

EXEHDA-TS: Um modelo para coordenação de aplicações na Computação Pervasiva

Vilnei Marins de Freitas das Neves, Rodrigo Moura, Adenauer Corrêa Yamin,
Rodrigo Santos de Souza

Centro Politécnico – Universidade Católica de Pelotas (UCPEL) – Pelotas – RS – Brazil

{vilnei, rodrigomoura, adenauer, rsouza}@ucpel.tche.br

Resumo. *Este trabalho tem por objetivo central colaborar com a modelagem e prototipação do EXEHDA-TS, que consiste em um mecanismo para a coordenação de aplicações na Computação Pervasiva.*

1. Introdução

Este trabalho está inserido nos esforços de pesquisa do Grupo de Pesquisa em Processamento Paralelo e Distribuído (G3PD/CPOLI/UCPEL) em Computação Pervasiva, a qual prevê a disponibilização do ambiente computacional do usuário independente de tempo e localização. Nessa perspectiva, as soluções computacionais são largamente distribuídas e caracterizam-se por manter uma operação integrada com o mundo real, provendo uma elevada transparência com a infraestrutura e com foco no usuário. Diante desse cenário complexo, é necessário um *middleware* que forneça suporte ao desenvolvimento e execução das aplicações, abstraindo aspectos da infraestrutura. Baseando-se nessas premissas, concebeu-se o EXEHDA (*Execution Environment for Highly Distributed Applications*), um *middleware* adaptativo ao contexto e baseado em serviços, que visa criar e gerenciar um ambiente pervasivo, bem como promover a execução, sob este ambiente, das aplicações que expressam a semântica siga-me. Estas aplicações são distribuídas, móveis e adaptativas ao contexto em que seu processamento ocorre, estando disponíveis a partir de qualquer lugar, todo o tempo (LOPES, 2007).

Para promover a cooperação entre processos em cenários de elevada distribuição, uma estratégia adotada é a utilização de modelos baseados em coordenação, que caracterizam-se por separar a computação da coordenação (TANENBAUM; STEEN, 2007). Um dos modelos de coordenação mais difundidos é o de Espaços de Tuplas, que consiste de uma memória associativa independente e compartilhada entre todos os nós do sistema (YAMIN, 2004). Esta memória pode ser considerada como uma espécie de repositório para as estruturas denominadas tuplas, permitindo que a comunicação entre processos ocorra através da inserção e da leitura das tuplas neste espaço compartilhado.

Os modelos de coordenação tradicionais apresentam limitações de utilizações considerando as necessidades da Computação Pervasiva (MAMEI; ZAMBONELLI, 2006a). No Modelo de Coordenação Direta, por exemplo, os processos distribuídos coordenam suas atividades com os demais através da comunicação direta e explícita. Assim, para que ocorra a comunicação entre os processos, esses devem conhecer-se antecipadamente através do uso, por exemplo, do endereço de rede e da porta dos parceiros de comunicação, caracterizando um acoplamento entre ambos, pois implica na coexistência no tempo e no conhecimento explícito das suas localidades. Por sua vez, no Modelo de Coordenação Baseada em Eventos, são utilizados mecanismos de publicação e subscrição (*publish/subscribe*) para promover a interação entre processos distribuídos

(MAMEI; ZAMBONELLI, 2006a), promovendo um desacoplamento referencial entre os processos, pois toda a comunicação é gerenciada pelo *middleware*. Porém, existe entre os participantes da coordenação um acoplamento temporal, o que significa que os envolvidos na comunicação precisam estar executando ao mesmo tempo para que possam ser notificados imediatamente após um evento ter sido publicado.

Dessa forma, para suprir as demandas por coordenação das aplicações alvo do *middleware* EXEHDA, que caracterizam-se pela distribuição, mobilidade e adaptação ao contexto, foi inserida a perspectiva de uso de um Espaço de Tuplas Distribuído com atuação proativa na coordenação de aplicações no ambiente pervasivo. Esse conceito serviu para modelagem do EXEHDA-TS, serviço disponibilizado para o *middleware* EXEHDA, possuindo como premissa gerenciar um Espaço de Tuplas de forma distribuída, com comportamento proativo, escalável e com características de desacoplamento referencial e temporal.

2. Modelagem do EXEHDA-TS

O EXEHDA-TS foi modelado como um serviço do *middleware* EXEHDA, fazendo parte do subsistema de comunicação do mesmo. No modelo de coordenação proposto pelo EXEHDA-TS, o armazenamento das tuplas distribui-se fisicamente entre os nodos do ambiente pervasivo. Para isso, os Espaços de Tuplas são fisicamente associados aos OX, de forma que cada OX tem seu próprio Espaço de Tuplas chamado TSox. O OX (*Objeto eXehda*) consiste na unidade computacional base das aplicações do *middleware* EXEHDA (YAMIN, 2004).

Os diversos TSox podem ser agrupados formando um ambiente virtual chamado TSvirt, que consiste em um conjunto de referências aos TSox distribuídos pelos nodos do sistema, servindo de base para operações realizadas pelos OX, fazendo que cada operação realizada por este atue diretamente sobre o TSvirt, abstraindo qualquer informação sobre a localização física das tuplas. A figura 1 demonstra a composição do Espaço de Tuplas segundo o modelo do EXEHDA-TS.

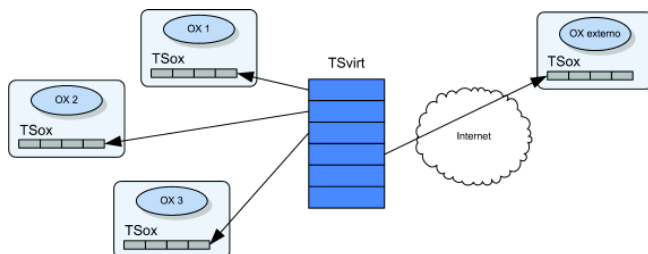


Figura 1. Composição do Espaço de Tuplas (SOUZA, 2009)

Como forma de diminuir os custo de comunicação entre o TSox e o TSvirt, o EXEHDA-TS disponibiliza dois modos de sincronização: permanente ou sob demanda. No modo sincronização permanente, uma vez que o TSox tenha sido integrado ao TSvirt, somente deixará de ser acessado nos momentos de desconexão, independentemente dos destinos dos OX decorrentes de processos de migração. Por sua vez, no modo de sincronização sob demanda, o TSox pode deixar de sincronizar com o TSvirt por determinados momentos a fim de reduzir os custos de comunicação.

O EXEHDA-TS ainda disponibiliza mecanismos que promovem a reatividade do Espaço de Tuplas, que são mapeados em duas operações básicas: *subscribe* e *find*. A

primeira refere-se à subscrição de eventos, mecanismo capaz de notificar os componentes de software sobre a existência de tuplas com informações relevantes imediatamente após elas terem sido inseridas no Espaço de Tuplas. A segunda refere-se a consulta pervasiva, mecanismo que realiza busca por informações no TSvirt através de operações repassadas através aos diversos dispositivos onde estão fisicamente localizados os TSox que compõem o mesmo. Inicialmente essas operações ocorrem em nível celular, sendo posteriormente repassados aos demais TSox distribuídos em outros escopos celulares. Basicamente o que diferencia a subscrição de eventos da consulta pervasiva é que a última é cancelada assim que a tupla desejada é encontrada, e caso isso não ocorra, seu cancelamento ocorre através do uso de um *timeout*, impedindo que a mesma seja mantida indefinidamente ativa. Na subscrição de eventos a operação deve ser explicitamente cancelada através de uma operação de *unsubscribe*.

3. Implementação do EXEHDA-TS

O protótipo atual do EXEHDA-TS possui uma implementação mínima de forma validar o modelo proposto. Atualmente implementação existente está sendo aprimorada e integrada ao *middleware* EXEHDA como um serviço a ser disponibilizado com a intenção de permitir a coordenação e a troca de informações de maneira pró-ativa entre componentes das aplicações em execução no ambiente pervasivo. Sendo assim, as tecnologias utilizadas na prototipação do EXEHDA-TS, possuem características semelhantes as utilizadas no desenvolvimento dos serviços do *middleware* EXEHDA. Esse aspecto motivou o uso da linguagem Java no desenvolvimento do serviço EXEHDA-TS, pois a mesma possui características oportunas para o desenvolvimento do EXEHDA-TS, as quais destacam-se: portabilidade, carga dinâmica de código, segurança, concorrência e sincronização, produtividade a desenvolvimento de software, e em especial, o suporte ao desenvolvimento de aplicações distribuídas através do uso de RMI.

Para construir o núcleo do EXEHDA-TS foi utilizada uma implementação minimalista e não distribuída do modelo de Espaço de Tuplas chamado LighTS. Além disso, ela é *opensource*, com código fonte disponibilizado em (PICCO, 2009) e desenvolvida em Java. O LighTS é utilizado como base para o núcleo do EXEHDA-TS, sendo o componente de armazenamento relacionado ao TSox e no qual são utilizadas algumas das operações disponíveis na sua API.

De forma a atender as funcionalidades no modelo proposto, as operações definidas para o modelo do EXEHDA-TS, sendo as mesmas mapeadas como métodos em três classes básicas (figura 2): *ETupleSpace*, *ETuple*, *EField*.

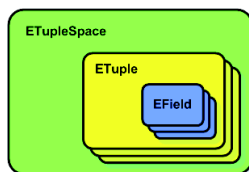


Figura 2. Organização das classes da API do EXEHDA-TS (SOUZA, 2009)

A classe *ETupleSpace* é responsável por implementar o Espaço de Tuplas no EXEHDA-TS, onde o construtor da classe cria o TSox permitindo que através dele possam ser realizadas todas as operações sobre o Espaço de Tuplas. Essencialmente, a criação do Espaço de Tuplas consiste na criação do TSox e em sua incorporação ao TSvirt da aplicação. No momento da criação do Espaço de Tupla, pode ser utilizada uma *flag* que indicará o modo de sincronização: permanente ou sob demanda, sendo

esse último o modo padrão. A alternância entre os modos de sincronização pode ser feita através operação *redefine*, na qual deve ser passado como parâmetro o modo desejado. Para inserção de tuplas, o EXEHDA-TS disponibiliza a operação *insert*, capaz de uma tuplas individualmente bem como um *array* de tuplas, dependendo do tipo de parâmetro de entrada. Já para a leitura e consumo das tuplas, são disponibilizadas as operações *read* e *consume* respectivamente, tendo essa última a característica de remover a tupla do Espaço de Tuplas ao realizar sua função. Essa duas operações possuem comportamento não bloqueante, retornando um elemento vazio caso não seja encontrada uma tupla que não satisfaça o padrão informado. Além dessa operações (inserção, leitura e consumo), o EXEHDA-TS provê operações que fornecem reatividade ao modelo através do uso das seguintes operações: *subscribe* e *unsubscribe*, que faz e remove a subscrição de um evento, respectivamente; a operação *find*, utilizada para implementar a consulta pervasiva; e a operação *adjustTime*, que está relacionada os gerenciamento de tempo das tentativas de acesso aos nodos.

As instâncias criadas a partir da classe *ETuple*, constituem as tuplas que são armazenadas no Espaço de Tuplas. Já os objetos instanciados a partir da classe *EField* consistem nos campos dos quais a tupla é composta.

4. Considerações Finais e Trabalhos Futuros

O EXEHDA-TS é um modelo de Espaço de Tuplas focado na coordenação de aplicações para Computação Pervasiva, com comportamento proativo, escalável e com características de desacoplamento referencial e temporal. Os trabalhos futuros baseiam-se na colaboração no processo de prototipação, desenvolvimento e avaliação de funcionalidades do EXEHDA-TS que serão desenvolvidas para aplicações relacionadas à área médica. O desenvolvimento será incremental seguindo o padrão adotado no G3PD consistindo de etapas de modelagem, implementação e validação. Nesse sentido, a identificação da necessidade de modificações em uma etapa específica, poderá provocar um retorno à etapa imediatamente anterior para necessárias providências.

5. Referências

- LOPES, J. L.; PILLA, M. L.; YAMIN, A. C. EXEHDA: a Middleware for Complex, Heterogeneous and Distributed Applications. Conferência Nacional em Inteligência Computacional Aplicada a Indústria de Petróleo, [S.l.], junho 2007.
- MAMEI, M.; ZAMBONELLI, F.; LEONARDI, L. Tuples On The Air: A Middleware for Context-Aware Computing in Dynamic Networks. In: ICDCS WORKSHOPS, 2003. Anais. . . IEEE Computer Society, 2003. p.342–347.
- PICCO, G. P. LighTS. <<http://lights.sourceforge.net/>>. Acesso em março de 2009.
- SOUZA, R. S. Uma Contribuição à Coordenação na Computação Pervasiva com Aplicações na Área Médica. 2009. Dissertação (Mestrado em Informática) - Universidade Católica de Pelotas, . Orientador: Adenauer Correa Yamin.
- TANENBAUM, A. S.; STEEN, M. V. Distributed Systems: Principles and Paradigms. 2.ed. [S.l.]: Prentice Hall, 2007.
- YAMIN, A. (2004). “Arquitetura para um Ambiente de Grade Computacional Direcionado as Aplicações Distribuídas Móveis e Conscientes do Contexto da Computação Pervasiva”. Tese de Doutorado, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil.