

能力共享架构及其在 IM 和 SNS 中的应用研究

崔金红¹ 王旭²

(对外经济贸易大学信息学院 北京 100029)¹ (贝尔实验室基础科学研究院 北京 100080)²

摘要 目前的 IM(即时通信)和 SNS(社交网络服务)系统主要为用户及其好友或组成成员间提供即时信息发送、语音或视频通话、多媒体文件共享等功能,不能在好友或组成成员间共享应用程序、处理器、内存、磁盘或其它设备资源以及设备的通讯能力等等,而这些功能常常正是用户所迫切需要的。介绍了一种新的、潜在的 Web3.0 或 Web 2.0 + 架构,该架构可用于 IM 和 SNS 中共享好友或组成成员的计算机上或智能设备的各种能力,为每个用户的计算机或智能设备提供了一种简单可行的、可扩展的能力发现和共享机制。该机制可极大地提高用户体验,并可成倍地提升现有资源的利用率。

关键词 Web2.0, Web3.0, 即时信息, 社交网络服务, 能力共享

Capability Sharing Architecture and Implementation in IM or SNS

CUI Jin-hong¹ WANG Xu²

(School of Information Technology & Management Engineering, University of International Business and Economics, Beijing 100029, China)¹
(Bell Labs Research China, Beijing 100080, China)²

Abstract Current IM or SNS system provides a way for users to send message, talk, and share pictures and so on among buddies. However, it's not easy to share an application or other resources among the buddies. A user cannot use the resources in other devices which are lacked locally in run time, but sometimes that is what users want. This paper described a new and a potential Web3.0 or Web 2.0 plus way to share the capability of different computers or devices among buddies in IM or SNS system. It provides an easy way for each buddy device to advertise their capability, automatically discover other buddy devices' capability and share the capabilities.

Keywords Web2.0, Web3.0, IM, SNS, Capability sharing

1 引言

根据维基百科的定义,“Web 1.0 是只读的,是以静态数据与简单的标记语言为基础的。Web 2.0 是可读写的动态数据,通过网络服务定制网站和管理条目。而 Web 3.0 则是可读写并可执行的。”在 Web2.0 中,用户不仅从互联网获取信息,而且同时也提供信息,通过互联网与他人实现信息共享及互动,目前有许多 Web 2.0 互动式应用程序,如博客、移动博客、播客、标签、RSS /Atom、Wiki、P2P、SNS、Adsense、Mash-up 等等。与 Web2.0 相比较而言,Web 3.0 到现在还没有一个很清晰的定义。Web 3.0 用来形容未来的万维网,而对下一阶段的万维网的发展分歧较大:有人认为新兴技术如语义网的出现将改变现有万维网的使用模式,使万维网具有人工智能的特性;也有人认为上网速度的提高、模块化的 Web 应用以及计算机图形学的发展,都将在未来的万维网中发挥关键作用;而另一种看法则是把 Web 3.0 看作一个“可执行”的网络抽象层。

目前大受欢迎的 IM 和 SNS 系统如 MSN, Skype 及 Facebook 等是典型的 Web2.0 应用。在好友或组成成员间共享信息、发送消息或谈话是其主要特征。但目前在即时通信和社区网络服务系统中还没有合适的方式或方法能够用来在好友或组成成员间共享计算机或其他智能设备的 CPU、内存、磁盘、

应用软件及其他资源或能力。用户不能够合理利用好友或其他组成成员现有的资源或能力,而某些资源和能力往往又是用户自有的计算机或智能设备所欠缺的。

传统技术如 RPC 和 CORBA 为用户提供了访问远程资源的途径,但是用户必须满足它们的编程规范,而且要访问的资源也必须是已知的。

Mobile Agent 则是将某些计算组件转移到远程站点,再加上它们的状态以及运行所需要的数据等。这样可以带来更大的灵活性,然而在不可控状态下转移代码及部署组件可能造成严重的安全问题。

网格计算系统则提出了异构系统的分布式计算和资源共享之间的架构。因为它一般需要将各节点的资源组织并统一管理,从而形成一台虚拟的超级计算机,所以一般情况下,该系统的底层架构都比较复杂,而且用户都应该遵循其一定的编程准则。

本文将提出一种潜在的 Web3.0 的能力共享架构,及其在即时通信和社区服务网络的应用实例。每个用户都可以通过因特网和大家分享其计算机或智能设备的资源或能力,整个架构是一个抽象的任务执行层或环境,用户可利用好友或组成成员的资源或能力来完成或协同完成他们的任务,并取得预想的效果。

崔金红 博士,副教授,主要从事信息系统、工作流等研究;王旭 博士,高级研究员,主要从事计算机图形学、科学可视化、计算机通信及应用、IMS/NGN 及 3G 应用等研究。

2 共享机制

用户可将其需要完成的任务打包并发送给好友,如果某些好友的计算机或智能设备有合适的资源或能力能够帮助用户完成该任务,这些好友将答复用户的请求并附带自己的设备能力描述。然后,用户端可以自动或手动选择一个或多个合适的好友设备来帮助完成其任务,并得到任务完成的结果。

通过这种机制,用户间可以充分共享其计算机或智能设备上的资源或能力。采用 Web 服务接口发布任务,控制任务的执行并获得任务的执行结果。

相对于传统的 RPC, CORBA, Mobile Agent 和网络计算等系统,该机制具有实施简单、扩展容易的优点;更重要的是,它为现有的即时通信和 SNS 系统增加了共享应用程序、设备资源与能力的特性,并且所有的资源或能力的共享完全由用户自己控制;为专业用户创造更强大的应用或服务提供了一个更灵活的架构和平台。

3 系统设计

3.1 系统架构

跟传统的即时通信系统一样,该系统可以是基于服务器或基于 P2P 的。但不同于传统的 IM 系统的是,系统每个节点有一个描述其能力的清单,如:CPU、内存、操作系统、通信能力(例如短信、彩信)以及安装的软件及主要功能等等,这些软件的功能清单还可以发布在一个公开的网站上,每个节点只保存其所安装软件的功能描述缓存。

各节点间采用 XML 格式来进行信息交互,这样做的好处是容易理解和扩展。

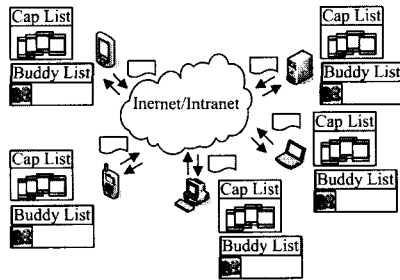


图 1 能力共享网络系统架构

3.2 功能描述

3.1.1 应用软件共享

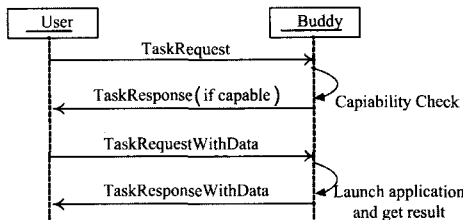


图 2 应用软件共享

如图 2 所示,应用软件共享流程如下:

(1)当用户需要使用一些软件或功能,但该软件或功能并没有安装在他的计算机、移动电话或其它智能设备上,则他可以打包其要求到某个任务中,并公布给所有好友或组成员。例如任务描述如下:

```
<task id>10000001</task id>
```

```
<task type> File Conversion </task type>
```

```
<input> DOC</input>
```

```
<output>PDF</output>
```

(2)收到任务请求的好友或组成员,如果有能力完成用户的请求,则给予回应并附带自己相关的能力描述。例如回应如下:

```
<task id>10000001</task id>
```

```
<cpu> 2.0G AMD Atholon </cpu>
```

```
<meomery>1.0G </meomery>
```

```
<application> PDFWriter</application>
```

```
<version>4.0 </version>
```

(3)该任务的发起节点可自动或手动选择一个或多个合适的节点来帮助其完成任务,并将任务描述与输入的数据派发给这些选中的节点,示例数据如下:

```
<task id>10000001</task id>
```

```
<task type> File Conversion </task type>
```

```
<input> DOC</input>
```

```
<input data> ...</input data>
```

```
<output>PDF</output>
```

(4)任务完成后,其结果将采用如下格式返回到发起端:

```
<task id>10000001</task id>
```

```
<task type> File Conversion </task type>
```

```
<input> DOC</input>
```

```
<output>PDF</output>
```

```
<output data> ...</output data>
```

在任务描述中,任务类型可以轻易扩展以支持 MATLAB 等计算任务、三维图形软件的灰土任务以及数据分析软件等等。对于那些不能自动载入的应用程序,目标节点可以收到通知,并可由用户手动启动应用程序,加载用户的输入数据,并使用系统提供的打包工具将执行结果返回给发起节点。任务执行的结果也可根据发起节点的要求通过电子邮件或其他指定的方式返回。

3.1.2 CPU、内存等资源共

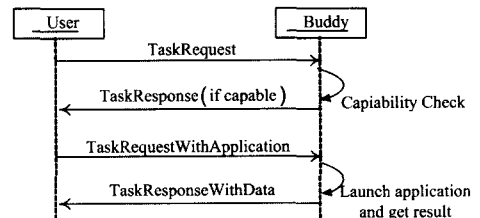


图 3 资源共享

如果用户的计算机或智能设备功能不是足够强大,他可以选择在好友或组成员的节点上执行部分或全部程序,如图 3 所示。

(1)用户编译自己的应用为可执行代码如 .exe 或 .bin 等,也可将自己应用程序的某些模块提取出来编译为 .exe, .bin, .dll 或 .class 等可执行代码。

(2)用户查询好友或组成员寻找合适的执行环境。任务描述开采用如下的格式:

```
<task id>10000002</task id>
```

```
<task type> External Application </task type>
```

```
<type> dll</type>
```

(3)收到任务请求的好友或组成员,如果有能力完成用户的请求,则给予回应并附带自己相关应用运行环境的描述。例如回应如下:

```

<task id>10000002</task id>
<cpu> 2.0G AMD Atholon </cpu>
<memory>1.0G </memory>
<os>Windows XP</os>

```

(4)该任务的发起节点可自动或手动选择一个或多个合适的节点来帮助其完成任务,并将任务描述、执行代码及入口地址等派发给这些选中的节点,示例数据如下:

```

<task id>10000002</task id>
<task type> External Application </task type>
<type>dll</type>
<app> ...</app>
<start entry>main</start entry>

```

(5)任务完成后,其结果将采用如下格式返回到发起端:

```

<task id>10000002</task id>
<task type> External Application </task type>
<type>dll</type>
<result> ...</result>

```

分布在各节点中的应用程序部件间也可以直接通过 Socket 或用其他的方式进行交互,该过程完全由用户编程决定。

3.1.3 通讯能力共享

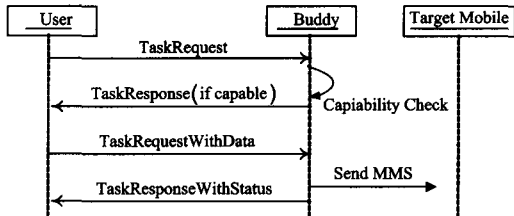


图 4 通讯能力共享

当用户需要发送彩信到某个手机,但该手机没有彩信收发功能,或手机不在身边,则其可以选择使用好友或组员的手机代为发送,步骤如图 4 所示:

(1)打包发送请求到某个任务中,并公布给所有好友或组成员。例如任务描述如下:

```

<task id>10000003</task id>
<task type> Message Send </task type>
<type> MMS</type>

```

(2)收到彩信发送请求的好友或组成员,如果有能力完成用户的请求,则给予回应并附带自己相关应用运行环境的描述。例如回应如下:

```

<task id>10000003</task id>
<mobile type> Nokia N73 </mobile type>
<os>symbian </os>
<version>3.0 </version>

```

(3)该任务的发起节点可自动或手动选择一个或多个合适的节点来帮助其发送彩信,并将任务描述、彩信内容等派发给这些选中的节点,示例数据如下:

```

<task id>10000003</task id>
<task type> Message Send </task type>
<title>Hello from Alex</title>
<input data> ...</input data>

```

(4)任务完成后,其结果将采用如下格式返回到发起端:

```

<task id>10000003</task id>
<task type> Message Send </task type>
<title> Hello from Alex</title>
<status>Successful</status>

```

在任务描绘中,任务类型可以非常容易地扩展以支持电子邮件及短消息发送等等。

3.1.4 相同用户不同设备间的能力共享

本架构同样可以非常容易地扩展以支持相同用户不同计算机或智能设备间的能力共享,以使得其所有的计算机或智能设备形成一个联合的任务执行环境。

3.2 能力共享 IM/SNS 软件架构

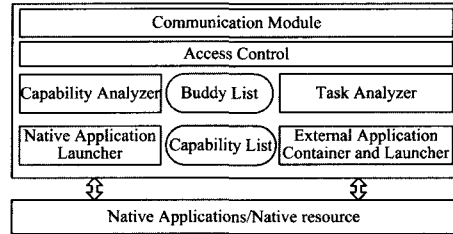


图 5 能力共享软件架构

图 5 是具有能力共享功能的 IM 或 SNS 系统的软件架构图。该系统包括 8 个主要功能模块,其各自功能描述如下:

通讯模块:传统的即时通信、社区通讯功能,以及好友或组成员间的任务和运行结果的传送。

访问控制:本地应用程序和资源的访问控制模块,支持能力访问策略控制政策功能,比如可以控制好友或组成员在什么时候或以什么频率使用用户的那些资源。

任务分析:通过能力分析仪模块,分析本地应用程序或资源是否能够支持好友或组成员的任务请求;将用户请求打包为任务并通过通讯模块发送给好友或组成员,获得任务执行的结果并返回给用户。

能力分析:结合本地系统信息及能力清单,以分析本地环境能否支持某个任务请求。

本地应用调用:本地应用调用,应用数据传送。

外部应用存储及调用:存储和调用从好友或组成员传送而来的外部应用,应用数据传送。它可以在接受完完整的外部应用后自动调用它,也可由主控用户端通过 XML 格式的指令通知调用,以同步各个分布节点上部分应用的运行。

好友列表:用户的好友列表或组成员列表,用户利用这个名单,发出任务请求、检查请求以及能力访问控制。

能力清单:列举本地系统、软件等的功能及调用模式。

结束语 本文提供了一种在不同的用户或智能设备中共享应用程序和资源如 CPU、内存以及通讯能力等的架构,它将极大地提高用户计算机或智能设备的资源利用率。

通过该架构的使用,本文首次尝试用一种全新的方式在 IM 或 SNS 的好友或组成员间实时共享设备的能力,而不仅仅是完成简单的、传统的通讯或信息共享功能。

相对于传统的 RPC, CORBA 技术、移动 Agent 以及网格运算系统等,本文提出的架构和方法具有结构简单、使用方便以及易于扩展等特点,而且本地的资源访问完全由用户自己通过资源访问策略来决定。在这个架构的基础上,专业用户能够创造功能更强大的应用程序或服务,而不仅限于由传统的方法或体系结构。

参考文献

[1] RFC 2778. A Model for Presence and Instant Messaging. <http://www.ietf.org/rfc/rfc2778.txt>, 2000

由(3)式可知,I/O请求丢失率受数传率THR、实施策略要用到的OBSD个数 k 和系统开销 $\Delta t(i)$ 等因素的影响。由 $\lambda kt + \frac{\lambda T}{k} \geq 2\lambda \sqrt{Ti}$ 知,当 $\lambda kt + \frac{\lambda T}{k}$ 取最小值时, η_{loss} 最小。与(3)式相对应的走势图如图5所示。其中横轴表示实施对象文件均分策略所用到的OBSD个数 k ,纵轴表示I/O请求丢失率 η_{loss} 。显然,越是热点对象文件,其服务请求到达率就越高,其相应的I/O请求丢失率也就越高。此外,由图5可知,图中的每条曲线都存在一个很明显的拐点, η_{loss} 在拐点处取得最小值。由此可见,在实施均匀分块策略的过程中,并不是将热点对象文件分块越多,I/O请求丢失率的改善效果越好,而是存在一个最优的分块数 $k_{opt} = \sqrt{\frac{T}{t}}$,使得系统开销的增加对系统性能改善影响最小,从而使得 η_{loss} 降到最低。另外,(3)式还反映了 η_{loss} 与THR的紧密关系,在对象文件大小OFS一定,且其他因素不变的情况下,THR越大,其相应的 η_{loss} 越小。这说明 η_{loss} 的变化与THR息息相关。

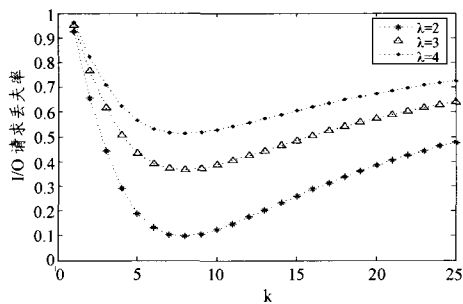


图5 各个相关指标关系图

结束语 本文将一种新的网络存储架构——对象存储应用于多媒体服务系统,提出并构建了MOSS,这将是发展下一代多媒体服务系统,尤其是建设超大规模多媒体应用的一个方向。此外,本文还讨论了热点对象文件的存储优化策略,通过建立与系统相对应的数学模型,对数传率、I/O请求丢失率等性能指标作了定性分析。该策略对消除系统I/O瓶颈、设计分布式多媒体服务系统具有实际的指导意义。

参考文献

[1] Bitton D, Gray J. Disk shadowing // The 34th IEEE COMPCON. San Francisco, CA, 1998

[2] Ghandeharizadeh S, et al. On disk scheduling and data placement for video servers. Tech Rep, USC-CS-97-650. USC, USA, 1997

[3] Shenoy P J, Vin H M. Efficient striping techniques for variable bit rate continuous media file servers. Performance Evaluation, 1999, 38(3/4): 175-199

[4] Schindler J, et al. Track-aligned extents: Matching access patterns to disk drive characteristics // Conf. File and Storage Technologies (FAST). Monterey, CA, USA, 2002

[5] Yu Xiang, Gum Benjamin, et al. Trading capacity for performance in a disk array // Symposium in a disk array. Symposium on Operating Systems Design and Implementation. San Diego, CA, USA, 2000

[6] On G, Zink M, et al. Replication for a distributed multimedia system // The 8th Int'l Conf. Parallel and Distributed Systems. Kyongju City, Korea, 2001

[7] Wu Song, Jin Hai. Symmetrical pair scheme: A load balancing strategy to solve intra-movie skewness for parallel video servers // The Int'l Parallel and Distributed Processing Symposium. Marriott Marina, Fort Lauderdale, Florida, 2002

[8] 李宇, 张江陵, 冯丹. 一种面向视频播放系统的RAID并行预取技术及实现. 计算机研究与发展, 2002, 39(11): 1526-1530

[9] Mesnier M, et al. Object-Based Storage. IEEE Communications Magazine, August 2003

[10] 李钢江, 杨士强. 分布式多媒体点播系统的结构设计. 小型微型计算机系统, 2001, 22(3): 257-260

[11] 孟玉柯. 排队论基础及应用. 上海: 同济大学出版社, 1989

[12] Factor M, et al. Object storage: the future building block for storage systems. IBM Haifa Research Laboratories, 2005

[13] Zeng Ling-fang, Feng Dan, et al. Object replication and migration policy based OSS // Proceedings of the Fourth International Conference on Machine Learning and Cybernetics. Guangzhou, 2005

[14] Zeng Ling-fang, Feng Dan, et al. A strategy of load balancing in object storage system // Proceedings of the Fifth International Conference on Computer and Information Technology (CIT'05). 2005

[15] 覃灵军, 冯丹, 曾令仿, 等. 基于对象存储系统的动态负载均衡算法. 计算机科学, 2006, 33(5): 88-91

[16] Little T, Venkatesh D. Popularity-based assignment of movies to storage devices in a video-on-demand system. Multimedia Systems, 1995, 2(8): 280-287

[17] 靳超, 郑纬民, 张悠慧. 主动存储系统结构. 计算机学报, 2005, 28(6): 1013-1020

[18] 陈鑫林. 现代通信中的排队论. 电子工业出版社, 1999

(上接第75页)

[2] RFC 2779. Instant Messaging / Presence Protocol Requirements. <http://www.ietf.org/rfc/rfc2779.txt>, 2000

[3] RFC 3920. Extensible Messaging and Presence Protocol (XMPP): Core. <http://www.ietf.org/rfc/rfc3920.txt>, 2004

[4] RFC 3921. Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. <http://www.ietf.org/rfc/rfc3921.txt>, 2004

[5] RFC 3922. Mapping the Extensible Messaging and Presence Protocol (XMPP) to Common Presence and Instant Messaging. <http://www.ietf.org/rfc/rfc3922.txt>, 2004

[6] RFC 3923. End-to-End Object Encryption in the Extensible Messaging and Presence Protocol (XMPP). <http://www.ietf.org/rfc/rfc3923.txt>, 2004

[7] SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE). <http://www.ietf.org/html.charters/simple-charter.html>, 2007

[8] Zhang Yun-chuan. Standardized Instant Messaging Protocols: Comparative Analysis of SIMPLE and XMPP. Journal of Wuhan University of Science and Technology (Natural Science Edition), 2005, 4

[9] Petrack S. SIMPLE Aims for IM Interoperability. Network World, 2004, 21(3): 39

[10] Hildebrand J. XMPP Transports Presence Data. Network World, 2004, 21(10): 31

[11] van der Vlist E, Ayers D, Bruchez E. Professional Web 2.0 Programming, Wrox Press, 2007

[12] Gómez-Pérez A. The Semantic Web: Research and Applications. Springer Press, 2004

[13] OMG's CORBA. <http://www.corba.org/>

[14] Mobile Agent. http://en.wikipedia.org/wiki/Mobile_agent

[15] Tiecke S. Open grid services infrastructure (OGSI) Version 1.0. http://www.globus.org/toolkit/draft-ggf-ogsi-gridservice-33_2003-06-27.pdf