HiTactix-BSD 連動システムを応用した大規模双方向ストリームサーバの設計と実装

竹内 理[†] レ・モアル ダミエン[†]

高速アクセス網の普及，および iDC 等のサーバコンテンツ提供者以外の場所におけるサーバ管理サービスの普及にともない，高速 I/O 機能，I/O レート保証機能，管理支援機能を兼ね備えたストリームサーバ（大規模双方向ストリームサーバ）のニーズが高まっている．上記ストリームサーバの実現のため，我々は，HiTactix（ストリーム処理に特化した専用 OS）上に，新規の周期 I/O 方式（サイクリックパイプライン I/O 方式）およびディスク I/O スケジューラ（サイクリックディスク I/O スケジューリング方式）を実装した．さらに，HiTactix と BSD/OS を 1 ノード上で連動させるシステム（HiTactx-BSD 連動システム）上にストリームサーバアプリケーションを実装した．本稿では，上記ストリームサーバのシステムの概要，および新規に実装した周期 I/O 方式とディスク I/O スケジューラの概要につき述べる．さらに，実装したストリームサーバの性能評価結果の概要について述べ，本ストリームサーバが汎用 OS 上のストリームサーバと比して，5.25〜11.0 倍の I/O 性能の向上，および 200 倍の I/O レート保証性能の向上を実現していることを示す．

Design and Implementation of a Highly Scalable Bi-directional Stream Server Using Cooperating HiTactix-BSD System

Tadashi Takeuchi[†] and Damien Le Moal[†]

Recently, fast access network services and server management services in iDCs (Internet Data Centers) become wide spread. In order to make full use of these services, demand for next generation stream servers (*highly scalable bi-directional stream server*) is increasing. Next generation stream servers should provide three mechanisms: fast I/O mechanim, constant rate I/O mechanism, and management mechanism. In order to provide these mechanisms, the following two activities should be achieved. First, we implemented our orignal periodic I/O method (*Cyclic Pipeline I/O method*) and disk I/O scheduler (*Cyclic Disk I/O Scheduler*) on top of HiTactix which is our original OS suitable for stream data processing. Second, we implemented a stream server application on top of *cooperating HiTactix-BSD system*. Cooperating HiTactix-BSD system enables HiTactix and BSD/OS to coexist on one SMP hardware. In this paper, the system architecture of our stream server is first described. Then, the original periodic I/O method and disk I/O scheduler are explained. Finally, evaluation results of our stream server are shown. These results confirmed that our stream server can execute I/O from 5.25 to 11.0 times faster than conventional servers, and can execute constant rate I/O 200 times more precisely.

1. はじめに

近年，FTTH（Fiber To The Home）網，CATV 網，ADSL 網をはじめとする高速アクセス網が普及しつつある．それにともない，今までアクセス網の帯域が不十分で実現できなかった大規模ストリーミングサービスの実現可能性が高まりつつある[14]．大規模ストリーミングサービスの実現には，大容量ストリームデータを同時に多数のクライアントに配信可能なストリームサーバ，すなわち高速 I/O 機能を備えたストリームサーバが必要である．

また一方で，近年，維持管理の容易さ等の理由により，ストリームサーバが iDC（Internet Data Center）のようなコンテンツ提供者以外の場所に設置されることが多くなってきている[11]．上記のようなストリームサーバは，クライアントへのストリームデータの配信（ダウンロード）ばかりでなく，コンテンツ提供者からのストリームデータのアップロードを受理する（双方向のストリームデータ通信に対応する）必要があ

† 株式会社日立製作所システム開発研究所
Systems Development Laboratory, Hitachi Ltd.

る．そのため，アップロード処理の実行中にも，ダウンロード処理のI/Oレートを保証する機能が必要になる．また，iDC内におけるサーバ群の一括管理を実現するため，管理を支援する機能（たとえばネットワーク経由での管理を実現するSNMPのような通信プロトコルスタックや，定期的な監視アプリケーションの起動を実現するcrontabのようなアプリケーション）もストリームサーバは備えている必要がある．

以上のように，高速I/O機能，I/Oレート保証機能，管理支援機能を兼ね備えたストリームサーバ（以後，「大規模双方向ストリームサーバ」と呼ぶ）の要求が近年高まっているが，従来のストリームサーバは，汎用OSまたはストリーム処理に特化した専用OS上で実現されているため，上記すべての機能を同時に提供することが困難であった．すなわち，汎用OS上で実現されているストリームサーバは，高速I/O機能やI/Oレートの保証機能の実現が困難であり[8]，専用OS上で実現されているストリームサーバは，管理支援機能の提供のために膨大なソフトウェア開発コストが必要であった[10]．

著者らは，HiTactix-BSD連動システム[13]を用いることにより，上記3機能を同時に提供する大規模双方向ストリームサーバを実現した．本稿では，まず上記大規模双方向ストリームサーバのシステム構成概要を述べ，本ストリームサーバが上記3機能を同時に提供していることを明らかにする．さらに，高速I/O機能，およびI/Oレート保証機能を実現するためにHiTactix（著者らが独自開発したストリーム処理向け専用OS）[6),12),13)]上に実装した，サイクリックパイプラインI/O方式，およびサイクリックディスクI/Oスケジューリング方式の概要につき述べる．最後に，本ストリームサーバの高速I/O機能，およびI/Oレート保証機能の定量的な評価結果につき述べる．

2. システム構成概要

1章で述べたとおり，本ストリームサーバは，
機能A）高速I/O機能，
機能B）I/Oレート保証機能，
機能C）管理支援機能，
の同時提供を目的に設計，実装されている．機能B）は，ネットワークI/Oレート保証機能（機能B1）と，アップロード処理のディスクI/O実行中にも，ダウンロード処理のディスクI/Oレートを保証する機能（機能B2）の2つからなる．

上記実現のため，本ストリームサーバは**図1**に示すシステム構成をとっている．

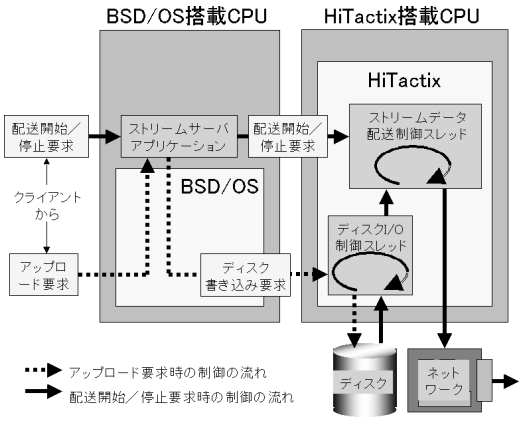


図1 システム構成
Fig. 1 System architecture.

(1) 本ストリームサーバはSMPハードウェア（2CPU搭載）上で動作する．SMPハードウェアの1CPUを占有して汎用OS（BSD/OS☆）が，他CPUを占有してストリーム処理向き専用OS（HiTactix）が動作する（HiTactix-BSD連動システム）．

(2) 「ストリームサーバアプリケーション」はBSD/OS上で動作する．BSD/OSの機能を用いることにより，本ストリームサーバ上でcrontabやSNMP等の管理支援機能を提供するソフトウェアを動作させることができる．

(3) 「ストリームサーバアプリケーション」は，メディアデータの配送開始/停止要求を受理した場合，HiTactix上の「ストリームデータ配送制御スレッド」に対して当該要求を転送する．開始要求発行から停止要求発行までの間，「ストリームデータ配送制御スレッド」は周期的に，定レートでのディスク読み出し（「ディスクI/O制御スレッド」に対するディスク読み出し要求の発行），およびネットワーク送信を繰り返す．本ディスクI/Oおよびネットワーク I/Oは，3章で述べるサイクリックパイプラインI/O方式により，高速に（低いCPU負荷で），かつ厳密に一定周期にて実行されることが保証される．

(4) 「ストリームサーバアプリケーション」は，ストリームデータのアップロード要求を受理した場合，アップロードデータのディスクへの書き込み要求を「ディスクI/O制御スレッド」に対して発行する．「ディスクI/O制御スレッド」

☆ 本稿で表記されているBSD/OS等の各製品名は各社の登録商標です．

は，4 章で述べるサイクリックディスク I/O スケジューリング方式を用いることにより，当該ディスク書き込みが発生しても，「ストリームデータ配送制御スレッド」のディスク読み出しレートが変動しないことを保証する．

本ストリームサーバは，上記 (2) により機能 C）を，(3) により機能 A）機能 B1）を，(4) により機能 B2）を提供していることが分かる．3 章および 4 章では，機能 A），機能 B1）を実現すべく新規に本ストリームサーバに実装したサイクリックパイプライン I/O 方式，および機能 B2）を実現すべく新規に実装したサイクリックディスク I/O スケジューリング方式の概要につき述べる．

3. サイクリックパイプライン I/O 方式

本章では

(1) 高速 I/O 機能，

(2) ネットワーク I/O レートの保証機能，

を実現するサイクリックパイプライン I/O 方式の概要につき述べる．まず，汎用 OS における I/O 方式の概要につき説明する．次に本方式の概要を説明し，本方式が上記 2 機能を実現可能であることを定性的に明らかにする．

3.1 汎用 OS における I/O 方式

汎用 OS における I/O 方式の概要を**図 2** に示す．

汎用 OS では，ディスク I/O およびネットワーク I/O の実行を要求する API は別々である．そのため，ストリームデータを配送するアプリケーションは以下の手順で I/O を行う．

(1) アプリケーションはディスク I/O 実行要求を発行し，ディスクからストリームデータをユーザバッファに読み込む．この際に，カーネルバッファとユーザバッファとの間でメモリコピーが発生する．

(2) アプリケーションは，ネットワーク I/O 実行要求を発行し，ユーザバッファ内のストリームデータをネットワークに送信する．この際にもカーネルバッファとユーザバッファとの間でメモリコピーが発生する．

(3) ユーザバッファサイズはストリームデータと比較して小さいため，ユーザバッファサイズ単位で上記 (1)，(2) に示す I/O 処理を繰り返し周期実行する．

汎用 OS における I/O 方式を用いた場合，以下の問題が発生する．

(1) バッファ間のメモリコピー，および API（システムコール）呼び出しが多発し，I/O 実行の際に要する処理オーバヘッドが大きくなる．

(2) 汎用 OS は，アプリケーションの周期スケジューリング機能を持たないため，アプリケーションの駆動周期がゆらぐ可能性がある．この際，ネットワーク I/O 実行要求の発行周期もゆらぐため，ネットワーク I/O レートが一定にならない．

3.2 サイクリックパイプライン I/O 方式の概要

サイクリックパイプライン I/O 方式の方式概要を**図 3** に示す．

サイクリックパイプライン I/O 方式では，BSD/OS カーネル内の「HiTactix ドライバ」と HiTactix カーネル内の「ストリームデータ配送制御スレッド」の連動により実現する．

(1) 「HiTactix ドライバ」はアプリケーションに対してストリームデータの配送開始（start）および配送停止（stop）を要求可能な API を提供する．上記 API が呼び出されると，「HiTactix ドライバ」は当該要求を「ストリームデータ配送制御スレッド」に転送する．

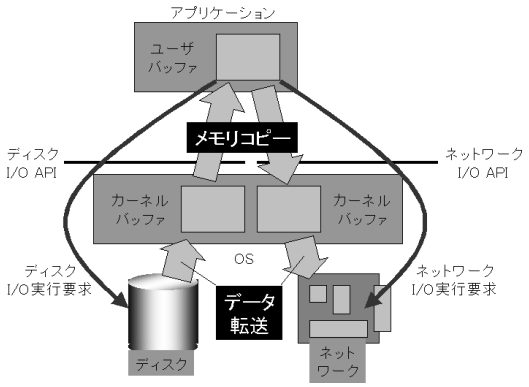


図 2 汎用 OS における I/O 方式の概要
Fig. 2 Conventional OS I/O mechanism.

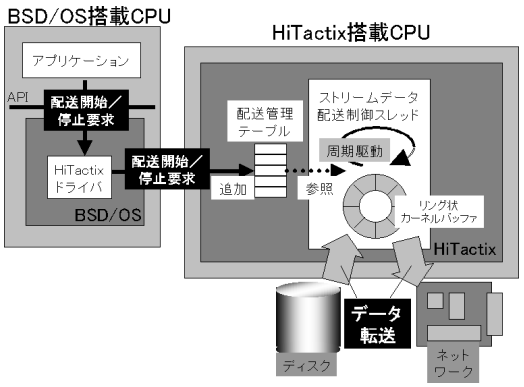


図 3 サイクリックパイプライン I/O 方式
Fig. 3 Cyclic Pipleline I/O Mechanism.

(2) 「ストリームデータ配送制御スレッド」は配送開始要求を受理すると，配送管理テーブルのエントリの追加を行う．配送管理テーブルのエントリは，上記要求ごとに存在する．そして各エントリには，読み出すべきファイルに関する情報，およびネットワーク送信先となるクライアントのアドレス情報，レート情報等を格納する．以後，「ストリームデータ配送制御スレッド」は周期的に駆動し，配送管理テーブルに格納されているすべてのエントリに関して，1 周期分のストリームデータのディスクからの読み出し，および当該データのネットワーク送信を行う．本 I/O はリング状のカーネルバッファとデバイスとの間でストリームデータを転送することにより行う．本周期実行は，配送停止要求を受理するまで継続する．

上記のような構成にすることにより，以下の利点が得られる．

(1) アプリケーションからの API 呼び出しは，配送開始時と停止時に 1 回ずつ行えばよいため，API 呼び出しに要する処理オーバヘッドを削減できる．

(2) I/O の際にメモリコピーが発生しないため，メモリコピーに要する処理オーバヘッドも削減できる．

(3) BSD/OS 上のアプリケーションではなく，Hi-Tactix のアイソクロナススケジューラ[12]により周期駆動を保証されているスレッドが I/O を実行する．そのため，BSD/OS 上のアプリケーションのスケジューリングに関係なく，I/O 実行の周期性の保証，結果としてネットワーク I/O レートの保証が実現できる．

4. サイクリックディスク I/O スケジューリング方式

本章では，大容量データのディスクへの書き込み要求が発生した際にも，ディスクからの読み出しレートを保証するサイクリックディスク I/O スケジューリング方式を説明する．まず，汎用 OS におけるディスク I/O スケジューリング方式について説明する．次に本スケジューリング方式の概要につき説明し，本方式が上記を実現可能であることを定性的に明らかにする．

4.1 汎用 OS におけるディスク I/O スケジューリング方式

汎用 OS では，アプリケーションからのディスク I/O 要求を FIFO 順に処理する．そのため，**図 4** に示すように，定レートでの（図中「読み出し周期」で示される周期で周期的に）ストリームデータの読み出し要求を発行している際に，大容量データのディスク書き込み要求が発生した場合，ストリームデータの読み出し実行が大きく遅延する可能性がある．この結果，ストリームデータの読み出しレートの保証ができなくなる．

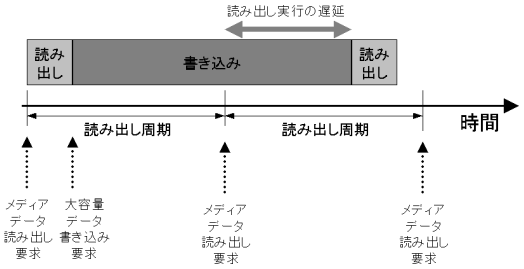


図 4 汎用 OS におけるディスク I/O スケジューリング方式
Fig. 4 Conventional OS disk I/O scheduling mechanism.

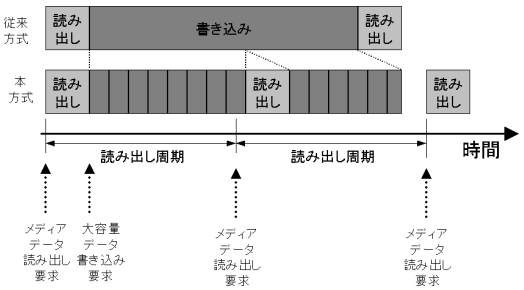


図 5 サイクリックディスク I/O スケジューリング方式
Fig. 5 Cyclic disk I/O scheduling mechanism.

4.2 サイクリックディスク I/O スケジューリング方式の概要

サイクリックディスク I/O スケジューリング方式では，**図 5** に示すように，大容量データのディスクへの書き込み要求を，複数の小容量データ（現状の実装では最大 128 KB）のディスクへの書き込み要求に変換する．そして，ディスク書き込み処理時には，1 回の処理が完了するたびに，次の読み出しを実行すべき時間に到達しているか否かの判定を行う．本判定により，読み出し実行の最大遅延時間は，1 回の書き込み処理実行時間以下になり，大容量データの書き込みが発生した際にも，ディスクの読み出しレートを保証できる☆．

サイクリックディスク I/O スケジューリング方式

☆ ディスクのヘッドシーク時間等によりディスク読み出しや書き込み時間が揺らいでも，本方式は，1 周期あたりのディスクの読み出し量（読み出しレート）を一定に保つ．1 周期あたりのディスクの書き込み量を変動させることにより，上記揺らぎによる読み出しレートの変動を防いでいる．

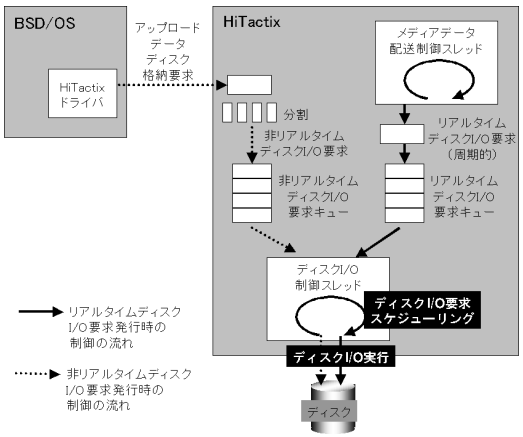


図 6 ディスク I/O 制御スレッド
Fig. 6 Disk I/O Control Thread.

は「ディスク I/O 制御スレッド」を用いて実現されている．

「ディスク I/O 制御スレッド」の動作概要を**図 6** に示す．「ディスク I/O 制御スレッド」は，定レートディスク I/O 要求（以後，「リアルタイムディスク I/O 要求」と表記）と，それ以外のディスク I/O 要求（以後，「非リアルタイムディスク I/O 要求」と表記）を区別して（別々の要求キューを用いて）受け付ける．

「ストリームデータ配送制御スレッド」は周期的にリアルタイムディスク I/O 要求を発行する．また，「HiTactix ドライバ」は，アップロードされたストリームデータのディスク書き込み要求を非リアルタイムディスク I/O 要求として発行する．非リアルタイムディスク I/O 要求を要求キューにエンキューする際には，当該要求を複数の小容量のディスク I/O 要求に分割しておく．

「ディスク I/O 制御スレッド」は，「リアルタイムディスク I/O 要求キュー」「非リアルタイムディスク I/O 要求キュー」にキューイングされている要求群を適切にスケジューリングすることにより，リアルタイムディスク I/O の定レート実行を保証する．

以下，「ディスク I/O 制御スレッド」が実行するディスク I/O 要求群のスケジューリング方法につき説明する．

「ディスク I/O 制御スレッド」は**図 7** に示すフェーズ遷移を行う．そして，各フェーズごとに以下に示す動作を行うことにより，リアルタイムディスク I/O 要求の定レート実行を保証する．

リアルタイムディスク I/O 実行フェーズ 本フェーズは「リアルタイムディスク I/O 要求キュー」から要求をデキューし，当該要求に応じたディスク I/O 実行を行う．「リアルタイムディスク I/O 要求キュー」が空になったら，非リアルタイムディスク I/O 実行フェーズに遷移する．

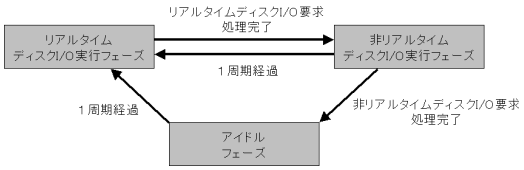


図 7 ディスク I/O 制御スレッドのフェーズ遷移
Fig. 7 Disk I/O Control Thread phase transitions.

非リアルタイムディスク I/O 実行フェーズ 本フェーズは「非リアルタイムディスク I/O 要求キュー」から要求をデキューし，当該要求に応じたディスク I/O 実行を行う．「非リアルタイムディスク I/O 要求キュー」が空になったらアイドルフェーズに遷移する．「非リアルタイムディスク I/O 要求キュー」が空でなくても，リアルタイムディスク I/O 実行フェーズ開始から一定時間（「ストリームデータ配送制御スレッド」の駆動周期，現状の実装では 256 ms）が経過したら，リアルタイムディスク I/O 実行フェーズに遷移する．本フェーズ遷移により，非リアルタイムディスク I/O の連続実行により，リアルタイムディスク I/O のレートが低下することを防いでいる（図 4，図 5 参照）．

アイドルフェーズ 本フェーズではディスク I/O を実行しない．リアルタイムディスク I/O 実行フェーズ開始から一定時間（「ストリームデータ配送制御スレッド」の駆動周期）が経過したら，リアルタイムディスク I/O 実行フェーズに遷移する．

5. 性能評価

本章では，実装した大規模双方向ストリームサーバの高速 I/O 機能，および I/O レート保証機能の定量的な評価を行うために構築した実験システムと，実験結果の概要につき述べる．

5.1 高速 I/O 機能の評価

5.1.1 実験システム概要

高速 I/O 機能の評価のために構築した実験システムを**図 8** に示す．

表 1 に示した構成を持つストリームサーバに対して，1 ストリームあたり約 8.8 Mbps のストリームデータをディスクから読み出しネットワークに送信する要求をクライアントから発行する．要求ストリーム数を変動させた場合に，ストリームサーバが実際にネットワークに送出するストリームデータのレートの変動を測定

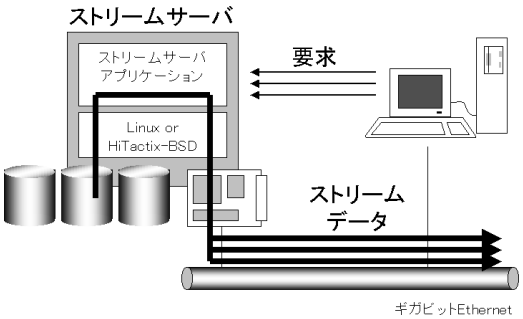


図 8 高速 I/O 機能評価用実験システム
Fig. 8 Experimental system architecture for evaluatiing fast I/O mechanism.

表 1 測定条件
Table 1 Measurement conditions.

条件	内容
本体	PC/AT 互換機 （PentiumIII 600 MHz 搭載）
SCSI ボート	LSI53C895 チップ搭載ボート
SCSI ディスク	Ultra2 対応 HDD （連続アクセス時 27 MB/s）
Ethernet ボード	Alteon AceNIC Ethernet ボード （1000Base-SX Ethernet ボード）

した．そして，ストリームサーバが実際に送出するストリームデータのレートの総計（以下，「送出レート」と表記）が，クライアントが要求しているストリームデータの配送レートの総計（以下，「要求レート」と表記）より下回りはじめるレートを測定することにより，当該ストリームサーバの I/O 性能を評価した．

実験は，HiTactix-BSD 連動システム上に構築したストリームサーバを用いた場合と，Linux 上に構築した☆ストリームサーバを用いた場合の両方について行い，結果を比較した．

5.1.2 実験結果概要

実験結果を**図 9** に示す．グラフの横軸は要求レートを，縦軸は送出レートを示す．

グラフから明らかなように，Linux 上のストリームサーバは，要求レートが 80 Mbps を超えると，送出レートが要求レートを下回るのに対して，HiTactix-BSD 連動システム上のストリームサーバは要求レートが 420 Mbps を超えても，要求レートと送出レートが厳密に一致している．すなわち，HiTactix-BSD 連動システム上のストリームサーバは，Linux 上のストリームサーバと比して最低でも 5.25 倍以上の I/O 性能向上を実現していることが分かる．

☆ 本来は BSD/OS 上に構築したストリームサーバを用いるべきだが，BSD/OS がギガビット Ethernet ボードをサポートしていなかったため，Linux を用いた．

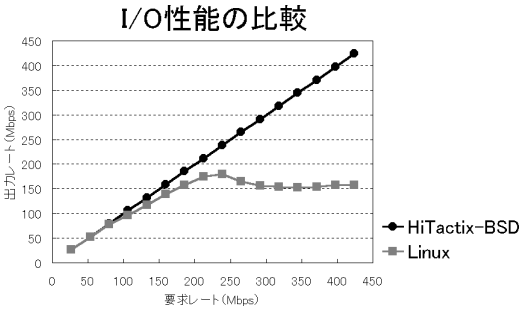


図 9 高速 I/O 機能評価実験結果
Fig. 9 Fast I/O mechanism evaluation result.

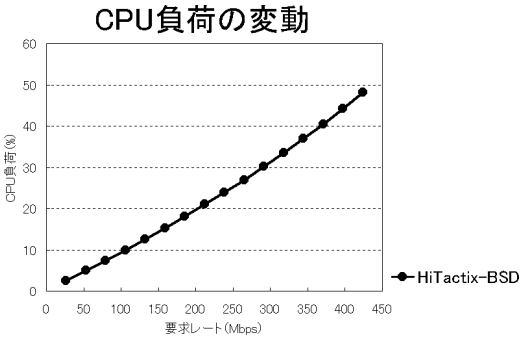


図 10 HiTactix-BSD 連動システムの CPU 負荷の変動
Fig. 10 Cooperating HiTactix-BSD system CPU load.

図 9 では，420 Mbps を超える要求レートについての測定は行わなかった．これは SCSI ディスク（実験では 3 台使用）のハードウェア性能の限界により，ストリームサーバがそれ以上のレートでストリームデータを読み出すことが不可能であったためである．HiTactx-BSD 連動システム上のストリームサーバの I/O 性能をより正確に見積もるため，HiTactix-BSD 連動システムを用いた場合のみ，図 8 におけるストリームサーバの CPU 負荷の変動もあわせて測定した．

測定結果を**図 10** に示す．グラフの横軸が要求レートを，縦軸が CPU 負荷を示す．グラフから，CPU 負荷は要求レートにほぼ比例して増加すること，および 880 Mbps の要求レート時に CPU 負荷がほぼ 100%になると予測できることが明らかになった．このことから，HiTactix-BSD 連動システム上のストリームサーバは最大 880 Mbps の I/O 性能を持ちうる，すなわち Linux 上のストリームサーバと比して最大で約 11.0 倍の I/O 性能の向上を実現しうることが分かった．

5.2 I/O レート保証機能の評価

5.2.1 実験システム概要

I/O レート保証機能の評価のために構築した実験システムを**図 11** に示す．

表 1 で示した構成を持ち，かつ HiTactix-BSD 連

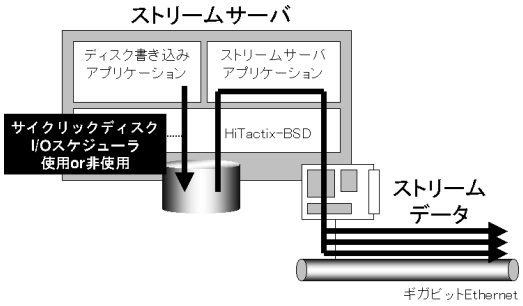


図 11 I/O レート保証機能評価用実験システム
Fig. 11 Experimental system architecture for evaluating Constant Rate I/O Mechanism.

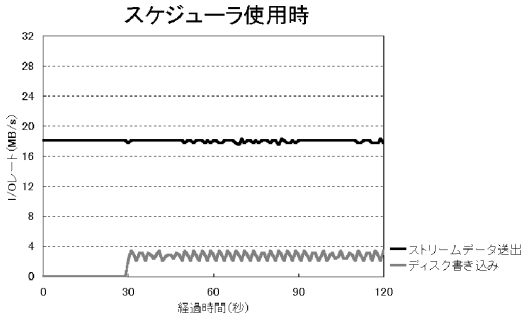


図 12 I/O レート保証機能評価実験結果（1）
Fig. 12 Constant Rate I/O Mechanism evaluation result (1).

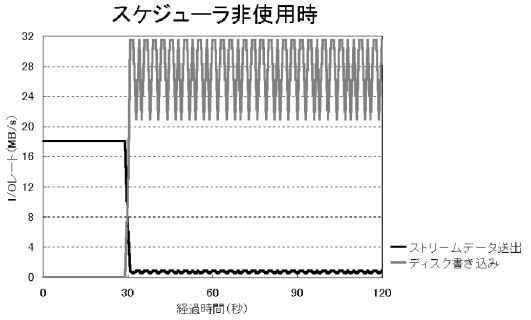


図 13 I/O レート保証機能評価実験結果（2）
Fig. 13 Constant Rate I/O Mechanism evaluation result (2).

動システムを搭載したストリームサーバ上で，約 18 MB/s のレートでストリームデータをディスクから読み出し，ネットワークに送出させた．30 秒間経過後，さらにディスク書き込みアプリケーション（約 10 MB のデータのディスクへの書き込み要求を連続して発行する）を起動した．そして，ディスクへの書き込み処理により上記ストリームデータの送出レートがどの程度揺らぐかを測定し，I/O レート保証性能を評価した．ストリームデータの送出レートの揺らぎは，以下の式で与えられる値 J の時間平均を求めることにより測定した．

$$J = |Real\_Rate - Exp\_Rate| / Exp\_Rate \times 100$$

上記で，*Real_Rate* は単位時間（今回の測定では 1 秒）あたりにストリームサーバが実際に送出したストリームデータの量，*Exp_Rate* はストリームサーバが単位時間あたりに送出すると期待されるストリームデータ量を示す．

実験は，サイクリックディスク I/O スケジューリング方式を用いた場合とそうでない場合（HiTactix が上記ディスクへの書き込み要求を，4 章で述べたディスク I/O 制御スレッドを介して処理する場合と介さずに処理する場合）の両方につき行い，結果を比較した．

5.2.2 実験結果概要

実験結果を**図 12** および**図 13** に示す．グラフの横軸が経過時間を，縦軸が I/O レート（ストリームデータの送出レート，またはディスク書き込みレート）を示す．

グラフから，サイクリックディスク I/O スケジューリング方式を用いた場合は，前章で述べたディスク書き込みを発生させても 0.48%程度しかストリームデータの送出レートの揺らぎが発生しないことが明らかになった．それに対し，サイクリックディスク I/O スケジューリング方式を用いなかった場合は，96%程度の送出レートの揺らぎ（落ち込み）が発生している．すなわち，サイクリックディスク I/O スケジューリング方式を用いることにより，200 倍程度の I/O レート保証性能の向上が実現できていることが分かる．

6. 関連研究

ストリームサーバへの適用を狙い，高速 I/O 機能や I/O レート保証機能を OS に組み込む，またはアプリケーションレベルで実現することを試みた研究例は数多く存在する．本章では，代表的な上記研究例と本研究との比較につき述べる．

6.1 高速 I/O 機能に関する関連研究

高速 I/O 機能を OS に組み込む研究例としては，Fbufs [4]，WDM（Win32 Driver Model）Kernel Streaming Architecture [7]，splice 機能[5]，RoadRunner [9] の I/O システム等が知られている．

Fbufs は，カーネル空間とユーザ空間との間のデータコピーを，ページテーブル情報の操作により物理メモリコピーなく実現する．Fbufs を用いた OS では，3.1 節で述べた方法と同様な方法でアプリケーションが I/O を実行する．そのため，3.1 節で述べたとおり，システムコール呼び出しの多発による処理オーバヘッドの増大や，アプリケーションの駆動周期の揺らぎに

よるネットワークI/Oレートの揺らぎが発生しうる．本研究の大規模ストリームサーバは，サイクリックパイプラインI/O方式を用いて，上記問題を解決している．

WDM Kernel Streaming Achitectureは，デバイスドライバ間におけるデータの受渡しを物理メモリコピーなく実現する．しかしながら上記は，デバイスドライバレベルでデータの受渡しを実行するため，アプリケーションはファイルシステムやネットワークプロトコルスタックを利用できない．また，同一デバイスドライバを使用してI/Oを実行するアプリケーションが複数存在した場合，各アプリケーションが実行するI/Oのレートを保証できない（たとえば，大量のディスク書き込み要求を発行するアプリケーションと，同一ディスクから定レートでのストリームデータの読み出しを要求するアプリケーションを同時に動作させても，後者が望んだレートにてディスクを読み出せない可能性がある）．本研究の大規模双方向ストリームサーバは，アプリケーションがHiTactixの提供するファイルシステムやプロトコルスタックを利用できる．また，サイクリックディスクI/Oスケジューリング方式を用いて，I/Oレートの保証も実現できる．

splice機能は，ネットワークプロトコルスタックとファイルシステム間におけるデータの受渡しを物理メモリコピーなく実現する．そのため，アプリケーションからファイルシステムやネットワークプロトコルスタックの利用は可能である．しかしながら，splice機能も，同一デバイスドライバを使用してI/Oを実行するアプリケーションが複数存在した場合に，各アプリケーションが実行するI/Oのレートを保証できないという問題を解決していない．

RoadRunnerのI/Oシステムは，各アプリケーションを実行するスレッドとは別に，I/Oを専門に実行するカーネルスレッドを設け，当該スレッドに周期I/Oを実行させている．上記カーネルスレッドを厳密に一定周期で駆動することにより，高速I/O機能のほかに，ある程度のI/Oレート保証機能も提供可能である．しかしながら，RoadRunnerは既存の汎用OSとの互換性がないため，ストリームサーバを実現する際には，管理支援機能を実現するアプリケーションをすべて実装する必要が生じ，ソフトウェア開発コストが膨大になる．本研究の大規模ストリームサーバは，BSD/OS上の管理支援機能を実現するソフトウェアを変更なく使用できるため，上記ソフトウェア開発コストが必要ない．

6.2 I/Oレート保証機能に関する関連研究

I/Oレート保証機能をOSに組み込む，またはアプリケーションレベルで実現する研究例としては，Tigerビデオファイルサーバ[2]，ΔLディスクスケジューラ[3]，NemesisのDDM（The Device Data-Path Module）[1]等が知られている．

Tigerビデオファイルサーバは，アプリケーションレベルでI/Oレート保証機能を実現している．各スケジューリングスロットごとに，1ストリーム分のI/Oしか同時実行しないことを仮定しているうえ，各ストリームのデータレートが異なる場合におけるレート保証方式を完全に実現していない．そのため，本研究の大規模双方向サーバでは実現可能である，アップロード処理実行中にもストリームデータのダウンロードのレートを保証する機能や，さまざまなレートのストリームデータを同時にダウンロードする機能を実現できない．

ΔLディスクスケジューラは，ディスクI/Oスケジューリング方式を変更し，非リアルタイムディスクI/O混在時にも，リアルタイムディスクI/Oのレートを保証可能にしている．リアルタイムディスクI/Oに要する時間を最悪ケースを想定して予測し，その予測時間に基づき非リアルタイムディスクI/Oの処理量を決定する．そのため，リアルタイムディスクI/Oに要する時間のばらつきが大きく正確に予測できない場合，非リアルタイムディスクI/Oの処理量が小さくなる可能性がある．サイクリックディスクI/Oスケジューリング方式では，実際に要したディスクI/O時間を基に，非リアルタイムディスクI/Oの処理量を決定しているため，上記処理量が小さくなることはない．

NemesisのDDMは，ディスクのデバイスドライバレベルでI/Oレート保証を実現するディスクI/Oケジューラを実現している．しかしながら当該ディスクI/Oスケジューラは，非リアルタイムディスクI/O要求を分割して処理しないため，大容量データの非リアルタイムディスクI/O要求が到達した際に，リアルタイムディスクI/O要求の処理時間が大きく遅延する可能性がある．サイクリックディスクI/Oスケジューリング方式は，非リアルタイムディスクI/O要求を分割して処理することにより，上記を防いでいる．

7. ま と め

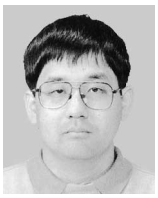
本稿では，HiTactix-BSD連動システム上に構築したストリームサーバの構成の概要につき述べた．本ストリームサーバは，サイクリックパイプラインI/O方

式，サイクリックディスクI/Oスケジューリング方式を用いて高速I/O機能，I/Oレート保証機能を提供している．また，HiTact-BSD連動システムを用いて，上記機能を維持しつつ管理支援機能を提供することも可能にしている．さらに本稿では，本ストリームサーバの性能評価の概要についても述べた．性能評価の結果，本ストリームサーバはLinux上のストリームサーバと比して，5.25〜11.0倍程度のI/O性能の向上，および200倍程度のI/Oレート保証性能の向上を実現していることが明らかになった．

参 考 文 献

1) Barham, P.R.: A Fresh Approach to File System Quality of Service, *7th International Workshop on Network and Operating System Support for Digital Audio and Video*, pp.119–128 (1997).
2) Bolosky, W.J., Fitzgerald, R.P. and Douceur, J.R.: Distributed Schedule Management in the Tiger Video Fileserver, *16th ACM Symposium on Operating Systems Principles*, pp.212–223 (1997).
3) Bosch, P.: Real-Time Disk Scheduling in a Mixed-Media File System, *6th IEEE Real Time Technology and Applications Symposium*, pp.23–32 (2000).
4) Druschel, P. and Peterson, L.L.: Fbufs: A High-Bandwidth Cross-Domain Transfer Facility, *14th ACM Symposium on Operating System Principles*, pp.189–202 (1993).
5) Fall, K. and Pasquale, J.: Improving Continuous Media Playback Performance with In-Kernel Data Paths, *International Conference on Multimedia Computing and Systems*, pp.100–109 (1994).
6) Iwasaki, M., Takeuchi, T., Nakano, T. and Nakahara, M.: Isochronous Scheduling and its Application to Traffic Control, *19th IEEE Real-Time System Symposium*, pp.14–25 (1998).
7) Microsoft: WDM Kernel Streaming Architecture (1998). http://www.microsoft.com/hwdev/desinit/csa1.htm
8) Miller, F.W., Keleher, P. and Tripathi, S.K.: General Data Streaming, *19th IEEE Real-Time System Symposium*, pp.232–241 (1998).
9) Miller, F.W. and Tripathi, S.K.: An Integrated Input/Output System for Kernel Data Streaming, *SPIE/ACM Multimedia Computing and Networking*, pp.57–68 (1998).
10) 新井利明ほか：ナノカーネルによる異種OS共存技術「DARMA」の提案，第59回全国大会講演論文集（1），pp.139–140 (1999).
11) 松原 敦ほか：ASP/IDCの実像，日経バイト，No.208, pp.94–111 (2000).
12) 竹内 理ほか：連続メディア処理向きOSの周期駆動保証機構の設計と実装，情報処理学会論文誌，Vol.40, No.Mar, pp.1204–1215 (1998).
13) 竹内 理ほか：OS接続モジュールSymbioseを用いたBSD-HiTactix連動システムの設計と実装，情報処理学会研究報告，Vol.2000-OS-83, pp.31–36 (2000).
14) 阿蘇和人：ブロードバンドに命を吹き込め，日経コミュニケーション，No.343, pp.94–111 (2001).

(平成13年 9月 6日受付)
(平成13年11月14日採録)



竹内 理（正会員）

昭和44年生．平成4年東京大学理学部情報科学科卒業．平成6年同大学大学院理学系研究科情報科学専攻修士課程修了．同年（株）日立製作所システム開発研究所入社．連続メディア処理向きマイクロカーネルの研究，特にリアルタイムスケジューリング方式，リアルタイム通信方式，異種OS共存技術，ストリーミングサービスアーキテクチャの研究に従事．

レ・モアル ダミエン

1972年生．1995年ENSEEIHT（National Engineering School of Electrotechnics, Electronics, Informatics and Hydraulics of Toulouse, France）卒業．2000年京都大学大学院情報学研究科通信情報システム専攻修士課程修了．同年（株）日立製作所システム開発研究所入社．連続メディア処理向きマイクロカーネルの研究，特にリアルタイムスケジューリング方式，サービス品質保証可能なシステムアーキテクチャ，ストリームサーバ向けミドルウェアの研究に従事．