

并程序性能可视化分析系统的研究与实现

马 捷

国家智能计算机研究开发中心

中国科学院计算技术研究所

北京 2704 信箱 (100080)

majie@ncic.ac.cn

摘 要

随着网络计算、并行技术及并行应用的不断发展, 并程序的设计、开发、调试、优化成为并行技术应用及网络计算的关键。本文提出了一个性能可视化分析系统的框架模型。VENUS 系统是基于这一模型的一个性能可视化分析与优化的软件系统, 它的可视化部分通过使用动态调整视图播放细节度等技术, 使 VENUS 系统能够高效、准确地展现并程序性能, 提高了系统的可用性。本文详细论述了性能可视化模型的结构, 并在文章中给出了 VENUS 系统的性能。

Research and Implementation of a Parallel Program Visual Performance Analyzing System

Ma Jie

National Research Center for Intelligent Computing Systems

Institute of Computing Technology

Chinese Academy of Sciences

Beijing 100080, P.R.China

majie@ncic.ac.cn

Abstract

With the development of the network computing, parallel computing technique and application, parallel program designing, debugging and optimizing become a key technique of application and network computing. In this paper, a frame model of the visual performance analyzing system is introduced. VENUS is a visual performance analyzing and optimizing system based on this model. By using dynamic-adjust-play-scale technique, VENUS can visualize the performance of parallel program more efficiently. In this paper, the structure of the model is discussed in detail. And the

performance of VENUS is given.

1. 引言

随着网络、并行技术及并行应用的不断发展，并程序的设计、开发、调试、优化成为并行技术应用及网络计算的关键。由于并行系统硬件体系结构、网络协议、软件平台等方面的差异，并程序在不同平台上的运行效率差别也很大。因此，并程序的优化就成为并行应用方面的一个关键性问题。为了充分发挥并行计算机的潜力，提高并程序的性能，国内外开展了大量的关于并程序开发及网络计算的支撑环境的研究工作，其中并程序的性能分析是重要研究之一。并程序的性能分析大致采用三种方法：基于模型的性能分析方法；基于测量的性能分析方法；基于模拟的性能分析方法。可视化是近年来在性能数据处理分析中经常采用的技术。将数据以可视化的方式表现出来，对于了解数据变化的趋势、直观表现数据的内在含义有着十分重要的作用。该技术给用户提供了直观明确的界面，有助于对系统性能的分析，定位系统瓶颈。

国外的相关研究主要有 ParaGraph、Paradyn、EDPEPPS 仿真系统等。ParaGraph 是用于显示并行应用程序的实际运行行为和性能的软件，它提供一个可视的重放机制来重放真实并行应用程序在实际并行平台上的运行事件记录。Paradyn 是对大规模并行应用程序进行性能分析的一个辅助软件工具包。它通过使用动态指令插入技术和自动瓶颈测定技术，在线式地监测和分析并行应用程序的性能。EDPEPPS 是基于快速原型机制进行并程序设计的一个工具，它使用一种面向仿真的语言进行程序原型的设计，并在特定的仿真平台上进行仿真，最终将所获得的性能数据以可视化的方式表现出来。

国内的相关研究主要有清华的 IPCE 和中国科学院的 ParaVision。IPCE 是清华大学研制的可视化人机交互集成开发环境。它是在对 XPVM 和 PVM 修改的基础上集成的一个综合性的并程序开发环境。ParaVision 是曙光天潮系列机上的并程序可视化工具。

此外，基于 PVM 的一些调试工具也具有一定的并程序分析与可视化的功能。现在常见的此类工具软件有 XPVM、PGPVM、PVaniM 2.0 等。

本文结合 863 项目大规模并行处理体系结构的关键技术，针对基于消息传递机制的并行系统性能进行可视化及分析的研究。文章提出了一个对并行应用程序进行性能可视化分析的系统的框架模型，并按照这一模型实现了 VENUS 系统。VENUS 系统的主要特点是：

- 模块式(插件式)系统结构，有利于系统功能的划分及在不同系统上的移植；
- 基于修改源码的事件跟踪法获取并程序运行性能，实时测量程序运行情况；
- 三层视图模型，增强可视化模式(视图)的可扩展性；
- 二级视图控制模型，提高系统的易管理性及易用性；
- 基于测量建立系统硬件模型，基于性能瓶颈假设进行验证分析的性能分析；
- 分析过程中采用可信度模型，进行模糊推理，提高能性能瓶颈的探测效果；
- 友好的用户界面，提高系统的易用性。

2. VENUS 系统的框架模型

VENUS 系统采用了基于测量的性能分析方法。它不依赖于对硬件平台所建立的模型，直接根据实际测量所得的数据(性能数据集)进行分析，从而摆脱了分析系统对硬件平台的依赖性。性能可视化分析系统从功能上可以分为四个模块：性能监测模块、性能可视化模块、性能分析模块和用户界面及辅助管理模块 (如图 1所示)。

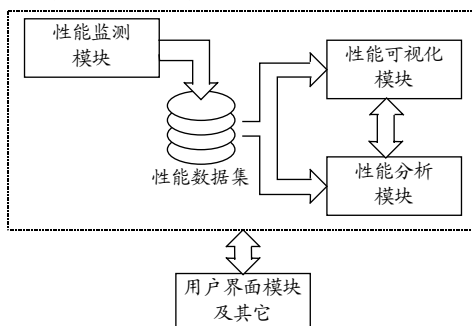


图1 性能可视化分析系统模型的功能框图

性能监测模块是系统中最基本的模块，整个系统通过它来获取应用程序在硬件平台上的性能，并以性能数据集的形式表示出来。一个应用程序的运行性能可以通过记录该程序的运行时间、同步、通信及其它一些重要事件发生的时间(时刻)和相关信息来表示。我们可以通过对上述数据(性能数据集)的分析，重构出其运行状况或其某一侧面的性能状况。

性能可视化模块是系统中的一个主要模块。简单地说，它的功能就是把系统内部的数据以直观的方式表示出来。首先，它需要对性能数据集中的数据进行分析理解，重构出应用程序运行的性能状况；然后，根据用户所需，以不同的形式表示出应用程序性能侧面。另一方面，系统中性能分析模块所得出的优化结论也应在这里表示出来。

性能分析模块是系统中另一个主要模块。它分析系统内部所获取的性能数据集中的相关性能数据，根据已有的经验(模型)以及系统内部的逻辑模型，通过推理产生出相应的分析结果。为了能够更好地进行性能分析，这一模块采用了基于测量建立系统硬件模型，基于性能瓶颈假设进行验证分析的方式，同时在分析过程中采用可信度模型，进行模糊推理，提高性能瓶颈的探测效果。

用户界面模块及其它辅助功能模块为系统提供一个友好的可视界面和一些管理功能。在本文所述的框架模型中，没有对用户界面的实现做过多的限制，只是从总体上提出了一些设计原则，作为系统在实现过程中的规范。

3. VENUS 系统的可视化模块

性能可视化是科学可视化的一种，它把并行应用程序的性能以可视的方式展现给用户(程序员)。可视化的性能可以是直接由性能监测模块所监测到的数据，或经由性能分析模块分析所得的结论，也可以是由可视化模块根据已有的数据进行统计分析所得的结论。严格地说，可视化部分所做的简单统计分析也应属于性能分析的范畴，但由于这部分工作过于简单，甚至可以在读取性能数据的同时进行，因此，在这里所提及的可视化也包括这一类的简单统计分析。

在 VENUS 系统中，可视化模块以三层视图模型为中心，并以二级视图控制模型对性能可视化过程进行控制管理。

● 三层视图模型

如图 2所示，三层视图模型由性能数据分析抽取层，性能数据可视化表示层和可视化数据显示层组成。

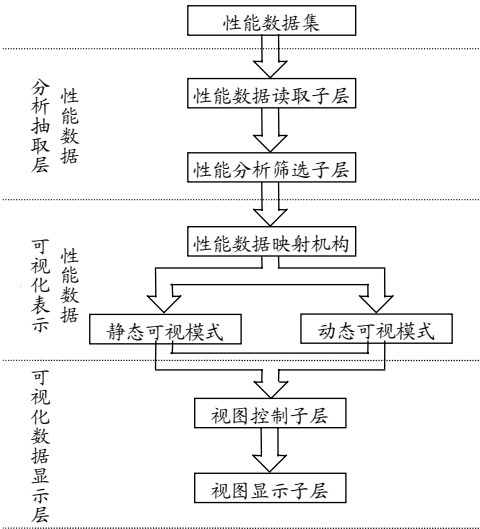


图2 三层视图模型

性能数据分析抽取层由性能数据集中获取性能数据，是进行可视化的第一步。这一层模型主要用于解决可视化的内容问题。一个视图通过这一层与性能监测模块交换信息，系统则通过这一层来控制视图所表示的性能侧面。

性能数据可视化表示层由一个性能数据映射机构和若干视图模式组成。性能数据映射机构将筛选分析后的性能数据映射到不同的视图模式上，形成以可视形式表示的数据流(可视化数据)。视图可以分为静态类视图和动态类视图两大类模式。静态类视图将要表现的性能数据一次性显示出来，使程序员可以获得并行应用程序性能的总体特征。动态类视图反映了并行应用程序性能的变化过程，它按照时间顺序将程序运行中每一时刻(或时间区间)的性能显示出来，使程序员可以对系统性能的变化趋势有所了解。这类视图适于表现程序某一方面(如消息通信等)随着程序运行时间的推移而变化的情况。

可视化数据显示层由视图控制子层和视图显示子层构成。视图控制子层按照视图显示(播放)的要求，将上层形成的可视数据送给视图显示子层，由视图显示子层完成视图在物理设备上的显示。视图控制子层由性能数据可视化表示层获得待显示的可视数据后，按照相应的视图模式进行控制。对于静态类视图模式，其控制子层功能较为简单，主要包括视图的初始化及复位控制、数据查询控制及视图布局控制。对于动态类视图模式，其视图控制子层还包括视图播放控制及视图关联控制等。

视图显示子层按照视图控制子层的控制在物理设备上显示视图。这一子层将设备的物理特性对上层屏蔽，按照系统设置为上层绘制视图的几何原语赋以视觉特性，即确定视图的帧、颜

色等，并启动视图绘制过程，完成视图显示。

● 二级视图管理机制

视图只表现了并行应用程序性能的一个侧面。为了能够全面地表现程序的性能，就需要使用多个视图共同表现程序的性能。各个视图相互关联、相互参照，形成程序完整的性能模型。为了能够更好地管理各个视图的显示与关联，我们采用了视图管理器和视图二级管理机制。

视图管理器用于视图的全局管理。它利用每个视图所提供的全局管理控制接口控制各个视图，以保证视图间的相关性、一致性。视图管理器从高层着手，在视图统一管理的同时，为用户提供了一个良好的用户界面，方便用户使用。视图级管理用于单一视图的显示控制。它是二级性能视图管理机制的内层。为外层视图管理器的全局管理提供一致的控制接口。

VENUS 系统提供了系统信息类视图、利用率类视图、通信类视图、并行任务类视图等四类共 17 种视图，其中包括了能有效表示并程序执行情况的任务生成树(Task Spawn Tree)、利用率 Kiviat 图(Utilization Kiviat)、通信动画(Communication Animation)、消息队列(Message Queue)、时空图(Space-time)和并行任务 Gantt 图(Task Gantt)等。

表1 VENUS 系统对并行应用程序性能优化的测试结果

应用程序	执行时间	分析结果	优化提示	优化程序执行时间
并行遗传算法(PGA)解决旅行商问题(TSP)	120.33	同步相持时间过长	调整计算与通信的位置	77.44
快速排序问题	12.24	消息通信过多	减少消息通信次数	8.99
线性插值问题	2.29	消息通信过多	减少消息通信次数	0.80
矩阵的 LU 分解问题	19.01	并行任务粒度过细	增大任务粒度	1.65
高斯消去法解线性方程组	22.16	并行任务粒度过细，通信次数过多	增大任务粒度	2.97

4. 性能分析

为了对 VENUS 系统的性能进行分析，这里对一组典型并行应用程序进行测试。通过对这些典型应用运行性能的实测、分析与优化，对用 VENUS 系统进行性能分析优化的效果进行评价。

表 1列出了 VENUS 系统在清华工作站机群上测试的结果。通过对一组并行应用程序性能优化的测试可以得出 VENUS 系统能够有效地对并行应用程序进行分析优化。

5. 结论及进一步工作

本文在结合 863 项目大规模并行处理体系结构的关键技术的研究过程中，针对基于消息传递机制的并行系统性能进行可视化及分析的研究。从理论上建立可视化的基本模型，并根据这一模型实现了 VENUS 系统。VENUS 系统可应用于曙光计算机(MPP)结构、可扩展机群硬件平台以及采用以太网连接的局域网平台。通过对应用程序的优化测试，VENUS 系统能够有效地对并程序进行优化，并具有一定的调试功能。

在以后的工作中，我们还需对 VENUS 系统做以下的改进：

- 性能数据可视化表现形式的进一步研究
- 系统性能的改善
- 使用方便性的提高

参考文献

- [1] E.Kraemer, J.Stasko. "The Visualization of Parallel System: an Overview". J. Parallel and Distributed Computing, Vol.28, No.11, pp.30~36
- [2] Michael T. Heath (University of Illinois), Jennifer A. Etheridge (Oak Ridge National Laboratory), "Visualizing the Performance of Parallel Programs", IEEE Software, September 1991, pages 29~39.
- [3] Barton P. Miller, Jeffrey K. Hollingsworth, Mark D. Callaghan, "The Paradyn Parallel Performance Tools and PVM"
- [4] D. Kranzlmuller, R. Koppler, S. Grabner, Ch. Holzner, J. Volkert, "Parallel Program Visualization with MUCH".
- [5] Adam Beguelin and Vaidy Sunderam, "Tools for Monitoring, Debugging, and Programming in PVM", PVM'96, pages 7~13.
- [6] Thierry Delaitre, Francois Spies, Stephen Winter, "Simulation Modeling of Parallel Systems in the EDPEPPS Project", UK Parallel '96, pages 1~13.
- [7] V. S. Sunderam, G. A. Geist, J. J. Dongarra, etal. "The PVM Concurrent Computing System: Evolution, Experiences and Trends". J. Parallel Computing, 1994, Vol.20, No.4, pp.531~546