

マルチメディア処理のためのソフトウェア基盤に関する研究

メディア処理基盤プロジェクト

大久保 英嗣、毛利 公一

Abstract: メディア処理基盤プロジェクトでは、リアルタイムシステム、分散オペレーティングシステム、コンテキストの変動に適応可能なソフトウェア基盤の3つを柱に研究を進めている。本プロジェクトでは、これらの研究の成果を基に、様々な環境におけるマルチメディア処理を効率良く行うためのソフトウェア基盤の構築を目指している。

1. はじめに

マイクロプロセッサのコスト性能比の向上により、様々な組み込みデバイスが開発・実用化されている。一方、インターネットのブロードバンド化や無線通信技術などの進展・普及により、それらのデバイスをネットワークに接続して、今まで以上に広範な分野で多様なサービスが提供されるようになってきている。特に、動画や音声といった連続メディアデータを扱うマルチメディアアプリケーションの構築が活発に行われている。このようなマルチメディア処理においては、連続メディアデータの特性や使用環境によって、実時間制約を始めた様々な制約が発生する。従って、アプリケーションのみならず、連続メディア処理のプラットフォームとなるオペレーティングシステムやミドルウェアに対しては、コンテンツや処理環境への適応性が求められる。以上の背景から、本研究では、ネットワーク、オペレーティングシステム、ミドルウェアにおける各種適応性の要件について検討を行い、マルチメディア処理を始めた様々な処理やサービスのための適応的なソフトウェア基盤の確立を目指している。

2. リアルタイムシステムの研究

従来のオペレーティングシステムでは、すべての処理を公平に行うために、マルチメディアアプリ

ケーションが要求する実時間制約を満たすことが困難となっている。特に、システムの多くは、ネットワークで相互接続された分散環境上に構築されており、任意のハードウェア、任意のアプリケーション、任意のデータにより構成される開放型システムである。本研究では、このような開放型システムであるマルチメディア処理環境において、連続メディア処理の時間制約を満たしつつ、非連続メディアを処理するイベント駆動型処理の高い応答性を達成するために、リアルタイムオペレーティングシステムEaselを開発している。Easelの主な特徴を以下に示す。

(1)スケジューリング

優先度を考慮したEarliest Deadline Firstスケジューラを実現している。これに加えて、イベント駆動型のスレッドに対しては、Sporadic Serverを始めとしたサーバ型のスケジューラの導入を予定している。これらにより、各スレッドに優先度とデッドラインを設定し、それらのパラメータに従ってスケジューリングが可能となる。

(2)プロセス・スレッド管理

スレッドに対して実時間制約を持つリアルタイムスレッドと、実時間制約を持たない非リアルタイムスレッドを混在して扱える管理方式を導入している。さらに、このような特性や実時間制約を動的に変更可能とする方式も実現している。

(3) 割込み処理

割込みハンドラによる割込み処理に加えて、カーネルスレッドによる割込み処理を実行する機構を持つ。カーネルスレッドによる割込み処理を行う場合、任意の時点でブロック可能であり、より優先される処理の実行遅延を低減することができる。

(4) リアルタイムJavaVM

JavaVMにリアルタイム拡張を行ったリアルタイムJavaVMを構築している。本VMは、Easel上に実装しており、クラスロード、ガーベージコレクション、スレッド管理に独自の方式を採用している。

以上がEaselの特徴であるが、現在行っている研究を以下に上げる。

(1) デバイス入出力機構

マルチメディアアプリケーションが要求する連続メディアデータを定期的かつ安定して提供する機構の構築を行う。

(2) ストリーミングサーバに適した通信機構

ストリーミングサーバにおいて、冗長なメモリコピーやコンテキストスイッチによるオーバーヘッドを削減する。さらに、フロー毎の帯域制御を行うことにより、安定したデータ転送を提供する。

(3) 仮想記憶管理手法

マルチメディアアプリケーションのメモリ参照特性を考慮したページ管理を行う。これにより、マルチメディアアプリケーションのページフォルトを削減し、円滑な処理を可能とする。

(4) リアルタイムクラスライブラリ

ガーベージコレクションのスレッドの操作や、ガーベージコレクションの対象外のメモリ管理領域にオブジェクトを生成するためのクラスを提供する。これにより、リアルタイムスレッドの実行中のガーベージコレクションを回避することが可能となる。

(5) リアルタイムファイルシステム

マルチメディアアプリケーションの特徴を考慮

したディスクの配置手法、さらに、リアルタイムパラメータを指定可能とすることによって、アプリケーションに対して安定したデータ供給を可能とする。

3. 分散オペレーティングシステムの研究

本研究では、環境の変化に適応可能とすることを目的としたオペレーティングシステムについて検討している。そのため、計算機資源の抽象化によるシステム構成法と、リフレクションを使用したシステム構成法の2つを中心として検討を進めている。

3.1 計算機資源の抽象化

近年、ネットワーク技術の普及により、多数の計算機がネットワークに接続され利用されている。それらの計算機は、それぞれ異なる処理能力を持ち、また利用できる周辺デバイスも異なっており、アプリケーションが一様な環境において処理を行うことが困難となっている。さらに、モバイル環境など、環境自体が動的に変化する場合もある。以上の背景から、本研究では、異なる環境、さらには動的に変化する環境に適応可能な分散オペレーティングシステムの構成法について研究している。具体的には、アプリケーションに対して一様な処理環境を提供するために、メモリ、プロセッサ、デバイスなどの各種システム資源を抽象化し、これらを位置透過に使用できる環境を実現する機構について研究している。さらに、この機構に基づいたオペレーティングシステムSolelcを開発している。Solelcにおいては、各システム資源は次のように抽象化される。

- メモリは、すべての計算機間で1つの仮想アドレス空間を共有する分散仮想記憶として抽象化される。分散仮想記憶内では、すべての計算機でアドレスが一意的な意味を持つ。
- プロセッサは、レジスタセットなどをデータとして持つスレッドとして抽象化される。スレッドは、任意の計算機に生成可能であり、ネットワーク透過に動作する。

- デバイスの抽象化は、デバイスの存在する計算機上で動作するデバイスドライバによって行われる。各デバイスドライバは、任意の計算機からその位置を意識することなく利用可能である。
- システムコールや保護違反などの割込みはイベントとして抽象化される。イベントは、任意の計算機で取得可能であり、また取得したイベントに対する処理も任意の計算機で行うことができる。

このような資源を抽象化する機構によって、オペレーティングシステムおよびその上のアプリケーションをネットワーク透過に動作させ、環境の変化に適応させることが可能となる。本研究では、このような分散された資源を抽象化する機構を構築し、またその上で動作するオペレーティングシステムの構成法を明らかにし、実際にこれに基づいたオペレーティングシステムの開発を行うことを目的としている。

3.2 リフレクティブアーキテクチャ

分散環境では、計算機やデバイスの追加・削除などによるシステム構成の変化と、計算機負荷の変化やデバイスの競合などにより、実行環境が変化する。これらの変化が発生する時刻を予測することは困難であり、また、変化によって発生する影響は、大域的なものではなく、一部のシステム構成や特定の資源に関するものなど、局所的であることが多い。このような環境の変化にオペレーティングシステム全体が適応するのではなく、影響を受ける部分のみが適応することで、急激な変化にも対応することが可能になると考えられる。

本研究で開発しているエージェント指向オペレーティングシステムAGでは、このような環境に柔軟に対応するために、リフレクション(reflection)を用いている。リフレクションとは、プログラムコードやその解釈・実行機構をプログラム自身から参照・変更することである。リフレクションを用いることで、エージェントは自らの動作を変更し、その環

境に適した処理を行うことが可能となる。

本研究では、適応性を実現するために必要となるリフレクティブエージェントを実現するための手法について検討している。この手法では、即応型スレッドと熟考型スレッドと呼ばれる2種類のスレッドを用いてエージェントを構成する。即応型スレッドは、外部からの要求を処理する。熟考型スレッドは、環境の変化に応じて、即応型スレッドの動作を制御する。このようにエージェントを構成することで、容易にリフレクティブエージェントの実現と拡張を行うことができる。

4. コンテキストの変動に適応可能なソフトウェア基盤の研究

移動型ネットワークとそれを利用するユビキタスコンピューティングが注目されている。この新しいコンピューティング環境におけるシステムの特徴は、その構成要素(モバイルPC、PDA、デジタルデバイス)が移動可能な点にある。従来のシステムでは、その構成要素が固定されており、単一のコンテキスト(例えば、オフィスにおけるデスクトップなど)のみを想定している。一方、新たなシステムでは、その構成要素が移動可能であるため、そのコンテキストは移動とともに、また時間とともに変動する。

このような移動や時間とともに変動するコンテキストでは、利用可能なアプリケーション、ミドルウェアサービス、システムリソースもまた変動する。したがって、動的なマルチコンテキストでは、コンテキストの変動を自ら発見・検知し、それに適応する新たなソフトウェアが必要となる。このような新たなクラスのソフトウェアは、コンテキストウェアなソフトウェアと呼ばれている。

本研究では、移動するユーザに対して様々なサービスを提供するために、次のようなコンテキストウェアなソフトウェア基盤のシステムアーキテクチャとそのメカニズムについて検討している。

- ユーザの移動に従い変動するコンピューティ

ング環境のリソース(例えば、利用可能なデバイス/サービス/コンテンツ、ネットワーク帯域など)の制約を知覚/発見する。

- ユーザの移動に従い変動するユーザ要求(タスク)を認識する。
- 上記の2つの相互作用から、利用可能なリソースを部品としてユーザタスクを即興的に合成する。

上記のようなユーザタスクに応じたサービスの合成は、パーベイシブコンピューティングにおいて研究が進められている。しかし、現状の研究は、限定された空間(例えば、部屋や家など)の限定されたユーザのタスクを、十分に知り得ているコンピューティングリソースから合成するに留まっている。一方、モバイルコンピューティングでは、広域に移動するユーザへサービスを提供/配信する研究が進められている。しかし、現状の研究は、ユーザの位置に関わらず継続的なサービスを提供することが主要目標であり、利用可能なリソースを探索し、サービスを合成することは考えられていない。

従って、本研究では、ユーザが移動する広域なユビキタス空間を複数/異種のパーベイシブコンピューティング空間により構成される空間として考え、このような広域な空間の不特定なリソースからユーザのタスクを合成し、さらにユーザタスクの処理の継続性を実現するコンテキストウェアシステムのソフトウェア基盤を構築する。具体的な研究課題としては、以下のものが挙げられる。

(1)セマンティックモデル

サービスの合成を可能とするために、移動空間におけるコンテキストを構成するユーザ要求(タスク)やコンピューティングリソース(デバイス/ソフトウェア部品/コンテンツなど)の特性を記述する方式を明らかにする。

(2)サービスの知覚/合成のための基盤

ユビキタス空間を、以下の3つの環境がそれぞれ異なる技術で実現され、かつ混在する環境

としてとらえる。

- 固定サーバが存在する従来型の分散環境
- 複数の固定サーバ間を移動可能な有線/無線ネットワークにより構成されるノーマディック分散環境
- 無線ネットワークにより構成され、固定的なサーバが存在しないモバイル分散環境

このようなユビキタス空間を移動するユーザ(あるいはデバイス)のために、ユーザタスクの認識とコンピューティングリソースの発見/知覚を可能とするコンテキストウェアシステムのアーキテクチャを構築する。また、認識したユーザタスクと発見/知覚されたコンピューティングリソースとの相関からサービスを合成/更新するコンテキスト適応メカニズムを開発する。

(3)セキュリティとプライバシーの両立

セキュアな社会と個人のプライバシーとは相互に相容れない関係にある。ユビキタス社会ではこれらを両立させることが重要となる。この両立を実現するために、電子的なメカニズムだけでなく、コンテキストにおける物理的な制約事項を利用する。

(4)コンテキスト適応システムの試作と実証実験

(1)、(2)、(3)を用いて、複数の小規模なユビキタス空間から構成される環境において、複数のコンテキスト適応システムを試作し、任意のユビキタス空間を出入りするユーザによりシステム全体の有効性を検証する。

5. おわりに

本稿では、マルチメディア処理におけるリアルタイム性を保証するリアルタイムオペレーティングシステムの研究、処理環境の変化に適応可能とするための資源の抽象化とリフレクティブアーキテクチャを基本としたオペレーティングシステムの研究、コンテキスト変動に適応可能なソフトウェア基盤の研究の概要について述べた。今後は、これらの3つのサブテーマにおける研究を進め、最終的にマルチメディア処理のための適応的なソフトウェア基盤を構築していく予定である。