

# SACH: Uma Proposta Distribuída para Reconfiguração de Dispositivos na Computação Pervasiva

Eduardo da Silva Möller<sup>1</sup>, Wagner G. Al-Alam<sup>1</sup>,  
Maurício Lima Pilla<sup>1</sup>, Adenauer Corrêa Yamin<sup>1</sup>

<sup>1</sup>Universidade Católica de Pelotas  
Centro Politécnico  
Caixa Postal 402 – 96010-000 – Pelotas – RS – Brazil

eduardomoller, wagnerg, pilla, adenauer {@ucpel.tche.br}

**Resumo.** *A Computação Pervasiva pode ser definida como um ambiente computacional, no qual sistemas distribuídos, mobilidade de código e de sistemas, e elevada heterogeneidade somam-se de forma integrada para prover aos usuários um ambiente que se ajusta de forma autônoma para realizar as tarefas. A Computação Reconfigurável, por sua vez, disponibiliza dispositivos lógicos com a possibilidade de agregar o desempenho de um hardware fixo com a flexibilidade do hardware programável através de software. No entanto, identifica-se uma não disponibilidade de ferramentas que possam executar reconfigurações nestes dispositivos em ambientes de alta heterogeneidade, elevada dispersão geográfica e grande número de dispositivos. Neste sentido, este trabalho implementa um Serviço de Atualização de Configurações de Hardware - SACH, que possibilita a propagação de pacotes de bitstream de forma distribuída e concorrente para vários dispositivos de borda do tipo FPGA (Field Programmable Gate Array). O SACH foi modelado para atender as demandas impostas pela Computação Pervasiva, tendo sido integrado ao middleware EXEHDA.*

## 1. Introdução

Em se tratando de sistemas embarcados, aspectos como tamanho e dissipação de energia (que incorre em consumo das baterias e calor) muitas vezes são impeditivos para o uso de soluções em software rodando sobre processadores de propósito geral. Por outro lado, considerando os requisitos atuais de redução de tempo no projeto, prototipação e produção de sistemas embarcados, associados aos altos custos de projetos em hardware dedicados, as tecnologias de hardware reconfigurável emergem como uma alternativa promissora.

Os sistemas com hardware reconfigurável permitem que um sistema seja produzido em tempos mais curtos e ainda apresentam a possibilidade de que futuras modificações sejam feitas sem que haja a necessidade de intervenção física nos dispositivos (i.e., troca de circuitos). Por exemplo, em FPGAs (*Field-Programmable Gate Arrays*), uma *stream* de bits pode ser usada para reconfigurar os roteamentos e os dispositivos ativos em um conjunto de colunas. Dentre os casos em que a reconfiguração dinâmica é desejável, ou até mesmo necessária, incluem falhas de partes dos dispositivos e adaptações para otimizar o consumo ou desempenho. O principal inconveniente para a disseminação de sistemas embarcados em hardware reconfigurável para os quais pretenda-se aplicar atualizações futuras é a dificuldade da operação de atualização em

si, que normalmente envolve um operador no local e uma conexão específica, como por exemplo, cabos paralelos ou USB.

Dentro desse cenário, pode-se imaginar uma miríade de dispositivos reconfiguráveis, de diferentes fabricantes, modelos e periféricos, os quais tornam inviáveis a continuidade das técnicas correntes, que tem por base procedimentos manuais e individuais, para atualização das configurações de hardware. Para atacar esse problema, propomos o *Serviço de Atualização de Configurações de Hardware*, o qual integra-se como um serviço ao *middleware* para Computação Pervasiva EXEHDA [2] e provê a funcionalidade básica para que diferentes dispositivos possam ter suas configurações de hardware gerenciadas automaticamente e de modo remoto.

Este artigo está dividido nas seguintes seções: a seção 2 apresenta o Serviço de Atualização de Configurações de Hardware desenvolvidos nesta pesquisa. Os resultados obtidos com o protótipo do SACH são discutidos na seção 3. Por sua vez, a seção 4 apresenta a revisão do estado-da-arte, recorrendo sobre Computação Pervasiva e Dispositivos Reconfiguráveis. Finalmente, a seção 5 apresenta as conclusões e os trabalhos futuros.

## 2. Serviço de Atualização de Configurações de Hardware - SACH

Atualmente, o grande problema que se verifica em relação as arquiteturas reconfiguráveis do tipo FPGA é a forma de reconfiguração dos dispositivos lógicas programáveis, que é executada manualmente. A distribuição de novas configurações ocorre individualmente, tornando a tarefa muito onerosa. Esta situação se potencializa em um ambiente pervasivo onde poderá haver milhares de dispositivos instalados na rede. A partir deste momento, surgiram três motivações centrais para este trabalho: (i) explorar a capacidade das arquiteturas reconfiguráveis para solucionar as demandas introduzidas pela Computação Pervasiva; (ii) desenvolver uma proposta que permitisse a reprogramação de dispositivos reconfiguráveis - FPGAs de forma distribuída e concorrente, sem que haja a intervenção direta do ser humano; e (iii) dotar o *middleware* EXEHDA de um novo Serviço.

O **Serviço de Atualização de Configurações de Hardware** tem por objetivo propagar pacotes de *bitstream* para reconfiguração dinâmica parcial de dispositivos reconfiguráveis, dentro de um ambiente pervasivo. Esta reconfiguração de pacotes para os FPGAs acontece de maneira dinâmica e parcial, distribuindo pacotes para vários tipos de dispositivos lógicos concorrentemente, e sem a intervenção direta do operador. Entende-se por reconfiguração dinâmica parcial aquela que ocorre quando algumas funções lógicas dos FPGAs são alteradas dinamicamente, sem ter a necessidade de desativar o dispositivo. Estas reconfigurações são possíveis apenas sobre algumas arquiteturas reconfiguráveis, como por exemplo o Virtex II-Pro [3].

### 2.1. Arquitetura do SACH

O servidor é formado por dez serviços: (i) **Busca**, que localiza pacotes entre os nodos; (ii) **Distribuição**, que propaga o pacote entre as células ou nodos; (iii) **Monitoramento**, o qual verifica a chegada de novos pacotes; (iv) **Importação**, ativada quando o Monitoramento detecta um novo pacote; (v) **Leitura**, que processa o XML correspondente a cada pacote; (vi) **Validação**, que confere a integridade do pacote; (vii) **Remoção**, responsável pela exclusão de pacotes mais antigo do Repositório Celular do Serviço - RCS; (viii) **Instalação**, o qual grava o novo pacote no RCS; (ix) **Despacho**, que dispara o serviço de exportação; e (x) **Exportação**, que propaga os pacotes aos dispositivos

de borda. No cliente, há três serviços: (i) Recebimento dos pacotes de reconfiguração *bitstream*, (ii) Atualização de pacotes nos *FPGAs*, e (iii) Finalização, o qual conclui a reconfiguração do dispositivo.

O SACH dispara o processo de reconfiguração face a dois eventos:

- I Quando são feitas atualizações de pacotes de reconfiguração de dispositivos do tipo *FPGAs* que estão armazenados no RCS de cada nodo ou célula;
- II Quando ocorre a substituição de um ou mais dispositivos, por motivo de defeito ou pelo acréscimo de novos dispositivos lógicos na rede. O processo de configuração ocorre quando o novo dispositivo é conectado na rede, e a partir deste momento, a atualização é solicitada ao gerente por meio de uma requisição automática do serviço ou pelo próprio operador da área que instalou o novo dispositivo.

Uma das contribuições de um serviço como *SACH* é a possibilidade que o mesmo confere ao sistema de poder de distribuir pacotes de reconfiguração a dispositivos lógicos de forma concorrente, diminuindo o custo de tempo e também pela diminuição dos problemas de sobrecarga na rede pelo uso de reconfigurações parciais. O tipo de envio de pacotes do SACH aos dispositivos é executado de maneira que permita diminuir o uso da banda, enviando pacotes de carga maior apenas para um determinado grupo de dispositivos ou células concorrentemente e de maneira eficiente. O modelo utiliza este tipo de tecnologia de transmissão para que não haja um tráfego na rede excessivo. Já a comunicação entre as células é executada pelo serviço *WORB* do EXEHDA, que executa o acesso remoto as células de forma *peer-to-peer*. O serviço é desenvolvido para proceder propagações de pacotes de reconfiguração de maneira dinâmica e parcial dos *FPGAs*, podendo reconfigurar um dispositivo ou um grupo de dispositivos totalmente de forma distribuída e concorrente, sem a intervenção direta do ser humano. Este serviço é integrado ao *middleware* EXEHDA que proporciona o suporte às aplicações distribuídas dentro de um ambiente pervasivo, utilizando parcialmente alguns serviços como por exemplo o *Executor* (serviço de disparo de execução), *BDA* (serviço da base de dados pervasiva das aplicações) e o *CIB* (serviço que implementa a base de informações da célula), buscando assim atender as demandas da Computação Pervasiva.

A estratégia empregada foi integrar o SACH instalando o mesmo na EXEHDA-base de cada célula. Sobre a comunicação interna do serviço é feita na forma Cliente-Servidor e entre as EXEHDAbases é empregado o protocolo P2P, e inerente ao EXEHDA. O protocolo P2P é organizado na forma denominada super-peer, na qual existe o conceito de células vizinhas. As células vizinhas são priorizadas no momento das comunicações ou buscas [1].

O protocolo de comunicação é utilizado TCP/IP de modo a simplificar o suporte necessário de *middleware* nos dispositivos reconfiguráveis. Além disso, as EXEHDAbases usam um esquema de comunicação ponto a ponto, minimizando a centralização e aumentando o potencial de escalabilidade e esta comunicação utiliza um serviço do EXEHDA chamado de *WORB* que é um protocolo de Invocação Remota de Métodos - RMI, que administra a desconexão, suportando assim o perfil das comunicações no meio pervasivo. O SACH situa-se no *middleware* EXEHDA. O serviço é instalado em cada EXEHDAbase dentro deste ambiente. Vários dispositivos de borda, em especial os *FPGAs*, são instalados em cada célula.

Cada célula tem um RCS onde se encontram armazenados todos os pacotes referentes a cada FPGA, juntamente com seu arquivo descritor. O modelo utiliza este tipo de tecnologia de transmissão para que não haja um tráfego na rede excessivo, diminuindo assim a sobrecarga. Já a comunicação entre as células é executada pelo Serviço *WORB* do EXEHDA, que executa o acesso remoto as células, onde a comunicação entre as mesmas é executada ponto a ponto, minimizando a centralização e aumentando o potencial de escalabilidade.

O SACH utiliza principalmente os seguintes serviços do EXEHDA:

- EXECUTOR: que controla e dispara os componentes de software da aplicação;
- CIB (Cell Information Base): onde se encontram as informações da célula de execução, bem como as dos usuários;
- WORB: responsável pela comunicação dentro do ambiente pervasivo provido pelo EXEHDA, onde o seu mecanismo é semelhante a RMI, sem que necessite que seja mantida uma conexão ativa durante toda a execução remota da aplicação;
- BDA (Code Repository): é o serviço responsável pelo armazenamento do código pervasivo da aplicação ou serviço que é executada, no caso o SACH é armazenado dentro deste repositório.

## 2.2. Funcionamento

A Figura 1 apresenta um resumo do que foi visto anteriormente no que se refere aos diagramas de atividades do funcionamento do SACH instalado no EXEHDA. A figura ilustra os módulos de funcionamento de uma maneira mais resumida desde a inserção do pacote de configuração no RCS, passando por todas as validações propostas pelo serviço até conseguir o objetivo que é a propagação de pacotes aos FPGAs que necessitem ser atualizados.

O Serviço apresenta três módulos de operação, onde o **primeiro módulo** é o de inserção de pacotes.

O **segundo módulo** do serviço disponibiliza três operações: a inserção de um novo FPGA, a edição e a exclusão de um FPGA já instalado na célula. A inserção é feita incluindo no arquivo descritor dos dispositivos um novo FPGA. A mesmo procedimento ocorre em relação a edição, isto é, alterar as informações do arquivo descritor do referido dispositivo por exemplo e por fim a exclusão que retira do arquivo descritor as informações relativas a esse dispositivo que foi retirado da área de operação fisicamente.

O **terceiro módulo** é representado pela operação de atualização de pacote de um novo FPGA que é instalado na célula e que necessita ser atualizado. Isso acontece em dois momentos: quando um novo FPGA é instalado ou quando o arquivo descritor de um FPGA foi modificado e que necessita de um pacote de *bitstream*.

## 3. Resultados

Para a avaliação do SACH, foi desenvolvido um simulador que realiza o recebimento e a confirmação de cada pacote enviado, de forma análoga a um sistema reconfigurável real em um ambiente pervasivo. Os primeiros testes demonstraram a corretude da funcionalidade de distribuição do SACH, sendo que os pacotes foram entregues sem erros em uma rede local. Após, diferentes cargas de números de FPGAs simulados e tamanhos de *bitstreams* foram executadas. Foram realizados testes para medir o tempo que

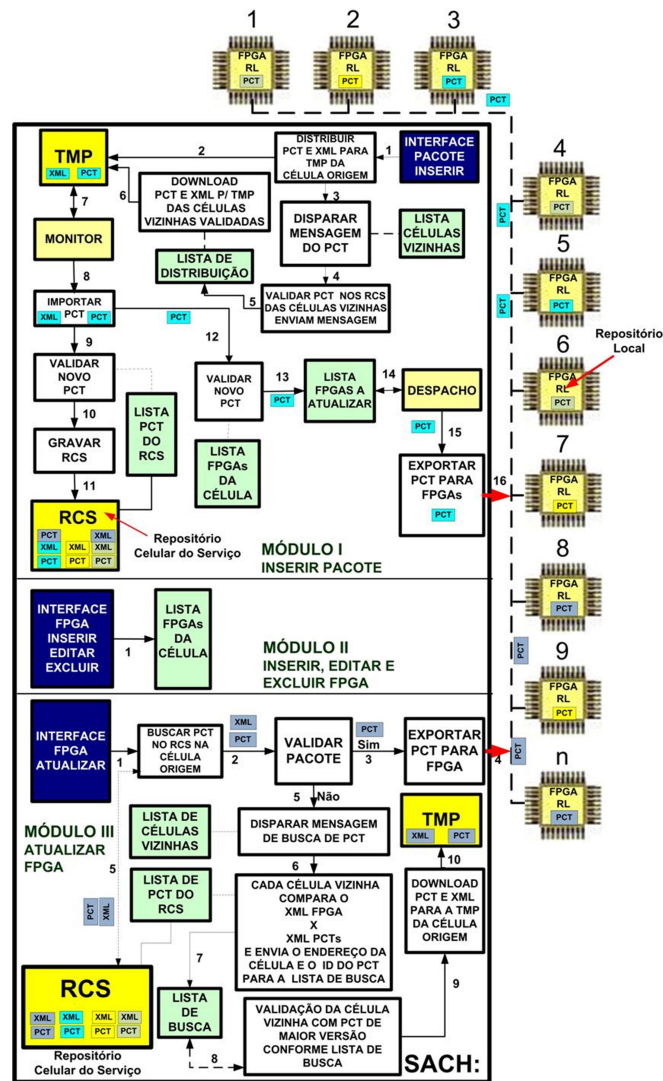


Figura 1. Funcionamento do SACH na EXEHDatabase

o protótipo SACH leva desde a inserção do pacote no serviço, passando por todas as validações necessárias, até a exportação do pacote aos dispositivos. Este tempo é denominado de Tempo de Processamento e Distribuição de pacotes para os dispositivos a serem reconfigurados.

### 3.1. Ambiente de Avaliação

Na Tabela 1, estão relacionadas as características dos equipamentos que foram utilizados para os testes do protótipo, juntamente com as informações técnicas de cada máquina. Foram usadas seis máquinas, sendo que uma para o Simulador de FPGAs, onde são realizadas as simulações do recebimento de pacote de reconfiguração pelos dispositivos FPGAs. Nos testes é usado a célula número 1 como sendo a célula origem e as células de números 0, 2, 3 e 4 são as chamadas células vizinhas ilustrada na Tabela 1.

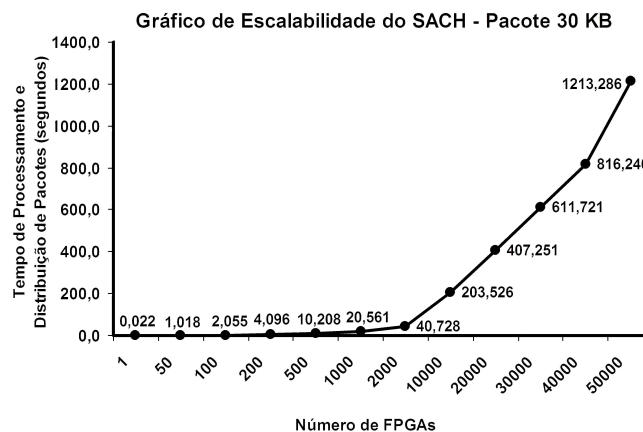
### 3.2. Variação do Número de Dispositivos Reconfiguráveis

O primeiros testes realizados com o SACH foram em relação a sua escalabilidade, estando os resultados obtidos ilustrados nos gráficos da Figura 2. Estes testes apresentam todas as

**Tabela 1. Configurações dos equipamentos usados nos testes**

Descrição	Célula 0	Célula 1	Célula 2	Célula 3	Célula 4	Simulador
Processador	<i>Celeron</i>	<i>Core Duo</i>	<i>Celeron</i>	<i>Celeron</i>	<i>Celeron</i>	<i>Celeron</i>
Velocidade (Hz)	300M	1.66G	300M	300M	300M	2.66G
HD (GB)	6.4	60	6.4	6.4	6.4	60
RAM (MB)	128	512	128	128	128	512

quantidades simuladas de FPGAs com o referido Tempo de Processamento e Distribuição de pacotes de cada grupo de dispositivos, de pacotes de tamanhos fixos de 30 KB e 1000 KB. Nos testes referentes à Figura 2, verificou-se que, para até 2.000 FPGAs, o tempo gasto para realizar todo o procedimento de validação e distribuição de pacotes, utilizando um pacote de 30 KB, ficou entre 0,022 e 41 segundos. Entre 10.000 e 40.000 FPGAs, o tempo variou entre 203 e 818 s, representando uma variação de 204 a 780 s. A cada 10.000 FPGAs acrescentados à simulação (até o limite de 40.000 FPGAs), o tempo cresceu 200 s. No caso de 50.000 FPGAs, obteve nos testes um valor de tempo gasto que ficou em 1213,286 s, ou seja, em torno de 20 minutos para reconfigurar 50.000 dispositivos.

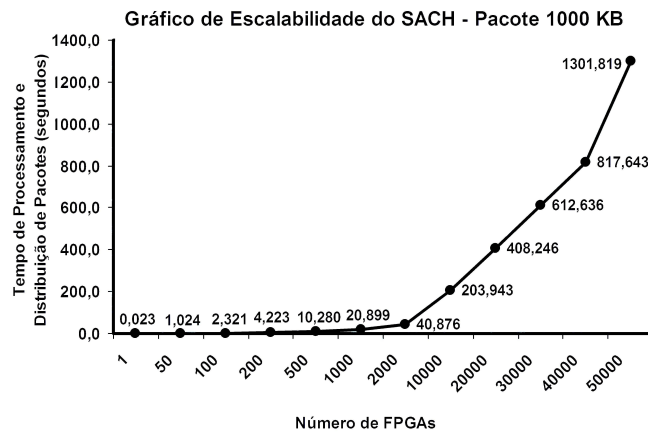


**Figura 2. Gráfico de Tempo de Processamento e Distribuição de pacotes do SACH - Pacote de 30 KB**

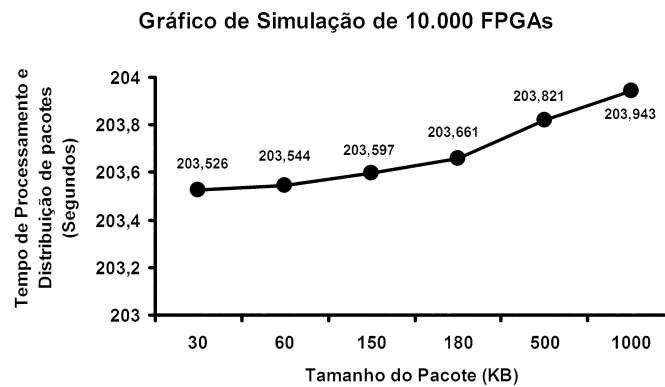
Já com 1000 KB (Figura 2) verificou-se que até 2.000 FPGAs, o tempo gasto para realizar todo o procedimento e a distribuição de pacotes ficou entre 0,023 e 41 s. Entre 10.000 e 40.000 FPGAs simulados, o tempo variou entre 204 e 819 s, ou seja, um aumento linear de 200 s a cada 10.000 FPGAs adicionais. Com relação a 50.000 FPGAs, obteve-se um valor de tempo que se manteve em torno de 1301,819 s.

### 3.3. Variação do Tamanho dos Pacotes

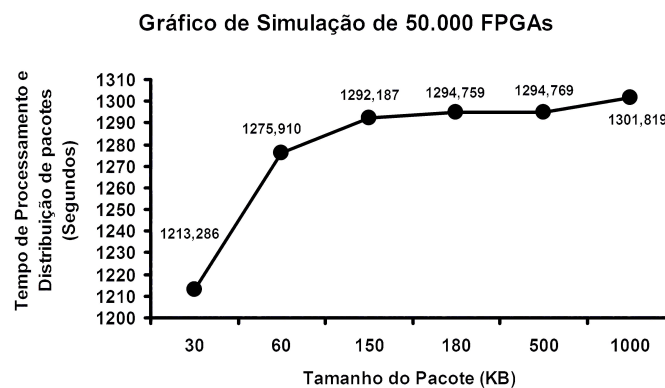
Em um segundo momento, o protótipo foi analisado em função do tamanho dos pacotes enviados. Para um mesmo número de dispositivos reconfiguráveis simulados, foram simulados um tamanho crescente de pacote sendo distribuído. O gráfico da Figura 4 apresenta a simulação de 10.000 FPGAs que recebem pacotes de reconfiguração de tamanhos entre 30 KB a 1000 KB. O Tempo de Processamento e Distribuição permaneceu entre 203,526 e 203,943 s, havendo um incremento de apenas 0,205% no tempo necessário para completar a distribuição. Outros testes com número de FPGAs simulados até 40.000 apresentaram comportamentos semelhantes, não sendo por isto discutidos neste artigo.



**Figura 3. Gráfico de Tempo de Processamento e Distribuição de pacotes do SACH - Pacote de 10000 KB**



**Figura 4. Gráfico de Tempo de Processamento e Distribuição de pacotes do SACH - 10.000 FPGAs**



**Figura 5. Gráfico de Tempo de Processamento e Distribuição de pacotes do SACH - 50.000 FPGAs**

Por sua vez, resultados diferentes foram obtidos com 50.000 FPGAs. A Figura 4 apresenta estes resultados – note-se que a escala é diferente da Figura 4. Entre 30 e

1000 KB, o tempo cresceu 7,296%, mostrando que o sistema já começava a aproximar-se do limite de uma única célula. Testes com 60.000 FPGAs esbarraram em uma barreira de memória da execução, levando o sistema a um bloqueio da execução. Isto deveu-se principalmente ao formato de armazenamento das informações dos FPGAs, os quais foram representados em arquivos XML, tecnologia escolhida não pela eficiência mas pela conveniência para implementação do protótipo.

#### 4. Conclusões e Trabalhos Futuros

Neste artigo, o Serviço de Atualização de Configurações de Hardware foi apresentado como uma solução para o gerenciamento remoto de múltiplos dispositivos reconfiguráveis em um ambiente pervasivo. O SACH está integrado ao EXEHDA, um *middleware* de grade pervasiva e, desta forma, pode aproveitar os serviços já implementados no *middleware* para reduzir a complexidade de implementação, melhorar o gerenciamento e otimizar a distribuição dos pacotes. Uma estratégia *peer-to-peer* é usada para comunicação entre as bases, equivalentes aos servidores em sistemas cliente-servidor, evitando a centralização excessiva e aumentando a escalabilidade do sistema. O protótipo comportou-se adequadamente e com escalabilidade, servindo configurações entre 30 e 1000 KB para até 50.000 FPGAs simulados com tempos aceitáveis, visto que o sistema não é projetado para distribuição de conteúdo em tempo real. Acima de 50.000 FPGAs, uma única célula não é mais suficiente para lidar com os arquivos de XML que armazenam tanto os FPGAs disponíveis e os pacotes a serem distribuídos, esbarrando no limite de memória do computador onde o serviço é executado. Como trabalhos futuros, a implementação do cliente do SACH a ser executado nos sistemas embarcados é uma meta importante para permitir que o sistema seja utilizado em situações reais. Além disso, pretende-se pesquisar alternativas de comunicação em grupos (*multicast*) para otimizar a transmissão de atualizações para dispositivos com as mesmas características. Uma otimização que permitirá aumentar o número de dispositivos servidos por uma única célula é o uso de uma base de dados relacional, por exemplo, para armazenamento das informações sobre os FPGAs e pacotes já instalados.

#### Referências

- [1] A. C. Yamin. *Arquitetura para um Ambiente de Grade Computacional Direcionado, às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva*. 2004. Tese de Doutorado, UFRGS, Brasil.
- [2] A. C. Yamin, I. Augustin, J. Barbosa, L. C. da Silva, R. A. Real, A. S. Filho, and C. F. R. Geyer, "Exehda: Adaptive middleware for building a pervasive grid environment," *Frontiers in Artificial Intelligence and Applications - Self-Organization and Autonomic Informatics*, vol. 135, pp.203–219, 2005.
- [3] Xilinx, "Virtex II Pro Platform FPGAs: Introduction and Overview." Disponível em: <http://www.xilinx.com/partinfo/ds083.pdf/>. Acesso em: Abril 2007, jun, 2007.