



Universidade de São Paulo
Instituto de Matemática e Estatística
Departamento de Ciência da Computação

**MAC0219: Introdução à Programação Concorrente,
Paralela e Distribuída**

Consumo de Energia na Computação

Professor: Alfredo Goldman

Monitor: Pedro Bruel

Bárbara Fernandes - 7577351

Duílio Elias - 6799722

Eduardo Bier - 8536148

São Paulo - SP, 16 de junho de 2017

Sumário

1	Introdução	4
2	Consumo de energia	4
2.1	Tipos de fontes de energia	5
2.1.1	Renováveis vs. não-renováveis	5
2.1.2	Primárias vs. secundárias	6
2.1.3	Convencionais vs. alternativas	6
2.2	Geração elétrica e emissões	7
2.3	Consumo de energia pela computação	7
2.4	Green500	8
2.5	Dissipação de energia	8
2.6	Lei de Koomey e o Princípio de Landauer	9
3	Soluções em Hardware	10
3.1	Computação Conservadora	10
3.2	Computação Reversível	11
3.3	Computação Magnética	12
4	Soluções em Software	12
4.1	Algoritmos Reversíveis	12
4.1.1	Exemplo: Merge Sort Reversível	14
4.2	Virtualização	15
4.3	Computação em Nuvem	15
5	Conclusão	15

Lista de Figuras

1	Exemplo de uma porta XOR com dissipação máxima de energia.	9
2	Exemplo de um portão XOR usando <i>dual-rail logic</i>	11
3	Alguns dos algoritmos estudados desenvolvidos em [1] com seus respectivos custos de tempo, espaço e energia.	13

1. Introdução

Desde o início da computação, a eficiência de um programa de computador foi sempre medida usando principalmente duas características: sua velocidade e seu uso de memória. Assim, enquanto desenvolvedores de hardware focavam em aumentar a capacidade de processamento e a velocidade de transferência de dados dos dispositivos, os de software buscavam implementar algoritmos e estruturas de dados que fizessem um bom balanço entre o tempo de execução e o uso de memória dos programas.

Porém, nos últimos anos, um novo fator tem sido considerado ao se falar em eficiência na computação: o uso de energia. Há duas principais razões para essa crescente preocupação com a eficiência energética. Primeiramente, o uso difundido de aparelhos móveis, como celulares, *tablets* e *notebooks*, trouxe um importante estímulo para a indústria de tecnologia pesquisar e investir nessa área, uma vez que um dispositivo móvel perde grande parte de sua mobilidade se tiver que ser constantemente conectado a uma fonte de energia para carregar sua bateria. A segunda razão é de cunho ambiental e econômico. A produção de energia no mundo é algo que é caro e tem sempre algum tipo de impacto ambiental, seja ele mais evidente e em maior escala com a queima de combustíveis fósseis, ou mais discreto e em menor escala com o uso da energia solar. Assim, em um momento no qual existe uma crescente preocupação ambiental no mundo, somado ao alto custo do consumo desta energia, tanto para o consumidor quanto para as empresas, a eficiência energética dos computadores entra quase que naturalmente nas considerações de desempenho de um computador.

O objetivo desta monografia é explorar algumas soluções interessantes para a diminuição do uso de energia dos computadores, tratando tanto de soluções que já estão em prática quanto soluções que ainda estão em fase de desenvolvimento.

2. Consumo de energia

A preocupação com o consumo de energia se tornou uma questão ainda mais relevante nos últimos tempos por diversas razões, entre elas as questões econômicas e ambientais se destacam, principalmente pela grande necessidade da diminuição do uso de combustíveis fósseis e pelo grande aumento valor do petróleo. Com isso, a busca de alternativas para tornar mais eficiente o consumo de energia trouxe para a área dos desenvolvedores de *softwares* e *hardwares* novos questionamentos: como consumir o mínimo possível de energia nos diversos dispositivos eletrônicos e como fazer com que um determinado algoritmo consuma menos energia.

Para se ter uma ideia da importância destes questionamentos para o setor de telecomunicações, iremos apresentar algumas informações sobre como está distribuído o consumo de energia e, posteriormente, apresentar algumas soluções.

2.1. Tipos de fontes de energia

Podemos dividir os tipos de fontes de energia em renováveis e não renováveis; primárias e secundárias; convencionais e alternativas.

2.1.1. Renováveis vs. não-renováveis

Nossos recursos de **energias renováveis** nunca se esgotarão, já que seu fornecimento não é limitado. Também não há custos sobre o combustível e eles tipicamente geram muito menos poluição do que combustíveis fósseis.

Exemplos de fontes de energia renovável incluem:

- energia eólica;
- energia hidráulica, como usinas hidrelétricas e usinas maremotrizes;
- energia geotérmica;
- energia solar;
- energia de biomassa, como a biomassa arborícola (energia liberada pela madeira).

No entanto, há certas desvantagens na geração de energia renovável. Parques eólicos, por exemplo, são barulhentos e podem estragar a vista das pessoas que moram nas proximidades. A quantidade de eletricidade gerada depende da força do vento. Ademais, se não houver vento não haverá eletricidade.

Há um fornecimento limitado de recursos energéticos **não-renováveis**, os quais eventualmente se esgotarão. Exemplos deles incluem: combustíveis fósseis, como carvão, óleo e gás natural; combustíveis nucleares, como o urânio.

Combustíveis fósseis liberam dióxido de carbono quando queimam, o que contribui com o **efeito estufa** e agrava o **aquecimento global**. Dos três combustíveis fósseis mencionados, o carvão é o que produz a maior quantidade de dióxido de carbono para uma dada quantidade de energia liberada, enquanto o gás natural é o que produz menos.

O combustível para as estações de energia nuclear são relativamente baratos, mas as estações de energia em si são caras para se construir. Também é bastante caro o desmonte de estações de energia antigas e o armazenamento de detritos radioativos, que representam ameaças sérias à saúde.

2.1.2. Primárias vs. secundárias

A característica fundamental da **energia primária** é o seu processo de extração ou captura. Essa energia é extraída ou capturada de fontes, e através dessa extração ou captura, suas características físico-químicas não são alteradas. São exemplos o carvão, óleo, gás natural, madeira, combustíveis nucleares, o sol, o vento, as marés e os rios.

Já as **fontes secundárias** são caracterizadas pelos seus processos de transformação. Transformação de energia é qualquer processo de transformar uma forma de energia em outra. Energia de combustíveis fósseis, radiação solar ou combustíveis nucleares, que são todos primários, podem ser convertidos em outras formas de energia como eletricidade e calor, que podem ser mais úteis para nós. Toda energia que foi submetida a transformações humanas é energia secundária.

Para muitas necessidades energéticas, o uso da eletricidade é mais fácil que o de uma fonte de energia primária. Por exemplo, utilizar eletricidade para a iluminação é mais seguro do que utilizar petróleo em velas ou lâmpadas de querosene. A eletricidade também permite a utilização de dispositivos como televisões, computadores e telefones celulares, que não conseguem ser carregados diretamente por fontes de energia primárias.

2.1.3. Convencionais vs. alternativas

Fontes de energia **convencionais** são aquelas que vêm sendo utilizadas desde os tempos antigos. O uso de tecnologias convencionais para produzir energia elétrica normalmente resulta em poluição em grande escala. Elas constantemente dependem da queima de combustíveis fósseis que produzem gases nocivos e acabam na atmosfera. Pessoas e animais respiram o ar poluído e plantas absorvem a poluição. Grandes quantidades de energia elétrica podem ser produzidas através do uso de tecnologias convencionais. Carvão, gás natural, óleo e madeira de queima são exemplos de fontes de energia convencionais.

Energias alternativas são normalmente produzidas aproveitando-se do mundo natural ao redor de nós e dos elementos. Normalmente eles não produzem poluição na atmosfera ou, pelo menos, produzem somente pequenas quantidades. Atualmente, energias alternativas produzem somente pequenas quantidades de energia elétrica e muito investimento é necessário para aumentar a quantidade de energia produzida através desse jeito mais natural. Fontes de energia não-convencionais incluem: energia solar, energia hidroelétrica, energia eólica, energia geotérmica, biomassa e biocombustível.

2.2. Geração elétrica e emissões

Globalmente, o uso de eletricidade vem aumentando rapidamente à medida que novas grandes economias se desenvolvem em locais como a China e a Índia. Essa necessidade por eletricidade leva a uma demanda crescente pela geração desse recurso, com milhares de novas usinas de energia sendo necessárias nas próximas décadas ao redor do mundo.

Por muitas décadas quase toda a eletricidade consumida no mundo tem sido gerada através de três tipos de usinas de energia - fósseis, hidroelétricas e nucleares. Fontes renováveis atualmente geram uma porção pequena da eletricidade mundial, apesar de esta porção crescer rapidamente.

A demanda de eletricidade pode ser suprida de duas formas diferentes. O principal método até agora tem sido a construção de projetos centralizados públicos ou privados em grande escala para gerar e transmitir a eletricidade requerida para abastecer as economias.

Muitos desses projetos têm causado efeitos ambientais indesejáveis, como a poluição do ar, poluição radioativa ou o alagamento de grandes áreas de terra. A geração distribuída cria energia em pequena escala em pontos da rede de eletricidade.

Frequentemente esses pontos criam eletricidade como subproduto de outros processos industriais, como a utilização de gás proveniente de aterros sanitários para a movimentação de turbinas.

Emissões da geração de energia elétrica são uma função com dois fatores: a quantidade de eletricidade utilizada e a quantidade de dióxido de carbono liberada por essa eletricidade.

2.3. Consumo de energia pela computação

A energia consumida pelos computadores não está relacionado apenas ao funcionamento do computador em si, mas também com toda a infraestrutura relacionada a computação. Assim, um computador pessoal conectado à internet gasta muito mais energia do que parece, uma vez que deve-se levar em conta os enormes data centers, com seu grande número de servidores e equipamentos de rede e seu sistema de refrigeração, umidificação e iluminação do ambiente. O uso de energia dos data centers representam uma parte significativa do consumo mundial de energia. Em 2014, por exemplo, data centers americanos usaram 2% de toda a energia consumida nos EUA, o equivalente a 70 bilhões kWh [2]. Além disso, estima-se que em 2010 computadores pessoais e servidores tenham gastado, respectivamente, 1% e 1,5% de toda a energia consumido no mundo. Esses dados mostram como o consumo de energia na computação está extremamente alto, ainda mais considerando que coisas como a rede móvel não entraram nessas contas.

Apesar disso, os avanços tecnológicos dos últimos anos fizeram com que a curva de crescimento do consumo de energia tenha sido bastante suavizada. Estima-se que, caso esse crescimento tivesse permanecido constante entre 2010 e 2014, o consumo total de energia pelos *data centers* em 2014 teria sido mais de 55% maior do que foi [2]. Isso implicaria em um consumo extra de 40 bilhões de kWh.

Existem técnicas para se medir a eficiência de um data center. Uma dessas medidas é chamada PUE - power usage effectiveness - calculado da seguinte forma:

$$\text{PUE} = \frac{P_t}{P_s} \quad (1)$$

onde P_t e P_s representam, respectivamente, o consumo de energia total do data center e o consumo de energia dos equipamentos de TI. Em média, este valor é em torno de 1.7, mostrando que grande parte da energia desses datacenters é usada para manter os computadores em bom estado. Algumas empresas, como a Google, com uma grande quantidade de recursos conseguem valores bem inferiores, no entanto. Os data centers da Google, por exemplo, tem um PUE de 1.12, em média.

2.4. *Green500*

Criada em 2007 e atualizada semestralmente, a **Green500** é a lista dos supercomputadores mais eficientes em termos de consumo energético no mundo e representa a crescente preocupação com o uso de energia dos computadores. Atualmente, o topo da lista é ocupado por um computador da NVIDIA, com 9462.1 MFLOPS/W de eficiência.

Apesar de muitos dos computadores que entram nessa lista serem desenvolvidos especificamente para gastar menos energia, muitos deles acabam entrando nela pelo simples fato de terem um enorme poder computacional. É o caso do *Sunway MPP*, que ocupa a posição 4 da lista Green500 e posição 1 da TOP500, lista que classifica os 500 supercomputadores mais rápidos do mundo. Apesar de este computador ter um enorme consumo de energia, 15.371kW, seu poder computacional acaba compensando esse consumo, permitindo que ele fique uma posição acima do *PRIMERGY CX1640 M1*, computador da Fujitsu que gasta apenas 77kW.

2.5. *Dissipação de energia*

A dissipação de energia nos computadores ocorre principalmente de duas formas. A primeira delas é o "vazamento" da energia nos transistores, algo sobre o qual não se tem muito controle e é da natureza do material, em que uma baixa quantidade de energia está sempre passando pelos transistores

mesmo que teoricamente ele esteja desligado. A segunda, por sua vez, é resultado da forma com que operações são feitas nos computadores.

Nos computadores atuais, o valor dos *bits* são representados pela presença ou não de uma corrente elétrica. Assim, quando um par de *bits* passa por uma porta lógica e gera uma saída, nem sempre a quantidade de energia que entrou será a mesma que passa pela porta de saída. Esse problema é bastante claro no caso da porta XOR, onde se temos uma dessas portas com entrada de dois *bits*, cada um com valor igual a 1, teremos como saída um *bit* valendo 0, como mostra a figura 1. A energia que entrou nessa porta, equivalente a 2 *bits* ligados, foi completamente perdida no processo, uma vez que o *bit* de saída não a utiliza.



Figura 1: Exemplo de uma porta XOR com dissipação máxima de energia.

Em ambos os casos a dissipação de energia acontece como geração de calor, o que leva ao aumento da temperatura do computador. Isso, por sua vez, torna necessária sua refrigeração, o que resulta num gasto ainda maior de energia. Assim, um bom jeito de diminuir esse gasto energético é diminuir essa dissipação. Nas sessões seguintes, veremos alguns métodos para que isso aconteça.

2.6. Lei de Koomey e o Princípio de Landauer

A Lei de Koomey é a tendência descrita por Jonathan Koomey de a eficiência computacional dobrar aproximadamente a cada 18 meses e tem sido chamada de a "Lei de Moore da eficiência energética" devido à clara similaridade entre ambas. Koomey observa que desde os anos 1950 a quantidade de computações por *joule* tem dobrado a cada ano e meio, o que demonstra os grandes avanços que já sido feitos nessa área.

No entanto, a Lei de Koomey não será válida para sempre - pelo menos não com computadores irreversíveis. De acordo com o Princípio de Landauer, derivado da segunda lei da termodinâmica, existe um limite inferior para a energia dissipada pelos computadores irreversíveis, dado pela equação 2, onde k é a constante de Boltzmann, T a temperatura e E a energia dissipada.

$$E \geq kT \ln 2 \quad (2)$$

Dessa forma, estima-se que a Lei de Koomey irá atingir seu limite em um futuro relativamente próximo, tão cedo quanto 2048. Vale notar que

esse limite só é válido para computadores irreversíveis e, portanto, como será discutido na seção 3, outras formas de computação poderão fazer com que a Lei de Koomey não seja quebrada tão cedo.

3. Soluções em Hardware

As soluções em hardware para a diminuição do consumo de energia levam em conta uma série de fatores para a sua implementação. As três soluções apresentadas nesta seção, de uma forma geral, procuram diminuir a dissipação de energia na forma de calor. Isso faz com que não só menos energia seja gasta inutilmente, mas também com que a quantidade de calor gerada pelo computador diminua, o que por sua vez diminui a demanda por refrigeração, diminuindo ainda mais a energia necessária para manter o computador rodando.

3.1. Computação Conservadora

Como visto na Seção 2.5, um portão XOR acaba desperdiçando energia ao ter que zerar o valor do bit de saída quando a entrada é (1, 1). A computação conservadora se propõe a resolver esse problema reutilizando essa energia ao em vez de dissipá-la como calor, o que significa que a quantidade de 1s da entrada de um portão lógico deve ser igual à da saída. Isso faz com que a energia da entrada de um portão lógico não seja perdida, podendo ser reutilizada. Mas como o hardware pode ser modificado para fazer isso?

Uma possibilidade é usar *dual-rail logic*, onde cada bit é representado por um par de valores ao invés de só um, sendo apenas o primeiro dos valores usado para codificar o valor real do bit. A Tabela 1 mostra os valores reais de um bit representados pelos diferentes pares de valores possíveis. Com esse sistema de codificação de bits é possível adicionar uma saída nas portas lógicas, onde os 1s extras, isto é, aqueles que não são necessários para codificar a saída, são "jogados fora". Esses bits extras podem então ser reciclados, sendo levados à diante para outras portas até que sejam finalmente usados quando o computador precisar daqueles volts que estão sobrando.

A Figura 2 mostra o esquema geral destes portões lógicos usando dois trilhos. Note que a quantidade de 1s que entra no portão é exatamente igual ao número de 1s que sai. Isto significa que a energia não foi dissipada da mesma forma que aconteceria em um computador normal, cortando a quantidade de energia dissipada e calor gerado pela metade. Além disso, este modelo aumentaria também a segurança dos computadores, protegendo-os de ataques que contem com a variação de voltagem no computador para fazer engenharia reversa e descobrir o que está lá.

x_1	x_2	x
0	0	0
0	1	0
1	0	1
1	1	1

Tabela 1: Valor real x do bit representado pelo par (x_1, x_2) usando o sistema *dual-rail logic*.

É importante notar, no entanto, que a quantidade de fios e transistores necessários para implementar isso cresceria consideravelmente, que por sua vez faria com que mais energia fosse dissipada por vazamentos dos transistores e mais calor fosse gerado pelo chip. Isso certamente diminuiria parte dos ganhos energéticos usados nessa implementação, já que esses chips precisariam de uma maior refrigeração. Ainda assim, a computação conservadora se mostra uma importante opção para o desenvolvimento de hardware com o objetivo de aumentar a eficiência energética.

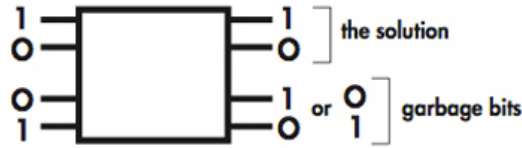


Figura 2: Exemplo de um portão XOR usando *dual-rail logic*.

3.2. Computação Reversível

Computadores reversíveis são capazes de gerar a entrada de uma instrução a partir de sua saída usando circuitos adiabáticos. Devido ao fato de serem reversíveis, estes computadores não são limitados pelo Princípio de Landauer discutido na Seção 2.6 e podem ultrapassar a eficiência energética de computadores normais. Para isso, é necessário, porém, que estes computadores sejam usados para rodar algoritmos reversíveis, que serão explorados na Seção 4.1.

Apesar de ser uma área ainda em desenvolvimento e ainda muito estudada, alguns projetos concretos já foram desenvolvidos, incluindo uma CPU desenvolvida pela MIT em 1999 e um *clock* adiabático desenvolvido pela AMD em 2012. O uso desse *clock* reduziu o consumo total de energia dos chips da AMD em 24%, o que mostra que não só a computação reversível realmente consegue melhorar a eficiência energética dos computadores, mas

também que ela não está tão longe de se tornar algo mais comum na computação.

3.3. Computação Magnética

A computação magnética é um campo que ainda está em desenvolvimento, mas que já mostra resultados promissores. Na computação magnética, o valor de um bit é representado pela direção em que um nano ímã está magnetizado, assumindo dois valores possíveis. A grande vantagem do uso desse sistema é que a energia dos dois estados possíveis de um bit requerem a mesma quantidade de energia, ou seja, são estados energeticamente balanceados, ao contrário da computação normal, em que um bit desligado tem voltagem baixa e um bit ligado tem uma voltagem alta.

Assim, o problema que a computação conservadora tenta resolver não existe na computação magnética, uma vez que os estados estão balanceados energeticamente. Assim, a energia dissipada por bits magnéticos é muito baixa: um experimento em Berkeley mostrou que a quantidade de energia dissipada na mudança de um valor de um bit magnético à temperatura ambiente é de 3 zeptojoules, isto é, 3×10^{-21} joules [3]. Este valor é especial por dois motivos. Primeiro, claramente é um valor extremamente baixo, o que reduziria muito a quantidade de calor gerada e aumentaria drasticamente a eficiência energética dos computadores. O segundo motivo, talvez mais impressionante, é que este valor é exatamente o limite inferior dado pelo Princípio de Landauer. Assim, a computação irreversível atingiria um limite muito importante e teria que buscar outras formas de reduzir o consumo de energia.

4. Soluções em Software

Ao contrário das soluções em hardware para o problema da eficiência energética, as soluções em software apresentadas aqui são sustentadas por algumas modificações na parte de hardware, sejam elas nos circuitos ou na quantidade de computadores rodando. Apesar disso, existem outras soluções em software que não foram exploradas nessa monografia e que não dependem tanto de modificações no hardware, como o modo de hibernação e o modo de pouca energia, que reduzem drasticamente a quantidade de energia usada pelos dispositivos.

4.1. Algoritmos Reversíveis

Algoritmos reversíveis procuram gastar o mínimo de energia possível, explorando as vantagens dos computadores reversíveis, como já mencionado anteriormente. Assim, esse tipo de algoritmo depende totalmente no uso de

computadores reversíveis, pois caso contrário tais algoritmos não terão nenhuma vantagem sobre algoritmos normais. Para diminuir o uso de energia, esses algoritmos tentam manter o número de 0s e 1s constantes durante toda a sua execução.

Pesquisadores da MIT mostraram que é possível fazer não só primitivas como *jumps*, *ifs* e chamadas de função sem nenhum desperdício de energia, mas também que é possível criar uma série de algoritmos muito úteis, como algoritmos de ordenação e de grafos, que também não gastam energia [1]. Assim, com o uso dessas variantes desses algoritmos é possível aumentar a eficiência energética dos computadores. Apesar disso, é importante notar que nem todo algoritmo pode ser reescrito para não gastar energia e, como mostra a Figura 3, os algoritmos reversíveis às vezes gastam mais tempo e memória para serem executados. Por exemplo, enquanto um BFS normal gasta a mesma quantidade de espaço e tempo que sua versão reversível, o algoritmo Floyd-Warshall gasta mais espaço em sua versão reversível.

Algorithm	Time	Space (words)	Energy (bits)	Thm.
Graph Algorithms				
Breadth-first Search	$\Theta(V + E)$	$\Theta(V + E)$	$\Theta(wV + E)$	6.9
Reversible BFS [Fra99]	$\Theta(V + E)$	$\Theta(V + E)$	0	6.10
Bellman-Ford	$\Theta(VE)$	$\Theta(V)$	$\Theta(VEw)$	6.12
Reversible Bellman-Ford	$\Theta(VE)$	$\Theta(VE)$	0	6.13
Floyd-Warshall	$\Theta(V^3)$	$\Theta(V^2)$	$\Theta(wV^3)$	6.14
Reversible Floyd-Warshall [Fra99]	$\Theta(V^3)$	$\Theta(V^3)$	0	6.15
Matrix APSP	$\Theta(V^3 \lg V)$	$\Theta(V^2)$	$\Theta(wV^3 \lg V)$	6.17
Reversible Matrix APSP [Fra99]	$\Theta(V^3 \lg V)$	$\Theta(V^2 \lg V)$	0	6.16
Semi-reversible Matrix APSP	$\Theta(V^3 \lg V)$	$\Theta(V^2)$	$\Theta(wV^2 \lg V)$	6.16
Data Structures				
Standard AVL Trees (build)	$O(n \lg n)$	$O(n)$	$O(w \cdot n \lg n)$	
(search)	$O(\lg n)$	$O(1)$	$O(\lg n)$	5.4
(insert)	$O(\lg n)$	$O(1)$	$O(w \lg n)$	5.5
(k deletes)	$O(k \lg n)$	$O(1)$	$O(w \lg n)$	5.6
Reversible AVL Trees (build)	$O(n \lg n)$	$O(n)$	0	
(search)	$O(\lg n)$	$O(1)$	0	5.7
(insert)	$O(\lg n)$	$O(1)$	0	5.8
(k deletes)	$O(k \lg n)$	$O(k)$	0	5.9
Standard Binary Heap (insert)	$O(\lg n)$	$O(1)$	$O(\lg n)$	5.10
(delete max)	$O(\lg n)$	$O(\lg n)$	$O(w \lg n)$	5.11
Reversible Binary Heap (insert)	$O(\lg n)$	$O(1)$	0	5.10
(delete max)	$O(\lg n)$	$O(\lg n)$	0	5.12
Dynamic Array (build)	$O(n)$	$O(n)$	0	
(query)	$O(1)$	$O(1)$	0	5.3
(add)	$O(1)$	$O(1)$	0	5.3
(delete)	$O(1)$	$O(1)$	0	5.3

Figura 3: Alguns dos algoritmos estudados desenvolvidos em [1] com seus respectivos custos de tempo, espaço e energia.

Os algoritmos reversíveis devem ser cuidadosamente desenhados para que haja o menor desperdício de energia possível e isso nem sempre é trivial. Para que isso aconteça, cada uma das operações feitas pelo computador deve ser reversível, o que significa que cada parte do algoritmo deve ser reversível.

Assim, um algoritmo é reversível se e só se cada etapa dele também é. Na sessão a seguir, vamos descrever uma versão reversível do algoritmo *merge sort* proposta em [1].

4.1.1. Exemplo: Merge Sort Reversível

Merge sort é um algoritmo recursivo de ordenação que pode ser dividido em três partes:

- a) Dividir o vetor em duas partes
- b) Usar cada uma das partes na recursão
- c) *Merge* das duas partes, produzindo um vetor ordenado

A base da recursão deste algoritmo é quando o tamanho do vetor é igual a um e, portanto, já está ordenado.

Para montar uma versão reversível deste algoritmo, basta fazer com que cada uma dessas partes sejam também reversíveis. O item a) é claramente reversível, uma vez que o inverso de separar um vetor v em $x = (e_0, \dots, e_k)$ e $y = (e_{k+1}, \dots, e_n)$ é montá-lo novamente calculando v a partir da Equação 3. O item b), por sua vez, será reversível se os outros dois itens também forem, já que estaremos chamando uma função recursivamente. Assim, para que o algoritmo seja reversível, basta mostrar que o item c) também é reversível.

$$v = x \oplus y = (e_0, \dots, e_k, e_{k+1}, \dots, e_n) \quad (3)$$

À primeira vista, é difícil perceber como o item c) pode ser de fato reversível. Ou seja, é necessário descobrir como que, a partir de um vetor ordenado, pode-se chegar a dois vetores ordenados específicos, aqueles que foram dados como entrada para fazer o *merge*. É fácil ver que tal tarefa é impossível se não forem guardadas mais informações sobre o *merge* - não há como saber de qual vetor um dado elemento e_i foi tirado para formar o vetor v .

Felizmente, há um truque simples para que isso seja trivial. Antes de começar o algoritmo, o vetor v deve ser aumentado, colocando ao lado de cada elemento e_i o valor i , formando o vetor \bar{v} de tamanho $2n$. Assim, $\bar{v} = (e_0, 0, e_1, 1, e_2, 2, \dots, e_n, n)$. Cada par de elementos de \bar{v} é então tratado como um único elemento e apenas o primeiro é usado para as comparações de ordenação. Assim, no início de cada *merge* tem-se dois vetores:

$$\begin{aligned} x &= (e_r, r, \dots, e_s, s) \\ y &= (e_{s+1}, s+1, \dots, e_t, t) \end{aligned}$$

Note que todos os elementos de x terão índices $i \leq s$ e que todos os elementos de y terão $i > s$. Assim, a partir do vetor resultante \bar{v} do *merge* é fácil chegar de volta nos vetores x e y , já que eles são ordenados e já que sabemos a qual deles cada elemento pertence, a partir do seu índice i .

Dessa forma, como os itens a) e c) são reversíveis, segue que b) também é e, portanto, esta implementação de *merge sort* também é. O custo energético desta implementação é zero e usa apenas o dobro da memória da versão normal.

4.2. Virtualização

A virtualização é um dos principais responsáveis pela queda consumo de energia, principalmente pela não necessidade das empresas terem um grande diversidade de *hardwares* para realizar determinada tarefa, ou seja, é possível criar dentro de uma mesma máquina diversas outras máquinas virtuais com suas particularidades para atender a suas necessidades. Assim pode-se ter várias máquinas com diferentes sistemas operacionais, por exemplo rodando em uma mesma máquina. Logo não precisa-se criar toda uma infra-estrutura de refrigeração para refrigerar as novas máquinas que seriam necessário se não tivesse a virtualização e por consequência diminuem quantidade energia gasta.

4.3. Computação em Nuvem

Não iremos definir aqui o que seja computação em nuvem, iremos pressupor que o leitor já esteja familiarizado com o assunto, no entanto iremos ressaltar superficialmente alguns aspectos que a computação em nuvem que contribuem com a computação verde.

O compartilhamento de recursos é um dos aspectos mais importantes para computação verde, principalmente devido a não necessidade de empresas pequenas e grandes de construir toda sua infra-estrutura para atender uma determinada necessidade, ou seja, a computação em nuvem tornou possível alocação dinâmica de recursos computacionais e foi além tornou possível que o chamamos hoje SaaS (*Software as a services*), que trás para muitos usuários softwares como um serviço, que podem serem acessados via o navegador.

Outro aspecto importante é a técnica DCD (dynamic component deactivation) que consiste em desligar os componentes quando não estão em uso [4]. Isso torna possível que as grandes e pequenas empresas pagarem apenas pelo tempo que realmente os recursos estão em uso.

5. Conclusão

A preocupação com o consumo de energia dos computadores está crescendo ao redor do mundo e muitas pesquisas estão sendo feitas para que isso

aconteça. Existe de fato um esforço global para que o consumo de energia dessas tecnologias seja controlado e isso está gerando resultados muito importantes. As mudanças propostas, no entanto, tem um grande impacto na forma que softwares e hardwares são projetados, alterando desde a forma que bits são representados até os algoritmos mais básicos da computação.

As soluções vistas nesta monografia são apenas algumas que selecionamos para discutir, mas existem muitas outras propostas sendo feitas. Assim, mesmo que as respostas aqui apresentadas não sejam definitivas ou até mesmo implementadas até o final, a computação energeticamente eficiente continuará se desenvolver, trazendo soluções cada vez melhores.

Referências

- [1] Erik D. Demaine, Jayson Lynch, Geronimo J. Mirano, and Nirvan Tyagi. Energy-efficient algorithms. 2016.
- [2] Here's how much energy all us data centers consume. <http://www.datacenterknowledge.com/archives/2016/06/27/heres-how-much-energy-all-us-data-centers-consume/>. Accessed: 2017-06-15.
- [3] Brian Lambson, David Carlton, and Jeffrey Bokor. Exploring the thermodynamic limits of computation in integrated systems: Magnetic memory, nanomagnetic logic and the landauer limit. *Physical Review Letters*, 107(1):1–3, 2011.
- [4] Carlos Becker Westphall and Sergio Roberto Villarreal. Princípios e tendências em green cloud computing. *Revista Eletrônica de Sistemas de Informação*, 12(1):1–19, 2013.