

EDI システムにおけるエンドユーザレスポンスを 予測するための性能モデル

佐藤 雅之^{*†}、桑野 克彦[‡]

EDI (Electronic Data Interchange)システムにおける処理時間(エンドユーザレスポンス)を予測するパラメータとして、従来、EDI の受信データ量が用いられていた。しかし、受信データを宛先により仕分することにより、エンドユーザレスポンスの高速化を図った EDI システムにおいては、EDI の受信データ量のみでは、エンドユーザレスポンスの予測は不正確になる。このような EDI システムを対象に、新たに、EDI データ中の宛先数も用いることで、より正確にエンドユーザレスポンスを予測する事例を報告する。

A performance model to predict an end user response in an EDI system

MASAYUKI SATO[†], KATSUHIKO HATANO[‡]

As a parameter to predict the processing time (the end user response) in an EDI(Electronic Data Interchange) system, quantity of reception data of EDI was used conventionally. However, in the EDI system which speeded up more by dividing reception data by an address, a prediction of the end user response is incorrect only with quantity of reception data of EDI. For such an EDI system, I report an example to predict the end user response precisely in an EDI system by using the address number of EDI data.

1. はじめに

システムの運用においては、システムの性能を把握することが重要である。会社間、VAN(Value-Added Network)間のデータ交換を

行う EDI システムにおいては、サービスの安定した提供のため、短期的、中期的なエンドユーザレスポンスの予測を行い、対策を計画する。中期的なエンドユーザレスポンスの劣化が予測される場合、システムのアップグレードなど

[†]三菱電機株式会社 情報技術総合研究所
Information Technology R&D Center, Mitsubishi Electric Corporation.

[‡]三菱電機情報ネットワーク株式会社
Mitsubishi Electric Information Network Corporation.

必要な対策を検討する。このため、システムのエンドユーザレスポンスの正確な予測を行う必要がある。

従来使用していた EDI システムにおいては、次のようにデータが処理されていた。

- ① データの着信 (=集信)
- ② データのメッセージグループ毎の分割 (=仕分)
- ③ 宛先の同じデータ毎の結合 (=統合)
- ④ 宛先への送信 (=配信)

EDI システムでは、仕分後のデータをディスクに書き出す。従来使用していた EDI システムでは、この際のディスク I/O がデータ量に依存した回数が発生し、時間がかかっていた。このため、着信したデータ量に基づいて、エンドユーザレスポンスを予測する手法がとられ、正しく予測を行うことができていた。

しかし、現在用いている、エンドユーザレスポンスのより高速化を行った EDI システムでは、仕分の段階で、データを単位毎に分割せず、メッセージグループ中の宛先の等しいもので

分割する方式を採り、ディスク I/O を抑える手法を採る。

このような、より高速化した EDI システムでは、従来のようなデータ量に基づいた方式では、発生するディスク I/O がデータ量へは依存せず、正しくエンドユーザレスポンスの予測を行うことができない。

本報告では、このように高速化を目指した EDI システムにおいて、より正確にエンドユーザレスポンスを予測するための性能モデルを示す。さらに、本性能モデルについて、性能予測を行い、実際に取得したデータを用いて検証を行った結果、本性能モデルが有効に機能することを示すことができた。

2. EDI システム

EDI とは、企業や行政機関などが、コンピュータをネットワークで繋ぎ、伝票や文書を電子データで、自動的に交換することである。従来使用していた EDI システムにおけるデータの流れは、図 1 のようになる。

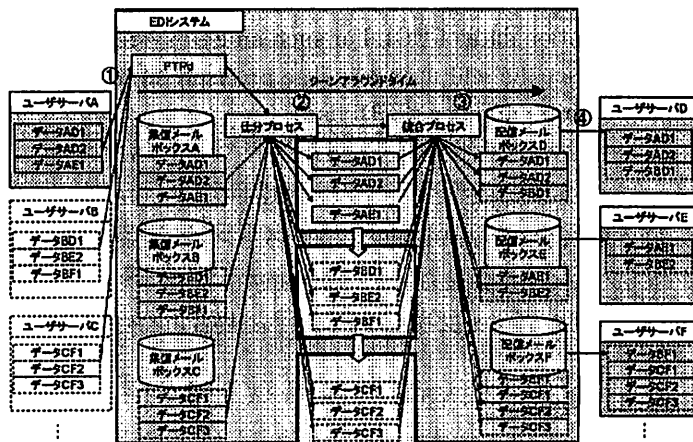


図 1 EDI システム構成図

以下に、処理の流れを示す。

- ① ユーザーバ A から EDI システム上の FTPd に対しデータ A が送られ、集信メールボックスにデータ A として保管される。データ B、データ C についても同様。
- ② 仕分プロセスは、集信メールボックスへの着信を契機に FTPd より起動される。データ A を宛先毎に、データ AD1～データ AE1 に分割する。データ B、データ C についても同様。仕分けプロセスは、分割されたデータをディスクへ書き込んだ後、直ちに、統合プロセスを起動する。
- ③ 統合プロセスは、ディスク上のデータ AD1～データ CF3 を宛先毎に統合し、配信メールボックス D～配信メールボックス F に配信する。
- ④ ユーザーバ D は、配信メールボックス D からデータを引き取る。

なお、ここに示す EDI システムでは、複数あるいは単数の集信データから複数あるいは単数の配信データを作成する。

3. EDI システムにおけるエンドユーザレスポンス

VAN 会社における EDI サービスにおいては、銀行、各社の経理、他社 VAN とのデータ通信が多くあり、24 時間運用に近い状態でシステムを稼働させている。また、顧客サービスの観点から、顧客が障害を認識する以前にシステムの障害を検出し、対応を行うことが目標とされる。

このため、EDI システム運用上のエンドユ

ーザレスポンス管理の目的は、次の 2 点である。

- ① 短期的なエンドユーザレスポンスの低下の検出による、EDI システム障害の検出
- ② 中期的なエンドユーザレスポンスの低下の検出による、EDI システムのアップグレードの検討

従来の EDI システムでは、中期的なエンドユーザレスポンスの低下の検出において、ユーザデータ量に基づき、性能予測を行う。実際に用いている、性能予測モデルは以下である。

$$t = ax + b, \quad (1)$$

t エンドユーザレスポンス

x データ量

a, b 定数

すなわち、パラメータ“データ量”に対して、エンドユーザレスポンスが比例の関係にあるものとして、予測する。

4. より高速化した EDI システム

従来使用していた EDI システムにおいては、1 件のデータ中に多数のメッセージグループが含まれる場合、多量のディスク IO が発生する。一方、より高速化した EDI システムでは、次のようにデータの仕分方式を変更したデータの処理手順を採用することにより、ディスク IO を抑える。

- ① データの着信 (=集信)
- ② データの宛先の等しいメッセージグループ毎の分割 (=仕分)
- ③ 宛先の同じデータ毎の結合 (=統合)
- ④ 宛先への送信 (=配信)

なお、ここで用いる宛先は、「連続するメッセージグループ中で、宛先の等しいメッセージグループの数」として定義した。データ中の宛先の概念を図2に示す。

このような処理手順を採用した高速なEDIシステムにおいては、実際にディスクIO以外のCPUがボトルネックとなる。

より高速化されたEDIシステムの構成より、ボトルネックとなりうるのは次の3種である。

- ① CPU
- ② メモリ
- ③ ディスク IO

これらのリソースについて、最大処理時間が発生しているときのデータを使用し、以下のようにボトルネックを求める。

- ① 一日分の実データから、最大処理時間が発生しているピーク時と、平常時と比較し、リソースの使用状況を求める。
- ② 負荷の高いリソースを求め、ボトルネックを求める。

具体的には、次のデータを使用する。

- ① CPUの使用率
- ② CPUのIO待ち率
- ③ メモリの使用率

それぞれの取得方法は、以下のとおり。

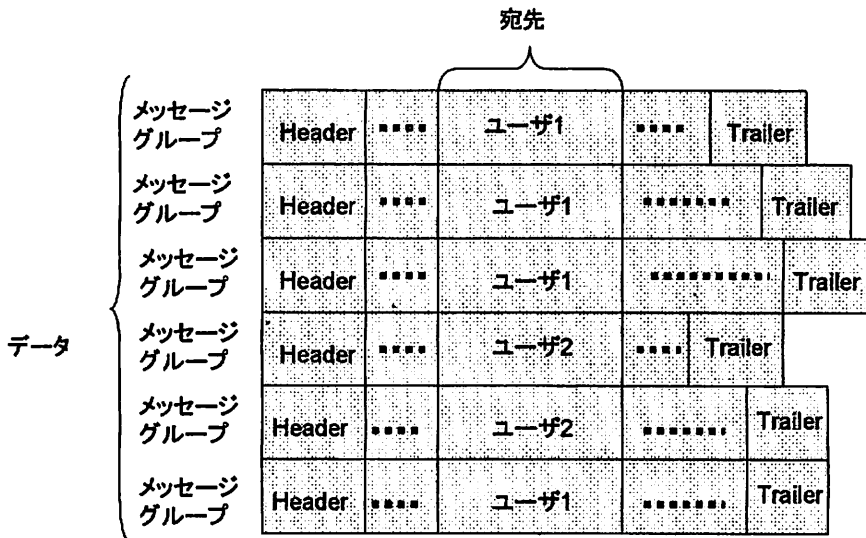


図2 データフォーマット

- ① `cpustat(cpu.use =cpu.usr + cpu.system + cpu.iowait[%]`。 `100 - cpu.idle[%]` と同値)… CPU 使用率を求めることができる。
- ② `cpustat(cpu.iowait[%])` IO 待ち率が求められる。
- ③ `vmstat` リソース割り当て待機中の実行可能プロセス数、IO 等でブロックされている実行可能プロセスの数、スワップアウトされている実行可能なプロセス数、操作したページ数を求めることができる。

平均時からピーク時に推移する際の CPU 使用率を示すグラフを図 3 に示す。

ピーク時において、CPU 使用率(`cpu.use = cpu.usr + cpu.system + cpu.wait`)は、平常時(平均 15[%]程度)に比べ高くなり、平均 70[%]の使用率に達する。以上から、ピーク時におい

て、CPU 使用率は常に 50%を超え、ボトルネックとなっている可能性があるといえる。

CPU 使用率の内訳としては、ユーザプロセスによる CPU 使用率=77%、システムプロセスによる CPU 使用率=11%、IO 待ちによる CPU 使用率=12%であり、ディスクがボトルネックとなっている可能性はないと考える。

なお、EDI システム内部では、仕分・統合および集信および配信が行われているが、実際には、仕分・統合を行う時間がエンドユーザレスポンスに対して支配的である。

メモリ使用率については、メモリ割り当て待ちとなるプロセス数は瞬間的に 1 となる瞬間があるが、CPU 数は 2 であり、ボトルネックとなっている可能性はないと考える。

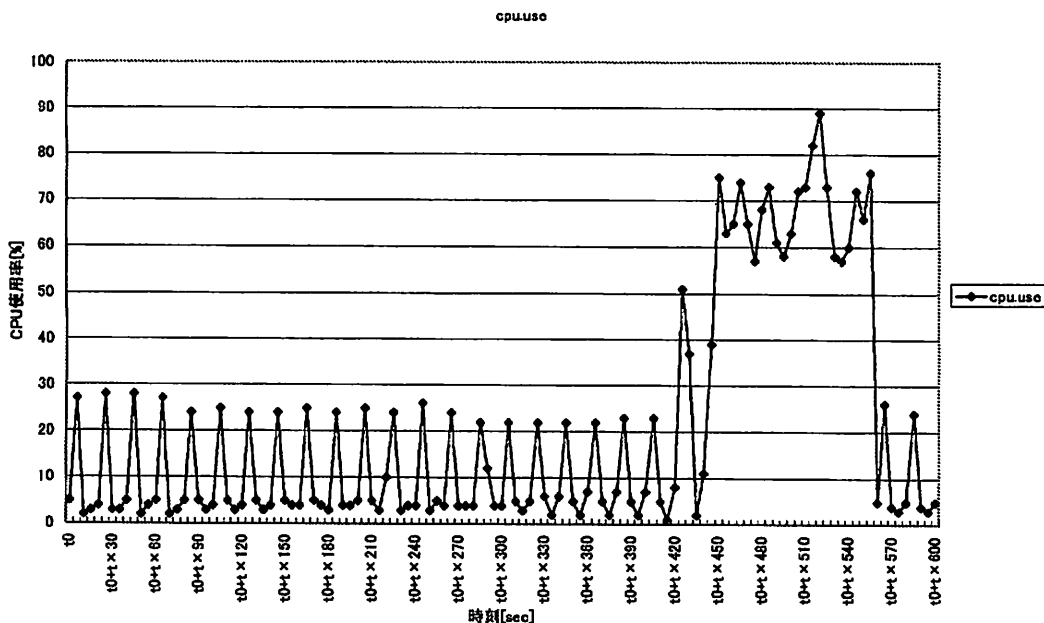


図 3 CPU 使用率

さらに、本 EDI システムでは、CPU 数が 2 であるが、メモリ割り当てを待つプロセス数が 1 となる瞬間が 1 秒程度発生することがあるが、この影響は少ないと考える。

以上より、CPU がボトルネックとなっており、ディスクアクセスはボトルネックになっていないと推測できる。このように、より高速化した EDI システムでは、性能モデルを変える必要がある。

5. 高速化された EDI システムにおけるデータ量によるエンドユーザレスポンスの予測

このような高速化手法を採用したシステムでは、従来のようにデータ量のみから予測を行った場合、正確な予測値を得ることができない。

式(1)について、実データを用いて検証を行った。ある一日にデータを採取し、採取したデータについて、処理時間の短い順にデータ No を与えた。図 4 には、採取したデータのうち一部をサンプルデータとして、サンプル No を新たに与え、横軸にサンプル No、縦軸には、処理時間をとったグラフと、縦軸にデータ量をと

ったグラフを重ね合わせている(図は、一日のデータ中の一部)。

これらのデータの間でデータ量のみから処理時間を予測した期待値が、実測値を近似しているという仮説を立て、処理時間を求める算式を、最小二乗法を用いて決定し、実測の処理時間を用いて、 χ^2 乗検定を行った(有意水準 5%)。結果、仮説は棄却された。

このように、従来のようにデータ量のみから予測を行った場合、正確な予測値を得ることができないといえる。

6. データ量と宛先数によるエンドユーザレスポンス予測のための性能モデル

高速化された EDI システムにおける、仕分の動作は、次のようになる。

- ① データ中の区切りを検索する。
- ② データ中の宛先欄を確認する。
- ③ 次の区切りを検索する。

データ量と処理時間

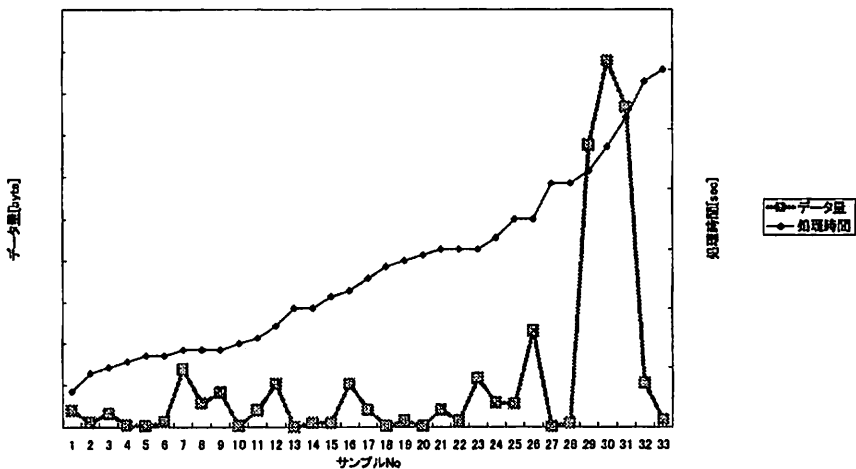


図 4 EDI システムにおけるデータ量と処理時間

- ④ 次の宛先欄を確認する。
- ⑤ ②で得た宛先欄と④で得た宛先欄が異なる場合、③の区切り位置でデータを切り分けて、ディスク中に書き出す。
- ⑥ ②で得た宛先欄と④で得た宛先欄が等しい場合、データを切り分けず、次の区切り位置を検索する。

統合の動作は、次のようになる。

- ① ディスク中に書き出された仕分けされたデータの宛先欄を確認し、同一の宛先欄を持つデータがある配信メールアドレス中のデータに結合する。

これらの動作から、エンドユーザレスポンスの予測において、データ量と宛先数をパラメータとした数式による予測値が、実測値を近似しているという仮説を立てた。

なお、宛先数とは、連続する単位データ中で、宛先の等しい単位データの数として定義した。図2のデータにおいては、宛先に注目し、連続し、かつ、同じ宛先は、ユーザ1、ユーザ2、ユーザ1のように並んでいる。すなわち、図2のデータにおける宛先数は3である。

これらの宛先数や、データ量をパラメータとして用い、エンドユーザレスポンス予測のための性能モデルを構築する。先に示したフローより、以下のような数式を構築した。

$$t = ax + by + c, \quad (2)$$

t エンドユーザレスポンス

x データ量

y 宛先数

a, b, c 定数

7. データ量と宛先数によるエンドユーザレスポンス予測のための性能モデルの検証

式(2)について、実際のデータを用いて検証を行った。ある一日に採取したデータを用いて、採取したデータについて、処理時間の短い順にデータ No を与えた。

図5は、採取したデータのうち一部をサンプルデータとして、サンプル No を新たに与え、横軸にサンプル No、縦軸には、処理時間をとったグラフ①と、縦軸にデータ量をとったグラフ②と、縦軸に宛先数を採ったグラフ③を重ね合わせている(図は、一日のデータ中の一部)。

データ量および宛先数と処理時間

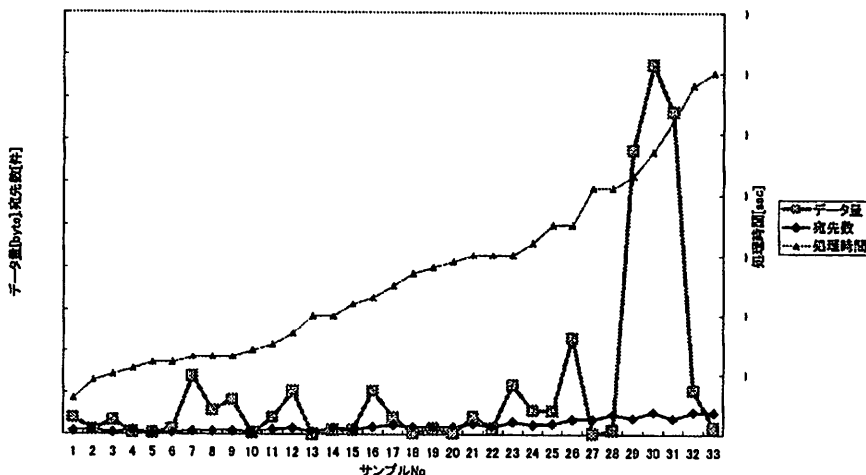


図5 EDI システムにおける宛先数とデータ量と処理時間

(2)式を、最小二乗法を用いて決定し、実測の処理時間を用いて、 χ^2 乗検定を行った(有意水準5%)。結果、仮説は棄却されなかった。

このように、データ量および宛先数から予測を行うことにより、より正確な予測値を得ることができるといえる。これにより、実運用においても、本予測方式を採用する予定である。

8. まとめ

EDI システムにおいて、従来型のエンドユーザレスポンス予測のための性能モデルを、より高速化された EDI システムに対し、適用した場合、正確な性能予測値を得ることができない事例を示した。

このため、宛先数の概念を導入し、より高速化された EDI システムに対応したエンドユーザレスポンス予測のための性能モデルを構築した。本性能モデルについて、実データを用いて、検証を行い、正確な予測値を得ることができることを確認した。

より高速化された EDI システムにおける性能予測モデルは、実際に EDI システムの性能検証に用い、中期的なシステムアップデート計画の立案に用いる予定である。

今後は、人工的なデータを用いて、性能モデルの精度を向上させる。また、データ中の宛先数は、EDI システムにおいて仕分けが実行されることにより判明する値であることから、リアルタイムにデータの処理時間を予測しながらプロアクティブに障害検出を行う方法には適さない。このため、短期的な障害検出手段について検討を行う。

参 考 文 献

- [1] JEITA EC センター, EIAJ-EDI 標準 2001 年版, 2001.
- [2] 佐藤雅之, “データ変換による異なる EDI

標準間のユーザ移行,” 情報処理学会第 68 回全国大会論文集, 第 4 分冊, no. 4-337, 03/2006.

- [3] 北澤 博, “EDI 入門 ビジネスネットワーク・プロトコル”, KK ソフトウェア・リサーチセンター, 1995.
- [4] OBN, EDI 入門, 2001.
- [5] 三菱総合研究所, 平成 16 年度我が国の IT 利活用に関する調査研究 企業間電子商取引事例等に関する調査研究報告書-企業間の情報連携・協調実現のための EDI の導入・利用拡大に向けて-, 2004.
- [6] JEDIC, EDI 取引導入ガイド, 2000.
- [7] JEITA EC センター, EIAJ 取引情報化対応標準, 1999.
- [8] JEITA EC センター, 標準納品システム導入ガイド, 1998.
- [9] JEITA EC センター, EIAJ 版 Web-EDI 導入の手引き (入門編), 1998.
- [10] CII, CII-EDI サービス運用ガイドライン, 1993.
- [11] CII, -CII-EDI サービスを用いた-EDI 導入ガイド, 1995.
- [12] JIPDEC, CII シンタックスルール (JIS-X7012) 解説書, 2000.
- [13] JEDIC, EIAJ 標準準拠オンライン取引導入ガイド, 1991.