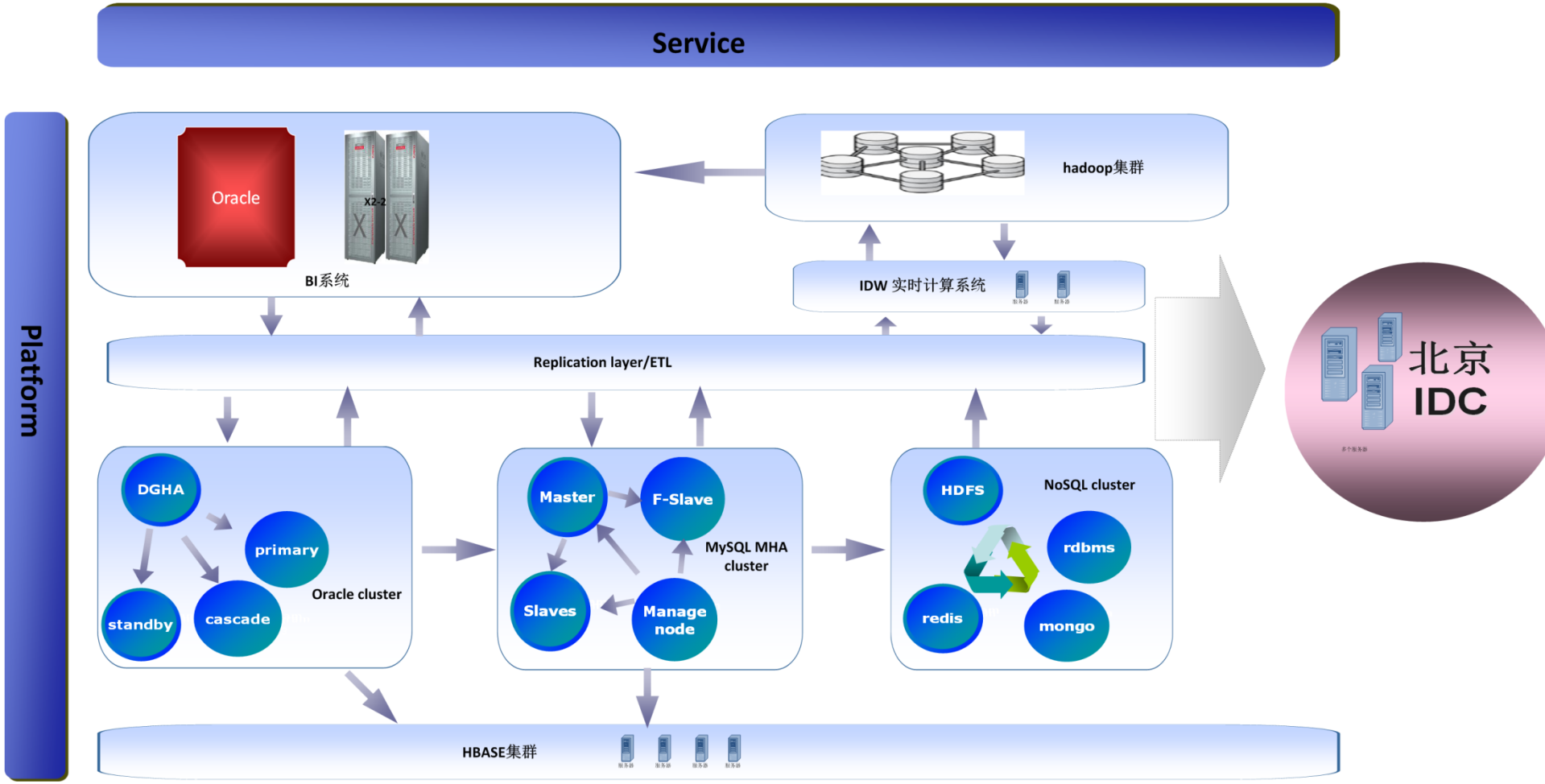
1号店数据库架构

介绍内容

- 1号店数据库架构
- Oracle集群与MySQL集群HA
- Oracle DGHA的特性
- Oracle DGHA：典型切换场景
- Oracle DGHA：故障转移过程
- Oracle DGHA：Failover机制
- MySQL MHA是什么？
- MySQL 为什么使用MHA？
- MySQL MHA：典型的场景
- MySQL MHA：切换过程
- MySQL MHA：Failover机制
- HBase 架构

1号店数据库架构


Service
Platform
Oracle
X2-2
BI系统
hadoop集群
IDW 实时计算系统
Replication layer/ETL
北京
IDC
DGHA
primary
standby
cascade
Oracle cluster
Master
F-Slave
Slaves
Manage node
MySQL MHA cluster
HDFS
NoSQL cluster
rdbms
redis
mongo
HBASE集群

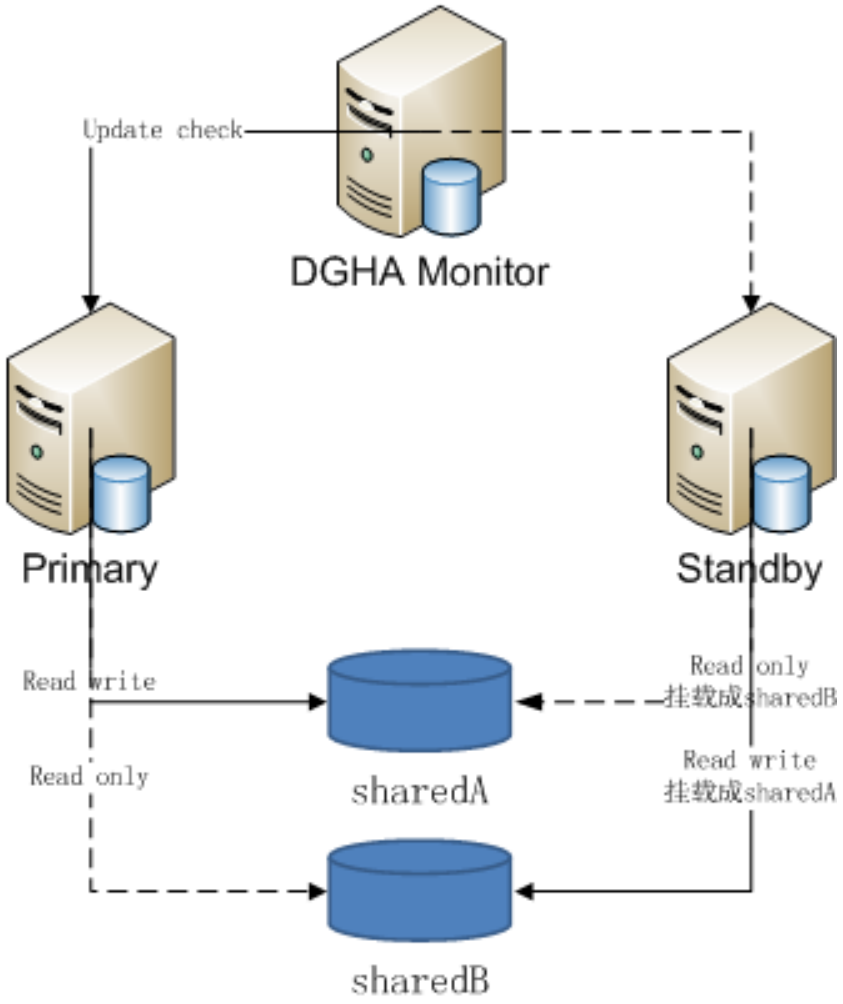
Oracle集群与MySQL集群HA

- **Oracle HA : DGHA**
  - --Prod
  - --User
  - --Edm
  - --WMS

- **MySQL HA : MHA**
  - --Gss
  - --团购
  - --Mobile
  - --订单状态
  - --评论
  - --第三方订单
  - --1mall
  - --frontweb

DGHA是什么？

- DGHA:此脚本针对oracle dataguard设计，使用共享存储存放redo,spfile以及controlfile从而达到了切换数据零丢失。
- 目的：自动管理主库备库切换以便最少化当机时间
- 由一个的Perl主脚本和几个shell脚本组成
- 可采用集中管理模式，可以管理多套数据库集群。
- 原有主备库不需要安装额外的软件模块



现有的DATAGUARD HA方案

- DataGuard Broker
  --只能failover到某个指定实例
  --是oracle官方方案

- 共享存储模式
  --需要共享一份数据，需要存储
  --不好利用PCIE等高性能IO设备

- Dataguard模式
  --简单、不需要存储
  --可能有数据丢失，可能会破坏整个集群

- 共享redo/controlfile/spfile模式
  --需要小的存储
  --数据零丢失，不会破坏集群

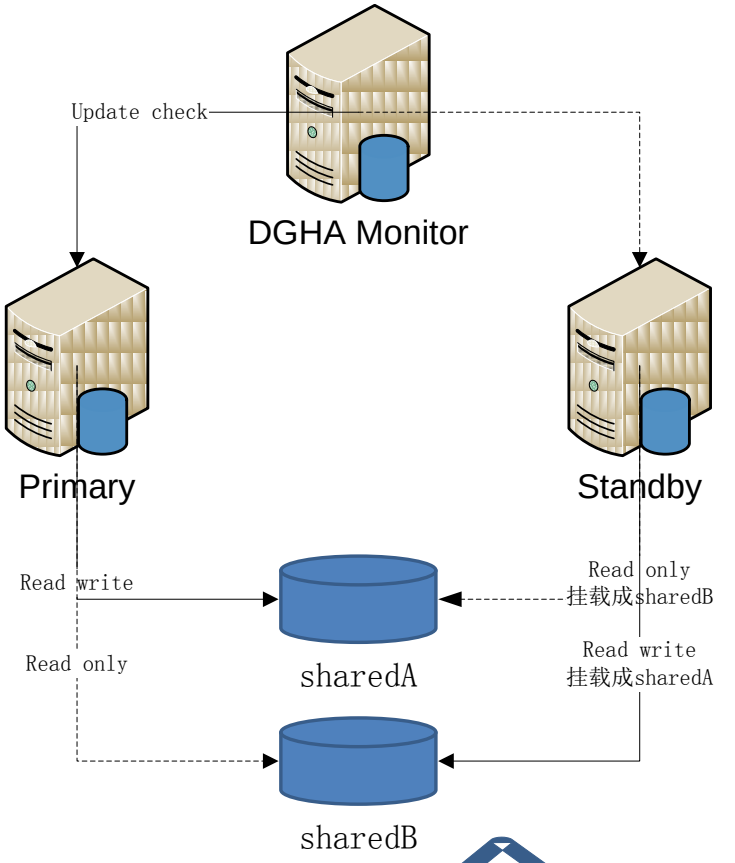
DGHA的特性

- 主服务器的自动监控和故障转移
- 切换数据零丢失
- Power off功能，避免脑裂
- 详细的日志输出
- 短信和邮件报警功能

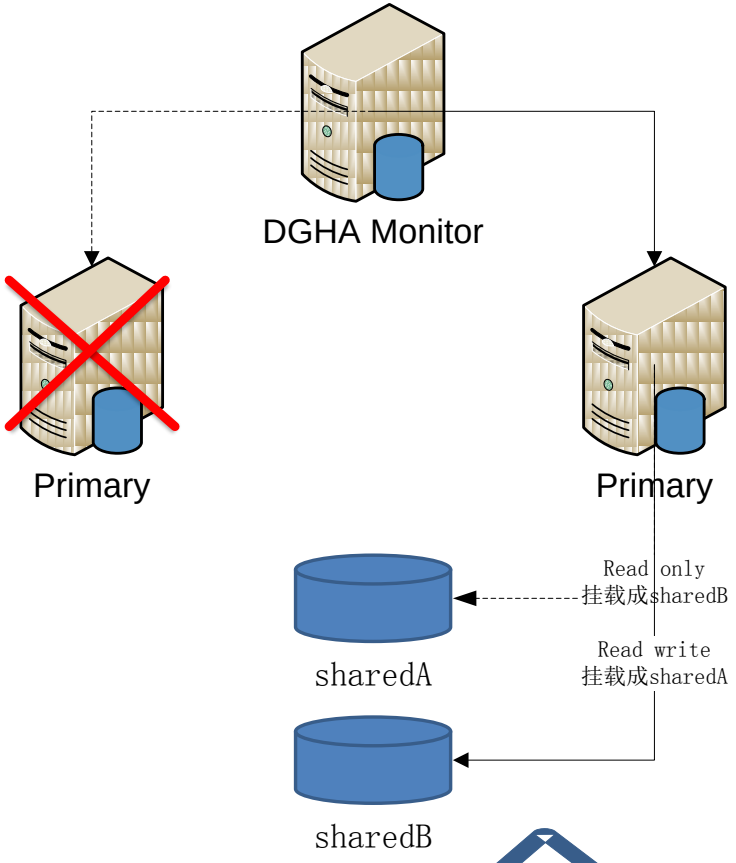
DGHA：典型切换场景

- 监控整个集群：定期check主库
- 如果检测到primary故障，启动切换
- 检查备库是否达到要求
- 关闭主库，断电
- 用主库的redo等替换备库的，recover备库，然后打开
- 发送Email failover 报告，停止监控

DGHA：故障转移过程



A 主备库为DG，redo,spfile和control文件存放于共享存储，存储对主备服务器都可见
B 更新主库心跳表，判断主库是否可用



A 关闭原主库实例，Power off原主库服务器
B 备份原备库的online redo和control文件；拷贝原主库的online redo和control file
C 备库应用原主库online redo恢复，保证数据零丢失
D 将备库切换到主库,其他备库同步新的主库

DGHA：Failover机制

- 一般情况下完成整个过程需3-5分钟
- check频率一般设置为10秒
- 检测主库一般为1分钟：3次check+连接超时+ssh超时
- 主库关闭+备库检测+power off 一般为1-2分钟
- 备库关闭+恢复+打开一般为1-2分钟

DGHA：Failover机制

主库检测频率
主库连续3次，每次间隔一定时间（比如10s）

主库检测类型
长连接UPDATE HACHECK SET CHECK_TIME=SYSDATE
本地UPDATE HACHECK2 SET CHECK_TIME=SYSDATE

备库lag检测
最大延迟3分钟

本地检测最大重试次数（for maximum number of processes (xxx) exceeded）
最大次数8次

Power off 状态检测
最大次数8次

Ssh超时时间
每次为5秒，重试3次

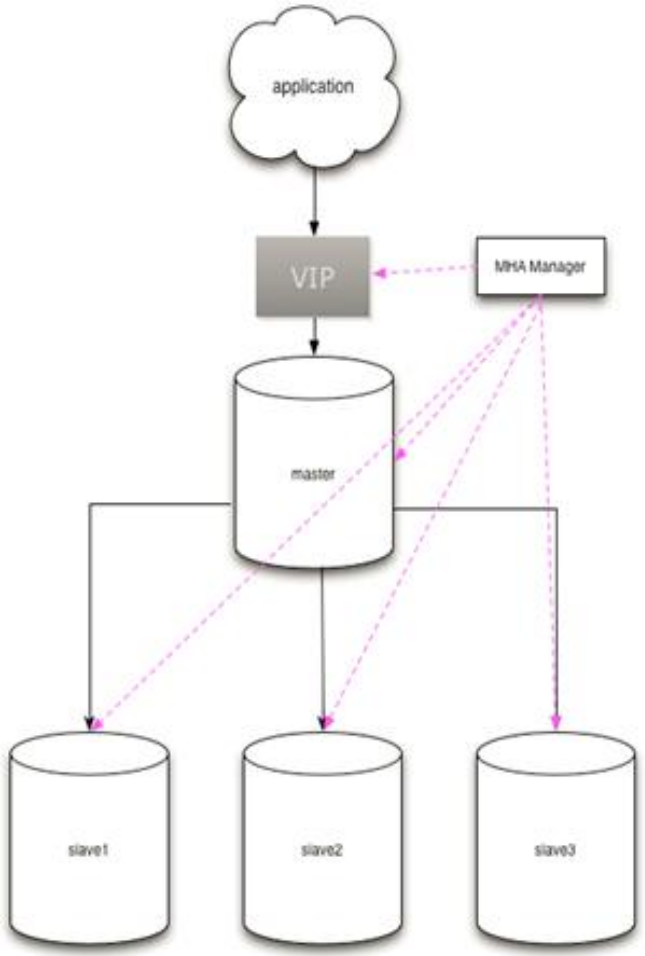
MHA是什么？

- MHA for MySQL: Master High Availability
  Manager tools for MySQL

- 目的: 自动管理master failover & slave promotion以便最少化当机时间

- 由一系列的Perl脚本组成

- http://code.google.com/p/MySQL-master-ha/



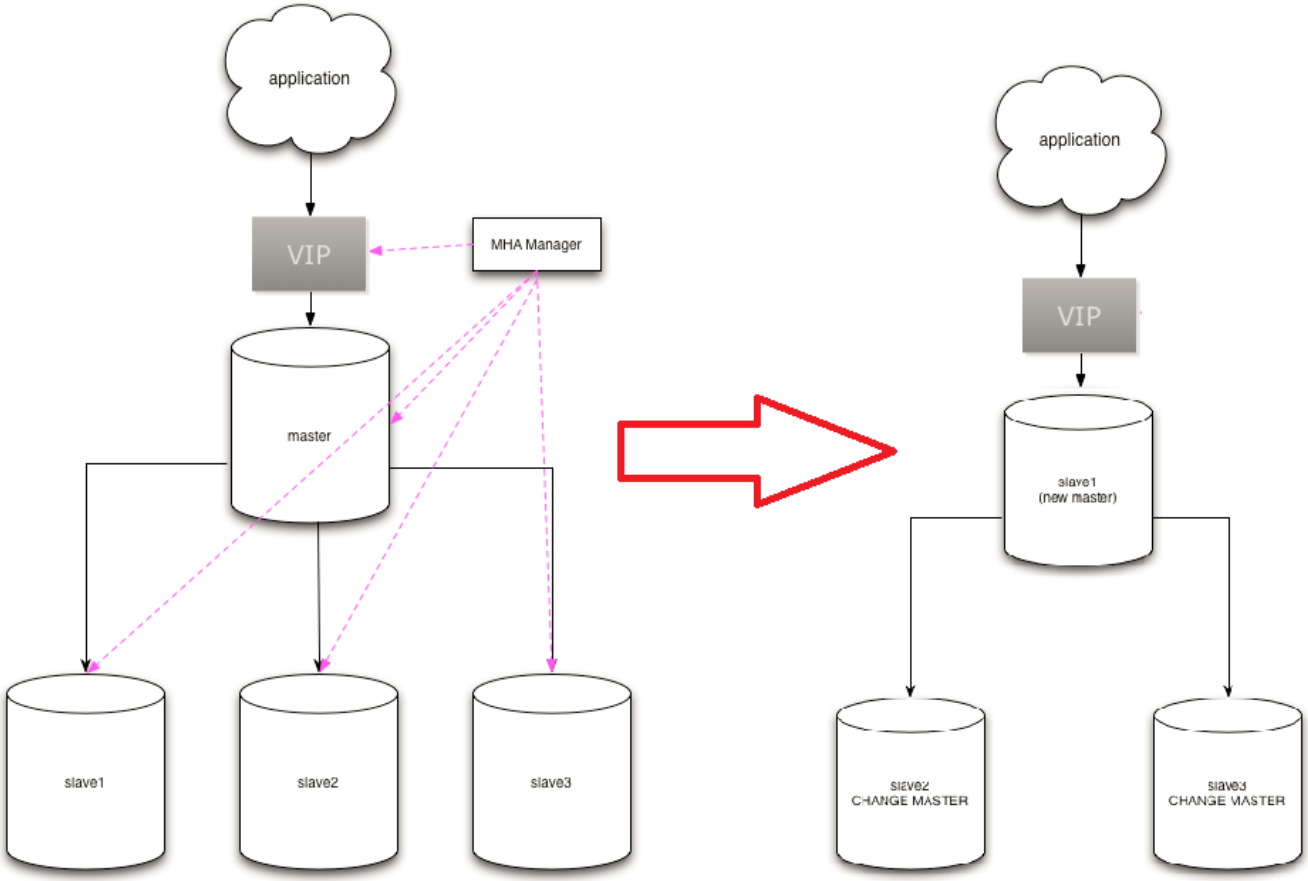
为什么使用MHA？

- 自动的监控整个集群，自动进行故障转移
- 可以进行有计划的master切换，进行在线维护
  OPTIMIZE/ALTER table,
  Software or hardware upgrade
- 可以自动的或交互式的故障转移
- 可以管理VIP或者集成VIP管理软件

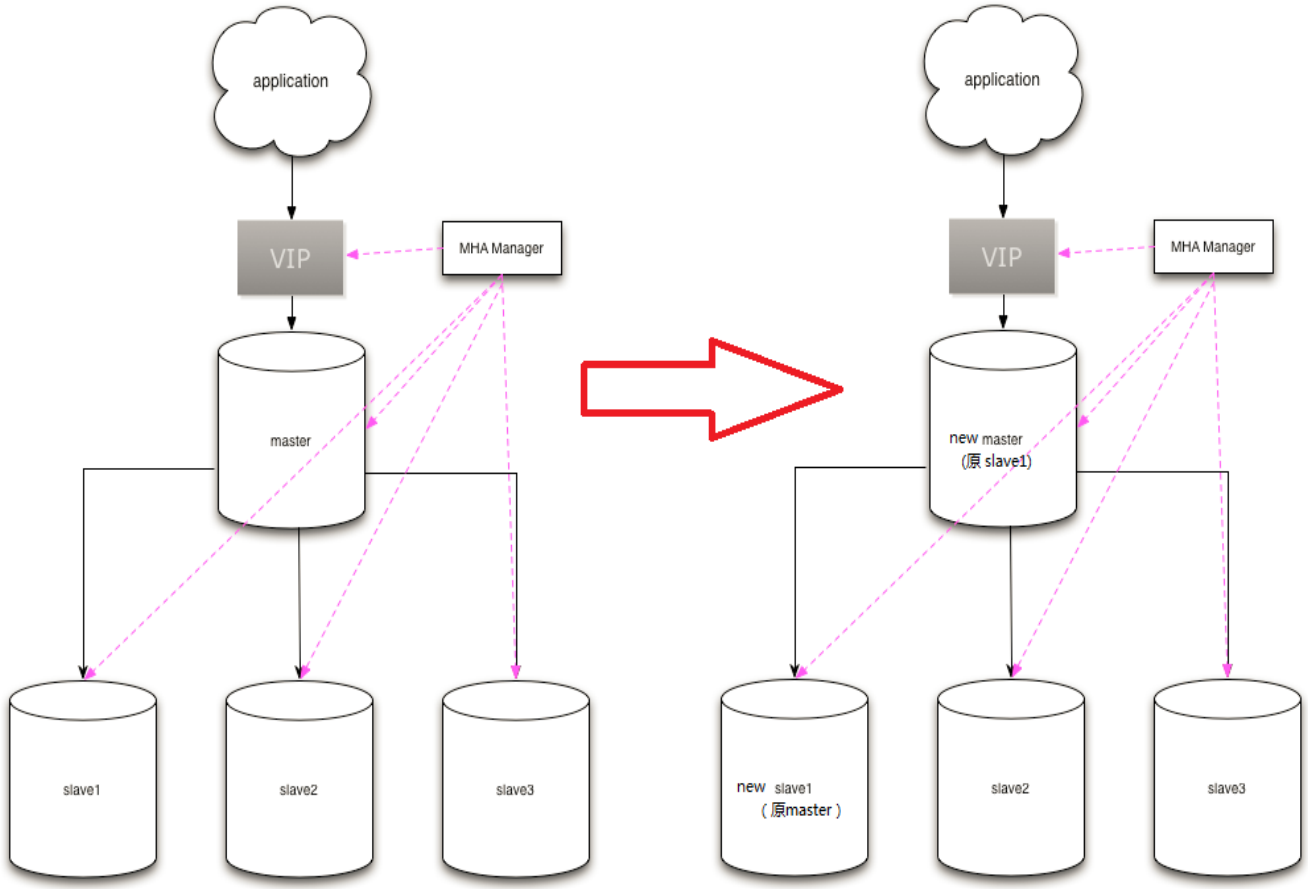
MHA：典型的场景

- 监控整个集群
- 如果检测到Master故障，则立即切换到一个候选Master或者拥有最新日志的Slave ，从而成为新的Master
- 把其他的slave切换到新的master上
- 输出或者Email failover 报告，停止监控

MHA：故障转移过程


application
VIP
MHA Manager
master
slave1
slave2
slave3
application
VIP
slave1
(new master)
slave2
CHANGE MASTER
slave3
CHANGE MASTER

MHA：在线切换过程


application
VIP
MHA Manager
master
slave1
slave2
slave3
application
VIP
MHA Manager
new master
(原 slave1)
new slave1
(原master)
slave2
slave3

MHA：典型的时间点

- Usually no more than 10-30 seconds
- 0-10s: Master failover detected in around 10 seconds
- (optional) 10-20s: 10 seconds to power off master
- 10-20s:apply differential relay logs to new master
- Practice: 4s @ DeNA, usually less than 10s

MHA：Failover机制

ping_interval：检测频率

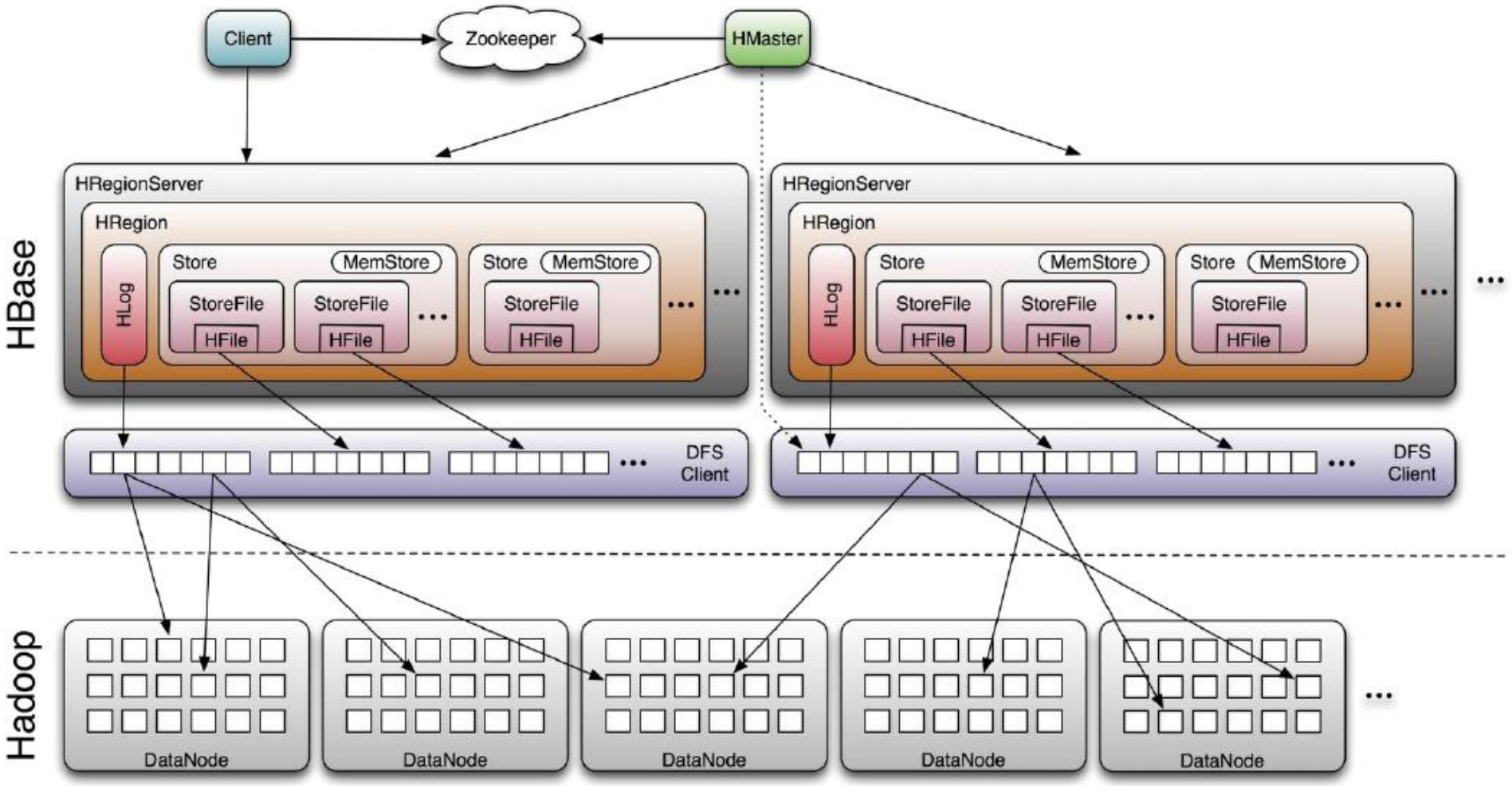
This parameter states how often MHA Manager pings(executes ping SQL statement) the master. After **missing three connection intervals in a row, MHA Manager decides that the MySQL master is dead.** Thus, the maximum time for discovering a failure through the ping mechanism is four times the ping interval. The default is 3 (3 seconds).

If MHA Manager fails to connect by too many connections or authentication errors, it doesn't count that the master is dead.

ping_type：检测类型

(Supported from 0.53) By default, MHA establishes a persistent connection to a master and checks master's availability by executing "SELECT 1" (**ping_type=SELECT**). But in some cases, it is better to check by connecting/disconnecting every time, because it's more strict and it can detect TCP connection level failure more quickly. Setting **ping_type=CONNECT** makes it possible.

HBase架构


Client
Zookeeper
HMaster
HBase
HRegionServer
HRegion
HLog
Store
MemStore
StoreFile
HFile
DFS Client
Hadoop
DataNode

参考资料

1 http://search.cpan.org/~salva/Net-OpenSSH-0.60/lib/Net/OpenSSH.pm

2 MHA mha4MySQL-manager-0.55\samples\scripts\power_manager.pl

3 https://code.google.com/p/MySQL-master-ha/

4 Automated, Non-Stop MySQL Operations and Failover(Yoshinori Matsunobu)

5 MHA: Getting Started & Moving Past Quirks

6 https://code.google.com/p/MySQL-master-ha/