

7D-6

パラレルシステムDBへの
分散DBアクセスにおける接続プロセッサ決定方式

松野英樹 江島新吉 石川博道 山本裕
(株)日立製作所

1. はじめに

パラレルシステム下における非共用DB方式での分散DBアクセスにおける、接続プロセッサの決定方式を挙げ、それぞれの性能、互換性、実現性等について考察する。

2. 共用DB方式と非共用DB方式

パラレルシステム下におけるDBの共用方式として、共用DB方式(Shared Everything)と非共用DB方式(Shared Nothing)がある(図1)。

共用DB方式は、全プロセッサが全DBを共有する方式で、どのプロセッサからでも直接DBアクセスすることが可能である反面、プロセッサ間でのDBアクセス排他制御方式やそのオーバーヘッドに課題がある。

一方、非共用DB方式は、各DBが、あるプロセッサにくくりつけられる方式で、排他制御は単一プロセッサの場合とほぼ同様であるが、自プロセッサ下でないDBへのアクセスにプロセッサ間通信のオーバーヘッドが生じる。

上記の特徴は分散DBアクセスを行う場合でも同様である。

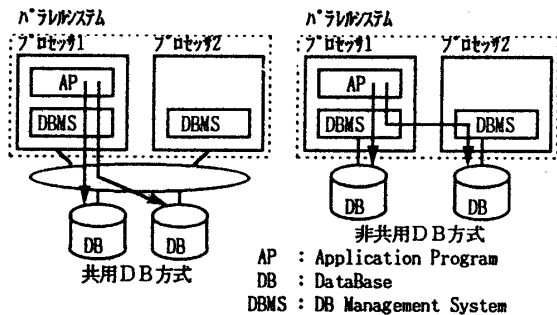


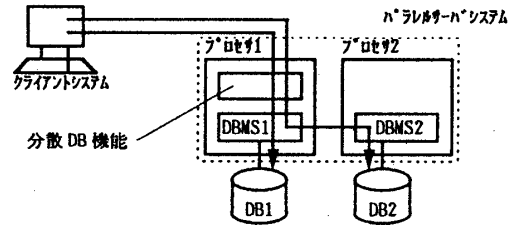
図1 共用DB方式と非共用DB方式

3. 接続プロセッサ決定のモデル

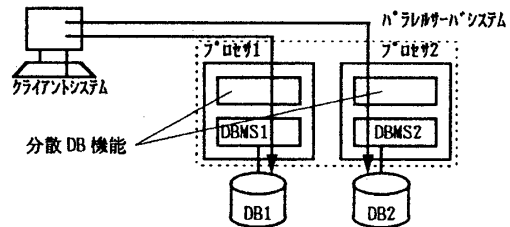
本文では、非共用DB方式パラレルDBサーバへの分散DBアクセスの3つのモデルに対して比較を行う(図2)。

モデル1は、パラレルシステム中の1プロセッサのみを分散DB機能の受け口とする方式である。例えばプロセッサ1に分散DB機能の受け口を置いた場合、プロセッサ1にAPを置いたローカルアクセスと同様の形態となる。

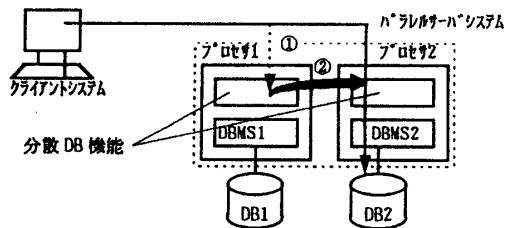
モデル2は、パラレルシステム中の全プロセッサに分散DB機能の受け口を置き、クライアントシステムがアクセス対象DBの位置を意識して、接続



(a) モデル1



(b) モデル2



(c) モデル3

図2 接続プロセッサ決定のモデル

先プロセッサを決定する方式である。

モデル3は、最初に接続するプロセッサはいずれか1つに決めておき、そのプロセッサに一旦接続後アクセス対象のDBが解った時点(OSI-RDA使用の場合、R-Open受信直後)で、DBのあるプロセッサに通信路を接続し直す方式である。

4. 性能の比較

図2のモデルについて、分散DBアクセスを行う際の性能の比較を行う。ここでは、簡単化のため、モデル1, 2, 3で

- クライアントサーバ間の通信時間 : xn
- プロセッサ間通信時間 : yn
- プロセッサ1での処理時間 : an
- プロセッサ2での処理時間 : bn
- モデル3での通信路切換え時間 : α

(n はモデルとする(図3))。

The method of connection processor selection
for parallel system DB on distributed DB access.
Hideki Matsuno, Shinkichi Ebata,
Hiromichi Ishikawa, Hiroshi Yamamoto
Hitachi, Ltd.

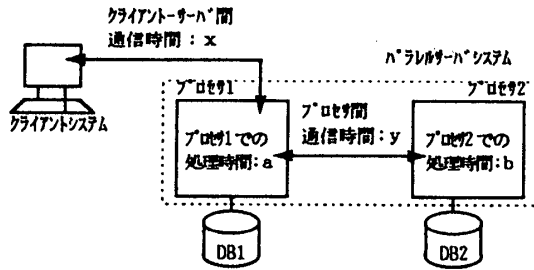


図3 性能比較のモデル

DB1へアクセスするときの時間 T_n は

$$T1 = T2 = T3 = xn + an$$

であり、各モデルとも差はない。

これに対しDB2へのアクセス時間は、

$$T1 = x1 + a1 + y1 + b1$$

$$T2 = x2 + b2$$

$$T3 = x3 + a3 + b3 + \alpha$$

となり差が生じる。

$$x1 \approx x2 \approx x3, b1 \approx b2 \approx b3$$

と考えられ、 $T2$ が最も優れていることはわかるので、 $T2$ と $T1$ 、 $T3$ の比較を行う。

x 側と y 側でのデータの転送量はおおむね同程度であるが、データ転送速度は一般にプロセッサ間(y 側)が100MB/S程度であるのに対し、クライアント-サーバ間(x 側)は、LANの場合で数M~数10MB/S、WANなら数K~数MB/S程度であり、クライアント-サーバ間の通信路がWANのように低速である場合、プロセッサ間通信時間は無視できる程度である($xn \gg yn$)。

また、DBアクセス条件が数100行以上の一括検索であったり、複数表のジョイン処理のようにプロセッサ2側で高負荷を要する場合など $bn \gg an$ となる。

さらに、低負荷なアクセス条件の時に通信路が低速であれば $xn \gg an, bn$ といえる。

以上の事項をふまえて、 $T1$ の概算値を求めると表1のようになる。これから解るようにクライアント-サーバ間の通信路が低速である場合は $T2$ との性能差は無視できる程度であるが、高速通信路の場合プロセッサ間通信のオーバーヘッドやプロセッサ内での処理時間が性能差として表面化する。

表1 $T1$ の概算値

	高速通信路	低速通信路
高負荷な処理	$x1+y1+b1 (\approx T2+y1)$	$x1+b1 (\approx T2)$
低負荷な処理	$x1+a1+y1+b1 (\approx T2+a1+y1)$	$x1 (\approx T2)$

$T3$ に関しては、クライアント-サーバ間の通信路が低速な場合 $T1$ のときと同様に $T2$ との性能差は問題にならない。また、高速通信路である場合も $a3, \alpha$ は通信路切換までに要する負荷であり、通信路切換後DB2へのアクセスが続くような処理では問題にならないが、DB1とDB2へのアクセスが交互に連続するような場合、通信路の切換えが頻発して性能悪化を招き $T2$ より劣る。

5. 移行性・拡張性の比較

図2のモデルMについて、単一プロセッサをマルチプロセッサに移行する場合、すでにあるマルチプロセッサにさらにプロセッサの追加を行う場合、あるいはマルチプロセッサ内でDBの所在地等の構成変更を行う場合等の、移行性や拡張性について比較する。移行性・拡張性を考える場合、クライアントシステムから見て、マルチプロセッサシステムが1システムイメージとなるか否かが重要なポイントとなる。

モデルM1, M3はクライアントからの接続要求の受け口が1つであり、1システムイメージであると言える。これに対して、モデルM2は複数の受け口を持ち、1システムイメージとはなっていない。

例えば、単一プロセッサをマルチプロセッサに移行する場合、モデルM1, 3はクライアントシステムのノード対応定義等を変更する必要がないのに対して、モデルM2では全クライアントシステムの定義変更を行う必要があり、モデルM1, 3に対して、移行性・拡張性が劣ると言える。

6. 実現性の比較

単一プロセッサでの分散DB処理機能が既にあり、またマルチプロセッサ内でのローカルDBアクセスのためのプロセッサ間通信機能はサポートすることを前提に、実現性について考察する。

モデルM1はローカルアクセス時のAPを分散DB機能に置き換えた形態であり、実現は比較的容易である。また、モデルM2は単一プロセッサでの分散DB処理機能のままでの対応が可能である。これに対して、モデルM3では、通信プログラムでの通信路切換え、分散DB処理プログラムでの切換え前後プロセッサ間での情報の引き継ぎ処理等、大きな開発量が必要となり、実現性でモデルM1, 2に劣る。

7. まとめ

性能、移行性・拡張性、実現性の各比較から解るように、各モデルとも一長一短があり絶対的に有効な方式といえるものはない。

実装の際には実現が容易であることから、選択によりモデルM1, M2の両方式が可能できるようにし、対象となる業務内容に従って両モデルを使い分けることで対応するのが良い。

参考文献

- [1] Information technology - Open Systems Interconnection Remote Database Access Part 1: Generic model, service and protocol
- [2] 石川博道 他：データマネジメントシステムAXDM(6) - 分散データベースアクセスサービス機能 - 第36回情報全大
- [3] 山本裕 他：異種データベースシステム接続環境でのOSI-RDA実装方式について 第48回情報全大