

アクセス分布に基づいた多段キャッシュ方式による 性能の評価

青山朋義^{†1} 黒瀬浩^{†2}

概要: コンテンツ配信の効率化を図るため CDN (Content Delivery Network) や CCN (Content Centric Network) などで通信経路短縮方法が提案されている。キャッシュ容量の上限に達した際、キャッシュエントリを廃棄するため通信経路長が長くなる。本稿では、多段キャッシュ方式において、キャッシュサーバ間のエントリを移動・複製することで経路長の短縮効果を Zipf アクセス分布でシミュレーション評価した。

キーワード: コンテンツキャッシュ, 多段キャッシュ, 通信経路長

Performance Evaluation with Cascaded Cache method based on Access Frequency

Tomoyoshi AOYAMA^{†1} Hiroshi KUROSE^{†2}

Abstract: Content Delivery Network (CDN) or Content Centric Network (CCN) has been proposed that path length shortening method achieve efficiency of content distribution. When cache size reached limit, Path get longer due to discard a cache entry. This paper shows evaluation of shortening effect by moving and copying the entry between the cache server by Zipf access distribution within Cascaded Cache method

Keywords: Content Cache, Cascaded Cache, Path Length

1. はじめに

Web2.0 の普及により、YouTube[1]やニコニコ動画[2]に代表される CGM (Consumer Generated Media) コンテンツが急増している。CGM コンテンツは画像や動画が用いられデータ量が従来のテキストベースのコンテンツより大幅に増大するため、近年のインターネットトラフィックの多くをコンテンツデータが占めている。コンテンツ配信の効率化を進める技術として CCN (Content Centric Network) [3]や CDN (Content Delivery Network) などコンテンツ配信技術が研究・運用されている。

従来のネットワークは、IP アドレスを用いたクライアント/サーバモデルがベースであり、DNS による名前解決によってコンテンツを保持しているサーバ IP アドレスを特定し、コンテンツを取得している。しかし、コンテンツ要求者はコンテンツの所在には関心がなく、要求したコンテンツが期待した速度以内で取得できればよい。CCN, CDN

ではコンテンツ要求者はコンテンツ取得場所を意識することなく取得でき、データ取得場所の誘導によってコンテンツの取得にかかる通信経路の短縮を図る。

CDN では、同一のコンテンツを保持したサーバを複数用意し、通信経路長が長ならないように地理的に分散配置し、名前解決をする際に、コンテンツ要求者からの距離が近いサーバまたは負荷の少ないサーバなどに誘導する [4]。

CCN では、コンテンツルータがコンテンツの保持と誘導を行っており、コンテンツ要求者がコンテンツを保持しているサーバを知らなくともネットワーク自体が要求コンテンツの発見と転送を行う。

キャッシュサーバのキャッシュ容量の上限に達した際、キャッシュ置換方法によりキャッシュサーバ内のエントリが廃棄されるため、廃棄されたエントリと同一コンテンツの再取得の際にコンテンツを保持している別のサーバまでアクセスが発生し、通信経路長が長くなる。

本稿では、多段キャッシュ方式において、通常のキャッシュ登録方法に加え、サーバ側へのキャッシュを考慮したキャッシュ登録方法によるキャッシュサーバ間のエントリを移動・複製することによる通信経路長短縮の効果をアク

^{†1} 金沢工業大学大学院工学研究科
Kanazawa Institute of Technology
^{†2} 金沢工業大学工学部・工学研究科
Kanazawa Institute of Technology

セス分布 Zipf[5]に基づいたシミュレーションにより評価する。

2. 関連技術

多段キャッシュ構成、コンテンツキャッシュにおける要素技術としてキャッシュサーバでのエントリの登録方法と置換方法、Zipf アクセス分布について説明する。

2.1 多段キャッシュ構成

多段キャッシュモデルを図1に示す。Client がコンテンツ要求者、 C_n がキャッシュサーバ、 M_n がキャッシュスロット数である。キャッシュサーバにはコンテンツメタデータがキャッシュされる。CMS はオリジナルのコンテンツメタデータの管理を行う Content Management System である。

初期状態では、全てのコンテンツが CMS で保持されており、各キャッシュサーバのエントリは空である。コンテンツの取得時には、Client はコンテンツの Request をまず C_0 に送る。 C_0 は要求コンテンツと同一のエントリの有無を確認、存在する場合、コンテンツメタデータを Client に返す。一方エントリが存在しない場合、Request を1段上位に送る。上位のキャッシュサーバはこれを繰り返し、経路上でコンテンツメタデータが見つからない場合、終端の CMS でコンテンツメタデータを取得できる。

C_0 以外のノードでコンテンツがヒットした際には、キャッシュ登録方法に従ってエントリの移動・複製が行われる。その際、エントリの登録を行うキャッシュサーバのキャッシュ容量が上限に達している場合、キャッシュ置換方法により、エントリの選択が行われ、エントリの置換が発生する。

2.2 キャッシュ方法

2.2.1 キャッシュ登録方法

多段キャッシュで用いられるキャッシュ登録方法として以下の3つある。

- MCD (Move Copy Down) : ヒットしたノードの1段下位に移動し、エントリを削除
- LCD (Leave Copy Down) : ヒットしたノードの1段下位に複製
- LCE (Leave Copy Everywhere) : ヒットしたノードの下位すべてのノードに複製

ヒットしたノードの1段末端よりエントリを移動する MCD、ヒットした末端の1段末端よりエントリを複製する LCD、ヒットしたノードから末端までの経路上の全てのノードに複製する LCE などがある[6]。他にも各キャッシュサーバにおいて確率により登録判断をする方法[7]がある。キャッシュ登録方法はキャッシュサーバのサービス起動時に1つ設定する。多段キャッシュ方式における動作例を図2に示す。

コンテンツ要求者 client より要求されたコンテンツのメ

タデータと同一のエントリが C_0 , C_1 にはなく C_2 にあるとする。Client から C_0 に Request (req) を送り、① (C_0 の Request が送られてきた際の処理) では要求コンテンツと同一のエントリの有無を確認、ない場合、1段上位のキャッシュサーバに Request を送る。② (C_1 の Request が送られてきた際の処理) でも①と同様である。③ (C_2 の Request が送られてきた際の処理) も同様に要求コンテンツと同一のエントリの有無を確認、保持していた場合、エントリを下位のキャッシュサーバへ送る。この際、キャッシュ登録方法に MCD を使用していた場合、エントリの移動を行うため、エントリが削除される。④ (C_1 の Response (rsp) が送られてきた際の処理) で送られてきたコンテンツをキャッシュサーバのエントリに登録し、下位のキャッシュサーバに転送する。⑤ (C_0 の Response が送られてきた際の処理) ではコンテンツ要求者に対してコンテンツを転送するが、LCE を使用していた場合エントリの登録を行う。

2.2.2 キャッシュ置換方法

キャッシュサーバの容量上限に達した際のキャッシュ置換方法を挙げる。

- LRU (Least Recently Used) : アクセスが最も古いものを廃棄する
- LFU (Least Frequently Used) : アクセスが最も少ないものを廃棄する

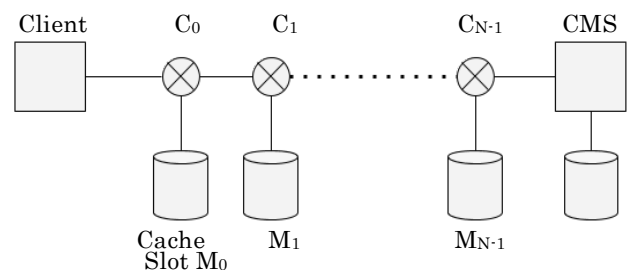


図1 多段キャッシュモデル
 Figure 1 Cascaded Model

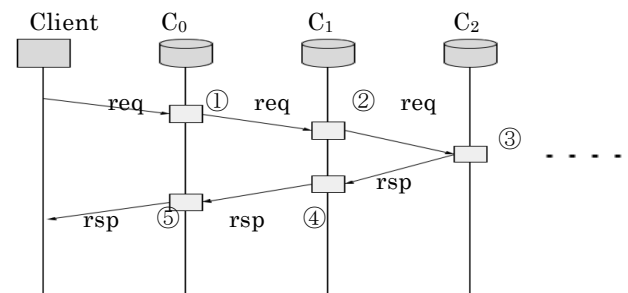


図2 多段キャッシュ方式の動作例
 Figure 2 Cascaded Cache method Example

- FIFO (First-in First-out) : 登録順に廃棄する
- Random (Random Probability) : 確率により無作為に選択し、廃棄する

ユーザの位置によって確率的に LRU を実行する手法[8]や、置換をコストにより変更する手法[9]があるが、経路上のノードの状態を把握するために経路上のノード間の通信が必要となる。

図 2 中の④, ⑤において, エントリが送られてきた際に, Cache Slot の容量が上限に達していた際に, キャッシュ置換方法を用いてエントリの廃棄が行われる。

2.3 アクセス分布

Web2.0 のコンテンツのアクセス頻度は一般に Zipf 分布 (式 1) に従うとされている。そのため様々なシミュレーションでネットワークトラヒックとして使用されている。

$$f(k; \alpha, N) = \frac{1/k^\alpha}{\sum_{n=1}^N 1/n^\alpha} \quad (1)$$

式中の N はコンテンツ数, k がランキング, 係数 α がアクセス頻度の偏りを示す。 $\alpha=1.0$ が通常の Zipf 関数となり, α の値が大きいくほどコンテンツ毎のアクセス頻度の偏りもでにくくなる。 CGM コンテンツのようにコンテンツ毎のアクセス頻度の偏りがでやすいもののシミュレーションを行う際には α の値も大きくなる。

Zipf 係数 $\alpha=1.0$ の他に, 0.6, 0.9, 2.0, 2.5 以下などが使用されている[10]。同様に 0.6, 1.0 が使用されている[11]。

3. 評価方法

3.1 シミュレーション環境

表 1 にシミュレーション環境を示す。Linux 環境下で Virtual Box を用いて内部ネットワークを構成し, 仮想環境での多段キャッシュモデルを作成した。キャッシュサーバとしてメモリ上で動作する Key-value 型の揮発性データベースである Memcached[12]を使用し, 操作には Python 言語で提供されている python3-memcached1.51 パッケージを用いた。

3.2 シミュレーションパラメータ

シミュレーションに用いたパラメータを表 2 に示す。

コンテンツ数を 1000 とし, データセンター内での運用を考える。複雑ネットワークにおける Barabasi-Albert モデル[13]ではノード数が 100 であった場合, 1 つの経路の経路長は平均で 5 程度となる。よってノード数を 5 とし, VirtualBox[14]を用いて Client を合わせた 6 台でネットワークを構成した。Cache Slot は, コンテンツ数に対しての割合とし, 終端のサーバが全コンテンツを保持しているものとして, 経路上のキャッシュサーバですべてのコンテンツを保持できる 25%を最高値とし 5%刻みで 10%までの範囲 250, 200, 150, 100 を用いる。アクセス頻度には Zipf

関数によるアクセス分布を利用した。係数 α が 1.0 のとき一般的な Zipf 関数となるが, コンテンツ毎のアクセス頻度の偏りがでにくいため, CGM コンテンツのようなアクセス頻度の偏りがでやすいものに適していない, そのため YouTube のアクセス頻度に近似している $\alpha=1.5$ [10]も使用する。キャッシュ登録方法は本稿の目的である通信経路長短縮に大きく関わってくるため, MCD, LCD, LCE の 3 方式を用いる。キャッシュ廃棄方法は前述したように複数存在するが, 一般に広く使われているのは LRU であるためこれを採用した。キャッシュするデータ形式として, コンテンツ取得場所の誘導のためコンテンツメタデータを想定し, Content ID を 4 桁の文字列, データを IP アドレスまたは URL とし, 8 桁の文字列とした。またキャッシュサーバ内では同一コンテンツの重複はできないものとする。

先行研究における, キャッシュ容量の上限に達した際, キャッシュエントリの廃棄による, 再取得時の通信経路長が長くなる問題に対し, 本稿では, キャッシュエントリの

表 1 シミュレーション環境

項目	諸元
ホスト OS	Ubuntu14.04.5 LTS
HDD	500GB
CPU	Intel Core i3-4150
CPU クロック	3.50GHz
HT (Hyper Thread) 数	4/8
物理 Memory	8GB
仮想環境	Virtual Box4.3.36
ゲスト OS	Ubuntu16.04.1 LTS
仮想 HDD	8GB
仮想 Memory	512MB
CPU 割当率	100%
Key Value Store Server	Memcached1.4.25
プログラム言語	Python3.5.2

表 2 シミュレーションパラメータ

パラメータ	値
コンテンツ数	1000
アクセス回数	100000
経路長 N	5
スロット数 M	100, 150, 200, 250
Zipf 係数 α	1.0, 1.5
キャッシュ登録方法	MCD, LCD, LCE
キャッシュ廃棄方法	LRU
Content ID	4 桁
データ	8 桁

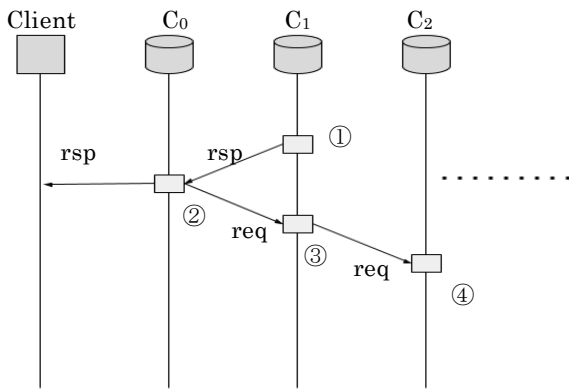


図 3 MCU 動作例
 Figure 3 MCU Example

廃棄ではなく、サーバ側にある 1 段上位のキャッシュサーバにエントリを移動することによる通信経路長短縮を提案する。これにより、再取得の際の通信経路長が長くなることを緩和する。1 段上位のキャッシュサーバにあげることを MCU (Move Copy Up) と呼ぶことにする。

1 段上位にキャッシュ置換方法で選択されたエントリと同一のコンテンツを保持するエントリがあった際には、エントリを廃棄し、同一コンテンツのエントリが保持されていない場合に MCU を実行する。同一コンテンツのエントリが存在しないかつ、キャッシュ容量の上限に達している場合は再度キャッシュ置換方法により、エントリを選択し、MCU を実行する。MCU の動作例を図 3 に示す。

C_1 でコンテンツがヒットし、 C_0 , C_1 のキャッシュ容量は上限に達しており、 C_2 のキャッシュ容量は上限に達していないものとする。① (C_1 でコンテンツがヒットした際の処理) で要求されたコンテンツと同一のエントリを返す。エントリはキャッシュサーバ起動時に設定されたキャッシュ登録方法により、下位のキャッシュサーバ C_1 に移動・複製される。② (C_0 でエントリを受信した際の処理) では、受信したエントリを Client に送り、設定されたキャッシュ置換方法により、エントリを選択、 C_0 内でエントリの置換を行う。選択されたエントリを C_1 に送る。③ (C_1 における MCU によるエントリを受信した際の処理) では、受信したエントリと同一のエントリの有無を確認、ない場合、キャッシュ置換方法により C_1 でエントリを選択し、 C_0 から受信したエントリを登録、選択されたエントリを再度 MCU する。④ (C_2 における MCU によるエントリを受信した際の処理) では送られてきたエントリと同一のエントリの有無の確認、ない場合エントリの登録、ある場合はエントリの廃棄となる。

ユーザ側へのキャッシュ登録方法が MCD であった場合は、MCU が行われても必ず 1 段上位のキャッシュサーバには容量に空きができるため、MCU が繰り返し実行されることはなく、LCD, LCE の際に繰り返し実行される。

評価項目として、応答速度、キャッシュヒット率、キャッシュ生存率などがあるが、本稿では、通信経路長と置換回数に着目し、先行研究と提案手法の比較を行う。コンテンツ取得までの hop 数を経路長とする。同じパラメータのシミュレーションを 5 回試行し、その平均を取る。

4. 性能評価

各グラフで平均通信経路長が 1 に収束したところまでで拡大し、移動平均 8 で描画している。平均通信経路長は横軸をアクセス頻度、縦軸を平均通信経路長とし対数でとっている。改善率は横軸をアクセス頻度、縦軸を先行研究を 1 として比率を出している。平均経路長に関して、Cache Slot 毎に経路長の収束に必要なアクセス頻度以外に変化が見られなかったため Zipf 係数 $\alpha=1.0$, 1.5 での平均経路長のグラフを載せることとする。

$\alpha=1.0$, $M=100$ での平均通信経路長を図 3, 改善率を図 4 に示す。各キャッシュ登録方法と MCU を使用したほうが経路長の短縮と 1 への収束が早いことがわかる。経路長の収束は順に LCE, LCE+MCU, MCD, LCD, MCD+MCU, LCD+MCU で早く収束おり。アクセス頻度の低いものでも経路長の短縮の効果が見込める。経路長短縮の改善率は LCE, LCD, MCD の順で高くなっている。最大で MCD は約 74%, LCD は約 54%, LCE は約 39%改善されている。MCD, LCD ともにアクセス頻度が 600 程度までは改善されている。LCE では経路長収束までのアクセス頻度が高いため 2000 程度まで改善されている。

$\alpha=1.5$, $M=100$ での平均通信経路長を図 4, 改善率を図 5 に示す。 $\alpha=1.0$ と同様に各キャッシュ登録方法と MCU を使用したほうが経路長の短縮と 1 への収束が早いことがわかる。経路長の改善率は LCD, LCE, MCD の順で高くなっている。最大で MCD は約 36%, LCE は約 31%, LCD は約 17%改善されている。MCD, LCD ともにアクセス頻度が 50 程度までは改善されている。LCE では 150 程度まで改善されている。

$\alpha=1.0$ のキャッシュ登録方法毎の置換回数比率を表 3 から表 5 に、 $\alpha=1.5$ のキャッシュ登録方法毎の置換回数比率を表 6 から表 8 に示す。置換回数比率は先行研究を 1 として計算している。MCD と LCD の置換回数比率は Cache Slot が減少すると、比率も減少する傾向にある。反対に LCE は Cache Slot が減少すると増加する傾向にあった。

先行研究では、Cache Slot が十分にあると、MCD や LCD では Cache Slot に空きができてしまう、Cache Slot が減少すると空きがなくなるため置換回数が増加してしまうが、MCU を併用していた場合には、Cache Slot によらず十分なアクセス数になると常に Cache Slot の上限に達しているので置換回数に大きな変化はない。LCE では、経路上全てに登録を行うため Cache Slot に空きができづらいため、MCU を併用している際と条件が変わらず、Cache Slot の

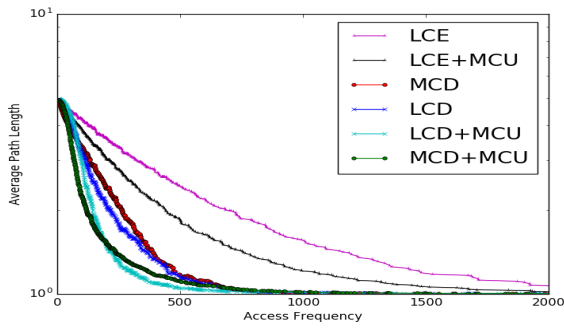


図4 平均通信経路長 ($\alpha=1.0$, $M=100$)

Figure 4 Average Pass Length ($\alpha=1.0$, $M=100$)

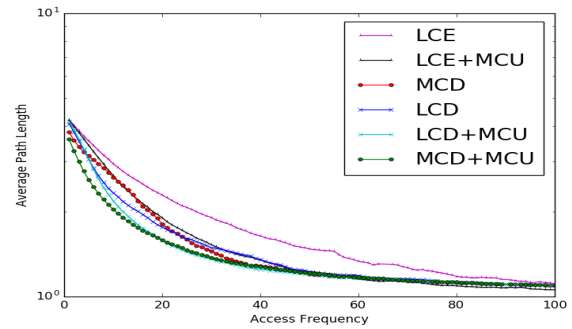


図6 平均通信経路長 ($\alpha=1.5$, $M=100$)

Figure 6 Average Pass Length ($\alpha=1.0$, $M=100$)

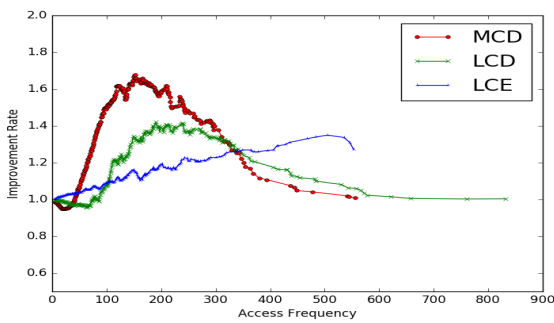


図5 改善率 ($\alpha=1.0$, $M=100$)

Figure 5 Improvement Rate ($\alpha=1.0$, $M=100$)

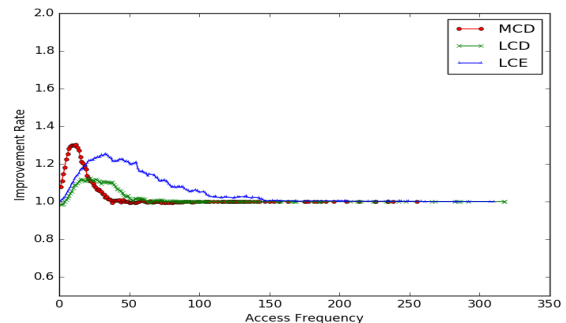


図7 改善率 ($\alpha=1.5$, $M=100$)

Figure 7 Improvement Rate ($\alpha=1.5$, $M=100$)

表3 置換回数比率 ($\alpha=1.0$, MCD)

Table 3 Replacement Rate ($\alpha=1.0$, MCD)

	Access 1K	Access 10K	Access 1M
CacheSlot100	1.05	1.37	1.69
CacheSlot150	1.19	1.49	2.21
CacheSlot200	4.21	1.54	3.21
CacheSlot250	-	1.69	3.69

表6 置換回数比率 ($\alpha=1.5$, MCD)

Table 6 Replacement Rate ($\alpha=1.5$, MCD)

	Access 1K	Access 10K	Access 1M
CacheSlot100	-	0.94	1.96
CacheSlot150	-	1.12	2.22
CacheSlot200	-	0.41	2.55
CacheSlot250	-	-	2.90

表4 置換回数比率 ($\alpha=1.0$, LCD)

Table 4 Replacement Rate ($\alpha=1.0$, LCD)

	Access 1K	Access 10K	Access 1M
CacheSlot100	0.98	1.32	1.41
CacheSlot150	1.02	1.34	1.51
CacheSlot200	0.98	1.40	1.61
CacheSlot250	1.04	1.35	1.69

表7 置換回数比率 ($\alpha=1.5$, LCD)

Table 7 Replacement Rate ($\alpha=1.5$, LCD)

	Access 1K	Access 10K	Access 1M
CacheSlot100	1.21	1.19	1.36
CacheSlot150	-	1.13	1.39
CacheSlot200	-	1.01	1.41
CacheSlot250	-	1.05	1.42

表5 置換回数比率 ($\alpha=1.0$, LCE)

Table 5 Replacement Rate ($\alpha=1.0$, LCE)

	Access 1K	Access 10K	Access 1M
CacheSlot100	1.05	1.95	2.25
CacheSlot150	0.99	1.47	1.77
CacheSlot200	1.00	1.23	1.59
CacheSlot250	1.05	1.17	1.48

表8 置換回数比率 ($\alpha=1.5$, LCE)

Table 8 Replacement Rate ($\alpha=1.5$, LCE)

	Access 1K	Access 10K	Access 1M
CacheSlot100	1.14	1.20	1.56
CacheSlot150	-	0.90	1.42
CacheSlot200	-	0.97	1.11
CacheSlot250	-	0.98	1.09

減少に伴い置換回数比率も増加した。MCD と LCD で置換回数比率の減少に差がある理由は LCD はキャッシュエントリの複製のため Cache Slot の空きが減少することによると考える。

5. おわりに

本稿では、通信経路長短縮の効果を高めるためにキャッシュエントリの廃棄ではなく 1 段上位のキャッシュサーバに移動させる手法を提案した。シミュレーションによる評価で先行研究との比較を行った結果、エントリの置換回数の増加はあるが、多段キャッシュ方式においては、キャッシュ容量の上限に達した際に、エントリの廃棄ではなく 1 段上位への移動 (MCU) を行った方が経路長の短縮を見込めることが確認された。MCU を使用した際には、アクセス頻度の偏りによらずアクセス回数の少ないコンテンツに対しても経路長の短縮が見込め、経路長 1 の収束にかかるアクセス回数も先行研究に比べ少なく済む。また、コンテンツのアクセス頻度の偏りがでにくい方が通信経路長短縮の効果が出やすく、アクセス回数の少ないコンテンツでも素早く取得できる。

今後の課題として、実データを用いた応答速度の比較やサーバ側へのキャッシュ方法の検討、キャッシュ置換方法の違いによる性能評価、ネットワークトポロジの変更などが挙げられる。

参考文献

- [1] YouTube, <http://www.youtube.com>, (参照 2016-12-17).
- [2] ニコニコ動画, <http://www.nicovideo.jp>, (参照 2016-12-17).
- [3] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, 'Networking named content,' ACM CoNEXT 2009. 2009.
- [4] Akamai Technologies, <http://www.akamai.com>, (参照 2016-12-17).
- [5] L. Breslan, Pei Cao, Li Fan, G. Phillips, and S. Shenker, 'Web caching and Zipf-like Distributions: Evidence and Implications,' Proceedings INFOCOM'99, Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Vol.1, IEEE, 1999.
- [6] N. Laoutaris, S. Syntila, and I. Stavrakakis, 'Meta algorithms for hierarchical web caches,' Performance, Computing, and Communications 2004, IEEE International Conference on IEEE, 2004.
- [7] S. Arianfar, P. Nikander, and Jörg Ott, 'Packet-level caching for information-centric networking,' ACM SIGCOMM, ReArch Workshop, 2010.
- [8] Psaras, Ioannis, Wei Koong Chai, and George Pavlou, 'Probabilistic in-network caching for information-centric networks,' Proceedings of the second edition of the ICN workshop on Information-centric networking, ACM, 2012.
- [9] Wenzhong Li, Edward Chan, Yilin Wang, Daoxu Chen, and, Sanglu Lu. 'Cache placement optimization in hierarchical networks: Analysis and performance evaluation,' International Conference on Research in Networking. Springer Berlin Heidelberg, 2008.
- [10] Rossi, Dario, and Giuseppe Rossini. 'Caching performance

- of content centric networks under multi-path routing (and more),' Relatório técnico, Telecom ParisTech (2011).
- [11] Rosensweig, Elisha J., Jim Kurose, and Don Towsley, 'Approximate models for general cache networks,' IN-FOCOM, 2010 Proceedings IEEE, IEEE, 2010.
 - [12] "memcached", <http://memcached.org>, (参照 2016-12-17).
 - [13] Barabási, Albert-László, and Réka Albert. 'Emergence of scaling in random networks,' Science, 1999.
 - [14] "VirtualBox", <http://www.virtualbox.org/>, (参照 2016-12-18).