银河麒麟服务器操作系统 V4

Hadoop 软件适配手册

天津麒麟信息技术有限公司

2019 年 5 月

目 录

1 概述

1.1 系统概述

银河麒麟服务器操作系统主要面向军队综合电子信息系统、金融系统以及电力系统等国家关键行业的服务器应用领域，突出高安全性、高可用性、高效数据处理、虚拟化等关键技术优势，针对关键业务构建的丰富高效、安全可靠的功能特性，兼容适配长城、联想、浪潮、华为、曙光等国内主流厂商的服务器整机产品，以及达梦、金仓、神通等主要国产数据库和中创、金蝶、东方通等国产中间件，满足虚拟化、云计算和大数据时代，服务器业务对操作系统在性能、安全性及可扩展性等方面的需求，是一款具有高安全、高可用、高可靠、高性能的自主可控服务器操作系统。

1.2 环境概述

服务器型号	长城信安擎天 DF720 服务器
CPU 类型	飞腾 2000+处理器
操作系统版本	Kylin-4.0.2-server-sp2-2000-19050910.Z1
内核版本	4.4.131
hadoop 版本	2.7.7

1.3 Hadoop 软件简介

Hadoop 是一个由 Apache 基金会所开发的分布式系统基础架构。用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力进行高速运算和存储。

Hadoop 实现了一个分布式文件系统（Hadoop Distributed File System），简称 HDFS。HDFS 有高容错性的特点，并且设计用来部署在低廉的（low-cost）硬件上；而且它提供高吞吐量（high throughput）来访问应用程序的数据，适合那些有着超大数据集（large data set）的应用程序。HDFS 放宽了（relax）POSIX 的要求，可以以流的形式访问（streaming access）文件系统中的数据。

Hadoop 的框架最核心的设计就是：HDFS 和 MapReduce。HDFS 为海量的数据提供了存储，而 MapReduce 则为海量的数据提供了计算。

1.4 HDFS 架构原理

HDFS 是 Hadoop 分布式文件系统（Hadoop Distributed File System）的缩写，为分布式计算存储提供了底层支持。采用 Java 语言开发，可以部署在多种普通的廉价机器上，以集群处理数量积达到大型主机处理性能。

HDFS 采用 master/slave 架构。一个 HDFS 集群包含一个单独的 NameNode 和多个 DataNode。

NameNode 作为 master 服务，它负责管理文件系统的命名空间和客户端对文件的访问。NameNode 会保存文件系统的具体信息，包括文件信息、文件被分割成具体 block 块的信息、以及每一个 block 块归属的 DataNode 的信息。对于整个集群来说，HDFS 通过 NameNode 对用户提供了一个单一的命名空间。

DataNode 作为 slave 服务，在集群中可以存在多个。通常每一个 DataNode 都对应于一个物理节点。DataNode 负责管理节点上它们拥有的存储，它将存储划分为多个 block 块，管理 block 块信息，同时周期性的将其所有的 block 块信息发送给 NameNode。

1.5 MapReduce 介绍

MapReduce 是一种计算模型，该模型可以将大型数据处理任务分解成很多单个的、可以在服务器集群中并行执行的任务，而这些任务的计算结果可以合并在一起来计算最终的结果。简而言之，Hadoop Mapreduce 是一个易于编程并且能在大型集群（上千节点）快速地并行得处理大量数据的软件框架，以可靠，容错的方式部署在商用机器上。MapReduce 这个术语来自两个基本的数据转换操作：map 过程和 reduce 过程。

- map：

map 操作会将集合中的元素从一种形式转化成另一种形式，在这种情况下，输入的键值对会被转换成零到多个键值对输出。其中输入和输出的键必须完全不同，而输入和输出的值则可能完全不同。

- reduce：

某个键的所有键值对都会被分发到同一个 reduce 操作中。确切的说，这个键和这个键所对应的所有值都会被传递给同一个 Reducer。reduce
过程的目的是将值的集合转换成一个值（例如求和或者求平均），或者转换成另一个集合。这个 Reducer 最终会产生一个键值对。需要说明的是，如果 job 不需要 reduce 过程的话，那么 reduce 过程也是可以不用的。

- task：

Hadoop 提供了一套基础设计来处理大多数困难的工作以保证任务可以成功执行，比如 Hadoop 决定如果将提交的 job 分解为多个独立的 map 和 reduce 任务（task）来执行，它就会对这些 task 进行调度并为其分配合适的资源，决定将某个 task 分配到集群中哪个位置（如果可能，通常是这个 task 所要处理的数据所在的位置，这样可以最小化网络开销）。Hadoop 会监控每一个 task 确保其成功完

成，并重启一些失败的 task。

1.6 YARN 介绍

YARN 是 Hadoop 2.0 中的资源管理系统，它的基本设计思想是将 MRv1 中的 JobTracker 拆分成了两个独立的服务：一个全局的资源管理器 ResourceManager 和每个应用程序特有的 ApplicationMaster。其中 ResourceManager 负责整个系统的资源管理和分配，而 ApplicationMaster 负责单个应用程序的管理。

YARN 总体上仍然是 master/slave 结构，在整个资源管理框架中，resourcemanager 为 master，nodemanager 是 slave。Resourcemanager 负责对各个 nademanger 上资源进行统一管理和调度。当用户提交一个应用程序时，需要提供一个用以跟踪和管理这个程序的 ApplicationMaster，它负责向 ResourceManager 申请资源，并要求 NodeManger 启动可以占用一定资源的任务。由于不同的 ApplicationMaster 被分布到不同的节点上，因此它们之间不会相互影响。

YARN 的基本组成结构，YARN 主要由 ResourceManager、NodeManager、ApplicationMaster 和 Container 等几个组件构成。

ResourceManager 是 Master 上一个独立运行的进程，负责集群统一的资源管理、调度、分配等等；NodeManager 是 Slave 上一个独立运行的进程，负责上报节点的状态；App Master 和 Container 是运行在 Slave 上的组件，Container 是 yarn 中分配资源的一个单位，包涵内存、CPU 等等资源，yarn 以 Container 为单位分配资源。

Client 向 ResourceManager 提交的每一个应用程序都必须有一个 Application Master，它经过 ResourceManager 分配资源后，运行于某一个 Slave 节点的 Container 中，具体做事情的 Task，同样也运行与某一个 Slave 节点的 Container 中。RM，NM，AM 乃至普通的 Container 之间的通信，都是用 RPC 机制。

2 Hadoop 软件适配

2.1 解压 hadoop 软件

```
$ tar -xvf hadoop-2.7.7.tar.gz -C /usr/local/
$ cd /usr/local/hadoop-2.7.7/etc/hadoop/
```

2.2 配置文件修改

2.2.1 配置 hadoop-env.sh

```
$ vim hadoop-env.sh
```

修改 JAVA_HOME:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-arm64
```

2.2.2 配置 yarn-env.sh

```
$ vim yarn-env.sh
```

修改:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-arm64
```

2.2.3 配置 core-site.xml

```
$ vim core-site.xml
```

内容如下：

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://Kylin:8020</value>
    <description>HDFS 的 URI，文件系统://namenode 标识:端口号</description>
  </property>

  <property>
    <name>hadoop.tmp.dir</name>
    <value>/usr/local/hadoop-2.7.7/tmp</value>
    <description>namenode 上本地的 hadoop 临时文件夹</description>
  </property>
</configuration>
```

2.2.4 配置 hdfs-size.xml

内容如下：

```
<configuration>
  <property>
    <name>dfs.name.dir</name>
    <value>/usr/local/hadoop-2.7.7/hdfs/name</value>
    <description>namenode 上存储 hdfs 名字空间元数据 </description>
  </property>

  <property>
    <name>dfs.data.dir</name>
    <value>/usr/local/hadoop-2.7.7/hdfs/data</value>
```

```
    <description>datanode 上数据块的物理存储位置</description>
  </property>

  <property>
    <name>dfs.replication</name>
    <value>1</value>
    <description>副本个数，配置默认是 3,应小于 datanode 机器数量</description>
  </property>
</configuration>
```

2.2.5 配置 mapred-site.xml

```
$ cp mapred-site.xml.template mapred-site.xml
$ vim mapred-site.xml
```

内容如下：

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

2.2.6 配置 yarn-site.xml

```
$ vim yarn-site.xml
```

内容如下：

```
<configuration>

<!-- Site specific YARN configuration properties -->

  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.resourcemanager.webapp.address</name>
    <value>Kylin:8099</value>
```

```
    </property>
    <property>
        <name>yarn.resourcemanager.hostname</name>
        <value>Kylin</value>
    </property>
</configuration>
```

2.2.7 配置 slaves

```
$ vim slaves
```

内容如下：

```
Kylin
```

3 格式化并启动集群

3.1 格式化 namenode

```
$ cd /usr/local/hadoop-2.7.7/
$ bin/hdfs namenode -format
```

3.2 启动 namenode 和 datanode 守护进程

```
$ sbin/start-dfs.sh
```

3.3 启动 ResourceManager 和 NodeManager 守护进程

```
$ sbin/start-yarn.sh
```

4 执行 wordcount 测试用例

```
$ bin/hdfs dfs -ls /

$ bin/hdfs dfs -mkdir /input
$ bin/hdfs dfs -put /usr/local/hadoop-2.7.7/README.txt /input

$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.7.jar wordcount /input /output

$ bin/hdfs dfs -ls /output
$ bin/hdfs dfs -cat /output/part-r-00000
```