Title:
JavaPIE: An Internet Research Extension of the Parallel Programming Interactive Environment

Authors: Yu,Y.,Wang,Q.,Shi,W.,Zang,B.,Zhu,C.Q.

Address: Institute of Parallel Processing, Fudan University,Shanghai 200433
Email: yjyu@moboss.fudan.edu.cn
Date: Mar 10, 1998

Abstract

Keywords Internet, Java, Parallel Programming Environment, User Interaction

JavaPIE：在 Internet 上延伸并行编程交互环境研究

俞一峻，王琦，施武，臧斌宇，朱传琪[1]
复旦大学并行处理研究所，上海 200433
Email:yjyu@moboss.fudan.edu.cn

摘要

关键词 Internet, Java, 并行编程交互环境

1. 背景

在不改变语义的前提下使多年来积累的串行软件尽量并行化和优化，是解决并行软件开发的简捷途径。并行化编译通过对串行源程序进行各种分析，能利用存在于这些程序中的潜在并行性，把串行程序自动转换成对应的并行形式，有效地提高它们在并行计算机上执行的性能，同时也解决了在不同并行计算机之间移植代码的难题，使在并行机上继承已有串行程序成为可能。作为连接并行计算机与应用程序之间的桥梁，并行化技术必然成为并行计算机上不可缺少的系统软件的组成要素 。目前国际上技术领先的自动并行化系统 SUIF[5], Polaris, FPT[11]和复旦大学开发的 AFT[12]等对 Fortran 已经作出了一定效果。而基于 AFT 的成功经验，我们同明尼苏达大学[2]合作正在建设的并行程序编译研究平台 Agassiz 更注重对并行化和各种优化技术开展国际合作研究。

尽管对于许多情形自动并行化编译技术都很有效，但是由于自动并行化编译器必须采取保守的处理方法，即，如果编译器不能证明其并行执行能导致与串行执行相同的结果，它就只能产生串行代码，所以仍然存在一些场合不能运用自动并行化编译技术。例如影响数据相关性分析精度的符号表达式和非线性下标表达式、过程调用和依赖于输入的复杂控制流、临时变量、递归变量和归约变量识别、下标化的数组等等都成为自动并行化的拦路虎。

要超越这些障碍，就需要借助程序员人工的干预。由于并行化编译器对绝大多数较简单的情况自动利用并行性，节省了程序员大量时间，程序员反过来只要集中注意那些更复杂的情况来进一步提高目标程序的性能。为了结合自动并行化编译技术和用户交互，提高并行化能力，国际上开展了对并行编程交互环境和交互编程工具的研究 ，如 PTOOL, R^n, ParaScope[1], Faust, SUPERB, PAT, ParaDyn，我们与 Gent 大学[3]合作开发的 PEFPT[2,6,7,9,10]和复旦大学独立开发的 ParaPIE 等等[8]。

作为有效的并行程序分析环境，ParaPIE/PEFPT 的实践活动取得了很好的应用效果，例如被公认为难以并行化的三个 SPEC95 测试程序也通过交互分析找出了影响并行化的症结，得到了一定的并行化结果 [3]。但是作为理想中的实用并行化编译技术开放研究平台 ，PEFPT/ParaPIE 尚存在一定的差距，主要体现在以下几个方面：

●缺乏充分的软件移植性、伸缩性和开放性: 即使使用 C/Motif 开发，在不同 UNIX 开发平台上实际上仍存在着许多不一致性，增加了软件移植难度。我们的 AFT/ParaPIE 系统在经过 HP9000，SUN Sparc, IBM RS6000, DEC Alpha AXP, Siemens RM600 等一系列平台移植后才逐渐趋于稳定，而且因为基于不同并行化编译工具实现了不同的程序中间表示，这也使并行编程环境缺乏功能相互移植的足够开放性。

●信息文档组织欠缺与松散：参与系统设计的人员之间、系统各个时期的文档之间缺乏组织，系统文档与用户说明之间缺乏一致性和可维护性，容易导致重要功能缺乏必要说明，以及重要说明文档缺乏检索支持等缺憾。

●不能及时更新以满足需求变化，跟不上合作研究共享技术的需要：由于工具集成于一个环境中，不同环境（如 ParaPIE 同 PEFPT）之间新添的功能，即使由同一小组开发也必须重新编码才能互相移植，其他研究小组开发的新工具或技术由于我们缺乏对其内部实现结构的足够了解，也就更难加入不断更新的环境中了。

因此，为了达到更理想的移植性、开放性、及时共享性，我们在 Internet 上设计了一个编程环境 JavaPIE(Java Parallel programming Internet Environment), 利用 Internet 上的 Java 语

[1] 本课题部分得到国家自然科学基金，国家 863 攀登计划，欧盟 ITDC-94 合作项目和博士点基金资助。
[2] 美国明尼苏达大学计算机系主页：http://www.cs.umn.edu
[3] 比利时皇家根特大学的电子工程系主页：http://www.elis.rug.ac.be

言[13]实现部分交互功能，并有机地延伸出来，为蓬勃地开展并行编程交互环境国际化合作研究开辟了广阔的前景。本文下面分节介绍已经获得成果的 ParaPIE 系统和正在开发中的 JavaPIE 系统的实现原理，希望对从事编程环境研究的同仁有所借鉴。

2. 并行编程交互环境 ParaPIE

ParaPIE 结合自动并行化编译技术和用户交互与一体，目标是帮助程序员继承过去岁月里积累的串行程序资源；减轻程序员开发并行程序的负担；减少在不同并行计算机间移植代码的难度，帮助程序员理解和分析程序的并行语义；更有效地利用优化和并行化技术；提供研究并行化编译技术的平台，验证各种并行化技术的正确性和有效性。

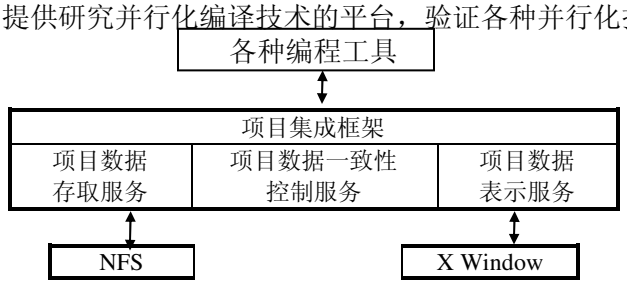


图 1、ParaPIE的系统结构

为了达到以上目标，ParaPIE 将程序分析、可视化和交互或自动变换工具，如程序编辑、程序并行语义分析和显示、并行和优化程序变换、并行程序性能分析等等，有效地集成在一个统一的编程交 互环境中 。我们在 ParaPIE 中试图基于并行化技术，实现基于数据分解的程序内部中间表示，并在保证用户界面独立性和环境集成一致性的原则下为用户提供帮助开发高性能并行程序的一系列工具。系统层次结构反映了基本服务和编程工具之间的集成界面层次，如图 1所示。编程工具利用集成框架的服务，这些服务调用底层支持软件实现。

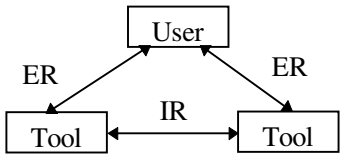


图 2、ParaPIE系统工具与程序员之间的关系

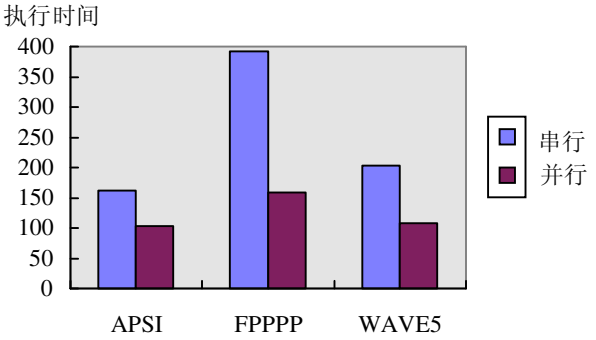
在自动并行化编译过程中，*中间表示*(IR) 反映了工具互相之间交换信息的不为用户所见的内部数据结构。在交互编程环境中，用户不可见的许多中间表示 IR 必须转换为终极用户可以见到的*外部表示*(ER)，从而允许程序员通过直接修改外部表示 ER 干预中间表示，辅助和引导系统完成各种功能，图 2反映了 ParaPIE 工具与程序员之间的相互关系。表 1 反映了 ParaPIE 的功能侧面。与象 AFT 这样的自动并行化编译器不同，ParaPIE 的功能涉及更多用户交互，从用户得到的反馈能用来修正系统对程序的解释，导致对程序更有效的并行化。

表 1、ParaPIE系统集成的主要工具和数据集

工具名称	利用的中间表示	可视化的外部表示	用户交互操作
程序编辑器	程序正文缓冲区	程序正文	浏览和编辑程序正文
语法分析器	抽象语法树	串行源程序	打开文件并初始化项目数据表
代码生成器	抽象语法树	并行目标程序	选择目标代码类型来保存文件
数据流分析器	变量数据流信息	变量信息表	指定私有化变量和数组范围
变量过滤器	同上	同上	过滤出感兴趣变量信息
相关性分析器	数据相关性信息	变量相关信息表	删除虚假相关对、检测递归相关语句
相关性过滤器	同上	同上	根据语句范围、变量范围、相关类型过滤出感兴趣相关信息
语句相关图分析器	语句级数据相关性	语句相关图	语句相关图浏览、过滤查询
迭代相关图分析器	跨循环迭代相关性	循环迭代相关图	循环迭代相关图浏览和过滤查询
调用图分析器	过程调用关系	过程调用图	查找过程
结构化变换工具	控制流图	结构化的源程序	对指定程序结构化
并行化变换工具	结构化程序的语法树、数据相关信息	程序段是否可并行化及可用的并行化技术	用指定并行化技术对指定程序段并行化或自动并行化
程序对照工具	源-目标语法树的语句对照关系	语句对照表	对照变换前后的源程序和目标程序
程序计算量分析器	插入计时语句	过程及循环计算量表	数据排序和统计图表显示
最大并行度分析器	插入时间映射变量和计时语句	循环最大并行度表	同上
项目管理器	工具及外部数据项间依赖关系	项目数据项	调用相应工具新建、修改项目数据项
宏操作记录器	用户操作事件序列	操作宏记录文件	记忆和重放一个操作序列

图形打印支持工具	绘图命令原语	Postscript 格式文件	生成、显示和打印各种图形

表 2、利用ParaPIE对SPEC95测试程序包的并行化效果



SPECcfp95 benchmarks 由 10 个串行 Fortran 程序组成，是衡量计算机性能及并行化编译效果的通用测试程序包 [4]。除了其中 APPLU, HYDRO2D, MGRID, SU2COR, SWIM, TOMCATV, TURB3D 七个程序可以并行化外，141.APSI, 145.FPPPP and 146.WAVE5 这三个程序曾经被公认为是极难并行化或几乎无法由程序员手工并行化的 [3]。因此先进的自动并行化编译工具（AFT 和 SUIF）能对其中 APPLU, HYDRO2D, MGRID, SU2COR, SWIM, TOMCATV, TURB3D，都取得了显著的效果，但是对于另外三个程序 APSI, FPPPP 和 WAVE5 却无能为力，无法并行化（加速比为 1）。完成 ParaPIE 的设计后，作为 ParaPIE 应用实验，我们通过 ParaPIE 的程序计算量分析工具、最大潜在并行度分析工具、相关性分析工具和数组私有化分析工具等，对这些程序进行人机交互辅助分析和变换，取得了明显的并行化加速效果。为了说明 ParaPIE 的应用效果，我们选取硬件环境（具有 4 个 MIPS R10000 处理器的 SGI Origin 2000 并行计算机）作为实验条件，通过分别将串行程序、并行化后的并行程序在上述环境下执行，根据得到的实际执行时间绘制了表 2。

3. JavaPIE 实现原理

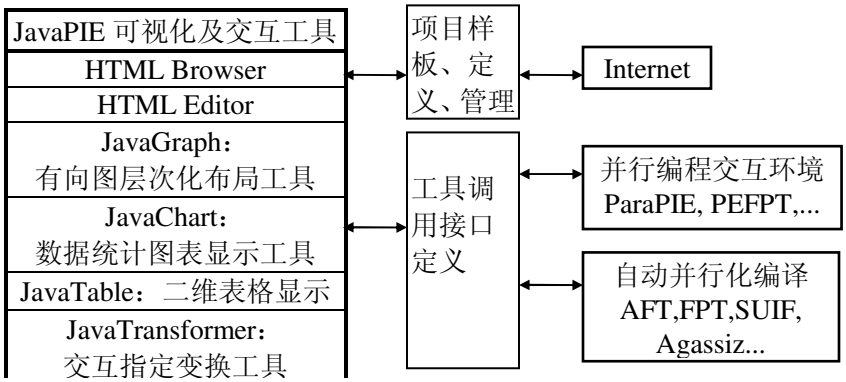


图 3、JavaPIE 的系统结构

JavaPIE 的目标是利用 Internet 更好地开展并行编程交互环境的国际合作研究，因此如何将 Internet 的网络资源同 ParaPIE 的交互功能相结合，对不同层次和耦合度的合作研究项目组提供安全可靠而又及时的信息交流，是设计 JavaPIE 时必须考虑的决定性因素。

表 3、JavaPIE系统的主要工具和数据集

JavaPIE 主要工具	功能描述	对应 AFT/ParaPIE 支持
HTML 编辑器	编辑串行源程序正文	程序编辑器、代码生成器、程序对照工具
HTML 编辑器	编辑项目文档与样板脚本	项目管理器、宏操作记录
HTML 浏览器	获取 Internet 信息	无
JavaTable	观察变量信息表	数据流分析器、变量过滤器
JavaTable	观察数据相关信息表	数据相关性分析器、数据相关过滤器
JavaGraph	显示数据相关图	数据相关图分析器：操作、语句、任务、循环迭代级
JavaGraph	显示控制流图	结构化变换工具
JavaGraph	显示过程调用图	调用图分析器
JavaChart	循环最大并行性数据	最大并行度分析器
JavaChart	过程及循环计算量数据	程序计算量分析器
JavaTransformer	选择程序段及变换	并行化变换工具

[4] SPEC benchmarks 国际组织的 Internet 主页：http://www.specbench.org

从 JavaPIE 的系统结构（图 3）和主要工具（表 3）我们可以看到 JavaPIE 与 ParaPIE 的继承调用关系。由于 JavaPIE 克服了依赖于程序内部表示的固有缺陷，所以能够方便地融合不同并行化编译工具和并行编程环境的功能，通过项目管理通过超文本 HTML 有效地组织文档和描述实验过程和可重复实验结果，为同行交流程序并行化研究的心得提供了共同的平台。JavaGraph、JavaChart 、JavaTable、JavaTransformer，可以使用 Java Applet 实现并嵌入到 html 页中。由于统计图、扩展表格、以及菜单式交互程序设计可以借鉴比较一般的 Java 应用接口工具包 awt 加以实现，这里不加赘述。对于 JavaGraph，由于过程调用图，语句相关图，数据相关图等是程序语义信息的可视化图形显示的重要功能，所以必须设计一个能够清楚显示带圈有向图的图形布局算法，并允许用户交互拖曳调整图形布局，以达到直观、直接显示程序语义信息的目标。下面就简单介绍 JavaGraph 这个重要算法的实现原理：

INPUT: graphwidht,graphheight,Graph = <N,E>，N 为结点集合，E 为边集合。

OUTPUT: {node 的 x,y 坐标| node in N}

ALGORITHM

1. 根据边集 E 建立结点集 N

   N = {}, for e=<n1,n2> in E do N = N+{n1,n2} end
2. 根据深度优先树为结点编号(dfn: N→Z)
3. 强连通分支划分，为结点编号强连通分支中最小的 dfn (sccn: N→Z)
4. 垂直分层布局:

   N1 = {}，maxLevel = 0，

   repeat

   for n in N do

   cnt(n) = count{n1|<n1,n> in E and not n1 in N1 and scc(n1)<>scc(n)}

   if cnt(n)=0 then level(n)=maxLevel

   end

   N1=N1+{n|level(n)=maxLevel}

   maxLevel++

   until N1=N

   //以下是可选的步骤，通过增加虚拟结点避免跨层次边、点交叉

   for e=<n_1,n_2> in E and delta=level(n_2)-level(n_1)>=2 do

   N = N+{nn_k| level(nn_k)=j(k)=level(n_1)+k, k=1,...,delta-1}

   E = E-{<n_1,n_2>}+{<n_1,nn_1>,...,<nn_delta-1,n_2>}

   end
5. 坐标计算:

   for level from 0 to maxLevel do maxPosition(level) = 0 end

   for n in N do position(n)=maxPosition(level(n))++ end

   for n in N do

   x(n)=graphwidth*(position(n)+0.5)/maxPosition(level(n))

   y(n)=graphheight*(level(n)+0.5)/maxLevel;

   end

图 4、布局算法描述

下面介绍两个 JavaPIE 应用的样板，一个是结合程序计算量分析的过程调用图显示（JavaGraph+JavaChart)(图 5)，另一个是操作级和语句级数据相关图显示（JavaGraph+JavaGraph)（图 6），具体实现参见[4,8]。

- 从 Linpack.f 测试程序得到的程序计算量数据反映在 JavaGraph 过程调用图的调用边上，表示子过程或嵌套子循环执行时间占上一级过程或循环执行时间的百分比，通过单击分层调用图的不同结点，在 JavaChart 中立即刷新统计饼图以反映新的整体-局部调用关系。此外，由于 JavaGraph 支持对强连通图的处理（强连通分支结点布局在同一水平层中），所以不仅可以显示 Fortran77 的非递归过程调用图，也可以显示 C 或 Fortran90 的递归过程调用图。
- 再看下面例子：

  (1) DO I=2, N

  (2) A(i)=B(i)+C(i)

(3)　D(i)=(A(i-1)+A(i+1))/2
(4) ENDDO
如果分析语句级数据相关图，则会显示流相关 $S2^{f}\rightarrow S3$ 和反相关 $S2^{a}\rightarrow S3$ 构成一个圈，而通过观察操作级数据相关图，则发现反相关 $S2^{a}\rightarrow S3$ 并不引起相关圈。

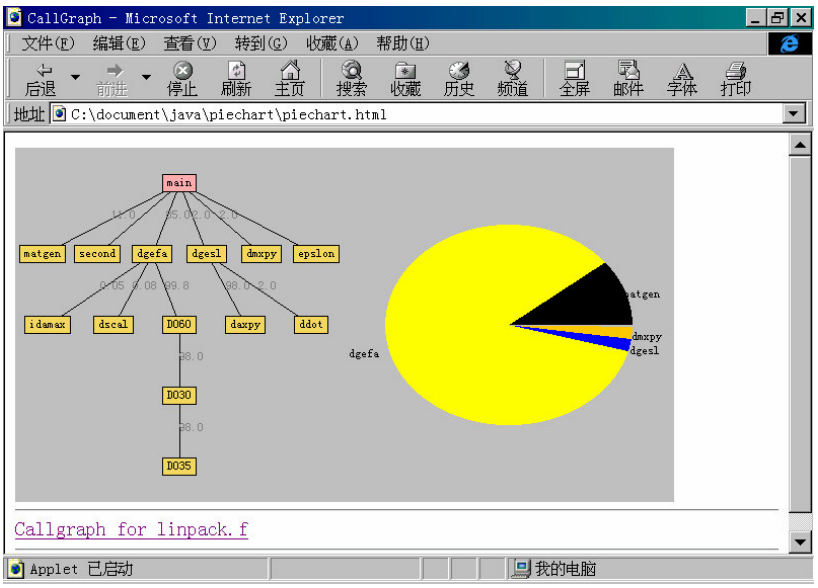


图 5、JavaPIE过程调用图及程序计算量统计图应用样板

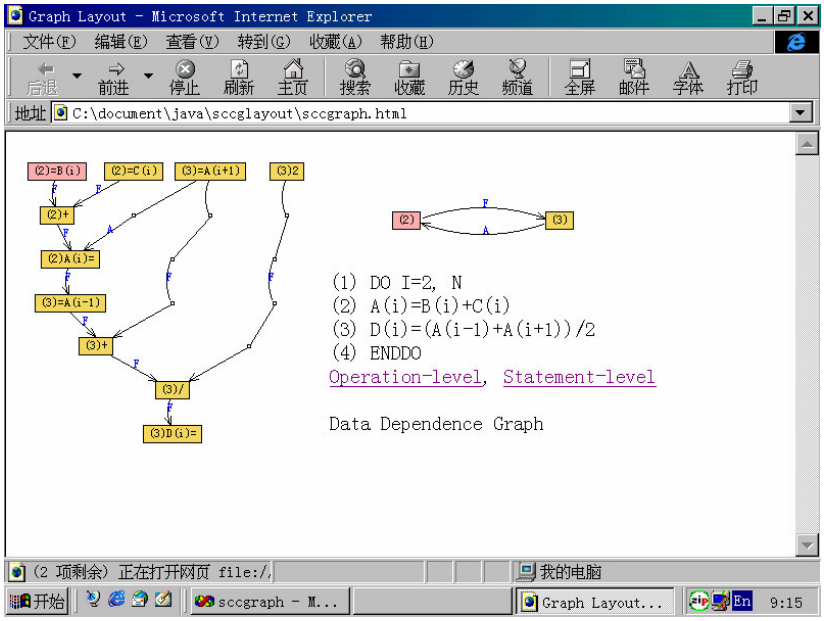


图 6、JavaPIE操作级和语句级数据相关图应用样板

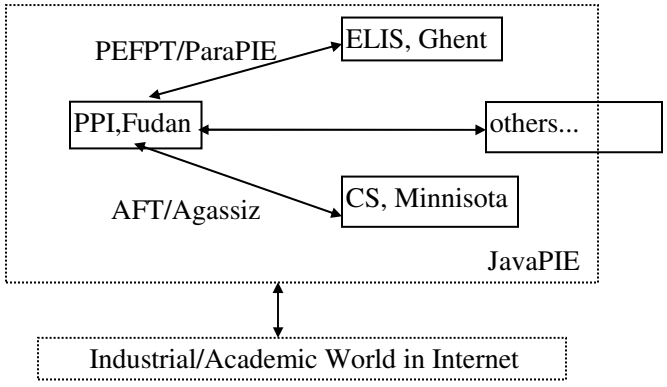


图 7、JavaPIE的现状

4. 结论

在过去短短的几年中，我们同国际合作伙伴的信息交流已经从代价高昂的人员互访，软盘或磁带携带数据的脱机方式改变成通过 Email 交流信息、通过 FTP 传递数据、通过 HTTP 建立交互媒介的联机方式。与此同时，ParaPIE 系统也日见成熟，通过 Java 在 Internet 上的扩展（图 7），我们的 JavaPIE 已经同国内外的研究伙伴建立了更为密切的联系。

参考文献

1. Cooper,K.et al,"The ParaScope Parallel Programming Environment",Proceedings of the IEEE, 81(2),Feb. 1993.
2. D'Hollander,E.H., "Partitioning and labeling of loops by unimodular transformations", IEEE Trans. on Parallel and Distributed Systems, 3(4):465-476,1992.
3. Reilly,J., “A Brief Introduction to The SPEC CPU95 Benchmarks”, SPEC Newsletter, Sep 1995.
4. Shi,W., Zhu,C.Q., "An Implementation of Maximum Potential Parallelism Analyzing Method", J. Computer Engineering, Vol 18, No 2,pp9-15 1996.
5. Stanford Compiler Group,The SUIF Parallelizing Compiler Guide,http://suif.stanford.edu,1993.
6. Wang,Q., Yu,Y., D'Hollander,E.H., “Visualizing the Iteration Space in PEFPT”, in Proc. High Perf. Computing and Network Intl. Conf. and Exhibition,Viena, Austria, pp908-915, Apr. 1997.
7. Wang,Q., Yu,Y., D'Hollander,E.H., “Interactive Programming using PEFPT”, Proc. Seminar Parallel Computing: Software,Architecture and Operating Systems, pp123-129,Oct. 1996.
8. Yu,Y.,ParaPIE: The Parallel Programming Interactive Environment, PhD thesis, Fudan Univ.,May 1998.
9. Yu,Y., Wang,Q., Zhu,C.Q. The Design of a Parallel Programming Environment for FPT, Proc. Topic in Knowledge and Information Technology, pp69-80 Gent, Bel., Sept.1996.
10. Yu,Y., Wang,Q., Zang,B., Shi,W., Zhu,C.Q., D'Hollander,E.H., “Interactively Studying the Unimodular Loop Parallelizing Transformations using PEFPT Programming Environment”, Int'l Seminar of Parallel Processing, Gent, Bel., Jan 1998.
11. Zhang,F.B., “The FPT Programming Environment”, PhD Thesis, University of Gent, Bel.,1996
12. Zhu,C.Q., Zang,B., Chen,T., "An Automatic Parallelizer ".Journal of Software(China), Mar 1996, Vol.7, No.3, pp180-186.
13. SUN Microsystem Corp., Java API Document,1996.