

既存APに対して資源管理を行うための制御方式

5H-6

吉田 芳浩

(NTT 情報システム本部)

1. はじめに

我々は、社内システムにおいて、遠隔マシンのDBアクセス、遠隔マシンのAP起動等の機能を提供するミドルソフトをすでに開発した。

しかし、その適用システムにおいて使用しているUNIXマシンは、CPU、メモリ、プロセス、IPC資源などの共有資源を様々なAPが平等に共用して走行する環境であり、他のAPの動作によっては性能が低下する危険性がある。

そこで本稿では、資源管理の第1ステップとして行った、上記ミドルソフトの機能を利用しているAPの使用する資源の管理について、その実現方式と有効性を述べる。

2. 解決すべき問題点

我々がすでに開発したミドルソフトの適用されているシステムは、ホスト、UNIXマシン、システムユーザの使う多数のPCが接続されている。

そのUNIXマシンにおいて、負荷が集中する事により性能低下が発生する、という問題が発生している。この問題について以下分析する。

3. 問題点の分析および検討対象の絞り込み

前述した環境において、UNIXマシンの処理内容には、

- ・ホストからの処理依頼
- ・自マシン発起の処理依頼
- ・ユーザがPC上で使うAPからの処理依頼

の3つが存在し、それぞれが相互の関連なく処理が受け付けられ実行されている。このうち特に、ユーザのシステム利用によるもの、その中でもDBアク

セスに伴う処理が、システム運用時間帯においては、UNIXマシンの能力のほとんどを使用し、また性能的にもシステム自体の評価に直結するところから、この処理について、過負荷状態を発生させないために負荷制御を実現させることを以下検討する。

4. 解決案を考える上での条件

1. 現在運用されているシステムのAPについて負荷制御を考える上で、そのAPの修正は全くできない。
2. また負荷制御する事により性能があまりにも落ちる事も許されない。従って、以下の2つを条件とする。
3. 戻り値の条件も含めてインタフェースの変更が無く
4. 性能に及ぼす影響が最も小さい方式とする

5. 過負荷状態の発生原因および事象

1. 通常のUNIXマシンでは、CPU、メモリ、プロセス、IPC資源などの限られた共有資源を様々なAPが平等に共用して走行する環境であり、他のAPの動作によっては資源の割り当てがうまくいかずに性能が低下する危険性がある。具体的な事象としては、以下のような事が考えられる。
2. プロセス数が多くなりスケジューリングの負荷が大きくなる
3. メモリSWAPが多発する

6. 過負荷状態を発生させないための機構

過負荷状態となる事を防ぐために、あらかじめ資源を確保し、その範囲でできる処理のみを行わせる機構を組み立てる事とする。

A study of UNIX resource management

Yoshihiro YOSHIDA

NTT Information Systems Headquarters

7. 実現方式

しきい値を設定し、現在の値を調べ、その値との比較により動作制御を行うこととして、比較する資源、現在の値の調べ方、制御の方法、について示す。

(1) 比較する資源、その現在の使用量の調べ方

また、UNIXにおいて処理は、プロセス生成から消滅までであり、その動作過程でメモリなどの資源を確保、開放を行うので、その処理の開始時点、つまりプロセス生成時点で制御を行うことによっても、資源使用量を1処理ごとの資源使用量と処理数のかけ算程度で大枠では抑えることができる。

そこで、制御の基点とする資源等とその情報の入手の方法には下表のようなものが考えられる。

資源等	情報入手手段	負荷の軽さ
処理受付数	受付時に記憶	◎
プロセス数	既存コマンド	△
	独自コマンド	○
メモリ使用量	既存コマンド	△
	独自コマンド	○
CPU負荷	既存コマンド	△
	独自コマンド	○

負荷の軽さを評価条件として、処理受付数を受付時に記憶しておく方式を採る。

(2) 制御の方法

上記のように、しきい値との比較を処理要求受付時に行う、とした場合に、その比較後、しきい値以上の値となってしまう場合の処理をどうするかは以下の事が考えられる。

- ・処理ができるようになるまで待たせる
- ・要求に対して処理不可という結果を返す

これについては、DBアクセスは一連の業務処理の中の1つの処理にすぎないため、処理不可という結果はAPとしては異常処理扱いとなってしまう。APに影響を及ぼさないためには、「処理ができるようになるまで待たせる」こととする。

8. 処理の流れ

処理の流れを図1に示す。

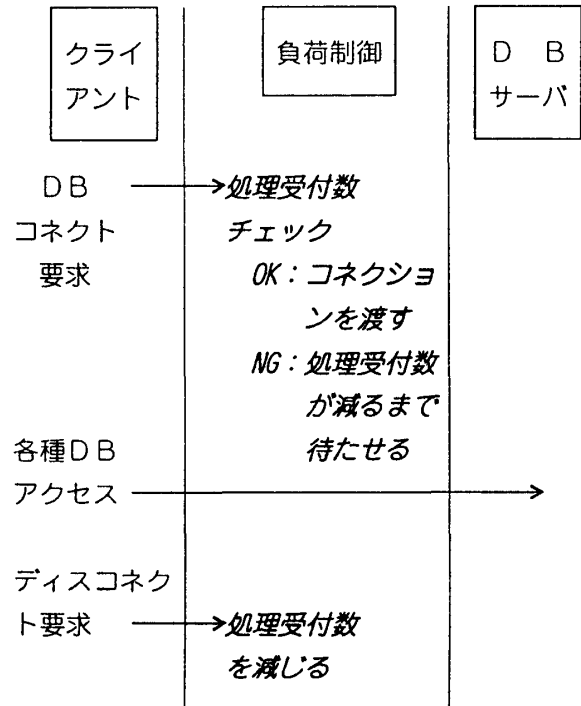


図1

9. まとめ

本稿では、資源管理の第1ステップとして行った、我々が開発したミドルソフトのDBアクセス機能を利用してAPの資源管理について、その実現方式と有効性を述べた。

いくつかの要点を示すと、

- ・過負荷状態となる事を防ぐために、あらかじめ資源を確保すること
- ・管理のための負荷を小さく抑えるための考慮
- ・返り値の条件が変わらないための考慮

があげられる。

これからの検討課題としては、実際の運用環境において、適切なしきい値を見いだす事、さらに上記ミドルソフトとインタフェースを持たないAPについても追加型のインタフェースを定義する事により管理下におくための方式の検討、を予定している。