

## HBase、Hive 的安装与使用

### 实验过程及结果

#### HBase

安装 HBase，使用 HBase 创建表：Wuxia

```
hbase(main):004:0> create 'Wuxia','avgcount'
0 row(s) in 1.2660 seconds

=> Hbase::Table - Wuxia
hbase(main):005:0> count 'Wuxia'
0 row(s) in 0.0300 seconds

=> 0
```

修改倒排索引的 reduce 程序，将每个词语及其对应的平均出现次数写入 Wuxia 表中，运行修改后的倒排索引程序：

```
hadoop jar mapreduce-lab.jar hbase.InvertedIndex input output
```

Job 的运行结果如下图所示：

```
17/11/09 00:00:39 INFO mapreduce.Job: Counters: 49
  File System Counters
    FILE: Number of bytes read=152383640
    FILE: Number of bytes written=331240253
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=268308546
    HDFS: Number of bytes written=118238028
    HDFS: Number of read operations=657
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=2
  Job Counters
    Launched map tasks=218
    Launched reduce tasks=1
    Data-local map tasks=218
    Total time spent by all maps in occupied slots (ms)=607534
    Total time spent by all reduces in occupied slots (ms)=8077418
    Total time spent by all map tasks (ms)=607534
    Total time spent by all reduce tasks (ms)=4038709
    Total vcore-milliseconds taken by all map tasks=607534
    Total vcore-milliseconds taken by all reduce tasks=4038709
    Total megabyte-milliseconds taken by all map tasks=622114816
    Total megabyte-milliseconds taken by all reduce tasks=8271276032
```

```
Map-Reduce Framework
  Map input records=1954746
  Map output records=45567096
  Map output bytes=1571080862
  Map output materialized bytes=152384936
  Input split bytes=28643
  Combine input records=45567096
  Combine output records=3976539
  Reduce input groups=3976539
  Reduce shuffle bytes=152384936
  Reduce input records=3976539
  Reduce output records=134846
  Spilled Records=7953078
  Shuffled Maps =218
  Failed Shuffles=0
  Merged Map outputs=218
  GC time elapsed (ms)=60817
  CPU time spent (ms)=1630520
  Physical memory (bytes) snapshot=63478341632
  Virtual memory (bytes) snapshot=435691368448
  Total committed heap usage (bytes)=42988470272

Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
```

Cluster

About

Nodes

Node Labels

Applications

NEW

NEW SAVING

SUBMITTED

ACCEPTED

RUNNING

FINISHED

FAILED

KILLED

Scheduler

Tools

hadoop

All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes
2	0	1	1	3	4 GB	4 GB	0 B	3	8	0	1	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:4096, vCores:8>

Show 20 entries

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
application_1510196833849_0002	mrgaoxx	inverted index hbase	MAPREDUCE	default	Wed Nov 8 19:54:43 -0800 2017	N/A	RUNNING	UNDEFINED		ApplicationMaster
application_1510196833849_0001	mrgaoxx	inverted index hbase	MAPREDUCE	default	Wed Nov 8 19:10:30 -0800 2017	Wed Nov 8 19:20:40 -0800 2017	FINISHED	SUCCEEDED		History

Showing 1 to 2 of 2 entries

运行结束后通过 scan 命令遍历 HBase 中的 Wuxia 表，结果如下图所示，其中单词显示为 16 进制编码：

```

\xE9\xBE\x9F\xE7\xBA column=avgcount:avgcount, timestamp=1510214435913, value=3
\xB9 .0
\xE9\xBE\x9F\xE7\xBC column=avgcount:avgcount, timestamp=1510214435931, value=1
\xA9 .4375
\xE9\xBE\x9F\xE8\x82 column=avgcount:avgcount, timestamp=1510214435952, value=1
\x89 .3333333333333333
\xE9\xBE\x9F\xE8\x83 column=avgcount:avgcount, timestamp=1510214435974, value=2
\x8C .5
\xE9\xBE\x9F\xE8\xA3 column=avgcount:avgcount, timestamp=1510214435991, value=1
\x82 .2857142857142858
\xE9\xBE\x9F\xE9\xB3 column=avgcount:avgcount, timestamp=1510214436009, value=2
\x96 .0
\xE9\xBE\x9F\xE9\xB9 column=avgcount:avgcount, timestamp=1510214436028, value=1
\xA4\xE9\x81\x90\xE9 .0
\xBE\x84
\xEF\xA8\x8C column=avgcount:avgcount, timestamp=1510214436051, value=5
.0
\xEF\xBF\xA1 column=avgcount:avgcount, timestamp=1510214436071, value=1
.5
\xEF\xBF\xA5 column=avgcount:avgcount, timestamp=1510214436115, value=3
.3333333333333333
134846 row(s) in 72.2510 seconds
hbase(main):007:0>

```

编写 Java 程序从 HBase 中读取计数信息，并将其保存到本地文件 result.txt 中。部分结果如下图所示：

```

1 0 1.6875
2 007 1.0
3 01 1.0
4 01章 1.0
5 02章 1.0
6 03章 1.0
7 04 1.0
8 04章 1.0
9 05 1.0
10 05章 1.0
11 06 1.0
12 06章 1.0
13 07章 1.0
14 08张 1.0
15 08章 1.0
16 09章 1.0
17 0一 1.5
18 0年 2.5
19 1 3.3114754098360657
20 1. 1.0
21 10 2.0
22 1000万钱 1.0
23 100间 1.0
24 102个 1.0
25 107 1.0
26 10zz 9.0
27 10岁 1.0

```

## Hive

安装 Hive，使用 Hive Shell 命令创建表：

```

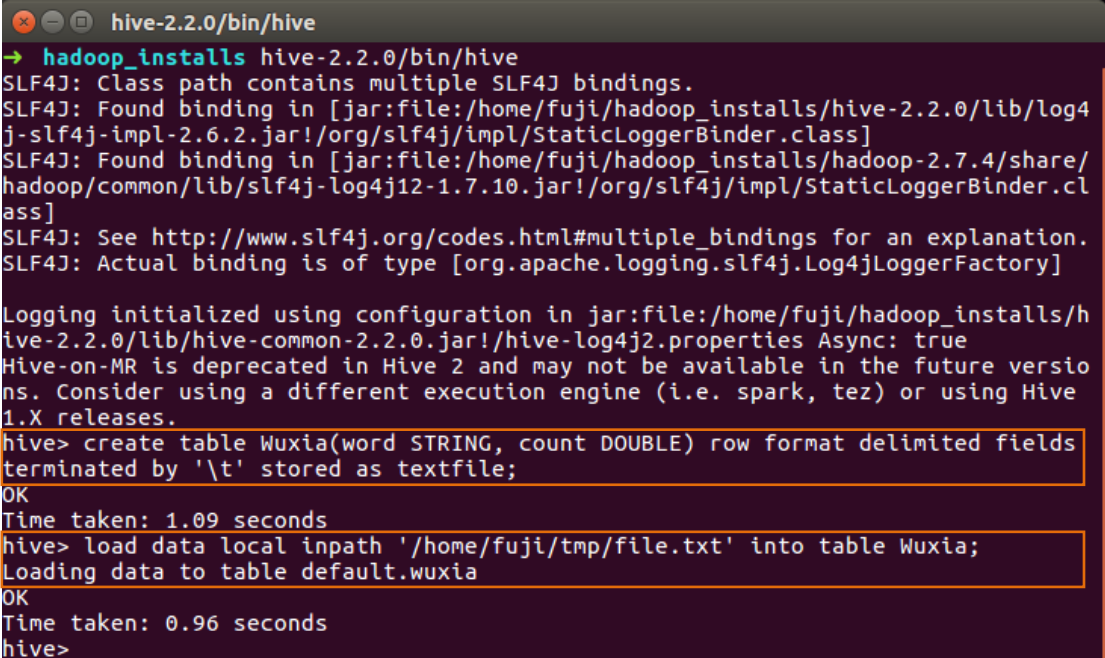
create table Wuxia(word STRING, count DOUBLE) row format delimited fields
terminated by '\t' stored as textfile;

```

从本地文件 file.txt 中导入数据至 Wuxia:

```
load data local inpath '/path/to/result.txt' into table Wuxia;
```

运行结果如下图所示:

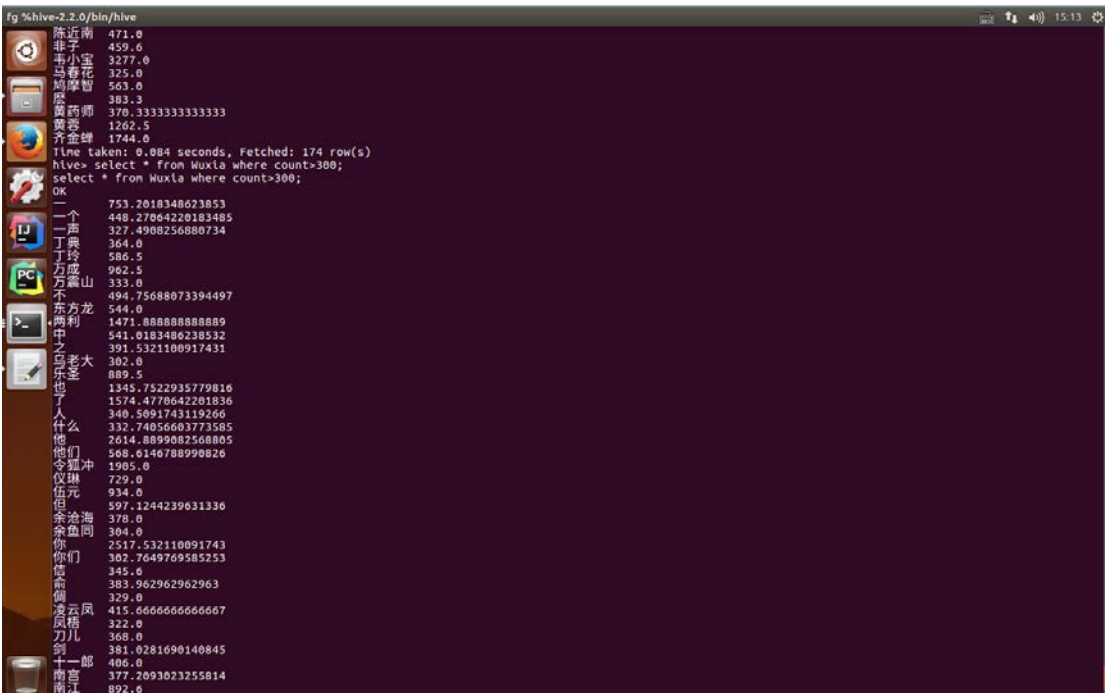


```
hive-2.2.0/bin/hive
→ hadoop_installs hive-2.2.0/bin/hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/fuji/hadoop_installs/hive-2.2.0/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/fuji/hadoop_installs/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/fuji/hadoop_installs/hive-2.2.0/lib/hive-common-2.2.0.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive> create table Wuxia(word STRING, count DOUBLE) row format delimited fields terminated by '\t' stored as textfile;
OK
Time taken: 1.09 seconds
hive> load data local inpath '/home/fuji/tmp/file.txt' into table Wuxia;
Loading data to table default.wuxia
OK
Time taken: 0.96 seconds
hive>
```

使用 Hive 查询出现次数大于 300 的词语:

```
select * from Wuxia where count>300;
```

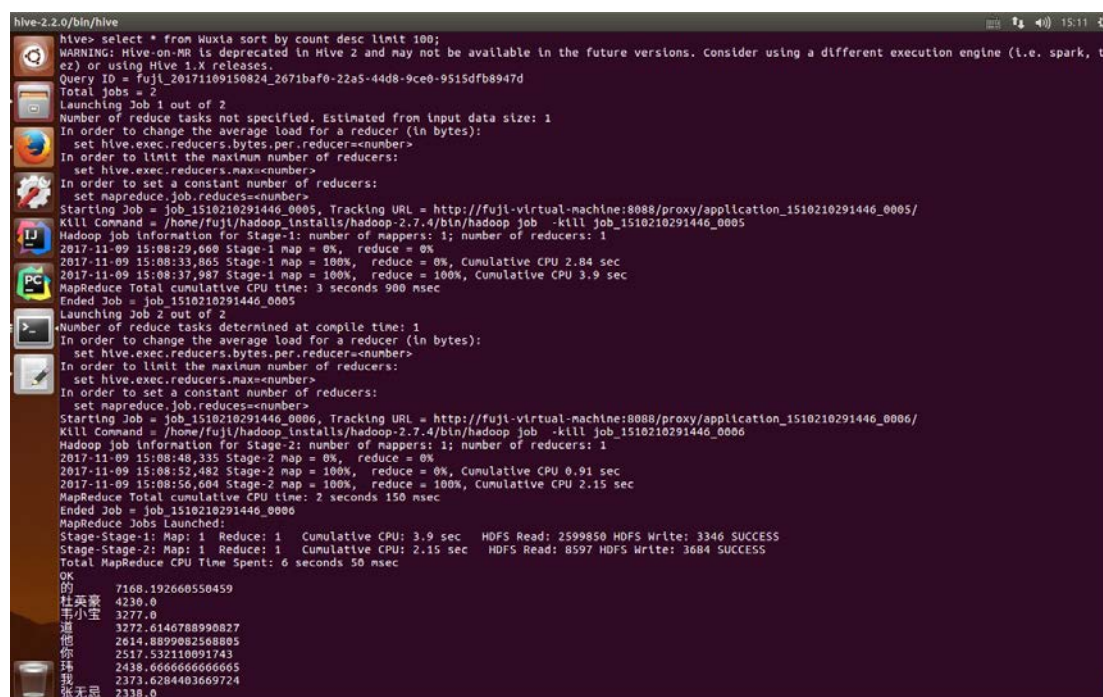


```
fg %hive-2.2.0/bin/hive
陈近南 471.0
非子 459.6
韦小宝 3277.0
马春花 325.0
周芷若 563.0
殷素素 383.3
黄药师 370.3333333333333
黄蓉 1262.5
齐金铃 1744.0
Time taken: 0.084 seconds, Fetched: 174 row(s)
hive> select * from Wuxia where count>300;
select * from Wuxia where count>300;
OK
一个 753.2018348623853
一声 440.27064220183485
丁典 327.4908256880734
丁玲 364.0
方应 586.5
方黛山 962.5
不 333.0
东方龙 494.75688073394497
两利 544.0
中 1471.0888888888889
之 541.0183486238532
老 391.5321100917431
大 302.0
乐 889.5
也 1345.7522935779816
了 1574.4770642201836
人 340.5091743119266
什么 332.74056603773585
他 2614.8899082568805
他们 568.0146788990826
令狐冲 1905.0
仪琳 729.0
伍元 934.0
但 597.1244239631336
余沧海 378.0
余鱼同 304.0
你 2517.532110091743
你们 302.7649769585253
段 345.0
高 383.962962962963
个 329.0
凌云凤 415.6666666666667
凤梧 322.0
刀儿 368.0
到 381.0281690140845
十一郎 486.0
南宮 377.2093023255814
南江 892.6
```

查询前 100 个出现次数最多的词:

```
select * from Wuxia sort by count desc limit 100;
```

运行结果如下图所示：



```
hive-2.2.0/bin/hive
hive> select * from Wuxia sort by count desc limit 100;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = fuji_20171109150824_2671baf0-22a5-44d8-9ce0-9515dfb8947d
Total Jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1510210291446_0005, Tracking URL = http://fuji-virtual-machine:8088/proxy/application_1510210291446_0005/
Kill Command = /home/fuji/hadoop_installs/hadoop-2.7.4/bin/hadoop job -kill job_1510210291446_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-11-09 15:08:29.600 Stage-1 map = 0%, reduce = 0%
2017-11-09 15:08:33.865 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.84 sec
2017-11-09 15:08:37.987 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.9 sec
MapReduce Total cumulative CPU time: 3 seconds 900 msec
Ended Job = job_1510210291446_0005
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1510210291446_0006, Tracking URL = http://fuji-virtual-machine:8088/proxy/application_1510210291446_0006/
Kill Command = /home/fuji/hadoop_installs/hadoop-2.7.4/bin/hadoop job -kill job_1510210291446_0006
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2017-11-09 15:08:48.335 Stage-2 map = 0%, reduce = 0%
2017-11-09 15:08:52.482 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 0.91 sec
2017-11-09 15:08:56.604 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 2.15 sec
MapReduce Total cumulative CPU time: 2 seconds 150 msec
Ended Job = job_1510210291446_0006
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 3.9 sec HDFS Read: 2599850 HDFS Write: 3346 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 2.15 sec HDFS Read: 8597 HDFS Write: 3684 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 50 msec
OK
的 7168 192660550459
杜英豪 4230 0
韦小宝 3277 0
道 3272 6146788990827
他 2614 8899082568805
你 2517 532110091743
殊 2438 6666666666666
我 2373 6284403669724
张无忌 2338 0
```

## 实验体会

本以为 HBase、Hive 等环境的安装不会很费力，但安装过程中发现会遇到各种各样的问题，然后就需要不断地查找资料，大部分还是各种配置出现问题导致。当然，在安装配置问题没有解决的时候会非常烦躁，最后发现环境安装下来耗时要比编码还要久。想要熟练运用 hadoop 还是首先需要了解所涉及的系统的基本原理，如 Hbase 的非结构化的数据库，尤其是存储查询都很快，使得实验运行时间非常短。不得不佩服一下设计分布式大数据处理的工程师们。实验中我们还发现，书中有部分代码已不再适用，这也提醒我们系统在不断更新，我们也需要不断学习来跟上其步伐。