上机实验八
实验报告

刘瀚文

517030910294

2018 年 11 月 8 日

目录

Part I

实验八

Chapter 1

实验准备

1.1 环境配置

1.1.1 hadoop 安装

主要的难点就是 hadoop 的配置和安装，完成了环境的配置就几乎要成功了。

Prerequiste

Install Ubuntu

Create Hadoop User

Setup SSH Certification

Install Java and ssh-server

Download Hadoop 2.2.0

Setup Hadoop Enironment

Configure Hadoop

Format Namenode

Start Hadoop Service

Stop Services

1.2 背景知识

1.2.1 Brief Introduction for Hadoop

Hadoop 简介

Formally speaking, Hadoop is an open source framework for writing and running distributed applications that process large amounts of data.

A Hadoop cluster has many parallel machines that store and process large data sets. Client computers send jobs into this computer cloud and obtain results.

Hadoop 优点

Accessible Hadoop runs on large clusters of commodity machines or on cloud computing services such as Amazon’ s Elastic Compute Cloud (EC2).

Robust Because it is intended to run on commodity hardware, Hadoop is architected with the assumption of frequent hardware malfunctions. It can gracefully handle most such failures.

Scalable Hadoop scales linearly to handle larger data by adding more nodes to the cluster.

Simple Hadoop allows users to quickly write efficient parallel code.

1.2.2 MapReduce Overview

Characteristic

Automatic parallelization & distribution

Fault-tolerant

Provides status and monitoring tools

Clean abstraction for programmers

1.2.3 Map

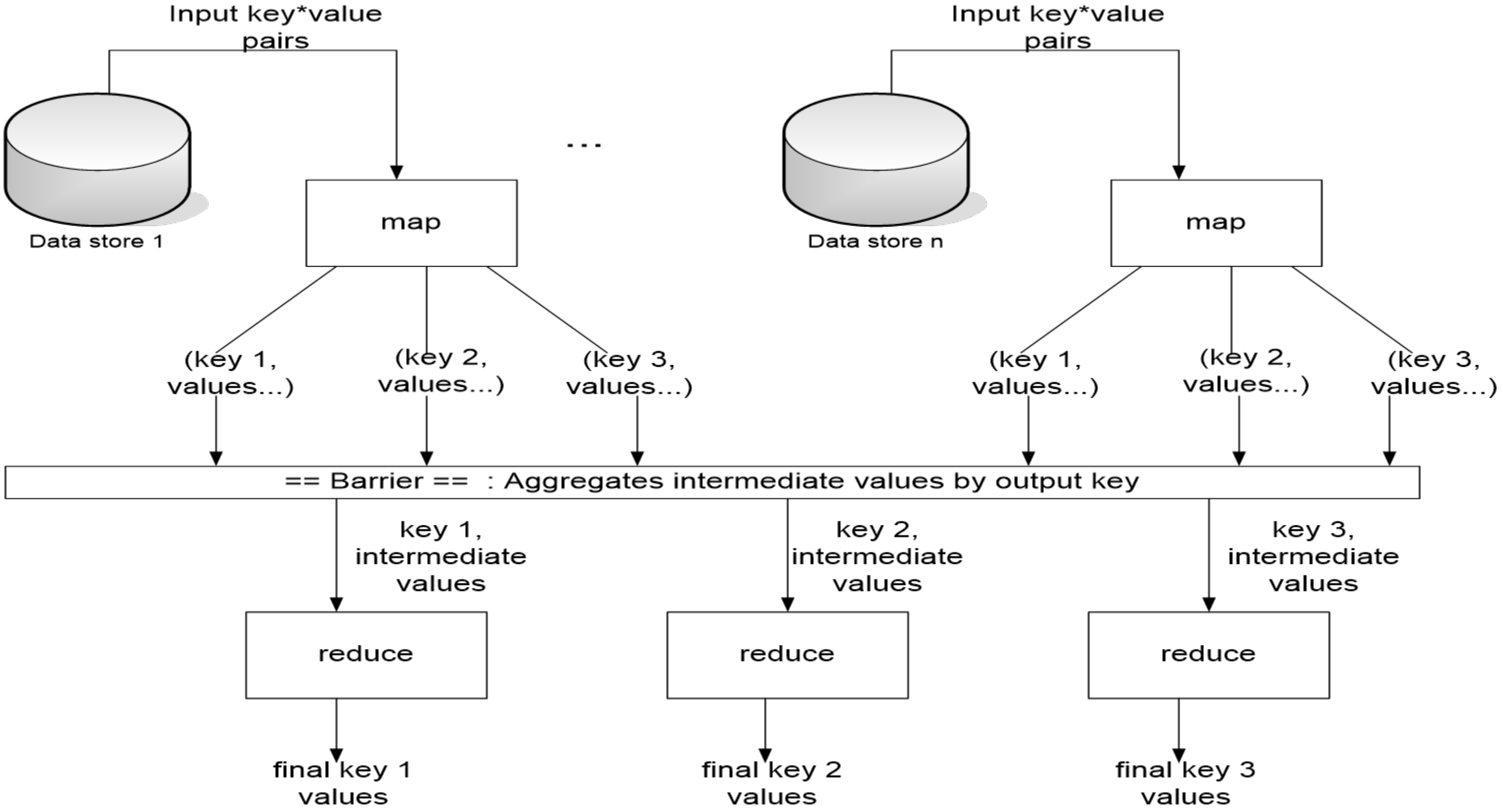
Records from the data source (lines out of files, rows of a database, etc) are fed into the map function as key*value pairs: e.g., (filename, line).

map() produces one or more intermediate values along with an output key from the input.

After the map phase is over, all the intermediate values for a given output key are combined together into a list

reduce() combines those intermediate values into one or more final values for that same output key.(in practice, usually only one final value per key)

1.2.4 Architecture



1.3 简单的自我尝试

WordCount on Hadoop
比较简单

Chapter 2

Mini Exercise

2.1 Exercise 1

2.1.1 要求

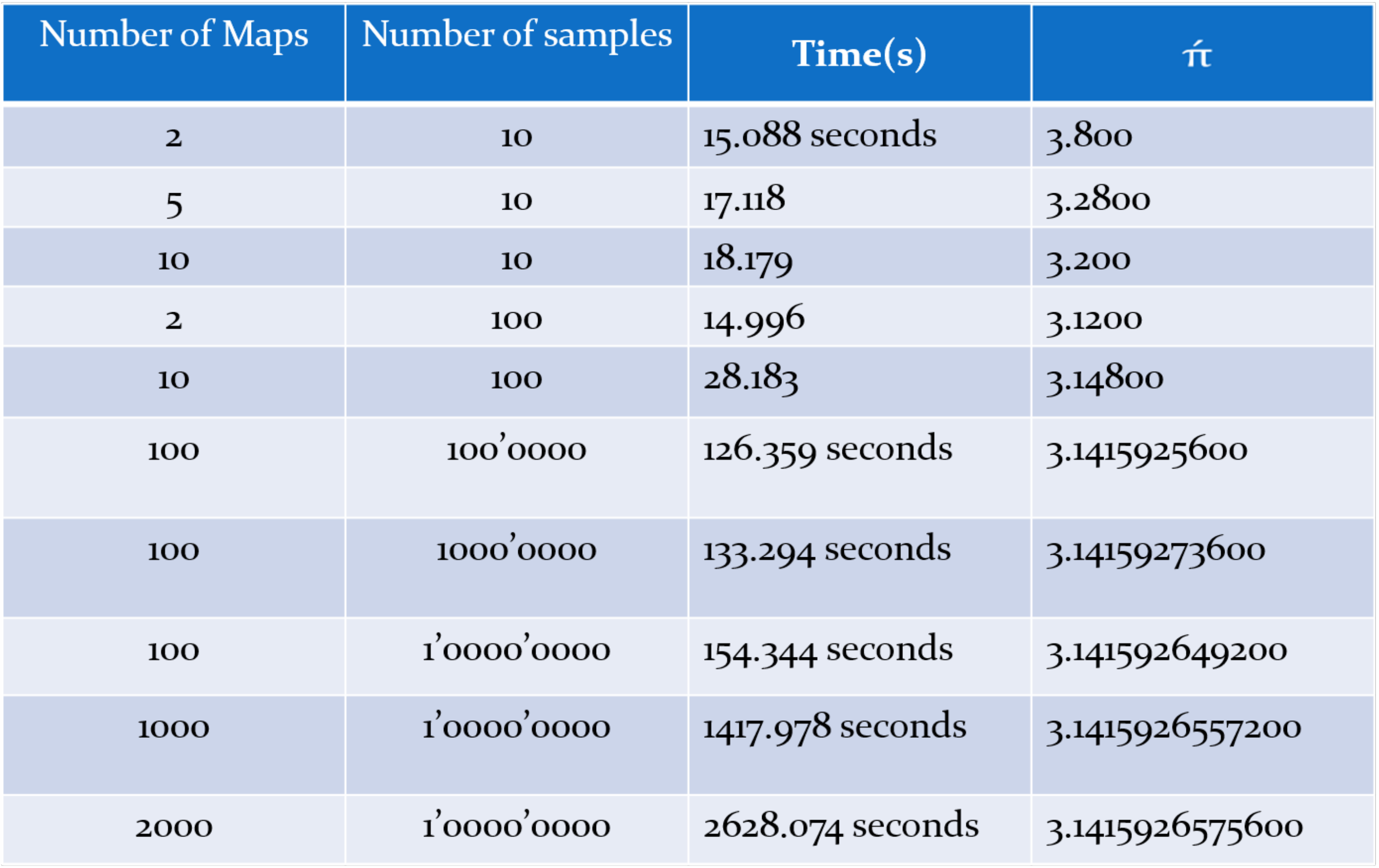
Practise using basic hadoop command and fill in the following table

2.1.2 实验过程

Start hadoop

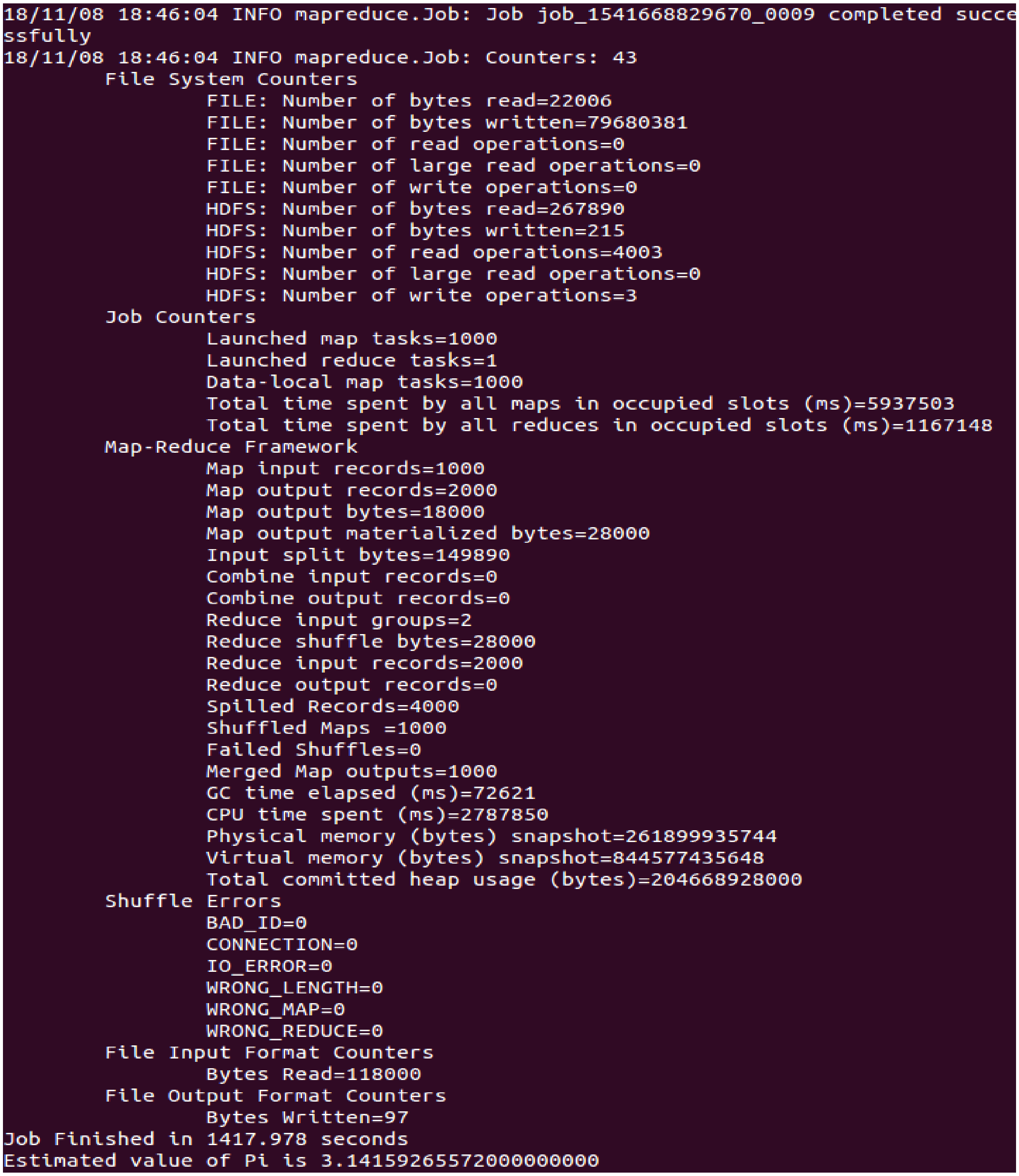
Use command to compute π <nMaps> is the number of mapper jobs and <nSamples> is the number of samples

```
hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-
                                    examples-2.2.0.jar pi  <nMaps>  <
                                    nSamples>
```



Number of Maps	Number of samples	Time(s)	π́
2	10	15.088 seconds	3.800
5	10	17.118	3.2800
10	10	18.179	3.200
2	100	14.996	3.1200
10	100	28.183	3.14800
100	100’0000	126.359 seconds	3.1415925600
100	1000’0000	133.294 seconds	3.14159273600
100	1’0000’0000	154.344 seconds	3.141592649200
1000	1’0000’0000	1417.978 seconds	3.1415926557200
2000	1’0000’0000	2628.074 seconds	3.1415926575600

随着测试数量的上升，时间和精确程度都在上升！



```
18/11/08 18:46:04 INFO mapreduce.Job: Job job_1541668829670_0009 completed succe
ssfully
18/11/08 18:46:04 INFO mapreduce.Job: Counters: 43
        File System Counters
                FILE: Number of bytes read=22006
                FILE: Number of bytes written=79680381
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=267890
                HDFS: Number of bytes written=215
                HDFS: Number of read operations=4003
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=3
        Job Counters
                Launched map tasks=1000
                Launched reduce tasks=1
                Data-local map tasks=1000
                Total time spent by all maps in occupied slots (ms)=5937503
                Total time spent by all reduces in occupied slots (ms)=1167148
        Map-Reduce Framework
                Map input records=1000
                Map output records=2000
                Map output bytes=18000
                Map output materialized bytes=28000
                Input split bytes=149890
                Combine input records=0
                Combine output records=0
                Reduce input groups=2
                Reduce shuffle bytes=28000
                Reduce input records=2000
                Reduce output records=0
                Spilled Records=4000
                Shuffled Maps =1000
                Failed Shuffles=0
                Merged Map outputs=1000
                GC time elapsed (ms)=72621
                CPU time spent (ms)=2787850
                Physical memory (bytes) snapshot=261899935744
                Virtual memory (bytes) snapshot=844577435648
                Total committed heap usage (bytes)=204668928000
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=118000
        File Output Format Counters
                Bytes Written=97
Job Finished in 1417.978 seconds
Estimated value of Pi is 3.14159265572000000000
```

展示 1000(Map) * 1’0000’0000(Samples) 的结果！

Get the result

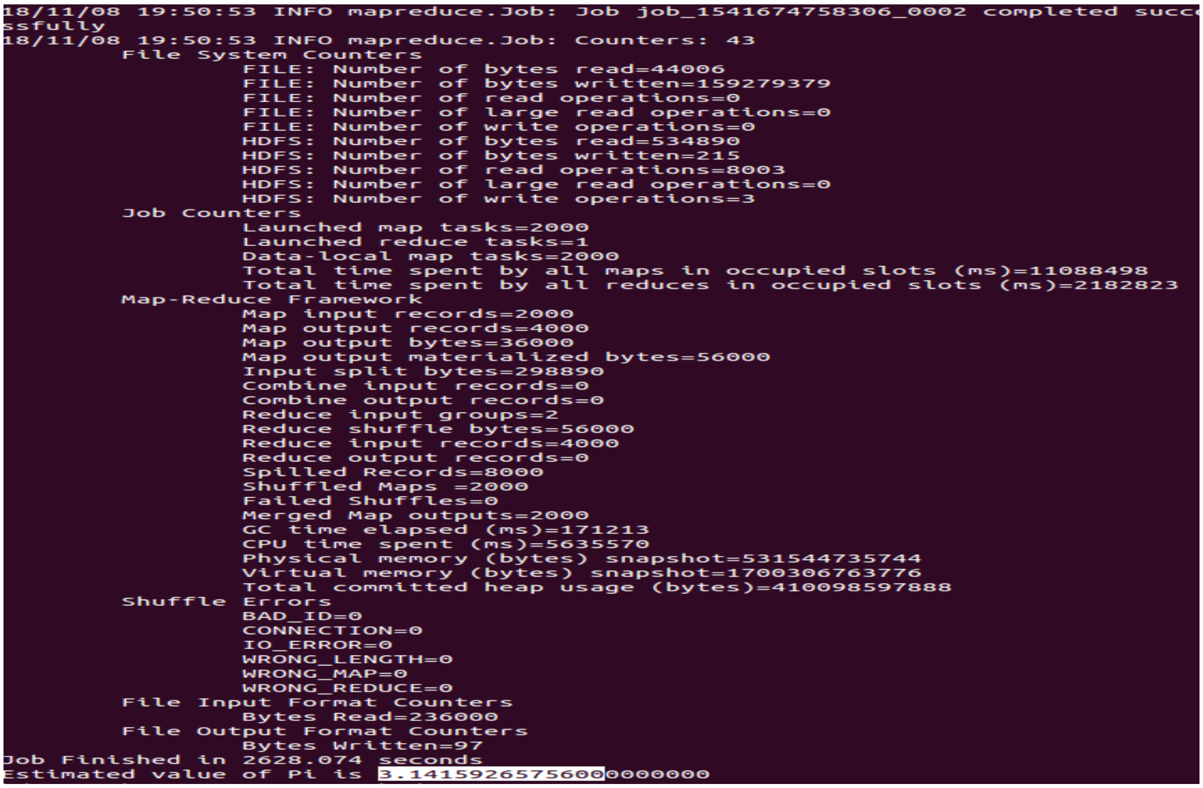
2.2 Exercise 2

2.2.1 要求

Work out a solution to make the computed $\pi approximatethe5thdigitafterthedecimaldotcorrectly$.

2.2.2 实验过程

实验过程和 1 类似，为了进行更精确的计算，调大了内存 (6G -> 7.5G)，分配了更多处理器内核 (6 -> 8)，观察是否有更快的结果输出。



```
18/11/08 19:50:53 INFO mapreduce.Job: Job job_1541674758306_0002 completed succe
ssfully
18/11/08 19:50:53 INFO mapreduce.Job: Counters: 43
        File System Counters
                FILE: Number of bytes read=44006
                FILE: Number of bytes written=159279379
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=534890
                HDFS: Number of bytes written=215
                HDFS: Number of read operations=8003
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=3
        Job Counters
                Launched map tasks=2000
                Launched reduce tasks=1
                Data-local map tasks=2000
                Total time spent by all maps in occupied slots (ms)=11088498
                Total time spent by all reduces in occupied slots (ms)=2182823
        Map-Reduce Framework
                Map input records=2000
                Map output records=4000
                Map output bytes=36000
                Map output materialized bytes=56000
                Input split bytes=298890
                Combine input records=0
                Combine output records=0
                Reduce input groups=2
                Reduce shuffle bytes=56000
                Reduce input records=4000
                Reduce output records=0
                Spilled Records=8000
                Shuffled Maps =2000
                Failed Shuffles=0
                Merged Map outputs=2000
                GC time elapsed (ms)=171213
                CPU time spent (ms)=5635570
                Physical memory (bytes) snapshot=531544735744
                Virtual memory (bytes) snapshot=1700306763776
                Total committed heap usage (bytes)=410098597888
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=236000
        File Output Format Counters
                Bytes Written=97
Job Finished in 2628.074 seconds
Estimated value of Pi is 3.14159265756000000000
```

展示 2000(Map) * 1’0000’0000(Samples) 的结果！结果是：3.1415926575600 和 (3.1415926535898) 比较接近，满足练习要求！

Part II

实验总结

这次实验主要是学习新的 Hadoop，很有趣！

在配置环境的过程中，出现了一些奇妙的问题。通过使用 VMVare 的快照和自己 DeBUG 的过程学习到了新的 Ubuntu 的知识，有所进步！

期待在下一次实验中学习更多的知识！