

コンテンツ分割キャッシュを用いた配信ネットワークの効率化

研究代表者

吉永 努

電気通信大学 大学院情報システム学研究所 教授

1 はじめに

本研究では、急激な人気変動を伴うVOD (Video-On-Demand)通信ネットワークにキャッシュを導入し、通信量を削減する手法を提案する。提案するキャッシュは、アクセス頻度に応じてキャッシュするコンテンツを選択するLFU(Least Frequently Used)と、急激な人気変動に強いLRU(Least Recently Used)を階層的に組み合わせる構成とする。ネットワーク・シミュレーションの結果、提案方式が人気変動に追従したキャッシュの高ヒット率を示し、通信ホップ数を削減することを示す。

2 異種キャッシュ混在ネットワーク

2-1 動画の人気変動に追従するキャッシュ

VOD サービスを対象とした調査[1, 2]の結果、動画アクセスの偏りは従来知られていたZipf則の分布よりも偏りが大きく、ガンマ分布に近いことが明らかになっている。図1に示すように、人気上位10%の動画をキャッシュすると、Zipf則のアクセス分布で35.2%、ガンマ分布のアクセスでは71.6%のアクセスがキャッシュヒットする。一方、新規に人気動画が追加されたり、ニュースやSNSで話題になった古い動画にアクセスが集中したりすると、動画が事前にキャッシュされていないため、ヒット率が大幅に下落し、通信量も増加してしまう。

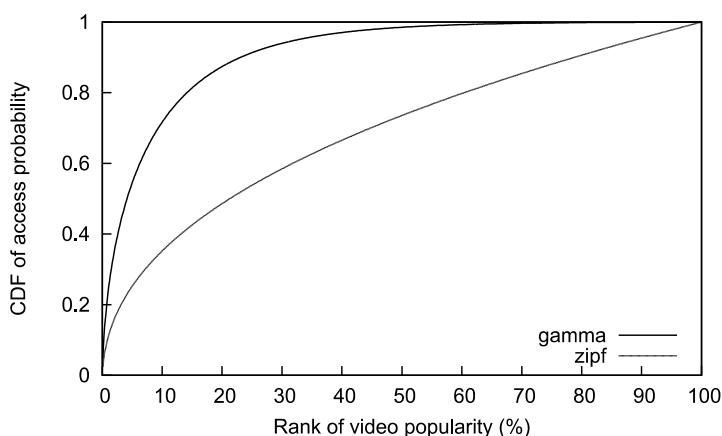


図1. Zipf則とガンマ分布による人気動画の偏り

本研究では、アクセスネットワーク、メ

トロネットワーク、コアネットワークのように多段構成のネットワークを対象として、各階層にキャッシュを配置することを想定し、VOD通信に適した階層キャッシュの仕組みを考察する。キャッシュを多段に並べることで、利用可能なキャッシュ領域を大きくして通信量を削減する。なお、本研究では基本的な性質を明らかにするために、キャッシュを1列に並べた構成について検討する。

動画の人気傾向に変動がなければ、過去のアクセスを統計して人気動画をキャッシュするLFUのヒット率が最大となる。しかし、新たに人気動画が追加されたり、ニュースやSNSを通じて古い動画が急に人気になったりすると、アクセス統計の更新が間に合わず、ヒット率が下落してしまう[3, 4]。一方、LRUのように直近のアクセスのみからキャッシュする動画を決定する手法では、人気傾向の急激な変動時もヒット率がほとんど落ちない。そこで、大容量なLFUキャッシュ領域で高いヒット率を実現しつつ、急激な人気変動に備えて少量のLRUキャッシュ領域を組み合わせることで、変動に強いキャッシュネットワークを構成する。異種キャッシュの組み合わせ方は、図2に示す2種類を検討する。

2-2 異種キャッシュを階層的に並べる方法

異なる種類のキャッシュを組み合わせる第1の方法として、多段構成のキャッシュサーバのうち、1台をLRU、その他をLFUで制御する方法を検討する。既存のキャッシュサーバの置き換えアルゴリズムを変更して多段に並べればよいため、シンプルな構成で実現できる。以下ではこの方針を「ヘテロ方式」と呼ぶ。

ある傾向のアクセスが下段キャッシュサーバを経由すると、中段のサーバへ通過するアクセス傾向が変わるため[7]、キャッシュの並べ方によってヒット率および人気変動追従の効果が異なる。下段と中段をLFU、上段をLRUにすると、下段と中段で人気のコンテンツをキャッシュしつつ、上段で急激な人気変動に対応でき

る。しかし、人気変動時にはLFUの人気順位の更新が間に合わず、下段2台のサーバを通過するため、ホップ数が増大する。一方、LRUを下段、中段と上段をLFUにすると、アクセスの変動がなくても中段へ通過する動画アクセスの順位が一意に定まらないため、中段と上段のLFUのヒット率が下がる。しかし、人気変動時も人気動画が下段でキャッシュされるため、中段と上段のLFUキャッシュヒット率とホップ数を維持できる。

2-3 単一キャッシュ領域に異種キャッシュを混在させる方法

異なる種類のキャッシュを組み合わせ

利用する第2の方法として、単一のキャッシュサーバのキャッシュ領域を2つに分割し、大容量なLFUと小容量なLRUで混合する方法を検討する。以下ではこの方針を「ハイブリッド方式」と呼ぶ。

ヘテロ方式と比較すると構成が複雑になるが、LRUとLFUの混合割合をキャッシュサーバ毎に設定できるので、ヒット率向上のための調整が可能になる。また、下段をハイブリッド方式、中段と上段をLFUの多段構成にすれば、実効キャッシュ容量全体のうちLRUが占める割合が小さくなるため、人気変動がない場合も高いヒット率を維持できる。

3 実装と評価

3-1 キャッシュ・アルゴリズムの実装

LRU、LFU、ハイブリッド方式それぞれのキャッシュ・アルゴリズムをRubyで実装した。LRUキャッシュは、最新のアクセス時刻が最も古いものを追い出して新しい動画を挿入する、一般的な方法で実装した。LFUキャッシュは、[5]をもとに一定時間ごとにアクセスログを解析し、キャッシュする動画を更新するよう実装した。LFUとハイブリッド方式のキャッシュの実装を図3、4にそれぞれ示す。

LFUキャッシュは、アクセスログから生成した人気コンテンツのテーブルを一定時間ごとに更新し、テーブルに存在するコンテンツのみをキャッシュする(図3)。このテーブルは人気上位のコンテンツからキャッシュに収まる分だけを取り出して作成する。テーブルの更新時に不要なコンテンツの削除は行わず、テーブルから既に削除されたコンテンツでもキャッシュ領域に存在していればキャッシュヒットする。キャッシュの追い出しは、テーブルから削除されたコンテンツをキャッシュ領域からランダムに1つ選択して削除する。このように実装することで、小さな演算コストでLFUと同等以上のヒット率を達成できる[6]。

ハイブリッド方式のキャッシュは、キャッシュ領域をLRUとLFUの2つの領域に分割して利用する(図4)。分割する割合は事前に設定しておく。(1) キャッシュサーバが動画要求を受けると、(2) その要求をアクセスログに追記する。その後、(3) LRU領域と(4) LFU領域でそれぞれ動画がキャッシュされていないか確認し、

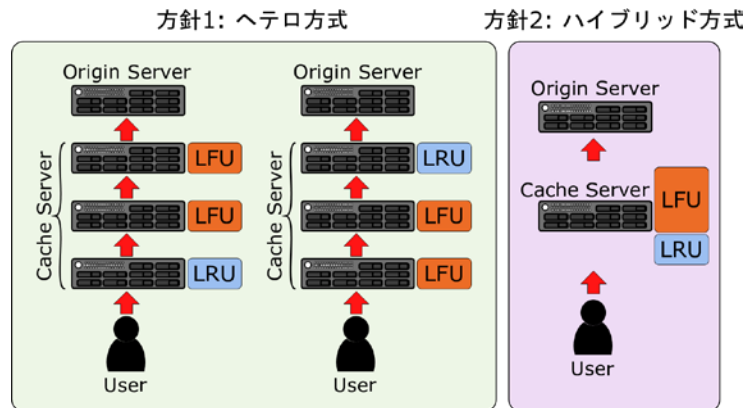


図2. 異種キャッシュを利用する2つの方針

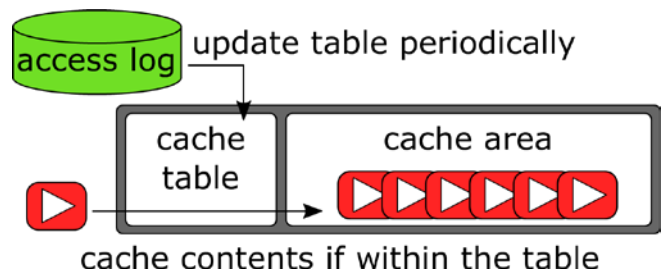


図3. LFUキャッシュの動作

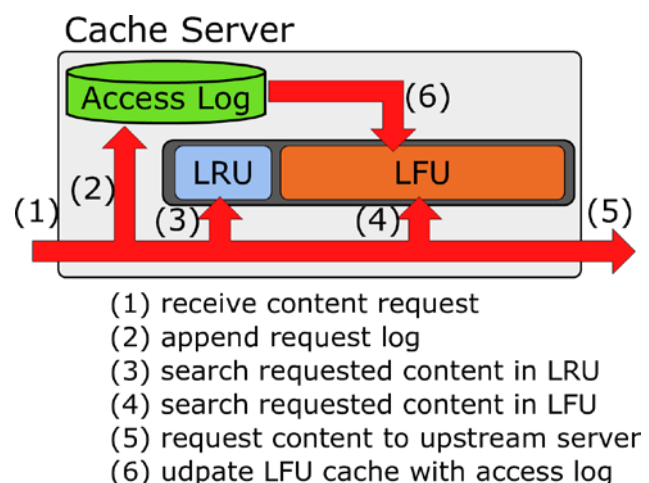


図4. LRU/LFUハイブリッドキャッシュの動作

キャッシュがあればそのまま返答する。キャッシュ領域に動画がなければ、(5) 上段のキャッシュまたは動画配信サーバに要求を転送する。上段から返答された動画は、常にLRU領域でキャッシュされる。LFU領域のテーブルにマッチすれば、LFU領域にも追加する。(6) 一定時間経過後、アクセスログからLFU領域のテーブルを更新する。

階層キャッシュは、これらLRU キャッシュ、LFU キャッシュ、ハイブリッド方式のキャッシュを直列に並べる。キャッシュサーバ間でアクセスログの等の情報交換は行わない。そのため、各キャッシュで保持される動画は、そのキャッシュが受けたリクエストによってのみ決定される。

3-2 評価環境

実装したプログラムを用いてシミュレーション評価を行った。問題を簡単化するため、各動画アクセスは、直前の動画アクセスが完了するまで発生しないように設定した。アクセスされる動画は1020種類、すべての動画を同一ファイルサイズとし、各キャッシュサーバは100個の動画をキャッシュできるように容量を設定した。動画のアクセスは文献[1]をもとに、パラメータ $\theta = 0.475$, $k = 170.6067$ のガンマ分布に従って生成した。LFU キャッシュのテーブルは過去100k回のアクセスをもとに、100kアクセスごとに更新するよう設定した。

急激な人気変動時のヒット率と通信量を検討するため、まずは1020種類の動画のうち1000種類を選択し、500k回アクセスを発生させた。その後、新規動画として残りの20種類を追加し、人気を最上位に設定した。人気最下位の20動画は、その後アクセスされないように設定した。すなわち、動画アクセスは常に1000種類の動画からガンマ分布の偏りで生成される。

実験では、階層キャッシュ全体のヒット率とホップ数を評価した。ヒット率は、ユーザが発生した動画リクエストのうち、Webサーバに到達しなかった割合を示す。一方、通信量はホップ数に比例するため、1つの動画を取得するのに必要な平均ホップ数を算出した。ホップ数は各階層のキャッシュヒット率をもとに算出し、アクセスがk段目でヒットする確率にk段目までのホップ数を乗じた値の総和を求めた。ユーザから直上のキャッシュサーバまで、キャッシュサーバ間、上段のキャッシュサーバからWebサーバまでのホップ数はそれぞれ1と設定し、Webサーバでヒットする確率は1として計算した。

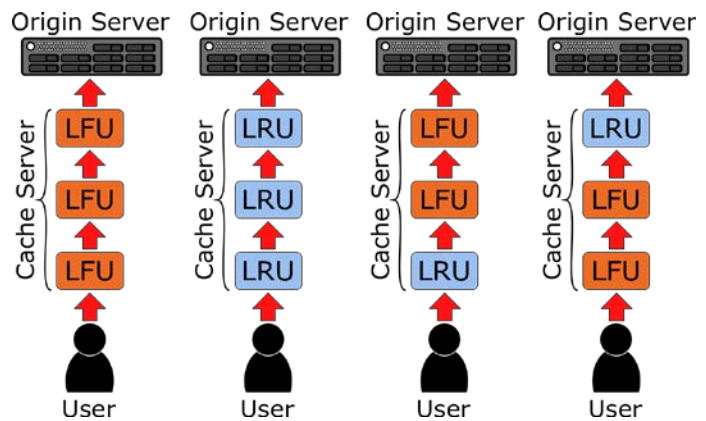


図5. ヘテロ方式のキャッシュ構成

3-3 ヘテロ方式の評価

図5に示すようにキャッシュを直列に並べてアクセスを生成し、ヒット率とホップ数を評価した。それぞれの評価結果を図6, 7にそれぞれ示す。凡例の括弧内は、左から順に下段、中段、上段のキャッシュ・アルゴリズムを示す。

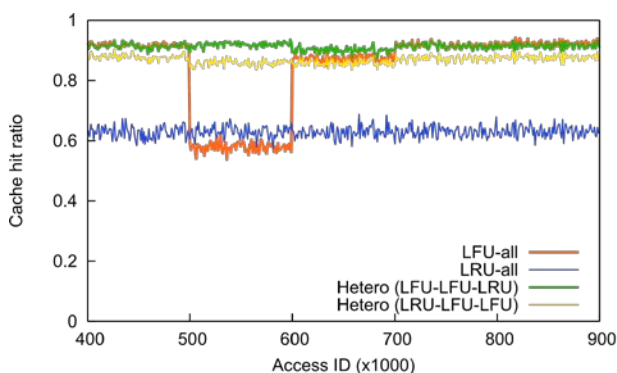


図6. 3段キャッシュ構成全体のヒット率

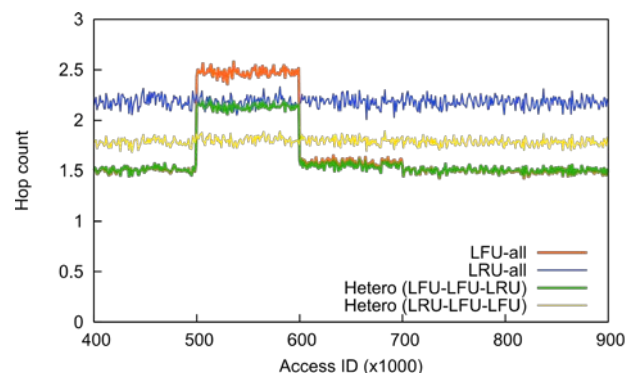


図7. 3段キャッシュ構成全体のホップ数

人気変動がなければLFUを3台並べた構成のヒット率が最も高いが、新規動画が追加されたタイミングでヒット率が大幅に下がっている。この構成では上位30%の動画を保持できるため、アクセス全体の94%がキャッシュヒットする(図1を参照)。そのため、変動直前のヒット率は92-94%程度を推移している。一方、新規に動画が追加されると、3段のキャッシュで保持されている動画の順位がそれぞれ20位だけ下落するため、キャッシュされている動画は上位2-32%となる。ガンマ分布では、これら動画はアクセス全体の59.8%を占めるため、新規動画が追加された直後はヒット率が58-60%程度に落ちている。その結果、ホップ数も急激に増加し、LFUが更新されるまではLRUよりもホップ数が長くなるため、Webサーバとネットワークに大きな負荷を加えてしまう。変動後のヒット率とホップ数が階段状に改善しているのは、下段から順に正確な人気情報が設定されるためである。

LRUの3段構成では、新規動画の追加によるヒット率の変動がほとんどない。しかし、他の構成に比べてヒット率が低く、ホップ数も長いと、通信量の削減効果が低い。これは、LRUがキャッシュ操作に利用するパラメータが、最終アクセス時刻のみであることが原因である。すなわち、動画のアクセス回数を考慮しないため、人気下位の動画がアクセスされた際に、人気上位の動画を追い出してヒット率を落としてしまう。一方、過去のアクセス情報に影響されにくいと、アクセスの変動には強く、ホップ数もほぼ一定の値を推移している。

ヘテロ方式では、LRUを配置する位置によって結果が変化した。LRUを上段にすると、変動時のヒット率が高い一方で、平均ホップ数が増大する。これは、新規動画が下段・中段を通過して上段でヒットするため、変動時には下位2段のホップ数が余計にかかってしまうためである。一方、LRUを下段にすると、ヒット率が5%程度落ちるが、変動時も安定して短いホップ数を維持している。ヒット率が落ちるのは、下段のLRUが中段へ通過するアクセス傾向を変化させるために、LFUで正確な人気順位が算出できなくなることが原因である。また、変動時のホップ数が安定するのは、新規動画がすべて下段でキャッシュされ、中段以上へとアクセスが通過する確率が低くなるためである。これらのことから、下段をLRUにしたヘテロ構成が、ヒット率とホップ数それぞれでバランスよく効果があることがわかった。

3-4 ハイブリッド方式の評価

ハイブリッド方式は1台構成で実験を行い、LRUとLFUの割合を変化させてヒット率を評価した。評価結果を図8に示す。1台構成では1台のみのヒット率を考えればよいと、ホップ数のグラフは縦軸を逆転させた形状となる。

左側の棒グラフはLFU領域が多く、右側はLRU領域が大きい。人気変動がないときは、LRU領域の拡大により、人気動画を確実にキャッシュするLFU領域が減少して効率が悪化するため、ヒット率が緩やかに減少する。一方で、LRU領域を少量追加すると、人気変動時のヒット率が大きく向上する。LFU 100%では人気変動時のヒット率が42%程度下落していたのに対し、全体の30%をLRU領域とすれば、変動時のヒット率の下落幅は10%程度に抑えられる。LRU領域とLFU領域の割合は、要求するヒット率と許容する下落幅に合った形で決定すればよい。

3-5 ヘテロ方式とハイブリッド方式を組み合わせた場合の評価

ヘテロ方式の評価で、LRUは下段に置くとヒット率とホップ数をバランスよく改善できることを確認した。そこで、少量のLRU領域を含むハイブリッド方式のキャッシュを下段に設定し、中段

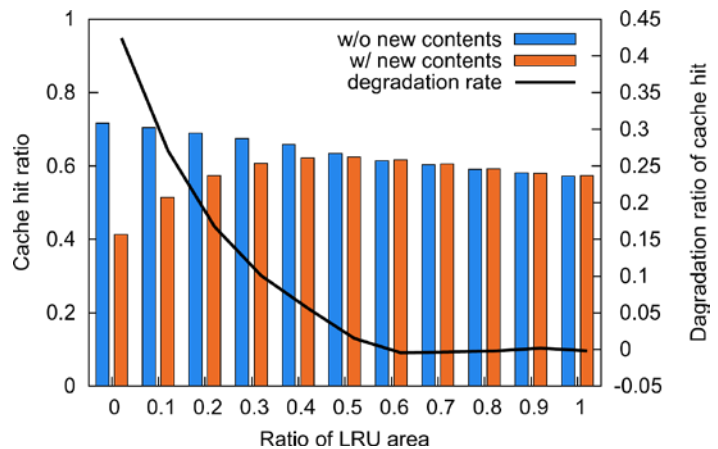


図8. LRUの割合によるヒット率の変化

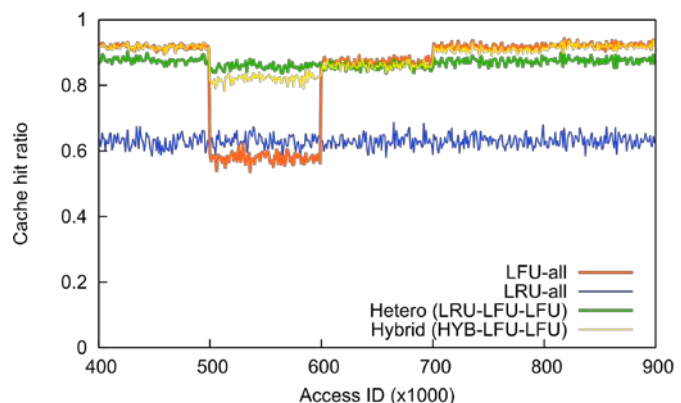


図9. ハイブリッド/3段構成のヒット率の比較

と上段にLFUキャッシュを並べた構成でヒット率とホップ数を評価した。ハイブリッド方式のキャッシュは、LRU 30%、LFU 70%で設定した。評価結果を図9、10にそれぞれ示す。

下段をハイブリッド方式にすると、新規動画の追加によりヒット率が10%程度低下する。下段のLRU 領域で保持できる動画が全体の3%であり、2%の新規動画はすべてキャッシュできる容量である。しかし、人気下位のアクセスも発生するため、新規動画がLRU領域から追い出されることがある。その結果、新規動画の追加によりヒット率が下がり、ホップ数が伸びてしまう。一方、変動がないときのホップ数はLFUに近く、変動時もヘテロ方式と同等のホップ数を維持している。全体の3%のLRU領域は、人気変動がある場合・ない場合で共に良好なヒット率とホップ数を実現できる。

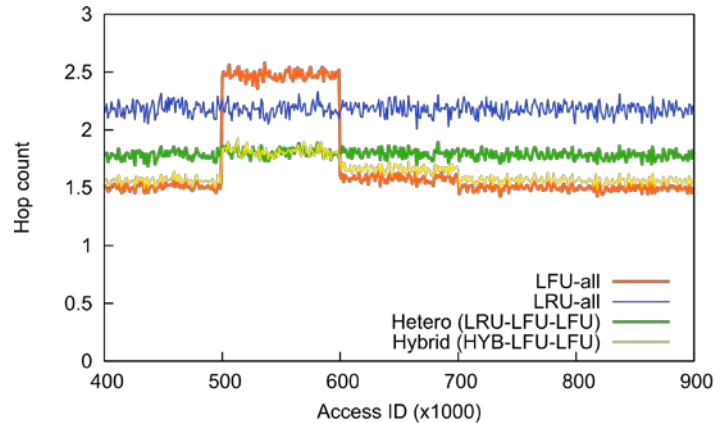


図 10. ハイブリッド構成と 3 段構成のホップ数の比較

4 議 論

4-1 ツリー構造のネットワークにおける効果

3節では、3台のキャッシュを直列に並べたキャッシュ構成で評価を行った。しかし、実際のインターネットにおいて階層キャッシュを利用する際には、ネットワークトポロジはツリー構造に近い。下段からのアクセスパターンが同じならば、ネットワークをツリー構造にしても類似した結果になると予想されるが、下段からの人気変動にタイムラグがあると、異なる結果になるはずである。

例えば、あるキャッシュサーバA、Bが上位のキャッシュサーバCに接続されており、動画の人気傾向がAからBへと地域遷移する状況を考える。すると、Aからの新規リクエストはキャッシュサーバCで常にキャッシュミスし、Bからの新規リクエストは常にキャッシュヒットする。すなわち、2段目以上のキャッシュでも変動に対応する必要があるため、下段だけでなく中段以上もハイブリッド方式を採用することで、多段構成と同じようにヒット率の向上とホップ数の削減が可能である。

4-2 ホップ数削減による副次的効果

本研究では、ヒット率とホップ数に着目して評価を行ったが、消費電力削減と遅延の短縮にも効果がある。ホップ数を削減すると、通信量も同時に削減されるため、転送電力が減少する。また、ホップ数を短縮すると遅延も短縮されるため、低遅延で快適な動画視聴環境の提供に貢献する。加えて、インターネットサービスプロバイダ（ISP）内のキャッシュで通信を完結させられる確率が高くなり、ISPをまたぐ高コストな通信を削減できる。

また、今回キャッシュを多段にしてヒット率を向上させる方針をとったが、同一階層の複数キャッシュをまとめて実効キャッシュ容量を増やす手法[6]と組み合わせると、縦方向・横方向それぞれでキャッシュを効率利用できる。すべてのキャッシュでハイブリッド方式を採用し、各キャッシュが独立して制御する少量のLRU領域と、複数キャッシュサーバで協調動作する大容量なLFU領域を組み合わせることで、ヒット率のさらなる向上が期待できる。

5 まとめ

本研究では、VOD 通信を対象として急激な動画の人気変動に追従する階層キャッシュの検討を行った。ヒット率の高い大容量LFU と、人気変動に強い少量のLRU を混在させることで、変動がないときのヒット率とホップ数を大きく悪化させずに、急激な変動時もそれぞれ大幅に改善できることを確認した。

今後の課題として、ツリー構造のネットワークで効果を検証することが挙げられる。

【参考文献】

- [1] X. Cheng, J. Liu, and C. Dale, "Understanding the Characteristics of Internet Short Video Sharing: A YouTube-Based Measurement Study," *IEEE Transactions on Multimedia*, vol.15, no.5, pp.1184–1194, Aug. 2013.
- [2] X. Che, B. Ip, and L. Lin, "A Survey of Current YouTube Video Characteristics," *IEEE MultiMedia*, vol.22, no.2, pp.56–63, April 2015.
- [3] H. Yu, D. Zheng, B.Y. Zhao, and W. Zheng, "Understanding User Behavior in Large-scale Video-on-demand Systems," *Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006*, pp.333–344, 2006.
- [4] R.K. Sitaraman, M. Kasbekar, W. Lichtenstein, and M. Jain, "Overlay Networks: An Akamai Perspective," *Advanced Content Delivery, Streaming, and Cloud Services*, eds. by M. Pathan, R.K. Sitaraman, and D. Robinson, pp.305–328, John Wiley & Sons, Inc., 2014.
- [5] I. Tatarinov, A. Rousskov, and V. Soloviev, "Static caching in Web servers," *Proceedings of Sixth International Conference on Computer Communications and Networks*, pp.410–417, Sept. 1997.
- [6] T. Shiroma, T. Nakajima, M. Yoshimi, and T. Yoshinaga, "An Efficient Cache Grouping Strategy for Multinode Cache Networks," *Proceedings of the 8th International Workshop on Autonomous Self-Organizing Networks*, pp.295–298, Dec. 2015.
- [7] C. Williamson, "On Filter Effects in Web Caching Hierarchies," *ACM Transactions on Internet Technology*, vol.2, no.1, pp.47–77, Feb. 2002.

〈発表資料〉

題 名	掲載誌・学会名等	発表年月
An Efficient Cache Grouping Strategy for Multinode Cache Networks	Proc. of the 8th International Workshop on Autonomous Self-Organizing Networks (ASON'15)	2015/12
動画の人気変動に追従する異種キャッシュ混在ネットワークの検討	電子情報通信学会, 信学技報 CPSY2015-154, 115/518	2016/03