

基于 REST 与北斗定位系统的位置网^①

徐 栋, 何通能

(浙江工业大学 信息工程学院, 杭州 310032)

摘 要: 随着互联网技术日新月异的发展, 我们现在已经进入了以服务为基础的互联网时代. 位置服务是目前快速发展并且广泛受欢迎的一项服务, 但是现有平台存在的复杂度高, 性能弱等问题. 为了改善上述缺点, 在北斗定位系统的技术上提出一种基于 REST 架构风格的位置网业务平台方案, 将位置数据看做资源, 使用户可以采用统一接口来简洁、快速的开发位置网应用. 经过试验, 可以使用简单的 URL 访问资源, 降低了系统的复杂度, 并且在轻松通过了并发量 100, 访问量 10000 的并发测试, 大大增强了系统的性能.

关键词: 北斗; RESTful; 位置网

Location Network Design Based on REST and BDS

XU Dong, HE Tong-Neng

(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310032, China)

Abstract: With the rapid development of Internet technology, we have now entered a service-based Internet time. Location services is a widely welcomed and development service, but it has some problems—high complexity and weak Performance. To alleviate these drawbacks, this paper proposes a solution based on REST. After testing, it can use simple URL to get resource. And it can also pass 100 clients 10000 requests test.

Key words: BDS; RESTful; location network

基于我国个人消费者对于智能化的需求, 位置信息服务随着定位技术和无线上网技术的发展, 呈现快速发展的趋势. 相关统计显示, 截止 2009 年, 我国移动用户已经超过 7 亿, 机动车超过 7500 万辆. 如果以位置服务作为动力, 将大大推动国家与公共安全、节能减排、救灾减灾、交通运输、物联网、精细农业等领域的成长^[1].

而随着北斗卫星导航系统的发展与完善, 基于北斗系统开展的定位服务越来越受到人们的重视. 目前在定位服务领域中, 主要以全球定位系统(Global Positioning System GPS)为主. 国家安全为此存在巨大隐患.

同时目前常用的 SOCKET 套接字链接方式在复杂多变的移动互联网领域显得力不从心, Web 服务越来越多的被用于解决异构平台, 而 SOAP 风格的 Web 服务逐渐显露出复杂度高, 性能弱等缺点, 为了解决这

些问题, REST(Representational State Transfer 表述性状态转移)架构风格的 Web 服务逐渐进入人们的视野.

本文针对目前位置网的特定要求, 在分析已有集成方案的基础上, 提出了基于北斗卫星定位系统, 采用 REST 风格的 Web 服务的系统设计, 实现了聚集位置信息的位置网.

1 北斗系统概述

北斗卫星导航系统(BDS)是一个全球性的独立开发的卫星导航系统, 中国正在实施的自主经营, 致力于提供高质量的定位, 导航, 时间服务. 中国在 2003 年完成了具有区域导航功能的北斗卫星导航试验系统, 并且在 2012 年起向亚太大部分地区正式提供服务.

其主要特色主要为以下几点:

- 实现导航与通信集成
- 多系统兼容

^① 收稿时间:2014-03-12;收到修改稿时间:2014-05-04

● 提供 RDSS 双向授时授权服务

北斗卫星导航系统计划由 5 颗同步卫星和 30 颗非同步卫星构成, 其设计定位精度小于 10 米. 北斗卫星导航有两种系统, 分为卫星无线电测定系统(Radio Determination Satellite System, RDSS)和卫星无线电导航系统(Radio Navigation Satellite System, RNSS)^[2].

RDSS 采用主动定位方法, 请求由用户发出, 地面控制中心接收到定位请求后, 根据用户提交的信息测量并计算出用户到卫星的距离, 并根据地面中心保存的地图数据或用户请求中所带的高度信息测算出用户到地心的距离, 根据 3 球交汇的测量原理, 再将这 3 个距离进行定位运算, 将运算结果即位置信息使用卫星对用户广播, 最终由该用户终端接收, 这样就完成了一次快速实时的定位, 其精度与 GPS 近似. 除了定位功能, 这种定位方法还能发送每次多达 100 个以上汉字短信, 具有几十纳秒级精度的精密授时功能.

RNSS 采用被动定位方式, 卫星位置由地面系统监测, 并且通过地面系统进行卫星时间同步, 从而保证卫星星历的准确性. 该方法与美国 GPS 系统以及欧洲 Galileo 系统的定位原理相同, 最大的优点是接收机的容量在理论上不受限制, 并且隐蔽性强^[3].

2 REST 概述

REST 这个概念于 2000 年由 Roy Fielding 在其学术论文“Architectural Styles and the Design of Network-based Software Architectures”初次提出. 论文对使用 Web 服务作为分布式计算平台软件结构的系统, 进行了一些体系结构原则方面的讨论

REST (Representational State Transfer, 表述性状态转移) 指的是一组架构约束条件和原则. 满足这些约束条件和原则的应用设计就可以称为 RESTful.

REST 定义了一组体系架构原则, 用户可以根据这些原则设计这种以系统资源为中心的 Web 服务——包括使用不同语言编写的客户端如何通过 HTTP 协议处理和传输资源状态. REST 把所有需访问的事务视为资源. 各资源通过一个 URI 标识来区分. 用户可以通过简洁通用的接口来操作 Web 上的资源. 一般使用 GET、POST、PUT 和 DELETE 对应获取资源、创建资源、更新资源和删除资源四种方法^[4].

比如要取得 id 为 3 的设备当前位置信息, 可以使用逻辑 URI <http://demo.com/id/3/>, 返回客户响应式一

个资源的物理标书, 假设是 [location.html](#), 该页面包含一系列状态信息, 包括经纬度, 速度, 高度, 时间等信息. 对应的, 如果想要提交一个设备最新的位置信息给服务器, 可以使用 HTTP 协议的 PUT 方法向 URI <http://demo.com/id/3/>, 更新数据. 其优势有:

(1) 可扩展性强

REST 的无状态交互约束使得其架构的扩展性提高. 无状态交互要求客户端的每个请求包含该请求的所有信息. 例如, 状态信息不应该保存在服务器上, 也不包含在之前的请求中, 这就大大降低了系统提高规模所需的代价^[5].

(2) 耦合性低

在 REST 中, 数据接口和类型高度统一, 资源都是通过表示来处理的. 可以通过指定不同的 content-type 和 accept 头来选择消息中的数据格式. 这种数据的处理方式减轻了客户端与服务器端之间的耦合, 并且允许客户端选择它们惯用的数据格式.

(3) 安全性强

REST 中所有的资源有唯一的 URI, 而所有的请求都是基于 HTTP 的, 每个接口语义清晰, 分别对 HTTP 的四种操作设置权限即可形成行之有效的安全策略.

(4) 性能强

REST 建立在广泛使用的 Web 标准上, 减少了服务端和客户端对系统资源的占用. 另外, 在实际应用中, REST 可以采用现有的 WEB 服务器提供服务, 大大提高了其性能, 使之特别适合大规模数据应用.

3 位置网设计

3.1 系统设计模型

一个完整的系统由终端网络和业务平台两部分组成. 终端网络是由各种支持 REST 风格的设备组成的, 主要分成两种: 一种是向业务平台提供位置信息的数据源设备, 另一种是向业务平台发送请求, 获取设备位置信息的设备. 设计人员只要专注于 REST 的支持实现, 而无需关注终端网络的具体实现. 本文着重于数据源设备, 使用北斗的技术, 配合 GRPS 网络等技术实现, 图 1 为一个典型的使用 REST 方法的位置网系统的模型图, 各终端用户通过 REST 与位置网系统交互.

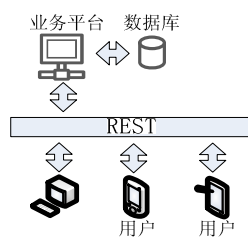


图 1 系统模型

业务平台：从总体上来看业务支撑平台是一个基于 REST 形式的中间服务平台。具有发现终端节点、生成资源描述信息、统一管理和定义节点资源、连接应用系统和终端网络的功能。使用中介器/包装器模型^[6]：中介器将请求分解成针对各个数据库的子查询请求，并将该请求发送给包装器，包装器将数据从各个子数据库中抽取出来，返回给中介器，最后中介器集成各个包装器发送过来的查询结果，并将结果格式化为开发者请求返回的数据格式再返回给第三方开发者，以支持他们开发具有 REST 风格的应用程序。

此外业务平台还应具有一个友好、易于操作的管理平台，授予用户权限，并据此暴露对应权限的 URL。当用户使用此 URL 向平台发出请求时，平台返回对应数据。本文的案例是使用 ThinkPHP 框架并配合 HTML5、CSS3、JQUERY 技术实现的

3.2 业务平台设计

业务平台由网站系统、后台维护系统、本地数据资源等 3 部分组成。从功能上来看，其具有设备节点注册，设备节点管理，图形化分析，用户注册，用户授权，日志管理等功能。

具体设计如下：

(1) 数据库设计

本系统使用 mysql 作为数据库服务软件。我们将保存设备节点的数据库叫做数据源库，考虑到数据来源的多样性，系统将数据分库存放，但基本结构类似，方便以后实现分布式系统。其基本结构如图 2 所示，数据源库中，由设备号，设备名，位置信息，更新时间，设备状态等基本数据构成一张基础表，如果有其他属性，可以自定义扩展表来满足需求。另一种保存管理数据的数据库我们把它叫做管理数据库，它包含了用户表，授权表，数据源信息表，设备基础信息表等。

(2) 数据处理

数据处理主要有两种，一种是更新设备信息，另

外一种是查询设备信息。

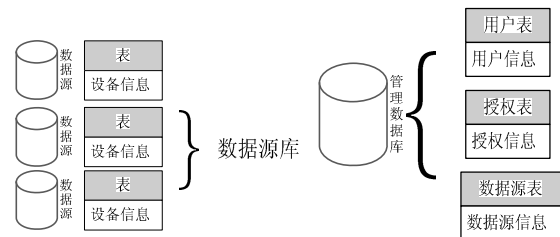


图 2 数据库模型

设备信息的更新就是设备节点向业务平台发送其最近信息，业务平台接受信息后，首先判断该设备是否合法，通过后，通过设备信息表得到该设备的对应的数据源库，最后向该库添加该设备提交的信息，完成这一过程。

数据查询采用上文提到的中介器/包装器模型：各种请求通过中介器发送给包装器，然后包装器将中介器需要的数据发送给中介器，最后由中介器将数据打包发送出去。

(3) URI 资源分配

由于位置网具有 REST 风格的业务平台需要暴露很多资源，为了简化模型，本文将其中的区域信息，设备节点列表，设备节点信息以及设备节点位置四种信息抽象成 3 种资源来分配 URI。

/area/list:显示区域位置(如杭州、嘉兴等)

/area/area_id:显示对应区域设备节点概况

/area/node/:显示对应区域设备列表

/node/node_id:显示对应设备信息

/node/node_id/location/date:显示对应设备对应日期(date)位置历史记录

/node/node_id/location/now:显示对应设备位置最新记录

由于只有设备节点位置信息允许通过该接口更新，即只有/node/node/location/info 允许 PUT 方法。

(4) REST 业务平台实现设计

ThinkPHP 是一个快速、简单的轻量级 PHP 开发框架，遵从 MVC 架构模式，并且正式引入 SAE、REST 和 Mongo 支持^[7]。

ThinkPHP 配置 REST 接口的相关参数如下，需要在其配置文件 Conf.php 做出配置才能使框架原生支持 REST。具体项目配置关键代码如下：

```
'REST_METHOD_LIST' => 'get,post,put,delete',
```

```
//REST 支持的请求方法
'REST_DEFAULT_METHOD' =>'get',
//默认方法为 get
'REST_CONTENT_TYPE_LIST' =>'json',
//返回数据类型
'REST_DEFAULT_TYPE' => 'json',
//默认返回数据类型为 json
'URL_ROUTER_ON' => true,
//开启路由
'URL_ROUTE_RULES' => array(
//开始定义路由规则
'Area/area_id/:id' => '/area/:1',
//隐藏不必要的 url 字符
由于标准的 ThinkPHP 的 URL 带有 index.php 为
了实现文中 3.2 所设计的 URI 规则, 对项目根目录
下.htaccess 文件做如下配置:
RewriteEngine on
#使能 rewrite 模块
RewriteCond %{REQUEST_FILENAME} !-d
#对所有请求做转发
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ index.php/$1 [QSA,PT,L] #为
所有请求添加 index.php 入口
```

其原理是使用 Apache 的 Rewrite 模块重写请求 URI, 由于此模块支持正则表达式[8], 所以可以使用正则表达式的语法将请求 URL 补充完整.

以上的配置使整个系统得以完善.

由于各个数据可能存在于各个数据源中, 设计一个中介器与一个包装器就显得非常必要. 外界交互的方法相当于一个中介器, 所以只要设计一个包装器就可以了. 对于包装器而言, 只要告诉它数据源, 搜索条件, 表名以及数据就可以了, 其主要代码流程图如图 3 所示.

所有方法中, 设备信息的相关方法最为典型, 我们以单个设备为例, 进行了查询及更新操作, 其流程图如图 4、图 5 所示.

● 获取设备信息

获取设备 ID 后, 首先查询设备的基本信息, 根据该信息确定数据源库, 然后将这些信息与查询条件传递给包装器, 尤其进行查询的操作, 最后返回所需要的信息.

● 更新设备信息

获取设备 ID 后, 首先查询设备的基本信息, 根据该信息确定数据源库, 然后将这些信息与最新位置数据传递给包装器, 尤其进行数据插入的操作.

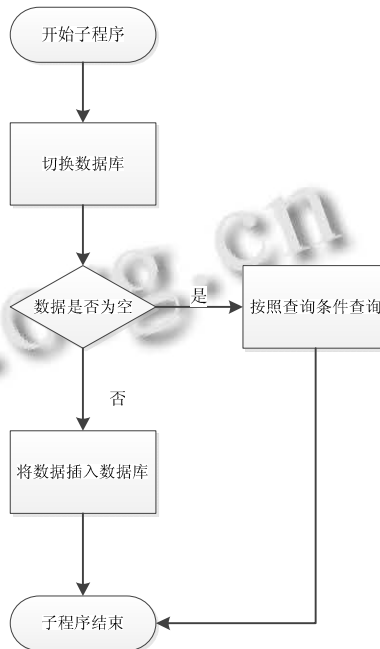


图 3 包装器工作流程图

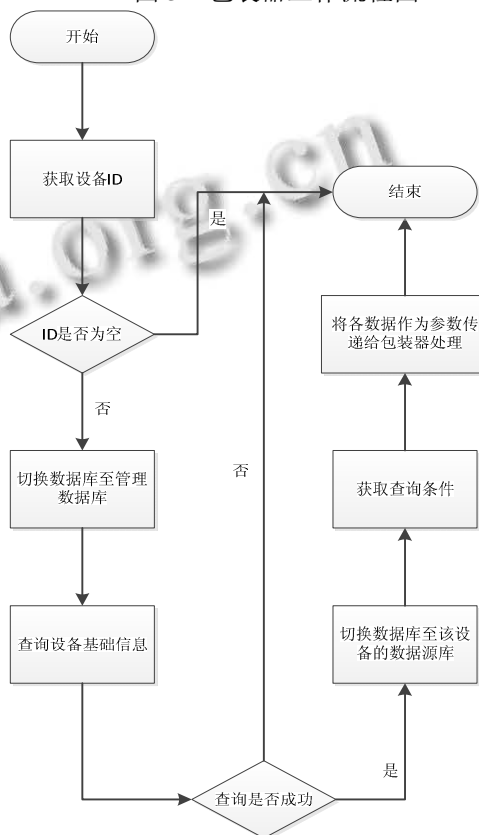


图 4 获取设备信息流程图

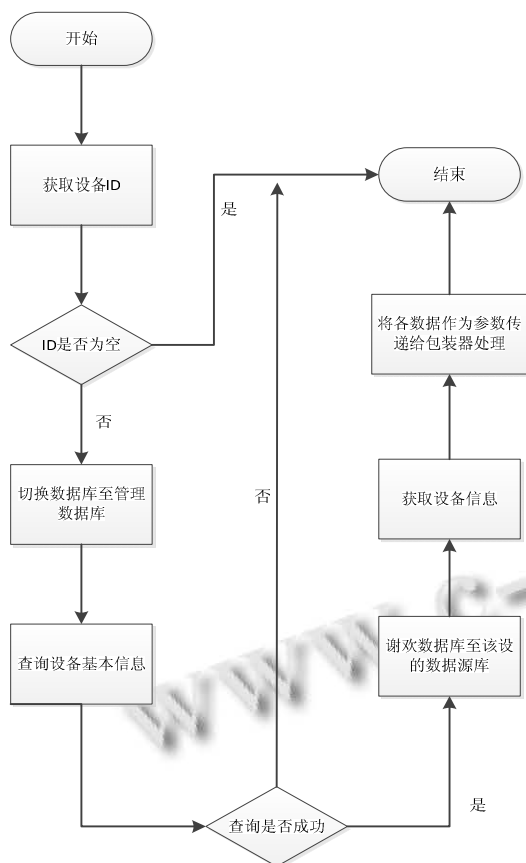


图 5 更新设备信息流程图

其余方法与此类似，故不再赘述。

如果查询并非一个设备而是一个地区的数据，其请求是这样被系统处理的：假设用户需要杭州市所有设备节点的信息，向系统发出请求，中介器接受请求后，从管理数据库中查找所有属于杭州的数据源，后向包装器发送请求，包装器从数据源中找到数据后返回给中介器，最后数据结果由中介器根据请求源将数据格式化后返回给请求者。

至此，一个采用 ThinkPHP 框架开发的基于 REST 风格的业务平台已经成型，可以将设备节点通过接口结合，将资源通过统一的 URI 接口提供给用户使用。

3.3 设备节点设计

图 6 为一个典型的设备节点框图。定位装置由北斗接收机，GSM 模块，单片机构成。接收机采用泰斗公司研发的 TD3020 型号的导航模块。其内部集成双模基带芯片和双模射频芯片，可以同时接收北斗和 GPS 信号。GSM 通信模块采用国产 SIM900 模块。其采用工业标准接口，工作频率为 GSM/GPRS 850/900/1800/1900MHz，可以低功耗实现语音、SMS、

数据和传真信息的传输。单片机采用基于 arm cotex M3 内核的 STM32F103。接收机与 GSM 模块通过串口与单片机通信。

TD3020 上电后，会自动搜星并定位，通过串口上报数据给单片机，其数据格式符合 NEMA 标准。

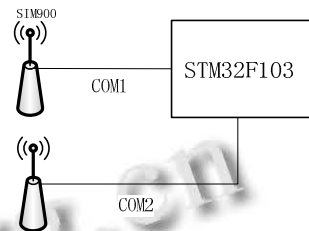


图 6 设备节点设计框图

使用单片机，通过一个简单的有限状态机，将位置信息及时间信息分离出来形成一帧数据，其帧头为设备识别号，数据区为其位置信息及时间戳。

准备好之后通过 at 指令里面 http 中 post 方式，将数据发送至服务器完成位置更新操作。其核心流程图如图 7 所示。



图 7 单片机程序流程图

3.4 系统测试

由于位置网系统链接着成千上万的位置设备，每时每刻都有位置设备会将把自己的位置信息更新到服务端，同时也会有各种请求获取位置信息的请求，所以并发性能是该系统的一个重要指标。

测试使用 ApacheBench(AB)^[8]这款软件。AB 是 Apache HTTP 的性能测试工具，其设计意图是描绘当前所安装的 Apache 的执行性能。限于篇幅，本文只做

了获取设备信息的并发测试

测试指令为 `ab -c 100 -n 10000 http://localhost/node/1`, 意义为, 对 `http://localhost/node/1` 这个 URL 进行测试, 每次并发客户端微 100, 一共有 10000 次请求。

图 8 中显示的测试结果表明, 该系统每秒响应能力达到 7000 余次, 具有实用性。

```
Requests per second: 7596.74 [# /sec] (mean)
Time per request: 13.164 [ms] (mean)
Time per request: 0.132 [ms] (mean, across all concurrent requests)
```

图 8 并发压力测试图

上述结果表明, 该系统完全能够承受大规模数据交互要求, 能够作为一个位置网系统面向终端用户。

4 结论

基于 REST 与北斗定位系统的位置网使用各种新技术的融合而成, 发展还有待更深入地研究和讨论。本文对整个系统架构进行初步的研究, 采用 ThinkPHP 框架实现基于 REST 风格的应用管理系统, 达到了设计的目标要求。由于位置网本身大规模与高并发的特点, 传统关系型数据库将渐渐力不从心, 从而成为整

个系统的性能瓶颈。如何突破这个瓶颈, NoSQL 这类对于超高规模和高并发方面的极大优势非关系型数据库可能是今后的研究重点。

参考文献

- 1 我国首次发布全球卫星导航“位置云”概念. 中国新通信, 2010, (16): 84.
- 2 谭述森. 北斗卫星导航系统的发展与思考. 宇航学报, 2008, 29(2): 391-396.
- 3 赵东艳, 原义栋, 石磊, 等. 用于智能电网建设的北斗/GPS 高精度授时方案关键技术. 电网技术, 2013, (9): 2621-2625.
- 4 潘冰. 面向资源的 RESTful Web 应用研究. 微计算机应用, 2010, 31(7): 38-43.
- 5 冯新扬, 沈建京. REST 和 RPC: 两种 Web 服务架构风格比较分析. 小型微型计算机系统, 2010, 31(7): 1393-1395.
- 6 孟小峰. Web 信息集成技术研究. 计算机应用与软件, 2003, 20(11).
- 7 杜冬梅, 许彩欣, 苏健. 浅谈正则表达式在 web 系统中的应用. 计算机系统应用, 2007, 16(8): 87-90.
- 8 邵芬, 于国防, 付海燕. AB: 一种简单的性能测试工具. 计算机时代, 2007, (12).