

分散システムの信頼性評価のためのシステム特性要因 の分析と測定

井上哲哉 勅使河原可海

創価大学大学院工学研究科

e-mail: tinoue@edu.t.soka.ac.jp

概要

通信ネットワークを利用するソフトウェアの品質がよいかどうかを判断するためには、その「品質のよさ」を性能評価指標として示す必要がある。そのために、分散システムの構成要素を計算機システム、ネットワーク関連、ソフトウェア、利用者として捉え、それぞれの特性要因に着目した。この性能評価モデルを検討するにあたり、具体的なシステムにおいて、TCP 接続された2つのプロセス間でデータ転送を行い、システムの利用者の視点から、クライアントの応答時間を測定し、サーバ側の処理能力の視点からサーバプロセスのCPU実行時間、およびそのメモリ消費量を測定した。本論文では、これらの実験の測定結果に起因する要素を明らかにし、分散システムの信頼性確保のためのサービス品質の効率性に影響を与える要因について議論する。

Analysis and Measurement of System Characteristic Factors of Reliability Evaluation in Distributed Systems

Tetsuya Inoue Yoshimi Teshigawara

Graduate School of Engineering, Soka University

Abstract

To judge whether quality of software using communication networks is high or not, it needs to indicate the quality level by performance evaluation indices. This paper defines that distributed system components are composed of computer system, networks, software and users and it focuses on their characteristic factors. In order to build a performance evaluation model for distributed systems, response time on a client site are measured in a user's viewpoint by transferring data between two processes connected by TCP/IP in an existing system. In addition, CPU time and memory consumption in server process are measured in a viewpoint of processing ability by the server.

Then, this paper clarifies specific factors derived from their results of measurement and discusses about factors affecting efficiency of quality of service to maintain distributed systems reliable.

1. はじめに

FTP を使ってデータ転送を行ったとき、その利用者が転送時間に対して「遅さ」を感じることもある。この「遅さ」の原因として考えられることは、データ転

送中にパケット損失が発生し、TCP の再送機能により転送時間に時間がかかっているかもしれない。もしくは、パケット損失はないが、サーバ側のプロセスに CPU 負荷がかかり、データ転送の処理に時間がかか

っているかもしれない。また、FTPに限らず、アプリケーションソフトウェアがネットワークの特性を十分に考慮して設計されていないために、利用者の感じる「遅さ」がソフトウェアの仕様の問題やプログラムの記述的な問題に起因することも考えられる。このように、システムのある事象をもたらす原因の特定は容易でなく、その事象がネットワーク性能に依存するのか、またはアプリケーション側に依存するのか、それともハードウェア装置そのものに問題があるのか、その切り分けは難しい。さらに、問題視される事象が利用者側の入力データによって再現しないことさえある。

本研究では、このような事象の原因を究明するために、システムの利用者の視点から対象システムのサービス提供によって生じたシステムの事象を状態としてとらえ、この状態をサービス品質特性とする。そして、この品質特性の「良さ」を判断するために、性能評価によって定量化し、サービス品質特性とそれに関係するさまざまな要因について考察する。さらに、抽出された要因に基づいて、分散システムの信頼性評価モデルを構築することを目的とする。

本論文では、分散システムを扱い、通信ネットワークを介したプロセス間通信において、データ転送時のサービス品質特性に影響を与える要因を考察し、エンドツーエンドの品質計測からIP層およびTCP層、アプリケーション層の各階層の性能評価を行い、サービス品質特性とその要因について議論する。さらに、システムの信頼性の観点から各特性要因が分散システムの信頼性にどのような影響を与えるのか議論する。

2. サービス品質特性とその要因の明確化

本格的なマルチメディアサービスに向けて、利用者にとって使いやすく、信頼性の高いサービスを提供するために、サービス品質の特性とその要因について明らかにする必要がある。しかしながら、エンドツーエンド通信のデータの送受信に関するサービス品質をデータ系列の相関として捉えることがあっても、その品質特性に対する要因から対象システムのサービス品質を評価することはあまり議論されていない。そこで、システムの利用者の視点から、アプリケーション品質を含むエンドツーエンド通信のサービス品質特性とその要因について考える。システムが扱うサービスは、TCP 接続された2台の計算機システムにおけるデー

表1 システムの構成要素と主な要因

Components	Main characteristic factors
Computer system	Buffer of TCP socket, Buffer of application, MSS, MTU, Memory consumption, CPU time
Network	Packet loss, Packet delay, Unit failure
Software	Software design and specification, Software failure, Software fault
Users	Requested file size, Requested service

タ転送サービスとする。このサービスによって生じたシステムの状態（結果）をサービス品質特性としてとらえる。サービス品質特性を接続性、効率性、信頼性と定義する。接続性は、片端から他端にネットワーク接続できる能力とし、通信を開始するときの両者間のネゴシエーションに関する性質である。効率性は、ソフトウェアの実行時間やシステム資源の使用率など効率に関する性質である。信頼性は、意図する期間中に、対象システムが仕様を満足するサービスを提供し続ける能力とし、効率性がボトルネックとなるシステム障害が原因で利用者がその仕様を満足するサービスを受けられなくなったとき、信頼性は失われるものとする。

次に、これらのサービス品質特性に対する要因系要素として、分散システムを構成している要素を次の4つに大きく分けることにする。表1は、本論文で扱うシステム構成要素とその主な要因である。

- ・計算機システム
CPU やメモリ、外付けディスク装置など周辺機器を含んだ計算機システムとしての要素
- ・ネットワーク関連
2台以上の計算機システムを接続する通信ケーブルや通信制御装置など主にデータ伝送に使用される通信ネットワークに関連する要素
- ・ソフトウェア
計算機システムのOS上で実行されるアプリケーションプログラムなどのソフトウェアとしての要素
- ・利用者
対象システムのサービスを利用する利用者としての要素

さらに、分散システムの信頼性評価を行うために、システム障害をもたらす主な原因として、サービス品質特性の効率性に着目する。そして、この効率性に対する要因について考察するために、計算機システムと

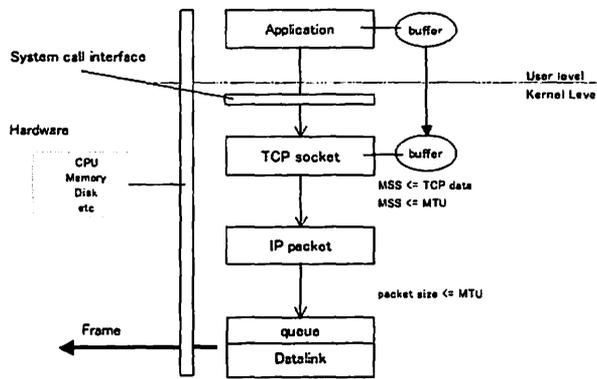


図1 TCPソケットの動作とバッファの関係

ネットワーク関連の要因系要素を系統的側面とし、ソフトウェアと利用者の要因系要素をソフトウェア的側面として、それぞれの側面から議論する。

2.1 システム的側面

UNIX等のOS上でアプリケーション(ユーザプログラム)を実行するとき、プロセスが生成される。このとき、プロセスはCPUによって実行され、メモリを使用し、メモリはプログラムの記述によって書き換わっていく。アプリケーションが通信ネットワークのサービスを要求するとき、そのアプリケーションはネットワークサービスを提供するシステムコールを実行する[1]。例えば、C言語で記述されたプログラムでは通常、ネットワーク入出力を行うためのシステムコールとしてソケットを使用する。図1に示すように、TCPソケットは送信バッファを持っており、アプリケーションがwrite関数を呼び出すとき、カーネルはアプリケーションがもつバッファからTCPソケットの送信バッファにデータをコピーする。このとき、プロセスはカーネルによってユーザレベルからカーネルレベルへ制御が移される。カーネルレベルでのプロセスのTCP送信処理は、TCPソケットの送信バッファのデータをTCPの複雑な規則に基づきパケットとしてIPに渡す。このとき、IPパケットは、MSS(Maximum Segment Size)か、それ以下のサイズとなる(セグメント化)。パケットを受け取ったIPは経路テーブルから終点アドレスを検索し、そのパケットを適切なデータリンクの出力キューに渡す。そして、出力キューに渡されたパケットは、フレームとしてネットワークが接続されている相手側のプロセスに送信される[2]。

このようなシステム特性から、図2に示すシステムの側面から、2つの要因系要素が考えられる。

まず、図2のネットワーク関連の要因系要素では、ネットワークインターフェースカード、通信ケーブル、HUB、そしてSwitchなどの通信制御装置を含み、この要素内におけるネットワークの効率性に対する要因として、通信ケーブル不良やパケット損失、パケット遅延などが考えられる。次に、計算機システムの要因系要素は、ネットワーク関連に対する要因系要素として位置付けられる。ネットワークサービスを提供する実行プロセスは、他のプロセスとシステム資源を共有しながらデータ転送を行う。システム資源は、システム性能を左右する要因であり、その主なものにメモリ性能があげられる。メモリに関するバッファサイズの大きさを考慮すると、実行プロセスの効率性の要因として、アプリケーションがもつバッファサイズ、TCPソケットのバッファサイズ、MSS、MTUなどがあげられる。さらに、システム資源の効率性の要因として、実行プロセスのCPU実行時間やメモリ消費量、そして、ディスク入出力などが考えられる。

2.2 ソフトウェア的側面

サービス品質特性の効率性の問題は、計算機システムやネットワークに関連する要素だけでなく、プログラムの仕様やソフトウェアの設計に依存することは十分に考えられる。図2に示すソフトウェア的側面から、サービス品質特性の効率性に対するソフトウェアの要因系要素では、ソフトウェア故障(failure)やソフトウェア障害(fault)、ソフトウェアの設計・仕様などがあげられる。ソフトウェア故障は、ソフトウェアが利用者の期待通りに動作しないことである。ソフトウェア障害は、ソフトウェア故障を引き起こしたプログラム内の欠陥や誤りである。これらは、単なるプログラムの記述ミスだけではなく、ユーザプログラム内で呼ばれるシステムコールなどのライブラリ関数の不具合も含む。ソフトウェアの設計・仕様は、ソフトウェア障害ではないが、つまりプログラムのバグではないが、ソフトウェアの設計やプログラムの仕様を変更することで、サービス品質を改善されることを意味する。例えば、実行プロセスに対するメモリ増大の問題があげられる。この問題は、メモリリークとして知られ、システム資源が不足し、計算機システム全体が遅くな

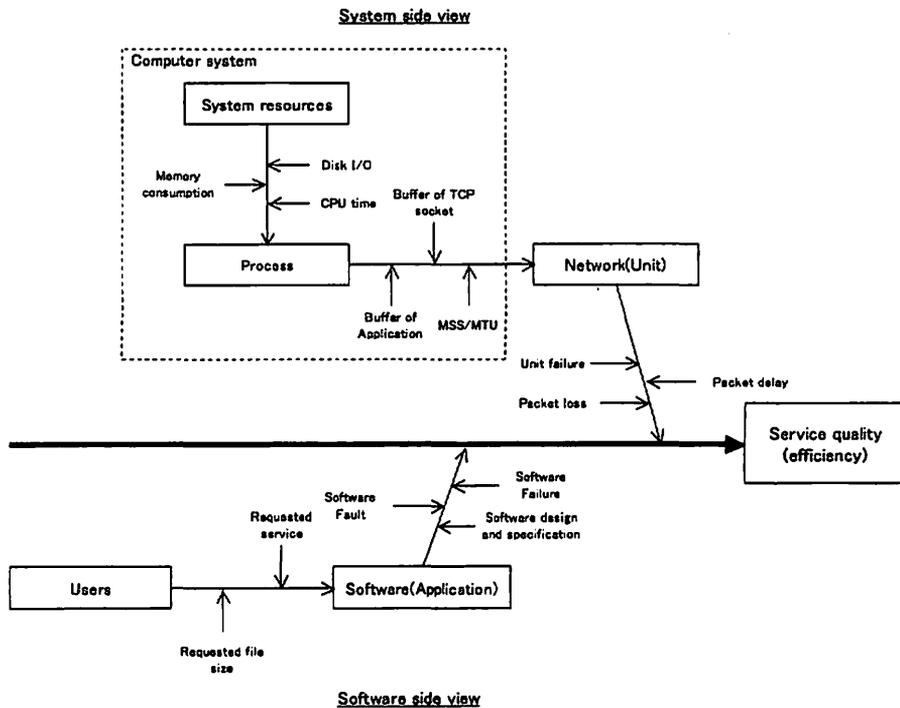


図2 サービス品質特性の効率性に対するさまざまな要因

ることがある。メモリーク問題のほとんどは、ユーザプログラムの仕様や記述を見直すことで改善することができ、サービス品質の効率性を上げることができる。次に、利用者の要因系要素は、ソフトウェアに対する要因系要素として位置付けられる。利用者（クライアント）が相手側（サーバ）に要求したファイルサイズやサービス内容などは、サービス品質特性の効率性に影響を与えることが考えられる。

3. 性能評価の測定

本章では、システムの利用者の視点から、アプリケーション品質を含むエンドツーエンド通信のサービス品質特性の効率性を性能評価によって定量化する。

3.1 性能評価指標

性能評価の目標は、システムの利用者が要求したファイルを取得した際、データ転送サービスの要求を開始してから完了するまでの間でサービス品質特性の効率性に対してどの要因がどれだけ影響を与えるのかを明らかにすることである。そして、対象システムの性能評価指標を決定するために、ネットワークに依存する部分とアプリケーションに依存する部分に分けることにする。

ネットワークの性能評価指標として、パケット損失、パケット遅延、スループットを扱う[3]。パケット損失は、片端から送信されたパケットが他端まで届かない事象である。パケット遅延は、片端から送信したパケットが他端に届くまでの経過時間である。スループットは、単位時間あたりの転送データ量である。

アプリケーションの性能評価指標は、利用者側の視点からのものと管理者側からの視点のものと大きく2つに分けることができる[4]。利用者側の視点から応答時間、管理者側の視点として、サービス時間、実行プロセスのCPU実行時間とメモリ消費量を扱うことにする。応答時間は、クライアント側がサービス要求を開始から終了するまでの経過時間である。サービス時間は、サーバ側が利用者に対するサービス提供に費やした時間である。また、プロセスが生成されるときの仮想アドレス空間は、カーネルによって割り当てられるが、通常、この空間はユーザアドレス空間とカーネルアドレス空間にわけられることから、CPUの実行時間は、これらのアドレス空間の生成から消滅までの時間とし、メモリ消費量は、ユーザアドレス空間の消費量とする。

なお、パケット損失とパケット遅延、CPU実行時間とメモリ消費量は、サービス品質特性の効率性に対

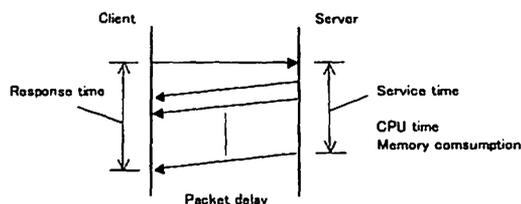


図3 測定する性能評価指標

する要因としてあげているが、ここでは、性能評価指標として用いることにする。

3.2 実験システムの構成

今回の実験におけるハードウェア構成は HP 9000/800 サーバ (CPU は PA8200 240MHz×4, メモリは 1024MB) を 2 台使用し, 両ホスト間は HUB を介して Ethernet 10Mbps の LAN で接続している。OS は 2 台ともに HP-UX 11.0 である。

3.3 測定ツール

IP 層および TCP 層, アプリケーション層の各階層の性能評価を行う。

アプリケーション層レベルでの測定ツールとして, 実際にアプリケーションが実行されたときのプロセスの振る舞いを測定するために, クライアント用測定プログラムとサーバ用測定プログラムを作成した。サーバプログラムは, Web システムの処理を想定して, 複数の子プロセスの生成が可能である。これらのプログラムは, プロセス間通信を行い, データ転送を行うと同時に, クライアント側の応答時間, サーバ側のサービス時間, CPU 実行時間およびメモリ消費量を測定する。

IP 層および TCP 層レベルでの測定ツールとして, 2 つのプロセスがデータ転送を行うときに流れるパケットを測定するために, パケット監視プログラムを作成した。このプログラムで扱われるパケットデータは, クライアントとサーバがプロセス間通信で使用する際のポート番号を指定可能であり, 他のプロセス間通信で使用されるパケットは取得されない。

4. 効率性に対する要因の影響

今回の実験の主な目的は, 性能評価指標からサービス品質特性の効率性に対する各要因の影響度を測るた

め, 1 クライアント 1 サーバプロセスとし, ファイルサイズを 434 KB, 1095 KB, 1696 KB の 3 種類の PDF ファイルを使用してデータ転送を行った。試験パターンは, クライアントとサーバのアプリケーションがもつバッファサイズに 256 バイトから 2048 バイトまでの値を設定し, クライアント側の応答時間 (効率性) に対するアプリケーションのバッファサイズ (要因) から, サーバ側の CPU 時間, メモリ消費量, パケット遅延を測定した (図3参照)。そして, この測定結果からシステムの効率性に対してどの要因がどれだけ影響を与えるのかを考察する。

まず, 図4に示すように, それぞれのファイルサイズに対するアプリケーションのバッファサイズは, クライアント側の応答時間にあまり効果をもたらすものではないと思われる。しかし, 図5, 図6に示すように, アプリケーションのバッファサイズがサーバ側の CPU 時間およびメモリ消費量に影響を与えていることが確認された。特に, アプリケーションのバッファサイズを 256 バイトから 512 バイトに増やしたとき, CPU 時間, メモリ消費量ともに, 256 バイトの約半分以下の数値を示している。この結果から, 今回のようなパターンでは, バッファサイズを 256 バイトからその 2 倍の 512 バイト以上にすることで, CPU 時間, メモリ消費量を約半分にできるが, 応答時間には影響がないと言える。また, このことは, アプリケーションのつくりが対象システムの効率性を左右しているとも言えることができる。

4. 分散システムの信頼性の考察

通信ネットワークを利用するソフトウェアを設計する上で, 重点を置かなければならないことは, 正常処理よりも例外処理であると言える。例外処理とは, システム障害を避けるために, システムコールやプログラム特有の関数の戻り値から仕様に基づいた適切な処理を行うことであり, システムの信頼性を確保するための処理と言える。例えば, サービス品質特性の効率性から言えば, 実行プロセスのメモリの増大はシステムがもつ特性であり, プログラムの記述を見直すことによってある程度まではメモリの増大を抑えることができる。しかし, プログラムを見直したからと言って, 実行プロセスのメモリの増大はなくなるわけではない。この場合, メモリを割り当てる関数の戻り値がこれ以

上割り当てることができないというエラーを返したならば、例外処理として、そのプロセスは終了すべきであり、実行プロセスのメモリの増大値がある一定の値を超えたなら、システムの仕様に基づいた適切な処理を行うべきである。このような処理を行うことは、分散システムの信頼性を確保するために最低限必要なことであると言える。

5. 今後の課題と検討

サービス品質特性の信頼性の定義から、効率性がボトルネックとなってシステム障害を引き起こし、システムのサービスを停止（システム故障）させないことが重要である。今後の課題として、サービス品質特性の効率性の評価により信頼性を予測することは本研究の課題である。また、実際に測定された値から各特性要因の関連性を見つけ、対象システムの信頼性評価のための確率・統計的なモデルを検討し、さまざまな試験パターンからそのモデルを検証していきたい。

参考文献

- [1] Curt Schimmel: UNIX Systems for Modern Architecture, Addison-Wesley(1994). (監訳: 前川守, 訳: 岩川信一, “UNIX カーネル内部解析”, ソフトバンク(1996)).
- [2] W.Richard Stevens: UNIX Network Programming Volume 1, Second Edition Networking APIs: Sockets and XTI, Prentice Hall(1998). (訳: 篠田陽一, “UNIX ネットワークプログラミング 第2版 Vol. 1 ネットワーク API : ソケットと XTI”, ピアソン・エデュケーション(1999)).
- [3] 鶴正人, 尾家祐二: インターネットの特性計測技術とその研究開発動向, 情報処理学会誌 Vol.42 No.2(2001).
- [4] 亀田壽夫, 紀一誠, 李頡: 性能評価の基礎と応用, 共立出版(1998).

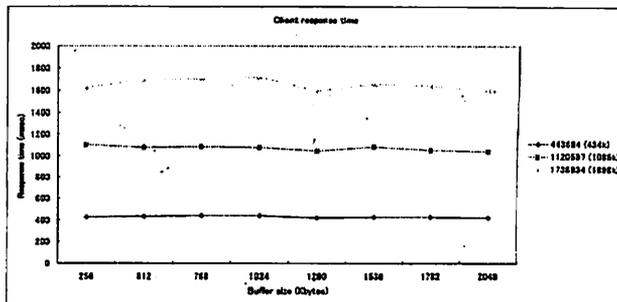


図4 クライアントの応答時間

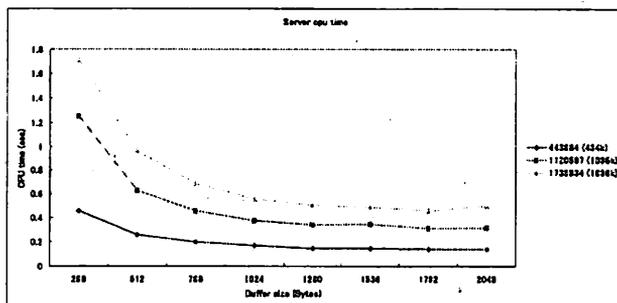


図5 サーバプロセスのCPU実行時間

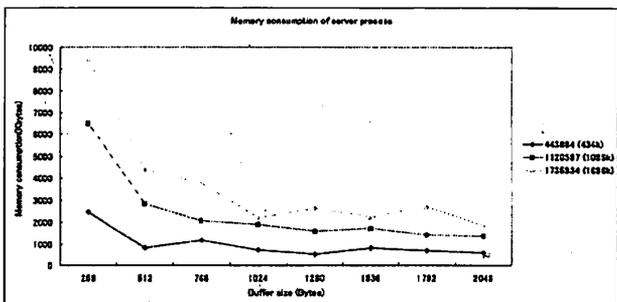


図6 サーバプロセスのメモリ消費量