云平台应用系统迁移方法的研究

张 靓 范冰冰 郑伟平

(华南师范大学计算机学院 广州 510630)

摘 要 针对多组件应用系统迁移到云平台的目标物理服务器的选择问题,提出了一种基于组件间通信量的迁移方法。迁移方法根据形式化后的应用系统逻辑架构图和云平台物理资源组织结构图,通过计算组件间的通信量和虚拟资源需求值,为应用系统各组件在云平台选择目标物理服务器。通过在品高云计算平台上的多租户精品课程网站群系统的迁移实验,证明了所提迁移方法较好地减少了应用系统在使用时,云平台内部物理服务器间的通信量。

关键词 云计算,应用系统,迁移,组件,通信

中图法分类号 TP393 **文献标识码** A

Study of Methods for the Cloud to Migrate Application Systems

ZHANG Liang FAN Bing-bing ZHENG Wei-ping

(School of Computer, South China Normal University, Guangzhou 510630, China)

Abstract Regarding the problem of selecting target physical servers when migrating multi-component application systems to the cloud, a migration method based on the inter-component communication amount was proposed. According to the formalization of the application system logical architecture and cloud platform physical resources organizational structure, the method is to select target physical server for each system component by calculating the communication amount and virtual resource requirements of them. By the experiment of migrating multi-tenancy excellent course websites system to Bingo Cloud, the result shows that the migration method can reduce the traffic between physical servers of the cloud platform.

Keywords Cloud computing, Application system, Migration, Component, Communication

1 引言

随着基于虚拟化技术的云计算平台的日益成熟,将应用系统迁移到云计算平台上,可以有效地利用云平台实现提升资源利用率、动态调度资源和统一运营管理[1]。然而对于多组件的应用系统,组件定义为迁移在一台虚拟机里具有特定功能的系统模块,在迁移的过程中,仍需要解决的一个关键问题是为待迁移的应用系统的各组件合理地选择云平台上的目标物理服务器。

现有的云计算平台已有资源调度策略如负载均衡策略、低能耗策略等来确定虚拟机在云平台的放置位置,然而其中的物理服务器选择方法、成熟的云平台运营商、如 Amazon[2] 是不公开的,而开源的云平台产品,如 Eucalyptus[3] 则留有接口给用户编写。在将应用系统迁移到云计算平台的实例[4-6]中,都给出了具体的迁移过程,但迁移过程没有考虑到应用系统组件间的联系。如果不考虑组件间的联系,会造成在迁移系统时,使各组件(如 Web 工程、数据库等)在特定资源调度的策略下,分别迁移到了云平台中不同网段的物理服务器,造成应用系统在使用时,云平台内部跨网段的大量通信,即云平台内部带宽的浪费。文献[7,8]考虑了虚拟机(组件)在云平台上的通信情况,但文献[7]重在研究应用系统本地组件和云平台上组件的通信,而文献[8]旨在解决云平台运行时虚拟机的动态放置问题。

针对上述问题,本文给出多组件应用系统迁移到云平台的迁移方法,迁移方法满足云平台使用的资源调度策略,使得应用系统各组件在满足虚拟资源需求的条件下,尽量部署在云平台的同一台物理服务器、同一个网段下。

2 形式化方法

为更好地描述迁移方法,将待迁移的应用系统的逻辑架构和云平台的物理资源架构进行形式化定义。

2.1 应用系统逻辑架构形式化定义

将系统组件逻辑架构形式化为图[7]

$G=(V,E)$

其共包括 m 个组件,组件用 C 表示,定义为迁移在一台虚拟机里、具有特定功能的系统模块,则

$V=\{C_i\}_{i=1}^{m} \cup \{I/O\}_{j=1}^{n}$

I/O 代表系统的使用者。如果节点 p 和 q 有通信,则节

本文受广州市科技计划重大项目(7421162366392)资助。

张 靓(1987—),女,硕士生,主要研究方向为云计算,E-mail:564585499@qq.com;范冰冰 男,教授,主要研究方向为云计算、移动互联网;郑伟平 男,高级工程师,主要研究方向为云计算。

点 p 和 q 通过边(p,q)连接。边的权值代表节点间单位时间通信量的大小。每个组件有 4 个属性，即 CPU、MEM、HD 和 BW，其分别代表组件对虚拟资源 CPU、内存、硬盘和带宽的需求值。其中带宽需求值包含两部分，即 $BW=BW_1+BW_2$，BW_1 为与应用系统内部组件通信的需求值，BW_2 为与外部系统使用者通信的需求值，两部分值之和为总的带宽需求值。

如形式化图 1 所展示的应用系统的逻辑架构后的图如图 2 所示，表 1 列出了应用系统各组件的虚拟资源需求值，这里假设云平台物理服务器采用同一配置。

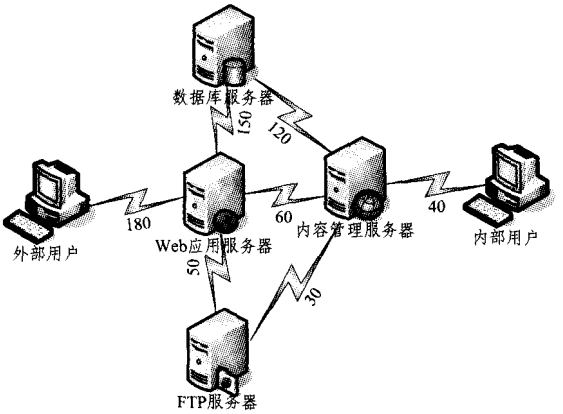

图 1　应用系统逻辑架构

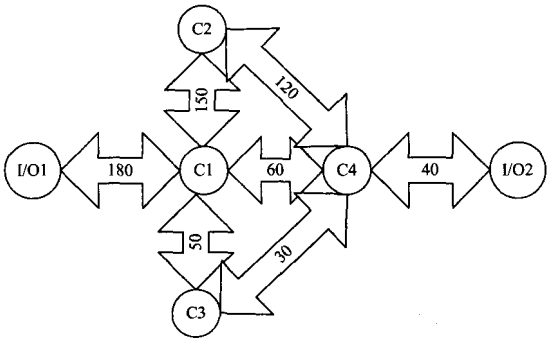

图 2　形式化后的应用系统逻辑架构图 G

表 1　应用系统虚拟资源需求值

	Web 应用服务器	数据库服务器	FTP 服务器	内容管理服务器
组件	C1	C2	C3	C4
CPU(核)	4	2	2	4
MEM(G)	10	8	8	10
HD(GB)	30	50	100	40
BW(kB/s)	440	270	80	250
BW1(kB/s)	260	270	80	210
BW2(kB/s)	180	0	0	40

2.2　云平台物理资源形式化定义

云平台物理服务器的组织结构采用 Eucalyptus[3] 的结构，可形式化为树形图 T。可提供资源的物理服务器为叶子节点。假设共包含有 k 个叶子节点，叶子节点用 NC 表示，每个叶子节点可提供的 CPU、内存、硬盘、网络带宽分别表示为 CPU、MEM、HD 和 BW。根据集群控制节点 CC 的个数 f，将叶子节点分成 f 个集群，此外，云平台还包含云控制节点 CLC，其负责管理整个云平台系统。如形式化图 3 所示的云平台物理节点的组织结构后的树形图如图 4 所示，其共包含 3 个集群和 9 个叶子节点。表 2 列出了云平台上的各物理服务器提供虚拟资源的情况。

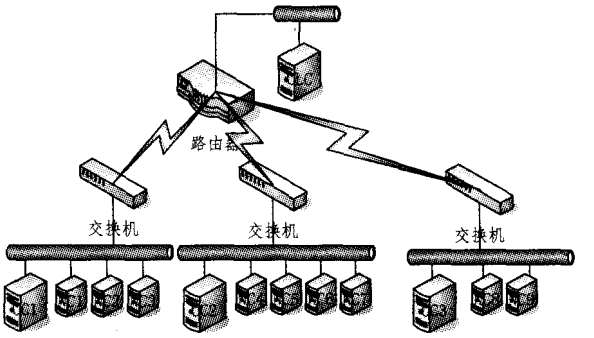

图 3　云平台物理资源组织结构

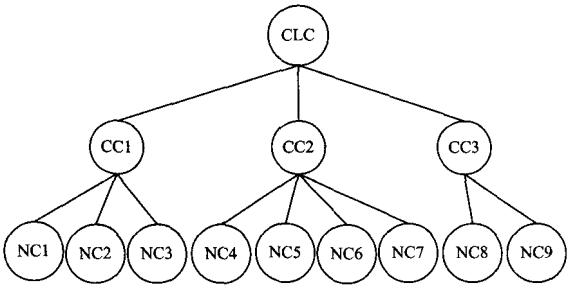

图 4　形式化后的树形图 T

表 2　云平台物理服务器提供虚拟资源的情况

	N_1	N_2	……	N_9
CPU(核)	8	8	……	4
MEM(G)	6	6	……	6
HD(GB)	500	400	……	300
BW(kB/s)	1024	10240	……	880

3　基于组件间通信量的迁移方法

迁移方法根据应用系统形式化后的组件逻辑架构图、各组件的虚拟资源需求和形式化后的云平台物理资源结构图、叶子节点可提供的虚拟资源，决定应用系统的各组件具体迁移到云平台的哪个叶子节点上，以实现在满足组件的虚拟资源需求、云平台的资源调度策略的条件下，使组成系统的各组件尽量部署在同一台物理服务器、同一个集群下。

假设云平台可提供的虚拟资源远大于应用系统需求的虚拟资源，迁移方法具体步骤如下：

(1)集群选择。从云平台资源 f 个集群里，选择一个集群为应用系统的目标迁移集群。

第 1 步　计算应用系统组件逻辑架构图中，所有组件对 CPU、内存、硬盘、带宽的总需求量分别记为 $SysNeedCPU$，$SysNeedMEM$，$SysNeedHD$，$SysNeedBW$。

第 2 步　计算云平台 f 个集群每个集群的叶子节点个数，及集群内 CPU、内存、硬盘、带宽的总剩余量和平均剩余量，即

$$Cloud=\{CC_i\}_{i=1}^{f}$$

$$NodeCC_i=\{NC_j\}_{j=1}^{k}$$

$$SumCpuCC_i=\sum_{j=1}^{k}CpuNC_j$$

$$AvgCpuCC_i=\sum_{j=1}^{k}CpuNC_j/k$$

式中，$Cloud$ 为云平台包含的 f 个集群，$NodeCC_i$ 为第 i 个集群的叶子节点，$SumCpuCC_i$ 为第 i 个集群的 CPU 总数，$AvgCpuCC_i$ 为第 i 个集群的 CPU 平均数，内存、硬盘、带宽

同 CPU 的计算。

第 3 步　选择出集群内各资源总量大于各资源所需总量的 F' 集群，即

$$F'=\{CC_i\in Cloud/SumCpuCC_i\geqslant SysNeedCPU\cap SumMemCC_i\geqslant SysNeedMEM\cap SumHdCC_i\geqslant SysNeedHD\cap SumBwCC_i\geqslant SysNeedBW\}$$

若 $F'=\{\Phi\}$，则选择 f 个集群中 CPU 或内存资源总量最多的集群；

若 $|F'|=1$，则选择集合中的集群；

若 $|F'|>1$，则需从 F' 集群里选择一个最合适的集群，选择的方法需根据云平台当前的资源调度策略，如采用负载均衡策略，可选择 F' 中平均资源总量最多的集群，如采用资源利用率最高的策略，可选择 F' 中平均资源总量最少的集群。选择方法视具体云计算平台及其资源调度策略而定。

(2)集群内叶子节点选择。设已选择的云平台集群为 F_t，决定应用系统迁移的到 F_t 上的物理节点。

第 1 步　初始化通信量序列。按 G 图组件间通信量的大小对组件对排序，设有序集合序列为 S，则

$S=\{\langle C_i,C_j\rangle/C_i,C_j\in V\cap$元素按组件间通信量大小排序$\}$

如按图 2，

$$S=\{\langle C_1,C_2\rangle,\langle C_2,C_4\rangle,\langle C_1,C_4\rangle,\langle C_1,C_3\rangle,\langle C_3,C_4\rangle\}$$

第 2 步　初始化迁移。按 S 顺序，选择通信量最大的两个组件，计算 CPU、MEM、HD、BW 资源需求值的和，和 F_t 集群里各叶子节点可提供的虚拟资源值，计算在 F_t 集群内是否有叶子节点满足，设满足的叶子节点的集合为 N'，则

$$N'=\{N_k\in NodeF_t/CPUN_k\geqslant SumCPUC_iC_j\cap MEMN_k\geqslant SumMEMC_iC_j\cap HDN_k\geqslant SumHDC_iC_j\cap BWN_k\geqslant SumBWC_iC_j\}$$

若 $|N'|=1$，有一个叶子节点满足，则将两个组件分别迁移到这台物理服务器满足虚拟资源的两个虚拟机里，更新图 G。图 G 的更新方法为将两个组件节点合为一个组件节点，合并后的节点与其他节点通信量的权值为合并前两节点与其他节点通信量的和。如图 2，假设 C_1，C_2 组合后的资源在集群 F_t 里有一个物理节点满足，则将 C_1，C_2 分别迁移到这台物理服务器的两个虚拟机里，图 2 更新为图 5，灰色节点 C_{12} 表示为 C_1 和 C_2 已迁移到一台物理服务器上，C_{12} 和 C_4 间的通信量更新为原 C_1、C_2 与 C_4 的通信量的和。

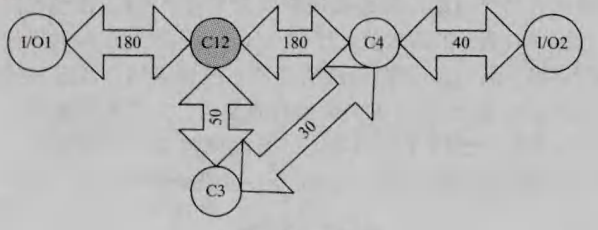

图 5　更新图 G

若 $|N'|>1$，有多个叶子节点满足，则需从 N' 中选择一个最合适的节点，选择的方法需根据云平台 F_t 集群当前的资源调度策略从 N' 中选择。

若 $N'=\{\Phi\}$，没有叶子节点满足，则按 S 的顺序，依次选择组件对直到有叶子节点满足组件对的虚拟资源需求的和，则将两个组件分别迁移到满足资源的物理服务器的两个虚拟机里。若遍历完 S 的组件对，仍没有叶子节点满足，则依次选择 S 里的组件对，直到有两台物理服务器分别满足组件的虚拟资源需求，则组件分别进行迁移。如图 2，假设 F_t 集群里没有叶子节点满足 S 序列里所有组件对的资源需求的和，有分别满足 C_1 和 C_2 资源需求的两台物理服务器，则迁移 C_1 和 C_2，更新图 G，如图 6 所示。

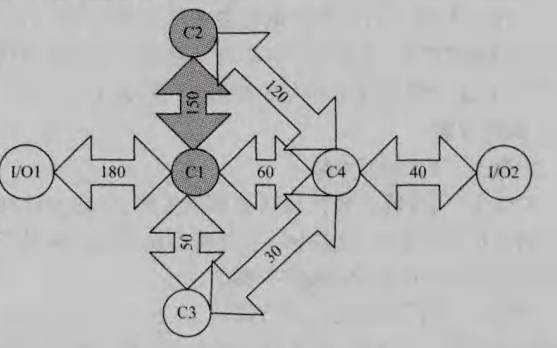

图 6　更新图 G

初始化迁移后的结果是，实现了第一次有组件迁移到云平台上。

第 3 步　集群内遍历序列 S 迁移。根据初始化迁移后的图 G，更新序列 S，按通信量大小重新排列组件对，遍历 S，依次选择组件对进行计算，当计算结果满足迁移的条件时，进行组件迁移。计算的方法按组件对中是否有已迁移的组件分为两种。

第 1 种　组件对中的组件，含有已迁移的。计算已迁移组件所在的物理服务器的剩余资源，是否满足未迁移的组件的虚拟资源需求，若满足，则将该组件迁移至该物理服务器。

第 2 种　组件对中的组件，都为未迁移的。则组件对资源需求和的计算方法和目标物理节点的选择方法同第 2 步中的 N' 的计算方法，且在满足 $|N'|=1$ 或 $|N'|>1$ 的情况下进行组件迁移。

集群内遍历序列 S 迁移后的结果是尽可能将通信量大的组件对部署在同一台物理服务器上。

第 4 步　集群内单独迁移。经上述步骤后，仍有组件未迁移的，将各组件单独迁移在集群内满足资源需求的物理服务器内，若物理服务器的数量大于 1，则根据云平台资源调度策略选择最适合的节点。

集群内单独迁移后的结果是尽可能将组成应用系统的各组件迁移在同一个集群内。

(3)集群外叶子节点的选择。经过上述过程，若应用系统在集群 F_t 里仍有未迁移的组件，则集群 F_t 的剩余资源不能满足未迁移的组件需求，需选择另一集群 F_p，选择方法同 F'，即

$$F'=\{CC_i\in Cloud/SumCpuCC_i\geqslant SysNeedCPU\cap SumMemCC_i\geqslant SysNeedMEM\cap SumHdCC_i\geqslant SysNeedHD\cap SumBwCC_i\geqslant SysNeedBW\}$$

此后，集群内节点选择和集群外节点选择同上述过程(2)、(3)，直至应用系统所有的组件在云平台物理节点上找到相应的迁移位置。

4　实验内容

本文实验使用多租户精品课程网站群应用系统(以下简称多租户应用系统)和品高云计算平台(版本 3.2.4)，采用两种方案分别将本文提出的迁移方法与使用云平台原有的贪婪和节约调度策略作对比，将多租户应用系统的各组件迁移到云平台。对不同的迁移结果分别使用 Load Runner 工具测试

应用系统的请求响应情况，并利用云平台监控系统监控云平台的通信情况。经测试，对于 POST 请求的事务响应时间，本文的迁移方法比云计算平台的贪婪调度策略平均减少 23.5%，比云计算平台的节约调度策略平均减少 11.5%；对于云平台网络流量，其比云计算平台的贪婪调度策略平均减少 17%，比云计算平台的节约调度策略平均减少 20.5%。

4.1 实验方案

方案 1　贪婪调度策略。

实验 1　云计算平台使用贪婪调度策略，即向云计算平台申请满足组件需求的虚拟机时，将虚拟机创建在满足资源需求且剩余资源最多的物理服务器上。

实验 2　使用本文的迁移方法，且在有多个集群或多台物理服务器满足需求时，使用贪婪算法选择最佳集群或物理服务器。

方案 2　节约调度策略。

实验 3　云计算平台使用节约调度策略，即向云计算平台申请满足组件需求的虚拟机时，将虚拟机创建在满足资源需求且剩余资源最少的物理服务器上。

实验 4　使用本文的迁移方法，且在有多个集群或多台物理服务器满足需求时，使用节约算法选择最佳集群或物理服务器。

在 4 种迁移结果下，使用 Load Runner 工具分别模拟并发 20 人及 100 人向多租户应用系统发送 GET 请求及 POST 请求，测试事务响应时间，并监控云平台的通信量。

4.2 结果分析

从实验结果的事务响应时间数据可以看出，方案 1 中实验 1 和实验 2、方案 2 中实验 3 和实验 4 的 GET 请求的事务响应时间相差不多，如方案 1 中并发 20 人时相同，并发 100 人时，实验 2 比实验 1 减少 1 秒(见图 7)。而对于 POST 请求的事务响应时间，实验 2 比实验 1、实验 4 比实验 3 都有减少，分别并发 20 人减少 13%、4%，并发 100 人减少 34%、19%。即本文的迁移方法对于 POST 请求的事务响应时间，比云计算平台的贪婪调度策略平均减少 23.5%，比云计算平台的节约调度策略平均减少 11.5%，且并发人数越多，POST 请求的事务响应时间减少得越多(见图 8、图 10)。

从实验结果的云平台网络流量数据(见图 7—图 10)可以看出，方案 1 中实验 2 比实验 1、方案 2 中实验 4 比实验 3 在相同的请求及并发人数下，网络流量均有减少。其中实验 2 比实验 1，GET 请求并发 20 人减少 18.5%，100 人减少11.4%，POST 请求并发 20 人减少 16.2%，100 人减少 22.2%；实验 4 比实验 3 GET 请求并发 20 人减少 13.9%，100 人减少19.8%，POST 请求并发 20 人减少 15.7%，100 人减少32.6%。即本文的迁移方法对于云平台网络流量，比云计算平台的贪婪调度策略平均减少 17%，比云计算平台的节约调度策略平均减少 20.5%。

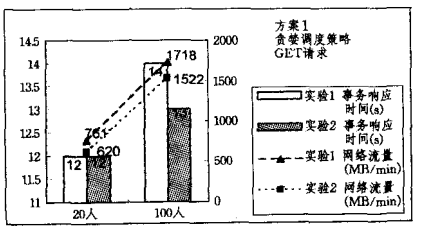

图 7　方案 1　GET 请求测试结果

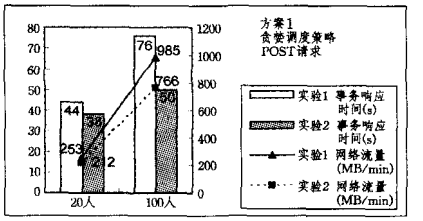

图 8　方案 1　POST 请求测试结果

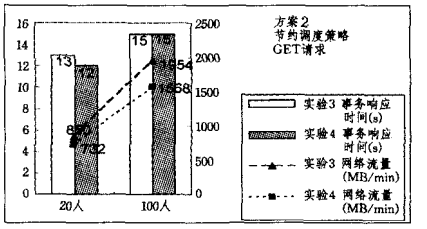

图 9　方案 2　GET 请求测试结果

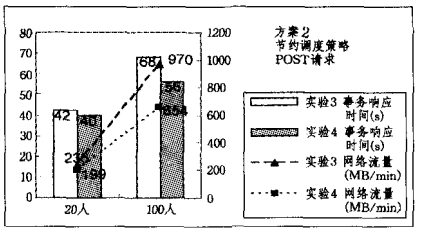

图 10　方案 2　POST 请求测试结果

分析实验结果，因为多租户应用系统设有缓存，所以实验 1 对比实验 2、实验 3 对比实验 4 中 GET 请求的事务响应时间相差不多。而对于 POST 请求，因为实验 2 和实验 4 应用本文的迁移方法，比实验 1 和实验 3 将多租户应用系统的部分组件集中迁移在同一台物理服务器上，组件间通信使用单物理服务器的虚拟网卡，减少了跨物理服务器底层通信的连接过程，在请求人数较多的情况下，事务响应时间减少得较明显。同样，也因此大大减少了云平台实际的网络流量。

结束语　本文提出了一种基于组件间通信量的迁移方法，使得在迁移多组件的应用系统到云平台时，尽量将通信量大的组件迁移到同一台物理服务器、同一个网段下；使组件在通信时，有效地减少了云平台内部物理服务器间的网络流量，同时迁移方法满足于云平台的资源调度策略。本文的迁移方法适用于将已有的应用系统迁移到云平台的过程，迁移时假定各组件对资源的需求值是评估及计算原系统使用时的统计数据所得。然而在迁移后，随着应用系统接入用户数量和请求模式的变化，组件需求的虚拟资源及组件间的通信情况也将发生变化。如何在应用系统运行时调整组件的资源和位置，以提高云计算平台的性能，是下一步要研究的问题。

参 考 文 献

[1] 虚拟化与云计算小组. 云计算宝典技术与实践[M]. 北京：电子工业出版社，2011：165-167

[2] Amazon Web Services. Amazon Elastic Compute Cloud(Amazon EC2) [EB/OL]. http://aws.amazon.com/cn/ec2/，2012-10-10

[3] Eucalyptus Systems. EUCALYPTUS CLOUD [EB/OL]. http://www.eucalyptus.com/eucalyptus-cloud，2012-09-13

[4] 张磊. 微软技术大会(TechED2011)如何将 Web 应用迁移到 Windows Azure 平台[EB/OL]. http://technet.microsoft.com/zh-cn/hh533894，2012-09-20

[5] Walberg S A. 将您的 Linux 应用程序迁移到 Amazon 云[EB/

OL]. http://www.ibm.com/developerworks/cn/linux/l-migrate2cloud-1/,2012-07-20

[6] Khajeh-Hosseini A, Greenwood D, Sommerville I. Cloud Migration: A Case Study of Migrating an Enterprise IT System to IaaS [C]// 2010 IEEE 3rd International Conference on Cloud Computing. Miami, USA: IEEE Computer Society, 2010: 450-457

[7] Hajjat M, Sun Xin, Sung Yu-wei E, et al. Cloudward Bound: Planning for Beneficial Migration of Enterprise Applications to the Cloud [C]// SIGCOMM 2010. Association for Computing Machinery, 2010

[8] 李强，郝沁汾，肖利民，等. 云计算中虚拟机放置的自适应管理与多目标优化[J]. 计算机学报，2011，34(12)

(上接第 238 页)

通过该技术，利用客户端资源来实现数据的计算等复杂操作。XMLHttpRequest 组件可以利用 Javascript 程序来创建其对象及其函数的应用来实现客户端对服务器端数据的访问及计算更新。

客户端处理具体步骤：首先将计算函数嵌入 Javascript 脚本；然后将该函数定义成能实现链接的 onClick 事件处理程序；最后当用户点击界面按钮能链接该事件时，查找相应的计算函数来计算参数，该参数即为服务器传至浏览器的数值。

3.5 客户端缓存

XMLHttpRequest 组件与服务器端的信息交互需要经常与服务器建立连接的原因在于该组件向服务区提交请求的方法为 HTTP 法。而一旦需要交互、更新的数据过多，就需要大量多次地连接服务器，使得本来资源有限的服务器负载增大。为了避免这种缺陷，本文在客户端中增加一种缓存机制。该缓存机制在网页中是不可视的，它的任务在于减少 XMLHttpRequest发送请求时与服务器进行的连接次数。该缓存可以将多个请求整合，由此来进行集体对服务器的连接，使得连接次数减少。服务器端也会将请求的资源内容进行整合封装，从而形成一个应答报文与浏览器端进行交互。这样也能减少两者的连接次数，减少服务器端的负载。而封装的信息由缓存来解析，并由相应的回调函数进行处理。

4 性能测试

在 SX52+RTL8019AS 平台上对该嵌入式瘦 Web 服务器从收到请求建立连接到响应结束断开连接的时间进行了测试。在实验中将本文设计的嵌入式瘦 Web 服务器与使用 Applet 的 Web 服务器进行比较，从而找出其性能优势及不足。其中比较的前提是访问的是同一个网页且完成的是同一个功能。其性能比较如表 1 所列。

表 1 性能比较表

静态页面	数据流量(k)	测试次数	平均响应时间(ms)
本文 Web	0.83	20	2.72
使用 Applet	1.22	20	2.85
动态页面	数据流量(k)	测试次数	平均响应时间(ms)
本文 Web	3.25	20	22.98
使用 Applet	6.43	20	24.46

通过对性能测试表的分析，表明该嵌入式瘦 Web 服务器具有比较快的服务响应速度，可以正常执行设计要求的各项功能。AJAX 技术的使用明显降低了服务器的响应时间，并减少了 Web 流量。AJAX 技术的使用，一方面可以利用客户端闲置的处理能力承担一部分服务器工作，减轻服务器和带宽的负担；另一方面可以降低页面重载的频率，减少带宽消耗，优化性能，可以得到更好的用户体验。

上述结果证明了使用 AJAX+CGI 的思想构造嵌入式瘦 Web 服务器的方法是可行的。该思想可以充分利用客户端资源，明显降低服务器端负载，提高在低端嵌入式设备中嵌入式 Web 服务器的性能。

结束语 本文主要是在低端单片机的环境下实现具有交互功能的嵌入式瘦 Web 服务器。通过此设计，进一步了解了嵌入式系统网络接入和嵌入式瘦 Web 服务器的基本原理、组成、结构和实现过程。该嵌入式瘦 Web 服务器能够满足低端嵌入式设备接入 Internet 并通过 Web 服务器进行监控的要求，且能够基本符合嵌入式系统开发过程中低成本、高可用性的要求。但是在系统中嵌入式瘦 Web 服务器对信息以及数据的传递上，需要加强安全保密工作，以提高数据传输的安全性。

参 考 文 献

[1] Su Ying-qiang, Zhang Wei-qiang. Design of wireless embedded thin web server based on zigbee [C]// 2010 2nd International Conference on Future Computer and Communication (ICFCC). 2010(1): 525-528

[2] Liu Li-xia. Research on technology of embedded web server application [C]// 2010 The 2nd IEEE International Conference on Information Management and Engineering (ICIME). 2010: 187-189

[3] Liu Jian-hong, Chen Jing, Tai Yu-chin, et al. Supporting Audio Streaming in Application Cloud for Embedded Systems [C]// 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS). 2012 IEEE 14th International Conference on High Performance Computing and Communication, 2012: 1800-1805

[4] Liu Jian-hong, Jing Chen, Chuan Tai-yi, et al. ACES-Application Cloud for Embedded Systems [C]// 2011 IEEE/IPSJ 11th International Symposium on Applications and the Internet. 2011: 145-151

[5] Kovatsch M, Mayer S, Ostermaier B. Moving Application Logic from the Firmware to the Cloud: Towards the Thin Server Architecture for the Internet of Things [C]// 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. 2012: 751-756

[6] Limpraptono F Y, Ratna A A P, Sudibyo H. Remote laboratories multiuser based on embedded web server [C]// 2012 9th International Conference on Remote Engineering and Virtual Instrumentation (REV). 2012: 1-7

[7] Limpraptono F Y, Sudibyo H, Ratna A A P, et al. The design of embedded web server for remote laboratories microcontroller system experiment [C]// TENCON 2011-2011 IEEE Region 10 Conference. 2011: 1198-1202

[8] Karia D C, Adajania V, Agrawal M, et al. Embedded Web server application based automation and monitoring system [C]// 2011 International Conference on DSignal Processing, Communication, Computing and Networking Technologies (ICSCCN). 2011: 634-637

[9] Amiri R, Elkeelany O. An embedded TCP / IP hard core for Smart Grid information and communication networks [C]// 2012 44th Southeastern Symposium on System Theory (SSST). 2012: 185-189