

分散システムにおけるデータアクセスの 待ち行列モデルによる性能解析方法の検討

宮西 洋太郎†, 川口 昇‡, 佐藤 文明‡, 水野 忠則‡
† 三菱電機 ‡ 静岡大学工学部

分散システムにおけるデータベースアクセス時には、並行処理制御や一貫性維持のために、ロック操作が用いられる場合が多い。複製を有する場合の更新処理は、全複製へのロック操作の後に全複製の更新処理が行なわれるのが一般的である。全複製へのロック操作および書込み処理は応答時間に影響を与える。本論文では、分散システムをモデル化し、待ち行列解析により近似的にデータアクセスの応答時間を求めた。その概略は、ロックを論理的処理とみなし、実際のデータアクセスを物理的処理とみなし、両者の階層をモデル化した。また並列及び直列のサービス時間の分布を求めて解析に使用した。今回は待ち行列モデルの一部についてシミュレーションを行った。

Analysis of Data Access Performance in Distributed Systems and some Simulations

Yohtaro Miyanishi*, Noboru Kawaguchi+, Fumiaki Sato+, and Tadanori Mizuno+

* Mitsubishi Electric Corp.
7-10-4 Nishigotanda Shinagawa, Tokyo, 141

+ Shizuoka University
3-5-1 Johoku, Hamamatsu, 432, Japan

LOCK operations are usually used in the case of updating the distributed database. When the database has several replicas, the data may be updated after all replicas have been locked. Locking and updating of all replicas tends to degrade the performance. We analyze the response time for accessing data by using a queueing model. We show some results of simulations.

1. はじめに

分散システムのデータアクセス、特にトランザクション処理などでは通常、読取り時には読取りロック、書込み時には書込みロックの後に、アクセスが行われる。複製を有する場合には、全複製へのロック操作及び読込み・書込み処理はデータ更新応答時間性能低下の原因の一つとなりうる。本発表では、ロック操作及び読込み・書込み処理が応答時間に与える影響について、待行列により解析し、近似的に応答時間を求める方法を示す。

2. 関連する研究

分散データの更新処理において、データの並行処理制御やデータ相互間、複製間の一貫性維持のために、ロック操作を用いた、2PL方式、2PC方式が実用化されている。

分散システムの性能解析の多くは待行列モデルを用いて行われている[1][2]が、ロックとの関連

については、あまり検討がなされていない[3]。

待ち行列を主な解析手段としない研究もある。例えば文献[4]では、統計的な観測値から性能評価式を導出する方法であるが、全体のシステム構造そのものを設計する立場からは、適用しがたい。文献[3]は評価指数としてコストを設定し、コストを最小にする設計方法である。

本発表の3. では、解析の準備を行い、4. では、「全複製へのロックを行ない、全複製のアクセスを行ない、その後ロックを解除する」という単純化したモデルで、アクセス応答時間性能に焦点を絞り待行列モデルによる性能解析を行ない、シミュレーションを行う。前回までの発表[5][6]からの進展は、①タイムスタンプ付き待ち行列のシミュレーションの実施、②簡略した状況でのシミュレーションの実施、③自サイトに複製が存在する場合の考慮、の3点である。

3. 対象データのモデル

データアクセスは、読取りロック及び書込みロックによって排他制御される場合が多いが、ここでは、書込みロック種類のみとして近似する。

3. 1 ロックとデータアクセスとの関係

ここではロックとデータアクセスを論理的サーバと物理的サーバという関係で整理を行う。ここで、サーバとは待ち行列理論での用語である。

(1) 二つのアクセスグループの場合

まず、二つの論理的なアクセスのグループがあり、物理的には一つのサーバ（ディスク、CPU）、これを物理的サーバと称する、を共有しているモデルを考える。グループごとにロックを行い、ロックが確保されて物理的サーバに処理要求を行う。物理的サーバはFIFOにて処理を行い、処理が終われば、該当するグループのロックを解除する。これを図1に示す。それぞれの到着率（1/秒）、平均サービス時間（秒）を、

$\lambda_1, s_1, \lambda_2, s_2$ とする。

論理的アクセスの処理部分をここでは、論理的サーバと称する。論理的サーバの前段には論理的待ち行列、物理的サーバの前段には物理的待ち行列をもつものとする。物理的待ち行列には、ロックが確保された要求だけが入るので、同一グループの要求は高々一つ存在するのみである。

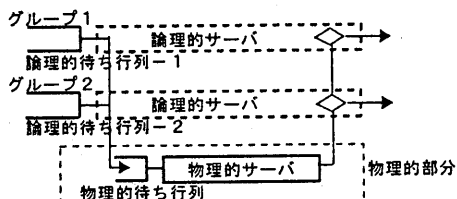


図1 二つのアクセスグループのロックモデル

論理的待ち行列に待ち中の要求の数を含めたモデル全体の状態を考え、この状態遷移をマルコフ過程として、定常状態での各状態の確率を計算することが考えられるが、待ち中の要求数に制限はなく、状態の数が多く複雑となる。

そこで、近似として物理的部分（待ち行列及びサーバ）のみに着目し、5つの状態の近似的な確率を求め、応答時間 T_1, T_2 を計算する。この方法がほぼ妥当であることがシミュレーションによって示されている。[7]

到着率の小さい方が物理的待ち行列の中に同種の要求が存在する確率は小さく、現在処理中の次に処理順序が確保されやすい。従って、結果的に到着率の小さい方が処理順序が優先されて、応答

時間が小さく（速く）なる。

(2) 複数のアクセスグループの場合

上記の近似方法でもグループの数が多くなるに従って物理的部分（サーバ、待ち行列）の状態の数が階乗で増加し、計算が困難となる。

そこで、図2に示す近似的モデルを考える。

グループ g の到着率、平均処理時間、処理時間の分散を $\lambda g, s g, \sigma g^2$ とする。

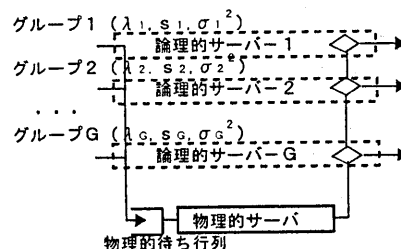


図2 複数アクセスグループの近似的モデル

アクセスのグループ数を G とすると、以下の式により、物理的待ち行列における待ち時間 T_w 及びグループ g の応答時間 T_{qg} が求められる。

$$\lambda = \sum_{g=1}^G \lambda g, \quad \rho = \sum_{g=1}^G \lambda g \cdot s g,$$

$$S = \sum_{g=1}^G \frac{\lambda g}{\lambda} \cdot s g, \quad \sigma_s^2 = \sum_{g=1}^G \frac{\lambda g}{\lambda} \cdot \sigma g^2.$$

各グループの処理時間は指数分布としたが、合成された処理時間の分布は、指数分布とはならない。従って、M/G/1のポラチェックヒンチンの公式[7]を用いて、次式が得られる。

$$T_w = \rho S (1 + \sigma_s^2 / S^2) / (2(1 - \rho)),$$

$$T_{qg} = T_w + s g.$$

このモデルでは上式のように、応答時間のグループによる相違は処理時間 $s g$ の相違のみとなり、前述の、物理的待ち行列における到着率の小さいグループが優先される現象を表現できていない。しかし、これは「処理の優先順序は、要求発生頻度（到着率）の相違に依存するものではなく、明示的な優先度の指定によって行うべきである。」という考え方に整合している。

これは、物理的待ち行列のメカニズムを次のように変更することにより近似的に実現できる。

「各グループの要求に受付時にタイムスタンプを持たせ、物理的待ち行列に要求が到着するとき、そのタイムスタンプにより要求発生順に待ち行列の中を並び替える」。グループ数が4の場合の改修後のシミュレーションの結果を表1に示す。

表1 従来型とタイムスタンプ型のシミュレーション結果

到着率	従来型	タイムスタンプ型
$\lambda_1=0.3065$	T1=18.812	T1=16.009
$\lambda_2=0.3065$	T2=15.876	T2=15.798
$\lambda_3=0.3065$	T3=20.844	T3=15.912
$\lambda_4=0.0306$	T4= 3.658	T4=15.037
$\lambda_1=0.2900$	T1= 8.454	T1= 9.739
$\lambda_2=0.2900$	T2= 9.084	T2=10.024
$\lambda_3=0.2900$	T3= 8.920	T3= 9.822
$\lambda_4=0.0300$	T4= 3.324	T4= 8.776
$\lambda_1=0.2581$	T1= 4.783	T1= 5.548
$\lambda_2=0.2581$	T2= 4.978	T2= 5.667
$\lambda_3=0.2581$	T3= 4.782	T3= 5.805
$\lambda_4=0.0258$	T4= 2.934	T4= 5.363
$\lambda_1=0.1290$	T1= 1.680	T1= 1.627
$\lambda_2=0.1290$	T2= 1.679	T2= 1.615
$\lambda_3=0.1290$	T3= 1.733	T3= 1.736
$\lambda_4=0.0130$	T4= 1.709	T4= 1.730

上表よりタイムスタンプ型では、各グループの応答時間は個別の到着率に、ほとんど依存しないことが分かる。従って、このような待ち行列メカニズムを前提とすれば、グループごとの到着率の不均質性については、検討の対象外にできる。

次に、システム全体としての性能解析には、論理的サーバの待ち行列ネットワークにより、応答時間を算定することが必要である。このために、物理的サーバの能力がどの程度、論理的サーバに配分されるかを考える。各論理的サーバはM/M/1であると考え、その応答時間が前述の値になるようにサーバ能力が配分されていると考え、グループgに配分されたサーバの能力Cgは、次の応答時間の式から求められる。

$$T_{qg} = (s_g / C_g) / (1 - \lambda_g (s_g / C_g))$$

$$C_g = \lambda_g \cdot s_g + s_g / T_{qg}$$

この配分されたサーバ能力で、論理的グループそれぞれの処理時間 $s_g' = s_g / C_g$ を求める。これにより物理的サーバを共有しているという事実から、近似的に離れ、図3のように、各グループに仮想的に専属のサーバが接続されているものとみなせる。このサーバ能力配分計算を行う部分を「計算パッケージ1」と称することとする。

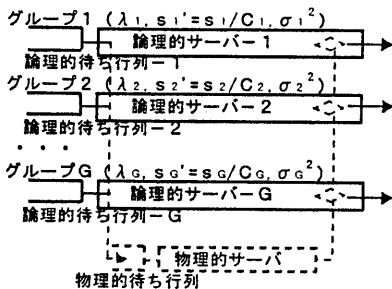


図3 論理的サーバによる近似モデル

3. 2 直列、並列合成サーバの処理時間

図4で表されるモデルを考える。ここでは、並列数、直列数は、高々3とする。また並列処理の完了は、AND型とOR型があるものとする。処理の各要素は平均処理時間が $1/\mu_{ij}$ の指数分布に従うものとする。

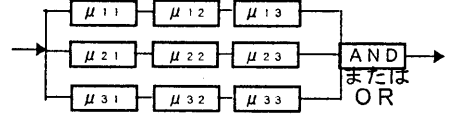


図4 直列、並列合成サーバ

このモデルの平均処理時間 \bar{t} と処理時間の分散 σ_s^2 を求める。

$$\bar{t} = \sum_{i=1}^3 \sum_{j=1}^3 \sum_{k=1}^3 C_{1i} C_{2j} C_{3k} (1/\mu_{1i} + 1/\mu_{2j} + 1/\mu_{3k} - 1/(\mu_{1i} + \mu_{2j}) - 1/(\mu_{2j} + \mu_{3k}) - 1/(\mu_{3k} + \mu_{1i}) + 1/(\mu_{1i} + \mu_{2j} + \mu_{3k}))$$

$$\sigma_s^2 = \bar{t}^2 - (\bar{t})^2 + 2/(\mu_{1i} + \mu_{2j} + \mu_{3k})^2 - 2/(\mu_{1i} + \mu_{2j})^2 - 2/(\mu_{2j} + \mu_{3k})^2 - 2/(\mu_{3k} + \mu_{1i})^2$$

ただし、

$$C_{m1} = \mu_{m2} \mu_{m3} / ((\mu_{m1} - \mu_{m2})(\mu_{m1} - \mu_{m3}))$$

$$C_{m2} = \mu_{m3} \mu_{m1} / ((\mu_{m2} - \mu_{m1})(\mu_{m2} - \mu_{m3}))$$

$$C_{m3} = \mu_{m1} \mu_{m2} / ((\mu_{m3} - \mu_{m2})(\mu_{m3} - \mu_{m1}))$$

ただし $m = 1, 2, \text{又は} 3$

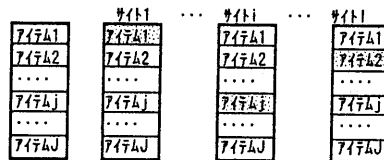
ほぼ同様にOR型の平均処理時間と分散も求められるが、ここでは省略する。平均処理時間と分散が求まるので、M/G/1のポラチェックヒンチンの公式から待ち時間も計算できる。

上記を 1×1 から 3×3 まで対応できるように拡張して「計算パッケージ2」と称する。

4. システムのモデル

4. 1 対象データとアクセスパラメータ

検討の対象となるデータベースを図5に示す。対象の分散システムは、LAN型のネットワークにI個のサイトが接続しており、通信時間は、どのサイトからどのサイトへの時間も同じとする。



例えば、ハッチング部分には複製が存在する

図5 対象データベース

ここでは、データのアクセスは読取り操作及び更新操作を検討する。更新操作は旧データを読取

り、新データを作成し、その後書込む操作とする。
 対象となるデータは、データアイテムj単位にアクセス（読取り、更新）され、またアクセス要求は地域的に分散した複数サイトの内の一つに到着し、そのサイトiでアクセスの管理（協調者の役割）がなされるものとする。アイテムjの読取り又は更新を各アクセスグループ（AGと略す）とする。

全体の動作を定める次の変数を定義する。

$\lambda_{r_{ij}}$: サイトiにデータアイテムjの読取り要求が到着する到着率(1/秒) ,

$\lambda_{u_{ij}}$: サイトiにデータアイテムjの更新要求が到着する到着率(1/秒) ,

d_1 : サイトiからサイトi'までのタイプ1通信時間, (タイプ1:制御情報のみ)

d_2 : サイトiからサイトi'までのタイプ2通信時間, (タイプ2:データを含む)

X_{ij} : 複製存在変数(二値変数) ,

$X_{ij}=1$ ならばサイトiにアイテムjの複製あり

$X_{ij}=0$ ならばサイトiにアイテムjの複製なし

$s_{r_{ij}}$: サイトiに複製がある場合の、この複製単独の平均読取り時間(秒) ,

$\mu_{r_{ij}}$: 読取り処理率, $\mu_{r_{ij}}=1/s_{r_{ij}}$,

$s_{w_{ij}}$: サイトiに複製がある場合の、この複製単独の平均書込み時間(秒) ,

$\mu_{w_{ij}}$: 書込み処理率, $\mu_{w_{ij}}=1/s_{w_{ij}}$,

これらをパラメータとして、各サイトからのAGごと(さらに細分して複製の存在するサイト、しないサイトごと)の応答時間を求めることが、ここでの目標である。

4. 2 物理的サーバへの負荷の集計

サイトiに、データアイテムjのアクセス要求が1回発生すると、アイテムjを保有する複製保持サイトi, すなわち物理的サーバに読取り、更新要求それぞれにつき、次の負荷がかかる。

$$s_{r_{ij}} , s_{r_{ij}} + s_{w_{ij}} .$$

従って、サイトiから見て、各アイテムjのアクセスグループごとの負荷はつぎようになる。

・読取り要求による負荷,

jの到着率: $X_{ij} \lambda_{r_{ij}}$, 処理時間: $s_{r_{ij}}$.

・更新要求による負荷(読取り, 書込み) ,

jの到着率: $X_{ij} \lambda_{u_{ij}}$,

処理時間: $s_{r_{ij}} + s_{w_{ij}}$.

ただし、 $\lambda_{r_{ij}} = \sum_{i=1}^n \lambda_{r_{ij}}$, $\lambda_{u_{ij}} = \sum_{i=1}^n \lambda_{u_{ij}}$.

これを、「計算パッケージ1」を用いて、AGごとに能力の分割を行う。

その結果を $C_{r_{ij}}$, $C_{u_{ij}}$ とする。

これを応答時間計算のときの論理的サーバの能力として、処理時間を再設定する。

$$s_{r_{ij}}' = s_{r_{ij}} / C_{r_{ij}} ,$$

$$s_{u_{ij}}' = s_{u_{ij}} / C_{u_{ij}} , s_{w_{ij}}' = s_{w_{ij}} / C_{u_{ij}} .$$

4. 3 通信システムの負荷の集計

AGごとのネットワークの負荷を求める。jについての複製の数を X_j とする。

ただし、 $X_j = \sum_i X_{ij}$ である。

要求受付サイトに複製があれば、通信は不要であり、その分だけ通信への到着率は減少する。

・読取り要求による負荷

ネットへの到着率 λ_{nr_j} は、

$$\lambda_{nr_j} = \lambda_{r_j} X_j - \sum_i \lambda_{rij} X_{ij} .$$

ネットの処理時間 s_{nr_j} は、

$$s_{nr_j} = (5d_1 + d_2) X_j .$$

処理時間の分散 $\sigma^2_{nr_j}$ は、

$$\sigma^2_{nr_j} = (5d_1^2 + d_2^2) X_j .$$

・更新要求による負荷

ネットへの到着率 λ_{nu_j} は、

$$\lambda_{nu_j} = \lambda_{u_j} X_j - \sum_i \lambda_{uij} X_{ij} .$$

ネットの処理時間 s_{nu_j} は、

$$s_{nu_j} = (6d_1 + 2d_2) X_j .$$

処理時間の分散 $\sigma^2_{nu_j}$ は、

$$\sigma^2_{nu_j} = (6d_1^2 + 2d_2^2) X_j .$$

上記の負荷を、「計算パッケージ1」を用いて、AGごとにネットワークの能力を分割する。

4. 4 待ち行列モデル

図6から図9に待ち行列モデルを示す。

次の手順によって、データアイテムjのサイトiからの読取り、更新の応答時間を求める。

(1) 共有リソースのAGへの能力分割

前述の方法でAGごとに共有リソース(ファイルサーバとしてのサイト及びネットワーク)の能力を分割する。

(2) AGごとに平均処理時間、分散を計算

AGを、要求受付サイトiにデータアイテムjの複製が存在する場合、しない場合に細分して扱う。複製が存在する場合は、以下の1)から4)までの部分での通信の部分の一部(宛先が自己サイトの分)が減少する。

1) 要求受付サイトにおける到着率

複製が存在する場合 ($X_{ij}=1$) の到着率,

読取りAG: $\lambda_{1r_j} = \sum_i X_{ij} \lambda_{rij}$,

更新AG: $\lambda_{1u_j} = \sum_i X_{ij} \lambda_{uij}$.

複製が存在しない場合 ($X_{ij}=0$) の到着率,

読取りAG: $\lambda_{2r_j} = \sum_i (1 - X_{ij}) \lambda_{rij}$,

更新AG: $\lambda_{2u_j} = \sum_i (1 - X_{ij}) \lambda_{uij}$.

2) ロック要求(ロック取得前)

上記と同様に、要求受付サイト i にデータアイテム j の複製の有無により場合を細分する。複製をもたない場合は、図7により、直列数は2、並列数は複製数(最大3)のAND型並列処理である。この合成されたサーバの処理時間と分散を「計算パッケージ2」により求め、この処理時間と分散から $M/G/1$ のポラチェックヒンチンの公式から、この部分の読取り、更新AGの応答時間それぞれ、 $T2qr1$ 、 $T2qu1$ が求められる。

要求受付サイトにアイテム j の複製をもつ場合は、上記の並列数が1減少し、この部分の読取り、更新AGの応答時間それぞれ、 $T1qr1$ 、 $T1qu1$ が求められる。

3) 読込処理

ロック以降の処理時間と分散を求める。要求受付サイトにアイテム j の複製をもたない場合は、図8に示すように、

- ・ 3直列、複製数OR型並列部分(平均処理時間と分散を $S2r1_j$ 、 $\sigma 2r1_j^2$ とする)及び、
 - ・ 2直列、複製数AND型並列部分(平均処理時間と分散を $S2r2_j$ 、 $\sigma 2r2_j^2$ とする)、
- の直列であり、それぞれ平均処理時間及び分散を「計算パッケージ2」にて求める。

要求受付サイトにアイテム j の複製をもつ場合には、通信の宛先が一方向減少し、上記のそれぞれが、 $S1r1_j$ 、 $\sigma 1r1_j^2$ 、 $S1r2_j$ 、 $\sigma 1r2_j^2$ となる。

読込処理全体の平均処理時間及び分散は二つの部分の和である。

複製がある場合、

$$S1r_j = S1r1_j + S1r2_j, \\ \sigma 1r_j^2 = \sigma 1r1_j^2 + \sigma 1r2_j^2,$$

複製がない場合、

$$S2r_j = S2r1_j + S2r2_j, \\ \sigma 2r_j^2 = \sigma 2r1_j^2 + \sigma 2r2_j^2.$$

4) 更新処理

ロック以降の処理時間と分散を求める。要求受付サイトに複製が存在しない場合は、図9に示すように、

- ・ 3直列、複製数OR型並列部分(平均処理時間と分散を $S2u1_j$ 、 $\sigma 2u1_j^2$ とする)、
 - ・ 3直列、複製数AND型並列部分(平均処理時間と分散を $S2u2_j$ 、 $\sigma 2u2_j^2$ とする)及び、
 - ・ 2直列、複製数AND型並列部分(平均処理時間と分散を $S2u3_j$ 、 $\sigma 2u3_j^2$ とする)、
- の三つの部分の直列であり、それぞれ平均処理時間と分散を「計算パッケージ2」により求める。

要求受付サイトに複製が存在する場合は、前述のように通信の一部が減少し、上記がそれぞれ、 $S1u1_j$ 、 $\sigma 1u1_j^2$ 、 $S1u2_j$ 、 $\sigma 1u2_j^2$ 、 $S1u3_j$ 、

$\sigma 1u3_j^2$ となる。

更新処理全体の平均処理時間及び分散は三つの部分の和である。

複製がある場合、

$$S1u_j = S1u1_j + S1u2_j + S1u3_j, \\ \sigma 1u_j^2 = \sigma 1u1_j^2 + \sigma 1u2_j^2 + \sigma 1u3_j^2.$$

複製がない場合、

$$S2u_j = S2u1_j + S2u2_j + S2u3_j, \\ \sigma 2u_j^2 = \sigma 2u1_j^2 + \sigma 2u2_j^2 + \sigma 2u3_j^2.$$

(3) AGごとに応答時間を求める

同一データアイテムのアクセスがポアソン過程で到着すると仮定する。これにより図6に示すロック待ち行列が生じる。読取りと更新が同一の論理的リソースすなわちロックを使用するので、待ち時間 Tw_j は共通で両者の合成された、平均処理時間、分散を $M/G/1$ のポラチェックヒンチンの公式に適用して求めることができる。

$TW_j = \rho_j S_j (1 + \sigma_j^2) / (2(1 - \rho_j S_j))$ 。ただし、

$$\rho_j = \lambda 1r_j S1r_j + \lambda 1u_j S1u_j \\ + \lambda 2r_j S2r_j + \lambda 2u_j S2u_j, \\ S_j = (\lambda 1r_j S1r_j + \lambda 1u_j S1u_j + \lambda 2r_j S2r_j \\ + \lambda 2u_j S2u_j) / (\lambda 1r_j + \lambda 1u_j + \lambda 2r_j + \lambda 2u_j), \\ \sigma_j^2 = (\lambda 1r_j \sigma 1r_j^2 + \lambda 1u_j \sigma 1u_j^2 \\ + \lambda 2r_j \sigma 2r_j^2 + \lambda 2u_j \sigma 2u_j^2) \\ / (\lambda 1r_j + \lambda 1u_j + \lambda 2r_j + \lambda 2u_j).$$

従って、データアイテム j の複製のある場合、ない場合の読取りAGの応答時間 $T1qr_j$ 、 $T2qr_j$ 、更新AGの応答時間 $T1qu_j$ 、 $T2qu_j$ はそれぞれ、

・ 複製がある場合、

$$\text{読取りAG: } T1qr_j = T1qr1 + Tw_j + S1r_j, \\ \text{更新AG: } T1qu_j = T1qu1 + Tw_j + S1u_j,$$

・ 複製がない場合、

$$\text{読取りAG: } T2qr_j = T2qr1 + Tw_j + S2r_j, \\ \text{更新AG: } T2qu_j = T2qu1 + Tw_j + S2u_j.$$

となる。

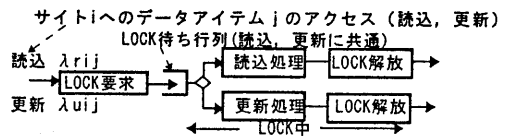


図6 全体の処理

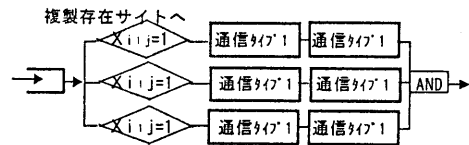


図7 ロック要求の処理

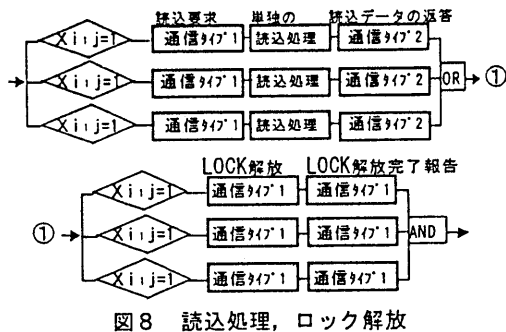


図8 読込処理, ロック解放

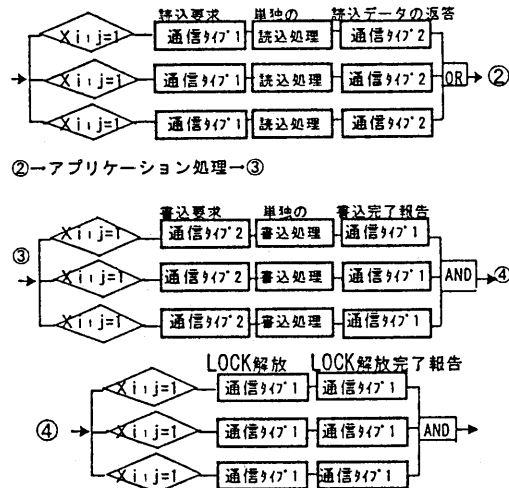


図9 更新処理, ロック解放

(4) シミュレーション結果

上記の計算方式について、簡略化したケースでシミュレーションを行った。その結果を図10に示す。

簡略ケースは、通信時間=0、データアクセスは書込みのみ（更新の場合であって、読込なし）とした。swi_{ij}は同一（=1秒）、λ_{uij}は同一値で変化させた。

ケース1は、サイト数I=3、アイテム数J=4、

ケース2は、サイト数I=3、アイテム数J=6。

図10のグラフにより、ここで述べた計算方式は概略妥当であると判断できる。

5. おわりに

以上をまとめると、

- (1) タイムスタンプ付きの待ち行列について、シミュレーションにより動作の例を示した。
- (2) ロックを用いて複数の複製をアクセスする場

合の応答時間性能についての近似的な解析的評価方法を示した。またシミュレーションとの対比により近似方法がほぼ妥当であることを示した。

今後の課題として、

- (1) 通信時間、読取りを含めたシミュレーションを充実させ、計算方式の妥当性を精査し、本計算方式の適用限界を探求する、
 - (2) 今回のような単純なデータアクセスから、トランザクション処理を行う場合に現れる、読取り、書込みセットが複雑なアクセスへの拡張を試みる、
 - (3) 本計算方式を複製配置設計問題に適用し、統合された手法に整理する、
- などが課題である。

応答時間Sec

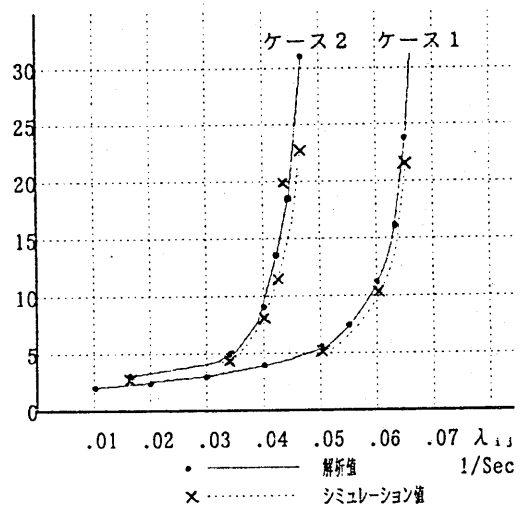


図10 解析値とシミュレーション値のグラフ

参考文献

- [1] Kleinrock L. "On the Modeling and Analysis of Computer Network." Proceedings of IEEE, Vol. 81, No. 8, '93
- [2] Singhal M. "Update Transport: A New Technique for Update Synchronization in Replicated Database Systems" IEEE Transactions on Soft. Eng., Vol. 16, No. 12, '90
- [3] Hellerstein J.L. "Construction Quantitative Models Using Monotone Relationships." IEEE Trans. on Knowledge & D. E. Vol. 7, No. 2, 1995
- [4] Salvatore T.M. et al., "Allocating Data and Operations to Nodes in Distributed Database Design." IEEE Trans. on Knowledge and Data Eng., Vol. 7, No. 2, '95
- [5] 宮西, 中村, 渡辺, 水野. "分散システムにおけるデータ更新性能解析と更新アルゴリズムの提案". 情報処理学会 DPS研, 95-DPS-69(1995)
- [6] 宮西, 中村, 渡辺, 水野. "分散システムにおけるデータアクセスの待ち行列による性能解析方法". 情報処理学会 MPS研, 95-MPS-3(1995)
- [7] 森村, 大前 "応用待ち行列理論" 日科技連 '75