社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－ARC－176 (1)
2008－EMB－ 7 (1)
2008／1／15

機能・性能シミュレーション連携方式による性能予測手法

関誠司† 尾崎敦夫† 古市昌一† 角田直之† 渡部修介†
三菱電機株式会社†

多数の異種システム同士の連携により構成する大規模で複雑なシステムである SoS（System of Systems）を開発する機会が増えている。大規模で複雑なシステムを構築するにあたり、性能に対する問題は開発のできるだけ早い段階に対処しないと、問題の対策に要するコストは増すばかりである。そこで本稿では、システム開発の上流設計段階で目標性能に対するアーキテクチャのボトルネックを早期に見つけ、それを改善することにより後戻りの少ない効率的な開発を行うことを目的とした、機能・性能シミュレーション連携方式による性能予測手法と、その評価について報告する。

A performance prediction technique by co-operative simulation method of a functional simulator and a performance simulator

Seiji Seki† Atsuo Ozaki† Masakazu Furuichi† Naoyuki Tsunoda† Shusuke Watanabe†
Mitsubishi Electric Corporation†

The chance to develop SoS(System of Systems) that is a large-scale, complex system that composes by the cooperation of a lot of different kind systems has increased. The cost that development requires for the measurement of the performance issues goes up if not dealing at the early stage. This paper reports the performance prediction technique and the evaluation by the method for co-operation of a functional simulator and a performance simulator to aim to do efficient development that going back is few.

1. はじめに

近年、大規模防災システムに代表される、多数の異種システム同士の連携により構成する大規模で複雑なシステムである SoS（System of Systems）を開発する機会が増えている。[1]またこのようなシステムにおいては、機能の増加にともないハードウェア／ソフトウェアともに規模が拡大している。特に、機能／性能の向上を目的として、並列処理実行を可能とする構成を適用する場合が増加し、またリアルタイム性の確保や処理性能に対する要求は高まっている。

従来、システム構築後に処理時間やシステム構成などのアーキテクチャに関する精度の高い性能評価を行っていたが、その段階でシステム全体の問題が判明してからソフトウェアによる性能チューニングや、ハードウェアを含めた設計の見直しを行うことはシステム規模や機能の増加につれ困難になってきた。システム構築における性能の問題は、開発のできるだけ早い段階に対処しないと、問題の対策に要するコストは増すばかりである。そこでシステム開発の上流設計段階で目標性能に対するアーキテクチャのボトルネックを早期に見つけ、それを改善することにより後戻りの少ない効率的な開発を行う必要がある。

2. 課題

システム開発の上流設計段階でシステム性能の予測を行うにあたり、経験豊富なシステムアーキテクトによる経験則や、機能の静的な処理時間やデータの転送時間を見積もることによりシステム全体の性能を予測することは、システムの規模拡大や機能の複雑化により困難になってきている。また、命令レベルでシミュレーションを行うことにより性能を予測する手法[2]などが提案されているが、性能評価にあたりシステムの構想段階では準備が困難な実プログラムを用意する

などのモデル作成コストや、シミュレーション時間がかかるなど課題がある。

上流設計段階でシステム性能を予測するにあたっては、モデル作成やシミュレーション実行に対するコストが低く、高い予測精度を保ち、OS等のプラットフォーム環境に依存しない汎用性をもった性能予測手法が望まれている。しかしながら、予測精度とモデル作成等によるコストは、両立することは難しくトレードオフの関係になっている。

このような課題に対し、我々はシステムの構想検討段階における概略レベルでの性能予測を行うにあたり、簡便性や予測精度を保つ工夫を考えた機能・性能シミュレーション連携方式による性能予測手法を提案している。[3]

本提案方式では、システムの機能検証シミュレーションと連携して性能評価を行うことによりモデル作成のコスト抑える工夫をしている。また、計算機で実行にかかる処理時間を理論値ではなく実測値を用いることや、処理間の競合に対して待ち行列モデルによるシミュレーションを行うといったことにより予測精度を確保している。

3. 提案手法

3.1. 性能モデル構築手法

性能を予測するにあたり、評価対象システムのモデル化を行う。S/W、H/W 設計の基礎となる情報をモデル定義し、そのモデルを利用して処理時間の見積りを行う。評価対象システムのモデル化にあたっては、構成要素として S/W 機能（プロセス）、H/W モジュール（H/W 機能）、チャネル（通信路）によるモデル化を行う。

はじめに、評価対象システムにおける複数の S/W 機能の流れと連携を示す処理フロー図を作成し、以下の表 1 に示した各モデルの作成を行う。

表 1　システムのモデル化

モデル	説明
処理ﾌﾛｰ図	複数の S/W 機能の流れと連携を示す
S/W ｼｰｹﾝｽ図	処理フロー図における S/W 機能のシーケンシャルな動作を示す
H/W ﾌﾞﾛｯｸ図	S/W 機能を実行する H/W モジュールやチャネル間の関係を示す
性能ﾊﾟﾗﾒｰﾀ情報	S/W 機能や H/W モジュールにおける予測処理時間算出のための情報

性能パラメータ情報においては、性能見積りを行うための S/W 機能の処理量や、H/W モジュールの仕様などの処理時間を予測するための基礎となるベース性能値に対する情報を集積する。集積にあたっては、以下の集積方法がある。

・H/W データシートからの動作想定による算出
・実測、ベース性能値からの換算による算出
・処理アルゴリズムに基づく命令数による算出
・順次処理、周期処理別の算出

評価対象システムに対してこのようなモデル化を行い、これらのモデルを用いて我々は機能・性能シミュレーション連携方式による処理時間の性能予測を行う。

3.2. 機能・性能シミュレーション連携方式

機能・性能シミュレーション連携方式は、機能検証シミュレーションと性能評価シミュレーションを分離実行可能としつつ、且つ、連携した動作が可能なシミュレーション方式である。図 1 に本方式の構成を示す。

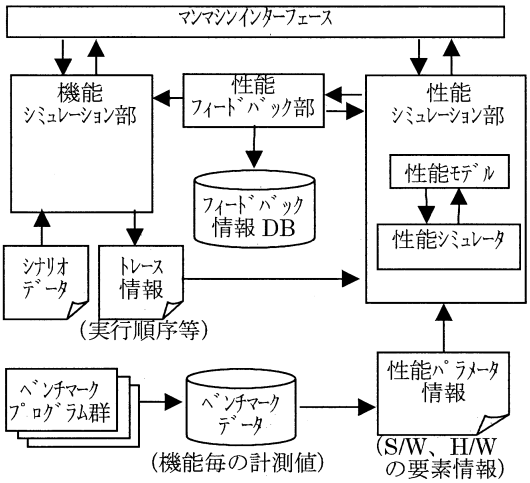

図 1　機能・性能シミュレーション連携方式

本方式では、評価対象システムに対して機能検証シミュレーションを実施した際に、システムのモデル化で示した S/W シーケンス図に該当するトレース情報を取得する。例えば、一般的な評価対象システムの処理フロー図を図 2 に示す。機能検証シミュレータにおいて、計測データを入力すると入力した計測データに従い、機能ブロックにおける S/W 機能が実行され、次の機能ブロックに遷移し実行するといった動作が、制御指示まで繰り返される。この際に機能ブロック間の遷移情報をトレース情報として取得する。

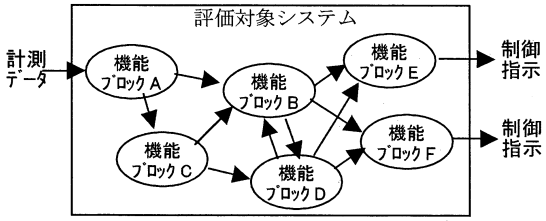

図 2　処理フロー図

本方式では、各機能ブロックで必要とされる処理量を計測用のベンチマークプログラムを用いて、様々な H/W プラットフォーム上で計測しておく。それらの計測した情報をベンチマークデータとして、システムのモデル化で示した性能パラメータ情報とする。計測用のベンチマークプログラムは、各機能ブロックの処理を模擬したもの、または実物そのもののプログラムとする。

各機能ブロックをどの H/W モジュールで実行するかの指定や H/W モジュール間の接続を示した H/W ブロック図をもとに、トレース情報と、ベンチマークデータからなる性能パラメータ情報を入力する性能シミュレーション可能な性能モデルを作成する。

性能評価シミュレーションでは、各機能ブロックの処理を行う際にリソース競合する場合の競合時間を見積もることができる、待ち行列モデルによるシミュレーション実行可能な性能シミュレータを用いる。

この性能モデルと、性能シミュレータを用いて性能評価シミュレーションを実行することにより、処理時間の予測結果を得ることができる。

本方式では、性能予測の精度を高める方法として、次のような手法も備えている。機能ブロックでは入力データに対して S/W 機能が実行されるが、機能ブロックによっては入力データにより機能ブロック内で条件分岐し、処理内容が異なることが考えられる。その場合、ベンチマークデータは双方の処理に対する処理時間の平均値に近い値となることが予想される。処理内容の違いから処理量が大きく異なった場合は、平均値に近いベンチマークデータでは処理時間の見積もり誤差が大きくなる。これを改善するために、入力データにより機能ブロック内で分岐する機能ブロックに対しては、分岐ごとのベンチマークデータを計測する。また、機能検証シミュレーションにおいて、分岐ごとに分岐パターン識別子をトレース情報に出力することにより、トレース情報からどのベンチマークデータかを判別できるようにする。このように分岐別にベンチマークデータを取得することにより予測精度を高めている。

上記以外に機能ブロックによっては、入力データ以外の情報により処理時間が大きく変わるケースが考えられる。例えば検索機能においては、登録された複数のデータの中から入力データに対して検索する場合、入力データそのものではなく検索範囲であるデータの登録数によって検索にかかる処理時間が決まる。このようなケースに対応するために、上記事例のデータ登録数のような情報を含めた算術式を導き、これによりベンチマークデータを算出している。

4.　提案手法の評価

4.1.　評価対象システム

実システムの一部の動作を簡易に抽象化した図 3 に示す模擬アプリケーションシステムを用いて、機能・性能シミュレーション連携方式による性能予測手法を評価した。模擬アプリケーションシステムは、周期的に各種センサーからの複数個のデータを入力し、それら各データに対していくつかの条件に合致するか判断し、最終的にある条件に合致したもののみを出力・表示するシステムである。このシステムではブロック＃2 において、ある条件により順次処理されるものと、周期処理されるものに分かれるため、異なる処理が混ざることにより処理時間の予測が難しくなると考える。表 2 に対象システムである模擬アプリケーションシステムの動作概要を示す。

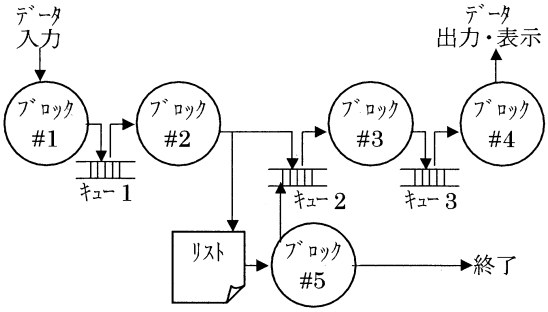

図 3　模擬アプリケーションシステム

表 2　対象システムの動作概要

機能ﾌﾞﾛｯｸ	説明
ﾌﾞﾛｯｸ#1	周期的に送られる複数個のデータを受信しキュー1 に入れる
ﾌﾞﾛｯｸ#2	キュー1 を監視し、ある条件でデータをリストに追加するかキュー2 に入れる
ﾌﾞﾛｯｸ#3	キュー2 を監視し、条件に応じた処理を行いキュー3 に入れる
ﾌﾞﾛｯｸ#4	キュー3 を監視し、データを出力・表示する
ﾌﾞﾛｯｸ#5	周期的にリストを読み出し、条件に合致するデータをリストから削除してキュー2 に入れる

4.2.　システムのモデル化

模擬アプリケーションシステムの性能を予測するにあたり、システムのモデル化について述べる。処理フロー図は、図３の模擬アプリケーションシステムの図に該当する。S/W シーケンス図は、模擬アプリケーションシステムに対する機能検証シミュレーションから取得するトレース情報に該当する。トレース情報には、ブロック＃1 からブロック＃4 やブロック＃5 までのシーケンスや、ブロック＃5 からブロック＃4 までのシーケンスの情報が含まれている。

今回評価するH/Wモジュールとしては、1CPU コアの計算機と、2CPU コアの計算機を用いて評価を行った。2CPU コアのケースでは、各機能ブロックをどちらの CPU コアで実行するか指定しない場合と、機能ブロックごとにどちらの CPU コアで実行するか指定した場合とで評価を行った。２CPU コアのケースにおいて、CPU コア指定しない場合と、CPU コア指定した場合の H/W ブロック図を図 4、図 5 に示す。

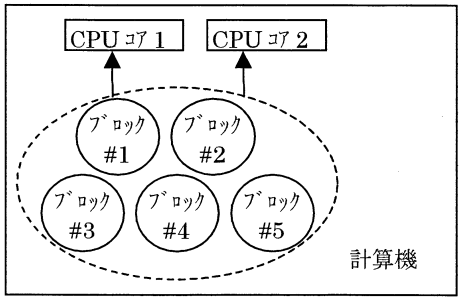

図 4　コア指定なし

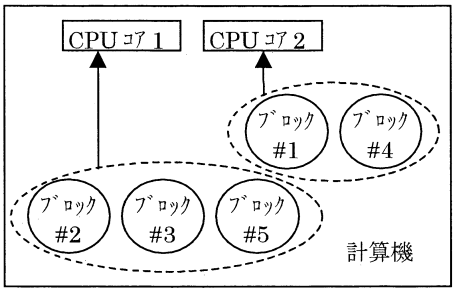

図 5　コア指定あり

性能パラメータ情報は、模擬アプリケーションシステムのブロックごとに、そのブロックのみを実行するベンチマークプログラムを用いて、今回評価する H/W モジュールの H/W プラットフォーム上で計測したベンチマークデータに該当する。1CPU コアの計算機と、2CPU コアの計算機において、ブロックごとのベンチマークプログラムを実行しベンチマークデータを取得した。

模擬アプリケーションシステムのベンチマークデータにおいては、ブロック＃2、＃3、＃4、＃5 に対しては、機能ブロック内の分岐条件ごとにベンチマークデータを計測した。またブロック＃5 においては、リストの登録データ数をもとに算出したベンチマークデータとした。

以上のようにシステムのモデル化定義を行い、H/W ブロック図をもとに、トレース情報と、ベンチマークデータからなる性能パラメータ情報を入力する性能モデルを作成し、性能評価シミュレーションを実行することにより、性能予測を実施した。

4.3.　評価の実施

前述した提案手法により求めた性能予測結果と、実機において性能を実測した実測結果を比較することにより、提案手法による性能予測結果の予測精度を評価した。評価する計算機の構成としては、1CPU コアの計算機と、2CPU コアの計算機のコア指定なしと、コア指定ありの３つのケースについて評価を行った。実機での性能実測にあたり、今回用いた 1CPU コアの計算機と、2CPU コアの計算機の実機評価環境を表３、表 4 に示す。

表 3　1 CPU コアの計算機評価環境

計算機	Dell OptiplexGC
CPU	Intel PentiumⅢ (933MHz)
メモリ	512MB
OS	Linux ｶｰﾈﾙ 2.6

表 4　2CPU コアの計算機評価環境

計算機	EPSON DIRECT MR2100
CPU	Intel Pentium D (3GHz)
メモリ	3GB
OS	Linux ｶｰﾈﾙ 2.6.11

各計算機構成に対して、模擬アプリケーションシステムを実装し、周期的に合計 20 個のデータを入力しプログラムの実行処理時間を測定した。各データに対する処理時間を測定し、それらの平均値を実測結果とした。同様に各データに対する処理時間を予測し、それらの平均値を予測結果とし、以下の計算式により予測精度を求めた。表 5 に評価結果を示す。

予測精度[%]＝｜予測結果－実測結果｜／実測結果×100

表 5　予測精度の評価結果

計算機構成	予測精度
1CPU コア	7.9%
2CPU コア指定なし	22.5%
2CPU コア指定あり	7.3%

システムの構築にあたっては、性能的な処理能力に対してある程度の余裕を持たせる。50%程度の余裕を持たせることを想定した場合において、今回の評価結果における予測精度は 20%～30%内に収まっており、本提案方式の有効性が確認できた。従って、本方式によりどの計算機構成を選択するか判断することができるものと考える。なお、2CPU コア指定なしにおける誤差要因としては、複数のタスクが必ずしも複数の CPU コアに割当てられ並列に処理されないなどプロセススケジューリングによる誤差が大きくなることがわかった。

システムのモデル化及び性能モデルの作成、性能評価シミュレーションまでの一連の作業量については、主にベンチマークデータの取得と性能モデルの作成及びモデルの検証に時間を要した。機能検証シミュレーションとの I/F 用のモデルやベンチマークプログラムの雛型など、共通的なモデルやプログラムを流用できる程度にもよるが、今回のシステム例であればおおよそ 1 週間以内には可能な作業量と考える。性能モデルが正当なものか検証する作業の効率化が課題である。

性能評価シミュレーションの実行速度に関しては、今回用いた 1CPU コアの計算機相当の性能を持つ計算機で、おおよそ 1 分以内で高速にシミュレーション実行できた。

5. おわりに

本稿ではシステムの上流設計段階において性能予測を行う、機能・性能シミュレーション連携方式による性能予測手法について述べた。またその評価においては、複数の計算機構成に対して模擬アプリケーションシステムを実装した際の性能を本提案方式により予測し、その有効性を確認した。今後さらに本方式の適用評価を進めるとともに、簡便さを保ちつつ、より予測精度の高い手法にしていくことが今後の課題である。

参考文献

[1] M. Furuichi et. al, "MARS: A M&S Framework for Large Scale Simulations Based on the HLA", Proc. of the 2005 Fall Simulation Interoperability Workshop, 2005.

[2] 木村, など：″スーパスカラプロセッサ設計支援ツールの構築とその適用事例″, 電子情報通信学会 論文誌 1999.

[3] 鶴, など：″大規模で複雑なシステム開発支援のための機能・性能シミュレーション連携方式の提案″, 計測自動制御学会 SI2006 (2006).