

1. 新特性

- 最低Java版本要求从Java7变为Java8
 - 所有Hadoop的jar都是基于Java 8运行是版本进行编译执行的，仍在使用Java 7或更低Java版本的用户需要升级到Java 8。
- HDFS支持纠删码 (erasure coding)
 - 纠删码是一种比副本存储更节省存储空间的数据持久化存储方法。比如Reed-Solomon(10,4)标准编码技术只需要1.4倍的空间开销，而标准的HDFS副本技术则需要3倍的空间开销。由于纠删码额外开销主要在于重建和远程读写，它通常用来存储不经常使用的数据（冷数据）。另外，在使用这个新特性时，用户还需要考虑网络和CPU开销。
 - Hadoop 2.x - 可以通过复制（浪费空间）来处理容错。
Hadoop 3.x - 可以通过Erasure编码处理容错。
- YARN时间线服务 v.2(YARN Timeline Service v.2)
 - YARN Timeline Service v.2用来应对两个主要挑战：（1）提高时间线服务的可扩展性、可靠性，（2）通过引入流(flow)和聚合(aggregation)来增强可用性。为了替代Timeline Service v.1.x，YARN Timeline Service v.2 alpha 2被提出来，这样用户和开发者就可以进行测试，并提供反馈和建议，不过YARN Timeline Service v.2还只能用在测试容器中。
 - 在hadoop2.4版本之前对任务执行的监控只开发了针对MR的**Job History Server**，它可以提供给用户用户查询已经运行完成的作业的信息，但是后来，随着在YARN上面集成的越来越多的计算框架，比如spark、Tez，也有必要为基于这些计算引擎的技术开发相应的作业任务监控工具，所以hadoop的开发人员就考虑开发一款更加通用的Job History Server，即YARN Timeline Server
- 重写Shell脚本
 - Hadoop的shell脚本被重写，修补了许多**长期存在的bug**，并增加了一些新的特性（**what特性**）。
- 覆盖客户端的jar (Shaded client jars)
 - 在2.x版本中，hadoop-client Maven artifact配置将会拉取hadoop的传递依赖到hadoop应用程序的环境变量，这回带来传递依赖的版本和应用程序的版本相冲突的问题。
 - HADOOP-11804 添加新 hadoop-client-api和hadoop-client-runtime artifcat，将hadoop的依赖隔离在一个单一Jar包中，也就避免hadoop依赖渗透到应用程序的类路径中。
- 支持Opportunistic Containers和Distributed Scheduling
 - ExecutionType概念被引入，这样一来，应用能够通过Opportunistic的一个执行类型来请求容器。即使在调度时，没有可用的资源，这种类型的容器也会分发给NM中执行程序。在这种情况下，容器将被放入NM的队列中，等待可用资源，以便执行。Opportunistic container优先级要比默认Guaranteedcontainer低，在需要的情况下，其资源会被抢占，以便Guaranteed container使用。这样就需要提高集群的使用率。

Opportunistic container默认被中央RM分配，但是，目前已经增加分布式调度器的支持，该分布式调度器做为AMRProtocol解析器来实现。
- MapReduce任务级本地优化
 - MapReduce添加了映射输出收集器的本地化实现的支持。对于密集型的洗牌操作（shuffle-intensive）jobs，可以带来30%的性能提升。
- 支持多余2个以上的NameNodes

- 针对HDFS NameNode的高可用性，最初实现方式是提供一个活跃的（ active ）NameNode和一个备用的（ Standby ）NameNode。通过对3个JournalNode的法定数量的复制编辑，使得这种架构能够对系统中任何一个节点的故障进行容错。

该功能能够通过运行更多备用NameNode来提供更高的容错性，满足一些部署的需求。比如，通过配置3个NameNode和5个JournalNode，集群能够实现两个节点故障的容错。
- 修改了多重服务的默认端口
 - 在之前的Hadoop版本中，多重Hadoop服务的默认端口在Linux临时端口范围内容（ 32768-61000 ），这就意味着，在启动过程中，一些服务器由于端口冲突会启动失败。这些冲突端口已经从临时端口范围移除，NameNode、Secondary NameNode、DataNode和KMS会受到影响。我们的文档已经做了相应的修改，可以通过阅读发布说明 **HDFS-9427**和**HADOOP-12811**详细了解所有被修改的端口
- 提供文件系统连接器（ filesystem connector ）,支持Microsoft Azure Data Lake和Aliyun对象存储系统
 - Hadoop支持和Microsoft Azure Data Lake和Aliyun对象存储系统集成，并将其作为Hadoop兼容的文件系统
- **数据节点内置平衡器（ Intra-datanode balancer ）**
 - 在单一DataNode管理多个磁盘情况下，在执行普通的写操作时，每个磁盘用量比较平均。但是，当添加或者更换磁盘时，将会导致一个DataNode磁盘用量的严重不均衡。由于目前**HDFS均衡器**关注点在于DataNode之间（ inter- ），而不是intra-，所以不能处理这种不均衡情况。

在hadoop3 中，通过DataNode内部均衡功能Intra-data节点平衡器已经可以处理上述情况，可以通过**hdfs diskbalancer Cli**来调用。
- 重写了守护进程和任务的堆管理机制
 - 针对Hadoop**守护进程**和MapReduce**任务的堆管理机制**，Hadoop3 做了一系列的修改。

HADOOP-10950 引入配置守护进程堆大小的新方法。特别地，HADOOP_HEAPSIZE配置方式已经被弃用，可以根据主机的内存大小进行自动调整。

MAPREDUCE-5785 简化了MAP的配置，减少了任务堆的大小，所以不需要再任务配置和Java可选项中明确指出需要的堆大小。已经明确指出堆大小的现有配置不会受到该改变的影响。
- S3Gurad:为S3A文件系统客户端提供一致性和元数据缓存
 - HADOOP-13345 为亚马逊S3存储的S3A客户端提供了可选特性：能够使用DynamoDB表作为文件和目录元数据的快速、一致性存储。
- HDFS的基于路由器互联（ HDFS Router-Based Federation ）
 - HDFS Router-Based Federation添加了一个**RPC路由层**，为**多个HDFS命名空间**提供了一个联合视图。这和现有的ViewFs、HDFS Federation功能类似，区别在于通过服务端管理表加载，而不是原来的客户端管理。从而简化了现存HDFS客户端接入federated cluster的操作。
- 基于API配置的Capacity Scheduler queue configuration
 - OrgQueue扩展了capacity scheduler，提供了一种编程方法，该方法提供了一个REST API来修改配置，用户可以通过远程调用来修改队列配置。这样一来，队列的administer_queue ACL的管理员就可以实现自动化的队列配置管理。 TODO 实践 **how?**
- **YARN资源类型**
 - Yarn资源模型已经被一般化，可以支持用户自定义的可计算资源类型，而不仅仅是CPU和内存。比如，集群管理员可以定义像GPU数量，软件序列号、本地连接的存储的资源。然后，Yarn任务能够

在这些可用资源上进行调度。TODO how ?

如何选择版本

- 是否为开源软件 TODO，即是否免费。
- 是否有稳定版 TODO，这个一般软件官方网站会给出说明。
- 是否经实践验证 TODO，这个可通过检查是否有一些大点的公司已经在生产环境中使用知道
- 是否有强大的社区支持 TODO，当出现一个问题时，能够通过社区、论坛等网络资源快速获取解决方法。