

## 不揮発性メモリを利用した *Tender* における動作継続制御の基本評価 Basic Evaluation of Persistent Mechanism Using Non-volatile Memory on *Tender*

田中 雅大      山内 利宏      谷口 秀夫  
Masahiro Tanaka    Toshihiro Yamauchi    Hideo Taniguchi

### 1. はじめに

計算機の主記憶に用いられるメモリは揮発性メモリである。このため、*Tender* オペレーティングシステム (以降、*Tender*) では、揮発性の主記憶をあたかも不揮発性であるかのように見せるプレート機能[1]を実現している。この機能は、揮発性メモリ上の仮想記憶空間のデータを外部記憶装置上へ書き出し、計算機再起動時に保存したデータを仮想記憶空間上へ復元することにより、仮想記憶空間上のデータを永続化する。また、プレート機能を利用し、カーネル用とユーザ用の仮想記憶空間上に存在するすべてのデータを永続化することにより、OS の処理と AP の処理を永続化する動作継続制御[2]を実現している。

不揮発性メモリを利用するソフトウェアの技術が研究されている[3]。不揮発性メモリを主記憶に利用する場合、*Tender* では、プレート機能を使用せずとも、不揮発性メモリ上に残ったデータを利用して動作継続制御が可能となる。本稿では、不揮発性メモリを利用した *Tender* の動作継続制御の基本評価として、不揮発性メモリの主記憶をエミュレートした仮想計算機上での動作確認と起動処理の処理時間の評価結果を述べる。

## 2. *Tender* オペレーティングシステム

### 2.1 資源の分離と独立化

*Tender* では、OS が制御し管理する対象を資源と呼び、資源の分離と独立化を行っている。資源の種類ごとに、資源を管理する管理表と資源を操作するプログラムが存在する。資源は資源名と資源識別子によって識別される。

*Tender* のメモリ関係資源を図 1 に示し、説明する。資源「仮想空間」とは、特定のアドレス領域をもつ仮想的な空間であり、仮想アドレスから実アドレスへのアドレス変換表に相当する。資源「仮想領域」は、メモリイメージを仮想化した資源であり、実体は実メモリもしくは外部記憶装置上に存在する。資源「仮想ユーザ空間」は、資源「仮想領域」をユーザ用の資源「仮想空間」に貼り付けることで生成される。「貼り付ける」とは、仮想アドレスを実アドレスに対応付けすることである。資源「仮想カーネル空間」は、資源「仮想領域」をカーネル用の資源「仮想空間」に貼り付けることで生成される。

### 2.2 プレート機能

プレート機能[1]とは、OS が揮発性メモリ上の仮想記憶空間のデータを外部記憶装置上へ自動的に書き出すことにより、データを永続化する機能である。永続化の対象となる領域を「プレート」と呼ぶ。プレートは、仮想記憶空間上に存在することを基本とし、仮想記憶空間と外部記憶装置の入出力契機を OS が判断する。*Tender* では、プレ-

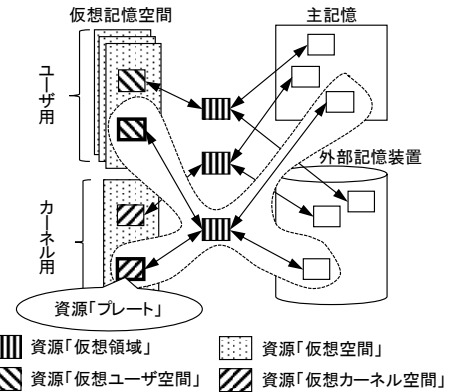


図 1 メモリ関係資源の様子とプレートの構成

ト機能を資源「プレート」として実現している。プレートの構成を図 1 に示す。1 つのプレートは、仮想カーネル空間、仮想ユーザ空間、仮想領域、実メモリ、および外部記憶装置上の領域からなる。

### 2.3 揮発性メモリを利用した動作継続制御

揮発性メモリを利用した動作継続制御[2]は、計算機停止前の処理を計算機再起動後に継続して実行する機能を持つ。揮発性メモリを利用した動作継続制御は、以下の 3 つの処理からなる。

- (1) 仮想記憶空間上のデータの永続化  
OS や AP が利用する仮想記憶空間上のデータをプレート機能によって永続化する。
- (2) プレートの書き出し  
プレート機能によって管理される仮想記憶空間上のデータを外部記憶装置上の領域に書き出す。
- (3) プレートの復元  
揮発性メモリを利用した動作継続制御の起動処理の流れを図 2(A) に示す。プレートの復元処理に必要なデータを生成後、外部記憶装置上にプレート管理表が存在する場合、プレート管理表を用いて、外部記憶装置上に保存されたデータから仮想記憶空間上にプレートを復元する。その後、復元したプロセスの動作継続を行う。

### 2.4 不揮発性メモリを利用した動作継続制御

不揮発性メモリを利用した動作継続制御の設計を文献[4]において述べた。不揮発性メモリを利用した動作継続制御は、不揮発性の主記憶上に残った計算機終了前のデータを利用して、計算機再起動後に処理を継続する。不揮発性メモリを利用した動作継続制御の起動処理の流れを図 2(B) に示す。計算機終了前に生成した資源の管理表といった OS の処理に必要なデータは、主記憶上に残ったままであるため、計算機再起動時に管理表の確保と初期化処理を実行しない。また、プレート機能を使用しないため、プレート管

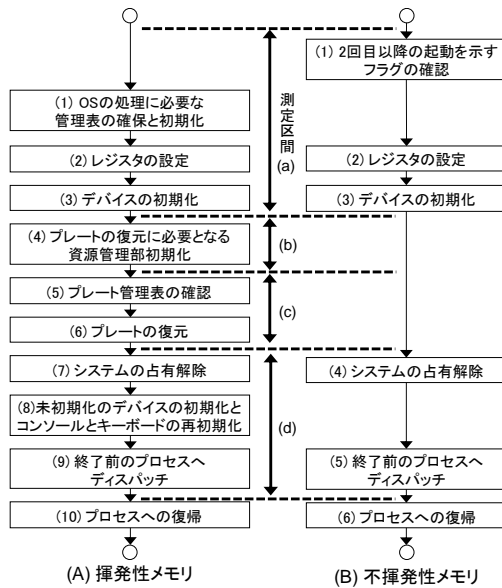


図 2 動作継続制御の起動処理の流れ

理表の有無ではなく、2 回目以降の起動を示すフラグを利用して計算機処理を継続できるか確認する。

終了処理時には、プロセッサ内部のキャッシュをフラッシュする。これは、更新されたデータがキャッシュ上に存在する場合、計算機終了時に消失してしまい、正常に処理が継続できなくなるためである。

### 3. 評価

#### 3.1 内容

不揮発性メモリを利用した動作継続制御の動作確認、および性能を明らかにするため、不揮発性メモリの主記憶をエミュレートした仮想計算機上で以下の評価を行った。

(1) AP 処理の継続の確認

(2) 再起動時の起動処理の処理時間

評価には、Intel Core i3-6100T (3.20GHz, 2 コア), メモリ 8192MB の実計算機を用いて、ゲスト OS のメモリを 256MB とし、**Tender** を起動した。(1)について、次の2つのプログラムを同時に走行させる。1つのプログラムは、クイックソートを行い、ソートが完了した後に共有領域に結果を書き込む。もう1つのプログラムは、共有領域を確認し、ソートの結果が書き出されていれば、結果を画面に出力する。前者は、処理の途中で終了処理を要求し、計算機の電源を切る。その後、計算機を再起動し、AP の処理が継続できるか確認を行った。

(2)について、起動処理の処理時間として、OS の処理開始直後からプロセスへの復帰の直前までの処理時間を測定した。測定区間を図 2 に示す。また、結果を比較するため、揮発性メモリを利用した動作継続制御の起動処理の処理時間を測定した。保存する計算機状態は、いずれも起動処理終了直後の状態であり、このとき復元されるプレートの数は、161 個である。

#### 3.2 結果

##### 3.2.1 AP 処理の継続の確認

実験の結果、計算機再起動後にソートの結果が正しく表示されることを確認した。このことから、計算機終了前と

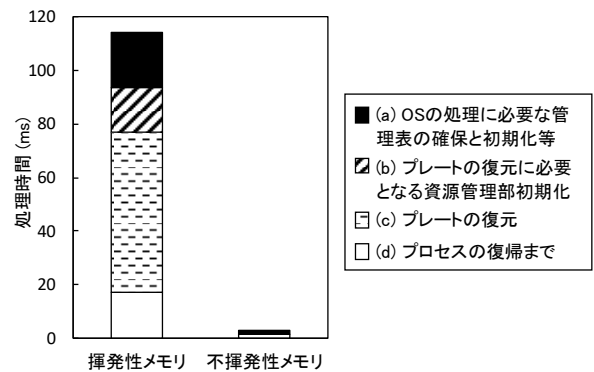


図 3 動作継続制御の起動処理の処理時間

計算機再起動後の間に AP の利用するデータに変化がなかったことが分かる。また、AP の処理が正常に再開し、終了したことから、不揮発性メモリ上に残ったデータを利用する場合でも、計算機処理を継続する機能を実現できたことが確認できた。

##### 3.2.2 再起動時の起動処理の処理時間

動作継続制御の起動処理の処理時間を図 3 に示す。プレートの復元処理の時間については、外部記憶装置との入出力に要する時間を除いた時間を示す。

不揮発性メモリを利用した動作継続制御は、計算機終了前の主記憶のデータを利用し、処理を継続する。このため、プレートの復元処理とこれに必要な資源管理部の初期化処理がない。さらに、OS の処理に必要な管理表の確保と初期化処理を行わないことにより、処理時間を  $111.72\text{ms} (= (114.42 - 2.70))$  短縮できた。

今回の評価で保存した計算機状態は、プレートの数が最も少ない起動処理終了直後の状態である。動作するプロセスの数が増えるにつれて、プレートの数も増えるため、プレートの復元処理の時間は長くなる。このため、動作するプロセスの数が多い計算機状態を復元する場合、揮発性メモリを利用したものと不揮発性メモリを利用したものの間で処理時間の差が大きくなると推察できる。

### 4. おわりに

不揮発性メモリを利用した **Tender** の動作継続制御の基本評価を述べた。不揮発性メモリを利用した動作継続制御は、不揮発性メモリ上に残ったデータを利用して計算機処理を永続化する。再起動時の起動処理の処理時間の評価として、不揮発性メモリ上のデータを利用することで、揮発性の主記憶を前提とした処理から処理時間を  $111.72\text{ms} (97.64\% (= 111.72 / 114.42 * 100))$  短縮できることを示した。

#### 参考文献

- [1] Yamauchi, T., Yamamoto, Y., Nagai, K., et al.: Plate: Persistent Memory Management for Nonvolatile Main Memory, Proceedings of 31st ACM Symposium on Applied Computing (SAC 2016), pp.1885-1892, ACM (2016).
- [2] 山本悠太, 田端利宏, 谷口秀夫: **Tender** の動作継続制御機能における入出力デバイスの扱い, 情報処理学会研究報告, 2008-OS-109, Vol.2008, No.77, pp.61-68 (2008).
- [3] Zheng, S., Hoseinzadeh, M. and Swanson, S.: Ziggurat: A Tiered File System for Non-Volatile Main Memories and Disks, Proceedings of the 17th USENIX Conference on File and Storage Technologies (FAST '19), pp.207-219 (2019).
- [4] 田中雅大, 山内利宏, 谷口秀夫: 不揮発性メモリを利用した **Tender** における動作継続制御の実現, 第 147 回システムソフトウェアとオペレーティング・システム研究会(2019). (発表予定)