

プロセス資源の事前生成機構の提案

Proposal of Advance Creation Mechanism of Process Resources

田端 利宏†
Toshihiro Tabata

谷口 秀夫†
Hideo Taniguchi

1. まえがき

計算機ハードウェアの高性能化に伴い、処理負荷の変動は、プロセッサがアイドル時に処理できる量を大きくし、少量のプログラムを実行しているときの未使用メモリ量を大きくしている。したがって、これらの計算機資源を活用できれば、特別なコストを要することなく処理の高速化を実現でき、有益である。現在のOSを考えた場合に、OSは応用プログラムからの処理要求を受け取ってから処理を始める。しかし、さらなる処理の高速化を考えたときに、事前に行える処理をOSレベルで済ませておくことで処理をさらに高速化できることが考えられる。

本稿では、OSが将来必要となる資源を予測し、資源を事前に準備しておくことで、OS処理の高速化を実現する機構を提案する。具体的には、プロセッサのアイドル時間に事前処理を行い、その結果をOSの資源という形で保持する。この事前生成した資源をOS処理時に利用することで、処理の高速化を実現する。本稿ではプロセス資源を対象として提案機構について述べる。

2. 資源の事前生成機構

資源の事前生成機構とは、処理高速化のために事前に処理を実行し、その結果を資源として保持し、保持している資源を要求時に利用することで、処理を高速化する手法である。このためには、事前に処理すべき資源の予測、事前生成した資源の管理、資源の利用状況に合わせた資源生成と削除、及び事前生成処理の実行が必要となる。したがって、資源の事前生成機構は、大きく以下の4つの処理部分からなる。

- (1) 予測部
- (2) 資源管理部
- (3) 要求部
- (4) 事前生成部

これらの処理部分の関係を図1に示し、資源の事前生成機構について説明する。

予測部は、計算機の事前生成対象資源について、将来的に必要な最大資源量を予測する。

資源管理部は、事前生成した資源を管理する。

要求部は、予測部の予測結果、事前生成した資源量、及び現在のメモリ使用状況から、事前生成すべき資源とその量を分析し、必要に応じて事前生成処理部に対し処理要求を出す。

事前生成部は、要求部の要求に基づき、資源を事前生成または削除する。

また、予測に頼るだけでなく、計算機管理者が必要な資

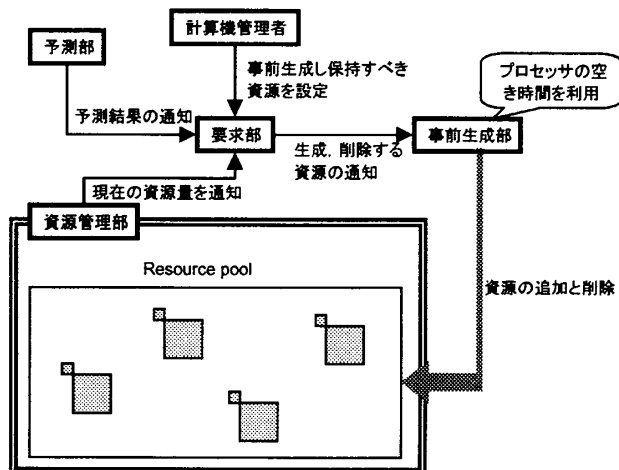


図1 事前生成機構の概要

源を指定する機能も備えておく。これにより、計算機管理者が必要と判断した資源を常に確保しできる。

3. 課題

3.1. 事前生成における課題

事前生成における検討課題として、事前生成の対象資源、事前生成する資源量、及び事前生成の契機がある。また、計算機の処理負荷や資源の利用状況が動的に変化するため、事前生成した資源を動的に管理する機構が必要である。以降で、各課題について述べる。

3.2. 資源の種類

OS処理に伴い生成し削除される資源が事前生成の対象となりうる。事前生成の効果の観点から、処理負荷が比較的大きいプロセス資源やメモリ関連資源が対象となる。ここでは、プロセス構成資源について議論する。

OSのプロセス生成処理に係る資源で事前生成の対象となりうる資源として、プロセス全体と、プロセスを構成する資源（プロセスの一部）が考えられる。プロセスはプログラムの実行単位であるため、実行頻度が多いプログラムについて事前生成しておく効果が大きいと推察できる。

3.2. 資源の予測

事前生成する資源の予測方法として、過去のプログラムの実行履歴に基づく方法と、システム管理者が設定した情報に基づく方法が考えられる。

過去のプログラムの実行履歴に基づく手法としては、著者の所属するグループではPOS (Program Oriented Schedule)

†岡山大学 大学院自然科学研究科, Graduate School of Natural Science and Technology, Okayama University

[1]を提案している。予測実現手法の一つとして、この手法を参考にして、OS 上でのプログラム実行予測機構を構築することが考えられる。

システム管理者が設定する情報としては、プログラム名とその最大事前生成数がある。これは、実行頻度の多いプログラム名とその実行頻度に見合った事前生成数が設定されることを意図している。また、プロセスを構成する資源を事前生成するための情報として、OS 上でのそのプログラムの最大同時実行プロセス数も考えられる。これらの情報はいずれもシステム管理者の予測に基づく情報であるが、その内容を信頼して利用することとする。

3.3. 処理の契機

事前生成は、OS 処理時間を短縮するためのものであるため、事前生成のための処理が本来の OS 処理に影響を与えることを避ける必要がある。したがって、基本的にはプロセスの空き時間を利用して資源の管理や事前生成を行うこととする。

事前生成の契機として、OS の起動時（サービスの開始前）と、プロセスの空き時間がある。

3.4. 資源の管理

事前生成した資源を保持するには、それに見合ったメモリを消費するため、事前生成のために資源を確保しすぎると、メモリが足りなくなり、本来の計算機処理がかえって遅くなってしまうことが考えられる。したがって、計算機のメモリ利用状況と事前生成した資源量を把握し、資源の事前生成と削除を適切に行う必要がある。

4. 実装と評価

4.1 実装

資源の事前生成の効果を検証するために、基本的な評価を行った。このために *Tender* [2]のカーネルの拡張部に資源の事前生成機構の一部を実装した。具体的には、資源管理部に事前生成したプロセス資源を管理する機能を追加した。また、この資源管理部に対し、事前生成した資源の有無を問い合わせるインタフェースを実現した。以下に実現したインタフェースを表1に示す。

Tender では、プロセスは独立した資源として管理されており、実行対象としない状態でプロセスの作り置きが可能である。そこで、今回はプロセス全体を事前生成することとした。

表1 実装した事前生成機構のインタフェース

関数名	機能
get_precreated_rsc(plateid)	指定されたプレートから事前生成されたプロセスの有無を確認する。もし事前生成されていれば、そのプロセス識別子を返す。
precreation(plateid, n)	プレート plateid からプロセスを n 個生成し、資源管理部の管理表にプレート識別子 plateid とプロセス識別子 pid の組を登録する。

表2 実験結果

	事前生成なし	事前生成あり
平均応答時間 (秒)	7.14	6.60

4.2. 評価

評価では、サーバ計算機 (PentiumIII 750MHz, OS: *Tender*) 上で Apache 1.3 を実行し、クライアント計算機 (Pentium4 2.0GHz, OS: BSD/OS 3.1) 上で Web クライアントプログラムを実行し、Apache の応答時間を測定した。サーバ計算機には、CGI プログラムを実行する Web ページを用意した。この CGI プログラムは簡単なアクセスカウンタプログラムである。また、Web クライアントプログラムは、同時に 10 個の子プロセスを生成し、シグナルを送ってこれらの子プロセスに同時に Web サーバにアクセスさせる。子プロセスはそれぞれ3回連続して用意した Web ページにアクセスする。このため、事前生成ありの場合には、サーバ側では事前に httpd プロセスと sh プロセスを 30 個ずつ用意した。

Tender の fork システムコールを改良し、処理の開始時に事前生成されたプロセスの有無をチェックし、もし事前生成されたプロセスが存在していれば、そのプロセスを利用して fork システムコール処理を実行し、処理を高速化するようにした。実験結果を表2に示し、説明する。

プロセスを事前生成した場合、事前生成しない場合に比べて、約 8% 応答時間を短縮できることがわかる。これは、アクセス集中時に事前生成したプロセスをそのまま利用することにより、fork システムコールの処理時間を短縮できるためである。

5. おわりに

本稿では、資源を事前生成することにより、OS 処理を高速化する手法を提案した。事前生成機構として、予測部、資源管理部、要求部、事前生成部がある。提案手法の有効性を確認するため、資源管理部と事前生成部の一部を実装し Apache を利用して評価した。評価の結果、CGI プログラムの実行において資源の事前生成機構を適用することで、応答時間を約 8% 短縮できることを示した。

今後の課題として、未実装部分の実現がある。

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: 16700038) による補助のもので行われた。

参考文献

- [1] S. Suranauwarat and H. Taniguchi, "The Design, Implementation and Initial Evaluation of an Advanced Knowledge-based Process Scheduler," ACM Operating Systems Review, vol.35, no.4, pp.61-81 (10, 2001).
- [2] 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 (12, 2000).