

分散型ディスパッチャを用いたジョブ配送法について

7D-3

染葉佳代子 渡辺尚 太田剛 水野忠則

静岡大学 工学部

1 はじめに

分散システムにおいては、資源の有効利用と応答特性の改善のために動的負荷分散が必須である。これまでに、ジョブの配送先の決定方法について多くの研究が行われてきた。ジョブの配送先を決定するディスパッチャの形態は、一つのディスパッチャですべてのジョブを扱う集中型と、各計算機ごとにディスパッチャを持つ分散型に分けられる。集中型による研究は、BanawanとZahhorjan[1]やBonomiとKumar[2]らによって行なわれている。本研究では、ユーザの要求への柔軟な対処、大きなネットワークへの対応といった点から分散型ディスパッチャについて考える。そして、集中型ディスパッチャにおける有効なジョブ配送方式としてShenkerとWeinrib[3]により提案されたD方式を分散型ディスパッチャに適用した配送法を提案し、シミュレーションにより検討する。

2 モデルと配送アルゴリズム

2.1 モデル

本研究で対象とするシステムを図1に示す。

- ・  $n$  個のディスパッチャと  $n$  個のサーバを考える。サーバ  $S_i$  をディスパッチャ  $D_i$  のローカルサーバ、 $S_j$  を  $D_i$  ( $i \neq j$ ) のリモートサーバと呼ぶ。
- ・  $S_i$  の処理速度を  $c_i$  命令/秒、 $\vec{C} = (c_1, c_2, \dots, c_n)$  とする。
- ・ ディスパッチ時間  $0$  秒とする。
- ・ 通信遅延はローカル間を  $0$  秒、リモート間を一定値  $d$  秒とする。
- ・  $D_i$  にはジョブが平均  $\lambda_i$  個/秒でポアソン到着し待ち行列を形成する。
- ・ ジョブの処理要求量は平均  $\frac{1}{\mu}$  命令の指数分布に従う。ただし、本稿では  $\mu = 1$  とする。
- ・ サーバに割り当てられたジョブはそのサーバごとに待ち行列を形成し、先着順に処理される。
- ・ ローカルサーバの負荷状況は瞬時にわかる。

2.2 配送アルゴリズム

D方式

ShenkerとWeinrib [3]により提案されたD (deterministic)方式は、有限個の速いサーバと無限個の遅いサーバ

Load balancing policies with distributed dispatchers developed from a centralized schedule strategy  
 Kayoko Someha, Takashi Watanabe,  
 Tsuyoshi Ohta, Tadanori Mizuno,  
 Faculty of engineering, Shizuoka University  
 3-5-1, Johoku, Hamamatsu, 432 Japan

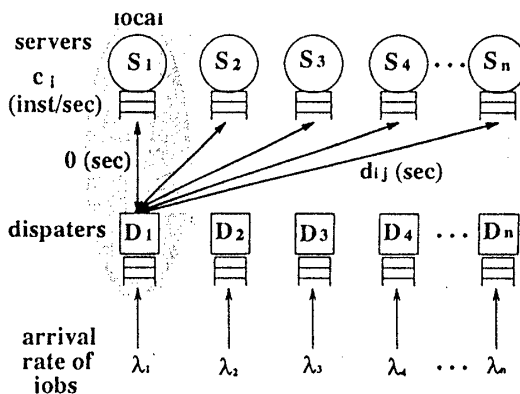


図1: 分散型ディスパッチャのモデル

バがあるモデルにおいて、ディスパッチャに待ち行列を形成し閾値を用いてジョブを配送する方法である。

ディスパッチャは、ジョブが到着した時、サーバ  $i$  が空いているサーバの中で最も速ければ閾値  $\tau_i$  の閾値制御に従う。すなわち、到着したジョブを含めた待ち行列長が閾値  $\tau_i$  以上であればサーバ  $i$  へ配送し、そうでなければ待ち行列に並べる。待ち行列の先頭のジョブは、処理中である速いサーバを待って、空いたら配送される。

閾値は以下のように決定される。空いている最も速いサーバの処理速度を  $\mu_{open}$  とし、 $\mu_{open}$  よりも速いサーバの数を  $N_{faster}$ 、 $N_{faster}$  の平均処理速度を  $\mu_{ave}$  とする。このとき、平均遅延  $D_\tau$  は、

$$D_\tau = \frac{N_\tau}{N_{faster} \rho \mu_{ave}} + \frac{Z_\tau}{\mu_{open}}$$

で表せる。ここで、 $N_\tau$  は  $M/M/N_{faster}/k$  システムにおけるジョブ数の平均、 $Z_\tau$  はブロック確率であり、 $\rho = \frac{\lambda}{N_{faster} \mu_{ave}}$  である。呼損率を求めるアラン B 関数：

$B(N_{faster}, N_{faster} \rho)$  を  $B$  で表すと、 $N_\tau, Z_\tau$  は、

$$N_\tau = \frac{N_{faster} \rho + B N_{faster} \rho (-1 + \frac{1-\rho^\tau}{1-\rho}) + B \rho (\frac{1-(1+\tau)\rho^\tau + \tau\rho^{\tau+1}}{(1-\rho)^2})}{1 + B \rho \frac{1-\rho^\tau}{1-\rho}}$$

$$Z_\tau = \frac{B \rho^\tau}{1 + B \rho \frac{1-\rho^\tau}{1-\rho}}$$

となる。次に、差分関数  $\Delta(\tau) \equiv D_\tau - D_{\tau-1}$  を以下のように定義すると、

$$\Delta(\tau) \propto \tau + N_{faster} (R-1) (\rho-1) + B \rho (N_{faster} (1-R) + \frac{\tau}{1-\rho} + \frac{\rho^\tau - 1}{(1-\rho)^2})$$

ここで、 $R \equiv \frac{\mu_{\text{avg}}}{\mu_{\text{open}}}$  である。 $\Delta(\tau) < 0$  である最大の整数値  $\tau$  が集中型における最適閾値である。対象とするジョブを含めた待ち行列長が閾値以上であれば、空いている最も速いサーバへジョブを配送する。

### DD 方式

本稿では、D 方式を分散型ディスパッチャを用いたシステムにそのまま適用した DD(distributed deterministic) 方式をまず提案する。各ディスパッチャはそれぞれに D 方式にしたがった配送を行なう。即ち、各ディスパッチャごとに各サーバに対して閾値を設定する。各ディスパッチャは、実際にサーバが空いているかではなく、自分がそのサーバにジョブを一つ送っているかどうかにより、空いているサーバを認識する。ここで、D 方式では処理速度の遅いサーバが無限個あると仮定されているが、本研究では有限であることを注意する。よって、空いているサーバがない場合には、ジョブは到着したディスパッチャの待ち行列に並ぶとする。

### DDER 方式

DD 方式は、他のディスパッチャから配送されるジョブについてはまったく考慮していない。つまり、各サーバの能力を全て利用できるものとして、閾値を設定している。各ディスパッチャは、閾値を越えていなければ速いサーバを待った方が有効であるとしてジョブを待ち行列に並べる。しかし実際には、サーバは他のディスパッチャからのジョブも処理しているため、待ち行列が長い場合には空いているサーバへ配送した方が有効となる。そこで、DD 方式を改良した DDER(DD based on equivalent service rate) 方式を提案する。

DDER 方式では、各ディスパッチャは他のディスパッチャが自分と同程度のジョブを配送していると仮定し、 $c_i$  を実際の  $\frac{1}{n}$  であると認識する。即ち閾値を決定する時に、 $c_i$  を実際の  $\frac{1}{n}$  として計算する。その結果、DD 方式と比較して閾値が小さくなり空いているサーバへジョブを配送しやすくなる。

### 比較する方式

DD 方式と DDER 方式の特性を検討するために、著者らがすでに提案している三つの方式 [4] と比較する。

LR\_PB 方式 ビギンバックを用いて環境を観測し、それに基づいて最小の時間で応答すると予想されるサーバへ配送する方式。

TC( $\tau$ ) 方式 ローカルサーバの予想応答時間が閾値  $\tau$  を越えるときには、(配送後の経過時間 /  $c_i$ ) が最小のサーバへ配送する。

TR( $\tau$ ) 方式 ローカルサーバの予想応答時間が閾値  $\tau$  を越える時には、全てのサーバの中から処理速度の比に相当する確率でランダムに選ぶ。

### 3 シミュレーション結果と考察

DD、DDER、LR\_PB、TR、TC の各方式の比較を行なった。シミュレーション仮定を以下に示す。

- ・サーバ数  $n = 10$
- ・サーバの処理速度  $\vec{C} = (1, 1, 2, 2, 2, 2, 4, 4, 8, 8)$
- ・到着率  $\lambda$  はどのディスパッチャも等しい

図 2 に  $d = 0$  の場合の到着率と応答時間の関係を示す。DD 方式、DDER 方式は、低負荷領域において有効である。しかし、DD 方式は各ディスパッチャが他のディ

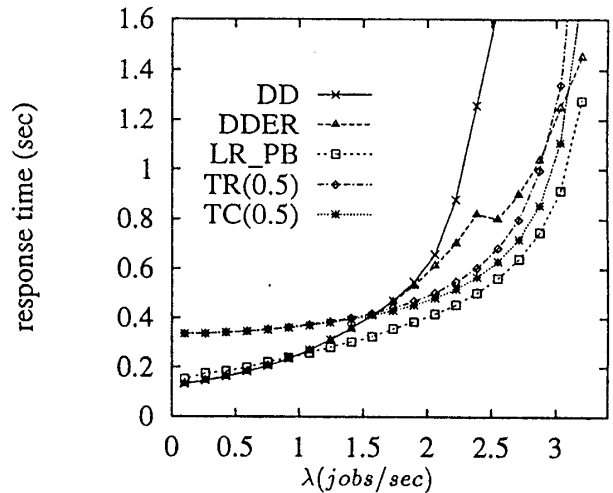


図 2: 各方式の応答時間特性  
( $n = 10, \vec{C} = (1, 1, 2, 2, 2, 2, 4, 4, 8, 8), d = 0$ )

スパッチャに到着するジョブを考慮していないため、中・高負荷領域では特性が悪い。DDER 方式は、中・高負荷領域において DD 方式より改善されているが、その他の方式と比較すると有効な方式とはいえない。DD 方式は、低・中負荷領域では  $c = 8$ 、 $c = 4$  のサーバのみを使用し、 $\lambda = 1.9$  付近から  $c = 2$  のサーバを利用し始める。DDER 方式は、 $\lambda = 1.3$  付近から  $c = 2$  のサーバを、 $\lambda = 2.5$  付近から  $c = 1$  のサーバを利用し始める。また、通信遅延が大きい  $d = 0.5$  の場合も評価した。その結果、DD 方式及び DDER 方式は通信遅延を考慮していないため LR\_PB と比べるとかなり特性が劣化することがわかった。

### 4 まとめ

本稿では、集中型ディスパッチャに適した配送方式として考案された D 方式を分散型ディスパッチャに適用した、DD 方式と DDER 方式を提案した。そしてこれらと、すでに提案した他の方式とを比較、検討した。その結果、他のディスパッチャが配送するジョブ及び、通信遅延を考慮して、DDER 方式を改善する必要があることがわかった。

### 参考文献

- [1] S.A.Banawan and J.Zahorjan: "Load Sharing in Heterogeneous Queueing Systems," IEEE INFO-COM '89,1989.
- [2] Flavio Bonomi and Anurag Kumar: "Adaptive Optimal Load Balancing in a Nonhomogeneous Multiserver System with a Central job Scheduler," IEEE Trans. on Comput., Vol.28, No.5,1990.
- [3] Scott Shenker and Abel Weinrib: "The Optimal Control of Heterogeneous Queueing Systems: A Paradigm for Load-Sharing and Routing," IEEE Trans. on Comput., Vol.38, No.12,1989.
- [4] 渡辺尚, 太田剛, 水野忠則: "双方向ビギンバックを用いた動的負荷分散方式について," 電子情報通信学会春季全国大会,1994.