本文是作者在ACMUG 2016 MySQL年会上的演讲内容，版权归作者所有。

中国MySQL用户组（China MySQL User Group）简称ACMUG。
ACMUG是覆盖中国MySQL技术爱好者的一个技术社区，是Oracle User Group Community和MairaDB Foundation共同认可的MySQL技术社区。

我们关注MySQL，MariaDB，以及其他一切周边的开源数据库和开源工具，我们交流使用经验，推广开源技术，为开源贡献力量。

我们是开放社区，欢迎任何关注MySQL及其相关技术的人加入，我愿意跟其他任何技术组织和团体保持沟通和展开合作。

我们期望在我们的活动中大家都能以开心的、轻松的姿态交流技术，分享技术，形成一个良性循环，从而每个人都可以有一份收获。

ACMUG的口号：开源，开放，开心

关注ACMUG公众号，参与社区活动，交流开源技术，分享学习心得，一起共同进步。

MySQL DBA打开Hadoop的正确姿势

林水彬

waterbinlin@tencent.com

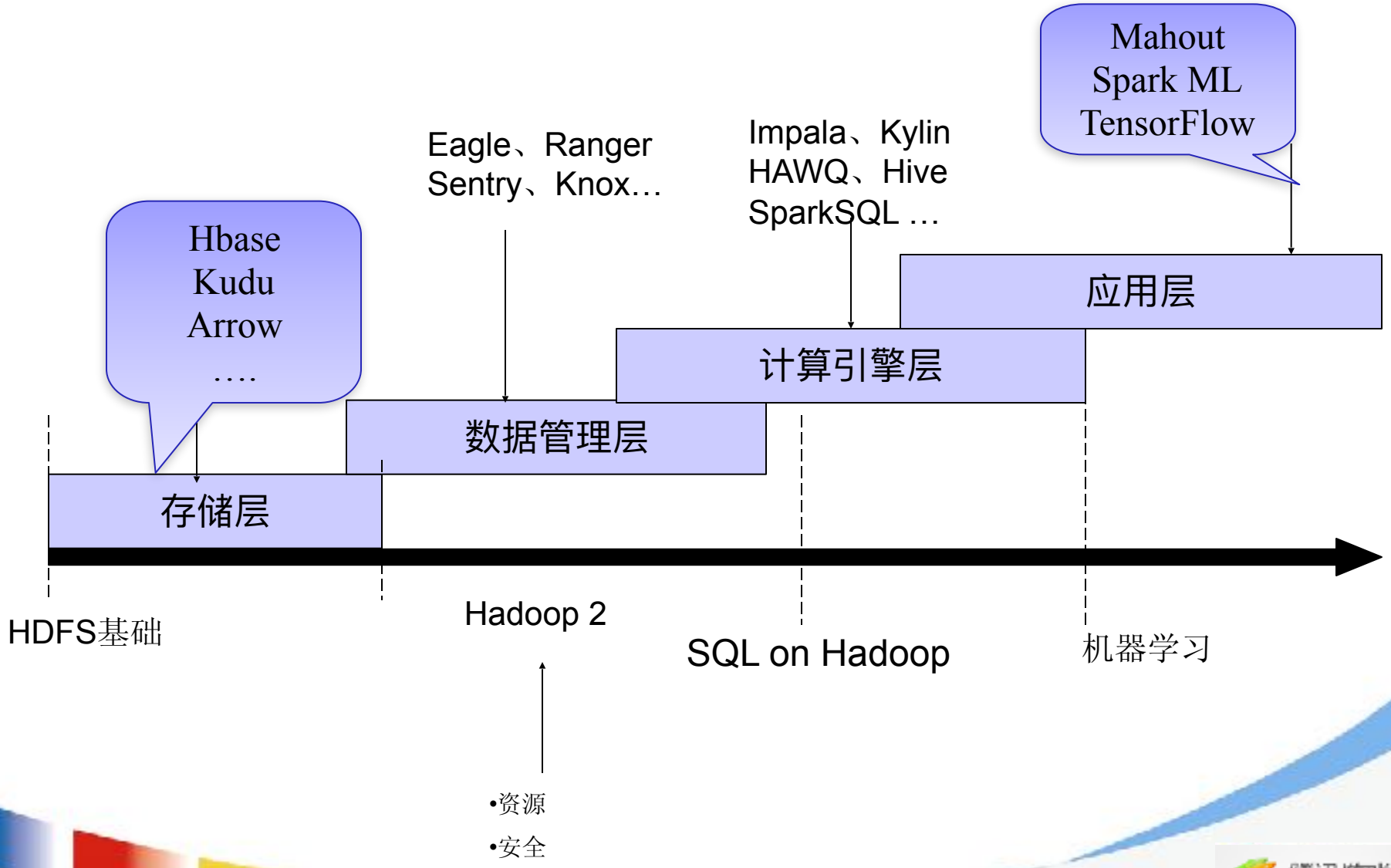
腾讯游戏
Tencent Games

议程

- Hadoop 生态介绍
- HDFS 原理简介
- MySQL 和 Hadoop 概念、操作的对比
- HDFS运营实战
- Hadoop在腾讯游戏内部系统的运用

生态发展

生态发展

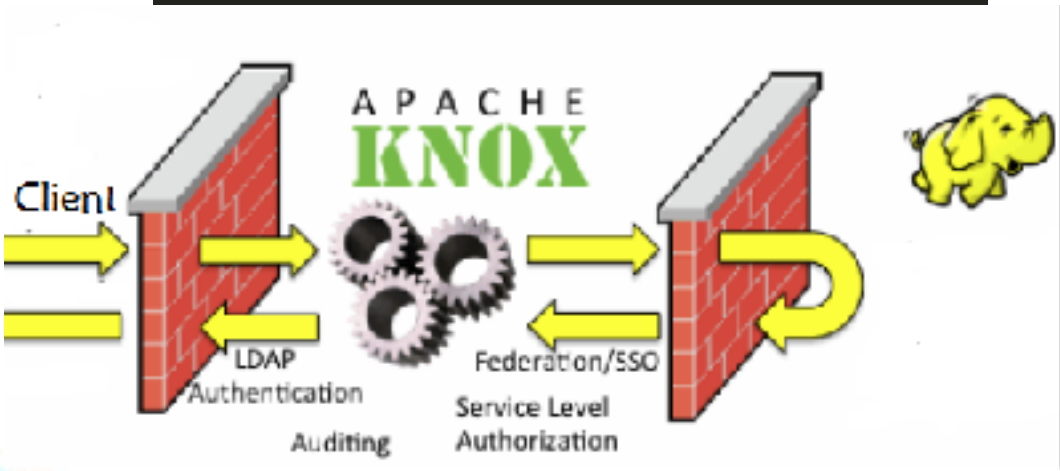

Mahout
Spark ML
TensorFlow
Eagle、Ranger
Sentry、Knox…
Impala、Kylin
HAWQ、Hive
SparkSQL …
Hbase
Kudu
Arrow
….
应用层
计算引擎层
数据管理层
存储层
HDFS基础
Hadoop 2
SQL on Hadoop
机器学习
•资源
•安全

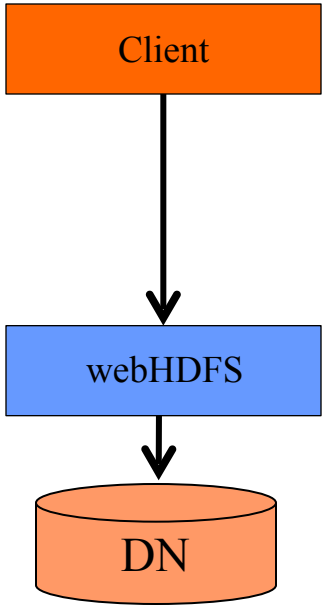
生态发展

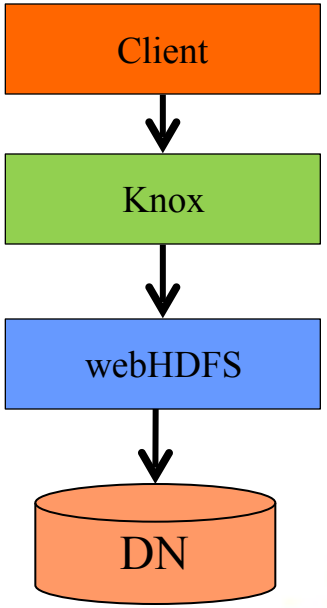
- Knox 架构
- 如何禁用原来的入口？

  通过把webHDFS服务绑定在本地，从而就禁用了从任何地方发起的请求，避免DataNode的裸奔

```
<property>
   <name>dfs.datanode.http.address</name>
   <value>127.0.0.1:50075</value>
</property>
```







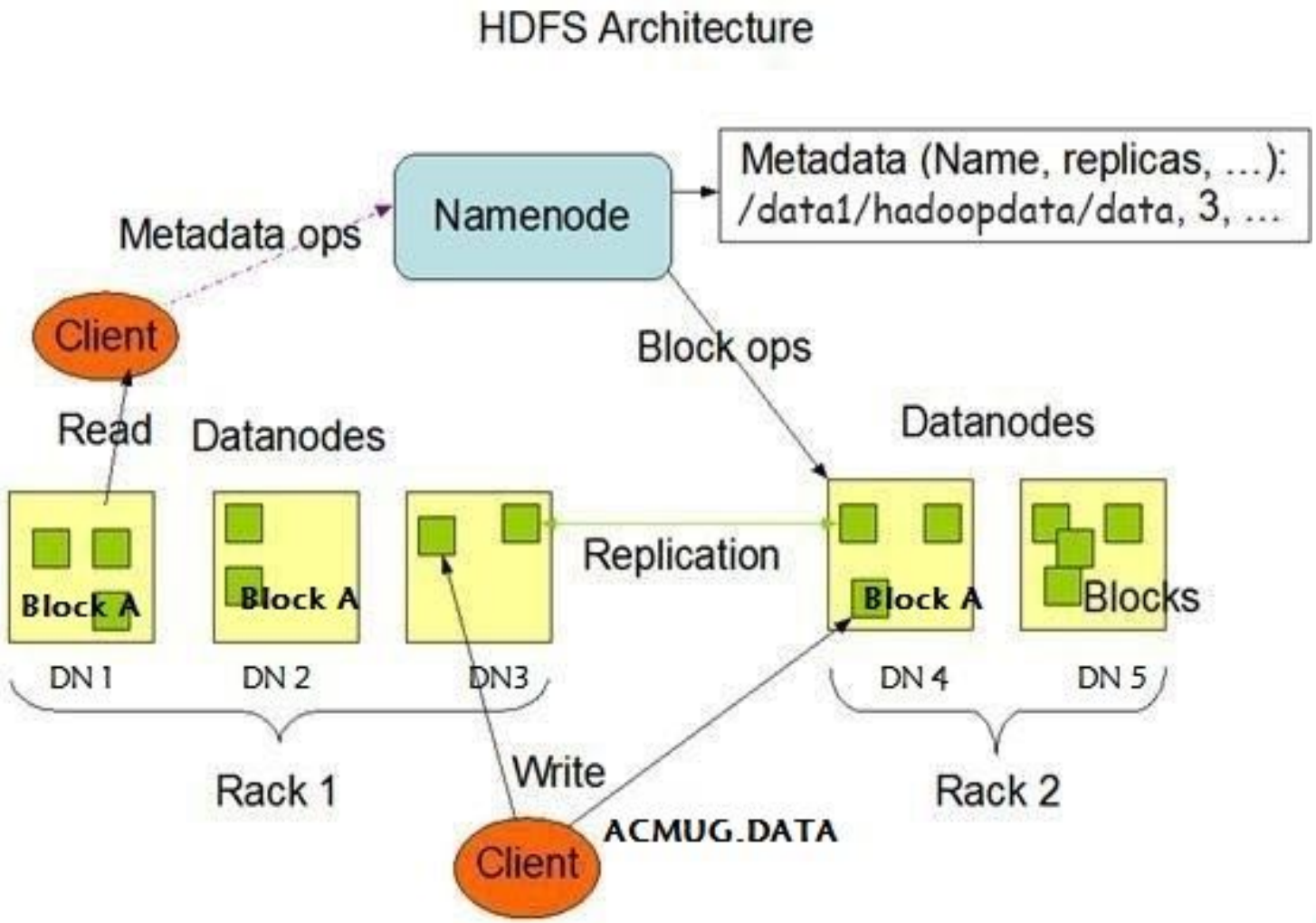
生态发展

- Knox-793
  最新release版本都被影响：0.10、0.9、0.9.1

- 官方确认

>
> Key: KNOX-793
> URL: https://issues.apache.org/jira/browse/KNOX-793
> Project: Apache Knox
> Issue Type: Bug
> Components: Server
> Affects Versions: 0.9.1
> Reporter: linwaterbin
> Fix For: 0.11.0

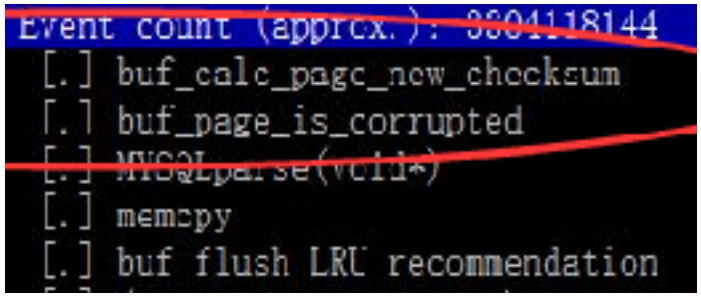
HDFS原理


HDFS Architecture
Namenode
Metadata (Name, replicas, …):
/data1/hadoopdata/data, 3, …
Metadata ops
Client
Block ops
Read
Datanodes
Datanodes
Block A
Block A
Replication
Block A
Blocks
DN 1
DN 2
DN3
DN 4
DN 5
Rack 1
Rack 2
Write
ACMUG.DATA
Client

数据安全

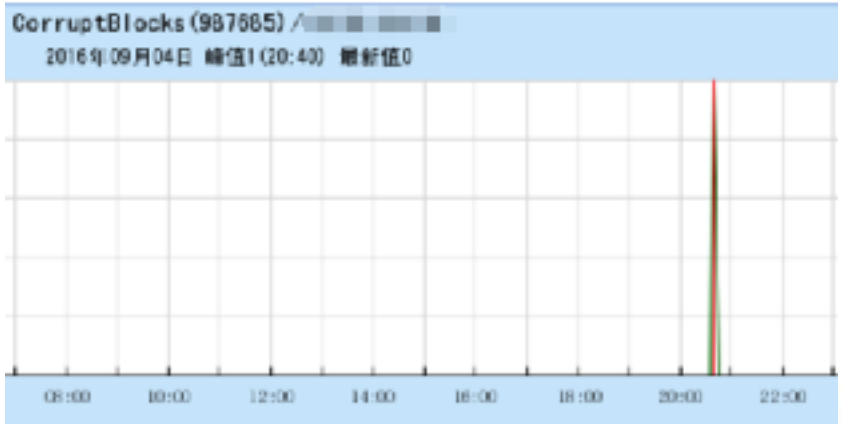
MySQL

1、InnoDB 页头页尾checksum



Hadoop

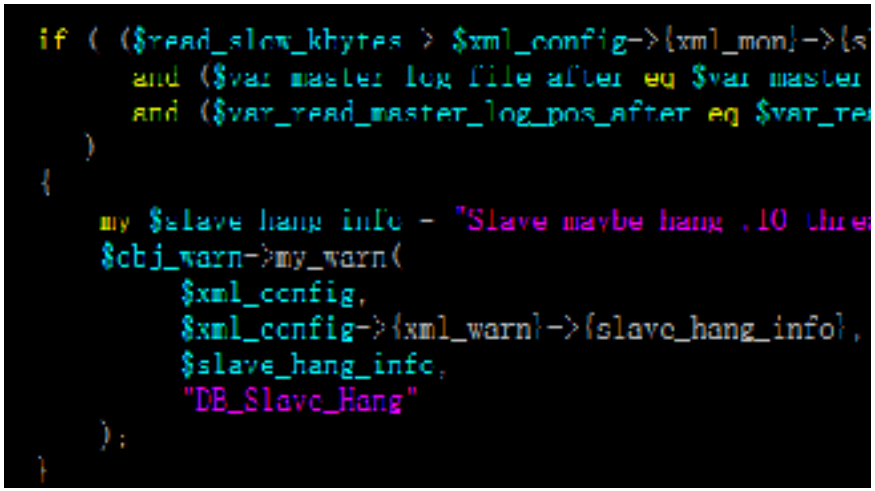
- 案例：业务A回档因某个binlog文件损坏导致数据丢失
- Read Checksum
- BlockScanner：系统自动触发坏页修复



架构

MySQL

1、基于主从复制
2、基于NDB引擎
3、基于中间件/proxy
4、基于Galera协议



```
if ( ($read_slow_kbytes > $xml_config->{xml_mon}->{s
      and ($var_master_log_file_after eq $var_master
      and ($var_read_master_log_pos_after eq $var_re
   )
{
    my $slave_hang_info = "Slave maybe hang ,10 thre
    $obj_warn->my_warn(
        $xml_config,
        $xml_config->{xml_warn}->{slave_hang_info},
        $slave_hang_info,
        "DB_Slave_Hang"
    );
}
```

Hadoop

- NameNode HA有QJM、BackupNode、NFS
- DataNode 多副本冗余存储
- DN每隔3秒向NN上报心跳
- DN无论主动故障还是被动故障（模拟只读），在630S+都会默认启动副本重建，原机器服务恢复以后，会自动停止副本重建流程，某次磁盘故障观察发现3.5T数据迁移耗用1小时25分钟，3.5TB即使物理备份导入导出，除了DB硬盘负载，还需要网络IO，还需要N小时的时间进行数据恢复和slave跟进。

运维

MySQL

1、show variable like
2、safe-updates=1
3、show global status

Hadoop

- hdfs getconf -confKey
- hdfs dfsadmin -safemode
- curl -s http://$self->{LOCAL_HOSTNAME}:50070/jmx?qry=$qry->[0]

运维实践

- 部署
- QJM高可用
- summer监控
- NFS Gateway
- balancer策略
- 集群数据同步
- 在线扩容
- webHDFS API
- Streaming编程

部署

KEY	VALUE
基础软件	•目录规划 •版本建议用软链接 •整个集群共用一套配置 •配置SSH互信时，记得NameNode本身也得配置 •机架感知，脚本所获取的输入参数是IP，不是域名
QJM	JournalNode和DataNode不建议部署在同一个盘 •fencing的 2 种策略，只能用回车键分割
验证	hdfs dfsadmin -report 验证 yarn健康判断 client随机put一个文件 active和standby角色确认

部署

- 机架感知 (shell版)

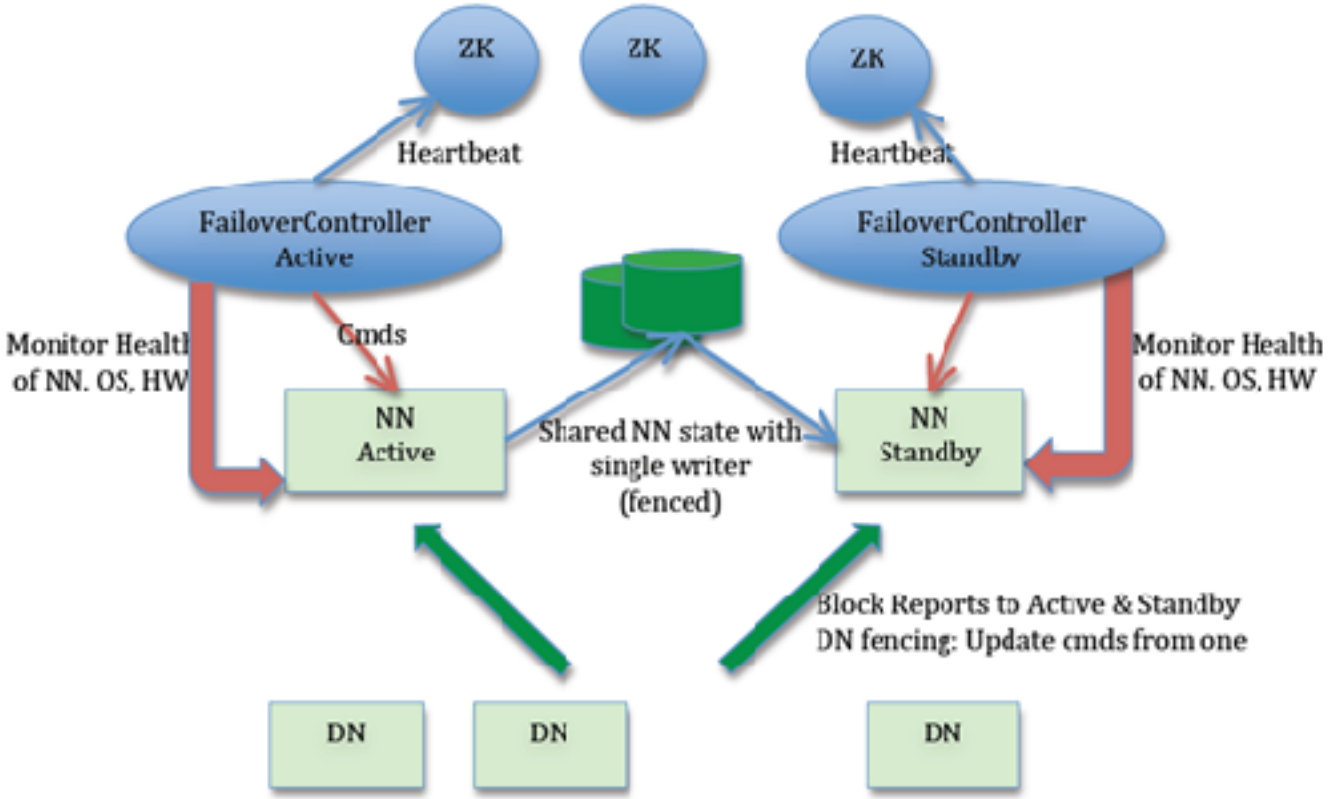
```
mysqlconn="mysql  u  p  h[illegible]"
RACK_PREFIX=default


for i in `echo $@`
do
  rack_id=$($mysqlconn  N  e "select concat_ws('/',IDCUnitID,Equipment)
  echo "/$RACK_PREFIX/$rack_id"
done
```

- fencing策略

```
<property>
  <name>dfs.ha.fencing.methods</name>
  <value>sshfence(hadoop:[illegible]0)
    shell(/bin/true)
  </value>
</property>
```

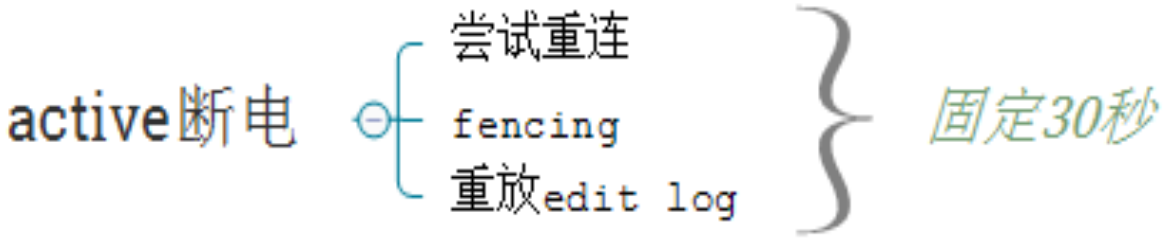
QJM

- 原理介绍


ZK
ZK
ZK
Heartbeat
Heartbeat
FailoverController
Active
FailoverController
Standby
Monitor Health
of NN. OS. HW
Cmds
NN
Active
Shared NN state with
single writer
(fenced)
NN
Standby
Monitor Health
of NN. OS. HW
Block Reports to Active & Standby
DN fencing: Update cmds from one
DN
DN
DN

QJM

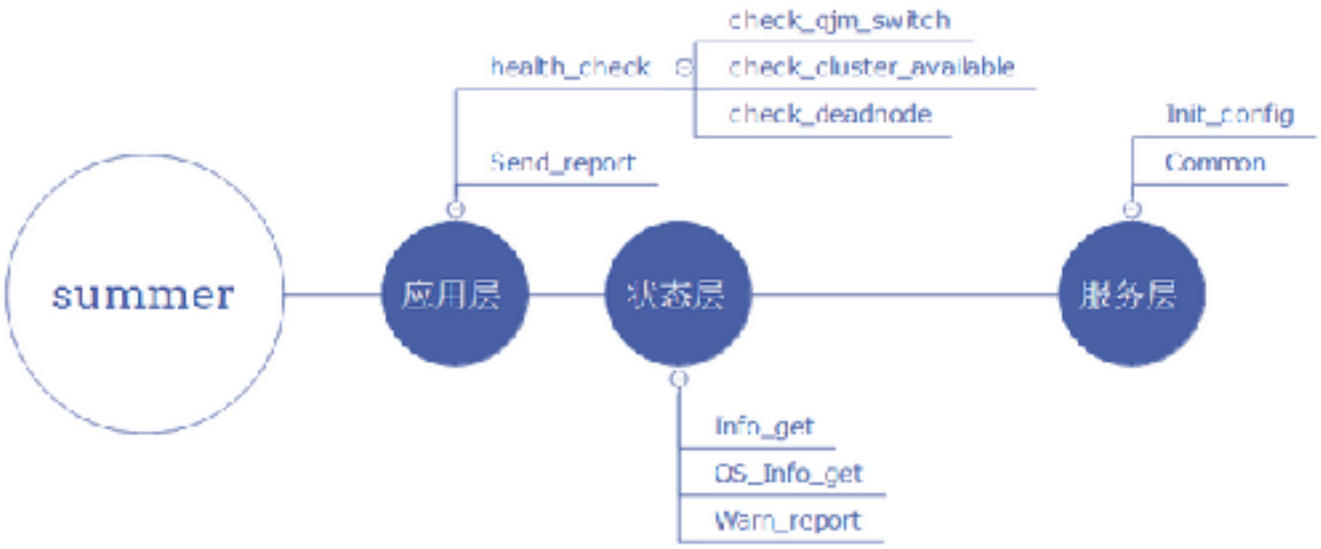
- 切换时间



- 重建热备
  - hdfs namenode -bootstrapStandby

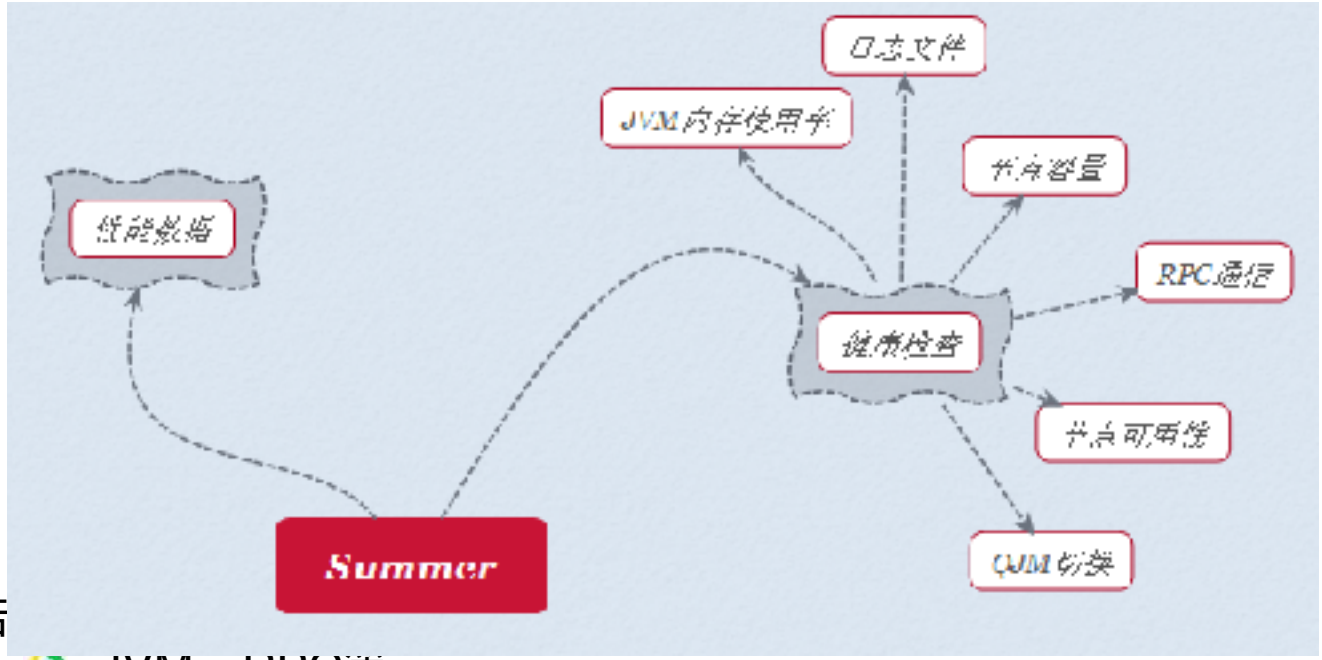
监控

- Ganglia 不足
  - 依赖rrdtool，安装复杂，若要持久化需要写脚本转存
  - 不支持电话/RTX/短信告警
  - 虽然提供Python扩展API，但Perl才是我们团队第一语言
- Summer 架构



监控

- Summer 实现



- 告
  - JVM、RPC等
  - 空间告警归类到 Space 类 、QJM切换归类到 DBHA...

NFS Gateway

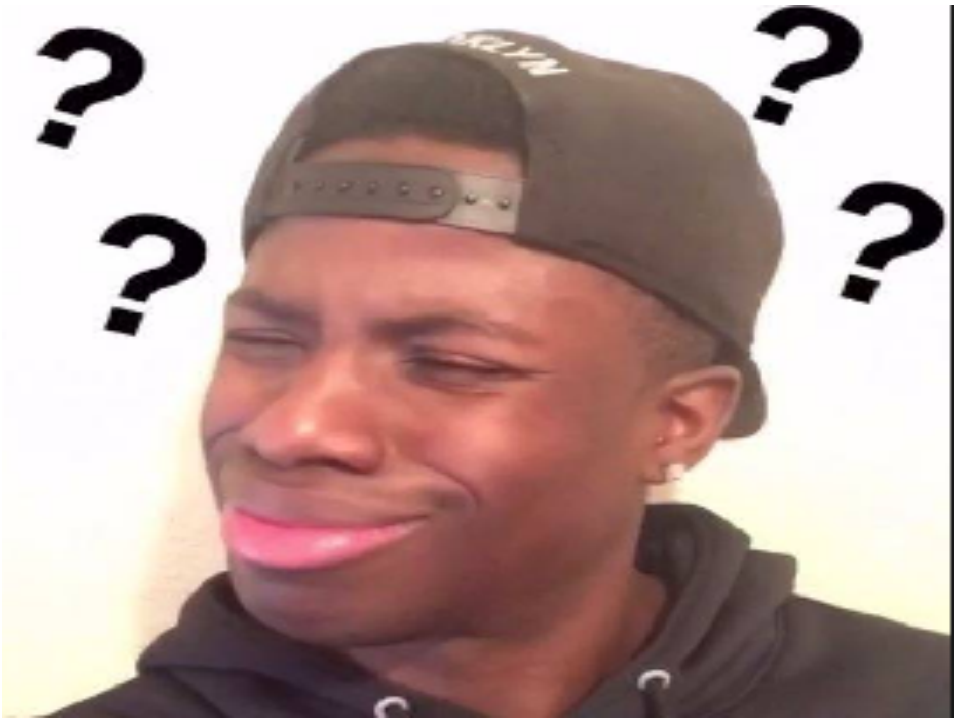
- 打开NFS
  - 修改配置文件，并分发到整个集群
  - root用户启动portmap
  - Hadoop用户启动nfs3

Balancer

- 为什么我的balancer失效？
- Balancer如何才能正常工作?

```
sh xlark_balancer.sh \
    threshold 5 \
   -include \
   10.1.1.0,10.1.1.1,10.1.1.2,10.1.1.3
```

- 最佳实践



集群间数据同步

- 15T 的数据怎么迁移？
- 举个例子

```
hadoop distcp hdfs://1.2.3.5:9000/kafka/data/$1/$2/$3/$4/$i/$2 0 0.data.$3$4$
hdfs://sz-cluster/kafka/data/$1/$2/$3/$4/$i/$j/
```

在线扩容

扩容

- 初始化环境
  - 创建Hadoop用户
  - 配置ssh互信
  - 拷贝namenode文件（包括JDK和Hadoop目录）
  - hosts文件
- 修改namenode slaves include文件，并分发给所有DN
- 启动
  - sh hadoop-daemon.sh start datanode
  - sh yarn-daemon.sh start nodemanager
- 跑Balancer以重平衡数据

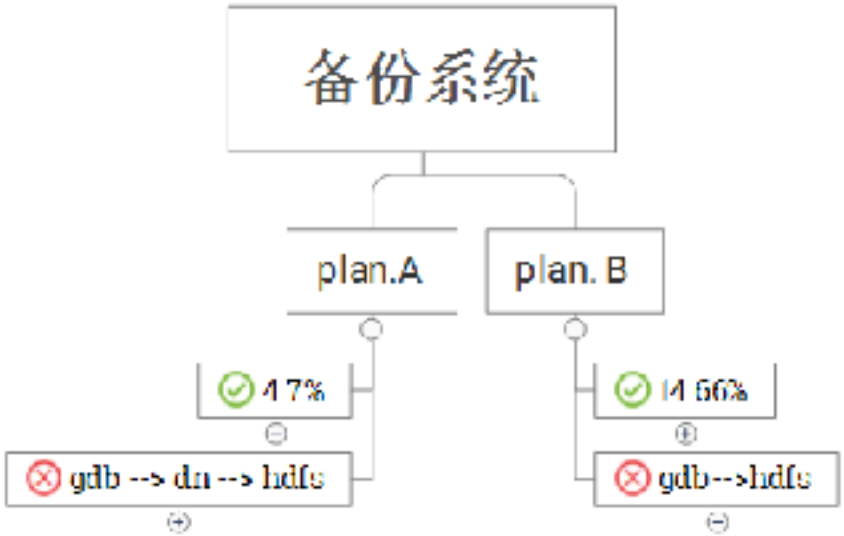
webHDFS

- 开启webHDFS

```
<property>
        <name>dfs.webhdfs.enabled</name>
        <value>true</value>
</property>
```

- 应用

  备份系统日增数据量



Streaming编程

- 原理：Java实现一个包装用户程序的MR程序，该程序负责调用MR Java接口获取key/value对输入，创建一个新的进程启动包装过的用户程序，将数据通过管道传递给包装过的用户处理程序，然后调用MR Java接口将用户程序的输出切分成key/value对输入

- 分布式并行压缩

```
hadoop jar /data/hadoopenv/hadoop/share/hadoop/tools/lib/hadoop streaming 2.6.0 cdh5.4.1.jar \
       -Dmapred.reduce.tasks=0 \
        input hdfs://sz 1/water/input_binlog  \
       -output /user/hadoop/gzipped.log  \
       -mapper "/usr/bin/perl /data/hadoopenv/mp.pl " \
       -inputformat org.apache.hadoop.mapred.lib.NLineInputFormat \
       -file /data/hadoopenv/mp.pl
```

Streaming编程

- map 函数

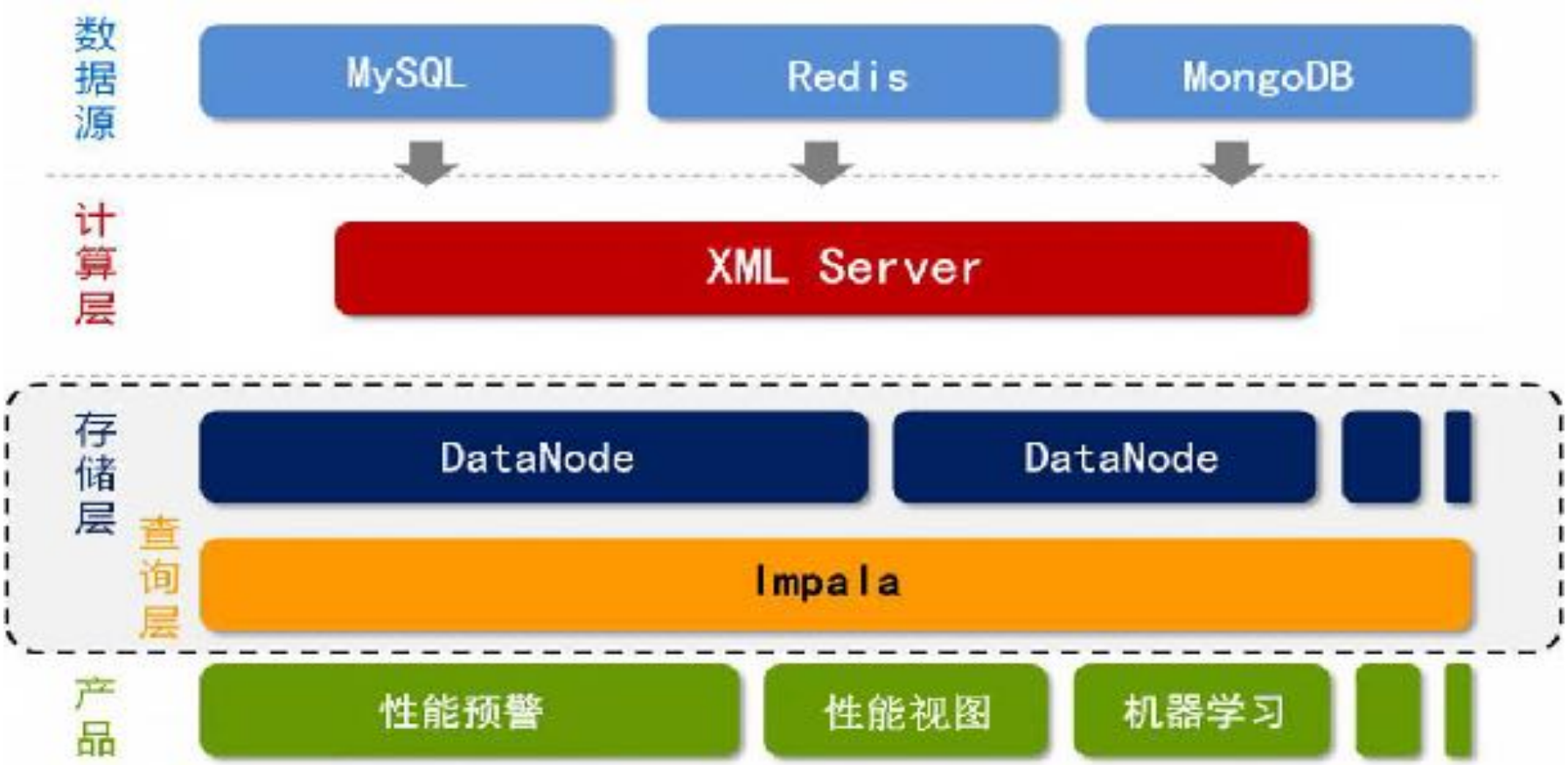
```
#!/usr/bin/perl

use File::Basename;
use strict;
use warnings;


foreach(<>)
{
   chomp;
   print $ ,"\n";
   my $tmpfile = '/data/hadoopenv/mp.log';
   open(my $fh, '>', $tmpfile) or die "Could not open file '$tmpfile' $!";
   my $fname = `echo $_ | awk  F ' ' '{print \$2}'`;
   chomp $fname;
   print $fname,"\n";
   `/data/hadoopenv/hadoop/bin/hdfs dfs -copyToLocal $fname .`;
   my ($name,$path,$suffix) = fileparse($fname);
   `/bin/gzip $name`;
   $name .= ".gz";
   `/data/hadoopenv/hadoop/bin/hdfs dfs -copyFromLocal $name  $path`;
}
```

Streaming编程

- 注意事项
  - 外部可执行程序一定得用绝对路径，否则无法识别，譬如，gzip、perl
  - 程序调试可通过在Map里面执行 print，结果会输出到 --output所指定的目录下
- 收益

  测试6台DN压缩 518个binlog(每个log大概256M)所耗费时间 4 分钟左右

应用


数据源
MySQL
Redis
MongoDB
计算层
XML Server
存储层
DataNode
DataNode
查询层
Impala
产品
性能预警
性能视图
机器学习

应用

- 数据特点
  - 腾讯游戏监控数据的特点是时间序列，其原始数据格式是：
    key (ip+port) + time + value
- 存储方案
  - Module + time + ip%100

设计原则
- 数据打散到更多节点
- 扫描的文件越少越好

- e.g =2016/mm=11/dd=30/tt=80

谢谢大家