

# Keio-MMP における

5R-2

## X ウィンドウシステムの実時間拡張方式†

多田 征司\*†

斎藤 信男\*\*

\*横河・ヒューレット・パッカード (株)

\*\*慶應義塾大学環境情報学部

### 1 はじめに

慶應義塾大学では、情報処理振興事業協会 (IPA) からの委託研究として10社の企業と共同研究として、マルチメディア統合環境基盤ソフトウェアプロジェクト (Keio-MMP プロジェクト) を行なっている [1]。グラフィカルユーザインターフェース (GUI) がコンピュータの利用環境としてその重要性を増している今日、ウィンドウシステムはマルチメディア基盤環境を構築するのに不可欠な要素であると考えられる。X ウィンドウシステムは、移植性と拡張性に優れていることから広く普及しているが、動画データのような連続メディアを扱う際には、描画処理のタイミングがデータの時間制約を満たす必要があり、標準の X ウィンドウシステムでは問題がでてくる。本稿では、Keio-MMP における X ウィンドウシステムの実時間拡張方式について述べる。

### 2 Keio-MMP ソフトウェアアーキテクチャ

著者らが提案している Keio-MMP ソフトウェアアーキテクチャでは、実時間マイクロカーネル上で互いに協調して動作するサーバ群によって分散マルチメディア統合環境を構築する [2]。Keio-MMP アーキテクチャでは、メディアの処理を行うサーバ群をパフォーマンス、また、それらを管理するサーバをコンダクタと呼ぶ。コンダクタは、メディアの時間制約にしたがって、パフォーマンスに処理を依頼する指揮者としての役割を担うほか、パフォーマンス間のプロセス間通信におけるオーバーヘッドを軽減するために、通信に必要な共有バッファを管理する。図1にアーキテクチャの概略を示す。本稿で述べる拡張 X サーバは、動画再生時の表示系パフォーマンスとしての役割を担う。

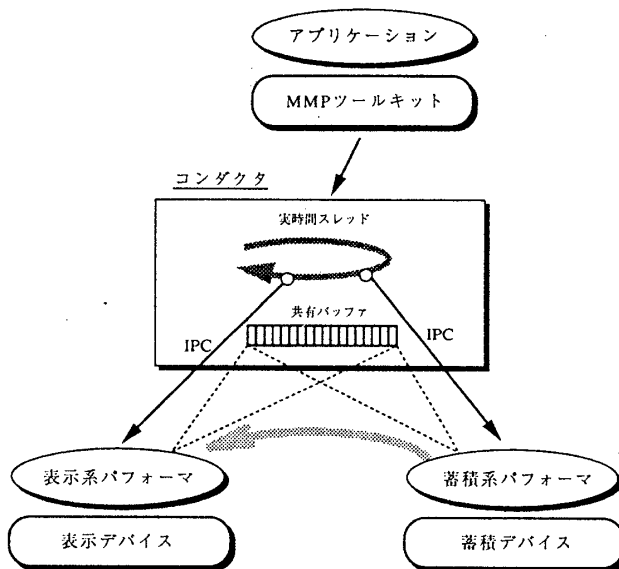


図1: Keio-MMP アーキテクチャ

### 3 X サーバの実時間拡張方式

#### 3.1 X サーバの問題点

標準の X サーバでは、複数のクライアントに対するサービス切り替えにおいて単純なラウンドロビンスケジューリング方式を採用している。実時間システムでは、あるクライアントのリクエストが他のクライアントのリクエストより高いプライオリティを持っていたり、そのプライオリティが動的に変化することがあり、従来の X サーバではそれに対処することができない。そこで、

- プライオリティ
- プリエンプタビリティ

をスケジューラに導入する必要がある。

これまでに、X サーバのスケジューラにプライオリティキューを設ける方式 [4] が報告されている。また、プライオリティベースではないが、X サーバをマルチスレッド化することによってサーバのレイテンシを向上させた報告 [5] もある。しかしながら、これらの方式では、移植性が犠牲になる、X プロトコルを変更する必要がある、将来の拡張に対応することが困難になる、などの問題がある。

#### 3.2 拡張方式の概要

そこで著者らは、X ウィンドウシステムを拡張し、動画表示に焦点を絞った簡易実時間マルチスレッド化を試みた。本方式では、動画表示のように時間制約を伴う描画処理を行う部分を X サーバから分離して実時間マルチスレッドサーバ (以後、動画サーバ) として実装し、従来の X サーバと共存させる。動画サーバの実装には、実時間マイクロカーネル Real-Time Mach3.0 [3] (以下、RT-Mach3.0) が提供している実時間スレッドを用いている。X サーバは従来通りウィンドウ制御等を行い、ウィンドウ枠内の動画表示の実際を動画サーバが担当する。図2に拡張方式の概要を示す。

動画サーバは X サーバとは独立したリクエストキューを持っており、前述の Keio-MMP アーキテクチャに準拠したインターフェースを提供する。すなわち、動画サーバは、コンダクタからの処理依頼にしたがって、共有バッファ上に置かれた動画データを表示デバイスに直接描画

“Realtime Extension of X Window System on the Integrated Multimedia Environment Keio-MMP”†  
Seiji Tada\*, Nobuo Saito\*\*

\*Hewlett-Packard Company, Asia Pacific Product Operation, 1-3-2, Murotani, Nishi-Ku, Kobe-shi, Hyogo, 651-22 Japan, \*\*Keio University, 5322, Endo, Fujisawa-shi, Kanagawa, 252 Japan

†この研究は、情報処理振興事業協会 (IPA) が実施している開放型基盤ソフトウェア研究開発評価事業「マルチメディア統合環境基盤ソフトウェア」プロジェクトのもとに行なわれた。

‡開放型基盤ソフトウェア湘南藤沢キャンパス研究室の研究員としてIPAに登録されている

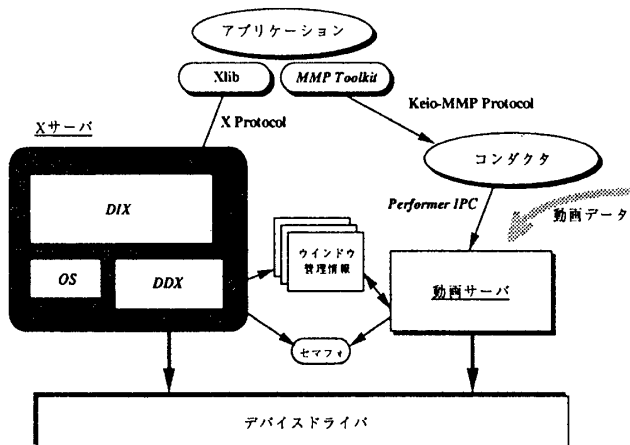


図 2: 拡張方式概要

する。その他の描画処理については、X サーバが行う。

### 3.3 X サーバと動画サーバの共存方式

X サーバと動画サーバとは互いに独立して描画処理を行うため、描画処理における一貫性をサーバ間で保つ必要がある。このため本方式では、ウィンドウ管理情報とカラーマップとを両サーバ間で共有させている。ウィンドウ管理情報は、ウィンドウの位置情報とクリッピング情報がウィンドウ毎に格納され、それらが双方向リストで連結された構造になっている。ウィンドウ管理情報は動画サーバが描画を行うごとにアクセスされるため、実記憶空間に固定しておきページングによる遅延を抑制するようにした。このため、管理情報が不必要に肥大化しないように配慮する必要がある。そこで、拡張 X プロトコルを導入し、アプリケーションもしくはコンダクタがウィンドウの共有状態を制御 (On/Off) できるようにした。拡張プロトコルを表 1 に示す。アプリケーションは `XmmpUnregisterWindow()` を用いてウィンドウの共有状態を解くことにより、ウィンドウ管理情報から当該ウィンドウの情報を削除することができる。

```
XmmpRegisterWindow(dpy, winID)
XmmpUnregisterWindow(dpy, winID)
```

表 1: プロトコル拡張

本方式のように複数の処理主体が協調して動作する実時間システムでは、処理の一貫性制御<sup>1</sup>において、

- それらを効率良く実現すること
- 優先度逆転による予測不能な遅延を抑制することが課題となる。

本方式においては X サーバと動画サーバとを異なるアドレススペースで実現することも可能であるが、アドレススペース間での排他制御は負荷が大きくなる。そこで、両サーバを同じアドレススペースで走行させることにより排他制御の効率化をはかった。また、排他同期制御に伴う優先度逆転問題については、RT-Mach3.0 が提供し

<sup>1</sup>ウィンドウ管理情報/カラーマップの更新時の排他・同期制御やグラフィックスデバイスへのアクセス時の排他制御など

ている実時間同期機構 [6] を用いることにより対処している。

### 3.4 本方式の利点

- ここで提案した拡張方式には、以下のような利点がある。
- X サーバの拡張部分を DDX 層にまとめることができ、移植性や拡張性を犠牲にすることが少ない。
  - 動画表示を行う部分を別モジュールとしているため、動画表示のための圧縮伸長ハードウェアに対して、ウィンドウシステムに大きな変更を加えることなく対応できる。
  - 動画サーバに対するリクエストに Keio-MMP アーキテクチャに準拠したインターフェースを採用することにより、
    - 連続メディアを扱うサーバ群に統一的なインターフェースを与えることが可能になる。
    - 共有バッファに基づくパフォーマンス間通信機構を利用することができ、タスク間通信におけるオーバーヘッドを低減することができる。

## 4 おわりに

本稿では、Keio-MMP プロジェクトにおける連続メディアのための X サーバの実時間拡張方式について述べた。本拡張方式では動画表示のための実時間化に焦点を絞ったが、マルチメディアアプリケーションのための基盤ソフトウェアにおいては、対話的処理におけるシステムの応答性もまた重要な課題である。このためには、入力デバイスの割り込み処理や対話的ウィンドウの描画処理を遅滞なく行うことが必要になってくる。また、ウィンドウマネージャやツールキットとの関係におけるプライオリティ管理も重要になってくる。これらについては今後の課題としたい。

## 謝辞

本研究を行なうにあたり協力/助言していただいた慶応義塾大学「マルチメディア統合環境基盤ソフトウェア」プロジェクトの皆様に感謝いたします。

## 参考文献

- [1] 徳田, 萩野, 斎藤: “分散マルチメディア統合環境 Keio-MMP プロジェクトにおける連続メディア処理のためのソフトウェアアーキテクチャ,” 第 49 回情報大全 7R-01 (1994).
- [2] 西尾, 多田, 徳田, 萩野, 斎藤: “Keio-MMP における Conductor/Performer アーキテクチャの協調性能評価,” 第 49 回情報大全 7R-07 (1994).
- [3] H. Tokuda, T. Nakajima and P. Rao: “Real-Time Mach: Towards a Predictable Real-Time System,” USENIX Mach Workshop, pp.73-82 (1990).
- [4] R. Chesler: “Experiences In Extending X To Provide Determinism,” The X Resource, Issue 7, pp.79-86 (1993).
- [5] J. A. Smith: “The Multi-Threaded X Server,” Proceedings of the 6th Annual X Technical Conference, pp.73-89 (1992).
- [6] H. Tokuda and T. Nakajima: “Evaluation of Real-Time Synchronization in Real-Time Mach,” Proceedings of the Second USENIX Mach Symposium, October (1991).