

# Proposta de Rede P2P Organizada por Índices

Gabriel Epsztejn e Otto Carlos M. B. Duarte

Grupo de Teleinformática e Automação  
Universidade Federal do Rio de Janeiro – COPPE-Poli/UFRJ  
Rio de Janeiro – RJ – Brasil

ge@gta.ufrj.br, otto@gta.ufrj.br

**Abstract.** *The progress made in P2P networks and the fast adoption by the users has led to the challenge of implementing solutions that support the sharing of different types of resources and that are capable to offer more complex services. In this paper, we propose a P2P architecture with an organization of the nodes by indexes, aiming at offering a flexible and scalable service for complex queries.*

**Resumo.** *Os avanços em redes P2P e a rápida adoção pelos usuários conduziram ao desafio de se implementar soluções que suportem o compartilhamento de diferentes tipos de recursos e que sejam capazes de oferecer serviços mais complexos. Neste trabalho propomos uma arquitetura de redes P2P com organização dos nós por índices, que objetiva oferecer um serviço flexível e escalável de busca com regras complexas.*

## 1. Introdução

O uso de redes P2P como suporte a serviços mais complexos tem sido tema de diversas pesquisas. As arquiteturas propostas por Ratnasamy [Ratnasamy *et al.*, 2001] e Stoica [Stoica *et al.*, 2001] abriram o caminho para o projeto de arquiteturas mais estruturadas e escaláveis, baseadas em tabelas *hash* distribuídas (DHTs). Ao associar a localização dos nós na rede à responsabilidade por um conjunto de recursos, a abordagem por DHT representou uma importante evolução em comparação às arquiteturas não-estruturadas adotadas pelos usuários de Internet, como o Gnutella e o KaZaA.

No entanto, o emprego das tabelas *hash* distribuídas (DHTs) permite apenas buscas baseadas em identificadores únicos, o que dificulta a oferta de novos serviços que requeiram buscas mais complexas, como descoberta de recursos por intervalos de valor dos atributos ou valores máximos e mínimos. Assim, algumas técnicas foram propostas para viabilizar a execução de buscas semânticas em redes P2P utilizando as arquiteturas disponíveis.

Neste sentido, Crespo [Crespo e Garcia-Molina, 2002] propõe *Semantic Overlay Networks* onde nós são agrupados em redes de sobreposição por assuntos. Handurukande [Handurukande *et al.*, 2004] avalia o efeito da criação de *clusters* semânticos para a rede eDonkey, pela criação de ligações adicionais entre nós, baseadas em heurísticas para identificação dinâmica da correlação semântica entre nós. Entretanto, estas propostas mantêm as limitações inerentes às arquiteturas não-

estruturadas, utilizando buscas baseadas em inundação, sem permitir buscas completas na rede.

Assim, a utilização de uma única infra-estrutura P2P para compartilhamento de diferentes tipos de recursos, com diferentes parâmetros de busca e requisitos de qualidade de serviço (QoS) de forma eficiente ainda está em aberto [Talia e Trunfio, 2003].

O desenvolvimento de uma solução para suportar tais requisitos desponta como o ponto de interseção entre redes P2P e serviços de computação em GRID [Foster e Iamnitchi, 2003], a partir de onde será possível utilizar a infra-estrutura completamente descentralizada da primeira para oferecer os serviços profissionais usualmente associados à segunda. Isso permitirá a utilização de um grande número de computadores menores, geralmente utilizados em serviços P2P, para implementar serviços típicos de computação em GRID, que tem tido maior foco na utilização de recursos mais robustos.

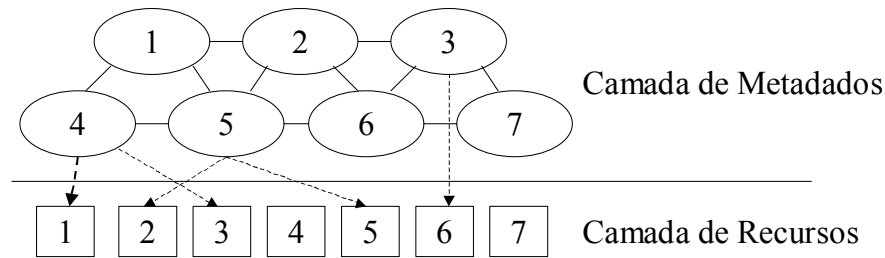
A arquitetura proposta deve permitir a execução de buscas complexas e a implementação de algoritmos de qualidade de serviço diferenciados. Por outro lado, deve manter as características fundamentais de escalabilidade e tolerância a falhas. Trabalhos anteriores sobre soluções P2P orientadas a serviços, como [Broekstra *et al.*, 2003] e [Rusitschka e Southall, 2003], já exploraram a maior parte das questões referentes à utilização de metadados para descrições dos recursos e sua aplicação em redes P2P orientadas a serviços. Entretanto, não abordaram a necessidade de organização diferenciada dos nós para implementação das propostas.

Nas próximas seções, apresentamos e analisamos a viabilidade de implementar tal solução tendo como fundamento a organização dos nós por índices relevantes para buscas. Uma vez que diferentes tipos de recursos requerem índices diferentes, torna-se um importante desafio permitir que uma única rede seja capaz de suportar o compartilhamento de recursos de naturezas distintas.

O artigo está organizado da seguinte forma. Na seção 2, é apresentada a arquitetura proposta, após uma breve revisão de propostas anteriores. Na seção 3, são demonstradas análises teóricas e resultados de desempenho obtidos em simulação. Por fim, são relatadas na seção 4 as conclusões obtidas desse trabalho.

## **2. Arquitetura Proposta**

A arquitetura proposta é uma rede de sobreposição com duas camadas: a Camada de Metadados e a Camada de Recursos. A Camada de Metadados é responsável por prover a descrição dos recursos disponíveis, incluindo sua localização e atributos, e pela execução das buscas. A Figura 1 é uma representação gráfica dos nós e da rede construída.



**Figura 1. Representação do posicionamento dos nós físicos na rede de sobreposição.**

Na Figura 1, os números representam os nós físicos. Cada nó faz parte da camada de recursos, como provedor e consumidor, e também da camada de metadados, armazenando a descrição de uma parcela dos recursos disponíveis na rede. O conceito da camada de metadados já foi implicitamente sugerido em [Ratnasamy *et al.*, 2001] e mais profundamente explorado na arquitetura P2P orientada a serviços sugerida em [Pitoura *et al.*, 2003]. Entretanto, como mencionado anteriormente, ela deve possuir uma organização dos nós que suporte buscas complexas de forma eficiente, baseadas em diferentes parâmetros.

Em nossa proposta, a camada de metadados além de servir como estrutura de busca, passa a ser responsável por assegurar o nível de serviço adequado a cada um dos recursos disponíveis. Isso quer dizer que cada nó dessa rede também é responsável pela execução de algoritmos que garantam, quando aplicável, disponibilidade e balanceamento de carga para o grupo de recursos sob sua responsabilidade. Sempre que um novo recurso é disponibilizado e registrado na camada de metadados, sua descrição também pode incluir requisitos de qualidade de serviço.

É interessante atribuir essa responsabilidade desta forma uma vez que os nós que contêm os metadados podem manter o histórico dos acessos feitos ao recurso, uma vez que o consumo de um recurso é precedido por uma busca. Os consumidores também podem relatar, aos nós responsáveis, parâmetros de qualidade de serviço verificados ao utilizar os recursos.

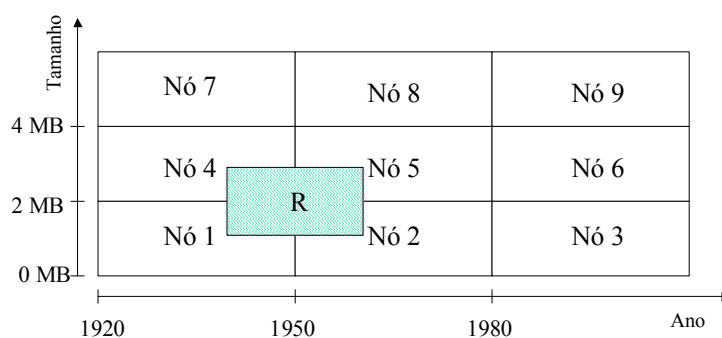
## 2.1. Organização dos Nós

A questão fundamental no desenvolvimento da arquitetura é a escolha adequada para a organização dos nós. Essa escolha determinará o nível de complexidade e escalabilidade para implementação dos requisitos evolutivos dos serviços. Considerando isso, a arquitetura escolhida deve suportar as seguintes características fundamentais: escalabilidade, balanceamento de carga, alta disponibilidade e suporte para buscas completas e por intervalos.

As arquiteturas não-estruturadas não se adequam bem a nossos objetivos, uma vez que utilizam algoritmos de inundação para execução das buscas e não prevêm nenhuma coordenação para a distribuição de responsabilidades entre nós. Isso também dificulta a implementação de técnicas de balanceamento de carga e de alta disponibilidade. O uso de soluções previamente sugeridas para agrupamento semântico de nós [Crespo e Garcia-Molina, 2002] proveria um certo nível de busca semântica, mais ainda assim seria inadequado para atender aos outros requisitos.

Uma solução baseada em DHT, como a proposta em [Stoica *et al.*, 2001] proveria um bom nível de escalabilidade e a possibilidade de implementar balanceamento de carga e disponibilidade. Entretanto, a execução de buscas por intervalos em tais redes requer adaptações para implementação e não provê a flexibilidade necessária ao suporte de vários tipos de recurso. Uma proposta mais aderente foi feita por Ratnasamy [Ratnasamy *et al.*, 2003], apresentando a utilização de uma estrutura em árvore sobre DHT, denominada PHT. Entretanto, importantes questões permaneceram em aberto, como buscas por múltiplos atributos e a necessidade de execução de múltiplas buscas para resolver uma única solicitação.

A arquitetura proposta e avaliada é derivada de *Content Addressable Network* (CAN) [Ratnasamy *et al.*, 2001], onde um espaço multidimensional é criado e cada nó é responsável por um hipercubo em tal espaço. Entretanto, em nossa proposta, as dimensões representam alguns dos atributos dos recursos com relevância para buscas, aos quais denominamos índices. Esses atributos são utilizados como parâmetros para execução de buscas, permitindo que qualquer busca por intervalos de valor seja mapeada em um hipercubo no espaço d-dimensional.



**Figura 2. Organização dos nós e regiões de responsabilidade.**

A Figura 2 apresenta um espaço bi-dimensional, onde a dimensão horizontal representa o ano de publicação e a vertical, o tamanho do arquivo. A região R representa o retângulo associado a busca por “todos os arquivos publicados entre 1940 e 1960, com tamanho maior que 1 MB e menor que 3 MB”.

## 2.2. Execução de Buscas

O algoritmo para localização de um item único é similar ao utilizado na arquitetura CAN, baseada em encaminhamento por menor distância geométrica. Já a execução de buscas por intervalos é composta por duas etapas. Inicialmente, é executada uma busca para alcance de um nó da rede que faça parte da região R desejada. Este nó inicia, então, uma busca por inundação, limitada à região buscada.

A primeira etapa utiliza um algoritmo semelhante ao definido para a localização de um único item da rede. Sua única particularidade é que o ponto geométrico buscado não é resultado da aplicação de diversas funções *hash* a um identificador único, mas sim pelo cálculo, feito pelo nó solicitante, do ponto da região desejada mais próximo às fronteiras de sua própria região de responsabilidade.

Dado que, durante a fase de resposta, cada nó pode informar ao solicitante sua própria região de responsabilidade, é possível saber quando a busca foi completamente executada. Também é possível realizar buscas por máximos e mínimos, solicitando a execução de uma busca em uma direção e sentido pré-definidos e suspendendo-a quando um item for encontrado.

Entretanto, é importante observar que, uma vez que não há aplicação de funções *hash*, deve-se esperar a ocorrência de regiões no espaço  $d$ -dimensional com diferentes densidades de itens. Isso quer dizer que, para implementação de balanceamento de carga, deverá haver nós com regiões de responsabilidade maiores que outros. Isso considerado, não se pode afirmar que a distância geométrica mínima entre dois pontos no espaço se traduza num menor número de saltos, o que faz com que o comprimento médio para as buscas de um único item salte de  $(d/4)(n^{1/d})$  [Ratnasamy *et al.*, 2001] para  $(d/2)(n^{1/d})$ , onde  $d$  é o número de dimensões e  $n$  é o número de nós na rede.

### 2.3. Balanceamento de Carga

A implementação de balanceamento de carga em soluções baseadas em DHT foca na distribuição uniforme da responsabilidade pelos ponteiros para os recursos entre os nós. Isso pode ser alcançado de forma relativamente simples, pela divisão igualitária de regiões de responsabilidade entre os nós, dado o caráter aleatório da localização associada a cada recurso na rede. Muitas vezes, é necessária a criação de nós virtuais, o que impõe sérias desvantagens à rede [Karger e Ruhl, 2004]. Outra questão importante é que, na maior parte das vezes, os nós possuem capacidades diferentes de processamento e memória, assim como diferentes níveis de robustez, o que significa que distribuir igualmente a responsabilidade não é necessariamente a melhor opção.

No nosso caso, um particionamento igualitário do espaço  $d$ -dimensional entre os nós não asseguraria nem mesmo uma distribuição igualitária da responsabilidade por itens. Ainda que o fizesse, não há razão para crer que tal equilíbrio proveria um balanceamento de carga adequado, uma vez os níveis de requisição por recursos são díspares e os nós não possuem capacidade idêntica para responder a pedidos. Sugerimos o uso de um algoritmo de balanceamento de carga dinâmico, baseado em uma auto-análise de carga recorrente em cada nó e comparações com seus vizinhos, como proposto em [Karger e M.Ruhl, 2004]. Sempre que um nó identifica que sua carga é muito maior (valor de disparo) que a de um de seus vizinhos, executa o re-particionamento da sua região, passando o controle de uma parte da mesma ao vizinho.

### 2.4. Disponibilidade

Apesar de diferentes tipos recursos demandarem diferentes níveis de disponibilidade, a camada de metadados deve garantir, com um alto grau de certeza que, em qualquer dado momento, haverá ao menos um nó acessível responsável pelos metadados de cada recurso da rede.

Existem algumas opções para implementar tolerância a falhas inesperadas dos nós. Todas elas incluem, de alguma forma, a replicação dos metadados. A proposta feita em [Ratnasamy *et al.*, 2001] é a criação de mais “realidades”, o que também ajudaria na aceleração das buscas. Entretanto, quando consideramos a execução de buscas por intervalos, observamos que outra estratégia pode ser mais interessante. No exemplo da

Figura 2, apesar de a região buscada ocupar apenas 1/12 do espaço completo, participariam da busca 4/9 dos nós. Isso ocorre devido a uma defasagem entre os limites da região desejada em relação às regiões de responsabilidade dos nós. Assim, sugerimos que cada nó contenha a replicação dos metadados de seu vizinho em uma determinada direção definida. Dessa forma, podemos garantir a disponibilidade e otimizar a busca por intervalos (no exemplo acima, apenas 2 nós participariam da busca).

## 2.5. Flexibilidade

Para permitir a busca por vários tipos de recursos, com atributos distintos, as dimensões não são de tipos estritamente definidos, como ano, tamanho de arquivo, nome ou data, mas de um tipo único: inteiro. Antes da busca, o nó solicitante executa um algoritmo para mapeamento de cada parâmetro de busca do tipo específico em um número inteiro, e inclui na descrição da busca o tipo de recurso sendo solicitado. Esse algoritmo mantém o ordenamento durante o mapeamento. No caso da Figura 2, a busca seria traduzida de “todos os itens publicados de 1940 a 1960 com tamanho entre 1MB e 3MB” para “todos os ARQUIVOS com  $60 > D1 > 80$  e  $30 > D2 > 40$ ”. Fica claro que, para cada tipo de recurso, cada dimensão assume um significado individual. Entretanto, a informação sobre o tipo de recurso sendo requisitado não precisa ser avaliada em todos os saltos durante a busca, apenas por aqueles nós na região buscada.

O algoritmo de mapeamento a ser executado é específico para cada tipo de recurso mas pode ser derivado de uma descrição do tipo de recurso sendo buscado. Essa descrição inclui os parâmetros de busca suportados e suas associações às dimensões da rede. Caso o nó não possua a descrição referente ao tipo de recurso desejado, pode buscá-la na própria rede, uma vez que é ela própria, um recurso disponível na rede. Dessa forma, a única informação que um nó necessita ter antes de ingressar na rede é o algoritmo de mapeamento para o tipo de recurso “arquivo de descrição de tipo de recurso”.

O número máximo de índices de busca suportado na rede é a quantidade de dimensões da mesma. Já para os tipos de recursos que não utilizem todas as dimensões como índices, implementa-se uma simples distribuição aleatória dos itens pelas dimensões restantes, utilizando-se a aplicação de funções *hash* a atributos ou identificadores únicos dos recursos.

## 3. Avaliação de Desempenho

A avaliação de desempenho e escalabilidade da arquitetura proposta é feita pela análise do número de nós participantes nas buscas. Nesta seção apresentamos o desenvolvimento do cálculo teórico para esse valor. Também simulamos e exploramos os resultados da utilização da técnica de disponibilidade proposta e os efeitos da existência, na mesma rede, de tipos de recursos suportando quantidades diferentes de parâmetros de busca.

### 3.1 Previsão Teórica

Conforme mencionado na seção 2, o número médio de nós envolvidos em uma busca por um único recurso na rede é de  $(d/2)(n^{1/d})$ .

Podemos definir o número de nós envolvidos em uma busca baseada em intervalos de valor por:

$$n_t = n_a + n_v$$

Onde:  $n_t$  é o número total de nós participantes.

$n_a$  é o número de nós participantes na fase de alcance da região, quando a busca é encaminhada até o ponto mais próximo da região desejada.

$n_v$  é o número de nós participantes na segunda fase da busca, quando o encaminhamento é feito dentro da região desejada.

$$E(n_t) = E(n_a + n_v) = E(n_a) + E(n_v)$$

$$E(n_t) = (d/2)(n^{1/d}) + E(n_v)$$

Onde:  $E(n)$  é o valor médio de  $n$ .

$d$  é o número de dimensões utilizadas na organização da rede.

$n$  é o número total de nós na rede.

Na expressão acima,  $E(n_a)$  foi determinado tendo como base o número médio de nós necessários para a execução de uma busca no espaço  $d$ -dimensional, demonstrado em [Ratnasamy *et al.*, 2001]. Essa aproximação é válida quando consideramos que a região buscada é muito pequena quando comparada ao espaço completo.

Para determinar o número médio de nós envolvidos na segunda etapa, consideramos que todas as dimensões estão divididas em números iguais de segmentos. Também consideramos que a busca é igualmente restritiva em todas as direções. Isso quer dizer por exemplo que, se a região buscada é de 1% do espaço total, o filtro em cada direção representa  $(0,01)^{1/k}$  de todo seu comprimento, sendo  $k$  número de dimensões restringidas pela busca. Desejamos escrever a equação em função da relação  $R/U$  entre o tamanho da região buscada e o tamanho total do universo. Percebemos que, em cada dimensão restringida, pode haver uma defasagem de 1 ou 2 nós entre a região buscada e as regiões de responsabilidade. Dessa forma, chegamos à seguinte expressão:

$$E(n_v) = [(\text{Inteiro}(W \cdot n^{1/d}) + 2) \cdot P + (\text{Inteiro}(W \cdot n^{1/d}) + 1) \cdot (1 - P)]^k \cdot (n^{1/d})^{d-k}$$

Onde:  $\text{Inteiro}(x)$  é a parte inteira do número real  $x$ .

$$W = (R/U)^{1/k}$$

$$P = W \cdot n^{1/d} - \text{Inteiro}(W \cdot n^{1/d})$$

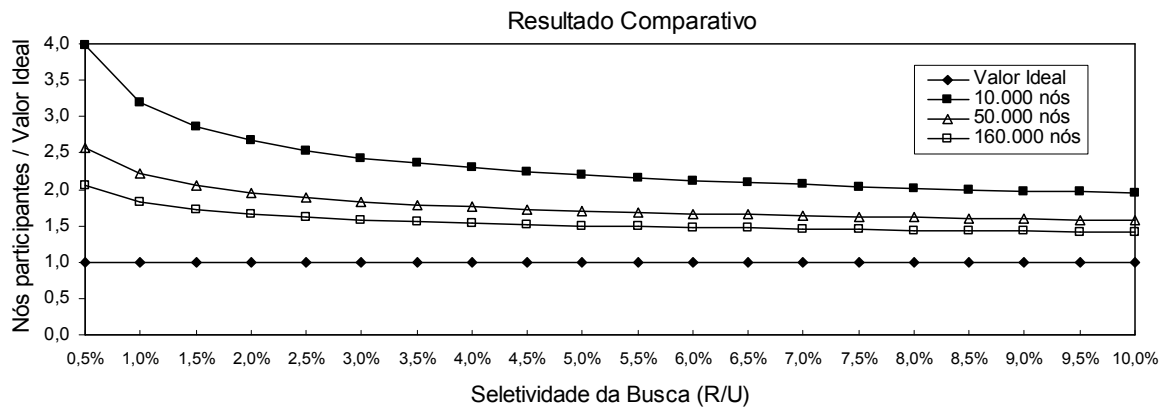
Na expressão acima,  $P$  representa a probabilidade da defasagem entre os limites da busca em uma dimensão e as regiões de responsabilidade dos nós implicar na participação de 2 nós a mais naquela direção. Já  $W$  representa a restrição da busca em cada um dos índices utilizados.

### 3.2 Simulações

Foram realizadas várias simulações, com o objetivo de comprovação da previsão teórica e observação do comportamento da arquitetura em comparação a resultados ideais e frente a mudanças nos parâmetros considerados. Também foram realizadas simulações para identificação do impacto da utilização da técnica de disponibilidade

proposta e do efeito da existência de tipos de recursos que utilizem números diferentes de índices. Para cada um dos pontos das simulações, foram realizadas 1000 buscas na rede construída, iniciadas individualmente por nós aleatórios. As regiões buscadas também foram definidas aleatoriamente para cada busca, respeitando o tamanho de região definido para cada simulação.

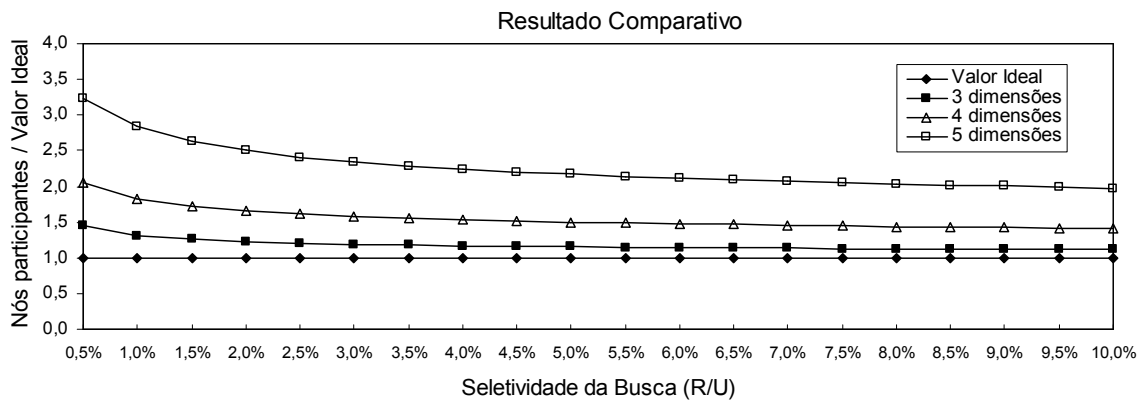
Primeiramente, foram realizadas simulações visando comparar o resultado médio da rede proposta a resultados ideais, isto é, onde a relação entre o número de nós envolvidos na busca e o número total de nós da rede é exatamente igual à relação entre o tamanho da região buscada e o tamanho de todo o espaço d-dimensional. Vemos a seguir os resultados médios obtidos para redes de 4 dimensões, utilizando 4 índices para as buscas, em redes com diversas quantidades de nós (10, 50 e 160 mil). Os resultados apresentados foram normalizados em relação ao valor ideal,  $(R/U)n$ , sendo  $n$  o número de nós da rede.



**Figura 3. Resultados normalizados em relação ao valor ideal**

Percebemos que os resultados das simulações se aproximam bastante dos resultados teóricos esperados, sem haver nenhuma ruptura na tendência de comportamento em nenhum ponto do gráfico. O resultado obtido com redes maiores se aproxima mais do resultado ideal, em todos os pontos. Essa característica é fundamental para a verificação da escalabilidade da solução. Também notamos que a rede se comporta melhor em buscas por regiões maiores, o que também é positivo como garantia de que não há aumento exponencial do número de nós envolvidos na busca e de que as buscas que demandam mais recursos para encaminhamento na rede são executadas de maneira mais eficiente.

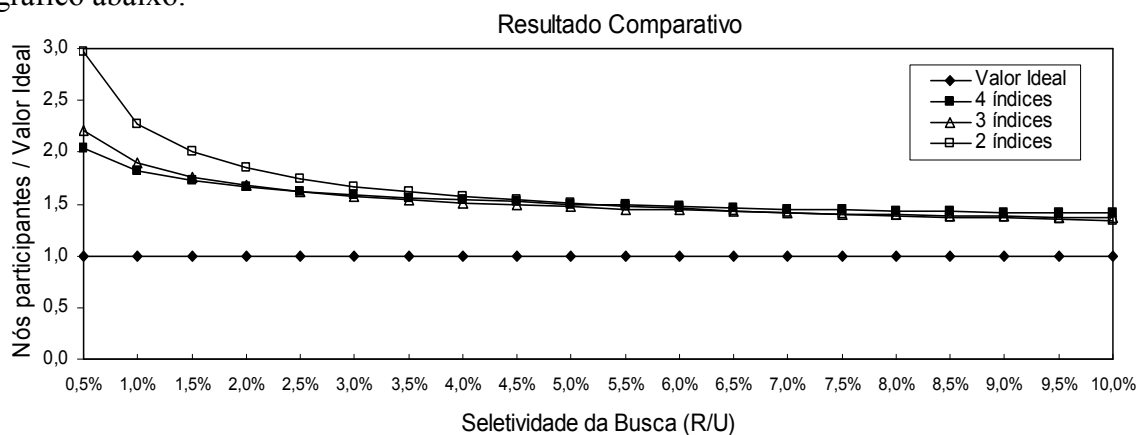
O próximo passo é a verificação do efeito da variação no número de dimensões e índices da rede. Vemos na Figura 4 um gráfico representando buscas por intervalos em redes de 160 mil nós, organizadas utilizando 3, 4 e 5 dimensões (todas as dimensões são utilizadas como índices para buscas).



**Figura 4. Efeito da variação do número de dimensões sobre o desempenho da rede.**

Pelo gráfico podemos perceber um impacto significativo da escolha adequada do número de dimensões da rede na eficiência das buscas. Quanto maior o número de dimensões existentes e utilizadas para busca na rede, maior a flexibilidade em relação aos atributos disponíveis para busca, mas há um impacto negativo no desempenho.

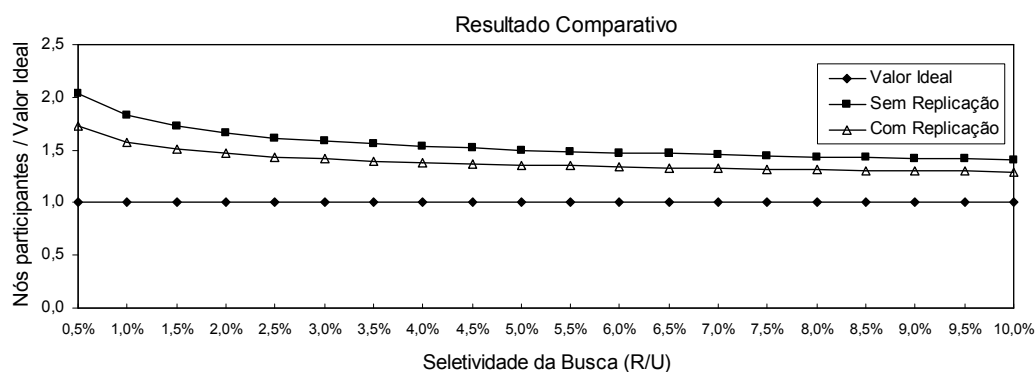
Para verificar a capacidade da rede de suportar diversos tipos de recursos, que utilizem quantidades diferentes de índices para busca, foram realizadas simulações mantendo o número de dimensões (4), a quantidade de nós (160 mil) e variando o número de índices utilizados para busca (2, 3 e 4). Os resultados são apresentados no gráfico abaixo.



**Figura 5. Resultados de buscas por recursos de diversos tipos, utilizando quantidades diferentes de índices.**

Podemos perceber que os resultados para as buscas baseadas em números diferentes de índices convergem rapidamente para desempenhos semelhantes, e próximos àquela de melhor desempenho. Assim, não foi identificado obstáculo à utilização de redes que suportem maior quantidade de índices para a execução de buscas mais simples. Isso indica que uma única rede consegue suportar vários tipos de recurso, com quantidades diferentes de índices.

Por fim, foi testada a utilização da técnica de implementação de disponibilidade proposta. Nela, cada nó replica para si as informações contidas por seu vizinho superior em uma determinada dimensão. Vemos a seguir a comparação do resultado obtido para a rede de 160 mil nós e 4 dimensões, em buscas utilizando 4 índices:



**Figura 6. Efeito da utilização de replicação.**

Podemos perceber que, de fato, a técnica de replicação utilizada causa impacto positivo no desempenho das buscas. Esse impacto é maior, aproximadamente 20% de redução, em buscas menores e se mantém praticamente estável em 10% para buscas maiores.

#### 4. Conclusões e Trabalho Futuro

A arquitetura avaliada endereça de maneira efetiva algumas das mais importantes questões para a implementação de uma infra-estrutura P2P eficiente e flexível para a implementação de serviços novos e em constante evolução. Os resultados teóricos e simulações apontam para a viabilidade da implementação da arquitetura com relação à escalabilidade e suporte a buscas flexíveis. Percebemos que a arquitetura proposta apresenta desempenho melhor para redes maiores, aproximando-se do resultado ideal. Existe também uma característica de troca entre flexibilidade e desempenho, que deve ser considerada para a determinação do número de dimensões da rede.

Há ainda espaço para trabalhos futuros de avaliação da associação da arquitetura a algoritmos mais avançados para balanceamento de carga e disponibilidade.

#### Referências

- Broekstra, J., Ehrig, M., Haase, P., van Harmelen, F., Kampman, A., Sabou, M., Siebes, R., Staab, S., Stuckenschmidt, H. e Tempich, C. (2003) "A Metadata Model for Semantics-Based Peer-to-Peer systems", Proceedings of the WWW'03 Workshop on Semantics in Peer-to-Peer and Grid Computing.
- Crespo, A. e Garcia-Molina, H. (2002) "Semantic Overlay Networks for P2P Systems", Technical report, Computer Science Department, Stanford University.
- Foster, I. e Iamnitchi, A. (2003) "On Death, Taxes and the Convergence of Peer-to-Peer and Grid Computing", Proceedings of the 2nd International Workshop on Peer-to-Peer Systems.
- Handurukande, S., Kermarrec, A., Le Fessant, F. e Massoulié, L. (2004) "Exploiting Semantic Clustering in the eDonkey P2P Network", 11th ACM SIGOPS European Workshop (SIGOPS).

- Karger, D. e Ruhl, M. (2004) "Simple Efficient Load Balancing Algorithms for Peer-to-Peer Systems", Proceedings of the sixteenth annual ACM symposium on Parallelism in algorithms and architectures, páginas 36-43.
- Pitoura, E., Abiteboul, S., Pfoser, D., Samaras, G. e Varziannis, M. (2003) "DBGlobe: A Service-Oriented P2P System for Global Computing", SIGMOD Record, Volume 32, No. 3, páginas 77-82.
- Ratnasamy, S., Francis, P., Handle, M., Karp, R., e Shenker, S. (2001) "A Scalable Content-Addressable Network", Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, páginas 161-172.
- Ratnasamy, S., Hellerstein, J. e Shenker, S. (2003) "Range queries over DHTs", Technical Report IRB-TR-03-009, Intel.
- Rusitschka, S. e Southall, A. (2003) "The Resource Management Framework: A System for Managing Metadata in Decentralized Networks Using Peer-to-Peer Technology", Springer-Verlag GmbH, Lecture Notes in Computer Science, Volume 2530, páginas 144 - 149.
- Stoica, I., Morris, R., Karger, D., Kaashoek, M., e Balakrishnan, H. (2001) "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications", Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, páginas 1-12.
- Talia, D. e Trunfio, P. (2003) "Toward a Synergy Between P2P and Grids", IEEE Internet Computing Volume 7, Edição 4.