L-016

Partial Cache 型 CDN の Simulation による Cache Hit 率の評価

Evaluation of Cache Hit Ratio by Simulation in Partial Cache type CDN

栗原 望*
Nozomu Kurihara
(*) 日本大学大学院 理工学研究科
Graduate School of Science and Technology, Nihon University

栗野 俊一**
Shun-ichi Kurino
(**) 日本大学理工学部
College of Science and Technology, Nihon University

1. 背景と目的

今日，SNS 等の普及により，Web Contents 利用者の要求は高まってきている．よって，Web Contents を高速に Web Contents 利用者へ配信することは重要であると考えられる．Web Cache Service は，Web Contents の配信を高速化する為の技術の 1 つであり，それには様々な工夫がなされている [1, 2]．一般に，Web Contents 配信を高速化する為には，より大きな Cache 容量が必要となる(Cache Server がより多くの Web Contents を，より多く保持できるようにする為)．そこで我々は，Cache する Content を複数個に分割し，それを複数の Cache Server に分散的に配置することにより，その Web Content の配信速度を維持したまま，Cache Size を小さく抑える仕組みである Partial Cache を提案している [3].

本報告では，Partial Cache の仕組みを利用した Cache Server 群と単なる Cache Server 群とを比較して，性能評価を行った．今回は，性能の 1 つである Cache Hit 率に着目した．Simulation によって Cache Hit 率を測定し，Partial Cache の有効性を確かめた.

2. 先行研究

Partial Cache とは，Cache Server が，1 つの Content に対して，その一部分 (以下，Piece と呼ぶ) だけを Cache することにより，Content 1 つ当たりの Cache Size を小さく抑える仕組みである.

この仕組みを利用した Cache Server 群は，ある Content が要求されたとき，Cache している Piece を Web Contents 利用者 (以下，Client と呼ぶ) へ配信しつつ，Cache していない Piece を他の Cache Server から並行して獲得し，順次 Client へ配信する (Client へ直接 Content 配信を行う Cache Server のことを他の Cache Server と区別する為，以下，Collect Server と呼ぶ)．但し，Collect Server は Content 配信時に Client を待たせることのないよう，Client へ配信中の Piece を配信し終える前に，次の Piece を他の Cache Server から獲得し終えるものとする．この条件を満たす為には，Client・Collect Server 間の通信速度と Collect Server・Cache Server 群間の通信速度の比 (以下，T と呼ぶ) の値によって，各 Piece Size を調整する必要がある.

T の値が大きくなると，各 Piece Size をより小さく抑えることができる [3]．T の値が大きくなる，つまり各 Piece Size が小さくなればなるほど，Cache Server 群はより多くの種類の Content を Cache することが可能となり，Cache Hit 率が上がることが期待できる.

しかし，[3] では，Cache Hit 率に関して，実験や Simulation による具体的な分析は行われていない．よって，以下では，Partial Cache を利用した Cache Server 群における Cache Hit 率の評価を Simulation により行う.

3. Simulation による Cache Hit 率の評価

Partial Cache の導入によって，Cache Hit 率がどのように変化するのかを Simulation によって測定した．図 3 の Simulation は，表 1 の Parameter 下で行なった．また，[zipf 分布: s = 0.1] として Parameter を変更して予備 Simulation を行なった結果が 図 4 である.

Simulation は T の値を変動させて行なっている．これは，前節で述べた， Piece Size が小さくなると Cache Hit 率が上がるという予想を確認する為である.

表 1　Simulation で用いた基本となる Parameter

Parameter 名	値
P_1: Contents 総要求回数	5000 (回)
P_2: Cache Server 台数	200 (台)
P_3: Cache Server 容量	1000 (MB)
P_4: Content Size	10 (MB)
P_5: Content 種類数	1000 (種類)
P_6: Content 要求確率分布: zipf 分布	s = 1.0
P_7: Piece 分割数	6
P_8: T	$0.1 \leqq T < 1$
P_9: $Piece_0$ の Size	略 ([3])
P_{10}: 各 Cache Server が Cache 可能な総種類数	P_3 / P_4
P_{11}: 各 Cache Server への Content 要求頻度分布	略 (②)

また，図 3，図 4 の Simulation において，T 以外の Parameter は固定して Simulation を行なった．これは，Partial Cache の利用とは無関係に，各 Parameter が Cache Hit 率に与える傾向(上がる/下がる)は変わらないと考えた為である．以下，その根拠を示す.

① Cache Server 1 台あたりが Cache することが可能な Content 総種類数 P_{10} は,

$$\frac{P_3}{P_4} = P_{10}$$

であり (今回は話を簡単にする為に，P_4 の Size は全種類同じとして扱う)，特に，Partial Cache を利用するのであれば，分母を P_9 とし,

$$\frac{P_3}{P_9} \fallingdotseq P_{10}$$

で近似することが可能である(これは，$Piece_0$ 以外の Piece ($Piece_1$ 〜 $Piece_5$) を持つ Cache Server は，全 Cache Server 台数のうち，($P_7 - 1 = 5$) 台のみであり，ほとんどの Cache

Server は各 Content に対して $Piece_0$ しか持たない為である．また，$Piece_0$ の Size は P_7, P_8 から計算して求められる [3]). P_{10} の値が大きい，つまり，より多くの Content を同時に保持できるようになれば，より多くの種類の Content 要求に対して，Cache Server は Cache Hit する確率を上げることができる． また，一般に Size において，

$$P_4 > P_9$$

であるので，Partial Cache を利用した方が，P_{10} の値がより大きくなることがわかる．

② Cache Server 1 台あたりにおける各 Content の Access 頻度分布 P_{11} を考える．P_{11} は，

$$\frac{P_1}{P_2} \times \frac{\frac{1}{rank^s}}{\sum_{n=1}^{P_5} n^s} = P_{11} \quad (1 \le rank \le P_5)$$

である(今回は，話を簡単にする為に，各 Cache Server への Access は一様分布を仮定しているので，$\frac{P_1}{P_2}$ としている)．また，Partial Cache を利用する場合は，Content の分散配置を行うことによって，各 Cache Server の持つ Content が他の Cache Server へと同期される．つまり，実質的に，

$$\frac{P_1}{1} \times \frac{\frac{1}{rank^s}}{\sum_{n=1}^{P_5} n^s} = P_{11} \quad (1 \le rank < P_5)$$

となる．各 Cache Server への Access は一様分布を仮定しているが，上式から Cache Server への Access は分散せずに 1 つに集中しているかのように振る舞うことがわかる．つまり，同一 Content の要求を異なる Cache Server へ行なっても Cache Hit するので，分散配置を利用した方が，Cache Hit 率が上がることがわかる．

③ $s = 1.0$, $s = 0.1$ のとき，P_{11} はそれぞれ，

図 1　Content 要求頻度分布 ($s = 1.0$)

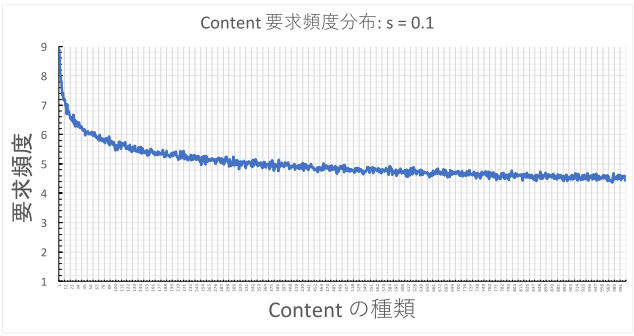

図 2　Content 要求頻度分布 ($s = 0.1$)

となる．前者は，Content 毎の要求頻度 にかなり偏りがあり，Content が 合計で何回 Access されるのかという期待値としても，Cache Hit する確率が高いといえる．一方，後者は，Content 毎の要求頻度にあまり偏りがなく，Cache 内に存在し続ける Content はない，つまりは Cache Hit する確率が低くなるといえる．

4. Simulation による Cache Hit 率の測定結果

Partial Cache を利用する Cache Server 群と単なる Cache Server 群とで Cache Hit 率の比較を Simulator を用いて行なった．この結果を図 3，図 4 に示す．

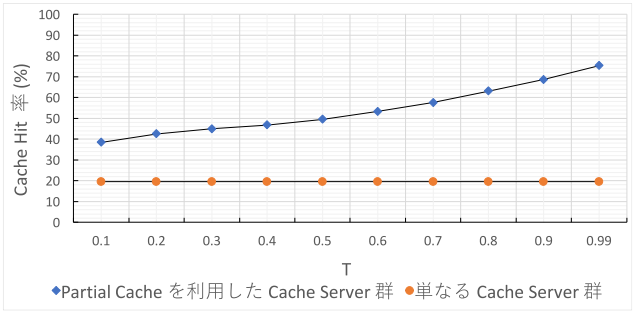

図 3　Partial Cache を利用した Cache Server 群と単なる Cache Server 群とでの Cache Hit 率の測定結果比較

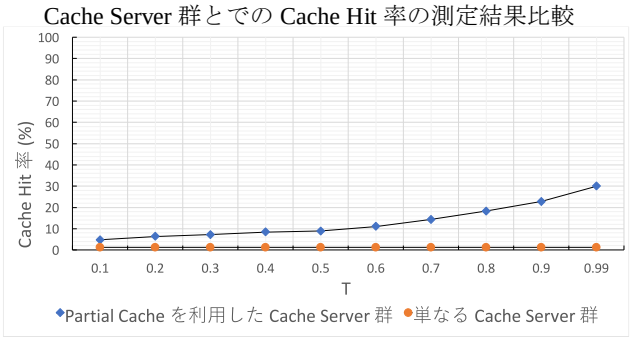

図 4　Partial Cache を利用した Cache Server 群と単なる Cache Server 群とでの Cache Hit 率の測定結果比較 ($s = 0.1$)

5. 結果

① より，T が大きくなるにつれて(P_9 が小さくなるにつれて) Cache Hit 率が上がっていることがわかる．また ② より，T の値が小さくても(P_9 の値が P_4 に近い値であっても)，Cache Hit 率に差があることがわかる．さらに ③ より，P_{11} の偏りが，Cache Hit 率に大きく影響していることがわかる．

第 3 節で述べた理論通り，Partial Cache を利用した Cache Server 群の方が，Cache Hit 率が高い傾向にあることがわかった．また，2 つの頻度分布で試した結果，それらによらず，Partial Cache の方が Cache Hit 率が高いこともわかった．

なお，本研究は，科学研究費補助金 No.26330386 の助成を受けて実施している．

参考文献

[1] 市川 類，“コンテンツ配信(CDN)技術と P2P 技術を巡る動向”，ニューヨークだより(IPA)，5 月号，2009.
[2] 小川 愛，“キャッシュ効率の上げ方 その 2”，Akamai Japan ブログ，(January 31, 2013), https://blogs.akamai.com/jp/2013/01/caching02.html, (2017/06/28)
[3] 北野 拓也，”コミュニティ情報を用いたパーシャルキャッシュ型 CDN の研究”，日本大学理工学研究家数学先行修士論文，(March 2017)