

基于关系和状态的移动云位置信息服务

向劲锋¹, 雷州¹, 张龙², 沈文枫¹, 段峰¹

(1. 上海大学 计算机工程与科学学院, 上海 200444; 2. 联想移动通信科技有限公司 创新中心, 北京 100091)

摘要: 受限的资源、易泄漏的用户信息一直制约着基于位置服务(location-based services, LBS)的应用程序的发展和推广。通过结合云计算强大的计算能力和存储能力,以及3G网络发展下的高速移动网络,提出一种移动终端加云服务器端的开发模式。该模式能够将移动端中复杂繁琐的计算任务和大量的存储任务交付给具有高速计算能力和海量存储能力的云端执行。提出一种基于关系和状态的信息推送策略。该策略能够很好地保护用户的位置等私密信息,为用户提供灵活的权限设置。通过一个原型系统的运行,证明系统的可行性和实用性。

关键词: 云计算; 移动云; 基于位置服务; 关系推送; 海量存储

中图分类号: TP 393

文献标志码: A

文章编号: 1007-2861(2013)01-0049-05

Mobile Cloud Position Information Service Based on Relationship and State

XIANG Jin-feng¹, LEI Zhou¹, ZHANG Long², SHEN Wen-feng¹, DUAN Feng¹

(1. School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China;

2. Lenovo Mobile Communication Technology Limited Haidian District, Beijing 100091, China)

Abstract: Limited resources and user information leak restrict the development and promotion of location-based services (LBS) applications. By combining the powerful computing power and storage capacity of cloud computing and the high-speed mobile network in the development of 3G network, a development model with a mobile terminal and cloud server-side are adopted. It can deliver complicated computing tasks and storage tasks at the mobile end to the cloud which has high-speed computing power and massive storage capacity. In addition, this paper proposes an information push strategy based on relations and states, which can well protect the user's location and other private information, and provide users with flexible permission settings. It is showed feasibility and practicality of the system through operation of a prototype system.

Key words: cloud computing; mobile cloud; location-based services (LBS); relationship pushing; mass storage

随着3G网络以及智能手机普及度的不断提高,移动互联网^[1]所具有的巨大的发展潜力也初现峥嵘,同时,作为其杀手级的应用——基于位置服务(location-based services, LBS)也受到了广泛关注。然而, LBS所需要的大量数据和强大的计算能力及其隐私保护问题^[2],都在一定程度上限制了LBS的发展。

云计算^[3]作为近几年计算机领域最热门的话题之一,已被业内人士普遍认为是互联网发展的一种趋势。同时,云计算也可以解决移动终端设备在资源上受限的问题,即通过将移动应用程序所需要的计算任务和存储任务放

置到高效“云”中,从而将移动设备从耗时耗能的计算任务中解救出来,并利用“云”的高速处理能力来弥补部分网络的时间开销。这不仅能够提高用户的体验度,还有助于用户使用更多种类的应用服务。目前,国内外已有多家大型公司投入到云计算开发中,中国移动也已结合云计算提出了“大云”计划。

本研究基于将移动互联网和云计算相结合的架构思想,提出了一种基于关系和状态的位置信息推送策略:海量的相关数据和复杂的关系运算放置于云端,采用分布式文件系统Hadoop^[4]和NoSQL^[5]数据库HBase存储

收稿日期: 2012-11-25

基金项目: 国家科技重大专项资助项目(LX201102103)

通信作者: 雷州(1972—),男,副教授,博士,研究方向为云计算。E-mail: leiz@shu.edu.cn

数据,使用MapReduce^[6]编程模型处理数据.这样既降低了手机端的负载,也提高了数据的处理效率和扩展度.同时,云存储技术也提高了用户数据的安全性^[7],通过推送策略为用户提供灵活的限制策略,从而使用户能够随心所欲地向其所希望的对象公开相应的数据.

1 系统描述

图 1 给出了基于云计算的移动 LBS 系统的基本框架.本系统主要分为 3 个模块:移动终端、关系服务器云、位置信息数据服务器云,其中移动终端使用 Android 操作系统,关系服务器集群和位置信息数据服务集群均处于远端的云计算数据中心内.本系统的开发思路是,在尽可能少地改变手机端编程模式的情况下,将现有的移动应用扩展到云计算中.

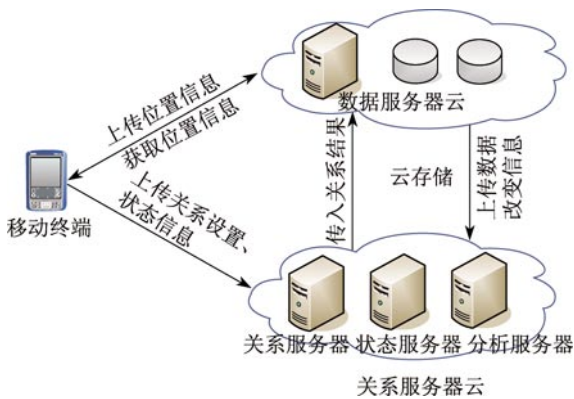


图 1 LBS 系统的基本框架
Fig. 1 Basic framework of LBS system

移动终端作为用户直接的操作对象,决定了用户体验的好坏.本系统首先考虑了用户和开发人员的习惯,缩小了手机端的程序编写与传统的移动应用编写之间的区别,主要是在服务器端设计了一个 API 接口,将传统的应用迁移到云池中.其次,终端和服务器之间的联系以短连接为主,从而减少了终端的能耗.

关系服务器云是本系统的核心,由关系服务器集群、状态服务器集群和分析服务器集群组成,其作用是进行用户关系数据的存储和分析.关系服务器用来存储用户的个人信息以及用户对其好友设置的关系级别信息;状态服务器用来存放用户的实时状态信息和历史状态信息;分析服务器则是在有用户请求他人位置信息时,从关系服务器和状态服务器中提取相关信息并进行相应的分析,最后将分析计算出的结果发送给数据服务器云,使其可将相关级别的位置信息发送给请求者.

位置信息服务器云是本系统的存储核心,主要负责存储用户上传的位置信息,以及为相关用户推送相应的位置信息.当新用户在本系统中注册成功之后,数据服务器就会从关系服务器中获取用户的 ID 信息.同时,移动

终端可获取位置信息服务器的 IP 地址.此后,用户就可直接向数据服务器上传自身的位置信息,且当数据服务器接收到位置信息后,将更改用户对应的状态,并发送给关系服务器.当关系服务器向该用户发送数据推送请求时,数据服务器会根据得到的分析结果,将相关级别的位置信息发送给相应的用户.

相比于传统的基于 Android 的 LBS 系统,本系统的后台是以弹性云计算技术作为基础支撑的,因此,在数据存储、分析以及弹性扩展等方面具有很强的优势.

(1) 支持海量的数据存储.本系统采用的是云存储系统,其特有的庞大的存储空间能够很好地支持海量数据的存储,从而使本系统能够更好地对资源进行整合,为用户存储更完善的资料,支持更多的用户使用,其廉价的分布式结构也更有利于对数据的备份和处理.

(2) 支持海量数据的高效分析.本系统使用 NoSQL 数据库对数据进行管理,再结合相应的并行编程模型,从而能够更高效地对用户的关系数据、位置信息数据等进行分析.

(3) 本系统后台的所有服务器都是以虚拟机的形式存在于弹性云系统之中.弹性云系统可以根据用户的负载来动态地启动和关闭虚拟服务器,从而具有很强的弹性扩展能力,并可以很好地应对突发的、激增的并行访问量.

2 系统工作流程设计

基于关系和状态的移动云位置信息服务系统的工作流程主要分为两个部分:常规运行和位置请求.

常规运行流程如图 2 所示.当有新用户要进入本系统时,需首先向认证服务器进行注册操作;注册成功之后,需要将用户的用户名、连接用 ID 等信息写入关系服务器中的用户信息服务器中;然后,用户信息服务器将用户的 ID 发送给数据服务器,同时用户还可执行添加好友、为好友设定权限的操作(此操作可于注册成功后的任意时刻执行).信息录入完毕之后,用户就可将个人的位置信息上传到数据服务器中,此时新用户就可获取其所需的服务了,而老用户只需要进行登录并上传位置信息即可.在后期的运行中,当用户的位置发生符合用户事先设定标准的变化时,就可将新的位置信息上传到数据服务器中.

当发送位置请求操作时,其主要的操作集中于后台服务器中,服务器端的流程如图 3 所示.当用户接收到位置请求信息时,或者用户上传的位置信息发生符合相关设定的变化时,关系服务器会对这些请求进行分析和计算,最终得出一个用户联系 ID 和位置信息级别的数据集,并将其发送给数据服务器.然后,数据服务器再根据得到的结果将相应的位置信息发送给相应的用户.

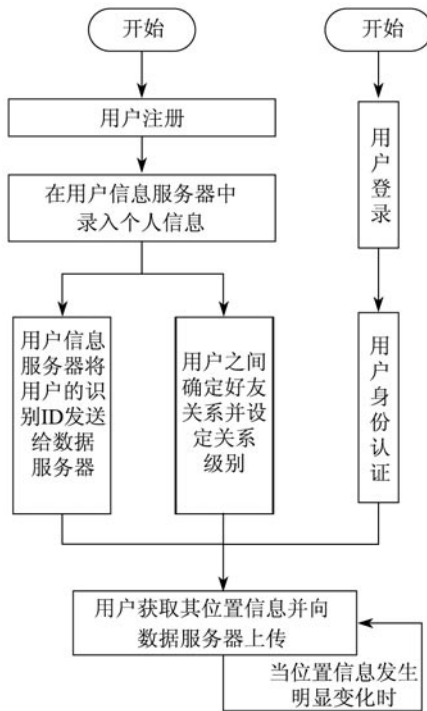


图 2 常规运行流程

Fig. 2 Conventional operation flow chart

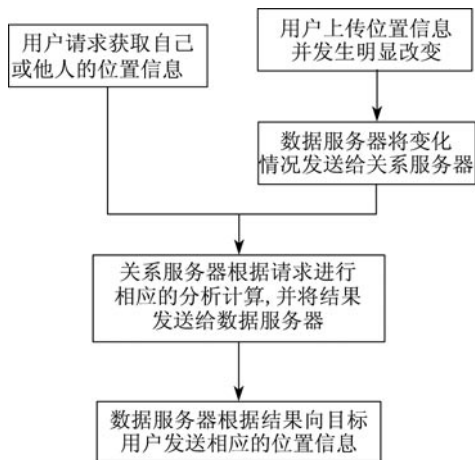


图 3 位置请求流程

Fig. 3 Position request flow chart

3 基于关系与状态的移动位置信息推送策略

本系统使用了一种基于关系与状态的位置信息推送策略,在一定程度上保证了用户位置信息的私密性和灵活性. 通过使用本策略, 可以允许用户只向其所认可的用户发送相应级别的位置信息, 主要内容包括: 判断用户是否允许向联系人推送位置信息; 当用户允许向联系人推送位置信息时, 获取用户的移动终端所处的位置信息; 确定联系人的关系级别, 向联系人推送对应关系级别的位置信息.

3.1 用户关系、状态以及位置信息级别设定

首先, 由用户之间设定好友关系, 待好友关系确定之后, 用户可独立地为其联系人设定相应的关系级别, 其中用户的联系人可以对应两个以上的关系级别. 具体地, 在本系统中用户可以为其联系人定义 4 个关系级别: “至亲”、“至交好友”、“普通朋友”和“黑名单”, 其中“至亲”和“黑名单”为特权级别, “至交好友”的级别权限大于“普通朋友”. 在用户为其联系人设定关系级别之后, 关系服务器存储用户的联系人信息.

然后, 对位置信息的精度级别进行设置, 其中位置信息可以对应两个以上的精度级别. 具体地, 本系统定义了 4 个精度级别的位置信息: “详细”、“城区”、“城市”和“不显示”, 其中“详细”表示位置信息具体到街道门牌号, “城区”表示位置信息具体到城区, “城市”表示位置信息具体到城市, “不显示”则表示无法获取用户的位置信息.

最后, 设置用户自身的运行状态, 并将用户的运行状态存入关系服务器中. 用户可以设置自身两个以上的运行状态, 既可以用来向联系人显示用户的运行状态, 也能够影响联系人对应用户位置信息的获取. 具体地, 在本系统中用户可以定义 4 种运行状态: “发现我吧”、“活动”、“休息”和“下线”, 其中“发现我吧”和“下线”状态为特权状态, “活动”状态的级别高于“休息”状态.

3.2 建立基础关系规则

设定了关系级别之后, 在联系人的关系级别与位置信息的精度级别之间建立一个初始的对应关系: “至亲”对应“详细”, “至交好友”对应“详细”, “普通朋友”对应“城区”, “黑名单”对应“不显示”. 不同关系级别的联系人将会获取用户不同精度级别的位置信息. 具体地, 关系级别为“至亲”和“至交好友”的联系人可以获取用户具体到街道门牌号的位置信息, 关系级别为“普通好友”的联系人可获取用户所在城区的位置信息, “黑名单”对应的联系人则无法获取用户的位置信息.

同时, 根据用户的运行状态, 可以向用户的联系人推送对应关系级别的位置信息. 首先, 根据用户的运行状态判断用户是否允许向联系人推送位置信息. 若允许, 则确定用户与联系人的关系级别所对应的精度级别, 并向联系人推送相应的位置信息.

当用户的运行状态发生改变时, 向关系服务器上改变后的运行状态. 关系服务器将用户的运行状态发生改变这一信息上传给数据服务器. 数据服务器根据用户改变后的运行状态判断用户是否允许向联系人推送位置信息. 当用户的运行状态为“发现我吧”和“活动”时, 说明用户允许向联系人推送位置信息, 并且允许方圆 $k \text{ km}^2$ 范围内使用相同软件的其他用户获取其位置信息和用户信息, 而这些用户可以不必是本用户的联系人.

当用户的运行状态为“休息”和“下线”时,说明用户不允许向联系人推送位置信息,其中在关系级别和运行状态中有 4 个特权集:“至亲”、“发现我吧”、“黑名单”、“下线”,且其权限级别为{“黑名单”,“下线”}>{“至亲”,“发现我吧”}.当联系人的关系级别为“黑名单”或者用户的运行状态为“下线”时,则联系人只能获取精度级别为“不显示”的位置信息,即无法获取该用户的位置信息.当联系人的关系级别为“至亲”或者用户的运行状态为“发现我吧”时,则关系级别为“至亲”的联系人能够获取精度级别为“详细”的位置信息,即可以获取该用户最详细的位置信息.

进一步地,用户设置的运行状态还可用以表明用户自身是否接受推送的位置信息,即当用户的联系人的位置信息发生了明显变化时——此处所谓的明显变化可由用户自定义,可将联系人所处街道发生变化视为明显变化,也可以将联系人所处城区发生变化设定为明显变化,通过用户设置的运行状态,判断用户是否接受位置信息推送.当用户的运行状态为“发现我吧”或“活动”时,则用户接受位置信息推送.当用户的运行状态为“休息”或“下线”时,则用户不接受位置信息推送,其中当用户的运行状态为“发现我吧”时,用户会向数据服务器申请获取方圆 $k \text{ km}^2$ 范围内使用相同软件,且运行状态同样为“发现我吧”的其他用户的位置信息和用户信息,且这些用户可以不必是本用户的联系人.

3.3 获取最终规则

根据 3.2 节所述的规则,将所有条件进行综合考虑,最终得出条件结果集,结果如表 1 所示.

表 1 条件结果集
Table 1 Set of results

关系级别	运行状态			
	发现我吧	活动	休息	下线
至亲	{接受, 详细}	{接受, 详细}	{不接受, 详细}	{不接受, 不显示}
至交好友	{接受, 详细}	{接受, 详细}	{不接受, 城区}	{不接受, 不显示}
普通好友	{接受, 详细}	{接受, 城区}	{不接受, 城市}	{不接受, 不显示}
黑名单	{接受, 不显示}	{接受, 不显示}	{不接受, 不显示}	{不接受, 不显示}

下面就普通情况和条件中包含特权级别时的情况各举一例进行说明.

例 1 普通情况.当条件为{“至交好友”,“活动”}时,即用户 A 设置的运行状态为“活动”,用户 A 对用户 B 设定的关系级别为“至交好友”时,最终的用户行

为如下:①当用户 A 的联系人的位置信息发生了明显变化时,用户 A 允许数据服务器将联系人变化后的位置信息推送给自身;②用户 A 允许数据服务器推送或者当用户 B 主动请求时,给予用户 B 自身精度级别为“详细”的位置信息.

例 2 特殊情况.当条件为{“至亲”/“至交好友”/“普通朋友”,“发现我吧”}时,即用户 A 设置的运行状态为“发现我吧”,用户 A 对用户 B 设定的关系级别为“至亲”、“至交好友”或者“普通朋友”时,最终的用户行为如下:①当用户 A 的联系人的位置信息发生了明显变化时,用户 A 允许数据服务器将联系人变化后的位置信息推送给自身;②用户 A 允许数据服务器推送或者当用户 B 主动请求时,给予用户 B 自身精度级别为“详细”的位置信息;③用户 A 会向数据服务器申请获取方圆 $k \text{ km}^2$ 范围内使用相同软件,且运行状态同样为“发现我吧”的用户的位置信息和用户信息,同时也允许这些用户从数据服务器中获取自身的用户信息和精度级别为“详细”的位置信息.

3.4 实验结果

到目前为止,本研究已经完成了原型系统的研发工作:以两台物理机为后台服务器集群,并在其上搭载了以 XenServer 为基础的、自主研发的一套虚拟机管理系统,启动了相对应的虚拟关系服务器集群、虚拟状态服务器集群和虚拟分析服务器集群;之后又以三台服务器为基础,通过在其上搭建 Hadoop 和 HBase,形成了相应的云存储系统.

实际运行情况表明,本系统可以准确地确定好友的位置信息,且其对应的关系模型也在实验中得到了验证.

4 安全策略

除了使用基于关系和状态的推送策略来保证用户数据的安全性以外,本系统还采用了如下几种安全策略.

首先,本系统将用户的个人信息数据和连接识别数据进行了分离操作.当用户与关系服务模块连接时,网络传输的数据中只包含了用户的个人信息,而当用户和数据服务器进行通信时,其用来确定身份的是连接所使用的 ID 信息.因此,即使有不法分子在网络中截取了传输的数据包,也无法获取完整的用户个人信息和位置信息,从而保证了用户数据的私密性.

其次,云数据中心有专业的安全设施和工作人员,能够更有效地防范黑客的非法入侵.同时,本系统的关系服务器和数据服务器之间是通过内网进行联通的,因此,能够进一步地防止用户个人信息和位置信息的匹配泄漏.

最后,用户与用户之间是禁止直接传送位置信息的,这样能够有效地防范中间人攻击等,而且只有用户自身和一些特殊成员才可访问其历史的位置信息.通过在用

户和特殊成员的移动终端维护一个时间和当时经纬度的数据表,在需要获取历史数据时,要将时间以及经纬度一起上传,匹配成功后才会将相应的信息返回,从而能够在一定程度上保护用户的历史私密信息.

5 结束语

随着移动互联网和云计算的持续发展,移动互联网应用的需求量也必将随之不断增加.针对此机遇,如何设计应用程序使其能够更适合用户的使用就变得尤为重要.本研究采用了云计算和移动互联网相结合的设计模式,即“端+云”的开发模式.实验证明,这种设计模式能够很好地解决移动终端资源限制的问题,使其能够运行计算任务和存储任务更加复杂的应用程序,为移动终端应用程序的开发开辟了更加广阔的道路.同时,本研究提出的基于关系和状态的位置信息推送策略能够为用户信息的私密性提供保护,实现了用户只为特定人群公开特定位置信息的功能,再通过其他的安全措施,能够较好地保护用户的信息,使用户能放心地使用本系统.本系统使用的均为模块化开发方式,且有很好的对外支持接口,能够很好地接入智能推送等计算密集型算法,充分利用了云池的计算能力,提高了算法效率.

本研究只是初步使用了“端+云”的开发模式,而且提出的安全推送策略也不够完善.因此,在未来的工作中,还将进一步研究移动互联网和云计算相结合的应用程序的开发细节,使其性能变得更好,并修改部分人工智能算法,使其可利用云池资源,提高工作效率.同时,也

将进一步完善安全策略,为用户的私密信息提供更周全的保护.

参考文献:

- [1] 肖志辉. 移动互联网研究综述 [J]. 电信科学, 2009, 25(10): 30-36.
- [2] 沈晶歆. 移动互联网关键技术及典型业务产品研究 [J]. 电信科学, 2010(10): 5-12.
- [3] MICHAEL A, ARMANDO F, REAN G, et al. Above the clouds: a Berkeley view of cloud computing [R]. Berkeley: University of California, 2009.
- [4] FAY C, JEFFREY D, SANJAY G, et al. Bigtable: a distributed storage system for structured data [C]// Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation. 2006: 205-218.
- [5] BOGDAN G T, CRISTIAN B. A comparison between several NoSQL databases with comments and notes [C]// Proceedings of the 10th Roedunet International Conference. 2011: 1-5.
- [6] DEAN J, GHEMAWAT S. MapReduce: simplified data processing on large clusters [C]// Proceedings of the 6th Symposium on Operating Systems Design and Implementation. 2004: 137-150.
- [7] MAHADEV S, PARAMVIR B, RAMÓN C, et al. The case for VM-based cloudlets in mobile computing [C]// IEEE Pervasive Computing. 2009: 14-23.