

多様な接続形態を実現するスタック可能なデバイスアクセス機構

姜 亨明 小林 良岳 紅谷 順 中山 健 前川 守

電気通信大学 大学院情報システム学研究科

1 はじめに

ネットワーク上の様々なデバイスを有効利用する分散ミドルウェアとしては、Jini[3] や Universal Plug Play[4] 等があるが、これらは独自のプロトコルに基づいており、混在させて相互運用はできない。このような分散ミドルウェアは今後さらに重要になってくるが、これらが1つのプロトコルとして統合される見通しはない。

このような状況では、現実的な利便性の観点から、異なったプロトコルのネットワークデバイスを、一つのアクセス手法を用いて透過的に利用するための手段が必要となる。

そこで本研究では、動的再構成可能な OS *Aya*[1] の特徴を生かし、ネットワーク上の多様なプロトコルのデバイスに対する透過的なアクセスを実現するための *Seki* I/O マネージャを提案する。

Seki は、I/O サブシステム部、デバイスドライバ部において、プロトコル変換機構の積み重ね(スタック)が可能なデバイスドライバと、サービスを受け付けるためのチケットマスタで構成される。

2 タスクにおけるデバイス空間

コンピュータにつながれているデバイスには様々な種類とアクセス方式が存在する。これらのデバイスを有効に活用するためには、OS 上で稼働しているタスクが、そのデバイスがどこにあるか、また使用する権限が与えられているかを考慮する必要がある。さらに、同じ特徴を持つデバイスがあるときには、その中のどのデバイスを使用するか考えねばならない。

透過的なデバイスアクセスを実現するためには、ローカルとネットワークのデバイスが同じデバイスファイルとしてアクセスできることが望ましい。しかし、複数のタスクが同種のデバイスを利用する場合、既存の UNIX システムのように、1つのデバイスファイルが単一のデバイスを指すシステム設計だと、ローカルとネットワークにまたがる同種のデバイスをタスク毎に振り分けることが容易でない。

そこで *Aya* では、カーネル内にある *Tsuzuri* ファイルシステム [2] によってタスク毎にデバイス空間を分離している (図 1)。

Tsuzuri は vnode/VFS ファイルシステム上に実装されているファイルシステムの一部として、タスク毎の固有デバイス情報を管理している。*Tsuzuri* のデバイス空間は、システムグローバルなデバイスファイルであるグローバル部と、タスク固有のデバイスファイルであるプライベート部に分離されている。プログラムからのデバイス

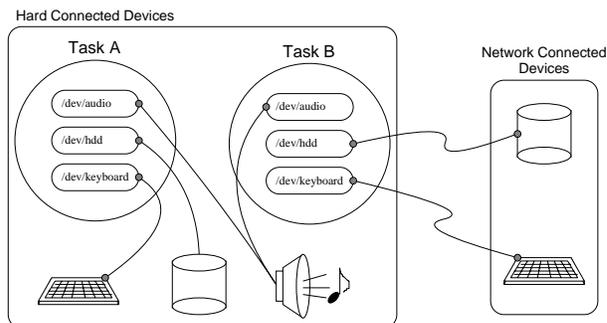


図 1: 個々のタスク毎のデバイス空間

アクセスは、グローバル部に対して行い、ファイルシステム内部でプライベート部へのアクセスへと変換される。

3 *Seki* I/O マネージャ

本研究の課題である *Seki* I/O マネージャでは、ローカル・タスクからのデバイス要求に対してローカル・デバイスの割り当てが不可能な場合、ネットワーク上のデバイスを参照し、可能であればネットワーク・デバイスの割り当てを行なう。この機構を実現するためには、ネットワーク・デバイスの着脱通知と機能通知を取得し、それらの情報をなんらかの手段を用いて一元管理し、その情報を元にタスク毎に適切なプロトコルへの変換を行う必要がある。

そこで *Seki* は、適切に機能を分離するため、次の2つの部分で構成した (図 2)。

- ・ チケットマスタ
- ・ *SDD*(Stackable Device Driver)

3.1 チケットマスタ

複数のプロトコルを透過的に利用する場合、利用可能ネットワークデバイスを参照するための機構が必要となる。しかし、様々なプロトコルのデバイス郡を割り当てのたびに何度も参照しに行くことは非効率である。

そこで *Seki* では、ローカル、ネットワークに関する複数のプロトコルに対応し、タスクに対するデバイス割り当てを統一的行うためにチケットを用いる。このチケットを発行する機構をチケットマスタと呼ぶ。

チケットマスタは割り当て対象となるデバイスに関する情報をチケットとして保持する。個々のチケットは以下の情報を含む。

- ・ デバイスの位置
- ・ サポートするプロトコル
- ・ アクセス制御情報

チケットにはローカルチケットとリモートチケット、変換チケットがある。ローカルチケットはチケットマスタのインターフェースを統一するために、ローカルデバイスの割り当てに関する情報を提供するチケットである。これに対し、リモートチケットは *Aya* がネットワーク上の

A Stackable Device Driver for Heterogeneous Network Environment.

Kang Hyong Myong, Yoshitake Kobayashi, Jun Beniya, Ken Nakayama, Mamoru Maekawa
Graduate School of Information Systems, University of Electro-Communications

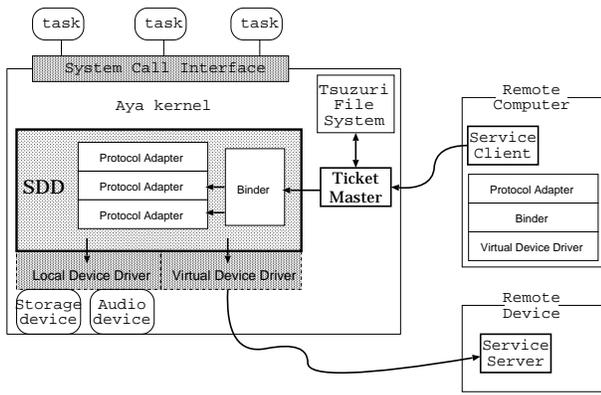


図 2: Seki の構成

デバイスを利用するための情報を提供するチケットである。変換チケットは、Aya のチケットマスタが対応している様々なプロトコルから、ローカルデバイスへのアクセス要求があった時に、その要求に応えるためのものである。

ネットワーク上に存在するデバイスは、その特徴に応じたロケータが存在する可能性がある。Jini[3] などに対応するコンピュータのデバイスに対して Aya では、それらロケータに個々に対応し、ロケータから割り当て可能なデバイスの一覧を得る。そして、どのプロトコルに基づいてデバイス一覧の取得が行われたかを確認し、リモートチケット、変換チケットを生成する。もし、その情報取得の際に実際に割り当てが行われてしまうプロトコルである場合は、そのデバイスを即座に解放し、情報のみを保持する。

3.2 SDD(Stackable Device Driver)

Seki はデバイスドライバ・インターフェースを用いてローカル、ネットワークのすべてのデバイスにアクセスする。つまり、I/O サブシステム部、デバイスドライバ部にプロトコル変換機構とネットワーク機能が備わっている必要がある。そこで Seki では、デバイスドライバ上にプロトコル変換機構であるプロトコルアダプタをスタックするような形を取った。これを SDD(Stackable Device Driver) と呼ぶ。

SDD は次の 3 つの部分から構成される。

- ・ デバイスドライバ
- ・ プロトコルアダプタ
- ・ バインダ

デバイスドライバ

デバイスドライバはローカルデバイスドライバと仮想デバイスドライバの 2 つに分離される。ローカルデバイスドライバが直接デバイスにアクセスするのに対し、仮想デバイスドライバはアクセスを抽象化し、ネットワーク上のデバイスにアクセスするためのものである。

仮想デバイスドライバはネットワーク上の情報とやりとりを行うため、ネットワークインターフェースとの連携をとる。また複数のプロトコルに対応するためにはチケットのポート管理を行なう必要がある。

プロトコルアダプタ

プロトコルアダプタは、プロトコルごとにあり、Aya のデバイスドライバが使用可能なアクセス方式に変換する。デバイスへのアクセス方針は統一したインターフェースを持っているため、1 つのプロトコルを処理するアダプタは、いくつものデバイスに対してバインドすることが可能である。最も簡単なプロトコルアダプタは、ローカルデバイスへの直接アクセスを行なうものである。

ネットワーク上に新しいプロトコルをサポートするデバイスが現れ、それを利用したい時、Seki では Aya の動的再構成の機能を用いてプロトコルアダプタの動的な追加や更新を行なう。これによりコンピュータを再起動することなく新しいプロトコルを利用することが可能である。

バインダ

SDD におけるバインダの役割は、チケットマスタからのチケットを受け取った後、プロトコルアダプタの選出を行ない、プロトコルアダプタとデバイスドライバの間連結を行うことである。

ここで、ローカルデバイスの割り当てが不可能な場合、バインダは仮想デバイスドライバを通じてネットワーク上のデバイスに割り当てを試みる。そのデバイスが対応するプロトコルに応じて、プロトコルアダプタモジュールの差し替えを試みる。

4 関連研究

ホームネットワークの研究で実用化されているものとしては、前述の Jini や UPnP の他にも、HAVi[5] などがある。しかし、これらの手法を用いる場合は、各コンピュータが実際にそのサービスを受けられる状態を常に保てなければならず、また複数のプロトコルを同時に処理しきれないので、柔軟性に欠ける。

5 まとめ

本稿では、動的再構成可能なオペレーティングシステム Aya において、異なったプロトコルをサポートするネットワークデバイスを、透過的なアクセス手法を用いて利用するための機構である I/O マネージャ Seki について述べた。デバイス割り当てに関する機構を SDD、チケットマスタに機能分離することにより、動的再構成を用いた簡潔なプロトコル変換を行なうことが可能となり、効率的なデバイス管理を行なえる可能性を示した。

参考文献

- [1] 小林良岳, 佐藤友隆, 唐野雅樹, 結城理憲, 前川守: 彩: コンパイル時に自動生成される Portal を基に動的再構成可能なオペレーティングシステム, 電子情報通信学会論文誌, Vol. J84-D-I No.6, pp.605 616, 2001.
- [2] 結城理憲, 小林良岳, 唐野雅樹, 紅谷順, 姜亨明, 前川守: タスクの再構成を支援するためのファイルシステム, 第 64 回全国大会予稿集, 情報処理学会, March, 2002.
- [3] Ken Arnold, Bryan O'Sullivan, Robert W. Scheifler, Jim Waldo, Ann Wollrath: The Jini Specification, Addison-Wesley, 1999.
- [4] Microsoft Corporation: Universal Plug and Play Device Architecture, http://www.upnp.org/download/UPnPDA10_20000613.htm, 2000.
- [5] HAVi Organization: HAVi, the A/V digital network revolution. White Paper, <http://www.havi.org> 1999.