



# 用 Go 打造 Grab 的路径规划和 ETA 引擎

胡泊

Grab  
bo.hu@grab.com



探探 Gopher China 2019

# 大纲

- **Who's Grab**
- 团队角色
- 路径规划和 ETA 引擎的构建和演进
- **Go in Grab**

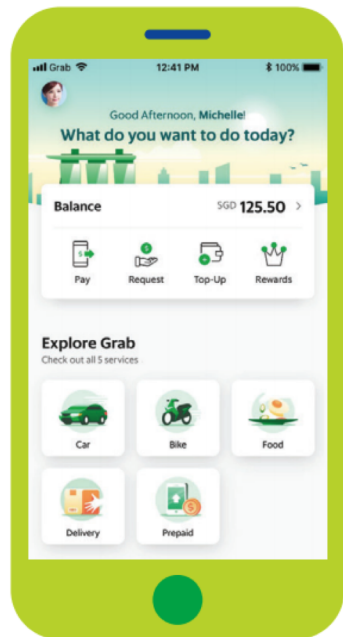


# Who's Grab



# 东南亚的超级APP

Grab是东南亚领先的超级应用程序，为消费者提供最重要的日常服务。通过开放式平台战略，Grab与合作伙伴为数百万名东南亚消费者提供安全、方便及经济实惠的交通、送餐、快递、杂货配送、线上支付和金融服务。



## 带领东南亚前进



Grab 让东南亚消费者更专注于对他们重要的事情，借此服务社区

超过  
**1.44 亿**  
的移动下载量



我们的网络覆盖超过  
**900 万**  
微型企业家

## 336 个城市\* 8 个国家

\*包括印尼各城市和统治区，以符合政府对各省作出二级行政区划的定义



我们为消费者提供不可缺少的日常服务

我们提供安全、方便和经济实惠的出行服务



# 大纲

- Who's Grab
- 团队角色
- 路径规划和 ETA 引擎的构建和演进
- Go in Grab



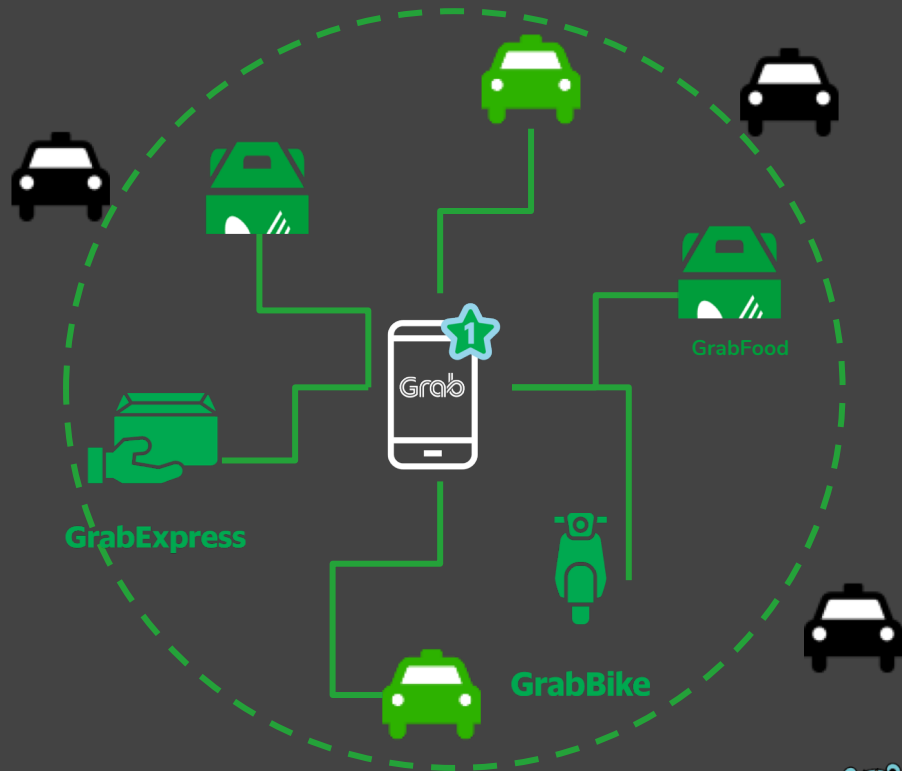
# 团队角色

- 路径规划  
给定起终点在连通图中寻找合理路线
- ETA(Estimated Time of Arrival)  
给定路线预估行驶时间



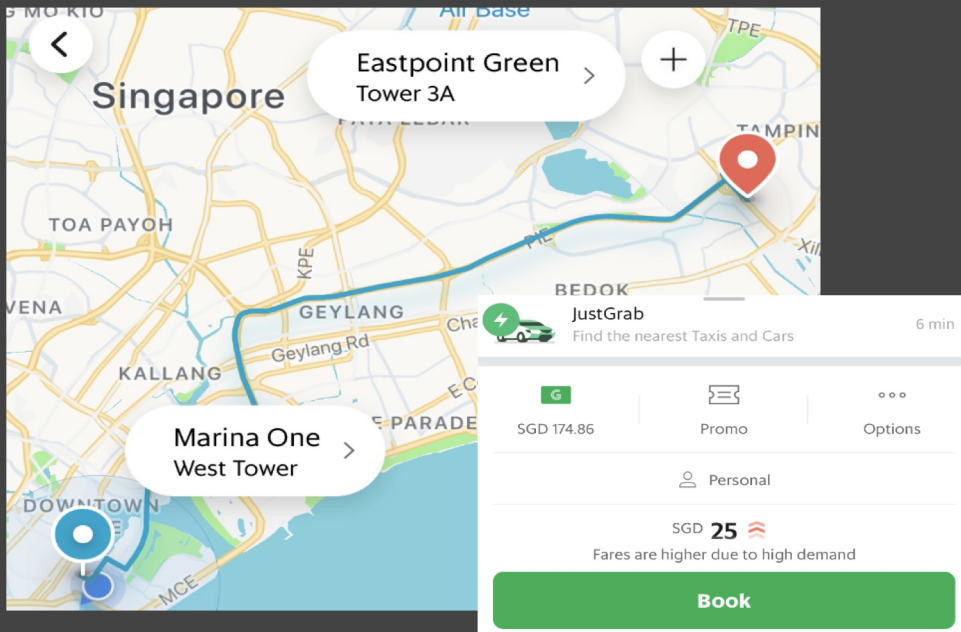
# 业务场景

- 派单
- 计价
- 司乘体验



# 业务场景

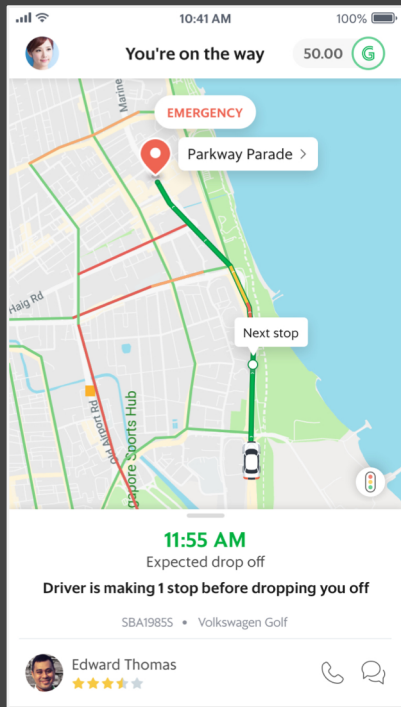
- 派单
- 计价
- 司乘体验





# 业务场景

- 派单
- 计价
- 司乘体验



# 大纲

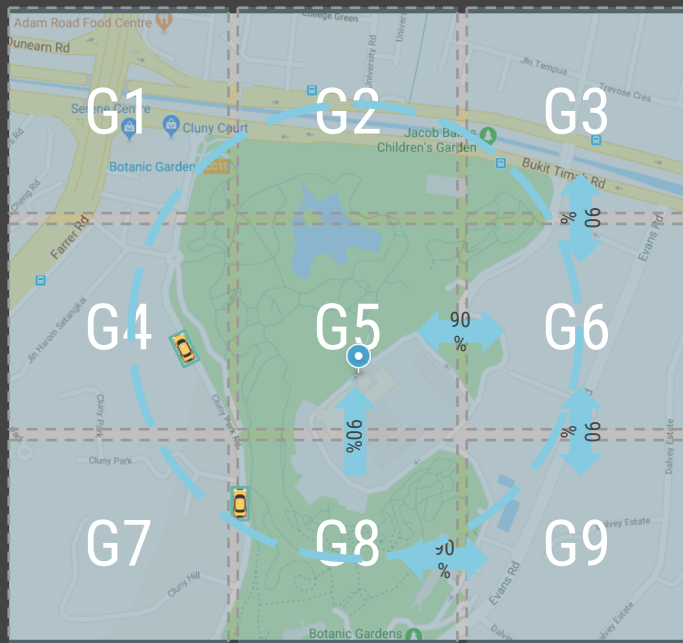
- Who's Grab
- 团队角色
- 路径规划和 ETA 引擎的构建和演进
- Go in Grab



# 路径规划和ETA引擎的构建和演进



# 技术选型 - 概率 vs 路网



# 路径规划和ETA引擎的构建和演进

- 地图
- 司机定位
- 路况
- 图搜索
- ETA模型调优



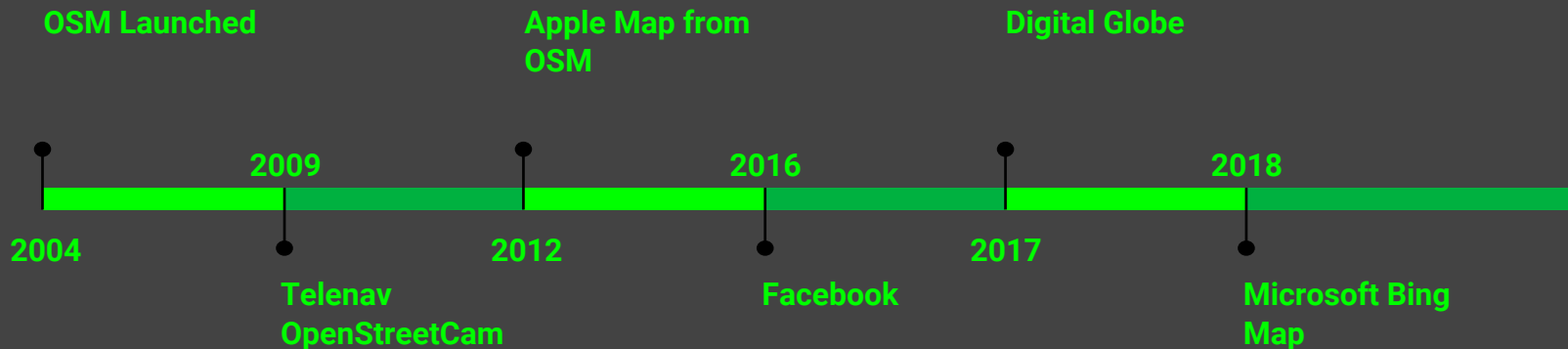
# 地图 - 开源地图OSM



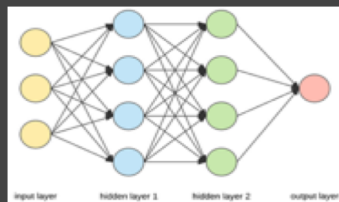
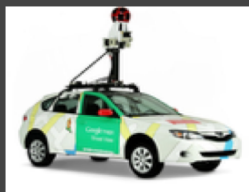
<https://www.openstreetmap.org>



# 地图 - OSM发展历程



测绘车  
成本高昂

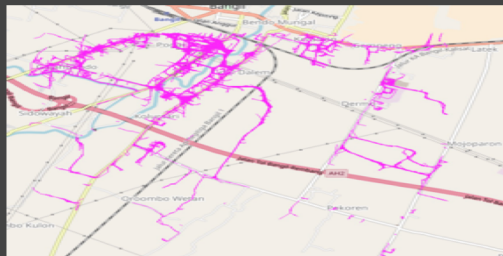


AI  
成本低廉



# 地图 - 路网学习

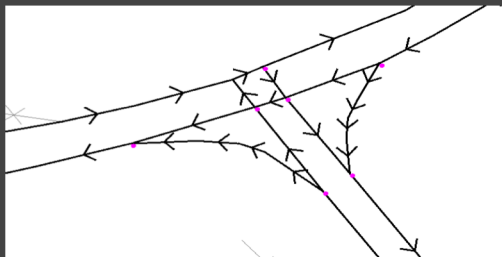
- 学习对象
  - 路网形状及拓扑
  - 属性：单行，禁行
- 数据源
  - GPS
  - 卫星图
- 方法论
  - 计算机视觉
  - 非监督学习
  - 深度学习



GPS



卫星图



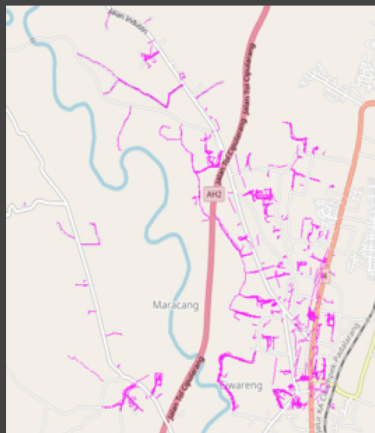
属性



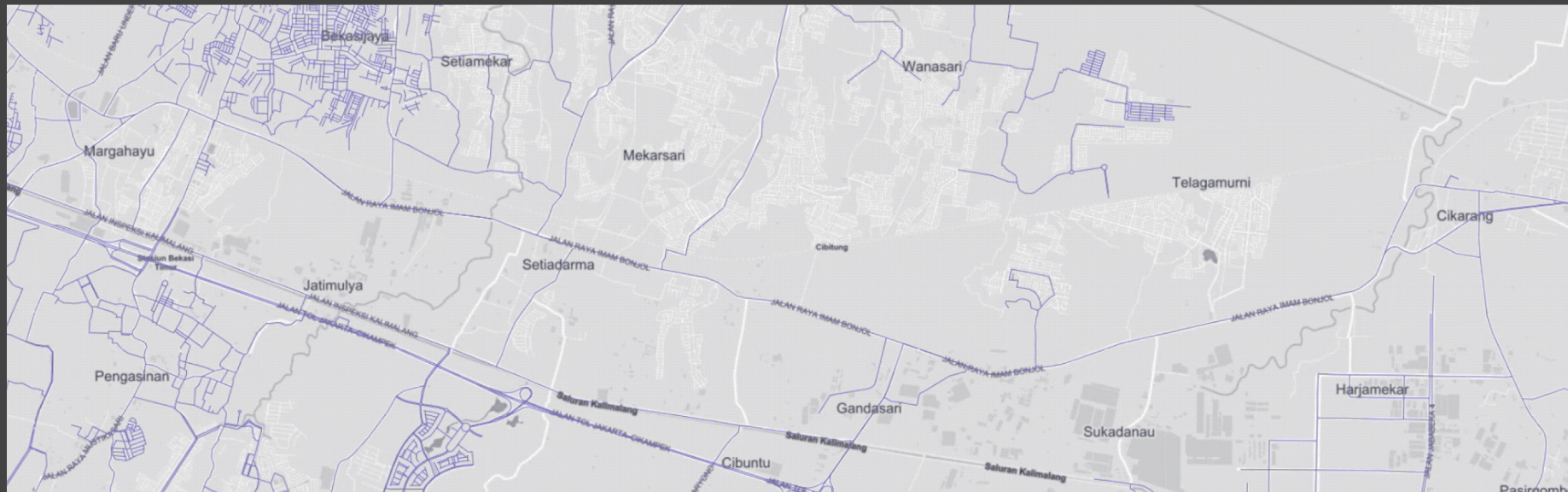


# 地图 - 路网学习

- 学习对象
  - 路网形状及拓扑
  - 属性：单行，禁行
- 数据源
  - GPS
  - 卫星图
- 方法论
  - 计算机视觉
  - 非监督学习
  - 深度学习



# 地图 - Jakarta Before and After



2016 **19,825** Kms

2019 **55,692** Kms

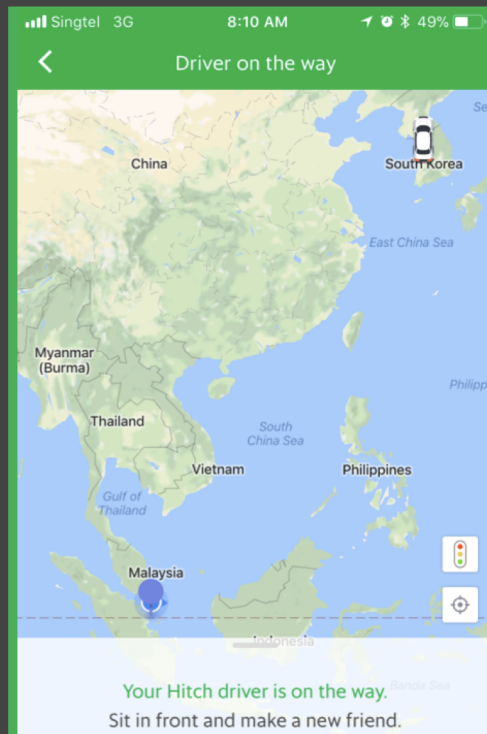


# 路径规划和ETA引擎的构建和演进

- 地图
- 司机定位
- 路况
- 图搜索
- ETA模型调优

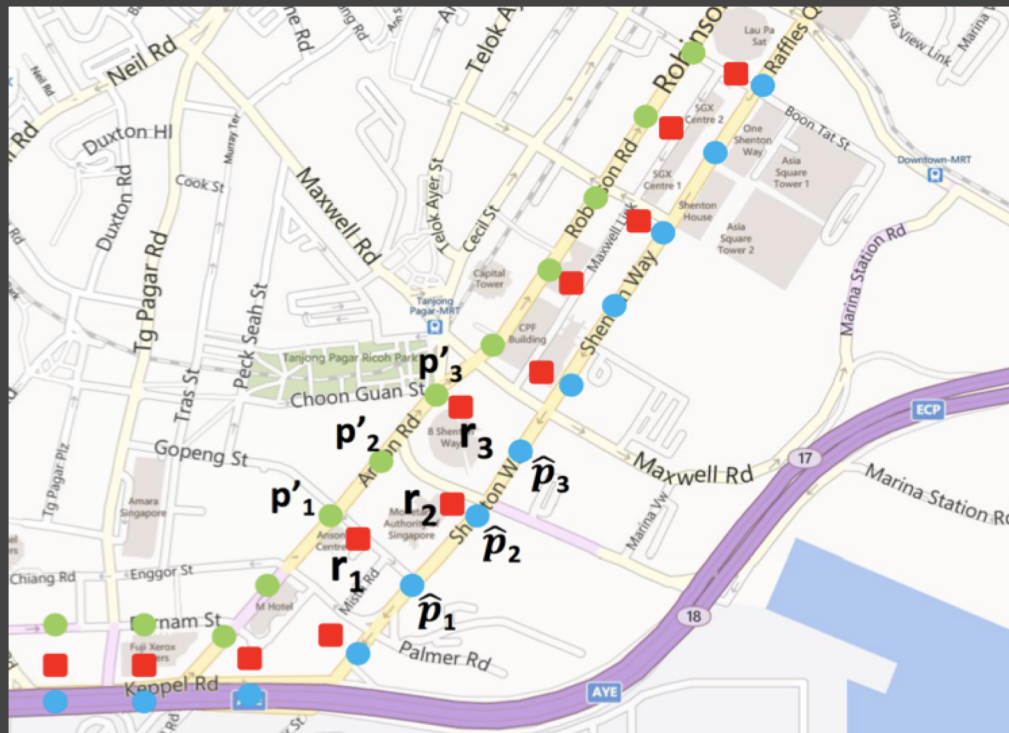


# 司机定位 - GPS误差



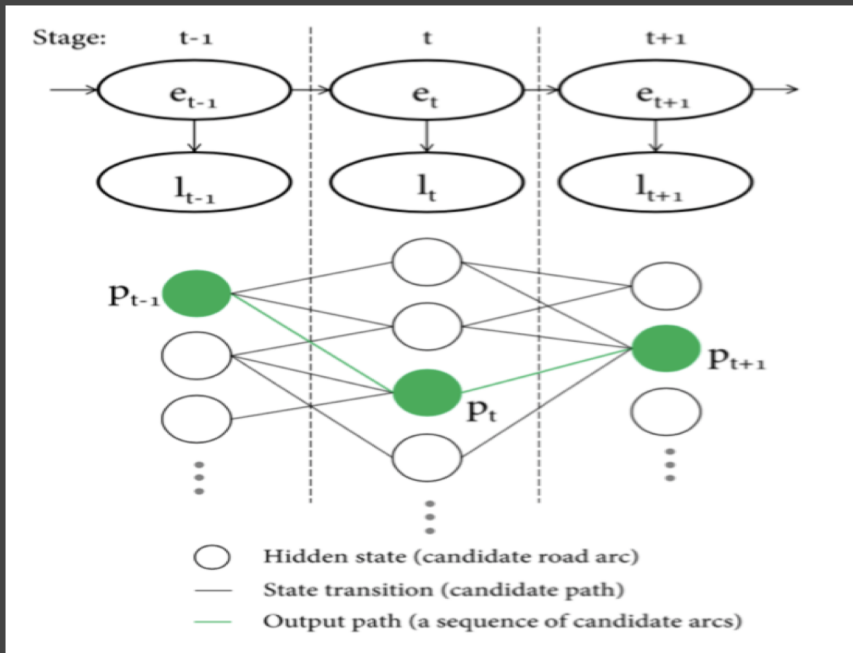
# 司机定位 - GPS误差

红色的司机到底走了哪条路？



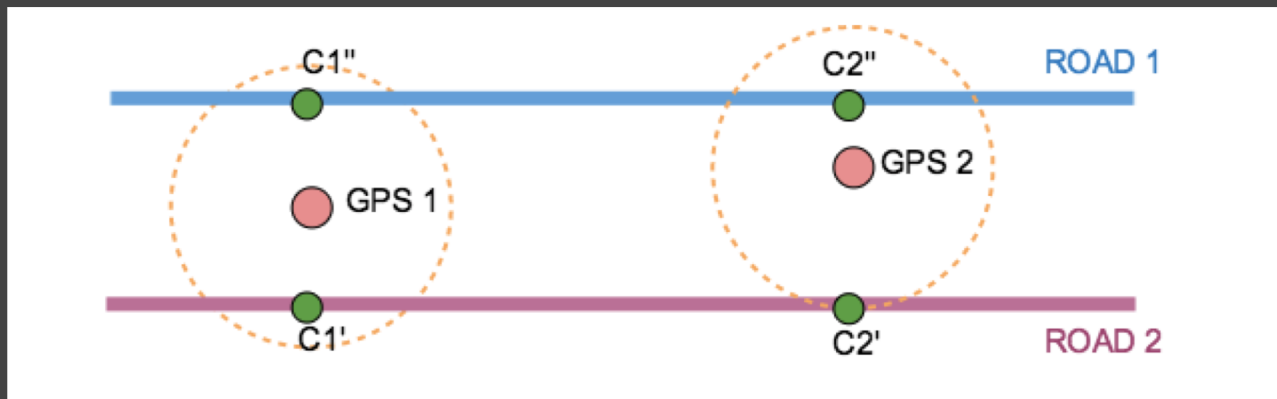
# 司机定位 - Map Matching (HMM)

- 隐藏状态 vs 观测状态
  - 真实位置 vs GPS
- 发射概率
- 转移概率
- [Hidden Markov Map Matching Through Noise and Sparseness](#)



# 司机定位 - Map Matching (HMM)

- 寻找投射点
  - RTree Nearest Search
- 发射概率
- 转移概率

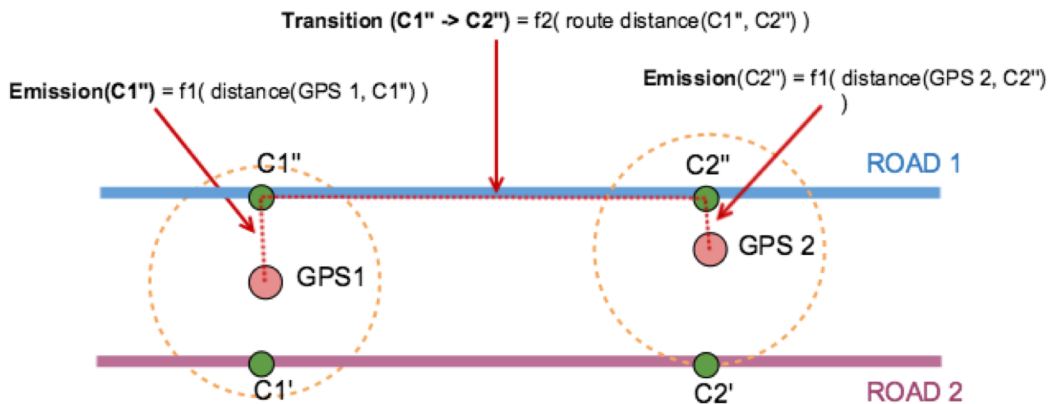


# 司机定位 - Map Matching (HMM)

- 寻找投射点
  - RTree Nearest Search
- 发射概率
  - 高斯分布
- 转移概率
  - 指数分布

$$p(z_t | r_i) = \frac{1}{\sqrt{2\pi}\sigma_z} e^{-0.5 \left( \frac{\|z_t - x_{t,i}\|_{\text{great circle}}}{\sigma_z} \right)^2}$$

$$p(d_t) = \frac{1}{\beta} e^{-d_t/\beta}$$



MAX of {

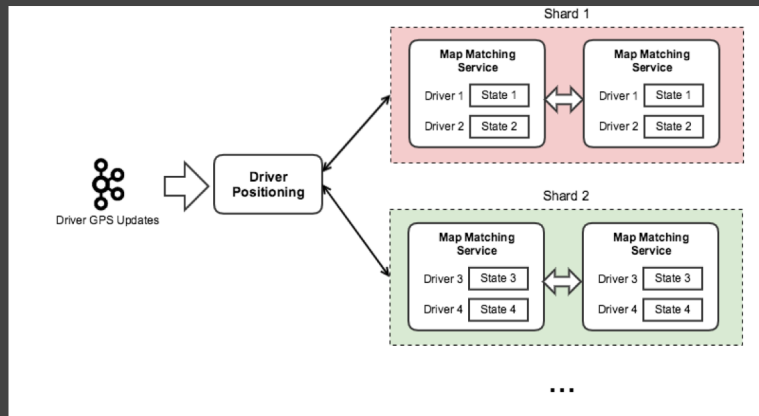
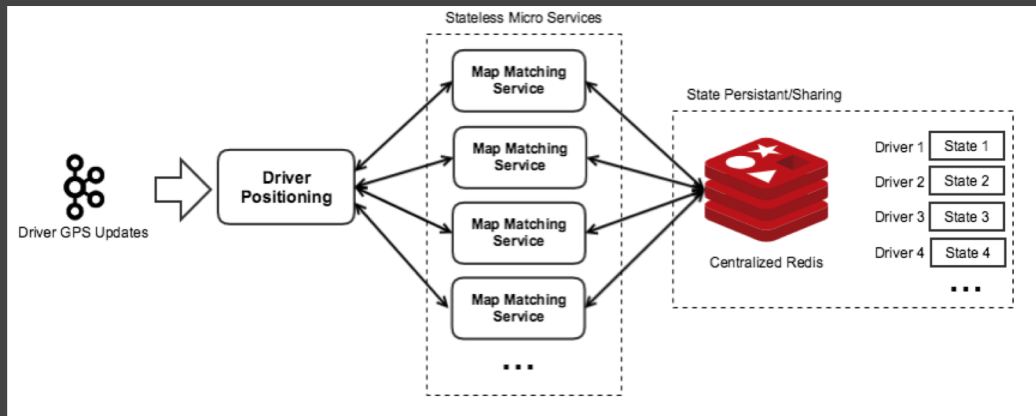
- Probability(C1'' -> C2'') = Emission(C1'') \* Emission(C2'') \* Transition (C1'' -> C2'')
- Probability(C1' -> C2'') = Emission(C1') \* Emission(C2'') \* Transition (C1' -> C2'')
- Probability(C1'' -> C2') = Emission(C1'') \* Emission(C2') \* Transition (C1'' -> C2')
- Probability(C1' -> C2') = Emission(C1') \* Emission(C2') \* Transition (C1' -> C2')





# 司机定位 - 面临的挑战

- 算法
  - 低速、路口和GPS偏差过大
- 工程
  - 如何管理带状态的数据
  - 如何在微服务间传播海量数据



# 司机定位 - 数据压缩

- 剪枝
  - 快速丢弃状态为 -INF 的状态
- 增量存储
  - [Encoded Polyline Algorithm Format \(Google\)](#)
- encoding/gob
  - Golang原生支持
  - 无需额外schema定义
  - Serialize/Deserialize速度快



# 路径规划和ETA引擎的构建和演进

- 地图
- 司机定位
- 路况
- 图搜索
- ETA模型调优





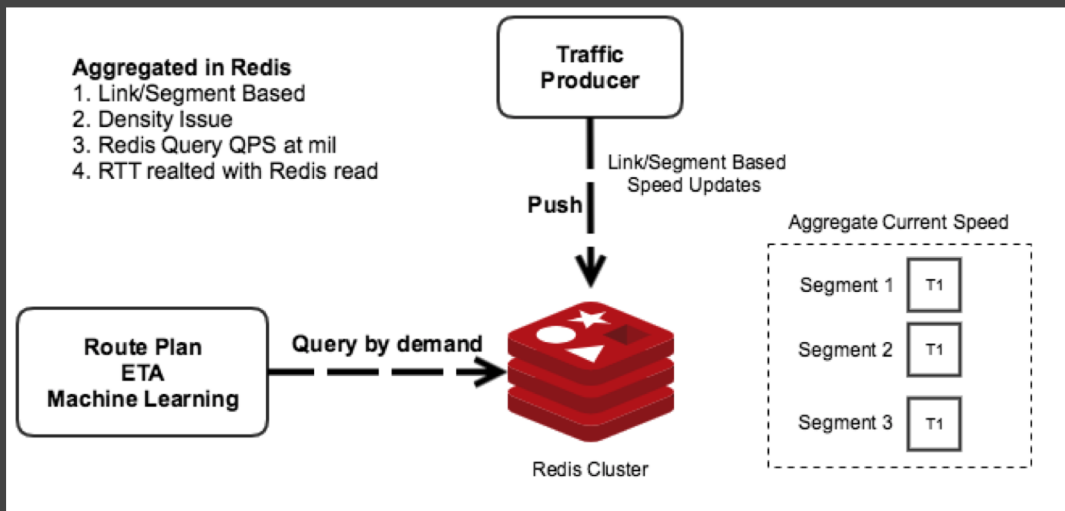
# 路况 - 挑战和演进

- 挑战
  - 超高的读QPS
- 演进
  - V1 - 计算存储分离 (centralized Redis)
  - V2 - 计算存储合并的缓存层 (lock free ring buffer)
  - V3 - spark streaming接管数据聚合和分发



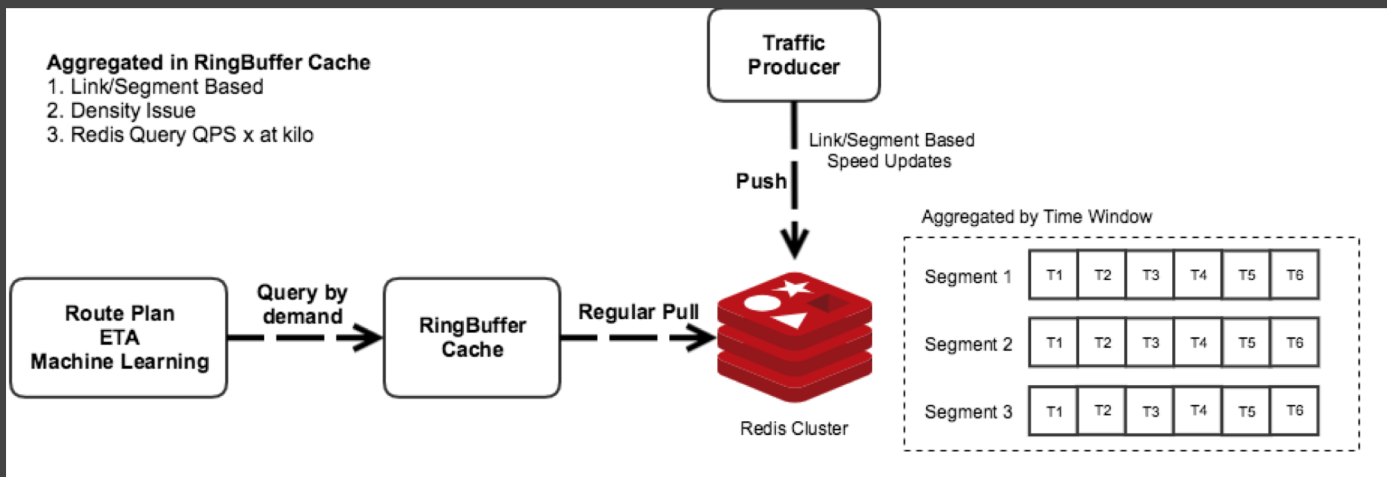
# 路况 - 演进

- V1 - Redis Aggregation
  - 问题：Read QPS高，容易Spike



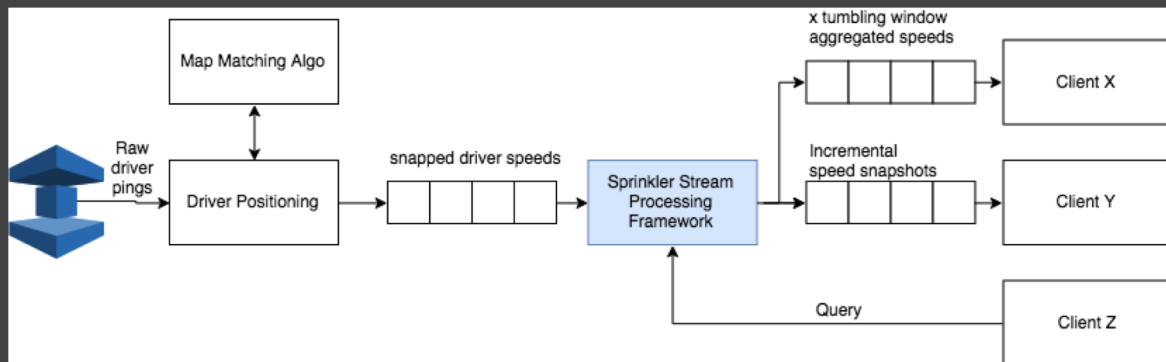
# 路况 - 演进

- V2 - Redis + Cache + Time Window
  - 问题：内存及CPU占用高
  - 问题：数据一致性没有保证



# 路况 - 演进

- V3 - 流式处理
  - 聚合多样化
  - 增量分发
  - 轻量客户端





# 路径规划和ETA引擎的构建和演进

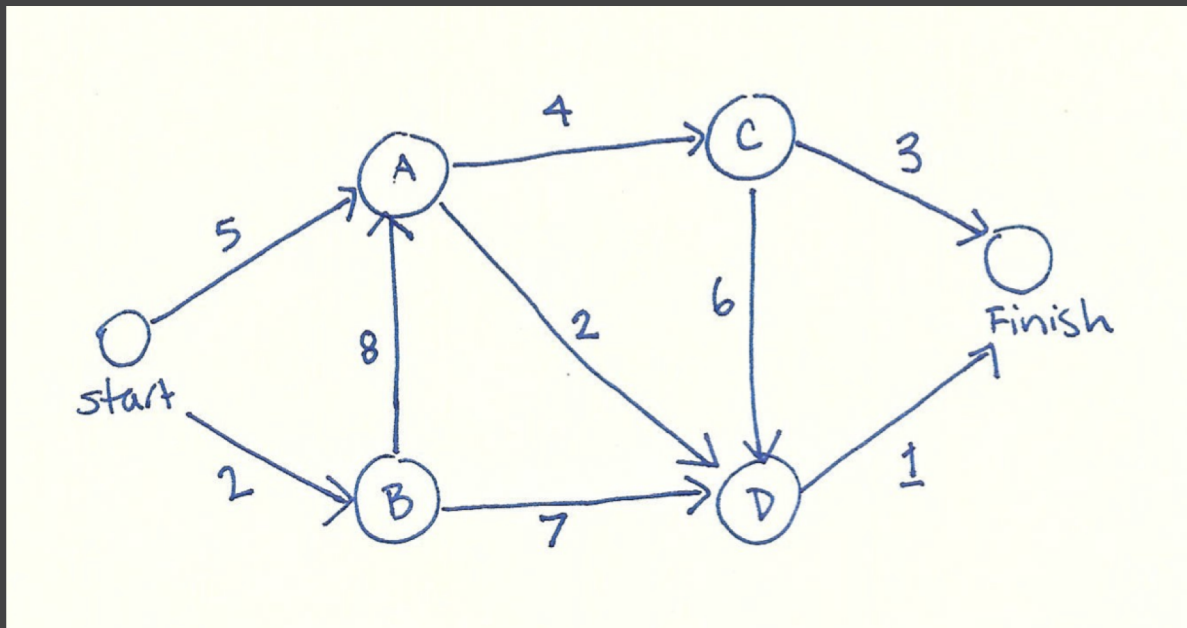
- 地图
- 司机定位
- 路况
- 路径规划
- ETA模型调优



# 路径规划 - 选择什么图搜索算法？

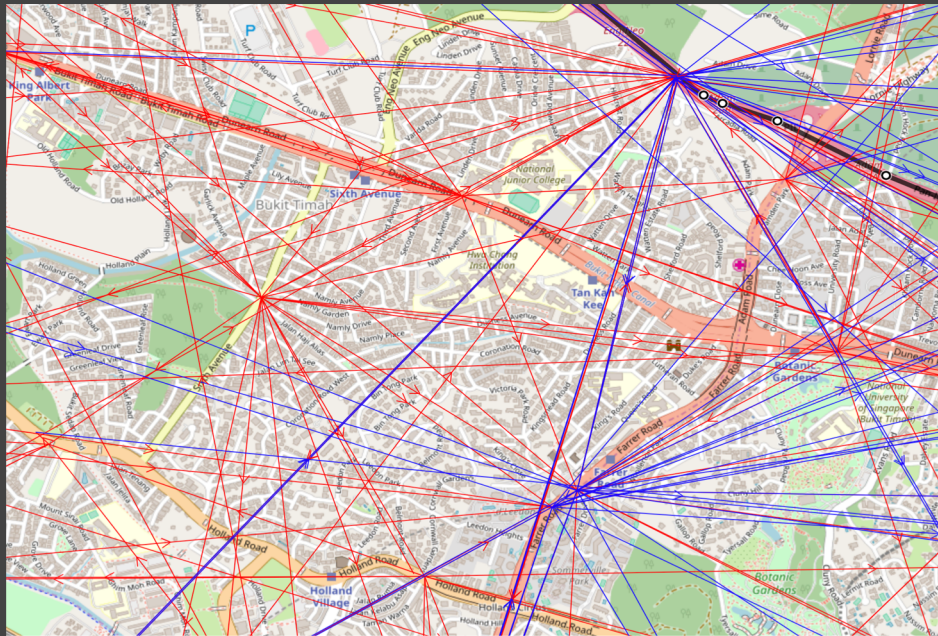
Dijkstra ?

远远无法扛住业务流量



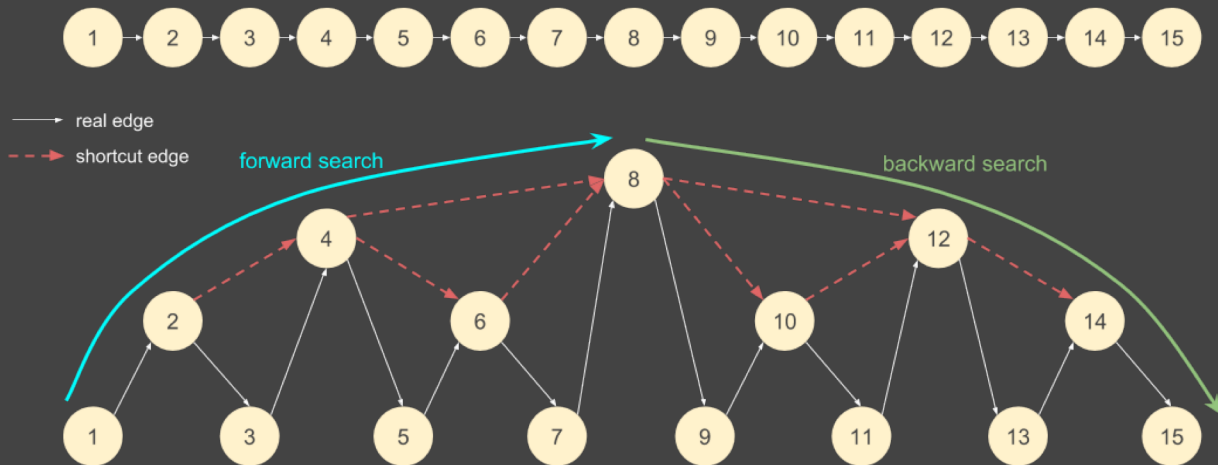
# 路径规划 - Contraction Hierarchies (CH)

- Trade off: 预处理时间 vs 查询时间
- “Basic idea: far away from source / target only use “important” roads (think of highways)”



# 路径规划 - Contraction Hierarchies (CH)

- 分层和压缩
- 双向Dijkstra



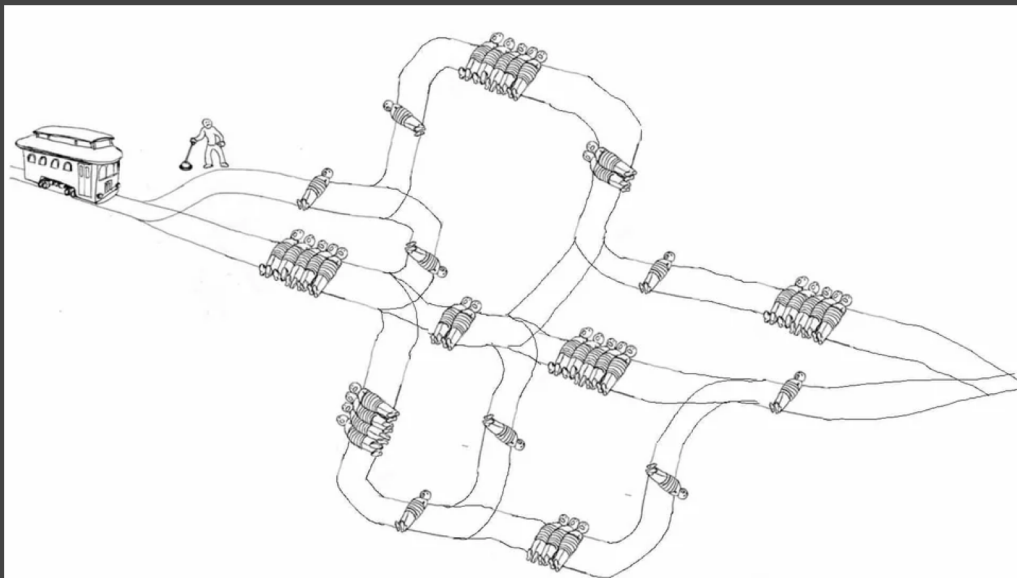
# 路径规划 - 性能优化

- 内存优化
  - 重用 slice/map 避免频繁内存分配和回收: 比如 Dijkstra 用到的 priority queue
  - 优化图的数据结构, flatten成数组存储
- 性能优化
  - 谨慎使用 interface: Geo中通常会使用到两种坐标, WGS84 和 Web Mercator, 虽然使用 interface 将他们统一起来看起来不错, 但是会消耗额外的空间和 runtime conversion 性能



# 路径规划 - 图权重调优

- 最快路 != 最佳路
  - 司机偏好
- 指标
  - 偏航率
- 运筹学方法全局最优解
  - 变量
  - 优化



The moral implications of implementing shortest path algorithms



# 路径规划和ETA引擎的构建和演进

- 地图
- 司机定位
- 路况
- 路径规划
- **ETA模型调优**

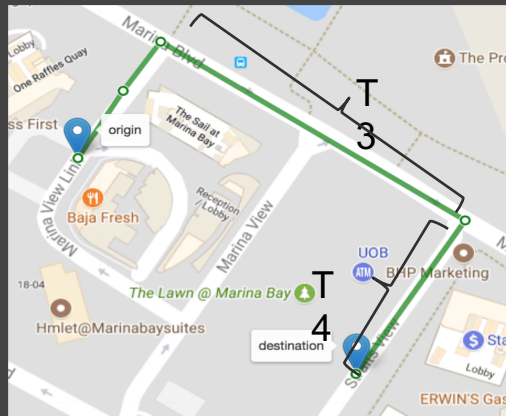


# 为什么需要调优?





# ETA - 机器学习调优



特征提取

$$ETA = T1 + T2 + T3 + T4$$

路径规划

路径距离

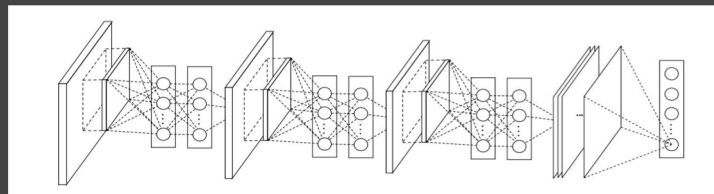
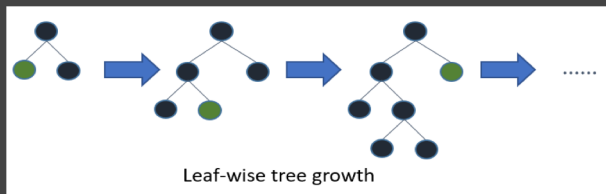
红绿灯个数, 转弯个数

地域特性: CBD

New ETA

预测

机器学习模型



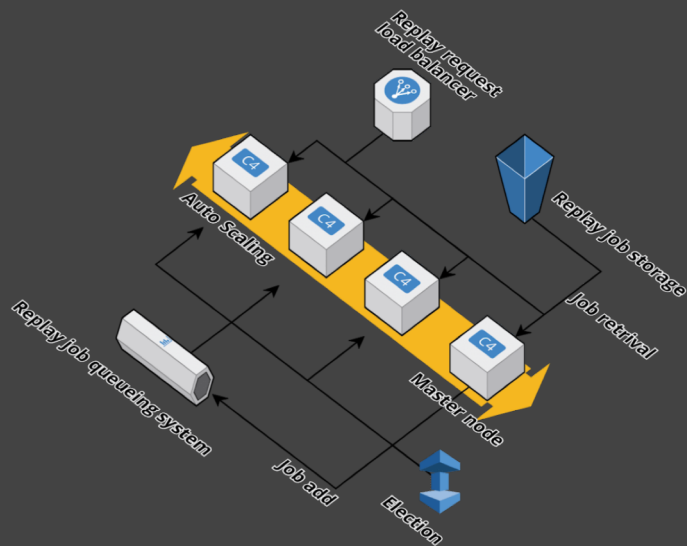
# ETA - 数据

- 训练数据准备
  - Data lake
  - Data warehouse
  - Geo-computed features
- 线下千亿级数据计算



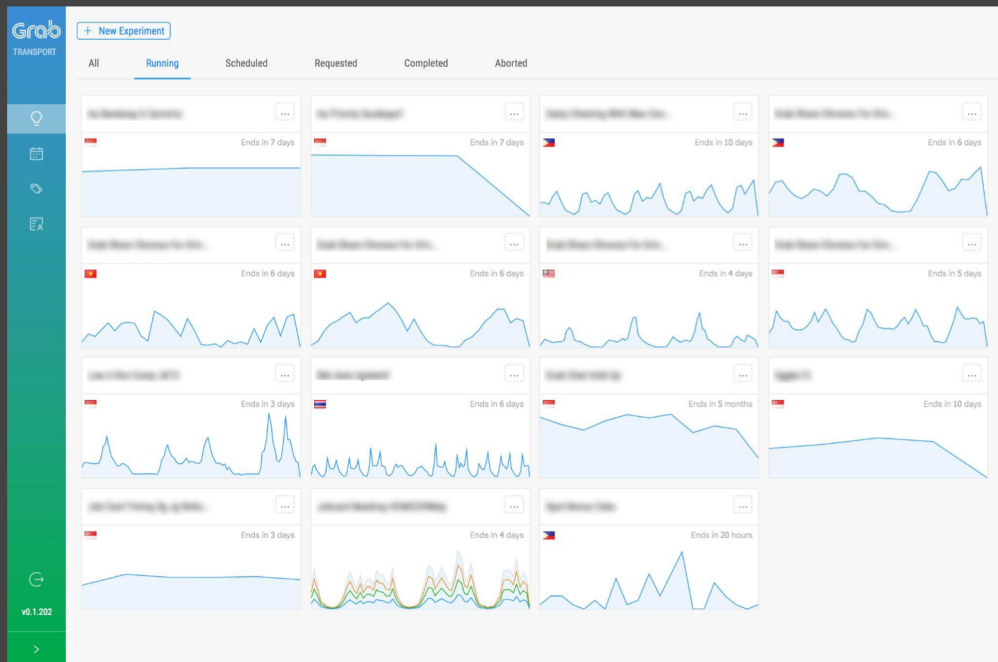
# ETA - Go实现分布式重放系统

- 重用线上代码
- 低维护成本
- 负载均衡
- 高并发
- 业务场景定制
- 支撑千亿级训练数据计算



# ETA - 试验平台 Grab-X

- 数据驱动
- 定义系统关键指标
- 创建控制组和对比组的配置
- 生产环境进行试验



# 大纲

- Who's Grab
- 团队角色
- 路径规划和ETA引擎的构建和演进
- Go in Grab



Go in Grab



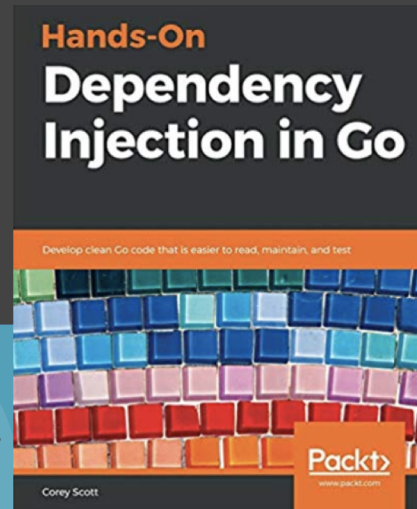
# Go in Geo

- [golang/geo](#)
- [qedus/osmpbf](#)
- [go-spatial/tegola](#)
- [tidwall/buntdb](#)
- [lukeroth/gdal](#)
- [jonas-p/go-shp](#)
- [llgcode/draw2d](#)



# Go in Grab

- 工具
  - GrabKit
  - Gandalf
  - Golang Camp
- 文化
  - Drive SEA Go Go Go
  - book





# We're hiring

Say NO to 996: <https://github.com/formulahendry/955.WLB>



bo.hu@grab.com



# Thanks

感谢聆听!



[bo.hu@grab.com](mailto:bo.hu@grab.com)

