

Apache Flink 零基础入门（七）：状态管理及容错机制

作者：孙梦瑶

整理：韩非

校对：邱从贤（山智）

本文主要分享内容如下：

- 状态管理的基本概念；
- 状态的类型与使用示例；
- 容错机制与故障恢复。

一. 状态管理的基本概念

1. 什么是状态

首先举一个无状态计算的例子：消费延迟计算。假设现在有一个消息队列，消息队列中有一个生产者持续往消费队列写入消息，多个消费者分别从消息队列中读取消息。从图上可以看出，生产者已经写入 16 条消息，Offset 停留在 15；有 3 个消费者，有的消费快，而有的消费慢。消费快的已经消费了 13 条数据，消费者慢的才消费了 7、8 条数据。

如何实时统计每个消费者落后多少条数据，如图给出了输入输出的示例。可以了解到输入的时间点有一个时间戳，生产者将消息写到了某个时间点的位置，每个消费者同一时间点分别读到了什么位置。刚才也提到了生产者写入了 15 条，消费者分别读取了 10、7、12 条。那么问题来了，怎么将生产者、消费者的进度转换为右侧示意图信息呢？

consumer 0 落后了 5 条，consumer 1 落后了 8 条，consumer 2 落后了 3 条，根据 Flink 的原理，此处需进行 Map 操作。Map 首先把消息读取进来，然后分别相减，即可知道每个 consumer 分别落后了几条。Map 一直往下发，则会得出最终结果。

大家会发现，在这种模式的计算中，无论这条输入进来多少次，输出的结果都是一样的，因为单条输入中已经包含了所需的所有信息。消费落后等于生产者减去消费者。生产者的消费在单条数据中可以得到，消费者的数据也可以在单条数据中得到，所以相同输入可以得到相同输出，这就是一个无状态的计算。

<https://ververica.cn/developers/state-management/>