大域情報によらない Social Cache 配置ノード決定手法

千 野 雄 貴[†1,†2] 堀 江 光[†1] 河 野 健 二[†1]

1. はじめに

近年, Twitter[1] や Facebook[2] のような Online Social Network (OSN) の需要が高まっている. OSN を使ったコミュニケーションツールは人間関係を形成するのになくてはならない物である. OSN ではユーザのプロフィールといった個人データの管理, 所持をサーバが担っている. そのため, ユーザがそれらのデータを更新する際はサーバ にクエリを送る必要があり, ユーザ自身にそれらのデータの直接的な管理権はない. しかし, Diaspora[3] のような Distributed Online Social Network (DOSN) ではユーザがデータを所持する P2P 構造であるため, ユーザが直接データの管理することができる. これらはユーザ同士がコネクションを利用して互いの持つデータを授受し, 通信を行う. しかし, ユーザがオフラインになるとユーザが個々にデータを所持しているデータにもアクセスできない問題がある.

そこで, SocialButterfly[4] は互いに信頼できる隣接ノードに Social Cache を配置し, 共有データ更新時に予めデータをキャッシュし, そこからデータを取得することでオフライン時でもデータ取得を可能にした. この時の動作を図 1 に示す. この Social Cache の配置先ノード決定手法として SocialCDN[5] がある. SocialCDN はアクセスされない無駄な Social Cache を持つノードを減らすためできるだけ Social Cache を保つノードが少なくなる様に配置する. そのため, そのノードへ負荷が集中する可能性がある. 実際に Facebook グラフに SocialCDN のアルゴリズムで Social Cache を配置 するノードを決定すると, ソーシャルグラフの全ノード数 4039 に対してあるノー ドに 1027 の Social Cache が配置される.

そこで, 一部のノードへの負荷集中を避け,Social Cache 配置ノードを決定するアルゴリズム (Limited Span Elimination Algorithm : LSEA) を提案する.

†1 Keio University
†2 yuuki@sslab.ics.keio.ac.jp

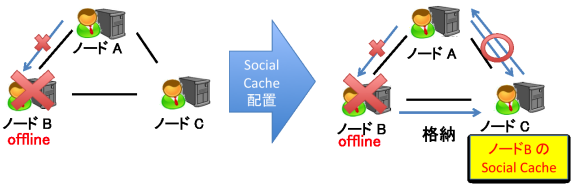

図 1 Social Cache を保持するノードの割合

ノードの Social Cache 保持数に上限を設け, 上限に達していないノードのみを候補に加えることで, 特定のノードへ多数の Social Cache が配置されることを防ぐ. また, 上限を可変にすることで, 全ノードの Social Cache を配置しつつ, 負荷集中を避ける. ここで, 上限を設けたことによりどのノードから Social Cache を配置先を決定するかが重要となる. なぜなら, 隣接ノードの少ないノード程 Social Cache 配置先の候補も少なくなり, これが保持数に影響を与えるからだ. しかし, DOSN では全ユーザの情報を集めることが難しい. その対策として, 大域情報を使用しない 3 つの Social Cache 配置先決定アルゴリズム (Random Selection, Few Adjacent Edge, Fewest Adjacent Edge) を組み合わせることで, DOSN において大域情報を使用せずにノードへの負荷集中を避ける. また, それぞれを 4 つのソーシャルグラフに対し実験を行いどれがより優れているか評価, 分析を行う.

2. SocialCDN

SocialCDN では Social Cache をネットワーク上に効率良く配置することで無駄な Social Cache の配置による 資源の浪費を減らす手法を提案している. Social Cache を隣接ノードに配置することでオフライン時でも Social Cache にアクセスし, データの取得ができる. しかし, Social Cache を無作為に多くのユーザに配置すると, 全くアクセスされない Social Cache を持つノード が生じ, Social Cache は配置されたノードの資源を使用するため, 資源は無駄になる. そこで, Span Elimination Algorithm (SEA) によって配置することでこれらの問題に対応する. SEA はノードの次数を基準に エッジ毎の Social Cache 配置ノード

の候補を選出する. 選出した候補ノードの中から最も選出回数が多いものに Social Cache を配置する.

しかし, SEA を用いた Social Cache 配置では無駄な Social Cache を減らすために Social Cache 保持数に大きな偏りが生まれてしまい, 保持するノードに大きな負荷がかかってしまう.

3. 提案：LSEA

LSEA は Social Cache 保持数に上限を設けて, ノード に負荷が集中せずに Social Cache 配置ノードを決定する. SEA 同様にエッジ数を考慮して候補を選出するが, この際にすでに上限に達しているノードがある場合そのノードを候補選択から外す. 全ノードの Social Cache を配置できるように上限値を可変にし, エッジ数の少ないノードから Social Cache 配置ノードを決定する. なぜなら, エッジ数の少ないノード程 Social Cache 候補になりうるノードの数が少ないため, もし, エッジ数の多いノードが先に Social Cache 配置ノードを決定すると, 上限値が過度に上昇する.

しかし, DOSN はユーザを一括管理するサーバがないため全ユーザのエッジ数を集めることは難しい. そこで, ネットワークの大域的な情報を使用せずにどのノードから Social Cache を配置するかの順序を決める 3 つのアルゴリズム Random Selection (RS) , Few Adjacent Edge (FAE) , FewEst Adjacent Edge (FEAE) を挙げる.

RS はランダムに決める. FAE はランダムに選択されたノードの周囲でエッジ数の小さい順に順序に加える. そして, FEAE はまず, 同様にランダムにノードを選択し, 隣接ノードの中からエッジ数が最小のノードの配置を行う. 配置を行った ノードの隣接ノードの中から再度エッジ数が最小のノー ドが配置を行う. これを繰り返し, もし隣接ノード全てが Social Cache 配置を終えた場合再びランダムにノードを選択し繰り返す. FEAE はノードの周囲の中で最小エッジ数のノードを常に選ぶ. これらの手法は全ノードの情報が必要といった大域的な情報を使用しないため DOSN で実装可能である.

4. 実　　験

4 つのソーシャルグラフ[6] (Random, Facebook, Enron e-mail[7], Amazon[8]) に対し, LSEA と大域情報を使用しない 3 つの順序決めアルゴリズム (RS, FAE, FEAE) を組み合わせた手法を用いて, Social Cache 配置をシミュレートした. また, 比較対象として, 既存手法 (SEA) と大域情報を使用した場合の

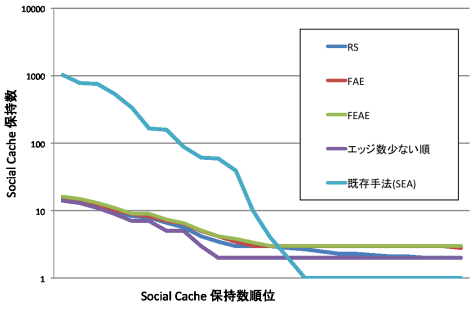

図 2　それぞれの手法による Social Cache 保持数：Facebook

LSEA も同様に実験した. その結果, 図 2 より Facebook において大域情報を使用しない 3 つの手法の内 RS が最も Social Cache 保持数を低く抑えることができ, 負荷が集中しなかった. RS での最大保持数は全ノード数 4039 に対し 約 15 であり, 大域情報を使用した場合とも約 7 % しか差がなかった.

5. ま と め

ノードへの負荷集中を防ぐため Social Cache 保持数に上限を設ける手法 LSEA を提案し, ネットワークの大域情報を使用しない順序決定アルゴリズムと組み合わせることで, DOSN で実装可能にした. Facebook ソーシャルグラフにおいて LSEA と順序決定アルゴリズムを組み合わせ, シミュレートした結果, RS が最も保持数を抑えることができ, 既存手法に比べ平均で約 50 % 減らすことができた.

参 考 文 献

1) Twitter. http://www.twitter.com/.
2) Facebook. http://www.facebook.com/.
3) Diaspora. https://joindiaspora.com/.
4) Lu Han, Badri Nath, Liviu Iftode, and S.Muthukrishnan. Social butterfly: Social caches for distributed social networks. In *IEEE International Conference on Social Computing 2011*, pages 81–86, October 2011.
5) Lu Han, Magdalena Punceva, Badri Nath, S.Muthukrishman, and Liviu Iftode. Social-cdn : Caching techniques for distributed social networks. In *IEEE International Conference on Peer-to-Peer Computing*, pages 191–202, September 2012.
6) STANFORD UNIVERSITY. http://snap.stanford.edu/data/.
7) Enron. http://www.enron.com/.
8) Amazon. http://www.amazon.com/.