

オープンシステムにおける性能評価手法

5 D-4

田中 淳裕 堀川 隆 小林 和朝 紀一 誠

E-mail {omochi,horikawa,koba,kino}@sbl.cl.nec.co.jp

NEC C&C 研究所

1 はじめに

各種インターフェイスの標準化が進み異機種混合型のシステムが盛んに構築されるようなオープン時代では、入出力に関する仕様がわかっても、性能情報が決定的に不足してしまうため、機能面では仕様を満足しても、性能的には満足のかないシステムができてしまうことがある。そのための解決策として、システム性能を事前に見積もるための手法であるシングルプロファイル法を、我々は提案している [1].

本稿では、この手法を DBMS の性能評価に適した形にし、UNIX 上の oracle システムの性能を実際に評価した結果を述べ、その有効性を報告する。

2 シングルプロファイル法の適用

シングルプロファイル法の特徴はトランザクションの種類ごとに、無競合状態における計算機資源の正確な使用時間（これをシングルプロファイルと呼ぶ）を測定し、競合が起きた時に生じる待ち時間を待ち行列網理論に基づき計算するというものである。

ところが、DBMS の性能予測をこの手法を用いて行おうとすると、1つの処理が複数のプロセスによって実行されるために、1つのシングルプロファイルが得られるのではなく、プロセスごとのシングルプロファイルが多数得られてしまう。

もちろん複数個のシングルプロファイルを用いた予測も可能ではあるが、プロセスの種類が増えるに連れて予測のための計算量が増加したり、また、複数プロセス間の同期処理をモデル化する必要等が生じてしまい、考慮すべき待ち行列モデルが複雑なものとなってしまふという欠点がある。

そこで我々は、プロセスごとに得られる複数のシングルプロファイルを単純な形にまとめあげることにより、システム提案段階でのおおまかな性能見積もりに利用できるような手法を提案する。本稿では特に、UNIX 上の oracle システムを対象として、提案する手法の有効性を、実際の測定結果と比較しながら述べる。

Oracle では、次のようなプロセスが協調して作業を行うことにより、データベースの検索・更新など

のトランザクションを行う (図 1 参照).

DB1, DB2, DB3: データベースの更新・追加作業を行う。

DBWR: 上記 DB[123] を管理する (以降 DB と記す)。

LGWR: トランザクションの履歴をディスクに書き込む (以降 LG と記す)。

SHADOW: クライアントプロセスからのトランザクション処理要求を受け付ける (以降 SH と記す)。

ORASRV: クライアントからの接続要求を受け付け、クライアントプロセスと 1対1 に対応するように、SHADOW を生成する。

SMON, PMON: システムやプロセスの動作状況をモニタする。

トランザクション処理においては、ORASRV, SMON, PMON プロセスの負荷はほとんど無視できる程度であったので、以降の議論ではこの 2つのプロセスは考慮にいれない。

クライアントが直接やりとりを行うのは SH プロセスであり、また、TinyTOPAZ の測定によると、SH と LG はプロセス間通信を用いて同期的に動作している。これらの事実から、ユーザからみたシステムのレスポンスタイムは SH と LG の動作時間であると考えられる。

一方、データベースへの書き込み作業を行うプロセスは DB と DB[123] であり、DB と各 DB[123] がプロセス間通信による同期をとって動作している。

上記複数プロセスをまとめあげて、単純なモデルにするために以下のような点に着目して改良を行う。

- プロセス間通信を用いて同期的に動くプロセスをひとまとめにして 1つの処理とする。これにより、SH と LG が、また、DB と DB[123] をまとめたものが、それぞれ 1つの処理になる。(以降、処理名をそれぞれ SL, D と記す。)

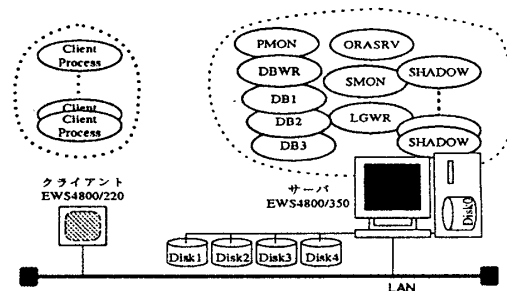


図 1: システム構成

このような改良をほどこすことにより、複雑なモデルを構築することなく、古くからその性質が明らかにされている BCMP 型待ち行列網 [2] などのようなモデルによる予測が可能となる。

3 予測例

図 1 に示すようなシステムで、TPC-A[3] のトランザクション処理を行った。

このトランザクション処理では、サーバにある口座 (Account)、窓口 (Teller)、支店 (Branch)、履歴 (History) の 4 種類のデータベースを、クライアントがネットワークを経由してアクセスし、1) update A: 口座の残高に、取り引き額を加える。2) select A: 口座残高を得る。3) insert H: 履歴 DB に取り引き履歴を格納。4) update T: 窓口残高に取り引き額を加える。5) update B: 支店残高に取り引き額を加える。6) commit: トランザクションをコミット、という一連の処理を行う。

ユーザの思考時間を 500ms として上記トランザクション処理を 10000 回行い、性能測定ツール TinyTOPAZ で測定したプロセスごとの計算機資源使用時間の平均を、表 1 に示す。なお、数値は SH プロセスの資源使用時間のトータルを 1.0 とした時の相対値である。

表 1: TinyTOPAZ による測定結果

	SH	LG	DB	DB1	DB2	DB3
CPU	35.8	1.75	0.519	0.358	0.265	0.228
DISK0	0.03	0	0	0	0	0
DISK1	0	10.7	0	0	0	0
DISK2	15.8	0	0	0.116	0.126	0.0914
DISK3	9.67	0	0	2.16	1.62	1.19
DISK4	10.0	0	0	2.22	1.69	1.27
CLIENT	28.7	0	0	0	0	0

$\times 10^{-2}$

表 2: ひとまとめにしたシングルプロファイル

CPU	DISK0	DISK1	DISK2
38.9	0.03	10.7	16.1
DISK3	DISK4	CLIENT	
14.6	15.2	28.7	

$\times 10^{-2}$

予測モデル 1: まずは、プロセスごとに得られたシングルプロファイルを、各資源ごとに足し合わせてひとまとめにしたシングルプロファイル (表 2) で、システム性能を予測する。この予測では、処理 D を実行するための時間もユーザのレスポンスタイムに含まれてしまう。そのため実際よりも悪い方向へ予測していることになる。

予測モデル 2: 次に、ユーザが実際に観測するレスポンスタイムを計算することを考える。これは、表 2 を用いた予測結果を、表 1 で示されるプロセスごとの資源使用時間に基づいて比例配分すればよい。この際、SH と LG プロセスとが同期関係にあるこ

とに注意する必要がある。例えば CPU の滞在時間 (待ちを含む) が α であれば、ユーザのレスポンスタイムに関係のある部分は SH と LG プロセスの CPU 滞在時間を足し合わせた分で、 $\alpha \times \frac{35.8+1.75}{38.9}$ により得られる。各資源に関してこれと同様のことを行えばレスポンスタイムが得られる。

以上のような 2 つの考え方にに基づき、性能予測ツール QM-Open を用いてサーバのレスポンスタイムを予測した結果と、実測との比較を図 2 に示す。

予測結果は 2 つともシステム提案段階の見積もりを使用するには十分な予測精度が得られている。今回の実験では、予測モデル 1 と 2 であまり大きな差が現れなかった。これは、データベースの更新処理がほとんどなく、ほとんどが検索処理であったためと推測される。

実際の利用に際しては、トランザクションが、検索処理中心なのか更新処理中心なのかを見極めた上で利用する必要がある。

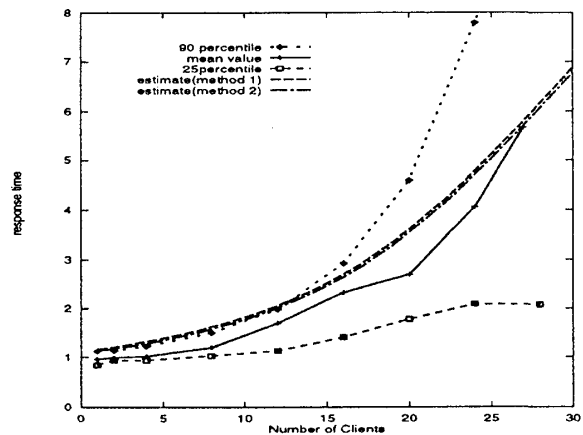


図 2: 予測と実測とのレスポンスタイムの比較

4 おわりに

我々が提案しているシングルプロファイル法に関して、特に UNIX 上の oracle サーバの性能評価が簡単に行えるような改良を行った。これにより、複雑な待ち行列網モデルを用いずとも、システム性能を予測することができることを、実測結果と比較することにより確認した。

参考文献

- [1] 堀川 隆, 紀 一誠: DBMS 動作特性の測定・解析手法, 情報研報, 94-ARC-104-4/94-OS-62-4 (1994).
- [2] Baskett, F., Chandy, K.M., Muntz, R.R. and Palacios, J.: Open, Closed, and Mixed Networks with Different Class of Customers, J. ACM, Vol. 22, No. 2, pp. 248-260 (1975).
- [3] Jim Gray, ed.: The Benchmark Handbook for database and transaction processing systems, Morgan Kaufmann Publishers, Inc., 1991.