大規模 Web アーカイブにおけるコンテンツ解析支援機構

田村 孝之† 喜連川 優‡

三菱電機（株） 東京大学 生産技術研究所‡

東京大学 生産技術研究所†

1. はじめに

筆者らは Web の構造や時間変化を解析することにより，実社会に起こる事象の背景や予兆を探る試みに取り組んでいる．Web 時空間解析の基盤として日本語 Web ページを中心とする大規模アーカイブ（Web アーカイブ基盤）の構築を進めており，約 9 年分の情報を蓄積するに至った．Web アーカイブ基盤では，Web 構造を網羅的に把握すると共に詳細な時間変化を追跡するという要求を満たすため，URL 毎に異なる時間分解能で複数バージョンを蓄積しており，最小時間分解能は現在 1 日となっている．

Web 時空間解析においては，個別 Web 文書の取得よりも，時間や Web サイト名を条件とする文書集合の一括取得が重要となる．また，文書集合に含まれる大量データを高速に解析するために並列処理が不可欠である．本稿では，以下，Web アーカイブ基盤が備える Web 時空間解析アプリケーション支援機構として，スナップショット生成機構及び並列スキャン機構について述べる．

2. スナップショット生成機構

Web アーカイブ基盤は URL 毎に異なる時間分解能を持つため，ある時点での Web スナップショットを解析する際は，URL 毎に適切なバージョンを取り出して動的にスナップショットを生成する必要がある．そこで，ある URL の隣接するバージョンが時刻 D^-および D^+について存在する場合，時刻 D^-におけるバージョンが半開区間 $[D^-, D^+)$に亘って有効なものとし，この文書有効期間 D と解析アプリケーションが指定するスナップショット選択期間 $Q=[Q^-, Q^+)$との時区間論理演算[1]によりバージョンの選択を行う．例えば，期間 Q にアクセスされ，Q の終点直前まで有効なバージョンを選択するための条件は，(D overlapped-by $Q \vee D$ started-by $Q \vee D$ finishes $Q \vee D$ $= Q$)となる．

Web アーカイブ基盤において文書データは個別に圧縮され，任意の順序で一定サイズまで連結してフラットファイルに格納される．各文書へのランダムアクセスは，（URL，有効期間始点）をキーとする B-tree 索引により実現している．有効期間の終点は明示的に格納せず，同一 URL に対応する後続エントリの有効期間始点を用いる（後続エントリがない場合は∞）．これにより，Web アーカイブへの文書データの追加は，直前のエントリに影響することなく，新たなエントリの単純な追加として処理できる．また，索引エントリは，上記キー及び文書データへのポインタに加えてフラグを含んでおり，「エラー応答」，「正常応答：更新非検出」，および「正常応答：更新検出」を識別できるようになっている．「正常応答：更新非検出」のエントリに対応する文書データは HTTP の Not Modified 応答メッセージとなっており，文書データの実体を取得するには索引を遡って「正常応答：更新検出」エントリを探索する必要がある．

また，スナップショットのように大量の文書データを取得する際は，各索引エントリのポインタを直ちに辿ると，ランダム I/O によりディスクアクセス効率が低下してしまう．そこで，このような場合はポインタのリストをファイルに書き出し，論理アドレス順にソートしてから文書データの実体をアクセスするようにしている．

前記条件を用いたスナップショット生成処理は，以下の通りとなる．

1. 初期位置を索引末尾に設定．
2. 索引を逆順に走査し，$Q^- \le D^- < Q^+$を満たすエントリを検索．索引先頭に到達した場合はポインタリストファイルをソートして終了．
3. フラグが「正常応答：更新非検出」であれば，「正常応答：更新検出」エントリに到達するまでさらに走査を続ける．
4. ポインタの値をポインタリストファイルに出力．
5. 同一 URL のエントリをスキップして 2 に戻る．

Web アーカイブ基盤は，URL 文字列に基づく索引も備えており，特定サイト内あるいはと特定ディレクトリ下のコンテンツからなるスナップショットを生成することも可能である．

Mechanisms for mining application support in a large-scale Web archive.

†Takayuki Tamura, Mitsubishi Electric Corporation / IIS, The University of Tokyo

‡Masaru Kitsuregawa, IIS, The University of Tokyo

3. 並列スキャン機構

スナップショットは億単位の文書を含むため，その解析には並列処理が不可欠である．Web アーカイブ基盤は，文書データの取り出しとその解析処理を連携させる仕組みとして，図 1に示す並列スキャン機構を備えている．並列スキャン機構はマスタ・スレーブ型の並列処理方式に基づいており，単一のディスパッチャノードと，複数の処理ノードで構成される．

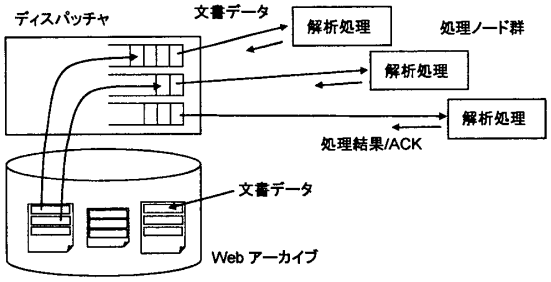

図 1 Web アーカイブ基盤における並列スキャン機構

ディスパッチャは Web アーカイブから解析対象期間に対応するスナップショットを抽出すると共に，文書メタデータ（URL 文字列や Content-type 属性等）に基づく選択演算を実施する．文書データの本体は Web アーカイブから圧縮した状態で取り出され，そのままいずれかの処理ノードに送付される．処理ノードはディスパッチャから文書データを受け取ってデータ伸張と解析を行い，結果をディスパッチャに返すという処理を繰り返す．

ディスパッチャは各処理ノードに対応するキューを維持しており，処理ノードへの送信開始から処理結果受信完了まで，文書データをキューに保持する．新たな文書データを処理ノードに割り当てる際は，未送信データが残っていない処理ノードを選択することで，処理ノード間の動的負荷分散が実現される．また，パイプライン的にデータを供給するため，文書データの送信完了後，処理結果の受信を待たずに次の文書データの送信を可能としている．

一部の処理ノードで障害が発生した場合は，キューに残っている文書データを別ノードに送付することで処理を継続する．ただし，Web 上のコンテンツは HTML 等の規格に従っていないものも多く，データの問題で解析プロセスが異常終了することもある．そのため，引き続いて異常を引き起こしたデータは処理をスキップするようにしている．また，解析プロセスの実行時間が長くなることで，メモリリーク等の問題が顕在化して異常終了を引き起こすことがある．このような問題に対処するため，一定数の文書データを処理するごとに解析プロセスを再起動する仕組みも用意している．

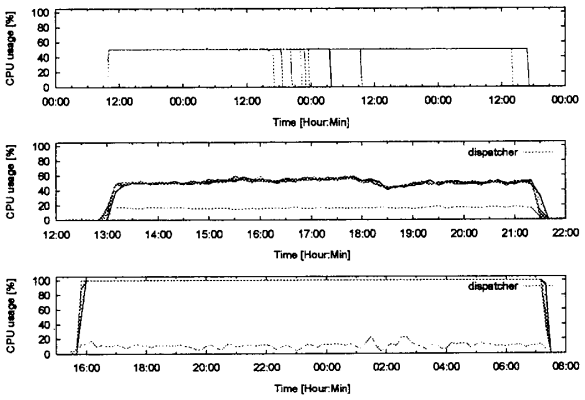

図 2 並列コンテンツ解析時の CPU 使用率トレース（上：処理ノード静的割当，中：動的割当／同期的データ供給，下：動的割当／パイプラインデータ供給）

図 2に解析アプリケーションの実行中に採取した各ノードの CPU 利用率トレースを示す．比較のため，上段に各処理ノードに一定サイズのデータを静的に割り当てた場合を，中段に前の文書データの処理結果を受信してから次の文書データの送信を開始する同期的データ供給を行った場合を，それぞれ示した．これらと比べて下段の動的処理ノード割り当てとパイプラインデータ供給を行った場合では，各処理ノードの実行完了がほぼ同時で，CPU 使用率も 100%を維持しており，非常に効率よく解析処理を実行できていることが分かる．

4. おわりに

筆者らが構築した大規模 Web アーカイブにおいて，Web 時空間解析を支援するためのスナップショット生成機構および並列スキャン機構について述べた．今後はアベイラビリティの向上とさらなる大規模化を目指して Web アーカイブ基盤の分散ストレージ化を検討していきたい[2].

謝辞

本研究の一部は文部科学省リーディングプロジェクト e-Society 基盤ソフトウェアの総合開発「先進的なストレージ技術および Web 解析技術」による．

参考文献

[1] Allen, J.: “Maintaining Knowledge about Temporal Intervals,” *Communications of the ACM*, Vol.26, No.11, Nov. 1983, pp.832-843.
[2] Ghemawat, S., et al.: “The Google File System,” *Proc. of ACM SOSP '03*, Oct. 2003.