

UNIX システムチューニング講習会に参加して

センター系（情報メディア教育センター） 早坂 成人

1. 研修期間・場所

期 間 2000年2月7日（月）～9日（水）
場 所 富士通 品川ラーニングセンター

2. 研修目的

UNIX システムの最適な運用を行うための性能評価の手法と、その評価に基づいたチューニングのための基本的な考え方及び方法の習得を目的とする。

3. 研修内容

3. 1 システムチューニングのポイント

システムチューニングとは、システムをより効果的に使用するために、システムを調整することである。システムのチューニングを行う時のポイントは、以下の3つである。

(1) ボトルネックを知る

システムの中に潜むいくつかのボトルネック（問題点）について、事前に理解しておく必要がある。システムのボトルネックとなる個所を知っておくことは、チューニングの方針を決定する判断材料となる。

(2) チューニング方針を決定する

チューニングの方針は、スループット（単位時間内に処理できる仕事量）の向上、またはレスポンスタイム（応答時間）の短縮の2つに大きく分けることができる。一般的にバッチ系ジョブには、スループットが要求され、会話型ジョブにはレスポンスが要求される。スループットとレスポンスは互いに相反するものなので、チューニングの方法も異なってくる。したがって、これら2つのどちらに重点を置いてチューニングを行うのか、事前に決定しておく必要がある。

(3) チューニングの影響を知る

チューニングを行うことにより、ある処理のパフォーマンスは向上するが、他の処理のパフォーマンスは逆に悪化することがある。そのためチューニング作業を行う際は、システム全体から見た影響を把握しておく必要がある。

3. 2 システムのボトルネック

(1) CPU の使用率

CPU の負荷が高くなると、システムのパフォーマンスは低下する。一般的にシステムのパフォーマンスが悪い場合、CPU の使用率が90%以上であれば、そのCPU の負荷は非常に高いと言える。その負荷が連続して高いか、それとも一時的であるか調査し、連続していればCPU の性能に問題があると考えられる。

(2) メモリの問題

仮想メモリシステムでは、ページングやスワッピングが発生する。特にページングは比

較的頻繁に発生する。空きメモリ量が少なくなるとページングが非常に高い確率で発生するようになり、最終的にはスワッピングが発生する可能性も高い。その結果、メモリ不足によるシステムの性能低下を招く。頻繁にページングが発生した場合、多くの入出力動作が発生し、そのためのオーバヘッドがシステムの性能に影響を与える。

(3) プロセスによる負荷の問題

カーネルはプロセスの優先順位を調整し、特定のプロセスが CPU を占有しないようにしており、複数のプロセスのレスポンスが悪化しないようにしている。しかし、多くのプロセスが同時期に集中した実行依頼をするとシステムのレスポンスが極端に悪化する。

そのため特定の時間に負荷が集中しないように、アプリケーション、コマンドなどのプロセス実行時刻の調整が必要となる。

(4) ファイルシステムの問題

ディスク装置の効率化を考えた場合、ディスクのスループットとスペースの効率を考慮する必要がある。論理ブロックサイズが大きい場合は1回のアクセスデータ量が多く、アクセス回数は少なくなるが、無駄なスペースを生じる場合がある。論理ブロックサイズが小さい場合はこの反対となる。また長時間ディスク装置を使いつづけるとフラグメンテーション（断片化）によりディスクスペースを十分に利用できなくなる。この場合にはファイルシステムの再構築などの修復が必要となる。このほかにファイルシステム及びパーティションの構成による問題もある。

3. 3 稼動状態を採取するツール

(1) システムアカウンティングツール

システム資源の使用率を示す各種の情報を、タイプ別にアカウンティングデータとして記録する。システムアカウンティングツールは、もともとシステムの利用者に課金をする目的で使用されてきたツールであるが、チューニングを行う際に有益なデータも含んでいる。特に、コネクト、プロセス、ディスクの各アカウンティング情報はシステムのボトルネックを探す手がかりを与えてくれる。

(2) システムアクティビティツール

システムの運用中に定期的に性能監視のためのデータを記録するツールである。

3. 4 ボトルネックの発見と分析

ボトルネックを発見するためには、過去のデータや問題発生時のデータを採取し、システムの性能問題の評価基準値と比較する。評価基準値はデータからまとめられた一般的な値でこれと比較し、ボトルネックとなる個所の分析を行う。

利用状況を把握するのに有用なコマンドとして `sar` コマンドがあるが、表 1 でも分かるようにこのコマンドだけでも多数の情報を収集することができる。`sar` コマンドのオプションを表 1 に示す。

表 1 sar コマンドのオプション

項目	内容	オプション
CPU	CPU使用状況	-u
	プロセスキュー状況	-q

メモリ	未使用メモリ情報	-r
	ページイン状況	-p
	ページアウト状況	-g
	スワップとスイッチングの状況	-w
入出力	システムバッファ使用状況	-b
	ディスク使用状況	-d
カーネル	システム内制御表の使用状況	-v
	カーネルメモリアロケーション状況	-k
アプリケーション	システムコール発行状況	-c
	メッセージ、セマフォ状況	-m
端末	端末装置の動作状況	-y
その他	全てのデータ表示	-A

チューニングの項目を大きく分けると、メモリ、プロセス、入出力、アプリケーション、ネットワークの5つになる。各項目は互いに関連し合っているため、チューニング時にはそれぞれの特徴を念頭に置いた作業が必要になる。

本来、第1のボトルネックとなるのはCPUの処理能力である。しかし、CPUへの負荷はCPUの性能の問題だけではなく、CPUを利用する各サブシステムやアプリケーションにも起因していることを忘れてはならない。従ってシステムの負荷が高くなっている場合は、CPU処理性能が限界になっていると考える前に、メモリ不足やアプリケーションの実行状態、入出力が効率的かどうかという点について、チェックする必要がある。

CPUの使用率から想像できるボトルネックの個所を図1に示す。

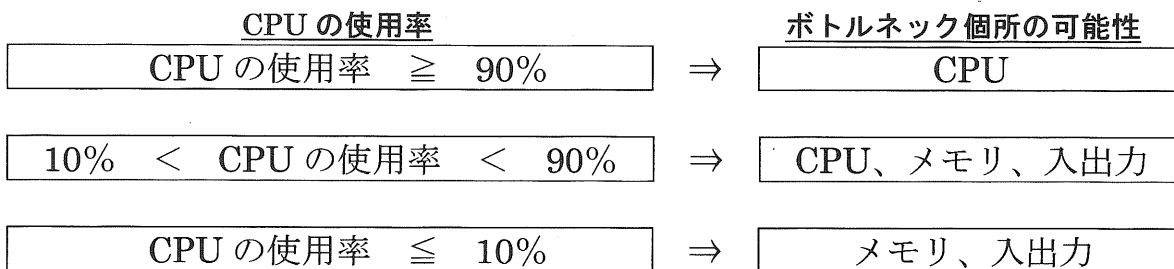


図1 ボトルネックの個所

4. 所感

この研修では実際にシステムに負荷を与えるプログラムを実行し、システムのどの部分がボトルネックとなっているのか演習を行った。実際にはかなり経験をつまなければ正確な追求は出来ないことが分かった。これは負荷のかかっているシステムの数値と評価基準値との比較を行うと、微かに値を超える項目が複数存在する場合もあるからである。複数の項目がボトルネックとなっている場合もあるが、その中の1項目だけが直接の原因の場合もある。最後にこれからもUNIXシステムを管理していくに当たり、この研修で学んできたことを生かしていきたいと思う。