

1

Introdução

Recentemente, tem-se observado o aumento da utilização, e consequentemente da relevância, de arquiteturas *peer-to-peer* (P2P) ocasionado, principalmente, pela sua escalabilidade e resiliência no contexto de redes geograficamente distribuídas. Os principais responsáveis por este crescimento foram os sistemas de compartilhamento de conteúdo, dentre os quais pode-se destacar três: Napster, Gnutella[12, 16] e BitTorrent[2]. Entretanto outros tipos de sistemas, com finalidades bem distintas daqueles, também utilizam arquiteturas P2P. Por exemplo, as ferramentas de buscas na web FAROO[4] e YaCy[6], e os sistemas de proxy server CoDeeN[34, 3], de web cache Squirrel[17] e de computação em grade OurGrid[7].

As redes com a arquitetura P2P podem ser classificadas quanto à sua topologia, em estruturadas e não-estruturadas. Nas redes estruturadas a topologia é rígida, enquanto em redes não-estruturadas poucas regras a definem. A técnica mais comum para a criação de topologia em redes estruturadas é *distributed hash table* (DHT) e sua implementação pode ser vista em diversos protocolos para distribuição de conteúdo: Pastry[28], CAN[26], Chord[31], Kademia[22] e Tapestry[35]. A técnica de DHT continua sendo utilizada primordialmente em sistemas de distribuição de conteúdo, porém ela também pode ser encontrada em outros tipos de sistemas. No banco de dados distribuído Apache Cassandra[19, 1], ela é utilizada para a definição do local de armazenamento e na recuperação de dados. Já em [27], é apresentada uma proposta de utilização da técnica de DHT para a localização de recursos em uma grade computacional.

A técnica de DHT possui duas características importantes para sistemas distribuídos baseados em redes P2P que são relevantes em vários tipos de aplicações: a localização eficiente de recursos e a utilização uniforme dos nós. A localização eficiente de recursos é atingida pois em média uma mensagem precisa de poucos *hops* para chegar ao destino e permite que o desempenho da comunicação entre eles seja mantido em um nível mínimo mesmo em redes P2P da ordem de milhares de nós. Já a segunda característica implica na possibilidade de implementar uma maior tolerância a falhas e na distribuição mais uniforme entre todos os nós da rede.

Neste contexto, esta dissertação investigará a aplicação das técnicas de DHT em sistemas de computação distribuída, onde o recurso compartilhado pelos nós é seu processamento. E como estas técnicas podem ser utilizadas

para localizar recursos e para aumentar a cooperação entre os nós de modo a distribuir a carga entre eles e prover tolerância a falhas.

1.1

Motivação

Nos últimos anos, a utilização de redes P2P estruturadas baseadas na técnica de DHT mostrou-se extremamente eficiente em sistemas de localização e compartilhamento de conteúdo, principalmente com a disseminação de sistemas baseados no protocolo BitTorrent. Entretanto, será que a utilização desse mesmo paradigma também é vantajoso para outros tipos de sistemas distribuídos? Será que em redes P2P nas quais o recurso compartilhado não é o armazenamento, e sim o processamento, a técnica de DHT é adequada?

A localização de recursos na técnica de DHT é baseada nos mecanismos de armazenamento e busca de pares chave/valor. Em sistemas de compartilhamento de conteúdo, a chave é utilizada para identificar o arquivo desejado — esta chave pode ser o nome do arquivo ou o *hash* de seu conteúdo, e deve ser única na rede P2P — e o valor pode ser o próprio arquivo ou o endereço onde o mesmo está localizado. Neste aspecto as redes estruturadas formadas pela técnica de DHT se destacam em relação a outras alternativas que utilizam um servidor central ou que são baseadas em inundação. No caso do servidor central, uma falha no servidor inviabiliza a operação de busca, e no caso da inundação, os nós tendem a ficar sobrecarregados a medida que a quantidade de nós na rede aumenta, pois a rede é inundada com mensagens para localização e atualizações.

No contexto de sistemas de computação distribuída, a busca de recursos pode ser vista como a busca pelo nó no qual uma dada operação será executada. Utilizando a técnica de DHT, pode-se associar uma operação/serviço a uma chave de modo que a execução desta operação seja realizada no nó “responsável” pela chave. Outro exemplo são aplicações do tipo SPMD (Single Process, Multiple Data) nas quais os dados de entrada são divididos por múltiplos processos independentes para paralelizar o processamento. A divisão dos dados de entrada pode ser feito utilizando a técnica de DHT.

A cooperação entre nós pode ser alcançada através do roteamento de mensagens e da criação de réplicas de pares chave/valor. O roteamento de mensagens utiliza os nós da rede para enviar a mensagem do nó de origem ao nó de destino e é garantido que o tamanho da rota não será maior que um limite superior. Este limite é da ordem de $O(\log(N))$, onde N é o número de nós na rede, para grande parte das técnicas de DHT, o que permite o roteamento

de mensagens em redes com milhares de nós. A criação de réplicas diminui a carga exercida no nó que está disponibilizando o recurso, ou seja, distribui a carga entre mais de um nó, e aumenta a disponibilidade do recurso na rede, aumentando assim a tolerância a falhas.

Transpondo estes princípios para sistemas de computação distribuída, o roteamento garante uma forma eficiente de troca de mensagens entre os nós utilizando poucos *hops* para atingir o destino, sem a necessidade de manter conexões diretas entre todos os nós participantes. Já a replicação de recursos não ocorre de forma direta, pois não é possível replicar processadores. Porém o conceito de réplica pode ser aplicado de outra forma: no exemplo anterior, de mapeamento de operações/serviços a chaves, em vez de associar uma chave a um único nó, pode-se associá-la a um grupo de nós. Desta forma mais de um nó estará associado a uma operação, o que pode ser visto como replicação da capacidade de processamento, pois uma mesma operação esta sendo oferecida por múltiplos nós. Assim, a execução de operações será distribuída entre os nós membros do grupo o que ocasionará o aumento da disponibilidade dos recursos na rede, e por conseqüência o aumento da tolerância a falhas, e a distribuição de carga entre os nós.

Ao criar uma infra-estrutura para a construção de sistema, é importante definir qual será a API que o programador utilizará. No caso específico da infra-estrutura apresentada neste trabalho deve-se analisar o quanto da camada DHT será exposta a camada de aplicação, por exemplo, se a lista de vizinhos de um nó será acessível pela aplicação.

1.2

Objetivos e contribuições

O objetivo deste trabalho é aplicar a técnica de DHT a sistemas de computação distribuída. Para isto será utilizado o sistema ALua[33, 30], que é uma infra-estrutura para a distribuição de computação, como plataforma de estudo. A comunicação entre os processos será feita pelo roteamento de mensagens dentro da rede P2P, enquanto a criação de réplicas e o balanceamento de cargas serão implementados através de grupos de processos. Além disto serão analisadas quais partes da camada DHT serão expostas para a camada de aplicação.

Como em sistemas de computação distribuída não é comum utilizar a técnica de DHT, as contribuições desta dissertação serão as seguintes:

1. avaliação de protocolos de roteamento de mensagens entre os processos;
2. adição de suporte a grupo de processos;

3. implementação do balanceamento de carga entre os processos (com base em grupos de processos);
4. investigação sobre como deve ser a API da camada DHT.

1.3

Estrutura da Dissertação

O documento está organizado da seguinte forma: o Capítulo 2 contém uma visão geral sobre a arquitetura P2P e em especial sobre a técnica de DHT. Alguns sistemas distribuídos que utilizam a arquitetura P2P são comentados no Capítulo 3, inclusive o ALua que é utilizado com base nas investigações. No Capítulo 4 são detalhadas as aplicações da técnica de DHT acrescentadas ao ALua com algumas considerações sobre sua implementação. Já no Capítulo 5 é detalhada e discutida a aplicação de teste e, em seguida, são apresentados e comentados os resultados dos testes experimentais. Por fim, no Capítulo 6 são discutidas as conclusões encontradas nesta dissertação e apontados trabalhos futuros.