

Hybrid Simulator における性能実測とその課題

6K-4

青野 博 荒野 高志

NTT ソフトウェア研究所

1 はじめに

プロトタイピングの新しい手法として, [荒野 93a] は Hybrid Prototyping を提案し, [荒野 93b][青野 93] はこのプロトタイピングをサポートするシミュレータ Hybrid Simulator の設計, 試作を行なった. この Hybrid Prototyping 手法の特徴の一つに, 状態遷移のシミュレートと部分的にインプリメントされたサブシステムの実行の混在させることができる. これを利用しインプリメント部の性能実測を行なうことができる.

一方, [Smith90] は Software Performance Engineering(SPE)として, 開発早期からの性能解析/見積りを行なう方法論を提案している. 本稿では, Hybrid Simulator によって得られる性能実測結果を SPE のソフトウェア実行モデルのためのパラメータとして使うというアイデアをもとに, Hybrid Simulator の SPE への適用とその方法および課題について述べる.

2 Hybrid Simulator

Hybrid Simulator は, シミュレータプロセスと実プログラムプロセスからなる(図1を参照). シミュレータプロセスでは, 状態遷移で記述された仕様をシミュレートする. また, メッセージを送った先のオブジェクトがインプリメントされていた場合には, 実プログラムプロセスへそのメッセージをフォワードする. 実プログラムプロセスでは, 受けとったメッセージから対応するメソッドを実行し, その結果をメッセージとしてシミュレータプロセスへ返す. この時, プロセス間のインタフェースであるオブジェクトハンドラは仕様レベルのメッセージと属性をインプリメントレベルの変数と値に変換を行なう.

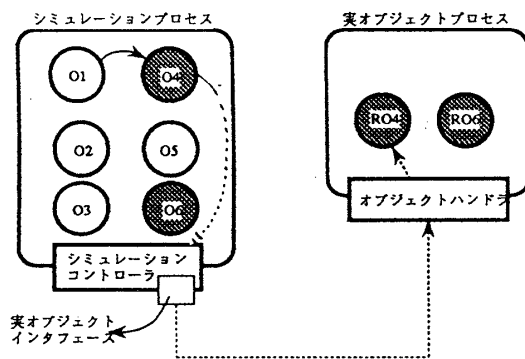


図 1: Hybrid Simulator の実行イメージ

Performance Measurement and Problem for Hybrid Simulator.
 Hiroshi AONO(aono@nttslb.ntt.jp),
 Takashi ARANO(arano@nttslb.ntt.jp),
 NTT Software Laboratories.

仕様記述とインプリメントされたものが混在して動く特徴を利用することにより, ユーザインタフェース部分を先にインプリメントして, 実際にユーザに使用してもらい, その動作の確認などを行なうことができる. また, 実コードが動作するため, 実際の性能の測定も可能である.

3 性能評価

3.1 従来の性能評価

従来の性能評価(見積り)は, 開発のテストフェーズのシステムまたは, 現在運用中のシステムの実測データを基にシステム実行モデルを作る. そのモデルに対し, 各種パラメータを変更することによって, 性能を見積もる. この方法は実データをもとにモデルを作るために, 精度の高い性能の見積りが可能であるが, 要求された性能を達成するためのシステムの変更/修正への後戻りが大きい.

3.2 SPE

一方, [Smith90] によって提案されている SPE は, ソフトウェア開発の初期段階からライフサイクルを通してソフトウェアの性能を予測するための方法を提案している.

具体的には, Performance Walkthrough と呼ばれるカスタマや開発者やユーザによるレビューを行ない, 要求仕様書からソフトウェア実行モデルのためのデータを収集する. このソフトウェア実行モデルに必要なデータとして, 以下のものがあげられている.

性能目標 (Performance Goals)

ワークロード仕様書 (Workload Specification) ワークロード仕様書は, システムの使い方や要求される機能をあらわしたもので, これらの典型的な組合せ, またこれらが実際に使われる割合などを示す. 各ワークロードは, シナリオと呼ばれる.

ソフトウェア実行構造 (Software Execution Structure) ソフトウェア実行構造は, 各ワークロードシナリオ毎の実行パスを定義するものである.

実行環境 (Execution Environment) システムが実行するハードウェア構成のことであり, ワークロードシナリオの性能は, 実行環境に依存する.

リソースの使用法 (Resource Usage) リソース使用法は, 各ワークロードシナリオに対してデータを集める. モジュール内にあるリソースアクセスの命令の数ではなく, 実行される命令の数を評価する.

これらのデータを基にソフトウェア実行モデルをグラフ解析し, システムの実行モデルのためのパラメータを収集する. このパラメータを基にシステム実行モデル(キューイングネットワークモデル)を解析し, 性能を見積もる(図2aを参照).

この方法の性能の見積りの正確さは、システム実行モデルのパラメタの正確さに依存する。しかし、SPEにおいてこれらは机上で得られるデータであり、そのデータを基に得られたパラメタであるため、実測データに比べて精度は落ちる。すなわち、開発早期からの性能見積りは可能であるが、精度は従来のものよりも劣ると思われる。

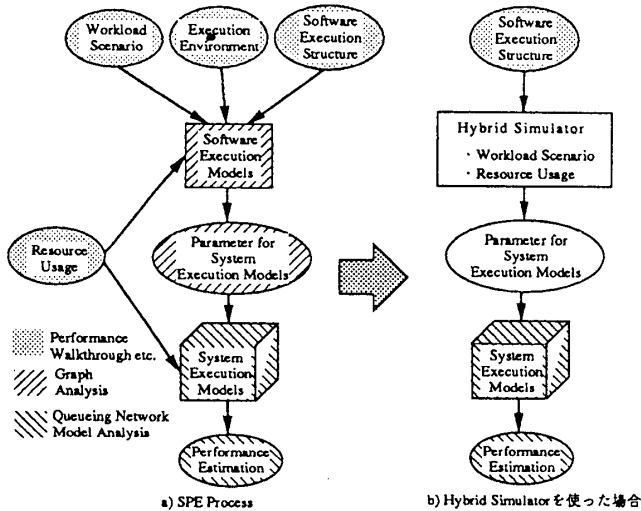


図 2: SPE へ Hybrid Simulator の応用

3.3 Hybrid Simulator による性能見積り

上記の問題を解決するための方法として、Performance Walkthrough によるデータとソフトウェア実行モデルの代わりに Hybrid Simulator を使用する (図 2 b を参照)。

Hybrid Simulator は、状態遷移のシミュレートだけでなく、部分的にインプリメントされたオブジェクトを同時にシミュレートすることが可能であるため、より正確なパラメタが測定できる。一般に、ユーザインタフェースを持つシステムの場合は、インタフェース部分を先にインプリメントし、実際にユーザに使用してもらうことにより、より実データに近いワークロード仕様やリソースの使用法などのデータを使ってパラメタの測定が可能である。これにより、SPE のように開発初期段階から開発ライフサイクルを通して性能の見積りができるだけでなく、より精度の高い性能見積りが可能である。

3.3.1 性能実測方法

Hybrid Simulator では、システム実行モデルのパラメタとして、以下のようなものが測定可能である。

- オブジェクトの使用頻度
各シナリオを実行したとき、各オブジェクトが何回使われているかを測定。Hybrid Prototyping では、オブジェクト指向でシステムを分析/設計するので、リソース=オブジェクトと考える。
- オブジェクト間の通信頻度
各シナリオを実行した時に、各オブジェクト間のメッセージのやりとりが何回あるかを測定。これにより、仕様レベルでの通信が原因のボトルネックを予測するために使うことが可能である。
- インプリメントされたオブジェクトトータルの実時間
実際の処理にかかる時間を測定する。言い替えると、

あるオブジェクトがメッセージを受けとってからの、それに対する応答を返すまでの時間を測定する。これは、キューイングネットワークのサーバのスループットタイムの実データとなる。

- インプリメントされたオブジェクトの CPU 占有時間
一つのメソッドを実行するのにかかる CPU 占有時間を測定する。これは一つのメソッドの実行の効率を測るために測定する。

また、性能を評価するために必要なデータを測定するために、従来の Hybrid Simulator に計測機能を追加する必要がある。そこで、図 1 におけるシミュレーションコントローラとオブジェクトハンドラに以下のような計測機能を追加する。

シミュレーションコントローラ: ここでは、オブジェクトの使用頻度、オブジェクト間の通信頻度を計測する。シミュレーションコントローラは、オブジェクト間のメッセージ通信は必ずここを介して行なわれるためその時にそれぞれのオブジェクトに関してカウントする。

オブジェクトハンドラ: ここでは、インプリメントされたオブジェクトトータルの実時間、インプリメントされたオブジェクトの CPU 占有時間を計測する。オブジェクトハンドラは、シミュレーションコントローラの実オブジェクトインタフェースからオブジェクト名とメッセージ名を受けると、それに対応するメソッドを実行する。そのメソッドが実行が終わると、オブジェクトハンドラはその結果として実オブジェクトインタフェースに対してオブジェクト名とメッセージを送り返す。実時間は、このメソッドを実行してから結果が出るまでを測定する。CPU 占有時間は、あるメソッドを実行し終わった時に CPU 占有時間を計測する。また、メソッドのなかで他のメソッドを呼んでいる場合は、その CPU 占有時間を同様に測定し、元のメソッドの CPU 占有時間から引く。これは、分散型のシステムを開発する時に、プロセスの割当を考慮する時に使うことが可能になる。

4 今後の課題

本稿では、Hybrid Simulator における測定可能なシステム実行モデルのためのパラメタおよび、その測定方法について述べた。今後は、これらを実現し、実際のシステム開発に適用してこの測定方法の評価を行っていく。

また、分散のシステムを開発する時には、オブジェクト間の通信コストを測定する必要があるが、現在の Hybrid Simulator の実現方法では不可能である。そこで、より分散を意識した Hybrid Simulator の実現方法の検討が必要である ([藤崎 93] を参照)。

参考文献

- [荒野 93a] Arano, etc.: A New Simulation Technique in Software Prototyping, Summer Computer Simulation Conference '93.
- [荒野 93b] Arano, etc.: An Object-Oriented Prototyping Approach to System Development, COMPSAC '93, to be published.
- [青野 93] 青野, 荒野: Partial Simulator とその構成法, '93 年電子情報通信学会春季大会.
- [Smith90] C.U. Smith: Performance Engineering of Software Systems, Addison-Wesley, 1990.
- [藤崎 93] 藤崎, 他: Hybrid Simulator に必要な機能, '93 年情報処理学会第 47 回全国大会.