Red Hat Enterprise Linux 6

Handbuch zur Energieverwaltung

Verwaltung des Energieverbrauchs unter Red Hat Enterprise Linux 6

Ausgabe 1.0

Last Updated: 2017-10-12

Red Hat Enterprise Linux 6 Handbuch zur Energieverwaltung

Verwaltung des Energieverbrauchs unter Red Hat Enterprise Linux 6
Ausgabe 1.0

Don Domingo
Red Hat Engineering Content Services

Rüdiger Landmann
Red Hat Engineering Content Services
r.landmann@redhat.com

Red Hat Inc.

Rechtlicher Hinweis

Zusammenfassung

Dieses Dokument erklärt, wie die Energieverwaltung unter Red Hat Enterprise Linux 6 Systemen effektiv verwaltet werden kann. Die folgenden Abschnitte behandeln verschiedene Techniken zur Verringerung des Energieverbrauchs (sowohl für Server, also auch für Laptops) und wie sich jede dieser Techniken auf die Gesamtleistung Ihres Systems auswirken.

Inhaltsverzeichnis

KAPITEL 1. ÜBERBLICK

Energieverwaltung war einer der Schwerpunkte für Verbesserungen in Red Hat Enterprise Linux 6. Das Senken des Stromverbrauchs von Computer-Systemen ist einer der wichtigsten Aspekte der *Green IT* (umweltfreundliches Betreiben von Computer-Systemen), welches wesentlich tiefer geht und Probleme wie recyclebare Materialien, die umweltfreundliche Produktion von Systemen sowie ordnungsgemäßes Design und Planen von umweltfreundlichen Systemen beinhaltet. In diesem Dokument liefern wir Hilfestellungen und Informationen bezüglich der Energieverwaltung Ihrer Systeme, auf denen Red Hat Enterprise Linux 6 läuft.

1.1. BEDEUTUNG VON ENERGIEVERWALTUNG

Ein zentrales Element der Energieverwaltung ist ein Verständnis dafür, wie der Energieverbrauch jeder Systemkomponente effektiv optimiert werden kann. Folglich werden die verschiedenen Aufgaben, die Ihr System ausführt, studiert und jede Komponente so konfiguriert, dass sichergestellt ist, dass ihre Performanz genau mit dem Job abgestimmt ist.

Ein Haupt-Motivator für Energieverwaltung ist:

- Reduzierung des Gesamt-Stromverbrauchs zur Kostensenkung

Ordnungsgemäße Umsetzung von Energieverwaltung resultiert in:

- Wärmereduzierung für Server und Rechenzentren
- verminderte Zweitkosten, inklusive Kühlung, Platz, Kabel, Generatoren und *Uninterruptible Power Supplies* (UPS)
- verlängerte Lebensdauer für Akkus für Laptops
- niedrigerer Kohlendioxid-Ausstoß
- Einhaltung von Regulierungen oder legalen Anforderungen in Bezug auf Green-IT seitens Regierungen, z.B. Energy Star
- Umsetzung von Unternehmens-Richtlinien für neue Systeme

In der Regel führt die Reduzierung des Stromverbrauchs einer speziellen Komponente (oder des Systems als Ganzes) zu weniger Wärme und natürlich Performanz. Daher sollten Sie die Einbußen bei der Leistung, die durch irgendwelche Konfigurationen, die Sie durchführen, geschaffen werden, gründlich studieren und testen, besonders für unternehmenskritische Systeme.

Indem Sie die verschiedenen Aufgaben, die Ihr System durchführt, untersuchen und jede Komponente so konfigurieren, dass gewährleistet ist, dass seine Leistung genau für die Aufgabe ausreicht, können Sie Energie sparen, weniger Wärme erzeugen und die Lebensdauer von Akkus für Laptops optimieren. Viele Prinzipien zur Analyse und Abstimmung eines Systems bezüglich des Stromverbrauchs entsprechen denen der Leistungsoptimierung. Bis zu einem gewissen Grad widersprechen sich Energieverwaltung und Leistungsoptimierung bei der Systemkonfiguration, da Systeme entweder in Sachen Leistung oder Strom optimiert sind. Dieses Handbuch beschreibt die von Red Hat zur Verfügung gestellten Werkzeuge und die Techniken, die wir zu Ihrer Unterstützung bei diesem Prozess entwickelt haben.

Red Hat Enterprise Linux 6 kommt bereits mit einigen neuen Features zur Energieverwaltung, die standardmäßig aktiviert sind. Sie wurden alle selektiv ausgewählt, um die Leistung eines typischen Server- oder Desktop-Anwendungsfalls nicht zu beeinträchtigen. Für sehr spezielle Anwendungsfälle

jedoch, bei denen maximaler Durchsatz, niedrigste Latenz oder höchste CPU-Leistung absolut erforderlich sind, ist eine Nachbearbeitung dieser Standardvorgaben ggf. notwendig.

Fragen Sie sich die nachfolgend aufgeführten Fragen, um zu entscheiden, ob Sie Ihre Maschinen unter Verwendung der in diesem Dokument beschriebenen Techniken optimieren sollten:

F: Muss ich optimieren?

A: Die Bedeutung von Strom-Optimierung hängt davon ab, ob Ihr Unternehmen Leitfäden besitzt, denen Sie folgen müssen, oder ob es irgendwelche Regulierungen gibt, die Sie einhalten müssen.

F: Wie viel muss ich optimieren?

A: Einige der von uns präsentierten Techniken erfordern keinen kompletten Durchlauf hinsichtlich Auditing und Analyse Ihrer Maschine bis ins Detail, sondern liefern stattdessen eine allgemeine Optimierung, die üblicherweise den Stromverbrauch reduziert. Sie sind natürlich nicht typischerweise so gut, wie manuell geprüfte und optimierte Systeme, aber bieten einen guten Kompromiss.

F: Reduziert die Optimierung die Systemleistung auf ein nicht akzeptables Level?

A: Die meisten in diesem Dokument beschriebenen Techniken haben beträchtliche Auswirkungen auf Ihr System. Falls Sie beabsichtigen, Energieverwaltung über die im Rahmen von Red Hat Enterprise Linux 6 gelieferten Vorgaben hinaus zu implementieren, sollten Sie die Leistung des Systems nach der Strom-Optimierung überwachen und entscheiden, ob der Leistungsabfall akzeptabel ist.

F: Übertrifft der Aufwand bei der Optimierung hinsichtlich Zeit und Ressourcen die erzielten Gewinne?

A: Die manuelle Optimierung eines einzelnen Systems unter Durchführung des kompletten Prozesses lohnt sich üblicherweise nicht, da die Zeit und die Kosten dafür wesentlich höher liegen, als der eigentliche Gewinn, den Sie im Rahmen der Lebensdauer einer einzelnen Maschine erhalten. Wenn Sie andererseits 10000 Desktop-Systeme in Ihren Büros ausliefern und alle dieselbe Konfiguration und Einstellung verwenden, dann ist das Erstellen einer optimierten Einstellung und deren Anwendung auf alle 10000 Maschinen ganz sicher eine gute Idee.

Die folgenden Abschnitte erklären, wie sich optimierte Hardware-Performanz positiv auf den Energieverbrauch Ihres Systems auswirkt.

1.2. GRUNDLAGEN ZUR ENERGIEVERWALTUNG

Effektive Energieverwaltung basiert auf den folgenden Prinzipien:

Eine CPU im Leerlauf sollte nur dann aktiv werden, wenn nötig

Der Red Hat Enterprise Linux 5 Kernel verwendete einen periodischen Taktgeber für jede CPU. Dieser Taktgeber verhindert, dass sich die CPU tatsächlich in den Leerlauf versetzt, da die CPU jede Anfrage des Taktgebers verarbeiten muss (was alle paar Millisekunden passieren würde, je nach Einstellung), unabhängig davon, ob irgendein Prozess lief oder nicht. Ein großer Teil effizienter Energieverwaltung umfasst die Reduzierung der Frequenz, bei der CPU-Wakeups initiiert werden.

Aus diesem Grund schließt der Linux-Kernel in Red Hat Enterprise Linux 6 den periodischen Taktgeber aus: folglich ist der Leerlauf-Status einer CPU nun *tickless*. So wird verhindert, dass die CPU unnötig Strom verbraucht, wenn sie sich im Leerlauf befindet. Der Nutzen dieses Features kann jedoch außer Kraft gesetzt werden, wenn Ihr System Anwendungen besitzt, die unnötige Taktgeber-Ereignisse erstellen. "Polling"-Ereignisse (wie beispielsweise Checks auf Volume-Änderungen, Mausbewegung und dergleichen) sind Beispiele für solche Ereignisse.

Red Hat Enterprise Linux 6 beinhaltet Werkzeuge, mit denen Sie Anwendungen auf Basis ihres CPU-Verbrauchs prüfen und identifizieren können. Werfen Sie einen Blick auf Kapitel 2, *Energieverwaltung - Auditing und Analyse* für Details.

Nicht benutzte Hardware und Geräte sollten komplett deaktiviert werden

Dies trifft besonders auf Geräte zu, die bewegliche Teile besitzen (z.B. Festplatten). Darüber hinaus halten einige Anwendungen die Verbindung zu einem nicht benutzten, jedoch aktivierten Gerät "open". Wenn dies passiert, nimmt der Kernel an, dass das Gerät in Gebrauch ist, was verhindert, dass das Gerät in einen Energiesparmodus versetzt wird.

Geringe Aktivität sollte in niedrigen Wattverbrauch umgesetzt werden

In vielen Fällen hängt dies jedoch von moderner Hardware und einer korrekten BIOS-Konfiguration ab. Ältere Systemkomponenten unterstützen oft einige der neuen Features, die jetzt unter Red Hat Enterprise Linux 6 unterstützt werden, nicht. Stellen Sie sicher, dass Sie die aktuellste, offizielle Firmware für Ihre Systeme verwenden und dass die Energieverwaltungs-Features in den Abschnitten 'Energieverwaltung' oder 'Gerätekonfiguration' des BIOS aktiviert sind. Einige Features, auf die geachtet werden sollte, sind:

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

Falls Ihre Hardware diese Features unterstützt und sie im BIOS aktiviert sind, verwendet Red Hat Enterprise Linux 6 sie standardmäßig.

Verschiedene Arten von CPU-Zuständen und Ihre Auswirkungen

Moderne CPUs liefern im Zusammenhang mit *Advanced Configuration and Power Interface*(ACPI) verschiedene Strom-Zustände. Die drei verschiedenen Zustände sind:

- Sleep (C-Zustände)
- Frequency (P-Zustände)
- Wärmeausgabe (T-Zustände oder "Thermal States")

Eine CPU, die im niedrigsten verfügbaren Schlafzustand betrieben wird, verbraucht die geringste Menge an Watt. Es wird jedoch auch beträchtlich mehr Zeit benötigt, um sie bei Bedarf aus diesem Zustand zu aktivieren. In einigen seltenen Fällen kann dies dazu führen, dass die CPU jedes Mal, wenn sie gerade in den Schlafzustand versetzt wurde, umgehend wieder aktiviert werden muss. Diese Situation resultiert in einer permanent beschäftigten CPU und verliert einiges an der möglichen Energieeinsparung, falls ein anderer Zustand verwendet wurde.

Eine abgeschaltete Maschine verbraucht die geringste Menge Strom.

So offensichtlich dies erscheinen mag - einer der besten Wege, um tatsächlich Strom zu sparen, ist das Abschalten eines Systems. Ihr Unternehmen kann beispielsweise eine Unternehmenskultur entwickeln, die auf dem Bewusstsein von "Green IT" ausgerichtet ist und eine Richtlinie zur Abschaltung von Maschinen während der Mittagspause oder nach Feierabend liefert. Sie können außerdem in Betracht ziehen, mehrere physikalische Server in einen größeren Server zusammenzufassen und mit Hilfe der Virtualisierungstechnologie, die wir mit Red Hat Enterprise Linux 6 ausliefern, zu virtualisieren.

KAPITEL 2. ENERGIEVERWALTUNG - AUDITING UND ANALYSE

2.1. ÜBERBLICK ÜBER AUDIT UND ANALYSE

Der detaillierte manuelle Audit, Analyse und Abstimmung eines einzelnen Systems ist normalerweise die Ausnahme, da die investierte Zeit und die Kosten üblicherweise den Gewinn dieser abschließenden Systemanpassung aufheben. Dagegen kann das einmalige Durchführen dieser Aufgaben für eine große Anzahl von fast identischen Systemen, auf denen Sie die gleichen Einstellungen für alle Systeme verwenden können, sehr hilfreich sein. Bedenken Sie beispielsweise den Einsatz tausender Desktop-Systeme oder eines HPC-Clusters, wo die Maschinen fast identisch sind. Ein weiterer Grund für die Durchführung von Auditing und Analyse ist das Schaffen einer Vergleichsbasis, anhand derer Sie Regressionen oder Änderungen des Systemverhaltens in der Zukunft identifizieren können. Die Ergebnisse dieser Analyse können sehr hilfreich in solchen Fällen sein, in denen regelmäßig Hardware-, BIOS oder Software-Aktualisierungen durchgeführt werden und sie Überraschungen in Bezug auf den Energieverbrauch vermeiden möchten. Allgemein liefern Ihnen ein gründliches Audit und eine gründliche Analyse eine sehr viel bessere Idee, was tatsächlich auf einem System passiert.

Das Auditing und die Analyse eines Systems in Bezug auf Stromverbrauch ist recht schwierig, auch mit den aktuellsten System, die verfügbar sind. Die meisten Systeme liefern keine nötigen Mittel zum Messen des Stromverbrauchs via Software. Es gibt jedoch Ausnahmen: die ILO-Management-Konsole von Hewlett Packard Server-Systemen bieten ein Modul zur Energieverwaltung, auf das Sie via Web zugreifen können. IBM bietet eine ähnliche Lösung in ihren BladeCenter Modul zur Energieverwaltung. Auf einigen Dell-Systemen bietet auch der IT-Assistant Möglichkeiten zur Energie-Überwachung. Andere Hersteller bieten wahrscheinlich ähnliche Optionen auf ihren Server-Plattformen, aber wie man sieht, gibt es keine einheitliche Lösung, die von allen Herstellern unterstützt wird. Falls Ihr System keine eingebauten Mechanismen zum Messen des Stromverbrauchs besitzen, gibt es ein paar andere Möglichkeiten. Sie könnten beispielsweise ein spezielles Netzteil für Ihr System installieren, welches Informationen zum Stromverbrauch via USB preisgibt. Das Netzteil Gigabyte Odin GT 550 W ist ein solches Beispiel und Software zum Auslesen dieser Werte unter Linux ist extern unter http://mgmt.sth.sze.hu/~andras/dev/gopsu/ erhältlich. Als letzter Ausweg besitzen einige externe Watt-Messgeräte wie das Watts up? PRO einen USB-Anschluß.

Das direkte Messen des Stromverbrauchs ist oft nur zur optimalen Maximierung der Einsparungen notwendig. Zum Glück stehen andere Möglichkeiten zur Verfügung, um Änderungen effektiv sind oder wie sich das System verhält. Dieses Kapitel beschreibt die notwendigen Werkzeuge.

2.2. POWERTOP

Die Einführung des "tickless" Kernels in Red Hat Enterprise Linux 6 (siehe Abschnitt 3.4, »Tickless-Kernel«) ermöglicht es der CPU, den Leerlauf-Zustand häufiger zu belegen, was den Stromverbrauch senkt und die Energieverwaltung verbessert. Das neue **PowerTOP**-Werkzeug ermittelt bestimmte Komponenten von Kernel- und Userspace-Anwendungen, die die CPU häufig aktivieren. **PowerTOP** wurde in der Entwicklung verwendet, um die in Abschnitt 3.11, »Optimierungen im User Space« beschriebenen Audits durchzuführen, welches zu einer Feinabstimmung vieler Anwendungen in diesem Release führte und unnötige Aktivierungen der CPU um ein Zehnfaches reduzierte.

Installieren Sie **PowerTOP** mit dem folgenden Befehl:

```
yum install powertop
```

Führen Sie **PowerTOP** mit dem folgenden Befehl aus:

```
powertop
```

Beachten Sie bitte, dass Sie **PowerTOP** mit Root-Privilegien ausführen müssen, damit die Anwendung ordnungsgemäß funktioniert.

Bei der Ausführung sammelt **PowerTOP** Statistiken vom System und präsentiert Ihnen eine Liste der Komponenten, welche am häufigsten "wakeups" an die CPU senden. **PowerTOP** liefert außerdem Vorschläge zur Abstimmung des Systems für einen niedrigeren Stromverbrauch. Diese Vorschläge werden am unteren Bereich des Bildschirms angezeigt und definieren eine Tastaturbelegung, die Sie drücken können, um die Vorschläge von **PowerTOP** anzunehmen. Da sich **PowerTOP** in regelmäßigen Abständen aktualisiert, werden weitere Vorschläge angezeigt. Beachten Sie unter Abbildung 2.1, »PowerTOP in Betrieb« den Vorschlag, **die VM Dirty Writeback-Zeit zu erhöhen**, und die Tastaturbelegung (**W**), um den Vorschlag zu akzeptieren.

Bei der Ausführung sammelt **PowerTOP** Statistiken vom System und präsentiert Ihnen mehrere wichtige Listen mit Informationen. Ganz oben befindet sich eine Liste, in der angezeigt wird, wie lange sich Ihre CPU-Kerne in jedem der verfügbaren C- und P-Zuständen befanden. Je länger sich die CPU in den höheren C- oder P-Zuständen befindet, desto besser (wobei **C4** höher ist, als **C3**). Gleichzeitig ist dies ein guter Indikator dafür, wie gut das System bezüglich der CPU-Verwendung abgestimmt ist. Ihr Ziel sollte ein Wert von 90% oder höher im höchsten C- oder P-Zustand sein, wenn das System sich im Leerlaufbetrieb befindet.

Der zweite Teil der Information ist eine Zusammenfassung der tatsächlichen Wakeups der Maschine pro Sekunde. Die Anzahl der Wakeups pro Sekunde gibt Ihnen einen guten Anhaltspunkt, wie gut die Dienste oder Geräte und Treiber des Kernels bezüglich des Stromverbrauchs auf Ihrem System funktionieren. Je mehr Wakeups pro Sekunde Sie haben, desto mehr Strom wird verbraucht. Weniger ist demnach besser an dieser Stelle.

Als nächstes liefert **PowerTOP** eine Schätzung des tatsächlichen Stromverbrauchs des Systems, falls verfügbar. Rechnen Sie damit, dass **PowerTOP** diese Zahlen auf Laptops liefert, die im Batteriebetrieb laufen.

Jeglichen verfügbaren Schätzungen des Stromverbrauchs folgt eine detaillierte Liste der Komponenten, die am häufigsten Wakeups an die CPU senden. An der Spitze der Liste befinden sich die Komponenten, die Sie genauer untersuchen sollten, um Ihr System bei der Reduzierung des Stromverbrauchs zu optimieren. Falls es sich bei ihnen um Kernel-Komponenten handelt (ersichtlich am Namen der in **<>** aufgelisteten Komponente), dann sind die Wakeups oft mit einem speziellen Treiber verknüpft, der sie auslöst. Das Abstimmen von Treibern erfordert für gewöhnlich Änderungen am Kernel, die über das Ausmaß dieses Dokuments hinausgehen. Userland-Prozesse, die Wakeups senden, können jedoch einfacher verwaltet werden. Identifizieren Sie zunächst, ob dieser Dienst oder diese Anwendung überhaupt auf diesem System laufen muss. Falls nicht, deaktivieren Sie diesen/sie einfach. Um einen Dienst permanent zu deaktivieren, führen Sie folgenden Befehl aus:

```
chkconfig servicename off
```

Führen Sie Folgendes aus, wenn Sie mehr Informationen darüber benötigen, was die Komponente tatsächlich tut:

```
ps -awux | grep componentname
strace -p processid
```

Falls es so scheint, als ob sich der Trace wiederholt, haben Sie ggf. eine "busy loop" entdeckt. Eine Behebung würde eine Code-Änderung in dieser Komponente erfordern, was wiederum über den Rahmen dieses Dokuments hinausgeht.

Abschließend liefert **PowerTOP** liefert außerdem Vorschläge zur Abstimmung des Systems für einen

niedrigeren Stromverbrauch. Diese Vorschläge werden am unteren Bereich des Bildschirms angezeigt und definieren eine Tastaturbelegung, die Sie drücken können, um die Vorschläge von **PowerTOP** anzunehmen. Da sich **PowerTOP** in regelmäßigen Abständen aktualisiert, werden weitere Vorschläge angezeigt. Beachten Sie unter Abbildung 2.1, »PowerTOP in Betrieb« den Vorschlag, **`die VM Dirty Writeback-Zeit zu erhöhen`**, und die Tastaturbelegung (**`W`**), um den Vorschlag zu akzeptieren.

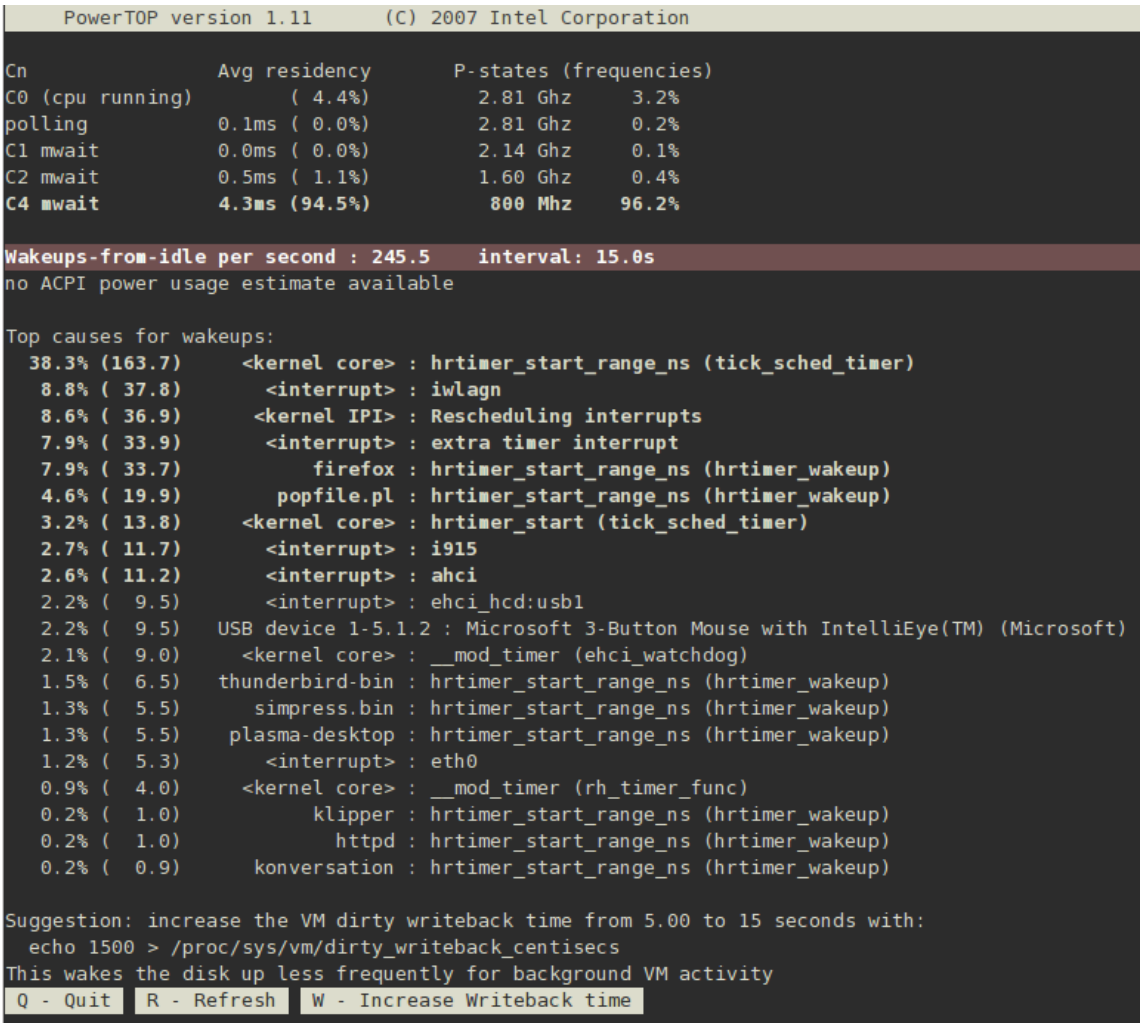


Abbildung 2.1. PowerTOP in Betrieb

Die Website *Less Watts* veröffentlicht eine Liste von Anwendungen, die von **PowerTOP** als solche Anwendungen ermittelt wurden, die die CPU aktiv halten. Sie auch http://www.lesswatts.org/projects/powertop/known.php.

2.3. DISKDEVSTAT UND NETDEVSTAT

Diskdevstat und **netdevstat** sind **SystemTap**-Werkzeuge, die detaillierte Informationen zur Platten- und Netzwerkaktivität aller auf einem System laufenden Anwendungen sammeln. Diese Werkzeuge wurden durch **PowerTOP** inspiriert, welches die Anzahl der CPU-Wakeups von jeder Anwendung pro Sekunde anzeigt (siehe Abschnitt 2.2, »PowerTOP«). Mit Hilfe dieser Statistiken, die diese Werkzeuge sammeln, können Sie Anwendungen identifizieren, die Energie mit vielen kleinen I/O-Operationen verschwenden, anstatt weniger größere Operationen zu verwenden. Andere Werkzeuge zur Überwachung, die nur die Übertragungsraten messen, helfen bei dieser Art der Verwendung nicht.

Installieren Sie diese Werkzeuge mit **SystemTap** mit dem Befehl:

```
yum install systemtap tuned-utils kernel-debuginfo
```

Führen Sie die Werkzeuge mit folgendem Befehl aus:

```
diskdevstat
```

oder dem Befehl:

```
netdevstat
```

Beide Befehle können bis zu drei Parameter, wie folgt, annehmen:

diskdevstat ***update_interval total_duration display_histogram***

netdevstat ***update_interval total_duration display_histogram***

update_interval

Die Zeit in Sekunden zwischen Aktualisierung der Anzeige. Standardwert: **5**

total_duration

Die Zeit in Sekunden für den gesamten Durchlauf. Standardwert: **86400** (1 Tag)

display_histogram

Ein Flag zum Erstellen eines Histogramms aus allen gesammelten Daten am Ende eines Durchlaufs.

Die Ausgabe ähnelt der von **PowerTOP**. Nachfolgend ist eine Beispiel-Ausgabe eines längeren **diskdevstat**-Durchlaufs auf einem Fedora 10 System mit KDE 4.2 aufgeführt:

```
  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
15494     0 sda1             0     0.000     0.000     0.000         758
0.000     0.012     0.000 0logwatch
15520     0 sda1             0     0.000     0.000     0.000         140
0.000     0.009     0.000 perl
15549     0 sda1             0     0.000     0.000     0.000         140
0.000     0.009     0.000 perl
15585     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
15429     0 sda1             0     0.000     0.000     0.000          62
0.009     0.009     0.000 crond
15379     0 sda1             0     0.000     0.000     0.000          62
0.008     0.008     0.000 crond
15473     0 sda1             0     0.000     0.000     0.000          62
0.008     0.008     0.000 crond
15415     0 sda1             0     0.000     0.000     0.000          62
0.008     0.008     0.000 crond
15433     0 sda1             0     0.000     0.000     0.000          62
0.008     0.008     0.000 crond
15425     0 sda1             0     0.000     0.000     0.000          62
0.007     0.007     0.000 crond
15375     0 sda1             0     0.000     0.000     0.000          62
```

```
0.008     0.008     0.000 crond
15477     0 sda1            0     0.000     0.000     0.000           62
0.007     0.007     0.000 crond
15469     0 sda1            0     0.000     0.000     0.000           62
0.007     0.007     0.000 crond
15419     0 sda1            0     0.000     0.000     0.000           62
0.008     0.008     0.000 crond
15481     0 sda1            0     0.000     0.000     0.000           61
0.000     0.001     0.000 crond
15355     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
15575     0 sda1            0     0.000     0.000     0.000           16
0.000     0.000     0.000 cat
15581     0 sda1            0     0.000     0.000     0.000           12
0.001     0.002     0.000 perl
15582     0 sda1            0     0.000     0.000     0.000           12
0.001     0.002     0.000 perl
15579     0 sda1            0     0.000     0.000     0.000           12
0.000     0.001     0.000 perl
15580     0 sda1            0     0.000     0.000     0.000           12
0.001     0.001     0.000 perl
15354     0 sda1            0     0.000     0.000     0.000           12
0.000     0.170     0.014 sh
15584     0 sda1            0     0.000     0.000     0.000           12
0.001     0.002     0.000 perl
15548     0 sda1            0     0.000     0.000     0.000           12
0.001     0.014     0.001 perl
15577     0 sda1            0     0.000     0.000     0.000           12
0.001     0.003     0.000 perl
15519     0 sda1            0     0.000     0.000     0.000           12
0.001     0.005     0.000 perl
15578     0 sda1            0     0.000     0.000     0.000           12
0.001     0.001     0.000 perl
15583     0 sda1            0     0.000     0.000     0.000           12
0.001     0.001     0.000 perl
15547     0 sda1            0     0.000     0.000     0.000           11
0.000     0.002     0.000 perl
15576     0 sda1            0     0.000     0.000     0.000           11
0.001     0.001     0.000 perl
15518     0 sda1            0     0.000     0.000     0.000           11
0.000     0.001     0.000 perl
15354     0 sda1            0     0.000     0.000     0.000           10
0.053     0.053     0.005 lm_lid.sh
```

Die Spalten sind:

PID

Die Prozess-ID der Anwendung

UID

Die Benutzer-ID, unter welcher die Anwendungen laufen

DEV

Das Gerät auf welchem der I/O stattfand

WRITE_CNT

Die Gesamtanzahl der Schreiboperationen

WRITE_MIN

Die niedrigste Zeit, für zwei aufeinander folgende Schreibprozesse (in Sekunden)

WRITE_MAX

Die maximale Zeit, für zwei aufeinander folgende Schreibprozesse (in Sekunden)

WRITE_AVG

Die durchschnittliche Zeit, für zwei aufeinander folgende Schreibprozesse (in Sekunden)

READ_CNT

Die Gesamtanzahl für Lese-Operationen

READ_MIN

Die niedrigste Zeit, für zwei aufeinander folgende Leseprozesse (in Sekunden)

READ_MAX

Die maximale Zeit, für zwei aufeinander folgende Leseprozesse (in Sekunden)

READ_AVG

Die durchschnittliche Zeit, für zwei aufeinander folgende Leseprozesse (in Sekunden)

COMMAND

Der Name des Prozesses

In diesem Beispiel ragen drei sehr auffallende Anwendungen heraus:

```
  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
```

Diese drei Anwendungen besitzen ein **WRITE_CNT** größer als **0**, was bedeutet, dass sie eine Form von Schreibprozess während des Meßvorgangs durchgeführt haben. Von diesen war **plasma** bei weitem schlimmste Missetäter: es führte die meisten Schreiboperationen durch und natürlich war die durchschnittliche Zeit zwischen den Schreiboperationen die niedrigste. Aus diesem Grund wäre **Plasma** der beste Kandidat für Nachforschungen, wenn Sie sich Gedanken zu strom-ineffizienten Anwendungen machen würden.

Verwenden Sie die Befehle **strace** und **ltrace**, um Anwendungen näher zu untersuchen, indem Sie alle Systemaufrufe der angegebenen Prozess-ID nachverfolgen. Im aktuellen Beispiel könnten Sie beispielsweise ausführen:

```
strace -p 2789
```

In diesem Beispiel enthielt die Ausgabe von **strace** ein sich alle 45 Sekunden wiederholendes Muster, welches die KDE-Symbol Cache-Datei des Benutzers zum Schreiben öffnete, gefolgt von einem unmittelbaren Schließen der Datei. Daraus ergab sich ein notwendiges physikalisches Schreiben auf die Festplatte, da sich die Meta-Informationen der Datei (speziell der Änderungszeit) geändert hatten. Die abschließende Behebung war das Vermeiden dieser unnötigen Aufrufe, wenn keine Aktualisierungen an den Symbolen aufgetreten waren.

2.4. BATTERY LIFE TOOL KIT

Red Hat Enterprise Linux 6 führt das **Battery Life Tool Kit** (BLTK) ein, eine Test-Suite, die die Lebensdauer und die Leistung einer Batterie analysiert und simuliert. BLTK geht dabei so vor, dass eine Reihe von Aufgaben, die spezielle Gruppen und Benutzer simulieren ausgeführt wird und die Ergebnisse anschließend angezeigt werden. Auch wenn BLTK speziell für das Testen von Notebook-Leistung entwickelt wurde, kann es auch die Leistung von Desktop-Computern anzeigen, wenn es mit der Option **-a** gestartet wird.

BLTK ermöglicht es Ihnen, sehr reproduzierbare Workloads zu generieren, welche mit dem tatsächlichen Gebrauch einer Maschine vergleichbar sind. So schreibt der **office**-Workload beispielsweise einen Text, korrigiert den Inhalt und wiederholt den Vorgang für eine Tabellenkalkulation. Wenn Sie BLTK in Kombination mit **PowerTOP** oder jedem anderen Auditing- oder Analyse-Werkzeug ausführen, können Sie testen, ob die von Ihnen durchgeführten Optimierungen eine Auswirkung haben, wenn die Maschine aktiv benutzt wird, anstatt nur im Leerlauf zu sein. Da Sie den exakt selben Workload mehrere Male für verschiedene Einstellungen ausführen können, können Sie die Ergebnisse für verschiedene Einstellungen vergleichen.

Installieren Sie BLTK mit dem folgenden Befehl:

```
yum install bltk
```

Führen Sie BLTK mit dem folgenden Befehl aus:

```
bltk workload options
```

Um beispielsweise den **idle**-Workload für 120 Sekunden auszuführen:

```
bltk -I -T 120
```

Die standardmäßig verfügbaren Workloads sind:

-I, **--idle**

Das System befindet sich im Leerlauf und kann als Vergleichsbasis für anderen Workloads herangezogen werden

-R, **--reader**

simuliert das Lesen von Dokumenten (standardmäßig mit **Firefox**)

-P, **--player**

simuliert das Ansehen von multimedialen Dateien von einem CD- oder DVD-Laufwerk (standardmäßig mit **mplayer**)

-O, **--office**

simuliert das Bearbeiten von Dokumenten mit der **OpenOffice.org**-Suite

Mit Hilfe von anderen Optionen können sie Folgendes angeben:

-a, **--ac-ignore**

ignoriert, ob AC-Strom zur Verfügung steht (wird bei Desktop-Gebrauch benötigt)

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

die Dauer (in Sekunden), wie lange der Test laufen soll. Verwenden Sie diese Option mit **idle**-Workload

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

bestimmt eine Datei, die mir einem bestimmten Workload verwendet werden soll, z.B. einer Datei für den **player**-Workload, die abgespielt werden soll, anstatt auf das CD- oder DVD-Laufwerk zuzugreifen

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

definiert eine Anwendung, die mit einem bestimmten Workload verwendet werden soll, z.B. einem anderen Browser als **Firefox** für den **reader**-Workload

BLTK unterstützt eine große Anzahl an spezialisierten Optionen. Werfen Sie einen Blick auf die **bltk**-Handbuchseite für weitere Details.

BLTK speichert die Ergebnisse, die es generiert, in einem in der Konfigurationsdatei **/etc/bltk.conf** definierten Verzeichnis — standardmäßig **~/.bltk/*workload*.results.*number*/**. So beinhaltet das Verzeichnis **~/.bltk/reader.results.002/** beispielsweise das Ergebnis des dritten Tests mit dem **reader**-Workload (der erste Test wird nicht gezählt). Das Ergebnis wird über mehrere Textdateien verteilt. Um diese Ergebnisse in ein leicht zu lesendes Format zusammenzufassen, führen Sie Folgendes aus:

```
bltk_report path_to_results_directory
```

Die Ergebnisse erscheinen nun in einer Textdatei mit dem Namen **Report** im Ergebnis-Verzeichnis. Um die Ergebnisse alternativ in einem Terminal anzusehen, verwenden Sie die Option **-o**:

```
bltk_report -o path_to_results_directory
```

2.5. TUNED UND KTUNE

Tuned ist ein Daemon, der die Verwendung von Systemkomponenten überwacht und Systemeinstellungen dynamisch anhand der Überwachungsinformationen anpasst. Dynamisches Anpassen bewirkt, dass verschiedene Systemkomponenten unterschiedlich verwendet werden für die Dauer der Laufzeit für jedes vorgegebene System. So wird die Festplatte beispielsweise stark zum Zeitpunkt des Systemstarts und -Logins beansprucht. Im weiteren Verlauf, wenn der Benutzer ggf.

hauptsächlich mit Anwendungen wie OpenOffice oder E-Mail-Klienten arbeitet, wird sie jedoch kaum benutzt. Gleichermaßen werden CPU und Netzwerkgeräte unterschiedlich zu verschiedenen Zeiten verwendet. **Tuned** überwacht die Aktivität dieser Komponenten und reagiert auf Veränderungen bei Ihrer Verwendung.

Als ein praktisches Beispiel stellen Sie sich einen typischen Büro-Arbeitsrechner vor. Die meiste Zeit ist das Ethernet Netzwerkgerät sehr inaktiv. Es werden nur ein paar E-Mails ab und zu abgerufen oder versendet oder ein paar Webseiten geladen. Für diese Art von Auslastung muss das Netzwerkgerät nicht permanent mit voller Geschwindigkeit betrieben werden, wie es dies standardmäßig tut. **Tuned** besitzt ein Überwachungs- und Anpassungs-Plugin für Netzwerkgeräte, welches diese niedrige Aktivität erkennen und dann automatisch die Geschwindigkeit dieser Schnittstelle drosseln kann, was üblicherweise in geringerem Stromverbrauch resultiert. Falls die Aktivität sich über einen längeren Zeitraum deutlich erhöht, beispielsweise durch das Herunterladen eines DCD-Images oder das Öffnen einer E-Mail mit großem Anhang, erkennt **tuned** dies und setzt die Geschwindigkeit der Schnittstelle auf das Maximum, um die beste Leistung während dieses hohen Aktivitätslevels zu ermöglichen. Dieses Prinzip wird auch für andere Plugins für die CPU und die Festplatten verwendet.

Das Verhalten der Netzwerkgeräte ist standardmäßig nicht so konfiguriert, da Änderungen bei der Geschwindigkeit mehrere Sekunden dauern können, bis sie umgesetzt sind und sich somit direkt und sichtbar auf die Erfahrung des Benutzers auswirken. Ähnliche Erwägungen treffen auf die Anpassungs-Plugins für CPU und Festplatte zu. Wenn eine Festplatte die Drehzahl verringert hat, kann es einige Sekunden dauern, bis sie wieder auf volle Drehzahl beschleunigt, was in einer Verzögerung bei der Reaktionsfähigkeit des Systems während dieser Zeitspanne resultiert. Dieser Latenz-Nebeneffekt ist am geringsten für das CPU-Plugin, kann aber immer noch gemessen werden. Er ist jedoch für einen Benutzer kaum spürbar.

Neben **tuned** bieten wir jetzt auch **ktune** an. **ktune** wurde im Rahmen von Red Hat Enterprise Linux 5.3 als ein Framework und Dienst zur Optimierung der Leistung einer Maschine für spezielle Anwendungsfälle eingeführt. Seitdem hat sich **ktune** in einem solchen Maße verbessert, dass wir es jetzt als statischen Teil unseres allgemeinen Tuning-Frameworks verwenden. Es wird hauptsächlich in den verschiedenen vordefinierten Profilen, die unter Abschnitt 2.5.2, »Tuned-adm« beschrieben werden, verwendet.

Installieren Sie das Paket tuned und die damit verknüpften **systemtap**-Skripte mit dem Befehl:

```
yum install tuned
```

Bei der Installation des Pakets tuned wird ebenfalls eine Beispiel-Konfigurationsdatei unter **`/etc/tuned.conf`** eingerichtet und das Standardprofil aktiviert.

Starten Sie **tuned**, indem Sie Folgendes ausführen:

```
service tuned start
```

Um **tuned**bei jedem Systemstart zu starten, führen Sie Folgendes aus:

```
chkconfig tuned on
```

Tuned selbst besitzt zusätzliche Optionen, die Sie verwenden können, wenn Sie den Daemon manuell ausführen. Die verfügbaren Optionen sind:

`-d, --daemon`

tuned als Daemon starten, anstatt es im Vordergrund laufen zu lassen.

`-c, --conffile`

eine Konfigurationsdatei mit dem angegeben Namen und Pfad verwenden, z.B. `--conffile=/etc/tuned2.conf`. Der Standard ist `/etc/tuned.conf`.

`-D`, `--debug`

die höchste Log-Stufe verwenden.

2.5.1. Die Datei `tuned.conf`.

Die Datei `tuned.conf` beinhaltet Konfigurationseinstellungen für **tuned**. Standardmäßig befindet sie sich unter `/etc/tuned.conf`, aber Sie können einen anderen Namen und Ort angeben, indem Sie **tuned** mit der Option `--conffile` starten.

Die Konfigurationsdatei muss immer einen Abschnitt `[main]` beinhalten, der die allgemeinen Parameter für **tuned** definiert. Die Datei umfasst dann einen Abschnitt pro Plugin.

Der Abschnitt `[main]` enthält die folgenden Optionen:

`interval`

das Intervall in Sekunden, in dem **tuned** das System überwachen und abstimmen soll. Der Standardwert ist `10`.

`verbose`

gibt an, ob die Ausgabe umfangreich sein soll. Der Standardwert ist `False`.

`logging`

gibt die unterste Priorität für Meldungen an, die protokolliert werden sollen. In abfallender Reihenfolge sind die folgenden Werte gültig: `critical`, `error`, `warning`, `info` und `debug`. Der Standardwert ist `info`.

`logging_disable`

gibt die oberste Priorität für Meldungen an, die protokolliert werden sollen. Jede Meldung mit dieser Priorität oder niedriger wird nicht protokolliert. In aufsteigender Reihenfolge sind die folgenden Werte gültig: `critical`, `error`, `warning`, `info` und `debug`. Der Wert `notset` deaktiviert diese Option.

Jedes Plugin besitzt seinen eigenen Abschnitt, der mit dem Namen des Plugins in eckigen Klammern angegeben wird, z.B. `[CPUTuning]`. Jedes Plugin kann seine eigenen Optionen besitzen, Folgendes trifft jedoch auf alle Plugins zu:

`enabled`

gibt an, ob das Plugin aktiviert ist, oder nicht. Der Standardwert ist `True`.

`verbose`

gibt an, ob die Ausgabe umfangreich sein soll. Falls nicht für dieses Plugin gesetzt, wird der Wert von `[main]` übernommen.

`logging`

gibt die unterste Priorität für Meldungen an, die protokolliert werden sollen. Falls nicht für dieses Plugin gesetzt, wird der Wert von **[main]** übernommen.

Nachfolgend ist eine Beispiel-Konfigurationsdatei aufgeführt:

```
[main]
interval=10
pidfile=/var/run/tuned.pid
logging=info
logging_disable=notset

# Disk monitoring section

[DiskMonitor]
enabled=True
logging=debug

# Disk tuning section

[DiskTuning]
enabled=True
hdparm=False
alpm=False
logging=debug

# Net monitoring section

[NetMonitor]
enabled=True
logging=debug

# Net tuning section

[NetTuning]
enabled=True
logging=debug

# CPU monitoring section

[CPUMonitor]
# Enabled or disable the plugin. Default is True. Any other value
# disables it.
enabled=True

# CPU tuning section

[CPUTuning]
# Enabled or disable the plugin. Default is True. Any other value
# disables it.
enabled=True
```

2.5.2. Tuned-adm

Ein detailliertes Audit und eine Analyse eines Systems kann sehr zeitaufwendig sein und ist ggf. nicht

zu rechtfertigen, um nur ein paar zusätzliche Watt zu sparen. Bisher gab es nur die Alternative, die Vorgabewerte zu verwenden. Red Hat Enterprise Linux 6 umfasst daher getrennte Profile für spezielle Anwendungsfälle als eine Alternative zwischen diesen beiden Extremen, zusammen mit dem **tuned-adm**-Werkzeug, welches Ihnen ermöglicht, einfach zwischen diesen Profilen auf der Kommandozeile hin- und her zu wechseln. Red Hat Enterprise Linux 6 umfasst eine Reihe von vordefinierten Profilen für typische Anwendungsfälle, die sie mit dem Befehl **tuned-adm** einfach auswählen können. Sie können jedoch Profile auch selbst erstellen, modifizieren oder löschen.

Um alle derzeit aktiven Profile anzuzeigen und das derzeit aktive Profil zu identifizieren, führen Sie Folgendes aus:

```
tuned-adm list
```

Um nur das derzeit aktive Profil anzuzeigen, führen Sie Folgendes aus:

```
tuned-adm active
```

Um zu einem der verfügbaren Profile zu wechseln, führen Sie Folgendes aus:

```
tuned-adm profile profile_name
```

Zum Beispiel:

```
tuned-adm profile server-powersave
```

Um Tuning komplett zu deaktivieren:

```
tuned-adm off
```

Bei der ersten Installation von **tuned** wird das Profil **default** aktiviert. Daneben umfasst Red Hat Enterprise Linux 6 auch die folgenden vordefinierten Profile:

default

das standardmäßige Stromspar-Profil. Es hat von allen verfügbaren Profilen die geringste Auswirkung auf das Stromsparen und aktiviert lediglich CPU- und Platten-Plugins von **tuned**.

desktop-powersave

ein Stromspar-Profil, das auf Desktop-Systeme ausgerichtet ist. Aktiviert ALPM-Stromsparen für SATA-Host-Adapter (siehe Abschnitt 3.6, »Aggressive Link Power Management«), sowie die Plugins des **tuned** für CPU, Ethernet und Festplatte.

server-powersave

ein Stromspar-Profil, das auf Server-Systeme ausgerichtet ist. Aktiviert ALPM-Stromsparen für SATA-Host-Adapter, deaktiviert das Abfragen von CD-ROM via **HAL** (siehe hal-disable-polling-Handbuchseite) und aktiviert die CPU- und Festplatten-Plugins des **tuned**.

laptop-ac-powersave

ein Stromspar-Profil mit mittlerer Auswirkung, ausgerichtet auf Laptops im Strombetrieb. Aktiviert ALPM-Stromsparen für SATA-Host-Adapter, Wi-Fi-Stromsparen, sowie die CPU-, Ethernet- und Festplatten-Plugins des **tuned**.

laptop-battery-powersave

ein Stromspar-Profil mit großer Auswirkung, das auf Laptops im Batteriebetrieb ausgerichtet ist. Es aktiviert alle Stromspar-Mechanismen der vorherigen Profile und aktiviert zusätzlich noch den Mehrkern-Stromspar-Scheduler für niedrige Wakeup-Systeme und stellt sicher, dass der Ondemand-Governor, sowie AC97 Audio-Stromsparen aktiviert sind. Sie können dieses Profil verwenden, um die maximale Menge an Strom auf jeglicher Art von System zu sparen, nicht nur Laptops im Batteriebetrieb.

throughput-performance

ein Server-Profil für typisches Anpassen der Durchsatz-Leistung. Es deaktiviert die **tuned** und **ktune** Stromspar-Mechanismen, aktiviert **sysctl**-Einstellungen, die die Durchsatz-Leistung Ihrer Festplatten- und Netzwerk-I/O verbessern und wechselt zum **Deadline Scheduler**.

latency-performance

ein Server-Profil für typisches Anpassen der Latenz-Leistung. Es deaktiviert die **tuned** und **ktune** Stromspar-Mechanismen und aktiviert **sysctl**-Einstellungen, die die Latenz-Leistung ihrer Netzwerk-I/O verbessern.

Alle Profile sind in getrennten Unterverzeichnissen unter **/etc/tune-profiles** abgelegt. **/etc/tune-profiles/desktop-powersave** umfasst somit alle notwendigen Dateien und Einstellungen für dieses Profil. Jedes dieser Verzeichnisse enthält bis zu vier Dateien:

tuned.conf

die Konfiguration des Tuned-Dienstes, die für dieses Profil aktiv sein soll.

sysctl.ktune

die von **ktune** verwendeten **sysctl**-Einstellungen. Das Format ist identisch mit der Datei **/etc/sysconfig/sysctl** (siehe sysctl und sysctl.conf Handbuchseiten).

ktune.sysconfig

die Konfigurationsdatei von ktune selber, üblicherweise **/etc/sysconfig/ktune**.

ktune.sh

ein Shell-Skript im Stil von **init**, welches vom **ktune**-Dienst verwendet wird und mit Hilfe dessen spezielle Befehle während des Systemstarts zum Anpassen des Systems ausgeführt werden können.

Der einfachste Weg, ein neues Profil zu erstellen, ist das Kopieren eines vorhandenen Profils. Das Profil **laptop-battery-powersave** beinhaltet bereits ein umfangreiches Set an Anpassungen und ist somit ein nützlicher Ausgangspunkt. Kopieren Sie einfach das gesamte Verzeichnis zu dem neuen Profilnamen, wie folgt:

```
cp -a /etc/tune-profiles/laptop-battery-powersave/ /etc/tune-
profiles/myprofile
```

Passen Sie jede beliebige Datei des neuen Profils so an, dass es Ihren persönlichen Anforderungen entspricht. Wenn Sie beispielsweise das Ermitteln von CD-Änderungen benötigen, können Sie die dazugehörige Optimierung deaktivieren, indem Sie die entsprechende Zeile im ktune.sh-Skript auskommentieren:

```
# Disable HAL polling of CDROMS
# for i in /dev/scd*; do hal-disable-polling --device $i; done > /dev/null
2>&1
```

2.6. DEVICEKIT-POWER UND DEVKIT-POWER

Unter Red Hat Enterprise Linux 6 übernimmt **DeviceKit-power** die Energieverwaltungsfunktionen, die Teil von **HAL** waren, sowie einige der Energieverwaltungsfunktionen, die Teil des **GNOME Power Manager**s in vorherigen Veröffentlichungen von Red Hat Enterprise Linux waren (siehe auch Abschnitt 2.7, »GNOME Power Manager«. **DeviceKit-power** liefert einen Daemon, eine API und eine Reihe von Kommandozeilen-Werkzeuge. Jede Stromquelle auf dem System wird als Gerät dargestellt, unabhängig davon, ob es sich um ein physikalisches Gerät handelt, oder nicht. So werden beispielsweise eine Laptop-Batterie und eine Stromquelle beide als Geräte dargestellt.

Sie können mit dem Befehl **`devkit-power`** und den folgenden Optionen auf die Kommandozeilen-Werkzeuge zugreifen:

`--enumerate`, `-e`

zeigt einen Objektpfad für jedes Energiegerät auf dem System an, z.B.:

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

`--dump`, `-d`

zeigt die Parameter für alle Energiegeräte auf dem System an.

`--wakeups`, `-w`

zeigt die CPU-Wakeups auf dem System an.

`--monitor`, `-m`

überwacht das System auf Änderungen an den Energiegeräten, z.B. dem Anschließen/Entfernen einer Stromquelle oder dem Entleeren einer Batterie. Drücken Sie **`Strg+C`**, um die Überwachung des Systems zu beenden.

`--monitor-detail`

überwacht das System auf Änderungen an den Energiegeräten, z.B. dem Anschließen/Entfernen einer Stromquelle oder dem Entleeren einer Batterie. Die Option **`--monitor-detail`** liefert mehr Details, als die Option **`--monitor`**. Drücken Sie **`Strg+C`**, um die Überwachung des Systems zu beenden.

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

zeigt alle Informationen an, die für einen speziellen Objektpfad zur Verfügung stehen. Um beispielsweise Informationen über eine Batterie auf Ihrem System abzurufen, welche durch den Pfad **`/org/freedesktop/UPower/DeviceKit/power/battery_BAT0`** dargestellt wird, führen Sie Folgendes aus:

```
devkit-power -i /org/freedesktop/UPower/DeviceKit/power/battery_BAT0
```

2.7. GNOME POWER MANAGER

GNOME Power Manager ist ein Daemon, der als Teil des GNOME-Desktops installiert wird. Die meisten der Energieverwaltungs-Funktionalitäten, die **GNOME Power Manager** im Rahmen früherer Versionen von Red Hat Enterprise Linux bereitstellte, sind nun Teil von **DeviceKit-power** in Red Hat Enterprise Linux 6 (siehe Abschnitt 2.6, »DeviceKit-power und devkit-power«. **GNOME Power Manager** bleibt jedoch weiterhin das Frontend für diese Funktionalitäten. Via Applet im Infobereich informiert Sie **GNOME Power Manager** über Änderungen hinsichtlich des Energiezustands Ihres Systems, beispielsweise ein Wechsel von Batterie- zu Strombetrieb.

GNOME Power Manager ermöglicht Ihnen weiterhin die Konfiguration einiger Grundeinstellungen hinsichtlich der Energieverwaltung. Um auf diese Einstellungen zuzugreifen, klicken Sie auf das Symbol **GNOME Power Manager** im Infobereich, und anschließend auf **Einstellungen**

Der Bildschirm **Einstellungen der Energieverwaltung** ist in drei Tabulatoren unterteilt:

- **Im Netzbetrieb**
- **Im Batteriebetrieb**
- **Allgemein**

Verwenden Sie die Tabulatoren **Im Netzbetrieb** und **Im Batteriebetrieb** um anzugeben, wie viel Zeit verstreichen muss, bis die Anzeige auf einem inaktiven System abgeschaltet wird, wie viel Zeit verstreichen muss, bis ein inaktives System in den Ruhezustand versetzt wird und ob das System Festplatten herunterfahren soll, wenn diese nicht benutzt werden. Der Tabulator **Im Batteriebetrieb** gestattet es Ihnen weiterhin, die Anzeigehelligkeit einzustellen, sowie das Verhalten für ein System mit fast leerer Batterie zu definieren. Standardmäßig versetzt **GNOME Power Manager** beispielsweise ein System in den Ruhezustand, wenn das Batterielevel einen kritischen niedrigen Wert erreicht. Verwenden Sie den Tabulator **Allgemein**, um das Verhalten für den (physikalischen) Ein-/Ausschalter und die Bereitschaftstaste auf Ihrem System einzustellen, sowie die Umstände zu definieren, unter welchen das Symbol des **GNOME Power Manager** im Infobereich erscheinen soll.

2.8. ANDERE AUDITING-MITTEL

Red Hat Enterprise Linux 6 bietet noch mehr Werkzeuge, mit denen System-Auditing und -Analyse durchgeführt werden können. Die meisten können als zusätzliche Informationsquellen verwendet werden, falls Sie überprüfen möchten, was Sie bereits entdeckt haben, oder falls Sie weitere eingehende Informationen zu bestimmten Bereichen benötigen. Viele dieser Werkzeuge werden auch zur Leistungsoptimierung verwendet. Sie umfassen:

vmstat

vmstat liefert Ihnen detaillierte Informationen zu Prozessen, Speicher, Paging, Block-I/O, Traps und CPU-Aktivität. Verwenden Sie es, um einen genaueren Überblick zu bekommen, was das System insgesamt tut und wo es beschäftigt ist.

iostat

iostat ist ähnlich wie **vmstat**, allerdings nur für I/O auf Blockgeräten. Es liefert außerdem eine detailliertere Ausgabe und Statistiken.

blktrace

blktrace ist ein sehr detailliertes Block-I/O-Trace-Programm. Es unterteilt Informationen in einzelne Blöcke, die mit Anwendungen verknüpft sind. Es ist sehr nützlich in Kombination mit **diskdevstat**.

KAPITEL 3. ZENTRALE INFRASTRUKTUR UND MECHANISMEN

3.1. CPU-LEERLAUF-ZUSTÄNDE

CPUs mit der x86-Architektur unterstützen verschiedene Zustände, in denen Teile der CPU deaktiviert sind oder mit eingeschränkten Performanz-Einstellungen laufen. Diese Zustände, bekannt als *C-states*, ermöglichen es Systemen, Strom zu sparen, indem sie teilweise nicht benutzte CPUs deaktivieren. C-States werden am C0 aufwärts nummeriert, wobei höhere Nummern eine geringere CPU-Funktionalität und größeres Stromsparen repräsentieren. C-Zustände einer vorgegeben Anzahl sind überwiegend gleich auf verteilten Prozessoren, auch wenn die genauen Details des speziellen Feature-Sets des Zustands zwischen den Prozessorfamilien variieren kann. C-Zustände 0-3 werden wie folgt definiert:

C0

der Betriebs- oder Lauf-Zustand. In diesem Zustand arbeitet die CPU und befindet sich überhaupt nicht im Leerzustand.

C1, Halt

ein Zustand, in dem der Prozessor keinerlei Anweisungen ausführt, sich jedoch in keinem Niedrigstrom-Zustand befindet. Die CPU kann mit der Verarbeitung von Prozessen fast ohne Verzögerung fortfahren. Alle Prozessoren, die C-Zustände bieten, müssen diesen Zustand unterstützen. Pentium 4 Prozessoren unterstützen einen verbesserten C1-Zustand, genannt C1E, der tatsächlich ein Zustand für niedrigeren Stromverbrauch ist.

C2, Stop-Clock

ein Zustand, bei dem die Taktrate für diesen Prozessor eingefroren wird, der komplette Zustand für dessen Register und Cache jedoch behalten wird, so dass beim erneuten Start der Taktrate, die Verarbeitung von Prozessen fortfahren kann. Dies ist ein optionaler Zustand.

C3, Sleep

ein Zustand, in dem der Prozessor tatsächlich in einen Schlafzustand versetzt wird und der Cache nicht beibehalten werden muss. Das Aufwachen aus diesem Zustand dauert aus diesem Grund deutlich länger, als aus C2. Auch dies ist ein optionaler Zustand.

Aktuelle Intel CPUs mit der "Nehalem" Mikroarchitektur unterstützen einen neuen C-State, C6, der die Volt-Versorgung einer CPU auf Null reduzieren kann. Typischerweise wird der Stromverbrauch jedoch um 80% - 90% gesenkt. Der Kernel in Red Hat Enterprise Linux 6 beinhaltet eine Optimierung für diesen neuen C-State.

3.2. CPUFREQ-GOVERNORS VERWENDEN

Einer der effektivsten Wege, den Stromverbrauch und die Wärmeabgabe auf Ihrem System zu reduzieren, ist die Verwendung von CPUfreq. Mit Hilfe von CPUfreq — auch als Abgleichen der CPU-Geschwindigkeit bezeichnet — kann die Taktrate des Prozessors im laufenden Betrieb angepasst werden. Auf diese Weise kann das System mit einer reduzierten Taktrate laufen, um Strom zu sparen. Die Regeln zur Änderung von Frequenzen, sei es eine schnellere oder langsamere Taktrate, sowie dem Zeitpunkt der Änderung werden durch den CPUfreq-Governor definiert.

Der Governor definiert die Strom-Charakteristiken der System-CPU, was im Gegenzug Einfluss auf die CPU-Performanz hat. Jeder Governor besitzt eigene, einzigartige Merkmale hinsichtlich Verhalten, Zweck und Tauglichkeit in Bezug auf die Workload. Dieser Abschnitt beschreibt, wie ein CPUfreq-

Governor ausgewählt und konfiguriert werden kann, die Charakteristiken eines jeden Governors und für welche Art von Workload jeder Governor geeignet ist.

3.2.1. CPUfreq Regler-Typen

Dieser Abschnitt listet die verschiedenen Typen von CPUfreq-Reglern auf, die unter Red Hat Enterprise Linux 6 zur Verfügung stehen und beschreibt diese.

cpufreq_performance

Der Performanz-Regler zwingt die CPU, die höchstmögliche Taktfrequenz zu verwenden. Diese Frequenz wird statisch festgesetzt und ändert sich nicht. Aus diesem Grund bietet dieser Regler *keine Stromsparvorteile*. Es ist nur für Zeitspannen mit hoher Workload geeignet und auch nur dann, wenn sich die CPU kaum (oder nie) im Leerlauf befindet.

cpufreq_powersave

Im Gegensatz dazu zwingt der Powersave-Regler die CPU, die geringstmögliche Taktfrequenz zu verwenden. Aus diesem Grund bietet dieser spezielle Regler maximale Stromsparvorteile, allerdings auf Kosten der *geringsten CPU-Performanz*.

Der Begriff "powersave" kann manchmal jedoch irreführend sein, da (prinzipiell) eine langsame CPU mit voller Auslastung mehr Strom verbraucht, als eine schnelle CPU, die nicht ausgelastet ist. Aus diesem Grund kann jegliche unerwartete hohe Auslastung dazu führen, dass das System tatsächlich mehr Strom verbraucht, auch wenn es ratsam erscheint, den Stromspar-Regler in Zeiten von erwarteten Zeitspannen mit geringer Aktivität zu setzen.

Der Stromspar-Regler ist, einfach ausgedrückt, mehr ein "speed limiter" für die CPU, als ein "power saver". Er ist in Systemen und Umgebungen, in denen Überhitzung ein Problem sein kann, am nützlichsten.

cpufreq_ondemand

Der Ondemand-Regler ist ein dynamischer Regler, der es der CPU ermöglicht, maximale Taktfrequenz zu erreichen, wenn die Systemauslastung hoch ist, sowie die minimale Taktfrequenz, wenn sich das System im Leerlauf befindet. Während dies dem System ermöglicht, den Stromverbrauch entsprechend in Bezug auf die System-Auslastung anzupassen, geschieht dies zu Lasten der *Latenz zwischen dem Hin- und Herschalten von Frequenzen*. Aus diesem Grund kann Latenz jeglichen durch den Ondemand-Regler offerierten Nutzen bei der Performanz bzw. dem Stromsparen außer Kraft setzen, wenn das System zu oft zwischen Leerlauf und großen Workloads hin- und herwechselt.

Für die meisten Systeme kann der Ondemand-Regler den besten Kompromiss zwischen Wärmeabgabe, Stromverbrauch, Performanz und Handhabbarkeit bereitstellen. Wenn das System nur an bestimmten Zeiten des Tages beschäftigt ist, schaltet der Ondemand-Regler abhängig von der Auslastung automatisch zwischen maximaler und minimaler Frequenz ohne weiteren Eingriff hin und her.

cpufreq_userspace

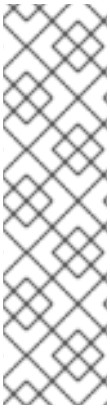
Der Userspace-Regler ermöglicht es Userspace-Programmen (oder jeglichen Prozessen, die als Root ausgeführt werden), die Frequenz zu bestimmen. Dieser Regler wird normalerweise in Zusammenhang mit dem **cpuspeed**-Daemon verwendet. Von allen Reglern ist der Userspace-Regler derjenige, der am meisten angepasst werden kann. Abhängig davon, wie er konfiguriert ist, kann er die beste Balance zwischen Performanz und Verbrauch für Ihr System bieten.

cpufreq_conservative

Wie der Ondemand-Regler passt der Conservative-Regler auch die Taktfrequenz, abhängig vom Gebrauch (wie der Ondemand-Regler). Während der Ondemand-Regler dies jedoch in einer aggressiveren Art und Weise tut (d.h. vom Maximalwert zum Minimalwert und zurück), wechselt der

Conservative-Regler mehr schrittweise zwischen Frequenzen hin- und her.

Dies bedeutet, dass der Conservative-Regler in eine Taktfrequenz wechselt, die er für geeignet für die Auslastung hält, anstatt einfach zwischen Maximalwert und Minimalwert zu wählen. Auch wenn dies möglicherweise erhebliche Einsparungen beim Stromverbrauch liefern kann, tut es dies mit einer noch *größeren Latenz*, als der Ondemand-Regler.



ANMERKUNG

Sie können einen Regler unter Verwendung von **cron**-Jobs aktivieren. Dies erlaubt es Ihnen, spezielle Regler automatisch während speziellen Tageszeiten zu bestimmen. Daher können Sie einen Regler mit niedriger Frequenz während Leerlaufzeiten (z.B. nach Büroschluß) definieren und zu einem Regler mit höherer Frequenz während Zeiten mit hohem Workload wechseln.

Werfen Sie einen Blick auf Prozedur 3.2, »Aktivierung eines CPUfreq-Governors« in Abschnitt 3.2.2, »CPUfreq-Einrichtung« für Anleitungen, wie ein spezieller Regler aktiviert werden kann.

3.2.2. CPUfreq-Einrichtung

Vor der Auswahl und Konfiguration eines CPUfreq-Governors müssen Sie zunächst einen entsprechenden CPUfreq-Treiber hinzufügen.

Prozedur 3.1. So wird ein CPUfreq-Treiber hinzugefügt

1. Verwenden Sie den folgenden Befehl, um anzuzeigen, welche CPUfreq-Treiber für Ihr System zur Verfügung stehen:

```
ls /lib/modules/[kernel version]/kernel/arch/[architecture]/kernel/cpu/cpufreq/
```

2. Verwenden Sie den Befehl **modprobe**, um den entsprechenden CPUfreq-Treiber hinzuzufügen.

```
modprobe [CPUfreq driver]
```

Stellen Sie sicher, dass Sie den **.ko** Datei-Suffix entfernen, wenn Sie den oben aufgeführten Befehl verwenden.

WICHTIG

Wählen Sie bei der Auswahl eines entsprechenden CPUfreq-Treibers immer **acpi-cpufreq** vor **p4-clockmod**. Auch wenn die Verwendung des **p4-clockmod**-Treibers die Taktfrequenz einer CPU verringert, reduziert es nicht die Spannung. Im Gegensatz dazu verringert **acpi-cpufreq** die Spannung zusammen mit der CPU-Taktfrequenz. Dies ermöglicht einen geringeren Stromverbrauch und Wärmeabgabe für jedes Teil, was allerdings zu Lasten der Performanz geht.

3. Sobald der CPUfreq-Treiber eingerichtet ist, können Sie sich ansehen, welchen Governor das System derzeit verwendet:

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

Mit Hilfe des folgenden Befehls können Sie weiterhin betrachte, welche Governors für den Gebrauch für eine bestimmte CPU zur Verfügung stehen:

```
cat /sys/devices/system/cpu/[cpu ID]/cpufreq/scaling_available_governors
```

Einige CPUfreq-Governors stehen ggf. nicht für den Gebrauch zur Verfügung. Verwenden Sie in diesem Fall den Befehl **modprobe**, um die notwendigen Kernel-Module hinzuzufügen, die den spezifischen CPUfreq-Governor, den Sie verwenden möchten, aktivieren. Diese Kernel-Module stehen unter **/lib/modules/*[kernel version]*/kernel/drivers/cpufreq/** zur Verfügung.

Prozedur 3.2. Aktivierung eines CPUfreq-Governors

1. Verwenden Sie den Befehl **modprobe**, um den Governor, den Sie verwenden möchten, zu aktivieren. Falls beispielsweise der **ondemand**-Governor nicht für Ihre CPU zur Verfügung steht, verwenden Sie den folgenden Befehl:

   ```
   modprobe cpufreq_ondemand
   ```

2. Sobald ein Governor für Ihre CPU als verfügbar aufgelistet ist, können Sie ihn mit dem folgenden Befehl aktivieren:

   ```
   echo [governor] >
   /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
   ```

3.2.3. CPUfreq-Richtlinie und Geschwindigkeit abstimmen

Sobald Sie einen passenden CPUfreq-Governor ausgewählt haben, können Sie zusätzlich die Geschwindigkeit von jeder CPU abstimmen, indem Sie die tunables unter **/sys/devices/system/cpu/*[cpu ID]*/cpufreq/** verwenden. Diese tunables sind:

- **cpuinfo_min_freq** – Zeigt die verfügbare minimale Betriebsfrequenz der CPU (in KHz) an.
- **cpuinfo_max_freq** – Zeigt die verfügbare maximale Betriebsfrequenz der CPU (in KHz) an.
- **scaling_driver** – Zeigt an, welcher CPUfreq-Treiber zur Einstellung der Frequenz auf dieser CPU verwendet wird.
- **scaling_available_governors** – Zeigt die im Rahmen dieses Kernels verfügbaren CPUfreq-Governors an. Falls Sie einen CPUfreq-Governors verwenden möchten, der nicht in dieser Datei aufgeführt ist, werfen Sie einen Blick auf Prozedur 3.2, »Aktivierung eines CPUfreq-Governors« in Abschnitt 3.2.2, »CPUfreq-Einrichtung« für Anweisungen, wie Sie dies tun können.
- **scaling_governor** – Zeigt an, welcher CPUfreq-Governor derzeit verwendet wird. Um einen anderen Governor zu verwenden, führen Sie einfach **echo *[governor]* > /sys/devices/system/cpu/*[cpu ID]*/cpufreq/scaling_governor** aus (siehe Prozedur 3.2, »Aktivierung eines CPUfreq-Governors« in Abschnitt 3.2.2, »CPUfreq-Einrichtung« für weitere Informationen).
- **cpuinfo_cur_freq** – Zeigt die aktuelle Geschwindigkeit der CPU (in KHz) an.

- **`scaling_available_frequencies`** — Listet verfügbare Frequenzen für die CPU (in KHz) auf.
- **`scaling_min_freq`** und **`scaling_max_freq`** — Setzt die *policy limits* der CPU (in KHz).

WICHTIG

Bei der Definition der Richtlinien-Limits sollten sie **`scaling_max_freq`** vor **`scaling_min_freq`** setzen.

- **`affected_cpus`** — Listet CPUs auf, die Software zur Frequenzkoordination benötigen.
- **`scaling_setspeed`** — Wird verwendet, um die Taktrate der CPU zu ändern (in KHz). Sie können nur eine Rate im Rahmen der Richtlinien-Limits der CPU (gemäß **`scaling_min_freq`** und **`scaling_max_freq`**) setzen.

Um den aktuellen Wert von jedem "tunable" anzusehen, verwenden Sie **`cat [tunable]`**. Um beispielsweise die aktuelle Geschwindigkeit der cpu0 (in KHz) anzusehen:

`cat /sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_cur_freq`.

Um den Wert eines beliebigen tunable-Wertes zu ändern, verwenden Sie **`echo [value] > /sys/devices/system/cpu/[cpu ID]/cpufreq/[tunable]`**. Um beispielsweise die Minimal-Taktrate der cpu0 auf 360 KHz zu setzen, verwenden Sie:

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

3.3. SUSPEND (RUHEZUSTAND) UND RESUME

Wenn ein System in den Ruhezustand versetzt wird, ruft der Kernel Treiber dazu auf, deren Zustände zu speichern und entlädt sie anschließend. Wenn das System wieder aus dem Ruhezustand erweckt wird, lädt es diese Treiber erneut, welche anschließend versuchen, ihre Geräte erneut zu programmieren. Die Fähigkeit der Treiber, diese Aufgabe umzusetzen bestimmt, ob ein System erfolgreich aus dem Ruhezustand erweckt werden kann.

Grafiktreiber sind in diesem Zusammenhang besonders problematisch, weil die *Advanced Configuration and Power Interface* (ACPI) Spezifikation keine Anforderungen an die System-Firmware stellt, Grafik-Hardware neu programmieren zu können. Wenn Grafiktreiber daher nicht in der Lage sind, Hardware aus einem komplett nicht initialisierten Zustand heraus zu programmieren, können Sie verhindern, dass das System aus dem Ruhezustand erweckt wird.

Red Hat Enterprise Linux 6 beinhaltet eine größere Unterstützung für neue Grafik-Chipsätze, so dass gewährleistet wird, dass Suspend und Resume auf einer größeren Anzahl von Plattformen funktioniert. Speziell die Unterstützung für NVIDIA-Chipsätze wurde enorm verbessert, speziell für die GeForce 8800 Serie.

3.4. TICKLESS-KERNEL

Bisher unterbrach der Linux-Kernel jede CPU auf einem System periodisch zu einer vordefinierten Frequenz — 100 Hz, 250 Hz oder 1000 Hz, abhängig von der Plattform. Der Kernel erkundigt sich bei der CPU nach den Prozessen, die ausgeführt wurden und verwendete die Ergebnisse für das Verwalten von Prozessen und Lastverteilung. Bekannt als *timer tick* führte der Kernel diesen Interrupt unabhängig vom Strom-Zustand der CPU durch. Daher reagierte auch eine CPU im Leerlauf bis zu

1000 Mal pro Sekunde auf diese Anfragen. Auf Systemen mit implementierten Stromsparmaßnahmen für CPUs im Leerlauf hinderte der "timer tick" die CPU daran, lange genug im Leerlaufbetrieb zu verbleiben, um von diesen Stromsparmaßnahmen zu profitieren.

Der Kernel unter Red Hat Enterprise Linux 6 läuft im *tickless*-Betrieb: d.h., er ersetzt die alten, periodischen Timer-Interrupts mit On-Demand-Interrupts. CPUs im Leerlauf können daher im Leerlaufbetrieb verbleiben, bis sich ein neues zu verarbeitendes Task in der Warteschleife befindet und CPUs, die in einen niedrigeren Strom-Zustand gewechselt sind, können länger in diesem Zustand verbleiben.

3.5. ACTIVE-STATE POWER MANAGEMENT

Active-State Power Management (ASPM) spart Strom auf dem *Peripheral Component Interconnect Express* (PCI Express oder PCIe) Subsystem, indem ein niedrigerer Strom-Zustand für PCIe-Verknüpfungen gesetzt wird, wenn die Geräte, mit denen die Verbindung hergestellt wird, nicht benutzt werden. ASPM kontrolliert den Strom-Zustand auf beiden Seiten der Verknüpfung und spart Strom im Rahmen der Verknüpfung, auch wenn sich das Gerät am Ende der Verknüpfung in einem Vollbetrieb-Zustand befindet.

Wenn ASPM aktiviert ist, erhöht sich die Geräte-Latenz aufgrund der Zeit, die benötigt wird, um die Verknüpfung in die verschiedenen Strom-Zustände zu versetzen. ASPM besitzt drei Richtlinien zur Ermittlung des Strom-Zustandes:

default

setzt Strom-Zustände der PCIe-Verknüpfung auf den von der Firmware des Systems (z.B. BIOS) definierten Standard. Dies ist der standardmäßige Zustand für ASPM.

powersave

stellt ASPM so ein, dass Strom gespart wird, wann immer möglich und unabhängig von Einbußen bei der Performanz.

performance

deaktiviert ASPM, damit es PCIe-Verknüpfungen möglich ist, mit der maximalen Performanz zu operieren.

ASPM-Richtlinien werden in **`/sys/module/pcie_aspm/parameters/policy`** gesetzt, können aber auch zum Zeitpunkt des Systemstarts mit dem ***`pcie_aspm`*** Kernel-Parameter angegeben werden, wobei **`pcie_aspm=off`** ASPM deaktiviert und **`pcie_aspm=force`** ASPM aktiviert, sogar auf Geräten, die kein ASPM unterstützen.

WARNUNG

Wenn **`pcie_aspm=force`** gesetzt wird, kann Hardware, die kein ASPM unterstützt, dazu führen, dass das System nicht mehr reagiert. Stellen Sie sicher, dass alle PCIe-Hardware auf dem System ASPM unterstützt, bevor Sie **`pcie_aspm=force`** setzen.

3.6. AGGRESSIVE LINK POWER MANAGEMENT

Aggressive Link Power Management (ALPM) ist eine Methode zum Stromsparen, mit Hilfe derer die Platte Strom sparen kann, indem eine SATA-Verknüpfung von der Platte zu einer stromsparenden Einstellung während der Leerlaufzeit (d.h. wenn keine I/O stattfindet). ALPM setzt die SATA-Verknüpfung automatisch zurück auf einen aktiven Strom-Status, sobald sich I/O-Anfragen für diese Verknüpfung ansammeln.

Die von ALPM vorgestellte Stromsparoption geht zu Lasten der Platten-Latenz. Aus diesem Grund sollten Sie ALPM nur verwenden, falls Sie erwarten, dass das System längere Abschnitte ohne I/O aufweist.

ALPM steht nur auf SATA-Controllern, die das *Advanced Host Controller Interface* (AHCI) verwenden, zur Verfügung. Weitere Informationen zu AHCI finden Sie unter http://www.intel.com/technology/serialata/ahci.htm.

Falls verfügbar wird ALPM standardmäßig aktiviert. ALPM besitzt drei Modi:

min_power

Dieser Modus setzt die Verknüpfung auf den untersten Strom-Status (SLUMBER), wenn keine I/O auf der Platte vorhanden sind. Dieser Modus ist dann geeignet, wenn längere Leerlaufzeiten erwartet werden.

medium_power

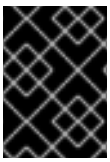
Dieser Modus setzt die Verknüpfung auf den zweitniedrigsten Strom-Status (PARTIAL), wenn keine I/O auf der Platte stattfindet. Dieser Modus wurde konzipiert, um Übergänge bei den Verknüpfungen von Strom-Status (z.B. in Zeiten von sporadischen, größeren I/O und I/O im Leerlauf) mit geringstmöglichen Auswirkungen auf die Leistung zu ermöglichen.

Der Modus **`medium_power`** ermöglicht die Verknüpfung von Übergängen zwischen PARTIAL und "ACTIVE" (ohne Stromeinsparung) Zuständen, abhängig von der Auslastung. Beachten Sie, dass es nicht immer möglich ist, eine Verknüpfung direkt von PARTIAL zu SLUMBER und umgekehrt zu erstellen. In diesem Fall kann der jeweilige Strom-Status nicht in den anderen übergehen, ohne zunächst in den ACTIVE-Status zu wechseln.

max_performance

ALPM ist deaktiviert. Die Verknüpfung geht in keinen Niedrigstrom-Zustand über, wenn es keine I/O auf der Platte gibt.

Um herauszufinden, ob Ihre SATA-Host-Adapter tatsächlich ALPM unterstützen, können Sie überprüfen, ob die Datei **`/sys/class/scsi_host/host*/link_power_management_policy`** existiert. Schreiben Sie einfach die in diesem Abschnitt beschriebenen Werte in diese Dateien, um die Einstellungen zu ändern oder betrachten Sie die Dateien zur Überprüfung der aktuellen Einstellungen.



WICHTIG

Das Setzen von ALPM auf **`min_power`** or **`medium_power`** deaktiviert das "Hot Plug" Feature automatisch.

3.7. RELATIME DRIVE ACCESS OPTIMIZATION

Der POSIX-Standard legt fest, dass Betriebssysteme Metadaten von Dateisystemen speichern, welche festhalten, wann auf jede Datei zuletzt zugegriffen wurde. Dieser Zeitstempel wird als **`atime`** bezeichnet und seine Wartung erfordert eine beständige Serie von Schreiboperationen auf das

Speichergerät. Diese Schreiboperationen beschäftigen Speichergeräte und ihre Verknüpfungen permanent und sorgen dafür, dass Sie angeschaltet bleiben. Da nur wenige Anwendungen von den **atime**-Daten tatsächlich Gebrauch machen, verschwendet diese Aktivität auf Speichergeräten Strom. Bezeichnenderweise würde das Schreiben auf das Speichergerät auch dann stattfinden, wenn die Datei nicht vom Speichergerät, sondern aus dem Cache gelesen würde. Der Linux-Kernel unterstützt seit einiger Zeit eine **noatime**-Option für **mount** und schreibt daher keine **atime**-Daten auf Dateisysteme, die mit dieser Option eingehängt sind. Das simple Deaktivieren dieses Features ist jedoch problematisch, da einige Anwendungen auf **atime**-Daten angewiesen sind und fehlschlagen, wenn diese nicht zur Verfügung stehen.

Der unter Red Hat Enterprise Linux 6 verwendete Kernel unterstützt eine weitere Alternative — **relatime**. **Relatime** speichert die **atime**-Daten, allerdings nicht für jedes Mal, bei dem auf eine Datei zugegriffen wird. Wird diese Option aktiviert, werden **atime**-Daten nur dann auf die Platte geschrieben, wenn die Datei seit der letzten Aktualisierung der **atime**-Daten modifiziert wurde (**mtime**), oder wenn der letzte Zugriff auf die Datei eine bestimmte Länge überschreitet (standardmäßig ein Tag).

Standardmäßig werden alle Dateisysteme jetzt mit **relatime** aktiviert eingehängt. Um dieses Feature auf einem gesamten System zu unterdrücken, verwenden Sie den Boot-Parameter ***default_relatime=0***. Falls **relatime** standardmäßig auf einem System aktiviert ist, können Sie es für jedes beliebige Dateisystem unterdrücken, indem Sie dieses Dateisystem mit der Option **norelatime** einhängen. Zu guter Letzt, um die vorgegebene Dauer, bis ein System die **atime**-Daten einer Datei aktualisiert, zu ändern, verwenden Sie den Boot-Parameter ***relatime_interval=***, welcher die Frist in Sekunden angibt. Der Standardwert ist **86400**.

3.8. POWER-CAPPING

Red Hat Enterprise Linux 6 unterstützt die "Power Capping" Feature, die in aktueller Hardware, wie HP *Dynamic Power Capping* (DPC) und Intel Node Manager (NM) Technologie zu finden sind. Mit Hilfe von "Power Capping" können Administratoren den Stromverbrauch von Servern einschränken. Weiterhin ermöglicht es Managern, Rechenzentren effektiver zu planen, da das Risiko, bestehende Stromversorgungen zu überlasten, deutlich verringert wird. Manager können mehr Server innerhalb derselben physikalischen Grundfläche platzieren und darauf vertrauen, dass wenn der Stromverbrauch eines Servers nach oben begrenzt wird, die Anforderungen an die Stromversorgung nicht den verfügbaren Strom während hoher Auslastung überschreiten .

HP Dynamic Power Capping

"Dynamic Power Capping" ist ein Feature, dass auf ausgewählten ProLiant- und BladeSystem-Servern zur Verfügung steht und das es Systemadministratoren ermöglicht, den Stromverbrauch eines Servers oder einer Gruppe von Servern nach oben zu begrenzen. Die Begrenzung ist ein definitives Limit, welches der Server nicht überschreitet, unabhängig von seiner aktuellen Workload. Die Begrenzung tritt nur dann in Kraft, wenn der Server sein Stromverbrauch-Limit erreicht. An dieser Stelle passt ein Management-Prozessor CPU P-Zustände und Taktdrosselung zur Einschränkung des Stromverbrauchs an.

"Dynamic Power Capping" modifiziert CPU-Verhalten unabhängig vom Betriebssystem. HPs *integrated Lights-Out 2* (iLO2) Firmware ermöglicht es Betriebssystemen jedoch, auf den Management-Prozessor zuzugreifen. Daher können Anwendungen im User Space den Management-Prozessor abfragen. Der unter Red Hat Enterprise Linux 6 verwendete Kernel umfasst Treiber für HP iLO- und iLO2-Firmware, die es Programmen ermöglichen, Management-Prozessoren unter **/dev/hpilo/d*Xccb*N** abzufragen. Weiterhin umfasst der Kernel eine Erweiterung der **hwmon sysfs** Schnittstelle zur Unterstützung von Power-Capping-Features, sowie einen **hwmon**-Treiber für ACPI 4.0 Stromzähler, welcher die **sysfs**-

Schnittstelle verwenden. Insgesamt ermöglichen diese Features Betriebssystemen und User-Space-Werkzeuge, den Wert auszulesen, der für die Power-Cap konfiguriert ist, sowie den aktuellen Stromverbrauch des Systems.

Werfen Sie einen Blick auf *HP Power Capping and HP Dynamic Power Capping for ProLiant Servers*, verfügbar unter http://h20000.www2.hp.com/bc/docs/support/SupportManual/c01549455/c01549455.pdf, für weitere Details zu HP Dynamic Power Capping.

Intel Node Manager

Intel Node Manager verhängt eine Power-Cap für Systeme und verwendet dabei P-Zustände und T-Zustände eines Prozessors zur Einschränkung der CPU-Performanz und somit des Stromverbrauchs. Durch das Umsetzen einer Energieverwaltungsrichtlinie können Administratoren Systeme so konfigurieren, dass diese weniger Strom während Zeiten, an denen die Systemauslastung niedrig ist (z.B. in der Nacht oder an Wochenenden), verbrauchen.

Intel Node Manager passt die CPU-Performanz unter Verwendung von *Operating System-directed configuration and Power Management* (OSPM) an via standardmäßigem *Advanced Configuration and Power Interface*. Wenn der Intel Node Manager den OSPM-Treiber über Änderungen an T-Zuständen informiert, setzt der Treiber die entsprechenden Änderungen an den Prozessor P-Zuständen um. Gleichzeitig ändert der Treiber die T-Zustände entsprechend, wenn der Intel Node Manager den OSPM-Treiber über Änderungen an P-Zuständen informiert. Diese Änderungen passieren automatisch und benötigen keine weitere Eingabe seitens des Betriebssystems. Administratoren konfigurieren und überwachen den Intel Node Manager mit der *Intel Data Center Manager* (DCM) Software.

Werfen Sie einen Blick auf *Node Manager — A Dynamic Approach To Managing Power In The Data Center*, erhältlich unter http://communities.intel.com/docs/DOC-4766, für weitere Details zum Intel Node Manager.

3.9. ERWEITERTE GRAFIK-ENERGIEVERWALTUNG

Red Hat Enterprise Linux 6 spart Strom bei Grafik- und Anzeigegeräten, indem diverse Quellen von unnötigem Verbrauch entfernt wurden.

LVDS-Reclocking

Low-voltage differential signalling (LVDS) ist ein System zur Übertragung von elektronischen Signalen über Kupferkabel. Eine bedeutende Umsetzung dieses Systems ist das Senden von Pixel-Informationen an *liquid crystal display* (LCD) Bildschirme in Notebook-Computern. Alle Bildschirme besitzen eine *Wiederholungsrate* — die Rate, in der sie neue Daten von einem Grafik-Controller erhalten und ein Bild auf dem Bildschirm erzeugen. Üblicherweise erhält der Bildschirm aktualisierte Daten 60 Mal pro Sekunde (eine Frequenz von 60 Hz). Wenn ein Bildschirm und ein Grafik-Controller via LVDS verbunden sind, verbraucht das LVDS-System Strom bei jedem Wiederholungszyklus. Im Leerlaufbetrieb kann die Wiederholungsrate vieler LCD-Bildschirme ohne spürbare Änderung auf 30 Hz reduziert werden (im Gegensatz zu *cathode ray tube* (CRT) Monitore, wo eine Reduzierung der Wiederholungsrate ein charakteristisches Flimmern verursacht). Der im von Red Hat Enterprise Linux 6 verwendeten Kernel-Treiber für Intel Grafikadapter führt dies *downclocking* automatisch durch und spart dabei ca. 0.5 W im Leerlaufbetrieb des Bildschirms ein.

"Self-refresh" des Speichers aktivieren

Synchronous Dynamic Random Access Memory (SDRAM) — wie es für Videospeicher in Grafikadaptern verwendet wird — wird tausendmal pro Sekunde neu aufgeladen, so dass die individuellen Speicherzellen die in ihnen gespeicherten Daten beibehalten können. Neben der Hauptfunktion, den Datenfluß in und aus dem Speicher heraus zu verwalten, ist der Speicher-Controller normalerweise verantwortlich für die Initiierung dieser Refresh-Zyklen. SDRAM besitzt jedoch auch einen *self-refresh*-Modus. In diesem Modus verwendet der Speicher einen internen Taktgeber, um seine eigenen

Refresh-Zyklen zu generieren. Dies gestattet dem System, den Speicher-Controller herunterzufahren, ohne die Daten, die sich gerade im Speicher befinden, zu gefährden. Der in Red Hat Enterprise Linux 6 verwendete Kernel kann Speicher "self-refresh" in Intel Grafikadaptern initiieren, wenn sich diese im Leerlauf befinden. Dies spart in etwa 0.8 W.

GPU-Taktreduzierung

Typische Graphical Processing Units (GPUs) beinhalten interne Taktgeber, die verschiedene Teile ihrer internen Schaltkreise verwalten. Der in Red Hat Enterprise Linux 6 verwendete Kernel kann die Frequenz einiger interner Taktgeber in Intel und ATI GPUs reduzieren. Die Verringerung der Anzahl der Zyklen, die GPU-Komponenten in einer vorgegebenen Zeitspanne durchlaufen, spart den Strom, den sie ansonsten in Zyklen verbraucht hätten, in denen Sie keine Leistung erbracht hätten. Der Kernel reduziert die Geschwindigkeit dieser Taktgeber automatisch, wenn sich die GPU im Leerlauf befindet und erhöht sie entsprechend, wenn die GPU-Aktivität steigt. Die Reduzierung von GPU-Taktgeber-Zyklen kann bis zu 5 W einsparen.

GPU-Powerdown

Die Intel und ATI Grafiktreiber in Red Hat Enterprise Linux 6 können ermitteln, wenn kein Monitor an einen Adapter angeschlossen ist und somit die GPU komplett deaktivieren. Dieses Feature ist besonders bedeutend für Server, an die keine Monitore regulär angeschlossen sind.

3.10. RFKILL

Viele Computersysteme besitzen Sender, inklusive Wi-Fi, Bluetooth und 3G-Geräte. Diese Geräte verbrauchen Strom, der verschwendet wird, wenn das Gerät nicht verwendet wird.

RFKill ist ein Subsystem im Linux-Kernel, das eine Schnittstelle bietet, durch die Sender in einem Computersystem abgefragt, aktiviert und deaktiviert werden können. Beim Deaktivieren können Transmitter in einen Zustand versetzt werden, in dem sie Software reaktivieren können (einem *soft block*), oder wo sie nicht via Software reaktiviert werden können, via *hard block*.

Der RFKill-Kern liefert die Schnittstelle zur Anwendungsprogrammierung (API) für das Subsystem. Kernel-Treiber, die für die Unterstützung von RFKill entworfen wurden, verwenden diese API zur Registrierung mit dem Kernel und beinhalten Maßnahmen zur Aktivierung und Deaktivierung des Geräts. Zusätzlich liefert der RFKill-Kern Benachrichtigungen, die Benutzer-Anwendungen interpretieren können, sowie Möglichkeiten, wie Benutzer-Applikationen Transmitter-Zustände abfragen können.

Die RFKill-Schnittstelle befindet sich unter **`/dev/rfkill`**, welches den aktuellen Zustand aller Sender auf dem System beinhaltet. Für jedes Gerät ist der aktuelle RFKill-Zustand in **`sysfs`** registriert. Zusätzlich liefert RFKill *uevents* für jede Zustandsänderung bei einem RFKill-aktivierten Gerät.

Rfkill ist ein Kommandozeilenwerkzeug, mit dem Sie RFKill-aktivierte Geräte auf dem System abfragen und verändern können. Installieren Sie das Paket rfkill, um das Werkzeug zu erhalten.

Verwenden Sie den Befehl **`rfkill list`**, um eine Liste von Geräten zu erhalten, welches jeweils mit einer *Index-Nummer* verknüpft ist, beginnend mit **`0`**. Sie können diese Index-Nummer verwenden, um **rfkill** darüber zu informieren, ob ein Gerät geblockt oder entblockt werden soll. Zum Beispiel:

```
rfkill block 0
```

blockiert das erste RFKill-aktivierte Gerät auf dem System.

Mit Hilfe von **rfkill** können Sie weiterhin bestimmte Kategorien von Geräten, oder alle RFKill-aktivierte Geräte blockieren. Zum Beispiel:

–

```
rfkill block wifi
```

blockiert alle Wi-Fi-Geräte auf dem System. Führen Sie folgenden Befehl aus, um alle RFKill-aktivierten Geräte zu blockieren:

```
rfkill block all
```

Führen Sie **rfkill unblock** anstatt **rfkill block** aus, um Geräte zu entblockieren. Um eine komplette Liste von Geräte-Kategorien, die **rfkill** blockieren kann, zu erhalten, führen Sie **rfkill help** aus.

3.11. OPTIMIERUNGEN IM USER SPACE

Die Reduzierung der Arbeitslast, die von der System-Hardware geleistet wird, ist grundlegend für das Sparen von Strom. Auch wenn die in Kapitel 3, *Zentrale Infrastruktur und Mechanismen* beschriebenen Änderungen es dem System ermöglichen, in verschiedenen reduzierten Stromverbrauch-Zuständen zu operieren, hindern Anwendungen im User Space, die unnötige Arbeit von der System-Hardware anfordern, daher die Hardware am Wechseln in diese Zustände. Während der Entwicklung von Red Hat Enterprise Linux 6 wurden Audits in den folgenden Bereichen durchgeführt, um unnötige Anforderungen an die Hardware zu reduzieren:

Reduzierte "wakeups"

Red Hat Enterprise Linux 6 verwendet einen *Tickless Kernel* (siehe Abschnitt 3.4, »Tickless-Kernel«), der es den CPUs ermöglicht, länger in einem tieferen Leerlauf-Zustand zu verbleiben. Allerdings ist der *Timer Tick* nicht die einzige Quelle für exzessive CPU-Wakeups und Funktionsaufrufe von Anwendungen können die CPU auch daran hindern, in Leerlauf-Zustände zu wechseln oder in diesen zu verbleiben. Unnötige Funktionsaufrufe wurden in über 50 Anwendungen reduziert.

Reduzierte Speicher- und Netzwerk-I/O

Input oder Output (IO) auf Speichergeräten und Netzwerkschnittstellen zwingen Geräte zum Stromverbrauch. In Speicher- und Netzwerkgeräten, die im Leerlauf reduzierte Stromzustände unterstützen (z.B. ALPM oder ASPM), kann dieser Datenfluss verhindern, dass das Gerät in einen Leerlauf-Zustand versetzt wird oder in diesem verbleibt. Auch kann es verhindern, dass Festplatten die Drehzahl reduzieren, wenn Sie nicht mehr verwendet werden. Exzessive und unnötige Anforderungen an Speicher wurden in mehreren Anwendungen minimiert. Speziell in solchen, die verhinderten, dass Festplatten die Drehzahl reduzieren.

Initscript-Audit

Dienste, die automatisch starten, unabhängig davon, ob sie benötigt werden, oder nicht, bergen ein großes Potential bei der Verschwendung von Systemressourcen. Dienste sollten stattdessen wann immer möglich standardmäßig "aus" oder "auf Anfrage" sein. So wurde beispielsweise bisher der **BlueZ**-Dienst, der Bluetooth-Unterstützung aktiviert, automatisch beim Systemstart aktiviert, unabhängig davon, ob Bluetooth-Hardware vorhanden war, oder nicht. Das **BlueZ** initscript überprüft nun, ob Bluetooth-Hardware auf dem System vorhanden ist, bevor es den Dienst startet.

KAPITEL 4. ANWENDUNGSFÄLLE

Dieses Kapitel beschreibt zwei Arten von Anwendungsfällen, um Methoden zur Analyse und Konfiguration zu verdeutlichen, die an anderer Stelle in diesem Handbuch beschrieben werden. Das erste Beispiel behandelt typische Server und das zweite ein typisches Laptop.

4.1. BEISPIEL – SERVER

Ein typischer Standard-Server kommt heutzutage mit grundsätzlich allen Hardware-Features, die unter Red Hat Enterprise Linux 6 unterstützt werden. Die erste Sache, die in Betracht gezogen werden sollte, ist die Art des Workloads, für die der Server hauptsächlich verwendet werden soll. Basierend auf diesen Informationen können Sie entscheiden, welche Komponenten für die Energieeinsparung optimiert werden können.

Unabhängig von der Art des Servers sind Leistungen bei der Grafik im Allgemeinen nicht erforderlich. Daher können die GPU-Stromsparoptionen aktiviert bleiben.

Webserver

Ein Webserver benötigt Netzwerk- und Platten-I/O. Abhängig von der externen Verbindungsgeschwindigkeit könnten 100 Mbit/s ausreichen. Falls die Maschine eher statische Seiten anbietet, ist die CPU-Performanz ggf. nicht so wichtig. Optionen bei der Energieverwaltung könnten daher sein:

- keine Platten- oder Netzwerk-Plugins für **tuned**.
- ALPM angeschaltet.
- **ondemand**-Governor angeschaltet.
- Netzwerkkarte auf 100 Mbit/s limitiert.

Rechen-Server

Ein Rechen-Server benötigt hauptsächlich CPU. Optionen bei der Energieverwaltung können sein:

- abhängig von den Jobs und wo die Datenspeicherung stattfindet, Platten- oder Netzwerk-Plugins für **tuned** oder für Batch-Modus-Systeme, voll aktiver **tuned**.
- abhängig von der Verwendung evtl. der **performance**-Governor.

Mailserver

Ein Mailserver benötigt überwiegend I/O und CPU. Optionen bei der Energieverwaltung können sein:

- **ondemand**-Governor angeschaltet, da die letzten paar Prozent der CPU-Performanz nicht wichtig sind.
- keine Platten- oder Netzwerk-Plugins für **tuned**.
- Netzwerkgeschwindigkeit sollte nicht beschränkt werden, da Mail oft intern verschickt wird und somit aus einem 1 Gbit/s oder einem 10 Gbit/s-Link Nutzen ziehen kann.

Datei-Server

Die Anforderungen für Datei-Server ähneln denen für Mailserver, aber abhängig von dem verwendeten Protokoll, benötigen sie ggf. mehr CPU-Performanz. Samba-basierte Server benötigen üblicherweise mehr CPU als NFS und NFS üblicherweise mehr als iSCSI. Nichtsdestrotrotz sollten Sie in der Lage sein,

den **ondemand**-Governor zu verwenden.

Verzeichnis-Server

Ein Verzeichnis-Server besitzt üblicherweise geringere Anforderungen für Platten-I/O, besonders wenn er mit genügend RAM ausgestattet ist. Netzwerk-Latenz ist wichtig, Netzwerk-I/O dagegen nicht so. Sie können erwägen, die Netzwerk-Latenz mit einer geringeren Link-Geschwindigkeit abzustimmen, sollten dies jedoch sorgfältig in Ihrem jeweiligen Netzwerk testen.

4.2. BEISPIEL – LAPTOP

Ein sehr üblicher Fall, bei dem Energieverwaltung und -sparen tatsächlich einen Unterschied machen können, sind Laptops. Da Laptops vom Design her normalerweise sowie schon deutlich weniger Energie verbrauchen, als Workstations oder Server, ist das Potential für die Gesamt-Einsparung geringer, als für andere Maschinen. Im Batteriebetrieb kann jedoch jegliche Form von Einsparung zur Verlängerung der Lebensdauer der Batterie in einem Laptop beitragen. Auch wenn sich dieser Abschnitt auf Laptops im Batteriebetrieb konzentriert, können Sie dennoch einige oder sogar alle dieser Abstimmungen auch auf den Strombetrieb anwenden.

Einsparungen für einzelne Komponenten machen üblicherweise einen größeren relativen Unterschied auf Laptops aus, als auf Workstations. Eine 1 Gbit/s Netzwerkschnittstelle, die unter 100 Mbits/s läuft, spart beispielsweise 3–4 Watt. Bei einem typischen Server mit einem Gesamt-Stromverbrauch von etwa 400 Watt entspricht diese Einsparung in etwa 1 %. Auf einem Laptop mit einem Gesamt-Stromverbrauch von etwa 40 Watt entspricht diese Einsparung in etwa 10 % des gesamten Verbrauchs.

Spezielle Optimierungen zum Einsparen von Strom auf einem typischen Laptop umfassen:

- Konfigurieren Sie das System-BIOS so, dass sämtliche nicht verwendete Hardware deaktiviert ist. Beispielsweise Parallel- oder Serielle-Ports, Kartenleser, Webcams, Wi-Fi und Bluetooth, um nur ein paar mögliche Kandidaten zu nennen.

- Dimmen Sie die Anzeige in dunkleren Umgebungen, in denen Sie keine volle Beleuchtung zum Lesen des Bildschirms benötigen. Verwenden Sie **System+Einstellungen** → **Energieverwaltung** auf dem GNOME-Desktop, **Kickoff Application Launcher+Computer+Systemeinstellungen+Erweitert** → **Energieverwaltung** auf dem KDE-Desktop; oder **gnome-power-manager** oder **xbacklight** auf der Kommandozeile; oder die Funktionstasten auf Ihrem Laptop.

- Verwenden Sie das Profil **laptop-battery-powersave** von **tuned-adm**, um eine ganze Reihe an Stromsparmechanismen einzustellen. Beachten Sie, dass die Leistung und Latenz für die Festplatte und die Netzwerkschnittstelle davon betroffen sind.

Zusätzlich (oder alternativ) können Sie viele kleine Anpassungen manuell an verschiedenen Systemeinstellungen durchführen:

- den **ondemand**-Governor verwenden (standardmäßig unter Red Hat Enterprise Linux 6 aktiviert)

- Laptop-Modus aktivieren (Teil des **laptop-battery-powersave**-Profils):

```
echo 5 > /proc/sys/vm/laptop_mode
```

- Flush-Zeit auf Platte erhöhen (Teil des **laptop-battery-powersave**-Profils):

```
echo 1500 > /proc/sys/vm/dirty_writeback_centisecs
```

- NMI-Watchdog deaktivieren (Teil des **laptop-battery-powersave**-Profils):

```
echo 0 > /proc/sys/kernel/nmi_watchdog
```

- AC97 Audio-Energiesparen aktivieren (standardmäßig unter Red Hat Enterprise Linux 6 aktiviert):

```
echo Y > /sys/module/snd_ac97_codec/parameters/power_save
```

- Multi-Core Stromsparen aktivieren (Teil des **laptop-battery-powersave**-Profils):

```
echo 1 > /sys/devices/system/cpu/sched_mc_power_savings
```

- USB auto-suspend aktivieren:

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

  Beachten Sie, dass USB auto-suspend nicht mit allen USB-Geräten korrekt funktioniert.

- minimale Stromeinstellungen für ALPM aktivieren (Teil des **laptop-battery-powersave**-Profils):

```
echo min_power >
/sys/class/scsi_host/host*/link_power_management_policy
```

- Dateisystem unter Verwendung von relatime einhängen (Standard in Red Hat Enterprise Linux 6):

```
mount -o remount,relatime mountpoint
```

- den besten Modus für das Stromsparen für Festplatten aktivieren (Teil des **laptop-battery-powersave**-Profils):

```
hdparm -B 1 -S 200 /dev/sd*
```

- CD-ROM-Abfrage deaktivieren (Teil des **laptop-battery-powersave**-Profils):

```
hal-disable-polling --device /dev/scd*
```

- Bildschirmhelligkeit auf **50** oder weniger reduzieren, z.B.:

```
xbacklight -set 50
```

- DPMS für Bildschirme im Leerlaufbetrieb aktivieren:

```
xset +dpms; xset dpms 0 0 300
```

- Wi-Fi Strom-Level reduzerien (Teil des **laptop-battery-powersave**-Profils):

```
for i in /sys/bus/pci/devices/*/power_level ; do echo 5 > $i ; done
```

- Wi-Fi deaktivieren:

```
echo 1 > /sys/bus/pci/devices/*/rf_kill
```

- Kabel-Netzwerk auf 100 Mbit/s limitieren (Teil des **`laptop-battery-powersave`**-Profils):

```
>ethtool -s eth0 advertise 0x0F
```

ANHANG A. TIPPS FÜR ENTWICKLER

Jedes gute Programmier-Lehrbuch umfasst Probleme mit Speicherzuweisung und der Performanz spezieller Funktionen. Achten Sie bei Ihrer Entwicklung von Software auf Probleme, die den Energieverbrauch auf den Systemen, auf denen die Software läuft, erhöhen könnte. Auch wenn diese Berücksichtigungen keinen Einfluß auf jede Zeile des Codes haben, können Sie diesen in Bereichen optimieren, die häufig Flaschenhälse bei der Performanz darstellen.

Einige Techniken, die häufig problematisch sind, beinhalten:

- das Verwenden von Threads.
- unnötige und ineffiziente CPU-Wake-Ups. Falls Sie den Ruhezustand beenden müssen, machen Sie alles gleichzeitig (race to idle) und so schnell wie möglich.
- unnötiges Verwenden von `[f]sync()`.
- unnötiges aktives Abfragen (polling) oder das Verwenden von kurzen, regelmäßigen Timeouts (stattdessen auf Ereignisse reagieren).
- ineffektives Verwenden von Wake-Ups.
- ineffizienter Zugriff auf Platten. Verwenden Sie große Puffer, um häufigen Zugriff auf die Platte zu vermeiden. Schreiben Sie jeweils einen großen Block.
- ineffiziente Verwendung von Timer. Gruppieren Sie Timer über Anwendungen (oder sogar Systemen) verteilt, falls möglich.
- exzessive I/O. Stromverbrauch oder Speichergebrauch (inklusive Speicherlecks)
- Durchführen unnötiger Berechnungen.

Die folgenden Abschnitte untersuchen einige dieser Bereiche in größerem Detail.

A.1. DAS VERWENDEN VON THREADS

Es gilt als weit verbreitet, dass die Verwendung von Threads die Performanz von Anwendungen verbessert und beschleunigt. Dies trifft aber nicht in jedem Fall zu.

Python

Python verwendet Global Lock Interpreter[1], so dass Threading nur für größere I/O-Operationen profitabel ist. **Unladen-swallow** [2] ist eine schnellere Python-Implementierung, mit der Sie ggf. Ihren Code optimieren können.

Perl

Perl-Threads wurden ursprünglich für Systeme ohne Forking geschaffen (wie beispielsweise Systeme mit 32-Bit Windows-Betriebssystemen). Bei Perl-Threads werden Daten für jeden einzelnen Thread kopiert (Copy On Write). Daten werden standardmäßig nicht gemeinsam genutzt, da Benutzer in der Lage sein sollten, das Level an Daten-Sharing zu bestimmen. Das Modul **threads::shared** muss für Daten-Sharing eingebunden sein. Daten werden jedoch nicht nur dann kopiert (Copy On Write), sondern das Modul erstellt auch eng verknüpfte Variablen für die Daten, was noch mehr Zeit kostet und noch langsamer ist [3].

C

C-Threads nutzen den Speicher gemeinsam, jeder Thread besitzt seinen eigenen Stapel (stack) und der Kernel muss keine neuen Dateideskriptoren erstellen und neuen Speicherplatz zuweisen. C kann wirklich die Unterstützung von mehreren CPUs für mehrere Threads nutzen. Um daher die Performanz Ihrer Threads zu maximieren, verwenden Sie eine höhere Sprache wie C oder C++. Falls Sie eine Skripting-Sprache verwenden, ziehen Sie das Schreiben eines C-Bindings in Betracht. Benutzen Sie Profilers zur Identifizierung von schlecht funktionierenden Teilen in Ihrem Code [4].

A.2. WAKE-UPS

Viele Anwendungen überprüfen Konfigurationsdateien auf Änderungen. In vielen Fällen findet die Überprüfung zu einem festgesetzten Intervall statt, z.B. jede Minute. Dies kann ein Problem darstellen, da es Platten dazu zwingt, von Spindowns in den aktiven Modus zu wechseln. Die beste Lösung ist das Ermitteln eines guten Intervalls, ein guter Mechanismus zur Überprüfung oder das Prüfen von Änderungen mit **inotify** und Reaktionen auf Ereignisse. **Inotify** kann eine Vielfalt von Änderungen an einer Datei oder einem Verzeichnis überprüfen.

Zum Beispiel:

```
int fd;
fd = inotify_init();
int wd;
/* checking modification of a file - writing into */
wd = inotify_add_watch(fd, "./myConfig", IN_MODIFY);
if (wd < 0) {
  inotify_cant_be_used();
  switching_back_to_previous_checking();
}
...
fd_set rdfs;
struct timeval tv;
int retval;
FD_ZERO(&rdfs);
FD_SET(0, &rdfs);

tv.tv_sec = 5;
value = select(1, &rdfs, NULL, NULL, &tv);
if (value == -1)
  perror(select);
else {
  do_some_stuff();
}
...
```

Der Vorteil dieser Herangehensweise ist die Vielfalt der Checks, die Sie durchführen können.

Die Haupt-Beschränkung ist, dass nur eine begrenzte Anzahl von Watches auf einem System zur Verfügung stehen. Die Anzahl kann unter **`/proc/sys/fs/inotify/max_user_watches`** abgerufen werden und obwohl sie geändert werden kann, wird dies nicht empfohlen. Darüber hinaus, falls **inotify** fehlschlägt, muss der Code auf eine andere Check-Methode zurückgreifen, was üblicherweise bedeutet, dass **`#if #define`** oft im Quellcode vorkommt.

Werfen Sie einen Blick auf die inotify-Handbuchseite für weitere Informationen zu **inotify**.

A.3. FSYNC

Fsync ist für seinen I/O-intensiven Betrieb bekannt, aber das ist nur die halbe Wahrheit. Werfen Sie beispielsweise einen Blick auf Theodore Ts'o's Artikel *Don't fear the fsync!* [5] und die dazugehörige Diskussion.

Firefox rief bisher die **sqlite**-Bibliothek bei jedem Aufruf einer neuen Seite durch den Benutzer auf. **Sqlite** rief **fsync** auf und aufgrund der Einstellungen des Dateisystems (hauptsächlich ext3 mit data-ordered Modus) entstand eine hohe Latenz bei Inaktivität. Dies konnte lange dauern (bis zu 30 Sekunden), wenn ein andere Prozess eine große Datei zur selben Zeit kopierte.

Bei anderen Fällen jedoch, bei denen **fsync** überhaupt nicht benutzt wurde, traten Probleme mit dem Wechsel zum ext4-Dateisystem auf. Ext3 war auf data-ordered Modus gesetzt, der den Speicher alle paar Sekunden löschte und auf Platte speicherte. Mit ext4 jedoch und laptop_mode, wurde das Intervall zwischen Speicherungen länger und Daten konnten verloren gehen, wenn das System unerwartet ausgeschaltet wurde. Auch wenn ext4 nun gepatcht ist, müssen wir das Design unserer Anwendungen nach wie vor vorsichtig überdenken und **fsync** verwenden, wo es angebracht ist.

Das nachfolgende, einfache Beispiel für das Lesen und das Schreiben in eine Konfigurationsdatei zeigt, wie eine Sicherung einer Datei erstellt werden kann, oder wie Daten verloren gehen können:

```
/* open and read configuration file e.g. ~/.kde/myconfig */
fd = open("./kde/myconfig", O_WRONLY|O_TRUNC|O_CREAT);
read(myconfig);
...
write(fd, bufferOfNewData, sizeof(bufferOfNewData));
close(fd);
```

Ein besserer Ansatz wäre:

```
open("/.kde/myconfig", O_WRONLY|O_TRUNC|O_CREAT);
read(myconfig);
...
fd = open("/.kde/myconfig.suffix", O_WRONLY|O_TRUNC|O_CREAT);
write(fd, bufferOfNewData, sizeof(bufferOfNewData));
fsync; /* paranoia - optional */
...
close(fd);
rename("/.kde/myconfig", "/.kde/myconfig~"); /* paranoia - optional */
rename("/.kde/myconfig.suffix", "/.kde/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

[5] http://thunk.org/tytso/blog/2009/03/15/dont-fear-the-fsync/

ANHANG B. REVISIONSVERLAUF

Version 1.0-7.400 **2013-10-31** **Rüdiger Landmann**
Rebuild with publican 4.0.0

Version 1.0-7 **2012-07-18** **Anthony Towns**
Rebuild for Publican 3.0

Version 1.0-2 **Fri Oct 22 2010** **Rüdiger Landmann**
Kleinere Fehlerkorrekturen im Text durch den Autor

Version 1.0-1 **Thu Oct 7 2010** **Rüdiger Landmann**
"Draft"-Tag wurde entfernt

Version 1.0-0 **Thu Oct 7 2010** **Rüdiger Landmann**
GA-Release