

論文の内容の要旨

論文題目

複数 OS を活用した高コストパフォーマンスを実現する
ストリームサーバアーキテクチャの提案

氏名 竹内 理

商用ストリーム配信サービスの出現に伴い、低コストでありながらも、高い配信性能や QoS 保証性能を達成できるコストパフォーマンスが高いストリーム配信サーバに対するニーズが高まりつつある。しかし、既存のストリーム配信サーバは、汎用 OS 上のアプリケーション層で配信機能と管理機能を実現していることが多く、アプリケーション層の最適化だけで、高い配信性能や QoS 保証性能を達成することが困難であった。また、ストリーム専用 OS を利用し、その上で配信機能と管理機能を実現するアプローチもありうるが、開発環境が充実していない専用 OS 上で膨大な量のアプリケーション、管理ツールの開発が必要となるために開発コストが増大する、という課題があった。

このような課題を解決し、ストリーム配信サーバのコストパフォーマンスの飛躍的な改善を図るためには、汎用 OS と複数 OS を活用したストリームサーバアーキテクチャを新規に提案し、両 OS の利点を兼ね備えさせることが解決策の一つとして考えられる。本研究では、上記実現を目指し、外付け I/O エンジン方式という新しいストリーム配信サーバの構築方式を提案する。さらに、当該アーキテクチャに基づいて既存市販ストリームサーバと機能互換なストリーム配信サーバを実装し、その性能向上効果や開発コード量削減効果などを定量的に評価する。

上記アーキテクチャに基づいてストリーム配信サーバを実装しても、一定量の専用 OS のモジュール（特に、I/O デバイスドライバ）、及び専用 OS 上のアプリケーションの開発は

不可避となる。そのため、この開発効率を改善も、ストリーム配信サーバのコストパフォーマンス改善に大きく役立つ。そこで、本研究では、さらに、専用 OS と専用 OS 上の開発効率を改善できる OS デバッガとして、「軽量仮想計算機モニタを利用した OS デバッガ」の提案と試作を行い、さらなるストリーム配信サーバのコストパフォーマンス改善を目指す。そして、上記デバッガを用いることで、開発環境の安定稼働、ログ&リプレイ機能などの開発効率の向上に寄与する機能を提供できると共に、このような機能を現実的な CPU 負荷で実現できることを示す。

外付け I/O エンジン方式は、既存の市販ストリーム配信サーバの機能互換性を維持しつつ、当該サーバの配信性能の向上、配信品質保証機能追加を少ない開発工数で実現するストリーム配信サーバの構成方式である。本方式は、汎用 OS 上で動作する既存の市販ストリームサーバをベースとして使用するが、配信性能の向上、配信品質保証のため、そのサーバの配信機能のみを専用 OS（筆者らが開発した HiTactix など）搭載サーバに移植、汎用 OS 搭載サーバと専用 OS 搭載サーバを連動させて、一つのストリーム配信サーバを構築する。本方式の最適な実現方式を設計するため、まず、既存のストリーム配信サーバのモジュール構成を調査した。その結果、市販ストリームサーバは、クライアントからの配信開始・停止要求を受け付ける配信要求処理モジュール、当該要求到達を契機に管理処理を行う管理処理モジュール、当該要求到達を契機にストリームデータの I/O スケジューリングを行う配信制御モジュール、当該スケジューリングに応じて実際の I/O 処理を実行する配信ドライバモジュールから通常なっていることがわかった。さらに、上記 4 モジュールのうちどのモジュールを専用 OS 搭載サーバに移植すれば移植量、配信性能、連動オーバーヘッド等を最適化できるかを検討した結果、配信制御モジュールと配信モジュールのみを専用 OS に移植し、汎用 OS 搭載サーバと専用 OS 搭載サーバを、配信スケジューリング開始・停止を行うインタフェースで接続するのが最善であることがわかった。Darwin ストリームサーバ（Apple 社が提供する市販ストリーム配信サーバ）と HiTactix 搭載の外付け I/O エンジンを実方式に基づき連動させたところ、上記連動は 9.1K 行程度のコード追加で実現できること、連動オーバーヘッドによるストリーム配信性能の劣化は 1%以下に抑えられること、配信性能は既存の Darwin ストリーム配信サーバと比べて 5 倍程度向上できること等を確認できた。

「軽量仮想計算機モニタを用いた OS デバッガ」は、専用 OS の特に I/O デバイスドライバや、専用 OS 上のアプリケーションの開発効率向上の実現を目的としている。特に、デバッグ環境の安定稼働保証、多様な OS や I/O デバイスドライバ開発への適用、デバッグ時に高い I/O 性能達成、の 3 条件の同時充足ができないのが従来デバッグ環境の課題であり、これらの同時充足を目指した。さらに、提案した軽量仮想計算機モニタにロギング&リプレイ機能を追加するための実装方式を提案し、タイミングクリティカルなバグのデバッグ効率の向上を実現した。

提案したデバッグ方式は、従来のソフトウェアリモートデバッグ方式の改良である。しかし、従来方式と異なり、ターゲットマシン上でリモートデバッグ機能も内部に持つ軽量

仮想計算機モニタが動作する。軽量仮想計算機モニタは、リモートデバッグ機能が使用するハードウェア資源のみを仮想化する部分ハードウェアエミュレーション機能を持つ。リモートデバッグ機能が使用するハードウェア資源はデバッグ対象 OS が直接アクセスしないため、OS の異常動作時にも、デバッグ環境の安定稼働が保証できる。また、軽量仮想計算機モニタが OS に提供するインタフェースは実ハードウェアインタフェースと互換性があるため、様々な OS のデバッグに使用できる。さらに、NIC や HBA のような高速 I/O デバイスに関しては、軽量仮想計算機モニタでエミュレーション処理を行わず、デバッグ対象 OS が直接アクセスを実行するので、多様な I/O デバイスドライバ開発への適用が可能になる上、デバッグ時の高い I/O 性能達成も可能になる。実装した軽量仮想計算機モニタの I/O 性能の評価を行った結果、従来の Hosted Virtual Machine Monitor と比べて 5.4 倍の I/O 性能は達成し、現実的な CPU 負荷で上記 3 条件を充足するデバッグ環境が実現できることがわかった。

さらに、軽量仮想計算機モニタを用いた OS デバッガに、ログ&リプレイ機能を追加するため、その実装方式の提案をおこなった。軽量仮想計算機モニタを用いた OS デバッガでは、デバッグ対象の OS が NIC や HBA などのハードウェアに直接アクセスを行い、デバイスレジスタアクセスや DMA 転送時に仮想計算機モニタを介さない。ロギング&リプレイ機能の追加を行うためには、これらの履歴の保存・再生が必要になるため、軽量仮想計算機モニタの起動契機付与方式、及びデバッグ対象 OS による直接アクセス動作の追跡方式を新規に設計・実装する必要がある。起動契機付与方式、及び直接アクセス動作の追跡方式の設計は DMA 転送履歴の保存・再生の際に特に問題となる。本研究では、NIC 受信時の動作を中心に設計・実装を行った。NIC ドライバ動作をモデル化して上記方式検討を行った結果、(1) デバッグ対象 OS (NIC ドライバ) 割り込み要因レジスタ参照時、受信インデックスレジスタ更新時に軽量仮想計算機モニタの起動契機を与える、(2) 受信ディスクリプタ、受信済みインデックスレジスタの更新履歴の追跡により、1 回の割り込みにおいて受信処理された DMA 転送アドレス、サイズ、データ群の取得を行う方式にて履歴の保存・再生が可能であることがわかった。さらに、実装した軽量仮想計算機モニタによるロギング&リプレイ機能の実用性を評価するため、ネットワーク受信処理の履歴保存に要する CPU 負荷の測定を行った。評価の結果、ロギング&リプレイ処理を行うと、VMWare によるネットワーク受信処理と比較しても約 38%程度の CPU 負荷の上昇に抑えられ、現実的な CPU 負荷でログ&リプレイ機能の実装が動作することがわかった。

最後に、外付け I/O エンジン方式及び軽量仮想計算機モニタを用いた OS デバッガによる、ストリーム配信サーバのコストパフォーマンス改善効果の見積りを行った。コストパフォーマンスの見積りは、想定している規模のストリーム配信サービスを提供するために、サービス事業者が負担すべきコストの総計の見積り値を比較することにより行った。見積りの結果、上記 2 方式を使用することにより、従来のストリーム配信サーバと比べて、34%~59%のコストパフォーマンスの改善が達成できる見込みがあること、特に、性能向上によ

るハードウェア購入・保守・運用維持コストの低減量が大きく、専用 OS 使用によるソフトウェア購入・保守コスト（ストリーミング配信サーバベンダのソフトウェア開発費用）の増大量を上回ることを十分期待できることがわかった。