小象科技
ChinaHadoop.cn

Hadoop数据收集与入库系统 Flume与Sqoop

讲师：董西成

主要内容

小象科技
ChinaHadoop.cn

1. 背景介绍
2. Hadoop数据收集系统
3. 传统数据库与Hadoop间数据同步
4. 总结

主要内容

小象科技
ChinaHadoop.cn

1. 背景介绍
2. Hadoop数据收集系统
3. 传统数据库与Hadoop间数据同步
4. 总结

背景介绍

小象科技 ChinaHadoop.cn

- **Hadoop提供了一个中央化的存储系统：**
  - ✓ 有利于进行集中式的数据分析与数据共享
- **Hadoop对存储格式没有要求：**
  - ✓ 用户访问日志；
  - ✓ 产品信息
  - ✓ 网页数据等
- **如何将数据存入Hadoop：**
  - ✓ 数据分散在各个离散的设备上
  - ✓ 数据保存在传统的存储设备和系统中

常见的两种数据来源

小象科技 ChinaHadoop.cn

- 分散的数据源：
  - ✓机器产生的数据；
  - ✓用户访问日志；
  - ✓用户购买日志；
- 传统系统中的数据：
  - ✓传统关系型数据库:MySQL、Oracle等；
  - ✓磁盘阵列；
  - ✓磁带.

Hadoop收集和入库基本要求

小象科技 ChinaHadoop.cn

- 分布式
  - ✓ 数据源多样化
  - ✓ 数据源分散
- 可靠性
  - ✓ 保证不丢数据
  - ✓ 允许丢部分数据
- 可扩展
  - ✓ 数据源可能会不断增加
- 通过并行提高性能

常见的Hadoop收集与入库系统

小象科技 ChinaHadoop.cn

- 数据收集
  - ✓ Flume
  - ✓ Kafka
  - ✓ Scribe
- 传统数据库与Hadoop同步
  - ✓ Sqoop

主要内容

小象科技
ChinaHadoop.cn

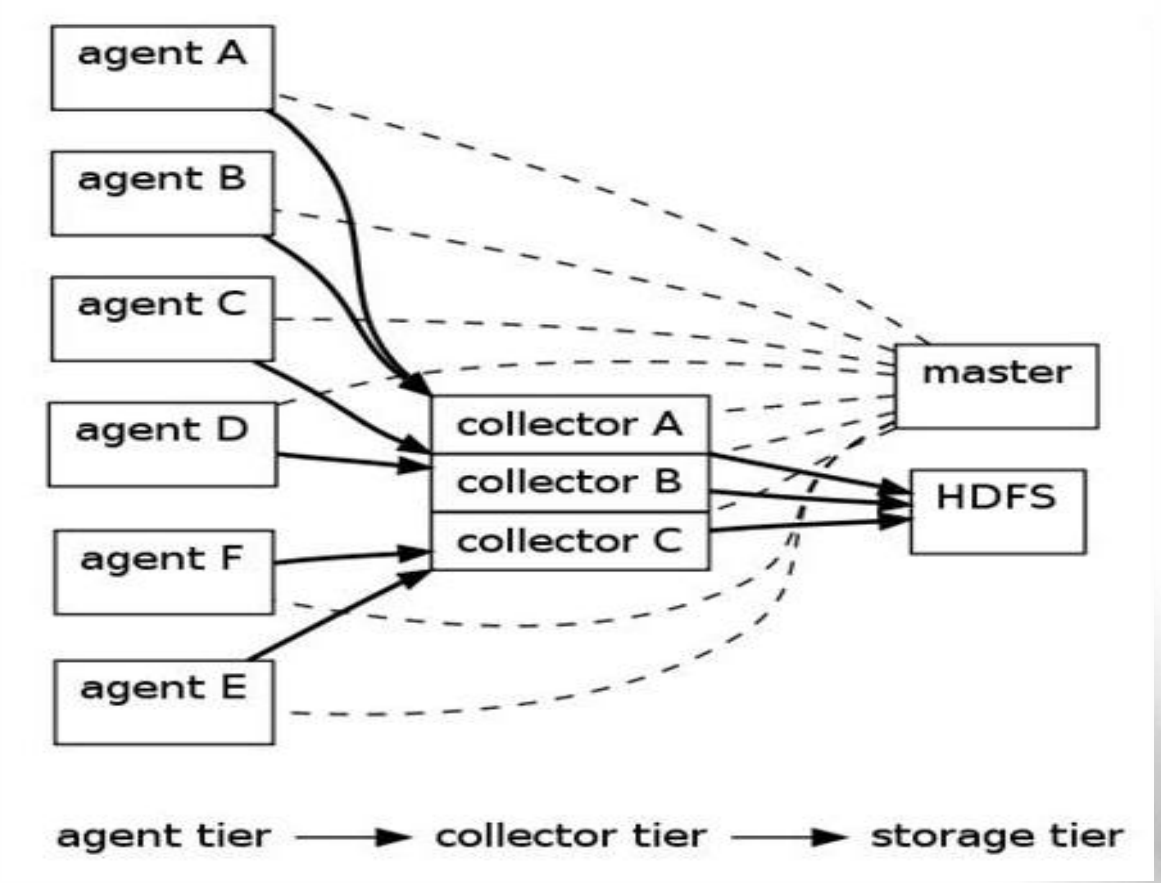
1. 背景介绍
2. Hadoop数据收集系统
3. 传统数据库与Hadoop间数据同步
4. 总结

Hadoop数据收集系统—Flume

小象科技 ChinaHadoop.cn

- **Flume OG**
  - ✓ OG：“Original Generation”
  - ✓ 0.9.x或cdh3以及更早版本
  - ✓ 由agent、collector、master等组件构成
- **Flume NG**
  - ✓ NG：“Next/New Generation”
  - ✓ 1.x或cdh4以及之后的版本
  - ✓ 由Agent、Client等组件构成
- **为什么要推出NG版本**
  - ✓ 精简代码
  - ✓ 架构简化

Flume OG基本架构



Flume OG基本架构

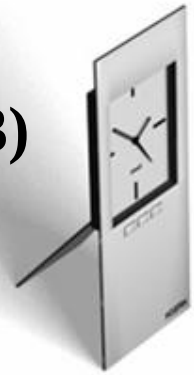
小象科技 ChinaHadoop.cn

角色	简介
Master	Master 负责配置及通信管理，是集群的控制器
Collector	Collector 用于对数据进行聚合（数据收集器），往往会产生一个更大的数据流，然后加载到 storage（存储）中
Agent	Agent 用于采集数据，Agent 是 flume 中产生数据流的地方，同时 Agent 会将产生的数据流传输到 Collector

Agent

小象科技 ChinaHadoop.cn

- 用于采集数据
- 数据流产生的地方
- 通常由source和sink两部分组成
  - ✓ Source用于获取数据，可从文本文件，syslog，HTTP等获取数据；
  - ✓ Sink将Source获得的数据进一步传输给后面的Collector。
- Flume自带了很多source和sink实现
  - ✓ syslogTcp(5140) | agentSink("localhost",35853)
  - ✓ tail("/etc/services") | agentSink("localhost",35853)



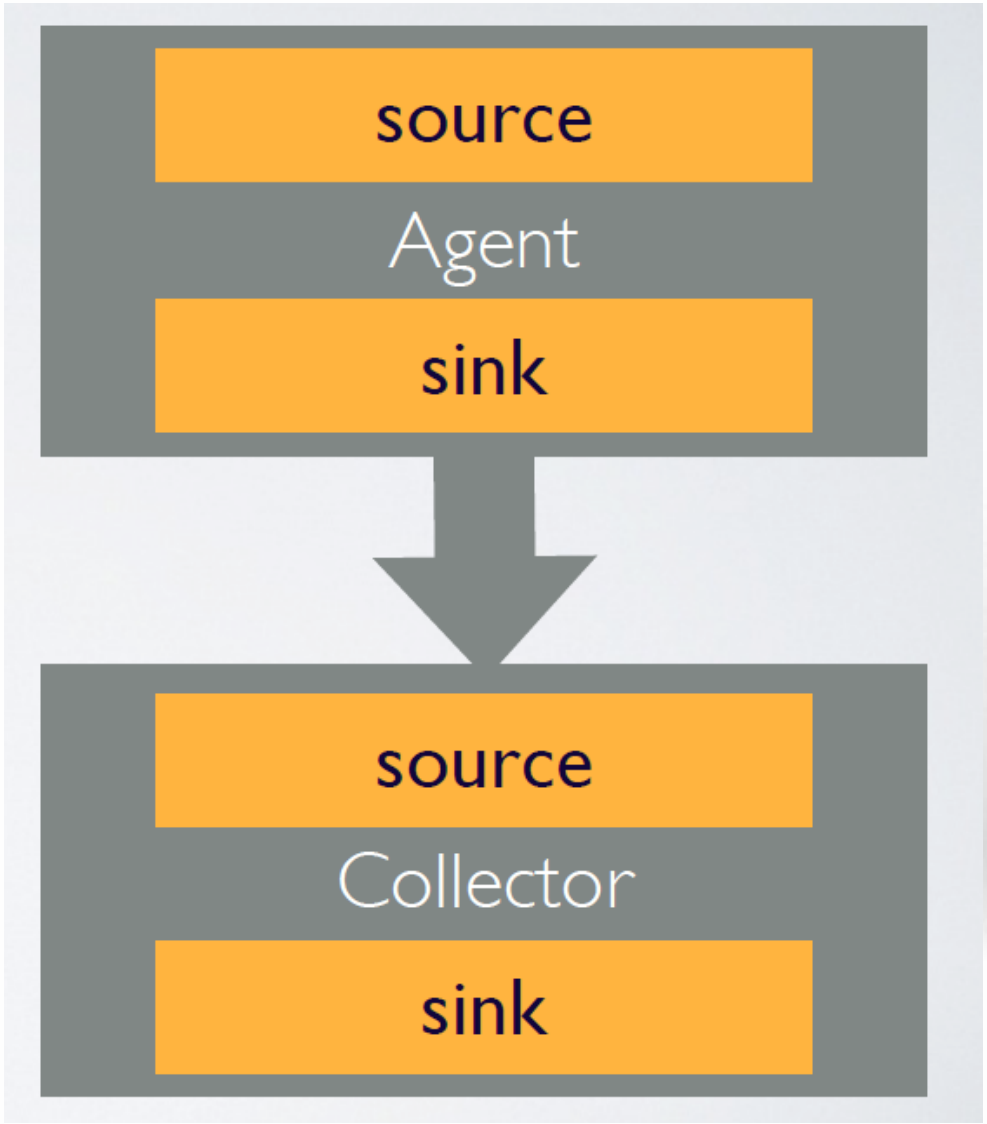
Collector

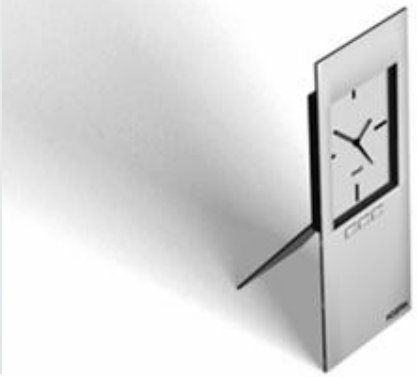
小象科技 ChinaHadoop.cn

- 汇总多个Agent结果
- 将汇总结果导入后端存储系统，比如HDFS，HBase
- Flume自带了很多collector实现
  - ✓collectorSource(35853) | console
  - ✓CollectorSource(35853) |
    collectorSink("file:///tmp/flume/collected", "syslog");
  - ✓collectorSource(35853) |
    collectorSink("hdfs://namenode/user/flume/ ","syslog");

Agent与Collector对应关系

小象科技
ChinaHadoop.cn

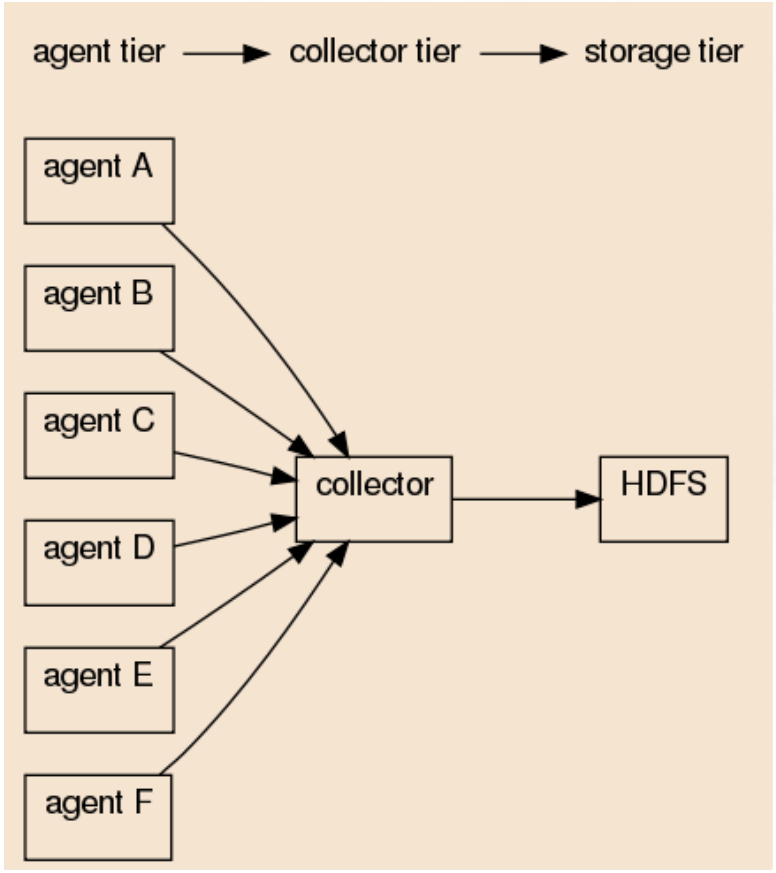


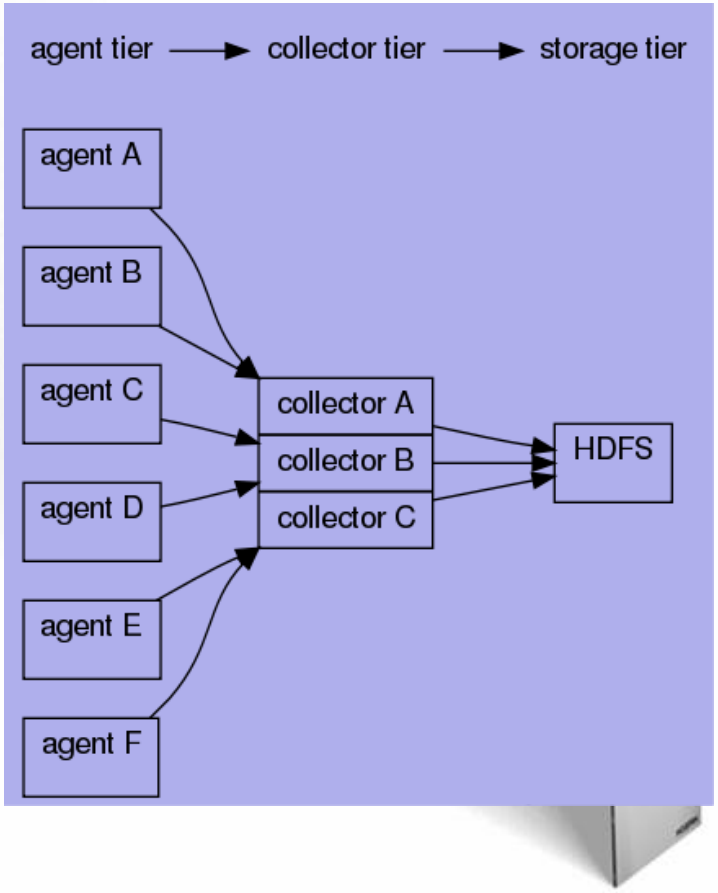


Agent与Collector对应关系

小象科技 ChinaHadoop.cn

- 可手动指定，也可自动匹配
- 自动匹配的情况下，master会平衡collector之间的负载。





问题：为什么引入Collector？

小象科技 ChinaHadoop.cn

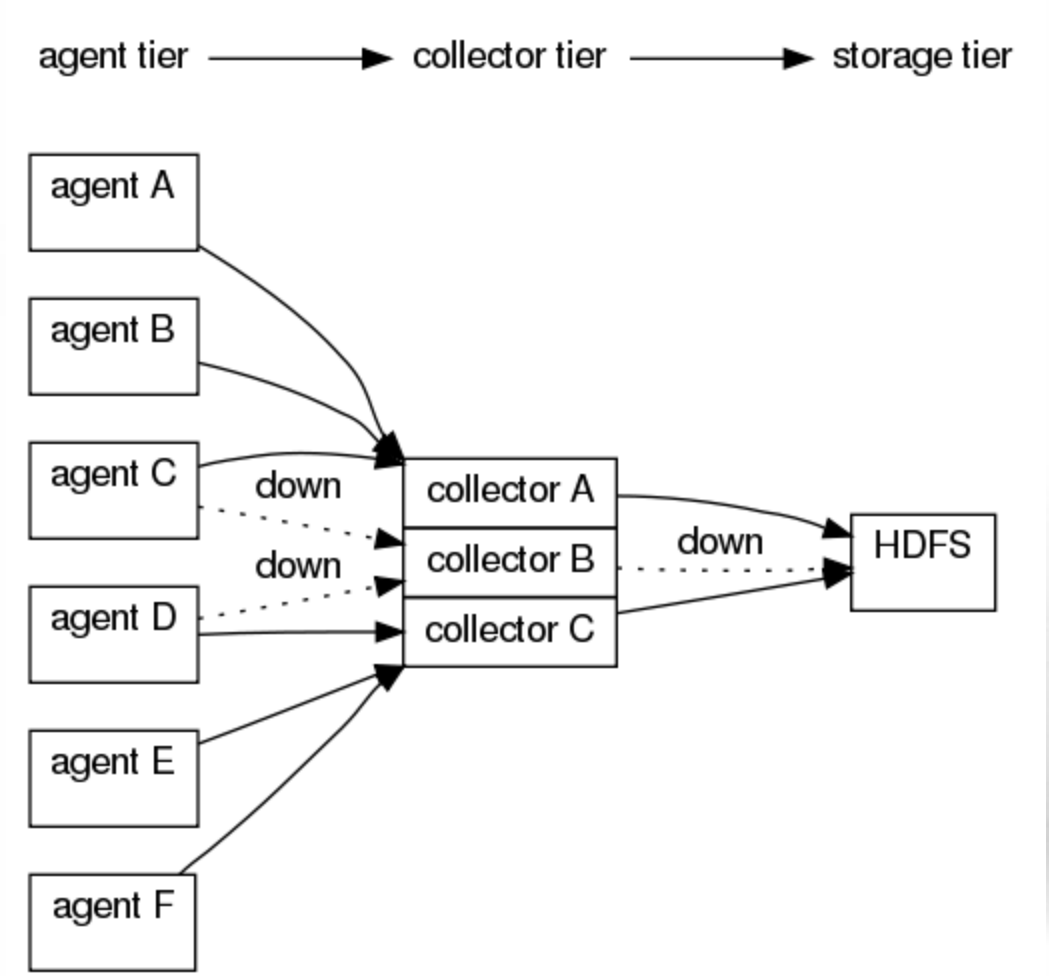
- **对Agent数据进行汇总，避免产生过多小文件；**
- **避免多个agent连接对Hadoop造成过大压力；**
- **中间件，屏蔽agent和hadoop间的异构性。**

Master

小象科技 ChinaHadoop.cn

- 管理协调 agent 和collector的配置信息；
- Flume集群的控制器；
- 跟踪数据流的最后确认信息，并通知agent；
- 通常需配置多个master以防止单点故障；
- 借助zookeeper管理管理多Master。

容错机制


agent tier
collector tier
storage tier
agent A
agent B
agent C
agent D
agent E
agent F
down
down
collector A
collector B
collector C
down
HDFS

三种可靠性级别

小象科技 ChinaHadoop.cn

- agentE2ESink[("machine"[,port])]

  agent收到确认消息才认为数据发送成功，否则重试.

- agentDFOSink[("machine"[,port])]

  当agent发现在collector操作失败的时候，agent写入到本地硬盘上，当collctor恢复后，再重新发送数据。

- agentBESink[("machine"[,port])]

  效率最好，agent不写入到本地任何数据，如果在collector 发现处理失败，直接删除消息。

构建基于Flume的数据收集系统

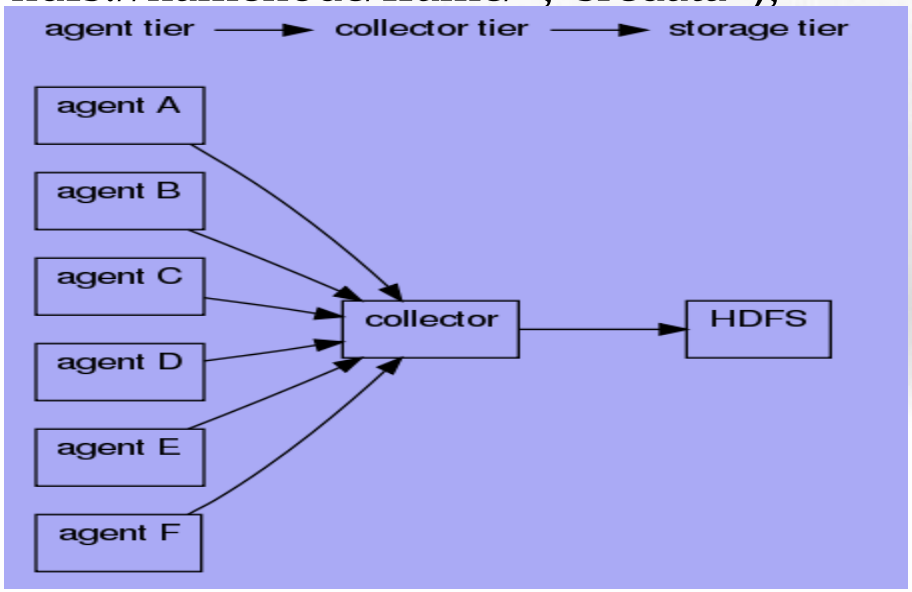
小象科技 ChinaHadoop.cn

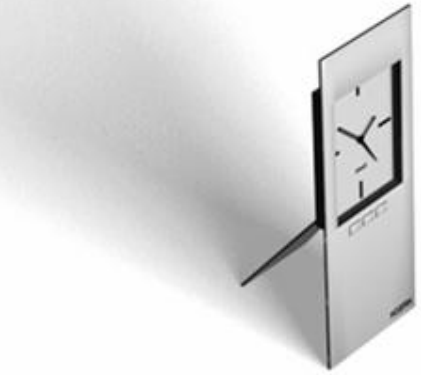
- Agent和Collector均可以动态配置
- 可通过命令行或Web界面配置
- 命令行配置
  - ✓ 在已经启动的master节点上，依次输入”flume shell”➔”connect localhost ”
    如执行 exec config a1 ‘tailDir(“/data/logfile”)’ ‘agentSink’
- Web界面
  - ✓ 选中节点，填写source、sink等信息

常用架构举例—拓扑1

小象科技 ChinaHadoop.cn

```
agentA : tail(“/ngnix/logs”) | agentSink("collector",35853);
agentB : tail(“/ngnix/logs”) | agentSink("collector",35853);
agentC : tail(“/ngnix/logs”) | agentSink("collector",35853);
agentD : tail(“/ngnix/logs”) | agentSink("collector",35853);
agentE : tail(“/ngnix/logs”) | agentSink("collector",35853);
agentF : tail(“/ngnix/logs”) | agentSink("collector",35853);
collector : collectorSource(35853) |
        collectorSink("hdfs://namenode/flume/","srcdata");
```

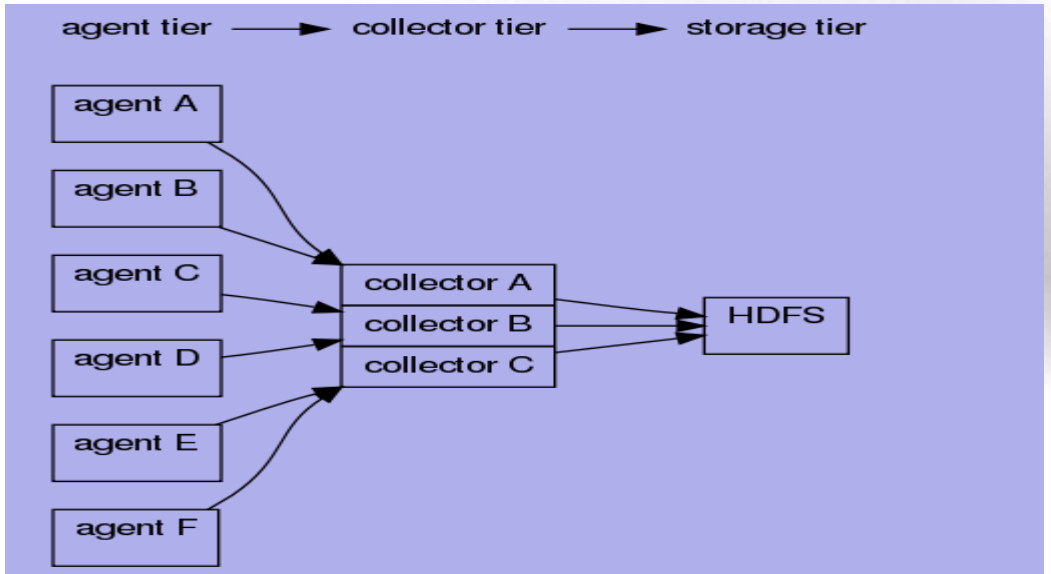




常用架构举例—拓扑2

小象科技 ChinaHadoop.cn

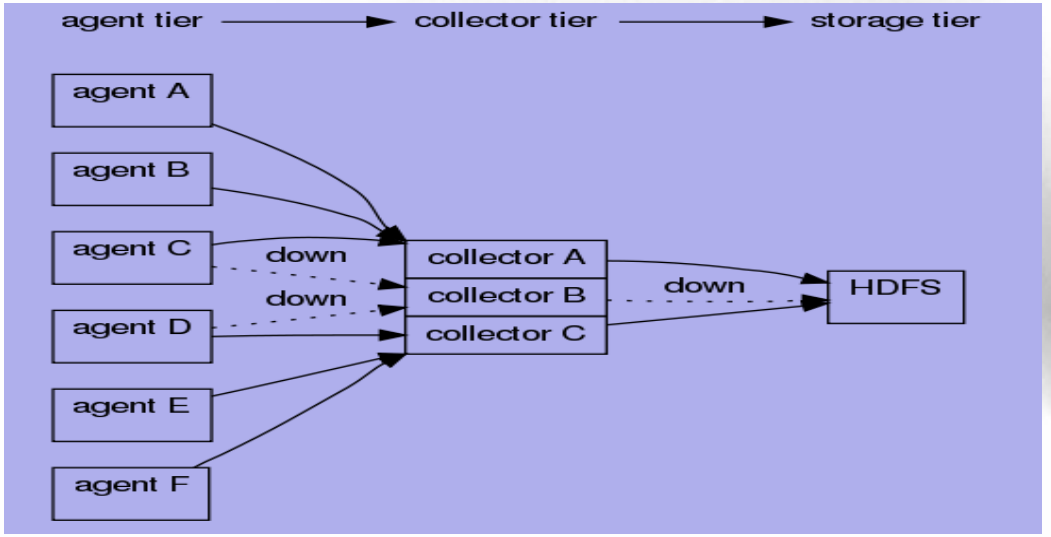
```
agentA : src | agentE2ESink("collectorA",35853);
agentB : src | agentE2ESink("collectorA",35853);
agentC : src | agentE2ESink("collectorB",35853);
agentD : src | agentE2ESink("collectorB",35853);
agentE : src | agentE2ESink("collectorC",35853);
agentF : src | agentE2ESink("collectorC",35853);
collectorA : collectorSource(35853) | collectorSink("hdfs://...","src");
collectorB : collectorSource(35853) | collectorSink("hdfs://...","src");
collectorC : collectorSource(35853) | collectorSink("hdfs://...","src");
```



常用架构举例—拓扑3

小象科技 ChinaHadoop.cn

```
agentA : src | agentE2EChain("collectorA:35853","collectorB:35853");
agentB : src | agentE2EChain("collectorA:35853","collectorC:35853");
agentC : src | agentE2EChain("collectorB:35853","collectorA:35853");
agentD : src | agentE2EChain("collectorB:35853","collectorC:35853");
agentE : src | agentE2EChain("collectorC:35853","collectorA:35853");
agentF : src | agentE2EChain("collectorC:35853","collectorB:35853");
collectorA : collectorSource(35853) | collectorSink("hdfs://...","src");
collectorB : collectorSource(35853) | collectorSink("hdfs://...","src");
collectorC : collectorSource(35853) | collectorSink("hdfs://...","src");
```



主要内容

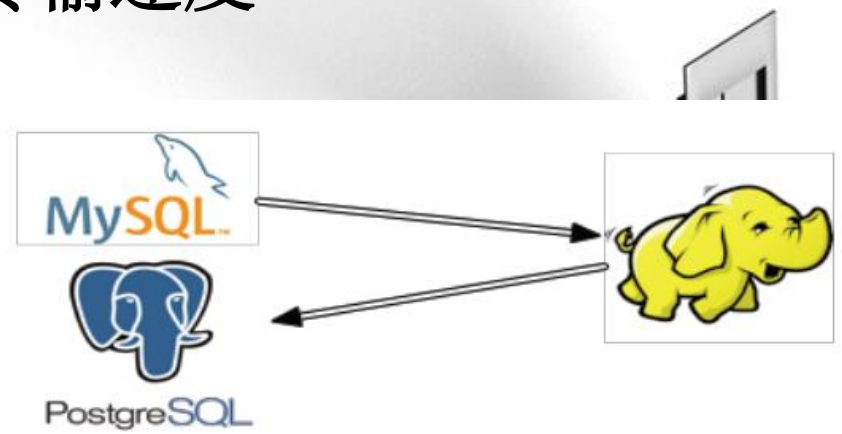
小象科技
ChinaHadoop.cn

1. 背景介绍
2. Hadoop数据收集系统
3. 传统数据库与Hadoop间数据同步
4. 总结

Sqoop是什么

小象科技 ChinaHadoop.cn

- Sqoop：SQL-to-Hadoop
- **连接 传统关系型数据库 和 Hadoop 的桥梁**
  - ✓把关系型数据库的数据导入到 Hadoop 系统 (如 HDFS HBase 和 Hive) 中；
  - ✓把数据从 Hadoop 系统里抽取并导出到关系型数据库里。
- **利用MapReduce加快数据传输速度**
- **批处理方式进行数据传输**

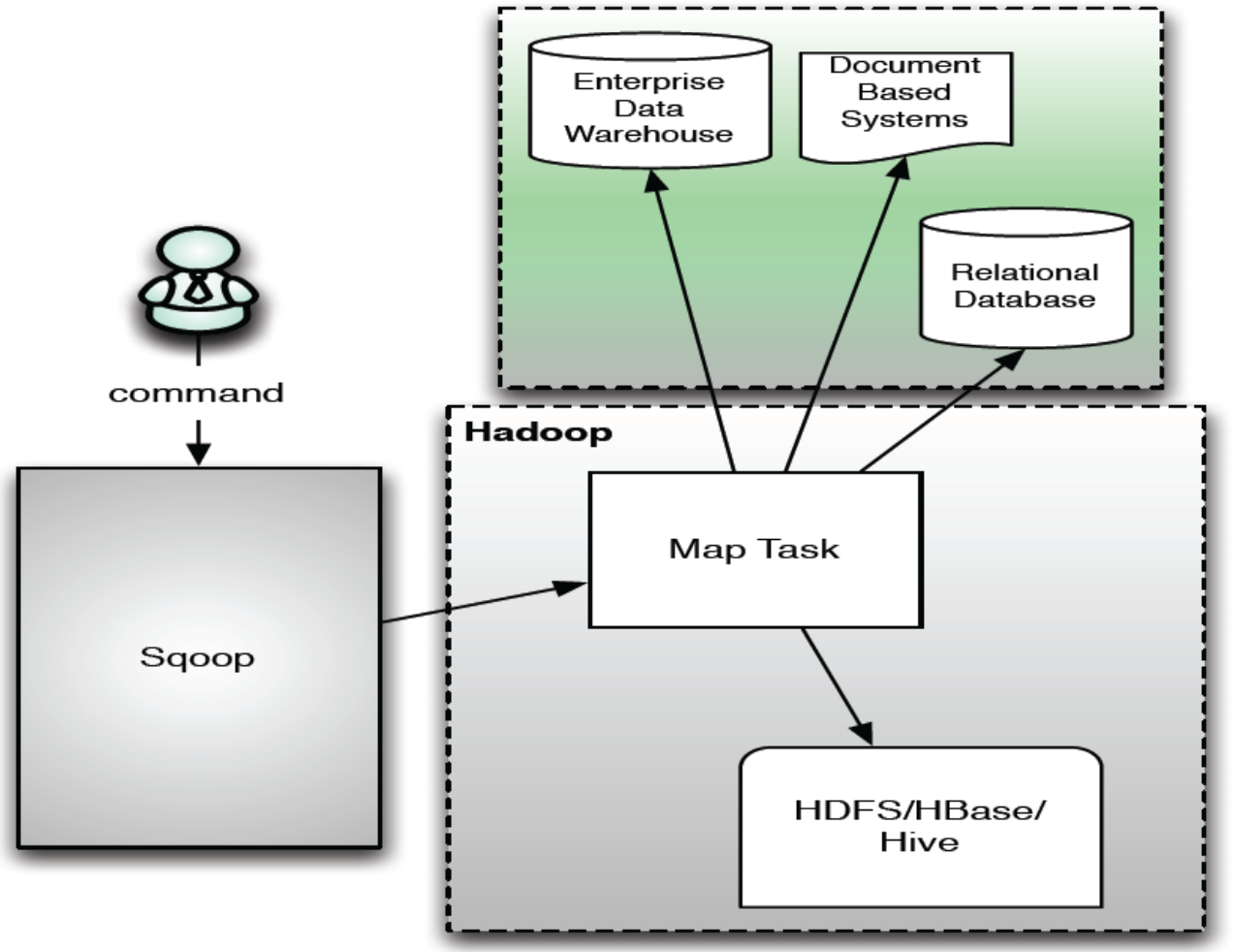


Sqoop优势

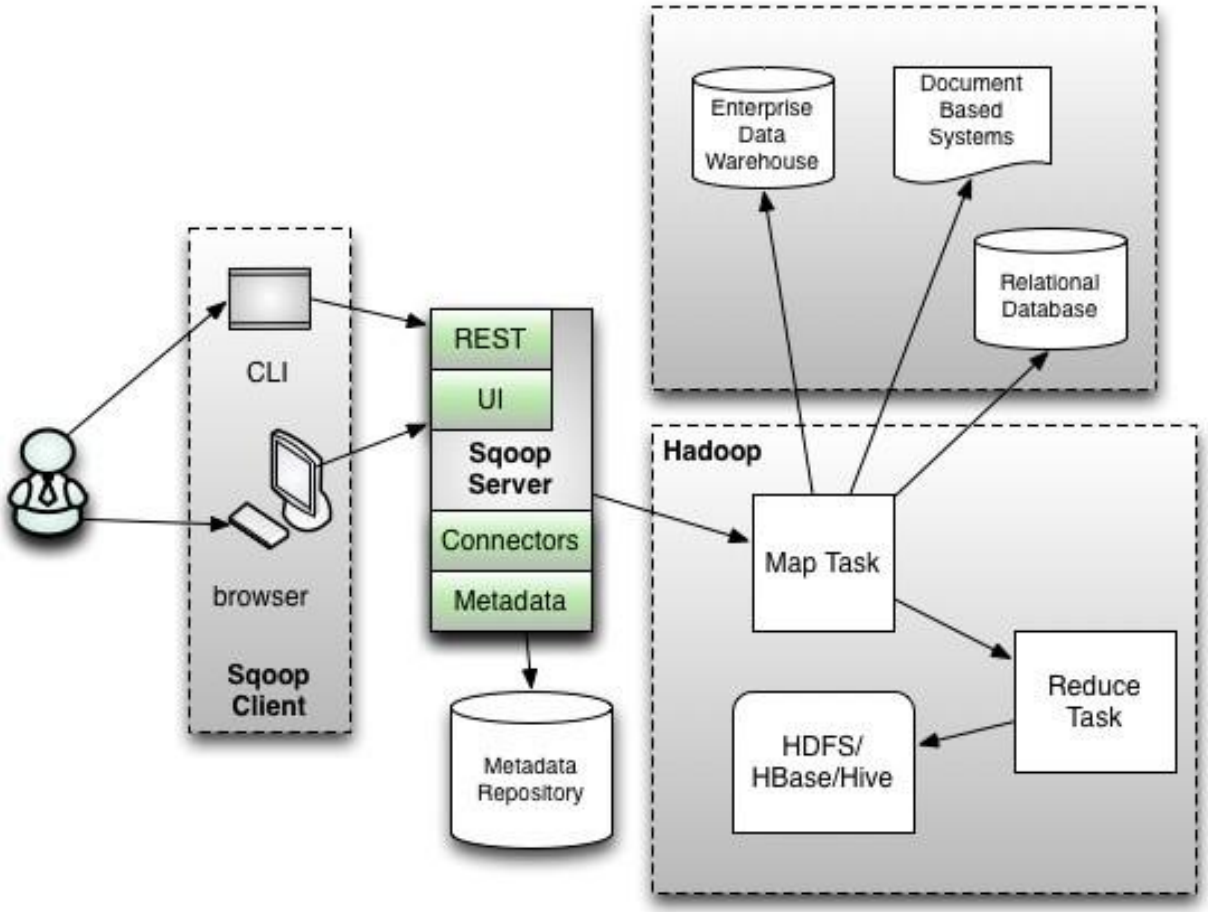
小象科技 ChinaHadoop.cn

- 高效、可控地利用资源
  - ✓ 任务并行度，超时时间等
- 数据类型映射与转换
  - ✓ 可自动进行，用户也可自定义
- 支持多种数据库
  - ✓ **MySQL**
  - ✓ **Oracle**
  - ✓ **PostgreSQL**

Sqoop1架构


Enterprise Data Warehouse
Document Based Systems
Relational Database
command
Hadoop
Map Task
Sqoop
HDFS/HBase/ Hive

Sqoop2架构


Enterprise Data Warehouse
Document Based Systems
Relational Database
CLI
REST
UI
Sqoop Server
Connectors
Metadata
browser
Sqoop Client
Metadata Repository
Hadoop
Map Task
Reduce Task
HDFS/ HBase/Hive

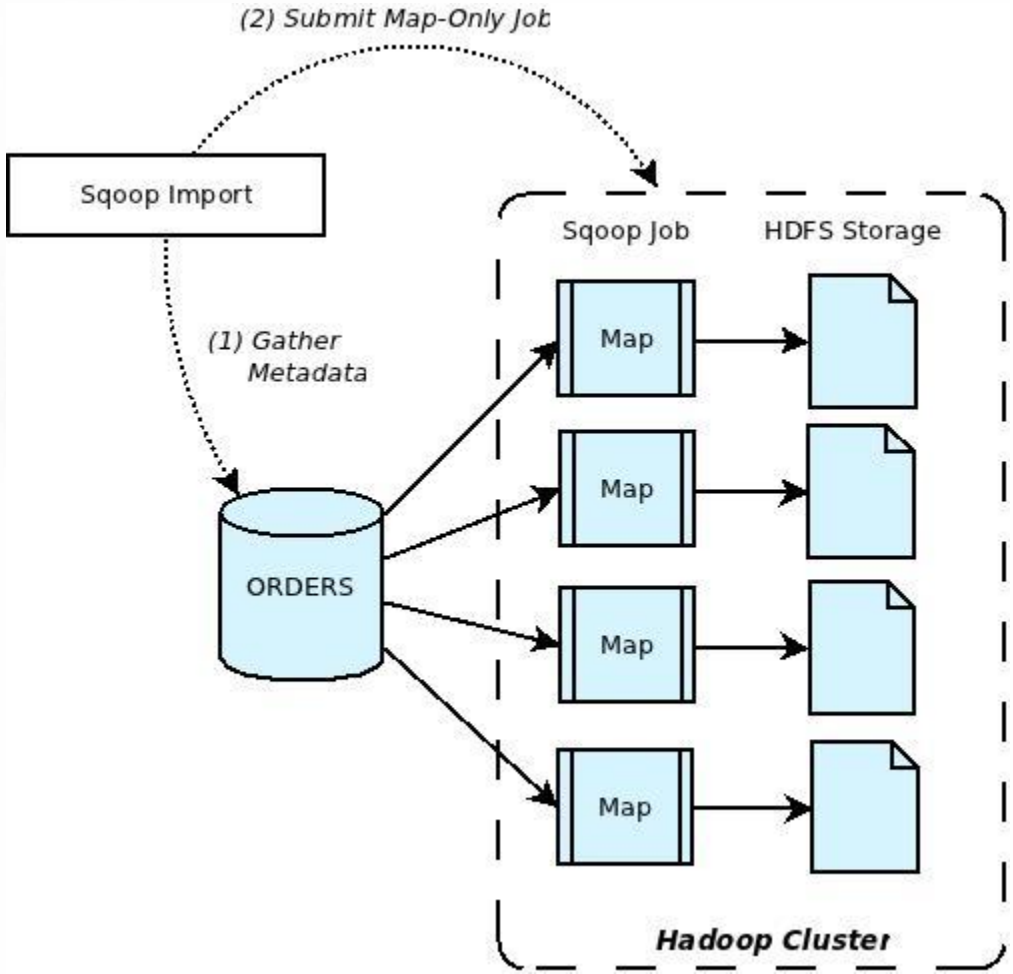
Sqoop import

小象科技 ChinaHadoop.cn

- 将数据从关系型数据库导入Hadoop中

步骤1：Sqoop与数据库Server通信，获取数据库表的元数据信息；

步骤2：Sqoop启动一个Map-Only的MR作业，利用元数据信息并行将数据写入Hadoop。



Sqoop import使用

小象科技 ChinaHadoop.cn

```
sqoop import \
    --connect jdbc:mysql://mysql.example.com/sqoop \
    --username sqoop \
    --password sqoop \
    --table cities
```

- **--connnect: 指定JDBC URL**
- **--username/password：mysql数据库的用户名**
- **--table：要读取的数据库表**

bin/hadoop fs -cat cities/part-m-*

1,USA,Palo Alto
2,Czech Republic,Brno
3,USA,Sunnyvale

Sqoop import示例

小象科技 ChinaHadoop.cn

```
sqoop import \
    --connect jdbc:mysql://mysql.example.com/sqoop \
    --username sqoop \
    --password sqoop \
    --table cities \
    --target-dir /etl/input/cities
sqoop import \
    --connect jdbc:mysql://mysql.example.com/sqoop \
    --username sqoop \
    --password sqoop \
    --table cities \
    --where "country = 'USA'"
```

Sqoop import示例

小象科技 ChinaHadoop.cn

```
sqoop import \
    --connect jdbc:mysql://mysql.example.com/sqoop \
    --username sqoop \
    --password sqoop \
    --table cities \
    --as-sequencefile
sqoop import \
    --connect jdbc:mysql://mysql.example.com/sqoop \
    --username sqoop \
    --password sqoop \
    --table cities \
    --num-mappers 10
```

Sqoop import—导入多个表

小象科技 ChinaHadoop.cn

```
sqoop import \
  --connect jdbc:mysql://mysql.example.com/sqoop \
  --username sqoop \
  --password sqoop \
  --query 'SELECT normcities.id, \
                countries.country, \
                normcities.city \
                FROM normcities \
                JOIN countries USING(country_id) \
                WHERE $CONDITIONS' \
--split-by id \
--target-dir cities
```

Sqoop import增量导入

小象科技 ChinaHadoop.cn

```
sqoop import \
    --connect jdbc:mysql://mysql.example.com/sqoop \
    --username sqoop \
    --password sqoop \
    --table visits \
    --incremental append \
    --check-column id \
    --last-value 1
```

Sqoop import增量导入（一）

小象科技 ChinaHadoop.cn

```
sqoop import \
    --connect jdbc:mysql://mysql.example.com/sqoop \
    --username sqoop \
    --password sqoop \
    --table visits \
    --incremental append \
    --check-column id \
    --last-value 1
```

- 适用于数据每次被追加到数据库中，而已有数据不变的情况；
- 仅导入id这一列值大于1的记录。

Sqoop import增量导入（二）

小象科技 ChinaHadoop.cn

```
sqoop job \
  --create visits \
  --import \
  --connect jdbc:mysql://mysql.example.com/sqoop \
  --username sqoop \
  --password sqoop \
  --table visits \
  --incremental append \
  --check-column id \
  --last-value 0
```

运行sqoop作业：sqoop job --exec visits

- 每次成功运行后，sqoop将最后一条记录的id值保存到metastore中，供下次使用。

Sqoop import增量导入（三）

小象科技 ChinaHadoop.cn

```
sqoop import \
  --connect jdbc:mysql://mysql.example.com/sqoop \
  --username sqoop \
  --password sqoop \
  --table visits \
  --incremental lastmodified \
  --check-column last_update_date \
  --last-value “2013-05-22 01:01:01”
```

- 数据库中有一列last_update_date，记录了上次修改时间；
- Sqoop仅将某时刻后的数据导入Hadoop。

Sqoop Export

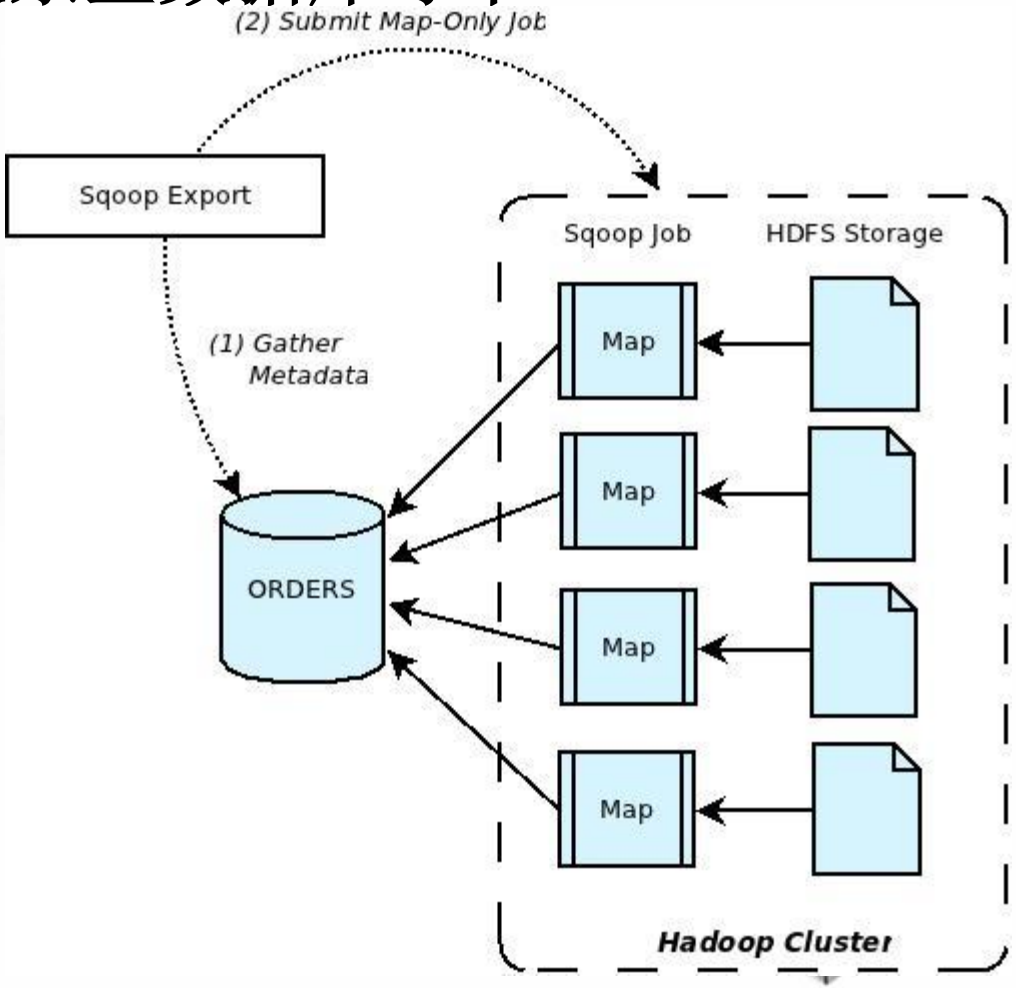
小象科技 ChinaHadoop.cn

- 将数据从Hadoop导入关系型数据库导中

步骤1：Sqoop与数据库Server通信，获取数据库表的元数据信息；

步骤2：并行导入数据：

- ✓ 将Hadoop上文件划分成若干个split；
- ✓ 每个split由一个Map Task进行数据导入。



Sqoop Export使用方法

小象科技 ChinaHadoop.cn

```
sqoop export \
    --connect jdbc:mysql://mysql.example.com/sqoop \
    --username sqoop \
    --password sqoop \
    --table cities \
    --export-dir cities
```

- --connnect: 指定JDBC URL
- --username/password：mysql数据库的用户名
- --table：要导入的数据库表
- export-dir：数据在HDFS上存放目录

Sqoop Export—保证原子性

小象科技 ChinaHadoop.cn

```
sqoop export \
    --connect jdbc:mysql://mysql.example.com/sqoop \
    --username sqoop \
    --password sqoop \
    --table cities \
    --staging-table staging_cities
```

Sqoop Export—更新已有数据

小象科技 ChinaHadoop.cn

```
sqoop export \
    --connect jdbc:mysql://mysql.example.com/sqoop \
    --username sqoop \
    --password sqoop \
    --table cities \
    --update-key id
sqoop export \
    --connect jdbc:mysql://mysql.example.com/sqoop \
    --username sqoop \
    --password sqoop \
    --table cities \
    --update-key id \
    --update-mode allowinsert
```

Sqoop Export—选择性插入

小象科技 ChinaHadoop.cn

```
sqoop export \
    --connect jdbc:mysql://mysql.example.com/sqoop \
    --username sqoop \
    --password sqoop \
    --table cities \
    --columns country,city
```

Sqoop与其他系统结合

小象科技 ChinaHadoop.cn

- **Sqoop可以与Oozie、Hive、Hbase等系统结合；**
- **用户需要在sqoop-env.sh中增加HBASE_HOME、HIVE_HOME等环境变量。**

Sqoop与Hive结合

小象科技 ChinaHadoop.cn

```
sqoop import \
    --connect jdbc:mysql://mysql.example.com/sqoop \
    --username sqoop \
    --password sqoop \
    --table cities \
    --hive-import
```

Sqoop与HBase结合

小象科技 ChinaHadoop.cn

```
sqoop import \
    --connect jdbc:mysql://mysql.example.com/sqoop \
    --username sqoop \
    --password sqoop \
    --table cities \
    --hbase-table cities \
    --column-family world
```

Flume与Sqoop参考资料

小象科技 ChinaHadoop.cn

- **Flume官网：http://flume.apache.org/**
- **Sqoop官网：http://sqoop.apache.org/**
- **Sqoop书籍：Apache Sqoop Cookbook**

主要内容

小象科技
ChinaHadoop.cn

1. 背景介绍
2. Hadoop数据收集系统
3. 传统数据库与Hadoop间数据同步
4. 总结

总结

- **Flume可提供分布式、高可靠、扩展性好地数据收集服务；**
  - ✓ **Flume各类组件均是分布式的，不存在单点问题**
- **Sqoop是一个关系型数据库与Hadoop间的数据同步的工具**
  - ✓ **将数据同步问题转化为MR作业**
  - ✓ **实时性不够好**

小象科技 ChinaHadoop.cn

联系我们：

- 新浪微博：ChinaHadoop
- 微信公号：ChinaHadoop

让你的数据产生价值！