实验一：Batch job in cloud

Jiahua Wang

Part A 认识 Spark 与 Hadoop

Hadoop 和 Spark 和配置运行结果见附录图 1 和 图 2 。

运行 Hadoop Word Count 样例

这里的输入文件是Python之禅，结果见附录图 3。

运行 Spark Connected Components 样例

结果见附录图 4。

华为云 ECS 是一种 PaaS

华为云 ECS 提供操作系统等一些基本的系统环境，可以在此基础上进一步安装其他软件。尽管华为云 ECS 的环境并不丰富，比如需要自己配置 Hadoop 环境，但仍然属于 PaaS。

Hadoop 与 Spark 的区别

- 应用场景
  - Hadoop 包括分布式存储系统 HDFS 和数据分析框架 MapReduce，本身就可以完成对海量数据的存储和分析。
  - Spark 不提供文件系统，专门用来处理分布式存储的大数据，它要借助 HDFS 或其他文件系统存储。
- 处理速度
  - Hadoop 的 MapReduce 分步对处理数据：先读磁盘进行 Map，将结果写到磁盘，然后再读磁盘进行 Reduce，再将结果写到磁盘，多次 I/O 影响速度。
  - Spark 从磁盘中读取数据，接下来中间数据都存储在内存中，最后将结果写回磁盘，所以 Spark 比 Hadoop 更快。
- 容错性
  - Hadoop 将每次处理后的数据都写入到磁盘上，容错性好。
  - Spark 的数据对象存储在 RDD 上，通过 checkpoint 和 Lineage 等机制实现容错。

Part B 使用 Spark 执行 PageRank 算法

候选人名单

```
(4037,13.687824661001775)
(15,10.9328059520621)
(6634,10.656469713599291)
(2625,9.755679770957746)
(2398,7.750205920765446)
(2470,7.498077775768758)
(2237,7.417430386169762)
(4191,6.7377444603752865)
(7553,6.44622789746703)
```

```
(5254,6.38790776195012)
(2328,6.058602112229776)
(1186,6.0475330647209775)
(1297,5.78105575667395)
(4335,5.754084280496569)
(7620,5.740174509100523)
(5412,5.70106209414574)
(7632,5.667866508823103)
(4875,5.567064113522038)
(6946,5.372790269370961)
(3352,5.300100134345485)
```

PageRank 算法

描述

算法分为两个阶段。在初始化阶段，首先将图的每个顶点关联到该点的出度上，然后设置每条边的权重为出度的倒数，再初始化每个顶点的权重为 `1.0` 。

```
var ranks = graph
    .outerJoinVertices(graph.outDegrees) {(vid, vdata, deg) ⇒
deg.getOrElse(0)}
    .mapTriplets(e ⇒ 1.0 / e.srcAttr, TripletFields.Src)
    .mapVertices{(id, attr) ⇒ 1.0}
```

在迭代阶段，每个顶点将自身的权重平均分配到相邻的目标点上，从而更新所有点的权重。迭代数轮之后权重收敛。

```
for (_ ← 1 to 100) {
    ranks.cache()
    val updates = ranks.aggregateMessages[Double](
        ctx ⇒ ctx.sendToDst(ctx.srcAttr * ctx.attr), _ + _,
TripletFields.Src
    )
    ranks = ranks.outerJoinVertices(updates) {
        (id, oldRank, msgSumOpt) ⇒ resetProb +
        (1.0 - resetProb) * msgSumOpt.getOrElse(0.0)
    }
}
```

最后将权重进行降序排序，得到声望前 20 高的候选人名单。

```
ranks.vertices.collect().sortBy(- _._2).slice(0,20)
```

部署

假定我们为 `root` 用户，以下操作均在 master 节点上进行。首先启动 Hadoop 和 Spark 。

```
cd /usr/local/hadoop/ && sbin/start-all.sh
cd /usr/local/spark/ && sbin/start-master.sh && sbin/start-slaves.sh
```

假定数据集位于 `~/Wiki-Vote.txt` ，将数据集上传到 HDFS。

```
cd ~
hadoop fs -mkdir -p /user/root/data/graphx/
hadoop fs -put Wiki-Vote.txt /user/root/data/graphx/
```

假定 PageRank 应用位于 `~/pagerank` ，目录结构为

```
.
├── simple.sbt
└── src
    └── main
        └── scala
            └── PageRank.scala
```

将项目编译打包成 jar 文件。

```
cd ~/pagerank && /usr/local/sbt/sbt package
```

运行

将生成的 jar 包提交到 Spark 中运行。

```
cd ~/pagerank
/usr/local/spark/bin/spark-submit --class "PageRank"
target/scala-2.11/simple-project_2.11-1.0.jar
```

输出得到声望前 20 高的候选人名单，详见上节 候选人名单 和附录图 5。

附录

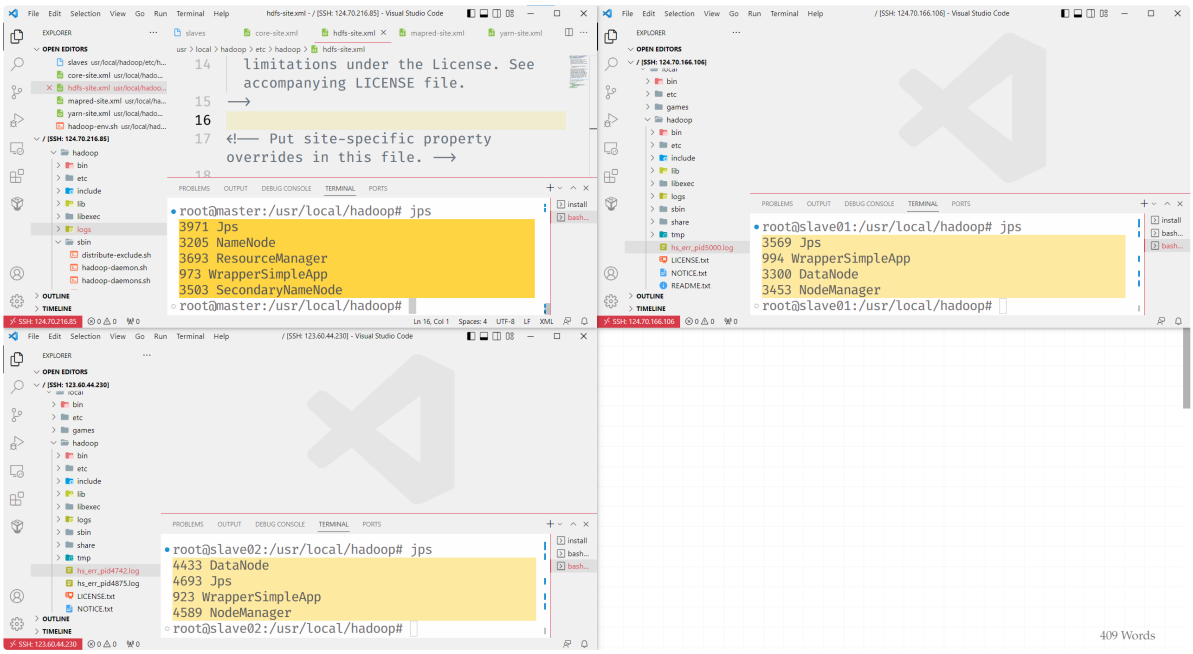

图1 配置运行 Hadoop 结果

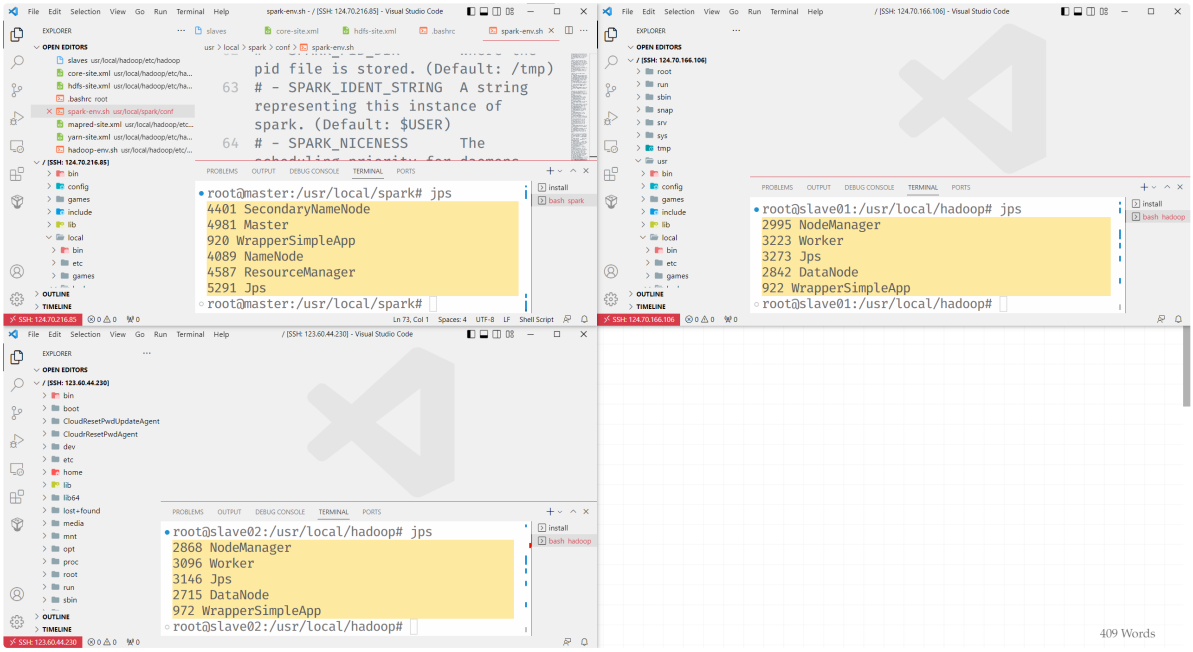

图2 配置运行 Spark 结果

```
root@master:~# hadoop fs -cat /output/part-r-00000
*right* 1
--      1
--obvious       1
Although        3
Beautiful       1
Complex 1
Dutch.  1
Errors  1
Explicit        1
Flat    1
If      2
In      1
```

图3 Hadoop Word Count 样例运行结果

```
22/11/19 15:37:45 INFO executor.Executor: 1 block locks were n
[rdd_32_0]
22/11/19 15:37:45 INFO executor.Executor: Finished task 0.0 in
22/11/19 15:37:45 INFO scheduler.TaskSetManager: Finished task
22/11/19 15:37:45 INFO scheduler.TaskSchedulerImpl: Removed Ta
22/11/19 15:37:45 INFO scheduler.DAGScheduler: ResultStage 19
22/11/19 15:37:45 INFO scheduler.DAGScheduler: Job 3 finished:
(justinbieber,1)
(matei_zaharia,3)
(ladygaga,1)
(BarackObama,1)
(jeresig,3)
(odersky,3)
22/11/19 15:37:45 INFO server.AbstractConnector: Stopped Spark
22/11/19 15:37:46 INFO ui.SparkUI: Stopped Spark web UI at htt
22/11/19 15:37:46 INFO spark.MapOutputTrackerMasterEndpoint: M
22/11/19 15:37:46 INFO memory.MemoryStore: MemoryStore cleared
22/11/19 15:37:46 INFO storage.BlockManager: BlockManager stop
```

图4 Spark Connected Components 样例运行结果

```
22/11/19 23:41:52 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 205.0, wh
22/11/19 23:41:52 INFO scheduler.DAGScheduler: ResultStage 205 (collect at Pa
22/11/19 23:41:52 INFO scheduler.DAGScheduler: Job 1 finished: collect at Pag
(4037,13.687824661001775)
(15,10.9328059520621)
(6634,10.656469713599291)
(2625,9.755679770957746)
(2398,7.750205920765446)
(2470,7.498077775768758)
(2237,7.417430386169762)
(4191,6.7377444603752865)
(7553,6.44622789746703)
(5254,6.38790776195012)
(2328,6.058602112229776)
(1186,6.0475330647209775)
(1297,5.78105575667395)
(4335,5.754084280496569)
(7620,5.740174509100523)
(5412,5.70106209414574)
(7632,5.667866508823103)
(4875,5.567064113522038)
(6946,5.372790269370961)
(3352,5.300100134345485)
22/11/19 23:41:52 INFO server.AbstractConnector: Stopped Spark@14c053c6{HTTP/
22/11/19 23:41:52 INFO ui.SparkUI: Stopped Spark web UI at http://master:4040
22/11/19 23:41:52 INFO spark.MapOutputTrackerMasterEndpoint: MapOutputTracker
```

图5 PageRank 算法运行结果