

パフォーマンス モニタ ログの分析について

このマニュアルは、システム・リソースが原因か、または原因の一部である可能性があるパフォーマンスの問題を解決しようとしているユーザを対象としています。

はじめに

パフォーマンスの問題の原因を追跡することは困難です。インデックスを追加するのと同じくらい簡単なときもありますが、ほとんどの場合、互いに複合し合ういくつかの要素が原因となって問題が起きています。調査の早い段階で除外すべき潜在的な原因は、システムおよびエンジンのリソースです。いずれかのリソースが不足していると、当然、データベースのパフォーマンスが低下します。

パフォーマンス モニタ ログは複雑で、読み取ることは容易ではありません。システムが原因かどうかを判断するのに簡単な近道はありません。このマニュアルでは、注意すべきいくつかの重要部分と、分析を行う際の注意点について説明します。

パフォーマンス モニタ ログを分析する際は、何が問題であるか、システムがどのようにデータベースのパフォーマンスを低下させているか、そして最終的には、なぜシステムの速度低下の原因がデータベース・サーバであると思うかを常に考慮することが重要です。

一般的なヒント

パフォーマンス モニタ ログを分析する場合、不安定または一定でないシステム要素を探します。つまり、それ以上高速に動作できなくなるまで利用されているコンポーネントを探します。システム内でその部分を簡単に特定できる場合もありますが、直感や経験、または運が必要な場合もあります。

バグやその他のプロセスが問題の根源である可能性もありますが、一般に、エンジンのパフォーマンス低下の理由は、効率の悪いクエリ、スキーマ、プランの 3 つが主に考えられます。

パフォーマンス モニタ ログを調査する場合、特に注意すべきシステム領域は、CPU、メモリ、物理ディスクの 3 つです。これらの領域を詳しく調べたら、サーバの実行可能ファイルに関する情報を調査し、それが現時点までの調査結果とどのように関係しているかを確認してください。

パフォーマンス モニタ ログの読み方

パフォーマンス モニタ は、システム内でモニタしているすべての属性のスナップショットを定期的に取り得します。タイム・スライスごとに直線上のポイントで、各属性のその間隔における最小値、最大値、および平均値が示されます。

CPU

CPU は、システムのボトルネックになりやすい要素です。データベース・サーバが 1 億行のテーブルの結合を行っていたり、誰も見ていないときに駄目なシステム管理者が MPEG-4 ムービーをエンコードしていたりすると、簡単にシステムの動きが止まります。当然、CPU が最大限の能力を使用していた場合、システム上のすべてに影響が及びます。

マルチプロセッサやハイパー・スレッド・システムの CPU の統計情報を調べる場合、1 つだけではなくすべての CPU の統計情報を調べるのが重要です。

CPU パフォーマンスを分析する場合、調べるべきいくつかの重要な統計情報があります。

キューイング・レート

処理する操作が CPU に送られると、操作はキューに配置されます。このキューが定期的に急増するのは正常な動作ですが、キューがなくなりかけていないかどうかを確認する必要があります。この値が急激に増加している場合、システムが現在の負荷に耐えられなくなっています。このキューが非常に大きくなっている場合、システムが応答できなくなりそうだという事です。

1 秒あたりの割り込み数

1 つのソフトウェアがメモリの読み込みやディスクへの書き込みなどの操作を実行しようとするたびに、割り込みが呼び出されます。この値が最高値や最低値に達することがあるのは正常なことです。上昇し続けて安定しない場合は、プロセッサが受け取る要求の量に耐えられなくなっています。

プロセッサ時間 (%)

システムの CPU の使用量です。システムの負荷量を測る良い目安になります。マルチプロセッサ環境の Windows では、1 つのプロセッサは、システム・リソース合計のうちの 1 つのプロセッサ分の容量しか使用できません。要するに、システムの平均負荷は $100/N\%$ になるということです。N はシステム内のプロセッサ数です。1 つのプロセッサが 1 つの CPU を完全に使用することは可能ですが、その場合でも平均は必ず $100/N$ 以下でなければなりません。

割り込み時間 (%)

オペレーティング・システムが割り込み要求を処理する時間量です。この値は、他の値よりもかなり低い値で、比較的安定していなければなりません。

特権時間 (%)

オペレーティング・システムがシステム・アカウント（具体的にはカーネル）のために作業を行った時間量です。この値は、他の値よりもかなり低い値で、比較的安定していなければなりません。

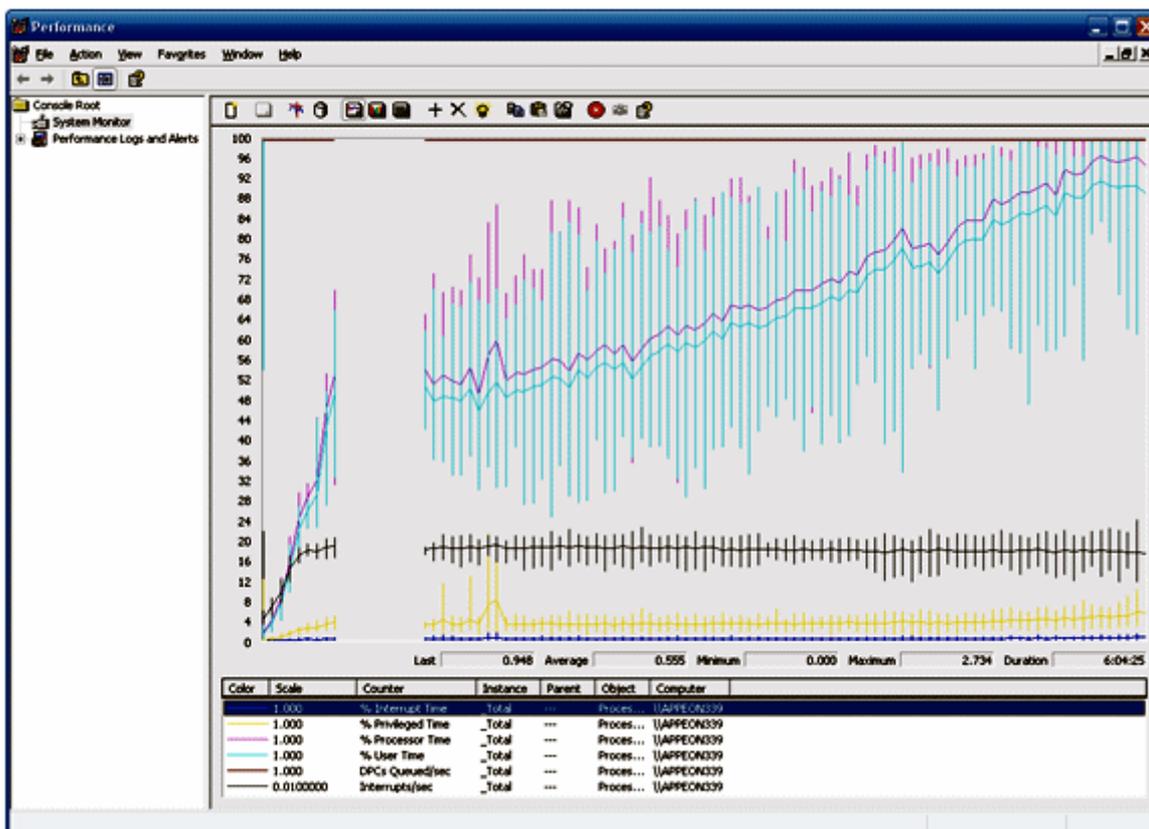
ユーザ時間 (%)

システムまたはカーネルとして実行されていないすべてが、ユーザ・プロセスとして分類されてこのカテゴリに分類されます。

例

CPU 統計のパフォーマンス・グラフの例として図 1 を参照してください。グラフから、モニタ期間中にプロセッサのリソース消費量が徐々に増加していることがわかります。特権命令は比較的安定しているため、プロセッサの負荷の原因はユーザ・プロセスにあると考えられます。また、命令キューおよび 1 秒あたりの割り込みもかなり安定しているため、これらはパフォーマンスの問題の原因ではなさそうです。

図 1 CPU のパフォーマンス統計



メモリ

メモリの統計情報を調べる場合、メモリ不足に注意することが重要です。利用可能なすべてのメモリを使い果たすことがシステム速度を低下させる主な要因であり、幸いにも比較的修正しやすい部分です。

利用可能量 (MB)

システムの利用可能なメモリの総量です。正常に稼働しているシステムでは、Windows がディスクにページ・アウトせずにプログラムにスペースを割り当てられるだけの十分な空き容量があります。常に 64MB ~ 128MB の空き容量を確保できるようにすることをおすすめします。この値が下降し続けて上昇しない場合、システムのどこかで隔離すべきメモリ・リークが発生しています。

1 秒あたりのページ・フォルト

現在メモリ内にはない情報に Windows がアクセスしようとする、Windows がページ・フォルトを起こします。これは、データの処理を行う前に、Windows がまずデータをメモリに読み込まなければならないということです。ページ・フォルトの急激な増加は、初めて読み込まれたプログラムであることを表します。一方、ページ・フォルトのレベルが継続的に高い場合は、システムの RAM が不足していて、Windows がプログラムの実行にスワップ・スペースを使い過ぎている可能性があります。スワップ・ス

ペースを使い過ぎていると、プログラムの実行パフォーマンスが大幅に低下する原因になります。

非ページ・プール (バイト)

非ページ・プールは、Windows がデバイス・ドライバおよびカーネル用のメモリ割り当てに使用するメモリ領域です。システムが安定稼働している場合、このメモリ量は比較的安定します。システムが RAM を使い果たしている場合、Windows がプールから RAM を取得してプロセスに割り当て始めます。この量が低下している場合、Windows がこのプールからメモリを取得しなければならない状況になっているか、またはデバイス・ドライバでメモリ・リークが発生しています。

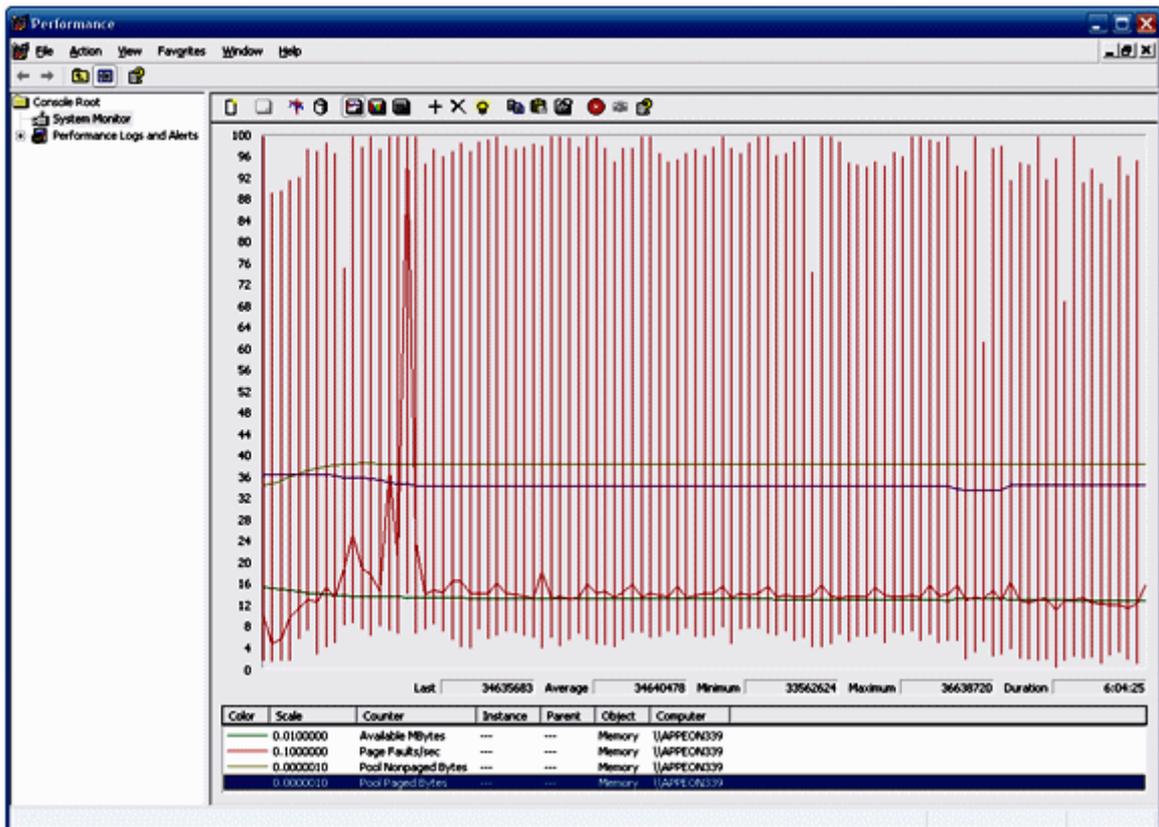
ページ・プール (バイト)

カーネルの外側で実行しているものはすべて、ページ・プールからメモリを取得します。この値が低下したまま上昇しない場合、ユーザ・プログラムでメモリ・リークが発生している可能性があります。

例

図 2 から、すべてが非常に安定していることがわかります。ページ・フォルトが一時的に上昇するところがありますが、何らかのものが初めて読み込まれたことが原因であると思われます。オペレーティング・システムが何らかのものを初めてメモリに読み込むと、ページ・フォルトが発生します。

図 2 – メモリのパフォーマンス統計



ディスク IO

ディスク書き込み

システムの物理ディスクで行われた書き込み操作の総数です。この数は比較的安定します。

ディスク読み込み

システムの物理ディスクで行われた読み込み操作の総数です。この数は比較的安定します。

現在のディスク・キューの長さ

コンピュータの IO サブシステムが現在の要求を処理できない場合、その要求はキューに配置されます。キューにアイテムがあるのは正常なことです。一定期間キューの長さが 5 を超えたままである場合は注意する必要があります。値が高いと、IO サブシステムがシステムの負荷を処理できず、パフォーマンスの低下につながる可能性があります。

平均のディスク・キューの長さ

この統計情報と現在のディスク・キューの長さから、システムの IO サブシステムによる現在の負荷の処理状態を判断できます。この値の最小値が 0 よりも大きい場合、IO サブシステムを調査すれば、パフォーマンスを改善できる可能性があります。

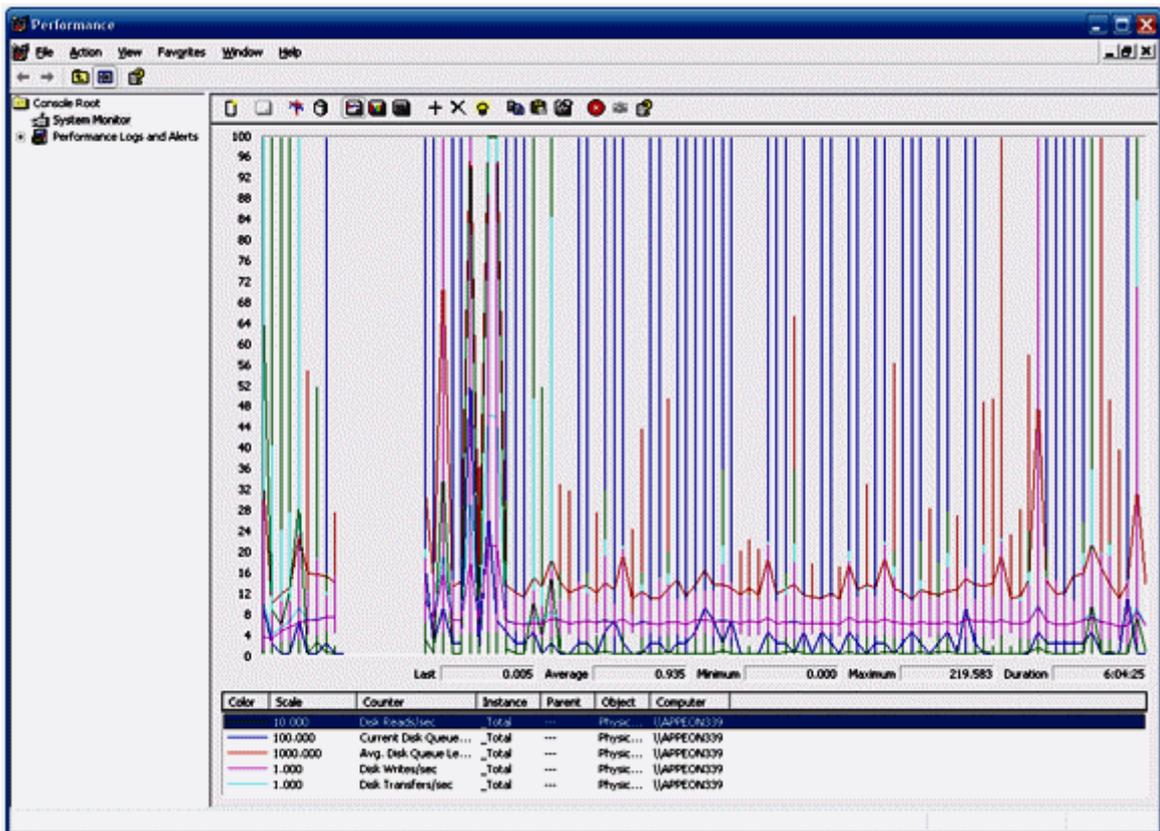
1 秒あたりのディスク転送

IO 操作の総数は知っておくと便利です。この情報から、物理ドライブ上のデータへのアクセスにどれだけの時間が実際に費やされているかを求めることができます。ハード・ディスク・ドライブの場合、1 分あたり 7200 回転が標準で、最大回転遅延は 60/7200 秒または約 8 ミリ秒です。最大回転遅延の半分である平均回転遅延も有益な概念です。エン트리・レベルのハード・ディスクの標準的なシーク・タイムは約 8 ミリ秒で、エンタープライズ・クラスのハイ・エンド・ディスク・ドライブの場合は 4 ミリ秒です。ほとんどの計算において、あらゆる操作のアクセス時間合計が 8 ~ 10 ミリ秒前後であると想定するのが無難です。

例

図 3 は非常に混乱しているように見えるかもしれませんが、すべての操作の平均値は比較的安定しています。平均のディスク・キューの長さは、読み取りやすくするために大幅に拡大されています。実際の平均値はほとんど 0 です。すべてのページ・フォルトとほぼ同じ時間にアクティビティが急増しています。このことから、プログラムが初めてメモリに読み込まれたことがわかります。

図 3 – ディスク IO のパフォーマンス統計



エンジン

うまくいけば、これまでに見たデータからシステムの具体的な 1 つの領域が判明し、そこを調べることができます。次に、データベース・エンジンの具体的な統計情報を調べることで、うまくいけば、エンジンがパフォーマンス低下の原因であるという仮説を確定または否定できます。

プロセッサ時間

この値は、エンジンがどれくらいプロセッサ時間を使用しているかを表します。プロセッサの利用率が平均で $100/N$ を超えることがない場合でも、エンジンがシステム内のいずれかの CPU の 100% の上限に達する可能性があることに注意してください。

ハンドル数

Windows では、ほとんどすべてのリソースがハンドルを介して取得されます。ハンドル数が上昇したまま下らない場合、エンジンがリソースを解放していません。

1 秒あたりの IO データ操作

この値は、ディスクの読み込みと書き込みの合計数です。この情報から、物理ドライブ上のデータへのアクセスにどれだけの時間が実際に費やされているかを求めることができます。ハード・ディスク・ドライブの場合、1分あたり7200回転が標準で、最大回転遅延は60/7200秒または約8ミリ秒です。最大回転遅延の半分である平均回転遅延も有益な概念です。エントリ・レベルのハード・ディスクの標準的なシーク・タイムは約8ミリ秒、エンタープライズ・クラスのハイ・エンド・ディスク・ドライブの場合は4ミリ秒です。ほとんどの計算において、あらゆる操作のアクセス時間合計が8～10ミリ秒前後であると想定するのが無難です。

1 秒あたりの読み込み操作

データベースによって要求された読み込み操作の総数です。この数と、同じ期間のオペレーティング・システムの読み込み数を比較します。これら2つの値がほぼ同じである場合、データベース・エンジンがシステムのIOを独占していると考えられます。エンジンの読み込み操作数がオペレーティング・システムの読み込み数よりも著しく少ない場合、他のプロセスがディスク・アクセスにおいてデータベース・エンジンと競合しており、その結果、エンジンのパフォーマンスに影響が及んでいると考えられます。

1 秒あたりの書き込み操作

データベースによって要求された書き込み操作の総数です。読み込み操作数の比較と同じ考え方が書き込み操作にも当てはまります。

スレッド数

エンジンが実行しているスレッドの総数です。スレッド数が上昇したまま下がらない場合、エンジンでスレッド・リークが発生しています。エンジニアリングを行って問題を詳しく調査する必要があります。高負荷状態に対応するためにこの数が上昇することはありますが、ほとんどの場合、この数はかなり安定します。

仮想バイト

エンジンに割り当てられた仮想メモリの総量です。この数が増えている場合、システムでメモリが不足し、Windowsがデータベースのデータの一部をディスクにページ・アウトしています。その場合、データベースのパフォーマンスに非常に悪い影響が及びます。

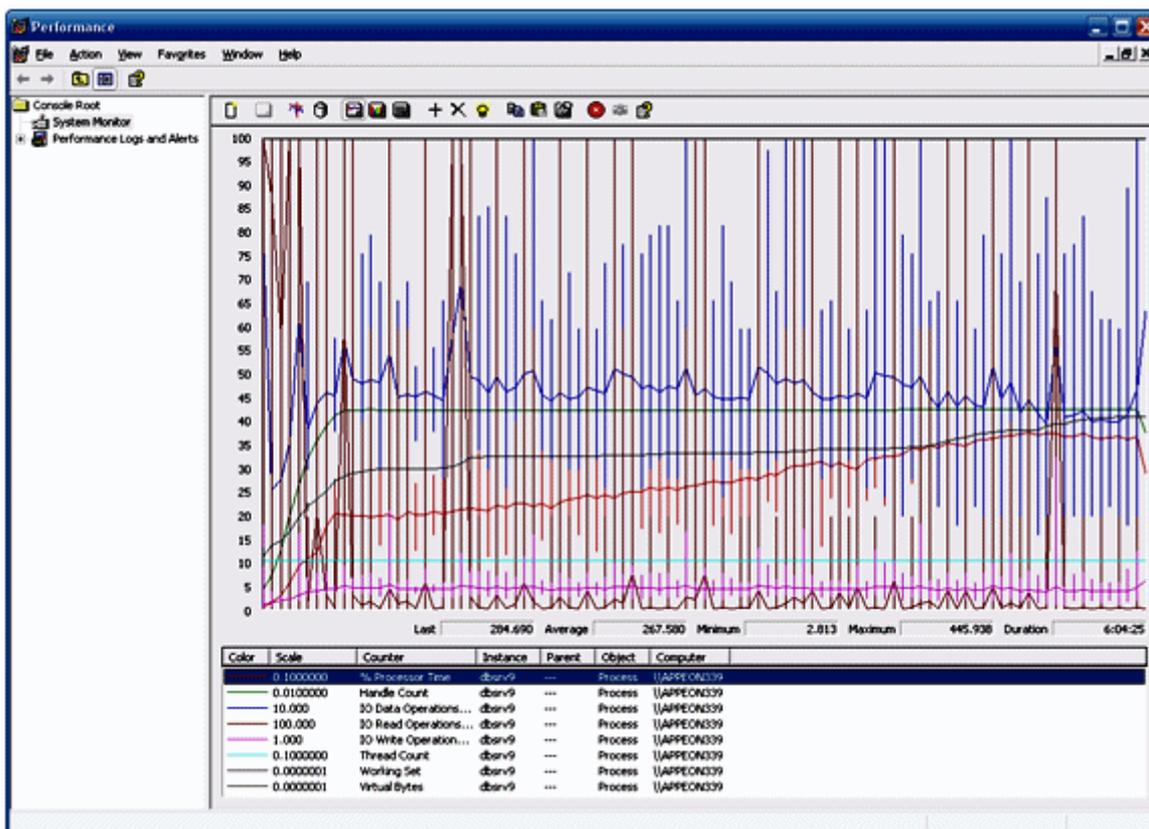
ワーキング・セット

エンジンが利用できるメモリの総量です。この数値が増していないことを確認します。上昇している場合、システムでメモリ・リークが発生しています。

例

図4でも、すべてがほぼ安定しています。時々、アクティビティが急増することがありますが、これは、負荷がかかるサーバでは予想されることです。特に興味深いのは、エンジンがCPUリソースを徐々に占有していくように見えることです。これは、エンジンに徐々に負荷がかかってゆき、その結果、CPUの負荷が直線的に上昇していることを表します。

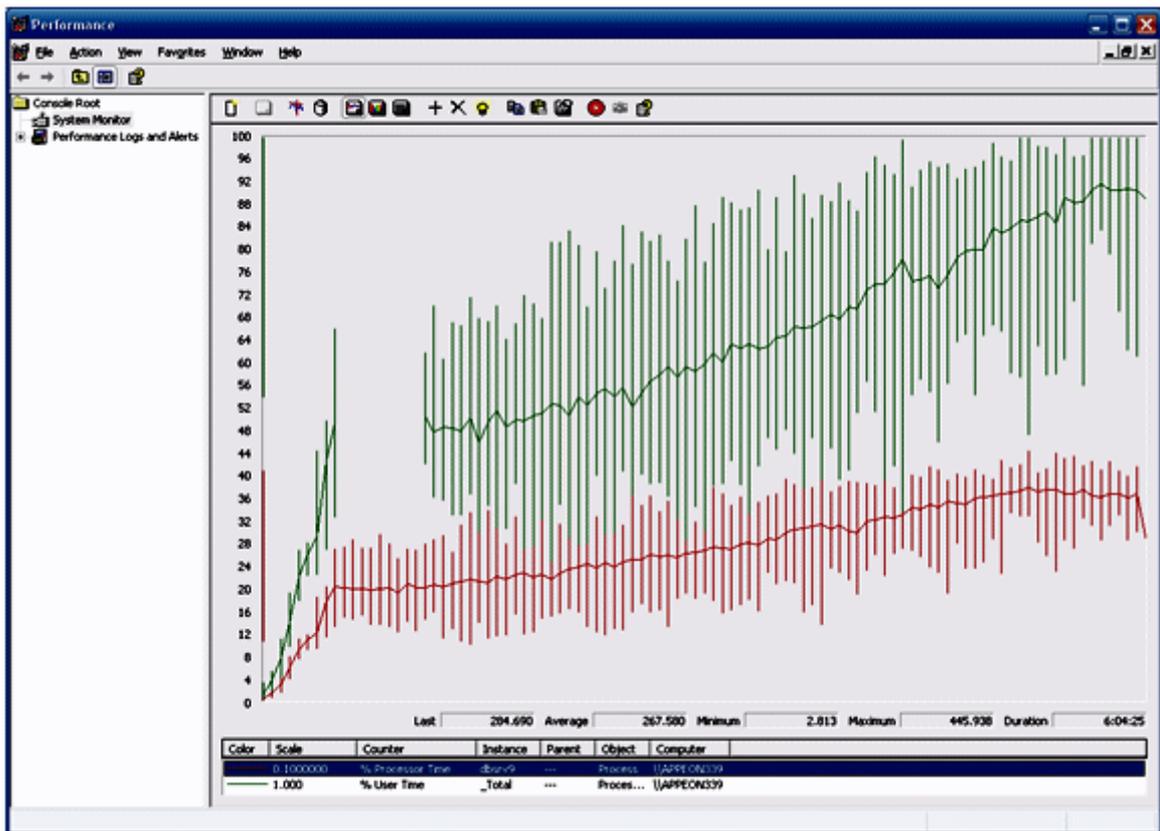
図 4 - DBSRV9.exe のパフォーマンス統計



判定

図 5 から、ユーザ・プロセッサの消費時間のパーセンテージと、データベース・エンジンのプロセッサ時間が、だいたい同じ比率で上昇していることがわかります。それぞれのラインのカーブが非常に似ているので、システムの CPU リソースが不足しているのはデータベース・サーバの負荷が原因であると考えるのが妥当です。

図 5 – エンジンの CPU 使用量とユーザ・プロセッサ時間の比較



結論

パフォーマンス モニタ のさまざまな統計情報を使用して、システムのパフォーマンスを調べられることがわかりました。このマニュアルに記載されている例は非常に理解しやすくなっていますが、速度が低下しない場合や考えられる原因が複数ある場合は、パフォーマンス モニタ ログの分析が困難な場合もあります。パフォーマンス モニタ ログに決定的な兆候が見られない場合、データベースに対して実行されているクエリを調査し、それらを最適化する方法を見つけた方がよい場合があります。

パフォーマンス モニタ ログの分析は、やればやるほどうまくなります。パフォーマンスとチューニングは、どちらも部品の芸術科学であるため、上達するには練習するしかありません。

このマニュアルで解説した パフォーマンス モニタ の出力の収集方法については、関連技術ドキュメント『パフォーマンス モニタ でのパフォーマンス情報の収集と読み込み』を参照してください。