

Scudware Mobile: 支持可穿戴设备协同的移动中间件

丁 扬 王淑刚 李石坚 潘 纲

(浙江大学计算机科学与技术学院 杭州 310027)

摘 要 最近几年可穿戴设备发展迅速,各种样式和用途的可穿戴设备大量出现。然而这些可穿戴设备大多都是独立工作的,设备与设备之间鲜有联系。为此提出了一个以智能手机为中心的、支持个人范围的和人与人之间的设备的协同模型。基于该模型设计并实现了 Scudware Mobile 移动中间件,该中间件运行于智能手机上,可以汇聚智能手机和可穿戴设备的数据和服务,且以开放授权机制开放给用户,实现了数据和服务的协同。在 Scudware Mobile 移动中间件的基础上实现了 MobileTrace 足迹记录器应用,验证了 Scudware Mobile 移动中间件的可用性。

关键词 可穿戴设备,协同,中间件

中图分类号 TP311.56 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.9.004

Scudware Mobile: Mobile Middleware for Collaboration of Wearable Devices

DING Yang WANG Shu-gang LI Shi-jian PAN Gang

(College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China)

Abstract In recent years the wearable devices develop rapidly. Wearable devices of various styles and usage appear in abundance. However, most of these wearable devices work independently and there is little contact between device and device. Thus it is very hard to realize the collaboration of the data and services of the devices. To solve these issues, we proposed a model which is smartphone-centered and supports collaboration of the data and services among devices on the scale of individuals and between man and man. We designed and implemented Scudware Mobile, a data and services collaboration middleware. The middleware runs on the smartphone. It can gather the data and services of the smartphone and wearable devices and be open to the users based on the open authorization mechanism. It realizes the collaboration of data and services. We also implemented an application named MobileTrace based on the Scudware Mobile platform. The application verifies the usability of Scudware Mobile.

Keywords Wearable device, Collaboration, Middleware

1 引言

最近几年可穿戴设备大量涌现,以往受制于硬件的限制,往往做得比较笨重,仅能应用于实验室里的研究。近年来随着硬件技术的飞速发展,许多可穿戴设备已经变得十分小巧,涌现出了大量的具有实用价值的可穿戴设备。如今可穿戴设备已经涉及到了个人健康(如 Jawbone Up 手环、Nike+ Fuelband 腕带)、现实增强(如 Google 的 Google Glass)和智能交互(NeuroSky 的 Mindwave 脑电耳机)等多个领域。同时,智能手机的功能更加丰富,其联网能力、计算能力和存储能力已经逐渐接近传统 PC,智能手机成为了人们生活的计算中心和通信中心^[1]。如今的可穿戴设备能够通过蓝牙、WIFI 等方式连接智能手机,通过智能手机上的应用,可以可视化地看到来自可穿戴设备采集的数据。同时用户的个人数据能够备份到可穿戴设备开发商提供的云端服务器上,网络用户可以看到设备的历史数据,获得劝导激励的反馈。可穿戴设备提供给

用户一个数据感知、数据展示以及服务反馈的闭环。

但是现在用户随身的可穿戴设备越来越多,这些闭环都是相对独立的,设备与设备之间鲜有或者没有联系。每个设备都是一个孤岛,设备之间的数据和信息无法共享。即使是功能强大的并且能够访问全球信息网络的笔记本电脑,也是将注意力集中于一个单一的盒子中^[2],对于现在的可穿戴设备也是如此。虽然可穿戴设备越发强大,但是单个设备往往只能专注于某一领域,提供的功能有限。如果能把全身的可穿戴设备协同起来,则可以给使用者提供更优质的服务,并且手机需要与其他手机配合打造基于协同感知的新传感项目,如 Dartmouth 的 Smartphone Sensing 小组研究的 NeuroPhone 项目^[3]、HeartPhone 项目^[4]、EyePhone 项目^[5]。这些项目都是具体的设备协同的实现,比如 NeuroPhone 是神经信号传感器与智能手机协同,HeartPhone 是放在耳机上的传感器监测心率与智能手机协同,EyePhone 是用摄像头捕捉眼球状态信息与智能手机的协同。虽然已经有一些利用几类传

到稿日期:2014-06-12 返修日期:2014-09-01 本文受国家科技支撑计划(2012BAH94F03)资助。

丁 扬(1989-),男,硕士生,主要研究领域为普适计算,E-mail: baroque@zju.edu.cn;王淑刚(1989-),男,硕士生,CCF 学生会会员,主要研究领域为普适计算,E-mail: wshg@zju.edu.cn;李石坚(1979-),男,博士,副教授,CCF 会员,主要研究领域为普适计算、移动互联网、社会计算等,E-mail: shijianli@zju.edu.cn(通信作者);潘 纲(1976-),男,博士,教授,CCF 会员,主要研究领域为普适计算、计算机视觉、智能系统等。

感器协同工作来做一件事情的实现,但至今仍然缺少一个能够融合各类穿戴设备、把不同的穿戴设备与智能手机协同起来提供服务的平台。这就需要提出一些技术来支持智能手机和可穿戴设备的协同。首先,实现智能手机和可穿戴设备协同需要能够获取可穿戴设备的数据与服务,这需要服务的汇聚与开放机制;其次,在数据汇聚和开放的需求上,保护个人隐私变得尤为重要,这就需要开放授权机制;最后,若要支持各种可穿戴设备,则需服务的扩展机制。

本文分析了可穿戴设备协同所需要解决的问题;提出了支持可穿戴设备协同的协同模型;设计并实现了 Scudware Mobile 移动中间件,该中间件以智能手机为中心,实现了可穿戴设备的协同;基于 Scudware Mobile 移动中间件,设计和实现了个人 MobileTrace 足迹记录器,验证了 Scudware Mobile 移动中间件在可穿戴设备协同方面的性能及可用性。

2 相关工作

上面提到了 SmartPhone Sensing 小组的 NeuroPhone、HeartPhone 和 EyePhone 3 个项目,这 3 个项目都是试图以其它的交互方式替代传统的手指触摸的方式来控制手机。NeuroPhone 是用神经信号来操控手机,实现了用脑控制拨号等操作;HeartPhone 则是将测心率的传感器嵌入到耳机中,耳机再连到手机上,手机可以显示心率;EyePhone 是用摄像头通过监测眼珠的转动来获取操作手机的信息。这 3 个项目都体现了传感器设备与智能手机的协同作业。

Suranga Nanayakkara 等人设计的 EyeRing^[6]平台可以根据手指的不同指向触发不同的操作或任务,这些操作或任务可以由开发者自己设计和指定,实现了 EyeRing 与其它设备的协同工作。Kathrin Probst 等人提出的 Active Office 平台^[7]专注于办公室内各设备任务的协同以及无缝的迁移,该平台将办公室内的工作分成传统 desk、站立 desk 和 whiteboard 3 类,办公室内的设备按照这 3 类划分,实现了一个任务在这 3 类设备上的协同和无缝地迁移。Christian Berkhoff 等人设计的 Clairvoyance 框架^[8]实现了共享屏幕与移动终端设备的协同工作。Anna Maria Polli 等人实现了一个 Local Area Artworks^[9],该平台允许多个人协同完成一幅绘画作品,还允许其他人对作品进行评论和标记等操作,体现了多个人协同完成一项任务的思想。上面提到的平台和框架都是针对一类场景或者一项应用专门设计的协同平台或者框架,并不具有通用性或者支持其他类型设备的接入。

在物联网方面,研究者提出的一系列平台可以实现将各种设备接入互联网,实现数据或服务的共享。目前这些平台只是提供将设备接入互联网的方式,而没有提供这些设备之间直接通信的方式。Xively^[10]是一个通用云平台,任何接入 Xively 的设备都能进行相互间的直接通信。它帮助用户在世界范围内连接和共享来自物体、设备、建筑和环境的感应装置的实时数据,实现数据的共享。Open. Sen. Se^[11]则提供了各种封装好的基础 APP,可以接入 Arduino、Withings、Fitbit 等设备以及来自 Xively 的数据。这些数据汇聚平台专注于传感器等数据的汇集与共享,并没有提供可穿戴设备协同的解决方案,而且在这些平台上智能手机没有扮演重要角色。

国内对可穿戴设备的研究也很多,而且有一些研究已经

涉及到可穿戴设备间的协同问题。孙效华等人就可穿戴设备的交互设计方面^[12],从人机交互、界面设计、应用与服务 3 个方面进行了研究与论述。郑能干、吴朝晖等人则对电子织物这一类特殊的可穿戴设备的研究进展和发展趋势进行了论述与总结^[13],进而提出了一个统一的研究框架。徐强等人论述了可穿戴设备协同工作与普通计算机协同工作的不同,进而提出了可穿戴设备协同软件设计应该考虑的问题^[14]。

综上所述,目前还缺少一个能够接入各种可穿戴设备并提供可穿戴设备间数据和服务协同的统一平台。因此提出了支持可穿戴设备协同的协同模型,在此基础上设计并实现了 Scudware Mobile 移动中间件,以智能手机为中心,实现了可穿戴设备和手机的服务协同。

3 问题分析与协同模型的提出

3.1 问题分析

设备之间的协同即服务的组合,是指将若干个简单的原子服务合成新服务或组装成更加复杂的组合服务,来实现用户或应用程序的需求。由于可穿戴设备都是以人为中心的,根据协同的范围可以将协同划分成两类,一类是个人设备之间的协同模型,一类是人与人之间的协同。以智能手机作为以个人为单位的服务协同中心,这里一个智能手机代表一个人。

3.1.1 个人范围内设备之间的协同

个人设备之间的协同不会跨出智能手机。本地应用向移动中间件请求服务,移动中间件根据特定的流程,组合服务提供给上层应用。

3.1.2 人与人之间的设备协同

运行在智能手机上的移动中间件可以有访问权限的另一个人的移动中间件上请求自己没有的服务。这需要提供以下机制来实现人与人之间的设备协同:

1) 远程调用接口,以实现调用远程智能手机上管理的可穿戴设备提供的数据或服务。

2) 数据和服务开放授权机制,对于远程的调用,用户无法验证应用或者应用者的身份,所以需要远程应用进行授权。仅在远程应用可信时,才把服务授权给远程应用。

3.1.3 其它问题

为了把智能手机和可穿戴设备提供的服务接入平台,且屏蔽设备和服务的异构性,需要对服务进行抽象。抽象可以对外提供统一的接口,上层应用程序不涉及下层的具体实现。

由于需要支持各种可穿戴设备,每个可穿戴设备会提供多种服务,因此需要提供服务的扩展机制以实现各种可穿戴设备无缝接入协同平台。

3.2 协同模型

针对上面提到的可穿戴设备协同的问题和需求,提出了一个解决可穿戴设备协同的协同模型,如图 1 所示。

协同模型分成 4 个层次。最底层是设备抽象层,该层的主要工作是消除可穿戴设备的异构性,提供统一的服务访问接口;该层按照数据服务与应用服务划分首先对其进行抽象,消除不同设备的差异,然后提供统一的服务访问接口以供上层调用。第二层是服务管理层,该层的工作是管理可穿戴设备的各种服务,而且支持服务扩展机制,支持实时的服务注册

与更新。第三层是协同模型的核心层,首先依据服务开放与授权机制实现服务的远程访问,在此基础上实现跨协同平台的可穿戴设备服务的协同。协同模块处理具体的协同工作,其中,设备发现主要是发现附近其它协同平台及其管理的服务以供使用;服务组合可以把不同的服务组合起来提供新的服务。应用接口层实现服务的调用接口,分为本地调用接口和远程调用接口,分别用来访问本地服务和远程服务。本文提出的协同模型关键部分为:数据服务与应用服务抽象、本地调用和远程调用、开放与授权和协同模块。

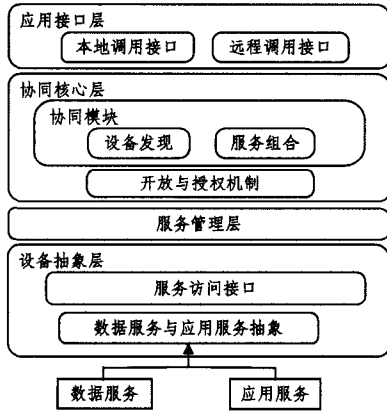


图1 协同模型

3.2.1 数据与服务抽象

可穿戴设备提供的数据与服务都是与设备本身相关联的,而且即便是对于同一种可穿戴设备,不同的厂商提供的接口标准也可能不尽相同,所以我们要做的工作就是将可穿戴设备提供的数据与服务从设备上抽象出来,消除它们与设备的关联,这里将数据与服务分为数据服务和应用服务。

数据服务是指智能手机和可穿戴设备作为数据的提供者,将数据获取的接口暴露成服务,提供给中间件或者上层应用使用。

应用服务和数据服务不同,不是以提供感知数据为主,而是为智能手机和可穿戴设备的计算能力或者反馈能力提供服务。

3.2.2 本地调用和远程调用

本地调用主要对应于数据服务或者应用服务在本地的情况,而远程调用主要对应于数据服务或者应用服务在远程的情况。

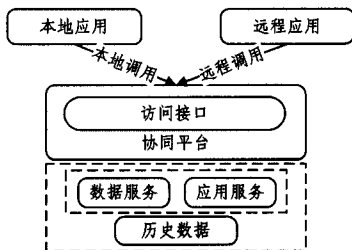


图2 数据和服务的调用接口

需要对上层应用做一个划分。根据应用所在的位置,将其分为本地应用和远程应用。本地应用是与协同平台运行在同一设备上的应用程序,本地应用采用操作系统提供的进程间通信的方式,通过协同平台提供的 API 进行服务调用。远程应用区别于本地应用,运行在协同平台所在的设备之外的

设备或者 Web 中,远程应用通过协同平台的 Web 接口获取服务,如图 2 所示。

3.2.3 开放与授权

本地应用是与协同平台运行在同一设备上的应用程序,本地应用一般由用户自己安装,使用者为用户本人,具有一定的可信程度。所以对于本地应用,协同平台的开放程度较高。用户可以信任所有的本地应用,允许应用访问所有的数据服务和应用服务,也可以对应用进行单独的授权。

区别于本地应用,远程应用通过协同平台的 Web 接口获取服务。对于远程应用,用户无法验证应用或应用使用者的身份,所以需要远程应用进行授权。当用户认为远程应用可信,并授权给远程应用时,远程应用才能访问到协同平台提供的数据和应用程序,如图 3 所示。

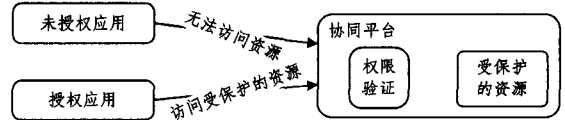


图3 开放与授权机制

3.2.4 协同模块

协同模块包括两部分的工作,即其它协同平台的发现与服务组合。

协同平台不仅要管理本地的服务,还要管理可以发现的其它平台上的服务,发现附近相同的协同平台,需要对每个协同平台进行唯一标识,比如可以使用用户账户进行标识。要发现附近可用的平台,需要利用设备发现协议,比如 SSDP 简单设备发现协议。

服务组合主要用来产生新的服务,服务的组合工作并不一定全由协同平台来做,服务组合的逻辑可以由服务制造者或者应用开发者来设计。

4 架构及关键技术

4.1 移动中间件架构

根据上文提出的协同模型,设计了一个 Scudware Mobile 移动中间件架构。中间件架构如图 4 所示。

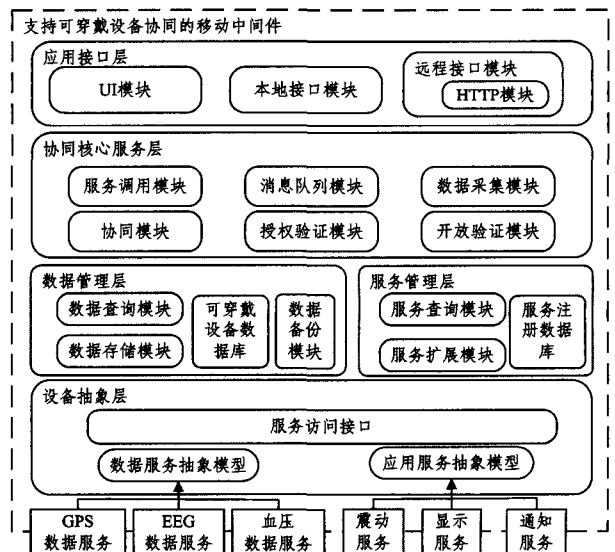


图4 Scudware Mobile 架构

通过分析,我们把移动中间件分为 4 层进行设计,从上到

下分别是应用接口层、中间件核心服务层、管理层以及设备抽象层。4层结构把移动中间件的各种功能很好地隔离开来,降低了系统的耦合度,方便了以后的扩展。

设备抽象层解决了智能手机和可穿戴设备服务的抽象问题,使这些设备的服务能够方便地接入到移动中间件中;同时提供了扩展性,支持后期服务的扩展。管理层对移动中间件采集的数据以及移动中间件的服务进行管理,对数据和服务进行持久化存储,提供一系列操作接口,供中间件的其他模块调用。核心服务层是数据服务移动中间件中最重要的一层,处理移动中间件中关于服务调用、数据集采等的核心逻辑。协同模块是整个中间件的核心模块,提供可穿戴设备协同的相关操作与处理。应用接口层把移动中间件的接口暴露给使用者和上层应用。通过应用接口层,可以使用中间件核心服务层提供的功能。

根据上面的问题分析,要实现个人范围之内的设备协同和人与人之间的设备协同,需要解决的关键技术问题主要有:数据服务与应用服务抽象,只有将可穿戴设备提供的服务进行抽象,才能屏蔽不同设备的异构性,方便对外提供统一接口;服务的插件化扩展,要支持可穿戴设备方便地接入到移动中间件,就要支持可穿戴设备服务的插件化扩展;提供服务的本地访问接口,以实现个人范围之内的服务访问;提供服务的远程访问接口,以实现人与人范围的服务访问;提供服务的开放授权机制,以保证远程访问服务的安全;最后要实现的核心技术就是协同机制。

4.2 数据服务与应用服务抽象

要屏蔽设备和服务的异构性,就要对服务进行抽象。对外提供统一的接口,上层应用不用涉及下层的具体实现。

定义一个继承 Android 中 Service 类的类 DeviceService,该类实现了一个消息队列,用来处理和分发内部或者外部发给 DeviceService 的消息,该类还提供了服务生命周期的方法。数据服务与应用服务都继承该类。对于数据服务和应用服务,分别定义了 AIDL 接口 IDataService 和 IApplicationService 作为服务接口。AIDL 是一种比 Broadcast 效率更高的进程间通信方式^[15]。

实现一个服务时,要继承 DeviceService 类并且实现 IDataService 或者 IApplicationService 接口。

在连接设备之后,要使用消息队列接收设备发送的消息,需要使用 recentData 对象来缓存最新的数据,在移动中间件调用 getData 接口后,把缓存的 recentData 返回。具体实现方法依设备的不同而不同。

对于服务所提供的内容的描述,采用了 JSON 格式进行服务描述。同时服务的通信数据格式也都统一采用 JSON 格式进行定义。

4.3 服务的插件化扩展

Scudware Mobile 需要支持各种可穿戴设备,并且各种设备可以即时加入到 Scudware Mobile 中,这就需要服务的动态更新,而 Android 系统本身不支持程序的动态更新。基于此提出了一种插件化的扩展方式。可穿戴设备的服务可以被打包成 APK 格式文件,使服务的动态安装变得可能。

针对服务插件,制定了描述规范,所以一个服务包含了

APK 安装包和描述文件。服务若要被安装使用,服务插件需先在协同服务器上进行注册,提交服务描述,放在协同服务器上托管。协同服务器提供插件的上传、下载以及信息录入和查询功能。当移动中间件需要安装一个服务时,可以从服务器上查询并下载,将其安装到中间件所在的手机上以供使用。

移动中间件的插件管理模块中有一个系统广播 InstallReceiver,可以监听应用程序的安装与卸载。当监测到 PACKAGE_ADDED 应用安装广播后,中间件向服务器查询安装的这个应用是否在服务器上有记录,如果有,则证明安装的是一个服务插件,就从服务器获取服务信息,进行服务注册;如果查询不到这个应用,则说明这是手机的一个普通应用安装,不进行操作。

4.4 服务访问接口

服务访问接口分为本地访问接口和远程访问接口。本地访问接口访问描述信息,以 JSON 格式定义,调用的触发方式采用的是 Handler 和 Message 机制。

远程调用接口则需要数据与服务的开放,以供他人使用。为了使数据与服务开放,采用了 RESTful 架构组织数据与服务。REST 概念中资源是核心^[16],我们将数据与服务都看成统一的抽象资源。定义了一套 RESTful API,以方便远程访问数据与服务资源。本文实现了一个轻量级的 http server 来实现远程接口,如图 5 所示。

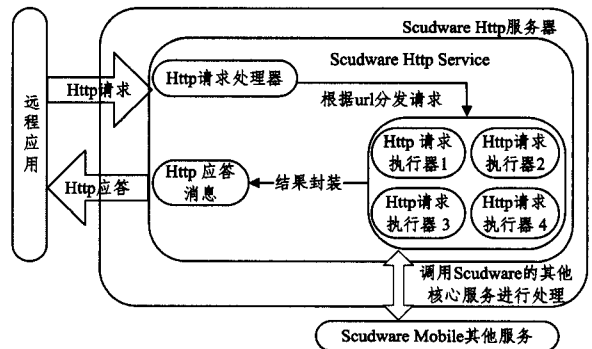


图 5 远程接口模块结构

Scudware Http 服务器监听 HTTP 请求。当收到请求后,由一个 Http Service 线程处理请求。Http Service 线程通过解析请求的 URL,将请求分发给不同的 HttpHandler 进行处理。HttpHandler 通过进程间通信的方式与移动中间件的其他模块进行交互,获得需要的服务。最后 HttpHandler 把结果封装在响应对象中返回。

4.5 数据和服务授权机制

在 Scudware Mobile 中,不同应用的开放策略不同。

本地应用被认为是可信应用,允许访问所有的数据和应用服务,也可以对应用进行单独授权。远程应用通过 Web 接口获取服务,无法验证身份,所以需要远程应用进行授权。

采用 OAuth2.0^[17] 标准实现数据和服务的授权,OAuth 允许用户授权给第三方应用一个具有一定有效时间的令牌,只有持有令牌的第三方应用才能访问私密资源。我们对 OAuth 认证过程进行了修改。

当第三方或其他移动中间件对用户开放的服务进行请求时,请求消息发送到协同服务器,消息会被保存在协同服务器上。协同服务器以消息的形式通知用户,用户在收到消息之后可以马上进行认证或者推迟认证。当用户进行授权许可之

后,继续进行 OAuth2.0 授权码认证,这样就完成了用户授权,具体的流程如图 6 所示。所有权限的数据保存在协同服务器上。

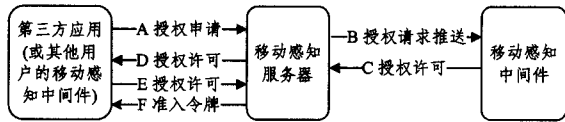


图 6 授权过程

对于个人的隐私数据,采取了有选择性的加密策略。移动中间件中有两种数据,一种是实时数据,这部分不需要加密;另一种是持久化存储的历史数据,这部分数据有加密的需要。提供了三级加密策略:不加密、可选加密、全部加密。加密算法采用的是 RSA 非对称加密算法,数据用私钥加密,访问者用公钥解密。数据加密策略在一定程度上保护了用户隐私安全。

4.6 协同机制

移动中间件的设备协同是在数据和服务的开放和授权的基础上实现的。

设备协同的第一步就是附近相同的移动中间件的发现。采用了 SSDP 简单设备发现协议^[18],根据协议在 239.255.255.250;1900 这个多播地址进行服务发现。在 SSDP 中,当设备加入局域网时,需要发送 ssdp:alive 消息;当设备需要发现服务时,可以向多播地址发送 ssdp:discover 消息,消息格式如表 1 所列,当其他设备接收到 ssdp:discover 消息后,就会发送 ssdp:alive 消息,如表 2 所列。

表 1 ssdp:discover 消息格式

M-SEARCH * HTTP/1.1
HOST:239.255.255.250;1900
MAN:"ssdp:discover"
MX:延迟响应时间
ST:搜索目标

表 2 ssdp:alive 消息格式

HTTP/1.1 200 OK
CACHE-CONTROL:过期时间
DATE:响应产生时间
LOCATION:设备描述地址
SERVER:OS/Version UPnP/1.0 product/version
ST:搜索目标
USN:UUID

在移动中间件的协同机制中,只采用了 SSDP 协议进行设备发现以及 UPnP 中的设备来描述解析的过程,没有使用 UPnP 中后续的服务描述获取以及服务解析和调用等过程,服务解析以及调用参见 4.4 节中设计的 RESTful API。SSDP 协议以及 UPnP 协议的实现使用了 Cling^[19] 开发包。我们定义了新的 UPnP 设备类型-ScudwareMobile,移动中间件只对这类设备进行发现以及响应,忽略其他 UPnP 设备的消息,设备通过 IP 以及用户账户进行描述。在设备发现之后,通过移动中间件的开放和授权等接口,就能向该移动中间件请求调用数据服务和应用服务。移动中间件可以把来自其他移动中间件的服务作为远程服务在本机进行注册,上层应用可以通过移动中间件像调用本地服务一样使用来自其他用户开放的服务。

4.7 Scudware Mobile 移动中间件的特色

根据上文的架构及关键件技术,设计并实现的 Scudware

Mobile 移动中间件具有以下特色:

1)支持服务的插件化扩展,可穿戴设备的服务安装后自动注册,不需要其他操作即可调用,实现了“安装即使用”的效果。

2)以服务的开放授权机制控制可穿戴设备协同,只有经过授权的应用才可以远程调用服务。

5 移动中间件的性能测试及应用

5.1 移动中间件性能实验

移动中间件运行于设备和上层应用之间,它提供了访问本地服务和远程服务的统一接口。但在解决这些问题的同时,需要保证移动中间件的性能。所以设计了实验对移动中间件的性能进行测试。

在实验中,系统的运行环境为 Android 4.1.1 系统,手机配置为 CPU 双核 1741MHz,内存 1GB。使用了 5 个接入移动中间件的服务进行测试,这 5 个服务都是具有代表性的服务,如表 3 所列。

表 3 实验中采用的服务

序号	服务名	说明
1	ccnt.pvc.scudwaremobile.GyroscopeService	陀螺仪
2	ccnt.pvc.scudwaremobile.BatteryService	电量
3	ccnt.pvc.scudwaremobile.AccelerometerService	加速度
4	ccnt.pvc.scudwaremobile.WifiInfoService	Wifi 扫描结果
5	ccnt.pvc.scudwaremobile.EEGAttentionService	脑电-专注度

实验主要测试移动中间件支持调用本地服务和远程服务的性能。

针对调用本地服务的性能测试,基于移动中间件设计了一个简单的应用,利用移动中间件调用本地的服务,模拟在 1s 内产生 50 次、100 次、150 次服务请求,另外有一组实验在 30s 内产生 1500 次请求。实验的另一个变量是应用调用的服务个数 C ,发起的 N 次请求随机调用 C 个服务中的一个。加入服务个数 C 是为了衡量多个服务调用对系统性能的影响。实验结果如表 4 所列。

表 4 本地调用性能实验结果

服务个数	调用数			
	50 次/1s	100 次/1s	150 次/1s	1500 次/30s
1	14ms	22ms	26ms	15ms
3	15ms	20ms	23ms	14ms
5	18ms	26ms	40ms	18ms

单独测试得出单次本地调用的平均响应时间为 9ms。从结果分析可知,在多线程调用的情况下,本地接口的响应时间理想,在 150 次/s 请求的情况下,响应时间为 26ms,并且能够承受较高的并发量,这样的并发量对于手机应用已经足够。从服务个数 C 的变化而对调用时间产生的影响可以得出目前系统响应时间的瓶颈在于应用接口层的本地接口模块。结果显示:移动中间件的本地接口以及底层模块的逻辑处理和 AIDL 接口调用可以满足本地应用的使用。

在远程服务调用性能实验中,衡量的指标与本地服务调用实验所采用的指标相同,远程调用的实验在 WIFI 环境下进行,使用 TP-link WR740N 路由器,带宽为 150M。实验结果如表 5 所列。

表 5 远程调用性能实验结果

服务个数	调用数			
	50 次/1s	100 次/1s	150 次/1s	1500 次/30s
1	44ms	145ms	437ms	40ms
3	29ms	130ms	345ms	28ms
5	24ms	50ms	227ms	22ms

单独测试得出单次远程调用的平均响应时间为 12ms。在并发数较低的情况下,服务远程调用的响应时间理想。随着并发请求数目的增加,响应时间变化明显,但是响应结果正确。对于手机远程服务来说,不会有像服务器一样的远程并发访问。我们认为这样的并发量和响应时间已经能够满足远程服务的提供以及设备协同的要求。

分析移动中间件在支持本地服务调用和远程服务调用的两个性能实验中的结果可以得出结论:通过移动中间件调用本地服务和远程服务不会影响服务的响应时间。

5.2 移动中间件的应用

依托移动中间件,提出并实现了 MobileTrace 足迹记录器应用,它可以根据不同的上下文环境选择调用本地位置服务或者附近中间件的位置服务来定位和追踪用户的行走足迹。

个人的位置信息是十分重要的感知数据,通过位置轨迹的分析可以挖掘用户的兴趣,针对用户所处的位置可以进行个性化的推荐。

MobileTrace 是一个记录用户足迹的应用,它定期采集用户的位置信息并进行记录。MobileTrace 向移动中间件请求用户位置信息,移动中间件根据情况自动选择用户的位置服务推荐给它使用。当位于室内时,移动中间件采用 WIFI 或者信号塔进行定位;当位于室外时,移动中间件调用 GPS 定位服务。MobileTrace 与移动中间件的调用如图 7 所示。

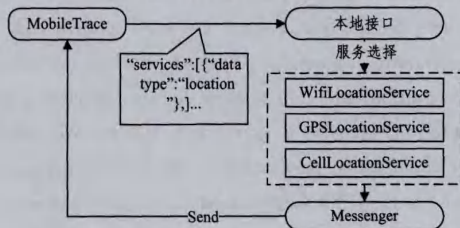


图 7 MobileTrace 和 Scudware Mobile 交互过程

同时,借助于移动中间件,MobileTrace 能够使用附近的用户提供的定位服务。通过对方的授权后,MobileTrace 就能获得依靠临近用户的位置服务来记录的位置信息。图 8 展示了 MobileTrace 记录的用户足迹。

MobileTrace 足迹记录器应用展示了 Scudware Mobile 移动中间件在处理多个中间件管理的设备层面上服务协同的能力。MobileTrace 可以不用处理调用本地的服务还是远程的服务,这些工作都交给 Scudware Mobile 移动中间件来处理,Scudware Mobile 如果检测到本地有相应的服务(定位服务),则直接调用本地服务;如果本地没有相应的服务(定位服务),则通过设备发现来检测附近 Scudware Mobile 是否提供相应的服务(定位服务)。如果找到,则通过远程调用接口调用远程的服务,而这些工作都不需要 MobileTrace 来做,这体现了 Scudware Mobile 具有使不同可穿戴设备服务协同工作的能力。



图 8 MobileTrace 记录的一次足迹

通过上面对移动中间件的性能测试,验证了移动中间件在支持本地服务调用和远程服务调用方面对服务响应时间的影响极小,在可以接受的范围之内;通过基于移动中间件开发的 MobileTrace 应用,验证了移动中间件的可用性,也体现了移动中间件的应用潜力。

结束语 智能手机和可穿戴设备的发展,为可穿戴设备的协同带来了突破,但可穿戴设备的协同还面临着许多的挑战。本文分析了当前可穿戴设备协同领域存在的问题,提出了支持可穿戴设备协同的协同模型,设计并实现了 Scudware Mobile 移动中间件。本文的主要研究内容有:

1)分析了当前可穿戴设备协同研究和可穿戴设备协同应用开发存在的问题,并总结了可穿戴设备协同开发框架和数据汇聚平台的相关工作。

2)提出了支持可穿戴设备协同的移动中间件的解决方案和协同模型,分析了其主要特点,并从数据和应用服务模型、开放与授权机制以及协同 3 个方面对其进行阐述。

3)设计并实现了 Scudware Mobile 移动中间件,这是一个 4 层架构,包括设备抽象层、管理层、核心服务层和应用接口层。Scudware Mobile 汇聚智能手机和可穿戴设备的数据和服务,以统一的接口开放给上层应用,采用授权的机制保护用户隐私,并支持设备间的数据和服务协同。

4)在 Scudware Mobile 移动中间件的基础上,设计了两个测试移动中间件性能的实验,验证了移动中间件的高效性;设计并实现了 MobileTrace 应用,体现了移动中间件的概念,并验证了其可用性。

下一步的工作有:

1)由于 Scudware Mobile 移动中间件是一个体现可穿戴设备间数据和服务协同的原型系统,在系统上 Scudware Mobile 还存在着一些问题,需要加强系统的健壮性,丰富上层应用的 API。

2)由于 Scudware Mobile 是运行在智能手机上的中间件系统,需要测试耗电和 CPU 占用率等其他性能,借此对中间件进行优化。

3)目前实现的协同应用只是移动中间件几个概念的体现,希望在下一步的工作中,在 Scudware Mobile 移动中间件的基础上开发更丰富的服务和感知应用。

(下转第 32 页)

块进行计算,相同特征点个数会反映出不同的计算时间,时间较低的为特征点密集型的图像。

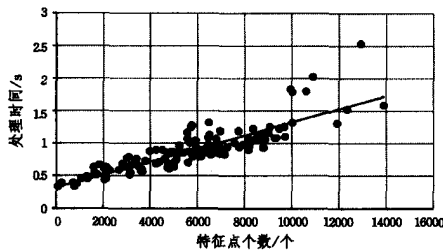


图4 计算时间与特征点数目散点图

结束语 本文提出了一种基于 SIFT 的高分辨率遥感图像的配准并行加速算法,分别利用 CUDA 架构以及 OpenMP 多线程技术对特征点提取以及匹配过程进行了加速优化,提高了传统 SIFT 算法的配准速度。

参考文献

[1] Zitova B, Flusser J. Image registration methods; a survey[J]. Image and Vision Computing, 2003, 21: 997-1000

[2] Li Qiao-liang, Wang Guo-you, Liu Jian-guo. Robust scale-invariant feature matching for remote sensing image registration[J]. IEEE Geoscience and Remote Sensing Letters, 2009, 6(2): 187-291

[3] Zhang Yun-sheng, Zhou Pei-long, Ren Yue, et al. GPU-accele-

rated large-size VHR images registration via coarse-to-fine matching[J]. Computers and Geosciences, 2014, 66: 54-65

[4] 雷小群,李芳芳,肖本林.一种基于改进 SIFT 算法的遥感影像配准方法[J].测绘科学,2010,35(3):143-145
Lei Xiao-qun, Li Fang-fang, Xiao Ben-lin. A registration method of RS image based on improved SIFT algorithm[J]. Science of Surveying and Mapping, 2010, 35(3): 143-145

[5] Kirk D B, Wen-mei W. Programming massively parallel processors; a hands-on approach[M]. Morgan Kaufmann, 2010

[6] Dagum L, Menon R. OpenMP; an industry standard API for shared-memory programming[J]. Computational Science & Engineering, 1998, 5(1): 46-55

[7] Lowe D G. Distinctive image features from scale-invariant keypoints[J]. International Journal of Computer Vision, 2004, 60(2): 91-110

[8] CUDA C Programming Guide[OL]. <http://docs.nvidia.com/cuda/cuda-C-programming-guide/#axzz3iTPutLEx>

[9] Nvidia CUDA Computer Unified Device Architecture[S]. Programming Guide, Version 2.0 beta 2, 2008

[10] 周海芳,赵进.基于 GPU 的遥感图像配准并行程序设计与存储优化[J].计算机研究与发展,2012,49(S1):281-286
Zhou Hai-fang, Zhao Jin. Parallel programming design and storage optimization of remote sensing image registration based on GPU[J]. Journal of Computer Research and Development, 2012, 49(S1): 281-286

(上接第 23 页)

参考文献

[1] Lane N D, Miluzzo E, Lu H, et al. A survey of mobile phone sensing[J]. Communications Magazine, IEEE, 2010, 48(9): 140-150

[2] Weiser M. The computer for the 21st century[J]. Scientific american, 1991, 265(3): 94-104

[3] Campbell A, Choudhury T, Hu S, et al. NeuroPhone: brain-mobile phone interface using a wireless EEG headset[C]//Proceedings of the Second ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds, ACM, 2010: 3-8

[4] Poh M Z, Kim K, Goessling A D, et al. Heartphones: Sensor earphones and mobile application for non-obtrusive health monitoring[C]//International Symposium on Wearable Computers, 2009 (ISWC'09). IEEE, 2009: 153-154

[5] Miluzzo E, Wang T, Campbell A T. Eyephone: activating mobile phones with your eyes[C]//Proceedings of the Second ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds, ACM, 2010: 15-20

[6] Nanayakkara S, Shilkrot R, Yeo K P, et al. EyeRing: a finger-worn input device for seamless interactions with our surroundings[C]//Proceedings of the 4th Augmented Human International Conference. ACM, 2013: 13-20

[7] Probst K, Perteneder F, Leitner J, et al. Active office: towards an activity-promoting office workplace design[C]//Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts. ACM, 2012: 2165-2170

[8] Berkhoff C, Ochoa S F, Pino J A, et al. Clairvoyance: A framework to integrate shared displays and mobile computing devices

[J]. Future Generation Computer Systems, 2014, 34: 190-200

[9] Polli A M, Korn M, Nylandstedt Klokmose C. Local area artworks; collaborative art interpretation on-site[C]//Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication. ACM, 2013: 79-82

[10] Xively[OL]. <https://xively.com/>

[11] Open. Sen. Se[OL]. <http://open.sen.se/>

[12] 孙效华,冯泽西.可穿戴设备交互设计研究[J].装饰,2014(2): 28-33
Sun Xiao-hua, Feng Ze-xi. Interaction Design for Wearable Devices[J]. Zhuang Shi, 2014(2): 28-33

[13] 郑能干,吴朝晖,林曼,等.电子织物研究进展[J].计算机学报, 2011, 34(7): 1172-1187
Zheng Neng-gan, Wu Zhao-hui, Lin Man, et al. A Survey on electronic textiles[J]. Chinese Journal of Computer, 2011, 34(7): 1172-1187

[14] 徐强,孙建华.可穿戴多人协同支撑软件系统研究[J].电子科技大学学报,2010(S1):57-60
Xu Qiang, Sun Jian-hua. Research on Wearable Computing in the Multi-collaborative of System[J]. Journal of University of Electronic Science and Technology of China, 2010(S1): 57-60

[15] Brunette W, Sodr R, Chaudhri R, et al. Open data kit sensors; a sensor integration framework for android at the application-level [C]//Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services. ACM, 2012: 351-364

[16] Fielding R T. Architectural styles and the design of network-based software architectures[D]. University of California, 2000

[17] OAuth[OL]. <http://oauth.net/>

[18] UPnP Device Architecture V1.1[OL]. <http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf>

[19] Cling[OL]. <http://4thline.org/projects/cling>