# Hadoop集群搭建

# 目录

1	目的	. 2
2	先决条件	. 2
3	安装	. 2
4	配置	. 2
	4.1 配置文件	. 2
	4.2 集群配置	. 2
5	Hadoop的机架感知	. 6
6	启动Hadoop	. 7
7	停止Hadoop	. 7

# 1. 目的

本文描述了如何安装、配置和管理有实际意义的Hadoop集群,其规模可从几个节点的小集群到几千个节点的超大集群。

如果你希望在单机上安装Hadoop玩玩,从这里能找到相关细节。

- 2. 先决条件
- 1. 确保在你集群中的每个节点上都安装了所有必需软件。
- 2. 获取Hadoop软件包。

# 3. 安装

安装Hadoop集群通常要将安装软件解压到集群内的所有机器上。

通常,集群里的一台机器被指定为 NameNode, 另一台不同的机器被指定为JobTracker。这些机器是masters。余下的机器即作为DataNode也作为TaskTracker。这些机器是slaves。

我们用HADOOP\_HOME指代安装的根路径。通常,集群里的所有机器的HADOOP\_HOME路径相同。

# 4. 配置

接下来的几节描述了如何配置Hadoop集群。

#### 4.1. 配置文件

对Hadoop的配置通过conf/目录下的两个重要配置文件完成:

- 1. hadoop-default.xml 只读的默认配置。
- 2. hadoop-site.xml 集群特有的配置。

要了解更多关于这些配置文件如何影响Hadoop框架的细节,请看这里。

此外,通过设置conf/hadoop-env.sh中的变量为集群特有的值,你可以对bin/目录下的Hadoop脚本进行控制。

#### 4.2. 集群配置

要配置Hadoop集群,你需要设置Hadoop守护进程的运行环境和Hadoop守护进程的运行参数。

Hadoop守护进程指NameNode/DataNode 和JobTracker/TaskTracker。

#### 4.2.1. 配置Hadoop守护进程的运行环境

管理员可在conf/hadoop-env.sh脚本内对Hadoop守护进程的运行环境做特别指定。

至少,你得设定JAVA\_HOME使之在每一远端节点上都被正确设置。

管理员可以通过配置选项HADOOP\_\*\_OPTS来分别配置各个守护进程。 下表是可以配置的选项。

守护进程	
NameNode	HADOOP_NAMENODE_OPTS
DataNode	HADOOP_DATANODE_OPTS
SecondaryNamenode	HADOOP_SECONDARYNAMENODE_OPTS
JobTracker	HADOOP_JOBTRACKER_OPTS
TaskTracker	HADOOP_TASKTRACKER_OPTS

例如,配置Namenode时,为了使其能够并行回收垃圾(parallelGC), 要把下面的代码加入到hadoop-env.sh:

export HADOOP\_NAMENODE\_OPTS="-XX:+UsePara11e1GC \${HADOOP\_NAMENODE\_OPTS}"

其它可定制的常用参数还包括:

- HADOOP\_LOG\_DIR 守护进程日志文件的存放目录。如果不存在会被自动创建。
- HADOOP\_HEAPSIZE 最大可用的堆大小,单位为MB。比如,1000MB。 这个参数用于设置hadoop守护进程的堆大小。缺省大小是1000MB。

### 4.2.2. 配置Hadoop守护进程的运行参数

这部分涉及Hadoop集群的重要参数,这些参数在conf/hadoop-site.xm1中指定。

参数		备注
fs.default.name	NameNode的URI。	hdfs://主机名/
mapred.job.tracker	JobTracker的主机(或者IP)和	主机:端口。

	端口。	
dfs.name.dir	NameNode持久存储名字空间及事 务日志的本地文件系统路径。	当这个值是一个逗号分割的目录 列表时, nametable数据将会被 复制到所有目录中做冗余备份。
dfs.data.dir	DataNode存放块数据的本地文件 系统路径,逗号分割的列表。	当这个值是逗号分割的目录列表时,数据将被存储在所有目录下,通常分布在不同设备上。
mapred.system.dir	Map/Reduce框架存储系统文件的 HDFS路径。比如 /hadoop/mapred/system/。	这个路径是默认文件系统(HDFS )下的路径, 须从服务器和客 户端上均可访问。
mapred.local.dir	本地文件系统下逗号分割的路径列表,Map/Reducel临时数据存放的地方。	多路径有助于利用磁盘i/o。
mapred.tasktracker.{map reduc	某一TaskTracker上可运行的最大Map/Reduce任务数,这些任务将同时各自运行。	默认为2(2个map和2个reduce) ,可依据硬件情况更改。
dfs.hosts/dfs.hosts.exclude	许可/拒绝DataNode列表。	如有必要,用这个文件控制许可 的datanode列表。
mapred.hosts/mapred.hosts.exc	许可/拒绝TaskTracker列表。	如有必要,用这个文件控制许可 的TaskTracker列表。

通常,上述参数被标记为 final 以确保它们不被用户应用更改。

# 4.2.2.1. 现实世界的集群配置

这节罗列在大规模集群上运行sort基准测试(benchmark)时使用到的一些非缺省配置。

• 运行sort900的一些非缺省配置值, sort900即在900个节点的集群上对9TB的数据进行排序:

参数		备注
dfs.block.size	134217728	针对大文件系统,HDFS的块 大小取128MB。
dfs.namenode.handler.count	40	启动更多的NameNode服务线 程去处理来自大量DataNode 的RPC请求。

mapred.reduce.parallel.com	20	reduce启动更多的并行拷贝 器以获取大量map的输出。
mapred.child.java.opts	-Xmx512M	为map/reduce子虚拟机使用 更大的堆。
fs.inmemory.size.mb	200	为reduce阶段合并map输出 所需的内存文件系统分配更 多的内存。
io.sort.factor	100	文件排序时更多的流将同时 被归并。
io.sort.mb	200	提高排序时的内存上限。
io.file.buffer.size	131072	SequenceFile中用到的读/ 写缓存大小。

• 运行sort1400和sort2000时需要更新的配置,即在1400个节点上对14TB的数据进行 排序和在2000个节点上对20TB的数据进行排序:

参数		
mapred.job.tracker.handlen	60	启用更多的JobTracker服务 线程去处理来自大量 TaskTracker的RPC请求。
mapred.reduce.parallel.cop	50	
tasktracker.http.threads	50	为TaskTracker的Http服务 启用更多的工作线程。 reduce通过Http服务获取 map的中间输出。
mapred.child.java.opts	-Xmx1024M	使用更大的堆用于 maps/reduces的子虚拟机

#### 4.2.3. Slaves

通常,你选择集群中的一台机器作为NameNode,另外一台不同的机器作为JobTracker。余下的机器即作为DataNode又作为TaskTracker,这些被称之为slaves。

在conf/slaves文件中列出所有slave的主机名或者IP地址,一行一个。

#### 4.2.4. 日志

Hadoop使用<u>Apache log4.j</u>来记录日志,它由<u>Apache Commons Logging</u>框架来实现。编辑conf/log4.j.properties文件可以改变Hadoop守护进程的日志配置(日志格式等)。

#### 4.2.4.1. 历史日志

作业的历史文件集中存放在hadoop.job.history.location,这个也可以是在分布式文件系统下的路径,其默认值为\${HADOOP\_LOG\_DIR}/history。jobtracker的web UI上有历史日志的web UI链接。

历史文件在用户指定的目录hadoop.job.history.user.location也会记录一份,这个配置的缺省值为作业的输出目录。这些文件被存放在指定路径下的"\_logs/history/"目录中。因此,默认情况下日志文件会在"mapred.output.dir/\_logs/history/"下。如果将hadoop.job.history.user.location指定为值none,系统将不再记录此日志。

用户可使用以下命令在指定路径下查看历史日志汇总

\$ bin/hadoop job -history output-dir

这条命令会显示作业的细节信息, 失败和终止的任务细节。

关于作业的更多细节,比如成功的任务,以及对每个任务的所做的尝试次数等可以用下面的命令查看

\$ bin/hadoop job -history all output-dir

一但全部必要的配置完成,将这些文件分发到所有机器的HADOOP\_CONF\_DIR路径下,通常是\${HADOOP\_HOME}/conf。

# 5. Hadoop的机架感知

HDFS和Map/Reduce的组件是能够感知机架的。

NameNode和JobTracker通过调用管理员配置模块中的API<u>resolve</u>来获取集群里每个slave的机架id。该API将slave的DNS名称(或者IP地址)转换成机架id。使用哪个模块是通过配置项topology.node.switch.mapping.impl来指定的。模块的默认实现会调用topology.script.file.name配置项指定的一个的脚本/命令。 如果topology.script.file.name未被设置,对于所有传入的IP地址,模块会返回/default-rack作为机架id。在Map/Reduce部分还有一个额外的配置项mapred.cache.task.levels,该参数决定cache的级数(在网络拓扑中)。例如,如果

默认值是2, 会建立两级的cache— 一级针对主机(主机 -> 任务的映射)另一级针对机架(机架 -> 任务的映射)。

# 6. 启动Hadoop

启动Hadoop集群需要启动HDFS集群和Map/Reduce集群。

格式化一个新的分布式文件系统:

\$ bin/hadoop namenode -format

在分配的NameNode上,运行下面的命令启动HDFS:

\$ bin/start-dfs.sh

bin/start-dfs.sh脚本会参照NameNode上\${HADOOP\_CONF\_DIR}/slaves文件的内容,在 所有列出的slave上启动DataNode守护进程。

在分配的JobTracker上,运行下面的命令启动Map/Reduce:

\$ bin/start-mapred.sh

bin/start-mapred.sh脚本会参照JobTracker上\${HADOOP\_CONF\_DIR}/slaves文件的内容,在所有列出的slave上启动TaskTracker守护进程。

# 7. 停止Hadoop

在分配的NameNode上, 执行下面的命令停止HDFS:

\$ bin/stop-dfs.sh

bin/stop-dfs.sh脚本会参照NameNode上\${HADOOP\_CONF\_DIR}/slaves文件的内容,在所有列出的slave上停止DataNode守护进程。

在分配的JobTracker上,运行下面的命令停止Map/Reduce:

\$ bin/stop-mapred.sh

bin/stop-mapred.sh脚本会参照JobTracker上\${HADOOP\_CONF\_DIR}/slaves文件的内容,在所有列出的slave上停止TaskTracker守护进程。