



Red Hat Enterprise Linux 7

電力管理ガイド

RHEL 7 での電力消費の管理と最適化

Red Hat Enterprise Linux 7 電力管理ガイド

RHEL 7 での電力消費の管理と最適化

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

Copyright © 2017 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

本書は、Red Hat Enterprise Linux 7 システムで電力消費を効果的に管理する方法を説明します。以下のセクションでは、消費電力を下げるさまざまな技術と、サーバーおよびラップトップの両方で、各テクニックがシステムのパフォーマンス全体に与える影響を説明しています。

目次

第1章 概要	3
1.1. 電源管理の重要性	3
1.2. 電源管理の基礎	4
第2章 電源管理監査および分析	6
2.1. 監査および分析の概要	6
2.2. POWERTOP	6
2.3. DISKDEVSTAT および NETDEVSTAT	9
2.4. BATTERY LIFE TOOL KIT	12
2.5. TUNED	15
2.6. UPOWER	16
2.7. GNOME POWER MANAGER	17
2.8. 監査のその他のツール	17
第3章 コアインフラストラクチャーおよび MECHANICS	19
3.1. CPU ID の状態	19
3.2. CPUFREQ	20
3.3. CPU モニター	26
3.4. CPU 省電力ポリシー	27
3.5. 一時停止および再開	28
3.6. ランタイムデバイスの電源管理	28
3.7. ACTIVE-STATE 電源管理	29
3.8. アグレッシブなリンク電源管理	31
3.9. 再リレードライブへのアクセスの最適化	32
3.10. POWER CAPPING	32
3.11. 強化されたグラフィック電源管理	33
3.12. RFKILL	34
第4章 使用例	36
4.1. 例：サーバー	36
4.2. 例：LAPTOP	37
付録A 開発者に関するヒント	40
A.1. スレッドの使用	41
A.2. WAKE-UPS	41
A.3. FSYNC	42
付録B 改訂履歴	45

第1章 概要

電源管理は、Red Hat Enterprise Linux 7 の改良点の1つです。コンピューターシステムによって使用される電源を制限することは、緑のIT（環境フレンドリーなコンピューティング）の最も重要な側面の1つ、反復可能なマテリアルの使用、ハードウェア実稼働の環境への影響、およびシステム設計およびデプロイメントにおいて環境による認識にも対応します。本書では、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

電源管理の中核となるのは、各システムコンポーネントの消費を効果的に最適化する方法を理解しています。これにより、システムが実行する各種タスクを確認し、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適しているようにします。

電源管理の主な motivator は以下のとおりです。

- 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 heat 削減
- 照合、スペース、ケーブル、ジェネレーター、割り込み不可能な電力メッカー(UPS)を含むセカンダリーコストの削減
- ノートパソコン拡張バッテリーのライフサイクル
- 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
- 新しいシステムの会社ガイドラインの会議

ルールとして、特定のコンポーネント（またはシステム全体として）の消費量を下げると、heat と自然的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向けに行う設定によるパフォーマンス低下の低下を發揮し、テストしてください。

システムが実行する各種タスクを確認し、各コンポーネントを設定して、ジョブに対してパフォーマンスが十分になるように設定すると、プラグジーや heat が少なく、ノートパソコン用にバッテリーのライフサイクルを最適化することができます。電力消費に関するシステムの分析および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よく似ています。ある程度のレベル、電源管理、およびパフォーマンスチューニングは、通常システムのパフォーマンスや電源が最適化されるため、システム設定への反対のアプローチとなります。このマニュアルでは、Red Hat が提供するツールと、このプロセスで開発した手法を説明します。

Red Hat Enterprise Linux 7 には、デフォルトで有効になっている多くの電源管理機能が含まれています。これらは、典型的なサーバーや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的に選択されています。ただし、スループット、最も低いレイテンシー、または CPU パフォーマンスを最大限必要という非常に特殊なユースケースでは、これらのデフォルトを確認する必要がある場合があります。

本書で説明されているテクニックを使用してマシンを最適化すべきかどうかを決定するには、十分にいくつかの質問をお願いします。

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： 最適化する必要なのはどれくらいですか？

答： 存在するいくつかのテクニックは、監査プロセス全体を調べてマシンを分析する必要はありませんが、代わりに、電力使用量を向上させる一般的な最適化セットを提供します。このため、当然ながら、手動で監査対象と最適化されたシステムは適切ではありませんが、不正性が向上します。

問： システムのパフォーマンスを容認できないレベルに最適化しますか？

答： 本ガイドで説明されているテクニックのほとんどは、おそらく、システムのパフォーマンスに影響を及ぼします。Red Hat Enterprise Linux 7ですでに有効なデフォルトを超えた電源管理を実装する場合は、電源最適化後にシステムのパフォーマンスを監視し、パフォーマンス損失が許容可能であるかどうかを判断する必要があります。

問： システム全体でシステムの最適化に費やす時間とリソースが費やしますか？

答： プロセス全体の後に1つのシステムを手動で最適化することは、通常、この処理に費やすことが重要ではありません。したがって、1台のマシンの有効期間を引き継ぐのよりは、一般的な利点よりも高くなります。一方、10000 デスクトップシステムの場合、同じ設定と設定を使用するすべてのオフィスでロールアウトすると、最適化されたセットアップを作成し、すべての10000 マシンに適用することが推奨されます。

以下のセクションでは、電力消費に関して、システムで最適なハードウェアパフォーマンスの利点を説明します。

1.2. 電源管理の基礎

効果的な電源管理が以下の原則で構築されます。

アイドル CPU は必要な場合にのみ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を実行し、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ました。そのため、アイドル状態の CPU は、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を保つことができます。また、電力状態に入る CPU は、これらの状態が長く続きます。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ある場合には、この機能の利点をオフセットにすることができます。ボリュームの変更やマウスの移動の確認など、ポーリングイベントにはこのようなイベントの例があります。

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特定し、監査できるツールが含まれています。[2章 電源管理監査および分析](#)。

未使用のハードウェアとデバイスを完全に無効にする必要があります。

これは特に、移動部分（ハードディスクなど）となるデバイスに対してとくに当てはまります。これに加えて、一部のアプリケーションは未使用のままですが、有効になっているデバイス「open;」を残すと、カーネルはデバイスが使用されていることを前提としています。これにより、デバイスが省電力状態になるのを防ぐことができます。

低アクティビティーが低ウォーピングに変換される

ただし、多くの場合、最新のハードウェアと正しい BIOS 設定により異なります。以前のシステムコンポーネントでは、多くの場合、Red Hat Enterprise Linux 7 でサポートされていた新機能の一部のサポートはありません。システムの最新の公式ファームウェアを使用し、電源管理機能の BIOS の電源管理またはデバイス設定セクションで使用していることを確認してください。検索すべき機能の一部は次のとおりです。

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

お使いのハードウェアがこの機能をサポートしており、BIOS で有効になっている場合、Red Hat Enterprise Linux 7 はデフォルトでこれらを使用します。

CPU の状態とその影響の各種形式

最新の CPU と *Advanced Configuration and Power Interface* (ACPI) は、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。

- sleep (C-states)
- 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 出力 (ヒントまたは「thermal state」)

スリープ状態の低い状態で実行中の CPU は、可能な限り少ないウォートを使用できますが、必要に応じてその状態から起動するのにかなり時間がかかります。非常にまれなケースでは、これにより CPU がスリープ状態になる直前に CPU をウェイクアップする必要が生じる場合があります。このような場合は、実質的に完全にビジーな CPU が発生し、別の状態が使用された場合は、電力節約の一部が失われます。

電源がオフのマシンには、最小容量の電源を使用する

これがどのようにサウンドするかは、実際に節電に最も良い方法の1つとして、システムの電源を切ることにあります。たとえば、企業が「Green IT」に焦点を合わせた企業の文法を開発して、未指定の破損時や自宅にマシンを有効にできるようになっています。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します。

第2章 電源管理監査および分析

2.1. 監査および分析の概要

単一システムの詳細な手動監査、分析、およびチューニングは通常、このようなシステムチューニングの最終部分から得られた利点を得られるため、通常例外になります。ただし、全システムに同じ設定を再利用できる、ほぼ同一システムにこのタスクを一度行うと非常に便利です。たとえば、数千のデスクトップシステムのデプロイ、またはマシンがほぼ同一の HPC クラスターのデプロイについて考えてみましょう。監査および分析を行う別の理由は、今後システム動作でリグレッションや変更を特定できる比較のベース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発生する場合に非常に役立ち、消費量に関する出現を回避したい場合があります。通常は詳細な監査と分析を行うことで、特定のシステムで実際に何が起きているのかについての理解がはるかに優れています。

電力消費に関するシステムの監査および分析は、最新のシステムが利用可能な最も新しいシステムであっても、比較的困難です。多くのシステムでは、ソフトウェアを介して電力の使用を測定する方法はありません。例外は、Hewlett Packard サーバーシステムの ILO 管理コンソールは、Web を介してアクセス可能な電源管理モジュールを持ちます。IBM は、BladeCenter 電源管理モジュールに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 は電力監視機能を提供します。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サポートされるソリューションは1つではありません。

電力消費の直接測定は、可能な限りの削減を最大限に活用するために必要なことがよくあります。つまり、変更が有効かどうかや、システムを動作させる方法を測定するために他の方法を使用できます。本章では、必要なツールを説明します。

2.2. POWERTOP

Red Hat Enterprise Linux 7 のティックレスカーネルが導入され、CPU はアイドル状態状態をより頻繁に入力できるようになり、電力消費を削減し、電源管理を改善できます。**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識別します。**PowerTOP** は、本リリースで多数のアプリケーションをチューニングした監査を実行するために開発に使用されており、不要な CPU の数を数十 10 まで減らします。

Red Hat Enterprise Linux 7 には、**PowerTOP** のバージョン 2.x が同梱されています。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す。これにより、より正確なデータを提供するために、より明確なタブベースのユーザーインターフェースが備えられ、カーネルの「perf」インフラストラクチャーを使用します。システムデバイスの電源動作が追跡され、目立ち、問題がすぐに表示されます。2.x のコードベースには、個別デバイスおよびプロセスが消費している電力量を示す、電力予測エンジンが含まれます。[図2.1 「powertop in Operation」](#)。

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** で次のコマンドを実行します。

```
~]# powertop
```

PowerTOP は、システムの総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々の電力使用量を表示します。ラップトップは、このタスク中のバッテリー電源で実行する必要があります。**電力予測エンジンを調整するには、root** で以下を実行します。

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

調整には時間がかかります。プロセスは、さまざまなテストを実行しますが、明るさのレベルやデバイスのオンとオフを切り替えることができます。プロセスは終了し、調整時にマシンと対話しないようにします。調整プロセスが完了すると、PowerTOP が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、電力予測マークが最初の列に表示されます。

ノートパソコンで実行している場合は、利用可能なすべてのデータが提示されるように、バッテリー電源で実行している必要があります。

PowerTOP は、実行中に、システムから統計を収集します。Overview タブでは、ウェイクアップを最も頻繁に CPU に送信するか、最も電力を消費するコンポーネントの一覧を表示できます (図 2.1 「powertop in Operation」)。隣接する列には、リソースの使用法、毎秒のウェイクアップ、プロセス、デバイス、タイマーなどの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、サービスまたはデバイス、ならびにカーネルのドライバーが効率的に実行されているかを示します。ウェイクアップが少ないほど、消費電力が少なくなります。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。

ドライバーコンポーネントの調整には通常、カーネルの変更が必要になります。これについては、本書では扱いません。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されるプロセスです。まず、このシステムでこのサービスまたはアプリケーションを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、非アクティブにします。古い System V サービスを永続的にオフにするには、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
```

トレース自体が繰り返されるように見える場合は、おそらくビジーループとなります。このようなバグの修正には、通常、そのコンポーネントでコードの変更が必要になります。

図 2.1 「powertop in Operation」、電力消費と残りのバッテリーの寿命はそちらに表示されます (該当する場合)。以下は、1秒あたりの合計ウェイクアップ、秒ごとの 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合計を特長とする簡単な概要です。残りの画面では、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、その他のリソースの一覧が表示されます。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 Shift+Tab キーを使用して、Idle stats タブで、すべてのプロセッサおよびコアについて C-states を使用できます。Frequency stats タブで、Turbo モード (該当する場合) がすべてのプロセッサおよびコアについて表示されます。CPU の C 状態または P 状態が高いままになるほど、適切なサイズ (C4 は C3 より高) になります。これは、CPU 使用率がどの程度最適化されているかを示すのに適しています。システムがアイドル状態のときに、最高の C 状態または P 状態では 90% 以上が理想的であることが理想的です。

Device Stats タブは Overview タブと同様の情報を提供しますが、デバイス専用です。

Tunables タブには、低消費電力にシステムの最適化に関する提案が含まれています。up キーおよび down キーを使用して提案を移動し、Enter キーを使用して提案をオンとオフを切り替えます。

図2.1 powertop in Operation

```
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
```

[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 秒間隔で測定値を取り、--time オプションで変更できます。

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

PowerTOP の詳細は、[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。Intel 64 プロセッサのプロセッサトポロジー、周波数、アイドル電力状態の統計、温度、および電力使用量についての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は、[turbostat\(8\) man 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 および NETDEVSTAT

Diskdevstat および **netdevstat** は、システムで実行している全アプリケーションのディスクアクティビティとネットワークアクティビティ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、**PowerTOP** により提供され、毎秒各アプリケーションによる CPU ウェイクアップ数が表示されます（「**PowerTOP**」）。これらのツールが収集する統計により、大規模な操作ではなく、多くの小規模の I/O 操作で電源が入っているアプリケーションを特定することができます。転送速度のみを測定する他の監視ツールは、このタイプの用途を特定するのに役立ちません。

root で以下のコマンドを実行して、**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
```

以下のコマンドでツールを実行します。

```
~]# diskdevstat
```

または、以下のコマンドを実行します。

```
~]# netdevstat
```

いずれのコマンドも、以下に示すように最大 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

ディスプレイの更新間隔（秒単位）。初期値：5

total_duration

全体の実行時間（秒単位）。初期値：86400（1日）

display_histogram

実行の最後に収集したすべてのデータのヒームをヒュームにするかどうかフラグ。

出力は、PowerTOP のようになります。以下は、長い `diskdevstat run` からの出力例です。

```

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

```

列は以下のようになります。

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

この例では、3 つの非常に明確なアプリケーションは、以下のようになります。

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には 0 よりも大きな WRITE_CNT が 0 を超えています。これは、測定中に何らかの書き込み形式を実行したことを意味します。これらこれらでは、plasma は最も大きな書き込み動作で実行され、書き込み間の平均時間が最小限に抑えられていました。したがって、電力効率の非効率のアプリケーションについて懸念するのは、Plasma が最適です。

strace コマンドおよび ltrace コマンドを使用して、指定のプロセス ID のすべてのシステムコールを追跡することで、アプリケーションをより密接に確認します。この例では、以下を実行できます。

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

この例では、strace の出力に、ユーザーの KDE アイコンキャッシュファイルを開いた 45 秒ごとに繰り返しパターンが含まれており、その後ファイルを再度閉じます。これにより、ファイルのメタデータ (特に変更時間) がハードディスクに必要な物理書き込みが行われました。最終の修正は、アイコンへの更新が行われなかった場合に不要な呼び出しを防ぐために生じました。

2.4. BATTERY LIFE TOOL KIT

Red Hat Enterprise Linux 7 では、BLTK (Battery Life Tool Kit) が導入されています。これは、バッテリーの寿命とパフォーマンスのシミュレーションおよび分析を行うテストスイートです。BLTK は、

特定のユーザーグループをシミュレートし、結果について報告するタスクセットを実行して、これを実現します。特にノートブックのパフォーマンスをテストするために開発されていますが、BLTKは `-a` で起動するとデスクトップコンピュータのパフォーマンスも報告できます。

BLTKでは、マシンの実用として使用できる非常に再現可能なワークロードを生成することができます。たとえば、オフィスのワークロードではテキストの書き込み、修正、スプレッドシートで同じ処理が行われます。PowerTOP またはその他の監査または分析ツールで BLTK を組み合わせて実行すると、マシンがアイドルだけでなく、機械がアクティブに使用されているかどうかをテストできます。異なる設定に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
```

以下のコマンドで BLTK を実行します。

```
~]# bltk workload options
```

たとえば、120 秒間アイドル状態のワークロードを実行するには、次のコマンドを実行します。

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

デフォルトで利用可能なワークロードは次のとおりです。

-I, --idle

他のワークロードとの比較のベースラインとして使用するためのシステムがアイドル状態です。

-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（デフォルトでは mplayer を使用）からのマルチメディアファイルを監視するシミュレーション

-O, --office

OpenOffice.org スイートでドキュメントの編集をシミュレートします。

他のオプションにより、以下を指定できます。

-a, --ac-ignore

AC 電源が利用可能かどうかを無視します (デスクトップの使用には不必要)

-T number_of_seconds, --time number_of_seconds

テストを実行する時間 (秒単位)。アイドルワークロードでこのオプションを使用します。

-F filename, --file filename

CD や DVD ドライブにアクセスするのではなく、Player ワークロードのファイルなど、特定のワークロードで使用するファイルを指定します。

-W application, --prog application

特定のワークロードで使用するアプリケーションを指定します (例: 読者のワークロードに Firefox 以外のブラウザー)。

BLTK は、より特殊なオプションを多数サポートしています。詳細は、man ページの bltk を参照してください。

BLTK は、`/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/`) で生成される結果を保存します。たとえば、`~/.bltk/reader.results.002/` ディレクトリーには、リーダーのワークロードが含まれる 3 番目のテストの結果を保持します (最初のテストは番号されません)。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、読みやすい形式に集約するには、以下を実行します。

```
~]$ bltk_report path_to_results_directory
```

結果は、結果ディレクトリーに **Report** という名前のテキストファイルに表示されるようになります。代わりにターミナルエミュレーターの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し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を有効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、スループットや低レイテンシー、省電力などの一般的なユースケースを処理する事前定義プロファイルを提供します。各プロファイルの Tuned ルールを変更し、特定のデバイスのチューニング方法をカスタマイズできます。PowerTOP の提案からカスタムの Tuned [「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 コマンドを使用して、特定の製品に最適なプロファイルとして、Red Hat が推奨するプロファイルを確認することができます。推奨事項がない場合は、balanced プロファイルが設定されます。

大半のワークロードに適した balanced プロファイル。消費消費、パフォーマンス、およびレイテンシーのバランスを取ります。balanced プロファイルでは、利用可能な最大コンピューティング能力でタスクを迅速に完了することで、コンピューティング能力が少ない長期で同じタスクを実行するよりも、通常少なくなります。

ノートパソコンがアイドル状態の状態にある場合、省電力プロファイルを使用すると、計算の非オンデマンド操作のみが実行されます。このような操作では、より低い電力消費に対してレイテンシーが高くなるのは一般的です。また、IRC、簡単な Web ページの表示、オーディオおよびビデオファイルの再生など、直ちに操作を完了する必要はありません。

tuned-adm で提供される Tuned [『プロファイルおよび省電力オプションの詳細は、『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 ユーティリティを使用すると、PowerTOP の提案からカスタムの Tuned プロファイルを作成できます。PowerTOP [「PowerTOP」](#)。

powertop2tuned ユーティリティをインストールするには、次のコマンドを実行します。

```
~]# 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、すべての PowerTOP チューニングは最初に新しいプロファイルで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` ファイルでコメントを解除します。

`--enable` オプションまたは `-e` オプションを使用して、PowerTOP が提案するチューニングのほとんどを可能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、新規プロファイルはアクティベートされません。アクティベートするには、次のコマンドを実行します。

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

`powertop2tuned` に対応しているオプションの完全リストを表示するには、以下を使用します。

```
~]# powertop2tuned --help
```

2.6. UPOWER

Red Hat Enterprise Linux 6 DeviceKit-power では、HAL に含まれていた電源管理機能と、Red Hat Enterprise Linux の以前のリリースで GNOME Power Manager 「[GNOME Power Manager](#)」。Red Hat Enterprise Linux 7、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を提供します。システムの各電源ソースは、物理デバイスであるかに関係なく、デバイスとして表示されます。たとえば、ラップトップのバッテリーと AC 電源ソースは、いずれもデバイスとして表示されます。

`theupower` コマンドと、以下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`

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

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

--dump, -d

システムにあるすべての電源デバイスのパラメーターを表示します。

--wakeups, -w

システムで CPU ウェイクアップを表示します。

--monitor, -m

システムが、AC 電力のソースの接続や接続解除など、電源デバイスへの変更がないか監視します。Ctrl+C を押して、システムの監視を停止します。

--monitor-detail

システムが、AC 電力のソースの接続や接続解除など、電源デバイスへの変更がないか監視します。--monitor-detail オプションは、--monitor オプションの詳細情報を表示します。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 で表現されてい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。Red Hat Enterprise Linux 6 の以前のバージョンで提供される GNOME Power Manager の機能の多くは、Red Hat Enterprise Linux 6 の DeviceKit-power ツールに代わり、Red Hat Enterprise Linux 7 の UPower に変更されています [UPower])。ただし、GNOME Power Manager は、その機能のフロントエンドになります。GNOME Power Manager は、システムトレイのアプレットを介してシステムの電源状態の変更を通知します。たとえば、バッテリーから AC 電源への変更への変更を通知します。また、バッテリーの電力状態も報告し、バッテリーの電源が低い場合に警告します。

2.8. 監査のその他のツール

Red Hat Enterprise Linux7 は、システム監査および分析を実行するツールを多数提供します。そのほとんどは、すでに検出した内容、または特定の部分に関する詳細な情報が必要な場合には、情報の補足情報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。これには、以下が含まれます。

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 アクティビティーの詳細情報を提供します。システム全体の内容と、ビジー状態を確認するには、それを使用します。

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。さらに、より詳細な出力および統計が提供されます。

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 1 つのブロックに情報をダウンさせる。 **diskdevstat** と組み合わせて使用す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および MECHANICS

**重要**

本章の `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ます。

3.1. CPU ID の状態

x86 アーキテクチャーを持つ CPU は、CPU の一部が非アクティブになるか、パフォーマンスが低い設定で実行するさまざまな状態をサポートします。C 状態として知られるこれらの状態により、未使用の CPU を部分的に非アクティブ化することで、システムの電源を節約できます。c-states は、C0 以降の番号で、CPU 機能の縮小と電力削減がより大きくなります。特定の数の c-States はプロセッサ間で大きく似ていますが、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によって異なる場合があります。c-States 0-3 は以下のように定義されます。

C0

操作または稼働状態。この状態では、CPU は機能しており、アイドル状態ではありません。

C1, halt

プロセッサが命令を実行していないが、通常は少ない電源状態ではない状態。CPU は、実際には遅延しない状態で処理を継続できます。C-States を提供するプロセッサはすべて、この状態に対応する必要があります。Pentium 4 プロセッサは、低消費電力の状態である C1E と呼ばれる、強化された C1E 状態をサポートします。

C2, stop-Clock

このプロセッサのクロックがフリーズしている状態で、レジスタとキャッシュの完全な状態を保持するため、クロックを再び開始した後、再度処理を開始できます。これはオプションの状態です。

C3, sleep

プロセッサがスリープ状態になり、キャッシュを最新の状態に維持する必要がない状態。この状態からのウェイクアップは、このために C2 よりもはるかに長い時間がかかります。これはオプションの状態です。

CPUidle ドライバーで利用可能なアイドル状態およびその他の統計を表示するには、以下を入力します。

```
~]# cpupower idle-info
```

新しい C-State、C6 (Nehalem" マイクロアーキテクチャー) を持つ最近の Intel CPU は、新しい C-State、C6 です。これは、CPU をゼロに減らすことができますが、通常は 80% と 90% の間で電力消費を削減します。Red Hat Enterprise Linux 7 のカーネルには、この新しい C-State の最適化が含まれています。

3.2. CPUFREQ

システムの電力消費と heat の出力を減らす最も効果的な方法の 1 つは CPUfreq です。cpufreq: CPU 速度のスケーリングとも呼ばれる - 節電のために CPU 周波数のスケーリングを可能にするインフラストラクチャーです。CPU のコールは、システムの負荷、ACPI イベントへの応答、またはユーザー空間プログラムにより手動で対話でき、プロセッサのクロック速度を柔軟に調整できます。これにより、システムクロックのスピードが減少して電力を節約できます。シフトの周波数、高速または遅いクロック速度を使用するかどうか、およびシフトの周波数を行うタイミングは、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ます。ドライバーは、Intel Xeon E シリーズアーキテクチャーまたはそれ以降のアーキテクチャーに基づいたプロセッサ上の P-state 選択を制御するインターフェースを提供します。Intel P-state は setpolicy () コールバックを実装します。ドライバーは、cpufreq コアから要求されるポリシーに基づいて、使用する P-state を決定します。プロセッサが次の P 状態を内部から選択できる場合、ドライバーはこの役割をプロセッサにオフロードします。そうでない場合は、ドライバーはアルゴリズムを実装し、次の P-state を選択します。

Intel P-state は独自の sysfs ファイルを提供し、P-state 選択を制御します。このファイルは、/sys/devices/system/cpu/intel_pstate/ ディレクトリーにあります。ファイルに加えられた変更は、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ます。

- **max_perf_pct:** ドライバーで要求される最大 P 状態を制限します。利用可能なパフォーマンスの割合で表現します。利用可能な P-state のパフォーマンスは、no_turbo 設定により軽減

できます（下記参照）。

- **min_perf_pct: min_perf_pct:** ドライバーで要求される最小 P-state を制限します。これは、最大（保護されていない）パフォーマンスレベルをパーセンテージで表現します。
- **no_turbo:** ドライバーを制限して、turbo 周波数の範囲の下にある P 状態を選択します。
- **turbo_pct:** turbo 範囲にあるハードウェアがサポートする合計パフォーマンスの割合を表示します。この数は、turbo が無効になっているかどうかとは独立しています。
- **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。この数は、turbo が無効になっているかどうかとは独立しています。

現在、サポートされる CPU には、Intel P-state がデフォルトで使用されています。カーネルコマンドラインに以下を追加することにより、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 Governors

CPufreq ガバナーは、システム CPU の特性を定義します。これにより、CPU パフォーマンスに影響します。各ガバナーには、ワークロードに関して独自の固有の動作、目的、および適合性があります。本セクションでは、CPufreq ガバナー、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選択および設定する方法を説明します。



警告

Red Hat Enterprise Linux 7 には、複数のコア CPufreq ガバナーが複数含まれています。Intel P-state ドライバーは、デフォルトでアクティブモードで動作します。このモードでは、2 つの CPufreq ガバナーのみが利用できます（performance および powersave）。パフォーマンスおよび省電力 Intel P-state CPufreq ガバナーは、コアとなる CPufreq ガバナーと同じものとは異なることに注意してください。

3.2.2.1. コア CPufreq Governors

Red Hat Enterprise Linux 7 で利用可能なさまざまな CPUfreq ガバナーは以下のとおりです。

cpufreq_performance

Performance ガバナーは、CPU が、可能な限り最も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るため、変更されません。そのため、このガバナーでは省電力性は提供しません。ワークロードが大きい時間にのみ適しており、CPU のアイドル状態ではない（またはしない）時にのみ適しています。

cpufreq_powersave

一方、**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るため、変更されません。そのため、このガバナーは最大省電力を提供しますが、CPU のパフォーマンスは最も低くなります。

ただし、「powersave」という用語は、ロードされない高速 CPU よりも多くの電力を消費するため、（原則）フルロード時の CPU の速度が低下するためです。したがって、予想される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、その間に予期しない負荷があると、システムが実際に電力を消費する可能性があります。

Powersave ガバナーは、シンプル用語で、「power saver」よりも「スピードリミッター」よりも多くなります。これは、オーバートラフィックが問題となる可能性のあるシステムと環境で最も役立ちます。

cpufreq_ondemand

Ondemand ガバナーは、CPU がシステム負荷が高い場合に最大クロック周波数を実現できる動的ガバナーです。また、システムがアイドル状態の時の最小クロック周波数も実現できます。これにより、システムの負荷に応じて電力消費を調整することができますが、周波数スイッチ間のレイテンシーが高くなります。したがって、レイテンシーは、システムがアイドル状態のワークロードと大きいワークロード間を過剰に切り替わる場合に、**Ondemand** ガバナーによって提供されるパフォーマンス/省電力の利点をオフセットできます。

多くのシステムでは、**Ondemand** ガバナーは、heat のエミッション、電力消費、パフォーマンス、管理性の最良の危険性を提供します。システムが特定の時間にのみビジー状態になると、**Ondemand** ガバナーは、さらなる介入なしに負荷に応じて、最大周波数と最小周波数の間で自動的に切り替えられます。

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 として実行中のプロセスによる頻度を設定できます。すべてのガバナー、ユーザー空間、設定方法によっては、システムのパフォーマンスと消費のバランスを最大限に提供できます。

cpufreq_conservative

Conservative ガバナーと同様に、**Conservative** ガバナーは、（**Ondemand** ガバナーのように）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、より積極的な方法（最小値とバック）で、**Conservative** ガバナー(**frequencor**)のスイッチをより段階的に行うようにします。

よって、**Conservative** ガバナーは、最大値と最小値のいずれかを選択するのではなく、負荷に適合するクロック周波数に調整されます。これにより電力消費量が大幅に節約できますが、**Ondemand** ガバナーよりもレイテンシーが長いこともあります。



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、特定の期間に特定のガバナーを自動的に設定することができます。そのため、アイドル時の低頻度ガバナー（例：勤務時間後など）を指定でき、ワークロードが大きい時間の高いときにより高い頻度ガバナーに戻ることができます。

特定のガバナーを有効にする方法は、[「cpufreq 設定」](#)。

3.2.2.2. Intel P-state CPUfreq Governors

Intel P-state ドライバーは、3つのモードで動作できます。

- アクティブモード (ハードウェア管理の P-state (HWP) あり)
- アクティブモード (ハードウェア管理の P-state (HWP) なし)
- パッシブモード

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

ハードウェア管理の P-state を使用するアクティブモード

HWP を使用したアクティブモードが使用される場合、**Intel P-state** ドライバーは CPU に P-state 選択を指示します。ドライバーは頻度のヒントを提供できます。ただし、最終的な選択肢は、CPU 内部ロジックに依存します。

HWP を使用するアクティブモードでは、Intel P-state ドライバーは 2 つの P 状態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、ドライバーは内部 CPU ロジックにパフォーマンス指向を指示します。許可される P 状態の範囲は、ドライバーが使用できる範囲の下限に制限されます。

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を指示します。

アクティブモード (ハードウェア管理の P-state なし)

HWP を使用しないアクティブモードを使用すると、Intel P-state ドライバーは 2 つの P 状態選択アルゴリズムを提供します。

- パフォーマンス
- `powersave`

Performance ガバナーでは、ドライバーは、使用できる最大 P-state を選択します。

Powersave ガバナーでは、ドライバーは現在の CPU 使用率に比例する P-states を選択します。動作は、Ondemand CPUfreq core ガバナーと似ています。

パッシブモード

パッシブモードが使用される場合、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 設定

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、自動的に選択されるため、CPUfreq を設定するには、ガバナーを選択する必要があります。

以下を使用して、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
```

その後、以下を使用して、全 CPU でこれらのガバナーのいずれかを有効にすることができます。

```
~]# 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、`-c` を範囲またはコンマ区切りの CPU 番号で指定します。たとえば、CPU 1-3 および 5 でユーザー空間ガバナーを有効にするには、コマンドは以下のようになります。

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

3.2.4. CPUfreq ポリシーおよび Speed のチューニング

適切な CPUfreq ガバナーを選択すると、`cpupower frequency-info` コマンドで CPU の速度とポリシー情報を表示し、`cpupower frequency-set` のオプションを指定して各 CPU の速度をさらに調整できます。

`Forcpupower frequency-info` では、以下のオプションを使用できます。

- `--freq - KHz` で CPUfreq コアに従って CPU の現在の速度を表示します。
- `--hwfreq - KHz` (root としてのみ利用可能) で、ハードウェアに従って CPU の現在の速度を表示します。
- `--driver` - この CPU の周波数を設定するために使用される CPUfreq ドライバーを表示します。
- `--governors` - このカーネルで利用可能な CPUfreq ガバナーを表示します。このファイルに記載されていない CPUfreq ガバナーを使用する場合は、「[cpufreq 設定](#)」。

- **--affected-cpus:** 周波数の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 ポリシーの範囲を KHz および現在アクティブなガバナーを表示します。
- **--hwlimits** - KHz で CPU で使用可能な周波数の一覧を表示します。

Forcpupower 周波数は、以下のオプションを使用できます。

- **--min <freq>** および **--max <freq>**: KHz で CPU のポリシー制限を設定します。



重要

ポリシーの制限を設定する場合は、**--min** の前に **--max** を設定する必要があります。

- **--freq <freq>** - KHz で CPU に特定のクロック速度を設定します。（**--min** および **--max** ごとに）CPU のポリシー制限内でのみ速度を設定できます。
- **--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、`/sys/devices/system/cpu/[cpuid]/cpufreq/` にある **tunable** で CPUfreq 設定を表示できます。これらの **tunable** に書き込むと、設定や値を変更できます。たとえば、`cpu0` の最小クロック速度を 360 KHz に設定するには、以下を使用します。

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

3.3. CPU モニター

cpuPower 機能は、アイドル状態かつスリープの状態の統計と頻度情報を提供するモニターの選択肢

と、プロセッサトポロジーの報告を行います。一部のモニターはプロセッサ固有であり、その他のモニターはどのプロセッサにも互換性があります。各モニター計測値と、互換性のあるシステムの詳細は、`cpupower-monitor` の `man` ページを参照してください。

`cpupower monitor` コマンドで以下のオプションを使用します。

- `-l` - システムで利用可能な全モニターを一覧表示します。
- `-m <monitor1>, <monitor2>` - 特定のモニターを表示します。その識別子は、`-l` を実行して確認できます。
- `command`: 特定コマンドのアイドル統計および CPU 要求を表示します。

3.4. CPU 省電力ポリシー

`cpuPower` は、プロセッサの省電力ポリシーを制限する方法を提供します。

`cpupower set` コマンドで以下のオプションを使用します。

`--perf-bias <0-15>`

サポートされる Intel プロセッサのソフトウェアが積極的に貢献し、最高のパフォーマンスと省電力のバランスを判断できるようにします。他の省電力ポリシーをオーバーライドしません。割り当てられる値の範囲が 0 から 15 までで、0 はパフォーマンスが最適化され、15 は最高の電力効率になります。

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。個別のコアだけに適用するには、`--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す前に、システムプロセスによる電源の 1 つの CPU パッケージのコアに制限する。0 は制限なし、最初の 1 つの CPU パッケージのみを採用し、タスクウェイクを処理するために半アイドル CPU パッケージを取得する他に、2 はこれを行います。

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

システムプロセスによる電源の使用を、他のコアに描画する前に、1つのCPUコアのスレッドシプリングに制限します。0は制限なし、最初の1つのCPUパッケージのみを採用し、タスクウェイクを処理するために半アイドルCPUパッケージを取得する他に、2はこれを行います。

3.5. 一時停止および再開

システムが一時停止すると、カーネルはドライバーに呼び出しを行い、状態を保存し、アンロードします。システムが再開されると、デバイスの再プログラミングを試行するこれらのドライバーを再読み込みします。このタスクを実行するドライバーの機能は、システムを正常に再開できるかどうかを判断します。

Advanced Configuration and Power Interface (ACPI)仕様では、ビデオハードウェアを再プログラミングできるシステムファームウェアは必要ないため、ビデオドライバーは、これに関する特に問題となります。したがって、ビデオドライバーは完全な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ていない限り、システムが再開できない可能性があります。

Red Hat Enterprise Linux 7には、新しいグラフィックチップセットのサポートが含まれるため、サスペンドおよび再開が多数のプラットフォームで機能されるようになります。特に、NVIDIA チップセットのサポートが大幅に改善されました。特に、GeForce 8800 シリーズに固有のものです。

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、ユーザーに表示される最小の影響で電力消費を減らすのに役立ちます。デバイスが十分な時間でアイドル状態になり、RDPM ハードウェアの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に、デバイスは電力状態になります。低い電源状態からの復旧は、このデバイスの外部 I/O イベントによって認識されます。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、デバイスを稼働状態に戻すことができます。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 設定ファイルに属性を設定することで、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/デバイス/power/` ディレクトリーにあります。device は、特定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 用に RDPM を設定するには、このディレクトリーにアクセスします。

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

デバイスを低い電源状態から実行状態に戻すと、追加のレイテンシーが次の I/O 操作に追加されます。その追加の遅延期間は、デバイスに固有です。ここで説明されている設定スキームでは、システム

管理者はデバイスベースで RDPM を無効にし、その他のパラメーターを調べて制御できます。すべての `/sys/devices/デバイス/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。

コントロール

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、control ファイルの属性に以下の2つの値があります。

auto

すべてのデバイスのデフォルトは、ドライバーに応じて自動 RDPM になる可能性があります。

on

実行時に、ドライバーがデバイスの電源状態を管理しないようにします。

autosuspend_delay_ms

このファイルは、自動サスペンドの遅延を制御します。この遅延は、アイドル状態の状態からデバイスの一時停止間の非アクティブ期間になります。ファイルには、auto-suspend 遅延値（ミリ秒単位）が含まれます。負の値は、デバイスがランタイム時に一時停止されることを防ぐため、`/sys/devices/デバイス/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 を超える値は最も近い秒に丸められます。

3.7. ACTIVE-STATE 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 リンクの少ない電源状態を設定することで、PCI Express または PCIe (PCI Express)サブシステムに電源を保存します。ASPM は、リンクの終了時に両方の電源状態を制御し、リンクの末尾にデバイスが完全に電源オフの状態である場合でも、リンクに電力を節約します。

ASPM を有効にすると、さまざまな電源状態間でリンクの移行に必要な時間が原因で、デバイスのレイテンシーが増加します。ASPM には、電源状態を判断するポリシーが3つあります。

デフォルト

システムのファームウェアが指定するデフォルトに応じて PCIe リンク電源状態を設定します（例：BIOS）。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

ASPM を無効にして、**PCIe** リンクが最大パフォーマンスに使用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** サポートを強制有効または無効にできます。

- **pcie_aspm=off** disables **ASPM**
- **pcie_aspm=force** は、**ASPM** に対応していないデバイスであっても **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、オペレーティングシステムは起動時に **ASPM** を自動的に有効にします。**ASPM** サポートを確認するには、以下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 を実行すると、システムが応答しなくなる

ASPM に対応していないハードウェアで **pcie_aspm=force** を使用して **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、以下のオプションのいずれかを使用します。

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

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. アグレッシブなリンク電源管理

アグレッシブなリンク電源管理 (ALPM) は、アイドル状態の時間帯 (I/O がない場合) に SATA リンクを、ディスクの省電力設定に設定して、ディスクの切り替えに役立つ省電力技術です。ALPM は、I/O 要求がそのリンクキューにキューに入れられると、SATA リンクをアクティブな電源状態に戻します。

ALPM で導入された省電力には、ディスクレイテンシーが経費がかかります。このため、システムがアイドル状態の I/O 時間の長い期間が発生することを想定している場合は、ALPM のみを使用する必要があります。

ALPM は、Advanced Host Controller Interface (AHCI) を使用する SATA コントローラーでのみ利用できます。AHCI に関する詳しい情報は、[を参照してください](#)。

ALPM が利用できる場合には、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、最も低い電源状態 (SLUMBER) へのリンクを設定します。このモードは、アイドル状態の期間が長くなった場合に役に立ちます。

`medium_power`

このモードは、ディスクに I/O がない場合に、2 番目の最も小さい電源状態 (PARTIAL) へのリンクを設定します。このモードは、可能な限り少ない I/O およびアイドル状態の I/O ができる限りリンク電源の状態に移行できるように設計されています (たとえば、断続的な I/O およびアイドル状態の I/O)。

`medium_power` モードでは、負荷に応じて PARTIAL と完全電源 (つまり「ACTIVE」の状態) の間でリンクを移行できます。ACTIVE の状態から移行せずに、直接 PARTIAL から SLUMBER へのリンクを移行することができないことに注意してください。この場合、どちらの電源状態は、ACTIVE の状態から移行されません。

`max_performance`

ALPM は無効になっています。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 ALPM を実際にサポートしているかどうかを確認するには、`/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。設定を変更する場合は、本セクションで説明した値をこれらのファイルに書き込むか、現在の設定を確認するファイルを表示します。



重要： 設定によってはホットプラグを無効にする

ALPM を `min_power` または `medium_power` に設定すると、「Hot Plug」機能が自動的に無効になります。

3.9. 再リロードドライブへのアクセスの最適化

POSIX 標準では、オペレーティングシステムが、各ファイルが最後にアクセスされた時に記録するファイルシステムメタデータを維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、ストレージへの定数一連の書き込み操作が必要になります。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、電源が入ります。いくつかのアプリケーションが `atime` データを使用するため、このストレージデバイスのアクティビティは電源をかけていました。大幅に、ファイルがストレージから読み取られず、キャッシュからファイルが読み取りされていても、ストレージへの書き込みが発生します。場合によっては、Linux カーネルはマウントの `noatime` オプションをサポートし、このオプションでマウントされたファイルシステムに `atime` データを書き込むことはありません。ただし、この機能をオフにするだけでは、一部のアプリケーションが `atime` データに依存するため、問題がないと失敗します。

Red Hat Enterprise Linux 7 で使用されるカーネルは、別の代替 `-relatime` をサポートします。Relatime は `atime` データを維持しますが、ファイルにアクセスされる度は適していません。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てからファイルが変更された場合や、ファイルが最終アクセスされた場合に限り、`atime` データがディスクに書き込まれます（デフォルトでは 1 日）。

デフォルトでは、すべてのファイルシステムは `relatime` が有効な状態でマウントされるようになりました。特定のファイルシステムでは、`norelatime` オプションでファイルシステムをマウントすることで、特定のファイルシステムに対して抑制できます。

3.10. POWER CAPPING

Red Hat Enterprise Linux 7 は、HP Dynamic Power Capping (DPC)、Intel Node Manager (NM) テクノロジーなどの最近のハードウェアにある電源制限機能をサポートしています。電源上限は、管理者がサーバーで消費する能力を制限することができますが、また、既存の電源をオーバーロードするリスクは大幅に軽減されるため、マネージャーがデータセンターをより効果的に計画できます。マネージャーは、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することができ、サーバーの電力消費が抑えられると、負荷の高いロード中の電源は利用可能な電力を越えなくなります。

HP Dynamic Power Capping

動的電源の Capping は、管理者がサーバーの電源消費を限定できる ProLiant および BladeSystem サーバーで利用可能な機能です。上限は、現在のワークロードに関係なく、サーバーが超過しない遅延です。この容量は、サーバーが電力消費制限に達するまで影響を与えません。この時点で、管理プロセッサは CPU の P 状態およびクロックスロットリングを調整して消費した電源を制限します。

動的電源 Capping は、オペレーティングシステムとは独立して CPU の動作を変更しますが、HP の統合 Lights-Out 2 (iLO2)ファームウェアにより、オペレーティングシステムに管理プロセッサにアクセスできるため、ユーザー空間のアプリケーションは管理プロセッサをクエリーできます。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います。これにより、プログラムは `/dev/hpilo/dXccbN` で管理プロセッサをクエリーできます。また、カーネルには、`sysfs` インターフェースを使用する ACPI 4.0 電力メーターの `hwmon` ドライバー、`hwmon` ドライバーも含まれます。これらの機能により、オペレーティングシステムとユーザー空間ツールは、現在のシステムの電力使用量とともに、電源の上限に設定した値を読み取れることができます。

HP Dynamic Power Capping の詳細は、HPE ナレッジ記事『HP ProLiant Gen8 Server Series - Power Capping Settings in iLO 4 および RBSU』を参照してください。

Intel Node Manager

Intel Node Manager は、プロセッサの P-states と T-states を使用してシステムの電源上限を課すため、CPU パフォーマンスを制限するため、電力消費量が消費されます。電源管理ポリシーを設定すると、システムの負荷が不足しているとき（たとえば、夜間や週末など）にシステムが消費できるように設定できます。

Intel Node Manager は、標準の Advanced Configuration and Power Interface を使用して、オペレーティングシステムの直接構成および電源管理(OSPM)を使用して CPU パフォーマンスを調整します。Intel Node Manager が T-states に変更の OSPM ドライバーに通知すると、ドライバーはプロセッサ P 状態に対して対応する変更を加えます。同様に、Intel Node Manager が P-states への変更の OSPM ドライバーに通知すると、そのドライバーは T-states 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追加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 を設定し、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 では、不要な消費ソースを複数削除することで、グラフィックおよびディスプレイデバイスの電源を節約します。

LVDS の再データ

電圧変的な信号 (LVDS)は、回線上的における電子シグナルを実行するシステムです。システムの重要なアプリケーションの 1 つに、ノートブックコンピューターで Liquid crystal display (LCD)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、更新レート（グラフィックコントロー

ラーから新しいデータを受信するレート)と画面のイメージを赤いレートします。通常、画面は1秒ごとに新しいデータ6回(60 Hzの周波数)を受け取ります。画面とグラフィックスコントローラーをLVDSからリンクすると、LVDSシステムは、更新サイクルごとに電源を使用します。アイドルが発生すると、多くのLCD画面の更新レートは、通知可能な効果(cathode ray tube(CRT)のように)モニターが30 Hzにドロップされ、更新レートで減少すると、文字的なファックが表示されるようになります。Red Hat Enterprise Linux 7で使用されているカーネルに組み込まれたIntelグラフィックアダプターのドライバーは、自動的にこのdownclockingを実施し、画面がアイドル状態の場合に約0.5 Wを保存します。

メモリーの自己更新の有効化

同期動的アクセスメモリー(SDRAM)-グラフィックアダプターのビデオメモリーに使用される時間毎秒単位で何回か減り、個々のメモリーセルが格納されているデータを保持できるようにします。メモリー内内のデータフローとしてデータを管理する主な機能のほかに、メモリーコントローラーは通常、これらの更新サイクルを開始します。ただし、SDRAMには低電力更新モードもあります。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更新サイクルを生成します。これにより、システムは、現在メモリーに保持されるデータなしで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、アイドル状態のときにIntelグラフィックアダプターでメモリーの自己更新をトリガーできます。これにより、0.8 Wで保存されます。

GPUクロックの削減

通常のグラフィカル処理ユニット(GPU)には、内部サーキットのさまざまな部分を管理する内部クロックが含まれます。Red Hat Enterprise Linux 7で使用されるカーネルは、IntelおよびATI GPUの内部クロックの一部の頻度を減らすことができます。GPUコンポーネントが特定の時間で実行するサイクルの数を減らすと、実行する必要がないサイクルで消費した電力が節約されます。カーネルは、GPUがアイドル状態のときに、このクロックの速度を自動的に減らし、GPUアクティビティーが増えたときに増加します。GPUクロックサイクルを短くすると、最大5 Wを保存できます。

GPU省電力

Red Hat Enterprise Linux 7のIntelおよびATIグラフィックドライバーは、モニターがアダプターに接続されているタイミングを検出できるため、GPUを完全にシャットダウンします。この機能は、定期的にアタッチされていないサーバーについて特に重要です。

3.12. RFKILL

多くのコンピューターシステムには、Wi-Fi、Bluetooth、GGデバイスなどのラジオメンタターが含まれています。このデバイスは、デバイスが使用されていないときに無駄な消費電力を消費します。

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にラッチベッターをクエリー、アクティベート、非アクティブ化できるインターフェースを提供します。送信先が非アクティブになる場合は、ソフトウェアがリアクティブな(ソフトブロック)やソフトウェアが反応できない(ハードブロック)状態で配置できます。

RFKill コアは、サブシステムの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方法を追加します。また、RFKill コアは、ユーザーアプリケーションが送信者の状態をクエリーするためのユーザーアプリケーションが解釈できる通知を提供します。

RFKill インターフェースは /dev/rfkill にあります。これには、システム上のすべての無線送信機能の現在の状態が含まれます。各デバイスには、sysfs に現在の RFKill 状態が登録されています。また、RFKill 対応デバイスで状態が変更されるたびに、RFKill issues uevents を発行します。

rfkill は、システムで RFKill 対応のデバイスをクエリーし、変更できるコマンドラインツールです。ツールを取得するには、rfkill パッケージをインストールします。

コマンド rfkill 一覧を使用してデバイスの一覧を取得します。それぞれのデバイス一覧には 0 から開始され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 に対してデバイスをブロックしたり、ブロックを解除したりできます。

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

システムの最初の RFKill 対応デバイスをブロックします。

rfkill を使用して、デバイスの特定の Kategorie 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
```

システムの Wi-Fi デバイスをすべてブロックします。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
```

ブロックデバイスを解除するには、rfkill ブロックの代わりに 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詳細一覧を取得するには、rfkill help を実行します。

第4章 使用例

本章では、2種類のユースケースについて説明します。この2つの種類のユースケースで、本ガイドの他の項で説明されている分析方法および設定方法を説明します。最初の例では、通常のサーバーが考慮され、2つ目は通常のラップトップです。

4.1. 例：サーバー

通常の標準サーバーには、Red Hat Enterprise Linux 7 でサポートされる必要なハードウェア機能の大半が同梱されています。最初に考慮することが、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、省電力に最適化できるコンポーネントを決定できます。

サーバーのタイプに関係なく、グラフィックのパフォーマンスは必須ではありません。したがって、GPU 省電力は、オンのままにすることができます。

Webserver

Web サーバーにはネットワークおよびディスク I/O が必要です。外部接続速度 100 Mbit/s によっては十分である場合があります。マシンがほとんどの静的ページに対応している場合、CPU パフォーマンスは非常に重要な可能性があります。したがって、電源管理の選択肢には以下が含まれます。

- tuned 用のディスクやネットワークプラグインはありません。
- ALPM がオンになります。
- オンデマンドガバナーがオンになります。
- ネットワークカードは、100 Mbit/s に制限されます。

コンピュータサーバー

コンピュータサーバーには、主に CPU が必要です。電源管理の選択肢には、以下が含まれます。

- ジョブやデータストレージが発生した場所、tuned 向け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（バッチモードシステムの場合）に応じて、完全にアクティブな Tuned。

- 使用率によっては、パフォーマンスガバナーなどに応じて異なります。

Mailserver

メールサーバーには、主にディスク I/O および CPU が必要です。電源管理の選択肢には、以下が含まれます。

- 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ーがオンになります。
- tuned 用のディスクやネットワークプラグインはありません。
- メールは内部にあり、1 Gbit/s または 10 Gbit/s リンクの利点を活用できるため、ネットワーク速度は制限しないでください。

Fileserver

Fileserver 要件はメールサーバーのものに似ていますが、使用されるプロトコルによっては CPU パフォーマンスがさらに必要になる場合があります。通常、Samba ベースのサーバーでは NFS を超える CPU が必要です。NFS は通常、複数の iSCSI が必要です。それでも、オンデマンドガバナーを使用することができるはずです。

ディレクトリーサーバー

ディレクトリーサーバーは通常、特に RAM を十分に持つ場合にディスク I/O の要件が低くなっています。ネットワークの I/O はより小さくても、ネットワークレイテンシーは重要です。リンク速度が低い状態でレイテンシーのネットワークチューニングを検討することはできますが、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 : LAPTOP

電源管理や保存が本当に移動するような重要な場所が、ノートパソコンがノートパソコンになります。設計により、ノートパソコンは、すでにワークステーションやサーバーで、絶対保存の可能性が他のマシンにとっても少なくなっています。ただし、バッテリーモードでは、あらゆる保存で、ラップトップからバッテリーの寿命を数分進めることができます。本セクションでは、バッテリーモードにおけるラップトップに重点を置けていますが、AC 電源で実行している間に、一部のチューニングやすべての調整を使用できます。

単一コンポーネントの保存は通常、ワークステーションで実行する場合よりもノートパソコンに対してより相対的な違いになります。たとえば、100 Mbits/s ネットワークインターフェースが 100 Mbits/s

で実行されている 1 Gbit/s ネットワークインターフェースは、約 3-4 の watts を保存します。総電力が 400 の約 400 の電力消費量を備えた通常のサーバーの場合、この節約は約 1% になります。40 の総消費電力を持つラップトップでは、この 1 つのコンポーネントだけの省電力だけを、合計 10% に削減します。

一般的なラップトップにおける具体的な省電力の最適化には、以下が含まれます。

- 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、並列またはシリアルポート、カードリーダー、Webcams、WiFi、および Bluetooth はいくつかの候補に名前を付けるだけです。
- DIM は darker 環境でディスプレイを読み取れず、画面の成熟的な読み取る必要がない。GNOME+System+Preferences 電源管理、Kickoff Application Launcher+Computer+System Settings+Advanced → Power Management、コマンドライン上の `gnome-power-manager` または `xbacklight`、またはラップトップの関数キーを使用します。

また、さまざまなシステム設定に対して、多くの調整を実施することもできます。

- オンデマンドガバナナー (Red Hat Enterprise Linux 7 でデフォルトで有効) を使用
- AC97 音声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ります)。

```
~]# 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 デバイスでは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 でデフォルト)。

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

- **画面の明るさを 50 以下の方まで下げます。以下に例を示します。**

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

- **画面アイドルのために DPMS をアクティブにします。**

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

- **Wi-Fi を非アクティブにします。**

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

付録A 開発者に関するヒント

適切なプログラミングテキストブックはすべて、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を説明します。ソフトウェアを開発する際には、ソフトウェアが実行されるシステムの消費電力を向上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、コードごとに影響を与えませんが、パフォーマンスでボトルネックとなるエリアでコードを最適化することができます。

問題が生じるいくつかの手法には、以下が含まれます。

- スレッドの使用
- ウェイクアップを効率的に使用せず、不要な CPU ウェイクアップも使用しません。ウェイクアップを実行する必要がある場合は、すべてを一度に（アイドル状態に）、できるだけ早期に実施してください。
- `[f]sync()` を不必要に使用している。
- 不要なアクティブなポーリングまたは短期タイムアウトの使用。（代わりに `events` に反応）
- ウェイクアップを効率的に使用していません。
- 非効率なディスクアクセス。ディスクアクセスが頻繁に行われないように、大きなバッファを使用します。一度に 1 つの大きなブロックを書き込みます。
- タイマーの非効率の使用。可能であれば、複数のアプリケーション（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 はグローバルロックインタプリターを使用します。[1]、スレッドは大規模な I/O 操作のみを対象とします。Unladen-swallow[2] は、コードを最適化する可能性のある Python を迅速に実装しています。

perl

Perl スレッドは、最初にフォークなしでシステムで実行されるアプリケーション用に作成されました (32 ビットの Windows オペレーティングシステムがあるシステムなど)。Perl スレッドでは、データはすべての単一スレッド(Copy On Write)に対してコピーされます。ユーザーはデータ共有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。threads::shared モジュールを保存するには、含める必要があります。ただし、データはコピーされません(Copy On Write)。ただし、モジュールはデータの関連する変数も作成します。これはさらに時間がかかり、時間がかかります。[3]

C

C スレッドは同じメモリーを共有します。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、新しいメモリー領域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より多くの CPU のサポートを参照してください。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++ などの低レベルの言語を使用します。スクリプト言語を使用する場合は、C バインディングの作成を検討してください。プロファイラーを使用して、コードの一部を不適切に実行することを特定します。[4]

A.2. WAKE-UPS

多くのアプリケーションは、変更のために設定ファイルをスキャンします。多くの場合、スキャンは 1 分ごとに固定の間隔で実行されます。これは、ディスクがスピンドウンから起動するよう強制するため、問題になる可能性があります。最適なソリューションとしては、適切な間隔、適切なチェックメカニズム、または inotify の変更を確認し、イベントに反応させることです。inotify は、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 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、システムで利用できる監視の数に制限がある点です。数値は `/proc/sys/fs/inotify/max_user_watches` から取得可能ですが、これを変更することはできません。さらに、`inotify` が失敗すると、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。通常、これはソースコードの `#if #define` の多数を意味します。

`inotify` の詳細については、`inotify(7)` の `man` ページを参照してください。

A.3. FSYNC

`Fsync` I/O コストのかかる操作と呼ばれますが、これは完全に `true` ではありません。

ユーザーがリンクをクリックすると、新たなページに移動するたびに `sqlite` ライブラリーを呼び出すために使用される `Firefox`。 `fsync` と呼ばれ、ファイルシステムの設定（主にデータ順モードを含む `ext3`）のために、何も生じなかったときのレイテンシーが長くなっていました。別のプロセスが同時に大きなファイルをコピーする場合は、時間がかかることがあります（最大 30 秒）。

ただし、`fsync` が全く使用されていないその他の状況では、`ext4` ファイルシステムへの切り替えと共に問題が発生することがあります。`ext3` は、データの順序付けモードに設定されました。このモードは、数秒ごとにメモリーをフラッシュし、これをディスクに保存しました。`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におつされたときにデータが失われる可能性があります。これで `ext4` にはパッチが当てられますが、アプリケーションを慎重に設計することを検討し、必要に応じて `fsync` を使用する必要があります。

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の作成方法とデータが失われる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);
```

次のような適切なアプローチは以下のとおりです。

```
/* 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-interpretor-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 2.2-9 7.7 GA 公開用ドキュメントバージョン	Mon Aug 05 2019	Marie Doleželová
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 非同期更新： Tuned 章の再書き込み	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 リリース向けのバージョン	Wed 11 Nov 2015	Jana Heves
改訂 1-3 コアインフラストラクチャーおよび Mechanics のパッケージ名が正しくないように修正されました。	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 向けのバージョン	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 スタイルの変更に再構築。	Fri May 9 2014	Yoana Ruseva
改訂 0.9-0 確認する目的で、Red Hat Enterprise Linux 7.0 リリースの書籍です。	Wed May 7 2014	Yoana Ruseva
改訂 0.1-1 Red Hat Enterprise Linux 6 バージョンのドキュメントのブランチ	Thu Jan 17 2013	Jack Reed