促进软件开发领域知识与创新的传播

InfoQueue

QCon
全球软件开发大会

[上海] 2015年10月15-17日

ArchSummit
全 球 架 构 师 峰 会

[北京] 2015年12月18日-19日

关注InfoQ官方微信
及时获取ArchSummit演讲视频信息

ArchSummit全球架构师峰会深圳站2015

基于地理位置的实时搜索引擎

易到用车 余庆

个人简介

- 互联网技术老兵，热爱开源
- 曾任职于新浪、雅虎中国和淘宝
- 现在易到用车担任架构师
- 开源分布式文件系统FastDFS作者
- 参与过Apache Traffic Server改进

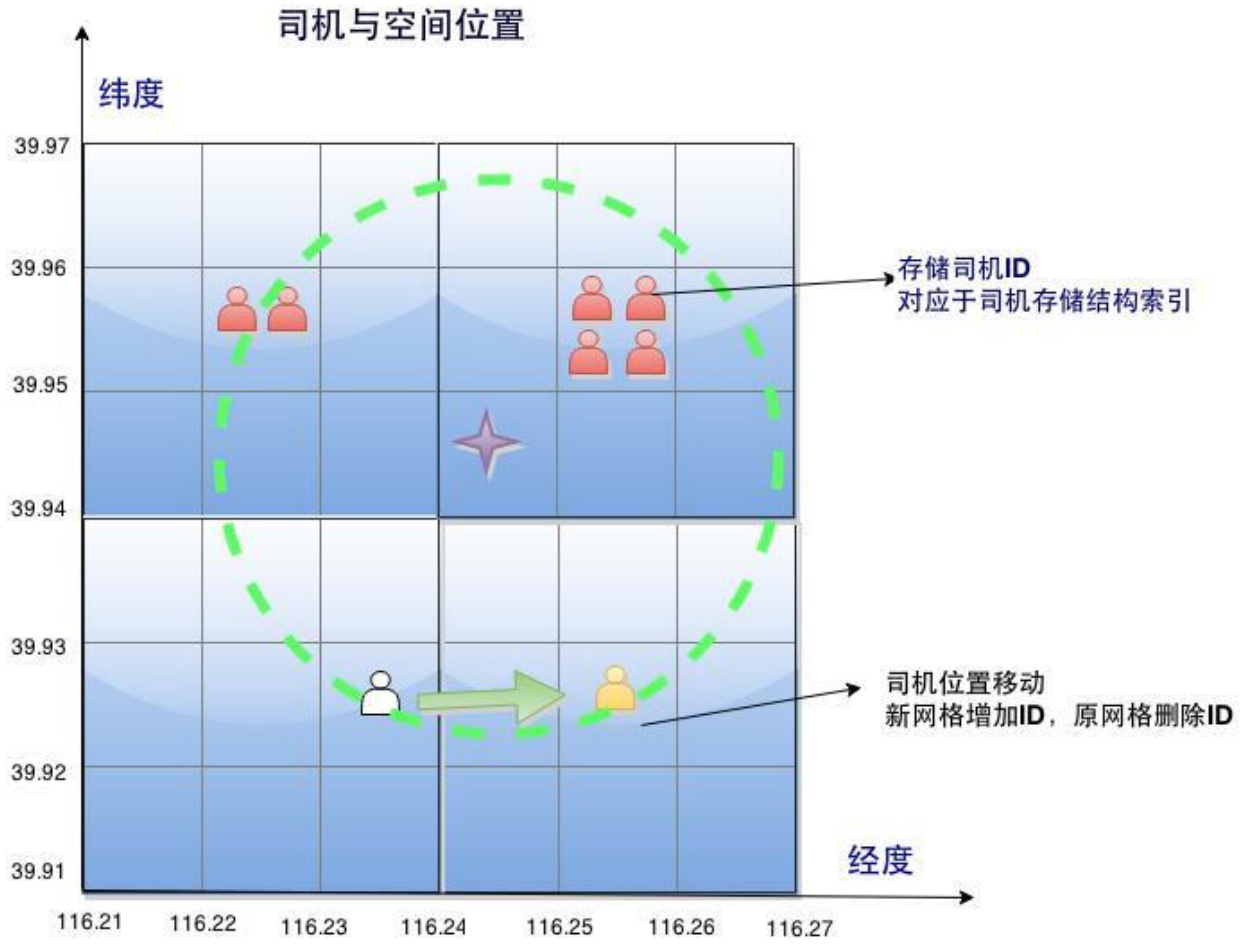
演讲提纲

- O2O行业的地理位置搜索问题
- 易到用车的解决方案历程
- 易到基于地理位置的实时搜索引擎

O2O行业的地理位置搜索问题

- 搜索附近的人和物
- 本质就是搜索引擎问题
- 空间索引通常采用网格化方式

空间位置网格化


司机与空间位置
纬度
39.97
39.96
39.95
39.94
39.93
39.92
39.91
116.21
116.22
116.23
116.24
116.25
116.26
116.27
经度
存储司机ID
对应于司机存储结构索引
司机位置移动
新网格增加ID，原网格删除ID

网格编码方式

- GeoHash
- 自定义编码

GeoHash

- GeoHash将二维的经纬度转换成字符串
- 字符串相似的表示距离相近，可以采用字符串前缀匹配
- 最大的缺点就是突变性，有些编码相邻但距离却相差很远
- 除了使用定位点的GeoHash编码进行匹配外，还使用周围8个区域的GeoHash编码

自定义编码

- 可以编码为整数：int(经度 / D) * (10 ^ P) + (纬度 / D)
- D为一个格子对应度数，例如1KM对应的度数为 0.008998
- P为(经度 / D)的最大位数，例如5
- (116.388055, 39.907500) => 3293914437

附近位置检索方式

- 按正方形搜索，搜索正方形中包含的格子即可
- 经度和维度两个方向，组合出搜索的格子ID
- 若一个格子纬度方向为1KM，经度方向为1KM * cos(纬度)
- 检索附近N公里，遍历的格子数：2N * (2N / cos(纬度))

该方案问题及对策

- 搜索格子过多

1. N值较大：使用更大的格子
2. cos(纬度)较小，即纬度较大

南极圈和北极圈度数66.5度，cos值约为0.40

限制搜索格子数

易到用车的解决方案历程

- 第一阶段基于数据库查询
- 第二阶段基于位置缓存和检索
- 第三阶段基于搜索引擎

基于数据库查询方案

- 直接拼写SQL
- 优点：简单，同时支持距离检索和其他查询条件
- 缺点：

  数据更新存在瓶颈，查询性能一般

  不具备水平扩展性

基于位置缓存和检索方案

- 使用redis缓存司机信息和位置格子中的司机ID
- 优点：实现简单，位置索引更新较快
- 缺点：网络IO较高

  索引和数据分离，需要两次查询

  redis并不适合条件查询场景

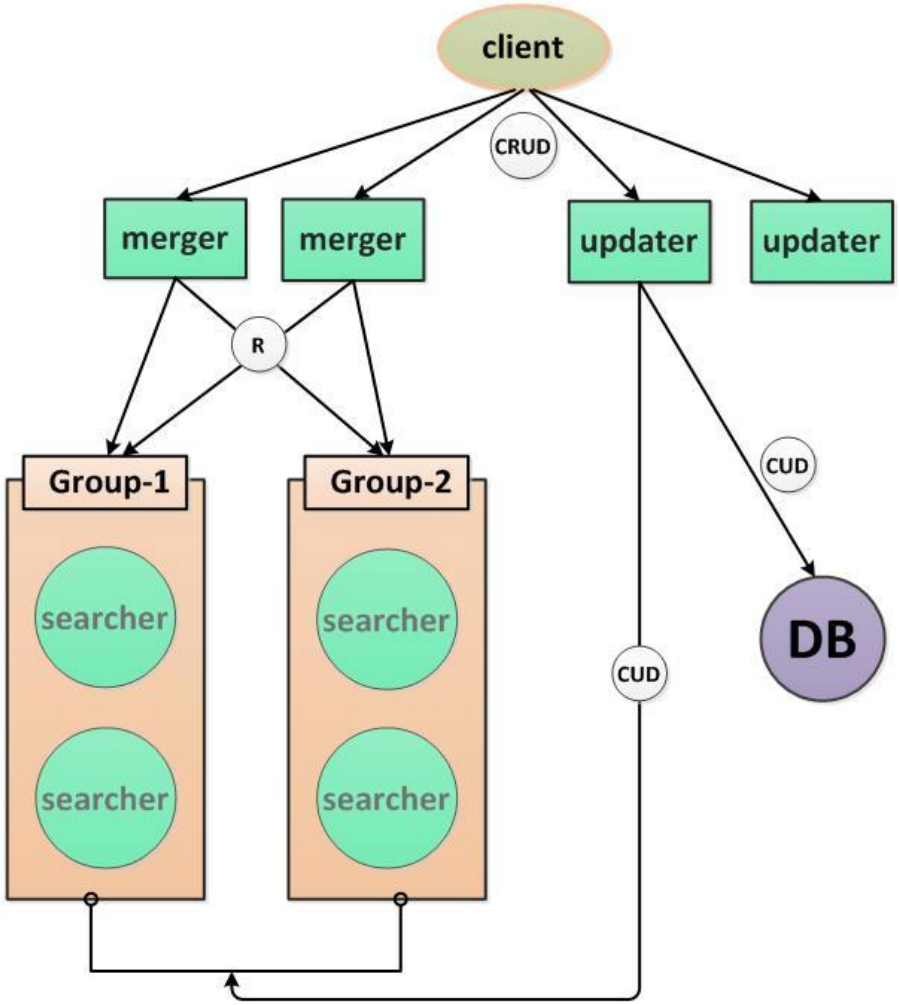
基于传统搜索引擎方案

- 尝试过solr和ElasticSearch，基本满足要求
- 实时更新是传统搜索引擎的短板

我们对搜索引擎的要求

- 需要支持大量实时更新，司机当前位置并不需要持久化
- 复杂和灵活的rank逻辑，数据和计算不应分开
- 我们需要自己的实时搜索引擎

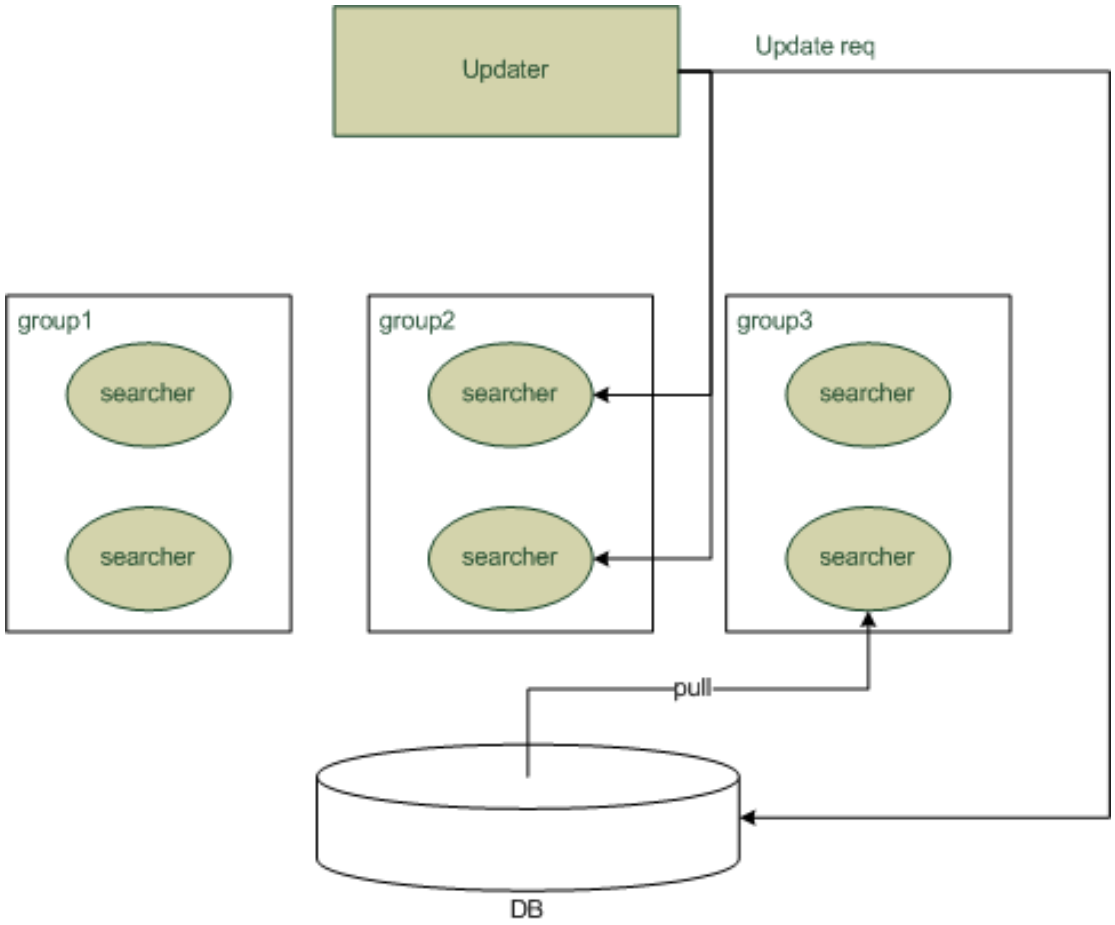
易到实时搜索引擎架构


client
CRUD
merger
merger
updater
updater
R
Group-1
Group-2
searcher
searcher
searcher
searcher
CUD
CUD
DB

各个角色职责

- searcher：数据索引和搜索服务
- updater：负责数据更新
- merger：转发查询到一行searcher，收集、合并查询结果
- DB：存储文档

索引加载和更新机制


Updater
Update req
group1
searcher
searcher
group2
searcher
searcher
group3
searcher
searcher
pull
DB

如何保证数据一致性

- searcher启动时通知updater准备向其更新数据
- searcher从DB中加载数据
- 加载数据完毕，通知updater向其更新数据
- update持续把更新请求转发给searcher

位置索引

- 使用hash table存储位置格子
- 格子中存储司机列表，采用双向链表
- 检索附近N公里的司机，遍历正方形中的所有格子
- 遍历的格子数：2N * (2N / cos(纬度))

架构特点

- doc统一存储，searcher不持久化存储数据和索引
- 不保证数据强一致，只保证数据最终一致
- searcher采用分组方式，简单可控
- 冗余设计无单点

技术实现特点

- 量身定做，耦合业务逻辑
- 性能上追求极致

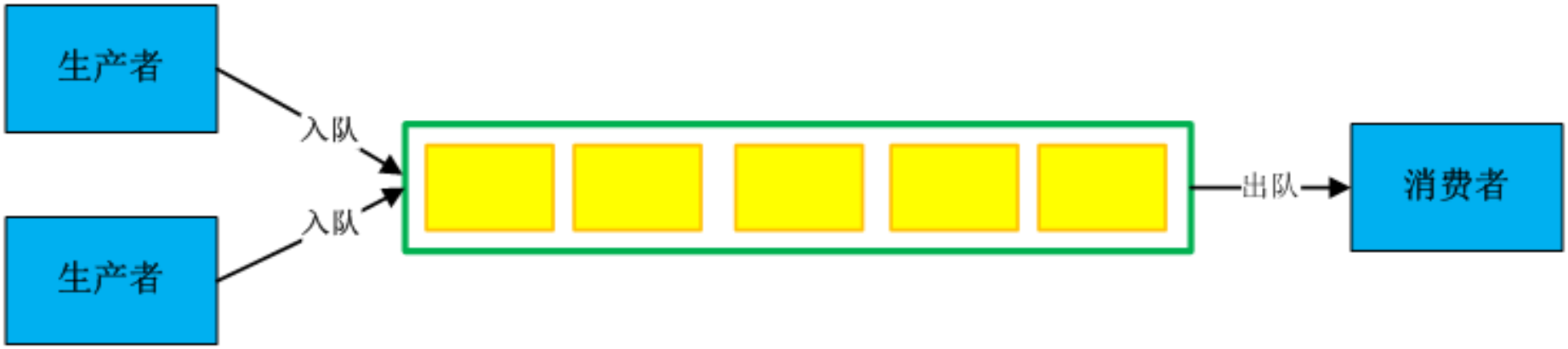
1. 采用高性能服务化框架
2. 无锁化：索引读写无锁，延迟释放机制
3. thread local：pthread_getspecific?
4. 使用msg-pack实现数据序列化
5. 使用内存池、buffer循环使用

如何实现索引无锁化

- 一写多读线程模型
- 使用无锁数据结构：hash table

如何做到只有一个线程更改索引

- 对外支持并发数据更新
- 内部采用多生产者单消费者模型



性能压测结果

- 8核16G内存开发机
- 更新当前位置：2w+
- 查询司机列表：4K+

我的联系方式

- 微博：http://www.weibo.com/fastdfs
- 邮箱：yuqing@yongche.com

Thanks!