

## 大規模基幹系オープンシステムを対象とした性能見積り手法の検証

5 D-6

持田 直穂・金子 貴夫・村田 博  
NTT データ通信(株)技術開発本部

## 1. はじめに

性能見積りとは、実現する処理や採用するハードウェア・ソフトウェアを仮定して、その処理能力を概算することを指す。見積り結果は業務処理システムの提案、設計段階において、機種選定の根拠や妥当性の評価資料となるばかりではなく、試験工程の評価基準やチューニングの目標値としても利用されている。後工程において機種の変更などを行うと、予定した開発費用を超過する危険があり、ハードウェアの処理能力の不足や過剰を極力回避するため、適切な性能見積りが必要とされる。

従来主流であった汎用機集中型システムの性能見積りにおいては、見積り指標である CPU の MIPS 値とソフトウェアの DS(Dynamic Step)が、実測、経験則などを問わず把握できることを前提としていた。ところが、今日主流のいわゆるオープンシステムの構築では、個々のハード・ソフトはブラックボックス同然であり、それらの見積り指標を把握するのは難しい。殊に UNIX マシンでは CPU の能力指標として MIPS が標準になっていないなどの条件も重なり、従来の手法ではシステムの処理能力を見積ることが困難であり、最適設計を実施する上での妨げとなっている。

一方で、TPC などのベンチマーク測定結果が公表され、機種選定の手助けとなっている。しかし、公表される製品の組み合わせには限りがあり、市販製品すべての比較評価尺度にはなり得ないこと、システム構成や業務内容がいくつかのパターンに固定されていて、構築する業務処理システムと一致する例は少ないことなどから、顧客の業務要件にあわせたシステムの設計・製造を手掛ける場合、完全な代替手段とすることはできない。

## 2. 見積り手法の概要

## 2.1 見積り式

今回は、我々の研究開発のターゲットである大規模金融基幹系オープンシステムを例に検討を行った。システムの主な特徴には SMP マシンの採用、RDB 操作中心の業務 AP が挙げられる。提案や設計初期の段階で利用するために、見積り対象を「サーバの機種(CPU)のみに依存する最大スループット」に絞る。

従来の見積り手法で基本とされる式は、

$$3600[\text{sec}] / (\text{APの総DS} + \text{MIPS}) \quad (\text{式1})$$

ここで分母は「トランザクションの CPU 処理時間」を意味しているので、これを直接把握することが可能であれば、MIPS と DS を使わずに性能見積りが実施できる。そこで、AP 内で発行している SQL の CPU 処理時間(以降、SQL 処理時間と呼ぶ)に注目する。SQL 処理時間と AP 中の SQL の数より AP の CPU 処理時間を把握する。見積り式を示すと、

$$3600[\text{sec}] + \sum_{\text{SQL数}} (\text{SQL処理時間}) \times \text{CPU数} \quad (\text{式2})$$

## 2.2 SQL 処理時間の予測

SQL 処理時間の内訳は、「CPU の処理時間」と「I/O 装置の処理時間」に大別できる。このうち、I/O 装置の処理時間部分は、より高速な I/O 装置の導入によって短縮が可能であるため、あらかじめ最小の 0 であるとみなす。CPU の処理時間は、「純粋実行時間」と「overhead 時間」とに分解する。純粋実行時間は CPU の処理能力と SQL の種類(参照、更新など)によって確定すると考える。一方、overhead の主要因は共有メモリ、共有 I/O 装置の競合・排他

制御によるもので、コンピュータの CPU 数、CPU の平均負荷という外部から観測可能なパラメータに依存するとみなす。そこで、CPU の処理時間について SQL の種類、CPU 数、CPU の平均負荷の関数として予測式を作る。

### 3. 検証

検証作業には大型 SMP サーバマシンと銀行振替業務のプロトタイプ AP を使用した。

#### 3.1 「SQL 処理時間合計≒AP の CPU 処理時間」の確認

1 トランザクションにおける AP の総処理時間と SQL 処理時間の合計とを比較し、SQL 処理時間の合計を AP の CPU 処理時間とみなせることを確認した。

#### 3.2 見積り式の妥当性

サーバ内の CPU 数、CPU の平均負荷を変化させつつ計 15 パターンについて SQL 処理時間を測定し、(式 2)に代入した。この値と、同一機種で性能評価試験を実施した際の測定値とを比較した。見積り式の計算値は測定値から予想される範囲に留まり、見積り式は妥当なものであると判断できる。

#### 3.3 SQL 処理時間の予測式の作成

検証環境で select と update について回帰分析した予測式を以下に示す。N は CPU 数、 $\alpha$  は CPU 負荷係数(CPU の平均負荷)である。係数の寄与率や、式の持つ意味などについての検討は今後の課題として残されている。

$$\text{SQL 処理時間(select)[msec]} = (-0.00514 \times N + 1.119532) + (0.064553 \times N + 0.794187) \times \alpha \quad (\text{式 3})$$

$$\text{SQL 処理時間(update)[msec]} = (0.003954 \times N + 1.325902) + (0.071277 \times N + 1.072968) \times \alpha \quad (\text{式 4})$$

#### 3.4 見積り例

##### 3.4.1 検証環境と同一機種の場合(例 1)

検証用の AP には、select が 40、update が 15 含まれている。 $\alpha=1.0$  と仮定し、(式 2)(式 3)(式 4)から求めた「筐体内 CPU 数—スループット」のグラフを図 1 に示す。あわせて別に実施した CPU スケーラビリティ評価における実測値をプロットしてある。ほぼ正確な見積り値が得られていることが確認できる。

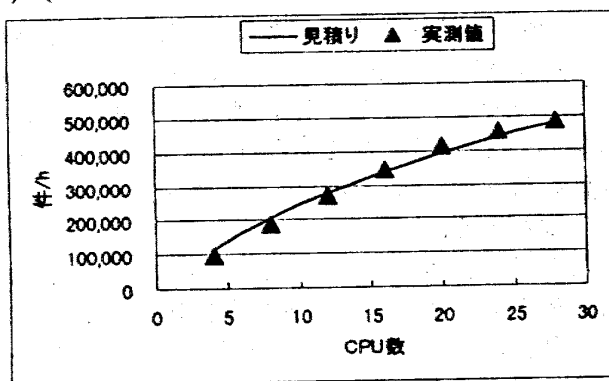


図 1 筐体内 CPU 数—スループット(例 1)

##### 3.4.2 上位機種への適用(例 2)

CPU のクロックと搭載数を向上させた上位機種に対して見積り式を適用する。select/update の数、 $\alpha$  の値は例 1 と等しいとし、SQL 処理時間を CPU のクロックの比(2:3)で換算する。図 2 は「筐体内 CPU 数—スループット」のグラフである。同様に実測値をプロットしてある。ここでも十分な精度の見積り値が得られ、あるサーバで SQL 処理時間予測式を作成しておけば、同一 CPU アーキテクチャの機種についても見積りが可能であることが示された。

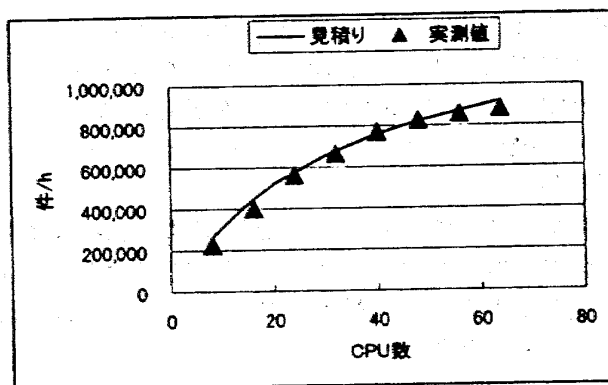


図 2 筐体内 CPU 数—スループット(例 2)

### 4. まとめ

今回の取り組みで、UNIX マシンを対象とした、汎用機向けと同程度に簡素で扱いやすく、かつ顧客のシステム要件を反映できる性能見積り手法を示すことができた。特に、見積り式が同一 CPU アーキテクチャの機種間で換算可能であることが確認された点は、予測式作成のためのデータ収集ターゲットを絞りこむことができる可能性を示唆していて注目できる。今後の展開には 2 つの方向性を考えている。一方は、より多くの機種を対象に検証作業を実施することであり、もう一方は、SQL 以外のデータアクセス方式を有する AP に対して見積り手法を適用し、検証することである。