



How to design Uber

CS103 系统设计

Producer: 冯沁原

1.01



课程咨询



Copyright Policy

All content included on the Site or third-party platforms as part of the class, such as text, graphics, logos, button icons, images, audio clips, video clips, live streams, digital downloads, data compilations, and software, is the property of BitTiger or its content suppliers and protected by copyright laws.

Any attempt to redistribute or resell BitTiger content will result in the appropriate legal action being taken.

We thank you in advance for respecting our copyrighted content.

For more info see <https://www.bittiger.io/termsofuse> and <https://www.bittiger.io/termsofservice>



版权声明

所有太阁官方网站以及在第三方平台课程中所产生的课程内容，如文本，图形，徽标，按钮图标，图像，音频剪辑，视频剪辑，直播流，数字下载，数据编辑和软件均属于太阁和合作伙伴所有并受版权法保护。

对于任何尝试散播或转售太阁所属资料的行为，太阁将采取相应的法律行动。

我们非常感谢您尊重我们的版权内容。

详情参阅，

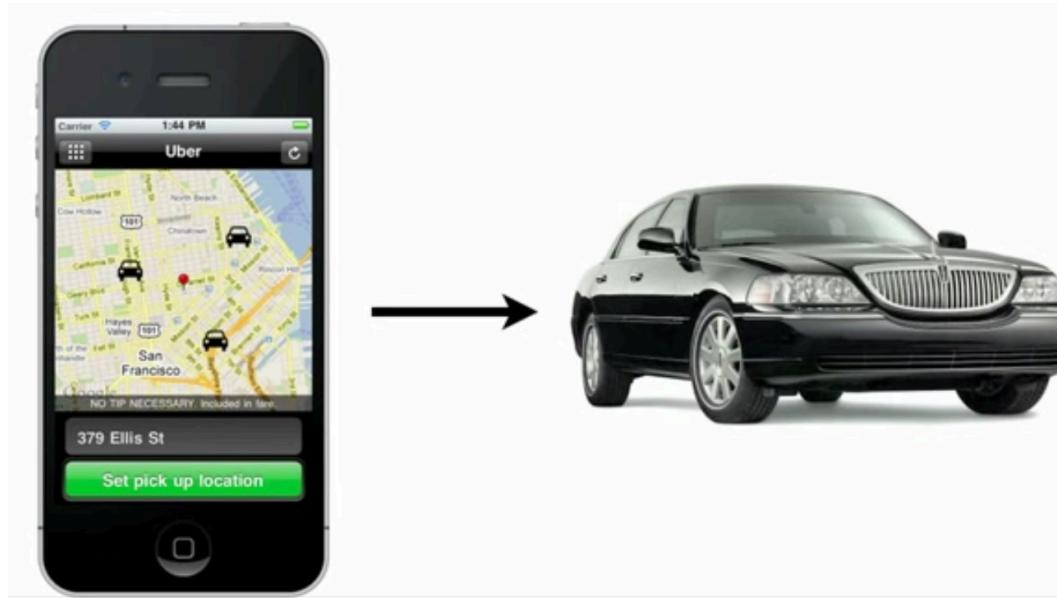
<https://www.bittiger.io/termsofuse>

<https://www.bittiger.io/termsofservice>

Outline

- From 0 to 1
- From 1 to million
- S2
- Ringpop

What is Uber?



One button for your taxi

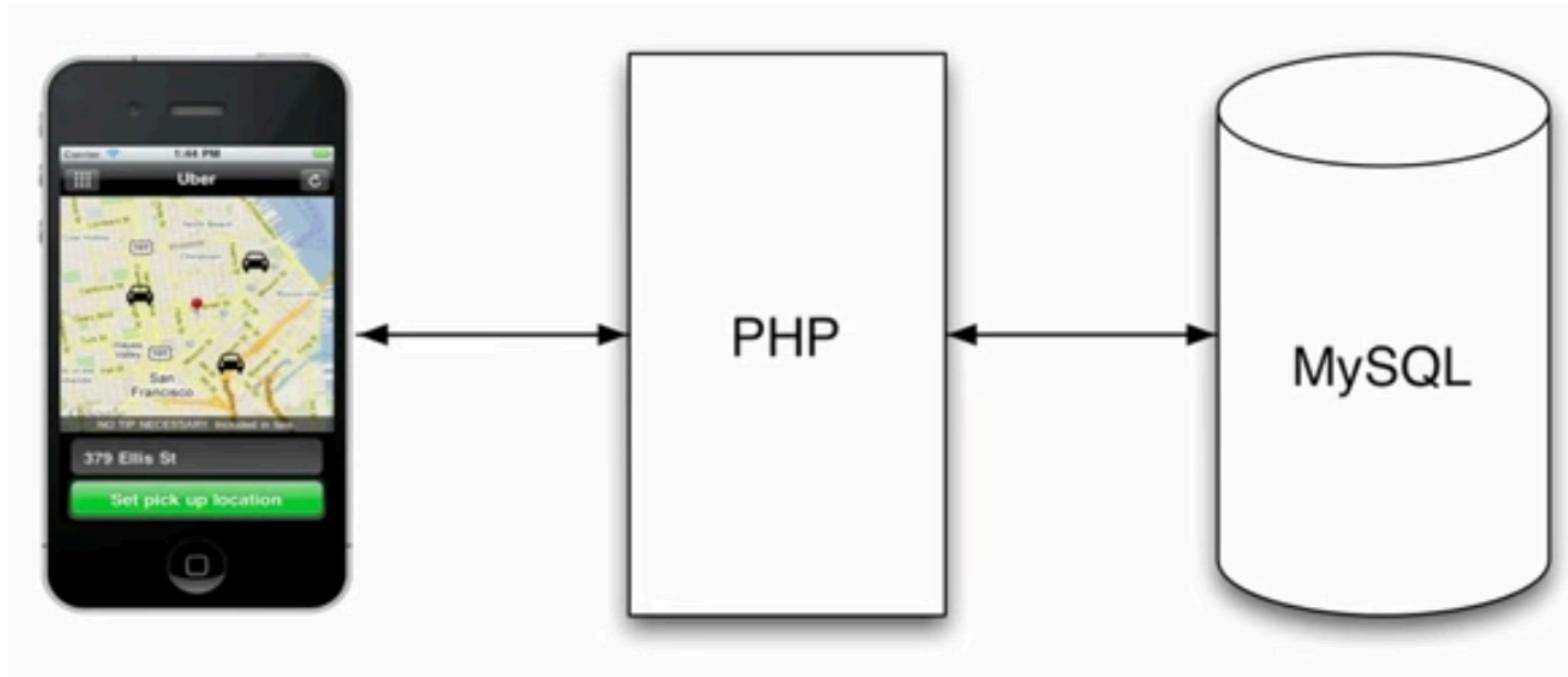
What is BitTiger?



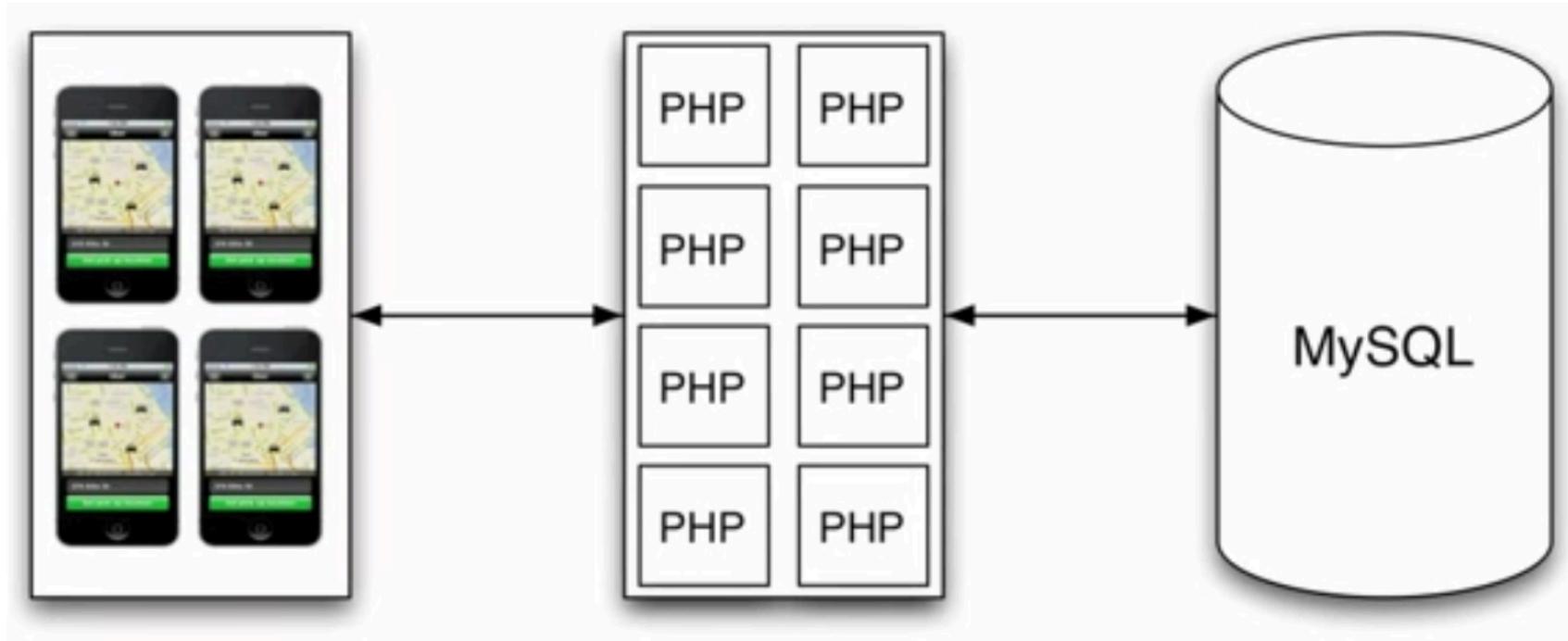
你与Offer
其实只有一个屏幕的距离

硅谷精英全程伴你 留学美国 入职硅谷

First Architecture of Uber

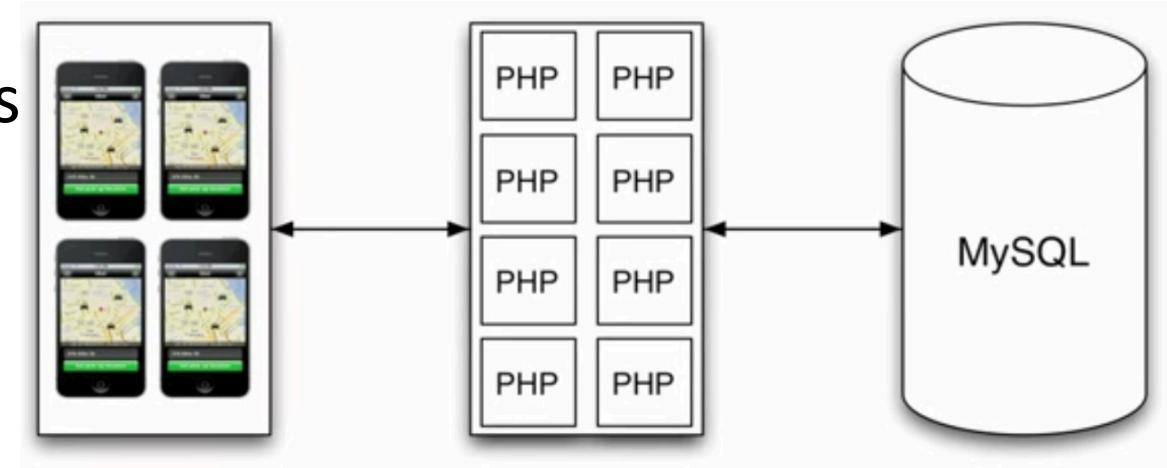


How to scale?

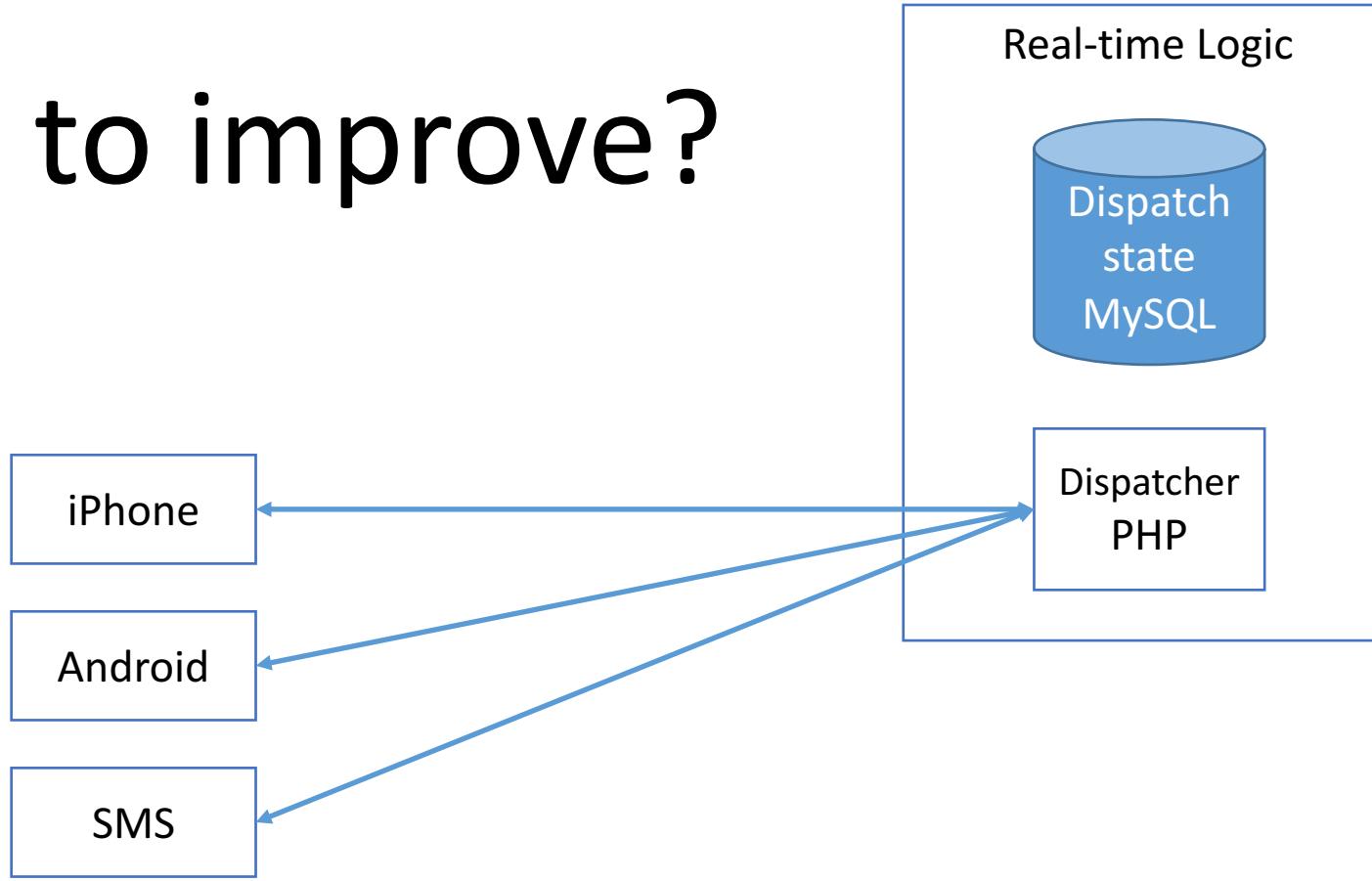


Challenges

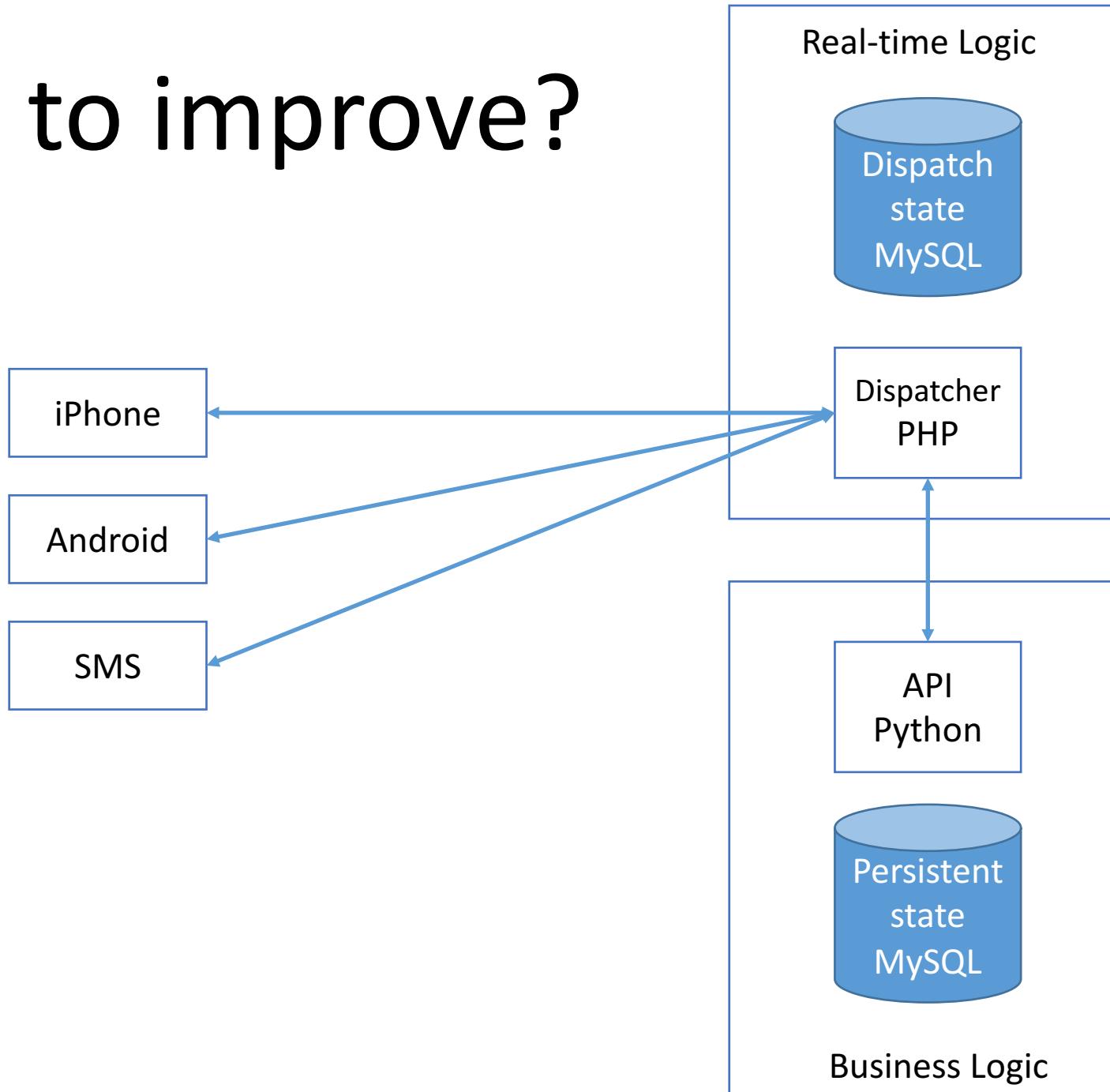
- One driver and Two Passengers
- Two drivers and One Car
- One account and Two Cars



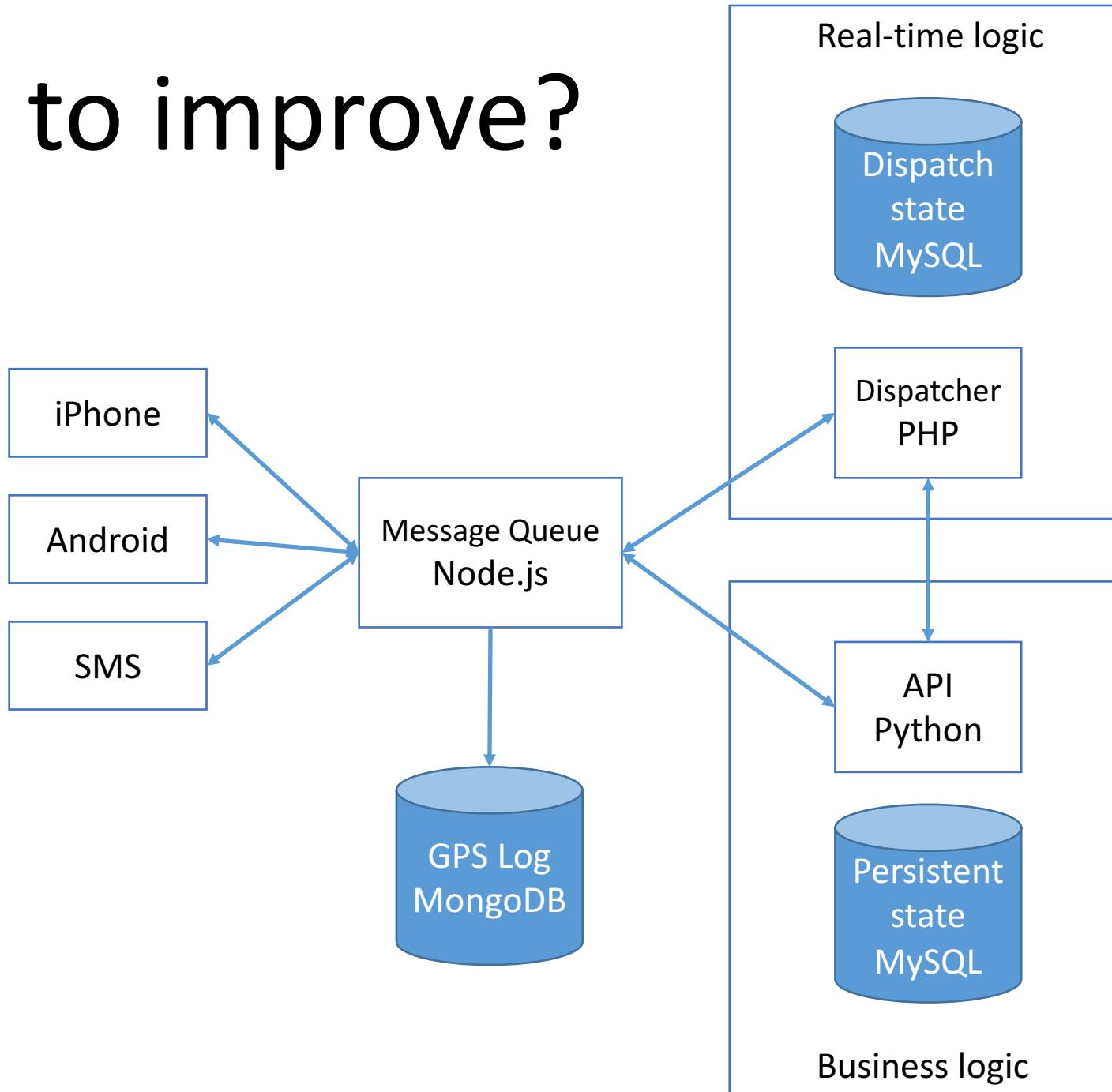
How to improve?



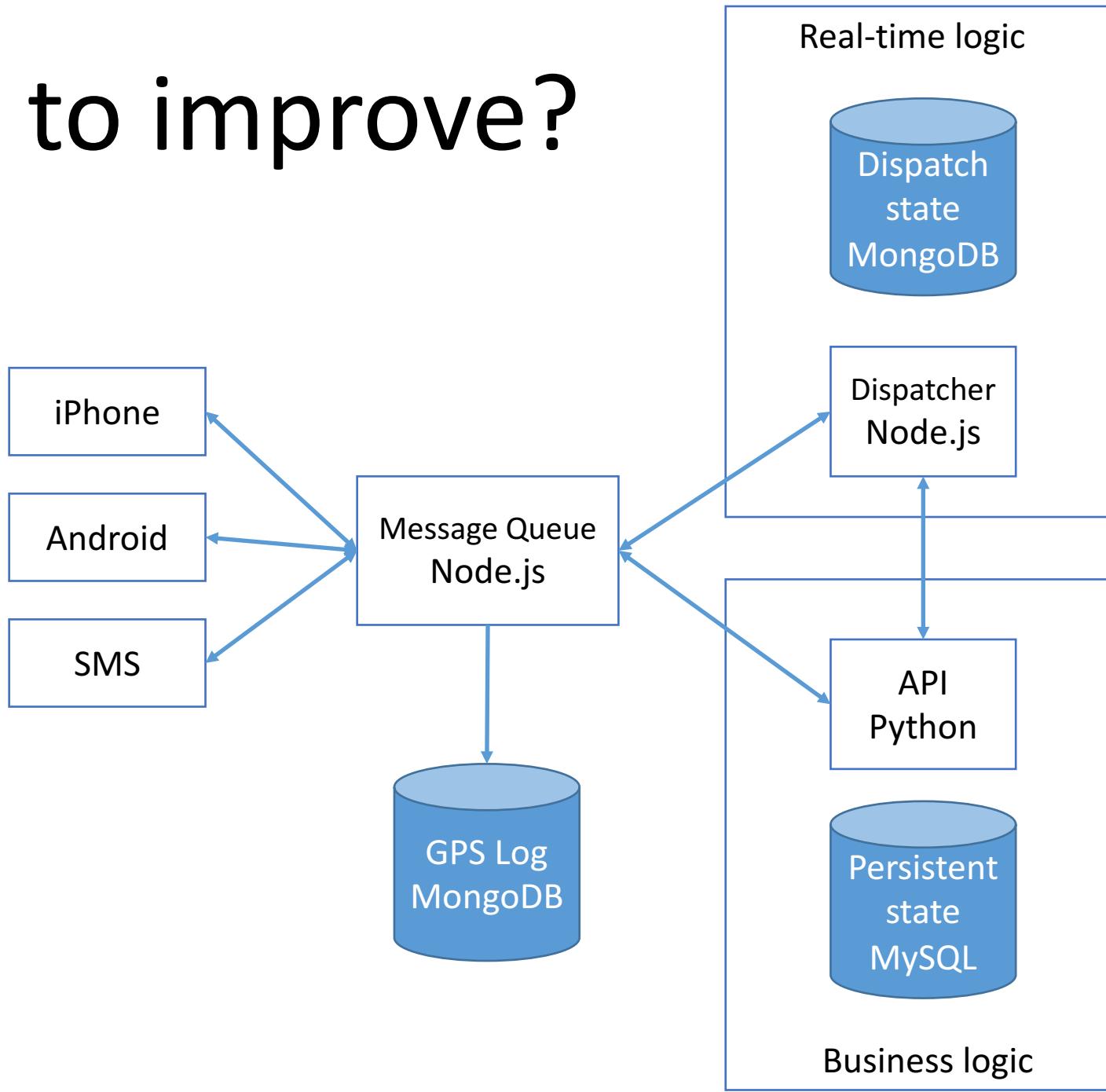
How to improve?



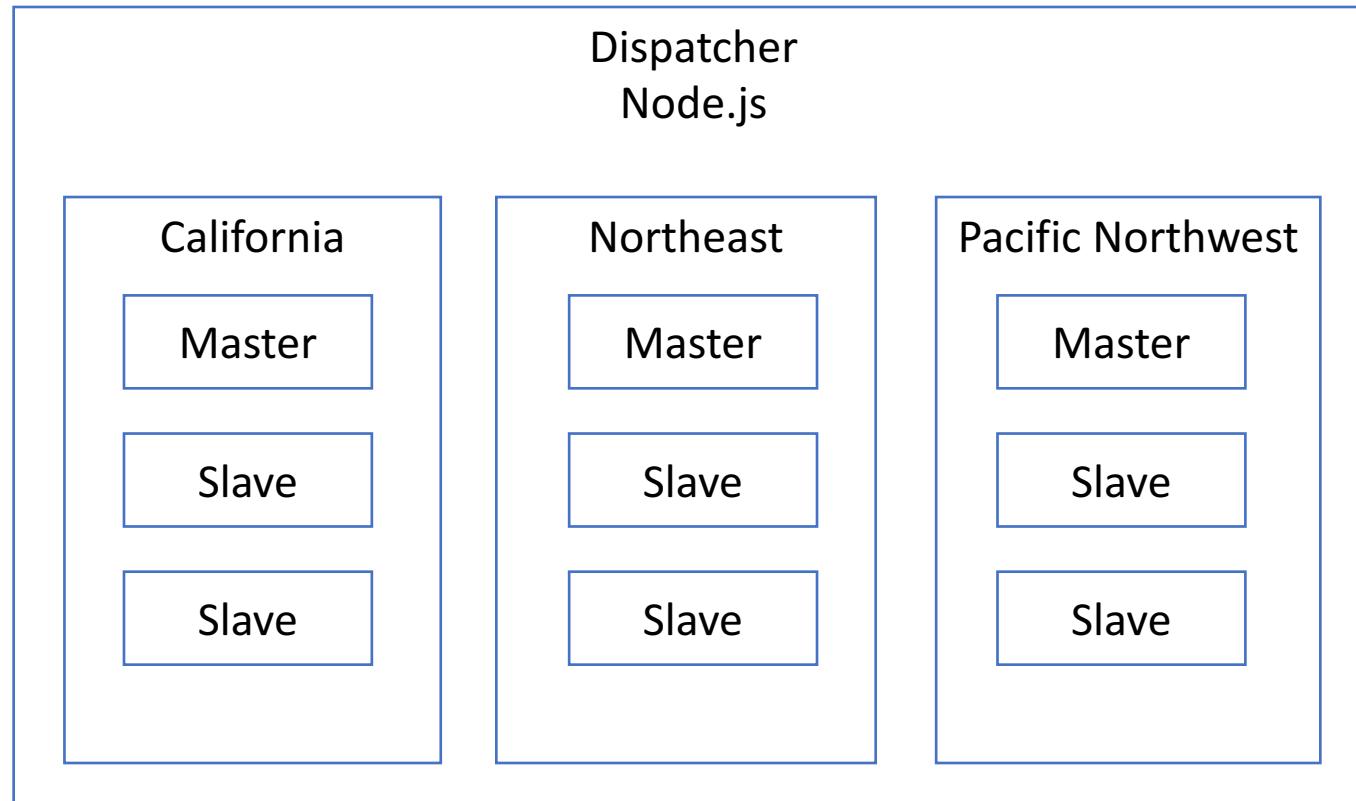
How to improve?



How to improve?



How to avoid single point of failure?



How to deal with message?

- RequestManager

```
require('request-manager');

// In constructor
this.requestManager = new RequestManager(9000);

// Inside createServer
request.addListener('end', function () {
    var connection = self.requestManager.createContext({
        request:      request,
        requestBody: requestBody,
        response:     response
    });

    if (connection) {
        self.processMessage(connection);
    }
});
```

How to backup?

- SlaveMaster

```
require('slave-master');

var serverList: [ {
    id: 1,
    host: 'localhost',
    path: '/',
    port: 8000,
    rmPort: 8001,
    smPort: 8002
}, ... ];

// In constructor
this.slaveMaster = new SlaveMaster(serverID, serverList);
this.slaveMaster.onBecomingMaster = function() {
    // Load data from Mongo
};
```

How to visit MongoDB?

- MongoWrapper

```
require('mongo-wrapper');

// In constructor
this.mongo = new MongoWrapper({
    replicaSet: settings.mongo.replicaSet,
    databaseName: 'testDB',
    retryLimit: settings.mongo.retryLimit,
    retryInterval: settings.mongo.retryInterval,
    backupDir: settings.mongo.backupDir,
    deadDir: settings.mongo.deadDir,
    restoreInterval: settings.mongo.restoreInterval
});

this.mongo.insert('gpsLogs', gpsLog);
```

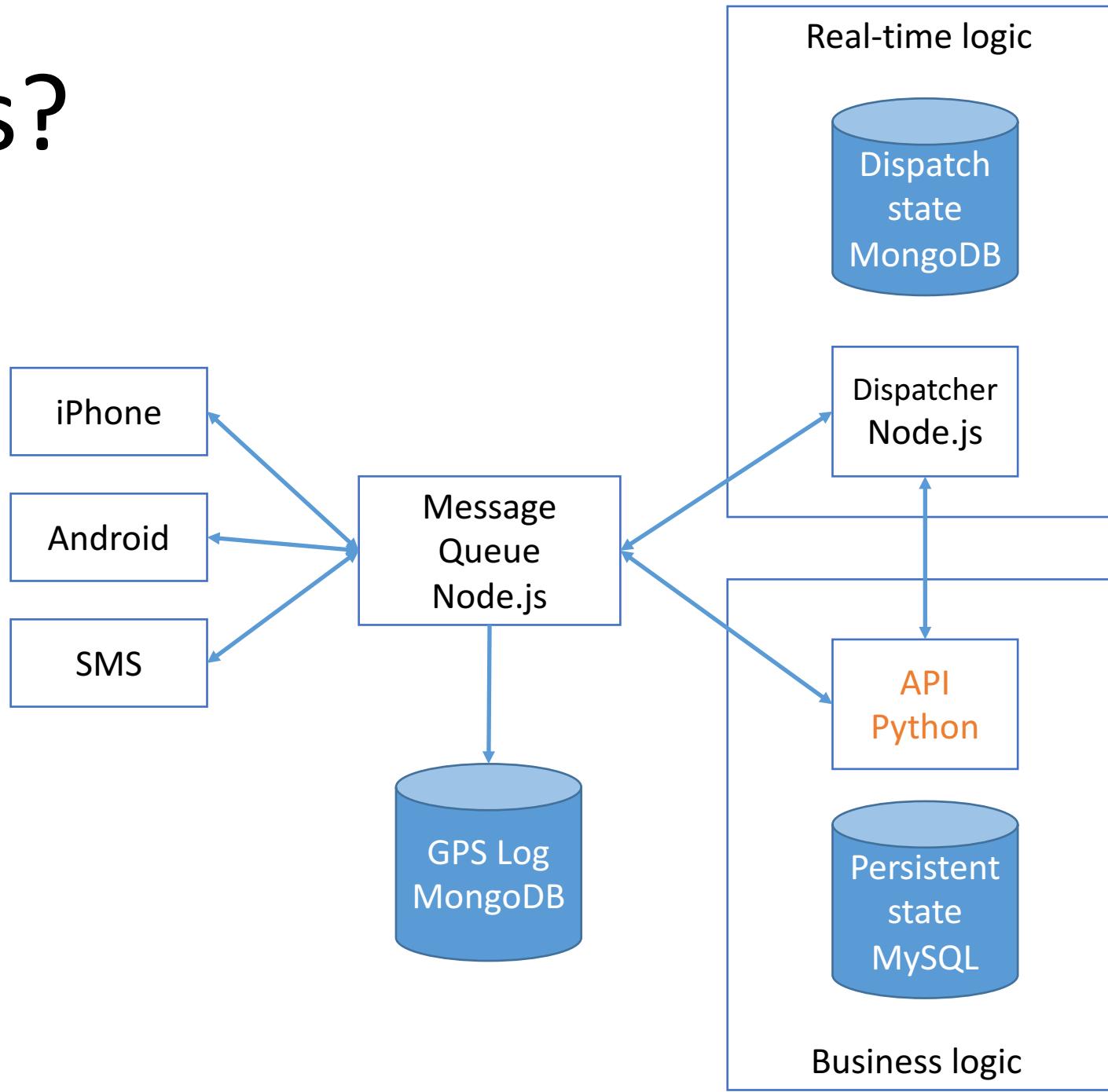
Numbers for 2011.8

- Average QPS
 - 25/s
- Peak QPS
 - 125/s
- Is it enough?
 - X10
 - 1000 users
 - 14% CPU + 60MB RAM

How to monitor?

- Reboot
- Email notification
- Dashboard

Risks?



The mission of Uber

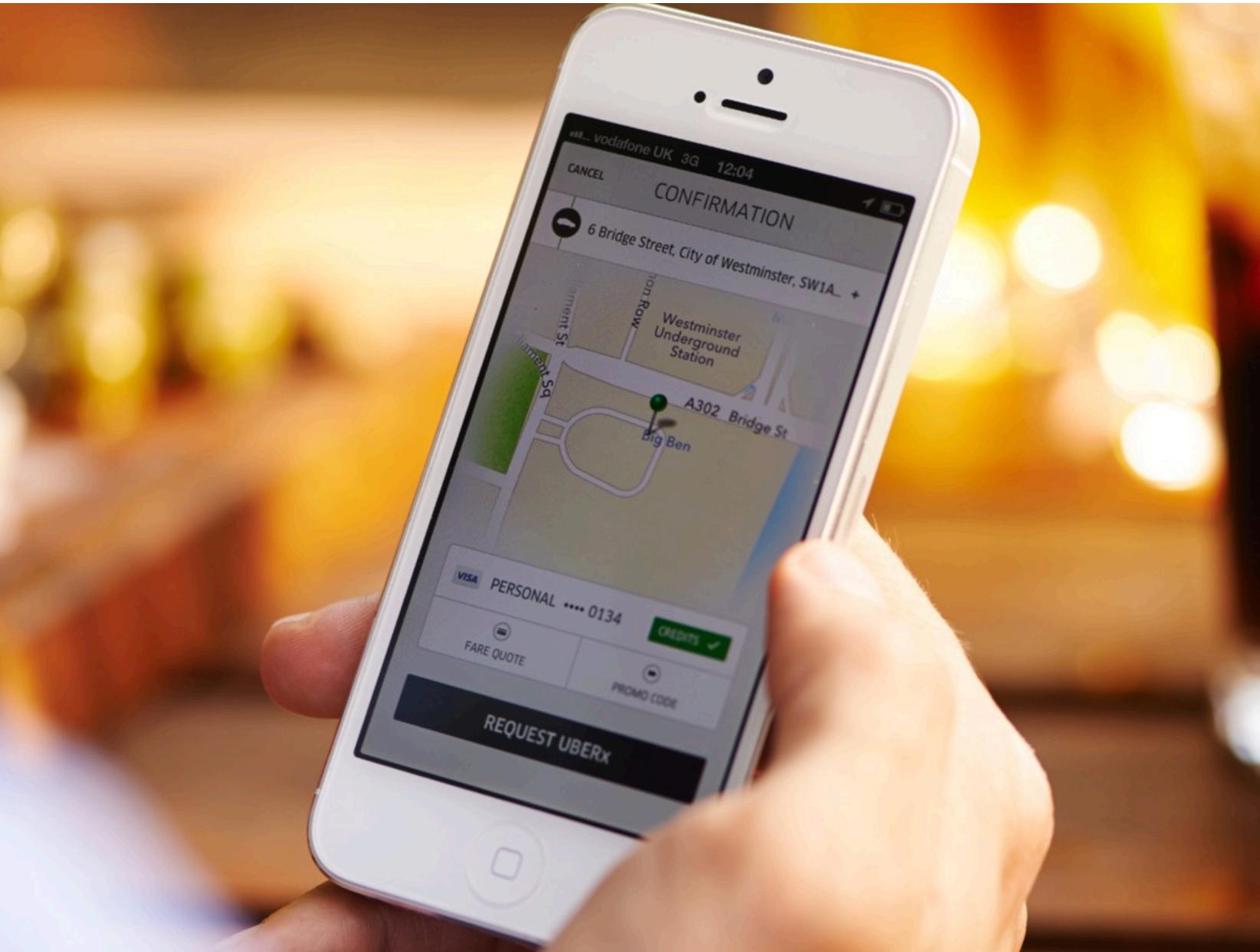
Transportation as reliable as running water,
everywhere, for everyone

Outline

- From 0 to 1
- From 1 to million
- S2
- Ringpop

What is Uber?

A platform connecting drivers and passengers



New challenges for Uber

- Dynamic supports
- Dynamic requests

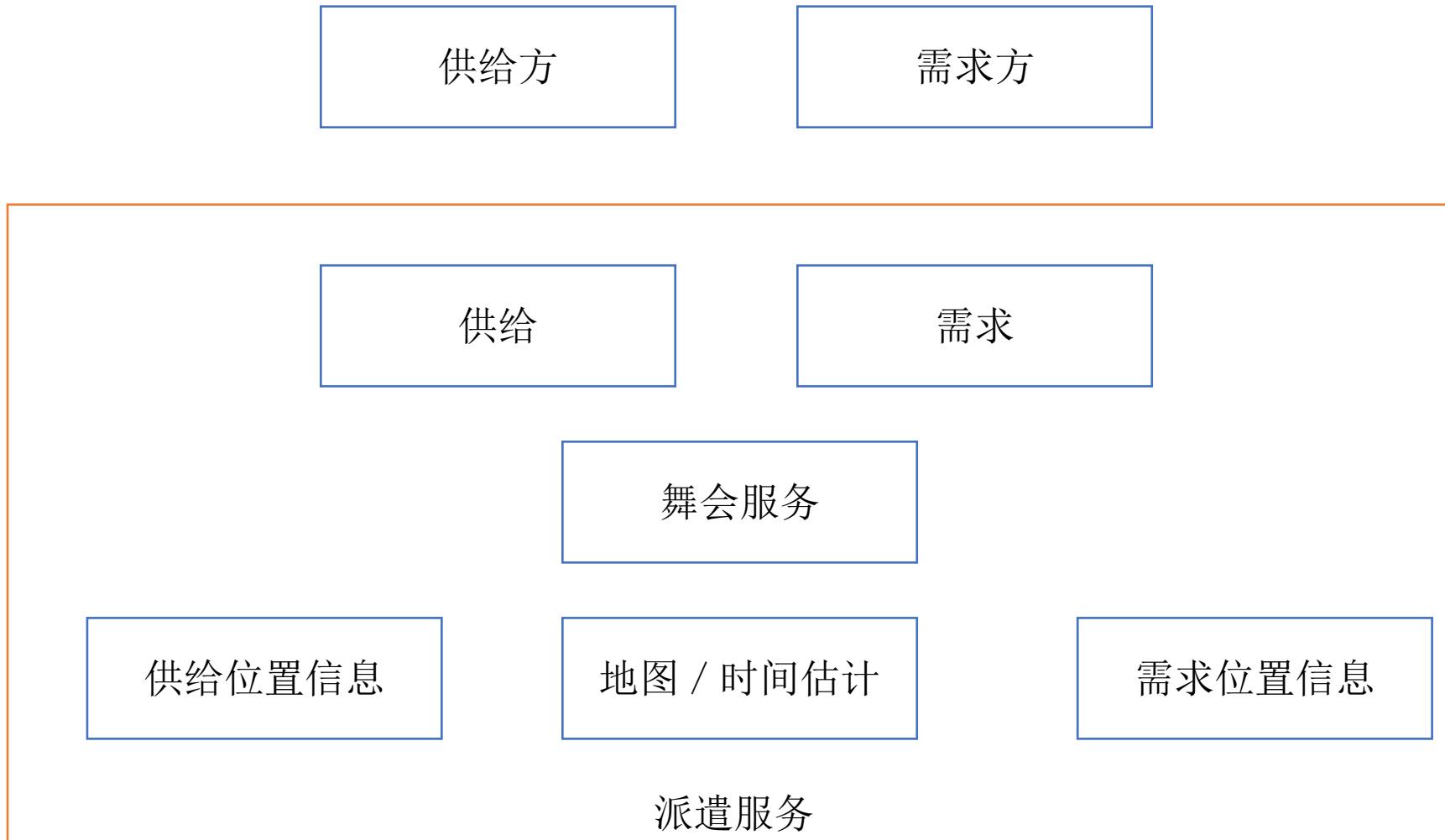
Uber的架构



挑战

- 顺风车
 - 假设：1个司机+1个乘客
- 送餐
 - 假设：传输的是人
- 跨城市运输
 - 按照城市切分数据
- 多点失败

如何重构？



目标

- 写操作: 1M/s

如何唯一标识一块空间？

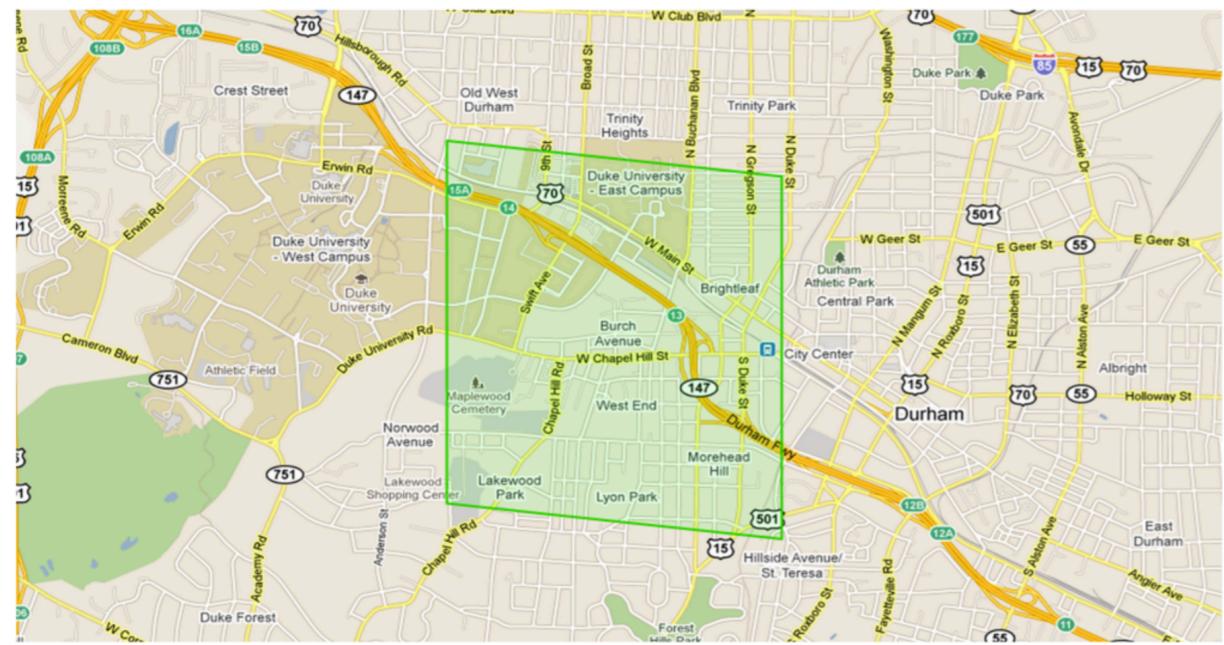
- Google S2

Level	Min Area	Max Area
0	85,011,012 km ²	85,011,012 km ²
1	21,252,753 km ²	21,252,753 km ²
12	3.31 km ²	6.38 km ²
30	0.48 cm ²	0.93 cm ²

↑
smallest cell

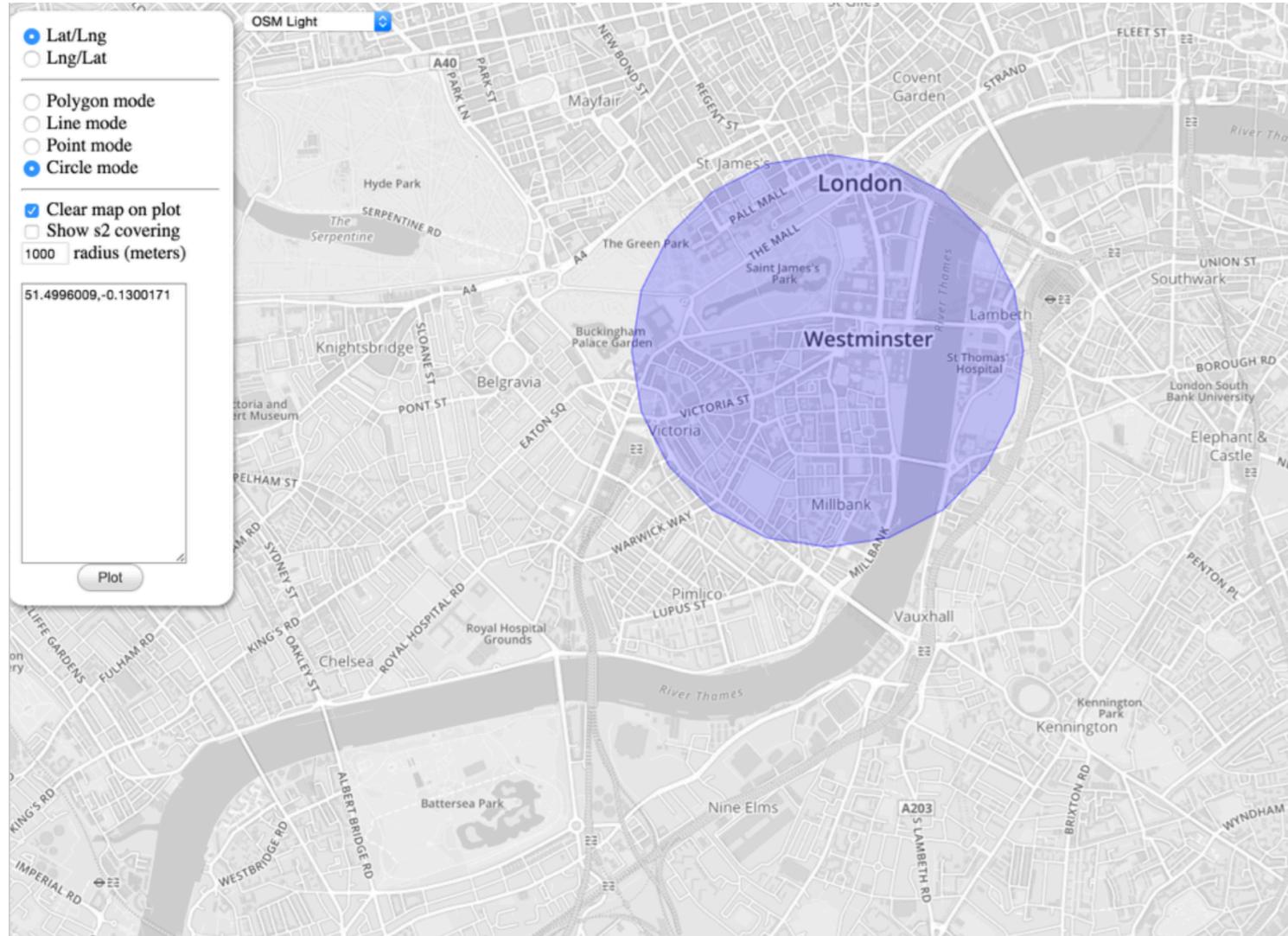
Every cm² on Earth can be represented using a 64-bit integer.

Id: 0x89ace41000000000 (0b1000100110101100111001000001000...), Level: 12

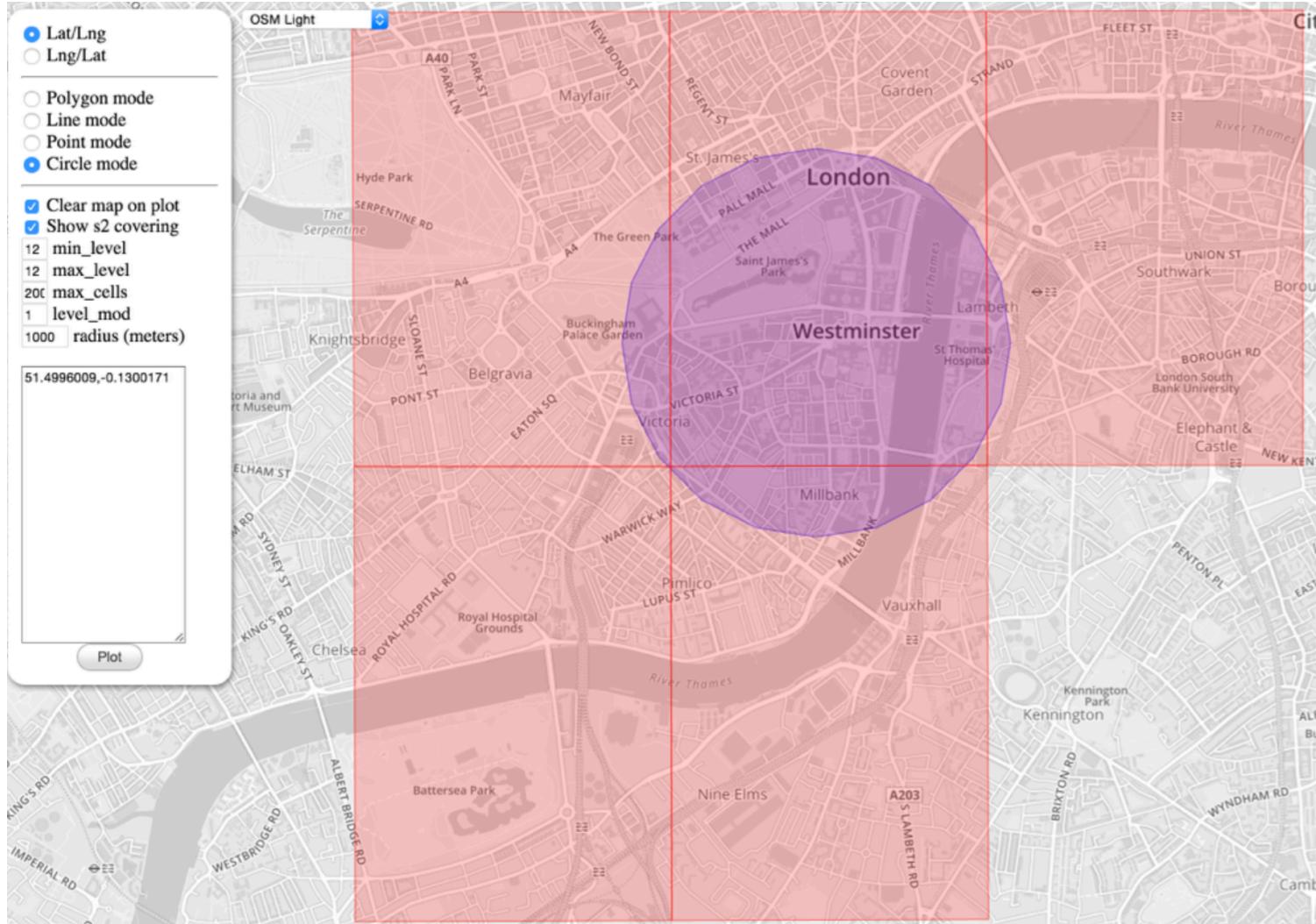


Source: Geometry on the Sphere: Google's S2 Library

如何表示一个区间？



区间覆盖



如何匹配供需？

- 減少等待
- 減少空駕
- 減少通勤時間

什么是最优策略？

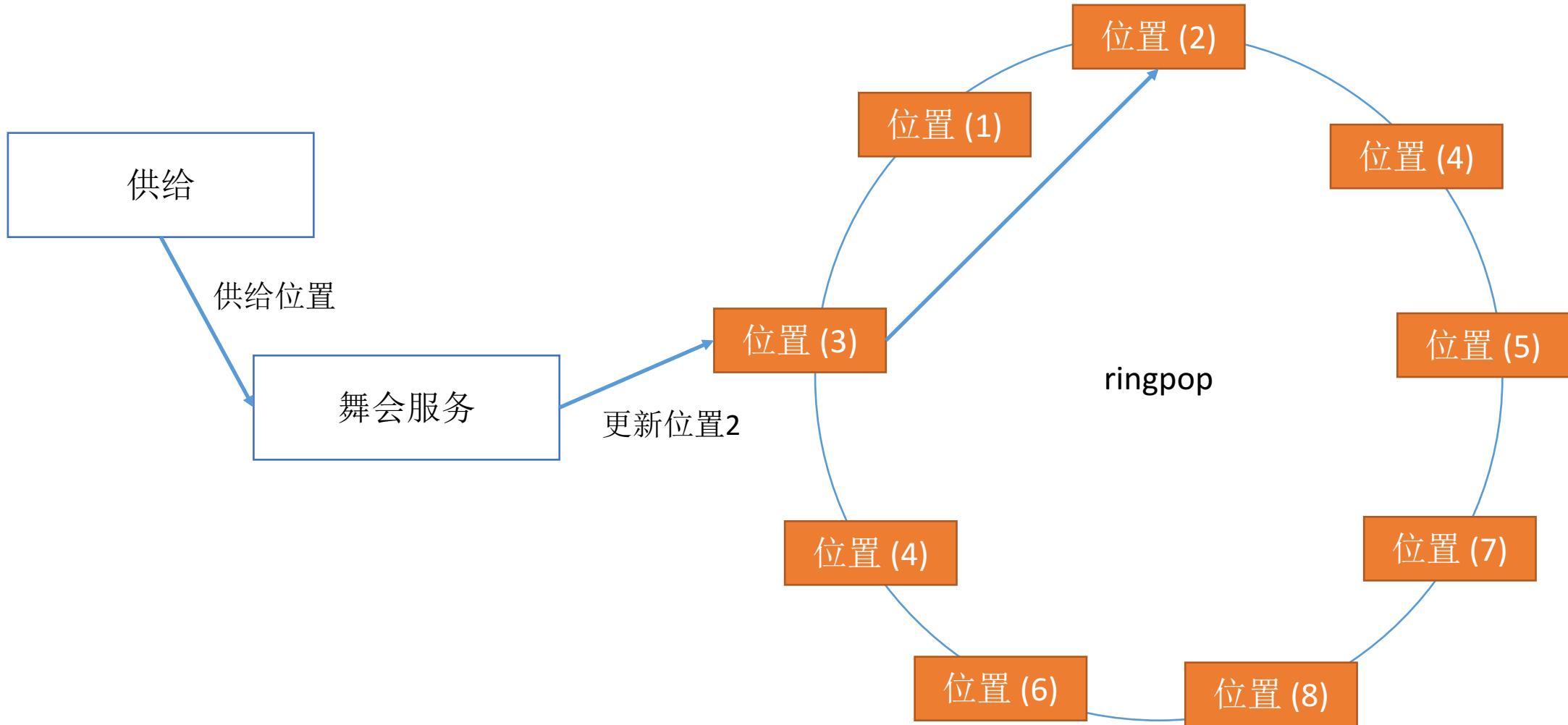
乘客1：发送请求

司机1：距离8分钟

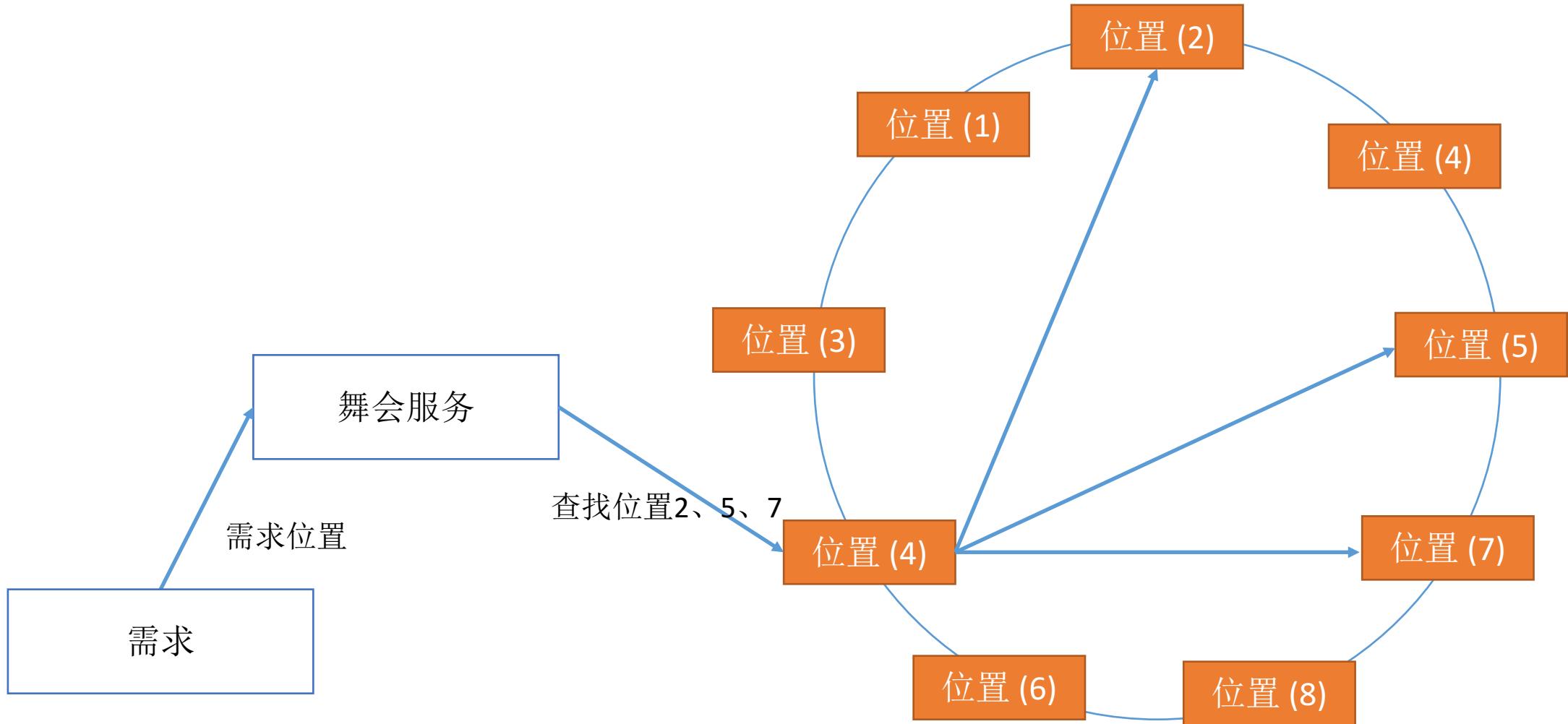
司机2：在2分钟后完成派送；距离乘客1分钟



如何保存供给？



如何匹配需求？



如何远程通讯？

- 需求

- 性能优秀 ($20x > \text{HTTP}$)
- 消息转发
- 跨语言支持
- 消息调度优化
- 校验和追踪
- 消息封装

- 解决方案

- TChannel

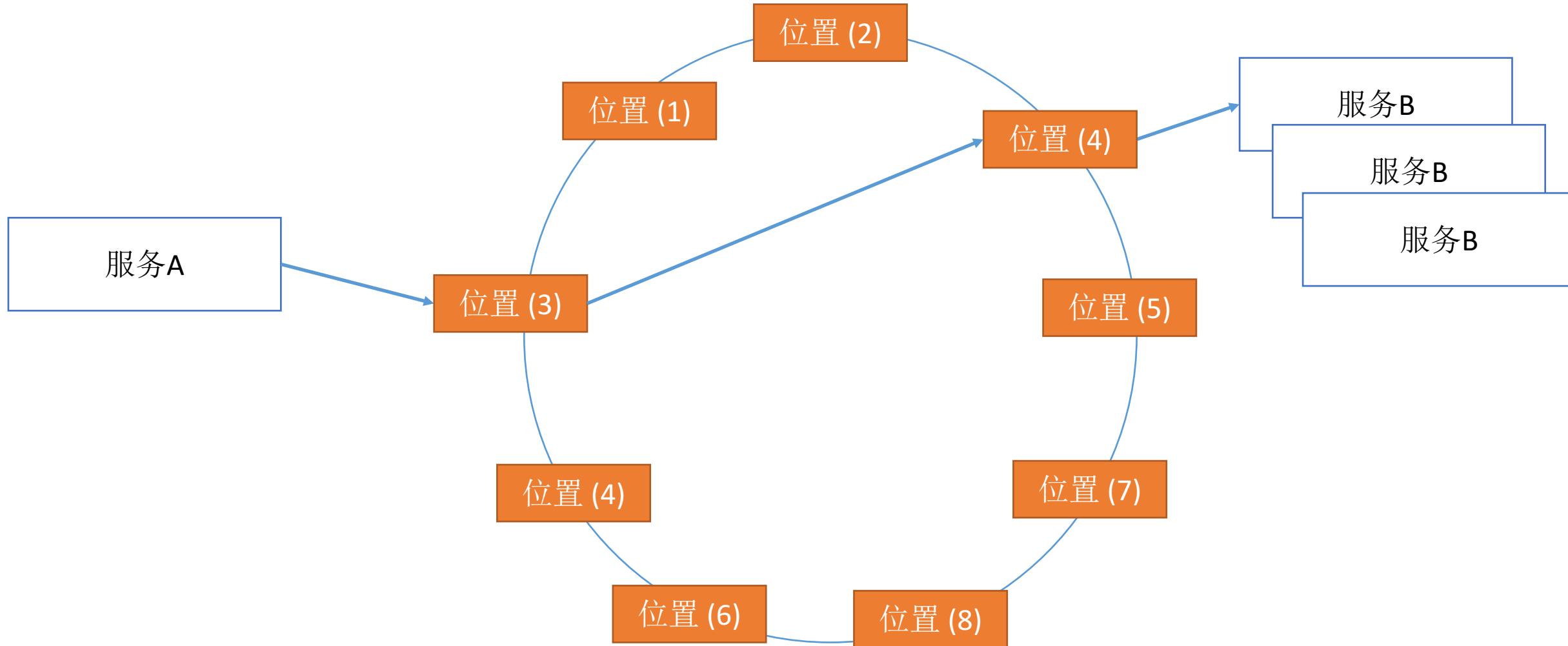
服务设计原则

- 可以重试
- 可以被杀
- 服务切分

如何负载均衡？



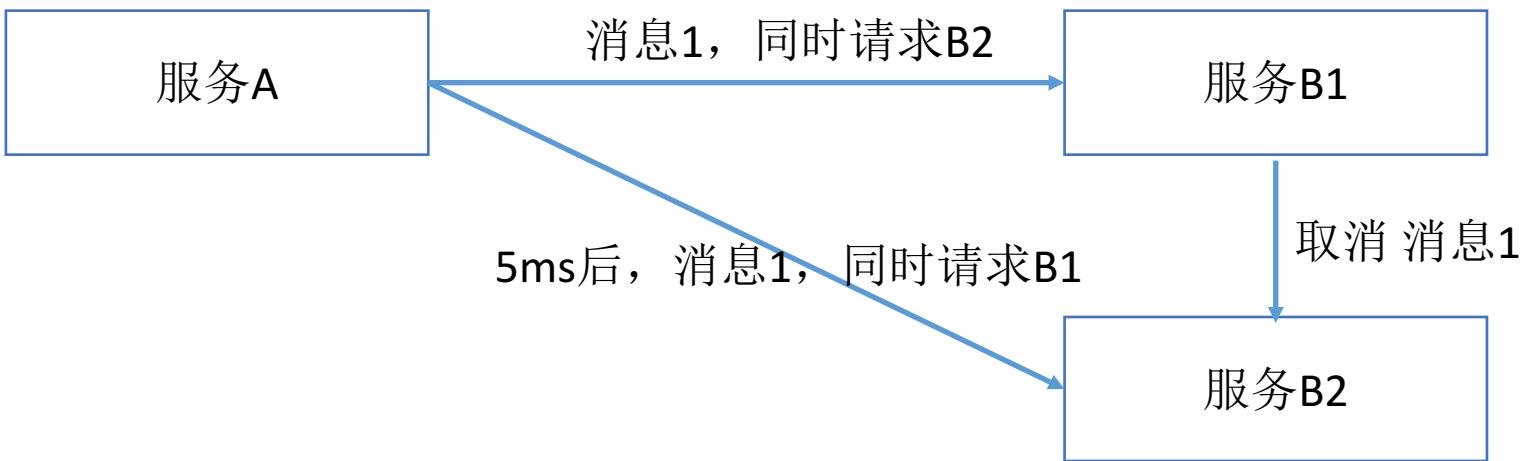
如何改进负载均衡？



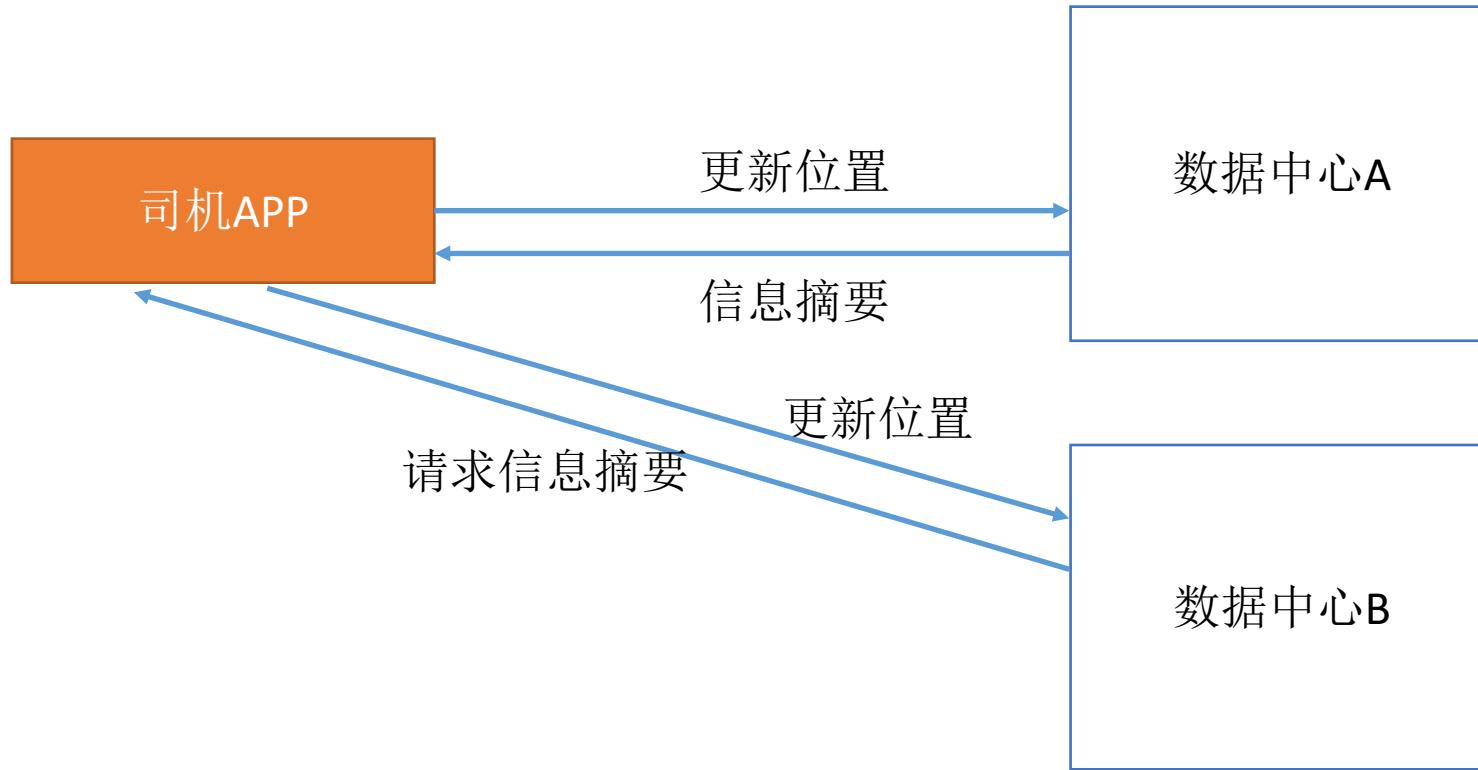
木桶延迟问题

- 1个消息
 - 平均1ms
 - 1%是1000ms
- 100个消息的集合
 - $1 - 99\%^{100} = 63\%$

如何木桶延迟？



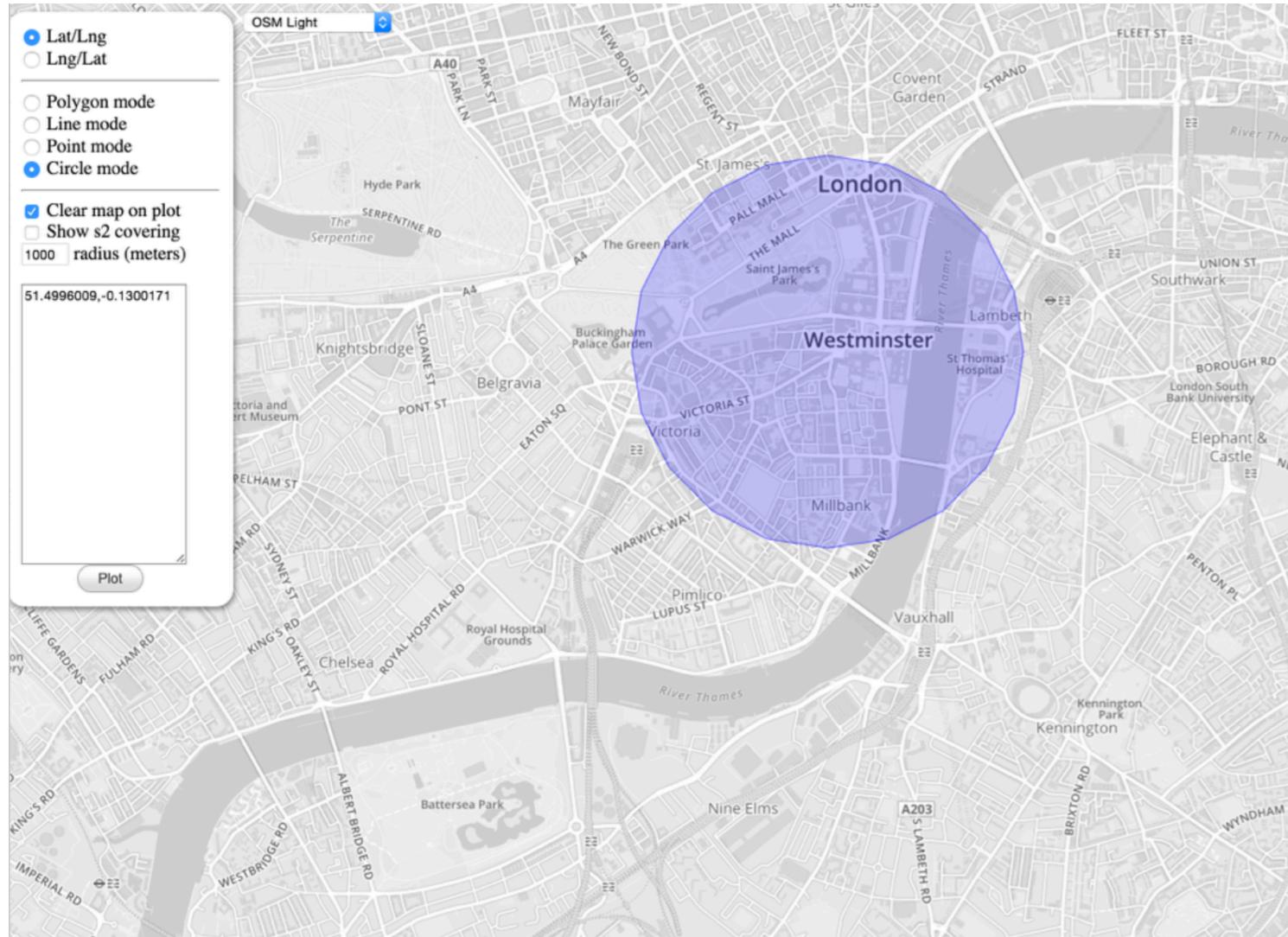
数据中心挂掉怎么破？



Outline

- From 0 to 1
- From 1 to million
- S2
- Ringpop

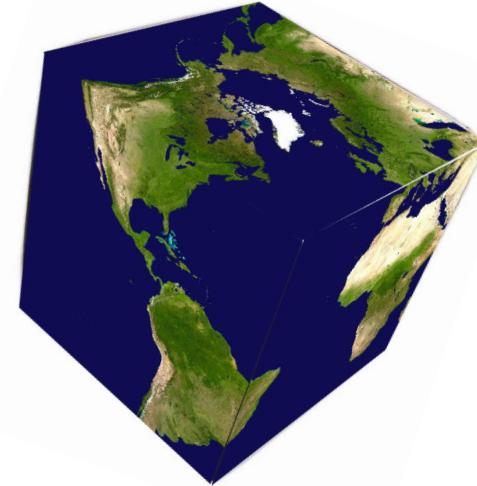
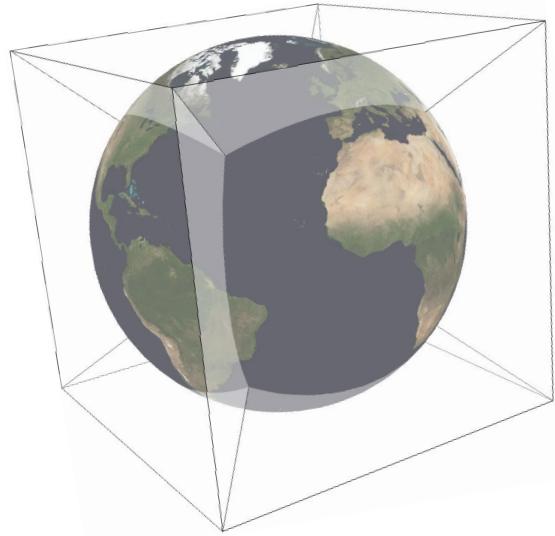
Uber的挑战：如何查找身边的司机？



挑战

- 如何唯一表示地球上的一块空间？
- 如何将地球切分成大小近似的区块？
 - 支持不同粒度的表示

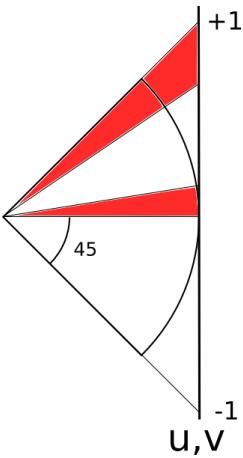
如何将三维变二维？



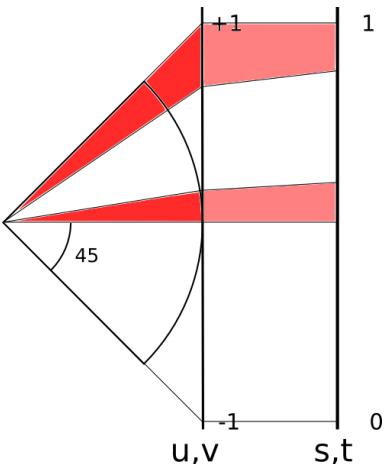
- 把地球放在正方体中
- 从地球中心发光
- 求地球表面到正方体的投射点
- => 6个正方形

投射的区间比例不同

- 中间短两边长



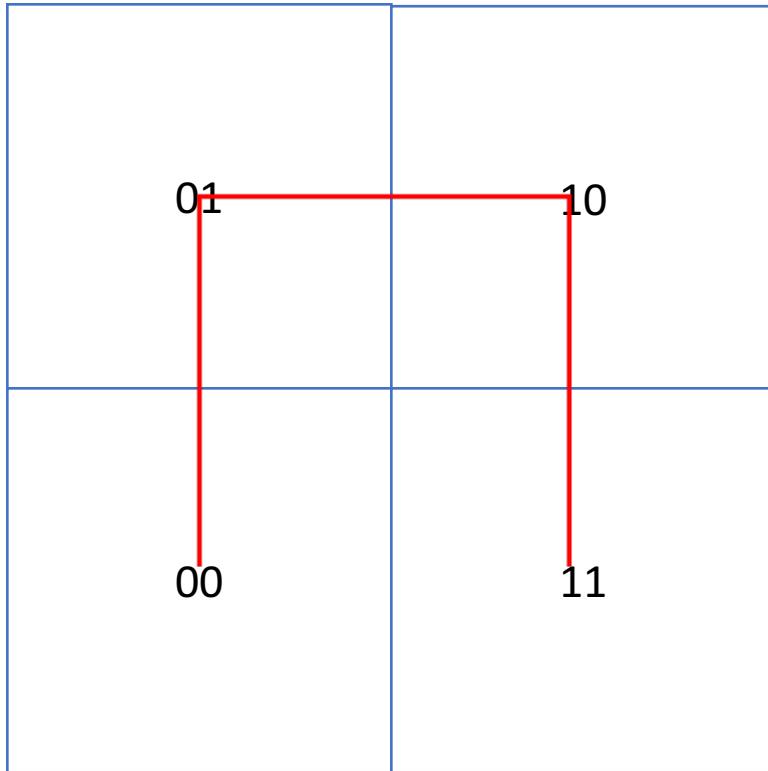
- 加入区间转换



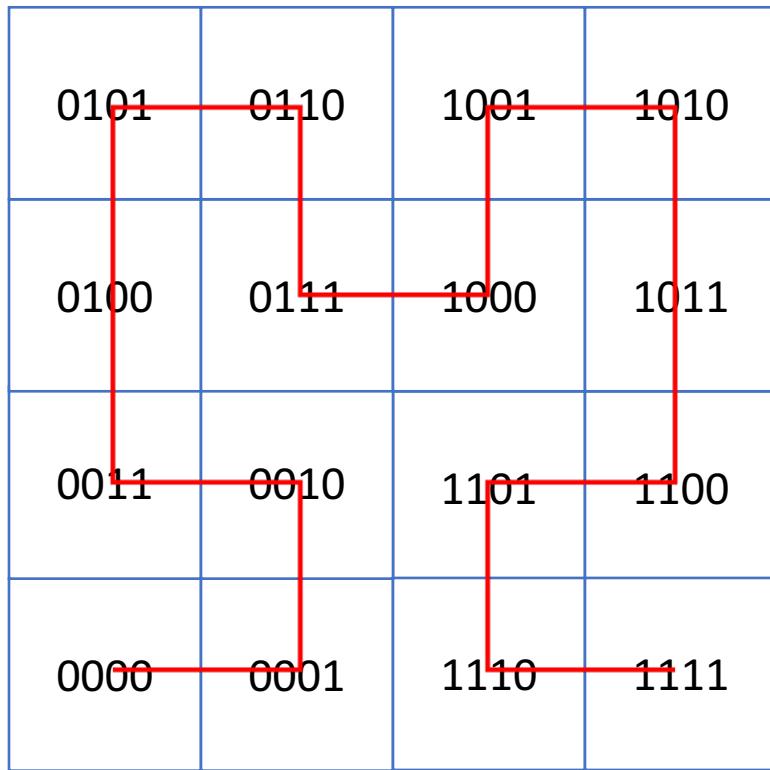
如何选择转换算法？

候选方案	平衡度（最大面积/最小面积）	计算效率
线性	5.20	0.086
正切	1.41	0.279
二次	2.08	0.102

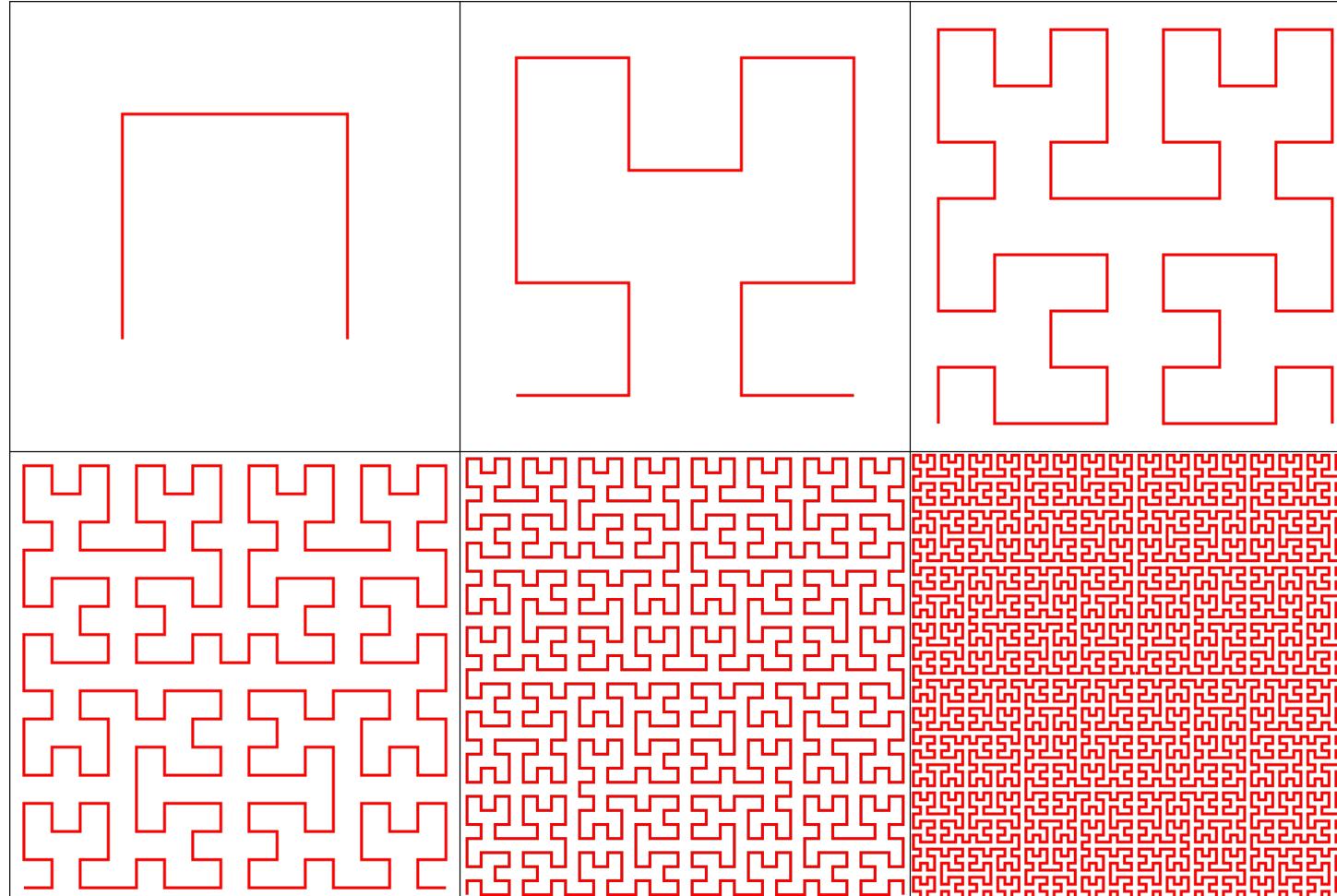
如何将二维变一维



如何表示不同粒度



更多的粒度 (Hilbert curve)



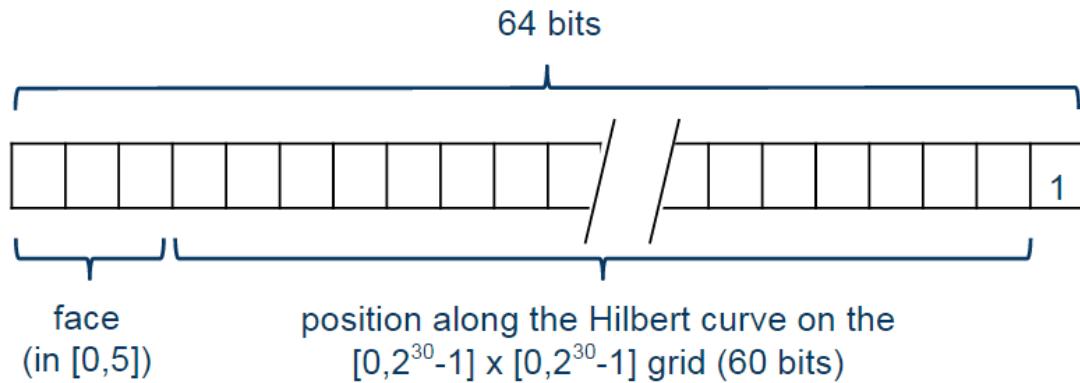
Google S2的粒度

Level	Min Area	Max Area
0	85,011,012 km ²	85,011,012 km ²
1	21,252,753 km ²	21,252,753 km ²
12	3.31 km ²	6.38 km ²
30	0.48 cm ²	0.93 cm ²

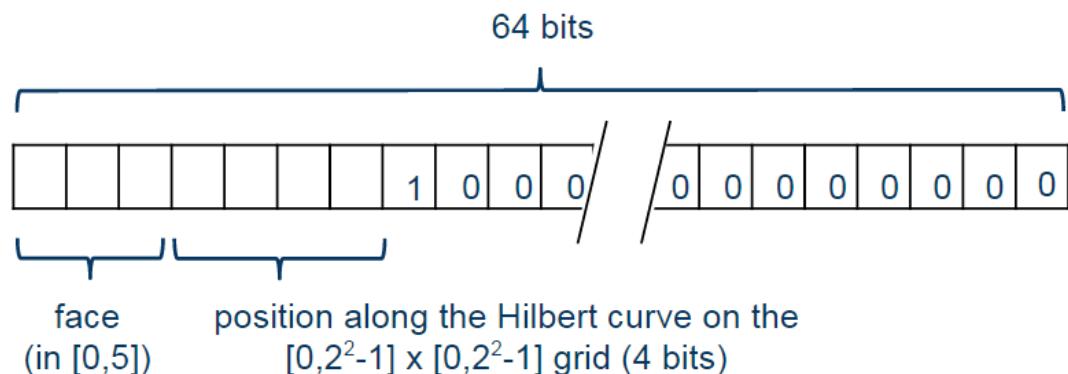
↑
smallest cell

二进制表示一个区间

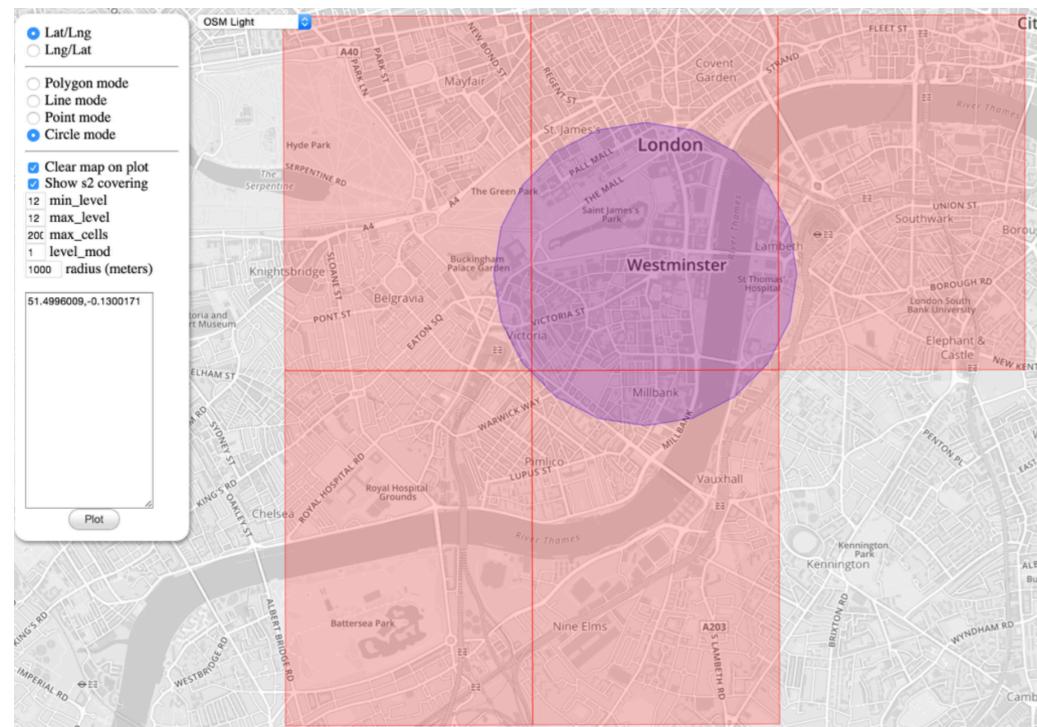
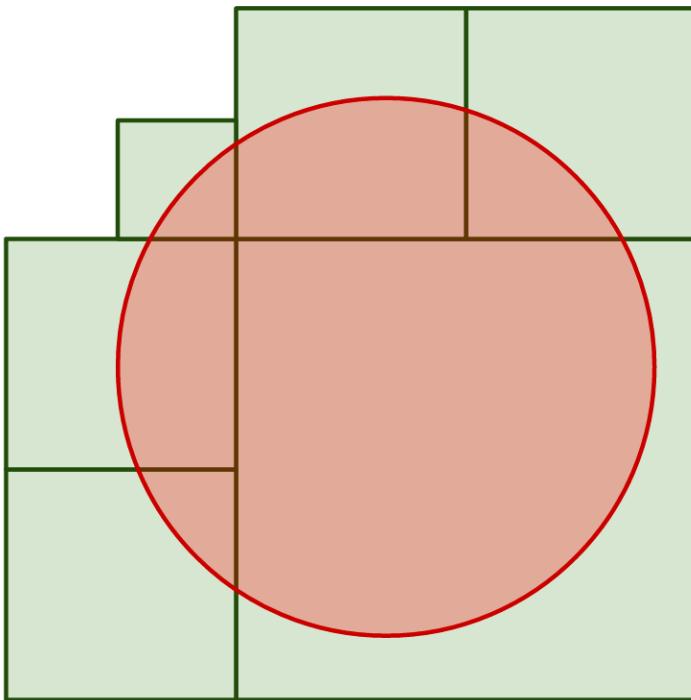
S2 Cell ID of a **leaf** cell (level 30):



S2 Cell ID of a **level-2** cell:



如何覆盖一个非规则区间



总结

- 地球发光变平面
- 希尔伯特变成线
- 区间覆盖连成片

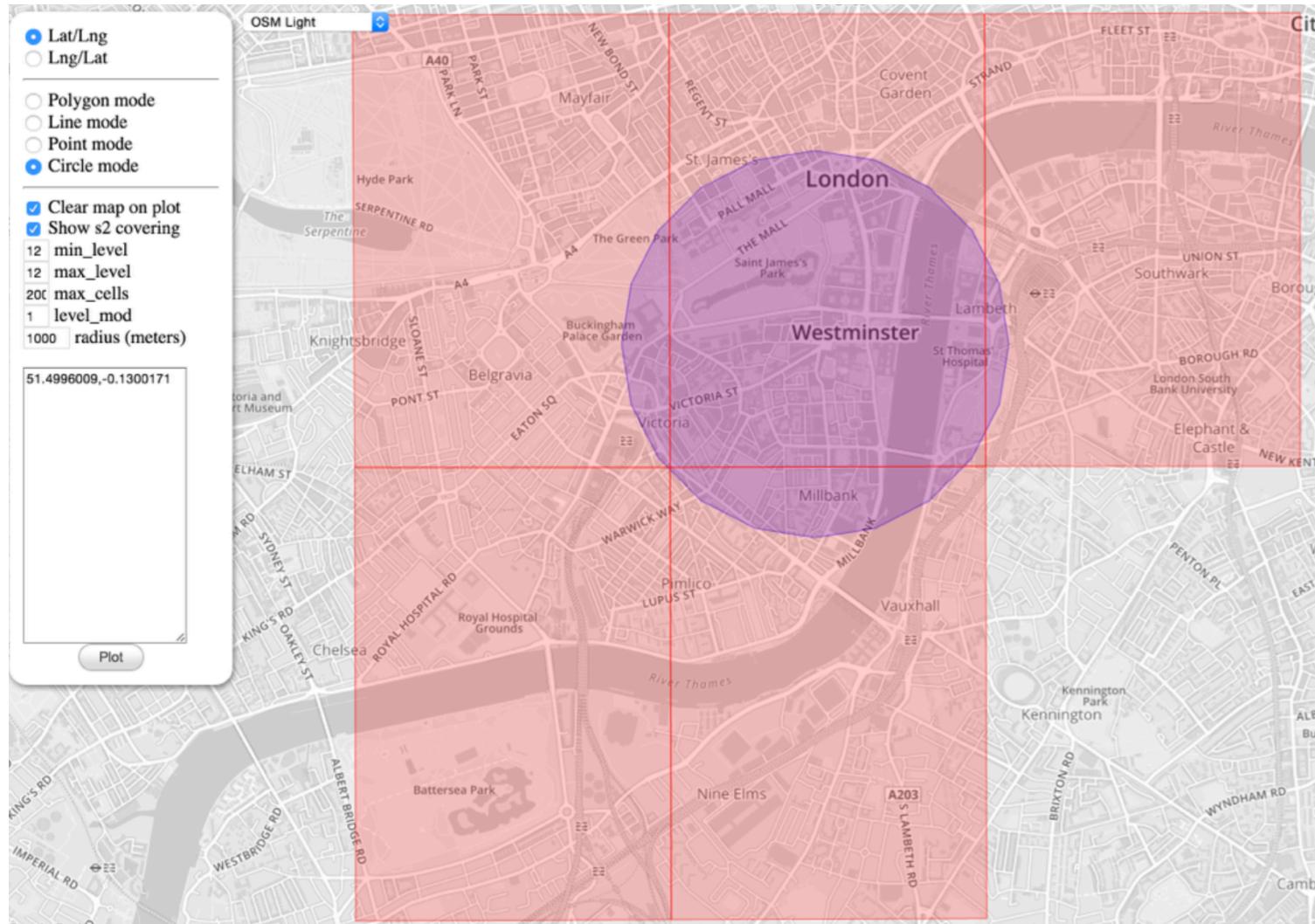
Outline

- From 0 to 1
- From 1 to million
- S2
- Ringpop

如何支持Uber的派遣服务

- 每4秒更新一次司机的位置信息
 - 写操作: 1,000,000 / 秒
- 乘客查找身边的车辆
 - 检索操作: 10,000 / 秒

如何查找身边的司机？



挑战

- 单机无法提供服务
- 有中心的架构存在单点失败
- 如何实现无中心的架构

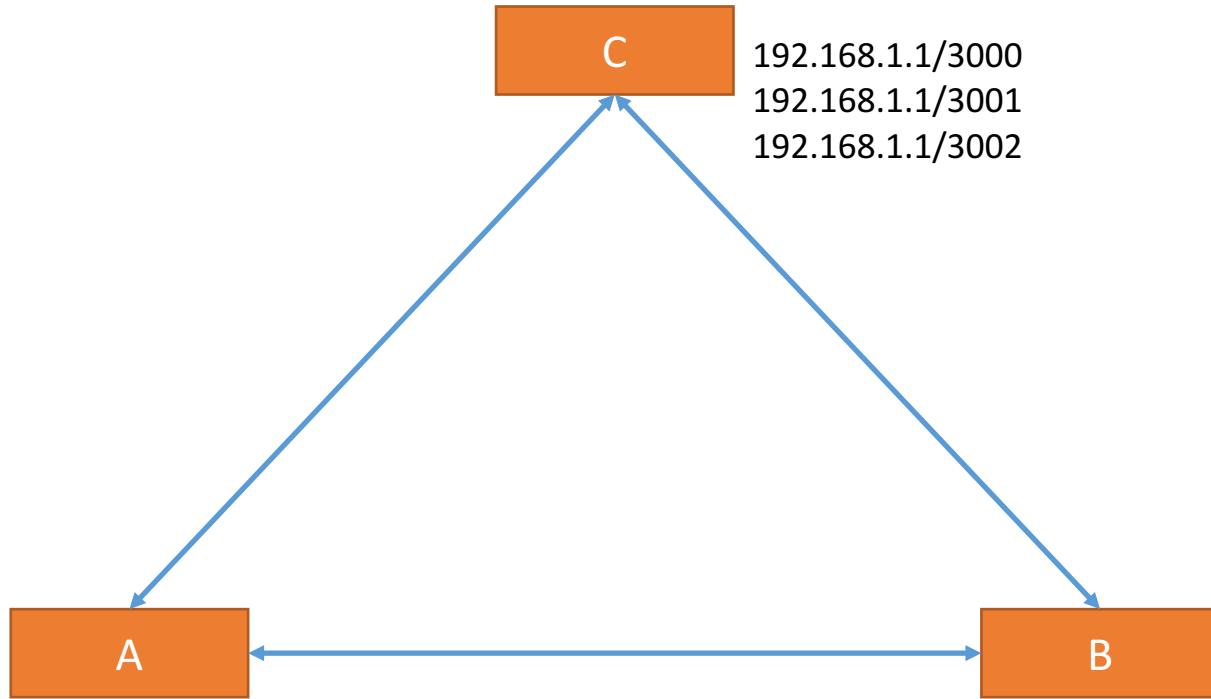
如何让服务器连接起来

- 读取初始列表



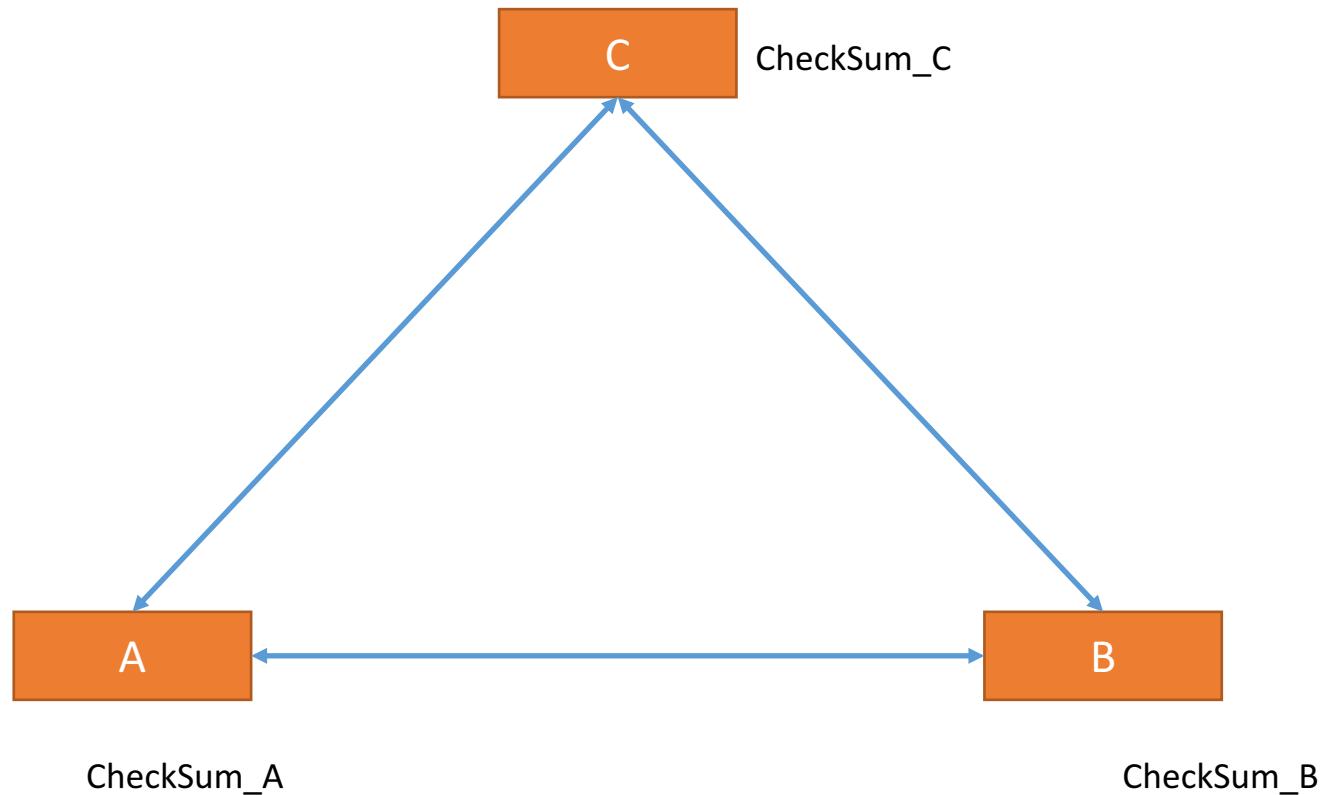
如何加入更多节点

- 读取初始列表
- 通过随机Ping互相传递消息



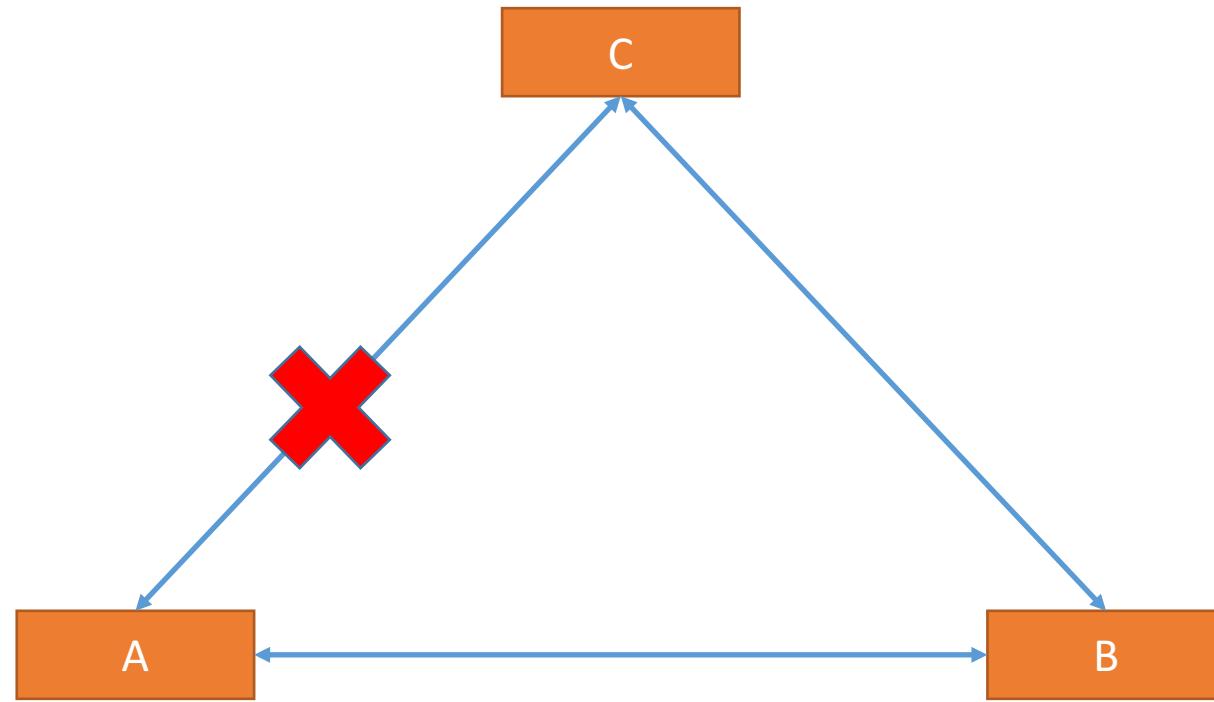
如何同步信息？

- Ping的时候校验CheckSum



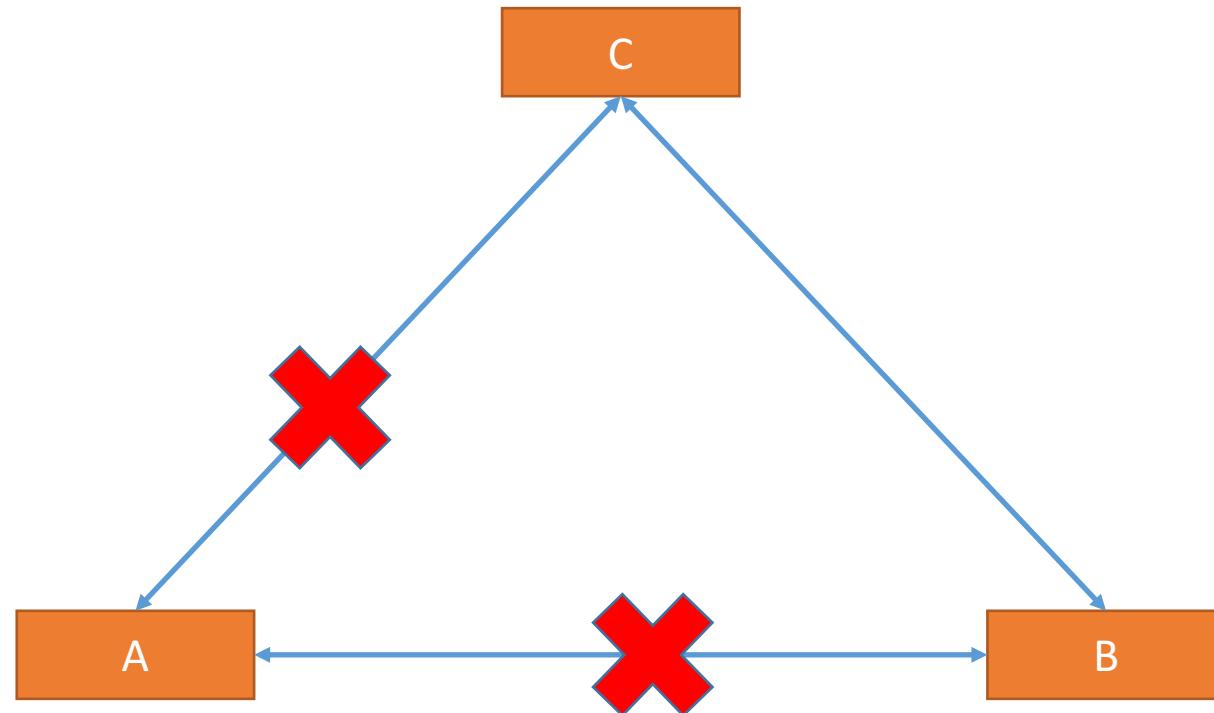
如何发现失败节点？

- Ping的时候发现异常



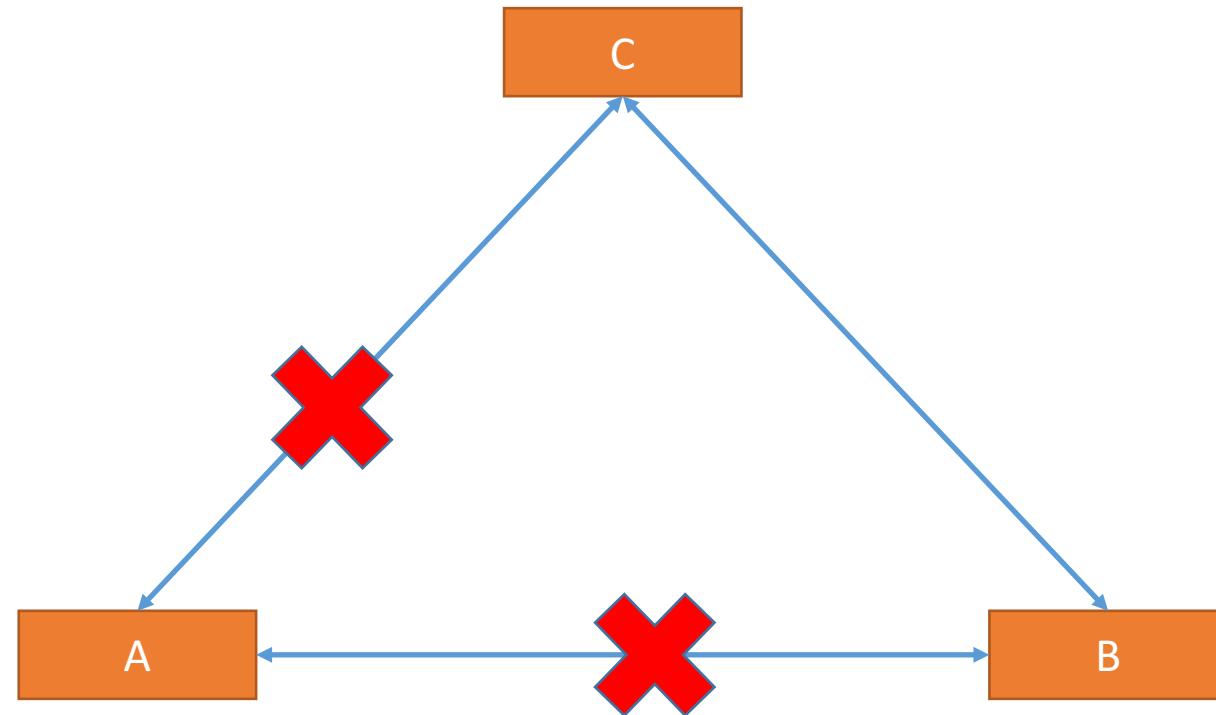
如何避免是自己的问题？

- 询问身边k名伙伴



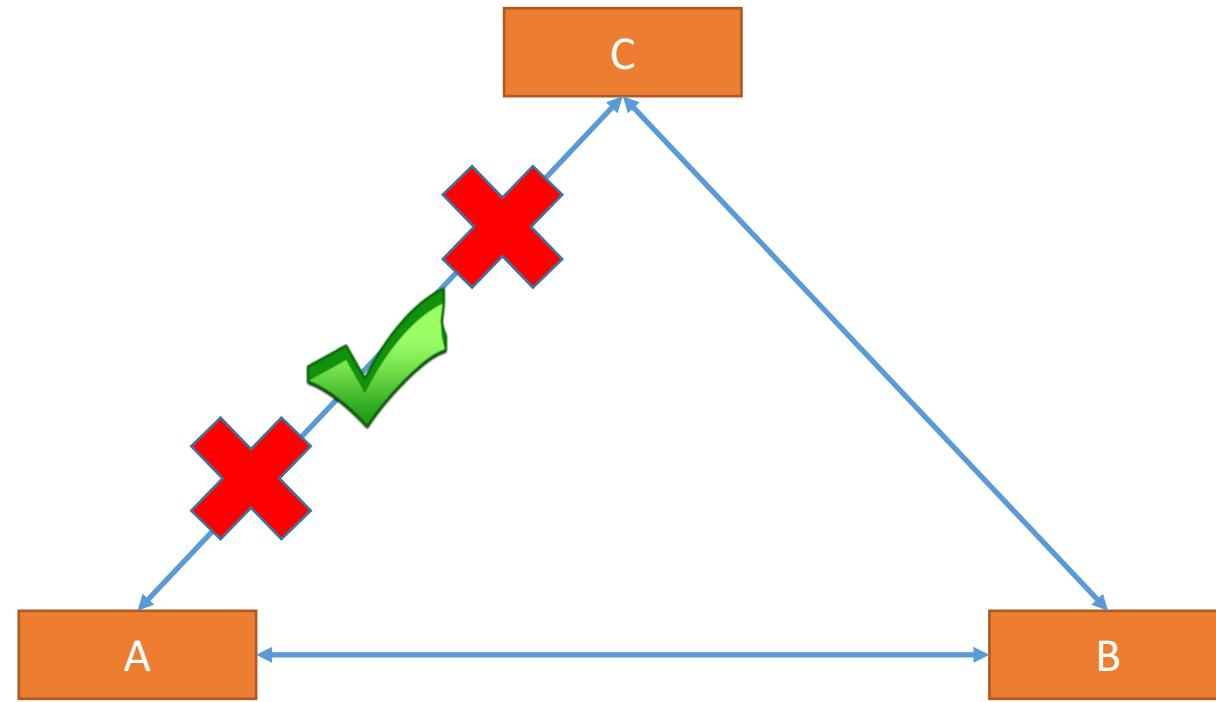
如何避免短暂的网络异常？

- 标记为“嫌疑人”
- 需要再经历一段时间
- 再标记为“死亡”



如何避免不稳定节点？？

- 发现异常：混乱度++
- 恢复正常：混乱度--
- 混乱度>阈值
- 询问身边k个伙伴
- 达成一致则驱除一段时间

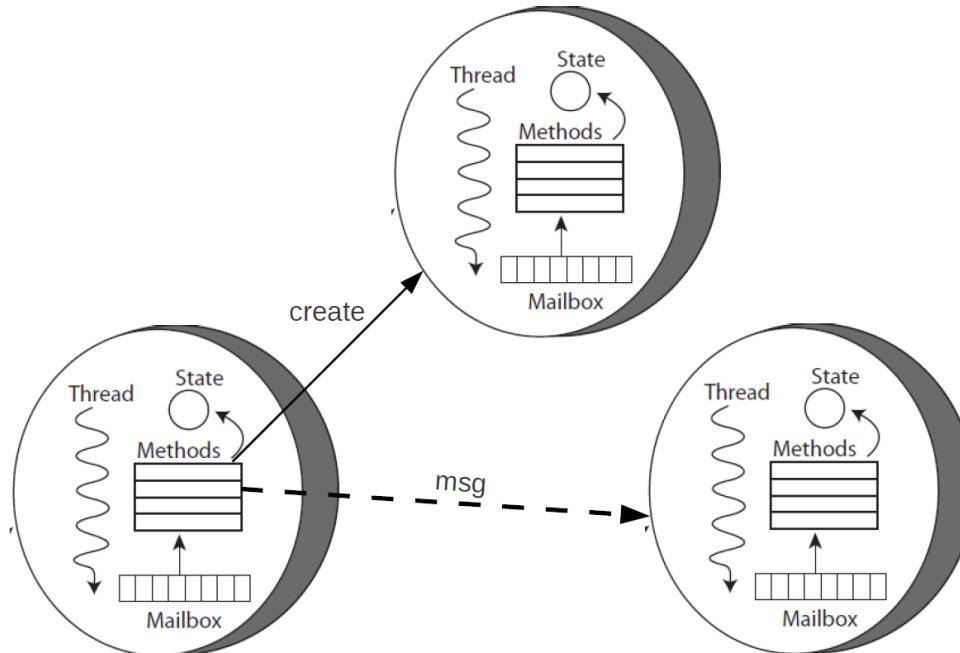


如何保证数据不丢？

- 复制
 - Hash(Data+0)
 - Hash(Data+1)
 - Hash(Data+2)

如何处理消息？

- 每个节点有个收件箱
 - 保存收到的所有消息
- 顺序处理每个消息
 - IF 消息属于自己
 - THEN 处理消息
 - ELSE
 - 转发消息
- 这就是Actor模型



REF: <http://yanjiankang.cn/2015/07/26/actor-model-research/>

总结

- 初始列表连成环，信息摘要四处窜
- 确定失败和混乱，不要忘了问伙伴
- 数据不丢多哈西，演员模型最耐看

References

- <https://www.youtube.com/watch?v=vujVmugFsKc>
- <http://www.infoq.com/presentations/uber-market-platform>
- https://docs.google.com/presentation/d/1HI4KapfAENAOf4gv-pSngKwvS_jwNVHRPZTTDzXXn6Q
- <https://www.youtube.com/watch?v=OQyqJWQHp3g>

Review: SNAKE

- **S**cenario: case/interface
 - Enumerate & Sort
- **N**ecessary: constrain/hypothesis
 - Ask & Predict
- **A**pplication: service/algorithum
 - Replay & Merge
- **K**ilobit: data
 - Append & Choose
- **E**volve
 - Analyze & Go back



One more thing

- Payment~

CS103 Schedule

Date (PST)	Time (PST)	Topic
11月12日	19:00	How to design Uber
11月15日	19:00	How to design Big Data Platform
11月19日	19:00	How to design GFS/BigTable/MapReduce
11月22日	19:00	How to design WhatsApp/WeChat
11月26日	19:00	How to design LinkedIn
11月29日	19:00	How to design Netflix
12月03日	19:00	How to design Recommender System
12月06日	19:00	How to design TinyURL

Join in the class

- 报名: <https://www.bittiger.io/>
- 咨询: xiuting@BitTiger.io
- 微信:



关注微信



官网: BitTiger.io

邮件: qinyuan@BitTiger.io