

統合運用ソフトウェア JP1 の性能評価

渡辺 聡 鈴木 芳生 長澤 幹夫

(株)日立製作所中央研究所

〒185-8601 東京都国分寺市東恋ヶ窪 1-280

E-mail:{wata-s,yosuzuki,m-nagasa}@crl.hitachi.co.jp

近年マルチタスク環境の普及により、複数のソフトウェアが同時に実行される環境が一般化している。このような環境では、複数のソフトウェアの動作が互いに関係するため、ソフトウェアの内部挙動がシステム性能に与える影響が大きくなる特徴がある。そのため、シミュレーション用のソフトウェア性能モデルにおいてもソフトウェアの内部挙動を反映させる必要が生じている。本報告では、OS がモニタする消費リソース情報とプロセス間通信ログをもとに、ソフトウェアのプロセスレベルでの内部挙動を再現する性能モデルの作成手法を提案し、本モデリング手法による JP1 性能シミュレーション実施例を示す。

Performance analysis of system management software JP1

Satoru WATANABE Yoshio SUZUKI Mikio NAGASAWA

Central Research Laboratory, Hitachi Ltd.

1-280 Higashi-Koigakubo, Kokubunji-shi, Tokyo 185-8601, Japan

E-mail:{wata-s,yosuzuki,m-nagasa}@crl.hitachi.co.jp

A lot of software can work concurrently and software inner behavior influences each other on multi-task computers. Software models for performance simulation have to reflect software behavior. In this paper we propose a method to make software model, which reflects software process behavior, based on OS monitoring data and process log data. We show the cases of JP1 performance simulation.

1. 背景

JP1¹⁾は、企業情報システムのジョブ管理、ネットワーク管理、アプリケーション管理などを支援する統合運用管理ツールである。近年、金融機関の提携や統合によるシステムの大規模化、インターネットによるビジネス形態の変革、ASP や iDC などへのアウトソーシング化等、情報システムの大規模統合が急速に進んでいる。JP1 のジョブ管理機能 AJS2 (Automatic Job management System²⁾) も、Web システムを通したトランザクション処理に用いられるようになってきている。Web システムでは、ジョブ処理の短時間化と単位時間あたりのジョブ配信件数の増加が特徴となっており、AJS2 自身の処理時間がシステム全体の処理時間に対して無視できない可能性がある。このため、

2000 年 6 月の JP1 Version6 のリリースに先立ち、AJS2 のジョブ配信性能を事前評価することとした。

AJS2 のジョブ配信性能を評価する方法としては、プログラム実行ステップ数の積み上げや、実システムでの実測が考えられる。しかしながら、積み上げ計算はジョブの配信処理が複数個同時に実行されたときに発生する遅延が正確に見積もれないため、単位時間あたりのジョブ配信件数の増加を想定した性能評価には適さない。また、実測だけでは AJS2 サーバのスペック変更に対応して、再評価を行う必要があるため現実的とはいえない。

一方、情報システムの性能を定量的に予測する方法として、システム性能シミュレーションが注目されている。システム性能シミュレーションでは、性能予測対象システムの構成部品で

あるハードウェア、ソフトウェアの動作、ユーザからのサービス要求などを計算機上にモデル化し、待ち行列のイベントシミュレーションを行なう。これにより、ジョブ配信件数が増加した場合のような、複数の処理が同時に実行される状況を対象とした性能予測も可能となる。

2. モデリング対象機能

AJS2 によるジョブ配信は、マネージャとエージェントの二つのソフトウェアによって制御される（図1）。マネージャは、ユーザが定義したジョブ情報を予めデータベースに格納し、ジョブ情報に従ってジョブの実行をエージェントに指示する（図1 - ）。エージェントはマネージャからの指示に従いジョブを実行し、ジョブ終了をマネージャに通知する（図1 - ）。ジョブ管理機能はこれを繰り返すことによって、ユーザが定義したジョブの実行を自動化している。

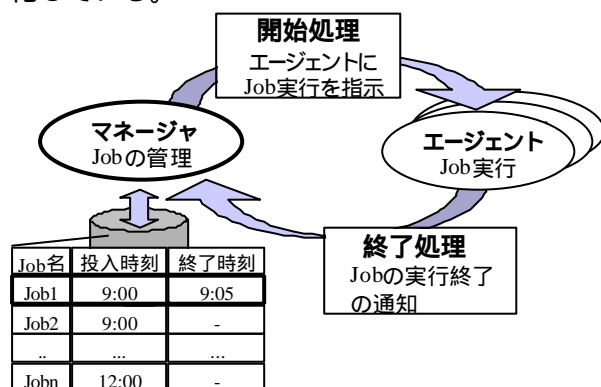


図1 ジョブ配信の概要

本研究では頻繁に利用される次の2つの処理をモデル化の対象とした。

- 開始処理（マネージャからエージェントへのジョブ実行指示を行なう動作）
- 終了処理（マネージャがエージェントから終了通知を受け取る動作）

ジョブ開始処理、終了処理に関する AJS2 内部のプロセスの関連図を図2に示す。括弧内にプロセスの大まかな機能を示し、開始処理、終了処理の流れをそれぞれ、図2の太矢印(1)、(2)に示した。ジョブ開始処理は Ajsflowd（ジョブネット解析）プロセスがデータベースにアクセスすることから始まり、Ajssubd（開始通知）、Jpqman（ジョブ投入）プロセス内の処理を経て、Jpqagt（ジョブ実行）プロセスがジョブを開始することで終了する。終了処理は

Jpqagt プロセスがジョブの終了を Jpqman プロセスに通知することから始まり、Jpqnfyfd（終了通知受付）、Ajssubwd（終了通知）プロセス内の処理を経て、Ajsflowd プロセスがデータベースにジョブの終了を書き込むことで終了する。

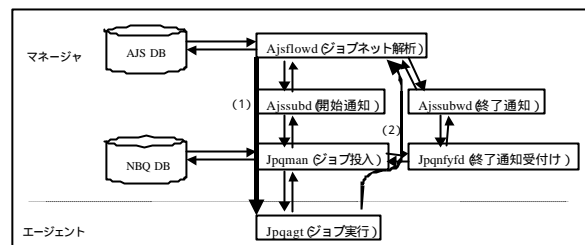


図2 プロセス関連図

3. シミュレーションモデルの作成

3.1 性能シミュレータ

本研究では、システム性能シミュレータに、業界トップシェアである米国 HyPerformix 社の Strategizer²⁾を用いた。

Strategizer は、システムを構成するクライアント、サーバネットワーク、データベース、アプリケーションのスペックや要求負荷を定義することにより、システム動作シミュレーションを行うシミュレータである。本節では、Strategizer のシミュレーションモデル作成方法について述べる。図3に Strategizer によりモデリングされたシステム構成例を示す。Strategizer では、次の手順でシミュレーションモデルの対象とするシステムを記述する。

- (1) システムを構成するクライアント、サーバ、データベースなどの要素と、その間のネットワークを GUI を用いて定義する。
- (2) ハードウェアの性能パラメータ（CPU の MIPS 値、メモリ量、ネットワーク帯域幅など）を定義する。
- (3) 各クライアント、サーバに対し、その上で動作するアプリケーションプロセスを定義する。各アプリケーションは、サービス要求、またはサービス処理などの処理内容、リソース（CPU、メモリ、ディスクなど）消費量を記述するアプリケーションモデルにより記述される。

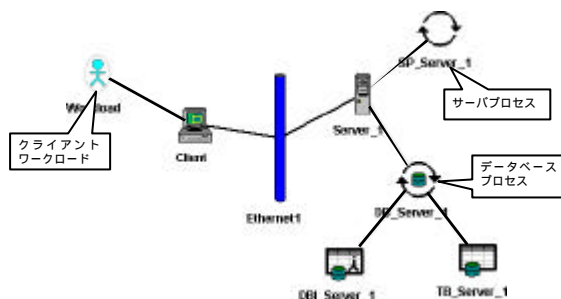


図 3 Strategizer によるモデリング例

アプリケーションモデルの作成では、各処理に対するリソース（CPU、メモリ、ディスク）消費量を記述するため、これらのメトリックの採取が必要である。

3.2 モデルメトリックの見積もり

シミュレーションに必要なメトリックを得るため、テストジョブを AJS2 のマネージャに登録し、NT パフォーマンスモニタにより開始処理及び終了処理実行時のリソース（CPU、メモリ、ディスク）消費量の測定を行った。図 4 のテストジョブを実行した場合、マネージャは開始処理を実行し、5 秒後に終了処理を行う動作を 10 回繰り返す。測定時に AJS2 が行う処理以外の負荷がかからないようにするため、図 4 に 5s と示すジョブは 5 秒スリープ後、処理を行わずに終了するジョブとした。

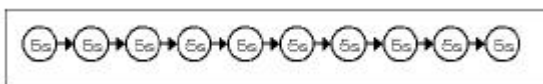


図 4 テストジョブ

WindowsNT 付属のパフォーマンスモニタの表示によるテストジョブ実行中の CPU 使用率の推移を図 5 に示す。グラフ内の各線はプロセスの CPU 使用率を表わしている。パフォーマンスモニタは各プロセスの CPU 使用率を 1 秒単位で平均して出力するため、プロセスの動作が複数のサンプル周期にまたがった場合には低く表示される。

測定された開始処理及び終了処理実行時のリソース消費量には 5 % 程度のばらつきがみられる。この原因には、使用したモニタ精度の限界や OS による負荷のばらつき等が考えられる。しかしながら、5 % 程度のばらつきは、システム設計時に用いられるソフトウェアのリソース消費見積もり式の誤差係数 30 % ~ 50 % と比べて小さいため、モデルでのばらつ

き再現は行わず、複数回の測定を実施し平均値をモデル作成に用いた。

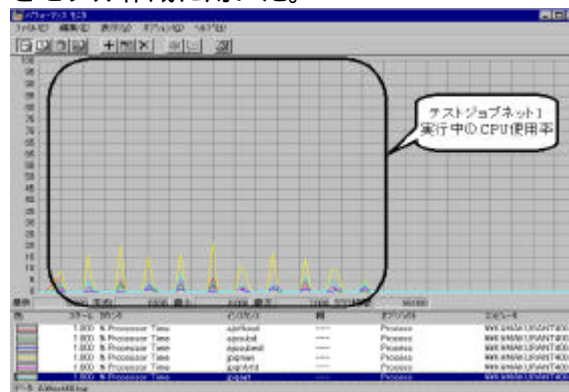


図 5 CPU 使用率の推移

3.3 ログ情報の組み込み

前節のメトリックの見積もりにより、性能モデル作成が可能となる。しかしながら、OS がモニタする消費リソースに関する情報だけでは、ソフトウェア内部の挙動がわからない。そのため、本研究では AJS2 が出力するログのなかからプロセス間通信部分を抜き出し、プロセス動作のダイアグラムを作成した（図 6）。図 6 では、矢印でプロセス間通信を表し、太線でプロセスが動作している時間帯を示している。これをもとに、各プロセスの実行順序や連携のタイミングをモデルに組み込むことで、各プロセスが持つスレッド数をモデルに反映させ、モデルの挙動を実際のソフトウェアの挙動に近づけた。

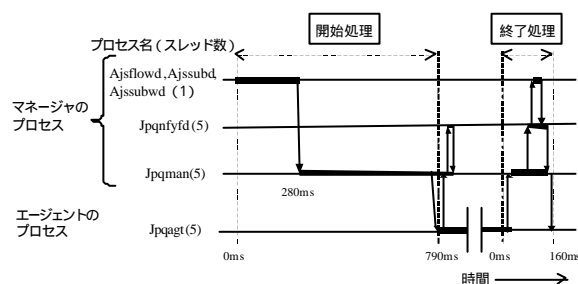


図 6 プロセスの処理ダイアグラム

3.4 性能モデルの検証

AJS2 マネージャが、複数個の開始処理を同時に行う並列度を変化させる状況の実測とシミュレーションを行い、ジョブ開始までにかかる処理時間の平均を比較した。メトリックの見積もりだけを用いて作成した性能モデルを「モニタ連携モデル」、メトリックの見積もりに加

えてログ情報を組み込んだ性能モデルを「ログ連携モデル」として図 7 に結果を示す。

モニタ連携モデルでは、図 6 のダイアグラムに示したプロセスの実行順序やスレッド数が反映されていないため、並列度の増加に伴って実測とのずれが大きくなる傾向が見られる。一方、モニタ連携モデルでは実測とのずれの増加が抑えられていることがわかる。

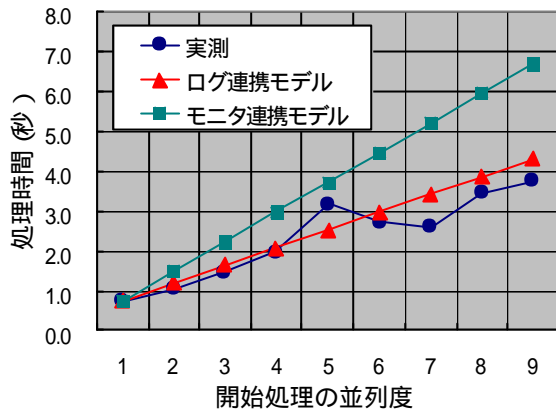


図 7 モデルの検証結果

4. シミュレーション実施例

4.1 限界性能予測

AJS2 サーバに搭載 CPU : Pentium III 500MHz、搭載メモリ 128MB のスペックを想定し、1 分あたりの投入ジョブ数を変化させ、ジョブの処理完了率(完了したジョブ数 / 投入したジョブ数)をログ連携モデルによるシミュレーションで予測した。一定の間隔でジョブを開始する場合の結果を図 8 に示した。この結果は、前述の仮定のもとで AJS2 が毎分約 130 ジョブまで、ジョブの処理完了率 100%の安定運用が可能なことを示している。但し、シミュレーション結果はジョブ投入間隔等によって影響を受けるため、システム的设计支援に用いるためには、顧客の利用状況を反映したシミュレーションが望ましい。

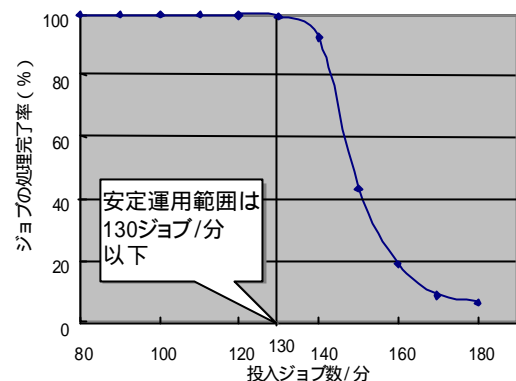


図 8 ジョブ管理機能の評価結果

4.2 サーバスペックの変更

AJS2 サーバの MIPS 値、ディスクシークタイムを変更し、1 つのジョブ投入に必要な処理時間の変化を調べた結果を図 9、図 10 に示す。MIPS 値の変更と比較して、ディスクシークタイムの変更が大きく性能に影響することがわかる。Starategizer を用いたシミュレーションでは、実機では困難なハードウェアの変更が容易に行えるため、システムのコスト対性能の評価を容易に実施可能である。

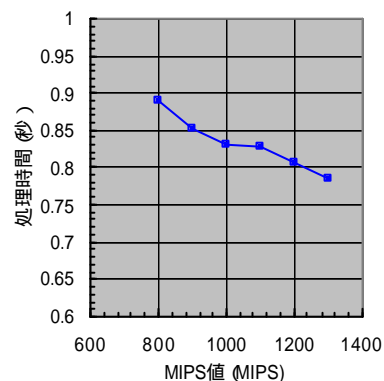


図 9 CPU 性能の影響

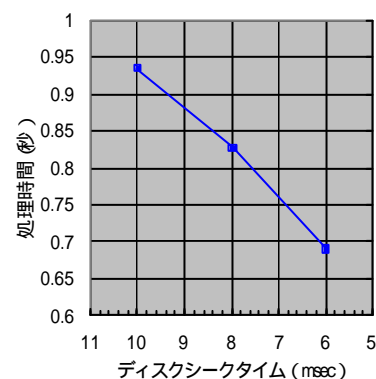


図 10 ディスク性能の影響

5 . まとめ

本研究では、システム構築時の設計支援を目的として、統合システム運用管理ソフトウェア JP1 ジョブ管理機能の性能モデルを作成した。OS のモニタ情報だけを用いたモニタ連携モデルと、OS 情報に加えてソフトウェアのログ情報を用いたログ連携モデルの 2 つのモデルを作成し、それぞれのモデルを用いたシミュレーション結果と実測の比較を行った。

シミュレーションの実施例として、ジョブ管理機能の限界性能(一定時間に配信可能な最大ジョブ数)及びジョブ管理を行うサーバのスペック変更が性能に与える影響をシミュレーションにより予測できることを示した。

6 . 今後の課題

ソフトウェアのモデル化では、プロセスの挙動だけでなく、レジスタ使用やメモリ確保等のより詳細な内部挙動のモデル化も考えられる。性能モデル作成手法の開発とともに、モデルに組み込むソフトウェアの内部挙動と性能モデルの予測精度との関係を明らかにする研究が必要である。

7 . 参考文献

- 1) <http://www.hitachi.co.jp/Prod/comp/soft1/jp1/index.html>
- 2) <http://www.hyperformix.com> (日本語情報は <http://www.best.co.jp>)