Red Hat Enterprise Linux 7

電源管理ガイド

RHEL 7 での消費電力の管理と最適化

Last Updated: 2024-09-24

Red Hat Enterprise Linux 7 電源管理ガイド

RHEL 7 での消費電力の管理と最適化

Marie Doleželová
Red Hat Customer Content Services
mdolezel@redhat.com

Jana Heves
Red Hat Customer Content Services

Jacquelynn East
Red Hat Customer Content Services

Don Domingo
Red Hat Customer Content Services

Rüdiger Landmann
Red Hat Customer Content Services

Jack Reed
Red Hat Customer Content Services

Red Hat, Inc.

法律上の通知

概要

このドキュメントでは、Red Hat Enterprise Linux 7 システムの電力消費を効果的に管理する方法を説明します。次のセクションでは、(サーバーとラップトップの両方で) 消費電力を削減するさまざまな手法と、各手法がシステムの全体的なパフォーマンスに与える影響を説明します。

目次

第1章 概要

電源管理は、Red Hat Enterprise Linux 7 の改善の焦点の 1 つです。コンピューターシステムで使用される電力を制限することは、*グリーン IT* (環境にやさしいコンピューティング) の最も重要な側面の 1 つであり、リサイクル可能な材料の使用、ハードウェアの生産による環境影響、システムの設計と展開における環境意識も含む一連の考慮事項です。このドキュメントでは、Red Hat Enterprise Linux 7 を実行しているシステムの電源管理に関するガイダンスと情報を提供します。

1.1. 電源管理の重要性

電源管理のコアとなるのは、各システムコンポーネントのエネルギー消費を効果的に最適化する方法を理解することです。これには、システムが実行するさまざまなタスクを調査し、各コンポーネントを設定して、そのパフォーマンスがジョブに最適であることを確認する必要があります。

電源管理の主な動機は次のとおりです。

- 全体的な消費電力を削減してコストを節約

電源管理を適切に使用すると、次の結果が得られます。

- サーバーやコンピューティングセンターにおける熱の削減
- 冷却、空間、ケーブル、ジェネレーター、*無停電電源装置* (UPS) などの二次コストの削減
- ノートパソコンのバッテリー寿命の延長
- 二酸化炭素排出量の削減
- 政府の規制や Green IT (Energy Star など) に関する法的要件に合致する
- 新システムに関する企業ガイドラインの遵守

原則として、特定のコンポーネント (またはシステム全体) の消費電力を下げると、熱が下がり、自然にパフォーマンスが低下します。そのため、特にミッションクリティカルなシステムの場合は、作成した設定によってもたらされるパフォーマンスの低下を徹底的に調査およびテストする必要があります。

システムが実行するさまざまなタスクを調査し、各コンポーネントを設定してそのパフォーマンスがジョブに十分であることを確認することで、エネルギーを節約し、熱の発生を抑え、ラップトップのバッテリー寿命を最適化できます。消費電力に関するシステムの分析とチューニングの原則の多くは、パフォーマンスチューニングの場合と似ています。通常、システムはパフォーマンスまたは電力のいずれかに向かって最適化されるため、電力管理とパフォーマンスチューニングは、システム設定に対する反対のアプローチです。このマニュアルでは、Red Hat が提供するツールと、このプロセスを支援するために開発した技術を説明します。

Red Hat Enterprise Linux 7 には、デフォルトで有効になっている多くの新しい電源管理機能がすでに付属しています。それらはすべて、典型的なサーバーまたはデスクトップのユースケースのパフォーマンスに影響を与えないように選択されています.ただし、最大のスループット、最小の待機時間、または最高の CPU パフォーマンスが絶対に必要な非常に特殊なユースケースでは、これらのデフォルトの見直しが必要になる場合があります。

このドキュメントで説明されている手法を使用してマシンを最適化する必要があるかどうかを判断するには、以下の質問を活用してください。

問： 最適化する必要がありますか?

答： 電力最適化の重要性は、従う必要のあるガイドラインがあるかどうか、または満たす必要のある規制があるかどうかによって異なります。

問： どのくらい最適化する必要がありますか?

答： ここで紹介するいくつかの手法では、マシンを詳細に監査および分析するプロセス全体を実行する必要はありませんが、代わりに、一般的に電力使用量を改善する一連の一般的な最適化を提供します。もちろん、それらは通常、手動で監査および最適化されたシステムほど優れていませんが、適切な妥協点を提供します.

問： 最適化によって、システムのパフォーマンスが許容できないレベルまで低下しますか?

答： このドキュメントで説明されている手法のほとんどは、システムのパフォーマンスに大きな影響を与えます。Red Hat Enterprise Linux 7 ですでに導入されているデフォルト以外の電源管理を実装することを選択した場合は、電源の最適化後にシステムのパフォーマンスを監視し、パフォーマンスの低下が許容できるかどうかを判断する必要があります。

問： システムを最適化するために費やされる時間とリソースは、得られる利益を上回りますか?

答： プロセス全体に従って単一のシステムを手動で最適化することは、通常は価値がありません。これに費やされる時間とコストは、単一のマシンの寿命にわたって得られる典型的なメリットよりもはるかに高いからです。一方、たとえば、10000 台のデスクトップシステムをすべて同じ設定とセットアップを使用してオフィスにデプロイメントする場合は、1 つの最適化されたセットアップを作成し、それを 10000 台のマシンすべてに適用することを推奨します。

次のセクションでは、エネルギー消費に関して、最適なハードウェアパフォーマンスがシステムにどのように役立つかを説明します。

1.2. 電源管理の基本

効果的な電源管理は、以下の原則に基づいて行われます。

An idle CPU should only wake up when needed

Red Hat Enterprise Linux 6 以降では、カーネルが *tickless* を実行しています。つまり、以前の定期的なタイマー割り込みが、オンデマンド割り込みに置き換えられたことを意味します。そのため、新しいタスクが処理のキューに追加されるまで、アイドル状態の CPU はアイドル状態を維持できます。低電力状態にある CPU は、この状態を持続できます。ただし、システムに、不要なタイマーイベントを作成するアプリケーションが存在する場合は、この機能の利点が相殺される可能性があります。ボリュームの変更やマウスの動きの確認などのポーリングイベントは、このようなイベントの例です。

Red Hat Enterprise Linux 7 には、CPU 使用率に基づいてアプリケーションを識別し、監査するツールが同梱されています。詳細は、*2章 電源管理の監査と分析* を参照してください。

Unused hardware and devices should be disabled completely

これは特に、可動部品 (ハードディスクなど) を持つデバイスに当てはまります。また、一部のアプリケーションでは、使用されていない有効なデバイスが "open" 状態のままにすることがあります。これが発生すると、カーネルは、そのデバイスが使用中であることを想定します。これにより、そのデバイスが省電力状態にならないようにできます。

Low activity should translate to low wattage

ただし、多くの場合は、これは最新のハードウェアと正しい BIOS 設定に依存します。古いシステムコンポーネントは、Red Hat Enterprise Linux 7 で現在サポートできる新機能の一部をサポートしていないことがよくあります。システムに最新の公式ファームウェアを使用していること、および BIOS の電源管理またはデバイス設定セクションで電源管理機能が有効になっていることを確認してください。以下のような機能を確認してください。

- SpeedStep
- PowerNow!
- Cool'n'Quiet
- ACPI (C 状態)
- Smart

ハードウェアでこの機能に対応し、BIOS で有効になっている場合は、Red Hat Enterprise Linux 7 がデフォルトで使用します。

Different forms of CPU states and their effects

最新の CPU は、*ACPI (Advanced Configuration and Power Interface)* とともに、さまざまな電源状態を提供します。3 つの異なる状態は以下のとおりです。

- スリープ (C-state)
- 周波数と電圧 (P-state)

  P 状態は、プロセッサーの周波数とその電圧動作点を表し、どちらも P 状態が増加するにつれてスケーリングされます。

- 熱の出力 (T-states または「熱状態」)

可能な限り低いスリープ状態で実行している CPU は、消費するワット量が最も少なくなりますが、必要に応じてその状態からウェイクアップするのにかなりの時間がかかります。まれに、スリープ状態に切り替わるたびに CPU が即座にウェイクアップしなければならなくなることがあります。この状況は、実質的に永続的に CPU がビジー状態になり、別の状態を使用すると潜在的な省電力の一部が失われます。

A turned off machine uses the least amount of power

当たり前のように聞こえるかもしれませんが、実際に電力を節約する最善の方法の 1 つは、システムの電源を切ることです。たとえば、会社では、昼休みや帰宅時にマシンをオフにするガイドラインを使用して、Green IT を意識することに焦点をあてた企業文化を育成できます。また、複数の物理サーバーを 1 つの大きなサーバーに統合し、Red Hat Enterprise Linux 7 に同梱される仮想化技術を使用して仮想化することもできます。

第2章 電源管理の監査と分析

2.1. 監査および分析の概要

通常、1つのシステムで詳細な手動の監査、分析、およびチューニングを行う場合は例外となります。これは、このようなシステム調整の最後の部分から得られる利点よりも、その実行にかかる時間とコストの方が長いためです。ただし、すべてのシステムで同じ設定を再利用できるほぼ同一のシステムでこれらのタスクを1回実行することは、非常に便利です。たとえば、数千ものデスクトップシステムや、マシンがほぼ同一の HPC クラスターをデプロイメントする場合を考えてください。監査と分析を行うもう1つの理由は、将来のシステム動作のリグレッションまたは変更を特定できる比較の基礎を提供することです。この分析の結果は、ハードウェア、BIOS、またはソフトウェアの更新が定期的に行われ、消費電力に関する予期しない事態を回避したい場合に非常に役立ちます。通常、徹底的な監査と分析により、特定システムで実際に起こっていることをより的確に把握できます。

利用可能な最新のシステムを使用しても、消費電力に関するシステムの監査と分析は比較的困難です。ほとんどのシステムは、ソフトウェアを介して電力使用量を測定するために必要な手段を提供していません。ただし、例外があります。Hewlett Packard サーバーシステムの iLO 管理コンソールには、Web からアクセスできる電源管理モジュールがあります。IBM は、BladeCenter 電源管理モジュールで同様のソリューションを提供します。一部の Dell システムでは、IT Assistant は電力監視機能も提供します。他のベンダーは、サーバープラットフォームで同様の機能を提供する可能性が高くなりますが、すべてのベンダーで対応している唯一のソリューションは存在しません。

多くの場合、消費電力を直接測定する必要があるのは、可能な限り節約を最大化するためだけです。幸いなことに、変更が有効かどうか、システムがどのように動作しているかを測定する他の手段を利用できます。この章では、必要なツールについて説明します。

2.2. POWERTOP

Red Hat Enterprise Linux 7 でのティックレスカーネルの導入により、CPU がより頻繁にアイドル状態になることができるようになり、消費電力が削減され、電力管理が改善されます。**PowerTOP** ツールは、CPU を頻繁にウェイクアップするカーネルおよびユーザー空間アプリケーションの特定のコンポーネントを特定します。**PowerTOP は** 開発時に監査を実行するために使用され、このリリースでは多くのアプリケーションが調整され、不必要な CPU ウェイクアップが 10 分の 1 に削減されました。

Red Hat Enterprise Linux 7 には、**PowerTOP** のバージョン 2.x が付属しています。このバージョンは、1.x コードベースを完全に書き直したものです。より明確なタブベースのユーザーインターフェイスが特徴で、カーネルの "perf" インフラストラクチャーを広範囲に使用して、より正確なデータを提供します。システムデバイスの電源動作が追跡され、目立つように表示されるため、問題をすばやく特定できます。より実験的には、2.x コードベースには、個々のデバイスやプロセスが消費している電力を示すことができる電力推定エンジンが含まれています。図2.1「稼働中の PowerTOP」 を参照してください。

PowerTOP を インストールするには、**root** として次のコマンドを実行します。

```
~]# yum install powertop
```

PowerTOP を実行するには、**root** として次のコマンドを使用します。

```
~]# powertop
```

PowerTOP は、 システムの合計電力使用量の推定値を提供し、各プロセス、デバイス、カーネル作業、タイマー、および割り込みハンドラーの個別の電力使用量を表示できます。このタスク中、ラップトップはバッテリ電源で動作する必要があります。電力推定エンジンを調整するには、**root** として次のコマンドを実行します。

```
~]# powertop --calibrate
```

調整には時間がかかります。このプロセスではさまざまなテストが実行し、輝度レベルが繰り返され、デバイスのオンとオフが切り替えられます。プロセスを終了し、キャリブレーション中はマシンを操作しないでください。調整プロセスが完了すると、**PowerTOP は** 通常どおり起動します。データを収集するために約1時間実行します。十分なデータが収集されると、消費電力の見積もりが最初の列に表示されます。

ラップトップでコマンドを実行している場合は、使用可能なすべてのデータが表示されるように、バッテリー電源で実行する必要があります。

PowerTOP は 実行中にシステムから統計を収集します。**概要** タブでは、CPU にウェイクアップを最も頻繁に送信しているコンポーネント、または最も多くの電力を消費しているコンポーネントのリストを表示できます (図2.1「稼働中の PowerTOP」)。隣接する列には、電力の見積もり、リソースの使用状況、1秒あたりのウェイクアップ、コンポーネントの分類 (プロセス、デバイス、タイマーなど)、およびコンポーネントの説明が表示されます。1秒あたりのウェイクアップ数は、サービスまたはデバイス、ならびにカーネルのドライバーがいかに効率的に実行しているかを示します。ウェイクアップが少ないということは、消費電力が少ないことを意味します。コンポーネントは、電力使用率をどの程度まで最適化できるかによって順序付けられます。

通常、ドライバーコンポーネントのチューニングにはカーネルの変更が必要ですが、これはこのドキュメントでは扱いません。ただし、ウェイクアップを送信するユーザーランドプロセスは、より簡単に管理できます。まず、このサービスまたはアプリケーションをこのシステムで実行する必要があるかどうかを判断します。そうでない場合は、単に無効にします。古い System V サービスを完全にオフにするには、次のコマンドを実行します。

```
~]# systemctl disable servicename.service
```

プロセスの詳細については、**root** として次のコマンドを実行してください。

```
~]# ps -awux | grep processname
~]# strace -p processid
```

トレースが繰り返されているように見える場合は、おそらくビジーループです。通常、このようなバグを修正するには、そのコンポーネントのコードを変更する必要があります。

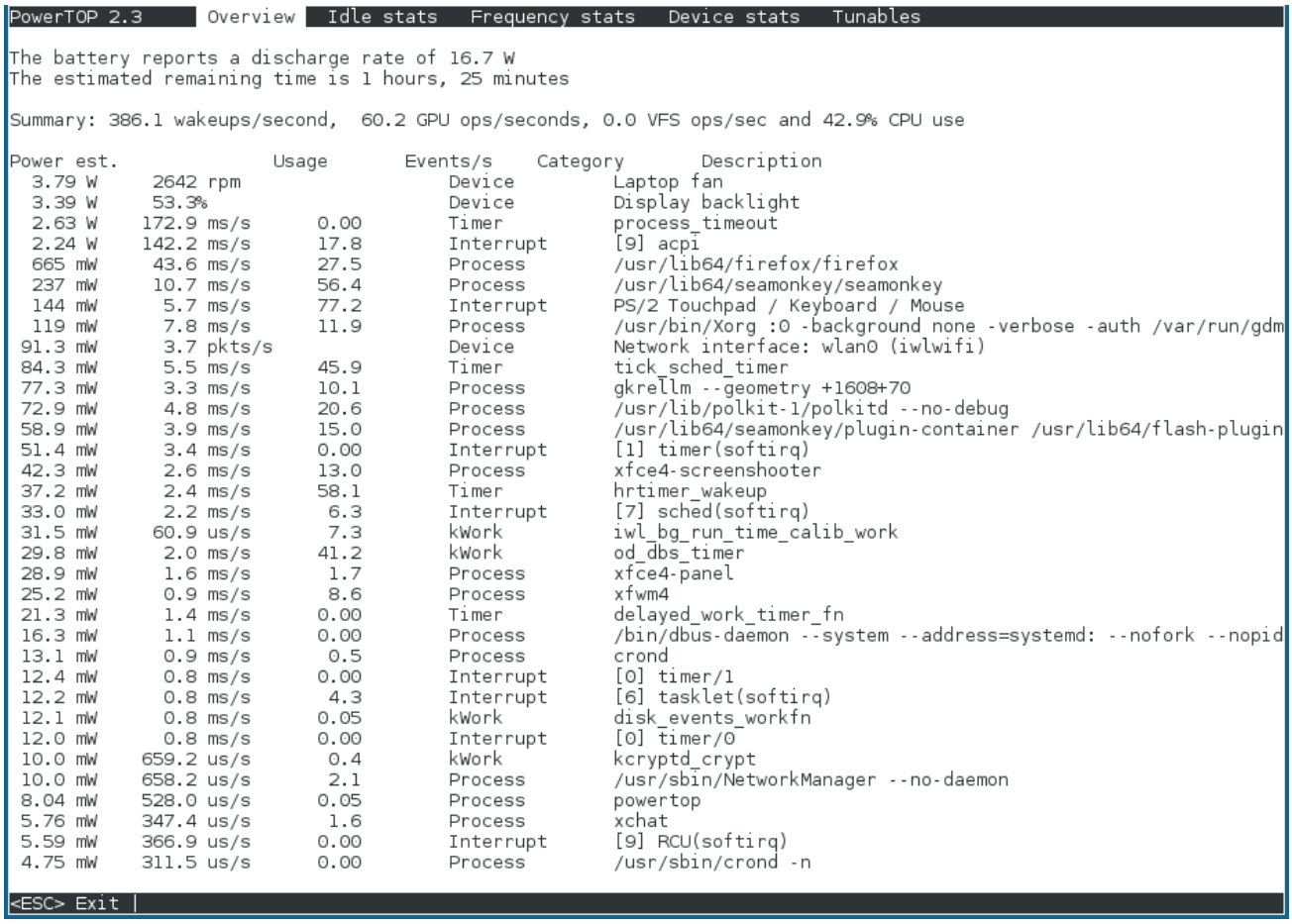
図2.1「稼働中の PowerTOP」 に見られるように、総電力消費量とバッテリー残量が表示されます (該当する場合)。これらの下には、1秒あたりの合計ウェイクアップ、1秒あたりの GPU 操作、および1秒あたりの仮想ファイルシステム操作を示す短い要約があります。画面の残りの部分には、使用率に従ってソートされたプロセス、割り込み、デバイス、およびその他のリソースのリストがあります。適切に調整すると、最初の列にリストされているすべての項目に対する電力消費予測も表示されます。

Tab キーと **Shift+Tab** キーを使用してタブを切り替えます。**アイドル統計** タブには、すべてのプロセッサーとコアの C ステートの使用が表示されます。**周波数統計** タブには、ターボモード (該当する場合) を含む P ステートの使用状況がすべてのプロセッサーとコアに対して表示されます。CPU がより高い C または P ステートに長く留まるほど、より良い状態になります (**C4 はC3** よりも高い)。これは、CPU 使用率がどの程度最適化されているかを示す良い指標です。レジデンシーは、システムがアイドル状態の間、最高の C または P 状態で 90% 以上であることが理想的です。

デバイス統計 タブには、**概要** タブと同様の情報が表示されますが、デバイスのみが対象となります。

Tunables タブには、消費電力を低減するためにシステムを最適化するための提案が含まれています。**上下** キーを使用して候補内を移動し、**Enter** キーを使用して候補 **の** オンとオフを切り替えます。

図2.1 稼働中の PowerTOP



[D]

--html オプションを指定して **PowerTOP** を実行して、HTML レポートを生成することもできます。*htmlfile.html* パラメーターを、出力ファイルに必要な名前に置き換えます。

```
~]# powertop --html=htmlfile.html
```

デフォルトでは、**PowerTOP は** 20 秒間隔で測定を行いますが、**--time** オプションを使用して変更できます。

```
~]# powertop --html=htmlfile.html --time=seconds
```

PowerTOP の詳細については、PowerTOP のホームページ を参照してください。

PowerTOP は、ターボスタット ユーティリティーと組み合わせて使用することもできます。Intel 64 プロセッサーのプロセッサートポロジー、周波数、アイドル状態の電源状態の統計情報、温度、および電力使用量に関する情報を表示するレポートツールです。**turbostat** ユーティリティーの詳細については、**turbostat** (8) のマニュアルページを参照するか、パフォーマンスチューニングガイドを 参照してください。

2.3. DISKDEVSTAT AND NETDEVSTAT

Diskdevstat と **netdevstat は**、システム上で実行されているすべてのアプリケーションのディスクアクティビティーとネットワークアクティビティーに関する詳細情報を収集する **SystemTap** ツールです。これらのツールは、各アプリケーションによる 1 秒あたりの CPU ウェイクアップ数を表示する **PowerTOP** からインスピレーションを得たものです (「PowerTOP」)。これらのツールが収集する統計

により、少数の大規模な操作ではなく、多くの小規模な I/O 操作で電力を浪費しているアプリケーションを特定できます。転送速度のみを測定する他の監視ツールは、このタイプの使用状況を特定するのに役立ちません。

root として次のコマンドを使用して、**SystemTap** でこれらのツールをインストールします。

```
~]# yum install tuned-utils-systemtap kernel-debuginfo
```

次のコマンドでツールを実行します。

```
~]# diskdevstat
```

またはコマンド:

```
~]# netdevstat
```

どちらのコマンドも、次のように最大 3 つのパラメーターを取ることができます。

diskdevstat *update_interval total_duration display_histogram*

netdevstat *update_interval total_duration display_histogram*

update_interval
表示の更新間隔 (秒単位)。デフォルト: **5**

total_duration
実行全体の秒単位の時間。デフォルト: **86400** (1 日)

display_histogram
実行の最後に収集されたすべてのデータのヒストグラムを作成するかどうかをフラグします。

出力は **PowerTOP** の出力に似ています。より長い時間の **diskdevstat** 実行からのサンプル出力を次に示します。

```
PID  UID  DEV WRITE_CNT WRITE_MIN WRITE_MAX WRITE_AVG READ_CNT READ_MIN
READ_MAX READ_AVG COMMAND
 2789  2903 sda1     854     0.000   120.000    39.836      0     0.000     0.000     0.000 plasma
 5494     0 sda1       0     0.000     0.000     0.000    758     0.000     0.012     0.000 0logwatch
 5520     0 sda1       0     0.000     0.000     0.000    140     0.000     0.009     0.000 perl
 5549     0 sda1       0     0.000     0.000     0.000    140     0.000     0.009     0.000 perl
 5585     0 sda1       0     0.000     0.000     0.000    108     0.001     0.002     0.000 perl
 2573     0 sda1      63     0.033  3600.015   515.226      0     0.000     0.000     0.000 auditd
 5429     0 sda1       0     0.000     0.000     0.000     62     0.009     0.009     0.000 crond
 5379     0 sda1       0     0.000     0.000     0.000     62     0.008     0.008     0.000 crond
 5473     0 sda1       0     0.000     0.000     0.000     62     0.008     0.008     0.000 crond
 5415     0 sda1       0     0.000     0.000     0.000     62     0.008     0.008     0.000 crond
 5433     0 sda1       0     0.000     0.000     0.000     62     0.008     0.008     0.000 crond
 5425     0 sda1       0     0.000     0.000     0.000     62     0.007     0.007     0.000 crond
 5375     0 sda1       0     0.000     0.000     0.000     62     0.008     0.008     0.000 crond
 5477     0 sda1       0     0.000     0.000     0.000     62     0.007     0.007     0.000 crond
 5469     0 sda1       0     0.000     0.000     0.000     62     0.007     0.007     0.000 crond
 5419     0 sda1       0     0.000     0.000     0.000     62     0.008     0.008     0.000 crond
```

```
5481    0 sda1      0    0.000    0.000    0.000    61    0.000    0.001    0.000 crond
5355    0 sda1      0    0.000    0.000    0.000    37    0.000    0.014    0.001 laptop_mode
2153    0 sda1     26    0.003 3600.029 1290.730     0    0.000    0.000    0.000 rsyslogd
5575    0 sda1      0    0.000    0.000    0.000    16    0.000    0.000    0.000 cat
5581    0 sda1      0    0.000    0.000    0.000    12    0.001    0.002    0.000 perl
5582    0 sda1      0    0.000    0.000    0.000    12    0.001    0.002    0.000 perl
5579    0 sda1      0    0.000    0.000    0.000    12    0.000    0.001    0.000 perl
5580    0 sda1      0    0.000    0.000    0.000    12    0.001    0.001    0.000 perl
5354    0 sda1      0    0.000    0.000    0.000    12    0.000    0.170    0.014 s  h
5584    0 sda1      0    0.000    0.000    0.000    12    0.001    0.002    0.000 perl
5548    0 sda1      0    0.000    0.000    0.000    12    0.001    0.014    0.001 perl
5577    0 sda1      0    0.000    0.000    0.000    12    0.001    0.003    0.000 perl
5519    0 sda1      0    0.000    0.000    0.000    12    0.001    0.005    0.000 perl
5578    0 sda1      0    0.000    0.000    0.000    12    0.001    0.001    0.000 perl
5583    0 sda1      0    0.000    0.000    0.000    12    0.001    0.001    0.000 perl
5547    0 sda1      0    0.000    0.000    0.000    11    0.000    0.002    0.000 perl
5576    0 sda1      0    0.000    0.000    0.000    11    0.001    0.001    0.000 perl
5518    0 sda1      0    0.000    0.000    0.000    11    0.000    0.001    0.000 perl
5354    0 sda1      0    0.000    0.000    0.000    10    0.053    0.053    0.005 lm_lid.sh
```

列は次のとおりです。

PID

アプリケーションのプロセス ID

UID

アプリケーションが実行しているユーザー ID

DEV

I/O が発生したデバイス

WRITE_CNT

書き込み操作の総数

WRITE_MIN

2 つの連続した書き込みにかかった最短時間 (秒単位)

WRITE_MAX

2 つの連続した書き込みにかかった最大時間 (秒単位)

WRITE_AVG

2 つの連続した書き込みにかかった平均時間 (秒単位)

READ_CNT

読み取り操作の総数

READ_MIN

2 つの連続した読み取りにかかった最短時間 (秒単位)

READ_MAX

2 つの連続した読み取りにかかった最大時間 (秒単位)

READ_AVG

2 つの連続した読み取りにかかった平均時間 (秒単位)

COMMAND

プロセスの名前

この例では、3 つの非常に明白なアプリケーションが際立っています。

```
PID  UID  DEV WRITE_CNT WRITE_MIN WRITE_MAX WRITE_AVG READ_CNT READ_MIN
READ_MAX READ_AVG COMMAND
 2789  2903 sda1     854     0.000   120.000    39.836      0     0.000     0.000     0.000 plasma
2573     0 sda1      63     0.033  3600.015   515.226      0     0.000     0.000     0.000 auditd
 2153     0 sda1      26     0.003  3600.029  1290.730      0     0.000     0.000     0.000 rsyslogd
```

これら 3 つのアプリケーションの **WRITE_CNT は0** より大きく、これは測定中に何らかの形式の書き込みを実行したことを意味します。それらの中で、**プラズマは** かなりの程度で最悪の攻撃者でした。プラズマは最も多くの書き込み操作を実行し、当然ながら書き込み間の平均時間は最も短かったのです。したがって、電力効率の悪いアプリケーションが懸念される場合には、**プラズマ** が調査対象となる最適な候補となります。

strace および **ltrace** コマンドを使用すると、指定されたプロセス ID のすべてのシステムコールをトレースすることで、アプリケーションをより詳細に検査できます。現在の例では、次を実行できます。

```
~]# strace -p 2789
```

この例では、**strace** の出力には、ユーザーの KDE アイコンキャッシュファイルを書き込み用に開き、すぐにファイルを再度閉じるという 45 秒ごとの繰り返しパターンが含まれていました。これにより、ファイルのメタデータ (具体的には変更時刻) が変更されたため、ハードディスクへの物理的な書き込みが必要になりました。最後の修正は、アイコンが更新されていないときに不要な呼び出しが行われないようにすることでした。

2.4. バッテリ寿命ツールキット

Red Hat Enterprise Linux 7 では、バッテリー寿命とパフォーマンスをシミュレートおよび分析するテストスイートである **Battery Life Tool Kit**(BLTK) が導入されています。BLTK は、特定のユーザーグループをシミュレートする一連のタスクを実行し、結果を報告することでこれを実現します。BLTK はノートブックのパフォーマンスをテストするために特別に開発されましたが、**-a** を指定して起動すると、デスクトップコンピューターのパフォーマンスについても報告できます。

BLTK を使用すると、マシンの実際の使用に匹敵する非常に再現性の高いワークロードを生成できます。たとえば、**オフィスの** ワークロードでは、テキストを作成し、その内容を修正し、スプレッドシートに対しても同じことを行います。BLTK を **PowerTOP** またはその他の監査または分析ツールと組み合わせて実行すると、実行した最適化が、マシンがアイドリング時だけでなくアクティブに使用されているときに効果があるかどうかをテストできます。異なる設定でまったく同じワークロードを複数回実行できるため、異なる設定の結果を比較できます。

次のコマンドで BLTK をインストールします。

```
~]# yum install bltk
```

次のコマンドで BLTK を実行します。

```
~]$ bltk workload options
```

たとえば、**アイドル状態の** ワークロードを 120 秒間実行するには、次のようにします。

```
~]$ bltk -I -T 120
```

デフォルトで使用可能なワークロードは次のとおりです。

-I, --idle
システムがアイドル状態で、他のワークロードと比較するためのベースラインとして使用

-R, --reader
ドキュメントの読み取りをシミュレートします (デフォルトでは Firefox を使用)

-P, --player
CD または DVD ドライブからマルチメディアファイルの視聴をシミュレートします (デフォルトでは mplayer を使用)

-O, --office
OpenOffice.org スイートを使用したドキュメントの編集をシミュレートします

その他のオプションでは、次を指定できます。

-a, --ac-ignore
AC 電源が利用可能かどうかを無視します (デスクトップでの使用に必要)

-T *number_of_seconds*, --time *number_of_seconds*
テストを実行する時間 (秒単位)。**アイドル状態の** ワークロードにはこのオプションを使用してください

-F *filename*, --file *filename*
特定のワークロードで使用されるファイルを指定します。たとえば、CD または DVD ドライブにアクセスする代わりに **プレーヤー** のワークロードが再生するファイルです。

-W *application*, --prog *application*
特定のワークロードで使用されるアプリケーションを指定します。たとえば、**リーダーの** ワークロードには Firefox 以外のブラウザーを使用します。

BLTK は、より専門的なオプションを多数サポートしています。詳細については、**bltk の** マニュアルページを参照してください。

BLTK は、 生成した結果を **/etc/bltk.conf** 設定ファイルで指定されたディレクトリー (デフォルトでは **~/.bltk/*workload* .results) に保存します。*番号*/**。たとえば、**~/.bltk/reader.results.002/** ディレクトリーには、**リーダー** ワークロードの 3 番目のテストの結果が保持されます (最初のテストには番号が付けられていません)。結果は複数のテキストファイルに分散されます。これらの結果を読みやすい形式に要約するには、次を実行します。

```
~]$ bltk_report path_to_results_directory
```

結果は、結果ディレクトリー内の **Report** という名前のテキストファイルに表示されます。代わりに端末エミュレーターで結果を表示するには、**-o** オプションを使用します。

```
~]$ bltk_report -o path_to_results_directory
```

2.5. TUNED

Tuned は、**udev** デバイスマネージャーを使用して接続されたデバイスを監視し、システム設定の静的および動的チューニングの両方を可能にする、プロファイルベースのシステムチューニングツールです。動的チューニングは実験的な機能であり、Red Hat Enterprise Linux 7 ではデフォルトでオフになっています。

Tuned は、高スループット、低遅延、省電力などの一般的なユースケースを処理するための事前定義されたプロファイルを提供します。各プロファイルの **調整** ルールを変更し、特定のデバイスの調整方法をカスタマイズできます。**PowerTOP の** 提案からカスタム **Tuned** プロファイルを作成する方法については、次を参照してください。「powertop2tuned の使用」。

プロファイルは、使用中の製品に基づいてデフォルトとして自動的に設定されます。**tuned-adm recommend** コマンドを使用すると、Red Hat が特定の製品に最も適したプロファイルとして推奨するプロファイルを判断できます。利用可能な推奨事項がない場合は、**バランスの取れた** プロファイルが設定されます。

バランスの取れた プロファイルは、ほとんどのワークロードに適しており、エネルギー消費、パフォーマンス、遅延のバランスが取れています。**バランスの取れた** プロファイルを使用すると、利用可能な最大のコンピューティング能力でタスクを迅速に完了する方が、通常、より少ないコンピューティング能力で同じタスクを長時間実行するよりも少ないエネルギーで済みます。

ラップトップがアイドル状態にある場合、または計算負荷の低い操作のみを実行している場合、**省電力** プロファイルを使用すると、バッテリー寿命を延ばすことができます。このような操作では、エネルギー消費を抑える代わりに待ち時間が長くなることは一般的に許容されます。または、IRC を使用したり、単純な Web ページを表示したり、オーディオファイルやビデオファイルを再生したりするなど、操作をすばやく終了する必要はありません。

tuned -adm で提供される調整プロファイルと省電力プロファイルの詳細については、『Red Hat Enterprise Linux 7 パフォーマンスチューニングガイド』 の 調整の章 を参照してください。

powertop2tuned の使用

powertop2tuned ユーティリティーを使用すると、**PowerTOP の** 提案からカスタム **Tuned** プロファイルを作成できます。**PowerTOP** については、次を参照してください。「PowerTOP」。

powertop2tuned ユーティリティーをインストールするには、次を使用します。

```
~]# yum install tuned-utils
```

カスタムプロファイルを作成するには、次を使用します。

```
~]# powertop2tuned new_profile_name
```

デフォルトでは、**powertop2tuned は /etc/tuned/** ディレクトリーにプロファイルを作成し、現在選択されている **Tuned** プロファイルに基づいてカスタムプロファイルを作成します。安全上の理由から、新しいプロファイルでは最初はすべての **PowerTOP** チューニングが無効になっています。チューニング

を有効にするには、**/etc/tuned/*profile_name*/tuned.conf** ファイル内のチューニングのコメントを解除します。

--enable または **-e** オプションを使用すると、**PowerTOP** が提案するほとんどのチューニングを有効にする新しいプロファイルを生成できます。USB 自動サスペンドなど、既知の問題のある特定のチューニングはデフォルトで無効になっているため、手動でコメントを解除する必要があります。

デフォルトでは、新しいプロファイルはアクティブ化されていません。有効にするには、次を使用します。

```
~]# tuned-adm profile new_profile_name
```

powertop2tuned が サポートするオプションの完全なリストについては、次を使用してください。

```
~]$ powertop2tuned --help
```

2.6. UPOWER

Red Hat Enterprise Linux 6 では、**DeviceKit-power は、 HAL** の一部であった電源管理機能と、Red Hat Enterprise Linux の以前のリリースの **GNOME Power Manager** の一部であった機能の一部を引き受けました (「GNOME Power Manager」。Red Hat Enterprise Linux 7 では、 **DeviceKit-power の** 名前が **UPower** に変更されました。**UPower は**、デーモン、API、およびコマンドラインツールのセットを提供します。システム上の各電源は、物理デバイスであるかどうかにかかわらず、デバイスとして表されます。たとえば、ラップトップのバッテリーと AC 電源は両方ともデバイスとして表されます。

upower コマンドと次のオプションを使用してコマンドラインツールにアクセスできます。

--enumerate, -e

システム上の各電源デバイスのオブジェクトパスを表示します。次に例を示します。

```
/org/freedesktop/UPower/devices/line_power_AC
/org/freedesktop/UPower/devices/battery_BAT0
```

--dump, -d

システム上のすべての電源装置のパラメーターを表示します。

--wakeups, -w

システムの CPU ウェイクアップを表示します。

--monitor, -m

AC 電源の接続または切断、バッテリーの消耗など、電源装置の変更についてシステムを監視します。**Ctrl**+**C** を押してシステムの監視を停止します。

--monitor-detail

AC 電源の接続または切断、バッテリーの消耗など、電源装置の変更についてシステムを監視します。**--monitor-detail** オプションは、**--monitor** オプションよりも詳細を表示します。**Ctrl**+**C** を押してシステムの監視を停止します。

--show-info *object_path*, -i *object_path*

特定のオブジェクトパスで利用可能なすべての情報を表示します。たとえば、オブジェクトパス **/org/freedesktop/UPower/devices/battery_BAT0** で表されるシステム上のバッテリーに関する情報を取得するには、次を実行します。

```
~]$ upower -i /org/freedesktop/UPower/devices/battery_BAT0
```

2.7. GNOME POWER MANAGER

GNOME Power Manager は、 GNOME デスクトップ環境の一部としてインストールされるデーモンです。以前のバージョンの Red Hat Enterprise Linux で **GNOME Power Manager が** 提供していた電源管理機能の多くは、Red Hat Enterprise Linux 6 では **DeviceKit-power** ツールの一部となり、Red Hat Enterprise Linux 7 では **UPower** に名前が変更されました (「UPower」)。ただし、**GNOME Power Manager は** 引き続きその機能のフロントエンドです。**GNOME Power Manager は**、システムトレイのアプレットを通じて、システムの電源ステータスの変化を通知します。たとえば、バッテリから AC 電源への変更です。また、バッテリーのステータスを報告し、バッテリー残量が少なくなると警告を表示します。

2.8. 監査用のその他のツール

Red Hat Enterprise Linux7 は、システムの監査と分析を実行するためのツールをいくつか提供しています。これらのほとんどは、検出された情報を検証する場合や、特定部品に関する詳細な情報が必要な場合に備えて、補助情報源として使用できます。これらのツールの多くは、パフォーマンスチューニングにも使用されます。使用可能なオプションには、以下のものがあります。

vmstat

vmstat は、プロセス、メモリー、ページング、ブロック I/O、トラップ、CPU アクティビティーに関する詳細情報を提供します。これを使用して、システム全体が何をしているか、どこがビジーかを詳しく調べます。

iostat

iostat は **vmstat** に似ていますが、ブロックデバイス上の I/O のみを対象としています。また、より詳細な出力と統計も提供します。

blktrace

blktrace は、非常に詳細なブロック I/O トレースプログラムです。アプリケーションに関連付けられた単一のブロックに情報を分解します。これは、**discdevstat** と組み合わせると非常に便利です。

第3章 コアインフラストラクチャーとメカニズム

重要

この章で説明する **cpupower** コマンドを使用するには、kernel-tools パッケージがインストールされていることを確認してください。

3.1. CPU アイドル状態

x86 アーキテクチャーの CPU は、CPU の一部が非アクティブになったり、パフォーマンス設定を下げて実行されたりするさまざまな状態をサポートします。*C 状態* と呼ばれるこれらの状態により、システムは使用されていない CPU を部分的に非アクティブ化することで電力を節約できます。C-state には C0 から順に番号が付けられ、番号が大きいほど CPU 機能が低下し、省電力が大きくなります。指定された数の C-State はプロセッサー間でほぼ同じですが、状態の特定の機能セットの詳細はプロセッサーファミリーごとに異なります。C-States 0-3 は、以下のように定義されます。

C0

動作中または実行中の状態。この状態では、CPU は動作しており、アイドル状態ではありません。

C1, Halt

プロセッサーが命令を実行していないが、通常は低電力状態ではない状態。CPU は事実上遅延なしで処理を継続できます。C-State を提供するすべてのプロセッサーが、この状態に対応する必要があります。Pentium 4 プロセッサーは、C1E と呼ばれる拡張された C1 状態に対応しています。これは、低消費電力を実現する状態です。

C2, Stop-Clock

このプロセッサーのクロックが凍結されている状態ですが、レジスターとキャッシュの完全な状態を維持しているため、クロックを再開した後、すぐに処理を再開できます。この状態はオプションになります。

C3, Sleep

プロセッサーが実際にスリープ状態になり、キャッシュを最新の状態に保つ必要がない状態。このため、この状態からの起動は C2 からの起動よりかなり時間がかかります。繰り返しますが、これはオプションの状態です。

CPUidle ドライバーの使用可能なアイドル状態とその他の統計を表示するには、次のように入力します。

```
~]$ cpupower idle-info
```

Nehalem マイクロアーキテクチャーを備えた最近の Intel CPU は、CPU の電圧供給をゼロに下げることができる新しい C-State C6 が特徴ですが、通常は消費電力を 80% から 90% 削減します。Red Hat Enterprise Linux 7 のカーネルには、この新しい C-State の最適化が含まれます。

3.2. CPUFREQ

システムの電力消費と熱出力を削減する最も効果的な方法の 1 つは、CPUfreq です。CPUfreq は CPU 速度スケーリングとも呼ばれ、Linux カーネルのインフラストラクチャーで、電力を節約するために CPU 周波数をスケーリングできます。CPU スケーリングは、システム負荷に応じて、ACPI イベントに応答して自動的に、またはユーザー空間プログラムによって手動で行うことができ、プロセッサーのク

ロック速度をその場で調整できます。これにより、システムは減速したクロック速度で実行でき、電力を節約できます。周波数のシフトに関するルール (クロック速度の高速化または低速化、および周波数のシフト) は、CPUfreq ガバナーで定義されています。

3.2.1. CPUfreq ドライバー

CPUfreq には、ACPI CPUfreq ドライバーと Intel P-state ドライバーの 2 つのドライバーを使用できます。

ACPI CPUfreq

ACPI CPUfreq ドライバーは、カーネルとハードウェア間の通信を保証する ACPI を介して特定の CPU の周波数を制御するカーネルドライバーです。

Intel P-state

Red Hat Enterprise Linux 7 では、Intel P-state ドライバーに対応しています。このドライバーは、Intel Xeon E シリーズアーキテクチャーまたは新しいアーキテクチャーに基づくプロセッサーで、P-state 選択を制御するインターフェイスを提供します。Intel P-state は、setpolicy() コールバックを実装します。ドライバーは、cpufreq コアから要求されたポリシーに基づいて、使用する P-state を決定します。プロセッサーが次の P-state を内部で選択できる場合、ドライバーはこの責任をプロセッサーにオフロードします。そうでない場合は、次の P-state を選択するアルゴリズムがドライバーに実装されます。

Intel P-state は、P-state の選択を制御する独自の sysfs ファイルを提供します。これらのファイルは、/sys/devices/system/cpu/intel_pstate/ ディレクトリーにあります。ファイルに加えた変更は、すべての CPU に適用されます。このディレクトリーには、P-state パラメーターの設定に使用される 5 つのファイルが含まれています。

- **max_perf_pct** : ドライバーが要求する最大 P ステートを制限し、利用可能なパフォーマンスのパーセンテージで表します。利用可能な P-state パフォーマンスは、no_turbo 設定により削減できます (以下を参照)。
- **min_perf_pct** : min_perf_pct: ドライバーによって要求される最小 P ステートを制限します。最大 (ターボなし) パフォーマンスレベルのパーセンテージで表されます。
- **no_turbo** : ドライバーがターボ周波数範囲未満の P ステートを選択するように制限します。
- **turbo_pct** : ターボ範囲内のハードウェアによってサポートされる合計パフォーマンスの割合を表示します。この数は、ターボが無効になっているかどうかに関係ありません。
- **num_pstates** : ハードウェアによってサポートされている P ステートの数を表示します。この数は、ターボが無効になっているかどうかに関係ありません。

現在、Intel P-state は、対応している CPU にデフォルトで使用されています。ユーザーは、カーネルコマンドラインに以下を追加することで、ACPI CPUfreq の使用に切り替えることができます。

```
intel_pstate=disable
```

3.2.2. CPUfreq ガバナー

CPUfreq ガバナーは、システム CPU の電源特性を定義します。これは、CPU パフォーマンスに影響を及ぼします。各ガバナーには、ワークロードに関する固有の動作、目的、および適合性があります。このセクションでは、CPUfreq ガバナーを選択して設定する方法、各ガバナーの特性、および各ガバナーが適しているワークロードの種類を説明します。

警告

Red Hat Enterprise Linux 7 には、複数のコア CPUfreq ガバナーが同梱されています。デフォルトでは、Intel P-state ドライバーはアクティブモードで動作します。アクティブモードでは、**performance** と **powersave** の 2 つの CPUfreq ガバナーのみが使用可能です。**performance** および省電力 **powersave** Intel P-state CPUfreq ガバナーの機能は、同じ名前のコア CPUfreq ガバナーとは異なることに注意してください。

3.2.2.1. コア CPUfreq ガバナー

Red Hat Enterprise Linux 7 で利用可能なさまざまなタイプの CPUfreq ガバナーを以下に示します。

cpufreq_performance

パフォーマンスガバナーは、CPU が可能な限り高いクロック周波数を使用するように強制します。この頻度は静的に設定され、変更されません。このため、この特定のガバナーでは **省電力の利点はありません**。これは、数時間の負荷の高いワークロードにのみ適しています。その場合でも、CPU がめったに (またはまったく) アイドル状態にならない時間帯にのみ適しています。

cpufreq_powersave

対照的に、Powersave ガバナーは、CPU が可能な限り低いクロック周波数を使用するように強制します。この頻度は静的に設定され、変更されません。そのため、この特定のガバナーは電力を最大限に節約できますが、**CPU パフォーマンスが最も低く** なります。

ただし、(原則として) 全負荷時の低速 CPU は、負荷がかかっていない高速 CPU よりも多くの電力を消費するため、"powersave" という用語は誤解を招く場合があります。したがって、予想される低アクティビティー時に Powersave ガバナーを使用するように CPU を設定することが推奨されますが、その間に予想外の高負荷が発生すると、システムが実際により多くの電力を消費する可能性があります。

Powersave ガバナーは、簡単に言えば、「省電力」というよりも、CPU の「速度リミッター」です。これは、システムや、過熱が問題になる可能性がある環境で最も役立ちます。

cpufreq_ondemand

Ondemand ガバナーは動的なガバナーであり、システム負荷が高いときに CPU が最大クロック周波数を達成し、システムがアイドル状態のときに最小クロック周波数を達成できるようにします。これにより、システムはシステム負荷に応じて消費電力を調整できますが、**周波数切り替えの間の待ち時間** はかかります。このため、システムがアイドル状態と高負荷のワークロードを頻繁に切り替える場合、レイテンシーは、Ondemand ガバナーが提供するパフォーマンスや省電力の利点を相殺することができます。

ほとんどのシステムでは、Ondemand ガバナーにより、放熱、電力消耗、性能、および管理可能性について最適な妥協点を見つけることができます。システムが 1 日の特定の時間にのみビジー状態の場合、Ondemand ガバナーは、それ以上の介入なしに、負荷に応じて最大周波数と最小周波数を自動的に切り替えます。

cpufreq_userspace

Userspace ガバナーを使用すると、ユーザー空間プログラム、または root で実行しているプロセスで頻度を設定できます。すべてのガバナーの中で、Userspace は最もカスタマイズ可能で、設定方法に応じて、システムのパフォーマンスと消費の最適なバランスを実現できます。

cpufreq_conservative

Ondemand ガバナーと同様に、Conservative ガバナーも使用状況に応じてクロック周波数を調整します。ただし、Ondemand ガバナーはより早く (つまり、最大から最小へ、またその逆) 切り替えますが、Conservative ガバナーはより時間をかけて周波数を切り替えます。

これは、Conservative ガバナーが、単純に最大値と最小値の間で選択するのではなく、負荷に適していると見なされるクロック周波数に調整することを意味します。これにより、消費電力を大幅に節約できる可能性がありますが、Ondemand ガバナーよりも **はるかに長い遅延** が発生します。

注記

cron ジョブを使用してガバナーを有効にできます。これにより、指定した時間帯に特定のガバナーを自動的に設定できます。このため、(勤務時間後など) アイドル時間帯には低周波ガバナーを指定し、作業負荷が高い時間帯には高周波ガバナーに戻すことができます。
特定のガバナーを有効にする方法は、「CPUfreq セットアップ」 を参照してください。

3.2.2.2. Intel P-state の CPUfreq ガバナー

Intel P-state ドライバーは、次の 3 つの異なるモードで動作できます。

- ハードウェア管理の P 状態 (HWP) によるアクティブモード
- ハードウェア管理の P 状態 (HWP) を使用しないアクティブモード
- パッシブモード

デフォルトで、Intel P-state ドライバーは、CPU が HWP に対応しているかどうかに応じて、HWP の有無にかかわらずアクティブモードで動作します。

Active mode with hardware-managed P-states
HWP でアクティブモードが使用されている場合、Intel P-state ドライバーは、P-state 選択を実行するように CPU に指示します。ドライバーは、周波数のヒントを提供できます。ただし、最終的な選択は CPU の内部ロジックによって異なります。

HWP でアクティブモードにすると、Intel P-state ドライバーにより、2 つの P-state 選択アルゴリズムが提供されます。

- パフォーマンス
- Powersave

Performance ガバナーを使用すると、ドライバーは内部 CPU ロジックにパフォーマンス指向になるように指示します。P-state の範囲は、ドライバーが使用できる範囲の上限に制限されます。

Powersave ガバナーを使用すると、ドライバーは、内部 CPU ロジックに省電力指向になるように指示します。

Active mode without hardware-managed P-states
HWP を使用しないアクティブモードの場合、Intel P-state ドライバーは次の 2 つの P-state 選択アルゴリズムを提供します。

- パフォーマンス
- Powersave

Performance ガバナーを使用すると、ドライバーは使用できる最大の P-state を選択します。

Powersave ガバナーを使用すると、ドライバーは、現在の CPU 使用率に比例する P-state を選択します。この動作は、Ondemand CPUfreq コアガバナーに似ています。

パッシブモード
passive モードを使用すると、Intel P-state ドライバーは、従来の CPUfreq スケーリングドライバーと同じように機能します。利用可能なすべての汎用 CPUFreq コアガバナーを使用できます。

Intel P-state ガバナーの詳細は、intel_pstate CPU Performance Scaling Driver を参照してください。

3.2.3. CPUfreq セットアップ

すべての CPUfreq ドライバーは kernel-tools パッケージの一部として組み込まれており、自動的に選択されるため、CPUfreq をセットアップするには、ガバナーを選択するだけで済みます。

以下を使用して、特定の CPU で使用できるガバナーを表示できます。

```
~]# cpupower frequency-info --governors
```

次に、以下を使用して、すべての CPU でこれらのガバナーの 1 つを有効にできます。

```
~]# cpupower frequency-set --governor [governor]
```

特定のコアでのみガバナーを有効にするには、CPU 番号の範囲またはコンマ区切りのリストを指定して **-c** を使用します。たとえば、CPU 1-3 および 5 の Userspace ガバナーを有効にする場合、コマンドは次のようになります。

```
~]# cpupower -c 1-3,5 frequency-set --governor cpufreq_userspace
```

3.2.4. CPUfreq ポリシーと速度の調整

適切な CPUfreq ガバナーを選択したら、**cpupowerfrequency-info** コマンドを使用して CPU 速度とポリシー情報を表示し、**cpupowerfrequency-set** のオプションを使用して各 CPU の速度をさらに調整できます。

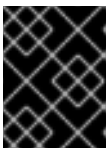
cpupowerfrequency-info では、次のオプションが利用可能です。

- **--freq** – CPUfreq コアに応じた CPU の現在の速度を KHz 単位で表示します。
- **--hwfreq** – ハードウェアに応じた CPU の現在の速度を KHz 単位で表示します (root としてのみ使用可能)。
- **--driver** – この CPU の周波数を設定するためにどの CPUfreq ドライバーが使用されているかを示します。
- **--governors** – このカーネルで使用可能な CPUfreq ガバナーを表示します。このファイルにリストされていない CPUfreq ガバナーを使用する場合は、その方法を 「CPUfreq セットアップ」 で参照してください。
- **--affected-cpus** – 周波数調整ソフトウェアを必要とする CPU をリストします。
- **--policy** – 現在の CPUfreq ポリシーの範囲 (KHz 単位)、および現在アクティブなガバナーを表示します。

- **--hwlimits** – CPU で使用可能な周波数を KHz 単位でリストします。

cpupowerfrequency-set では、次のオプションが利用可能です。

- **--min *<freq>*** および **--max *<freq>*** – CPU の *ポリシー制限を* KHz 単位で設定します。



重要な影響

ポリシー制限を設定するときは、**--min** の前に **--max** を設定する必要があります。

- **--freq *<freq>*** – CPU の特定のクロック速度を KHz 単位で設定します。CPU のポリシー制限内でのみ速度を設定できます (**--min** および **--max** に従って)。

- **--governor *<gov>*** – 新しい CPUfreq ガバナーを設定します。

注記

kernel-tools パッケージがインストールされていない場合は、**/sys/devices/system/cpu/*cpuid*/cpufreq/** にある調整パラメーターで CPUfreq 設定を確認できます。設定および値は、この調整可能パラメーターに書き込むことで変更できます。たとえば、最小クロック速度の cpu0 から 360 KHz を設定するには、次のコマンドを使用します。

```
echo 360000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq
```

3.3. CPU モニター

cpupower は、アイドル状態とスリープ状態の統計と周波数情報を提供し、プロセッサートポロジーをレポートする一連のモニターを備えています。一部のモニターはプロセッサー固有ですが、他のモニターはすべてのプロセッサーと互換性があります。各モニターの測定内容と互換性のあるシステムの詳細については 、**cpupower-monitor の** マニュアルページを参照してください。

cpupower Monitor コマンドで次のオプションを使用します。

- **-l** – システムで利用可能なすべてのモニターをリスト表示します。

- **-m *<monitor1>*、*<monitor2>*** – 特定のモニターを表示します。それらの識別子は、**-l** を実行すると見つかります。

- **コマンド** – 特定のコマンドのアイドル統計と CPU 要求を表示します。

3.4. CPU 省電力ポリシー

cpupower は、 プロセッサーの省電力ポリシーを調整する方法を提供します。

cpupower set コマンドで次のオプションを使用します。

--perf-bias *<0-15>*

サポートされている Intel プロセッサーのソフトウェアが、最適なパフォーマンスと省電力のバランスを決定するために、より積極的に貢献できるようにします。これは、他の省電力ポリシーを上書きしません。割り当てる値は 0 から 15 まであり、0 はパフォーマンスを最適化し、15 は電力効率を最適化します。

デフォルトでは、このオプションはすべてのコアに適用されます。個々のコアにのみ適用するには、**--cpu *<cpulist>*** オプションを追加します。

--sched-mc <0|1|2>

他の CPU パッケージが引き出される前に、システムプロセスによる電力の使用を 1 つの CPU パッケージのコアに制限します。0 は制限を設定せず、1 は最初に単一の CPU パッケージのみを採用し、2 はこれを行い、タスクのウェイクアップを処理するためにセミアイドル CPU パッケージを優先します。

--sched-smt <0|1|2>

システムプロセスによる電力の使用を、1 つの CPU コアのスレッドシブリングに制限してから、他のコアを使用します。0 は制限を設定せず、1 は最初に単一の CPU パッケージのみを採用し、2 はこれを行い、タスクのウェイクアップを処理するためにセミアイドル CPU パッケージを優先します。

3.5. 一時停止と再開

システムが一時停止すると、カーネルはドライバーを呼び出して状態を保存し、アンロードします。システムが再開すると、これらのドライバーが再ロードされ、デバイスの再プログラムが試みられます。このタスクを実行するドライバーの機能によって、システムを正常に再開できるかどうかが決まります。

Advanced Configuration and Power Interface (ACPI) 仕様では、システムファームウェアがビデオハードウェアを再プログラムできる必要がないため、ビデオドライバーはこの点で特に問題があります。したがって、ビデオドライバーが完全に初期化されていない状態からハードウェアをプログラムできない限り、システムの再開が妨げられる可能性があります。

Red Hat Enterprise Linux 7 には、新しいグラフィックスチップセットのサポートが強化されているため、一時停止と再開がより多くのプラットフォームで機能することが保証されます。特に、(特に GeForce 8800 シリーズの場合に) NVIDIA チップセットのサポートが大幅に改善されました。

3.6. ランタイムデバイスの電源管理

ランタイムデバイスの電源管理 (RDPM) は、ユーザーに見える影響を最小限に抑えて消費電力を削減するのに役立ちます。デバイスが十分な時間アイドル状態であり、デバイスとドライバーの両方に RDPM ハードウェアサポートが存在する場合、デバイスは低電力状態になります。低電力状態からの回復は、このデバイスの外部 I/O イベントによって保証されます。これにより、カーネルとデバイスドライバーがトリガーされ、デバイスが実行状態に戻ります。RDPM はデフォルトで有効になっているため、これはすべて自動的に行われます。

ユーザーは、特定の RDPM 設定ファイルで属性を設定することにより、デバイスの RDPM を制御できます。特定のデバイスの RDPM 設定ファイルは、**/sys/devices/*device*/power/** ディレクトリーにあります。ここで、*device は* 特定のデバイスのディレクトリーへのパスを置き換えます。

たとえば、CPU の RDPM を設定するには、次のディレクトリーにアクセスします。

```
/sys/devices/system/cpu/power/
```

デバイスを低電力状態から実行状態に戻すと、次の I/O 操作の待ち時間が長くなります。その追加の遅延の期間は、デバイス固有です。ここで説明する設定スキームにより、システム管理者はデバイスごとに RDPM を無効にし、他のパラメーターのいくつかを調べて制御することができます。すべての **/sys/devices/*device*/power** ディレクトリーには、次の設定ファイルが含まれています。

control
このファイルは、特定のデバイスの RDPM を有効または無効にするために使用されます。すべてのデバイスには、**制御** ファイル内の属性の次の 2 つの値のいずれかが含まれます。

auto

すべてのデバイスのデフォルト。ドライバーによっては、自動 RDPM の対象となる場合があります。

on

ドライバーが実行時にデバイスの電源状態を管理できないようにします

autosuspend_delay_ms
このファイルは、自動の一時停止遅延を制御します。これは、アイドル状態とデバイスのサスペンドの間の非アクティブの最小期間です。このファイルには、ミリ秒単位の自動の一時停止遅延値が含まれています。負の値を指定すると、実行時にデバイスが一時停止されなくなり、/sys/devices/*device*/power/control ファイルの属性を on に設定するのと同じ効果になります。1000 を超える値は、最も近い秒に切り上げられます。

3.7. アクティブ状態の電源管理

Active-State Power Management (ASPM) は、接続先のデバイスが使用されていないときに PCIe リンクの電力状態を低く設定することで、*Peripheral Component Interconnect Express* (PCI Express または PCIe) サブシステムの電力を節約します。ASPM は、リンクの両端で電力状態を制御し、リンクの端にあるデバイスが完全に電源オンの状態であっても、リンクの電力を節約します。

ASPM を有効にすると、異なる電源状態間でリンクを遷移させるのに時間がかかるため、デバイスの遅延が増加します。ASPM には、電源状態を決定する 3 つのポリシーがあります。

default
システムのファームウェア (BIOS など) によって指定されたデフォルトに従って、PCIe リンクの電源状態を設定します。これは ASPM のデフォルトの状態です。

powersave
パフォーマンスへのコストに関係なく、可能な限り電力を節約するように ASPM を設定します。

performance
ASPM を無効にして、PCIe リンクが最大のパフォーマンスで動作できるようにします。

pcie_aspm カーネルパラメーターを使用して、ASPM サポートを強制的に有効または無効にできます。

- **pcie_aspm=off** ASPM を無効にします
- **pcie_aspm=force** は、ASPM をサポートしていないデバイスでも ASPM を有効にします

ハードウェアが ASPM をサポートしている場合、オペレーティングシステムは起動時に自動的に ASPM を有効にします。ASPM のサポートを確認するには、次のコマンドの出力を参照してください。

```
~]$ journalctl -b | grep ASPM
```

警告 – PCIE_ASPM=FORCE により、システムが応答しなくなる可能性があります

ASPM をサポートしていないハードウェア上で pcie_aspm=force を使用して ASPM を強制的に有効にすると、システムが応答しなくなる可能性があります。pcie_aspm=force を設定する前に、システム上のすべての PCIe ハードウェアが ASPM をサポートしていることを確認してください。

ASPM ポリシーを設定するには、次のいずれかのオプションを使用します。

- /sys/module/pcie_aspm/parameters/policy ファイルの設定を変更します。
- ブート時に *pcie_aspm.policy* カーネルパラメーターを指定します

  たとえば、pcie_aspm.policy=performance は ASPM パフォーマンスポリシーを設定します。

3.8. アグレッシブリンク電源管理

Aggressive Link Power Management (ALPM) は、アイドル時間 (I/O がないとき) にディスクへの SATA リンクを低電力設定に設定することで、ディスクの電力を節約するのに役立つ省電力技術です。I/O 要求がそのリンクのキューに入れられると、ALPM は自動的に SATA リンクをアクティブな電源状態に戻します。

ALPM によって導入される省電力は、ディスクの遅延を犠牲にして実現されます。そのため、システムで長時間のアイドル I/O 時間が発生することが予想される場合にのみ、ALPM を使用してください。

ALPM は、*Advanced Host Controller Interface* (AHCI) を使用する SATA コントローラーでのみ使用できます。AHCI の詳細は、http://www.intel.com/technology/serialata/ahci.htm を参照してください。

利用可能な場合、ALPM はデフォルトで有効になっています。ALPM には 3 つのモードがあります。

min_power

このモードは、ディスクに I/O がない場合に、リンクを最低電力状態 (SLUMBER) に設定します。このモードは、長時間のアイドル時間が予想される場合に便利です。

medium_power

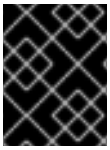
このモードは、ディスクに I/O がない場合に、リンクを 2 番目に低い電力状態 (PARTIAL) に設定します。このモードは、パフォーマンスへの影響を最小限に抑えながら、リンクの電源状態を移行できるように設計されています (たとえば、断続的な重い I/O とアイドル I/O の間)。

Medium_power モードでは、負荷に応じて、リンクが PARTIAL 状態とフル電力供給 (つまり ACTIVE) 状態の間で遷移できます。リンクを PARTIAL から SLUMBER に直接移行したり、その逆に移行したりすることはできないことに注意してください。この場合、どちらの電源状態も、最初に ACTIVE 状態を経由しないと、もう一方の状態に移行できません。

max_performance

ALPM は無効です。ディスクに I/O がない場合、リンクは低電力状態になりません。

SATA ホストアダプターが実際に ALPM をサポートしているかどうかを確認するには、ファイル /sys/class/scsi_host/host*/link_power_management_policy が存在するかどうかを確認します。設定を変更するには、このセクションで説明されている値をこれらのファイルに書き込むか、ファイルを表示して現在の設定を確認します。



重要 – ホットプラグを無効にする設定もあります

ALPM を min_power または middle_power に設定すると、ホットプラグ機能が自動的に無効になります。

3.9. RELATIME ドライブアクセスの最適化

POSIX 規格では、各ファイルが最後にアクセスされた日時を記録するファイルシステムメタデータをオペレーティングシステムが保持する必要があります。このタイムスタンプは atime と呼ばれ、これを維持するにはストレージへの一定の一連の書き込み操作が必要です。これらの書き込みは、ストレージデバイスとそのリンクをビジー状態に保ち、電源を入れたままにします。atime データを利用するアプリケーションはほとんどないため、このストレージデバイスの動作により電力が無駄になります。重要なのは、ファイルがストレージから読み取られたのではなく、キャッシュから読み取られた場合でも、ストレージへの書き込みが発生することです。しばらくの間、Linux カーネルは マウント の noatime オプションをサポートしていましたが、このオプションでマウントされたファイルシステムには atime データを書き込みませんでした。ただし、一部のアプリケーションは atime データに依存しており、それが利用できない場合は失敗するため、この機能を単にオフにするだけでは問題があります。

Red Hat Enterprise Linux 7 で使用されるカーネルは、別の代替手段である relatime をサポートしています。Relatime はatime データを維持しますが、ファイルがアクセスされるたびに維持するわけではありません。このオプションを有効にすると、atime データが最後に更新された (mtime) 以降にファイルが変更された場合、またはファイルが一定期間 (デフォルトでは 1 回) よりも前に最後にアクセスされた場合にのみ、atime データがディスクに書き込まれます。日)。

デフォルトでは、すべてのファイルシステムが relatime を有効にしてマウントされるようになりました。オプション noelatime を使用してファイルシステムをマウントすることにより、特定のファイルシステムに対してこれを抑制することができます。

3.10. 電力上限

Red Hat Enterprise Linux 7 は、HP *Dynamic Power Capping* (DPC) や Intel Node Manager (NM) テクノロジーなど、最近のハードウェアに見られる電力上限機能をサポートしています。電力上限により、管理者はサーバーが消費する電力を制限できますが、既存の電源装置が過負荷になるリスクが大幅に減少するため、管理者はデータセンターをより効率的に計画することもできます。管理者は、同じ物理フットプリント内により多くのサーバーを配置でき、サーバーの電力消費が制限されていれば、高負荷時の電力需要が利用可能な電力を超えないという確信が持てます。

HP 動的電力上限

動的電力上限は、一部の ProLiant および BladeSystem サーバーで利用できる機能で、システム管理者がサーバーまたはサーバーグループの電力消費を制限できるようにします。上限は、現在のワークロードに関係なく、サーバーが超えない決定的な制限です。サーバーが消費電力の上限に達するまで、上限は効果がありません。その時、管理プロセッサーは CPU の P ステートとクロックスロットリングを調整して、消費電力を制限します。

Dynamic Power Capping は、オペレーティングシステムとは無関係に CPU の動作を変更しますが、HP の *統合 Lights-Out 2* (iLO2) ファームウェアは、オペレーティングシステムが管理プロセッサーにアクセスできるようにするため、ユーザー空間のアプリケーションは管理プロセッサーにクエリーを実行できます。Red Hat Enterprise Linux 7 で使用されるカーネルには、HP iLO および iLO2 ファームウェア用のドライバーが含まれており、これによりプログラムは /dev/hpilo/d *X* ccb *N* にある管理プロ

セッサーにクエリーを実行できます。このカーネルには、電力制限機能をサポートするための **hwmon sysfs** インターフェイスの拡張機能と、**sysfs** インターフェイスを使用する ACPI 4.0 パワーメーター用の **hwmon** ドライバーも含まれています。これらの機能を組み合わせることで、オペレーティングシステムとユーザー空間ツールは、システムの現在の電力使用量と共に、電力上限に設定された値を読み取ることができます。

HP 動的消費電力上限の詳細は、https://support.hpe.com/hpsc/doc/public/display?docId=mmr_sf-EN_US000006556 で利用できる HPE のナレッジ記事 『HP ProLiant Gen8 Server Series - Power Capping and Dynamic Power Capping Settings in iLO 4 and RBSU』 を参照してください。

Intel Node Manager

Intel Node Manager は、プロセッサーの P ステートと T ステートを使用してシステムに電力上限を課し、CPU のパフォーマンスを制限して消費電力を制限します。管理者は、電源管理ポリシーを設定することにより、夜間や週末など、システムの負荷が低い時間帯に消費電力を抑えるようにシステムを設定できます。

Intel Node Manager は、標準の *高度な設定および電源管理* を介して、*オペレーティングシステム主導の設定と電源管理* (OSPM) を使用して CPU パフォーマンスを調整します。Intel Node Manager が OSPM ドライバーに T ステートへの変更を通知すると、ドライバーは、プロセッサーの P ステートに対応する変更を行います。同様に、Intel Node Manager が OSPM ドライバーに P ステートへの変更を通知すると、ドライバーはそれに応じて T ステートを変更します。これらの変更は自動的に行われ、オペレーティングシステムからの入力は必要ありません。管理者は、*Intel Data Center Manager* (DCM) ソフトウェアを使用して Intel Node Manager を設定および監視します。

3.11. 強化されたグラフィック電源管理

Red Hat Enterprise Linux 7 は、不要な消費の原因をいくつか排除することで、グラフィックスおよびディスプレイデバイスの電力を節約します。

LVDS リクロッキング

LVDS (*低電圧差動信号*) は、銅線を介して電子信号を伝送するシステムです。このシステムの重要なアプリケーションの 1 つは、ノートブックコンピューターの LCD (*liquid crystal display*) 画面にピクセル情報を送信することです。すべてのディスプレイには *リフレッシュレート* があります。これは、グラフィックコントローラーから新しいデータを受け取り、画面にイメージを再描画するレートです。通常、画面は 1 秒間に 60 回 (60 Hz の周波数) 新しいデータを受信します。画面とグラフィックスコントローラーが LVDS によってリンクされている場合、LVDS システムはリフレッシュサイクルごとに電源を使用します。アイドル状態の場合は、多くの LCD 画面のリフレッシュレートを 30 Hz に落としても、目立った影響はありません (リフレッシュレートの低下によって特徴的なちらつきが発生する *ブラウン管* (CRT) モニターとは異なります)。Red Hat Enterprise Linux 7 で使用されるカーネルに組み込まれている Intel グラフィックスアダプターのドライバーは、この *ダウンクロック* を自動的に実行し、画面がアイドル状態のときに約 0.5 W 節約します。

メモリーのセルフリフレッシュを有効にする

同期ダイナミックランダムアクセスメモリー (SDRAM) は、グラフィックスアダプターのビデオメモリーに使用され、1 秒あたり何千回も再充電されるため、個々のメモリーセルはそこに保存されているデータを保持します。メモリーに出入りするデータを管理するという主な機能とは別に、メモリーコントローラーは通常、これらのリフレッシュサイクルを開始します。ただし、SDRAM には低電力の *セルフリフレッシュ* モードもあります。このモードでは、メモリーは内部タイマーを使用して独自のリフレッシュサイクルを生成します。これにより、現在メモリーに保持されているデータを危険にさらすことなく、システムがメモリーコントローラーをシャットダウンできます。Red Hat Enterprise Linux 7 で使用されるカーネルは、Intel グラフィックスアダプターがアイドル状態のときにメモリーのセルフリフレッシュをトリガーできるため、約 0.8 W 節約できます。

GPU クロックの削減

一般的なGPU (graphical processing units) には、内部回路のさまざまな部分を制御する内部クロックが含まれています。Red Hat Enterprise Linux 7 で使用されるカーネルは、Intel および ATI GPU の一部の内部クロックの周波数を下げることができます。GPU コンポーネントが特定の時間内に実行するサイクル数を減らすと、実行する必要のないサイクルで消費される電力を節約できます。カーネルは、GPU がアイドル状態のときにこれらのクロックの速度を自動的に下げ、GPU のアクティビティーが増加すると速度を上げます。GPU クロックサイクルを減らすと、最大 5 W 節約できます。

GPU パワーダウン

Red Hat Enterprise Linux 7 の Intel および ATI グラフィックスドライバーは、アダプターにモニターが接続されていないことを検出できるため、GPU を完全にシャットダウンします。この機能は、モニターが定期的に接続されていないサーバーにとって特に重要です。

3.12. RFKILL

多くのコンピューターシステムには、Wi-Fi、Bluetooth、3G デバイスなどの無線送信機が含まれています。これらのデバイスは電力を消費し、デバイスが使用されていないときに無駄になります。

RFKill は Linux カーネルのサブシステムであり、コンピューターシステム内の無線送信機をクエリー、アクティブ化、および非アクティブ化できるインターフェイスを提供します。送信機が非アクティブ化されると、ソフトウェアがそれらを反応できる状態 (*ソフトブロック*) またはソフトウェアが反応できない状態 (*ハードブロック*) に置かれる可能性があります。

RFKill コアは、サブシステム用のアプリケーションプログラミングインターフェイス (API) を提供します。RFkill をサポートするように設計されたカーネルドライバーは、この API を使用してカーネルに登録し、デバイスを有効または無効にするメソッドを含めます。さらに、RFKill コアは、ユーザーアプリケーションが解釈できる通知と、ユーザーアプリケーションが送信機の状態をクエリーする方法を提供します。

RFKill インターフェイスは **/dev/rfkill** にあり、システム上のすべての無線送信機の現在の状態が含まれています。各デバイスの現在の RFKill 状態は **sysfs** に登録されています。さらに、RFKill は、RFKill 対応デバイスの状態が変化するたびに *uevent* を発行します。

Rfkill は、システム上の RFKill 対応デバイスを照会および変更できるコマンドラインツールです。ツールを入手するには、rfkill パッケージをインストールします。

コマンド **rfkill list** を使用してデバイスのリストを取得します。各デバイスには **0** から始まる *インデックス番号* が関連付けられています。このインデックス番号を使用して、rfkill にデバイスをブロックまたはブロック解除するように指示できます。次に例を示します。

```
~]# rfkill block 0
```

システム上の最初の RFKill 対応デバイスをブロックします。

rfkill を使用して、特定のカテゴリーのデバイス、またはすべての RFKill 対応デバイスをブロックすることもできます。以下に例を示します。

```
~]# rfkill block wifi
```

システム上のすべての Wi-Fi デバイスをブロックします。すべての RFKill 対応デバイスをブロックするには、次を実行します。

```
~]# rfkill block all
```

デバイスのブロックを解除するには、rfkill block の代わりに rfkill unblock を実行します。rfkill がブロックできるデバイスカテゴリーの完全なリストを取得するには、rfkill help を実行します。

第4章 ユースケース

この章では、このガイドの他の場所で説明されている分析方法と設定方法を説明するために、2 種類のユースケースについて説明します。最初の例は典型的なサーバーを考慮しており、2 番目の例は典型的なラップトップです。

4.1. 例 – サーバー

最近の典型的な標準サーバーには、Red Hat Enterprise Linux 7 でサポートされている必要なハードウェア機能の大部分が付属しています。最初に考慮すべきことは、サーバーが主に使用されるワークロードの種類です。この情報に基づいて、節電のために最適化できるコンポーネントを決定できます。

サーバーの種類に関係なく、一般的にグラフィックスパフォーマンスは必要ありません。したがって、GPU の省電力はオンのままにしておくことができます。

Webserver

Web サーバーには、ネットワークとディスク I/O が必要です。外部接続速度によっては、100 Mbit/s で十分な場合があります。マシンが主に静的ページを提供する場合、CPU パフォーマンスはそれほど重要ではない可能性があります。したがって、電源管理の選択肢には次のようなものがあります。

- tuned 用のディスクまたはネットワークプラグインはありません。
- ALPM がオンになりました。
- オンデマンド ガバナーがオンになりました。
- ネットワークカードは 100 Mbit/s に制限されています。

コンピュートサーバー

コンピュートサーバーは主に CPU を必要とします。電源管理の選択肢には次のものがあります。

- ジョブとデータの保存場所に応じて、ディスクまたはネットワークのプラグインが 調整されます。または、バッチモードシステムの場合は、完全にアクティブな tuned。
- 使用状況によっては、パフォーマンス ガバナーが使用される可能性があります。

メールサーバー

メールサーバーは主にディスク I/O と CPU を必要とします。電源管理の選択肢には次のものがあります。

- CPU パフォーマンスの最後の数パーセントは重要ではないため、オンデマンド ガバナはオンになっています。
- tuned 用のディスクまたはネットワークプラグインはありません。
- メールは内部にあることが多く、1 Gbit/s または 10 Gbit/s リンクの恩恵を受けることができるため、ネットワーク速度は制限されるべきではありません。

ファイルサーバー

ファイルサーバーの要件はメールサーバーの要件と似ていますが、使用するプロトコルによっては、より多くの CPU パフォーマンスが必要になる場合があります。通常、Samba ベースのサーバーは NFS より多くの CPU を必要とし、NFS は通常 iSCSI よりも多くの CPU を必要とします。それでも、オンデマンド ガバナーを使用できるはずです。

Directory server

特に十分な RAM が装備されている場合、通常、Directory Server はディスク I/O の要件が低くなります。ネットワーク I/O はそれほど重要ではありませんが、ネットワーク遅延は重要です。より遅いリンク速度での遅延ネットワークチューニングを検討することもできますが、特定のネットワークについてこれを慎重にテストする必要があります。

4.2. 例 – ラップトップ

電源管理と節電が実際で差がつく非常に一般的なものとしては、他にもラップトップがあります。通常、ラップトップは設計上、ワークステーションやサーバーよりも消費電力が大幅に少ないため、絶対的な節約の可能性は他のマシンよりも低くなります。ただし、バッテリーモードの場合は、ラップトップのバッテリー寿命を数分延ばすのに少しでも節約できます。このセクションでは、バッテリーモードのラップトップに焦点を当てていますが、AC 電源で実行している間も、これらのチューニングの一部またはすべてを使用することができます。

通常、単一コンポーネントの節約は、ワークステーションよりもラップトップの方が相対的に大きな違いを生みます。たとえば、100 Mbits/s で動作する 1 Gbit/s ネットワークインターフェイスは、約 3-4 ワットを節約します。総消費電力が約 400 ワットの一般的なサーバーの場合、この節約は約 1 % です。総消費電力が約 40 ワットのラップトップでは、この 1 つのコンポーネントだけで全体の 10 % の電力を節約できます。

一般的なラップトップでの具体的な省電力の最適化には、次のようなものがあります。

- システム BIOS を設定して、使用しないすべてのハードウェアを無効にします。たとえば、パラレルポートまたはシリアルポート、カードリーダー、Web カメラ、WiFi、Bluetooth などは、考えられる候補をいくつか挙げただけです。
- 画面を快適に読むために完全な照明を必要としない暗い環境では、ディスプレイを暗くします。GNOME デスクトップでは システム+環境設定の → 電源管理 を使用し、KDE デスクトップでは キックオフアプリケーションランチャー+コンピューター+システム設定の+詳細な → 電源管理 使用します。または、コマンドラインで gnome-power-manager または xbacklight を使用します。またはラップトップのファンクションキー。

さらに (または代わりに)、さまざまなシステム設定に対して多くの小さな調整を実行できます。

- オンデマンド ガバナーを使用します (Red Hat Enterprise Linux 7 ではデフォルトで有効になります)
- AC97 オーディオ省電力を有効にします (Red Hat Enterprise Linux 7 ではデフォルトで有効になっています):

```
~]# echo Y > /sys/module/snd_ac97_codec/parameters/power_save
```

- USB 自動サスペンドを有効にします。

```
~]# for i in /sys/bus/usb/devices/*/power/autosuspend; do echo 1 > $i; done
```

USB 自動サスペンドは、すべての USB デバイスで正しく機能しないことに注意してください。

- relatime を使用してファイルシステムをマウントします (Red Hat Enterprise Linux 7 のデフォルト):

```
~]# mount -o remount,relatime mountpoint
```

- **画面の明るさを 50 以下に下げます。例:**

```
~]$ xbacklight -set 50
```

- **画面アイドル状態の DPMS を有効にします。**

```
~]$ xset +dpms; xset dpms 0 0 300
```

- **Wi-Fi を無効にします。**

```
~]# echo 1 > /sys/bus/pci/devices/*/rf_kill
```

付録A 開発者向けのヒント

優れたプログラミングの参考書はすべて、メモリー割り当てと特定の関数のパフォーマンスに関する問題をカバーしています。ソフトウェアを開発するときは、ソフトウェアが実行するシステムの電力消費を増加させる可能性がある問題に注意してください。これらの考慮事項はすべてのコード行に影響するわけではありませんが、パフォーマンスのボトルネックとなることが多い領域でコードを最適化できます。

問題となることが多いテクニックには、次のようなものがあります。

- スレッドを使用します。
- 不必要な CPU ウェイクアップが発生し、ウェイクアップを効率的に使用していません。起動しないといけない場合は、(アイドリングに対して) すべてを一度に、できるだけ早く行います。
- を使用して[f]sync()不必要に。
- 不必要なアクティブポーリングまたは短い定期的なタイムアウトの使用。(代わりにイベントに反応します)。
- ウェイクアップを効率的に使用していません。
- 非効率的なディスクアクセス。頻繁なディスクアクセスを避けるために、大きなバッファーを使用します。一度に 1 つの大きなブロックを書き込みます。
- タイマーの非効率的な使用。可能であれば、アプリケーション全体 (またはシステム全体) でタイマーをグループ化します。
- 過剰な I/O、電力消費、またはメモリー使用量 (メモリーリークを含む)
- 不要な計算を行っています。

以下のセクションでは、これらの領域のいくつかをより詳細に検討します。

A.1. スレッドの使用

スレッドを使用するとアプリケーションのパフォーマンスが向上し、高速になると広く信じられていますが、すべての場合に当てはまるわけではありません。

Python

Python は Global Lock Interpreter を使用します[1]であるため、スレッド化は大規模な I/O 操作でのみ有効です。Unladen-swallow [2] は、コードを最適化できる Python のより高速な実装です。

Perl

Perl スレッドはもともと、フォークしないシステム (32 ビット Windows オペレーティングシステムのシステムなど) で実行するアプリケーション用に作成されました。Perl スレッドでは、データはスレッドごとにコピーされます (Copy On Write)。ユーザーはデータ共有のレベルを定義できる必要があるため、デフォルトではデータは共有されません。データを共有するには、threads::shared モジュールを含める必要があります。ただし、データがコピーされるだけでなく (Copy On Write)、モジュールはデータに関連付けられた変数も作成します。これにはさらに時間がかかり、さらに遅くなります。[3]

C

C スレッドは同じメモリーを共有し、各スレッドには独自のスタックがあり、カーネルは新しいファイル記述子を作成して新しいメモリー空間を割り当てる必要はありません。C は、より多くのスレッドに

対してより多くの CPU のサポートを実際に使用できます。したがって、スレッドのパフォーマンスを最大化するには、C や C++ などの低水準言語を使用してください。スクリプト言語を使用する場合は、C バインディングを作成することを検討してください。プロファイラーを使用して、コードのパフォーマンスが低下している部分を特定します。[4]

A.2. ウェイクアップ

多くのアプリケーションは、設定ファイルの変更をスキャンします。多くの場合、スキャンは一定の間隔で、たとえば 1分ごとに実行されます。ディスクがスピンダウンから強制的にウェイクアップするため、これは問題になる可能性があります。最善の解決策は、適切な間隔、適切なチェックメカニズムを見つけるか、inotify で変更をチェックしてイベントに反応することです。Inotify は、 ファイルまたはディレクトリーに対するさまざまな変更をチェックできます。

以下に例を示します。

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>
#include <sys/types.h>
#include <sys/inotify.h>
#include <unistd.h>

int main(int argc, char *argv[]) {
  int fd;
  int wd;
  int retval;
  struct timeval tv;

  fd = inotify_init();

  /* checking modification of a file - writing into */
  wd = inotify_add_watch(fd, "./myConfig", IN_MODIFY);
  if (wd < 0) {
    printf("inotify cannot be used\n");
    /* switch back to previous checking */
  }

  fd_set rfds;
  FD_ZERO(&rfds);
  FD_SET(fd, &rfds);
  tv.tv_sec = 5;
  tv.tv_usec = 0;
  retval = select(fd + 1, &rfds, NULL, NULL, &tv);
  if (retval == -1)
    perror("select()");
  else if (retval) {
    printf("file was modified\n");
  }
  else
    printf("timeout\n");

  return EXIT_SUCCESS;
}
```

このアプローチの利点は、実行できるさまざまなチェックです。

主な制限は、システムで使用できる監視の数が限られていることです。この番号は /proc/sys/fs/inotify/max_user_watches から取得でき、変更することもできますが、これは推奨できません。さらに、inotify が失敗した場合、コードは別のチェックメソッドにフォールバックする必要があります。これは通常、次のような問題が多数発生することを意味します。#if #defineソースコード内で。

inotify の詳細については、inotify(7)マニュアルページ。

A.3. FSYNC

Fsync は I/O 負荷の高い操作として知られていますが、これは完全に真実ではありません。

Firefox は、ユーザーがリンクをクリックして新しいページに移動するたびに、sqlite ライブラリーを呼び出していました。Sqlite が呼び出すfsyncまた、ファイルシステム設定 (主にデータ順序モードの ext3) が原因で、何も起こらないときに長い遅延が発生しました。別のプロセスが同時に大きなファイルをコピーしていた場合、これには長時間 (最大 30 秒) かかることがあります。

ただし、他の場合では、fsyncまったく使用されていなかったため、ext4 ファイルシステムへの切り替えで問題が発生しました。Ext3 はデータ順モードに設定され、数秒ごとにメモリーをフラッシュしてディスクに保存しました。ただし、ext4 と laptop_mode では、保存の間隔が長くなり、システムが予期せずオフになったときにデータが失われる可能性がありました。現在、ext4 にはパッチが適用されていますが、アプリケーションの設計を慎重に検討し、使用する必要があります。fsync適切に。

次の設定ファイルの読み取りと書き込みの簡単な例は、ファイルのバックアップを作成する方法またはデータを失う方法を示しています。

```
/* open and read configuration file e.g. ./myconfig */
fd = open("./myconfig", O_RDONLY);
read(fd, myconfig_buf, sizeof(myconfig_buf));
close(fd);
...
fd = open("./myconfig", O_WRONLY | O_TRUNC | O_CREAT, S_IRUSR | S_IWUSR);
write(fd, myconfig_buf, sizeof(myconfig_buf));
close(fd);
```

より良いアプローチは次のとおりです。

```
/* open and read configuration file e.g. ./myconfig */
fd = open("./myconfig", O_RDONLY);
read(fd, myconfig_buf, sizeof(myconfig_buf));
close(fd);
...
fd = open("./myconfig.suffix", O_WRONLY | O_TRUNC | O_CREAT, S_IRUSR | S_IWUSR
write(fd, myconfig_buf, sizeof(myconfig_buf));
fsync(fd); /* paranoia - optional */
...
close(fd);
rename("./myconfig", "./myconfig~"); /* paranoia - optional */
rename("./myconfig.suffix", "./myconfig");
```

[1] http://docs.python.org/c-api/init.html#thread-state-and-the-global-interpreter-lock

[2] http://code.google.com/p/unladen-swallow/

[3] http://www.perlmonks.org/?node_id=288022

[4] http://people.redhat.com/drepper/lt2009.pdf

付録B 更新履歴

改訂 2.2-9 Mon Aug 05 2019 Marie Doleželová
7.7 GA 公開用ドキュメントバージョン

改訂 2.2-6 Mon Jul 24 2017 Marie Doleželová
7.4 GA 公開用ドキュメントバージョン

改訂 2.2-5 Tue Mar 21 2017 Milan Navrátil
非同期更新: Tuned 章の書き直し

改訂 2.0-2 Fri Oct 14 2016 Marie Doleželová
7.3 GA リリースのバージョン

改訂 2.0-1 Wed 11 Nov 2015 Jana Heves
7.2 GA リリース向けのバージョン。

改訂 1-3 Fri 19 Jun 2015 Jacquelynn East
Core Infrastructure と Mechanics の誤ったパッケージ名を修正しました。

改訂 1-2 Wed 18 Feb 2015 Jacquelynn East
7.1 GA 向けバージョン

改訂 1-1 Thu Dec 4 2014 Jacquelynn East
7.1 ベータ版バージョン

改訂 1.0-9 Tue Jun 9 2014 Yoana Ruseva
7.0 GA リリース向けバージョン

改訂 0.9-1 Fri May 9 2014 Yoana Ruseva
スタイル変更に伴う再構築

改訂 0.9-0 Wed May 7 2014 Yoana Ruseva
レビュー用の Red Hat Enterprise Linux 7.0 リリース。

改訂 0.1-1 Thu Jan 17 2013 Jack Reed
ドキュメントの Red Hat Enterprise Linux 6 バージョンからのブランチ