Chapter 1：Introduction to Stateful Stream Processing

(第一章：有状态流处理简介)

前言

1.2014.04 成为 Apache 孵化项目，2015.01 成为顶级项目

2.引出有状态的流处理概念，并介绍受人喜爱的原因

3.介绍开源流处理的发展及 flink 样例程序

1. Traditional Data Infrastructures

（传统数据架构）

1.1. Transactional Processing

(事务处理)

1.简单系统的增删改查

2.当出现多个 app 访问相同数据源时，程序的迭代和扩展都会遇到问题，解决上述场景的架构是使用微服务处

理方式，遵循 unix 设计原则

(解决多程序访问相同数据源，就是专事专做，订单系统、用户系统等分开)

1.1.1.

事务型应用的设计

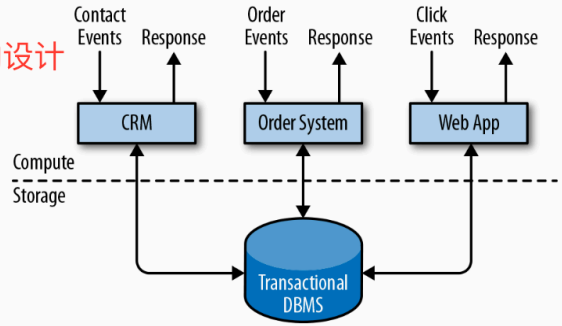


Figure 1-1. Traditional design of transactional applications that store data in a remote database system

1.1.2.

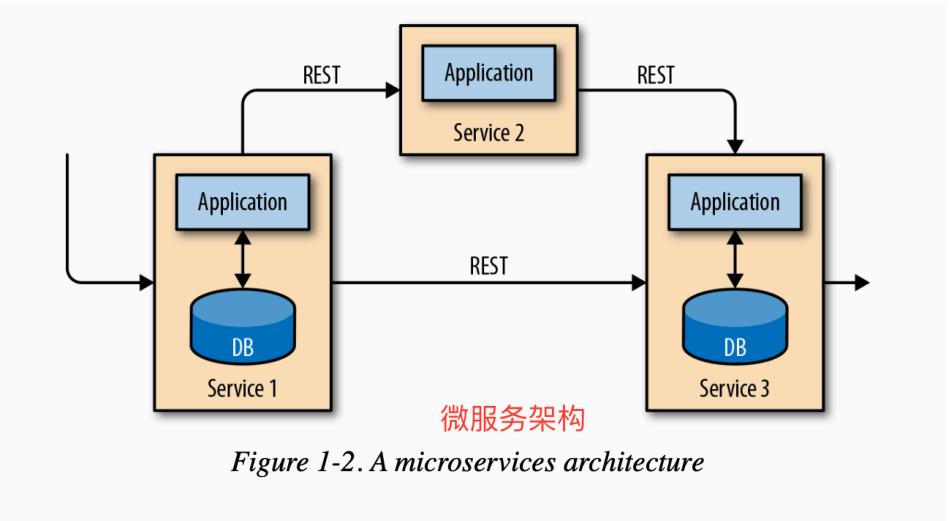


Figure 1-2. A microservices architecture

1.2. Analytical Processing

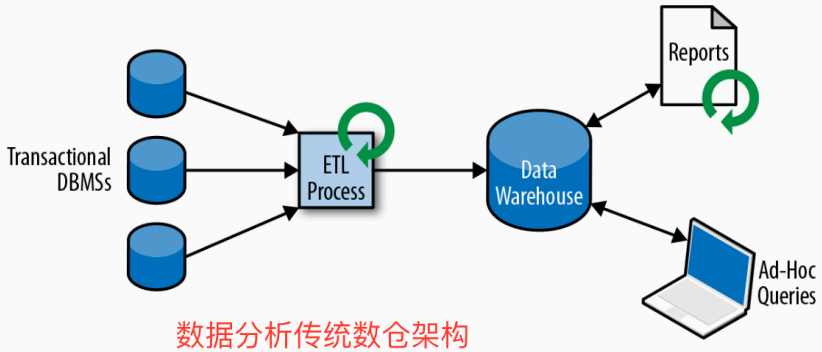
（分析处理）

事务型的数据库一般存储的数据比较单一类型，当进行多种数据的联合分析时，分析查询一般不会运行在事务数据库上，需要 etl 从事务数据库中同步到数据仓库中，数仓的查询分为两种

1.历史定期 etl 的 job 类型

2.即席查询

1.2.1.



数据分析传统数仓架构

Figure 1-3. A traditional data warehouse architecture for data analytics

2. Stateful Stream Processing

（有状态的流处理）

前言

1.flink 通过内存或者远程数据库存储状态

2.可以通过流量重放机制来恢复任务

2.1. Event-Driven Applications

（事件驱动型的程序）

1.例如 实时推荐、模式检测、异常检测

2.事件驱动型是微服务的演化，不同在于将状态存储在本地而不是远程去数据库访问

3.与 oltp 或者微服务相比的优点

*本地状态存储的性能更优秀

*扩展性和容错性交给流处理器自动处理

*exactly-once state consistency 精确状态一致性和应用程序的可扩展性是事件驱动程序的基本要求

2.1.1.

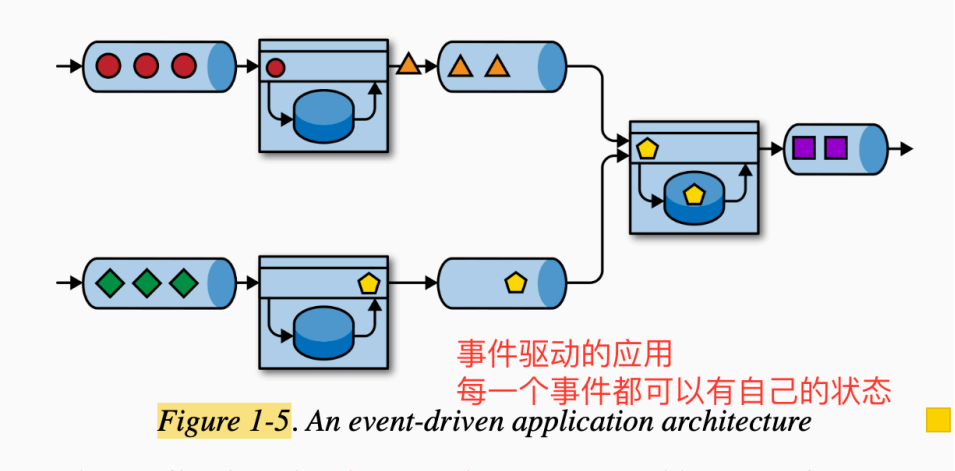


Figure 1-5. An event-driven application architecture

2.2. Data Pipelines

（数据管道）

1.解决各类存储之间的数据一致性问题

例如：将更新日志分发到不同系统，使用 flink 更新受影响的数据

2.3. Streaming Analytics

（基于流的分析）

1.本书中没有介绍基于流的 sql

2.主要是低延迟作用，将结果更新到支持 update 的数据库中，以便直接访问

3. The Evolution of Open Source Stream Processing

（开源流处理的发展）

前言

开源流处理器最早可追溯到 20 世纪 90 年代，开源软件起到了很重要的作用

（人人可用，社区贡献）

3.1. A Bit of History

1.第一代的流处理器（2011）更关注与毫秒及延迟，但不支持准确的一致性结果，因为他是基于事件的到达顺序来处理的，存在多次消费

2.基于第一代演化出 lamda 架构，最初的目标是解决批处理架构带来的高延迟，但 lamda 结构依然存在以下缺点

*一套逻辑写两遍

*流处理通道计算是近似值

*启动和维护困难

3.第二代流处理引擎（2013） 以牺牲毫秒延迟为代价，提高了吞吐量和故障保障，但结果依然是取决与事件的到达时间

4.第三代流处理器（2015）

*解决了对到达时间的依赖

*支持精确一次语义

*消除低延迟/高吞吐吞吐的权衡，可以服务与频谱的两端

*高可用、可扩展、与资源管理的集成、作业迁移

4. A Quick Look at Flink

（快速浏览 flink）

前言

***两种时间语义支持**

***精确一次性状态保证**

***毫秒级延迟、可扩展性**

***多层 api 支持**

***多数据连接支持**

***7*24 小时连续运行、易与资源框架集成**

***不丢失状态迁移和更新 job**

***批流融合**

***嵌入式执行可在单个 jvm 程序中启动 flink 程序进行开发和测试**

4.1. Running Your First Flink Application

（flink demo 程序）

1. 下载 flink 二进制发行版地址：https://archive.apache.org/dist/flink/flink-1.7.1/

2. 解压命令 $ tar xvfz flink-1.7.1-bin-scala_2.12.tgz

3. 启动本地 flink cluster

$ cd flink-1.7.1

$ ./bin/start-cluster.sh

4. web ui :http://localhost:8081

5.下载 example 代码

https://streaming-with-flink.github.io/examples/download/examples-scala.jar

6.提交 example 的 jar 包

./bin/flink run \

-c io.github.streamingwithflink.chapter1.AverageSensorReadings \

/下载路径/examples-scala.jar

7. 程序日志是 worker 进程输出的 查看命令

$ tail -f ./log/flink-<user>-taskexecutor-<n>-<hostname>.out

8. 关闭集群

$ ./bin/stop-cluster.sh