

前言

大数据组件之版本升级问题，组件迭代这么快（举个例子，Flink 从 1.8 到 1.9，再到 1.10，这几个版本功能都有很大的变化，版本更新的时间也很快；ElasticSearch 从 5.x、6.x、7.x 版本，速度简直飞快），公司使用是否要跟上社区的版本？如果要跟上最新的版本那么升级方案又该怎样呢？

是否要跟上最新的版本？

关于是否要使用最新的版本，其实这个通常还是要分析现在的场景是否真的只能在新版本才支持，比如新版本中的某个功能点刚好可以解决我们线上的某种需求，或者能够解决之前版本的重大 bug。

这里拿我现在公司举例，我从进入公司到现在，快 6 个月了，这六个月内，我们的 Flink 版本从 1.7.2 --> 1.9.1 --> 1.10.0，可以说是升级迭代的频率对于很多公司来说是比较快的，那么我们为什么要这么快速度的更新版本呢？难道是为了 KPI？

先说下目前我们线上的情况，50% 左右的作业都已经是 1.10.0，仍有 50% 左右的作业是 1.9.1，1.7.2 版本的作业已经没有。

其实推动 1.10.0 版本已经已经有差不多 1 个月了，但是为啥现在 1.10.0 版本的作业还是这么少呢？其中有几点原因我想了一下：

- 线上的流作业很稳定，不方便把作业停了，升级一下版本再启动，其他业务线的用户才不管版本到底是啥，我的作业运行的好好的，这不是很稳定吗？我不需要升级，目前这个版本已经能解决我的问题，如果强行升级，我还要变更代码的 Flink 版本，还要操心升级后是否依旧能之前一样运行好好的？安稳点不好吗？
- 新版本的 API 和之前版本有大的变动，1.9.1 版本的作业直接跑在 1.10.0 上，会出现包冲突、找不到 class、启动不起来等各种异常，无奈，业务方他也想配合我们组件这边做升级改造，但是出现这种异常我只好依旧回滚使用 1.9.1 版本启动我的老作业。

但是为啥线上还有 50% 的作业是新版本的啊？其实这部分作业主要是新上线的作业，就直接推动说使用新版本了。

回到正题（我们是否要跟上最新的版本），其实这么快上线 Flink 1.10 的主要原因是因为 **Flink 1.10 SQL 功能的完善，我们公司的 SQL 平台需要 1.10.0 版本的很多功能才能够支持**，所以也是为啥我这边急着上线 1.10.0 版本，也没有等到 1.10.1 版本（通常在 1.10.0 版本上修复不少 bug，稍稳定）。

1.10.0 版本对比 1.9 版本来说，SQL 功能确实有很大的完善，除此之外，也有不少变动，比如 TaskManager 的内存管理分配也是一个很大的变动，假设你原先在 1.9 版本给一个 TaskManager 分配 10GB 的内存，他会很大部分都显示是在堆内存，但是在 1.10 中，你可能会发现分配到堆内存中怎么才 5GB 左右？其实这就是在 1.10 版本中的 TaskManager 的内存变化，它现在分的很细，每个组成部分的名称你可能之前都没注意到过的。

调研

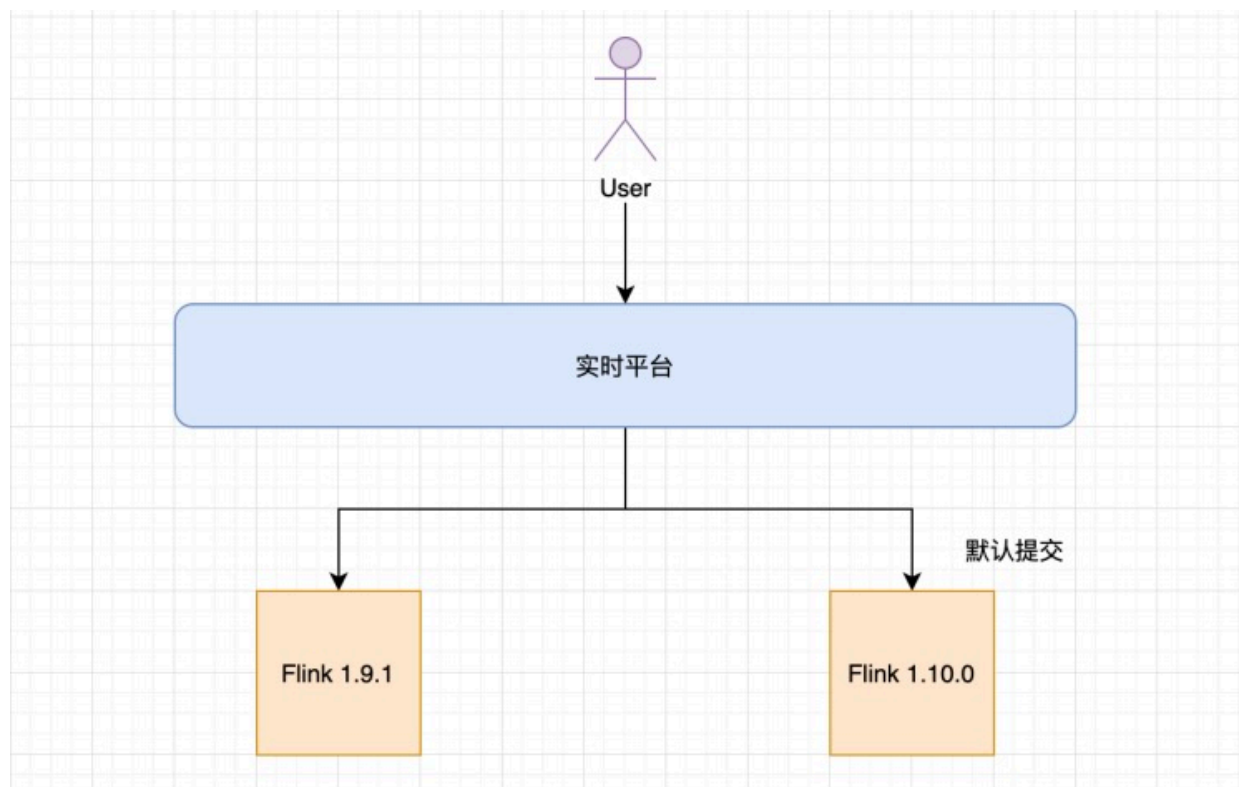
在升级一个组件，首先得先去对新版本做个调研，清楚新版本有什么新功能，修复了什么历史 bug，通常这个调研的工作可以参考官网新版本的 Release Note 就可以知道会有什么大的 feature 以及修复了什么历史问题。另外关于新版本的稳定性或者新的 bug 可能就得去多留意社区的一些邮件列表或者交流群里面大家提前趟坑的吐槽，还有就是去对应的组件的 Apache Issue 或者 GitHub Issue，里面会有专门的 Issue 来跟踪这些 bug 的提出和讨论、修复。

测试阶段

官网已经更新了的版本，则可以自己下载下来，在测试环境测试一下各种功能，比如出 1.10 时，自己就测试了新版本的 SQL 功能，也就是从那个时候才开始正式接触 Flink SQL，之前一直很少去使用 SQL，也觉得时机不对，另外也测试了用 1.9.1 的作业运行在 1.10 上面会是什么反应，作业是否能够正常启动（后面也证明了这个测试是非常有必要的，因为我们业务方就有很多同类型的作业，就是 1.9.1 依赖的作业跑在 1.10 上面就是会出现各种问题，根本启动不了，切回运行在 1.9.1 上面则 OK 的，另外还有使用 1.10 启动抛 ClassNotFound 的异常），另外还测试了 Flink 1.9.1 的作业停止的时候做 savepoint，然后启动的时候使用 1.10 启动，看是否可以正常的去初始化状态，另外还测试了作业的资源使用率，等等。

总之，在升级一个组件之前，做稍微长时间的测试是很有必要的，为了后面在正式生产环境不被坑，总的先去躺一躺，看看坑多不多，深不深！

升级方案



Flink 版本升级后，当用户在实时提交平台重启作业或者提交新作业的时候默认使用 1.10 版本，如果作业有状态的话，则取消作业的时候需要触发 Savepoint 保存作业状态，启动的时候从状态恢复。如果使用 1.10 版本启动异常，可以在实时平台上选择 1.9.1 版本运行或者开发将代码中的 Flink 依赖升级到 1.10，然后再从 Savepoint 中恢复作业。如果线上作业没有重新启动则依旧运行在 1.9.1 版本，

保留 Flink 1.9.1 一个版本的迭代周期，这个版本内开发可以自己将代码中的 Flink 依赖升级到 1.10 版本，到下一个版本我们页面则不再支持 Flink 1.9.1。

准备工作

- Flink 1.10 版本的部署
- Flink 1.10 版本配置调整，比如 Prometheus PushGateway Report 的修改、JVM 内存配置优化、Slot 个数调整
- Streaming/SQL 作业在 Flink 1.9.1/1.10 版本的兼容性测试
- 实时平台提交逻辑的控制
- SQL 的扩展

版本滚动升级阶段

- 升级前，要及时通知用户方，我们升级的方案和后期作业的情况
- 作业启动异常业务方找过来的时候积极配合他们排查，看是否回滚到 1.9.1 还是让业务方升级依赖版本到 1.10.0
- 积极推动新作业使用 1.10.0 版本