1
Introdução

Uma máquina de busca precisa manter um repositório local de páginas *Web* para que possa responder às consultas dos usuários de forma eficiente (Bae99, Cho01). Entretanto, surge o problema de manter este repositório atualizado, visto que constantemente ocorrem modificações nas páginas *Web*, e a maioria destas modificações não são notificadas às máquinas de busca. Quando o repositório tem páginas obsoletas, a resposta a uma consulta à máquina de busca pode conter *links* para páginas que não existem mais, ou *links* para páginas que já não estão mais relacionadas à consulta. Portanto, para evitar que a cópia local de uma página permaneça obsoleta, é necessário realizar o *download* periódico desta página. Esta operação é chamada de *revisitação/atualização*, e é realizada pelo *Web crawler*. Estes conceitos são ilustrados na Figura 1.1. Devido à importância comercial, pouco se sabe como os *Web crawlers* são implementados pelas principais máquinas de busca.

Ao longo do texto dizemos que uma página foi *modificada* quando a cópia no servidor *Web* sofreu alguma alteração. Por outro lado, dizemos que a página foi *revisitada/atualizada* quando sua cópia no repositório da máquina de busca foi substituída por uma versão mais recente obtida através de *requisição* ao servidor *Web* que hospeda a página.

Limitações do ambiente computacional restringem a taxa de revisitações realizadas pelo *crawler*, portanto é necessário definir um escalonamento de revisitações que mantenha o repositório o mais atualizado possível utilizando as revisitações disponíveis. Nesta tese propomos e analisamos estratégias para a construção deste escalonamento, chamadas *políticas de revisitação*. Apresentamos a seguir mais detalhes sobre o problema de revisitação de páginas *Web*. As Seções 1.1 e 1.2 fornecem o objetivo e a metodologia desta pesquisa, respectivamente.

Restrições

Duas limitações do ambiente computacional são consideradas: (i) a taxa de *download* do *crawler* é limitada pela capacidade de processamento e pela capacidade de transmissão do canal de comunicação, e (ii) um tempo mínimo

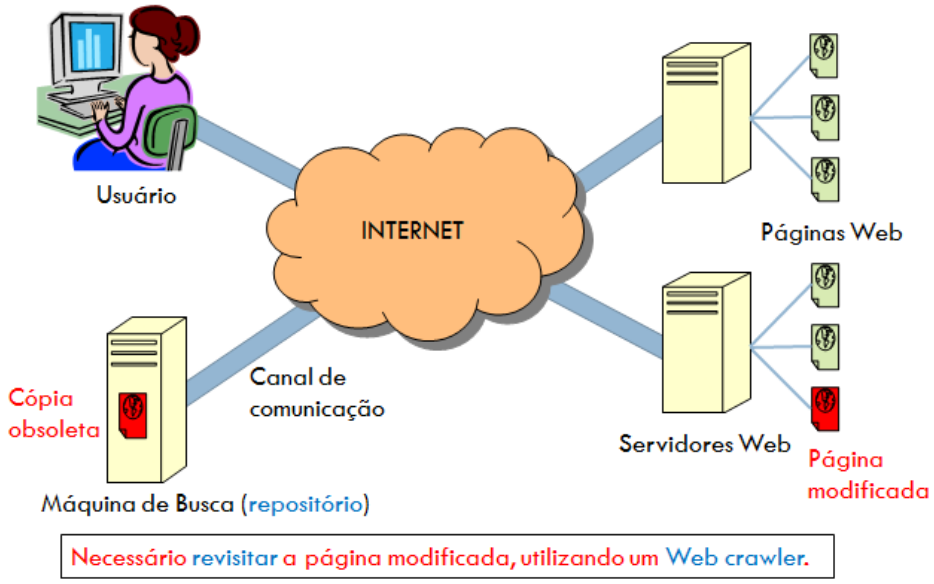

Figura 1.1: Principais elementos do problema de revisitação de páginas *Web*.

deve ser respeitado entre requisições consecutivas a um mesmo servidor, para evitar sobrecarga de servidores *Web*. De fato, uma taxa elevada de requisições a um mesmo servidor pode ser interpretada como um ataque do tipo *denial of service* (DOS10), fazendo com que as requisições futuras do *crawler* não sejam atendidas. Estas restrições são formalizadas pelas Definições 1.1 e 1.2.

Definição 1.1 *A* **restrição de taxa de revisitação** *limita em C* downloads *por unidade de tempo a soma das frequências de revisitações das páginas do repositório. Ou seja, se f_i denota a frequência de revisitação da página i e n o número de páginas no repositório, então*

$$\sum_{i=1}^{n} f_i \leq C. \tag{1-1}$$

Definição 1.2 *A* **restrição de politeness** *estabelece um tempo mínimo P entre duas requisições consecutivas a um mesmo servidor. Ou seja, se $t_{s,j}$ é o instante da j-ésima requisição feita ao servidor s, então*

$$t_{s,j+1} - t_{s,j} \geq P, \quad \textit{para todo servidor } s, \textit{ e toda requisição } j > 0. \tag{1-2}$$

Embora a restrição de *politeness* não tenha sido considerada em algumas políticas de revisitação propostas na literatura (Cho03a, Cof98), dois fatos sugerem grande impacto desta restrição sobre o escalonamento de revisitações. Primeiro, o tempo mínimo entre requisições consecutivas a um mesmo servidor

é tipicamente fixado entre 15 e 60 segundos (Cas04b), o que consiste em um tempo geralmente muito maior que o tempo médio de *download* de uma página. Segundo, temos na Internet uma alta concentração de páginas por servidor, conforme observado em (Cas04b) e em nosso repositório experimental (Seção 3.2). Note que servidores com mais páginas tendem a receber mais requisições. Como um tempo mínimo deve ser respeitado entre requisições a um mesmo servidor, esperamos maior impacto da restrição de *politeness* em servidores com mais páginas.

Por outro lado, os estudos (Shi99, Bro97) indicam que mais de 30% das páginas na Internet são páginas duplicadas. Podemos ter uma redução do impacto da restrição de *politeness* para as páginas que estão replicadas em mais de um servidor, pois a última requisição a alguns destes servidores pode ter ocorrido a mais tempo que o tempo mínimo permitido entre requisições. Uma dificuldade em explorar estas réplicas é o fato de algumas delas poderem estar mais obsoletas que a cópia no repositório da máquina de busca. Não exploramos nesta pesquisa a existência de réplicas das páginas *Web*.

Uso Efetivo do Canal de Comunicação

Para utilizar o canal de comunicação de forma efetiva, a política de revisitação deve distribuir as revisitações uniformemente no tempo. Quando o escalonamento tem períodos com maior concentração de revisitações, podemos observar alternâncias entre períodos de congestionamento e ociosidade do canal de comunicação. Em um período de ociosidade temos desperdício de recursos, e em um período de congestionamento podemos ter atrasos e/ou perdas de requisições. Portanto, é importante que a política de revisitação evite estes picos de requisições para utilizar de forma efetiva o canal de comunicação.

Processo de Modificação das Páginas

Como a maioria das páginas na Internet não comunicam suas modificações para as máquinas de busca, as políticas de revisitação precisam definir o escalonamento com base no padrão de modificação das páginas inferido através dos dados coletados nas revisitações anteriores.

A Suposição 1.3 utilizada nesta tese e em outros trabalhos na literatura (Cho03a, Lee02, Cof98) restringe as modificações das páginas a um *processo de renovação estacionário* (Ros07). Ou seja, a duração dos intervalos entre modificações de uma página é modelada por observações independentes e identicamente distribuídas de uma variável aleatória não negativa. Além disso, nenhum conhecimento é assumido sobre o instante exato de uma modificação, e portanto podemos assumir que o processo estocástico de geração

das modificações foi iniciado há muito tempo, possuindo portanto propriedades estacionárias. Como consequência da Suposição 1.3, o padrão de modificação de uma página i é determinado por uma função de distribuição acumulada $F_i(.)$, que pode ser inferida com base nos dados coletados nas revisitações anteriores.

Suposição 1.3 (Processo de Renovação Estacionário) *A duração do intervalo entre cada par de modificações consecutivas de uma página i é uma observação independente e identicamente distribuída de uma variável aleatória não negativa X_i com função de distribuição acumulada $F_i(.)$. Além disso, o processo de geração das modificações possui propriedades estacionárias.*

Conforme observado em (Cho03a) e nos experimentos realizados nesta tese (Seção 3.3), as modificações da maioria das páginas na Internet podem ser modeladas por um processo de Poisson. Esta propriedade é formalizada na Suposição 1.4.

Suposição 1.4 *As páginas na Internet são modificadas segundo um* **processo de Poisson**. *Portanto, a duração do intervalo entre modificações X_i de uma página i tem função de distribuição acumulada*

$$F_i(x) = 1 - \exp(-\lambda_i x), \tag{1-3}$$

onde λ_i é a taxa de modificação da página i.

Medida do Nível de Atualização do Repositório

Para avaliar a qualidade de um escalonamento de revisitações, precisamos estabelecer uma métrica para o nível de atualização do repositório. A Seção 2.1 descreve várias métricas encontradas na literatura. A medida de atualização do repositório utilizada nesta tese é o *freshness*, proposto em (Cho03a). Esta medida fornece a média no tempo do número de páginas atualizadas no repositório. Uma definição formal do *freshness* de uma página é apresentada na Seção 2.1.1, e o *freshness* do repositório é então definido como a média do *freshness* das páginas ponderada por alguma medida de importância das páginas.

Políticas de Revisitação

A política de revisitação é a estratégia utilizada pelo *crawler* para construir o escalonamento de revisitações, ou seja, ela decide os instantes em que cada página será revisitada.

Quando consideramos a restrição de *politeness*, decidir o instante de cada revisitação de modo a maximizar o *freshness* é uma tarefa computacionalmente

cara. Alguns trabalhos (Wol02, Eck08) fornecem algoritmos polinomiais com solução aproximada. Mais detalhes sobre estas abordagens são apresentados no Capítulo 2. Entretanto, devido ao grande número de páginas e revisitações realizadas por máquinas de busca típicas, consideramos estas abordagens ineficientes. No estudo realizado em (Bro06), estimou-se que os repositórios das principais máquinas de busca contêm mais de um bilhão de páginas.

Políticas eficientes podem ser obtidas fixando regras simples sobre como as revisitações estão distribuídas no tempo. Embora esta abordagem não tenha garantia de solução ótima, geralmente ela facilita a determinação de limites para a qualidade da solução. Por esta razão, adotamos a Definição 1.5 e fixamos *a priori* as possíveis *políticas de tempo*. Esta abordagem foi utilizada em (Cho03a, Wol02), embora em (Wol02) uma política de tempo ótima é determinada.

Definição 1.5 *Uma* **política de revisitação** *determina um* **escalonamento de revisitações**, *que consiste nos instantes* $t_{i,1} < t_{i,2} < \ldots$ *onde serão feitas as revisitações de cada página* i. *A construção do escalonamento é dividida em dois sub-problemas interdependentes:*

Política de tempo: *dada a frequência de revisitação* f_i *de cada página* i, *como distribuir no tempo as revisitações de cada página? Esta regra deve garantir um tempo médio de* $1/f_i$ *entre revisitações consecutivas à página* i.

Alocação de recursos: *dada a frequência total de revisitações realizadas pelo* crawler, *como distribuir esta frequência entre as páginas de modo a maximizar o nível de atualização do repositório? A alocação de recursos ótima depende da política de tempo adotada.*

Por exemplo, podemos estabelecer que o tempo entre revisitações consecutivas de cada página i tem sempre duração $1/f_i$ unidades de tempo. Uma vez fixada a política de tempo, buscamos uma expressão para o *freshness* $A_i(f_i)$ da página i quando esta página dispõe de uma frequência de revisitação f_i. Deste modo, quando não temos a restrição de *politeness*, a alocação de recursos pode ser formulada como

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in R} w_i A_i(f_i) \\ \text{sujeito à} \quad & \sum_{i \in R} f_i \leq C, \\ & f_i \geq 0, \text{ para toda página } i, \end{aligned}$$

onde R é o conjunto de páginas no repositório, w_i é a importância da página i, e C é a frequência total de revisitações realizadas pelo *crawler*.

Note que a restrição de *politeness* é apresentada na Definição 1.2 utilizando os instantes em que ocorrem requisições a cada servidor. Considerar estes instantes aumenta muito o número de variáveis. Além disso, como o *freshness* da página depende dos instantes que a página é revisitada, variáveis inteiras devem ser utilizadas para associar cada revisitação com a correspondente requisição ao servidor. No Capítulo 4 fornecemos um conjunto de restrições que devem ser respeitadas por todo escalonamento que respeita a restrição de *politeness*, e estas restrições dependem apenas das frequências de revisitação das páginas.

Um aspecto prático importante que não é considerado nesta tese é como as estruturas de dados utilizados pelas política de revisitação são gerenciadas em um ambiente distribuído. Devido ao porte dos repositórios de máquinas de busca típicas, geralmente não é possível manter as estruturas de dados na memória principal de uma única máquina, exigindo a distribuição e o gerenciamento destas estruturas em várias máquinas.

1.1 Objetivo e Justificativa

Como as máquinas de busca geralmente possuem repositórios com um grande número de páginas, e seus *crawlers* realizam revisitações em uma taxa muito elevada, consideramos neste estudo que uma política de revisitação é eficiente se o tempo médio para escalonar uma revisitação é sublinear no número de páginas. Além disso, as políticas devem utilizar memória linear no número de páginas, e independente do número de revisitações pois assumimos tempo infinito de operação do *crawler*.

Neste sentido, temos na literatura algoritmos ótimos e eficientes apenas para o caso em que a restrição de *politeness* não é considerada. Embora existam na literatura (Eck08, Wol02) algoritmos polinomiais para o problema com a restrição de *politeness* baseados em técnicas de programação dinâmica (Cor98) ou fluxo em redes (Ahu93), não consideramos estas abordagens eficientes devido ao porte do problema. Conforme discutido no Capítulo 2, o tempo médio para escalonar uma revisitação nestas abordagens não é sublinear no número de páginas.

O objetivo principal desta pesquisa é investigar o potencial da utilização de heurísticas simples e eficientes para otimizar o *freshness* de repositórios *Web*. Ou seja, desejamos saber se vale a pena investir tempo de execução para construir escalonamentos com políticas mais sofisticadas.

1.2 Metodologia

As políticas de revisitação são avaliadas do ponto de vista teórico e experimental. A avaliação teórica é feita através da demonstração de fatores de aproximação para o *freshness* fornecido por cada política, enquanto a avaliação experimental envolve a realização de simulações.

Para obter fatores de aproximação é necessário buscar limites superiores para o *freshness* de qualquer política que produza um escalonamento sujeito à restrição de *politeness*. Em seguida, prova-se limites inferiores para o *freshness* fornecido pela política avaliada. A razão entre estes limites fornece o fator de aproximação.

A avaliação experimental deve reproduzir através de simulações o ambiente real em que são tomadas as decisões sobre o escalonamento de revisitações. Neste caso, é necessário realizar a construção e caracterização de alguns repositórios contendo páginas *Web* reais. Este conjunto de páginas deve ser o mais representativo possível do conjunto de páginas encontradas nos repositórios de máquinas de busca. As simulações devem então mostrar o *freshness* fornecido por cada política caso essa fosse utilizada para manter o nível de atualização dos repositórios experimentais.

1.3 Contribuições

Consideramos duas formas simples de distribuir as revisitações no tempo: (i) as revisitações de uma página são igualmente espaçadas no tempo, e (ii) as requisições a um mesmo servidor são igualmente espaçadas no tempo. No primeiro caso, ajustes devem ser feitos no escalonamento para que a restrição de *politeness* não seja violada. No segundo caso, a restrição de *politeness* é respeitada desde que a frequência de requisições a um servidor não seja maior que o inverso do tempo mínimo permitido entre requisições.

Investigamos uma política igualmente espaçada por página chamada DELAYED, e duas políticas igualmente espaçadas por servidor chamadas RANDOM e MERGE. A política DELAYED parte de revisitações igualmente espaçadas por página, e insere em cada revisitação o menor atraso possível de modo a não violar o *politeness*. Nas políticas igualmente espaçadas por servidor os instantes de requisições aos servidores não precisam de ajustes, mas é necessário decidir qual página revisitar em cada requisição. A política MERGE adota a mesma sequência de páginas revisitadas que a política DELAYED, enquanto a política RANDOM sorteia uma página do servidor com chance proporcional a sua frequência de revisitação definida pela alocação

de recursos. A política MERGE foi desenvolvida nesta pesquisa, enquanto a política RANDOM é proposta na literatura (Cho03a, Cof98). A política DELAYED é natural para o caso em que construímos o escalonamento sem considerar a restrição de *politeness*, e em seguida ajustamos este escalonamento para respeitar a restrição de *politeness*.

As principais contribuições teóricas desta tese são:

- Proposta de um limite superior para o *freshness* do repositório quando aplicamos políticas que respeitam a restrição de *politeness*, chamado limite superior POLITE. Um algoritmo eficiente para alocação de recursos, chamado OPT_POLITE, decorre diretamente da formulação deste limite superior. Note que a solução ótima do problema sem a restrição de *politeness* também fornece um limite superior, chamado de UNPOLITE.
- Prova de expressões analíticas para o *freshness* de uma página quando aplicamos as políticas RANDOM ou MERGE. Com base nestas expressões provamos fator de aproximação 0,77 para o *freshness* do repositório quando aplicamos a política RANDOM, e apresentamos uma conjectura de que este fator de aproximação é superior a 0,927 quando aplicamos a política MERGE. Estes limites foram validados nos resultados experimentais.
- Análise fornecendo um limite superior para a probabilidade de produzir picos de utilização do canal de comunicação. Este limite superior favorece a políticas de tempo igualmente espaçada por servidor.

Dois repositórios experimentais foram construídos e caracterizados para a avaliação experimental das políticas: (i) WEBBASE, com cerca de 14,5 milhões de páginas hospedadas em 5.462 servidores, e (ii) WIKIPEDIA, com 10 mil artigos hospedados em 1 servidor. Mais detalhes sobre estes repositórios são apresentados no Capítulo 3.

O repositório WEBBASE é projetado para conter páginas "típicas"da Internet, e o padrão de modificação de suas páginas pode ser bem modelado por um processo de Poisson. Portanto, o repositório WEBBASE pode ser utilizado para validar os resultados teóricos. As principais contribuições experimentais obtidas através do repositório WEBBASE são:

- Observação do impacto da restrição de *politeness* sobre a qualidade do escalonamento: diferença entre os limites superiores POLITE e UNPOLITE. Note que o limite superior UNPOLITE é justo, visto que a política *fixed-order* proposta em (Cho03a) atinge este limite sem

respeitar a restrição de *politeness*. Observou-se que o impacto da restrição de *politeness* cresce com o aumento da frequência total de revisitação realizada pelo *crawler*, chegando a comprometer 20% de *freshness*.

- Embora nenhum fator de aproximação seja obtido para a política DELAYED, os resultados experimentais mostram que esta política fornece um *freshness* praticamente igual à política MERGE.
- A diferença entre o *freshness* do repositório fornecido pelas políticas DELAYED/MERGE e o limite superior POLITE é inferior a 2,4%, confirmando os resultados teóricos.
- Perdemos mais de 6% de *freshness* do repositório quando utilizamos a política RANDOM ao invés das políticas MERGE/DELAYED. E esta diferença de *freshness* aumenta com a redução da frequência total de revisitação, chegando a 8,2%.
- A alocação de recursos OPT_POLITE, em conjunto com as políticas de tempo DELAYED/MERGE, produziu *freshness* melhor que uma alocação de recursos proposta na literatura para o problema sem a restrição de *politeness*, chegando a uma diferença de 2,8% de *freshness* do repositório.

Uma vantagem do repositório WIKIPEDIA com relação ao repositório WEBBASE é o fato do histórico de modificações dos artigos do Projeto Wikipedia (Wik07) estarem disponíveis, permitindo uma identificação mais precisa do padrão de modificação das páginas. Entretanto, conforme estudo apresentado na literatura (Alm07), e confirmado em nossa análise, os artigos da Wikipedia não se modificam segundo um processo de Poisson. Portanto, este repositório não pode ser utilizado para validar nossos resultados teóricos. Entretanto, este repositório foi utilizado para comparar a política MERGE com uma política gulosa que utiliza informações sobre as últimas modificações para decidir qual o próximo artigo a revisitar. Os experimentos mostraram que embora a política gulosa seja mais bem informada, o *freshness* obtido é cerca de 10% abaixo do obtido pela política MERGE.

Os resultados teóricos e experimentais permitem concluir que políticas simples e eficientes, como DELAYED ou MERGE em conjunto com a alocação de recursos OPT_POLITE, fornecem um *freshness* muito próximo ao melhor que pode ser obtido em ambientes como a Internet, onde temos que respeitar a restrição de *politeness* e temos em média muitas páginas por servidor. Portanto, neste caso existe pouco ganho em se adotar estratégias mais sofisticadas e menos eficientes.

1.4 Organização da Tese

Várias métricas para o nível de atualização do repositório foram sugeridas na literatura, e várias políticas de revisitação foram propostas e avaliadas de acordo com uma ou mais destas métricas. O Capítulo 2 apresenta uma visão destes trabalhos e os relaciona com as contribuições desta tese. Alguns destes trabalhos fornecem a fundamentação teórica utilizada ao longo da tese.

A avaliação experimental das políticas é parametrizada através das características de um repositório de páginas *Web*. A construção e caracterização deste repositório experimental é apresentada no Capítulo 3. Além de permitir simulações mais próximas da realidade, a caracterização do repositório experimental permite justificar (i) a hipótese de modificações das páginas segundo um processo de Poisson, e (ii) o impacto da restrição de *politeness* devido à alta concentração de páginas em poucos servidores.

As políticas de tempo igualmente espaçadas por página e por servidor são investigadas nos Capítulos 4 e 5, respectivamente. Limites superiores para o *freshness* de uma página são apresentados no Capítulo 4, com base em resultados encontrados na literatura para a política de tempo igualmente espaçada por página. Neste capítulo temos também a avaliação experimental da política DELAYED e a observação experimental do impacto da restrição de *politeness*. O Capítulo 5 fornece a prova de limites inferiores para o *freshness* de uma página quando empregamos as política RANDOM e MERGE, bem como a avaliação experimental destas políticas. Temos também neste capítulo a análise indicando baixa chance de congestionar o canal de comunicação quando empregamos a política de tempo igualmente espaçadas por servidor.

No Capítulo 6 empregamos um repositório de artigos da Wikipedia para comparar a política MERGE com uma estratégia gulosa que leva em conta os instantes das últimas modificações. Finalmente, o Capítulo 7 apresenta as principais conclusões e direções de trabalhos futuros.