



# Oracle技术嘉年华

Oracle Technology Carnival 2015

稳健•高效•云端 - 数据技术最佳实践

讲师：杨尚刚

主题：MySQL运维优化实践和发展



## 个人简介

- 2011年加入新浪
- 负责新浪微博核心数据库架构设计
- 负责新浪数据库平台底层软硬件平台优化
- 2015年加入美图网，负责后端数据存储平台建设
- 理念：设计简洁的架构



# 目录

- 运维系统建设
- 性能优化
- 未来发展

# MySQL当前存在问题

- 优化器对复杂SQL支持不好
- 对SQL标准支持不好
- 大规模集群方案不成熟，主要指中间件
- ID生成器
- 逻辑复制
- Online DDL
- HA方案不完善
- 备份和恢复方案还是比较复杂，需要依赖外部组件
- 展现给用户信息过少
- 众多分支

# DBA Life

- 满足各种各样的开发需求
- 各式各样的Schema审核
- SQL优化
- 各种救火和处理报警 :主库故障，缓存“雪崩”
- 各种业务和项目上线
- 业务沟通



# 解放DBA的双手

- 制定运维规范
- 流程自动化
- 减少沟通成本



# 运维体系建设

- 规范化
- 自动化
- 平台化

# 规范化

- 部署规范
- 软件规范
- 业务开发规范
- 日常运维规范



# 软件规范

- 操作系统
- MySQL版本
- 相关工具版本



我的建议

MySQL社区版 > Percona Server  
> MariaDB > MySQL 企业版



# 业务开发规范

## ➤数据库开发规范：

开发规范是针对内部开发的一系列建议或规则

由DBA制定，如果有DBA的话

开发规范也包含几部分：基本命名和约束规范，字段设计规范，索引规范，使用规范

## ➤意义：

保证线上数据库schema规范

减少出问题概率

方便自动化管理

需要长期坚持，是一个双赢的事情

# 开发规范示例

- 表字符集选择UTF8 ，如果需要存储emoj表情，需要使用UTF8mb4(MySQL 5.5.3以后支持)
- 存储引擎使用InnoDB
- 变长字符串尽量使用varchar 和varbinary
- 不在数据库中存储图片、文件等
- 每张表数据量控制在1亿以下
- 所有字段均定义为NOT NULL ，除非你就想存Null
- 字段类型 在满足需求条件下，越小越好
- 使用varchar存储变长字符串 ，当然要注意varchar(M)里的M指的是字符数不是字节数

# 元数据存储

- 端口
- 产品线
- 备份
- 监控配置
- 服务器
- 集群拓扑

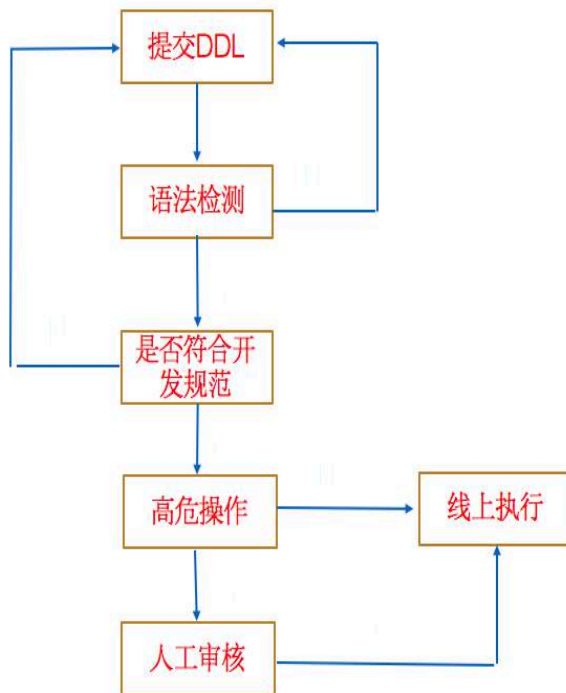
元数据信息的可靠性和完整性是自动化的基础之一

# 运维自动化

- DDL自动化
- 备份系统
- 慢日志系统

# DDL自动化

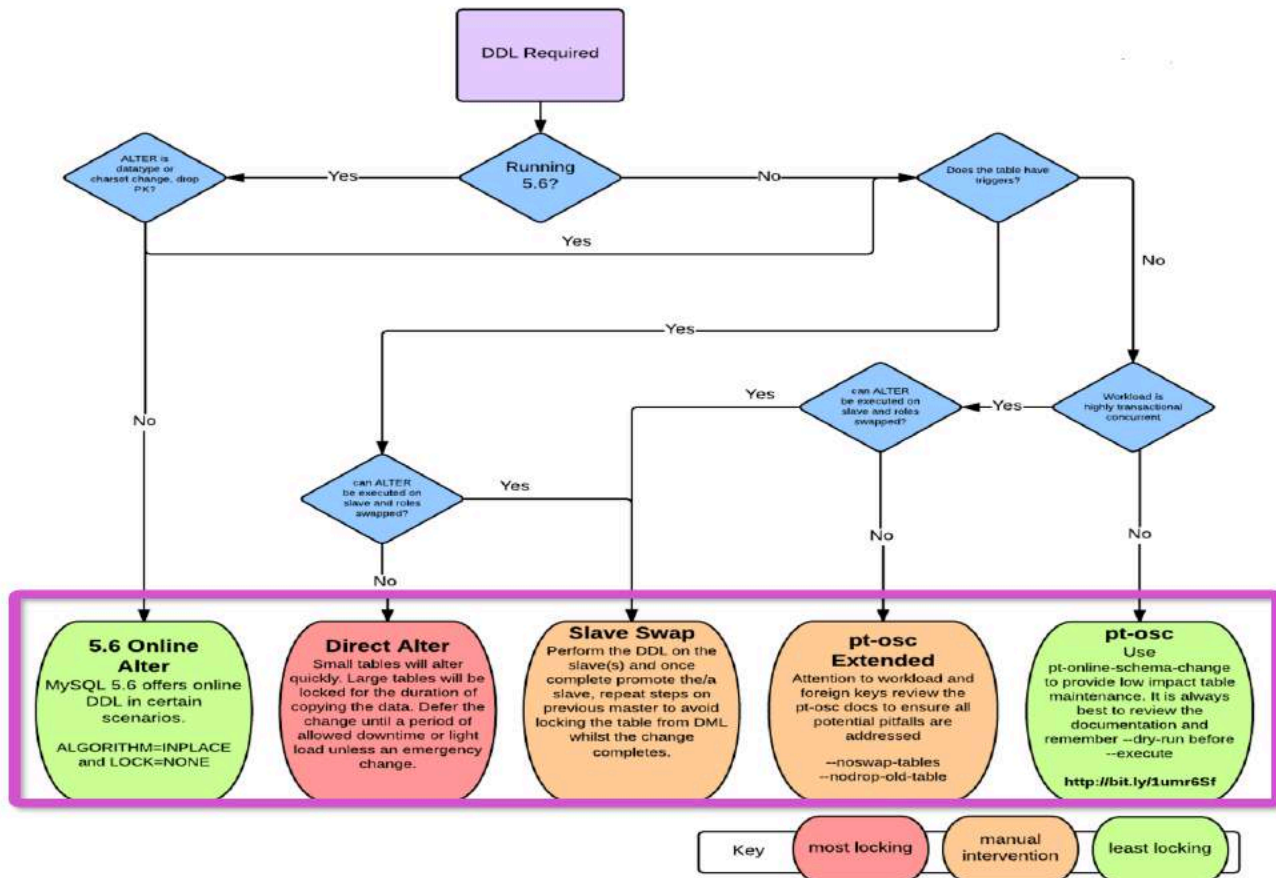
- 建表自动化
- 审表自动化
- 改表自动化



## Online DDL

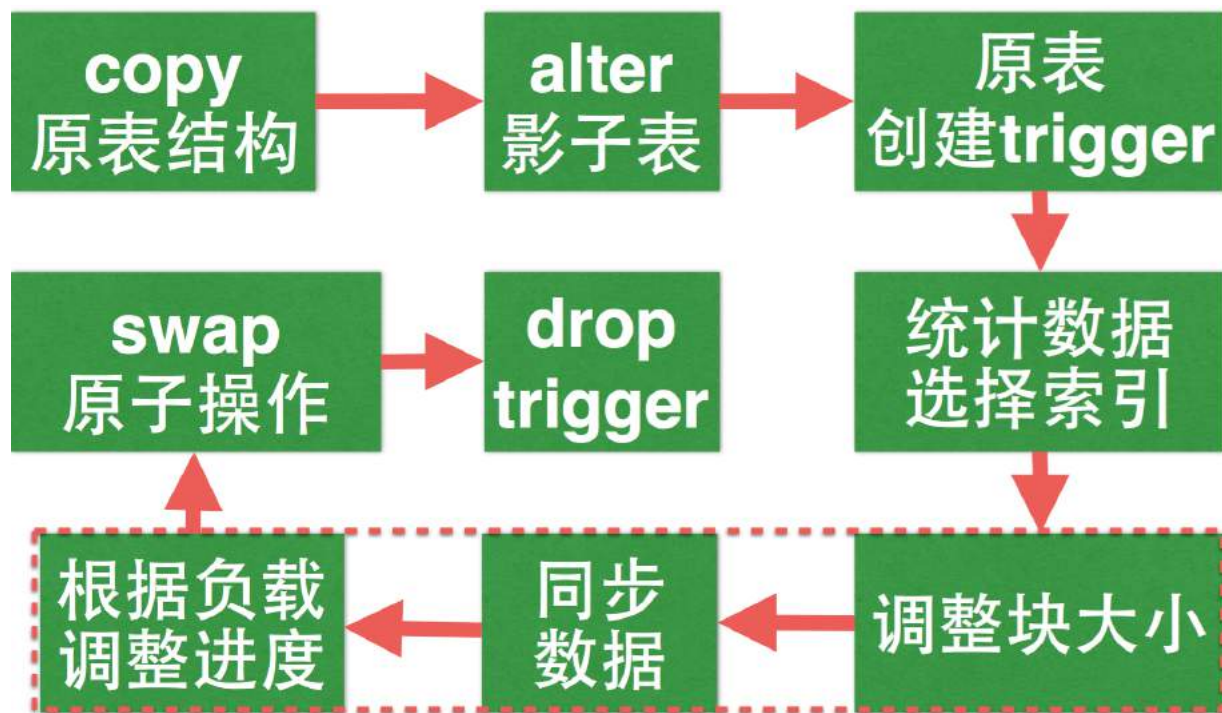
- 原生MySQL执行DDL是需要锁表的
- 对服务影响很大
- MySQL在这方面支持的是比较差的，对DBA来说是很痛苦的，
- 如何做到Online DDL呢，是不是就无解了呢

# 可选DDL方案





# Online Schema Change原理



# Online DDL 对比

CHANGE OPERATION	ONLINE DDL			PT-ONLINE-SCHEMA-CHANGE		
	ROW(S) AFFECTED	IS TABLE LOCKED?	TIME (SEC)	ROW(S) AFFECTED	IS TABLE LOCKED?	TIME (SEC)
Add Index	0	No	3.76	All rows	No	38.12
Drop Index	0	No	0.34	All rows	No	36.04
Add Column	0	No	27.61	All rows	No	37.21
Rename Column	0	No	0.06	All rows	No	34.16
Rename Column + change its data type	All rows	Yes	30.21	All rows	No	34.23
Drop Column	0	No	22.41	All rows	No	31.57
Change table ENGINE	All rows	Yes	25.30	All rows	No	35.54

# OSC的一些坑

- 添加唯一键，导致数据丢失
- 延时备份
- 行格式下，只在从库使用OSC，丢数据

# 备份系统

- 数据库数据安全性是首先要保证的，也是最核心的
- 备份的意义是什么呢
- **数据恢复！**
- **数据恢复！**
- **数据恢复！**

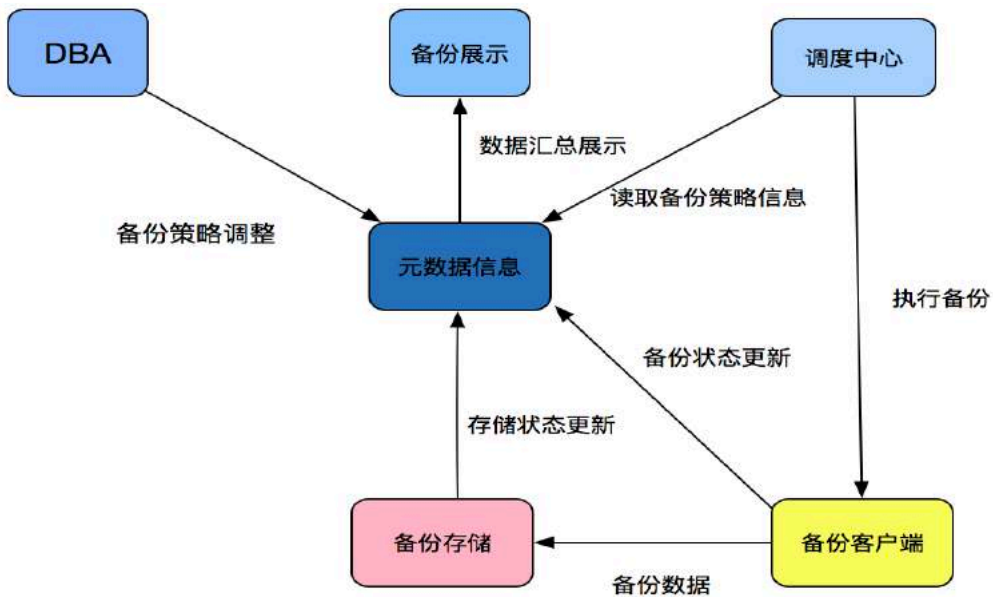


**DROP  
DATABASE**

# 备份方式

- 全量备份 VS 增量备份
- 热备 VS 冷备
- 物理备份 VS 逻辑备份
- 延时备份
- 全量binlog备份
- 建议方式
- 热备 + 物理备份，核心业务：延时备份 + 逻辑备份

# 备份框架



# 备份优化

- MyISAM表多flush tables with read lock时间过长问题
- Binlog备份



# 未来备份优化

## ➤采用分布式文件系统原因

解决存储分配的问题

解决存储NFS备份效率低问题

存储集中式管理

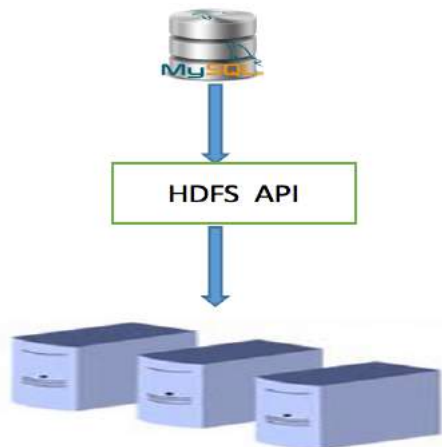
数据可靠性更好

## ➤使用分布式文件系统优化点

Pbzip压缩降低网络带宽消耗

元数据节点HA

erasure code方案





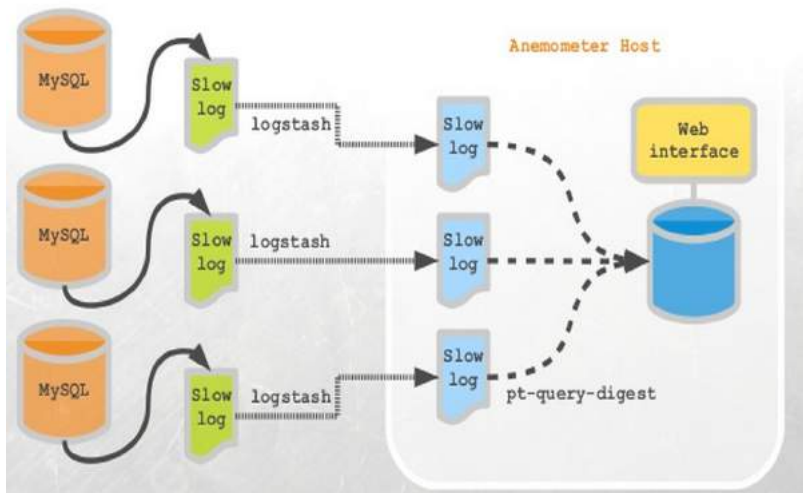
# 慢日志系统

- 慢日志是MySQL内部提供的SQL执行的统计日志
- 用于线上数据库性能诊断
- 为SQL性能优化提供依据





# 依赖组件

- pt-query-digest
- Logstash
- Anemometer



# 系统界面

 Box Anemometer

Datasources < 



Graph Search 

Table Search 

checksum

Find Query

From  
2015-08-20 18:09:49

To  
2015-08-21 18:09:49

Query first seen since

Table Fields

Custom Fields

checksum

date

hour

hour\_ts

minute\_ts

minute

snippet

index\_ratio

query\_time\_avg

rows\_sent\_avg

global\_query\_review

checksum

fingerprint

sample

first\_seen

last\_seen

reviewed\_by

reviewed\_on

comments

reviewed\_status

Filter By Host

Filter By Port

Where

Group By

checksum

Order By

Query\_time\_sum DESC

Having

Limit

5

Search

Query Sample Contains

Reviewed Status

Checksum

+ Show Raw SQL

Permalink

JSON

Showing 5 results

checksum	snippet	index_ratio	query_time_avg	rows_sent_avg	db_max	ts_cnt	Query_time_sum	Lock_time_sum	Rows_sent_sum	Rows_examined_sum
2181DD697BD2268F	SELECT	18.12	1.054931244498768	4990		184742	194890.1079711914	43.40777403116226	921827105	16702121652
FD25C60B1AD3B4DE	SELECT	13093.05	0.7499685775483659	32		72907	54677.95908331871	4.788258011576545	2305958	30192023432

## 一些优化

- Pt-query-digest加入ip和port指标收集
- Anemometer加入端口过滤条件支持
- 后续会加入对产品线和业务过滤
- 方便查询

# 性能优化

- MySQL优化
- 系统优化
- 硬件优化

# MySQL优化

- 更大的redo日志
- bufferpool dump
- 禁用Query Cache
- InnoDB
- jemalloc
- group commit
- transportable tablespace



# InnoDB文件

- 数据文件目录
- redo
- binlog

***datadir***

***innodb\_data\_file\_path***

***innodb\_undo\_directory***

***innodb\_log\_group\_home\_dir***

***log-bin***

***tmpdir***

随机IO , SSD

顺序IO , 机械硬盘

随机IO , SSD

# MySQL 5.6 VS 5.7

VARIABLE	5.6.26	5.7.8
binlog_error_action	IGNORE_ERROR	ABORT_SERVER
binlog_format	STATEMENT	ROW
binlog_gtid_simple_recovery	OFF	ON
eq_range_index_dive_limit	10	200
innodb_buffer_pool_dump_at_shutdown	OFF	ON
innodb_buffer_pool_instances	8	1
innodb_buffer_pool_load_at_startup	OFF	ON
innodb_checksum_algorithm	innodb	crc32
innodb_file_format	Antelope	Barracuda
innodb_large_prefix	OFF	ON
innodb_log_buffer_size	8388608	16777216
innodb_purge_threads	1	4
innodb_strict_mode	OFF	ON
sync_binlog	0	1





# MySQL复制

- GTID
- loss-less semi-replication
- group replication

		Where are transactions run?	
		Primary Copy	Update Everywhere
When does synchronization happen?	Eager	(MySQL semi-synch Replication)	MySQL Cluster MySQL Group 3 <sup>rd</sup> party: Galera
	Lazy	MySQL Replication/Fabric 3 <sup>rd</sup> party: Tungsten	MySQL Cluster Replication

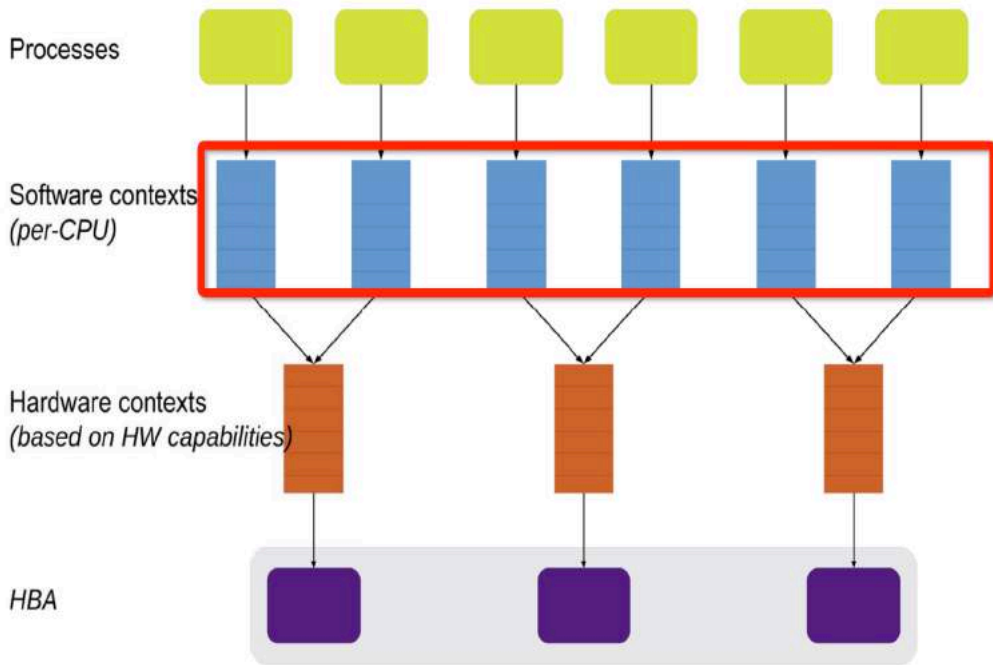
# 系统优化

- NUMA问题，建议关闭
- 调整swappiness
- 修改IO调度算法为noop/deadline
- 文件系统XFS/Ext4
- 系统limits限制，尤其是CentOS6上对max processes的限制，默认1024
- 网卡多队列，当然一般可能遇不到这种场景

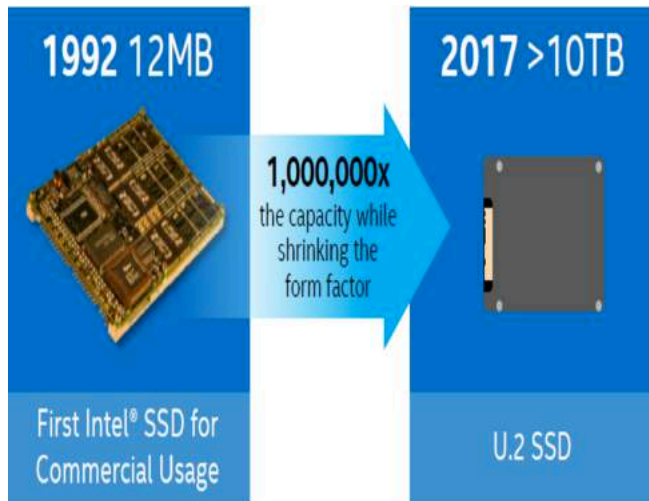
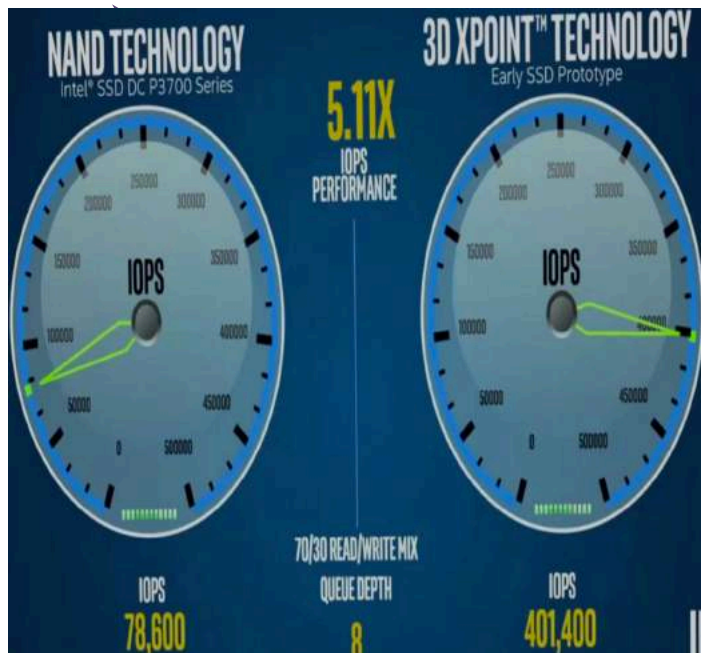
# 硬件优化

- 硬件发展很快，反而现在软件是瓶颈
- 使用SSD，SATA 或者PCIe
- CPU多核
- 大内存，单机多实例
- blk-mq
- scsi-mq

# block-mq



# 硬件发展



# 人还是比机器贵的

Cost of human: HIGH

Cost of computer: LOW

Good!!

# 未来发展

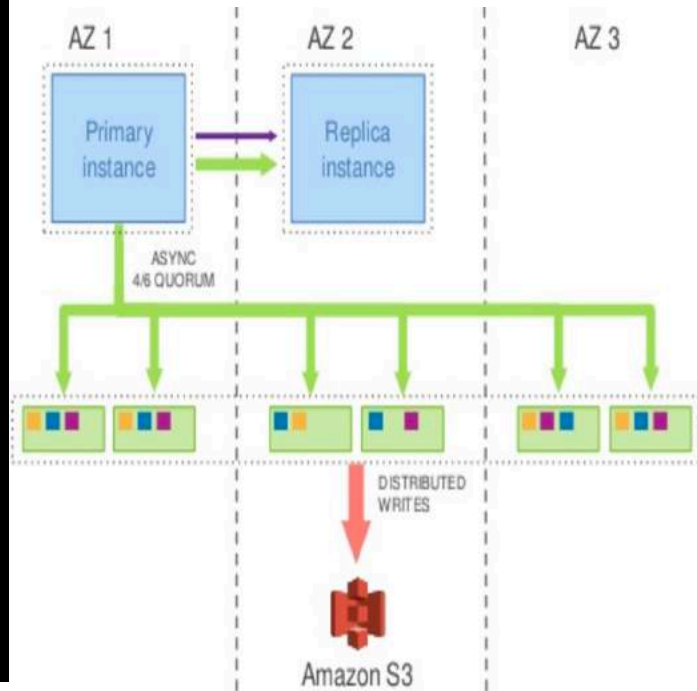
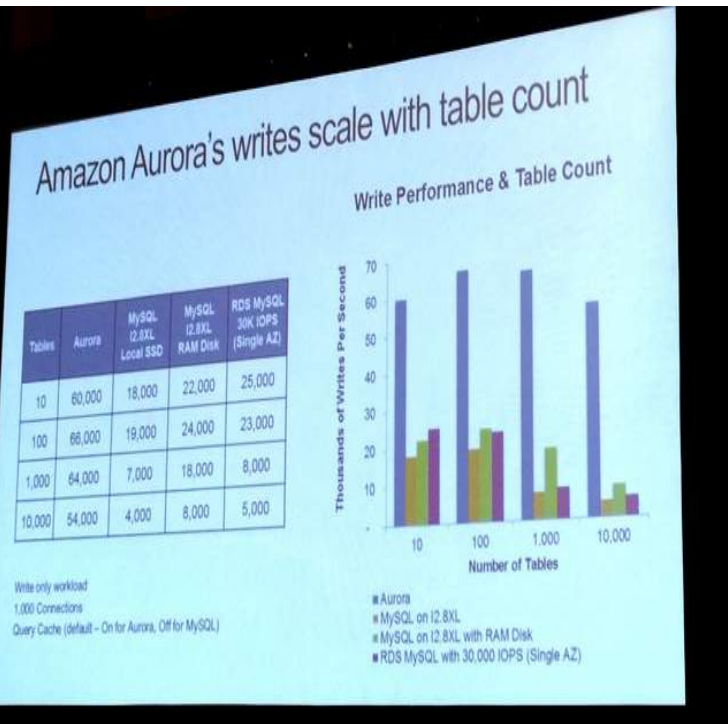
- 定制
- 容器
- 数据平台

# 定制

- Amazon Aurora
  - Compatible with the open source MySQL
  - Most of the smarts are in the storage
  - **A data insert in MySQL requires six writes , Aurora requires only two**
  - 软硬件结合
  - 最重要的地方就是可用性的提升，性能是其次



# Amazon Aurora



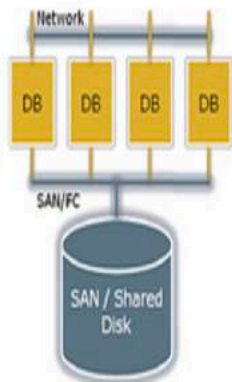
# 数据库架构

- shared nothing
- shared everything
- module

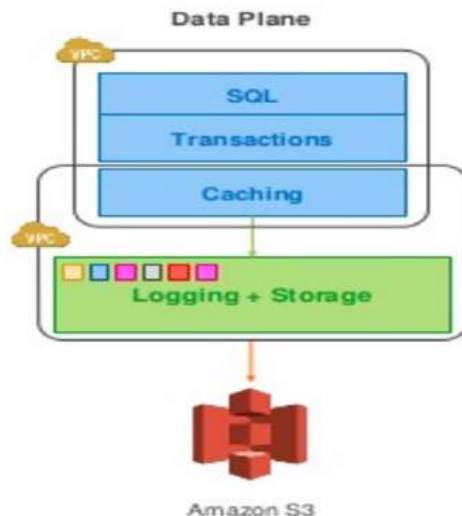
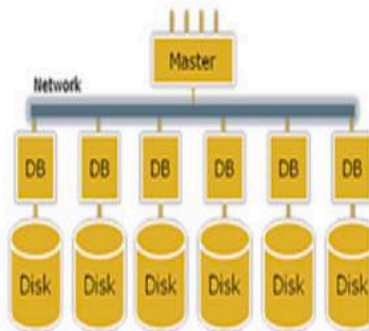
Shared-Everything  
(e.g. SMP server)



Shared-Disk (e.g. Oracle RAC)



Shared-Nothing (e.g. Greenplum)



# 容器

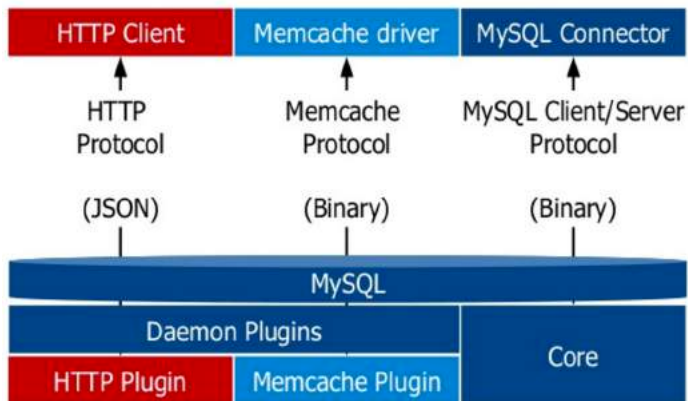
- 资源隔离
- 快速部署
- 性能瓶颈
- 存储限制

# 分布式存储

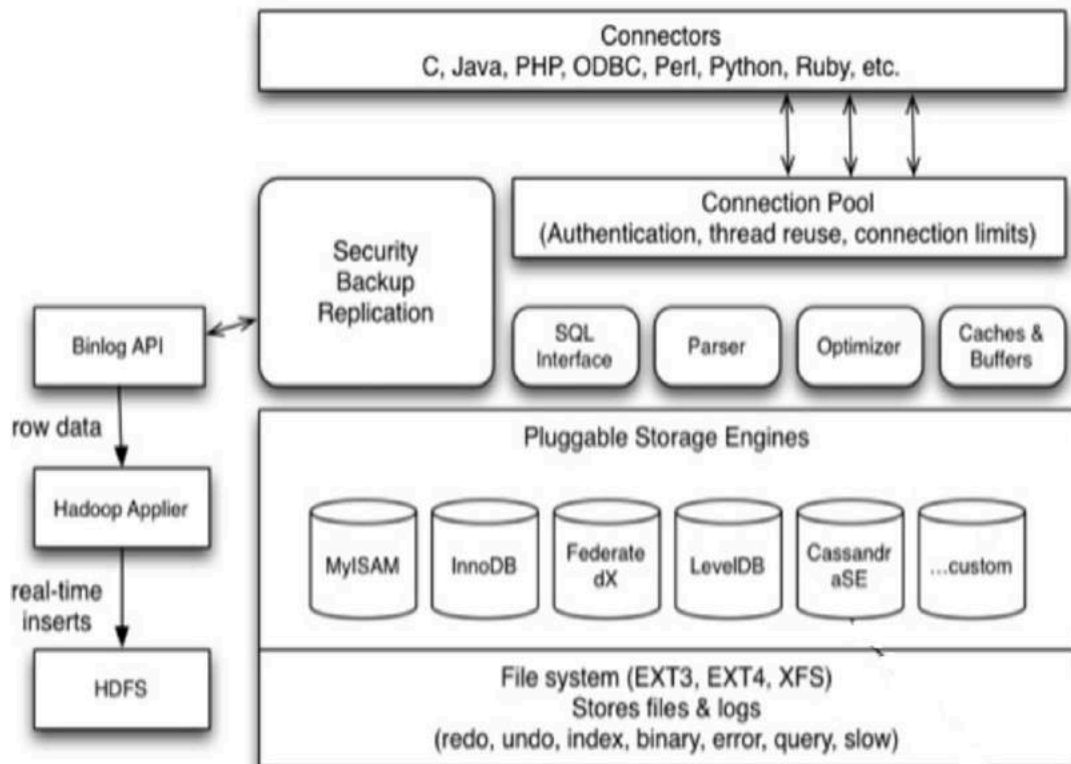
- 按照提供接口主要分成两类：基于文件接口和基于块设备
- 考虑到数据库场景的需求，使用分布式块设备接口是一个比较合理的选择
- 目前分布式块设备存储方案
  - Sheepdog
  - GlusterFS
  - Ceph
- 数据库的可伸缩性
- 高性能，相对而言，目前看性能在数据库上并不理想
- 提高整体资源利用率
- 简化存储管理

# 数据平台

- 多引擎
- 多种数据格式



# 数据平台



# 如何选择数据库

- 应用场景
- 数据量
- 可用性要求
- 数据安全性要求
- 运维复杂度



中国菜刀  
MySQL



瑞士军刀  
MongoDB



大象兵  
HBase



金箍棒  
Redis

# 数据库对比

数据库	优点	缺点	场景
MySQL	结构简单，部署方便，社区成熟，稳定性非常好，良好的事务和SQL支持	扩展性差，软件本身性能瓶颈大，没有成熟的集群方案，schema限制	百亿以内的数据存储，对数据安全性和事务支持有要求
MongoDB	Schema-free，快速开发，本身支持集群和sharding，支持空间索引	锁的粒度大，并发性能差，性能受限于内存，解决方案上有待考验	LBS，缓存，小文件存储
HBase	类Bigtable系统，基于Hadoop生态系统，良好的扩展性，高写入能力，数据自动分片	架构复杂，运维成本高	搜索，数据写入非常高，监控数据
Redis	高性能，部署简单，丰富的数据类型支持，支持数据持久化，集群方案支持	性能受限于内存，单进程问题	适合小数据高读写场景



# 总结

- 运维的规范化和自动化很重要
- 平台化和服务化是目标
- 优化是无止境的
- 适合自己的才是最好的



# 联系方式



zolker **Lv.26**

598

关注

5507

粉丝

3121

微博



zolker 

北京 海淀



扫一扫上面的二维码图案，加我微信

# 关注微信公众号 获取文档和更新



云和恩墨

(支持ORA错误自动查询)



恩墨学院



Oracle新闻



## Z3 – SQL审核工具提升SQL质量

### • 独特的 SQL 视图

- SQL生命周期管理 – 捕获、分析、拦截、形成SQL全周期管理平台；



z3 – SQL审核



## zData – 高性能弹性分布式存储解决方案

- 大数据整合与集中面临的平台压力
  - 去“门限”架构。通过Virtual SAN替代FC SAN
  - 同样的成本获得 20倍+ 的IO性能
  - 动态扩展。高性能的存储解决方案



zData – 分布式存储



## Announcing : BayMax 自动化巡检 即将开放云服务

- 自动化巡检 – 让DBA去完成20%最有价值的工作



BayMax自动化巡检

# 特别感谢 合作伙伴



云和恩墨  
ENMOTECH



Shannon Systems  
宝 存 科 技

ACOUUG  
All China Oracle User Group  
中国 Oracle 用户组

恩墨学院  
ENMOEDU

Data Driven World  
EN•CORE  
恩核(北京)信息技术有限公司

OCMU  
OCM 联盟

SDOUG  
Shandong Oracle User Group  
山东 Oracle 用户组

ITPUB

ChinaUnix

IT168.com  
www.it168.com

31 会议

Mellanox  
TECHNOLOGIES  
Connect. Accelerate. Outperform.™

乐导科技  
alidao.com

慕课网  
imooc.com



清华大学出版社

Broadview®  
www.broadview.com.cn

TURING  
图灵教育

华章科技

