JAIST Repository
https://dspace.jaist.ac.jp/

Title	分散型データベースシステムにおける複製データの動的配置制御に関する研究
Author(s)	狩野，光徳
Citation	
Issue Date	1998-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1131
Rights	
Description	Supervisor:横田 治夫，情報科学研究科，修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

修 士 論 文

分散型データベースシステムにおける複製データの動的配置制御に関する研究

指導教官　横田治夫 助教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

狩野光徳

1998 年 2 月 13 日

Copyright © 1998 by Mitsunori Karino

目 次

1 はじめに **1**

1.1 研究の背景と目的 1

1.2 本論文の構成 3

2 データベースのレプリケーション **4**

2.1 レプリケーションの概要 4

2.2 レプリケーション手法によるデッドロック率 8

3 マスタサイト移動手法 **10**

3.1 マスタサイト移動の必要性と有効性 10

3.2 マスタサイト移動手法概要 11

3.3 マスタサイト移動／選定の仕組み 13

4 効果の見積もり **15**

4.1 前提とするシステム／条件 15

4.2 アクセス要求頻度の地理的／時間的偏りの扱い 16

4.2.1 サイト・スキュー 16

4.2.2 サイト・スキューの経時変動の扱い 17

4.3 スループット向上率の導出 18

4.3.1 マスタサイトを固定した場合のスループット 18

4.3.2 マスタサイトを移動させる場合のスループット 20

4.3.3 スループット向上率 22

4.4 試算結果および考察 22

4.4.1 基本パラメータの値 22

4.4.2 試算結果および考察 23
4.4.3 マスタサイト移動効果に関する試算結果のまとめ 31

5 実装に関する考察 32
5.1 マスタサイト移動のための付加モジュール 32
5.2 処理の流れ 34

6 おわりに 36
6.1 まとめ 36
6.2 今後の課題 37

図 目 次

2.1 レプリケーションの仕組み 5
2.2 複製データベースのシステム形態 6
2.3 複製テーブルの形態 7
2.4 2 相コミットプロトコルによる更新処理の流れ 8
2.5 非同期型レプリケーションのシステム形態 9

3.1 マスタサイト固定と移動 11
3.2 マスタサイトの移動による通信コストの削減 13
3.3 マスタサイト移動手法の仕組み 14

4.1 サイト・スキューの例 17
4.2 アクセス頻度の時間変動の近似 18
4.3 マスタサイト移動間隔とスループット向上率の関係 24
4.4 NW 回線速度とスループット向上率の関係 25
4.5 サイト数とスループット向上率の関係 27
4.6 サイト数、サイト・スキューとスループット向上率の関係 28
4.7 サイト数、マスタサイト移動間隔とスループット向上率の関係 30

5.1 マスタサイト移動モジュールを組み込んだ分散データベースシステムの構想図 33
5.2 フロントエンド・モジュールの内部構成 34
5.3 フロントエンド・モジュールでの処理の流れ 35

第 1 章

はじめに

1.1 研究の背景と目的

複数の物流拠点（倉庫）における在庫管理などのように、データソースやアクセス要求の発生地点が地理的に分散しているようなケースでは、通信コストの節減や信頼性等の理由から、分散データベースシステムを構築する必要性が生じる。ここで、分散型データベースシステムとは、一般的には、コンピュータ・ネットワーク上の分散拠点（サイト）にデータベースを分散配置し、データの位置に対する透過性を保証する仕組みを備えたシステムを指す [1]。

通常分散型データベースシステムは、ユーザにとって集中型データベースと同じように扱えなければならない。この観点から、分散データベース構築のルールとして「C.J.Date の分散データベース 12 のルール」が提唱された [2]。同ルール中の「分散トランザクション管理」に関するルールを満たすためには、全サイト間で常時データの一貫性を保証する必要がある。このため、近年までの分散型データベースシステムでは、データの一貫性を保つ更新手法として、一般的に 2 相コミット [3] と呼ばれる手法が広く用いられてきた。しかし同手法の場合、常時全サイト間での同期が取れる反面、更新のたびに全サイトにアクセスするため、通信コスト等が掛かり過ぎるという問題が新たに発生する。

そこで、常時最新の情報を必要としているわけではなく、一時的な更新遅延が許容できる場合の構築手法として、近年、更新を非同期に行なう、レプリケーション手法が注目されるようになってきている [4]。実際、非同期レプリケーションをサポートした商用システムも多い [5]。

非同期レプリケーションは、オリジナルのデータの複製を、アクセス要求が発生し得る（遠隔地の）別サイトに配置し、相互の更新差分については非同期に転送する手法である。ここで特に、オリジナルデータを保持するサイトをマスタサイト、複製データを保持するサイトをレプリカサイトと呼ぶ。

非同期型のレプリケーションは、データ更新の方式などにより以下のようにさらに2つに分類される（各手法についての説明は2章に譲る）[6]。

- Lazy-Group-Replication
- Lazy-Master-Replication

上記2手法のパフォーマンスの違い等については、過去にいくつかのグループで研究が行なわれた。Jim Gray らの研究グループは、常に更新同期をとるレプリケーション手法（Eager-Replication）は、Lazy-Replication に比べデッドロックやトランザクションの失敗率が高くなるという解析結果を示した。同グループでは同時に Lazy-Master-Replication 手法の方が、Lazy-Group-Replication 手法よりもさらにデッドロック率が低くなることも明らかにしている [6]。また San-Yih Hwang らの研究グループでは、システムの平均応答時間の解析で、Lazy-Replication の方が Eager-Replication よりも短く、さらにトランザクション発生頻度が上がるほど、その差は拡大することを示した [7]。

以上のことから、常に複製データ間で更新の同期を図る必要がないケースで分散型データシステムを構築する場合、トランザクション処理効率／コストの観点から、Lazy-Master-Replication 手法が望ましいといえる。しかし、Lazy-Master-Replication 手法では、更新要求や最新情報の参照要求がレプリカサイトで発生した場合、ネットワークを経由してマスタサイトにアクセスする必要が生じる。そこで、最新のデータへのアクセス要求が全体の一部に限られ、かつそうした要求先が、ある時間帯ごとに変わるような場合、オリジナルデータに対するアクセス要求の発生地点と、マスタサイトの稼働地点をなるべく一致させることができれば、さらなる通信コストの節減が望める。それゆえに、発生地点の偏り状態によっては、マスタサイトの移動により通信コストをさらに節減することができると考えられる。そしてそのことにより、トランザクション1件当たりの平均処理時間の短縮を図ることができ、トランザクションスループットのさらなる向上が期待できる。

本研究では、特定のアクセス状況下において、Lazy-Master-Replication 手法におけるトランザクション処理効率をさらに向上させ得る、マスタサイト移動について考察する。また、同移動による効果について解析し、効果が期待できる条件についての考察も行なう。

1.2　本論文の構成

本論文の構成は次のとおりである。この後の 2 章では前に述べた通り、既存のデータベースのレプリケーション 2 手法について説明する。3 章では、本研究で考案したマスタサイト移動手法について述べる。また 4 章では、同手法の導入効果に関する試算結果を示し、効果が得られる条件について考察する。そして 5 章では、実装に関する考察として、システムが自立的にマスタサイト移動をするために必要な機能と、それら機能の実装方法に関して考察を行なう。最後に 6 章で本研究内容についてまとめる。

第 2 章

データベースのレプリケーション

2.1 レプリケーションの概要

データアクセス要求の発生場所が地理的に分散していて、なおかつサイト間におけるデータの一時的な更新遅延を許容できる場合、レプリケーションは分散型データベースシステムを構築するための、現実的かつ有力な手法である。本章では、分散データベースシステムの構築手法の 1 つで、本研究で対象としている（データベースの）レプリケーション手法について述べる。

レプリケーションとは、通信コストの節減や信頼性等の理由から、オリジナルのデータベースの複製を、アクセス要求等に応じて別サイトに配置する手法である。また同手法を用いた商用システムの多くは、レプリカの更新を非同期に行うように設計されている [4]。このため本研究では、扱うレプリケーション手法のタイプを非同期型レプリケーションに限定し、今後、非同期更新型レプリケーションを単にレプリケーションと呼ぶ。

図 2.1 はレプリケーションの仕組みを示したものである。更新要求は、データの整合性の理由から全てマスタサイトに転送される。そしてマスタサイトのテーブルの更新に応じて、更新情報の内容が更新ログに蓄積されていき、あるタイミングでこの更新ログがレプリカサイトに送られる。レプリカサイトは更新ログを受けとると、このログを基に、自サイトのレプリカテーブルにマスタテーブルの更新内容を反映させる。なお更新ログの転送はレプリケーション用のプログラムが行なう。

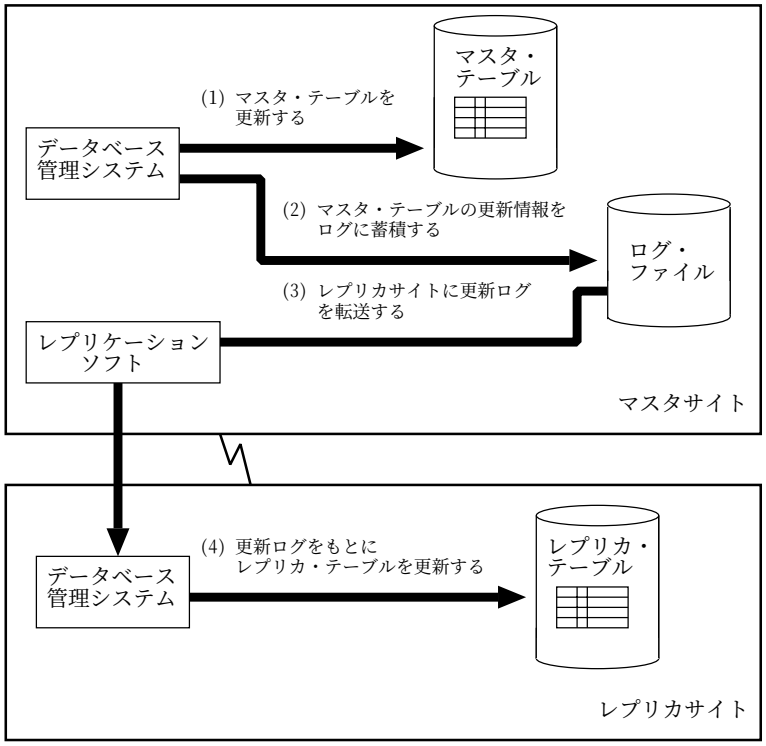

図 2.1: レプリケーションの仕組み

次に複製データベースの配置方式によるレプリケーションの分類について説明する。

複製データベースの各種システム形態を図 2.2 に示す。レプリケーション手法を用いるシステムの形態は、図にある通り 6 つの形態ある。(1) の 1 対 1 型は、マスタテーブルとレプリカテーブルが 1 対 1 で対応する、最も単純で基本的な形態である。更新トランザクションはマスタテーブルで行ない、レプリカテーブルは参照専用で利用する。基幹業務系のデータベースをマスタとして、情報運用系システムにその複製を持つケースがこれにあたる。(2) の 1 対多型はマスタデータベースを複数に分割し、それぞれ別のレプリカサイトに配置する形態である。この形態は、基幹業務系システムのデータベースの一部の複製を各業務部門に配るケースに用いられる。(3) の多対 1 型は、(2) のケースとは逆に、支店や部門ごとに業務処理を行ない、後でその内容を全社的なデータベースに反映させる場合に用いられる形態である。現時点では以上 3 形態が、業務用システムにおける主流となっている。残りの 3 形態については次のとおりである。(4) の水平分割型は 1 つのテーブルの行を論理的に分割してマスタとしての所有権を分ける形態である。(5) の並立

型は複数のテーブルが完全に同格で、両方ともマスタでありレプリカである形態である。最後の (6) の連鎖型は、マスタ／レプリカの関係が入れ子状になっている形態である。この形態では、マスタテーブルの更新内容は、「子」テーブルを経由して「孫」テーブルに伝搬される。

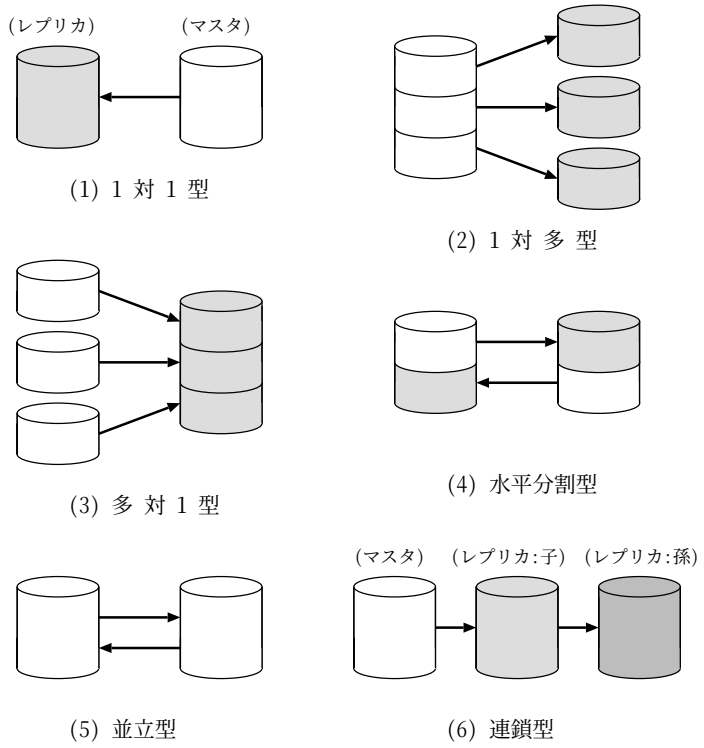

図 2.2: 複製データベースのシステム形態

現在の商用システムでは、(1)、(2) の形態に対する需要が最も多い。よって後の解析を簡潔化のため、本研究では対象とするシステム形態として (1)1 対 1 型を選択する。

次に複製テーブルの形態について簡単にふれる。図 2.3 に複製テーブルの各種形態を示す。図にあるとおりレプリカテーブルは、実際の商用システムにおいては、必ずしもマスタテーブルの完全な複製である必要はない。該当する項目／行のデータを抽出して得られたテーブル、あるいは、ある操作で新たに得られたテーブルの複製をマスタテーブルのレプリカとすることも許される。ただし、本研究では解析の簡潔化のため、複製テーブルの形態は、レプリカテーブルは、マスタテーブルのサブセットではなくフルセットであるとする。

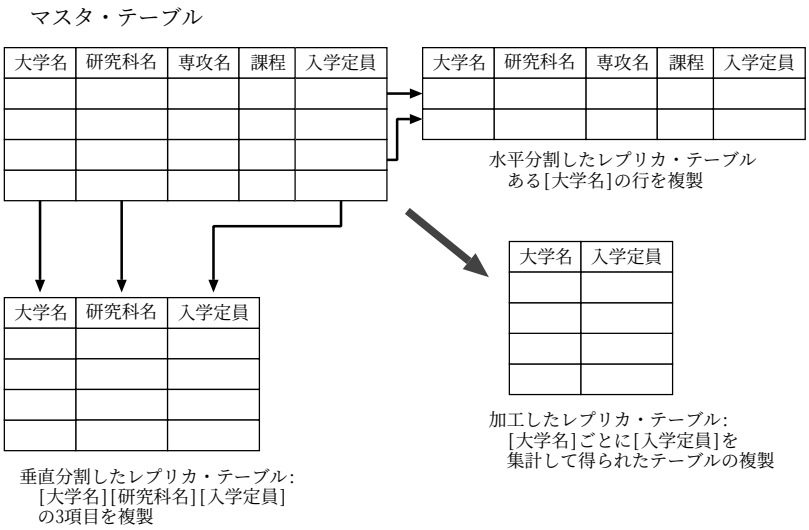

図 2.3: 複製テーブルの形態

レプリケーションに関する説明の最後として、レプリカの差分更新の方法について説明する。現在の商用システムの多くは、レプリカの差分更新にに 2 相コミットプロトコルを利用している [3]。本プロトコルによる更新処理の流れを図 2.4 に示す。本プロトコルによる処理の流れは次の通りである。まずマスタサイト側で各レプリカサイトに更新の通知を行ない、応答を待つ。各レプリカサイトは、更新準備が整い次第マスタに返事を返す。マスタサイトは全てのレプリカサイトの返事がそろった時点で、各レプリカサイトに対して更新ログの転送を開始する。そして、全レプリカサイトからの応答の確認がとれた時点でコミットを指示する。各レプリカサイトはコミット通知により差分更新処理を行ない、マスタサイトと更新の同期をとる。ただし、更新通知、更新ログ転送に対する返事が全レプリカサイトから得られなかった場合、その時点でマスタサイトは各レプリカに対しアボートを指示する。

また現在の多くの商用システムでは、更新ログ形式として次の 2 形式のうちいずれか一方を用いている。1 つは「ジャーナル」に近い形式でもう 1 つは更新要求に似た形式である。「ジャーナル」とはデータベース管理システム（以下 DBMS と呼ぶ）がトランザクション処理に伴って蓄積しているレコード単位の更新履歴情報を指す。また後者の形式によるログのイメージはデータベース操作言語の SQL の記述イメージとほぼ同じものになっている。本研究では、差分更新にかかるコストの解析上の理由から、更新ログの形式

として後者の形式を想定する。

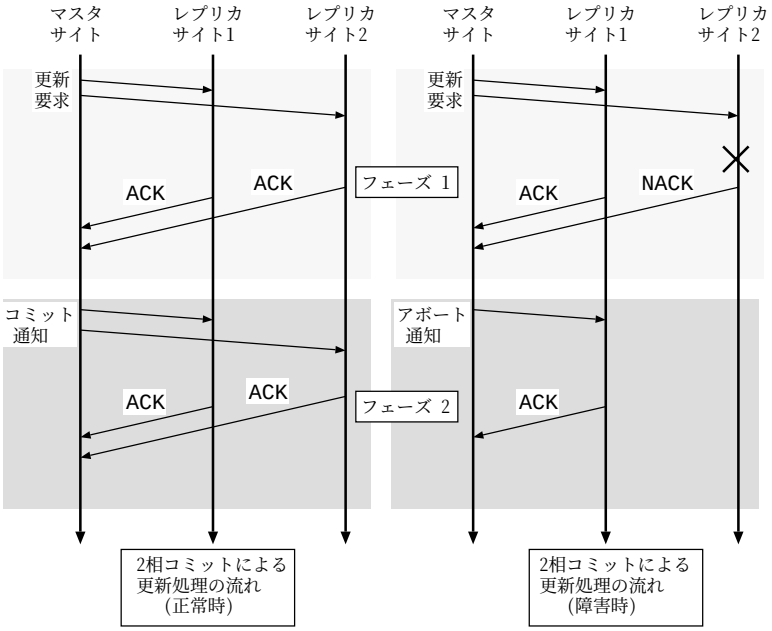

図 2.4: 2 相コミットプロトコルによる更新処理の流れ

2.2 レプリケーション手法によるデッドロック率

分散データベースシステムでは、各サイトで更新要求が同時に発生するなどして、サイト間で、ファイル、ロックまたはその他の資源の排他アクセスをめぐって競合状態が発生し、システム全体がデッドロックに陥る場合がある [8]。

Jim Gray、Patrick O'Neil らの研究グループは、レプリケーションにおけるデッドロック率に関する研究 [6] で、システム形態により非同期型のレプリケーションを以下の 2 種類に分類し、解析を行なった（図 2.5 に同解析における非同期型レプリケーションの各システム形態を示す）。

Lazy-Group-Replication 各サイトは、他サイトのデータの複製を保持する。そして全てのサイトに対し、ローカルデータの更新を許す。更新時には、ローカルデータに対する更新トランザクションと同じトランザクションが、他の全サイトへ転送され、

時刻刻印で矛盾が生じた場合は調停する。同手法は前節の複製形態による分類では、並立型に相当する。

Lazy-Master-Replication 各サイトは、他サイトのデータの複製を保持する。ここで、各データに所有権が定められ、所有権を持つサイトのみが当該データの更新を行なうことができる。同手法は前節の複製形態による分類では、1 対 1 型に相当する。

解析では、1 サイト当たり、1 トランザクション当たり、データベースの単位大きさ当たりの、それぞれのデッドロックの発生率の計算が行なわれた。そして、Lazy-Master-Replication の方が、トランザクション 1 件あたりの処理時間が通信コストの分短いため、デッドロック率の方も若干小さくなることが明らかにされた。

現在の商用システムでのシステム形態に関する需要や、処理コストの解析上の簡便化、およびシステム形態におけるデッドロック率の研究報告から、本研究では、想定するレプリケーション手法として、Lazy-Master-Replication を選択する。

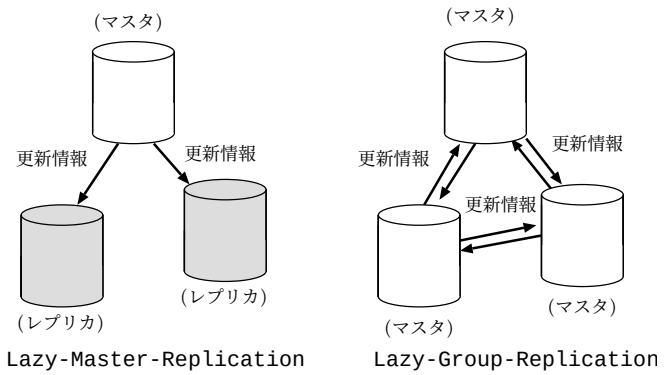

図 2.5: 非同期型レプリケーションのシステム形態

第 3 章

マスタサイト移動手法

3.1 マスタサイト移動の必要性と有効性

Lazy-Master-Replication 手法は、レプリカの更新を非同期に行なうため、ネットワーク回線速度に影響されにくく、また回線自体の障害にも対応が可能である。このため、一般的にサイト間の一時的な更新遅延を許容できるケースでは、分散データベースシステムの構築手法として、非常に有力な手法であるといえる。しかし、更新は非同期に行なわれるため、別のサイトが当該データのマスタサイトである場合、更新や最新情報の参照を行なうためには、ネットワークを経由してこのマスタサイトへアクセスしなければならない。よって、Lazy-Master-Replication 手法の有効性は、アクセス全体に占めるマスタデータへのアクセス件数の割合に左右される。マスタデータへのアクセス要求の発生件数に地域的な偏りが存在し、さらにその偏り状態が経時変動する場合、アクセス件数に応じてマスタサイトを移動させることで、要求処理の効率を向上させることが可能である。図 3.1 は、アクセス要求件数の時間的変動に応じた、マスタサイトの移動のイメージを図示したものである。

図 3.2 はオリジナルデータへのアクセス要求の発生地点が、時間帯により変動したときのトランザクションの総処理時間のイメージを図示したもので、図 3.1 に対応している。同図は、「全てのアクセスはマスタデータへのアクセスであり、アクセス要求発生地点は、つねに 1 サイトに集中している」という非常に極端な例を用いて、マスタサイト移動の有用性を説明している。マスタサイトが固定されている場合、各アクセス毎に通信が必要となるが、マスタサイトを移動することにより、通信時間を削減することができる。また、

サイト間の同期は移動時のみになり、更新処理をまとめることができるので、アクセス毎に 2 相コミットを行なうより通信コストは小さく抑えることができる。よって平均処理時間の短縮が図ることができ、トランザクション処理効率のさらなる向上が期待できる。ただしここで 1 つ注意が必要である。図のケースでは、アクセス要求の地理的偏りと偏り状態の経時変動が、非常に極端な場合を想定しているので、マスタサイト移動により、マスタデータへのアクセスコストの分が全て短縮されるが、実際に短縮できるコストは、上記の偏りと経時変動の状態に依存する。このためマスタサイトを移動させる場合は、これらの偏り状態を常に見究める必要がある。

次節では、本研究で考察したマスタサイト移動手法について説明する。

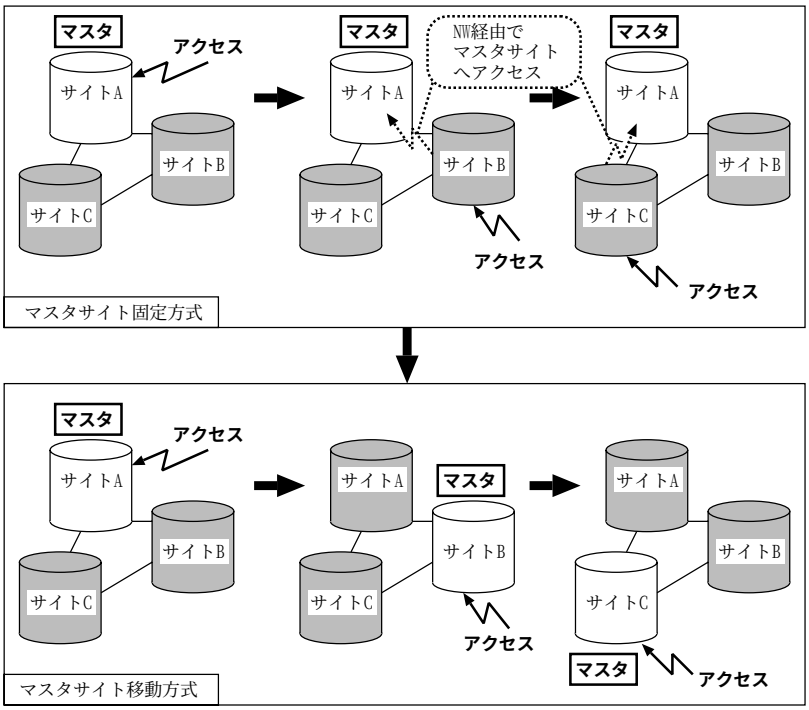

図 3.1: マスタサイト固定と移動

3.2 マスタサイト移動手法概要

本節では、マスタサイト移動手法の概要を説明する。

はじめに本手法では、扱うトランザクションの種類として、データベースへのアクセス

の種類に準じ次の 3 つを想定している。

- dirty-read:更新遅延データの参照
- read:最新情報の参照
- write:データの更新

全サイトは同じ内容のデータベースを保持する。そしてデータベース中の各テーブルには所有権が定められており、所有者の持つテーブルがマスタテーブルとなる。また同テーブルに対するアクセスが最も頻繁に発生する地域のサイトが、そのテーブルの所有者になる。このためテーブルによって所有者が異なる場合がある。

図 3.2 は通常のデータベースのレプリケーションに、マスタサイト移動手法を採り入れた時の、サイト内部の動作の概略を示したものである。システム稼働中、マスタサイトは、トランザクション監視機構により、マスタテーブルへのアクセス件数を監視する。アクセス要求の内、更新要求についてはその内容を更新ログとして保存しておく。そしてマスタテーブルにある一定件数のアクセスが行なわれると、次期マスタサイトを選定する処理に移るため、一時アクセス要求の受付を中止する。また同時に他の全サイトへ、次期マスタサイト選定処理に移ったことを通知する。次期マスタサイトの選定は、アクセスのログを参照して、以前のマスタ選定処理終了後、最もアクセス要求頻度が高かった要求元サイトを次期マスタサイトに選定する。他サイトが次期マスタサイトに選ばれた場合、システムはマスタサイト移動処理に移る。マスタサイトを移動させるときは、直前の移動からそれまでの更新ログと次期マスタサイトの情報を他の全サイトに転送し、2 相コミットで同期をとる。ここまでの処理は、それぞれ次期マスタサイト選定機構と、マスタサイト移動機構により実行される。またこれらの機構は、DBMS とユーザの間をとり持つフロントエンド・モジュールに含まれる。

以上がマスタサイト移動手法の概要である。同手法の仕組みで最も重要である部分は、次期マスタサイトの選定部と更新差分転送部である。これらについては次節でさらに詳しく述べる。

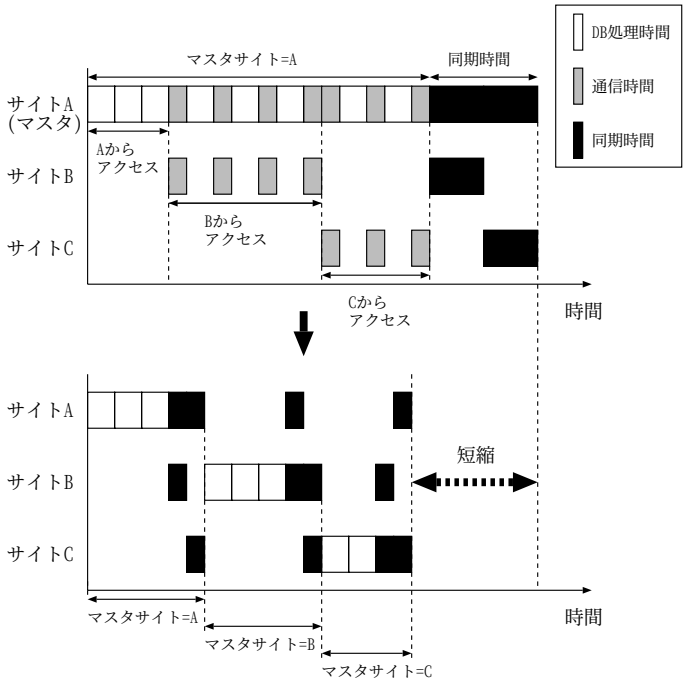

図 3.2: マスタサイトの移動による通信コストの削減

3.3 マスタサイト移動／選定の仕組み

マスタサイト移動手法で中心になる仕組みは以下の 2 つである。

1. アクセス頻度に応じて次期マスタサイトを選定する仕組み
2. マスタサイト移動時に他サイトへ更新差分を転送する仕組み

以上 2 つの仕組みに関する詳細な仕組みについては以下のとおりである。

1. **次期マスタサイトの選定**: 全トランザクションメッセージ（SQL 文）に、要求元サイトの ID を格納するヘッダを付加しておく。全サイトのフロントエンド部は、現時点での各テーブルのマスタサイト番号と自サイトの番号を保持する。更新トランザクションメッセージについては、マスタサイトが切り替わるまで、全てログに保存する。マスタサイトは一定件数のトランザクションを受け付けると、要求サイト元ごとに件数を集計し、要求件数がある程度突出しているサイトを次期マスタサイトに選定する。

2. 他サイトへの更新差分の転送: 選定処理でレプリカサイトが次期マスタサイトに選定された場合、現マスタサイトは、全サイトにメッセージを送信し、トランザクション処理を一時中断させ、応答を待つ。この間にマスタサイトに送られたトランザクションについては、キューに貯められる。現マスタサイトは全ての応答を受信後、自分がマスタサイトであった期間のログと、新マスタサイト番号を他の全サイトに転送する。ここで更新ログの形式については、現在の商用システムで採用されている主要な形式の 1 つに準じ、更新トランザクションの SQL 文とする。各サイトは、ログと新マスタサイト番号を受けとると、ログにある更新を圧縮して実行し、全サイトで 2 相コミットによる同期をとった後、トランザクション処理を再開する。

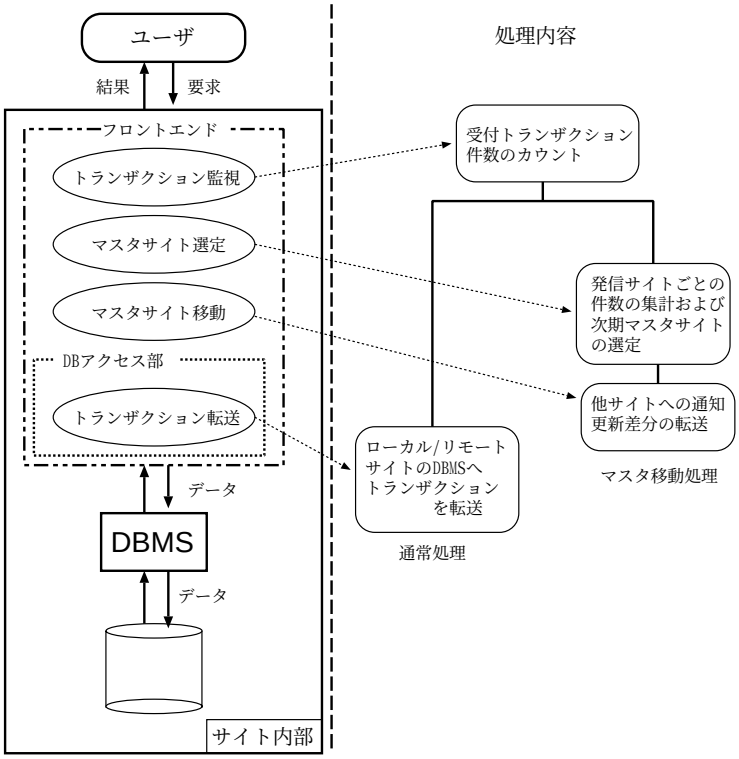

図 3.3: マスタサイト移動手法の仕組み

第 4 章

効果の見積もり

4.1 前提とするシステム／条件

3 章では、マスタサイトの移動効果として、特定環境下でトランザクション処理の効率化が期待できることを挙げた。本研究ではさらに、トランザクション処理の効率化について、計算による検証を行なった。

検証のための解析では、上記の効果の評価値として、マスタサイトの移動に伴う、システムの平均スループット向上率を用いた。解析は次の手順で行なった。はじめに想定するシステムを限定した上で、トランザクション処理コストに関わる要素をパラメータ化し、所定のパラメータ値を変動させた時のスループット向上率の挙動をグラフ化した。そして、グラフを基にスループット向上率を左右する要因の把握と、マスタ移動効果を増大させる条件について考察を行なった。

が大きくなる条件について考察を行なった。

以下に本解析におけるシステムの前提条件を示す。

- メッセージの送信遅延は、回線経路によらず一様であるとする。
- トランザクションにおける各種類の割合は全サイト同一とする。
- システムが受け付ける総トランザクション件数は常時一定とする。
- サイト自身の処理能力は全サイト同等であるとする。
- 各サイトでは、トランザクションは逐次的に処理されるものとする。

- ネットワーク回線の故障は考慮しない。
- ローカルからのアクセスとリモートからのアクセス件数の割合は、各サイトとも等しい。
- トランザクションは、通常、複数の参照／更新処理要求で構成されるが、解析を簡潔にするため、1 つのトランザクションは、単一のアクセス要求で構成されているとする。
- システム起動直後のマスタサイトの位置は、アクセス要求頻度によらない。

また上記のシステムに対する仮定以外に、解析に関して次の前提を置いた。システムの平均スループットの比較では、基準時間を、マスタサイト移動手法を採り入れる前の同期間隔時間とした。この前提は、スループット算出の前段階である、システムの平均トランザクションコストの算出過程で、更新の同期にかかるコストの計算を簡潔にするためのものである。

本研究では、マスタサイト移動手法を用いるべき状況として、アクセス要求の発生地点に地理的に偏りがあり、かつ偏り状態が経時変動することを前提としている。このため本解析では、この 2 つの要素を数値化した。これについては次節で説明する。

4.2 アクセス要求頻度の地理的／時間的偏りの扱い

本試算では、アクセス要求頻度の地理的／時間的偏りについて、以下のように数値化した。

4.2.1 サイト・スキュー

システムは n 個のサイトで構成されているとする。マスタサイト移動手法では、マスタサイトの選定は、マスタサイト上のトランザクション受付件数が、ある規定の件数に達した時点で行なわれる。この期間がマスタサイト移動手法における同期間隔となる。同間隔内での、受付件数のサイト間における偏りを、サイト・スキューと呼称する。上記同期間隔内において、最多受付サイト以外のサイトにおける平均受付件数は、システムの総受付件数から累計最多受付件数を差し引いた件数を、$n-1$ で割って得られた件数になる。こ

こで、サイト・スキューの度合を示す指標として、最多受付サイトの件数と、その他のサイトの平均受付件数の比を用いる。また、この比を特にサイトスキュー比と呼ぶ。アクセス要求の地理的偏り度合をサイト・スキュー比で表す。図 4.1 に全体の受付件数が 30 件で、サイト・スキュー比 R_s がそれぞれ 1.0、3.0 の場合の、サイト間における受付件数の偏り状態の例を示した。

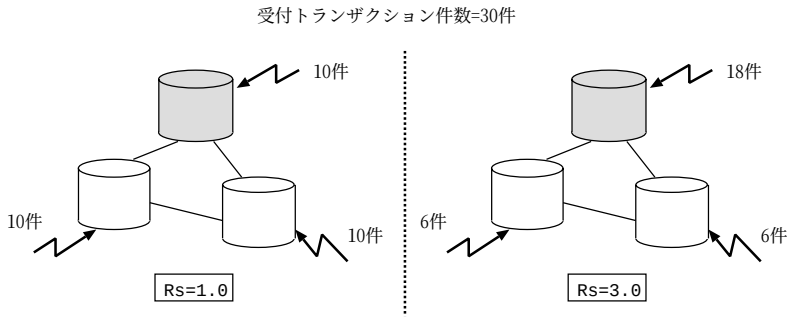

図 4.1: サイト・スキューの例

4.2.2 サイト・スキューの経時変動の扱い

マスタサイト移動手法は、ある受付件数間隔でサイトごとのアクセス要求頻度を監視し、必要に応じてマスタサイトを移動する手法である。このため、同手法を用いるケースでは、マスタサイトを固定したシステムにおける同期間隔内で、サイト・スキューの状態が 1 回以上変化すると仮定する。具体的には、最多アクセスサイトの移動（交替）間隔をマスタサイト移動間隔とし、サイト・スキューの経時変動の指標に用いる。ここで解析の簡便化のため、本解析では、元のシステムの同期時間内で発生する、マスタサイト移動の間隔を平均化する。さらに、1 つのマスタサイト移動間隔区間における、単位時間当たりのアクセス要求頻度も同様に平均化して扱う。また 1 マスタサイト移動間隔区間の長さは、次期のマスタサイト選定にかかるコストに比べて、十分長いとする。この仮定により、本解析ではマスタサイト移動手法を用いる場合は、つねにマスタサイトが最多受付サイトになるものとする。

図 4.2 は、上記のアクセス要求頻度における偏りの、時間変動に関する仮定を図示したものである。縦軸に単位時間当たりのアクセス要求頻度、横軸に時間をとってある。図中

の各長方形の面積は、該当時間帯における、各サイトの累計アクセス受付件数を表す。一般的には、グラフは連続的な曲線になると考えられるが、前述のとおり本解析では、単位時間当たりのアクセス頻度とマスタサイト移動間隔を平均化して考えるので、図のとおりグラフの概形が矩形状に近似される。

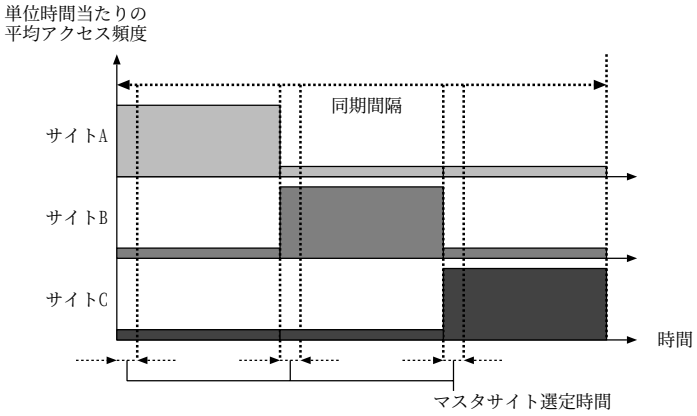

図 4.2: アクセス頻度の時間変動の近似

4.3 スループット向上率の導出

本節では、3 章で説明したマスタサイト移動手法の仕組みと、本章前節までの偏りに関する考察を基に、マスタサイト移動手法の導入による、システムのスループット向上率について解析を試みる。

4.3.1 マスタサイトを固定した場合のスループット

本小節では、マスタサイト移動手法導入する前のシステムにおける、最大平均スループットを計算する。

アクセスに占める dirty-read、read、write の割合をそれぞれ R_d、R_r、R_w とする。また各サイトにおける平均 read、write コストを t_r、t_w[msec]、平均メッセージ送（受）信コストを t_c[msec]、そしてローカルアクセスがマスタデータにアクセスする確率を P_l とすると、通常のトランザクション 1 件にかかる平均コストは、以下の通りになる。

$$T_{fn}[sec] = R_d t_r + P_l(R_r t_r + R_w t_w) + (1 - P_l)\{R_r(2t_c + t_r) + R_w(2t_c + t_w)\} \tag{4.1}$$

次に、更新の同期にかかるコストについて計算する。2 章で述べたとおり更新の同期は、SQL 文である差分更新情報を転送し、2 相コミットプロトコルを用いて更新の同期処理を行なう。ここで、更新トランザクション 1 件当たりの平均 SQL 文の大きさを D_w[Bytes]、通信回線のスループットを v[bps]、マスタサイト固定状態における、更新同期間隔内に受け付けた平均トランザクション件数を N_r[件] とする。このとき 1 つのレプリカサイトと更新の同期を図るためにかかるコスト t_{fs}[sec] は、1 往復分メッセージ転送コストと、更新差分転送コスト、相手のサイト上における差分更新コスト、コミット後に返す Ack のためのメッセージ転送コストの合計になる。

$$t_{fs}[sec] = 3t_c + N_r R_w \left(\frac{8D_W}{v} + t_w\right) \tag{4.2}$$

システム中の全サイト数を n とすると、レプリカサイトは $n-1$ サイトあるからシステム全体での更新同期コスト T_{fs}[sec] は以下のとおりになる。

$$T_{fs}[sec] = (n-1)t_{fs} \tag{4.3}$$

よって更新の同期も考慮に入れた、トランザクション 1 件当たりの平均コスト T_{fix}[sec] は以下のとおりになる。

$$T_{fix}[sec] = \frac{N_r}{N_r + 1}T_{fn} + \frac{1}{N_r + 1}T_{fs} \tag{4.4}$$

ここで本解析では、アクセス要求件数のサイト間おける偏りを想定しているので、4.2.1 節の考察に従いサイト・スキュー比を R_sとおく。

この時、ローカルアクセスがマスタデータにアクセスできる確率 P_lは、サイト・スキュー比によって変動し、またマスタサイトと最多受付サイトが一致しているか否かで、2 通りに分かれる。本解析では、受け付けた全トランザクション中に占める、read、write トランザクションの割合は、全サイト同じであると仮定している。このため全トランザクション中で、マスタサイトが受け付けた件数の割合が P_lとなる。

$$P_l = \begin{cases} \frac{R_s}{R_s+n-1} & \text{マスタサイト＝最多受付サイト} \\ \frac{1}{R_s+n-1} & \text{マスタサイト}\neq\text{最多受付サイト} \end{cases} \tag{4.5}$$

また各マスタアクセス率による、通常処理の平均コストはそれぞれ以下のとおりになる。

最多受付サイトがマスタサイトである場合:

$$T_{fn}[sec] = (R_d + R_r)t_r + R_w t_w + 2t_c \left(1 - \frac{R_s}{R_s + n - 1}\right)(R_r + R_w) \tag{4.6}$$

最多受付サイトがマスタサイトでない場合:

$$T'_{fn}[sec] = (R_d + R_r)t_r + R_w t_w + 2t_c \left(1 - \frac{1}{R_s + n - 1}\right)(R_r + R_w) \tag{4.7}$$

更新の同期間隔内での平均総アクセス件数を N_r[件]、マスタサイト移動手法による、マスタサイト移動間隔を R_t[件] とすると、元の同期間隔内におけるマスタサイトの移動回数 N_{mig}[回] は以下のとおりになる。

$$N_{mig} = \lceil \frac{N_r}{R_t} \rceil \tag{4.8}$$

自サイトが最多サイトになる確率は $\frac{1}{n}$ であるからマスタサイトの位置も考慮にいれた、最終的なトランザクション 1 件当たりの平均コストは、以下のとおりになる。

$$T_{fix_cost} = \left(\frac{1}{n}\right) T_{fix} + \left(1 - \frac{1}{n}\right) T'_{fix} \tag{4.9}$$

ただし

$$T'_{fix} = \frac{N_r}{N_r + 1} T'_{fn} + \frac{1}{N_r + 1} T_{fs} \tag{4.10}$$

マスタ固定のままのシステムの最大平均スループットは以下の式で計算される。

$$TH_{fix} = \frac{1}{T_{fix_cost}} \tag{4.11}$$

4.3.2 マスタサイトを移動させる場合のスループット

次にマスタサイト移動手法を採り入れた場合の、システムの最大平均スループットの算出式を行なう。

まず通常のトランザクション 1 件あたりの平均処理コスト T_{mn}[sec] は、マスタサイトの固定ままの時と同じであるから

$$T_{mn}[sec] = T_{mn} = R_d t_r + P_l(R_r t_r + R_w t_w) + (1 - P_l)\{R_r(2t_c + t_r) + R_w(2t_c + t_w)\} \quad (4.12)$$

ここで、4.2.1 節の、アクセス要求件数の時間的偏りの扱いに関する考察に従い、マスタサイトは常に最多受付サイトであるとして計算を進める。サイト・スキュー比を R_s とすると、$P_l = \frac{R_s}{R_s+n-1}$ であるから、アクセス要求件数におけるサイト間での偏りを含めた、システムの 1 件当たりのトランザクションの平均処理コスト T_{mn}[sec] は、以下のとおりになる。

$$T_{mn}[sec] = (R_d + R_r)t_r + R_w t_w + 2t_c\left(1 - \frac{R_s}{R_s + n - 1}\right)(R_r + R_w) \quad (4.13)$$

また、1 レプリカサイトとの更新同期処理にかかるコスト t_{ms}[sec] は、更新差分情報の大きさが、マスタサイト固定時の場合の $\frac{1}{N_{mig}}$ であるから、式 (4.2) より以下のとおりになる。

$$t_{ms}[sec] = 3t_c + \frac{N_r R_w}{N_{mig}}\left(\frac{8D_W}{v} + t_w\right) \quad (4.14)$$

よってシステム全体の同期コスト T_{ms}[sec] は、次のとおりになる。

$$T_{ms}[sec] = (n-1)t_{ms} \quad (4.15)$$

以上より、更新の同期にかかるコストを含めた、トランザクション 1 件当たりの平均処理コスト T_{mig_cost}[sec] は次のとおりになる。

$$T_{mig_cost}[sec] = \frac{N_r - N_{mig}}{N_r}T_{mn} + \frac{N_{mig}}{N_r}T_{ms} \quad (4.16)$$

マスタサイト移動手法の導入による、システムの最大平均スループットは以下の式で計算される。

$$TH_{mig} = \frac{1}{T_{mig_cost}} \quad (4.17)$$

4.3.3 スループット向上率

4.3.1、4.3.2 節で導出した、それぞれの最大平均スループット TH_{fix}、TH_{mig}から、スループット向上率 P_{TH}[%] は以下のとおりになる。

$$P_{TH} = \left(\frac{TH_{mig}}{TH_{fix}} - 1\right) \times 100[\%] \tag{4.18}$$

次節のマスタサイト移動手法の導入効果の試算では、評価の指標値として P_{TH}の値を用いる。

4.4 試算結果および考察

4.4.1 基本パラメータの値

マスタサイト移動手法の導入効果に関する試算では、基本パラメータ値を以下のとおりに設定した。試算において、特に操作の必要がないパラメータ値についてはすべて下表の値を用いた。

サイト数 (n)	5
DB の平均 read,write 時間（t_r, t_w）	10[msec]
平均メッセージ長	1[KB]
HDD の write スループット	1[MB/sec]
サイト・スキュー比 (R_s)	10.0
遅延データ参照の割合 (R_d)	50.0 [%]
更新トランザクションの割合 (R_w)	25.0 [%]
ネットワーク回線速度 (v)	8k(bps)
トランザクション件数 (N_r)	10000[件]
マスタサイト移動間隔件数 (R_t)	1000[件]

表 4.1: 試算に用いた基本パラメータ値

4.4.2 試算結果および考察

本節で示す試算結果は、表 4.1 の基本パラメータ値を基に、スループットに関わる各要素を変化させた時のスループット向上率を試算したものである。試算結果のグラフ中で、特に断りがないパラメータ値については、同表の値を用いてある。

マスタサイト移動によるスループット向上率は、マスタサイト移動の仕組みから総トランザクション件数と、マスタデータに対するアクセス中のローカルアクセス率、そしてトランザクション処理コスト中に占める、同期処理コストの割合によって決まる。このため本試算では、マスタサイト移動効果に影響する要素として、トランザクション件数、ネットワーク回線速度（以下 NW 回線速度と略す）、サイト数、サイト・スキュー比、マスタサイト移動間隔の 5 要素に着目した。

本試算では、まずマスタサイト移動によるスループット向上特性の基本特性を調査するため、トランザクション件数とマスタサイト移動間隔、NW 回線速度、サイト数とスループット向上率の関係について試算と検証を行なった。またその後、サイト数とサイト・スキュー比、マスタサイト移動間隔の間での影響度の違いについても検証した。

各ケースにおける検証に入る前に、試算結果に関してここで 1 点確認しておく。全ての試算結果において、トランザクション件数とは、マスタサイトを固定したままのシステムにおいて、更新の同期間隔内に受け付ける平均トランザクション件数を指す。

図 4.3 は、マスタサイト移動間隔別に、トランザクション件数とスループット向上率の関係をグラフ化したものである。グラフからスループット向上率は、マスタサイト移動間隔によらず、トランザクション件数の増加に伴って収束していくことが読みとれる。この理由は次のとおりである。マスタサイト移動手法では、マスタサイトでのアクセス件数が、ある規定件数に達した時点で移動を行なう。ここでマスタサイト固定状態での同期間隔内のトランザクション件数が、規定件数の整数倍でない場合、最後の半端区間のために 1 回分余計に更新の同期をとらなければならない。このため最後の半端区間では、1 件あたりの平均トランザクション処理コストにかかる同期コストの大きさが相対的に増大する。よってこの時、スループット向上率が相対的に落ち込むことになる。また当然のことながら、トランザクション件数が少ない時ほど同期回数の増加の影響が大きい。よって、トランザクション件数が比較的少ないところほど、マスタサイト移動によって同期間隔にずれが生じた場合に発生するスループット向上率の上下動が目立つ。逆にトランザクション件数の増大に伴い、スループット向上率は収束（安定）していく。

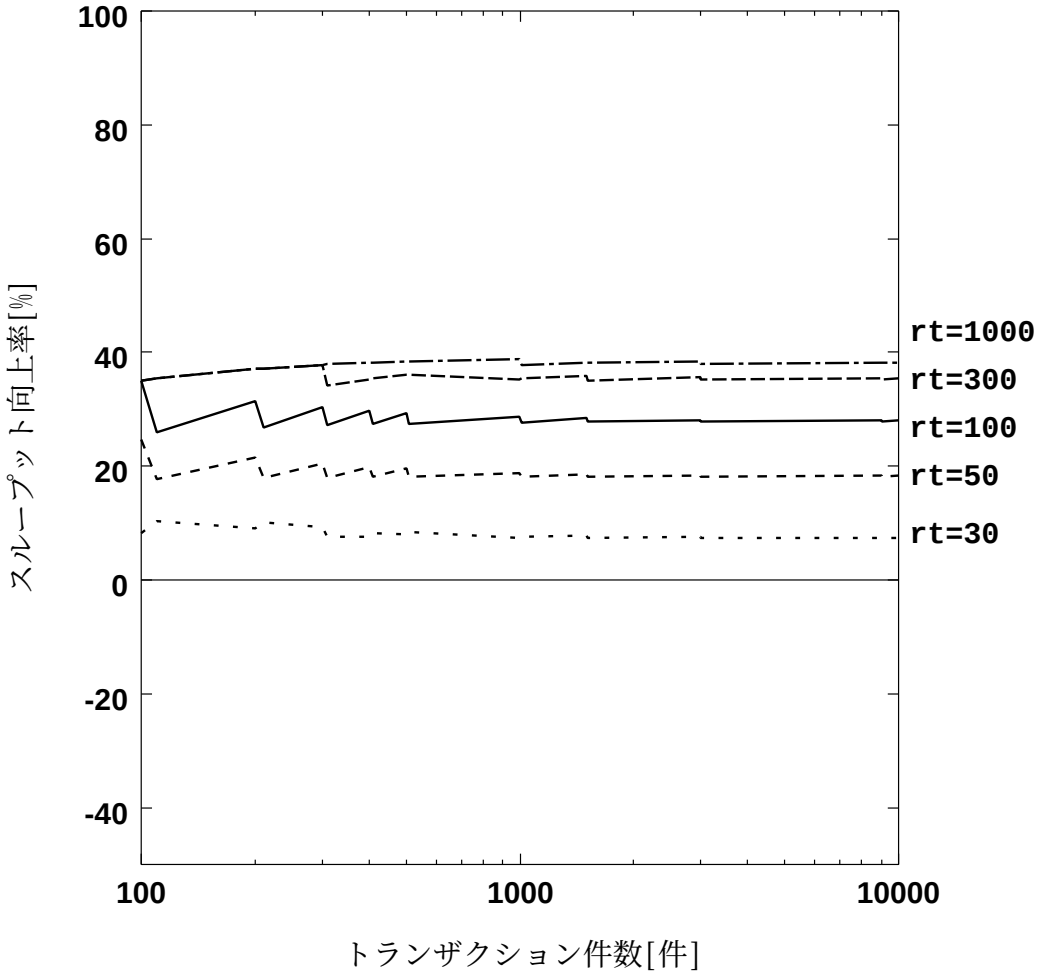

図 4.3: マスタサイト移動間隔とスループット向上率の関係

次にマスタサイト移動間隔とスループット向上率の関係に着目すると、移動間隔の拡大に伴ってスループット向上率が安定していくことが読みとれる。この理由は、トランザクション件数の増加に伴う向上率の収束の理由と同じである。ただし、移動間隔がある程度より短くなると（グラフではおよそ 50 件以下）、トランザクション処理コスト中に占める、同期処理コストの割合が大きくなり、1 回分の同期処理コストの増分が相対的に目立たなくなるので、移動間隔がある程度を越えて小さくなる場合も、スループット向上率の上下動が小さくなる。

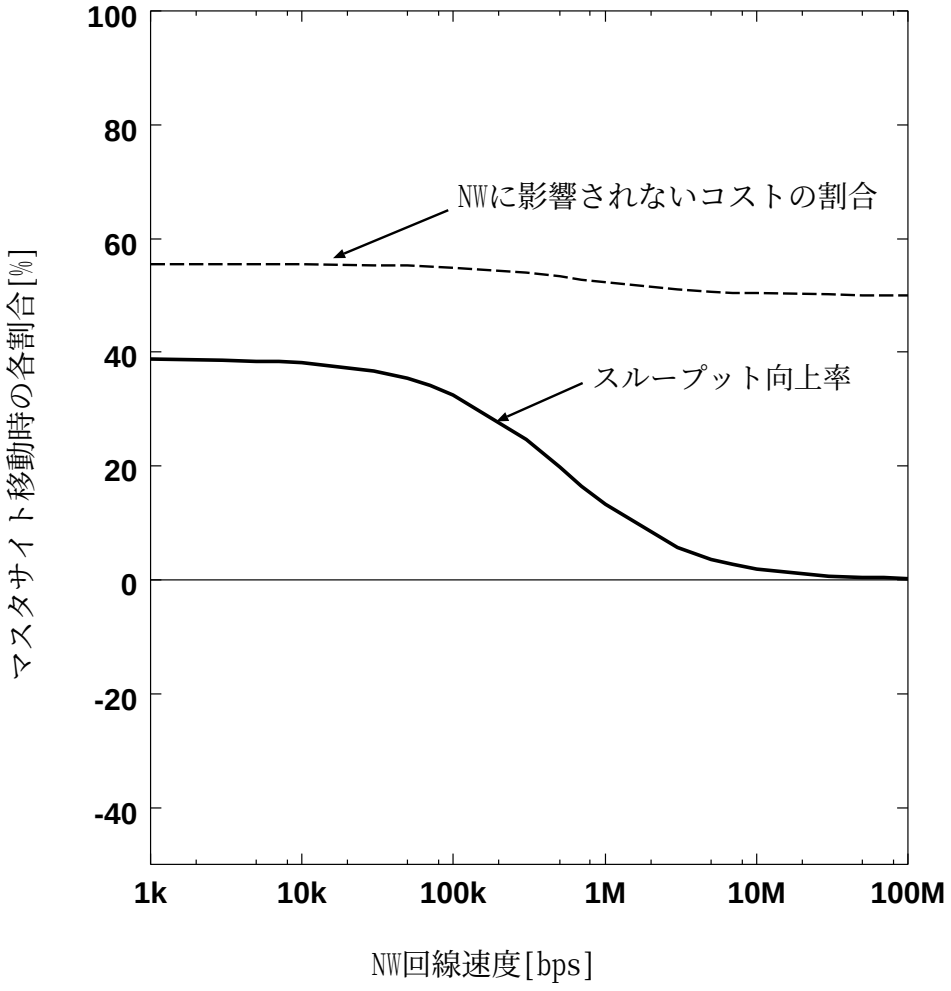

図 4.4: NW 回線速度とスループット向上率の関係

図 4.4 はネットワーク回線速度の大きさが、1k[bps]〜100M[bps] まで変化した時のスループット向上特性の傾向を示したものである。NW 回線速度以外の値については、基

本パラメータ値に従って試算を行なった。また図 4.3 よりトランザクション件数が 10,000 件付近にまで達すると、スループット向上率が収束するので、以降の試算ではトランザクション件数を 10,000 件として計算した。

グラフより、マスタサイト移動手法によるスループット向上効果は、回線速度が小さいケースほど大きくなることを確認した。また同時に、トランザクション処理コストに占める同期処理コストの割合により、向上率はある程度のところで頭打ちになることもグラフから読みとれる。マスタサイト移動手法では、マスタデータへのアクセス要求発生地点とマスタサイトの場所をなるべく一致させることにより、リモートサイトへのアクセス件数を相対的に減らし、通信コストを節減して処理効率を向上させる。このため、リモートアクセスによる通信コストが大きくなるほど、スループット向上効果が大きくなる。ゆえに回線速度が小さい場合ほど、スループット向上率が高くなる。ただし、ローカル DB 上での処理や、同期処理コストの内の更新差分のコミット処理分のコストについては、回線速度の影響を受けない。このため回線速度が小さくなっても、スループット向上率は、トランザクション処理コスト中の上記の固定処理コストの割合に応じて頭打ちになる。

図 4.5 はシステム中のサイト数とスループット向上特性の関係を示したものである。サイト数が増加すると、マスタデータがローカルサイトからアクセスされる割合が相対的に低下する。そして同時に、スループット向上率も低下する。またこれに加えて、更新の同期を取るためにアクセスしなければならないサイト数が増加するため、トランザクション処理コストに占める、同期処理コストの割合が相対的に増大する。そのためマスタサイトの移動により、圧縮が期待できる処理コスト分の割合が相対的に減少する。

スループット向上率を低下させる主要因として、NW 回線速度の増大、サイト数の増加、マスタサイト選定間隔の短縮が挙げられる。図 4.3 から図 4.5 までのグラフから上記の要因の中では、特にサイト数の増加がスループット向上率を最も大きく低下させることがわかる。これは前述のとおりサイト数の増加はスループット向上率に対して、2 つの負の影響を与えるためである。

一方サイト数の増加とは逆に、NW 回線速度の高速化は、上記の他 2 つの要因に比べてスループット向上に与える悪影響の度合が小さい。これは、回線速度が向上すると、通信コストの削減率が相対的に減少するのと同時に、マスタサイト移動手法における同期コストのオーバヘッドも相対的に小さくなるためである。

次に、サイト数の増加がマスタサイト移動に与える悪影響は、運用環境上の要因（サイ

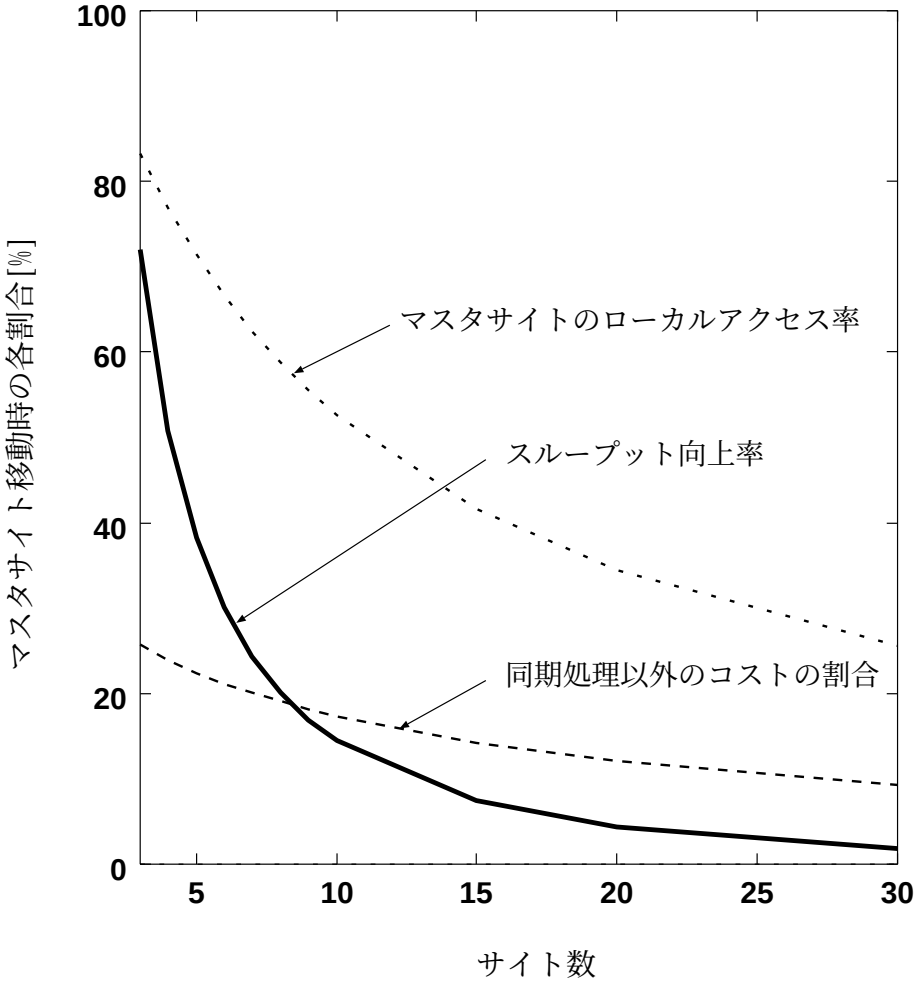

図 4.5: サイト数とスループット向上率の関係

ト・スキュー比、マスタサイト移動間隔）で、どこまで吸収できるのか検証するため、以下のような試算を行なった。マスタサイト移動効果に影響を与える要素の値を同時に変動させ、スループット向上率への影響度という観点で 2 つの要素の影響度を比較した。ただし、組み合わせ次の 2 種類である。（サイト数、サイト・スキュー比）、（サイト数、マスタサイト移動間隔）。

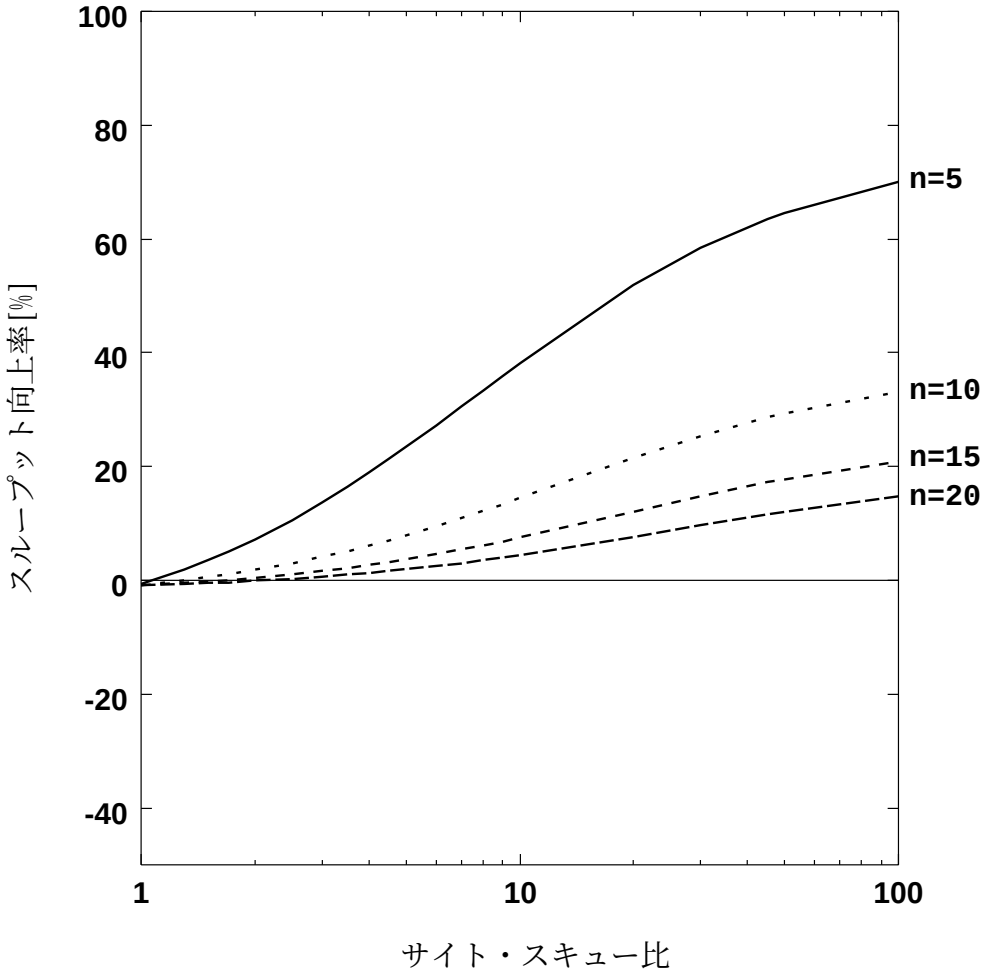

図 4.6: サイト数、サイト・スキューとスループット向上率の関係

図 4.6 はサイト数、サイト・スキュー比とスループット向上率の関係を示したものである。グラフからサイトスキュー比の増加に伴い、スループット向上率は増大し、ある程度までいくと（本試算では、サイト・スキュー比にしておよそ 20 、スループット向上率にして 60[%] 前後）スループット向上率が飽和する。一方、サイト数の増加はスループット

向上効果を抑制をする。特に本試算のケースの場合、サイト数が 5 から 10 までの範囲においてその度合が非常に大きい。

サイト・スキュー比が拡大すると、マスタデータがローカルサイトからアクセスされる割合が相対的に増加するので、通信コストの削減率が上昇し、スループット向上率が上昇する。一方、サイト間におけるアクセス頻度の偏りがない場合にマスタサイト移動手法を導入すると、他サイトとのメッセージ送信回数が増加し、トランザクション 1 件当たりの平均コストが増加する。このため、サイト・スキュー比が 1 のときは、スループット向上率は負になる[1]。また、トランザクション件数がマスタサイト移動間隔における件数より 1 件少ない時に、スループット向上率は 0 になる。この時は同期処理回数が一致し、かつマスタサイトの選定を行なう前に受付が全て終了するため、マスタサイト選定コストが余計にかからない。

以上の考察およびグラフの結果から、本試算のようなケース（システム）で、サイト数が 5 から 10 の範囲では、サイト・スキュー比よりもサイト数による影響の方が大きいといえる。実際、サイト数が 5 の時は、スキュー比が 10 前後でもスループット向上率が 60[%] と非常に高い値を示しているのに対し、サイト数が 10 をを越えてしまうと、スキュー比が 100 に拡大しても向上率は 20[%] 前後で頭打ちになってしまっている。

図 4.7 はサイト数、マスタサイト移動間隔とスループット向上率の関係について示したものである。

移動間隔が短く、サイト数が 20 を越えるような場合、スループット向上率は負になる。これは、マスタサイト移動による通信コスト削減分よりも、同期処理回数の増加による、メッセージオーバヘッド分の方が上回ってしまい、マスタサイト移動によりトランザクション処理コストが増大してしまうためである。逆に、移動間隔が拡大すると、1 件当たりのトランザクションコストに占めるメッセージオーバヘッドの割合が、相対的に小さくなるためスループット向上率は緩やかに上昇する。ただし、当然のことながら、上記のメッセージオーバヘッド分が、トランザクション処理コスト全体に比べてある程度以上小さくなると、スループット向上率は移動間隔に影響されなくなる（飽和する）。一方サイト数が増加すると、トランザクション処理コストに占める同期処理コストの割合が増大するため、向上率が抑制される。また、移動間隔の拡大により向上率は緩やかに上昇するのに対し、サイト数の増加によって向上率は、（特にサイト数が 5 から 10 の範囲で）急激に

[1] ただし、本試算におけるパラメータの設定値では、メッセージコストは同期処理コストに比べて十分小さいので、メッセージオーバヘッドによるマイナス分は非常に小さい

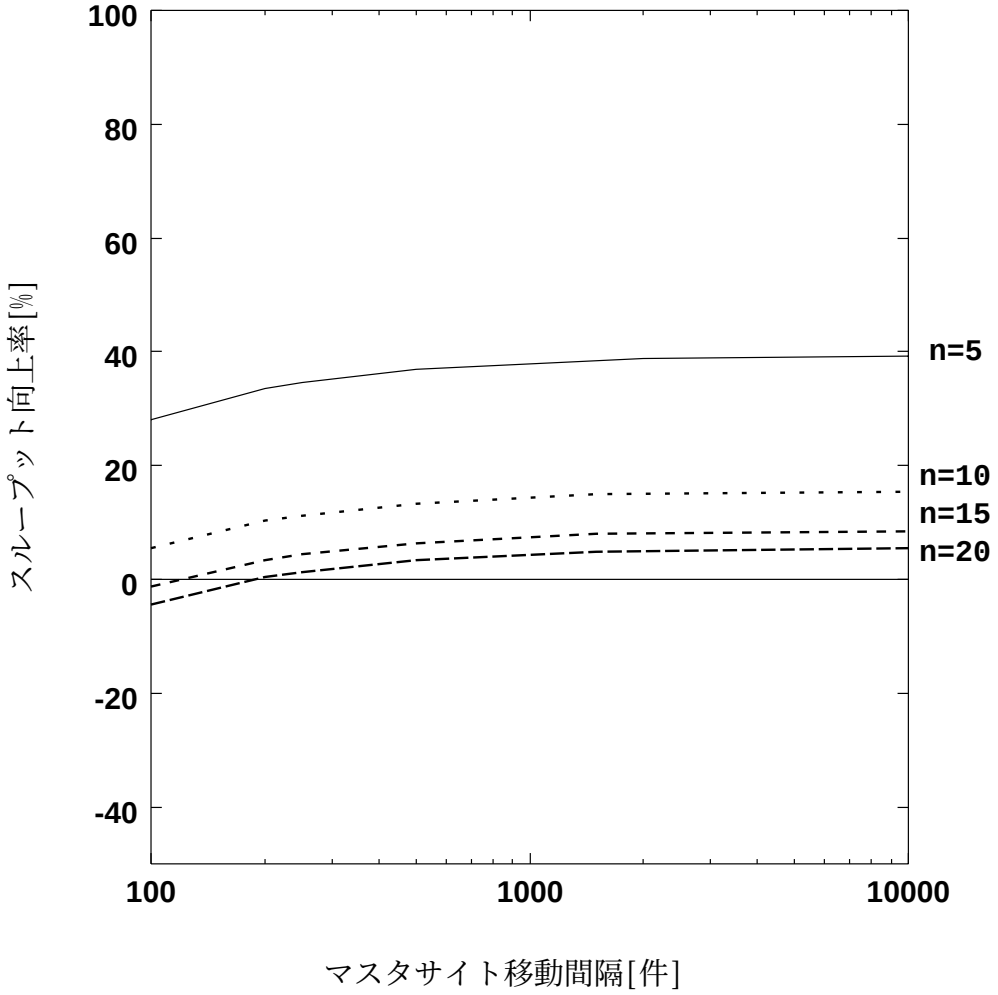

図 4.7: サイト数、マスタサイト移動間隔とスループット向上率の関係

抑制される。よって本試算のようなケースでは、サイト数は、移動間隔以上にマスタサイト移動効果の大きさに影響を与えると言える。

4.4.3 マスタサイト移動効果に関する試算結果のまとめ

4.4.2 節で行なった試算および得られたグラフに関する考察より、スループットの向上率の観点から、マスタサイト移動効果に関してまとめると以下のとおりになる。

- サイト・スキュー比、移動間隔の拡大はマスタサイト移動効果を増大させるが、ある程度までいくとその増大効果は頭打ちになる。これは、トランザクション処理コスト中に、マスタサイト固定／移動に影響されないコスト（同期処理コスト）が存在するためである。上記 2 つの要因によるマスタサイト移動効果の増大量の上限は、同期処理コストの比率によって決まる。

- マスタサイト移動間隔が、サイト・スキューの平均変動間隔に比べ若干短いかまたは、十分長いケースで、同期処理効率が最大になる。この時マスタサイト移動効果が最も引き出される。

- 本研究で考案したマスタサイト移動手法では、トランザクション処理コスト中に占める、同期処理コストの割合が大きいため（ほぼ 50[%] 以上）、移動効果はサイト数に最も影響を受ける。とりわけ、本研究における試算ケースでは、システムを構成するサイト数は 5 サイト以内が望ましい。スループット向上率は、サイト数の増加に伴い急激に低下する。サイト数が 5 から 10 の範囲の場合、とくにその傾向が強い。

- NW 回線の高速化と、サイト・スキュー比の拡大は、ほぼ等しくマスタサイト移動効果の大きさに影響する。よって、NW 回線が高速化されても、その割合とほぼ等しい割合のスキュー比の拡大があれば、マスタサイト移動効果の大きさは保存される。

第 5 章

実装に関する考察

5.1 マスタサイト移動のための付加モジュール

本研究では、マスタサイト移動効果に関する解析の他に、3 章で考察したマスタサイト移動手法の仕組みを基にして、試作モジュールの設計まで試みた。本章では、実装に関する考察と題して、マスタサイト移動手法を実現するための付加モジュールの構想、および設計の概要を示す。

まず本節では、試作モジュールの構想および内部構成の概要を示す。

図 5.1 は、マスタサイト移動手法を実現するためのモジュールを組み込んだ、分散データベースシステムの構想を示したものである。

データベースとユーザの間にフロントエンド・モジュール (以下 FEM) として、マスタサイト移動に必要な機能を実現するモジュールを配置する。ここでマスタサイトの移動を実現するため主要な機能として、以下のものが挙げられる。

トランザクション件数監視機能： 最もアクセス需要があるサイトを該当するリレーションのマスタサイトに選定するためには、毎回アクセス要求サイトをログに保存し、あるタイミングで要求サイトごとのアクセス件数を集計する機能が必要である。本機能はログ更新／集計モジュールと、ログを保存するデータベースにより実現される。

データ通信機能： リモートサイトへアクセス要求の転送や更新差分の転送、また、更新の同期をとる際に他サイトの FEP との連携を図るためのメッセージ送（受）信を行なう機能が必要である。データの種類ごとに通信ポートを分けておく。

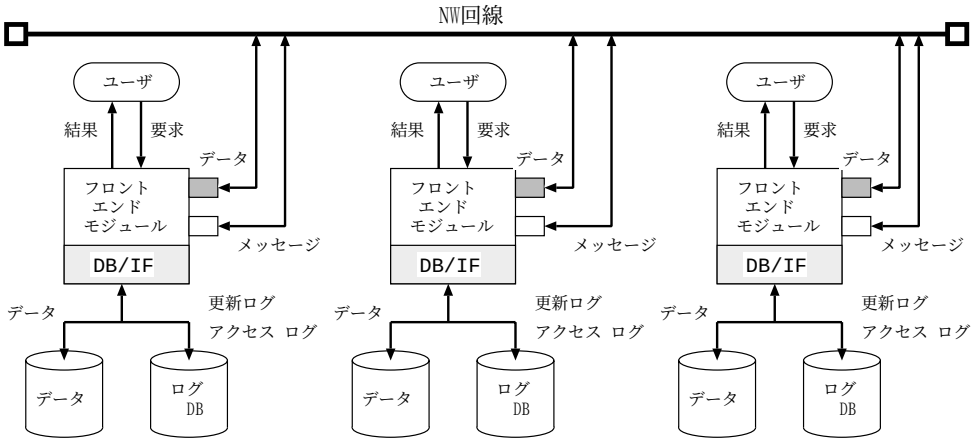

図 5.1: マスタサイト移動モジュールを組み込んだ分散データベースシステムの構想図

ローカルデータベースへのアクセス機能： 受け付けた要求情報（SQL 文）をローカルの DBMS に送り、結果をユーザにに返す機能が必要である。本機能は、使用する DBMS に予め用意されているデータベース・インターフェイスライブラリにより実現される。

マスタサイト選定／移動機能： サイト・スキューに応じたマスタサイト移動を行なうためには、アクセス・ログから、アクセス要求件数をサイトごとに集計し、各リレーションの次期マスタサイトを選定する機能が不可欠である。またこの機能と併せて、更新の同期をとる処理で、コミットの調停を行なう機能も必要である。

キューイング機能： マスタサイト移動処理中は通常のデータベース上の処理を一時中止するので、他サイトまたはローカルサイトのユーザから送られたアクセス要求情報を一時的に保管しておく機能が必要である。本機能は、アクセス要求内容を一時的に格納するログ格納用データベースと、ログを管理するプログラムにより実現される。

各機能ごとにモジュール化し、FEM 内部に組み込む。データ通信機能については、OS に用意されている通信用インターフェイスのより、実装の方法が異なるが、例えば UNIX の場合ソケットを用いる。またメッセージ通信とデータ通信でポートを分け、常に 2 つのポートを監視する。FEM の内部構成を図 5.2 に示す[1]。上記の必要機能とその機能を受け持つモジュールの対応は以下の通りになる。

[1] 図中のポート番号は便宜上適当に選択した値である

- トランザクション件数監視機能 → トランザクション監視・モジュール
- データ通信機能 → データ通信・モジュール
- ローカルデータベースへのアクセス機能 → DB アクセス・モジュール
- マスタサイト選定／移動機能 → マスタサイト移動、データ転送モジュール
- キューイング機能 → データ通信・モジュール、DB アクセス・モジュール

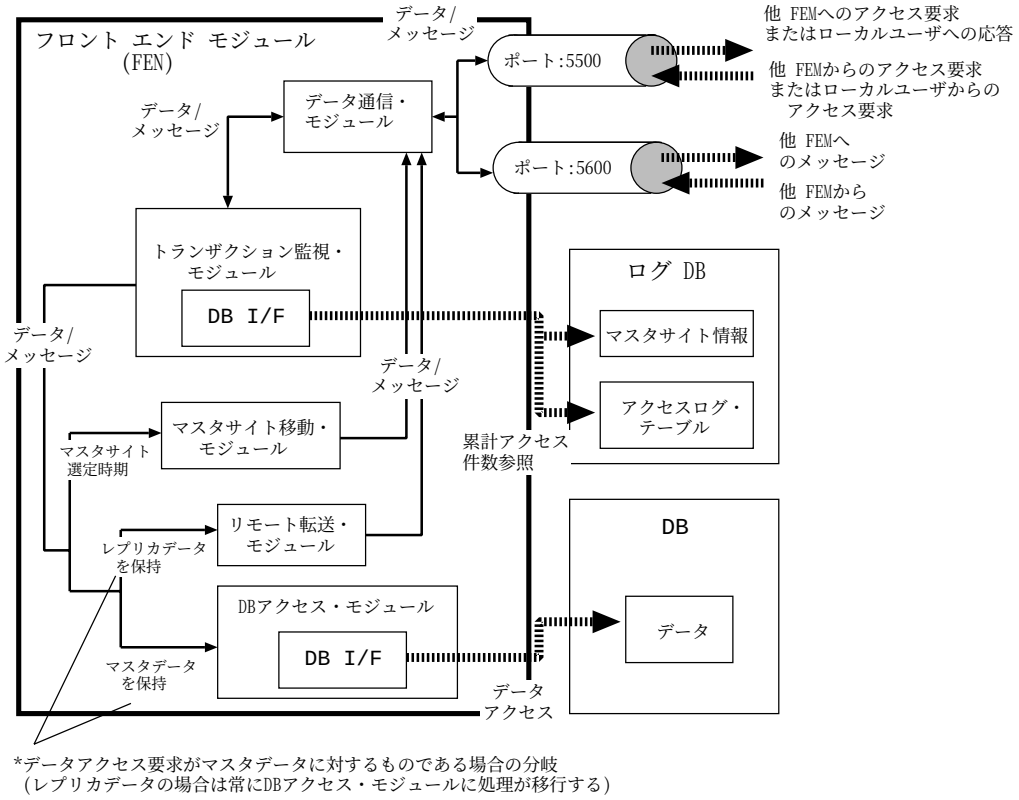

図 5.2: フロントエンド・モジュールの内部構成

5.2 処理の流れ

本章の最後として、FEM の動作について説明する。FEM の動作の大まかな流れは次の通りである。まずはじめに、他サイトの FEM からマスタサイト移動処理の通知が来て

いるかを調べ、来ていればマスタサイト移動処理に入る。来ていなければ、次に自サイトが保持するマスタデータに対するアクセス・ログから、マスタサイト選定時期かどうか判断する。現時点が選定時期であれば、そのまま選定処理に入り、必要であれば他のサイトと更新の同期を取りにいく。逆に選定時期でなければ、ユーザからトランザクションを受け付ける通常の DB 上での処理に移っていく。ここでトランザクションで該当するデータのマスタデータを自サイトが保持しておらず、かつトランザクションがマスタデータにアクセスする類のものである場合は、該当データについてマスタサイトになっているサイトの FEM に処理を依頼する。また上記以外の場合は、ローカルの DBMS に処理を依頼する。以上の処理を繰り返していく。図 5.3 は処理の流れの概略図である。

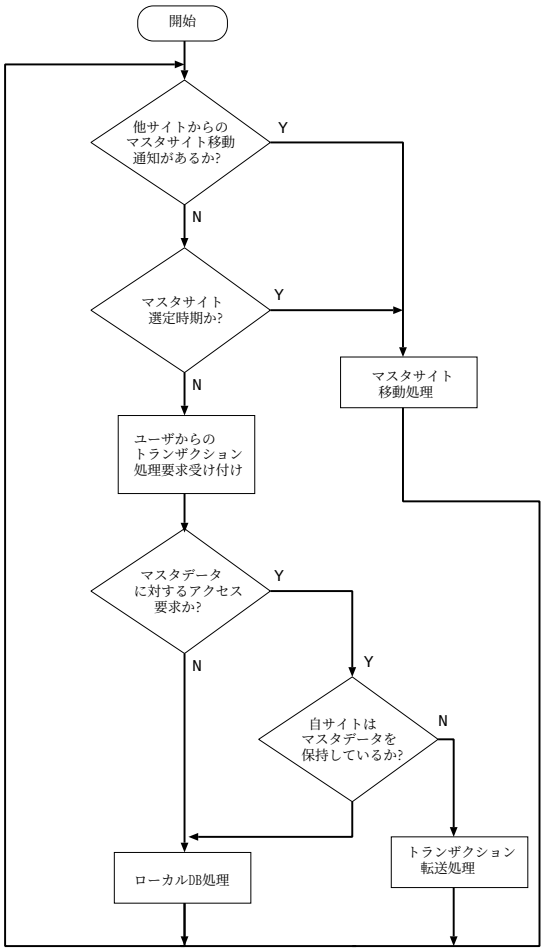

図 5.3: フロントエンド・モジュールでの処理の流れ

第 6 章

おわりに

6.1 まとめ

本研究では、サイト・スキューが時間的に変動する場合に露呈する、Lazy-Master-Replication 手法の非効率な部分を補う手段として、マスタサイトの移動を選択した。そしてマスタサイト移動手法の仕組みを考案し、本手法導入後のシステムのスループット向上率の解析、試算を行ない有用性を検証した。その結果、特定の条件下で非常に高いスループット向上効果が期待できることを確認した。なお、上記の解析、試算結果並びにマスタサイト移動手法の可能性に関する考察については、DEWS'98 にて発表する予定である。

以下に、スループット向上率の試算、解析で確認できた内容の内、主要なものについて示しておく。

- マスタサイト移動手法によるスループット向上効果が、十分に引き出されるためには、マスタサイト移動間隔が、サイト・スキューの平均変動間隔に比べ若干短いかまたは、十分長くなければならない。

- システムを構成するサイト数はできれば、5 サイト以内が望ましい。スループット向上率は、サイト数の増加に伴い大幅にに低下する。サイト数が 10 を超える場合、サイト・スキュー比が 100 の場合でも、スループット向上率にして約 20[%] 程度の効果に留まる。

- NW 回線の高速化によるスループット向上率低下の度合と、サイト・スキュー比の増大に伴うスループット向上率上昇の度合はほぼ等しい。このため、NW 回線の高

速化率をサイト・スキュー比の拡大率が上回る場合、スループット向上率の上昇が期待できる。

6.2 今後の課題

本研究で考案したマスタサイト移動手法では、更新の同期をとる際、更新差分情報の転送については、非圧縮で行なうことを想定している。このため、トランザクション処理コストに占める、同期処理の比重が非常に大きく、その結果、小数のサイト（3〜5 サイト）で構成されたシステムでしか実用的な効果を得られないという試算結果を得た。更新差分情報の圧縮については、既に実装されている商用システムが実在するので、本手法に、既に確立されている更新差分の圧縮手法を採り入れることができれば、サイト数が 10 におよぶシステムでも対応できると考えられる。また、本研究における解析で選択したレプリケーションのシステム形態は、最も単純な 1 対 1 型である。今後、より現実に近い水平分割型のようなシステム形態をモデルにした解析が必要である。

最後に、本論文では解析内容に相当のページを割いてきたが、本研究の解析内容を、より実際に即したものにしていくため、5 章で示した設計に準じたフロントエンドモジュールの試作が待たれる。

謝辞

本研究を進めるにあたり、終始熱心な御指導を賜わりました横田治夫助教授に心よりお礼申し上げます。

適切な御指導、御助言を頂きました日比野靖教授に深く感謝致します。さらに、杉野栄二助手、宮崎純助手をはじめ、日比野／横田研究室の皆様には種々の面でお世話になりました。ここに深い感謝の意を表します。

最後に、私を快く大学院に留学させて頂き、貴重な研究の機会を与えて頂いたうえ、生活面でも全面的に支援下さいましたコマツソフト株式会社に心より感謝致します。

参考文献

[1] 野口正一, 疋田定幸「分散型データベースシステム入門」, オーム社 , 36-50pp,1989.

[2] C.J.DATE "*An Instruction of Databese System*", Addison-Wesley, 596-605pp,1990.

[3] Jim Gray and Andreas Reuter "*TRANSACTION PROCESSING:CONCEPTS AND TECHNIQUES*", MORGAN KAUFMANN, 562-573pp,1993.

[4] 中村 正弘「分散データベース」, NIKKEI ELCTRONICS (no.609), 101-110pp,1994.

[5] 中川路 哲男「OSI と UNIX 分散トランザクション処理技術解説」, ソフトリサーチセンター刊 , 1996.

[6] Jim Gray,Pat Helland,Patrick O'Neil and Dennis Shasha "*The Dangers of Replication and a Solution*", SIGMOD'96, 173-182pp,1996.

[7] San-Yih Hwang and Keith K.S.Lee and Y.H.Chin "*Data Replication in a Distributed System: A Performance Study.*", DEXA, 708-717pp,1996.

[8] A.S. タネンバウム （邦訳：引地信行 ／他）「OS の基礎と応用」, トッパン刊 , 562-569pp,1995.