



高级Hadoop 2.x (三)

谭唐华

课程大纲

- 二次排序
- Hadoop 版本选择
- Nginx服务器日志分析

二次排序

■ 需求说明

首先依据第一个字段排序，然后对第二个字段进行排序

a,1
z,3
b,2
a,100
a,3
b,1



a,1
a,3
a,100
b,1
b,2
z,3

二次排序

- 1) 自定义Key 数据类型，实现WritableComparable接口。
- 2) 自定义分区函数类，实现Partitioner接口，此为Key的第一次比较，在Job中使用setPartitionerClass 设置。
- 3) 自定义分组类，继承WritableComparator类，在Job中使用setGroupingComparatorClass设置。在Reduce阶段，构造一个与Key相对应的Value迭代器的时候，只要first相同就属于同一个组，放在一个Value迭代器中。

RawComparator Class

- Hadoop为序列化提供了优化，类型的比较对M/R而言至关重要，Key和Key的比较也是在排序阶段完成的，hadoop提供了原生的比较器接口RawComparator<T>用于序列化字节间的比较，该接口允许其实现直接比较数据流中的记录，无需反序列化为对象，RawComparator是一个原生的优化接口类，它只是简单的提供了用于数据流中简单的数据对比方法，从而提供优化。
- 该类并非被多数的衍生类所实现，其具体的子类为WritableComparator，多数情况下是作为实现Writable接口的类的内置类，提供序列化字节的比较。似于一个注册表，里面记录了所有Comparator类的集合，Comparators成员用一张Hash表记录Key=Class，value=WritableComparator的注册信息。

二次排序

原始数据
Map input

a 1
z 3
b 2
a 100
a 3
b 1

组合 key
Map output

a#1,1
z#3,3
b#2, 2
a#100,100
a#3,3
b#1,1

分区与排序

a#1,1
a#3,3
a#100,100
b#1,1
b#2,2
z#3,3

分组

a#1 <1,3,100>
b#1 <1,2>
z#3,<3>

Shuffle 过程

Reduce

Reduce output

a, 1
a, 3
a, 100
b, 1
b, 2
z, 3

MapReduce Join

- Reduce 端 Join: Join的操作实在Reduce 端执行。
- Map 端 Join:
针对以下场景: 两个待连接的表, 其中一个非常大, 另一个非常小, 可将小表直接放于内存中, DistributeCache实现。
- 半连接 Semi Join: map 端Join和reduce 端Join结合。

Reduce Join

We use a couple toy data sets to better illustrate joining in Hadoop. Let's take a comma-separated Customers file where each record has three fields: Customer ID, Name, and Phone Number. We put four records in the file for illustration:

```
1,Stephanie Leung,555-555-5555
2,Edward Kim,123-456-7890
3,Jose Madriz,281-330-8004
4,David Stork,408-555-0000
```

We store Customer orders in a separate file, called Orders. It's in CSV format, with four fields: Customer ID, Order ID, Price, and Purchase Date.

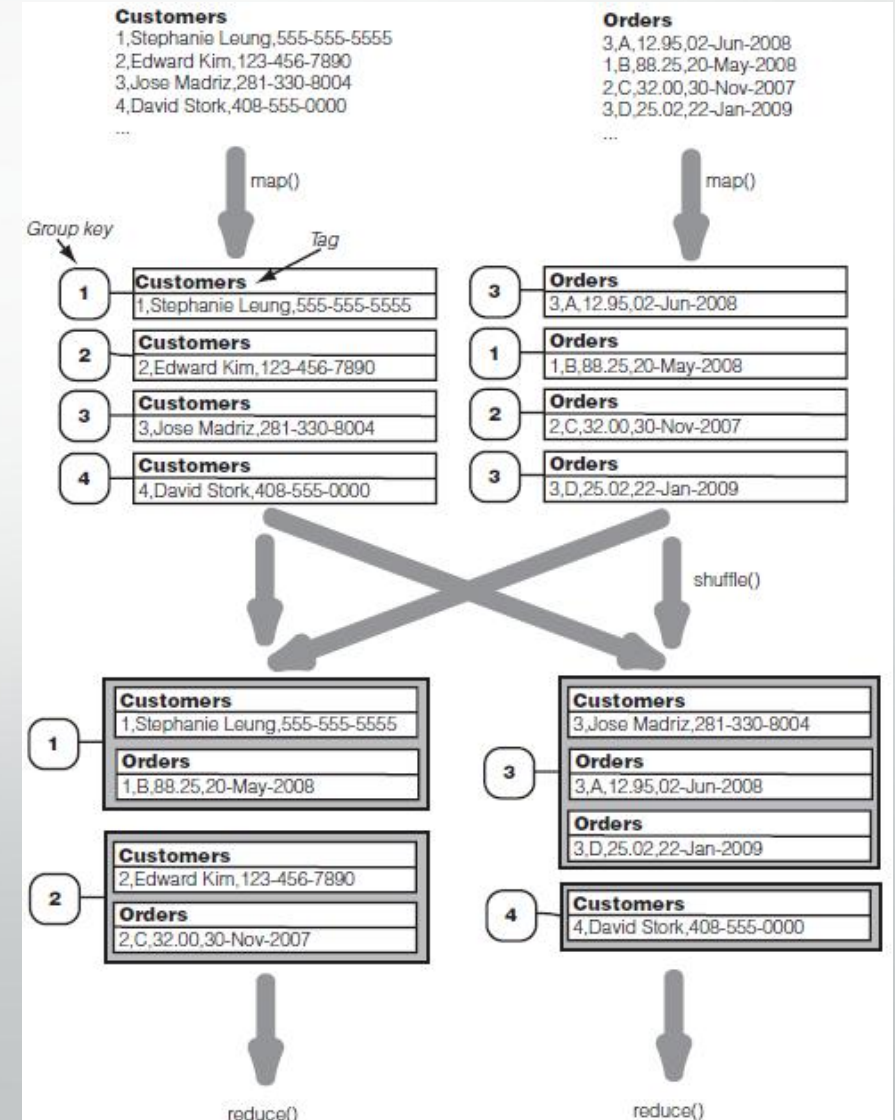
```
3,A,12.95,02-Jun-2008
1,B,88.25,20-May-2008
2,C,32.00,30-Nov-2007
3,D,25.02,22-Jan-2009
```

If we want an inner join of the two data sets above, the desired output would look a listing 5.2.

Listing 5.2 Desired output of an inner join between Customers and Orders data

```
1,Stephanie Leung,555-555-5555,B,88.25,20-May-2008
2,Edward Kim,123-456-7890,C,32.00,30-Nov-2007
3,Jose Madriz,281-330-8004,A,12.95,02-Jun-2008
3,Jose Madriz,281-330-8004,D,25.02,22-Jan-2009
```

Hadoop can also perform outer joins, although to simplify explanation we focus on inner joins.



Distributed Copy

- **DistCp Version 2 (distributed copy) is a tool used for large inter/intra-cluster copying.** It uses MapReduce to effect its distribution, error handling and recovery, and reporting. It expands a list of files and directories into input to map tasks, each of which will copy a partition of the files specified in the source list.
- [The erstwhile implementation of DistCp](#) has its share of quirks and drawbacks, both in its usage, as well as its extensibility and performance. **The purpose of the DistCp refactor was to fix these shortcomings, enabling it to be used and extended programmatically.** New paradigms have been introduced to improve runtime and setup performance, while simultaneously retaining the legacy behaviour as default.

HFTP Introduction

- HFTP is a Hadoop filesystem implementation that lets you read data from a remote Hadoop HDFS cluster. The reads are done via HTTP, and data is sourced from DataNodes.
- HFTP is a read-only filesystem, and will throw exceptions if you try to use it to write data or modify the filesystem state.
- HFTP is primarily useful if you have **multiple HDFS clusters with different versions** and you need to move data from one to another. **HFTP is wire-compatible even between different versions of HDFS.**

课程大纲

- 二次排序
- Hadoop 版本选择
- Nginx服务器日志分析

Cloudera Hadoop

- 2008年成立的 **Cloudera** 是最早将 Hadoop 商用的公司，为合作伙伴提供 Hadoop 的商用解决方案，主要是包括支持，咨询服务，培训。
- 2009年Hadoop的创始人 Doug Cutting也加盟 Cloudera公司。Cloudera 产品主要为CDH, Cloudera Manager, Cloudera Support
- CDH是Cloudera的Hadoop发行版，完全开源，比Apache Hadoop在兼容性，安全性，稳定性上有所增强。
- Cloudera Manager是集群的软件分发及管理监控平台，可以在几个小时内部署好一个Hadoop集群，并对集群的节点及服务进行实时监控。Cloudera Support即是对Hadoop的技术支持。
- Cloudera 的标价为每年每个节点4000美元。Cloudera开发并贡献了可实时处理大数据的Impala项目。

Hortonworks Hadoop

- 2011年成立的Hortonworks是雅虎与硅谷风投公司Benchmark Capital合资组建
- 公司成立之初就吸纳了大约25名至30名专门研究Hadoop的雅虎工程师，上述工程师均在2005年开始协助雅虎开发Hadoop，贡献了Hadoop 80%的代码。
- 雅虎工程副总裁、雅虎Hadoop开发团队负责人Eric Baldeschwieler出任Hortonworks的首席执行官。
- Hortonworks 的主打产品是Hortonworks Data Platform (HDP)，也同样是100%开源的产品，HDP除常见的项目外还包含了Ambari，一款开源的安
装和管理系统
- HCatalog，一个元数据管理系统，HCatalog现已集成到Facebook 开源的Hive中。Hortonworks的Stinger开创性地极大地优化了Hive项目。
Hortonworks为入门提供了一个非常好的，易于使用的沙盒。
- Hortonworks开发了很多增强特性并提交至核心主干，这使得Apache Hadoop能够在包括Windows Server和Windows Azure在内的Microsoft
Windows平台上本地运行。定价以集群为基础，每10个节点每年为12500美元。

版本选择

- Hadoop安装有好几种选择。你可以只使用Apache Hadoop项目并从Hadoop生态系统中创建自己的发行版本。像Cloudera、Hortonworks或MapR这样的Hadoop发行版本提供商为了减少用户需要付出的工作，在Apache Hadoop之上添加了如工具、商业支持等特性。在Hadoop发行版本之上，为了使用如建模、代码生成、大数据作业调度、所有不同种类的数据源集成等附加特性，你可以使用一个大数据套件。一定要评估不同的选择来为自己的大数据项目做出正确的决策。

课程大纲

- 二次排序
- Hadoop 版本选择
- Nginx服务器日志分析

Nginx 服务器日志格式

在Nginx中日志文件是由log_format这个指令来定义的，它的语法如下：

```
log_format name format
```

```
## name: 指的是日志格式的名称（后面调用）
```

```
## format: 设置日志具体格式的
```

在Nginx中有自己默认的日志格式，如下内容：

```
#log_format main '$remote_addr - $remote_user [$time_local] "$request" '
#                 '$status $body_bytes_sent "$http_referer" '
#                 '"$http_user_agent" "$http_x_forwarded_for";
```

Nginx 服务器日志格式

<code>\$remote_addr</code>	客户端的ip地址（如果中间有代理服务器那么这里显示的ip就为代理服务器的ip地址）
<code>\$remote_user</code>	用于记录远程客户端的用户名称（一般为“-”）
<code>\$time_local</code>	用于记录访问时间和时区
<code>\$request</code>	用于记录请求的url以及请求方法
<code>\$status</code>	响应状态码
<code>\$body_bytes_sent</code>	给客户端发送的文件主体内容大小
<code>\$http_user_agent</code>	用户所使用的代理（一般为浏览器）
<code>\$http_x_forwarded_for</code>	可以记录客户端IP，通过代理服务器来记录客户端的ip地址
<code>\$http_referer</code>	可以记录用户是从哪个链接访问过来的

测试日志格式

```
$remote_addr  
    客户端的ip地址（如果中间有代理服务器那么这里  
    显示的ip就为代理服务器的ip地址）  
$remote_user  
    用于记录远程客户端的用户名称（一般为“-”）  
$time_local  
    用于记录访问时间和时区  
$request  
    用于记录请求的url以及请求方法  
$status  
    响应状态码  
$body_bytes_sent  
    给客户端发送的文件主体内容大小  
$request_body  
    为post的数据  
$http_referer  
    可以记录用户是从哪个链接访问过来的  
$http_user_agent  
    用户所使用的代理（一般为浏览器）  
$http_x_forwarded_for  
    可以记录客户端IP，通过代理服务器来记录客户端的ip地址  
$host  
    服务器主机名称
```

日志分析项目

数据收集

数据存储

数据建模

数据分析

数据应用

