

排他的ディスクキャッシュを利用したストリーミングサーバの負荷分散方式*

中西 亮 品川 高廣 吉澤 康文

東京農工大学工学部情報コミュニケーション工学科

1. はじめに

近年のブロードバンドの普及に伴い、ストリームデータの配信サービスが注目されてきている。ストリーミングは、ディスクやネットワーク等の I/O 処理を大量に必要とするため、I/O 処理がボトルネックとなって、1 台のマシンでは多数のクライアントにサービスを提供することが難しい。そこで、複数のサーバを用いた負荷分散 (Server Load Balancing : SLB) という手法が用いられている。しかし現在の負荷分散でよく使われている最小接続数などのアルゴリズムでは、ストリーミングの特徴を考慮しているわけではないため、複数のサーバで同じコンテンツをキャッシュするなど、システム全体でキャッシュが効率よく用いられていない。

そこで本報告では、(1)各サーバが重複するデータを配信しないことで、キャッシュを有効利用する排他的ディスクキャッシュと、(2)サーバへのクライアント数の偏りを解消するクライアント数の調整の 2 つの手法により、サービス可能なクライアント数の向上を実現する。

2. システムの構成

本システムでは、図 1 のようにクライアントからの要求はいったん負荷分散を行なうマシン (SLB) で受け付け、各サーバへと振り分ける。また、SLB はサーバからのパケットを中継し、必要であればヘッダの書き換えなども行なう。

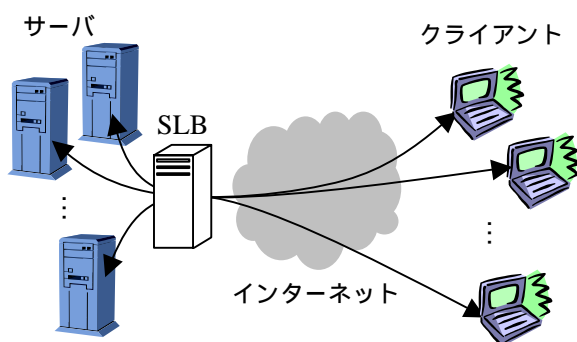


図 1 SLB の構成

従来の WEB サーバなどの SLB では、クライアントから要求をいったん振り分けた後は、1 台のサーバが要求を最後まで処理し、途中でサーバが切り換わることはない。しかし、ストリーム配信は WEB サーバと異なり、サービス中にサーバを切り換えることができるという特徴がある。これは、ストリームの配信が UDP で行なわれているためである。そこで、SLB はクライアントがどのサーバでサービスを受けるべきかを常に評価し、必要ならば適切なサーバに切り換える処理を行なう。

3. 排他的ディスクキャッシュ

ストリーミング配信では、データのサイズが大きいため、Web サーバなどと比べてディスク読み出しなどの I/O 処理がネックとなりがちである。従ってディスク I/O を削減するために、キャッシュを有効利用することが重要となってくる。しかし、従来の最小接続数による負荷分散では、要求されたコンテンツの内容にかかわらず接続数が最も少ないサーバに振り分けるため、同じコンテンツを複数のサーバがキャッシュしてしまう可能性がある。

そこで本研究では、各サーバが同じコンテンツを重複してキャッシュしないようにするために、同一タイトルは同一サーバで配信させるように SLB で振り分ける。各サーバが同じコンテンツを重複してキャッシュしない、すなわちあるコンテンツは 1 台のサーバが排他的にキャッシュすることにより、サーバ全体でのキャッシュの利用効率を向上させる。

しかしストリームデータは一般に 1 タイトルのサイズが大きいため、1 台のマシンではタイトル全体をキャッシュしきれない場合がある。その場合、どんなに参照頻度が高くてもキャッシュが置き換えられてしまうため、ディスク I/O がネックになってしまう。そこで、参照頻度が非常に高いタイトルは、1 つのタイトルをブロックに分けて複数のサーバで分割してキャッシュさせ、クライアントへの配信は、キャッシュを持っているサーバへと動的に切り換える。これにより、参照頻度の高いタイトル全体をメモリにキャッシュすることが可能になり、I/O 処理がネックとなることを避けることができる。こ

*Load Balancing Mechanism with Exclusive Disk Cache for Streaming Servers
Ryo NAKANISHI, Takahiro SHINAGAWA and Yasufumi YOSHIZAWA
Department of Computer, Information and Communication Sciences,
Faculty of Engineering, Tokyo University of Agriculture and Technology

の手法は、新作映画の配信など、少数のタイトルにアクセスが集中するケースにおいて有効であると考えられる。複数サーバで1タイトルを配信する様子の例を図2に示す。

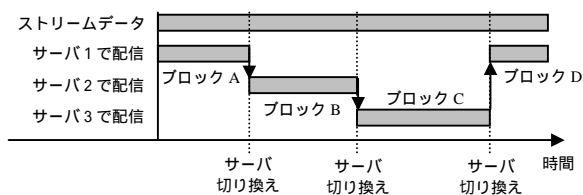


図2 排他的な配信の様子

4. クライアント数の調整

排他的ディスクキャッシュを行なうと、クライアントが接続するサーバが極端に偏る場合があります。CPU性能がネックとなってしまう可能性があります。

そこで、接続数が極端に偏った場合は、最もアクセスが多いブロックを接続数が少ないサーバに振り分ける処理を行なう。これによって、サーバ毎の処理するクライアント数を平滑化し、負荷を適切に分散する。

5. 性能実験

提案システムの性能を測定する実験を行なった。実験内容は、配信要求するクライアントの数を増加させながら、サーバ全体として配信できるストリーム配信レートの合計値がどのように変動するかを測定した。クライアントが要求するタイトルの種類は、図3に示す割合になるようにランダムに選択する。図3の割合は2003年12月5-7日の全米映画興行収入ランキングのトップ10のものである[1]。ただし、実際に配信するストリームデータは、全て表2に示すMPEG4のデータである。最小接続数アルゴリズムと本サーバ選択アルゴリズムのそれぞれで配信レートを測定し、結果を比較した。

実験マシンとしては、サーバには表1の各マシン、SLB及びクライアントにはそれぞれ1台のPentium4 2.4GHz、メモリサイズ512MBのマシンを使用した。各マシンは100BASEのNICで接続している。ストリームサーバにはDarwinストリームサーバ、OSはLinux 2.4.19を使用した。

表1 サーバのスペック

	CPU	メモリサイズ	HDD
サーバ1	PentiumMMX 200MHz	96MB	WDC AC26400R
サーバ2	PentiumMMX 233MHz	96MB	IBM-DTTA-350430
サーバ3	PentiumPro 200MHz	80MB	QUANTUM FIREBALL EX6.4A
サーバ4	PentiumMMX 200MHz	128MB	QUANTUM FIREBALL TM2110A

表2 MPEG データ

ビットレート	長さ	サイズ
1Mbps	20分	152MB

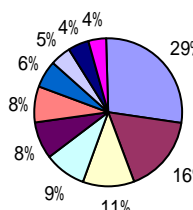


図3 各タイトルへの要求の割合

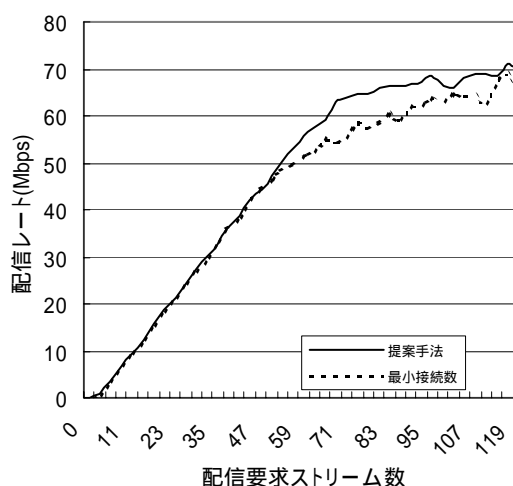


図4 配信レートの変動

図4に測定結果を示す。測定結果より、最小接続数により負荷分散した場合は、配信要求ストリーム数が50ストリームになると配信レートが低下し、配信品質が保たれなくなることがわかる。提案手法では68ストリームになるまで配信品質が保たれている。よって提案手法を用いることで、最小接続数で負荷分散するよりも、配信システム全体の性能が136%向上した。

6. まとめ

本報告では、排他的ディスクキャッシュを利用したストリーミングサーバの負荷分散について述べた。本方式では排他的にサーバにストリームデータを配信させることの有効性について述べた。本方式の負荷分散により、ストリーミングサーバの性能が136%向上することを確認した。今後はMPEGのデータやサーバのスペックなどを、より現実に近い環境で実験をおこない、本方式の有効性を検証していきたい。

参考文献

- [1]ロイター・ジャパン 映画の全米興行収入トップ10 (2003年12月5日-7日)