

監視/最適化/アーキテクチャ/ファイル・システム

パフォーマンス改善

小林 明

プロセスの実行速度をもう少し改善できないかと思ったことはないでしょうか。一般的なアプローチとしては、改善対象のプロセスのボトルネックが何なのかを探り、改善していくことです。ここではシステ

ム・パフォーマンスを監視するツールの紹介や、GCCコンパイラの最適化オプション、Arm32ビットと64ビットの比較、ファイル・システムの性能比較など、実測データを交えて紹介したいと思います。

5-1 パフォーマンスの監視

システムをチューニングし、パフォーマンスを改善するには、どこにボトルネックがあるのか、チューニング対象が何なのかを特定する必要があります。その原因が、コンピューティング・リソースの不足や性能

によることもあります。

表1に、システム・パフォーマンスを監視するのに役立つコマンドを示します。

表1 システム・パフォーマンス監視に役立つコマンド

コマンド	説明
ps	選択したアクティブなプロセスに関する情報を表示する。プロセスの状態、CPU使用率、メモリ、スケジューラ、I/O、oom、cgroupに関する情報、スリープしているカーネル関数などが確認できる。選択内容と表示された情報を繰り返し更新する場合は、代わりにtopコマンドを使用する
free	システム内の空きメモリと使用済みの物理メモリ、スワップ・メモリの合計量、およびカーネルによって使用されるバッファとキャッシュを表示する。繰り返し更新する場合は、代わりにtopを使用する
htopまたはtop	実行中のシステムの動的なりリアルタイム・ビューアを提供する。サマリ・エリアでuptime、(1、5、15分間の)Load average、タスク数、CPU使用時間(コアごと)、メモリ使用量を確認できる ※Load averageに関しては/proc/loadavgを参照
vmstat	プロセス、メモリ、ページング、ブロックI/O、トラップ、ディスク、CPU アクティビティに関する情報を報告する。インターバルと繰り返し回数を指定できる
iostat	CPU使用状況とデバイス使用状況に関する情報を報告する。インターバルと繰り返し回数を指定できる
netstat	ネットワーク接続、ルーティング・テーブル、インターフェースの統計情報などLinux ネットワーク・サブシステムに関する情報を出力する
perf	Linuxカーネルが提供するパフォーマンス・カウンタにより、システム全体や特定のプロセスのパフォーマンスを詳細に解析。 Linuxのパフォーマンス・カウンタは、あらゆるパフォーマンス分析のためのフレームワークを提供する新しいカーネル・ベースのサブシステムである。ハードウェア・レベル(CPU、パフォーマンス監視ユニット)の機能とソフトウェア機能(ソフトウェア・カウンタ、トレース・ポイント)もカバーしている。 パフォーマンス・カウンタは、実行された命令、発生したキャッシュ・ミス、分岐の誤予測などのハードウェア・イベントをカウントするCPUハードウェア・レジスタ。 トレース・ポイントは、システム・コール、TCP/IPイベント、ファイル・システム操作など、コード内の論理的な場所に配置される計装ポイント
ftrace	ftraceは、Linuxカーネルに組み込まれているトレース機構。これを使用することで、動作中のカーネルのイベントを記録し、ログとして採取することができ、遅延やパフォーマンスの問題をデバッグまたは分析するために使用できる。トレーサとして、関数トレース、レイテンシ・トレース、イベント・トレースなどがある