AVALIANDO A FAIXA DE FREQUÊNCIA DE OPERAÇÃO DA GPU MALI-T628 NO MPSoC EXYNOS 5422

DOUGLAS WAHAST DA COSTA; MATEUS SANTOS DE MELO; CARLOS MICHEL BETEMPS; BRUNO ZATT

Universidade Federal de Pelotas – {dwdcosta, msdmelo, cm.betemps, zatt}@inf.ufpel.edu.br

1. INTRODUÇÃO

Com a crescente utilização de dispositivos contendo MPSoCs (*Multi processor System-on-Chip*), principalmente na categoria de Sistemas Embarcados Heterogêneos (SEH), temos a possibilidade de obtermos grande flexibilidade na execução das cargas de trabalho em virtude dos distintos Elementos de Processamento (EPs) (como, por exemplo, CPU, GPU, GPGPU, FPGA e DSP) - tipicamente contendo CPU + GPU nos MPSoCs atuais (SINGH, 2017). Evidenciada a presença de distintos EPs para a execução das aplicações, é possível a exploração de variáveis como energia e latência em função de cada aplicação quando executada em diferentes conjuntos de EPs.

Diante de cenários nos quais é necessário a abrangência entre as diversas categorias de aplicações, os benchmarks são indispensáveis. Os benchmarks são conhecidos tipicamente pelo emprego na avaliação de hardware, onde podem analisar as diferentes combinações de EPs. O benchmark Polybench/GPU (GRAUER-GRAY et al., 2012) inclui 15 aplicações para as EPs CPU e GPU implementadas em CUDA, OpenCL, e HMPP.

Avaliações passadas (COSTA et al., 2019) sobre a plataforma Odroid-XU3, empregando aplicações do Polybench, trouxeram resultados sobre valores de tempo e energia para variadas frequências de operação sobre as CPUs (A15 e A7 - arquitetura *big-LITTLE*), porém com frequência única para a GPU. Neste trabalho foram avaliadas as execuções com mesmo objetivo para diferentes frequências de operação para a GPU Mali-T628 existente nesta plataforma.

O cluster-A15 apresenta os melhores valores relacionados a tempo de execução em frequência máxima, mas com alto consumo de energia (COSTA et al., 2019). Sendo que a GPU apresenta bons resultados em tempo e com reduzido consumo de energia para maioria das aplicações. Logo, o estudo sobre as diversas frequências de operação da GPU se torna essencial para entendimento do comportamento deste tipo de EP na execução das aplicações sobre essas diferentes configurações.

2. METODOLOGIA

Para execução e estudo das aplicações neste projeto foi utilizada a plataforma Odroid-XU3 (HardKernel, 2018), a qual contém o MPSoC Exynos 5422 classificado como SEH com 3 categorias de processadores. A categoria de CPU contém dois *cluster Quad-Core*, um *Cortex-A15* de alto desempenho e um *Cortex-A7* voltado para economia de energia, para categoria de GPU há disponível uma *ARM Mali T628 MP6*.

Diante do Sistema Heterogêneo apresentado, a execução das aplicações entre os diferentes tipos e modelos de dispositivos se tornaria inconveniente pelo fato da necessidade da implementação de cada carga de trabalho para cada dispositivo. Com isso, para este estudo, foram utilizadas aplicações

implementadas em OpenCL (KAELI et al., 2015) e C encontradas no benchmark Polybench/GPU, trazendo portabilidade para o código entre os diferentes EPs que possuem suporte ao OCL.

O benchmark Polybench/GPU traz 15 aplicações divididas em 4 categorias (*Convolution*, *Linear Algebra*, *Stencils*, *Datamining*), possibilitando, através dessa categorização, uma análise sobre os tipos de aplicações, fazendo parte destas categorias operações como multiplicação de matrizes de duas e três dimensões, como também operações de convolução, ambas as operações são muito presentes em processamento de imagens e vídeo digitais.

Inicialmente as aplicações foram divididas em "*Tasks*" através da geração de um grafo de fluxo de execução, buscando similaridades (bastante presentes por parte do modelo de programação da Linguagem OpenCL), para assim obter uma visualização da representatividade de cada *Task* no processo de execução (Betemps, 2018).

Para obtenção dos dados de latência de cada *Task* das aplicações (*Host* limitado a CPU pela arquitetura), foi feita uma instrumentalização do código das aplicações do Polybench para geração de *Logs*, utilizando a biblioteca "time.h" da linguagem C, e em específico a função "rtclock". Para obtenção dos valores de potência foi utilizado um *script* o qual retorna a energia acumulada de cada dispositivo (CPUs, GPU, MEM) durante o processo de execução, junto do tempo total da execução da aplicação, obtido através do comando TIME do sistema Linux. O *script* de obtenção de dados de energia acessa o sistema de arquivos virtuais do Linux para obter dados dos sensores existentes na plataforma Odroid-XU3 (HardKernel, 2018), já tendo sido utilizado anteriormente (COSTA et al., 2019).

No primeiro conjunto de testes do experimento foram feitas execuções sobre os EPs disponíveis (A15, A7 e GPU) para as diversas frequências de operação e configurações de *Device* e *Host*, seguindo os padrões do OpenCL, onde a GPU se manteve com frequência máxima de 0.6GHz (COSTA et al., 2019).

As configurações de frequência adotadas para este experimento estão presentes na Tabela 1. Para as execuções de teste da GPU, a frequência do processador em *Idle* (A7) e daquele usado como *Host* (A15) ficaram fixas em 1GHz, sendo uma frequência mediana, apresentando uma eficiência intermediária sobre as variáveis de energia e latência. Todas as execuções de avaliação do experimento foram feitas 30 vezes para contornar problemas de desvio padrão.

Tabela 1. Configurações de Frequência.

Device@Freq (GHz)	Host@Freq (GHz)	Idle@Freq (GHz)	Device@Freq (GHz)	Host@Freq (GHz)	Idle@Freq (GHz)
A15@1.000	A15@1.0	A7@1.0	**Mali@0.420**	A15@1.0	A7@1.0
Mali@0.177	A15@1.0	A7@1.0	**Mali@0.543**	A15@1.0	A7@1.0
Mali@0.266	A15@1.0	A7@1.0	**Mali@0.600**	A15@1.0	A7@1.0
Mali@0.350	A15@1.0	A7@1.0	-	-	-

3. RESULTADOS E DISCUSSÃO

A Figura 1 apresenta o consumo energético e de tempo no processo de execução de cada aplicação, sobre cada configuração de frequência da Mali T628, considerando a frequência base para os *cluster* de CPU de 1GHz.

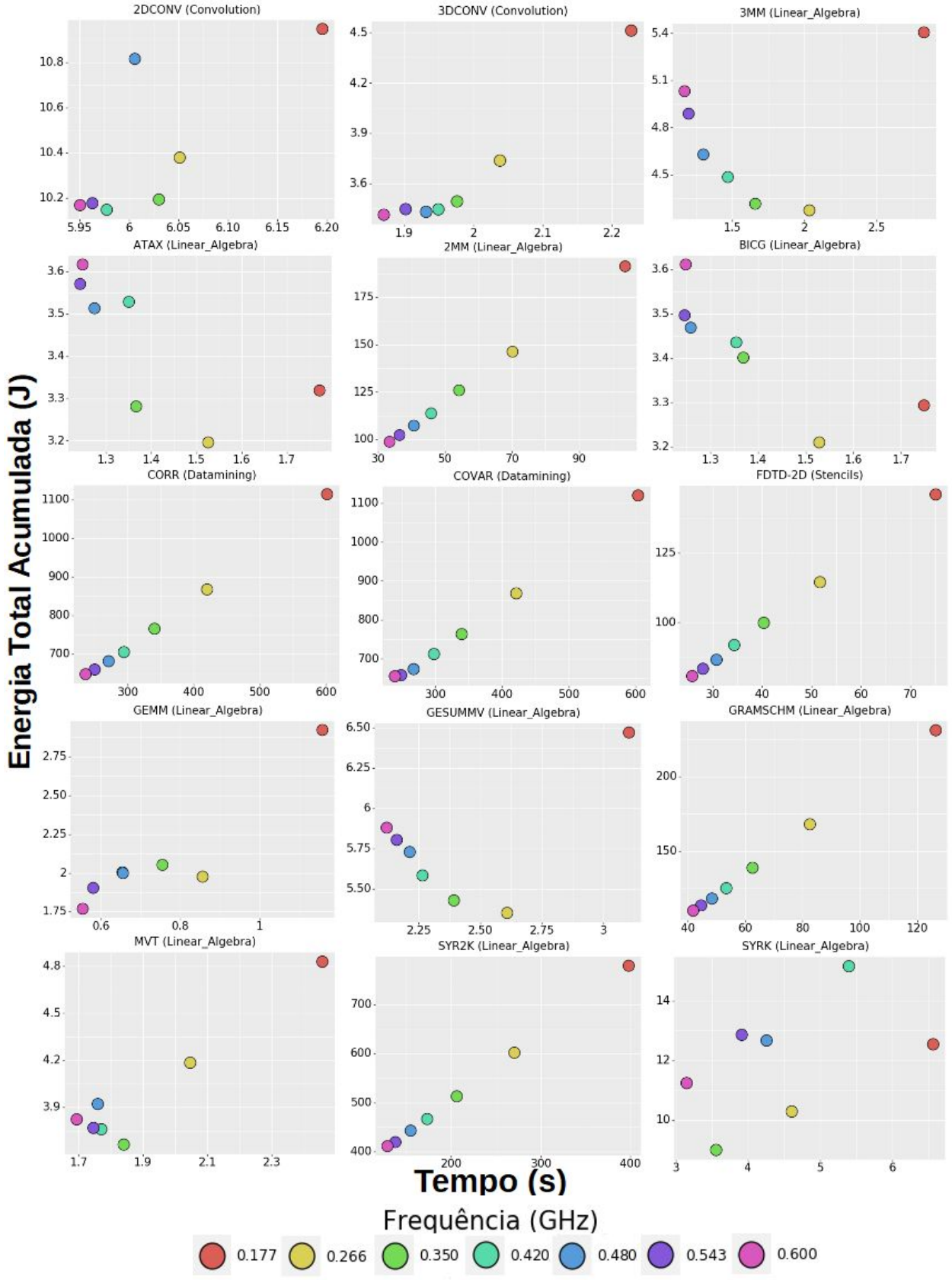

Figura 1. Energia total acumulada por Tempo de execução de cada aplicação nas diferentes frequências de operação da GPU.

Os resultados mais próximos do canto inferior esquerdo de cada gráfico na Figura 1 apresentam uma execução mais rápida e menor energia consumida, sendo essas as configurações a serem escolhidas para obter-se melhor eficiência na execução das aplicações.

Sobre esses resultados vemos que a frequência de 0.177GHz retorna o pior tempo de execução para todas as aplicações, e o maior consumo de energia para 12 das 15 aplicações avaliadas. Para as aplicações ATAX, BICG, SYRK foram obtidos resultados intermediários sobre o consumo de energia, comparada sobre a execução em outras frequências.

Para a maioria das aplicações observadas a frequência máxima (0.6GHz) obtém os melhores resultados em tempo e energia, mas para alguns casos a configuração dos resultados apresenta uma característica a qual temos um limite onde não é possível diminuir a latência da execução sem ter um aumento do consumo de energia. Assim, sendo do usuário a escolha da configuração, a frequência máxima de operação propicia bons resultados em tempo para a maioria das aplicações, mas para algumas delas (ATAX, BICG, 3MM, GESUMMV, SYRK) isso vem acompanhado de um elevado consumo de energia.

4. CONCLUSÕES

Considerando que a GPU obtém bons resultados para o tempo de execução da maioria das aplicações da categoria *Linear Algebra* operando em frequência máxima (COSTA et al., 2019) é possível avaliar através deste trabalho que, para se obter bons resultados de tempo e energia consumida durante a execução das aplicações, é necessário para maior eficiência a utilização de frequências superiores a 266MHz para a GPU Mali T628. Assim como também é evidenciado que a utilização de frequências de operação menores não trazem economia de energia para a execução da aplicação, pelo fato de tomarem mais tempo do processamento do conjunto de EPs responsável pela execução.

5. REFERÊNCIAS BIBLIOGRÁFICAS

COSTA, Douglas W. et al. Characterizing the Polybench/GPU on an MPSoC. **34rd South Symposium on Microelectronics**, Pelotas, ISSN 2177-5176, p.91-94, 2019.

BETEMPS, C. M. et al. Exploring Heterogeneous Task-Level Parallelism in a BMA Video Coding Application using System-Level Simulation. **Computing Systems Engineering (SBESC), 2018 VIII Brazilian Symposium on IEEE**, p.74–81, 2018.

HardKernel. **Odroid-XU3**. Hardkernel co., 2018. Acessado em 05 set. 2018. Online.URL:http://www.hardkernel.com/main/products/prdt_info.php?g_code=g140448267127

KAELI, David R. et al. **Heterogeneous computing with OpenCL 2.0**. Morgan Kaufmann, 2015.

PolyBench/GPU. **Implementation of PolyBench codes for GPU processing**. *University of Delaware*, 2012. Acessado em 10 set. 2019. Online. Disponível em: http://web.cse.ohio-state.edu/~pouchet.2/software/polybench/GPU/index.html

SINGH, Amit K. et al. Energy-Efficient Run-Time Mapping and Thread Partitioning of Concurrent OpenCL Applications on CPU-GPU MPSoCs. **ACM Transactions on Embedded Computing Systems** (TECS), v. 16, n. 5s, p. 147, 2017.

GRAUER-GRAY S., “Auto-tuning a high-level language targeted to gpu codes,” in **2012 Innovative Parallel Computing (InPar)**, pp. 1–10, 2012.