

RESENHA 1 – 26 de Setembro de 2011  
ACH2026 - Redes de Computadores (Prof. Dr. Valdinei Freire da Silva)  
Nome: Murilo Galvão Honório NUSP: 6411927

O artigo *Profiling a Million User DHT* relata medições de desempenho do protocolo DHT (tabelas de hash distribuídas) subjacente ao cliente BitTorrent Azureus, em condições de implantação em larga escala e sob carga de trabalho realista. Caracterizam-se a rotatividade, sobrecarga e desempenho nesse ambiente, propondo então um algoritmo de busca modificado, do lado cliente, que reduz o tempo de busca em uma ordem de grandeza, dentre outros benefícios.

O Azureus mantém uma DHT para tratar o gargalo de escalabilidade do protocolo BitTorrent, que prevê um coordenador para a lista de pares previamente conectados. A DHT permite inclusive distribuição sem coordenador central. As medições foram realizadas entre fevereiro e maio de 2007 a partir de clientes PlanetLab (uma organização que disponibiliza serviços para o desenvolvimento de novas tecnologias de rede). Os clientes foram utilizados para rastrear detalhadamente a atividade da rede DHT. Os dados brutos foram disponibilizados para a comunidade e analisados pelos autores.

Essa análise obteve os seguintes resultados: na média, os tempos de sessão são curtos; as buscas são robustas; e o nó mais próximo para uma dada chave é parcialmente inconsistente, do ponto de vista da rede. Na prática, os fatores dominantes que influenciam o desempenho são as falhas e controle de recursos baseado em limite de tempo.

Ilustram-se os principais desafios do algoritmo Kademlia, bem como a maneira como a implementação Azureus os tratou. Os desafios são: a alta rotatividade, pois há pares chegando e partindo rapidamente, tornando a tabela de roteamento obsoleta (o Azureus consulta até 10 pares paralelamente, ao invés de um); manter a consistência, pois buscas concorrentes podem não ter a mesma visão, se os nós partirem prematuramente (Azureus usa replicação de cada valor nos 20 nós mais próximos da chave); e as falhas, pois como o transporte de mensagens é feito via UDP - a detecção de falhas deve ser implementada no nível de aplicação (Azureus aplica tempo de expiração de 20 segundos, considera que um nó cai se não responde a duas mensagens consecutivas e busca por chave a cada 5 minutos, para verificar se a rede está ativa).

Em seguida, os autores apresentam as medições efetuadas nas características principais da DHT Azureus: mensurar a distribuição dos tempos de sessão dos nós na DHT, identificar como se comporta a inicialização e sobrecarga para manter a DHT, avaliar a consistência e persistência e, por fim, identificar a probabilidade de resposta. Tais medições utilizaram uma série de métodos distintos, detalhados no artigo. Algumas, como a duração das sessões, possivelmente possuem viés, dada a natureza do experimento. Chama atenção o fato de que 81% das operações (mensagens) são destinadas à sobrecarga de manutenção da DHT. Foi identificado que a inserção periódica de chaves causa pouco impacto na persistência dos dados. A probabilidade de resposta de uma mensagem foi medida como uma função do tempo após a mesma ter sido enviada, para 45 milhões de mensagens. Observou-se que ela manteve-se notavelmente estável (no curto e longo prazo).

O desempenho é dominado por parâmetros que impactam o atraso de roteamento: paralelismo de busca, tempos-limite fixos e limitação nas taxas de processamento das mensagens (para restringir o uso de recursos). Propõe-se modificações no algoritmo Azureus, buscando encontrar o equilíbrio entre desempenho e sobrecarga. Utilizaram a probabilidade de resposta de mensagem como parâmetro base (calculada com base no rastreamento obtido nas medições), removeram as restrições por tempo-limite e fixaram o número de buscas em paralelo (simplificando o algoritmo). Com isso, obteve-se uma melhoria enorme no tempo médio de busca, sendo que o número de nós contatados decresceu por uma ordem de grandeza.

O artigo expôs, por fim, a necessidade de realização de buscas frequentes nas redes baseadas no Kademlia, em ambientes com ausência de demanda e sob forte rotatividade de pares, como o BitTorrent, visando manter as tabelas de roteamento atualizadas. O algoritmo proposto não foi testado em condições reais, tratando-se de proposta visando obter uma DHT adaptada para suportar serviços sensíveis à latência.