マルチメディア通信と分散処理 74－26
グループウェア 15－26
（1996. 1. 26）

双方向ピギーバックを用いた動的負荷分散における負荷情報の補間法

染葉 佳代子　太田 剛　渡辺 尚　水野 忠則

静岡大学

分散システムにおいて、各ノードへのジョブの到着は一定ではないため負荷の不均衡が生じる。これを解消し、リソースを有効利用するための負荷分散制御について研究がなされている。我々はこれまでに双方向ピギーバックによる環境観測で得た情報を用いて予想応答時間が最小のサーバへジョブを配送する方式を提案した。しかしこの方式には、高負荷であったノードが低負荷になったときの情報が伝達されにくいという問題点があるため、本稿ではそれを解消するような二つの改良方式を提案する。改良方式は、古い情報をいつまでも利用することがないように、各ノードが保持している情報を時間経過に従って変化させる。これらの方式の性能を評価するためにシミュレーションを行い、その有効性について検討する。

Interpolation scheme of load information for adaptive load balancing with bidirectional piggybacking

Kayoko Someha　Tsuyoshi Ohta　Takashi Watanabe　Tadanori Mizuno

Shizuoka University

In distributed systems, various job arrival rate on each node causes load imbalance among computers. Load balancing schemes have been studied to improve the above problem as well as an effective use of distributed resources. We have proposed an adaptive load balancing strategy, called LR-PB, with bidirectional piggybacking. However, this strategy is hard to notify load information to the other nodes when a heavy-loaded node turns to light-loaded. In this paper, we propose two improved strategies which adjust load information based on elapsed time from receipt so that we shouldn't use dated information. We show simulation results to evaluate and discuss the effectiveness of the improved strategies.

1　まえがき

個々にリソースを有する計算機 (ノード) をネットワークで接続した分散システムでは、各ノードへのジョブ到着は一定ではない。そのため、過負荷のノードがある一方で、アイドルのノードがあるといった状態が生じる。システムのリソースを有効利用するためには、そのような状態を解消し、各ノードの負荷量を均等化するような負荷分散制御を行う必要がある。現在のシステムでは、各ノードの状況をユーザが判断してジョブを投入しなければならず、リソースの分散化やネットワークの複雑化が進んでいる現状を考えると、この負担はユーザにとって大きなものとなる。そこで、ユーザがシステムの状況を意識せずとも最適な計算機によって処理が受けられるような、負荷分散制御を計算機が行うシステムについてこれまで種々の研究がなされてきた。

負荷分散の制御を行うディスパッチャ形態は集中型と分散型の 2 種類に分類できる。集中型は、システム内のノードに関する情報やシステムに投入されたジョブを一元管理する形態である。分散型は、各ノードがそれぞれ独自にシステム内のノードの状況を収集し、ノードに投入されたジョブを管理する形態である。Tantawi

と Towsley[3] によって最適なジョブの分配率を求めるアルゴリズムが提案されている。また、Mirchandaney と Towsley、Stankovic[1] や Shivaratri と Krueger、Singhal[2] は、しきい値を越えているサーバが越えていないサーバを、あるいはしきい値を越えていないサーバが越えているサーバを通信により探し、ジョブの移動を行う方式が有効であるとしている。サイクリック方式は、サーバの処理能力や各ノードへのジョブの到着率が均質な場合には良い性能を示すことも示されている [6]。集中型制御の研究はこれまでに多くなされていること、最近のシステムはリソースの分散化が進んでおり、それに対応できる負荷分散を行う必要があることから、本研究では分散型制御を対象としている。

分散型制御における負荷分散戦略はスタティック、ダイナミック、アダプティヴに分類することができる。スタティックな負荷分散戦略はシステムのその時の状態に独立であり、あらかじめ設定されているノードの性能値のみを考慮する。ダイナミックな戦略は常にシステムの状態を考慮する。アダプティヴな戦略はシステム状態の変化を反映するようにパラメータや方式を変化させる。

アダプティヴな戦略は次の 4 つの要素から成る (Shivaratri,Krueger and Singhal [2])。

インフォメーション方式 他のノードについて、いつ、どこで、どのような情報を収集するかを決定する。

ロケーション方式 ジョブの受け渡しの相手ノードを決定する。

トランスファー方式 ノードがジョブを別のノードに配送する側であるのか、別のノードからジョブを受け取る側であるのかを決定する。

セレクション方式 どのジョブを移動するかを決定する。

我々の研究では、インフォメーション方式にピギーバックを用いる (詳細は次節に述べる)。ロケーション方式は、インフォメーション方式に基づいて収集された情報と、サーバのサービス率や通信遅延から、ジョブを各ノードへ配送した場合の予想応答時間を計算し、これが最小となるノードを配送先に決定する。トランスファー方式は予想応答時間を他のノードと相対的に比較することにより決定する方法をとっている。セレクション方式はそのときに外部からノードに到着したジョブとする。本研究では、インフォメーション方式に焦点をあてる。

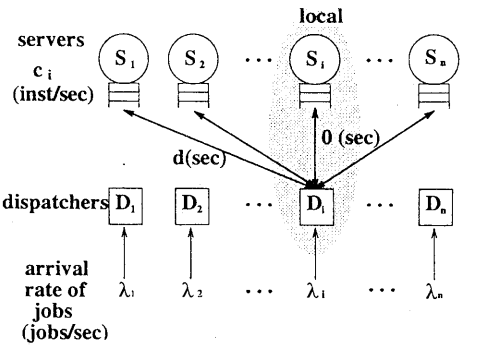

図 1: システムモデル

本稿では、第 2 節で我々がこれまでに提案している方式について説明しその問題点を述べる。第 3 節で本稿で提案する二つの負荷分散方式の詳しいアルゴリズムについて述べ、第 4 節でシミュレーションを用いた性能評価を行い、第 5 節でシミュレーション結果について議論する。そして第 6 節でまとめと今後の課題について述べる。

2 LR-PB 方式

我々はこれまでに [5][6] において、LR-PB 方式 (load balancing with Least Response time scheme based on piggybacking information; [5][6] では TLR[0] と表記) を提案し、その性能について検討してきた。

この方式では図 1のようなシステムモデルを対象としている。

- システムは n 個のノードから成り、各ノードはディスパッチャとサーバを一つずつ持っている。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。
- D_i にはジョブが平均 λ_i 個／秒でポアソン到着する。$\vec{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_n)$ とおく。
- 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。
- 通信遅延はローカル間を 0 秒、リモート間を d_{ij} 秒とする
- ディスパッチャはローカルサーバの負荷状況が瞬時にわかる。
- S_i の処理率を c_i 命令／秒とする。$\vec{C} = (c_1, c_2, \ldots, c_n)$ とおく。

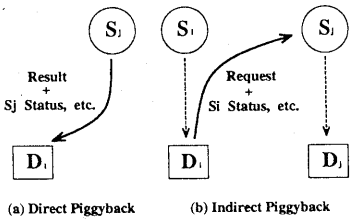

図 2: 直接ピギーバックと間接ピギーバック

LR-PB 方式では、インフォメーション方式として双方向ピギーバックを用いる。双方向ピギーバックは直接ピギーバックと間接ピギーバックからなる (図 2)。S_j が D_i から配送されてきたジョブを処理し終ったとき、ジョブの処理結果に S_j の負荷情報 (システム内客数) を付加して D_i に配送することにより、ノード j の情報をノード i に伝達するのが直接ピギーバックである。一方、間接ピギーバックは、D_i が S_j へジョブを配送するとき S_i の負荷情報を配送するジョブに付加し、ノード i の情報をノード j に伝達する方法である。これにより、ジョブのリモート配送が行われるごとに、ディスパッチャが保持している他のノードの負荷情報は更新される。この情報収集方法は、同報通信を用いる方法に比べトラフィックへの影響が少ない点、実行して始めてわかるジョブ量などに応じてジョブのクラス分けを行い、ユーザの要求に柔軟に対処する負荷分散への拡張性の点を考慮して選ばれた。

また、ロケーション方式については、ディスパッチャ D_i は双方向ピギーバックにより収集した情報と、そのサーバのサービス率や通信遅延から、ジョブを各ノードのサーバへ配送した場合の予想応答時間を計算し、これが最小となるノードをデスティネーションノードに決定する。トランスファー方式は、他のノードとの相対的な予想応答時間の比較により決定する方法をとっている。セレクション方式はそのときに外部からノードに到着したジョブとし、他のノードから負荷分散戦略によって配送されてきたジョブは対象としない。

ただし、これまでの研究から、双方向ピギーバックを用いたインフォメーション方式には次に示す問題点があることがわかっている。例えば、ノード A へ他のノード B から高負荷であるという情報が伝えられたとする。ノード A はその情報を保持し、自ノードを含む他のノードの負荷がノード B よりも高くならない限り、ノード B へジョブを配送しなくなる。一方、ノード B は負荷が高い間はジョブを他のノードへ配送するが、負荷が軽くなると自ノードで処理するようになり、自ノードの負荷が低くなったという情報は他のノードへは伝えられにくい。よって、ノード A はノード B が高負荷であると思ったままであり、ノード B は有効に利用されない。つまり、各ディスパッチャは他のノードの負荷を実際より高く見積りやすい傾向にある。

この欠点を補う方法として次の二つが考えられる。1) ローカルサーバの情報だけでなく、他のサーバについての情報もピギーバックする。つまり、多くの情報を伝達して少しでも新しい情報を収集する。2) ピギーバックで得た情報の信頼度を考慮する。1) に関しては選択的多重ピギーバックをインフォメーション方式に用いる LR-SMP 方式を [7] において検討した。しかし、複数のノードが同じ情報を持つことになるため一つのサーバへジョブが集中するという問題があり、ローカルサーバの情報を扱う (LR-PB 方式) だけで十分であることがわかった。そこで、本稿では 2) の方法について検討する。

3 他サーバの情報の信頼性を考慮した方式

前章で述べた LR-PB 方式の問題点を解消するために、インフォメーション方式を改良する。各ディスパッチャでは、他のサーバの負荷を高く見積るのを防ぐため、ピギーバックで得た情報をそのまま保持するのではなく、時間経過とともに変化させていく。本稿では、次の二つの方式を提案する。

LR-RPB 方式 (load balancing with Least Response time scheme based on Reseted Piggybacking information) ピギーバックで得た情報が有効である時間 (有効時間) を設定し、有効時間を越えたときには情報をリセットする方式である。

ディスパッチャ D_i に新しいジョブの到着があったとき、D_i が保持している各ノードの負荷情報 $q_{ij}(i = 1, 2, 3, \ldots, n, j = 1, 2, 3, \ldots, n)$ について次のような処理を行う。

$$if \ \text{(経過時間} \geq \text{有効時間)} \ then$$

$$q_{ij} = J_{ij}$$

J_{ij} は D_i から s_j へ配送したがまだ返送されていないジョブ数である。有効時間が t 秒のものを LR-RPB[t] と表す。

LR-ESR 方式 (load balancing with Least Response time scheme based on Equivalent Service Rate) LR-RPB 方式において、情報の有効時間はサーバの内客数に依存せず、すべて一率である。また、突然 0 にリセットするのも実状に即しているとは言えない。そこでこの方式では、時間の経過にともない一定の割合で情報の内客数を減少させる。減少させることによって、負荷を高く見積る傾向にあるピギーバック情報に補正を加えることができると考えられる。

ディスパッチャ D_i に新しいジョブの到着があったとき、D_i が保持している各ノードの負荷情報 $q_{ij}(i = 1,2,3,\ldots,n, j = 1,2,3,\ldots,n)$ を用いて予想負荷 $\hat{q}_{ij}$ を求める。

$$\hat{q}_{ij} = -a * \text{経過時間} + q_{ij}$$

$$if \ \ (\hat{q}_{ij} < J_{ij}) \ \ then \ \ \hat{q}_{ij} = J_{ij}$$

ここで、a は内客数の減少率である。ジョブの配送先を決定するときの予想応答時間 R_j の計算には $\hat{q}_{ij}$ を用いる。情報の内客数の減少率が a のものを LR-ESR[a] と表す。

4 シミュレーション

本稿で提案した二つの方式の性能をシミュレーションを用いて評価する。システムモデルは第 2 節で示したものを用いるが、簡単のために $\mu = 1$、$d_{ij} = d$、ディスパッチに要する時間は 0 秒とする。また、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。また、$n = 10$ とする。

4.1 均質システム

まず、同じ処理能力を持つノード群が通信遅延を無視できる回線でつながれており、各ノードへのジョブの到着は均等であるシステムについて考察する。すなわち、$d_{ij} = 0$、$\vec{\lambda} = (\lambda, \lambda, \ldots, \lambda)$、$\vec{C} = (1,1,\ldots,1)$ の場合を図 3 に示す。

LR-RPB 方式の有効時間による性能差は、低負荷においてほとんどないが、中・高負荷領域では LR-RPB[30] 方式が最も良い性能を示した。情報を常に新しくしておくためには有効時間が短い方が良いが、有効時間を短くしすぎると、実際は負荷が高いかもしれないサーバの情報を 0 にリセットするため効率が悪くなる。仮定した条件では、有効時間を 30 秒とした場合が最適であった。LR-ESR 方式の情報の内客数の減少率は、$0.01 \leq a \leq 1$ では、a が小さくなるにつれて性能が良くなるが、$a \leq 0.01$ になるとほとんど性能差はない。

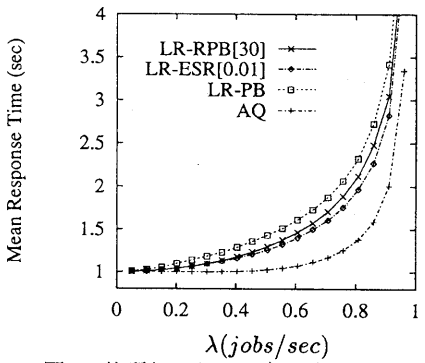

図 3: 均質網における各方式の比較

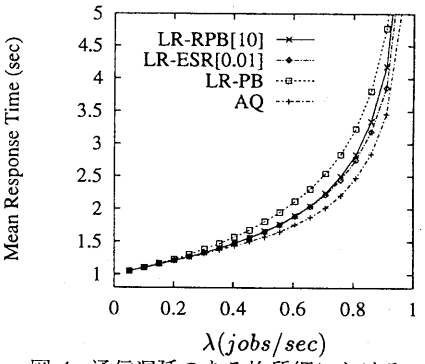

図 4: 通信遅延のある均質網における各方式の比較

LR-RPB[30] 方式、LR-ESR[0.01] 方式ともに LR-PB 方式より良い性能を示している。低負荷領域では同程度の性能であるが、負荷が高くなるにつれて LR-ESR[0.01] 方式は LR-RPB[30] 方式より良くなる。

4.2 通信遅延のあるシステム

次に、通信遅延が無視できない場合 ($d_{ij} = 0.5$) の結果を図 4 に示す。

LR-RPB 方式は、有効時間による性能差がほどんどないが、中負荷で LR-RPB[10] 方式が、高負荷で LR-RPB[20] 方式が最も良い性能を示した。LR-ESR 方式も減少率による性能差はほとんどないが、LR-ESR[0.01] が最も性能が良い。

LR-RPB[10] 方式、LR-ESR[0.01] 方式ともに LR-PB 方式より良い性能を示しているが、低・中負荷領域で二つの方式にほとんど差はない。高負荷になると LR-ESR[0.01] 方式が良くなる。

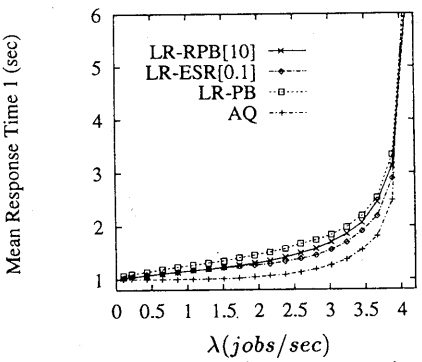

図 5: ジョブ到着の不均質なシステムにおける各方式の比較 (ノード 1、 2)

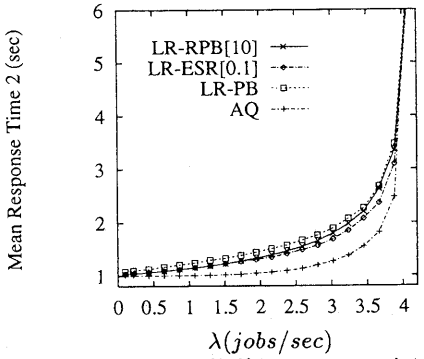

図 6: ジョブ到着の不均質なシステムにおける各方式の比較 (ノード 3 ～ 10)

4.3 到着率の異なるシステム

均質網において、各ノードへのジョブの到着率が異なる場合について考察する。ここでは、$d_{ij} = 0$、$\vec{\lambda} = (\lambda, \lambda, 0.2, 0.2, \ldots, 0.2)$、$\vec{C} = (1, 1, \ldots, 1)$ の場合を図 5、 6 に示す。

LR-RPB 方式に関して言えば、ジョブ到着率の高いノードに到着したジョブの平均応答時間については LR-RPB[10] 方式が有効であるが、その他の到着率の低いノードについてはほとんど差がない。ジョブ到着率の高いサーバは低いサーバへジョブを多く配送する必要があること、そのためジョブ到着率の低いサーバが持っている高いサーバの情報は頻繁に更新されていることから、高いサーバに関する情報をリセットすることはあまりないと思われる。LR-ESR 方式では、ジョブ到着率の高いノードに到着したジョブの平均応答時間に関しては、低負荷では LR-ESR[0.001] 方式が良いが、負荷が高くなるに従って減少率が大きいものが有効になり、超高負荷では LR-ESR[1] 方式が良い。高負荷になるにつれてジョブ到着率の差が大きくなり、高いノードに到着したジョブのみが負荷量に影響を与えるようになるためであると考えられる。

図 5、 6のように、すべての領域において LR-RPB 方式、 LR-ESR 方式ともに LR-PB 方式より良い性能を示している。

5 議論

負荷情報を高く見積りやすいという LR-PB 方式の問題点を解決するために、 LR-ESR 方式では、保持している情報の内容数を時間経過とともに減少させるという方法をとった。シミュレーション結果に関して情報の正確さの観点から議論する。図 7 ～図 9 は通信遅延の無視できる均質システムにおける配送先決定時の (サーバ S_i の負荷情報の内容数)−(サーバ S_i の真の内容数) を用いて、情報の正確さを表したものである ($n = 10$、 $Ci = 1$、 $\lambda_i = 0.7$、 $d_{ij} = 0$、 $i = 1, 2, \ldots, n$)。図 7 に示すように、 LR-PB 方式では実際より高く負荷を見積ることが多い。

図 8 に示すように、 LR-ESR[0.01] 方式においては、 LR-PB 方式における高い見積りがかなり解消されている。さらに減少率の大きい LR-ESR[1] 方式では、図 9 に示すように、負荷を低く見積りすぎる。このように、各ノードが保持している情報の正確さについての最適な傾きが存在する。

しかし、ジョブの配送先として選択したノードに関する負荷情報の正確さは、減少率が大きすぎると負荷を低く見積りすぎるが、減少率を小さくしてもある程度以上の高い見積りをするようにはならない。よって前節で述べた通り、通信遅延の無視できる均質システムにおいて、$a \leq 0.01$ の LR-ESR[a] 方式ではほとんど性能差はない。

LR-RPB 方式と LR-ESR 方式を比較すると、すべてのシステム仮定において LR-ESR 方式が有効である。両方式は、負荷を高く見積るという欠点を解消することにより、 LR-PB 方式より性能が向上した。しかし LR-ESR 方式は、予想応答時間が等しい複数のサーバが存在した場合の配送先決定に際して、ランダム選択を行っていたものがサイクリック選択に近くなるという副次的効果をも持つ。この効果が性能の向上にどのくらい影響しているかを、 LR-PB 方式にこの効果と同様の制御を加えることで示す。 LR-PB 方式において予想応答時間が最小のサーバが

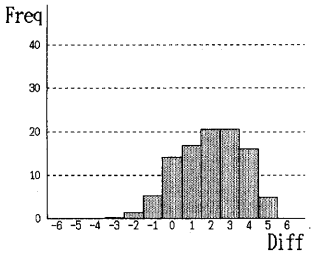

図 7: LR-PB における
負荷情報の正確さ

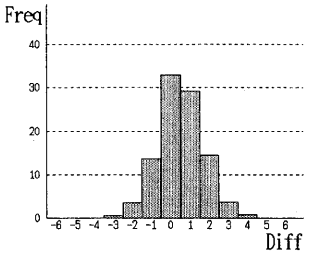

図 8: LR-ESR[0.01] における
負荷情報の正確さ

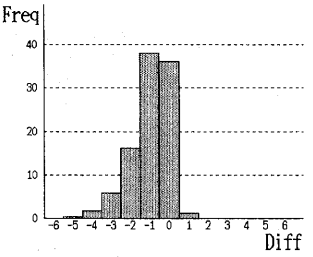

図 9: LR-ESR[1] における
負荷情報の正確さ

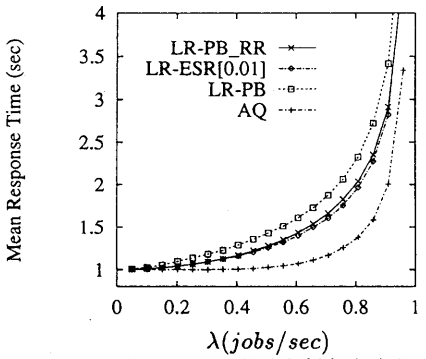

図 10: サイクリック配送を行う方式と
各方式の比較

複数ある場合、次のような制御を行うと LR-ESR 方式と同様の配送となる。1) ローカルサーバの内客数が 0 のときはローカルサーバへ配送する。2) ローカルサーバの内客数が 0 でないときは、予想応答時間が最小のリモートサーバの中から、最後の配送後、最も時間が経過しているサーバを選択する。この条件でシミュレーションを行った結果、図 10に示すように LR-PB 方式と比較して性能が向上することが確認された。

6 むすび

本稿では、以前提案した LR-PB 方式のインフォメーション方式を改良した方式を提案した。LR-PB 方式で用いられる双方向ピギーバックは、サーバの負荷を実際より高く見積る傾向にある。本稿では、これを改良するための二方式について述べた。一つは双方向ピギーバックで得た情報に有効時間を設定しそれを過ぎた情報はリセットされる LR-RPB 方式であり、もう一つは、時間の経過にともない一定の割合で情報の内客数を減少させる LR-ESR 方式である。そしてこれらの負荷分散方式がどのような性能を示すのかを、シミュレーションを用いて検討した。

その結果、LR-RPB 方式、LR-ESR 方式ともに、システム仮定によって、情報の有効時間や情報の内客数の減少率の最適値は異なることがわかった。検討したシステム仮定すべてにおいて、最適なパラメータを持つ二つの改良方式は LR-PB 方式より性能が向上した。中でも、LR-ESR 方式は良い性能を示した。

本稿では、負荷分散制御にかかる CPU オーバヘッドについて考察していない。負荷分散制御が複雑化することを考えると、今後検討する必要があると思われる。

参考文献

[1] R. Mirchandaney, D. Towsley and J. A. Stankovic, "Analysis of the Effects of Delays on Load Sharing," IEEE Trans.on Comput., vol.38, No.11, 1989.

[2] N. G. Shivaratri, P. Krueger and M. Singhal, "Load Distributing for Locally Distributed Systems," IEEE Computer, Dec 1992.

[3] A. N. Tantawi and D. Towsley, "Optimal Static Load Balancing in Distributed Computer Systems," JACM., Vol.32, No.2., 1985.

[4] T.Ohta, T.Watanabe and T.Mizuno, "A Job Dependent Dispatching Scheme in a Heterogeneous Multiserver Network," IEICE Trans. commun., Vol. E77-B, No.11 Nov, 1994.

[5] 渡辺, 太田, 水野, 中西, "双方向ピギーバックに基づいた動的負荷分散方式," 電子情報通信学会論文誌 D-I Vol. J78-D-I, No.3, 1995.

[6] 染葉, 渡辺, 太田, 水野, "双方向ピギーバックを用いたジョブ配送法について," 情報処理学会研究報告 Vol.94, No.56, 1994.

[7] 染葉, 太田, 渡辺, 水野, "選択的多重ピギーバックを用いた負荷分散とその特性" 電子情報通信学会技術研究報告 Vol.94, No.537, 1995.