

アクセス要求発生に時間間隔をもつ相関データの放送スケジューリングについて

6F-1

内田 渉[†] 原 隆浩[†] 塚本昌彦[†] 矢島 悦子[‡] 西尾章治郎[†]
[†]大阪大学工学部情報システム工学教室 [‡](株)エフエム大阪 東京支社営業部

1 はじめに

近年、有線および無線の通信技術の発展に伴い、放送通信を用いたプッシュ型配送に関する研究の関心が高まっている。プッシュ型配送に基づく情報システムでは、サーバが周期的に様々なデータアイテムを配送する[1]。クライアントは、放送されているアイテムに対してアクセス要求を発生し、そのアイテムが放送された時点で、アクセスが完了する。プッシュ型の情報システムでは、クライアント数が増加してもシステム全体の負担コストはほとんど変化しないため、クライアント数が非常に多い分散データベースシステムにおいて、アクセスに対する応答時間の短縮が期待できる。

プッシュ型情報システムの性能を向上するために、様々な放送プログラムのスケジューリング戦略が提案されている[2, 3, 4]。これまでに筆者らは、クライアントが相関性をもつ2つのアイテムに対して時間間隔をおいてアクセス要求を発生する環境を想定して、2つのアイテムの放送時間間隔の最適値を計算し、その最適値を用いて放送プログラムのスケジューリングを行う手法を提案した[4]。しかし、この手法の性能評価の結果、プログラムをランダムに作成する場合に比べて、2%程度しか平均応答時間を短縮できないことを確認している。そこで本稿では、この手法を改良したスケジューリング方式を提案し、その性能評価の結果を示す。

2 スケジューリングアルゴリズム

本章では、まず文献[4]で提案したUFLアルゴリズムについて説明し、その後、改良したスケジューリングアルゴリズムを提案する。以下では、次のようなシステム環境を想定する。

- サーバは一つとする。
- 各アイテムの放送にかかる時間は全て等しい。また、サーバは M 個のアイテムを1周期で1度ずつ放送する。
- クライアントはキャッシュをもたない。

2.1 UFL アルゴリズム

一般に、アイテム i に対するアクセス要求が発生した後、アイテム j のアクセス要求が発生するまでの時間間隔は、確率密度関数 $F_{i,j}(t)$ で表される。ここで i の放送時刻を $t=0$ とし、アイテム j の放送時刻を τ としたとき、アイテム j へのアクセス要求に対する平均応答時間 $avg_{i,j}$ は次式で表される。

$$avg_{i,j} = \int_0^{\tau} F_{i,j}(t) \cdot (\tau - t) dt + \int_{\tau}^T F_{i,j}(t) \cdot \{(T + \tau) - t\} dt$$

ここで、アイテム i から j への連続したアクセス要求の発生する確率を $P_{i,j}(\sum_{j=1}^M \sum_{i=1}^M P_{i,j} = 1)$ とすると、クライアントのアクセス要求の平均応答時間は、次式で表される。

$$\sum_{j=1}^M \sum_{i=1}^M P_{i,j} \cdot avg_{i,j} \quad (1)$$

これを最小にするプログラムを発見するために、可能な全ての順列を調べると、計算量が膨大となる。そこで文献[4]では、ヒューリスティックにスケジューリングを行うUFLアルゴリズムを提案した。このアルゴリズムでは、すべてのアイテムの組 (i, j) に、次の手順を繰り返し実行して、プログラムを表す次元の離散的座標系に各アイテムを配置する。

On Broadcast Scheduling of Correlative Data with Access Interval
Wataru UCHIDA[†], Takahiro HARA[†], Masahiko TSUKAMOTO[†],
Etsuko YAJIMA[‡], and Shojiro NISHIO[†].

[†]Dep. of Info. Sys. Eng., Faculty of Engineering, Osaka Univ.

[‡]Sales Dept., Tokyo Office, FM Osaka Co., Ltd.

- $P_{i,j}$ の値が大きい順に、アイテムの組 (i, j) を選択する。
- それらを、最適放送間隔 $U_{i,j}$ あけて座標系に配置する。
- アイテムを配置すべき座標に既に他のアイテムが配置されている場合は、その前後で空いている場所に配置する。

2.2 改良アルゴリズム

UFLは、プログラムをランダムに配置する場合と比較して2%程度しか応答時間を短縮できない。これは、適当な放送時間間隔で配置されているアイテムの組み合わせの割合が非常に小さいことが原因と考えられる。そこで、UFLで決定したプログラムから、理論上の平均応答時間が小さくなるようにアイテムの入れ換えを行う、RACアルゴリズムを提案する。アルゴリズムの手順を以下に示す。

1. UFLで決定したプログラムをアイテムの初期順列とする。
2. 初期順列に対して、先頭のアイテムと各アイテムを入れ換えたものを候補順列の集合 G とする。このとき、 G の要素数が候補順列要素数 k を越えるとき、平均応答時間(理論値)の小さい上位 k 個以外は破棄する。
3. G から平均応答時間が最小となる順列を消去する。その順列に対して、再びデータアイテムの入れ換えを行い、入れ換え前よりも平均応答時間を小さくなるものを新たに G に加える。 G の要素の数をステップ2と同様に制限する。
4. ステップ3を繰り返し行い、 G から順列がなくなった時点でアルゴリズムを終了する。この過程で、平均応答時間が最も小さい順列を最終的な結果とする。

3 性能評価

本章では、提案した改良アルゴリズムの性能評価を行う。評価に用いた想定環境は次の通りである。

- 放送されるアイテム数は120とし、1アイテムの放送にかかる時間は10単位時間とする。
- アイテム i と j に連続してアクセスが起こる確率は 120×120 の正方行列の i - j 要素で与えられる。行列の各要素のうち、ある割合の要素のみ正の定数値をもつ。

3.1 UFL と RAC の性能比較

まず、RACとUFLおよびランダムな配置方法の性能比較を行う。アクセス行列の正の値をもつ要素の割合を変化させて、各アルゴリズムによって作成したプログラムの平均応答時間を計算する。アクセス要求の到着時間間隔の確率密度関数 $F_{i,j}(t)$ は次の3つを用いる。

1. $F_{i,j}(t) = q$ ($0 \leq t \leq 1/q$)。 q の値はランダムに決定する。
2. $F_{i,j}(t) = \begin{cases} 10^{-4}(t-m) + 10^{-2} & (m-10^2 \leq t \leq m) \\ 10^{-4}(t-m) - 10^{-2} & (m \leq t \leq m+10^2) \end{cases}$
これは $(0, m-10^2)$, $(m, 10^{-2})$, $(0, m+10^2)$ を結ぶ突起形となる。 m の値はランダムに決定する。
3. $F_{i,j}(t) = 10^{-2} e^{-10^{-2}(t-10)}$ ($10 \leq t \leq 1200$)。これは平均値100のポアソン分布である。

評価結果を図1、図2、図3に示す。各図では、横軸がアクセス行列で正の値をもつ要素の割合、縦軸が平均応答時間を表している。SCはプログラムをランダムに配置する方法の結果を表している。RACにおいて、 k の値は10とした。この結果から、RACはUFLと比較すると、平均して10%程度平均応答時間を短縮していることがわかる。また、正の値をもつ要素の割合が小さいほど、RACとUFLの性能差が大きくなっていく。

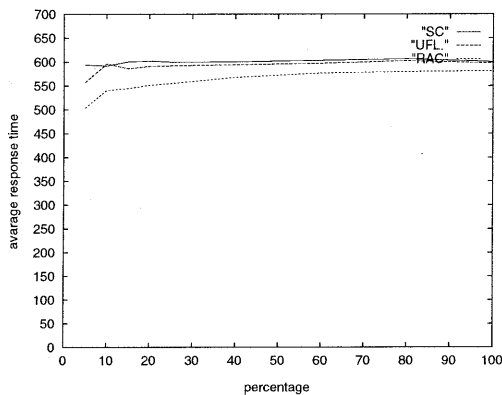


図 1: $F_{i,j}(t) = q$ の結果

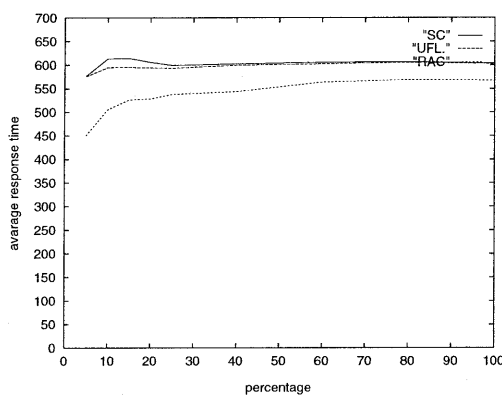


図 2: $F_{i,j}(t) = 10^{-4}(t - m) + / - 10^{-2}$ の結果

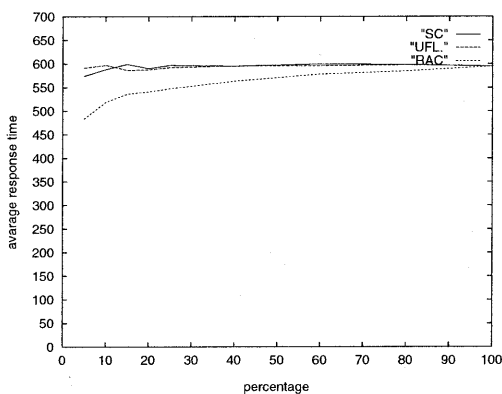


図 3: $F_{i,j}(t) = 10^{-2}e^{-10^{-2}(t-10)}$ の結果

3.2 候補順位要素数 k の影響

RAC アルゴリズムにおいて、候補順位要素数 k の値を大きくすることによって、より多くの候補順位を調べることが可能となるため、性能が向上するものと考えられる。そこで、 k の値を 5 から 25 まで変化させて、RAC の平均応答時間を計算する。アクセス行列の正の要素の割合は 10% とし、 $F_{i,j}(t)$ には 2 番目のものを用いる。また、 k の値を大きくすると計算量も大きくなり、RAC アルゴリズムが実時間で終了しない場合が考えられるため、次のような RAC の計算量削減のための変更版を考案する。

1. RAC のステップ 2 において、アイテムの入れ換えを行って作成した順列のうち、平均応答時間が最小となる順列のみを候補順位とする。

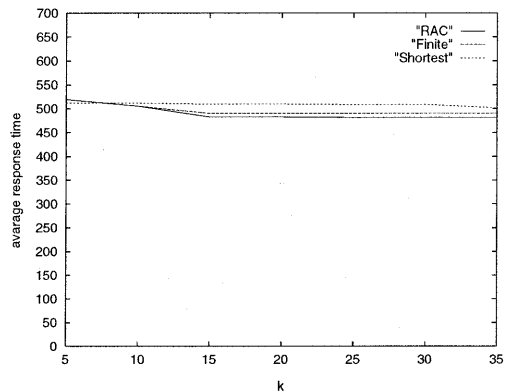


図 4: RAC の改良型の性能比較

2. ステップ 4 において、平均応答時間の最小値がある繰り返し回数 (ここでは 100 回) 更新されなかった場合、 G が空ではなくてもアルゴリズムを終了する。

本来の RAC およびこの 2 つの変更版によって作成したプログラムの平均応答時間を計算した結果を図 4 に示す。図 4 では、横軸が k 、縦軸が平均応答時間を表している。RAC が本来の RAC、Shortest が変更版 1、Finite が変更版 2 を示している。この結果から、本来の RAC は k の値が大きいくとき、3 方式のなかで最も良い性能を示す。また、ある程度 k の値が大きくなると、ほとんど性能が変化しなくなる。これは、 k の値がある程度大きくなると、ほぼ最適配置が発見されるためであると考えられる。変更版 2 は、 k の値にほとんど影響されないため、 k の値が大きいくときには、3 方式のなかで最も性能が悪くなる。変更版 2 は、 k の値が小さいときは、本来の RAC とほぼ同じ性能を示すが、 k の値が大きくなると、途中でアルゴリズムを終了するため、RAC より性能が悪くなる。ステップ 4 の繰り返し回数を調べた結果、 $k = 30$ のとき本来の RAC が 8557 回であったのに対し、変更版 2 は 6111 回で終了している。

4 おわりに

本稿では、放送型情報システムにおけるサーバ側のスケジューリング戦略として、UFL アルゴリズムの配置からアイテムの入れ換えを行う RAC アルゴリズムを提案した。性能評価の結果、UFL と比較して RAC は大幅に性能が向上することを確認した。また、RAC の変更版アルゴリズムを考案し、それらの性能を比較した。

今後は、アイテムが 1 周期中に複数回放送されるような場合のスケジューリング方式について検討する予定である。

謝辞 本研究は、日本学術振興会基盤研究 (B)(2) (12480095) および日本学術振興会未来開拓学術研究推進事業における研究プロジェクト「マルチメディア・コンテンツの高次処理の研究」(Project No. JSPS-RFTF97P00501) の研究助成によるものである。ここに記して謝意を表す。

参考文献

- [1] Acharya, S., et al., "Broadcast Disks: Data Management for Asymmetric Communication Environments," *Proc. ACM SIGMOD*, pp. 199-210 (1995).
- [2] Hameed, S. and Vaidya, N.H., "Log-time Algorithms for Scheduling Single and Multiple Channel Data Broadcast," *Proc. ACM MOBICOM*, pp. 90-99 (1997).
- [3] Su, C.J., et al., "Broadcast Scheduling for Information Distribution," *Wireless Networks*, Vol.5, No.2, pp. 137-147 (1999).
- [4] 矢島 悦子 ほか, "相関性を持つデータ間の放送時間間隔について," 情処論, Vol.40, No.1, pp.188-196 (Jan. 1999).
- [5] 矢島 悦子 ほか, "データ間の相関性を考慮した放送データのスケジューリング法およびキャッシング法," 情処論, Vol.40, No.9, pp.3577-3585 (Sep. 1999).