(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2010-40043
(P2010-40043A)
(43)公開日　平成22年2月18日(2010.2.18)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 13/10	(2006.01)	G06F	13/10	310Z	5B014
G06F 9/46	(2006.01)	G06F	13/10	330C	
G06F 9/50	(2006.01)	G06F	9/46	350	
		G06F	9/46	462Z	

審査請求 未請求　請求項の数 19　OL　（全 14 頁）

(21)出願番号　特願2009-168623 (P2009-168623)
(22)出願日　平成21年7月17日 (2009.7.17)
(31)優先権主張番号　10-2008-0076820
(32)優先日　平成20年8月6日 (2008.8.6)
(33)優先権主張国　韓国 (KR)
(31)優先権主張番号　10-2009-0044971
(32)優先日　平成21年5月22日 (2009.5.22)
(33)優先権主張国　韓国 (KR)

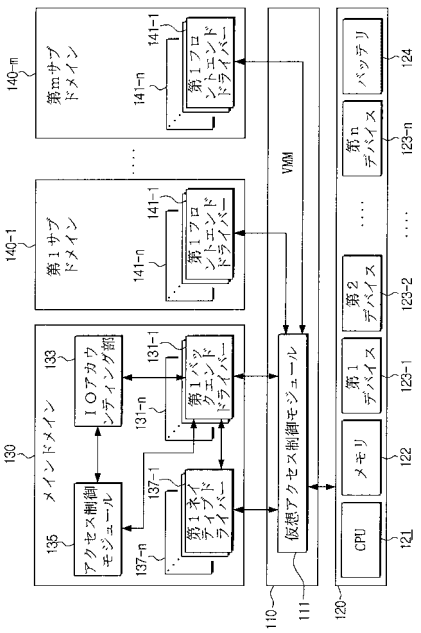
（特許庁注：以下のものは登録商標）
１．Ｌｉｎｕｘ

(71)出願人　390019839
三星電子株式会社
ＳＡＭＳＵＮＧ　ＥＬＥＣＴＲＯＮＩＣＳ　ＣＯ．，ＬＴＤ．
大韓民国京畿道水原市靈通区梅灘洞４１６
４１６，Ｍａｅｔａｎ－ｄｏｎｇ，Ｙｅｏｎｇｔｏｎｇ－ｇｕ，Ｓｕｗｏｎ－ｓｉ，Ｇｙｅｏｎｇｇｉ－ｄｏ　４４２－７４２（ＫＲ）
(74)代理人　100070150
弁理士　伊東　忠彦
(74)代理人　100091214
弁理士　大貫　進介
(74)代理人　100107766
弁理士　伊東　忠重

最終頁に続く

(54)【発明の名称】仮想化装置の制御方法及び仮想化装置

(57)【要約】
【課題】　本発明は、デバイス毎に使用可能なリソースを考慮して、ＩＯリクエストのアクセスを制御する仮想化装置の制御方法及び仮想化装置を提供する。
【解決手段】　メインドメイン、一つ以上のサブドメイン及び一つ以上のハードウェアデバイスを含む仮想化装置の制御方法は、サブドメインからハードウェアデバイスに対するＩＯ（Ｉｎｐｕｔ/Ｏｕｔｐｕｔ）リクエストを受信するステップと、ＩＯリクエストを行うのに必要なリソースに応じて、前記ＩＯリクエストによるハードウェアデバイスへのアクセスを制御するステップと、を含む。
【選択図】　図１



【特許請求の範囲】
【請求項１】
メインドメイン、一つ以上のサブドメイン及び一つ以上のハードウェアデバイスを含む仮想化装置の制御方法において、
サブドメインからハードウェアデバイスに対するＩＯ（Ｉｎｐｕｔ/Ｏｕｔｐｕｔ）リクエストを受信するステップと、
前記ＩＯリクエストを行うのに必要なリソースに応じて、前記ＩＯリクエストによる前記ハードウェアデバイスへのアクセスを制御するステップと、
を含むことを特徴とする仮想化装置の制御方法。
【請求項２】
前記制御するステップは、
前記ＩＯリクエストを行うのに必要なリソースと基準とを比較するステップと、
前記ＩＯリクエストを行うのに必要なリソースが前記基準より大きいと、前記ハードウェアデバイスへのアクセスを遮断するステップと、
を含むことを特徴とする請求項１に記載の仮想化装置の制御方法。
【請求項３】
前記制御するステップは、
前記ＩＯリクエストを行うのに必要なリソースと基準とを比較するステップと、
前記ＩＯリクエストを行うのに必要なリソースが前記基準より大きいと、前記基準に対応する量の前記ＩＯリクエストを処理し、前記処理が終わるまで残余ＩＯリクエストを遮断するステップと、
を含むことを特徴とする請求項１に記載の仮想化装置の制御方法。
【請求項４】
前記ＩＯリクエストを行うのに必要なリソースは、前記ＩＯリクエストのサイズであることを特徴とする請求項１に記載の仮想化装置の制御方法。
【請求項５】

前記ＩＯリクエストを行うのに必要なリソースは、前記ＩＯリクエストを行うのに必要なＣＰＵタイムであることを特徴とする請求項１に記載の仮想化装置の制御方法。
【請求項６】
前記ＩＯリクエストを行うのに必要なリソースは、前記ＩＯリクエストを行うのに必要なＣＰＵタイムの最大値であることを特徴とする請求項１に記載の仮想化装置の制御方法。
【請求項７】
バッテリの残量を確認するステップと、
前記確認された残量が基準値より小さいと、前記ハードウェアデバイスへのアクセスを遮断するステップと、
を更に含むことを特徴とする請求項１に記載の仮想化装置の制御方法。
【請求項８】
悪性プログラムコードが検出されるステップと、
前記悪性プログラムコードによって発生するＩＯリクエストによる前記ハードウェアデバイスへのアクセスを遮断するステップと、
を含むことを特徴とする請求項１に記載の仮想化装置の制御方法。
【請求項９】
一つ以上のハードウェアデバイスを含むシステムリソース部と、
前記ハードウェアデバイスに対するネイティブドライバーを含むメインドメインと、
前記ハードウェアデバイスに対するＩＯ（Ｉｎｐｕｔ/Ｏｕｔｐｕｔ）リクエストを前記メインドメインに伝送する一つ以上のサブドメインと、を含み、
前記メインドメインは、前記ＩＯリクエストを行うのに必要なリソースに応じて、前記ハードウェアデバイスへのアクセスを制御することを特徴とする仮想化装置。

【請求項１０】
前記メインドメインは、
前記サブドメインから前記ＩＯリクエストを受信し、前記ハードウェアデバイスと連動するバックエンドドライバーを更に含むことを特徴とする請求項９に記載の仮想化装置。
【請求項１１】
前記メインドメインは、
前記ＩＯリクエストを行うのに必要なリソースと基準とを比較し、前記ＩＯリクエストを行うのに必要なリソースが前記基準より大きい場合、前記ＩＯリクエストによる前記ハードウェアデバイスへのアクセスを遮断するアクセス制御モジュールを更に含むことを特徴とする請求項９に記載の仮想化装置。
【請求項１２】
前記メインドメインは、
前記ＩＯリクエストを行うのに必要なリソースと基準とを比較し、前記ＩＯリクエストを行うのに必要なリソースが前記基準より大きいと、前記基準に対応する量の前記ＩＯリクエストを処理し、前記処理が終わるまで残余ＩＯリクエストを遮断するアクセス制御モジュールを更に含むことを特徴とする請求項９に記載の仮想化装置。
【請求項１３】
前記メインドメインは、
前記ＩＯリクエストを行うのに必要なリソースを算出するＩＯアカウンティング部を更に含むことを特徴とする請求項９に記載の仮想化装置。
【請求項１４】
前記ＩＯリクエストを行うのに必要なリソースは、前記ＩＯリクエストのサイズであることを特徴とする請求項９に記載の仮想化装置。
【請求項１５】
前記ＩＯリクエストを行うのに必要なリソースは、前記ＩＯリクエストを行うのに必要なＣＰＵタイムであることを特徴とする請求項９に記載の仮想化装置。
【請求項１６】
前記メインドメインは、
バッテリの残量を確認し、前記確認された残量が基準より小さいと、前記ハードウェアデバイスへのアクセスを遮断するアクセス制御モジュールを更に含むことを特徴とする請求項９に記載の仮想化装置。
【請求項１７】
前記メインドメインは、
悪性プログラムコードを検出し、前記悪性プログラムコードによって発生するＩＯリクエストによる前記ハードウェアデバイスへのアクセスを遮断するアクセス制御モジュールを更に含むことを特徴とする請求項９に記載の仮想化装置。
【請求項１８】
前記仮想化装置は、ＣＥ（Ｃｏｎｓｕｍｅｒ　Ｅｌｅｃｔｒｏｎｉｃ）デバイスであることを特徴とする請求項９に記載の仮想化装置。
【請求項１９】
前記仮想化装置は、モバイルデバイスであることを特徴とする請求項９又は１８に記載の仮想化装置。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、仮想化装置の制御方法及び仮想化装置に関し、より詳細には、デバイス毎に使用可能なリソースを考慮してＩＯリクエストのアクセスを制御する仮想化装置の制御方法及び仮想化装置に関する。
【背景技術】
【０００２】

コンピューティング技術における仮想化は、一つのコンピューティング環境において多数の運営体制が動作できるようにする。各運営体制によって動作する環境をそれぞれのドメインとする場合、各ドメインはコンピューティング装置に設けられたデバイスを共有する。また、仮想化技術においては、多数のドメインのうちメインドメインがデバイスにアクセスすることができるネイティブドライバーを搭載する。従って、その以外のサブドメインはネイティブドライバーを介して実際のデバイスにアクセスする。

【０００３】

上記のような仮想化を実現するために、ＶＭＭ（Ｖｉｒｔｕａｌ Ｍａｃｈｉｎｅ Ｍｏｎｉｔｏｒ）は各ドメインに対して初期にＣＰＵタイムを割り当てる。従って、サブドメインからリクエストがある場合、メインドメインはサブドメインのリクエストを処理するためにメインドメインに割り当てられたＣＰＵタイムを使用し、これにより初期に割り当てられた時間よりさらに多い時間の間ＣＰＵを占有するようになる。

【０００４】

また、サブドメインにマルウェア（Ｍａｌｗａｒｅ）が設置されている場合、マルウェアは過渡なＩＯリクエストを送り、メインドメインはサブドメインのためのジョブを行うためにＣＰＵを消耗することになる。すなわち、メインドメインはサブドメインからのＤｏＳ攻撃によってメインドメインのＣＰＵを消耗することになり、結果としてメインドメインの主要サービスまたはアプリケーションの性能は低下する。また、マルウェアの過渡なリクエストはバッテリ又は電力消耗を増加させる。

【先行技術文献】

【特許文献】

【０００５】

【特許文献１】国際公開特許２００４－００６５１３号

【特許文献２】国際公開特許１９９９－０５６４３６号

【特許文献３】米国特許第５６０８７２６号明細書

【発明の概要】

【発明が解決しようとする課題】

【０００６】

そこで、本発明は、上記問題に鑑みてなされたものであり、本発明の目的とするところは、仮想化の環境において発生する主要サービスに対する性能低下、アプリケーションの性能障害、バッテリ消耗等を解決することができる、仮想化装置の制御方法及び仮想化装置を提供することにある。

【課題を解決するための手段】

【０００７】

上記の目的を達成するための本発明の一実施形態に係るメインドメイン、一つ以上のサブドメイン及び一つ以上のハードウェアデバイスを含む仮想化装置の制御方法は、サブドメインからハードウェアデバイスに対するＩＯ（Ｉｎｐｕｔ/Ｏｕｔｐｕｔ）リクエストを受信するステップと、前記ＩＯリクエストを行うのに必要なリソースに応じて、前記ＩＯリクエストによる前記ハードウェアデバイスへのアクセスを制御するステップと、を含む。

【０００８】

前記制御するステップは、前記ＩＯリクエストを行うのに必要なリソースと基準とを比較するステップと、前記ＩＯリクエストを行うのに必要なリソースが前記基準より大きいと、前記ハードウェアデバイスへのアクセスを遮断するステップと、を含むことができる。

【０００９】

前記制御するステップは、前記ＩＯリクエストを行うのに必要なリソースと基準とを比較するステップと、前記ＩＯリクエストを行うのに必要なリソースが前記基準より大きいと、前記基準に対応する量の前記ＩＯリクエストを処理し、前記処理が終わるまで残余ＩＯリクエストを遮断するステップと、を含むことができる。

【００１０】
前記ＩＯリクエストを行うのに必要なリソースは、前記ＩＯリクエストのサイズであってもよい。
【００１１】
前記ＩＯリクエストを行うのに必要なリソースは、前記ＩＯリクエストを行うのに必要なＣＰＵタイムであってもよい。
【００１２】
前記ＩＯリクエストを行うのに必要なリソースは、前記ＩＯリクエストを行うのに必要なＣＰＵタイムの最大値であってもよい。
【００１３】
バッテリの残量を確認するステップと、前記確認された残量が基準値より小さいと、前記ハードウェアデバイスへのアクセスを遮断するステップと、を更に含むことができる。
【００１４】
悪性プログラムコードが検出されるステップと、前記悪性コードによって発生するＩＯリクエストによる前記ハードウェアデバイスへのアクセスを遮断するステップと、を含むことができる。
【００１５】
本発明に係る仮想化装置は、一つ以上のハードウェアデバイスを含むシステムリソース部と、前記ハードウェアデバイスに対するネイティブドライバーを含むメインドメインと、前記ハードウェアデバイスに対するＩＯ（Ｉｎｐｕｔ/Ｏｕｔｐｕｔ）リクエストを前記メインドメインに伝送する一つ以上のサブドメインと、を含み、前記メインドメインは、前記ＩＯリクエストを行うのに必要なリソースに応じて、前記ハードウェアデバイスへのアクセスを制御する。
【００１６】
前記メインドメインは、前記サブドメインから前記ＩＯリクエストを受信し、前記ハードウェアデバイスと連動するバックエンドドライバーを更に含むことができる。
【００１７】
前記メインドメインは、前記ＩＯリクエストを行うのに必要なリソースと基準とを比較し、前記ＩＯリクエストを行うのに必要なリソースが前記基準より大きい場合、前記ＩＯリクエストによる前記ハードウェアデバイスへのアクセスを遮断するアクセス制御モジュールを更に含むことができる。
【００１８】
前記メインドメインは、前記ＩＯリクエストを行うのに必要なリソースと基準とを比較し、前記ＩＯリクエストを行うのに必要なリソースが前記基準より大きいと、前記基準に対応する量の前記ＩＯリクエストを処理し、前記処理が終わるまで残余ＩＯリクエストを遮断するアクセス制御モジュールを更に含むことができる。
【００１９】
前記メインドメインは、前記ＩＯリクエストを行うのに必要なリソースを算出するＩＯアカウンティング部を更に含むことができる。
【００２０】
前記ＩＯリクエストを行うのに必要なリソースは、前記ＩＯリクエストのサイズであってもよい。
【００２１】
前記ＩＯリクエストを行うのに必要なリソースは、前記ＩＯリクエストを行うのに必要なＣＰＵタイムであってもよい。
【００２２】
前記メインドメインは、バッテリの残量を確認し、前記確認された残量が基準より小さいと、前記ハードウェアデバイスへのアクセスを遮断するアクセス制御モジュールを更に含むことができる。
【００２３】

前記メインドメインは、悪性プログラムコードを検出し、前記悪性コードによって発生するＩＯリクエストによる前記ハードウェアデバイスへのアクセスを遮断するアクセス制御モジュールを更に含むことができる。

【００２４】

前記仮想化装置は、ＣＥ（Ｃｏｎｓｕｍｅｒ　Ｅｌｅｃｔｒｏｎｉｃ）デバイス又はモバイルデバイスであってもよい。

【発明の効果】

【００２５】

以上説明したように本発明によれば、各デバイス毎に使用可能なリソースを予め設定し、ＩＯリクエストに応じてメインドメイン又は各デバイスへのアクセスを制御することにより、Ｆｉｎｅ－Ｇｒａｉｎｅｄ　ＩＯ　Ａｃｃｅｓｓ　Ｃｏｎｔｒｏｌ（ＦＧＡＣ）を提供することができる。すなわち、本発明はＦＧＡＣを提供するために、メインドメインが第ｍサブドメインからのＩＯリクエストを処理する際、各デバイス毎に予め定義されたアクセス制御政策に基づいてＣＰＵのようなリソースの使用を制限することが可能である。

【００２６】

また、本発明によると、仮想化装置に設けられたマルウェアが過渡なＩＯリクエストを出力することによって発生するシステム性能の低下、バッテリ、ＣＰＵ等の不要な消耗を防止することができる。すなわち、ＩＯリクエスト毎にアクセスを制御するかまたは遮断することにより、リソースがなくなることを防止することができる。

【００２７】

また、本発明によると、各デバイス毎に細分化されたアクセス制御を提供することにより、ＤｏＳ攻撃を受けなかったデバイスがリソースを使用することにおいて制限を受けず、主要アプリケーションがリソースを活用して動作することが可能となる。

【００２８】

特に、仮想化装置がセルラーフォンのようにリソースの制限されたＣＥ（Ｃｏｎｓｕｍｅｒ　Ｅｌｅｃｔｏｎｉｃｓ）装置である場合、ＤｏＳ攻撃は致命的であるため、仮想化装置にＦＧＡＣ、すなわち、デバイス毎のアクセス制御をサポートすることでＤｏＳ攻撃から保護することができる。

【図面の簡単な説明】

【００２９】

【図１】本発明の好適な実施形態にかかる仮想化装置を示すブロック図である。

【図２】本発明の一実施形態にかかるメインドメインにおいてアクセス制御のための初期化を説明するためのフローチャートである。

【図３】本発明の実施形態にかかるデバイスに対するアクセス制御方法を説明するための図である。

【発明を実施するための形態】

【００３０】

以下に添付図面を参照しながら、本発明の好適な実施の形態について詳細に説明する。

【００３１】

図１は、本発明の好適な実施形態に係る仮想化装置を示すブロック図である。同図を参照すると、仮想化装置は、仮想マシンモニター（Ｖｉｒｔｕａｌ　Ｍａｃｈｉｎｅ　Ｍｏｎｉｔｏｒ：ＶＭＭ）１１０、システムリソース部１２０、メインドメイン１３０及び複数のサブドメイン１４０－１、・・・、１４０－ｍを含む。

【００３２】

物理的な仮想化装置は、例えばパソコン（例えば、デスクトップパソコン又はラップトップパソコン）、オーディオ装備又はモバイルデバイスのようなＣＥ（Ｃｏｎｓｕｍｅｒ　Ｅｌｅｃｔｒｏｎｉｃ）を含む。

【００３３】

ＶＭＭ１１０は、複数の運営体制を同時に実行させる。ＶＭＭ１１０を使用する仮想化

環境においては、バックエンドドライバー１３１－１、・・・、１３１－ｎとネイティブドライバー１３７－１、・・・、１３７－ｎがメインドメイン１３０に位置し、フロントエンドドライバー１４１－１、・・・、１４１－ｎが複数のサブドメイン１４０－１、・・・、１４０－ｍに位置する。仮想アクセス制御モジュール１１１は、各ドメイン１３０、１４０－１、・・・、１４０－ｍ毎にＣＰＵ使用量を制限する。

【００３４】

システムリソース部１２０は、仮想化装置のハードウェアデバイスを構成する。システムリソース部１２０は、中央処理装置（Ｃｅｎｔｒａｌ　Ｐｒｏｃｅｓｓｉｎｇ　Ｕｎｉｔ：ＣＰＵ）１２１、メモリ１２２、第１乃至第ｎデバイス１２３－１，・・・，１２３－ｎ及びバッテリ１２４を含み、その他の複数のデバイス又はストレージが備えられてもよい。

【００３５】

ＣＰＵ１２１は、仮想化装置においてデバイスの動作を処理する。

【００３６】

メモリ１２２には、ＩＯリクエストを行うのに必要なＣＰＵ使用量、すなわちリソースを算出するためのリソース予測公式が各デバイス１２３－１、・・・、１２３－ｎ毎にまたは各デバイス１２３－１、・・・、１２３－ｎに対して複数のＩＯリクエスト毎に保存される。ＣＰＵ使用量はＩＯリクエストを行うリソースを意味する。ＣＰＵ使用量は、例えばＣＰＵ使用時間、ＣＰＵタイムであってもよい。

【００３７】

また、メモリ１２２には、複数のサブドメイン１４０－１、・・・、１４０－ｍから出力されるＩＯリクエストに応じて、ＩＯリクエストに対応するデバイスへのアクセスを制御するのに必要なアクセス制御政策（Ａｃｃｅｓｓ　ｃｏｎｔｒｏｌ　ｐｏｌｉｃｙ）がローディングされて保存される。アクセス制御政策は、第ｍサブドメイン１４０－ｍから伝送されるＩＯリクエストに応じて、第ｍサブドメイン１４０－ｍが第ｎデバイス１２３－ｎへアクセスすることを制御又はアクセス方式をスケジューリングするための基準を含む。

【００３８】

アクセス制御政策は、システムリソース１２０に設けられた各デバイス１２３－１、・・・、１２３－ｎ毎に使用可能なＩＯリクエストサイズ又はＣＰＵ使用量の限界値をアクセス制御に必要な基準として設定している。言い換えれば、限界値は、各デバイス１２３－１、・・・、１２３－ｎにアクセスするためにアクセス制御政策が許容するＩＯリクエストサイズ又はＣＰＵ使用量の最大値であってよい。例えば、アクセス制御政策は、各フロントエンドドライバー１４１－１、・・・、１４１－ｎのＩＯリクエスト毎に、特定期間の間に使用可能なＣＰＵ使用量に対する最大値（又は、基準値）を設定することができる。ここで、特定期間は、ＣＰＵ　ｓｃｈｅｄｕｌｉｎｇ　ｐｅｒｉｏｄを意味することができる。以下では、各デバイス１２３－１、・・・、１２３－ｎ毎に設定された基準を例にとって説明する。

【００３９】

第１乃至第ｎデバイス１２３－１、・・・、１２３－ｎはＩＯ（ｉｎ/ｏｕｔ）デバイスであって、装置に情報を入力させる機能、装置の情報を出力させる機能、又は入力・出力の両機能を行う機器として、例えばインターフェースカード、ストレージ、ＭＴＤ（Ｍｅｍｏｒｙ　Ｔｅｃｈｎｏｌｏｇｙ　Ｄｅｖｉｃｅ）、コンソールデバイス等がある。

【００４０】

バッテリ１２４は、システムリソース部１２０に電力を供給するために備えられる。

【００４１】

メインドメイン１３０及び一つ以上のサブドメイン１４０－１、・・・、１４０－ｍは、個別運営体制で駆動される仮想化環境を提供する。メインドメイン１３０及び一つ以上のサブドメイン１４０－１、・・・、１４０－ｍは、同一の運営体制で駆動されてもよい。運営体制の例としては、Ｐａｌｍ、Ｓｙｍｂｉａｎ　ＯＳ、Ａｎｄｒｏｉｄ、Ｗｉｎｄ

ｏｗｓ（登録商標）Ｍｏｂｉｌｅ、Ｗｉｎｄｏｗｓ　Ｒ、Ｌｉｎｕｘ、Ｕｎｉｘ（登録商標）及びＤＯＳ等、様々である。

【００４２】

メインドメイン１３０は、インターネットバンキング、保安サービス、ボイスコール（ｖｏｉｃｅ　ｃａｌｌ）等、主要サービスまたは主要アプリケーションを提供する。メインドメイン１３０は、メインドメイン１３０だけでなく複数のサブドメイン１４０－１、・・・、１４０－ｍが実際のデバイス、すなわち、第１乃至第ｎデバイス１２３－１、・・・、１２３－ｎにアクセスできる経路を提供する。従って、メインドメイン１３０はサーバ概念で動作し、複数のサブドメイン１４０－１、・・・、１４０－ｍはクライアント概念で動作する。

【００４３】

このため、メインドメイン１３０は、第１乃至第ｎバックエンドドライバー１３１－１、・・・、１３１－ｎ、ＩＯアカウンティング部１３３、アクセス制御モジュール（Ａｃｃｅｓｓ　Ｃｏｎｔｒｏｌ　Ｍｏｄｕｌｅ）１３５及び第１乃至第ｎネイティブドライバー１３７－１、・・・、１３７－ｎを含む。

【００４４】

第１乃至第ｎバックエンドドライバー１３１－１、・・・、１３１－ｎは、複数のサブドメイン１４０－１、・・・、１４０－ｍとメインドメイン１３０との通信経路を提供し、それぞれ第１乃至第ｎデバイス１２３－１、・・・、１２３－ｎに対応する。第１乃至第ｎバックエンドドライバー１３１－１、・・・、１３１－ｎはサブドメイン１４０－１、・・・、１４０－ｍから入力されるＩＯリクエストをＩＯアカウンティング部１３３に提供する。

【００４５】

例えば、第１サブドメイン１４０－１のアプリケーション（図示せず）から第１デバイス１２３－１の使用がリクエストされると、第１フロントエンドドライバー１４１－１は第１デバイス１２３－１に対するＩＯリクエストを伝送し、第１バックエンドドライバー１３１－１が前記ＩＯリクエストを受信してＩＯアカウンティング部１３３に提供する。ここで、第１デバイス１２３－１はターゲットデバイスである。

【００４６】

ＩＯアカウンティング部１３３は、如何に多いＣＰＵリソースがＩＯリクエストのために使用されるのかを測定するための公式を提供する。メモリ１２２が新たにインストールされたデバイスのための公式を持っていなければ、又は、新たな公式の生成が要請されると、リソース予測公式を生成する。たとえば、仮想化環境が適用される装置の工場出荷時、又は新たなデバイスがシステムリソース部１２０に設置される場合、又はユーザが公式のアップデートを要請する場合に、ＩＯアカウンティング部１３３はリソース予測公式を新たに生成する。

【００４７】

公式の生成が要請されると、ＩＯアカウンティング部１３３は、各デバイス１２３－１、・・・、１２３－ｎ毎にＩＯリクエスト当りリソースをどのくらい使用するかを予測する公式を算出する。公式を算出する方法は次のようである。

【００４８】

第ｍサブドメイン１４０－ｍは、各デバイス１２３－１、・・・、１２３－ｎ毎に予め定義された多様なサイズのＩＯリクエストをメインドメイン１３０に伝送し、ＩＯアカウンティング部１３３は、多様なサイズのＩＯリクエストを処理するのに所要されるリソース、例えば、ＣＰＵタイムから各デバイス１２３－１、・・・、１２３－ｎ毎または各フロントエンドドライバー１４１－１、・・・、１４１－ｎ毎の第ｍサブドメイン１４０－ｍのリソース予測公式を算出する。

【００４９】

すなわち、ＩＯアカウンティング部１３３は、第ｎフロントエンドドライバー１４１－ｎが第ｎバックエンドドライバー１３１－ｎに多様なサイズのテストパケットを伝送した

場合、第ｎネイティブドライバー１３７－ｎ又は第ｎデバイス１２３－ｎが各パケットを処理するのに所要されるＣＰＵタイムの変化量をテストして回帰式、すなわち、リソース予測公式を算出する。

【００５０】

デバイス１２３－１、・・・、１２３－ｎ毎の公式は、アクセス制御モジュール１３５又はメモリ１２２に設定され、今後の各デバイス１２３－１、・・・、１２３－ｎの駆動時にデバイスアクセス制御に使用される。

【００５１】

アクセス制御モジュール１３５は、設定された各デバイス１２３－１、・・・、１２３－ｎ毎の公式を用いて、サブドメイン１４０－１、・・・、１４０－ｍから伝送されるＩＯリクエストがメインドメイン１３０又はデバイス１２３－１、・・・、１２３－ｎへアクセスすることを制御する。すなわち、アクセス制御モジュール１３５は、ＩＯリクエストを行うのに必要なリソースを該当リソース予測公式から算出し、算出されたリソースと保存された基準とを比較して、ＩＯリクエストのアクセスを制御したり、又はアクセス方式をスケジューリングする。

【００５２】

特に、アクセス制御モジュール１３５は、第ｍサブドメイン１４０－ｍからのＩＯリクエストが、第ｎデバイス１２３－ｎにおいて行われるべきリクエストであり、前記ＩＯリクエストを行うのに必要なリソースがメモリ１２２に保存された第ｎデバイス１２３－ｎに対する基準を超過すると、予め設定された基準に相当する量のＩＯリクエストを処理し、残りのＩＯリクエストはブロッキングしてキュー（Ｑｕｅｕｅ）（図示せず）のような付加保存機器（ａｄｄｉｔｉｏｎａｌ　ｓｔｏｒａｇｅ　ｄｅｖｉｃｅ）に保存する。

【００５３】

例えば、第１デバイス１２３－１に関連したＩＯリクエストのパケットを処理するのに必要なリソースが１０ｍｓｅｃであり、第１デバイス１２３－１にマッピングされた基準が２ｍｓｅｃであると、アクセス制御モジュール１３５は、２ｍｓｅｃのＩＯリクエストが先に処理されるようにし、処理される間に残りの８ｍｓｅｃに該当するＩＯリクエストはブロッキングする。２ｍｓｅｃのＩＯリクエストが処理されると、アクセス制御モジュール１３５は、残りの８ｍｓｅｃのうち２ｍｓｅｃを処理し残りの６ｍｓｅｃはブロッキングする。

【００５４】

アクセス制御モジュール１３５は、ＩＯリクエストを行うのに必要なリソースだけのＩＯリクエストサイズを使用することができる。例えば、第１デバイス１２３－１に対するＩＯリクエストに関連したパケットサイズが４ＭＢ（ｍｅｇａ　ｂｙｔｅ）であり、第１デバイス１２３－１に対する基準値が１ＭＢである場合、アクセス制御モジュール１３５は、まず１ＭＢに対応する量のＩＯリクエストを処理し、前記１ＭＢに対応するＩＯリクエストを処理する間に３ＭＢに対応する量の残余ＩＯリクエストを遮断する。１ＭＢに対応する量のＩＯリクエストの処理が完了すると、アクセス制御モジュール１３５は、残りの３ＭＢのうち１ＭＢを処理し、その以外の２ＭＢを遮断する。残りの２ＭＢは同一の方式で処理される。

【００５５】

また、ＩＯリクエストを行うのに必要なリソースがメモリ１２２に保存された基準よりｋ倍以上大きい場合、アクセス制御モジュール１３５は、ＩＯリクエストがマルウェアによる攻撃であると判断し、ＩＯリクエストを完全に遮断してＩＯリクエストが処理されないようにする。ｋ値は設計者またはユーザによって設定及び変更されても良い。マルウェアは、第１乃至第ｍサブドメイン１４０－１、・・・、１４０－ｍのうち少なくとも一つに設けられてもよい。

【００５６】

第１乃至第ｎネイティブドライバー１３７－１、・・・、１３７－ｎは、システムリソース部１２０に備えられた第１乃至第ｎデバイス１２３－１、・・・、１２３－ｎにアク

セスできるドライバー、即ち実際のドライバーである。第１乃至第ｎネイティブドライバー１３７－１、・・・、１３７－ｎはそれぞれ第１乃至第ｎデバイス１２３－１、・・・、１２３－ｎに対応する。

【００５７】

第１乃至第ｍサブドメイン１４０－１、・・・、１４０－ｍ（ｍは、定数）は、第１乃至第ｎフロントエンドドライバー１４１－１、・・・、１４１－ｎを介してメインドメイン１３０と第１乃至第ｎデバイス１２３－１、・・・、１２３－ｎにアクセスする。第１乃至第ｎバックエンドドライバー１３１－１、・・・、１３１－ｎと第１乃至第ｎフロントエンドドライバー１４１－１、・・・、１４１－ｎは、第１乃至第ｎネイティブドライバー１３７－１、・・・、１３７－ｎ及び第１乃至第ｎデバイス１２３－１、・・・、１２３－ｎと連動し、第１乃至第ｎデバイス１２３－１、・・・、１２３－ｎの第１乃至第ｎネイティブドライバー１３７－１、・・・、１３７－ｎが設置されるときに自動で生成されても良い。

【００５８】

図２は、本発明の一実施形態に係るメインドメインにおいてアクセス制御のための初期化を説明するためのフローチャートである。

【００５９】

図１及び図２を参照すると、第ｎデバイス１２３－ｎに対するリソース予測公式の生成が要請されると（Ｓ２１０）、第ｎフロントエンドドライバー１４１－ｎは第ｎデバイス１２３－ｎに対して定義された多様なサイズのＩＯリクエストをメインドメイン１３０の第ｎバックエンドドライバー１３１－ｎに伝送し、第ｎバックエンドドライバー１３１－ｎは多様なサイズのＩＯリクエストをＩＯアカウンティング部１３３に提供する（Ｓ２２０）。

【００６０】

ＩＯアカウンティング部１３３は、多様なサイズのＩＯリクエストを処理するのに所要されるリソースを用いて、第ｎデバイス１２３－ｎに対するリソース予測公式を生成する（Ｓ２３０）。また、ＩＯアカウンティング部１３３は、毎ＣＰＵ Ｓｃｈｅｄｕｌｉｎｇ ｐｅｒｉｏｄの間、特定のドメインが特定のデバイスを使用するために要請したデータ（ＣＰＵ ｔｉｍｅ）の和を計算する。Ｓ２１０乃至Ｓ２３０ステップは、第１乃至第ｎデバイス１２３－１、・・・、１２３－ｎに対する公式を生成する場合に動作するが、説明の便宜上、第ｎデバイス１２３－ｎに対して記載する。

【００６１】

第１乃至第ｎデバイス１２３－１、・・・、１２３－ｎ毎の公式が生成されると、アクセス制御モジュール１３５は、生成された各公式を各デバイス１２３－１、・・・、１２３－ｎ毎にアクセス制御モジュール１３５又はメモリ１２２に設定する（Ｓ２４０）。

【００６２】

アクセス制御モジュール１３５は、予め設定されたアクセス制御政策をメモリ１２２にローディングする（Ｓ２５０）。

【００６３】

一方、Ｓ２１０ステップにおいて、公式の生成が要請されない場合、アクセス制御モジュール１３５は、アクセス制御政策がメモリ１２２にローディングされているかを判断する（Ｓ２６０）。ローディングされていなければ、アクセス制御モジュール１３５はアクセス制御政策をメモリ１２２にローディングする。これにより、ＩＯリクエストに対するアクセス制御のための初期化が完了する。

【００６４】

本発明の一つ以上の実施形態に係る仮想化装置の制御方法及び仮想化装置によると、仮想化環境時に発生する主要サービスに対する性能低下、アプリケーションの性能障害などを解決し、バッテリの消耗を最小化することが可能である。

【００６５】

図３は、本発明の実施形態に係る仮想化装置のアクセス制御方法を説明するための図で

ある。
【００６６】
図１乃至図３を参照すると、第１乃至第ｍサブドメイン１４０－１、・・・、１４０－ｎのうち、一つのフロントエンドドライバーを介してＩＯリクエストがメインドメイン１３０に伝送されると（Ｓ３１０）、ＩＯアカウンティング部１３３は、ＩＯリクエストに対応する公式を用いてＩＯリクエストが使用するリソースを算出する（Ｓ３２０）。
【００６７】
例えば、第ｎフロントエンドドライバー１４１－ｎからＩＯリクエストが伝送されると、第ｎバックエンドドライバー１３１－１、・・・、１３１－ｎはＩＯリクエストを受信してＩＯアカウンティング部１３３に提供する。ＩＯアカウンティング部１３３はメモリ１２２に保存された第ｎデバイス１２３－ｎに対応するリソース予測公式を読み出し、Ｓ３１０ステップで伝送されたＩＯリクエストを読み出したリソース予測公式に代入して、ＩＯリクエストが使用するリソースを算出及び予測する。このとき、また、ＩＯアカウンティング部１３３は、毎ＣＰＵ ｓｃｈｅｄｕｌｉｎｇ ｐｅｒｉｏｄの間、特定のドメインが特定のデバイスを使用するために要請したデータ（ＣＰＵ ｔｉｍｅ）の和を計算する。この総和は次回のＣＰＵ ｓｃｈｅｄｕｌｉｎｇ ｐｅｒｉｏｄのとき０にリセットされる。
【００６８】
アクセス制御モジュール１３５は、Ｓ３２０ステップで算出されたリソースとメモリ１２２にローディングされた第ｎデバイス１２３－ｎに対応するアクセス制御政策の基準とを比較して、第ｎデバイス１２３－ｎがＩＯリクエストを行うことができるかを判断する（Ｓ３３０）。アクセス制御政策は、毎ＣＰＵ Ｓｃｈｅｄｕｌｉｎｇ ｐｅｒｉｏｄの間、特定のサブドメインがメインドメインの提供する特定のデバイスの使用のために、メインドメインがこれに対する処理のために如何に多いＣＰＵタイムを使用できるかを明示する。
【００６９】
比較結果、算出されたリソースが第ｎデバイス１２３－ｎに対応する基準より小さいと、アクセス制御モジュール１３５は、第ｎデバイス１２３－ｎがＩＯリクエストを行うことができると判断し、Ｓ３１０ステップにおいて伝送されたＩＯリクエストが行われるようにする（Ｓ３４０）。例えば、アクセス制御モジュール１３５は、ＩＯリクエストが第ｎバックエンドドライバー１３１－１、・・・、１３１－ｎ及び第ｎネイティブドライバー１３７－１、・・・、１３７－ｎを介して第ｎデバイス１２３－ｎに伝送され、ＩＯリクエストに該当する動作が行われるようにする。
【００７０】
Ｓ３４０ステップは、すなわち、ＩＯアカウンティング部１３３において、毎ＣＰＵ ｓｃｈｅｄｕｌｉｎｇ ｐｅｒｉｏｄの間、特定のサブドメインがメインドメインの特定のデバイスに対して要請した和がアクセス制御政策に記述された基準より小さければ行われる。
【００７１】
一方、算出されたリソースが第ｎデバイス１２３－ｎに対応する基準より大きいと（Ｓ３３０－Ｎ）、アクセス制御モジュール１３５は、第ｎデバイス１２３－ｎに対するアクセス制御政策に従ってリソースの使用を制限する（Ｓ３５０）。特に、アクセス制御モジュール１３５はアクセス制御政策が許容する基準までのＩＯリクエストを、次回のＳｃｈｅｄｕｌｉｎｇ ｐｅｒｉｏｄに順次処理するか又は特定の時間の間ＩＯリクエストをブロッキングする。例えば、アクセス制御モジュール１３５は、メモリ１２２にローディングされたアクセス制御政策のＣＰＵタイム又はＩＯリクエストサイズまでＩＯリクエストが順次行われるようにする。また、ＩＯリクエストがマルウェアからの攻撃であると判断されると、アクセス制御モジュール１３５はＩＯリクエストのアクセスを最初から遮断し、ターゲットデバイス（例えば、第ｎデバイス）へのアクセスを制御する。
【００７２】

一方、本発明の例によると、アクセス制御モジュール１３５又はその他のセンサ（図示せず）は、バッテリ１２４の残量を確認することが可能であり、アクセス制御政策はバッテリ残量によるアクセス制御政策を含む。確認されたバッテリ１２４の残量がアクセス制御政策に設定されたバッテリ残量の臨界値に到達すると、アクセス制御モジュール１３５はＣＰＵ使用量を制限することができる。この場合、アクセス制御政策には各デバイス１２３－１、・・・、１２３－ｎ毎に異なるバッテリの残量が設定されることができ、アクセス制御モジュール１３５は相違に設定されたバッテリの残量を考慮して各デバイス１２３－１、・・・、１２３－ｎ毎にＣＰＵ使用量を制限することができる。

【００７３】

また、メインドメイン１３０には動的にマルウェアのような悪性コードの侵入を探知する侵入探知モジュール（図示せず）が備えられ、アクセス制御モジュール１３５は侵入探知モジュールにより悪性コードの侵入が探知されると、ＩＯアクセスができないように制御することができる。

【００７４】

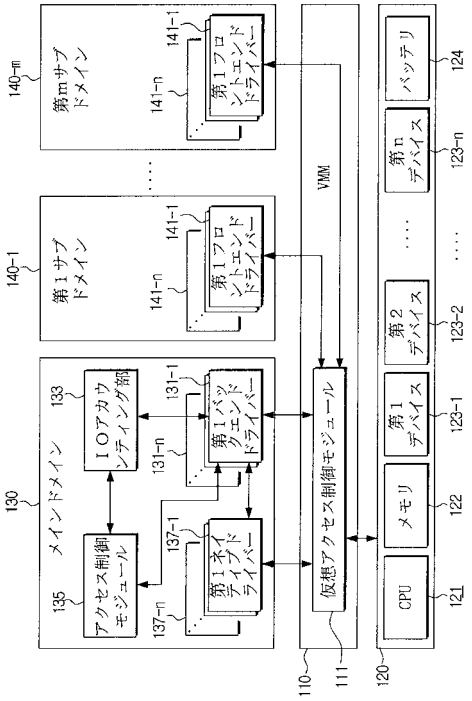
以上、添付図面を参照しながら本発明の好適な実施形態について詳細に説明したが、本発明はかかる例に限定されない。本発明の属する技術の分野における通常の知識を有する者であれば、特許請求の範囲に記載された技術的思想の範疇内において、各種の変更例または修正例に想到し得ることは明らかであり、これらについても、当然に本発明の技術的範囲に属するものと了解される。

【符号の説明】

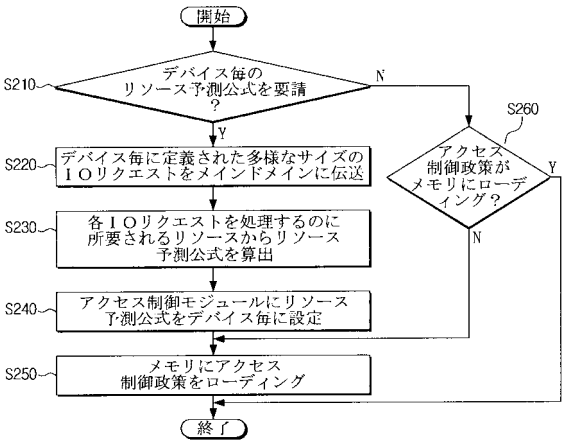
【００７５】

１１０　仮想マシンモニター
１２０　システムリソース部
１３０　メインドメイン
１４０－１、・・・、１４０－ｎ　サブドメイン
１３１－１、・・・、１３１－ｎ　第１乃至第ｎバックエンドドライバー
１３３　ＩＯアカウンティング部
１３５　アクセス制御モジュール
１３７－１、・・・、１３７－ｎ　第１乃至第ｎネイティブドライバー
１４１－１、・・・、１４１－ｎ　第１乃至第ｎフロントエンドドライバー

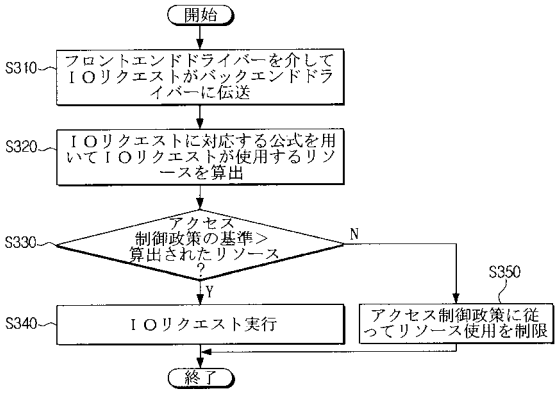
【図１】


第mサブドメイン
140-m
141-1
141-n
第1フロントエンドドライバー
第1サブドメイン
140-1
メインドメイン
130
IOアカウンティング部
133
131-1
131-n
第1バックエンドドライバー
アクセス制御モジュール
135
137-1
137-n
第1ネイティブドライバー
VMM
仮想アクセス制御モジュール
111
110
バッテリ
124
第nデバイス
123-n
第2デバイス
123-2
第1デバイス
123-1
メモリ
122
CPU
121
120

【図２】


開始
S210
デバイス毎のリソース予測公式を要請？
N
Y
S220
デバイス毎に定義された多様なサイズのIOリクエストをメインドメインに伝送
S230
各IOリクエストを処理するのに所要されるリソースからリソース予測公式を算出
S240
アクセス制御モジュールにリソース予測公式をデバイス毎に設定
S250
メモリにアクセス制御政策をローディング
S260
アクセス制御政策がメモリにローディング？
Y
N
終了

【図３】


開始
S310
フロントエンドドライバーを介してIOリクエストがバックエンドドライバーに伝送
S320
IOリクエストに対応する公式を用いてIOリクエストが使用するリソースを算出
S330
アクセス制御政策の基準＞算出されたリソース？
N
Y
S340
IOリクエスト実行
S350
アクセス制御政策に従ってリソース使用を制限
終了

フロントページの続き

(72)発明者　李　聖　民
大韓民国京畿道龍仁市器興区農書洞山１４－１番地　三星綜合技術院内
(72)発明者　鄭　福　得
大韓民国京畿道龍仁市器興区農書洞山１４－１番地　三星綜合技術院内
(72)発明者　徐　尚　範
大韓民国京畿道龍仁市器興区農書洞山１４－１番地　三星綜合技術院内

Ｆターム(参考) 5B014 FB05