

Integrando Dispositivos Móveis ao Middleware Integrade

Diego Souza Gomes e
Francisco José da Silva e Silva
Departamento de Informática
Universidade Federal do Maranhão
São Luís, MA - Brasil
diegosg1@gmail.com
fssilva@deinf.ufma.br

Markus Endler
Pontifícia Universidade Católica do Rio de Janeiro
Departamento de Informática
Rua Marques de São Vicente 225, Gávea
Rio de Janeiro, RJ - Brasil
endler@inf.puc-rio.br

Resumo

Este artigo apresenta uma infra-estrutura de software para acesso aos serviços do middleware de grade Integrade por parte de usuários de dispositivos móveis. Através deste mecanismo, clientes móveis podem solicitar a execução de aplicações na grade, realizar o acompanhamento da execução das aplicações e visualizar o resultado de computações já concluídas. Para alcançar estes objetivos, a arquitetura proposta possui mecanismos de ciência de contexto, provendo suporte a períodos de desconexão e baixa conectividade bem como à adaptação de conteúdo dos resultados das computações realizadas pela grade.

1. Introdução

Dispositivos computacionais móveis estão hoje disponíveis nas mais variadas formas como, por exemplo, *laptops*, *tablet PCs*, PDAs (*Personal Digital Assistants*), *smartphones* e telefones celulares, entre outros. Tecnologias de computação móvel e de rede sem fio tem evoluído muito rapidamente, de forma que muitos destes dispositivos possuem hoje considerável capacidade de processamento, armazenamento e comunicação. As diversas tecnologias de rede sem fio (*wireless networks*) que existem atualmente, tais como sistemas celulares, WLANs (*Wireless Local Area Networks*) e *Bluetooth*, permitem que seus usuários possam ter acesso a dados corporativos, pessoais e conteúdos da Internet de modo conveniente em qualquer lugar e a qualquer hora. Todas estas funcionalidades tem tornado estes dispositivos cada vez mais populares e utilizados por diversos grupos de pessoas para os mais variados fins.

Devido a esta popularização da computação móvel, usuários de dispositivos móveis formam um importante e novo segmento de clientes da computação em grade. A

computação em grade envolve a agregação de computadores conectados em rede para formar um sistema distribuído de larga escala que pode ser utilizado para realizar as mais variadas computações. Através da divisão da carga de trabalho em uma grande quantidade de computadores, uma grade pode dispor de um enorme poder computacional, de armazenamento, de transferência de dados, além de outros recursos compartilháveis. Esta infra-estrutura pode ser utilizada como uma extensão dos recursos computacionais de dispositivos móveis. Uma segunda abordagem para a integração de dispositivos móveis à ambientes de grades computacionais é a utilização dos mesmos como provedores de recursos (nós da grade). Esta abordagem é motivada pela crescente capacidade computacional destes dispositivos e pela presença cada vez maior das redes sem fio nos mais diversos ambientes.

O acesso a grades de computadores por dispositivos móveis requer a investigação de um conjunto de desafios, entre os quais destacam-se [14] [16]:

- Uma das grandes preocupações em sistemas de computação em grade está relacionada ao gerenciamento da heterogeneidade dos recursos que compõem a infra-estrutura de computação. A introdução de dispositivos móveis requer uma extensão do modelo que trata esta heterogeneidade, de forma a levar em consideração características como UCPs lentas, telas com diferentes capacidades em termos de resolução e definição de cores, variados dispositivos de entrada e limitações de memória e armazenamento, típicos de dispositivos portáteis [11];
- Tecnologias de comunicação sem fio apresentam menor largura de banda, maior taxa de erro, altas variações na qualidade da comunicação a medida que o usuário se desloca, além de períodos de desconexões ou conectividade intermitente [18]. Estes aspectos devem ser levados em consideração pelo *middleware* da

grade [15];

- Um esquema de mapeamento deve decidir como diferentes componentes de uma aplicação e da infra-estrutura da grade devem ser mapeados: no dispositivo móvel (*front-end*) ou em nós fixos da grade (*back-end*). Este mapeamento deve levar em consideração aspectos como capacidade de processamento, consumo de energia (devido a limitações das baterias que equipam dispositivos móveis), requisitos de recursos, entre outros [13].

Este trabalho descreve a infra-estrutura de software desenvolvida para integrar dispositivos computacionais móveis ao *middleware* de grade Integrate. Este artigo está estruturado como segue: a seção 2 descreve uma visão geral do *middleware* Integrate, a seção 3 apresenta a integração de dispositivos móveis à grade: seus requisitos, arquitetura de acesso aos serviços do *middleware* e aspectos de sua implementação. A seção 4 discute alguns trabalhos relacionados. Na seção 5 mostramos as conclusões decorrentes deste trabalho e na seção 6 apresentamos perspectivas futuras para a extensão do mesmo.

2 Visão Geral do Integrate

O projeto Integrate [9] é o resultado de um esforço multi-universitário com o objetivo de construir uma infra-estrutura de grade que usufrua o poder computacional das estações de trabalho ociosas para a execução de aplicações paralelas de alto-desempenho.

A unidade arquitetural básica do Integrate é o aglomerado (*cluster*), uma coleção de máquinas normalmente conectadas por um rede local. Aglomerados podem ser organizado hierarquicamente, permitindo a inclusão de um grande número de máquinas. Cada aglomerado contém um nó denominado *Cluster Manager*, que executa os componentes do Integrate responsáveis por gerenciar os recursos computacionais do mesmo e pela comunicação entre aglomerados.

Outros nós do aglomerado são chamados de *Workstations*. Estes exportam parte de seus recursos para os usuários da grade. Estes nós podem ser máquinas dedicadas ou compartilhadas. Os principais componentes da arquitetura do Integrate são:

- **Application Submission and Control Tool (ASCT)**: uma interface gráfica que permite que os usuários submetam aplicações e controlem suas execuções;
- **Application Repository (AR)**: armazena o código das aplicações que podem ser executados na grade;

- **Local Resource Manager (LRM)**: um componente que roda em cada nó do aglomerado, coletando informações sobre o estado dos recursos como memória, CPU, disco e uso da rede. Ele é responsável também por instanciar e executar aplicações escalonadas para o nó;
- **Global Resource Manager (GRM)**: gerencia os recursos do aglomerado recebendo notificações dos recursos utilizados dos LRMs (através de um protocolo de atualização de informações), e executa o escalonador que aloca as tarefas para os nós baseado na disponibilidade dos recursos;
- **Execution Manager (EM)**: mantém informações sobre cada submissão de aplicações, como seu estado, nó onde executa, parâmetros de entrada e saída, *timestamps* de submissão e conclusão. Ele também coordena o processo de recuperação das aplicações, caso ocorram falhas.

O Integrate atualmente permite a execução de três classes de aplicações:

1. **Aplicações regulares**, onde o código executável é designado para um único nó da grade;
2. **Aplicações paramétricas** ou BoT (*Bag-of-tasks*), onde várias cópias do código executável são designadas para nós diferentes da grade. Cada nó recebe um subconjunto dos dados de entrada da aplicação e computa o resultado em paralelo com os demais. Neste caso, não existe comunicação entre os nós;
3. **Aplicações paralelas** que seguem o modelo BSP ou MPI, onde vários nós são designados para a execução da aplicação e as computações são realizadas em cada um dos nós participantes, mas neste caso, os processos ocasionalmente trocam dados entre eles.

3. Acesso aos Serviços do Integrate Através de Dispositivos Móveis

A arquitetura da infra-estrutura de software para acesso aos serviços do Integrate por usuários de dispositivos móveis foi elaborada de modo a causar o mínimo impacto aos componentes já implementados deste *middleware*. Para tanto, o padrão arquitetural adotado foi o modelo cliente / *proxy* / servidor descrito em Pitoura et al. [17]. Nesta arquitetura, há um componente que exerce o papel de um *proxy* responsável por intermediar toda a comunicação entre os

clientes móveis e o servidor (a grade). Cabe a este *proxy* realizar adaptações de conteúdo, tradução de protocolos e *buffering* de mensagens, tornando os desafios da computação móvel transparentes à grade.

A arquitetura também prevê a possibilidade dos dispositivos móveis conectados à grade passarem por períodos de desconexão ou de baixa conectividade, comuns em redes sem fio, como quando os dispositivos adentram uma área fora do raio de abrangência dos pontos de acesso à rede ou mesmo em regiões dentro desse raio que, devido à própria topologia do local, formem sombras impedindo o alcance do sinal. Neste caso, requisições de início ou finalização da execução de aplicações podem ser agendadas para serem realizadas quando o dispositivo voltar a ter conectividade. Da mesma forma, informações oriundas da grade como o resultado de computações solicitadas anteriormente à desconexão podem ser armazenadas em uma *cache* na rede para que possam ser repassadas posteriormente ao dispositivo.

Por fim, um outro requisito atendido pela infra-estrutura de software implementada é o suporte a grande heterogeneidade tecnológica dos dispositivos, tanto com relação a propriedades de hardware quanto de software, decorrente das diversas famílias de equipamentos. Este objetivo é alcançado através da adaptação do conteúdo dos resultados das computações realizadas a serem recebidos pelos dispositivos móveis. Por exemplo, quando imagens de alta resolução não podem ser exibidas em determinados dispositivos, a resolução da mesma é adaptada para permitir exibição no visor do mesmo.

3.1. MoCA

A arquitetura para acesso aos serviços do Integrate através de dispositivos móveis foi implementada com o auxílio do *middleware* MoCA [4]. MoCA é uma arquitetura de *middleware* para o desenvolvimento de aplicações distribuídas cientes de contextos para redes móveis infra-estruturadas. A arquitetura MoCA oferece um conjunto de serviços que coletam e distribuem informações do contexto de execução de cada dispositivo.

Aplicações baseadas na MoCA são aplicações compostas por três partes: servidor, *proxy* e cliente (móvel). O *proxy* na MoCA tem o papel de intermediar a comunicação entre servidor e clientes, executando adaptações, se necessário, de acordo com o estado do cliente. Um arcabouço, denominado *Proxy Framework* [3], é disponibilizado para facilitar o desenvolvimento do componente *proxy*. Ele oferece facilidades para a configuração de contextos de interesse e associação das ações que devem ser tomadas quando da ocorrência de determinados contextos (ou estados). MoCA também disponibiliza aos desenvolvedores uma *ClientAPI* e uma *ServerAPI*, de modo a facilitar o desenvolvimento dos componentes cliente e servidor que

compõem a aplicação. Os clientes MoCA geralmente executam em nós móveis e o servidor tipicamente executa em um nó da rede fixa.

Quando executada, a *ClientAPI* automaticamente inicia um monitor MoCA, um *daemon* que executa em cada dispositivo móvel, responsável por coletar informações de contexto do dispositivo. Entre as informações coletadas, estão a qualidade da conexão, energia disponível, uso de CPU, quantidade de memória livre e a intensidade de sinal recebida de pontos de acessos visíveis. Todos os dados coletados são enviados para um serviço MoCA chamado *Context Information Service* (CIS). O CIS é um serviço distribuído onde cada servidor CIS, rodando geralmente em um nó da rede infra-estruturada, recebe, armazena e processa informações de contexto enviadas pelo monitores.

3.2. Arquitetura Proposta

A arquitetura proposta para a integração de dispositivos móveis ao Integrate foi elaborada tentando tornar o mais transparente possível os desafios provenientes da computação móvel descritos na seção 1. Baseada no modelo cliente/*proxy*/servidor, o dispositivo móvel assume características de cliente e o *middleware* de grade de servidor. O cliente móvel executa três componentes de *software*: o MASCT (*Mobile Application Submission and Control Tool*, um Monitor e o mCIS). No lado servidor, foram definidos os seguintes componentes: *GridProxy*, *ProxyAdapter*, CIS e o próprio *middleware* Integrate. O papel do *proxy* cabe ao *ProxyAdapter*. Esta arquitetura é ilustrada na figura 1 e seus componentes são descritos a seguir.

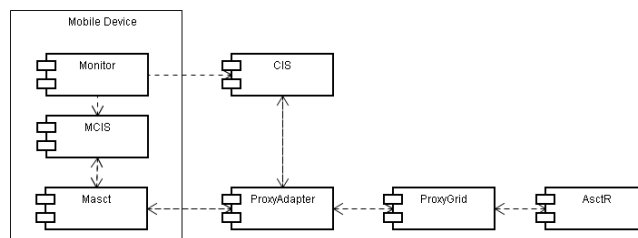


Figura 1. Componentes da arquitetura de acesso a grade através de dispositivos móveis

- **Mobile Application Submission and Control Tool (MASCT):** interface gráfica que permite aos usuários de dispositivos móveis requisitar os seguintes serviços da grade: submeter aplicações, acompanhar o andamento da execução das aplicações submetidas, receber notificações e visualizar os resultados das

computações finalizadas. Durante períodos de desconexão ou baixa conectividade, o MASCT armazena em *log* as requisições de execução efetuadas por seus usuários. As mesmas são enviadas à grade assim que a conexão é restabelecida;

- **mCIS**: um serviço de contexto instalado no dispositivo móvel capaz de prover informações de contexto para aplicações locais do dispositivo. Este componente informa ao MASCT eventuais períodos de desconexão;
- **Monitor**: um *daemon* que executa em cada dispositivo móvel e é responsável por coletar informações de contexto do dispositivo, que incluem a qualidade da conexão, energia disponível, uso de CPU, quantidade de memória livre, ponto de acesso à rede atual, lista de pontos de acesso dentro do raio de cobertura e suas intensidades de sinal;
- **Context Information System (CIS)**: componente da arquitetura MoCA que processa as informações de contexto enviadas pelo Monitor e envia notificações de eventos relacionados ao contexto dos dispositivos móveis ao *ProxyAdapter*. O CIS disponibiliza uma interface *publish/subscribe*. Notificações sobre eventos de contexto são geradas sempre que forem detectadas mudanças no estado dos dispositivos móveis para as quais o *ProxyAdapter* tenha registrado interesse;
- **ProxyAdapter**: componente responsável por intermediar toda a comunicação entre clientes móveis e o *GridProxy*, tratando eventuais desconexões através do armazenamento em uma *cache* das mensagens direcionadas ao dispositivo até que a conectividade seja restabelecida. É também responsável pela adaptação do conteúdo dos resultados das computações submetidas à grade, de modo a permitir sua exibição nos dispositivos móveis, utilizando o *Proxy Framework* do *middleware* MoCA;
- **GridProxy**: é o componente responsável por tornar os dispositivos móveis transparentes aos demais componentes da grade. O *GridProxy* recebe as requisições de cada MASCT executando nos dispositivos móveis e as encaminha para execução na grade. As notificações de aplicações finalizadas são também encaminhadas pela grade a este componente para serem, então, enviadas aos dispositivos móveis que as requisitaram;
- **Application Submission and Control Tool Refactored (ASCTR)**: corresponde a uma interface para o sistema de grade Integrate. Através das chamadas às APIs fornecidas pelo ASCTR, componentes podem ter acesso aos serviços da grade, como a solicitação para execução de aplicações regulares, paramétricas ou

paralelas, acompanhamento do estado das execuções submetidas e obtenção de seus resultados.

3.3. Aspectos de Implementação

O diagrama de seqüência ilustrado na figura 2 mostra os passos de uma requisição por parte de um usuário de dispositivo móvel para a execução de uma dada aplicação na grade. A requisição para execução parte do dispositivo através da interface do MASCT e é enviada ao *ProxyAdapter* (1). O *ProxyAdapter* implementa o *Proxy Framework* e utiliza suas funcionalidades, de modo a intermediar toda a comunicação entre clientes móveis e o servidor da aplicação (neste caso, o *GridProxy*). A comunicação entre MASCT e *ProxyAdapter* é realizada através de *sockets* TCP.

O *ProxyAdapter* repassa a mensagem com a solicitação de execução da aplicação ao *GridProxy* (2). Este componente mantém um banco de dados que relaciona cada solicitação a um serviço da grade ao dispositivo móvel responsável por sua emissão (3). Além disto, o *GridProxy* é responsável por manter os arquivos de entrada utilizados na execução da aplicação. Estes arquivos podem ser fornecidos pelo dispositivo móvel (no corpo da solicitação) ou podem ser obtidos através de uma url fornecida. Desta forma, o *GridProxy* formata os dados constantes da requisição de execução e os envia para a grade (4), de forma a tornar transparente o fato da requisição ter sido realizada através de um dispositivo móvel. A comunicação entre o *GridProxy* e a interface para submissão de aplicações do Integrate (ASCTR) é realizada através da tecnologia de objetos distribuídos CORBA (*Common Object Request Broker Architecture*).

Após constatar a disponibilidade de recursos para executar a aplicação solicitada, a grade retorna uma mensagem informando que a requisição foi aceita para execução (5). Esta mensagem é enviada ao *GridProxy*, que atualiza a entrada em seu banco de dados referente àquela requisição (6). Uma mensagem de notificação é, então, enviada ao dispositivo móvel através do *ProxyAdapter* (7). O resultado da computação realizada pela grade segue o mesmo percurso. Como citado anteriormente, o *ProxyAdapter* trata períodos de desconexão através de uma *cache* local e também realiza transformações nas mensagens de forma a adaptá-la ao dispositivo móvel específico ao qual ela se destina.

4. Trabalhos Relacionados

Uma abordagem utilizada para desenvolver clientes de grade para dispositivos móveis, consiste na utilização de uma implementação da especificação OGSi (Open Grid Service Infrastructure) [19] para estes dispositivos, chamada de Mobile OGSi.NET [7], contudo esta abordagem só é apropriada para grades que utilizam o OGSi para

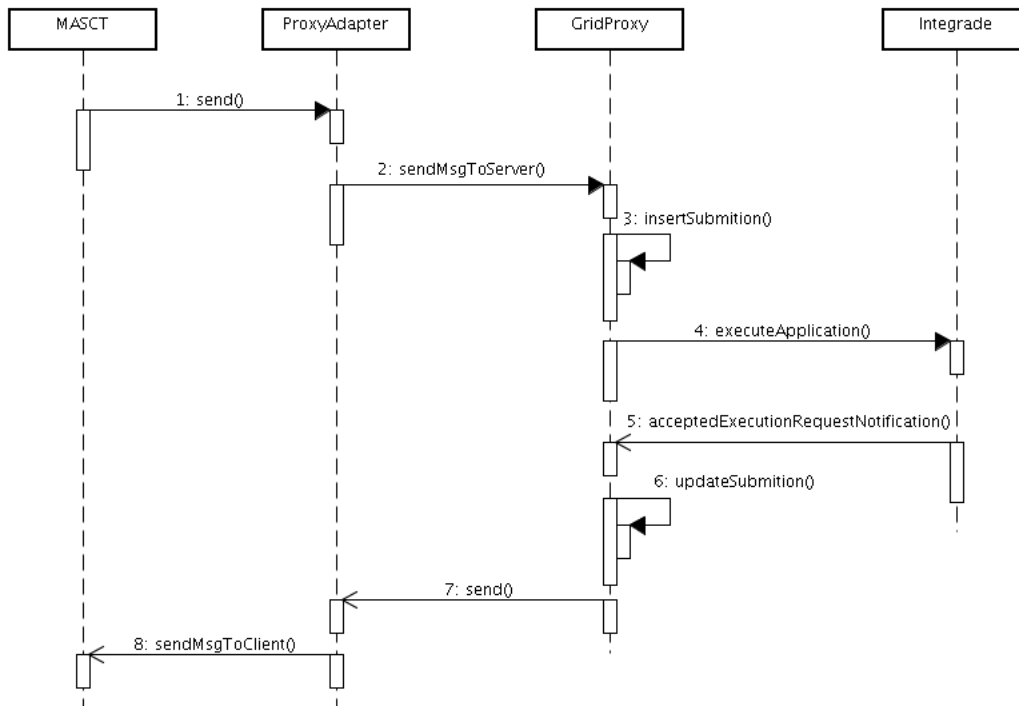


Figura 2. Diagrama de execução de uma aplicação na grade submetida utilizando dispositivos móveis

definir seus serviços, como é o caso do Globus Toolkit (GT3.2). A especificação OGSi está sendo substituída por outra especificação totalmente baseada em *web services*, a WSRF [5], o que levou a descontinuação do OGSi.NET em favor do WSRF.NET [12], que ainda não possui suporte para clientes móveis.

Outra opção para o desenvolvimento de clientes móveis para grades se baseia na utilização de *web proxies*. As requisições são feitas via uma interface web para os dispositivos móveis. Estas requisições são inicialmente tratadas pelo *proxy* que então realiza a requisição apropriada para a grade. Uma vez que o *proxy* recebe respostas da grade, ele gera e serve páginas web apropriadas para as capacidades limitadas dos *displays* dos dispositivos móveis, contendo estas informações. Nossa arquitetura também é baseada no modelo cliente/*proxy*/servidor, porém em nossa implementação, não utilizamos o protocolo http para realizar as requisições que serão enviadas a grade.

O sistema de grade Condor [1] provê uma metodologia hierárquica para acesso a grade por dispositivos móveis descrita em [10]. Nesta abordagem, as plataformas móveis são classificadas levando-se em conta suas restrições para acesso aos serviços da grade. Como resultado, são determinadas que funções da grade podem ser disponibilizadas

a cada camada da hierarquia. Uma característica fundamental desta abordagem consiste no fato de que as funcionalidades providas para dispositivos localizados em uma determinada camada, são disponibilizada também para os dispositivos das camadas mais baixas. Por exemplo, todos os serviços que podem ser obtidos através de um telefone celular, também podem ser obtidos com um PDA, mas o contrário não é garantido. Nossa implementação utiliza Java ME (CDC) *The Connected Device Configuration* [2], uma configuração da máquina virtual Java para dispositivos móveis com maior capacidade de *hardware* e conexão com a rede, para desenvolver o software cliente para acesso aos serviços do Integrate. Portanto, diferentemente do Condor nossa arquitetura não permite o acesso aos serviços da grade através de dispositivos que possuem recursos mais limitados, como os telefones celulares, somente através de dispositivos PDAs (Personal Digital Assistants). Contudo a arquitetura do Condor assume que os dispositivos têm acesso permanente a rede durante suas interações, não tratando como no Integrate, períodos de desconexão dos aparelhos.

5. Conclusões

Este artigo apresentou uma infra-estrutura de *software* através da qual clientes de dispositivos móveis podem ter acesso à grade computacional disponibilizada pelo projeto Integrate. Através desta infra-estrutura, usuários de aparelhos móveis podem solicitar a execução de aplicações, realizar o acompanhamento da execução de aplicações e visualizar o resultado de computações já concluídas utilizando a ferramenta MASCT.

A arquitetura da infra-estrutura de software proposta é baseada no modelo cliente-*proxy*-servidor, tornado os problemas relacionados ao uso de dispositivos móveis e redes sem fio transparentes aos componentes já implementados do *middleware* Integrate. A infra-estrutura utiliza componentes do *framework* MoCA para tratar períodos de desconexão e realizar adaptações no conteúdo dos resultados das computações retornados pela grade, aspectos pouco abordados em outros trabalhos relacionados a este.

6 Perspectivas futuras

A infra-estrutura de software atual torna possível que os dispositivos móveis tenham acesso aos serviços de grade fornecidos pelo *middleware* Integrate. Futuramente desejamos que estes dispositivos se integrem à grade não somente como consumidores de recursos, mas também como fornecedores, de modo que aplicações submetidas à grade possam ser escalonadas para execução nestes dispositivos. Para isso, a arquitetura supracitada deverá ser estendida para coletar e exportar informações de contexto dos dispositivos necessárias aos algoritmos de escalonamento de aplicações da grade, como, por exemplo, energia disponível, estado de utilização de CPU, quantidade de memória livre e capacidade de armazenamento de massa.

Temos também como objetivo implementar aplicações que explorem todas as funcionalidades adquiridas com a integração entre dispositivos móveis e grades de computadores. Em especial, pretendemos utilizar esta infra-estrutura conjuntamente com sistemas PACS (*Picture Archiving and Communication systems*) para acesso, compartilhamento, armazenamento e processamento de imagens. Pretendemos focar no domínio da área da saúde, compartilhando imagens de exames distribuídas entre vários domínios administrativos, de forma a integrar as informações disponíveis nas diversas instituições médicas existentes.

Agradecimentos

Este trabalho faz parte do projeto Integrate2, financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq.

Referências

- [1] Condor website. <http://www.cs.wisc.edu/condor>.
- [2] Java ME website. <http://java.sun.com/javame/technology/index.jsp>.
- [3] *MoCA Proxy Framework Manual de utilização*.
- [4] MoCA website. <http://www.lac.inf.puc-rio.br/moca/>.
- [5] The WS-Resource Framework website. <http://www.globus.org/wsrf/>.
- [6] I. M. Author. Some related article I wrote. *Some Fine Journal*, 99(7):1–100, January 1999.
- [7] D. Chu and M. Humphrey. Mobile OGS .NET: Grid Computing on Mobile Devices. *2004 Grid Computing Workshop (associated with Supercomputing 2004)*.
- [8] A. N. Expert. *A Book He Wrote*. His Publisher, Erewhon, NC, 1999.
- [9] A. Goldchleger, F. Kon, A. Goldman, M. Finger, and G. C. Bezerra. Integrate: Object-oriented grid middleware leveraging idle computing power of desktop machines. *Concurrency and Computation: Practice & Experience*. Vol. 16, pp. 449-459, 2004.
- [10] F. González-Castaño, J. Vales-Alonso, M. Livny, E. Costa-Montenegro, and L. Anido-Rifón. Condor grid computing from mobile handheld devices. *ACM SIGMOBILE Mobile Computing and Communications Review*, 7(1):117–126, 2003.
- [11] A. Helal. *Any Time, Anywhere Computing: Mobile Computing Concepts and Technology*. Kluwer Academic Pub, 1999.
- [12] M. Humphrey and G. Wasson. Architectural Foundations of WSRF .NET. *International Journal of Web Services Research*, 2(2):83–97, 2005.
- [13] J. JING, A. HELAL, and A. ELMAGARMID. Client-Server Computing in Mobile Environments. *ACM Computing Surveys*, 31(2), 1999.
- [14] M. Migliardi, M. Maheswaran, B. Maniymaran, P. Card, and F. Azzedin. Mobile interfaces to computational, data, and service grid systems. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(4):71–73, 2002.
- [15] S. Park, Y. Ko, and J. Kim. Disconnected Operation Service in Mobile Grid Computing. *International Conference on Service Oriented Computing (ICSOC'2003)*, Trento, Italy, Dec, 2003.
- [16] T. Phan, L. Huang, and C. Dulan. Challenge:: integrating mobile wireless devices into the computational grid. *Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 271–278, 2002.
- [17] E. Pitoura and G. Samaras. *Data Management for Mobile Computing*. Kluwer Academic Publisher, 1998.
- [18] W. Stallings. *Wireless Communications and Networks*. Pearson Prentice Hall, 2005.
- [19] S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maguire, T. Sandholm, P. Vanderbilt, and D. Snelling. Open Grid Services Infrastructure (OGSI) Version 1.0. *Global Grid Forum Draft Recommendation*, 6:27, 2003.