

# HBase案例分析

讲师：董西成  
博客：[dongxicheng.org](http://dongxicheng.org)  
微信二维码见右。





1. **HBase 案例分析概述**
2. 时间序列数据库**OpenTSDB**
3. 基于**HBase**的爬虫调度库
4. 基于**HBase**的文档库
5. 银行人民币查询系统
6. 总结



## ➤ HBase 应用场景

- ✓ 随机读或随机写;
- ✓ 大数据上高并发操作, 比如每秒对PB级数据进行上千次操作;
- ✓ 读写均是非常简单的操作, 比如没有join操作。

## ➤ HBase Schema设计

- ✓ Rowkey设计是关键

## ➤ HBase编程

- ✓ 根据语言偏好选择, 可以采用Java, 也可借助thrift采用其他语言。

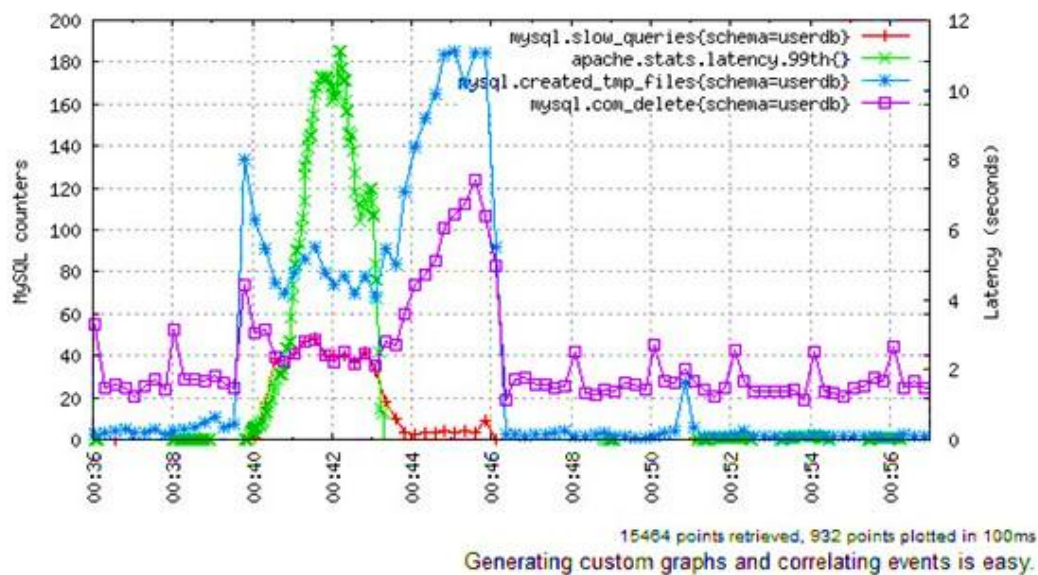


1. Hbase 案例分析概述
2. 时间序列数据库OpenTSDB
3. 基于HBase的爬虫调度库
4. 基于Hbase的文档库
5. 银行人民币查询系统
6. 总结

# OpenTSDB是什么



- 基于Hbase构建的；
- 分布式、可伸缩的时间序列数据库；
- 秒级数据采集所有metrics，支持永久存储，可以做容量规划；
- 可以从大规模的集群（包括集群中的网络设备、操作系统、应用程序）中获取相应的metrics并进行存储、索引以及服务。





[opentsdb-hbasecon.pdf](#)



- 加宽行可增加扫描速度
- 采用组合rowkey，利用数据本地性加快扫描
- 少数宽行并不比多数窄行节省空间
  - ✓ 缩短Column family和column的名字
  - ✓ 合并若干列

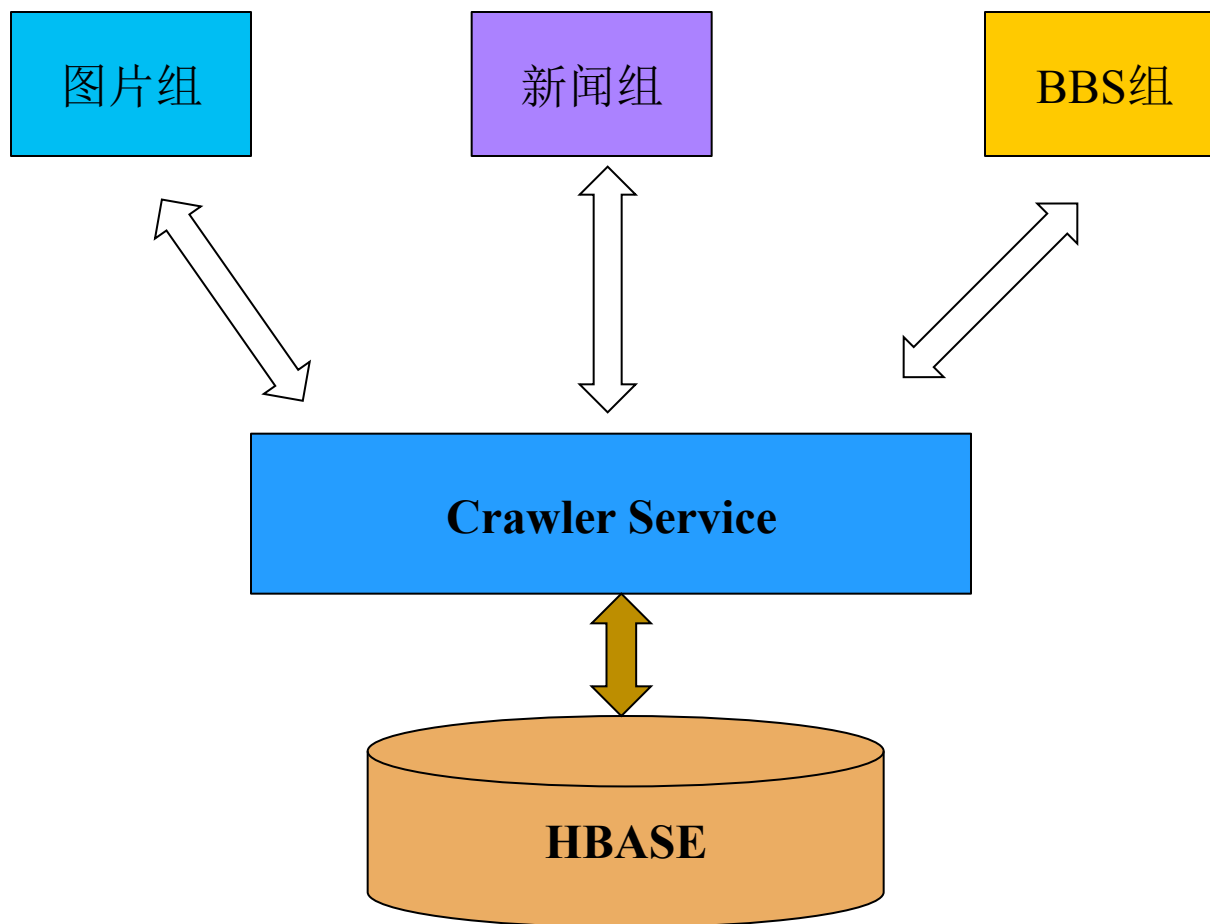




1. Hbase 案例分析概述
2. 时间序列数据库OpenTSDB
3. 基于HBase的爬虫调度库
4. 基于Hbase的文档库
5. 银行人民币查询系统
6. 总结



# 基于HBase的爬虫调度库—背景



# 基于HBase的爬虫调度库—背景



- 搜索引擎的爬虫需要存储、调度待爬取的url
  - ✓ 不同URL对爬虫有实时性、优先级等爬取要求；
- 不同的URL分属于不同的垂直搜索频道
  - ✓ 比如图片搜索、新闻搜索等；
- 不同频道的人向爬虫提交url
  - ✓ 爬虫为不同频道维护了一个url列表。



- 部分垂直频道的URL需满足队列特征，即先进先出
  - ✓ 先插入的URL优先爬取；
- 部分频道的URL有优先级特征，需满足优先级高的URL优先被调度
  - ✓ 优先级可以是自己定义的，也可以是时间戳；
- 不同频道的数据量差异很大
  - ✓ 图片的数据量很大，库存维持在亿级别，新闻则维持在万级别。

# 基于HBase的爬虫调度库—目标



- 多个垂直业务同时调度；
- 按照约定的抓取速度进行调度抓取，不同站点的抓取速度会有不同；
- 天级别进行调度抓取统计。



- 每个垂直频道建两个table: **NAME\_HOST**和**NAME\_URL**
  - ✓ NAME\_HOST保存host列表, 比如IMAGE\_HOST
  - ✓ NAME\_URL保存每个host对应的url列表, 比如IMAGE\_URL
- NAME\_HOST表
  - ✓ Rowkey为hostid, column中存储host、抓取速度、扫描时间等信息
- NAME\_URL表
  - ✓ Rowkey为Hostid+PID+UrlID
  - ✓ 先按hostid排序, 再按照pid (优先级) 排序
  - ✓ 可通过scan操作获取一个host下所有的url列表



- 不同进程从不同的业务从表中扫描出**URL**用于调度抓取，按照不同**Host**的抓取速度，以及扫描的时间间隔进行数量提取和删除
  - ✓ 比如，对于www.dongxicheng.org 这个站点是5s抓取1个链接，扫描进程中的扫描线程是5分钟一次，则每次扫描会扫描出最多60个url用于抓取。
- 记录入库和出库信息，发送统计报表

# 我们学到了什么？



- 充分利用rowkey有序的特性
- 在rowkey中融入有用字段
  - ✓ Hostid+PID+UrlID
- 不要直接使用字符串作为rowkey，而是使用ID（整数）
  - ✓ MD5
  - ✓ Hash





1. Hbase 案例分析概述
2. 时间序列数据库OpenTSDB
3. 基于HBase的爬虫调度库
4. 基于Hbase的文档库
5. 银行人民币查询系统
6. 总结

# 基于HBase的文档库—背景



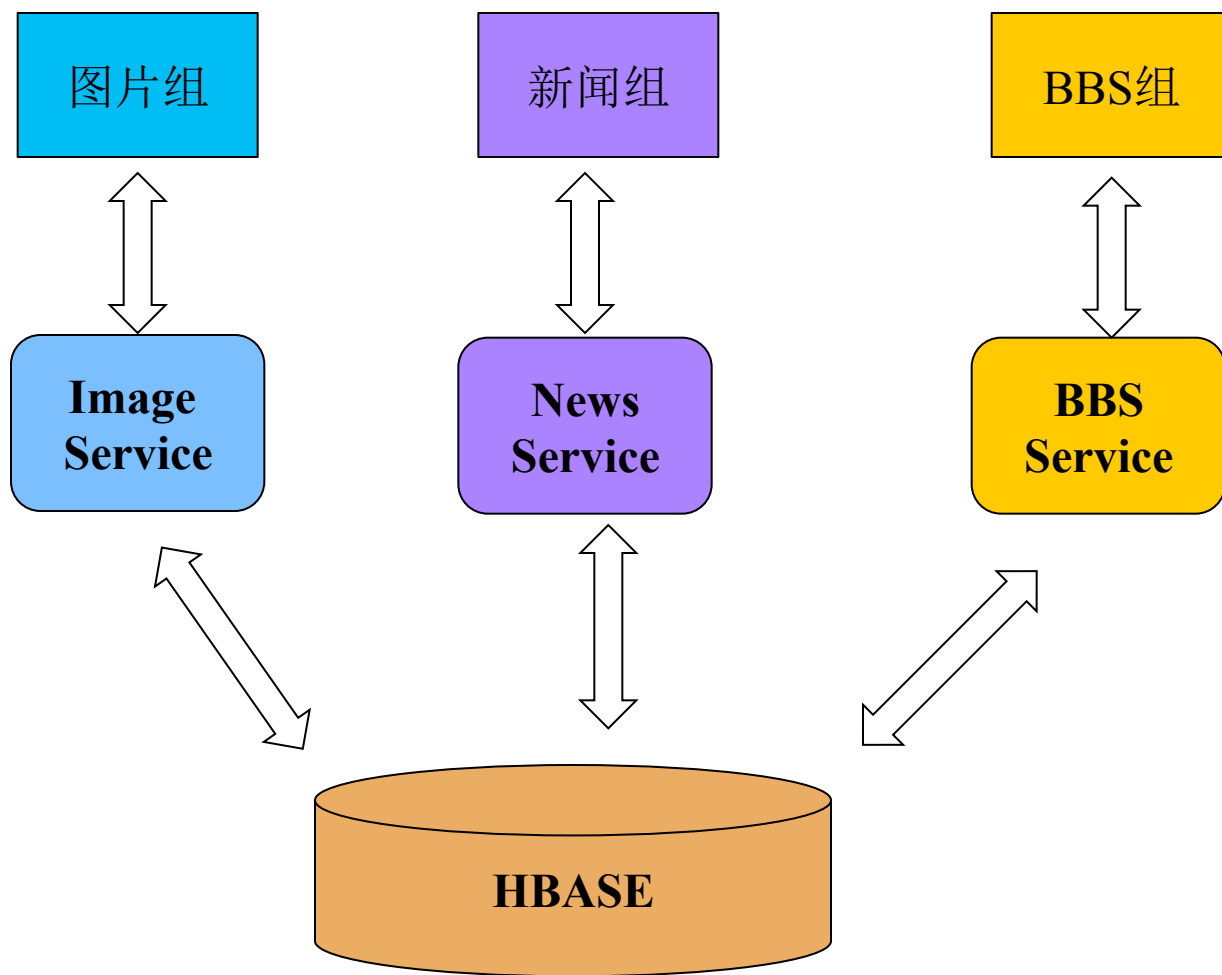
- 存储网页分析后的精细化数据
  - ✓ 比如论坛、博客网页分析数据；

# 基于HBase的文档库—基本特性



- 数据格式多样，例如BBS和Blog的数据格式，字段不同；
- 实时读取和写入，读取量小于写入（包含更新），读取后可能多次发生人工修改后写回；
- 每天写入的数据量在2KW条数据左右；
- 数据之间有关联，例如Blog的评论和正文之间存储的关联。

# HBase解决方案





- 每类数据一个**table**，包含两个**Column famlily**；
  - ✓ 基础数据存在BI中，比如网页标题，内容，创建时间等；
  - ✓ 动态数据存在DI中，比如浏览量、情感倾向等
- 数据写入的时候，会先经过自己写的**Server**，将数据拆分，批量写入**HBase**；
- 数据读取的时候，分单条记录读取，**Server**负责组装一个**Rowkey**对应的信息；批量扫描的时候发生在**Blog**关联的评论，评论的**RowKey**前缀是这个**Blog**的**rowkey**；
- 部署在**8**台主机上，每台主机被分配的磁盘空间是**2T**。



1. Hbase 案例分析概述
2. 时间序列数据库OpenTSDB
3. 基于HBase的爬虫调度库
4. 基于Hbase的文档库
5. 银行人民币查询系统
6. 总结

# 银行人民币查询系统—背景



- 银行内部人民币数目每天将进行结算；
- 数据记录规模十亿到百亿条，后续可能会更多；
- 人民币分散在各个设备上，比如ATM、清分机、点钞机等；
- 作为一个尝试，想基于Hadoop实现一个人民币查询系统。
  - ✓ 比如根据人民币冠字号查询对应信息

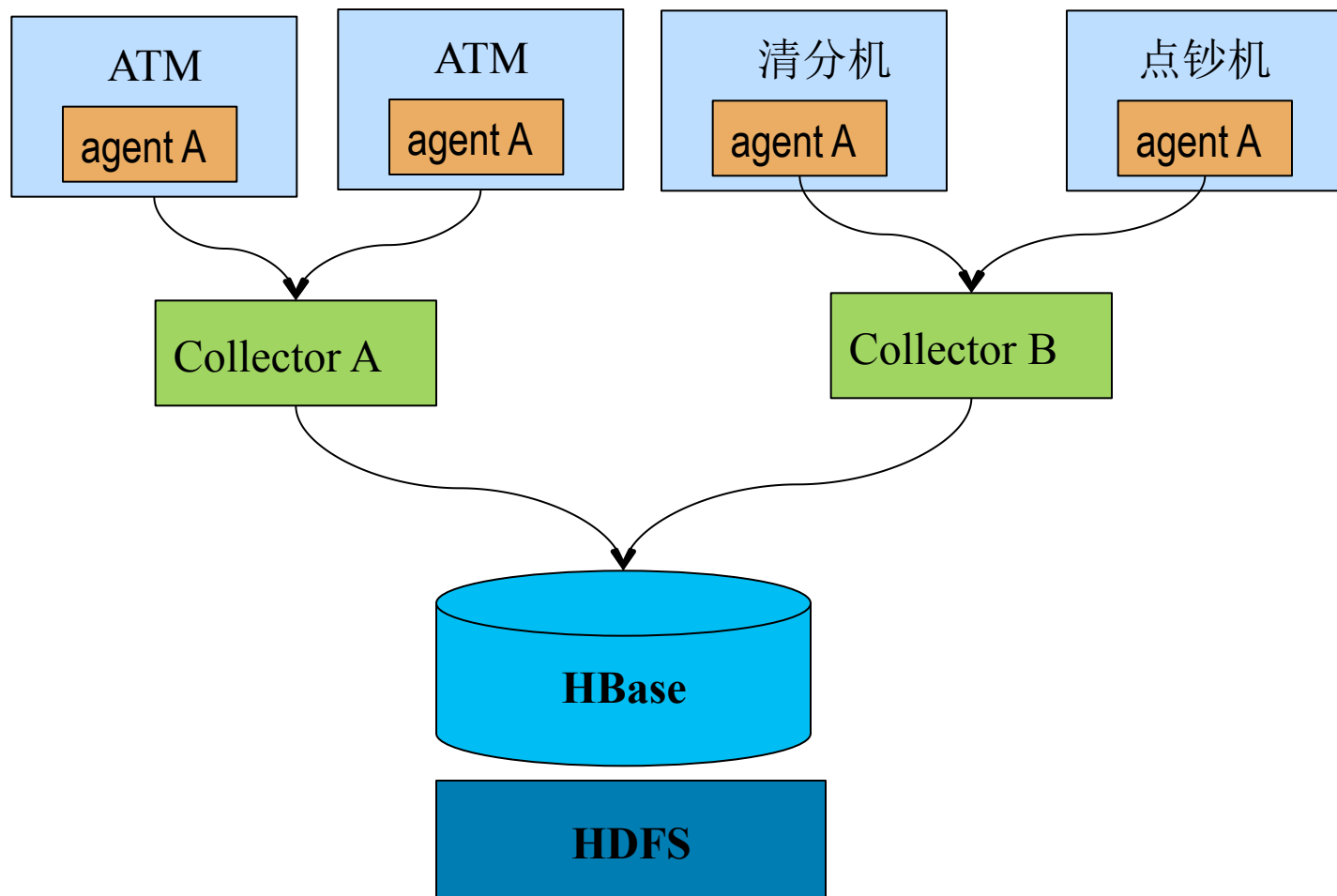


# 银行人民币查询系统—目标



- 对各个设备（比如ATM、点钞机）上的数据进行采集
  - ✓ 数据采集相应及时；
  - ✓ 保证数据不能有遗漏（可靠性）；
- 支持数据有效存储及快速检索
  - ✓ 比如X秒返回结果等；
- 系统具有良好的扩展性
  - ✓ 接入设备会持续增长；
  - ✓ 数据量会持续增长。

# Flume+HBase解决方案





## ➤ Cloudera开源的日志收集系统

## ➤ Flume特点

- ✓ 分布式
- ✓ 高可靠性
- ✓ 高容错性
- ✓ 易于定制与扩展



- 该系统只需按照冠字号查询人民币信息
- 直接用冠字号作为rowkey可能造成访问热点，无法利用HBase并行特点
- 可将冠字号作Hash值，或者逆转冠字号

比如：XN15115812→21851151NX

# 我们学到了什么？



- **Hbase**只是一个数据库，通常需要跟其他系统结合使用
- 避免产生访问热点
  - ✓ 方案：**Hash**、**翻转ID**或**组合rowkey**
  - ✓ 避免采用直接使用时间作为**rowkey**（大部分情况）



1. **Hbase** 案例分析概述
2. 时间序列数据库**OpenTSDB**
3. 基于**HBase**的爬虫调度库
4. 基于**Hbase**的文档库
5. 银行人民币查询系统
6. 总结



## ➤ HBase 应用场景

- ✓ 随机读或随机写；
- ✓ 大数据上高并发操作，比如每秒对PB级数据进行上千次操作；
- ✓ 读写均是非常简单的操作，比如没有join操作。

## ➤ HBase Schema设计

- ✓ Rowkey设计关键点
  - 避免Rowkey热点；
  - 充分利用Rowkey有序特点；
  - 可使用多个字段组合成Rowkey。

## ➤ HBase编程

- ✓ 根据语言偏好选择，可以采用Java，也可借助thrift采用其他语言。