

Java を使った分散システムのベンチマークテストの試作

堤 宏智^{*}, 望月雅光^{**}, 山之上卓^{**}, 甲斐郷子^{**}

^{*}九州工業大学大学院工学研究科, ^{**}九州工業大学情報科学センター

あらまし: 学校などで利用されている情報処理システムは, ワークステーション, パーソナルコンピュータを LAN で相互接続した分散システムが一般的になってきている. 機器を多数組み合わせた分散システムを構築する場合, ユーザ側もメーカー側もシステム全体に対する十分な経験や知識を蓄積することは難しく, 分散システムの導入作業や管理運用には多大な労力を伴うことが多い. 我々は, この問題を解決する方法として実運用時間にユーザの挙動に近い負荷をかけることのできる分散システムのためのベンチマークテストを試作している. 本稿では, 基本的なベンチマークテストを開発し, その解析結果を示す.

A Benchmark-Test for a distributed Computing System on Java

Hiroto TSUSUMI^{*}, Masamitsu MOCHIZUKI^{**}, Takashi YAMANOUE^{**}, Kyoko KAI^{**}

^{*} Graduate School of Engineering, Kyushu Institute of Technology,

^{**} Kyushu Information Science Center, Kyushu Institute of Technology

Abstract: A distributed computing system, which is a connected workstations and personal computers by a LAN, is becoming popular among universities and others as an information facility recently. When we build this type of system, It is difficult for users' and makers' side to accumulate enough experiences and informations all about that system. So it is common that it will require a great deal of burden to build and administer the system. We are designing a benchmark-test for a distributed computing system, which is based on real users' behavior as a method to lighten this burden. We have used the basic benchmark-test for some distributed computing systems and analyzed the result.

1.はじめに

企業や学校で利用されている情報処理システムは, NFS などの分散ファイルシステムによって管理している. このような施設における LAN では, ワークステーション, パーソナルコンピュータを相互接続した分散コンピュータシステムが一般的になってきている. このようなシステムは, ここ数年のわず

かな間で急激に普及しており, 構成する機器も技術的進歩により目まぐるしく変化している. 機器を多数組み合わせた分散システムを構築する場合, ユーザー側もメーカー側もシステム全体に対する十分な経験や知識を蓄積することは難しく, 分散コンピュータシステムの導入作業や管理運用作業には多大な労力を伴うことが多い. また, システムの実運用時間に原因不明のエラーが起こった時どの場所でエラ

一が起こったのかを究明するのは、容易ではない。この問題を解決する方法として、筆者らは分散システムのためのベンチマークテストを試作している。このベンチマークテストは、実運用時間のユーザに近い負荷をかけることによってシステムの性能を測定できるものを目指している。以下 2 節で、ベンチマークテストシステムの概要と負荷モデルについて述べ、3 節では基本的なベンチマークテストを動作させ、その解析結果を示す。

2. ベンチマークテストの概要

2.1 開発の目的

単体のコンピュータや、ネットワーク機器に対する性能評価についてはすでに様々な研究が行われ、それにもなうベンチマークテストが数多く存在している[1][2]。しかしながら、複数のコンピュータとネットワーク機器を組み合わせた性能評価方法についてはまだ不十分であり、特に多くのユーザが実際に利用する場合を想定した一般の評価基準はあまりみられない[3]。分散システム全体を客観的に評価できれば、分散システムの構築、導入、管理、運用時の負担を軽減することが期待できる。また、負荷によるエラーを計測することにより MTBF、MTTF を求めシステムの信頼性、安定性を示す事ができる。ベンチマークテストから得られた評価基準の存在により、分散システムの性能向上を期待できる。

2.2 システムの構成

本システムは、図 1 に構造を示すように JAVA アプリケーションが、TCP/IP によって相互に通信を行う構成になっている。マスターはベンチマーク使用者が実際に操作を行うアプリケーションであり、エージェントはベンチマークを実際に行う LAN 上のコンピュータ上で動作するアプリケーションである。マスターはシステムの操作を行う GUI 部、GUI の操作を処理する測定部、他のエージェント群

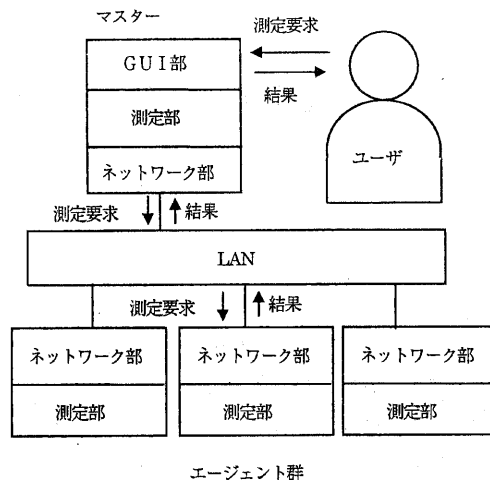


図 1：システムの構成

との通信を処理するネットワーク部で構成される。またエージェントは、測定部とネットワーク部で構成される。

2.3 設計方針

ベンチマークシステムの機能を考える際に、次の点を設計方針とした。

- (1) どのような分散システム上でも容易にベンチマークシステムを構築できる。
- (2) 実物の分散システム上で実運用時間中に分散システムに加わる負荷を擬似的に再現できる。

(1) の問題を解決するため、我々は最もプラットフォームの依存性の少ない言語である JAVA を利用することにした。現時点で、できるだけ多くのシステムで動作することを考え、今回は JDK1.02 を選択した。(2) については 2.4 説で述べる。

2.4 システムの動作

測定者はサーバの GUI を通して、分散コンピュ

ター環境に負荷を加えるようにエージェントに指示することができる。また、長時間にわたって、自動的にエージェントに連続的な負荷を発生させることができる。エージェントは測定部で要求に応じた負荷を発生させる。2.3 節の(2)については、システムのさまざまなログから適切な情報を抽出し、実存する分散コンピュータシステムでユーザが実際にアプリケーションを使用している際に生じる負荷を擬似的に発生させることを考えている。例えば、コマンドログからは、コマンド実行時間、使用されたりソースなどの情報が得られる。この情報に基づいた負荷を発生させる。ログを基本にして負荷を発生させることによってその分散システム専用のベンチマークとなる。分散システムコンピュータで負荷のタイプは現時点で以下の4種類を想定している。

1) 外部記憶装置 (ハードディスク等)

指定のファイルに複数のエージェントがアクセスすることにより、共有ハードディスク等に負荷を発生させる。

2) 内部記憶装置 (メモリ等)

スレッドの生成、変数領域の確保を行うことによりメモリに負荷を発生させる。

3) ネットワーク

TCP,UDP パケットの送出をエージェント間で行い、ネットワークに負荷を発生させる[4]。

4) CPU

各エージェントで演算処理させることによって負荷を発生させる。

この4つのタイプを使い分けることによって、ベンチマークシステムは分散コンピュータシステムでユーザが実際にアプリケーションを使用している際に生じる負荷を擬似的に発生させることができる。単一項目に対しテストを実行するだけでなく、連続的な実際の運用状態の再現が可能となる。分散コンピュータシステムの評価は処理速度の変動、誤動作発生回数、最長連続運用時間、MTTF、MTBF等のデータをもとに行なう。

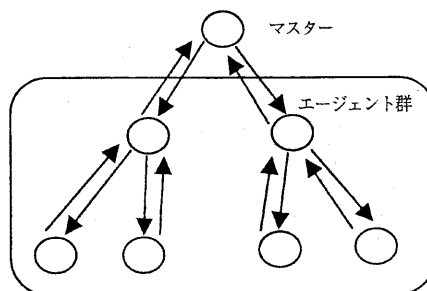


図2：メッセージの伝播

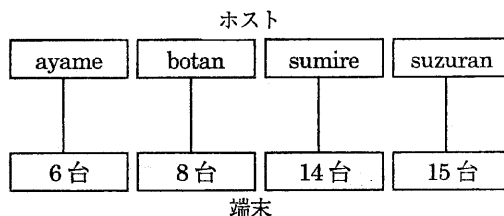


図3：実験システム構成

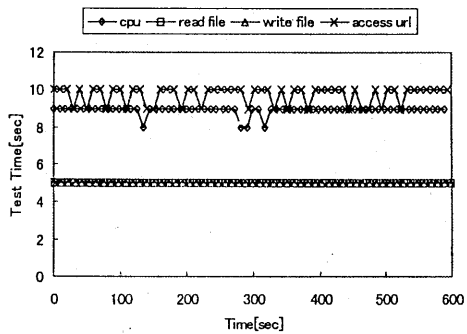
2.5 測定要求

エージェントへの負荷発生及び測定要求はメッセージパッシングにより行われる。メッセージはサーバからエージェント、エージェントからエージェントへ、ツリー状に伝播していく。図2はそのイメージである。通信経路をツリー状にすることでブロードキャストの高速化を図っている[5][6]。測定の結果は、このツリーを逆方向に伝播していく。

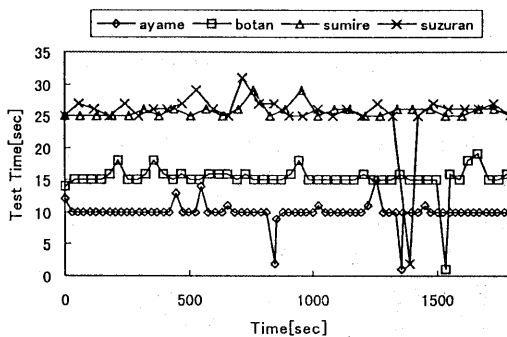
3.基本的なベンチマークテストの結果と解析

3.1 基本的なベンチマークテスト

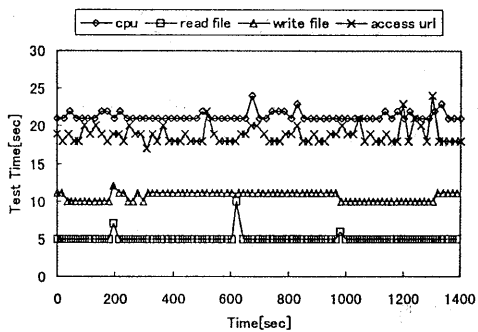
図3のようなシステムに対して、基本的な負荷を連続的にエージェントで発生させた場合どのような応答が結果として得られるか実験、解析する。また、LANで接続されたパーソナルコンピュータ1台とUNIX端末1台に対して実験を行い、図3のシステムの結果と比較、検討する。



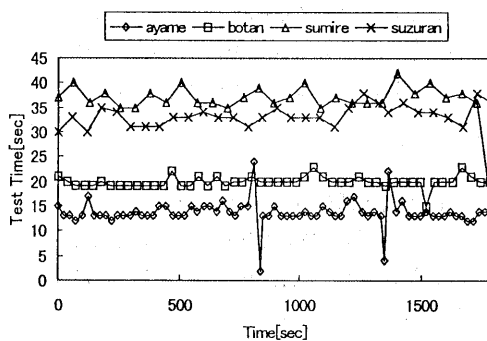
(a) 一台の win95 の対してテスト



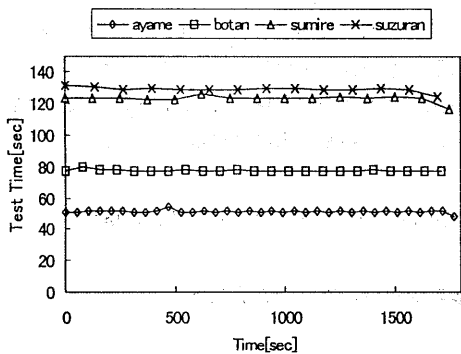
(d) ファイル読み込み負荷



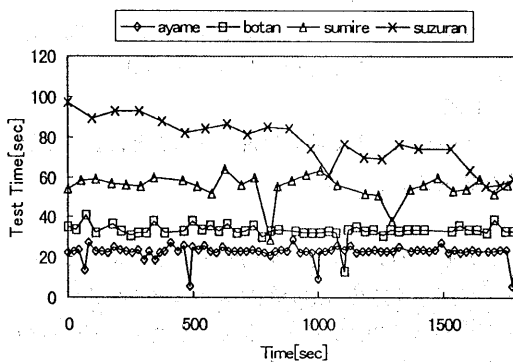
(b) 1 台の Unix 端末に対してテスト



(e) ファイル書き込み負荷



(c) CPU に対する負荷



(f) URL アクセスによる負荷

図4：実験結果

3.2 結果と解析

図4に、実験の結果を示す。(a)と(b)は、1台に対してのベンチマークテスト結果、(c)以降は図3に示す複数台に対してベンチマークテストを動作させ、各々のホストにつながっている端末1台についての結果である。

(a)と(b)の処理時間を比較すると、Win95の動作しているパーソナルコンピュータの性能の良さが分かる。また、この1台への計測結果と分散システムに対する負荷の計測結果を比較すると処理速度に大きな差が生じている。これは、複数の端末から同時に連続した負荷をかけているからである。この分散システムの大きな挙動を読み取れば、どの部分に負荷が集中しているか示すことができる。(d)と(e)の結果の中では、ファイルの読み取りに失敗している箇所がある。このような誤動作を計測すれば、システムの安定性を示せる。

今回の実験で、ベンチマークを動作させることでシステムの全体像を把握するのに必要な情報をいくつか得られたといえる。

4.まとめ

本システムは次にあげる3つの点で有効である。

- 1) 分散システムを定量的に比較することができる。また、MTBF、MTTF、誤動作回数等よりシステムの信頼性と安定性を評価できる。
- 2) 新たに分散システムを導入する際、運用開始前にベンチマークテストを動作させることによって、

運用開始後のトラブル発生を抑えることができる。

3) このシステムはJAVAを用いているため、現存する数多くの分散システム上で容易に実行が可能である。

今後の課題として、このベンチマークを利用するうえで分散システム運用時のログの分析と、それに基づくユーザ利用形態のモデル化が必要不可欠である。特にログをそのまま再現するのではなく、ログから如何にして評価に有効なデータを抽出し、活用するかを考える必要がある。また、ログからスクリプトを作成し、そのスクリプトからさまざまなユーザの挙動を再現出来るような可搬性のあるベンチマークを開発するのが、今後の課題である。

参考文献

- [1] Peter C. and David P., "Storage Performance Metric and Benchmarks, Proc.IEEE, pp1151-1165, 1993.
- [2] J. Bunch, J. Dongarra, C. Moler, and G.W. Stewart. LINPACK User's Guide. SIAM, Philadelphia, PA, 1979.
- [3] 石原 進, 小島 英樹 他, "ユーザ挙動モデルに基づく教育用システムに適したネットワークシステム構成の検討", 分散システム運用技術 11-2, 1998.
- [4] Kury J. Maly, Ajay K. Gupta, Satish Mynam, "BTU:A Host Communication Benchmark", Computer, pp66-74, 1993.
- [5] 山之上 卓, 山根 真人 他, "Javaとコンピュータネットワークを利用した教育支援システム", 平成8年度情報処理教育研究集会講演論文集, pp.117-120, 1996.
- [6] A. Farley, "Minimum Broadcast Network", Network vol.9, pp.133-332, 1979.