2002年日本オペレーションズ・リサーチ学会
秋季研究発表会

# 周期観測下での最適負荷分散政策

名古屋工業大学 *井家 敦
名古屋工業大学 01002633 大野 勝久

## 1. はじめに

分散サーバシステムにおけるサーバ間負荷分散制御がレスポンスの改善に有効な方法である．負荷分散は幅広い意味を持つが，本論文ではクライアントからのファイル配信等のリクエストをシステム全体のレスポンスを向上できるよう，各サーバに効率よく送信する処理と定義する．

Eager et al.[1] はシステムの現在の情報を利用した負荷分散が利用しない負荷分散と比較してよりよいパフォーマンスを与えることを示した．しかしながら，実際の大規模なシステムで，リクエストが到着するたびに現在の情報を収集することは非常に困難である [2]．よって，情報を収集するタイミングを決め，その情報を負荷分散に有効活用することが要求される．Shivaratri et al.[3] は情報を収集するタイミングの決め方を *Information Policy* と定義した．Information Policy を考慮した最近の研究として，[2] や [4] などがある．本研究では，その中でも特に周期的に情報を収集する方法に重点をおき，情報を収集する時間区間を単位期間とおくことにする．

本研究では，図 1 に示すような，1 つのスイッチに $n$ 個のサーバが接続されたシステムを考え，このシステムをマルコフ決定過程 (Markov decision process，以下 MDP) として定式化し，時間平均コストを最小化する最適負荷分散政策を求める．

## 2. MDP による定式化

MDP は，各状態での決定によって，マルコフ的に遷移するシステムにおける時間平均利得を最小化する政策を導く．

スイッチの番号を 0 とし，各サーバの番号を $i, i = 1, \ldots, n$ とする．そして，スイッチとサーバ $i$ のバッファ容量をそれぞれ $B_0, B_i$ とする．クライアントからのリクエストがスイッチに到着し，そのバッファに格納される．ただし，バッファを超える到着はリジェクトされる．スイッチでは，負荷分散政策に従いサーバ間にリクエストが送信される．負荷分散政策はスイッチ内のリクエスト数 $x_0$，サーバ $i$ 内のリクエスト数 $x_i, i = 1, \ldots, n$ を情報として利用し決定される．(ただし，$x_i$ はサーバ $i$ で処理中のリクエストを含むものとする.) システムの情報の収集は各期間の最初に行なわれる．

負荷分散政策はその期間でのサーバ $i$ へのリクエスト転送数 $a_i, i = 1, \ldots, n$ を与える．$a_i$ の制約として，サーバ $i$ のバッファ容量を超える転送を許さないものとし，負荷分散処理に要する時間は無視できるものとする．

単位期間あたりのシステムへのリクエストの到着数を $A$，サーバ $i$ のリクエスト処理数を $S_i$ とする．$A$, $S_i$ はそれぞれ独立な確率変数であり，おのおのの分布は，$\alpha(l) = \Pr\{A = l\}, l = 0, 1, \ldots, A_{MAX}$，$\beta_i(l) = \Pr\{S_i = l\}, l = 0, 1, \ldots, S_{iMAX}$ $i = 1, \ldots, n$ で与えられる．ここで，$A_{MAX}, S_{iMAX}$ はそれぞれ到着数，サーバ $i$ の処理数の最大値である．

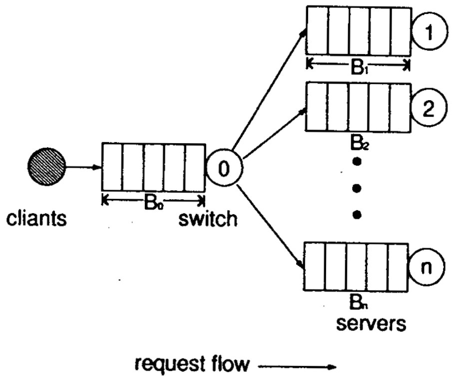


図 1: サーバ負荷分散システム

### 2.1. 状態空間

システムの状態を $\boldsymbol{x} = (x_0, x_1, \ldots, x_n)$，状態空間を $\boldsymbol{X}$ とおく．ここで $0 \le x_i \le B_i, i = 0, \ldots, n$ である．

### 2.2. 各状態での決定

状態 $\boldsymbol{x}$ での決定を $f(\boldsymbol{x}) = (a_1, \ldots, a_n)$ で表現する．$a_i$ はそれぞれ

$$\sum_{i=1}^{n} a_i \le x_0$$
$$a_i + x_i \le B_i \quad i = 1, \ldots, n$$

を満たす．状態 $\boldsymbol{x}$ でとりうる決定の集合を $K(\boldsymbol{x})$ とおき，政策 $\boldsymbol{\pi}$ を各状態での決定の集合，すなわち

$$\boldsymbol{\pi} = \{f(\boldsymbol{x}) | \boldsymbol{x} \in \boldsymbol{X}\} \tag{1}$$

で与える．

### 2.3. 遷移確率

状態 $\boldsymbol{x}$ で決定 $f$ が与えられたとき，次の時点での状態 $\boldsymbol{x}' = (x'_0, x'_1, \ldots, x'_n)$ は

$$x'_0 = B_0 - [B_0 - x_0 + \sum_{i=1}^{n} a_i - A]^+ \tag{2}$$

$$x'_i = [x_i + a_i - S_i]^+, i = 1, \ldots, n \tag{3}$$

で与えられる．ここで，$[a]^+ = \max\{a, 0\}$ である．その遷移確率 $p(\boldsymbol{x}, \boldsymbol{x}'; f)$，$\boldsymbol{x}, \boldsymbol{x}' \in \boldsymbol{X}, f \in K(\boldsymbol{x})$ を，状態 $x_i$ から $x'_i$ への遷移確率 $p_i(x_i, x'_i; f), i = 0, 1, \ldots, n$ の積，すなわち

$$p(\boldsymbol{x}, \boldsymbol{x}'; f) = \prod_{i=0}^{n} p_i(x_i, x'_i; f) \tag{4}$$

で表現する．ここで，$p_i(x_i, x'_i; f), i = 0, \ldots, n$ は

$$p_0(x_0, x'_0; f) = \begin{cases} \alpha(x'_0 - (x_0 - \sum_{i=1}^{n} a_i)), \\ \quad \text{for } 0 \le x'_0 < B_0 \text{ and} \\ \quad\quad 0 \le x'_0 - (x_0 - \sum_{i=1}^{n} a_i) \le A_{MAX} \\ \sum_{l=B_0-(x_0-\sum_{i=1}^{n} a_i)}^{A_{MAX}} \alpha(l), \\ \quad \text{for } x'_0 = B_0 \text{ and} \\ \quad\quad B_0 - (x_0 - \sum_{i=1}^{n} a_i) \le A_{MAX} \\ 0, \quad \text{otherwise} \end{cases} \tag{5}$$

$$p_i(x_i, x'_i; f) = \begin{cases} \beta_i((x_i + a_i) - x'_i), \\ \quad \text{for } 0 < x'_i \le B_i \text{ and} \\ \quad\quad 0 \le (x_i + a_i) - x'_i \le S_{iMAX} \\ \sum_{l=x_i+a_i}^{S_{iMAX}} \beta_i(l), \\ \quad \text{for } x'_i = 0 \text{ and } (x_i + a_i) \le S_{iMAX} \\ 0, \quad \text{otherwise} \end{cases} \tag{6}$$

$i = 1, \ldots, n$

で与えられる．

### 2.4. コスト関数

1期間あたりの1リクエストの滞留コスト，リジェクトコストをそれぞれ $C_{hld}, C_{rej}$ とすると，状態 $\boldsymbol{x}$ で決定 $f$ が与えられ，次の期間の最初の状態が $\boldsymbol{x}'$ になった場合に被るコスト $r(\boldsymbol{x}, \boldsymbol{x}'; f)$ は

$$\begin{aligned} r(\boldsymbol{x}, \boldsymbol{x}'; f) &= C_{hld} \sum_{i=0}^{n} x'_i \\ &+ I_{\{x'_0 = B_0\}} C_{rej} \sum_{l=u}^{A_{MAX}} (l - u)\alpha(l), \\ &\qquad \boldsymbol{x}, \boldsymbol{x}' \in \boldsymbol{X},\ f \in K(\boldsymbol{x}) \end{aligned} \tag{7}$$

で表現される．ここで，$u = B_0 - (x_0 - \sum_{i=1}^{n} a_i)$ および $I_{\{x'_0 = B_0\}}$ は，$x'_0 = B_0$ のとき1，さもなければ0を与える指示関数である．また，$r(\boldsymbol{x}; f)$ は

$$r(\boldsymbol{x}; f) = \sum_{\boldsymbol{x}' \in \boldsymbol{X}} r(\boldsymbol{x}, \boldsymbol{x}'; f) p(\boldsymbol{x}, \boldsymbol{x}'; f) \tag{8}$$

で与えられる．

### 2.5. 最適性方程式

各時点での平均コスト，状態 $\boldsymbol{x}$ での相対コストをそれぞれ $g$，$h(\boldsymbol{x})$ とすると，次の最適性方程式

$$\begin{aligned} g + h(\boldsymbol{x}) = \min_{f(\boldsymbol{x}) \in K(\boldsymbol{x})} \{ & r(\boldsymbol{x}, f(\boldsymbol{x})) \\ & + \sum_{\boldsymbol{x}' \in \boldsymbol{X}} p(\boldsymbol{x}, \boldsymbol{x}'; f(\boldsymbol{x})) h(\boldsymbol{x}') \}, \quad \boldsymbol{x} \in \boldsymbol{X} \end{aligned} \tag{9}$$

を満たす政策 $\boldsymbol{\pi}^*$ が最適負荷分散政策である．

## 3. 数値例

$n = 3, A_{MAX} = 10, \lambda = 0.8, B_i = 10, i = 0, 1, \ldots, n,$ $\mu_i = 0.4, S_{iMAX} = 10, i = 1, \ldots, n, C_{hld} = 3, C_{rej} = 10$ とし，$A, S_i$ がそれぞれ $B(A_{MAX}, \lambda), B(S_{iMAX}, \mu_i)$ に従うものとする．ここで，$X$ が $B(n, p)$ に従うとは

$$\Pr\{X = l\} = \binom{n}{l} p^l (1-p)^{n-l} \tag{10}$$

のことである．この問題に対して修正政策反復法 [5] を用いたところ，最小平均コストは 26.152 であった．

## 4. 今後の課題

前節の問題は，状態数が 1000 程度のものであるが，実際にはさらに大規模な問題を解くことが求められる．よって Neuro-DP[6] あるいは強化学習 [7] に基づくアルゴリズムを用いた近似政策の評価を今後の課題としたい．

## 参考文献


[1] D. Eager, E. Lazowska, J. Zahorjan, Adaptive load sharing in homogeneous distributed systems, IEEE Transactions on Software Engineering vol. 12, no. 5, pp. 662-675, May 1986.

[2] M. Dahlin, Interpreting stale load information, IEEE Transactions on Parallel and Distributed Systems vol. 11, no. 10, pp. 1033-1047, Oct 2000.

[3] N. G. Shivaratri, P. Krueger, M. Shinghal, Load distributing for locally distributed systems, Computer, pp. 33-43, Dec 1992.

[4] M. Mitzenmacher, How useful is old information, IEEE Transactions on Parallel and Distributed Systems vol. 11, no. 1, pp.6-20, Jan 2000.

[5] K. Ohno, Modified policy iteration algorithm with nonoptimality tests for undiscounted Markov decision process, Working Paper, Dept. of Information System and Management Science, Konan University, Japan, 1985.

[6] D. P. Bertsekas, J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, 1996.

[7] R. S. Sutton, A. G. Barto, *Reinforcement Learning*, MIT Press, 1998.