

2015年8月22日星期六

HUAWEI ENTERPRISE **A BETTER WAY**

StreamCQL 构建在分布式流处理 平台上的查询语言

enterprise.huawei.com

HUAWEI TECHNOLOGIES CO., LTD.



1

个人简介

2

StreamCQL简介

3

StreamCQL与流处理平台

4

StreamCQL功能

5

StreamCQL后期开发计划

个人简介

何志强，StreamCQL
核心作者之一，实时处理资
深架构师。

2011年4月加入华为，
目前在华为进行实时分析开
发。一直从事大数据分析、
分布式计算、流式处理等方
面的研究及开发。



StreamCQL简介

StreamCQL(Stream Continuous Query Language)是建立在分布式流处理平台基础上的查询语言(CQL)，架构支持构建在多种流处理引擎之上，目前主要适配 Storm。

当前多数流处理平台仅提供分布式处理能力，业务逻辑开发复杂，流计算业务功能较弱，存在业务逻辑重用性不高、重复开发、开发效率低下等问题。StreamCQL提供了较丰富的分布式流计算功能，除了具有过滤、转换等传统的SQL基本能力之外，StreamCQL引入基于窗口的计算，提供窗口数据的统计、关联等能力，以及流数据的拆分、合并等功能。

CQL

2003年，斯坦福大学《The CQL Continuous Query Language: Semantic Foundations and Query Execution*》

CQL(continuous query language)是一个基于SQL的声明式语言，它用于在流(streams)和可更新关系(updatable relation)上进行可持续查询。

The CQL Continuous Query Language: Semantic Foundations and Query Execution*

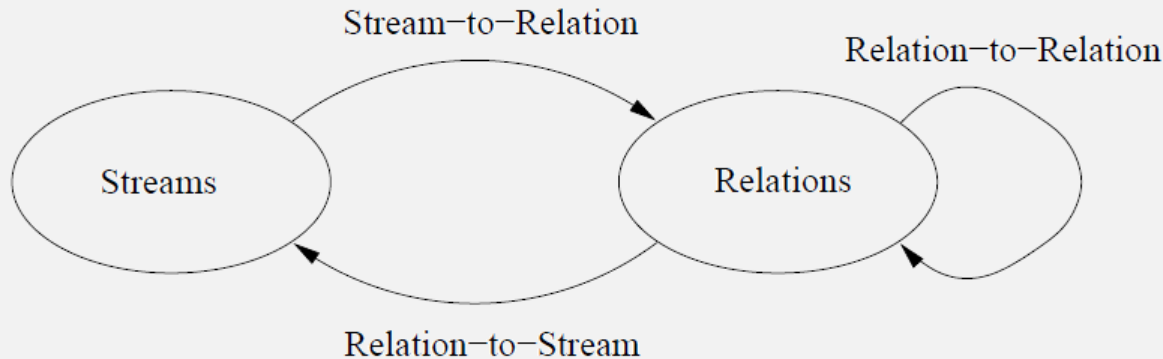
Arvind Arasu and Shivnath Babu and Jennifer Widom
Stanford University
{arvinda,shivnath,widom}@cs.stanford.edu

Abstract

CQL, a Continuous Query Language, is supported by the STREAM prototype Data Stream Management System at Stanford. CQL is an expressive SQL-based declarative language for registering continuous queries against streams and updatable relations. We begin by presenting an abstract semantics that relies only on “black box” mappings among streams and relations. From these mappings we define a precise and general interpretation for continuous queries. CQL is an instantiation of our abstract semantics using SQL to map from relations to relations, window specifications derived from SQL-99 to map from streams to relations, and three new operators to map from relations to streams. Most of the CQL language is operational in the STREAM system. We present the structure of CQL’s query execution plans as well as details of the most important components: operators, inter-operator queues, synopses, and sharing of components among multiple operators and queries. Examples throughout the paper are drawn from the *Linear Road* benchmark recently proposed for Data Stream Management Systems. We also curate a public repository of data stream applications that includes a wide variety of queries expressed in CQL.

CQL

- 流：一个流就是一组(无穷)元素的集合, $\langle s, t \rangle$ 表示元组s在时间t到达流.
- 关系: 一个关系就是一个随时间变化的数据集。



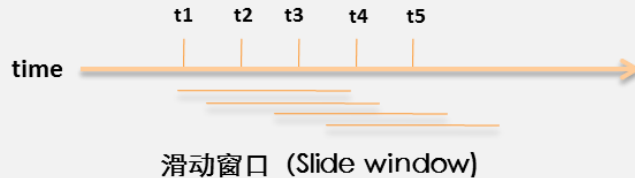
```
Select Distinct L.vehicleId, L.segNo, L.dir, L.hwy
From SegSpeedStr [Range 30 Seconds] as A,
      SegSpeedStr [Partition by vehicleId Rows 1] as L
Where A.vehicleId = L.vehicleId
```

CQL窗口

窗口就是一个有限范围内、任意一个时间点的数据状态快照。



- “窗口”概念的提出，让“流”中的数据有了“边界”
- 窗口中的数据 = 数据库中的一张静态表

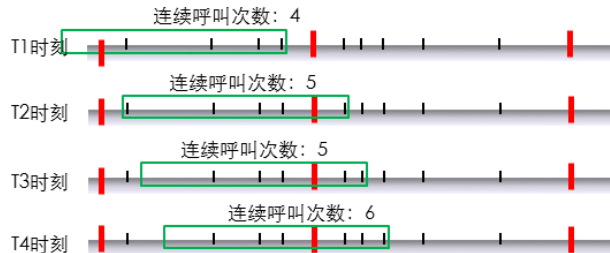


电信信令监控的一个业务（恶意呼叫）：

寻找在连续1分钟内发起大于5次呼叫的手机用户

```
SELECT * FROM CallEvent[range 60 seconds slide]
```

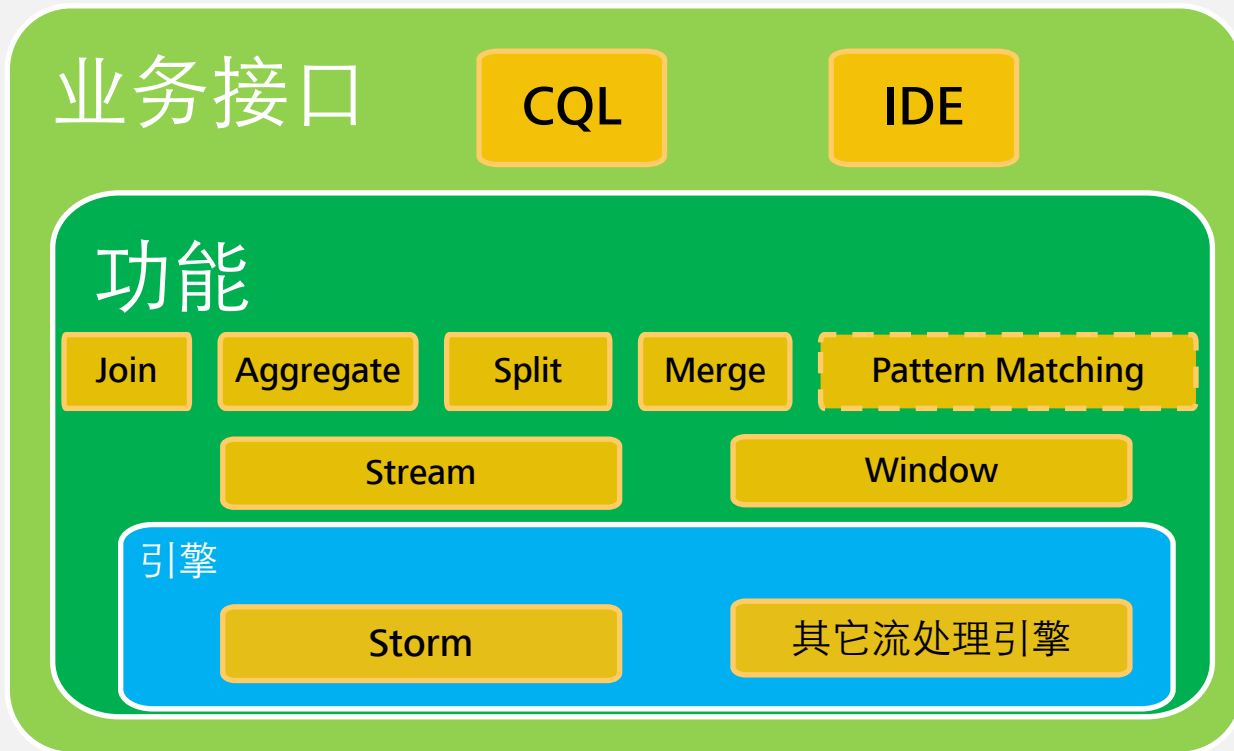
```
GROUP BY strlmsi HAVING count(*)>5
```



enterprise.huawei.com ■ Huawei Confidential ■ 8

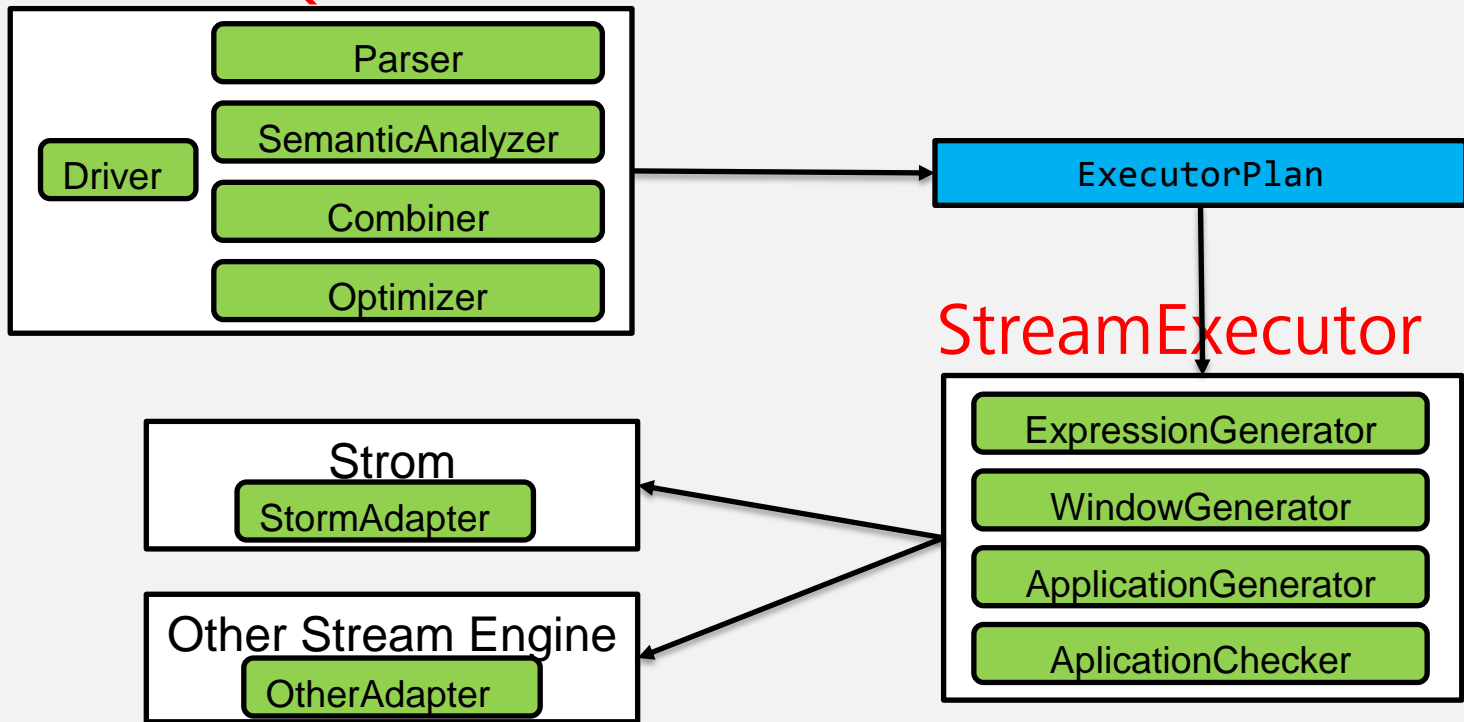


StreamCQL与流处理平台



StreamCQL架构

StreamCQL



StreamCQL带来Storm开发方式的转变

Storm 原生API

```
//Def Input:
public void open(Map conf, TopologyContext
context, SpoutOutputCollector collector) {...}
public void nextTuple() {...}
public void ack(Object id) { ...}
public void
declareOutputFields(OutputFieldsDeclarer declarer)
{...}
//Def logic:
public void execute(Tuple tuple,
BasicOutputCollector collector) {...}
    public void
declareOutputFields(OutputFieldsDeclarer ofd) {...}
//Def Output:
    public void execute(Tuple tuple,
BasicOutputCollector collector) {...}
    public void
declareOutputFields(OutputFieldsDeclarer ofd) {...}
//Def Topology:
public static void main(String[] args) throws
Exception {...}
```

StreamCQL

```
--Def Input:
CREATE INPUT STREAM S1 ...

--Def logic:
INSERT INTO STREAM filterstr SELECT * FROM S1
WHERE name="HUAWEI";

--Def Output:
CREATE OUTPUT STREAM S2...

--Def Topology:
SUBMIT APPLICATION test;
```

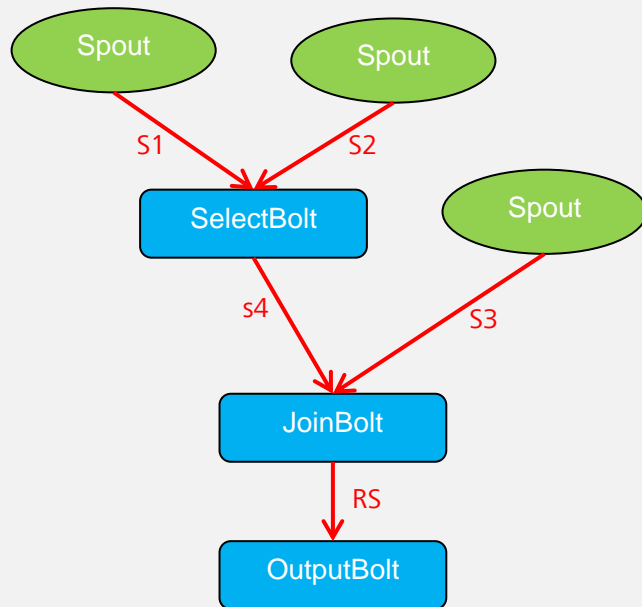
StreamCQL生成Storm拓扑

```
CREATE INPUT STREAM S1
(...)
SOURCE KafkaInput PROPERTIES(...);

CREATE INPUT STREAM S2...;
CREATE INPUT STREAM S3...;

CREATE OUTPUT STREAM RS(...)
SINK kafkaOutput PROPERTIES(...)

INSERT INTO s4 SELECT *,1 FROM S1;
INSERT INTO S4 SELECT *,2 FROM S2;
INSERT INTO STREAM rs
  SELECT * FROM S4[ROWS 10 BATCH]
    INNER JOIN S3[RANGE 3 HOURS SLIDE]
    ON s4.id=s3.id
  WHERE s4.id > 5;
```



StreamCQL功能—创建输入流

```
CREATE INPUT STREAM example
(
    eventId INT,
    eventDesc STRING
)
COMMENT "this is a example of create input stream."
SERDE SimpleSerDe
    PROPERTIES (separator = "|")
SOURCE TCPClientInput
    PROPERTIES ( server = "127.0.0.1",port = "9999" )
PARALLEL 2;
```

StreamCQL功能—Join

```
INSERT INTO STREAM rs
  SELECT * FROM S1[RANGE 20 SECONDS BATCH]
  JOIN S2[RANGE UNBOUNDED] ON s1.id=s2.id WHERE s1.id > 5;
```

```
INSERT INTO STREAM rs
  SELECT * FROM S1[ROWS 10 SLIDE]
  LEFT JOIN S2[range today ts] ON s1.id=s2.id;
```

StreamCQL功能—窗口

语法	名称	说明
S[ROWS N1 BATCH]	长度跳动窗	窗口内最大保存N1个事件，当有新事件产生的时候，窗口内每攒满N1个事件，就过期一次。同时过期的所有事件处于同一批次。
S[RANGE T1 SLIDE]	时间滑动窗	窗口内保存最近T1时间范围内的数据，T1是一个时间单位，可以加入Seconds等时间单位。窗口内的事件依次过期。每个过期事件的批次都不同。
S[ROWS N1 SLIDE PARTITION BY EXP1]	分组长度滑动窗	同长度滑动窗，但是加入了分组的概念，事件归属于不同的分组，每个分组的长度为N1，逐个过期。
S[RANGE T1 SLIDE TRIGGER BY EXP1]	事件驱动时间滑动窗	窗口内保存最近T1时间单位的数据，exp1是一个返回值为时间类型的表达式，每次产生数据之后，都会和窗口内的数据做对比，然后将大于T1时间单位的数据吐出，每次只吐出一个数据。
.....		

StreamCQL功能—窗口

--按照type对窗口内数据进行分组，每组容量为10

```
SELECT * FROM transformEvent[ROWS 10 SLIDE PARTITION BY TYPE];
```

--时间排序窗，一般用来解决数据乱序问题

```
SELECT * FROM transformEvent[RANGE 1000 MILLISECONDS SORT BY dtc];
```

--事件驱动时间滑动窗

```
INSERT INTO STREAM rs sum(OrderPrice),avg(OrderPrice),count(OrderPrice)  
FROM transformEvent[RANGE 10 SECONDS SLIDE TRIGGER by TS EXCLUDE now];
```

--保存周期为一个自然天的分组窗

```
INSERT INTO STREAM rs select id,name,count(id)  
FROM transformEvent[RANGE TODAY ts PARTITION BY TYPE]  
WHERE id > 5 GROUP BY TYPE HAVING id > 10;
```


StreamCQL功能—Split功能

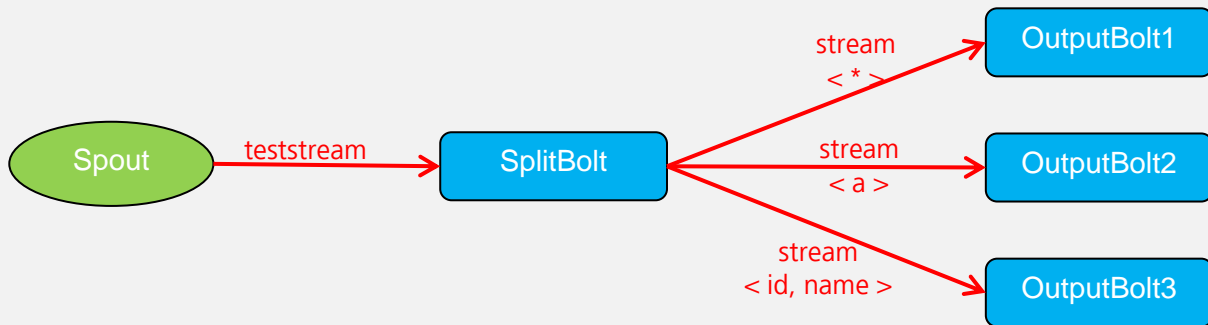
FROM teststream

INSERT INTO STREAM s1 SELECT *

INSERT INTO STREAM s2 SELECT a

INSERT INTO STREAM s3 SELECT id, name WHERE id > 10

PRARLLEL 4;



StreamCQL后期

- 模式匹配(follow by)
- CQLServer(CQL远程提交，多语言接口，JDBC接口)
- 可靠性增强
- Github开源

团队介绍

华为实时分析团队，致力于大数据的实时计算、实时分析及挖掘等领域；团队中每位成员都在大数据和实时分析领域深耕细作多年，同时在电信、银行、医疗、公安等多行业具有丰富的实战经验，欢迎各位有志之士加入，和我们一起玩转大数据！

产品公众号：FusionInsight

联系人：沙欧 shaou@huawei.com



HUAWEI ENTERPRISE ICT SOLUTIONS **A BETTER WAY**

Copyright©2013 Huawei Technologies Co., Ltd. All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.