

機能分散型マルチプロセッサ構造を有する オンライン・データベースシステムの応答解析

近 藤 康 男

(日本電気 オープンシステム)

1 はじめに

密結合，ないし疎結合マルチプロセッサシステム構成において，負荷分散をも兼ね，機能分散化をはかる場合，処理能力の向上率もさる事ながら，応答性能について，従来のシステムとの比較など，十分な工夫・配慮が必要となる。

特に，データベースシステムにおいては，従来に比較し数倍のオーバヘッドが消費される可能性が高く，システムの規模の増大が促進される一方，要求性能は増々高くなり，特に高速な応答性が期待されて来ている。

本稿は，このような動向に鑑み，オンラインデータベースシステムを分散型システムにて実現する場合の，技術検討の一環として行なわれた性能検討の一経過について報告するものである。

性能検討の経過としては，

- ① 中規模なオンラインシステムを対象として，データベースシステム化のモデルを作成。
- ② 当該モデルを性能向上するための分散化アーキテクチャを数案作成。
- ③ 性能検討(比較評価等)。本評価において，フローズド・キューイング理論をプログラミング化した性能計算プログラム MARCOF1 (マルコフ1)を用いた。

であるが，以下ここに報告するのは，性能検討の一例と共に 性能計算プログラムマルコフ1 の概要紹介，ならびに評価結果としての一般的と思われる傾向について，である。

内容

1	はじめに	P. 1
2	分散化アーキテクチャ	P. 2
3	応答解析	P. 3
4	フローズド・キューイングモデル	P. 4
4.1	マルコフ1	
4.2	モデル理論	
5	まとめ(定性的意義)	P. 7
6	おわりに	P. 8

2 分散化アーキテクチャ

まず、応答解析のモデルとしてのアーキテクチャについて概要を述べる。

トランザクション処理機能として正常時機能を大別すると、大略以下の如くなる。

- i) メッセージ処理機能(メッセージ送受信処理)
- ii) トランザクション処理機能(トランザクション制御としてのタスク処理)
- iii) アプリケーションプログラム処理機能(アプリケーション処理)
- iv) データベースアクセス処理機能(レコード、ファイルアクセス処理)

これ等の機能は、従来単一プロセスとして統合されている訳けであるが、本モデルは上記各々の機能をプロセスとして独立化し、パラレルプロセッシング可能なように分散化する(図-1参照)。

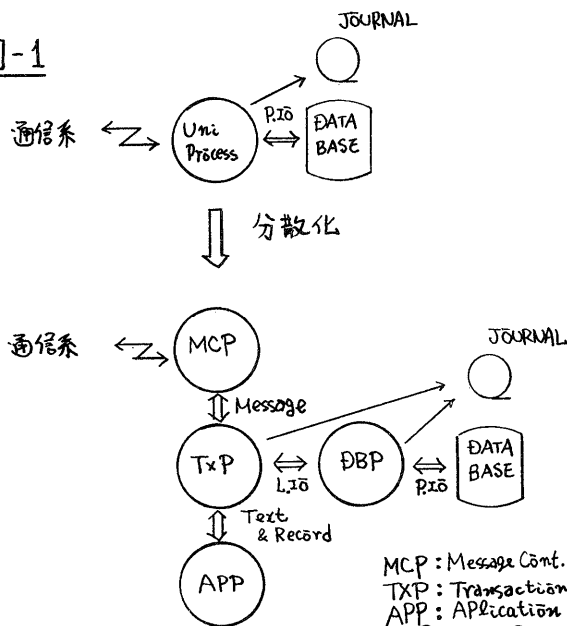
ここで各プロセスは、

- 1) プロセッサとは独立(プロセッサに固定する事も可能)。最大呼量は1 exrl.
- 2) プロセスとしてマルチプロセス化も可能。最大呼量は n exrl (n は整数)。
- 3) 疎結合、疎結合を向わない(ただし、ある一つのプロセスが疎結合システム上に同時に存在する事はない。プロセスとしてシステムには固定)。

とする(図-2参照)。

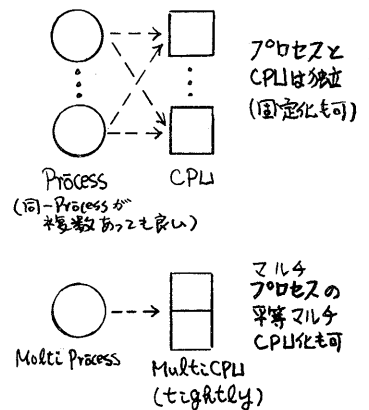
当然の事ながら、トランザクションとしてはタスクとしてシステム上を一連に渡り歩き、同一トランザクションがマルチになる事はない(マルチプログラミングとしては従来に同様)。トランザクションの競合制御(トラフィックコントロール)として、ローディングファクタ N を設定する(N はトランザクションの最大同時多重度)。

図-1



MCP: Message Cont. Process
 TxP: Transaction Process
 APP: Application Process
 DBP: Data Base Process

図-2



3 応答解析

ある中規模なオンラインシステムをデータベース化する事により、トランザクションの負荷が次の如くなるものとする。

CPU負荷：約40ms / Tx / Uni Process . . . ケースA

(これをプロセス分散すると、分割処理を含め次の如くなるものとする。

約10ms / Tx / MCP
約46ms / Tx / TxP
約24ms / Tx / APP
約20ms / Tx / DBP } . . . ケースB

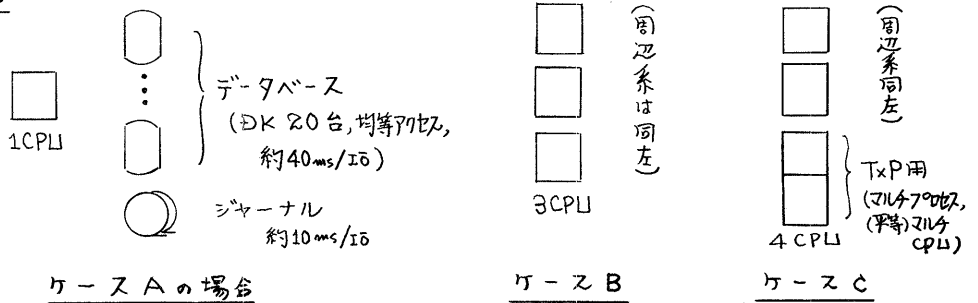
さらに、TxPのみをマルチプロセス化すると、マルチ化処理を含め、次の如くなるものとする(他プロセスは不変)。

約52ms / Tx / TxP . . . ケースC.)

データベースI/O負荷：約5回(Read, Write) / Tx
ジャーナリングI/O負荷：約1回 / Tx } I/Oは分散化に不変

これを下記図-3のシステム構成上で稼働させるものとする。

図-3

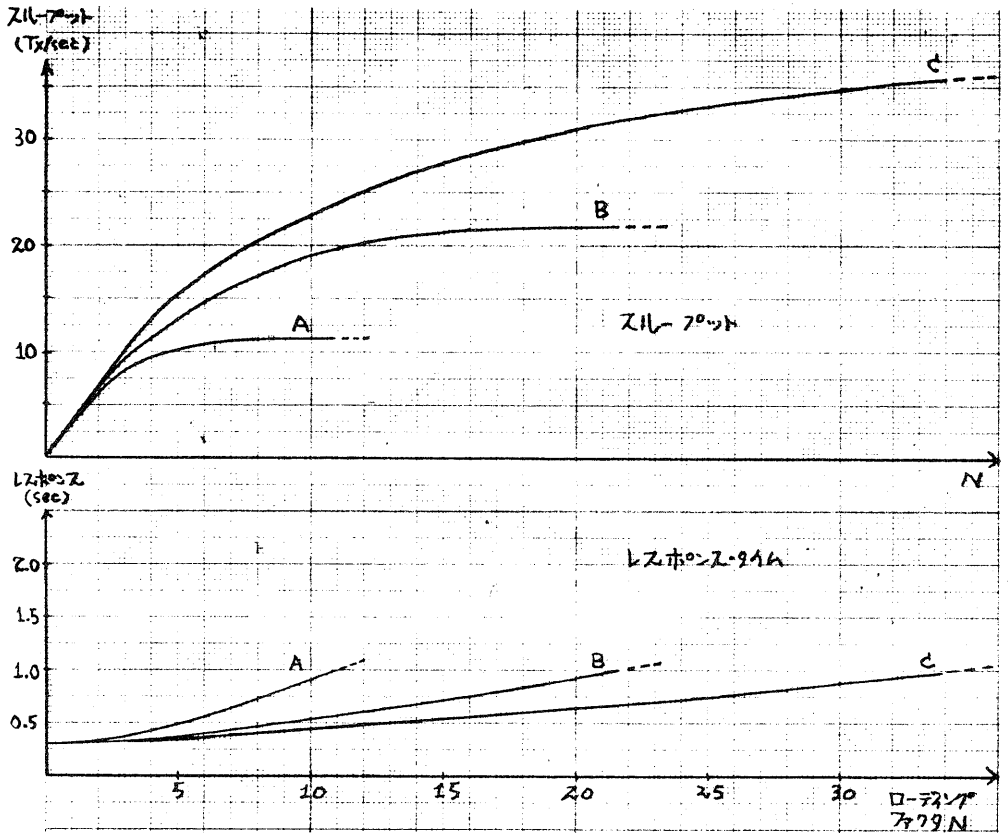


さて、ここにおいて前述したプログラム、マルチプロセッサにて性能計算を行ない、スループット(処理能力)、レスポンスタイムを求める事になる。

この結果を次頁の図-4に示す。図は、先に述べたローディングファクタNをパラメータとして示してある。

図から、分散化による応答性へのデメリットは認められない事、スループットとレスポンスタイムのバランスおよびこのためのロードファクタとの関係(最適ロードファクタ)などが知れよう。

図-4



4 クローズドキューイングモデル

クローズドキューイングモデルないし理論とは、リソースが直列に、また並列に存在する系(ネットワーク系)において、リソースのコーザ(トランザクションなど)の数が有限である場合の、競合状況を定量化(数式化)したものであり、既に発表されている論文の数は相当数にのぼる。

本解析にて用いたマルコフ1も、このモデルに沿うものである。以下、マルコフ1と、その背景理論について概説する。

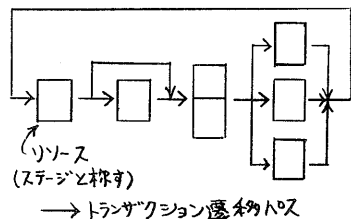
4.1 マルコフ1

マルコフ1は筆者が、既に3年前に発表した会話型の数値計算パッケージである(バッチでも使用可)。プログラムはFORTRANで、約500ステートメントである。

マルコフ1が対象とするネットワーク系の例を図-5に示す。

マルコフ1の入力パラメータ、および出力内容を次頁に示す。

図-5



マルコフ1
入カパラメータ

入カパラメータ	内 容
ステージの数	図-5の如く、待合車の単位による
各ステージについての 保留時間(サービス時間)	トランザクションを単位として 与える
各ステージについての サーバ数	_____
トランザクションの ステージ間遷移確率	トランザクションのステージから への遷移確率を与える
トランザクションの数	系内じ動か作用するトランザクションの数 (トランザクションの全量)

出力内容例

トランザクション ショウホウ ← トランザクションによる全システム性能 (タンイ シカシ:)

トランザクション スウ	=	4	
トランザクション シヨリケン スウ	=	0.247533D+00	
トランザクション シヨリシカシ	=	0.161595D+02	
(サイタ イ シヨリ ステージ = 3)			
ステージ #, サーバ スウ	シヨウリツ	マチシカシ	ホリウシカシ
1, 3 ()	0.825109D-01	0.411015D-03	0.100000D+01
2, 2 ()	0.247533D+00	0.103576D+00	0.200000D+01
3, 1 ()	0.990131D+00	0.905549D+01	0.400000D+01

ステージ ショウホウ ← ステージ 詳 系内性能 (タンイ シカシ:)

ステージ 1	シヨリケン スウ = 0.247533D+00	シヨリシカシ = 0.100041D+01
サーバ 3	シヨウリツ = 0.825109D-01	コリヨウ = 0.247533D+00
()	マチコ スウ = 0.101740D-03	ケイナイコ スウ = 0.247635D+00
	マチシカシ = 0.411015D-03	ホリウシカシ = 0.100000D+01
ステージ 2	シヨリケン スウ = 0.247533D+00	シヨリシカシ = 0.210358D+01
サーバ 2	シヨウリツ = 0.247533D+00	コリヨウ = 0.495066D+00
()	マチコ スウ = 0.256384D-01	ケイナイコ スウ = 0.520704D+00
	マチシカシ = 0.103576D+00	ホリウシカシ = 0.200000D+01
ステージ 3	シヨリケン スウ = 0.247533D+00	シヨリシカシ = 0.130555D+02
サーバ 1	シヨウリツ = 0.990131D+00	コリヨウ = 0.990131D+00
()	マチコ スウ = 0.224153D+01	ケイナイコ スウ = 0.323166D+01
	マチシカシ = 0.905549D+01	ホリウシカシ = 0.400000D+01

4.2 モデル理論

マルコフ1の背景理論は, Gordon & Newell (以下 G & N と略称)の, 次の平衡方程式によっている。

$$p(n_1, n_2, \dots, n_M) = \frac{1}{G(N)} \prod_{i=1}^M \frac{X_i^{n_i}}{\beta_i(n_i)}$$

$$\text{ここに, } G(N) = \sum_{n \in S(N, M)} \prod_{i=1}^M \frac{X_i^{n_i}}{\beta_i(n_i)}, \quad S(N, M) = \{(n_1, \dots, n_M) \mid \sum_{i=1}^M n_i = N \text{ \& } n_i \geq 0, \forall i\}$$

もちろん, これでは実用計算は出来ない。

実用計算を可能にしたのは(しかも計算し易い形で), Buzenであるが, ただしシングルサーバでしか提示されなかった。

このため, マルコフ1の数値計算法はBuzen流ではあるが, 筆者が導いたマルチサーバでの計算式によっている。

以下, この点について簡単に示しておく(本節はGEN, Buzenの論文内容を教数の向題もあり, 前提としている事を断っておく)。

GENの式を実用計算するためには, $G(N)$ を容易に計算出来る事が必要となる。このためには,

$$g(n, m) = \sum_{n \in S(n, m)} \prod_{i=1}^m \frac{X_i^{n_i}}{\beta_i(n_i)} = \sum_{k=0}^n g(n-k, m-1) \frac{X_m^k}{\beta_m(k)}$$

$$g(n, 1) = \frac{X_1^n}{\beta_1(n)} \quad (n = 0, 1, \dots, N)$$

$$g(0, m) = 1 \quad (m = 0, 1, \dots, M)$$

$$g(n, M) = G(n) \quad (n = 0, 1, \dots, N)$$

$$g(N, M) = G(N)$$

として $G(N)$ を順に計算すれば良い(こゝではBuzenを示してあった)。

さて, ここから性能計算に必要な量を導けば良い。以下に, 呼量と待ち行列長を示して, 本節を終える。

$$\text{ステージ } i \text{ の呼量 } h_i = X_i \frac{G(N-1)}{G(N)} \quad \left(\text{Buzenのシングルサーバの呼量(使用率)に形式的に一致} \right)$$

ステージ i の待ち行列長 Q_i : これは平易な式にならない。そこで,

$$Q_i(k) = \frac{X_i^k G(N-k)}{\beta_i(k) G(N)}$$

とおくと,

$$Q_i(N) = \frac{X_i^N G(0)}{\beta_i(N) G(N)} = \frac{\gamma_i^0}{\beta_i(0)} P_i(N) \quad (\text{以下同様})$$

$$Q_i(N-1) = \frac{\gamma_i^0}{\beta_i(0)} P_i(N-1) + \frac{\gamma_i^1}{\beta_i(1)} P_i(N) \quad (\gamma_i: \text{ステージ } i \text{ のサーバ数})$$

\vdots

$$Q_i(\gamma_i+1) = \sum_{k=0}^{N-(\gamma_i+1)} \frac{\gamma_i^k}{\beta_i(k)} P_i(\gamma_i+1+k)$$

を利用し $P_i(k)$ ($k = \gamma_i+1, \dots, N$) を求め, Q_i を計算。

ここで, $P_i(k)$ とは

$$P_i(k) = P(n_i = k) = P(n_i \geq k) - P(n_i \geq k+1)$$

$$P(n_i \geq k) = \sum_{n \in S(N-k, M)} \frac{1}{G(N)} \frac{X_i^{n_i+k}}{\beta_i(n_i+k)} \prod_{j=1}^M \frac{X_j^{n_j}}{\beta_j(n_j)} \quad \begin{array}{l} \text{ステージ } i \text{ にトランザクション} \\ \text{が } k \text{ 以上ある確率} \\ \text{(これは上平易にならない)} \end{array}$$

である(呼量 h_i は $P(n_i \geq 1)$ をもとに得るが, h_i は容易に上式の如く定まる)。

なお、シングルサーバの場合の Q_i は、Buzenは陽には示していないが、筆者の計算では次の如く平易な式となる。

$$P(n_i = k) = \frac{x_i^k}{G(N)} \{ G(N-k) - x_i G(N-k-1) \}$$

より、

$$\begin{aligned} Q_i &= \sum_{k=1}^N (k-1) P(n_i = k) \\ &= \sum_{k=2}^N x_i^k \frac{G(N-k)}{G(N)} \end{aligned}$$

となる。

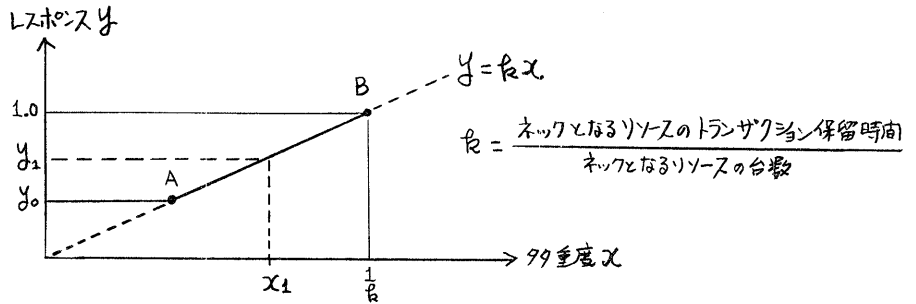
5 まとめ(定性的意義)

以上、分散構造システムの応答解析を、クローズドキューイングモデルによるマルコフ1にて行なった結果を述べた。

この結果から導出される一般論的結論としては、次の二点に集約できよう。

① 応答性について、特に分散構造か否かを問う必要はない(分散構造か否かは本質的でない)。

② マルコフ1による応答解析は、次の如くの定性的意義を与えた。



図の如く、 ρ 重度 x_1 の時のレスポンスは y_1 (近似可)。

(y_0 は無競合時レスポンス。)

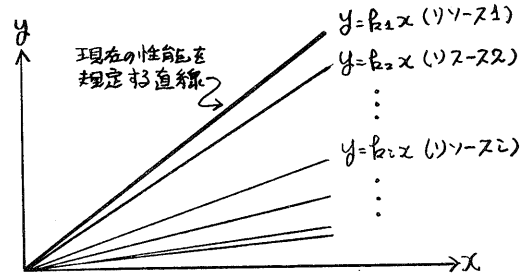
性能(スループット)とレスポンスのバランス点(最適点)は、図のA点へB点向に見出される。

(スループット θ は、 $\theta \equiv x/y$ であるが、 $x \rightarrow B$ 点で $\theta \rightarrow x/y$ 。)

分散化(負荷分散を含む)は、上記 k を小にする方向に働かせる事になるが、これは通常の性能向上方策(マルチプロセッサ化ほか)と、本質的に異なるものではなく方策の手法の問題と解釈できよう。

また②は、性能向上を行なう場合、次の如くの配慮の必要性を示唆するのに有効であろう。

右図の如く、性能向上を行なう場合(k_1 を小にする場合)、他のリソースの性能改善も同時に行なう必要があるか如何かの判定に便宜を与えよう。(k_1 を k'_1 に向上した時、 $k_2 > k'_2$ ならば向上の期待効果が得られない。)



6 おわりに

マルチプロセッサは、必要にして複雑な性能計算を容易にする事により、計算能力の向上を促した数値計算パッケージである。

この種の数値計算パッケージを豊富に与える事が一層望まれるものと思われるが、同時に必要な事は、

- ① まず暗算可能な近似式を提示する事
- ② 定性的な結論、傾向を導く事
- ③ そして、常識の境を向上して行く事。
演算・改造に理論付けを行なって行く事

すなわち、法則の導出・附与であろう。本稿がこのために、多少の示唆を与えたものであるなら、望外の幸せである。

おわりに、本稿の発表の機会を与えていただいたオ-公共システム事業部システム部・枝本部長、天野部長代理、小泉課長、および本稿の執筆にあたり、時間的余裕を賜った諸氏、有益な助言をいただいた諸氏に感謝いたします。

[参考文献]

- 1) Gordon & Newell "Closed Queuing System with Exponential Servers"
- 2) Buzen "Computational Algorithms for Closed Networks with Exponential Servers"
- 3) 堀-誠也 "性能評価ツール QM-1について", 情報研究会・システム性能評価, 1977.9.9
- 4) 近藤康男 "TSS/BATCH混在システムの一解析" 51年度オ-情報全国大会予稿集
- 5) 雁部鏡一 "電話交換トランスラック" オ-ム社