

# Trabalho de Avaliação de Desempenho

Professores: Paulo Aguiar e Daniel S. Menasché

## 1 Introdução

Os objetos de estudo deste trabalho são sistemas peer-to-peer, responsáveis por uma significativa fração do tráfego da Internet nos dias de hoje. Os usuários dos sistemas considerados neste trabalho estão interessados em um arquivo, e dividem-se em *publishers*, *peers* e *seeds*. Os *publishers* estão interessados na *disseminação* do conteúdo e os *peers* estão interessados no *download* do conteúdo. Quando um *peer* conclui o *download* de um conteúdo, este passa a atuar como um *seed*, até sair do sistema.

Cada peer atua tanto como cliente, recebendo dados de outros usuários, quanto como servidor, fornecendo dados a outros peers. Cada peer, ao juntar-se ao sistema, contribui com recursos como banda e memória, e desta forma, a quantidade de recursos no sistema aumenta em função do tamanho da população.

O throughput do sistema é o número médio de usuários que saem do sistema por unidade de tempo. Quando um peer obtém o conteúdo total buscado, ele pode sair imediatamente do sistema ou ainda permanecer como seed por algum tempo, como acontece em redes reais P2P. Caso o seed com o conteúdo completo permaneça no sistema, ele estará contribuindo para o aumento do throughput. Logo, a saída imediata de um peer após ter o conteúdo requerido recebido é um pior caso para o throughput.

Um sistema é dito *escalável* se existe um tempo de download médio que não vai a infinito, mesmo quando o tamanho da população cresce. Em outras palavras, em um sistema escalável o throughput aumenta com a taxa de entrada de usuários. Uma pergunta chave que queremos responder com este trabalho é: para que conjuntos de parâmetros os sistemas peer-to-peer são escaláveis?

## 2 Modelo

Considere um arquivo dividido em  $B$  blocos. Seja  $\mathcal{B} = \{1, 2, \dots, B\}$  o conjunto de blocos do arquivo, e seja  $\mathcal{C}$  o conjunto de partes de  $\mathcal{B}$  (ou seja,  $\mathcal{C}$  é o conjunto formado por todos os subconjuntos de  $\mathcal{B}$ ). Um publisher estável possui todos os blocos do arquivo e está sempre online no sistema. Peers chegam ao sistema de acordo com um processo Poisson com taxa  $\lambda$ , ou seja, o tempo entre chegadas de peers é distribuído exponencialmente com taxa  $\lambda$ . Peers chegam sem nenhum bloco, e estão interessados no download do arquivo. Peers obtêm blocos do arquivo do publisher e de outros peers e seeds. Um peer que possua  $c \in \mathcal{C}$  blocos do arquivo é referido como um peer com assinatura  $c$ , ou ainda um peer *do tipo*  $c$ . Um peer, ao chegar ao sistema, não possui nenhum bloco (sua assinatura é  $\emptyset$ ).

Cada usuário contacta outros peers buscando por oportunidades para fazer upload de um bloco (push). Assim que um peer é encontrado, estabelece-se conexão e possivelmente realiza-se o upload. O processo de busca por oportunidades de upload então reinicia.

Os peers e seeds fazem upload em instantes determinados por um relógio interno com tempo entre tiques exponencialmente distribuído com taxa  $\mu$  (relógio Poisson). O relógio do publisher opera com uma taxa  $U$  diferenciada.

O tempo de upload de um bloco em um rede P2P real envolve tanto o tempo de busca de um vizinho como o tempo de transmissão confiável de um bloco entre os peers. Este tempo de upload é uma variável aleatória complexa que assumimos ter um comportamento exponencial para simplificação. Dessa forma, quando o upload de um bloco ocorre em um determinado instante, o final do upload é assumido ser o próximo tique do relógio, quando um novo upload poderá ser executado. Neste modelo, cada peer processa apenas um upload por vez. O tempo médio de upload de um bloco é assumido ser o tempo médio da distribuição exponencial de cada relógio, que tem valores diferentes entre peers e publisher. Na simulação, todos os procedimentos de busca, consulta e transferência para upload de um bloco podem ser assumidos ocorrer em tempo zero, sendo que o tempo do próximo tique é que de fato indicará o final da operação. Os parâmetros do modelo estão resumidos na Tabela 1.

Para cada oportunidade de transmissão, precisa-se tomar duas decisões, nesta ordem: 1) para quem transmitir? 2) qual bloco transmitir? As respostas a estas duas perguntas determinam a política de seleção de vizinhos e a política de seleção de blocos, respectivamente.



Parâmetro	Cenário 1	Cenário 2	Cenário 3	Cenário 4	Cenário 5	Cenário 6
$\lambda$ (peers/s)	(0.1, 0.5, 0.9)	(0.1, 0.5, 0.9)	0	0	0	0
$B$ (blocos)	1	1	1	2	2	2
$\mu$ (blocos/s)	-	0.1	-	1	1	1
$U$ (blocos/s)	1	1	1	1	1	0.5
$\gamma$ (1/s)	$\infty$	0.1	$\infty$	1	$\infty$	1
$p$	0	0	1	1	1	1
População inicial	Sistema aberto		Sistema fechado			
$N_0$ (usuários)	0	0	(1 ... 50)	(1 ... 50)	(1 ... 50)	(1 ... 50)

Tabela 2: Cenários de interesse

### 3 Objetivo

O objetivo deste trabalho é avaliar como a vazão dos sistemas peer-to-peer varia em função dos diferentes parâmetros indicados na Tabela 1 e das políticas de serviço do publisher, peers e seeds.

O estado inicial do sistema é de uma população de  $N_0$  peers, todos eles sem nenhum bloco do arquivo.

#### 3.1 Arquivo com um bloco

1. simule o sistema no cenário 1 (Tabela 2). Plote a pmf do número de usuários no sistema, e compare com aquela de uma fila  $M/M/1$ .
2. simule o sistema no cenário 2.

- (a) calcule o tempo médio de *download* em cada um dos cenários 1 e 2 na Tabela 2, e preencha a seguinte tabela, considerando intervalos de confiança de 95%

$\lambda$	$\mu$	$\gamma$	Tempo médio de download	Limite inferior do intervalo de confiança	Limite superior do intervalo de confiança
0.1	-	$\infty$			
0.5	-	$\infty$			
0.9	-	$\infty$			
0.1	0.1	0.1			
0.5	0.1	0.1			
0.9	0.1	0.1			

Tabela 3: Seis conjuntos de parâmetros com arquivo de 1 bloco

- (b) plote a CDF do tempo de *download* nos 6 conjuntos de parâmetros indicados na Tabela 3. Para cada conjunto de parâmetros, você deverá gerar um gráfico. Cada gráfico deve conter  $R$  curvas, onde  $R$  é o número de rodadas que você usou para gerar cada linha da Tabela 3.
- (c) a partir da CDF, calcule a mediana do tempo de *download*, e preencha a seguinte tabela, considerando intervalos de confiança de 95%

$\lambda$	$\mu$	$\gamma$	Mediana do tempo de download	Limite inferior do intervalo de confiança	Limite superior do intervalo de confiança
0.1	-	$\infty$			
0.5	-	$\infty$			
0.9	-	$\infty$			
0.1	0.1	0.1			
0.5	0.1	0.1			
0.9	0.1	0.1			

Tabela 4: Seis conjuntos de parâmetros com arquivo de 1 bloco

- (d) discuta o efeito dos peers permanecerem no sistema como seeds após concluírem seus downloads
3. simule o sistema fechado no cenário 3 na Tabela 2. Plote a curva da vazão do sistema em função do número de peers. Inclua na curva os intervalos de confiança de 95%.

## 3.2 Arquivo com dois blocos

### 3.2.1 Random peer/Random useful piece

1. simule o sistema nos cenários 4, 5 e 6 na Tabela 2. Plote a curva da vazão do sistema em função do número de peers. Inclua na curva os intervalos de confiança de 95%.
2. compare as curvas obtidas para os cenários 4 e 5 na Tabela 2, discutindo o efeito de os peers saírem imediatamente do sistema após concluírem seus downloads.
3. compare as curvas obtidas para os cenários 4 e 6 na Tabela 2, discutindo o efeito de o publisher ter sua capacidade de serviço aumentada.

### 3.2.2 Random peer/Rarest first piece

Repita os itens da Seção 3.2.1, mas considerando agora que os usuários (peers, seeds e publisher) adotem a política random peer/rarest first piece, segundo a qual o bloco a ser transmitido é o mais raro no sistema dentre aqueles que o emissor possui e o receptor não.

### 3.3 Arquivo com dez blocos

Repita os itens da Seção 3.2.1, mas considerando agora um arquivo com 8 blocos ao invés de 2.

### 3.4 Variando as condições iniciais

Considere o cenário 5, no caso  $N_0 = 50$  (vide Tabela 2), sob duas condições iniciais distintas:

- todos os peers possuem o bloco 2 e não possuem o bloco 1;
- nenhum peer possui nenhum bloco (como anteriormente).

Estime o tempo médio que a população leva para alcançar estado estacionário sob estas duas condições iniciais.

### 3.5 Desafio/Competição: Otimizando o sistema

Considere o cenário 5, com  $N_0 = 50$  (vide Tabela 2). Assuma que tanto o publisher quando os peers têm liberdade para variar suas políticas de seleção de vizinhos e de blocos. Experimente diferentes possibilidades, discuta-as e apresente aquela que maximiza a vazão do sistema.

A equipe que apresentar a política que gere maior vazão ganhará bonus.