Adaptação à variabilidade de hardware para eficiência energética na Internet das Coisas

Patrick de C. T. R. Ferreira[1], **Lucas F. Wanner**[2]

[1]Faculdade de Engenharia Elétrica e de Computação
Universidade Estadual de Campinas (UNICAMP) – Campinas – SP – Brasil

[2]Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
Campinas – SP – Brasil

Resumo. *Considerando-se projetos de circuitos com transistores compostos por quantidades contáveis de átomos, pequenas variações no processo de construção podem acarretar em gastos energéticos da ordem de 10 vezes maiores para um mesmo processamento. Sistemas de software que identificam características do hardware e se adaptam ao contexto de execução amortizam gastos desnecessários de energia. O objetivo deste projeto foi aumentar a eficiência energética dos dispositivos de IoT ao criar um sistema capaz de se reorganizar e escalonar suas tarefas, além de redistribuí-las em rede para serem executadas nos nós mais eficientes no consumo de recursos. Ganhos nos testes considerados alcançaram 24% em prolongamento de autonomia.*

1. Introdução

Em decorrência da evolução dos processos de manufatura dos chips de silício em escalas nanométricas e cada vez menores, os substratos de computação e memória estão apresentando cada vez mais variações, especialmente sob o ponto de vista energético[Borkar et al. 2003]. Considerando-se projetos com transistores compostos por quantidades contáveis de átomos, pequenas variações no processo de construção podem acarretar em gastos energéticos da ordem de 10 vezes maiores para um mesmo processamento. Sistemas de software que identificam características do hardware e tomam medidas para se adaptar ao contexto de execução amortizam gastos desnecessários de energia. Esta economia é crucial para a continuidade da execução de sistemas embarcados e/ou utilizados em aplicações de Internet das Coisas (Internet of Things - IoT), onde os recursos energéticos são limitados. Uma forma de realizar esta ação consiste em escalonar as tarefas eficientemente, postergando serviços de baixa urgência e executando todos em apenas um período de atividade da CPU.

A premissa desta proposta é que variações físicas do hardware devem ser expostas às diversas camadas de software, que por sua vez poderá monitorar, reagir, e adaptar-se para economizar energia ou aumentar a qualidade de serviço. O objetivo deste projeto foi, portanto, aumentar a eficiência energética dos dispositivos de IoT ao criar um sistema capaz de se reorganizar e escalonar suas tarefas, além de redistribuí-las em rede para serem executadas em nós com características de menor consumo de recursos. Os ganhos nos testes considerados alcançaram 24% em prolongamento de autonomia.

2. Trabalhos Relacionados

O sistema VaRTOS exposto em [Martin et al. 2014] apresenta estudos referentes aos efeitos da variabilidade de hardware em diferentes âmbitos, como o comportamento do dis-

positivo quando este é submetido a diferentes temperaturas ou a variação de consumo apresentada por diferentes microcontroladores, a qual é diferente até mesmo em função do *power mode* ao qual eles estão submetidos. No entanto, não foram exploradas possibilidades de migração de tarefas para diferentes nós da rede e nem a variação de consumo em função dos diferentes periféricos de cada placa. Utilizando a premissa de adaptação à variabilidade de hardware de VaRTOS, exploraremos os métodos de economia ainda não avaliados, visando obter técnicas mais aprimoradas para a economia de energia em sistemas embarcados para IoT.

3. Arquitetura do Sistema

A programação de microcontroladores requer, tipicamente, uma extensa revisão de configuração de hardware para manipulação de periféricos, interrupções, modos de desempenho, consumo de energia e afins. O projeto EPOS[LISHA 2010] - do inglês, *Embedded Parallel Operating System* - promove uma série de abstrações de hardware, acelerando o processo de desenvolvimento, facilitando a migração de código por não necessitar reescrita ao trocar o hardware e evitando possíveis atrasos de desenvolvimento pelos erros inerentes à interação com registradores do baixo nível. Assim, foi possível desenvolver uma metodologia para otimização do consumo de energia baseando-se na oportunidade ainda pouco explorada sobre a variabilidade de hardware, de forma que as técnicas resultantes fossem generalistas e de fácil aplicações para outros projetos. O EPOS é o sistema operacional sendo executado no microcontrolador EPOS Mote III (figura 1), fornecido pelos seus mesmos desenvolvedores e utilizado neste trabalho como plataforma base.

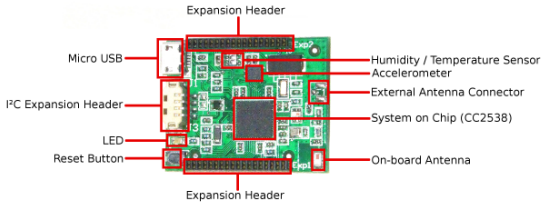

Figura 1. Exemplar do eposmote III

As técnicas para controle de potência consumida são implementadas geralmente sob a denominação de Modos de Consumo ou, mais recorrente na literatura, *Power Modes*. Estas consistem, basicamente, em um conjunto de definições para o hardware nas quais é possível selecionar a faixa de consumo de determinados componentes por meio de limitação de desempenho (diminuindo o *clock*, por exemplo), suspensão de sua atividade ou mesmo desligamento de todo um bloco do sistema. Pode-se colocar a estrutura central do microcontrolador (seu processador ou CPU, *Central Processing Unit*) em um diferente *Power Mode* dos demais periféricos, mas neste experimento eles são sempre comutados em conjunto, pois não há demanda de continuar atuando os periféricos durante o repouso da CPU.Chamamos o *Power Mode* cujo mote está totalmente operante de **Full**, modo de economia com rápida retomada de atividades de **Sleep** e alta economia com demanda de tempo para reinicialização de **Deep Sleep**.

4. Métodos

No caso trabalhado, todas as placas desenvolvidas pelo projeto EPOS (denominadas EPOS Mote) possuem um transceptor de rádio de baixo consumo, além de vários outros sensores já embarcados nas mesmas. Os dispositivos começam a execução com a mesma carga em suas respectivas baterias, que são individuais e idênticas, consistindo em *packs* de 2600mAh. Em cada dispositivo são coletados dados de temperatura e umidade ambiente e feitos alguns processamentos de correlação e análise, os quais não serão abordados a fundo, pois o interesse neste caso é apenas no fato de que há uma carga de processamento a ser realizada e que a mesma é igual em todos os nós da rede. Conforme são geradas amostras, o consumo em cada mote terá sido diferente, devido à variabilidade do hardware. Os motes, que medem independentemente o nível de suas baterias através de seus ADCs (*Analog Digital Converter*), trocarão estas informações entre si por rádio e verificarão se a discrepância do gasto energético resultante compensa uma transferência de carga computacional entre eles e, em caso afirmativo, enviarão seus dados coletados ao nó da rede cuja eficiência energética se mostrou maior. Os nós (motes) que enviaram os dados entram em *Deep Sleep* até a próxima coleta e compartilhamento de dados, e o nó que recebeu o faz depois de processá-los.

Um último microcontrolador, que não está sendo analisado neste teste, monitora a comunicação e fica responsável por verificar qual dos motes está sendo trabalhado e como está se dando a velocidade de descarga do sistema em função dos motes e do tempo.

5. Resultados e Análise

Tabela 1. Consumo médio de corrente dos Motes sob os diferentes Power Modes.

Power Mode	Corrente [mA]	Potência Elétrica [mW]
Full	12.0	60.0
Sleep	1.2	6.0
Deep Sleep	0.5	2.5

O consumo energético verificado para cada microcontrolador em modo *full* foi, em média, 10 vezes maior que que em modo *sleep* e 24 vezes maior que em modo *deep sleep*, explícito na tabela 1. O objetivo agora é de verificar quanto a metodologia proposta é capaz de salvar em energia em comparação com um sistema que executasse as mesmas tarefas sem levar em conta a variabilidade de hardware. A fim de se obter os dados das amostras de **controle**, instalou-se uma programação que consistia em cada placa amostrar periodicamente dados de temperatura e umidade, cada uma em seu ambiente, e realizar em seu próprio processador os cálculos de correlação e análise, deslocando - através do rádio - o processamento das amostras para o mote com mais carga na bateria quando ultrapassasse um limiar de diferença entre estas. Os testes foram repetidos algumas vezes e os resultados das médias de autonomia encontradas para cada microcontrolador estão expressos na Tabela 2. Note que foram dados nomes às placas de acordo com seu melhor desempenho, o que foi escolhido após a coleta dos resultados deste teste.

O algoritmo já descrito na seção de Métodos foi executado até que um dos dispositivos ficasse sem carga, quando não é mais possível traçar qualquer correlação entre os dados de temperatura e umidade entre os ambientes dos três motes. Os resultados estão

Tabela 2. Duração média de cada um dos três microcontroladores sem a otimização por variabilidade de hardware.

Mote	Autonomia [h]
Alfa	47.150
Beta	44.567
Gama	35.233

dispostos na figura 2 - onde 0 significa nenhum processamento de amostras naquele mote e 1 signifcia que só aquele dispositivo está recebendo as amostras e processando-as - e mostram que a placa Alfa processou os dados das demais a maior parte do tempo, exceto por um intervalo em que esta maior concentração de carga de processamento fez com que a placa Beta possuísse um pouco mais de energia na bateria.

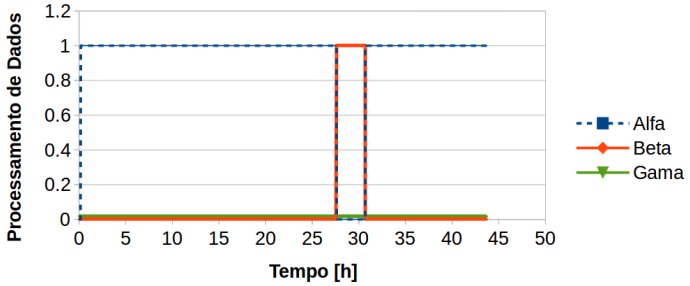

Figura 2. Liderança em processar os dados dos demais motes.

O prolongamento do sistema, que durava 35,23h sem a otimização por variabilidade de hardware, foi cerca de 8,5h, se estendendo até 43,73h com a otimização, uma melhora de mais de 24% em autonomia. Após as 43,73h, Gama desliga e não há mais como equiparar os dados dos ambientes dos três motes, objetivo do teste proposto.

6. Conclusão

Os testes demonstraram até 24% de economia para este tipo de migração de tarefas em rede. Esta técnica é, portanto, útil para casos em que os dispositivos inevitavelmente estarão o tempo todo conectados em rede, não havendo custo energético adicional para se ativar esta parte do hardware, como é o caso de dispositivos para Internet das Coisas (IoT). Entretanto, para aplicação em sistemas cujos nós não estabelecem conexão constante com a rede, deve-se ponderar se o custo energético para estabelecer esta comunicação não é maior que a própria economia gerada.

Referências

Borkar, S., Karnik, T., Narendra, S., Tschanz, J., Keshavarzi, A., and De, V. (2003). Parameter variations and impact on circuits and microarchitecture. In *Proc. Design Automation Conf. (DAC)*, pages 338–342.

LISHA (2010). Embedded Parallel Operating System - Eposmote. http://epos.lisha.ufsc.br/EPOSMote+III. Acessado em: 30-10-2018.

Martin, P., Wanner, L., and Srivastava, M. (2014). Runtime Optimization of System Utility with Variable Hardware. *ACM Transactions on Embedded Computing Systems*, 14(2):24:1—-24:25.