

複数のロードバランサによる Webシステムの応答時間最適化

株式会社 アイ・アイ・エム 河野 知行

Webシステムのスケーラビリティや可用性を確保するためにロードバランサが採用される。ロードバランサで処理できるトラフィック量にも制限があるため、トラフィック量が多いWebシステムでは、複数のロードバランサを使用することも考えられる。本研究では、複数のロードバランサで単一のWebシステム群の負荷分散を行う場合に考慮すべき事項を、初心者に判りやすく説明する手法を考察する。

Performance impact of Load Balancers on Server Farms.

Tomoyuki Kawano, IIM Corporation

Server Farms have gained popularity for providing scalable and reliable computing services. A load balancer plays a key role in this architecture. Selecting and using the proper load balancer to match the characteristics of the servers will have a significant performance impact. This paper examines some commonly used load balancing algorithms for server farm, introduces a performance model as a basis for the analysis, and show how to select a load balancer to maximize the performance potential of the server farms.

はじめに

ロードバランサは、ネットワークとWebサーバ（以降サーバと呼ぶ）の間に位置し、到着した通信パケットの内容を検査し、その処理要求を振り分けるサーバを決定する。候補となったサーバが複数存在する場合、それらのサーバの状況に応じて、振り分け先となるサーバを決定している。

一台のロードバランサで、処理要求を振り分ける方式はシンプルであり理解しやすい。しかし、ロードバランサもコンピュータであり、処理能力に限界がある。ロードバランサのCPU使用率が100%に達した場合、到着した処理要求の検査に遅延が生じ、性能問題を引き起こす。

ロードバランサに性能問題が生じた場合、より性能の高いロードバランサにリプレースすることになる。もし、性能の高いロードバランサがない場合、複数のロードバランサで同じサーバ群の負荷分散を行うことになる。

一台のロードバランサで負荷分散を行っているシステムの接続図を図1に示す。このレポートで考察するのは、複数のロードバランサで負荷分散を行うシステムである。そのシステムの接続図を図2に示す。この接続形態では、2台のロードバランサが、サーバ群（サーバ1とサーバ2）に処理要求を渡すパスを持つ。

図2では、2台のロードバランサと2台のサーバ

が示されているが、台数には制限がないと考えていただきたい。本研究では話を簡単にするため、ロードバランサとサーバを共に2台であると限定しているだけである。

図2に示された2台のロードバランサ（バランサ1とバランサ2）は、それぞれが相手の状況を知らずに負荷分散を行うものとする。2台のロードバランサがお互いに通信し、相手の状態を知って負荷分散を行うのであれば、一台のロードバランサで負荷分散を行っているのと等価になってしまうからである。

また、全てのサーバの処理能力が同じであれば話は簡単である。このため、サーバの処理能力が違った場合も考察しよう。

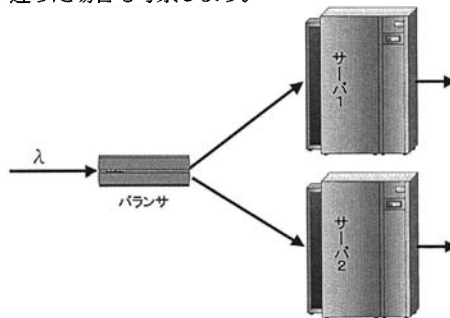


図1 ロードバランサのシステム構成

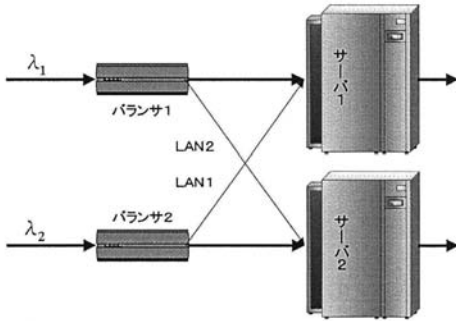


図2 本研究対象のシステム構成

ロードバランサの効果を評価するには、いくつかの方法が考えられる。例えば、サーバの稼働率（もしくはCPU使用率）が均等になっているかを判定する方法がある。しかし、あまりお勧めできるものではない。それよりも、そのアプリケーションの処理に必要であった平均応答時間の優劣で、ロードバランサの効果を判定するのが最適であろう。

システムの概要

考察するシステムの概要を定義しておこう。先ず、2台のサーバでは全く同じアプリケーションが稼働しており、そのサーバ内で処理が完結するものとする。また2台のサーバは独立して動作し、一切相互に干渉しないものとする。

前述したように2台のロードバランサは他系のバランサを意識することなく動作する。また、バランサ1はサーバ1に内蔵され、バランサ2はサーバ2に内蔵されていると想定する。

バランサの内部処理による遅延は生じないものとし、内蔵されたバランサとサーバ間（自系通信でありバランサ1とサーバ1、もしくはバランサ2とサーバ2の間）の通信時間は必要ないものとする。一方、他系通信（バランサ1とサーバ2、もしくはバランサ2とサーバ1の間）の通信では1ミリ秒の通信時間が生じるものとする。

今回の検証で想定するアプリケーションは、サーバのCPU時間で10ミリ秒/処理（平均サービス時間）が必要であるとする。CPU使用以外の要因による遅延は一切生じないものとする。また、バランサ1とバランサ2には、この処理要求が秒あたりの総計で100件到着するものとする。もし、この全ての処理要求を一台のサーバで処理したとすると、そのサーバのCPU使用率は100%となり、応答時間が非常に長くなることは容易に想像できる。そこでロードバランサの出番となる。

一台のロードバランサで負荷分散を行う場合、それぞれのサーバが均等の処理を行う環境が提供される。つまり、サーバ1とサーバ2の稼働率は均等に、CPU使用率が50%でバランスするであろうことは容易に想像できる。複数のロードバランサで負荷分散を行った場合は、どうなるであろうか。この考察を行うため、バランサ1とバランサ2に秒あたり1件と99件、2件と98件、3件と97件・・・と到着率を変化させながらロードバランサの効果を判定したい。

前述したがサーバの処理能力が均等な場合と、均等でない場合（ここでは非均等と呼ぶ）の2つのケースを考察する。均等な場合は、サーバ1とサーバ2は共に10ミリ秒の平均サービス時間が必要であるとする。一方、非均等な場合にはサーバ1が7.5ミリ秒、サーバ2が10ミリ秒の平均サービス時間であるとする。

ロードバランサで提供される処理要求の振り分け技法には、幾つかのものがある。本研究では定率分散、状態分散、応答時間分散の3種について考察する。それぞれの技法における応答時間を求め、それら技法の特徴を把握したい。

ロードバランスなし

先ず、ロードバランサを使用しない場合の応答時間を考察してみよう。バランサ1に到着した処理要求は全てサーバ1で、バランサ2に到着した処理要求は全てサーバ2で処理されるケースである。バランサでの遅延は生じないし、自系のバランサとサーバ間の通信遅延は生じないとしている。このため、この場合にはバランサ自体が存在しないものとして取り扱うことができる。

サーバ1とサーバ2の応答時間は、次の待ち行列式（M/M/1）で求めることができるものとする：

$$R_1 = \frac{S_1}{1 - \rho_1} = \frac{S_1}{1 - \lambda_1 S_1}$$

$$R_2 = \frac{S_2}{1 - \rho_2} = \frac{S_2}{1 - \lambda_2 S_2}$$

$$\lambda_1 + \lambda_2 = 100$$

式3 サーバ応答時間の待ち行列計算式

R_1 はサーバ1の応答時間、 S_1 はサーバ1のサービス時間（10ミリ秒）、 ρ_1 はサーバ1の使用率、 λ_1 はサーバ1の到着率（処理要求数/秒）である。 λ_1 を1から99まで変化させることにより、サーバ1とサーバ2で処理される処理要求の比率が変化

し、サーバ1とサーバ2の応答時間も変化する。

サーバ1の到着率が1であれば、サーバ2の到着率は99であり、サーバ2の平均応答時間は1000ミリ秒となる。処理要求全体の平均応答時間は990ミリ秒となる。サーバ1の到着率が50の場合、バラサ2の到着率も50となり、平均応答時間はサーバ1とサーバ2共に20ミリ秒となる。この結果、ロードバラサがない場合、少しでも到着率に偏りが生じると、極端に応答時間が悪化することが判る。

均等サーバ構成の場合は、サーバ1とサーバ2の到着率が同じとき、平均応答時間が最も短くなる。非均等サーバ構成の場合は、どうであろうか。

サーバ1のサービス時間が7.5ミリ秒、サーバ2のサービス時間が10ミリ秒であると仮定する。つまり、サーバ1はサーバ2に比べ1.33倍の処理能力を持っていることになる。この非均等型のサーバ構成においても、式3に示した待ち行列式を使用することができる。

この非均等型の構成では、それぞれのサーバの到着率が99の場合、サーバ1は応答時間が29ミリ秒であり、サーバ2は1000ミリ秒である。均等型と同様に到着率に偏りがある場合、平均応答時間は急激に大きくなる。均等型と違い、両サーバの稼働率（CPU使用率）が均等になる到着率は50件/秒にはならない。その値は、次の計算式で求めることができる。

$$\begin{aligned} \frac{S_1}{(1-\rho_1)} &= \frac{S_2}{(1-\rho_2)} & S_1 &= 7.5ms \\ S_1(1-\rho_2) &= S_2(1-\rho_1) & S_2 &= 10ms \\ S_1 - S_1\rho_2 &= S_2 - S_2\rho_1 & \lambda_1 + \lambda_2 &= 100 \\ S_1 - S_2 &= S_1\rho_2 - S_2\rho_1 & \lambda_2 - \lambda_1 &= \lambda_2 - 100 + \lambda_2 \\ S_1 - S_2 &= S_1S_2\lambda_2 - S_2S_1\lambda_1 & 2\lambda_2 &= \frac{0.0075 - 0.01}{0.0075 \times 0.01} + 100 \\ & & &= 66.66666 \\ \frac{S_1 - S_2}{S_1S_2} &= \lambda_2 - \lambda_1 & \lambda_2 &= 33.33 \\ & & \lambda_1 &= 100 - \lambda_2 = 66.67 \end{aligned}$$

式4 非均等型での使用率の均衡

サーバ1の到着率が66.67件/秒、サーバ2の到着率が33.33件/秒のところ、両サーバの稼働率が同じになるところである。その時のサーバの平均応答時間は15ミリ秒である。注意深く考察していただければ、平均応答時間が最小となるのは、稼働率が同じになるときではない。

平均応答時間が最小となる到着率を計算で求めることもできるが、ここではその説明を割愛する。但し、その時の到着率はサーバ1で61.88件/秒、サーバ2で38.12件/秒である。また、その時の平均応答時間は14.82ミリ秒となる。

定率分散

ロードバラサがない場合を想定し、到着する処理要求数が増えたときに平均応答時間が変動する様子を検証した。到着する処理要求の総数（ λ_1 と λ_2 の合計）は100件/秒で一定であるが、 λ_1 と λ_2 の割合を変化させて考察した。

平均応答時間が最も小さくなるのは、次の表に示す処理要求数が与えられた場合である。尚、均等型はサーバ1とサーバ2の処理速度が同じであり、非均等型はサーバ1がサーバ2に比べ1.33倍の処理能力を持っている。

構成	λ_1 (件/秒)	λ_2 (件/秒)	平均応答時間 (ミリ秒)
均等型	50	50	20
非均等型	61.88	38.12	14.82

ここに示された処理要求数に少しでも偏りが生じると、平均応答時間は指数的に悪くなる。最悪のケースは片側のサーバに全ての処理要求が与えられた（ λ_1 もしくは λ_2 の一方が100件/秒となる）場合で、均等型ではサーバの使用率が100%（10ミリ秒の処理時間×秒当たり100件の処理要求）となり、待ち行列では応答時間が無限大となってしまふ。非均等型では、サーバ1の処理能力が高いため、その使用率は75%にしかならず、応答時間は30ミリ秒となる。但し、非均等型であってもサーバ2側にすべての処理要求が集中すると、使用率は100%となり応答時間は無限大となる。

ロードバラサを使用することなく、到着する処理要求が前述のベストバランス状態になることは非常に稀である。言い換えると、このベストバランス状態を期待することはできない。

そこでロードバラサを導入し、到着する処理要求に偏りが生じて、2台のサーバの処理要求数を上記のベストバランス状態に近づける。均等型である場合、バラサは到着した処理要求をサーバ1とサーバ2に50%ずつ分散させれば、ベストバランス（それぞれのサーバが半分ずつの処理要求を分担）することになる。

図5に均等型のシステム構成における処理要求の振り分け状況を示す。バラサ1に到着した処理要求の50%はサーバ1へ、残りの処理要求はサーバ2へ送られることになる。

自系サーバ（バラサ1の場合はサーバ1）への処理要求の転送時間はゼロであると想定している。但し、他系サーバ（バラサ1の場合はサーバ2）へ処理要求を転送する場合にはLANを使

用し、このLANでは1ミリ秒の転送時間が必要であるとしている。なお、LANの応答時間はM/M/1で計算できるものと仮定する。

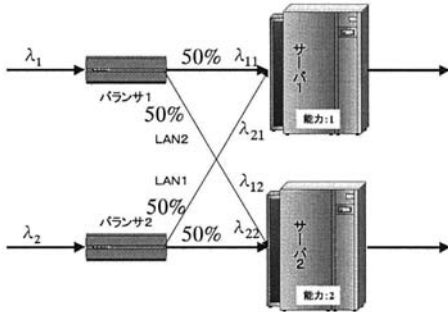


図5 均等型での定率分散

すると、次の待ち行列式で処理要求の平均応答時間を求めることができる。バランサ1は到着した処理要求 λ_1 を λ_{11} と λ_{12} に分割し、それぞれサーバ1とサーバ2に送る。バランサ1に到着した処理要求の平均応答時間 R_1 は、サーバ1の応答時間 SRV_1 とLAN2の応答時間 LAN_2 およびサーバ2の応答時間 SRV_2 により決定される。バランサ2に到着した処理要求の応答時間 R_2 も同様の計算で求めることができる。

$$\lambda_{11} = \lambda_{12} = \frac{\lambda_1}{2}$$

$$\lambda_{21} = \lambda_{22} = \frac{\lambda_2}{2}$$

$$R_1 = \frac{SRV_1}{2} + \frac{LAN_2 + SRV_2}{2} = \left(\frac{S_{srv1}}{1 - \rho_{srv1}} + \frac{S_{lan2}}{1 - \rho_{lan2}} + \frac{S_{srv2}}{1 - \rho_{srv2}} \right) + 2$$

$$= \left(\frac{S_{srv1}}{1 - (\lambda_{11} + \lambda_{21})S_{srv1}} + \frac{S_{lan2}}{1 - \lambda_{12}S_{lan2}} + \frac{S_{srv2}}{1 - (\lambda_{12} + \lambda_{22})S_{srv2}} \right) + 2$$

$$\text{処理要求の平均応答時間} = \frac{R_1\lambda_1 + R_2\lambda_2}{\lambda_1 + \lambda_2}$$

バランサ1とバランサ2に到着した処理要求の半分ずつがサーバ1とサーバ2に割り振られるので、サーバ1とサーバ2の使用率は同じとなり安定する。その応答時間は20ミリ秒である。

バランサが他系サーバと通信する際に利用するLANのアクセス回数は、 λ_1 と λ_2 の半分ずつである。LAN1とLAN2のアクセス回数と同じになるのは、 λ_1 と λ_2 が同じになるときであり、25回/秒となる。その際の応答時間は1.025ミリ秒となる。

$$R_{LAN} = \frac{S_{LAN}}{1 - \rho_{LAN}} = \frac{1}{1 - 0.001 \times 25} = 1.025$$

λ_1 と λ_2 が同じでない場合、LAN1もしくはLAN2へアクセス回数の偏りが生じ、アクセス回数の多い側の平均応答時間が長くなる。もし一方のLANに50件/秒のアクセスが生じると、その平均応答時間は1.052ミリ秒となる。

前述したようにサーバ1とサーバ2の応答時間は20ミリ秒である。また、到着した処理要求の半分が他系サーバに転送される。 λ_1 と λ_2 が同じであれば、LANの平均転送時間は1.025ミリ秒であった。すると、総処理要求におけるLANを使用した平均転送時間は0.5125ミリ秒となり、処理要求の平均応答時間は約20.5ミリ秒となる。

図6に示すのが、均等型でのバランサなしと定率分散の応答時間の推移である。横軸がバランサ1に到着した処理要求数、縦軸が応答時間(ミリ秒)である。バランサがない場合、バランサ1と2に到着する処理要求数に偏りがあると、応答時間は指数的に悪くなる。定率分散では、到着する処理要求数に偏りがあっても、指数的に応答時間が悪化する問題は解消される。図7は、図6の縦軸を拡大したものである。応答時間が最小となっているポイントを確認すると、バランサがない場合で20ミリ秒、定率分散で約20.5ミリ秒となっている。定率分散では、LANの転送遅延時間(0.5ミリ秒)が上乘せされていることが判る。

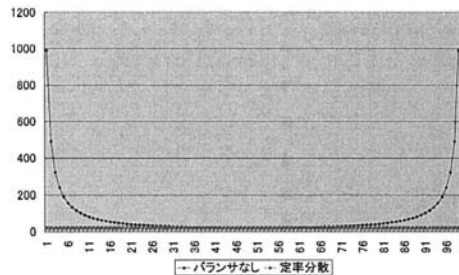


図6 均等型でのバランサなしと定率分散 (1)

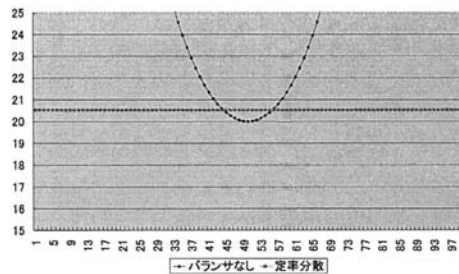


図7 均等型でのバランサなしと定率分散 (2)

非均等型のサーバ構成では、サーバ能力に応じた処理要求の振り分け方式を検討しなければならない。本研究で想定する非均等型のシステム構成では、一つの要求を処理するのにサーバ1では7.5ミリ秒、サーバ2では10ミリ秒のCPU時間を必要とする。つまり、サーバ1はサーバ2の1.33倍の処理能力を持っている。

この条件の下に、両サーバの稼働率が等しくなる処理要求数は、式4より λ_1 が66.67件/秒、 λ_2 が33.33件/秒となる。つまり、サーバ1に3分の2、サーバ2に3分の1の処理要求を振り分けることにより、応答時間が安定化することになる。

その時の応答時間は次式で求めることができる。 R_1 はバランサ1に到着した処理要求の応答時間であるが、バランサ2側の応答時間 R_2 についても同様に算出することができる。

$$\begin{aligned} \lambda_1 &= \lambda_{11} + \lambda_{12} \\ \lambda_{11} &= \lambda_1 \times 0.6667 \\ \lambda_{12} &= \lambda_1 \times 0.3333 \\ \lambda_2 &= \lambda_{21} + \lambda_{22} \\ \lambda_{21} &= \lambda_2 \times 0.6667 \\ \lambda_{22} &= \lambda_2 \times 0.3333 \end{aligned}$$

$$R_1 = \frac{2SRV_1}{3} + \frac{LAN_2 + SRV_2}{3} = \left(\frac{2S_{srv1}}{1-\rho_{srv1}} + \frac{S_{lan2}}{1-\rho_{lan2}} + \frac{S_{srv2}}{1-\rho_{srv2}} \right) + 3$$

$$= \left(\frac{2S_{srv1}}{1-(\lambda_{11} + \lambda_{21})S_{srv1}} + \frac{S_{lan2}}{1-\lambda_{12}S_{lan2}} + \frac{S_{srv2}}{1-(\lambda_{12} + \lambda_{22})S_{srv2}} \right) + 3$$

処理要求の平均応答時間 = $\frac{R_1\lambda_1 + R_2\lambda_2}{\lambda_1 + \lambda_2}$

到着した処理要求の3分の2がサーバ1へ、残りがサーバ2に割り振られるので、サーバ1とサーバ2の応答時間は同じとなり安定する。その時のサーバの平均応答時間は15ミリ秒である。

バランサが他系サーバを通信する際に利用するLANのアクセス回数は、 λ_1 の3分の1、 λ_2 の3分の2である。このため、 λ_2 側に多くの処理要求が到着すると、より多くの処理要求が他系サーバへ転送されることになる。つまり、LAN1の使用率が高まり転送遅延時間も大きくなる。

図8に示すのが、非均等型でのバランサなしと定率分散の応答時間の推移である。バランサがない場合は到着する処理要求数に偏りがあると応答時間は指数的に悪くなる。一方、定率分散ではこの問題は解消されている。図9は、図8の縦軸を拡大したものである。応答時間が最小となっているポイントを確認すると、バランサがない場合で14.82ミリ秒、定率分散では λ_1 が99件/秒となったときの約15.34ミリ秒である。このときが、最もLANの転送遅延時間が短くなる。

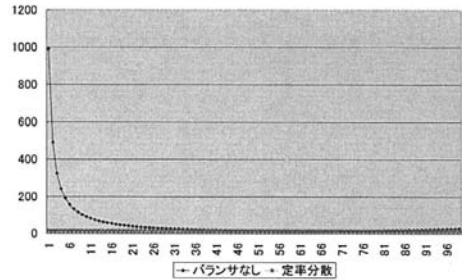


図8 非均等型でのバランサなしと定率分散(1)

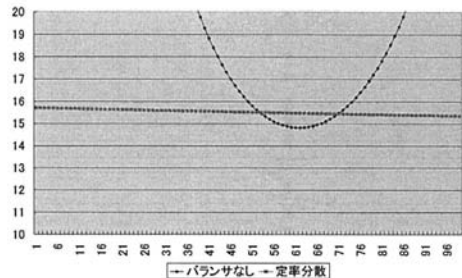


図9 非均等型でのバランサなしと定率分散(2)

状態分散

定率分散はバランサに到着した処理要求を、予め定められた比率で振り分けるものである。候補サーバの状態を無視してバランサの都合で振り分け作業を行っているといっても過言ではない。

もう少し、高度な負荷分散機能を有したロードバランサのことを考えてみよう。サーバの状態を監視しつつ、その時点で最適と考えられる候補サーバを動的に選択するものである。

Webサーバでは、スレッド数を制御することにより、並列処理が可能な処理要求数を制限している。もし、並列処理可能な数以上の処理要求が到着した場合、新たに到着した処理要求は待ち行列に加えられ、その処理の開始は待たされる。

状態分散とは、バランサがサーバの待ち行列の状況を監視しながら、新たに到着した処理要求を振り分ける候補サーバを決定するものである。本研究では話を簡単にするため、サーバにおける並列処理可能な処理要求数は1であると想定する。

図10に状態分散で使用される制御ロジックを示す。この図に示されたロジックでは、新たな処理要求が到着する度に自系サーバと他系サーバの状態を比較し、待ち状態となっている処理要求が少ないサーバ(処理要求の処理が早く開始されるサーバ)に到着した処理要求を振り分けるようにしている。

このロジックでは、待ち行列の状態として3つを想定している。まず、アイドル状態。この状態は、該当サーバで処理されている処理要求がないことを示す。次は、待ちなし状態。現在、サーバでは処理要求を処理しているが、その処理の完了を待っている処理要求はないことを示す。最後は、待ちは一つの状態。サーバでは処理要求の処理が進行中であり、また、その処理要求の完了を待っている処理要求が一つある状態を示す。

自系サーバと他系サーバの状態を比較するために、バランスはサーバとの通信を行う必要がある。バランスは新たな処理要求が到着する度に、サーバの状態情報を取得するが、自系サーバとの通信には遅延が生じない。一方、他系サーバからの状態情報の取得には通信時間が必要となる。この他系サーバの状態情報の取得にはLANを使用し、その通信時間は1ミリ秒であるとする。またその通信遅延時間は $M/M/1$ の計算式で求められるものとする。

この条件の下での処理要求の応答時間を、待ち行列式で算出することもできるであろう。しかし、 $M/M/1$ などの簡素な式で表現することはできないため、この条件でシミュレーションを行い、その結果を基に考察する。

ここで一つ、考察しておくべきことがある。他系サーバ(バランス1から見たサーバ2としよう)の状態情報を取得するためにLANの1ミリ秒の通信時間が必要となる。この通信時間の中に、他のバランスが新たな処理要求をサーバ2に振り分けた場合、取得した状態情報と実際のサーバ2の状態に相違が生じる。シミュレーションでは、この事象も正確に再現することにした。

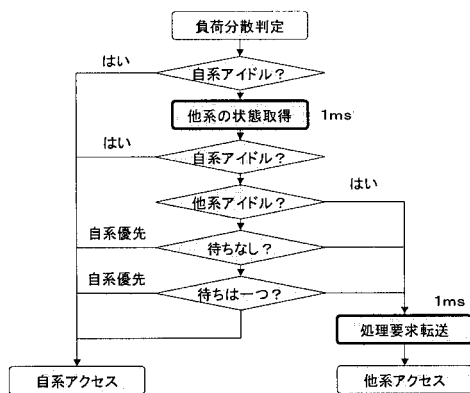


図10 状態分散での候補サーバの選択

応答時間分散

定率分散は、バランスに到着した処理要求を予め定められた比率で振り分けた。この場合、バランスはサーバの動作状況を監視していない。一方、状態分散ではサーバの状態を比較しながら、処理待ち時間が短いと思われるサーバに到着した処理要求を振り分けた。

ロードバランスの負荷分散機能として、これら以外の技法はないのか。もう一つの方法として、サーバの応答時間を比較し、その時点で短い応答時間で処理を行っているサーバに、新たな処理要求を振り分ける技法を考察してみよう。

まず、ある一定量の処理要求を2つのサーバに振り分ける。以降、応答時間の短かったサーバに処理要求を振り分けることにする。しかし、複数のバランスで同じサーバ群を制御していれば、全てのバランスからの処理要求が、応答時間の短いサーバへ集中してしまう。このため、サーバでの処理遅延が生じ、そのサーバの応答時間は悪くなるであろう。

一旦、応答時間の短いサーバが決定されると、そのサーバへの処理要求の振り分けが集中し、他のサーバへの処理要求の振り分けは激減する。すると、処理要求が振り分けられなくなったサーバの処理遅延は解消され、応答時間が短くなるはずである。バランスは、負荷が軽くなったサーバの応答時間も継続的に監視しておく必要がある。

このために、バランスが応答時間の短いサーバへ処理要求を振り分けている間も、ある一定量の処理要求は、他のサーバに振り分け、それらのサーバの応答時間を計測しておく。このようにして、常時サーバごとの応答時間を把握しておき、サーバの応答時間比較を継続するものとする。

本研究では、バランスは応答時間の短いサーバへ99%の処理要求を振り分け、残り1%の処理要求は、他のサーバへ振り分けるものとする。このような振り分けを行い、処理要求の応答時間を最小化すると同時に、その時点でのサーバの応答時間計測を可能とする。

このような制御ロジックの負荷分散機能の応答時間を待ち行列式で求めるのは困難である。そこで状態分散の場合と同様に、シミュレーションによって応答時間を求める。

結果の考察

バランスがない場合、バランス1と2に到着する処理要求に偏りが生じると、応答時間は指数的に悪くなった。定率分散のバランスを採用することにより、到着する処理要求に偏りが生じても、応答時間を安定させることが可能になった。

状態分散や応答時間分散の効果は、どうである

うか。均等型のシステム構成における、各負荷分散技法の応答時間の推移を図11に示す。また、それぞれの最小応答時間を表12に示す。これらの図表を見比べていただければ、状態分散が応答時間を短縮するのに、効果的であることが判る。

本研究では、状態分散でアイドル、待ちはなし、待ちが一つの3つの状態比較を行った。この3つの結果から断定するのは難しいかもしれないが、待ちが2つ、待ちが3つ・・・と条件を複雑化しても、応答時間の改善率は低いと思われる。

応答時間分散は、定率分散よりも複雑な振り分けロジックを必要とするが、その効果はロジックの複雑さに比例していない。定率分散よりも応答時間が若干改善される程度である。

図13に応答時間分散での処理要求の振り分け状況を示す。他系サーバへの処理要求の転送(λ_{12} と λ_{21})が、少ない割合でも一定量、確保されている部分が存在する。この部分が、待ち状況が悪いが応答時間を計測するために処理要求を振り分けている部分である。

図14に非均等型での応答時間の推移を示す。サーバ1とサーバ2の処理性能が違うため、最小応答時間が記録される処理要求の到着率が均等型とは若干違う。

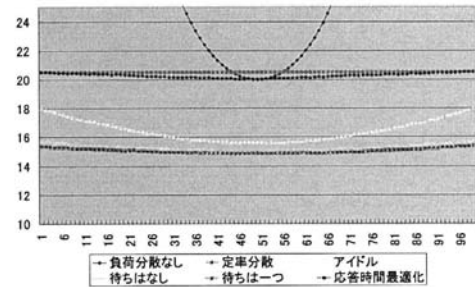


図11 均等型での応答時間比較

技法	λ_1 (件/秒)	λ_2 (件/秒)	平均応答時間 (ミリ秒)
バラバラなし	50	50	20
定率分散	50	50	20.5
アイドル	50	50	15.59
待ちはなし	50	50	15.01
待ちが一つ	50	50	14.91
応答時間分散	50	50	20.04

表12 均等型での最小応答時間

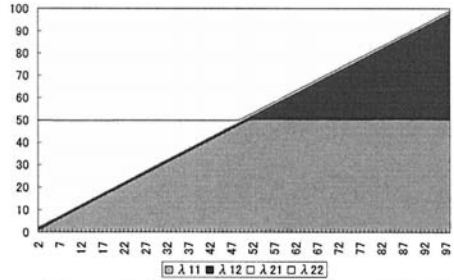


図13 均等型の応答時間分散での到着率

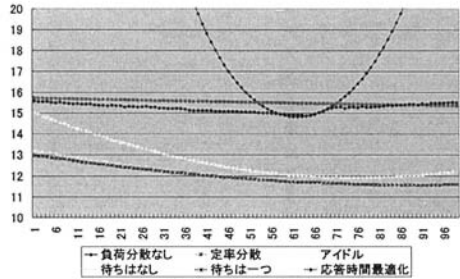


図14 非均等型の応答時間比較

まとめ

バラバラなし、定率分散、状態分散、応答時間分散の4つのケースを比較した。それらの振り分け技法の特徴を整理してみよう。

- ① バラバラがなくとも、処理要求がバラバラ1と2に偏りなく到着すれば、速い応答時間が確保される。しかし、偏りが生じれば、応答時間は指数的に悪くなる。
- ② 定率分散は到着する処理要求に偏りが生じても、応答時間が指数的に悪くなる問題は解消できる。但し、偏りが無い場合は、バラバラがない場合より応答時間は長くなる。
- ③ 状態分散を行うと、応答時間は大幅に短縮される。M/M/n構成の待ち行列のような効果を得ることができる。
- ④ 応答時間分散は、ロジックの複雑さに比例した効果を得ることができない。定率分散の改善型と位置づけられる技法であろう。

本研究では、一般的なロードバラバラが提供していると考えられる処理要求の振り分け技法を検証した。その結果より、状態分散が最も効果的な振り分け技法であるとの結論を得た。

しかしこの結論は、本研究で仮定した条件の下でのものであり、実際のWebシステムに適用できるものとは考えていない。大切なのは、先入観ではなく、十分な検討を行って振り分け技法を選

択すべきであるということである。

参考文献

野瀬純郎：情報システムの性能評価技法、
IIMレポート1998 NO105～109
亀田壽夫、紀一誠、李韻：性能評価の基礎と応用
共立出版 1998

以上