

分散システムの性能予測方法の検討

臼淵啓明[†] 持田直穂 岩金純市 林真紀

bucci@open.rd.nttdata.jp[†]

NTTデータ通信(株) 技術開発本部

〒135 東京都江東区豊洲3-3-3

本報告では、分散システムの設計初期において重要な性能予測の方式について述べる。具体的には、性能値の積み上げ方式と離散型のシミュレーションを利用する方式を提案する。積み上げ方式は、モデルとなるシステムについて実測を行い、実測値を統計分析して回帰式を求める。この回帰式に見積もり対象のシステムのパラメータを代入し、基礎となる性能値を求める。この性能値を積み上げて見積もり値を求める。本方式についてTPC-Aを元にしたOLTP業務のモデルで評価を行なった結果、性能見積もりの方式として有望であるとの結論を得た。シミュレーションを利用する方式は、積み上げ方式と同じモデルでの評価の結果、積み上げ方式と同等の精度が得られた。しかし、設計初期の見積もり方式としては、実施の容易さと早さの点から積み上げ方式が有利である。積み上げ方式は、より多くのシステムのモデルと性能レンジでの評価を蓄積することにより、適用範囲を広げることができる。

A Study of Performance Estimation for Distributed Computing Systems

Hiroaki Usubuchi, Tadahito Mochida, Jun-ichi Iwagane and Maki Hayashi

Research and Development Headquarters

NTT DATA Communications Systems Corporation

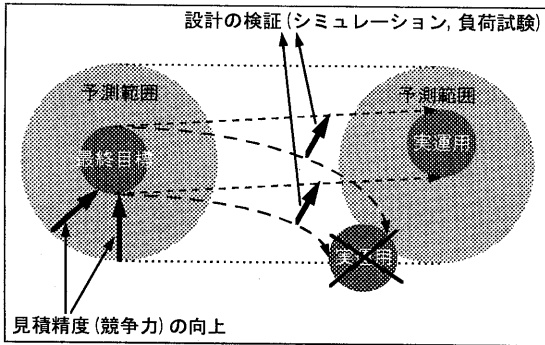
3-3-3, Toyosu, Koutou-ku, Tokyo, 135 Japan

We discussed the ways of performance estimation in early designing stages for distributed computing systems. We propose two methods, one is "cumulative method", another is "discrete simulation method". In cumulative method, we apply experimental design to model system's tests and generate various regression equations corresponding to performance factors such as response time. Next, we assign the real system's parameters to the each regression equation and get results. The cumulating of the results is what we want to estimate. We tested this method on OLTP system models which were based on TPC-A benchmark model and concluded it was usable. We also evaluated discrete simulation method and got as good results as cumulative method. But, in early designing stages, we recommend cumulative method because of its easiness and quickness. We expect that the more we will test various model systems in various performance ranges, the more the cumulative method will be applicable.

1. 分散システムの性能問題

分散システムは、標準化の進んだオープンな環境により、安価で自由度の高いシステム構築を可能にした。これにより、計算機システムの適用分野を拡大するとともに、テンポの速い技術革新をもたらしている。

反面、計算機システムの構成要素は多種に及び、性能問題の解決が困難になっている。性能問題は、計算機システムのライフサイクルの各段階にそれぞれ存在する。典型的な、計算機システムのご提案から構築、納入までを一括して行う形態について、段階ごとの性能問題を以下に列挙する。また、これらの問題の関係を図2-1に模式的に示す。



〈図2-1〉

(1) 設計初期

性能をいかに精度よく見積もるか。それにより提案時の費用をいかに抑えるか。

(2) 設計後期

計画の性能を実現する設計であることをいかに検証するか。それにより試験時の設計の手戻りを如何に削減するか。

(3) 試験

実運用で生じ得る負荷の環境をいかに実現し、問題なく運用できることを検証するか。

(4) 実運用

各種の機器の使用率が設計値を大きく越えるなど、計算機システムの運用に支障が生じる事態が生じていないか。

これらの問題の中で、もっとも重要なのはシステム構築の初期に存在する性能見積もりの問題である。システム構築の初期に決定した事項は、後の工程では前提となり、変更するためには大きなコストを要するからである。

2. 課題

分散システムの性能見積もり方式として、従来、3つの方式が考えられてきた。それぞれの比較を表3-1に示す。3方式とも大きな短所があり、対処が必要である。新しい方式を考える上で必要な要件を以下に示す。

(要件1) 分散システムの適用分野は拡大する傾向にある。これに対応するため、種々の計算機システムへの応用が可能であること

〈表3-1〉

方式	内容	長所	短所
1	DSやI/O時間の積み上げ	異なるシステム構成への応用が容易	アーキテクチャの多様化 ブラックボックス化 動的な挙動
2	シミュレーション	結果の正しさへの理論的裏づけがある	高コスト(時間, 工数) モデリング技術が発展途上 データ収集の困難さ
3	ベンチマーク値の類推	簡易 情報収集が容易	類推方式が未確立 条件の違いの変換

(要件2) 基礎となる設計値は設計初期に把握または推定可能であること

(要件3) 基礎となる設計値は外部から安価に観測可能であること

要件1を満足させるには、性能値を部品化し、適用するシステムごとに組み合わせることが可能な、積み上げ方式が有効である。しかし、積み上げる性能値として従来から用いられてきたDS(ダイナミックステップ)等は、分散システムにおいては要件2,3を満足しない。そこで、新たな性能値として、以下に示す手順で得た値を用いることを提案する。

- (1) 計算機システムを類型化し、該システムで要件2,3を満足する性能値を設定する。
- (2) 設定した性能値について、モデルとなるシステムにおいて、トラフィック量などの要件とハードウェアの設定などの環境条件を変化させて実測を行う。
- (3) 実測値を統計分析して、要件と環境を説明変数とする回帰式を得る。
- (4) 回帰式に要件と環境条件を代入して値を求める。それらを積算して、見積もり対象の性能値を得る。

3. 統計分析から得た性能値の積上げ方式

本報告では、各種の計算機システムの類型のうち、OLTP(オンライントランザクション処理)について、提案する見積り方式の評価を行う。

OLTPシステムの性能値として設定する性能値を、表4-1に示す。表に示すように、各性能値は求められる見積り方式の3要件を満たす。これらの性能値が要件と環境から説明できることが明らかになれば、見積りのための性能値として利用できる。

3.1 性能値の算出方法

性能値を算出するために、トラフィック量などの要件とハードウェアの設定などの環境条件を要因として抽出する。次に、各要因のとり得る値を変えて、実験計画に基づく多元配置実験を行う。この方式では、内部の仕組みを理解することにとらわれず、外部の条件を変化させたときの挙動の変化から、未知の条件での挙動を予測する。化学や農業の分野においてよく利用されてきたこの方式を、複雑化した計算機システムに適用することを考える。実際の手順を以下に示す。

- (1) 目標とするシステムのモデルを選定する。
- (2) 選定したモデルで性能に影響を与えると考えられる要因を抽出する。
- (3) 抽出した要因について実験計画に基づく多元配置実験を行う。
- (4) 各要因について分散分析を行い、有意性を判定する。
- (5) 有意と判定された要因について寄与率を算出する。
- (6) 要因間の交互作用の存否を確認する。

〈表4-1〉

性能	設定する性能値	要件1	要件2	要件3
応答時間	SQL単位の応答時間	標準化されたAPIである	業務内容とデータ種別から推定可能	クライアントAPで取得可能
スループット	ボトルネックの恐れのある各種資源の使用率			
	SQL単位の用途別ディスク使用率	標準化されたAPIである	業務内容とデータ種別から推定可能	sar等の標準ツール
	SQL単位のネットワーク使用率	標準化されたAPIである	業務内容とデータ種別から推定可能	netstat等の標準ツール LANアナライザ

- (7) 直交多項式、または、回帰式を算出する。
- (8) 回帰式について当てはまり具合を検証する。

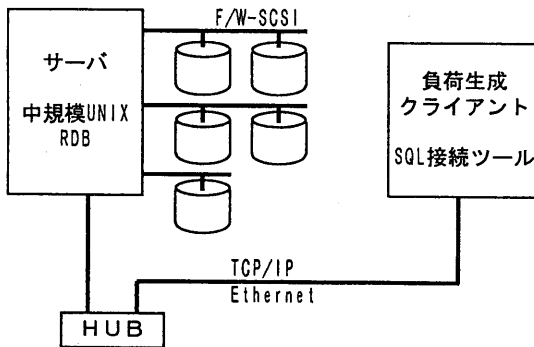
現在まで、手順(1)のシステムのモデルとして、LANを介してクライアントとサーバを結合し、SQLで接続するオンライントランザクション処理の環境で評価を行った。表4-2に、評価にあたり設定した主な要因の一覧を示す。表4-3に、測定項目の大別を示す。細かく数え上げると、20項目弱を測定した。図4-1にシステムの構成図を示す。

〈表4-2〉

項番	要因	水準
1	端末数(台)	~50
2	端末あたりのトランザクション間隔(秒)	5, 10
3	トランザクションあたりのSQL数	~20
4	データベースのキャッシュ量(ブロック)	~10000
5	マスタのテーブル行数(万行)	~300
6	ログのバッファ量(KB)	~100

〈表4-3〉

項番	測定項目
1	個々のSQL文の応答時間
2	CPU使用率
3	ボリューム毎のディスク使用率
4	ネットワーク使用率

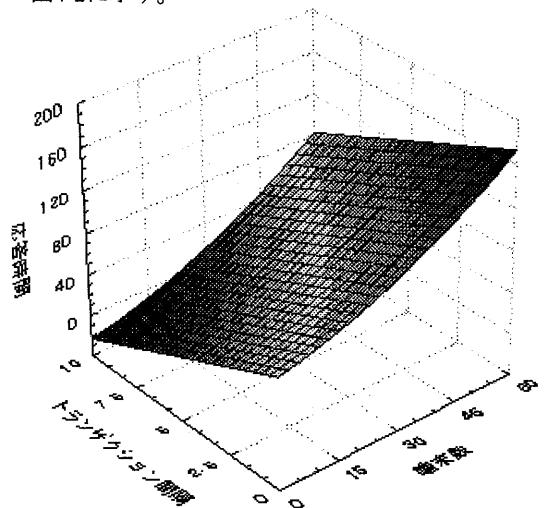


〈図4-1〉

3.2 積み上げ方式の評価

評価から以下の点が明らかになった。

- (1) 分散分析の結果、9割以上の測定項目について、上位3までの要因の寄与率の合計が過半数を越える(表4-4)。
- (2) 分散分析の結果、5割以上の測定項目について、寄与率の上位3までの要因に、交互作用を含まない(表4-4)。
- (3) 直交多項式を求め、その決定係数を求めると、約7割以上の多項式で5割を越える(表4-4)。例として、select文の応答時間について求めた直交多項式をグラフ化したものを、図4-2に示す。

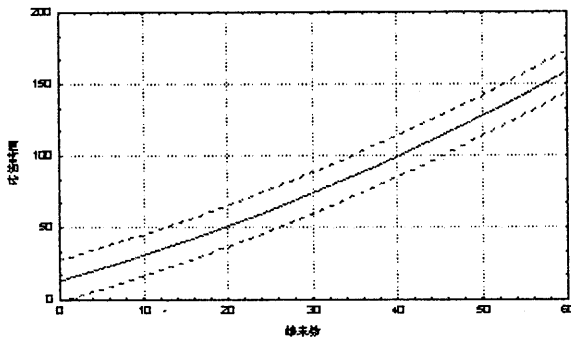


〈図4-2〉

- (4) 寄与率が最大の要因について、95%の区間推定を行ない、上下限値と平均値の乖離を見る。推定の区間は等幅なので、値のもっとも小さな水準で乖離が最大になる。この場合でも、9割以上が平均値の±50%以内に収まる(表4-4)。これは設計初期の見積もりとしては許容範囲内である。例として、1つの要因について、select文の応答時間の乖離の程度を、図4-3に示す。実線が平均値、上側の点線が95%の上限値、下側の点線が95%の下限値を示す。上記の結果、モデルとして設定したオンラ

〈表4-4〉

項番	種別	累積寄与率50%までの要因数	寄与率上位3位内の交互作用の数	直交多項式の決定係数	最小水準での95%区間推定の上下限と平均との乖離
1	CPU平均アイドル率	3	1	0.555	6.0%
2	TABLE平均ビジー率	3	1	0.588	8.8%
3	INDEX平均ビジー率	3	0	0.645	10.2%
4	RBS平均ビジー率	3	1	0.724	9.2%
5	LOG平均ビジー率	3	1	0.418	11.8%
6	スループット	3	1	0.459	8.9%
7	connect平均応答時間	3	1	0.476	31.0%
8	update平均応答時間	3	0	0.565	37.8%
9	select平均応答時間	3	0	0.422	54.8%
10	insert平均応答時間	3	0	0.549	45.9%
11	commit平均応答時間	4	0	0.503	37.0%
12	disconnect平均応答時間	3	0	0.565	47.3%
13	LANトラフィック	1	0	0.869	9.4%



〈図4-3〉

トランザクション処理のシステムについては、以下の値を積み上げの基礎となる性能値として利用可能であると判断できる。

(A) 応答時間

SQL文個別の応答時間

(B) スループット

SQL文個別のCPU使用率、ディスク使用率、ネットワーク使用率。逆算して100%以下になるトラフィック量を上限とする。

他のシステムのモデルについては、別途、検証を進めていく必要があるが、今回の結果から、本方式は有望であると判断する。

4. シミュレーションを利用する方式

これまで、積み上げ方式について検討状況を報告した。これとは別に表3-1で示した第2

の方式であるシミュレーションについて、設計初期の見積もりへの適用性を検討した。

離散型のシミュレーションによる性能見積もりは、汎用機システムで多くの実績がある。分散システムにおいても、見積もり対象にあわせて適切な資源と待ち行列を設定するという考え方自体は同じである[1]。ただし、何を資源として捉えるのがよいかという点は異なり得る。積み上げ方式と同じ実測環境を対象に、評価を行なう。評価方法を以下に示す。

- (1) 時間遅延が生じる資源を抽出し、その継続関係を明らかにする。設定する資源の例を表5-1に示す。
- (2) トラフィックに関する要件(端末台数、端末当たりのトランザクション間隔)を除き、その他の要因の組み合わせ毎にトランザクションを単独で発生させ、各測定項目について値を取得する。

〈表5-1〉

項番	種別	資源名
1	ハードウェア	CPU
2		物理ディスク
3		SCSIバス
4		LAN
5		クライアント
6	ソフトウェア	トランザクション振り分けプロセス
7		サーバプロセス
8		ログ書き出しプロセス
9		データ書き出しプロセス

- (3) トラフィックに関する以外の要件と測定値を各資源の属性として設定し、シミュレーションのモデルを完成する。
- (4) トラフィックに関する要件をパラメータとして、シミュレーションを実施する。

評価の結果、以下の点が明らかになった。

- (A) 要因の水準を変えたときの見積値の変化の挙動と、実測値の挙動はよく一致する。
- (B) 見積もり値の絶対値と実測値の乖離は、平均で25%程度である。見積もり対象とその当てはまり状況について、応答時間の例を表5-2に示す。

〈表5-2〉

試験	実測値	見積値	乖離率
1	0.74	1.17	36%
2	1.08	1.24	13%
3	1.43	1.60	11%
4	1.92	1.90	1%
5	0.84	1.17	28%
6	0.80	1.19	33%
7	1.62	1.64	1%
8	2.19	1.59	38%
9	0.83	1.17	29%
10	0.86	1.23	30%
11	1.06	1.56	32%
12	2.21	1.71	29%
13	0.75	1.17	36%
14	0.92	1.20	23%
15	1.15	1.14	1%
16	2.34	1.53	53%

上記の結果、積み上げ方式と同等の精度で見積もりが可能であるとの見通しを得た。ただし、シミュレーションを用いる方式は、基礎となる数値の測定、モデル作成、実施、分析まで約1週間程度の時間が必要である。これに対し、積み上げ方式は、即座に計算が可能である。したがって、設計初期の見積もりの方式としては、積み上げ方式が有効である。

シミュレーションが効果を発揮するには、計算機システム内部のより詳細な理解が必要である。設計初期の段階では把握できる内部の情報が少なく、積み上げ方式に対して大き

く優位とはならない。むしろ、詳細な情報がそろった先の工程で設計の検証を行なうのに向いている。

5. 結論

本報告をまとめる。

- (1) 計算機システムのモデルとトラフィックのレンジ別に性能値を用意すれば、統計分析から得た性能値の積み上げ方式は、設計初期に必要な精度での性能見積もりの有効な方式となり得る。
- (2) シミュレーションを用いる方式も、設計初期に必要な精度での性能見積もりが可能であるとの見通しを得たが、見積もりに要する時間と工数から、現段階では積み上げ方式の方が有効である。

6. 今後の課題

最後に、提案した見積もり方式を更に有用なものとするための課題を示す。

- (1) 計算機システムのモデル毎に必要な性能値の抽出
- (2) 計算機システムのモデルとトラフィックのレンジ別の性能値の蓄積
- (3) 見積もり方式を利用しやすくするためのインタフェースの検討
- (4) 複雑な計算機システムのモデルにおけるシミュレーションの適用性の検討

参考文献

- [1] 持田直穂, 「シミュレーションによる性能評価と設計改善の事例」, 構造計画研究所カスタマコンファレンス'95, 1995/10/13.