

性能シミュレータを用いた Oracle 10g の 性能予測

西岡大祐[†] 川崎正人[†] 瀬賀聡[†] 藤井俊勝^{††} 衛藤重豪^{††}

IT システム構築の際にデータベースの性能予測を実現可能にすべく、Oracle データベースシングルインスタンスを対象に、データベースのベンチマークである DBT-1 を用いてその性能を測定し、性能見積もりモデルを作成した。本報告では、SunFire 280R(1, 2 CPU)上の Oracle 10g をインストールし、1 CPU 時に DBT-1 で定義された 14 種類のトランザクションの性能を個別に測定してモデル化した。この性能モデルを用いて、DBT-1 の Shopping Mix ベンチマークを 1 CPU 時、2 CPU 時でシミュレーションと実施を行い、性能モデルのスケールアップ時における有効性を確認した。

Prediction of Performance of Oracle 10g by Performance Simulator

DAISUKE NISHIOKA[†] MASATO KAWASAKI[†] SATOSHI SEGA[†]
TOSHIKATSU FUJII^{††} SHIGEHIDE ETOH^{††}

The performance was measured by using DBT-1 that was the bench mark of the data base for Oracle data base single instance to enable the achievement of the performance forecast of the data base in the IT system construction, and the performance estimate model was made. In this report, Oracle 10g of SunFire 280R(1, 2 CPU) was installed, and 1 The performance of 14 kinds of transactions that had been defined by DBT-1 at CPU was modeled individually measuring it. This performance model was used, and the Shopping Mix bench mark of DBT-1 was 1, 2 at CPU The simulation and execution were done at the CPU time, and the effectiveness of the performance model at the scale-up was confirmed.

1. はじめに

1.1 目的

近年の IT 環境では、オープン化の流れの中でさまざまな技術が導入されている。また、技術のライフサイクルも短縮化されている。このような状況下で、IT 資源の拡張性と可用性を高めるためにはクラスタリングシステムが有効である。

クラスタリングシステムとは、サーバー複数台でひとつのシステムを構成することであり、このときの各サーバーをノードと呼ぶ。性能が足りないときにノードを追加したり、ひとつのノードがダウンしたりしても他のノードで処理が継続できる。その反面、複数のノードがデータを共有するため、競合するノードが増えれば増えるほどディスク I/O にボトルネックが生じやすくなる。

IT システムの中でも特に重要であるデータベースにクラスタリング手法を導入した代表的な製品として、米国 Oracle 社の Oracle Real Application Clusters (Oracle RAC)^[1] がある。データ競合に対しては、Oracle RAC

では、データの同期作業をバッファ転送で対応する「キャッシュフュージョン」という技術が採用されている。そのため、Oracle 内部では負荷分散やデータの同期など複雑な処理が行われており、Oracle システムの構築時に性能単価の積み上げや、過去の事例からの類推といった経験則だけで予測することは困難であるという問題があった。そのため、解析的な手法や性能シミュレーションによる性能予測技術が期待されている。

本報告では、後者の性能シミュレーションによるアプローチによって、Oracle システムの性能を予測し、システム構築の生産性を向上させることを目的とする。

1.2 測定対象

データベースとして広く使われている Oracle 10g を対象とし、第一段階として、シングルインスタンスを対象として性能モデルを作成する。シングルインスタンスモデルの評価を行った後に、第二段階として Oracle RAC システムのモデル化に取り組むこととした。本報告では、第一段階の結果を報告する。

データベースの性能測定方法としては TPC^[2] (Transaction Processing Performance Council)によって策定されたベンチマークを用いるのが標準的であるが、本報告では、Web システム向きに策定されたベンチマークである、TPC-W の簡易版である DBT-1^[3]を対象とした。その理由として、

[†] 株式会社ケイ・ジー・ティー
KGT Inc.

^{††} 株式会社ベンチャーシステムズ
VENTURY SYSTEMS CO., LTD.

- (1)環境が構築しやすい、
- (2)ソースで提供されているため、測定する目的に合わせて修正が可能、
- (3)無償で利用できる

が挙げられる。

2. Oracle 性能モデルの作成方針

Oracle の RAC 構成は複雑なため、Oracle RAC モデル作成を二つの段階に分けて実施する。本報告では、前述のとおり第一段階について説明する。

(1) 第一段階

Oracle のシングルインスタンス (RAC 構成でない) を対象に、1CPU 時の性能単価を測定してモデル化する。バリデーションには、1CPU のほか、同構成で 2CPU にした場合についても実施する。

(2) 第二段階

RAC 構成を構築し、フェーズ 1 と同じ測定を行い、両者の差異の把握とモデル化を行う。

3. Oracle シングルインスタンスモデルの測定方法

Oracle RAC モデル作成の最初の段階として、DBT-1 ベンチマークプログラム ver.2.1 を対象に、フェーズ 1 の実測を行った。

3.1 測定環境

DBT-1 ベンチマークプログラムの測定環境を表 1 に示す。それぞれのマシンでは DBT-1 プログラムとモニタ用ツール以外のプログラムは使用せずに測定する。ネットワーク環境は、ローカルネットワークであり、他のマシンも接続されているが、他のマシンのトラフィック量は非常に少ないため無視できる。

表 1 DBT-1 ベンチマークプログラム測定環境

	サーバー	クライアント
機種	SunFire 280R	Dell Precision 650
スペック	UltraSPARC III 1.2GHz x 2 (712.SPEC2000) 2GB メモリ	Xeon 3.5GHz x 1 Hyper-Thread Off 1GB メモリ
OS	Solaris 9	RHLE 3.0
アプリ	Oracle 10g	DBT-1 2.1

DBT-1 は、通常クライアントとデータベースが同一のマシン上で実行される。しかし、その場合、DBT-1

自身の負荷が性能評価に影響を及ぼしてしまうため、本報告では、DBT-1 をクライアント部分と、SQL トランザクション実行部分に分けてインストールした。

3.2 測定方法

本報告の測定対象は CPU 稼働率および、ネットワーク伝送量とする。それは、チューニングが十分に行われた状態の Oracle に発生するボトルネックは CPU であることが分かっているためである。本報告でも事前にチューニングを行い、ほぼすべてのアクセスがキャッシュにヒットするようにして、ディスクの負荷がかからないように設定した。

DBT-1 では e-コマースを想定した 14 種類のインタラクションが定義され、それらの混在比率による 3 種類の Mix が用意されている。DBT-1 のインタラクションと 3 種類うち Shopping Mix の混合比率を表 2 に示す。下記のインタラクションのうち、Search Result インタラクションについては、DBT-1 ver.2.1 では、search_results_author、search_results_subject、search_results_title の 3 種類のインタラクションに分かれているため、さらに 3 等分してそれぞれのインタラクションに分けることとした。

表 2 DBT-1 のインタラクションと、Shopping Mix 混合比率

インタラクション	Shopping Mix の割合
Home	16%
New Product	5%
Best Seller	5%
Product Detail	17%
Search Result	20%
Search Request	17%
Shopping Cart	11.6%
Customer Registration	3%
Buy Request	2.6%
Buy Confirm	1.2%
Order Inquiry	0.75%
Order Display	0.66%
Admin Request	0.1%
Admin Confirm	0.09%

本報告では、DBT-1 を用いた Oracle シングルインスタンス (フェーズ 1) モデルを作成するために、次の三段階で測定を行う。

3.2.1 性能単価の測定

各インタラクショナルを個別に測定し、各インタラクショナルが要するCPU使用量と、ネットワーク伝送量を求め、それらを性能単価として性能モデルに定義する。CPU使用量は、CPU使用時間をOracle付属のAWR (Automatic Workload Repository)⁶⁾を用いて測定した。AWRを用いることで、実際にOracleが使用したCPU時間を正確に求めることができる。ネットワーク伝送量は、オープンソースソフトウェアのWindows版Wireshark⁵⁾を用いて測定した。各インタラクショナルにはOracleへの接続と切断にかかわる処理が加わるため、インタラクショナルを行わない接続切断だけの処理(blank)のときの伝送量を求めておき、各インタラクショナルの総伝送量から接続切断に要する伝送量を引いた値を、各インタラクショナルの伝送量と定義する。なお、性能単価を求める際は、サーバーのCPU数を1にして測定した。また、性能単価の検証方法として、各インタラクショナルを0.1秒ごとに連続実行したときの実測結果とシミュレーション結果を比較するようにした。

3.2.2 Shopping Mixの再現性の検証

Oracle 10g シングルインスタンスDBT-1で定められている三種類のMix、すなわち、Browsing Mix、Shopping Mix、Ordering Mixのうち、参照系と更新系の割合が同等なShopping Mixに対して、実測結果と、(1)で求めた性能単価を組み合わせて構成したシミュレーション結果を比較し、モデルの再現性を検証する。この際も、サーバーのCPU数を1にして測定した。

3.2.3 ハードウェア構成変更による影響調査

構築したモデルを用いて、構成変更の影響の予測と比較を行う。具体的には、CPU数を測定時の1から2へ変更したとき(スケールアップ時)のシミュレーションを行う。また、その結果と実際にCPU数を2にしたときの実測結果と比較して再現性を検証する。

4. Oracle 10g シングルインスタンスモデルの作成

本章では、Oracle 10g シングルインスタンスモデルの実装と、3.2節で示した測定方法による測定結果と、それに基づいて作成した性能モデルのシミュレーション結果の検証について述べる。

4.1 実装方法

本報告で作成したOracle 10g シングルインスタンスモデルは、性能シミュレータ米国HyPerformix社製のIPS Performance Optimizer 3.6.1上で、シミュレータ用記述言語ADN (Application Definition Notation)を用いて性能モデルを実装した。

4.2 CPU時間の測定

各インタラクショナルのCPU単価をAWRのCPU_TIME_DELTAから取得した性能単価を表3に示す。ここで、blankはDBT-1の定義にはないインタラクショナルで接続と切断だけを行う。このインタラクショナルでの性能単価を求めておくことで、各インタラクショナルが純粋に使用する性能単価を得ることができる。

AWRでは各インタラクショナルでOracleがSQL処理に使用するCPU使用量が秒で得られる。そのため、接続と切断だけのblankインタラクショナルのCPU使用量はゼロである。

性能シミュレーションを行う際は、モデルに記述するCPU単位をSPEC_Cint2000に変換する必要がある。表1に示したように、測定機であるSunFire 280RのSPEC値は712と、AWR単価(秒)との積がシミュレーションに必要なCPU単価となる。

表3 CPU単価の測定結果

Interaction	AWR 単価 (秒)	AWR 単価 (SPEC)
admin_confirm	0.002663307	1.896274894
admin_request	0.000985948	0.701994812
best_sellers	0.042162208	30.01949174
buy_confirm	0.00189486	1.349140607
buy_request	0.005974356	4.253741131
home	0.002945741	2.0973678
new_products	0.041984171	29.89272984
order_display	0.005775417	4.11209665
order_inquiry	0.000939183	0.668698531
product_detail	0.001443529	1.02779287
search_results_author	0.002104066	1.498095108
search_results_subject	0.040403251	28.76711468
search_results_title	0.03052188	21.73157889
shopping_cart	0.213672072	152.1345149
blank	0.006528457	4.648261339

4.3 サーバー滞留時間の算出

AWRでは、CPU_TIME_DELTAに加えて、ELAPSED_TIME_DELTAがレポートされる。前者は純粋にCPU使用時間であり、後者はI/O待ちなども含めたひとつのトランザクションの処理時間であり、したがって、滞留時間はELAPSED_TIME_DELTAとCPU_TIME_DELTAの差がサーバーで求められる。そのように求めたサーバー滞留時間を表4に示す。

表 4 サーバー滞留時間

Interaction	ELAPSED_TIME _DELTA (秒)	滞留時間 (秒)
admin_confirm	0.00274775	8.44429E-05
admin_request	0.001013356	2.74084E-05
best_seller	0.044800588	0.00263838
buy_confirm	2.809006357	0.095164385
buy_request	0.006265952	0.000291596
home	0.003028295	8.25534E-05
new_products	0.044475654	0.002491483
order_display	0.005992464	0.000217047
order_inquiry	0.000970285	3.11013E-05
product_detail	0.001465129	2.15993E-05
search_request	0.002139536	3.54698E-05
search_result_ author	0.041783525	0.001380274
search_result_ subject	0.031611428	0.001089547
search_result_ title	0.220955518	0.007283446
shopping_cart	0.006716725	0.000188268
blank	0	0

次に、ネットワーク伝送量について、Wiresharkを用いて測定した結果を表 5 に示す。ネットワーク伝送にはクライアントから Oracle へ SQL アクセスする送信方向と Oracle からクライアントへ結果を返す受信の二つの方向がある。ネットワーク伝送にはクライアントから Oracle へ SQL アクセスする送信方向と Oracle からクライアントへ結果を返す受信の二つの方向がある。

表 5 ネットワーク伝送量の測定結果

Interaction	方向	総転送量 (byte)	実転送量* (byte)
admin_confirm	送信	10935	3493
	受信	9579	2304
admin_request	送信	9495	2053
	受信	8903	1628
best_sellers	送信	15856	8414
	受信	34909	27634
buy_confirm	送信	22791	15349
	受信	7746	471
buy_request	送信	23104	15662
	受信	11054	3779

home	送信	10210	2768
	受信	8920	1645
new_products	送信	34567	27125
	受信	15843	8568
order_display	送信	27110	19668
	受信	11954	4679
order_inquiry	送信	8861	1419
	受信	8846	1571
product_detail	送信	10555	3113
	受信	9519	2244
search_request	送信	8602	1160
	受信	7481	206
search_result_ author	送信	28956	21514
	受信	12355	5080
search_result_ subject	送信	28969	21527
	受信	15587	8312
search_result_ title	送信	28966	21524
	受信	12033	4758
shopping_cart	送信	21481	14039
	受信	7680	405
blank	送信	7442	0
	受信	7275	0

5. Oracle 10g シングルインスタンス モデルの検証

4 章で測定した性能単価を用いた性能モデルを作成した。本章では、各々のインタラクションの再現性および、Shopping Mix について表 2 に示した混合比率でトランザクションを発生させたときの再現性を比較した。

5.1 性能単価の検証

はじめに各インタラクションについて個別にトランザクションを流した状況を実測とシミュレーションで比較する。性能単価の検証に用いた性能モデルのトポロジーを図 1 に示す。この性能モデルはインタラクションごとにクライアントを用意して、必要なインタラクションに対するクライアントだけを稼働させるようになっている。検証にあたっての目安として、本報告ではシステム構築者への定性的なヒアリングから 30% 以内という差異を妥当な範囲とする。

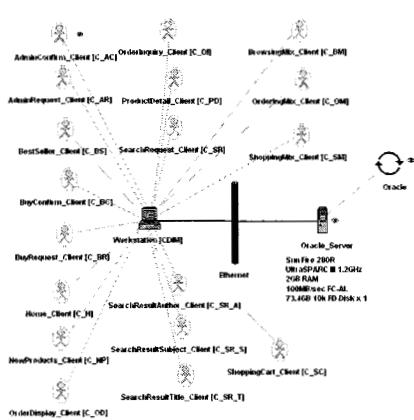


図 1 性能単価測定用モデル

この性能モデルを用いて、各インタラクションについて ThinkTime（トランザクションが戻って次のトランザクションを発生させる間隔）を 0.1 秒と設定した状況を、CPU 時間、実行時間について実測結果とシミュレーションによる予測結果を比較して検証する。はじめに CPU 時間について比較した結果を図 2 に示す（グラフ上の CPU 時間のオーダーが異なる search_results_title はグラフからはずしている）。

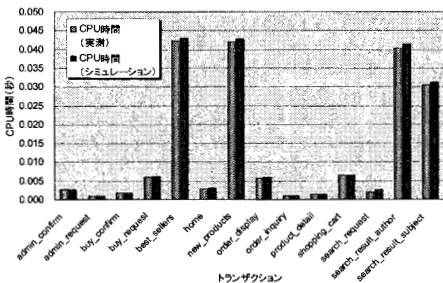


図 2 インタラクションごとの CPU 時間の比較

また、すべてのインタラクションの値と、差異の一覧を表 6 に示す。

表 6 インタラクションごとの CPU 時間の差異

トランザクション	CPU 実測(秒)	CPU 予測(秒)	差異 (%)
admin_confirm	0.002663	0.002720	2.13
admin_request	0.000986	0.001010	2.44
buy_confirm	0.001895	0.00194	2.38
buy_request	0.005974	0.006100	2.10
best_sellers	0.042162	0.043100	2.22

home	0.002946	0.003010	2.18
new_products	0.041984	0.042900	2.18
order_display	0.005775	0.005900	2.16
order_inquiry	0.000939	0.000959	2.11
product_detail	0.001444	0.001470	1.83
shopping_cart	0.006528	0.006670	2.17
search_request	0.002104	0.002720	29.27
search_result_	0.040403	0.041300	2.22
search_result_	0.030522	0.031200	2.22
subject			
search_result_title	0.213672	0.218000	2.03

この結果、CPU 時間に関して、各トランザクションの差異の平均値は 3.98%、最小は 1.83%(product_detail)、最大は 29.27%(search_request)となった。

次に、I/O 待ちなども含んだ実行時間について、CPU 時間と同様、オーダーが異なる search_results_title を除いた、実測とシミュレーションによる予測結果を比較したグラフを図 3 に示す。また、すべてのインタラクションの値と、差異の一覧を表 7 に示す。

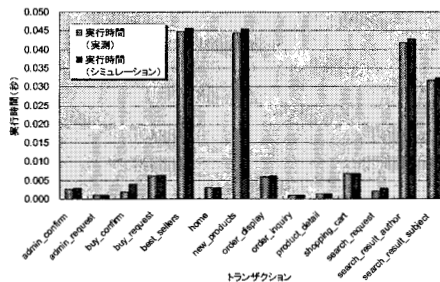


図 3 インタラクションごとの実行時間の比較

表 7 インタラクションごとの実行時間の差異

トランザクション	実行時間 実測(秒)	実行時間 予測(秒)	差異 (%)
admin_confirm	0.002748	0.002810	2.22
admin_request	0.001013	0.001030	1.62
buy_confirm	0.001912	0.003850	50.35
buy_request	0.006266	0.006390	1.94
best_sellers	0.044801	0.045700	1.97
home	0.003028	0.003090	2.00
new_products	0.044476	0.045400	2.04
order_display	0.005992	0.006120	2.08

order_inquiry	0.000970	0.000991	2.09
product_detail	0.001465	0.001500	2.32
shopping_cart	0.006717	0.006860	2.09
search_request	0.002140	0.002810	23.86
search_result_ author	0.041784	0.042700	2.15
search_result_ subject	0.031611	0.032300	2.13
search_result_title	0.220956	0.226000	2.23

この結果、実行時間に関して、各トランザクションの差異の平均値は6.74%、最小は1.62%(admin_request)、最大は50.35%(buy_confirm)となった。ここで、差異の最大値が妥当な範囲を満たしていないが、表2にあるように、当該インタラクションのbuy_confirmは、Shopping Mixのうち1.2%しか発生しないため、このまま用いても全体性能予測に大きな影響を及ぼさないと考え、このままこの値を用いてShopping Mixの測定を行うこととした。

5.2 Shopping Mix 再現性の検証

はじめに Shopping Mix の検証に用いた性能モデルのトポロジーを示す。

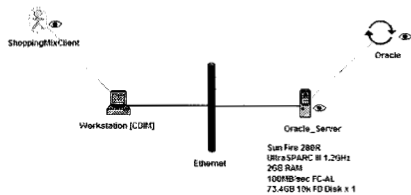


図4 Shopping Mix の検証モデル

このモデルを用いて、DBT-1のShopping Mixについて、1CPU構成で実測とシミュレーション予測を実施した。Shopping Mixでは、同時アクセスを行う仮想ユーザ数（EU: Emulate User）を増加させたときのCPU稼働率、応答時間、スループットの3種類のデータを取得した。これらの比較結果をそれぞれ、図5、図6、および図7に示す。

各グラフとも実測とシミュレーションの間に類似性が見られるが、一般にスループットが最大になる箇所が性能限界とされる。そこでスループットが増加から減少へ転ずる箇所をグラフから読み取ると、実測での性能限界EU=410、シミュレーションで予測した性能限界EU=360となり、両者の差は約12.2%である。以上のことから、1CPU時のShopping Mixの再現性に関しては、妥当な範囲で予測できたことになる。

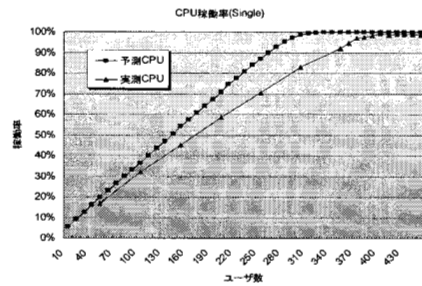


図5 1CPU時のShopping Mix CPU稼働率の比較

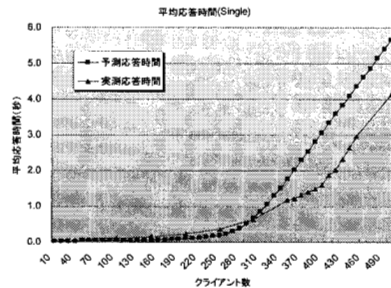


図6 1CPU時のShopping Mix 平均応答時間の比較

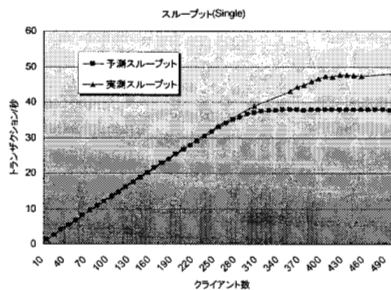


図7 1CPU時のShopping Mix スループットの比較

5.3 2CPU時のShopping Mix 再現性の検証

次にスケールアップの予測の再現性を確かめるために、同じモデルを用いてサーバーのCPUを二つにした状態で、シミュレーションと実測を行った。1CPU時と同様に、Shopping Mixでは、同時アクセスを行うユーザ数（EU数）を増加させたときのCPU稼働率、応答時間、スループットの3種類のデータを取得した。これらの比較結果をそれぞれ、図8、図9、および図10に示す。

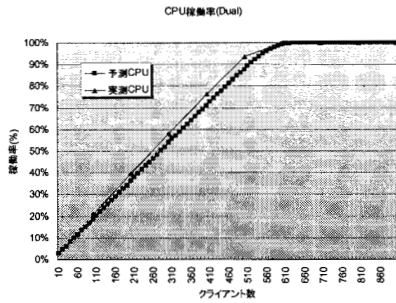


図 8 2CPU 時の Shopping Mix CPU 稼働率の比較

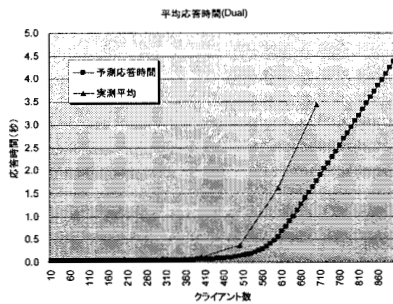


図 9 2CPU 時の Shopping Mix 平均応答時間の比較

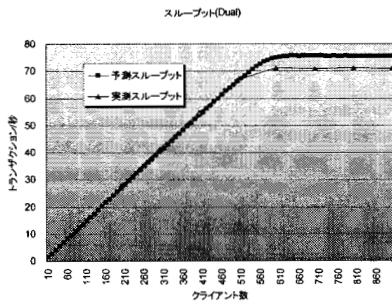


図 10 2CPU 時の Shopping Mix スループットの比較

ここでもスループットが増加から減少へ転ずる箇所をグラフから読み取ると、実測での性能限界 EU=600、シミュレーションで予測した性能限界 EU=640 となり、両者の差は約 6.67%である。以上のことから、2CPU 時の Shopping Mix の再現性に関して、妥当な範囲で予測できた。つまり、本報告の性能モデルはスケールアップ時の状況の性能予測が可能であることを示していると言える。

6. 結言

6.1 結論

Oracle 10g の性能予測を実現するために、第一段階

として、Oracle シングルインスタンスシステムを対象に、SunFire 280R (Solaris 9) 上で DBT-1 ベンチマークを実施し、その結果を反映させた性能モデルを作成した。シミュレーション結果を実測と比較した結果、次のことがわかった。

- (1) DBT-1 を構成する各インタラクションの性能予測では、一部を除き実測との間に有意な差が認められなかったこと
- (2) 1 CPU 時の DBT-1 の Shopping Mix ベンチマークにおいて、限界負荷を約 12% で予測できること
- (3) 2CPU 時のスケールアップした状態での Shopping Mix ベンチマークにおいて、限界負荷を約 7% で予測できること

6.2 今後の課題

本報告は Oracle に対するシミュレーションによる性能予測のアプローチであり、今回実験したスケールアップだけでなく、RAC 構成によるオーバーヘッドに関する Oracle 社のホワイトペーパー^[7]などを参考に RAC 構成 (スケールアウト) での性能予測ができるようにしたい。

7. 参考文献

※下記に示した URL は、報告日時点で有効である。

- [1] Oracle Japan / Oracle Database 10g / Oracle Real Application Clusters 10g:
<http://www.oracle.co.jp/database/rac/>
- [2] <http://www.tpc.org/>
- [3] 「OSS 性能・信頼性評価 / 障害解析ツール開発」DB 層の評価：独立行政法人 情報処理推進機構、2004
- [4] Automatic Workload Repository,
http://otn.oracle.co.jp/skillup/stats_diag/1/index.html#c9
- [5] <http://www.wireshark.org/>
- [7] IBM System p 上での Oracle Application Clusters 10g 性能検証：日本オラクル株式会社、2007
http://otndnld.oracle.co.jp/tech/grnd/pdf/IBM_p5_RAC_WhitePaper.pdf

商標について

本報告中で用いられている製品名については、各社の登録商標です。