

仮想化 PC 環境を用いた加速器制御オペレータコンソール THE OPERATOR CONSOLE FOR ACCELERATOR CONTROL SYSTEMS ON A VIRTUAL MACHINE

中尾圭佐*, 早川建, 境武志

Keisuke Nakao*, Ken Hayakawa, Takeshi Sakai

Laboratory for Electron Beam Research Application (LEBRA), Nihon University

Abstract

By progress of computer technology, the performance of PC is growing and the price is low. PC is used for control of a small-scale accelerator facility from the merit of availability and low price. However, the reliability of PC is lower than the computer for factories, such as VME, cPCI, and TCA. It seems that using virtual PC has an advantage in maintenance of an accelerator control system. This paper describes the advantage and problem of virtual PC for accelerator control.

1. はじめに

日本大学電子線利用研究施設 (LEBRA) の 125MeV linac は PC ベースの制御システムを採用している。加速器のオペレーターコンソール及び、偏向電磁石、FEL 共振器ミラー等、多くのコンポーネントが PC を介して制御されている。市販の PC は、価格は安い が信頼性が低く、数年に 1 度は故障し、時には利用実験に支障をきたすこともある。更新作業で最も時間がかかるのが、ハードウェアの構成が変わることによる環境の再構築作業であるが、PC のチップセットや CPU、周辺機器類の進化は日進月歩で、故障時に更新時と同じ構成のパーツを購入することは難しい。近年、CPU のマルチコア化、メモリの高速化、低価格化により、PC 上で PC をエミュレーションする仮想化技術が一般的になっている。このような仮想化された PC は、標準的なネットワークカードとストレージで構成されている。また、仮想化されたストレージの実態はファイルなので、環境構築時のストレージファイルをバックアップしておけば、短時間でシステムを復旧させることができる。本発表では、仮想環境下における加速器およびコンポーネントのオペレーターコンソールを仮想化することによる利点、問題点を議論する。

2. LEBRA 加速器制御システムの問題点

LEBRA の加速器制御プログラムは、WindowsXP で動作している Visual Basic 6 (以下 VB6) で開発されたプログラムである。Visual Basic version 6 (以下 VB6) は、1998 年に発売されたが、Microsoft .NET フレームワークに VB が統合された際に言語仕様が大きく変更され、VB と称しているが全く別のプログラミング言語となってしまった。既に VB6 の開発環境のサポートは 2005 年に終わっており、VB6 で書かれたソースコードを修正できる開発環境で正式なサポートがあるものは存在しない。そのため制御プログラムの更新、改良は 1998 年に発売された開発環境を使用して行われている。マイクロソフト社は、VB6 で開発されたプログラムの実行に必要なライブラリを提供しており、最新の Windows8 でのプログラムの実行はサポートしている。しかし 5 年後には Windows8 のサポートは終わり、Windows9 でサ

ポートされるかは未定である。

一方、制御プログラムが動作しているコンピュータのハードウェアは、民生用の PC を用いている。民生用の PC は価格が安く入手性が良いため、長く使われてきた。しかし信頼性が低いため数年に一度は故障し、復旧に数日の運転停止を伴うこともある。また、加速器制御プログラムだけでなく、偏向電磁石電源、電子銃、RF 位相等も別の民生用 PC で動作しているため年に 1 台は、どれかが故障している。現在は、古い環境をできるだけ維持しながら PC を更新しているが、民生用 PC の進化は非常に速く、例えば 2 年前と同じ構成の PC を調達することはほぼ不可能である。よって入手できるハードウェアを購入して環境を構築するのだが、ハードウェアベンダが古い OS のサポートをやめてしまうこともあり、実行環境を構築するのに大きな労力と時間を要している。

このように、加速器制御プログラムといった加速器の運転に重要なプログラムが VB6 で書かれており、比較的古い実行環境を準備しなければならない、年々制御システムの維持が困難になっている。制御システムの大規模な更新が必要だが、LEBRA のような小規模な加速器施設では、人的資源に限られており難しい。そこで、近年実用に耐えるようになった仮想化技術 (Virtualization Technology) を、ハードウェアの詳細を隠蔽する、ハードウェア・ラッパー (Hardware Wrapper) として捉え、仮想環境下で古い実行環境を動作させ、プログラム実行環境の維持していくことを考えている。

3. コンピューターの仮想化

コンピュータの仮想化とは、「ハードウェアとソフトウェアの結びつきを抽象化し、OS などのソフトウェアがハードウェアを直接制御するのではなく、抽象レイヤーを介して動作できるようにすること」である [1]。Figure 1 に、仮想化されたコンピュータの構造を示す。仮想化されていない PC と同様に、物理的なハードウェア上で OS が動作している。この OS をホスト OS と呼ぶ。そのホスト OS 上で、仮想化抽象レイヤーにあたる仮想マシン (Virtual Machine) を動作させる。仮想マシンは、PC のハードウェアをエミュレートするので、仮想マシンで動く OS (以下ゲスト OS) は、仮想マシンで仮想化されたハードウェアのみを認識する。すなわち、ゲスト OS からは仮想マシンで仮想化されたハードウェアしか見え

* nakao@lebra.nihon-u.ac.jp

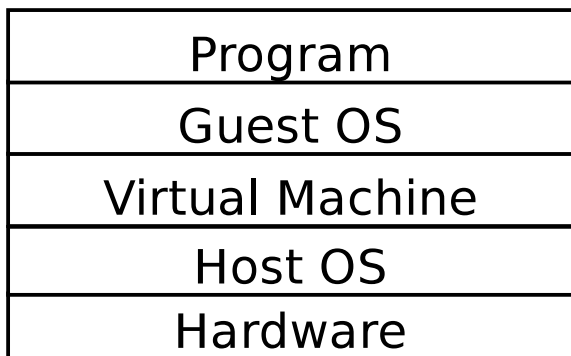


Figure 1: Structure of Virtual PC

ず、物理的なハードウェアを隠蔽していることになる。

一般に OS は、データや機能を保護するために、リングプロテクションという複数の特権レベルを持っている。各特権レベルに応じて機能が制限されており、もっとも信頼性の高いリング 0 では、ハードウェアやメモリに直接アクセスすることができる。

一般に、OS はスーパーバイザーモードとユーザーモードの 2 つのモードを使用している。リング 0 に相当するスーパーバイザーモードは OS のカーネル部分が配置され、ユーザープログラムはより低い特権レベルで実行される。低い特権レベルから高い特権レベルのハードウェアにアクセスするには、決められた方法でアクセスする必要があり、任意のアクセスを許していない。ユーザープログラムに欠陥があり、暴走したとしてもシステム全体を破壊するようなリソースにアクセスを許可しないことでシステムを保護している。

ゲスト OS は、一般にスーパーバイザーモードで動作する前提で作成されている。仮想マシンは特権モードでは動作しないために、ゲスト OS がハードウェアリソースに直接アクセスしようとする時と特権違反が発生する。仮想マシンはこの特権違反をフックにしてホスト OS に中継する。

仮想化されたデバイスは、既に多くの OS でポートされているものをエミュレート物理ハードウェアがどう変わろうとも、仮想マシンが同じであればゲスト OS から見たハードウェアに変化はない。また仮想ストレージは、ホスト OS からは一つのファイルなので、そのファイルのバックアップをとっておけばバックアップファイルのコピー程度で、容易にディスクを復旧できる。

このように、仮想マシンはゲスト OS からのリソースへのアクセスをホスト OS に中継することでハードウェアを抽象化している。もしも仮想マシンがゲスト OS からリソースアクセス要求を受け取れば特権違反をフックする必要はない。ゲスト OS を修正して、直接仮想マシンにリソースアクセス要求することで実現する仮想化を、準仮想化という。完全にハードウェアをエミュレートする完全仮想化は、ゲスト OS を修正する必要はないが特権違反をフックするため、準仮想化に比べてオーバーヘッドが大きい。準仮想化の実装には、Linux で動く Xen がある。完全仮想化の実装には、VMWare、Virtual PC、Virtual Box、QEMU 等がある。

4. 加速器制御への応用

加速器制御の標準的なモデルでは、加速器を構成するコンポーネントを直接制御プログラムが制御するのではなく PLC といったローカルコントローラを経由して制御している。オペレーターコンソールプログラムや、フィードバックといったアプリケーションは上位のコンピュータで実行され、ローカルコントローラにコマンドを送ることで加速器を制御している。

仮想マシンが仮想化しているペリフェラルデバイスには、Ethernet、USB、シリアルがありローカルコントローラとの通信は問題なく行えると思われる。ただし、USB は、USB 機器のデバイスドライバがホスト OS にインストールされていなければならない。問題は、PCI バス等に挿入されている A/D コンバーターや GPIB インターフェースといった拡張カードの扱いである。拡張カードはホスト OS にデバイスドライバがインストールされていなければならないだけでなく、仮想マシンでその拡張カードが仮想化されていなければならない。オープンソースの仮想環境を用いれば自らブリッジを書くことはできるが、現実的ではない。よってこれらの機能を外部化する必要がある。LEBRA では、制御用 PC および FEL 共振器ミラー制御用 PC にデジタル IO ボードおよびモータードライバーボード、GPIB インターフェースが、PCI バスに挿入されている。これらを CPU モジュールを持った cPCI に置き換え、Ethernet で遠隔制御することを考えている。

5. 仮想化したことによる問題点

ゲスト PC で加速器制御関連のプログラムを実行するときの問題は、オーバーヘッドの増加である。特に完全仮想化では、ゲスト OS の通常の処理に加え、ホスト OS での仮想マシンの実行の処理およびゲスト OS の I/O の中継にかかる処理等の余分なオーバーヘッドが発生する。多くの I/O が発生する用途に用いる場合は、パフォーマンステストが必要である。

仮想化により、サポートの終わった古い実行環境が温存されるが、セキュリティの問題は解決しない。サポートが終わった OS やソフトウェアは、security fix も行われていないことが多く、ウィルス対策ソフト等もきちんと動作しない可能性がある。これはネットワークの構成や運用で対処するしかない。

6. まとめ

仮想化された PC を加速器制御に用いることは、サポートが終わったコンピューター環境の温存および PC 故障時のシステム停止時間の抑制できると思われる。しかし、仮想 PC はオーバーヘッドが大きくリアルタイム処理、I/O が大量に発生する用途には向いていないと思われる。

参考文献

[1] K. Miyamoto, et al., "Xen 徹底入門" 翔泳社 ISBN978-4-7981-2002-7
 [2] Paul Barham, et al., "Xen and the Art of Virtualization"