

DNS を利用した負荷分散機構で用いる重みを決めるアルゴリズムの評価

15T266 萬木 大志 (最所研究室)

DNS を用いた分散 Web システムにおいて、可変式 TTL を用いた重み付けラウンドロビン機能で用いる重みを決める稼働率に応じた基本増減値の評価と負荷分散の検証について述べる。

1 はじめに

当研究室では、負荷量に応じてクラウド上のキャッシュサーバの台数を動的に増減することで、応答性を維持しつつ運用コストを削減する分散 Web システムを開発している。先行研究では、DNS ラウンドロビンを使用したリクエスト振り分け機構が開発された [1]。そこでは単純な DNS ラウンドロビンが使用されており、サーバ間で負荷の大きな偏りが発生した。そのため、負荷の偏りを防ぐことを目的とした可変式 TTL を用いた重み付けラウンドロビン機能が開発された [2]。この機能では、リクエスト振り分け先に選ばれる確率とリクエストを振り分ける比率を決定するためのパラメタとして重みを用いる。先行研究では、重みや稼働率に応じた増減値に暫定的な値を使用していた。本研究では、重みの決定で最も重要な、稼働率に応じた重みの増減値が負荷分散に与える影響について評価した。また、推定通りに権威 DNS サーバのリクエスト振り分けによる負荷分散が行われているかについての検証も行った。本稿では、重みの基本増減値の評価とサーバの負荷分散の検証について述べる。

2 DNS を用いた分散 Web システム

図 1 に DNS を用いた分散 Web システムを示す。本システムは、管理サーバと権威 DNS サーバ、オリジンサーバ、キャッシュサーバで構成されている。管理サーバは、サーバ群の負荷監視とキャッシュサーバの起動・停止、DNS ゾーン情報の更新を行う機能を持つ。負荷量として稼働中の全てのサーバの稼働率(現在のリクエスト数/最大同時処理数)の平均を使用する。平均稼働率が閾値 Th_{high} を上回った場合にはキャッシュサーバを起動。閾値 Th_{low} を下回った場合には停止させる。起動時には、起動完了後にゾーン情報を更新し、停止時には、ゾーン情報を変更し、DNS キャッシュの期限が切れた後にサーバを停止する。DNS ラウンドロビンを用いたリクエスト振り分けは、稼働中の全てのサーバに順番に振り分ける。

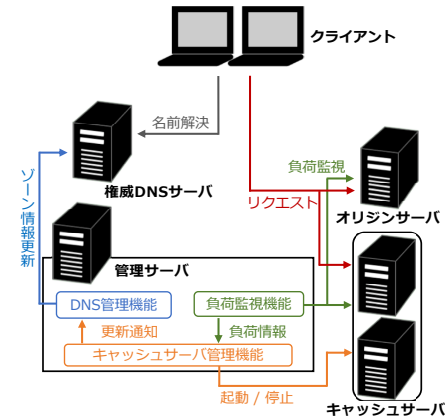


図 1: DNS を用いた分散 Web システム

3 可変式 TTL を用いた重み付けラウンドロビン

サーバ間での負荷の偏りに対処するために、高負荷サーバへのリクエスト振り分けを一時的に停止する機能が実装された。この機能は、稼働中の全てのサーバにリクエスト振り分けを行わずに、リクエストを振り分けるサーバの数を制限した。また、サーバの負荷に応じて重みを動的に決定する重み付けラウンドロビン方式を導入した。リクエスト振り分け先サーバは、全体の重みに対する各サーバの重みの割合で選択される。これにより、低負荷サーバがリクエスト振り分け先に選ばれる確率を高くした。またこの機能は、短い TTL を設定するほど負荷の偏りが緩和されることが確認されている。そこで先行研究では、低負荷サーバには通常の TTL を設定し、高負荷サーバには負荷状態に応じて短くした TTL を設定する機能を実装した。これにより、高負荷サーバへのリクエスト振り分けを低負荷サーバへ迅速に切り替えることが出来るようになった。

この機能によって、従来の DNS ラウンドロビンを用いた場合に比べ、平均的に TTL の値が大きくても応答性やコストパフォーマンスが改善することが確認されている。しかし、重みのデフォルト値や稼働率に応じて増減する値などは暫定的な値であった。そのため、各パラメタの最適値の評価が必要であった。

4 重みの基本増減値の評価

前節の機能で用いる重みは、リクエストに振り分ける比率と名前解決の IP アドレスに選ばれる確率に関わる。本研究では、重みを決定するのに最も重要な稼働率に応じた重みの基本増減値だけを一律に変化させて評価実験を行った。表 1 に重みの基本増減値を示す。実験環境として、5 台のホストマシン上に仮想マシンとしてクライアント 10 台、DNS ルートサーバ 1 台、権威 DNS サーバ 1 台、DNS キャッシュサーバ 10 台、管理サーバ 1 台、オリジンサーバとキャッシュサーバ 10 台を構築した。クライアントからのリクエストを段階的に増加させ、オリジンサーバとキャッシュサーバへ負荷をかけていく実験を行った。表 2 に実験結果を示す。平均応答時間について見る。基本増減値を 2.0 倍した場合が 0.73 秒と最も短く、元の基本増減値の場合が 2.1 秒と最も長いことが分かる。また、1 コストあたりの処理数について見る。基本増減値を 2.0 倍した場合が最も多い 118.06 であり、元の基本増減値の場合が 73.71 と最も少ない処理数ということが分かる。以上のことから、サービスの応答性とコストパフォーマンスの観点で、基本増減値を 2.0 倍したパラメタを用いることで、より最適な負荷分散を行うことができると言える。

基本増減値を 2.0 倍した場合が最も良い結果が得られた要因について考察する。重みのデフォルト値は 100 である。増減値を 2.0 倍した場合、サーバが極めて高負荷状態だと判断された場合は重みが 0 に設定されるため、リクエスト振り分けが行われなくなる。そのため、効率よく負荷を軽減することができたと考えられる。また、低負荷状態に判断された場合の重みが 2.0 倍しているため、低負荷サーバにはリクエストが多く振り分けられることも分散がうまくできた要因だと考えられる。以上のことから、重みの基本増減値を 2.0 倍した場合、高負荷サーバを効率よく負荷を軽減させ、低負荷サーバにリクエストが多く振り分けられることで負荷分散がより改善させられたと考えられる。

5 サーバの負荷分散の検証

権威 DNS サーバに振り分けるリクエストの割合通りに各サーバがリクエストを処理しているかについて検証を行った。権威 DNS サーバに、名前解決の問い合わせをした DNS キャッシュサーバと問い合わせ時間、回答 IP アドレスを記録する機能を追加して前節と同じ実験を再度行った。稼働率に応じた重みの基本増減値は 2.0 倍に設定し、他のパラメタは前節と同じ値である。この機能により時間毎に各 DNS キャッシュサーバが保持する回答 IP アドレスが分かる。このデータから時間毎のオリジンサーバと稼働中のキャッシュサーバへ振り分けられるリクエストの割合を推定する。図 2 にオリジンサーバの処理割合を示す。時間

表 1: 稼働率に応じた重みと TTL の基本増減値

稼働率	重み	TTL
$0.0 \leq x \leq 0.1$ (低負荷)	+30	0
$0.1 < x \leq 0.4$ (通常負荷)	0	0
$0.4 < x \leq 0.6$ (中負荷)	-30	-18
$0.6 < x \leq 0.8$ (高負荷)	-40	-24
$0.8 < x \leq 1.0$ (極めて高負荷)	-50	-30

表 2: 各パターンの基本増減値による実験結果

増減値の倍率	コスト	合計 応答時間	処理数	平均 応答時間	1 コスト あたりの処理数
1.0	3125	241195	229704	2.1	73.71
1.5	3849	211865	291342	1.38	77.37
2.0	3293	188619	386291	0.73	118.06
2.1	2993	213296	318122	0.88	106.36
2.2	3598	203836	362128	0.96	100.93
2.5	3635	201375	361906	0.96	99.92

毎の推定リクエスト割合と実際の処理数割合の値が大きくずれていないことが分かる。また、オリジンサーバ以外にもキャッシュサーバ 1~4 も稼働していたがどのサーバもオリジンサーバと同様の結果になった。

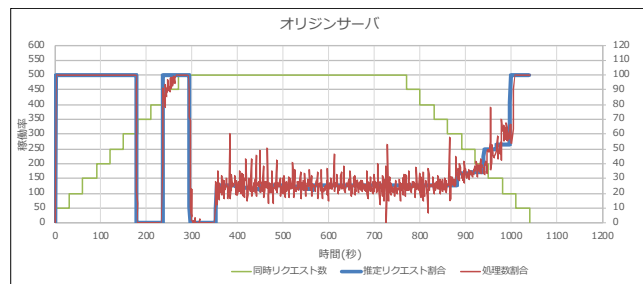


図 2: オリジンサーバの処理割合

6 まとめ

先行研究で用いた稼働率に応じた重みの基本増減値を一律 2.0 倍することでより最適な負荷分散を行うことができたことを確認した。また、権威 DNS サーバのリクエスト振り分けの割合通りに各サーバでリクエストが処理されていることも確認した。

今後の課題として、稼働率に応じた TTL の基本増減値や増減する基準などの改良、負荷状態を測定するパラメタの検証が考えられる。

参考文献

- [1] 森垣航太, 最所圭三, “分散 Web システムにおける DNS を用いた負荷分散機構の開発”, 平成 28 年度 電気関係学会四国支部連合大会論文集, 2016
- [2] 森垣航太, “DNS を用いた分散 Web システムにおける負荷分散機構の改良”, 香川大学, 修士論文, 2017.