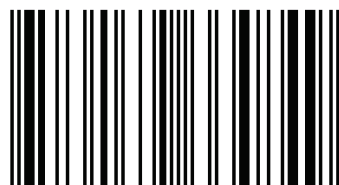


Criação de um regulador para o gerenciamento de energia de uma CPU

Com o grande aumento do uso de dispositivos móveis, existe um crescente interesse para o desenvolvimento de novas tecnologias mais eficientes, que gastem menos recursos e ofereçam mais tempo de uso por parte dos seus usuários. Neste trabalho, mostramos como foi feito o desenvolvimento de um algoritmo para a construção de um novo regulador de frequência para o gerenciamento da energia gasta pela unidade central de processamento de um computador. Foi feito também um comparativo com o regulador atualmente utilizado nas distribuições Linux do mercado, e o novo regulador mostrou-se eficiente tanto quanto os já usados nas atuais versões do kernel Linux.

Graduado em Sistemas de Informação pela Universidade Federal Rural do Semi-Árido - UFERSA, trabalhou com projetos de pesquisa envolvendo Aprendizado de Máquina, Inteligência Artificial, e desenvolveu seu trabalho de conclusão de curso na criação de um novo regulador para o gerenciamento de energia gasta em uma unidade central de processamento.



978-613-9-66484-9



Josinaldo Roberto Rocha Gomes · Leiva C. Oliveira

Criação de um regulador para o gerenciamento de energia de uma CPU

Desenvolvimento de um novo regulador de frequência da CPU a partir de estudo sobre o módulo de gerenciamento CPUFreq

Josinaldo Roberto Rocha Gomes
Leiva C. Oliveira

**Criação de um regulador para o gerenciamento de energia de uma
CPU**

**Josinaldo Roberto Rocha Gomes
Leiva C. Oliveira**

Criação de um regulador para o gerenciamento de energia de uma CPU

**Desenvolvimento de um novo regulador de
frequência da CPU a partir de estudo sobre o
módulo de gerenciamento CPUFreq**

Novas Edições Acadêmicas

Imprint

Any brand names and product names mentioned in this book are subject to trademark, brand or patent protection and are trademarks or registered trademarks of their respective holders. The use of brand names, product names, common names, trade names, product descriptions etc. even without a particular marking in this work is in no way to be construed to mean that such names may be regarded as unrestricted in respect of trademark and brand protection legislation and could thus be used by anyone.

Cover image: www.ingimage.com

Publisher:

Novas Edições Acadêmicas

is a trademark of

International Book Market Service Ltd., member of OmniScriptum Publishing Group

17 Meldrum Street, Beau Bassin 71504, Mauritius

Printed at: see last page

ISBN: 978-613-9-66484-9

Copyright © Josinaldo Roberto Rocha Gomes, Leiva C. Oliveira

Copyright © 2018 International Book Market Service Ltd., member of
OmniScriptum Publishing Group

All rights reserved. Beau Bassin 2018

Dedico este trabalho a minha família: **Josimar Gomes**, meu pai; **Nadja Rocha**, minha mãe; **Karla Oliveira**, minha esposa, e **Dolôres Neta**, enteada, por se constituírem pessoas admiráveis em minha existência e me impulsionarem a buscar vida nova a cada dia, privando-se de minha companhia quando precisei me dedicar aos momentos de estudo e me oportunizando a realização do sonho do crescimento pessoal e profissional. Também pela compreensão e apoio para conclusão deste trabalho.

Bem-aventurado o homem que acha sabedoria, e o homem que adquire conhecimento;
Porque é melhor a sua mercadoria do que artigos de prata, e maior o seu lucro que o ouro mais fino.

Provérbios 3:13,14

AGRADECIMENTOS

A **Deus**, pela oportunidade de ter direcionado a minha vida para o caminho do conhecimento, da sabedoria e da instrução. Também por ter me dado conhecimento para fazer as escolhas mais acertadas que puderam me trazer até aqui.

Ao meu pai, **Josimar Gomes**, que sempre foi fundamental para o meu crescimento, pelo seu amor e cuidado comigo, mesmo nas limitações. Amo-o muito. Sem sua dedicação eu não teria chegado até aqui.

À minha mãe, **Nadja Rocha**, que cuidou de mim como ninguém e sempre lutou para me dar um futuro melhor e de sucesso. E tem cumprido muitíssimo bem o seu papel de mãe na minha vida. Também a amo muito.

À minha esposa, **Karla Oliveira**, que tem sempre me apoiado e lutado comigo para que eu possa transpor esta fase da minha vida. Ter você ao meu lado só me dá mais força e satisfação em buscar uma carreira que antes de tudo é a de sermos melhores sempre.

Ao meu orientador, o professor **Leiva Casemiro**, pelas orientações e contribuições sempre valiosas durante as disciplinas TCC I e TCC II, as quais culminaram na realização deste trabalho.

Aos professores **Daniel Sabino** e **Francisco de Assis**, pela leitura atenta do meu Trabalho de Conclusão de Curso I e pelas valiosas contribuições para que chegássemos ao produto que agora apresento.

Aos meus **professores** do curso de Bacharelado em Sistemas de Informação da UFERSA/Campus Angicos, com os quais pude aprender um pouco mais e enriquecer minhas experiências. Agradeço pela sabedoria e o conhecimento repassados para que eu seja um profissional competente.

A todos os **amigos** e **amigas** que contribuíram direta ou indiretamente para a minha caminhada até aqui.

Aos professores: **Rita Diana, Daniel Sabino, Araken de Medeiros e Anderson Paiva**, pela oportunidade de compartilhar experiências de trabalho nos projetos Novos Talentos e TICs na Educação, no âmbito da Inclusão Digital.

Aos professores **Araken de Medeiros e Daniel Sabino**, pela oportunidade de desenvolver atividades de Iniciação Científica, visto ter sido uma experiência essencial na minha formação acadêmica, encaminhando-me para a pesquisa e para o universo da ciência como sujeito ativo.

À **Universidade Federal Rural do Semi-Árido**, por meio da **Coordenação de Assuntos Comunitários**, que através das ações de permanência acadêmica, oportunizou-me condições financeiras básicas para parte da minha estada em Angicos/RN.

Aos colegas do **Diretório Central dos Estudantes – DCE** da UFRSA/Angicos, pela oportunidade do exercício constante de cidadania.

À **cidade de Angicos**, pela acolhida, pelo tempo que nela residi, materializando a realização do meu sonho de cursar o ensino superior e me preparar para uma carreira profissional.

Obrigado!

LISTA DE FIGURAS

Figura 1 – Elementos do subsistema CPUFreq.....	19
Figura 2 – Reguladores de frequência do CPUFreq.....	20
Figura 3 – Núcleo do CPUFreq.....	21
Figura 4 – Drivers de dispositivos.....	21
Figura 5 – Aplicativo indicator-cpufreq para alteração de frequência.....	24
Figura 6 – Fluxograma do regulador proposto.....	31
Figura 7 – Estrutura de decisão do novo regulador.....	34

LISTA DE TABELAS

Tabela 1 – Dados estatísticos extraídos do arquivo /proc/stat.....	27
--	----

LISTA DE FÓRMULAS

Fórmula 1 – Fórmula da quantização dos contadores do arquivo /proc/stat.....	28
Fórmula 2 – Fórmula para calcular a taxa de utilização da(s) CPU(s).....	28
Fórmula 3 – Fórmula para o cálculo da penalidade de utilização da(s) CPU(s).....	30

LISTA DE GRÁFICOS

Gráfico 1 – Utilização X número de iterações para a frequência de 1,3GHz e taxa de amostragem variável.....	32
Gráfico 2 – Utilização X frequência variante e taxa de amostragem de 0,1s.....	33
Gráfico 3 – Cálculo da utilização do processador realizado pelo novo regulador em uma carga variante no tempo e uma taxa de amostragem de 0,1s.....	35
Gráfico 4 – Cálculo da utilização do processador realizado pelo regulador Ondemand em uma carga variante no tempo e uma taxa de amostragem de 0,1s.....	35
Gráfico 5 – Cálculo da penalidade do processador realizado pelo novo regulador em uma carga variante no tempo e uma taxa de amostragem de 0,1s.....	36
Gráfico 6 – Cálculo da penalidade do processador realizado pelo regulador Ondemand em uma carga variante no tempo e uma taxa de amostragem de 0,1s.....	37
Gráfico 7 – Frequência X número de iterações com taxa de amostragem de 0,1s e carga de trabalho variante no tempo.....	38

RESUMO

O presente trabalho, intitulado Estudo e desenvolvimento de regulador para o gerenciamento de energia de uma unidade central de processamento, partiu de um estudo sobre os reguladores do módulo do Kernel Linux, CPUFreq, que gerenciam o gasto energético da(s) CPU(s). Visa propor a construção de um novo regulador para o gerenciamento da energia gasta pelo processador. Para tanto, o trabalho de LUIZ(2012) foi importante, pelas contribuições teóricas oferecidas. Assim, foram investigados os aspectos que mais influenciam em uma eficiente economia de energia de processadores. A taxa de amostragem, o cálculo da utilização e o cálculo da penalidade foram os três indicadores que possibilitaram, de maneira viável, o tratamento das informações coletadas e utilizadas na execução do nosso regulador, e consequentemente, uma satisfatória tomada de decisão no gerenciamento de energia da CPU. A partir dos três indicadores citados, foi feito um comparativo entre o nosso regulador e o regulador Ondemand, sendo este último o regulador padrão no Kernel Linux a partir da versão 2.9. Na economia de energia o nosso regulador obteve uma eficiência melhor do que o Ondemand, tendo em vista que nas mudanças dos valores da frequência de operação da(s) CPU(s), o nosso regulador manteve-se constante para o aumento e diminuição desses valores. Já o Ondemand, teve uma eficiência equivalente ao nosso regulador apenas no aumento dos valores da frequência, mas na diminuição, gastou mais tempo para reduzir a frequência, consequentemente gastando mais energia do que o nosso regulador.

Palavras-chave: Gerenciamento de energia; CPUFreq; Kernel Linux.

ABSTRACT

The aim of this paper, entitled ‘Study and ruling development for energy management of a processing central unit’, deals with a study about the regulators of the Kernel Linux modules–CPUFreq, which manage the energy waste of the CPU. Its aim is to propose the building of a new regulator for the management of the energy wasted by the processor. For this purpose, Luiz’s work (2012) played an important role due its theoretic contributions. In this way, we investigated the most influencing aspects for turning the processors economically efficient. The sample tax, the usage and the penalty calculation work as three indices that met the treatment of collected and managed information in the execution of our regulator and, therefore, met a satisfactory decision-making in energy management of the CPU. With the support of the three indices, we made a comparison between our regulator and the Ondemand’s regulator, observing that the latter is the standard regulator in Kernel Linux in its 2.9 version. In the item energy economy our regulator resulted more efficient than the Ondemand’s one, seen that in the changes of frequency values’ of the CPU operation, our regulator stayed stable at the increase and the decrease of these values. Ondemand’s one, by its turn, proved to be as efficient as our regulator only in what concerned the increase of frequency values, but at the decrease, it spent more energy than ours.

Key words: Energy Management; CPUFreq; Kernel Linux.

Sumário

1	INTRODUÇÃO.....	14
2	OBJETIVOS.....	17
2.1	OBJETIVO GERAL.....	17
2.2	OBJETIVOS ESPECÍFICOS.....	17
3	FUNDAMENTAÇÃO TEÓRICA.....	18
3.1	TECNOLOGIAS PARA GERENCIAMENTO DE PROCESSADORES.....	18
3.1.1	CPUFreq do Kernel Linux.....	19
3.1.2	Reguladores do CPUFreq.....	22
3.1.3	Regulador Conservative.....	22
3.1.4	Regulador Ondemand.....	23
3.1.5	Regulador Powersave.....	23
3.1.6	Regulador Performance.....	23
3.1.7	Regulador Userspace.....	24
3.1.8	Intel Turbo Boost 2.0.....	24
3.1.9	AMD Cool'n Quiet.....	25
4	MATERIAL E MÉTODOS.....	26
4.1	MATERIAL.....	26
4.2	MÉTODO.....	27
5	RESULTADOS E DISCUSSÕES.....	32
5.1	NOVO REGULADOR.....	34
6	CONSIDERAÇÕES FINAIS.....	39
	REFERÊNCIAS.....	41

1 INTRODUÇÃO

A eficiência energética da Unidade Central de Processamento (UCP ou processador) nos diversos tipos de sistemas computacionais – dispositivos móveis, computadores embarcados, desktops, servidores e clusters – desempenha um papel importante no tempo e na forma de execução da carga de trabalho a ser processada pela UCP, bem como no consumo energético total do sistema computacional.

Para os dispositivos móveis e computadores embarcados alimentados por baterias, a eficiência energética da UCP torna-se uma característica determinante na duração da carga, e, consequentemente, no tempo de utilização do dispositivo pelo usuário.

A busca pela diminuição do gasto de energia é um fator muito importante para a manutenção e economia dos recursos naturais existentes que são utilizados na fabricação e utilização das tecnologias de informação e comunicações (TICs). Após os principais fabricantes de processadores do mercado, Intel e AMD, lançarem diversos modelos de processadores com frequências de operações cada vez maiores, chegou-se ao limite físico de espaço em que as altas temperaturas geradas pelos processadores não podiam mais ser dissipadas pelos meios convencionais, e isso gerava altas temperaturas nos demais componentes de hardware. Assim, uma solução criada pelos fabricantes foi a de construir processadores que não precisassem de altas taxas de frequência de operação, e em vez disso, taxas mais equilibradas e um número maior de núcleos, ou seja, processadores multinúcleos. (ALMEIDA, 2009).

O uso de multiprocessadores aumentou de forma significativa o poder de processamento, mas o gasto de energia continuou sendo uma inquietação constante para os engenheiros. Com o advento dos dispositivos móveis, maximizou-se a preocupação de o hardware de computadores gastar menos energia, tendo em vista que o hardware de dispositivos móveis é bastante minimalista quando comparado a computadores maiores como servidores e clusters.

O Gerenciamento Dinâmico de Energia (GDE) tem como objetivo reduzir o consumo de energia em tempo de execução. Diferentes abordagens têm sido propostas para seletivamente desligar/paralisar componentes de hardware ociosos. Para os processadores, os fabricantes têm desenvolvido várias técnicas para o gerenciamento de energia, permitindo que o processador assuma estados de baixo consumo ou que seja possível o escalonamento

dinâmico de tensão e frequência do mesmo.

Para regulação dos estados de baixo consumo, uma interface avançada para configuração de energia (Advanced Configuration and Power Interface – ACPI¹) é utilizada (ACPI, 2013). A ACPI contém padrões para que o sistema operacional realize configuração, gerenciamento de energia e gerenciamento térmico dos dispositivos. Tecnologias como o Turbo Boost 2.0 (INTEL, 2015), criada pela Intel e utilizada nos seus processadores, e o AMD Cool'n Quiet (AMD, 2004), criada pela AMD, são exemplos dessa forma de gerenciamento.

Com base nas interfaces ACPI exclusivas de cada processador, os fabricantes desenvolveram métodos para o escalonamento dinâmico de tensão e frequência (EDTF) independente do processador, permitindo ajustar a frequência de operação das UCPs em nível de usuário, diminuindo a corrente/tensão dos processadores e, consequentemente, o consumo energético. O CPUFreq utilizado no Kernel Linux é um exemplo dessa forma de gerenciamento. (ALMEIDA, 2009).

Dada a problemática levantada sobre o gasto de energia da(s) UCP(s) pelo processamento de informações, o presente trabalho objetiva estudar o escalonamento dinâmico de tensão e frequência de processadores através do CPUFreq, apresentando seu funcionamento em diferentes cenários de utilização, e demonstrar a implementação de um novo regulador do CPUFreq para o gerenciamento da energia gasta pela(s) CPU(s).

O conteúdo deste trabalho está dividido em 6 capítulos, conforme detalhamento a seguir:

Capítulo 1 – Na introdução foram apresentadas a contextualização e a motivação do trabalho.

Capítulo 2 – Neste capítulo, apresentamos o principal objetivo deste trabalho, o qual visa a construir um novo regulador para o gerenciamento da energia gasta pela(s) CPU(s), e os passos que serão executados para tal.

Capítulo 3 – Neste capítulo, evidenciamos a preocupação com o gerenciamento do gasto de energia por parte dos processadores, assim como as soluções desenvolvidas pelos principais fabricantes de processadores e a solução criada pela comunidade de software livre.

Capítulo 4 – Neste capítulo, são mostrados os materiais utilizados para a construção do regulador proposto. As configurações realizadas no sistema operacional, as

¹ A ACPI é uma especificação industrial para a realização, por meio do sistema operacional, de gerenciamento de energia e gerenciamento térmico de dispositivos móveis, desktops e servidores.

fórmulas utilizadas nos cálculos, as configuração do computador utilizado neste trabalho, os arquivos contidos nos diretórios /proc e /sys do Kernel que foram acessados para a coleta dos dados necessários, e a explicação do algoritmo do regulador proposto.

Capítulo 5 – Neste capítulo, são mostrados, através de gráficos e discussões, quais resultados foram obtidos com a execução do novo regulador em comparação com o regulador Ondemand, a eficiência alcançada por cada um deles, a influência da taxa de amostragem adotada, a influência do valor de frequência de operação da(s) CPU(s), e em que aspectos o regulador proposto obteve melhorias no gerenciamento energético do processador.

Capítulo 6 – Neste capítulo, é mostrado o que se comprovou com a realização dos estudos e a construção do novo regulador para o gerenciamento de energia da(s) CPU(s), como esse novo regulador atendeu as expectativas da problemática levantada na motivação do trabalho, bem como a contribuição teórica para novas pesquisas relacionadas ao tema, discutindo e refletindo sobre o que já está posto e promovendo embasamento para novas tomadas de decisões.

2 OBJETIVOS

Neste capítulo, apresentamos os objetivos deste trabalho, e os passos que serão executados para tal.

2.1 OBJETIVO GERAL

Desenvolver um regulador do CPUFreq para o gerenciamento da energia gasta pela(s) CPU(s).

2.2 OBJETIVOS ESPECÍFICOS

- Realizar uma revisão bibliográfica referente ao subsistema CPUFreq;
- Realizar a coleta de dados gerados a partir dos arquivos de configuração existentes no módulo do Kernel, CPUFreq, organizá-los em gráficos e analisá-los, para a posterior construção do novo regulador proposto neste trabalho;
- Construir um novo regulador para o gerenciamento de energia da(s) CPU(s), baseado na taxa de utilização do processador;
- Elaborar ambientes de testes com cargas de trabalho leve, moderada e intensa, para medir a eficiência do gerenciamento de energia do processador com o regulador Userspace do CPUFreq, em frequências estáticas para avaliação da utilização do processador;
- Executar o novo regulador nos ambientes de teste. Ambientes elaborados para avaliar a utilização do processador e a forma como o novo regulador tomará as decisões de mudança de frequência para a possível economia de energia gasta pela(s) CPU(s).

3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, evidenciamos a preocupação com o gerenciamento do gasto de energia por parte dos processadores, assim como, as soluções desenvolvidas pelos principais fabricantes de processadores e a solução criada pela comunidade de software livre.

A partir de 1996, os computadores desktops passaram a utilizar uma especificação de interface de monitoramento de energia e configuração de dispositivos, o Advanced Configuration and Power Interface – ACPI, tecnologia codesenvolvida pelos fabricantes: Hewlett-Packard, Intel, Microsoft, Phoenix e Toshiba. O ACPI permite a configuração do monitoramento de energia via sistema operacional, o que permite uma configuração mais avançada. São objetivos do ACPI: facilitar e implementar funções de monitoramento de energia; melhorar o monitoramento de energia, bem como sua robustez; e, criar uma interface robusta para a configuração de dispositivos.

Com a disponibilidade do ACPI, outras tecnologias e drivers foram desenvolvidos pelos fabricantes de computadores e a comunidade, como, por exemplo, o CPUFreq, que é o módulo ou subsistema do Kernel Linux responsável pelo gerenciamento de energia gasta pela(s) CPU(s), implementada a partir do Kernel 2.6; o Turbo Boost Intel, que é a tecnologia utilizada pela Intel nos seus processadores para o gerenciamento da eficiência de processamento e energia da(s) CPU(s); e o AMD cool'n Quiet, já mencionados anteriormente.

3.1 TECNOLOGIAS PARA GERENCIAMENTO DE PROCESSADORES

O consumo de energia nos microprocessadores depende diretamente da voltagem do núcleo e da frequência de operação. A voltagem do núcleo, por sua vez, também depende diretamente da frequência de operação, dessa forma, para o microprocessador operar em uma baixa voltagem, também é necessário reduzir sua frequência. Baseada nessa relação entre frequência de operação, voltagem do núcleo e consumo de energia, grande parte dos processadores atuais possui mecanismos para poupar energia mediante a mudança de frequência e de voltagem em tempo de execução. As técnicas de mudança de frequência e de voltagem são conhecidas, respectivamente, como escalonamento dinâmico de frequência e de

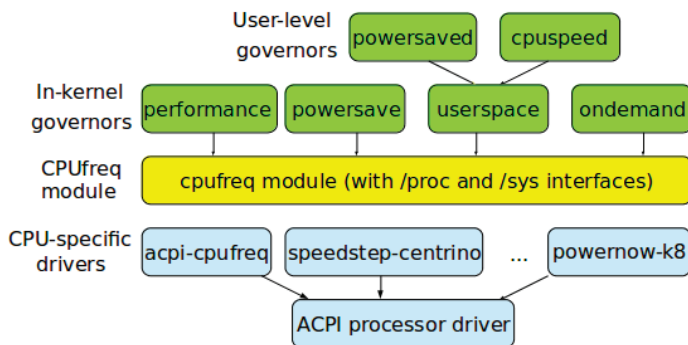
voltagem, e foram inicialmente aplicadas em dispositivos móveis para estender a duração de bateria sem comprometer o desempenho. (VIEIRA NETO, 2011).

A tecnologia ACPI permitiu que iniciativas eficientes pudessem ser desenvolvidas e, neste trabalho, daremos um enfoque maior ao CPUFreq, pois é uma tecnologia de código aberto, o que possibilitou o acesso às informações de projeto e implementação, bem como aos seus arquivos de configuração existentes no módulo do Kernel Linux. As outras duas tecnologias citadas, Turbo Boost Intel 2.0 e AMD Cool'n Quiet, são de empresas proprietárias, e isso impossibilitou o acesso às informações de projeto e implementação de maneira aprofundada.

3.1.1 CPUFreq do Kernel Linux

Uma das tecnologias existentes que trata do gerenciamento de energia gasta pelo processador é o CPUFreq. O CPUFreq é um subsistema ou módulo do Kernel Linux que, baseado nos mecanismos de escalonamento citados acima, permite melhorar a eficiência energética do sistema por meio da regulagem, em tempo de execução, da frequência e voltagem do processador. Esse subsistema é composto por três tipos de elementos – reguladores de frequência (verde), um núcleo (amarelo) e drivers de dispositivos (azul) – os quais encontram-se detalhados na figura a seguir:

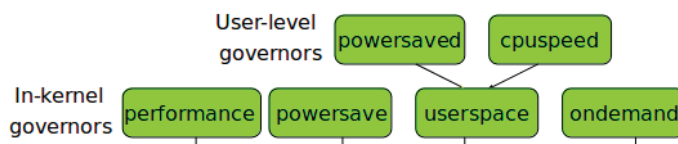
Figura 1 – Elementos do subsistema CPUFreq



Fonte: PALLIPADI, V.; STARIKOVSKIY, A. (2006, p. 223-238)

Os reguladores de frequência são algoritmos carregados no subsistema ou módulo do Kernel CPUFreq, responsáveis pelas implementações de diferentes políticas de controle de frequência do processador. A figura a seguir mostra os principais reguladores do CPUFreq:

Figura 2 – Reguladores de frequência do CPUFreq



Fonte: Adaptada pelo autor a partir de PALLIPADI, V.; STARIKOVSKIY, A. (2006, p. 223-238)

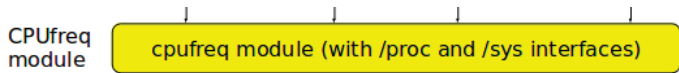
De forma geral, esses reguladores podem ser classificados em dois tipos: estáticos e dinâmicos. Reguladores estáticos definem apenas uma política geral de controle de frequência, privilegiando, por exemplo, desempenho ou economia de energia. Os reguladores “*Performance*”, “*Powersave*” e “*Userspace*”, que respectivamente, mantêm o processador na maior e menor frequência disponíveis, e em um valor mínimo (*powersaved*) ou máximo (*cpuspeed*) definido pelo usuário, são exemplos de implementações de reguladores estáticos embutidos no CPUFreq.

Reguladores dinâmicos, por sua vez, baseiam sua política de controle de frequência na avaliação periódica de algum indicador – tipicamente a utilização do processador – determinando, após cada avaliação, a frequência adequada de operação. Um exemplo de regulador dinâmico embutido no CPUFreq é o “*Ondemand*”, que para balancear desempenho e eficiência energética, aumenta a frequência de operação do processador quando sua taxa de utilização é alta, diminuindo essa frequência, para poupar energia, quando uma carga leve é observada. (PALLIPADI, 2006).

Apenas um regulador de frequência pode estar ativo no sistema em um dado momento. Um regulador é definido durante a inicialização do sistema, porém, um outro regulador pode ser selecionado em tempo de execução.

O núcleo do CPUFreq é responsável por intermediar a comunicação entre os reguladores e os drivers por meio dos subsistemas /proc e /sys. Ambos, reguladores e drivers, devem registrar-se no núcleo e oferecer um conjunto predefinido de interfaces existentes nesses subsistemas. Através dessas interfaces, um regulador pode, por exemplo, solicitar ao driver que ele informe as frequências de operação disponíveis ou que o microprocessador passe a operar em determinada frequência. O núcleo do CPUFreq também implementa, via sysfs², uma interface que permite alterar, em tempo de execução, a seleção do regulador de frequência ativo. A figura a seguir ilustra a estrutura do núcleo:

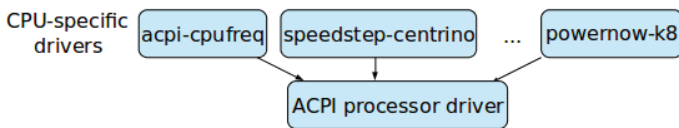
Figura 3 – Núcleo do CPUFreq



Fonte: Adaptada pelo autor a partir de PALLIPADI, V.; STARIKOVSKIY, A. (2006, p. 223-238)

Os drivers de dispositivos implementam os mecanismos de baixo nível necessários para o controle de frequência e de voltagem da(s) CPU(s). Esses mecanismos são específicos por plataformas. Cada tecnologia diferente precisa de um driver específico. Como a voltagem do núcleo é dependente da frequência de operação, o controle da voltagem do núcleo é realizado pelo driver, de acordo com a mudança de frequência.

Figura 4 – Drivers de dispositivos



Fonte: Adaptada pelo autor a partir de PALLIPADI, V.; STARIKOVSKIY, A. (2006, p. 223-238)

Além das políticas implementadas por seus reguladores embutidos, o CPUFreq pode incorporar dinamicamente outras opções de controle de frequência, através da instalação de novos reguladores implementados como módulos do Kernel. O CPUFreq permite também a incorporação de reguladores executados no espaço de usuário, disponibilizando, para isso, um

² Sysfs é um sistema de arquivos que fornece um meio para exportar as estruturas de dados do Kernel, seus atributos e as ligações entre eles para o ambiente do usuário. (MOCHEL; MURPHY, 2011).

regulador embutido especial (*Userspace*) e um conjunto de interfaces via sysfs. Os reguladores executados no espaço de usuário apresentam a vantagem de serem mais fáceis de ser implementados do que os reguladores embutidos no Kernel, entretanto, eles exigem troca de contexto a cada execução, o que acarreta em um maior dispêndio de recursos. Os reguladores embutidos no Kernel, por sua vez, podem se beneficiar das próprias interfaces e estruturas de dados internas do Kernel, além de não sofrerem dos dispêndios de recursos vinculados a trocas de contexto.

3.1.2 Reguladores do CPUFreq

Como dito anteriormente, um regulador é um algoritmo com definições pré-configuradas de frequência para a CPU. Esses reguladores têm de ser carregados como módulos do Kernel para serem acessados por outros programas no sistema. É possível carregar vários reguladores como módulos do Kernel, entretanto, não pode haver mais de um regulador tratando da frequência da CPU. Os reguladores disponíveis atualmente são: (I) Performance: configura a(s) CPU(s) para a frequência máxima; (II) Ondemand: Aumenta ou diminui, de forma dinâmica, a frequência da(s) CPU(s) baseado na necessidade do sistema; (III) Conservative: semelhante ao “*Ondemand*”, contudo mais conservador, configura a frequência da(s) CPU(s) de forma não repentina e baseia-se em médias de utilização para alterar gradativamente a frequência; (IV) Powersave: oposto ao regulador “*Performance*”, pois configura a(s) CPU(s) para a frequência mínima; (V) Userspace: esse regulador possibilita ajustar a frequência de operação da(s) CPU(s) para um valor estático definido pelo usuário.

3.1.3 Regulador Conservative

O Conservative é um regulador que altera a frequência da CPU de forma gradual e não diretamente para seu valor superior limite. Se houver uma atividade constante da CPU, esse regulador irá aumentar a frequência em saltos pequenos, como, por exemplo, de 1.0 GHz para 1.1GHz, 1.2GHz, 1.3 GHz e assim sucessivamente. Dessa forma, a economia de energia com esse regulador em operação é mais eficiente do que com o Ondemand. (BRODOWSKI, 2014).

3.1.4 Regulador Ondemand

O Ondemand é um regulador que vem por padrão no Kernel Linux, a partir da versão 2.6.9. (PALLIPADI; STARIKOVSKIY, 2006). Ele altera a frequência da CPU para o limite máximo quando a atividade da CPU é alta, em média maior ou igual a 95%, e diminui gradativamente quando essa atividade é minimizada. Esse é um regulador que possui certa eficiência por aumentar a frequência da CPU apenas quando sua utilização é alta, entretanto, ele não possui uma economia de energia satisfatória pelo fato de configurar rapidamente a frequência para seu valor máximo, gastando mais energia. (PALLIPADI, 2006; BRODOWSKI, 2014).

3.1.5 Regulador Powersave

O Powersave é um regulador que ajusta a frequência da(s) CPU(s) para o valor mais baixo dentro do intervalo compreendido entre a frequência máxima e a frequência mínima. É o regulador que possui o maior ganho de economia de energia, tendo em vista que trabalha apenas na frequência mais baixa de operação da(s) CPU(s). Entretanto, isso ocorre com um custo, a perda de eficiência do processador, já que este não poderá trabalhar sobre frequências maiores que a menor possível daquela(s) CPU(s). (BRODOWSKI, 2014).

3.1.6 Regulador Performance

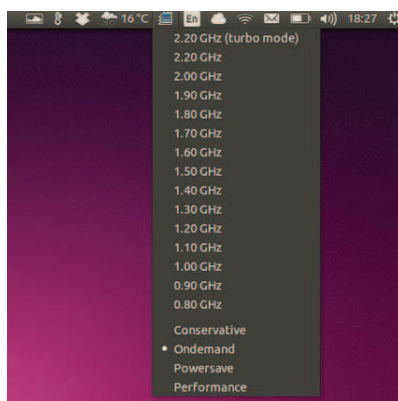
O Performance é um regulador que ajusta a frequência de operação da(s) CPU(s) para o valor mais alto dentro do intervalo compreendido entre a frequência máxima e a frequência mínima. Dessa forma, ocorre o maior ganho de eficiência da(s) CPU(s), contudo, não ocorre economia de energia devido a(s) CPU(s) trabalhar(em) na maior frequência possível para aquela arquitetura. (BRODOWSKI, 2014).

3.1.7 Regulador Userspace

Esse regulador pertence à categoria dos reguladores estáticos existentes no Kernel Linux, tendo em vista que só é possível configurar um único valor de frequência de operação para o processamento da(s) CPU(s).

O Userspace é um regulador que possibilita ajustar a frequência de operação da(s) CPU(s) para um valor estático, de acordo com os valores disponíveis para a CPU. Se a alteração for realizada via linha de comando, o usuário precisa ter privilégios de root³ para tal, e caso seja feita a alteração via interface gráfica, o aplicativo cpufreq-indicator⁴ pode ser utilizado. A figura a seguir mostra o indicador:

Figura 5 – Aplicativo indicator-cpufreq para alteração de frequência



Fonte: <http://inversotech.com.br/wp-content/uploads/2014/01/indicator-cpufreq.png>

3.1.8 Intel Turbo Boost 2.0

O Turbo Boost é uma tecnologia implementada pela Intel em certas versões de suas CPU(s) que permite automaticamente que os núcleos do processador trabalhem mais rápido do que trabalhariam na frequência padrão de operação quando estiverem operando abaixo dos

³ Em sistemas GNU/Linux, o usuário root possui todos os privilégios possíveis para a administração do sistema operacional.

⁴ O cpufreq-indicator é um aplicativo para a área de trabalho do sistema GNU/Linux que serve para escolher de forma intuitiva o regulador que irá operar no gerenciamento de energia do processador, bem como escolher frequências estáticas para a operação da(s) CPU(s).

limites especificados para a energia, a corrente e a temperatura. Quando o processador não está no seu limite térmico e elétrico e ocorre a demanda por parte da carga de trabalho em execução, o aumento da voltagem e frequência dos núcleos pode ser incrementado de acordo com a microarquitetura implementada neles. Se for a Nehalem, seu incremento será de 133MHz por vez, se for baseada na tecnologia Sandy Bridge, o incremento será de 100MHz por vez. A tecnologia é ativada quando o sistema operacional solicita uma frequência superior a padrão do processador ou a frequência atual em que estiver operando, desde que esta não seja maior que a frequência de limite superior. (INTEL, 2014).

A frequência máxima do Turbo Boost 2.0 é a mais elevada alcançada pelo processador quando as condições permitem, dependendo da carga de trabalho, do hardware, do software e das configurações do sistema. O aproveitamento da disponibilidade e da frequência do estado da tecnologia Turbo Boost 2.0 depende de vários fatores como: (I) Tipo de carga de trabalho; (II) Número de núcleos ativos; (III) Consumo estimado de corrente; (IV) Consumo estimado de energia; (V) Temperatura do processador.

Quando o processador estiver operando abaixo desses limites e a carga de trabalho do usuário exigir desempenho adicional, a frequência do processador aumentará dinamicamente até que o limite superior de frequência seja alcançado. (INTEL, 2014).

3.1.9 AMD Cool'n Quiet

A AMD Cool'n Quiet é uma tecnologia desenvolvida pela empresa AMD que possibilita a alteração da frequência de operação da(s) CPU(s) do computador enquanto este está em funcionamento. Semelhantemente as tecnologias CPUFreq e Intel Turbo Boost 2.0, a AMD Cool'n Quiet também diminui ou aumenta a taxa de frequência do processador, e para isso, leva em consideração o tempo de ociosidade em que o computador pode ser encontrado em um dado instante de tempo. Se houver uma ociosidade suficiente por parte da utilização do computador, o AMD Cool'n Quiet diminui a frequência da(s) CPU(s) para poupar energia e também diminui a velocidade da ventoinha para minimizar o barulho desta. Quando a taxa de utilização do processador aumenta, o AMD Cool'n Quiet aumenta a frequência de operação da(s) CPU(s), e consequentemente, a velocidade da ventoinha do processador. (AMD, 2015).

4 MATERIAL E MÉTODOS

Neste capítulo, são mostrados os materiais utilizados para a construção do regulador proposto. As configurações realizadas no sistema operacional, as fórmulas utilizadas nos cálculos, as configurações do computador utilizado neste trabalho, e como os arquivos do Kernel contidos nos diretórios /proc e /sys foram acessados para a coleta dos dados necessários, bem como a explicação do algoritmo do regulador proposto.

4.1 MATERIAL

A necessidade de um computador que estivesse sempre disponível para a realização dos experimentos, bem como para alterações no algoritmo do regulador e a constante análise dos resultados motivou o uso de um notebook pessoal ao invés de uma máquina de laboratório. Assim, os experimentos deste trabalho foram realizados em um Notebook Dell com as seguintes configurações:

- modelo Vostro 5470 A50;
- processador Intel® Core™ i7-4500U CPU @ 1.80GHz × 4;
- 8Gb de memória RAM;
- disco Rígido de 500Gb a 5400 rpm Sata;
- placa de vídeo NVIDIA GeForce GT 740M de 2Gb;
- o sistema operacional utilizado foi o GNU/Linux Ubuntu na versão 12.04 LTS, pois o seu Kernel permitiu o acesso ao regulador Userspace e as alterações de frequência em espaço de usuário. Na versão atual do GNU/Linux Ubuntu 14.04 LTS não foi possível esse tipo de configuração;
- o aplicativo cpufreq-utils, versão 0.07, pois é a versão disponível no GNU/Linux Ubuntu 12.04 LTS.

Os softwares aplicativos utilizados para gerar as cargas de trabalho leve, moderada e intensa, foram, respectivamente:

- a música on000001.mp3⁵, executada no tocador de músicas padrão do Ubuntu,

⁵ Disponível em <<http://www.dominiopublico.gov.br/download/som/on000001.mp3>>

- o Rhythmbox⁶;
- o vídeo Citizen Engineering⁷, volume 1, phones, executado no reprodutor de vídeo padrão do Ubuntu, o Totem⁸;
- e o aplicativo de medição de performance de disco rígido, Bonnie++⁹.

4.2 MÉTODOS

Após a realização do estudo sobre a documentação do CPUFreq do Kernel Linux, foi realizada a coleta dos dados estatísticos no arquivo */proc/stat*¹⁰, onde existem contadores, e o número de contadores depende da versão do Kernel, sendo 10 a quantidade existente na versão utilizada neste trabalho, que é o Kernel 3.13, cada um responsável por armazenar valores relacionados à informações diversas usadas pelo Kernel. As primeiras linhas desse arquivo, obtidas com o comando *cat /proc/stat*, são mostradas na tabela abaixo:

Tabela 1 – Dados estatísticos extraídos do arquivo */proc/stat*

Nome	user	nice	system	idle	iowait	irq	softirq	steal	guest	guest_nice
cpu	471562	4225	121050	7479385	49974	3	179	0	0	0
cpu0	123850	2123	37101	2416561	40973	2	133	0	0	0
cpu1	119394	1059	24714	1689428	2464	0	11	0	0	0
cpu2	118133	695	30416	1683314	4479	0	27	0	0	0
cpu3	110184	348	28817	1690081	2057	0	7	0	0	0

Fonte: Elaborado pelo autor

Conforme ilustrado na tabela acima, as quatro últimas linhas representam os valores de cada CPU existente no computador. A segunda linha, de nome *cpu*, representa a soma dos valores dos contadores de cada CPU mais 1. Cada coluna corresponde a um contador que é incrementado quando é realizada determinada atividade no processador:

- user*: corresponde ao número de processos executados no espaço de usuário;
- nice*: corresponde ao número de processos executados em espaço de usuário e com baixa prioridade;

⁶ Disponível em <<http://projects.gnome.org/rhythmbox/>>

⁷ Disponível em <<https://www.youtube.com/watch?v=-ikOgzOApIc>>

⁸ Disponível em <<http://www.gnome.org/projects/totem/>>

⁹ Disponível em <<http://sourceforge.net/projects/bonnie/>>

¹⁰ O */proc/stat* contém dados relacionados à utilização do processador. Esse arquivo é lido pelo gerenciador de energia, e a frequência escolhida para o processador operar é escrita na interface do CPUFreq.

- `system`: corresponde ao número de processos executados em espaço de núcleo;
- `idle`: corresponde ao valor incrementado quando o processador está inativo;
- `iowait`: corresponde ao número de espera de operações de entrada/saída que devem ser completadas;
- `irq`: é o número de serviços de interrupções;
- `softirq`: corresponde ao número de serviços de interrupções de software;
- `steal`: é o tempo utilizado na execução de outros sistemas operacionais num ambiente virtualizado;
- `guest`: tempo utilizado executando uma cpu virtual para outro sistema operacional sob controle do núcleo do sistema operacional Linux;
- `guest_nice`: corresponde à execução de um processo guest virtual remoto.

A coleta dos dados estatísticos dos contadores foi realizada conforme o comando `arquivo = fopen("/proc/stat", "r")`, que abre o arquivo para realizar a leitura dos dados, e `fscanf(arquivo,"%d", &<nome do parâmetro>)` que lê os dados e armazena na variável correspondente ao parâmetro.

Para medir a razão do tempo de inatividade do processador pelo período de amostragem, se fez necessário levar em conta que os contadores do arquivo `/proc/stat` são sempre incrementados. O cálculo dos incrementos dos contadores existentes no `/proc/stat` foi feito pela razão entre a diferença do valor atual e anterior da coluna `idle` $Idle(t) - Idle(t - 1)$ e a diferença entre a soma atual e a anterior dos contadores da linha `cpu` $Contador(t) - Contador(t - 1)$, em um instante de tempo t (LUIZ, 2012), conforme fórmula a seguir:

$$incremento(t) = \frac{\Delta Idle}{\Delta Contador} \quad (1)$$

Para o cálculo da utilização do processador, foi utilizada a fórmula abaixo, onde foi definida a utilização como sendo o complemento de $incremento(t)$ ilustrado na fórmula acima:

$$utilização(t) = 1 - incremento(t) \quad (2)$$

Para o tempo de amostragem usado no nosso regulador consideramos a política de

amostragem utilizada no regulador Ondemand. No regulador Ondemand, o tempo de amostragem é igual a $1000 * 10\mu s$, sendo este último parâmetro a latência de transição, ou seja, o intervalo de tempo que o Kernel verifica as alterações ocorridas na utilização do processador pelos processos. Esses valores são armazenados nos contadores que foram mostrados na tabela 1.

No regulador proposto, adotamos o valor de 10000 vezes a latência de transição, considerando que os dados estatísticos obtidos com esse valor de amostragem foram mais satisfatórios, tendo em vista que se for adotado um período de amostragem pequeno, o valor de saída calculado pela fórmula (1), acima citada, será pequeno. Entretanto, se o período de amostragem for muito grande, uma alta utilização do processador poderá ocorrer em curto intervalo, dentro do tempo de amostragem, mas não ser possível a tomada de ação por parte do regulador diante dessa rápida mudança na carga de trabalho, perspectiva que corrobora com o pensamento de Saulo (2012, p. 115), que aplicou uma política de amostragem semelhante em seu trabalho de doutoramento, sobre o gerenciamento de energia em processadores com cargas de trabalho variantes no tempo através de um microcontrolador, ao afirmar que

[...] não é interessante adotar um período de amostragem muito pequeno, pois a resolução da saída será pequena. E também não é interessante adotar um período de amostragem muito grande, pois, a utilização alta devido a uma carga em rajada durante um curto intervalo de tempo durante o período de amostragem pode ser filtrado, e não haverá reação no controlador a essa mudança na carga de trabalho.

Para a execução das cargas de trabalho, foi necessário desabilitar três dos quatro núcleos do processador utilizado, para que assim não houvesse paralelismo de processamento na execução das cargas de trabalho, e com o processamento em apenas um núcleo, os resultados fossem satisfatórios.

A desativação dos três núcleos foi realizada via arquivo de configuração no terminal de comandos do Linux, acessando o diretório `/sys/devices/system/cpu/`, onde existem os diretórios de cada núcleo do processador juntamente com seus arquivos de configuração. O comando para desativar um núcleo é o `echo 0 > online`, sendo o parâmetro `online` a variável que armazena dois possíveis valores: 1 para núcleo ativo e 0 para inativo.

As cargas de trabalho foram divididas em três tipos: carga leve, carga moderada e

carga intensa. Foram executadas de forma constante, ao contrário do que apontou Saulo (2012), sobre executar cargas de trabalho em rajada, pois, no nosso trabalho, foi priorizado executar o novo regulador em um ambiente de constante demanda de requisições de processamento. Todas as cargas de trabalho foram executadas durante o tempo de 1000 iterações do regulador proposto.

Na carga leve foi executada apenas a música de domínio público on000001.mp3, no reprodutor de músicas Rhythmbox, versão 2.96. Na carga moderada foi executado o vídeo de domínio público citizenengine-ervm.mov, no reprodutor de vídeos Totem, versão 3.0.1. E por fim, na carga de trabalho intensa foi executado o aplicativo de medição de performance de disco rígido, Bonnie++.

Os passos que o regulador proposto realizou foram: (I) fazer a leitura dos valores dos contadores no arquivo */proc/stat* e o cálculo da quantização através da fórmula (1); (II) efetuar o cálculo da utilização através da fórmula (2), segundo o critério adotado pela condição de que se a utilização do processador *utilização(t)* é igual a 1, ou seja, 100%, novas requisições não serão servidas imediatamente, assim, será calculado um valor de penalidade *penalidade(t)*, conforme está ilustrado na fórmula a seguir:

$$penalidade(t) = \begin{cases} 1, & \text{se } utilização(t) = 1 \\ 0, & \text{caso contrário} \end{cases} \quad (3)$$

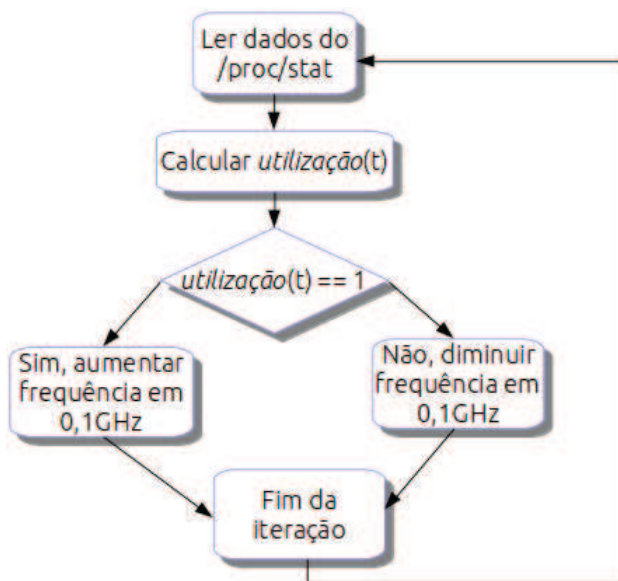
(III) se *utilização(t)* for igual a 1, aumentar a frequência de operação da(s) CPU(s) em 0,1GHz, caso contrário, diminuir a frequência de operação da(s) CPU(s) em 0,1GHz.

Assim, para efetivar o gerenciamento da energia gasta pela(s) CPU(s), o regulador que propomos leva em consideração o valor da taxa de utilização para configurar, em tempo de execução, o valor da frequência de operação da(s) CPU(s), buscando o valor mínimo sempre que possível.

Naturalmente, em momentos de alta requisição de processamento, será escolhida uma taxa de frequência suficiente para a não ocorrência de penalidade. Isso implicará em disponibilizar mais potência para o processador em detrimento a um gasto de energia maior.

O fluxograma abaixo ilustra o funcionamento do algoritmo do nosso regulador e o momento em que é feita a tomada de decisão baseada na taxa de utilização.

Figura 6 – Fluxograma do regulador proposto



Fonte: Elaborado pelo autor

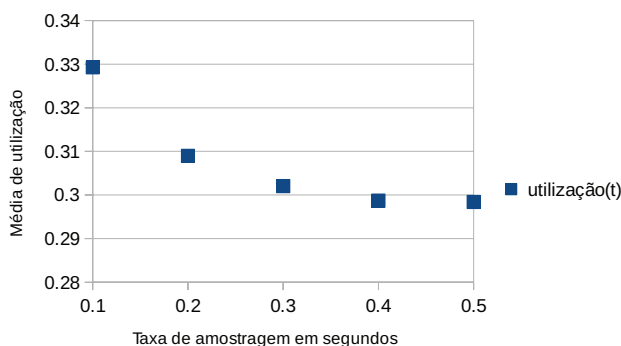
5 RESULTADOS E DISCUSSÕES

Neste capítulo, são mostrados, com gráficos e discussões, quais os resultados que foram obtidos com a execução do novo regulador em comparação com o regulador Ondemand, a eficiência alcançada por cada um deles, a influência da taxa de amostragem adotada, a influência do valor de frequência de operação da(s) CPU(s), e em que aspectos o regulador proposto obteve melhorias no gerenciamento energético do processador.

A taxa de amostragem afeta consideravelmente a forma como um regulador filtra as variações na taxa de utilização da(s) CPU(s). Os valores testados neste trabalho foram: 0,1s; 0,2s; 0,3s; 0,4s e 0,5s. Para o valor de 0,1s, foi detectada uma variação da utilização do processador mais acurada, durante a execução da carga de trabalho moderada.

A eficiência obtida no resultado anterior com a taxa de amostragem de 0,1s não se repete quando utilizamos valores maiores, pelos motivos discutidos anteriormente, ou seja, que se for adotado um período de amostragem pequeno, o valor de saída calculado, será pequeno, entretanto, se o período de amostragem for muito grande, uma alta utilização do processador poderá ocorrer em curto intervalo, dentro do tempo de amostragem, mas não ser filtrado e não ser possível a tomada de ação por parte do regulador diante dessa rápida mudança na carga de trabalho, conforme podemos verificar no gráfico 1, a seguir:

Gráfico 1 – Utilização X número de iterações para a frequência de 1,3GHz e taxa de amostragem variável

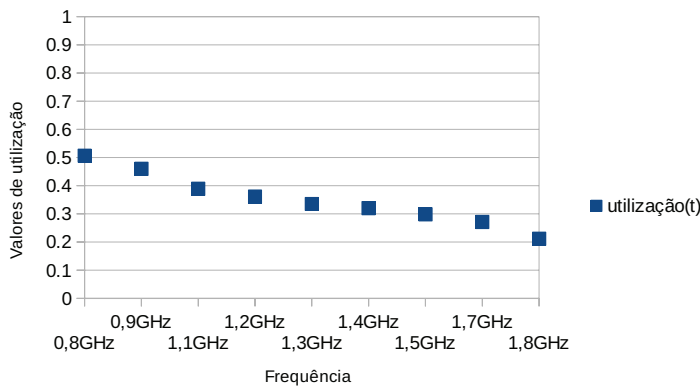


Fonte: Elaborado pelo autor

A comparação entre os valores para a taxa de amostragem, e a escolha de 0,1s, reforçam o que foi mencionado anteriormente sobre o intervalo de tempo para a amostragem, não sendo viável adotar um intervalo muito grande, nem muito pequeno, com o intuito de obter filtragens mais eficientes.

O valor da frequência de operação precisa ser sempre alterado para atender as demandas de processamento do sistema operacional. Quando o valor da frequência é alto, o processo em execução terminará tão logo quanto possível, entretanto, após seu término, o processador ficará algum tempo ocioso, operando em uma alta frequência até que a execução de outro processo seja solicitada pelo sistema operacional. Assim, é verídico que uma alteração de frequência dinâmica seja aplicada para uma economia de energia eficiente gasta pela CPU. Para representar com mais clareza a influência do valor da frequência de operação, observemos o gráfico 2, a seguir:

Gráfico 2 – Utilização X frequência variante e taxa de amostragem de 0,1s



Fonte: Elaborado pelo autor

5.1 NOVO REGULADOR

O regulador proposto leva em consideração o valor da taxa de utilização do processador para configurar, de forma dinâmica, o valor da frequência de operação da(s) CPU(s), buscando a frequência mínima sempre que possível. A tomada de decisão é feita a partir da verificação da utilização atual do processador. Se houver 100% de demanda de processamento, o valor da frequência de operação receberá um incremento de 0,1GHz a cada iteração, até que se chegue ao valor máximo de frequência de operação do processador, e se a demanda não for igual a 100%, será feito um decremento de 0,1GHz a cada iteração, até que se chegue ao valor mínimo de frequência de operação do processador. A figura a seguir, ilustra a parte do código onde é feita a tomada de decisão do nosso regulador:

Figura 7 – Estrutura de decisão do novo regulador

```
if(ys_t >= 1 && valor_frequencia < 1800000){
    valor_frequencia = (valor_frequencia + 100000);
    sprintf(comando, "cpufreq-set -f %d", valor_frequencia);
    system(comando);
}
else if(ys_t < 1 && valor_frequencia > 768000){
    valor_frequencia = (valor_frequencia - 100000);
    sprintf(comando, "cpufreq-set -f %d", valor_frequencia);
    system(comando);
}
```

Fonte: Elaborado pelo autor

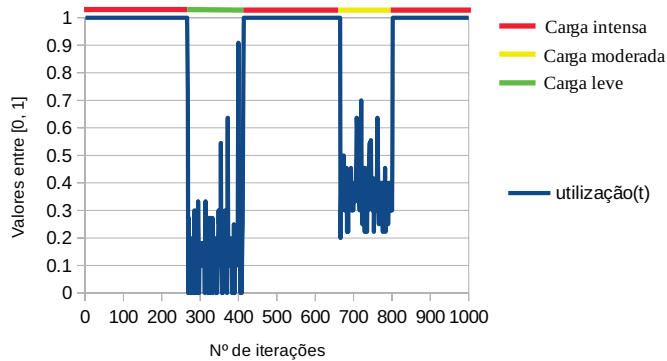
No regulador proposto, a frequência só é alterada para a taxa máxima se ocorrer uma demanda de 100% de processamento, pois não faz sentido manter a(s) CPU(s) gastando energia enquanto há uma constante ociosidade do processador, ou uma carga de trabalho leve está sendo executada.

O regulador Ondemand configura de forma um pouco diferente a política de frequência de operação do processador. Se houver uma utilização maior que 95%, ele já altera o valor da frequência para o máximo suportado pela(s) CPU(s), e isso não é feito de forma gradativa. A diminuição da frequência é que é feita de forma gradual.

Em uma carga de trabalho variante no tempo, começando com a carga intensa, seguida da carga leve, e terminando com a carga moderada, sendo as duas últimas intercaladas pela carga intensa, podemos verificar que a taxa de utilização varia bastante quando o

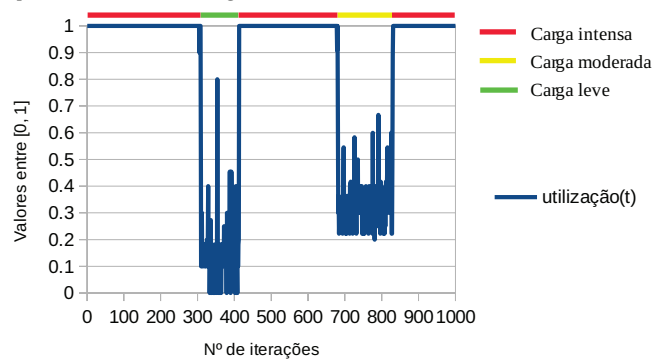
processamento também oscila, e isso não ocorre na carga intensa, porque o processador se encontra sob uma demanda altíssima de processamento. Analisando a forma como o novo regulador e o regulador Ondemand filtraram a utilização do processador no cenário descrito anteriormente, verificamos que o novo regulador filtrou de forma mais eficiente as variações da utilização da(s) CPU(s). Conforme observamos nos gráfico 3 e 4, a seguir:

Gráfico 3 – Cálculo da utilização do processador realizado pelo novo regulador em uma carga variante no tempo e uma taxa de amostragem de 0,1s



Fonte: Elaborado pelo autor

Gráfico 4 – Cálculo da utilização do processador realizado pelo regulador Ondemand em uma carga variante no tempo e uma taxa de amostragem de 0,1s

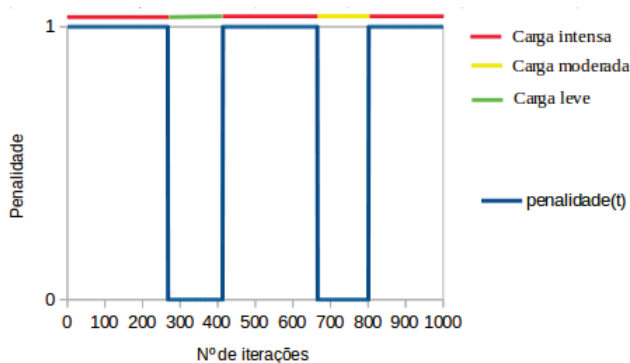


Fonte: Elaborado pelo autor

Com uma alta utilização do processador, resultante de uma carga intensa, a ocorrência de penalidade existirá em alguns momentos. Logo, novas demandas feitas pelo sistema operacional não serão servidas imediatamente, indicando que foi exigido 100% da(s) CPU(s).

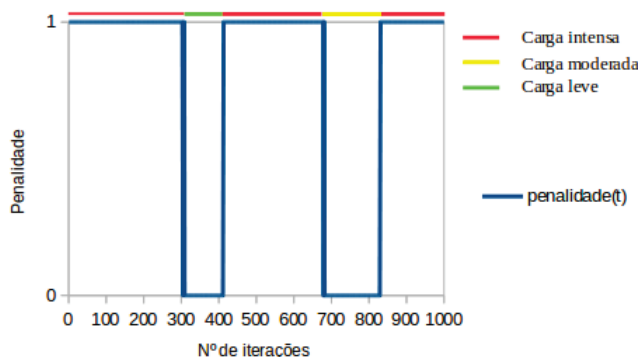
A política de gerenciamento de energia do nosso regulador faz com que a taxa de frequência volte gradativamente para o menor valor possível em intervalos de 0,1GHz. Entretanto, para chegar a essa frequência mínima, o regulador Ondemand gasta mais tempo. Os comportamentos descritos estão exemplificados nos gráficos 5 e 6, a seguir:

Gráfico 5 – Cálculo da penalidade do processador realizado pelo novo regulador em uma carga variante no tempo e uma taxa de amostragem de 0,1s



Fonte: Elaborado pelo autor

Gráfico 6 – Cálculo da penalidade do processador realizado pelo regulador Ondemand em uma carga variante no tempo e uma taxa de amostragem de 0,1s



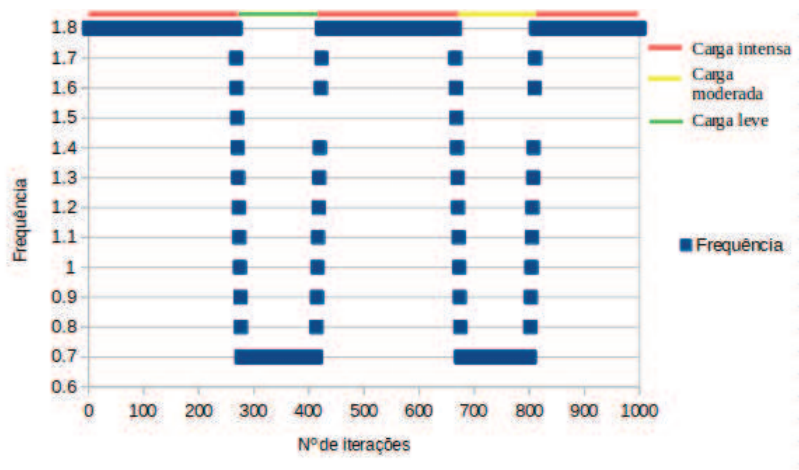
Fonte: Elaborado pelo autor

Quando o sistema operacional exigiu eficiência máxima para o processamento da carga de trabalho intensa, o regulador proposto alterou a frequência de operação da(s) CPU(s) para o valor máximo, 1,8GHz, e no decorrer do tempo esse valor foi reduzido quando a carga de trabalho leve entrou em execução, não demandando mais a máxima eficiência do processador.

Para a carga de trabalho leve, o nosso regulador manteve na maior parte do tempo o valor mínimo de frequência, e novamente, só houve mudança na frequência quando a carga de trabalho intensa passou novamente a ser requisitada pelo sistema operacional, e entrou em execução, fazendo com que o nosso regulador alterasse gradativamente em alguns instantes de tempo a frequência para o valor máximo. Em seguida, a carga moderada foi executada e a frequência voltou a ser reduzida, e voltou a subir o seu valor quando a carga intensa voltou a ser executada, terminando as iterações.

As alterações da frequência de operação da(s) CPU(s) feitas pelo novo regulador, podem ser verificadas no gráfico 7, a seguir:

Gráfico 7 – Frequência X número de iterações com taxa de amostragem de 0,1s e carga de trabalho variante no tempo



Fonte: Elaborado pelo autor

A taxa de amostragem, o cálculo da utilização e o cálculo da penalidade foram os três indicadores que possibilitaram de maneira viável, o tratamento das informações coletadas e utilizadas para a execução do nosso regulador, e consequentemente, uma satisfatória tomada de decisão no gerenciamento de energia, quando comparada nos mesmos aspectos com o regulador Ondemand, que obteve um gasto de tempo maior para reduzir a frequência do processador, e com isso, gastando mais energia.

Para altas demandas de processamento, tanto o nosso regulador quanto o Ondemand obtiveram resultados semelhantes, entretanto, na redução da frequência do processador, o nosso regulador obteve um melhor resultado, culminando em uma economia mais eficiente da energia gasta pelas CPU(s).

6 CONSIDERAÇÕES FINAIS

O propósito geral deste trabalho foi o de construir um novo regulador para o gerenciamento da energia gasta pela(s) CPU(s). Sua comparação com o regulador Ondemand, que é o regulador padrão nas atuais versões do Kernel Linux, trouxe bons aprendizados e mostrou que conseguimos cumprir a nossa proposta de construir um novo regulador para o gerenciamento energético de CPU(s).

Os estudos realizados pautaram-se em tecnologias para o gerenciamento de processadores, a saber: CPUFreq do kernel Linux; reguladores do CPUFreq – Conservative, Ondemand, Userspace, Powersave e Performance, incluindo, Intel Turbo Boost 2.0 e AMD Coll'n Quiet, de forma menos aprofundada, pelo fato de se tratar de tecnologias proprietárias, o que implica em não ser possível ter acesso à documentação, ao projeto e à implementação das tecnologias.

A partir da elaboração do novo regulador, viu-se que as soluções já existentes não atendem a todas as demandas nesse campo, e existe muito o que se investigar sobre como desenvolver melhores técnicas para o gerenciamento energético de processadores.

Os reguladores dinâmicos são aqueles que melhor se adaptam às variações de cargas de trabalho ao longo do tempo, assim, a construção do nosso regulador para gerenciar de forma dinâmica a alteração da frequência de operação da(s) CPU(s) para o melhor gerenciamento energético deste(s), correspondeu às expectativas iniciais de contribuir com uma nova solução para gerenciar o gasto energético em processadores.

A utilização de um regulador estático para gerenciar o gasto de energia de processadores é inviável quando trata-se de cargas de trabalho variantes no tempo, pois, inicialmente, não se sabe qual tipo de carga de trabalho o processador irá lidar, então, mantê-lo a todo o tempo operando em apenas uma frequência, acarretará em algum momento em falta de desempenho ou gasto de energia desnecessário.

Isto posto, utilizar um regulador dinâmico para o gerenciamento energético de processadores é a solução mais adequada, tendo em vista que o processamento de informações atualmente também é dinâmico e bastante variável. Assim, percebemos que o novo regulador proposto neste estudo conseguiu adaptar-se de forma satisfatória às variações ocorridas no processamento das cargas de trabalho variantes no tempo, tendo em vista que sua execução foi em nível de usuário e isso implica que os dados precisam ser lidos de arquivos no diretórios

/proc e /sys, diferentemente do regulador Ondemand que está em nível de Kernel e, dessa forma, tem acesso aos dados diretamente no Kernel não necessitando fazer leituras de arquivos dentro de diretórios.

A pesquisa trouxe um importante esclarecimento de como as empresas, representadas pela Intel e a AMD, e a comunidade Linux, representada pela tecnologia aberta CPUFreq, têm empregado esforços para minimizar o gasto de energia dos computadores e dispositivos existentes. Confirma, ainda, a viabilidade existente para a construção de um novo regulador para o gerenciamento de energia do processador e que ainda há o que melhorar nas propostas atuais. Além disso, espera-se que o trabalho possa contribuir para o surgimento de uma nova solução para o gerenciamento de energia gasta pelos processadores atuais, em nível do núcleo do sistema operacional Linux. Que seja também um estudo que possa contribuir teoricamente para novas pesquisas relacionadas ao tema.

REFERÊNCIAS

ACPI. Advanced Configuration & Power Interface. Disponível em: <<http://www.acpi.info/>>. Acesso em: 27 nov. 2015.

ALMEIDA, Rafael Bruno. Evolução dos Processadores: Comparação das famílias de processadores intel e AMD, 2009. Disponível em: <<http://www.ic.unicamp.br/~ducatte/mo401/1s2009/T2/089065-t2.pdf>>. Acesso em: 11 out. 2014.

AMD. Cool'n Quiet™: Technology Installation Guide for AMD Athlon™ 64 Processor Based Systems. 2004. Disponível em: <http://web.archive.org/web/20070409045621/http://www.amd.com/us-en/assets/content_type/DownloadableAssets/Cool_N_Quiet_Installation_Guide3.pdf>. Acesso em: 28 jan. 2015.

BRODOWSKI, Dominik. *Linux CPUFreq: USER GUIDE*. 2014. Disponível em: <<https://www.kernel.org/doc/Documentation/cpu-freq/user-guide.txt>>. Acesso em: 06 out. 2014.

_____. *Linux CPUFreq: CPUFreq Governors - information for users and developers*. 2014. Disponível em: <<https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt>>. Acesso em: 06 out. 2014.

_____. *Linux CPUFreq: CPU Drivers*. 2014. Disponível em: <<https://www.kernel.org/doc/Documentation/cpu-freq/cpu-drivers.txt>>. Acesso em: 06 out. 2014.

INTEL. Tecnologia Intel Turbo Boost 2.0: brave visão geral. 2014. Disponível em: <<http://www.intel.com.br/content/www/br/pt/architecture-and-technology/turbo-boost/turbo-boost-technology.html>>. Acesso em: 10 out. 2014.

LUIZ, Saulo Oliveira Dornellas. **Gerenciamento Dinâmico de Energia em Processadores com Cargas de Trabalho Variantes no Tempo**. 2012. 148 f. Tese (Doutorado) - Curso de Pós- Graduação em Engenharia Elétrica, Universidade Federal de Campina Grande, Campina Grande, 2012.

MOCHEL, Patrick; MURPHY, Mike. Sysfs: The filesystem for exporting kernel objects. 2011. Disponível em: <<https://www.kernel.org/doc/Documentation/filesystems/sysfs.txt>>. Acesso em: 28 jan. 2015.

PALLIPADI, Venkatesh. `Linuxcpufreq-statsdriver`: information for users. 2014. Disponível em: <<https://www.kernel.org/doc/Documentation/cpu-freq/CPUFreq-stats.txt>>. Acesso em: 06 out. 2014.

PALLIPADI, V.; STARIKOVSKIY, A. The ondemand governor. In: **Proceedings of the Linux Symposium**. [S.l.: s.n.], 2006. v. 2, p. 215–230. (document), 4.3.1, 4.3, 4.3.1, 4.3.1, 4.4.

VIEIRA NETO, Lourival. **Lunatik**: Scripting de Kernel de Sistema Operacional com Lua. 2011. 69 f. Dissertação (Mestrado) - Curso de Pós-graduação em Informática, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2011. Cap. 4.

**More
Books!**



yes
I want morebooks!

Buy your books fast and straightforward online - at one of the world's fastest growing online book stores! Environmentally sound due to Print-on-Demand technologies.

Buy your books online at
www.get-morebooks.com

Compre os seus livros mais rápido e diretamente na internet, em uma das livrarias on-line com o maior crescimento no mundo! Produção que protege o meio ambiente através das tecnologias de impressão sob demanda.

Compre os seus livros on-line em
www.morebooks.es

SIA OmniScriptum Publishing
Brīvības gatve 197
LV-103 9 Rīga, Latvia
Telefax: +371 68620455

info@omniscryptum.com
www.omniscryptum.com

OMNIScriptum



