2R－5

Virtual Web Space Accessのキャッシュ機構の実装と性能評価*

斎藤 淳, 中津 利秋, 堀切 和典, 川邉 恵久†
富士ゼロックス（株）IT事業開発部‡

1 はじめに

筆者らは、CGIより柔軟で分散的に協調動作する新しいWWWサーバとしてVirtual Web Space Access（以下VSA）を考案し、実装をおこなった[1][2]。このアーキテクチャに中間処理結果を管理するキャッシュ機構を組み込み、結果を評価したので報告する。

2 VSA概要

VSAでは、データを処理する複数のコンポーネントプログラムを起動するプロシジャをURLに埋め込む。このURLをヴァーチャルURLと呼ぶ(以下VURL)。VURLは、コンポーネントの名前fと、原料となるデータのURLから、模式的には$f(URL_1,...URL_n)$のように構成される。各$URL_1,...URL_n$は、再帰的にVURLであってもよい。

VSAサーバには、VURLを解釈するコンテキストがあり、プロシジャに従って各コンポーネントを起動し、さらに他のVSAサーバにHTTPストリームで接続して、コンポーネントを分散的に起動し、処理のパイプラインを構成する。その結果、処理結果はバーチャルページとして、ブラウザに出力される。

図1は次式を計算するパイプラインとなっている。

```
(a1+a2)*(A1+A2)+(b1+b2)*(A1+A2)
+(a1+a2)*(B1+B2)+(b1+b2)*(B1+B2)
```

図1のような計算をする場合、VSAサーバ2および4からのリクエスト(a1+a2)は同じ計算であるので、再計算を省く最適化が必要になるが、複数のサーバにまたがった最適化は簡単ではない。

また、インタネットのような悪条件のネットワークでは、処理中にいずれかのコンポーネント間の接続が絶たれる確率が増えるため、繰り返しリクエストしても結果を得られないことがおこる。

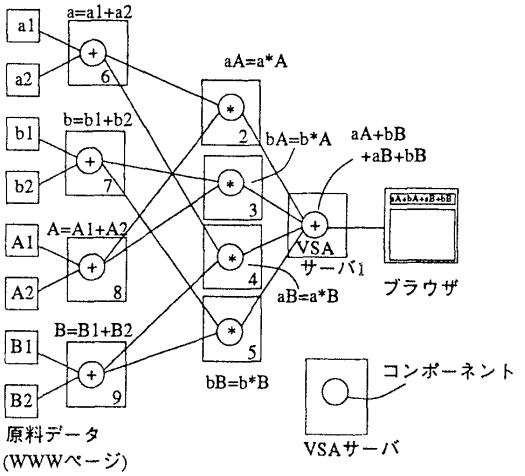

図1: 処理パイプラインの例

3 キャッシュ機構の設計

筆者らは、上述の2つの問題を解決するため、中間結果を格納し再利用するための方法を検討した。

中間結果を明示的に特定のサーバに格納する方式では、セキュリティや効率の問題があり、また中間結果に一意なID(変数名）を割り当て、VSAサーバがIDを共有する機構の実現も簡単ではない。

関連した技術として、HTTPの分散キャッシュ技術[3]がある。このようなキャッシュサーバを利用すれば、格納場所とキャッシュのIDの問題は解決するが、中間結果のキャッシュを保持するにはすべてのコンポーネント間接続をキャッシュサーバ経由にする必要があり、トラフィックの集中が起きる。

そこで、処理パイプラインと同じ配置の分散キャッシュに中間結果を格納し、かつ個々のVSAサーバ上でベストエフォートで最適化を行なう方式を検討した。

VSAの特徴として、処理の組み合わせがVURLに表されているので、VURLと過去の計算履歴を比較して計算が省けるか否かの判定ができる。そこで、各VSAサーバが中間結果とVURLをそれぞれのキャッシュに格納しておき、その後のリクエストのVURLと

*Implementation and Evaluation of Cache Mechanism for Virtual Web Space Access

†Jun SAITO, Toshiaki NAKATSU, Kazunori HORIKIRI, Shigehisa KAWABE
‡IT Business Development. Fuji Xerox,Co.Ltd.

比較して一致したときにはキャッシュされた中間結果を返す方式を考案した。このキャッシュ機構は利用者から隠蔽され、従来の VURL を変更せずに性能向上が期待できる。

図 1 に示した計算の例では、同じ VURL を用いても、VSA サーバ 6〜9 での計算が 1/2 になる。

4 キャッシュ機構の実装

考案したキャッシュ機構の実装を図 2 に示す。中間結果の保存は、図 2 に実線で示したように処理のパイプラインの出力側と入力側の両方で行われる。中間結果の取得時は、図 2 に点線で示したように動作する。

格納する中間結果に対して、サーバ内で一意な ID(cid) を生成し、同じ計算から生成された中間結果を時系列的に区別して扱うことも可能にした。

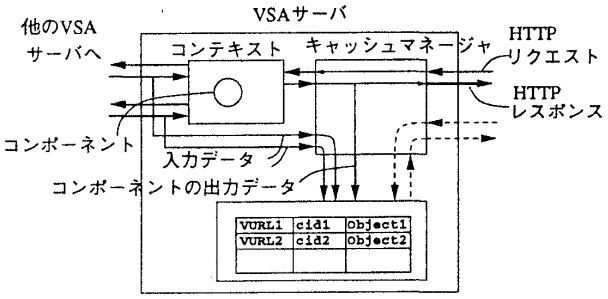

図 2: キャッシュ機構の動作

キャッシュを利用するかどうかは、キャッシュの有効性によって決まる。キャッシュの有効性を調べるには、中間結果を生成した処理の依存関係を調べ、依存するデータのキャッシュの有効性を調べる動作を再帰的に行って決める。

また、キャッシュの利用方法をブラウザ側でコントロールするため、ブラウザの操作方法によって変化する HTTP リクエストヘッダを調べて、保存されたキャッシュの置き換え、有効性をチェックしてのキャッシュ利用、などの動作を切替えるように実装した。

5 評価

複数の VSA サーバ上の処理を組み合わせた利用を想定し、キャッシュ機構の効果を調べた。

複数の入力データにフィルタ処理を行って合成し、1 つのデータを出力する処理である f, g, h をサーバをまたがって組み合わせ、

```
f(g(h(A),h(A),h(A)),g(h(A),h(A),h(A)),
g(h(A),h(A),h(A)))
```

とあらわされる処理を構成した。

キャッシュがサーバに存在する確率をヒット率と定め, これを変化させて、ブラウザが上記の処理のリクエストを送出してからレスポンスの受信を完了するまでを処理時間として測定した。また、悪条件のネットワークを想定し、VSA サーバ間に遅延やパケット破棄を設定できる回線シミュレータをおいた。

その結果、サーバ間の遅延 323ms, パケット破棄率 15%という条件で図 3 に示す結果を得た。

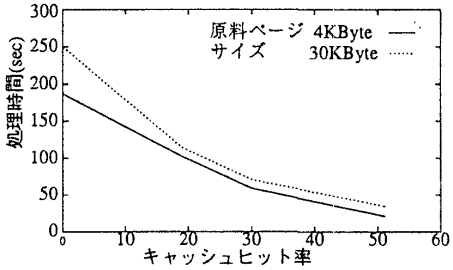

図 3: ヒット率と処理時間の関係

また、全体の処理時間のタイムアウトを 10 分と決め、パケット破棄率 65%という条件でタイムアウトの発生率を評価した結果を表 1 に示す。

キャッシュヒット率	タイムアウト発生率
50%	13
30%	49
0%	100

表 1: タイムアウト発生率 (%)

6 まとめ

VSA に部分計算の結果を保存するキャッシュ機構を実装し、評価を行なった。

評価の結果、30 %のヒット率で、2 倍以上処理時間が向上する効果があった。また、キャッシュ機構により処理の完了前にタイムアウトする確率が減少した。

今後は、キャッシュの更新、移動などの方法に改良を加える予定である。

参考文献

[1] 川邊 他. “WWW の分散コンピューティング : Virtual Web Space Access”, 第 58 回情処全大, 1P-05, 1999.

[2] 中津 他. “Virtual Web Space Access を実現した WWW プロセスサーバのデザインと実装”, 第 58 回情処全大, 2R-03, 1999.

[3] “Squid Internet Object Cache”, http://squid.nlanr.net/Squid/