ＵＮＩＸ上ＲＤＢアプリケーション制御方式の検討

3Ｒ－5

加藤謹詞　川手寛　長岡満夫

ＮＴＴソフトウェア研究所

１．はじめに

サーバ型ＷＳは、ＭＩＰＳ値の飛躍的な向上、ＯＬＴＰ向けリレーショナルデータベース（以下ＲＤＢと略す）管理システム技術の進展等により、ＲＤＢを利用したトランザクション型業務システムへの適用が進みつつある。

トランザクション型業務システムを支援するクライアント・サーバ形態での定形型ＲＤＢアプリケーションは、端末とサーバマシンとの分散協調・連携型の方式をとり次の特徴がある。

(a)サーバに接続される端末数が多く（数百～数千台）、通常運用時間に沿ってこれらの端末からサービス要求が発生する。

(b)個々のＲＤＢアクセス処理は予め予測可能な時間（短時間）で終了する様に設計される。

しかし、ＵＮＩＸ－ＷＳ上で上記特徴を持つトランザクション型サービスを構築しようとすると、以下の問題がある。

①ＵＮＩＸ上では、通常、ｆｏｒｋ（），ｅｘｅｃ（）システムコールにより、１端末に対し１ＡＰ(Application Program)プロセスが張り付く方式（いわゆるＴＳＳ方式）が一般的である。更に、ＡＰは、ＲＤＢアクセスのためのライブラリのリンクに伴うコードの増加とともに、シャドウプロセスと呼ばれるＲＤＢアクセス用のプロセスが１ＡＰに対し張り付く。従って、トランザクション型サービスで使用される端末起呼対応にＡＰプロセスを起動する本方式では、予測不可能なメモリ消費量となり、事実上システム設計不可能である。
また、ＣＰＵ／メモリ資源の利用の点に関しても、端末数分のプロセスがメモリ上に常駐すると、プロセススイッチ、メモリスワップのオーバヘッド等により著しく性能が劣化し性能予測困難である。

②上記①に対し、同時接続数対応にトランザクション要求発生時に、プロセスを起動し、多数のプロセスがメモリ上に常駐するのを抑える方式が取られる場合があるが、ｆｏｒｋ（），ｅｘｅｃ（）シス　テムコールでは、性能が悪く、トランザクションの単位での使用には問題がある。

そこで、本稿では、ＵＮＩＸ－ＷＳをトランザクション型業務支援システムに適用する際の上記問題点に対し、リアルタイム型のＤＢ／ＤＣ系プロセス（以下デーモンと称す）を導入し、トランザクション処理を実行するサービスプロセスを制御する方式（以降ＲＴＳ方式と呼ぶ）を提案し、その有効性の評価する。

以下では、ＵＮＩＸ上での一般的なプロセス構成と本稿で提案する方式の概要を説明し、本方式の有効性を評価するための評価実験について述べる。

２．ＵＮＩＸ上でのＴＳＳ方式プロセス構成

図１に、telenet,ftp等のサービスに一般的なＵＮＩＸ上でのプロセス構成を示し、以下に処理の概要を説明する。（本方式を以後ＴＳＳ方式と称す。）

①サーバ上のデーモンは、特定の通信ポートを監視し、端末からの電文を待つ。

②端末からの通信パスが確立され、サービス要求電文をデーモンが受信するとデーモンは、ｆｏｒｋ（），ｅｘｅｃ（）システムコールによりサービスプロセス（ＲＤＢアクセスを実行し処理結果を端末へ返却するプロセス）をロードし実行する。この際、通信パスの引継を行う。

③以降、端末からの電文は、直接、サービスプロセスへ送られ、サービスプロセスは、サービス処理実行の後、結果を端末へ送信する。

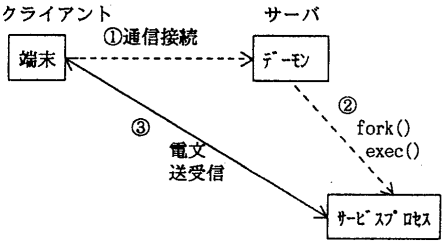

図１　ＵＮＩＸ上での一般的なプロセス構成（ＴＳＳ方式）

３．ＵＮＩＸ上でのＲＴＳ方式プロセス構成

提案するＲＴＳ方式プロセス構成案を図２に示し、以下に概要を説明する。

①サービスプロセスは、デーモン起動時に起動され、必要な資源確保、
各種初期化処理、デーモンとサービスプロセス間の通信パスの確立等を実行する。

②デーモンは、端末からのサービス要求電文を待つ。

③端末は、通信パスの確立のみ事前に行い、サービス処理必要時にのみサービス要求電文を送信する。ここでの通信パスの確立では、デーモンに通信パスが張られるだけである。

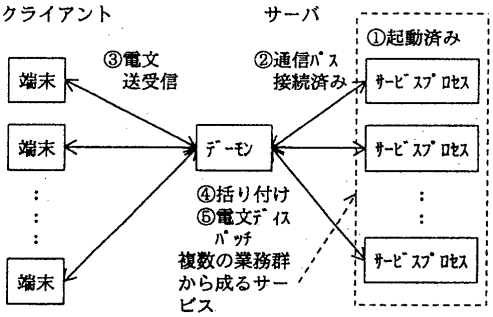

図２　提案するプロセス構成（ＲＴＳ方式）

④デーモンは、端末からのサービス要求により、空きサービスプロセスを端末と対応させる。（括り付けと称す。）

⑤括り付け完了後、デーモンは、対応する端末とサービスプロセス間の電文の授受を行う。

なお、各デーモン対応に『サービス』という概念を設けることによって、システム全体の業務を制御可能であり、複数のデーモン時には、『サービス』の閉塞等の運用が可能となる。

４．ＲＴＳ方式プロセス構成の利点

上記ＲＴＳ方式プロセス構成によって、１．で述べた問題点を解決することができる。即ち、

①デーモンに接続される端末は、トランザクションを実行するサービスプロセスとは直接対応せず、トランザクション要求発生時にのみ対応ずけられるため、端末の台数とサービスプロセスは、Ｍ：Ｎ（ここで、Ｍ>>Ｎ）関係となるため、サービスプロセスを実行可能な個数まで減らし、スループットを最大にすることができる。

②トランザクション要求発生時、サービスプロセスは、既に、ｆｏｒｋ（），ｅｘｅｃ（）、及び、各種資源確保，初期化処理済みであり、括り付け処理終了後、直ちに、ＤＢアクセス，デ

Control methods for RDB application on UNIX systems.
Kinji Kato, Hiroshi Kawate, Mitsuo Nagaoka
NTT Software Laboratories

ータ処理を実行するため高速なレスポンスが期待できる。

次に、ＲＴＳ／ＴＳＳ方式での各処理時間の実測比較法を示す。

５．測定方法

ＴＳＳ，ＲＴＳ方式でのトランザクション処理時間を測定する。
表１にハードウェア構成、図３に測定環境構成を示す。
測定シーケンスは図４に示す。

表１　ハードウェア構成

サーバマシン	通信ﾛｸﾞ取得ﾏｼﾝ	端　末	通信回線
HP9000 827S	SUN SS2	PC H98 model90	Ethernet (10Mbps)
ﾒﾓﾘ量(64MB) ﾃﾞｨｽｸ容量 (1.3GB)		LANﾎﾞｰﾄﾞ (Ungermann -Bass)	

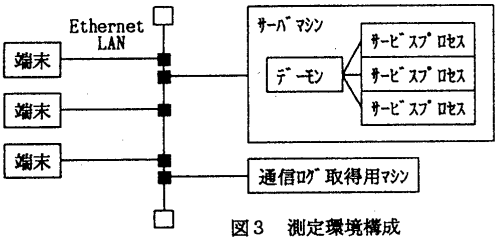

図３　測定環境構成

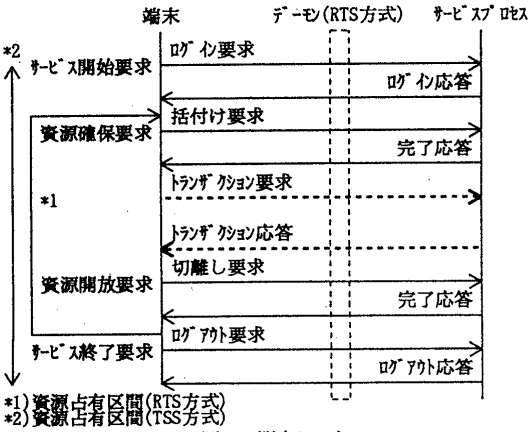

図４　測定シーケンス

以下に測定内容及び有意差比較内容を説明する。

（１）トランザクション処理時間の実測方法

サービスプロセス数個を起動し、端末数台より、同時にサービス開始要求からサービス終了要求までを発行し、サーバ側のレスポンスタイムを通信ログ取得マシンで取得し、算出する。ここでＴＳＳ、ＲＴＳ毎に実測し、トランザクション要求に関してはＴＳＳとＲＴＳ方式の有意差を明確にするため、実データ転送（２ＫＢ／１回の転送を５００回繰り返し）の送信／受信に関して実測を行っている。

（２）ＲＴＳ方式とＴＳＳ方式での有意差比較式

比較に関しては、サーバ資源の保留時間を対象とし以下とする。
保留時間(TSS)=サービス要求時間＋資源要求時間＋ＲＤＢアクセス時間＋業務処理時間＋データ転送時間
保留時間(RTS)=資源要求時間＋ＲＤＢアクセス時間＋業務処理時間＋データ転送時間
但し同一業務実施として、以下を仮定する。
〔1〕データ転送は通常トランザクションで２KB／２回(送信／受信）程度で済む。
〔2〕ＲＤＢアクセスと業務処理時間はＴＳＳ／ＲＴＳ同等とする。実際にはﾃﾞｰﾀ転送に関し、有意差のｵｰﾊﾞﾍｯﾄﾞが含まれる。

６．測定結果

表２、３に処理時間の測定結果を示す。表４には、表２の各処理時間の主な処理内容を示す。

表２　処理時間（サービス開始から終了までｰﾄﾗﾝｻﾞｸｼｮﾝ除く）

方式	ﾛｸﾞｲﾝ	括り付け処理	切り離し処理	ﾛｸﾞｱｳﾄ	合計	備　考
TSS	1	1	1	1	1	相対比
RTS	0.04	0.71	0.90	0.004	0.39	で示す

（注１処理時間は、端末数で全体処理時間を平均してある。

表３　トランザクション系での送受信時間比較

方式	ＰＣ→サーバ送信 端末含む	ｻｰﾊﾞ内	サーバ→ＰＣ送信 端末含む	ｻｰﾊﾞ内	備　考
TSS	1	1	1	1	相対比
RTS	1.054	1.377	1.068	1.397	で示す

（注２）２ＫＢ／５００回の転送に関し３回の実測結果の平均

表４　ＴＳＳ，ＲＴＳにおける主な処理内容

処理項目	ＴＳＳタイプ	ＲＴＳタイプ
ログイン	①ﾛｸﾞｲﾝﾕｰｻﾞ名ﾁｪｯｸ ②fork(),exec()によりﾌﾟﾛｾｽ起動 ③資源確保、初期化	①ﾛｸﾞｲﾝﾕｰｻﾞ名ﾁｪｯｸ
括り付け処理	①RDBｵｰﾌﾟﾝ ②RDBｼｬﾄﾞｳﾌﾟﾛｾｽ起動 ③RDB関連初期化	①空きｻｰﾋﾞｽﾌﾟﾛｾｽ括り付け ②RDB関連初期化
切り離し処理	①RDBｸﾛｰｽﾞ	①端末ｰｻｰﾋﾞｽﾌﾟﾛｾｽ対応ﾃｰﾌﾞﾙのｸﾘｱ
ログアウト	①各種資源解放 ②ﾛｸﾞｱｳﾄ電文送信	①ﾛｸﾞｱｳﾄ電文送信

７．測定結果の考察

(1)表２の結果は、ＲＴＳ方式の場合、ＴＳＳ方式に比較して、サービス要求／資源要求処理等の事前処理系が、約４０％で済み、ＲＴＳ方式がトランザクション処理で有効である。
(2)また、表３での結果、データ転送に関するデーモンのオーバヘッドは、サーバ系でのみで約４０％程度であるが、通常のトランザクションレベルでは、データ転送時間が極小(14～18msec)であり、レスポンスへの影響度合いは極めて少ない。
(3)有意差としてサーバ保留時間に関しては、表２、表３、及び実測結果より、別途測定済ＴＰＣ－Ａのベンチマーク結果を考慮すると、１．０≧Trst／Tssで、Trts／Tss＝０．４３程度になり、最適解としてのサーバプロセスの保留時間はＴＳＳ方式に比較し、約半分程度削減可能である。

８．まとめ

本稿では、ＵＮＩＸ－ＷＳをトランザクション型業務処理システムに適用するために、トランザクション処理を効率的に処理可能なプロセス構成方式を提案し、その有効性を実測結果によって評価及び考察した。評価及び考察の結果、
①トランザクション型業務処理システムで必要とする多数の端末が効率的に接続及びサーバ上サービス享受可能な方式であること、
②ＵＮＩＸ上で、本方式を用いた場合、一般的にトランザクション処理方式であるＴＳＳ方式に比較して、単一のトランザクション処理時間がＴＳＳ方式での相対比で約４０％で実行可能であること、等を示すことができ、トランザクション型業務処理システムへの有効性を確認した。

今後は、実際の業務システムへ本方式を適用し、大規模システムでの実用性等を検証していく予定である。