

Ambientes de *Clusters* e *Grids* Computacionais: Características, Facilidades e Desafios

Taís Appel Colvero¹, MAR Dantas¹ e Daniel Pezzi da Cunha²

¹PGCC – Universidade Federal de Santa Catarina (UFSC) – SC – Brasil
{tappel@inf.ufsc.br, mario@inf.ufsc.br}

²DCC – Universidade do Extremo Sul Catarinense (UNESC) – SC – Brasil
{dpc@unesc.net}

Resumo – Muitos pesquisadores e organizações têm visto a utilização de *clusters* e *grids* computacionais como soluções viáveis para um maior desempenho de suas aplicações. Desta forma, é importante que as características diferenciadas desses ambientes sejam conhecidas. Este artigo tem como objetivo apresentar uma pesquisa destes dois ambientes, especificando suas peculiaridades, seus desafios e suas complexidades. O conteúdo explorado neste trabalho permite uma visão ampla e objetiva sobre o uso destes dois paradigmas. Deste modo, serão mostradas as diferenças mais significativas destas configurações, em qual espaço podem ser utilizadas e como estas formas de processamento podem ser consideradas complementares.

Palavras chave – Ambiente de Alto-Desempenho, *Cluster* Computacional, Computação Paralela e Distribuída e *Grid* Computacional.

1. INTRODUÇÃO

Algumas áreas de conhecimento (por exemplo, astronomia, meteorologia e genética) requerem, para os problemas estudados, de muitos recursos computacionais com alto desempenho para suprir cálculos complexos e repetitivos. Os avanços da tecnologia de computadores geralmente não acompanham a demanda solicitada e, às vezes, a utilização de supercomputadores é inviável financeiramente. Uma alternativa a ser adotada pode ser a soma dos recursos computacionais já existentes utilizando-os de forma mais apropriada e equilibrada, resultando em um ganho substancial de desempenho (*speedup*). Neste contexto, podem ser aplicados os paradigmas de *cluster* e *grid* computacional que melhor usufruem, respectivamente, dos recursos e serviços de maneira local e geograficamente distribuída.

Os ambientes de alto desempenho, vistos tanto nos *clusters* como em *grids*, são alcançados através de uma melhor taxa na execução de aplicativos e no número de dados na aplicação se comparados à execução em uma máquina local. Para obter este cenário, três abordagens são consideradas: aumento na velocidade do processador, algoritmos mais otimizados e ambientes de computação paralela e distribuída [1]. Em ambos os paradigmas, estes esforços são somados garantindo a baixa latência de comunicação (na ordem de centenas de μ s) e a elevada taxa de transmissão (a partir de dezenas de Mbps até, atualmente, dezenas de Gbps) [2].

Um *cluster* computacional é um ambiente de computação paralela formado por um conjunto de computadores, chamados *nós*, interligados, muitas vezes, por dispositivos

do tipo *switches* em uma rede LAN de alto desempenho (exemplos são a *Myrinet* e *ATM*). Os nós cooperam entre si para atingir um determinado objetivo comum [3]. A arquitetura de um *cluster* é classificada, segundo [4], como MIMD do tipo fracamente acoplado [5], ou seja, tem memória distribuída (multicomputadores), por isto os nós devem se comunicar a fim de coordenar e organizar todas as ações a serem tomadas. Deste modo, externamente o *cluster* é visto como sendo um único sistema.

Por outro lado, uma configuração de *grid* computacional é um ambiente computacional geograficamente distribuído que permite a interoperabilidade entre organizações a ele associadas [6], [7], [8], [9]. Seu principal objetivo é compartilhar e agregar recursos de forma a disponibilizá-los como serviços. Dentre outros, os serviços oferecidos podem ser a alocação de recursos [10], o gerenciamento de processos, a comunicação, a autenticação e a segurança [11], [12], [13], [14]. Os recursos que compõem um *grid* sejam eles físicos, virtuais ou humanos, são interligados por redes WAN de alta velocidade (via cabo ou *wireless*). Por conseguinte, um *grid* pode ser considerado como um ambiente de alto desempenho [15], [16], [17], [18].

Neste artigo são enfocadas as peculiaridades dos *clusters* e *grids*, englobando seus espaços de utilização e mostrando que são formas de processamento que podem ser consideradas complementares. Desta forma, o artigo é organizado como segue. Na seção II são apresentados os ambientes de alto-desempenho. Na seção III, os ambientes *clusters*. Na seção IV, os *grids* computacionais. Por fim, na seção V, as conclusões.

2. AMBIENTES DE ALTO-DESEMPENHO

Um ambiente com processamento de alto-desempenho (PAD) é caracterizado, principalmente, pela execução de tarefas paralelas complexas e milhares de aplicações ao mesmo instante. Até pouco tempo, este ambiente era formado por computadores paralelos específicos (por exemplo, MPP e SMP). Atualmente, podemos encontrar uma estrutura desta categoria construída com componentes encontrados em lojas especializadas de informática (conhecido como *Components Off The Shelf* - COTS) e podem ser classificados como *clusters* e *grids* computacionais.

Os fatores que tangem ao desempenho dos *clusters* e *grids* são as políticas adotadas em relação aos recursos compartilhados, geralmente divididas em: política de transferência, política de seleção, política de localização e política de informação.

a) *Política de transferência*

A política de transferência define quais elementos de processamento estão aptos a receber e transmitir tarefas. Nesta etapa são definidos como receptores os elementos ociosos ou com poucas cargas de trabalho e transmissores os elementos sobrecarregados [19].

b) *Política de seleção*

Dependendo do caso, uma troca de tarefas pode ser feita, o que denota a segunda política: escolher, dentre tantas, a tarefa mais adequada a ser enviada a um determinado recurso. A esta função é dado o nome de política de seleção. Esta política define se o escalonamento será *preemptivo* ou *não-preemptivo*, ou seja, tarefas que já iniciaram sua execução ou tarefas que ainda não foram executadas [20].

c) Política de localização

A política de localização define, dentre todos os recursos disponíveis, qual o mais adequado a receber a carga. A escolha do elemento pode ser aleatória ou através de investigação para saber qual o mais adequado no momento.

d) Política de informação

Esta política define os locais onde as informações sobre o estado dos recursos são armazenadas e com que frequência são atualizadas. As informações podem ser armazenadas em um local específico ou em todos os recursos de duas formas: a) cada recurso possui informações sobre todos os outros, através de envio por *broadcast* ou; b) cada recurso tem apenas seu próprio estado.

As principais diferenças entre os ambientes de *cluster* e *grid* estão indicados na tabela 1, onde se destacam a autoridade reguladora (o *cluster* tem autoridade única e o *grid* contém múltiplas), a segurança (no *cluster* pode ser desnecessária e no *grid*, indispensável [13]) e o sistema operacional (no *cluster* deve ser homogêneo e no *grid* pode ser heterogêneo).

Tabela 1 – Diferenças entre as configurações de *cluster* e *grid* [21]

Configuração	<i>Cluster</i>	<i>Grid</i>
Domínio	Único	Múltiplos
Nós	Milhares	Milhões
Segurança do processamento e do recurso	Desnecessária	Necessária
Custo	Alto, pertencente a um único domínio	Alto, todavia dividido entre domínios
Granularidade do problema	Grande	Muito grande
Sistema operacional	Homogêneo	Heterogêneo

3. AMBIENTES *CLUSTERS*

Em um ambiente *cluster* a alocação de recursos é efetuada por domínio administrativo centralizado, sendo desnecessária a segurança do processo e do recurso, caso a rede de interconexão (*intracluster*) seja desacoplada da rede de acesso externo. Além disso, este tipo de ambiente pode se beneficiar de protocolos de comunicação mais eficientes entre suas unidades de processamento, pois como a rede de interconexão pertence ao mesmo domínio administrativo, o recurso é controlado [2], [21].

A abordagem *cluster* possui alta escalabilidade, pois tarefas de inclusão ou exclusão de nós escravos não exigem que sejam feitas modificações no ambiente, sendo realizadas de forma isolada, partindo do nó mestre, por meio da execução de algum comando específico do *software* escolhido. No entanto, um fator limitante é o número de nós, na ordem de dezenas de recursos. A figura 1 ilustra, resumidamente, este ambiente.

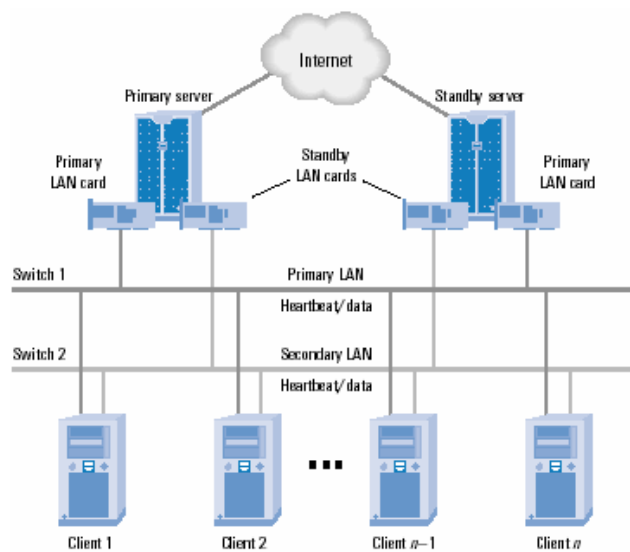


Figura 1 – Exemplo de uma arquitetura *cluster* [22]

Uma configuração de *cluster* pode ser conferida em [22] e [23] no qual foi montado em nosso laboratório de pesquisa. Para esta configuração foi utilizado o pacote de *software Open Source Cluster Application Resource - OSCAR - versão 2.3* para o gerenciamento. O *cluster* consiste de cinco computadores pessoais, onde um desempenha a função de nó mestre (servidor) e os restantes formam os nós escravos (clientes). No momento da instalação de um *cluster* é gerada uma imagem do sistema operacional do computador *mestre* nos *escravos*, por este motivo o ambiente requer homogeneidade [2], [23]. A interconexão dos nós é feita através de um *switch Fast Ethernet*.

O ambiente OSCAR provê uma solução para configuração de *clusters*, incluindo desenvolvimento de um conjunto de ferramentas para instalação e administração do mesmo.

Um dos fatores considerados para escolha do OSCAR foi por possuir 23% de aceitação por parte da comunidade de desenvolvedores de *clusters*. Além disso, o ambiente possui código aberto, o que possibilita a adequação de configurações às necessidades do *cluster*.

Dentre as ferramentas que compõem o OSCAR, serão caracterizadas as que desempenham funções de: apoio à administração e gerenciamento (*Cluster Command Control - C3*), colaboração para atuar como computador paralelo (*Local Area Multicomputer/Message Passing Interface - LAM/MPI*) e cooperação para envio de processos a múltiplos nós (*Portable Batch System - PBS*). Cabe salientar que estas ferramentas foram utilizadas especificamente na configuração do *cluster* criado, por causa da maior documentação existente. O ambiente OSCAR não impõe o uso das ferramentas citadas, permitindo-nos optar pelo uso de outras alternativas para, por exemplo, monitoramento e gerenciamento de processos.

4. GRIDS COMPUTACIONAIS

Em um ambiente *grid* a alocação de recursos é realizada por administração descentralizada, onde cada organização controla seus próprios recursos aplicando políticas conforme sua demanda de utilização [2], [24].

A possibilidade de heterogeneidade das arquiteturas e sistemas operacionais empregados é outra característica típica de um *grid* [7], [25], [26]. Um grande desafio é obter uma forma de encapsular estas diferenças sem comprometer a boa performance. A infra-estrutura deste ambiente deve permitir o acesso consistente aos recursos através de serviços padronizados, com interfaces e parâmetros muito bem definidos. Sem estas normas, os serviços prestados podem ser ineficazes. Entretanto, apesar de não existir uma formalização mundial, existem dois modelos [21], [27], [28], [29], [30] apresentados na figura 2, que têm boa aceitação na comunidade científica e que podem se tornar padrões (*de facto* ou *de direito*) de uma arquitetura de *grid* [31].

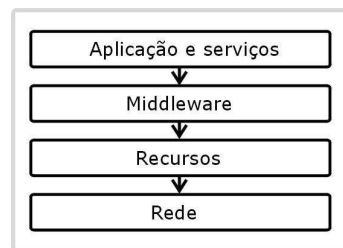


Figura 2(a)

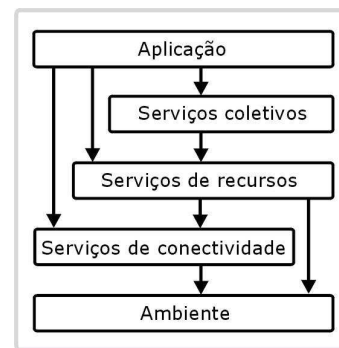


Figura 2(b)

Figura 2 – Modelos de arquitetura *grid*

Na figura 2(a) ilustramos um modelo de *grid* que deve ser compreendido da seguinte maneira:

a) *Aplicação e serviços*

Camada formada por pacotes de *software* de aplicação, incluindo os conjuntos de ferramentas de desenvolvimento e serviços. A porção de serviço deve prover diversas funções de gerenciamento, tais como, faturamento, contabilidade e medidas de métricas utilizáveis – todos parâmetros importantes para o uso virtual dos recursos compartilhados entre diferentes usuários, departamentos e empresas.

b) *Middleware*

Esta camada deve fornecer protocolos que permitam múltiplos elementos (servidores, ambientes de armazenamento, redes, dentre outros) participarem de um ambiente de *grid* unificado. Diversos protocolos e funções devem existir nesta camada no intuito de fornecer suporte aos elementos heterogêneos de uma configuração de *grid*, adaptando-se aos diferentes sistemas operacionais, sistemas de arquivos e protocolos de comunicação.

c) Recursos

Esta camada é constituída pelo conjunto de recursos que fazem parte de um *grid*, incluindo servidores primários e dispositivos de armazenamento. Podemos citar como exemplos de configurações os *clusters* (por exemplo, as *Nows*), os serviços de armazenamento e os computadores especiais (por exemplo, os supercomputadores).

d) Rede

Compõe a base da conectividade para os recursos de um *grid*. Assim, podemos imaginar os *switches*, roteadores e a infraestrutura das redes de comunicação, tais como Sonet/SDH/DWDM.

Uma outra arquitetura do modelo *grid*, mais genérica, representada na figura 2(b), é composta por cinco níveis entendidos como:

a) Aplicação

Compreende as aplicações dos usuários que operam no ambiente da organização virtual, incluindo bibliotecas de funções, que usufruem os serviços prestados pelas demais camadas.

b) Serviços coletivos

Enquanto a camada de recursos destina-se as interações feitas em recursos individuais, esta camada define protocolos globais que se ocupam das interações entre coleções de recursos. Os componentes dessa camada baseiam-se nos níveis de recursos e de aplicação, que implementam uma grande variedade de serviços, tais como: serviços de diretório, monitoração e diagnóstico, replicação de dados, gerenciamento de carga de trabalho e descoberta de recursos, autorização, verificação e colaboração.

c) Serviços de recursos

Esta categoria corresponde à definição dos protocolos e APIs que fornecem segurança na negociação, iniciação, monitoramento, controle e na contagem dos recursos compartilhados. As implementações destes protocolos chamam as funções do nível de ambiente para acessar e controlar os recursos locais. Duas classes de protocolos podem ser distinguidas neste ponto: os protocolos de informação e os protocolos de gerenciamento.

d) Serviços de conectividade

Nível que define os protocolos básicos de comunicação e autenticação necessários para as transações de rede específicas do *grid*. Os protocolos de comunicação permitem a troca de dados entre os níveis de ambiente e recursos, sendo atribuídos serviços como transporte e roteamento. Os protocolos de autenticação possibilitam verificar a identidade de usuários e recursos, com a aplicação de métodos de autenticação e criptografia.

e) Ambiente

Esta camada tem a finalidade de traduzir as primitivas de compartilhamento de alto nível nas operações específicas de cada recurso como resultado das operações de compartilhamento nos níveis superiores. Deve incluir a implementação de mecanismos de negociação que obtenham informações sobre a estrutura, o estado e as possibilidades

dos recursos, e ainda, mecanismos de gerenciamento de recursos que forneçam meios para monitoramento da qualidade de serviço.

Os recursos disponíveis em um *grid*, tipicamente, são da ordem de milhares, dedicados ou não, distribuídos de forma global e ligados por uma rede de interconexão que apresenta características bem diversas das normalmente empregadas em *clusters*, por exemplo. Por estas razões, é necessário estabelecer interconexões para acesso aos recursos e monitoramento e controle dos mesmos, através de condições adequadas de *hardware* e *software*, respectivamente [32].

5. CONCLUSÕES

Nos últimos anos, temos verificado uma mudança na execução de um grande número de aplicações. Outrora era comum a utilização dos computadores paralelos na execução de trabalhos complexos. Atualmente, na grande maioria das vezes, é convencional a utilização de computadores pessoais para a configuração de infraestrutura de ambientes de alto-desempenho que podem ser caracterizados como os *clusters* e os *grids*. Todavia, é reconhecida a necessidade de políticas de utilização dos recursos para que estes ambientes sejam considerados eficazes.

Basicamente, as diferenças entre *clusters* e *grids* são o que definem em uma organização como utilizar um tipo de configuração ou outro. Como *grids* têm a característica de expandir geograficamente por uma extensa área, há a necessidade de planos de execução para o gerenciamento de variadas configurações, potencialmente compostas por agrupamentos de computadores em *cluster*. Estas estratégias tendem para um balanceamento de carga mais eficaz, considerando recursos múltiplos em cada nó. Nem toda aplicação é adequada para executar em um *cluster* ou *grid*. Alguns tipos de aplicações simplesmente não podem ser paralelizadas. Para determinadas soluções, pode levar uma quantia grande de trabalho para alcançar um índice de processamento requerido. A configuração de um *grid* pode afetar o seu desempenho e a confiabilidade, influenciando na segurança de uma organização. Por todas estas razões, é importante verificarmos que ocorre uma evolução gradativa quanto a novas propostas de algoritmos e novos recursos de *hardware* que visam o alto desempenho.

Atualmente, sistemas em *grid* ainda estão em fase inicial quanto à otimização do gerenciamento e distribuição do processamento. Futuramente veremos produtos que tornarão o trabalho do administrador mais fácil, automatizando decisões, incluindo a identificação de problemas em tempo real com ações de correção e restauração da normalidade de funcionamento antes que problemas prejudiquem seriamente o sistema. Além disso, incluirão a manipulação de dados interoperáveis e que alcançarão melhor desempenho, integração, confiabilidade e capacidade.

Neste artigo, foi apresentado um estudo envolvendo ambientes de alto-desempenho e as peculiaridades de *cluster* e *grid* computacionais. Pode-se verificar que são formas de processamento complementares, ou seja, é possível construir um ambiente do tipo *grid* no qual é inserida uma ou mais configurações de *cluster*. Além disso, são soluções viáveis no sentido custo/benefício para obter um alto desempenho de aplicações. Contudo, esses são ambientes que tem muito a serem explorados, para que se tenha uma padronização e, conseqüentemente, maior abrangência de utilização em organizações.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] M. A. R. Dantas, “Tecnologias de Redes de Comunicação e Computadores”, 1ª ed., Rio de Janeiro: Axcel Books, 2003.
- [2] R. Buyya, “High Performance Cluster Computing: Architectures and Systems”, USA: Prentice Hall, 1999.
- [3] M. Baker, “Cluster Computing Trends,” Physics Seminar, Liverpool University, 2000.
- [4] A. S. Tanenbaum, “Sistemas Operacionais Modernos”, Rio de Janeiro: Livros Técnicos e Científicos Editora, 1999.
- [5] M. A. R. Dantas e A. Hosken, “Computação Oportunista de Alto-Desempenho: Características, Desafios e Performance”, II Encontro de Ciência e Tecnologia de Lages, 2003.
- [6] I. Foster, C. Kesselman, S. Tuecke, “The Anatomy of the Grid: Enabling Scalable Virtual Organizations”, International Journal of Supercomputer Applications, 2001.
- [7] D. B. Skillicorn, “Motivating Computational Grids”, 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, 2002.
- [8] I. Foster, C. Kesselman, J. Nick and S. Tuecke, “The Physiology of the Grid - An Open Grid Services Architecture for Distributed Systems Integration”, Global Grid Forum, 2002.
- [9] R. Buyya, D. Abramson, and J. Giddy, “Nimrod/G: An Architecture of a Resource Management and Scheduling System in a Global Computational Grid”, 4th Intl. Conf. on High Performance Computing in Asia-Pacific Region, 2000.
- [10] C. Ernemann, V. Hamscher, U. Schwiegelshohn, R. Yahyapour and A. Streit, “On Advantages of Grid Computing for Parallel Job Scheduling”, 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, 2002.
- [11] B. Jacob, L. Ferreira, N. Bieberstein, C. Gilzean, J. Girard, R. Strachowski and S. Yu, “Enabling Applications for Grid Computing with Globus”, USA: IBM Redbooks, 2003.
- [12] L. Ferreira, V. Berstis, J. Armstrong, M. Kendzierski, A. Neukoetter, M. Takagi, R. Bing-Wo, A. Amir, R. Murakawa, O. Hernandez, J. Magowan and N. Bieberstein, “Introduction to Grid Computing with Globus”, USA: IBM Redbooks, 2002.
- [13] I. T. Foster, C. Kesselman, G. Tsudik, and S. Tuecke, “A Security Architecture for Computational Grids”, ACM Conference on Computer and Communications Security, 1998.
- [14] W. Allcock, J. Bester, J. Bresnahan, A. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, and S. Tuecke, “Data Management and Transfer in High-Performance Computational Grid Environments”, Parallel Computing, 2001.
- [15] D. Feitelson, “Metric and Workload Effects on Computer Systems Evaluation”, IEEE Computer Society Press, 2003.
- [16] D. Zagorodnov, F. Berman, and R. Wolski, “Application scheduling on the information power Grid”, Technical Report CS2000-0644, 1998.
- [17] I. Foster and C. Kesselman, “Globus: A Metacomputing Infrastructure Toolkit”, The International Journal of Supercomputer Applications and High Performance Computing, 1996.

- [18]H. Casanova, M. Kim, J. S. Plank, and J. J. Dongarra, "Adaptive Scheduling for Task Farming with Grid Middleware", The International Journal of High Performance Computing, 1999.
- [19]N. G. Shivaratri, P. Krueger, and M. Singhal, "Load Distributing for Locally Distributed Systems", IEEE Computer Society Press, 1992.
- [20]T. L. Casavant and J. G. Kuhl, "A Taxonomy of Scheduling in General-Purpose Distributed Computing Systems", IEEE Transactions on Software Engineering, 1988.
- [21]M. Pitanga, "Computação em Cluster", 1ª ed., Rio de Janeiro: Brasport, 2004.
- [22]C. Rista, A. R. Pinto e M. A. R. Dantas, "OSCAR: Um Gerenciador de Agregado para Ambiente Operacional Linux", 4ª Escola Regional de Alto Desempenho, 2004.
- [23]T. A. Colvero, A. M. Pernas e M. A. R. Dantas, "Facilidades para Gerenciamento de uma Configuração de Agregado", 4ª Escola Regional de Alto Desempenho, 2004.
- [24]V. Berstis, "Fundamentals of Grid Computing," REDP3613, Redbooks Paper IBM, Nov. 2002.
- [25]E. Heymann, M. A. Senar, E. Luque and M. Livny, "Adaptive Scheduling for Master-Worker Applications on the Computational Grid", First IEEE/ACM International Workshop on Grid Computing, 2000.
- [26]M. A. R. Dantas and A. Hosken, "The Atha Environment: Experience With A User Friendly Environment For Opportunistic Computing", 18th International Symposium on High Performance Computing Systems and Applications, Winnipeg, 2004.
- [27]I. Foster and C. Kesselman, "The Grid: Blueprint for a New Computing Infrastructure", San Francisco: Morgan Kaufmann Publishers, 1999.
- [28]F. Berman, G. Fox and T. Hey, "Grid Computing: Making The Global Infrastructure a Reality", England: John Wiley & Sons, 2003.
- [29]A. Alvarenga e M. A. R. Dantas, "Um Ambiente para Processamento Paralelo Oportunístico na Internet", Dissertação de Mestrado, Ciência da Computação, UNB, 2003.
- [30]M. A. R. Dantas, L. Brasil and N. Allemand, "An Evaluation of Globus and Legion Software Environments", High Performance Computing Systems and Applications and OSCAR Symposium, 2003.
- [31]S. Tuecke, K. Czajkowski, I. Foster, J. Rey, F. Steve, and G. Carl, "Grid Service Specification", Technical Report in Globus Project, 2002.
- [32]R. Buyya, D. Abramson, J. Giddy and H. Stockinger, "Economic Models for Resource Management and Scheduling in Grid Computing", The Journal of Concurrency and Computation: Practice and Experience, 2002.