情報処理学会論文誌　データベース　Vol. 2　No. 3　67–75 (Sep. 2009)

大規模 Web アーカイブ更新のための階層的スケジューリング手法

田 村 孝 之[†1,†2]　喜 連 川　優[†2]

筆者らは Web 情報からの社会知の抽出を目指し，大規模 Web アーカイブの構築を行っている．刻々と変化する Web 情報に追随するため，10 億規模の Web ページのそれぞれについて更新間隔の推定を行い，独立したタイミングで再収集を行う更新クローラの開発を進めてきた．しかし，実際のクローリングにおいては通信リソースや Web サーバにもたらす負荷に上限があり，スケジュールが実施不能に陥ることもある．本稿では，Web ページが属する Web サーバごとにアクセス要求を集約し，通信リソースの割当てを行うとともに，実施不能なスケジュールを検出して Web ページの再収集スケジュールの修正を可能にする階層的スケジューリング手法について提案する．さらに実際のクロールログを用いてその有効性を示す．

A Hierarchical Scheduling Method for Refreshing Large Scale Web Archives

Takayuki Tamura[†1,†2] and Masaru Kitsuregawa[†2]

We are building a large scale Web archive for exploiting societal knowledge from the Web. To catch up with the ever-changing Web, we have developed an incremental crawler, which revisits Web pages individually according to their estimated change frequencies. The real-world crawling, however, suffers from the constraints on the amount of communication resources and the politeness of the server access behavior, causing the revisit schedule to be infeasible. This paper proposes a hierarchical scheduling method, which allocates communication resources to Web servers that accommodate Web pages, and triggers rescheduling upon detection of infeasible schedules. Its effectiveness is convinced using actural crawl logs.

1. はじめに

筆者らは膨大な Web 情報の大域的な構造や時間変化の分析に基づく社会知の抽出に取り組んでおり，その基盤として日本の Web 情報（非.jp ドメインまで含めた日本語 Web ページとその周辺）を網羅的に収集・蓄積した大規模な Web アーカイブの構築を行っている[1),2)]．本 Web アーカイブにおいては網羅性と時間分解能の両立が求められるため，Web ページごとに独立したタイミングで自立的に収集を繰り返す更新クローラの開発を進めてきた[3)–5)]．更新クローラは，**図 1** に示すとおり，既知の Web ページを一定時間後に再収集することで内容の変化をとらえるとともに（refresh path），通常の一括クローラと同様，収集した Web ページから未知の URL を抽出し（discovery path），構造の変化（Web の成長）に備える．

筆者らはすでに，Web ページ再収集におけるアクセスタイミングを決定するため，Web ページごとに更新有無の観測結果から平均更新間隔を推定するスケジューリング手法について述べた[4)]．しかし，実際のクローリングにおいては，さらにクローラシステムや Web サーバの物理的な制約を考慮に入れる必要がある．そこで本稿では，Web ページアクセススケジュールから，Web サーバに対するアクセススケジュールを導出し，Web サーバごとに通信リソースの割当てを行う階層的スケジューリング手法について述べる．本手法は，同一 Web サーバ上の複数 Web ページに対するアクセスを集約することで，Web サーバに及ぼすアクセス負荷の把握・抑制を可能にする．また，過去のダウンロード経過時間の実績を用いることで，所与の通信リソースで Web サーバアクセススケジュールが実施可能かどうか判定する基準を与える．

一方，未知 URL は可能な限り早期にアクセスすることが望ましいため，通信リソースが不足していることを前提に，複数 URL 間の順序付けを行うことがスケジューリングの課題となる[6),7)]．未知 URL 用の通信リソースをより多く確保するためにも，Web ページ再収集における通信リソース割当ての効率化が求められる．

以下，2 章で階層的スケジューリング手法の詳細について述べる．3 章では実際のクローリングログを用いて最小リソース量を見積もるとともに，ログの一部を用いたシミュレーションにより本手法の効果を評価する．4 章で関連研究をあげ，5 章で全体のまとめを行う．

†1 三菱電機株式会社情報技術総合研究所
Information Technology R&D Center, Mitsubishi Electric Corporation
†2 東京大学生産技術研究所
Institute of Industrial Science, The University of Tokyo

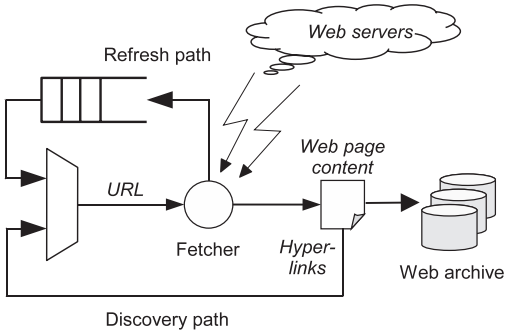

図 1　Web アーカイブ更新クローリング
Fig. 1　Refresh crawling for Web archive.

2. 階層的スケジューリング手法

2.1 全 体 構 成

更新クローリングにおける階層的スケジューリング手法の構成を**図 2** に示す．更新間隔推定（Change Interval Estimation）および再アクセススケジューリング（Revisit Scheduling）は Web ページごとに独立して行われる[4]．直列化（Serialization）は Web サーバに対するアクセスマナーへの配慮から，同一 Web サーバに属する Web ページを逐次にアクセスするための処理であり，各 Web ページのアクセス間隔をもとに Web サーバに対するアクセス間隔が決定される．通信リソース割当て（Communication Resource Assignment）は，Web サーバを対象に通信リソースプール（Communication Resource Pool）からのリソース割当てを行う．

ここで，Web サーバは Web ページ URL の authority 部分[8]（ホスト名とポート番号の組）で識別される論理的な対象を指すものとする．Web サーバ名とサーバ H/W の対応は，クラスタリング，仮想ホストおよび仮想マシン等の実装技術により一般には多対多となり，外部から正確に把握することはできない．以降の議論を単純化するため，Web サーバ名ごとに十分な H/W リソースが背後に存在するものと仮定する．また，通信リソースは BSD ソケット等通信端点を指し，通信リソースプールは新たに生成可能な通信端点数を管理するものとする（通信帯域の明示的な割当て等は，不特定 Web ページのダウンロードにおいては現実的でない）．通信リソースを割り当てられた複数の Web サーバに対しては並行してアクセスを行う．

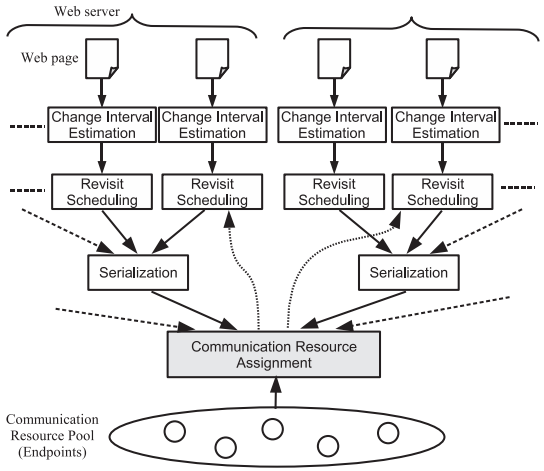

図 2　階層的スケジューリング手法の構成
Fig. 2　Architecture of the hierarchical scheduling method.

通信リソース割当ては，直列化により決定された Web サーバアクセス間隔がタイミング制約となるため，リアルタイムスケジューリングの問題ととらえることができる．ここで，Web サーバアクセスが周期的タスクに相当し，Web サーバアクセス間隔がタスクの実行周期となる．リアルタイムスケジューリングはさらにタスクの実行時間を必要とするが，Web ページダウンロードの経過時間がこれに該当する．そのため，ダウンロード経過時間の実績を把握しておくことが必要となる．

一方，与えられたタイミング制約を満たす通信リソース割当てが実施できないこともある．このような状況では，下位に位置する通信リソース割当てが局所的に対処するのは適切でなく，上位の再アクセススケジューリングにフィードバックし（図 2 矢印），アクセス間隔の延長やアクセス対象 Web ページの取捨選択により Web ページアクセススケジュール自体を定量的に修正することが望ましい[4]．そのためには，このようなスケジュール実施不能状況を的確に検知することも重要である．

次節以降で，直列化による Web サーバアクセス間隔の導出，スケジュール実施不能状況検出のための通信リソース見積り，および通信リソース割当て対象 Web サーバの選択手法のそれぞれについて述べる．

2.2 Web サーバアクセス間隔の導出

ここでは，直列化により，通信リソース割当てに対するタイミング制約（実行周期）となる Web サーバアクセス間隔を導出する過程について述べる．

まず，Web サーバ i に属する j 番目の Web ページを P_{ij} とし，再アクセススケジューリング[4] により設定された P_{ij} の再アクセス間隔を I_{ij}，十分長い期間 τ における P_{ij} のアクセス回数を N_{ij} とする．Web サーバ i へのアクセスを時間 I_i ごとに逐次的に行うという制約の下で各ページの取得が可能となるには，

$$I_i \sum_j N_{ij} \leq \tau \tag{1}$$

が成り立つ必要がある．$N_{ij} = \lfloor \tau / I_{ij} \rfloor \simeq \tau / I_{ij}$ により，式 (1) から以下が得られる．

$$I_i \leq \frac{\tau}{\sum_j N_{ij}} \simeq \left(\sum_j \frac{1}{I_{ij}} \right)^{-1} \equiv {L_i}^{-1} \tag{2}$$

L_i は Web サーバ i に属する Web ページのアクセス頻度の総和であり，Web サーバ i の負荷を表す指標と考えられる．その逆数 ${L_i}^{-1}$ は Web ページ再アクセススケジューリングに従ったアクセスに必要な Web サーバアクセス間隔の理論的な上限となっている．Web サーバに属する Web ページ数の増加や各 Web ページの再アクセス間隔の短縮により Web サーバの負荷指標が大きくなると，より小さな間隔で Web サーバにアクセスする必要が生じる．

実際には，Web サーバへのマナーに配慮してアクセス間隔の下限値 $I_{\min}$ を設け[⋆1]，

$$I_i = \max({L_i}^{-1}, I_{\min}) \tag{3}$$

で定められる I_i を Web サーバ i へのアクセス間隔とする．

2.3 通信リソース見積り

次に，前節の Web サーバアクセス間隔（実行周期）と Web ページダウンロードの経過時間（実行時間）を組み合わせ，Web ページ再アクセススケジュールの実施に必要なリソース量の見積り指標を得る．

Web サーバ i から Web ページをダウンロードするのに要した平均経過時間を e_i とすると，Web サーバ i に対して周期 I_i でダウンロードを繰り返すことによる通信リソース使用率は少なくとも e_i/I_i となる．プリエンプティブな割当てを想定したスケジューリングにおいて，全通信リソースを 100%利用できた場合，必要な通信リソース数 C_p は e_i/I_i の総和と等しくなる[11]．ただし，プリエンプティブな通信リソースの割当ては，1 度開始したダウンロードを中断して別の Web ページのダウンロードを割り込ませることを意味し，Web ページダウンロードの経過時間に対して Web ページ切換えのオーバヘッドが大きく，適切でないと考えられる．一方，非プリエンプティブなスケジューリングでは，必要な通信リソース数を求める問題は NP 困難といわれている[12),13]．そこで，通信リソースの割当ては非プリエンプティブとするが，C_p の値を見積りの下限として用いることにする．

実際の Web サーバにおいては，$e_i/I_i > 1$ となるものも存在し，このような Web サーバは単一の通信リソースを占有するものとする．したがって，S を通信対象 Web サーバの集合とするとき，Web ページ再アクセススケジュールの実施に必要な通信リソース数 C は以下を満たす．

$$C \geq \left\lceil \sum_{i \in S} \min(\frac{e_i}{I_i}, 1) \right\rceil \tag{4}$$

通信リソースプールのリソース量が C を下回っている場合，明らかにスケジュールは実施不能である．

2.4 Web サーバ選択手法

最後に，Web サーバへの通信リソース割当ての具体的な手順を述べる．各 Web サーバは，1) 通信待ち状態，2) ダウンロード状態，3) アイドル状態のいずれかをとり，通信リソースの割当てにより通信待ち状態からダウンロード状態に遷移し，ダウンロード完了後，アクセス間隔調整のためのアイドル状態を経由して次の通信待ち状態に戻るものとする．

ここで，通信リソースに空きが発生するつど，通信待ち状態の Web サーバから割当て対象の Web サーバを選択する方式を考える．このような動的スケジューリングにおいては，EDF（Earliest Deadline First）アルゴリズムが最適であることが知られており[12]，これに基づいて次のように Web サーバの選択を行う．すなわち，時刻 t において通信待ち状態に入った Web サーバ i の次回ダウンロード開始期限 D_i を以下のように定め，D_i が小さい Web サーバに高い優先度を与える．

$$D_i = (\lfloor t/I_i \rfloor + 1) \cdot I_i - e_i \tag{5}$$

通信リソースが豊富にある場合，Web サーバ i からのダウンロード回数を k_i とすると，$D_i = (k_i + 1)I_i - e_i$ となる．すなわち，期間 I_i ごとに 1 回のダウンロードをスケジュー

⋆1 連続する Web ページアクセスの間隔をどの程度空けるべきかについては様々なガイドラインが存在するが，Web サイト上の `robots.txt` ファイルにおける `Crawl-delay` フィールドにより，管理者が明示的に指定することが広まりつつある[9),10]．

ルする．一方，通信リソースが十分でないとダウンロードが実施できず，$k_i < \lfloor t/I_i \rfloor$ となることもある．式 (5) において，k_i ではなく $\lfloor t/I_i \rfloor$ を用いることは，実施されなかったタスクをキャンセルすることに相当する．クローリングにおいては現在の状態を観測することしかできないため，このようにすることが合理的である．

なお，期間 I_i 内で 2 回ダウンロードを行うことを避けるため，時刻 t でアイドル状態に入る Web サーバ i のアイドル状態継続時間 Q_i を，

$$Q_i = \max(I_{\min}, \lceil t/I_i \rceil \cdot I_i - t) \tag{6}$$

とする．また，複数通信リソースが利用可能となっている場合は，通信リソースまたは通信待ち Web サーバのいずれかが尽きるまで処理を繰り返す．

3. 大規模クロールログを用いた評価

3.1 Web サーバ特性とリソース使用量の見積り

階層的スケジューリングの挙動をシミュレーションにより分析するため，大規模な実クロールログからパラメータの抽出を行った．本節では，クロールログに含まれる Web サーバに関する特性値と，それに基づく通信リソース使用量の見積りについて述べる．

ダウンロード経過時間（e）の実績値を得るため，2007 年 5 月から 6 月にかけて延べ 28 日間に実施した 15,745,505 件のダウンロードのログを用いた（非 HTML，非成功応答，robots.txt は除外）．ログ中の一意 URL 数は 11,697,860 件であり，2 回以上アクセスした URL も含まれる．また，一意な Web サーバ数は 82,034 であった．このクロールログは 32 ノードで並列に実施した網羅的クロールの 1 ノード分であり，全体では約 250 万 Web サーバがアクセス対象であった（URL は Web サーバ名のハッシュ値に基づいて各ノードに分散）．なお，実クロール自体は，Web サーバ選択アルゴリズムとして 2.4 節の EDF ではなく後述する FCFS を用いており，メンテナンスのための中断期間を含む等，最適なアクセススケジューリングには基づいていない．そのため，クロールログからは Web サーバごとのダウンロード経過時間の列のみを抽出し，ダウンロード時刻やダウンロード回数は無視した．実クロールのパラメータ（ノードあたり通信リソース量 $C = 100$，Web サーバアクセス間隔の最小値 $I_{\min} = 10$ 秒，等）も次節のシミュレーション結果には大きな影響を及ぼさないものと考えられる．

図 3 は，クロールログに含まれる 82,034 Web サーバについて，クロールログ終了時点における Web サーバアクセス間隔理論値（I）の分布を示したものである．この値の逆数は Web サーバに属する Web ページの更新間隔推定値の逆数の総和として与えられ（式 (2)），

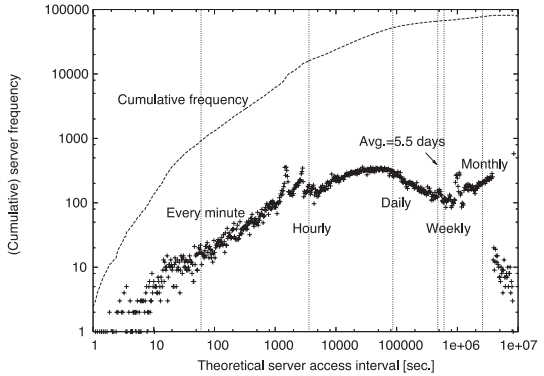

図 3　実クロール時の Web サーバアクセス間隔理論値の分布
Fig. 3　Distribution of theoretical web server access intervals.

クローリングの進捗とともに修正が加えられるが，単純化のため，以降の議論では Web サーバごとに一定の値をとるものとして扱う．図 3 に示したように，平均で 5.5 日に 1 回となっており，Web アーカイブの時間分解能である 1 日より大きな間隔をとる Web サーバが多いことが分かる．1 分以内にアクセスする必要があるのは全体の 1%程度である．その一方で，理論上は毎秒アクセスする必要がある Web サーバも存在している．

図 3 の値には 2007 年 5 月以前の実クロールの結果も反映されており（Web アーカイブの最も古いデータは 1999 年 7 月のもの[1]），アクセス間隔の大きな領域での精度は十分高いと考えられる．一方，アクセス間隔が小さい領域では実クロールの性能が不足し，Web ページ更新を見逃すことにより，精度が低下している可能性がある．この問題に対しては，本稿の結果を適用して実クロールの性能を高め，再度 I の値を算出するという手順を継続的に実施していく必要がある．

図 4 は Web サーバごとの平均および最大ダウンロード経過時間（e）の分布を示す．Web サーバごとの平均ダウンロード経過時間の全 Web サーバに関する平均は約 1 秒であり，最大ダウンロード経過時間の平均は 5.8 秒となっている．ダウンロード経過時間に影響を与える Web ページのサイズは，63 Byte から 2 MByte（クロール時の上限）まで広く分布し，平均では 22 KByte であった．

図 5 は，図 4 のダウンロード経過時間（e）と図 3 の Web サーバアクセス間隔理論値（I）の比（e/I）で定まる通信リソース使用率の分布を示したものである．ほとんどの Web

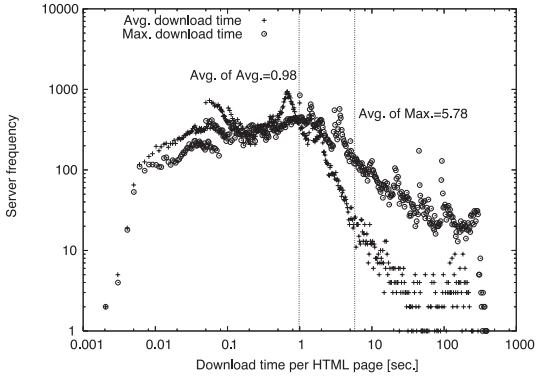

図 4　実クロールログにおける Web サーバごとのダウンロード経過時間の分布
Fig. 4　Distribution of download time per server.

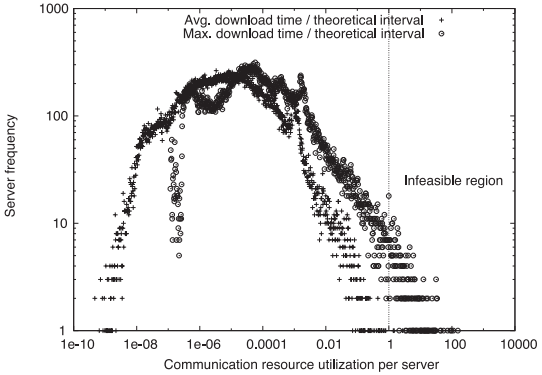

図 5　実クロールログにおける Web サーバごとの通信リソース使用率の分布
Fig. 5　Distribution of communication resource utilization per server.

サーバで通信リソース使用率は 1%未満の非常に小さな値をとっていることが分かる．また，使用率が 1 を超える Web サーバについては，逐次的なダウンロードでは Web サーバアクセス間隔理論値を達成できないため，理想的なスケジュールが実施不能であることを表している．平均ダウンロード経過時間に対応する平均使用率が 1 を超えるもの（定常的に実施不能）は 10 未満であるが，最大ダウンロード経過時間に対応する最大使用率は，約 380 の Web サーバにおいて 1 を上回った．これらの Web サーバでは，一時的にスケジュールが正しく実施できなくなる可能性がある．

各 Web サーバの通信リソース平均使用率を用い，式 (4) に基づいて見積もった通信リソース数下限値は 62 となった．ただし，平均使用率が 1 より大きい Web サーバに対しては，通信リソースを 1 つずつ割り当てるものとした．また，最大使用率が 1 を超える Web サーバを除外すると，通信リソース数下限値は 28 となった．以下，最大使用率が 1 を超える Web サーバを実施不能サーバ，それ以外の Web サーバを実施可能サーバと呼ぶ．

3.2　シミュレーション結果

次に，前記のクロールログに含まれる個々の Web ダウンロード経過時間と，対応する期間の Web サーバアクセス間隔理論値を用いて，シミュレーションにより通信リソース割当て手法の評価を行った．Web サーバ選択手法として以下を評価対象とし，EDF との比較を行った．

- EDF（Earliest Deadline First）
  次回ダウンロード開始期限 D_i（式 (5)）が小さい Web サーバ i に高い優先度を与える．
- FCFS（First Come First Served）
  Web サーバが通信待ち状態に入った順に高い優先度を与える．
- FREQ（Frequency-based）
  アクセス間隔 I_i が小さい Web サーバ i に高い優先度を与える．
- RM（Rate Monotonic）
  通信リソース使用率 e_i/I_i が大きい Web サーバ i に高い優先度を与える．
- SJF（Shortest Job First）
  平均ダウンロード経過時間 e_i が小さい Web サーバ i に高い優先度を与える．

シミュレータは各 Web サーバを通信待ち状態として待ち行列に投入し，いずれかの Web サーバ選択手法に基づいて通信リソース割当て対象の Web サーバを選択する．次に，クロールログから当該 Web サーバのダウンロード経過時間を取り出し，ダウンロード状態の持続時間とする（3.1 節で述べたとおり，ダウンロード時刻や他の Web サーバとの前後関係は用いない）．その後，式 (6) の Q_i で定まる時間をアイドル状態とし，再度通信待ち状態として待ち行列に投入するという動作を繰り返す．なお，アクセス間隔下限値 $I_{\min}$ は 1 秒とした．また，図 3 に基づき，すべての Web サーバが少なくとも 3 回（スケジュール実施不能でない場合）ダウンロードを行うよう，シミュレーションの継続期間は 300 日間とした．ログの末尾に到達した場合は先頭に戻ってログを繰り返し用いた．

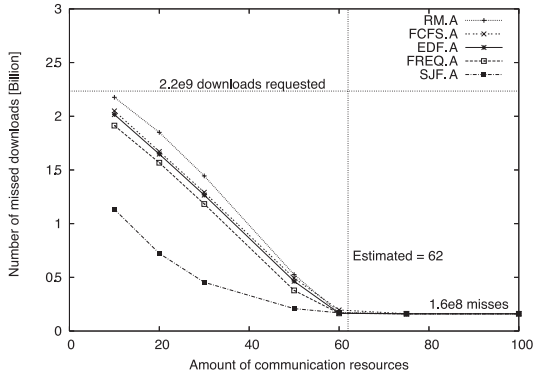

図 6　取得漏れ数のシミュレーション結果（全サーバ対象）
Fig. 6　Simulation result of number of missed downloads (with all servers).

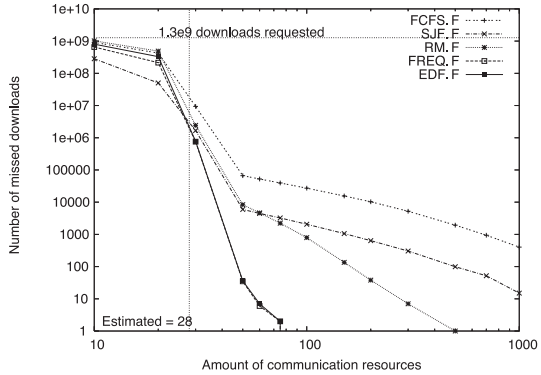

図 7　取得漏れ数のシミュレーション結果（実施可能サーバ対象）
Fig. 7　Simulation result of number of missed downloads (with feasible servers only).

図 6 は，利用可能な通信リソース数に対する取得漏れ数のシミュレーション結果を示したものである．取得漏れ数は，シミュレーション期間を各 Web サーバのアクセス間隔理論値で割ったものの総和と，実際に実行できたダウンロード数との差である（式 (6) の Q_i の設定により，Web サーバアクセス間隔理論値の期間ごとにたかだか 1 回のダウンロードが実施されることに注意）．また，垂直線は，通信リソース消費量の下限見積り値を表し，水平線は理想的なダウンロード数（シミュレーション期間を各 Web サーバのアクセス間隔理論値で割ったものの総和）に対応し，取得漏れ率 100%に相当する．

通信リソース量を見積り値付近まで増加させると取得漏れ数が急激に減少しているが，1.6 億件の取得漏れが残っている．これは，一部の Web サーバが実施不能サーバであり，単一の通信リソースを占有しても取得漏れの発生を防ぐことができないためである．そこで，以下の評価では実施可能サーバのみを対象としてシミュレーションを行った．本来は，上位のスケジュールを変更してすべての Web サーバを実施可能サーバとすべきであるが，除外した実施不能サーバについては，別の通信リソースプールを用いて修正後のスケジュールを適用すると考えればよいので，結果の有効性は失われない．

図 7 が取得漏れに関する結果である．図 6 と同様，通信リソース量が不足する領域では SJF が優位であり，RM の漏れが若干大きいものの，他の手法は同様の傾向を示している．SJF は高速な Web サーバからのダウンロードを優先するので，低速な Web サーバからのダウンロードにより通信リソースが占有されることを防いでいる．一方，図 7 で通信リソース量を増やした領域では，漏れの割合は全体に対して小さくなっているものの，EDF および FREQ 以外では完全には 0 になっていない．特に SJF では，単純な FCFS と同様の振舞いを示し，見積り値の 30 倍以上のリソースを与えても漏れが生じてしまう．これに対し，EDF と FREQ は，見積り値の約 3 倍のリソースにより，漏れが解消している．約 8 万の Web サーバに対する更新クローリングを，80 多重の通信リソースでスケジュールどおり実現できることになる．この値は，PC 1 台で十分に提供可能な範囲であり，ノード数を増加させることにより，容易に大規模 Web に対応することができる．なお，クローラの通信リソース制約によっては，漏れのない Web ページ再収集に必要な通信リソースを確保することができない状況も生じる．このような場合は，無秩序な取得漏れを防ぐために，一定の基準に基づいて Web ページアクセススケジュールの修正を行う（たとえばアクセス間隔を一律 2 倍にする，アクセス間隔の下限を 2 日にする，等）必要がある[4]．

FREQ は EDF とほとんど変わらない挙動を示しているが，前者ではダウンロード経過時間 e を用いていない．この値は Web サーバアクセス間隔 I に対して十分小さく，無視しても影響がほとんどないことを意味している．したがって，FREQ は EDF の簡易な代替手法として用いることができる．ただし，FREQ が SJF より優位なのは通信リソースが十分に存在する場合であり，その条件の成立を判断するために e を観測しておく必要がある．更新クローリングの実施においては，単純にアクセス間隔を制御するだけではなく，通信相手の Web サーバの応答性能を監視することが性能維持のために重要といえる．

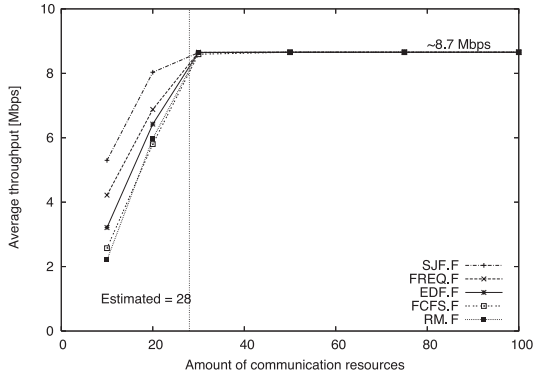

図 8　ダウンロード速度のシミュレーション結果（実施可能サーバ対象）
Fig. 8　Simulation result of download throughput.

なお，3.1 節で述べたとおり実クロールでは再収集スケジュールを正しく実施できておらず，約 56 万ダウンロード/日にとどまっているのに対し，シミュレーションでは約 433 万ダウンロード/日（実施可能サーバ対象の場合）に達している．本シミュレーションは実クロールの単純な再現ではなく，実クロールの性能や規模が変化しても，上記の結果がただちに無効になることはないといえる．

図 8 は，ダウンロードの平均スループットを示している．十分なリソースが与えられた場合，スケジュールの実施には 8.7 Mbps のバンド幅が必要となることが分かる．この値自体は，一般的な PC で容易に達成可能な値であるが，実クロールがこの 32 倍の規模であったことを考慮し，実施不能サーバの効果としてさらに 2 倍のバンド幅が必要になると仮定すると，トータルのバンド幅は約 550 Mbps となる．今後の Web の成長や更新頻度の増加に対応するには，ネットワーク経路がボトルネックになる可能性があり，その際は複数拠点化が必須になると考えられる．

以上をまとめると，Web サーバを単位とする更新クローリングの挙動は，Web サーバアクセス間隔理論値と Web ページ平均取得時間により求められる必要リソース数の下限値を境に大きく変化し，リソース不足状態では SJF 方式が，リソース充足状態では EDF または FREQ 方式が，適していることが分かった．リソース不足状態を許容することは，Web サーバごとの取得状況がバランスを欠くことにつながるため，むしろスケジュールの修正を行うべきであるが，そのための判断基準として上記の Web サーバ特性値を把握することが重要である．

4. 関 連 研 究

本稿では，Web ページアクセスにおけるタイミング制約と通信リソース制約を同時に扱っているが，クローラに関する文献でこれらに言及したものはほとんどない．一括クローラ（batch crawler）に関する研究では，有用度の高い Web ページを早期に収集するための Web ページ優先度制御が考慮されているが，タイミング制約は存在しないため扱われていない[6),7),14)]．WebFountain クローラは更新クローラであり，各 Web ページの更新確率やページ収集後の経過時間等を変数とする非線形連立方程式を一定期間ごとに解いて収集対象 Web ページを決定する[15)]．クローラリソースの制約は考慮されているものの，Web サーバ単位での制御は行われていない．また，恒常的なリソース不足を検出し，アクセス戦略（時間分解能，収集範囲等）を定量的に修正することもできない．Web ページ再アクセスタイミングに関する Olston らの研究[16)] についても同様である．

Cho ら[17),18)] はクローラリソースが不足する場合に，Web ページの更新頻度に基づいてアクセスの優先度を決定する方法を論じており，特に，更新頻度が非常に高い Web ページは取得した内容が最新である期間も短いため，他の Web ページの収集を優先した方がスナップショット全体の新鮮度が向上するという結論を導いている．しかし，この議論は最新情報の検索が目的であることを前提としたものであり，過去の任意の時点における分析を目的とするアーカイブ用途には必ずしもあてはまらない．User-centric クローリング[19)] はリソース負荷を低減するために検索ニーズに応じて Web ページに重み付けするものであるが，収集対象に偏りを生じることになるため，やはりアーカイブ用途には適さない．

リアルタイムスケジューリングにおける研究の多くはプリエンプティブな CPU タスクを扱っているが[11)]，更新クローラは 2 章で述べたように非プリエンプティブなリソース割当てを行うため，これらを直接適用することはできない．非プリエンプティブ EDF（Earliest Deadline First）アルゴリズムはこのような場合に最適であることが知られているが，その厳密な解析は困難であり，スケジュールの実施可能性を容易に判断する手法は得られていない[12),13)]．

5. ま と め

本稿では，大規模 Web アーカイブ構築のための更新クローラにおいて，適応的に決定した Web ページ周期アクセスのスケジューリング結果を Web サーバごとに集約し，通信

リソースの割当てを行う階層的スケジューリング手法について述べた．Web サーバごとに Web ページアクセスを集約することにより，Web サーバに及ぼすアクセス負荷の把握・抑制が可能になる．また，Web ページアクセススケジュールから導出される Web サーバアクセス間隔理論値と，過去の Web サーバアクセス実績に基づく平均ダウンロード経過時間を用いることで，スケジュールの実施に必要な通信リソース量を見積もるとともに，通信リソース割当てにおける Web サーバの優先度を動的に決定することが可能になる．

実クローリングの実績に基づく Web サーバアクセス間隔理論値とダウンロード経過時間を用いたシミュレーションの結果，通信リソースが十分に利用できる場合は，EDF および FREQ 方式による Web サーバ選択が優れていることが確認できた．EDF は理論的に最適であることが知られている手法であるが，実クローリングのパラメータに対しては，Web サーバアクセス間隔理論値のみで Web サーバの優先度を設定する FREQ 方式も EDF と遜色なく，実装を簡略化できることが分かった．ただし，通信リソースが不足すると EDF，FREQ 方式のいずれも取得漏れ数が急激に大きくなるため，必要通信リソース量を見積もるために Web サーバの平均ダウンロード経過時間を把握することが重要である．実際に漏れのない Web ページ再収集を実現するには，アクセス間隔理論値と平均ダウンロード経過時間から見積もった下限値の約 3 倍程度の通信リソースが必要であった．クローラの通信リソース制約によっては，漏れのない Web ページ再収集に必要な通信リソースを確保することができない状況も生じるが，このような場合は，無秩序な取得漏れを防ぐために一定の基準に基づいて Web ページアクセススケジュールの修正を行う必要がある．

本稿で述べた手法は既知 Web ページの再収集を対象としており，未知 URL のアクセスには適用できない．しかし，既知 Web ページの再収集に必要な通信リソース量を最小化するとともにその値をあらかじめ見積もることで，余剰分の通信リソースをすべて未知 URL のアクセスに割り当てることが可能となる．これにより，経験則に頼らない適切なリソース配分が可能となり，間接的に未知 URL アクセスの効率化に寄与することができる．

大規模 Web クローリングの定常的実施においては，大量のリソースが消費されるが，本稿により，通信リソースに関する定量的な見積りの基準が得られた．一方，別の重要なリソースとして，収集データや状態情報を格納するディスク I/O が存在する．今後は大量データの定常的な読み書きにおけるディスク I/O の効率化により，大規模更新クローリングの性能を高める方式についても検討を進めていきたい．

参 考 文 献

1) 喜連川優，豊田正史，田村孝之，鍛治伸裕，今村　誠，高山泰博，藤原聡子：Socio Sense：過去 9 年に及ぶ Web アーカイブから社会の動きを読む，情報処理，Vol.49, No.11, pp.1290–1296 (2008).
2) Kitsuregawa, M., Tamura, T., Toyoda, M. and Kaji, N.: Socio-Sense: A System for Analysing the Societal Behavior from Long Term Web Archive, *APWeb*, pp.1–8 (2008).
3) 田村孝之，喜連川優：大規模 Web アーカイブのための更新クローラの設計と実装，日本データベース学会 Letters，Vol.6, No.1, pp.173–176 (2007).
4) 田村孝之，喜連川優：大規模 Web アーカイブ更新クローラにおけるスケジューリング手法の評価，電子情報通信学会論文誌 D，Vol.J91-D, No.3, pp.551–559 (2008).
5) 田村孝之，喜連川優：大規模 Web アーカイブ更新クローラにおける Web サーバアクセススケジューリング手法，*Proc. DEWS* (2008).
6) Cho, J., Garcia-Molina, H. and Page, L.: Efficient crawling through URL ordering, *Comput. Netw. ISDN Syst.*, Vol.30, No.1-7, pp.161–172 (1998).
7) Najork, M. and Wiener, J.L.: Breadth-first crawling yields high-quality pages, *Proc. WWW '01*, pp.114–118 (2001).
8) Berners-Lee, T., Fielding, R. and Masinter, L.: RFC 3986: Uniform Resource Identifier (URI): Generic Syntax (2005).
9) Koster, M.: Guidelines for Robot Writers (1993). http://www.robotstxt.org/guidelines.html
10) Search Tools Consulting: The Elements of Robots.txt (2008). http://www.searchtools.com/robots/robots-txt-elements.html
11) Liu, C.L. and Layland, J.W.: Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment, *J. ACM*, Vol.20, No.1, pp.46–61 (1973).
12) Jeffay, K., Stanat, D.F. and Martel, C.U.: On Non-Preemptive Scheduling of Periodic and Sporadic Tasks, *Proc. IEEE Real-Time Systems Symposium* (*RTSS*), pp.129–139 (1991).
13) George, L., Muhlethaler, P. and Rivierre, N.: Optimality and Non-Preemptive Real-Time Scheduling Revisited, Rapport de Recherche 2516, INRIA (1995).
14) Lee, H.-T., Leonard, D., Wang, X. and Loguinov, D.: IRLbot: scaling to 6 billion pages and beyond, *Proc. WWW '08*, pp.427–436 (2008).
15) Edwards, J., McCurley, K.S. and Tomlin, J.A.: An adaptive model for optimizing performance of an incremental web crawler, *Proc. WWW '01*, pp.106–113 (2001).
16) Olston, C. and Pandey, S.: Recrawl scheduling based on information longevity, *Proc. WWW '08*, pp.437–446 (2008).
17) Cho, J. and Garcia-Molina, H.: The Evolution of the Web and Implications for an

Incremental Crawler, *Proc. VLDB 2000*, pp.200–209 (2000).
18) Cho, J. and Garcia-Molina, H.: Effective page refresh policies for Web crawlers, *ACM TODS*, Vol.28, No.4, pp.390–426 (2003).
19) Pandey, S. and Olston, C.: User-centric Web crawling, *Proc. WWW '05*, pp.401–411 (2005).

(平成 21 年 3 月 20 日受付)
(平成 21 年 7 月 7 日採録)

(担当編集委員　望月 源)

田村　孝之（正会員）

1991 年東京大学工学部電子工学科卒業．1996 年同大学大学院工学系研究科情報工学専攻博士課程単位取得退学，NEDO 最先端分野技術研究員等を経て，1998 年三菱電機株式会社に入社．博士（工学）．現在，同社情報技術総合研究所専任ならびに東京大学生産技術研究所共同研究員．並列データベース処理，Web クローリング・Web マイニングに関する研究に従事．電子情報通信学会，日本データベース学会，ACM，IEEE Computer Society 各会員．

喜連川　優（正会員）

1978 年東京大学工学部電子工学科卒業．1983 年同大学大学院工学系研究科情報工学専攻博士課程修了．工学博士．同年同大学生産技術研究所講師．現在，同教授．2003 年より同所戦略情報融合国際研究センター長．データベース工学，並列処理，Web マイニングに関する研究に従事．2009 年 ACM SIGMOD Edgar F. Codd Innovations Award 受賞．現在，本会副会長，日本データベース学会理事，電子情報通信学会フェロー．電子情報通信学会データ工学研究専門委員会委員長（1997〜1998），ACM SIGMOD Japan Chapter Chair（1999〜2002）歴任．VLDB Trustee（1997〜2002），IEEE ICDE，PAKDD，WAIM 等ステアリング委員，IEEE ICDE Program Co-chair（1999），General Co-chair（2005）．