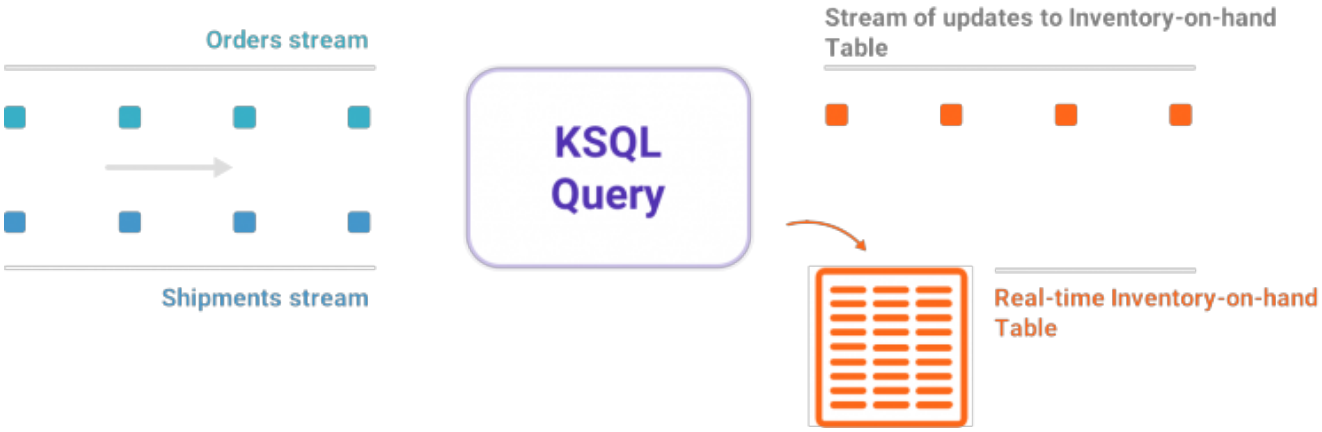
KSQL介绍：面向Apache Kafka的开源Streaming SQL引擎

我非常高兴地宣布KSQL，这是面向Apache Kafka的一种数据流SQL引擎。KSQL降低了数据流处理这个领域的准入门槛，为使用Kafka处理数据提供了一种简单的、完全交互的SQL界面。你不再需要用Java或Python之类的编程语言编写代码了！KSQL具有这些特点：开源（采用Apache 2.0许可证）、分布式、可扩展、可靠、实时。它支持众多功能强大的数据流处理操作，包括聚合、连接、加窗（windowing）和sessionization（捕获单一访问者的网站会话时间范围内所有的点击流事件）等等。

一个简单的例子



如果想及时了
解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

查询流式数据意味着什么？这与SQL数据库相比怎样？

可以说它实际上与SQL数据库大不一样。大多数数据库用于对存储的数据执行按需查找和改动。KSQL还无法执行查询，它所做的是连续转换——也就是说，数据流处理。比如设想一下：我有来自用户的点击流和帐户信息表（这些信息关于不断更新的那些用户）。KSQL让我可以对该点击流和用户表进行建模，并将两者连接起来，尽管那两者当中的一个是无限的。

所以，KSQL对Kafka话题中的数据流执行连续查询——随着新数据不断流入，转换在连续进行。相比之下，针对关系数据库的查询是一次性查询——对数据集运行一次即可完成，就像针对数据库中有限行的SELECT语句。

KSQL适用于什么？

很好，现在你可以连续查询无限数据流。那有什么好处呢？

1、实时监控遇上实时分析

```
CREATE TABLE error_counts AS
SELECT error_code, count(*)FROM monitoring_stream
WINDOW TUMBLING (SIZE 1 MINUTE)
WHERE type = 'ERROR'
```

这方面的一个用途是，定义实时计算的自定义业务级别度量指标，你可以监控并发出警报，就像监控CPU负载那样。另一个用途是，在KSQL中为应用程序定义正确性概念，并核实它在生产环境中运行时满足这个概念。我们一提到监控，常常想到跟踪低级别性能统计数字的计数器（counter）和计量器（gauge）。这些种类的计量器常常可以告诉你CPU负载很高，但其实无法告诉你应用程序是否在做它应该做的事情。KSQL允许针对应用程序生成的原始事件流定义自定义度量指标，无论它们是日志事件、数据库更新还是其他任何类型的事件。

比如说，一个Web应用程序可能需要核实：每当新客户注册，就发送欢迎电子邮件，创建新的用户记录，并对其信用卡计费。这些功能可能分散在不同的服务或应用程序中，你需要监控，确保对每个新客户而言，每个操作都在某个服务级别协议（SLA）里面（比如30秒）。

2、安全和异常检测

```
CREATE STREAM possible_fraud AS
SELECT card_number, count(*)
FROM authorization_attempts
WINDOW TUMBLING (SIZE 5 SECONDS)
GROUP BY card_number
HAVING count(*) > 3;
```

这方面的简单版本是你在上面演示中看到的：KSQL查询将事件流转换成数值时间序列聚合，这些聚合使用Kafka-Elastic连接件输入到Elastic，并在Grafana UI中加以可视化。安全用例常常酷似监控和分析。你寻找欺诈、滥用、垃圾邮件、入侵或其他不良行为方面的模式，而不是监控应用程序行为或业务行为。KSQL为定义这些模式并查询实时数据流提供了一种简单、先进、实时的方法。

3、联机数据整合

```
CREATE STREAM vip_users AS
SELECT userid, page, action
FROM clickstream c
LEFT JOIN users u ON c.userid = u.user_id
WHERE u.level = 'Platinum';
```

许多公司进行的数据处理大部分属于数据丰富（data enrichment）这个范畴：拿来来自几个数据库的数据，将其转换，连接起来，并将数据存储到键值存储库、搜索索引、缓存或其他数据服务系统。长期以来，用于数据整合的ETL（提取、转换和加载）作为定期的批处理作业来加以执行。比如说，实时转储原始数据，然后每隔几小时进行转换，以实现高效查询。对于许多用例而言，这种延迟是不可接受的。如果结合Kafka连接件使用，KSQL能够实现由批量数据整合转变为联机数据整合。你可以使用数据流-表连接，借助存储在表中的元数据来丰富数据流，或者将数据流加载到另一个系统之前，对PII（个人身份信息）数据执行简单的过滤。

4、应用程序开发

许多应用程序将输入数据流转换成输出数据流。比如说，负责为在线商店重新排序库存少的产品的进程可能需要销售和发货方面的数据流，才能计算订单数据流。

至于用Java编写的更复杂的应用程序，Kafka的原生数据流API可能正是我们所需要的。不过针对简单的应用程序，或者对Java编程不感兴趣的团队，简单的SQL界面也许正是它们所寻找的。

KSQL中的核心抽象

KSQL在内部使用Kafka的Streams API（https://kafka.apache.org/documentation/streams/），它们使用同样的核心抽象来用于Kafka端的数据流处理。KSQL中有两个核心抽象，它们对应于Kafka Streams中的两个核心抽象，让你可以处理Kafka主题：

1、STREAM
：数据流是无限序列的结构化数据（“事实”，fact）。比如说，我们可能有一个财务交易数据流，比如“Alice向Bob打款100美元，然后Charlie向Bob打款50美元”。数据流中的事实是不可变的，这意味着新的事实可以插入到数据流中，但现有的事实根本无法被更新或删除。数据流可以由Kafka主题来创建，或由现有的数据流和表来生成。

```
CREATE STREAM pageviews (viewtime BIGINT, userid VARCHAR, pageid VARCHAR)
WITH (kafka_topic='pageviews', value_format='JSON');
```

2.、TABLE
：表是STREAM或另一个TABLE的视图，它表示不断变化的事实的集合。比如说，我们可能有一个表，含有最新的财务信息，比如“Bob当前的帐户余额是150美元”。它相当于传统的数据库表，但是由数据加窗之类的数据流语义加以丰富。表中的事实是不可变的，这意味着新的事实可以插入到表中，但现有的事实根本无法被更新或删除。表可以由Kafka主题来创建，或由现有的数据流和表来生成。

```
CREATE TABLE users (registertime BIGINT, gender VARCHAR, regionid VARCHAR, userid  VARCHAR)
```

```
WITH (kafka_topic='users', value_format='DELIMITED');
```

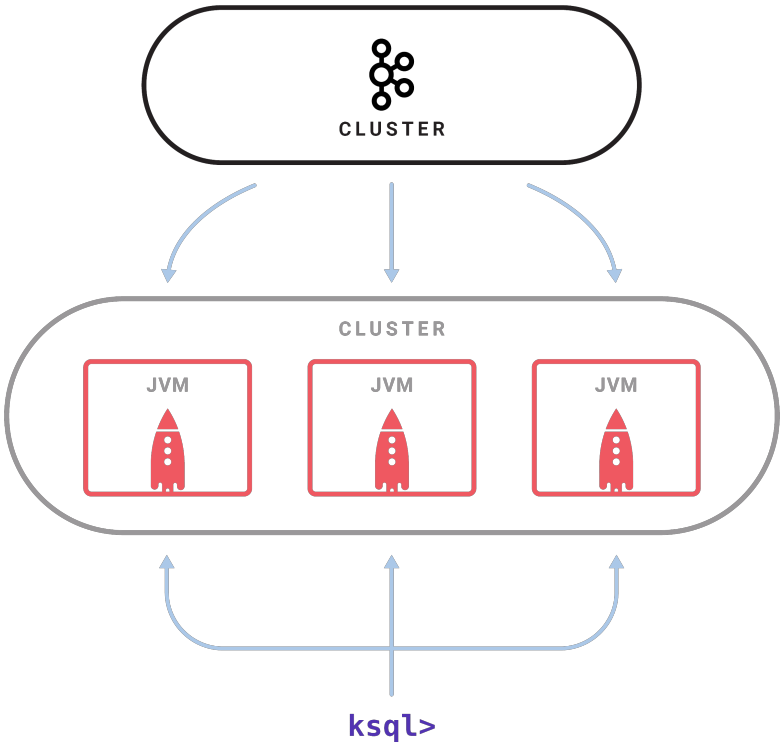
KSQL简化了数据流应用程序，因为它完全整合了表和数据流这两个概念，允许将代表事实现状的数据流与代表当前发生的事件的表连接起来。Apache Kafka中的主题可以表示为KSQL中的STREAM或TABLE，具体取决于主题处理的预期语义。比如说，如果你想把主题中的数据作为一系列独立值来读取，可以使用CREATE STREAM。这种数据流的一个示例是获取页面视图事件的主题，其中每个页面视图事件是不相关的，彼此独立。另一方面，如果你想把主题中的数据作为可更新值的不断变化的集合来读取，可以使用CREATE TABLE。说到应该在KSQL中作为TABLE来读取的话题，这方面的一个例子是获取用户元数据的主题，其中每个事件表示特定用户ID的最新元数据，无论是用户的姓名、地址还是喜好选择。

KSQL实战：实时点击流分析和异常检测

不妨看一个实际的演示。此演示显示了你如何将KSQL用于实时监控、异常检测和警报。针对点击流数据的实时日志分析有好几种形式。在本文例子中，我们标记出了在测试Web服务器上占用太多带宽的恶意用户会话。监控恶意用户会话是sessionization的许多应用之一。不过笼统地说，会话是用户行为分析的基本模块。一旦你按照特定的会话标识符将用户和事件关联起来，就可以构建许多类型的分析机制，从简单的度量指标（比如访问次数），到更复杂的度量指标（比如客户转换漏斗和事件流），不一而足。我们在演示的最后环节显示了如何在Elastic支持的Grafana仪表板上实时显示KSQL查询的输出结果。

https://www.iteblog.com/pic/kafka/kafka-ksql-iteblog.mp4

内部结构



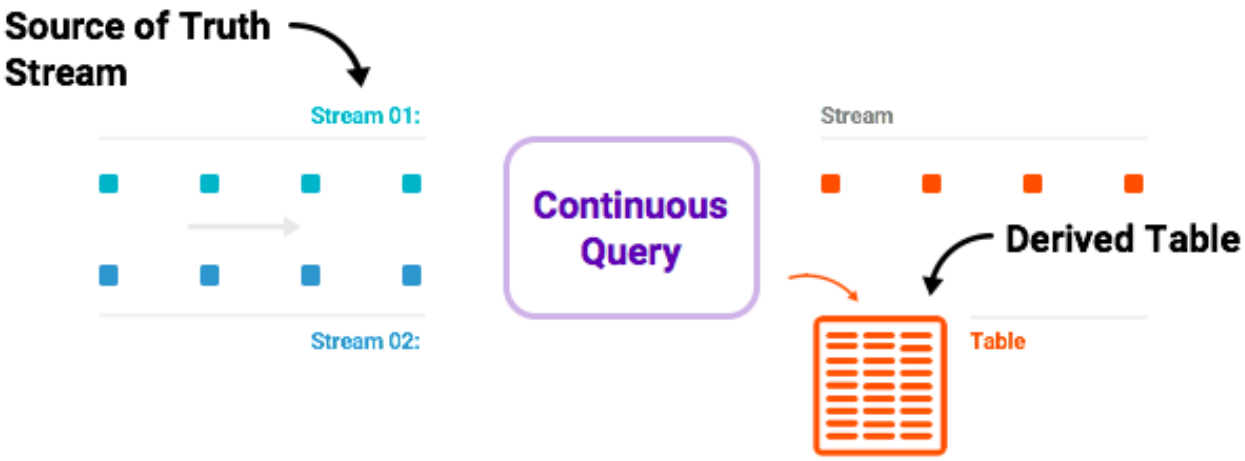
如果想及时了
解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

有一个KSQL服务器进程执行查询。一组KSQL进程作为一个集群来运行。可以通过启动KSQL服务器的更多实例来动态添加更多的处理能力。这些实例具有容错性：如果一个实例失效，另外几个会接过它处理的工作。使用交互式KSQL命令行客户软件来启动查询，客户软件通过REST API向集群发送命令。命令行让你可以检查可用的数据流和表，执行新的查询，检查运行中查询的状态，并终止运行中查询。在内部，KSQL是使用Kafka的Streams API构建的；它继承了Kafka的弹性可扩展性、先进的状态管理及容错功能，还支持Kafka最近推出的只处理一次（exactly-once proecessing）语义。KSQL服务器嵌入这个机制，另外添加了分布式SQL引擎（包括一些新颖的功能，比如提升查询性能的字节码自动生成）以及用于查询和控制的REST API。

Kafka + KSQL颠覆数据库

过去我们谈论了颠覆数据库（https://www.confluent.io/blog/turning-the-database-inside-out-with-apache-samza/），现在我们通过为由内到外发生变化的数据库添加SQL层来实现颠覆。

在关系数据库中，表是核心抽象，日志是实现细节。而在以事件为中心的世界，数据库已被颠覆，核心抽象不是表，而是日志。这些表只是来源于日志，随着新数据进入到日志，表不断更新。中央日志是Kafka，KSQL是引擎，让你可以创建所需的物化视图，并将它们表示为持续更新的表。然后，你可以针对这类流式表运行时间点查询（KSQL即将发布该功能），为日志中的每个键获得最新值，采取持续不断的方式。



如果想及时了
解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

使用Kafka和KSQL彻底颠覆数据库，这对公司中可以用数据流方式来表示和处理的所有数据派什么用场带来了很大的影响。Kafka日志是数据流的核心存储抽象，允许进入到离线数据仓库的相同数据现在可用于数据流处理。其他一切数据是基于日志的流式物化视图，无论是各种数据库、搜索索引还是公司中的其他数据服务系统。现在可以使用KSQL，以数据流的方式，执行创建这些派生视图所需的所有数据丰富和ETL。监控、安全、异常及威胁检测、分析以及故障应对可以实时执行，而不是为时太晚才执行。所有这些可供任何人使用，只要借助一种对你的所有Kafka数据而言简单又熟悉的SQL界面：KSQL。

如果想及时了
解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

KSQL的下一站是什么？

我们在发布开发者预览版的KSQL，开始围绕它构建社区，并征集反馈意见。我们计划与开源社区合作，增添另外几项功能，从而将它变成一种可准备部署到生产环境的系统：从KSQL的质量、稳定性和可操作性，直到支持更丰富的SQL语法（包括进一步的聚合功能和针对连续表的时间点SELECT），对迄今为止计算的数据执行快速查询。

如何获取KSQL？

你可以试一试KSQL快速入门和上述演示来实际体验一下。欢迎你反馈缺少什么功能，或者哪些方面可以改进：欢迎到Confluent Community Slack上的#KSQL频道发表任何想法或反馈，如果你发现了错误，欢迎提交GitHub问题单；我们很乐意与早期采用者密切合作，所以请踊跃参与。我们期待与开源社区的其余人合作，让KSQL变成一项出色的技术。

本文英文原文：https://www.confluent.io/blog/ksql-open-source-streaming-sql-for-apache-kafka/