基于代理的移动云服务访问机制的研究与实现

张拥军 史殿习 肖 玺 吴振东 丁 博
(国防科学技术大学计算机学院 长沙 410073)

摘 要 移动设备的计算能力、存储能力、网络带宽、电池续航等都非常受限,而且现有的云服务基本没有针对移动设备的上述特点做优化,严重降低了移动设备访问云服务的质量。针对存在的问题,提出了面向移动设备的数据传输格式优化模型和服务Mashup模型,设计并实现了一个基于代理的移动云服务访问框架,使得移动设备通过代理服务器来访问云服务。在4种不同情况下对移动设备的服务请求响应时间、响应结果处理时间、请求响应流量和能耗分别进行了测试,结果表明实现的框架具有一定的实际意义。

关键词 移动计算,代理,云服务,服务Mashup
中图法分类号 TP311 **文献标识码** A

Research and Implementation of Mobile Cloud Services Access Mechanism Based on Proxy

ZHANG Yong-jun SHI Dian-xi XIAO Xi WU Zhen-dong DING Bo
(School of Computer Science, National University of Defense Technology, Changsha 410073, China)

Abstract The mobile devices are usually limited in its computing cabability, storage capacity, network bandwidth and battery life. Existing cloud services are not optimized for those features of the mobile devices. As a result, the quality of the mobile devices to access cloud services is seriously reduced. Aiming at this problem, a data transmission format optimization model and a servics mashup model were presented for mobile devices. And a framework of mobile cloud services access based on proxy was designed and implemented, which enables mobile devices to access cloud services through a proxy server. Under four different situations, the request response time, response process time, bandwidth flow and energy consumption of the mobile devices were tested. The test result shows that this framework has some practical significance.

Keywords Mobile computing, Proxy, Cloud service, Service mashup

1 引言

随着移动设备和移动网络的普及,移动设备以按需、易扩展的方式访问云服务成为可能。然而,移动设备与传统的PC相比,无论是CPU处理能力、内存大小、电池容量,还是网络通信能力,均具有较大差距。因此,在移动设备资源和无线网络带宽受限的条件下,如何高效便捷地访问云服务成为迫切需要解决的关键问题之一。

现有的大部分云服务没有考虑到移动客户端的特性,采用传输数据量大的SOAP(Simple Object Access Protocol,简单对象访问协议)协议和解析复杂的XML数据传输格式等,使得移动设备带来较大的计算资源消耗和网络带宽占用。在计算资源方面,相比同时期的PC而言,移动设备在处理能力和内存上分别约等于对方的三至八分之一[1],这就使得移动设备对复杂过程的处理成为瓶颈(如XML数据格式的解析[2])。因此,需要考虑将移动设备对云服务访问的部分计算、存储处理任务转移到固定基础设施上,以后台方式完成。在网络传输方面,XML仍然是云服务用于数据传输的主流格式,然而其数据传输量大,在带宽受限的无线网络环境下可能会造成带宽浪费。如何以数据量更小的数据传输格式(如JSON、对象序列化等)来进行传输,同时增强服务响应结果的优化处理,也是移动云服务访问所面临的挑战。

移动设备还需要考虑如何个性化、便捷地实现云服务访问。移动用户经常会访问多个云服务来完成一项任务,移动设备如果要与每个云服务分别建立连接请求,势必会造成移动设备和无线网络的资源和带宽浪费。可以考虑通过代理服务器来完成多个云服务的Mashup,然后再向移动设备暴露新的服务。这样,移动设备只需跟代理服务器建立一次连接请求便可便捷地完成多个云服务的访问。

综合上述分析与考虑,本文对移动设备访问云服务进行了研究,基于代理的思想设计并实现了一个综合移动设备、代理服务器的云服务访问框架,即通过代理服务器实现移动设备对云服务访问的优化,为开发人员提供响应速度更快、访问流量更省和服务使用更便捷的应用开发框架。本文第2节描

到稿日期:2012-07-08 返修日期:2012-10-22 本文受国家新一代宽带无线移动通信网专项课题(2011ZX03002-004-01)资助。
张拥军(1972—),男,博士,副研究员,主要研究方向为分布计算、移动云计算,E-mail:yjzhang@nudt.edu.cn;史殿习(1966—),男,博士,研究员,主要研究方向为分布计算、普适计算、移动云计算;肖 玺(1984—),男,硕士,主要研究方向为移动云计算;吴振东(1986—),男,博士生,主要研究方向为移动云计算、分布计算;丁 博(1978—),男,博士,助理研究员,主要研究方向为分布计算、软件适应。

述了与移动云服务访问的相关工作；第 3 节描述了基于代理的移动云服务访问研究中的关键技术，包括总体架构、数据传输格式优化机制和基于代理的服务 Mashup 机制；第 4 节详细描述了本文设计并实现的基于代理的移动云服务访问框架；第 5 节对本文实现的框架进行了应用验证和性能评测；最后对本文进行了总结。

2 相关工作

针对移动设备资源受限的问题，Ioana[3] 采用 AlfredO[4] 技术和 R-OSGi[5] 技术实现了一个移动应用迁移框架，该框架对应用的各个模块间的依赖关系建模，在此基础上进行最优化划分，并自动地将应用按不同的层次分布到移动设备和云计算上执行，从而实现移动设备的资源向云计算扩展。文献[6]则提出一种弹性应用模型，该模型将单个应用划分成多个 Weblet 的构件，以支持 Weblet 构件动态地在云或移动设备上执行。

针对无线网络带宽紧缺且不稳定的问题，Jason[7] 在分析现有智能移动设备的基本特性以及 RESTful 技术的特性后，使用 RESTful Web-Services 与云服务创建了下一代移动应用，因为 REST(Representation State Transfer，表述性状态转移)请求及响应是纯 HTTP 的，较传统的基于 SOAP 协议的 Web 服务而言，它的传输不含多余的协议成分，更加节省网络带宽。在某些应用场景下对远端云服务访问时，无线网络可能不稳定或者无法接入，为此文献[8]通过虚拟机技术为移动设备提供可快速定制应用程序服务的 Cloudlet，在移动设备与 Cloudlet 之间建立高速的无线局域网来解决无线网络不稳定、接入失败等问题。

Qian[9] 在他的硕士论文中提出了一个用于连接移动设备和云服务的跨平台框架 MCC(Mobile Cloud Computing)，该框架包括一个平台独立的移动设备客户端和一个运行在云上的中间件，通过中间件技术来为移动设备提供便捷的云服务访问能力，而且该中间件还为用户提供个性化服务 Mashup。

本文针对移动设备直接访问云服务时产生的网络带宽低、设备资源受限、云服务便捷访问等问题，对移动设备访问云服务的模式进行了深入研究，设计了一种基于代理的移动云服务访问框架，在该框架中，实现了数据传输格式优化技术、多服务访问的 Mashup 技术。

3 基于代理的移动云服务访问关键技术

基于代理的移动云服务访问的基本思想是在移动设备访问端与云服务之间设立代理服务器，代理服务器的作用是将来自移动端的访问进行相应处理后转发给具体的云服务。在基于代理的移动云服务访问中涉及的主要关键技术有数据传输格式优化技术、面向移动设备的服务 Mashup 技术，本节将描述框架实现中如何应用这两种技术。

3.1 基于代理的云服务访问模式

移动云计算环境下移动设备资源和无线网络带宽受限，使得移动设备访问云服务时通常会带来较大的无线网络通信成本和移动设备资源消耗；同时，移动设备 Mashup 也会因多次服务的连接请求造成移动设备资源以及无线网络带宽的浪费；而且，移动计算环境时常变化，从而使得传统的请求/响应式服务使用模式已不能很好地满足移动用户所需等，为此本文引入了基于代理的移动服务访问模式，即在传统的移动设备到云服务的二层访问架构之间加入中间代理层，由中间代理层来代理移动设备的请求或云服务的响应，如图 1 所示。一方面，移动设备通过中间代理对云服务进行请求，并将对云服务请求的部分处理过程转移到资源更为充裕的中间代理完成，从而节省移动设备的计算、存储等资源消耗；另一方面，云服务通过中间代理向移动客户端进行响应，并借助中间代理对云服务进行传输协议、数据交换格式等桥接转换，从而使其更适宜于资源受限的移动设备访问。为此，中间代理需要解决的问题包括：数据传输格式转换、数据传输格式自定制、远端云服务封装、服务 Mashup 等。

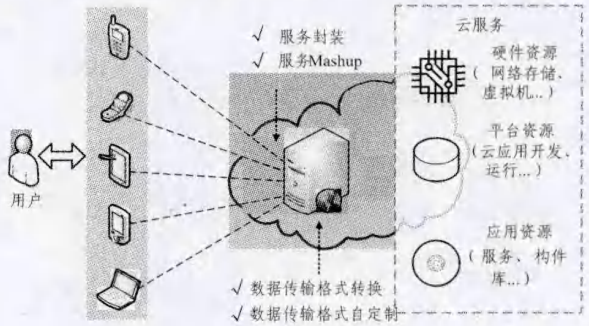

图 1 基于代理的云服务访问模式

3.2 数据传输格式优化机制

数据传输的主要格式有基于标记的 XML 和基于键值对的 JSON，其中 JSON 数据格式具有解析简单、空间占用少等优势。受移动设备资源、无线网络带宽等制约，XML 格式解析复杂且较占网络带宽，并不适合在无线网络环境下传输。但是现有的云服务在传输数据时，大多数都采用 XML 格式。为此，本文提出一种数据传输格式优化机制，将与移动设备交互的 XML 格式的数据转换成 JSON 格式，从而解决了移动设备解析 XML 格式效率低下的问题，减少了移动设备的网络传输流量。

在数据传输格式优化机制中，代理服务器在将数据传输格式为 XML 的云服务向移动设备响应时，将其转换成轻量级的 JSON 数据格式，再由模型进一步对 JSON 数据格式进行响应结果重定制，并最终返回给移动客户端。图 2 描述了数据传输格式优化模型的每一个过程(与图中序号相对应)。

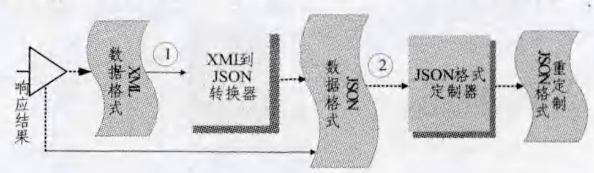

图 2 数据传输格式优化机制

(1)XML 到 JSON 的格式转换过程。根据云服务响应结果的数据格式进行判断，如果服务响应结果为 XML 数据格式，则经 XML 到 JSON 转换器将其转换成 JSON 格式，否则不作处理。

(2)JSON 格式重定制过程。对于云服务直接返回的或经转换器转换而来的 JSON 格式，开发人员借助 JSON 格式定制器可以完成 JSON 格式响应结果的重定制，比如对部分响应结果的删减、重塑，进一步减少代理服务器发送至移动设备的网络流量。

3.3 基于代理的服务 Mashup 机制

服务 Mashup 通常被设计用来满足移动用户复合型应用和个性化应用的需求，它不仅支持将多组服务响应结果融合在一张视图中供用户浏览，而且允许移动用户根据移动设备当前上下文来获取不一样的服务，比如基于地理位置的团购服务。这样不仅提高了移动用户的使用体验，也减少了用户在多张服务视图之间切换所带来的客户端资源及网络带宽的浪费。

本文在代理服务器上实现了 Mashup 服务机制，它能够动态地实现 Mashup 服务。图 3 描述了代理服务器上由接收 Mashup 服务描述信息到动态生成 Mashup 服务实现的每一个过程(与图中序号相对应)。

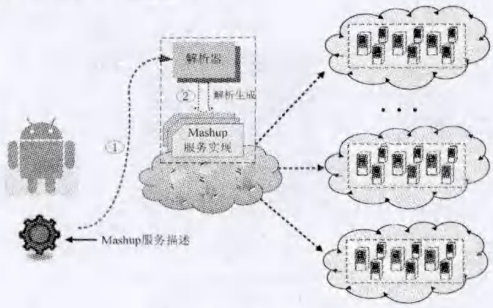

图 3 代理的服务 Mashup 机制

(1)Mashup 服务部署过程。移动设备上的应用如果需要一个新的 Mashup 服务，即在代理服务器上还不存在这种 Mashup 服务，则移动设备会将相应的 Mashup 服务接口描述文件发送至代理服务器，完成 Mashup 服务部署。

(2)Mashup 服务动态解析生成过程。代理服务器在接收到 Mashup 服务接口描述文件后，由解析器负责将其动态地解析生成对应的 Mashup 服务实现，即通过组装描述文件中所描述的各种云服务。

4 基于代理的移动云服务访问框架实现

基于代理的移动云服务访问框架包括运行在移动设备客户端的 MobileSDK 和代理服务器的 ProxyFramework 两部分，如图 4 所示。

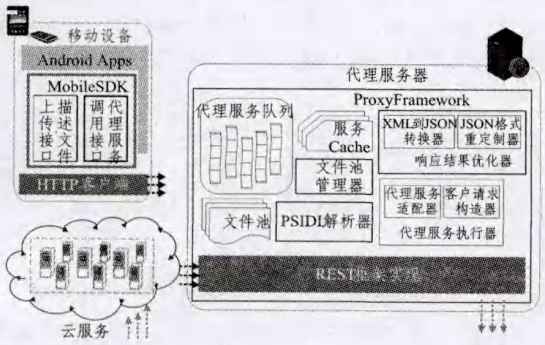

图 4 基于代理的移动云服务访问框架

(1) ProxyFramework 涵盖了数据传输格式优化、服务 Mashup 等功能实现，具体又可分为 XML 到 JSON 转换模块、JSON 格式重定制模块、服务 Mashup 动态生成模块、服务 Mashup 代理请求模块。ProxyFramework 需要 JDK 1.5 以上的版本支持，为服务 Mashup 提供了部署、运行及管理的支撑环境。

(2)MobileSDK 则为移动应用开发人员提供了软件开发接口，如描述文件上传接口、代理服务调用接口和业务规则写入接口等。目前，MobileSDK 只对 Android 2.2 版本及以上的操作系统提供支持。

在图 4 中移动终端部分，描述文件上传接口将 PSIDL 描述的服务接口文件上传至代理服务器，然后，由代理服务器上的解析器负责将其解析生成对应的服务实现，以供移动设备客户端访问。代理服务调用接口使得移动设备客户端能够对部署在代理服务器上的服务进行访问，然后由代理服务器代理移动设备完成远端云服务的访问。代理服务器在接收到客户端服务请求后，将根据请求的服务名称在代理服务队列中进行匹配，一旦匹配成功则从队列中获取云服务访问所需的相关参数并构造一条完整、合法的 URL 来实现对云服务的访问。在获得云服务响应结果后，代理服务器将通过响应结果优化模型对其进行优化处理，使其以一种更适宜移动计算环境的数据格式返回至移动设备。

在图 4 中代理服务器部分，响应结果优化器实现数据传输格式的优化，它包括：XML 到 JSON 转换器，用于云服务响应结果的数据格式的转换；JSON 格式定制器，用于对云服务响应结果的 JSON 格式重定制。PSIDL 解析器用于解析移动终端上传的 PSIDL 服务描述文件。代理服务执行器包括两个部分：代理服务适配器，用于对移动设备端请求的代理服务进行适配，以找到相应的目标服务。客户请求构造器，用于构造云服务客户请求，并完成云服务的访问。服务 Cache，用于缓存云服务的响应结果，从而加速代理服务器中间件对云服务的访问。代理服务队列，用于维护云服务访问的客户请求队列。以队列的形式维护来自客户端的请求，然后按照先来先服务的原则依次对每个请求进行响应。文件池用于存储客户端上传的代理服务描述文件。文件池管理器用于管理移动客户端上传的代理服务描述文件。

5 实验验证

为验证基于代理的移动云服务访问框架的有效性，本文分别构建 3 个涵盖其不同功能的应用案例，并基于这些案例对框架的应用效果进行了评测。

5.1 应用案例

如图 5 所示，应用案例(1)描述的是移动设备通过代理服务器对 Amazon 开放的 ItemSearch API 进行访问，服务返回结果的数据格式为 XML；应用案例(2)描述的是移动设备通过代理服务器对 eBay 开放的 FindPopularItems API 进行访问，服务返回结果的数据格式为 JSON；应用案例(3)描述的是移动设备通过代理服务器对 Amazon、eBay 服务 Mashup 进行访问。

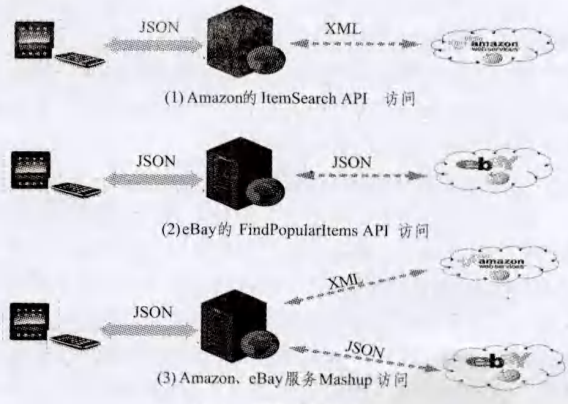

图 5 应用案例示意

应用案例(1)显示了代理服务器具有访问 XML 数据格式的云服务的能力;应用案例(2)显示了代理服务器具有访问 JSON 数据格式的云服务的能力;应用案例(3)则显示了代理服务器具有云服务 Mashup 的能力。上述 3 个应用案例的实现在移动设备上展示的效果如图 6 所示,从左至右分别对应应用案例(1)—(3)。

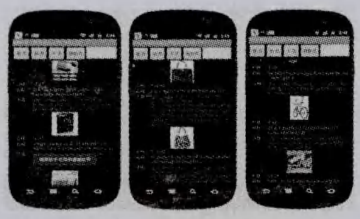

图 6 应用案例在移动设备上的展示效果

5.2 实验评测

基于上节的应用案例,我们设计了一组实验来对本文实现的框架的效果进行评测。本实验采用一台 DELL PC 机(Intel Core2Duo CPU 、2G 内存)作为代理服务器和一台三星 Nexus S 智能手机作为移动设备。在网络接入方面,代理服务器将通过 1Mb/s 带宽的固定网络接入 Internet,移动设备则是通过 1Mb/s 带宽的 Wifi 进行接入。实验主要关注移动设备访问云服务所带来的服务请求响应时间、响应结果处理时间、请求响应流量、能耗这 4 项指标,将对数据传输格式优化和基于代理的服务 Mashup 机制所带来的性能改进分别进行评估。其中能耗指标是通过 PowerTutor[11] 软件获得的。

(1)测试框架中数据传输格式优化所带来的性能改进。

针对 Amazon 开放的 XML 数据格式响应结果的 ItemSearch API,我们分别构建了两个应用程序对其进行访问。如图 7 所示,①描述的是移动设备通过 1Mb/s 带宽的 Wifi 接入 Amazon 的 ItemSearch 服务,服务的响应结果为原始的 XML 数据格式;②描述的则是移动设备通过 1Mb/s 带宽的 Wifi 接入代理服务器,然后由代理服务器来代理请求 Amazon 的 ItemSearch 服务,服务的响应结果为经代理服务器优化后的 JSON 数据格式。

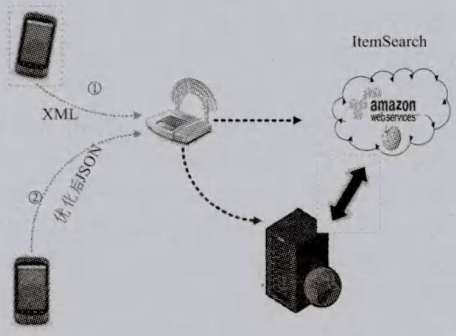

图 7 数据传输格式优化对比测试环境

测试获得的①和②的每次服务请求响应时间、响应结果处理时间、请求响应流量和能耗情况见表 1。

从实验结果数据得知,移动设备通过框架对 XML 格式的云服务进行优化访问要比直接访问在请求响应流量上节省 92%、响应结果处理时间上快 9.89s、移动设备能耗上节省 18%;但同时也造成了服务请求响应时间 4.85s 的延迟。

表 1 框架对 XML 格式的服务访问优化对比测试结果

服务请求响应时间(ms)					
第一组	719	729	705	716	726
第二组	5572	5579	5560	5582	5567
响应结果处理时间(ms)					
第一组	10067	10035	10072	10069	10092
第二组	176	165	188	173	178
请求响应流量(Byte)					
第一组	46147	46145	46147	46147	46147
第二组	3724	3724	3726	3724	3722
移动设备能耗(J)					
第一组	15.84	15.89	15.73	15.84	15.90
第二组	12.71	12.68	12.75	12.80	12.61

(2)测试基于代理的服务 Mashup 所带来的性能改进。

针对用户访问 Amazon、eBay 服务 Mashup 的需求,分别实现了两个移动应用程序。如图 8 所示,①描述的是移动设备通过 1Mb/s 带宽的 Wifi 接入 Amazon 和 eBay 服务,然后在移动设备完成服务 Mashup;②描述的则是移动设备通过 1Mb/s 带宽的 Wifi 接入代理服务器,然后由代理服务器来代理请求 Amazon 和 eBay 服务,并且完成服务 Mashup,将融合后的数据结果返回给移动设备。

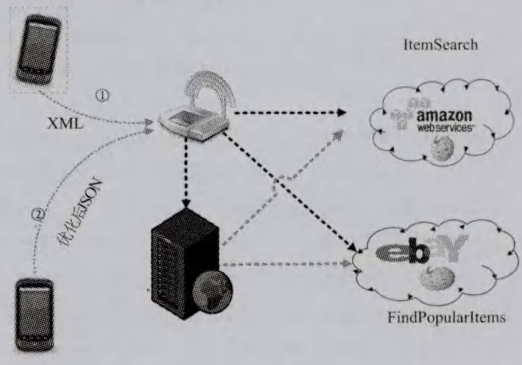

图 8 基于代理的服务 Mashup 对比测试环境

测试获得的①和②的每次服务请求响应时间、响应结果处理时间、请求响应流量和能耗情况见表 2。

表 2 框架在代理服务器上 Mashup 与移动设备上 Mashup 对比测试结果

服务请求响应时间(ms)					
第一组	1534	1521	1544	1530	1541
第二组	7390	7374	7402	7395	7389
响应结果处理时间(ms)					
第一组	10913	10897	10919	10921	10917
第二组	270	284	246	277	273
请求响应流量(Byte)					
第一组	61821	61823	61821	61821	61823
第二组	8700	8700	8696	8700	8700
移动设备能耗(J)					
第一组	17.74	17.71	17.68	17.85	17.72
第二组	16.52	16.78	16.43	16.37	16.49

从实验结果数据得知,移动设备通过框架在代理服务器上做 Amazon 和 eBay 服务 Mashup 要比直接在移动设备上完成 Mashup 效果好,在请求响应流量上节省 97%、响应结果处理时间快 11.39s、移动设备能耗上节省 6%;但同时也造成了服务请求响应时间 5.69s 的延迟。

结束语 随着以手机为代表的移动设备的日益普及和移动网络基础设施的持续改善,移动互联网和移动应用正蓬勃

(下转第 81 页)

结束语 通过对已有算法及其存在问题的分析，结合非均匀分簇的思想，提出了一种能量均衡的非均匀分簇路由算法——EBUCA。改进算法利用节点密度和剩余能量设置簇头的选择阈值，在节点密集区增加簇头数量并在簇的建立阶段利用簇头密度及距 sink 节点的距离建立不同的簇半径，达到平衡全网簇规模的目的。最后通过仿真实验验证算法的优越性，即通过与 LEACH、EEUC 和 DBCP 路由算法进行对比，证明 EBUCA 路由算法能有效平衡各节点能耗，延长网络生命周期。

参 考 文 献

[1] Heinzelman W, Chandrakasan A, Balakrishnan H. Energy-Efficient Communication Protocol for Wireless Microsensor Networks[C]// Proceedings of the 33rd Hawaii International Conference on System Science. Hawaii, 2000: 3005-3014

[2] Qing Li, Zhu Qing-xin, Wang Ming-wen. Design of adistributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks[C]// Computer Communications. 2006: 2230-2237

[3] Handy M J, Haase M, Timmermann D. Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection [C]// Proc of the 4th IEEE Cone on Mobile and Wireless Communications Networks. IEEE Communications Society, 2002: 368—372

[4] 徐久强，毕伟伟，朱剑. WSN 中多跳均匀分簇路由算法的设计与仿真[J]. 系统仿真学报，2011，23(5)：992-997

[5] 顾跃跃，白光伟，陶金晶. LEACH-CS：一种自定义的 WSN 跨区多跳路由机制[J]. 计算机科学，2011，38(1)：78-82

[6] Li Cheng-fa, Ye Mao, Chen Gui-hai. An energy-efficient unequal clustering mechanism for wireless sensor networks[C]// Proc of the 2nd IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS). 2005

[7] 蒋畅江，石为人，王平. 能量均衡的无线传感器网非均匀分簇路由协议[J]. 软件学报，2012，23(5)：1222-123

[8] Du Jiang, Wang Liang. Uneven Clustering Routing Algorithm for Wireless Sensor Networks Based on Ant Colony Optimization[C]// International Conference on Computer Research and Development (ICCRD). 2011: 67-71

[9] 乔俊峰，刘三阳，曹祥宇. 无线传感器网络中基于节点密度的簇算法[J]. 计算机科学，2009，36(12)：46-48

[10] Sungryoul L, Han C, Byoungchang P. LUCA: An Energy-efficient Unequal Clustering Algorithm Using Location Information for Wireless Sensor Networks[J]. Wireless Personal Communications, 2011, 56(4): 715-731

[11] Thein M C M, Thein T. An Energy Efficient Cluster-Head Selection for Wireless Sensor Networks[C]// International Conference on Intelligent Systems Modelling and Simulation (ISMS). 2010: 287-291

[12] Rappaport T. Wireless Communications: Principles & Practice [M]. Englewood Cliffs, NJ: Prentice-Hall, 1996

(上接第 61 页)

发展。然而，移动设备自身计算能力、存储能力、电池续航能力等不足；加之无线网络传输带宽有限且资费昂贵；与此同时，现有的大多数云服务并不关心移动客户端，这些都严重影响了移动云服务的市场发展。因此，本文研究了移动计算环境的特点，设计并实现了一种基于代理的移动云服务访问框架，即通过代理服务器对云服务做传输协议、数据格式等方面的桥接转换，使其在无线网络带宽占用和移动设备资源消耗上变得更适宜移动设备访问；代理服务器上的服务 Mashup 既为移动设备访问个性化服务提供了方便，又节省了移动设备计算、带宽等资源。在未来的研究中，要考虑到移动设备应支持更多的操作系统(如 iOS、Windows Mobile 等操作系统)。而代理服务器上的框架应考虑到随着移动用户的增加，其可能会成为瓶颈，应将这部分的框架移植到计算能力更强的云基础设施上(如 Amazon、Google 等)。

参 考 文 献

[1] Kyung M. Mobile Cloud Computing Challenges[OL]. http://www2. alcatel-lucent. com/blogs/techzine/2010/mobile- cloud-computing-challenges, 2010

[2] Chen Y, et al. EXPedite: A System for Encoded XML Processing [C]// Proc of the ACM 13th Confon Information and Knowledge Management. 2004

[3] Ioana G, Oriana R, Dejan J, et al. Calling the cloud: Enabling mobile phones as interfaces to cloud applications. Lecture Notes of the Institute for Computer Sciences[J]. Social Informatics and Telecommunications Engineering, 2010, 48(4): 161-174

[4] Jan S, Oriana R, Gustavo A. AlfredO: an architecture for flexible interaction with electronic devices[C]// Middleware'08 Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware. 2008

[5] Jan S, Gustavo A, Timothy R. R-OSGi: Distributed applications through software modularization [C] // MIDDLEWARE2007 Proceedings of the 8th ACM/IFIP/USENIX international conference on Middleware. 2007

[6] Xin wen Z, Sangoh J, Anugeetha K, et al. Towards an Elastic Application Model for Augmenting Computing Capabilities of Mobile Platforms[C]// Middleware'09 Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware. 2009

[7] Jason H. Using RESTful Web-Services and Cloud Computing to Create Next Generation Mobile Applications[C]// OOPSLA. 2009

[8] Mahadev S, Paramvir B, Ramón C, et al. The Case for VM-Based Cloudlets in mobile computing[J]. IEEE Pervasive Computing, 2009, 8(4): 14-23

[9] Qian W. Mobile Cloud Computing [D]. Master of Science, The Department of Computer Science in University of Saskatchewan, Saskatoon, 2011

[10] Roy T. Architectural Styles and the Design of Network-base Software Architectures [D]. Doctor of Philosophy, Dept. of Computer Science, Univ. of California, Irvine, 2000

[11] PowerTutor [OL]. http://ziyang. eecs. umich. edu/projects/powertutor/, 2011