

面向服务的计算网格中间件的实现及性能测试

杨林峰¹, 李捷², 李陶深¹, 程海英³

(1. 广西大学计算机与电子信息学院, 南宁 530004; 2. 广西职业技术学院计算机技术系, 南宁 530226;

3. 上海大学计算机工程与科学学院, 上海 200072)

摘要: 针对 NetSolve 系统与 Web 服务的结合技术, 提出面向服务的计算网格中间件的系统结构, 在该结构中对计算网格中间件系统的 3 层功能结构进行体现, 采用 Java 技术实现基于 Web 服务与原 NetSolve 系统的数值计算网格中间件 WebSolve。应用并行预条件共轭梯度算法在新系统上进行多次大规模方程组求解, 测试结果证明该系统接口友好, 对粗粒度并行的加速比可达 $O(n)$ 。

关键词: Web 服务; 计算网格中间件; WebSolve 系统; 预条件共轭梯度法

Implementation and Performance Testing of Services-oriented Computing Grid Middleware

YANG Lin-feng¹, LI Jie², LI Tao-shen¹, CHENG Hai-ying³

(1. School of Computer, Electronics and Information, Guangxi University, Nanning 530004; 2. Department of Computer Technology, Guangxi Polytechnic, Nanning 530226; 3. School of Computer Engineering and Science, Shanghai University, Shanghai 200072)

【Abstract】 According to the integrated technology of NetSolve and Web services, this paper gives services-oriented computing grid middleware architecture, and embeds the 3-layers function architecture of computing grid middleware in it. WebSolve system based on Web services is implemented with Java. performance evaluation with parallel Precondition Conjugate Gradients(PCG) is made to solve equations set several times, whose stepped-up is $O(n)$.

【Key words】 Web services; computing grid middleware; WebSolve system; Preconditioned Conjugate Gradients(PCG)

1 概述

计算网格中间件是网格研究领域的新焦点。其中, 网格中间件的体系结构是研究热点, 网格中间件体系结构包括: 对中间件基本组成部分和各部分功能的定义和描述, 对中间件各部分相互关系与集成方法的规定, 对网格有效运行机制的刻画。目前用于科学计算的网格中间件主要有 3 种: 由美国田纳西州大学和橡树岭实验室联合开发的 NetSolve^[1]项目, 由德国爱因斯坦研究所领衔开发的 Cactus 项目和日本开发的 Ninf 项目。但这些早期的计算网格中间件未能给出一致、明确的功能扩展机制, 使得网格系统之间的互操作问题日益突出, 网格应用之间互相孤立。

同时, Web 服务技术得到了快速的发展和运用。Web 服务采用 XML 语言定义协议栈, 通过 SOAP, WSDL, UDDI, WSFL, BPEL4WS 等开放协议和标准, 提供了面向分布式应用的统一资源注册、发现、绑定和集成机制, 成为分布式环境下实现互操作的一种优秀机制。将 Web 服务技术引入计算网格中间件, 将有助于解决基于计算网格中间件所面临的应用集成、资源共享和系统互操作等问题。

本文结合 Web 服务与 NetSolve 系统, 给出面向服务的计算网格中间件 WebSolve 的系统结构和实现, 并通过实验对系统的性能进行了测试与分析。

2 NetSolve 中间件系统与 Web 服务

2.1 NetSolve 系统

NetSolve 是一个分布式计算系统实例, 它主要由 3 个部分组成: 代理, 服务器以及客户端。图 1 给出了 NetSolve 系统的整体概念结构。

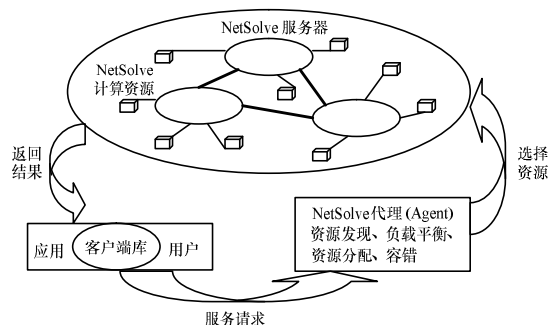


图 1 NetSolve 系统的概念结构

在系统中, 代理主要维护一个数据库, 该数据库保存了所有 NetSolve 服务器以及它们所能提供的服务和动态性能, 代理可根据这一数据库信息为客户端程序查找可以提供特定服务的服务器, 并在服务器间进行动态负载均衡, 同时监视服务器的运行情况以提供容错机制。NetSolve 服务器是一个运行于后台的守护进程, 它时刻准备接收客户端的请求。服务器可以在单独的 workstation, cluster, SMP, MPP 上运行。服务器的一个关键技术就是利用 NetSolve 的问题描述文件

基金项目: 广西自然科学基金资助项目(桂科自 0832056); 广西大学科研基金资助项目(X071103); 广西高校人才小高地建设创新团队计划基金资助项目(桂教人[2007]71号)

作者简介: 杨林峰(1979—), 男, 博士研究生, 主研方向: 并行计算, 非线性最优化; 李捷, 讲师、硕士研究生; 李陶深, 教授; 程海英, 博士研究生

收稿日期: 2008-10-27 **E-mail:** ylf@gxu.edu.cn

(PDF^[2]), 使系统能够把软件库例程集成进 NetSolve 服务器中。最后, 客户端用户使用简单的 API 访问 NetSolve 系统: 用户使用这些 API 向系统提出请求, 代理为请求选择服务器, 客户端连接服务器并输入数据开启服务, 服务器响应请求并提供服务。

2.2 Web 服务基本概念

Web 服务^[3]是独立于平台和实现的软件构件, 它能用服务描述语言描述; 在服务注册处发布; 通过标准的机制, 在运行或设计时被发现; 通过声明 API 调用, 通常是跨网络的; 与其他服务组合。Web 服务的基本架构由 3 个参与者和 3 个基本操作构成。3 个参与者分别是服务提供者、服务请求者和服务代理, 3 个基本操作分别为发布(publish)、查找(find)和绑定(bind)^[4]。服务提供者将其服务发布到服务代理的一个目录上; 当服务请求者需要调用该服务时, 首先利用服务代理提供的目录搜索该服务, 得到如何调用该服务的信息; 然后根据这些信息调用服务提供者发布的服务。当服务请求者从服务代理得到调用所需服务的信息后, 通信在服务请求者和提供者之间直接进行, 无须经过服务代理。

综上所述, 计算网格中间件 NetSolve 系统与 Web 服务有着相似的架构, 并且有相当的互补性, NetSolve 可以充分集成、扩展现有的数值计算软件包 PETSC, Linpack, Lapack, Aztec, Scalapack 等, 以实现细粒度的并行, 而 Web 服务则可以实现计算服务封装, 提高粗粒度的并行, 并且通过 Web 服务解决各种异构计算中间件之间的互操作问题。

3 面向服务的计算网格中间件

3.1 类 NetSolve 中间件结构抽象

通过对原 NetSolve 系统结构以及解决问题范围的分析, 得出计算网格中间件的基本功能为: (1)为网格用户提供使用 API; (2)对用户的任务请求进行划分、调度, 并分配资源进行计算; (3)屏蔽网格资源的分布性、动态性和异构性。为解决上述 3 个问题, 本文分 3 个模块设计计算网格中间件的功能模型: (1)问题求解模块为网格用户提供使用 API; (2)虚拟组织中间件模块管理任务、分配资源; (3)资源封装模块屏蔽各种资源差异。该功能模型如图 2 所示, 其中, EtR 代表实体资源; EcR 代表资源封装。设计一个计算网格中间件系统就是要实现这个功能模型。

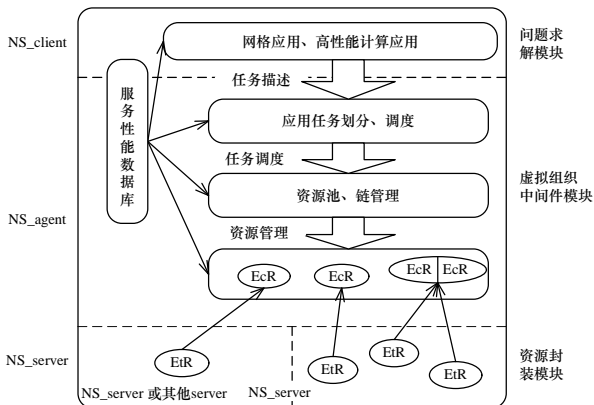


图 2 计算网格中间件系统的功能模型

在该结构中, 资源封装模块主要用来解决网格计算资源的异构性问题, 各种数值计算软件包(如 PETSC, Linpack, Lapack, Aztec, Scalapack)通过资源描述、封装成为系统统一的计算资源; 虚拟组织中间件模块是网格功能模型中的关键

部分, 它主要完成资源封装、资源池、资源链管理、应用任务划分、调度以及服务性能排序等功能; 问题求解模块借助领域相关的编程模型、人机交互机制为网格计算应用提供人机界面, 并精确地描述网格任务对资源的需求。本文讨论的 WebSolve 中间件向下兼容原 NetSolve 系统的 C, Fortran, Matlab 等接口, 同时增加了一个基于 Struts 的 Web 接口。

3.2 WebSolve 系统结构

基于计算网格中间件系统的功能模型, 结合 Web 服务以及原来的 NetSolve 系统, 本文设计了面向服务的网格计算中间件 WebSolve 系统。该系统主要由客户端、Web Services 服务器、Register Service、改进后的 NetSolve 系统组成。客户端计算机主要有 2 种方式使用系统: (1)客户端主机安装有支持 Web 服务的开发工具, 直接调用 WebSolve 系统编程 API 用于应用软件的开发。(2)用户可以直接以 IE 浏览器作为客户端, 通过浏览器直接浏览、查找并使用系统提供的服务。Web Services 服务器是采用相关平台 Axis 开发出的一个利用 NetSolve 系统提供 Web 服务的服务器。实现 Register Service 主要用于服务注册与查找。图 3 表示了 WebSolve 中间件系统的结构。

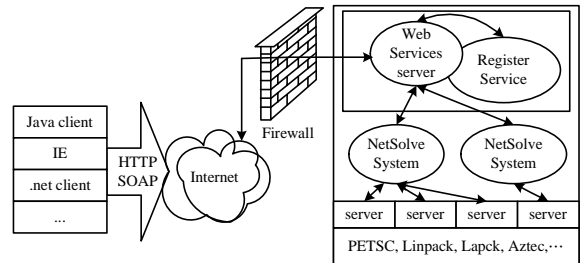


图 3 WebSolve 系统结构

4 性能测试与分析

4.1 测试环境

WebSolve 系统部署在上海大学计算机学院的 Dell 刀片式服务器集群上, 该系统的底层服务器采用 NetSolve-2.0。集群有 5 个节点: 节点 1 有 4 个 2.4 GHz 的 Intel Xeon 处理器以及 2 GB 内存和 1 MB 的 cache, 其余节点各有 2 个 2.0 GHz Intel Xeon 处理器、1 GB 内存和 512 KB 的 cache。节点间通过 100.0 Mb/s 以太网连接。节点上安装了 RedHat.9.0SMP 版。节点 1 部署了 WebSolve 系统的 SOA 层, Register Service, tomcat, apache-axis, NS_agent 和 NS_server, 其余 4 个节点上都安装了 apache-axis, NS_server 以及 Register Services。客户端在广西大学计算机学院图形图像实验室采用 IE 直接登录 WebSolve 系统。

4.2 性能测试数值结果

将有限元结构分析中常常需要求解的对称正定线性系统 $Ax=b$ (其中, A 为矩阵; x, b 为向量)进行图 4 所示的扩展^[5], 以便地进行较大粒度的并行测试。

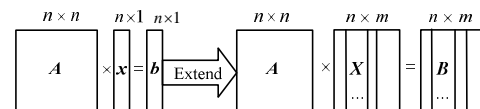


图 4 扩展的线性系统

可以将矩阵 X 与 B 的每一列都看成原系统中的一个 x 与 b , 这样, 进行 $AX=B$ 的求解相当于进行 m 次 $Ax=b$ 的求解。

实验在 4 种情况下分别进行 1 次~9 次预条件共轭梯度法 (Preconditioned Conjugate Gradients, PCG)^[5-6] 调用来求解

$AX=B$ (X 与 B 都对应地从1列扩展到9列): 1台工作站直接求解; WebSolve+1个本地 server 求解; WebSolve+1个远程 server 求解; 启动所有集群上的5个 server 并行求解。

图5是上述扩展应用的“加速比”情况。本加速比的图示较为特殊,它把在一个工作站上采用直接串行的PCG求解过程作为基准,考察采用5个WebSolve服务器并行求解的加速比。可以看到,在采用5个server进行求解时,加速比在 X 为6列($m=6$)时出现了下降,然后继续稳步上升。并且最终稳定在3.0左右。

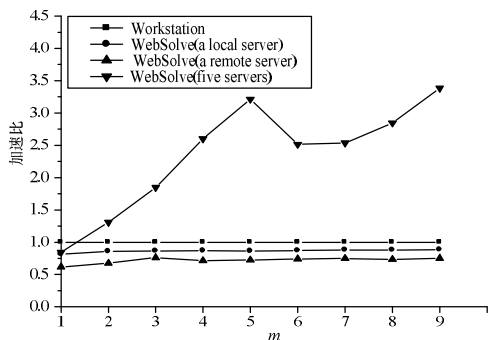


图5 扩展 $AX=B$ 求解的加速比

分析如下: 设WebSolve系统中server的个数为 s , $AX=B$ 系统中 X 的列数为 m (为了方便讨论, 假设 $m \leq 2s$), 每个任务求解的时间分别为 t_1, t_2, \dots, t_m , 则采用WebSolve系统求解 $AX=B$ 的加速比 r 为

$$r = \begin{cases} \frac{t_1 + t_2 + \dots + t_m}{\max(t_1, t_2, \dots, t_m)} & m = s \\ \frac{t_1 + t_2 + \dots + t_s + t_m}{\min(t_1, t_2, \dots, t_s) + t_m} & m = s + 1 \end{cases}$$

从上述公式可以看出, 每当 $m=ks+1$ (k 为自然数)时, 加

速比就会出现下降, 再逐渐上升。

5 结束语

本文结合Web服务技术与NetSolve系统实现了中间件WebSolve系统, 并利用WebSolve的服务非阻塞调用实现了并行PCG扩展求解器。扩展的PCG求解器在WebSolve系统上的加速比可达到 $O(n)$ 数量级, 其中, n 为WebSolve服务器个数, 获得的加速比较理想。由此可见, WebSolve系统的服务非阻塞调用适合大粒度、计算密集型任务的并行化。并且WebSolve系统可以对任务负载进行均衡分配, 当任务粒度大小差异比较大时, 优势比较明显。

参考文献

- [1] Arnold D, Casanova H, Dongarra J. Innovations of the NetSolve Grid Computing System[J]. Concurrency and Computation: Practice and Experience, 2002, 14(13):1457-1480.
- [2] Vadhiyar S, Dongarra J, YarKhan A. GrADSolve-RPC for High Performance Computing on the Grid[C]//Proceedings of the 9th International Euro-Par Conference. Berlin, Germany: [s. n.], 2003.
- [3] W3C Working Group. Web Services Architecture[EB/OL]. [2007-12-30]. <http://www.w3.org/TR/ws-arch/>.
- [4] 边小凡, 张宝山. 基于商业逻辑的Web服务合成方法的研究[J]. 计算机工程与设计, 2006, 27(13): 2381-2382.
- [5] 杨林峰, 张武, 付朝江. 基于NetSolve的并行PCG实现及其性能分析[J]. 计算机工程, 2005, 31(20): 110-112.
- [6] Coleman T F. Linearly Constrained Optimization and Projected Preconditioned Conjugate Gradients[C]//Proceedings of the 5th SIAM Conference on Applied Linear Algebra. Utah, America: [s. n.], 1994.

(上接第267页)

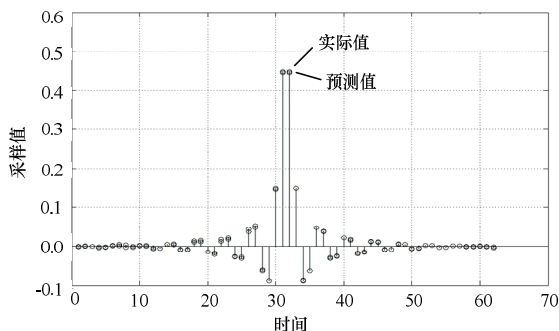


图3 LMS自适应滤波模拟仿真图

4 结束语

本文介绍了传统空气清新器空气信息采集系统的设计和实现。传统的空气清新器采集系统缺乏一定的自适应性, 只能简单地响应传感器传达的信号, 空气清新器没有一定的优化控制策略。设计出了一种新型的智能的空气清新器。利用自适应滤波器的自适应特点, 建立起有预测功能的, 提高抗干扰性能的空气清新器控制模型。由于引入了LMS自适应滤波器, 该模型可以在信息完全未知的条件下设计滤波器。引入了该模型系统的新型空气清新器能对不同环境下随机的空气信息进行采集, 然后通过LMS的预测能力对空气质量信息

进行快速获取, 并具有一定的预知性。通过预测的估计值迅速地产生控制策略, 开启执行系统, 从而令空气质量快速逼近期望值。本文设计的空气信息采集系统可以广泛应用于各种复杂的环境, 并可以应用于传统的空气清新器的改良, 为后续新一代的空气清新器的技术研发提供了一个很好的平台。但为了实现算法的优化控制, 芯片选择跟一般的空气清新器选择利用单片机不一样, 要选用运算能力更强的DSP, 在一定程度增加了成本。但是将来的发展方向总是向更高性能的智能发展的, 并且电子产品也会越来越便宜, 成本劣势只会越来越小。

参考文献

- [1] Haykin S. Adaptive Filter Theory[M]. 4th ed. Beijing: Publishing House of Electronic Industry, 2006.
- [2] 陈丰, 陈禾. 基于SOPC的LMS自适应滤波算法实现[EB/OL]. (2006-05-06). <http://www.paper.edu.cn>.
- [3] 胡晔, 姚鹏翼, 陈明. Protel 99 SE原理图与PCB设计教程[M]. 北京: 机械工业出版社, 2005.
- [4] 王益根. 基于LMS的自适应滤波器典型应用的MATLAB实现[J]. 扬州职业大学学报, 2007, 11(2): 35-37.

