



Red Hat Enterprise Linux 7

전원 관리 가이드

RHEL 7에서 전력 소비 관리 및 최적화

Red Hat Enterprise Linux 7 전원 관리 가이드

RHEL 7에서 전력 소비 관리 및 최적화

Marie Doleželová

Red Hat Customer Content Services

mdolezel@redhat.com

Jana Heves

Red Hat Customer Content Services

Jacquelynn East

Red Hat Customer Content Services

Don Domingo

Red Hat Customer Content Services

Rüdiger Landmann

Red Hat Customer Content Services

Jack Reed

Red Hat Customer Content Services

Red Hat, Inc.

법적 공지

Copyright © 2017 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

초록

이 문서에서는 Red Hat Enterprise Linux 7 시스템에서 전력 소비를 효과적으로 관리하는 방법을 설명합니다. 다음 섹션에서는 전력 소비를 줄이는 다양한 기술과 각 기술이 시스템의 전체 성능에 미치는 영향에 대해 설명합니다.

차례

1장. 개요	3
1.1. 전원 관리의 중요성	3
1.2. 전원 관리 기본 사항	4
2장. 전원 관리 감사 및 분석	6
2.1. 감사 및 분석 개요	6
2.2. POWERTOP	6
2.3. DISKDEVSTAT 및 NETDEVSTAT	8
2.4. 배터리 수명 도구 키트	11
2.5. TUNED	12
2.6. UPOWER	14
2.7. GNOME POWER MANAGER	14
2.8. 감사용 기타 툴	14
3장. 코어 인프라 및 메커니즘	16
3.1. CPU 유휴 상태	16
3.2. CPUFREQ	16
3.3.	19
3.4.	19
3.5.	19
3.6.	19
3.7.	20
3.8.	20
3.9.	20
3.10.	20
3.11.	20
3.12. RFKILL	20
4장. 사용 사례	22
4.1.	22
4.2. 예 - 랩탑	22
부록 A. 개발자를 위한 팁	24
A.1. 스레드 사용	24
A.2. WAKE-UPS	25
A.3. FSYNC	26
부록 B. 버전 내역	27

1장. 개요

전원 관리는 Red Hat Enterprise Linux 7의 개선 사항 중 하나입니다. 컴퓨터 시스템에서 사용하는 전력을 제한하는 것은 녹색 IT (환경 친화적인 컴퓨팅)의 가장 중요한 측면 중 하나이며, 재활용 가능한 자료 사용, 하드웨어 생산의 환경 영향, 시스템 설계 및 배포에 대한 환경 인식이 포함됩니다. 이 문서에서는 Red Hat Enterprise Linux 7을 실행하는 시스템의 전원 관리에 관한 지침 및 정보를 제공합니다.

1.1. 전원 관리의 중요성

전력 관리의 핵심은 각 시스템 구성 요소의 에너지 소비를 효과적으로 최적화하는 방법에 대한 이해입니다. 이를 위해서는 시스템에서 수행하는 다양한 작업을 연구하고 각 구성 요소를 구성하여 성능에 적합한 지 확인합니다.

전원 관리의 주요 동기는 다음과 같습니다.

- 비용 절감을 위해 전체 전력 소비 감소

전원 관리 결과의 적절한 사용:

- 서버 및 컴퓨팅 센터의 열 감소
- 재충전, 공간, 케이블, 발전기 및 중단되지 않는 전원 공급 장치(UPS)를 포함한 보조 비용 감소
- 랩탑의 배터리 수명 연장
- 탄소 dioxide 출력 감소
- 그린 IT에 대한 정부 규정 또는 법적 요구 사항 충족(예: energy Star)
- 새로운 시스템에 대한 회사 지침 준수

일반적으로 특정 구성 요소(또는 전체 시스템)의 전력 소비를 낮추면 열과 자연적으로 성능이 저하됩니다. 따라서 특히 미션 크리티컬 시스템의 경우 구성에서 발생하는 성능 저하를 철저히 조사하고 테스트해야 합니다.

시스템이 수행하는 다양한 작업을 반복하고 각 구성 요소를 구성하여 성능에 충분한지 확인하고, 에너지를 저장하고, 열을 줄이고, 랩탑에 대한 배터리 수명을 최적화할 수 있습니다. 전력 소비와 관련하여 시스템의 분석 및 튜닝을 위한 많은 원칙은 성능 튜닝의 경우와 유사합니다. 시스템은 일반적으로 성능 또는 전원을 대상으로 최적화되기 때문에 전원 관리 및 성능 튜닝은 시스템 구성에 대한 반대의 접근 방식입니다. 이 설명서에서는 Red Hat에서 제공하는 툴과 이 프로세스를 지원하기 위해 개발한 기술을 설명합니다.

Red Hat Enterprise Linux 7에는 이미 기본적으로 활성화되어 있는 많은 새로운 전원 관리 기능이 포함되어 있습니다. 모두 일반적인 서버 또는 데스크탑 사용 사례의 성능에 영향을 미치지 않도록 선택적으로 선택되었습니다. 그러나 최대 처리량, 최소 대기 시간 또는 가장 높은 CPU 성능이 필요한 매우 구체적인 사용 사례의 경우 해당 기본값을 검토해야 할 수 있습니다.

이 문서에 설명된 기술을 사용하여 머신을 최적화해야 하는지 여부를 결정하려면 다음 몇 가지 질문을 합니다.

질문 최적화 해야 하나?

답변 전력 최적화의 중요성은 회사에 따라야 할 지침이 있는지 또는 준수해야 하는 규제가 있는지에 따라 달라집니다.

질문 최적화하려면 얼마나 해야 하나?

답변 당사가 제공하는 기술 중 일부는 귀하의 머신의 감사 및 분석을 자세히 수행할 필요가 없지만 일반적으로 전력 사용량을 개선하는 일반적인 최적화 집합을 제공합니다. 물론 이러한 시스템은 일반적으로 수동으로 감사 및 최적화된 시스템만큼 좋지 않지만 좋은 손상을 제공합니다.

질문 최적화로 인해 허용되지 않는 수준으로 시스템 성능이 저하될 수 있습니까?

답변 이 문서에서 설명하는 대부분의 기술은 시스템 성능에 눈에 띄게 영향을 미칩니다. Red Hat Enterprise Linux 7에서 이미 적용된 기본값 이상으로 전원 관리를 구현하려면 전원 최적화 후 시스템의 성능을 모니터링하고 성능 손실이 허용되는지 여부를 결정해야 합니다.

질문 시스템 최적화에 소비된 시간과 리소스가 이익보다 크게 향상됩니까?

답변 전체 프로세스를 수동으로 따라 단일 시스템을 최적화하는 것은 일반적으로 시간과 비용이 단일 머신의 수명 동안 얻을 수 있는 일반적인 이점보다 훨씬 높지 않습니다. 반면 10000개의 데스크탑 시스템을 모두 동일한 구성과 설정을 사용하여 사무실에 돌아오게 하면 최적화된 하나의 설정을 만들고 모든 10000 대의 시스템에 적용하는 것이 가장 좋습니다.

다음 섹션에서는 최적의 하드웨어 성능이 에너지 소비 측면에서 시스템에 어떤 영향을 미치는지 설명합니다.

1.2. 전원 관리 기본 사항

효과적인 전원 관리는 다음 원칙에 따라 구축됩니다.

유휴 CPU는 필요한 경우에만 작동해야 합니다.

Red Hat Enterprise Linux 6 이후 커널은 *틱리스*를 실행하므로 이전의 주기적인 타이머 인터럽트가 온디맨드 인터럽트로 교체되었습니다. 따라서 유휴 CPU는 처리를 위해 새 작업이 대기열에 지정될 때까지 유휴 상태로 유지될 수 있으며 더 낮은 전원 상태를 입력한 CPU는 이러한 상태에 더 오래 남아 있을 수 있습니다. 그러나 불필요한 타이머 이벤트를 생성하는 애플리케이션이 있는 경우 이 기능의 이점을 오프셋할 수 있습니다. 볼륨 변경 또는 마우스 이동 확인과 같은 폴링 이벤트는 이러한 이벤트의 예입니다.

Red Hat Enterprise Linux 7에는 CPU 사용량을 기반으로 애플리케이션을 식별하고 감사할 수 있는 툴이 포함되어 있습니다. 자세한 내용은 [2장. 전원 관리 감사 및 분석](#) 을 참조하십시오.

사용되지 않는 하드웨어 및 장치는 완전히 비활성화되어야 합니다.

이는 특히 일부를 이동하는 장치(예: 하드 디스크)에서 마찬가지입니다. 이 외에도 일부 애플리케이션에서는 사용되지 않지만 활성화된 장치 "오픈"을 남겨 둘 수 있습니다. 이 경우 커널은 장치가 사용 중인 것으로 가정하여 장치가 절전 상태로 전환되지 않도록 할 수 있습니다.

낮은 활동이 낮은 와트로 변환되어야 합니다.

그러나 대부분의 경우 최신 하드웨어 및 올바른 BIOS 구성에 따라 다릅니다. 이전 시스템 구성 요소는 현재 Red Hat Enterprise Linux 7에서 지원할 수 있는 새로운 기능 중 일부를 지원하지 않는 경우가 많습니다. 시스템에 대한 최신 공식 펌웨어를 사용하고 BIOS의 전원 관리 또는 장치 구성 섹션에서 전원 관리 기능이 활성화되어 있는지 확인합니다. 찾을 몇 가지 기능은 다음과 같습니다.

- SpeedStep
- PowerNow!

- Tyy't'Quiet
- ACPI (C state)
- Smart

하드웨어가 이러한 기능에 대한 지원이 있고 BIOS에서 활성화되어 있는 경우 Red Hat Enterprise Linux 7은 기본적으로 이를 사용합니다.

다양한 형태의 CPU 상태 및 해당 효과

최신 CPU와 ACSPI(*Advanced Configuration and Power Interface*)는 서로 다른 전원 상태를 제공합니다. 세 가지 다른 상태는 다음과 같습니다.

- 수면 (C-states)
- 빈도 및 변동 (P-상태)

P-state는 프로세서의 빈도와 출력 작동 지점을 설명합니다. 이는 P-state가 증가함에 따라 스케일링됩니다.

- heat 출력(T-상태 또는 "thermal 상태")

가장 낮은 수면 상태에서 실행되는 CPU는 가능한 한 적은 양의 와트를 사용하지만, 필요한 경우 해당 상태에서 작동시키는데 훨씬 더 많은 시간이 걸립니다. 매우 드문 경우지만, CPU가 잠자기 상태로 전환될 때마다 즉시 작동해야 할 수 있습니다. 이 경우 효과적으로 영구적으로 사용 중인 CPU가 생성되고 다른 상태가 사용된 경우 일부 잠재적인 정전이 손실됩니다.

꺼진 머신은 가장 적은 양의 전원을 사용합니다.

이것이 잘 보일 수 있듯이, 실제로 전원을 저장하는 가장 좋은 방법 중 하나는 시스템을 끄는 것입니다. 예를 들어, 귀사는 시간당 중단 또는 출근 시 시스템을 켤 수 있는 지침으로 "녹색 IT"에 중점을 둔 기업 문화를 개발할 수 있습니다. 또한 Red Hat Enterprise Linux 7에서 제공하는 가상화 기술을 사용하여 여러 물리적 서버를 하나의 큰 서버로 통합하고 가상화할 수 있습니다.

2장. 전원 관리 감사 및 분석

2.1. 감사 및 분석 개요

단일 시스템의 상세한 수동 감사, 분석 및 튜닝은 일반적으로 시간과 비용이 최근 시스템 튜닝에서 얻을 수 있는 이점을 간과하기 때문에 예외입니다. 그러나 모든 시스템에 대해 동일한 설정을 재사용할 수 있는 거의 동일한 시스템의 많은 수에 대해 이러한 작업을 한 번 수행하면 매우 유용할 수 있습니다. 예를 들어, 수천 개의 데스크탑 시스템 또는 시스템이 거의 동일한 HPC 클러스터를 배포하는 것이 좋습니다. 감사 및 분석을 수행해야 하는 또 다른 이유는 향후 시스템 동작의 회귀 또는 변경 사항을 확인할 수 있는 비교 기준을 제공하기 위한 것입니다. 이 분석의 결과는 하드웨어, BIOS 또는 소프트웨어 업데이트가 정기적으로 발생하는 경우 매우 도움이 될 수 있으며 전력 소비와 관련하여 놀라운 것을 피하고자합니다. 일반적으로 철저히 감사 및 분석을 통해 특정 시스템에서 실제로 발생하는 상황을 훨씬 더 잘 파악할 수 있습니다.

전력 소비와 관련하여 시스템을 감사하고 분석하는 것은 최신 시스템을 사용할 수 있더라도 비교적 어렵습니다. 대부분의 시스템은 소프트웨어를 통해 전력 사용을 측정하는 데 필요한 수단을 제공하지 않습니다. 예외는 존재하지만: Hewlett Packard 서버 시스템의 ILO 관리 콘솔에는 웹을 통해 액세스할 수 있는 전원 관리 모듈이 있습니다. IBM은 BladeCenter 전원 관리 모듈에서 유사한 솔루션을 제공합니다. 일부 Dell 시스템에서 IT Assistant는 전원 모니터링 기능도 제공합니다. 다른 벤더는 서버 플랫폼에 대해 유사한 기능을 제공할 가능성이 있지만 모든 공급 업체에서 지원하는 단일 솔루션은 없습니다.

전력 소비에 대한 직접 측정은 종종 최대한의 비용 절감을 극대화하기 위해 필요합니다. 다행스럽게도 변경 사항이 적용되는지 또는 시스템이 어떻게 작동하는지 측정하는 다른 수단을 사용할 수 있습니다. 이 장에서는 필요한 툴을 설명합니다.

2.2. POWERTOP

Red Hat Enterprise Linux 7에 터리스 커널을 도입하면 CPU가 유휴 상태에 더 자주 진입하여 전력 소비를 줄이고 전력 관리를 개선할 수 있습니다. **PowerTOP** 툴은 CPU를 자주 발생시키는 커널 및 사용자 공간 애플리케이션의 특정 구성 요소를 식별합니다. **PowerTOP** 는 이 릴리스에서 많은 애플리케이션이 조정되어 불필요한 CPU를 10배까지 줄인 감사를 수행하는 데 사용되었습니다.

Red Hat Enterprise Linux 7에는 **PowerTOP** 버전 2.x가 포함되어 있습니다. 이 버전은 1.x 코드 베이스를 완전히 다시 작성합니다. 이는 탭 기반 사용자 인터페이스를 제공하며 커널 "perf" 인프라를 사용하여 보다 정확한 데이터를 제공합니다. 시스템 장치의 전원 동작이 추적되고 눈에 띄게 표시되므로 문제를 신속하게 파악할 수 있습니다. 보다 실험적으로 2.x 코드베이스에는 개별 장치와 프로세스가 얼마나 많이 소비되는지를 나타낼 수 있는 전력 추정 엔진이 포함되어 있습니다. [그림 2.1. "운영의 PowerTOP"](#)을 참조하십시오.

PowerTOP 를 설치하려면 **root** 로 다음 명령을 실행합니다.

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

PowerTOP 를 실행하려면 **root** 로 다음 명령을 사용합니다.

```
~]# powertop
```

PowerTOP 는 시스템의 총 전력 사용량 추정치를 제공하고 각 프로세스, 장치, 커널 작업, 타이머 및 인터럽트 처리기에 대한 개별 전원 사용량을 표시할 수 있습니다. 이 작업 중에 노트북은 배터리 전원으로 실행되어야 합니다. 전원 추정 엔진을 조정하려면 **root** 로 다음 명령을 실행합니다.

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

자격 증명은 시간이 걸립니다. 이 프로세스는 다양한 테스트를 수행하고 밝기 수준을 통해 순환하고 장치

를 켜거나 끄십시오. 프로세스가 완료되고 조각화 중에 시스템과 상호 작용하지 않도록 하십시오. 조각화 프로세스가 완료되면 **PowerTOP** 가 정상적으로 시작됩니다. 데이터를 수집하기 위해 약 1시간 동안 실행되도록 합니다. 충분한 데이터가 수집되면 첫 번째 열에 전력 추정 수치가 표시됩니다.

랩탑에서 명령을 실행하는 경우에도 사용 가능한 모든 데이터가 표시되도록 배터리 전원을 계속 실행해야 합니다.

실행되는 동안 **PowerTOP** 는 시스템에서 통계를 수집합니다. 개요 탭에서는 CPU에 가장 자주 슬립을 보내거나 가장 많은 전원을 소비하는 구성 요소 목록을 볼 수 있습니다([그림 2.1. "운영의 PowerTOP"](#) 참조). 인접한 열에는 전력 추정, 리소스 사용 방법, 초당 레이닝, 구성 요소의 분류(예: 프로세스, 장치 또는 타이머) 및 구성 요소에 대한 설명이 표시됩니다. Wakeups per second는 커널의 서비스 또는 장치 및 드라이버를 얼마나 효율적으로 수행하는지 나타냅니다. 적은 잠자리는 더 적은 전력을 소비한다는 것을 의미합니다. 구성 요소는 전력 사용량을 얼마나 더 많이 최적화할 수 있는지에 따라 정렬됩니다.

일반적으로 드라이버 구성 요소에는 커널 변경이 필요합니다. 이 변경 사항은 이 문서의 범위를 벗어납니다. 그러나 wakeup을 보내는 userland 프로세스는 더 쉽게 관리할 수 있습니다. 먼저 이 서비스 또는 애플리케이션을 이 시스템에서 전혀 실행해야 하는지 확인합니다. 그렇지 않으면 단순히 비활성화하십시오. 이전 시스템 V 서비스를 영구적으로 비활성화하려면 다음을 실행합니다.

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

프로세스에 대한 자세한 내용은 **root** 로 다음 명령을 실행합니다.

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

추적이 반복되는 것처럼 보이면 빠르게 루프가 될 수 있습니다. 이러한 버그를 수정하려면 일반적으로 해당 구성 요소의 코드 변경이 필요합니다.

[그림 2.1. "운영의 PowerTOP"](#) 에서 볼 수 있듯이 해당하는 경우 총 전력 소비와 나머지 배터리 수명이 표시됩니다. 다음은 초당 총 운동, 초당 GPU 작업 및 초당 가상 파일 시스템 작업을 나타내는 간략한 요약입니다. 화면의 나머지 부분에는 사용률에 따라 정렬된 프로세스, 인터럽트, 장치 및 기타 리소스 목록이 있습니다. 올바르게 조정하면 첫 번째 열에 나열된 모든 항목에 대한 전력 소비 추정도 표시됩니다.

Tab 및 **Shift+Tab** 키를 사용하여 탭을 순환합니다. **Idle stats** 탭에서는 모든 프로세서 및 코어에 대해 C-states의 사용이 표시됩니다. 빈도 통계 탭에서는 모든 프로세서 및 코어에 대해 ReplicaSet 모드(해당되는 경우)를 포함한 P-상태의 사용이 표시됩니다. CPU가 더 높은 C 또는 P 상태에 더 오래 남아 있을수록 **C4** 가 **C3**보다 높습니다. CPU 사용량이 얼마나 잘 최적화되었는지를 나타내는 좋은 지표입니다. 레지던스는 시스템이 유휴 상태인 동안 C- 또는 P-상태가 가장 높은 90% 이상이어야 합니다.

장치 통계 탭은 개요 탭과 유사한 정보를 제공하지만 장치에 대해서만 제공됩니다.

Tunables 탭에는 더 낮은 전력 소비를 위해 시스템을 최적화하기 위한 제안 사항이 포함되어 있습니다. 위쪽 및 아래쪽 키를 사용하여 제안 사항을 이동하고 **Enter** 키를 사용하여 제안을 켜거나 끕니다.

그림 2.1. 운영의 PowerTOP

```

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

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

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

[D]

--html 옵션으로 PowerTOP 를 실행하여 HTML 보고서를 생성할 수도 있습니다. *htmlfile.html* 매개변수를 출력 파일에 필요한 이름으로 바꿉니다.

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

기본적으로 PowerTOP 는 20초 간격으로 측정을 수행하므로 --time 옵션을 사용하여 변경할 수 있습니다.

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

PowerTOP 에 대한 자세한 내용은 [PowerTOP 홈 페이지](#)를 참조하십시오.

PowerTOP 는 **turbostat** 유틸리티와 함께 사용할 수도 있습니다. Intel 64 프로세서에서 프로세서 토폴로지, 빈도, 유희 전원 상태 통계, 온도 및 전원 사용량에 대한 정보를 표시하는 보고 틀입니다. **turbostat** 유틸리티에 대한 자세한 내용은 **turbostat(8)** 도움말 페이지를 참조하거나 [성능 튜닝 가이드](#)를 참조하십시오.

2.3. DISKDEVSTAT 및 NETDEVSTAT

Diskdevstat 및 netdevstat 는 시스템에서 실행되는 모든 애플리케이션의 디스크 활동 및 네트워크 활동에 대한 자세한 정보를 수집하는 **SystemTap** 틀입니다. 이러한 틀은 PowerTOP 에서 활성화했으며, 이는 초당 모든 애플리케이션에서 CPU 레이닝 수를 표시합니다(2.2절. "PowerTOP"참조). 이러한 틀에서 수집

하는 통계를 사용하면 더 적은 수의 작업이 아닌 많은 작은 I/O 작업으로 전력을 낭비하는 애플리케이션을 식별할 수 있습니다. 전송 속도만 측정하는 다른 모니터링 도구는 이러한 유형의 사용을 식별하는 데 도움이 되지 않습니다.

root 로 다음 명령을 사용하여 **SystemTap** 으로 이러한 툴을 설치합니다.

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

명령을 사용하여 툴을 실행합니다.

```
~]# diskdevstat
```

또는 명령:

```
~]# netdevstat
```

두 명령 모두 다음과 같이 최대 3개의 매개변수를 사용할 수 있습니다.

diskdevstat *update_interval total_duration display_histogram*

netdevstat *update_interval total_duration display_histogram*

update_interval

디스플레이 업데이트 사이의 시간(초)입니다. 기본값: **5**

total_duration

전체 실행에 대한 시간(초)입니다. 기본값: **86400** (1일)

display_histogram

실행 종료 시 수집된 모든 데이터에 대한 히스토그램에 플래그를 지정합니다.

출력은 **PowerTOP** 와 유사합니다. 다음은 더 긴 **diskdevstat** 실행의 샘플 출력입니다.

```
PID UID DEV WRITE_CNT WRITE_MIN WRITE_MAX WRITE_AVG READ_CNT READ_MIN
READ_MAX READ_AVG COMMAND
2789 2903 sda1 854 0.000 120.000 39.836 0 0.000 0.000 0.000 plasma
5494 0 sda1 0 0.000 0.000 0.000 758 0.000 0.012 0.000 ologwatch
5520 0 sda1 0 0.000 0.000 0.000 140 0.000 0.009 0.000 perl
5549 0 sda1 0 0.000 0.000 0.000 140 0.000 0.009 0.000 perl
5585 0 sda1 0 0.000 0.000 0.000 108 0.001 0.002 0.000 perl
2573 0 sda1 63 0.033 3600.015 515.226 0 0.000 0.000 0.000 auditd
5429 0 sda1 0 0.000 0.000 0.000 62 0.009 0.009 0.000 crond
5379 0 sda1 0 0.000 0.000 0.000 62 0.008 0.008 0.000 crond
5473 0 sda1 0 0.000 0.000 0.000 62 0.008 0.008 0.000 crond
5415 0 sda1 0 0.000 0.000 0.000 62 0.008 0.008 0.000 crond
5433 0 sda1 0 0.000 0.000 0.000 62 0.008 0.008 0.000 crond
5425 0 sda1 0 0.000 0.000 0.000 62 0.007 0.007 0.000 crond
5375 0 sda1 0 0.000 0.000 0.000 62 0.008 0.008 0.000 crond
5477 0 sda1 0 0.000 0.000 0.000 62 0.007 0.007 0.000 crond
5469 0 sda1 0 0.000 0.000 0.000 62 0.007 0.007 0.000 crond
5419 0 sda1 0 0.000 0.000 0.000 62 0.008 0.008 0.000 crond
5481 0 sda1 0 0.000 0.000 0.000 61 0.000 0.001 0.000 crond
```

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

열은 다음과 같습니다.

PID

애플리케이션의 프로세스 ID

UID

애플리케이션이 실행되는 사용자 ID

DEV

I/O가 발생한 장치

WRITE_CNT

총 쓰기 작업 수

WRITE_MIN

연속 두 번의 쓰기에 걸린 가장 짧은 시간(초)

WRITE_MAX

연속 두 번 쓰기에 가장 큰 시간 (초)

WRITE_AVG

두 개의 연속 쓰기에 걸린 평균 시간(초)

READ_CNT

총 읽기 작업 수

READ_MIN

연속 두 개의 읽기에 사용된 가장 짧은 시간(초)

READ_MAX

두 개의 연속 읽기에 걸린 가장 큰 시간(초)

READ_AVG

연속 두 읽기에 걸리는 평균 시간(초)

명령

프로세스의 이름

이 예에서는 매우 명확한 세 가지 애플리케이션이 존재합니다.

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

이 세 가지 애플리케이션은 **0** 보다 큰 **WRITE_CNT** 를 가지며 이는 측정 중에 어떤 형태의 쓰기를 수행했음을 의미합니다. 그 중 **Microsoft** 는 가장 큰 쓰기 작업을 수행했으며, 물론 쓰기 사이의 평균 시간이 가장 낮음에 따라 가장 큰 요인이었습니다. 따라서, 가상 시스템은 전력 효율이 없는 애플리케이션에 대해 우려하고 있는지 조사할 수 있는 최상의 후보입니다.

지정된 프로세스 ID의 모든 시스템 호출을 추적하여 **strace** 및 **ltrace** 명령을 사용하여 애플리케이션을 더 자세히 검사합니다. 현재 예제에서는 다음을 실행할 수 있습니다.

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

이 예에서 **strace** 의 출력에 45초마다 반복 패턴이 포함되어 있으며, 이 패턴은 사용자의 쓰기와 파일을 즉시 종료하는 데 필요한 반복 패턴을 열었습니다. 이로 인해 파일 메타데이터(특히 수정 시간)가 변경될 때 하드 디스크에 필요한 물리적 쓰기가 발생했습니다. 최종 수정은 아이콘 업데이트가 발생하지 않은 경우 불필요한 호출을 방지하는 것이었습니다.

2.4. 배터리 수명 도구 키트

Red Hat Enterprise Linux 7에는 배터리 수명과 성능을 시뮬레이션하고 분석하는 테스트 모음인 **BLTK(Battery Life Tool Kit)**가 도입되었습니다. BLTK 는 특정 사용자 그룹을 시뮬레이션하고 결과를 보고하는 일련의 작업을 수행하여 이를 달성합니다. 노트북 성능을 테스트하기 위해 특별히 개발되었지만 BLTK 는 **-a** 로 시작될 때 데스크탑 컴퓨터의 성능을 보고할 수도 있습니다.

BLTK를 사용하면 머신의 실제 사용과 비교할 수 있는 매우 재현 가능한 워크로드를 생성할 수 있습니다. 예를 들어, 사무실 워크로드는 텍스트를 작성하고, 그 안에 내용을 수정하며, 웹 사이트에 대해 동일한 작업을 수행합니다. **PowerTOP** 또는 다른 감사 또는 분석 도구를 결합한 BLTK를 실행하면 시스템이 유휴 상태일뿐만 아니라 시스템이 적극적으로 사용 중인 경우 효과가 있는지 테스트할 수 있습니다. 다른 설정에 대해 정확히 동일한 워크로드를 여러 번 실행할 수 있으므로 서로 다른 설정에 대한 결과를 비교할 수 있습니다.

명령을 사용하여 BLTK 를 설치합니다.

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

다음 명령을 사용하여 BLTK 를 실행합니다.

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

예를 들어 유휴 워크로드를 120초 동안 실행하려면 다음을 수행합니다.

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

기본적으로 사용 가능한 워크로드는 다음과 같습니다.

-l, --idle

시스템이 유휴 상태이며 다른 워크로드와 비교하기 위한 기준으로 사용됩니다.

-R, --reader

문서 읽기 시뮬레이션(기본적으로 **Firefox** 사용)

-P, --player

CD 또는 DVD 드라이브에서 멀티미디어 파일을 감시하는 시뮬레이션(기본적으로 **m Cryostat**)

-O, --office

Cryostat.org 제품군을 사용하여 문서 편집 시뮬레이션

다른 옵션을 사용하면 다음을 지정할 수 있습니다.

-a, --ac-ignore

AC 전원을 사용할 수 있는지 여부를 무시합니다(데스크탑 사용 필수)

-T number_of_seconds, --time number_of_seconds

테스트를 실행하는 시간(초)입니다. 이 옵션을 유휴 워크로드와 함께 사용하십시오.

-F filename, --file filename

특정 워크로드에서 사용할 파일을 지정합니다(예: CD 또는 DVD 드라이브에 액세스하는 대신 플레이어 워크로드 파일)

-W 애플리케이션, --prog application

리더 워크로드에 대해 **Firefox** 이외의 브라우저와 같은 특정 워크로드에서 사용할 애플리케이션을 지정합니다.

BLTK 는 더 많은 특수 옵션을 지원합니다. 자세한 내용은 **bltk** 도움말 페이지를 참조하십시오.

BLTK 는 **/etc/bltk.conf** 구성 파일에 지정된 디렉터리에 생성되는 결과를 저장합니다. 기본적으로 **~/.bltk/workload.results.number/**. 예를 들어 **~/.bltk/reader.results.002/** 디렉터리에 리더 워크로드와 함께 세 번째 테스트의 결과가 있습니다(첫 번째 테스트는 번호가 매겨지지 않음). 결과는 여러 텍스트 파일에 분산됩니다. 이러한 결과를 읽기 쉬운 형식으로 축소하려면 다음을 실행합니다.

```
~]$ bltk_report path_to_results_directory
```

이제 결과 디렉터리에 **Report** 라는 텍스트 파일이 표시됩니다. 대신 터미널 에뮬레이터에서 결과를 보려면 **-o** 옵션을 사용합니다.

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

2.5. TUNED

tuned 는 **udev** 장치 관리자를 사용하여 연결된 장치를 모니터링하는 프로필 기반 시스템 튜닝 툴이며 시스템 설정을 정적 및 동적 튜닝할 수 있습니다. 동적 튜닝은 실험적인 기능이며 Red Hat Enterprise Linux 7에서 기본적으로 꺼져 있습니다.

tuned 는 높은 처리량, 짧은 대기 시간 또는 절전과 같은 일반적인 사용 사례를 처리하는 사전 정의된 프로필을 제공합니다. 각 프로필에 대한 **Tuned** 규칙을 수정하고 특정 장치를 조정하는 방법을 사용자 지정할 수 있습니다. 사용자 정의 **Tuned** 프로필을 생성하는 방법에 대한 자세한 내용은 **PowerTOP** 제안에서 "[powertop2tuned 사용](#)" 을 참조하십시오.

프로필은 사용 중인 제품에 따라 자동으로 기본값으로 설정됩니다. **tuned-adm recommend** 명령을 사용하여 Red Hat이 특정 제품에 가장 적합한 프로필을 결정할 수 있습니다. 권장 사항이 없는 경우 **balanced** 프로필이 설정됩니다.

대부분의 워크로드에 적합한 균형 잡힌 프로파일은 에너지 소비, 성능 및 대기 시간 균형을 유지합니다. 균형 잡힌 프로필을 사용하면 사용 가능한 최대 컴퓨팅 성능을 갖춘 작업을 빠르게 완료하면 일반적으로 컴퓨팅 성능이 적은 기간 동안 동일한 작업을 수행하는 것보다 적은 에너지가 필요합니다.

powersave 프로필을 사용하면 랩탑이 유휴 상태인 경우 배터리 수명이 연장되거나 계산되지 않은 작업만 수행할 수 있습니다. 이러한 작업을 위해 더 낮은 에너지 소비를 위한 더 높은 대기 시간이 일반적으로 허용되거나, IRC를 사용하거나, 간단한 웹 페이지를 보거나, 오디오 및 비디오 파일을 재생하는 등 작업을 신속하게 완료할 필요가 없습니다.

tuned-adm 과 함께 제공되는 **Tuned** 및 전원 항상 프로파일에 대한 자세한 내용은 *Red Hat Enterprise Linux 7 성능 튜닝 가이드*의 [Tuned 장을](#) 참조하십시오.

powertop2tuned 사용

powertop2tuned 유틸리티를 사용하면 **PowerTOP** 제안에서 사용자 정의 **Tuned** 프로필을 생성할 수 있습니다. **PowerTOP** 에 대한 자세한 내용은 [2.2절. "PowerTOP"](#) 을 참조하십시오.

powertop2tuned 유틸리티를 설치하려면 다음을 사용합니다.

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

사용자 지정 프로필을 생성하려면 다음을 사용합니다.

```
~]# powertop2tuned new_profile_name
```

기본적으로 **powertop2tuned** 는 **/etc/tuned/** 디렉터리에 프로필을 생성하고 현재 선택한 **Tuned** 프로필에서 사용자 지정 프로필을 기반으로 합니다. 보안상의 이유로 모든 **PowerTOP** 튜닝은 처음에 새 프로필에서 비활성화됩니다. 튜닝을 활성화하려면 **/etc/tuned/profile_name/tuned.conf** 파일에서 주석 처리를 해제합니다.

--enable 또는 **-e** 옵션을 사용하여 **PowerTOP** 에서 제안하는 대부분의 튜닝을 활성화하는 새 프로필을 생성할 수 있습니다. USB 자동 일시 중지와 같은 문제가 발생할 수 있는 특정 튜닝은 기본적으로 비활성화되어 수동으로 주석을 제거해야 합니다.

기본적으로 새 프로필은 활성화되지 않습니다. 활성화하려면 다음을 사용합니다.

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

powertop2tuned 가 지원하는 전체 옵션 목록은 다음을 사용합니다.

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

2.6. UPOWER

Red Hat Enterprise Linux 6 **DeviceKit-power** 는 **HAL** 의 일부인 전원 관리 기능과 Red Hat Enterprise Linux의 이전 릴리스에서 **GNOME Power Manager** 의 일부인 일부 기능을 가정합니다(2.7절. “GNOME Power Manager” 참조). Red Hat Enterprise Linux 7의 **DeviceKit-power** 는 **UPower** 로 이름이 변경되었습니다. **upower**는 데몬, API 및 명령줄 툴 세트를 제공합니다. 시스템의 각 전원 소스는 물리적 장치인지 여부에 관계없이 장치로 표시됩니다. 예를 들어 노트북 배터리와 AC 전원 소스는 모두 장치로 표시됩니다.

upower 명령과 다음 옵션을 사용하여 명령줄 툴에 액세스할 수 있습니다.

--enumerate, -e

시스템의 각 전원 장치에 대한 오브젝트 경로를 표시합니다. 예를 들면 다음과 같습니다.

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

--dump, -d

시스템의 모든 전원 장치에 대한 매개 변수를 표시합니다.

--wakeups, -w

시스템의 CPU 발생을 표시합니다.

--monitor, -m

전원 장치 변경, 예를 들어 AC 전원 공급의 연결 또는 연결 해제 또는 배터리 소모를 모니터링하는 시스템을 모니터링합니다. **Ctrl+C** 눌러 시스템 모니터링을 중지합니다.

--monitor-detail

전원 장치 변경, 예를 들어 AC 전원 공급의 연결 또는 연결 해제 또는 배터리 소모를 모니터링하는 시스템을 모니터링합니다. **--monitor-detail** 옵션은 **--monitor** 옵션보다 자세히 표시됩니다. **Ctrl+C** 눌러 시스템 모니터링을 중지합니다.

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

특정 오브젝트 경로에 사용 가능한 모든 정보를 표시합니다. 예를 들어 오브젝트 경로 **/org/freedesktop/UPower/devices/battery_BAT0** 로 표시되는 시스템의 배터리에 대한 정보를 얻으려면 다음을 실행합니다.

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

2.7. GNOME POWER MANAGER

GNOME Power Manager 는 GNOME 데스크탑 환경의 일부로 설치된 데몬입니다. 이전 버전의 Red Hat Enterprise Linux에서 제공하는 대부분의 전원 관리 기능은 Red Hat Enterprise Linux 6의 **DeviceKit-power** 툴의 일부가 되었습니다. Red Hat Enterprise Linux 7에서 **UPower** 로 이름이 변경되었습니다(2.6절. “UPower” 참조). 그러나 **GNOME Power Manager** 는 해당 기능의 프론트 엔드로 유지됩니다. 시스템 트레이를 통해 **GNOME Power Manager** 는 시스템의 전원 상태에 대한 변경 사항을 알려줍니다. 예를 들어 배터리에서 AC 전원으로 변경됩니다. 또한 배터리 상태를 보고하고 배터리 전원이 낮은 경우 경고합니다.

2.8. 감사용 기타 툴

Red Hat Enterprise Linux7은 시스템 감사 및 분석을 수행하는 몇 가지 추가 툴을 제공합니다. 대부분은 귀하가 이미 발견한 내용을 확인하거나 특정 부분에 대한 보다 심층적인 정보가 필요한 경우 정보의 보조 소스로 사용할 수 있습니다. 이러한 툴 중 대부분은 성능 튜닝에도 사용됩니다. 다음이 포함됩니다.

vmstat

vmstat 는 프로세스, 메모리, 페이징, 블록 I/O, 트랩 및 CPU 활동에 대한 자세한 정보를 제공합니다. 이를 사용하여 시스템 전체에서 수행하는 작업과 사용 중인 위치를 자세히 살펴보십시오.

iostat

iostat 은 **vmstat** 와 유사하지만 블록 장치의 I/O에만 해당합니다. 또한 더 자세한 출력 및 통계도 제공합니다.

blktrace

blktrace 는 매우 자세한 블록 I/O 추적 프로그램입니다. 애플리케이션과 관련된 단일 블록으로 정보를 분리합니다. **diskdevstat** 와 함께 매우 유용합니다.

3장. 코어 인프라 및 메커니즘



중요

이 장에서 제공하는 **cpupower** 명령을 사용하려면 kernel-tools 패키지가 설치되어 있어야 합니다.

3.1. CPU 유휴 상태

x86 아키텍처가 있는 CPU는 CPU의 일부가 비활성화되거나 더 낮은 성능 설정에서 실행되는 다양한 상태를 지원합니다. C 상태라고도 하는 이러한 상태를 사용하면 사용하지 않는 CPU를 부분적으로 비활성화하여 시스템을 전원을 절약할 수 있습니다. C-states는 C0 이상에서 번호가 매겨지며, 숫자가 클수록 CPU 기능이 줄어들고 전력 절약이 향상됩니다. 지정된 수의 C-States는 프로세서마다 광범위하게 유사하지만 특정 상태 집합의 정확한 세부 사항은 프로세서 제품군마다 다를 수 있습니다. C-States 0-3는 다음과 같이 정의됩니다.

C0

작동 상태 또는 실행 상태입니다. 이 상태에서 CPU가 작동 중이며 전혀 유휴 상태가 아닙니다.

C1, halt

프로세서가 명령을 실행하지 않고 일반적으로 더 낮은 전원 상태가 아닌 상태입니다. CPU는 지연 없이 거의 처리를 계속할 수 있습니다. C-States를 제공하는 모든 프로세서는 이 상태를 지원해야 합니다. Pentium 4 프로세서는 실제로 더 낮은 전력 소비를 위한 상태인 C1E라는 향상된 C1 상태를 지원합니다.

C2, stop-Clock

이 프로세서에 대해 시계가 고정되지만 레지스터 및 캐시에 대한 전체 상태를 유지하므로 시계를 다시 시작하면 즉시 처리를 다시 시작할 수 있습니다. 이는 선택적 상태입니다.

C3, 절전

프로세서가 실제로 잠자고 캐시를 최신 상태로 유지할 필요가 없는 상태입니다. 이 상태에서 시작하는 것은 이로 인해 C2보다 훨씬 오래 걸립니다. 이는 선택적 상태입니다.

CPUidle 드라이버의 사용 가능한 유휴 상태 및 기타 통계를 보려면 다음을 입력합니다.

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

"Nehalem" 마이크로 아키텍처가 있는 최근 Intel CPU는 CPU의 전원 공급을 0으로 줄일 수 있는 새로운 C-State, C6을 갖추고 있지만 일반적으로 전력 소비를 80%에서 90%로 줄일 수 있습니다. Red Hat Enterprise Linux 7의 커널에는 이 새로운 C-State에 대한 최적화가 포함되어 있습니다.

3.2. CPUFREQ

시스템의 전력 소비와 열 출력을 줄이는 가장 효과적인 방법 중 하나는 CPUfreq입니다. cpufreq (CPU 속도 스케일링이라고도 함)는 전원을 절약하기 위해 CPU 빈도를 확장할 수 있는 Linux 커널의 인프라입니다. CPU 호출은 시스템 로드, ACPI 이벤트에 대한 응답으로 자동으로 또는 사용자 공간 프로그램에 의해 수동으로 수행될 수 있으며 프로세서의 클럭 속도를 즉시 조정할 수 있습니다. 이를 통해 시스템은 전력을 절약하기 위해 클럭 속도를 줄일 수 있습니다. 빈도를 더 빠르거나 느린 클럭 속도로 전환하는 규칙, 주파수를 전환할 시기는 CPUfreq governor에 의해 정의됩니다.

3.2.1. cpufreq 드라이버

CPufreq용 두 가지 드라이버인 ACPI CPUfreq 드라이버와 Intel P-state 드라이버를 사용할 수 있습니다.

ACPI CPUfreq

ACPI CPUfreq 드라이버는 커널과 하드웨어 간의 통신을 보장하는 ACPI를 통해 특정 CPU의 빈도를 제어하는 커널 드라이버입니다.

Intel P-state

Red Hat Enterprise Linux 7에서는 Intel P-state 드라이버가 지원됩니다. 이 드라이버는 Intel Xeon E 시리즈 아키텍처 또는 최신 아키텍처를 기반으로 프로세서에서 P-state 선택을 제어하기 위한 인터페이스를 제공합니다. Intel P-state는 `setpolicy()` 콜백을 구현합니다. 드라이버는 `cpufreq` 코어에서 요청한 정책에 따라 사용할 P-state를 결정합니다. 프로세서가 내부적으로 다음 P-상태를 선택할 수 있는 경우, 드라이버는 이러한 책임을 프로세서에 오프로드합니다. 그렇지 않은 경우 드라이버는 알고리즘을 구현하여 다음 P-state를 선택합니다.

Intel P-state는 자체 `sysfs` 파일을 제공하여 P-state 선택을 제어합니다. 이러한 파일은 `/sys/devices/system/cpu/intel_pstate/` 디렉터리에 있습니다. 파일의 모든 변경 사항은 모든 CPU에 적용됩니다. 이 디렉터리에는 P-state 매개변수 설정에 사용되는 5개의 파일이 포함되어 있습니다.

- **max_perf_pct:** 사용 가능한 성능의 백분율로 표시된 드라이버에서 요청한 최대 P-상태를 제한합니다. `no_turbo` 설정으로 사용 가능한 P-state 성능을 줄일 수 있습니다(아래 참조).
- **min_perf_pct:** 드라이버에서 요청한 최소 P-상태를 제한하고 최대 (no-turbo) 성능 수준의 백분율로 표시됩니다.
- **no_turbo:** turbo 주파수 범위 아래에 P-state를 선택하도록 드라이버를 제한합니다.
- **turbo_pct:** turbo 범위에 있는 하드웨어에서 지원하는 총 성능의 백분율을 표시합니다. 이 수는 turbo가 비활성화되었는지 여부에 관계없이 독립적입니다.
- **num_pstates:** 하드웨어에서 지원하는 P 상태 수를 표시합니다. 이 수는 turbo가 비활성화되었는지 여부에 관계없이 독립적입니다.

현재 Intel P-state는 기본적으로 지원되는 CPU에 사용됩니다. 사용자는 커널 명령줄에 다음을 추가하여 ACPI CPUfreq 사용으로 전환할 수 있습니다.

```
intel_pstate=disable
```

3.2.2. cpufreq Cryostats

CPufreq governor는 시스템 CPU의 전원 특성을 정의합니다. 이 특성은 CPU 성능에 영향을 미칩니다. 각 관리자는 워크로드 측면에서 고유한 동작, 목적 및 적합성을 갖습니다. 이 섹션에서는 CPUfreq governor를 선택하고 구성하는 방법, 각 governor의 특성, 각 관리자가 적합한 워크로드 유형에 대해 설명합니다.



주의

Red Hat Enterprise Linux 7에는 여러 코어 CPUfreq governor가 포함되어 있습니다. Intel P-state 드라이버는 기본적으로 활성 모드에서 작동하며 두 개의 CPUfreq governor만 사용할 수 있습니다(*performance* 및 *powersave*). 성능 및 전원 저장 Intel P-state CPUfreq governor의 기능은 동일한 이름의 코어 CPUfreq governor와 다릅니다.

3.2.2.1. 코어 CPUfreq Cryostat

Red Hat Enterprise Linux 7에서 제공되는 다양한 유형의 CPUfreq governor는 다음과 같습니다.

cpufreq_performance

Performance governor는 CPU에서 가능한 가장 높은 클럭 빈도를 사용하도록 강제 적용합니다. 이 빈도는 정적으로 설정되며 변경되지 않습니다. 이와 같이, 이 특정 관리자는 *비용 절감 이점을 제공하지 않습니다*. 워크로드가 많은 시간 동안만 적합하며 CPU가 거의 (또는 never) 유휴 상태가 아닌 경우에만 적합합니다.

cpufreq_powersave

반대로 Powersave governor는 CPU가 가능한 가장 낮은 클럭 빈도를 사용하도록 강제 적용합니다.

cpufreq_ondemand

cpufreq_userspace

cpufreq_conservative



참고

3.2.2.2.

-
-
-
- 성능
-
- 성능
-

3.2.3.

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

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

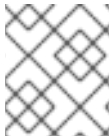
```
~]# cpupower -c 1-3,5 frequency-set --governor cpufreq_userspace
```

3.2.4.

-
-
-
-
-
-
-
-
-
-

•  중요

-
-



참고

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

3.3.

-
-
-

3.4.

--perf-bias <0-15>

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

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

3.5.

3.6.

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

auto

autosuspend_delay_ms

3.7.

default

powersave

performance

- `pcie_aspm=off` disables ASPM
-

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



-
-

3.8.

min_power

medium_power

max_performance

3.9.

3.10.

3.11.

3.12. RFKILL


```
~]# rfcill block 0
```

예를 들면 다음과 같습니다.

```
~]# rfcill block wifi
```

```
~]# rfcill block all
```

4장. 사용 사례

4.1.

Webserver

웹 서버에는 네트워크 및 디스크 I/O가 필요합니다. 외부 연결 속도 **100 Mbit/s**에 따라 충분합니다. 시스템이 대부분 정적 페이지를 제공하는 경우 **CPU** 성능은 중요하지 않을 수 있습니다. 따라서 전원 관리 선택에는 다음이 포함될 수 있습니다.

- **tuned**의 디스크 또는 네트워크 플러그인이 없습니다.
- **ALPM**이 켜졌습니다.
- 온 디맨드 관리자가 설정되었습니다.
- 네트워크 카드가 **100Mbit/s**로 제한됩니다.

컴퓨팅 서버

컴퓨팅 서버에는 주로 **CPU**가 필요합니다. 전원 관리 선택 사항은 다음과 같습니다.

- 작업과 데이터 스토리지가 발생하는 위치에 따라 **tuned**의 디스크 또는 네트워크 플러그인 또는 일괄 모드 시스템의 경우 완전히 활성화된.
- 사용률에 따라 성능 관리자일 수 있습니다.

mailserver

메일 서버에는 대부분 디스크 I/O 및 **CPU**가 필요합니다. 전원 관리 선택 사항은 다음과 같습니다.

- 온 디맨드 관리자는 지난 몇 %의 **CPU** 성능이 중요하지 않기 때문에 켜졌습니다.
- **tuned**의 디스크 또는 네트워크 플러그인이 없습니다.
- 메일은 종종 내부적인 경우가 많기 때문에 네트워크 속도를 제한해서는 안되며 **1Gbit/s** 또는 **10 Gbit/s** 링크의 이점을 얻을 수 있습니다.

fileserver

fileserver 요구 사항은 **mailserver**와 유사하지만 사용된 프로토콜에 따라 더 많은 **CPU** 성능이 필요할 수 있습니다. 일반적으로 **Samba** 기반 서버에는 **NFS**보다 더 많은 **CPU**가 필요하며 **NFS**에는 일반적으로 **iSCSI**보다 많은 **CPU**가 필요합니다. 마찬가지로, 귀하는 온 디맨드 매니저를 사용할 수 있어야 합니다.

디렉터리 서버

일반적으로 디렉터리 서버에 디스크 I/O 요구 사항이 더 적습니다(특히 충분한 **RAM**이 탑재된 경우). 네트워크 I/O가 덜지만 네트워크 대기 시간이 중요합니다. 연결 속도가 낮은 대기 시간 네트워크 튜닝을 고려할 수 있지만 특정 네트워크에 대해 신중하게 테스트해야 합니다.

4.2. 예 - 랩탑

전력 관리 및 비용 절감이 실제로 랩탑을 만들 수 있는 또 다른 매우 일반적인 공간은 다음과 같습니다. 일반적으로 설계된 랩탑은 이미 워크스테이션 또는 서버보다 훨씬 적은 에너지를 사용하므로 절대 비용 절감 가능성이 다른 시스템의 경우보다 적습니다. 그러나 배터리 모드에서 저장은 노트북에서 몇 분 더 많은 배터리 수명을 얻는 데 도움이 될 수 있습니다. 이 섹션은 배터리 모드의 랩탑에 중점을 두고 있지만 **AC** 전원으로 실행되는 동안 여전히 일부 또는 모든 튜닝을 사용할 수 있습니다.

일반적으로 단일 구성 요소에 대한 비용 절감은 노트북에서 워크스테이션보다 더 큰 상대적 차이를 만듭니다. 예를 들어 100Mbit/s에서 실행되는 1Gbit/s 네트워크 인터페이스는 약 3~4 와트에 저장됩니다. 총 전력 소비가 약 400 와트인 일반적인 서버의 경우 이 저장은 약 1%입니다. 전체 전력 소비가 약 40 와트의 노트북에서, 이 하나의 구성 요소에만 전력 절감은 합계의 10%입니다.

일반적인 랩탑에서 특정 전원 개선 최적화는 다음과 같습니다.

- 사용하지 않는 모든 하드웨어를 비활성화하도록 시스템 BIOS를 구성합니다. 예를 들어 병렬 또는 직렬 포트, 카드 리더, *ines, everyone* 및 Bluetooth와 같은 몇 가지 가능한 후보 이름을 지정할 수 있습니다.
- 화면을 읽는 데 완전히 어울릴 필요가 없는 어두운 환경에서 디스플레이를 표시합니다. GNOME 데스크탑에서 시스템+환경 설정, Kickoff Application Launcher+Computer+System Settings+Advanced → Power Management 를 사용하거나 명령줄의 `gnome-power-manager` 또는 `xbacklight` 또는 랩탑의 기능 키를 사용합니다.

또한 (또는 대안으로) 다양한 시스템 설정에 대해 많은 작은 조정을 수행할 수 있습니다.

- 온디맨드 관리자 사용 (Red Hat Enterprise Linux 7에서 기본적으로 사용)
- AC97 오디오 전원 활성화 (Red Hat Enterprise Linux 7에서 기본적으로 사용)

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

- USB 자동 중지를 활성화합니다.

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

USB 자동 일시 중지가 모든 USB 장치에서 제대로 작동하지 않습니다.

- `relatime`을 사용하여 파일 시스템을 마운트합니다 (Red Hat Enterprise Linux 7의 기본값).

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

- 화면 밝기를 50 이하로 줄입니다. 예를 들면 다음과 같습니다.

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

- 화면 유휴 상태에 대해 DPMS를 활성화합니다.

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

- Wi-Fi 비활성화:

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

부록 A. 개발자를 위한 팁

모든 좋은 프로그래밍 텍스트북은 메모리 할당 및 특정 기능의 성능에 대한 문제를 다룹니다. 소프트웨어를 개발할 때 소프트웨어가 실행되는 시스템에서 전력 소비를 증가시킬 수 있는 문제를 알고 있어야 합니다. 이러한 고려 사항은 모든 코드 라인에 영향을 미치지 않지만 성능을 위해 자주 병목 현상이 발생하는 영역에서 코드를 최적화할 수 있습니다.

종종 문제가 되는 일부 기술은 다음과 같습니다.

- 스레드 사용.
- 불필요한 **CPU** 발생 및 작동을 효율적으로 사용하지 않습니다. 잠길 필요가 있는 경우 한 번에 모든 작업을 수행하고 가능한 한 빨리 유휴 상태로 두십시오.
- 불필요하게 **[f]sync()** 사용.
- 불필요한 활성 폴링 또는 짧은 일반 타임아웃을 사용합니다. 대신 이벤트로 돌아갑니다.
- **wake-up**을 효율적으로 사용하지 않습니다.
- 비효율적인 디스크 액세스. 디스크 액세스가 자주 발생하지 않도록 큰 버퍼를 사용합니다. 한 번에 하나의 큰 블록을 작성합니다.
- 타이머를 비효율적으로 사용합니다. 가능한 경우 애플리케이션(또는 시스템 전반에 걸쳐)을 그룹화합니다.
- 과도한 **I/O**, 전력 사용량 또는 메모리 사용량(메모리 누출 포함)
- 불필요한 계산을 수행합니다.

다음 섹션에서는 이러한 영역 중 일부를 자세히 설명합니다.

A.1. 스레드 사용

스레드를 사용하면 애플리케이션이 개선되고 빨라지지만 모든 경우에 해당되지는 않습니다.

Python

Python은 **Global Lock Interpreter** 사용^[1] 따라서 스레드는 더 큰 **I/O** 작업에 대해서만 유용합니다. **unladen-swallow**^[2] 더 빠른 Python 구현으로 코드를 최적화할 수 있습니다.

Perl

Perl 스레드는 원래 포크 없이 시스템에서 실행되는 애플리케이션을 위해 생성되었습니다(예: **32비트 Windows** 운영 체제가 있는 시스템). Perl 스레드에서는 모든 단일 스레드(**Copy On Write**)에 대한 데이터가 복사됩니다. 사용자가 데이터 공유 수준을 정의할 수 있어야 하므로 데이터는 기본적으로 공유되지 않습니다. 데이터 공유의 경우 **threads::shared** 모듈을 포함해야 합니다. 그러나 데이터는 복사될 뿐만 아니라 (쓰기 시 복사) 모듈은 데이터에 대한 연결된 변수도 생성하므로 더 많은 시간이 걸리며 속도가 더 느립니다.^[3]

C

C 스레드는 동일한 메모리를 공유하고 각 스레드에는 자체 스택이 있으며 커널은 새 파일 설명자를 생성하고 새 메모리 공간을 할당할 필요가 없습니다. C는 더 많은 스레드에 대한 더 많은 **CPU** 지원을 실제로 사용할 수 있습니다. 따라서 스레드의 성능을 최대화하려면 C 또는 C++와 같은 하위 수준 언어를 사용합니다. 스크립

팅 언어를 사용하는 경우 C 바인딩 작성을 고려하십시오. **profilers**를 사용하여 코드의 잘못된 부분을 식별합니다.[4]

A.2. WAKE-UPS

많은 애플리케이션에서 구성 파일에서 변경 사항을 스캔합니다. 대부분의 경우 검사는 고정된 간격(예: 1분마다)으로 수행됩니다. 이 문제는 디스크가 스핀 다운에서 잠길 수 있기 때문에 문제가 될 수 있습니다. 가장 좋은 방법은 좋은 간격, 좋은 검사 메커니즘을 찾고, **inotify**를 사용하여 변경 사항을 확인하고 이벤트에 대응하는 것입니다. **inotify**는 파일 또는 디렉토리에 대한 다양한 변경 사항을 확인할 수 있습니다.

예를 들면 다음과 같습니다.

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

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

    fd = inotify_init();

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

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

    return EXIT_SUCCESS;
}
```

이 접근 방식의 장점은 수행할 수 있는 다양한 검사입니다.

주요 제한 사항은 시스템에서 제한된 수의 시계만 사용할 수 있다는 것입니다. 번호는 `/proc/sys/fs/inotify/max_user_watches`에서 얻을 수 있으며 변경할 수는 있지만 권장되지 않습니다. 또

한 `inotify` 가 실패하는 경우 코드가 다른 검사 방법으로 대체되어야 합니다. 즉, 일반적으로 소스 코드에서 `#if #define` 가 많은 발생을 의미합니다.

`inotify` 에 대한 자세한 내용은 `inotify(7)` 도움말 페이지를 참조하십시오.

A.3. FSYNC

`Fsync` I/O 비용이 많이 드는 작업으로 알려져 있지만 완전히 사실이 아닙니다.

`Firefox` 는 사용자가 링크를 클릭하여 새 페이지로 이동할 때마다 `sqlite` 라이브러리를 호출하는 데 사용되었습니다. `SQLite` 는 `fsync` 이라고 하며 파일 시스템 설정(데이터 정렬 모드가 있는 그대로 `ext3`) 때문에 아무 일도 없을 때 지연 시간이 길어졌습니다. 다른 프로세스가 대용량 파일을 동시에 복사하는 경우 시간이 오래 걸릴 수 있습니다(최대 30초).

그러나 `fsync` 이 전혀 사용되지 않은 다른 경우에는 `ext4` 파일 시스템으로 전환하면 문제가 발생했습니다. `Ext3` 은 몇 초마다 메모리를 플러시하고 디스크에 저장하는 데이터 정렬 모드로 설정되었습니다. 그러나 `ext4` 및 `laptop_mode` 를 사용하면 저장 간격이 길고 시스템이 예기치 않게 꺼지면 데이터가 손실될 수 있습니다. 이제 `ext4` 가 패치되어 있지만 애플리케이션의 설계를 신중하게 고려하고 `fsync` 를 적절하게 사용해야 합니다.

구성 파일을 읽고 쓰는 간단한 예는 파일의 백업 또는 데이터 손실 방법을 보여줍니다.

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

더 나은 접근 방식은 다음과 같습니다.

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

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

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

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

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

부록 B. 버전 내역

고침 2.2-9 7.7 GA 게시에 대한 문서 버전입니다.	Mon Aug 05 2019	Marie Doleželová
고침 2.2-6 7.4 GA 게시용 문서 버전입니다.	Mon Jul 24 2017	Marie Doleželová
고침 2.2-5 비동기 업데이트: Tuned 장 재작성	Tue Mar 21 2017	Milan Navrátil
고침 2.0-2 7.3 GA 게시 버전	Fri Oct 14 2016	Marie Doleželová
고침 2.0-1 7.2 GA 릴리스입니다.	Wed 11 Nov 2015	Jana Heves
고침 1-3 Core Infrastructure 및 Mechanics에서 잘못된 패키지 이름이 수정되었습니다.	Fri 19 Jun 2015	Jacquelynn East
고침 1-2 7.1 GA 버전	Wed 18 Feb 2015	Jacquelynn East
고침 1-1 7.1 베타 버전	Thu Dec 4 2014	Jacquelynn East
고침 1.0-9 7.0 GA 릴리스	Tue Jun 9 2014	Yoana Ruseva
고침 0.9-1 스타일 변경을 위한 재구축.	Fri May 9 2014	Yoana Ruseva
고침 0.9-0 검토 목적을 위해 이 도서의 Red Hat Enterprise Linux 7.0 릴리스.	Wed May 7 2014	Yoana Ruseva
고침 0.1-1 Red Hat Enterprise Linux 6 버전의 문서에서 분기됨	Thu Jan 17 2013	Jack Reed