

ISAM: Uma Arquitetura de Software para *Pervasive Computing*

Adenauer C. Yamin¹, Jorge L. V. Barbosa^{1,2}, Iara Augustin³,
Luciano C. da Silva⁴, Rodrigo A. Real⁴, Cláudio F. R. Geyer⁴

¹*Escola de Informática
Universidade Católica de Pelotas
Pelotas, RS, Brasil
{adenauer,barbosa}@ucpel.tche.br*

²*Engenharia da Computação
Universidade do Vale do Rio dos Sinos
São Leopoldo, RS, Brasil
barbosa@exatas.unisinos.br*

³*Departamento de Eletrônica e Computação
Universidade Federal de Santa Maria
Santa Maria, RS, Brasil
august@inf.ufsm.br*

⁴*Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre, RS, Brasil
{lucc,rreal,geyer}@inf.ufrgs.br*

Abstract

The next years will be characterized for high levels of mobility, heterogeneity and interactions among devices connected to global networks. These interconnected networks will use as much wired connections as wireless. The first researches involving wide-area distributed systems answered many questions concerning resource management, although they fail in treating questions related to heterogeneity and dynamic adaptation. More recent works, through technologies like CORBA and Java/Jini, deal with heterogeneity problem, but do not deepen into adaptability aspects. In this article, we propose inside of the project ISAM, a treatment of this subject. ISAM contemplates an integrated approach of software and execution environment addressed to the administration of adaptation in heterogeneous networks, supporting the logical (software) and physical (hardware) mobility, oriented to the execution of distributed applications in a global scale and based on software components.

Keywords: Network Computers, Distributed Systems.

Resumo

Os próximos anos serão caracterizados por elevados níveis de mobilidade, heterogeneidade e interação entre dispositivos conectados a redes globais. As primeiras pesquisas envolvendo sistemas distribuídos em redes wide-area, responderam a diversas questões pertinentes ao gerenciamento de recursos. Trabalhos mais recentes empregando tecnologias como CORBA e Java/Jini abordam a questão da heterogeneidade, porém não se aprofundam em aspectos pertinentes à adaptabilidade. Neste artigo, apresenta-se o modelo ISAM, uma proposta para adaptação em redes heterogêneas, com suporte às mobilidades de software e hardware, voltada à execução de aplicações distribuídas em escala global e baseada em componentes de software.

Palavras-chave: Redes de Computadores, Sistemas Distribuídos.

1 Introdução

Nos últimos anos, os avanços da microeletrônica vêm diminuindo o preço do hardware e aumentando seu poder computacional. Além disso, o desenvolvimento de soluções eficientes para interconexão dos sistemas computacionais fez com que a área de redes de computadores assumisse uma posição de destaque. O crescimento exponencial da *Internet* vem sendo considerado um fenômeno tecnológico e de mercado. Neste contexto, as plataformas computacionais vêm migrando de sua natureza centralizada para uma nova realidade distribuída. Esta nova perspectiva de processamento em rede assume diferentes perspectivas: *Metacomputing*, *Grid Computing*, *Internet Computing* e mais recentemente *Peer-to-Peer Computing* [18].

Atualmente, os estudos sobre mobilidade em sistemas distribuídos são impulsionados pela proliferação de dispositivos eletrônicos portáteis. Este novo paradigma computacional distribuído e móvel é denominado computação móvel. Amplia-se assim o conceito de rede-sem-fio. Nesta nova perspectiva, o usuário portando dispositivos móveis como *palmtops* e *notebooks*, independentemente da sua localização física, terá acesso a uma infra-estrutura de serviços [4].

Neste contexto observa-se um movimento em direção à *Pervasive Computing* [22], onde novas aplicações atendem as necessidades dos usuários que se deslocam. *Pervasive Computing* é a proposta de um novo paradigma computacional, que permite ao usuário o acesso ao seu ambiente computacional a partir de qualquer lugar, a qualquer tempo, usando vários tipos de dispositivos (móveis ou não). Nesse sentido, a aplicação ou o ambiente de execução pró-ativamente monitoram e controlam as condições do contexto. A aplicação reage às alterações no contexto através do processo de adaptação. Este processo requer a existência de múltiplos caminhos de execução para uma mesma aplicação ou configurações alternativas que exibam diferentes perfis de utilização (históricos) dos elementos computacionais.

Esta visão apresenta uma série de novos (e renovados) desafios, oriundos do dinamismo e heterogeneidade do ambiente, além dos novos requerimentos das aplicações no estilo *follow-me* da *Pervasive Computing*. O dinamismo está tanto na aplicação quanto no sistema de execução. Ambos operam em um ambiente cujas condições na disponibilidade e no acesso aos recursos são variáveis no tempo e no espaço. Como conseqüência, novos tipos de aplicações estão aparecendo, as quais têm um comportamento determinado pela sua sensibilidade à variação nas condições de alguns elementos do ambiente. O ambiente é definido por elementos computacionais que podem ser medidos, como largura de banda, latência da rede, consumo de energia, localização do usuário, preferências do usuário, entre outros [10]. A complexidade do tratamento da adaptação introduz custos, tanto no gerenciamento, como na execução da aplicação. Estes custos podem se tornar elevados devido à necessidade de predições e estimativas de comportamentos futuros. Por outro lado, os ganhos potenciais podem ser altos. Davies [24] acredita que somente através de um processo de adaptação baseado em informações gerenciais sobre trocas na infra-estrutura de suporte, será possível operar eficientemente em um ambiente distribuído altamente dinâmico.

É no cenário da *Pervasive Computing* que a necessidade da adaptação se potencializa. Somente através de um mecanismo eficiente de adaptação, as aplicações executando em um computador móvel podem gerenciar questões tais como: variação imprevisível na qualidade da rede, grande disparidade na disponibilidade de serviços remotos e limitações nos recursos locais impostos pelas restrições de peso, pelo tamanho dos dispositivos móveis e pelo consumo de bateria.

Sistemas distribuídos tradicionais são construídos com suposições sobre a infra-estrutura física de execução, como conectividade permanente e disponibilidade dos recursos necessários. Porém, essas suposições não são válidas na *Pervasive Computing* [22]. Conforme o usuário se movimenta, a localização do seu dispositivo móvel se altera, e conseqüentemente, a configuração da rede de acesso e o centro da atividade computacional também se modificam. A mobilidade de hardware impede o uso direto das soluções adotadas pelos sistemas distribuídos atuais, construídos com premissas que envolvem somente a mobilidade de software.

Neste novo cenário, o comportamento adaptativo das aplicações levanta um conjunto de questões: (i) “quais são os domínios de aplicações adequados?”; (ii) “como os programas devem ser estruturados?”; (iii) “como o sistema básico de suporte deve trabalhar?”. O modelo ISAM (Infraestrutura de Suporte às Aplicações Móveis Distribuídas) aborda essas questões. O ISAM contempla uma arquitetura de software que enfoca de forma integrada os problemas da adaptação considerando cenários onde as tarefas da aplicação apresentam a propriedade de mobilidade. Esta mobilidade pode ser tanto física (equipamento se desloca) como lógica (o software transita entre os equipamentos).

Este artigo aborda particularmente uma das questões da arquitetura ISAM: a adaptação colaborativa entre a aplicação (programa) e o ambiente de execução (*middleware*). Na arquitetura ISAM, o sistema se adapta para fornecer qualidade, enquanto que a aplicação se adapta para manter a qualidade dentro da expectativa do usuário móvel. A estrutura do texto é a seguinte. Na seção 2 é apresentada a arquitetura de software ISAM. A seção 3 apresenta o modelo de adaptação proposto. Na seção 4 é discutida uma aplicação. A seção 5 analisa os trabalhos relacionados, e as considerações finais são apresentadas na seção 6.

2 Arquitetura de Software ISAM

A arquitetura proposta é organizada em camadas com níveis diferenciados de abstração. O ISAM busca a manutenibilidade da qualidade de serviços oferecida ao usuário móvel, através do conceito de adaptação. Uma visão organizacional desta arquitetura é apresentada na figura 1. Salientam-se dois pontos: (i) a adaptação que permeia todo o sistema, por isto está colocada em destaque; (ii) o escalonador, que é parte central do EXEHDA (*middleware ISAM*) [3].

A camada superior (SUP) da arquitetura é composta pela aplicação móvel distribuída. A construção desta aplicação baseia-se nas abstrações do Holoparadigma [15] [16], as quais permitem expressar mobilidade, acrescidas de novas abstrações para expressar adaptabilidade providas pelo ISAMadapt [9].

Por sua vez, a camada inferior (INF) é composta pelas tecnologias empregadas nos sistemas distribuídos existentes, tais como sistemas operacionais nativos e a Máquina Virtual Java.

2.1 Camada Intermediária – Primeiro Nível

A camada intermediária (INTERM) é o núcleo funcional da arquitetura ISAM, sendo formada por três níveis de abstração. O primeiro nível é composto por dois módulos de serviço à aplicação: Escalonamento e Ambiente Virtual do Usuário. O escalonamento, por sua vez, é o componente-chave da adaptação na arquitetura ISAM.

O Ambiente Virtual do Usuário (AVU) compõe-se dos elementos que integram a interface de interação do usuário móvel com o sistema. Este módulo permite que uma aplicação sendo executada em uma localização possa ser instanciada e continuada em outra localização sem interrupção, suportando assim o estilo de aplicações *follow-me*. O modelo foi projetado para suportar a exploração de aplicações contextualizadas (adaptadas aos recursos, serviços e localização corrente) e individualizadas (adaptadas aos interesses e preferências do usuário móvel). O desafio da adaptabilidade é suportar os usuários em diferentes localizações, com diferentes sistemas de interação que demandam diferentes sistemas de apresentação, dentro dos limites da mobilidade. Este módulo deve caracterizar, selecionar e apresentar as informações de acordo com as necessidades e o contexto em que o usuário se encontra. Para realizar estas tarefas, o sistema se baseia num modelo de uso onde as informações sobre o ambiente de trabalho, preferências, padrões de uso, padrões de movimento físico e hardware do usuário são dinamicamente monitoradas e integram o Perfil do Usuário e da Aplicação.

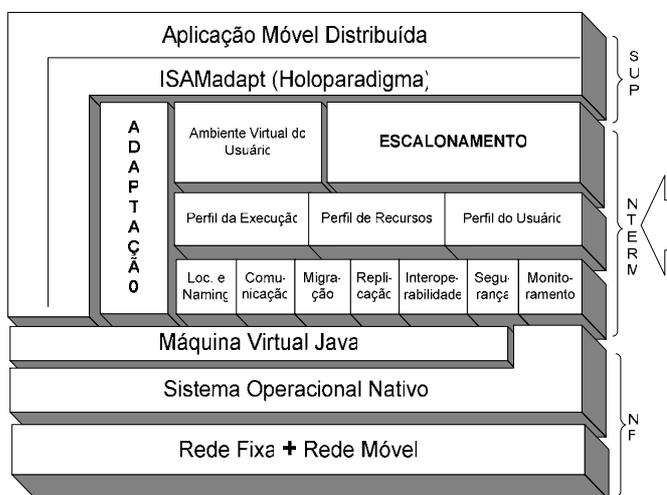


Figura 1. Arquitetura ISAM

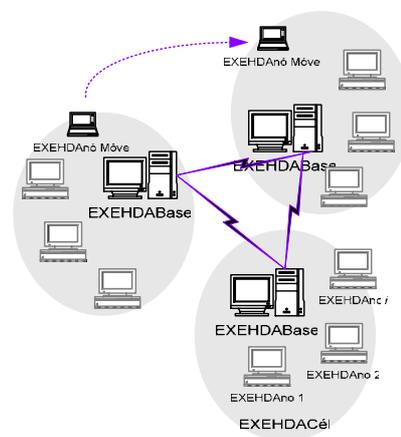


Figura 2. Ambiente de Execução ISAM

2.2 Camada Intermediária – Segundo Nível

Como já caracterizado anteriormente, na proposta ISAM busca-se um conceito flexível de adaptação que está relacionado ao contexto em que a aplicação está inserida. Por sua vez, a mobilidade de hardware introduz a possibilidade de movimentação do usuário durante a execução de uma aplicação. Desta forma, os recursos

disponíveis podem se alterar, tanto em função da área de cobertura e heterogeneidade das redes, quanto em função da disponibilidade dos recursos devido à alta dinamicidade do sistema. Assim, a localização corrente do usuário determina o contexto de execução, definido como “toda informação, relevante para a aplicação, que pode ser obtida e usada para definir seu comportamento” [9]. Numa análise preliminar, o contexto é determinado através de informações de quem, onde, quando, o que está sendo realizado e com o que está sendo realizado. Obter essas informações é a tarefa do módulo de monitoramento, que atua tanto na parte móvel quanto na parte fixa da rede.

As informações que dirigem as decisões do escalonador e dão suporte à aplicação para sua decisão de adaptação são advindas de quatro fontes: perfil da execução, perfil dos recursos, perfil do usuário e da aplicação (ISAMadapt). O módulo monitoramento do ISAM obtém informações do acompanhamento das aplicações executadas pelo usuário, em um dado tempo e em um dado local, com determinados parâmetros, o que permite determinar a evolução histórica e quantitativa das entidades monitoradas. A interpretação destas informações estabelece o perfil do usuário e das aplicações. Desta forma, as aplicações móveis ISAM poderão se adaptar à dimensão pessoal, além das dimensões temporal e espacial presentes nos demais sistemas móveis [10].

2.3 Camada Intermediária – Terceiro Nível

No terceiro nível da camada intermediária estão os serviços básicos do ambiente de execução ISAM que provêm a funcionalidade necessária para o segundo nível e cobrem vários aspectos, tais como migração – mecanismos para deslocar um ente de uma localização física para outra; replicação otimista – mecanismo para aumentar a disponibilidade e o desempenho do acesso aos dados; localização e *namimg* – para dar suporte ao movimento dos dispositivos móveis entre diferentes *EXEHDA Cêls* (veja figura 2), mantendo a execução durante o deslocamento.

3 A Adaptação Multinível Colaborativa no ISAM

A proposta ISAM contempla um comportamento adaptativo em dois segmentos: (i) na aplicação, a qual define o comportamento da adaptação (alternativas) e o contexto de seu interesse; (ii) no ambiente de execução (*middleware*), o qual tem um comportamento inerentemente adaptativo no momento em que processa a aplicação. Uma visão estrutural da Adaptação Multinível Colaborativa proposta no ISAM pode ser vista na figura 3.

Na codificação da aplicação, o programador especifica os elementos computacionais que afetam o comportamento da aplicação, os respectivos níveis de variação suportados e codifica comportamentos alternativos para atender à variação nas condições ambientais (variação nos elementos computacionais no contexto da aplicação).

Estas informações influenciam o comportamento adaptativo a ser adotado pelo ambiente de execução. Por sua vez, o ambiente de execução (*EXEHDA - middleware* ISAM) fornece meios: (a) para que sejam monitorados elementos computacionais do ambiente, (b) para que a aplicação possa registrar seu interesse em determinados elementos, (c) para notificar à aplicação das alterações ocorridas, e (d) para selecionar o comportamento alternativo mais adequado ao ajuste das novas condições ambientais. Como o sistema gerencia as aplicações, este também pode ter um comportamento pró-ativo, e executar adaptações relativas à administração e desempenho do sistema de forma global.

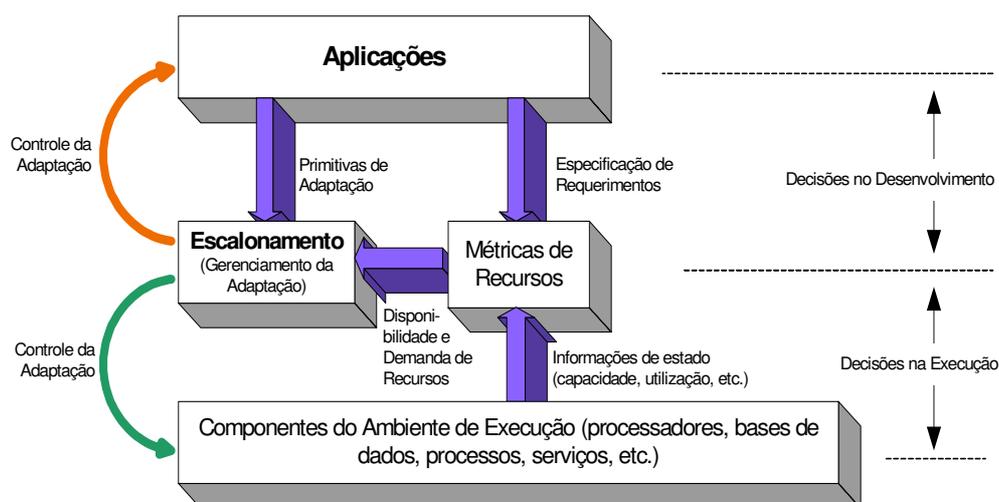


Figura 3. A Adaptação Multinível Colaborativa no ISAM

3.1 Adaptação no Nível da Aplicação

No ISAM o desenvolvimento de aplicações emprega as abstrações do Holoparadigma e do ISAMadapt através da HoloLinguagem para expressar o comportamento adaptativo da aplicação. A HoloLinguagem [16] é uma linguagem de programação que integra os paradigmas em lógica, imperativo e orientado a objetos. A mesma utiliza um modelo de coordenação que suporta invocações implícitas (*blackboard*). Este modelo de coordenação torna a linguagem apropriada ao ISAM, pois um requisito da *Pervasive Computing* é o desacoplamento temporal e espacial da comunicação [9]. A natureza da mobilidade não garante a interação contínua entre os componentes de uma aplicação distribuída neste ambiente. Desconexões são comuns, não somente devido ao meio físico mas sobretudo como uma estratégia para economia de energia nos dispositivos móveis. Logo, o ambiente móvel requer mecanismos de coordenação sem a premissa da conexão permanente. Além disso, devido à natureza imprecisa da mobilidade, o modelo de coordenação deve enfatizar a comunicação assíncrona e anônima.

A HoloLinguagem suporta ainda concorrência, modularidade, mobilidade e encapsulamento de *blackboards* em tipos abstratos de dados. No Holoparadigma [15], a aplicação é modelada com entes (entidade de existência) e símbolos (entidade de informação). Existem dois tipos de entes: elementar e composto. Um ente elementar é organizado em três partes: *Interface*, *Comportamento* e *História*. A interface descreve suas possíveis relações com os demais entes. O comportamento contém ações que implementam sua funcionalidade. Por sua vez, a história é um espaço de armazenamento compartilhado no interior de um ente. Um ente composto possui a mesma organização do ente elementar, no entanto, suporta a co-existência de outros entes na sua composição (entes componentes).

A Adaptação Multinível Colaborativa no nível de aplicação é implementada através das abstrações expressas na HoloLinguagem e da utilização das informações fornecidas pelo Monitoramento do Contexto (MC). Destaca-se aqui a existência dos *Holosensores* (seção 0), responsáveis por capturar informações que alimentam a tomada de decisões na arquitetura. Os *Holosensores* atuam em duas instâncias:

De sistema. Têm-se dois tipos principais de métricas: (i) uma para caracterização do *workload* dos *hosts* e (ii) outra para construção de perfis de comunicação entre objetos. Os mecanismos para capturar os perfis de comunicação entre os objetos são integrados com a API RMI de Java, preservando a compatibilidade com a semântica nativa da linguagem [17];

De aplicação. É feita uma monitoração utilizando a JVMPI (*Java Virtual Machine Profiler Interface*), a qual oferece a possibilidade de uma seleção dinâmica dos eventos de interesse da aplicação (por exemplo, ativação, interrupção e tempo de espera de métodos) que devem ser monitorados. No ISAM, esta monitoração não exige cuidados de programação, e os eventos a serem monitorados podem ser individualmente ativados e/ou desativados para redução de *overheads* desnecessários [6].

O MC é definido para manipular todos os recursos de interesse do sistema e das aplicações que podem afetar a arquitetura. Além disso, como as aplicações têm interesse em contextos particulares e podem requerer diferentes interpretações para os mesmos dados capturados (individualização do contexto), fez-se necessário definir uma abstração que definisse para a arquitetura os elementos do contexto de interesse específico das diversas aplicações: *HoloContexto da Aplicação* (HCA).

Para atender o requisito de independência, o sistema de monitoramento de recursos é desacoplado da aplicação, o que induz à necessidade de que as informações que este origina devam ser persistentes. Além disso, uma parte importante do contexto é a informação histórica que pode ser usada para deduzir o comportamento futuro. Esta predição é encapsulada na abstração *Interpretador do HoloContexto* (IHC), o qual produz uma informação parametrizada a partir de uma ou mais informações de contexto. Os geradores das informações de contexto são os *HoloSensores* (Hsensor), que são organizados de forma distribuída, próximos à fonte física/lógica que monitoram, embora sejam vistos como uma unidade pela aplicação (transparência). Os sensores devem trabalhar independentemente da aplicação, pois frequentemente fornecerão informações para múltiplas aplicações que estão em execução.

O uso das informações de contexto requer abstrações adicionais, pois estas informações frequentemente não estão na forma requerida pela aplicação. Por exemplo, um sistema GPS (*Global Positioning System*) fornece a informação na forma latitude-longitude; considerando a aplicação, esta pode requerer o nome da rua onde o usuário está. As aplicações trabalham em termos de informações de contexto em alto nível, com mnemônicos como “trabalho”, “casa”, “rede rápida”, “carga baixa”, que são fornecidas pelo MC, através da operação de interpretação do IHC. O modelo de contexto definido pelo projetista da aplicação (integrante do ISAMadapt) insere as regras de tradução das informações dos sensores em informações de alto nível consumidas pela aplicação. No exemplo, uma regra poderia ser posição GPS -> CEP -> <casa, rua, escritório>. O comportamento adaptativo da aplicação ISAM está esquematizado na figura 4.

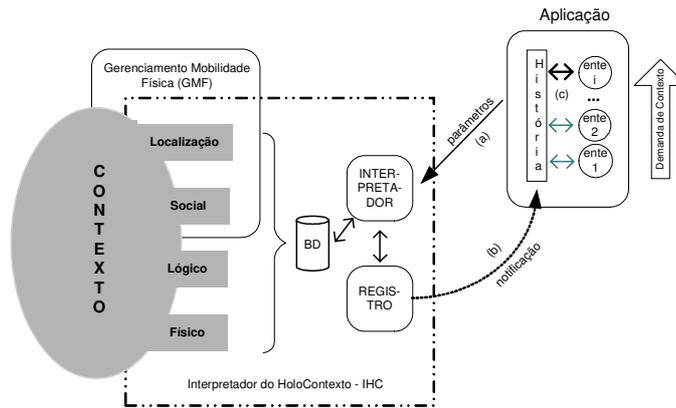


Figura 4. Adaptação no Nível de Aplicação

A aplicação é modelada com a adaptação em mente, onde o projetista implementa entes com funcionalidades semelhantes, porém com demandas de contexto diferentes. As aplicações, através do processo de REGISTRO do IHC, indicam os elementos do contexto que são de seu interesse (*ação a*). Junto com essa requisição de registro, a aplicação passa parâmetros utilizados pelo IHC para fazer a interpretação e notificação, quando houver uma alteração no contexto monitorado (*ação b*). O IHC formata as informações valoradas do contexto para a forma parametrizada pela aplicação. Por exemplo, o nível de bateria pode ser formatado como “alto”, “médio”, “baixo”. O sistema de notificação escreve no *blackboard* do ente que solicitou o registro a informação [bateria, baixa]. O mecanismo de invocação implícita do Holoparadigma [15] é acionado por esta escrita, e o ente que esperava pela notificação é ativado (*ação c*). Desta forma, o controle da adaptação é feito de forma automática pela invocação implícita do ente adequado ao contexto em questão.

3.2 Adaptação no Nível do Sistema

As aplicações ISAM solicitam, direta ou indiretamente, recursos do *middleware ISAM*. Algumas podem especificar uma determinada necessidade de qualidade de serviço (QoS), outras podem aceitar o “melhor-possível” nos níveis de serviço [19]. O escalonador é o artífice central no gerenciamento dos recursos e serviços, e a discussão sobre a estratégia de adaptação será feita com foco no mesmo.

A arquitetura ISAM gerencia diversas aplicações que concorrem na utilização dos recursos. A otimização da execução de um subconjunto do total de aplicações não pode comprometer o nível mínimo de QoS necessário para o restante. Desta forma, o escalonador precisa trabalhar com uma visão global das execuções em andamento. Neste caso, as estratégias de adaptação exigem do mecanismo de escalonamento o tratamento de problemas de otimização utilizando critérios múltiplos [3].

3.2.1 Organização Física do Middleware ISAM

A forma como é organizada a distribuição lógica dos equipamentos afeta diretamente todos os serviços do *middleware*, e naturalmente o escalonamento. O ISAM utiliza uma organização fisicamente distribuída e cooperativa, representada na figura 5. A proposta está baseada em dois escalonadores: (i) **EscCél** e (ii) **EscEnte**:

EscCél: fica localizado no nodo *EXEHDABase* e atua entre as *EXEHDACéls*. Suporta atribuições no gerenciamento global da arquitetura, tais como:

- localizar recursos (hardware e software) mais próximos, para reduzir custos de comunicação;
- decidir quando e onde replicar serviços e/ou componentes de software (entes);
- decidir quando e para onde migrar os componentes de software;
- instanciar o Ambiente Virtual do Usuário nos *EXEHDAnodos*. Esta instanciação é feita sob duas óticas: (i) balanceamento de carga - neste caso é escolhido o nodo menos carregado, (ii) aspectos de afinidade da aplicação - exigência de memória, bases de dados, etc.;
- disponibilizar antecipadamente, por usuário, a demanda de componentes das aplicações e dos dados;
- repassar ao escalonador EscEnte a carga de trabalho (componentes de software) proveniente de outras *EXEHDABases*.

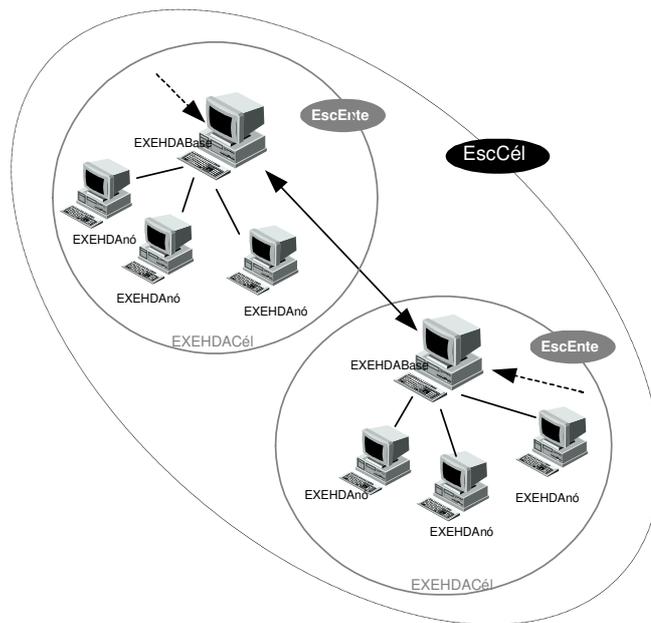


Figura 5. Organização do Escalonamento no Middleware ISAM

Pelas suas atribuições, além da consideração de custos de comunicação e balanceamento de carga, o escalonador EscCél atua de forma intensiva sobre aspectos de replicação e migração.

EscEnte: também existente em todas as *EXEHDABases*, tem atribuições no gerenciamento interno da *EXEHDACél*, tais como:

- efetuar o mapeamento dos componentes da aplicação nos *EXEHDAnodos* da *EXEHDACél*. Os critérios utilizados são balanceamento de carga e afinidade funcional;
- dar suporte aos procedimentos de adaptação colaborativa multinível com a aplicação.

Uma estratégia do escalonador EscEnte é associar o contexto (a *EXEHDACél*, as aplicações e os usuários) a grupos de escalonamento, onde cada grupo pode definir políticas específicas de balanceamento de carga. Cabe ao mecanismo de escalonamento gerir a evolução da execução das aplicações dos diferentes grupos segundo as políticas por ele selecionadas [8].

3.2.2 Principais Macro-Heurísticas Utilizadas

No ISAM, a adaptação permeia todas as decisões da arquitetura. O escalonador é tanto adaptador quanto adaptativo, ou seja, o escalonador é responsável pela execução do comportamento adaptativo da arquitetura, e ele próprio se adapta – altera-se conforme o ambiente corrente. As macro-heurísticas utilizadas pelo escalonador para adaptar-se são:

Aprendizado por reforço: à medida que o usuário interage com o sistema, seu comportamento é monitorado e seu perfil é construído. O escalonador emprega uma abordagem estocástica com aprendizado por reforço, na qual são construídas correlações estatísticas entre o usuário, o comportamento das suas aplicações e o ambiente de execução.

Instanciação otimizada das aplicações: o escalonador carrega nos *EXEHDAnodos* (móveis ou fixos) um conjunto mínimo de componentes de software que garantam a execução da aplicação (valendo-se do perfil do usuário), caracterizando uma estratégia *push* de operação. Os outros componentes, se necessários, serão solicitados sob demanda, caracterizando uma estratégia *pull* de operação.

Instanciação antecipada das aplicações: o processo de instanciação começa no momento em que o usuário efetiva sua autenticação na *EXEHDABase*, antes de solicitar a execução de aplicações. Neste caso, adota-se uma estratégia *push* de disseminação de componentes de software e informação. Esta instanciação também pode ocorrer com uma antecipação ainda maior, tendo por referência uma expectativa de roteiro de mobilidade do usuário já consolidada. Antecipar o tráfego na parte estruturada da rede (com conexão física) é uma opção da arquitetura proposta para aumentar o desempenho global da aplicação móvel, e conseqüentemente reduzir o tempo de espera/conexão do usuário do segmento de rede com suporte à mobilidade (conexão sem fio).

4 Aplicando a Adaptação Multinível Colaborativa ISAM

Esta seção tem por objetivo instanciar o emprego da estratégia de adaptação do ISAM. A avaliação da Adaptação Multinível Colaborativa será feita usando aplicações distribuídas voltadas para o aumento de desempenho.

A resolução de problemas numericamente intensivos é, em geral, computacionalmente dispendiosa, exigindo um elevado tempo de processamento. Muitos destes problemas apresentam características que facultam dividir o que deve ser processado em *Unidades de Trabalho* (UTs), as quais podem ser calculadas de forma independente.

Objetivando reduzir o tempo para solução desse tipo de problema, foi implementada uma ICP (*Internet Computing Platform*) sob a arquitetura ISAM, denominada UniCluster. A sua codificação em ISAMadapt, empregando o Holoparadigma foi mapeada para Java [26].

4.1 UniCluster

O UniCluster é composto por *servidores* distribuídos, alocando UTs, associados a um conjunto de *clientes* encarregados de realizar o processamento sobre as mesmas. A topologia de interconexão dos servidores é determinada pelo administrador do sistema. O principal critério de visibilidade entre os servidores é a velocidade da interconexão de rede existente. Os principais componentes da arquitetura do UniCluster são:

- servidores - compostos de dois fluxos de execução: (i) um dedicado a atender os pedidos de informações sobre o estado atual do sistema, e outro (ii) tanto para atender às solicitações de tarefas para processamento, como para recebimento de resultados parciais;
- clientes - a proposta UniCluster contempla a existência de três tipos de programas clientes:
 - *processor clients*: clientes processadores que executam os algoritmos de interesse do usuário e trabalham sobre as UTs, retornando resultados ao final do processamento de cada uma;
 - *loader clients*: clientes que submetem aplicações e seus parâmetros para execução;
 - *status clients*: clientes que buscam informações sobre o estado do sistema (processamentos em andamento, o que já foi realizado, etc.), bem como resultados finais de problemas já resolvidos.

Para permitir que o usuário possa iniciar e acompanhar suas execuções de qualquer lugar, sem precisar instalar software específico para tal, os *loader clients* e os *status clients* foram dotados de interface WEB. Isto facilita que, de qualquer navegador HTML, o usuário possa acompanhar os processamentos de seus problemas. Eventos significativos também podem ser notificados por e-mail. O estudo de caso deste artigo é baseado em um dos problemas resolvidos no UniCluster, denominado VIGAPI.

4.2 VIGAPI: Um Problema de Engenharia Civil

A análise de vigas e pilares de concreto normalmente é feita utilizando-se valores médios para as propriedades dos materiais e valores nominais para as dimensões da estrutura. Contudo, esses dados sempre apresentam desvios em relação aos valores reais. Para avaliar os efeitos destas incertezas sobre a resposta estática da estrutura, é empregado o método de Monte Carlo. Os dados de entrada de cada análise são gerados aleatoriamente, de acordo com suas distribuições de probabilidade. As análises são repetidas inúmeras vezes, obtendo-se diversas respostas, que são armazenadas. Um processamento estatístico sobre os dados permite determinar a média, o desvio padrão e estimar a distribuição de probabilidade de deslocamentos, deformações e tensões. Esta estratégia foi utilizada na concepção da aplicação VIGAPI [21].

Particularmente no caso da aplicação VIGAPI, o ganho a ser obtido com o uso da estratégia multinível colaborativa em redes de nodos heterogêneos, tem origem nas barreiras de sincronização que ocorrem durante as execuções paralelas. No momento de uma sincronização, uma tarefa que esteja em execução em um nodo lento, pode postergar a continuidade do processamento distribuído.

Por sua vez, aplicações baseadas em métodos estatísticos, semelhantes à VIGAPI, facultam serem decompostas em Unidades de Trabalho (UTs) com custo computacional diferenciado. A estratégia adaptativa ficou com a seguinte organização:

- no nível de aplicação: foram especificados três níveis de custo computacional para as UTs. Para expressar essas informações foram empregadas as primitivas de adaptação do ISAMadapt;
- no nível de arquitetura do sistema: o escalonador do gerente da execução, (*Servidor*) utilizando as especificações do usuário, entrega para os processadores (*processor clients*) as UTs compatíveis com o poder computacional do nodo que o aloja (os nodos processadores foram agrupados em três conjuntos - veja coluna "Grupos" na tabela 1).

4.3 Avaliando os Resultados

Para avaliar o comportamento da arquitetura adaptativa do ISAM, processou-se diversas vezes o problema de Engenharia Civil, empregando o UniCluster no ambiente apresentado na tabela 1. Este ambiente se caracteriza por uma elevada heterogeneidade do poder computacional dos equipamentos envolvidos (coluna “Poder Relativo” na tabela 1), condição comum em uma ICP. Os equipamentos estão interligados por uma rede Ethernet de 10 Mbps. As medidas foram feitas com os equipamentos e a infra-estrutura de rede, dedicadas ao experimento. A configuração contemplou um servidor e cinco clientes. O baixo custo computacional do servidor permitiu que o mesmo fosse executado juntamente com um dos clientes de processamento.

Tabela 1. Ambiente de Execução Utilizado

Nodo	Nome do Host	Plataforma	Sist. Op.	Poder Relativo ¹	Grupos 3 níveis	Grupos 2 níveis
1	guaiaca	Sun Ultra 10 – 128 Mbytes RAM	SunOS 5.8	6,80	1	1
2	spica	Sun Ultra 5 – 192 Mbytes RAM	SunOS 5.8	5,44	2	1
4	poncho	Sun SPARCstation 20 – 128 Mbytes RAM	SunOS 5.8	3,40	2	2
3	jacui	Sparc Enterprise 150 – 128 Mbytes RAM	SunOS 5.5.1	2,42	3	2
5	adhara	Sparc Statiton 4 – 128 Mbytes RAM	SunOS 5.8	1,00	3	2

A figura 6 compara os tempos de execução para a aplicação VIGAPI utilizando ou não a estratégia adaptativa. Os processamentos foram realizados para diferentes combinações possíveis dos dados de entrada: número de simulações, número de UTs, dimensões da peça de concreto, margens de segurança, etc. A linha diagonal representa tempos de execução iguais, e os valores registrados (em segundos) são decorrentes da média de oito execuções. O aumento de desempenho da execução distribuída da aplicação VIGAPI com o uso da Adaptação Multinível Colaborativa variou de 10% a 50%, atingindo um valor médio de 30%.

Em função dos custos de comunicação inerentes a toda proposta de ICP, o UniCluster consegue atender aplicações distribuídas de granulosidade² média e elevada. O UniCluster, como foi concebido, é bastante apropriado para problemas do tipo mestre/escravo (um servidor atendendo múltiplos agentes de cálculo), como a aplicação VIGAPI. Porém, este também pode ser aplicado a problemas com um nível moderado de interdependência, situação em que cresce o número de barreiras de sincronização necessárias. O crescimento no número de barreiras de sincronização potencializa o ganho com o uso da estratégia de adaptação multinível colaborativa.

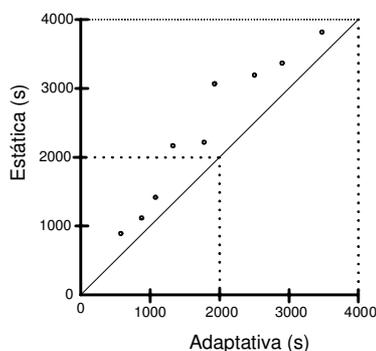


Figura 6. Tempos de Execução das Estratégias Estática e Adaptativa

A tabela 2 e a figura 7 registram o comportamento do desempenho na execução de uma instância do problema VIGAPI sob o ambiente da tabela 1, à medida que variam os níveis de adaptação. Entende-se por nível de adaptação, o número de classes de UTs (custo computacional) e de nodos processadores (em termos de poder computacional – veja tabela 2). No nível 1 todos os processadores são considerados iguais, e conseqüentemente as UTs também o são.

¹ Poder computacional relativo dos hosts, medido utilizando *benchmark* com comportamento semelhante à aplicação VIGAPI (manipulação de funções matemáticas e uso intensivo de processador).

² Granulosidade: razão entre o tempo de processamento da função a ser paralelizada e o tempo de comunicação necessário para o envio e o recebimento dos dados.

Tabela 2. Desempenho & Níveis de Adaptação

Níveis de UTs	5 Uts		10 Uts	
	Média (s)	Desvio Padrão (s)	Média (s)	Desvio Padrão (s)
1	3200	5,3	1970	2,7
2	2930	3,5	1890	2,3
3	2120	3,0	1510	2,1

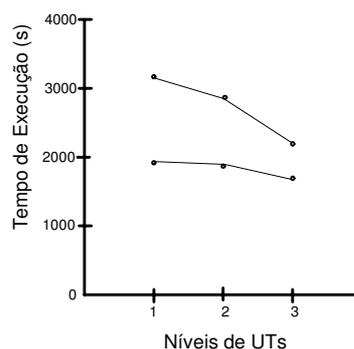


Figura 7. Desempenho & Níveis de Adaptação

O problema foi resolvido para dois níveis de granulosidade (a granulosidade com 5 UTs é o dobro da atingida com 10 UTs). O uso da estratégia adaptativa melhorou o desempenho nos dois níveis.

Pode-se observar que a redução da granulosidade melhorou o desempenho, reduzindo as perdas com a barreira de sincronização. Porém, é importante registrar que a redução da granulosidade é limitada tanto pelos custos de comunicação e de gerência da distribuição, como pela natureza do problema a ser resolvido. A figura 7 mostra que o uso da estratégia adaptativa aproximou os resultados atingidos com as duas granulosidades. Melhorar a possibilidade de trabalhar com granulosidades mais elevadas é particularmente conveniente para propostas que contemplem redes *wide-area*, nas quais os custos de comunicação sempre devem ser considerados.

Avaliando os testes, duas constatações se destacam:

comportamento adaptativo do paralelismo: o UniCluster permite a alteração do número de processadores alocados para uma aplicação, durante sua execução. Decorrem desta característica dois aspectos: (i) uma melhor possibilidade de aproveitamento da arquitetura, pois os nodos não ficam presos à determinada aplicação à medida que oscila a sua exigência de processadores, (ii) a possibilidade de não precisar aguardar que a aplicação atual encerre completamente, para direcionar os processadores em uso para novas aplicações. Isto diferencia o UniCluster de propostas bastante disseminadas como o PVM e o MPI [5]. Esta característica é uma decorrência da estratégia anônima e assíncrona de entrega das UTs (uso de *blackboard*: principal estratégia da arquitetura ISAM para comunicação/sincronização entre processos);

aumento do paralelismo médio: registrou-se que o emprego da adaptação multinível colaborativa para gerenciar a entrega seletiva de carga, considerando o poder computacional dos processadores envolvidos, conseguiu reduzir os custos com equipamentos parados nas barreiras de sincronização. Deste aumento é que resultaram as melhorias de desempenho observadas nas figuras 6 e 7.

5 Trabalhos Relacionados

Existem diversos trabalhos relacionados com a premissa de empregar a Internet como infra-estrutura para aplicações distribuídas. Sistemas como o Condor [13] são voltados para aplicações de alto desempenho em *clusters* de estações de trabalho. Diferentemente do ISAM, utilizam um mecanismo central para disparar processos. O projeto Globus [11] disponibiliza uma “grade de recursos computacionais” [12] integrando equipamentos heterogêneos em um único sistema. De forma similar à proposta ISAM, ele contempla uma estrutura escalável e distribuída para o gerenciamento de recursos. Apesar de conter um módulo específico para o controle de aplicações (GEM – *Globus Executable Management Service*), a atual versão trata as aplicações como um único módulo executável, ao invés de uma coleção de componentes que podem ser parcial e dinamicamente instanciados, como na arquitetura ISAM.

Por sua vez, sistemas como Globe [23], Legion [2] e WebOs [28], apesar de suportarem diferentes níveis de configuração, não consideram a adaptabilidade e a configuração automática do ambiente de execução como uma questão central. Por outro lado, as pesquisas em aplicações móveis adaptativas focalizam várias facetas da mobilidade e podem ser agrupadas em três categorias: (i) monitoramento de recursos, em especial recursos da rede [29][7][1]; (ii) projeto de aplicações móveis específicas [14][4][25]; (iii) *toolkits* para o desenvolvimento de aplicações [1][27][20]. Esses sistemas tratam de aspectos específicos do ambiente móvel. Diferente destes sistemas, a arquitetura ISAM é mais abrangente, propondo a integração do tratamento da mobilidade de hardware e de software. ISAM oferece o paradigma e a linguagem para o desenvolvimento das aplicações, e também o sistema de execução que monitora o contexto e fornece mecanismos de adaptação às alterações contextuais.

6 Considerações Finais

A arquitetura ISAM trata o problema da adaptação no cenário das aplicações distribuídas em escala global (*Pervasive Computing, MetaComputing*) através de um modelo multinível colaborativo. A aplicação estabelece políticas de adaptação, e tem um comportamento reativo à variação nas condições do seu contexto. Porém, diferentemente da maioria das outras propostas, esta reação é negociada com o ambiente de execução, que gerencia simultaneamente todas as aplicações em execução. Neste artigo foi resumida a proposta ISAM, e caracterizada a sua estratégia de Adaptação Multinível Colaborativa. Esta estratégia foi utilizada no desenvolvimento de um ambiente de ICP (*Internet Computing Platform*). O UniCluster executa aplicações distribuídas numericamente intensivas e proporciona melhoria de desempenho.

O desenvolvimento de aplicações em ambientes distribuídos *wide-area*, nos quais os recursos podem estar em estados não previsíveis, é uma tarefa complexa. A adição da mobilidade de hardware neste contexto traz um conjunto de novos requisitos e desafios para a produção de software, e cria demanda por novos tipos de aplicações. Esta perspectiva exige uma capacidade de adaptação maior que a apresentada pelos Sistemas de Gerenciamento de Recursos atuais. Exatamente este é o escopo de pesquisa explorado pelo projeto ISAM.

Referências Bibliográficas

- [1] A. Dey; G. Abowd. *Towards a Better Understanding of Context and Context-Awareness*. Proceedings of Conference on Human Factors in Computing Systems (HCI2000), 2000.
- [2] A. Grimshaw et al. *The Legion Vision of a World-Wide Virtual Computer*. Communications of the ACM. New York, v.40, n.1, 1997.
- [3] A. Yamin, I. Augustin, J.V. Barbosa, C.F.R. Geyer. *Exploring the Scheduling in Mobile Distributed Applications Performance*. Proc. Workshop in High Performance Computational Systems, Goias, Brazil, 2001.
- [4] B. Noble. *System Support for Mobile, Adaptive Applications*. IEEE Personal Computing Systems, 7(1), Feb. 2000.
- [5] D. Culler; J.P Singh. *Parallel Computer Architecture: a hardware and software approach*. Morgan Kaufmann. 1999. 479p.
- [6] E. Araujo; I. Augustin; A. Yamin; L. Silva; C.F.R. Geyer. *Uma Proposta de Monitoração para Visualização de Aplicações Distribuídas Java*. Jornadas Chilenas de Computación 2001. V Workshop en Sistemas Distribuídos y Paralelismo. Chile. 5-9 Nov. 2001.
- [7] G. Welling;B.R. Badrinath. *An Architecture for Exporting Environment Awareness to Mobile Computing Applications*. IEEE Transactions on Software Engineering. v. 24, n.5. 1998.
- [8] G.G.H. Cavalheiro; Y. Denneulin; J.L. Roch. *A General Modular Specification for Distributed Schedulers*. Proceedings of Europar'98. Southampton, Springer Verlag, LNCS 980. 1998.
- [9] I. Augustin, A . Yamin, C. Geyer. *Requisitos para o Projeto de Aplicações Móveis Distribuídas*. VIII CACIC Congreso Argentino de Ciencias de la Computación, Argentina, Oct, 2001.
- [10] I. Augustin, A. Yamin, J. Barbosa, C. Geyer. *Towards a Taxonomy for Mobile Applications with Adaptive Behavior*. International Symposium on Parallel and Distributed Computing and Networking (PDCN02), Innsbruck, Austria, Feb., 2002. Accepted for publication.
- [11] I. Foster; C. Kesselman. *The Globus Project: A Status Report*. Proceedings of the IPPS/SPDP - Heterogeneous Computing Workshop. 1988.
- [12] I. Foster; C. Kesselman: Editors. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers. San Francisco. 1999.
- [13] J. Basney and M. Livny. *Managing Network Resources in Condor*. Proceedings of the Ninth IEEE Symposium on High Performance Distributed Computing (HPDC9), Pittsburgh, Pennsylvania, August 2000, pp 298-299.
- [14] J. Jing; K. Huff. *Adaptation for Mobile Workflow Applications*. Proceedings of Workshop on Modeling and Simulation in Wireless Systems. Montreal, Canada. Jul. 1998.
- [15] J.L.V. Barbosa, A.C. Yamin, P.K. Vargas, Augustin I., C.F.R. Geyer. *Holoparadigm: a Multiparadigm Model Oriented to Development of Distributed Systems*. International Conference on Parallel and Distributed Systems (ICPADS), New York, IEEE Press, 2002.

- [16] J.L.V. Barbosa, C.F.R. Geyer. *A Multiparadigm Language Oriented to Distributed Software Development*. V Brazilian Symposium of Programming Languages. (SBLP), 2001.
- [17] L. Silva; A. Yamin; I. Augustin; J.L.V. Barbosa; C.F.R. Geyer. *Mecanismos de Suporte ao Escalonamento em Sistemas com Objetos Distribuídos Java*. VIII CACIC Congreso Argentino de Ciencias de la Computación. Santa Cruz, Argentina. Oct, 2001.
- [18] M. Baker, R. Buyya, D. Laforenza. *Grids and Grid Technologies for Wide-Area Distributed Computing*. Technical Report: 2001/92, Monash University, may 2001.
- [19] M. Maheswaran. *Quality of Service Driven Resource Management Algorithms for Network Computing*. The International Conference on Parallel and Distributed Processing Techniques and Applications. PDPTA, June 1999.
- [20] M. Ranganathan; A. Acharya; J. Saltz. *Sumatra: a Language for Resource-aware Mobile Programs*. Mobile Objects Systems: Towards the Programable Internet: Springer-Verlag Publisher, Serie Lecture Notes on Computer Science. v. 1222. Apr. 1997.
- [21] M. Real; A. Campos. *Aplicação do método de Monte Carlo no Estudo de Vigas e Pilares de Concreto Armado*. Revista Teoria e Prática na Engenharia Civil, Rio Grande, n.2, p.35-44, Maio 2001.
- [22] M. Satyanarayanan. *Pervasive Computing: Vision and Challenges*. IEEE Personal Communications, 2001.
- [23] M. Steen et al. *Globe: A Wide-Area Distributed System*. IEEE Concurrency. New York, v.7, n.1. 1999.
- [24] N. Davies, A. Friday, S. Wade, G. Blair. *Limbo: a Tuple Space Based Platform for Adaptive Mobile Applications*. Proceedings of the International Conference on Open Distributed Processing/Distributed Platforms (ICODP/ICDP'97), Toronto, Canada, 27-30, May 1997.
- [25] N. Davies, K. Cheverst, K. Mitchell. *A Friday, Caches in the air: Disseminating Tourist Information in the GUIDE System*. Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleand, USA, 1999.
- [26] R. Real; A. Yamin; M. Real; N. Duarte; O. Salvador. *UNICLUSTER: uma proposta de ICP voltada para PAD*. ERAD 2002. SBC. São Leopoldo, 2002.
- [27] T. Kunz; J.P. Black. *An Architecture for Adaptive Mobile Applications*. Proceedings 11th International Conference on Wireless Communications. Alberta, Canada. Jul. 1999.
- [28] T. Vahdat et al. *WebOS: Operating System Services for Wide Area Applications*. Proceedings of the Seventh Symposium on High Performance Distributed Computing. 1998.
- [29] T. DeWitt et al. *ReMoS: A Resource Monitoring System for Network-aware Applications*. Technical Report. CMU-CS-97-194. Carnegie Mellon University. Dec. 1997.
- [30] V. Bharchavan et al. *The Timely Adaptive Resource Management Architecture*. IEEE Personal Communications Magazine. v.5, n. 4, 1998.