移动网络车载设备

架构方案设计

修改履历

版本	修订日期	修改内容	作者	评审人
draft	2016-07-18	文档创建	Paul	

目录

1. 需求与背景

1.1. 需求概述

构建数据采集，添加对基于移动互联网通信车载设备集成与支持。提供基于 TCP 协议的 NIO 数据采集服务器。接受车载设备上传车辆数据、位置信息。并具备反馈实时数据信息至终端 APP 以及请求 DMS 服务完成数据后台持久化。

系统提供以下后台数据服务：

- 车队列表 查找（车牌）
- 车状态监控：在线、离线、所有。
- 车队实时监控。
- 轨迹回放（选择车辆 时间范围 回放 暂停 时间间隔）
- 统计报表（obd 数据、故障、报警、位置等等）
- 管理中心（司机管理、车辆管理、用户管理、电子围栏设置等等）
- 地图可视化显示
- 实时故障 实时报警

1.2. 参考文献

N/A

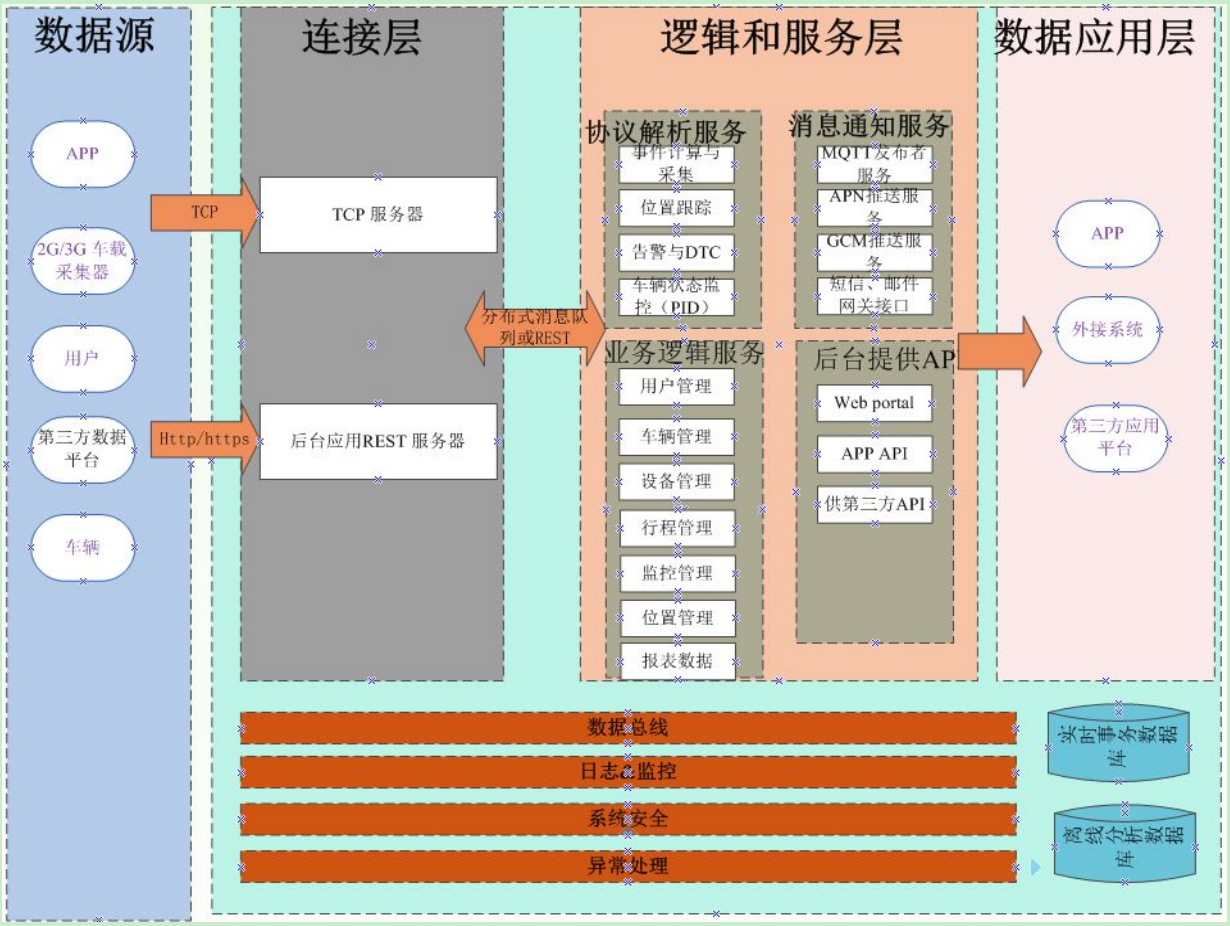
2. 开发主要工具表

功能	开发工具/语言	系统角色	简介
开发语言	JAVA	主要开发语言	面对对象工业开发语言
实时数据库	mysql 集群	实时数据持久化工具	开源、稳定高效的数据库集群
历史数据库	mongodb 集群	历史数据持久化工具	开源（AGPLO）、稳定、高效、存储量大易水平扩展
消息中间件	Kafka MQ	轻量化 TCP 服务器而引入的，基于订阅-发布模式的消息中间件。TCP 服务器收到 OBD 数据包完成验证后，立即由 Kafka 分发到各请求类型后台处理模块	开源、高效、分布式、大容量消息中间件
分布式缓存	Redis 3 集群	系统 2 级缓存	开源、分布式、大容量、高容错能力快速的 K-V 数据库
TCP 通信框架	Netty	TCP 通信底层框架	Netty 提供异步的、事件驱动的网络应用程序框快速开发高性能、高可靠性的网络服务器和客户Netty 基于 NIO，多 selector 线程、多 worker

负载均衡	Nginx	负责均衡反向代理	是一个高性能的 HTTP 和 反向代理 服务器，也是一个 IMAP/POP3/SMTP 代理服务器。能够支持高达 50,000 个并发连接数的响应
消息推送	MQTT/APN/GCM/短信网关(平台)		

3. 系统架构设计

3.1. 总体架构



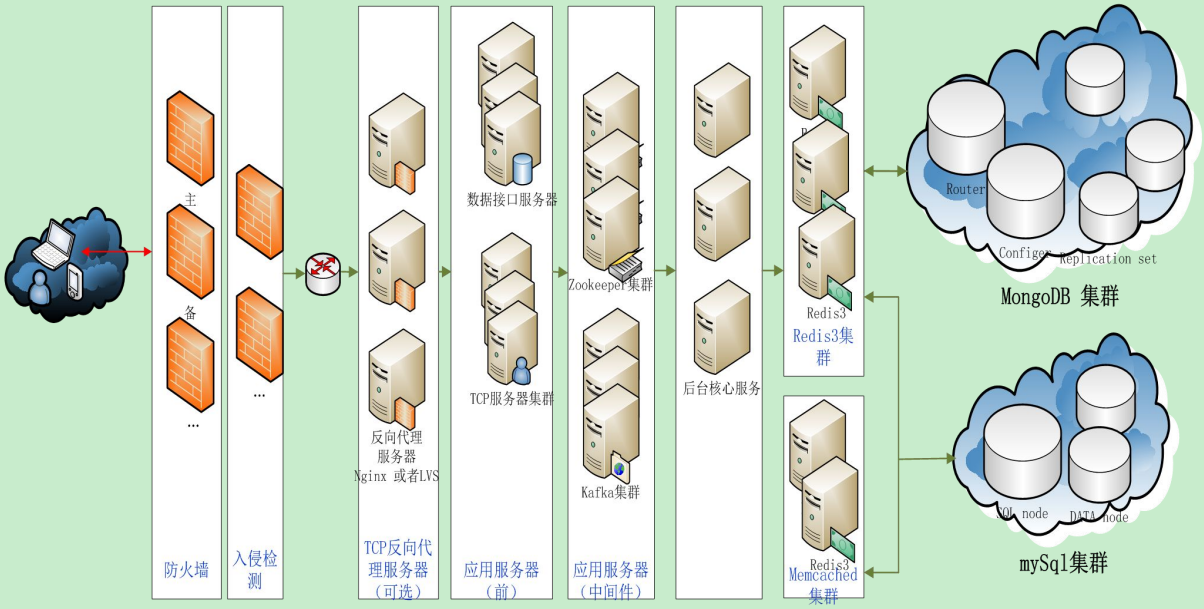
3.2. 总体架构说明

平台主按数据状态转化可分为 3 部分：数据源（原始状态）、连接层（结构化状态）、数据处理(数据应用层)：

- 数据源
  - **2G/3G 车载采 OBD**：收集车辆、GPS 数据由 TCP 协议发往后台采集器
  - **蓝牙设备 OBD**：与手机连接 BT 协议完成数据传输、再由手机 APP 与后台 http 连接完成数据上传。
  - **用户、车辆数据**：静态数据，后台提供数据注册/录入 API。
  - **第三方数据平台（可选）**：获取其他周边信息如：违章信息（交管平台）、车辆保养（经销商平台）。
- 连接层
  - TCP 服务器：基 NIO TCP 框架 Netty 开发，负责接收车载设备 TCP 链接请求，验证、解析、获取数据。
  - 后台应用 REST 服务器：提供实时、离线数据服务，数据后台。
  - 实时数据库：存放实时事务数据，数据特点为数据规模小，事务性要求高。如无特殊需求，采用关系型数据库 mysql 集群
  - 离线数据库：存放离线分析数据，数据特点为数据规模大，事务性要求不高。如无特殊需求，采用 No-SQL 数据库 mongodb 集群

- 数据消费端
  - 基于业务需求与采集到的结构化的数据。提供业务服务，主要包括：
    - 用户管理。
    - 车辆管理。
    - 设备管理。
    - 行程管理。
    - 位置管理。
    - 状态监控。
    - 报表服务
    - 等其他服务

3.3. 部署架构



说明：

- 安全考量：
  - 内外网隔离防火墙：除了访问控制外，还需要具备专业 DDoS 防范和 SYN Flood 防范能力能力。
  - 入侵检测/防御防火墙：对网络、系统的运行状况进行监视，尽可能发现各种攻击企图。以及攻击发时阻止攻击的恶意通信功能保障。
- 性能考量：
  - TCP 反向代理服务器：DCS-T 集群代理服务器
  - Kafka MQ: 业务逻辑异步化、消息分流。增加 DCS-T 负载能力。
- 高可用与容灾考量：
  - DDoS 防火墙、IDS/IPS 都应该配置成 HA。而且规则应当定期保存备份
  - 所有应用都不允许单点部署，至少配置成 HA。
  - Redis 持久化策略采用 aof。
  - 组件监控
  - 日志监控
- 数据库部署图
  - 见章节 3 详细说明

3.4. 系统吞吐量并发性能（估算）

估算前提：10 万的 TCP 客户端并发，每个客户端发包间隔 30 秒，平均每包 0.2kByte。

日中数据流量 55GByte

带宽需求：

峰值带宽

(数据包 size)KByte*最大并发数/1024*10 0.2KByte*100000/1024*10≈195.31Mbps

平均带宽

(日总流量)MByte/(24*3600)*10 (55*1024)MByte/86400≈6.52Mbps

CPU 及内存:

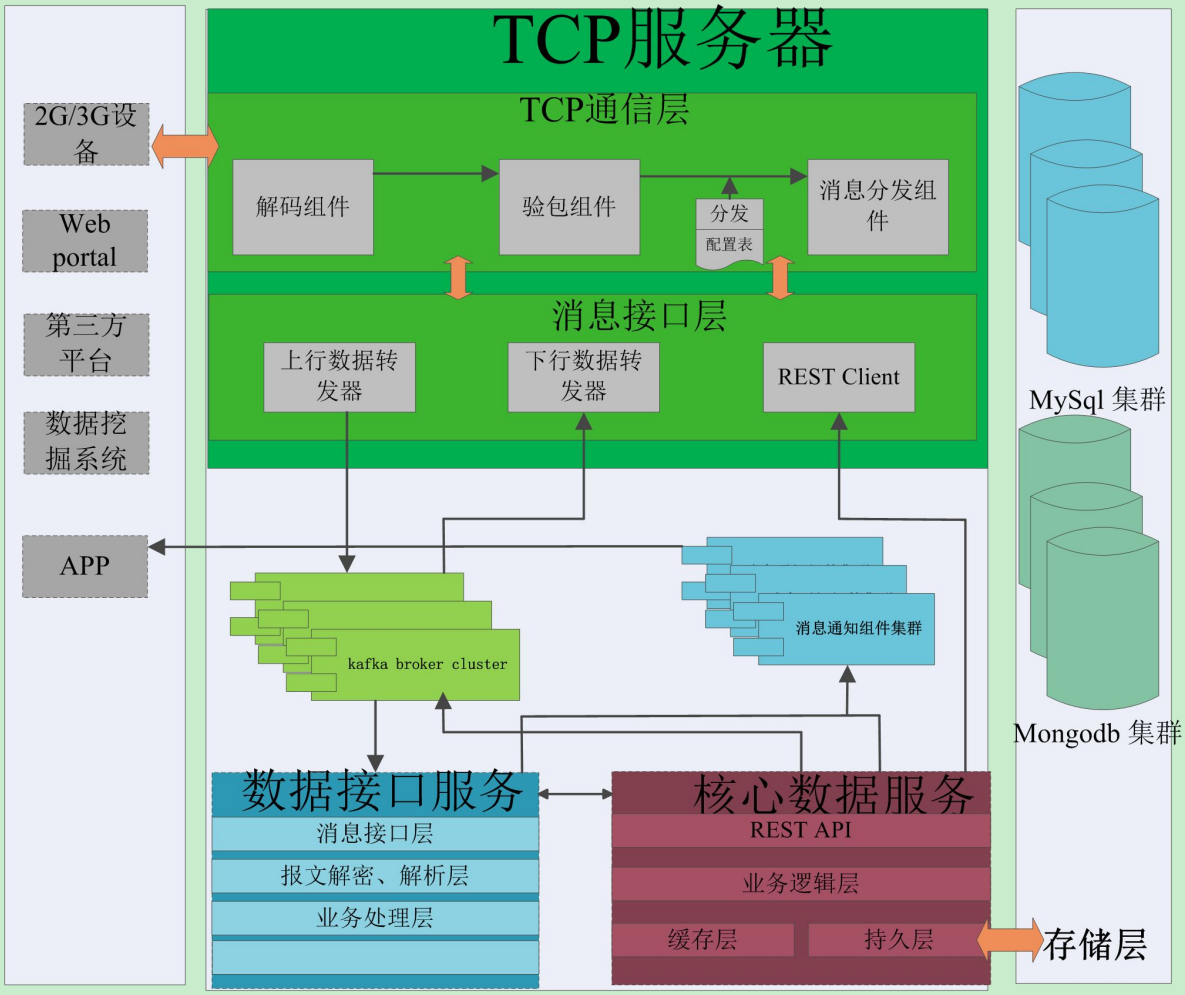
根据相关服务器压力测试结果

CPU	32 核* Dual Intel Xeon 5650
内存	64G
硬盘	2.8T
网卡	Intel Corporation 82576 Gigabit 1000Mb/s

在链接数到达 30000 时，尚未发现异常。CPU 占用峰值未超过 30%,内存占用率峰值未超过 20%。

3.5. TCP 服务器（车载设备数据采集服务器通信独立运行单元）

3.5.1. 系统架构



说明：

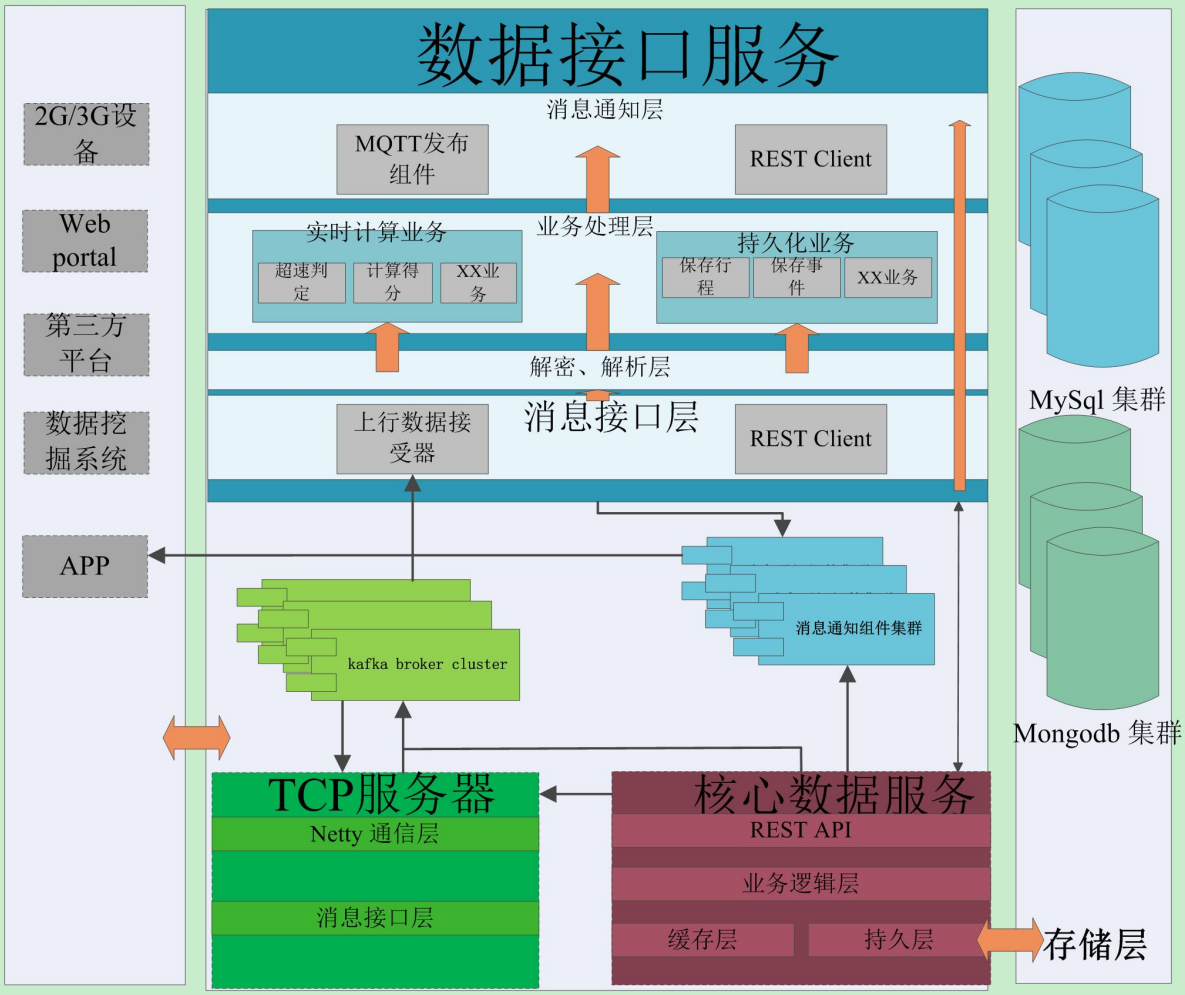
- TCP 通信层:处理 TCP 链接请求，采用 netty 框架（异步事件驱动网络应用服务器）以高并发能力、易操作、可扩展等能力
  - 解码组件：针对 TCP 网络传输中可能出现的拆包、粘包问题。
  - 验包组件：根据与硬件设备约定，做合法性校验。比如验证包头设备信合法性等
  - 分发配置表：与设备约定的消息类型到后台消息处理器映射表
  - 消息分发组件：基于 Kafka MQ（大吞吐、分布式、基于发布-订阅模式的消息系统）分发消息
- 消息接口层：发布/订阅来自 kafka 列数据，直接向 DMS 请求数据（基础数据）。
  - 上行数据转发器：向 Kafka 队列发布发布设备上行数据消息
  - 下行数据转发器：向 kafka 队列订阅后台对设备的配置更改、查询请求数据
  - REST Client：向 DMS 请求基础数据，辅助验证等任务。

3.5.2. 部署架构

请参考 2.3 章节

3.6. 车载设备数据业务逻辑处理独立运行单元

3.6.1. 系统架构



说明：

- 消息接口层：接受和请求数据层
  - 上行数据接收器：向 Kafka 订阅设备上报数据。
  - REST Client：直接向 DMS 请求相关业务支撑数据
- 解密、解析层：按照与设备约定，对报文完全解密、解析，将文件数据对象化，为后面持久化层组件做数据准备和逻辑关联、校验。
- 业务层：处理具体业务逻辑。此处按处理方式分为：实时计算类业务和持久化类业务。
  - 实时计算类业务：指需要实时反馈的运算逻辑，比如获取车辆超速信息（需

要实时获取车辆所在道路限速数据与行车速度对比），系统采用 Apache Storm 流计算框架。

- 持久化业务：对实时性要求不高，不适用流计算的场景。比如保持行程信息等

■ 消息通知层：通知 DSC-W 业务逻辑结果层代码（DCS-W 不直接写数据库）

- MQTT 发布组件：将实时数据通过 MQTT（轻量级、基于代理的发布/订阅式的消息协议）协议直接到手机 APP 端，实时反馈车辆、行车状态。
- REST Client：将持久化数据、部分实时运算结果发送至 DMS 完成最终持久化工作。

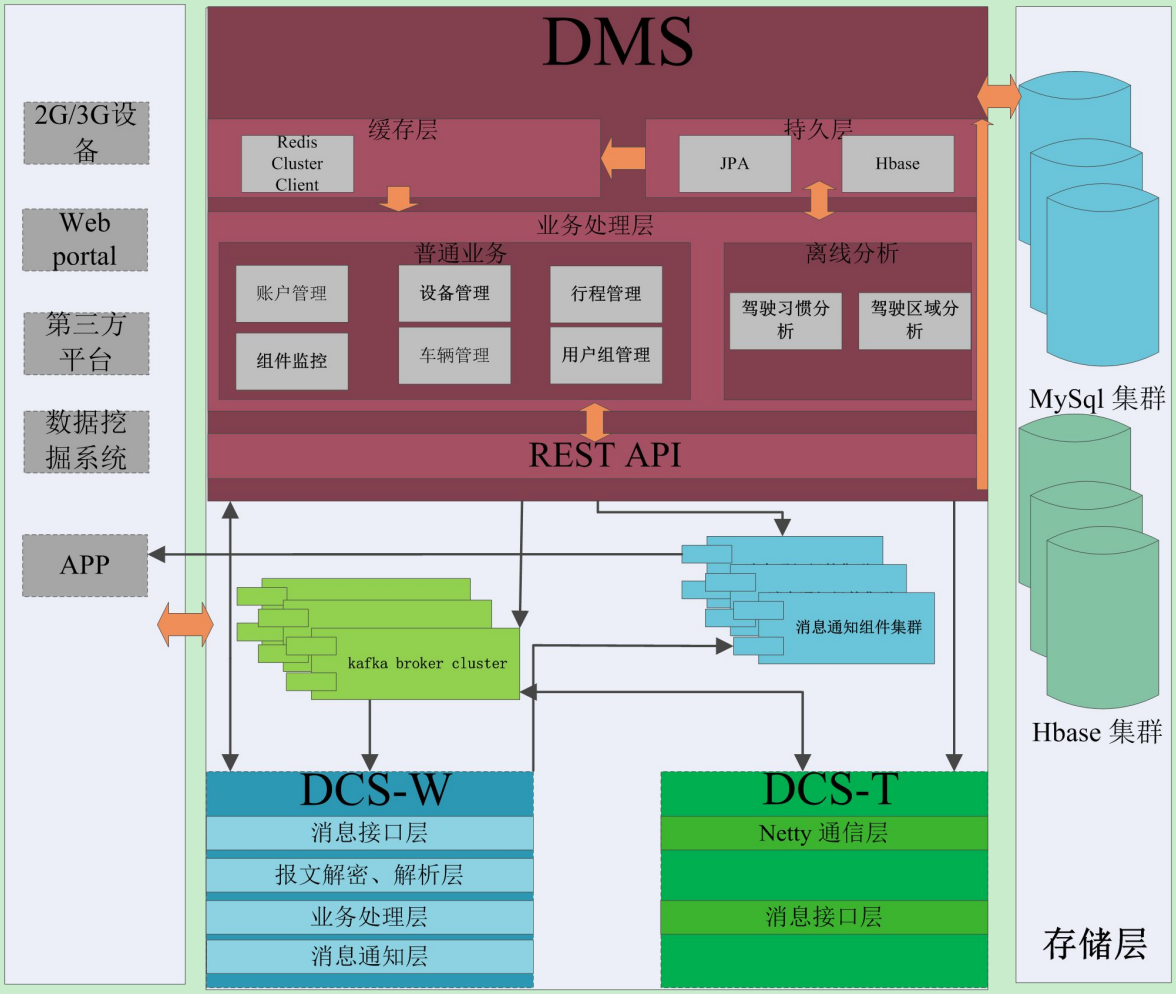
3.6.2. 部署架构

请参考 2.3 章节

3.7. 核心数据服务

3.7.1. 系统功能模块图

3.7.2. 系统架构



说明：

- REST API: 基于 http/https 的 REST ful 风格数据网络 API。
- 业务处理层：
  - 普通业务：通过对持久化数据操作完成账户管理等业务逻辑
  - 离线分析：基于 HBase 数据库数据，发布、执行离线批处理任务。比如驾驶习惯分析等任务。分析工具引用 Hive（基于 Hadoop 的一个数据仓库工具，可以将结构化的数据文件映射为一张数据库表，并提供简单的 sql 查询功能，可以将 sql 语句转换为 MapReduce 任务进行运行）
- 缓存层：系统二级缓存层，依赖 redis3（高性能可扩展 k-v 数据库） 缓存集群。
- 持久层：
  - JPA：采用 hibernate 实现版 JPA，完成应用程序到 mysql 对象-关系映射
  - Mongodb：NoSQL 数据库,分布式、可扩展的大数据仓库

3.7.3. 部署架构

参考 2.3 章节

4. 数据存储

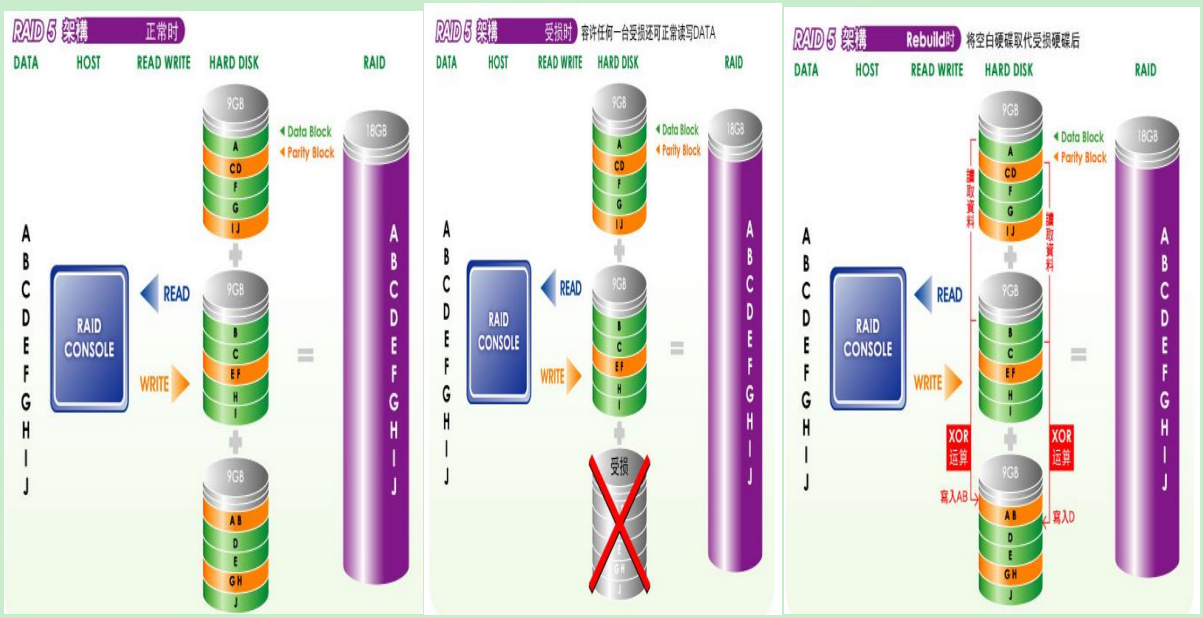
说明：数据存储估算都按在一年在线数据计算

4.1. 磁盘

综合考虑：可用性（数据冗余）、性能和成本三要素。结合实际需求，系统对数据存储性能、数据安全要求最高成本兼顾特性。采用 R5 策略即：RAID5

4.1.1. RAID5

图解 R5



RAID5 把数据和相对应的奇偶校验信息存储到组成 RAID5 的各个磁盘上，并且奇偶校验信息和相对应的数据分别存储于不同的磁盘上，其中任意 N-1 块磁盘上都存储完整的数据，也就是说有相当于一块磁盘容量的空间用于存储奇偶校验信息。因此当 RAID5 的一个磁盘发生损坏后，不会影响数据的完整性，从而保证了数据安全。当损坏的磁盘被替换后，RAID 还会自动利用剩下奇偶校验信息去重建此磁盘上的数据，来保持 RAID5 的高可靠性。

需要注意的是，做 RAID5 阵列所有磁盘容量最好一样大，否则当容量不同时，会以最小的容量为准。

4.2. 磁盘总量估算

RAID5 模式存储数据量比磁盘总空间系数计算公式（n 为磁盘块，n>=3）：

(n-1)/n

估算前提：每日 55G 数据（不考虑关系型数据库索引、关联等对象占用空间），以 3TB 磁盘来存储为例
每年需要：

每日数据 GB*365 日/102/3 55GB*365/1024/3≈7

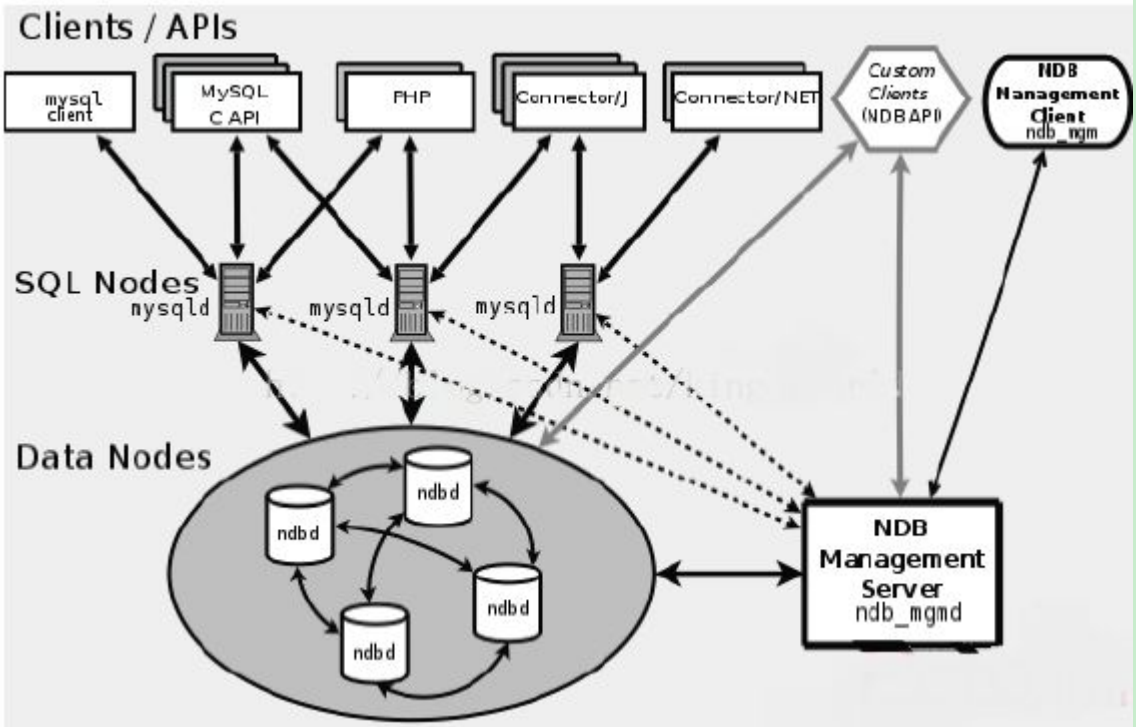
R5 策略下，实际需要磁盘数量 8 块

4.3. 在线数据（Mysql NDBClsuter)方案

4.3.1. 介绍

MySQL 官方集群部署方案，它的历史较久。支持通过自动分片支持读写扩展，通过实时备份冗余数据，是可用性最高的方案，可做到 99.999%的可用性。

架构图



说明：

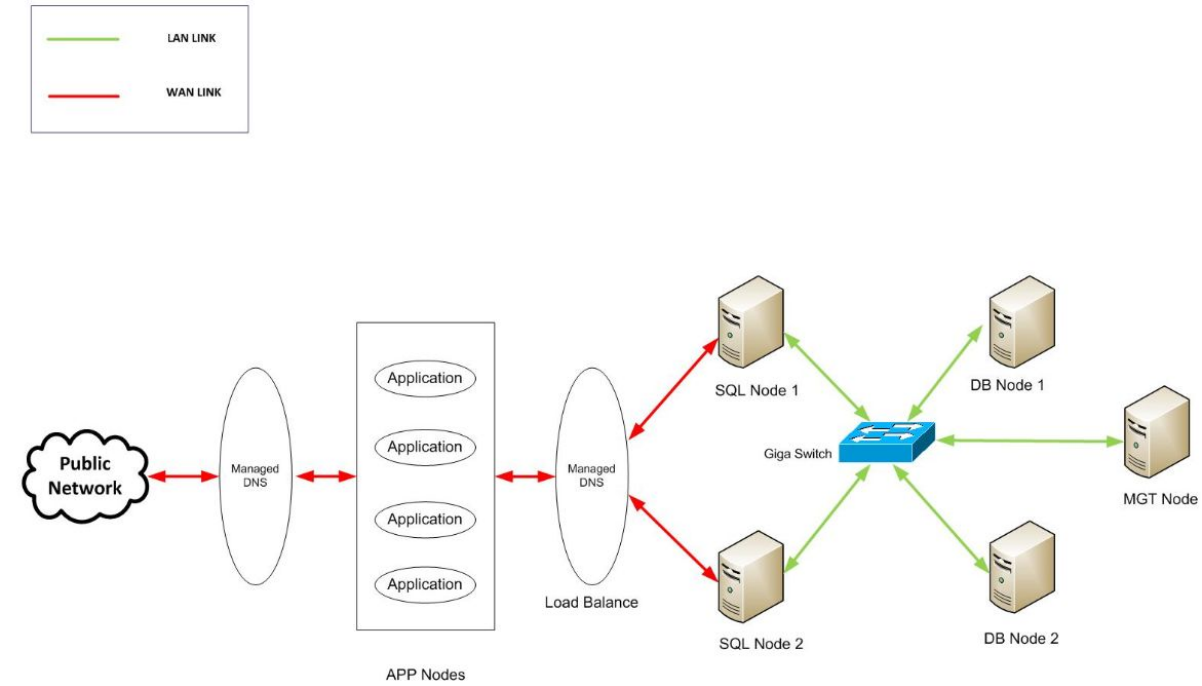
- NDB Management Server：管理服务器主要用于管理 cluster 中的其他类型节点（Data Node 和 SQL Node），通过它可以配置 Node 信息，启动和停止 Node
- SQL Node：在 MySQL Cluster 中，一个 SQL Node 就是一个使用 NDB 引擎的 mysql server 进程，用于供外部应用提供集群数据的访问入口。
- Data Node：用于存储集群数据；系统会尽量将数据放在内存中。

4.3.2. 特性

其优缺点总结如下：

优势	劣势
分布式存储引擎，可以由多个 NDBCluster 存储引擎组成集群分别存放整体数据的一部分	内存需求量巨大，索引以及被索引的数据必须存放在内存中
支持事务	单个请求响应慢
并发量高	与主流 Nosql 比，数存储量不足
属于 RDBMS，标准化 SQL，使用方便	与 Nosql 数据库比需要更多的磁盘空间
发布时间久，版本稳定	响应、并发不及 Nosql 数据库
节点间数据冗余备份，可用性高	与主流 Nosql 数据库比安装维护复杂，需要较高硬件成本
数据存储量大于 innodb 引擎	
成熟的活跃的社区支持	

4.3.3. 部署架构



4.4. 离线数据（Mongodb）

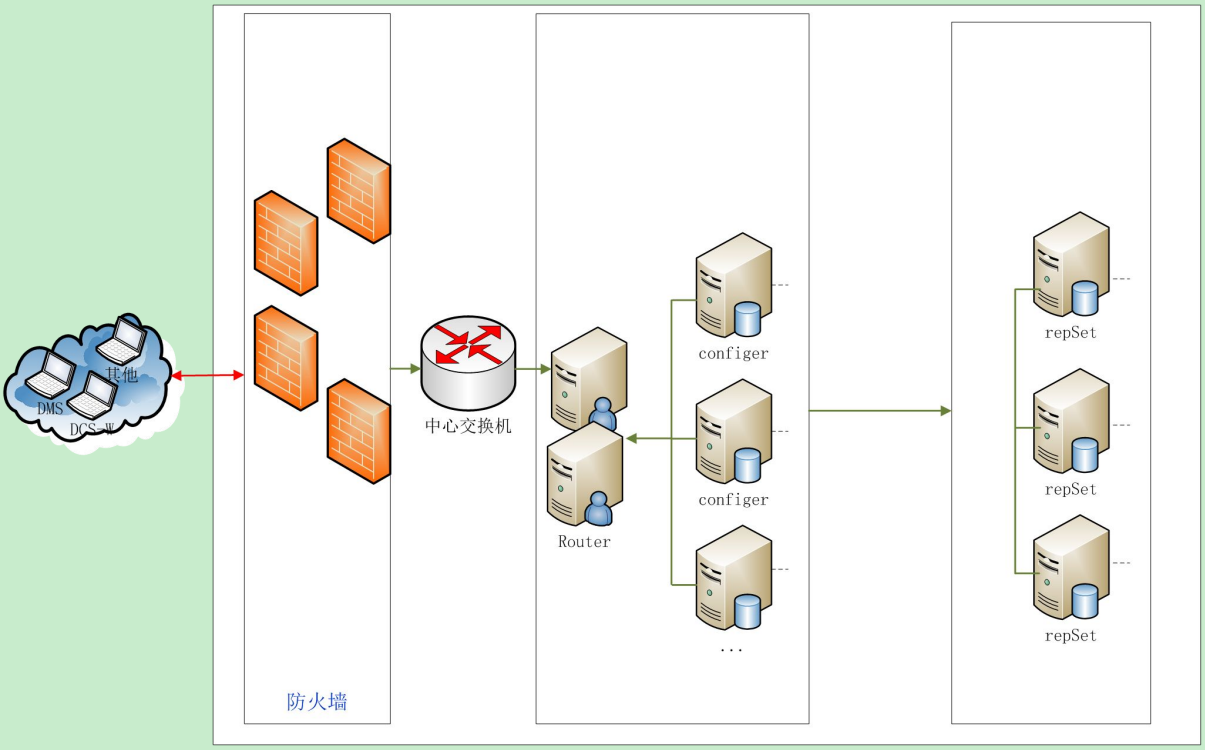
系统数据量巨大，在线一年数据量达到：55GB/日*365≈19.60TB、记录数达到：28800*365=10,512,000 万条。已远远操作传统 RDBMS 存储数据上限（即使采用 Oracle 全套 MAA 方案，高昂大代价也保证不了良好的性能）。考虑选择 Mongodb 作为系统离线数据存储数据库。

4.4.1.1. 特性

其优缺点总结如下：

优势	劣势
自带 sharding，可轻易扩展	不完备的事务支持（弱一致性、事务不隔离）
并发量高（同样条件下远远优于 NDBCluster）	新型数据库，有一定学习成本
单节点数据存储量大	
性能优越、功能强大	
存储量大	
单表数据容量大，支持亿行、百万列	
提供了基于位置的空间运行	
稀疏:对于为空(null)的列，并不占用存储空间	
非结构化	
容错性极高	

4.4.2. 部署架构



说明：

- 图中虚线并不表示数据/请求方向，而是状态监控依赖。
- 防火墙：内网资源保护、DDoS、IDS/IPS。
- Mongodb 集群：任何一点都应该避免单点故障。

5. 安全

5.1. 网络安全

- 内外网隔离，并采用不同的访问策略，
- 证书登录方式，并有登录统计分析,
- 外网机器的端口保护和监听，使用 fail2ban 和相关工具及时挡住入侵 IP
- 外网机器上部署访问策略和流量监控，对可疑行为实施阻止和邮件短信报警
- 对系统日志进行实时监控，对可疑行为进行报警
- 用 nginx + 防木马插件作为 tomcat 服务的前端，增加系统的可靠性。

5.2. 数据安全

5.2.1. 关键数据资产保护、链路保护

- 对系统的关键文件用密码方式保存指纹，每日进行对比，发现异常行为用邮件和短信报警

5.2.2. 数据备份与恢复

- 定时备份系统关键配置和重要文件，并加密保存到独立的备份机器，备份机器有主副两台，增加可靠性。
- 对数据库有多级备份机制，满足快速灾难恢复和快速指定日期恢复。

5.3. 监控和报警

- 使用 nagios 和大量定制插件，实现对流量，登录，服务，负载，内存，磁盘等多层次的监控和报警
- 使用 cron job 对关键后台服务的定时巡检和异常重启
- 其中定制了 nagios 图表插件，对各种服务的监控记录绘制变化趋势图，便于分析和排错。

6. 报价

尚未明确项目需求范围，按已知需求粗略估价：人民币 10 万元。具体报价应有需求范围确定后，在此基础上下浮动。

7. 其他