

Tender の世代管理機能の実現

長井 健悟^{†1} 山本 悠太^{†1}
山内 利宏^{†1} 谷口 秀夫^{†1}

計算機の主記憶は揮発性であるため、計算機の電源切断により、主記憶上のデータは消失する。このため、我々は、*Tender* オペレーティングシステムにおいて、計算機状態を永続化する動作継続制御を提案した。動作継続制御は、仮想記憶空間を利用して主記憶上のデータを不揮発性記憶媒体へ永続化するプレート機能を利用し、計算機停止前に行っていた処理の継続に必要なデータを永続化する。しかし、プレート機能により永続化可能なデータの状態は1つだけである。そこで、複数の計算機状態を保存可能にする世代管理機能を提案した。世代管理機能は、計算機状態単位で全プレートの永続データの複製を複数世代分保存、復元可能にする。本稿では、世代管理機能の実現について説明し、その処理時間とディスク占有量を評価した結果を報告する。

Realization and Evaluation of the Generation Management Function on *Tender* Operating System

KENGO NAGAI,^{†1} YUTA YAMAMOTO,^{†1}
TOSHIHIRO YAMAUCHI^{†1} and HIDEO TANIGUCHI^{†1}

Data on main memory is no longer available when computer is turned off, because main memory is volatile. Therefore, we proposed persistent mechanism for computer processing on *Tender* operating system. Persistent mechanism uses "plate" function to make persistent data which needed to continue processing before the shutdown. Plate manages a persistent data on virtual memory space. However, persistent mechanism can maintain only one state of persisted computer processing. Thus, we proposed generation management function which enable to maintain two or more states of the computer processing. It permits plural generations of all persisted plate data to be maintained in each state of the compute processing. In this paper, we explain realization of the generation management function, and furthermore, reports on the evaluation results at the processing time and amount of occupation on disk.

1. はじめに

多くの計算機は、揮発性メモリを主記憶に用いる。このため、既存のオペレーティングシステム（以降、OS と略す）において、停電や電源ケーブルの接触不良などにより、計算機への電力供給が停止した場合、計算機が異常終了し、個々のアプリケーション（以降、AP と略す）は、データを永続化し損ねる可能性がある。また、既存 OS は、AP が利用する一部のデータのみを永続化し、AP の状態は永続化しない。このため、計算機を再起動しても AP の処理途中状態は復元されない。

計算機処理の中断と再開を可能にする機能として、Windows の休止機能や Linux の swsusp, TuxOnIce¹⁾ のようなハイバネーション機能がある。ハイバネーション機能は、主記憶上の全てのデータを外部記憶装置に書き出し、計算機再起動後に保存したデータを主記憶に読み込む。これにより、計算機処理中断時の状態から処理を再開できる。しかし、ハイバネーション機能は、利用者が実行契機を与える必要があるため、予見できる停止には有効であるが、予見できない緊急停止には対応できない。

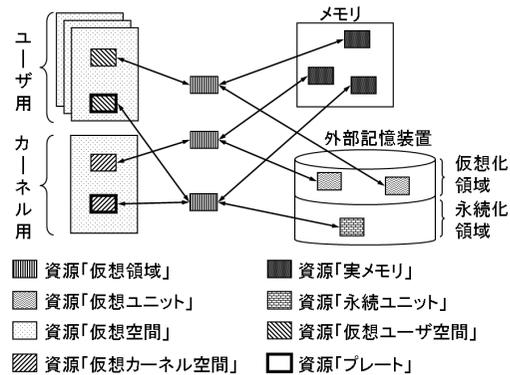
また、Membrane²⁾ はファイルシステムエラー発生時に、チェックポイントからの回復を行うことにより、AP がファイルシステムエラーの発生を意識せずに、動作を継続することを可能にする。しかし、電源の供給停止による計算機の停止や OS のバグによる機能停止には対応できない。また、チェックポイントを複数持たないため、直近の状態への回復のみしか行えない。

Tender オペレーティングシステム³⁾ (以降、*Tender* と略す) は、仮想記憶機構を利用した主記憶上のデータを永続化するプレート機能⁴⁾ を持ち、プレート機能を用いて主記憶上の計算機の処理状態 (以降、計算機状態と呼ぶ) を書き出し、復元できる。しかし、プレート機能により管理される永続データは、1つのプレートに対して1つであるため、動作継続制御⁵⁾ において、最新の計算機状態のみが復元可能である。このため、ウィルスやワームなどの有害なプログラムの感染や AP の重大なバグの発生後、プレート書き出しを行った場合、異常のある計算機状態のみが保存される。このとき、プレートの復元により計算機状態を復元しても、正常な計算機状態を復元できない。

そこで、文献 6) では、永続データの世代管理機能を提案し、その設計について述べた。世

^{†1} 岡山大学大学院自然科学研究科

Graduate School of Natural Science and Technology, Okayama University



代管理機能は、1つのプレートに対応する永続データを複数保存可能にする。これにより、複数時点の計算機状態の保存を実現する。

ファイルの世代管理に関する既存技術として、Mac OS X の Time Machine⁷⁾ や rsnapshot⁸⁾ がある。世代管理機能はこれらの技術と同様に、ハードリンクを用いて、ファイルの世代管理を行う。ただし、世代管理機能はハードリンクを用いることで、世代復元を複写レスで行う。これに対し、Time Machine や rsnapshot はバックアップファイルの容量を削減するためにハードリンクを用いるものの、復元には複写を必要とする。

本稿では、世代管理機能の実現について説明し、その処理時間とディスク占有量を評価した結果を報告する。

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して細分化し、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。

2.2 メモリ関連資源

Tender におけるメモリ関連資源の関係について、図 1 に示し、以下で説明する。

資源「仮想領域」は、メモリ資源をイメージ化した資源である。メモリ上のデータは、資源「実メモリ」上、もしくは外部記憶装置上の領域に存在する。資源「実メモリ」は主記憶

上の領域を表す資源である。資源「仮想ユニット」は、仮想化領域の管理単位である。仮想化領域とは、主記憶上のデータを一時的に保存するための領域であり、既存 OS のスワップ領域を利用して実現している。資源「永続ユニット」は、永続化領域の管理単位である。永続化領域とは、仮想記憶空間上のデータを永続化するために利用する領域であり、既存 OS のファイルシステム領域を利用して実現している。

資源「仮想空間」は、特定のアドレス領域を持つ仮想的な空間である。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「仮想ユーザ空間」は、ユーザ用の仮想空間に存在するデータを表す。資源「仮想ユーザ空間」は、OS や AP が資源「仮想領域」で管理されるデータにアクセスするために、資源「仮想領域」をユーザ用の資源「仮想空間」に貼り付けて生成する。「貼り付ける」とは、仮想アドレスを実アドレスに対応付けることである。一方、仮想アドレスと実アドレスの対応付け解除を「剥がし」と呼ぶ。資源「仮想カーネル空間」は、カーネル用の仮想空間に存在するデータを表し、資源「仮想領域」をカーネル用の資源「仮想空間」に貼り付けて生成する。

2.3 プレート機能

Tender では、永続化の対象となる領域を「プレート」と呼ぶ。プレート機能とは、OS が自動的に仮想記憶空間上に存在する上のデータを外部記憶装置上の永続化領域へ書き出すことにより、データを永続化する機能である。Tender では、プレート機能を資源「プレート」として実現している。プレート機能は以下の機能を持つ。

(機能 1) プレート書き出し機能

プレートを主記憶上から外部記憶装置上の永続化領域にファイルとして書き出すことにより、データを永続化する。

(機能 2) プレート復元機能

外部記憶装置上の永続化領域に存在するファイルを主記憶上に読み込むことにより、データを復元する。

AP は、必要に応じてプレートを自分の仮想記憶空間にマッピングして利用する。また、AP はプレート上のデータの書き出しを OS に依頼できる。プレート機能は、計算機の電源投入時にプレートを復元する。これにより、プレートは、計算機の電源再投入後でも、計算機の電源切断前と同じ仮想記憶空間のアドレス領域に存在し続ける性質を持つ。

2.4 動作継続制御

動作継続制御は、プレート機能を利用して、プレートを復元することで、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。動作継続制御の機能を以下に示す。

(1) 仮想記憶空間上のデータの揮発性化

プレート機能により、OS や AP が利用する仮想記憶空間上のデータをプレート化することで、永続化対象とし、プレート化されたデータを外部記憶装置上の領域に書き出す。

(2) 仮想記憶空間上のデータの復元

OS の開始処理において、プレート管理部初期化時、外部記憶装置上にプレート管理表が存在している場合、プレート管理表に格納されている情報をもとに、外部記憶装置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し、プロセスの動作継続を行う。

3. 世代管理機能

3.1 世代管理機能の基本方式

世代管理機能を、以下の 4 つの機能により構成する。

(機能 1) 世代システム

世代システムとは、ファイルシステムを利用し、全てのプレートの永続データの複製を複数世代分管理する機能である。具体的には、外部記憶装置上に永続化領域とは別にプレートの永続データを保存する領域（以降、世代保存領域と名づける）を確保し、各永続ユニットごとに対応付けられるファイルを複製する。また、世代情報は、世代管理表に記録する。

(機能 2) 世代保存機能

世代保存機能の処理の流れを図 2 に示す。世代保存機能とは、現在の計算機状態を 1 つの世代として保存する機能であり、計算機状態を主記憶上から永続化領域上に保存し（図 2(1)）、これを新たに確保した世代保存領域に複製する（図 2(2)）。

(機能 3) 世代復元機能

世代復元機能の処理の流れを図 3 に示す。世代復元機能とは、指定した世代の計算機状態を復元する機能であり、世代保存領域上に保存してある計算機状態を永続化領域に複製し（図 3(1)）、プレート復元を行うことにより主記憶上に読み込む（図 3(2)）。

(機能 4) 世代削除機能

世代削除機能とは、指定した世代の計算機状態を削除する機能であり、世代保存領域上に保存してある計算機状態を削除する。

3.2 資源「システム」

3.2.1 永続データ書き出しにおける要件

世代保存において、全プレート書き出しにより、永続化したデータを用いて計算機処理を正常に継続するための要件を以下に示す。

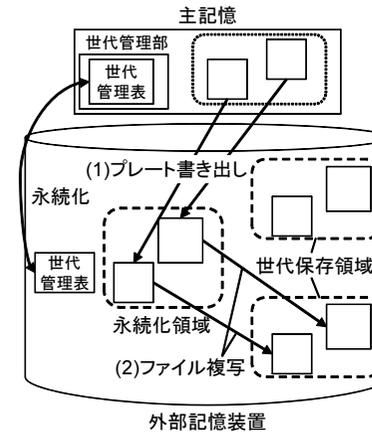


図 2 世代保存機能の処理の流れ

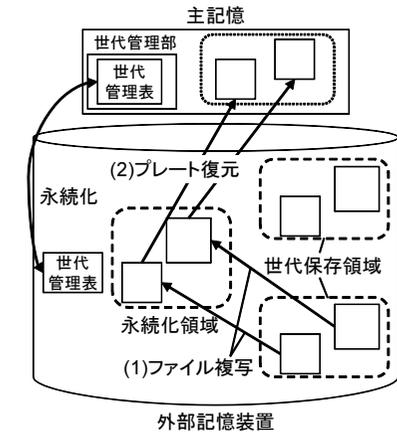


図 3 世代復元機能の処理の流れ

(要件 1) データ間の整合性を保証すること

(要件 2) 実入出力終了待ちのプロセスが存在しないこと

(要件 3) ファイルシステムのメタデータを操作中でないこと

3.2.2 基本方式

3.2.1 項で述べた要件を満たし、計算機処理を正常に継続するために実現した資源「システム」について述べる。Tender において、プログラムを実行する際に利用するハードウェア資源の構成を表す資源として資源「システム」を実現する。資源「システム」は、プロセッサ、外部記憶装置、通信用デバイスなどのハードウェア資源で構成される。ただし、本論文では、CPU と外部記憶装置から構成されるシステムを 1 つのみ作成でき、Myrinet や Ethernet 用の通信用デバイスは管理対象外とする。

資源「システム」は、システム管理処理部により管理される。システム管理と他の資源管理処理部との関係を図 4 に示す。図 4 において、システム管理処理部は、演算処理管理部にプリエンプションの設定を依頼し、割り込みコントローラを操作して、割り込みマスクの設定を行うことにより、CPU の利用状態を制御する。また、システム管理処理部は、世代管理部と UNIX ファイル制御を介して、外部記憶装置への入出力処理を行う。システム管理処理部は、資源「システム」を構成するハードウェア資源の利用を制御する機能を持つ。具体的には、2 つの機能を提供する。

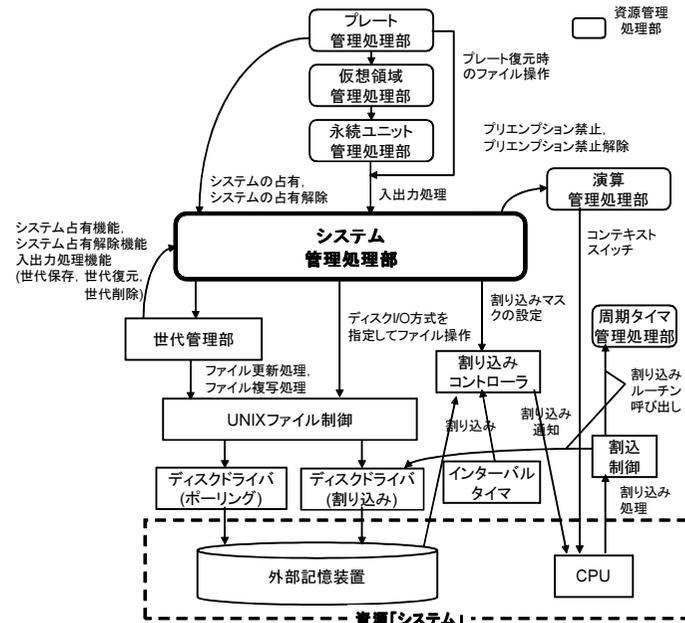


図4 システム管理と他の資源管理との関係

(機能1) ハードウェア資源への入出力処理機能

システム管理処理部は、資源「システム」を構成するハードウェア資源への入出力処理機能を提供する。各資源管理処理部は、この機能を利用して入出力処理を行う。入出力処理機能の一つとして、ファイル操作を提供する。

(機能2) ハードウェア資源の占有制御機能

システム管理処理部は、資源「システム」を構成するハードウェア資源の占有制御機能を提供する。システムの占有を要求された場合、システム管理処理部は、システム占有解除まで他のプロセスやデバイスの割り込みルーチンにより、システムが管理するハードウェア資源を利用されないように制御する。

3.3 世代システム

3.3.1 世代保存領域の実現方式

Tender が利用するファイルシステムのディレクトリ構成の例を図5に示す。永続化領域、

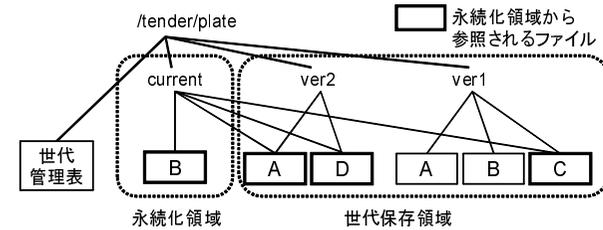


図5 ハードリンクによるプレートの状態の保存

世代保存領域、および世代管理表は、ファイルシステム上の/tender/plate 以下の領域に存在するものとする。プレートの永続データを保存する永続化領域として、/tender/plate/current 以下の領域を用いる。世代保存領域として、/tender/plate/XXX 以下の領域を用いる。ここで、XXX は、世代名とする。世代管理表は、/tender/plate 直下に保存する。世代保存時には、世代保存領域用のディレクトリを生成し、プレートの永続データを保存したファイルをそのディレクトリに複写する。世代復元時には、世代保存領域用のディレクトリから永続化領域上へプレートの永続データを保存したファイルをそのディレクトリに複写する。世代削除時には、世代保存領域用のディレクトリとその配下のファイルを削除する。

3.3.2 コピーオンライトによるファイル複写

世代保存処理と世代復元処理では、永続化領域と世代保存領域の間でファイル複写が行われる。世代保存処理や世代復元処理の際、毎回全てのファイルを複写することを防ぐため、ハードリンクを利用したコピーオンライトを用いて、ファイル複写処理を行うことにより、ファイル複写データ量を削減し、世代保存と世代復元に要する処理時間を短縮する。

具体的には、図5に示す通り、プレートの永続データを保存したファイルのハードリンクを永続化領域用ディレクトリと世代保存領域用ディレクトリの両方に生成し、永続化領域と世代保存領域間でファイルを共有する。永続化領域上の永続データを削除する際や世代削除により世代保存領域上の永続データを削除する場合、永続化領域用ディレクトリ、もしくは世代保存領域用ディレクトリに存在するプレートの永続データを保存したファイルのハードリンクを削除する。これにより、他の世代保存領域とファイルを共有している場合でも、ファイルは削除されない。

ファイルを更新する場合において、前回の世代保存、または前回の世代復元から初めて更新される永続データに対し、ファイル生成を行う必要がある。ファイル更新の際、ファイル

のリンク数を確認し、リンク数が2以上のファイルを初めて更新されるデータと判定する。この場合、ファイル生成を行った後に、更新されたメモリ上のデータを書き出す。

また、プレート管理処理部が世代管理を利用してプレートに対応するファイルを更新する際の処理流れを図4を基に、以下に説明する。

- (1) プレート管理処理部が、仮想領域管理処理部、永続ユニット管理処理部、システム管理処理部を経由して、世代管理にプレートの更新部分の書き出しを要求する。
- (2) 世代管理が、更新対象ファイルのハードリンクを削除する。
- (3) 世代管理が、更新対象ファイルとして空ファイルを生成する。
- (4) 世代管理が、システム管理処理部、永続ユニット管理処理部、仮想領域管理処理部を経由して、プレート管理処理部にプレート全体の書き出しが必要であることを処理(1)の戻り値として伝達する。
- (5) プレート管理処理部が、仮想領域管理処理部、永続ユニット管理処理部、システム管理処理部を経由して、世代管理にプレート全体の書き出しを要求する。
- (6) 世代管理が、更新対象ファイルにプレート全体を書き出す。

これにより、プレート管理処理部はファイル実体を意識せず、プレートを書き出せる。

4. 評価

4.1 評価内容

コピーオンライトによる処理時間の短縮とディスク占有量の削減の効果を示すため、世代管理機能利用時の処理時間とディスク占有量を評価した。具体的には、世代管理機能におけるファイル複写処理をコピーオンライトにより行う方式(以降、コピーオンライト方式と呼ぶ)と世代管理機能におけるファイル複写処理をファイル間複写により行う方式(以降、単純コピー方式と呼ぶ)を比較した。処理時間の評価では、3.1節で述べた世代管理機能の持つ各機能を利用する際の処理時間を評価した。また、ディスク占有量の評価では、世代管理機能の利用時に保存される世代がディスクを占有する容量を評価した。測定には計算機1台(CPU: Celeron D 2.8GHz, メモリ: 768MB, ハードディスク: 7200rpm Ultra ATA/100 HDD, OS: *Tender*)を用いた。

4.2 処理時間

4.2.1 世代システム

コピーオンライト方式では、ハードリンクを利用することにより、ファイル複写処理時間の短縮が期待できる。しかし、コピーオンライト方式により複写されたファイルは更新時に

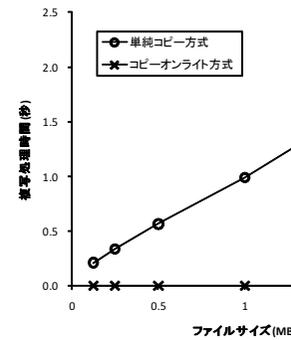


図6 ファイル複写処理時間

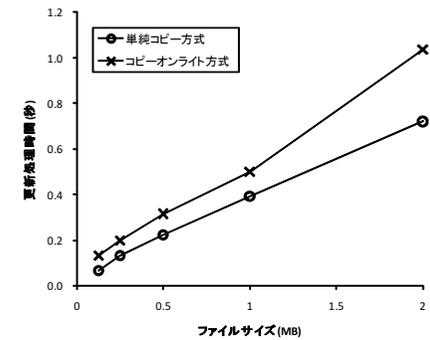


図7 ファイル更新処理時間

ファイルを生成する必要があるため、ファイル更新処理時間は長大化してしまう。そこで、両方式において、単一ファイルに対するファイル複写処理時間とファイル更新処理時間を比較した。具体的には、ファイルサイズを0.125MB, 0.25MB, 0.5MB, 1MB, 2MBと変化した場合における各方式のファイル複写処理時間とファイル更新処理時間を測定した。ファイル複写処理時間の測定結果を図6に、ファイル更新処理時間の測定結果を図7に示す。

図6より、単純コピー方式において、ファイル複写処理時間はファイルサイズに比例して増加することが分かる。一方、コピーオンライト方式においても、ファイル複写処理時間はファイルサイズに比例して増加するものの、増加率は非常に小さく、ファイル複写処理時間を大きく短縮できる。具体的には、ファイルサイズが2MBの場合、単純コピー方式では、2.0秒であるが、ハードリンク方式では、0.001秒と非常に短い。これは、コピーオンライト方式ではファイル実体を複写せず、ファイル実体へのリンクのみを作成するためである。

また、図7より以下のことが分かる。

- (1) ファイル更新処理時間はファイルサイズに比例する。
- (2) ファイルサイズによらず、コピーオンライト方式の方がファイル更新処理時間は長い。具体的には、ファイルサイズが2MBの場合、単純コピー方式では、0.7秒であるが、ハードリンク方式では、1.0秒である。これは、コピーオンライト方式ではファイル更新時に、ファイル生成処理が発生するためである。また、ファイル生成処理は、世代管理機能によりファイルが複写された後、1度目の更新処理でのみ発生する。このため、以降のファイル更新処理時間は、単純コピー方式と同じとなる。

以上のことより、コピーオンライト方式は単純コピー方式に比べ、ファイル複写処理時間は短縮できるものの、ファイル更新処理時間は長大化してしまうことが分かる。しかし、ファイル複写処理時間とファイル更新処理時間の合計時間では、コピーオンライト方式が単純コピー方式よりも短い。具体的には、ファイルサイズが2MBの場合、単純コピー方式では、2.7秒、ハードリンク方式では、1.0秒となる。したがって、コピーオンライト方式による処理時間の短縮効果は大きいといえる。

4.2.2 世代保存機能

仮想ユーザ空間のサイズ S と個数 N を変化させた場合において、世代保存処理に要する時間を測定した。ただし、ハードリンク方式では、初回の世代保存処理に要する時間（以降、新規世代保存処理時間と呼ぶ）に比べ、2回目以降の世代保存処理に要する時間（以降、差分世代保存処理時間と呼ぶ）は長大化する。これは、世代保存処理において、計算機状態を永続化領域上のファイルへ書き出す際、前回の世代保存以降に更新されたプレートを書き出すためである。このため、世代保存処理を2回行い、それぞれの世代保存処理に要する時間を測定した。測定手順を以下に示す。

- (1) サイズ S の仮想ユーザ空間を N 個生成する。
- (2) 世代 g_1 (新規世代) を世代保存する。
- (3) 世代 g_2 (差分世代) を世代保存する。

また、測定条件を以下に示す。

(条件1)仮想ユーザ空間のサイズを変化させた場合

N を8とし、 S を0MB, 0.125MB, 0.25MB, 0.5MB, 1MBと変化させた。

(条件2)仮想ユーザ空間の合計サイズを固定させた場合

N を1, 2, 4, 8と変化させた。また、 $S = 2MB/N$ とし、仮想ユーザ空間の合計サイズを固定した。

測定結果を図8~図11に示す。図8より、両方式において、生成する仮想ユーザ空間の合計サイズが多くなるにつれて処理時間が増加することが分かる。しかし、コピーオンライト方式の処理時間の増加率は非常に小さく、生成する仮想ユーザ空間の合計サイズの影響を受けにくい。これは、コピーオンライトによって、ファイル複写時にファイルの実体をコピーしないためである。また、図9より、仮想ユーザ空間数が多くなるにつれて処理時間が増加することが分かる。しかし、仮想ユーザ空間の合計サイズの増加による処理時間の増加率は、仮想ユーザ空間数の増加による処理時間の増加率よりも大きい。これは、仮想ユーザ空間の合計サイズが処理時間に大きな影響を与えることを示す。

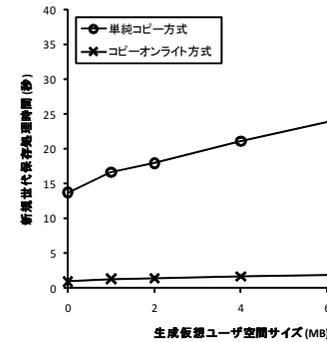


図8 Nが8の場合の新規世代保存処理時間

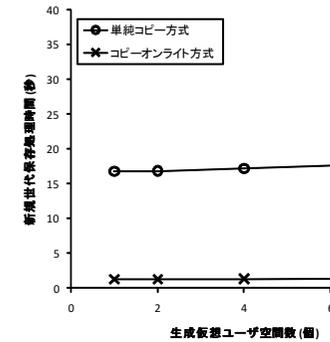


図9 $S \times N$ が2MBの場合の新規世代保存処理時間

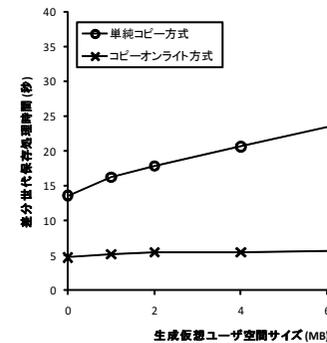


図10 Nが8の場合の差分世代保存処理時間

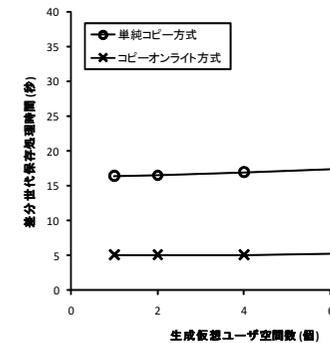


図11 $S \times N$ が2MBの場合の差分世代保存処理時間

図8と図10、および図9と図11を比較すると、単純コピー方式では、新規世代保存時と差分世代保存時で処理時間に大差がないのに対して、コピーオンライト方式では、仮想ユーザ空間のサイズや個数に関わらず、新規世代保存時は差分世代保存時よりも処理時間が約4秒増加していることが分かる。これは、コピーオンライト方式では、差分世代保存時に複数のハードリンクを持つファイルの更新処理が発生するためである。新規世代保存時と差分世代保存時の間で、一部のプレートは更新されるため、更新されたプレートの書き出しが発生し、ファイルの更新処理が発生する。この際に更新されるプレートは、タイマ割り込みによ

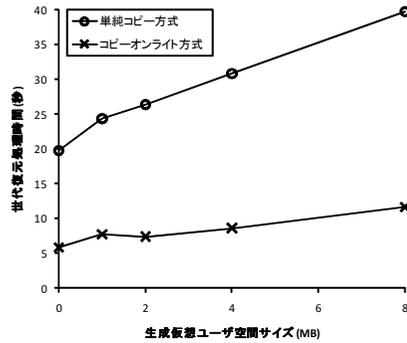


図 12 N が 8 の場合の世代復元処理時間

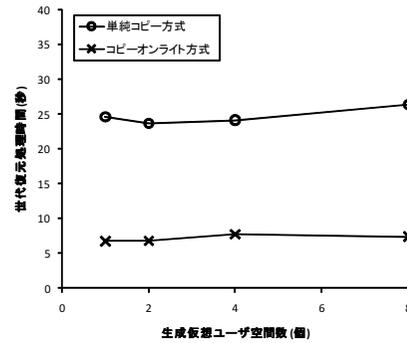


図 13 S x N が 2MB の場合の世代復元処理時間

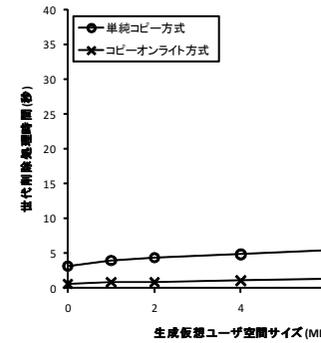


図 14 N が 8 の場合の世代削除処理時間

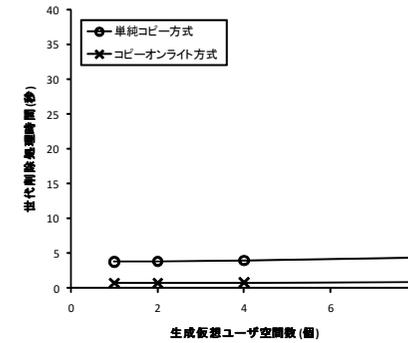


図 15 S x N が 2MB の場合の世代削除処理時間

り状態が変更されるもの、カーネルの動作により状態が変更されるもの、世代保存処理を行うプロセスの生成により生成されるものがある。

4.2.3 世代復元機能

仮想ユーザ空間のサイズ S と個数 N を変化した場合において、世代復元処理に要する時間を測定した。測定手順を以下に示す。なお測定条件は 4.2.2 項と同様である。

- (1) サイズ S の仮想ユーザ空間を N 個生成する。
- (2) 世代 g1(新規世代) を世代保存する。
- (3) 世代 g1(新規世代) を世代復元する。

測定結果を図 12, 図 13 に示す。図 12, 図 13 より、単純コピー方式はコピーオンライト方式に比べ、仮想ユーザ空間の合計サイズの影響を受けやすいこと、および両方式の処理時間は仮想ユーザ空間の数よりも、仮想ユーザ空間の合計サイズに大きく影響を受けることが分かる。この傾向は 4.2.2 項の世代保存処理と同じである。ただし、世代復元処理は世代保存処理に比べ、全体的に処理時間が増加している。これは、世代保存において、永続化領域への永続データの書き出し処理が更新部分のみを書き出すことで高速化されているためである。一方、世代復元において、永続化領域からの永続データの読み込み処理は全データを読み込む必要があるため、高速化できない。

4.2.4 世代削除機能

仮想ユーザ空間のサイズ S と個数 N を変化した場合において、世代削除処理に要する時間を測定した。測定手順を以下に示す。なお測定条件は 4.2.2 項と同様である。

- (1) サイズ S の仮想ユーザ空間を N 個生成する。
- (2) 世代 g1(新規世代) を世代保存する。
- (3) 世代 g1(新規世代) を世代削除する。

測定結果を図 14, 図 15 に示す。図 14, 図 15 より、単純コピー方式はコピーオンライト方式に比べ、仮想ユーザ空間の合計サイズの影響を受けやすいこと、および両方式の処理時間は仮想ユーザ空間の数よりも、仮想ユーザ空間の合計サイズに大きく影響を受けることが分かる。この傾向は 4.2.2 項の世代保存処理と同じである。ただし、世代削除処理は、世代保存処理や世代復元処理に比べ、全体的に処理時間が減少している。これは、ファイルの削除処理に要する処理時間は、ファイルの読み書き処理に比べて短いためである。

4.3 ディスク占有量

世代保存を行った場合において、各世代の保存時に、保存されている各世代と現在の世代(永続化領域上のファイル群)のディスク占有量の合計を測定した。測定手順を以下に示す。

- (1) 世代 g1(新規世代) を世代保存する。
- (2) 以下の処理を n を 2 から 5 として、繰り返す。
世代 g“n” (差分世代) を世代保存する。

測定結果を図 16 に示す。4.2.2 項で述べた通り、世代 g1 保存直後に世代 g2 を保存する場合において、g1 保存時と g2 保存時では、一部のプレートの状態が変化している。このため、g1 と g2 に含まれる一部のファイルはハードリンクにより共有できないため、その分、ディスク占有量が増加する。したがって、世代を連続で保存する場合において、コピーオン

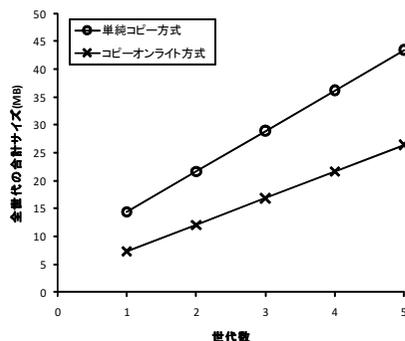


図 16 世代のディスク占有量

ライト方式でもディスク占有量が増加する。5回目の世代保存時において、ハードリンク方式と単純コピー方式のディスク占有量は43MBと26MBであり、ハードリンク方式のディスク占有量は単純コピー方式の約60%となる。

また、コピーオンライト方式でのディスク占有量は、更新されるプレートと更新されないプレートのそれぞれの合計サイズに依存する。例として、第1回目の世代保存時から第5回目の世代保存時まで更新されないプレートがさらに10MB存在した場合を仮定する。第5回目の世代保存時において、単純コピー方式では図16の例よりも、各世代(第1~5世代と現在の世代)の容量が10MB増加し、全世代のディスク占有量は $43 + 10 \times 6 = 103$ (MB)となるのに対し、ハードリンク方式では新規世代のディスク占有量が10MB増加するのみである。このため、全世代のディスク占有量は $26 + 10 = 36$ (MB)となる。したがって、この場合において、ハードリンク方式ではディスク占有量を単純コピー方式の約35%に削減できる。

以上のことから、ハードリンク方式のディスク占有量の削減効果は大きいといえる。

5. おわりに

*Tender*の動作継続制御において、永続データを複数世代保存可能にする世代管理機能について述べた。また、世代ごとの永続データを管理する世代システムについて述べた。世代システムにおけるファイル複写はコピーオンライトにより行われる。また、コピーオンライト方式と単純コピー方式の比較により、コピーオンライトによる世代管理機能の処理時間

の短縮とディスク占有量の削減の効果を示した。世代システムの評価において、1つのファイルを複写し、更新する場合において、処理時間は単純コピー方式よりも小さいことを示した。具体的にはファイルサイズが2MBであるファイルの複写処理時間と更新処理時間の合計処理時間は、単純コピー方式では、2.7秒、ハードリンク方式では、1.0秒となる。また、世代保存機能、世代復元機能、および世代削除機能の評価では、各機能利用時の処理時間が単純コピー方式の処理時間よりも少ないことを示した。各機能利用時の処理時間は仮想ユーザ空間数ではなく、仮想ユーザ空間の合計サイズの影響を強く受ける。さらに、ディスク占有量の評価では、連続で世代保存を5回行った場合、コピーオンライト方式のディスク占有量は、単純コピー方式の約60%であることを示した。また、この場合において、第1回目の世代保存時から第5回目の世代保存時まで更新されないプレートがさらに10MB存在した場合、コピーオンライト方式のディスク占有量は、単純コピー方式の約35%であり、ディスク占有量の削減効果が大きいことを示した。

残された課題として、資源「システム」によるプロセッサと外部記憶装置以外のハードウェアの管理と占有方法の検討、および他世代とハードリンクを持たない独立した世代の生成機能の検討がある。

参考文献

- 1) Cunningham, N.: TuxOnIce. <http://www.tuxonice.net/>
- 2) Sundararaman, S., Subramanian, S., Rajimwale, A., Arpaci-Dusseau, A. C., Arpaci-Dusseau, R. H. and Swift, M. M.: Membrane: Operating System Support for Restartable File Systems, *In Proceedings of the 8th USENIX Conference on File and Storage Technologies*, pp.281-294 (2010).
- 3) 谷口秀夫, 青木義則, 後藤真孝, 村上大介, 田端利宏: 資源の独立化機構による *Tender* オペレーティングシステム, 情報処理学会論文誌, Vol.41, No.12, pp.3363-3374 (2000).
- 4) 的野 司, 田端利宏, 谷口秀夫: *Tender* におけるプレート機能を利用した資源の永続化機構の設計, 情報処理学会研究報告, 2003-OS-93, Vol.2003, No.42, pp.147-154 (2003).
- 5) 大本拓実, 田端利宏, 谷口秀夫: 仮想記憶空間上のデータ不揮発性化による計算機処理永続化方式の提案, 情報処理学会研究報告, 2007-OS-104, Vol.2007, No.10, pp.25-32 (2007).
- 6) 長井健悟, 山本悠太, 山内利宏, 谷口秀夫: *Tender* の世代管理機能の設計, 情報処理学会研究報告, Vol.2010-OS-114, No.14, pp.1-8, 2010.
- 7) アップル: Mac OS X Snow Leopard. <http://www.apple.com/jp/macosex/>
- 8) Rubel, M.: rsnnapshot. <http://rsnapshot.org/>