



PROPOSIÇÃO E ANÁLISE DE UM ALGORITMO CIENTE DE CONSUMO DE ENERGIA PARA GRADES MÓVEIS

Luiz C. Borro, Sarita M. Bruschi, Geraldo P. R. Filho

Instituto de Ciências Matemáticas e de Computação - ICMC USP

Universidade de São Paulo (USP)

Caixa Postal 668 - São Carlos - SP - Brasil

{lborro, sarita, geraldop}@icmc.usp.br

RESUMO

Considerando-se o contexto de gerenciamento energético em grades móveis, neste trabalho foi proposto um algoritmo de escalonamento (ECCE) que, além de minimizar o consumo de energia, visa assegurar o cumprimento dos requisitos de qualidade de serviço das aplicações submetidas pelos usuários. Tal algoritmo foi projetado a partir de uma solução heurística para o problema de escalonamento ciente de consumo de energia em grades móveis, que foi modelado como um problema de otimização envolvendo variáveis binárias. Por meio de experimentos numéricos, foi demonstrada a viabilidade dos algoritmo proposto em relação à redução do consumo de energia.

PALAVRAS CHAVE. Escalonamento, avaliação de desempenho, grades móveis.

Área Principal: OA - Outras aplicações em PO.

ABSTRACT

Considering the context of energy management in mobile grids, this work proposes a scheduling algorithm (ECCE) that aims not only to reduce the energy consumption of the mobile devices, but also to ensure the QoS (Quality of Service) requirements of the running applications. This algorithm was designed based on a heuristic for the energy aware scheduling problem in mobile grids, which was modeled as an optimization problem with integer variables. The performance of the proposed scheduling algorithm was evaluated by an extensive set of experiments, which demonstrated the feasibility of the adopted approach regarding energy consumption minimization.

KEYWORDS. Independent tasks scheduling, computer systems performance analysis, mobile grid.

Main Area: OA - Other applications in OR.

1. Introdução

Na década de 1990, o aumento da complexidade das aplicações científicas motivou o desenvolvimento de uma infraestrutura de computação distribuída inspirada nas redes de energia elétrica (Buyya et al., 2005). Nessa infraestrutura, conhecida como grade computacional, recursos computacionais (por exemplo, processadores, discos e licenças de *software*) geograficamente distribuídos podem ser compartilhados de maneira coordenada (Foster et al., 2001). Dessa forma, problemas complexos que demandam uma grande quantidade de recursos podem ser resolvidos de maneira colaborativa.

Inicialmente concebido considerando-se apenas o compartilhamento de recursos entre nós fixos, o conceito de grade computacional tem sido estendido gradativamente para ambiente móveis (Furthmüller e Waldhorst, 2010; Rodriguez et al., 2011). A integração entre dispositivos móveis (por exemplo, *smartphones*, *tablets* e *notebooks*) e grades computacionais, denominada de grade móvel (*mobile grid*), pode ser realizada de duas maneiras distintas. Na primeira abordagem, os dispositivos móveis apenas consomem recursos computacionais, funcionando como uma interface de acesso à grade. Já na segunda abordagem, os dispositivos móveis podem atuar como provedores de recursos, tendo a possibilidade de participar diretamente da execução colaborativa de aplicações.

Infactível há alguns anos, a concepção de uma grade móvel voltada para o compartilhamento de recursos pode ser atualmente justificada por dois motivos (Furthmüller e Waldhorst, 2010; Rodriguez et al., 2011). Primeiro, a rápida evolução tecnológica dos dispositivos móveis tanto em termos de *hardware* quanto de *software*. Segundo, o grande número de dispositivos móveis, em especial *smartphones*, comercializados anualmente. De acordo com dados da consultoria Gartner¹, em 2011, foram vendidos mundialmente mais de 470 milhões de *smartphones*, caracterizando um aumento de 58% em relação ao ano anterior.

Existe potencialmente, portanto, uma grande quantidade de recursos computacionais que pode ser explorada por meio de uma grade móvel no desenvolvimento de aplicações para áreas diversas. Deve-se ressaltar, no entanto, que uma infraestrutura para compartilhamento de recursos ofertados por dispositivos móveis constitui um ambiente altamente volátil. A disponibilidade dos recursos pode ser afetada por problemas associados à mobilidade, comunicação e capacidade de bateria, fazendo com que o desempenho das aplicações submetidas seja prejudicado. Particularmente, a capacidade de bateria é um fator crítico, pois, além de afetar a disponibilidade, influencia diretamente os usuários de dispositivos móveis na decisão de compartilhar ou não seus recursos computacionais. Em uma pesquisa realizada por Furthmüller e Waldhorst (2012), constatou-se que a capacidade de energia de um dispositivo é preponderante para essa decisão. Mais da metade dos entrevistados (53%) afirmaram que não compartilhariam seus recursos em virtude do tempo de vida de bateria disponível.

Nesse contexto, considerando-se uma grade móvel em que o compartilhamento de recursos computacionais é incentivado por meio de compensações diretamente proporcionais à energia consumida pela execução das aplicações, este trabalho propõe um algoritmo de escalonamento ciente do consumo de energia. O objetivo é reduzir o custo de manutenção do sistema, caracterizado em função do consumo energético dos dispositivos móveis participantes, bem como assegurar que os requisitos de qualidade de serviço das aplicações submetidas sejam cumpridos.

O restante deste artigo está organizado da seguinte forma. A Seção 2 aborda os trabalhos relacionados a grades móveis, além de estratégias de escalonamento cientes do consumo de energia. Um algoritmo de escalonamento ciente de consumo de energia para grades móveis é proposto na Seção 3, e avaliado na Seção 4. Finalmente, a Seção 5 apresenta as principais conclusões deste trabalho, bem como sugestões de trabalhos futuros.

¹<http://www.gartner.com/it/page.jsp?id=1924314>

2. Trabalhos relacionados

2.1. Grades móveis

Considerando-se o contexto deste trabalho, a concepção de uma grade móvel voltada para compartilhamento de recursos pode ser justificada pelo crescente número de dispositivos móveis comercializados, juntamente com a evolução tecnológica desses dispositivos em termos de processamento, armazenamento e comunicação (Furthmüller e Waldhorst, 2010). Especificamente em relação à capacidade de processamento, em um estudo realizado por Rodriguez et al. (2012a), demonstrou-se que a utilização de um sistema distribuído composto por *smartphones* pode ser útil na resolução de problemas que envolvam algoritmos iterativos com predomínio de operações inteiras.

De acordo com Katsaros e Polyzos (2007), a provisão de recursos móveis pode ser realizada considerando-se duas arquiteturas fundamentalmente distintas: grades móveis *on-site* e grades móveis *ad hoc*.

As grades móveis *on-site* dependem necessariamente de uma infraestrutura de rede preexistente, sendo os dispositivos móveis coordenados por uma entidade central. Informações sobre a localização e disponibilidade dos recursos são monitoradas e repassadas para a entidade central, que mantém uma lista com o *status* de cada dispositivo. Um exemplo desse tipo de arquitetura *on-site* foi proposto por Ghosh e Das (2010). Tal arquitetura se baseia em uma rede de telefonia celular na qual cada célula é composta por um conjunto de dispositivos móveis e um ponto de acesso sem fio (WAP, do inglês *Wireless Access Point*). Como os autores consideraram o protocolo IEEE 802.1, cada célula caracteriza um conjunto básico de serviços (BSS, do inglês *Basic Service Set*). Em cada BSS, existe um servidor WAP que é responsável pelo monitoramento dos recursos disponibilizados pelos dispositivos móveis. Esses servidores estão conectados a um roteador de borda, que serve de interface com a entidade que controla a grade móvel.

Já as grades móveis *ad hoc* não dependem de uma infraestrutura de rede preestabelecida. A topologia da rede é formada de maneira dinâmica e independente, com os dispositivos móveis podendo participar ou sair da grade a qualquer momento, criando-se assim cenários de compartilhamento temporários. Além disso, tanto a demanda quanto a disponibilidade de serviços e recursos podem apresentar uma alta taxa de variação, em comparação com os cenários encontrados nas grades *on-site* (Lima, 2007).

Em relação a possíveis cenários de uso, as grades móveis têm sido empregadas em aplicações sensíveis ao contexto (Coronato e Pietro, 2008), *e-healthcare* (Viswanathan et al., 2012a,b), *e-learning* (D'Andria et al., 2008), gerenciamento instalações agrícolas (Wu et al., 2011), bem como na execução de aplicações científicas (Rodriguez et al., 2012a).

Particularmente em (Viswanathan et al., 2012b), os autores propuseram um *middleware* voltado para o desenvolvimento de aplicações de *e-healthcare* sensíveis ao contexto. Tal *middleware* permite que dispositivos heterogêneos em uma determinada vizinhança formem um conjunto elástico de recursos computacionais, que possa ser aproveitado para processar coletivamente grandes quantidades de dados obtidos por meio de sensores médicos. Os dispositivos participantes atuam como provedores de serviços ou árbitros. Os provedores de serviços podem prover dados (por exemplo, monitor cardíaco), recursos (por exemplo, *tablets* e *smartphones*) ou ambos. Já um dispositivo árbitro funciona como um *broker*, sendo responsável pelo atendimento das requisições submetidas pelos usuários (neste caso, profissionais de saúde), bem como pelo gerenciamento dos servidores de dados e recursos.

2.2. Escalonamento ciente do consumo de energia

Recentemente, a redução do consumo de energia em sistemas computacionais se tornou um importante tópico de pesquisa. Em particular, na área de sistemas distribuídos, diversos trabalhos que propõem estratégias de escalonamento cientes do consumo de energia foram publicados nos últimos anos (Kim et al., 2007; Li e Li, 2010; Rodriguez et al., 2010; Ma et al., 2012; Li e Li,

2012; Rodriguez et al., 2012b). Essas estratégias visam minimizar o consumo de energia sem que o desempenho das aplicações seja comprometido.

Em (Kim et al., 2007), são apresentados algoritmos de escalonamento cientes do consumo de energia para aplicações *bag-of-tasks* em *clusters* homogêneos com suporte a DVS (*Dynamic Voltage Scaling*). DVS é uma técnica que permite ajustar dinamicamente as voltagens de processadores, com o objetivo de reduzir o consumo de energia. O modelo de escalonamento adotado para os *clusters* considera restrições de *deadline* das aplicações, sendo aplicado um regime de admissão. Baseados na heurística EDF (*Earliest Deadline First*), os algoritmos propostos ajustam as voltagens dos processadores de forma a minimizar o consumo total de energia e garantir o cumprimento dos *deadlines* das aplicações. Nos experimentos executados, observou-se que os algoritmos propostos proporcionaram uma redução no consumo de energia às custas de uma degradação na taxa de aceitação das aplicações.

Ma et al. (2012) propuseram o algoritmo EES (*Energy Efficient Scheduling*) para o escalonamento de tarefas independentes em sistemas distribuídos heterogêneos de alto desempenho. O algoritmo foi projetado a partir de uma solução heurística para o problema de escalonamento ciente de consumo de energia com restrições de *deadline*, que foi formulado como um modelo de programação linear inteira. O escalonamento das tarefas é realizado de forma a garantir que a maior quantidade possível de carga de trabalho seja atribuída aos recursos mais eficientes do ponto de vista energético. Em comparação a um referencial ótimo, determinado pela solução da relaxação linear do problema de escalonamento considerado, o algoritmo EES apresentou bom desempenho em termos de minimização de energia e satisfação de *deadline*.

Em (Li e Li, 2012), visando equilibrar o consumo de energia em uma grade móvel, os autores propuseram um algoritmo de escalonamento colaborativo baseado em uma estratégia orientada ao mercado (*market-oriented*). Um modelo para determinação de preços foi adotado, sendo a energia tratada como um recurso que pode ser negociado entre os dispositivos móveis. O escalonamento das tarefas é realizado de maneira a maximizar a utilidade da grade móvel (definida em função das negociações realizadas), não excedendo o *deadline* das aplicações e a capacidade de bateria de cada dispositivo. Uma estratégia semelhante também é empregada em (Li e Li, 2010). Deve-se ressaltar, no entanto, que estratégias orientadas ao mercado são baseadas na lei da oferta e procura, sendo muitas vezes um processo lento (Ma et al., 2012), que pode ser inviável em alguns cenários de grade móvel.

Rodriguez et al. (2010) desenvolveram o SEAS (*Simple Energy-Aware Scheduler*), um escalonador para grades móveis com ênfase para tarefas que exigem uso intensivo do processador. A política de escalonamento adotada pelo SEAS se baseia em três aspectos dos dispositivos móveis: potência de processamento, carga de bateria e taxa de consumo de bateria. A partir de uma estimativa do tempo de vida de bateria, o escalonador determina periodicamente o número máximo de tarefas que pode ser alocado em cada dispositivo, métrica utilizada para determinar o mapeamento das tarefas (preferência é dada ao dispositivo que suporta o maior número de tarefas). Como a taxa de consumo pode variar em função de diferentes fatores (por exemplo, intensidade de uso da rede), expandiram a proposta original, adicionando mecanismos de balanceamento de cargas baseados em técnicas de *job stealing* (Rodriguez et al., 2012b).

3. Escalonamento ciente do consumo de energia em grades móveis

De maneira similar ao trabalho de Ma et al. (2012), o algoritmo proposto neste trabalho se baseia em uma solução heurística para o problema de escalonamento ciente de consumo de energia. No entanto, deve ser ressaltado que, no contexto de uma grade móvel, tal problema, além das restrições de *deadline* das aplicações, também envolve restrições relativas à capacidade de bateria dos dispositivos.

Para o projeto do algoritmo, foram consideradas três etapas. Primeiro, levando-se em consideração aspectos como arquitetura, tipo de aplicação e caracterização dos recursos, criou-se um modelo de grade móvel. Em seguida, a partir desse modelo, o escalonamento ciente de consumo

de energia foi formulado como um problema de otimização com variáveis binárias. Por fim, tendo-se em vista particularidades do problema, o algoritmo foi projetado com base em uma solução heurística. A seguir, cada uma dessas etapas será discutida em detalhes.

3.1. Modelo de grade móvel

O modelo de grade móvel adotado neste trabalho considera a arquitetura *on-site* ilustrada na Figura 1. Nessa arquitetura, os recursos, caracterizados por dispositivos móveis localizados em uma área bem definida (um *hotspot* de uma WLAN, por exemplo), são coordenados por um servidor *proxy* associado a um ponto de acesso (PA). Esse servidor *proxy* atua como um *broker*, sendo responsável por coletar informações sobre os recursos (potência de processamento, a estimativa de consumo de energia e a capacidade de bateria disponível), bem como por escalonar as tarefas que compõem as aplicações submetidas pelos usuários da grade móvel.

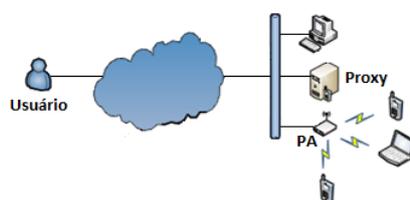


Figura 1: Arquitetura de grade móvel considerada na modelagem do problema de escalonamento.

As aplicações submetidas pelos usuários para execução na grade móvel são do tipo *bag-of-tasks*, sendo, portanto, compostas por tarefas independentes que podem ser escalonadas de maneira arbitrária. Além disso, cada aplicação está associada a um prazo máximo de execução (*deadline*), que funciona como um requisito de qualidade de serviço. O escalonamento é não-preemptivo, sendo as tarefas executadas sem interrupção, respeitando a ordem com que foram mapeadas aos recursos.

Como a capacidade de bateria dos dispositivos móveis influencia diretamente os usuários na decisão de compartilhar ou não seus recursos computacionais (Furthmüller e Waldhorst, 2012), a participação na grade móvel é incentivada por meio de compensações diretamente proporcionais a energia consumida pela execução das aplicações submetidas. É importante, logo, que a grade móvel seja capaz de gerenciar o consumo de energia de maneira adequada, reduzindo os custos associados, como também assegurando que o desempenho do sistema não seja comprometido.

3.2. Formulação do problema

Considerando-se o modelo descrito, o problema de escalonamento ciente do consumo de energia em grades móveis foi definido da seguinte forma: dado um conjunto de m recursos e uma aplicação *bag-of-tasks* composta por n tarefas, atribua cada uma das tarefas a exatamente um recurso, de maneira a minimizar o consumo de energia, bem como satisfazer as restrições relativas ao *deadline* e à capacidade de bateria dos recursos. Mais especificamente, esta definição é uma variação do problema generalizado de atribuição (GAP, do inglês *Generalized Assignment Problem*).

Logo, de acordo com a notação apresentada na Tabela 1, o problema de escalonamento pode ser formulado pelo modelo de programação linear inteira definido por (1)–(4), sendo x_{ij} uma variável binária que recebe 1 apenas quando j -ésima tarefa da aplicação é atribuída ao i -ésimo recurso, $t_{ij} = w_j/q_i$ e $e_{ij} = t_{ij} \times p_i$. A função objetivo (1) visa minimizar o consumo total de energia necessário para a execução da aplicação. As restrições definidas em (2) asseguram que o *deadline* da aplicação e a capacidade de bateria dos recursos não sejam excedidos. As restrições de atribuição (3) garantem que cada tarefa seja atribuída a exatamente um recurso. O domínio das variáveis é definido em (4). Deve-se ressaltar que, devido às restrições de atribuição (3), uma instância do GAP não necessariamente possui uma solução factível (Martello e Toth, 1990).

Tabela 1: Notação utilizada na formulação do problema de escalonamento ciente do consumo de energia.

Parâmetro	Descrição
q_i	Potência de processamento do recurso i (MIPS, Milhões de Instruções por Segundo)
p_i	Estimativa do consumo de energia do recurso i (Watt)
c_i	Capacidade de bateria do recurso i (Joule)
w_j	Carga de trabalho da j -ésima tarefa da aplicação (MI, Milhões de Instruções)
d	Deadline da aplicação (segundos)
t_{ij}	Tempo de execução da tarefa j no recurso i (segundos)
e_{ij}	Consumo de energia relativo à execução da tarefa j no recursos i (Joule)

$$\text{Minimizar } E = \sum_{i=1}^m \sum_{j=1}^n e_{ij} x_{ij} \quad (1)$$

$$\text{Sujeito a } \sum_{j=1}^n t_{ij} x_{ij} \leq \min\{d, c_i/p_i\}, \forall i \quad (2)$$

$$\sum_{i=1}^m x_{ij} = 1, \forall j \quad (3)$$

$$x_{ij} \in \{0, 1\}, \forall i, j \quad (4)$$

3.3. Algoritmo heurístico

Como o GAP é \mathcal{NP} -difícil no sentido forte (*strong sense*) (Martello e Toth, 1990), solucionar de maneira ótima o problema de programação linear inteira apresentado é um processo computacionalmente caro, inviável no contexto de uma grade móvel, uma vez que o tempo necessário pode ser sensivelmente superior à escala de tempo de execução das aplicações submetidas pelos usuários. Logo, para solucionar tal problema, foi proposto um algoritmo guloso, denominado ECCE (Escalonamento Ciente do Consumo de Energia), que privilegia os recursos mais energeticamente eficientes, caracterizado por duas etapas principais: seleção dos recursos (Linha 3); e determinação da carga de trabalho que pode ser alocada em tais recursos sem que as restrições de *deadline* e capacidade de bateria sejam violadas (Linha 4).

Algoritmo 1: ECCE

Entrada: Um conjunto de recursos $R = \{R_1 \dots R_m\}$ sendo $R_i = (q_i, p_i, c_i)$; Uma aplicação $A = (d, W = \{w_1 \dots w_n\})$

1 ordenar W em ordem decrescente

2 **repita**

3 $i^* \leftarrow \arg \min_{i: R_i \in R} \{p_i/q_i\}$

4 $g \leftarrow \text{MTGS}(W, \min\{d, c_{i^*}/p_{i^*}\}, q_{i^*})$

5 **para cada** $j \in g$ **faça**

6 \quad atribua a tarefa j ao recurso i^*

7 **fim para cada**

8 $R \leftarrow R \setminus \{R_{i^*}\}$

9 $W \leftarrow W \setminus \{w_j : j \in g\}$

10 **até** que todas as tarefas tenham sido atribuídas **Ou** mais nenhuma tarefa possa ser alocada nos recursos disponíveis;

Considerando-se os recursos disponíveis, seleciona-se o recurso i^* com menor valor para a razão p_i/q_i (custo em Joule por MI). Em seguida, deve-se determinar a maior carga de trabalho possível que pode ser alocada no recurso, processo que corresponde a solucionar o problema da soma do subconjunto (SSP, do inglês *Subset-Sum Problem*), definido por (5)–(7), onde x_j é variável binária que recebe 1 apenas se a tarefa j é atribuída ao recurso i^* . O SSP é um caso particular do problema da mochila (*knapsack problem*) no qual, em relação a um determinado valor alvo, espera-se a obter uma solução que minimize o desvio negativo (Martello e Toth, 1990).

$$\text{Maximizar } E = \sum_{j=1}^n w_j x_j \quad (5)$$

$$\text{Sujeito a } \sum_{j=1}^n w_j x_j \leq \min\{d, c_{i^*}/p_{i^*}\} \times q_{i^*} \quad (6)$$

$$x_j \in \{0, 1\}, \forall j \quad (7)$$

Assim como o GAP, o problema da soma do subconjunto pertence à classe \mathcal{NP} -difícil (Martello e Toth, 1990). Dessa forma, para obter soluções factíveis suficientemente próximas à solução ótima em um intervalo de tempo razoável, foi escolhido o algoritmo aproximativo MTGS, proposto por (Martello e Toth, 1984), que possui complexidade $O(n^2)$ e razão de desempenho no pior caso $r = \frac{3}{4}$. A ideia por trás do algoritmo MTGS é avaliar soluções obtidas para diferentes subconjuntos de tarefas. Mais precisamente, a partir de um conjunto inicial de tarefas $\{1, \dots, n\}$, são definidos n subconjuntos: $\{1, \dots, n\}$, $\{2, \dots, n\}$, $\{3, \dots, n\}$ e assim por diante. Para cada um desses subconjuntos, uma abordagem gulosa de atribuição de tarefas é aplicada, sendo escolhida a solução que proporciona a maior alocação de carga de trabalho no recurso.

4. Avaliação de desempenho

Nesta seção, o desempenho do algoritmo ECCE é avaliado através de diversos experimentos numéricos. O objetivo é determinar a acurácia do ECCE, isto é, sua capacidade de encontrar soluções próximas o suficiente de um referencial ótimo, determinado pelo *solver* Gurobi² (versão 5.5, licença acadêmica).

4.1. Planejamento de experimentos

O planejamento de experimentos é uma etapa fundamental na avaliação de desempenho de qualquer sistema computacional. Nela, definem-se as características que serão avaliadas, examinando-se quais delas podem influenciar no desempenho do sistema (Jain, 1991). O objetivo do planejamento de experimentos é manipular de forma planejada certas variáveis independentes (fatores), definindo-se os valores mais prováveis que essas variáveis podem assumir (níveis), com a finalidade de verificar o efeito que esta manipulação provoca na variável de resposta (variável dependente).

Neste trabalho, utilizou-se o planejamento fatorial completo, que contabiliza todas as combinações considerando todos os fatores e níveis. Particularmente, os experimentos tiveram como base quatro fatores com dois níveis (Tabela 2): solução para o problema de escalonamento, quantidade de recursos, quantidade de tarefas e o *deadline* da aplicação. Como o objetivo dos experimentos é determinar a capacidade do algoritmo proposto em obter boas soluções dentro de um intervalo de tempo razoável, foram consideradas como variáveis de resposta duas métricas: o consumo total de energia e o tempo de resposta.

As instâncias do problema de escalonamento ciente de consumo de energia foram derivadas a partir de cinco configurações de dispositivos móveis, distribuídas de maneira uniforme de acordo com o número de recursos disponíveis. A potência de processamento (q_i) e o consumo de

²<http://www.gurobi.com/>

Tabela 2: Fatores e níveis considerados no planejamento fatorial completo.

Fatores	Níveis
Solução	ECCE e Gurobi
Recursos	30 e 45
Tarefas	50 e 70
Deadline	curto e estendido

energia (p_i) para cada uma dessas configurações (Tabela 3) foram estimados com base no trabalho de Carroll e Heiser (2010). Com relação à carga de trabalho (w_j), os valores adotados dependem da granularidade da aplicação, determinada pela quantidade de tarefas (Tabela 4). Nesse sentido, as respectivas distribuições foram estabelecidas de maneira a garantir que o tamanho esperado de uma aplicação seja de $2,5 \times 10^5$ MI.

Tabela 3: Configuração dos dispositivos móveis considerados nos experimentos. Valores foram derivados das estimativas apresentadas por Carroll e Heiser (2010).

Dispositivo	q_i (MIPS)	p_i (mW)
1	110	60
2	295	435
3	440	180
4	460	500
5	539	234

Tabela 4: Carga de trabalho das tarefas que compõem uma aplicação. A distribuição uniforme considerada depende da granularidade da aplicação.

Número de Tarefas	w_i (MI)
50	[4000;6000]
70	[2900;4241]

Em relação ao *solver* Gurobi, os seguintes parâmetros foram modificados em relação à configuração padrão: TimeLimit = 60 segundos e MIPGap = 1×10^{-4} . O parâmetro TimeLimit limita o tempo total gasto na busca por soluções factíveis. Já o parâmetro MIPGap determina que o processo de busca será encerrado quando a diferença relativa entre os limitantes inferior e superior da função objetivo for menor que MIPGap vezes o limitante superior.

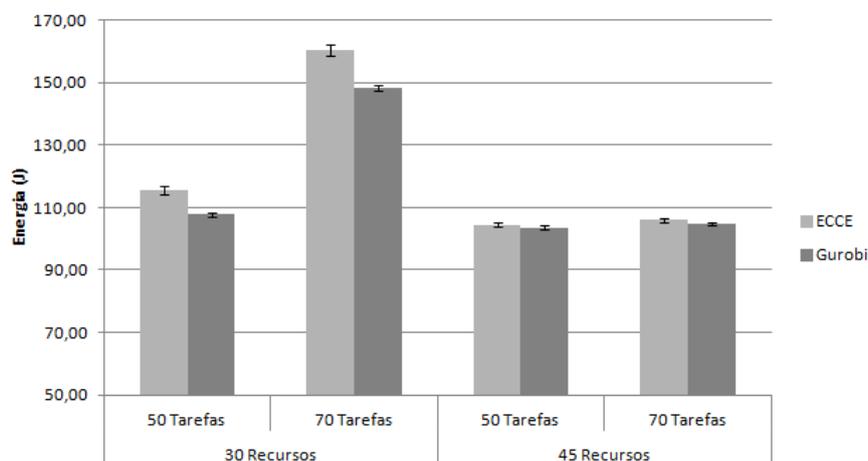
Para cada experimento, foram realizadas 30 replicações³, utilizadas para determinar a média, desvio padrão e intervalo de confiança de 95% de acordo com a distribuição *T-student*. A quantidade de replicações foi adotada para que fosse possível obter intervalos de confiança suficientemente pequenos. Já a capacidade de bateria de cada recurso segue uma distribuição uniforme com mínimo de 1000 J e máximo de 5000 J.

4.2. Resultados obtidos

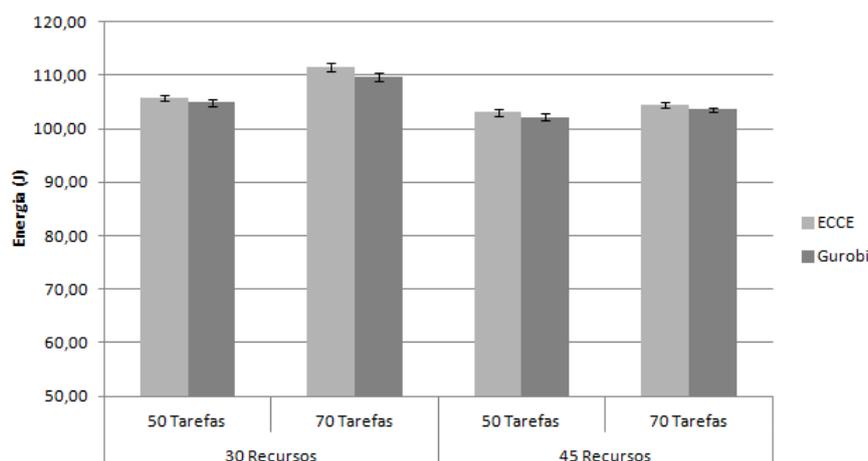
Na Figura 2, são apresentados os resultados relativos à métrica consumo de energia. Considerando-se o cenário com *deadline* curto (Figura 2(a)), observa-se que as soluções obtidas pelo algoritmo ECCE foram numericamente superiores que as do *solver* Gurobi, ou seja, foram inferiores em termos de otimizar o consumo de energia. Particularmente, a maior diferença no consumo de energia em relação ao referencial ótimo (30 recursos, 70 tarefas) foi de apenas 8,14%. Deve-se ressaltar, no entanto, que o aumento do número de recursos disponíveis de 30 para 45

³Neste caso, cada replicação representa uma instância do problema de escalonamento ciente do consumo de energia

fez com que essa diferença de desempenho fosse praticamente marginal. Isso ocorre, em grande parte, pois é possível escalar um número maior de tarefas nos dispositivos energeticamente mais eficientes, sem que haja sobrecarga dos mesmos.



(a) *deadline* curto



(b) *deadline* estendido

Figura 2: Consumo de energia para os experimentos planejados. Para cada experimento, é apresentado o valor médio das replicações e o respectivo intervalo de confiança. Os resultados foram organizados de acordo com o *deadline*.

Outro aspecto importante está relacionado ao *deadline* da aplicação. Nesse sentido, verifica-se que, independentemente dos demais fatores, a adoção de um *deadline* mais relaxado (Figura 2(b)) implica em uma redução significativa do consumo de energia. Nesse contexto, a maior diferença no consumo de energia do ECCE em relação ao referencial ótimo foi de apenas 1,60% (30 recursos, 70 tarefas).

Em relação à métrica tempo de resposta, os resultados obtidos nos experimentos planejados são apresentados na Tabela 5. Para algoritmo ECCE, observa-se que o tempo necessário para encontrar soluções factíveis é diretamente proporcional ao tamanho da instância do problema, refletindo, portanto, apenas a quantidade de variáveis envolvidas.

Por outro lado, para o *solver* Gurobi, a busca por uma solução ótima não dependeu apenas do número de variáveis, tendo sido também influenciada por características relativas à instância do problema, determinadas basicamente pela carga de trabalho das tarefas (elemento aleatório). Na maior parte das instâncias, as soluções providas pelo Gurobi atingiram o tempo limite estipulado,

Tabela 5: Tempo de processamento (segundos) necessário para a obtenção de soluções factíveis. Para cada experimento, é apresentado o valor médio das replicações juntamente com o respectivo intervalo de confiança. Os dados foram sumarizados de acordo com o *deadline*.

Solução	30 Recursos		45 Recursos	
	50 Tarefas	70 Tarefas	50 Tarefas	70 Tarefas
<i>deadline curto</i>				
ECCE	0,01 ± 0,00	0,02 ± 0,00	0,01 ± 0,00	0,02 ± 0,00
Gurobi	57,16 ± 4,08	60,02 ± 0,01	38,48 ± 8,30	49,67 ± 5,73
<i>deadline estendido</i>				
ECCE	0,01 ± 0,00	0,02 ± 0,00	0,01 ± 0,00	0,02 ± 0,00
Gurobi	24,63 ± 9,15	60,04 ± 0,01	52,70 ± 5,79	53,87 ± 5,60

não satisfazendo, portanto, a condição de parada determinada pelo parâmetro MIPGap. No entanto, em alguns casos, a busca convergiu rapidamente para uma solução. Tal comportamento é ilustrado no gráfico de *boxplot* para instâncias do problema com *deadline* estendido (Figura 3).

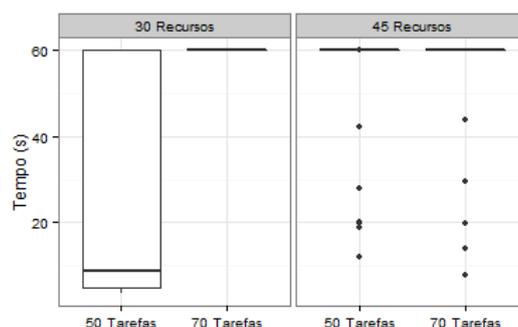


Figura 3: Tempo de resposta do *solver* Gurobi para instâncias do problema de escalonamento com *deadline* estendido.

Em gráfico de *boxplot*, a caixa é definida a partir do primeiro e terceiros quartis, sendo a barra interna definida pela mediana. A representação da amplitude interquartil (diferença entre o terceiro e primeiro quartis) indica o grau de dispersão dos dados, bem como auxilia na identificação de *outliers*. Especificamente para as instâncias do problema com 45 recursos, nota-se uma concentração de valores em 60 segundos (tempo limite estipulado) e alguns poucos *outliers*, indicados por pontos.

5. Conclusão

Considerando-se o contexto de gerenciamento energético em grades móveis, este trabalho propôs um algoritmo de escalonamento (ECCE) que, além de minimizar o consumo de energia, tem como objetivo assegurar o cumprimento dos requisitos de qualidade de serviço das aplicações submetidas pelos usuários. Tal algoritmo foi projetado a partir de uma solução heurística para o problema de escalonamento ciente de consumo de energia em grades móveis. Mais precisamente, o problema de escalonamento foi formulado como um modelo de programação linear inteira, cujo objetivo é minimizar o consumo total de energia necessário para a execução de aplicações *bag-of-tasks* em uma grade móvel *on-site*.

Em relação ao consumo total de energia, pôde-se demonstrar que algoritmo proposto foi capaz de encontrar soluções próximas ao referencial ótimo dentro de um intervalo de tempo razoável. Em seu pior caso, o algoritmo ECCE foi 8,14% pior que a solução provida pelo *solver* Gurobi. Deve-se ressaltar que o Gurobi nem sempre foi capaz de encontrar soluções ótimas (isto é,

que satisfizessem o parâmetro MIPGap) no tempo estipulado. Tal característica não é desejável em um sistema dinâmico como uma grade móvel, sendo justificada, portanto, a utilização de heurísticas para a solução do problema de escalonamento ciente de consumo de energia.

Como trabalhos futuros, pretende-se ampliar o estudo apresentado neste artigo, levando-se em consideração também aspectos relacionados à mobilidade dos dispositivos móveis. Como a taxa de mobilidade desses dispositivos pode ser alta, ocasionando problemas que podem afetar a disponibilidade dos recursos ofertados, faz-se necessário a adoção de técnicas que garantam um determinado nível de tolerância a falhas para a grade móvel. Além disso, planeja-se também aprimorar a caracterização dos dispositivos móveis em termos de consumo de energia, levando-se em consideração custos associados à comunicação (WiFi, Bluetooth), iluminação de tela e execução de aplicações locais.

Referências

- Buyya, R., Abramson, D. e Venugopal, S.** (2005). The grid economy. *Special Issue on Grid Computing, Proceedings of the IEEE*, 93:698–714.
- Carroll, A. e Heiser, G.** (2010). An analysis of power consumption in a smartphone. In *Proceedings of the 2010 USENIX Annual Technical Conference*, páginas 1–12, Boston, MA, USA.
- Coronato, A. e Pietro, G. D.** (2008). Mipeg: A middleware infrastructure for pervasive grids. *Future Generation Computer Systems*, 24(1):17–29.
- D’Andria, F., Martrat, J., Jiménez, S., Wesner, S., Ritrovato, P. e Laria, G.** (2008). The akogrimo mobile grid framework as enabling technology for next generation elearning paradigms. In *The Learning Grid Handbook*, volume 2 of *Future of Learning*, páginas 125–142. Ios Press.
- Foster, I., Kesselman, C., e Tuecke, S.** (2001). The anatomy of the grid: Enabling scalable virtual organizations. *The International Journal of High Performance Computing Applications*, 15:200–222.
- Furthmüller, J. e Waldhorst, O.** (2010). A survey on grid computing on mobile consumer devices. In *Handbook of Research on P2P e Grid Systems for Service-Oriented Computing*, capítulo 13, páginas 313–337. IGI-Global, Hershey, PA.
- Furthmüller, J. e Waldhorst, O.** (2012). Energy-aware resource sharing with mobile devices. *Computer Networks*, 56(7):1920–1934.
- Ghosh, P. e Das, S. K.** (2010). Mobility-aware cost-efficient job scheduling for single-class grid jobs in a generic mobile grid architecture. *Future Generation Computer Systems*, 26(8):1356–1367.
- Jain, R.** (1991). *The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, e modeling*. Wiley professional computing. Wiley.
- Katsaros, K. e Polyzos, G. C.** (2007). Optimizing operation of a hierarchical campus-wide mobile grid for intermittent wireless connectivity. In *Proceedings of the 15 IEEE Workshop on Local e Metropolitan Area Networks*, páginas 111–116, New York, NY, USA. IEEE Press.
- Kim, K. H., Buyya, R. e Kim, J.** (2007). Power aware scheduling of bag-of-tasks applications with deadline constraints on dvs-enabled clusters. In *Proceedings of the Seventh IEEE International Symposium on Cluster Computing e the Grid, CCGRID ’07*, páginas 541–548, Washington, DC, USA. IEEE Computer Society.
- Li, C. e Li, L.** (2010). Energy constrained resource allocation optimization for mobile grids. *Journal of Parallel e Distributed Computing*, 70(3):245–258.

- Li, C. e Li, L.** (2012). Collaboration among mobile agents for efficient energy allocation in mobile grid. *Information Systems Frontiers*, 14(3):711–723.
- Lima, L. d. S.** (2007). *Um Protocolo para Descoberta e Seleção de Recursos em Grades Móveis Ad hoc*. PhD thesis, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro.
- Ma, Y., Gong, B., Sugihara, R., e Gupta, R.** (2012). Energy-efficient deadline scheduling for heterogeneous systems. *Journal of Parallel e Distributed Computing*, 72(12):1725–1740.
- Martello, S. e Toth, P.** (1984). Worst-case analysis of greedy algorithms for the subset-sum problem. *Mathematical programming*, 28(2):198–205.
- Martello, S. e Toth, P.** (1990). *Knapsack problems: algorithms e computer implementations*. John Wiley & Sons, Inc., New York, NY, USA.
- Rodriguez, J. M., Mateos, C. e Zunino, A.** (2012a). Are smartphones really useful for scientific computing? In *Advances in New Technologies, Interactive Interfaces e Communicability*, páginas 38–47. Springer.
- Rodriguez, J. M., Mateos, C. e Zunino, A.** (2012b). Energy-efficient job stealing for cpu-intensive processing in mobile devices. *Computing*, páginas 1–31.
- Rodriguez, J. M., Zunino, A. e Campo, M.** (2010). Mobile grid seas: simple energy-aware scheduler. In *Proc. 3rd High-Performance Computing Symposium-39th JAIIO*.
- Rodriguez, J. M., Zunino, A. e Campo, M. R.** (2011). Introducing mobile devices into grid systems: a survey. *International Journal of Web e Grid Services*, 7(1):1–40.
- Viswanathan, H., Chen, B. e Pompili, D.** (2012a). Research challenges in computation, communication, e context awareness for ubiquitous healthcare. *IEEE Communications Magazine*, 50(5):92–99.
- Viswanathan, H., Lee, E. K. e Pompili, D.** (2012b). Mobile grid computing for data-e patient-centric ubiquitous healthcare. In *Enabling Technologies for Smartphone e Internet of Things (ETSIoT), 2012 First IEEE Workshop on*, páginas 36–41. IEEE.
- Wu, T., Yeh, K., Chen, R., Chen, C. C. e Chen, Y. C.** (2011). Radio frequency identification (rfid) with mobile-grid based framework for farmland irrigation facility management. *African Journal of Agricultural Research*, 6(31):6499–6512.