摘要:

2017年7月，我参与某互联公司自主研发全国运营的网约车出行平台。该平台主要是为了解决客户在线预约网约车。包含用户叫车、自动派单、司机接单、行程结算、车辆管理等功能模块。我在该项目中担任系统架构师设计师一职，负责系统架构设计、性能优化、核心功能模块开发等工作。本文以该项目为例，主要论述NoSQL数据库技术在该项目的具体应用。通过使用键值数据库（Redis）做内容缓存解决了项目的性能问题达到设计预期；通过键值数据库做分布式锁解决分布式事务状态不一致问题达到事务一致；通过键值数据库做高并发数据存储解决车辆位置查询需求达到了克服业务数据库性能的目的。事实证明，使用这些技术手段使得项目整体能够克服业务传统数据库瓶颈和应对大规模高并发场景。最终项目得以顺利完成，取得预期目标，获得用户好评。

正文:

2017年7月，我参加了某互联公司自主研发全国运营的网约车出行项目。 我公司致力于网络出行市场服务是首批获得当地“网约车出行牌照”, 并且先后获得四川、河南、海南、山东等各地的合法运营牌照背景下。以构建一个合法、合规安全的网约车出行环境为使命。以解决用户打车难、提升用户的出行幸福感，拉动了产业链的发展，构建了一个完整的出行生态链的为目标，构建一个全国性的网约车出行平台。 网约车出行平台包括乘客端、司机端、后台管理系统三部分组成。乘客端供乘客查询车辆、发布订单、支付车费、评论司机；司机端供司机车辆信息认证、出车接单、乘客接送、车费提现等；管理系统主要是提供系统报表查询、规则配置、乘客管理、司机管理、分公司管理、账务管理等。本平台提供了专车、快车、出租车业务，乘客可以通过自身的需求来选择不同类型业务出行，司机需要上传自身拥有的运营车辆等证照信息到平台审核，只有当平台审核通过后才能正常的再平台上合法运营。在该项目中本人担任系统架构师设计师一职，负责项目的架构设计以及软件开发的部分工作。

出行平台定位是一个全国性的互联网出行平台，如果用传统的关系型数据库会有很多难以克服的问题，于是我们决定采用NoSQL来解决大规模数据集合以及多种数据类型的挑战。NoSQL数据库的4大分类如下：1、键值（Key-Value）存储数据库，通用作内容缓存、处理高访问负载和日志系统。Key-Value的键值对数据模型底层使用hash table实现。具有查询速度快的优点。如：Redis、Oracle BDB。2、列存储数据库，通常用作分布式文件系统，以列簇形式存储。具有查找速度快、可拓展性强，容易分布式拓展的优点。如：HBase, Cassandra。3、文档型数据库，通常使用在Web应用中采用版本化文档数据模型以特定的半结构化文档格式（JSON）存储。如：MongoDb, CouchDB。4、图形数据库（Graph）通常使用在社交网络、推荐系统等采用图结构。如：Neo4J, InfoGrid。通过以上的分析和项目组内的讨论我们决定采用键值存储数据库（Redis）来作为内容缓存数据库，对频繁查询的数据内容做内容缓存提升查询性能。在使用过程中可以降低架构的复杂度、提升程序吞吐量、并且具有高水平拓展能力和低端硬件集群能力等优点。

内容缓存

通过键值数据库降低架构的复杂度。我们在数据库设计的时候需要满足3NF，但是一些页面为了满足页面信息的显示我们需要做非常复杂的关联查询。如我们在用户中心、司机登录后需要返回司机的基本信息、车辆信息、订单信息等。我们需要在业务数据库先查询司机的基本信息，再去关联司机的车辆信息，最后关联司机的订单信息，这个查询的关联过程会显得非常的复杂，由于互联网产品具有快速迭代的特征、如果出现通过冗余的方式来做可能会导致数据库表中存在很多冗余不便于维护和拓展，基于这样的情况我们可以在第一次创建司机信息的时候同步构建一个司机对象到键值数据库中，包含：基本信息、车辆信息、订单信息存储到同一个键值数据的Key中做一个清晰的定义，如果司机信息更新也，也对键值数据库做同步更新。可以使得查询的时候可以直接通过司机的手机号直接查询司机的所有信息，也不改变已有的业务数据库设计，又可以减少了业务系统对业务数据库的依赖，降低架构的复杂度的目标。

分布式锁

通过键值数据库提升系统的吞吐量。对于业务复杂，高并发和大流量的系统，内容缓存是重要的组成部分，提升性能的主要方式之一就是内容缓存，它可以挡掉大部分的数据库访问冲击，防止业务系统崩溃。系统中我们将用户信息、订单信息等。在同步存储业务数据的同时同步存储键到值数据库做内容缓存，需要在相关的业务节点快速的获取这些信息。比如：网关可以直接访问键值数据库来获取用户登录状态、以及访问权限等就可以直接通过NoSQL数据库中查询得到进行鉴别，来抵制流量冲击（DDos）、预防止爬虫，机器人等恶意攻击；位置中心需要将司机、乘客位置信息做3s一次实时上报到服务端进行存储，在派单的过程中需要获取周围的车辆信息来进行派单、以及在车辆监控平台实现行程监控，我们可以直接将这些实时信息存储到NoSQL数据库中，我们将位置实时位置信息存储到键值数据库中，能够做到高频的读写查询，据统计比存储到关系型数据库中速度要要快10倍以上，可以有效的提升平台的吞吐量。

实时位置上报

高水平拓展能力和低端硬件集群。NoSQL数据库能够很好的进行水平拓展，快速的提升系统性能。和关系型数据库集群方式不同。这种拓展不需要很大的代价。而基于低端硬件的设计理念为采用NoSQL数据库节省很多硬件的上的开销。例如：我们在项目使用两台普通的服务器做一个键值数据库Redis的集群就可以支撑线上的业务。我们可以使用之前替换下来的闲置资源。而且它的安装也是非常的简单不是特别依赖专门的DBA人员来维护。这样综合的利用了公司的服务器资源也避免了资源的浪费的到了运维部门的支持，同时也得到了领导的认可和支持。例如：我们在春运高峰的时候提前增加了一台NoSQL服务器就能够顺利的支撑业务的正常运行。对比业务数据库它拓展的成本非常的低。打破了人们对拓容的恐惧。增强其他部门对技术中心的信心，能够有效的支撑业务部门和客户。

经过全体成员的不懈努力。在2017年12月，先后在四川、重庆、河南、贵州、海南等城市全国开展内测。2018年2月，全国正式发布运营。上线1年多程序一直稳定可靠运行。无较大线上生产事故、查询缓存命中率在90%以上大大的减少了业务数据库的压力提升了系统的整体性能。在这段时间内NoSQL也进行过2次拓容而且性能都能够的能很好的提升。上线以来得到了多个地区交通部门的点名表扬和和上万用户的好评。也为我们后续的开发、迭代、运维奠定了一个良好的基础。为NoSQL数据库及其运用和性能优化积累了丰富的经验。

项目上线至今一直运行稳定，不论是司机端、乘客端、运营管理端，基本没有任何性能问题。但是有一次运维同事通过分析日志发现当Redis存储的内容超过100kb 过后会出现查询时间相对较长的情况。经过分析，这个是Redis本身的一个设计缺陷在这样的场景下会出现一个性能拐点。后面我们通过拆解存储设计来改善了这个问题。实践证明，项目能够顺利上线，并运行稳定，性能良好，与系统接入NoSQL数据库设计密不可分。我们对NoSQL数据库技术在项目的应用是一个持续的过程，我们接下来，还会继续不断完善NoSQL数据库技术在项目的应用，使整个出行平台更加稳定可靠。