

分散システムの性能予測システム

5 R-2

釜坂 等†, 関口 孝興†, 新堂 隆夫†

三菱電機 (株) †情報電子研究所 †コンピュータ製作所

1 はじめに

最近のアプリケーションシステムは、UNIX 機を中心にオープンシステム化、マルチベンダ化、分散化が進展している。このようなシステムでは、構成要素の多様化・分散化による複雑性、UNIX システム自身のレスポンス時間の非決定性、第 3 者ソフトウェア等のコンポーネントの性能が的確に把握できないことなどのために、システム性能の正しい予測が困難になっている。

そこで、我々はシステム設計の段階で決定できるパラメータに基づいて性能予測を行なうツール PROMIS (Process Modeling and Improvement System) を開発している。

本稿では PROMIS によるシステムの性能予測の特徴と機構について述べる。

2 PROMIS

特徴を以下に列挙する。

● 実機による測定

実システムと同じハードウェア、システムプログラム (OS, データベース等) を使用して測定を行なう。実機による測定の優位な点は、モデル化が複雑な不確定な要素 — 例えば、ページングやスワップ、バッファキャッシュの影響など — を設定あるいは計算しなくても、その影響が実現され測定結果に反映されるところである。

● 測定作業の自動化

性能データの測定は、性能に関わる部分のみを実現する模擬的なアプリケーションプログラムを実機上で実行させることにより行なう。この模擬アプリケーションプログラムの生成と実装、これらのプログラムがアクセスする模擬データベースと模擬ファイルの生成と実装、そして模擬アプリケーションプログラムの実行、性能データの測定および収集を自動的に行なう。

● LAN による分散システム対応

UNIX をオペレーティングシステムとするサーバ機やワークステーションが LAN によって接続された分散システムを対象としている。

● グラフィカルユーザインタフェースの採用

OSF/Motif によるグラフィカルユーザインタフェースを採用し、全ての測定作業の設定はボタン操作で行なえる。また性能データをグラフ表示することによりデータの解析を容易にしている。

2.1 性能予測の手順

PROMIS による性能予測の手順を図 1 に示す。

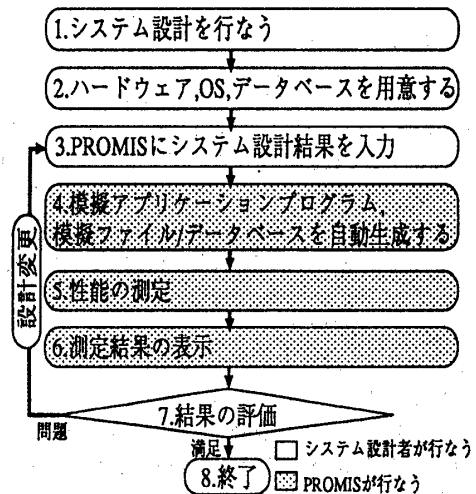


図 1: PROMIS による性能予測の手順

2.2 モデル化

PROMIS はシステム設計の段階で決定できるパラメータに基づいて測定対象となるシステムのモデル化を行なう。必要なパラメータを表 1 に示す。

パラメータはリソースの記述とアプリケーションプロセスの記述とに分けられる。

リソースの記述では、アプリケーションシステムを構成する計算機の台数や機種、使用するデータベース管理システムの種類やデータベースの構成、ファイルのサイズ等を記述する。

アプリケーションプロセスの記述では、プロセスの個数やサイズ、個々のプロセスの振舞い — 起動周期やデータベースへのアクセス回数、ファイルへの書き込みサイズなど — を記述する。また各アプリケーションプロセスが要求される時間内で実行が終了するかどうかを検証するために要求性能も同時に記述する。

表 1: パラメータ

リソースの記述:

CPU	台数, 機種, 主記憶サイズ, バッファサイズ, ディスクの本数とサイズ
データベース	データベース管理システムの種類, データベースの個数, テーブルの数, レコードの数や構成
ファイル	ファイルサイズと個数

アプリケーションプロセスの記述:

起動条件	優先度, 常駐 / 非常駐, 周期, 揺らぎ
処理内容	他のプロセスの起動, ファイルの入出力回数とサイズ, データベーステーブルへのアクセス回数, ネットワーク経由でのデータの送受信回数とサイズ, 画面出力回数, CPU 実行時間
要求性能	起動遅れ時間の閾値, 実行時間の閾値
その他	プログラムサイズ, 実行マシン

2.3 構成

性能測定時のシステム構成例を図 2 に示す。

PROMIS コントローラは、模擬アプリケーションプロセスの生成・実装および、模擬データベースや模擬ファイルの生成・実装を行なう。また、各マシン上の PROMIS モニタを制御する。PROMIS モニタは、各模擬アプリケーションプロセスの起動制御や性能データの測定を行なう。

起動された模擬アプリケーションプロセスは、設定されたパラメータに基づいてデータベースやファイルのアクセスなどを行なう。

PROMIS コントローラは、PROMIS モニタが測定した性能データを収集してデータ表示用ワークステーション上にグラフ表示する。

2.4 性能の予測および結果の評価

PROMIS は、次の性能データを出力する。システム設計者は、これらの性能データを評価し、ボトルネックの検出を行ない最適な設計を行なう。

- プロセスのレスポンス..... 各アプリケーションプロセスの起動遅れ時間(起動要求が生じてから実際に実行開始するまでの時間)や実行時間の平均値と最大値を出力する。また、これらの時間がパラメータで設定した要求性能を満たさなかった場合、その時刻と回数を同時に出力する。
- リソース負荷.... 測定中の各計算機の CPU 利用率、メモリ利用率、ディスク利用率、ネットワーク利用率などを時系列グラフで出力する。

性能データの評価例として、あるプロセスの実行が、要求される時間内で終了しなかった場合の障害を考える。この障害発生時の時刻情報とその時のリソース負荷データから、この時刻の CPU 負荷が非常に高いことが判明

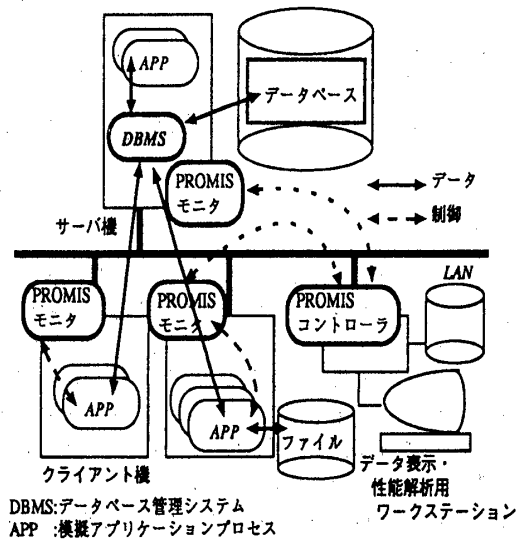


図 2: 性能測定時のシステム構成例

し、これがボトルネックと考えられる。そこで、プロセスを実行させる CPU を変更することにより、CPU の負荷分散を考えたシステムへ設計を変更して再度測定を行なう。

また、その他の設計変更として、プロセスの優先度や常駐 / 非常駐の変更、ファイルやデータベーステーブルのディスク分割の変更などが考えられる。

このようにシステムの設計変更と測定を、満足できる結果が得られるまで繰り返すことによって、最適な設計を行なう。

3 おわりに

今日のアプリケーションシステムは、すべて自社製品のみで構成されることよりも第 3 者ソフトウェアを用いて構築していくことが多い。このようなソフトウェアは詳細性能データや処理方式が不明であることが多く、実システムに近い環境で測定しないと適切な性能の予測が難しい。

このような場合にも、PROMIS により模擬アプリケーションシステムを構築して、動作させることにより信頼度の高い性能データを得ることができる。

また、PROMIS が生成した模擬アプリケーションプログラムを実システムの製作の進展に伴い、順次、実アプリケーションプログラムで置き換えていくことで、模擬システムを実システムに近づけ、性能評価の精度を上げることができる。