



# Conger 数据流管理系统 持续型查询语言的研究与实现

## 当 CQL 遇见



### 数据流管理系统简介

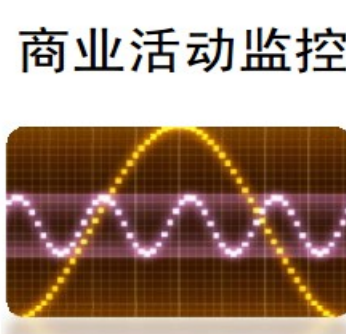
数据流的概念最早由 Henzinger 等人在 1998 年提出，指的是“只能以事先规定好的顺序被读取一次或几次的一个数据序列”。与传统数据相比，数据流的主要特点有：无限、大量、快速、数据量大小随时间变化且事先不确定。这些与传统数据不同的特点，使得数据流管理系统与传统数据库管理系统有着很大的差别。

数据流处理系统 (DSMS) 的应用

- 网络流量监控
- 道路交通监控
- 股票信息监控



网络攻击



商业活动监控

Aurora (Borealis) 是布兰代斯大学，布朗大学，麻省理工学院三个大学联合开发的通用目的的数据流处理系统。该系统提供了一种基于操作符网络的查询语言，其最重要的结构是 box 和 arrow。每一个 box 代表一个操作 (operator)，即处理单元；每一个有向的 arrow 代表数据流从一个 box 流到另外一个 box。Aurora 不支持 CQL 查询接口。

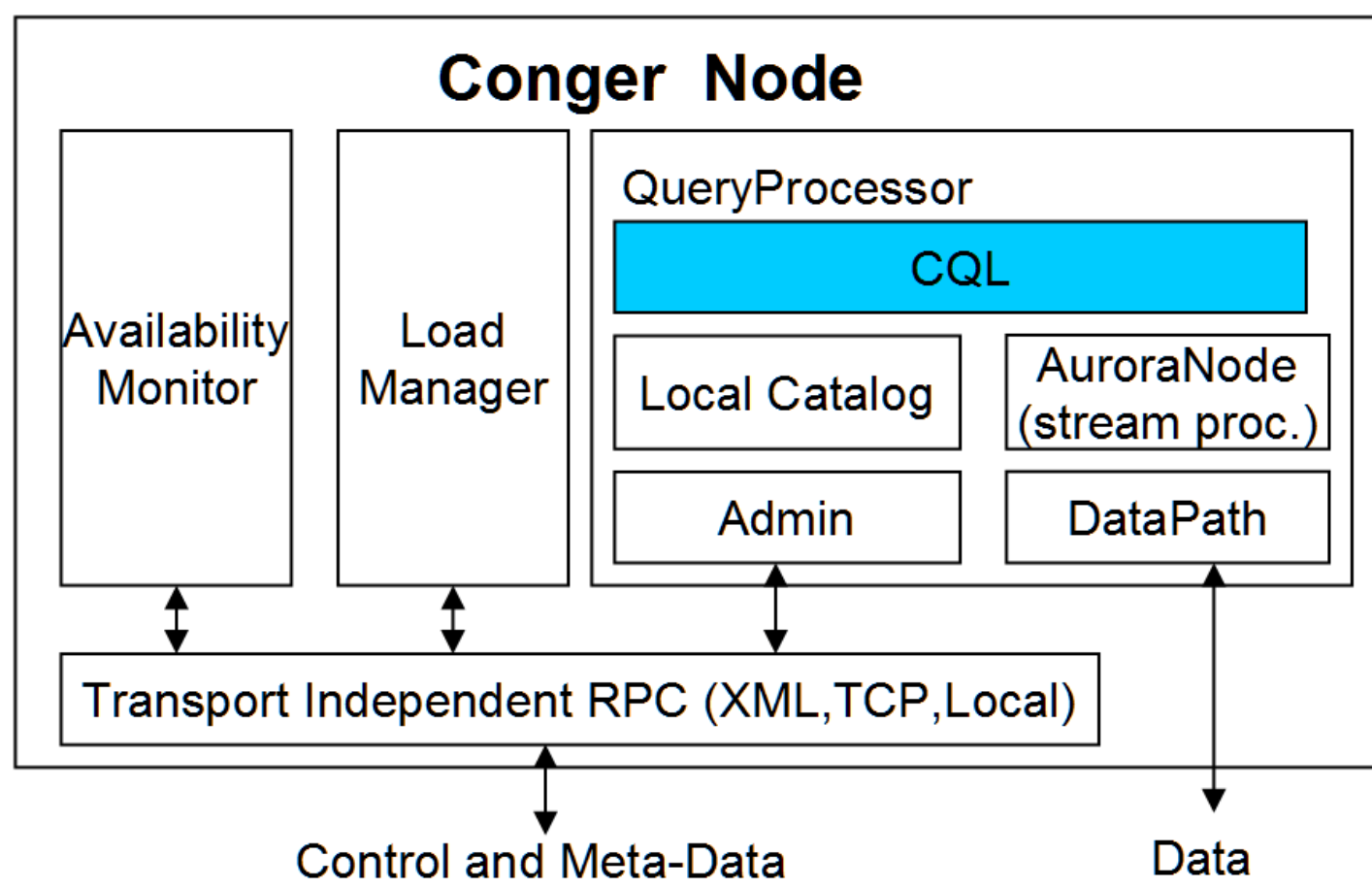
### 持续性查询语言 (CQL) 简介

Continuous Query Language：一种在数据流上进行持续型查询的语言，CQL 是 Stanford 最初在 2002 - 2006 年开发的流处理引擎 STREAM 中的查询语言。

```
SELECT time, COUNT(car_id) AS volume, MIN(speed) AS min_speed,
       MAX(speed) AS max_speed, AVG(speed) AS avg_speed
FROM positionreport [RANGE 30 SECONDS SLIDE 3 SECONDS]
```

### Conger 数据流管理系统

我们基于 Aurora 开发了支持 CQL 的数据流管理系统 Conger。我们结合 Aurora 的操作符特点设计了 Conger CQL，在 Aurora 基于操作符网络的查询语言的基础之上增加了 CQL 语言的查询接口。与 STREAM CQL 相比，Conger CQL 能够更加精确和简洁的支持 Aurora 提供的数据流查询语义。Conger CQL 参考 SQL：2003 设计持续型查询语言，我们在此之上进行扩展，增加数据流查询的语义。右图分别为 Conger 的系统结构图已经 Conger CQL 的语义定义的片段。



Conger 系统结构图

```
sfw_block
: select_clause from_clause
(opt_group_by_clause? opt_having_clause?
 | opt_where_clause opt_group_by_clause? opt_having_clause?
)
-> ^(TOK_SFW from_clause select_clause
    opt_where_clause? opt_group_by_clause? opt_having_clause?)
;

select_clause
: KW_SELECT (STAR | non_mt_projterm_list)
-> ^(TOK_SELECT STAR? non_mt_projterm_list?)
;
```

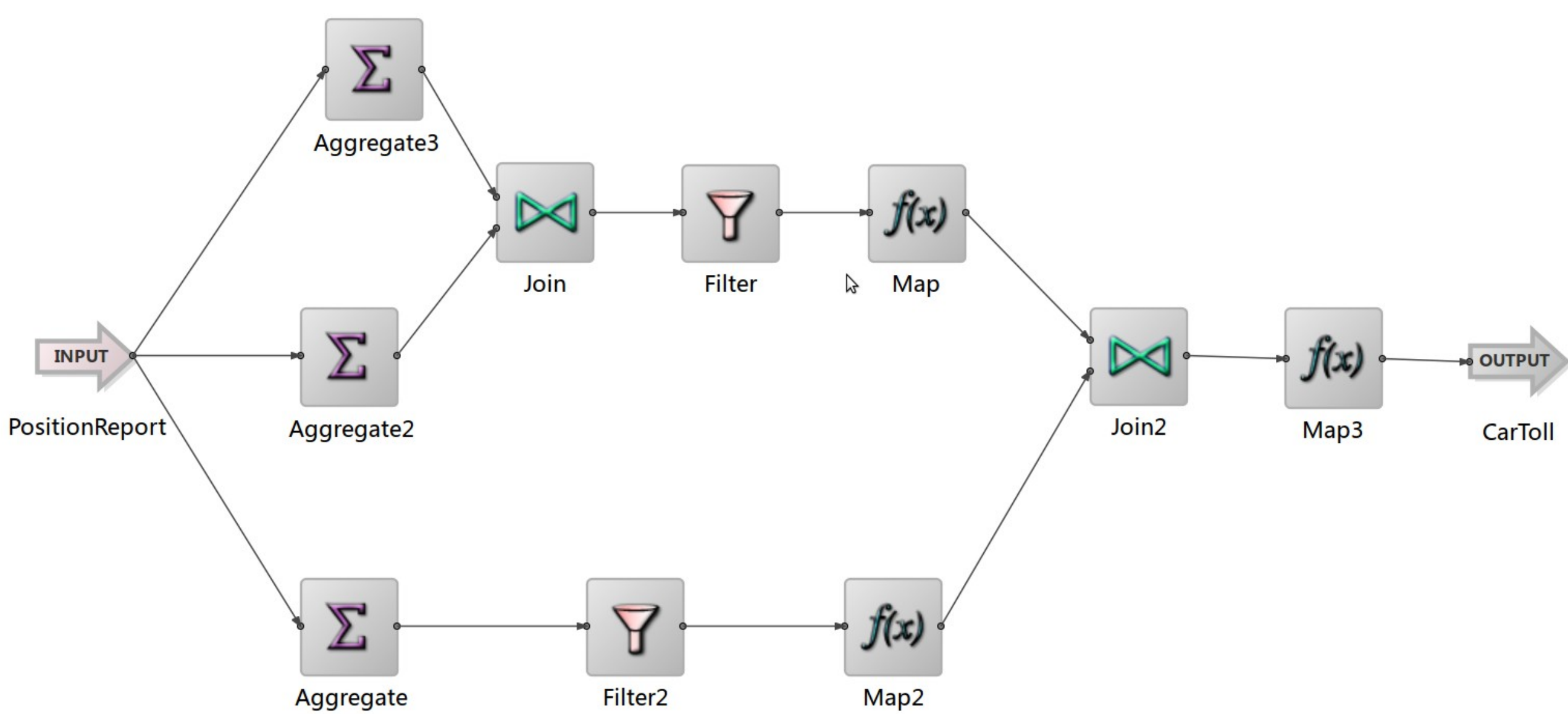
Conger CQL 语言定义 (片段)

### Linear Road Benchmark 的实现

Aurora 和 STREAM 的研究团队开发了一个数据流处理器的基准测试平台，称为 Linear Road Benchmark。Linear Road 模拟了一个高速公路的收费系统。在 Linear Road 中通过调节高速公路的条数，可以测试数据流管理系统的处理能力，同时验证查询语言的表达能力。Linear Road 假设一个城市由 10 条并行的高速公路组成，每条高速公路被分成 100 段，每段都有一个进出口，汽车可以随意根据当前的路况选择在某一段进入或离开某一个高速公路。本演示实现了 Linear Road Benchmark 的车辆计费模块，统计每段道路的路况，并根据路况计算收费额度，同时对离开该段道路的车辆发送计费信息。本演示的输入数据流为 MIT 开发的交通微观仿真软件 MITSIMLab 生成的三小时的道路交通信息。这些仿真数据由客户端实时的发送给 Conger 服务端。

```
SELECT newcareentry.car_id AS car_id, segmenttoll.toll AS toll,
       segmenttoll.way AS way, segmenttoll.dir AS dir,
       segmenttoll.seg AS seg
FROM newcareentry [ROWS 1 SLIDE 1]
JOIN segmenttoll [RANGE 30 SECONDS SLIDE 5 SECONDS]
ON newcareentry.way = segmenttoll.way
AND newcareentry.dir = segmenttoll.dir
AND newcareentry.seg = segmenttoll.seg
```

车辆收费信息查询的 CQL 片段



本演示最后生成的物理执行计划

```
info [Car收费.cc:19] 车辆: 694, 收费: 238140, 道路: 0, 方向: 1, 路段: 89
info [Car收费.cc:19] 车辆: 694, 收费: 208860, 道路: 0, 方向: 1, 路段: 88
info [Car收费.cc:19] 车辆: 694, 收费: 181500, 道路: 0, 方向: 1, 路段: 87
info [Car收费.cc:19] 车辆: 694, 收费: 171735, 道路: 0, 方向: 1, 路段: 86
info [Car收费.cc:19] 车辆: 694, 收费: 129735, 道路: 0, 方向: 1, 路段: 85
info [Car收费.cc:19] 车辆: 694, 收费: 126960, 道路: 0, 方向: 1, 路段: 84
info [Car收费.cc:19] 车辆: 694, 收费: 98415, 道路: 0, 方向: 1, 路段: 83
info [Car收费.cc:19] 车辆: 694, 收费: 93615, 道路: 0, 方向: 1, 路段: 82
info [Car收费.cc:19] 车辆: 694, 收费: 79935, 道路: 0, 方向: 1, 路段: 81
info [Car收费.cc:19] 车辆: 694, 收费: 63375, 道路: 0, 方向: 1, 路段: 80
info [Car收费.cc:19] 车辆: 694, 收费: 29040, 道路: 0, 方向: 1, 路段: 79
info [Car收费.cc:19] 车辆: 694, 收费: 18375, 道路: 0, 方向: 1, 路段: 78
info [Car收费.cc:19] 车辆: 694, 收费: 25215, 道路: 0, 方向: 1, 路段: 77
info [Car收费.cc:19] 车辆: 694, 收费: 21660, 道路: 0, 方向: 1, 路段: 76
info [Car收费.cc:19] 车辆: 694, 收费: 3375, 道路: 0, 方向: 1, 路段: 75
info [Car收费.cc:19] 车辆: 694, 收费: 7260, 道路: 0, 方向: 1, 路段: 74
```

编号为 694 的车辆连续通过多个路段的收费信息