



Red Hat Enterprise Linux 7

電力管理ガイド

Red Hat Enterprise Linux 7 での電力消費量の管理

Red Hat Enterprise Linux 7 電力管理ガイド

Red Hat Enterprise Linux 7 での電力消費量の管理

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

Copyright © 2017 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

本ガイドでは、Red Hat Enterprise Linux 7 システムで効果的に電力消費量を管理する方法について説明します。以下のセクションでは、電力消費量を削減するさまざまな技術 (サーバー向けとノート PC 向けの両方) と、その各技術がどのようにシステムの全体的なパフォーマンスに影響を与えるかについて説明します。

目次

| | |
|--|-----------|
| 第1章 概要 | 3 |
| 1.1. 電力管理の重要性 | 3 |
| 1.2. 電力管理の基礎 | 4 |
| 第2章 電力管理の監査と分析 | 6 |
| 2.1. 監査および分析の概要 | 6 |
| 2.2. POWERTOP | 6 |
| 2.3. DISKDEVSTAT と NETDEVSTAT | 8 |
| 2.4. BATTERY LIFE TOOL KIT | 11 |
| 2.5. TUNED | 13 |
| 2.6. UPOWER | 14 |
| 2.7. GNOME の電源管理 | 14 |
| 2.8. 他の監査ツール | 15 |
| 第3章 中核となるインフラストラクチャとメカニズム | 16 |
| 3.1. CPU のアイドル状態 | 16 |
| 3.2. CPUFREQ | 16 |
| 3.3. CPU 監視機能 | 20 |
| 3.4. CPU 省電力ポリシー | 20 |
| 3.5. サスペンドと復帰 | 21 |
| 3.6. 実行時デバイス電源管理 | 21 |
| 3.7. ACTIVE-STATE POWER MANAGEMENT | 22 |
| 3.8. AGGRESSIVE LINK POWER MANAGEMENT | 23 |
| 3.9. RELATIME ドライブアクセス最適化 | 24 |
| 3.10. パワーキャッピング (POWER CAPPING) | 24 |
| 3.11. 拡張グラフィックス電力管理 | 25 |
| 3.12. RFKILL | 26 |
| 第4章 使用例 | 28 |
| 4.1. 例: サーバー | 28 |
| 4.2. 例: ノート PC | 29 |
| 付録A 開発者へのヒント | 31 |
| A.1. スレッドの使用 | 31 |
| A.2. ウェイクアップ | 32 |
| A.3. FSYNC | 33 |
| 付録B 改訂履歴 | 35 |

第1章 概要

Red Hat Enterprise Linux 7 の重要な改善点の1つが電力管理です。コンピューターシステムで使用する電力を制限することは、グリーンIT (環境に優しいコンピューティング) の最も重要な側面の1つです。このグリーンITでは、リサイクル可能な資源の利用、ハードウェアの製造が環境に及ぼす影響、システム設計と導入における環境意識などについても考慮されます。本ガイドでは、Red Hat Enterprise Linux 7 が稼働するシステムの電力管理について説明します。

1.1. 電力管理の重要性

電力管理の中核は、各システムコンポーネントによるエネルギーの消費をいかに効果的に最適化するかを理解するところにあります。その場合は、システムによって行われるさまざまなタスクを調査し、そのパフォーマンスがジョブに対して最適になるよう各コンポーネントを設定する必要があります。

電力管理を行う主な要因を以下に示します。

- コスト削減のための全体的な消費電力量の抑制

電力管理を適切に活用すると、以下のような結果が得られます。

- サーバーおよびコンピューティングセンターの熱の抑制
- 冷却、空間、ケーブル、発電機、無停電電源装置 (UPS) などにかかる 2 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低減
- エナジースター (Energy Star) などグリーン IT に関する政府規則または法的要件の準拠
- 新システムにおける企業のガイドラインの準拠

通常は、特定のコンポーネント (またはシステム全体) の電力消費を抑制すると、発生熱量が低下し、パフォーマンスも低下します。したがって、特にミッションクリティカルなシステムの場合は、設定によるパフォーマンスの低下について十分な調査と検証を行ってください。

システムで行われるさまざまなタスクを調査し、パフォーマンスがジョブに最適になるよう各コンポーネントを設定することにより、エネルギーを節約し、発生熱を抑制して、ノートパソコンのバッテリー寿命を最適化できます。電力消費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、パフォーマンスのチューニングの原則と似ています。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対して最適化されるため、電力管理とパフォーマンスのチューニングは、ある意味、システムの設定と相反します。本ガイドでは、電源管理を行う上で役に立つ Red Hat 提供のツールと、Red Hat で開発された方法について説明します。

すでに Red Hat Enterprise Linux 7 には多くの新しい電力管理機能が含まれていて、デフォルトで有効になっています。これらの機能はすべて、サーバーまたはデスクトップの標準的な使用でパフォーマンスに影響を及ぼさ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、最小限のレイテンシー、または最大の CPU パフォーマンスが必須である非常に特殊なユースケースでは、これらのデフォルト値を再検討する必要がある場合があります。

以下の質問とその答えをお読みいただいた上で、本ガイドで説明している方法を使用してマシンを最適化すべきかどうかの判断を行ってください。

問： マシンを最適化すべきですか？

答： 電力を最適化する重要性は、会社で従うべきガイドラインがあるか、または順守すべき規則があるかによって異なります。

問： どの程度、最適化する必要がありますか？

答： ここで説明する複数の方法では、マシンを詳しく監査および分析する必要がなく、代わりに電力使用を一般的に改善する一連の汎用的な最適化が提供されます。これらの最適化は、手作業で行うシステムの監査および最適化ほど優れていませんが、役に立ちます。

問： 最適化によりシステムパフォーマンスが許容範囲を下回るレベルに低下しませんか？

答： 本ガイドで説明しているほとんどの方法により、システムパフォーマンスは明らかな影響を受けません。Red Hat Enterprise Linux 7 のデフォルト設定を越えた電力管理を実装する場合は、電力最適化の実行後にシステムパフォーマンスを監視し、パフォーマンスの低下が許容範囲内であるか確認する必要があります。

問： システムの最適化に費やす時間とリソースの負担が、得られる効果を上回りませんか？

答： 通常は、全プロセスに従って1台のシステムを最適化することは、費される時間とコストの負担が1台のマシンのライフタイムで得られる効果を大幅に上回るため、意味がありません。その一方で、たとえば、同じ設定とセットアップを使用して1万台のデスクトップシステムをオフィスに導入するときに、最適化されたセットアップを1つ作成し、1万台すべてのマシンに適用することは、多くの場合、効果的です。

次のセクションでは、最適なハードウェアのパフォーマンスがエネルギー消費の観点からどのようにシステムに恩恵をもたらすのかについて説明します。

1.2. 電力管理の基礎

効率的な電力管理は以下の原則に基づきます。

アイドル状態の CPU は必要な時にウェイクアップする

Red Hat Enterprise Linux 6 以降、カーネルは、ティックレス (*tickless*) で実行されます。つまり、以前の定期タイマーの割り込みが、オンデマンド型の割り込みに置き換えられました。したがって、アイドル状態の CPU を、新しいタスクが処理のためにキューに格納されるまでアイドル状態に維持し、低電力の状態にある CPU をより長くその状態に維持することができます。ただし、システムのアプリケーションが不必要なタイマーイベントを作成する場合は、この機能の利点が打ち消されることがあります。このようなイベントの例としては、ボリュームの変更またはマウスの移動のチェックなどのポーリングイベントがあります。

Red Hat Enterprise Linux 7 には、CPU 使用率に基いてアプリケーションを識別および監査できるツールが同梱されています。詳細については、「[2章 電力管理の監査と分析](#)」を参照してください。

使用していないハードウェアとデバイスを完全に無効にする

これは、可動パーツを持つデバイス (たとえば、ハードディスク) の場合に特に当てはまります。また、一部のアプリケーションでは、使用していないが有効なデバイスが「オープン」な状態のままになることがあります。この状況では、カーネルはデバイスが使用中であると見なし、デバイスが省電力状態になることが阻止される場合があります。

動作が少ないということは消費電力も少ない

ただし多くの場合、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。現在 Red Hat Enterprise Linux 7 でサポートされる新しい機能の一部は、多くの場合、従来のシステムコンポーネントではサポートされません。システムに最新の公式ファームウェアが使用されていることと、BIOS の電力管理またはデバイス設定のセクションで電力管理の機能が有効になっていることを確認してください。確認する機能は以下のとおりです。

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

上記の機能がハードウェアでサポートされ、BIOS で有効な場合、Red Hat Enterprise Linux 7 ではその機能がデフォルトで使用されます。

CPU の各種状態とその効果

ACPI (電力制御インタフェース: *Advanced Configuration and Power Interface*) を使用する最新の CPU は、以下の 3 つの電力状態を提供します。

- Sleep (C 状態)
- Frequency and voltage (P 状態)

P 状態はプロセッサの周波数および電圧動作点を表し、共に P 状態が増加すると変動します。

- Heat output (T 状態または「温度状態」)

最小のスリープ状態で稼働している CPU は、最小のワット数を消費しますが、必要なときにこの状態からウェイクアップするにはしばらく時間がかかります。非常に稀ですが、これにより CPU がスリープ状態になる度に CPU をすぐにウェイクアップする必要がある場合があります。この場合は、実質的に CPU が常にビジー状態になり、省電力の一部が失われます (別の状態が使用された場合)。

電源がオフになっているマシンの消費電力は最小となる

当たり前かもしれませんが、実際に節電を行う最善策の 1 つは、システムの電源をオフにすることです。たとえば、「グリーン IT」を意識する企業文化を育み、昼休みや帰宅時にマシンの電源をオフにするガイドラインを策定します。また、数台の物理サーバーを大きなサーバー 1 台に統合し、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 電力管理の監査と分析

2.1. 監査および分析の概要

通常は、1台のシステムを手動で詳細に監査、分析、およびチューニングすることは、そのような作業にかかる時間とコストの負担がそのシステムチューニングから得られる効果を上回るため、例外的な行為になります。ただし、ほぼ同一の大量のマシンがあり、すべてのシステムに同じ設定を再利用できる場合は、この作業を一度だけ行うことが非常に役に立つことがあります。たとえば、数千に及ぶデスクトップシステム(マシンほぼ同一の HPC クラスタなど)の導入を考えてください。監査と分析を行う別の理由は、将来にシステムの動作の劣化や変化を特定できるように比較の基礎を提供することです。この分析結果は、ハードウェア、BIOS、またはソフトウェアの定期更新で電力消費に関する問題を避けたい場合に非常に役立ちます。一般的に、監査と分析を十分に行うと、特定のシステムで実際に発生していることを把握できるようになります。

電力消費に関する監査と分析は、最新システムを使用しても比較的難しいものです。ほとんどのシステムでは、ソフトウェアを使用して電力使用量を測定できません。ただし、例外はあります。Hewlett Packard サーバシステムの ILO 管理コンソールには、ウェブ経由でアクセスできる電力管理モジュールが含まれます。IBM は、BladeCenter 電力管理モジュールで同様のソリューションを提供しています。一部の Dell システムでも、IT Assistant 機能により電力監視機能が提供されています。他のベンダーはサーバプラットフォーム向けに類似の機能を提供している可能性があります。すべてのベンダーで対応しているソリューションは存在しません。

多くの場合、電力消費の直接的な測定は、省電力量を最大化するためにのみ必要です。変更が反映されているか、システムがどのように動作しているかを確認するには、他の手段があります。この章では、そのような必須ツールについて説明します。

2.2. POWERTOP

Red Hat Enterprise Linux 7 ではティックレスカーネルが導入されたため、CPU がより頻繁にアイドル状態になるようになり、電力消費量が抑えられ、電力管理が向上しました。**PowerTOP** ツールは、CPU を頻繁にウェイクアップするカーネルとユーザースペースアプリケーションの特定のコンポーネントを識別します。**PowerTOP** は、監査を行うために開発段階で使用されていました。これにより、このリリースで多くのアプリケーションのチューニングが行われ、不要に CPU がウェイクアップする率が約 10 分の 1 に低下しました。

Red Hat Enterprise Linux 7 には、バージョン 2.x の **PowerTOP** が含まれます。このバージョンは、1.x コードベースの完全な書き換えであり、分かりやすいタブベースのユーザーインターフェースを備え、カーネルの「perf」インフラストラクチャーを広範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力動作が追跡され、明確に表示されるため、問題を迅速に特定です。試験的に、2.x コードベースには、個別のデバイスおよびプロセスが消費している電力を示すことができる電力予測エンジンが含まれています。[図2.1「実行中の PowerTOP」](#)を参照してください。

PowerTOP をインストールするには、**root** で以下のコマンドを実行します。

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

PowerTOP を実行するには、**root** で以下のコマンドを実行します。

```
~]# powertop
```

PowerTOP はシステム全体の電力使用量の予測を提供し、各プロセス、デバイス、カーネル作業、タイマー、割り込みハンドラーの電力使用量を表示することができます。このタスク中は、ノート PC をバッテリー電源で稼働してください。電力予測エンジンを調整するには、**root** で以下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され、輝度レベルおよびスイッチデバイスのオンとオフが繰り返されます。調整中はマシンに触れないでください。調整プロセスが終わると、**PowerTOP** が正常に開始されます。データを収集するために約1時間稼働させます。十分なデータが収集されると、最初のコラムに電力予測の数字が表示されます。

ノート PC でこのコマンドを実行する場合は、バッテリー電源で稼働することで利用可能なデータすべてが提供されます。

PowerTOP は実行中にシステムから統計数字を収集します。**Overview** タブでは、CPU にウェイクアップを最も頻繁に送信するコンポーネントまたは最も電力を消費しているコンポーネントのリストが表示されます (図2.1「実行中の PowerTOP」を参照)。その横のコラムでは、電力消費予測、リソースの使用法、1秒あたりのウェイクアップ、プロセス、デバイス、タイマーなどコンポーネントの分類、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は、サービスや、カーネルのデバイスおよびドライバーのパフォーマンスの効率性を示します。ウェイクアップが少ないと消費電力も少ないこととなります。コンポーネントは、電力使用量の最適化がさらに実行可能な度合いで並んでいます。

ドライバーコンポーネントのチューニングは通常、カーネルの変更を必要とし、本ガイドの対象外となります。ただし、ウェイクアップを送信するユーザスペースのプロセスは、管理がより簡単です。最初に該当するサービスまたはアプリケーションをこのシステム上で実行する必要があるかどうかを判断します。必要ない場合は、そのサービスまたはアプリケーションを単に非アクティブ化します。古い System V サービスを永続的に無効にするには、以下のコマンドを実行します。

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

このプロセスについてより詳細な情報を得るには、**root** で以下のコマンドを実行します。

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

トレースが繰り返し行われているように見える場合は、恐らくビジーループが発生しています。このようなバグを修復するには通常、そのコンポーネントでコードを変更する必要があります。

図2.1「実行中の PowerTOP」では、消費電力量の合計と、該当する場合はバッテリーの残量が表示されます。これらの中には、1秒あたりのウェイクアップ合計、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の概要があります。画面の残りに、使用量にしたがってプロセス、割り込み、デバイス、およびリソースの一覧が表示されます。適切に調整されると、一覧の各アイテムの最初のコラムに電力消費量の予測も表示されます。

タブを移動するには、**Tab** および **Shift+Tab** キーを使用します。**Idle stats** タブでは、すべてのプロセッサおよびコアの C 状態の使用が表示されます。**Frequency stats** タブでは、Turbo モード (該当する場合) を含む全プロセッサおよびコアの P 状態の使用が表示されます。CPU がより高い C または P 状態に長くいればいるほど、よいこととなります (**C4** の方が **C3** よりも高い)。これは、CPU 使用率がどの程度うまく最適化されているかを示す指標となります。システムのアイドル中の理想状態は、最高の C または P 状態が 90% 以上を維持していることです。

Device Stats タブは **Overview** タブと同様の情報を表示しますが、デバイスに限定されます。

Tunables タブには、システムの消費電力量を低減させるための提案が含まれています。**up** および **down** キーを使って各提案に移動し、**enter** キーでそれらのオン/オフを切り替えます。

```

PowerTOP 2.3 Overview Idle stats Frequency stats Device stats Tunables
The battery reports a discharge rate of 16.7 W
The estimated remaining time is 1 hours, 25 minutes

Summary: 386.1 wakeups/second, 60.2 GPU ops/seconds, 0.0 VFS ops/sec and 42.9% CPU use

Power est.      Usage      Events/s  Category  Description
3.79 W          2642 rpm   Device    Laptop fan
3.39 W          53.3%     Device    Display backlight
2.63 W          172.9 ms/s 0.00     Timer     process_timeout
2.24 W          142.2 ms/s 17.8     Interrupt [9] acpi
665 mW         43.6 ms/s 27.5     Process   /usr/lib64/firefox/firefox
237 mW         10.7 ms/s 56.4     Process   /usr/lib64/seamonkey/seamonkey
144 mW         5.7 ms/s 77.2     Interrupt PS/2 Touchpad / Keyboard / Mouse
119 mW         7.8 ms/s 11.9     Process   /usr/bin/Xorg :0 -background none -verbose -auth /var/run/gdm
91.3 mW        3.7 pkts/s Device    Network interface: wlan0 (iwlwifi)
84.3 mW        5.5 ms/s 45.9     Timer     tick_sched_timer
77.3 mW        3.3 ms/s 10.1     Process   gkrellm --geometry +1608+70
72.9 mW        4.8 ms/s 20.6     Process   /usr/lib/polkit-1/polkitd --no-debug
58.9 mW        3.9 ms/s 15.0     Process   /usr/lib64/seamonkey/plugin-container /usr/lib64/flash-plugin
51.4 mW        3.4 ms/s 0.00     Interrupt [1] timer(softirq)
42.3 mW        2.6 ms/s 13.0     Process   xfce4-screenshooter
37.2 mW        2.4 ms/s 58.1     Timer     hrtimer_wakeup
33.0 mW        2.2 ms/s 6.3      Interrupt [7] sched(softirq)
31.5 mW        60.9 us/s 7.3      kWork     iwl_bg_run_time_calib_work
29.8 mW        2.0 ms/s 41.2     kWork     od_dbs_timer
28.9 mW        1.6 ms/s 1.7      Process   xfce4-panel
25.2 mW        0.9 ms/s 8.6      Process   xfwm4
21.3 mW        1.4 ms/s 0.00     Timer     delayed_work_timer_fn
16.3 mW        1.1 ms/s 0.00     Process   /bin/dbus-daemon --system --address=systemd: --nofork --nopid
13.1 mW        0.9 ms/s 0.5      Process   crond
12.4 mW        0.8 ms/s 0.00     Interrupt [0] timer/1
12.2 mW        0.8 ms/s 4.3      Interrupt [6] tasklet(softirq)
12.1 mW        0.8 ms/s 0.05     kWork     disk_events_workfn
12.0 mW        0.8 ms/s 0.00     Interrupt [0] timer/0
10.0 mW        659.2 us/s 0.4      kWork     kcryptd_crypt
10.0 mW        658.2 us/s 2.1      Process   /usr/sbin/NetworkManager --no-daemon
8.04 mW        528.0 us/s 0.05     Process   powertop
5.76 mW        347.4 us/s 1.6      Process   xchat
5.59 mW        366.9 us/s 0.00     Interrupt [9] RCU(softirq)
4.75 mW        311.5 us/s 0.00     Process   /usr/sbin/crond -n
<ESC> Exit

```

図2.1実行中の PowerTOP

PowerTOP を `--html` オプションで実行すると、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、必要な出力ファイル名に置き換えます。

```
~]# 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。 `--time` オプションを使うとこれを変更することもできます。

```
~]# 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 `turbostat` ユーティリティーと併用することもできます。 `turbostat` ユーティリティーはレポートングツールであり、Intel 64 プロセッサのプロセッサトポロジー、周波数、アイドル状態の電力状態、温度、および電力使用量を表示します。 `turbostat` ユーティリティーの詳細については、 `turbostat(8)` man ページまたは『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 と NETDEVSTAT

Diskdevstat と netdevstat は、システム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活動とネットワーク活動の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 のウェイクアップ回数を示す PowerTOP を参考にしています (「PowerTOP」を参照)。これらのツールが収集する統計により、数多くの小規模な I/O 操作で電力を浪費す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では、このような種類の使用量を特定できません。

SystemTap とともにこれらのツールをインストールするには、**root** で以下のコマンドを実行します。

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

以下のコマンドでツールを実行します。

```
~]# diskdevstat
```

あるいは、以下のコマンドを実行します。

```
~]# netdevstat
```

これら両方のコマンドには、以下のように最大3つのパラメータを使用できます。

diskdevstat *update_interval total_duration display_histogram*

netdevstat *update_interval total_duration display_histogram*

update_interval

表示が更新される秒単位の間隔。デフォルト値: **5**

total_duration

実行完了にかかる秒単位の時間。デフォルト値: **86400** (1日)

display_histogram

実行完了時に全収集データで度数分布図(柱状グラフ)を作成するかどうかを指定するフラグ。

出力は PowerTOP の出力に似ています。以下は、実行された長い **diskdevstat** の出力例です。

```
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 ologwatch
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 |

各列の内容は、以下のとおりです。

PID

アプリケーションのプロセス ID

UID

アプリケーションの実行元となるユーザー ID

DEV

I/O が発生したデバイス

WRITE_CNT

書き込み操作の合計数

WRITE_MIN

2 回の連続書き込みに要した最短時間 (秒単位)

WRITE_MAX

2 回の連続書き込みに要した最長時間 (秒単位)

WRITE_AVG

2 回の連続書き込みに要した平均時間 (秒単位)

READ_CNT

読み込み操作の合計数

READ_MIN

2 回の連続読み込みに要した最短時間 (秒単位)

READ_MAX

2 回の連続読み込みに要した最長時間 (秒単位)

READ_AVG

2 回の連続読み込みに要した平均時間 (秒単位)

COMMAND

プロセスの名前

この例には、非常に目立つアプリケーションが3つあります。

```
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
```

これらの3つのアプリケーションでは、**WRITE_CNT** が0よりも大きいため、測定中になんらかの書き込みが実行されたことを意味します。その中でも、**plasma** は他と大差をつけて一番高い数値を示しています (最多の書き込み操作が実行されるため、当然書き込み間隔の平均時間は最短となります)。したがって、電力効率の悪いアプリケーションについて懸念がある場合は、**Plasma** が最有力候補になります。

strace コマンドと **ltrace** コマンドを使用して、所定のプロセス ID のすべてのシステムコールを追跡することによりアプリケーションをさらに詳しく調べることができます。この例では、以下のコマンドを実行できます。

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

この例では、**strace** の出力には、ユーザーの KDE アイコンのキャッシュファイルを書き込みのため開き、直後にそのファイルを再び閉じるという動作が 45 秒毎に繰り返される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 (特に変更時間) が変更されたため、必要な物理的な書き込みがハードディスクに行われました。最終的な修正では、アイコンに更新が加えられなかった場合に、こうした不要な呼び出しが発生しないようになりました。

2.4. BATTERY LIFE TOOL KIT

Red Hat Enterprise Linux 7 では、バッテリーの寿命とパフォーマンスをシミュレートして解析するテストスイートである **BLTK (Battery Life Tool Kit)** が採用されています。BLTK は、このために特定のユーザーグループをシミュレートするタスクセットを実行し、その結果を報告します。ノートブック PC のパフォーマンスをテストするために特別に開発された BLTK ですが、**-a** を付けて起動すると、デスクトップコンピューターのパフォーマンスも報告できます。

BLTK を使用すると、実際にマシンを使用しているのと同程度の再現可能な作業負荷を生成できます。たとえば、**office** の作業負荷はテキストを書き込み、その中で修正を行います。同じ作業を表計算でも行います。BLTK を **PowerTOP** や他の監査または解析ツールなどと併用することで、マシンがアイドル状態の時だけでなく頻繁に使用されている時にも、実行した最適化に効果があるかどうかを検証できます。全く同じ作業負荷を異なる設定で複数回実行できるため、異なる設定の結果を比較することができます。

以下のコマンドを使用して、BLTK をインストールします。

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

以下のコマンドを使用して、BLTK を実行します。

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

たとえば、**idle** の作業負荷を 120 秒間実行するには、以下のコマンドを実行します。

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

デフォルトで使用できる作業負荷は次のとおりです。

-l, --idle

システムがアイドル状態です。他の作業負荷と比較する場合に基準値として使用します。

-R, --reader

ドキュメントを読み込むシミュレートを行います (デフォルトでは、**Firefox** を使用)。

-P, --player

CD または DVD ドライブの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 スイートを使ったドキュメント編集のシミュレートを行います。

指定できる他のオプションは以下のとおりです。

-a, --ac-ignore

AC 電源が使用可能かどうか無視します (デスクトップで必要)。

-T *number_of_seconds*, --time *number_of_seconds*

テストを実行する期間 (秒単位) ; **idle** 作業負荷を使用してこのオプションを使用します。

-F *filename*, --file *filename*

特定の作業負荷で使用されるファイル (たとえば、CD または DVD ドライブにアクセスする代わりに、**player** の作業負荷で再生するファイル) を指定します。

-W *application*, --prog *application*

特定の作業負荷で使用されるアプリケーション (たとえば、**reader** の作業負荷向けの **Firefox** 以外のブラウザ) を指定します。

BLTK は、数多くの特別なオプションをサポートします。詳細は、man ページの **bltk** を参照してください。

BLTK は、生成する結果を **/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.number/**) に保存します。たとえば、**~/bltk/reader.results.002/** ディレクトリーには **reader** の作業負荷の 3 つ目のテスト結果が保持されます (1 つ目のテストは番号なし)。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取りやすい形式にまとめるには、以下のコマンドを実行します。

```
~]$ bltk_report path_to_results_directory
```

結果が、結果ディレクトリーの **Report** という名前のテキストファイルに表示されます。代わりにターミナルエミュレーターで結果を閲覧するには、**-o** オプションを使用します。

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


2.5. TUNED

Tuned はプロファイルベースのシステムチューニングツールで、**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します。動的チューニングは実験的な機能で、Red Hat Enterprise Linux 7 のデフォルトでは無効になっています。

Tuned では定義済みのプロファイルを利用でき、高スループット、低レイテンシー、または省電力などの一般的なユースケースに活用することができます。各プロファイルの Tuned ルールを修正し、特定デバイスのチューニング方法をカスタマイズすることができます。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、「[powertop2tuned の使用](#)」を参照してください。

プロファイルは、使用中の製品をベースに自動的にデフォルトに設定されます。**tuned-adm recommend** コマンドを使用して、特定の製品に対する Red Hat の推奨最適プロファイルを確認することができます。推奨プロファイルがない場合は、**balanced** プロファイルが設定されます。

balanced はほとんどの負荷に適するプロファイルで、エネルギー消費、パフォーマンス、およびレイテンシーのバランスに優れます。**balanced** プロファイルにより、利用可能な最大限のコンピューティングリソースを使用して、素早くタスクを完了することができます。同じタスクを少ないコンピューティングリソースで長時間実施する場合に比べて、少ないエネルギーしか必要としないのが通常です。

ノートパソコンがアイドル状態にある時、またはわずかなコンピューティングリソースしか必要としないタスクを実施中の場合、**powersave** 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。エネルギー消費を抑える代わりに大きなレイテンシーが許容される場合、またはタスクを素早く完了する必要がない場合などが、その例として挙げられます。具体的には、IRC の使用、簡単な Web ページの閲覧、またはオーディオおよびビデオファイルの再生などです。

Tuned および **tuned-adm** で利用できる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「[Tuned](#)」の章を参照してください。

powertop2tuned の使用

powertop2tuned ユーティリティにより、PowerTOP の提案からカスタム Tuned プロファイルを作成することができます。PowerTOP の詳細については、「[PowerTOP](#)」を参照してください。

powertop2tuned ユーティリティをインストールするには、以下のコマンドを使用します。

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

カスタムプロファイルを作成するには、以下のコマンドを使用します。

```
~]# powertop2tuned new_profile_name
```

デフォルトでは、**powertop2tuned** は現在選択されている Tuned プロファイルに基いて **/etc/tuned/** ディレクトリーにプロファイルを作成します。安全上の理由から、初めは新しいプロファイルではすべての 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** ファイルでチューニングをアンコメントします。

--enable または **-e** オプションを使用して、PowerTOP の提案するほとんどのチューニングが有効な新しいプロファイルを生成することができます。USB 自動サスペンドなど問題となる可能性のある特定のチューニングは、デフォルトでは無効になっているので、手動でアンコメント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。アクティブ化するには、以下のコマンドを使用します。

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

powertop2tuned がサポートする全オプションのリストを表示するには、以下のコマンドを使用します。

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

2.6. UPOWER

Red Hat Enterprise Linux 6 では、**DeviceKit-power** は、**HAL** の一部である電力管理機能と Red Hat Enterprise Linux の以前のリリースの **GNOME Power Manager** の一部である電力管理機能を引き継ぎました (「[GNOME の電源管理](#)」も参照)。Red Hat Enterprise Linux 7 では、**DeviceKit-power** は **UPower** という名前に変更されました。**UPower** は、デーモン、API、および一連のコマンドラインツールを提供します。物理デバイスかどうかに関係なく、システム上の各電源はデバイスとして表されます。たとえば、ノートパソコンのバッテリーと AC 電源は両方ともデバイスとして表されます。

コマンドラインツールにアクセスするには、**upower** コマンドと以下のオプションを使用します。

--enumerate, -e

システム上の電源デバイス用のオブジェクトパスを表示します。例えば以下のとおりです。

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

--dump, -d

システム上の全ての電源デバイス用のパラメータを表示します。

--wakeups, -w

システムの CPU のウェイクアップを表示します。

--monitor, -m

AC 電源の接続や切断、あるいはバッテリーの低下などの電源デバイスの変化についてシステムを監視します。システムの監視を止めるには、**Ctrl+C** を押します。

--monitor-detail

AC 電源の接続や切断、あるいはバッテリーの低下などの電源デバイスの変化についてシステムを監視します。**--monitor-detail** オプションでは、**--monitor** オプションよりも詳細を提供します。システムの監視を止めるには、**Ctrl+C** を押します。

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

特定のオブジェクトパスに利用可能なすべての情報が表示されます。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のバッテリーに関する情報を取得するには、以下のコマンドを実行します。

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

2.7. GNOME の電源管理

GNOME Power Manager は、GNOME デスクトップの一部としてインストールされるデーモンです。Red Hat Enterprise Linux の以前のバージョンで **GNOME Power Manager** が提供した電力管理機能の大部分は、Red Hat Enterprise Linux 6 で **DeviceKit-power** の一部となり、Red Hat Enterprise Linux 7

では **UPower** という名前に変更されました (「[UPower](#)」を参照)。ただし、**GNOME Power Manager** はその機能のフロントエンドとして残ります。システムトレイのアプレットを介して **GNOME Power Manager** は、バッテリーから AC 電源への切り替えなど、システムの電源状態の変化を通知します。また、バッテリーの状態を報告し、バッテリーの電力が低くなると警告を出します。

2.8. 他の監査ツール

Red Hat Enterprise Linux7 は、システムの監査と分析を実行する複数のツールを提供します。それらのほとんどは、すでに発見したものを検証する場合や特定の部分の詳細情報が必要な場合に補助の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。以下に、これらのツールを示します。

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、および CPU 活動について詳細情報を提供します。システム全体で実行している動作やビジーな部分を詳しく見るために使用します。

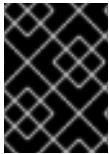
iostat

iostat は **vmstat** と似ていますが、ブロックデバイスの I/O 専用です。詳細な出力と統計も提供します。

blktrace

blktrace は、非常に詳細に渡るブロック I/O のトレースプログラムです。情報をアプリケーションに関連した1つずつのブロックに分割します。**diskdevstat** と併せて使用すると大変役立ちます。

第3章 中核となるインフラストラクチャとメカニズム



重要

本章で解説している **cpupower** コマンドを使用する場合は、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 のアイドル状態

x86 アーキテクチャの CPU は、CPU の一部が停止したり、低パフォーマンス設定で稼働したりするさまざまな状態をサポートします。これらの状態は C 状態と呼ばれ、使用されていない CPU を部分的に停止させることで節電を可能にします。C 状態は番号付けされ、C0 から始まります。数字が大きいと、CPU の機能が低下し、省電力量が向上します。特定の番号が付いた C 状態は、プロセッサ間でほとんど同じですが、状態の特定の機能セットの詳細はプロセッサファミリー間で異なる場合があります。C 状態 0-3 は以下のように定義されます。

C0

稼働中または実行中の状態。この状態では、CPU は動作中であり、アイドル状態ではありません。

C1, 停止

プロセッサが命令を実行していない状態ですが、一般的に電力が低い状態ではありません。CPU は実質的に遅延なく処理を継続できます。C 状態を提供するプロセッサはすべて、この状態をサポートする必要があります。Pentium 4 プロセッサは、実際には電力消費が低い状態の C1E と呼ばれる拡張 C1 状態をサポートします。

C2, クロック停止

このプロセッサのクロックが停止している状態ですが、そのレジスターとキャッシュは完全な状態で保持されるため、クロックを再開させると直ちに処理を再開することができます。この状態はオプションです。

C3, スリープ

プロセッサが実際にスリープ状態になり、キャッシュを更新する必要がない状態です。この状態からウェイクアップするには、C2 状態からのウェイクアップに比べ、かなり長い時間がかかります。この状態もオプションです。

利用可能なアイドル状態、および CPUidle ドライバーのその他の統計値を表示するには、次のコマンドを実行します。

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

Nehalem マイクロアーキテクチャーに基づく最近の Intel CPU には、新しい C 状態である C6 が備わっています。これにより、CPU の電圧供給をゼロに削減できますが、通常は 80%~90% 電力消費量が削減されます。Red Hat Enterprise Linux 7 のカーネルには、この新しい C 状態に対する最適化が含まれています。

3.2. CPUFREQ

ご使用のシステムで電力消費と熱の発生を低減する最も効果的な方法の1つは、CPUfreq を使用することです。CPUfreq (CPU 速度スケールリングとも呼ばれます) は Linux カーネルのインフラストラクチャーで、節電のために CPU 速度をスケールリングできます。CPU スケールリングは、ACPI イベントに

応じたシステム負荷を元に自動的に実施することも、ユーザースペースプログラムにより手動で実施することもできます。これにより、稼働中にプロセッサのクロック速度を調整できます。したがって、システムは減速したクロック速度で稼働し、節電できます。周波数の変更、クロック速度の変更、および周波数を変更するタイミングに関するルールは、CPUfreq ガバナーによって定義されます。

3.2.1. CPUfreq ドライバー

CPUfreq では、ACPI CPUfreq および Intel P-state の 2 種類のドライバーを使用することができます。

ACPI CPUfreq

ACPI CPUfreq ドライバーは、ACPI を通じて特定 CPU の周波数をコントロールするカーネルドライバーで、これによりカーネルとハードウェア間のコミュニケーションが可能になります。

Intel P-state

Red Hat Enterprise Linux 7 では、Intel P-state ドライバーがサポートされています。このドライバーの提供するインターフェースにより、Intel Xeon E シリーズアーキテクチャーまたはより新しいアーキテクチャーをベースとしたプロセッサにおける P 状態の選択をコントロールすることができます。Intel P-state では `setpolicy()` コールバックが実装されています。ドライバーは、`cpufreq` コアから要求されたポリシーに応じて、使用すべき P 状態を判断します。プロセッサが内部的に次の P 状態を選択する機能を持っている場合、ドライバーはこの責任をプロセッサにオフロードします。持っていない場合は、次の P 状態を選択するアルゴリズムをドライバーが実装します。

Intel P-state では、P 状態の選択をコントロールするための専用の `sysfs` ファイルを利用することができます。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。これらのファイルに加えたすべての変更は、すべての CPU に適用されます。このディレクトリーに含まれる 5 つのファイルを使用して、P 状態のパラメーターを設定します。

- **max_perf_pct**: ドライバーから要求される最大の P 状態を制限します (利用可能なパフォーマンスのパーセンテージで定義)。利用可能な P 状態のパフォーマンスは、`no_turbo` 設定により低く抑えることができます (下記を参照)。
- **min_perf_pct**: ドライバーから要求される最小の P 状態を制限します (最大 (`no-turbo`) パフォーマンスレベルのパーセンテージで定義)。
- **no_turbo**: `turbo` 周波数範囲以下で P 状態を選択するようにドライバーを制限します。
- **turbo_pct**: `turbo` 範囲にあるハードウェアがサポートするトータルパフォーマンスのパーセンテージを表示します。 `turbo` が無効になっているかどうかは、この数値に影響を与えません。
- **num_pstates**: ハードウェアがサポートする P 状態の数を表示します。 `turbo` が無効になっているかどうかは、この数値に影響を与えません。

現在、Intel P-state に対応する CPU では、デフォルトで Intel P-state が使用されます。ACPI CPUfreq の使用に切り替えるには、カーネルコマンドラインに以下のパラメーターを追加します。

```
intel_pstate=disable
```

3.2.2. CPUfreq ガバナー

ガバナーは、システムの CPU の電力特性を定義します。これにより、CPU のパフォーマンスに影響を受けます。各ガバナーには、作業負荷に関してそれぞれ固有の動作、目的、および適合性があります。このセクションでは、CPUfreq ガバナーの選択および設定方法、各ガバナーの特性、および各ガバナーに適している作業負荷の種類について説明します。

Red Hat Enterprise Linux 7 では、異なるタイプの CPUfreq ガバナーが利用可能です。それらを以下に示します。

cpufreq_performance

Performance ガバナーは、CPU が最高クロック周波数を使用するように強制します。この周波数は静的に設定され、変化しないため、このガバナーでは、**節電する利点はありません**。このガバナーは、何時間にも渡るような作業負荷が大きい時だけ、しかも CPU がアイドル状態になることがほとんどない（もしくはまったくならない）時のみに適しています。

cpufreq_powersave

一方、Powersave ガバナーは、CPU が最低クロック周波数を使用するように強制します。この周波数は静的に設定され変化しないため、このガバナーでは最大の節電を実現しますが、**CPU パフォーマンスが一番低く** なってしまいます。

しかし「節電 (Powersave)」という用語は時に誤解を招きます。全負荷で遅い CPU は (原則として)、負荷がない高速の CPU よりも多くの電力を消費します。そのため、低活動が予期できる時には Powersave ガバナーを使用するよう CPU を設定することが推奨されますが、この期間中に予期しない高負荷が発生するとシステムは実際にはより多くの電力を消費することがあります。

Powersave ガバナーは簡単にいうと、CPU にとっては「節電」よりも「スピードリミッター」の意味を持ちます。これは、過熱が問題となる恐れがあるシステムや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す。システム負荷が大きい時は、CPU は最高クロック周波数を実現し、システムがアイドル状態の時には、CPU は最低クロック周波数を実現します。これにより、システム負荷に対してシステムは電力消費量を適宜調節できますが、そうすることで **周波数変換の間の遅延** が発生してしまいます。そのため、システムがアイドル状態と高負荷の間で頻繁に替わりすぎると、遅延により Ondemand ガバナーが実現できるパフォーマンスおよび/または節電の利点が少なくなる恐れがあります。

ほとんどのシステムでは、Ondemand ガバナーは熱の放出、消費電力、パフォーマンス、および管理のしやすさの間で、最良の妥協策を提供します。1日の中で特定の時間帯にのみシステムがビジーになる場合は、Ondemand ガバナーはそれ以上介入せずに、負荷に応じて最高周波数と最低周波数の間で自動的に切り替わります。

cpufreq_userspace

Userspace ガバナーを使用すると、ユーザースペースプログラム (または、root で実行しているいずれのプロセス) が周波数を設定できます。Userspace ガバナーは、すべてのガバナーの中で最もカスタマイズ可能であり、設定によってはご使用のシステムでパフォーマンスと電力消費のバランスを最適化できます。

cpufreq_conservative

Ondemand ガバナーと同様に、Conservative ガバナーも使用量に応じてクロック周波数を調節します (Ondemand ガバナーと同様です)。ただし、Ondemand ガバナーがより積極的にクロック周波数を調節するのに対し (最高周波数から最低周波数、そして最高周波数に戻る)、Conservative ガバナーはもっとゆっくりと調節を行います。

これが意味しているのは、Conservative ガバナーは単に最高と最低の周波数を選択するのではなく、負荷に対して適切と判断するクロック周波数に合わせるということです。これは電力消費に著しく貢献する可能性があります、Ondemand ガバナーよりも **長い遅延** で行います。



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、1日のある時間帯にあるガバナーを自動的に設定することができます。そのため、アイドル状態(例えば終業後)の時には、低周波数のガバナーを指定し、高負荷となる時間帯には高周波数に戻るよう設定できます。

特定のガバナーを有効にする方法については、「[CPUfreq のセットアップ](#)」を参照してください。

3.2.3. CPUfreq のセットアップ

すべての CPUfreq ドライバーは、kernel-tools パッケージの一部としてビルドされ、自動的に選択されます。したがって、ガバナーを選択するだけで CPUfreq をセットアップできます。

以下のコマンドを実行すると、特定の CPU に使用できるガバナーを表示できます。

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

以下のコマンドを実行すると、すべての CPU に対してこれらのいずれかのガバナーを有効にできます。

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

特定のコアに対してのみガバナーを有効にするには、CPU メンバーの範囲またはカンマ区切りリストとともに **-c** を使用します。たとえば、CPU1~3 および 5 の Userspace ガバナーを有効にするには、以下のコマンドを実行します。

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

3.2.4. CPUfreq ポリシーおよび速度のチューニング

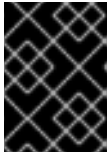
適切な CPUfreq ガバナーを選択した後で、**cpupower frequency-info** コマンドを使用して CPU 速度とポリシー情報を表示できます。さらに、**cpupower frequency-set** のオプションを使用して、各 CPU の速度をチューニングできます。

cpupower frequency-info には、以下のオプションを使用できます。

- **--freq**: CPUfreq コアに基いて現在の CPU の速度を KHz 単位で表示します。
- **--hwfreq**: ハードウェアに基いて現在の CPU の速度を KHz 単位で表示します (root でのみ利用可能)。
- **--driver**: この CPU の周波数を設定するために使用する CPUfreq ドライバーを表示します。
- **--governors**: このカーネルで使用できる CPUfreq ガバナーを表示します。このファイルには表示されていない CPUfreq ガバナーを使用したい場合は、手順について「[CPUfreq のセットアップ](#)」を参照してください。
- **--affected-cpus**: 周波数調整ソフトウェアを必要とする CPU を一覧表示します。
- **--policy**: 現在の CPUfreq ポリシーの範囲 (KHz 単位) と現在アクティブなガバナーを表示します。
- **--hwlimits**: CPU に使用できる周波数 (KHz 単位) を一覧表示します。

cpupower frequency-set では、以下のオプションを使用することができます。

- **--min <freq>** と **--max <freq>**: CPU の *ポリシー制限* を KHz 単位で設定します。



重要

ポリシー制限を設定する場合は、**--min** よりも前に **--max** を設定する必要があります。

- **--freq <freq>**: CPU に特定のクロック速度を KHz 単位で設定します。設定できる速度は CPU のポリシー制限内に限られます (**--min** と **--max**)。
- **--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、CPUfreq の設定は `/sys/devices/system/cpu/[cpuid]/cpufreq/` 内にあるパラメーターで確認できます。設定と値は、これらのパラメーターに書き込むことにより変更できます。たとえば、cpu0 の最小クロック速度を 360 KHz に設定する場合は、以下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 監視機能

cpupower には、アイドル状態とスリープ状態の統計値および周波数情報を提供し、プロセッサトポロジーを報告する各種の監視機能が備わっています。プロセッサ固有の監視機能もあれば、あらゆるプロセッサと互換性がある監視機能もあります。各監視機能の測定対象と互換性のあるシステムの詳細については、**cpupower-monitor** の man ページを参照してください。

次のオプションは、**cpupower monitor** コマンドとともに使用します。

- **-l**: システムで使用できる全監視機能を一覧表示します。
- **-m <monitor1>**、**<monitor2>**: 特定の監視機能を表示します。その識別子は、**-l** を実行して確認できます。
- **command**: 特定コマンドに関するアイドル統計値と CPU 要求を表示します。

3.4. CPU 省電力ポリシー

cpupower を使うと、プロセッサの省電力ポリシーを変更できます。

次のオプションは **cpupower set** コマンドとともに使用します。

--perf-bias <0-15>

サポートされた Intel プロセッサでソフトウェアがよりアクティブに最適なパフォーマンスと省電力とのバランスを決定できるようにします。このオプションは他の省電力ポリシーよりも優先されません。割り当てる値の範囲は 0~15 です。ここで、0 は最適なパフォーマンスであり、15 は最適な電力効率です。

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。コア別に適用する場合は、**--cpu <cpulist>** オプションを追加します。

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

他の CPU パッケージが選ばれるまで、一つの CPU パッケージ内のコアに対するシステムプロセスによる電力使用を制限します。0 は制限なし、1 は最初に CPU パッケージを1つだけ採用、2 は1の内容に加えてタスクのウェイクアップを処理する場合にセミアイドル状態の CPU パッケージを優先します。

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

他の コアが選ばれるまで、1つの CPU コアの複数スレッドに対するシステムプロセスによる電力使用を制限します。0 は制限なし、1 は最初に CPU パッケージを1つだけ採用、2 は1の内容に加えてタスクのウェイクアップを処理する場合にセミアイドル状態の CPU パッケージを優先します。

3.5. サスペンドと復帰

システムがサスペンド状態になると、カーネルはドライバーを呼び出してその状態を保存し、それからドライバーをアンロードします。システムが復帰する時には、ドライバーを再読み込みし、デバイス群を再プログラムします。このタスクを遂行するドライバーにより、システムが正常に復帰できるかどうかが決まります。

この点では、ビデオドライバーが特に問題です。その理由は、ACPI (電力制御インタフェース: *Advanced Configuration and Power Interface*) 規格では、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です。そのため、ビデオドライバーがハードウェアを完全な未初期化の状態からプログラムできない限りは、システムは復帰できないことがあります。

Red Hat Enterprise Linux 7 では、新しいグラフィックチップセットをより強力にサポートしています。これにより、サスペンドと復帰は以前より多くのプラットフォームで機能します。特に、NVIDIA チップセットに対するサポートは格別に向上しており、GeForce 8800 シリーズでは特に改善されています。

3.6. 実行時デバイス電源管理

実行時デバイス電源管理 (RDPM: Runtime Device Power Management) により、ユーザーへの影響を最小限に抑えて電力消費を削減できます。デバイスが一定の時間アイドル状態になり、RDPM ハードウェアサポートがデバイスとドライバーの両方に存在する場合、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 イベントにより行われ、デバイスを実行状態に戻すためにカーネルとデバイスドライバーがトリガーされます。RDPM はデフォルトで有効であるため、このすべての操作は自動的に行われます。

特定の RDPM 設定ファイルで属性を設定することにより、ユーザーはデバイスの RDPM を制御するよう許可されます。特定のデバイスの RDPM 設定ファイルは、`/sys/devices/device/power/` ディレクトリーになります。ここで、`device` は、特定のデバイスのディレクトリーのパスに置き換えます。

たとえば、CPU に対して RDPM を設定する場合は、以下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 操作のレイテンシーが増加します。このレイテンシーの増加の時間はデバイスに固有です。ここで説明する設定スキームを使用すると、システム管理者がデバイスごとに RDPM を無効にしたり、他のパラメーターの一部を調査および制御したりできます。各 `/sys/devices/device/power` ディレクトリーには、以下の設定ファイルが含まれます。

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。すべてのデバイスでは、**control** ファイルに属性の次の 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値。ドライバーによって自動的な RDPM になる場合があります。

on

実行時にドライバーがデバイスの電力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、デバイスのアイドル状態と中断状態の間のアクティビティーがない最小期間である自動サスペンドの遅延を制御します。このファイルにはミリ秒単位の自動サスペンド遅延値が含まれます。正の値の場合は、実行時にデバイスがサスペンドされず、`/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があります。1000 よりも大きい値は最も近い秒に丸められます。

3.7. ACTIVE-STATE POWER MANAGEMENT

Active-State Power Management (ASPM) は、接続するデバイスが使用中でない時に PCIe リンク用に電力状態を低く設定することにより、*Peripheral Component Interconnect Express* (PCI Express または PCIe) サブシステムで電力を節約します。ASPM はリンクの両端で電力状態を制御し、リンクの末端のデバイスが最大電力の状態の場合でもリンク内で電力を節約します。

ASPM が有効な場合は、異なる電力状態の間でリンクを切り替えるために必要な時間のため、デバイスのレイテンシーが大きくなります。ASPM には、電力状態を決定する以下の 3 つのポリシーがあります。

デフォルト

システムのファームウェア (たとえば、BIOS) で指定されたデフォルト値に従って、PCIe リンクの電力状態を設定します。これは ASPM のデフォルト状態です。

powersave

パフォーマンスの低下に関係なく、できる限り電力を節約するように ASPM を設定します。

performance

PCIe リンクが最大パフォーマンスで稼働できるように ASPM を無効にします。

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せん。

- **pcie_aspm=off** は ASPM を無効にします。
- ASPM に対応していないデバイスでも、**pcie_aspm=force** は ASPM を有効にします。

ハードウェアが ASPM に対応している場合は、システムの起動時に、オペレーティングシステム ASPM を自動的に有効になります。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

ASPM に対応していないハードウェアで **pcie_aspm=force** を使用して強制的に ASPM を有効にすると、システムが応答しなくなることがあります。**pcie_aspm=force** を設定する前に、システムにあるすべての PCIe ハードウェアが ASPM に対応することを確認してください。

ASPM ポリシーを設定するには、以下のいずれかのオプションを選択します。

- `/sys/module/pcie_aspm/parameters/policy` ファイルでこの設定を修正します。
- システムの起動時に **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** が ASPM パフォーマンスポリシーを設定します。

3.8. AGGRESSIVE LINK POWER MANAGEMENT

Aggressive Link Power Management (ALPM) は、アイドル時 (I/O が存在しない時) にディスクへの SATA リンクを低電力に設定することにより、ディスクの省電力を促進する省電力技術です。I/O リクエストがそのリンクへのキューに格納されると、ALPM によって、SATA リンクが自動的にアクティブな電力状態に戻ります。

ALPM で導入された省電力では、ディスクの遅延が発生します。したがって、アイドル状態の I/O 時間が長くなると思われる場合にのみ ALPM を使用してください。

ALPM は、*Advanced Host Controller Interface* (AHCI) を使用する SATA コントローラ上でのみ利用できます。AHCI の詳細については、<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ります。ALPM には以下の 3 つのモードがあります。

min_power

このモードでは、ディスクに I/O がない場合に、リンクが最小電力状態 (SLUMBER) に設定されます。このモードは、アイドル時間が長くなると思われる場合に役に立ちます。

medium_power

このモードでは、ディスク上に I/O がない場合に、2 番目に電力が低い状態 (PARTIAL) にリンクが設定されます。このモードは、パフォーマンスへの影響を最小化するために、リンクの電力状態を切り替えることができるよう設計されています (たとえば、一時的に I/O が多くなるときや I/O がアイドル状態になるとき)。

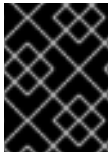
medium_power モードでは、負荷に応じてリンクを PARTIAL の状態と最大電力 (「ACTIVE」) の状態の間で切り替えることができます。PARTIAL から SLUMBER に、そして再び PARTIAL にリンクを直接切り替えることはできません。このような場合は、どちらの電力状態も最初に ACTIVE 状態を経由せずに、他の状態に切り替わることはできません。

max_performance

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

ご使用の SATA ホストアダプターが実際に ALPM をサポートするかどうかを調べるに

は、`/sys/class/scsi_host/host*/link_power_management_policy` ファイルが存在するかどうかを確認します。設定を変更するには、このセクションに記載された値をこのファイルに書き込むか、あるいはファイルを表示して現在の設定を確認します。



重要

ALPM を `min_power` または `medium_power` に設定すると、自動的に「ホットプラグ」機能が無効になります。

3.9. RELATIME ドライブアクセス最適化

POSIX 基準では、各ファイルが最後にアクセスされた時間を記録するファイルシステムのメタデータがオペレーティングシステムによって維持されていなければなりません。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に常時書き込みをする動作が必要になります。これらの書き込みにより、ストレージデバイスとそのリンクに常に電源が投入され、ビジー状態になります。`atime` データを使用するアプリケーションは少ないため、このストレージデバイスの動作が電力を浪費していることとなります。重要なことは、ストレージへの書き込みは、ファイルがストレージからではなくキャッシュから読み込まれた場合でも発生する点です。これまで、Linux カーネルでは `mount` 用の `noatime` オプションに対応してきたため、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` データを書き込んでいませんでした。しかし、単純に `atime` データを使用しないことにも問題があります。一部のアプリケーションは `atime` データに依存しているため、これが利用できないと機能しないためです。

Red Hat Enterprise Linux 7 で使用しているカーネルは、代替となる `relatime` に対応しています。`Relatime` では `atime` データを維持しますが、ファイルがアクセスされる度の書き込み動作はしません。このオプションを有効にすると、ファイルが変更された、つまり `atime` が更新された (`mtime`) 場合、またはファイルが最後にアクセスされてから一定以上の時間 (デフォルトでは 1 日) が経過している場合に限り、`atime` データがディスクに書き込まれます。

デフォルトでは、`relatime` が有効な状態ですべてのファイルシステムがマウントされるようになります。特定のファイルシステムに対してこのオプションを無効にしたい場合には、そのファイルシステムをマウントする際に `norelatime` オプションを使用します。

3.10. パワーキャッピング (POWER CAPPING)

Red Hat Enterprise Linux 7 では、HP の *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パワーキャッピング (電力制限) 機能を利用しています。パワーキャッピングにより、管理者はサーバーによる電力消費の上限を設定できるだけでなく、より効率的にデータセンターを計画できます。その理由は、既存の電力供給装置に過負荷をかけるリスクが大幅に減少するためです。また、管理者はさらに多くのサーバー群を同じ物理フットプリント (physical footprint) に配置でき、サーバーの電力消費が制限されると、確実に高負荷時に電力需要が利用可能な電力を超えないようにします。

HP Dynamic Power Capping

Dynamic Power Capping は、選ばれた ProLiant と BladeSystem のサーバーで利用できる機能であり、システム管理者が 1 つのサーバー、あるいはサーバーのグループの電力消費量を制限できるようにします。キャップとは、現時点の作業負荷に関係なく、サーバーが超過しない確実な上限のことです。キャップには、サーバーがその消費電力の上限に到達するまでは何の効果もありません。到達した時点で、管理プロセッサは CPU P 状態 と クロックスロットル (clock throttling) を調節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から独立して CPU の動作を個別に修正しますが、HP の *integrated Lights-Out 2* (iLO2) ファームウェアにより、オペレーティングシステムは管理プロセッサにアクセスでき、その結果ユーザースペースのアプリケーションは管理プロセッサにクエ

ることができます。Red Hat Enterprise Linux 7 で使用されているカーネルには HP iLO と iLO2 のファームウェア用のドライバーが含まれており、プログラムが `/dev/hpilo/dXccbN` で管理プロセッサにクエリできるようにします。カーネルには、パワーキャッピング機能をサポートするための `hwmon sysfs` インターフェースの拡張と、`sysfs` インターフェースを使用する ACPI 4.0 パワーメーター用の `hwmon` ドライバーが含まれています。これらの機能が一緒になって、オペレーティングシステムとユーザースペースのツールがパワーキャップ用に設定された値とシステムの現在の電力消費量を読み込めるようになります。

HP Dynamic Power Capping の詳細は、HPE のナレッジベース『「HP ProLiant Gen8 Server Series - Power Capping and Dynamic Power Capping Settings in iLO 4 and RBSU」』(https://support.hpe.com/hpsc/doc/public/display?docId=mmr_sf-EN_US000006556) を参照してください。

Intel Node Manager

Intel Node Manager は、CPU パフォーマンス、ひいては電力消費量を制限するためにプロセッサの P 状態と T 状態を使用して、システムにパワーキャップをかけます。電源管理ポリシーを設定することにより、管理者は、例えば夜間や週末などのシステムの負荷が低い時に電力消費が低くなるよう設定することができます。

Intel Node Manager は、標準の *電力制御インタフェース (Advanced Configuration and Power Interface)* を通じて、OSPM *オペレーティングシステム向け構成および電力管理 (Operating System-directed configuration and Power Management)* を使用することで CPU の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 状態への変更を通知すると、そのドライバーは P 状態に対応する変更を加えます。同様に Intel Node Manager が OSPM ドライバーに P 状態への変更を通知すると、ドライバーはそれに応じて T 状態を変更します。こうした変更は自動的に発生し、オペレーティングシステムの介入を必要としません。管理者は *Intel Data Center Manager (DCM)* ソフトウェアを使用して Intel Node Manager の設定と監視を行います。

Intel Node Manager の詳細は、<http://communities.intel.com/docs/DOC-4766> にある『Node Manager – A Dynamic Approach To Managing Power In The Data Center』を参照してください。

3.11. 拡張グラフィックス電力管理

Red Hat Enterprise Linux 7 は不必要な電力消費の複数の発生源を取り除くことにより、グラフィックスデバイスとディスプレイデバイスの節電を行います。

LVDS 再クロック

LVDS *低電圧差動信号 (Low-voltage differential signalling)* とは、電子信号を銅線上で伝えるシステムです。このシステムが応用されている重要な例の 1 つは、ピクセル情報をノート PC の *液晶ディスプレイ (LCD)* 画面に送信することです。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。これはディスプレイがグラフィックコントローラから新しいデータを受け取り、画像を画面に再表示する頻度です。通常、画面は毎秒 60 回新しいデータを受信します (60 Hz の周波数)。画面とグラフィックコントローラが LVDS でリンクされている時は、LVDS システムはリフレッシュのたびに電力を使用します。アイドル状態の時、多くの LCD 画面のリフレッシュレートは、目立った変化なく 30 Hz まで低下することがあります (リフレッシュレートが低下すると特有のフリッカーが起こる *ブラウン管 (CRT)* モニターとは異なります)。Red Hat Enterprise Linux 7 の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用のドライバーは、自動的にこの *ダウンスクロック (downclocking)* を実行し、画面がアイドル状態の時には約 0.5 W の節電をします。

メモリーのセルフリフレッシュの有効化

SDRAM *Synchronous dynamic random access memory*: これは、グラフィックスアダプタのビデオメモリーに使用されます。毎秒何千回もリチャージされるため、個々のメモリーセルは保管されているデータを保持します。データはメモリーの内外へと移動するためそのデータを管理するその主要機能の他に、メモリーコントローラには通常これらのリフレッシュサイクルを開始する役割があります。一方、

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、そのリフレッシュサイクルを生成します。これにより、現在メモリーに保存されているデータを危険にさらすことなく、システムはメモリーコントローラをシャットダウンできます。Red Hat Enterprise Linux 7 で使用されているカーネルは、アイドル状態の時に Intel グラフィックスアダプタのメモリーにセルフリフレッシュをさせることがあります。これにより約 0.8 W の節電ができます。

GPU クロックの低減

標準的なグラフィカルプロセッシングユニット (GPU) には、その内部回路の各種パーツを制御する内部クロックが含まれています。Red Hat Enterprise Linux 7 で使用されているカーネルは、Intel および ATI の GPU 内の内部クロックの一部の周波数を低くすることができます。GPU コンポーネントが所定時間内に実行するサイクル数を低減すると、それらが実行する必要がなかったサイクルで消費されていたであろう電力を節減します。GPU がアイドル状態の時には、カーネルは自動的にそうしたクロックの速度を遅くし、GPU の活動が増加すると速めます。GPU のクロックサイクルを低下させることで、最大で約 5 W の節電ができます。

GPU の電源オフ

Red Hat Enterprise Linux 7 の Intel と ATI グラフィックスドライバーは、アダプタにモニターが接続されていない時を検出できるため、GPU を完全にシャットダウンすることができます。この機能は、常時モニターを接続していないサーバーで特に重要です。

3.12. RFKILL

多くのコンピューターシステムには、Wi-Fi、Bluetooth、および 3G デバイスを含む無線送信器が搭載されています。これらのデバイスは電力を消費し、使用していない時には無駄になります。

RFKill は、Linux カーネルのサブシステムで、コンピューターシステムの無線送信器をクエリ、アクティブ化、非アクティブ化するインターフェースを提供します。無線送信器が非アクティブ化されると、それらはソフトウェアが再びアクティブ化できる状態 (ソフトブロック) に置かれるか、またはソフトウェアが再びアクティブ化できない状態 (ハードブロック) に置かれます。

RFKill コアは、サブシステムにアプリケーションプログラミングインターフェース (API) を提供します。RFkill をサポートするように設計されている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ます。また、デバイスを有効および無効にする方法を含んでいます。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器の状態をクエリする方法を提供します。

RFKill インターフェースは `/dev/rfkill` にありますが、システムのすべての無線送信器の現在の状態が含まれています。各デバイスの現在の RFKill の状態は、**sysfs** に登録されています。また、RFKill は RFKill 対応のデバイス内の状態の変化について *uevents* を発行します。

Rfkill は、システム上の RFKill 対応のデバイスをクエリ、変更できるコマンドラインツールです。このツールを取得するには、`rfkill` パッケージをインストールしてください。

コマンド `rfkill list` を使用すると、デバイスの一覧が取得できます。それぞれのデバイスにはそれに関連した **0** から始まるインデックス番号があります。このインデックス番号を使用して `rfkill` に対してデバイスのブロックとブロック解除を指示します。例を示します。

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

上記は、システムの最初の RFKill 対応デバイスをブロックします。

また、`rfkill` を使用してデバイスの特定のカテゴリ、またはすべての RFKill 対応のデバイスもブロックできます。例を示します。

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

システムのすべての 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、以下を実行します。

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

デバイスをブロック解除するには、**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の全一覧を取得するには、**rfkill help** を実行してください。

第4章 使用例

この章では、2種類のユースケースを使ってこのガイドで説明している分析と設定方法を説明しています。最初は、標準的なサーバーを例にとり、その次は標準的なノート PC を例にして考えてみます。

4.1. 例: サーバー

今日の一般的な標準サーバーは、Red Hat Enterprise Linux 7 でサポートされている必要なハードウェアの機能の多くが搭載されています。最初に考慮すべきなのは、サーバーの主要な使用目的となる作業負荷の種類です。この情報に基づき、節電のためにどのコンポーネントを最適化するか決定できます。

サーバーのタイプに関係なく、グラフィックス性能は一般的には必要ありません。そのため、GPU 節電はオンのままで結構です。

ウェブサーバー

ウェブサーバーにはネットワークとディスク I/O が必要です。外部の接続スピードによっては、100 Mbit/s で十分かも知れません。マシンがほとんど静的なページを使用する場合は、CPU のパフォーマンスはあまり重要ではないでしょう。以下のような電力管理の選択肢があります。

- **tuned** にはディスクまたはネットワークのプラグインなし。
- ALPM をオンにする
- **ondemand** ガバナーをオンにする
- ネットワークカードは 100 Mbit/s に制限する

計算サーバー

計算サーバーには主に CPU が必要です。以下のような電力管理の選択肢があります。

- ジョブとデータストレージが発生する場所に応じて、**tuned** のディスク、またはネットワークプラグイン。または バッチモードシステムには、完全にアクティブな **tuned**。
- 使用量によっては、**performance** ガバナー。

メールサーバー

メールサーバーには、多くの場合ディスク I/O と CPU が必要です。以下のような電力管理の選択肢があります。

- **ondemand** ガバナーはオン。CPU パフォーマンスの最後の数パーセントは重要でないためです。
- **tuned** にはディスクまたはネットワークのプラグインなし。
- メールは内部で発生することが多く、1 Gbit/秒 か 10 Gbit/秒 のリンクから利用できるためネットワークスピードは制限しません。

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に似ています。しかし使用するプロトコル次第では、さらなる CPU パフォーマンスが必要になる可能性があります。一般的に Samba ベースのサーバーは、NFS よりも CPU を要求して、NFS は一般的に iSCSI よりも CPU を要求します。それでも、**ondemand** ガバナーを使用できるはずです。

ディレクトリーサーバー

ディレクトリーサーバーのディスク I/O の要件は、一般的に低いものです。十分な RAM がある場合は特にそうです。ネットワーク遅延は重要ですが、ネットワーク I/O はそれほどでもありません。リンクの速度が遅い遅延のネットワークのチューニングを考えられるかも知れませんが、これを特定のネットワークに注意深くテストするようにしてください。

4.2. 例: ノート PC

電力管理と節電が実際に効果をもたらすもうひとつの非常に一般的な対象は、ノート PC です。ノート PC はもともとワークステーションやサーバーよりも大幅に少ないエネルギーを使用するように設計されているため、絶対的な節電ができる可能性は他のマシンよりも低くなります。ただし、バッテリーモードでは、どんな節電でもノートパソコンのバッテリー寿命を数分でも延長するのに役立ちます。このセクションでは、ノート PC のバッテリーモードにフォーカスしていますが、もちろん AC 電源での使用でもこうしたチューニングの一部、またはすべてを活用することができます。

1つのコンポーネントの節電は、通常ワークステーションよりもノートパソコンで相対的に大きな効果をもたらします。例えば、100 Mbits/秒 で実行している 1Gbit/秒 ネットワークインターフェースはおよそ 3-4 ワット節約します。約 400 ワットの合計消費電力を持つ標準的なサーバーには、この節約はおよそ 1% です。約 40 ワットの合計消費電力を持つノートパソコンでは、この 1つのコンポーネントの節電は合計でおよそ 10% になります。

標準的なノート PC での特定の節電最適化としては以下のものがあります。

- システムの BIOS を使用しないすべてのハードウェアを無効にするように設定します。例えば、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 および Bluetooth などが可能です。
- スクリーンを見るために最高輝度が必要ない暗めの場所では、ディスプレイ輝度を低くします。そのためには、GNOME デスクトップでは、システム+設定 → 電力管理 と進みます。KDE デスクトップでは、アプリケーション起動キックオフ (Kickoff Application Launcher)+コンピュータ+システム設定+高度な設定 → 電力管理 と進みます。または、コマンドラインで `gnome-power-manager` か、`xbacklight` を実行するか、ノート PC でファンクションキーを使用します。

また、(代わりに) 各種システム設定を微調整することもできます。

- **ondemand** 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です)。
- AC97 オーディオ節電機能を有効にします (Red Hat Enterprise Linux 7 ではデフォルトで有効です)。

```
~]# 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 デバイスで正常に機能するわけではありません。

- `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 ではデフォルトです)。

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

- 画面の輝度を **50** かそれ以下に下げます。例えば以下のとおりです。

-

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

- スクリーンのアイドル状態に DPMS をアクティベートします。

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

- Wi-Fi を非アクティブ化します。

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

付録A 開発者へのヒント

すべての優れたプログラミング教本では、メモリー割り当ての問題と特定の機能のパフォーマンスについて説明しています。ソフトウェアを開発する場合は、ソフトウェアが実行されるシステムで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。この場合は、コードのすべての行が影響を受けるわけではなく、頻繁にパフォーマンスのボトルネックになる領域のコードを最適化できます。

問題になることが多い手法は以下のとおりです。

- スレッドの使用。
- 不必要な CPU のウェイクアップとウェイクアップの非効率的な使用。ウェイクアップする必要がある場合は、すべての処理を一度にできるだけ迅速に実行します (すぐにアイドル状態になるように実行します)。
- `[f]sync()` の不必要な使用。
- 不必要なアクティブポーリングまたは短い通常のタイムアウトの使用 (代わりにイベントに反応する)。
- ウェイクアップの非効率的な使用。
- 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回避するために大きなバッファを使用してください。一度に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。可能な場合は、アプリケーション群 (またはシステム群) でタイマーをグループ化します。
- 過度の I/O、電力消費、またはメモリー使用 (メモリーリークを含む)。
- 不必要な計算の実行。

以下のセクションでは、これらの領域についてさらに詳しく説明します。

A.1. スレッドの使用

一般的に、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、高速になると思われていますが、これはすべてのケースで当てはまるわけではありません。

Python

Python は Global Lock Interpreter^[1] を使用するため、スレッドは大規模な I/O 操作でのみ効果的です。Unladen-swallow^[2] は、コードを最適化できる可能性がある Python の高速な実装です。

Perl

Perl のスレッドは、元々はフォークがないシステム (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開発されました。Perl のスレッドでは、データはすべての単独スレッドに対してコピーされます (コピーオンライト)。ユーザーはデータ共有のレベルを定義できるため、データはデフォルトでは共有されません。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 (コピーオンライト)、モジュールによってデータの関連変数も作成されます (さらに時間がかかり、処理が遅くなります)^[3]。

C

C のスレッドは同じメモリーを共有します。各スレッドは独自のスタックを持ち、カーネルは新しいファイル記述子を作成したり、新しいメモリースペースを割り当てたりする必要がありません。C はよ

り多くのスレッドにより多くの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ます。スクリプト言語を使用する場合は、Cバインディングを記述することを検討してください。プロファイラーを使用すると、適切に実行されていないコード部分を特定できます^[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確認するためにスキャンします。多くの場合、スキャンは、たとえば毎分、決まった間隔で実行されます。スキャンによりディスクがスピンドウンから強制的にウェイクアップさせられるため、スキャンが問題になることがあります。最善策は、適切な間隔または適切な確認メカニズムを見つけるか、`inotify` で変更を確認して、イベントに対応することです。`inotify` を使用すると、ファイルまたはディレクトリーのさまざまな変更を確認できます。

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

    return EXIT_SUCCESS;
}
```

この方法の利点は、さまざまな確認を実行できることです。

主な制限は、1つのシステムで利用できる監視の数が限定されることです。この数は `/proc/sys/fs/inotify/max_user_watches` から取得できます。この数値を変更することは可能ですが、推奨されません。また、`inotify` が失敗すると、コードは別の確認方法にフォールバックする必要がありますが、これは通常ソースコードで `#if #define` を数多く使用することを意味します。

`inotify` の詳細は、man ページの `inotify(7)` を参照してください。

A.3. FSYNC

`Fsync` は I/O コストが高い操作として知られていますが、実際にはそうでない場合もあります。

ユーザーが新しいページに移動するリンクをクリックする度に、Firefox は `sqlite` ライブラリーを呼び出していました。`sqlite` が `fsync` を呼び出すときに、ファイルシステム設定 (主に `data-ordered` モードの `ext3`) が原因で、長い遅延が発生し、何も処理が行われませんでした。また、この場合は、同時に別のプロセスが大きなファイルをコピーしているときに、最大で 30 秒の時間がかかっていました。

ただし、`fsync` が全く使用されない別のケースでは、`ext4` ファイルシステムへの切り替えで問題が発生していました。`Ext3` は `data-ordered` モードに設定され、数秒毎にメモリーがフラッシュされ、その内容がディスクに保存されていました。ただし、`ext4` と `laptop_mode` を使用する場合は、保存の間隔が長いいため、システムの電源が予期せずにオフになったときにデータが消失することがありました。現在、`ext4` にはパッチが適用されましたが、アプリケーションを慎重に設計し、適切に `fsync` を使用する必要があります。

設定ファイルに対する読み書きの以下の簡単な例は、ファイルのバックアップ方法と、データがどのように失われるかを示しています。

```
/* 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);
```

より適切な例は以下のようになります。

```
/* 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.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 改訂履歴

| | | |
|--|-----------------|------------------|
| 改訂 2.2-6 7.4 GA 公開用ドキュメントバージョン | Mon Jul 24 2017 | Marie Doleželová |
| 改訂 2.2-5 非同期のアップデート: 「Tuned」の章の書き直し | Tue Mar 21 2017 | Milan Navrátil |
| 改訂 2.0-2 7.3 GA 公開用バージョン | Fri Oct 14 2016 | Marie Doleželová |
| 改訂 2.0-1 7.2 GA 公開用バージョン | Wed 11 Nov 2015 | Jana Heves |
| 改訂 1-3 「中核となるインフラストラクチャとメカニズム」で間違ったパッケージ名を修正 | Fri 19 Jun 2015 | Jacquelynn East |
| 改訂 1-2 7.1 GA 向けバージョン | Wed 18 Feb 2015 | Jacquelynn East |
| 改訂 1-1 7.1 Beta 向けバージョン | Thu Dec 4 2014 | Jacquelynn East |
| 改訂 1.0-9 7.0 GA リリース向けバージョン | Tue Jun 9 2014 | Yoana Ruseva |
| 改訂 0.9-1 スタイル変更のための再ビルド | Fri May 9 2014 | Yoana Ruseva |
| 改訂 0.9-0 レビューのための Red Hat Enterprise Linux 7.0 リリースドキュメント | Wed May 7 2014 | Yoana Ruseva |
| 改訂 0.1-1 ドキュメントの Red Hat Enterprise Linux 6 バージョンからのブランチ | Thu Jan 17 2013 | Jack Reed |