

粒度に着目した異機種計算機間での最適負荷分散方式の提案

3W-3

岡村耕二 勝山光太郎
三菱電機(株) 情報電子研究所

1. はじめに

近年、計算機の能力の向上と廉価化に伴い、従来はホストコンピュータや汎用機で行っていた業務を、複数のワークステーションやパーソナルコンピュータといった比較的小型の計算機群で構成されるシステムで行なう形態が注目されている。このような流れを業界ではダウンサイジングと呼んでいる。このダウンサイジングを実現するシステムの一つとして、クライアント/サーバ型システムがある。クライアント/サーバ型システムは数台の高性能なサーバと、数十台のクライアントで構成されている。

クライアント/サーバ型システムでは、クライアントはデータの入出力を行ない、サーバはクライアントで入力されたデータの処理を行なうという形態の分散処理が行なわれている。本稿では、このようなクライアント/サーバ型システムにおける分散処理について、最適な負荷分散を行なうための方法を提案し、考察する。

2. 異機種計算機間での分散処理

クライアント/サーバ型システムでは、クライアントにはパーソナルコンピュータが用いられていることが多い。クライアントがパーソナルコンピュータである場合、以下の点に着目することができる。

1. クライアントの通信は遅い

パーソナルコンピュータのOSは、本来通信用の機能を持っていない。そのため、OSとは別の通信用のモジュールをメモリに常駐させることで通信を行なっている。このため、パーソナルコンピュータの通信は遅くなりがちである。

2. クライアントは非同期受信ができない

パーソナルコンピュータのOSは、通常シングルタスクであるため、受信用のプロセスを生成して受信待ちにすることはできない。そのため、非同期通信を行なうためには、通常ポーリングが用いられる。そのため、最悪の場合でそのポーリング間隔の遅れが生じる可能性がある。この受信の遅れは通信のオーバーヘッドと見ることもできる。

以上のようにクライアント/サーバ型システムにおける分散処理では、クライアントの通信のオーバーヘッドが比較的大きい。本稿では、分散処理の効率を向上させるために、このクライアントの通信のオーバーヘッドを小さくする負荷分散方式に着目した。

3. 負荷分散方式の提案

分散処理を通信に着目した場合、通信を粒度 [1] [2] の概念を用いて扱うことにより、その分散処理の通信形態のモデル化を容易にすることができる。

A proposal of optimum load balancing method among various computers taking account into granularity
K. OKAMURA and K. KATSUYAMA
Computer & Information Systems Laboratory,
MITSUBISHI Electric Corporation

3.1 分散処理のモデル

クライアント-サーバ間の分散処理と粒度の関係を図-1に示す。

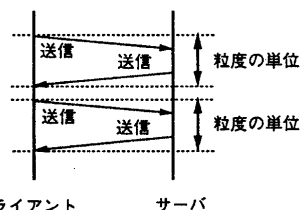


図-1: クライアント-サーバ間通信と粒度

粒度の単位を、クライアントからの処理要求のための送信から、サーバでの処理、サーバからの結果の送信までとする。クライアントからサーバへの通信が稀である時は、粒度は粗く、クライアントからサーバへの通信が頻繁な時は粒度が細かいとすることができる。

クライアントとサーバが分散処理を行なう場合に生じるクライアントとサーバ間の通信の粒度を最適にしながら負荷分散を行なうことにより通信のオーバーヘッドを小さくし、分散処理の効率を向上させることができる。

3.2 最適な負荷分散

例えば、クライアントとサーバが頻繁に分散処理を行なっている時は、通常では通信も頻繁に行なわれている。この場合、複数の処理のための通信をまとめて一回で行なえば、クライアントの通信のオーバーヘッドを小さくすることができる。回数を減らし、通信のオーバーヘッドを押えることができる。この様子を図-2に示す。

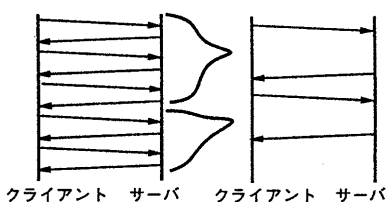


図-2: 粒度の変更

クライアント/サーバ型システムにおける分散処理において、多くの場合は通信の粒度を粗くすることにより分散処理の効率を向上させることができる。しかし、最適な粒度が必ずしも最も粒度が粗い場合であるとは限らない。例えば、粒度が粗い時は、いくつかの処理がまとめて通信されているため、応答時間は悪くなる。

このように、粒度を変更して負荷分散を行なう時、システムの分散処理効率の向上と、一方で悪くなる応答時間のトレードオフを考慮しながら最適な粒度を決定してゆく必要がある。

4. 分散処理の考察

以上述べた通信の粒度を最適にしながら負荷分散を行なう分散処理の解析とその効果の予想を行なう。本稿では、分散処理の効率の尺度として、スループット(単位時間内に行なえる処理の数)を用いた。

4.1 分散処理の解析

粒度を変更に対する、応答時間とスループットの変化の解析を行なう。

まず、2章で述べたクライアントの通信のオーバーヘッドを詳しく解析する。図-3で示されるように、クライアントでは、受信時に、最悪の場合、最大でポーリング間隔の時間の遅れが生じている。

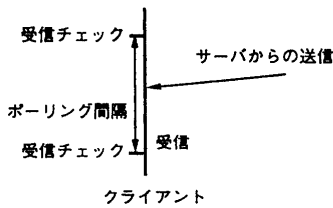


図-3: クライアントの受信

あるクライアントとサーバ間の処理の終了から次の処理が始まるまでの時間間隔を C_Time、サーバ上での処理時間を S_Time、通信時間を CM_Time、ポーリング間隔の時間を P_Time、とする。これらの関係を図-4に示す。

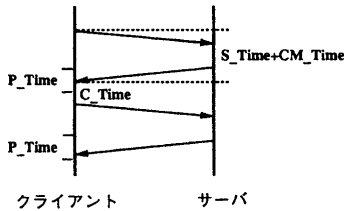


図-4: クライアント-サーバ間の分散処理

n個の分散処理が負荷分散によりm回で行なわれた(n ≥ m)とすると、応答時間、スループットは以下ようになる。この時クライアントとサーバ間の一回の通信には、 $\frac{n}{m}$ 個の分散処理データが含まれている。

応答時間は、以下の式で求めることができる。

$$S_Time \times \frac{n}{m} + C_Time \times (\frac{n}{m} - 1) + CM_Time + P_Time$$

スループットは以下の式で求めることができる。

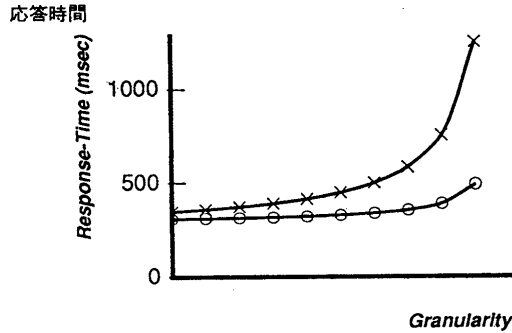
$$\frac{\frac{n}{m}}{S_Time \times \frac{n}{m} + C_Time \times (\frac{n}{m} - 1) + CM_Time + P_Time}$$

4.2 分散処理の予想

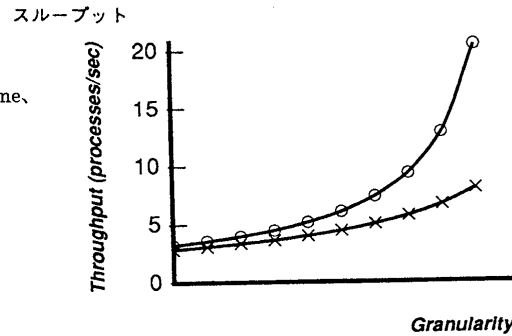
n=100個、CM_Time=200 msec、P_Time=100 msec として、m、S_Time、C_Timeの値を変えた時の応答時間、およ

びスループットの予想を行なった。これらの値は実測値の典型的な値である。

グラフ-1および、グラフ-2において、○で示される曲線はC_Timeおよび、S_Timeが短い場合で、×で示される曲線はそれぞれ長い場合である。グラフの横軸はmの値(100 ≥ m ≥ 10)であり、左から右に向かうにつれて粒度は粗くなる。横軸の一番左端はm=nの場合である。



グラフ-1: 粒度と応答時間



グラフ-2: 粒度とスループット

C_Time および S_Time が短い場合は通信の粒度が粗くなくても応答時間が悪くならず、スループットも高いことがわかる。このような場合は、最適な粒度はかなり粗い。一方、C_Time および S_Time が長い場合は、粒度が粗くなり過ぎた場合の応答時間が極めて悪くなるので、最適な粒度の決定には注意を払う必要がある。

5. まとめ

本稿では、クライアント/サーバ型システムにおける分散処理において、通信の粒度を最適にすることで最適な負荷分散処理を行なうことができることを示し、その時の分散処理の効率について考察を行なった。今後は、本負荷分散方式を実際のクライアント/サーバ型システムの分散処理に適用して、評価してゆく予定である。

参考文献

[1] 小柳: "数値計算の並列アルゴリズム", bit, Vol.21, No.4, pp.378-385 (1989)
 [2] 岡村 他: "パソコンシステムにおける分散処理効率について", 第44回 情報処理学会 全国大会 (1992)