

## Hadoop 2.7 不停服升级到 3.2 在滴滴的实践

### 为什么要升级

在2017年底，Hadoop3.0发布了，到目前为止，Hadoop发布的最新版本为3.2.1。在Hadoop3中有很多有用的新特性出现，如支持ErasureCoding、多NameNode、Standby NameNode read、DataNode Disk Balance、HDFS RBF等等。除此之外，还有很多性能优化以及BUG修复。

其中最吸引我们的就是ErasureCoding特性，数据可靠性保持不变的情况下可以降低数据的存储副本数量，结合公司的降成本目标以及用户的痛点，我们对此做了深入的调研。同时，在实际工作中我们发现，我们遇到的一些BUG以及想做的一些优化点，社区早已经修复或者实现。内部使用的Hadoop版本对应社区的2.7.2，由于社区很多BUG修复是不会移植到太低版本的，我们解决问题时花费了较多精力在移植与测试验证中。

如果升级到HDFS3.2版本，可以站在巨人肩膀上继续工作，做一些更有意义的事情。

### 调研升级方案

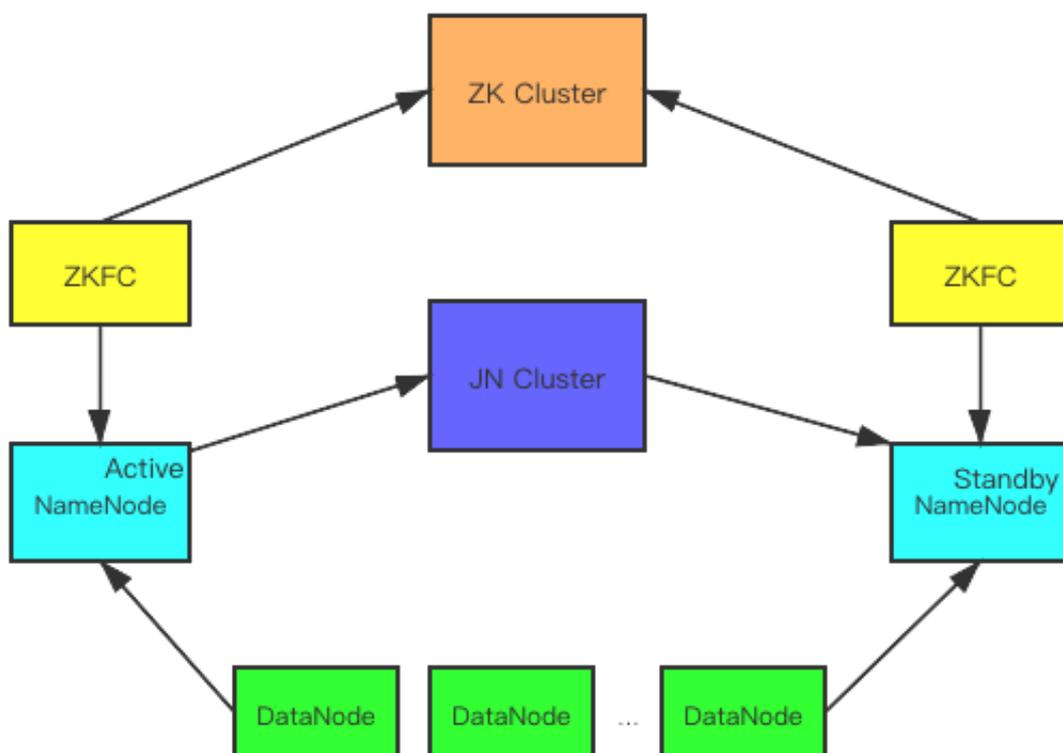
升级方式有两种：Express和Rolling，Express

升级过程是停止现有服务，然后使用新版本启动服务；Rolling升级过程是滚动升级，不停服务，对用户无感知。对于公司来说，当然滚动升级是最好的方案，离线集群用户非常之多，影响面非常大。

目前业界还没有滚动升级的方案从2.x版本升级到3.x版本，Cloudera和Hortonworks公司（目前两个公司已合并）给出的推荐方案仍然是Express升级，例如Hortonworks的文档中描述，目前滚动升级存在一些问题尚未解决，推荐用户做Express升级。

当前滚动升级存在的问题记录在Apache Hadoop Wiki中，主要问题是Edit Log不兼容，无法进行滚动升级。调研之后，我们对整个升级方案有了一个初步掌握，开始着手解决这些问题。

HDFS整体架构图（网络上获取）如下所示，我们准备对服务端进行升级，包括JournalNode，NameNode，ZKFC，DataNode组件。Client端受到Spark，Hive，Flink等等很多组件依赖，目前这些组件还不支持Hadoop3，因此Client版本暂时保持不变。



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

## 解决滚动升级中遇到的问题

滚动升级的操作流程在 Hadoop 官方升级文档中有介绍，概括起来大致步骤如下：

- JournalNode 升级，使用新版本依次重启 JournalNode
- NameNode 升级
  - 升级准备，生成 fallback fsimage 文件
  - 使用新版本 Hadoop 重启 Standby NameNode，重启 ZKFC
  - 做 failover，使升级后的 NameNode 变成 Active 节点
  - 使用新版本 Hadoop 重启另一个 NameNode，重启 ZKFC
- 升级 DataNode，使用新版本 Hadoop 重启所有 DataNode 节点
- 做 Finalize，确认集群变更到3.2

在测试环境验证 HDFS 滚动升级方案时，升级和降级过程中都遇到了一些问题。

在滚动升级中，当 Active NameNode 为3.2版本，Standby NameNode 为2.7版本时，会出现 EditLog 不兼容问题。此时，Active NameNode 写 EditLog 时会把 EC 相关的结构写入到 EditLog 当中，当 Standby NameNode 读取 EditLog 时，会出现识别不了的情况，导致 Standby NameNode 直接 Shutdown。我们的解决方案是，考虑当前有效版本是否支持 EC，如果支持 EC 则会写入 EC 信息到

EditLog，否则不会写入。而在升级过程中，有效版本实际上还是2.7，是不支持 EC 的，这个时候忽略 EC 即可，这样 Standby NameNode 读取 EditLog 做合并时，不会出现 EC 相关信息，可正常工作。解决问题的 ISSUE 为 HDFS-13596。

在滚动降级中，当3.2版本的 NameNode 使用3.2版本 Hadoop 重启时，如果当前最新的 Fsimage 是3.2版本 NameNode 产生的，则2.7版本 Hadoop 重启 NameNode 会直接 Shutdown，原因是，3.2版本 Hadoop 产生的 Fsimage 文件，2.7版本的 Hadoop 无法进行加载，这将导致如果升级中遇到问题想回滚的话，无法完成回滚操作。经过深入分析，我们发现有两个问题会导致这种情况出现。

第一个问题，Fsimage 的不兼容是由于3.2版本的 NameNode 将 EC 信息写入到了 Fsimage 当中，2.7版本的 Hadoop 无法识别 EC 信息，导致失败。解决方案与上面类似，在保存 Fsimage 时考虑当前的有效版本，如果不支持 EC 则不会将 EC 信息写入到 Fsimage 文件中。解决问题的 ISSUE 为 HDFS-14396。

第二个问题，由于 NameNode 对 StringTable 的修改导致了 Fsimage 的不兼容，目前该问题可以通过回滚 commit 进行解决，社区反馈修复也不是很必要，可以通过先升级到无该 commit 的版本，滚动升级稳定后，直接进行小版本升级，跨过这个不兼容特性。记录 ISSUE 为 HDFS-14831。

由于滴滴使用的是内部的用户名密码认证机制，社区出现的一个问题我们没有遇到，ISSUE 为 HDFS-14509，升级过程中 NameNode 和 DataNode 由于数据结构的变化，生成了不同的 password，导致无法认证，读写数据会失败。该 ISSUE 记录了这个问题，需要先升级到 2.x 的最新版本进行过度，之后才能滚动升级到 3.x 版本。

总结起来，需要做 HDFS2.x 到 3.x 的滚动升级，需要关注这些 ISSUE，HDFS-13596，HDFS-14396，HDFS-14831，HDFS-14509。

## 测试与上线

从19年初开始关注 HDFS 滚动升级，在解决遇到的已知问题之后，开发与测试不断讨论升级方案，将可能遇到的风险进行总结。

在这个过程中，我们详细阅读分析了滚动升级的源码，确定升级中 NameNode，DataNode 会做哪些动作，以明确风险点。同时我们还分析了从2.7到3.2版本引入的关于 HDFS 的4000左右的 Patch

，找出可能存在兼容性问题的点，进行深入地分析。同时我们对3.2中新引入的 Feature 也进行了分析，以确保新功能对升级没有影响。种种总结、分析、测试相关的工作，我们写了四五十篇的 WIKI 文档进行记录。在测试环境中升级步骤进行了数次演练，确认没问题之后，我们开始了升级之路。相关的具体里程碑上线过程如下：

- 19年5月左右，升级演练多次，准备全量 Hadoop、Hive、Spark Case 进行测试，确定方案没有问题
- 19年7月左右，离线小集群1（百台）升级到3.2版本，用户未受到影响。

- 19年10月左右，离线小集群2（数百台）升级到3.2版本，用户未受到影响。
- 19年11月底，离线大集群（数千台）升级到3.2版本，用户未受到影响。

升级过程中，DataNode 在删除 Block 时，是不会真的将 Block 删除的，而是先将Block 文件放到一个 Trash 目录中，为了能够使用原来的 FallBack Fsimage 恢复以前的数据。当升级周期比较长时，Trash 中的数据就会很多，例如我们这边大集群升级周期就有3周之长。升级操作在短时间之内，是可以确定是否有问题的，并且三周之后也不可能真的回滚到以前的数据，倘若真的遇到问题，是需要及时修复的。我们开发了额外的工具，对 Trash 中的 Block 文件进行按天归档，设置好保留时间，例如设置1天。我们会每天例行将1天之前的数据进行删除，这样可以大大减少 DataNode 上磁盘的存储压力。

升级之后，我们对各个集群进行都进行自己观察，目前服务一切正常。

## 总结

非常高兴在如此大规模的集群上完成从2.7到3.2的滚动升级，走在了行业的前列。HDFS 升级过程漫长，但是收益是非常多的。在此基础上，我们可以继续做非常有意义的工作，持续在稳定性、性能、成本等多个方面深入探索，使用技术为公司创造可见的价值。

本文作者费辉，滴滴出行大数据架构技术专家，负责离线存储。在加入滴滴之前，曾在阿里巴巴参与过JVM和EMR产品的开发。开源大数据爱好者，积极参与社区的交流讨论，Hadoop/Hive/Tez 社区贡献者。

**本博客文章除特别声明，全部都是原创！**  
**原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。**  
**本文链接: [【】（）](#)**