

优步流处理应用

袁泳 @ UBER

U B E R

优步简介



便捷交通，弹指之间





流数据 - 城市的脉搏

市场动态

当前全球有多少车可用

Query

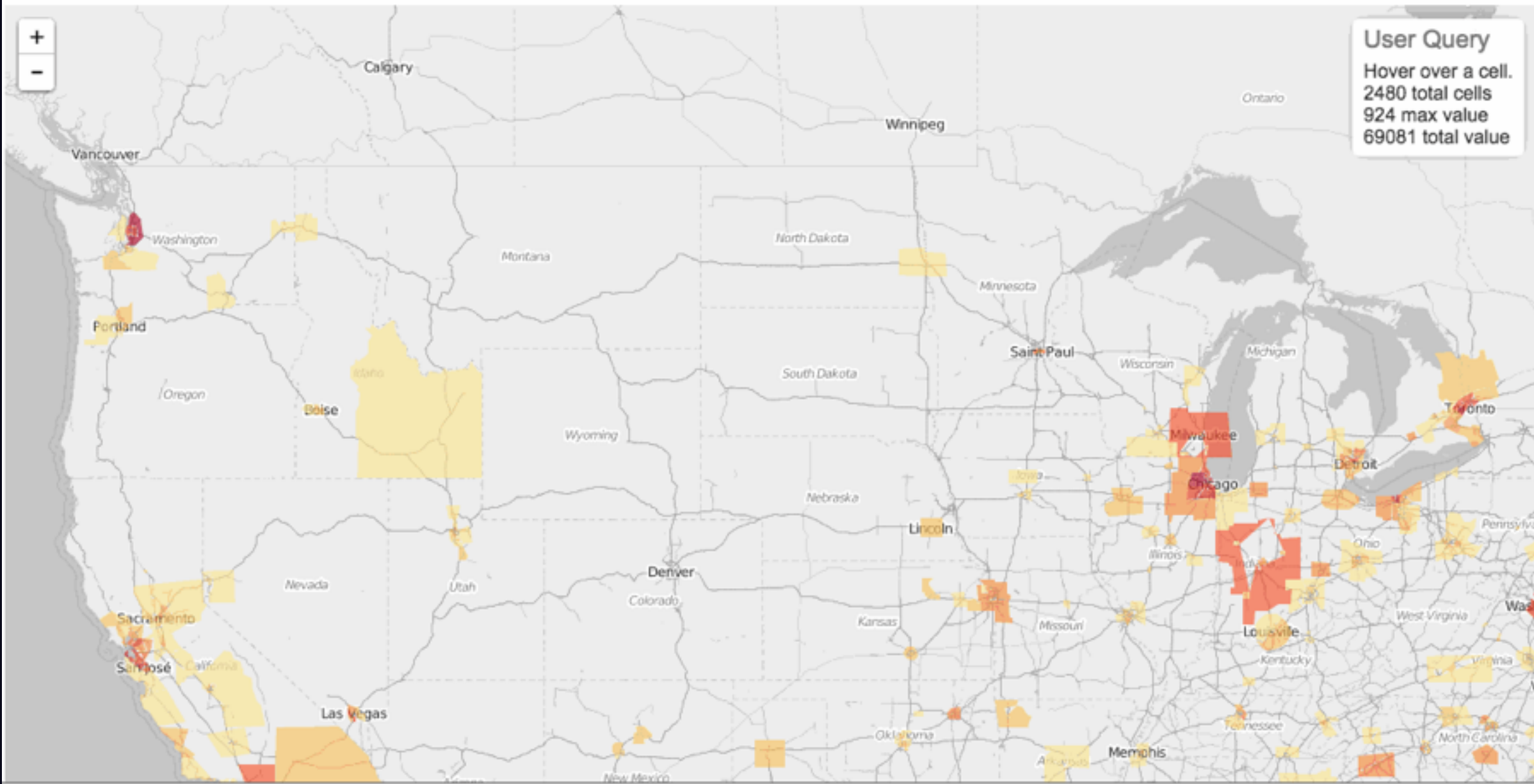
Table

Heatmap

Stats

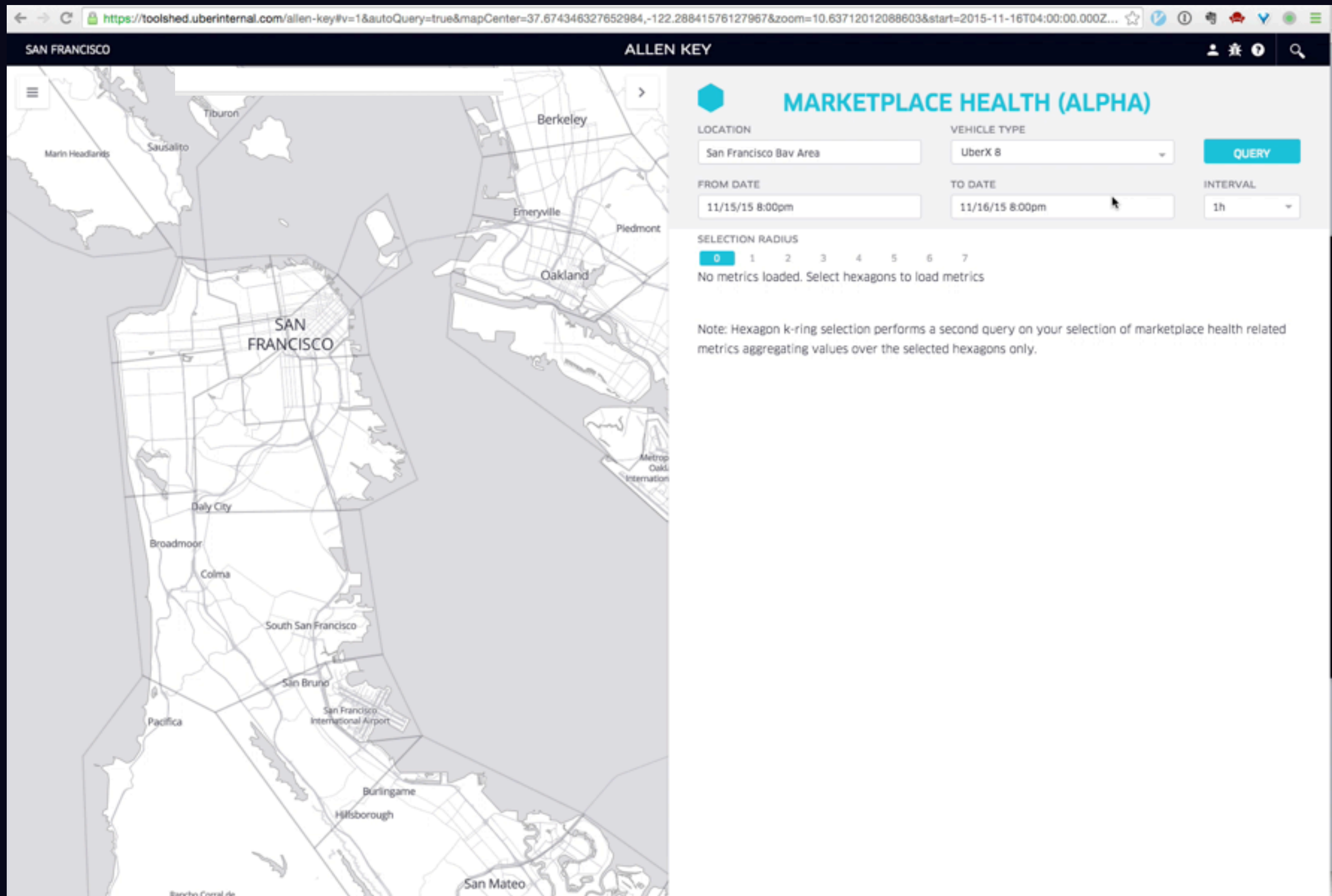


过去**10**分钟里旧金山里每个区域有多少车载客？

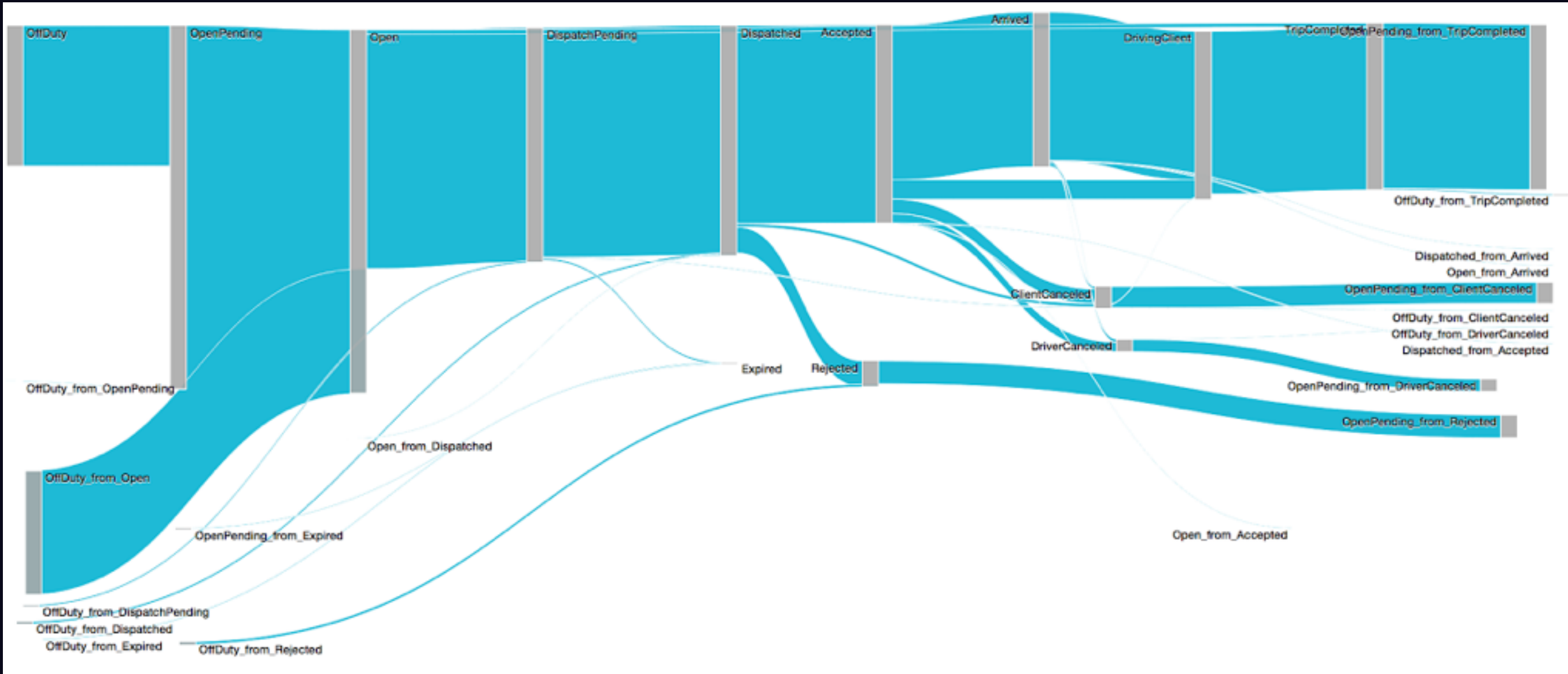
[Query](#)[Download as JSON](#)[Download as CSV](#)[Table](#)[Heatmap](#)[Stats](#)

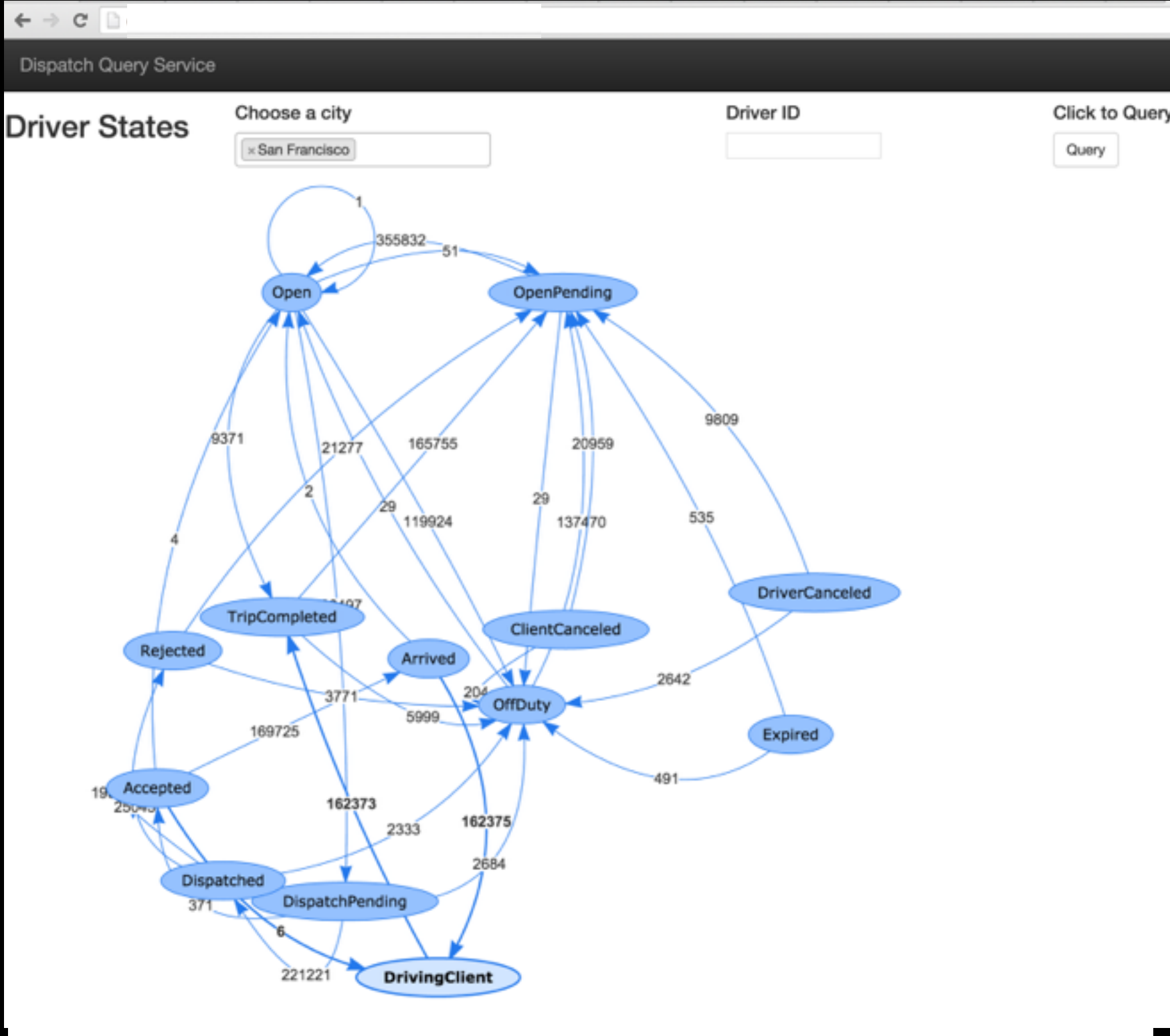
过去**10**分钟里旧金山里每个六边形有多少车载客？

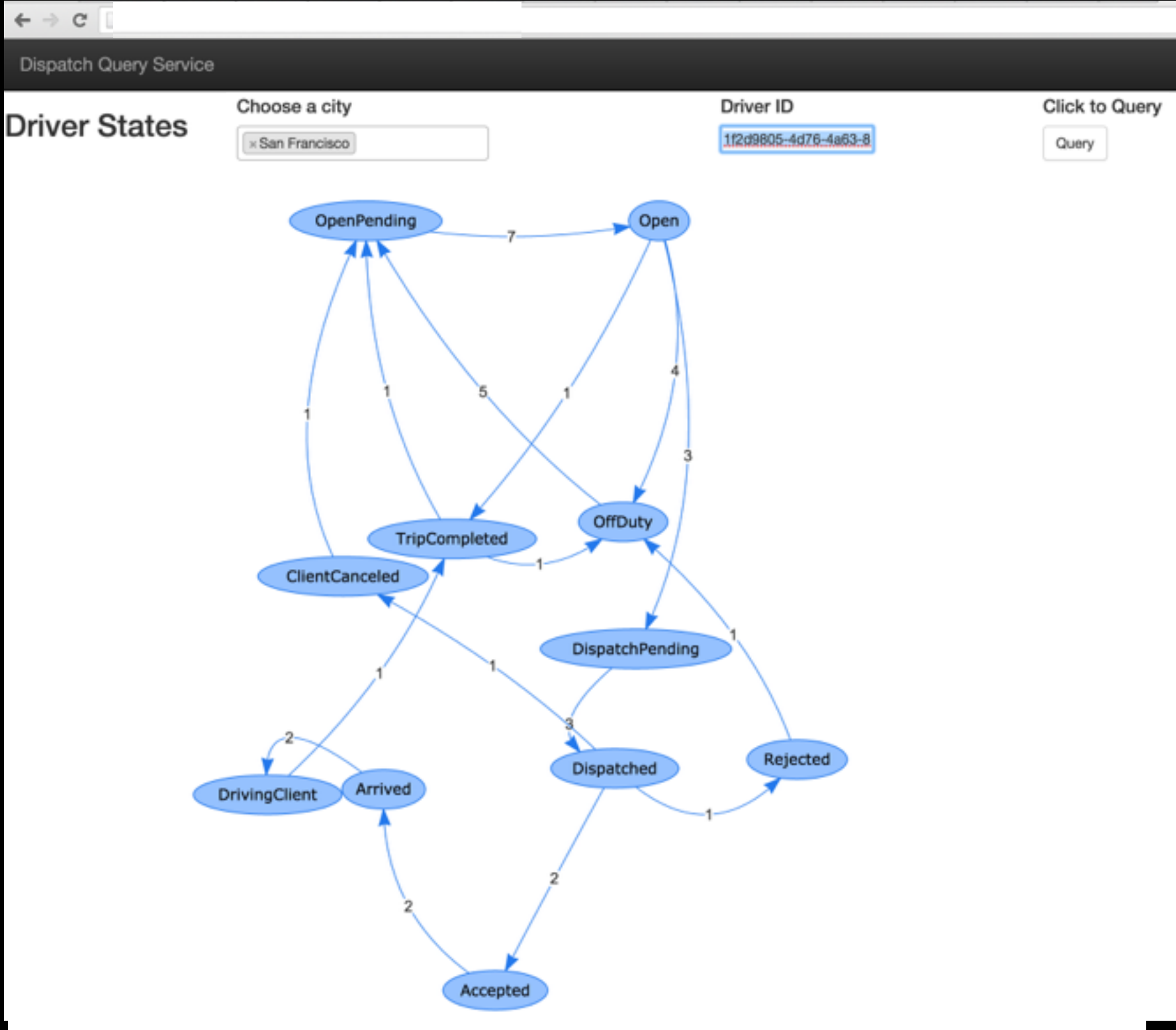
历史变化



状态跟踪



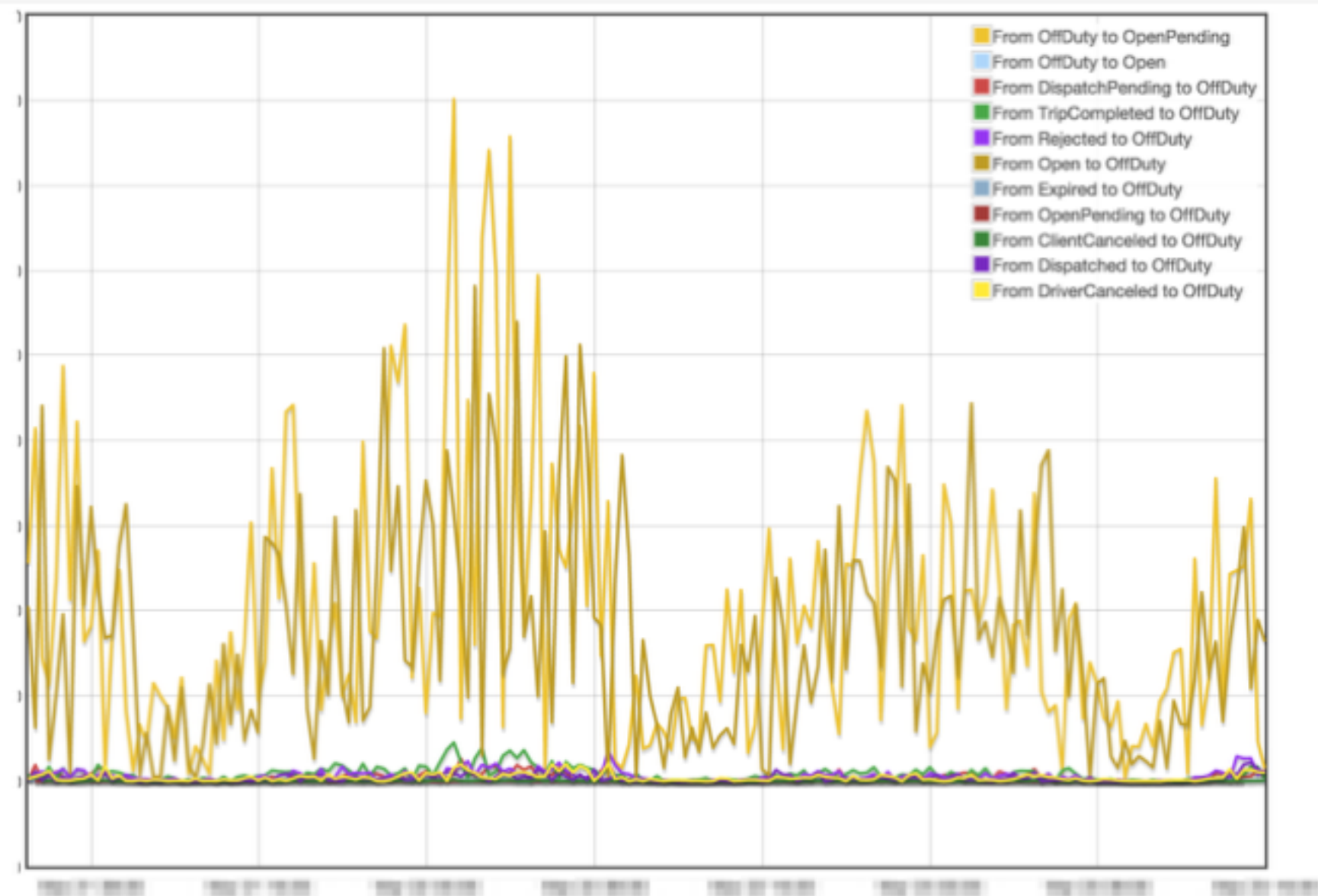




59

Driver S

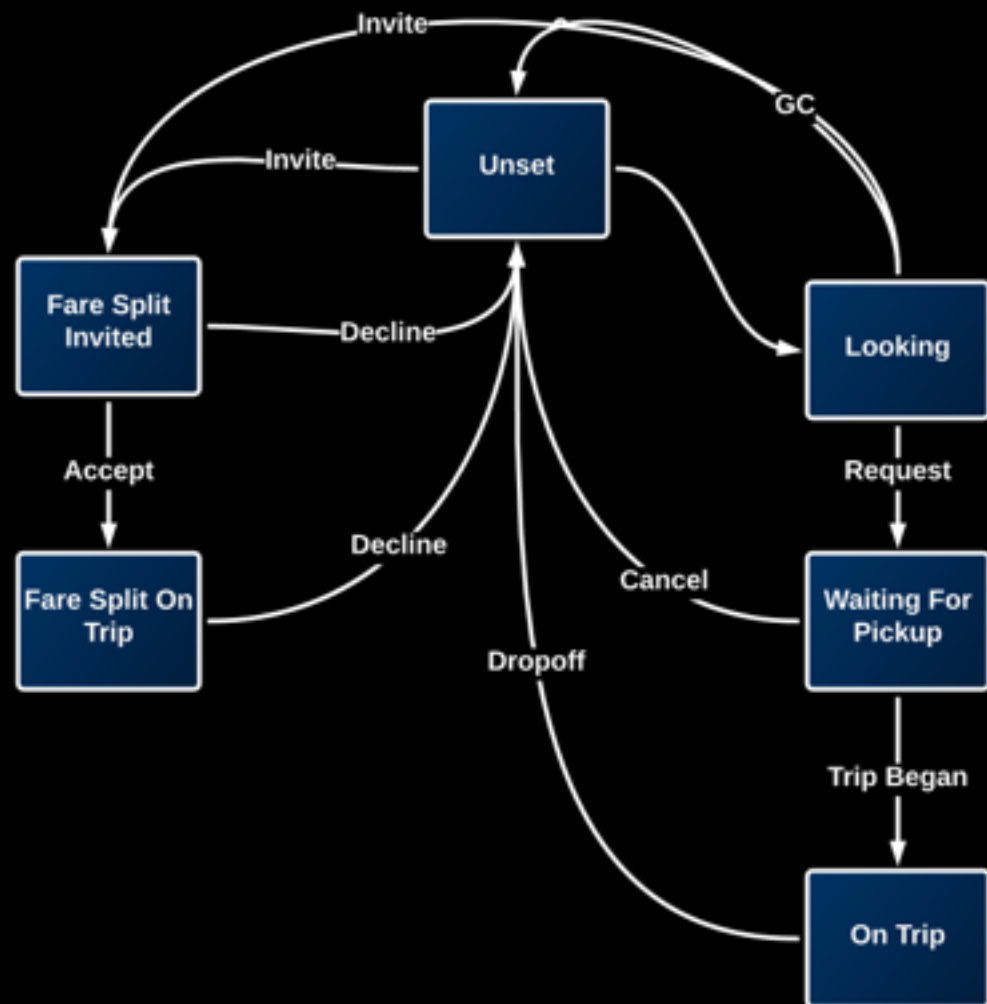
Count of State Change Over Time



出发点

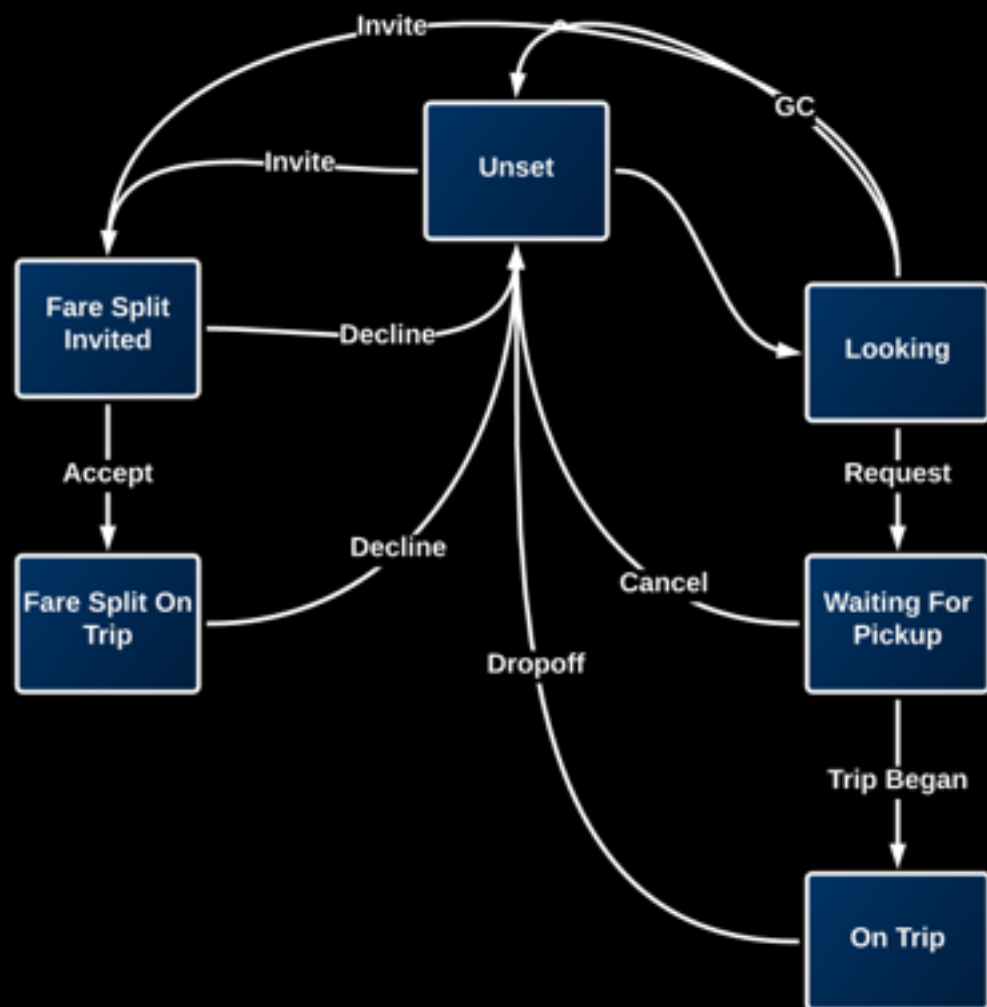
优步平台本质：分布式状态机

乘客状态

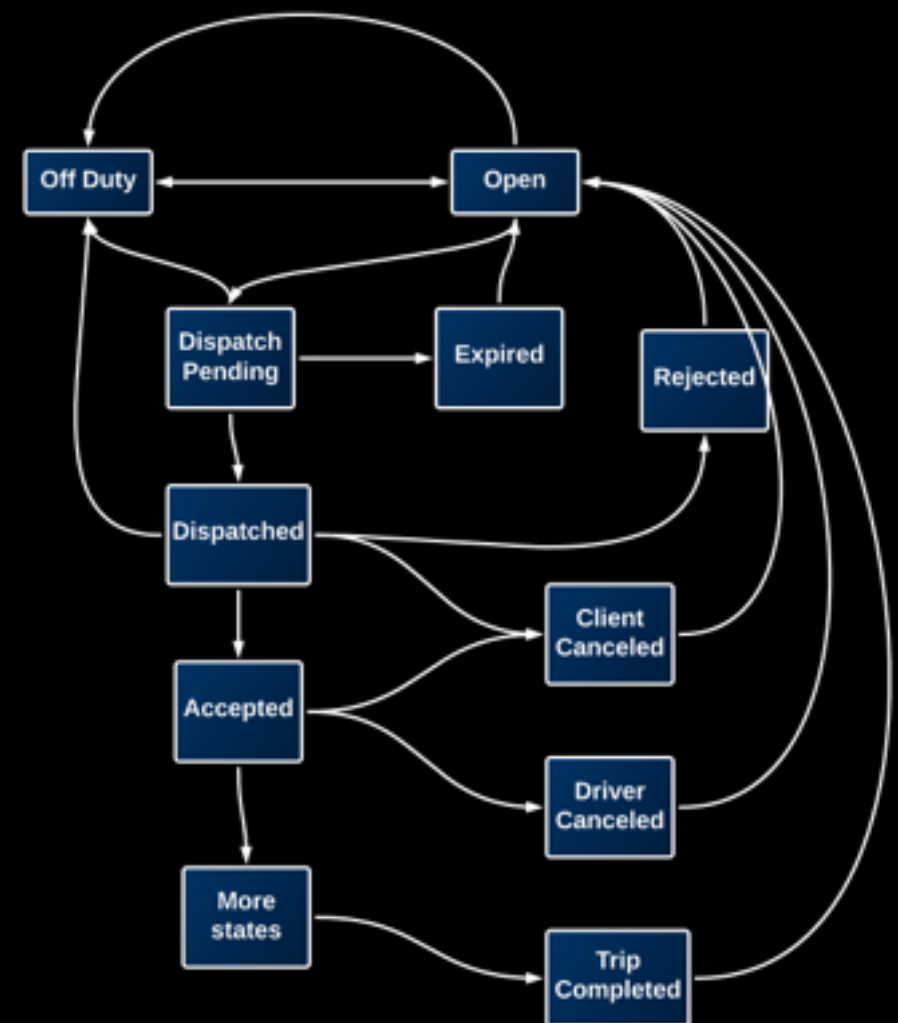


优步平台本质：分布式状态机

乘客状态



司机状态

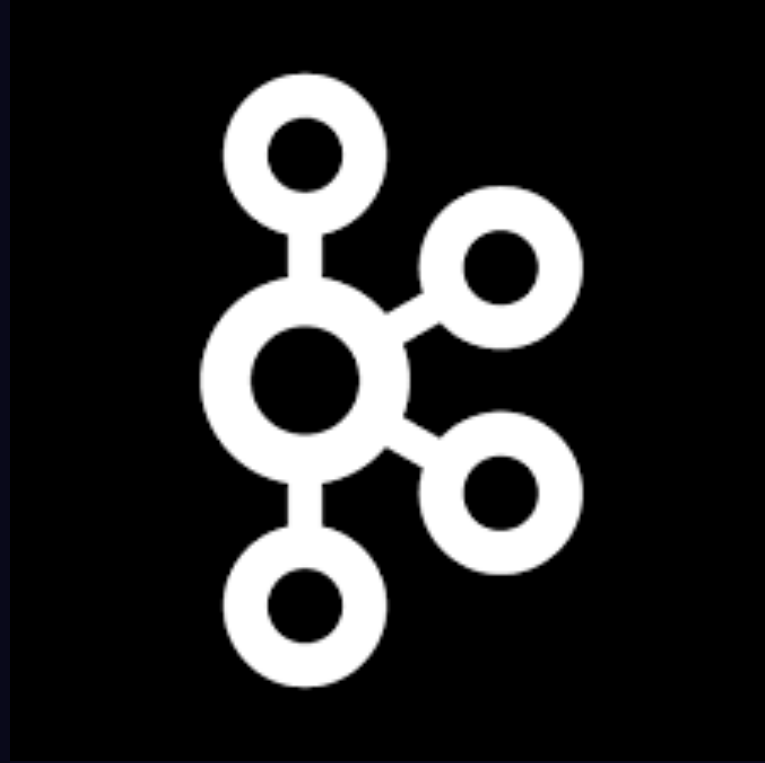


各类应用生成事件

秒级事件延迟

事件绝少丢失

收集事件消息 - 低开销、易扩展

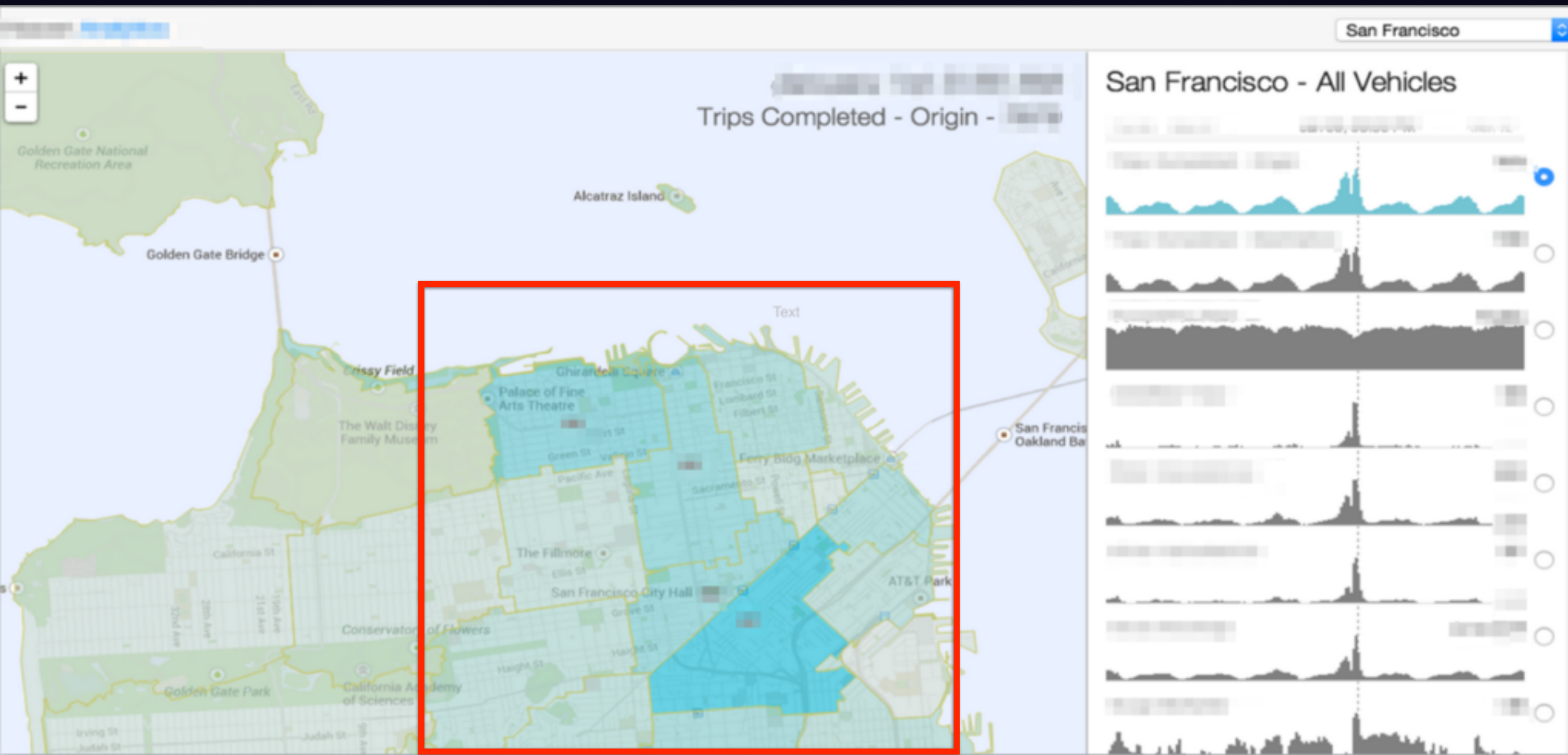


挑战在哪里？

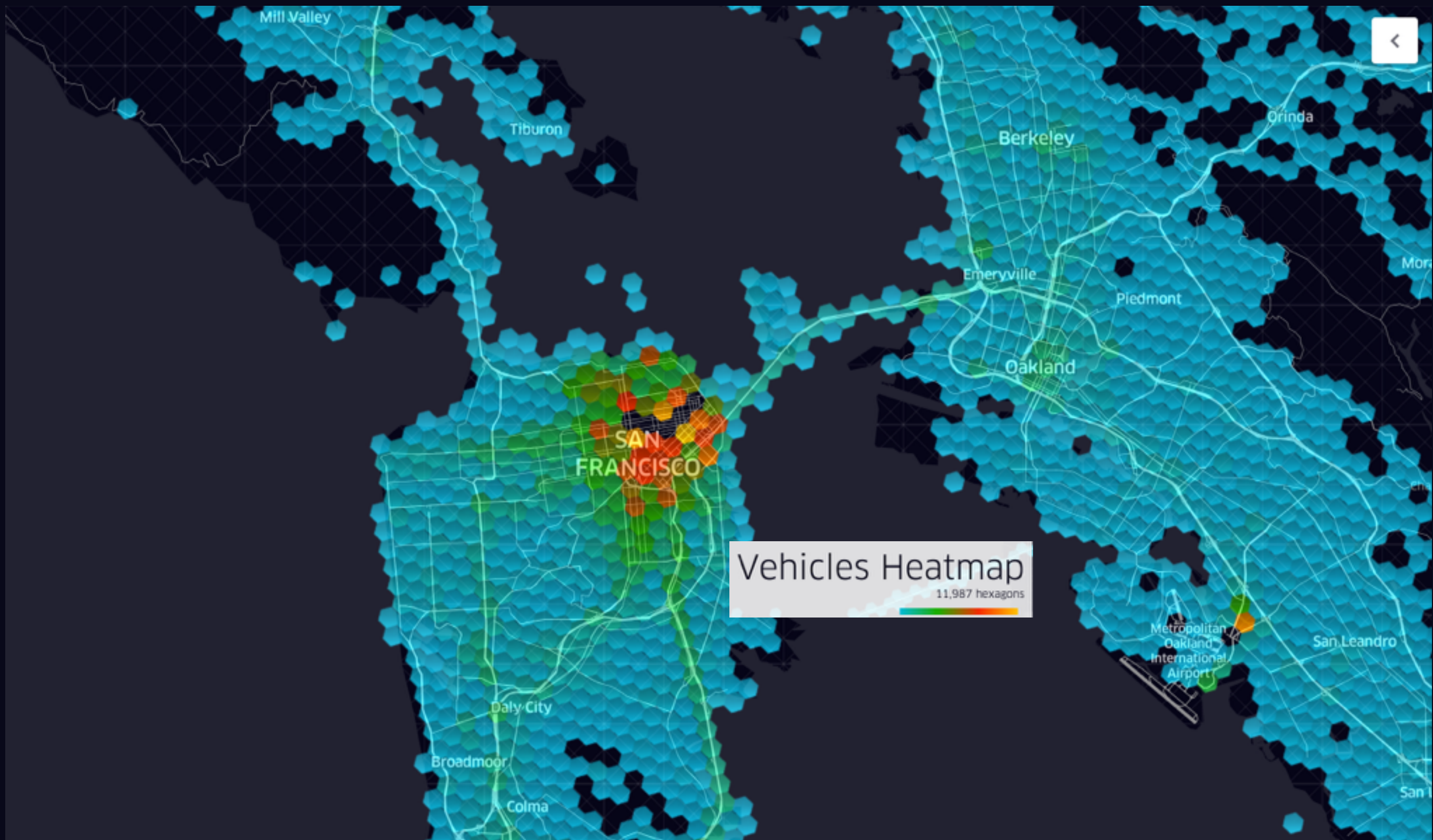
多维度

每个事件消息包含数十字段

细粒度数据

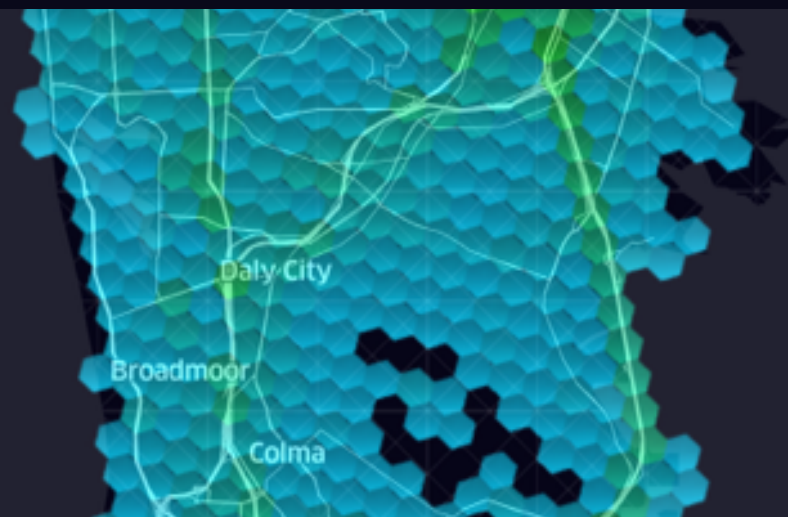


细粒度数据



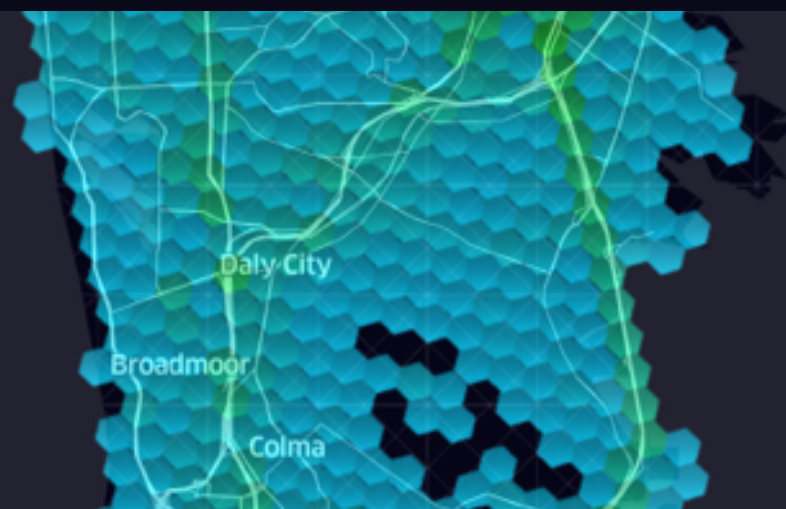
细粒度数据

每个城市多于 10,000 六边形



细粒度数据

7 种车型



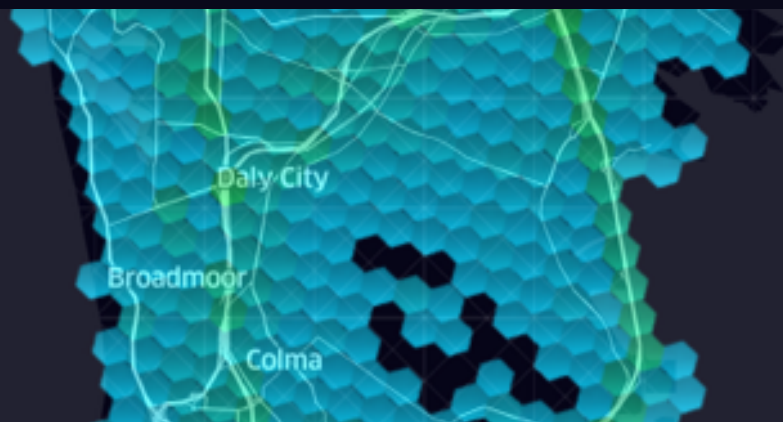
细粒度数据

一天1440分钟



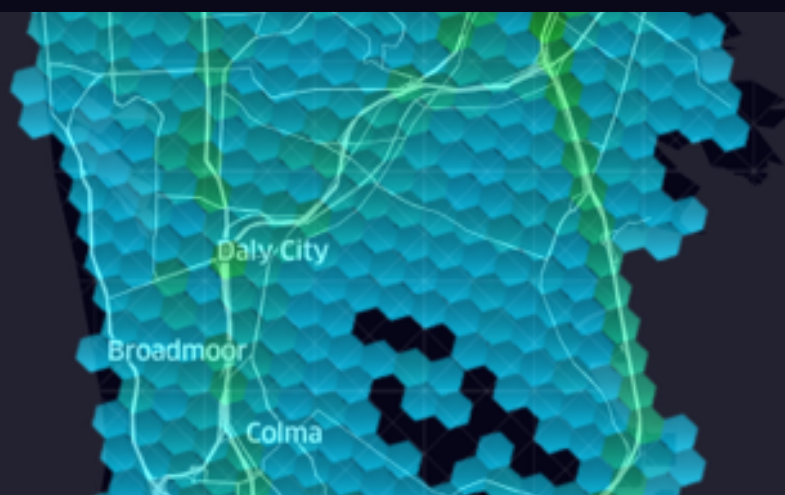
细粒度数据

13 种司机状态



细粒度数据

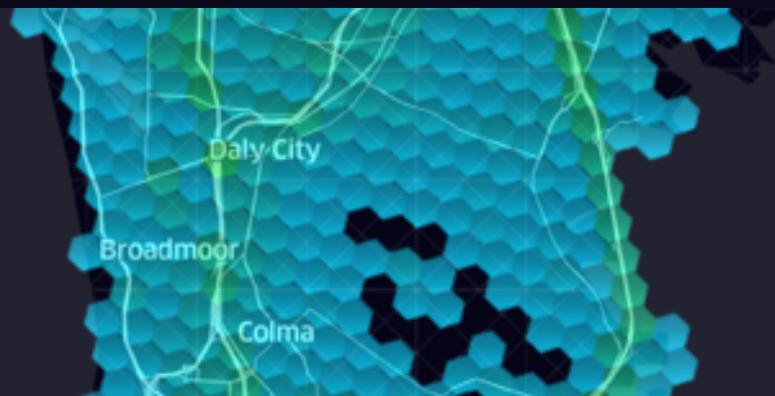
300 座城市



细粒度数据

一天数据量:

$300 \times 10,000 \times 7 \times 1440 \times 13 = 3930\text{亿}$ 可能组合



查询模式不定

任意维度的组合

多种聚合查询

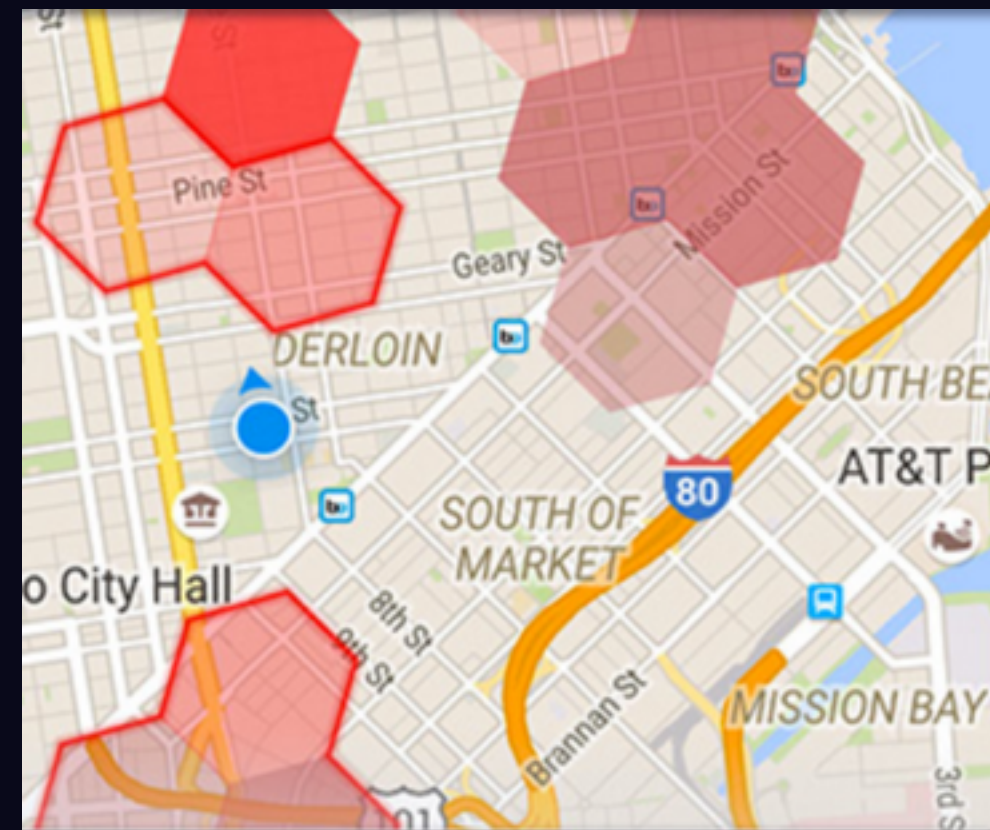
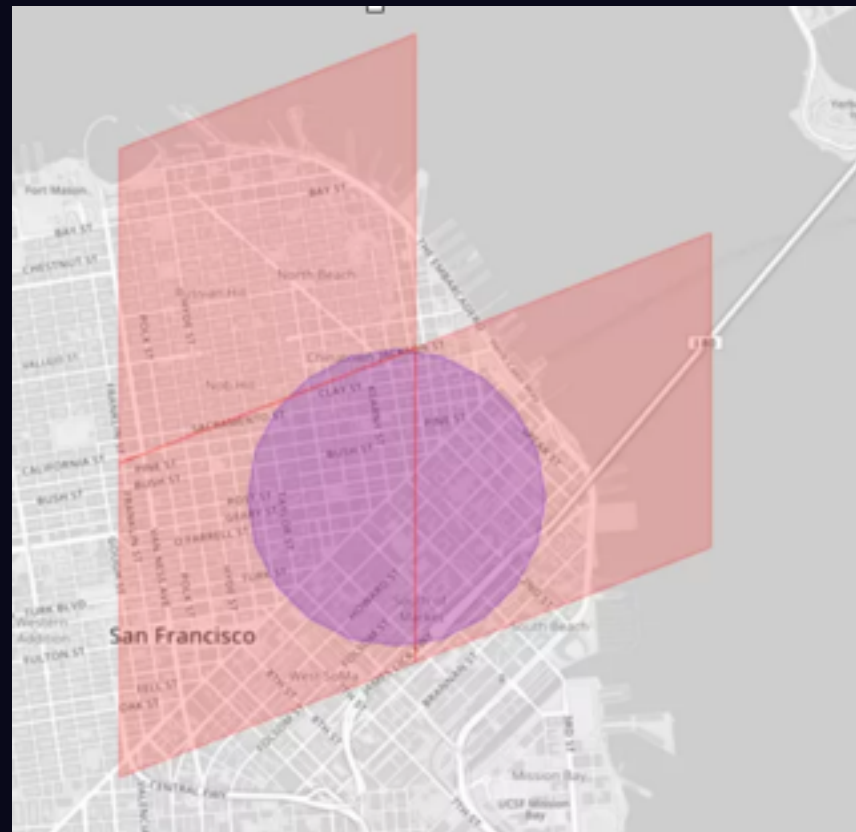
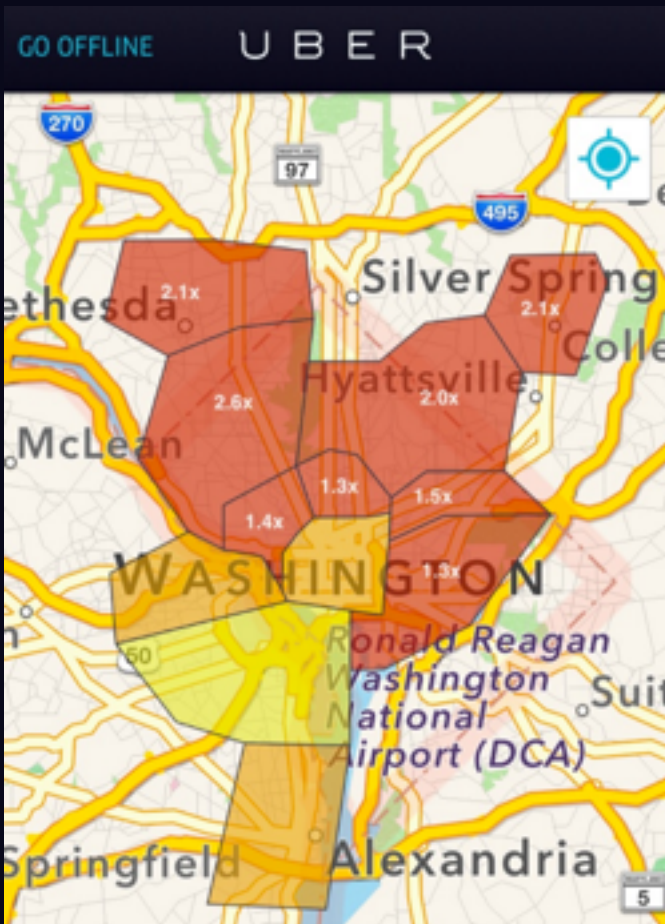
Heatmap

Top N

Histogram

count(), avg(), sum(), percent(), geo

多变的地理位置聚合查询



高流量

- 数十万条消息每秒，数百亿一天
- 每条消息包含数十字段

短时交货



关键: 把问题一般化

数据类型

- 多维时序数据

维度	值
state	driver_arrived
vehicle type	uber X
timestamp	13244323342
latitude	12.23
longitude	30.00

数据查询

- 基于单表时空数据的OLAP

```
SELECT <agg functions>, <dimensions>  
FROM <data_source>  
WHERE <boolean filter>  
GROUP BY <dimensions>  
HAVING <boolean filter>  
ORDER BY <sorting criterial>  
LIMIT <n>  
DO <post aggregation>
```

选择存储系统

最低要求

- 支持时序和地理空间的OLAP
- 支持大流量数据
- 支持秒级查询
- 支持原始数据查询

键值数据库

键值数据库

一键一值： 预算所有组合

维度	值
A	a
B	b

一键一值： 预算所有组合

- 布尔操作符: AND, OR, NOT

维度	值
A	a
B	b

一键一值：预算所有组合

维度	值
A	a
B	b

- 布尔操作符: AND, OR, NOT
 - A and (not B)
 - B and (not A)
 - A or B
 - not (A or B)

一键一值：预算所有组合

维度
A
B

- {A}
- {B}
- {A, B}
- {}

一键一值：预算所有组合

维度
A
B

- $\{A\}$
- $\{B\}$
- $\{A, B\}$
- $\{\}$

—> 计算幂集

键值数据库的局限

预算所有键值组合的时空复杂度: $O(2^n)$

关系数据库

~~关系数据库~~

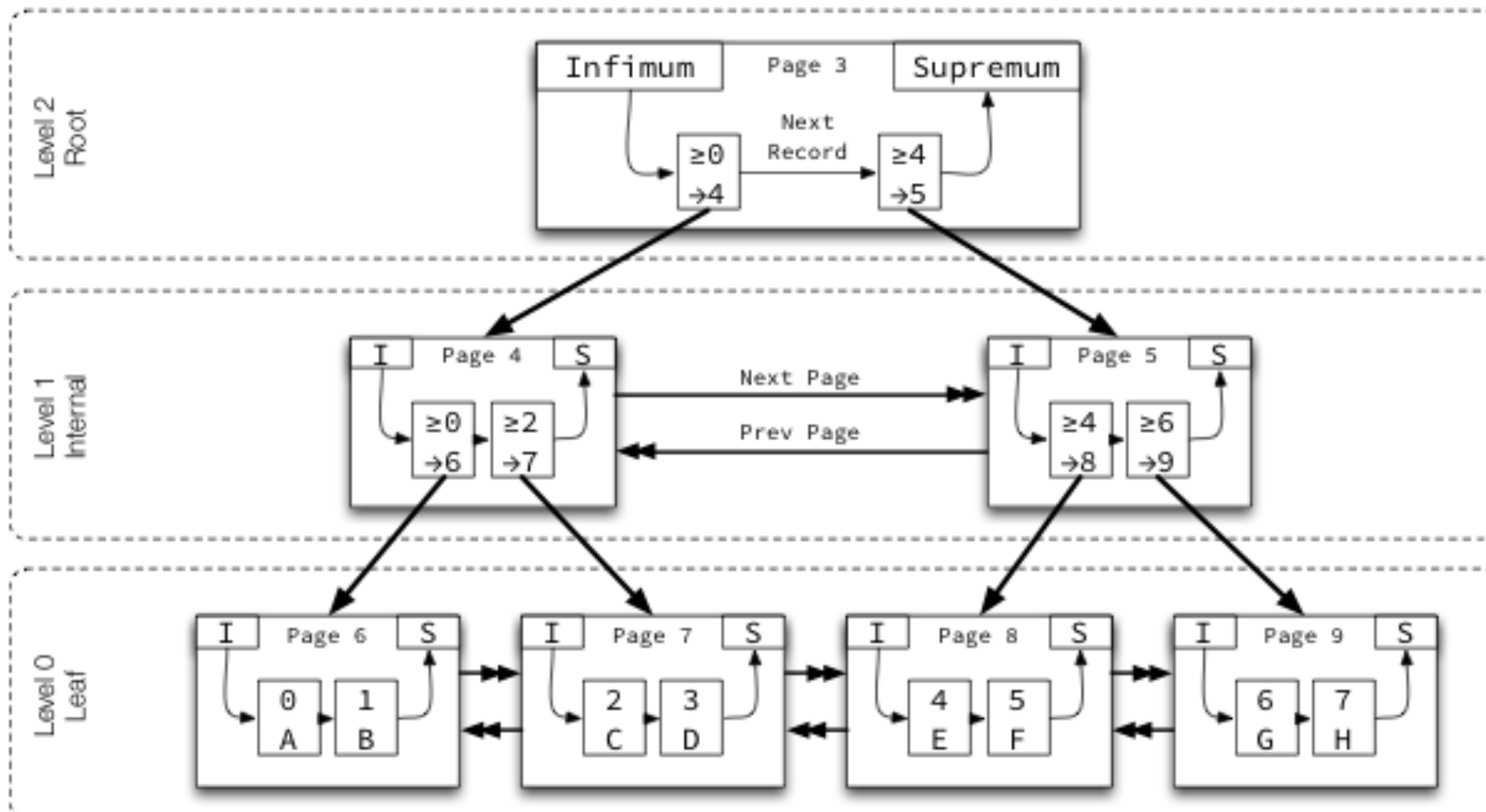
关系数据库的局限

- 不易管理多项索引

关系数据库的局限

- 扫描速度不够

B+Tree Structure



Levels are numbered starting from 0 at the leaf pages, incrementing up the tree.

Pages on each level are doubly-linked with previous and next pointers in ascending order by key.

Records within a page are singly-linked with a next pointer in ascending order by key.

Infimum represents a value lower than any key on the page, and is always the first record in the singly-linked list of records.

Supremum represents a value higher than any key on the page, and is always the last record in the singly-linked list of records.

Non-leaf pages contain the minimum key of the child page and the child page number, called a "node pointer".

但我的KV系统速度奇快

光快不行

一个城市里每个六边形里车的数目 => 18,000 次查询

平均延迟: 1ms

99.99百分位延迟: 2s

失败率: 0.001%

光快不行

一次查询延迟超过99.99百分位的概率: $(1 - 0.9999^{18000}) \times 100 = 83\%$

一次查询成功的概率: $(1 - 0.00001)^{18000} = 84\%$

系统必备功能

- 快速扫描
- 布尔查询
- 原始数据
- 各类聚合

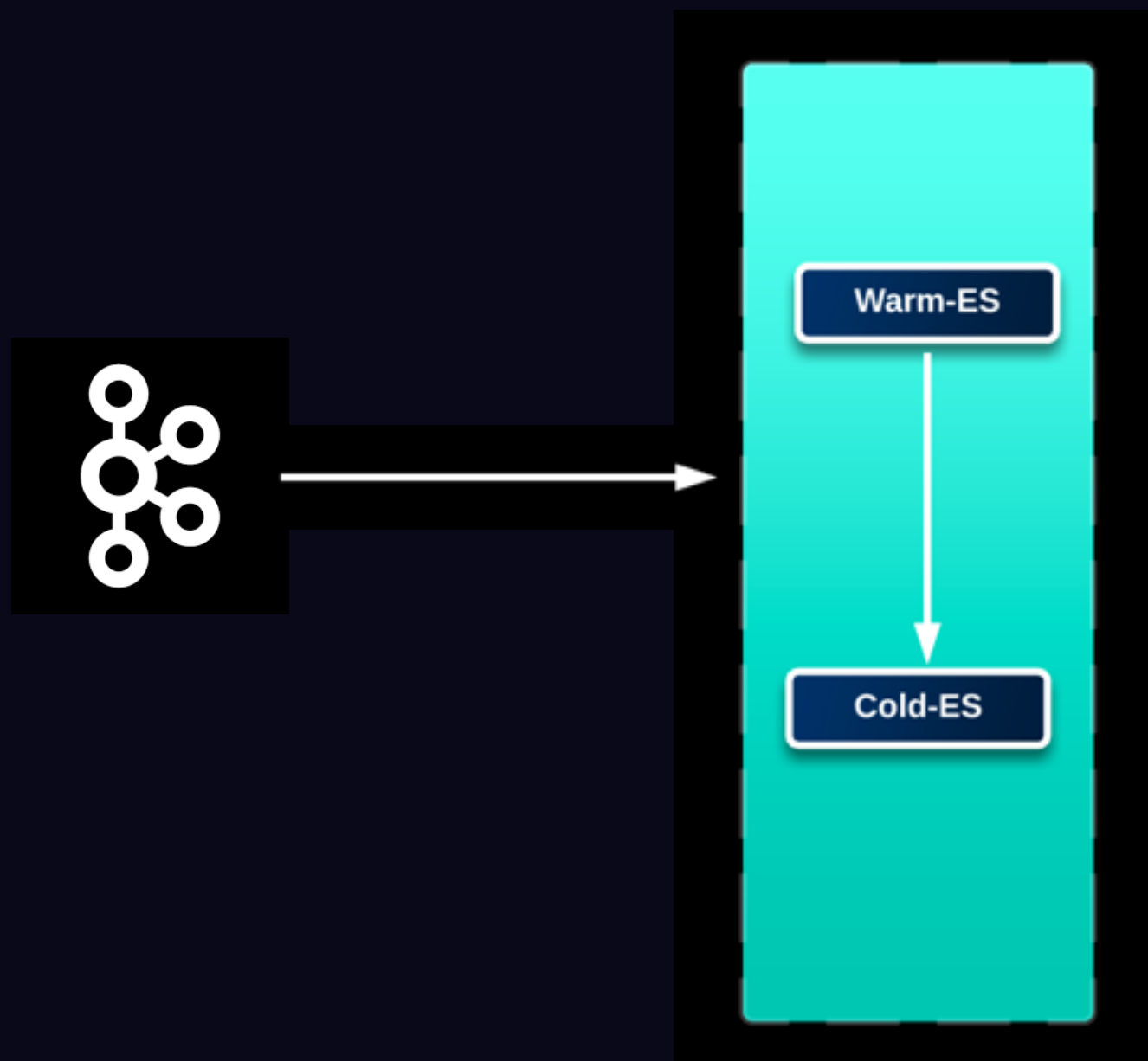
Elasticsearch

基于高效倒排索引的布尔查询

内建分布式查询

快速扫描， 灵活聚合

存储

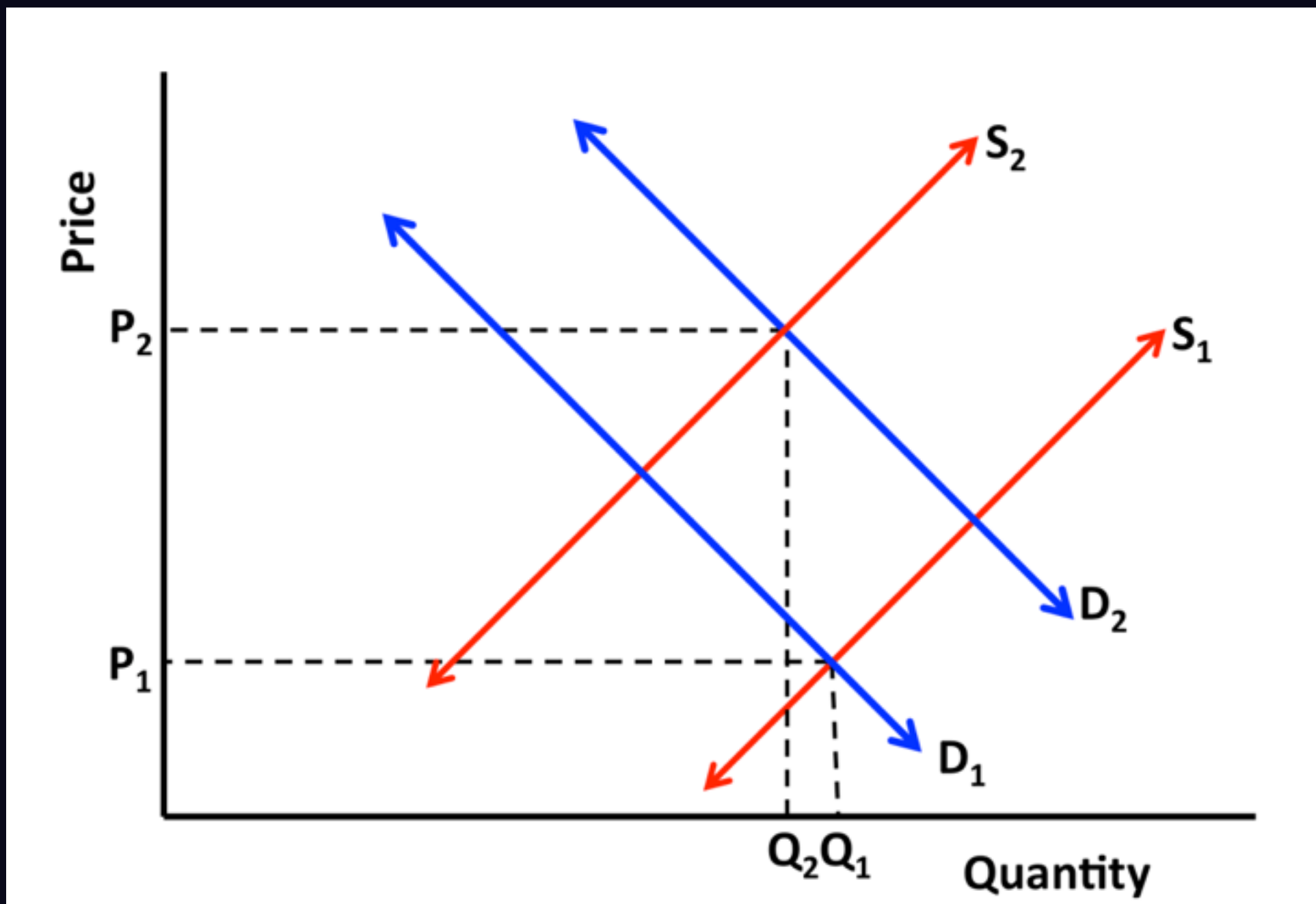


搞定没？

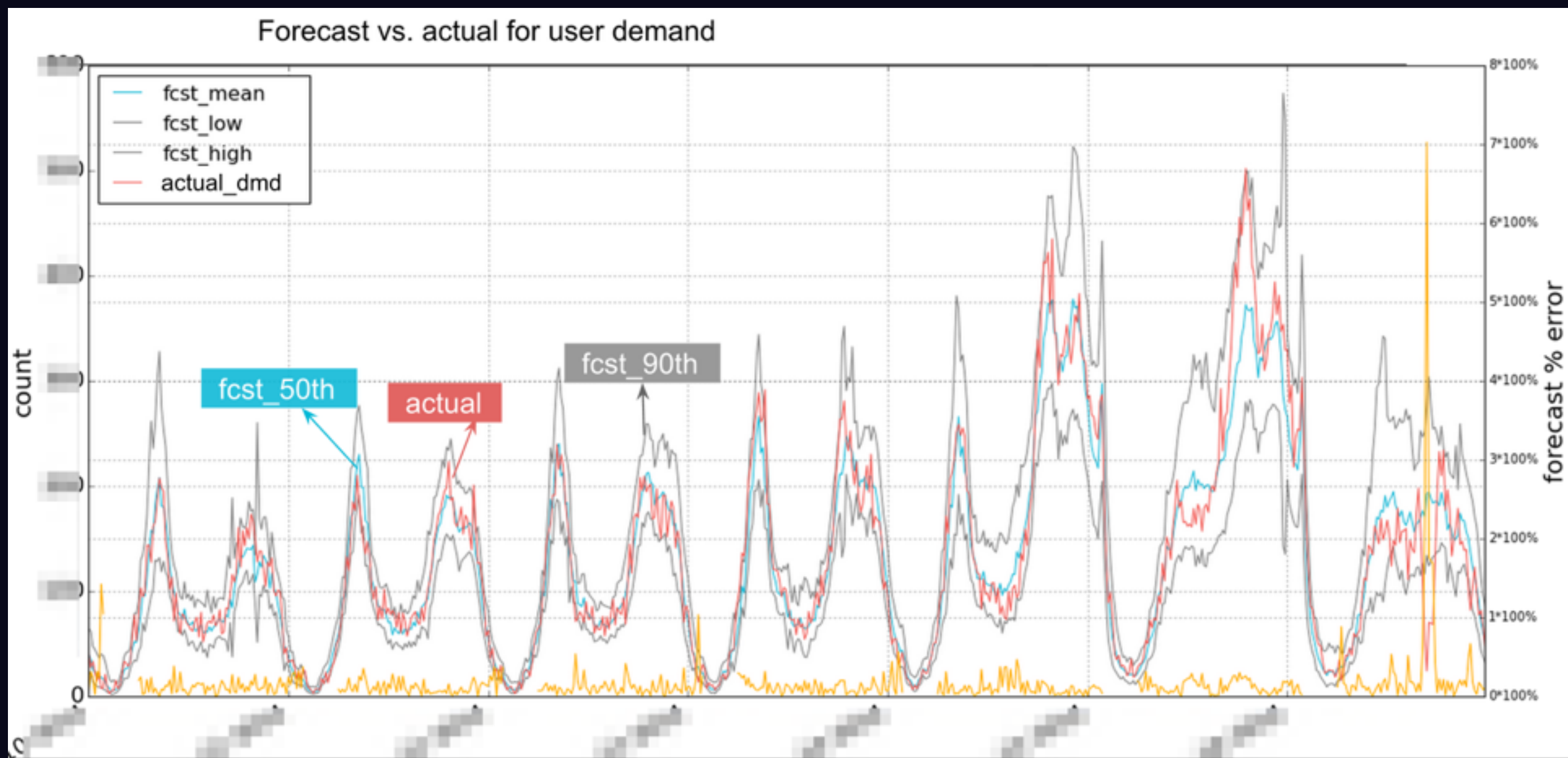
数据转换

e.g. (Lat, Long) -> (zipcode, hexagon)

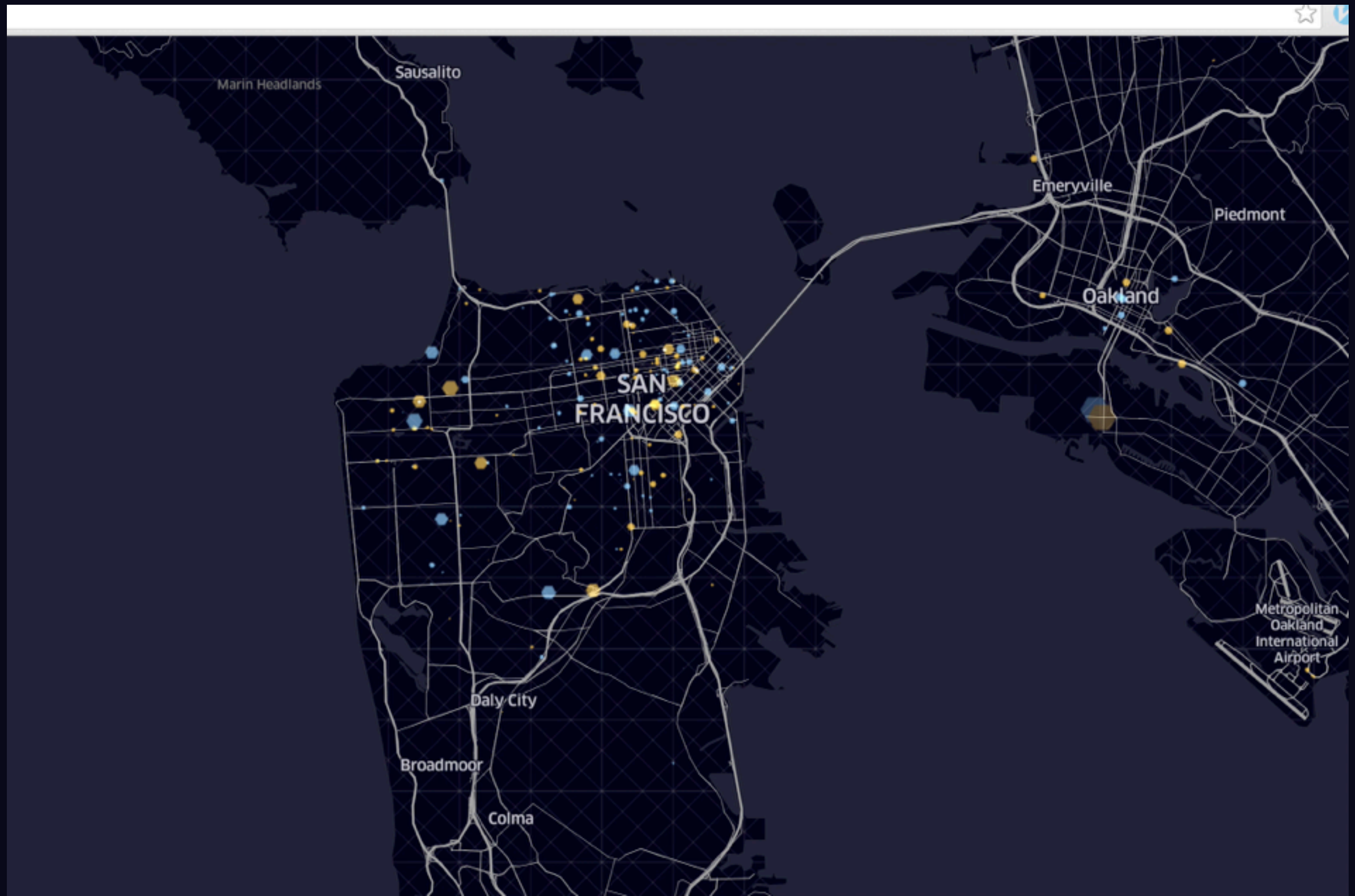
动态定价



趋势预测



供求分布



技术上说: Clustering & Pr(D, S, E)



新场景 —> 新需求

预处理

Joining Multiple Streams

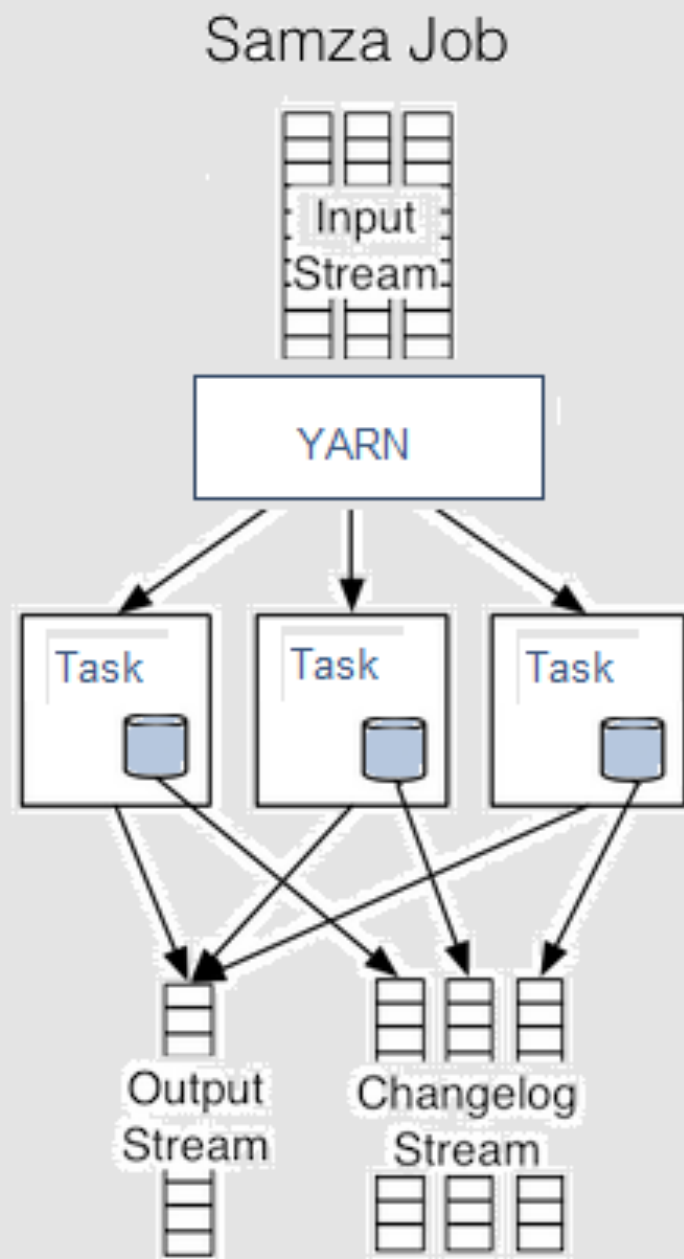
Sessionization

多级处理

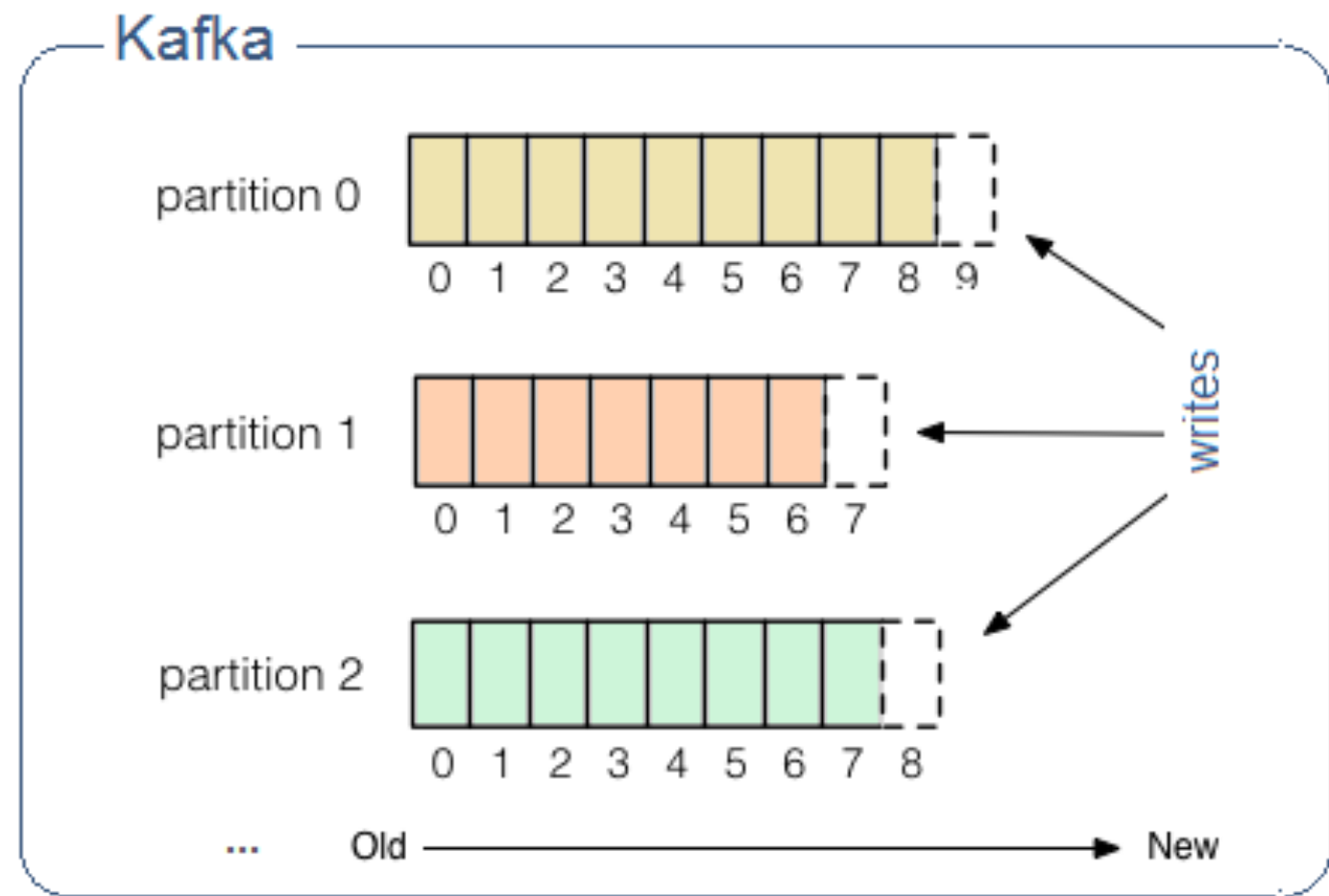
状态管理

Apache Samza

Why Apache Samza?



Partitioned Stream



samza

DAG on Kafka

与Kafka的一流整合

内置检查点

内置状态管理



处理

存储



Samza



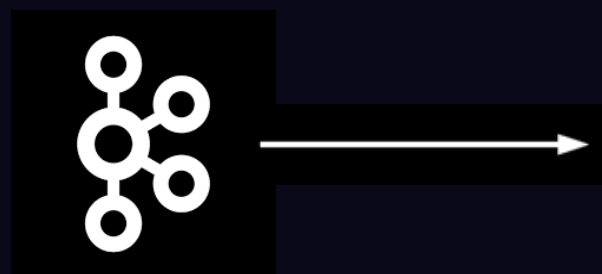
Warm-ES



Cold-ES

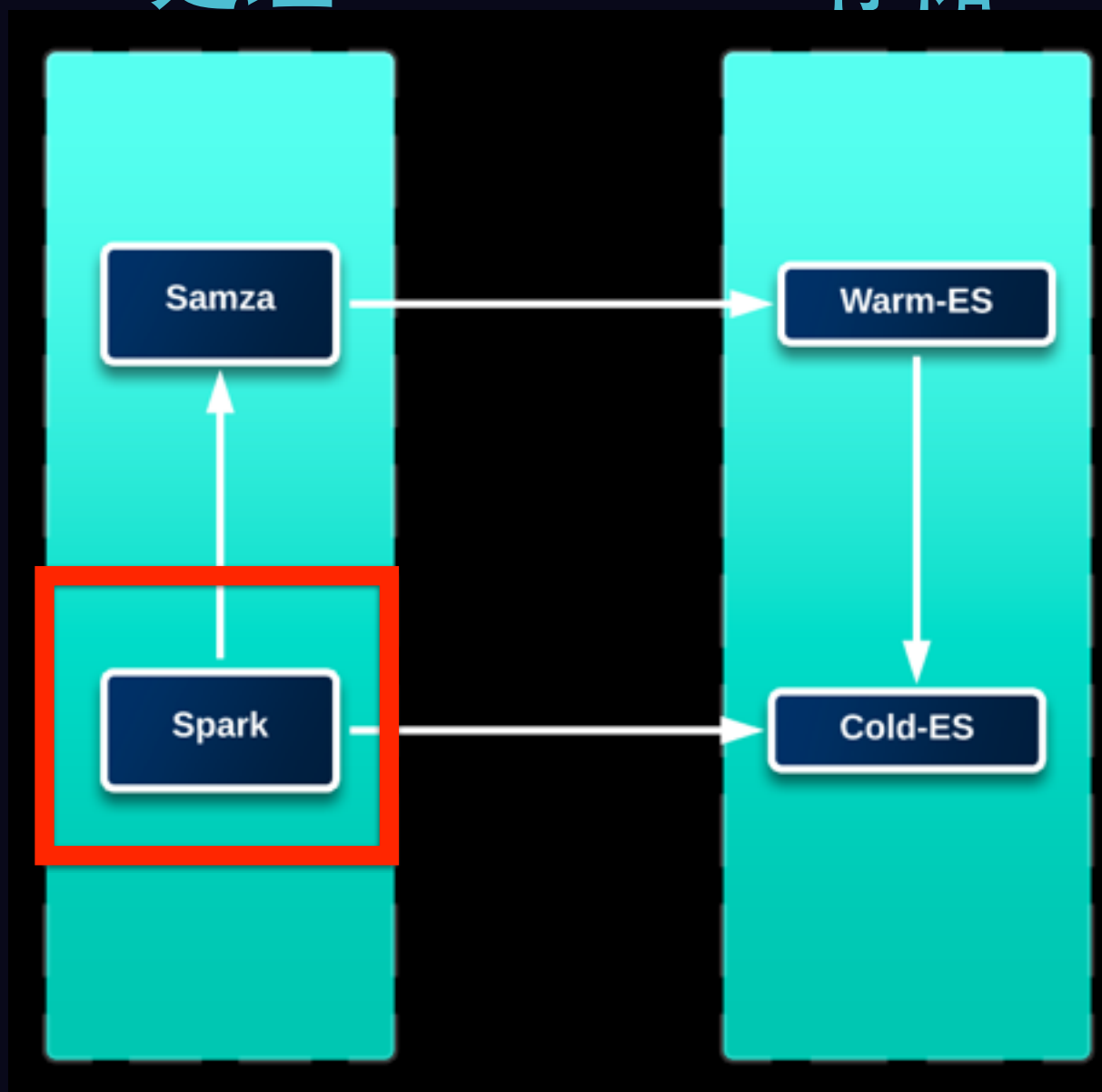
存储层当掉怎么办？

预处理耗时太久怎么办?



处理

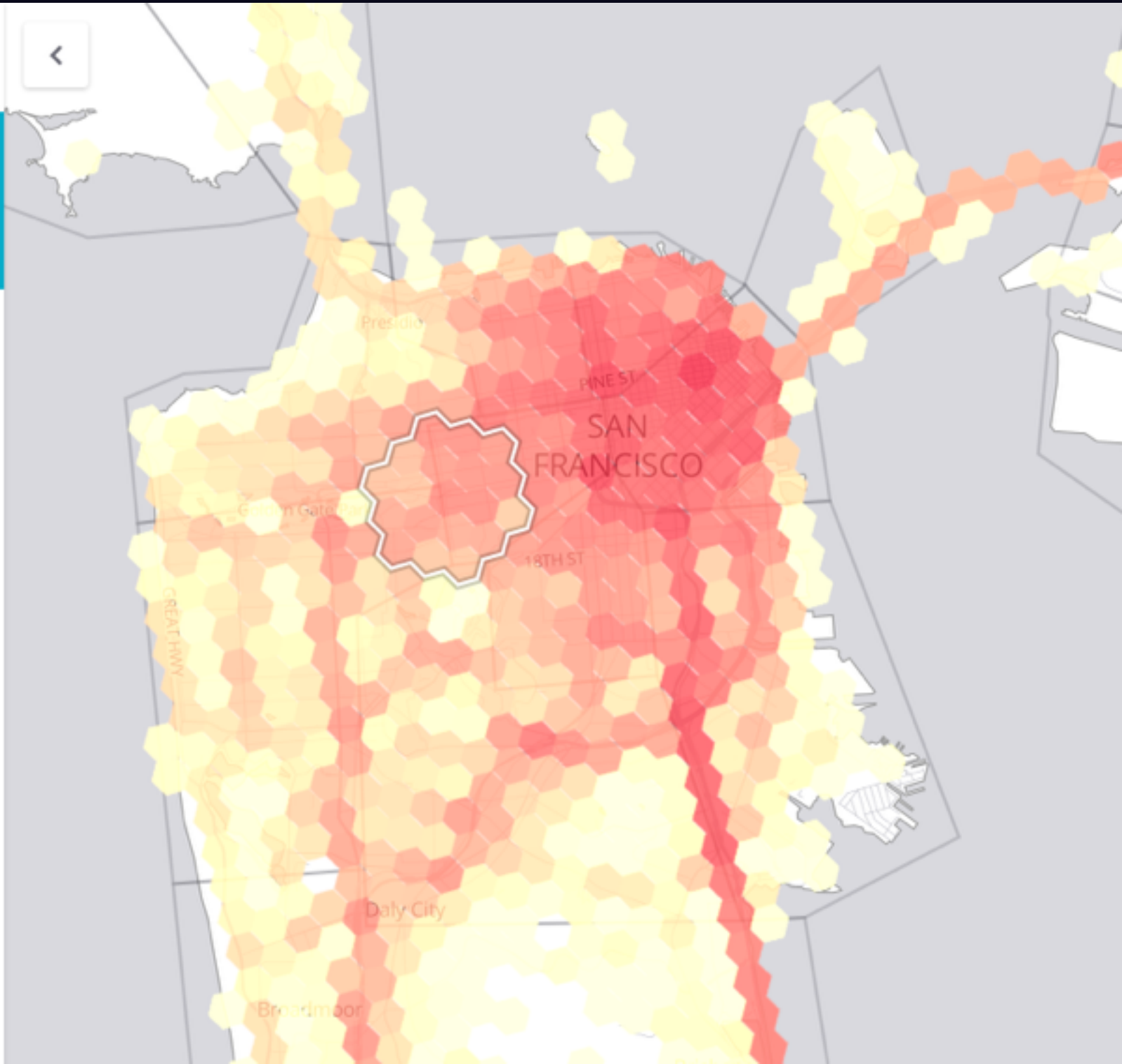
存储



终于搞定了？

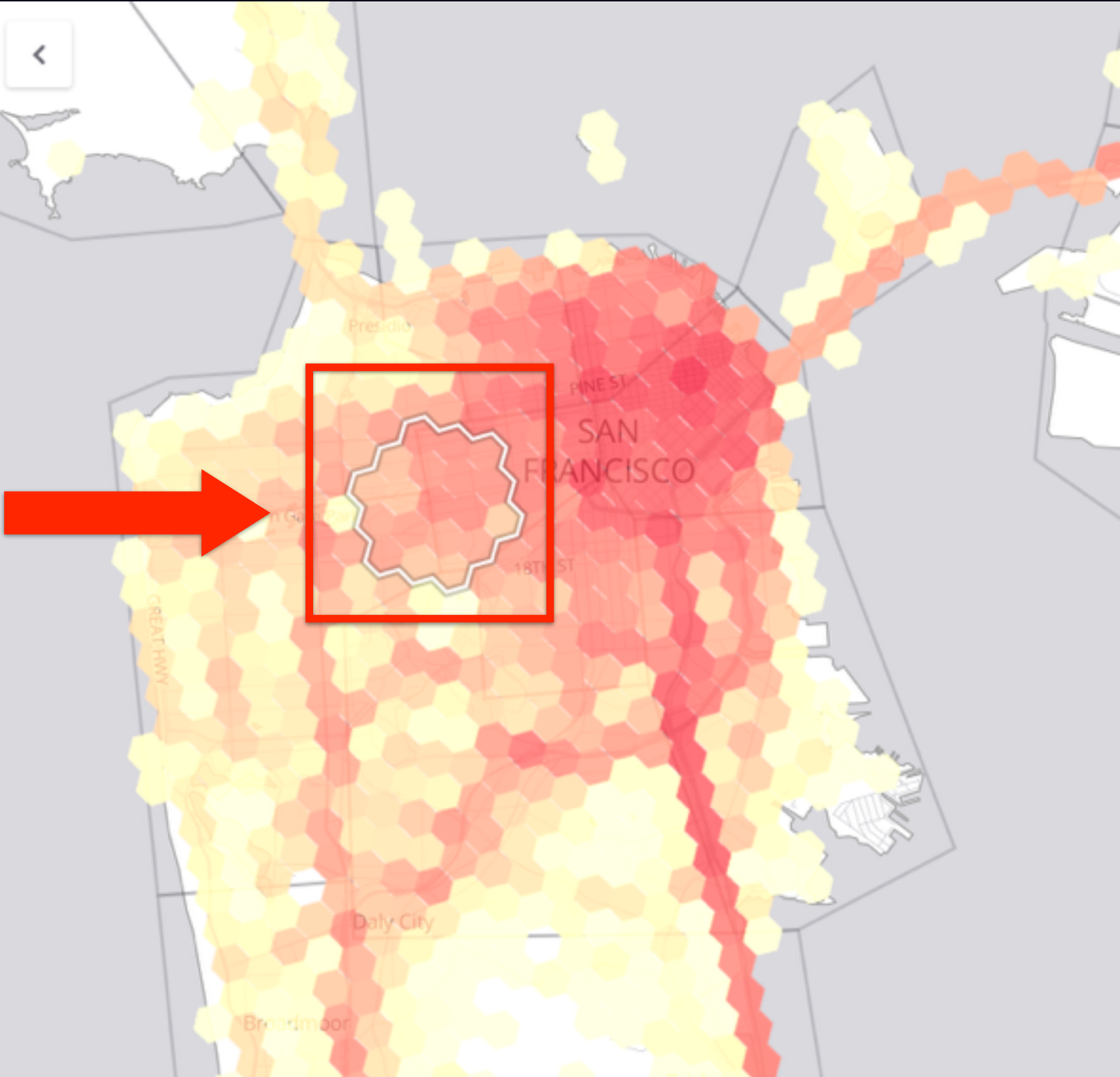
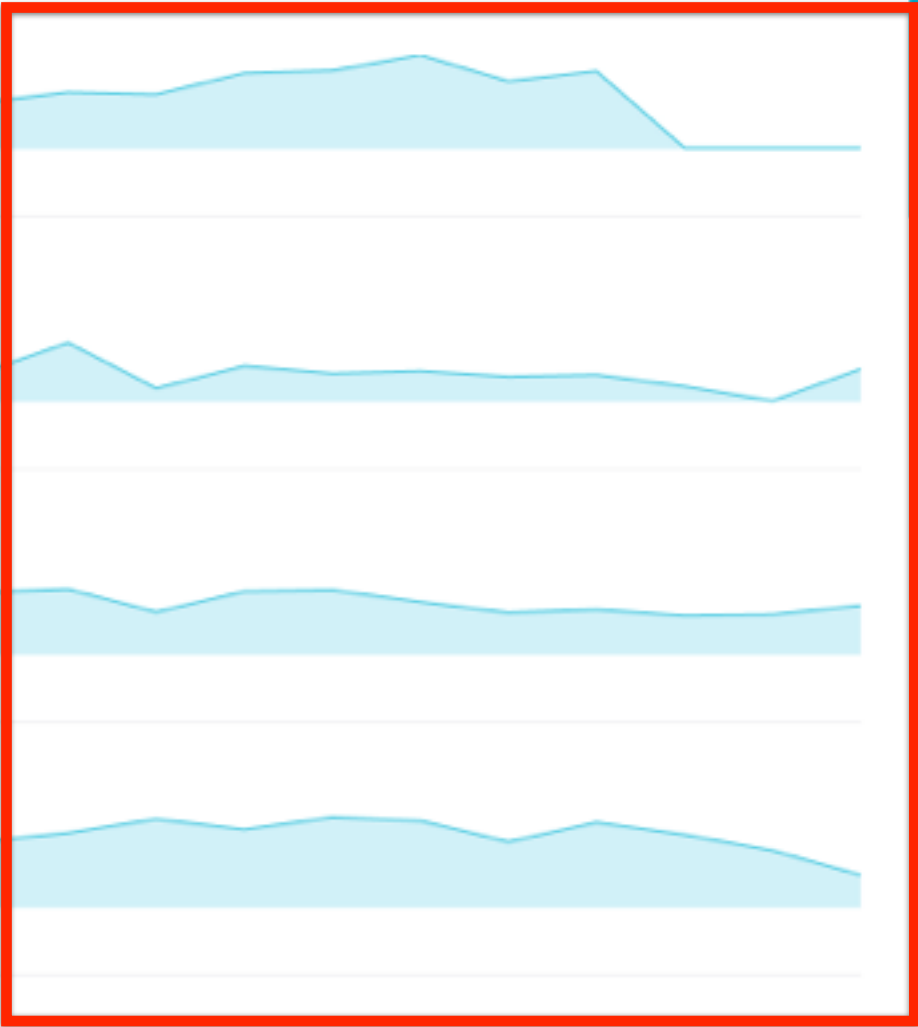
SELECTION RADIUS

0 1 2 3 4 5 6 7



SELECTION RADIUS

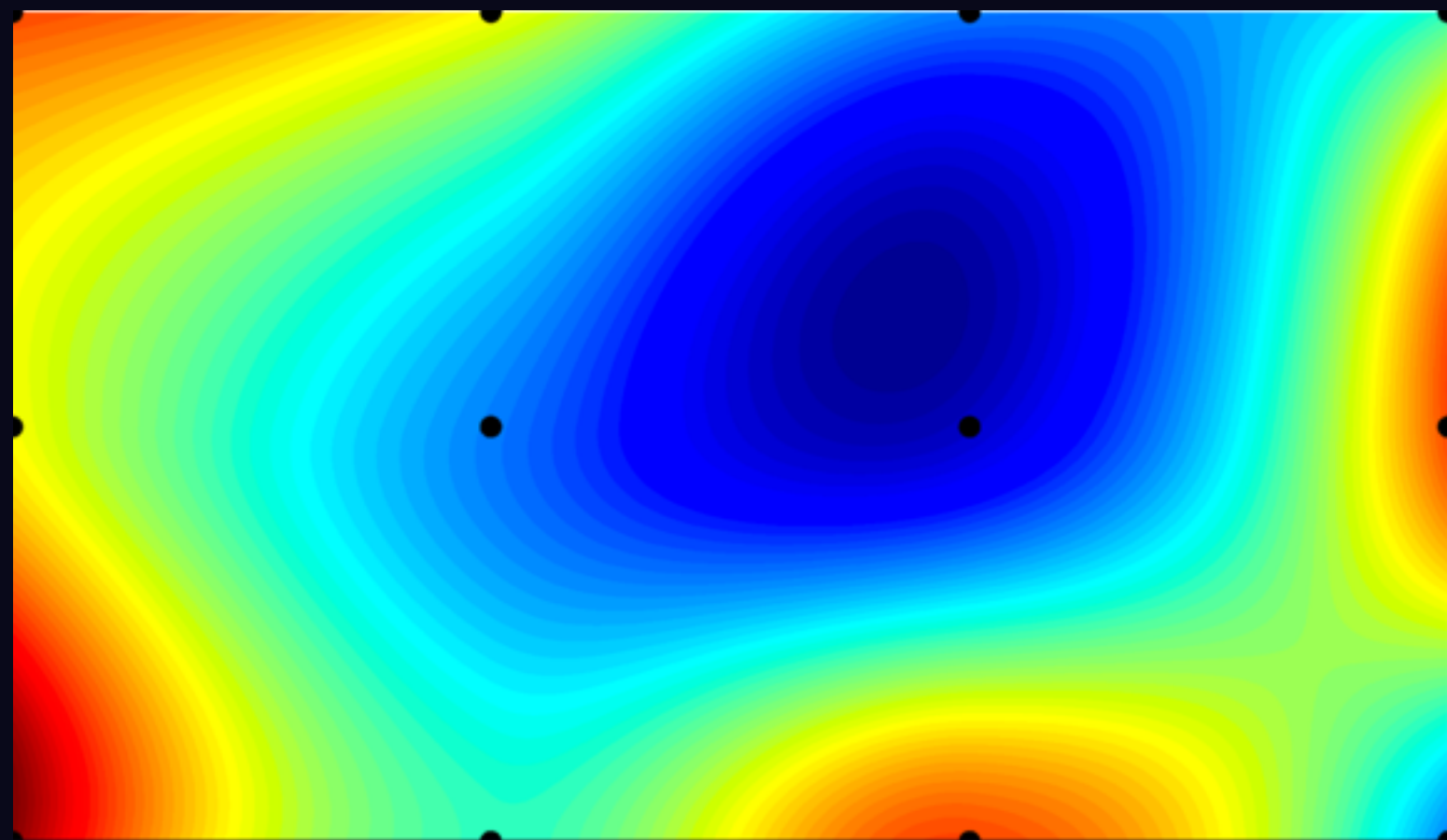
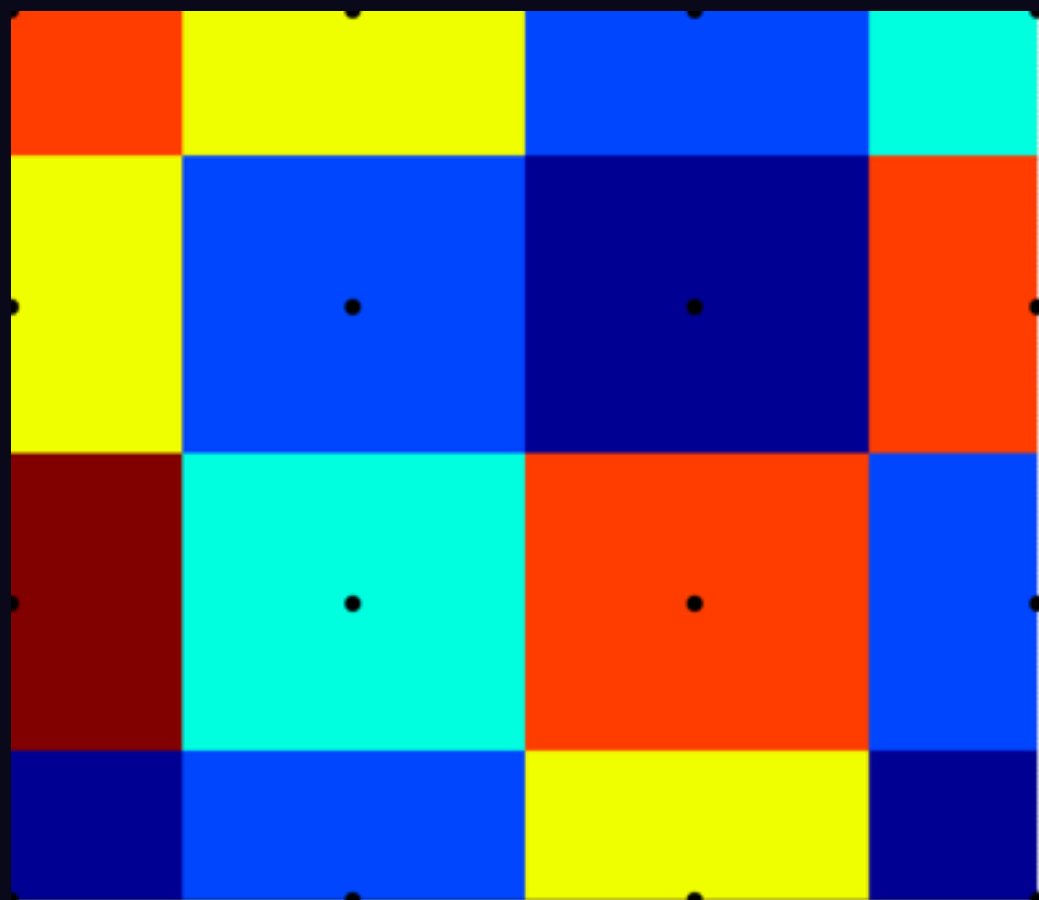
0 1 2 3 4 5 6 7



后期处理

查询结果转换和平滑处理

查询结果转换和平滑处理



计算规模

一个城市至少10,000六边形

计算规模

每个六边形331个邻居需要处理

计算规模

一次查询: $331 \times 10,000 = 310$ 万六边形

计算规模

99%-ile 处理时间: 70ms

简单架构

Scalability! But at what COST?

Frank McSherry
Unaffiliated

Michael Isard
Unaffiliated*

Derek G. Murray
Unaffiliated†

Abstract

We offer a new metric for big data platforms, COST, or the Configuration that Outperforms a Single Thread. The COST of a given platform for a given problem is the hardware configuration required before the platform outperforms a competent single-threaded implementation. COST weighs a system’s scalability against the overheads introduced by the system, and indicates the actual performance gains of the system, without rewarding systems that bring substantial but parallelizable overheads.

We survey measurements of data-parallel systems recently reported in SOSP and OSDI, and find that many systems have either a surprisingly large COST, often hundreds of cores, or simply underperform one thread for all of their reported configurations.

1 Introduction

“You can have a second computer once you’ve shown you know how to use the first one.”

—Paul Barham

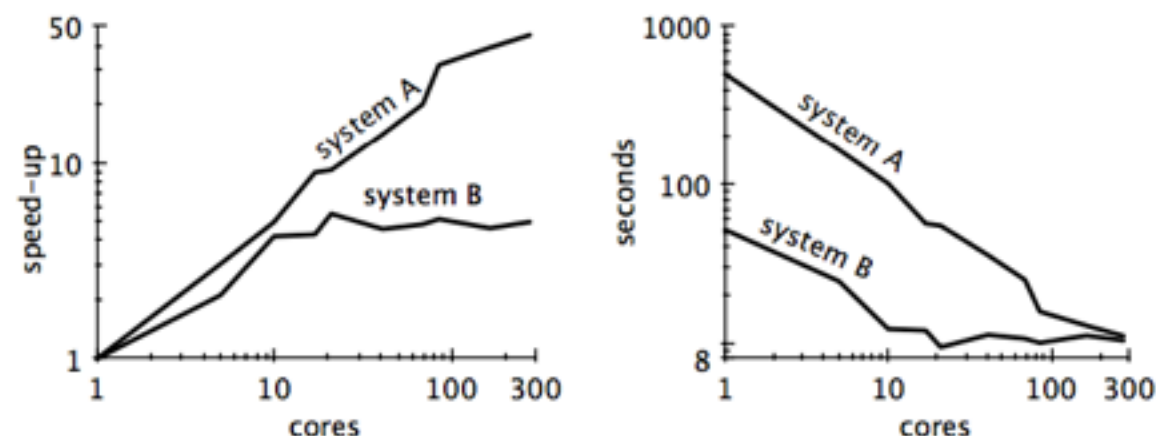


Figure 1: Scaling and performance measurements for a data-parallel algorithm, before (system A) and after (system B) a simple performance optimization. The unoptimized implementation “scales” far better, despite (or rather, because of) its poor performance.

While this may appear to be a contrived example, we will argue that many published big data systems more closely resemble system A than they resemble system B.

1.1 Methodology

In this paper we take several recent graph processing papers from the systems literature and compare their re-

Connectivity	cores	twitter_rv	uk_2007_05
Spark	128	1784s	8000s+
Giraph	128	200s	8000s+
GraphLab	128	242s	714s
GraphX	128	251s	800s

Connectivity	cores	twitter_rv	uk_2007_05
Spark	128	1784s	8000s+
Giraph	128	200s	8000s+
GraphLab	128	242s	714s
GraphX	128	251s	800s
Laptop	1	153s	417s

Connectivity	cores	twitter_rv	uk_2007_05
Spark	128	1784s	8000s+
Giraph	128	200s	8000s+
GraphLab	128	242s	714s
GraphX	128	251s	800s
Laptop	1	153s 15s	417s 30s

“You can have a second computer once you’ve shown you know how to use the first one.”

- Paul Graham

后期处理

- 每个处理单元都是纯函数
- 通过组合算子组合处理单元

后期处理

- 高度并行化的执行
- 流水线

务实考量

数据发现

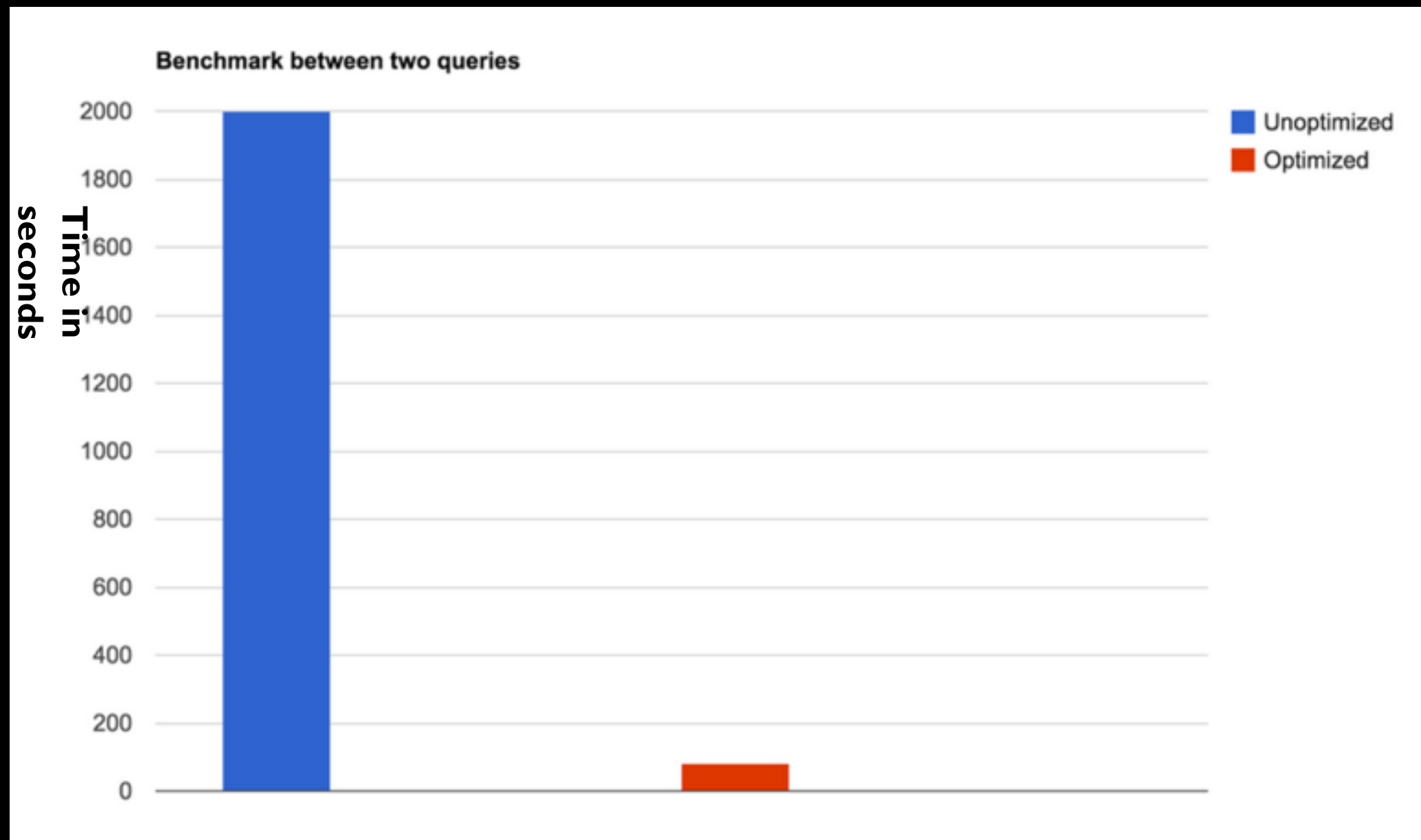
Elasticsearch 查询语句过度复杂

```
SELECT timeseries(7d)
FROM driverAcceptanceRate
WHERE geo_dist(10, [37, 22])
AND time IN (2015-02-04, 2015-03-06)
AND msg.driverId = 1
```

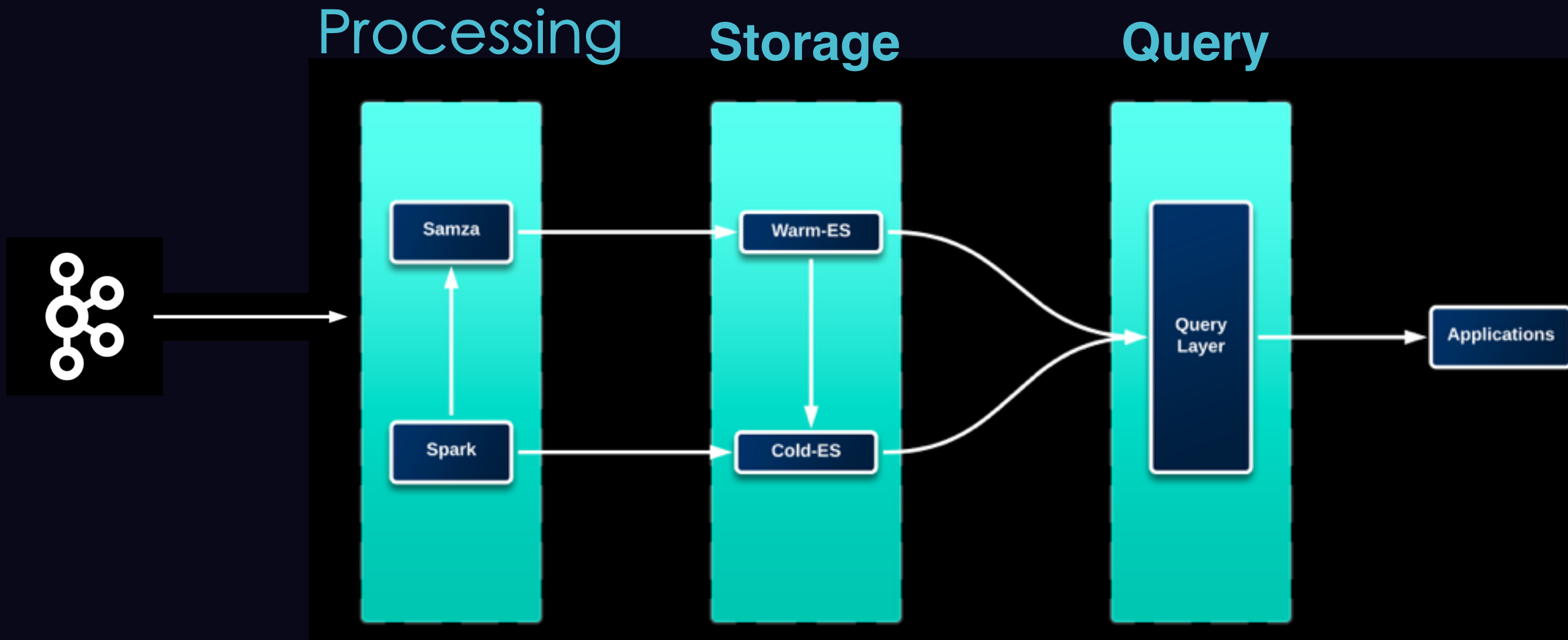
```
}
  }
    }
```

Elasticsearch 查询可以优化

- 流水线
- 查询验证
- 查询限速



Elasticsearch 也许会被替换



还有一件事

数据流里总有不同模式

总有快速探索发现模式的需要

多少司机在5分钟内连续取消请求5次以上

哪些乘客半小时内相距超过100公里的地方叫车？

Complex Event Processing

```
FROM driver_canceled#window.time(10 min)
SELECT clientUUID, count(clientUUID) as cancelCount
GROUP BY clientUUID HAVING cancelCount > 10
INSERT INTO hipchat(room);
```



Driver cancellation alert room

This is the room topic. Double click to change it.

CAG Bot 2 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/702e>) in city 5

CAG Bot 2 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/3b19>) in city 5

CAG Bot 2 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/d6e0>) in city 1

CAG Bot 2 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/8b58>) in city 1

CAG Bot 2 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/b191>) in city 5

CAG Bot 2 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/bc5c>) in city 5

CAG Bot 2 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/e487>) in city 5

Xiaoman Dong joined the room

CAG Bot 3 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/8d56>) in city 3

CAG Bot 2 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/461e>) in city 3

CAG Bot 4 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/9fdf7>) in city 5

CAG Bot 2 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/de39>) in city 1

CAG Bot 2 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/102a>) in city 3

CAG Bot 2 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/8c18>) in city 21

CAG Bot 2 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/8d1d>) in city 5

CAG Bot 3 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/627e>) in city 5

CAG Bot 2 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/cb9e>) in city 1

CAG Bot 2 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/6671>) in city 5

CAG Bot 4 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/cb1a>) in city 5

CAG Bot 2 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/be7d>) in city 1

CAG Bot 6 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/627e>) in city 5

CAG Bot 10 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/627e>) in city 5

CAG Bot 2 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/33d2>) in city 1

CAG Bot 10 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/627e>) in city 5

CAG Bot 12 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/627e>) in city 5

CAG Bot 3 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/cb9e>) in city 1

CAG Bot 3 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/540a>) in city 1

CAG Bot 2 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/e3c7>) in city 3

CAG Bot 2 driver cancellations detected within the last 10 minutes on the same client (<https://toolshed.uberinternal.com/v2/clients/e5a4>) in city 5

简单实现

谢谢！

U B E R

