

OneStepOffer 系统设计

第三讲

设计Uber

Design Uber- Scenario

- 需要设计哪些功能？设计到什么地步？

第一阶段

- Driver report locations 司机报告自己的位置——heartbeat 模式
- Rider request Uber, match a driver with rider 乘客叫车，匹配一辆车

Design Uber- Scenario

第二阶段

- Driver deny/accept a request 司机取消/启动 接单
- Driver cancel a match request 司机取消订单
- Ride cancel a request 乘客取消请求
- Driver pick up a ride/ start a trip 司机接人
- Driver drop off a rider/end a trip 司机送到人

Design Uber- Scenario

QPS

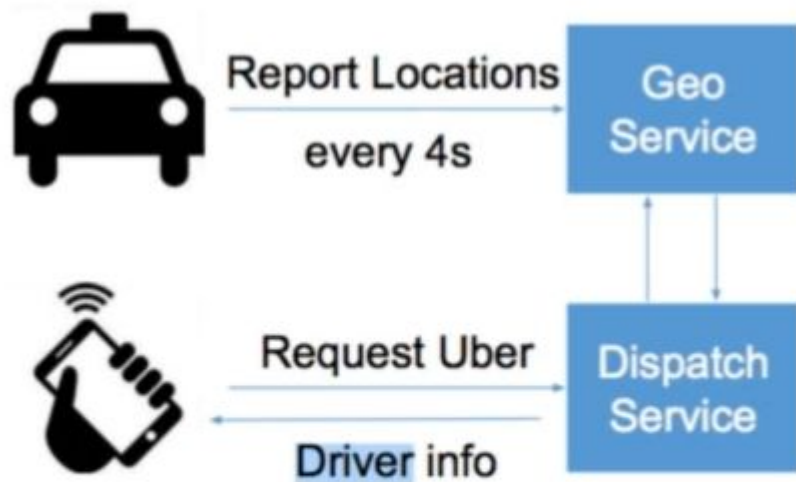
假设20w司机同时在线：

- Driver QPS = $200k / 4 = 50k$ ，每次汇报200k个请求，每4秒汇报一次；这是主体部分
- Peek Driver QPS = $50k \times 3 = 150k$
- Rider QPS可以忽略：不用随时汇报位置，一定远小于Driver QPS

Design Uber- Service

主要需求：

- 记录车的位置：GeoService
- 匹配打车请求：DispatchService



这张图里漏了什么？

Design Uber- Service

缺陷:

- 司机需要知道乘客位置
- 司机需要从GeoService拿用户定位
- GeoService负担过重

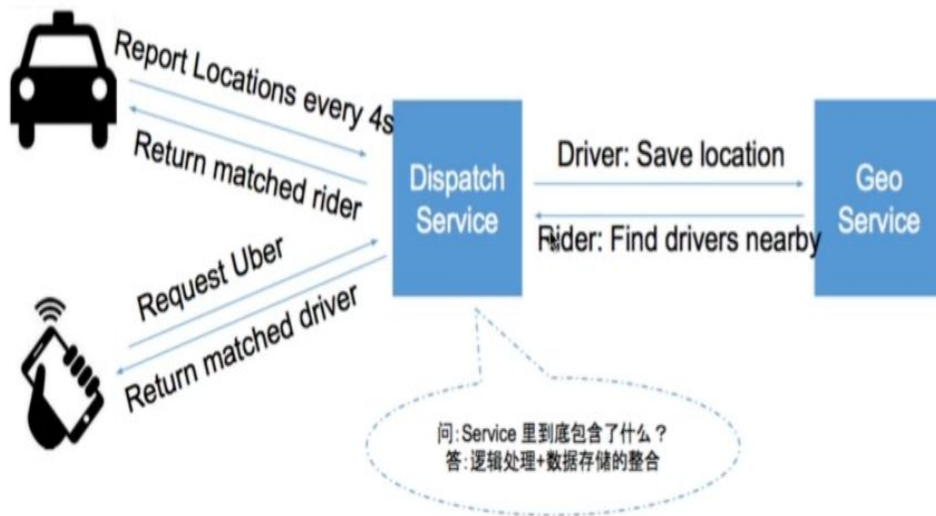


这张图里漏了什么？

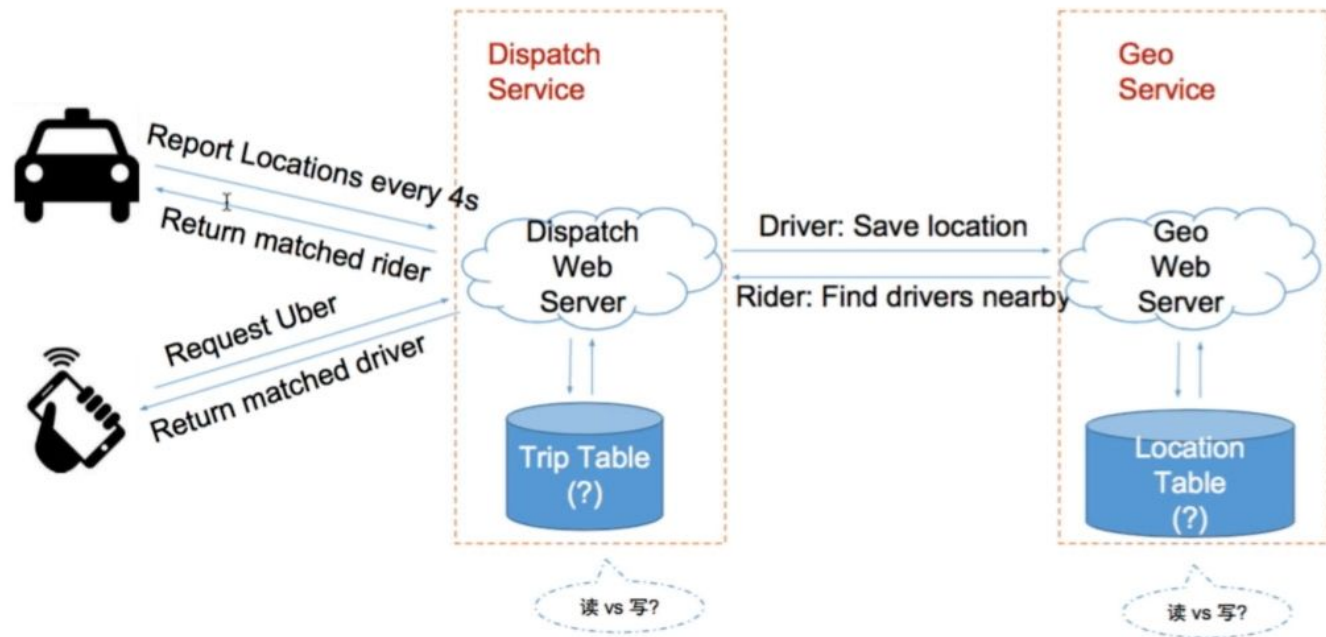
Design Uber- Service

改良:

- 司机向dispatch service发送位置信息, 同时返回匹配的乘客信息
- 乘客向dispatch service发送打车请求, 同时返回匹配司机



Design Uber- Storage



Design Uber- Storage

设计数据库表要考虑：

- Location Table——读多还是写多？—— 写多
- Trip Table—— 读多还是写多？—— 读多（司机每4秒request时会去读一下有没有匹配的）

Design Uber- Storage

设计数据库表

```
class Trip {  
    public Integer tripId;  
    public Integer driverId, riderId;  
    public Double Latitude, Longitude;  
    public Integer status;  
    public Datetime createdAt;  
}  
  
class Location {  
    public Integer driverId;  
    public Double Latitude, Longitude;  
    public Datetime updatedAt;  
}
```

Design Uber- Storage

问题——如何通过Location Table查询某个乘客周围5公里以内的司机？

```
SELECT * FROM Location WHERE lat < myLat + delta  
AND lat > myLat - delta  
AND lng < myLng + delta  
AND lng > myLng - delta
```

这是对location table的全局扫描

Design Uber- Storage

问题——如何通过Location Table查询某个乘客周围5公里以内的司机？

Geohash

- 将经纬度转化为字符串
- Base32: 0-9, a-z 去掉 (a, i, l, o)——为什么用 base32 ? 因为刚好 25 可以用 5 位二进制表示
- 特性: 公共前缀越长, 两个点越接近
- Example: (-30.043800, -51.140220) → 6feth68y4tb0

Design Uber- Storage

Geohash主要思想

- 把地图分成格子；
 - 每个格子由0-9, a-z 去掉 (a, i, l, o) 的32个字符组成
 - 在一个盒子内部，再划分32份；然后继续分分分
-
- 两个点之间到底有多远呢？主要参考表：如果两个公共前缀有geohash length个，那么就是对应的

Design Uber- Storage

查询Google半径2公里内的车辆

- 找到精度误差 > 2公里的最长长度

geohash length	lat bits	lng bits	lat error	lng error	km error
1	2	3	± 23	± 23	± 2500
2	5	5	± 2.8	± 5.6	± 630
3	7	8	± 0.70	± 0.7	± 78
4	10	10	± 0.087	± 0.18	± 20
5	12	13	± 0.022	± 0.022	± 2.4
6	15	15	± 0.0027	± 0.0055	± 0.61
7	17	18	± 0.00068	± 0.00068	± 0.076
8	20	20	± 0.000085	± 0.00017	± 0.019



- Google HQ: 9q9hvu7wbq2s
- 找到位置以9q9hv以开头的所有车辆



怎样在数据库中实现该功能？

Design Uber- Storage

如何找到位置开头以9q9hv开头的车辆？数据库怎么存？

SQL 数据库

- 首先需要对 geohash 建索引

```
CREATE INDEX on geohash;
```

- 使用 Like Query:

```
SELECT * FROM location WHERE geohash LIKE "9q9hv%";
```

NoSQL – Cassandra

- 将 geohash 设为 column key
- 使用 range query (9q9hv0, 9q9hvz)

Design Uber- Storage

用户打车时如何查询车辆?

(lat, lng) → geohash → [driver1, driver2, ...]

- 先查6位的 geohash 找0.6公里以内的
- 如果没有 再查5位的 geohash 找2.4公里以内的
- 如果没有 再查4位的 geohash 找20公里以内的

Design Uber- Solution

1. 乘客发出打车请求，服务器创建一次Trip
 1. 将 trip_id 返回给用户
 2. 乘客每隔几秒询问一次服务器是否匹配成功
2. 服务器找到匹配的司机，写入Trip，状态为等待司机回应
 1. 同时修改 Driver Table 中的司机状态为不可用，并存入对应的 trip_id
3. 司机汇报自己的位置
 1. 顺便在 Driver Table 中发现有分配给自己的 trip_id
 2. 去 Trip Table 查询对应的 Trip，返回给司机
4. 司机接受打车请求
 1. 修改 Driver Table, Trip 中的状态信息
 2. 乘客发现自己匹配成功，获得司机信息
5. 司机拒绝打车请求
 1. 修改 Driver Table, Trip 中的状态信息，标记该司机已经拒绝了该trip
 2. 重新匹配一个司机，重复第2步