

# 如何分析及处理 Flink 反压?

作者介绍:

林小铂，网易游戏高级开发工程师，负责游戏数据中心实时平台的开发及运维工作，目前专注于 Apache Flink 的开发及应用。探究问题本来就是一种乐趣。

本文授权转自: <http://1t.click/bgxw>

反压 (backpressure) 是实时计算应用开发中，特别是流式计算中，十分常见的问题。反压意味着数据管道中某个节点成为瓶颈，处理速率跟不上上游发送数据的速率，而需要对上游进行限速。由于实时计算应用通常使用消息队列来进行生产端和消费端的解耦，消费端数据源是 pull-based 的，所以反压通常是从某个节点传导至数据源并降低数据源（比如 Kafka consumer）的摄入速率。

关于 Flink 的反压机制，网上已经有不少博客介绍，中文博客推荐这两篇[1][2]。简单来说，Flink 拓扑中每个节点 (Task) 间的数据都以阻塞队列的方式传输，下游来不及消费导致队列被占满后，上游的生产也会被阻塞，最终导致数据源的摄入被阻塞。而本文将着重结合官方的博客[4]分享笔者在实践中分析和处理 Flink 反压的经验。

<https://ververica.cn/developers/how-to-analyze-and-deal-with-flink-back-pressure/>