

A-016

確率的弱コーラムシステムを用いた P2P オブジェクト検索アルゴリズムの実験的評価

An Evaluation of an Object Search Algorithm for P2P Environment with Probabilistic Weak Quorum Systems

田川 太郎[†]

Taro Tagawa

三浦 健[‡]

Miura Ken

角川 裕次[†]

Kakugawa Hirotsugu

1. はじめに

互いに通信しながら自律的に協調動作するプロセスの集合を分散システムという。近年、分散システムの一つである P2P(peer-to-peer) によるファイル共有が注目されている。本稿ではそれに関連した問題であるネットワーク上に分散しているオブジェクトの検索問題を取り上げる。P2P やアドホックネットワークなどの動的なネットワークでは、参加プロセスやオブジェクトなど分散システムの情報を完全に把握することは困難である。本研究では、分散システムでの情報通信構造の一つであるコーラムに基づく情報通信を考える。コートリ [1] は分散相互排除やマッチメイキングに広く用いられているが、コーラムを形成するのに全てのプロセス情報が必要であり動的なネットワーク上で用いるのは難しい。本研究では一部のプロセス情報でコーラム形成可能な確率的弱コーラムシステムを用いたオブジェクト検索アルゴリズムをシミュレーションにより評価を行い、大規模 P2P システムへの応用可能性を示す。

2. 定義

2.1 オブジェクト検索問題

本研究で想定するオブジェクトには、同一内容の複製が存在し、ネットワーク上の全プロセスに対するその複製を持つプロセス数の割合を定数 r とする。オブジェクト検索問題とはあるプロセスにおいてオブジェクト o に対する要求が発生したとき、 o もしくはその複製を持っているプロセスの内の一つを知る問題と定義する。オブジェクトの複製を持つプロセスを知ることが出来れば、そのプロセスに対しオブジェクトの要求を行うことが出来る。

2.2 確率的弱コーラムシステム (PWQS)

定義 1. (確率的弱コーラムシステム PWQS[3]): $U = \{p_1, \dots, p_n\}$ をプロセス集合とし、 U の部分集合の集合をセットシステム Q と呼ぶ。各プロセス p_i は確率分布 w_i に従いコーラム $Q_i \in Q$ を選択するとし、そのベクトル $\bar{w} = (w_1, w_2, \dots, w_n)$ をアクセス戦術ベクトルとする。 U の任意のプロセス p_x と任意の部分プロセス集合 $P_k = \{p_{y_1}, \dots, p_{y_k}\}$ について、 p_x が選択するコーラムが P_k の各プロセスが選択するコーラムの少なくとも一つと共通部分を持つ確率が少なくとも $1 - \epsilon$ 、つまり、

$$\sum_{\substack{Q_k = Q_{y_1} \cap \dots \cap Q_{y_k}, \\ Q_x \cap Q_k \neq \emptyset}} w_x(Q_x) \prod_{j=1}^k w_{y_j}(Q_{y_j}) \geq 1 - \epsilon$$

[†] 広島大学大学院工学研究科情報工学専攻

[‡] 株式会社沖データ

を満たすならば $(Q, \bar{w}, \epsilon, k)$ は位数 k の確率的弱コーラムシステム (k -PWQS) である。□

オブジェクト検索問題に k -PWQS を用いる手法を説明する。ネットワーク中でオブジェクトを保持しているプロセスの数を k 、つまり $k = r \cdot |U|$ とする。まずオブジェクトの複製を持つプロセスは予めランダムに選んだコーラムのメンバに複製を保持していることを伝える。オブジェクトを検索したいプロセスはランダムに選んだコーラムのメンバに問い合わせをする。そのとき、そのコーラムが複製を保持した k 個のプロセスが選んだコーラムのうち少なくとも一つと共通部分を持てばオブジェクトの複製を持つプロセスを知ることができる。 k -PWQS の性質より共通部分を持つ確率が $1 - \epsilon$ で保証されているので少なくとも $1 - \epsilon$ の確率でオブジェクト検索は成功する。

2.3 モデル

システムモデルとして次のようなものを仮定している。

- プロセス集合 $U = \{p_1, p_2, \dots, p_n\}$ とそれらを繋ぐ通信リンクにより構成
- 各プロセスは論理的に完全結合
- メッセージ交換により互いに通信を行う
- 各プロセス p_i は初期状態では U の部分集合 $N_0(p_i) \subseteq U$ のネットワークアドレスのみを知っている
- メッセージ送信は、ネットワークアドレスを知っているプロセスに対してのみ可能である
- 参加、離脱が可能であり U は時刻により変化する

3. アルゴリズム

本研究で評価を行ったオブジェクト検索アルゴリズムは文献 [3] で提案されたものである。

プロセス数に対するオブジェクトの複製の存在する割合 r とオブジェクト検索で正しく複製をもつプロセスを知ることができる確率 ρ を設定し、プロセスの把握、コーラム形成、オブジェクト検索を行う。

以下にアルゴリズムの概要を示す。

- **プロセス情報の収集** コーラム形成に必要な数のプロセス情報 (ネットワークアドレス) を集める。ここで収集には Name-Dropper アルゴリズム [2] を改良したものを用いる。Name-Dropper アルゴリズムが全てのプロセス情報を集めるのに対し、このアルゴリズムで用いる PWQS は r と ρ によって定まるパラメータ R 個のプロセスのネットワークアドレスを集めるだけでコーラムは形成できるので R 個以上になると情報の収集を終了する。

- **コラムの形成** 情報を得たプロセスの中からコラム集合を決定する。把握しているプロセスの中からランダムに大きさ R のコラムを作成する。オブジェクトの複製を持つプロセスはコラム集合の各プロセスに自身がオブジェクトの複製を持っていることを伝える。

- **オブジェクトの検索** オブジェクトの複製を持たないプロセスはコラム集合からコラムを選択しオブジェクト検索を行う。コラムのメンバーのうち少なくとも一つのプロセスがオブジェクトの複製を持つプロセスを知っていれば、オブジェクト検索は成功する。

コラム形成までにかかる通信複雑度は $O(\log n \log R + N_{\max})$ である。ここで $N_{\max} = \max_{p_i \in U} \{ |N(p_i)| \}$ である。 $N(p_i)$ というのはプロセス p_i の把握しているプロセスの集合のことで、通信複雑度を下げるには R 個以上把握したプロセスは $N(p_i)$ を増やさないようにすればよい。そのため R 個把握したプロセスは情報の収集を行わないようになっている。

4. シミュレーションによる評価

上述のアルゴリズムのシミュレータを作成し、評価した。ここで開発環境は、OS: Linux kernel 2.4、言語: C++、コンパイラ: gcc 2.96、である。オブジェクトの検索はコラムを形成してしまえばすぐに行うことができるので、コラムを形成するために必要なプロセス情報の収集の部分での評価を主に行った。各プロセスがコラムを形成するのに必要な R 個プロセスを把握するまでプロセス情報の収集を行い、そのときの $N(p_i)$ の平均を調べ Name-Dropper アルゴリズムと比較した。ここでシミュレーションは初期グラフをランダムな木、参加、離脱がないもとで試行回数 1000 回で行った。

図 1 の結果から、Name-Dropper アルゴリズムではプロセス数の増加にともない各プロセスが R 個のプロセス情報を収集したときの $N(p_i)$ の平均が大幅に増えている。これは Name-Dropper アルゴリズムが全てのプロセスを知るまで情報の収集を行うからだといえる。反対に評価アルゴリズムでは $N(p_i)$ の平均数が一定以上にならない結果となった。 $N(p_i)$ の理論的な最悪値は n だが、図 1 からノード数の増加に伴い $2R$ あたりに収束している。結果から評価アルゴリズムは比較的通信複雑度を抑えながら目標数プロセス収集が行えているといえる。

そして図 2 の結果から、評価アルゴリズムは Name-Dropper アルゴリズムより短い時間でプロセス収集が行えている。これは評価アルゴリズムでは R 個把握したプロセス p_i はプロセス情報収集を行わないが他のプロセス p_j からまだ情報収集中のメッセージ $N(p_j)$ を受け取ると p_i は p_j へ $N(p_i)$ を送るので、 p_j もただちに R 個以上把握することができるためであると考えられる。

5. おわりに

本稿では“確率的弱コラムシステムを用いたオブジェクト検索アルゴリズム”におけるプロセス情報の収集の部分 Name-Dropper アルゴリズムと比較した。シミュレーションにより、大規模な P2P システムにおいても

効率よく動作することが示された。今後の課題としては今回取り上げなかったオブジェクト検索部分についての考察などが挙げられる。

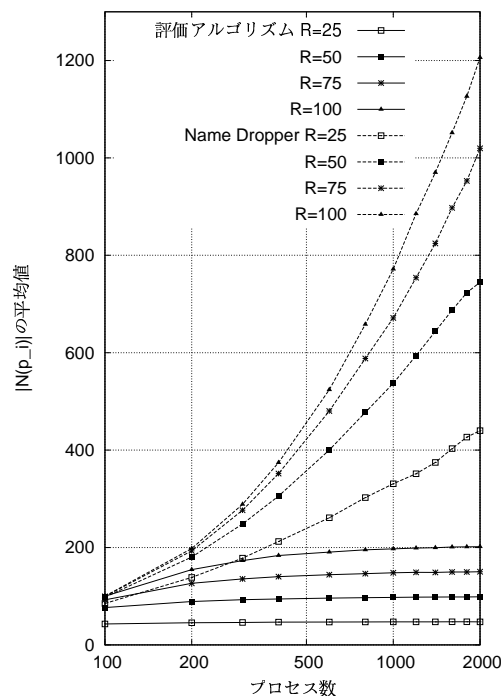


図 1: 全てのプロセスが情報収集終了時点での各プロセスの把握プロセス数の平均

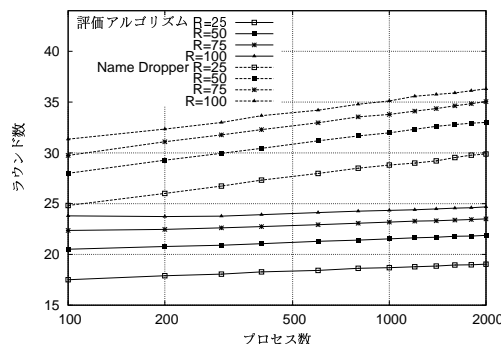


図 2: 全てのプロセスが情報収集を終了するまでにかかったラウンド数

参考文献

- [1] H.Garcia-Molina, D.Barbara, “How to assign votes in a distributed system”, J.ACM 32(4), 1985
- [2] M.Harchol-Baalter, T.Leighton, D.Lewin, “Resource Discovery in Distributed Networks”, In PODC, 1999.
- [3] 三浦健, 角川裕次, “確率的弱コラムシステムを用いた P2P 環境オブジェクト検索アルゴリズム”, 情報処理学会, 2004-AL-93, pp. 49-56, 2004 年 1 月.