhost:3306")
MySQL server at localhost:3306 will now be checked for compatibility issues
MySQL version: 5.7.19 - MySQL Community Server (GPL)

Page of db objects with names conflicting with reserved keywords in 8.0
Issues found

Page of utf8mb3 charset
Warning: The following objects use the utf8mb3 character set. It is recommended
to convert them to use utf8mb4 instead, for improved Unicode support.

e_schema.city.name - column's default character set: utf8
e_schema.city.country_code - column's default character set: utf8

Page of use ZEROFILL/display length type attributes
Warning: The following table columns specify a ZEROFILL/display length attribute.
Please be aware that they will be ignored in MySQL 8.0

big_table.ORDINAL_POSITION - bigint(21) unsigned
```

# 补充说明

- 在线手册的“2.11.1.1 MySQL Upgrade Strategies”章节内记载了升级时需要注意的问题及准备[Verifying Upgrade Prerequisites for Your MySQL 5.7 Installation](#)
- MySQL Shell 的升级检查程序包括了上面的全部步骤
  - [MySQL Shell 8.0.4: Introducing “Upgrade checker” utility](#)

# MySQL Cloud Service



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.



# MySQL as a Service



- **MySQL Enterprise Edition 和 Oracle Cloud Infrastructure 的强力结合**
  - 高性能的软硬件结合
- **自动管理&简单使用**
  - DBA的日常工作自动化(备份、打补丁、升级、配置复制等等)。
  - Web控制台、REST API、CLI、SDK、无缝整合
- **弹性&高可用性**
  - 不同物理地点间的复制
- **安全&合规**
  - 以最严要求的企业级规定为基准，开箱即用的安全选项

# Managed MySQL

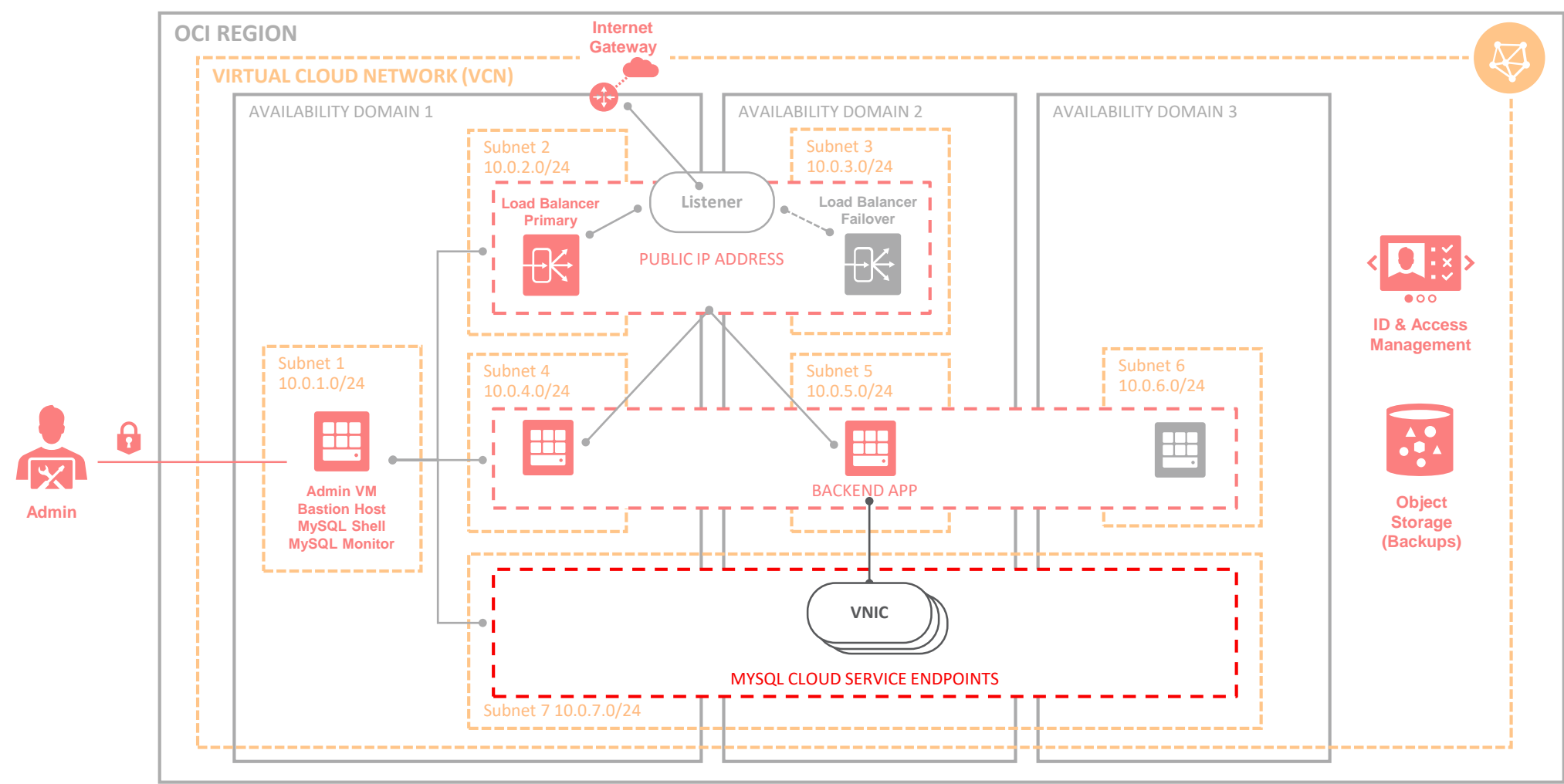
## Oracle

- 备份&恢复
- 自动故障转移实现高可用性
- 垂直及水平扩展
- 应用数据库和OS的补丁
- 监视及日志管理
- MySQL Enterprise Edition提供的安全性选项
- Oracle Premier Support

## You

- 设计schema
- SQL开发和优化
- 定义数据访问和保存的制度

# MySQL Cloud Service高可用性部署



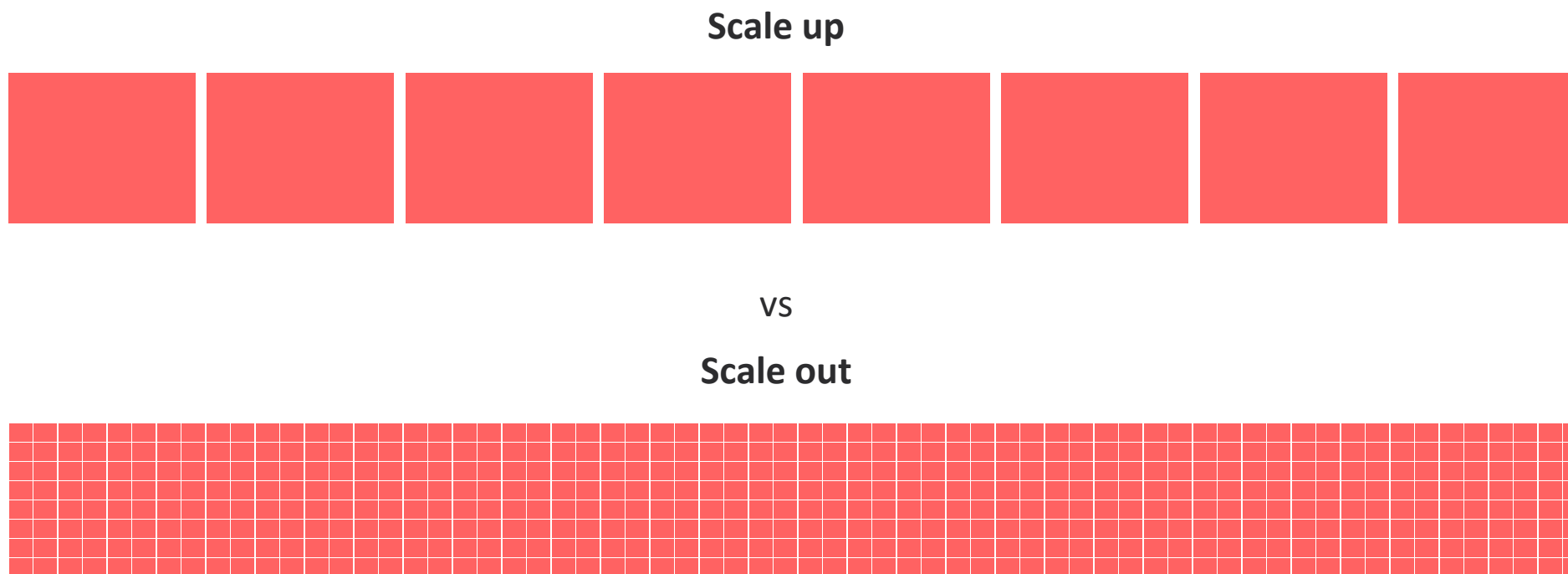
# MySQL Analytics



- Oracle Labs开发的突破性技术
- 在内存里运行分析，提供极限性能
- 与MySQL整合
  - 将新的服务实例化，或者追加到运行中的MySQL服务
- 用于实时性高的SQL分析
  - 分析在同一个事务存储中提供
- 基于Oracle Cloud Infrastructure设计
  - 设计具有出色的弹性
- 支持本地和云上的数据

# 面向分析的 In-Memory 架构

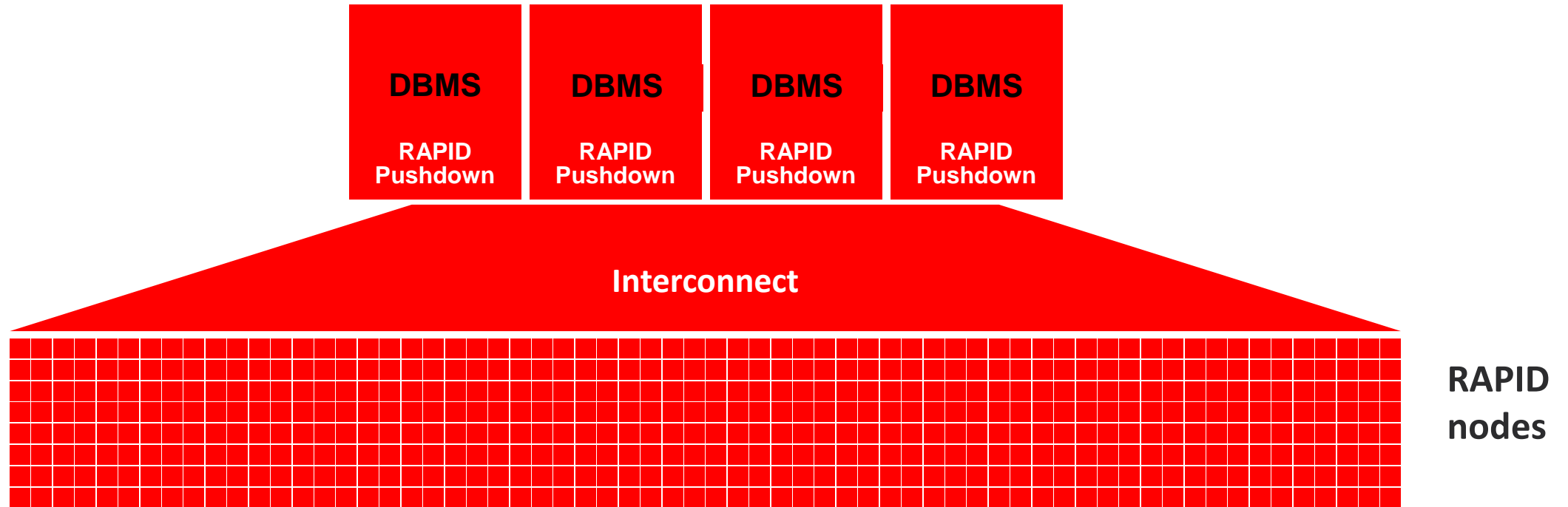
## Architectures for In-Memory Analytics



**What is the right architecture for 1+ TB analytic workloads for best Perf/watt?**

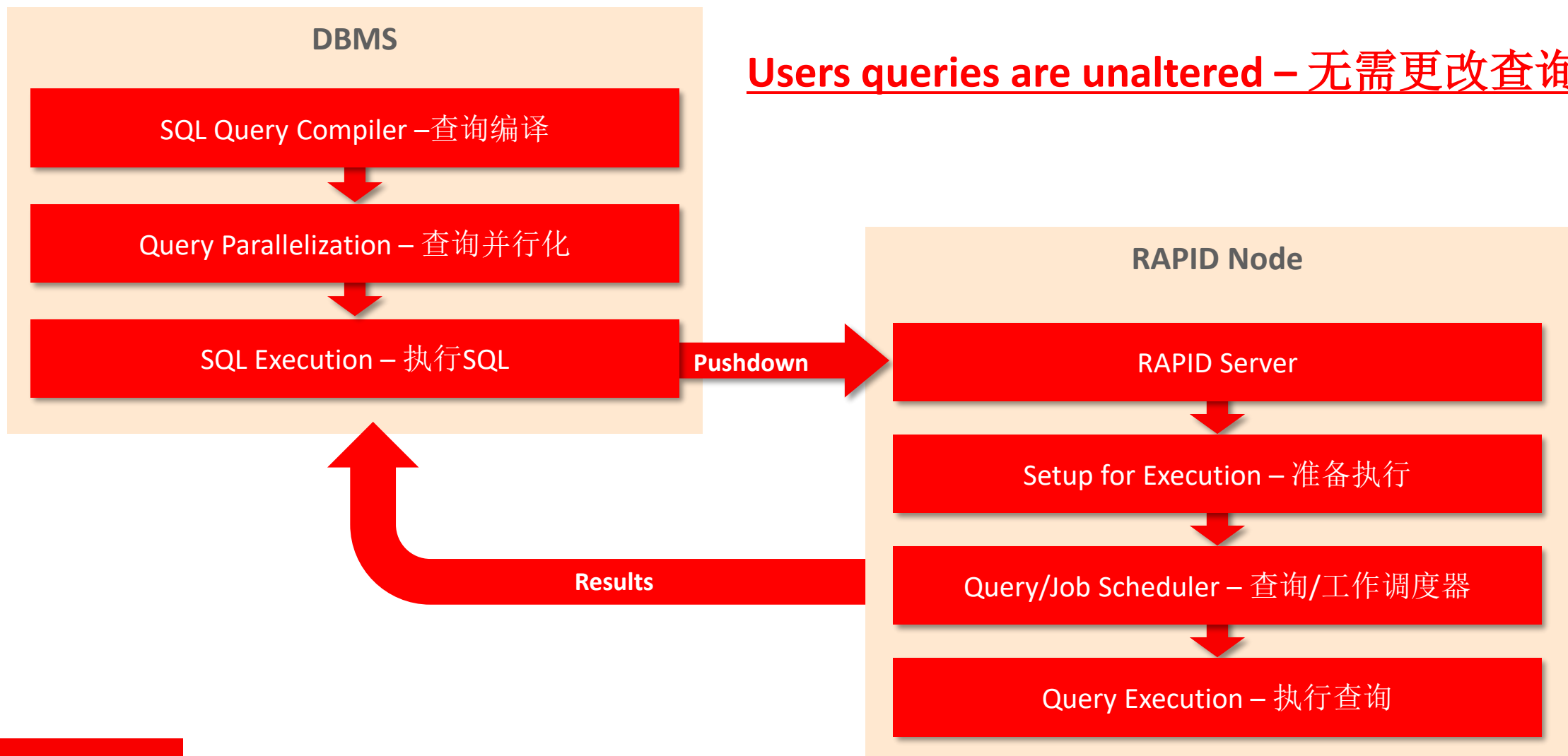


# Our Answer : **R**APID **A**nalytics **P**rocessing in **D**RAM



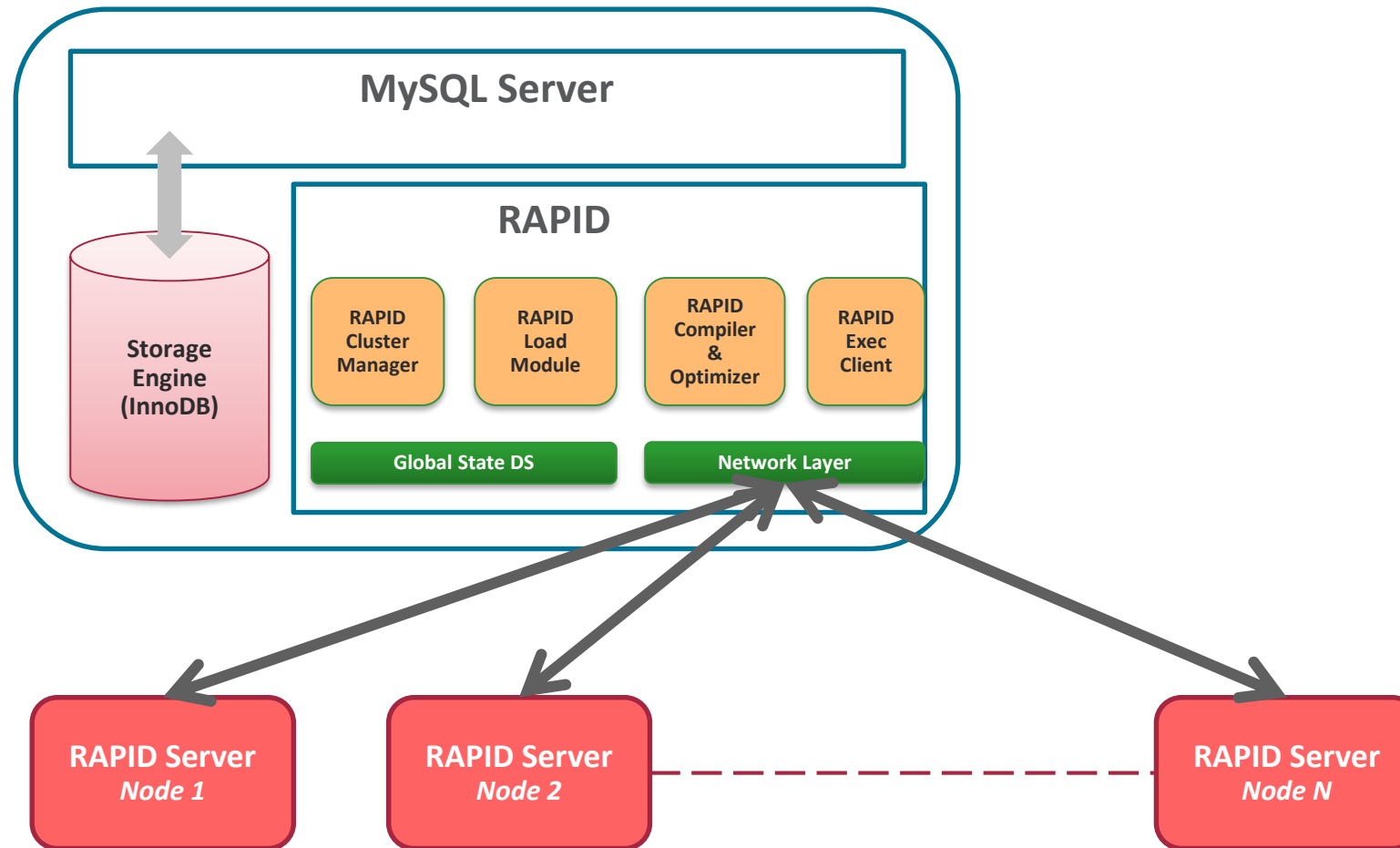
- Push down as much of the query as possible  
尽可能的将查询下推
- Extreme performance for analytic operators on a single node  
单一节点上分析性能最大化
- Extreme (near linear) scalability across nodes  
跨节点扩展性能最大化

# Query Processing Architecture – 实行查询的架构



# RAPID MySQL Integration Architecture

## - RAPID和MySQL整合的架构



ORACLE®

甲骨文