# 概述

MapReduce及其扩展解决了离线批处理问题，但是无法保证实时性。对于实时性要求高的场景，可以采用流式计算或者实时分析系统进行处理。

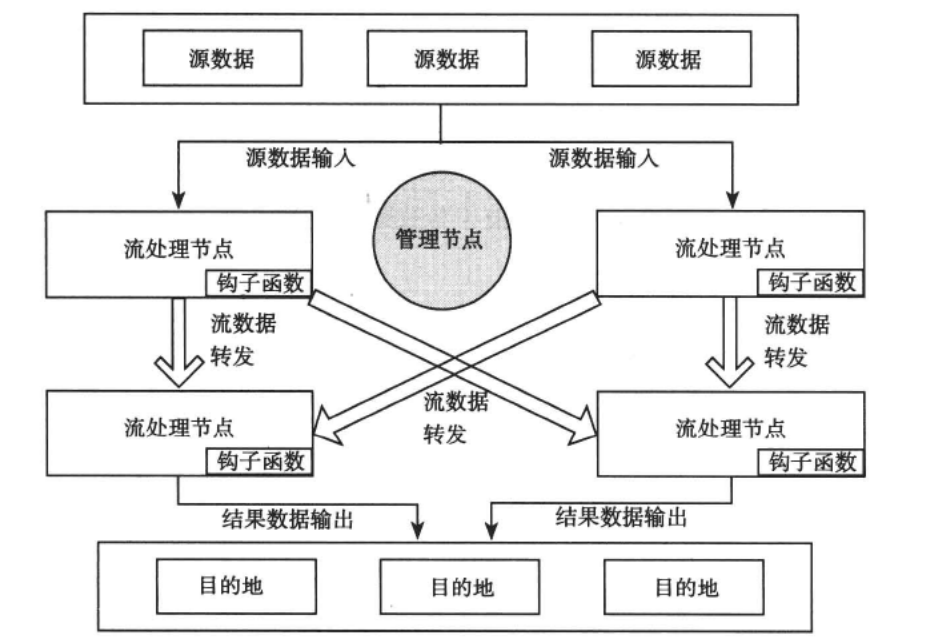
流式计算（Stream Processing）解决在线聚合（Online Aggregation）、在线过滤（Online Filter）等问题，流式计算同时具有存储系统和计算系统的特点，经常应用在一些类似反作弊、交易异常监控等场景。流式计算的操作算子和时间相关，处理最近一段时间窗口内的数据。

# 原理

流式计算强调的是数据流的实时性。MapReduce系统主要解决的是对静态数据的批量处理，当MapReduce作业启动时，已经准备好了输入数据，比如保存在分布式文件系统上。而流式计算系统在启动时，输入数据一般并没有完全到位，而是经由外部数据流源源不断地流入。另外，流式计算并不像批处理系统那样，重视数据处理的总吞吐量，而是更加重视对数据处理的延迟。

MapReduce及其扩展曹勇的是一种比较静态的模型，如果用它来做数据流的处理，首先需要将数据流缓存并分块，然后放入集群计算。如果MapReduce每次处理的数据量较小，缓存数据流的时间较短，但是，MapReduce架构造成的额外开销将会占很大比重；如果MapReduce每次处理的数据量较大，缓存数据流的时间会很长，无法满足实时性的要求。

流式计算系统架构如下：



源数据写入到流处理节点，流处理节点内部运行用户自定义钩子函数对输入流进行处理，处理完后根据一定的规则转发给下游的流处理节点来处理。另外，系统中往往还有管理节点，用来管理流处理节点的状态以及节点之间的路由规则。

典型钩子函数包括：

聚合函数：计算最近一段时间窗口内数据的聚合值，如max、min、avg、sum、count等。

过滤函数：过滤最近一段时间窗口内满足某些特性的数据，如过滤1秒钟内重复的点击数。

# 分类

## Yahoo S4

## Twitter Strom