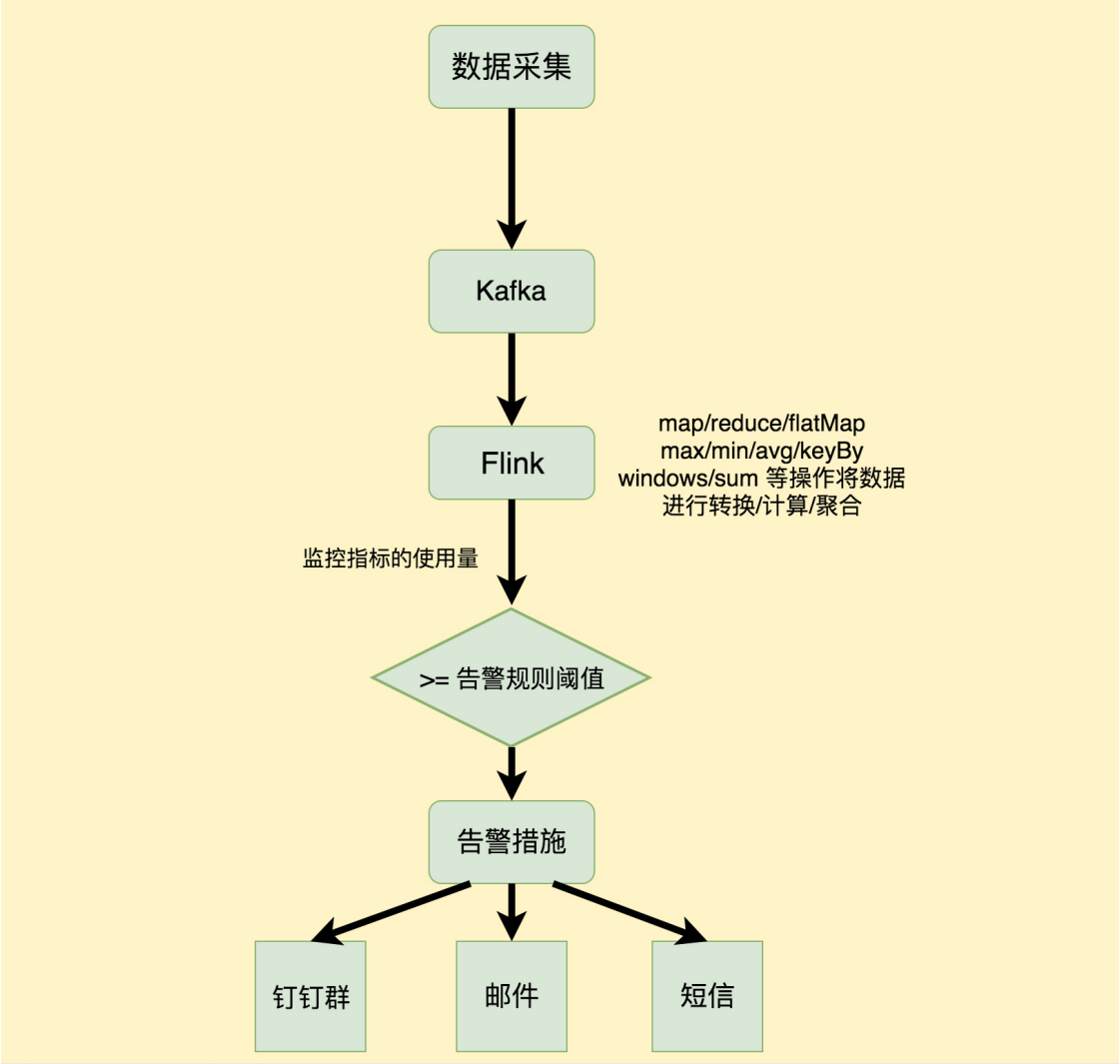
---

title: 《从0到1学习Flink》—— Apache Flink 介绍 toc: true date: 2018-10-13 tags:

- Flink
- 大数据
- 流式计算

前言

Flink 是一种流式计算框架，为什么我会接触到 Flink 呢？因为我目前在负责的是监控平台的告警部分，负责采集到的监控数据会直接往 kafka 里塞，然后告警这边需要从 kafka topic 里面实时读取到监控数据，并将读取到的监控数据做一些 聚合/转换/计算 等操作，然后将计算后的结果与告警规则的阈值进行比较，然后做出相应的告警措施（钉钉群、邮件、短信、电话等）。画了个简单的图如下：



目前告警这块的架构是这样的结构，刚进公司那会的时候，架构是所有的监控数据直接存在 ElasticSearch 中，然后我们告警是去 ElasticSearch 中搜索我们监控指标需要的数据，幸好 ElasticSearch 的搜索能力够强大。但是你有没有发现一个问题，就是所有的监控数据从采集、采集后的数据做一些 计算/转换/聚合、再通过 Kafka 消息队列、再存进 ElasticSearch 中，再而去 ElasticSearch 中查找我们的监控数据，然后做出告警策略。整个流程对监控来说看起来很按照常理，但是对于告警来说，如果中间某个环节出了问题，比如 Kafka 消息队列延迟、监控数据存到 ElasticSearch 中写入时间较长、你的查询姿势写的不对等原因，这都将导致告警从 ElasticSearch 查到的数据是有延迟的。也许是 30 秒、一分钟、或者更长，这样对于告警来说这无疑将导致告警的消息没有任何的意义。

为什么这么说呢？为什么需要监控告警平台呢？无非就是希望我们能够尽早的发现问题，把问题给告警出来，这样开发和运维人员才能够及时的处理解决好线上的问题，以免给公司造成巨大的损失。

更何况现在还有更多的公司在做那种提前预警呢！这种又该如何做呢？需要用大数据和机器学习的技术去分析周期性的历史数据，然后根据这些数据可以整理出来某些监控指标的一些周期性（一天/七天/一月/一季度/一年）走势图，这样就大概可以绘图出来。然后根据这个走势图，可以将当前时间点的监控指标的数据使用量和走势图进行对比，在快要达到我们告警规则的阈值时，这时就可以提前告一个预警出来，让运维提前知道预警，然后提前查找问题，这样就能够提早发现问题所在，避免损失，将损失降到最小！当然，这种也是我打算做的，应该可以学到不少东西的。

于是乎，我现在就在接触流式计算框架 Flink，类似的还有常用的 Spark 等。

自己也接触了 Flink 一段时间了，这块中文资料目前书籍是只有一本很薄的，英文书籍也是三本不超过。

我自己整理了些 Flink 的学习资料，目前已经全部放到微信公众号了。你可以关注我的公众号：**zhisheng**，然后回复关键字：**Flink** 即可无条件获取到。

另外这里也推荐一些博客可以看看：

1、官网：https://flink.apache.org/

2、GitHub: https://github.com/apache/flink

3、https://blog.csdn.net/column/details/apacheflink.html

4、https://blog.csdn.net/lmalds/article/category/6263085

5、http://wuchong.me/

6、https://blog.csdn.net/liguohuabigdata/article/category/7279020

下面的介绍可能也有不少参考以上所有的资料，感谢他们！在介绍 Flink 前，我们先看看 **数据集类型** 和 **数据运算模型** 的种类。

数据集类型有哪些呢：

- 无穷数据集：无穷的持续集成的数据集合
- 有界数据集：有限不会改变的数据集合

那么那些常见的无穷数据集有哪些呢?

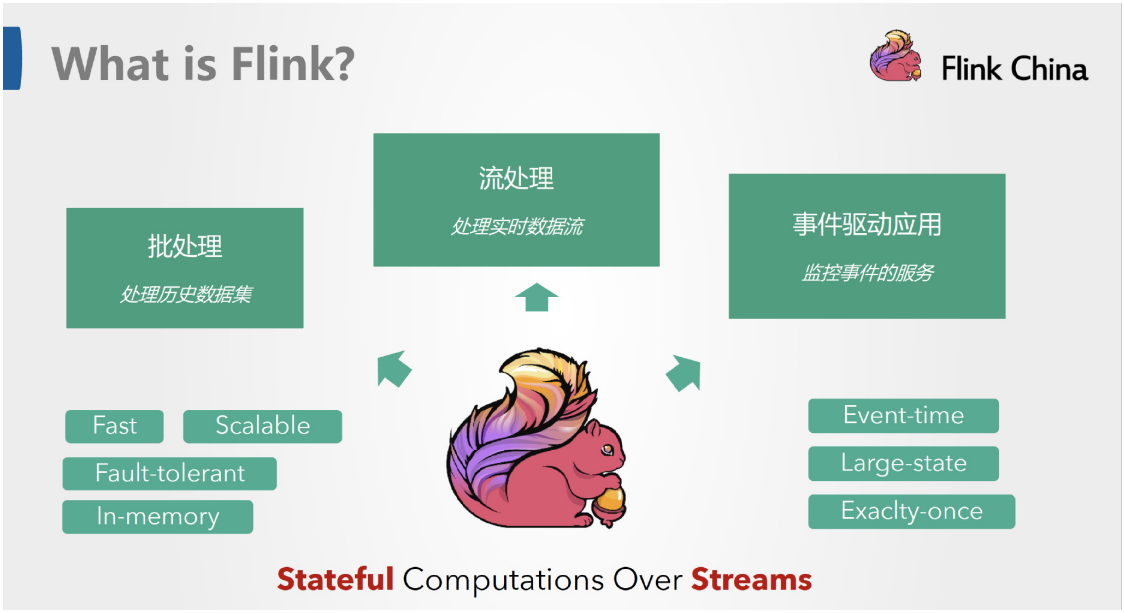
- 用户与客户端的实时交互数据
- 应用实时产生的日志
- 金融市场的实时交易记录
- ...

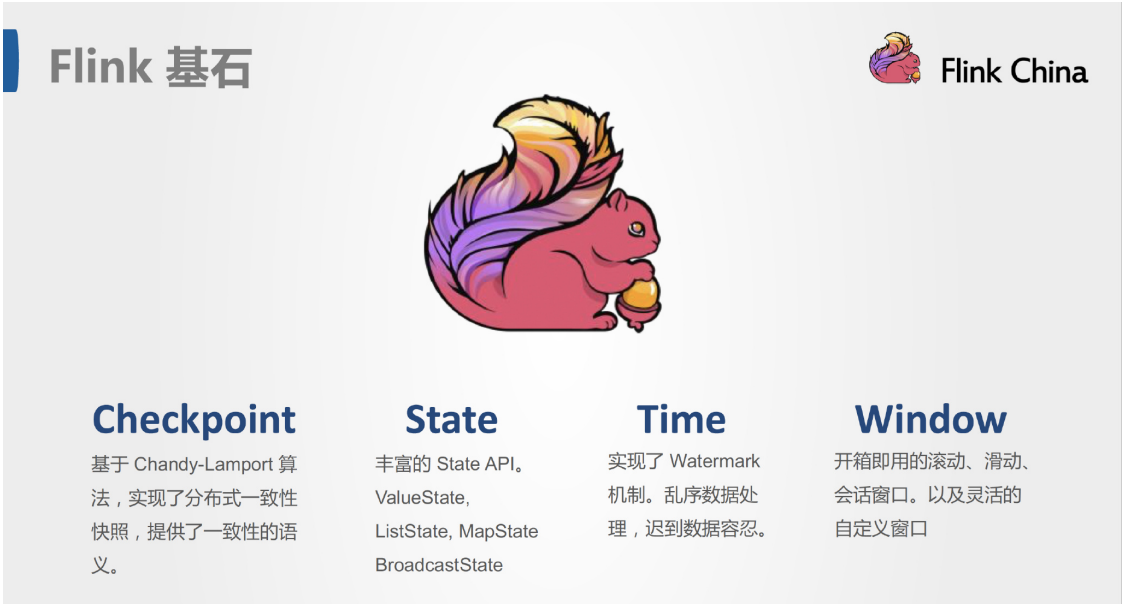
数据运算模型有哪些呢：

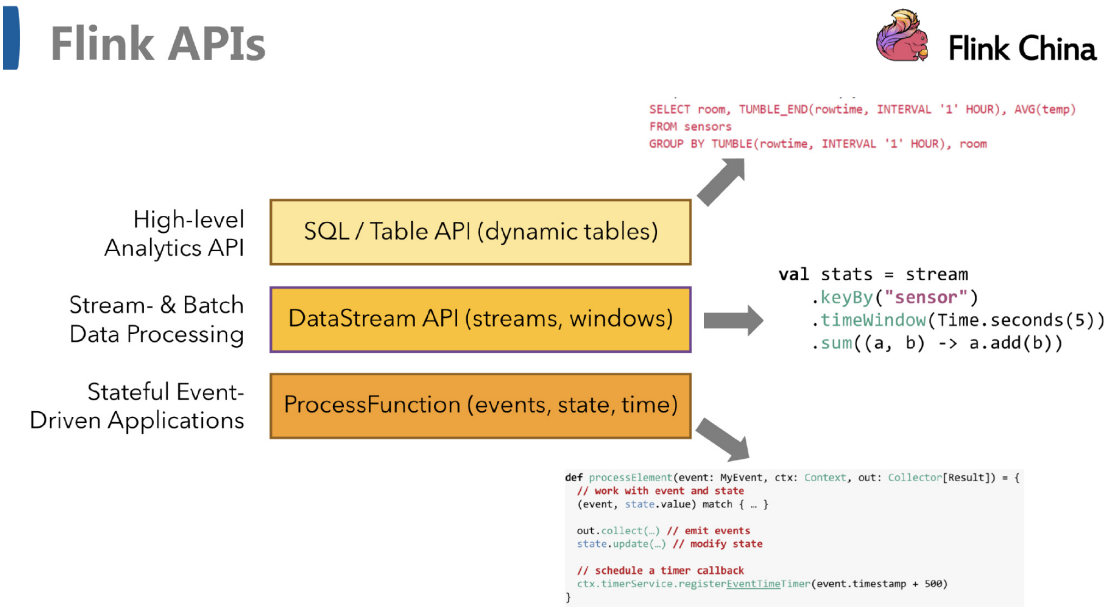
- 流式：只要数据一直在产生，计算就持续地进行
- 批处理：在预先定义的时间内运行计算，当完成时释放计算机资源

Flink 它可以处理有界的数据集、也可以处理无界的数据集、它可以流式的处理数据、也可以批量的处理数据。

Flink 是什么？

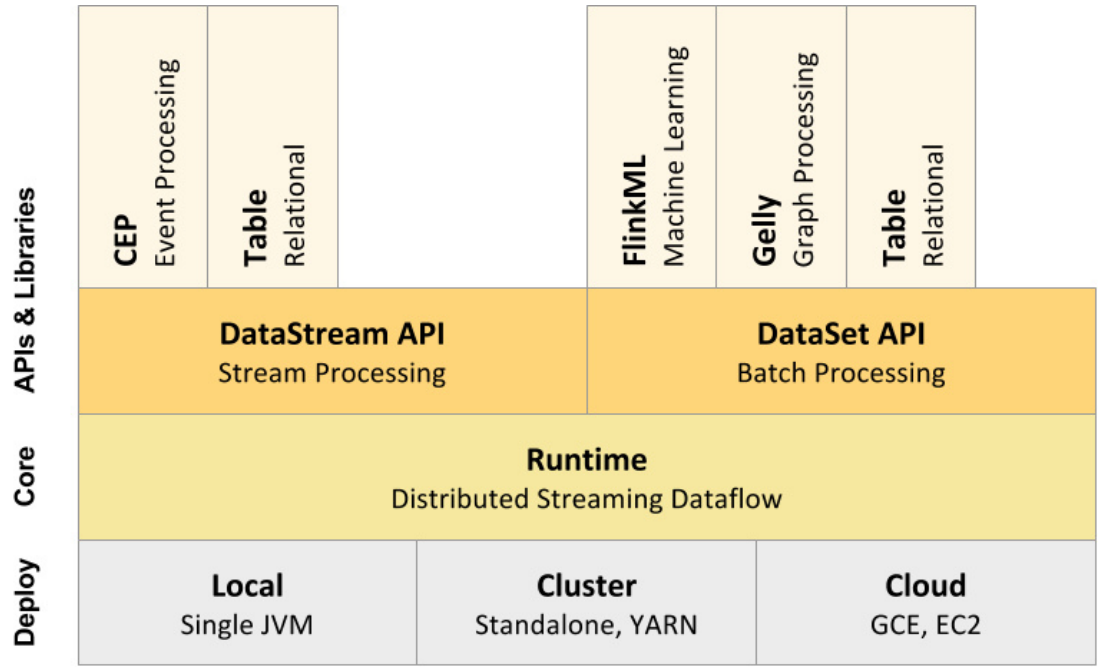






上面三张图转自 云邪 成都站 《Flink 技术介绍与未来展望》，侵删。

从下至上，Flink 整体结构



从下至上：

1、部署：Flink 支持本地运行、能在独立集群或者在被 YARN 或 Mesos 管理的集群上运行， 也能部署在云上。

2、运行：Flink 的核心是分布式流式数据引擎，意味着数据以一次一个事件的形式被处理。

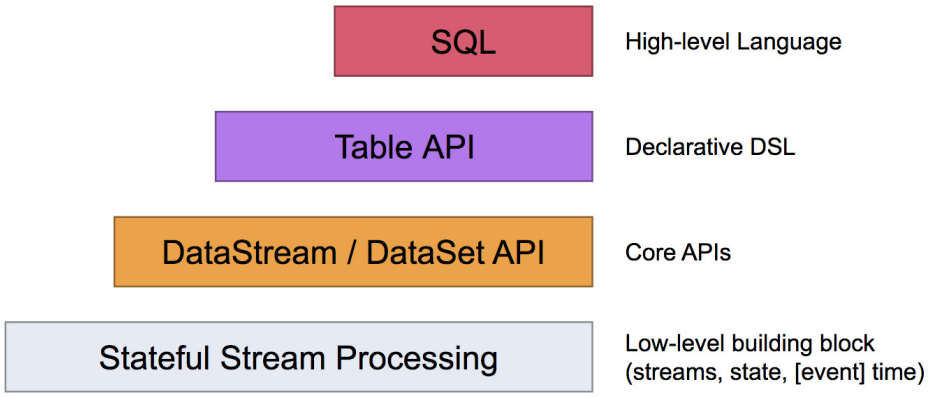
3、API：DataStream、DataSet、Table、SQL API。

4、扩展库：Flink 还包括用于复杂事件处理，机器学习，图形处理和 Apache Storm 兼容性的专用代码库。

Flink 数据流编程模型

抽象级别

Flink 提供了不同的抽象级别以开发流式或批处理应用。

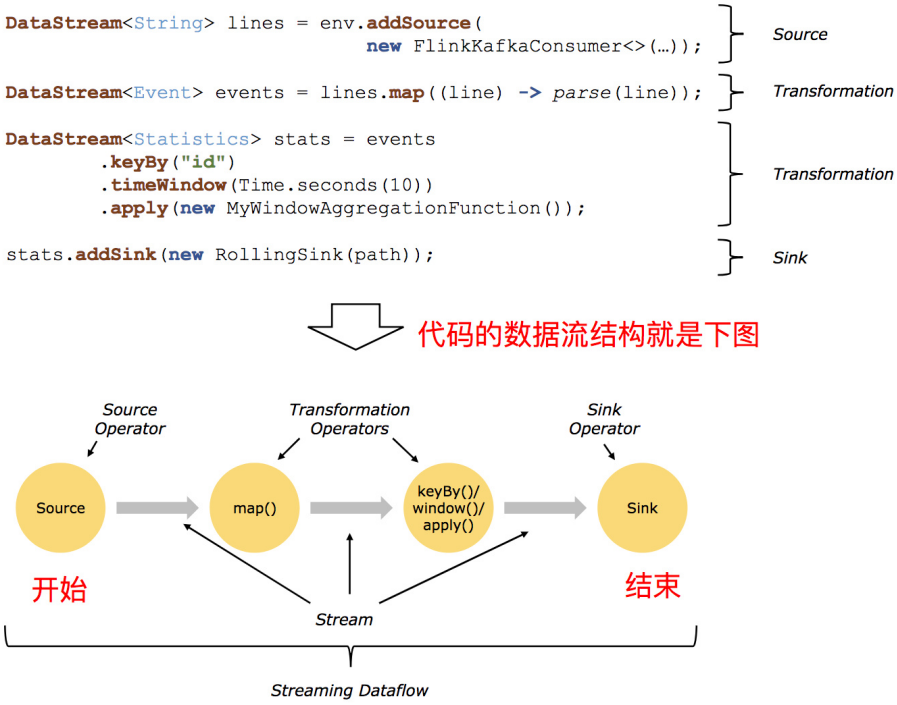


- 最底层提供了有状态流。它将通过 过程函数（Process Function）嵌入到 DataStream API 中。它允许用户可以自由地处理来自一个或多个流数据的事件，并使用一致、容错的状态。除此之外，用户可以注册事件时间和处理事件回调，从而使程序可以实现复杂的计算。
- DataStream / DataSet API 是 Flink 提供的核心 API ，DataSet 处理有界的数据集，DataStream 处理有界或者无界的数据流。用户可以通过各种方法（map / flatmap / window / keyby / sum / max / min / avg / join 等）将数据进行转换 / 计算。
- **Table API** 是以 *表* 为中心的声明式 DSL，其中表可能会动态变化（在表达流数据时）。Table API 提供了例如 select、project、join、group-by、aggregate 等操作，使用起来却更加简洁（代码量更少）。

你可以在表与 *DataStream*/*DataSet* 之间无缝切换，也允许程序将 *Table API* 与 *DataStream* 以及 *DataSet* 混合使用。

- Flink 提供的最高层级的抽象是 **SQL** 。这一层抽象在语法与表达能力上与 *Table API* 类似，但是是以 SQL查询表达式的形式表现程序。SQL 抽象与 Table API 交互密切，同时 SQL 查询可以直接在 Table API 定义的表上执行。

Flink 程序与数据流结构



Flink 应用程序结构就是如上图所示：

1、Source: 数据源，Flink 在流处理和批处理上的 source 大概有 4 类：基于本地集合的 source、基于文件的 source、基于网络套接字的 source、自定义的 source。自定义的 source 常见的有 Apache kafka、Amazon Kinesis Streams、RabbitMQ、Twitter Streaming API、Apache NiFi 等，当然你也可以定义自己的 source。

2、Transformation：数据转换的各种操作，有 Map / FlatMap / Filter / KeyBy / Reduce / Fold / Aggregations / Window / WindowAll / Union / Window join / Split / Select / Project 等，操作很多，可以将数据转换计算成你想要的数据。

3、Sink：接收器，Flink 将转换计算后的数据发送的地点 ，你可能需要存储下来，Flink 常见的 Sink 大概有如下几类：写入文件、打印出来、写入 socket 、自定义的 sink 。自定义的 sink 常见的有 Apache kafka、RabbitMQ、MySQL、ElasticSearch、Apache Cassandra、Hadoop FileSystem 等，同理你也可以定义自己的 sink。

为什么选择 Flink?

Flink 是一个开源的分布式流式处理框架：

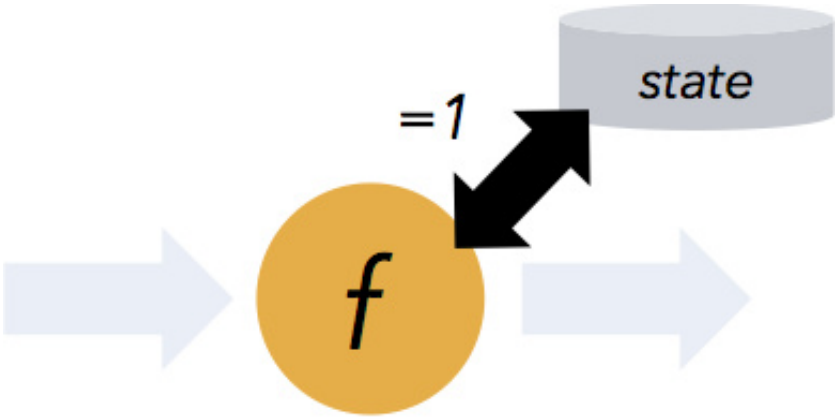
①提供准确的结果，甚至在出现无序或者延迟加载的数据的情况下。

②它是状态化的容错的，同时在维护一次完整的的应用状态时，能无缝修复错误。

③大规模运行，在上千个节点运行时有很好的吞吐量和低延迟。

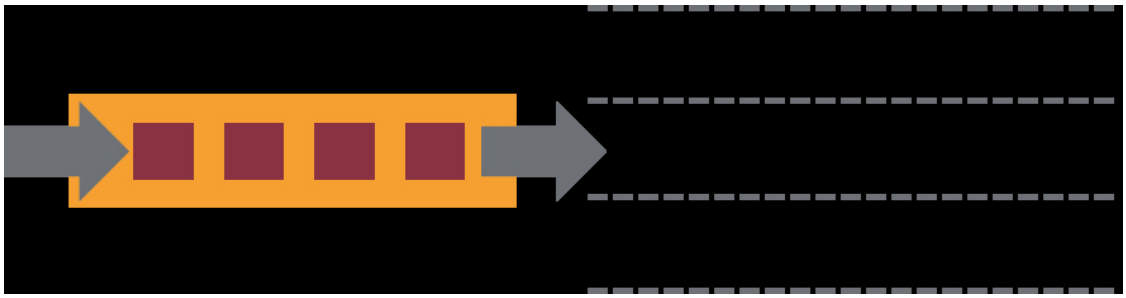
更早的时候，我们讨论了数据集类型（有界 vs 无穷）和运算模型（批处理 vs 流式）的匹配。Flink 的流式计算模型启用了很多功能特性，如状态管理，处理无序数据，灵活的视窗，这些功能对于得出无穷数据集的精确结果是很重要的。

- Flink 保证状态化计算强一致性。”状态化“意味着应用可以维护随着时间推移已经产生的数据聚合或者，并且 Filnk 的检查点机制在一次失败的事件中一个应用状态的强一致性。

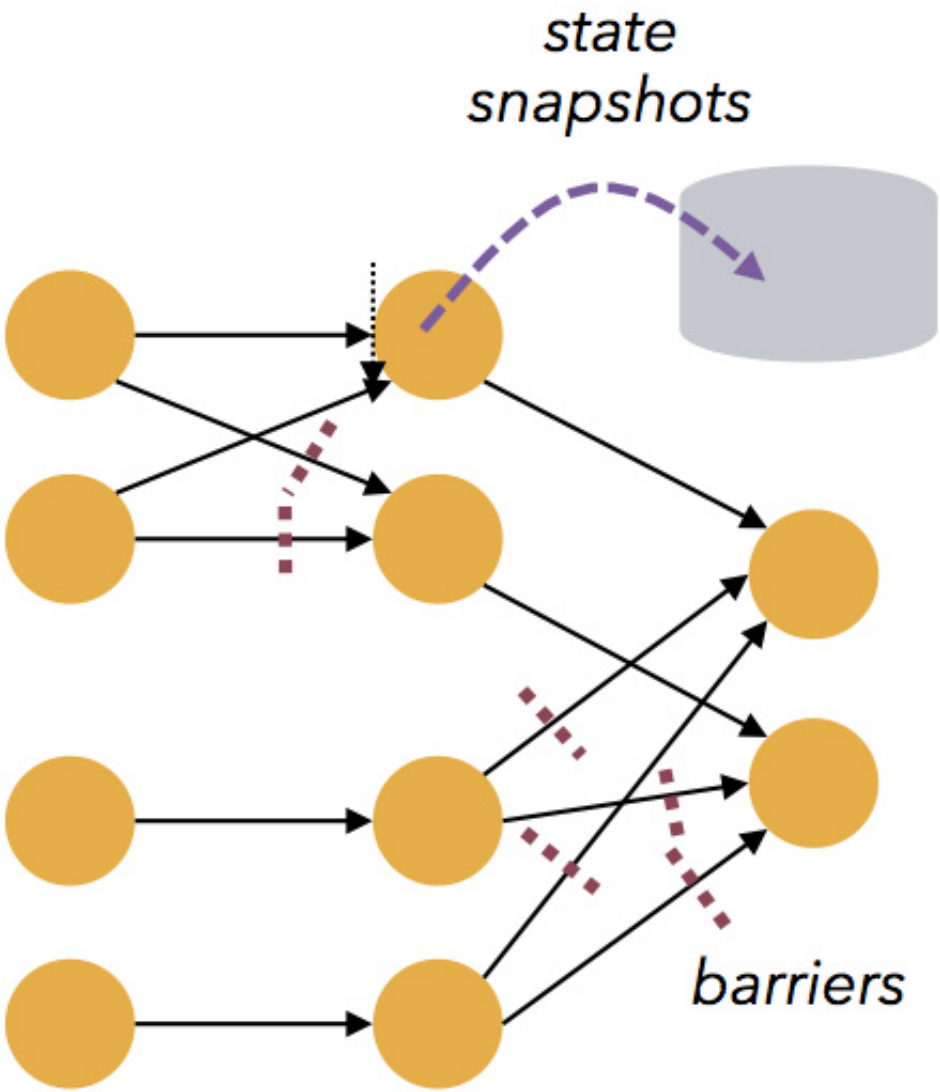


- Flink 支持流式计算和带有事件时间语义的视窗。事件时间机制使得那些事件无序到达甚至延迟到达的数据流能够计算出精确的结果。

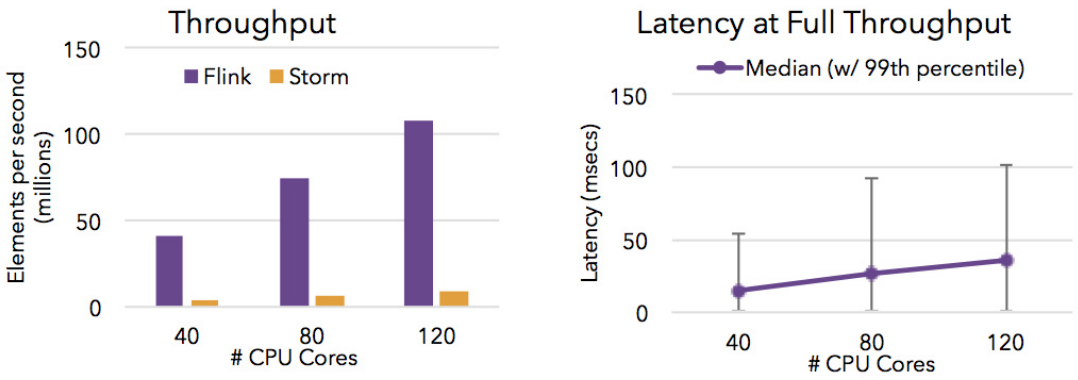
- 除了提供数据驱动的视窗外，Flink 还支持基于时间，计数，session 等的灵活视窗。视窗能够用灵活的触发条件定制化从而达到对复杂的流传输模式的支持。Flink 的视窗使得模拟真实的创建数据的环境成为可能。



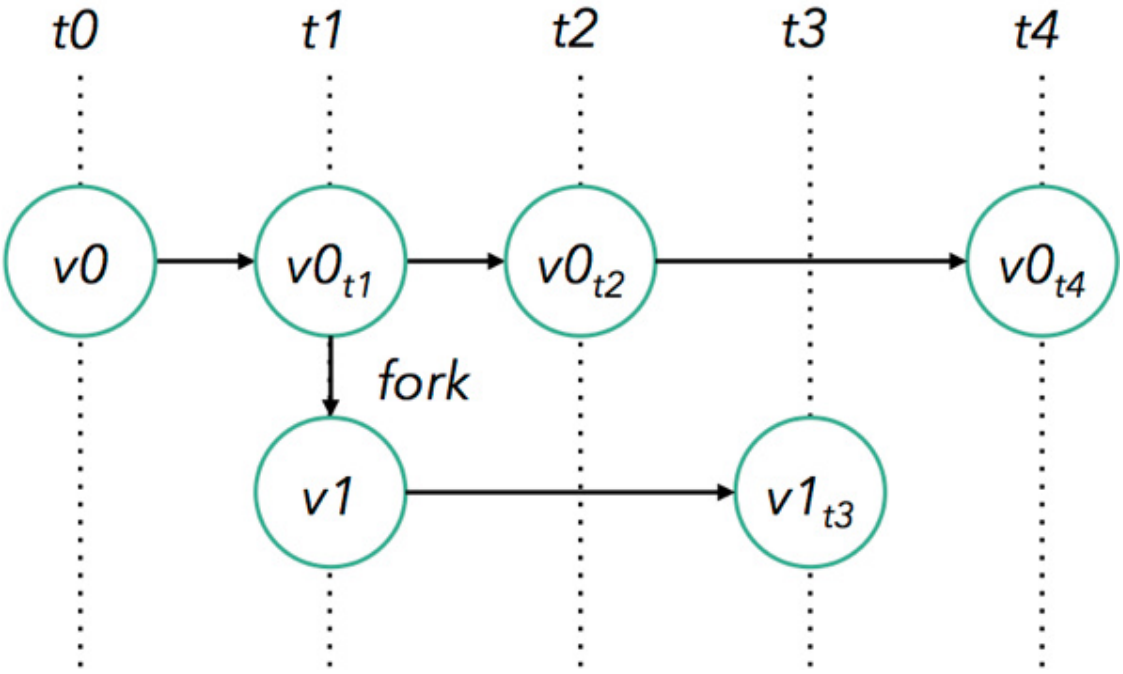
- Flink 的容错能力是轻量级的，允许系统保持高并发，同时在相同时间内提供强一致性保证。Flink 以零数据丢失的方式从故障中恢复，但没有考虑可靠性和延迟之间的折衷。



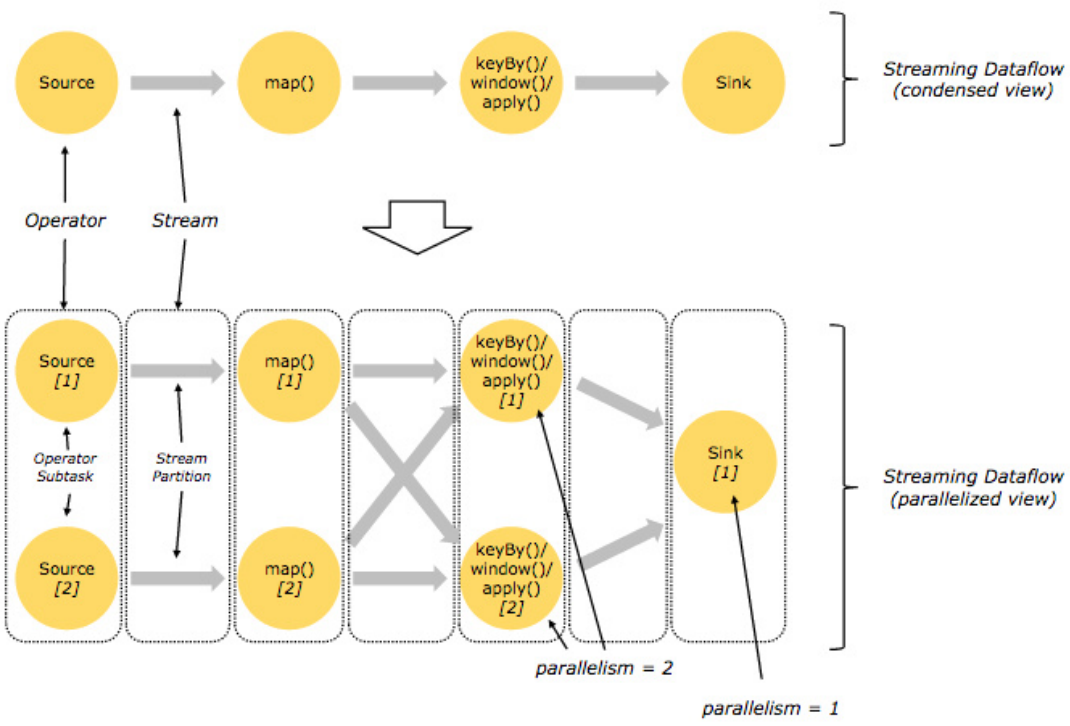
- Flink 能满足高并发和低延迟（计算大量数据很快）。下图显示了 Apache Flink 与 Apache Storm 在完成流数据清洗的分布式任务的性能对比。



- Flink 保存点提供了一个状态化的版本机制，使得能以无丢失状态和最短停机时间的方式更新应用或者回退历史数据。



- Flink 被设计成能用上千个点在大规模集群上运行。除了支持独立集群部署外，Flink 还支持 YARN 和Mesos 方式部署。
- Flink 的程序内在是并行和分布式的，数据流可以被分区成 **stream partitions**，operators 被划分为operator subtasks; 这些 subtasks 在不同的机器或容器中分不同的线程独立运行；operator subtasks 的数量在具体的 operator 就是并行计算数，程序不同的 operator 阶段可能有不同的并行数；如下图所示，source operator 的并行数为 2，但最后的 sink operator 为1；



- 自己的内存管理

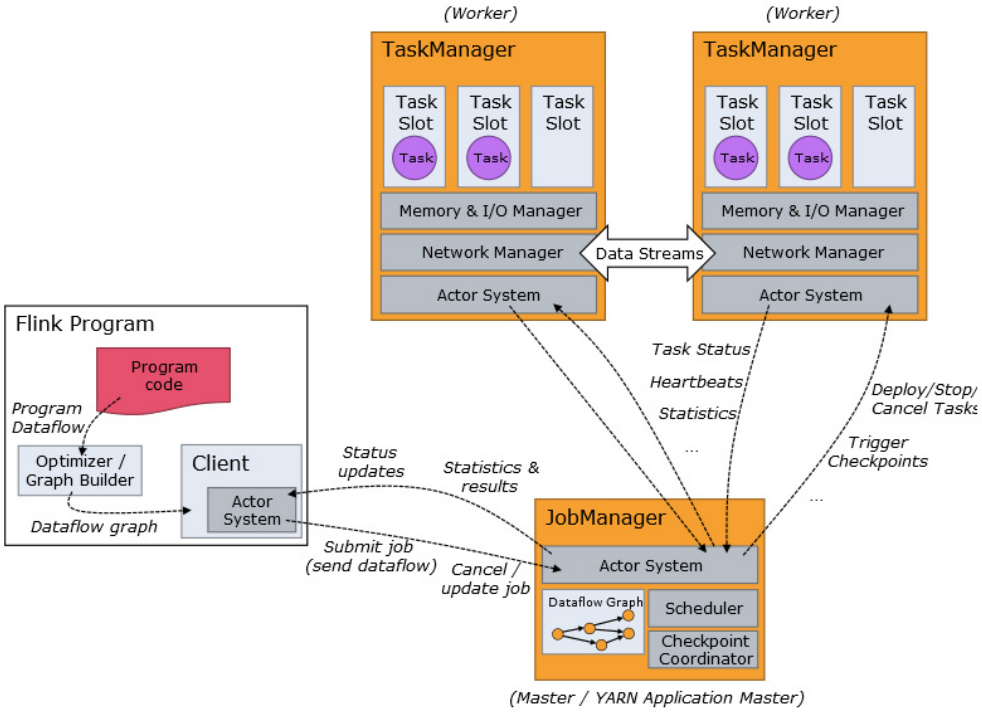
  Flink 在 JVM 中提供了自己的内存管理，使其独立于 Java 的默认垃圾收集器。它通过使用散列，索引，缓存和排序有效地进行内存管理。

- 丰富的库

  Flink 拥有丰富的库来进行机器学习，图形处理，关系数据处理等。 由于其架构，很容易执行复杂的事件处理和警报。

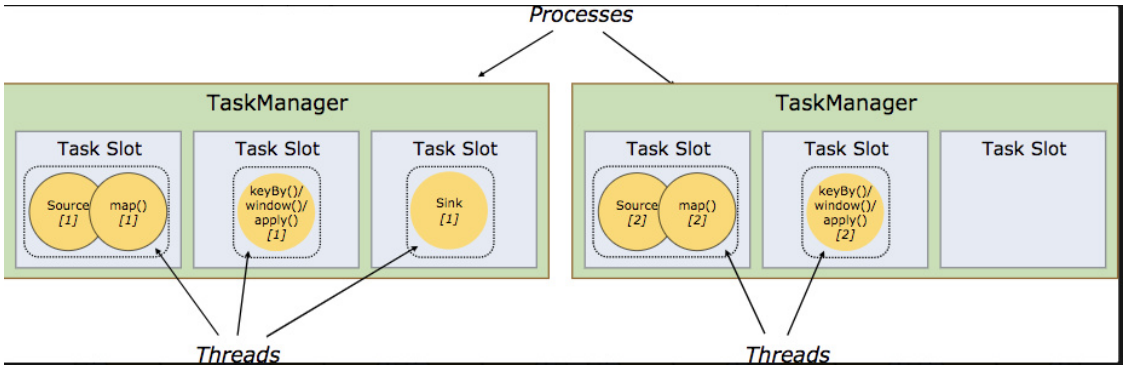
分布式运行

flink 作业提交架构流程可见下图：



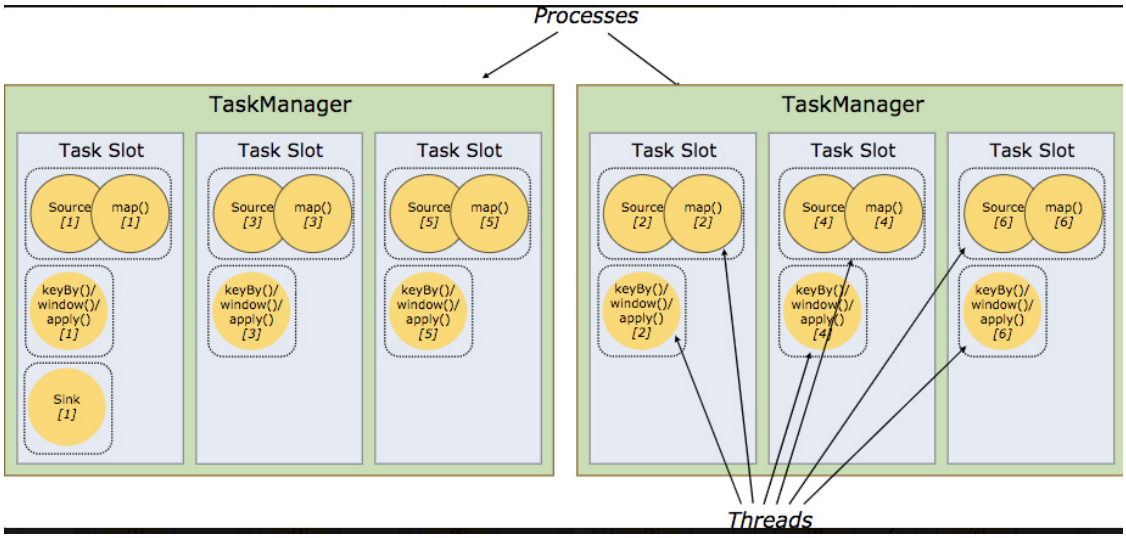
1、Program Code：我们编写的 Flink 应用程序代码

2、Job Client：Job Client 不是 Flink 程序执行的内部部分，但它是任务执行的起点。 Job Client 负责接受用户的程序代码，然后创建数据流，将数据流提交给 Job Manager 以便进一步执行。 执行完成后，Job Client 将结果返回给用户



3、Job Manager：主进程（也称为作业管理器）协调和管理程序的执行。 它的主要职责包括安排任务，管理checkpoint ， 故障恢复等。机器集群中至少要有一个 master，master 负责调度 task，协调 checkpoints 和容灾，高可用设置的话可以有多个 master，但要保证一个是 leader, 其他是 standby; Job Manager 包含 Actor system、Scheduler、Check pointing 三个重要的组件

4、Task Manager：从 Job Manager 处接收需要部署的 Task。Task Manager 是在 JVM 中的一个或多个线程中执行任务的工作节点。 任务执行的并行性由每个 Task Manager 上可用的任务槽决定。 每个任务代表分配给任务槽的一组资源。 例如，如果 Task Manager 有四个插槽，那么它将为每个插槽分配 25% 的内存。 可以在任务槽中运行一个或多个线程。 同一插槽中的线程共享相同的 JVM。 同一 JVM 中的任务共享 TCP 连接和心跳消息。Task Manager 的一个 Slot 代表一个可用线程，该线程具有固定的内存，注意 Slot 只对内存隔离，没有对 CPU 隔离。默认情况下，Flink 允许子任务共享 Slot，即使它们是不同 task 的 subtask，只要它们来自相同的 job。这种共享可以有更好的资源利用率。



最后

本文主要讲了我接触到 Flink 的缘由，然后从数据集类型和数据运算模型开始讲起，接着介绍了下 Flink 是什么、Flink 的整体架构、提供的 API、Flink 的优点所在以及 Flink 的分布式作业运行的方式。水文一篇，希望你能够对 Flink 稍微有一点概念了。

关注我

转载请务必注明原创地址为：http://www.54tianzhisheng.cn/2018/10/13/flink-introduction/

另外我自己整理了些 Flink 的学习资料，目前已经全部放到微信公众号了。你可以加我的微信：zhisheng_tian，然后回复关键字：Flink 即可无条件获取到。

Github 代码仓库

https://github.com/zhisheng17/flink-learning/

以后这个项目的所有代码都将放在这个仓库里，包含了自己学习 flink 的一些 demo 和博客

相关文章

1、《从0到1学习Flink》—— Apache Flink 介绍

2、《从0到1学习Flink》—— Mac 上搭建 Flink 1.6.0 环境并构建运行简单程序入门

3、《从0到1学习Flink》—— Flink 配置文件详解

4、《从0到1学习Flink》—— Data Source 介绍

5、《从0到1学习Flink》—— 如何自定义 Data Source ？

6、《从0到1学习Flink》—— Data Sink 介绍

7、《从0到1学习Flink》—— 如何自定义 Data Sink？

8、《从0到1学习Flink》—— Flink Data transformation(转换)

9、《从0到1学习Flink》—— 介绍Flink中的Stream Windows

10、《从0到1学习Flink》—— Flink 中的几种 Time 详解

11、《从0到1学习Flink》—— Flink 写入数据到 ElasticSearch