

Apache Hadoop 3.0.0-beta1 正式发布，下一个版本(GA)即可在线上使用

就在前几天，Apache Hadoop 3.0.0-beta1 正式发布了，这是3.0.0的第一个 beta 版本。本版本基于 3.0.0-alpha4 版本进行了Bug修复、性能提升以及其他一些加强。好消息是，这个版本之后会正式发行 Apache Hadoop 3.3.0 GA (General Availability,正式发布的版本) 版本，这意味着我们就可以正式在线上使用 Hadoop 3.0.0 了！目前预计 Apache Hadoop 3.3.0 GA 将会在 2017-11-01 正式发布

值得注意的是 Apache Hadoop 3.0.0-beta1 的 API 是稳定的，但是不保证它的质量（quality），不推荐在线上使用（等GA版本吧）。Apache Hadoop 3.0.0-beta1 相对于hadoop-2.x来说包含了许多重要的改进。下面是 Hadoop 3.0.0 的一些重要的特性介绍。



如果想及时了
解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

Java最低版本要求从Java7 更改成Java8

所有的Hadoop JARs都是针对Java 8 编译的。仍在用Java 7 或更低版本的用户必须升级至Java 8。

HDFS支持纠删码(Erasure Coding)

与副本相比纠删码是一种更节省空间的数据持久化存储方法。标准编码(比如Reed-Solomon(10,4))会有1.4 倍的空间开销；然而HDFS副本则会有3倍的空间开销。因为纠删码额外开销主要是在重建和执行远程读，它传统用于存储冷数据，即不经常访问的数据。当部署这个新特性时用户应该考虑纠删码的网络和CPU开销。更多关于HDFS的纠删码可以参见<http://hadoop.apache.org/docs/r3.0.0-beta1/hadoop-project-dist/hadoop-hdfs/HDFSErasureCoding.html>或者直接阅读本博客《[Hadoop 3.0纠删码\(Erasure Coding\)：节省一半存储空间](#)》的相关介绍。

YARN Timeline Service v.2

本版本引入了Yarn时间轴服务v.2，主要用于解决2大挑战：改善时间轴服务的可伸缩性和可靠性，通过引入流和聚合增强可用性。

YARN Timeline Service v.2 alpha

1可以让用户和开发者测试以及反馈，以便使得它可以替换现在的Timeline Service v.1.x。请在测试环境中使用。更多关于YARN Timeline Service v.2的知识请参见<http://hadoop.apache.org/docs/r3.0.0-beta1/hadoop-yarn/hadoop-yarn-site/TimelineServiceV2.html>

Shell脚本重写

Hadoop的Shell脚本被重写解决了之前很多长期存在的bug，并且引入了一些新的特性。绝大部分都保持兼容性，不过仍有些变化可能使得现有的安装不能正常运行。不兼容的改变可以参见HADOOP-9902。更多内容请参见[Unix Shell Guide](#)文档。即使你是资深用户，也建议看下这个文档，因为其描述了许多新的功能，特别是与可扩展性有关的功能。

Shaded client jars

在 Hadoop 2.x 版本，hadoop-client Maven artifact将 Hadoop 所有的依赖都加到 Hadoop 应用程序的环境变量中，这样可能会导致应用程序依赖的类和 Hadoop 依赖的类有冲突。这个问题在 HADOOP-11804 得到了解决。

支持 Opportunistic Containers 和分布式调度

Opportunistic Container引入新 Opportunistic 类型的 Container 后，这种 Container 可以利用节点上已分配但未真正使用的资源。原有 Container 类型定义为 Guaranteed 类型。相对于 Guaranteed 类型Container，Opportunistic 类型的Container优先级更低。

MapReduce任务级本地优化

MapReduce添加了Map输出collector的本地实现。对于shuffle密集型的作业来说，这将会有30%

以上的性能提升。更多内容请参见 MAPREDUCE-2841

支持多于2个的NameNodes

最初的HDFS NameNode high-availability实现仅仅提供了一个active NameNode和一个Standby NameNode；并且通过将编辑日志复制到三个JournalNodes上，这种架构能够容忍系统中的任何一个节点的失败。然而，一些部署需要更高的容错度。我们可以通过这个新特性来实现，其允许用户运行多个Standby NameNode。比如通过配置三个NameNode和五个JournalNodes，这个系统可以容忍2个节点的故障，而不是仅仅一个节点。[HDFS high-availability](#) 文档已经对这些信息进行了更新，我们可以阅读这篇文档了解如何配置多于2个NameNodes。

多个服务的默认端口被改变

在此之前，多个Hadoop服务的默认端口都属于Linux的临时端口范围（32768-61000）。这就意味着我们的服务在启动的时候可能因为和其他应用程序产生端口冲突而无法启动。现在这些可能会产生冲突的端口已经不再属于临时端口的范围，这些端口的改变会影响NameNode, Secondary NameNode, DataNode以及KMS。与此同时，官方文档也进行了相应的改变，具体可以参见HDFS-9427以及HADOOP-12811。下面表格列出了端口变化的情况

Daemon	App	Hadoop 2.x Port	Hadoop 3 Port
NameNode Ports	Hadoop HDFS NameNode	8020	9820
	NameNode HTTP UI	50070	9870
	NameNode HTTPS UI	50470	9871
Secondary NN Ports	Secondary NameNode HTTP	50091	9869
	Secondary NameNode HTTP UI	50090	9868
DataNode Ports	DataNode IPC	50020	9867
	DataNode	50010	9866
	DataNode HTTP UI	50075	9864
	DataNode HTTPS UI	50475	9865

支持Microsoft Azure Data Lake filesystem连接器

Hadoop现在支持集成Microsoft Azure Data Lake，并作为替代Hadoop默认的文件系统。

Intra-datanode均衡器

一个DataNode可以管理多个磁盘，正常写入操作，各磁盘会被均匀填满。然而，当添加或替换磁盘时可能导致此DataNode内部的磁盘存储的数据严重内斜。这种情况现有的HDFS balancer是无法处理的。这种情况是由新intra-DataNode平衡功能来处理，通过hdfs diskbalancer CLI来调用。更多请参考[HDFS Commands Guide](#)

重写守护进程以及任务的堆内存管理

Hadoop守护进程和MapReduce任务的堆内存管理发生了一系列变化。

HADOOP-10950：介绍了配置守护集成heap大小的新方法。主机内存大小可以自动调整，HADOOP_HEAPSIZE 已弃用。

MAPREDUCE-5785：map和reduce task堆大小的配置方法，所需的堆大小不再需要通过任务配置和Java选项实现。已经指定的现有配置不受此更改影响。

S3Guard：S3A文件系统客户机的一致性和元数据缓存

HADOOP-13345 里面为 Amazon S3 存储系统的 S3A

客户端引入了一个新的可选特性，也就是可以使用 DynamoDB 表作为文件和目录元数据的快速一致的存储。

本博客文章除特别声明，全部都是原创！

转载本文请加上：转载自过往记忆 (<https://www.iteblog.com/>)

本文链接: 【】 ()