

ALTIBASEテクニカルオーバービュー

Altibase Technical White Paper

2007/10/14

著作権

このドキュメントに記載されている情報は、このドキュメントの発行時点におけるアルティベース、及びシアンズ・アールの見解を反映したものです。アルティベース、及びシアンズ・アールは市場の変化に対応する必要があるため、このドキュメントの内容に関する責任を問わないものとします。また、発行日以降に発表される情報の正確性を保証できません。

このホワイト ペーパーに記載された内容は情報の提供のみを目的としており、明示、黙示または法律の規定にかかわらず、これらの情報についてアルティベース、及びシアンズ・アールはいかなる責任も負わないものとします。

このソフトウェアおよびマニュアルは、本製品の使用許諾契約書のもとでのみ使用することができます。このソフトウェアおよびマニュアルのいかなる部分も、アルティベース、及びシアンズ・アールの書面による許諾を受けることなく、その目的を問わず、どのような形態であっても、複製または譲渡することは禁じられています。ここでいう形態とは、複写や記録など、電子的な、または物理的なすべての手段を含みます。

アルティベース、及びシアンズ・アールは、このマニュアルに記載されている内容に関し、特許、特許申請、商標、著作権、またはその他の無体財産権を有する場合があります。このマニュアルはこれらの特許、商標、著作権、またはその他の無体財産権に関する権利をお客様に許諾するものではありません。

特に記載していない場合、例として登場する企業、組織、製品、ドメイン名、電子メールアドレス、ロゴ、人物、場所、およびイベントはすべて架空のものです。実在する企業、組織、製品、ドメイン名、電子メールアドレス、ロゴ、人物、場所、またはイベントとは一切関係ありません。

© 2007 ALTIBASE Corporation. All rights reserved.

ALTIBASEアーキテクチャ.....	5
メモリ管理機能	7
メモリデータベース管理	7
ディスクデータベース管理	8
フラグメントの解消	8
データベース空間	8
ロギングと復旧	9
トランザクション処理	11
同時実行効性制御	11
トランザクション管理	12
同時ユーザーアクセス処理	13
クエリ処理	14
プログラミングインタフェース	15
ODBC	15
SQLCLI	15
JDBC	16
SES PreCompiler	16
PSM(Persistent Stored Module)	16
レプリケーション	17
データベースの二重化とは?	17
ALTIBASEでの2重化実装方式	17
二重化の特徴	18
二重化インタフェース	18
二重化のミッション	19
Message Queuing System	20
ALTIBASE Message Queing System	20
ALTIBASE Message Queing Systemの特徴	21
ツール	21
DbAdmin	22

iSQL.....	22
iLoader.....	22
Shmutil.....	22
Audit.....	23
パフォーマンス.....	23
TPC-H性能.....	23
単純なクエリの性能.....	25

ALTIBASEテクニカルオーバービュー

1980年代に登場したRDBMS(Relational DBMS)は、IT環境の変化に歩調を合わせて発展を繰り返して来た結果、企業内情報システムの中核のインフラとして位置付けられるようになりました。一般的にディスクを保存メディアとするRDBMSは、基本データベースの管理だけでなく、データウェアハウス、データマイニングなど、多くのアプリケーション分野に発展しました。特に、データとアプリケーションを分離するという概念は非常に有用なものであり、DBMSの爆発的な成長をもたらす起爆剤として働きました。

ALTIBASEアーキテクチャ

ALTIBASEはリレーショナルデータベースモデルをサポートし、汎用的なアプリケーションシステムとリアルタイムアプリケーションシステムに対応した次のような特性を持っています。



クライアント・サーバー型アーキテクチャは、一般的なアプリケーションに適合した構造で、アプリケーションプログラムは様々な通信方法によってサーバーに接続します。一方、アプリケーション組み込み型アーキテクチャでは、アプリケーションプログラムをデータベースサーバーに内蔵することによって一つのプロセスで運用する方式です。このような構造は、アプリケーションプログラムとデータベースサーバー間の通信オーバーヘッドを省くことで、高性能のトランザクション処理が可能となります。ユーザは、アプリケーションプロ

グラムの特徴や運用環境の特徴に適合したデータベースサーバーアーキテクチャを選択することができる柔軟性を持つことができます。

一般的に知られているように、マルチプロセッサアーキテクチャのシステムは、ユーザが増加することにつれて必要なシステムリソースが急増し、プロセス間のスワップコストが非常に大きくかかります。一方、マルチスレッドアーキテクチャのシステムは、スレッド単位で管理されるため、システムリソースを少なく消費し、スレッド間のスワップコストも少なくなります。

マルチスレッドの長所を生かしたAltibaseサーバーは、サーバー機能をサポートする多数のシステムスレッドとユーザの接続によってサービスを提供するサービススレッドプールで構成されています。サービススレッドプールは、サーバー接続プールと連動し、限られた個数のスレッドを通じて多くのユーザにサービスを提供します。このような構造により、サーバーが使うリソースを最小化し、システムの拡張性と可用性を最大化することができます。

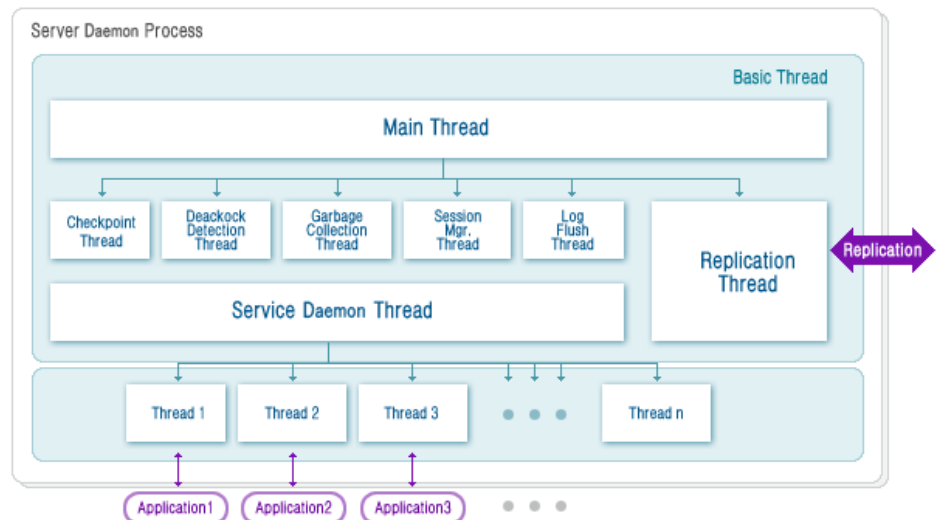


図 1 ALTIBASE サーバーの内部構造

ALTIBASEは、モジュール別の独立性を考慮し、階層型構造で設計することにより、アルゴリズムを単純化しました。各モジュールを単純化することで全体的な性能アップを図っています。また、プラットフォーム独立階層を開発したことにより、様々なプラットフォームへの移植が非常に簡単です。現在、ほとんどのUnix系列、Windows、そしてQNXのようなRTOSに移植されています。

また、クライアントの効率的な業務処理のため、TCP/IP、UNIX DOMAIN、

IPCなどの通信方法を提供することにより、ユーザの業務特性に合わせた様々な通信方法が選択できるので、アプリケーションプログラムの開発や運用に柔軟性を提供します。特に、IPC通信モジュールは、共有メモリによる高速な通信方法を提供します。

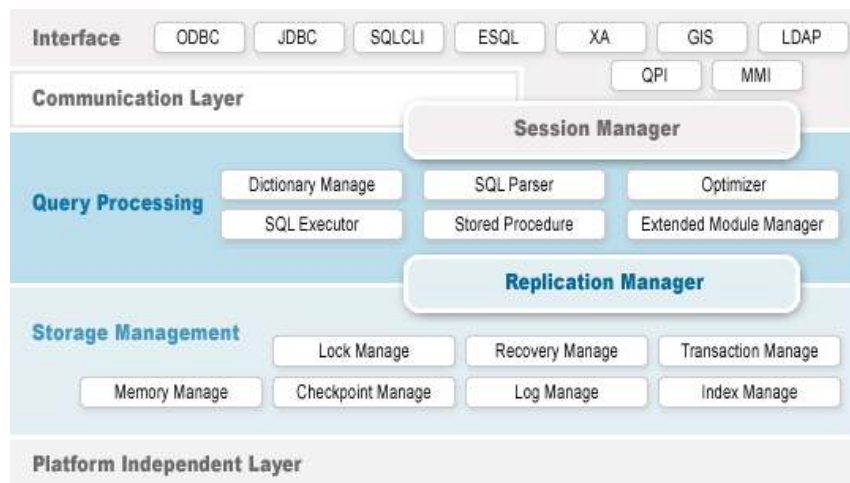


図 2 ALTIBASE 構成要素

メモリ管理機能

メインメモリデータベースのキーポイントは、データベース用として大量のメモリが必要な分、メモリを慎重にうまく利用することにあります。すなわち、メインメモリに最適化したデータベース構造を設計して管理する技術によって大きく性能の差が現れることになります。

メモリデータベース管理

高性能データベースを効率的に管理するため、ALTIBASEは、各階層で効率的にメモリを使用するよう設計されています。データベースのようなシステムソフトウェアでは、単純なメモリの割り当てや、値を設定する演算も性能に非常に大きな影響を与えるため、ALTIBASEは、メモリプールを利用したメモリ管理モジュールを非常に慎重に設計して実現しました。また、ALTIBASEのストレージ管理層は、メインメモリに最適化された保存単位でページを構成し、各ページの連携性を最大化することにより、データベースを効率的に保存・管理します。クエリープロセッサ層は、クエリー処理の際に必要な

一時的空間(作業領域)を効率的に管理するため、実行中に不要なメモリソースの割り当てや解放による性能低下の要素を最小化しました。

ディスクデータベース管理

大容量データベースの効率的な代案として、ALTIBASEは、メモリとディスクの2つの保存領域を一つのデータベースで提供します。メモリ領域と同様にディスクDBの保存領域は、従来のDRDBMSで提供するLRUアルゴリズムをベースとしたバッファプール領域や物理ディスク保存領域をサポートします。ユーザが要求するデータをバッファプールにキャッシュして使用し、ディスクのアクセス速度に対する性能低下の要素を最小化しました。

フラグメントの解消

メインメモリデータベースを運用してみると、実際に特定のテーブルに必要なメモリ空間よりはるかに多い量のメモリを占有してしまう場合があります。これは、主に大量のデータが特定のテーブルに挿入された後、変更や削除が頻繁に行われた時に発生しますが、このような場合、該当テーブルから不要なメモリをシステムに返還することができれば、より効率的にメモリを使うことができます。ALTIBASEは、テーブル単位のcompaction機能を提供しており、この機能を利用することで、メモリやテーブルを効率的に管理することができます。

データベース空間

ALTIBASEは、メインメモリ上のデータベース空間を共有メモリまたはプロセスローカルメモリに確保できる機能を提供します。共有メモリをデータベース空間として使うと、システムが正常な状態では、ALTIBASEの再実行にかかる時間を非常に短くすることができます。これは、共有メモリ内のデータベースが安全に維持されているため、バックアップデータベースからデータを読み込む必要がなく、共有メモリ内のデータベース領域をALTIBASEサーバープロセスにマッピングして使うことができるからです。

メインメモリ内のデータベース空間は、永続空間と一時的空間に分けられます。永続空間は、実際のテーブルとメタ情報に関するデータを保存しています。この空間は、ディスクに存在するバックアップデータベースの内容を反映しています。一方、一時的空間は、インデックスデータやクエリー処理中

に発生する作業用のテーブルが置かれる場所です。この空間は、ディスクにあるバックアップデータベースには反映されず、ALTIBASEサーバーが終了すると消えてしまう空間です。ALTIBASEのインデックスデータはバックアップデータベースに保存されないため、一時的空間に保存・管理されます。ALTIBASEは、サーバーが開始時にシステムカタログからインデックス情報を取得して、一時的空間にインデックスを速やかに生成します。インデックスをバックアップデータベースに保存しないことにより、ALTIBASEは、サーバー実行中にインデックスの変更に対してロギングする必要がないため、その分データベースの更新性能をアップさせることができました。

継続したデータの挿入により、メインメモリ内のデータベースの永続空間が足りない場合、ALTIBASEは、自動的にデータベースの永続空間を一定のサイズに拡張して管理します。もちろん拡張されたデータベースの永続空間はバックアップデータベースに反映され、データの永続性を保障します。

ALTIBASEには、メモリデータベース領域からディスクデータベース領域にテーブルを移動できる機能があります。この移動の機能は、同じスキーマのテーブルが存在する場合、メモリデータベース領域のデータをディスクデータベース領域に移し、該当データを元のテーブルから削除する機能です。このようにすることにより、アクセスの多いデータやアクセスが比較的少ないデータを分離して保存し、応答性能を最適化し、保存空間をさらに効率的に使うことができます。

ロギングと復旧

ALTIBASEは、伝統的なトランザクション概念のACID特性を完全にサポートすることを基本とします。

ACIDは、一般的なデータベースシステムがトランザクション処理のために基本的に提供する特性として、原子性(atomicity)、一貫性(consistency)、独立性(isolation)、永続性(durability)の4つを意味し、ALTIBASEではこのような特性を全て満足させるトランザクション処理方法を提供します。特に、永続性を提供するためにはディスクへの書き込みを保証する機能が必須であり、これはディスクへのアクセスによる性能低下を引き起こします。

ALTIBASEでは、独自の技術開発により高速のトランザクション処理を保

障しつつ、トランザクションの永続性を提供する特徴を持っています。

実際、サービスが行われる環境では、偶発的なサーバーの障害によりデータベースのシステムダウンが発生する可能性もあり、そのような状態でのデータベースの回復は進行中であるトランザクションの状態を正確に反映、もしくは完全に撤回できなければなりません。ALTIBASEは、ロギングの際、WALプロトコルを適用して様々な状態のトランザクションを回復時点で正確に反映することにより、トラブル発生前の状態に完全に復旧されることを保障します。

ALTIBASEは、構築されたデータベースをファイルに、暗黙的または明示的な方式でディスクのバックアップデータベースファイルに同期化(チェックポイント)します。このチェックポイント処理では、全ての変更されたページがディスクに反映されなければなりません、その時に該当ページの一貫性を維持するためにPage Latchをかけなければならない場合が発生します。しかし、ハイブリッドメモリデータベースの特徴である高速性能を維持することを考慮した場合、Page Latchを使うのは、現在実行中のトランザクションとの競合を引き起こし、それはチェックポイント処理の際にシステム全体の性能を急激に低下させる危険性があります。このような問題点を解決するため、ALTIBASEはチェックポイント処理の際に、Page Latchを使わない代わりにバックアップディスクを二つ置き、交互に反映するようにしたピンポンチェックポイントを基盤として設計されています。これにより、チェックポイント処理の実効により、現在実行中のトランザクションに全く負荷を与えなくなり、結果的にシステム全体の性能アップをはかることができました。

ALTIBASEシステムのページのサイズやファイルシステムの物理的なページサイズが異なった場合、Disk I/Oの実行中に異常終了されると、ページが完全ではない状態で残されることがあります。このような現象を防ぐためにALTIBASEは、ページをフラッシュする際に、ディスクの特定の領域に存在するダブルライトバッファ領域に同じイメージを予め保存しておきます。そして、ALTIBASEの再起動時に、ダブルライトバッファの内容と実際のページの内容をチェックし、不完全なページを復旧します。

トランザクション処理

同時実行効性制御

伝統的な 2PL(two phase locking)プロトコルを使うデータベースの場合は、仮に該当ロックのレベルがレコードだとしても、該当レコードに対する変更処理が発生した場合、該当レコードに対する読み込み処理が遅延される問題が発生します。また、特定のレコードに対する読み込み処理が発生した後は、そのトランザクションがコミットされない場合は、そのレコードに対する変更トランザクションが持続的にスタンバイしなければならない状況も発生します。更に、大量のレコードに対する読み込みまたは変更処理が発生した場合は、レコードに対する読み込みや書き込みコストより同時性制御を行うロックに対する処理により多いコストがかかることになり、ロックエスカレーションが頻発するため、トランザクションの同時実行性能を急激に低下させる危険性が存在します。

ALTIBASE が提供するマルチバージョン技法 MVCC では、トランザクション実行時間に対し、それぞれ異なったバージョンを維持するため、レコードレベルの読み込み処理は、変更処理と関係なく進めることができ、変更処理も読み込み処理とは関係なく進めることができます。

特に、ALTIBASE は、レコードレベルのロックメカニズムをサポートしつつ、該当レコードに対するロック情報を直接レコード内部に保存するため、ロック処理にかかるコストがほとんどゼロに近いだけでなく、読み込み処理の場合は、あえてロック情報を持たずに進められるように設計されました。また、一つのトランザクションが大量のレコードに対して読み込みや変更処理を実行しても、ロック管理に対するコストがほとんど発生しないため、高速な応答性能を保証します。

ALTIBASE は、メモリーテーブルやディスクテーブルに対し、外見上は同じ機能を提供しますが、お互いに異なるアクセス方法で MVCC を実現しました。メモリーテーブルは、行を変更する度に新しいバージョンを生成する out-place MVCC で実現されており、ディスクテーブルの場合は、変更されたデータを既存の行に上書きし、変更前の情報を undo table space に保存して参照する in-place MVCC 方式を採用しました。

メモリーテーブルで out-place マルチバージョン技法をサポートすることにおいて負担となる部分は、トランザクションがそれ以上アクセスできない前のバージョンのレコードに対する処理です。メモリーデータベースは、システムメモリーを持続的に使うため、削除されたか変更されたレコードに対する領域を解放をしなければ、いつかはそのシステムの使用可能なメモリーを全部使い切ってしまう、結局サービスが不可能な状態におちいる可能性があります。

このような理由から、メモリーデータベースでは、それ以上必要のないレコードまたはインデックスノードが存在した場合、すぐ回収して再利用できるメカニズムが必要です。ALTIBASE では、このような役割をするガーベージコレクションスレッドを生成し、最適なメモリー状態が維持できるよう保障します。

トランザクション管理

ALTIBASE は、MVCC を利用して同時性制御を行います。MVCC 環境では、特定の時点に特定のレコードに対するアクセスを行った他のトランザクションの状態、つまり現在実行中なのか、あるいは既に終了したものなのか等の情報を即座に決めなければならない制約が存在します。もしこのような判断のためのコストが多くかかる場合、マルチバージョン技法の処理コストが非常に大きくなり、返ってトランザクション処理性能を低下させてしまいます。

このような要求条件を満たすために ALTIBASE では、トランザクションプールを維持し、そのプールに対する直接的なアクセスにより高速にトランザクションの実行の有無を判断できるようにしました。特に、トランザクションプールを予め作成しておく構造は、デッドロック処理にも大きなメリットを持ちます。一般的なデッドロック検知技法は、トランザクションの間にサイクルが存在するかどうかを検査する別のプロセスあるいはスレッドが存在し、一定周期で全ての使用中のトランザクションを検査するものです。この構造は、必然的にデッドロックに参加したトランザクションの一時的なサービス中止状態をもたらします。高速な応答速度を保障しなければならないハイブリッドメモリーデータベースにおいて、デッドロックが発生した後、一定時間の間サービスが中止されるということは極めて致命的な結果をもたらすため、このようなデッドロック検知技法は適切ではありません。そのような問題点を解決するため、ALTIBASE では、トランザクションのロック要求の際に、トランザクシ

ョンプールを利用して非常に速い速度で自分がデッドロックを発生させたかどうかを検査するアルゴリズムを適用し、デッドロックによるトランザクションの致命的な遅延現象を抜本的に取り除けるように設計しました。

また、ALTIBASE のトランザクションの分離レベルは、commit read を基本として、repeatable read や no phantom read レベルを動的にサポートするため、ユーザの必要に合わせて適切な分離レベルを選択して使うことができます

同時ユーザーアクセス処理

データベースに対する同時ユーザ数は、データベースの特性や該当システムの容量と密接な関係があります。しかし、システム容量が増える比率だけ実際にデータベースがサポートする同時ユーザ数は比例関係を持たないことが一般的であり、特にデータベースがサポートするサービスアーキテクチャがこのようなリソースに関する問題と直接的な関連性があります。

サービス当り一つのプロセスを生成する構造を持つアーキテクチャでは、ユーザ数分のプロセスが生成されなければならず、それによるリソースの消費が激しい状態では、大容量ユーザをサポートするための解決策を見つけるのが困難です。限られたプロセスで大量のユーザをサポートするとしても、リソースをユーザ間で時間差をつけて分けて使うこととなり、完全な解決策であるとは言い難いと言えます。

ALTIBASEでは、スレッドアーキテクチャを基盤にサービススレッドプールやサービスセッションプールを提供し、二段階の構造的基盤を提供します。サービスセッションプールは、クライアントの要求を直接担当し、クライアントに情報を返還するセッションを維持するものであり、サービススレッドプールは、このようなクライアントのサービスを実際に下位モジュールにおいて実行するものです。サービスセッションプールやサービススレッドプールの数は、プロパティにより該当システムの負荷に合わせて適切な値を選択することができるので、必要以上のサーバーリソースを消耗することがないようにします。また、クライアントに対する高速な応答を保障するために、一定の数までのクライアントにはサービススレッドとの1:1接続により最大の性能を保障し、一定の数以上のクライアントが接続される場合は、N:Mの接続に自動変換し、サーバーリソースを効率的に利用します。

クエリ処理

特殊なアプリケーションに合わせて開発されたリアルタイムシステムの場合、多くのシステムはクエリ処理言語(SQL)を提供しないか、SQLの一部の機能だけを提供するAPI方式によりアプリケーションプログラムを作成する必要があります。これは、アプリケーションプログラムの開発を難しくし、開発費用を膨らませ、メンテナンスも困難にします。

ALTIBASEは、Ad-hoc方式のプログラミングの開発ではなく、業界標準のSQL-92をサポートすることにより簡単にデータを変更してアクセスできる方法を提供するため、開発期間を短縮し、メンテナンス費用も大幅に削減することができます。

また、単純なクエリ処理機能や性能のみに重点を置いている他の製品とは異なり、通信サービスなどのように単純なクエリ性能を要求する特定分野はもちろん、複雑な分析や処理のために様々な機能のクエリ実行を要求する汎用分野にも適用できるように、高性能、高機能、大容量のクエリ処理を提供します。

ALTIBASEのクエリ処理機能は、最適化されたメモリ管理やディスク管理により、高性能のクエリ処理を提供します。例えば、メモリやディスクの特性に合わせて最適化されたcost-based optimizerの提供、join optimizerによるNested Loop Join、Hashingまたはソートを利用したJoinの実行方法をサポートします。また、高速な照会のために効率的なインデックス使用アルゴリズムをサポートすることにより、高性能なクエリ処理を行います。

ALTIBASEのクエリ処理機能は、機能的な側面において一般的なInner Joinだけでなく、Outer Join(full, left, right outer join)など、様々なジョイン機能のサポート、複雑なスキーマのアプリケーションプログラムに適合したサブクエリ処理やインラインビュー機能、ユーザがデータベーススキーマによってクエリ実行プランを照会できる機能、多様なシステム提供の関数や条件文のサポート、SET演算(union、intersect、minus)のサポートなど、汎用的なRDBMSと同様に複雑なアプリケーションプログラムを簡単に開発できる多くの機能を提供します。また、データがメモリ領域に保存されていても、ディスク領域に保存されていても、意識することなくJoin文を自由かつ制限なしに使える特徴を持っています。

さらに、複雑なアプリケーションプログラムをデータベースに保存・管理することが容易なストアド・プロシージャ機能、sequence機能、foreign key機能などのDDL、DML、DCL、レプリケーション関連のSQL文などをサポートすることにより、SQL文を使ってデータベースを簡単に管理することができます。

ALTIBASEのクエリ処理機能は、アクセスするレコードがメインメモリに存在する特徴を生かし、それに最適化されたアルゴリズムで高機能のクエリを高性能で処理することにより、色々な分野に適用され、アプリケーションプログラムの開発を容易に行うことができます。

プログラミングインタフェース

ALTIBASEは、従来のディスク型データベースがサポートする多くの業界標準のプログラミングインタフェースを提供しており、既存の資産を活かしながら、データベースアプリケーションプログラム開発を容易に行なうことができます。

ODBC

Windows環境において、汎用的なアプリケーション開発ツールを使用して、データベースサーバーに接続しデータを活用するためには、ODBCドライバを利用します。ALTIBASEは、ODBCドライバを提供しており、ユーザはこのドライバを自分のWindowsシステムに登録することにより開発ツールからALTIBASEを利用することができます。ALTIBASEのODBCは、Core Level、Level 1、そして Level 2の仕様をサポートしています。

SQLCLI

ALTIBASEは、CまたはC++開発者のためにUnixやWindows OSを全てサポートするSQLCLIインタフェースを提供します。SQLCLIインタフェースの仕様は、ODBCと同じです。

ALTIBASEのCLIは、X/OPEN CLIの標準仕様を守ります。SQLCLIは、アプリケーションプログラムがデータベースサーバーと独立して稼動し、アプリケーションプログラムの移植性を高めるためのものです。

このインタフェースを有効に使うと、高速な性能を持つアプリケーションプログラムが作成できますが、慣れていないプログラマーが使うには多少難

しい点があります。

JDBC

Web環境において、データベースアプリケーションプログラムを開発したり、アプリケーションプログラムの移植性を考慮してJAVAの仮想マシンで行われるデータベースアプリケーションプログラムを開発するために、ALTIBASEは、JDBCをサポートします。JDBCのインターフェースを活用すると、ALTIBASEとBEA社のWebLogicなどのJavaアプリケーションサーバーとの連携が容易になり、JSPなどを利用してALTIBASEアプリケーションプログラムを開発することができます。

AltibaseのJDBCは、JDBC 2.0仕様をサポートしています。

SES PreCompiler

SES C/C++プリコンパイラを使用すると、SQLCLIに比べて比較的簡単にAltibaseアプリケーションプログラムが作成できます。CまたはC++プログラミング言語でEmbedded SQLを使うことにより、ALTIBASEのデータベースへ簡単にアクセスできますので、プログラマーの記述するコード量を削減し、SQLCLIと同等の性能を持つプログラムが作成できます。

SES C/C++は、OracleのPro*C/C++と同じ仕様をサポートするため、従来のOracleのアプリケーションプログラムをALTIBASEに簡単に移植できます。

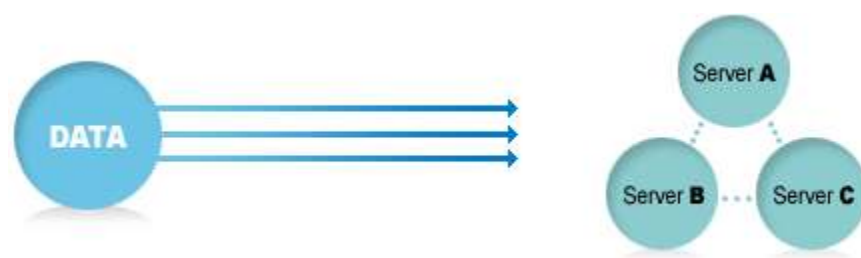
PSM (Persistent Stored Module)

PSMは、OracleのStored Procedureのような機能を提供するデータベーススクリプト言語です。ユーザは、複雑なビジネスロジックをPSMで作成してALTIBASEサーバーに保存しておき、必要な時にクライアントから呼出して使用することで、全体的にデータベースの性能を向上できます。PSMは、ストアド・プロシージャや関数をサポートし、ビジネスロジックが簡単に作成できるように、if、case、while、for、loop、continue、exit、null statementのようなフロー制御文を提供しています。

レプリケーション

ミッションクリティカルな分野のデータベースサービスは、どのような場合でもサービスが中断されてはいけません。もし、データベースサーバーシステムの障害またはデータベースメディア(ディスク)の破損により、サービスが中断された場合、結果的にサービスのできなかつた分だけ経済的な損失を被るのではなく、会社の信頼度も低下することになります。

このような問題を解決するための代案が、データベースの二重化機能です。



データベースの二重化とは？

物理的に離れている複数のデータベースに対し、ローカルデータベースの変更された内容を遠隔データベースにコピーして管理することを言います。従って、ユーザは、一つのデータベースについてのみ作業を行ってもデータベースの二重化システムで連携している他のデータベースにも作業内容が同様に適用され、複数のデータベースを同時に管理することができます。このようなデータベースの二重化は、データベースの無停止サービスを可能にします。

ALTIBASEでの2重化実装方式

ALTIBASEは、データベースの変更ログを基盤とする Point-to-Point の二重化技術を使用しています。ローカルサーバーがデータベースの変更ログを XLOG という実行計画に置換えて遠隔サーバーに転送し、遠隔サーバーは、この XLOG からトランザクションを回復するのと同様な方式で二重化を行います。

この際、ALTIBASEは二重化サーバーの間にローカル性を保障するため、二重化トランザクションはローカルトランザクションに影響を与えません。また、二重化を利用した負荷分散構成を実現するために、Active-Activeの二重化モードを提供しています。このような二重化の実現のため、ALTIBASEは、内部的にReplication Manager、Replication Sender、Replication Receiverという二重化スレッドを使用します。

二重化の特徴

- テーブル単位の二重化
- SQL と類似したユーザインタフェース
- 二重化のオブジェクトを導入し、二重化の情報をデータベースに保存して運用
- サーバーの障害またはネットワーク切断の自動検知・対応
- 二重化データの更新競合に対する自動解決
- 二重化中も独立システムの性能の 90%以上を維持
- 遠隔サーバーの性能がローカルサーバーの性能に影響を与えないため、サーバー間のローカル性(locality)を保障
- Active-Active の運用形態による負荷分散を提供
- 相手サーバーのシャットダウン中に発生した DB の変更も復旧後に二重化を保障

二重化インタフェース

ALTIBASE の二重化インタフェースは、SQL と類似した構文で提供することにより、対話型 SQL ツールまたはアプリケーションプログラムから利用することができます。二重化のインタフェースは、以下の通りです。

- `create replication rep1`
`with remotehost、 portno`
`from localtableA to remotetableA、`
`from localtableB to remotetableB、 ...;`
- `drop replication rep1;`
- `alter replication rep1 start;`
- `alter replication rep1 stop;`
- `alter replication rep1 sync;`

➤ alter replication rep1 quickstart;

二重化のミッション

ALTIBASE の二重化は以下のようなミッションを基に開発されました。

- High Availability

サービス中のシステムまたはソフトウェアで障害が発生した場合、使用可能なシステムへ即時にアクセスができるようにデータ二重化を提供します。また、複数のシステムにデータを二重化できるように水平的な拡張性をサポートします。

- Database Consistency

Active-Active の環境でデータベースの二重化を行うと、一つのデータベースサーバー内で二重化トランザクションとローカルトランザクションが同時に同じデータにアクセスする場合があります。そのような場合、データの衝突(conflict)が発生しますが、ALTIBASE はルールベースでデータの衝突を自動解決します。衝突した内容を特定のログファイルに記録しておくことにより、管理者がその内容を閲覧し、適切な対応をとることができます。

- High Performance

データベースを二重化することに伴うオーバーヘッドを最小化し、独立システムでトランザクションを処理する時の性能をほぼそのまま維持できます。ローカルサーバーのデータへのアクセスや二重化のための作業を最適化することにより、ALTIBASE のトランザクション処理にかかる負担を最小化しました。データベースの変更ログを二重化ログの XLOG 構造に変換して遠隔サーバーに転送し、この XLOG を利用してトランザクションを復旧するような方法で二重化を行います。この方法は少し複雑ではありますが、ローカルサーバーの性能低下を最小化するだけでなく、二重化そのものの性能も最大化します。

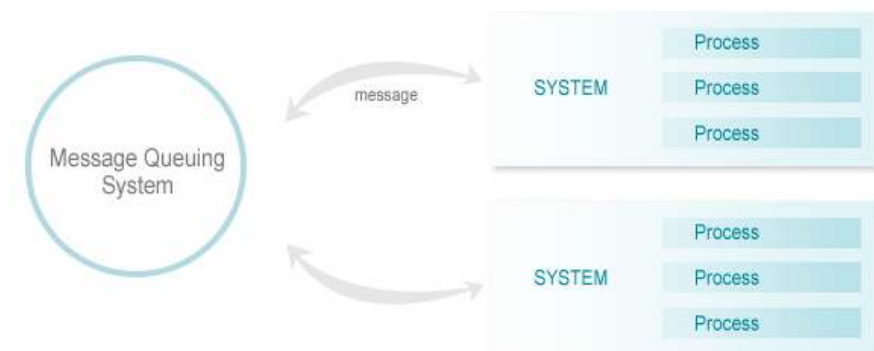
- Load Balancing & Scalability

ALTIBASE のマルチサーバー運用環境でサービスするトランザクションを二つのグループ以上に分け、それぞれのトランザクションが該当サーバーで行われるようにし、各サーバーで変更されるデータベースの内容を他のサーバーに反映させることにより、サーバーにかかる負荷を分散させることができます。

Message Queuing System

Message Queuing System は、それぞれのアプリケーションがメッセージを利用し、双方円滑に通信できるようにサポートするシステムのことを言います。この際、通信する送信側と受信側の2つのアプリケーションは、単一システムの内部または物理的に区分されていますが、ネットワークでつながった他のシステムに存在することもあります。

Message Queuing System は、2つのアプリケーションをつなぐためのソケットの生成や、パケットに対する flow control、例外状況の処理に対する全てのものをアプリケーションで直接処理しなければならない煩わしさを最小化します。



ALTIBASE Message Queing System

ALTIBASE は、Message Queuing System で使われる Queue を一般ユーザテーブルとして取り扱い管理します。それぞれの Queue に対するロギング機能やトランザクション機能、同時性制御などのような ALTIBASE の全ての機能を Message Queuing System でそのまま使うことができます。

ALTIBASE Message Queing Systemの特徴

- SQL 文の Queue 関連インターフェースの提供により、従来の INSERT/SELECT 文の機能をそのまま使うことができるため、ユーザが簡単に使うことができる。
- Queue テーブルに対する二重化構成のサポート。
- Queue に搭載する最大メッセージ数の指定が可能。
- Queue に挿入されるメッセージのデータタイプを任意で指定可能。一旦、Queue を生成した後、alter table 構文を利用してユーザが希望するデータタイプを持つカラムを追加して使うことができる。
- enqueue されるそれぞれのメッセージに対し、enqueue time を自動で記録。
- メッセージに対し、ユーザが付与するメッセージ id である correlation id 機能をサポート
- メッセージに優先順位をつけ、enqueue/dequeue の際に優先順位によるメッセージ処理が可能。
- dequeue するメッセージがない場合、waiting option を提供することにより、ユーザの柔軟な対処が可能。
- browse、remove などの二つの dequeue mode をサポート
- SQL 文の形の Queue 関連インターフェースの提供により、C/C++、JAVA などのような全てのアプリケーションプログラムで使用が可能。
- enqueue/dequeue されるメッセージ数だけでなく、システムに存在する Queue リスト、Queue が使うメモリサイズなど、Queue と関連した様々な内容に対するモニタリング方法を提供。

ツール

ユーザがデータベースアプリケーションプログラムを効率的に開発・運用するためには、様々な付加機能を提供するデータベースツールが必要となります。例えばクエリ文を簡単に実行できるツールやテキストファイルをデータベースにアップロードできる機能です。また、メモリの効率的な管理のためには、メインメモリデータベースのメモリ使用量を定期的に監視する必要があります。場合によっては大量にメモリを使用しているテーブルのデフラグを行う

必要もあります。ALTIBASEはユーザが効率的かつ便利な方法でメインメモリデータベースが管理できるように以下の様なユティリティを提供します。

DbAdmin

このユティリティは、ALTIBASEのサーバー管理ツールです。データベース管理者は、このユティリティを利用してALTIBASEサーバーを起動/停止させ、サーバー運用中にその状態を把握し管理することができます。特に、定義情報、セッション情報、メモリ使用情報、データベース情報、二重化情報を参照し、管理できる機能を提供します。

iSQL

iSQLは、対話型クエリ処理のユティリティです。このユティリティを利用してユーザはALTIBASEがサポートする全てのSQL文を直接実行することができます。また、Stored procedureやstored functionを生成・実行することもできます。その外にも、最近使ったコマンドの自動保存や再実行ができ、データ検索結果の出力フォーマットを編集することができます。

iLoader

データベースを決まったフォーマットのファイルにダウンロードしたり、アップロードしたりする機能をサポートするツールです。このダウンロードファイルフォーマットは、Oracle、SQL-Serverと互換性があり、OracleまたはSQL-ServerからダウンロードしたデータをAltibaseにアップロードが可能であり、AltibaseがダウンロードしたデータをOracleまたはSQL-Serverにそれぞれアップロードすることができます。

Shmutil

メインメモリ内のデータベース領域で、プロセスローカルメモリの代わりに共有メモリを使うことができます。共有メモリを使う場合は、特別な管理が必要ですが、そのためにALTIBASEはshmutilユティリティを提供します。このユティリティは、データベースを格納した共有メモリの情報を提供、その共有メモリをシステムに返還する機能、共有メモリ上のデータベースを特定のファイルにバックアップする機能などを提供します。また、ALTIBASEサーバーを再実行する前に、共有メモリ上のデータベース状態が完全なのかどうかを判断する機能も提供します。

Audit

Audit は、二重化関係にあるテーブル間の整合性を維持するための機能を提供するユーティリティです。二重化テーブルの間には、挿入の不一致、更新の不一致、削除の不一致がある場合があります。Audit は、テーブル単位で比較・検査し、自動的に不一致情報を出力する機能や、不一致が発生した場合、二つのテーブルを一致させる機能を提供します。

特に、不一致の情報の記録を SQL 文の形で残しておくことにより、管理者がその内容を検索し、直接管理できるようにします。また、Audit は、ALTIBASE のデータベースと Oracle のデータベースをテーブル単位で一致させる機能も提供しています。

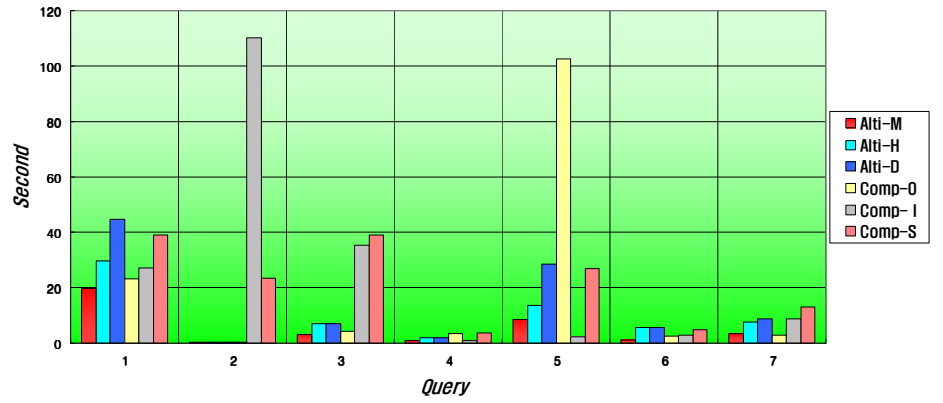
パフォーマンス

TPC-H性能

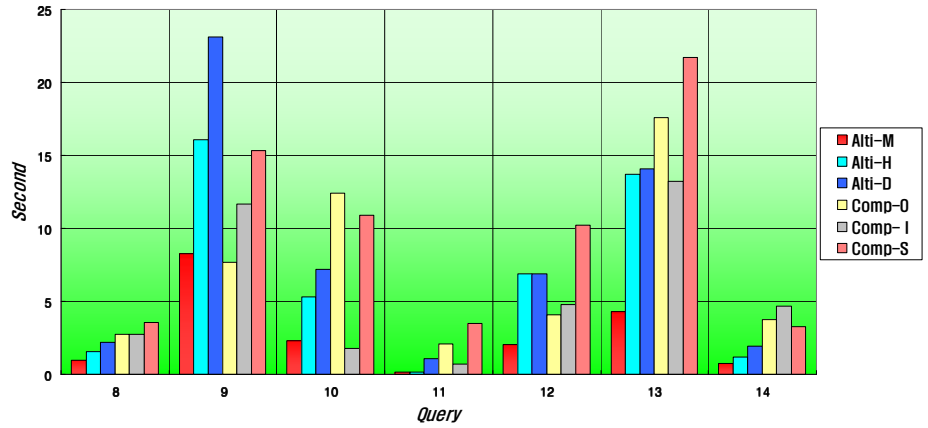
従来のメインメモリデータベースの用途は、通信分野などにおいて単なる簡単なトランザクションを高速で処理するために主に使われてきましたが、最近においてはビルディング、顧客管理など、複雑なアプリケーションにもメインメモリデータベースの適用が増えている傾向にあります。

また、その他のメインメモリデータベースがまだ単純なクエリ処理の性能に重点をおいていますが、ALTIBASEは複雑なクエリ処理をサポートするだけでなく、その処理性能もより優れたものとして開発されました。複雑なSQLの処理性能に対するベンチマーキングツールとしてはTPC-Hがあります。

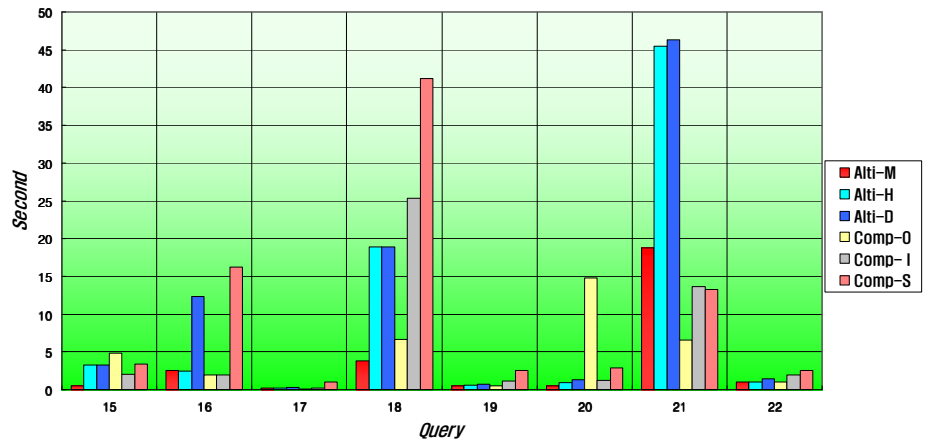
Altibase vs DRDBMS TPC-H Test Result



Altibase vs DRDBMS TPC-H Test Result



Altibase vs DRDBMS TPC-H Test Result



単純なクエリの性能

データベースの基本クエリについてのALTIBASEの性能は以下の通りです。

区分	スループット(TPS)
SELECT	Minimum 6,000～ Maximum 20,000
INSERT	Minimum 5,000～ Maximum 11,000
UPDATE	Minimum 5,000～ Maximum 13,000
DELETE	Minimum 5,000～ Maximum 12,000

測定環境:Sun E3500 4CPU X 400MHz 1G Memory