分散データベースシステムにおけるResponse Timeと通信コストに関する最適データ配置法の検討

吉田　誠，水町恭子，大宅伊久雄，松下　温
沖電気工業株式会社

1. はじめに

データベースをネットワーク上に分散させ，統一的な処理を可能とする分散データベースシステムの利点についての記述は数多い（1, 10）。一般的には，集中型に比べて応答時間の向上，通信コストの削減，高信頼性等の利点があげられる。一方，上記利点とは逆にデータの分散／冗長性に伴い，Concurrncy control，Commitment control，Location transparencyの保証，最適データ配置等の問題が複雑化する。分散データベースシステム構築においては，上記Trade－offの関係をいかに効果的に設定するかが問題となる。

これまでに，分散制御に伴うオーバヘッドを軽減するために，いろいろなアプローチ

（ⅰ）アプリケーションに適したネットワークアーキテクチャの構成（4, 5, 9, 11）
（ⅱ）効率的な分散制御アルゴリズムの適用（4, 5, 8, 9, 12）
（ⅲ）アプリケーションに適したデータ配置法（1, 2, 7, 11）

が試みられている。

本研究会61－6においても（ⅲ）のアプローチにより，応答時間を主な評価項目とした場合の最適データ配置法について報告した。本稿では，同様のアプローチ（アプリケーション環境に適したデータ配置法）で，応答時間，通信コストの両面に着目し，いろいろなアプリケーション環境の下での両者の関係，及び最適なデータ配置法について，シミュレーションから得た結果を報告する。

尚，本シミュレーションではConcurrency Controlの方式として，タイムスタンプとロッキングに基づく，Conflict-driven restarts(3, 12)方式を採用している。

2. シミュレーションモデル

Fig. 1に，分散データベースシステムのモデルを示す。本モデルでは，分散データベースシステムは5つのサイトから構成されており，各サイト間は完全メッシュ状に全2重回線で接続されているものとする。

Fig. 2に各サイトのモデルを示す。各サイトはtransaction managerとConflict-resolution managerから構成され，各managerは以下の機能を持つ。

- Transaction manager……各サイトで発生したトランザクション及びそのサイトでの処理を終了したトランザクションに対して，次にアクセスすべきデータの存在するサイトを指示するmanagerであり，本managerによりユーザのlocation transparencyが保証される。アクセスすべきデータがそのサイトに存在しないトランザクションは，データの存在するサイトに移動するため転送待ちキューQti(iはサイト番号を意味する）に送られ転送の順序を待つ。
- Conflict-resolution manager……そのサイトに対象データが存在するトランザクション及び他サイトから対象データを求めて移動してきたトランザクションに対してConflict（衝突）の検査を行うmanagerであり，本managerは，当該トランザクションのアクセス対象とするportionに対してlock tableをチェックし，衝突が発生した時は，それぞれのプロトコル（DIE-WAIT system／WOUND-WAIT

system)*に従い処理を行う。

衝突の無かったトランザクションは，I/Oアクセスのためのキュー Qi に転送され，処理の順序を待つ（Fig. 2－A。尚，I／O制御方式はFIFO制御方式を採用した）。本 manager により WAIT を要求されたトランザクションは，Qw に入って対象データのロックが解除されるまで待つ（Fig. 2－W）。
又，rollback を要求されたトランザクションは，rollback 処理に移行する（Fig. 2－R）。

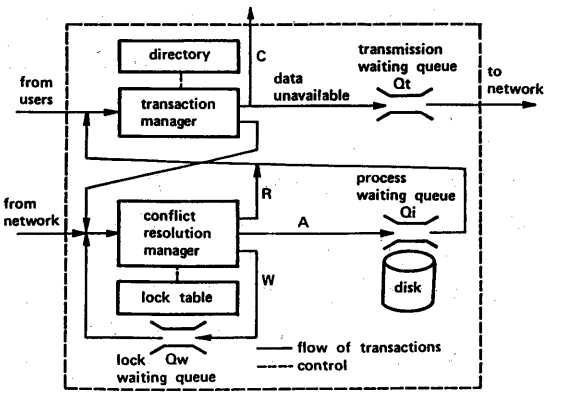


Fig.2 A model of individual site

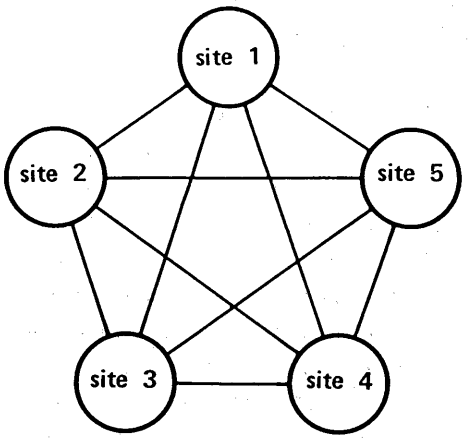


Fig. 1 A distributed database model

2. 1 User's view とデータ配置

本シミュレーションでは，アプリケーションにより決定される user's view とネットワーク上のデータ配置を独立なものとして設定し，各 user's view に対して最適なデータ配置形態をシミュレーションにより評価している。（例：Fig. 3 参照）

各サイトにおける view のケースを Fig. 4 に，又 user's view とは独立に設定した各サイト上のデータ配置法の各ケースを Fig. 5 に示す。

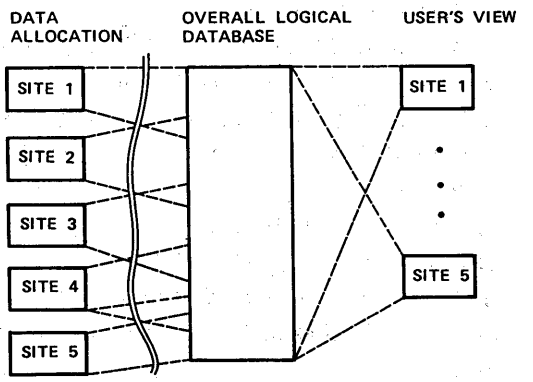
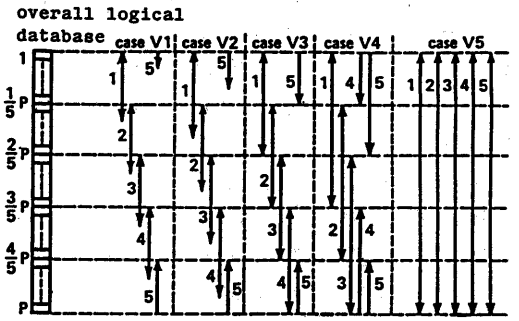


Fig.3 Example of data allocation

＊
- DIE–WAIT policy
  If Tg<Tp then WAIT else DIE
- WOUND–WAIT policy
  If Tg<Tp then WOUND else WAIT

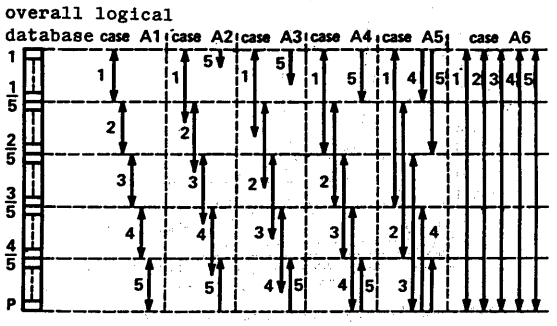
Tp：トランザクションP のタイムスタンプ
Tg：トランザクションQ のタイムスタンプ

尚，プロトコルの詳細については（2, 3 ）を参照のこと。



- The overall logical database is constructed from P number of portions. (A portion is a discrete unit of resources which may be physically allocated to sites.)
- 1～5 are site numbers.
- Vi (i=1, 5) represents each view of the five cases.

Fig.4 User's views



- Ai (i=1, 6) represents each data allocation of the six cases.

Fig.5 Data allocation methods

2.2 シミュレーションのパラメータ

本シミュレーションにおけるパラメータを以下に示す。

(1) 各サイトにおけるトランザクションの平均発生時間間隔（ＩＴ）。（指数分布）

(2) portion数（Ｐ）。分散データベースを構成するportion数であり，各portionはuser's viewとは独立に各サイトに配置される。

(3) User's view（Fig 4参照）。各サイトに属するユーザがアクセス可能なportionの集合。

(4) トランザクションの平均アクセスportion数（ＡＰ）。１つのトランザクションによりアクセスされるportionの平均であり，正規乱数より抽出される。

(5) データ配置（Fig 5参照）。各サイトに配置されるportionの集合。

(6) データの更新率（ＵＲ）。トランザクションは，read-only又はupdate-onlyの２種類が存在し，ＵＲは以下の式で与えられる。

$$UR=\frac{\text{update-only}\text{トランザクション}}{\text{全トランザクション}}$$

(7) 転送時間（Ｔ）。各サイト間通信に要する時間であり，メッセージの種類にかかわらず，constant 値Ｔをパラメータ値とする。ただし，同一回線が他メッセージ転送用に使用されている場合は，当該転送待ちキューQtiにおいて転送待ちとなる。

(8) 処理時間（Ｉ）。Ｉ／Ｏアクセス時間と処理時間の総和であり，constant値Ｉをパラメータ値とする。ただし，ロックテーブル等のディレクトリは主記憶上に存在し，かつＩ／Ｏアクセス以外の処理時間は無視できる程小さいものと仮定している。

(9) 転送コスト（Ｃ）。各サイト間での通信に要する１メッセージあたりの転送コスト。

2.3 収集量

次の諸量を測定した。

(1) 平均応答時間。トランザクション入力から答を得るまでの平均時間。

(2) 平均通信コスト。トランザクション入力から答を得るまでに要する転送コストの総和の平均。本モデルでは，通信コストは転送されるメッセージ数に比例すると仮定している。よって，平均通信コストは以下の式でで与えられる。

$$TC=\frac{\left(\begin{array}{c}\text{すべてのトランザクションによって}\\ \text{転送されたメッセージの数}\end{array}\right)}{\text{全トランザクション}}*C$$

(3) Rollback 数

3. シミュレーション結果

前述のモデルに従い，タイムスライス法によりシミュレーションを行なった結果を以下に述べる。

3.1 シミュレーション1：（DIE－WAIT systemとWOUND－WAIT systemの性能比較）

本シミュレーションにおける設定条件を以下に示す。

（条件）

(1) 各サイトのuser's view ：Fig.4のV5とする。
(2) データ配置：Fig.5のA1及びA6
(3) データ更新率：100%
(4) T：I＝1：1
(5) portion数：P＝90

Fig. 6にWOUND-WAIT systemにおける平均発生時間間隔と平均応答時間及び平均通信コストの関係を示す。又，Fig. 7にDIE-WAIT systemにおける平均発生時間間隔と平均応答時間及び平均通信コストの関係を示す。

両システムを比べてみると，トランザクションが頻繁に発生する状況下（100＜IT＜200）では，WOUND-WAIT systemの方が，通信コスト，応答時間の両面において優れている。更に，トランザクションの発生が密に（IT＜100）になると両システムとも通信コスト，応答時間の両面において急激に悪化している。又，トランザクションの発生が疎（IT＞200）の場合は，両システムとも通信コスト，応答時間の両面において，ある一定の値に収束していることが観察される。

Rollback数において両システムに相違があるのは，両システムのプロトコルにおけるconflictの解消法に起因するものである。（1，2）

本シミュレーションにおいては，データ更新率を100%に設定し，両システムの応答時間，通信コスト両面からの評価を試みた。（データ配置形態A1の方がA6より優れているのは，当該データ更新率の値に依存するものである。）

次のシミュレーションでは，データ更新率を変化させた場合の通信コスト及び応答時間の変化をWOUND-WAIT systemを採用して観察する。

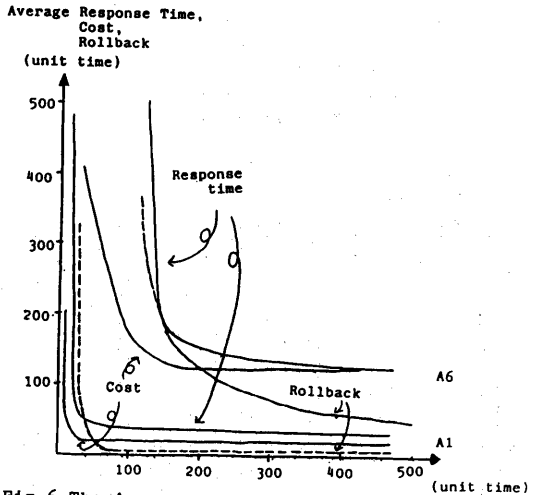


Fig.6 The Average response time and Cost of WOUND-WAIT SYSTEM

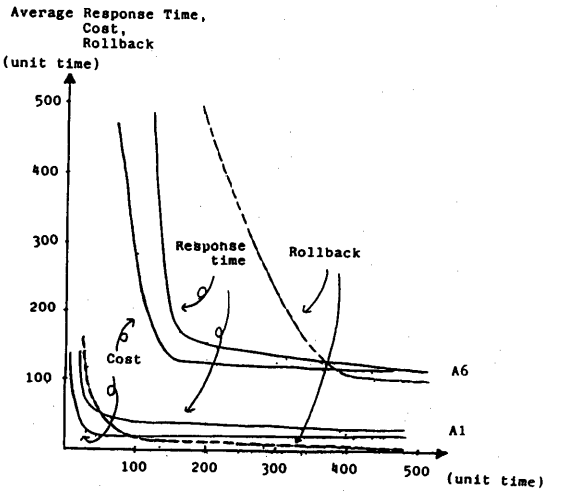


Fig.7 The Average response time and Cost of DIE-WAIT SYSTEM

3.2 シミュレーション2：（データ更新率の変化に伴う，応答時間及び通信コストの変化）

重複データを配置する場合には，ユーザがアクセス可能なデータは全て自サイトに置く形態が応答時間の面で最適であるという結果(1，2)に従い，当該データ配置とdisjoint なデータ配置形態を設定し，データ更新率及びuser's viewを変化させ，応答時間，通信コストの関係を測定した結果を示す。

以下に本シミュレーション上の条件を示す。

（条件）

(1) データ配置：user's view とデータ配置が一致する形態，及び disjoint なデータ配置形態を適用。

(2) I：T＝1：1

(3) トランザクションの平均到着時間間隔は，WOUND-WAIT systemの適用可能な範囲とし，平均応答時間及び通信コストが増加し始める点に設定。

Fig. 8 から Fig.12 にデータ更新率の変化に伴う，応答時間と通信コストの関係を user's view を変化させて測定した結果を示す。

各グラフ上で，交点Aよりデータ更新率の高い領域では，disjointなデータ配置の方が応答時間において優れており，一方，更新率の低い領域では重複データの形態が優れていることを示している。（グラフ上のA点を応答時間における分岐点，又B点を通信コストにおける分岐点と呼ぶ。）しかし，当該分岐点のデータ更新率がそのまま通信コストに対してのデータ配置の分岐点(B)に適用されるとは言えない。つまり，データ配置形態は，通信コスト重視の場合と応答時間重視の場合では異なることを示している。

User's view の変化に伴う，これら分岐点の変化を Fig.13 に示す。当該グラフにおいて領域Aで示される部分は，応答時間，通信コストの両面においてdisjoint なデータ配置形態が有利な領域であり，領域Bは，重複データ配置（user's view と一致するデータ配置形態）が通信コスト面においては有利となり，応答時間に関しては，disjoint な配置が有利となることを示している。一方，領域Cは，重複データ配置が応答時間に関しては有利であり，disjoint な配置が通信コストにおいては有利な領域を示す。同様に，領域Dは，両要素（応答時間及び通信コスト）に対して重複データ配置の有利な領域を示している。

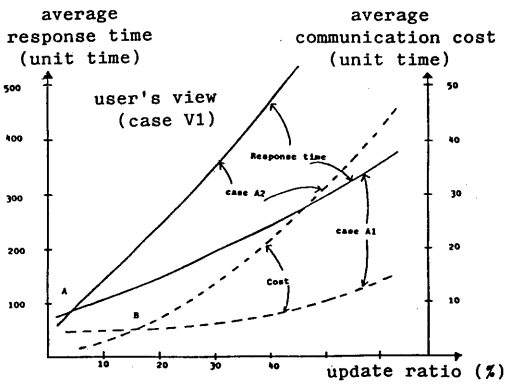


Fig.8 Response time and Communication cost v.s. Update ratio

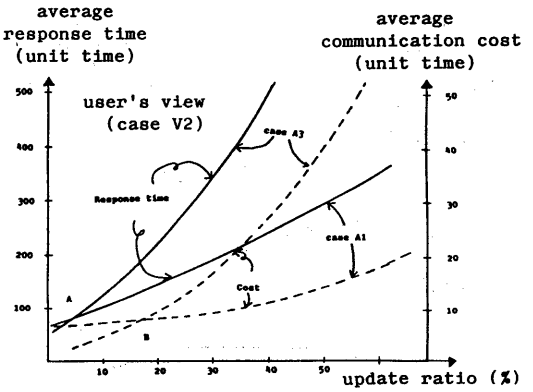


Fig.9 Response time and Communication cost v.s. Update ratio

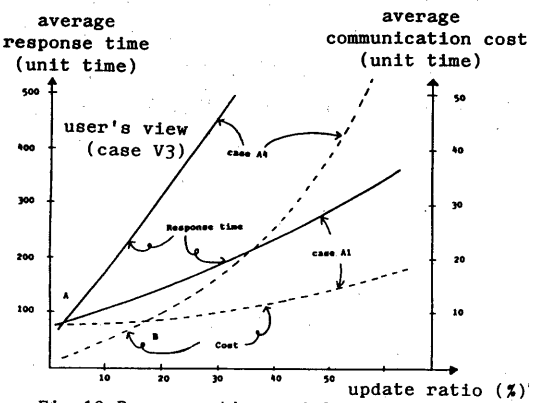


Fig.10 Response time and Communication cost v.s. Update ratio

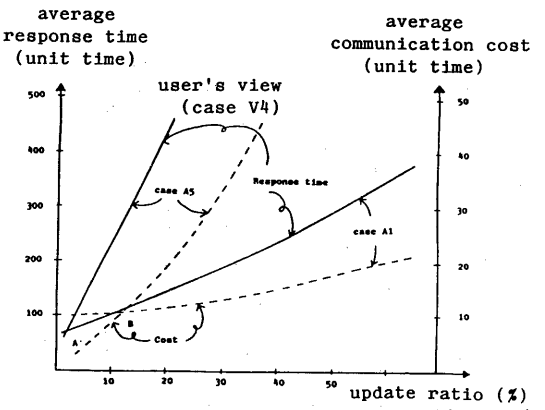


Fig.11 Response time and Communication cost v.s. Update ratio

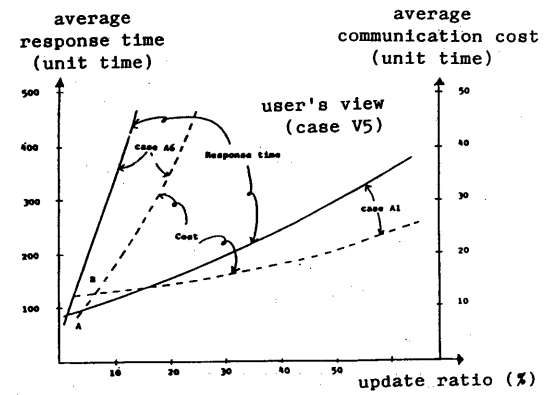


Fig.12 Response time and Communication cost v.s. Update ratio

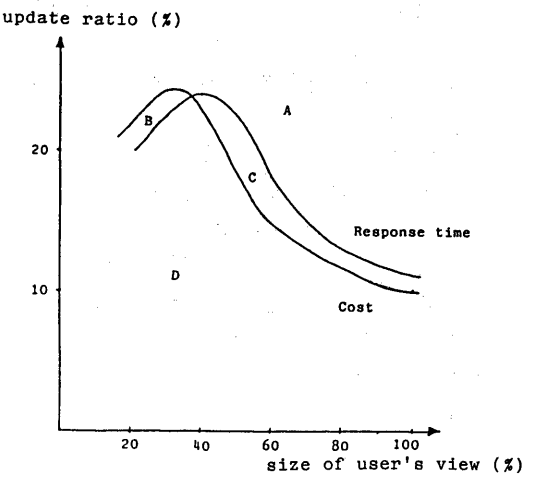


Fig.13 The shift behavior of the curve of threshold points (IT=150)

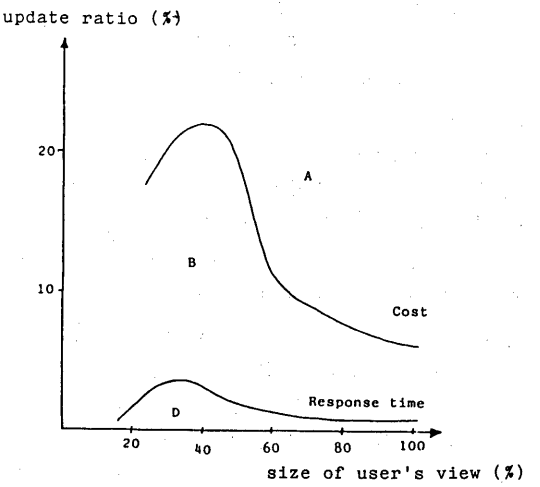


Fig.14 The shift behavior of the curve threshold points (IT=10)

3.3 シミュレーション3：（平均到着時間の変更に伴う分岐点の変化）

シミュレーション2では，データ更新率の変化に伴う分岐点の変化を観察した。本シミュレーションでは，平均発生時間間隔を変化させた場合の分岐点の変化の様子を示す。

Fig.14にトランザクションの平均到着時間を短かくした場合(IT=10)の分岐点をプロットした結果を示す。トランザクションの平均発生時間間隔が短かくなると，重複データ配置が有利となる領域が減少していることが観察できる。（平均発生時間間隔が短かくなるにしたがって，重複データ配置を有するアプリケーション環境が制限されることを示す。）

一方，Fig 13とFig 14を比較すると，トランザクションの平均発生時間間隔は，応答時間に関して非常に重要な影響を及ぼす要因となっているが，通信コスト面での影響は余りないことを示している。又，重複データ配置の形態が有利な環境は，user's view(サブスキーマ)が全体view(スキーマ)の30%から50%ぐらいであることを示している。

4. 結論

本稿では，concurrency controlの方式としてconflict-dviven pestartsを採用し，応答時間，通信コストの両面から2つのプロトコルの性能を定量的に評価すると同時に，アプリケーションによって決定されるuser's viewとそれに応じた最適データ配置法をシミュレーション結果から検討した。

以下にそのまとめを示す。

(1) トランザクションが頻繁に発生する時，つまり，平均到着時間が短かい時には，WOUND-WAIT systemの方が，通信コスト，応答時間の両面において優れている。しかしながら，平均発生時間間隔がある点以上に短かくなると，両システムとも急激に変化する。

(2) 重複データを配置する形態が有利な場合は，各サイトのviewとHomogeneousなデータ配置形態が応答時間に関して最適である。（1，2）

(3) 重複データが有利な領域は，（平均応答時間，通信コストの両面において）user's viewが全体viewの30%～50%において最適となる。

(4) トランザクションの平均発生時間間隔の応答時間に及ぼす影響は大であるが，通信コストへの影響は余りない。つまり，応答時間を重要視する分散データベースシステム構築においては，トランザクションの平均発生時間間隔は重要なperformance要因となり得るが，応答時間を重要視する分散データベースシステムの構築においては余り重要な要因とはならないことを意味している。

5. 参考文献

1) Y.Matsushita, M.Yoshida, A.Wakino. Allocation Schemes of Multiple Copies of Data in Distributed Database Systems. The 3rd International Conference on distributed Computing Systems. October, 1982.

2) 吉田，脇野，松下．分散データベースシステムに於ける最適データ配置法の検討．分散データベースシステム研究会16－6（19821118）

3) Rosenkrantz, R.E.Stearns, P.M.Lewis. A system level concurrency control for a distributed database systems. ACM trans on Database Systems. Vol.3, No.2, June, 1978.

4) P.A.Bernstein, N.Goodman. Concurrency control in Distributed

Database Systems. Computing surveys, Vol.13, No.2, June, 1981.

5) H.Yamazaki, S.Hikita, I.Yoshida, S.Kawakami, Y.Matsushita. A Hierarchical Structure for Concurrency Control in a Distributed Database System. Sixth Data communication Synposium, November, 1979.

6) Y.Matsushita, M.Yoshida, A.Wakino, L.T.Beng. A Safe and Fast Concurrency Model for Query-Oriented Fully Redundant Distributed Databases. International Conference on Communications 83, June, 1983.

7) M.Stonebraker. Concurrency Control and Consistency of Multiple Copies of Data in distributed INGRES. IEEE Trans on Software Engineering, Vol.3, May, 1979.

8) Kohler, W.H. A survey of techniques for synchronization and recovery in decentralized computer systems. Computing Surveys, Vol.13, No.2, June, 1981.

9) Hector Garcia-Molina. Performance comparison of two update algorithms for distributed databases. 3rd Berkely Workshop, 1978.

10) J.Martin. Design and Strategy for distributed Data Processing. Prentice-Hall 1981.

11) Y.Matsushita, H.Yamazaki, S.Hikita, I.Yoshida. Cost Evaluation of directory Management Schemes for Distributed Database Systems. SIGMOD 1980.

12) P.A.Bernstein, N.Goodman. Approaches to concurrency control in distributed database. National Computer Conference, 1979.