

サーバ/クライアントシステムにおける
クライアントキャッシング技法

1P-4

奥澤文一
(株)東芝 府中工場

〔1〕はじめに

計算機応用システムの世界にもダウンサイジングの波が訪れ、サーバ/クライアント型システムは情報システムモデルの主流となりつつある。そこでは一般的にシステムで使用する情報をサーバで一元的に管理し、修正の必要が生じた場合には、クライアントにLAN経由で情報を取り出し、修正の後サーバに再登録を行っている。従って、サーバの一元管理情報へアクセスするユーザアプリケーションはその通信時間が膨大なものとなり、業務の効率化に支障をきたす結果となっている。

本報告は、サーバ/クライアントシステムにおいて一元管理情報に矛盾を生じさせることなく、その通信データを減少させることを目的としたクライアントキャッシング技法に関する一考察である。

〔2〕クライアントキャッシング技法

本技法では、図1に示すようにサーバ/クライアントシステムにおいて、各クライアントはサーバ上で使用頻度の高い情報を仮想的に(ユーザに意識させることなく)保持し、この情報を利用して所用の処理を実行する。

そのため、キャッシュエリアに所望する情報が存在する場合には、あたかもサーバ情報を操作するが如くそのまま情報を操作できるため、処理速度の向上が期待できる。

一般にキャッシング機能を具備するシ

ステムにおけるそのアクセス時間 t は

$$t = p t_c + (1 - p) t_s \dots (1)$$

p : データがキャッシュ中にある確率
(キャッシュヒット率)

t_c : キャッシュのアクセス時間

t_s : サーバへのアクセス時間

で表される。当然、通常 $t_c \ll t_s$ である為、ヒット率 p をできるだけ1に近くする工夫が必要となる。また、ヒット率 p はデータの使用頻度や最終利用履歴等の関数になっていると考えられデータの種別に大きく関係している。通常クライアント上で許されるキャッシュエリアは小さく限られることから上記ヒット率 p の1に近いものを中心に残す工夫が大切となる。

キャッシュエリアがあふれてしまう場合の更新方法として例えば、アクセス時間の古いデータから順に削除を行う方法や使用頻度を計数し頻度の高いものを優先的に残す方法やこれらの組み合わせによって常にアップデートする方法等が考えられる。

上記の考慮以外に例えば、図1に於いてデータ1をユーザ作業エリアに取り出して修正するという業務を考えた場合、サーバからデータ1を作業エリアに取り出して修正を行う構成に比べ、クライアントにキャッシング機能を備えた構成では、キャッシュエリアにデータ1の存在を調査し、存在する場合にはサーバのデータ1との整合性をとる様なデータ管理機能が必要となる。

(他のクライアントからサーバのデー

データ1が更新されている可能性があるため)ここで整合性がとれている場合にはキャッシュエリアから作業エリアにデータ1をコピーし、とれていない場合にはキャッシュエリアのデータ1をサーバのデータ1を用いて更新した後作業エリアにコピーする。

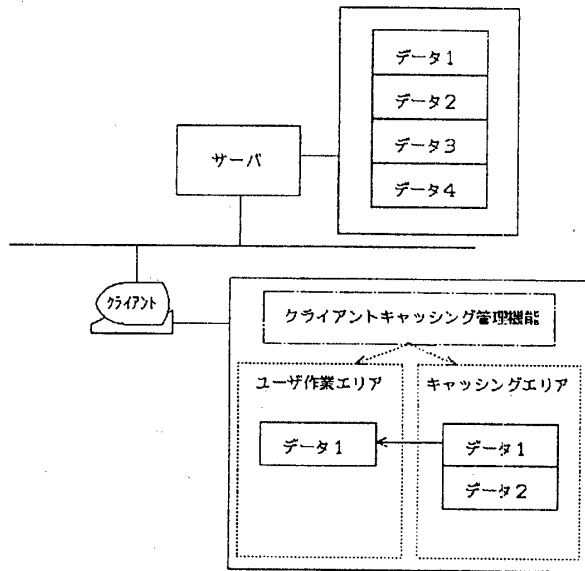
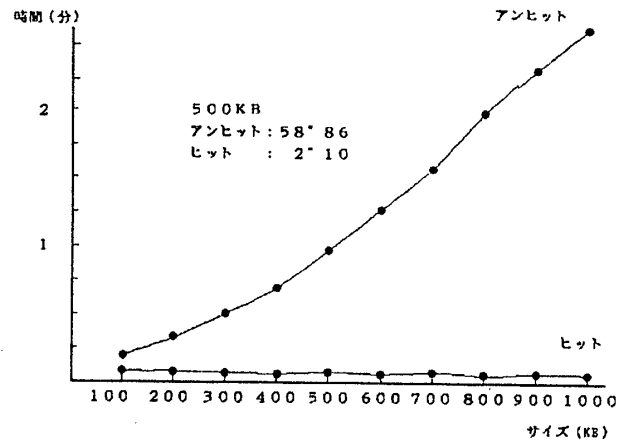


図1 クライアントキャッシング技法

〔3〕性能比較事例

グラフ1に上述のような考慮をした実システム例での処理の性能比較事例を示す。

グラフには様々なサイズのデータについて、キャッシュエリアに当該データが存在しないとき取り出す場合(ヒット率0でサーバから取り出す場合)、及び存在するとき取り出す場合(ヒット率1の場合)について示している。ここでヒット率0は従来の構成(クライアントキャッシング技法を用いない)を想定した近似環境と考えて良い。本システム例ではキャッシュエリアはハードディスク上に20MByte設けた。



グラフ1 性能比較事例

〔4〕考察

クライアントキャッシング技法は、システムのレスポンス向上に大きく貢献し、結果的に業務の効率化に劇的に寄与することは明白である。また、ここで特筆すべきは従来の構成(ヒット率0に近似)の取り出し時間は、大きな割合を占めるデータ転送時間の関係でデータのサイズに比例するのに対し、クライアントキャッシング技法はサイズに依存しないという点である。

本事例では(1)式に於ける t_c は2秒前後、 t_s は500KBのデータで1分弱となっている。

また、クライアントキャッシング技法では、通信回線の性能が性能の劣化につながるため、サーバ・クライアント間の回線性能が低い場合はその効果は極めて顕著であるといえる。

従って前述のキャッシュエリアあふれに対する更新方法には以上の考察も考慮して実施することが大切になる。

業務の種類によっては同じ業務を繰り返し実行するという運用を行うため、同じデータを繰り返し使用することとなり、ヒット率はほぼ1となるケースがある。この場合はクライアントキャッシング技法が極めて有効となる。