

# EXEHDA-CC x JADA: Uma análise de espaço de objetos compartilhados em Java

Dario Fernandes Franz<sup>1</sup>, Marcelo Augusto Cardozo Junior<sup>3</sup>  
Adenauer Corrêa Yamin<sup>1</sup>, Jorge Luis Victória Barbosa<sup>1</sup>,  
Cláudio Fernando Resin Geyer<sup>2</sup>

<sup>1</sup>Universidade Católica de Pelotas (UCPel)  
{franz,adenauer,barbosa}@ucpel.tche.br

<sup>2</sup>Universidade Federal do Rio Grande do Sul (UFRGS)  
geyer@inf.ufrgs.br

<sup>3</sup>Universidade Federal de Pelotas (UFPel)  
marcelocardozo@uol.com.br

## Introdução

O presente trabalho tem como objetivo comparar dois sistemas em JAVA que dão suporte a memória compartilhada em um ambiente distribuído: o JADA e o EXEHDA-CC.

O JADA [ROS 2002], criado por Davide Rossi, possui o seu código fechado, porém, é de livre distribuição. O JADA foi empregado durante algum tempo pelo Grupo de Pesquisa em Processamento Paralelo e Distribuído (G3PD) da Universidade Católica de Pelotas (UCPel), porém além de não atender todas as funcionalidades necessárias aos projetos em andamento, possui o código fechado. Essas foram as duas motivações gerais para a criação da ferramenta chamada EXEHDA-CC.

O EXEHDA-CC [CAR 2002] possui as mesmas características gerais do JADA, porém apresenta a prerrogativa de estar direcionado para os interesses das atividades em andamento no contexto do EXEHDA [YAM 2002].

## Espaço de Objetos

A inexistência de recursos globalmente compartilhados faz com que as arquiteturas distribuídas sejam intrinsecamente escaláveis. Entretanto, a necessidade de um espaço compartilhado é difícil de ser prescindido por várias razões, entre elas: (i) modelos de computação globais são potencialmente mais simples que os modelos locais; (ii) aplicações locais também necessitam de recursos compartilhados, por exemplo, para criar uma visão única de nomes. Os problemas mencionados acima podem ser resolvidos através de um espaço de objetos.

Um *espaço de objetos* é essencialmente uma abstração de uma memória compartilhada, garantindo que vários processos possam acessar o espaço de maneira anônima e assíncrona, sem introduzir os custos inerentes a memória distribuída.

O modelo LINDA [LIN 2002] propõe o modelo genérico do espaço de objetos, sendo um modelo de programação concorrente desenvolvido na Universidade de Yale

que tem como tema central ser uma alternativa aos dois tradicionais métodos de processamento paralelo: o de memória compartilhada e de troca de mensagens.

Do ponto de vista do programador LINDA é uma extensão da linguagem de programação. O que o modelo LINDA prevê são basicamente mecanismos para a comunicação entre processos.

O modelo LINDA tem quatro primitivas básicas:

- rd(t): faz a leitura não destrutiva de uma tupla “t” no espaço;
- out(t): insere uma tupla “t” no espaço;
- in(t): faz a leitura destrutiva de uma tupla “t” no espaço;
- eval(expressão): cria uma tupla nova com os campos de expressão dinamicamente calculados por processos independentes.

## JADA

O JADA [ROS 2002] é um pacote para Java, que possibilita aplicativos utilizarem múltiplos espaços de objetos compartilhados, sejam eles locais ou remotos, com objetivo de coordenação e compartilhamento de dados.

O JADA, baseado no modelo Linda, foi criado como parte integrante do projeto *PageSpace*, com o principal intuito de prover um sólido espaço de objetos. Por ser baseado em um conjunto de classes, permite que vários usuários mantenham suas ferramentas de desenvolvimento no padrão Java.

## EXEHDA-CC

O EXEHDA [YAM 2002] é uma proposta para tratamento da adaptação dinâmica da execução em sistemas distribuídos com suporte à mobilidade de hardware e de software. Para tanto, o EXEHDA é compatível com contextos de elevada heterogeneidade e escalabilidade e emprega uma estratégia multinível colaborativa, na qual tanto o sistema como as aplicações participam das decisões de adaptação. A sua linguagem alvo de programação é o Holoparadigma.

O Holoparadigma [BAR 2002] contempla de forma intrínseca questões de mobilidade e distribuição, provendo suporte a adaptação no tocante a linguagem de programação. O EXEHDA, por sua vez, viabiliza um comportamento reativo e ativo na gerência das entidades de modelagem do Holoparadigma, em um *middleware* com comportamento reflexivo, e que suporta cooperação com as definições em tempo de compilação no momento de decidir as ações adaptativas. As ações adaptativas têm como núcleo um mecanismo de escalonamento totalmente integrado ao controle da execução distribuída do Holoparadigma.

O EXEHDA-CC [CAR 2002] foi desenvolvido com a preocupação que fossem contemplados objetivos específicos, necessários pela arquitetura do EXEHDA, tais como: (1) integrável, como parte de um sistema, o módulo EXEHDA-CC tem que ser completamente integrável no restante, assim como deve ser compatível com o software já existente da arquitetura EXEHDA; (2) portátil, devido à proposta do EXEHDA ser um ambiente de execução portátil, assim devem ser todos os seus módulos e serviços; (3) suporte a múltiplos espaços de objetos; (4) transparência, uma das principais características do EXEHDA é ser transparente ao programador; (5) escalabilidade, o

EXEHDA-CC deve ser escalável para suportar o crescimento do número de nodos no sistema.

## Resultados obtidos

Antes de ter sido desenvolvido o EXEHDA-CC, o JADA foi utilizado pelo EXEHADA para fazer as simulações de sincronismo entre processos. Uma meta perseguida nesse trabalho foi comparar o EXEHDA-CC com a ferramenta então tida como a melhor para os interesses do grupo.

Os testes foram realizados nas máquinas do G3PD, as quais têm a seguinte configuração: Processador Intel Pentium® II 266 MHz; 64 Megabytes de memória RAM; Sistema Operacional *Linux Debian Woody*; Java 2 SDK da *Sun Microsystems* versão 1.4.0.

### Teste 1: Sincronismo de processos distribuídos

Esse teste mostra o desempenho do espaço de objetos na função de manter sincronismo entre dois processos distribuídos.

O teste foi feito de maneira que o primeiro envia um sinal de sincronismo e aguarda a resposta do outro processo, o segundo processo recebe os sinais enviados pelo primeiro e então devolve uma resposta. Deste modo são manipulados pelo espaço de objetos dois objetos a cada ciclo de repetição. O resultado deste teste está na Figura 1.

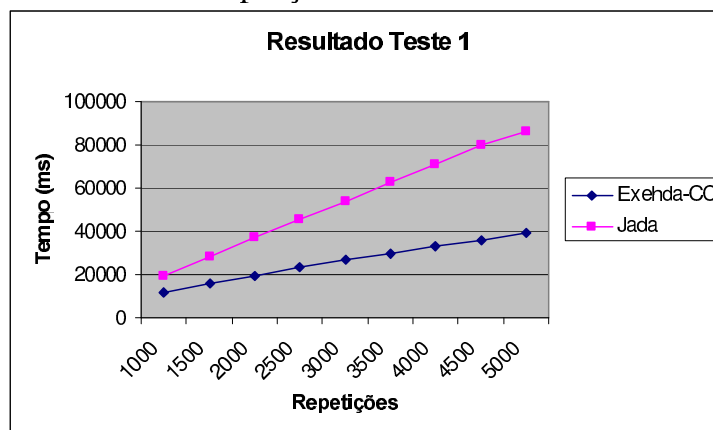


Figura 1 – Sincronismo de Processos Distribuídos

Os dados apontam que o EXEHDA-CC é mais rápido que o JADA quando se trata de enviar sinais de sincronismo entre processos. Também percebe-se que ambos têm um crescimento linear a medida que mais sinais são enviados, com o EXEHDA-CC possuindo um grau de crescimento menor que do JADA. Isso faz com que o EXEHDA-CC atenda melhor o EXEHDA.

### Teste 2: Comunicação entre processos

Esse teste tem o intuito de mostrar o desempenho do espaço quando fazendo o papel de meio de comunicação entre processos. O teste se deu com um processo A criando uma mensagem de 512 bytes, enviando esta mensagem para o processo B via espaço de objetos e aguardando uma confirmação de retirada de mensagem do espaço.

Somente após o recebimento dessa mensagem é que o próximo ciclo na repetição se daria. A figura 2 contém a variação gráfica dos resultados obtidos.

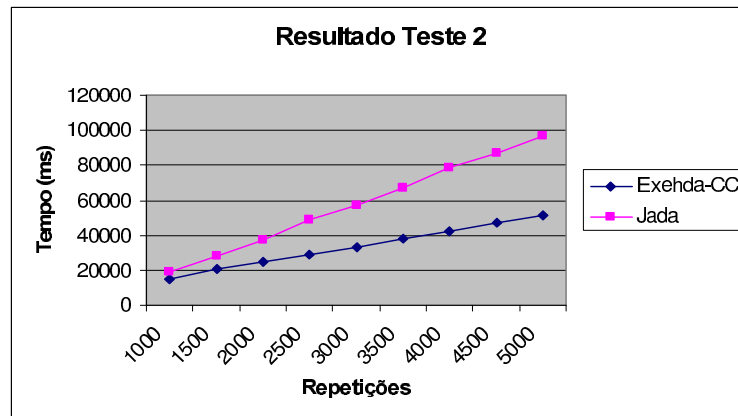


Figura 2 – Comunicação entre Processos

Quando se analisou a comunicação entre processos, ambos os espaços mostraram crescimento linear, com o EXEHDA-CC tendo um crescimento menor que o JADA. Isso se mostra fundamental, já que o EXEHDA-CC também é utilizado pelo EXEHDA como meio de comunicação entre vários processos.

## Considerações finais

O pacote do EXEHDA-CC está sendo distribuído e utilizado na realização de testes e desenvolvimento de aplicações distribuídas. A próxima etapa consiste em realizar outros *benchmarks*, já com o EXEHDA-CC integrado com os outros módulos do EXEHDA.

## Referências

- [BAR 2002] BARBOSA, Jorge L. V. **Holoparadigma: Um Modelo Multiparadigma Orientado ao Desenvolvimento de Software Distribuído**. 2002. 215p. Tese (Doutorado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [CAR 2002] CARDOZO, Marcelo A. J., **Uma Contribuição à Comunicação e a Coordenação no EXEHDA**. 2002, 61pg. Trabalho de Projeto de Graduação em Ciência da Computação, Universidade Federal de Pelotas (UFPel).
- [LIN 2002] Yale University, Department of Computer Science. **Yale Linda Group**, Disponível em <<http://www.cs.yale.edu/Linda/linda.html>>. Acesso em : Novembro de 2002.
- [ROS 2002] ROSSI, Davide. **JADA**, Disponível em: <<http://www.cs.unibo.it/~rossi/JADA/>>. Acesso em: Outubro 2002.
- [YAM 2002] YAMIN, Adenauer; BARBOSA, Jorge; GEYER, Cláudio; AUGUSTIN, Iara; CAVALHEIRO, Gerson; SILVA, Luciano; REAL, Rodrigo; **A Framework for Exploiting Adaptation in High Heterogenous Distributed Processing**. SBAC-PAD 2002, 14<sup>th</sup> Symposium on Computer Architecture and High Performance Computing, pg 125-132, October 2002.