



GRIDS COMPUTACIONAIS

-Revisão Bibliográfica-

FABIANE TAGLIARI¹; ELIAS CESAR ARAÚJO DE CARVALHO²

¹ Tecnologia em Sistemas de Informação, Integrado Colégio e Faculdade (CEI). Av. Irmãos Pereira, 670, CEP: 87.301-010 – Campo Mourão – PR – Brasil. E-mail: fabianetagliari@grupointegrado.br.

² Departamento de Informática, Centro Universitário de Maringá (CESUMAR). Av. Guedner, 1.610, CEP: 87.050-900 – Maringá – PR – Brasil. E-mail: elias@cesumar.br.

RESUMO

Os grids computacionais surgiram da idéia de se compartilhar recursos de computadores que estejam ociosos. Desta forma, houve o empenho de alguns pesquisadores em trabalhar com a computação distribuída geograficamente. A idéia dos grids é justamente poder utilizar o poder de processamento, armazenamento e outros recursos computacionais ao redor do mundo. Assim, tem-se alguns projetos com estes objetivos, e dentre eles estão o Condor, Legion, Set@home, MyGrid, OurGrid. Uma vez que Grid é uma nova área de pesquisa, o objetivo deste artigo é mostrar algumas características sobre os grids computacionais, e apresentar alguns projetos importantes nesta área.

Palavras-chave: grids; computação distribuída

COMPUTATIONAL GRIDS

ABSTRACT

Computational Grids appeared from the idea of sharing resources of idle computers. So, there was a persistence of some researchers in working with computes geographically distributed. The idea of grids is just to be able of use the processing power, storage and other computational resources around of the world. Then, there are some projects with these objectives and amongst them are: the Condor, Legion, Set@home, MyGrid, OurGrid. Once Grid is a new research area, the objective of this article is to show some characteristics about computational grids and present some important projects in this area.

Key-Words: grids; computing distributed

INTRODUÇÃO

A rede mundial de computadores, conhecida atualmente como “Internet”, surgiu durante a guerra fria (Estados Unidos e União Soviética). Foi criada pelo Departamento de Defesa (DoD) para estabelecer a liderança dos EUA na ciência e na tecnologia militar. A partir daí, foi se desenvolvendo até chegar ao atual conceito, ou seja, uma rede mundial composta de redes de computadores que utilizam o protocolo de comunicação comum TCP/IP (*Transmission Control Protocol and Internet Protocol*) como padrão (LYNCH, 1993).

A rede não é imutável e vem se desenvolvendo e evoluindo por meio de um processo complexo que envolve altas tecnologias, como protocolos e equipamentos que suportam processamento em massa.

Nessa constante busca para atender a demanda de informação em grande

quantidade, e que proporcione um resultado rápido, surgiram os clusters de computadores, também conhecidos como aglomerados ou agrupamento de computadores.

A primeira base de clusters surgiu da pesquisa de Donald Becker e Thomas Sterling junto ao CESDIS (Centro de Excelência em Dados Espaciais e Informações Científicas), onde construíram uma rede de computadores consistindo em 16 computadores Intel 486 DX4, rodando o sistema operacional Linux, cujo desempenho era de 4,5 Mflops, todos com placa mãe SIS 471 com cache de 256Kb, memória RAM de 16Mb, disco rígido de 540Mb IDE, 2 placas de rede 10Mbps (10baseT ou 10base2), chegando a um desempenho total de 60 Mflops, conectados por uma interface de rede Ethernet denominando-a de Beowulf

Os pesquisadores tinham como objetivo determinar a aplicação de computadores

paralelos para resolução de problemas envolvendo grandes quantidades de dados. Foi a partir daí que surgiram os clusters de computadores para auxiliar em aplicações que exigiam alto desempenho, com o objetivo de dividir um processamento entre os diversos processadores que se encontravam ociosos. Nesse sentido, Pitanga (2003) define *cluster* como “um conjunto de computadores autônomos e que interligados comportam-se como um único sistema do ponto de vista do usuário”. Dessa forma, pode-se argumentar que todos os aspectos relativos à distribuição de tarefas, comunicação, sincronização e organização física do sistema devem ser transparentes ao usuário.

Porém, devido aos altos custos que envolvem a manutenção dos clusters, começaram a realizar novas pesquisas em busca de tecnologias que possibilitassem um resultado tão eficiente quanto os clusters, mas que tivessem custos menores, chegando assim aos Grids Computacionais, que em português, significa Grades Computacionais.

O *Grid Computing* é um novo conceito que explora as potencialidades das redes de computadores, provendo um ambiente computacional de alto desempenho por meio de compartilhamento de recursos geograficamente distribuídos (FOSTER, 2003). Sendo assim, este artigo tem como objetivo definir e apresentar as funcionalidades de um Grid Computacional, demonstrando sua aplicação, arquitetura e seu funcionamento. Este texto está organizado em 4 sessões, distribuídos da seguinte maneira: esta sessão apresenta uma breve introdução ao assunto, seguido da definição de Grid e apresenta o objetivo do artigo. Na sessão 2, faz uma revisão sobre a arquitetura e funcionamento dos grids, e na sessão 3 estão demonstradas algumas referências sobre grids. Finalmente a sessão 4 traz conclusão e trabalhos futuros.

ARQUITETURA E FUNCIONAMENTO DE GRIDS

Para uma maior compreensão sobre o tema abordado, é importante definir o que vem a ser um Grid Computacional, assim, os Grids Computacionais são uma evolução dos sistemas distribuídos das décadas de 80 e 90, e tem uma proposta de infra-estrutura de computação distribuída para cientistas e engenheiros em termos mais avançados.

Ian Foster traduz os conceitos de Grids de duas formas clássicas: “Compartilhamento

de recursos coordenados e resolução de problemas em organizações multi-institucionais dinâmicas” e “Grids são sistemas de suporte à execução de aplicações paralelas que acoplam recursos heterogêneos distribuídos, oferecendo acesso consistente e barato aos recursos, independente de sua posição física”. A tecnologia de *Grids Computing* possibilita agregar recursos computacionais variados e dispersos em um único “supercomputador virtual”, acelerando a execução de várias aplicações paralelas (RABELLO, 2005).

Segundo Laszewski (2005) Grid Computacional é uma coleção de recursos heterogêneos e distribuídos possibilitando que sejam utilizados em grupo para executar aplicações em larga escala.

Desta forma, pode-se dizer que um ponto importante em Grids é o compartilhamento de recursos, além do processamento e acesso a dados. Assim, os Grids Computacionais são ambientes que somam recursos heterogêneos, que estão distribuídos e conectados através de redes de longa distância.

O termo Grid Computacional é uma comparação com a rede de energia elétrica. Sabe-se que a rede elétrica nos fornece energia e não nos importa a forma como ela é gerada, ou como a sua infra-estrutura está disposta, o que nos interessa realmente, é a existência dela. Essa também é a proposta de um Grid Computacional, ou seja, atender as necessidades computacionais dos usuários abstraído os detalhes de como o serviço é realizado.

Cirne (2005) oferece um conceito mais técnico para Grid, dizendo que deve-se pensar nele como uma plataforma para execução de aplicações paralelas, e segundo a visão que se tem sobre o tema como uma metáfora a rede elétrica. Assim, pode-se entender Grids, como um conjunto de recursos heterogêneos e distribuídos, que possibilitam alocar recursos (ex. processador, memória), e executar aplicações em larga escala.

Origem e conceito

Grids Computacionais surgiram da comunidade de Processamento de Alto Desempenho (PAD), e hoje fazem parte da estratégia de empresas como a HP/Compaq, Fujitsu, Sun, IBM. O Grid Computacional pode ser visto através de uma metáfora com a rede

elétrica. Essa nos fornece energia elétrica sob demanda e não se tem acesso a algumas informações que são relevantes para seu funcionamento, como por exemplo, sua origem ou mesmo sua complexidade de transmissão e distribuição. Esses detalhes ficam totalmente transparentes para o usuário, sendo importante para esse, apenas que a energia esteja disponível ao conectar seu aparelho elétrico em uma tomada (CIRNE, 2005).

O Grid Computacional pode ser entendido como uma rede transparente, onde o usuário se conecta para obter benefícios computacionais sob demanda. Esses recursos ficam previamente cadastrados pelas organizações virtuais como Universidades, Centros de Pesquisa e outros, sendo que para utilizá-los há um custo definido pelas mesmas.

Mas o Grid pode também ser formado dentro de uma organização, e um bom exemplo é o banco de dados Oracle 10g. Ele utiliza a Computação em Grade para dividir entre os computadores da rede, a carga de processamento que pode ser gerada. Além disso, caso ocorra falha entre um dos computadores, automaticamente a carga é distribuída entre os outros restantes na rede. Nesse caso, ele calcula como será distribuído o processamento entre os computadores que estão em funcionamento e realiza a divisão para obtenção de um melhor aproveitamento da rede.

Segundo Dantas (2005), alguns autores conceituam Grids como:

“Um ambiente computacional distribuído paralelo que permite o compartilhamento, a seleção, a agregação de recursos autônomos e geograficamente distribuídos. Estas operações e recursos podem ser utilizados durante a execução de uma aplicação dependendo de sua disponibilidade, capacidade, desempenho e custo. O objetivo é prover aos usuários serviços com os requisitos de qualidade corretos para o perfeito funcionamento de suas aplicações”.

Neste sentido, pode-se dizer que os Grids são distribuídos geograficamente, podem chegar a uma escala mundial, possuem componentes heterogêneos, cujo objetivo é o compartilhamento de recursos, bem como de serviços, além de serem controlados por diferentes entidades.

Mas para que seu funcionamento seja eficiente, deve prover um serviço que deixe a

Grid dinâmica, possuindo uma arquitetura que possa crescer à medida que mais sistemas se incorporem aos existentes, busque a auto configuração sempre que um determinado nó se torne indisponível, seja temporariamente ou porque sua organização saiu da Grid.

E é na busca de aperfeiçoamento que cada dia mais pesquisadores tem interesse em padronizar um modelo para os Grids, pois é uma tecnologia que vem ganhando espaço na informática com a possibilidade de interação com equipamentos que estejam disponíveis na rede para que se tenha um benefício, seja ele processamento, armazenamento ou outros.

Arquitetura

Hoje, ainda não existe uma topologia ou mesmo arquitetura pré-definida existente para os Grids Computacionais, porém pesquisadores e técnicos estão buscando estabelecer certos padrões visando o estabelecimento da homogeneidade entre as organizações virtuais. Essa padronização vai desde mecanismos de autenticação, até o compartilhamento de recursos.

Segundo Pitanga (2003) o objetivo da padronização da topologia e arquitetura não é determinar todos os protocolos e serviços, mas identificar requisitos para classes genéricas de componentes.

Baseado em Foster (1999), para uma futura padronização dos Grids Computacionais, é apresentado um modelo e na seqüência, será abordado um pouco sobre cada camada em que a arquitetura foi dividida.

a) Aplicação: compreende aplicações dos usuários que trabalham no ambiente da organização virtual. Os outros níveis provêm serviços necessários às aplicações desenvolvidas que as solicitam;

b) Coletivo: camada responsável pela definição dos protocolos e serviços utilizados para atuar nas interações entre coleções de recursos. Baseados nos níveis recursos e aplicações, os componentes dessa camada implementam serviços como:

- Serviços de diretório;
- Serviços de alocação conjunta e agendamento;
- Serviços de monitoramento e diagnóstico;
- Serviços de pesquisa de software;
- Serviços colaborativos;
- Sistema de programação;

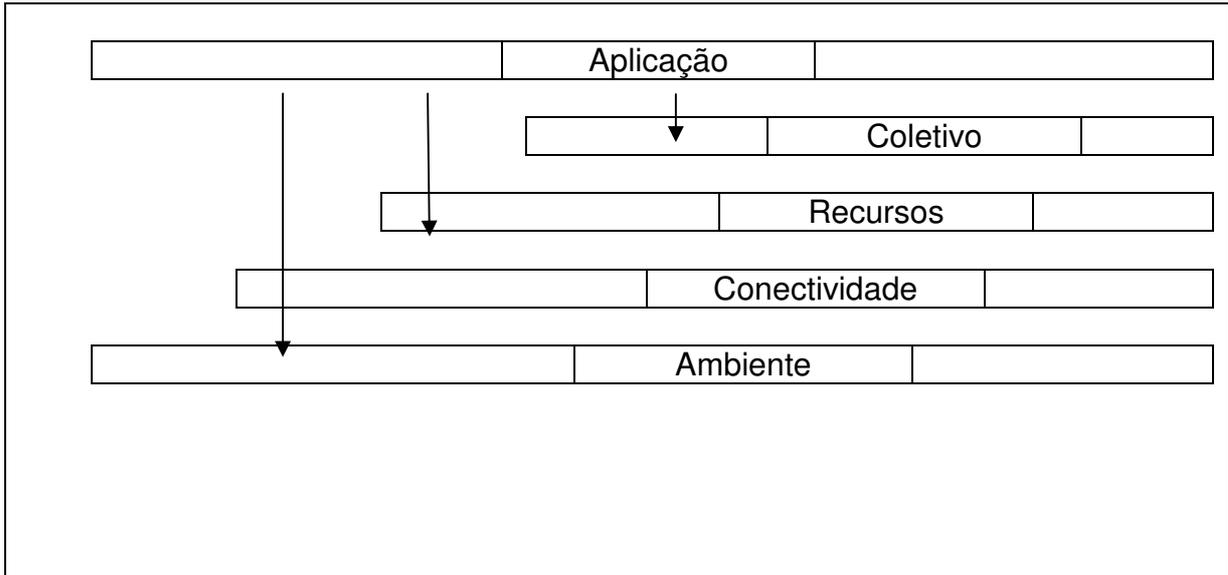


Figura 1. Modelo mais genérico de grids computacionais.

c) Recursos: é nesta camada onde encontra-se a definição dos protocolos utilizados, e são eles: Protocolos de Informação que são usados para obtenção de informações sobre a estrutura e o estado dos recursos compartilhados e Protocolos de Gerenciamento que negociam acesso a recursos compartilhados. As implementações dos protocolos dessa camada chamam as funções da camada de ambiente para acessar e controlar recursos locais;

d) Conectividade: nesta camada são definidos os protocolos básicos de comunicação e autenticação do Grids;

e) Ambiente: nesta camada existe a interface para controle local dos recursos disponibilizados. Devem ser implementados mecanismos que disponibilizem ao usuário informações sobre a estrutura, e as possibilidades de utilização do recurso, bem como mecanismo que possibilitem monitorar a qualidade dos serviços.

O modelo que foi apresentado é o mais genérico, e logo abaixo, fica demonstrado uma outra arquitetura e na seqüência será descrito cada uma das camadas para uma melhor compreensão:

a) Aplicações e Serviços: esta é a camada mais alta de um Grid, que inclui

conjuntos de ferramentas de desenvolvimento, serviços de gerenciamento, contabilidade, formas de compartilhamento e utilização entre os diversos usuários;

b) Middleware: camada que deve fornecer conjunto de bibliotecas, protocolos e serviços para que diferentes elementos heterogêneos sejam padronizados, através de software e domínio;

c) Recursos: nesta camada, tem-se os servidores primários e dispositivos de armazenamento;

d) Redes: este nível é a base para que exista a conexão de um Grid, e nela estão inclusos os switches, roteadores e toda a infraestrutura das redes de comunicação.

Para melhor entendimento dessas camadas, será apresentado uma comparação funcional entre o modelo TCP/IP, que hoje é utilizado na rede Internet, e o primeiro modelo proposto para os Grids. Essa comparação está baseada em Dantas (2005) e Pitanga (2003).

Como as camadas sugeridas para os Grids já foram explicadas, logo abaixo aos modelos, serão descritas apenas aspectos das camadas da Internet como forma de comparativo entre elas.

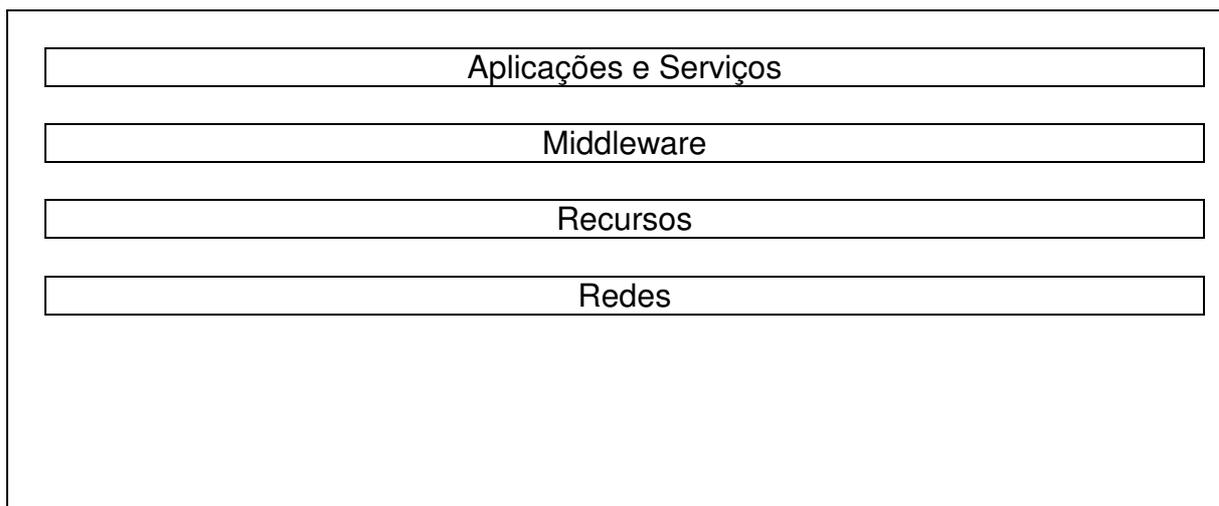


Figura 2. Modelo de Grids computacionais.

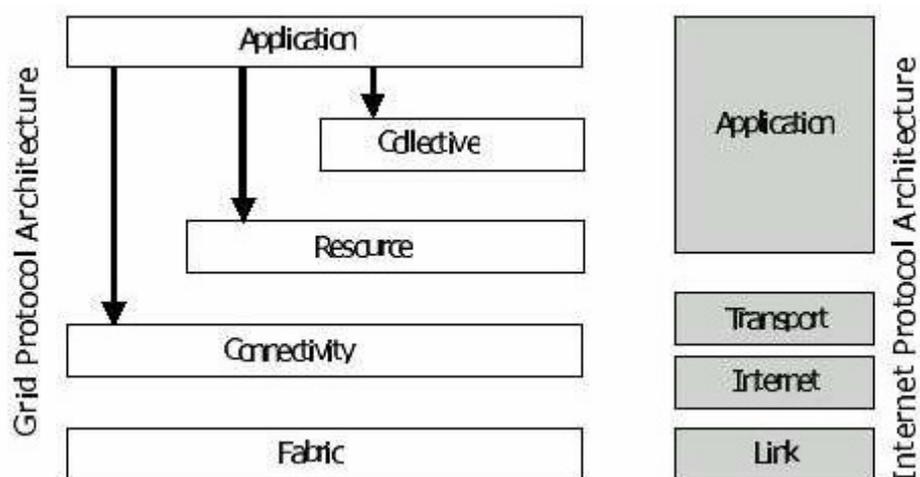


Figura 3. Arquitetura de Grid e a arquitetura TCP/IP.

Referente às camadas de TCP/IP, tem-se:

a) *Application* (aplicação): Na camada de aplicação, tem-se protocolos usados pelos programas aplicativos. Os programas aplicativos passam as mensagens apropriadas ao protocolo de transporte para transmissão;

b) *Transport* (transporte): Nesta camada encontram-se dois protocolos que são responsáveis pelo fluxo de informações e são descritos como: TCP: protocolo de transferência confiável, com conexão e com controle de fluxo; UDP: protocolo de transferência de mensagens não confiável, sem conexão e sem controle de fluxo;

c) *Internet*: Também pode ser encontrado como camada de Rede, e incluem os protocolos: IP que é o protocolo de roteamento, sem conexão; ICMP: é parte do

datagrama IP, protocolo para detecção de erros de rede usado por *hosts* e roteadores;

d) *Link* (enlace): Pode ser encontrada também como camada física, e são os protocolos que recebem e transmitem os datagramas IP através da rede conectada.

Segundo Pitanga (2004), os esforços para uma padronização da tecnologia Grid, começaram a aumentar pelo uso da mesma por inúmeras organizações. Assim, é usual a classificação das topologias dos Grids da seguinte maneira:

a) *Intragrid* – Está dentro de uma rede local, ou seja, utiliza recursos e serviços dispersos, dentro de uma mesma organização. O sistema de interligação se dá através de departamentos, podendo ter seus hardwares e softwares heterogêneos ou não. A configuração *Intragrid*, segundo Dantas (2005), pode ser vista como um cluster de

organizações virtuais, ou seja, tendo uma gerência geral de ambiente único;

b) Extragrid – O extragrid, possui quase o mesmo conceito do Intragrid, ou seja, interligação de redes, porém entre duas organizações, com interação entre os parceiros participantes da configuração, mas pode possuir várias organizações virtuais dentro de cada instituição. Um outro nome conhecido é *Partner Grids*;

c) Intergrid – É formada por várias organizações virtuais, ou seja, abrangem a internet. Cada organização possui um Grid interno, que interage com outras organizações virtuais parceiras possuidoras de Grids internos.

Mas podem existir diferentes visões sobre a arquitetura de um grid, sendo que do ponto de vista de um programador de aplicativos, é possível ter a seguinte arquitetura:

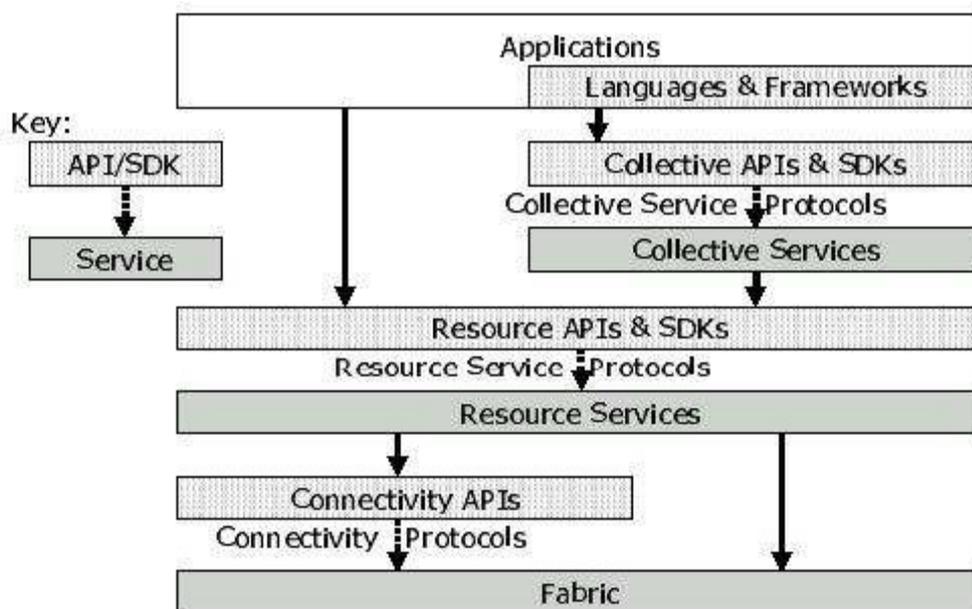


Figura 4. Arquitetura de Grid na visão da programação (FOSTER, 2005).

Um exemplo de esforço para realizar uma padronização dos ambientes grids é:

- Global Grid Fórum (GGF): é uma organização originada pela junção do Grid Forum, do eGRid, European Grid Forum e da comunidade de Grid da Ásia-Pacífico. Em abril de 2002 também se juntou ao GGF o Peer-to-Peer Working Group. Ela é composta por pesquisadores e vários profissionais que possuem grande experiência em sistemas distribuídos, e o objetivo principal é a definição das Interfaces e Protocolos dos grids.

Segundo Taurion (2004) e Pitanga (2004) os atuais grupos de trabalho do GGF são:

- Ambientes de programação (APME – *Applications and Programming Models and Environment*): está ligado as questões referentes ao desenvolvimento de aplicações para ambientes de Grid;

- Arquitetura: visa desenvolver uma arquitetura padrão para os grids;

- Dados: está focado em descobrir, descrever, armazenar e distribuir os dados de forma segura;

- Segurança: como promover uma melhor segurança nas grids;

- Desempenho: preocupação voltada para o desempenho das aplicações rodadas nas grids;

- *Peer-to-Peer* (P2P): visa questões onde cada estação, tem que ter a mesma potencialidade e equivalência na grid;

- Escalonamento (SRM – *Scheduling and Resource Management*): preocupado em escalonar e distribuir programas pela Grid, de uma forma eficiente.

Funcionamento dos Grids

Além da interface, é necessário implementar algumas funcionalidades, que serão abordadas abaixo:

a) Identificação e autenticação: para que o usuário se beneficie dos recursos oferecidos pela Grid, é necessário ter certeza que ele é quem diz ser. Neste caso, nos programas que permitem acesso ao Grid, faz-se a implementação de rotinas de segurança e quando o usuário tenta se conectar o sistema automaticamente solicita sua identificação (*login*) e senha;

b) Autorização e aderência a políticas: depois da identificação e autenticação, é necessário saber se a solicitação feita pelo usuário pode executar em determinado local, melhor dizendo, verificar onde é a melhor plataforma para a execução do trabalho solicitado. Para isso, também são implementadas rotinas que permitem fazer esta checagem;

c) Localização dos recursos: para se executar um *job* (trabalho), é necessário verificar onde os sistemas estão disponíveis para sua execução;

d) Caracterização dos recursos: nem sempre há disponibilidade dos recursos para a execução de um determinado trabalho, desta forma, é necessário fazer uma verificação da disponibilidade do mesmo para a execução de um *job*;

e) Alocação de recursos: quando o trabalho está pronto para ser executado no Grid, ou seja, é encontrado todo recurso disponível para sua execução, é necessário saber se não há um *job* concorrente. Se houver, o sistema deverá prover recursos ao que tem maior prioridade;

f) Contabilização/Billing/Nível de Serviço: não basta apenas a execução de um *job*, é necessário saber se os objetivos da execução foram alcançados;

g) Segurança: é necessário garantir a segurança de um trabalho no Grid, para isso, é quase uma prioridade, registrar e notificar eventuais violações que ocorram.

REFERÊNCIAS SOBRE GRIDS: GRID CAFÉ ANIMATIONS

O Grid Café é um site com muitas informações para quem está iniciando em grids. Destaca-se uma animação em flash que ilustra o funcionamento dos mesmos, onde se pode ter informações tanto em inglês, como em português, sobre o funcionamento dos grids. As informações em português só foram possíveis pela tradução realizada por Tagliari (2005).

Para obter acesso às informações do Grid Café, basta acessar o endereço <http://gridcafe.web.cern.ch/gridcafe/animations-P.html> e ir navegando sobre os itens desejados, conforme apresentado na Figura 5.



Figura 5. Site Grid Café Animations.

CONCLUSÃO E TRABALHOS FUTUROS

O presente artigo teve como objetivo geral, definir e apresentar as funcionalidades de um grid computacional, mostrando as relevâncias destes para a computação distribuída.

Desta forma, mostrou-se a arquitetura dos mesmos. Além disso, ficou demonstrado que apesar do conceito de grids computacionais ser ainda muito novo, utilizava-se este modelo há alguns anos dentro das organizações.

Assim, os pesquisadores tem um desafio: fazer com que os grids funcionem de forma mais transparente e simples ao usuário.

REFERÊNCIAS

CIRNE, W. On the efficacy, efficiency and emergent behavior of task replication in large distributed systems. Disponível em: <<http://walfredo.dsc.ufcg.edu.br/resume.html#publications>>. Acesso em: 02 out. 2005.

DANTAS, M. **Computação distribuída de alto desempenho**: redes, clusters e grids computacionais. Rio de Janeiro: Axcel, 2005.

FOSTER, I.; KESSELMAN, C; TUECKE, S. The anatomy of the grid: enabling scalable virtual organizations. Disponível em:

<<http://www.globus.org/research/papers/anatomy.pdf>>. Acesso em: 16 out. 2005.

FOSTER, I.; KESSELMAN, C. **The grid: blueprint for a new computing infrastructure**. San Francisco: Morgan Kaufmann Publishers, 1999.

LASZEWSKI, G. V. **Grid course**. Disponível em: <<http://wwwfp.mcs.anl.gov/~gregor/grid-iit/talks/Introduction%20to%20Grid%20Computing.htm>>. Acesso em: 02 dez. 2005.

LYNCH, D. C.; ROSE, M. T. **Internet system handbook**. [S.l.]: Addison-Wesley Publishing Company, 1993.

PITANGA, M. **Computação em cluster: o estado da arte**. Rio de Janeiro: Brasport, 2003.

PITANGA, M. **Construindo supercomputadores com linux**. Rio de Janeiro: Brasport, 2004.

REBELLO, V. Definição de grid. Disponível em: <<http://www.ic.uff.br/~cmaci/defineR.htm>>. Acesso em: 01 nov. 2005

TAGLIARI, F.; CARVALHO, E.C.A. **The grid in a flash**. Disponível em: <<http://www.gridcafe.web.cern.ch/gridcafe/animation-P.html>>. Acesso em: 01 nov. 2005.

TAURION, C. **Grid computing: um novo paradigma computacional**. Rio de Janeiro: Brasport, 2004.



Recebido 07 jul. 2006
Aceito 30 nov. 2006