

Tender 上の Web サーバにおける資源再利用機能の効果

佐伯 顕治[†] 田端 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

プロセスの生成処理は、メモリ空間の生成やプログラムの読み込み処理を必要とするため、オペレーティングシステムの処理の中でも負荷が大きい。このため、プロセスの生成と削除が頻繁する応用プログラム（以降 AP と略す）では、サービス処理性能が低下しやすい。そこで、文献[1]と[2]では、**Tender** オペレーティングシステム[3]において、プロセスを構成する資源の再利用を用いたプロセス生成処理とプロセス変身機能の高速化手法を示した。しかし、AP を用いた評価は行っていない。そこで、本稿は、AP として Web サーバを用いて資源再利用機能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 は、操作する対象を資源として分離し、独立化している。資源には、資源識別子と資源名を付与し、資源操作のインターフェースを統一している。また資源の分離と独立化を行うことで、資源の事前生成や保留が可能になり、資源の生成や削除に伴う処理を高速化できる。

2.2. プロセス変身機能

Tender はプロセスを構成する要素を変更し、プロセスの実行環境を変更する機能を備えている。この機能をプロセス変身機能[4]と呼び、以下の3つの機能で構成している。

(機能1) 実行プログラム変更機能

(機能2) 開始位置変更機能

(機能3) 動作空間変更機能

実行プログラム変更機能とは、プロセスとして実行されるプログラムを変更する機能である。開始位置変更機能とは、変更対象プロセスが次に走行する開始位置を変更する機能である。動作空間変更機能とは、プロセスが利用する仮想空間を別の仮想空間に変更する機能である。また、これらの3つの機能は、単独で利用するだけでなく、組み合わせて利用可能である。

The Effect of Recycling Resource in Web server on **Tender**

Kenji Saeki[†], Toshihiro Tabata[‡] and Hideo Taniguchi[‡]

[†]Faculty of Engineering, Okayama University

[‡]Graduate School of Natural Science and Technology, Okayama University

2.3. 資源再利用機能

資源再利用機能[1][3]とは、プロセス削除時に、プロセスを構成する資源(以降、プロセス構成資源と略す)の削除処理を保留し、プロセス生成時に、資源の生成処理の代わりに資源の再利用を行う機能である。再利用可能なプロセス構成資源は、プログラム、仮想領域、仮想空間、仮想ユーザ空間、およびワーク用仮想カーネル空間の5つである。**Tender** では、ワーク用仮想カーネル空間は再利用で常に効果があり、用いる領域の大きさも小さいので、常に再利用されている。また、仮想空間に仮想領域を貼り付けたまま、仮想空間を再利用することで、仮想空間とともに仮想ユーザ空間も再利用することができる。

3. 評価

3.1. Web サーバでのプロセス生成処理

Web サーバでのプロセス生成処理が行われる事例として、CGI プログラムの実行を伴う処理を取り上げる。評価に用いた Apache の例として、図1に CGI プログラム実行時の処理の流れを示す。CGI プログラムは、Web サーバがシェルを呼び出し、シェルが Perl を呼び出すことで実行されている。つまり一回の処理で2つのプロセスが生成される。CGI プログラムはアクセスカウンタ等で非常に多く用いられているため、これらのプロセス生成と削除処理が Web サーバの処理性能に大きな影響を与えている。

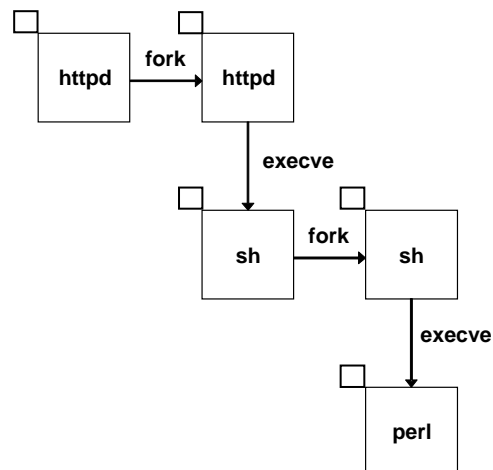


図1 CGI プログラム実行時の処理の流れ

本評価では、CGIプログラムとして、Perlによって記述されたアクセスカウンタを利用した。CGIプログラムはカウンタの値を1つ増やして、その値をWebサーバに送信し、ファイルに格納する。

Tenderでは、BSD/OS 互換システムコールインタフェース[5]を実装しており、その処理は**Tender**の内部インタフェースを用いて実現している。forkの処理は、カーネルプロセスの生成、プロセス変身機能(機能3)を用いた仮想空間の変更、プロセス変身機能(機能1)を用いた実行プログラムの変更、プロセス変身機能(機能2)を用いた開始位置の変更、および資源「演算」の割り当てによって実現している。execveの処理は、プロセス変身機能(機能1)によって実現している。

3.2. プロセス変身機能で再利用可能な資源

(機能1)では実行プログラムを変更するため、仮想領域、プログラムが再利用可能である。(機能2)ではdata部、bss部のロード処理を行うだけなので、再利用不可能である。(機能3)では空間を変更するため、仮想空間が再利用可能である。

forkは、プロセス変身機能(機能1)(機能2)(機能3)を用いて実現しているので、再利用可能な資源は仮想空間と仮想領域とプログラムである。仮想ユーザ空間は仮想空間と仮想領域を再利用するが、プロセス変身機能(機能3)を用いて変更した仮想空間に、プロセス変身機能(機能1)を用いて仮想領域を貼り付けるので、仮想ユーザ空間を再利用できない。execveは、プロセス変身機能(機能1)を用いて実現しているので、再利用可能な資源は仮想領域とプログラムである。

したがって、forkとexecveで再利用可能なプロセス構成資源は仮想空間、仮想領域、およびプログラムである。

3.3. 評価環境

サーバ計算機(CPU: Celeron 2.8GHz, OS: **Tender**) 1台とクライアント計算機(CPU: Pentium4 3.0GHz, OS: FreeBSD4.3-RELEASE) 1台を100Base-TXのEthernetで接続し、ベンチマークツールとしてApache Benchを利用して評価した。

3.4. 考察

Webサーバの応答時間を表1に示す。表1より、仮想空間の再利用によるWebサーバの応答

表1 Webサーバの応答時間

評価結果	再利用なし	再利用あり			
		仮想空間	仮想領域	プログラム	3資源
応答時間[ms]	14.37	10.24	13.11	14.08	9.02
応答時間の比	1.000	0.712	0.912	0.979	0.627

時間は28.8% $((1-0.712)*100)$ 向上する。仮想領域の再利用による応答時間は9.8% $((1-0.912)*100)$ 向上する。プログラムの再利用による応答時間は2.1% $((1-0.979)*100)$ 向上する。したがって、プログラムの再利用効果は小さい。これはプログラムを再利用する場合も、しない場合も、ディスクI/Oが発生しない条件で測定したためである。また、再利用可能な3つの資源を再利用する場合、応答時間は37.3% $((1-0.627)*100)$ 向上する。以上により、APにおいても資源再利用機能による性能向上効果があることを示した。

4. おわりに

forkとexecveで再利用可能なプロセス構成資源を明らかにし、Webサーバを利用して資源再利用機能の評価した。この結果、資源再利用機能を最大限に利用することで、最大37.3%の応答時間を短縮できることを示した。

残された課題としてテキスト部の内容を意識した仮想領域の再利用の実装と性能評価がある。

謝辞 本研究の一部は、科学研究費補助金 若手研究(B)(課題番号18700030)による。

参考文献

- [1]田端利宏, 谷口秀夫: プロセス構成資源の効率的な再利用を目指した資源管理方法の提案, 情報処理学会論文誌: コンピュータシステム, Vol.44, No. SIG10(ACS2), pp. 48-61(2003).
- [2]野村和孝, 田端利宏, 谷口秀夫: プロセス変身機能における資源再利用効果, 情報処理学会研究報告, 2004-OS-96, Vol.2004, No.21, pp.149-156(2004).
- [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]石井陽介, 谷口秀夫: 位置透過に利用可能な構成要素を用いたプロセスの変身機能, 情報処理学会論文誌, Vol.44, No.7, pp.1666-1679(2003).
- [5]田端利宏, 野口直樹, 中島耕太, 谷口秀夫: Webサーバを利用した**Tender**の資源「演算」の評価 - 複数プロセスの実行性能調整機能 -, 情報処理学会研究報告, 2002-OS-090, Vol.2002, No.60, pp.33-40(2002).