情報処理学会研究報告
IPSJ SIG Technical Report

Vol.2010-DPS-145 No.3
Vol.2010-GN-77 No.3
Vol.2010-EIP-50 No.3
2010/11/25

高速メッセージ基盤における低速端末対応方式の提案と評価

佐藤 竜也[†]　鈴木 芳生[†]
花井 知広[†]　直野 健[†]

多数端末を対象とした情報配信システムでは，送達保証と低遅延保証の両立が必要だが，特定の端末の受信が遅れた場合，送達保証を維持すると配信サーバの負荷が増大し，正常端末への低遅延が保証できない．本研究では低速端末発生時にも送達と低遅延保証を両立する方式を提案する．提案方式は，サーバの二重化により送達保証を実現し，低速端末発生時に低速端末への配信を一方のサーバに片寄せすることで正常端末の低遅延を実現する．プロトタイプを用いた評価の結果，提案方式によって，データを HDD に退避する従来方式と比較して，遅延時間の平均値は 20.6 倍以上，最悪値は 2.7 倍以上性能が向上した．従って本方式は低速端末発生時の品質保証に有効である．

Evaluation on a Performance Assurance Method of Messaging Platform for Relatively Slow Client Nodes

Tatsuya Sato,[†] Yoshio Suzuki,[†]
Tomohiro Hanai[†] and Ken Naono[†]

Information delivery system which publishes a message to a large number of client nodes must guarantee message delivery and low latency simultaneously. However, when specific client nodes slow down relatively, the system cannot guarantee low latency for normal client nodes because guaranteed delivery for slow client nodes causes server overload. Therefore, we propose a method to guarantee delivery and low latency simultaneously even when slow client nodes are detected. In order to achieve delivery guarantee, we adopt a dual server configuration (two-node HA cluster), and in order to achieve low latency for other normal client nodes, the system migrates slow client nodes to one delivery server when slow client nodes are detected. Evaluations on prototype show that proposed method can reduce the average latency by 20.6 times and the maximum latency by 2.7 times, compared to the traditional method spooling to disk. Consequently, proposed method is effective for delivery and low latency guarantee in information delivery systems.

1. はじめに

近年，開発生産性の向上やシステムコスト削減といった観点から，システム構築においても外部のサービスや既存システムとの連携を実現するための技術の重要性が高まっている．

システム連携の一形態として，多数端末に情報を送る情報配信システムがある．情報配信システムは，不特定多数の端末に対して同一のメッセージを配信するシステムであり，金融情報の配信サービス，ソーシャルコミュニケーションサービス等で利用される．このような情報配信システムにおいては，配信のリアルタイム性が重要性を増してきている．例えば，金融分野では，配信サービスによって配信される市場データに反応してシステムが自動的に発注を行うアルゴリズム取引が拡がりを見せている．アルゴリズム取引においては，リアルタイム性が重要である．したがって，その前提となっている配信サービスにおいてもリアルタイム性が重要であり，一定の遅延時間内での送達保証(以降，低遅延保証)は必須である．また，アルゴリズム取引の普及に伴って取引の小口化や処理の自動化が進んだ結果，処理対象のメッセージ量は増加の一途をたどっており[1, 2]，配信サービスではこれらの多量のデータに対して，送達保証を実現しなければならない．

情報配信サービスと顧客アプリケーションの接続するシステムの基盤としては，メッセージ基盤(Message Oriented Middleware, MOM)が用いられる．MOM はアプリケーション(以降，AP)間のメッセージング送受信を行うミドルウェアであり，複数 AP 間を連携させる基盤として広く利用されている．MOM としては複数の商用製品やオープンソース実装が存在しており，AP 連携の基盤として，様々な分野において広く利用されている．

ところが，既存技術では，Quality of Service(QoS)保証を実現する機能が不足しているため，今後益々リアルタイム性が必要となる情報配信システムへの適用は困難であった．前述の通り，情報配信システムの QoS 保証としては，低遅延保証が求められることに加えて，信頼性要件として，欠損無くメッセージを送信することを保証する必要がある(以降，送達保証)．従って，配信サービス向けの MOM では低遅延保証と送達保証を両立しなければならない．

しかし，既存技術においては，端末のうち一台の受信性能が低下した場合に，その端末への送達保証を維持しようとすると，未配信データを保持あるいは再送する必要が生じるため，端末への配信を行うサーバの処理負荷が増大する．負荷増大の影響を受け，他の正常な端末に対する遅延の劣化を引き起こしてしまう問題がある．

そのため，本研究では，低速端末発生時にも送達保証と低遅延保証を同時に実現す

[†] 株式会社 日立製作所 中央研究所
Central Research Laboratory, Hitachi, Ltd.

Vol.2010-DPS-145 No.3
Vol.2010-GN-77 No.3
Vol.2010-EIP-50 No.3
2010/11/25

る QoS 方式を提案することを目的とする．

2. 情報配信システムにおける低速端末発生問題と本研究の課題

2.1 情報配信システム

株価情報等の情報配信サービスを実現する情報配信システムでは，データ配信元から発行される情報を，複数の顧客端末に自動配信する．サービスの特性として複数端末に大量の情報を配信する必要があるため，サービスを提供するシステムは，配信サーバがデータ配信元と端末の間を仲介して配信を行うブローカ型アーキテクチャによって実現される．図 1 に，金融情報配信サービスのシステム構成例を示す．配信サーバは，データ配信元である取引所から，銘柄の株価値刻みデータをいったん受信した後で，各顧客端末に対して自動配信する．顧客端末は，配信サーバに対して購読条件を登録しておくことが可能であり，必要な情報のみを受信することが可能である．配信サーバは，購読条件に従って，指定された銘柄情報を，端末までリアルタイムかつ抜け漏れなく送信しなければならない．

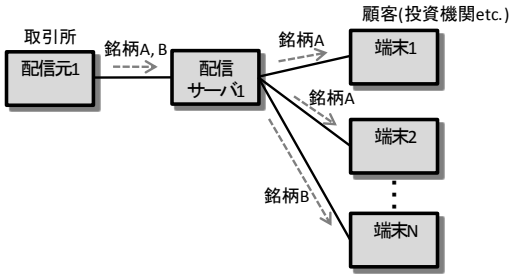


図 1　情報配信システムの例(株価情報配信)

情報配信システムでは，例えば株取引などのリアルタイムな取引における顧客の機会損失を防ぐ必要があるため，信頼性と性能要件を同時に実現しなければならない．

信頼性要件としては，まず，メッセージが欠損無く確実に配信されることを保証する送達保証を満たす必要がある．メッセージの欠損は処理結果の不整合を引き起こすなど障害の原因となるため，送達保証は重要である．さらに，システム継続性の観点から，配信サーバが障害でダウンした場合にも配信処理が継続できることを保証する耐障害保証が必要となる．性能要件としては，一定の遅延時間以内でメッセージを配信することを保証する低遅延保証を満たす必要がある． 以降では，サービスが顧客に対してすべての要件を保証することを Quality of Service(QoS)保証と呼ぶ．

これらの要件を同時に満たすために，配信システムでは，図 2 に示すような冗長化構成(現用/待機構成)をとることで信頼性を保証し，さらにインメモリでの配信処理を組み合わせることで性能を保証する方式が採用されるようになっている．配信サーバは現用系，待機系の 2 台で構成し，両サーバ間のメモリ上でメッセージを冗長保持する．すなわち，送達保証実現のために，配信が完了するまで，待機系配信サーバのメモリ上にメッセージを保持する．また，通常時は現用系サーバが配信を行い，障害発生時には系切替により待機系サーバが配信を引き継ぐことで耐障害保証を実現する．

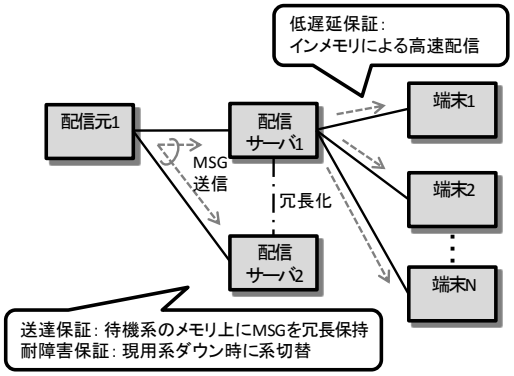


図 2　前提となるメッセージ基盤の高速/高信頼化方式

2.2 従来の低速端末対応方式とその問題点

低速端末発生に対応する従来方式としては, Flow to Disk 方式, 動的負荷均等化方式, 永続化装置利用方式の 3 つが挙げられる(表 1).

Flow to Disk 方式(従来方式(1))は， 配信サーバのメモリから溢れたメッセージを永続化装置に移動する送達保証方式である[3]. Flow to Disk を実現する機能は既存製品/オープンソースにて提供されている[4, 5, 6, 7]. Flow to Disk では，メモリ溢れ発生以降，永続化装置への I/O が発生するため，配信サーバの負荷が増大し，結果としてその配信サーバが担当する他の正常端末への配信遅延を引き起こす.

永続化専用装置利用方式(従来方式(2))は，永続化専用装置を備え，この永続化装置に常にメッセージを永続化し続ける. 低速端末発生時には, 担当する各配信サーバが, 専用装置から該当メッセージを読み込んだ上で，低速端末に対して再送する．永続化専用装置としては既にいくつかの製品が存在している [8, 9].

動的負荷均等化方式(従来方式(3))は，2 台以上の配信サーバを用いて構成し，低速端末発生時にはその配信担当を配信サーバ間で均等に割り当てることで負荷を分散す

Vol.2010-DPS-145 No.3
Vol.2010-GN-77 No.3
Vol.2010-EIP-50 No.3
2010/11/25

る．この方式は PADRES[10, 11]等，研究段階でいくつかの提案がなされている[11, 12].

これらの従来方式には 2 つの問題がある．1 つ目として，低速端末発生時において正常端末の遅延時間が劣化するという問題が挙げられる．従来方式においては，配信サーバの配信担当に低速端末と正常端末が混在する可能性がある．そのため，低速端末発生時には配信サーバの負荷増大による影響を受けるため，他の正常端末への低遅延保証が困難となる．

2 つ目としては，従来方式を適用するとシステム構成コストが高くなるという問題が挙げられる．従来方式(1)(3)では，各配信サーバが永続化装置を備える必要がある．高速に永続化を行うためには，Solid State Drive(SSD)や Redundant Arrays of Independent Disks(RAID)構成がなされたストレージを用いる必要がある．さらに，永続化による処理増大発生時にも低遅延を保証できる，サーバ性能および台数が必要となる．従来方式(2)では，永続化箇所が専用装置に集約されるため永続化装置数は 1 つでよいが，従来方式(1)(3)が低速端末発生によるメモリ溢れ時等，一時的な永続化を行うのに対し，常にメッセージを永続化し続けなければならない．そのため，従来方式(2)では，大容量のキャッシュの搭載等エンタープライズ向けのミッドレンジ/ハイエンドストレージと同等な性能・機能を有する永続化装置を備える必要があると考えられる．これらのストレージのコストは配信サーバのコストに比べて格段に大きいことから，従来方式(2)の構成コストは従来方式(1)(3)に比べても高くなると考えられる．

2.3 低速端末対応における課題

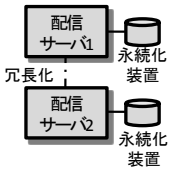
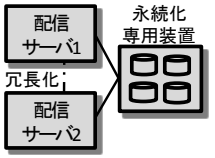
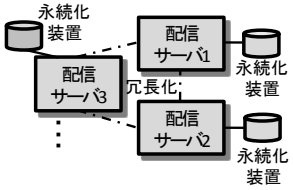
前節で挙げた従来方式の問題点から，低速端末対応においては，正常端末の遅延時間と構成コストに課題がある．

正常端末の遅延時間については，低速端末と正常端末の混在により，正常端末の遅延時間が劣化するため，劣化を回避し低遅延を保証可能な方式が必要となる．

構成コストについては，低速端末対応のための新たなリソースを追加することなく，低コストで送達保証と低遅延時間保証を両立可能な方式が必要となる．特に従来方式では永続化にかかるコストが高いという問題を解消する必要がある．

低速端末対応方式の問題点と課題を表 2 にまとめる．

表 1　従来方式比較一覧

方式	従来方式(1) Flow to Disk 方式	従来方式(2) 永続化専用装置利用方式	従来方式(3) 動的負荷均等化方式
システム構成	 配信サーバ 2 台(各サーバに永続化装置あり)	 配信サーバ 2 台＋永続化専用装置	 配信サーバ 2 台以上(階層化構成/ネットワーク構成，各サーバに永続化装置あり)
低速端末発生時の処理概要	低速端末が発生した配信サーバは，永続化装置に Flow to Disk を実施．	常時，専用装置にメッセージを永続化．低速端末発生時には，専用装置から該当メッセージを読み込んだ上で，低速端末に対して再送．	低速端末を検知して，その配信担当を負荷の軽い任意の配信サーバに移動． メモリ溢れ時については，各サーバが Flow to Disk を実施することで送達保証を実現すると仮定．
正常端末の遅延時間	× 低速/正常端末が混在する可能性有り．	△ 永続化専用装置利用時の排他発生による処理劣化可能性，データ取得時における遅延発生により，正常端末への性能影響可能性あり．	△ 低速/正常端末が混在する可能性有り． (動的負荷分散により，影響は緩和)
構成コスト	△ 各配信サーバに永続化装置が必要．	× 専用の永続化装置要(専用装置では，全メッセージを永続化する容量を備える)．	× 各配信サーバに永続化装置が必要． (Flow to Disk 方式よりサーバ台数多)

Vol.2010-DPS-145 No.3
Vol.2010-GN-77 No.3
Vol.2010-EIP-50 No.3
2010/11/25

表 2　低速端末対応方式の問題点と課題

#	項目	問題点	課題
1	正常端末の遅延時間	低速/正常端末が混在する可能性があり，正常端末への低遅延時間が保証困難.	低速/正常端末が混在による正常端末の遅延時間劣化を回避する方式が必要.
2	構成コスト	低速端末発生時にも送達保証と低遅延時間保証を両立するために十分なリソースを備えた機器が必要となり，構成コストが高くなる.	低速端末発生時に，低コストで送達保証と低遅延時間保証を両立可能な方式が必要.

3. 低速端末対応方式の提案

低速端末発生時にも送達保証と低遅延保証を実現する方式として低速端末対応方式を提案する．本章では，最初に提案する低速端末対応方式の概要を説明した後，検証のために開発したプロトタイプの詳細について説明する．

3.1 低速端末対応方式の概要

提案方式では，配信サーバの二重化により送達保証を実現し，低速端末発生時には低速端末への配信を一方の配信サーバに片寄せすることで，正常端末の低遅延保証を実現する．

提案方式の処理概要を図 3(右)に示す．提案方式は，Flow to Disk 方式(図 3(左))と同様に，配信サーバを二重化(現用/待機系)し，配信サーバ 1 を現用系，配信サーバ 2 を待機系として利用する．片寄処理は，低速端末監視機構によって実現する．ここで，低速端末監視機構は，低速端末の発生検知および，配信サーバへの切替指示を実施することで，片寄処理を実現するモジュールである．ここで，低速端末監視機構は，配信サーバとは別の計算機で動作させてもよいし，あるいは，配信サーバ上で稼働させてもよい．ただし，別の計算機で稼働させる場合であっても，低速端末監視機構は性能監視と切替指示を発行するのみで配信処理を行うわけではないので，配信サーバのような高スペックのサーバは不要であり，安価な PC 等での稼働も可能である．

端末への配信を行う配信サーバ 1 において，低速端末(ここでは端末 2)が発生した場合，低速端末監視装置が発生を検知し，低速端末の低速端末の配信担当を，片寄せ先である配信サーバ 2 に切り替える．切替後は，片寄せ先の配信サーバ 2 が低速端末への配信を行う．この時，低速端末向けメッセージは，配信サーバ 2 が Flow to Disk することで保持する．

従来方式(Flow to Disk 方式)と提案方式の比較を表 3 にまとめる．

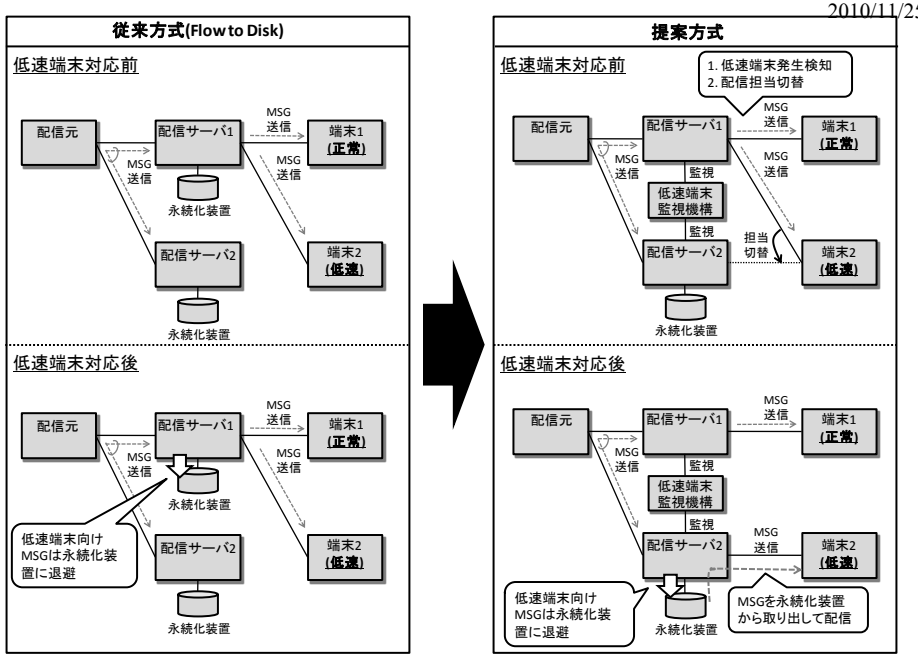


図 3 低速端末対応方式の概要

表 3　提案方式と従来方式(Flow to Disk 方式)の比較

方式	従来方式(Flow to Disk 方式) (図 1 左)	提案方式 (図 1 右)
概要	配信担当は，低速端末発生に関わらず変更なし．低速端末向けのメッセージは，現用系(配信サーバ 1)にて永続化.	低速端末の配信担当を現用系(配信サーバ 1)から待機系(配信サーバ 2)に切り替え．低速端末向けのメッセージは，待機系にて永続化.
構成(永続化装置)	待機系/現用系の両方に永続化装置要	待機系のみ永続化装置要
低速端末監視機構	無し	有り

Vol.2010-DPS-145 No.3
Vol.2010-GN-77 No.3
Vol.2010-EIP-50 No.3
2010/11/25

提案方式を用いることで，表 2 に示した課題を解決することができる．課題と解決方法の対応を表 4 にまとめる．まず，正常端末の遅延時間の課題に対しては，低速端末を特定の配信サーバに片寄せすることで，正常/低速端末の混在の回避が可能となる．混在回避により，正常端末が接続される配信サーバでは，低速端末発生による負荷が生じないため，低遅延保証が可能となる．次に構成コストの課題に対しては，耐障害性保証をするための最小構成である配信サーバ 2 台構成を活用し，低速端末対応のための機器追加を不要とする．さらに特定の配信サーバのみが永続化装置を必要とするため，永続化装置の構成コストが抑制できる．

表 4　低速端末対応方式の課題と解決方法

#	項目	課題	解決方法
1	正常端末の遅延時間	低速/正常端末が混在による正常端末の遅延時間劣化を回避する手法が必要.	低速端末を特定の配信サーバに片寄せすることで，正常/低速端末の混在を回避.
2	構成コスト	低速端末発生時に，低コストで送達保証と低遅延時間保証を両立可能な手法が必要.	・耐障害性保証をするための最小構成である配信サーバ 2 台構成を活用することで，機器の追加を不要とする. ・特定配信サーバのみが永続化装置を必要とする.

3.2 低速端末対応方式の実現方法

評価用のプロトタイプとして，提案方式である低速端末対応方式，及び，従来方式として Flow to Disk 方式を実装した．なお，それぞれのプロトタイプは，標準仕様 Advanced Message Queuing Protocol(AMQP)[13, 14]のオープンソース実装である Apache Qpid(Java 版)[5]をベースとして，機能拡張を行うことで開発した．本節では，提案方式と従来方式の実現方式について説明する．

3.2.1 提案方式の実現方式

プロトタイプでは低速端末の切替機能を以下のように実現した．

(1) システム構成と処理概要

配信サーバの構成としては，図 3(右)と同様の二重化構成(現用/待機構成)とし，現用系である配信サーバ 1 を正常端末用の配信サーバとして，待機系である配信サーバ 2 を低速端末の片寄せ先の配信サーバとして割り当てた．

従って，低速端末が発生する前は，配信サーバ 1 が全端末の配信を担当し，その間配信サーバ 2 は未配信メッセージの保持のみを実施する．両配信サーバは配信元から，並列配信されたメッセージを受け取り，そのメッセージの配信が完了するまで，両配信サーバのメモリ上に保持する．ここで配信サーバ 2 のメッセージ保持量がメモリ上限を超えた場合には，古い順にメッセージを永続化装置に退避する．

低速端末が発生した場合には，低速端末監視機構が，配信サーバ 1 上の低速端末発生を検知し，その配信担当を配信サーバ 2 に切り替える．

(2) 低速端末監視機構

本実装では，低速端末発生をメッセージキューの蓄積量のみから判断する．具体的には，配信サーバ 1 は，メッセージ新着時に各端末向けのメッセージキューのサイズを監視し，キューサイズがしきい値を超過した時，低速端末監視機構にその旨を通知する．低速端末監視機構は通知を受け，該当キューからメッセージを受け取っている端末を低速端末として検知し，その配信担当を配信サーバ 2 に切り替える処理を行う．

(3) 低速端末検知しきい値の設定

今回の実装では，低速端末検知しきい値は予め設定した固定値を用いることとした．低速端末の検知切替処理はメモリが溢れる前に完了しなければならないため，適切なしきい値を設定する必要がある．そのため，しきい値は，メモリ容量とメッセージ流量だけでなく，切替処理に要する時間も考慮して設定する必要がある．

4. 評価

4.1 実験方法

4.1.1 実験環境

評価として，図 4 に示す 3 台のマシン構成を用いて実験を行った．本構成では，2 台のマシンにそれぞれ現用系/待機系配信サーバを割り当て，残りの 1 台のマシンに性能測定用クライアントプログラム(配信元および端末)と低速端末監視機構を割り当てた．ここでクライアントマシンに低速端末監視機構を割り当てた理由は，測定のために端末との時刻を合わせる必要があったためである．低速端末監視機構の実施する処理は性能監視と切替指示発行のみであるため，処理発生に伴う影響は無視できる．本実験では，現用系である配信サーバ 1 に正常端末，低速端末を 1 基ずつ割り当てた．マシン性能と利用したソフトウェアは表 5 に示す通りである．

Vol.2010-DPS-145 No.3
Vol.2010-GN-77 No.3
Vol.2010-EIP-50 No.3
2010/11/25

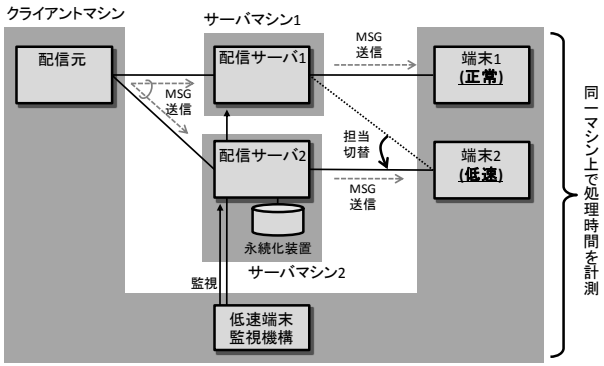


図 4 評価環境(図は提案方式)

表 5 評価に利用したマシンおよびソフトウェアのスペック

#	名称	スペック, バージョン, 設定
1	CPU	Intel ® Core™ 2 Quad プロセッサ Q6700(8MB, 2.66GHz x4)
2	メモリ	4GB (DDR2-SDRAM メモリ, 800MHz)
3	HDD (永続化先)	160GB, シリアル ATA ドライブ 7200rpm (サーバマシン: 永続化時の信頼性確保のため書き込みキャッシュ無効化)
4	NIC	1GbE インタフェース(Intel PRO /1000(ドライバ: e1000e, PCI Express:2.5GB/s: Width x 1)
5	OS	サーバマシン: Linux 2.6.27.5-117.fc10.x86_64(Fedora10) クライアントマシン: Linux 2.6.29.4-167.fc11.x86_64 (Fedora11)
6	Java	Sun Java HotSpot(TM) 64-Bit Server VM 1.6.0_13
7	Qpid	qpid-java-0.5 を機能拡張
8	Berkeley DB	je-3.3.82
9	MTU	1500Byte

4.1.2 評価方法

(1) 実験における処理の流れ

評価実験は，配信元から配信サーバに対して，指定した件数・レートでメッセージを送付し続けることで行う．実験では，送付の中途で低速端末を発生させ，低速端末発生前後の遅延時間を継続して測定することで, 提案方式の有効性を検証した. なお，低速端末の発生は，送付の中途で片方の端末のメッセージ受け取りレートを強制的に低下させることで，シミュレートする．

(2) 評価項目と測定方法

実験における評価項目とその測定方法について述べる．本実験では遅延時間，送達保証，切替処理時間の 3 項目を評価した．

遅延時間の評価では，正常端末において低速端末発生前後の遅延時間を比較することで，低遅延保証が実現できるかを検証する．ここで，遅延時間は，配信元でのメッセージ送信時と端末受信時のタイムスタンプの差をとることで測定した．

送達保証については，低速端末発生時であっても，全てのメッセージが欠損なく配信されていることを確認するが，これは，受信端末においてメッセージに付与した通し番号に欠損がないかどうかで判断した．

切替処理時間の検証については，切替処理の実行に必要となる複数のコマンドの実行時間の総和をとることで計測した．コマンドの内訳は，現用系から切替を開始するメッセージ番号を取得するコマンド，待機系からその番号より前のメッセージを削除するコマンド(ここまでを前処理と呼ぶ)，低速端末の接続先を変更するコマンド(接続先変更処理)，現用系のキュー要素をクリアするコマンド(後処理と呼ぶ)である．

4.1.3 実験条件

4.1.2 節に示した方法に従い，提案方式と従来方式の比較評価実験を行った．
実験条件としては, 金融分野での利用を想定し, メッセージサイズ 4KB, 16KB, 64KB, 256KB の 4 条件で測定を行った．なお，メッセージ送付件数はメッセージサイズに従って定め，配信元の送付レートは，Flow to Disk 時に HDD 処理性能がボトルネックとならない値を事前検証して設定した．

4.2 実験結果

(1) 遅延時間の評価

図 5, 6 に低速端末発生時の正常端末の遅延時間(100 件毎の平均値)を示す. グラフの横軸はメッセージ番号(送付した順に付与された番号)であり，縦軸は配信元が送信したメッセージが正常端末に受信されるまでの遅延時間を表す．縦軸は，切替発生前の正常端末の平均遅延時間を 1 とした時の相対値とした．

Vol.2010-DPS-145 No.3
Vol.2010-GN-77 No.3
Vol.2010-EIP-50 No.3
2010/11/25

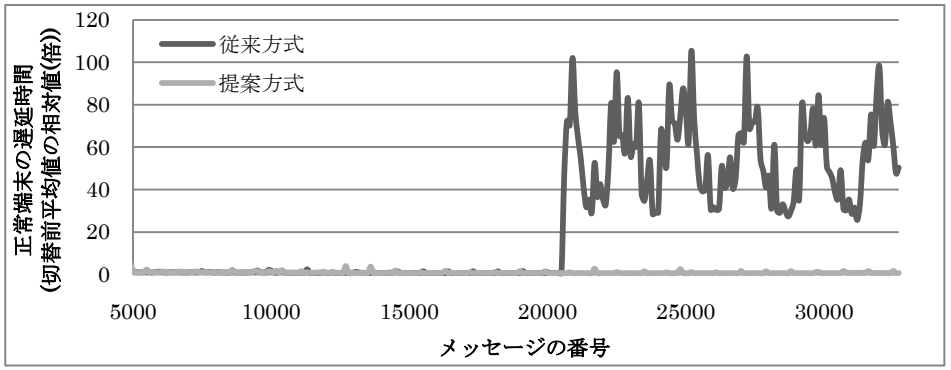


図 5　低速端末発生時の正常端末の遅延時間比較(メッセージサイズ 4KB)
(図中の遅延時間は 100 件毎の平均値を掲載)

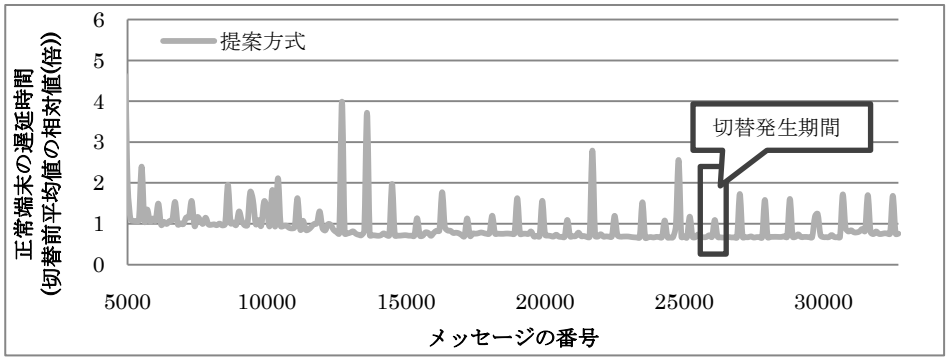


図 6　低速端末発生時の提案方式の遅延時間と切替処理影響(メッセージサイズ 4KB)
(図中の遅延時間は 100 件毎の平均値を掲載)

図 5 は，従来方式と提案方式における正常端末の遅延時間の比較結果を示す．従来方式では, 20600 番目のメッセージを超えたあたりで低速端末発生を検知し, 以降 Flow to Disk を実施しており, 正常端末の遅延時間(100 件毎の平均値)は最大で切替前の 105 倍に悪化している．

図 6 は，提案方式における正常端末の遅延時間のみを示したグラフであり，図中の枠内で切替処理が実施されている．切替発生後の遅延時間を比較すると，従来方式とは異なり，遅延時間の大きな変化は見られない．ここで，切替発生期間の枠内に見られるピークは，切替処理による劣化である．一方，枠外で一定件数毎に見られるピーク値は，Java のマイナーGarbage Collection (GC)処理発生に伴う遅延時間の劣化である．これらを比較すると，切替処理による遅延時間の劣化が，GC 処理による遅延時間の劣化に比べて小さいことが確認できる．

次に，表 6 に切替後における正常端末の遅延時間平均値の比較結果を示す．表中の遅延時間は，切替発生前の正常端末の平均遅延時間を 1 とした時の相対値とした．提案方式では，切替後にも正常端末の性能劣化はみられなかった．また，正常端末の遅延時間は，切替前に比べて切替後の方が短くなっているが，これは，切替により低速端末が配信サーバ 2 に片寄せされたので，配信サーバ 1 上が担当する端末数が減り，処理負荷が減少したためである．以上より，切替完了後は低速端末発生前と同等な遅延時間を保証できることが確かめられた．具体的な数値を比較すると，メッセージサイズ 4KB のときには，従来方式では切替前の 55.30 倍と遅延時間が大きく悪化にしたのに対して，提案方式では切替前の 0.81 倍と遅延時間の悪化は見られなかった．すなわち，遅延時間の平均値については，提案方式によって，従来方式の 69.5 倍の性能向上が確認できた．他のメッセージサイズにおいても，提案方式によって 20.6 倍以上性能が向上し，提案方式の効果が非常に大きいことが確認できた．

表 6　切替後における正常端末の遅延時間の平均値

メッセージサイズ	遅延時間の平均値（切替前平均値の相対値）(倍)			切替後遅延時間比 従来/提案 (倍)
	従来方式	提案方式		
	切替後	切替発生中	切替後	
4KB	55.30	4.60	0.81	69.5
16KB	25.20	2.48	0.80	33.1
64KB	30.44	0.93	0.79	35.9
256KB	15.37	0.93	0.83	20.6

Vol.2010-DPS-145 No.3
Vol.2010-GN-77 No.3
Vol.2010-EIP-50 No.3
2010/11/25

図 7 にメッセージサイズ 4KB の場合の遅延時間最悪値の比較結果を示す．縦軸は，切替発生前の正常端末の平均遅延時間を 1 とした時の相対値とした．図 5, 6 とは異なり，遅延時間の平滑化は行っていない．図中の丸枠は各方式の遅延時間の最悪値を示す．

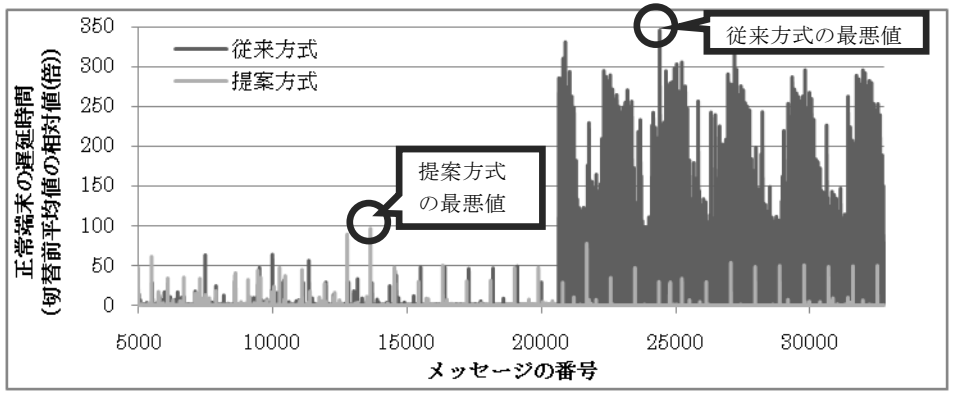


図 7　低速端末発生時の正常端末の遅延時間最悪値比較(メッセージサイズ 4KB)

表 7　正常端末の遅延時間の最悪値

メッセージサイズ	遅延時間の最悪値 (切替前平均値の相対値)(倍)		最悪遅延時間比 従来/提案（倍）
	従来方式	提案方式	
4KB	342	125	2.7
16KB	161	47	3.4
64KB	93	12	7.8
256KB	33	9	3.6

表 7 にメッセージサイズ毎の遅延時間最悪値の比較結果を示す．図 7 中の注釈に示す通り，提案方式では，最悪値は切替発生期間外で発生した．これはマイナーGC 処理発生による性能劣化であった．そのため，切替発生中に，同時にマイナーGC 処理が発生した場合を想定し，切替発生中の最悪値と切替発生期間外の最悪値の和を算出し，提案方式の最悪値とした．最悪値を比較すると，メッセージサイズ 4KB の場合，従来方式では切替前の平均値に比べ，342 倍悪化するのに対して，提案方式では，124 倍の劣化に留まった．すなわち，提案方式によって，遅延時間の最悪値は，従来方式に比べて 2.7 倍性能向上した．他のメッセージサイズの場合においても，提案方式によって，従来方式の 3.6～7.8 倍性能が向上することを確認した．

(2)　切替処理時間の評価

図 8 に切替処理時間を示す．切替処理時間は 62ms～184ms となった．切替処理時間内訳をみたところ，切替処理時間は前/後処理時間に応じて変化することがわかった．前/後処理は主にメッセージの削除命令により構成されることから，切替処理時間は削除するメッセージ件数に応じて定まると考えられる．

(3)　実験結果のまとめ

以上より，提案方式では，低速端末対応完了後の正常端末の遅延時間平均値が，低速端末発生前の遅延時間平均値を超えないことを確認した．提案方式によって，遅延時間の平均値については 20.6 倍以上, 最悪値については, 2.7 倍以上性能が向上した．従って，提案方式が，低速端末発生時の正常端末の低遅延保証に有効であることが確認できた．

また，提案方式のメッセージ到着状況を確認したところ，正常，低速端末共に全メッセージを欠損なく受信していた．従って，本方式はメッセージの送達保証という要件を十分に満たすことができることを確認した．

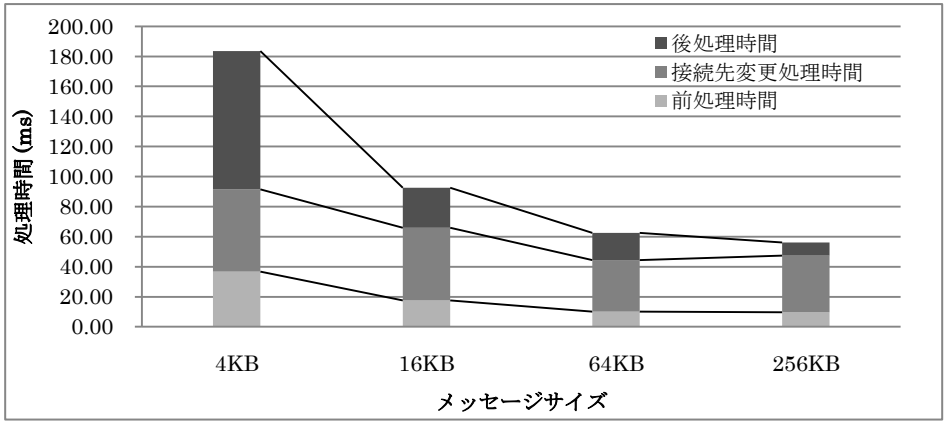


図 8 切替処理時間の内訳

Vol.2010-DPS-145 No.3
Vol.2010-GN-77 No.3
Vol.2010-EIP-50 No.3
2010/11/25

5. おわりに

5.1 結論

本研究では，低速端末発生時にも送達保証と品質保証を同時に実現するために，低速端末への配信を特定の配信サーバに片寄せする，低速端末対応方式を提案した．プロトタイプを用いた評価実験の結果，従来方式では，低速端末対応完了後の遅延時間が著しく劣化するのに対して，提案方式では，性能劣化がないことを確認した．具体的には，低遅延保証の達成目標である，低速端末対応完了後の正常端末の遅延時間平均値については，低速端末発生前の遅延時間の平均値を超えないことを確認した．提案方式によって，遅延時間の平均値については 20.6 倍以上，最悪値については 2.7 倍以上の性能向上が確認でき，提案方式が低遅延保証に大きな効果があることがわかった．また，送達保証の目標についてはすべての端末が，欠損無くメッセージを受け取ることを確認できた．

以上の結果から，提案方式は，低速端末発生時であっても，低遅延保証と送達保証を両立することが可能であり，金融情報配信など特に低遅延保証が要求されるような情報配信サービスへ適用可能な見通しを得た．

5.2 今後の課題

今回は片寄機能として，低速端末の切替機能を実現したが，今後は正常端末の切替機能も実現し，低速端末数がより動的に変動する環境での方式の有効性を検証する．

さらに今回のプロトタイプでは，低速端末検知しきい値は，事前に定めた値を用いた．運用・構築コストの削減のためには，適切なしきい値が自動的に設定されることが望ましい．そのため，今後はしきい値の自動チューニングについても実現検討を行う．

参考文献

1) Kevin McPartland: Reinforcing Your Operations Post Credit Crunch/Liquidity Crisis, Hedge Fund Operations & Technology, (2008).
2) OPRA: Updated Traffic Projections 2010 & 2011, http://www.opradata.com/specs/Updated_Traffic_Projections_2010_2011.pdf, (2010).
3) Apache Software Foundation: Apache Qpid Java Broker Design Flow to Disk, http://qpid.apache.org/java-broker-design-flow-to-disk.html#JavaBrokerDesign-FlowtoDisk-FlowsafterFlowToDisk/ .
4) IBM: WebSphere MQ V7.0, http://www-06.ibm.com/software/jp/websphere/integration/wmq/ .
5) Apache Software Foundation: Apache Qpid, http://qpid.apache.org/ .
6) Apache Software Foundation: Apache ActiveMQ, http://activemq.apache.org/ .
7) JBoss Community: HornetQ, http://www.jboss.org/hornetq/ .
8) 29West: Ultra Messaging® for the Enterprise 2.0, http://www.29west.co.jp/products/ume/ .
9) Tervela: Tervela TPE Persistence Engine, http://www.tervela.com/tpe/ .
10) Hans-Arno Jacobsen, Alex Cheung, Guoli Li, Balasubramaneyam Maniymaran, Vinod Muthusamy, and Reza Sherafat Kazemzadeh: The PADRES Publish/Subscribe System, IGI Global, 43 pages, (2009).
11) Alex King Yeung Cheung, and Hans-Arno Jacobsen: Dynamic Load Balancing in Distributed Content-based Publish/Subscribe, In Proceedings of ACM Middleware 2006, pp.141-161, (2006).
12) Abhishek Gupta, Ozgur Dogan Sahin, Divyakant Agrawal, and Amr El Abbadi: Meghdoot: content-based publish/subscribe over P2P networks, In Proceedings of ACM Middleware 2004, pp. 254-273, (2004).
13) AMQP Working Group: Advanced Message Queuing Protocol, http://www.amqp.org/ .
14) John O'Hara: Toward a Commodity Enterprise Middleware, ACM Queue, vol. 5, No. 4, pp.48-55, (2007).