

高性能OLTPの実現

野崎 正治 森 良哉 滝本 秀明

(株)東芝 府中工場

高性能、高信頼性を特長とするミッドレンジコンピュータDS6500シリーズ上にOLTP環境を実現した。このDS6500/OLTPは、ET1ベンチマークで200TPSの高速処理性能、24時間365日の無停止運用機能と端末1000回線接続能力を有し、マンマシンインタフェースを得意とし多くの流通ソフトを利用できるPC/WS群に対して強力なトランザクション処理パワーを提供している。本報告では、高信頼性、高機能と高性能を両立させて実現するためのシステムアーキテクチャについて報告する。

New Technology for High Performance

Online Transaction Processing System

Masaharu.Nozaki Ryoya.Mori Hideaki.Takimoto

Fuchu Works, Toshiba Corporation

We have developed OLTP environment on midrange computer DS6500 series which is characterized as high-performance and high-reliability. This OLTP environment provides (1)powerful transaction processing ability - 200TPS(Transaction Per Second) in ET1 benchmark test (2) 24hours and 365days availability (3)scalability by means of distributed processing facilities. The paper presents the OLTP system architecture and technologies to achieve above mentioned advantage.

1. 開発の背景と狙い

従来オンラインシステムは航空機座席予約システムなど大規模なシステムに限られ、大型汎用機の独断場で膨大な応用システム（COBOLで数十万から百数十万ステップ）を大量のプログラマを投入して開発していた。

ところが高度情報化社会が急速に進展し適用対象も中小規模システムへ拡大して行きつつある。ここで登場してきたのが高信頼性、高スループット能力と低コストで応用システムを開発できるミッドレンジコンピュータである。

非常に高い信頼性と高速応答性が要求される制御分野で多くの実績のあるミニコンピュータをベースに、UNIX環境と各種の世界標準ソフトウェアを装備し、最大4プロセッサの対称型マルチプロセッサシステムをサポートする基本ソフトウェア（VMP）に次の様な特長をもったオンラインシステム構築のためのソフトウェアを加えた。

- 大規模、高性能：同時1000回線接続とET1ベンチマーク200TPSの処理性能

- 無停止運用

- アプリケーション開発の生産性向上

今回は、高信頼性、高機能と高性能を両立させて実現するために採用したシステムアーキテクチャについて報告する。

2. トランザクション処理とは

図1にトランザクション（以下、Txと略す）処理の流れを示す。端末から投入された入力メッセージは、端末制御プログラムにより、メッセージ編集などが行われた後に、目的の業務処理を行う業務プログラムに渡される。

業務プログラムでは、Txの開始点を宣言し、データベース（DB）の検索・更新、端末へのメッセージ送信処理などの本来の業務処理を実行する。DBへのアクセスは、バッファであるデータバッファを経由して行う。業務プログラムによる効果は、正常に完了するまで他のTxに対して排他され、相互独立性、原子性が保証されている。

業務プログラムの正常完了によって、その効果がアフターイメージジャーナル（AIJ）や出力メッセージジャーナルなどのジャーナルデータとして作成され、ジャーナルライタによってジャーナルファイルに書き込まれる。その後、データライタによってDBの実更新を行い、端末制御プログラムによって端末へ応答、配信メッセージを出力する。

ジャーナル取得前に障害が発生すると、その効果は、業務プログラムによるTx開始点の宣言の前の状態に自動的に戻される（ロールバック）。ロールバックされたTxは入力メッセージのジャーナルを

用いて再試行を行うことができる。

ジャーナル取得後に障害が発生しても、ジャーナルによる復旧が行われ（ロールフォワード）、そのTxは有効となる。

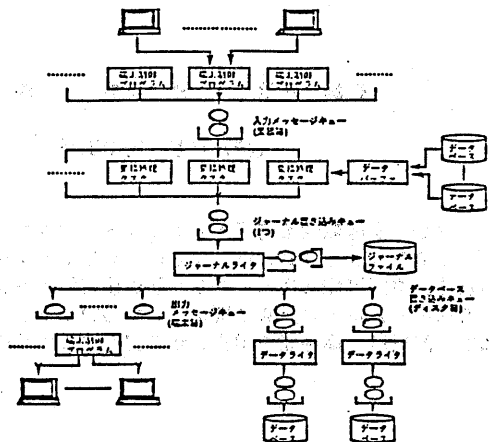


図1 トランザクションの流れ

3. 高性能の実現

以上が、1つのTxについて、時系列的にみた流れである。OLTPシステム内では、これらのTxが複数個、同時に流れる。OLTPの性能は、単位時間当たり、どれくらい多くのTxが流れるかで決まる。しかし、その高速性を実現するために越えなければならない、以下のような性能面でのボトルネックがある。

● ディスクアクセス

ジャーナルファイルへの書き出しとDBの読みだし書き出しのように、メカニカルな動作が伴い時間のかかるディスクアクセスをいかに少なくするか、並行動作させるかがポイントである。

● タスク間の排他競合

排他競合を少なくなるようにしないと、Txやジャーナルライタ、データライタがシーケンシャルに動作してしまい、並列性が得られず性能が出なくなる

● CPUの負荷

プログラムが多数並行して効率よく動作するのに十分なCPU能力が必要である。また、資源の確保、解放や、端末制御のためのCPU負荷が問題になる

● タスクの数

端末台数が数千台になり、投入されるTxが増えて

きた場合に、システムに存在するタスクの数が単純に増えると、それに伴い弊害が出て、性能が出ない。

これらのボトルネックは、処理要求が同時に大量に発生し、OLTPシステムとして信頼性を確保するための処理（ジャーナルを採取、相互排他を行うなど）が必要であることに起因するが、以下の高速化技法を取入れ、これらを解決して、高信頼性と高速性を両立させた。

- 高性能ハードウェアとOSの活用
- ジャーナルの一括書き出し
- DBの実更新の遅延一括書き出し
- DBのエクステント分散配置
- 排他期間の極小化
- システム資源のプール化
- パソコンやワークステーションへの負荷分散
- 1つの端末制御タスクでの複数端末サービス

3. 1 高性能ハードウェアとOSの活用

次節以降で、高速化のために取り入れた技法を述べるが、それらは、ハードウェアとOSをいかに効率良く利用するか工夫がなされている。しかし、それらの工夫をしてより高い性能を実現するためには、ベースとなるハードウェアとOSの性能がポイントとなる。DS6500シリーズとそのOSであるVMPの特徴のうち、以下の点を活用している。

(1) 対称型マルチプロセッサ(SMP)

最大4台の高性能プロセッサによる高いCPU能力をもつ。OSは、SMPをサポートしており、各プロセッサのOS内の処理が極力同時並行動作するように、OS内部の処理のきめ細かい競合排他制御が行われている。OLTPのように、OSの処理の比率が高い場合は、非常に有効である。

(2) 大規模主記憶

最大512MBまでの主記憶を実装できる。大規模主記憶を利用して、データバッファに割り当てる実メモリサイズを大きくすることにより、DBへのアクセスのためのディスクアクセスを低減することができる。

(3) 高速バス能力

DS6500シリーズは高いバス能力を持ち、多数のディスクを接続でき、同時に多くのディスク転送を行うことができる。

(4) 大規模システムに強いOS構造

VMPはシステムの規模が大きくなっても大規模なバッファや多くのプログラムの同時実行などを効率よく処理できるデータ構造と処理方式を採用しており、OLTPの大規模化に対応できる。

3. 2 ジャーナルの一括書き出し

オンラインシステムの実力は、通信能力・CPU能力・ディスク能力の中で最も低いものによって抑えられる。通信、CPUは近年飛躍的に伸びたがディスクはメカニカルな面で性能が抑えられ高負荷時にボトルネックが生じる。このボトルネックを解消するために複数のディスクドライブに分散書き込みを行う、不揮発性半導体記憶装置を1次記憶として用いるなどの方法が考えられるが本システムでは次のような方法でボトルネックを解消した。

オンラインシステムでは一件毎の応答は一般に数秒以内を保証すればよい。この点に注目し各トランザクション(Tx)のジャーナル書き出しを個別に処理せず複数のTxのジャーナルをまとめてジャーナルファイルに書き出しディスクアクセス回数を減らす方法を採用した。

1つのTx毎にジャーナルを1回ディスクへ書き出すとすると、1回のディスクアクセスは25msぐらいかかるので、1秒間に40Tx、すなわち、40TPSが限界である。ここで、例えば、6Tx分のジャーナルを一括して書き出すとすると、6Tx分のジャーナルの書き出しは30msぐらいになり、200TPSを突破できる。これは当然、ジャーナル書き出し以外の性能ボトルネックを解決したとしての場合である。

また、システムダウンリカバリ時にDBの実更新が完了していないジャーナルを判断するためのクリーンポイントの一括書き出しや、システムダウンリカバリ時にジャーナルファイル上の最新のジャーナルを見つけるための最新ジャーナルポイントの一括更新を行っている。これらにより、ロールフォワードすべきAIJが少し増えたり、最新のジャーナルを見つける処理が少し増えたりして、システムダウンリカバリの処理時間が少し増えるが、通常処理において、ジャーナルファイルへのアクセスが大幅に低減できる。

3. 3 実更新の遅延一括書き出し

DBの実更新は、ジャーナルファイルへの書き出しと同様ディスクへの書き出し能力がボトルネックなる。このボトルネックを解消するために遅延一括書き込み方式を使用している。この方式では、データライタが実更新を要求されたときに、直ちに行うのではなく、わざと遅らせて書き込みを行う。

遅延させることにより、複数のTxからの実更新要求がたまり、同一ページや近隣のページへの書き込み要求がたまる確率が高くなる。そして、データライタは、たまっている同一ページに対する複数の書き込み要求をまとめて1回で書き出し、近隣のページをまとめて1回で書き出す。更に、業務プログ

ラムがアクセス中でコミットが完了していないページの中に同一もしくは隣接するページがあれば、その実更新要求が来るまで待ってまとめて書き込みを行い、その間は別のページの更新要求処理を行う。(図2)

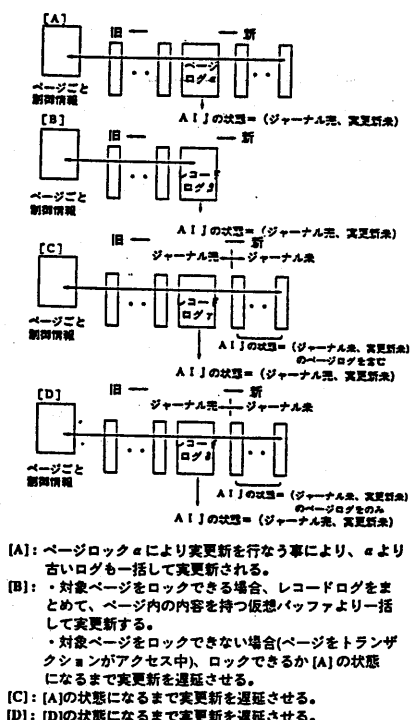


図2 遅延一括書き出し

多くのTxが、共通の比較的小さなDBへアクセスすることがある場合(後述するET1ベンチマークでの、預金係DB、支店DB、履歴DBなど)は、この遅延一括書き出しの効果は非常に大きく、ディスク書き込み回数を大幅に減少させることができる。

3.4 エクステント分散配置

一方、非常に大きなDBで同一、隣接ページの更新が希な場合、遅延一括書き出しによる効果はない。このような場合は1つのDBを複数のディスクドライブに分散配置(エクステント分散配置)し、書き込み処理の並列化によって高速化を計ることができる(図3)。また、DBの領域割付時にエクステント分散配置を容易に行え、チューニングも簡単に行うことができる。

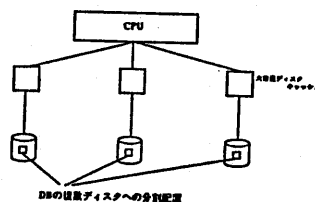


図3 エクステント分散配置

3.5 排他期間の極小化

排他期間の極小化の目的は、各々のトランザクション(Tx)の共通の排他期間を極小化し、性能を上げることである。

例えば、共通の排他期間が25msであるとする、その部分はシーケンシャルに実行されるので、40TPS(1秒間に40Tx)の性能が限界となる。性能を上げるためには、この共通の排他期間をできるだけ短くする必要があるが、以下の方式によりこれを実現した。

(1) ジャーナル書き込み前の排他解除

Tx同士の排他競合を極小化するために、Txのコミットの発行後、ジャーナルをジャーナル書き込みキューに入れた時点で、そのTxが保持しているページ排他をはずす。ジャーナル書き込みの完了前にページ排他をはずすので、共通の排他期間を大幅に短くすることが可能となる。ただし、このジャーナル書き出しの完了前にページ排他をはずすという方式は、障害時の処理が非常に難しく技術的な難解さが伴う。

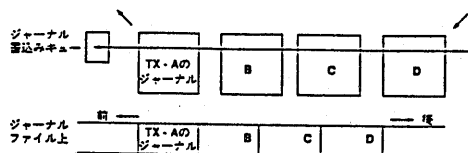


図4 ジャーナリング順序制御

各Txに対して、

- ジャーナルをジャーナル書き込みキューに入れる、
- ページ排他をはずす、
- ジャーナル書き出しを行う、
- Txの完了を外界に伝達する(DBの実

更新の許可／端末への完了通知／他プログラムへの通知のキックを行う）、

の順序を保証し、

更に、図4のように、ジャーナル書き込みキュー上のTxのジャーナルの順序とジャーナルファイル上のTxのジャーナルの順序を同一にすることにより、(b)と(c)の間でのシステム障害発生時も矛盾が起きないようにしている。ジャーナルファイルにジャーナルが書き込まれたTxが完了しており、書き込まれていないTxの外界への効果は一切ない。

(2) DB実更新でのページログ利用

TxのDBの使用頻度の高いページへのアクセスと、データライタのそのページの実更新との間でのページ排他の競合の確率は高い。データライタがそのページを排他占有すると、Txが排他待ちになり、高スループットが得られない。このため、同一ページ内でレコードログ(AIJ)は原則としてレコードログを採取する)が多く採取されている場合は、一定回数ごとにページログを採取し、データライタはページログを利用して、ページ排他の競合を起こさないようにして実更新を行っている。

また、データライタは、実更新を予定外に滞留させないために(計画的に遅延はしているが)、使用頻度の低いページの実更新を行う場合も、ページ排他待ちにならないように制御している。

(3) データバッファ制御

データバッファはDBのメモリキャッシュでありLRU方式で管理されているが、Tx途中のデータバッファ上のページは、データバッファからあふれる場合でもページングファイルに書き出され、Txが完了してアフターイメージジャーナル(AIJ)がジャーナルファイルに書き出されるまでは、元のDBに書き込むことはない。Txがロールバックされる場合は、図5のように、データバッファ上の該当ページを無効にして、DB上の該当ページの内容に対して、そのページが実更新されてからロールバックされるTxまでのAIJ群(これらは該当ページに対してメモリ上でキューイングされている)をかけることにより該当ページをロールバックされるTxの直前の状態にデータバッファ上に復元する。以上により、DBのジャーナルはAIJのみを採取し、ビッフォイメージジャーナル(BIJ)を採取不要にでき、BIJを採取する負荷が低減される。また、データバッファの領域管理は固定サイズの基本ページで管理し、可変サイズの要求には離散した複数の基本ページを割り付け、フラグメンテーションを防止している。

以上の通りデータバッファは要であり、全てのT

xからアクセスされ、データバッファ管理情報の排他競合がおりやすい。このため、データバッファを複数にして排他競合を低減している。しかし、単に、複数化すると、データバッファの利用状況に偏りが発生しうるので、ハッシングした後に使用するデータバッファを決めている。

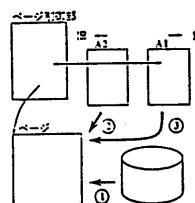


図5 ロールバック

(4) デッドロック制御

Tx間のページ排他でのデッドロックは発生した場合、早期にデッドロックを解消してTxの流れを停滞しないようにすべきである。ただし、Txがページ排他待ちになった場合もデッドロックになっている可能性は低く、直ちにデッドロックチェックを徹底的に行うのも無駄であることも多い。そのためTxがページ排他待ちになった場合、直ちにではなく一定時間後にチェックを行う。チェックの深さも限定し、チェック済みのパスは2度チェックしないようにし、チェックによる過大なオーバーヘッドが生じないようにしている。この方式で大体のデッドロックを検出できるが、残りのデッドロックは、Txのラップスのタイムアウトで検出する。

3. 6 資源のプールの化

トランザクション(Tx)は、実行時メッセージ領域、タスク、DBの資源が必要である。使用頻度の高い資源にたいしては、事前に資源をプールし、Tx実行時の資源取得／解放にともなう処理オーバーヘッドを低減した。

(1) メッセージ領域のプールの化

Txは、1件の処理で少なくとも入力／出力用の2つのメッセージ領域の獲得、解放を行う。この獲得、解放を高速に行いメモリ領域の断片化を防ぐためにメッセージサイズ毎にメモリブロックを事前にプールし最適なプールからブロックを取得している。プール自身も領域が不足すると数十ブロック単位で自動拡張、自動縮小を行う。

(2) 業務プログラムのプリロード

非常に頻繁に使用する業務の業務プログラムは、事前にディスクからロードし、メモリを割り付けてタスクとして起動しておきTxが発生すると即座に処理が開始できる。1つのタスクではTxが発生すると待ちが即座に解除され1件の業務を処理した後、次のTxの発生待ち状態になる。同一の業務に複数のタスクを割り付けて多重処理を行い処理能力を向上させることができる。

(3) DBのプリロード

TxがDBをアクセスするためにはまずDBのオープンを行わねばならない。しかしUNIXのような階層構造のファイルシステムの場合、ディレクトリファイルを開くため最低2回以上のディスクアクセスが必要となる。そのためオンラインで使用するDBへのアクセス情報はDBをオンライン状態にした時1度ディスクからメモリにロードしておきメモリ上の情報を用いてオープン処理を行う。

3.7 PC/WSへの負荷分散

PC/WS (パソコン/ワークステーション) 上の統合メニューシステム (Imenu) と連携して

- 業務選択のためのメニュー処理
- メッセージ編集処理
- 入力データのチェック処理

などのマンマシンインタフェース部の端末であるPC/WS側での分散処理環境を提供している。

(図6)

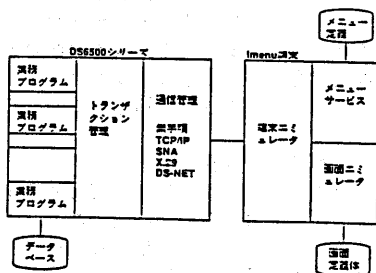


図6 PC/WSとの分散処理

これによってホストの負荷が分散されるとともに、PC/WSの持つ優れたオペレーティングフェースが利用でき、かつTxの投入と結果の表示の時だけホストとPC/WS間でメッセージが送信されるため通信コストを削減することができる。

また、PC/WS側の画面定義体は、ホスト側の画面定義体とのバージョンのチェックが行われ、ホ

スト側がバージョンアップされた場合は、ホストから画面定義体のオンラインダウンロードが可能である。

3.8 1端末制御タスクで複数端末サービス

OLTPシステム側から見ると、1つの端末からの処理要求の間隔は長い。これに着目し、1つの端末制御タスクで、複数の端末をサービスするようにした。これにより、システム内に存在するタスクの数を少なくでき、必要となるメモリサイズを低減でき、タスクのスケジューリングなどに伴うオーバーヘッドを低減できる。

3.9 ET1

以上のような高速化技法を取り入れた結果、OLTPの世界で広く評価に用いられているET1ベンチマーク (銀行の口座取引をモデル化した業務) において、プロセッサ4台構成、ディスク20ドライブの構成で、200TPS (1秒間に200件の処理性能) を実現した。200TPSという数字は、端末のオペレータが20秒に1件の預金処理を投入したとして $20 \times 200 = 4000$ 人のオペレータが操作している状況である。

ET1ベンチマークは、DEBIT/CREDITベンチマークともいい、下の図7に示すようなプログラムである。小さなDBである預金係DBと支店DB、またシーケンシャルにアクセスされる履歴DBに対しては、実更新の遅延一括書き出しが効果的であり、大きなDBである口座DBに対しては、エクステントの分散配置が有効である。また、端末側に関してはパソコン、ワークステーションへの負荷分散と、1端末制御タスクでの複数端末サービスの効果的である。さらに、ジャーナルの一括書き出し、排他期間の極小化、資源のプールの化は全体的に有効である。

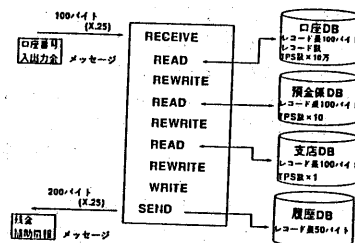


図7 ET1ベンチマーク

4. 無停止運用を目指して

公共システムではシステムの停止は社会的に重大な影響を及ぼす。また一般システムでも業務の国際化によって24時間無停止システムへのニーズが非常に高くなってきた。24時間365日無停止の運用を行うために様々なフォールトトレラント機能、支援機能を備えている。

(1) 高信頼化ディスク

ディスクの二重化機能と、電源とコントローラからディスクへの経路を二重化し動的な経路選択機能を提供している。片系障害の場合無停止で交換、復旧を行うことができる。

(2) プロセッサ縮退運転

最大4台までのマルチプロセッサで構成でき、プロセッサが故障すれば自動的に切り離されて縮退運転を行う。

(3) ホットスタンバイ機能

計算機システムを二重化し、運用系と待機系にわけ待機系は運用系を監視し障害が発生すると自動的に運用系に切り替わりダウン時間を短縮することができる。(図8)

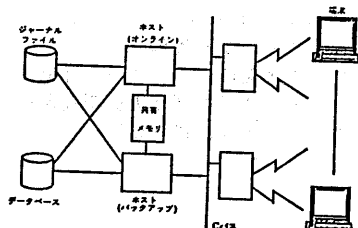


図8 ホットスタンバイ構成

(4) オンライン構成制御

オンライン運転中に端末、DB、業務プログラムの追加、削除、変更などの構成制御ができる。

(5) 運転自動化支援機能

システムの障害原因にはオペレータの操作ミス、オペレータの不在などもある。このような運用上のミスを防止するために業務プログラムがシステム管理者になりかわって運用ができる運転自動化支援機能を提供している。

(6) オンラインセーブ

ジャーナリングとトータルセーブを組み合わせることにより、オンラインセーブを実現している。DB破壊時は、トータルセーブとトータルセーブ開始

以降のジャーナルとをマージしてDBの復旧を行う(図9)

また、トータルセーブに加えて、シンスセーブ、インクリメンタルセーブも併用できる。これらではセーブすべき更新されたブロックを高速に見つけるための管理情報を持ち、セーブ時間の短縮を実現している。特に、インクリメンタルセーブでは、DBとDB内の各ブロックにセーブ履歴カウンタをもちそれらと比較することにより、オンラインでセーブできるようにした。

CdをDB全体のCp(i)をDBのページiのセーブ履歴カウンタとしたとき以下のような方式を採用する。

ページi更新時

$$Cp(i) \leftarrow Cd$$

インクリメンタルセーブ開始時

$$Cd \leftarrow Cd + 1$$

ページiのセーブ時

$$Cp(i) \geq Cd - 1 \text{ ならばセーブ}$$

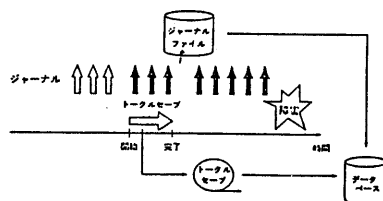


図9 オンラインセーブ

5. 応用システム生産性の向上

(1) 応用システム構築者から障害回復処理を解放

即時処理を使命とするオンラインシステムではデータは極めて重要で様々な障害に対して破壊されてはならない。そのために以下の障害復旧処理を提供している。

● システムリカバリ

電源断などのシステム障害からDB、メッセージ資源の運転状態の復旧を行う。処理の途中であったTxはキャンセルされ入力メッセージのジャーナルから自動的に再試行を行い復旧する。

● メッセージリカバリ

通信回線、端末に障害が発生した場合、出力未完了状態のメッセージはメッセージリカバリログファイルに退避され自端末及びオペレータコンソール端末に復元することができる。

● キャンセルリカバリ

業務プログラムに障害が発生するとDBへの更新、端末への出力メッセージはキャンセルされTX開始前の状態に戻される。

● 媒体リカバリ

ディスクの全系障害、オペレータミス、業務プログラムバグによってDBが破壊されるとDBのセーブデータとジャーナルデータを用いて任意の時点の状態にDBを復旧することができる。

(2) 統合オンライン業務プログラム開発支援

オンラインの業務プログラムの設計、製造から総合テストまでをサポートする統合オンライン業務プログラム開発支援ツールを提供している。図10に支援ツール体系を示す。

これらのツール群は次の様な特長を持っている。

- オンライン業務指向専門にカスタマイズ
- メニューによる操作
- プログラムデータ辞書による成果物の統合管理
- 稼働中システム上での開発

6. あとがき

オンラインシステムは計算機総合力の結晶で、DS6500のOLTP環境はハードウェア、ソフトウェアとも最先端技術を用いて高性能、高速処理を提供すると同時に高信頼、高品質のシステムを提供している。今後は、高速処理を追求すると共に以下のような機能を実現する計画である。

(1) PC/WSとの垂直分散処理、DS6500間での水平分散処理により端末数千台、数万台規模の世界ワイドなオンラインシステムの構築を可能にする

(2) DB、通信機能について世界標準機能をOLTP環境の世界に取り込む

(注1) UNIXはAT&Tが開発しライセンスしているオペレーティングシステムです。

文献

(1) 森、野崎、滝本 他：ミニコンピュータでのジャーナルリカバリ高速化手法、情報第41回全大(1990)

- (1) -基本方針-
- (2) -高速バッファ管理-
- (3) -索引順編成ファイルでの高速化手法-
- (4) -ジャーナリングと実更新高速化手法-
- (5) -リカバリ高速化手法-

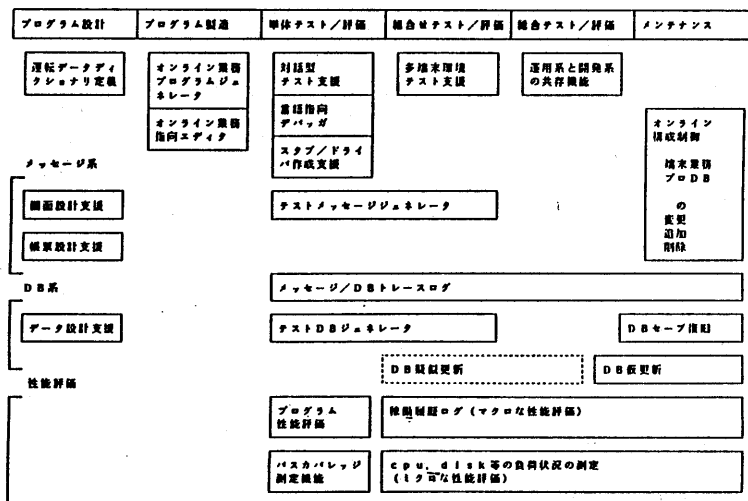


図10

統合プログラム
開発環境