

CALEBE DE PAULA BIANCHINI

UM AMBIENTE PARA PROGRAMAÇÃO ORIENTADA A OBJETOS
DISTRIBUÍDOS E PARALELOS EM GRADES COMPUTACIONAIS

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do título de Doutor em Engenharia
Elétrica.

São Paulo
2009

CALEBE DE PAULA BIANCHINI

UM AMBIENTE PARA PROGRAMAÇÃO ORIENTADA A OBJETOS
DISTRIBUÍDOS E PARALELOS EM GRADES COMPUTACIONAIS

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do título de Doutor em Engenharia
Elétrica.

Área de Concentração:
Sistemas Digitais

Orientador: Profa. Livre-Docente Liria
Matsumoto Sato

São Paulo
2009

AGRADECIMENTOS

Agradeço a Deus pelo milagre da minha vida, que está completa em Cristo Jesus: “Porque Dele, por Ele e para Ele são todas as coisas”. *Soli Deo gloria*.

Agradeço a minha orientadora, Profa Dra. Liria Matsumoto Sato, pelo apoio durante esses anos de estudos e pelo incentivo nos momentos de fraqueza. E, em especial, pela ajuda incondicional nos últimos dias de escrita deste trabalho.

Em especial, quero agradecer à minha amada esposa Grazielle de Arruda Werneck Bianchini que, durante boa parte do desenvolvimento deste trabalho, se mostrou prestativa em ouvir meus problemas técnicos, me consolar quando os defeitos apareciam, e me incentivar nos estudos, mesmo não conhecendo profundamente o tema deste trabalho.

Agradeço também minha família (Syrthes, Abiall, Lorise e Cinira), pelo suporte emocional durante os primeiros anos de meus estudos. Agradeço também àqueles que se juntaram a essa família tão querida (Atacy e Louise) e fizeram diferença nesse percurso.

Quero agradecer também aos meus dois irmãos, Francisco Isidro Massetto e Augusto Mendes Gomes Jr, que dividiram comigo momentos de alegria e angústia, mostrando o verdadeiro valor de uma amizade.

Agradeço também ao LAHPC pela infra-estrutura disponível e aos meus companheiros de laboratório, em especial: Nilton César, Fernando Kakugawa, Mathias, Charles, Darlon, Artur, Denis e Jean.

Agradeço também ao Francisco Ribacionka pelo apoio técnico e ao LCCA/USP – Laboratório de Computação Científica Avançada – pela infra-estrutura disponibilizada.

Não posso deixar de agradecer também meu antigo orientador de mestrado, Prof. Dr. Antonio Francisco do Prado, por me abrir os caminhos da pesquisa e do trabalho científico, permitindo que eu chegasse até esse ponto.

E, por fim, agradeço a todos os quais não pude mencionar neste trabalho por falta de espaço, mas que também estiveram comigo durante essa caminhada.

Meu sincero MUITO OBRIGADO!

“Que proveito tem o homem de todo o seu trabalho, com que se afadiga debaixo do sol?... Então vi que a sabedoria é mais proveitosa do que a estultícia, quanto a luz traz mais proveito do que as trevas... Ah! Morre o sábio, e da mesma sorte, o estulto! Pelo que aborreci a vida, pois me foi penosa a obra que se faz debaixo do sol; sim, tudo é vaidade e correr atrás do vento. Também aborreci todo o meu trabalho, com que me afadiguei debaixo do sol... Porque há homem cujo trabalho é feito com sabedoria, ciência e destreza; contudo, deixará o seu ganho como porção para aquele quem por ele não trabalhou;... Porque todos os seus dias são dores, e seu trabalho, desgosto; até de noite não descansa o seu coração; também isso é vaidade... Porque Deus dá sabedoria, conhecimento e prazer ao homem que lhe agrada... De tudo o que se tem ouvido, a suma é: teme a Deus e guarda os seus mandamentos; porque isso é o dever de todo homem. Porque Deus há de trazer a juízo todas as obras, até as que estão escondidas, quer sejam boas, quer sejam más.”

Livro do Eclesiastes

RESUMO

Grades Computacionais (*computational grid*) já é uma realidade tanto no meio acadêmico quanto no meio empresarial. Seu uso se tornou popular principalmente devido à divulgação dos trabalhos nesta área e pela propaganda de produtos e *softwares* que oferecem essa idéia. Apesar disso, ambientes para o desenvolvimento de aplicações orientadas a objetos em Java para uma infraestrutura de *grid* ainda é escasso. Algumas iniciativas oferecem bibliotecas para este desenvolvimento. Outras utilizam paradigmas diferentes, como o de passagem de mensagem, para o desenvolvimento de aplicações. Além disso, a própria infraestrutura de *grid*, formada por diferentes domínios administrativos com diferentes políticas de segurança e uso, impede que as aplicações sejam executadas nos diversos níveis existentes no *grid*. Estes níveis, formados por computadores e *clusters* de computadores com nós de execução, possuem endereçamento privado, impossibilitando que as aplicações alocadas em cada um desses computadores/nós, em diferentes domínios e diferentes endereços, se comuniquem de forma transparente. Visando uma solução para esses problemas, esta tese apresenta um ambiente para programação orientada a objetos distribuídos e paralelos, em Java, denominado J4GE. Nesse ambiente, o modelo orientado a objetos é base para a distribuição das classes, métodos e atributos existente em uma aplicação. Além disso, o ambiente oferece transparência no acesso aos objetos espalhados pelo *grid* através de um Serviço de Mensagem, independente do nível onde o recurso, computador ou nó, se encontra. Essa transparência permite também que o programador utilize a plataforma Java sem a necessidade de aprender ou conhecer novas bibliotecas ou paradigmas, diminuindo o esforço no desenvolvimento de aplicações para *grid*. E, juntamente com os recursos da plataforma Java e do ambiente J4GE, é possível criar objetos distribuídos com comportamento paralelo e concorrente, trazendo maior eficiência para a execução da aplicação.

Palavras-chave: Grades computacionais. Sistemas distribuídos. Ambiente de programação. Java.

ABSTRACT

Computing grid is already a reality both in academic and business world. Its use has become popular mainly because of the projects in this area and the advertising of products and software that offer this idea. Nevertheless, environments for development of object-oriented applications in Java for grid infrastructure are still scarce. Some initiatives offer libraries for this development. Others use different paradigms such as the message-passing for development of applications. Moreover, the infrastructure of grid, formed by different administrative domain with different security policies, prevents the execution of applications at various levels in the grid. These levels, formed by computers and clusters of computers with execution nodes, have private addresses, make impossible the transparent communication of the applications allocated in each of these computers at different levels in different domains. Focused on these problems, this thesis presents an environment for distributed and parallel object-oriented programming in Java, called J4GE. In this environment, the object-oriented model is the basis for the distribution of classes, methods and attributes in an existing application. Moreover, the environment offers transparency in objects access around the grid through a Message Service, regardless the level where is the resource, or the computer, or the execution node. This transparency also allows the programmer to use the Java platform without knowing or learning new libraries or paradigms, reducing the effort in developing applications for grid. The resources of the Java platform and the environment J4GE together can create distributed objects with parallel and concurrent behavior, bringing greater efficiency to the application.

Keywords: Computational Grid. Distributed systems. Programming environment. Java.

LISTA DE ILUSTRAÇÕES

Figura 1 – Especificação da classe para Produtor/Consumidor, com semáforo e monitor, em Java [23].	19
Figura 2 – Visão geral do ciclo de vida de compilação e execução de um programa Java [26].	20
Figura 3 – Classificação das Arquiteturas de <i>Grids</i> [58].	31
Figura 4 – Infra-estrutura integrada de Grid [1].	32
Figura 5 – Arquitetura Javelin [66].	34
Figura 6 – Arquitetura SUMA/G [67].	36
Figura 7 – Comunicação em grupo no ProActive [12].	38
Figura 8 – Arquitetura de componentes do <i>Globus Toolkit</i> [6].	39
Figura 9 – Infra-estrutura simples de computadores em um <i>grid</i> .	44
Figura 10 – Infra-estrutura de <i>grid</i> com <i>cluster</i> de computadores.	46
Figura 11 – Sintaxe única para anotação de código-fonte, no J4GE.	48
Figura 12 – Classe anotada para execução paralela através do ambiente J4GE.	49
Figura 13 – Aplicação utilizando objetos distribuídos e paralelos pelo <i>grid</i> .	50
Figura 14 – Arquitetura J4GE em um <i>grid</i> .	51
Figura 15 – Arquitetura do Serviço de Mensagem.	52
Figura 16 – Caminho de uma mensagem por uma infra-estrutura de <i>grid</i> através do Serviço de Mensagem.	53
Figura 17 – Diagrama de seqüência para uso de um recurso do Serviço de Mensagem [90].	55
Figura 18 – Especificação do protocolo do Serviço de Mensagem em WSDL.	56
Figura 19 – Diagramas de classes com a transformação de uma classe anotada.	60
Figura 20 – Diagrama de classes da transformação de uma referência de objeto usado em uma classe anotada.	60
Figura 21 – Código-fonte com anotação em classe para o ambiente J4GE.	61
Figura 22 – Diagramas de classes com a solução de um método anotado.	62
Figura 23 – Código-fonte com anotação em método para o ambiente J4GE.	63
Figura 24 – Diagramas de classes com a solução para anotação de atributo.	63
Figura 25 – Código-fonte para anotação em atributo para o ambiente J4GE.	64
Figura 26 – Infra-estrutura de <i>grid</i> para validação do ambiente J4GE.	67

Figura 27 – Diagrama de classes com dependência simples entre objetos.....	69
Figura 28 – Diagrama de classes com dependência entre duas classes.....	71
Figura 29 – Diagrama de classes com dependência entre duas classes remotas....	71
Figura 30 – Diagrama de classes com generalização de objeto personalizado.....	73
Figura 31 – Diagrama de classes com especialização da classe <i>Thread</i>	74
Figura 32 – Diagrama de classes para associação simples.....	76
Figura 33 – Diagrama de classes para associação entre classes remotas.....	77
Figura 34 – Diagrama de classes para teste da estratégia de método remoto.	78
Figura 35 – Diagrama de classes para teste da estratégia de atributo remoto.	79
Figura 36 – Diagrama de classes especificando a criação de um objeto- <i>thread</i>	81
Figura 37 – Diagrama de classes com a especificação da aplicação TSP.	84

LISTA DE TABELAS

Tabela 1 – Resumo das características e classificação das DJVMs.....	29
Tabela 2 – Resumo das características dos ambientes orientados a objetos em <i>grid computing</i>	41
Tabela 3 – Descrição dos recursos disponíveis em cada Domínio administrativo para a infra-estrutura de <i>grid</i> usada na validação do J4GE.....	67
Tabela 4 – Entradas e saídas para teste da dependência simples.	70
Tabela 5 – Entradas e saídas para teste da dependência entre duas classes.	72
Tabela 6 – Entradas e saídas para teste de herança em classes personalizadas....	73
Tabela 7 – Entradas e saídas para teste de especialização da classe <i>Thread</i>	75
Tabela 8 – Entradas e saídas para teste da associação simples entre classes.....	76
Tabela 9 – Entradas e saídas para teste da relação de associação entre duas classes remotas.....	77
Tabela 10 – Entradas e saídas para teste da estratégia de método remoto.	79
Tabela 11 – Entradas e saídas para teste da estratégia de atributo remoto.	80
Tabela 12 – Entradas e saídas para teste de objeto- <i>thread</i> remoto.....	81
Tabela 13 – Tempo de execução (segundos) da procura por números primos, tanto em uma máquina local quanto no J4GE.....	82
Tabela 14 – Tempo de execução (segundos) de diversas ações do J4GE.	83
Tabela 15 – Tempo de execução (segundos) da aplicação TSP.	85
Tabela 16 – Casos de testes para a aplicação TSP.....	86
Tabela 17 – Tabela comparativa entre o J4GE e os demais trabalhos.	87

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CoG	Commodity Grid
DJVM	Distributed Java Virtual Machine
DSM	Distributed Shared Memory
HPC	High Performance Computing
J4GE	Java for Grid Environment
JaDiMa	Java Distributed Machine
JAVAPD	International Workshop on Java and Components for Parallelism, Distribution and Concurrency
JDK	Java Development Kit
JESSICA2	Java-Enabled Single-System Image Computing Architecture
JIT	Just-in-time compiler
JVM	Java Virtual Machine
LRM	Local Resource Management
MPI	Message Passing Interface
MPP	Massively Parallel Processing
NOW	Network of Workstation
OGSA	Open Grid Services Architecture
OGSI	Open Grid Services Infrastructure
OOPSLA	International Conference on Object-Oriented Programming, Systems, Languages and Applications
P2P	Peer to Peer
RMI	Remote Method Invocation
SBLP	Simpósio Brasileiro de Linguagens de Programação
SMP	Symmetric MultiProcessors
SSI	Single-System Image
TSP	Taveling Salesman Problem
WSRF	WebService Resource Framework

Gracias por visitar este Libro Electrónico

Puedes leer la versión completa de este libro electrónico en diferentes formatos:

- HTML(Gratis / Disponible a todos los usuarios)
- PDF / TXT(Disponible a miembros V.I.P. Los miembros con una membresía básica pueden acceder hasta 5 libros electrónicos en formato PDF/TXT durante el mes.)
- Epub y Mobipocket (Exclusivos para miembros V.I.P.)

Para descargar este libro completo, tan solo seleccione el formato deseado, abajo:

