5th ACM/IFIP/USENIX International Middleware Conference

Middleware 2004

October 18th-22nd, 2004

Raphael Y. de Camargo

Departamento de Ciência da Computação

Universidade de São Paulo

Sumário

- Toronto e Centro de Conferências
- Middleware Conference
  - Workshops
  - Keynotes
  - Technical Papers
- Comentários

Toronto, Canadá

- Centro financeiro do Canadá
- Fica às margens do Ontario Lake
- Principal atração é a CN Tower, a construção mais alta do mundo, com 553m de altura
- Conferência realizada no Hotel Renaissence, localizado junto ao Estádio SkyDome

Toronto

Centro de Toronto

SkyDome

Brasileiros no Middleware

Middleware 2004

- Composta por:
  - Workshops (4 workshops)
  - Keynotes (3 keynotes)
  - Technical Papers (25 trabalhos aceitos)
    - Taxa de aceitação 12.8%, 25 de 194
  - Work in Progress (6 WiPs)

Workshops

- *2nd Workshop on Middleware for Grid Computing*
- *2nd Workshop on Middleware for Pervasive and Ad-Hoc Computing*
- *3rd Workshop on Adaptive and Reflective Middleware*
- *1st Middleware Doctoral Symposium*

WS1: Middleware for Grid Computing

- 48 submissões (15 trabalhos aceitos)
- Apresentei o artigo:

  Checkpointing-based Rollback Recovery for Parallel Applications on the InteGrade Grid Middleware

  R. Camargo, A. Goldchleger, F. Kon, and A. Goldman

- Artigos em Grid Management, Scheduling, Security, Data Grids, Tools, etc.

Technical Papers

- Publish / Subscribe
- Peer-to-Peer Computing
- Routing Protocols and Overlays
- Middleware for Replication and Transactions
- Web Services: Composition, Integration and Interoperability
- Middleware for Mobility
- Application Server, Enterprise Computing, and Software Engineering

The Peer Sampling Service: Experimental Evaluation of Unstructured Gossip-Based Implementations

Mark Jelasity (University of Bologna, Italy), Rachid Guerraoui (EPFL, Switzerland), Anne-Marie Kermarrec (INRIA, France), Maarten van Steen (Vrije Universiteit, The Netherlands)

- Popularização de modelos de comunicação do tipo ***Gossip***
  - Possuem aplicação em áreas como disseminação de informação, agregação, bolanço de carga, sincronização, etc.
  - Propriedade comum é que periodicamente nós da rede trocam informação com outros nós.

Peer Sampling Service: Experimental Evaluation of Unstructured Gossip-Based Implementations

- Modelos de comunicação do tipo ***Gossip***
  - Requerem um serviço de ***Peer Sampling***: Fornece aos nós outros nós com quem eles possam conversar.
  - Estudos analíticos relacionados a protocolos do tipo Gossip utilizam a suposição de que o serviço de Peer Sampling seleciona nós de acordo com uma distribuição uniforme.
  - Uma solução utilizada é que todos os nós do sistema sabem sobre todos os demais nós

Peer Sampling Service: Experimental Evaluation of Unstructured Gossip-Based Implementations

- Utilizar um protocolo de disseminação das associações baseado em Gossiping.
- Existem diversas variações para este protocolo.
  - Não existes estudos e comparações entre estes diferentes protocolos
  - Não é claro se alguma destas variantes gera uma distribuição uniforme de nós fornecidos.

Peer Sampling Service: Experimental Evaluation of Unstructured Gossip-Based Implementations

- Contribuições
  - Identificação do Peer Sampling Service
  - Protocolo genéricos para estes serviços
  - Apresentam uma metodologia experimental para comparar os diferentes protocolos
- Protocolo Genérico
  - Protocolos divididos em três dimensões:
    - Peer selection:
    - View propagation:
    - View selection:

Peer Sampling Service: Experimental Evaluation of Unstructured Gossip-Based Implementations

- Conclusões
  - Diferentes protocolos geram diferentes grafos de nós, mas nenhum deles gera grafos aleatórios tradicionais.
  - Eles geram um tipo de grafo chamado “small-world”, que possuem um diâmetro pequeno e grande aglomeração.
  - Diferentes protocolos geram grafos com diferentes propriedades: grau de conectividade, distância média e grau de aglomeração.

Dynamically Programmable and Reconfigurable Middleware Services

Manuel Roman, Nayeem Islam - DoCoMo Labs, USA

- Aparelhos celulares
  - Aumento na complexidade → Estrutura de middleware para facilitar o desenvolvimento de software.
  - Ex: RPC, discovery, segurança, QoS, Eventos, etc
- Requisitos:
  - Configurability
  - Updateability
  - Upgradeability

Dynamically Programmable and Reconfigurable Middleware Services

- Externalização da Arquitetura
  - ***Estrutura:*** componentes que compoem o sistema
  - ***Logica:*** as regras de interação entre os componentes
  - ***Estado:*** estado dos componentes do sistema
- Vantagens
  - Serviços podem ser construídos em tempo de execução utilizando os descritores de arquitetura.
  - Abilidade de virtualizar a arquitetura do sistema e criar snapshots do estado sistema.

Dynamically Programmable and Reconfigurable Middleware Services

- Dynamically Programmable and Reconfigurable Software (DPRS)
- Three abstractions
  - Micro-building Block (MBB):
    - Parâmetros Entrada → Ação → Parâmetros Saída
    - Estado é armazenado em tuplas (nome, valor)
  - Action:
    - Defines the MBB execution order (Lógica do sistema)
    - Interpretadas ou compiladas
  - Domain:
    - Agrega coleções de MBBs relacionados

Dynamically Programmable and Reconfigurable Middleware Services

- ExORB
  - Broker que permite a invocação e recebimentos de chamadas RPC utilizando protocolos como IIOP e XMLRPC
- Experimentos
  - Versão inicial: 50% da versão estática
  - Versão otimizada: 75% da versão estática
  - Overhead ações interpretadas x compiladas é pequeno (< 5%)

Portable and Efficient Distributed Threads for Java

Eli Tilevich, Yannis Smaragdakis - GeorgiaTech, USA

- Java RMI e CORBA não suportam coordenação entre threads em ambientes distribuídos
  - Operações de sincronização como `wait` e `synchronized` não são propagadas através da rede.
  - A identidade da thread não é mantida através da rede.

Portable and Efficient Distributed Threads for Java

- Soluções Existentes
  - Substituir o middleware existente por um que seja que permita multithreading através da rede
  - Modificar as chamadas de métodos na aplicação cliente de modo que um parâmetro extra contendo informações sobre a thread atual seja propagado

Portable and Efficient Distributed Threads for Java

- Realizam transformação de bytecode
  - Biblioteca que contém versões distribution-aware das operações de sincronização
  - Stubs gerados pelo Java RMI são modificados de modo que as chamadas de métodos RMI propagam a identidade da Thread
- J-Orchestra
  - Sistema que, com a supervisão do usuário, divide um programa Java em partes que são executadas em diferentes máquinas

Keynotes

- How Wrong Can You Be? Getting Lost on the Road to Massive Scalability
  - ***Werner Vogels***, Director of Systems Research, Amazon
- Experiences Building a 24x7 Real-time ASP Service at Citrix Online
  - ***Thornsten von Eicken***, Chief Architect, Citrix Online
- Aspect-Oriented Programming – The Promisse and the Controversy
  - ***Gregor Kiczales***, University of British Columbia

How Wrong Can You Be?
Getting Lost on the Road to Massive Scalability

- Discutiu sobre sistemas que utilizam dezenas de milhares de máquinas
- Utilizando uma taxa realista de falhas, pode-se ver que a cada hora algumas dezenas de máquinas irão falhar
- Devemos repensar nossas suposições sobre escalabilidade
- Deve-se programar pensando nas falhas
- Amazon migrou para Web Services.

Experiences Building a 24x7 Real-time ASP Service at Citrix Online

- ***Citrix Online:*** runs 24x7 real-time services for remote access (GoToMyPC), help desk (GoToAssist), and collaboration (GoToMeeting)
- ***Servidor Central:*** máquina cliente e o Usuário se comunicam através dele.
- ***Logging:*** é bastante importante para poder determinar onde os problemas ocorreram.
- ***Escalabilidade:*** A cada ano, um novo servidor mais rápido é comprado.

Aspect-Oriented Programming: The Promisse and the Controversy

- ***Aspect-oriented programming (AOP):*** objetivo é modularizar “crosscutting concerns”
- ***Controvérsia em AOP:*** Ela vai contra os atuais conceitos relativos à modularização de código.
- Exemplos onde AOP é útil
  - Logging e Observer Pattern
- AspectJ plug-in para o Eclipse
  - Mostra claramente os pontos no software onde os aspectos serão inseridos

Comentários

- Ótima experiência
  - Presença dos principais pesquisadores na área de Middleware
  - Permite ter uma visão sobre pesquisas sendo feitas na área de Middleware
- Boa qualidade dos trabalhos apresentados
  - Em função da alta competitividade
  - Taxa de aceitação de quase 1 para 8

Comentários

- Vale a pena tentar submeter artigos
  - Além dos *Technical Papers*, existem os *Workshops* e a sessão *Work in Progress*
  - *Student Travel Grants*: cobrem a maioria dos gastos com a viagem
- Próximo Middleware será em Grenoble, França
  - Deadline para Technical Papers: *30 de março*

- *Perguntas?*
- *Comentários!*