

VMMを用いたディスクアクセス高速化によるOS起動時間短縮

大久保 諒 † 表 祐 志 †
品川 高 廣 ‡ 加 藤 和 彦 †

1. はじめに

OSの起動は電源投入後や再起動時に必要な処理であり、起動時間の長さはユーザビリティに直結する。シマンテック社の調査によると、OS起動に要する時間は長ければ長いほどユーザにかかるストレスは大きくなる³⁾。OS起動処理には様々な要因が関係しているがディスクI/Oがボトルネックになることが多い。そのため、ディスクI/Oを高速化することによりOS起動時間を短縮することができると考えられる。OS起動時間を短縮する関連技術としてWindowsのReadyBoot²⁾やLinuxのsystemd¹⁾などの機構がある。しかしながらこれらの手法は特定のOSに依存した高速化手法である。

本研究では、OS起動時のディスクアクセスパターンに着目し、仮想マシンモニタ(VMM, Virtual Machine Monitor)を用いてOSに依存せずに、ディスクアクセスを高速化することによってOS起動時間を短縮する。具体的には、OS起動時に発生するランダムアクセスされるデータを、シーケンシャルアクセスでまとめて先読みする。特にハードディスクから読み出す場合は、ランダムアクセスは大きなオーバーヘッドとなるため、提案手法での高速化の効果が期待できる。OS起動後はハードウェアへのアクセスを全てパススルーとすることで、仮想マシンモニタのオーバーヘッドを抑える。

2. 既存のOS起動高速化手法

Windowsに搭載されたOS起動時間を短縮する機能として、Windows ReadyBoot²⁾がある。これはOS起動時のディスクアクセスを監視、解析し次回起動時にキャッシュすることでOS起動に要する時間を短縮する。Linuxにはsystemd¹⁾というOS初期化のプロセスを決定する機構が存在し、これはOS起動処理を

可能な限り並列化し、かつ不必要なプロセスの起動を回避する事でOS起動時間を短縮する。

しかし、これらの既存技術は特定のOSにのみ適用可能な手法である。

3. VMMによるOS起動時間短縮

本研究では、OS起動時のディスクアクセスパターンに着目し、仮想マシンモニタを用いてディスクアクセスを高速化することによりOS起動時間を短縮する。具体的には、OS起動時にアクセスされるデータを、予め連続領域にキャッシュしておき、起動時にシーケンシャルアクセスで先読みする。OSの下のレイヤで動作する仮想マシンモニタを用いることで、OSの種類に依存せず起動時間を短縮することを可能にする。OS起動後は全てのハードウェアアクセスをパススルーとすることで、仮想マシンモニタのオーバーヘッドを抑える。

本研究の提案する手法は、(1)ディスクアクセスを最適化するフェーズと、(2)最適化されたディスクアクセスにより高速起動するフェーズの二段階で構成される。(1)ディスクアクセスを最適化するフェーズにおける処理の流れを図1に示す。まずOS起動時にどのようなディスクアクセスが発生しているかのログを記録する。そして起動処理後にこのログを用い、OS起動時にアクセスされるデータを、ディスク上の連続的な領域に配置する。ここで、(2)高速起動フェーズにおいて図2に示す通り、OS起動に先立ち連続的な領域に配置されたデータをシーケンシャルアクセスにより先読みし、本来発生するランダムアクセスを避け、代わりに先読み済データへのアクセスに変換する。これよりランダムアクセスを回避し、I/Oの発生回数を集約する。

実装はType I仮想マシンモニタBitVisor⁴⁾のATAデバイスドライバをベースとし、ATAデバイスへのI/Oアクセスを捕捉・変換する形でおこなった。また、ブートディスク中に連続領域としてパーティションを切り、ここに先読みデータを保存した。

† 筑波大学システム情報工学研究科

‡ 東京大学情報基盤センター情報メディア教育研究部門

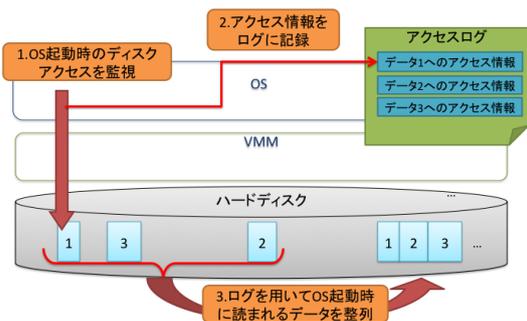


図 1 ディスクアクセス最適化

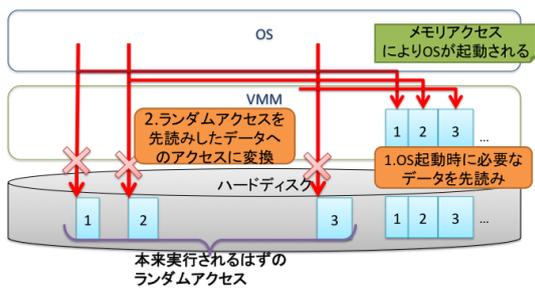


図 2 シーケンシャルアクセスによる OS 高速起動

表 1 実験環境

CPU	Core i5 670 3.47GHz
メモリ	4GB
ディスク	WD800BEVS-08RS (5400rpm)

4. 評価実験

提案手法により OS 起動時間が短縮されることを確認するため、OS を直接起動する場合と提案手法を適用した場合について、起動に要する時間を測定した。実験はストップウォッチを用い電源投入後ブートローダの選択画面が表示されるまでと、OS のログイン画面が表示されるまでの時間を測定した。また、提案手法を適用した場合は BitVisor の起動時間も測定した。測定対象の OS は Windows 7 と Ubuntu 10.04 とし、表 1 の環境を用いて行った。

Ubuntu 10.04 をインストールし実験した結果は図 3、Windows 7 をインストールして実験した結果は図 4 の通りである。図 3 については全体で 7 秒の OS 起動時間短縮に成功している。また図 4 についても、全体では 5 秒ほど起動時間が短縮されている。これより提案手法によって OS 起動時間が短縮した。しかし Windows 環境においては ReadyBoot による高速化がされた後では提案手法による高速化効果が実現出来ない場合もあり、OS レベルでの最適化と競合しないようするという課題がある。

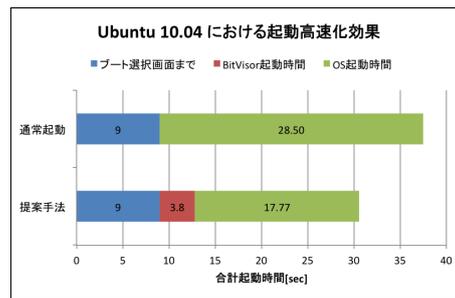


図 3 実験結果 1

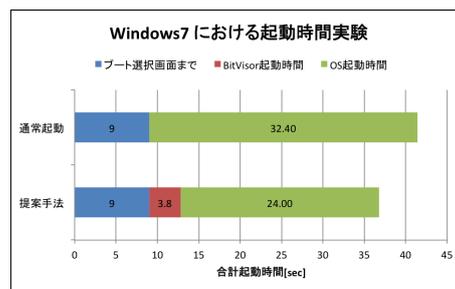


図 4 実験結果 2

5. まとめと今後の課題

本研究では、OS 起動処理においてボトルネックになりがちなディスク I/O を、仮想マシンモニタ BitVisor を用いて高速化し、OS に依存せず起動時間を短縮した。今後の課題は、OS 起動処理においてディスクアクセス以外に高速化できる要素の模索が挙げられる。また、OS の再起動やアプリケーションの起動など、ディスク I/O がボトルネックになりうるワークロードへの対応も考えている。更に BitVisor を用いることによるオーバーヘッドの低減や、OS レベルでの高速化手法との共存を考える必要がある。

参考文献

- freedesktop.org - software/systemd. <http://freedesktop.org/wiki/Software/systemd>.
- Windows pc アクセラレータ. <http://msdn.microsoft.com/ja-jp/windows/hardware/gg463388>.
- ユーザーは起動の遅さに不満—シマンテックの調査より. <http://pc.nikkeibp.co.jp/article/NEWS/20070613/274732/>.
- Takahiro Shinagawa et al., Bitvisor: a thin hypervisor for enforcing I/O device security. In *Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, VEE '09*, pages 121–130, New York, NY, USA, 2009. ACM.