

Tenderにおける資源「実メモリ」の機能拡張

楠 恒輝† 山内 利宏‡ 谷口 秀夫‡
 †岡山大学工学部 ‡岡山大学大学院自然科学研究科

1 はじめに

計算機のコア数の増加により、NUMA 構成が登場している。Tender オペレーティングシステム（以降、Tender）では、実メモリを資源「実メモリ」として管理し、資源の独立化を図ることで、多くの処理を効率化している。また、資源「実メモリ」は、UMA 構成を前提とすることで処理の高速化を図っている [1]。本稿では、資源「実メモリ」の特徴を生かして NUMA 構成を扱える方式について述べる。

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、資源を分離し、独立化している。資源は、資源名と資源識別子によって識別される。資源識別子を図 1 に示す。資源生成時に付与される資源識別子は、資源の場所、種類、および同一種類内の通番を有する。Tender は、プログラム構造として、資源の種類ごとの管理表（以降、資源管理表）を個別に用意し、また資源の種類ごとに管理するプログラム（以降、プログラム部品）も個別に用意し共有プログラムを排除する。プログラム部品は、資源インタフェース制御を介して呼び出される。資源インタフェース制御で用いられる資源操作のインタフェースは、全ての資源において共通である。

2.2 メモリ関連資源

Tender のメモリ関連資源を図 2 に示す。資源「仮想領域」は、メモリイメージを仮想化した領域を管理する資源であり、実体は、実メモリもしくは外部記憶装置に存在する。資源「仮想領域」は、資源「実メモリ」を生成することで実メモリを確保する。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ることである。具体的には、当該の仮想アドレスに対応するアドレス変換表のエントリに、実アドレスまたは外部記憶装置のアドレスを設定する。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」に貼り付けることにより生成される。

2.3 資源「実メモリ」

資源「実メモリ」は、4 KB 単位で実メモリを管理する。資源識別子の通番は 16 ビット幅であり、通番と実

31	24	23	16	15	0
場所		種類		同一種類内の通番	

図 1 資源識別子

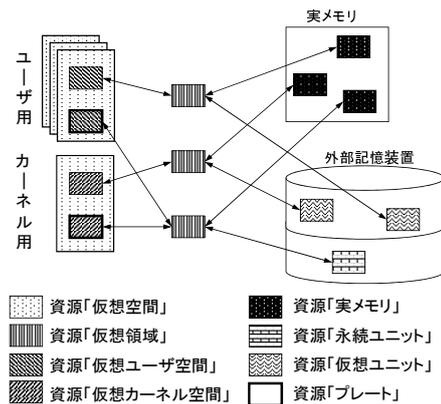


図 2 メモリ関連資源

アドレスを直接関係づけている。具体的には、通番を 12 ビット左シフトすることで実アドレスが得られる。これにより、資源「実メモリ」を利用する処理を高速化している。しかし、資源「実メモリ」で管理する実メモリは、最大 256 MB ($4 \text{ KB} \times 2^{16}$) に制限される。

3 資源「実メモリ」の機能拡張

3.1 背景

以下の背景に基づき、資源「実メモリ」を機能拡張する。

(背景 1) 計算機に搭載される実メモリ量の増加

近年、LSI の高集積化により、個人が所持している計算機にも 8 GB 以上の実メモリが搭載されている。

(背景 2) コア数の増加に伴う NUMA 構成の登場

UMA 構成では、コア数が増加するにつれてメモリアクセス時間が長くなる。これに対し、NUMA 構成では、多数のコアを複数のノードに分割し、ノード間を接続し、ノード内のメモリアクセスとノード間のメモリアクセスの速度が異なる構成としている。コア数の増加に伴い NUMA 構成が多くなっている。

3.2 設計

資源「実メモリ」の機能拡張では、資源操作のインタフェースを変更しない（要望）ことが望ましい。このため、実メモリを管理する単位（以降、管理単位）を 4 KB から大きくする（例えば、4 MB）案があるが、断片化によってメモリ不足が発生し易くなる。また、既存プログラムで不具合が発生する可能性が高い。

そこで、管理単位を 4 KB と 4 MB の二種類用意し、上記の問題に対処する。対処内容を以下に述べる。

Extension of Resource "Physical Memory" in Tender .
 Kouki Kusunoki†, Toshihiro Yamauchi‡, Hideo Taniguchi‡
 †Faculty of Engineering, Okayama University
 ‡Graduate School of Natural Science and Technology, Okayama University

- (1) 通番の上位1ビットを管理単位ビットとし、実アドレスに対応する情報を下位15ビットとする。
- (2) 管理単位ビットによって、管理単位を区別する。

資源「実メモリ」の新しい識別子の様子を図3に示す。管理単位ビットが0の場合は管理単位が4KBであり、管理単位ビットが1の場合は管理単位が4MBである。また、管理される実アドレス空間を図4に示す。管理単位ビットが0の場合は4KB × 2¹⁵の128MB空間を対象とし、実アドレスの0番地から0x00 07ff ffff (128MB)までを管理する。管理単位ビットが1の場合は4MB × 2¹⁵の128GB空間を対象とし、管理単位ビットが0の場合に管理される空間の後空間を管理する。つまり、管理単位ビットが1の場合に管理される実アドレスの範囲は、0x00 0800 0000から0x20 07ff ffffである。

管理単位ビットによって管理する実アドレスの範囲が異なるため、以下の方法で資源識別子と実アドレスを関係づける。

- (1) 管理単位ビットが0の場合の実アドレスは、通番の12ビット左シフトすることで得られる。
- (2) 管理単位ビットが1の場合の実アドレスは、資源識別子の通番を22ビット左シフトしたものに、管理単位ビットが0の場合に管理される空間の後空間の先頭アドレスを加えた和によって得られる。

これらにより、資源「実メモリ」を利用する処理の高速化を維持する。

3.3 期待される効果

本方式により期待される効果を示す。

- (1) 4KBで管理されている領域は、従来の資源「実メモリ」と同様である。このため、既存プログラムでの不具合の発生を防ぐことができる。
- (2) OSが管理できる実メモリ容量は、4KBと4MBの2つの管理単位で管理する空間の和となる。4KBと4MBのそれぞれで管理されている領域は各2¹⁵個ある。したがって、OSが管理できる実メモリ容量は128MB+128GBと拡大できる。
- (3) 4MBの管理単位を利用する場合に断片化を抑制するため、プロセスの使用メモリ量が増減しても既に確保している領域内に収まることを考慮し、実メモリの確保と解放の処理を工夫する。これにより、実メモリの確保と解放の回数の削減による高速化が期待できる。

4 課題

複数ノードを有する計算機の様子を図5に示す。ノード0の配置された実メモリは、実アドレスの若番地に配置されるとする。**Tender**は、共有型と個別型の2種類のマルチコア対応形式を実現している。

個別型は、各コア毎に、1つのコアを利用する**Tender**を独立して走行させる。資源管理表はコア毎に存在し、実メモリを空間分割して各**Tender**が管理する。カー

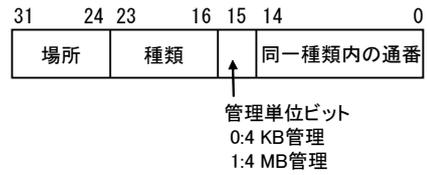


図3 資源「実メモリ」の資源識別子

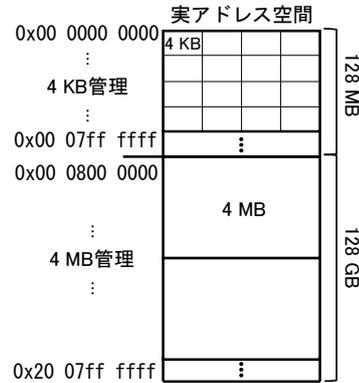


図4 実アドレス空間

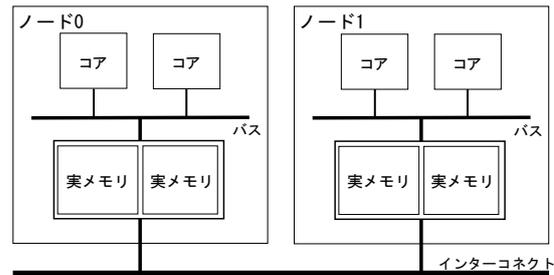


図5 複数ノードを有する計算機

ネル領域は実アドレスの若番地に配置し分割される。このため、ノード1のコアを利用する**Tender**はOS処理のメモリアクセスが遅くなり、改善が必要である。

共有型は、計算機全体で一つの**Tender**を走行させる。このため、ノード1のコアはノード1の実メモリをメモリアクセス速度の速いメモリとして扱うことができない課題がある。また、個別型と同様にカーネル領域は実アドレスの若番地に配置されるため、ノード1のコアで行う処理は遅くなる。

5 おわりに

Tenderの実メモリ管理機能の拡張について述べた。資源「実メモリ」の資源識別子の機能拡張により、管理できる実メモリ容量を、128MB+128GBにできる。残された課題として、提案方式の実装と評価、および複数ノードを持つ計算機への対応がある。

謝辞 本研究の一部は、共同研究(株式会社富士通研究所)による。

参考文献

- [1] 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).