Uma Arquitetura P2P Baseada na Hierarquia do Endereçamento IP

Marcos Madruga[1], Thaís Batista[2], Luiz Affonso Guedes[1]

[1]Departamento de Engenharia da Computação e Automação (DCA)
Universidade Federal do Rio Grande do Norte (UFRN)
Campus Universitário – Lagoa Nova – 59.072-970 - Natal - RN

[2]Departamento de Informática e Matemática Aplicada (DIMAp)
Universidade Federal do Rio Grande do Norte (UFRN)
Campus Universitário – Lagoa Nova – 59.072-970 - Natal - RN

madruga@unp.br, thais@ufrnet.br, affonso@dca.ufrn.br

Resumo. *Neste artigo propomos o 'SGrid' (Structured Grid), uma arquitetura hierárquica de rede peer-to-peer (P2P) composta por um espaço bi-dimensional com lados de mesmo tamanho e cujo mapeamento dos nós para o espaço bi-dimensional é baseado no endereço IP. Essa estratégia reflete na rede P2P a proximidade dos nós na rede IP sem a necessidade de troca de informações entre os nós. A arquitetura automaticamente adapta sua estrutura quando nós entram e saem da rede. O SGrid fornece uma operação de pesquisa que executa com tempo determinístico e resolve todas as pesquisas usando O(log N) mensagens para outros nós. Cada nó mantém apenas O(log N) informações de estado. Apresentamos também uma ferramenta que implementa o SGrid e gera gráficos referentes a pesquisa de nós.*

1. Introdução

Redes Peer-to-peer (P2P) fornecem uma infra-estrutura para compartilhamento de recursos e colaboração composta por milhares de nós distribuídos através da Internet que podem continuamente entrar e sair do sistema a qualquer tempo [Aberer 2005]. Tipicamente, redes P2P não possuem controle centralizado nem conhecimento global da estrutura. Um dos principais problemas em redes P2P é localizar o nó que armazena um determinado dado. Como os nós entram e saem do sistema constantemente, dados podem migrar de um nó para outro, de modo que a localização dos mesmos se torna um desafio importante.

Redes P2P podem ser classificadas como *não-estruturadas* e *estruturadas*. Redes *P2P não-estruturadas* utilizam, em geral, algoritmos de busca baseados em técnicas de inundação e não possuem boa escalabilidade. Exemplos desse tipo de rede incluem Napster [Byer 2000] e [Gnutella 2001]. Em redes *P2P estruturadas* a topologia da rede é controlada e os dados são armazenados em localizações específicas de modo a tornar o processo de busca mais simples. Tipicamente, essas redes utilizam algum esquema DHT (*Distributed Hash Table*) [Ratnasamy 2002] que mapeia chaves para nós na rede utilizando um algorimo de *hash*. Nessas redes baseadas em DHT o número de saltos (*hops)* para encontrar uma chave tem um limite superior de O(log N), onde N é o número de nós que compõem a rede. Exemplos de redes P2P escaláveis e estruturadas incluem Chord [Stoica 2001], Tapestry [Zhao 2001], Pastry [Rowstron 2001], e CAN [Ratnasamy 2001].

Embora a maioria das redes P2P que utilizam DHT satisfaça os requisitos de escalabilidade e desempenho, elas não organizam a distribuição dos nós na rede de acordo com sua proximidade física, mas, em geral, definem os nós vizinhos de acordo com o conteúdo armazenado nos nós [Doval 2002]. Diferença de topologia entre a rede lógica P2P e a rede física incrementa o tamanho dos caminhos (*path lengths*) devido ao fato que um salto na rede lógica pode ser equivalente a múltiplos saltos físicos na rede subjacente. Tráfego na rede física e latência são conseqüências dessa estratégia sem *ciência da localização* [Hightower 2001].

Neste artigo, propomos uma rede P2P estruturada chamada *SGrid* que considera a localização física dos nós para organizar sua estrutura lógica hierárquica. O SGrid é composto por um espaço bi-dimensional com lados idênticos. O mapeamento dos nós para o espaço bi-dimensional é baseado na hierarquia do endereçamento IP. Tal arquitetura adapta-se automaticamente quando nós entram ou saem da rede. Como a maioria das redes P2P estruturadas, o SGrid fornece uma operação de pesquisa que executa em tempo determinístico e resolve todas as pesquisas usando O(log N) mensagens para outros nós. O SGrid segue a idéia proposta no CAN (*Content-Addressable Network*) [Ratnasamy 2001] onde um mecanismo de endereçamento escalável suporta inserção e recuperação de conteúdo de modo eficiente.

Evidentemente existem diversas arquiteturas de redes P2P que possuem algumas das características presentes no SGrid, entretanto, nenhuma possui todas as características juntas. O Chord, por exemplo, embora apresente desempenho semelhante, possui os identificadores estruturados em forma de anel e não considera a localização física dos nós. Apesar do CAN também organizar seus identificadores em forma de grade, os nós mantêm apontadores apenas para os vizinhos imediatos o que acarreta buscas menos eficientes que no SGrid, onde cada nó mantém apontadores tanto para nós próximos, quanto para nós distantes. Entre as redes que utilizam grades hierárquicas como no SGrid, o Leopard [Yu 2004] foca na criação de réplicas dos dados e não nos mecanismos de atribuição das chaves aos nós nem no roteamento de mensagens, como o SGrid. Além do Leopard, o GLS [Li 2000] também possui um modelo de grade hierárquica, porém, a seleção dos nós vizinhos é baseada nos identificadores e não em coordenadas fixas da grade, como faz o SGrid. Além dessas características, a forma como o SGrid considera a localização físicas dos nós é baseada apenas nos endereços IPs dos mesmos e não requer troca de mensagens como em outras abordagens [Ng 2002].

Este artigo está estruturado da seguinte forma. A seção 2 apresenta a arquitetura do SGrid e detalha os procedimentos sobre o mapeamento dos nós, alocação de chaves, o algoritmo de pesquisa e os passos realizados quando um nó entra ou deixa a rede. A seção 3 comenta sobre a implementação do SGrid. A seção 4 apresenta alguns trabalhos relacionados. A seção 5 contém as considerações finais.

2. SGrid

A arquitetura do SGrid é composta por um espaço de identificadores representado por uma grade bi-dimensional, para onde os nós e os dados são mapeados. Cada nó é responsável por um conjunto de identificadores e pelos dados que são mapeados para esses identificadores. Uma operação de pesquisa consiste em procurar por um identificador e retornar o nó responsável pelo mesmo.

2.1 Mapeamento dos Nós

O mapeamento dos nós para o espaço bi-dimensional do SGrid é baseado no endereço IP de cada nó. O endereço IP é utilizado para identificar o par (x,y) no espaço de coordenadas para onde o nó é mapeado. Esse ponto representa a chave e o nó responsável por ela.

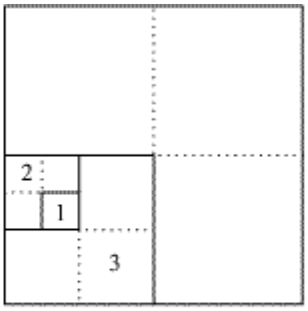
É essencial que o mapeamento dos nós para o espaço de identificadores considere a localização física dos nós, de modo a manter a correspondência entre a distância dos nós na rede P2P e a distância física dos nós na rede IP. Embora qualquer técnica que considere a localização física dos nós possa ser utilizada, como as recentes propostas em Coordenadas Virtuais [Ng 2002], nós desenvolvemos um modelo específico para o SGrid, que se diferencia dos demais por não necessitar que ocorra troca de informações entre os nós e por indicar a posição da grade associada ao nó de forma determinística (um endereço IP é mapeado sempre para o mesmo ponto da Grade). Além de melhorar o desempenho da rede diminuindo a latência na comunicação entre dois nós vizinhos, o esquema proposto facilita a criação de uma arquitetura estendida do SGrid que utiliza um tipo especial de super-nó, chamado *super-peer leve*. Esse nome indica que, nessa arquitetura, os *super-peers* não desempenham a função de roteamento de mensagens, o que permite a criação de uma topologia onde a entrada ou saída de nós não necessariamente causa reconfiguração na rede P2P global. Nesse artigo, entretanto, apresentamos apenas a arquitetura básica do SGrid (sem a utilização de *super-peers*).

O modelo do SGrid é baseado no esquema hierárquico de alocação dos endereços IP que torna o roteamento mais simples uma vez que agrupa rotas para diferentes redes através de uma única rota. Nesse esquema hierárquico, cada novo nível da hierarquia é representado por um conjunto de bits mais à direita no endereço IP e, em geral, os níveis representam as conexões de rede existentes. Seguindo esse esquema, o espaço de identificadores do SGrid é dividido inicialmente em quatro quadrantes e esse processo é repetido recursivamente em cada um deles até que sejam obtidos quadrantes de tamanho um. De acordo com esse modelo, o SGrid é uma grade hierárquica, onde cada nova divisão gera um novo nível da hierarquia. Comparando os níveis do endereço IP com os níveis do SGrid, notamos que o endereço IP de um nó é dividido em m grupos de n bits, onde m é o número de níveis da grade, e n é igual ao número de bits do endereço IP dividido por m. Cada grupo m, da esquerda para a direita, determina o quadrante em cada nível, do maior para o menor (área 1), onde o nó é posicionado. Este número é o resto da divisão do número equivalente aos n bits por 4. O valor resultante determina o quadrante, de acordo com a seguinte regra: 0, representa o quadrante superior esquerdo; 1 representa o quadrante superior direito; 2, representa o inferior esquerdo e 3 representa o quadrante inferior direito.

A Figura 1 ilustra a localização do nó com endereço IP igual a 200.241.86.134 em uma grade com tamanho de lado 256. Uma grade com tamanho de lado 256 possui 8 níveis e como um endereço IPv4 tem 32 bits, conseqüentemente existem 8 grupos de 4 bits cada. A Figura 1a mostra a divisão do endereço IP em grupos e o quadrante equivalente para cada grupo considerando que a representação do endereço 200.241.86.131 é 11001000.11110001.01010110.10000011.

Nível	Bits	Bits (decimal)	Bits (decimal) mod 4	Quadrante
8	1100	12	0	Superior esquerdo
7	1000	8	0	Superior esquerdo
6	1111	15	3	Inferior direito
5	0001	1	1	Superior direito
4	0101	5	1	Superior direito
3	0110	6	2	Inferior esquerdo
2	1000	8	0	Superior esquerdo
1	0011	3	3	Inferior direito

a) Os bits do endereço IP e o mapeamento para os quadrantes.

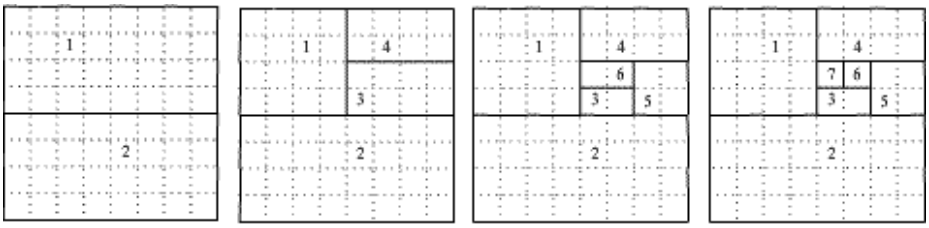


b) Representação gráfica do mapeamento para os niveis1, 2 e 3.

Figura 1: Mapeamento dos nós baseado no endereço IP

2.2 Alocação das Chaves

Conforme explicado na seção anterior, cada nó é associado com uma posição da grade e é responsável pelos dados que são mapeados para essa mesma coordenada. Dados são mapeados para uma coordenada no espaço da grade usando uma função *hash*. Entretanto, como em geral não existe um nó para cada ponto da grade, cada nó é responsável por um conjunto de chaves. O primeiro nó que entra na rede é responsável por todas as chaves. Quando um novo nó entra na rede, um nó existente divide suas chaves com o novo nó. Os seguintes procedimentos são aplicados: (1) inicialmente o nó que terá suas chaves divididas é o responsável pelo ponto da grade para onde o endereço IP do novo nó é mapeado; (2) as chaves são divididas separando-se, em duas partes do mesmo tamanho, a zona que contém as chaves do nó. As zonas precisam ser quadradas ou retangulares, de modo que a divisão deve ser horizontal ou vertical; (3) Se o mapeamento do endereço IP do novo nó não está na nova zona criada pela divisão das chaves, a coordenada associada com o nó é o ponto da nova zona mais próximo do ponto de mapeamento original.



a) Divisão das chaves após a entrada dos nós 1 e 2.

b) Divisão das chaves após a entrada dos nós 3 e 4.

c) Divisão das chaves após a entrada dos nós 5 e 6.

d) Divisão das chaves após a entrada do nó 7.

Figura 2: Divisão do espaço de chaves após a entrada dos nós

A Figura 2 ilustra a divisão das chaves após a entrada de sete nós em uma rede constituída de uma grade com tamanho de lado 8. Os números dentro dos quadrados representam a ordem de entrada dos nós na rede.

2.3 Conexões entre os nós

A grade do SGrid é manipulada por cada nó da rede como uma estrutura hierárquica composta de (Log N)/2 níveis, onde N é tamanho do quadrado (número de chaves). Cada nível é composto de um ou mais quadrados. Cada quadrado é composto de quatro quadrantes, onde um quadrado de nível *n*-1 é um quadrante de um quadrado de nível *n*. Dessa forma, os níveis são formados dividindo-se a grade em quatro quadrantes e repetindo-se esse processo recursivamente para cada um dos quadrantes, até obtermos quadrantes de tamanho 1 (um ponto na grade), que representam o nível 1 da hierarquia. A Figura 3 mostra uma grade com tamanho de lado 8 e três níveis.

Figura 3: Grade com 3 níveis e os nós apontados em cada nível

Quadrantes vizinhos em um nível são quadrantes que compartilham um de seus lados, horizontal ou verticalmente, e que pertencem a um mesmo quadrado do nível que engloba os quadrantes. Isto significa que pertencem ao mesmo quadrante do nível superior. Portanto, cada quadrante de um nível possui dois quadrantes vizinhos: um na horizontal e outro na vertical. Neste artigo, nos referiremos aos quatro quadrantes de um nível como: quadrante 0 é o quadrante superior esquerdo; quadrante 1 é o quadrante superior direito; quadrante 2 é o quadrante inferior esquerdo e quadrante 3 é o quadrante inferior direito.

Cada ponto da grade pertence a um quadrante em cada nível. Cada nó mantém um apontador para um nó em cada quadrante vizinho de todos os níveis. Como existem (Log N)/2 níveis e existem apenas dois vizinhos em cada nível, um na horizontal e outro na vertical, cada nó armazena ((Log N)/2)* 2 informações sobre outros nós da rede.

Os nós dos quadrantes vizinhos de um dado nível para onde um nó mantém apontadores são mostrados na Figura 3. O "x" corresponde à posição de um dado nó e os números, 1,2 e 3, correspondem aos nós apontados por "x" no nível associado com o apontador. O cálculo para determinar os nós apontados é feito da seguinte forma:

- Obtêm-se a coordenada do nó relativa ao início do seu próprio quadrante. Na Figura 3b, que ilustra os vizinhos de nível 2 do nó *x*, vemos que esse nó está na coordenada (1,1) do quadrante 0, considerando-se o quadrado de nível 2.
- Cada nó mantém apontadores para os nós nos quadrantes vizinhos que são responsáveis pelas coordenadas calculadas no passo anterior dentro desses quadrantes. Ainda na Figura 3b, os pontos identificados com *2*, são exatamente as coordenadas (1,1) dentro desses quadrantes.

A Figura 4 mostra a mesma grade da Figura 3, e os nós apontados em cada um dos três níveis, porém visualizando os níveis de forma hierárquica. O nó x da Figura 3 é representado pelo quadrado cinza claro e os nós para os quais esse nó aponta (os pontos marcados com 1, 2 e 3 na Figura 3) são representados em cinza escuro.

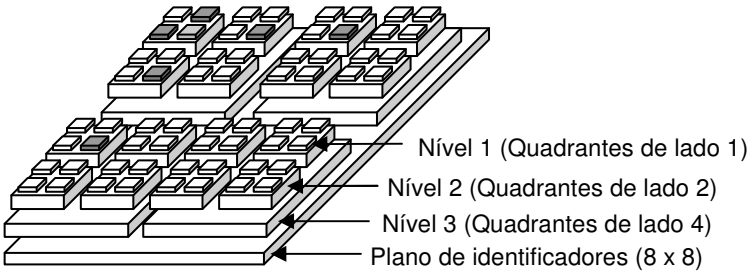


Figura 4. Visão hierárquica da grade com três níveis mostrada na Figura 3

2.4 Algoritmo de pesquisa (roteamento)

Quando um nó recebe uma pesquisa por uma chave, ele calcula o quadrado de menor nível que contém o nó e a chave. Depois disso, repassa a pesquisa para o nó no quadrante vizinho desse nível que é mais próximo da chave. Esse procedimento é repetido até que o nó responsável pela chave seja alcançado.

Existem quatro quadrantes em cada nível, e cada nó possui um ponteiro para o seu vizinho vertical e outro para o vizinho horizontal. Portanto, para que um nó e uma chave estejam localizados no mesmo quadrante em um dado nível, são necessários, no máximo, dois saltos (contatar dois nós apenas). Depois de alcançar um quadrante que contenha o nó e a chave, o processo de pesquisa continua em um nível inferior. Nesta situação existem (Log N)/2 níveis e o número máximo de nós consultados em um processo de pesquisa é ((Log N)/2) * 2, ou seja, Log N.

A Figura 5 ilustra uma pesquisa pela chave (62,1), ponto marcado com um círculo, em uma rede com 512 nós em uma grade com tamanho de lado 64. Os pontos sólidos representam os nós acessados durante a consulta (incluindo o nó onde a pesquisa iniciou). Os pontos marcados com um X no quadrado indicam os outros nós. As linhas delimitam a região de identificadores pela qual cada nó é responsável.

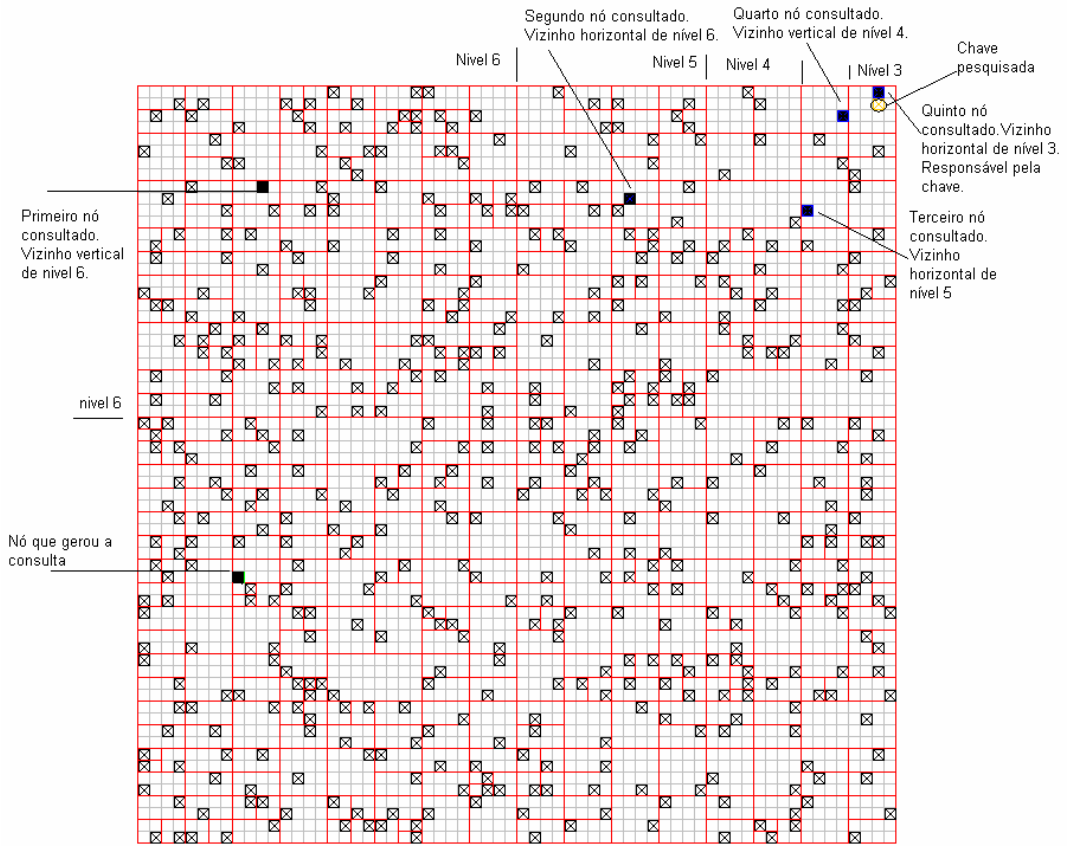


Figura 5: Processo de pesquisa em uma grade com 512 nós e tamanho de lado 64

Tabela 1 - Seqüência de uma consulta

Nó	Tamanho do lado do quadrado	Nível	Quadrante do nó	Quadrante da chave	Vizinho [nível]
Nó que iniciou a consulta	32	6	2	1	Vertical [6]
Primeiro Nó	32	6	0	1	Horizontal [6]
Segundo Nó	16	5	0	1	Horizontal[5]
Terceiro Nó	8	4	3	1	Vertical [4]
Quarto Nó	4	3	0	1	Horizontal [3]
Quinto Nó	-	-	-	-	-

A Tabela 1 resume o papel de cada nó no processo de pesquisa ilustrado na Figura 5. A coluna “Nó” identifica o nó. A coluna “tamanho do lado do quadrado” mostra o quadrado que contém o nó e a chave. A coluna “nível” identifica a que nível esse tamanho de quadrado se refere. As colunas “quadrante do nó” e “quadrante da chave” identificam o quadrante do quadrado que contém esses elementos. Os valores determinam se a pesquisa deve ser repassada para o vizinho horizontal ou vertical, de acordo com o valor da coluna “Vizinho [nível]”. Essa coluna representa também o nível para o qual o vizinho horizontal ou vertical se refere. A última linha não contém valores porque o quinto nó é responsável pela chave pesquisada.

2.5 Entrada de nós na rede

Quando um nó *n* entra na rede, os seguintes procedimentos são aplicados:

- Contatar o nó que contém as chaves a serem divididas e perguntar ao mesmo quais são as chaves que serão transferidas para *n*. Para que isso seja possível é necessário algum mecanismo de *bootstrapping* da rede para permitir que máquinas fora da rede encontrem algum nó da rede. Isso pode ser feito publicando o endereço de alguns nós no DNS (*Domain Name Service)*. Vale ressaltar que quaisquer nós podem ser escolhidos para serem publicados no DNS, uma vez que a função deles será iniciar as buscas pelos nós a terem as chaves divididas. Evidentemente, devem estar conectados na maior parte do tempo e possuírem boa velocidade de conexão.
- Inicializar a tabela de apontadores para os nós dos quadrantes vizinhos em cada nível. Isso é feito perguntando a algum nó na rede pelos nós responsáveis pelas chaves relacionadas com as coordenadas nos quadrantes vizinhos para as quais o nó deve apontar (ver seção 2.3). Contudo, algumas técnicas de otimização são utilizadas para reduzir a complexidade dessa operação para O(Log N). Por exemplo, obter uma cópia da tabela de apontadores do nó com as chaves a serem divididas e realizar as pesquisas baseadas nessa tabela.
- Atualizar os apontadores dos nós que apontam para as chaves transferidas. De acordo com os resultados das simulações, o número de nós que precisam ter seus apontadores reconfigurados quando um novo nó entra na rede é inferior a $O(\log^2 N)$.
- Notificar o software na camada superior (usuário da rede P2P) que utilize o nó com as chaves a serem divididas. Este software deve transferir os dados associados com as chaves.

O procedimento para identificar os nós que precisam ter seus apontadores reconfigurados depois que um novo nó entra na rede é baseado na identificação do menor quadrado que contém as chaves que foram divididas. Desse modo, as chaves que pertencem ao novo nó estão em um ou dois quadrantes deste quadrado. As coordenadas do canto superior esquerdo deste quadrado são utilizadas para calcular os vizinhos em cada nível deste quadrado. Após isso, para cada nova posição do quadrado, é necessário atualizar apenas os nós dos quadrantes que se referem às chaves do novo nó.

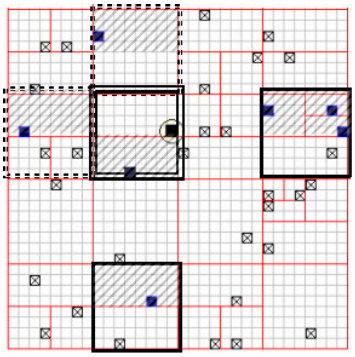


Figura 6: Nós a serem atualizados quando um novo nó entra na rede

A Figura 6 mostra como a estratégia do quadrado, e seus quadrantes internos, pode ser utilizada para identificar os nós que precisam ser reconfigurados quando um novo nó entra na rede. Nessa figura, o novo nó *n* é identificado com um círculo. Esse nó obteve suas chaves através de uma divisão horizontal do espaço de chaves do nó *n'*, que está no mesmo quadrado de *n*. Os nós que tiveram seus apontadores atualizados são identificados por um quadrado sólido. Os diferentes níveis envolvidos na identificação dos vizinhos são representados na figura por diferentes tipos de borda ao redor dos quadrados. Os quadrantes a serem atualizados dentro de cada quadrado são representados por linhas diagonais.

2.6 Saída de nós da rede

Quando um nó *n* deixa a rede os procedimentos são similares àqueles realizados quando um nó entra na rede:

- Identificar o nó vizinho *n'* para o qual *n* deve transferir suas chaves.
- Atualizar os apontadores dos outros nós que apontam para as chaves que *n* transferiu para *n'*.
- Notificar o software que utiliza a rede P2P para transferir os dados associados às chaves para *n'*.

A identificação do nó vizinho para receber as chaves durante o processo de remoção de um nó da rede segue a condição que o formato da região do espaço de chaves deve ser retangular ou quadrada. Portanto, quando um nó está deixando a rede, suas chaves são transferidas para o nó vizinho que mantém essa condição. Figura 7b e 7c ilustram o novo cenário do espaço de chaves depois de remover alguns nós ilustrados na Figura 7a.

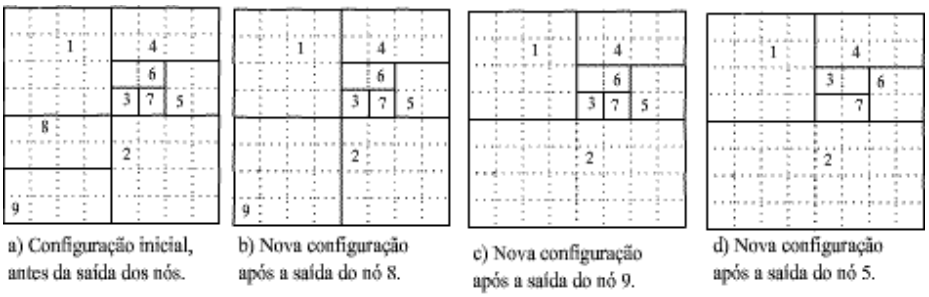


Figura 7: Reconfiguração da rede após a saída de nós

Quando a condição do formato das regiões não pode ser atendida reconfigurando apenas o nó vizinho, a reestruturação deve envolver também outros nós próximos do nó deixando a rede. O processo de reestruturação consiste em mover nós para outras coordenadas no espaço da grade. Quando um nó *n* está deixando a rede, ele escolhe um vizinho *n'* para ser movido para uma nova coordenada dentro da região de chaves do nó *n* que é mais próxima da região de chaves do nó *n'*. A Figura 7d ilustra essa situação – quando o nó 5 deixa a rede, os nós 6 e 3 são movidos.

3. Implementação do SGrid

A Figura 8 ilustra a interface gráfica do SGrid, implementado em Java. A interface oferece funções para criar uma rede e executar as operações descritas na seção anterior. Além das operações de adicionar e remover nós, que podem ser realizadas

individualmente ou em grupo, e a operação para pesquisar chaves na rede, o sistema também permite a geração de gráficos estatísticos para analisar o comportamento de algumas variáveis importantes para o desempenho e escalabilidade da rede.

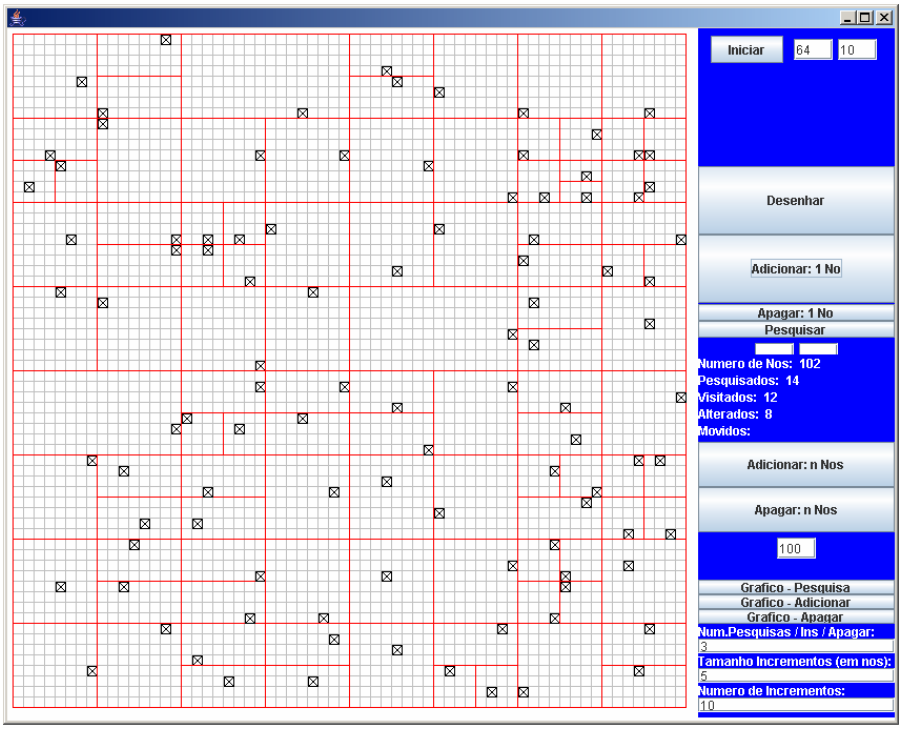


Figura 8: Interface Gráfica do SGrid

A ferramenta gera dois gráficos, ilustrados na Figura 9: (1) o primeiro gráfico mostra o número de nós visitados (média utilizando linha pontilhada e máximo utilizando linha cheia) em uma operação de pesquisa variando o número de nós na rede; (2) o segundo gráfico ilustra os nós reconfigurados (média utilizando linha pontilhada e máximo utilizando linha cheia) quando um novo nó entra na rede. Nesse segundo gráfico, também houve variação do número de nós na rede.

Para o experimento cujo gráfico de pesquisa está ilustrado na Figura 9, as consultas foram realizadas a cada intervalo de 50 novos nós inseridos na rede. O número de consultas foi 300 (trezentos). Para o segundo gráfico, a avaliação de desempenho relacionada a inserção de novos nós considerou redes com variações de 100 (cem) nós. O número de nós inseridos para avaliar a operação de reconfiguração para cada rede foi 50 (cinqüenta).

Os resultados mostraram que o número máximo de nós visitados em um processo de pesquisa é O(Log N). Contudo, na média, este valor é reduzido pela metade. Em uma rede com 1000 (mil) nós o número máximo nós visitados em uma pesquisa foi 10 (dez), que corresponde a O(Log N). Na média o número de nós visitados foi 4 (quatro), que é menor que O(Log N)/2. O número de nós reconfigurados depois de inserir um novo nó é inferior a $O(Log^2 N)$.

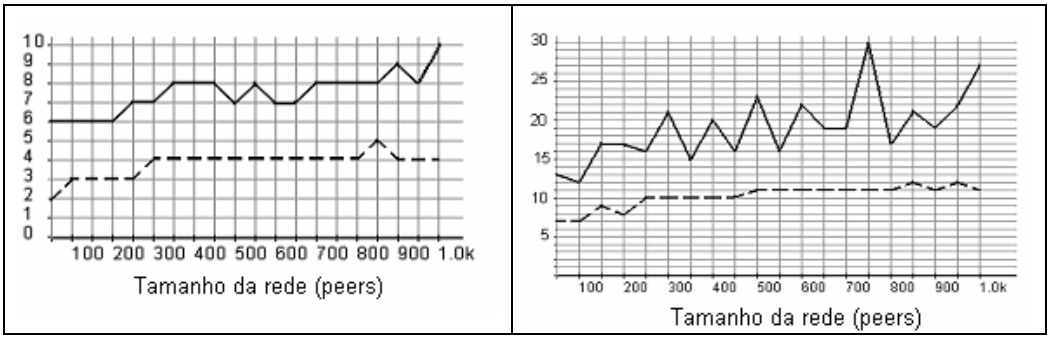


Figura 9: Gráficos (pesquisa à esquerda e inserção a direita) gerados pela ferramenta de simulação do SGrid

4. Trabalhos Relacionados

Com o objetivo de obter escalabilidade, garantir o sucesso das pesquisas por dados que existem na rede e fornecer operações de consulta com desempenhos determinísticos, muitas pesquisas recentes em redes P2P estão evitando estratégias centralizadas e mecanismos de consulta baseados em inundação. A principal alternativa para esses tipos de estratégias é baseada em redes P2P estruturadas.

Existem muitas propostas de redes P2P estruturadas que basicamente diferem em como os pares chave-valor são distribuídos entre os nós e no método que determina quais são os nós com os quais cada nó se comunica diretamente. No Chord [Stoica 2001] o espaço de chaves é organizado em um círculo e no CAN o espaço é organizado em múltiplas dimensões. P-Grid [Aberer 2001] utiliza uma árvore virtual de pesquisa distribuída. O Chord utiliza uma função *hash* para mapear nós e dados para o espaço de identificadores – o anel. Os nós são mapeados aplicando-se uma função *hash* nos seus endereços IP. Cada chave *k* é armazenada no nó que possui como identificador o menor número maior ou igual a *k*. A estrutura circular do Chord permite que cada nó mantenha apenas apontadores para seu sucessor e seu antecessor no anel. Contudo, para melhorar o desempenho das pesquisas e reduzir a complexidade dessa operação para O(log N), cada nó mantém apontadores para O(log N) nós. Embora o desempenho do Chord e do SGrid sejam similares, um modelo em grade fornece um suporte melhor para o desenvolvimento de protocolos de pesquisa que tiram proveito das propriedades geométricas e da visão hierárquica proporcionadas por uma arquitetura baseada em grade.

O CAN utiliza um espaço *d*-dimensional e implementa uma DHT onde cada nó mantém O(*d*) informações de estado. Cada nó é responsável por uma parte do espaço de chaves (zona), e os nós se comunicam com pares responsáveis por zonas adjacentes. Conforme exposto em [Stoica 2001] o custo das operações de pesquisa em CAN incrementa mais rápido que log N. A forma como mantém apontadores para outros nós acarreta uma desempenho inferior ao SGrid. O conceito de *`multiplas realidades´* de CAN pode levar ao mesmo desempenho do SGrid, mas para isso é necessário armazenar em cada nó uma quantidade maior de informações de estado.

Existem dois trabalhos que, semelhantemente ao SGrid, utilizam uma abordagem hierárquica em uma grade: Leopard [Yu 2004] e GLS (*Scalable Location Service*) [Li 2000]. Leopard tem como objetivo principal a replicação de dados e para isso insere ponteiros para o mesmo dado em vários níveis da hierarquia. Este mecanismo resolve as pesquisas contatando nós mais próximos do nó que iniciou a

pesquisa. Contudo, o foco principal do Leopard é prover uma arquitetura para criar índices para acessar dados. O Leopard não especifica uma topologia de rede nem como encontrar o nó responsável por uma dada chave. Além disso, não determina o procedimento para o roteamento de mensagens na rede P2P.

O GLS utiliza uma grade hierárquica que mantém apontadores para os quadrantes vizinhos em cada nível. O método utilizado para escolher os nós é diferente do proposto no SGrid. Enquanto no SGrid os nós são escolhidos de acordo com uma coordenada fixa no quadrante, o GLS escolhe os nós de acordo com seus identificadores. Como conseqüência, no GLS cada nó aponta para o nó no outro quadrante que possui o menor identificador que seja maior que seu próprio identificador (similar ao Chord). Além disso, como o GLS tem como enfoque principal a localização de dispositivos móveis, ele utiliza um esquema de atualização periódica de alguns nós que possuem funções especiais e são chamados de *servidores de localização*. Esse mecanismo limita a escalabilidade do GLS.

5. Considerações Finais

Sistemas P2P fornecem muitas vantagens para aplicações como sistemas de compartilhamento de arquivos sem haver necessidade de investimentos em hardware ou capacidade de transmissão da rede. Redes P2P não-estruturadas são flexíveis na seleção dos vizinhos e nos mecanismos de roteamento, porém, não possuem boa escalabilidade. Neste artigo, propomos uma rede P2P estruturada e cuja organização hierárquica é baseada no esquema de endereçamento IP - SGrid. Apresentamos a arquitetura do SGrid, seu esquema de índices e suas operações. Apresentamos também uma implementação do SGrid e mostramos sua interface de usuário e os gráficos que são gerados pela ferramenta. A ferramenta de simulação do SGrid validou a escalabilidade dessa arquitetura para uma rede com 20.000 (vinte mil) nós. Além disso, validou os procedimentos de inserção e remoção de nós mostrando que nessas operações são afetados um número logarítmico de nós.

Embora algumas outras redes P2P apresentem comportamento similar, uma diferença chave entre SGrid e esses outros trabalhos é que o SGrid utiliza uma estratégia com ciência da localização física dos nós baseada no endereço IP de cada nó. Essa estratégia reduz o tráfego de rede e a latência, uma vez que diminui significativamente as situações, comuns em muitas arquiteturas, onde dois nós que se comunicam diretamente na rede P2P estão muito distantes na rede IP. O SGrid é diferente do CAN na forma como mantém os identificadores dos nós vizinhos. O SGrid segue o principio conhecido como *Small World* [Kleinberg 2000] que procura manter uma pequena cadeia de relacionamento entre quaisquer dois elementos de um conjunto. Esta estratégia reduz os custos para localizar dados distantes. O modelo hierárquico do SGrid fornece uma estrutura adequada para a construção de aplicações baseadas em árvore como, por exemplo, a criação de índices hierárquicos (como índices para documentos XML) ou índices que utilizam *bloom filters* [Mitzenmacher 2001].

Trabalhos em andamento relacionados com o SGrid incluem a proposta de um protocolo de pesquisa complexo sobre a infra-estrutura do SGrid, um protocolo *multicast* e a definição de um serviço de segurança para evitar o ingresso de nós mal-intencionados no SGrid. Além disso, estamos desenvolvendo um modelo para a utilização de super-nós (*super-peers*) no SGrid. Esse modelo beneficia-se do esquema de mapeamento dos endereços IP do SGrid e propõe uma forma de utilização do NAT [Srisuresh 1999] para criar *super-peers leves*, onde procura-se evitar o *overhead*

imposto pelos *super-peers* atuais [Mizrak 2003] que trabalham na camada de aplicação. Outro objetivo desse modelo de *super-peers* é agrupar nós em *clusters* a fim de atingir redes com topologia relativamente estável.

Nesse artigo não foi abordado como lidar com o NAT, pois esse tema é tratado com especial atenção no modelo do SGrid com *super-peers*, onde é definido um protocolo, chamado NATal (*Routing and **NAT** **a**pplication **l**ayer)*, que realiza roteamento e NAT baseados nos dados da camada de aplicação. Nesse modelo o mapeamento dos nós para a grade é feito através de um método que combina o esquema hierárquico, definido nesse artigo, com uma função *hash*, considerando o endereço do roteador (IP público) e o endereço da máquina (possivelmente privado). Esse mesmo método pode ser aplicado a arquitetura do SGrid apresentada nesse artigo, bastando para isso, informar aos nós o IP público utilizado para as operações de NAT. Isto pode ser feito através de protocolos como SLP, DHCP, entre outros.

Referências

Aberer, K. (2001) P-Grid: A Self-Organizing Access Structure for P2P Information Systems. Sixth International Conference on Cooperative Information Systems (CoopIS 2001), Trento, Italy, Lecture Notes in Computer Science 2172, Springer Verlag, Heidelberg, 2001.

Aberer, K., Alima, L. O., Ghodsi, A., Girdzijauskas, S., Hauswirth, M. e Haridi, S. (2005) "The essence of P2P: A reference architecture for overlay networks", The Fifth IEEE International Conference on Peer-to-Peer Computing, Konstanz, September 2005.

Byer, B., Martin, E., Edwards, C. e Heijden, T. (2000) “Especificação do protocolo Napster”, Abril.

Doval, D., e O'Mahony, D. (2002) "Nom: Resource location and discovery for ad hoc mobile networks," in Med-hoc-Net 2002, (Sardegna, Italy), September 2002.

Hightower, J. Borriella, G. Location systems for ubiquitous computing. IEEE Computer, 34(8):57-66. 2001.

Kleinberg, J. (2000) The Small-World Phenomenon: An Algorithmic Perspective. In Proceedings of the 32nd ACM Symposium on Theory of Computing, pages 163-170, 2000.

Li, J., Jannotti, J., De Couto, D. S. J., Karger, D. R. e Morris, R. (2000) A Scalable Location Service for Geographic Ad Hoc Routing, ACM Mobicom 2000.

Mitzenmacher, M. (2001) Compressed Bloom Filters. Annual ACM Symposium on Principles of Distributed Computing. Proceedings of the twentieth annual ACM symposium on Principles of distributed computing. Newport, Rhode Island, United States, pp. 144 - 150

Mizrak, A., Cheng, Y., Kumar, V. e Savage, S. (2003) Structured Superpeers: Leveraging Heterogeneity to Provide Constant-Time Lookup. Proceedings of the 4th IEEE Workshop on Internet Applications, San Jose, CA, June 2003

Ng, T. E. e Zhang, H. (2002) Predicting Internet network distance with coordinates-based approaches. In Proc. of IEEE INFOCOM, June 2002.

Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Shenker, S. (2001) "A Scalable Content-Addressable Network", ACM SIGCOMM, August 2001.

Ratnasamy, S., Shenker, S. e Stoica, I. (2002) "Routing algorithms for DHTs: Some open questions," IPTPS02, Cambridge, USA, March 2002.

Rowstron, A. e Druschel, P. (2001) "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," International Conference on Distributed Systems Platforms (Middleware), November 2001.

Srisuresh, P. e Holdrege, M. (1999) IP Network Address Translator(NAT) Terminology and Considerations RFC 2663, IETF Network Working Group, August 1999.

Stoica, I., Morris, R., Karger, D., Kaashoek, M. F. e Balakrishnan, H. (2001) "Chord: A scalable peer-to-peer lookup service for internet applications," Proceedings of the 2001 ACM SIGCOMM Conference, pp. 149-160.

The Gnutella Protocol Specification v0.4 (Document Revision 1.2), June 2001. http://www9.limewire.com/developer/gnutella protocol 0.4.pdf.

Yu, Y., Lee, S. e Zhang, Z.-L. (2004) Leopard: A locality-aware peer-to-peer system with no hot spot, Tech. Report CSE Dept., U of Minnesota, 2004.

Zhao, B. Y., Kubiatowicz, J. D. e Joseph, A. D. (2001) "Tapestry: An infrastructure for fault-resilient wide-area location and routing," U.C. Berkeley, Berkeley Technical Report UCBCSD-01-1141, April 2001.