Computação em Grade e uma proposta para protocolos entre grades.

Luiz Gustavo Martins

MAC 5701 - Tópicos em ciência da computação Relatório de estudos

Área de concentração: Computação paralela e distribuída.
Orientador: Marco Dimas Gubitoso

IME – USP – São Paulo – 23 de junho de 2003

Introdução

Computação em Grade surgiu como um campo novo muito importante, distinto da computação distribuída convencional pelo seu foco em compartilhamento de recursos em larga escala, aplicações inovadoras e em alguns casos uma orientação à alta performance.

O termo “Grade” foi usado pela primeira vez em meados de 1990 denotando uma infra-estrutura de computação distribuída para ciência avançada e engenharia. Um considerável progresso foi feito na construção de tal infra-estrutura mas o termo Grade tem sido usado em outros contextos desde trabalho com redes avançadas até inteligência artificial. Podemos nos perguntar se realmente existe um “Problema de Grade“ distinto e portanto a necessidade de uma “tecnologia de Grade”. Se existe, qual a natureza dessa tecnologia e qual seu domínio de aplicação? Enquanto vários grupos têm interesse nos conceitos de Grade e compartilham, em porção significativa, uma visão comum de uma arquitetura de Grade, não se têm ainda um consenso nas respostas para essas perguntas.

O problema real e específico que se esconde por trás do conceito de Grade é compartilhamento de recursos coordenado e solução de problemas em organizações virtuais dinâmicas e multi-institucionais. O compartilhamento com o qual se preocupa não é principalmente relativo a arquivos, mas sim acesso direto a computadores, software, dados e outros recursos, tal como é requerido por uma parcela das estratégias colaborativas de solução de problemas que estão surgindo na industria, ciência e engenharia. Esse compartilhamento é, necessariamente altamente controlado, com provedores de recursos e consumidores definindo claramente e cuidadosamente o que deve ser compartilhado, quem tem direito e em que condições esse compartilhamento ocorre. Um conjunto de indivíduos que e / ou instituições definidas por tais regras de compartilhamento formam o que é chamado de organização virtual (OV).

Um exemplo de organização virtual é: um provedor de servidores de aplicação, um provedor de serviço de armazenagem, um provedor de ciclos e consultores contratados por uma fábrica de carros para fazer a avaliação de um cenário durante o planejamento de uma nova fábrica a ser construída. Outro exemplo seria uma equipe de gerenciamento de crise e os bancos de dados e sistemas de simulação que eles usam para planejar uma resposta para uma situação de emergência. Esses exemplos mostram uma aproximação da solução de problemas baseado em computação colaborativa e ambientes ricos em dados.

Organizações virtuais variam tremendamente em seu propósito, escopo, tamanho, duração e estrutura. Mesmo assim, um estudo cuidadoso dos requisitos da tecnologia envolvida, nos leva a identificar um vasto conjunto de considerações e requisitos comuns.

Em particular, vemos uma necessidade de um alto e flexível compartilhamento de relacionamentos, indo desde cliente-servidor até ponto-a-ponto; para compartilhamento de vários recursos, desde programas, arquivos e dados para computadores, sensores e redes; e para diversos modos de uso, desde monousuário até multiusuário, desde sensível à performance até sensível ao custo e até mesmo abrangendo pontos como qualidade de serviço até escalonamento.

As tecnologias de computação distribuída atuais não tratam dos problemas supra citados. Por exemplo a Internet atual lida com a comunicação e troca de informação entre computadores, mas não provem uma integração com o uso coordenado de recursos em múltiplos sítios computacionais. Tecnologias como CORBA e Java RMI permitem compartilhamento de recursos mas com apenas uma organização. Companhias emergentes de computação distribuída, procuram usar computadores ociosos em uma escala global mas, até agora, só suportam acesso altamente centralizado a esse recurso. Em resumo, a tecnologia atual ou não acomoda toda amplitude dos tipos de recursos ou não provem a flexibilidade e controle de compartilhamento de relacionamentos necessário para se estabelecer uma organização virtual.

É nesse ponto que a tecnologia de Grade entra em ação. Nos últimos cinco anos, pesquisa e esforço em desenvolvimento das pessoas que estudam Grades tem produzido protocolos, serviços e ferramentas que se direcionam aos desafios que aparecem quando se tenta criar organizações virtuais escaláveis. Essas tecnologias incluem soluções seguras que suportam gerenciamento de credenciais e políticas para quando a computação se estende a múltiplas instituições; protocolos de gerenciamento de recursos e serviços que suportam acesso remoto seguro à computação, aos dados e a co-alocação de múltiplos recursos; protocolos de busca de informação e serviços que provem configuração e informação do estado dos recursos, organizações e serviços; e serviços de gerenciamento de dados que localizam e transportam conjuntos de dados entre sistemas de armazenagem e aplicações.

Pelo fato de seu foco estar voltada para o compartilhamento dinâmico de recursos entre organizações, a tecnologia de Grade vem para complementar ao invés de competir com as tecnologias de computação distribuída.

Organizações Virtuais

Para poder melhor definir os requisitos de uma arquitetura de Grade, temos que exemplificar melhor algumas organizações virtuais.

1 – Um consórcio de industrias formado para desenvolver um estudo de praticidade de um novo avião supersônico empreende uma simulação altamente apurada de todo o avião. Essa simulação integra componentes de software proprietário desenvolvido por diferentes participantes, e com cada componente operando nos

computadores desse participante e tendo acesso a banco de dados e outros tipos de dados disponibilizados para esse consórcio por seus membros.

2 – Uma equipe de gerenciamento de crise para responder por vazamentos de produtos químicos utilizando modelos de clima e solo para poder estimar a propagação do vazamento, determinando o impacto causado na população local tal como nos elementos geográficos da região como rios e depósitos de água, criando um plano de limpeza de curto prazo (talvez baseado em modelos de reações químicas), recrutando pessoas para um planejamento e coordenação de evacuação e notificação de hospitais.

3 – Milhares de físicos em centenas de laboratórios e universidades distribuídos por todo o mundo juntam-se para desenvolver, criar, operar e analisar os resultados de um grande detector na CERN (laboratório europeu de física de partículas). Durante a fase de análise, eles juntam seus recursos de computação, armazenagem e rede para criar uma grade de dados capaz de analisar petabytes de informação.

Esses exemplos diferem muito entre si pelo número de participantes, tipo de atividades, duração e escala de interação e até mesmo nos recursos sendo compartilhados. Mas eles também têm muito em comum. Em cada caso os participantes desejam compartilhar recursos para poder concluir uma tarefa. Além disso, o compartilhamento esta além de apenas troca de documentos, isto pode envolver acesso direto a software remoto, computadores, dados, sensores e outros recursos. Por exemplo os membros do consórcio podem prover acesso a software especializado, dados ou até aos seus recursos de computação.

Compartilhamento de recursos é condicional: cada dono de recurso disponibiliza recursos sujeito a regras como quando, onde e o que pode ser feito. Por exemplo os desenvolvedores do avião supersônico podem deixar que os participantes da organização virtual use seus sistemas de simulação apenas para problemas simples. Consumidores de recursos podem também colocar restrições nas propriedades dos recursos que eles irão utilizar. Por exemplo, um usuário pode querer apenas acesso a recurso computacional que esteja certificado como seguro. A implementação de tais restrições requer mecanismos que irão expressar políticas para estabelecer a identidade do consumidor ou recurso, e para determinar quando uma operação esta consistente com os relacionamentos aplicados ao compartilhamento.

Relacionamentos de compartilhamento podem variar dinamicamente no decorrer do tempo, em termos dos recursos envolvidos, a natureza do acesso permitido, e o os participantes para os quais o acesso é permitido. Esses relacionamentos não envolvem necessariamente um conjunto definido de indivíduos, mas ao contrário pode ser definido pelas políticas que governam o acesso ao recurso. Por exemplo uma organização pode habilitar o acesso pra qualquer um que consiga mostrar que é um cliente ou um aluno.

A natureza dinâmica do compartilhamento de relacionamentos significa que requeremos um mecanismo para descobrir e caracterizar a natureza do relacionamento existente em um dado momento. Por exemplo, um novo participante se juntando a uma

organização virtual tem que estar apto a determinar quais recursos ele pode ter acesso, a qualidade desses recursos e as políticas que os governam.

Compartilhamento de relacionamentos normalmente não é só do tipo cliente-servidor, mas também ponto-a-ponto, provedores podem ser consumidores e compartilhamento de relacionamentos pode existir entre qualquer subconjunto de participantes. Compartilhamento de relacionamentos pode ser combinado para uso coordenado entre muitos recursos, cada um pertencente a uma organização diferente. Por exemplo, em uma organização a computação iniciada em um recurso computacional pode subseqüentemente acessar dados ou iniciar uma outra computação em outro lugar. A habilidade de delegar autoridade de forma controlada se torna importante em algumas situações, tal como mecanismos para coordenação de operações através de múltiplos recursos.

O mesmo recurso pode ser usado de formas diferentes, dependendo nas restrições colocadas no compartilhamento e no objetivo do compartilhamento. Por exemplo, um computador pode ser usado apenas para rodar um pedaço especifico de software em um esquema de compartilhamento, enquanto que ele pode ceder ciclos genéricos de computação em outro esquema. Por causa da falta de conhecimento a priori sobre como o recurso vai ser usado, métricas de performance, expectativas e limitações como qualidade de serviço podem ser parte das condições colocadas no compartilhamento do recurso ou na sua utilização.

Estas características e requisitos definem o que definimos como organização virtual, um conceito que se acredita ser fundamental para computação moderna. Organizações virtuais habilitam grupos de organizações ou indivíduos a compartilhar recursos de uma forma controlada, de tal forma que membros possam colaborar e chegar a um objetivo comum.

Descrição da Arquitetura da Grade

O objetivo em descrever uma arquitetura para a Grade não é prover uma enumeração completa de todas os protocolos, serviços, API´s e SDK´s requeridos mas ao contrário identificar requisitos para classes genéricas de componentes. O resultado é uma arquitetura estrutural aberta e extensível na qual podem ser postas soluções para os requisitos chaves da organização virtual. A arquitetura esta organizada em camadas de componentes como visto na figura abaixo:

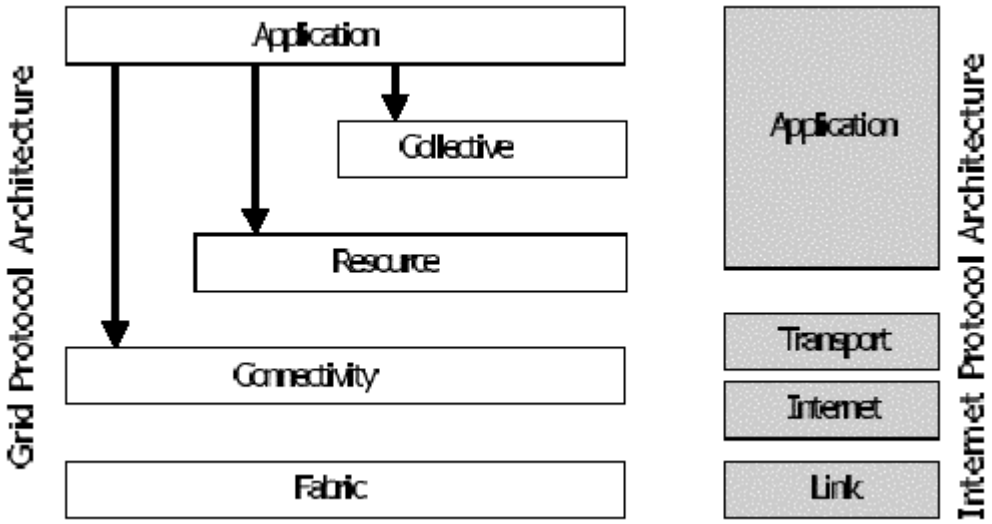

Figura 1: A arquitetura por camadas da Grade e seu relacionamento com a arquitetura de protocolos da Internet. Pelo fato da arquitetura de protocolos da Internet vir desde a rede até a aplicação, existe um mapeamento das camadas da Grade para as camadas da Internet.

Base: Interfaces para controle local

Os componentes da Camada de Base implementam localmente, operações específicas de cada recurso, seja físico ou lógico, como resultado de um compartilhamento de operações em níveis mais altos. Existe então uma interdependência forte entre as funções implementadas no nível da camada Base e as operações de compartilhamento suportadas. Funcionalidades mais ricas da camada Base habilitam operações de compartilhamento mais sofisticadas. Ao mesmo tempo, se exigirmos menos dos elementos da camada Base, a distribuição da infra-estrutura da Grade será simplificada. Por exemplo, suporte ao nível de recursos de reserva antecipada faz com que seja possível para serviços de nível mais alto agregar recursos de formas interessantes que não seriam possíveis de outra forma. No entanto, na prática, poucos recursos suportam reserva antecipada "direto da fábrica", um requisito de reserva antecipada aumenta o custo de incorporação de novos recursos à Grade. A experiência sugere que no mínimo, recursos implementem mecanismos de pesquisa que permitam descobrir sobre sua estrutura, estado e capacidades, por outro lado, mecanismos de gerenciamento de recursos provem algum controle sobre qualidade de serviço. A lista a seguir provém caracterizações das capacidades específicas dos recursos.

- Recursos Computacionais : Mecanismos são necessários para começar programas e para monitorar o controle de execução dos processos resultantes. Funções de pesquisa são necessárias para determinar o hardware e o software característico tal como informações de estado

relevantes tal como carga atual e estado da fila em casos de recursos gerenciados por um scheduler.

- Recursos de Armazenagem : Mecanismos são necessários para pegar e colocar arquivos. Também são necessários para ler e escrever subconjuntos de arquivo e executar funções para selecionar dados.

- Recursos de Rede : Funções de pesquisa devem ser fornecidas para determinar as características da rede e sua carga.

- Repositórios de Código : Esta forma especializada de armazenagem requer mecanismos para gerenciar versão de código e operações de atualização, temos como exemplo o sistema de controle CVS.

- Catálogos : Outra forma especial de armazenagem. Requer implementação de mecanismos de busca pelo catálogo e operações de atualização como por exemplo um banco de dados relacional.

Conectividade: Comunicação fácil e segura

A camada de conectividade define o núcleo dos protocolos necessário a comunicação e autenticação para transações pela rede específicas para a Grade. Protocolos de comunicação habilitam a troca de dados entre recursos da camada Base. Protocolos de autenticação são construídos sobre os serviços de comunicação para poder prover mecanismos criptografados e seguros para se verificar a identidade dos usuários e recursos.

As soluções de autenticação para os ambientes das organizações virtuais devem ter as seguintes características

- Login único : Usuários devem ser capazes de se logar no sistema apenas uma vez, a partir daí ter acesso a múltiplos recursos da Grade definidos na camada Base sem mais nenhuma intervenção do usuário.
- Delegação : Um usuário deve ser capaz de fazer um programa rodar em seu nome, de tal forma que o programa tenha acesso a qualquer recurso que o usuário tenha autorização.
- Integração com soluções de segurança local: Cada sítio ou provedor de recursos pode utilizar uma solução de segurança local, como Kerberos ou o sistema de segurança do Unix. A segurança da Grade deve ser capaz de agir em conjunto com essas várias soluções locais.
- Relações de confiança baseadas no usuário : Para que um usuário possa utilizar um recurso de muitos provedores ao mesmo tempo, o sistema de

segurança não deve requerer que cada um dos provedores de recurso coopere ou interaja entre si configurando o ambiente de segurança. Por exemplo, se um usuário tem o direito a usar os sítios A e B, então o usuário deve ser capaz de usar o sítio A e B ao mesmo tempo sem que os administradores de segurança de A e B interajam.

Recursos : Compartilhando recursos únicos

A camada de Recursos é feita baseada na comunicação e autenticação da Camada de Conectividade para poder definir protocolos (APIs e SDKs) para negociação, monitoramento, controle, pagamento de operações compartilhadas em recursos individuais de forma segura. As implementações dessa camada de protocolos chamam as funções da camada Base para acessar e controlar recursos locais.

Há duas classes principais dos protocolos da camada de Recursos:

- Protocolos de informação: são usados para obter informação sobre a estrutura dos recursos compartilhados e estado de um recurso, por exemplo, sua configuração, carga atual e políticas de uso, como custo.
- Protocolos de gerenciamento: são usados para negociar acesso a recursos compartilhados, especificando por exemplo requisitos de recurso e as operações a serem feitas, como criação de processos ou acesso a dados. Como os protocolos de gerenciamento são responsáveis por instanciar relacionamentos compartilhados, eles devem servir como um ponto de aplicação de políticas garantindo que as operações requisitadas estejam consistentes com a política sobre a qual o recurso vai ser compartilhado.

Coletiva : Coordenando recursos múltiplos

Enquanto a camada de Recurso é voltada para a interação com recursos únicos, a camada seguinte na arquitetura tem protocolos e serviços (e APIs e SDKs) que não estão associadas com nenhum recurso específico, mas ao contrário são globais por natureza e capturam interações entre coleções de recursos. Por essa razão essa camada é chamada de camada Coletiva. Os componentes dessa camada são produzidos sobre a camada de Recursos e de Conectividade. Podem implementar uma ampla variedade de comportamentos de compartilhamento sem colocar novos requisitos nos recursos sendo compartilhados, por exemplo:

- Serviço de Diretório permite que participantes de uma organização virtual descubram a existência e propriedade de recursos. Um serviço de diretório

pode permitir que seus usuários possam pesquisar por recursos pelo nome ou atributos como tipo, disponibilidade ou carga.

- Alocação Conjunta e Agendamento permitem aos participantes da organização virtual requisitarem alocação de um ou mais recursos para um determinado objetivo e agendamento de tarefas nos recursos apropriados.
- Serviços de Monitoração e Diagnóstico suportam a monitoração de recursos atrás de falhas, ataques (detecção de intrusos) ou sobrecarga.
- Serviços de Replicação de Dados dão suporte ao gerenciamento do armazenamento de recursos procurando maximizar a performance no acesso aos dados com respeito a métricas como tempo, custo e confiabilidade.
- Sistema de programação para Grades habilita modelos de programação familiares a serem usados no ambiente da Grade, utilizando vários serviços para descobrimento de recursos, segurança e alocação de recursos. Como exemplo temos implementações de MPI para grades.
- Sistemas de gerenciamento de Carga de Trabalho e Arcabouços de colaboração provem para descrição, uso e gerenciamento passo a passo, assíncrono e de multicomponente de fluxo de trabalho.
- Serviços de Descobrimento de Software descobrem e escolhem a melhor implementação de software e plataforma de execução baseado nos parâmetros do problema sendo resolvido, por exemplo o NetSolve.
- Serviços de autorização de comunidade reforçam as políticas de controle de acesso a recursos da comunidade, gerando permissões para que membros da comunidade possam ter acesso aos recursos da comunidade.
- Serviço de Colaboração suporta troca coordenada de informações com comunidades potencialmente grandes de usuários, tanto de forma síncrona ou assíncrona. Por exemplo o CAVERNSoft.

É importante perceber que enquanto a camada de Recursos tem que ser mais geral por natureza e é amplamente distribuída, a camada Coletiva muda o foco de um propósito geral para um altamente especifica, em alguns casos existindo em apenas algumas organizações virtuais especificas.

As Funções da camada Coletiva podem ser implementadas como serviço persistente, com protocolos associados ou como SDKs com APIs associadas projetadas para serem linkadas com aplicações. Em ambos os casos, sua implementação pode ser sobre a camada de Recursos.

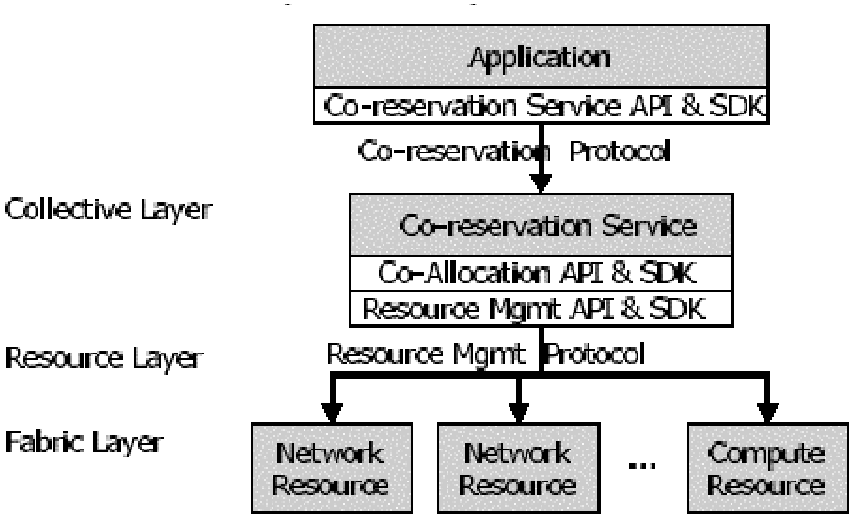

Figura 2: A camada Coletiva e de Recursos, serviços, APIs e SDKs podem ser combinados em uma variedade de formas para poder fornecer funcionalidades às aplicações.

Componentes da camada Coletiva podem ser especificados para se encaixar aos requisitos específicos de uma comunidade, organização virtual, domínio de aplicação, por exemplo, um SDK que implemente um protocolo de coerência específico ou um serviço de reserva conjunta para um recurso de rede específico.

Aplicações

A camada final na arquitetura de Grade compreende as aplicações do usuário que operam no ambiente da organização virtual.

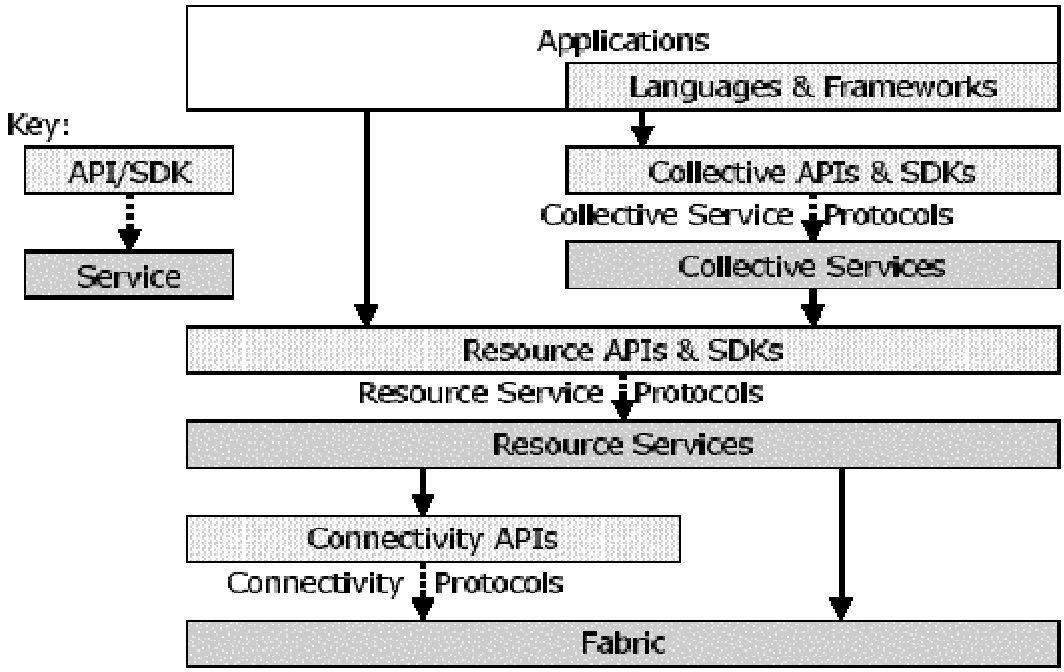

Figura 3: Na figura foi usada apenas uma camada para mostrar a aplicação, que pode ser na prática muito mais complexa, pode chamar arcabouços sofisticados e bibliotecas como SciRun, CORBA, Cactus, que se fossem mostrados na figura iriam deixá-la muitas vezes maior.

A necessidade de Protocolos entre grades

A arquitetura de Grade proposta, estabelece requisitos para protocolos e APIs que habilitam o compartilhamento de recursos, serviços e código. Ela não restringe as tecnologias que possam vir a ser usada para implementar esses protocolos e APIs. Na verdade é bem praticável definir múltiplas instâncias de elementos chave da arquitetura de Grade. Por exemplo, pode-se fazer um protocolo na camada de Conectividade usando tanto Kerberos ou algo baseado em chave publica e acessar esses mecanismos de segurança pela mesma API. No entanto, Grades construídas com esses protocolos diferentes não são interoperantes e não podem compartilhar serviços essenciais, ao menos não sem gateways. Por essa razão, para que se consiga o sucesso da Computação em Grade em longo prazo, é indispensável uma distribuição ampla de um conjunto de protocolos das camadas de Conectividade e Recursos, e numa menor proporção, a camada Coletiva. Tal como o núcleo dos protocolos da Internet habilita redes de computadores diferentes a interoperar e trocar informação, esses protocolos entre grades (Intergrid) habilitam diferentes organizações a interoperar e trocar ou compartilhar recursos. Recursos que “falem” esses protocolos podem ser classificados como estando na Grade. APIs padrões são altamente úteis se código da Grade deve ser compartilhado.

Relação com outras tecnologias

World Wide Web

A ubiqüidade da tecnologia da Web (TCP/IP, HTTP, SOAP, HTML, XML, etc) faz com que seja um ambiente atrativo para a construção sistemas e aplicações de organizações virtuais. No entanto, enquanto essas tecnologias fazem um excelente trabalho suportando interações entre navegadores dos clientes até os servidores web, faltam a eles características necessárias para que possa ocorrer o modelo mais rico de interação que acontecem nas organizações virtuais. Por exemplo, os navegadores atuais normalmente utilizam TLS para fazer autenticação, mas não suportam login único e delegação.

A integração entre as tecnologias de Grade e da Web tem um caminho claro. Por exemplo, o login único, se estendido para TLS, e integrado aos navegadores, poderia permitir login único em servidores Web. As capacidades de delegação poderiam permitir a um navegador delegar uma capacidade para um servidor Web de tal forma que o servidor pudesse agir em nome do cliente. Essas capacidades, poderiam tornar mais fácil o uso de tecnologias da Web para construir portais de organizações virtuais, que proveriam uma interface fina e sofisticada para as aplicações das organizações virtuais.

Sistemas de Computação Distribuída

Desenvolvimento de tecnologias como CORBA, Java Beans, J2EE e DCOM são todos sistemas projetados para habilitar a construção e aplicações distribuídas. Eles provem interfaces de recurso, invocação remota de mecanismos e serviço de troca para descoberta de recursos, tornando mais fácil compartilhar recursos com uma única organização. No entanto, esses mecanismos não resolvem os requisitos específicos listados anteriormente. A forma principal de interação é do tipo cliente-servidor, e não de uso coordenado de recursos múltiplos.

As tecnologias de desenvolvimento de aplicações distribuídas teriam que ser adaptadas aos requisitos da Grade. Por exemplo, poderíamos construir um serviço de Nomes que utiliza o serviço de informação da Grade para procurar fontes distribuídas de informação através de grandes organizações virtuais.

Internet e Computação Ponto-a-Ponto

Computação Ponto a Ponto (Emule, Gnutella, Freenet) e computação distribuída pela Internet (SETI@home [5], ProteinFolding@home) são exemplos mais gerais de programas que estão além do modelo cliente-servidor, e mostram bem o que foi dito como sendo uma organização virtual. Dessa forma eles tem muito em comum com a Grade.

À medida que essas aplicações forem se tornando mais sofisticadas, haverá uma necessidade de interoperarem entre si e então haverá uma convergência das tecnologias ponto a ponto, computação pela Internet e computação em Grade. Por exemplo, login único e delegação se tornam importantes quando serviços de compartilhamento devem interoperar.

Outras Perspectivas sobre Grades Computacionais.

"A Grade é a próxima geração da Internet" – A Grade não é uma alternativa à Intenet, é ao contrário um conjunto novo de protocolos e serviços que adicionam o suporte a criação e utilização de ambientes ricos em computação e dados. Qualquer recurso que esta na Grade, esta também por definição na Internet.

"A Grade é uma fonte de ciclos grátis" – A Grade não implica em acesso irrestrito a recursos. Computação em grade é sinônimo de controle sobre compartilhamento. Os donos dos recursos vão normalmente querer impor políticas que contenham o acesso de acordo com o usuário ou com a habilidade para pagar, etc.

"A Grade precisa de um sistema operacional distribuído" – Como uma das premissas de ampla distribuição e implantação, se fosse necessário um sistema operacional distribuído tornaria a premissa impraticável. O modelo previsto utiliza os protocolos da Internet pois esses já lidam com a maioria dos problemas de rede. A arquitetura proposta é bem simples, define apenas um conjunto compacto de protocolos para que um recurso possa entrar na grade e, além disso, apenas fornece um arcabouço para que muitos comportamentos possam ser especificados.

"A Grade requer novos modelos de programação" – Programar para o ambiente de Grade traz novos desafios não encontrados em ambientes seqüenciais ou até mesmo paralelos, tal como novos módulos de falha e grandes variações em performance. Mas isso pode ser encarado como algo secundário, a programação básica não é fundamentalmente diferente.

“A Grade faz com que super computadores se tornem desnecessários” – Muitos problemas requerem computadores fortemente acoplados com baixa latência e grandes bandas para comunicação. A computação em Grade pode aumentar a demanda por super computadores já que torna o acesso a eles mais fácil.

Projetos de Computação em Grade Existente

A seguir descrevo algumas das características dos sistemas de computação em grade que foram estudados. Existem muitos sistemas em desenvolvimento e até mesmo já prontos, mas nem todos trazem características importantes para serem estudadas.

As camadas de protocolos necessários para a interoperabilidade de grades descrita acima, foram propostas por membros da equipe de desenvolvimento do projeto Globus [2] (Ian Foster, Carl Kesselman, Steven Tuecke [1]), então é natural mencionar esse sistema, mesmo porque ele implementa todos os protocolos mantendo assim uma janela sempre aberta para que outros sistemas de grade possam acessar as grades gerenciadas pelo Globus.

O NetSolve [3] e o Condor [4] utilizam serviços do Globus. O Condor, em especial, tem uma expansão especial para acessar grades Globus, utilizando a API proposta, mostrando que existe uma convergência para um padrão de comunicação entre grades.

Globus

O projeto Globus conduz a pesquisa e o desenvolvimento para criar as tecnologias fundamentais por trás da Grade, a qual deixa as pessoas compartilharem poder computacional, bancos de dados, e outras ferramentas seguramente entre corporações, instituições e barreiras geográficas sem sacrificar a autonomia local.

O projeto do Kit de Ferramentas Globus tem código fonte aberto e inclue serviços e bibliotecas para monitoração, descobrimento e gerenciamento de recursos, tal como segurança e gerenciamento de arquivos. As ferramentas são o centro dos projetos científicos e de engenharia que somam aproximadamente meio bilhão de dólares por todo o mundo, e é a casca do que as maiores companhias em tecnologia da informação estão construindo comercialmente.

Tal como a Internet revolucionou a forma como encaramos as informações, o objetivo do projeto Globus é alcançar um resultado semelhante na computação. Mal se

pode imaginar todos os tipos de aplicação que podem ser feitas se for possível acessar supercomputadores, imagens ao vivo de um satélite e grandes espaços para armazenagem de forma tão fácil quanto acessar a Internet. Essa é a promessa da computação em Grade.

Baseado no laboratório nacional de Argonne, no Instituto da Ciência da Informação da Universidade do Sul da Califórnia e na Universidade de Chicago, o projeto Globus esta organizado em 4 atividades principais: Pesquisa, Ferramentas de Software, Ambiente de Teste e Aplicações.

Pesquisa : estudo dos problemas básicos em áreas como gerenciamento de recursos, segurança, serviços de informação e gerenciamento de dados.

A pesquisa é o centro do projeto Globus. Tem como foco não só questões associadas com a construção de infra-estruturas das grades computacionais, mas também com os problemas que surgem no planejamento e desenvolvimento de aplicações que utilizam serviços de grade.

As áreas de pesquisa atuais são:

Gerenciamento de Recursos: desenvolvimento de mecanismos uniformes e escaláveis para nomeação, localização e alocação de recursos computacionais em sistemas distribuídos.

Gerenciamento e Acesso a Dados: “Acesso a dados distribuídos é tipicamente tão importante quanto acesso a recursos de computação distribuída.”. Aplicações distribuídas cientificas e de engenharia geralmente requerem acesso a grandes quantidades de dados (terabytes ou petabytes). Aplicações futuras previstas pelo time de desenvolvimento do Globus também requerem acesso distribuído a dados, por exemplo, acesso em muitos lugares, por muitas pessoas em ambientes virtuais de colaboração. O esforço em produzir uma grade de dados é para tentar identificar, prototipar e avaliar as tecnologias chave requisitadas para suportar grades de dados para colaboração cientifica e de engenharia.

Ambientes de Desenvolvimento de Aplicações: integração de serviços da grade em arcabouços de desenvolvimento, ambientes e linguagens já existentes, como por exemplo CORBA, Java, MPI, Python.

Serviços de Informação: um trabalho já vem sendo feito nos requisitos, projetos e protótipos de um serviço de informação da Grade, que pode habilitar a configuração dinâmica e adaptações das aplicações.

Segurança: Desenvolvimento de algoritmos de segurança básicos para assegurar comunicação entre grupos, gerenciamento de relacionamentos seguros e desenvolvimento de novos mecanismos para controle granular de acesso.

Ferramentas de Software : O kit de ferramentas Globus esta sendo desenvolvido pelo projeto Globus e por muitos outros pelo mundo. Um número crescente de projetos e companhias está usando as ferramentas Globus para poder utilizar o poder da Grade em seu beneficio. A versão 1.0 do software foi lançada em 1998 e a versão 2.0 saiu em 2002, a versão 3.0 ainda esta da fase beta, mas já está disponível para download.

Ambientes de Teste: auxílio ao planejamento e construção de ambientes de teste em larga escala, tanto para a própria pesquisa do Globus, como para o uso de cientistas e engenheiros. Dentre os ambientes desenvolvidos têm-se:

Grade de tecnologia Nacional (National Technology Grid), feita pela National Partnership for Advanced Computational Infrastructure (NPACI) e a National Computational Science Alliance (NCSA) , que tem como objetivo construir, operar e encorajar o uso de uma grade computacional que ira suportar aplicações científicas e de engenharia e formar a base de uma estrutura computacional de âmbito nacional, da mesma forma que a NSFnet formou a base da Internet de hoje em dia.

Grade de Dados Européia (European Data Grid) , esse projeto europeu tem como objetivos dividir custos de pesquisa e de desenvolvimento de tecnologia. Dentre os seis maiores participantes temos CERN, CNRS, ESRIN, INFN, NIKHEF, PPARC e mais quinze parceiros.

NASA Information Power Grid, tem o objetivo de juntar o poder de supercomputadores e dispositivo de armazenagem das organizações participantes em um ambiente computacional único.

Fora os exemplos citados ai temos o Gerenciador de Recursos Distribuídos ASCI, o GUSTO e o EMERGE.

Aplicações : Desenvolvimento de aplicações de larga escala para usar a Grade e colaboração com cientistas e engenheiros. O objetivo principal do projeto Globus é prover tecnologia básica que possibilite uma classe totalmente nova de aplicações. As aplicações que podem se beneficiar com a tecnologia Globus podem ser classificadas como : supercomputação distribuída, instrumentos espertos, supercomputação em computadores pessoais e video-imersão.

NetSolve

Netsolve é um sistema de cliente servidor que habilita usuários a resolver problemas científicos complexos remotamente. O sistema permite que usuários acessem tanto hardware quanto recursos de software distribuídos pela rede. Netsolve busca por recursos computacionais pela rede, escolhe o melhor disponível, e usando repetidas tentativas para tolerância a falha, resolve o problema e retorna a resposta ao usuário. Uma política de balanceamento de carga é usada pelo sistema do NetSolve para garantir boa performance habilitando o sistema a usar, dentre os recursos computacionais disponíveis, o mais eficiente possível. O arcabouço é baseado na premissa que computação distribuída envolve recursos, processo, dados e usuários, e que o acesso seguro mas mesmo assim flexível para cooperar e comunicar entre as entidades é a chave para a infra-estrutura da metacomputação.

Alguns dos objetivos do projeto NetSolve são:

- facilidade de usar
- uso eficiente dos recursos
- habilidade de integrar componentes de software arbitrários como recursos no sistema.

Já foram projetadas interfaces para Fortran, C, Matlab e Mathematica que podem fazer o usuário acessar e usar o NetSolve com maior facilidade.

Uma das características chave de qualquer sistema de software é versatilidade. Para garantir o sucesso do NetSolve, o sistema tem sido projetado para incorporar qualquer pedaço de software com relativa facilidade. Não há restrições sobre o tipo de software que pode ser integrado ao sistema.

Aplicações que utilizam o NetSolve:

- IPARS (Integrated Parallel Accurate Reservoir Simulators) – é um arcabouço para desenvolvimento de modelos de fluxo de superfície e transporte através de ambiente poroso. A interface NetSolve feita para o sistema IPARS permite que os usuários tenham acesso a toda funcionalidade do sistema acessando via MATLAB, C, Mathematica ou Fortran. O sistema foi ainda melhorado de tal forma que através de um navegador o usuário pode submeter problemas para o sistema IPARS.
- MCell é um simulador da microfisiologia celular. O sistema usa difusão de Monte Carlo e algoritmos de reações químicas para simular interações bioquímicas complexas de moléculas dentro e fora de células vivas.

- CENTS (Collaborative Environment for Nuclear Technology Software): O objetivo desse projeto é desenvolver um ...
- DISP – é um software de processamento remoto de imagens. Usa o NetSolve para poder agregar recursos distribuídos pela Internet formando um sistema único com muito mais poder de processamento.

Integrações do sistema NetSolve

- Ninf – muito parecido com o próprio NetSolve, ambos são sistemas remotos orientados a prover computação numérica. Adaptadores foram desenvolvidos para que um possa usar as rotinas numéricas do outro e vice-versa.
- Legion – foi desenvolvido de tal forma que possibilita que o programa cliente do usuário utilize a interface NetSolve enquanto ainda usa recursos de meta-computador do próprio Legion. O cliente NetSolve utiliza os gráficos de fluxo de dados do Legion para manter as dependências de dados.
- Globus – é usado pelo NetSolve, utiliza serviços para detectar servidores com problemas. Já esta em sua fase de teste o novo cliente NetSolve que implementa um proxy que permite o cliente utilizar a infra-estrutura da grade Globus disponível.
- Condor – atualmente o NetSolve consegue acessar os conjuntos Condor como um conjunto computacional. Com pouco esforço, o servidor pode ser configurado para mandar requisições para um conjunto Condor e pegar os resultados e mandá-los de volta para o cliente

Condor

O objetivo do projeto Condor é desenvolver, distribuir e avaliar mecanismos e políticas para suporte a Computação de Alta Disponibilidade em grandes coleções de recursos computacionais distribuídos. Guiado pelos desafios tecnológicos e sociais desse ambiente de computação, a equipe de desenvolvimento do Condor tem feito ferramentas de software que possibilitam a cientistas e engenheiros aumentarem a disponibilidade de suas computações.

Condor utiliza de recursos computacionais que não estejam sendo utilizados, como ciclos de cpu que são perdidos. Para um código poder ser executado pelo Condor não é necessário recompilar, mas se isso for feito, utilizando as bibliotecas do Condor, o programa poderá fazer checkpoints (pontos de chegada) e fazer chamadas remotas ao próprio sistema. Com a utilização dos checkpoints, se tem um conjunto completo das informações do estado do programa. Tendo essas informações, mesmo que a máquina

que esta executando a tarefa caia, quando ela voltar poderá continuar a tarefa do ponto onde estava a não precisará começar tudo de novo.

Um usuário pode submeter uma tarefa e esta é executada em uma máquina remota dentro do conjunto de máquinas disponíveis ao Condor. Existe um impacto mínimo na segurança da máquina remota e isso é preservado pelo Condor através chamadas remotas ao sistema. Quando uma tarefa faz uma chamada ao sistema, por exemplo para fazer uma operação de entrada ou saída, os dados são mantidos na máquina que requisitou a tarefa.

O sistema Condor pode ajudar pequenas empresas que tem problemas que levam muito tempo para serem resolvidos. Essas empresas podem estar usando algum sistema de execução de tarefas em lotes, ou começam a executar um programa a noite e esperam que na manhã seguinte tenha terminado de rodar. Essas são as situações em que o Condor se destaca. Ele pode economizar tempo e esforço quando o tempo de execução de uma tarefa é mais do que alguns segundos, ou quando um grande número de tarefas deve ser iniciado.

Condor permite a quase qualquer aplicação que rode sem interação do usuário a ser gerenciado. Isto é diferente de sistemas como SETI@Home e ProteinFolding@Home. Esses programas são feitos sobre medida, e normalmente as companhias pequenas não têm recursos suficientes para construir sua própria solução, e nesse caso Condor pode oferecer uma solução geral.

Atualmente o Condor já pode acessar a grade do sistema Globus de forma transparente. Condor –G é uma parte do gerenciador de tarefas do Condor, que permite começar uma tarefa em uma máquina remota utilizando o Kit de Ferramentas Globus. Com isso, os usuários do sistema Condor têm uma janela para a Grade, de tal forma que possa tanto mandar tarefas e gerenciá-las em recursos remotos. Ainda se pode usar o Condor -G para olhar pela Grade e ver na mesma hora como suas tarefas estão se saindo.

Conclusão

Já existem muitos sistemas de grades em uso, mas como é um novo ramo de mercado, todos tentam estabelecer um monopólio. A iniciativa do Globus de propor uma arquitetura para comunicação entre as grades mostra um avanço significativo no sentido de termos grades heterogêneas se comunicando, aumentando ainda mais o poder dessa tecnologia, cuja idéia principal é agregar e não segregar.

Esta para sair uma versão do Globus para Windows, e com isso uma maior distribuição do sistema será possível, chegando assim mais próximo do objetivo que é ter uma ampla distribuição para poder aumentar o alcance da grade Globus. O NetSolve já disponibiliza um cliente Windows, mas sem o código fonte, que traz dlls para serem usadas com o Matlab permitindo acessar o sistema NetSolve. O Condor disponibiliza uma versão igual tanto para sistemas Unix como Windows. Com uma maior

popularização do sistema de grades, haverá também uma maior diversificação nos usuários.

Com uma maior popularização dos sistemas de grade, será possível a qualquer pessoa participar de grades gigantescas, colaborando com a grade compartilhando seus próprios recursos e até mesmo usufruindo desse potencial da Computação em Grade, podendo submeter problemas ou requisitar imagens de satélites ou informações de sensores para suas próprias pesquisas.

Com a diversificação de usuários, novas demandas serão feitas pois esses novos usuários podem ter problemas diferentes para serem resolvidos, aumentando assim a diversidade de problemas que as grades podem ajudar a solucionar.

Bibliografia

[1] Ian Foster and Carl Kesselman and Steve Tuecke, The Anatomy of the Grid, Enable Scalable Virtual Organizations, http://www.globus.org/research/papers/anatomy.pdf

[2] The Globus Project, http://www.globus.org, Scientific Programming, University of Chicago, 1998

[3] Arnold, D. and Agrawal, S. and Blackford, S. and Dongarra, J. and Miller, M. and Seymour, K. and Sagi, K. and Shi, Z. and Vadhiyar, S., Users' Guide to NetSolve V1.4.1, http://icl.cs.utk.edu/netsolve/, University of Tennessee, 2002

[4] Douglas Thain, Todd Tannenbaum, and Miron Livny, "Condor and the Grid", in Fran Berman, Anthony J.G. Hey, Geoffrey Fox, editors, *Grid Computing: Making The Global Infrastructure a Reality*, John Wiley, 2003. ISBN: 0-470-85319-0

[5] Eric Korpela and Dan Werthimer and David Anderson and Jeff Cobb and Matt Lebofsky, http://www.computer.org/cise/articles/seti.htm