Spark实验：Standalone模式安装部署

1、实验描述

- 以Standalone的运行模式安装Spark集群
- 主要步骤：
  - 解压安装Spark
  - 添加Spark 配置文件
  - 启动Spark 集群
  - 运行测试用例

2、实验环境

- 虚拟机数量：3
- 系统版本：Centos 7.5
- Hadoop版本：Apache Hadoop 2.7.3
- Spark版本：Apache Spark 2.1.1

3、相关技能

- Spark Standalone安装部署

4、知识点

- 常见linux命令的使用
- 通过修改.bash_profile文件配置Spark
- 验证Spark standalone安装
- 向集群提交application运行
- Spark web UI的使用

5、实验步骤

1. 解压Spark安装包

   将`master`节点中，路径`/home/zkpk/tgz/spark`下的Spark压缩包，解压至`/home/zkpk/`，并查看解压后的目录中的内容。

2. 配置Spark环境变量

   在`master`节点中，编辑路径`/home/zkpk/`下的`.bash_profile`文件，添加Spark相关的环境变量，参考内容如下，然后重新编译使其生效，完成后在`slave01`，`slave02`节点中也执行同样的操作。

   ```
   export SPARK_HOME=/home/zkpk/spark-2.1.1-bin-hadoop2.7
   export PATH=$SPARK_HOME/bin:$SPARK_HOME/sbin:$PATH
   ```

3. 配置`slaves`文件

   在路径`/home/spark-2.1.1-bin-hadoop2.7/conf/`下，将`slaves.template`文件更名为`slaves`，然后将其中指定从节点的内容更改如下，也即将从节点指定为`slave01`和`slave02`。

```
slave01
slave02
```

4. 配置`spark-env.sh`文件

    1. 在`master`节点的`/home/spark-2.1.1-bin-hadoop2.7/conf/`路径下，将`spark-env.sh.template`文件更名为`spark-env.sh`。
    2. 修改`spark-env.sh`文件，完成以下设置：
        1. 设置运行master进程的节点，`export`名为`SPARK_MASTER_HOST`的环境变量，将其值设为`master`。
        2. 设置master进程的通信端口为7077，`export`名为`SPARK_MASTER_PORT`的环境变量，将其值设为`7077`。
        3. 设置每个worker进程使用的cpu内核数量为1，`export`名为`SPARK_WORKER_CORES`的环境变量，将其值设为`1`。
        4. 设置每个worker进程使用的内存大小为1024M，`export`名为`SPARK_WORKER_MEMORY`的环境变量，将其值设为`1024M`。
        5. 设置master进程的web UI访问端口为8080，`export`名为`SPARK_MASTER_WEBUI_PORT`的环境变量，将其值设为`8080`。
        6. 指定Spark配置文件的所在目录，`export`名为`SPARK_CONF_DIR`的环境变量，将其值设为`/home/zkpk/spark-2.1.1-bin-hadoop2.7/conf`。
        7. 指定jdk的安装路径，`export`名为`JAVA_HOME`的环境变量，将其值设为`/usr/java/jdk1.8.0_131/`。
        *** ans ***

```
    export SPARK_MASTER_HOST=master #设置运行master进程的节点
    export SPARK_MASTER_PORT=7077 #设置master的通信端口
    export SPARK_WORKER_CORES=1 #每个worker使用的核数
    export SPARK_WORKER_MEMORY=1024M #每个worker使用的内存大小
    export SPARK_MASTER_WEBUI_PORT=8080 #master的webui端口
    export SPARK_CONF_DIR=/home/zkpk/spark-2.1.1-bin-hadoop2.7/conf
#spark的配置文件目录
    export JAVA_HOME=/usr/java/jdk1.8.0_131/ #jdk安装路径
```

5. 分发Spark安装包

    将经过配置的Spark包装目录，利用scp命令，拷贝到另外两个从节点`slave01`，`slave02`

    *** ans ***

```
scp -r ~/spark-2.1.1-bin-hadoop2.7 zkpk@slave01:~/
scp -r ~/spark-2.1.1-bin-hadoop2.7 zkpk@slave02:~/
```

6. 启动Spark集群

    在`master`节点中，运行`/home/zkpk/spark-2.1.1-bin-hadoop2.7/sbin/`路径下的脚本`start-all.sh`。

7. 验证安装部署成功

    1. 利用`jps`命令观察各个节点中的java进程

2. 在`master`节点中，打开浏览器，访问Spark Web UI的地址*http://master:8080*，观察Spark集群各项指标。

8. 提交示例应用到Spark集群

    1. 使用`spark-submit`命令，并指定以下选项：

        1. 指定应用的主类：`--class org.apache.spark.examples.SparkPi`
        2. 指定master进程的URL：`--master spark://master:7077`
        3. 指定所需的本地jar包路径：`examples/jars/spark-examples_2.11-2.1.1.jar`
        4. 输入该应用所需参数：`该参数取为同学们的学号后三位 mod 8`
        5. 将应用输出内容重定向到本地文档：`> pi.out`

        *** ans ***

        ```
        spark-submit --class org.apache.spark.examples.SparkPi --master
        spark://master:7077 examples/jars/spark-examples_2.11-2.1.1.jar 10
        ```

    2. 键入上述的`spark-submit`命令后，打开`pi.out`文件查看所得结果。