

修士論文 2002 年度 (平成 14 年度)

無線携帯型計算機に適した
トランザクション処理モデルの研究

慶應義塾大学大学院 政策・メディア研究科

梅染 充男

修士論文要旨 2002 年度 (平成 14 年度)

無線携帯型計算機に適した トランザクション処理システムの研究

論文要旨

近年、計算機技術やインターネット技術、トランザクション技術の普及と発展が急速に行われた。これにより、ユーザはインターネットに接続された計算機端末がそこであれば、時と場所を選ばず、買い物や証券取引、他様々なサービスの享受が可能であり、そして現在は一般的に利用されている。

特に最近では、携帯電話をはじめとした無線を媒体として通信を行う携帯型計算機の利用者の増加に加え、NTT DoCoMo の i-mode 等、無線環境下にて実行される様々な電子商取引サービスが登場し、無線携帯型計算機を媒体に実行されるトランザクション処理システムの必要性が非常に高まりつつある。しかしながら有線環境に比べ、無線を媒体に通信を行う場合、低通信速度や高パケットロス率、高移動性、バッテリーによる駆動時間の制限等、様々な制約や問題等が、電子商取引システムの核となるトランザクション処理システムに対し、性能低下を与えてしまうことも事実である。

本研究では、無線携帯型計算機を介して電子商取引を行うユーザに対し、トランザクション処理に関して無線特性が及ぼす悪影響を可能な限り意識させないトランザクション処理システムの設計および実装、評価について述べる。上記トランザクション処理システムでは、サーバ端末が管理するデータベースの一部とエージェントを、ユーザが操作するクライアント端末に分離・転送することで、無線通信を抽象化しトランザクション処理を実行することが可能である。また、エージェントのクライアント端末、サーバ端末間の通信においても、送信したメッセージやサーバでの処理等を逐次ステートとして保持することで、無線通信を介したインタラクションの不具合を補完することを可能とした。

評価において、各データベースのサイズやトランザクション処理に要する時間、エラー発生から復帰までの時間等の値を設定する評価用シミュレーションプログラムを実装した。通信の障害発生率やデータベースのサイズや、各種処理に所要する時間等を複数設定し、本トランザクション処理システムを含む3つのトランザクション処理システムと比較し、評価した。これにより、全体的な処理の長いトランザクション処理や、トランザクション処理に対し、障害が発生する確率が高い環境でのトランザクション処理に優れていることを示した。また、関連研究との評価は定性的評価にて行い、本トランザクション処理システムが優れた性能を示す環境を示した。

近い将来、無線携帯型計算機により電子マネーの取引がされるようになることで、有線通信型計算機以上に、無線携帯型計算機を介して様々な電子商取引が行われることが一般的になる時代が到来する。本研究は、以上の様な環境下にて実行される電子商取引システムの核となるトランザクション処理システムの発展に大きく貢献できる。

キーワード：

トランザクション処理, 携帯電話, 小型端末, 無線特性

慶應義塾大学大学院 政策・メディア研究科

梅染 充男

Abstract of Master's Thesis Academic Year 2002

The Split Database Transaction Processing System for Mobile Environment

Summary

Recently, Calculating, Inter-net or Transaction Technology has developed and popularized rapidly. Because of that, the users have been able to shopping, stock marketing or taking advantages of any other services at any time any place. Therefore those technologies are so popular at the present.

Especially nowadays, numbers of users for the wireless portable computer such as mobile phones are increasing. Also, many types of electronic services that based on the wireless system, such as NTT DOCOMO's I - mode are involved into the market. Therefore importance and need of having transaction system that execute under the wireless portable computer are continuously rising. However, compare to the wired system, the wireless system has many restrictions or problems, for instance lower transaction speed, high packet-loss rate, high chance of moving or limitation on battery. These troubles cause performance deterioration to the transaction system, which becomes core of E-commerce system.

In this study, I would like to introduce the transaction system that has no aware of recognizing bad influences by wireless system and show how to built, set and result of valuation for that. As mentioned above transaction system, separate and transfer the part of database and agent that controlled by server terminal to the client terminal that controlled by users. With this method it had made possibility to abstract the wireless communication and execute the transaction. Also between agent's client terminal and server terminal had been able to complementing interaction's obstacle with wireless communication by maintenance all transaction and transferred messages as a state.

For the valuation, I used simulation program that calculate the size of database, transaction time, time of the recovery from errors and set up couple of hindrance occurs rate for the communication, size of database and interaction numbers to compare between flat transaction system that hadn't take wireless portable computer's characteristic into consideration and this transaction system to valuate. As a result, we'd found out our new system has better performance in case of many interaction transaction or high possibility hindrance transactions. Also we'd evaluated other related studies by using fundamental analysis and evaluation and it found out our transaction system has an environment that secures the high performance.

In the near future, the time is coming for many different kind of E-commerce becomes more popular by using wireless portable computers more than using wired portable computers. I believe this study will be able to contribute to develop transaction system that core of E-commerce.

Keywords :

Transaction, Handy Phone Mobile Computer Wireless

Keio University Graduate School of Media and Governance

Mitsuo Umezome

目次

第1章	序論	1
1.1	研究背景	2
1.2	本研究の目的と意義	3
1.3	本論文の構成	3
第2章	トランザクション処理	5
2.1	基礎概念	6
2.2	トランザクションの特性	7
2.2.1	原子性 (Atomic)	7
2.2.2	一貫性 (Consistent)	8
2.2.3	隔離性 (Isolated)	8
2.2.4	耐久性 (Durable)	9
2.3	2相コミット	9
2.4	トランザクション処理システム	11
2.4.1	トランザクションクライアント	11
2.4.2	トランザクションサーバ	12
2.5	トランザクション処理の性能	12
2.5.1	可用性による性能評価	12
2.5.2	TPC ベンチマークによる性能評価	13
第3章	問題意識と解決手法	16
3.1	問題意識	17
3.1.1	無線通信特性	17
3.1.2	無線媒体の変移	19
3.1.3	無線携帯型計算機特性	20
3.2	解決アプローチ	21
3.2.1	エージェントベーストランザクション処理	22
3.2.2	非アポートトランザクション処理	22
3.2.3	問題点	22
3.3	関連研究	23
3.3.1	NTMS [15]	23
3.3.2	Kangaroo Transaction	23

3.3.3	2 Phase Recovery Transaction [17], [18]	24
3.3.4	Compensatable Transaction [19]	24
3.3.5	TranSuMA [20]	24
第4章	設計	26
4.1	Split Database Transaction Processing System の概要	27
4.1.1	コンポーネント	28
4.1.2	機構	29
4.1.3	データベース	30
4.2	トランザクション処理機構	31
4.2.1	トランザクション処理開始時の動作	32
4.2.2	無障害時の動作	34
4.2.3	トランザクション処理終了時の動作	36
4.2.4	障害発生時の動作	36
4.3	移動制御機構	36
4.4	トランザクション復帰機構	37
4.4.1	無線通信間に障害が発生した場合	38
4.4.2	トランザクションクライアントにおける障害が発生した場合	38
4.4.3	トランザクションサーバにおける障害が発生した場合	38
4.4.4	トランザクションエージェントにおいて障害が発生した場合	38
4.5	エージェント管理機構	39
4.5.1	トランザクションサーバ側での動作	39
4.5.2	トランザクションクライアント側での動作	39
4.6	アプリケーション保存機構	40
4.6.1	アプリケーション保存までの動作	40
4.6.2	トランザクション処理短縮時の動作	41
第5章	評価	44
5.1	定量的評価	45
5.1.1	評価目的	45
5.1.2	測定方法	45
5.1.3	測定結果	49
5.1.4	考察	52
5.2	定性的評価	53
5.2.1	評価方法	53
5.2.2	評価結果	54
5.2.3	考察	56
第6章	結論	57
6.1	今後の課題	58
6.1.1	セキュリティの強化	58

6.1.2	Split Database Transaction Processing System の一貫性保証	58
6.2	まとめ	59

目次

1.1	携帯 IP 接続, 携帯電話契約者数数の推移	2
2.1	トランザクション処理アプリケーションの部品	6
2.2	2相コミットにおける原子性の保証	10
2.3	トランザクション処理システムのソフトウェア構成	11
3.1	無線通信の切断とトランザクション処理の関係	18
3.2	通信切断確率とフラットトランザクション処理所要時間のグラフ	19
4.1	通常のトランザクション処理モデルの概要	27
4.2	エージェントベーストランザクション処理モデルの概要	27
4.3	Components of Split Database Transaction Processing System	28
4.4	正常時における Split Database Transaction Processing System の処理の流れ	32
4.5	Example of Usual Transaction Processing	34
4.6	Example of Split Database Transaction Processing Between Transaction Client and Transaction Agent	35
4.7	Authentication of Reconnection	37
4.8	アプリケーション保存までの動作	41
4.9	トランザクション処理短縮時の動作	42
5.1	Relation of Transaction Processing Application	50
5.2	Relation of Split Database Size	51
5.3	Relation of Wireless Transmission Speed	52
5.4	Relation of Disconnection Rate in Wireless Environment	53

表 目 次

2.1	トランザクション処理アプリケーションの例	7
2.2	可用性とトランザクション処理システムがダウンする時間の関係	13
4.1	Transaction Server State	33
4.2	Transaction Client State	33
4.3	Saving Application Client State	40
4.4	Saving Application Server State	40
5.1	Parameters of Normal Flat Transaction Processing System	46
5.2	Parameters 1 of Split Database Transaction Processing System	47
5.3	Parameters 2 of Split Database Transaction Processing System	48
5.4	定性的評価を行うトランザクション処理システム	54
5.5	定性的評価の指標	55
5.6	定性的評価	56

第1章

序論

本章では，本研究の背景および，研究の目的と意義について述べる．次に本論文の内容構成について述べる．

1.1 研究背景

パーソナルコンピュータや携帯電話等の情報端末の技術的進歩，インターネットの進化と普及により，ここ数年間に電子商取引が急速に実用のもとなり，既に市場取引において重要な地位を築いている．従来の電子商取引においては，重要な情報の送受信や秘密を要する情報のやりとりにおいては専用回線を利用した VAN(Value Added Network) を利用するケースがほとんどであった．しかし，近年では経済コスト面での大きなメリットにより，インターネットを利用した電子商取引が急激に拡大している．インターネットを利用することは，高価な専用回線を敷く必要がなく，また通信回線で結ばれた相手となれば，国境を問わず，誰とでも，いつでも取引ができるというメリットを持っている．経済産業省 [1] によると日本における電子商取引の市場規模は 1999 年の時点で約 8.7 兆円，2003 年には 71.6 兆円に拡大されると予測されている．インターネットの利用価格が低下し，また，高いスループットやセキュリティが向上するに従い，ますますインターネットによる電子商取引の範囲は拡大していくだろう．

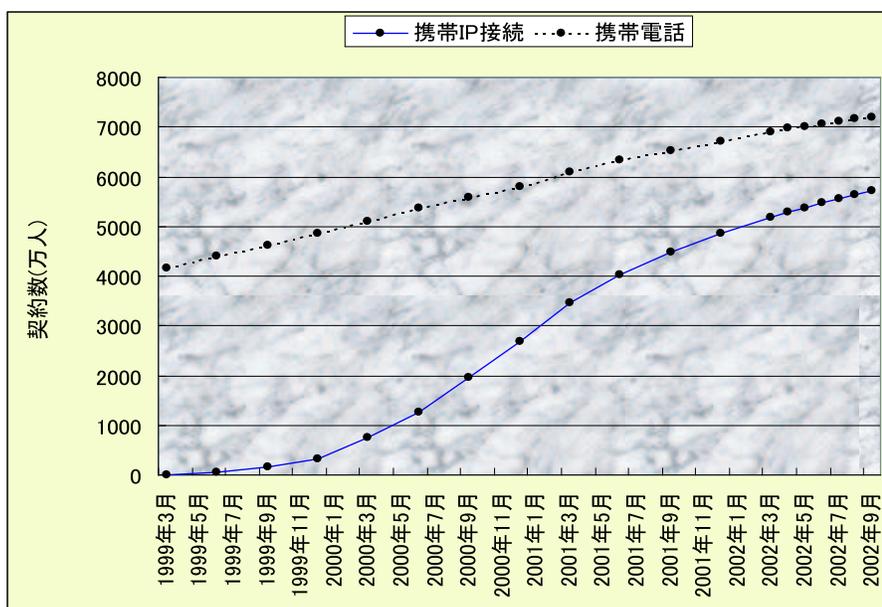


図 1.1: 携帯 IP 接続，携帯電話契約者数数の推移

一方で，近年携帯電話や PHS，PDA，ノート PC 等といった無線携帯端末の普及も目覚ましい．携帯電話の普及に関しては，電気通信事業者協会 [2] によれば図 1.1 のように，PHS を含む携帯電話の加入者数が急激に増加している事が読み取れる．2002 年 9 月の時点で日本の携帯電話普及率は 50 % を超え，近いうちに 8000 万人程度まで増加し安定期に入ると推測されている．それから，無線携帯端末の技術的革新も著しい．例えば携帯電話に関して，数年前は一般的に携帯電話は会話をするだけの目的で使用されていたが，図 1.1 のように，現在市場に出回っている携帯電話のおよそ 80 % の割合において，NTT DoCoMo[3] の i-mode をはじめとしたインターネットを利用した電子メールや Web 等の

各種サービスの使用が標準となっている。また、NTT DoCoMo は JavaVM を実装した携帯電話を実現した。さらに、次世代携帯電話規格である IMT-2000[4] の登場により、数年以内に従来より高い音声品質、高速データ通信、通信品質に加え、国際ローミングが一般的なものとなるだろう。世界各国は IMT-2000 に対応した携帯電話と携帯電話網の開発を急いでいる。

NTTDoCoMo 等は、10 年以内に携帯電話を利用する事で電子マネーを扱えることが可能となると宣言している。これが実現すれば、Bluetooth[5] や赤外線を利用する事で電車やバスの運賃支払いや、煙草や飲料等を販売する自動販売機への支払い等も携帯電話を利用する事でユーザが意識せずに自動的に行う事が可能となるだろう。携帯電話自体の発展も目覚しく、形態性を維持するため、非常に小型かつ軽量でなければならないという制約があるにも関わらず、マルチメディアファイルの扱い等も可能となり、性能も有線に接続された一般的なパーソナルコンピュータに段々と近づいている。

インターネットをネットワーク基盤とした電子商取引の範囲が拡大することに加え、インターネットに接続可能である無線携帯端末が一般に急速に普及している。この事から、無線携帯端末により実行される電子商取引の市場規模は非常に高まり、また端末自体の技術的発展により複雑で高負荷である電子商取引も実現され、今後は無線携帯端末より実行される電子商取引システムの需要はさらに高まるだろう。しかしながら、従来の有線環境下にて電子商取引を実行する場合に比べ、無線携帯型計算機より電子商取引を行う場合、無線通信特性や高移動性、貧弱な計算能力等、無線通信や無線携帯端末の一部の特性が電子商取引に対し性能の低下をもたらしてしまう。この問題に対する解決方法としては、工学的な無線技術の改善を行うアプローチと、電子商取引システムの核となるトランザクション処理システムにおける耐故障性の向上を図る手段があり、本研究では後者のアプローチを採用する。

1.2 本研究の目的と意義

本研究では、無線携帯端末を使用して複雑な処理を行う電子商取引システムを実行する際、ユーザに可能な限り無線特性による性能低下を感じさせない電子商取引システムをトランザクション処理システム内で実現する。また、無線環境を想定した電子商取引に関連する研究に対し貢献する。

また、無線特性による様々な欠点を改善したトランザクション処理システムを利用する事で、ユーザは有線環境下と比較しても、無線端末より実行されるトランザクション処理に対する性能低下をほとんど感じさせなくなり、無線環境下から実行する電子商取引の定着、普及にも貢献できる。

1.3 本論文の構成

本論文は本章を含め全 6 章から成る。第 2 章において、本研究を説明していく上で、重要な概念であるトランザクションと把握すべき特性について解説する。第 3 章にて無線

環境下からトランザクション処理を行った場合に起こり得る様々な問題点を列挙し，関連研究では，各研究においてそれぞれどのようなアプローチにより，問題を解決，改善したかをまとめる．第4章にて，無線携帯型計算機に適したトランザクション処理システムの設計について詳述し，第5章において作成したシミュレーションプログラムより本トランザクション処理システムの基本的性能を評価し，また他研究との性能評価を定性的評価により示す．第6章では本研究についてのまとめと今後の課題について言及する．

第2章

トランザクション処理

本章では、本研究において非常に重要な概念である「トランザクション」について説明する。第1節では、トランザクションに関する概要を述べる。第2節では、全てのトランザクションが満たさなくてはならない4つの重要な性質を解説する。第3節で無線携帯型計算機の高移動性に関連する、複数のサーバ間のリソースを一貫させるために不可欠な2相コミットについて言及する。第4節にて、サーバ・クライアントモデルのトランザクション処理システムを説明する。第5節では提唱するシステムを理解する上で、必要なトランザクションプログラムを処理するトランザクション処理システムのアーキテクチャの基本を説明し、そのトランザクション処理システム例を挙げ、解説する。第6節にてトランザクション処理システムの一般的な評価方法について述べる。(参照 [6], [7], [8])

2.1 基礎概念

トランザクションは通常は企業と個人の間で行われるやり取りを意味する。そのやり取りにおいて、金銭・製品・情報サービスの要求等が交換される。そこでは、それらの要求から何が生じたかを記録する必要がある、スケーラビリティを考慮し高い信頼性でコストを抑えるために、この記録作業はコンピュータによって自動処理されることが多い。この場合、トランザクションとは共有データベースにアクセスすることによって管理業務を実現するプログラムのことであり、正確にはオンライントランザクションと言う。ここで重要な点は、トランザクションはプログラムの実行である点である。つまり「トランザクション」という言葉の正確な定義は、トランザクションの実行を要求するためにコンピュータシステムに送られる要求メッセージのことである。また、本論文ではこのようにコンピュータの使用を前提としている。

コンピュータを利用することで、ビジネス活動を自動化し、必要な機能を実行するように設計されたトランザクションプログラムの集まりをトランザクション処理アプリケーションという。トランザクション処理アプリケーションは図2.1のように、3つの部分から成り立つ。

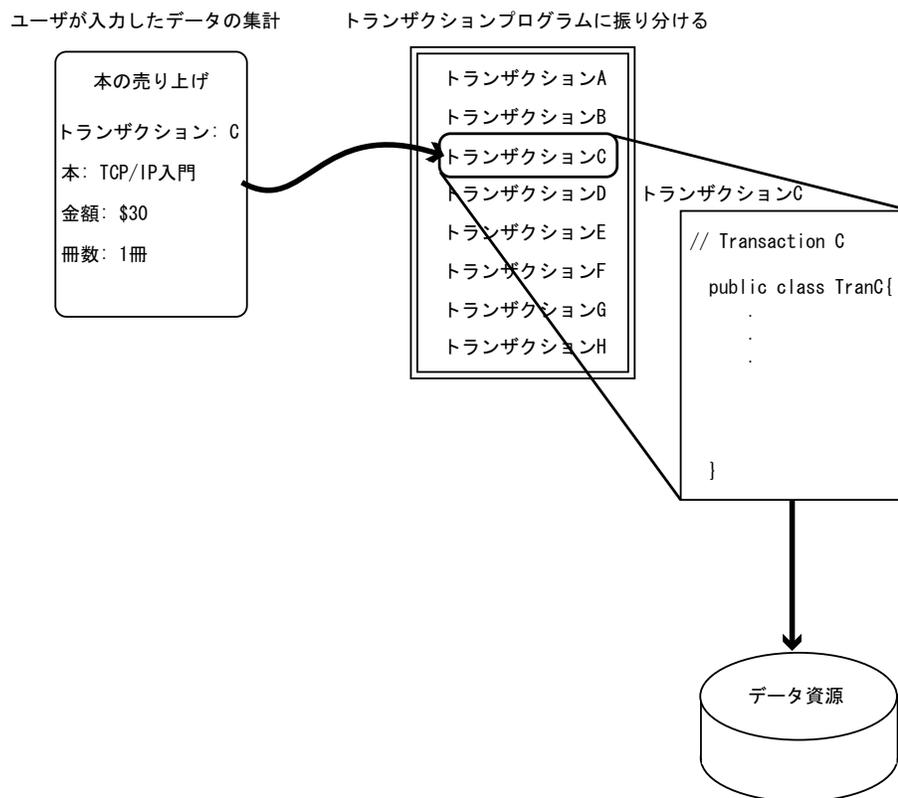


図 2.1: トランザクション処理アプリケーションの部品

- 最初の部分は、トランザクションプログラムへの入力を集める。例えば、ユーザへ入力をするためのフォームを表示し、どの主のトランザクションプログラムを走ら

せるかを選択できるメニューを表示する。この部分は、大抵ユーザによって入力されたデータが正しいものであると保証するための何らかの正当性試験を含んでいる。そして、メニューの選択やフォームへの入力等といったユーザからの入力を集めて、それを通信メッセージへ変換する。

- 第2の部分は、最初の部分からの通信メッセージを受け取り、対応するトランザクションプログラムの呼び出しへと変換する。分散トランザクション処理システムでは、プログラムが実際に存在する実行環境へのメッセージ送信という形で実現される。
- 第3の部分は、ユーザの期待した仕事の要求を処理するトランザクションプログラムである。典型的には、共有データベースへの読み出し、書き込みを行い、あるいは、他のプログラムを呼び出して処理を行う。さらに、入力を要求として受け付けた装置に応答を送り返すこともある。

現在、トランザクション処理は経済のほとんどの分野をカバーしている。実際、表 2.1 のような、多くの種類のトランザクション処理アプリケーションが存在する。

表 2.1: トランザクション処理アプリケーションの例

アプリケーション	トランザクションの例
銀行	口座へ入金する
証券取引	株を 100 株購入する
保険	保険料を払う
在庫管理	発送の受け付けを記録する
製造業	組立工程を記録する
小売業	売り上げを記録する

2.2 トランザクションの特性

トランザクションが必ず満たさなくてはならない重要な特性は4つあり、それぞれ原子性 (Atomic)、一貫性 (Consistent)、隔離性 (Isolated)、耐久性 (Durable) である。この特性を頭文字をとって ACID と呼ぶ。以下に各々の性質の解説とその性質を保証しなくてはならない理由を述べる。

2.2.1 原子性 (Atomic)

トランザクション処理の終了状態は、正常に全ての処理を完了した状態か、実行する直前の状態(または「全か無か」)でなくてはならない。トランザクションプログラムの

一部だけが実行される可能性があってはならない。トランザクションの正常な完了をコミット、失敗をアボートと呼ぶ。

2.2.2 一貫性 (Consistent)

トランザクションの第2の性質は一貫性である。トランザクションプログラムはデータベースとの一貫性を保たなくてはならない。即ち、最初に一貫性を保っているデータベース上で、あるトランザクションだけを実行するなら、そのトランザクションが終了したときもまたデータベースは一貫性を保っている。

一貫性を保つとは、内部的に一貫性が保たれていることを意味する。これはデータベース用語で整合性制約を満たすことで、整合性制約には多くの条件がある。以下にその一例をあげる。

- **全ての主キーが一意であること**
ex) どの証券取引のサービス利用者も同じ利用者番号を持っていない。
- **データベースは参照性合成を持つこと**
ex) 例えば、注文レコードが参照する部分のレコードと顧客のレコードが実際に存在している。
- **ある述語が満たされること**
ex) 各部門での支出の合計はその部門の経費予算と同じかそれ以下である。

トランザクション処理において一貫性が保証されない、電子商取引をはじめとしたあらゆるやり取りが成立しない。例えば、ある人が銀行の自分の口座に金額を入金した際、別の口座にその金額が入金されたり、自分の口座に異なった金額が入金される等の場合が一貫性が保証されていないトランザクション処理である。

原子性・隔離性・耐久性と異なり、一貫性はトランザクションプログラムとそれらを実行するトランザクション処理システムで保証しなければならない性質である。つまり、トランザクション処理システムはトランザクションの集合が、原子性・隔離性・耐久性を持つことは保証するが一貫性までを維持することは出来ない。トランザクションプログラムが一貫性を維持することを保証することはアプリケーションプログラマの責任である。

2.2.3 隔離性 (Isolated)

トランザクションの3番目の性質は隔離性である。一連のトランザクションが隔離されているということは、複数のトランザクションを実行した場合のシステムの効果が、一度に1つずつ実行することと同じであることである。隔離性の技術的な定義は直列可能性である。ある一連のトランザクションの実行が直列可能であるとは、トランザクシ

ンを直列に並べて、全てのトランザクションを時間的に重複することがないように実行することと同じである。

隔離性を備えていないトランザクション処理システムの例をあげる。ある銀行のシステムで、別の2つのトランザクションがある口座に残った最後の100ドルを引き出そうとしている場合、双方のトランザクションがそのどちらも口座の更新をしないうちに残高を読み出したとする。この時、両トランザクションもその要求を満たすだけの十分なお金があると判断し、それぞれが100ドルを引き出すことになってしまう。以上は明らかに間違った結果であり、直列可能でない結果である。直列にトランザクションを実行した場合、最初に実行されたトランザクションのみが最後の100ドルを引き出せるだろうし、2番目のトランザクションは口座が空になっていることを知らされなければならない。

2.2.4 耐久性 (Durable)

トランザクションの第4の性質は耐久性である。耐久性とは、トランザクションの実行が完了し、全ての更新がディスク等の記憶メディアに格納されたとき、それはトランザクション処理システムの障害の後でも残っていることを意味する。というのは、たとえトランザクションプログラムやオペレーティングシステムが障害を起こしても、一度トランザクションがコミットしていれば、その結果は引き続き記憶メディア上に格納されており、障害からのシステム復帰後にも参照することが出来る。

通常、各トランザクションはユーザとそのサービスを供給している企業の間での契約に当たるサービスを提供しているため、耐久性は重要である。例えば、あるユーザがある口座から別の口座へとお金を移しているとする、その人はトランザクションから実行されたと応答が来たなら実際にその結果が永続的であることを期待するだろうし、2つの口座間でお金が移動したことはユーザとシステムの間での法的な合意である。そのため、トランザクションが実際に更新が何らかの不揮発性の装置に格納されたことを確認し、トランザクションの実行終了後もその更新が失われえないことを保証することが不可欠となる。さらに言えば、結果の耐久性は多くの場合数年は管理されなければならない。

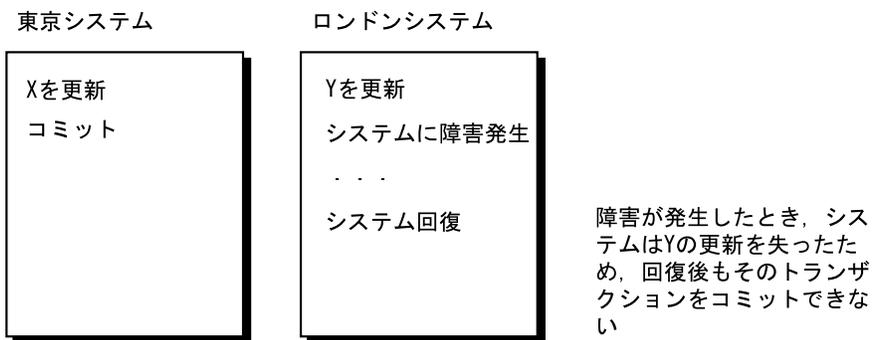
2.3 2相コミット

トランザクションが2つ以上に分散された環境でシステムのデータを更新する際でも、原子性の属性は保証しなくてはならない。即ち、両方のシステムが更新を永続的に記録したか、あるいは両方とも記録しないかを保証しなくてはならない。これはシステムは独立に障害・回復を発生させるため、非常に困難な問題である。この解決策が2相コミットと呼ばれる特別なプロトコルであり、それはトランザクションマネージャと呼ばれる特別なモジュールによって実行される。

この問題点は、トランザクションはその更新コミットをあるシステム上で行えるが、第

2のシステムはトランザクションがそこでコミットする前に障害を起こす可能性があるということである。このケースでは、障害を起こしたシステムが回復した際に、そのトランザクションをコミットすることが出来なくてはならない。そのトランザクションをコミットするためには、トランザクションの更新のコピーを持たなくてはならない。障害を起こした際には、システムのメインメモリーの内容が失われる可能性があるため、2相コミットを使ってトランザクションの更新の永続的なコピーを障害の前に格納し、回復後も保持しておけるようにしなくてはならない。トランザクションにアクセスされる各システムは、トランザクションがどこかでコミットする前に、その更新の一部を永続的に格納しなくてはならない。それによって、トランザクションのコミットの前に障害が発生した場合、その回復後にトランザクションをコミットすることが出来る。いかに2相コミットにおいて原子性を保証するかは図2.2に示す。

- a. 2相コミットなし。
障害によってYの更新が失われた



- b. 2相コミットあり。ロンドンシステムはYの更新を耐久的に保存していたため、回復後にコミットすることが出来る。

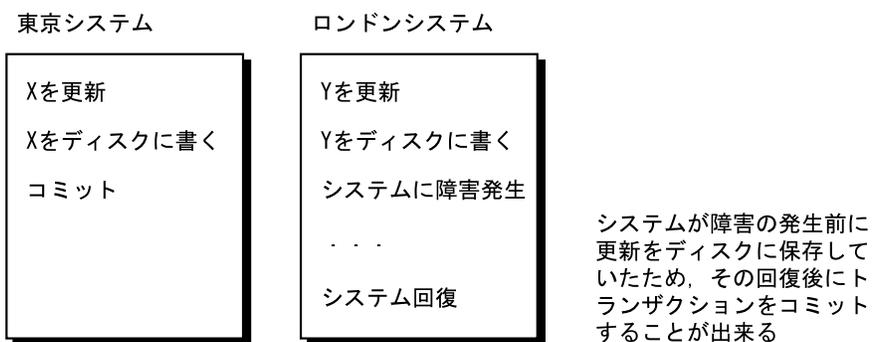


図 2.2: 2 相コミットにおける原子性の保証

2相コミットはトランザクションが2つ以上のリソースマネージャをアクセスする時

は必ず必要とされる。また、2相コミットを使用するということはオーバーヘッドを増加させる代わりに、良いスケーラビリティと可用性を提供できるように分散するための選択肢である。

2.4 トランザクション処理システム

トランザクション処理システムとは、トランザクションプログラムを処理するソフトウェアとハードウェア両方からなるコンピュータシステムのことである。トランザクション処理システムのソフトウェア部分は通常図 2.3 に示される構成になっている。ここで、トランザクションクライアントとトランザクションサーバ間の通信において、一方が発信したデータをもう一方が受信するやり取りをインタラクションと定義する。

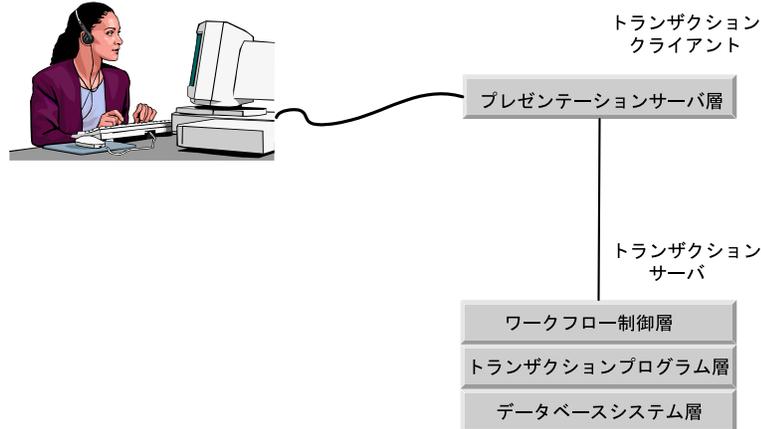


図 2.3: トランザクション処理システムのソフトウェア構成

2.4.1 トランザクションクライアント

クライアントはエンドユーザが直接操作するディスプレイ端末に実装される。またクライアント部分は、通常プレゼンテーションサーバ層のみから構成される。

プレゼンテーションサーバ層

プレゼンテーションサーバ層での役割は、入力を収集し出力を表示するためにユーザとやり取りすることと、トランザクションを処理するフォーマットで要求メッセージを構成する作業を行うことである。また、システムによってはユーザの認証やメッセージログを取得する役割もある。

2.4.2 トランザクションサーバ

サーバはワークフロー制御層，トランザクションプログラム層，データベースシステム層の3層から構成される。

ワークフロー制御層

ワークフロー制御層の主な目的は，プレゼンテーション管理層，もしくはトランザクションプログラム層から取得したメッセージを，要求を実行することが出来るトランザクションプログラムの識別子への写像を行い，適当な層にメッセージを受け渡すことである。また，ワークフロー制御層はトランザクションの前後をくくる「開始」，「コミット」，「アポート」の操作を発行する。

トランザクションプログラム層

トランザクションプログラムとは，要求を実行する本当の仕事を行うアプリケーションプログラムである。トランザクションプログラムは自身で完結するプログラムである場合もありえるし，自身が終了した際別のプログラムを呼び出す可能性もある。

データベースシステム層

データベースシステム層ではデータベースのアクセス制御や更新アーキテクチャ，データベース間でのやり取り等を担当している。データベースシステム層により，トランザクションプログラムからデータベースに関連する複雑な操作が抽象化される。

2.5 トランザクション処理の性能

トランザクション処理システムの重要な性能評価の重要な指標として，トランザクション処理システムが利用可能な時間の割合である可用性と，トランザクション処理性能評議会が提示したベンチマークによる，一定時間以内のトランザクション処理回数とそれをトランザクション処理システムに費やす費用で割ったコスト比が挙げられる。本節では以上の2つを解説する。

2.5.1 可用性による性能評価

可用性とは，トランザクション処理システムが稼動中で有益な仕事をする事が出来る，つまりハードウェアやソフトウェアの障害，またはオペレータのエラー，予防のためのメンテナンス，電源の障害等が生じていないトランザクション処理システムを利用することが可能な時間の割合であり，トランザクション処理システムの性能を評価する重要な指標の一つである。なぜなら，ほとんどのトランザクション処理システムは，通

常航空券予約や銀行での口座の管理、または証券取引における株売買の処理など、サービスを提供する企業の運営の根幹に関わる、失敗が許されないサービスを提供しているからである。従って、トランザクション処理システムはほぼいかなる時も運用されなければならない。トランザクション処理システムがダウンする時間と可用性の関係を表 2.2 に表した。

表 2.2: 可用性とトランザクション処理システムがダウンする時間の関係

トランザクション処理システムがダウンする時間	可用性 (%)
1 時間/日	95.8
1 時間/週	99.41
1 時間/月	99.86
1 時間/年	99.9886
1 時間/20 年	99.99942

それでは、トランザクション処理システムはどの位の可用性を実現すべきなのか。例えば、可用性が 96% であるトランザクション処理システムを考慮した場合、1 日に約 1 時間システムが利用できないことを意味する。それは多くの種類のビジネスにおいて致命的な時間であり、このシステムは受けられない。また、北米の電話システムでは 99.999% の可用性を実現していて、これは 20 年に約 1 時間システムが利用できないことを意味し、このレベルであればトランザクション処理システムのユーザは満足する。一般的にトランザクション処理システムの顧客は、投資額にもよるが 98%~99% の範囲の可用性を期待している。

通常高い可用性を達成するためには以下の 4 つの要素がある。

- 物理的な環境
- システム管理
- ハードウェア
- ソフトウェア

この内本論文で関係があるのは最後のソフトウェアであり、4 つの要素の中でもソフトウェアが最も可用性に影響を及ぼす場合が多い。さらに、ソフトウェアから生じる障害の原因はアプリケーション、データベースシステム、オペレーティングシステムの 3 つの範疇に分類でき、本研究ではアプリケーションから派生する障害のみを考慮する。

2.5.2 TPC ベンチマークによる性能評価

トランザクション処理システムは、通常トランザクション処理性能評議会 (TPC[9]) と呼ばれる協会が定義したベンチマークにより、性能評価が行われる。各 TPC ベンチマー

クは、基準となるトランザクションプログラムを定義し、システムがある負荷状況、あるデータベースの大きさ、保障する応答時間などの条件の下で処理できるスループットによってシステム性能を評価する。システムの性能には2つの主な評価基準がある。まず、それが達成できる最大スループットで、1分間あたりに正常終了するトランザクションの回数 (tpm) で測定されている。そして、tpm 当たりのドル高で測られるトランザクションのコスト比で、コストはハードウェアとソフトウェアの購入価格に、そのハードウェアとソフトウェアの5年間のベンダーによる維持費を加えたもので計算される。

現在、TPC が策定したベンチマークで重要なものとして、TPC-C、TPC-H、TPC-R、TPC-W の4つがあり、その1つである TPC-C について解説する。

TPC-C

TPC によって策定されたコンピュータシステムのはベンチマークの1つである。トランザクション処理システムの性能評価に広く使われていて、結果は1分間に処理できるトランザクション数 tpmC で表される。TPC-C は以下の5種類のトランザクションから構成される。

- **新規受注**

新しい注文を受けるために、まず与えられた倉庫、顧客とチクに関するレコードを検索し、その後、地区を更新する(次に使われる受注番号を1つ数え上げる)。そして、受注と新規受注のテーブルに新しいレコードを挿入する。5から15、平均して10の品物の受注ごとに品物のレコードを検索し、(もし存在しなければアポートする)在庫レコードを検索して更新し、そして受注表のレコードを挿入する。

- **支払い**

支払いを入力するために、まず与えられた倉庫、顧客と地区を記述しているレコードを検索、更新し、そして、履歴レコードを挿入する。もし、顧客がID番号ではなく、名前で特定されていれば、正しい顧客を見つけるために(平均して2つの)顧客レコードを検索しなければならない。

- **受注状態**

ある顧客の最新の注文の状態を決定するために、与えられた顧客レコード(支払いと同じく、もし名前で指定された場合は複数となる)を検索し、顧客の最新の注文と対応する受注表を検索する。

- **発送**

各倉庫の持つ10の地区それぞれに関しての新しい発注を処理するために、角逐の新規発注のうち最も古いものを得て、それを消し、対応する顧客レコード、注文レコード、受注表レコード中の注文を検索して更新する。これは1つのトランザクションとしても、10個のトランザクションとしても行える。

- **在庫レベル**

その倉庫の地区で、在庫が与えられた閾値を下回ったことから最近売れたと判断で

きる品目の数を決定するために、与えられた地区(次の注文番号を持っている)を記述しているレコードを検索する。また、その地区での以前の20の注文に対する受注表を検索し、その注文された品目それぞれに対して、与えられた閾値が在庫合計数より大きいかどうか決定する。

第3章

問題意識と解決手法

本章では、無線携帯型計算機よりトランザクション処理システムを実行する場合、どのような問題や性能低下が生じるかを明確にし、無線特性や無線通信が無線通信特性を考慮したトランザクション処理システムを構築するアプローチ手法について説明する。第1節にてトランザクション処理システムに悪影響を及ぼす無線特性を列挙し、具体的に無線特性がトランザクション処理にどのような性能低下を招くかを解説する。第2節にて、第1節で述べた問題意識を前提に、無線環境に適したトランザクション処理システムを構築するアプローチ手法を提示する。第3節では、本研究と異なるアプローチにより、無線環境に適したトランザクション処理システムを実現している関連研究の概要を記述する。

3.1 問題意識

無線通信は有線通信に比べ、低速、高誤り率、高遅延等が原因で予期せぬ通信の切断頻度が高くなるといった問題が存在し、それらは無線環境下にて実行するトランザクション処理の性能を低下させる。さらに無線携帯型計算機の高移動性や貧弱な演算能力の特性もトランザクション処理システムを構築する上で考慮する必要がある。ここではトランザクション処理システムを構築するに当たって非常に深刻な問題である上記3点について解説する。

3.1.1 無線通信特性

無線通信を行っている際、突然通信が切断されることが多々ある。この無線特性はトランザクション処理の性能を大きく低下させる。この予期せぬ通信切断問題の原因は主に以下の4点である。

1. 不安定な電波強度

無線通信中に、無線端末が無線基地局から距離が離れてしまったり、周囲に電波障害物が存在した場合、電波強度が低下してしまう。そして、電波強度が一定以下の値を示し、さらに一定時間が経過した際、無線通信は切断される。

2. 低い通信速度

有線通信と比較すると、通常無線通信のスループットは劣ってしまう。このことは無線通信切断の直接の原因ではないが、通信速度が遅い分、トランザクション処理のコミットまでに所要する平均的な時間が増加し、その結果、無線通信切断の可能性を高める。

3. 高いパケットロス率

無線通信は有線通信に比べ、最大スループットで劣るだけでなく、パケットのロス率においても大きく劣る。パケットがロスすると当然その再送が必要で、再送が多くなればなる程、通信時間を引き延ばしてしまう。よって、間接的に無線通信切断の可能性を高めることとなる。

4. 高遅延

無線携帯端末は通常、有線環境下を前提とした端末より性能面で大きく劣っている。さらに、低い通信速度や高いパケットロス率の影響も大きく関わってくる。よって無線携帯端末に実装されているアプリケーションを実行している時間が多くかかってしまい、その結果、無線通信切断の可能性を高める。

トランザクション処理中にこの通信切断が生じてしまった場合、一般的なトランザクション処理システムでは原子性を補完しなくてはならないため、通信が回復し次第トランザクションを再度実行し直さなくてはならない。

例えばあるユーザが電車の中から携帯電話を使用して証券取引を行う例を挙げる。ユーザはトランザクション処理を行っていたが、携帯電話の電波強度が一時的に不安定になり通信が切断されてしまった。ユーザは仕方がないので通信が回復するまで待って、トランザクションを再度実行し直した。しばらくして電車がトンネルに突入し、再びトランザクション処理中に通信が切断されてしまった。ユーザは先程と同様に通信の回復後にトランザクションを最初からやり直さなくてはならない。

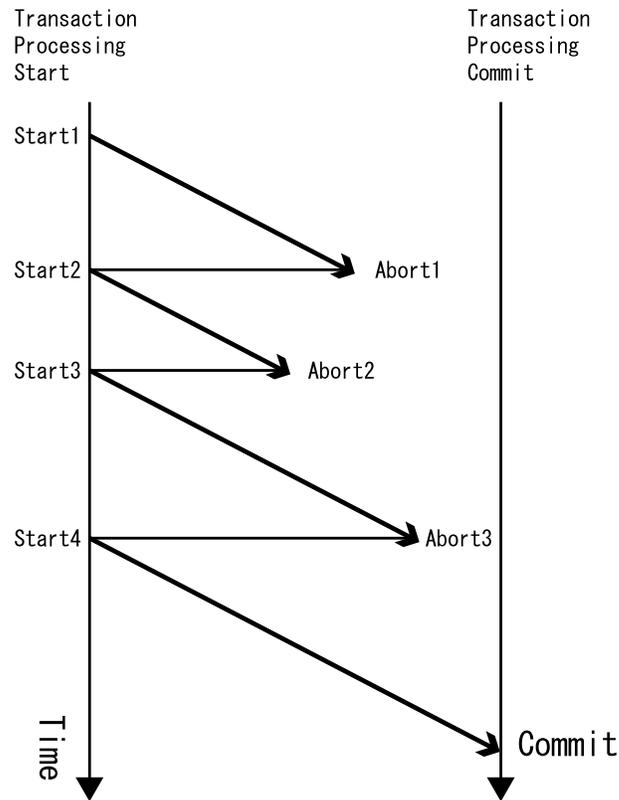


図 3.1: 無線通信の切断とトランザクション処理の関係

以上の例を元に、予期せぬ通信切断とトランザクション処理の関係を表したものが図 3.1 である。左の縦線はトランザクション処理の Start を右の縦線はコミットを表し、矢印が右の縦線に到達しない場合はトランザクション処理がアボートされる。縦軸は時間の進行を表していて、トランザクション処理を実行しているユーザは3回のトランザクション処理のアボートを経た後、コミットしている。この図から予期せぬ通信切断がトランザクション処理に与える悪影響は以下の2点であることが読み取れる。

- **アボートまでに所要した時間は全くの無駄である。**
つまり、図 3.1 では、Start1 から Abort1, Start2 から Abort2, Start3 から Abort3, Start4 からコミットまでに所要した時間の合計が、ユーザがトランザクション処理をコミットするまでに所要した時間であり、アボートが生じないままコミットに到達する場合に比べ、最大で4倍の時間がかかっている。

- アボート毎にユーザはデータを打ち直さなくてはならない。

例えば、証券取引トランザクション処理を実行している最中に何度もアボートが生じた場合、ユーザは無線携帯端末の貧弱な入力デバイスにより、長いID、複雑なパスワードをアボートの度に入力し、今まで調べてきた証券の値動きの分析をし直さなくてはならない。しかも、証券取引においては10分、20分単位の時間が非常にクリティカルになる場合も多い

このように、トランザクション処理中に通信切断が生じる度にユーザはトランザクションを最初から実行し直さなくてはならない。以下の図3.2は横軸にインタラクション毎の通信切断可能性を縦軸にトランザクションが成功するまでに所要する時間を取り、直線・点線はそれぞれ6、10インタラクションからなる通常のトランザクション処理システムを表したグラフである。グラフから無線通信の切断頻度が高くなればなる程、トランザクション処理のコミットまでに所要する平均時間が急激に上昇することが理解できる。

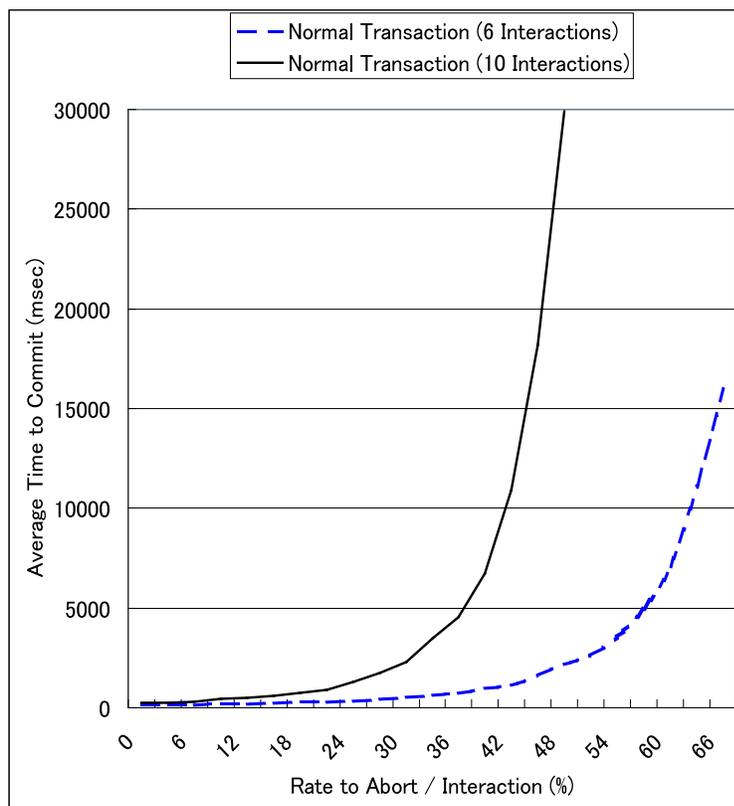


図 3.2: 通信切断確率とフラットトランザクション処理所要時間のグラフ

3.1.2 無線媒体の変移

2つ目は無線携帯端末特性で高い移動性である。必ずしも同一の基地局を介して通信が行われるとは限らなく、むしろ異なるネットワーク間にて通信が移動する可能性もあ

る。そのような場合でもトランザクションサーバはトランザクションクライアントとの通信を維持しなくてはならない。また、トランザクションサーバやデータベースサーバが複数存在する場合はさらに問題が複雑になる。

1. 通信基地局の切り替え

例えば、無線 LAN によりインターネットに通信接続された PDA を移動しながらの操作していて、基地局の電波強度が一定以上弱まったら、無線 LAN 媒体は通常他の接続可能な基地局との通信を行おうと試みる。無線携帯型計算機を操作し、金銭のやり取り等、重要なトランザクション処理を行っている場合、アクセスする無線基地局の変化の際に IP アドレス等が変化することで、セキュリティを考慮し、一般的には中断される。

2. 通信媒体の変移

例えば、無線 LAN を介しての無線通信中、移動等により基地局との距離が離れたために、無線 LAN による通信が不可能となり、PHS 等他の無線通信媒体より通信を再接続する場合である。無線通信媒体を変更することで、当然、IP アドレスを含めた所属ネットワークの変化等が生じる。当然ネットワークの変移により、一方がファイアウォールの設定を敷いている等、ネットワークのセキュリティ環境等も異なる可能性がある。

以上の問題解決手法には 2 種類あり、1 つはトランザクション処理システム側での解決、もう 1 つはネットワーク側での解決であり、関連研究として前者では無線端末の高移動性に伴うマルチデータベース間の移動をテーマとした Kangaroo Transaction[10] が、後者では Mobile IP[11]、Virtual IP[12] 等がある。

3.1.3 無線携帯型計算機特性

最後は無線携帯型計算機自体の特性について解説する。無線携帯型計算機は携帯性を高めるために、可能な限り計算機重量や体積を小さくするのが望ましい。また、携帯するためにはバッテリーをもユーザが所持しなくてはならない等の制約が発生する。

1. 制限ある電気容量

バッテリーを携帯することも無線携帯端末の重要な特性で、バッテリーの電力がなくなると、アプリケーションの実行を含めた全ての機能が使用できなくなる。そのため、アプリケーションは、電力消費を軽減するために、出来る限り処理の平均所要時間を短くするべきである。

2. 貧弱な演算能力

一般に無線携帯端末は小型化・軽量化が重要な性能評価の指標であり、これらを実現するためには演算能力を犠牲にしなくてはならない。CPU パワーやメモリ容量、記憶容量等はデスクトップコンピュータに比べ大幅に劣っているため、同じアプ

リケーションを実行してもその処理速度はどうしても劣ってしまう。また、アプリケーションによっては実行できないものもある。

例えば NTT DoCoMo は携帯電話に JAVA 開発環境を提供したが、これは通常の JAVA 開発環境に比べ機能的に様々な制限がある。JAVA 開発環境自体が携帯電話端末の計算機資源を占有すべきではないし、仮に JDK[13] を実装できたとしても多くのアプリケーションでは利用できない。

3. 小型入力媒体

上記の通り、より高い携帯性を実現するためには、計算機の体積を小さくする必要がある。体積が小さくなるということで、有線端末の一般的な入力媒体であるキーボード等を装備することは至難である。計算機サイズを若干大きくし、キーボードを装着するモデルや、音声やタッチパッド等を入力媒体とするモデル等、様々な形態の無線携帯型計算機があるが、有線端末の入力媒体より劣る。

4. 小型出力媒体

同様に、無線携帯型計算機の体積を小さくするためには、計算機が行った処理の結果を表示する出力デバイスもその制限を受ける。有線端末の出力媒体は、15 インチ以上のモニタが一般的であるが、無線携帯型計算機では、体積的な制限から、それ以下の大きさの出力媒体しか装備することが出来ない。

以上が無線携帯型計算機特性の問題であるが、これらの問題はトランザクション処理システムだけに限らず、すべてのアプリケーションに大しても発生する問題でもある。

3.2 解決アプローチ

本研究では、無線通信特性の不安定な電波強度・低い通信速度・高いパケットロス率・高遅延の問題、無線媒体の移動性の通信基地局の切り替え、通信媒体の変移の問題、そして無線携帯型計算機特性の制限ある電気容量の問題等 [14] を統括し、高通信切断問題と定義し、この問題を解決したトランザクション処理システムの実現を目的とする。

解決アプローチとしては、トランザクションサーバとトランザクションクライアントが無線通信を介してトランザクション処理を行うのではなく、データベースリソースの一部を含んだトランザクションエージェントという、エージェントプログラムにトランザクション処理の大部分を行わせることで、無線通信のインタラクション数を可能な限り少なくすることでトランザクション処理に対して無線通信の抽象化する手法を取る。また、トランザクション処理中に何らかの障害が発生しても、アボートせず、トランザクションエージェントに状態を常に保持させることで、障害からの回復後トランザクション処理を復帰させることでユーザが送信した情報を再入力する手間を省くことを実現する。

ただし、この解決アプローチを採用した場合、データベースリソース資源の確保に関して考慮しなくてはならない場合が存在するので、その可能性と、対処の方法についても言及する。

3.2.1 エージェントベーストランザクション処理

通常のトランザクション処理システムは、トランザクションクライアントとトランザクションサーバという2つのコンポーネントから構成され、無線通信を介してトランザクションクライアントとトランザクションサーバがトランザクション処理を行う。しかし、このモデルだとデータベースシステム等にアクセスする必要がある度に、障害の発生しやすい無線通信を介し、インタラクションを行わなくてはならない。

この問題を改善するために、トランザクションサーバは自身が管理するデータベースリソースの一部を、トランザクション処理を行うエージェントプログラムと一緒に、トランザクションクライアントに転送する。このエージェントプログラムはデータベースリソースを所持しているため、トランザクションクライアントがデータベースにアクセスする必要があるが、無線通信を介してトランザクションサーバにアクセスする必要がなくなる。

このように、インタラクションベースでなく、エージェントベースでトランザクション処理を行うことで無線通信を介したインタラクション数を減少させることになり、無線通信特性や無線媒体の移動性等による通信切断・切替の影響を弱めることが可能となる。

3.2.2 非アポートトランザクション処理

通常、トランザクション処理におけるアポートの原因は、トランザクションクライアント、トランザクションサーバと前記2者の通信間での障害が原因である。これらの障害は必然であるが、もしトランザクション処理がアポートしなければ、ユーザは原因が解明し、トランザクション処理システムが復帰次第、障害が発生する直前の状態から、そのトランザクション処理を続けて実行することができる。よって、前節で述べたアポートがトランザクション処理システムに及ぼす2つの性能低下である、アポートまでに浪費する時間と、ユーザにとって入力を何度も繰り返す必要を無くすることができる。

3.2.3 問題点

非アポートトランザクション処理を採用したトランザクション処理を採用した場合、一つの大きな問題が生じる。トランザクション処理がアポートされないということは、トランザクション処理が進行中であるということの意味し、トランザクション処理中のデータベースのリザーブをどう処置すべきかという点である。例えば、競争率の高いチケットを予約するトランザクション処理において、チケットを選択し、代金を支払う直前でトランザクション処理が中断され、数日後に再開した場合を考慮する。チケットがその間ずっとリザーブしておくのは、常識から考え他の人に対し不公平であるし、チケット販売者にとっても多くの場合、迷惑と感じるであろう。また、頻繁に値段が変わる証券取引等の例では、証券を選択し、代金を入金する直前でトランザクション処理をストップされて、一定時間経過後の値段と見比べた後、選択することも可能である。

これらの問題を解決することは、トランザクション処理をアポートさせないこととト

レードオフの状態にあり，根本的な解決は不可能である．解決方法としては，トランザクションサーバの管理者がトランザクション処理を同時に実行するトランザクションクライアントの最大数を設定できたり，トランザクション処理を実行することのできる最大時間を設定することができるようにする等，トランザクション処理システム側に豊富な設定を選択できるようにする手法をとる．

3.3 関連研究

本節では，第1節で述べた問題意識を本研究とは別のアプローチにより解決している関連研究について解説する．

3.3.1 NTMS [15]

正式名称は，A New Transaction Management Scheme for Mobile Computing Environments である．無線通信はスループットが小さく，通信スループットが不安定，アップリンクとダウンリンクが同じでない可能性が高く，無線端末の計算機資源は貧弱かつ，駆動時間に制限があり，無線通信は頻繁に切断される環境を想定し，総合的な処理実行時間を短縮し，無線の通信切断問題を改善することを目的としたトランザクション処理システム．

解決アプローチとして，サーバ側にあるデータベースリソースを丸ごと無線計算機に取り込み，内部でトランザクション処理を行うことで無線通信のインタラクションを最低限に抑え，通信インタラクションのエラーを極力抑える手法を採用している．また，無線端末に Fully Connected, Fully Disconnected, Partly Connected, Fully Disconnected という4つの状態を設定し，かつそれら3つの状態遷移をスムーズに行うことにも焦点を当てている．

3.3.2 Kangaroo Transaction

Kangaroo Transaction はトランザクション処理に性能低下を及ぼす，無線特性の特に高移動性の問題解決を主眼としている．例えば，あるユーザが乗用車に乗って移動している状態でトランザクション処理を行う場合を想定する．ユーザは最初，無線 LAN を利用し，ネットワーク A の無線基地局にアクセスしていたが，しばらくして，ネットワーク B の無線基地局に再接続した．さらにしばらくして，無線携帯型計算機はどの無線基地局とも接続できなくなり，携帯電話を用いて通信を開始した．ユーザは2種類の異なるブロードバンドの無線通信デバイスを使用し，異なる3つのネットワークセグメントに属した．このような状態でトランザクション処理を実行するのは困難である．

以上の問題の解決するために，Kangaroo Transaction はトランザクションクライアントに移動制御マネージャを設定し，異なるネットワークに属しても，トランザクション処理を続けることができるようになっている．

また、トランザクション処理中に何らかの障害が発生すると、原子性を保証するためにトランザクション処理を最初からやり直さなくてはならないという問題に関しては Split Transaction[16] というトランザクション処理システムのモデルを取り入れている。解決アプローチとしては、一つの塊であるトランザクション処理システムを複数の小さなトランザクション処理の塊に分解する手法を採用している。分解されたトランザクションはその単位で、原子性を保証しなくてはならないが、一つの大きな塊であるトランザクション処理システムと比べると、原子性の問題は大きく改善されている。

3.3.3 2 Phase Recovery Transaction [17], [18]

無線通信の特性である、不安定な電波強度、低い通信速度、高誤り率等により生じる予期せぬ通信切断問題の改善に焦点を当てたトランザクション処理システムである。トランザクションクライアントとトランザクションサーバの間にトランザクションエージェントというコンポーネントを設置し、クライアントやサーバがやり取りする情報等をステートとして保持することで、無線通信が切断されてもトランザクション処理をアポートさせず、通信復帰時にトランザクション処理も復帰させるという問題解決アプローチを採用している。また、無線通信だけでなく、トランザクションクライアント、トランザクションサーバ、トランザクションエージェントのいずれか一つのコンポーネントに対して、障害が発生しても、障害から回復後トランザクション処理を復帰できる。

3.3.4 Compensatable Transaction [19]

Compensatable Transaction 自体は、無線環境下からのトランザクション処理を想定して、無線特性によるトランザクション処理の性能低下の解決を目的としているわけではない。しかし、全てのトランザクションが満たさなくてはならない原子性が原因で、トランザクション処理において障害が発生した場合、トランザクションを最初から再実行しなくてはならないという問題意識を掲げている。

Compensatable Transaction では、前述した問題に対し、トランザクションサーバがステートを保持し、トランザクション処理中に障害が発生しても、復帰できる手法を採用しているのに比べ、本研究ではステートをトランザクションクライアントとトランザクションサーバから独立したコンポーネントにおいて保持させている。

3.3.5 TranSuMA [20]

TranSuMA の正式な名前は、Non-Blocking Transaction Support for Mobile Agent Execution である。インタラクションの大部分をモバイルエージェントに置き換え、トランザクション処理を実行する場合、モバイルエージェントが一度、トランザクションサーバを離れてしまうと、トランザクションサーバには、いつモバイルエージェントが帰還するかが全く予測できなくなる問題が生じる。この問題を改善したトランザクション処

理システムが TranSuMA であり, TranSuMA のトランザクションサーバはモバイルエージェントの移動をある程度把握する事ができる.

第4章

設計

本章では、無線携帯型計算機よりトランザクション処理を行った場合の問題や性能低下の改善を目的とした **Split Database Transaction Processing System** の設計について述べる。第1節にて **Split Database Transaction Processing System** の全体構成について概要を説明する。第2節から第6節までは、**Split Database Transaction Processing System** の5つの主な機構であるトランザクション処理機構、移動制御機構、トランザクション復帰機構、エージェント管理機構、アプリケーション保存機構をそれぞれ詳述する。

4.1 Split Database Transaction Processing System の概要

無線携帯型計算機より実行する通常のトランザクション処理システムでは、図 4.1 のように、トランザクションクライアントとトランザクションサーバが無線を介して直接通信を行う。一方、本システムでは図 4.2 で示されているように、トランザクションサーバがトランザクションエージェントを作成し、トランザクションサーバが管理するデータベースの一部のリソースごと、トランザクションクライアントに転送し、トランザクション処理を行うトランザクション処理モデルを採用している。

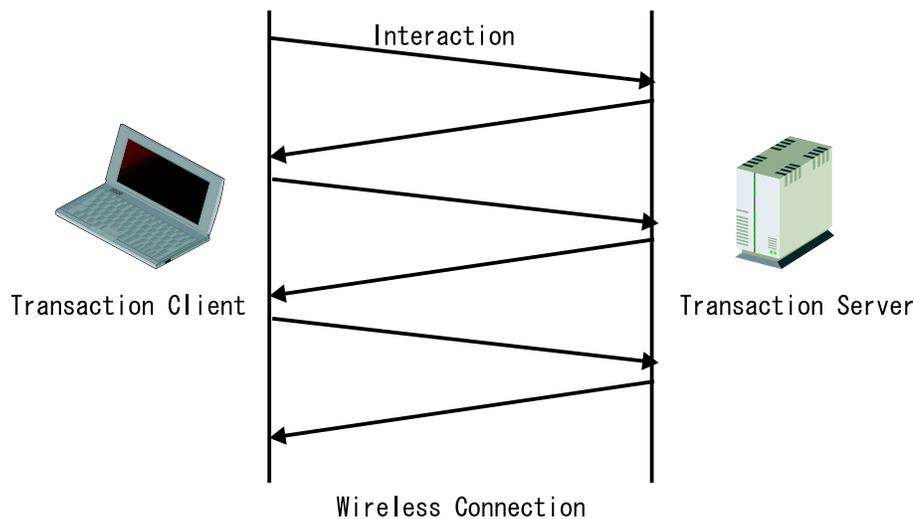


図 4.1: 通常のトランザクション処理モデルの概要

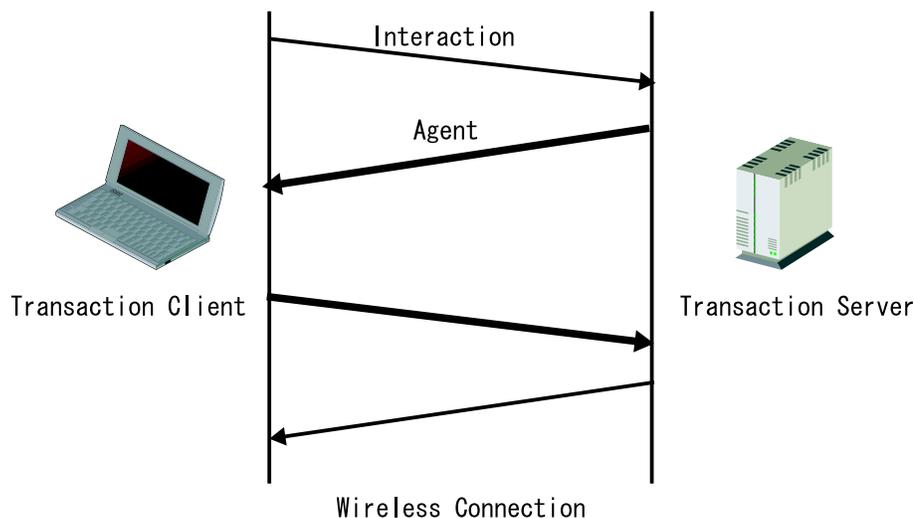


図 4.2: エージェントベーストランザクション処理モデルの概要

想定する環境としては、無線携帯型計算機にトランザクションクライアントが、そし

てトランザクション処理サーバにはトランザクションサーバが存在していて、トランザクションエージェントは無線携帯型計算機とトランザクション処理サーバを行き来する。また、トランザクションクライアントとトランザクションサーバ、トランザクションエージェントの実態はプロセスである。トランザクションクライアントは無線携帯型計算機特性のバッテリーの制限等より障害が発生する。また、無線通信特性や無線媒体の移動性により、無線通信間では、頻繁に障害が発生する。トランザクション処理サーバの前提として、高い可用性が不可欠であるため、トランザクションサーバに対する障害発生の可能性は非常に少ないと仮定する。

4.1.1 コンポーネント

Split Database Transaction Processing System の全体構成を図 4.3 に示す。Split Database Transaction Processing System は、トランザクションクライアント、トランザクションサーバ、トランザクションエージェントの3つのコンポーネントにより構成される。

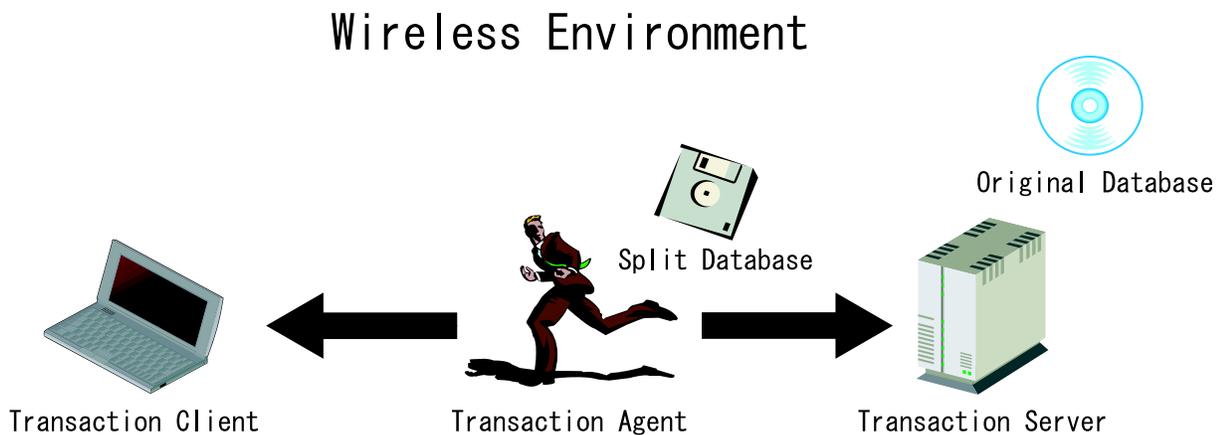


図 4.3: Components of Split Database Transaction Processing System

- トランザクションサーバ

トランザクションサーバは常にトランザクション処理サーバに存在する。また、トランザクション処理サーバは性能や対故障性が非常に高く、ネットワーク帯域も充分であることを前提としているため、トランザクションサーバでの処理は高速かつ障害が発生しにくいと仮定する。トランザクションサーバの主な役割はオリジナルデータベースという元となるデータベースの管理と、トランザクションエージェントの作成、送信、受信等である。トランザクションサーバはトランザクション処理の最初と最後だけトランザクションクライアントと直接インタラクションによる通信を行うが、後はエージェントと処理をやり取りする。

- トランザクションクライアント

トランザクションクライアントは常にユーザが操作する、無線携帯型計算機に常駐

する。主な役割はトランザクションエージェント、もしくはサーバにより要求された情報を入力、送信することである。トランザクションクライアントはトランザクション処理の最初と最後だけトランザクションサーバと直接インタラクションによる通信を行うが、後はトランザクションエージェントとのみ処理をやり取りする。また、トランザクション処理アプリケーションによっては、ステートやデータベース等を2次記憶に保存することで次回からの処理を早めることも可能とする。

- **トランザクションエージェント**

トランザクションエージェントはトランザクションサーバにて作成され、ユーザが操作する無線携帯型計算機にスプリットデータベースというオリジナルデータベースの一部を所持した上で転送され、無線携帯型計算機内にてトランザクションクライアントとプロセス間通信により通信を行う。処理終了後に、トランザクションエージェントはトランザクション処理サーバに転送され、その役目を終える。トランザクションエージェントの主な役割はトランザクション処理の大部分と、その過程において発生する要求する情報や結果等をトランザクションクライアントやトランザクションサーバに知らせることである。

4.1.2 機構

Split Database Transaction Processing System は、トランザクション処理機構、移動制御機構、トランザクション復帰機構、エージェント管理機構、アプリケーション保存機構の5つの機能により構成される。以下の各々の機構の概要を述べる。

- **トランザクション処理機構**

トランザクション処理の最初や最後のやり取り、また障害が発生しない場合のトランザクション処理の基本的な部分を担当する。トランザクション処理を処理中に障害が発生した場合は、トランザクションクライアント、トランザクションサーバ、トランザクションエージェントに各々ステートを保存させて、移動制御機構、もしくはトランザクション復帰機構を呼ぶ。

- **移動制御機構**

無線媒体の物理的な移動等による、ネットワークアドレスの変更からトランザクション処理を継続させることを目的とする。トランザクション処理中にネットワークアドレスが変更されると、移動制御機構が呼ばれ、トランザクションエージェントが保持しているステートとトランザクションサーバが保持しているステートを照らし合わせ、認証を行い、再接続、そして処理がどこまで進んだかをチェックし、トランザクション処理を継続させる。

- **トランザクション復帰機構**

無線通信が切断されたり、各コンポーネントに障害が発生する場合にトランザクション処理を復帰させることを目的とする。無線通信が切断された場合、トランザ

クシオンクライアント、トランザクションサーバ、トランザクションエージェントにて障害が発生した場合の4パターンに対応し、障害からの復帰後、トランザクション処理を復帰させる。

- **エージェント管理機構**

Split Database Transaction Processing System の大きな特徴として、トランザクション処理のアポートに耐性をつけることがある。しかしながら、トランザクション処理が継続されるということは、途中でデータベースリソースが予約され続けることも意味し、トランザクション処理の中断等による、データベースリソースの長時間予約の問題を解決することを目的とする。

- **アプリケーション保存機構**

頻繁に使用するデータベースリソースを予約しないトランザクション処理アプリケーションの場合、最初の通信の度にデータベースの一部やトランザクションエージェントを転送するのは効率的でない。データベースを含んだトランザクションエージェントを予め、ユーザが操作する無線携帯型計算機の2次記憶に保存することで、トランザクション処理所要時間と、ユーザに要求される入力回数を減少させることを実現できる。

4.1.3 データベース

Split Database Transaction Processing System では、トランザクションサーバがトランザクションエージェントを無線携帯型計算機に転送する時に、オリジナルデータベースの一部をスプリットデータベースとしてトランザクションエージェントに添付させる。この2種類のデータベースについて説明する。

- **オリジナルデータベース**

トランザクション処理を実行する元となるデータベースである。通常のOLTPシステムでは、データベースを分離、転送等行わず、オリジナルデータベースとのみトランザクション処理を行う。オリジナルデータベースはトランザクションサーバによってのみ、参照や書き込み等が行われる。

- **スプリットデータベース**

トランザクションエージェントに携帯させるデータベースで、オリジナルデータベースの一部である。データベースの参照や書き込みはトランザクションエージェントによって行われる。また、スプリットデータベース作成の過程において、以下の2つの場合が存在する。

- **オリジナルデータベースに変更を加えない場合**

電子情報を取り扱うトランザクション処理アプリケーションの場合、電子情報自体は幾らでも複製することが出来るため、オリジナルデータベースに変更

を加えずにスプリットデータベースを作成することが可能である。例えば、音楽ダウンロードアプリケーション等における、音楽の値段表等があげられる。この場合は、エージェントやデータベースを保存することで、再びトランザクション処理アプリケーションを実行する際の入力の手間や通信やトランザクション処理の時間を減少させることが出来る。ただ、オリジナルデータベースの更新による、スプリットデータベースとの同一性の保証は不可欠である。

– オリジナルデータベースに変更を加える場合

現実世界に存在する物質等を扱うトランザクション処理アプリケーション等の場合、対象物の個数等には限りがある。この場合、Split Database Transaction Processing System では、オリジナルデータベースのリソースを一部予約した上でスプリットデータベースを作成する手法を取る。例えば、オンラインショッピング等のサービスで在庫に余裕のある商品に対して、オリジナルデータベースの各商品に対し各在庫を1個ずつ減らし、各商品に対し在庫が1個ずつあるスプリットデータベースを作成する。

この場合は、スプリットデータベースが存在する限り、オリジナルデータベースの在庫が少ないままであるため、データベースごとエージェントを保存することは不可能である。そうでなくても、トランザクション処理が長引けば、その間に全て、もしくは複数の商品に対し、対応する在庫にロッキングをかけるのはリソースの無駄である。この問題の解決は、後述するエージェント管理機構にて行う。

Split Database Transaction Processing System では、データベースを2つに分離し、トランザクション処理を行う方法を基盤としているが、データベースの分離を行うということは、データベースの統合における完全な信頼性を保証しなくてはならない。この問題に関しては最終章にて記述する。

4.2 トランザクション処理機構

トランザクション処理機構では、全く障害が発生していない正常時におけるトランザクション処理を主に行う。この場合の全体的なトランザクションの処理の流れで、最もシンプルなものを図4.4にて表す。アプリケーション保存機構等が起動される場合は、図4.4とはやや異なる。

トランザクション処理機構は、トランザクション処理開始時、無障害時、障害発生時、トランザクション処理終了時の4つの場合において実行される。以下に各々の場合について、どのような動作を行うかを説明する。

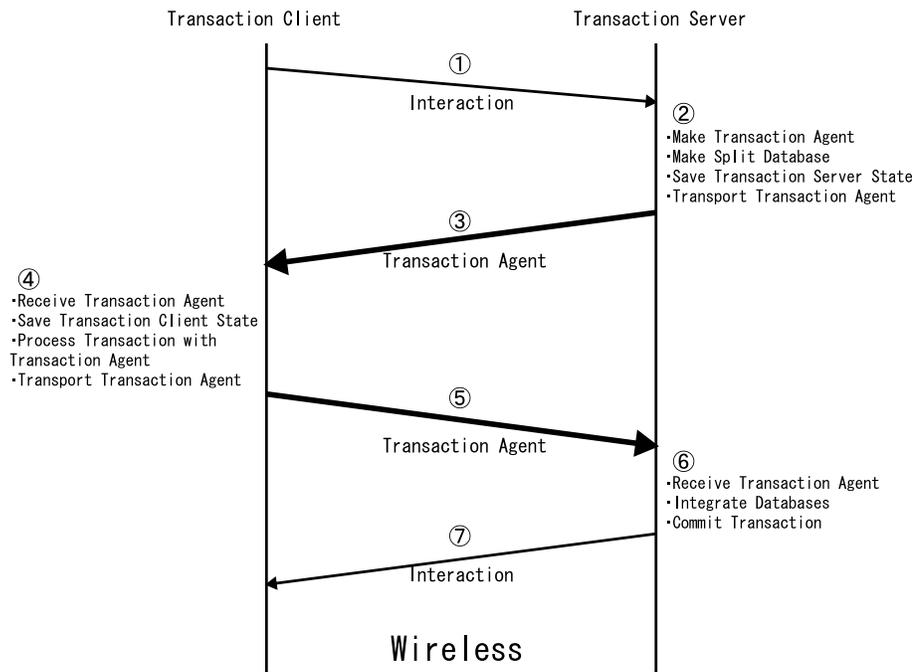


図 4.4: 正常時における Split Database Transaction Processing System の処理の流れ

4.2.1 トランザクション処理開始時の動作

トランザクション処理開始時の動作は、図 4.4 の 1 から 4 の途中までである。Split Database Transaction Processing System では、全てのトランザクション処理においてこの動作が最初に行われる。また、トランザクション処理開始時の動作にて、他の 4 つの機構に必要な情報をトランザクションクライアントは Transaction Client State(表 4.2) として、トランザクションサーバは Transaction Server State(表 4.1) として 2 次記憶に保存する。以下にこの動作についてを説明する。

1. 1 により、トランザクションクライアントはトランザクションサーバに対してトランザクション処理開始要求を行う。
2. 2 において、トランザクションサーバがトランザクションクライアントからの要求を受け取ると、オリジナルデータベースにアクセスし、データベースリソースの一部を一時的に取得し、スプリットデータベースを生成する。
3. オリジナルデータベースのリソースを取得する必要がない場合、オリジナルデータベースの一部をそのまま複製し、スプリットデータベースとして作成する。
4. この時、トランザクションサーバは Transaction Server State(表 4.1) を 2 次記憶に記録する。
5. 3 にて、トランザクションサーバは、トランザクションエージェントにスプリットデータベースを持たせ、トランザクションクライアントに対して送信する。
6. また、トランザクションサーバは、スプリットデータベースの複製をトランザクションサーバ内の 2 次記憶に記録する。

7. 4にて、トランザクションエージェントがユーザ端末に取り込まれたら、Transaction Client State(表 4.2)を作成し、保持する。

表 4.1: Transaction Server State

State Information
IP Address of Transaction Client
Port Number of Transaction Client
Transaction ID
Authentication Keyword
Expiration Date

表 4.2: Transaction Client State

State Information
IP Address of Transaction Server
Port Number of Transaction Server
Transaction ID
Split Database Version
Transaction Agent Version
Authentication Keyword
Expiration Date
Database Type

Transaction Server State(表 4.1)の各ステート情報について解説する。IP Address of Transaction Client, Port Number of Transaction Client はトランザクションクライアントとの通信に関する情報である。これはトランザクションサーバに障害が発生した場合、トランザクションサーバ側から、トランザクションクライアントにアクセスするために必要なステート情報である。Transaction ID とはトランザクション処理サーバが複数のトランザクション処理アプリケーションを実装している場合に区別をつけるためのステート情報である。Authentication Keyword は通信再接続の際や、アプリケーション復帰機構において必要とする情報である。Authentication Keyword は、Transaction Client State(表 4.2)にも全く同じ値が振られ、トランザクションエージェントがトランザクションサーバに再接続する際やアプリケーション復帰機構等、Split Database Transaction Processing Systemにおいて、セキュリティを維持したトランザクション処理を継続する際に非常に重要である。Expiration Date は、トランザクションエージェントの存在有効期間のことで、オリジナルデータベースのリソースが予約され続ける問題を回避することを目的としたエージェント管理機構にて使用される。

Transaction Client State(表 4.2) のステート情報の一部は、 Transaction Server State と同じ情報であるため、これらについては説明を省く。 Split Database Version や Transaction Agent Version はトランザクション処理中に何らかの障害が発生した場合、トランザクションクライアントとトランザクションサーバが通信を行っていない間にオリジナルデータベースやトランザクションエージェントに変更が加えられる問題を回避するために必要な情報である。 Database Type は、スプリットデータベース作成時にオリジナルデータベースに変更が加えられる場合と加えられない場合とでスプリットデータベースの種類を区別する指標である。

4.2.2 無障害時の動作

障害が全く発生しない、通常時のトランザクション処理はトランザクションクライアントとトランザクションエージェントによって行われる。一般的なトランザクション処理システムの処理の流れの一例は図 4.5 の通りで、これを Split Database Transaction Processing System のトランザクションクライアントとトランザクションサーバが行うトランザクション処理の流れに当てはめた場合、図 4.6 のように表せる。

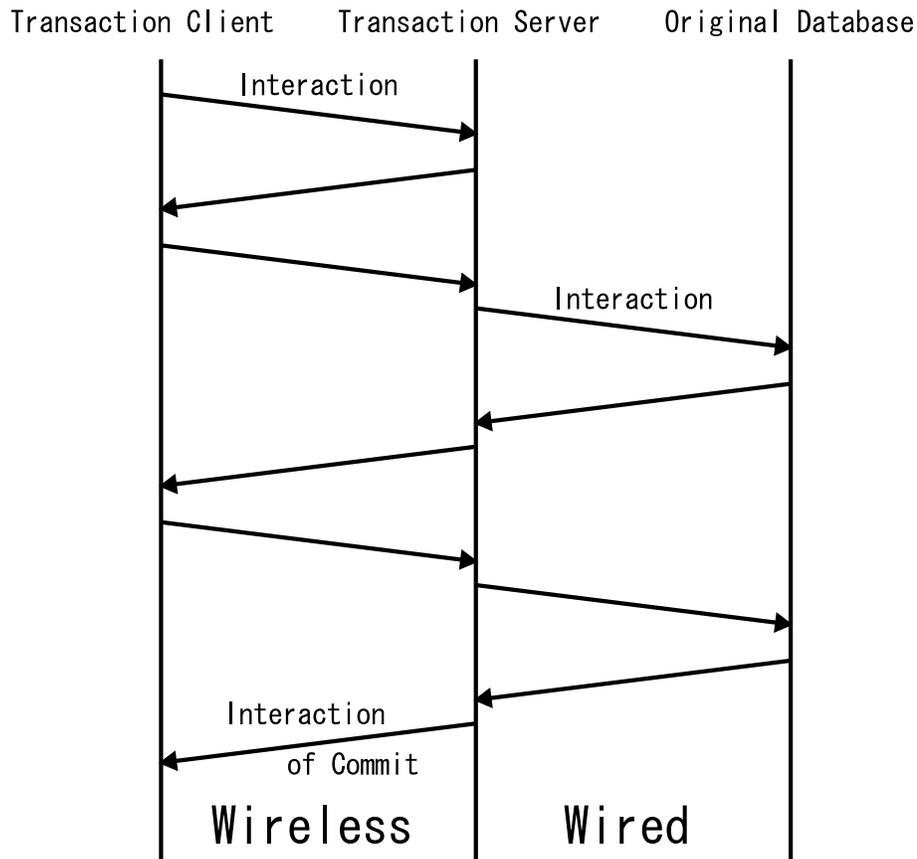


図 4.5: Example of Usual Transaction Processing

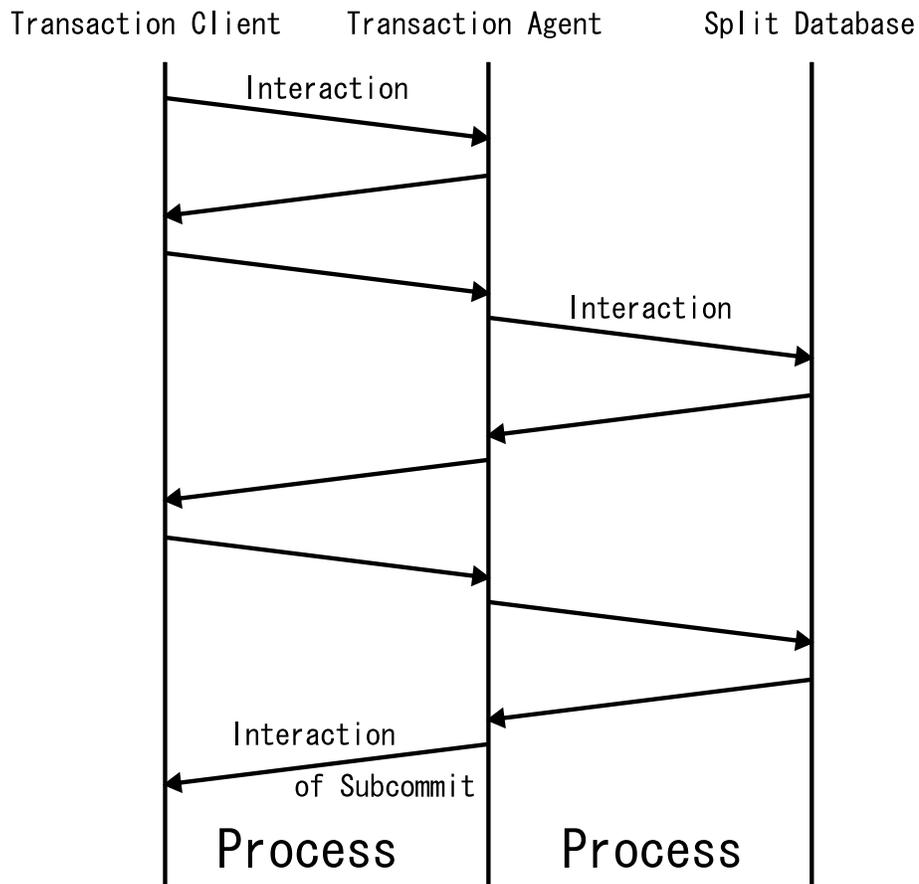


図 4.6: Example of Split Database Transaction Processing Between Transaction Client and Transaction Agent

一般的なトランザクション処理システムの場合、トランザクションクライアントとトランザクションサーバが無線通信を介して、インタラクションをやり取りする。またトランザクションサーバとオリジナルデータベースは有線環境下、もしくはプロセス間通信においてインタラクションをやり取りする。これに対して、**Split Database Transaction Processing System**におけるトランザクションクライアントとトランザクションサーバが行うトランザクション処理では、トランザクションクライアント、トランザクションエージェント、スプリットデータベースが全て同一計算機上に存在するため、インタラクションは全てプロセス間通信にて行われる。よって、**Split Database Transaction Processing System**のトランザクションクライアントとトランザクションエージェントの処理は、一般的なトランザクション処理システムのトランザクションクライアント、トランザクションサーバの通信に比べ、無線特性を抽象化することで、高通信切断問題を改善することが実現できる。

4.2.3 トランザクション処理終了時の動作

トランザクション処理終了時の動作は、図 4.4 の 5 から 7 までである。Split Database Transaction Processing System では、全てのトランザクション処理においてこの動作が最後に行われる。この動作においてアプリケーション保存機構が実行される場合は、若干異なる動作を行う。これについては、アプリケーション保存機構の節にて説明する。トランザクション処理終了時の動作をより詳細に解説すると以下ようになる。

1. トランザクションクライアントとトランザクションエージェントの処理が終了すると、5 のように、トランザクションエージェントをトランザクションサーバに転送する。
2. 6 にて、トランザクションエージェントを受け取ると、トランザクションサーバはスプリットデータベースとオリジナルデータベースの統合を行う。
3. 操作が正常に終了すると、トランザクションクライアントはトランザクションエージェントをスプリットデータベースごと消去する。もし、操作が正常に終了しない場合は Abort する。
4. また、Transaction Server State を消去する。
5. 7 にて、トランザクションサーバはトランザクションクライアントに対して Commit、もしくは Abort の通知を行う。
6. トランザクションクライアントは、Commit もしくは Abort の通知を受け取ったら、保存していた Transaction Client State を消去しトランザクション処理を終了する。

4.2.4 障害発生時の動作

図 4.4 における 3 までで障害が発生した場合は、インタラクション毎にタイムアウトが発生するまで、通信を不定期的に繰り返す。4 以降で障害が発生した場合は、障害の種類によって、移動制御機構かトランザクション復帰機構のどちらかに移動する。

4.3 移動制御機構

ユーザ操作端末の物理的な移動により、トランザクション処理中にネットワークアドレスが変更する可能性が存在する。この問題を改善する機構が移動制御機構である。移動制御機構はトランザクション処理機構において、トランザクション処理開始時の動作の終了後に、ネットワークが変更された場合に呼び出される。トランザクション処理の復帰において、特に問題となるのは、認証においてである。トランザクション処理の復帰の認証は、図 4.7 のように、トランザクションクライアントが保持している Transaction Client State とトランザクションサーバが保持している Transaction Server State と他少々の情報を照合することで行われる。以下に移動制御機構によりトランザクション処理を復帰するまでの手順を説明する。

1. トランザクションクライアントが再びインターネットに接続することが可能となる。

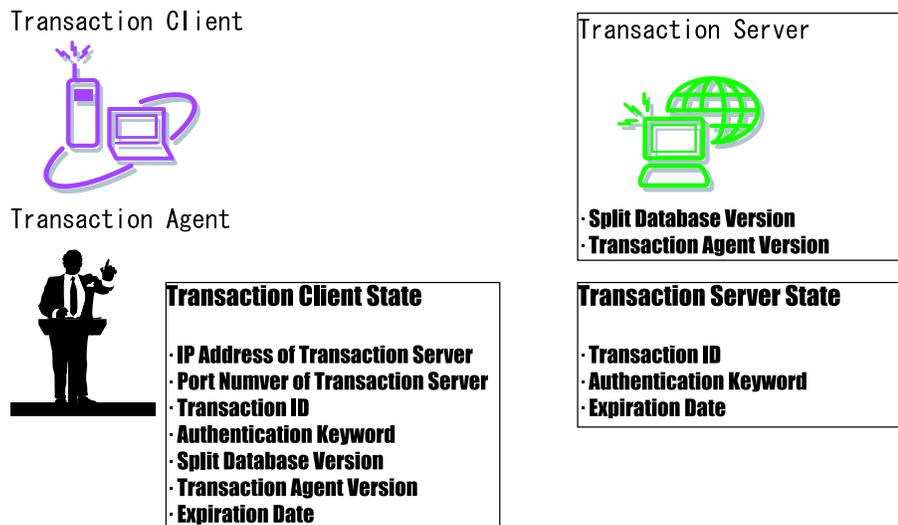


図 4.7: Authentication of Reconnection

2. Transaction Client State の Expiration Date の値により，有効期限を確認する。
3. トランザクションクライアントが保持している Transaction Client State の IP Address of Transaction Server と Port Number of Transaction Server の2つの情報より，トランザクションクライアントはトランザクションサーバに再接続を試みる。
4. トランザクションサーバにアクセスすることが実現したら，Transaction Client State の Transaction ID， Authentication Keyword， Split Database Version， Transaction Agent Version のステート情報をトランザクションサーバに渡す。
5. トランザクションサーバが管理している，オリジナルデータベースやトランザクションエージェントのバージョン，それに Transaction Server State に蓄えられている Transaction ID， Authentication Keyword のステート等の照合を行う。
6. 照合が適切で認証が実現したら，トランザクション処理機構の終了時の動作フェーズに移動し，そこからトランザクション処理を復帰させる。照合が適切でない場合は，Abort する。

4.4 トランザクション復帰機構

無線通信特性や無線媒体の移動性の影響で無線通信間において障害が発生したり，トランザクションエージェント，トランザクションクライアント，トランザクションサーバにおいて，障害が発生した場合に，可能な限りトランザクション処理を復帰させるのがトランザクション復帰機構の目的である。以下にそれぞれの場合について，トランザクション復帰機構はどのような動作を行うかを記述する。

4.4.1 無線通信間に障害が発生した場合

無線通信において障害が発生した場合とは、つまり、無線通信特性の高通信切断問題や、無線媒体の移動性によるネットワークアドレスの変更等であるが、この場合はトランザクションエージェントがトランザクションサーバに対して不定期的に再接続要求を行い続ける。無線通信が回復し、トランザクションサーバとコネクションを張ることが再び可能となったら、トランザクションクライアントが保持する Transaction Client State の一部のステート情報と、トランザクションサーバが保持する Transaction Server State の一部の情報に加え、オリジナルデータベースやトランザクションエージェントのバージョン情報等を照合し、全てにおいて照合が成功した場合、トランザクションを復帰する。設定された時間が過ぎても、無線通信が回復しない場合や照合に失敗した場合に、トランザクション処理を Abort する。

4.4.2 トランザクションクライアントにおける障害が発生した場合

トランザクションクライアントにおいて障害が発生した場合は、トランザクションエージェントに予め設定されたタイムアウトの値を超えるまで、トランザクションクライアントに対しアクセスを試みる。トランザクションクライアントが復帰したら、トランザクション処理がどこまで進んだかをトランザクションエージェントがトランザクションクライアントを提示し、ユーザの合意を得た上でトランザクション処理を復帰させる。もし、設定された時間を過ぎてもトランザクションクライアントと接続することが出来ない場合、またユーザの合意が得られない場合、トランザクションエージェントを消去しトランザクション処理を Abort する。

4.4.3 トランザクションサーバにおける障害が発生した場合

前述した通り、トランザクションサーバにおいて障害が発生する可能性は低いと仮定している。もし仮に障害が発生した場合は、トランザクションエージェントはトランザクションサーバが復帰するまで、不定期的にアクセスを試みる。トランザクションサーバが復帰したら、移動制御機構の時と同様に、トランザクションクライアントが保持する Transaction Client State と、トランザクションサーバが保持する Transaction Server State 等のステート情報の照合することで認証を行い、トランザクション処理を復帰させる。もし、一定時間トランザクションサーバとアクセスすることが出来ない、もしくはステート情報情報の照合による認証が失敗した場合は、トランザクションエージェントを消去し、Abort する。

4.4.4 トランザクションエージェントにおいて障害が発生した場合

トランザクションエージェントにおいて障害が発生した場合は、トランザクション復帰をすることが出来なくなる。一度トランザクション処理を Abort した上で、トランザ

クシヨソ処理を最初からやり直さなくてはならない。

4.5 エーエージェント管理機構

全ての中断されたトランザクシヨソ処理に対して、回復が可能である場合、トランザクシヨソ処理が中断している間もデータベースリソースが予約され続けることとなり、トランザクシヨソ処理を実行する他のユーザや、トランザクシヨソ処理システムの管理者に対して被害が及ぶ可能性がある。この問題点を解決するため、Split Database Transaction Processing System では、トランザクシヨソエージェントに有効期限を持たせ、データベースリソースの予約の時間に制限をすると言う方法を選択する。この有効期限とは、Transaction Client State(表 4.2) や Transaction Server State(表 4.1) の Expiration Date の値である。

基本的なエージェント管理機構の役割は、有効期限を過ぎたトランザクシヨソエージェントをスプリットデータベースごと消去し、また予約されたオリジナルデータベースのリソースを解放することである。このトランザクシヨソエージェントの有効期限はトランザクシヨソ処理開始時に、トランザクシヨソ処理システムの管理者によって決定された値に設定される。エージェント管理機構を、トランザクシヨソサーバ側とトランザクシヨソクライアント側の2つの視点から解説する。

4.5.1 トランザクシヨソサーバ側での動作

トランザクシヨソ処理アプリケーションごとにトランザクシヨソ処理サーバ管理者は、各々トランザクシヨソエージェントが存在することのできる有効期限を設定する。この有効期限の値は、トランザクシヨソエージェントが生成される毎にトランザクシヨソサーバとトランザクシヨソエージェントにステートとして振られる。

有効期限が過ぎた場合、トランザクシヨソサーバは、トランザクシヨソクライアントにトランザクシヨソエージェントの無効の通知を直接試み、また、無効化されたスプリットデータベースを作成する前の状態にオリジナルデータベースを戻し、予約されたデータベースリソースを解放する。

4.5.2 トランザクシヨソクライアント側での動作

トランザクシヨソエージェントがトランザクシヨソクライアントに接触したら、最初に有効期限の値を Transaction Client State として、トランザクシヨソクライアントに渡す。トランザクシヨソクライアントは有効期限が過ぎるか、トランザクシヨソサーバ側からトランザクシヨソエージェントの無効通知が来た場合、スプリットデータベースごとトランザクシヨソエージェントを破棄する。

4.6 アプリケーション保存機構

スプリットデータベースがオリジナルデータベースに対し、変更を加えない場合、スプリットデータベースを含んだトランザクションエージェントごと無線携帯型計算機の2次記憶に保存することで、次回から Split Database Transaction Processing System におけるトランザクションエージェント作成と転送処理を省いてトランザクション処理を行うことができる。これにより、トランザクション処理に所要する時間を短縮することが可能となる。

アプリケーション保存機構は、アプリケーション保存する時とトランザクション処理短縮時の2つの場合において動作を行う。また、トランザクション処理短縮時のトランザクション処理の流れは、図 4.4 で表した通常時のトランザクション処理の流れと多少異なる。

4.6.1 アプリケーション保存までの動作

トランザクションエージェントを保存し、トランザクション処理アプリケーションを次回から短縮するために、トランザクションクライアントとトランザクションサーバは各々の計算機の2次記憶に、それぞれ Saving Application Client State (表 4.3)、Saving Application Server State (表 4.4) として保存する。また、これらの動作について図 4.8 を参照しながら説明する。

表 4.3: Saving Application Client State

State Information
IP Address of Transaction Server
Port Number of Transaction Server
Transaction ID
Split Database Version
Transaction Agent Version
Authentication Keyword
Expiration Date

表 4.4: Saving Application Server State

State Information
Transaction ID
Authentication Keyword

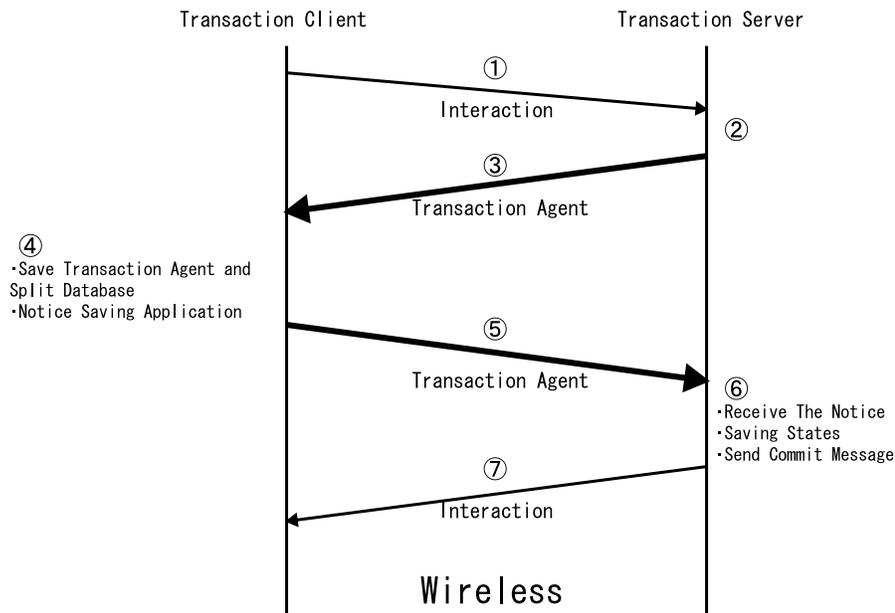


図 4.8: アプリケーション保存までの動作

1. 1～3 までの流れは正常時のトランザクション処理の流れと同じである。
2. 4 において、トランザクションクライアントとトランザクションエージェントにおけるトランザクション処理が終了する。
3. スプリットデータベースを含んだトランザクションエージェントの保存を選択する。
4. 同時に、トランザクションクライアントは Saving Application Client State(表 4.3) を 2 次記憶に保存する。
5. 5 にて、トランザクションエージェントはトランザクションサーバに「トランザクションクライアントがトランザクションエージェントの保存を選択したこと」を通達する。
6. 6 にて、 Saving Application Server State(表 4.4) として、2 次記憶に保存する。
7. 7 にて、アプリケーション保存機構に関する Commit メッセージを送信する。

4.6.2 トランザクション処理短縮時の動作

ユーザが操作する無線携帯型計算機の 2 次記憶に保存したトランザクションエージェントプログラムと Saving Application Client State より、2 回目以降のトランザクション処理アプリケーションに必要な、トランザクションエージェントの作成と受信等の手順を省くことが実現する。実際の動作について、図 4.9 を使用して説明する。

1. トランザクション処理前に Saving Application Client State の Expiration Date より、トランザクションエージェントの有効期限が切れていないかを確認する。

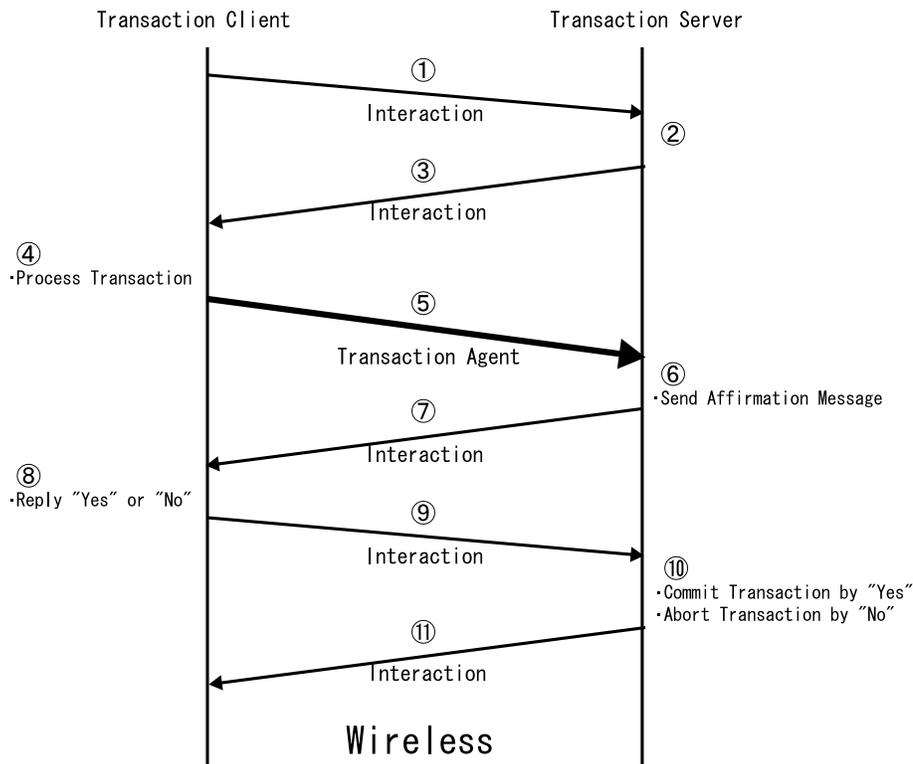


図 4.9: トランザクション処理短縮時の動作

2. Saving Application Client State の IP Address of Transaction Server と Port Number of Transaction Server より、トランザクションサーバにアクセスする。
3. 1 にて、Transaction ID, Split Database Version, Transaction Agent Version, Authentication Keyword を送信する。
4. 2 にて、上記ステートにより、トランザクションサーバはトランザクションエージェントが正しいかをチェックし、またバージョンも同じであることを確認する。
5. 3 にて、トランザクションサーバはトランザクション処理短縮可能である通知をトランザクションクライアントに行う。
6. 4 にて、ユーザ端末に保存されていたトランザクションエージェントと、トランザクションクライアントがトランザクション処理を行う。
7. 5 にて、トランザクション処理結果をトランザクションサーバに送信する。
8. 6 にて、トランザクションサーバ内のトランザクション処理を行う。処理が正しく行われない場合は Abort する。
9. 6 が正しく終了すると、7 にて、取引の概要と確認メッセージをトランザクションクライアントに送信する。
10. 9 にて、ユーザは確認メッセージが正しいかを "Yes" か "No" で答える。

11. 10 にて, "Yes"を受け取ったらトランザクション処理を Commit, "No"を受け取ったらトランザクション処理を Abort し, それをトランザクションクライアントに伝える.

第5章

評価

本章では、**Split Database Transaction Processing System** の定量的評価と定性的評価を行う。定量的評価では、**Split Database Transaction Processing System** の他に、無線通信特性を考慮しない **Flat Transaction** と、高無線通信切断問題に耐性のある **Split Transaction Processing System** を改良した、**Kangaroo Transaction Processing System** の3つのトランザクション処理システムを対象とする。定量的評価は、データベースサイズや各々のトランザクション処理に所要する時間等を多面的に設定することができるシミュレーションプログラムを使用して、測定評価を行う。定性的評価では、上記 **Flat Transaction**, **A New Transaction Management Scheme for Mobile Computing Environments**, **Kangaroo Transaction**, **2 Phase Recovery Transaction**, **Split Database Transaction** の5つのトランザクション処理システムを、9つの観点から分析する。

5.1 定量的評価

第2章にて述べたように、通常トランザクション処理システムはTPCが提唱するベンチマークテストによって行われる。しかしながら、そのベンチマークテストは商用のソフトウェアとハードウェアの両方を組み合わせた、総合的なトランザクション処理システムに用いられるため、トランザクション処理システムのソフトウェア部分に焦点を当てている本研究においては、TPCが提示するベンチマークテストによる性能評価は適切ではない。現実世界において、無線通信における通信切断可能性等、無線特性により生じる様々な現象を数値化し、再現する事が困難であるため、定量的評価をシミュレーションにより行う。

5.1.1 評価目的

図3.2にて示される通り、無線通信の切断率が高ければ高い程、Abort率が飛躍的に増大し、トランザクション処理回数が増えるため、無線特性を考慮していないFlat Transaction Processing Systemの性能は大幅に低下する。また2 Phase Recovery Transaction Processing System[18]の論文において、Flat Transaction Processing Systemのインタラクション数が増えれば増えるほど、同様に性能が大幅に低下するという問題がある。

Kangaroo Transaction Processing Systemでは、上記問題に対して、Split Transaction Processing Systemの全体的なトランザクション処理の中に細かい副トランザクションを設定することで、トランザクション処理中に障害が発生し、Abortしても、副トランザクション単位でトランザクション処理をやり直す手法により、無線通信の高切断問題を改善している。

Split Database Transaction Processing Systemの主な目的は、トランザクション処理がAbortする毎にユーザは必要な情報を入力しなければいけないという問題の改善と、無線通信の切断率が高まっても、トランザクション処理システムの性能をFlat Transaction Processing Systemに比べ性能低下を緩和することである。定量的評価の目的は、Split Database Transaction Processing Systemの後者の目的が達成されていることを証明することと、Split Database Transaction Processing Systemとは異なる手法で無線通信特性によるトランザクション処理システムの問題改善を実現しているKangaroo Transaction Processing Systemとの性能比較評価を行うことである。

5.1.2 測定方法

各種データベースサイズや、インタラクションの送信や、トランザクション処理開始、Commit処理等といった、各トランザクション処理に所要する時間、それにトランザクションクライアントやトランザクションサーバ、また無線通信などでの障害発生確率等を自由に設定することができる評価用シミュレーションプログラムを作成した。設定できる値は表5.1, 5.2, 5.3である。表5.1は、Flat Transaction Processing SystemとKangaroo Transaction Processing Systemを対象としたシミュレーションに必要なパラメータ群で

ある。表 5.2 と表 5.3 は、Split Database Transaction Processing System の性能を測定するために、必要なパラメータである。本シミュレーションプログラムに対し、以上の値を色々設定することで、多くのトランザクション処理アプリケーションを実行した際、Commit までに所要する平均時間を予測することができる。

表 5.1: Parameters of Normal Flat Transaction Processing System

Identifier	Parameter	Default Value
F01	Number of Transaction Phases	4 Phase
F02	Database Size	1024 Kbyte
F03	Start Transaction Delay	100 msec
F04	Restart Transaction Delay	100 msec
F05	Commit Transaction Delay	100 msec
F06	Abort Transaction Delay	100 msec
F07	User Input Delay	500 msec
F08	Read Database Delay	250 msec
F09	Write Database Delay	500 msec
F10	Interaction TC to TS Delay	500 msec
F11	Interaction TS to TC Delay	500 msec
F12	Disconnect TC to TS Rate	4 %/sec
F13	Disconnect TS to TC Rate	4 %/sec
F14	Error TC Rate	0 %/sec
F15	Error TS Rate	0 %/sec

測定対象トランザクション処理システム

次に、シミュレーションによる測定を行う対象であるトランザクション処理システムである、無線通携帯型計算機の特徴を全く考慮していない Flat Transaction Processing System と Kangaroo Transaction Processing System, Split Database Transaction Processing System の簡単な概要を説明する。

- **Flat Transaction Processing System**

Flat Transaction Processing System の処理の基本的な流れは、図 4.1 のようになる。全てのトランザクション処理フェーズにおいて、何らかの障害が発生したらトランザクション処理は Abort し、トランザクション処理を最初から実行し直す。

- **Kangaroo Transaction Processing System**

Kangaroo Transaction Processing System の基本的な流れは、Flat Transaction Processing System とほぼ同様である。しかしながら、Kangaroo Transaction Processing

表 5.2: Parameters 1 of Split Database Transaction Processing System

Identifier	Parameter	Default Value
S01	Number of Transaction Phases	4 Phase
S02	Original Database Size	1024 Kbyte
S03	Split Database Size	256 Kbyte
S04	Start Transaction Delay	100 msec
S05	Restart Transaction Delay	100 msec
S06	Commit Transaction Delay	100 msec
S07	Abort Transaction Delay	100 msec
S08	User Input Delay	500 msec
S09	Read Database Delay	250 msec
S10	Write Database Delay	500 msec
S11	Interaction TC to TS Delay	500 msec
S12	Interaction TS to TC Delay	500 msec

System は、全体的なトランザクションの中に、複数の副トランザクションという細かいトランザクションを作成する。これにより、トランザクション処理中に何らかの障害が発生しても、障害からの復帰後、現在処理を行っている副トランザクションを Abort し、Abort した副トランザクションを最初から実行し直す。

● Split Database Transaction Processing System

Split Database Transaction Processing System の処理の基本的な流れは、図 4.2 のようになる。最初のインタラクションと、トランザクションサーバがトランザクションエージェントをトランザクションクライアントに送信する前までに何らかの障害が発生したら、トランザクション処理を Abort し、トランザクション処理をやり直す。

以後の処理においては、トランザクション処理中に障害が発生しても、障害からの回復し、トランザクションを元の状態まで戻すために、移動制御機構やトランザクション復帰機構等の特殊な動作によって、Abort せずに、トランザクション処理を復帰させる。

測定の種類

定量的評価では、Transaction Processing Application, Split Database Size, Wireless Transmission Speed, Disconnection Rate in Wireless Environment の4つの値が変化が各トランザクション処理システムの性能にどう関わってくるかを検証する。各値は、シミュレーションを10000回実行し、平均値をとる。以下に、これら4つの測定について解説する。

表 5.3: Parameters 2 of Split Database Transaction Processing System

Identifier	Parameter	Default Value
S13	Make Split Database Delay	2000 msec
S14	Copy Split Database Delay	1000 msec
S15	Remove Split Database Delay	1000 msec
S16	Integrate Database Delay	2000 msec
S17	Save Transaction Client State Delay	100 msec
S18	Save Transaction Server State Delay	100 msec
S19	Transport TA from TC to TS Delay	2000 msec
S20	Transport TA from TS to TC Delay	2000 msec
S21	Reconnect Delay	100 msec
S22	Authenticate Delay	250 msec
S23	Disconnect TC to TS Rate	4 %
S24	Disconnect TS to TC Rate	4 %
S25	Error TC Rate	0 %
S26	Error TS Rate	0 %
S27	Error TA Rate	0 %

1. Relation of Transaction Processing Application

2 Phase Recovery Transaction Processing System の論文 [18] において、無線通信特性を考慮していないトランザクション処理システムでは、通信切断頻度がある程度以上高いと、トランザクション処理の全体的な長さが長くなる程、飛躍的にトランザクション処理がコミットするまでに所要する平均時間が短くなると述べられている。当測定において、Flat Transaction Processing System の性能低下の割合と、Kangaroo Transaction Processing System, Split Database Transaction Processing System の性能低下に対する耐性についてを検証する。当測定で使用するパラメータを以下に列挙する。

- F01: Number of Transaction Phase
- S01: Number of Transaction Phase

2. Relation of Split Database Size

Split Database Transaction Processing System は、データベースの一部をトランザクションクライアントに転送するため、分離したデータベースが大きければ、その分だけトランザクション処理の性能が低下する。当測定において、スプリットデータベースのサイズと Split Database Transaction Processing System の性能低下の関係を調べる。当測定で使用するパラメータを以下に列挙する。

- F02: Database Size
- S02: Original Database Size
- S03: Split Database Size

3. Relation of Wireless Transmission Speed

無線通信速度が変化すると、全トランザクションにおけるインタラクションにかかる時間が長くなる。また、Split Database Transaction Processing System においては、トランザクションエージェントの移動時間が増大する。当測定では、無線通信速度の低下によって、各トランザクション処理システムの処理性能がどれだけ低下するかを比較する。当測定で使用するパラメータを以下に列挙する。

- F10: Interaction TC to TS Delay
- F11: Interaction TS to TC Delay
- S11: Interaction TC to TS Delay
- S12: Interaction TS to TC Delay
- S19: Transport TA from TC to TS Delay
- S20: Transport TA from TS to TC Delay

4. Relation of Disconnection Rate in Wireless Environment

無線通信特性が原因で、トランザクション処理の性能が低下する、最も重要な問題は無線通信の高切断確率である。シミュレーションにおいて、通信の切断を擬似的に行い、どのトランザクション処理モデルがこの問題に対して耐性があるかを検証する。当測定で使用するパラメータを以下に列挙する。

- F12: Disconnect TC to TS Rate
- F13: Disconnect TS to TC Rate
- S23: Disconnect TC to TS Rate
- S24: Disconnect TS to TC Rate

5.1.3 測定結果

前節にて説明した4つの測定、Relation of Transaction Processing Application(図 5.1), Relation of Split Database Size(図 5.2), Relation of Wireless Transmission Speed(図 5.3), Relation of Disconnection Rate in Wireless Environment 図 (5.4) の結果を以下に示す。

Relation of Transaction Processing Application(図 5.1)

X 軸の Number of Transaction Phase とは、データベースの参照から、書き込みを行うまでとする。Number of Transaction Phase の値が大きい程、トランザクション処理アプリ

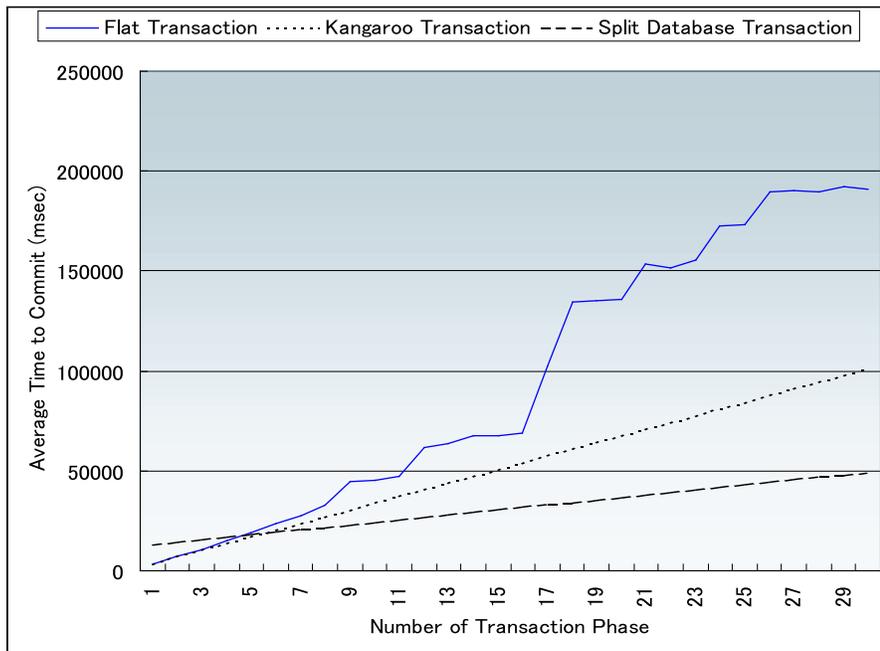


図 5.1: Relation of Transaction Processing Application

ケーションは大きくなる。Number of Transaction Phase の値が 4 までは、Split Database Transaction Processing System より、Flat Transaction Processing System, Kangaroo Transaction Processing System が優れた値を示す。Number of Transaction Phase の値が 5 を超えると、Split Database Transaction Processing System が最もトランザクション処理開始から Commit までに所要する時間の平均値が最も少なくなる。また、Split Database Transaction Processing System は Number of Transaction Phase の値の増加に対し、性能低下の割合が最も緩やかである。

Relation of Split Database Size(図 5.2)

X 軸の Split Database Size とは、Split Database Transaction Processing System において、トランザクション処理サーバがユーザ操作端末に対して送信するデータベースのサイズであり、他のトランザクション処理システムにおいては、無関係である。Split Database Transaction Processing System の長所の一つとして、データベースリソースをトランザクションクライアント端末に蓄えることで、無線通信環境を抽象化する点があるが、その代償として、スプリットデータベースのサイズが大きいく程、Split Database Transaction Processing System に対してより大きい性能低下を招いてしまう。図 5.2 によると、スプリットデータベースのサイズの上昇に比例して、Average Time to Commit の値が上昇している。

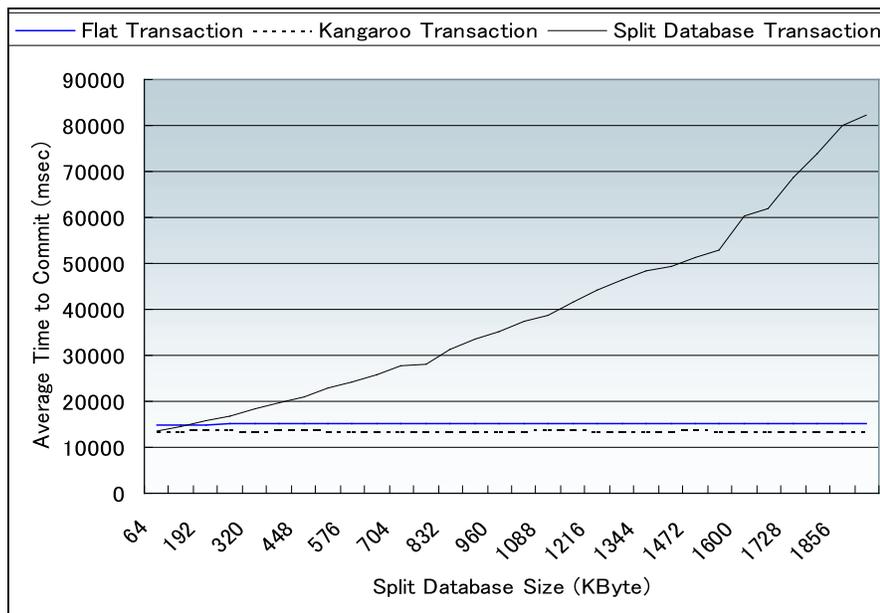


図 5.2: Relation of Split Database Size

Relation of Wireless Transmission Speed(図 5.3)

前節でも述べたように、関係する値は Interaction Delay と Transport Transaction Agent Delay である。X 軸には Interaction Delay の値しか書かれていないが、Transport Transaction Agent Delay の値は Interaction Delay の値の丁度 4 倍に値する。デフォルト値の 2 倍までは Flat Transaction Processing System と Kangaroo Transaction Processing System が Split Database Transaction Processing System より優れた値を示すが、Flat Transaction Processing System と Kangaroo Transaction Processing System は、Split Database Transaction Processing System より、急角度で性能が低下していくため、Interaction Delay の値が 1000 を超えると、Split Database Transaction Processing System が、常に Average Time to Commit の値が最少を示す。

Relation of Disconnection Rate in Wireless Environment(図 5.4)

Kangaroo Transaction Processing System が、常に最も優れた値を示す。性能の低下率自体は、Split Database Transaction Processing System は Kangaroo Transaction Processing System とほとんど変わらない。無障害時におけるトランザクション処理開始から Commit までに所要する時間が、Kangaroo Transaction Processing System の方が少ない事そのまま差となって表れている。Flat Transaction Processing System は、Disconnection Rate in Wireless Environment の値が 4 までは Split Database Transaction Processing System より、トランザクション処理の Commit までの平均所要時間が少ないが、Disconnection Rate の上昇と共に、Split Database Transaction Processing System 以上に Average Time to Commit

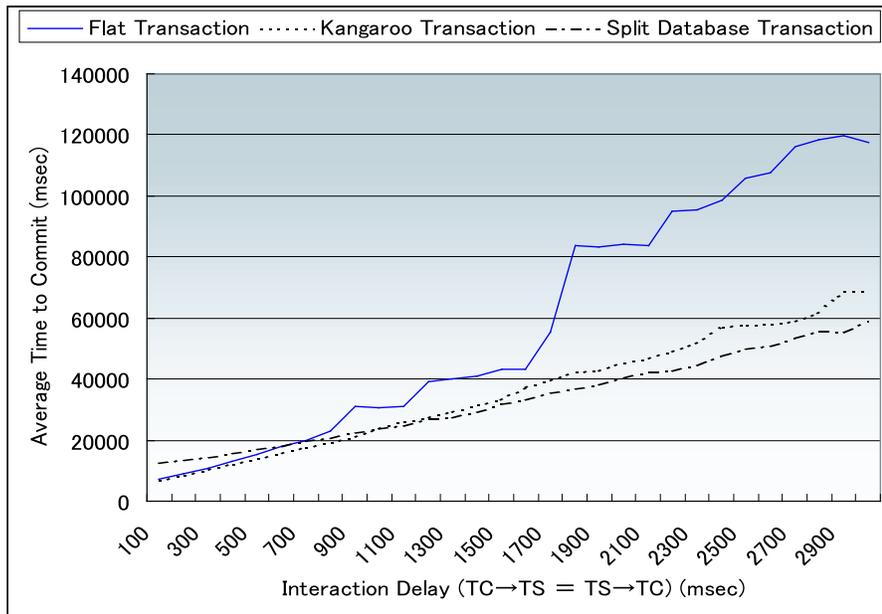


図 5.3: Relation of Wireless Transmission Speed

の値が増加するため、Disconnection Rate の値が5を超えると、Split Database Transaction Processing System は Flat Transaction Processing System に対し、全ての場合において、優れた性能を示す。

5.1.4 考察

前節の測定評価より、Kangaroo Transaction Processing System も、Split Database Transaction Processing System も、無線通信特性や無線携帯型計算機の実機によるトランザクション処理の性能低下に対して、Flat Transaction Processing System より耐性があることがいえる。ただし、Flat Transaction Processing System は無障害時においてトランザクション処理を行った場合は他2トランザクション処理システム以上の性能を発揮する。Split Database Transaction Processing System は、障害がほとんど発生しない場合は、他2種類のトランザクション処理システムより低い性能を示すが、測定の3パターンにおいて、最も低い性能低下率を示した。

よって、Split Database Transaction Processing System は、より障害が発生しやすい環境、無線通信速度が遅い環境、無線通信が頻繁に切断される環境において、優れた性能を発揮する。ただし、スプリットデータベースサイズが大きくなる程、トランザクション処理のコミットまでの平均時間が増加してしまうため、トランザクション処理アプリケーションが管理するデータベースの種類や大きさによっては、不適當である。

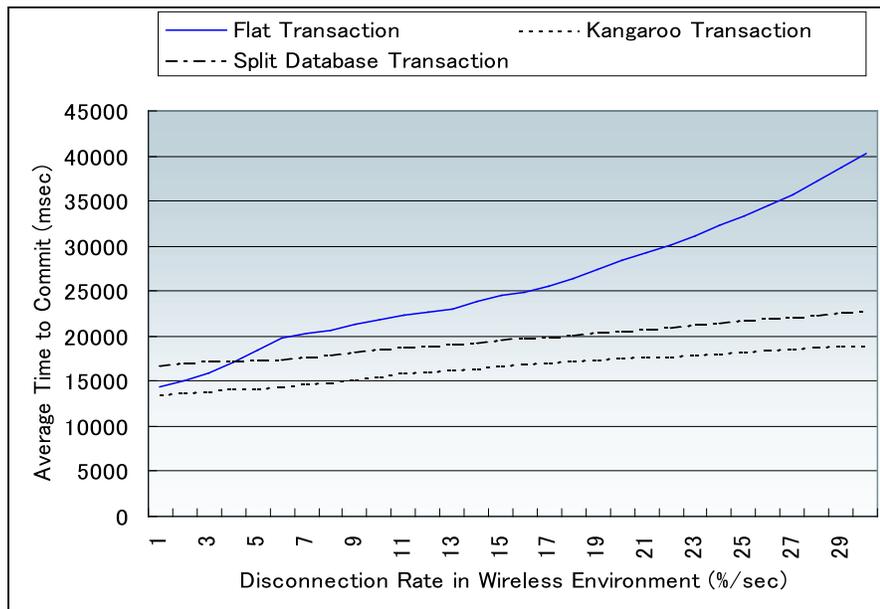


図 5.4: Relation of Disconnection Rate in Wireless Environment

5.2 定性的評価

Split Database Transaction Processing System の他に、無線通信特性を考慮しない Flat Transaction Processing System, A New Transaction Management Scheme for Mobile Computing Environments, Kangaroo Transaction Processing System, 2 Phase Recovery Transaction Processing System の5つのトランザクション処理システムを対象にした定性的評価を行う。

5.2.1 評価方法

性能の比較評価を行うトランザクション処理システムの条件として、無線通信特性や無線媒体の移動性、無線携帯型計算機の実態を考慮しているかどうかが挙げられる。ただ、上記の問題を全く考慮していない一般的な Flat Transaction Processing System を基準となるトランザクション処理システムとして加える。以上を満たす定性的評価を行うトランザクション処理システムを以下の表 5.4 にまとめる。

性能評価する項目としては、無線通信特性、無線通信媒体の移動性、無線携帯型計算機の実態による影響に対応しているか、またセキュリティやスケーラビリティ等基本的な性能を考慮しているか等、9項目を定性的評価の項目として、表 5.5 に下記する。

これらの評価項目より各トランザクション処理システムを比較評価することで、Split Database Transaction Processing System が優れた性能を示す環境を提示し、また、Split Database Transaction Processing System の利点や欠点等を明らかにすることを目的とする。

表 5.4: 定性的評価を行うトランザクション処理システム

Transaction	Transaction Name
Transaction A	Split Database Transaction Processing System (SDT)
Transaction B	Flat Transaction (FT)
Transaction C	A New Transaction Management Scheme for Mobile Computing Environments (NTMS)
Transaction D	Kangaroo Transaction Processing System (KT)
Transaction E	2 Phase Recovery Transaction Processing System (2PRT)

5.2.2 評価結果

前節で列挙したトランザクション処理システムを、同様に書き連ねた性能評価項目によって、定性的評価を行った結果を以下の表 5.6 に示す。

項目 1 の無線通信特性を考慮しているかについて、Flat Transaction Processing System 以外どのトランザクション処理システムも、無線通信特性によるトランザクション処理の性能低下を改善することを目的としているため、全てが各々の方法により無線通信特性問題を改善している。

項目 2 の無線通信媒体の移動性について、Flat Transaction Processing System と 2 Phase Recovery Transaction Processing System 以外の全トランザクション処理システムが対応している。Split Database Transaction Processing System や A New Transaction Management Scheme for Mobile Computing Environments はデータベースリソースをそのままクライアント端末に転送する手法を、Kangaroo Transaction Processing System は通信経路に分身を残す方法を利用している。

項目 3 の無線携帯型計算機特性について、2 Phase Recovery Transaction Processing System と Split Database Transaction Processing System はバッテリーの制限の問題にのみ対処している。基本的にハードウェアの問題は大部分がハードウェアの技術革新によってのみ改善されるため、ソフトウェアで無線携帯型計算機特性の問題を改善するのは非常に難しい。

項目 4 のエージェントベースかサーバ/クライアントモデルかについて、Flat Transaction Processing System と Kangaroo Transaction Processing System のみサーバ/クライアントモデルである。エージェントベースのトランザクション処理システムだと、無線通信を介すインタラクション数を減少させることは可能となるが、その分エージェント特有の問題である、セキュリティ等の問題も浮上する。この問題に関しては、第 6 章にて述べる

項目 5 のステートの有無に関して、Split Database Transaction Processing System, Kangaroo Transaction Processing System, 2 Phase Recovery Transaction Processing System がステートを利用している。

項目 6 のデータベースをユーザ操作端末に転送するかという問題に関して、Split Database Transaction Processing System と A New Transaction Management Scheme for Mobile Com-

表 5.5: 定性的評価の指標

評価項目	評価項目の説明
評価項目 1	無線通信特性への対応 (対応/未対応)
評価項目 2	無線通信媒体の移動性への対応 (対応/未対応)
評価項目 3	無線携帯型計算機特性への対応 (対応/一部/未対応)
評価項目 4	エージェントベース (Agent) か, サーバ/クライアントモデルか (S/C)
評価項目 5	ステート保持の有無 (有/無)
評価項目 6	データベースをユーザ端末に取り込むか (全部/一部/否)
評価項目 7	ユーザの同時トランザクション処理への耐性 (強/中/弱)
評価項目 8	トランザクション処理システムに対する スケーラビリティへの耐性 (強/中/弱)
評価項目 9	セキュリティ (高中低)

puting Environments がデータベースリソースをユーザ端末に転送する。データベースリソースをユーザ端末に持つことで無線通信特性や無線通信媒体の移動性を抽象化することが可能となる。ただし、データベースを分離することは一貫性の保証を証明する必要が生じる。この問題に関しては第 6 章にて述べる。

項目 7 のユーザが一度に沢山トランザクション処理を実行した場合の耐性については、Flat Transaction Processing System 以外が中か弱を示している。これは、無線携帯型計算機よりトランザクション処理を実行した場合の問題を改善するために余分な操作をしている分がそのまま、耐性を弱めることを示している。

項目 8 のトランザクション処理システムが巨大になる等といったスケーラビリティに関して、トランザクション処理のインタラクション数が増えるにつれてデータベースをユーザ端末に保持する Split Database Transaction Processing System や A New Transaction Management Scheme for Mobile Computing Environments や、アボート処理へ耐性を持たず手法を利用した 2 Phase Recovery Transaction Processing System 等が優位に立つ。

項目 9 のセキュリティに関して、現時点でセキュリティまで考慮したトランザクション処理システムは余り研究されていない。ただ、トランザクション処理プロトコルが複雑になるにつれ、バグ等の発生率が高くなることと、エージェントのセキュリティを余分に考慮しなくてはならないことを考慮して、Split Database Transaction Processing System や A New Transaction Management Scheme for Mobile Computing Environments, 2 Phase Recovery Transaction Processing System のセキュリティは低いと考えられる。

表 5.6: 定性的評価

	SDT	FT	NTMS	KT	2PRT
項目 1	対応	未対応	対応	対応	対応
項目 2	対応	未対応	対応	対応	未対応
項目 3	一部	未対応	未対応	未対応	一部
項目 4	Agent	S/C	Agent	S/C	Agent
項目 5	有	無	無	有	有
項目 6	一部	無	全部	無	無
項目 7	弱	強	弱	中	中
項目 8	強	弱	強	中	強
項目 9	低	高	低	中	低

5.2.3 考察

Flat Transaction Processing System 以外は全て、無線環境下からのトランザクション処理を想定したトランザクション処理システムである。しかしながら、無線通信特性や無線媒体の移動性等によるトランザクション処理の性能低下を避ける為、Split Database Transaction Processing System や A New Transaction Management Scheme for Mobile Computing Environments, 2 Phase Recovery Transaction Processing System 等はエージェントを使用し、Kangaroo Transaction Processing System はトランザクションクライアントとトランザクションサーバの通信の途中にユニットを残していく等といった、Flat Transaction Processing System より、複雑な機能が必要である。

しかしながら、無線通信特性等によるトランザクション処理に及ぼす性能低下の問題を改善する機能により、Flat Transaction Processing System に比べ、トランザクション処理に対して、余分な動作の分がそのまま欠点となる場合がある。例えば、Split Database Transaction Processing System はエージェントを利用することで、劣悪な通信環境においても、他トランザクション処理システムより少ない性能低下でトランザクション処理を行うことができる代わりに、トランザクションエージェントのセキュリティや、ユーザの多数アクセスの観点から比較すると、現時点では他システムに劣る。

同時に多数のトランザクションクライアントとトランザクション処理を行わないトランザクション処理アプリケーションの場合、無線携帯型計算機を使用してトランザクション処理を行う場合、ユーザは Abort 等による情報の再入力を嫌う場合等を満たす場合、Split Database Transaction Processing System は優れた性能を発揮する。

第6章

結論

本章では、今後の課題としてエージェントの送受信等におけるセキュリティの強化と、全ての場合における **Split Database Transaction Processing System** の一貫性の保証の問題を示した。上記2つの課題についてその重要性を説明し、また展望について述べる。最後に、本研究についてまとめる。

6.1 今後の課題

現在の Split Database Transaction Processing System では、クライアント端末が通信切断後に再接続する際等において、必要最小限のセキュリティ対策しか対応していない。また、Split Database Transaction Processing System の大きな特徴として、通常サーバ側において保持されているデータベースの一部を分離し、転送を行う点があげられるが、データベース分離・統合の際の信頼性の保証や適した転送サイズ等もより詳しく測定する必要がある。Split Database Transaction Processing System は複雑なプロトコルであるため、実際に実社会において使用される前提として、完全にバグをなくすことがあげられるだろう。

6.1.1 セキュリティの強化

Split Database Transaction Processing System では、トランザクションエージェントを中心にトランザクション処理を行う。現時点では、トランザクションエージェント自体にクライアントやサーバ間を移動する際に、最低限のセキュリティ機能しか付加していない。エージェントという概念自体が抱える問題でもあるが、Split Database Transaction Processing System のトランザクションエージェントに成りすまし、悪意を働くエージェントプログラムや、プログラムのミス等により発生するエージェントプログラムのバグ等のセキュリティホールに対しても、慎重かつ確実に対処する必要がある。Split Database Transaction Processing System は、特に電子マネーや個人情報等、重要度の高い情報をやり取りするトランザクション処理アプリケーションをも視野にいれているため、このセキュリティ問題への対処は、最も重要な課題の一つである。

6.1.2 Split Database Transaction Processing System の一貫性保証

Split Database Transaction Processing System では、最初の無線通信時にデータベースの一部のリソースを予約し、そのデータベースリソースごとエージェントとしてユーザ操作端末に転送する。例えば、データベースリソースを抱えたまま、エージェントが何らかの障害により消滅し、また同時にトランザクションサーバにおいても問題が発生した場合においても、トランザクション処理がアポートした時のデータベースリソースの還元が正しく保証されなくてはならない。今後はデータベースリソースの分離・統合に関して、あらゆる障害が発生するケースを想定し、全ての場合においてのデータベースリソースの整合性が実現されているかを検証する必要がある。

また、今回の測定では、複数のユーザがトランザクション処理を行う環境を想定していないため、複数のユーザが Split Database Transaction Processing System を行った際のデータベース分離・統合による遅延がどの程度になるかや、一貫性がきちんと保証されるかを検証しなくてはならない。

本 Split Database Transaction Processing System は他関連研究と比較し、複雑なプロトコルを使用しているため、他関連研究に比べ、一貫性の保証が非常に困難である。一貫

性を完全に保証するためには、Split Database Transaction Processing System の処理中のいかなる瞬間に、いかなる障害が発生したとしても、常に一貫性の保証がされているかを根気良く調査し続ける必要がある。調査の結果、完全に一貫性の保証が実現して初めて Split Database Transaction Processing System が実際に電子商取引のアプリケーションとして、使用されることが可能となる。

6.2 まとめ

本論文では、はじめに無線携帯型計算機からトランザクション処理を実行する場合、いくつかの無線特性や携帯型計算機の特徴が原因でトランザクション処理システムの性能を低下される問題について述べた。そして、それらの問題を解決するアプローチとして、一部のデータベースごとトランザクションを処理するエージェントプログラムをクライアント端末に転送し、クライアント端末内でトランザクションを行い、またそのトランザクション処理を出来る限りアポートさせないという手法で、上記問題を解決する方法を提示した。一方で前記のアプローチを採用した場合、データベースリソースの予約等に関して、問題が発生しうる可能性とその対処法について示唆した。以上を考慮して、無線携帯型計算機からトランザクション処理を行う場合のデメリットを改善する、Split Database Transaction Processing System を提案、設計した。

Split Database Transaction Processing System は、通常のトランザクション処理システムと異なり、トランザクションクライアントとトランザクションサーバでトランザクション処理のインタラクションのやり取りをするのではなく、エージェントベースで行う。トランザクションエージェントは、データベースリソースの一部を保持し、またトランザクション処理の記録を行う。これにより、サーバクライアント型のトランザクション処理モデルに比べ、無線通信を介すインタラクションの回数を減らし、無線通信特性や無線媒体の変移等の影響を弱めることを実現した。また、トランザクションエージェントが蓄積する状態情報により、トランザクション処理中に何らかの障害が発生しても、アポートせずに、障害から復帰後トランザクション処理を復帰させることが可能となる。トランザクション処理をアポートさせないことでユーザに求める情報の入力の手間を減少させることを実現した。

データベースサイズやインタラクション数、各トランザクション処理に所要する時間等を設定できるトランザクション処理測定シミュレーションプログラムを実装した。高通信切断問題を考慮していない Flat Transaction Processing System と、トランザクションを細かく分離することで Abort 時の代償を減少させる Kangaroo Transaction Processing System、そして、エージェントを作成し、大部分のトランザクション処理をトランザクションクライアント内で行うことで、無線環境を抽象化する Split Database Transaction Processing System の、3つのトランザクション処理システムに対し、複数の値を設定し、性能を多面的に比較評価をした。また他関連研究とは定性的評価により、多面的に性能・特性等比較を行った。以上2つの評価により、Split Database Transaction Processing System がインタラクション数が多いトランザクション処理アプリケーションや、トランザクション

処理中に障害が頻繁に発生する環境等において、優れた性能を示すことができることを示した。

謝辞

本研究を進めるにあたり、主査として御指導頂きました慶應義塾大学環境情報学部教授の徳田英幸博士を始め、副査として貴重な御助言を頂きました慶應義塾大学環境情報学部教授の清木康博士、慶應義塾大学環境情報学部助教授の楠本博之博士に大変感謝いたします。

特に、研究に関する議論で厳しい指摘や的確なアドバイスをして下さいました慶應義塾大学環境情報学部助教授の西尾信彦博士、慶應義塾大学政策・メディア研究科の永田智大修士に感謝いたします。

また、永田智大氏、榊原寛氏、村瀬正名氏、堀江裕隆氏、権藤俊一氏、鈴木源太氏、守分滋氏、柴田悟郎氏、小泉健吾氏、米山遼太氏、米澤拓郎氏を始めとする慶應義塾大学徳田英幸研究室のmove!研究グループのメンバーや慶應義塾大学政策メディア研究科、そして、徳田・村井・楠本・中村研究室の諸氏には、研究会の活動を通して多くの御意見、御助言を頂きましたことに感謝いたします。

平成 15 年 2 月 22 日

2003 年 1 月 14 日

慶應義塾大学 政策・メディア研究科 修士 2 年

参考文献

- [1] 経済産業庁: <http://www.meti.go.jp/>
- [2] 社団法人電気通信事業者協会: www.tca.or.jp
- [3] NTT DoCoMo: <http://www.nttdocomo.com/top.html>
- [4] The Ultimate IMT-2000 Gateway on the World Wide Web: <http://www.imt-2000.org/portal/index.asp>
- [5] The Official Bluetooth Wireless Info Site: <http://www.bluetooth.com/>
- [6] フィリップ. A. バーンスタイン, エリック. ニューカマー: トランザクション処理システム入門, 日経 BP 社, 1998
- [7] ジム. グレイ, アンドレアス. ロイター: トランザクション処理 上巻, 喜連川 優 訳, (2001)
- [8] ジム. グレイ, アンドレアス. ロイター: トランザクション処理 下巻, 喜連川 優 訳, (2001)
- [9] Transaction Processing Performance Council: www.tpc.org
- [10] M. H. Dunham, A. Helal and S. Balakrishnan: A Mobile Transaction Model That Captures Both the Data and Movement Behavior, ACM/Baltzer Journal on Special Topics in Mobile Networks and Applications (MONET), Vol2, pp.149-162 (1997).
- [11] Ioannidis, J., Duchamp, D., Maguire, G.: IP-Based Protocols for Mobile Internetworking, In the Proceedings of ACM SIGCOMM'91, (1991)
- [12] 寺岡文男: 移動透過性プロトコル, ワイヤレス LAN アーキテクチャ第 4 章, 1996, 共立出版
- [13] The Source for Java(TM) Technology: <http://java.sun.com/>
- [14] Sanjay Kumar Madria: Transaction Models for Mobile Computing, in proceedings of 6th IEEE SICON, Singapore, World Scientific, 1998

- [15] Khalil M. Ahmed, Mohamed A. Ismail, Nagwa M. El-Makky, Khaled M. Nagi: A New Transaction Management Scheme for Mobile Computing Environments, (2001)
- [16] C. Pu, G. Kaiser and N. Hutchinson: Split-transactions for Open-ended Activities, Proceedings of the 14th VLDB Congerence, pp.26-37 (1998).
- [17] 梅染 充男, 古坂 大地, 岩本 健嗣, 西尾 信彦, 徳田 英幸: 無線環境に適したトランザクション・システムの構築, 第 61 回情報処理学会全国大会, pp.27-28(2000)
- [18] 梅染 充男: 無線環境に適したトランザクション処理システムの研究, 2000 年度卒業論文集, 慶應義塾大学徳田研究室, (2001)
- [19] H. Tokuda: Compensatable Atomic Objects in Object-oriented Operationg Systems, Proceedings of the 1st PCCS, pp.210-219 (1985).
- [20] Stefan Pleisch, Andre Schiper: TranSuMA: Non-Blocking Transaction Support for Mobile Agent Execution, ICDCS 2002 The 22nd International Conference on Distributed Computing Systems, (2002)