

擬似不揮発性メモリを用いた OFF2F プログラム実行性能の評価

Evaluation of OFF2F Program Execution Performance Using Pseudo Non-Volatile Memory

高杉 頌
Sho Takasugi佐藤 将也
Masaya Sato谷口 秀夫
Hideo Taniguchi

1. はじめに

従来に比べ、アクセス速度が速い不揮発性メモリ (Non-Volatile Memory, 以降, NV メモリ) が登場しており, NV メモリを有効利用するソフトウェア技術の研究が行われている [1]. しかし, 揮発性メモリと比べると, NV メモリはアクセス速度が低速であり, 少容量かつ高価格である. このため, 実メモリが全て NV メモリ搭載になる構成ではなく, 揮発性メモリと NV メモリが混載する実メモリ構成のプロセッサ環境 (以降, 混載環境) が主流になると考えられる. 混載環境を対象に, 新たな実行ファイルの形式 (OFF2F: Object File Format consisting of 2 Files) が提案されている [2]. しかし, 現在のところ混載環境の計算機は身近に存在せず, OFF2F の効果を検証できない. このため, 擬似 NV メモリを構築し, OFF2F プログラムの実行を実現した [3].

本稿では, 擬似不揮発性メモリを用いた OFF2F プログラムの実行性能を評価し, 既存の実行ファイル形式のプログラムと比較して OFF2F の有効性を示す.

2. OFF2F

実行ファイルは, 4 つの内容 (ヘッダ部, テキスト部, データ部, および関係情報部) から構成されている. ELF (Executable and Linkable Format) などの実行ファイルでは, これらは 1 つのファイルに格納されている. これに対し, OFF2F は, テキスト部とそれ以外を別ファイルにする形式である. 具体的には, ヘッダ部, データ部, および関係情報部を 1 つのファイルとし, テキスト部のみを別ファイルとする.

OFF2F プログラムを実行する時のマッピングの様子を図 1 に示す. NV メモリの読み出しは速く, 書き込みは遅い. そこで, OFF2F を用いて, 読み出しのみ行われるテキスト部を NV メモリ上に格納し, そのまま仮想メモリ空間にマッピングする. これにより, 外部記憶装置から実メモリへの読み出しを削減し, プログラム実行を高速化する.

3. 擬似不揮発性メモリ

3.1 実現方式

揮発性メモリのみで構成される既存計算機を用いて, 擬似 NV メモリを実現する. 実メモリの一部をカーネルの使用しない領域 (以降, OS 管理外領域) とし, この領域を擬似 NV メモリとする. 擬似 NV メモリは, OS 管理外領域であるため, 初期化処理に利用されない. このため, ソフトウェアリセット時において, NV メモリの不

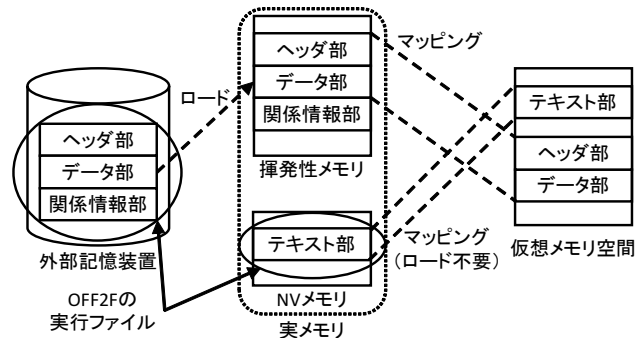


図 1 OFF2F プログラム実行時のマッピング

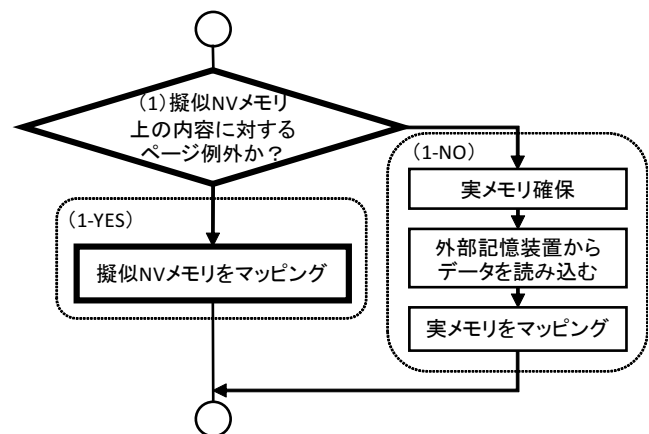


図 2 ページ例外処理の流れ

揮発性を再現する. また, 擬似 NV メモリには, OFF2F プログラムのテキスト部のファイルを格納する.

3.2 ページ例外処理

擬似 NV メモリを用いた OFF2F プログラム実行では, 擬似 NV メモリ上にあるテキスト部のファイルを利用する. このため, ページ例外処理を変更する必要がある. 擬似 NV メモリを利用したページ例外処理の流れを図 2 に示し, 以下に説明する.

- (1) 擬似 NV メモリ上の内容に対するページ例外であるかを判定する.
- (1-YES) 擬似 NV メモリ上の内容に対するページ例外である場合, 擬似 NV メモリを探索し, 対応する実メモリをマッピングする.
- (1-NO) 擬似 NV メモリ上の内容に対するページ例外ではない場合, 既存の要求時ページング処理を行う.

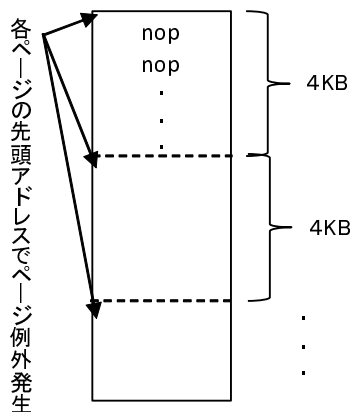


図3 評価対象のプログラムの処理流れ

4.OFF2F プログラムの評価

4.1 観点

評価として、テキスト部全体にアクセスが発生するELFおよびOFF2Fのプログラムを実行し、実行時間を比較することで、OFF2Fの有効性を明らかにする。評価対象のプログラムの処理流れを図3に示す。評価対象のプログラムでは、実行時にテキスト部のnop命令を読んだ後、returnにより処理が終了する。これにより、実行時の処理内容が単純となり、外部記憶装置からの読み出しやページ例外処理以外によるオーバーヘッドを少なくした。

評価環境であるFreeBSD 11.0-RELEASEでは、プログラム実行の際に実行ファイルの先頭64KBが読み込まれる。このため、64KBよりサイズの小さいプログラムでは、テキスト部に対するページ例外は発生しない。ここで、評価対象のプログラムとして、nop命令数を増やすことで、テキスト部の大きさが32KB、64KB、96KB、および128KBのプログラムを作成した。これにより、本評価では、テキスト部が64KBより大きいプログラムの実行時間を測定することで、ページ例外処理におけるOFF2Fの有効性について明らかにする。なお、いずれのプログラムにおいても、データ部の大きさは400B程度である。

評価対象のOFF2Fのプログラムについて、実行時間測定の際は、テキスト部のファイルを擬似NVメモリにあらかじめ格納し、ソフトウェアリセットを行った後に実行する。次に、測定した実行時間について、ELFプログラムのものと比較を行う。

4.2 評価環境

評価は、Intel Core i3-6100T(3.20GHz)と2GBのメモリを搭載した計算機を用いた。なお、本評価では、OSが利用できるメモリの大きさを1GBとし、残りの1GBを擬似NVメモリとした。OSは、FreeBSD 11.0-RELEASEを使用し、1ページの大きさが4KBの環境とした。

4.3 結果と考察

結果を図4に示す。図4より、以下のことが分かる。
(1) 32KB~128KBのどのプログラムにおいても、OFF2Fプログラムの実行時間はELFのものより短い。

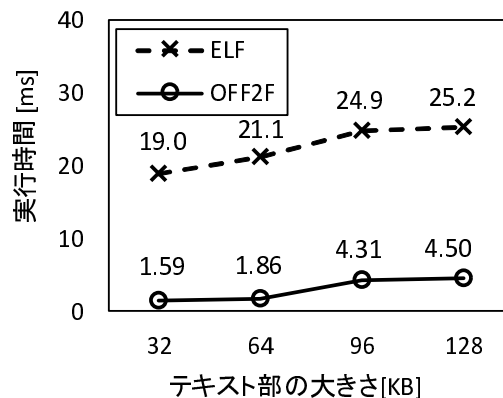


図4 プログラム実行時間

これは、テキスト部に対するページ例外の際に、ELFでは外部記憶装置からファイルを読み込む一方で、OFF2Fでは擬似NVメモリのページをそのままマッピングするためである。

(2) プログラムのテキスト部が大きくなるにつれ、OFF2Fに比べELFのプログラム実行時間は長くなる。ELFでは、テキスト部が大きくなるにつれ、外部記憶装置から読みだされるデータサイズが大きくなる。これに対し、OFF2Fプログラム実行時、外部記憶装置から読みだされるのはデータ部のみであるため、テキスト部が大きくなることによるオーバーヘッドはELFに比べ小さい。

(3) ELFについて、テキスト部の大きさが64KBのものに比べ、96KBのプログラム実行時には実行時間が3ms程度遅くなっている。これは、ファイルが64KB以上となり、ページ例外が発生したことによるオーバーヘッドが原因であると思われる。

5. おわりに

本稿では、OFF2Fの効果を検証する支援環境として構築した擬似NVメモリを用いてOFF2Fプログラムを実行し、有効性を示した。評価では、テキスト部の大きさが32KB、64KB、96KB、および128KBのプログラムについて、ELFとOFF2Fでのプログラムの実行時間を測定した。この結果、外部記憶装置からの読み出しの削減やページ例外処理の短縮により、OFF2Fの有効性を示した。

謝辞 本研究の一部は、JSPS KAKENHI 18K11244による。

参考文献

- [1] Eisenman, A., Gardner, D., AbdelRahman, I., et al.: Reducing DRAM Footprint with NVM in Facebook, Proceedings of the Thirteenth EuroSys Conference (EuroSys '18), Article No.42 (2018).
- [2] 谷口 秀夫: 揮発/不揮発メモリ混載環境を支援する仮想記憶機構向け実行ファイル形式: OFF2Fの提案, コンピュータシステム・シンポジウム論文集, Vol.2017, pp.35-40 (2017).
- [3] 高杉 頌, 佐藤 将也, 谷口 秀夫: 擬似不揮発性メモリを用いたOFF2Fプログラムの実行方式, 情報処理学会研究報告, vol.2019-OS-146, no.12, pp.1-7(2019).