Hadoop On Demand

目录

1. 概述

Hadoop On Demand (HOD)是一个能在一个共享集群上供应和管理相互独立的Hadoop Map/Reduce和Hadoop分布式文件系统 (HDFS) 实例的系统。它能让管理员和用户轻松地快速搭建和使用hadoop。HOD对Hadoop的开发人员和测试人员也非常有用，他们可以通过HOD共享一个物理集群来测试各自不同的Hadoop版本。

HOD依赖资源管理器(RM)来分配节点，这些节点被用来在之上运行hadoop实例。目前，HOD采用的是Torque资源管理器。

基本的HOD系统架构包含的下列组件：

- 一个资源管理器 (可能同时附带一个调度程序)
- 各种HOD的组件
- Hadoop Map/Reduce和HDFS守护进程

通过与以上组件交互，HOD在给定的集群上供应和维护Hadoop Map/Reduce实例，或者HDFS实例。集群中的节点可看作由两组节点构成：

- 提交节点(Submit nodes)：用户通过HOD客户端在这些节点上申请集群，之后通过Hadoop客户端提交Hadoop作业。
- 计算节点(Compute nodes)：利用资源管理器，HOD组件在这些节点上运行以供应Hadoop守护进程。之后，Hadoop作业在这些节点上运行。

下面是对申请集群及在之上运行作业所需操作步骤的简要描述。

- 用户在提交节点上用HOD客户端分配所需数目节点的集群，在上面供应Hadoop。
- HOD客户端利用资源管理器接口 (在Torque中是qsub) 提交一个被称为RingMaster的HOD进程作为一个资源管理器作业，申请理想数目的节点。这个作业被提交到资源管理器的中央服务器上 (在Torque中叫pbs_server) 。
- 在计算节点上，资源管理器的从(slave)守护程序(Torque中的pbs_moms)接受并处理中央服务器(Torque中的pbs_server)分配的作业。RingMaster进程在其中一个计算节点 (Torque中的mother superior) 上开始运行。
- 之后，Ringmaster通过资源管理器的另外一个接口(在Torque中是pbsdsh)在所有分配到的计算节点上运行第二个HOD组件HodRing，即分布式任务。
- HodRing初始化之后会与RingMaster通信获取Hadoop指令，并遵照执行。一旦Hadoop的命令开始启动，它们会向RingMaster登记，提供关于守护进程的信息。
- Hadoop实例所需的配置文件全部由HOD自己生成，有一些来自于用户在配置文件设置的选项。

- HOD客户端保持和RingMaster的通信，找出JobTracker和HDFS守护进程的位置所在。

之后的文档会讲述如何在一个物理集群的节点上安装HOD。

2．先决条件

要使用HOD，你的系统应包含下列的硬件和软件

操作系统：HOD目前在RHEL4上测试通过。
节点：HOD至少需要3个由资源管理器配置的节点。

软件

在使用HOD之前，以下组件必须被安装到所有节点上：

- Torque：资源管理器
- Python：HOD要求Python 2.5.1

下列组件是可选的，你可以安装以获取HOD更好的功能：

- Twisted Python：这个可以用来提升HOD的可扩展性。如果检测到这个模块已安装，HOD就用它，否则就使用默认的模块。
- Hadoop：HOD能自动将Hadoop分发到集群的所有节点上。不过，如果Hadoop在所有节点上已经可用，HOD也可以使用已经安装好的Hadoop。HOD目前支持Hadoop 0.15和其后续版本。

注释：HOD的配置需要以上这些组件的安装位置在集群所有节点上保持一致。如果在提交节点上的安装位置也相同，配置起来会更简单。

3．资源管理器

目前，HOD使用Torque资源管理器来分配节点和提交作业。Torque是一个开源的资源管理器，来自于Cluster Resources，是一个社区基于PBS项目努力的结晶。它提供对批处理作业和分散的计算节点(Compute nodes)的控制。你可以自由地从此处下载Torque。

所有torque相关的文档可以在这儿的TORQUE Resource Manager一节找到。在这里可以看到wiki文档。如果想订阅TORQUE的邮件列表或查看问题存档，访问这里。

使用带Torque的HOD：

- 安装Torque组件：在一个节点上(head node)安装pbs_server，所有计算节点上安装pbs_mom，所有计算节点和提交节点上安装PBS客户端。至少做最基本的配置，使Torque系统跑起来，也就是，使pbs_server能知道该和哪些机器通话。查看这里可以了解基本配置。要了解高级配置，请查看这里。
- 在pbs_server上创建一个作业提交队列。队列的名字和HOD的配置参数resource-manager.queue相同。Hod客户端利用此队列提交RingMaster进程作为Torque作业。
- 在集群的所有节点上指定一个cluster name作为property。这可以用qmgr命令做到。比如：qmgr -c "set node node properties=cluster-name"。集群名字和HOD的配置参数hod.cluster是相同的。
- 确保作业可以提交到节点上去。这可以通过使用qsub命令做到。比如：echo "sleep 30" | qsub -l nodes=3

4. 安装HOD

现在资源管理器已经安装好了，我们接着下载并安装HOD。

- 如果你想从Hadoop tar包中获取HOD，它在'contrib'下的'hod'的根目录下。
- 如果你从编译源码，可以在Hadoop根目录下的运行ant tar，生成Hadoop tar包。然后从获取HOD，参照上面。
- 把这个目录下的所有文件分发到集群的所有节点上。注意文件拷贝的位置应在所有节点上保持一致。
- 注意，编译hadoop时会创建HOD，同时会正确地设置所有HOD必须的脚本文件的权限。

5. 配置HOD

安装HOD后你就可以配置它。为了运行HOD需要做的最小配置会在下面讲述，更多高级的配置会在HOD配置指南里面讲解。

5.1. 最小配置

为运行HOD，以下的最小配置是必须要做的：

- 在你想要运行hod的节点上，编辑<install dir>/conf目录下的hodrc文件。这个文件包含了运行hod所必需的最少量的设置。
- 为这个配置文件中的定义的变量指定适合你环境的值。注意，有些变量在文件中出现了不止一次。

  - ${JAVA_HOME}：Hadoop的Java的安装位置。Hadoop支持Sun JDK 1.5.x及以上版本。
  - ${CLUSTER_NAME}：集群名称，由'node property'指定，在资源管理器配置中曾提到过。
  - ${HADOOP_HOME}：Hadoop在计算节点和提交节点上的安装位置。
  - ${RM_QUEUE}：在资源管理器配置中设置的作业提交队列。
  - ${RM_HOME}：资源管理器在计算节点和提交节点的安装位置。
- 以下环境变量可能需要设置，取决于你的系统环境。在你运行HOD客户端的地方这些变量必须被定义，也必须在HOD配置文件中通过设定resource_manager.env-vars的值指定。多个变量可指定为用逗号分隔的key=value对组成的列表。
  - HOD_PYTHON_HOME：如果python安装在计算节点或提交节点的非默认位置，那么这个值必须设定为python的可执行文件的实际位置。

5.2. 高级配置

你可以检查和修改其它配置选项来满足你的特定需要。关于HOD配置的更多信息，请参考配置指南。

6. 运行HOD

当HOD配置好后，你就可以运行它了。更多信息请参考HOD用户指南。

7. 支持工具和实用程序

此节描述一些可用于管理HOD部署的支持工具和应用程序。

7.1. logcondense.py - 管理日志文件

在HOD用户指南有提到，HOD可配置成将Hadoop日志上传到一个配置好的静态HDFS上。随着时间增加，日志数量会不断增长。logcondense.py可以帮助管理员清理上传到HDFS的日志文件。

7.1.1. 运行logcondense.py

logcondense.py在hod_install_location/support文件夹下。你可以使用python去运行它，比如python logcondense.py，或者授以执行权限，直接运行logcondense.py。

如果启用了权限，logcondense.py需要被有足够权限，能删除HDFS上上传目录下日志文件的用户运行。比如，在配置指南中提及过，用户可以配置将日志放在HDFS上的其主目录下。在这种情况下，你需要具有超级用户权限，才能运行logcondense.py删除所有用户主目录下的日志文件。

7.1.2. logcondense.py的命令行选项

logcondense.py支持以下命令行选项

短选项	长选项	含义	例子
-p	--package	hadoop脚本的全路径。Hadoop的版本必须和运行HDFS的版本一致。	/usr/bin/hadoop
-d	--days	删除超过指定天数的日志文件	7
-c	--config	Hadoop配置目录的路径，hadoop-site.xml存在于此目录中。hadoop-site.xml中须指明待删除日志存放的HDFS的NameNode。	/home/foo/hadoop/conf
-l	--logs	一个HDFS路径，须和log-destination-uri指定的是同一个HDFS路径，不带hdfs:// URI串，这点在配置指南中提到过。	/user
-n	--dynamicdfs	如果为true，logcondense.py除要删除Map/Reduce日志之外还需删除HDFS日志。否则，它只删除Map/Reduce日志，这也是不指定这个选项时的默认行为。这个选项对下面的情况非常有用：一个动态的HDFS由HOD供应，一个静态的HDFS用来收集日志文件 -	false

		也许这是测试集群中一个非常普遍的使用场景。	

比如，假如要删除所有7天之前的日志文件，hadoop-site.xml存放在~/hadoop-conf下，hadoop安装于~/hadoop-0.17.0，你可以这样：

```
python logcondense.py -p ~/hadoop-0.17.0/bin/hadoop -d 7 -c ~/hadoop-conf
-l /user
```

7.2. checklimits.sh - 监视资源限制

checklimits.sh是一个针对Torque/Maui环境的HOD工具（Maui集群调度器 是一个用于集群和超级计算机的开源作业调度器，来自clusterresourcces）。当新提交的作业违反或超过用户在Maui调度器里设置的限制时，checklimits.sh脚本更新torque的comment字段。它使用qstat在torque的job-list中做一次遍历确定作业是在队列中还是已完成，运行Maui工具checkjob检查每一个作业是否违反用户限制设定，之后运行torque的qalter工具更新作业的'comment'的属性。当前，它把那些违反限制的作业的comment的值更新为User-limits exceeded. Requested:([0-9]*) Used:([0-9]*) MaxLimit:([0-9]*)。之后，HOD根据这个注释内容做出相应处理。

7.2.1. 运行checklimits.sh

checklimits.sh可以在hod_install_location/support目录下下找到。在具有得执行权限后，这个shell脚本可以直接通过sh checklimits.sh 或者./checklimits.sh运行。这个工具运行的机器上应有Torque和Maui的二进制运行文件，并且这些文件要在这个shell脚本进程的路径中。为了更新不同用户作业的comment值，这个工具必须以torque的管理员权限运行。这个工具必须按照一定时间间隔重复运行，来保证更新job的约束条件，比如可以通过cron。请注意，这个脚本中用到的资源管理器和调度器命令运行代价可能会比价大，所以最好不要在没有sleeping的紧凑循环中运行。

7.3. verify-account - 用于核实用户提交作业所使用的帐号的脚本

生产系统一般使用帐号系统来对使用共享资源的用户收费。HOD支持一个叫resource_manager.pbs-account的参数，用户可以通过这个参数来指定提交作业时使用的帐号。核实这个帐户在帐号管理系统中的有效性是有必要的。脚本hod-install-dir/bin/verify-account提供了一种机制让用户插入自定义脚本来实现这个核实过程。

7.3.1. 在HOD中集成verify-account

在分配集群之前，HOD运行verify-account脚本，将resource_manager.pbs-account的值作为参数传递给用户自定义脚本来完成用户的确认。系统还可以通过这种方式来取代它本身的帐号系统。若该用户脚本中的返回值非0，就会导致HOD分配集群失败。并且在发生错误时，HOD还会将脚本中产生的错误信息打印出来。通过这种方式，任何描述性的错误信息都可以从用户脚本中返回给用户。

在HOD中自带的默认脚本是不做任何的用户核实，并返回0。

如果HOD没有找到上面提到的verify-account脚本，HOD就会认为该用户核实的功能被关闭，然后继续自己以后的分配工作。