UNIVERSIDADE FEDERAL DO PAMPA

Everton Camargo de Lima

Otimizando a Execução de Aplicações Paralelas em Ambiente de Nuvem Heterogênea

Alegrete
Maio de 2023

Everton Camargo de Lima

Otimizando a Execução de Aplicações Paralelas em Ambiente de Nuvem Heterogênea

Dissertação de tese apresentada ao Programa de Pós-graduação Stricto Sensu em Engenharia Elétrica da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Arthur Francisco Lorenzon
Coorientador: Prof. Dr. Fábio Diniz Rossi

Alegrete
Maio de 2023

Ficha catalográfica elaborada automaticamente com os dados fornecidos
pelo(a) autor(a) através do Módulo de Biblioteca do
Sistema GURI (Gestão Unificada de Recursos Institucionais) .

L732o Lima, Everton Camargo de
Otimizando a Execução de Aplicações Paralelas em Ambiente de Nuvem Heterogênea / Everton Camargo de Lima.
100 p.

Dissertação(Mestrado)-- Universidade Federal do Pampa, MESTRADO EM ENGENHARIA ELÉTRICA, 2023.
"Orientação: Arthur Franciso Lorenzon ".

1. Computação em Nuvem. 2. Computação de alto desempenho. 3. Aplicação Paralela. 4. Otimização de Sistemas. 5. Cluster heterogêneo. I. Título.

EVERTON CAMARGO DE LIMA

OTIMIZANDO A EXECUÇÃO DE APLICAÇÕES PARALELAS EM AMBIENTE DE NUVEM HETEROGÊNEA

Dissertação/Tese apresentada ao Programa de Pós–Graduação em Engenharia Elétrica da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Mestre em Engenharia Elétrica.

Dissertação defendida e aprovada em: 04/05/2023

Banca examinadora:

_____

Prof. Dr. Arthur Francisco Lorenzon

Orientador

UFRGS

_____

Prof. Dr. Marcelo Caggiani Luizelli

Unipampa

_____

Prof. Dr. Edson Luiz Padoin

UNIJUI

Assinado eletronicamente por **Arthur Francisco Lorenzon**, **Usuário Externo**, em 15/05/2023, às 13:55, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **EDSON LUIZ PADOIN**, **Usuário Externo**, em 17/05/2023, às 11:06, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **MARCELO CAGGIANI LUIZELLI**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 17/05/2023, às 13:44, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1129410** e o código CRC **8F10D0EA**.

À minha Família e Amigos e
À memória da minha querida amiga Carol Silva.

AGRADECIMENTOS

Primeiramente agradeço a Deus e minha querida família, agradeço por todo amor, apoio e suporte incondicional que vocês me deram em cada etapa da minha vida. Vocês são minha base e inspiração, e sou muito grato por tê-los em minha vida. Agradeço ao orientador desta dissertação Arthur Lorenzon, que esteve comigo nesta caminhada, ao meu coorientador Fábio Rossi ao qual me apresentou o PPGEE, e sem a ajuda de vocês, eu não teria chegado até aqui. Agradeço aos amigos, cada um de vocês me apoiou com palavras de encorajamento, conselhos sábios, críticas construtivas e gestos de carinho e amizade. Através de nossas conversas, trocas de ideias, e mensagens, eu fui inspirado e motivado a trabalhar duro e superar desafios. Enfim, sou profundamente grato a cada um de vocês por todo apoio, incentivo e carinho que recebi durante esses dois anos de mestrado. Foi uma jornada desafiadora, mas também foi uma das mais recompensadoras da minha vida, graças à companhia de vocês. Espero poder continuar nossa amizade e parceria no futuro.

"Os grandes feitos são conseguidos não pela força, mas pela perseverança- Samuel Johnson

RESUMO

A computação na nuvem é uma plataforma já consolidada para a execução de aplicações de alto desempenho devido entre outros fatores por sua capacidade de escalabilidade de recursos e alta disponibilidade. Simultaneamente, a atualização de nodos computacionais nestes sistemas pode levar a uma heterogeneidade de recursos. Neste sentido, o desafio de executar aplicações paralelas na nuvem não está apenas relacionado a definição do melhor número de *threads* para a aplicação, mas também, a escolha ideal da arquitetura que irá executar tal aplicação. Dessa maneira, nesta dissertação, apresentamos o *TLP-Allocator*, um framwerok para otimizar a execução de aplicações paralelas em ambiente de nuvem heterogêneo baseada no grau de TLP + arquitetura alvo ideal, implementada em cima da tecnologia *Kubernetes*. O *TLP-Allocator* é completamente transparente para usuários finais e clientes. Dado um conjunto de aplicações paralelas a serem executados, ele encontra grau de TLP ideal para cada aplicação e realiza *match* entre aplicação e arquitetura ideal, para melhor aproveitamento dos recursos disponíveis. Mostramos que a análise do impacto da escalabilidade de aplicações paralelas em nodos com diferentes capacidades de poder de processamento computacional pode trazer ganhos significativos de desempenho e consumo de energia quando combinada com o *TLP-Allocator*. Ao considerar o grau de paralelismo de uma aplicação e as características do nodo computacional, é possível obter uma alocação mais eficiente e otimizada dos recursos disponíveis assim alcançar o melhor EDP (*Energy-Delay product*) para aplicações com diferentes graus de paralelismo.

Palavras-chave: Computação em Nuvem, Computação de alto desempenho, Aplicação paralela, *Cluster* heterogêneo, Otimização de Sistemas, grau de TLP, desempenho, consumo de energia e EDP.

ABSTRACT

Cloud computing is already a well-established platform for high-performance application execution, among other factors, due to its resource scalability and high availability. At the same time, updating computational nodes in these systems can lead to resource heterogeneity. In this sense, the challenge of executing parallel applications in the cloud is not only related to defining the best number of threads for the application, but also to choosing the ideal architecture that will execute such application. Therefore, in this dissertation, we present the TLP-Allocator, a framework to optimize the execution of parallel applications in a heterogeneous cloud environment based on the ideal TLP degree + target architecture, implemented on top of Kubernetes technology. The TLP-Allocator is completely transparent to end-users and clients. Given a set of parallel applications to be executed, it finds the ideal TLP degree for each application and performs a match between application and ideal architecture to better utilize the available resources. We showed that analyzing the scalability impact of parallel applications on nodes with different computational processing power capabilities can bring significant gains in performance and energy consumption when combined with the TLP-Allocator. By considering the degree of parallelism of an application and the characteristics of the computational node, it is possible to obtain a more efficient and optimized allocation of the available resources, thus achieving the best EDP (Energy-Delay product) for applications with different degrees of parallelism.

Keywords: *Cloud Computing, High Performance Computing, Parallel Application, Heterogeneous Cluster, System Optimization, TLP degree, Performance, Energy Consumption, and EDP.*

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

EDP	Energy Delay Product
HPC	High-Performance Computing
TLP	Thread-Level Parallelism
CPU	Central Processing Unit
EDP	Power–delay product
I/O	Input/Output
API	Application Programming Interface
DVFS	Dynamic Voltage and Frequency Scaling
DNS	Domain Name System
IPC	Instructions Per Cycle
UMA	Uniform Memory Access
NUMA	Non-Uniform Memory Access
SMT	Simultaneous Multithreading
OpenMP	Open Multi-processing
RNAs	Redes neurais artificiais

SUMÁRIO

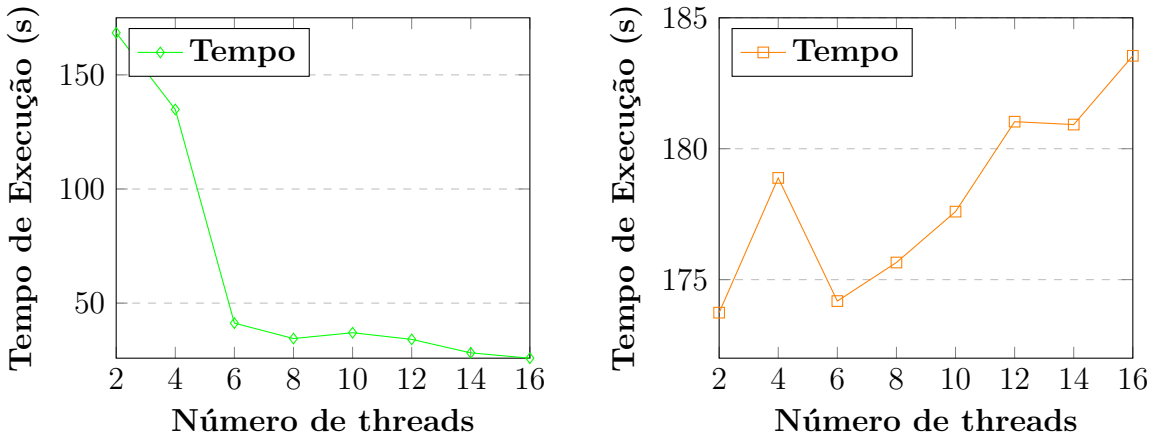
1 INTRODUÇÃO

A computação em nuvem se consolidou com uma plataforma alternativa para execução de aplicações de alto desempenho de diferentes domínios, como por exemplo, aprendizado de máquina e métodos de álgebra linear básica. No entanto, embora estes sistemas tenham capacidade de fornecer elasticidade (provisionamento de recursos sob demanda), customização e controle de recursos, eles são essencialmente *data centers* com uma alta exigência de energia/potência para manter a operação (MASANET et al., 2020). Portanto, o desafio de executar aplicações paralelas na nuvem não passa apenas pela otimização de desempenho, mas também pela melhor utilização dos recursos computacionais para reduzir o consumo de energia, mantendo baixo os custos de operação do provedor.

Para extrair o máximo de desempenho das aplicações, o paralelismo no nível de *threads* (TLP - *thread-level parallelism*) tem sido explorado pelos desenvolvedores de *software*. Quando isso é feito, a carga de trabalho da aplicação é dividida em tarefas menores, que são executadas de maneira concorrente por múltiplas unidades funcionais. Normalmente, a maneira padrão adotada por usuários e desenvolvedores consiste no uso de todos os recursos computacionais disponíveis na arquitetura (e.g., núcleos e memória). No entanto, uma vez que aplicações paralelas podem ter sua escalabilidade limitada (isto é, o aumento no número de threads não se traduz em melhorias de desempenho) por questões relacionadas à *software* e/ou *hardware*, atribuir todos os recursos para tais aplicações não irá necessariamente resultar no melhor desempenho e consumo de energia. As principais causas discutidas na literatura são: saturação de barramento de comunicação entre processador e memória principal, sobrecarga de sincronização de dados entre *threads* e número de acessos concorrentes à memória compartilhada (SULEMAN; QURESHI; PATT, 2008). Neste sentido, definir o número ideal de *threads* para executar aplicações paralelas pode levar a uma redução significativa no tempo de execução, diminuindo os custos relacionados a energia e alocação de recursos computacionais como podemos observar na Figura 2 para duas aplicações distintas.

Conforme observado na Figura 2(a), para a aplicação ***Poisson*** (PO), o tempo de execução diminui de acordo com o aumento no número de *threads*, indicando que o melhor desempenho é atingido com o número máximo de *threads*. Por outro lado, a Figura 2(b) demonstra um comportamento em que o número ideal de *threads* para executar a aplicação ***STREAM*** (ST) é de apenas 2 devido a questões de saturação do barramento *off-chip*, conforme discutido no Capítulo 2. Neste sentido, aumentar o número de *threads* além deste ponto não reflete em melhorias de desempenho, impactando negativamente na energia consumida para executar a aplicação.

Adicionalmente, durante o ciclo de vida de um centro de dados, os *clusters* (*i.e.*, agregados de computadores) podem ser atualizados com nós computacionais que empregam tecnologias mais recentes e geralmente apresentam capacidade de processamento superior em comparação aos nós iniciais. Esta atualização resultará em um *cluster* heterogêneo,



(a) Aplicação *Poisson (PO)* (b) Aplicação *Stream (ST)*

Figura 1 – Resultado da escalabilidade das Aplicações PO e ST.

onde cada nodo é capaz de entregar capacidade computacional em termos de desempenho e consumo de energia diferente de acordo com a aplicação alvo. Esta heterogeneidade no nível de recursos resulta em novos desafios relacionados a execução de aplicações paralelas na nuvem: *além de definir o melhor número de threads no nível da aplicação, também é importante alocar esta aplicação para o nodo computacional que forneça os melhores resultados de desempenho e consumo de energia.*

Neste sentido, enquanto um *cluster* homogêneo é capaz de entregar desempenho similar para uma aplicação independente da arquitetura onde a mesma será executada, o mesmo não é verdade em ambientes heterogêneos. A principal razão para este comportamento está na características de escalabilidade das aplicações e das características intrínsecas de cada arquitetura (e.g., hierarquia de memória, número de núcleos e frequência de operação). No cenário em que uma aplicação totalmente paralela que escala de acordo com o número de *threads*, o ideal é a alocação desta em nodos de trabalho com maior número de núcleos de processamento. Por outro lado, em cenários onde a aplicação tem escalabilidade limitada por alguma das razões discutidas no Capítulo 2, alocar ela em nodos com menor número de núcleos não apresentará impacto negativo no seu desempenho.Um exemplo destes cenários é destacado na Figura 2 para a execução da aplicação BT-NAS em três arquiteturas com poder de processamento diferente: *AMD-16* com 16 núcleos, *AMD-24* com 24, e *AMD-64* com 64 núcleos de processamento. Conforme observado na Figura 2(a), o melhor desempenho é atingido na arquitetura AMD-24; o menor consumo de energia na arquitetura AMD-16 (Figura 2(b)); e o melhor custo-benefício entre desempenho e energia (representado pela métrica EDP – *Energy-Delay Product*) é atingido no AMD-64 (Figura 2(c)). Isto mostra que a mesma aplicação possui escalabilidade diferente para diferentes nodos e que varia conforme a métrica avaliada.

Neste contexto, diferentes estratégias, como por exemplo algoritmos de aprendizado de máquina (SANGEETHA et al., 2022; BI et al., 2019; TUNCER et al., 2017), têm sido propostas para otimizar a execução de aplicações paralelas em um ambiente de nuvem

computacional, conforme discutido no Capítulo 3. Essas técnicas possibilitam encontrar a configuração ideal ou próxima da ideal para a execução de uma aplicação, considerando variáveis relevantes, tais como, número de *threads* e a definição do nodo que irá executar a aplicação. Quando comparado a outras técnicas de aprendizado de máquina (e.g., árvores de decisão e regressão linear), redes neurais fornecem uma vantagem significativa pois são capazes de lidar com dados complexos e não-lineares, além de reconhecer padrões em grandes conjuntos de dados e realizar previsões e classificações com alta precisão (GOODFELLOW; BENGIO; COURVILLE, 2016). Portanto, esta dissertação avança no estado da arte ao propor TLP *allocator*, uma abordagem que utiliza redes neurais para encontrar a melhor configuração de nodo computacional e grau de paralelismo de uma aplicação em um ambiente de nuvem heterogêneo para executar aplicações paralelas.

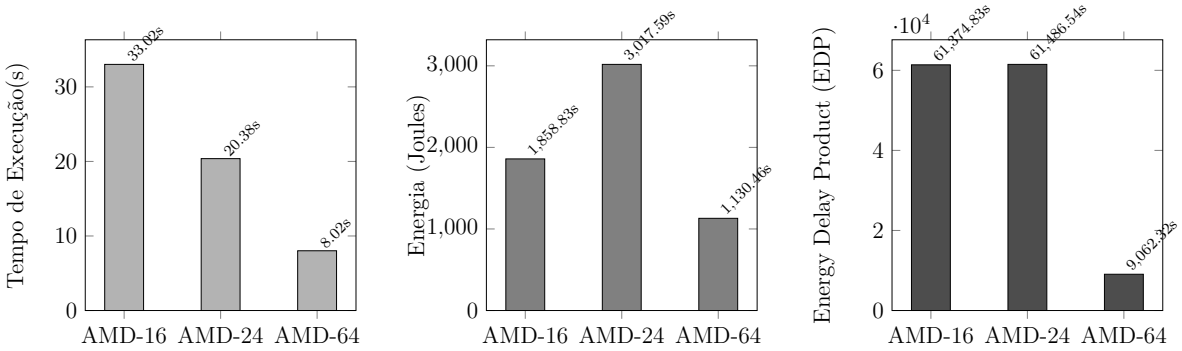


Figura 2 – Comportamento da aplicação (BT) para cada arquitetura e métrica avaliada.

1.1 OBJETIVOS

Em conformidade com o que foi destacado anteriormente, dispor de mais recursos computacionais para aplicações com baixa escalabilidade não resultará no melhor desempenho e consumo de energia. Portanto o objetivo geral desta dissertação consiste em **otimizar a execução de aplicações paralelas em ambiente de nuvem heterogênea considerando as métricas de desempenho, consumo de energia e EDP**. Assim, os objetivos específicos desta dissertação compreendem:

- Analisar o impacto no desempenho, consumo de energia e no EDP quando aplicações paralelas com diferentes graus de paralelismo são executadas em um ambiente com arquiteturas heterogêneas.

- TLP *allocator*, uma abordagem que utiliza redes neurais para encontrar a melhor configuração de nodo computacional e grau de paralelismo em um ambiente heterogêneo para executar aplicações paralelas.

1.2 ORGANIZAÇÃO DA DISSERTAÇÃO

O restante desta dissertação está organizado como segue.

No capítulo 2, é apresentado a fundamentação teórica, onde é abordado conceitos de arquitetura *multicore* e computação paralela. Descreve-se também sobre escalabilidade de aplicações paralelas e computação de alto desempenho na nuvem.

No capítulo 3, são discutidos os trabalhos relacionados a esta dissertação, onde são abordados trabalhos que otimizam o desempenho, energia e EDP de aplicações paralelas.

O Capítulo 4 é dedicado à exploração do espaço de projeto, onde é apresentada uma metodologia que inclui um conjunto de aplicações, ambiente de execução, criação do ambiente de execução e resultados experimentais. Esse capítulo é fundamental para apresentar os resultados das experimentações realizadas e como foram obtidos, além de fornecer uma base teórica para o capítulo seguinte.

O Capítulo 5, por sua vez, aborda a proposta do *framework TLP-Allocator*, que é um modelo de alocação de aplicações paralelas em ambiente de nuvem heterogêneo. Neste capítulo, são apresentadas as fases de aprendizado e fase de execução do modelo de Redes Neurais Artificiais (RNAs), bem como a metodologia de avaliação utilizada para a implementação do *TLP-Allocator*. Esse capítulo é crucial para demonstrar como foi desenvolvido o modelo proposto e como ele foi aplicado para resolver o problema de alocação de aplicações e arquitetura alvo.

Por fim, o Capítulo 6 e 7 apresentam a conclusão e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é apresentado uma introdução sobre arquiteturas *multicore*, computação paralela e computação na nuvem. Para tanto, ele começa descrevendo as arquiteturas *multicore* e computação paralela. Então, na Seção 2.2 a escalabilidade de aplicações paralelas em ambientes de alto desempenho, bem como, descreve o impacto das abordagens convencionais utilizadas por desenvolvedores de *software* paralelo. Por fim, é descrito o uso de nuvem computacional para a execução de aplicações paralelas.

2.1 ARQUITETURAS *MULTICORE* E COMPUTAÇÃO PARALELA

Arquiteturas *multicore* são projetadas para aproveitar ao máximo a capacidade de processamento dos dispositivos modernos. Elas consistem em vários núcleos (geralmente dois ou mais) em um único processador, cada um dos quais podendo ser utilizado para executar tarefas independentes de maneira simultânea. Isso permite a execução de múltiplas tarefas de forma mais eficiente do que se a arquitetura tivesse apenas um núcleo, aumentando a velocidade e a capacidade de processamento. De acordo com (PATTERSON; HENNESSY, 2016), as arquiteturas *multicore* são mais eficientes energeticamente do que as arquiteturas de processadores únicos, uma vez que permitem a desativação de núcleos quando não são necessários, reduzindo assim o consumo de energia.

As arquiteturas *multicore* podem ser classificadas de acordo com a organização de memória compartilhada (PATTERSON; HENNESSY; GOLDBERG, 1990): acesso uniforme à memória *UMA uniform memory access* e não uniforme *NUMA* (*non-uniform memory access*). Nas arquiteturas UMA, todas as unidades de processamento têm o mesmo tempo para acessar a memória principal. Por outro lado, em arquiteturas NUMA, cada núcleo pode ter um tempo diferente para acessar a memória principal (PATTERSON; HENNESSY; GOLDBERG, 1990).

No entanto, para que todos os recursos computacionais disponíveis em arquiteturas *multicore* (e.g., núcleos e memória) possam ser eficientemente utilizados, é necessário que os programas sejam escritos de forma adequada para aproveitá-los. Neste sentido, a programação paralela é empregada para dividir tarefas em tarefas menores que podem ser executadas em paralelo pelos núcleos disponíveis (PATTERSON; HENNESSY, 2016). Assim, a programação paralela tem sido fundamental para tirar proveito da capacidade das arquiteturas *multicore* e melhorar o desempenho das aplicações em computadores modernos (CULLER; SINGH; GUPTA, 1999).

Independentemente da arquitetura utilizada, a programação paralela permite trabalhar com um grande volume de dados explorando eventos simultâneos. Um dos padrões de programação mais utilizados consiste do *fork-join*, conforme ilustrado na Figura 3. Nele, inicialmente existe uma *thread* principal (chamada de *master*) que é responsável por iniciar a execução da aplicação. Quando esta *thread* atinge uma região paralela, novas *threads* são criadas (*fork*) para executar o bloco de código em paralelo. Ao finalizar a

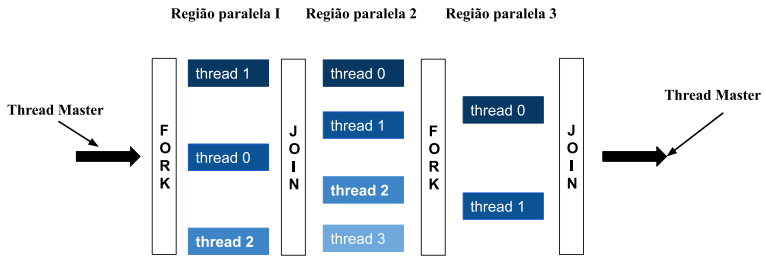


Figura 3 – Processamento de aplicação em paralelo com Modelo *Fork-Join* do OpenMP.

execução da região paralela, as *threads* são sincronizadas (*join*) e apenas a *thread* principal segue a execução da aplicação (CHAPMAN; JOST; PAS, 2007).

No entanto, para que a cooperação entre as *threads* ocorra, as mesmas precisam trocar dados em algum momento da execução. Dois modelos principais são utilizados: variáveis compartilhadas e troca de mensagens. Na comunicação por variáveis compartilhadas, todas as *threads* compartilham uma região de memória e a comunicação se dá através de operações de escrita e leitura nestas variáveis. Para garantir a coerência de dados após as operações de leitura/escrita, o acesso às variáveis é controlado por diretivas (e.g., funções) de sincronização (e.g., *mutex* e *atomic*). Por outro lado, no modelo de troca de mensagens, os processos não compartilham regiões de memória e a comunicação se dá através de operações de *send/receive*. Em se tratando de arquiteturas *multicore*, a comunicação através de variáveis compartilhadas permite a criação de códigos mais eficientes, uma vez que reduz a necessidade de operações de comunicação entre processadores (KIRK; WEN-MEI, 2016).

2.1.1 ESCALABILIDADE DE APLICAÇÕES PARALELAS

A escalabilidade de aplicações paralelas é um fator crucial para garantir que sistemas de alto desempenho possam lidar com cargas de trabalho crescentes sem perder desempenho. Assim, para garantir que uma aplicação possa aproveitar adequadamente os recursos disponíveis (*e.g., CPU, memória*) e dimensionar sua execução de forma eficiente, é necessário levar em consideração vários fatores, como a arquitetura do hardware, a distribuição de tarefas, a comunicação entre os processos, a sincronização e a minimização de gargalos. A escalabilidade é um desafio constante para os desenvolvedores de aplicações paralelas e é um tópico importante de pesquisa e desenvolvimento na computação paralela.

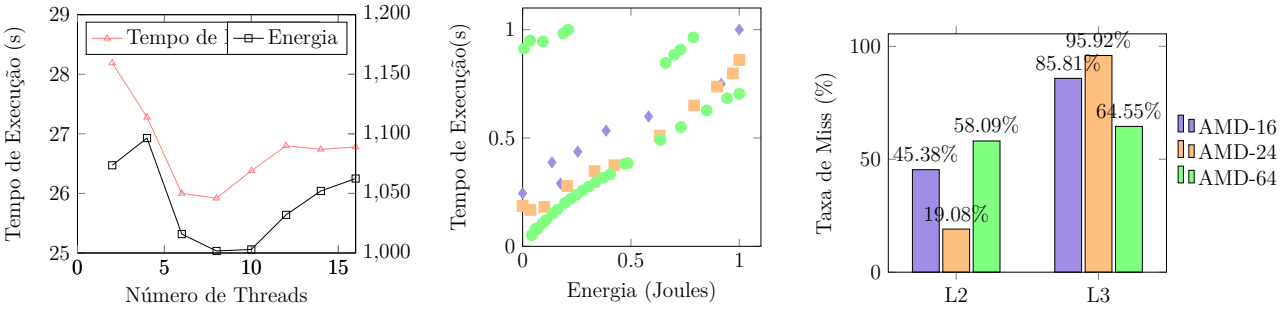
A prática comum adotada por desenvolvedores de *software* em HPC e computação em nuvem consiste da execução de aplicações paralelas utilizando todos os recursos disponíveis. No entanto, diferentes trabalhos têm mostrado que essa abordagem não necessariamente fornecerá o melhor desempenho, economia de energia e resultado de EDP (QASAIMEH et al., 2019; SULEMAN; QURESHI; PATT, 2008; TAFFONI et al., 2019; KRZYWANIAK; CZARNUL; PROFICZ, 2022). As causas estão relacionadas tanto a questões de *software* e *hardware*, e as principais encontradas na literatura são discutidas a seguir:

I. *Saturação do barramento de comunicação entre processador e memória principal.* Para aplicações que operam sob grande quantidade de dados privados à cada *thread* que devem ser constantemente buscados da memória principal, o barramento *off-chip* desempenha um papel decisivo na escalabilidade da aplicação paralela. Uma vez que cada *thread* opera sobre blocos de dados diferentes, a demanda pelo barramento de comunicação aumenta linearmente com o número de *threads*. No entanto, a largura de banda do barramento não aumenta em relação ao número de *threads*, pois é limitada pelo número de pinos de I/O (HAM et al., 2013). Portanto, quando o barramento de comunicação satura, nenhuma melhoria de desempenho é obtida se aumentar o número de *threads* ativas. A Figura 4(a) mostra o comportamento de escalabilidade da aplicação ***HPCG*** na arquitetura AMD-16, onde a relação de *threads* a medida que o número de *threads* é aumentado o consumo de energia e desempenho não são melhorados. Quando o barramento *off-chip* fica saturado não há ganho de desempenho da aplicação.

II. *Sobrecarga de sincronização de dados entre threads.* Em aplicações paralelas, seções críticas são implementadas para garantir a integridade dos dados durante a execução. Assim, apenas uma única *thread* executará esta região crítica de cada vez, serializando uma parte da execução. Neste sentido, quando o número de *threads* aumenta, mais *threads* devem ser serializadas dentro das seções críticas, influenciando negativamente no tempo total gasto para sincronização. Este cenário pode ser observado na Figura 4(b) para a execução da aplicação ***Nbody***. Quando a aplicação é executada com o número menor que quatro *threads*, podemos ver o consumo quase que constante de energia e redução do tempo de execução. No entanto, a partir deste ponto, o tempo que as *threads* gastam sincronizando supera o *speedup* alcançado na região paralela, afetando negativamente o desempenho e a energia.

III. *Número de acessos concorrentes à memória compartilhada.* Em arquiteturas de memória compartilhada, a comunicação entre as *threads* ocorre através de acesso a dados localizados em regiões compartilhadas. Uma vez que estas regiões estão mais distantes do processador (e.g., último nível da memória *cache* e memória principal), esta comunicação pode se tornar um gargalo do desempenho. Além disso, este cenário também pode ser afetado por outros fatores relacionados à exploração do TLP, como por exemplo, afinidade de *threads* e dados. Conforme está ilustrado na Figura 4(c) onde mostra acessos

as memórias *caches* L2 e L3 da aplicação ***BFS***. Quando uma aplicação executada em diferentes arquiteturas pode-se observar que a arquitetura que possui melhor desempenho é a que atingiu a menor taxa de *misses* nos níveis de memória *cache* mais distante do processador. O aumento do número de acessos as memórias *caches* afetam negativamente o consumo de energia.



(a) Aplicação *HPCG* (b) Aplicação *Nbody* (c) Aplicação *BFS*

Figura 4 – Comportamento de escalabilidade de aplicações paralelas em um ambiente de computação em nuvem heterogêneo.

2.2 COMPUTAÇÃO DE ALTO DESEMPENHO NA NUVEM

A computação em Nuvem foi estabelecida ao longo dos anos como uma alternativa para a execução de aplicações devido à sua capacidade de provisionar recursos sob demanda (LIU et al., 2012). A computação em nuvem oferece uma ampla gama de serviços que podem ser categorizados em três tipos: *IaaS* (Infraestrutura como Serviço), *PaaS* (Plataforma como Serviço), Sem Servidor e *SaaS* (Software como Serviço). Neste trabalho, o serviço de nuvem escolhido será o *IaaS* em uma nuvem privada, o que significa que o usuário terá total controle sobre os recursos disponíveis (*e.g.*, CPUs, GPUs e armazenamento), e possa realizar configurações diversas de recursos no *clusters*.

Embora o provisionamento de recursos em ambientes de nuvem seja transparente para os clientes, nos bastidores, distintas tecnologias trabalham em conjunto para oferecer suporte a funcionalidades importantes, como elasticidade e alta disponibilidade (MáRQUEZ; VILLEGAS; ASTUDILLO, 2018). De uma perspectiva de infraestrutura, os processadores multi-core aprimoraram a capacidade de processamento e o ressurgimento da virtualização proporcionou maior flexibilidade de provisionamento para atender às demandas dos clientes.

Do ponto de vista do *software*, arquiteturas distribuídas, como microsserviços, permitiram um melhor uso dos recursos, dando origem a aplicações projetadas para obter o melhor da nuvem (MáRQUEZ; VILLEGAS; ASTUDILLO, 2018). Apesar de seus benefícios, as primeiras implantações de nuvem não conseguiam atender à demanda de aplicações de computação intensiva que exigem respostas rápidas (por exemplo, *Big Data* e *Analytics*) devido à sobrecarga intrínseca gerada pelos *hypervisores* (BARHAM et al., 2003). Como resultado, tecnologias de *container* leves, como *Docker*, começaram a ganhar

terreno em ambientes de nuvem, oferecendo níveis de desempenho muito próximos aos ambientes não virtualizados (XAVIER et al., 2013). *Docker* é considerado o *framework* mais popular para construir, empacotar aplicações em imagens e executar aplicações dentro de *container* (Docker, 2023). Esses *container* contêm todos os dados necessários (por exemplo, bibliotecas e binários). No entanto, ele não oferece suporte a recursos de gerenciamento mais avançados, como balanceamento de carga, autorrecuperação e elasticidade automática.

Assim, plataformas de orquestração como *Kubernetes* (THURGOOD; LENNON, 2019) tornaram-se padrão em ambientes de nuvem ao fornecer recursos que abrangem todo o ciclo de vida da aplicação. Ela oferece diferentes ferramentas de escalonamento para se adaptar a ambiente de execução heterogêneos, como por exemplo, o *Kube-scheduler* que é o escalonador padrão do *Kubernetes* (Kubernetes, 2023). Ele consiste de um processo de plano de controle responsável por verificar quando novos *pods* contendo *containers* são criados e selecionar o nó computacional que irá processar a carga de trabalho baseado em uma estratégia de *round-robin*.

Além disso, é importante destacar que o termo (agregado de computadores), também conhecido como *clusters*, é amplamente utilizado para descrever a diversidade de arquiteturas que podem ser empregadas em diferentes ambientes computacionais, conforme ilustrado na 5(a). Os *clusters* homogêneos têm como característica principal a presença de nodos computacionais que possuem exatamente a mesma arquitetura de CPUs (*e.g.*, número de núcleos iguais em todos os nodos), conforme destacado na Figura 5(a) Por outro lado, *clusters* heterogêneos têm nodos computacionais possuindo arquiteturas de CPUs *multicore* distintas. Isto é, dispõem do número de núcleos diferentes nos nodos computacionais, conforme ilustrado na Figura 5(b) (CHAI; GAO; PANDA, 2007). Esta heterogeneidade traz desafios durante a execução de aplicações paralelas em ambientes de alto desempenho na nuvem.

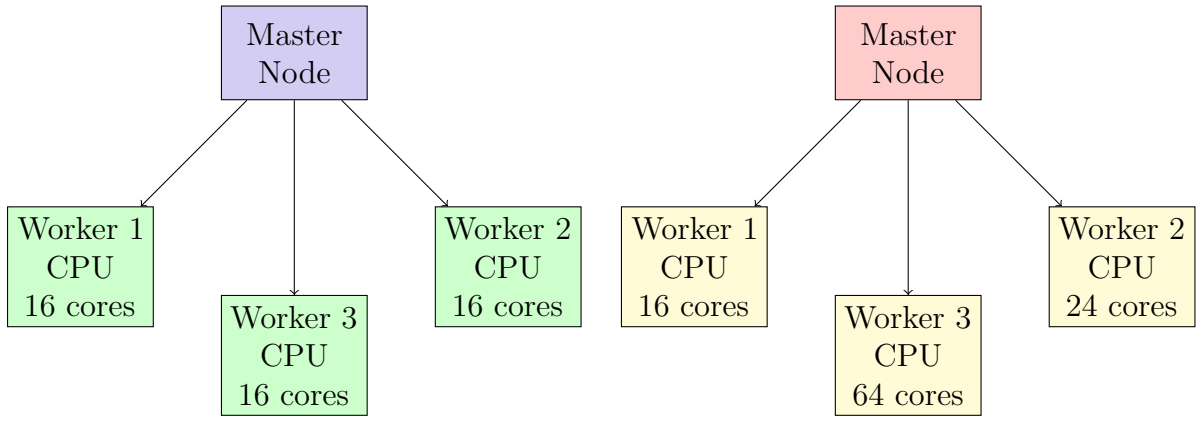


Figura 5 – *cluster multicore* homogênea (a) e heterogênea (b).

A computação de alto desempenho na nuvem vem ganhando cada vez mais popu-

laridade nos últimos anos, uma vez que oferece uma série de vantagens (*e.g.*, escalabilidade, flexibilidade e custos reduzidos) em relação aos sistemas tradicionais (ZVARA et al., 2019). Portanto, a execução de aplicações paralelas de alto desempenho podem ocorrer sem um investimento inicial em infraestrutura própria, reduzindo assim custos operacionais (IOSUP et al., 2011). No entanto, apesar das vantagens, a computação de alto desempenho na nuvem ainda apresenta desafios. Neste sentido, para alcançar um melhor desempenho e redução de custos, é necessário adotar técnicas avançadas de gerenciamento e otimização de recursos.

2.2.1 *KUBERNETES* E *DOCKER*

Com a finalidade de obter o melhor desempenho de aplicações com diferentes cargas de trabalhos e da arquitetura de nuvem, *frameworks* que ajudam a gerenciar cargas de trabalho desequilibradas surgem como alternativa para estas situações. Por exemplo, *kubernetes*[1] é um *framework* de código aberto, portátil e extensível para gerenciar cargas de trabalho e serviços distribuídos em *container* que facilitam a configuração declarativa e a automação. Devido sua capacidade de fornecer recursos que monitoram o ciclo de vida da execução da aplicação, o *kubernetes* tem sido utilizado na computação em nuvem.

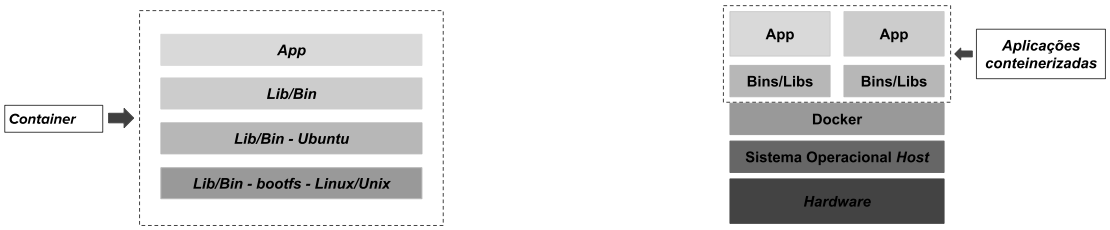
Além de sua capacidade de gerenciar cargas de trabalho, o *Kubernetes* também fornece recursos importantes para monitorar o ciclo de vida da execução da aplicação em nuvem. Ele pode detectar automaticamente falhas em *containers* e reprogramar os recursos necessários para garantir que a aplicação continue funcionando sem interrupções. O *Kubernetes* também pode fazer escalonamento automático com base no uso atual de recursos e prever a necessidade futura de recursos com base no histórico de uso. Isso significa que os desenvolvedores não precisam se preocupar com a complexidade de gerenciar a infraestrutura subjacente e podem se concentrar em desenvolver e implantar aplicativos com mais rapidez e eficiência. Com sua capacidade de automação, escalonamento e monitoramento de aplicativos distribuídos, o *Kubernetes* tornou-se um dos *frameworks* mais populares para gerenciamento de cargas de trabalho em nuvem (Kubernetes, 2023).

Do mesmo modo, o *Docker Container*[2], um *framework* de código aberto, tem se tornado popular para construir, empacotar e rodar aplicações dentro de *container*. Assim, a infraestrutura criada com a utilização desses dois *frameworks* possibilita um ambiente de computação em nuvem com alta disponibilidade, velocidade, portabilidade e escalabilidade, onde os *containers* apresentam uma desempenho semelhante ao ambiente nativo (XAVIER et al., 2013).

Na Figura 6, é ilustrada a infraestrutura que compõem um ambiente de virtualização leve utilizando a plataforma *Docker Container*. A Figura 6(a) destaca o empacotamento de uma aplicação em uma imagem *Docker* que possui todos os dados necessários (*e.g.*,

[1] Informação disponível em https://kubernetes.io/
[2] Informação disponível em https://docs.docker.com/

bibliotecas e binários) para executar uma determinada aplicação. Isso significa que os *Docker Container* são altamente portáveis e consistentes, o que facilita a implantação e a manutenção de aplicações em diferentes ambientes. Assim, quando executamos uma imagem no ambiente com *Docker Container*, geramos um *container*. Na Figura 6(b), pode ser observada a virtualização leve utilizando *Docker Containers*, onde um *Container* é executado sob o sistema operacional do *Host*.



(a) *Docker Container* (b) Virtualização baseada em *container*

Figura 6 – Infraestrutura da virtualização leve com *Docker*.

A virtualização acontece no nível de *kernel* da máquina hospedeira, que é compartilhado com o *container*. Isto possibilita que sejam integradas bibliotecas e dependências do seu programa junto com a aplicação, com menor impacto no desempenho do que a virtualização de *hardware* de uma máquina completa. Neste contexto, o uso de *Docker Container* e *Kubernetes* surgem como alternativas para a criação de ambientes de computação em nuvem e computação de alto desempenho. A virtualização leve a nível de sistema operacional permite melhor utilização de recursos em um servidor físico. Assim, viabilizando uma maior escalabilidade para aplicações paralelas que podem ser atualizadas com mais ou menos recursos facilmente. Dado isso, as cargas de trabalho paralelas ganham espaço nos ambientes de nuvem baseado em *container* (ZHOU et al., 2020). Assim, com a necessidade de distribuição e escalonamento, ferramentas como o *kube-scheduler* são utilizadas, conforme discutido na próxima seção.

2.2.2 *KUBE-SCHEDULER*

Para suprir a demanda de alocação e distribuição de tarefas em nodos de trabalho na nuvem, escalonadores são utilizados. De acordo com (SAHNI; VIDYARTHI, 2015) um escalonador de tarefas é um algoritmo que seleciona uma tarefa e determina onde e como será executada de modo a maximizar a alocação de recursos, minimizar a sobrecarga e balancear o uso dos componentes computacionais (KAPUR, 2015) (STEEN; TANENBAUM, 2017). Para isto, prioridades de escalonamento podem ser consideradas (*e.g*,tipo da aplicação, recursos e nodos disponíveis). O escalonador *kube-scheduler* do *Kubernetes* é um exemplo

de escalonador de tarefas que é responsável por selecionar os nós apropriados para executar os *pods*, levando em consideração as restrições de recursos e políticas definidas no *cluster*.

O escalonador de tarefas *Kube-Scheduler* é um componente importante do *Kubernetes*. Ele faz parte do conjunto de componentes que compõem a camada de gerenciamento do *Kubernetes*, também conhecido como *Control Plane*. O papel do *Kube-Scheduler* é monitorar os *pods* recém criados e selecionar um nó ideal para sua execução, garantindo a eficiência e o equilíbrio na utilização dos recursos disponíveis no *cluster*. Isso maximiza o uso dos recursos e garante a disponibilidade das aplicações (Kubernetes, 2023).

Entretanto, os *containers* em *pods kubernetes* podem ter necessidades distintas em relação a recursos. O *kube-scheduler* aborda essa questão por meio da identificação de nodos viáveis a partir da avaliação da pontuação concedida a cada nó disponível. Onde diversos fatores são levados em consideração (*e.g.*, *hardware*, *software*) pelo *kube-scheduler* para a decisão de qual nodo será escalonado a tarefa. Neste sentido, quando aplicações paralelas são escalonadas para execução em um *cluster* heterogêneo o desafio de escalonamento não está só associado aos recursos mas também as diferentes características das aplicações e das arquiteturas.

A Figura 7 descreve o *Kube-Scheduler*, que funciona da seguinte forma. Quando uma nova tarefa (*e.g.*, *pod*) é criada pelo *Kube-Controller Manager*, o *Kube-Scheduler* é notificado. Em seguida, o *Kube-Scheduler* examina o objeto *pod* e coleta informações sobre os recursos disponíveis no *cluster*. Com essas informações, o *Kube-Scheduler* decide qual é o melhor nodo para alocar a tarefa, levando em consideração fatores como a quantidade de recursos disponíveis, restrições de localização e políticas de prioridade. Após decidir qual nó deve alocar a tarefa, o *Kube-Scheduler* atualiza o estado do objeto *pod* para refletir o nó alocado. Por fim, o *Kubelet* no nodo alocado inicia a execução da tarefa.

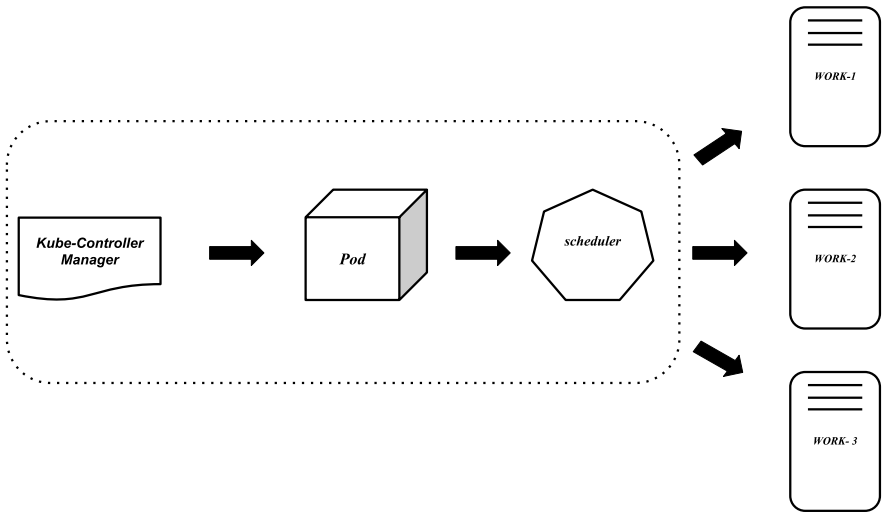


Figura 7 – Fluxo de escalonamento do *pod* realizado pelo *kube-scheduler*.

2.3 APRENDIZADO DE MÁQUINA

O avanço exponencial na geração de dados tem sido um desafio para a análise de informações (CHO; GUZMAN; EGGER, 2018). Nesse contexto, o aprendizado de máquina e as redes neurais surgem como uma solução para permitir que os computadores aprendam e melhorem seu desempenho a partir da análise de grandes conjuntos de dados de forma autônoma.

O aprendizado de máquina se baseia na ideia de que um sistema computacional pode aprender a partir de dados e melhorar seu desempenho em uma tarefa específica ao longo do tempo, sem a necessidade de ser explicitamente programado. Essa técnica deixa de lado as tradicionais técnicas de programação, em que um desenvolvedor precisa escrever um código que instrui o sistema a realizar uma tarefa de forma específica, e passa a utilizar um conjunto de dados para que o próprio sistema possa aprender a partir deles (ZHOU, 2021). As Redes Neurais Artificiais (RNAs) são uma das técnicas mais utilizadas em aprendizado de máquina, uma vez que esses modelos são capazes de aprender a partir de exemplos e identificar padrões em grandes conjuntos de dados.

As redes neurais são compostas por camadas de neurônios interconectados que processam informações e geram saídas. Elas são capazes de aprender a partir de um conjunto de dados de treinamento e, em seguida, aplicar esse conhecimento para fazer previsões em novos dados (CHOI et al., 2020). As redes neurais, baseadas na estrutura e funcionamento do cérebro humano, têm se mostrado uma poderosa ferramenta no campo do aprendizado de máquina, permitindo a criação de modelos complexos de análise de dados (LUCKERT; SCHAEFER-KEHNERT, 2016). Além disso, redes neurais podem ser utilizadas em diversas áreas (*e.g.,* reconhecimento de padrões, análise de dados, previsão de eventos, entre outras).

A aplicação de RNAs em computação de alto desempenho tem se mostrado promissora (QIAO et al., 2017), uma vez que esses modelos são capazes de lidar com grandes quantidades de dados de forma eficiente, realizando tarefas que seriam extremamente trabalhosas ou impossíveis para um ser humano. O desempenho das RNAs nas tomadas de decisões em HPC pode ser altamente eficiente, especialmente em situações que envolvem grande volume de dados e necessidade de análise em tempo real (CHO; GUZMAN; EGGER, 2018; CHOI et al., 2020).

A Figura 8 ilustra uma rede neural do tipo *feedforward neural network* contendo duas entradas (a1 e a2) e uma saída (s1). As entradas da rede neural são fornecidas à primeira camada de neurônios, que passam as entradas para as unidades na camada oculta seguinte. No exemplo, a camada oculta (*hidden layer*) é composta de cinco neurônios (n1...n5) que aplicam uma função de ativação às suas entradas ponderadas. Importante ressaltar que uma rede neural pode ter vários neurônios de entrada, na camada oculta e saída. As funções de ativação são geralmente não-lineares e usadas para introduzir não-linearidade na rede, permitindo que ela aprenda relações mais complexas entre as

entradas e a saída (LECUN; BENGIO; HINTON, 2015). Por fim, a camada de saída (s1) consiste em um único neurônio que recebe os valores da camada oculta e aplica uma função de ativação para produzir um valor de saída.

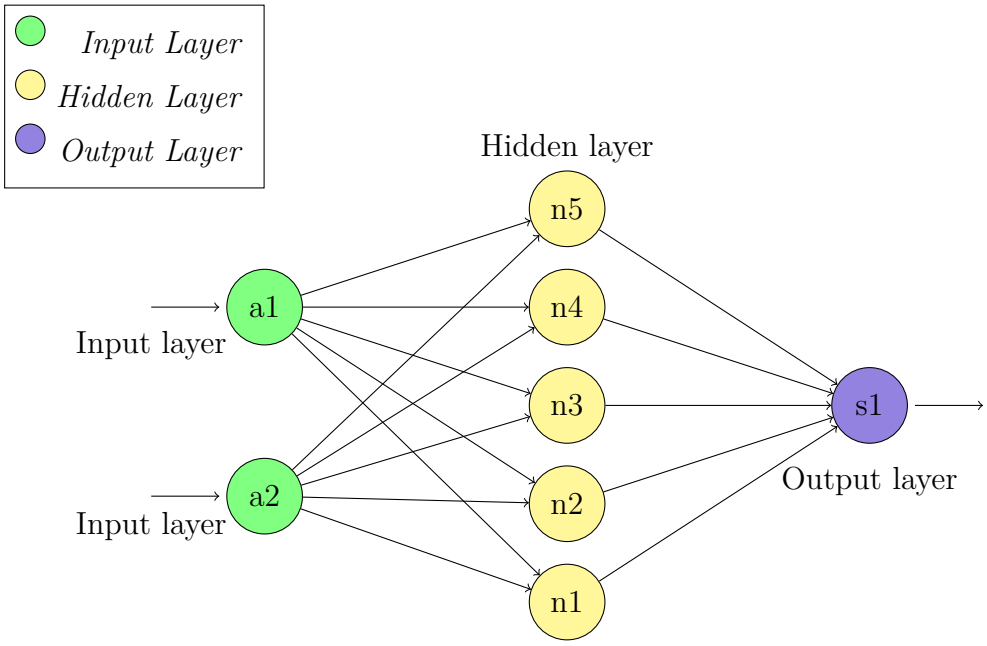


Figura 8 – Representação simplificada de uma Rede Neural.

Após definir a rede neural, é necessário treiná-la para que os graus de liberdade possam ser adaptados de maneira ótima para solucionar a tarefa em questão. Os paradigmas de Aprendizagem de Máquina são estratégias ou abordagens utilizadas para desenvolver algoritmos de aprendizagem de máquina. Esses paradigmas descrevem a forma como o sistema de aprendizagem deve ser projetado para aprender a partir de dados. Existem vários paradigmas de aprendizagem de máquina, incluindo aprendizagem supervisionada, aprendizagem não supervisionada e aprendizagem por reforço (KOVÁCS, 2002).

Os sistemas de aprendizado de máquina são categorizados de acordo com as metodologias de aprendizado subjacentes, caracterizadas pela capacidade inferencial do algoritmo.

- **Aprendizado supervisionado**: é uma técnica de aprendizado de máquina na qual um algoritmo é treinado em um conjunto de dados rotulados, ou seja, dados que já possuem as respostas corretas para o problema em questão. O objetivo do algoritmo é aprender a mapear as entradas para as saídas corretas, de modo que ele possa generalizar e produzir respostas precisas para novas entradas. Aprendizado supervisionado amplamente utilizado em diversas áreas, como processamento de linguagem natural, visão computacional, diagnóstico médico e reconhecimento de voz. Algoritmos de aprendizado supervisionado, como regressão linear, árvores de decisão e redes neurais, são comumente usados para tarefas de classificação e regressão (CHOI et al., 2020; MOLNAR, 2020).

- **Aprendizagem por Reforço**: é um ramo que se concentra em como um agente deve tomar decisões em um ambiente para maximizar uma noção de recompensa acumulada ao longo do tempo. Em outras palavras, um agente aprende a escolher ações que maximizam uma recompensa a longo prazo, com base nas informações que recebe do ambiente em que está inserido (HASTIE et al., 2009).

- **Redes neurais convolucionais** são altamente eficazes na análise de imagens, e são frequentemente usadas em aplicações de visão computacional, como detecção de objetos em imagens, reconhecimento facial, diagnóstico médico e automação industrial (MOLNAR, 2020; VOGADO et al., 2019; CUI; CHEN; CHEN, 2016).

- **Redes neurais recorrentes**: são usadas em tarefas que envolvem sequências de dados, como análise de texto e previsão de séries temporais, como previsão de demanda de energia elétrica, previsão de preços de ações e análise de sentimentos em mídias sociais. As redes neurais recorrentes são especialmente úteis para lidar com dados de sequência, que podem ter dependências complexas entre os pontos de dados (MOLNAR, 2020).

As RNAs são uma escolha eficaz para tomada de decisões em computação de alto desempenho. Assim, essa técnica foi selecionada para o desenvolvimento deste trabalho devido a diversas razões, incluindo sua capacidade de lidar com grandes volumes de dados complexos. Além disso, RNAs são altamente escaláveis e podem ser treinadas continuamente à medida que novos dados são coletados, permitindo que se adaptem a novas situações e melhorem sua precisão ao longo do tempo. Com essas características, as redes neurais podem ser uma ferramenta valiosa para a tomada de decisões em tempo real em diversas áreas da computação de alto desempenho.

2.4 CONCLUSÃO DO CAPÍTULO

Em suma, este capítulo descreveu tópicos relacionados a computação de alto desempenho, computação na nuvem e aprendizado de máquina. Arquiteturas *multicore* são sistemas de processadores que possuem múltiplos núcleos de processamento em um único *chip*, permitindo que múltiplas tarefas sejam executadas simultaneamente. Introduziu-se também o conceito de programação paralela, uma técnica que permite a exploração de paralelismo com o objetivo de melhorar o desempenho de um programa. Tendo em vista que aplicações paralelas podem ter sua escalabilidade limitada por algum fator de hardware ou software, discutiu-se as principais causas: sincronização de dados, acessos concorrentes à memória compartilhada e saturação do barramento de comunicação entre memória principal e processador. Descreveu-se também questões relacionadas a computação de alto desempenho na nuvem, destacando o uso de tecnologias de gerenciamento e criação de

containers, *Kubernetes* e *Docker*, com destaque para o escalonador *kube-scheduler*. Por fim, abordamos aprendizado de máquina e redes neurais.

3 TRABALHOS RELACIONADOS

Neste capítulo, os trabalhos relacionados ao tema desta dissertação são apresentados.

O trabalho de (CURTIS-MAURY et al., 2006) propõe um *framework* para otimizar o desempenho e consumo de energia em aplicações *multithreaded* através do gerenciamento de *threads* em processadores SMT (*Simultaneous multithreading*). A abordagem utiliza predição baseada em eventos de hardware para predizer a carga de trabalho da aplicação e ajustar dinamicamente a frequência e tensão dos núcleos computacionais, para atingir o equilíbrio entre desempenho e consumo de energia. Os resultados mostram uma economia de energia de até 40% em comparação com a configuração de frequência fixa, com impacto mínimo no desempenho.

Hotta (HOTTA et al., 2006) apresenta um método de otimização do desempenho e consumo de energia em um *cluster* de alto desempenho que seleciona a melhor frequência de operação do *core* considerando o *overhead* de transição entre frequências. O método proposto utiliza perfis de carga de trabalho para predizer a demanda futura de recursos do sistema e ajustar dinamicamente a tensão dos processadores para reduzir o consumo de energia sem comprometer o desempenho. Adicionalmente, a técnica aplica o desligamento de núcleos ociosos para reduzir ainda mais o consumo de energia. A otimização proposta é capaz de reduzir o consumo de energia em até 40% comparado a *Governor Ondemand*.

Nathan (VYDYANATHAN et al., 2006) discutem uma abordagem integrada para a alocação de processadores e escalonamento de aplicações paralelas híbridas (*e.g.*, MPI + OpenMP) com o objetivo de minimizar o tempo de conclusão paralela (*makespan*, ou seja o tempo mais longo que um conjunto de tarefas pode levar para ser concluído). O algoritmo com uma alocação inicial do processador e reduz iterativamente o *makespan*, aumentando o grau de paralelismo de dados e tarefas no caminho crítico da aplicação. Os resultados obtidos em um ambiente simulado mostram que a abordagem proposta melhora a eficiência energética em até 30% quando se comparada a *Minimum Processor Allocation Heuristic*.

Feedback-Driven Threading (*FDT*) (SULEMAN; QURESHI; PATT, 2008) é uma abordagem que define o número de *threads* de uma aplicação paralela implementada com OpenMP para otimizar o desempenho e correspondente consumo de energia. Para isso, *FDT* analisa o comportamento da aplicação (*e.g*, largura de banda ou sincronização de dados) e, então define o número de *threads* adequado. Os resultados mostram que as abordagens propostas conseguem definir o melhor número de *threads*, fornecendo em média uma redução de 17% no tempo de execução e 59% na energia consumida, quando comparada com o modo convencional, onde o número de *threads* é igual ao número de cores disponíveis na arquitetura.

A proposta de (ETINSKI et al., 2010) apresenta um algoritmo *Power-Aware* de escalonamento de trabalhos (*jobs*) paralelos. O objetivo é minimizar o consumo de energia

dos sistemas sem comprometer o desempenho do mesmo. O algoritmo proposto leva em consideração a utilização atual de recursos e a previsão de recursos futuros para determinar a ordem de execução das tarefas. O algoritmo também considera a eficiência energética de cada recurso e o tempo necessário para a transferência de dados entre eles. Os resultados experimentais mostram que o método proposto é capaz de reduzir o consumo de energia em até 30% comparado com métodos de escalonamento convencionais (e.g.,escalonador baseado em prioridade e estimativas de tempo de execução).

Thread Reinforcer (TR) (PUSUKURI; GUPTA; BHUYAN, 2011) é um *framework* que seleciona automaticamente o número ideal de *threads* para maximizar o desempenho de aplicações paralelas. Para isto, *Thread Reinforcer* baseia-se em observações de fatores do sistema operacional como, contenção de recursos compartilhados, taxa de migração de *threads*, taxa de troca de contexto e utilização do processador. TR está dividido em duas partes: na primeira, a aplicação é executada diversas vezes por um curto período, onde o comportamento da aplicação é monitorado. Com base nestas obervações, TR define o número de *threads* apropriado. Após, a segunda fase inicia, onde a aplicação é completamente reexecutada usando o número de *threads* encontrado na primeira parte. Os resultados mostraram melhorias de até 40% na utilização de recursos em comparação com a configuração estática de número de threads.

(TAKOUNA; DAWOUD; MEINEL, 2012) apresentam uma abordagem para escalonamento eficiente de trabalhos de alto desempenho em *clusters* virtualizados com o objetivo de maximizar a eficiência energética do sistema. A abordagem consiste em ajustar dinamicamente a configuração das máquinas virtuais e dos *hosts* de acordo com a carga de trabalho. Os resultados mostram uma economia de energia de até 21% em comparação com o escalonamento padrão. Similarmente, (CHARR et al., 2015) apresenta uma proposta para redução do consumo de energia em aplicações iterativas em arquiteturas heterogêneas. Os autores propõem o uso de *DVFS* (*Dynamic Voltage and Frequency Scaling*) para ajustar dinamicamente a frequência e tensão dos núcleos dos processadores, a fim de otimizar o consumo de energia. Os resultados mostraram uma economia de energia de até 35% em comparação com a configuração convencional de frequência fixa.

Callisto (HARRIS; MAAS; MARATHE, 2014) é uma abordagem para escalonar várias aplicações simultaneamente com o objetivo de maximizar a capacidade de processamento do sistema e evitar desperdícios através da alocação coordenada de recursos, como CPU, memória e largura de banda de rede. *Callisto* aplica uma técnica para fornecer escalonamento espacial dinâmico para reduzir a interferência de desempenho entre diferentes aplicações paralelas. Sua estratégia consiste em atribuir um número igual de *threads* para cada aplicação co-localizada. Os resultados mostram que a plataforma *Callisto* pode aumentar a escalabilidade das aplicações em até 20%, em comparação com abordagens onde o escalonamento é baseado em prioridade e nos requisitos de memória de cada tarefa.

NuCore (WANG; DAVIDSON; SOFFA, 2016) é um algoritmo para ajustar o

grau de paralelismo, alocação de núcleos e predição de largura de banda de memória em sistemas NUMA. Para isso, *NuCore* aplica predição baseada em modelos que usa dados de desempenho de aplicações para predizer a largura de banda de memória e a alocação de núcleos ótimos. Os resultados mostram que *NuCore* em comparação com a abordagem padrão de alocação de núcleos, os ganhos de desempenho variaram de 10% a 49%, com uma média de 28%. Adicionalmente, a ferramenta proposta apresenta ganhos de desempenho de até 15% em comparação com abordagens como alocação de recursos com base na topologia da maquina NUMA, sem levar em conta a natureza da carga de trabalho do aplicativo.

Diagnosing Performance Variations in HPC Applications Using Machine Learning (TUNCER et al., 2017) é uma abordagem para diagnosticar variações de desempenho em aplicações de computação de alto desempenho utilizando técnicas de aprendizado de máquina. A metodologia proposta utiliza a análise de componentes principais, que busca reduzir a dimensionalidade dos dados. Essa técnica é frequentemente utilizada para simplificar a representação dos dados e melhorar o desempenho de algoritmos de análise, para extrair as principais características dos dados de desempenho e, em seguida, aplicar um modelo de regressão linear para identificar os principais fatores que contribuem para as variações de desempenho. Os resultados mostram um ganho de 80% em identificar problemas de desempenho nas aplicações de HPC comparado a técnicas como análise de regressão e a análise de descomposição de frequência .

O provisionamento de recursos baseado em aprendizagem proposto por (MAKRANI et al., 2018) busca atribuir uma configuração de *hardware* adequada para otimizar o desempenho e eficiência energética de cargas de trabalho na nuvem. O método proposto utiliza aprendizado de máquina para modelar o consumo de energia e o desempenho em relação à alocação de recursos. Com base nessas informações, o sistema é capaz de ajustar dinamicamente a alocação de recursos para atingir um equilíbrio entre desempenho e consumo de energia. Assim, o método proposto é capaz de reduzir o consumo de energia na execução de aplicações em até 20% quando comparado a configuração padrão sem otimização que não ajusta os recursos com base na carga de trabalho.

O algoritmo proposto por (GUERRERO; LERA; JUIZ, 2018) realiza a otimização da alocação de *container* em arquiteturas de nuvem. O trabalho tem dois objetivos específicos: (*i*) otimizar o tempo de execução das aplicações; e (*ii*) minimizar o custo da infraestrutura de nuvem necessária para suportar as aplicações. Os autores utilizam um algoritmo genético para encontrar soluções ótimas, sendo composto por três etapas principais: seleção, cruzamento e mutação. Na etapa de seleção, as soluções mais aptas são selecionadas para reprodução. Na etapa de cruzamento, novas soluções são criadas a partir da combinação das soluções selecionadas. Por fim, na etapa de mutação, algumas soluções são modificadas aleatoriamente para introduzir variabilidade na população. Os resultados mostram que o algoritmo genético proposto é capaz de encontrar soluções que são em média 32% melhores em termos de custo e tempo de execução em comparação com

as soluções como a abordagem baseada em heurística *First Fit* e a abordagem baseada em algoritmo de enxame de partículas.

RLSK (*Reiforcement Learning Scheduler Kubernetes*) (HUANG; XIAO; WU, 2020) é uma estratégia baseada em aprendizado por reforço para escalonar tarefas independentes entre múltiplos *clusters* de maneira adaptável. RLSK é executado fora de cada cluster, para coletar os dados de cada *cluster kubernetes* e entregar tarefas ao *cluster* correspondente. RLSK recebe as informações da tarefa em lote enviadas pelos usuários e, em seguida, combina as informações de *status* coletadas atualmente de cada *cluster* do módulo assistente e toma a decisão de escalonamento. Experimentos demonstram uma eficácia 70% maior de RLSK quando comparado ao escalonamento padrão do sistema.

(ABBASI et al., 2021) apresentam o agrupamento de tarefas dependentes em um *cluster* usando o método *Bayesian* para resolver o problema de escalonamento de tarefas com afinidade em sistemas *multicore* heterogêneos. Para tanto, duas técnicas são propostas: CBS (*Chunk-Based Scheduler*), onde todas as tarefas a serem processadas são divididas em pedaços de tamanho variável de acordo com o número de núcleos disponíveis; e QBICTM(*Quantum-Based Intra-Core Task Migration*), onde todas as tarefas são divididas em pedaços de tamanho igual. Os resultados mostram uma melhoria de 30% a 55% no tempo de execução dos escalonadores CBS e QBICTM comparado com um escalonador baseado em tempo de execução e outro baseado em balanceamento de carga, respectivamente.

Peps (*Predictive energy-efficient*) (MAGHSOUD; NOORI; MOZAFFARI, 2021) é um escalonamento de tarefas em processadores *multicore*, que utiliza um modelo de aprendizado de máquina para predizer o consumo de energia e o desempenho de cada tarefa com diferentes configurações de tensão, frequência de operação e número de núcleos ativos. Os resultados mostram que *Peps* foi capaz de reduzir o consumo de energia em até 47% em comparação com as técnicas *FCFS (first-come, first-served)* e *EDF* (*earliest-deadline-first*)

(SANGEETHA et al., 2022) apresentam um *framework* de gerenciamento de recursos baseado em redes neurais profundas para ambientes de nuvem. O *framework* proposto é composto por dois módulos principais, (*i*) módulo de predição de recursos e (*ii*) módulo de alocação de recursos. O módulo de predição usa redes neurais profundas para predizer a carga de trabalho futura, enquanto o módulo de alocação utiliza um algoritmo de alocação baseado em aprendizagem por reforço para alocar recursos de forma eficiente. Os resultados mostram que o modelo proposto alcançou uma melhoria significativa na alocação de recursos em comparação com técnica de alocação padrão de gerenciamento de recursos em nuvem. Em particular, o modelo proposto alcançou uma melhoria de 30% na utilização de recursos e uma redução de 20% no tempo de resposta. Semelhante, (BI et al., 2019) utiliza de redes neurais profundas para predizer o tempo de execução de tarefas em sistemas de computação em nuvem. Os autores fazem o uso dos dados históricos de uso de recursos, como CPU e memória, para treinar a rede neural e predizer o tempo de

execução de novas tarefas.

3.1 CONTRIBUIÇÕES DESTA DISSERTAÇÃO

Conforme foi abordado na seção anterior, diversos trabalhos têm se aprofundado no estudo da otimização de aplicações paralelas. Entretanto, é possível notar que tal análise em ambiente de nuvem heterogênea possui espaço de exploração no contexto de computação de alto desempenho na nuvem.

Encontram-se poucos trabalhos que investigam o impacto da execução de aplicações paralelas com diferentes particularidades de grau de paralelismo em ambiente de nuvem onde a características das arquiteturas utilizadas são heterogêneas. Por sua vez, trabalhos como o de (VYDYANATHAN et al., 2006; SULEMAN; QURESHI; PATT, 2008; ETINSKI et al., 2010), objetivam aprimorar a otimização da execução de aplicações paralelas para otimizar seu desempenho e economia de energia. Já outras pesquisas, como as realizadas por (TUNCER et al., 2017; MAKRANI et al., 2018; MAGHSOUD; NOORI; MOZAFFARI, 2021; HUANG; XIAO; WU, 2020), empregam técnicas de aprendizado de máquina e redes neurais artificiais para tomada de decisões em sistemas de computação de alto desempenho. Entretanto, esses estudos não levam em consideração o grau de paralelismo e as características das arquiteturas alvo.

A presente dissertação tem como principal contribuição o *TLP-Allocator*, uma abordagem que utiliza redes neurais para encontrar o nodo computacional e grau de paralelismo ideal em um ambiente heterogêneo para executar aplicações paralelas. Além disso, também apresenta uma análise do impacto no desempenho, consumo de energia e no EDP quando aplicações paralelas com diferentes graus de paralelismo são executadas em um ambiente com arquiteturas heterogêneas.

4 EXPLORAÇÃO DE ESPAÇO E PROJETO

Neste capítulo, apresentamos uma análise do comportamento de diversas aplicações paralelas em um ambiente de nuvem privada heterogênea, considerando diferentes configurações de nodos computacionais e números de *threads*. Para tanto, descrevemos detalhadamente a metodologia utilizada na execução dos experimentos, incluindo informações sobre as aplicações empregadas e as especificações do ambiente de execução. Em seguida, apresentamos e discutimos os resultados obtidos a partir dessa análise.

4.1 METODOLOGIA

4.1.1 CONJUNTO DE APLICAÇÕES

Nós consideramos vinte aplicações escritas em Linguagem C/C++ de diferentes suítes de *benchmarks* e já paralelizadas com a interface de programação paralela OpenMP:

- ***Oito kernels e pseudo-aplicações da suíte do NAS Parallel Benchmarks***: *Numerical Aerodynamic Simulation Parallel Benchmarks (NAS-PB)* (BAILEY et al., 1991): Eles se concentram em dinâmica de fluídos computacional e em disciplinas aeronáuticas relacionadas. *Block tri-diagonal*(BT), uma aplicação de dinâmica de fluidos de computação simulada CFD (*Cumulative Flow Diagram*) que resolve sistemas de equações resultantes de uma discretização de diferenças finitas implícitas de maneira aproximada fatoradas das equações de *Navier-Stokes*; *Conjugate Gradient*(CG), um *kernel* que usa um método de gradiente conjugado para calcular uma aproximação para o menor autovalor de uma matriz grande, esparsa e não estruturada;*Embarrassingly Parallel* (EP), um *kernel* embaraçosamente paralelo que gera pares de desvios aleatórios gaussianos de acordo com um esquema específico;*discrete 3d fast Fourier transform*(FT), uma solução de equação diferencial parcial 3-D usando FFTs; *Lower-upper*(LU); uma aplicação de CFD que usa o método de super-relaxamento sucessivo simétrico para resolver um sistema de sete blocos diagonais; *Multi-grid*(MG), um *kernel multigrid* simplificado que calcula a solução da equação de Poisson escalar 3-D; *Scalar penta-diagonal*(SP); um aplicativo CFD com comportamento semelhante ao (BT);*Unstructured Adaptive* (UA), um *benchmark* que envolve a solução de um problema de transferência de calor estilizado em um domínio cúbico, discretizado em uma malha não estruturada adaptativamente refinada.

- ***Três aplicações da suíte do Rodinia Benchmark*** (CHE et al., 2009): *Hotspot* (HS), uma ferramenta de simulação térmica usada para estimar a temperatura do processador; Decomposição LU (*LU Decomposition – LUD*), um algoritmo que calcula as soluções de um conjunto de equações lineares; e *Streamcluster*(SC), uma aplicação

que, dado um fluxo de pontos de entrada, encontra um número predeterminado de medianas para que cada ponto seja atribuído ao seu centro mais próximo.

- ***Três aplicações da suíte do Parboil Benchmarks*** (STRATTON et al., 2012): Pesquisa em largura *Breadth-First Search*(BFS), que percorre todos os componentes conectados em um grafo; *Lattice-Boltzmann*(LBM), uma simulação de dinâmica de fluidos usando o Método *Lattice-Boltzmann*;*Magnetic Resonance Imaging*(MRI), um algoritmo que calcula uma grade regular de dados representando uma varredura de RM por interpolação ponderada de pontos de dados realmente adquiridos.

- **Seis aplicações de Diferentes domínios**: *Fast Fourier Transform* (FFT), um algoritmo que calcula a Transformada discreta de *Fourier* (DFT) e a sua inversa; *high performance conjugate gradient* (HPCG) (DONGARRA; HEROUX; LUSZCZEK, 2015), um algoritmo que modela os padrões de acesso a dados de aplicações do mundo real como cálculos de matrizes esparsas, testando assim o efeito das limitações do subsistema de memória e interconexão interna do supercomputador em seu desempenho computacional; Método de Jacobi (JA), um algoritmo para resolver sistemas de equações lineares; *n-body* (NB), que computa uma simulação de um sistema dinâmico de partículas; *Poisson* (PO), que calcula uma solução aproximada para a equação de *Poisson* em uma região retangular; *STREAM* (ST), um programa de *benchmark* sintético que mede a largura de banda sustentável da memória principal em MB/s e a taxa de computação correspondente para *kernels* de vetor simples.

As aplicações escolhidas abrangem uma ampla gama de instruções por ciclo (IPC) e comportamento de acesso às memórias *cache* L2 e L3 nas arquiteturas alvos. Este conjunto de aplicações é representativo para os nossos experimentos, pois possuem diferentes características com relação ao acesso à memória principal e uso de CPU: aplicações com baixo/médio/alto IPC e aplicações com número variado de *misses* nas *caches* L2 e L3. Os dados para esta métrica foram retirados diretamente dos contadores de *hardware* através da ferramenta *AMDuProf* (HACKENBERG et al., 2013).

4.1.2 AMBIENTE DE EXECUÇÃO

Os experimentos foram realizados em um ambiente de nuvem heterogêneo composto por quatro arquiteturas *multicore*, conforme mostrado na Figura 9: Um nodo *master*, responsável por distribuir as aplicações para execução em um dos três nodos trabalhadores (*AMD-16*, *AMD-24* e *AMD-64*). A frequência de operação de cada CPU foi configurada para ajustar de acordo com a carga de trabalho da aplicação, através do *governor* DVFS *ondemand*, que é o padrão usado na maioria das versões Linux. A frequência de operação base de cada arquitetura pode ser visto na Tabela 1, assim como as configurações das memórias *cache*. Cada nodo estava usando o Sistema Operacional Linux Ubuntu com

kernel v. 5.13.0, *kubernetes* v. 1.23.6 com o *Docker* 20.10.17, *build* 100c701. Cada aplicação foi compilada com GCC/G++ 10.2, usando a *flag* de otimização $-O3$. A alocação e afinidade de cada *thread* foi configurada através da variável de ambiente do OpenMP: `OMP_PROC_BIND=CLOSE` e `OMP_PLACES=CORES`. Esta definição faz com que as *threads* de uma mesma aplicação sejam alocadas próximas uma das outras.

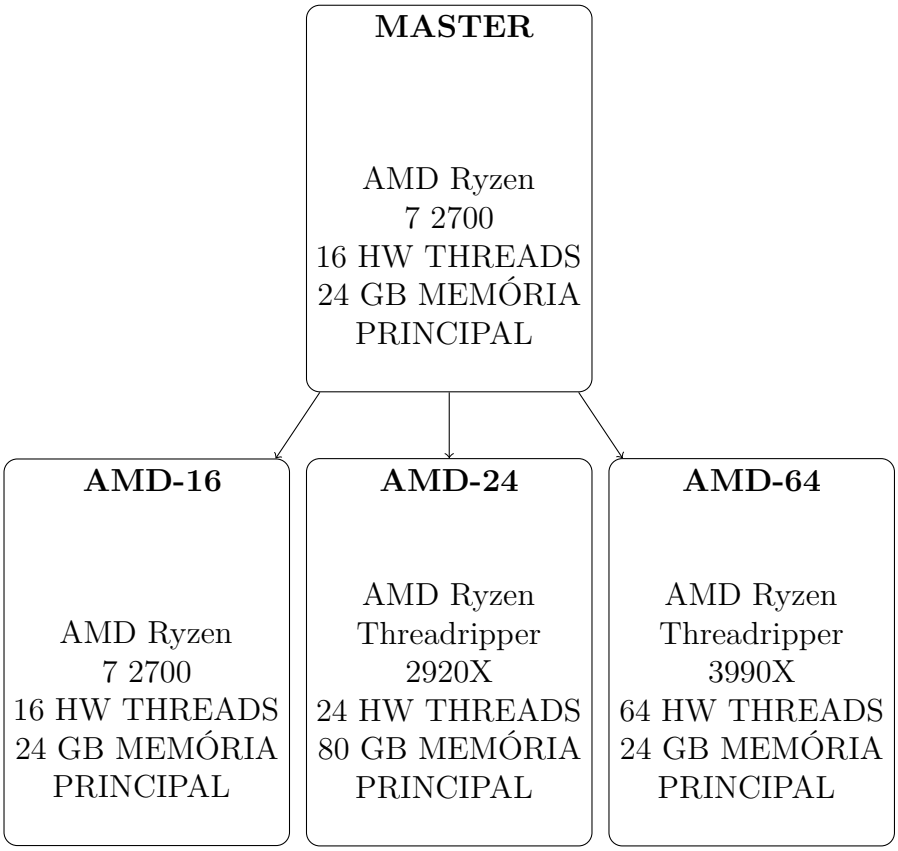


Figura 9 – Ambiente utilizados nos experimentos

Na análise apresentada neste capítulo, nós consideramos as métricas de tempo de execução, consumo de energia e EDP. O EDP é utilizado para avaliar o custo-benefício entre o total de energia consumida e o tempo de execução de uma aplicação. Sua fórmula consiste da multiplicação da energia pelo tempo de execução. Ela vem sendo bastante utilizada no meio acadêmico pois possibilita analisar, em um único valor, a relação entre o consumo de energia e o desempenho. Por exemplo, considerando dois cenários: (i) uma aplicação é executada em 100 segundos e consome 10 joules de energia; e (ii) uma aplicação é executada em 50 segundos e consome 40 joules de energia. O cenário i tem EDP de 1000, enquanto que o cenário ii tem EDP de 2000. Isso mostra que, embora o cenário i seja duas vezes mais lento, ele possui a melhor relação de consumo de energia e tempo de execução. O tempo foi obtido com a função do OpenMP `omp_get_wtime`. Por outro lado, o consumo de energia foi obtido diretamente dos contadores de hardware presentes nos processadores modernos via *Application Power Management* (HACKENBERG et

al., 2013). Os resultados apresentados a seguir são uma média de dez execuções de cada configuração (aplicação, máquina e número de *threads*) com um desvio padrão menor que 0.5%.

Tabela 1 – Tamanho das memórias *caches* e frequência (CHz) base de operação em suas respectivas arquiteturas.

	AMD-16	***AMD-24***	***AMD-64***
Cache L2	4 MiB	6 MiB	32 MiB
Cache L3	16 MiB	32 MiB	256 MiB
Frequência base de Operação	3.2 GHz	3.5 GHz	2.9 GHz

4.1.2.1 CRIAÇÃO DO *CLUSTER KUBERNETES*

Para possibilitar reprodutibilidade da nossa análise, descrevemos aqui o processo utilizado durante a configuração do ambiente de execução para realização dos experimentos. A criação do *cluster* heterogêneo foi realizado com a utilização do *framework kubernetes*. A infraestrutura consiste em um *cluster* com quatro nodos computacionais. Como pode-se observar na Figura 10, o nodo *master* tem o papel de gerenciador do ambiente *kubernetes* e os outros três compõem os nodos de trabalho. Neste contexto, o nodo *master* possui os componentes necessários para o *cluster* e os componentes fazem parte do plano de controle *kubernetes*.

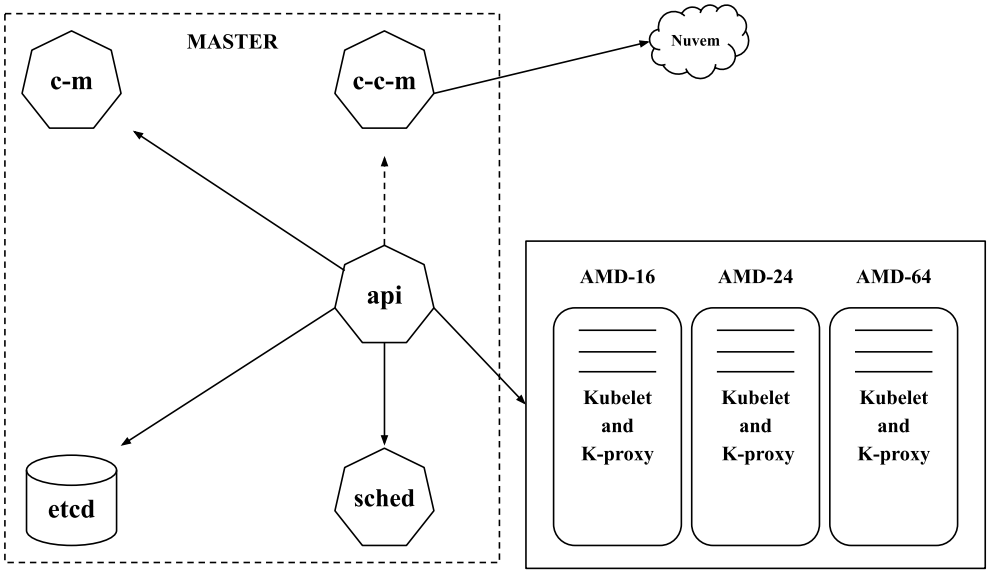


Figura 10 – Diagrama dos componentes de um *cluster Kubernetes*.

A camada de gerenciamento do *Cluster kubernetes* é composta por cinco componentes principais, como ilustrado na Figura 10: ***Kube-controller-manager (c-m)***, responsável por manter a integridade dos nodos; O banco de dados ***etcd***, responsável por armazenar todas informações do *cluster*; O gerenciador de controle da nuvem ***Cloud-controller-manager*** (***c-c-m***); ***Kube-apiserver***, o *front-end* para a camada de

gerenciamento do Kubernetes; e por fim, o componente ***Kube-scheduler*** (***sched***), da camada de gerenciamento que observa os *pods* recém-criados sem nenhum nó atribuído e seleciona um nó para executá-los, conforme mostra a Figura 7. A camada de gerenciamento é executada no nodo definido como *MASTER* do *Cluster Kubernetes*.

Para executar as aplicações em um nodo trabalhador, a seguinte metodologia foi utilizada. A execução de uma aplicação paralela está dividida em sete etapas. Primeiramente, é preciso implementar a aplicação paralela, definindo os processos paralelos e configurando os recursos necessários para o seu funcionamento (*i*). Em seguida, a aplicação é encapsulada em um *Container Docker* para permitir a sua execução de forma isolada em diferentes ambientes (*ii*). É criado um arquivo (*Kubernetes Manifest*) do *pod* para descrever como o *container* deveria executar em cada nodo computacional (*e.g.*, grau de TLP) (*iii*). Com o *Kubernetes*, o arquivo *YAML* é utilizado para despachar a aplicação, utilizando o *kubectl* ferramenta de linha de comando projetada para gerenciar objetos e ambiente *Kubernetes*, o seguinte comando foi utilizado *kubectl apply -f arquivo.yaml* onde o último parâmetro representa o arquivo de configuração do pod (*iv*). O *control plane* do *Kubernetes* cria um *pod*, que é a menor unidade gerenciável em um ambiente *Kubernetes*, contendo um ou mais *Containers Docker* que executam a aplicação (*v*). O *kube-scheduler* do *Kubernetes* reconhece o *pod* criado e aloca-o para um nodo de trabalho disponível (*vi*). Por fim, o *pod* é executado no nodo de trabalho alocado pelo *kube-scheduler*, permitindo a execução escalável e eficiente da aplicação paralela com o gerenciamento adequado de recursos pelo *Kubernetes* (*vii*) como mostra a Figura 11.

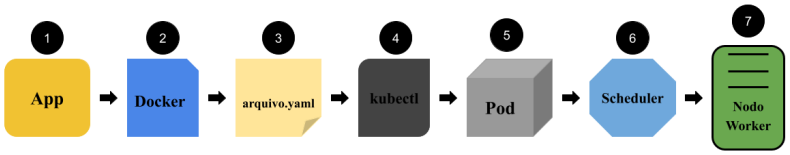


Figura 11 – Processo de encapsulamento e inicialização da leitura do arquivo descritor da aplicação e escalonamento para o nodo de trabalho.

4.2 RESULTADOS EXPERIMENTAIS

Para realizar uma análise completa das aplicações paralelas e encontrar a melhor combinação de número de *threads* e arquitetura, executamos cada aplicação em cada nodo computacional com diferentes números de *threads*, variando de 1 até o número máximo de núcleos disponíveis na arquitetura. Dessa forma, buscamos encontrar o ponto de melhor desempenho para cada aplicação em cada arquitetura, permitindo uma análise completa do comportamento das aplicações em diferentes configurações. A identificação do número ideal de *threads* para cada aplicação é um fator crítico para a otimização do desempenho e eficiência em ambientes de nuvem heterogênea.

Tabela 2 – Número de *threads* encontrado pela busca exaustiva em cada arquitetura, que entrega o melhor resultado de EDP.

	AMD-16	***AMD-24***	***AMD-64***
FFT	12	4	4
NB	6	2	4
HPCG	2	2	2
PO	16	12	24
BT	16	8	32
EP	16	24	64
JA	2	2	2
ST	16	24	64
CG	16	4	4
FT	12	6	8
LU	16	12	20
MG	12	12	4
UA	16	6	64
HS	16	24	32
SP	2	2	2
SC	16	4	12
LUD	16	8	12
BFS	2	2	2
MRI	2	2	2
LBM	16	24	64

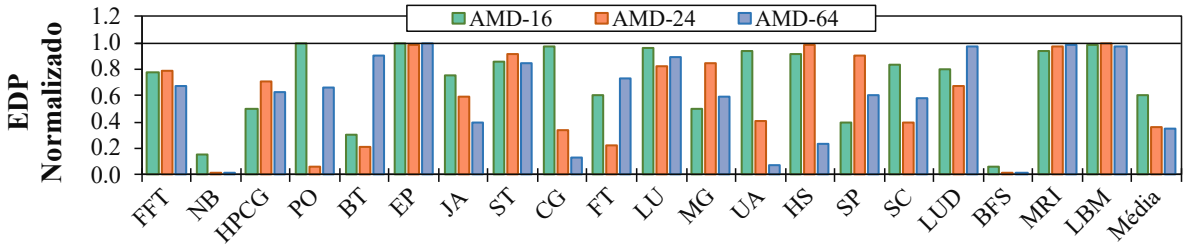


Figura 12 – Melhor resultado encontrado pela busca exaustiva normalizado pela execução padrão. Quanto menor o valor, melhor.

A Tabela 2 destaca o número de *threads* que apresentou o melhor resultado de EDP para cada aplicação e arquitetura multicore. De maneira complementar, a Figura 12 mostra o EDP da execução de cada aplicação com o melhor número de *threads* normalizado pela execução com o número de *threads* igual ao número de núcleos, que corresponde a maneira padrão que aplicações paralelas são executadas. Nesta figura, quanto menor for a barra, melhor (menor) é o EDP. Conforme pode-se observar, uma parte significativa das aplicações tem melhor resultado de EDP quando são executadas com um número de *threads* menor que o número de núcleos. Diferentes são as razões para a falta de escalabilidade, conforme já discutido na Seção 2: sincronização de dados, acessos concorrentes à memória compartilhada e saturação do barramento *off-chip* (SULEMAN; QURESHI; PATT, 2008).

Para melhor entender este cenário, vamos considerar o comportamento da aplicação ***BFS***, que apresenta nível baixo de escalabilidade independente da arquitetura alvo devido a quantidade de operações de sincronização de dados. Para garantir a integridade dos dados compartilhados, é necessário utilizar técnicas de exclusão mútua, como semáforos,

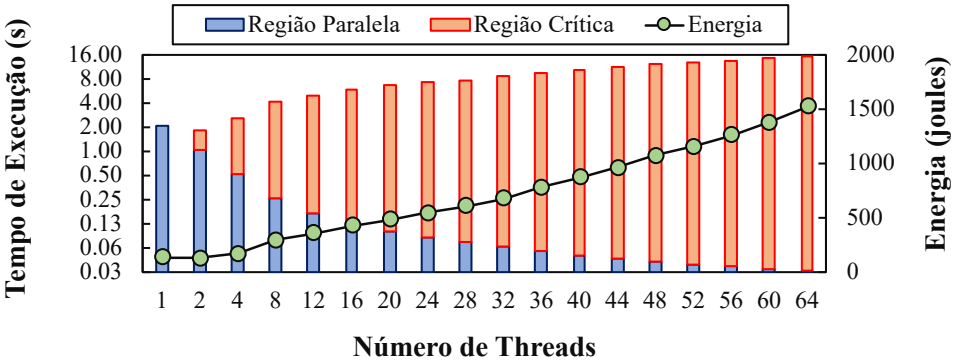


Figura 13 – Escalabilidade da aplicação *BFS* na arquitetura *AMD-64* (tempo em log_2).

locks ou barreiras de sincronização, que permitam que apenas uma *thread* execute a região crítica de cada vez, evitando condições de corrida e garantindo a coerência dos dados compartilhados. Assim, quanto maior o número de *threads* que precisam executar a região crítica, maior será o tempo de sincronização, o que aumentará o tempo de execução e consumo de energia, piorando o EDP resultante. Este comportamento é ilustrado na Figura 13 para a execução da aplicação no AMD-64. Ela destaca para cada número de *threads*, o tempo de execução (eixo y primário) dividido em duas partes: o tempo para executar as regiões paralelas e para sincronizar; e o consumo de energia (eixo y secundário). Conforme observado, a partir da execução com 4 *threads*, inclusive, a sincronização leva mais tempo para executar que a própria execução da região paralela. Portanto, o EDP piora e não é possível atingir melhores resultados com o aumento no número de *threads*.

Adicionalmente, conforme destacado na Tabela 2, o número ideal de *threads* para executar uma dada aplicação muda de acordo com a arquitetura. Por exemplo, a aplicação FT é melhor executada com 12 *threads* no AMD-16, enquanto o melhor resultado em EDP é obtido com 6 e 8 *threads* no AMD-24 e AMD-64, respectivamente. Esta heterogeneidade no melhor número de *threads* está diretamente relacionada às características da aplicação e da arquitetura alvo.

Neste sentido, a Tabela 3 apresenta a arquitetura que entrega o melhor resultado para a execução de cada aplicação paralela de acordo com as métricas avaliadas. A primeira linha da tabela é composta pelas três métricas (tempo, energia e EDP), enquanto a primeira coluna apresenta as 20 aplicações avaliadas. É possível observar que diferentes aplicações apresentam as melhores métricas em diferentes arquiteturas, como é o caso da aplicação BT que tem o melhor tempo de execução na arquitetura AMD-24, o menor consumo de energia na AMD-16, e por fim, o melhor EDP na AMD-64. Por outro lado, há aplicações como o HS que apresentam as melhores métricas na mesma arquitetura (AMD-64).

A Figura 18 compara a execução das aplicações em cada arquitetura alvo em termos de desempenho, consumo de energia e EDP. Cada barra representa o resultado da execução da aplicação em cada arquitetura normalizada pelo melhor resultado obtido entre as três arquiteturas alvo. Portanto, quanto mais próximo de 1.0, melhor o resultado.

Tabela 3 – Melhor arquitetura para executar cada aplicação de acordo com cada métrica avaliada.

	Tempo	**Energia**	**EDP**
FFT	*AMD-24*	*AMD-16*	*AMD-16*
NB	*AMD-64*	*AMD-16*	*AMD-64*
HPCG	*AMD-24*	*AMD-16*	*AMD-24*
PO	*AMD-64*	*AMD-64*	*AMD-64*
BT	*AMD-24*	*AMD-16*	*AMD-64*
EP	*AMD-64*	*AMD-64*	*AMD-64*
JA	*AMD-24*	*AMD-64*	*AMD-24*
ST	*AMD-64*	*AMD-64*	*AMD-64*
CG	*AMD-64*	*AMD-16*	*AMD-64*
FT	*AMD-24*	*AMD-16*	*AMD-24*
LU	*AMD-24*	*AMD-64*	*AMD-24*
MG	*AMD-24*	*AMD-16*	*AMD-24*
UA	*AMD-64*	*AMD-16*	*AMD-64*
HS	*AMD-64*	*AMD-64*	*AMD-64*
SP	*AMD-24*	*AMD-16*	*AMD-16*
SC	*AMD-24*	*AMD-16*	*AMD-24*
LUD	*AMD-24*	*AMD-16*	*AMD-16*
BFS	*AMD-24*	*AMD-16*	*AMD-16*
MRI	*AMD-24*	*AMD-24*	*AMD-24*
LBM	*AMD-24*	*AMD-16*	*AMD-24*

Conforme observado, não existe uma única arquitetura capaz de entregar o melhor resultado para todas as aplicações. Por exemplo, enquanto o melhor desempenho da HPCG é atingido na AMD-24, o melhor desempenho da EP é obtido na AMD-64.

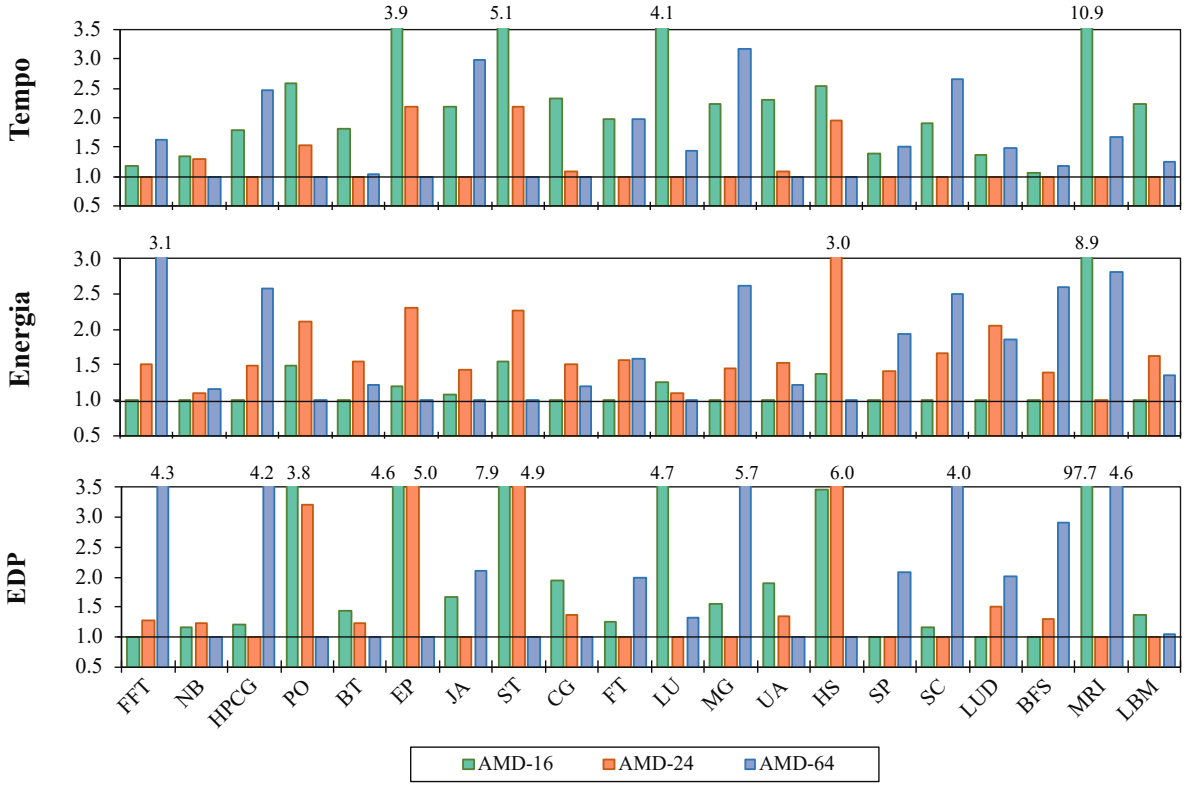
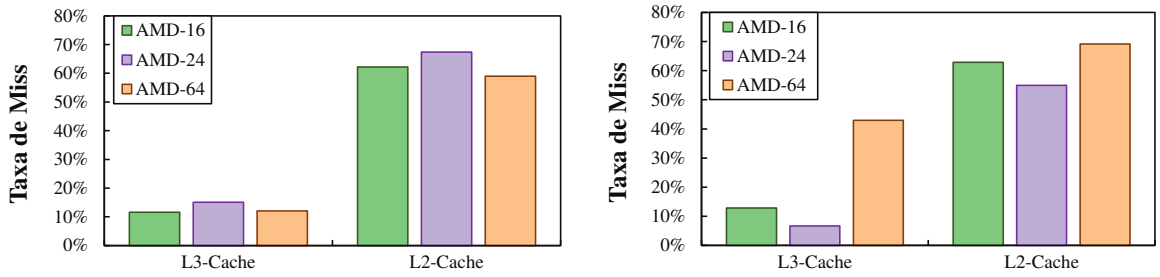


Figura 14 – Resultados de cada aplicação executando com o número ideal de *threads* normalizado pelo melhor resultado obtido entre as arquiteturas.

Um outro ponto importante a se notar é que a arquitetura que entrega o melhor

resultado de desempenho, energia e EDP para a mesma aplicação também muda, como por exemplo, para FFT, NB, HPCG, apenas para citar algumas. Para mostrar este comportamento, consideramos a aplicação FFT. O melhor desempenho é obtido no AMD-24, sendo 9% e 52% melhor quando comparada a execução nas arquiteturas AMD-16 e AMD-64, respectivamente. Por outro lado, o melhor consumo de energia é obtido no AMD-16 com o consumo de energia 49% e 60% menor que o obtido no AMD-24 e AMD-64, respectivamente. Por fim, o melhor EDP também é atingido na AMD-16, com significativa redução comparada às demais arquiteturas.

Para aplicações onde a escalabilidade é limitada pela quantidade de acessos à memória compartilhada (e.g., memória *cache* L3 e principal), a arquitetura que proporcionou o melhor resultado foi aquela que atingiu a menor taxa de *misses* nos níveis de memória *cache* mais distantes do processador. Um exemplo é a aplicação (HPCG), onde seu comportamento relacionado a taxa de *misses* nos níveis da *cache* é apresentado na Figura 15(b). Nela, podemos observar que a arquitetura AMD-24 teve menor taxa de *misses* em ambos os níveis de memória cache, contribuindo assim para entregar um melhor resultado. Por outro lado, para aplicações com boa escalabilidade (*e.g.*, EP, HS e PO), o comportamento de memória tem pouca influencia, uma vez que o mais importante para o desempenho é a quantidade de recursos de *hardware* e o poder computacional disponível. Assim, a arquitetura AMD-64 sobressai com relação às demais arquiteturas. Por fim, a Figura 15(b) ilustra que a taxa de misses entre as três arquiteturas permanece similar para a aplicação (EP).



(a) *EP* (b) *HPCG*

Figura 15 – Comportamento com relação ao número de *misses* nas *caches* L2 e L3.

A determinação da arquitetura alvo e do número de *threads* ideal para a execução de uma aplicação paralela é um desafio crucial na busca pelo melhor desempenho e menor consumo de energia. Como observado no estudo de caso da aplicação FFT, arquiteturas distintas podem apresentar resultados significativamente diversos em termos de desempenho, consumo de energia e EDP para uma mesma aplicação. Assim, é imprescindível dispor de uma ferramenta que possa identificar a arquitetura alvo e o número ideal de *threads* para cada aplicação paralela, levando em conta a heterogeneidade das arquiteturas e as particularidades de cada aplicação e a carga de trabalho. Tal técnica pode propiciar

economia de energia e tempo de execução, além de garantir o melhor desempenho para cada aplicação paralela em um ambiente de computação em nuvem heterogênea.

5 UM FRAMEWORK PARA OTIMIZAR O EDP DE APLICAÇÕES PARALELAS EM AMBIENTES HETEROGÊNEOS

Conforme discutido no Capítulo 4, utilizar uma combinação ideal de número de *threads* e arquitetura *multicore* para executar uma aplicação em um ambiente heterogêneo proporciona ganhos significativos em tempo de execução, energia e EDP. No entanto, quando um lote de aplicações é submetido para execução em um *cluster* heterogêneo, diferentes estratégias podem ser utilizadas para alocar uma aplicação em uma determinada arquitetura. Nesse sentido, este capítulo apresenta o comportamento do *TLP-Allocator* e compara os resultados obtidos com diferentes estratégias.

5.1 TLP-ALLOCATOR

TLP-Alocator objetiva otimizar a relação entre desempenho e consumo de energia, representado pela métrica EDP, da execução de aplicações paralelas em ambientes heterogêneos por meio da seleção da configuração ideal de número de *threads* e arquitetura alvo. *TLP-Allocator* é composto por duas fases principais, descritas nas próximas subseções: *(i)* fase de aprendizado e (*ii*) fase de execução.

5.1.1 FASE DE APRENDIZADO

O processo de aprendizado é composto de quatro etapas, conforme ilustrado na Figura 16: definição do conjunto de treinamento por parte do usuário, extração das características das aplicações, geração do modelo e armazenamento do modelo. As particularidades de cada etapa são descritas a seguir.

Conjunto de treinamento. Nesta etapa, o usuário define e fornece as aplicações utilizadas para treinar o modelo de RNA. Para tanto, *TLP-Allocator* encapsula as aplicações em uma imagem *Docker* que será executada em *containers*, permitindo que elas sejam facilmente executadas em diferentes arquiteturas, garantindo a portabilidade das aplicações. Adicionalmente, *TLP-Allocator* considera que as aplicações fornecidas pelo usuário já estão compiladas e prontas para execução.

Extração de recursos. Nesta etapa, cada aplicação é executada com um número diferente de *threads* (de 1 até o número máximo de *threads* de *hardware* disponível na arquitetura) nas arquiteturas presentes no *cluster*. Durante a execução, são coletadas as seguintes métricas de cada configuração composta por aplicação, número de *threads* e arquitetura alvo:

- Utilização da CPU, que indica o quão bem os núcleos estão sendo aproveitados pelas *threads* em execução. O valor retornado está entre 0 e 1, onde quanto mais próximo de 1, significa que a CPU está sendo melhor utilizada.

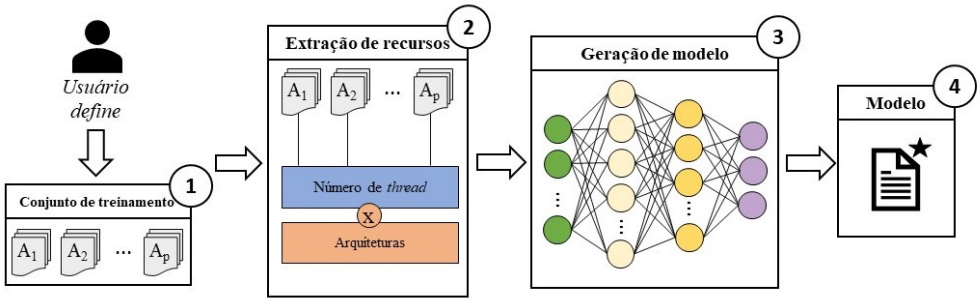


Figura 16 – Fase de aprendizado.

- IPC, que fornece o número de instruções executadas por ciclo de *clock*, permitindo avaliar a eficiência da aplicação na utilização da arquitetura de processamento disponível.

- Taxa de *hit/misses* na memória *cache*, os quais indicam a eficiência do acesso aos dados na memória.

- E as métricas de desempenho (em segundos), energia (em *joules*) e EDP (segundos por *joule*).

Os dados foram retirados diretamente dos contadores de *hardware* disponível em arquiteturas modernas através da ferramenta *AMDuProf*. Essas métricas foram escolhidas pois são importantes para medir o desempenho e a eficiência dos recursos de *hardware* utilizados nas análises de desempenho das aplicações paralelas (DEELMAN et al., 2005): A utilização da CPU nos mostra, indiretamente, a escalabilidade da aplicação paralela, isto é, se uma aplicação executada com o número máximo de *threads* tiver utilização da CPU próximo a 1.0, significa que a aplicação escala de acordo com o número de *threads*. No entanto, quanto mais distante este número for de 1.0, menor é a escalabilidade da aplicação. Assim, esta métrica inclui a informação de escalabilidade no modelo. Tanto o IPC quanto o comportamento de memória mostram o quão CPU-intensiva ou memória-intensiva uma aplicação é, o que, em conjunto com a métrica de utilização de CPU, auxilia em entender o comportamento que limita a escalabilidade da aplicação. Por exemplo, uma aplicação com alta taxa de *misses* no último nível da *cache* e utilização da CPU baixa indica uma aplicação onde o limite de escalabilidade se dá pela tarefa de comunicação entre as *threads*.

Geração do modelo. Com os dados coletados na etapa anterior, esta etapa realiza o treinamento da RNA para geração do modelo. A RNA utilizada possuí:

- Uma camada de entrada, contendo a configuração de execução e as métricas extraídas anteriormente; Considerando o ambiente de experimentos utilizado no Capítulo anterior (3 arquiteturas alvo), a camada de entrada da RNA seria composta de

56 neurônios: 52 para representar a configuração e 4 neurônios para representar as métricas (IPC, CPU, L2 e L3).

- Três camadas intermediárias ocultas [*h1*, *h2*, *h3*], configuradas da seguinte maneira. A cada nível mais interno da camada oculta, o número de neurônios é reduzido pela metade (i.e., 56 na camada de entrada, 28 na camada *h1*, 14 na *h2* e 7 na *h3*. O modelo possui uma função de ativação *ReLU* (IDE; KURITA, 2017), que retorna o próprio valor de entrada se for positivo, ou zero caso contrário. Já a função de ativação *sigmoid* (MARREIROS et al., 2008) (presente na última camada do modelo á saída) é uma função *sigmoide* que retorna um valor entre [0] e [1]. A função *model.fit* é responsável por realizar [100] épocas de treinamento com um tamanho de lote (batch_size) de [8]. Uma época é uma passagem completa por todo o conjunto de dados de treinamento durante o treinamento da rede neural. É possível dividir o conjunto em lotes (ou *batches*) para processá-los de forma mais eficiente. A rede atualiza seus pesos em cada época com base no erro cometido em cada lote. A validação utilizada é (validation_split). O *validation split* define a proporção dos dados que serão usados para validação durante o treinamento, sendo útil para monitorar o desempenho da rede neural em dados nunca vistos durante o treinamento e detectar *overfitting*. Por exemplo, com um *validation split* de [0.2] em um conjunto de [1000] amostras, [800] são usadas para treinamento e [200] para validação.

- uma camada de saída, que contém os neurônios representando a configuração ideal para o dado conjunto de entrada e sendo utilizada para validação do treinamento. Considerando o exemplo utilizado, a camada de saída possuí 52 neurônios que representam a melhor configuração.

Modelo. Por fim, nesta etapa, o modelo gerado após o treinamento é armazenado para ser utilizado durante a fase de execução.

5.1.2 FASE DE EXECUÇÃO

Uma vez que o modelo está treinado, o mesmo estará pronto para ser utilizado durante a fase de execução, que ocorre no momento que o usuário desejar executar qualquer aplicação paralela no ambiente heterogêneo. Para isto, o usuário é responsável por fornecer a aplicação de entrada com seu conjunto de dados e *TLP-Allocator* processa a informação e fornece como resultado a configuração ideal. Assim, esta fase consiste de 3 etapas, conforme destacado na Figura 17 e discutido a seguir.

Entrada (A). Nessa etapa, o usuário fornece a aplicação em um *container* juntamente com o conjunto de entrada da aplicação para *TLP-Allocator*. Esta informação é passada através de linha de comando, como por exemplo, *python3 tlpAllocator.py aplicação argumentos*.

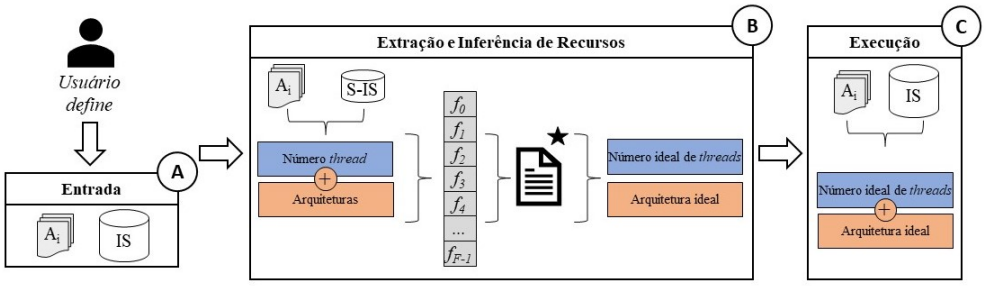


Figura 17 – Fase de execução.

Extração e inferência de recursos (B). Nesta etapa, a aplicação é executada no nodo *master*, e as métricas (IPC, CPU, L2 e L3) são coletadas. As métricas coletadas e a configuração utilizada são a entrada para o modelo gerado na fase anterior. Neste momento, *TLP-Allocator* irá predizer a melhor configuração.

Execução (C). Na etapa de Execução, as configuração de número de *threads* e arquitetura ideal encontradas pela etapa anterior são aplicadas na execução da aplicação, com o objetivo de otimizar o EDP do sistema.

5.1.3 IMPLEMENTAÇÃO

O *TLP-Allocator* foi implementado usando a linguagem Python v.3 e a biblioteca TensorFlow (FOUNDATION, 2022; ABADI et al., 2015). O modelo foi treinado na CPU. A inferência do *TLP-Allocator* também está configurada para ser realizada na CPU.

5.1.4 USANDO O *TLP-ALLOCATOR*

Para utilizar o *TLP-Allocator*, é necessário seguir algumas etapas após baixá-lo. Primeiramente, é preciso ter um sistema operacional Linux com *Python v3* instalado. Em seguida, é necessário possuir o conjunto de aplicações e as métricas avaliadas para realizar a execução do *TLP-Allocator*. Após cumprir esses requisitos, o usuário poderá executar o *TLP-Allocator* para realizar o mapeamento entre aplicação com número ideal de *threads* e arquitetura ideal para a execução da aplicação.

5.2 METODOLOGIA DE AVALIAÇÃO

5.2.1 AMBIENTE DE EXECUÇÃO

O ambiente de execução utilizado para a análise de *TLP-Allocator* é o mesmo ambiente utilizado no Capítulo 4, consistindo de uma configuração de um nodo *master* e três nodos escravos. Os nodos escravos, denominados *AMD-16*, *AMD-24* e *AMD-64*, possuem diferentes arquiteturas de CPU e capacidades de processamento. A máquina *AMD-16* possui 16 núcleos de processamento, enquanto a *AMD-24* tem 24 núcleos de

processamento e a *AMD-64* tem 64 núcleos de processamento. Essas máquinas heterogêneas são interconectadas e permitem que os experimentos sejam executados de forma distribuída.

5.2.2 CONJUNTO DE TREINAMENTO

Nós utilizamos 20 aplicações com diferentes características de grau de exploração de paralelismo no nível de *threads*, IPC e acessos à memória *cache* para realizar a fase de treinamento. conforme destacado na Tabela 4, as aplicações cobrem uma vasta gama de valores de IPC, variando de valores baixos, médios e altos (*e.g.*, aplicação *PO* IPC 2.30, aplicação *HS* IPC 1.03 e aplicação *MRI* IPC 0.21 na arquitetura *AMD-16* respectivamente), e também apresentam diferentes comportamentos de acessos á memória (*e.g.*, aplicação *NB miss-L3* 0.58, aplicação *UA* miss-L3 0.34 e Aplicação *SC miss-L3* 0.10 na arquitetura *AMD-64* respectivamente).

Tabela 4 – Características de cada aplicação em cada arquitetura.

	AMD-16		AMD-24		AMD-64	
	IPC	Miss-L3	IPC	Miss-L3	IPC	Miss-L3
CUTCP	1.36	0.21	1.21	0.10	1.47	0.33
NB	1.06	0.19	1.15	0.10	0.33	0.58
HPCG	0.64	0.31	0.56	0.09	0.21	0.47
PO	2.30	0.18	2.45	0.10	2.92	0.54
SGEMM	0.97	0.12	1.64	0.09	1.26	0.29
EP	0.47	0.20	0.49	0.16	0.59	0.47
JA	0.33	0.24	0.37	0.08	0.10	0.53
ST	0.18	0.21	0.21	0.08	0.07	0.10
CG	0.64	0.13	0.49	0.12	0.42	0.47
FT	1.34	0.22	1.34	0.10	0.58	0.40
LU	1.13	0.21	1.05	0.13	0.78	0.36
SPMV	1.19	0.19	1.15	0.12	1.00	0.35
SP	0.78	0.18	0.82	0.12	0.45	0.35
UA	0.73	0.24	0.85	0.12	0.87	0.34
HS	1.03	0.22	1.07	0.15	1.46	0.27
SC	0.96	0.20	0.83	0.09	0.38	0.10
LUD	1.42	0.17	1.04	0.12	0.79	0.31
BFS	1.01	0.21	1.09	0.07	0.31	0.41
MRI	0.21	0.26	0.24	0.08	0.13	0.47
TPACF	0.55	0.19	0.12	0.12	0.12	0.29

5.2.3 CONJUNTO DE VALIDAÇÃO

Para validar *TLP-Allocator*, nós utilizamos quatro aplicações que não compõem o conjunto de treinamento e possuem diferentes características em termos de IPC médio e comportamento de acesso à memória: *FFT, BT-NAS, MG-NAS, LBM*, que são destacados a seguir na Tabela 5.

Estas aplicações são representativas e foram escolhidas por apresentarem uma variedade de características em termos de IPC médio e comportamento de acesso à memória, permitindo uma validação mais completa do *TLP-Allocator*. A aplicação *FFT*, por exemplo, é caracterizada por ter o menor IPC médio entre as quatro aplicações avaliadas e um alto número de *misses* na *cache* L3. Já a aplicação *BT-NAS* possui o maior IPC médio

Tabela 5 – Características das aplicações utilizadas para validar o *TLP-Allocator*.

	AMD-16		AMD-24		AMD-64	
	IPC	Miss-L3	IPC	Miss-L3	IPC	Miss-L3
FFT	0.53	0.19	0.52	0.13	0.25	0.53
BT-NAS	1.06	0.19	1.15	0.10	0.33	0.58
MG-NAS	0.64	0.31	0.56	0.09	0.21	0.47
LBM	2.30	0.18	2.45	0.10	2.92	0.54

entre as aplicações e o menor número de *misses* na *cache* L3. A aplicação *MG-NAS* é intermediária em termos de IPC e *misses* na *cache* L3. Por fim, a aplicação *LBM* é caracterizada por ter o segundo maior IPC médio e o maior número de *misses* na *cache* L3 entre as aplicações avaliadas. Com essa diversidade de comportamentos, as aplicações selecionadas são representativas e fornecem um conjunto de casos de uso para validar a eficácia do *TLP-Allocator*.

5.2.4 ESTRATÉGIAS COMPARADAS

Os resultados obtidos pelo emprego do *TLP-Allocator* para executar as aplicações do conjunto de validação foram comparadas com as seguintes abordagens:

- *AMD-16*, *AMD-24* e *AMD-64*, onde cada aplicação do conjunto de validação foi executada em cada arquitetura alvo com o número de *threads* igual ao número máximo de *threads*, isto é, a maneira padrão na qual as aplicações paralelas são executadas.

- *Melhor-AMD-16*, *Melhor-AMD-24* e *Melhor-AMD-64*, que considera o melhor resultado obtido em cada arquitetura alvo. *Melhor Estático*, que, através de uma busca exaustiva, considera a melhor atribuição entre aplicação, grau de paralelismo e arquitetura.

- *Kube-Scheduler*, o escalonador padrão do *Kubernetes*, onde as aplicações são distribuídas entre os nodos através da política *round-robin*. Na política de *round-robin*, o *kube-scheduler* seleciona os nós disponíveis em uma ordem circular, em que cada nó é escolhido em sequência, garantindo que o agendamento seja feito de forma equilibrada entre os nós do *cluster* (Kubernetes, 2023).

5.3 RESULTADOS EXPERIMENTAIS

Nesta seção, apresentamos os resultados obtidos por meio da avaliação da solução desenvolvida. Inicialmente, analisamos os resultados referentes EDP. Em seguida, avaliaremos a acurácia da solução como um todo, ou seja, o quão próximo o *TLP-Alllocator* ficou da solução ideal.

5.3.1 ANÁLISE DO EDP

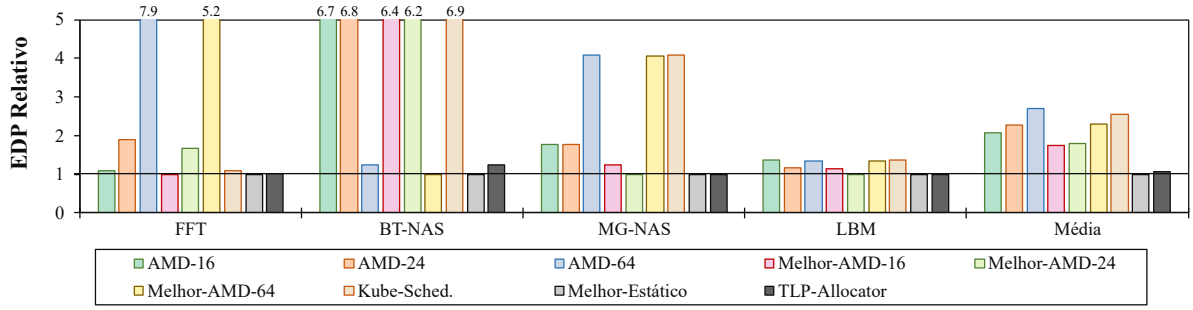


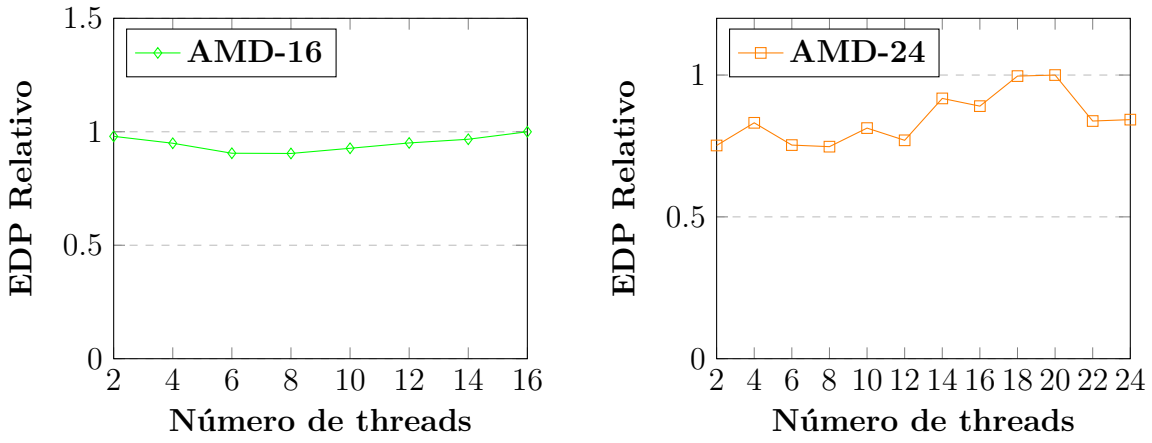
Figura 18 – Resultados de EDP de cada aplicação executando com as diferentes estratégias.

A Figura 18 apresenta o resultado do EDP das aplicações - *FFT, BT-NAS, MG-NAS* e *LBM* - executadas com as diferentes estratégias e com o melhor número de *threads* em cada arquitetura, conforme descritas na Seção 5.2.4, no ambiente de execução. O EDP de cada estratégia é normalizado pela execução com o número de *threads* igual ao número de núcleos, que corresponde à maneira padrão que aplicações paralelas são executadas. A linha preta na figura representa o resultado obtido pela solução "*Melhor-Estático*". Assim, quanto mais próximo da linha, melhor é o resultado, e quanto mais acima o resultado está, pior é o EDP. Através desta figura, é possível comparar o EDP das aplicações nas diferentes arquiteturas e técnicas de alocação, identificando a estratégia mais eficiente em termos de EDP para cada aplicação.

Considerando a aplicação *FFT*, observa-se que as estratégias *AMD-16, Melhor-AMD-16, Kube-Sched e Melhor-estático* apresentaram um bom desempenho em termos de EDP, enquanto as estratégias *AMD-24* e *Melhor-AMD-24* tiveram resultados medianos. Por outro lado, as estratégias AMD-64 e *Melhor-AMD-64* apresentaram os piores resultados de EDP. Em relação à estratégia com o *TLP-Allocator*, ela apresentou EDP igual ao da execução ideal, significando que conseguiu encontrar a configuração ideal para executar a aplicação.

Para compreender melhor este cenário, vamos analisar o comportamento da aplicação *FFT* nas diferentes arquiteturas. *FFT* é uma aplicação que apresenta baixo nível de escalabilidade, independentemente da arquitetura utilizada, devido à grande quantidade de operações de sincronização de dados. Este comportamento pode ser visto na Figura 19, onde até um certo ponto da execução com maior número *threads* igual a 8 na AMD-16 e AMD-24 o EDP diminui, porém a partir deste ponto o EDP é piorado com o número maior de *threads*. Isto ocorre pois quanto maior o número de *threads* que precisam executar a região crítica, maior será o tempo de sincronização.

Com base no cenário anterior, é possível observar que o aumento do número de *threads* nem sempre resulta em um melhor desempenho para uma aplicação. Na Figura 20, podemos visualizar claramente este cenário, mesmo com o aumento do número de *threads*, o EDP aumenta. Essa observação indica que a utilização de um número excessivo



(a) *AMD-16* (b) *AMD-24*

Figura 19 – Resultado do EDP da aplicação *FFT* executada naS arquiteturaS *AMD-16* e *AMD-24* com diferentes números de *threads* o EDP está normalizado.

de *threads* pode gerar gargalos, o que pode prejudicar o desempenho da aplicação em tempo e consumo de energia assim ocasionando um alto EDP.

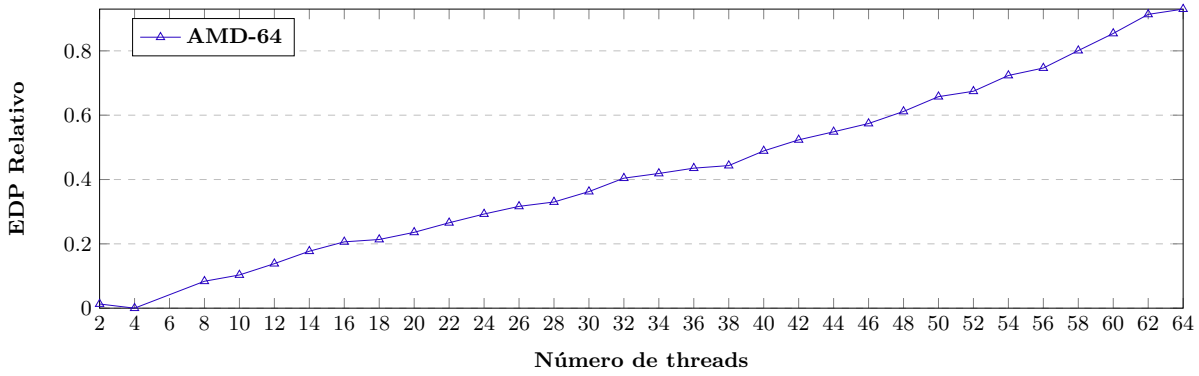


Figura 20 – Resultado do EDP da aplicação FFT executada na arquitetura AMD-64 com diferentes números de *threads* o EDP está normalizado.

Ao analisar os resultados da aplicação *BT-NAS* na Figura 18, podemos constatar que as estratégias *AMD-16, AMD-24, Melhor-AMD-16, Melhor-AMD-24* e *Kube-Sched* apresentam os piores resultados em relação ao EDP. Por outro lado, as estratégias *AMD-64* e *TLP-Allocator* apresentam um desempenho mediano em relação ao EDP. Por fim, pode-se observar que a estratégia textit Melhor-AMD-64 apresenta melhor EDP. Isso se deve ao fato de que a aplicação *BT-NAS* consegue ter uma melhor escalabilidade na arquitetura *AMD-64*. E isto poder ser visto na Figura 21, onde quanto maior número de *threads* menor o EDP.

Ao analisar as estratégias para a aplicação *MG-NAS* na Figura 18, pode-se observar que as estratégias *AMD-64, Melhor-AMD-64* e *Kube-Sched* apresentaram os piores resultados em relação ao EDP. Isso ocorre porque a aplicação *MG-NAS* não consegue obter um melhor desempenho ao ser executada com um número de *threads* igual ao número de cores da arquitetura. Já as estratégias *AMD-16, AMD-24* e *Melhor-AMD-16*

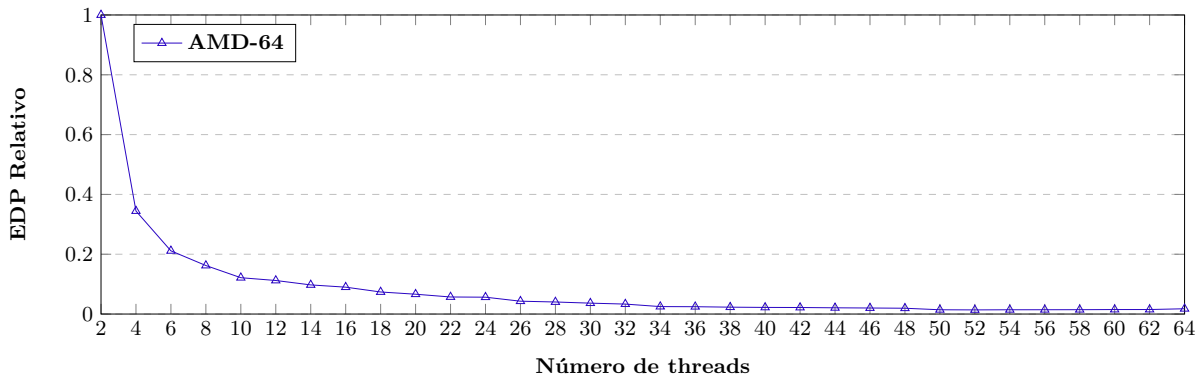


Figura 21 – Resultado do EDP da aplicação *BT-NAS* executada na arquitetura AMD-64 com diferentes números de *threads* o EDP está normalizado.

apresentaram um desempenho mediano. Por outro lado, as estratégias *Melhor-AMD-24*, *Melhor-Estático* e *TLP-Allocator* apresentaram os melhores resultados em relação ao EDP para a aplicação em questão. Que também pode ser visto na Figura 22, onde com o número de 12 *threads* se obtém o menor EDP. É importante destacar que a estratégia *TLP-Allocator* apresentou um desempenho semelhante ao ideal, indicando que esta estratégia pode ser uma boa opção para otimizar a alocação de recursos para a aplicação em questão.

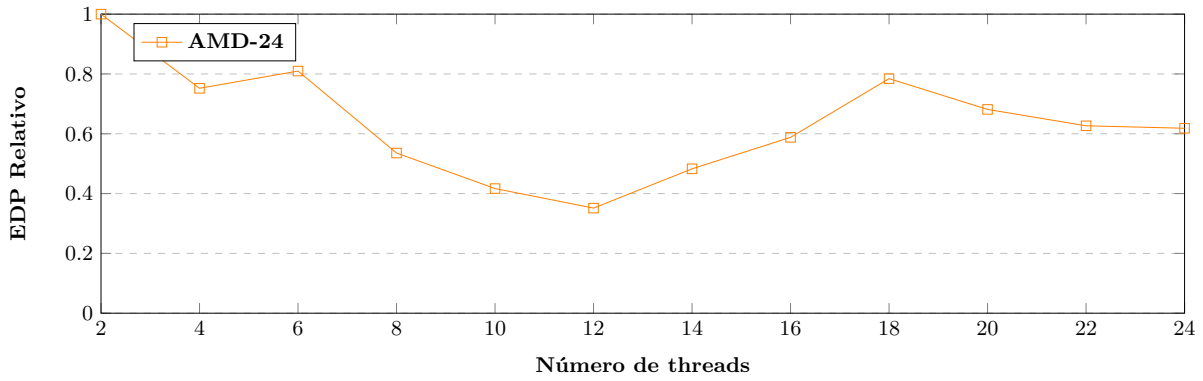


Figura 22 – Resultado do EDP da aplicação *MG-NAS* executada na arquitetura AMD-24 com diferentes números de *threads* o EDP está normalizado.

Baseado na análise da Figura 18, para a aplicação *LBM*, pode-se observar que as estratégias de alocação *Melhor-AMD-24, Melhor-Estático* e *TLP-Allocator* apresentaram melhor desempenho em relação ao EDP. A Figura 23 ilustra este comportamento, onde a EDP até a *thread* 12 decai após o EDP tem comportamento quase linear. As estratégias *AMD-24* e *Melhor-AMD-16* tiveram desempenho mediano de EDP, enquanto as estratégias *AMD-16, AMD-64, Melhor-AMD-64* e *Kube-Sched* apresentaram os piores resultados em EDP. As estratégias com melhor desempenho em relação ao EDP para a aplicação *LBM* foram as estratégias Melhor-Estático e *TLP-Allocator* que convergem para a alocação ideal.

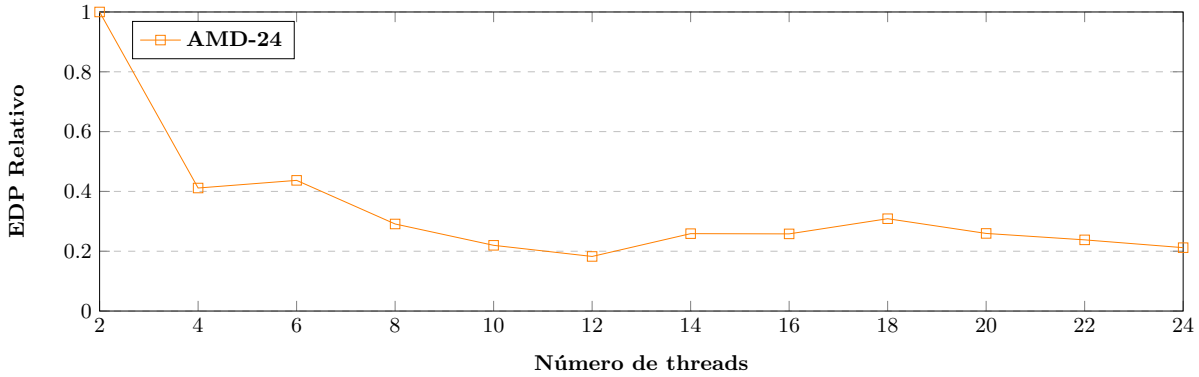


Figura 23 – Resultado do EDP da aplicação *LBM* executada na arquitetura AMD-24 com diferentes números de *threads* o EDP está normalizado.

5.3.2 ACURÁCIA DO *TLP-ALLOCATOR*

As Figuras 24, 25, 26 e 27, apresentam o resultados de tempo de execução e energia obtidos da execução das aplicações *FFT, NAS-BT, MG-NAS e LBM*. Nela pode-se observar, a relação entre o consumo de energia em Joules e o tempo de execução em segundos das aplicações avaliadas para validar o *TLP-Allocator* nas métricas de grau de TLP, IPC e acessos à memória *cache* L3.

As figuras 24, 25, 26 e 27 possuem as seguintes características em comum: elas ilustram o tempo, e o consumo de energia das quatro aplicações mencionadas anteriormente e executadas nas diferentes estratégias. Onde o eixo x representa o tempo de execução em (segundos), e o eixo y reapresenta o consumo de energia em (*joules*) e cada simbolo dentro da figura corresponde a um número de *threads* nas diferentes arquiteturas.

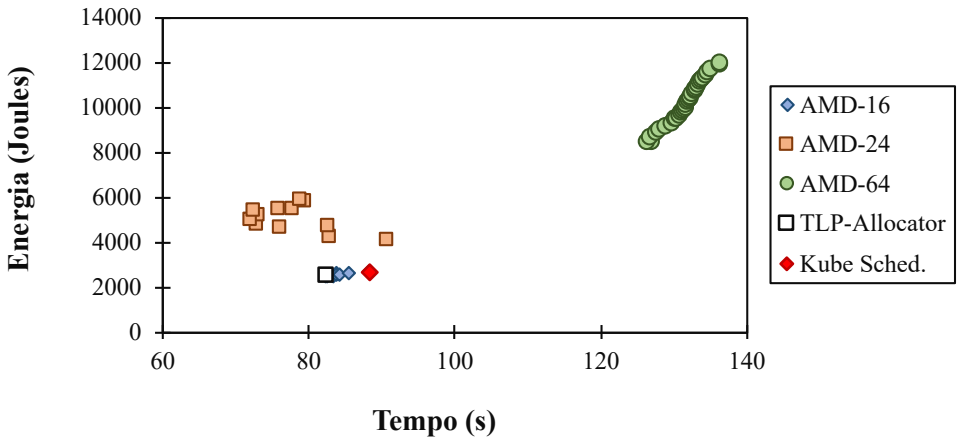


Figura 24 – Resultado de tempo de execução e consumo de energia para á aplicação *FFT*.

Ao analisar o espaço de exploração do projeto na Figura 24, pode-se constatar que os resultados apresentados pelo *TLP-Allocator* são bons. Nesse espaço, o TLP-Allocator consegue encontrar o melhor caso ou, no mínimo, o segundo melhor caso em termos de desempenho e consumo de energia, demonstrando a efetividade dessa abordagem na otimização da aplicação *FFT*.

De acordo com a observação da Figura 25, o *TLP-Allocator* conseguiu encontrar a segunda melhor combinação de desempenho e consumo de energia para a aplicação *BT-NAS*. Como pode ser visto na figura, o primeiro melhor resultado foi obtido pelo *AMD-64*, que apresentou um desempenho superior em comparação com todas as outras técnicas. No entanto, o *TLP-Allocator* foi capaz de encontrar uma configuração de parâmetros que se aproximou muito do desempenho do *AMD-64*, demonstrando sua eficiência e capacidade de encontrar soluções ótimas.

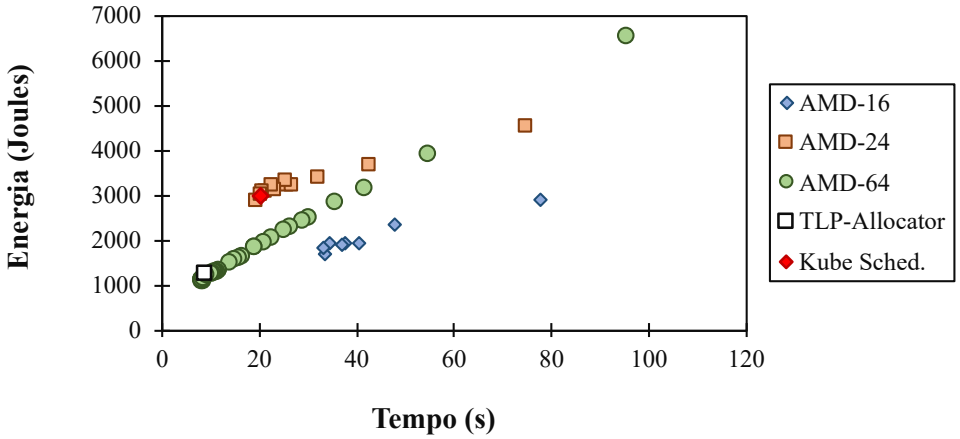


Figura 25 – Resultado de tempo de execução e consumo de energia para á aplicação *BT-NAS*.

Ao analisarmos os resultados obtidos execução da aplicação *MG-NAS*, que pode ser vista na Figura 26. O *TLP-Allocator*, foi capaz de encontrar a terceira melhor configuração de parâmetros, mas com um desempenho tão próximo do segundo melhor resultado *AMD-24*, que não há diferença estatisticamente significativa entre eles, e em termos de desempenho o *TLP-Allocator* se sobressai em relação as demais.

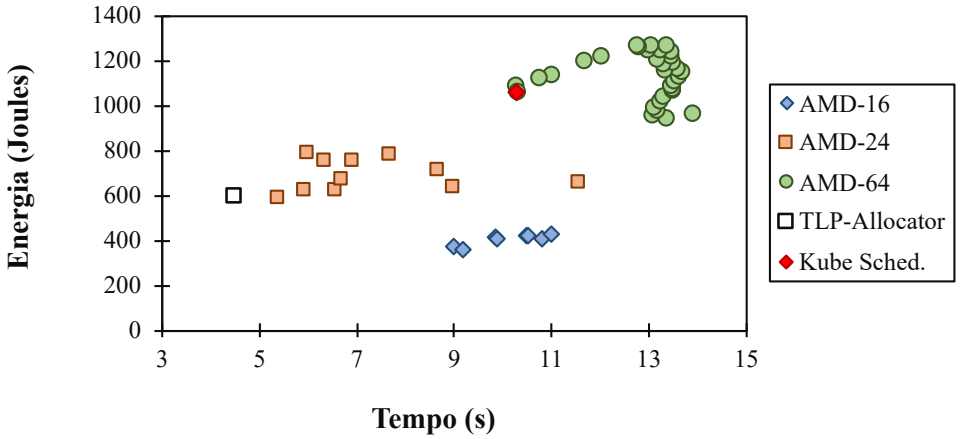


Figura 26 – Resultado de tempo de execução e consumo de energia para á aplicação *MG-NAS*.

Por fim, os resultados da aplicação *LBM* foram analisados na Figura 27. Nesta aplicação, é possível observar que o *TLP-Allocator*, comparado com as outras estratégias,

consegue encontrar o quarto melhor resultado em consumo de energia. Entretanto, em relação ao desempenho, o *TLP-Allocator* é o primeiro colocado.

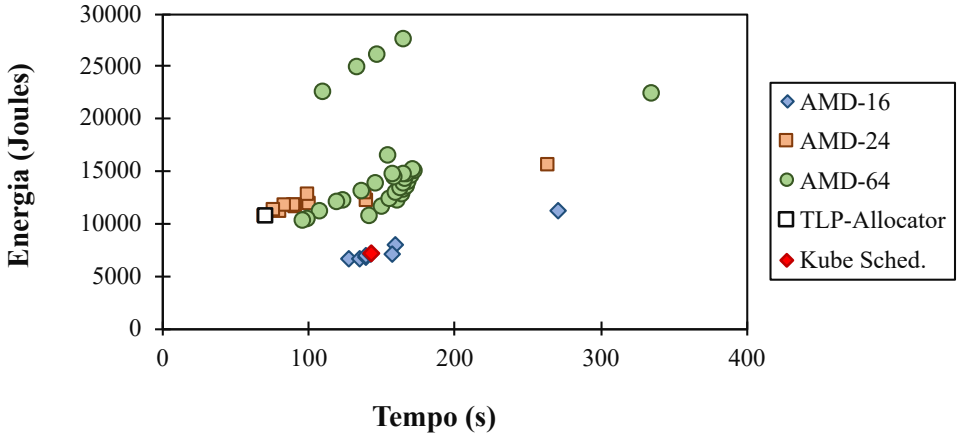


Figura 27 – Resultado de tempo de execução e consumo de energia para á aplicação *LBM*.

Após a análise do espaço de exploração de projeto, pode-se concluir que o *TLP-Allocator* apresenta ótimos resultados na otimização da aplicação em questão. Uma vez que o modelo é treinado para determinar a melhor combinação de número de *threads* e arquitetura alvo, basta executá-lo uma vez no nodo *master* para definir o número de *threads* e equalizar a máquina ideal do *cluster*. Esse processo é muito mais eficiente do que realizar a mesma tarefa por meio da força bruta, que demandaria a execução de todas as aplicações nas três arquiteturas alvo, totalizando 320 execuções na *AMD-16*, 480 na *AMD-24* e 1280 na *AMD-64*. O *TLP-Allocator*, por sua vez, só precisa ser executado uma vez para definir a melhor combinação, o que torna a abordagem ainda mais vantajosa em termos de eficiência.

6 CONCLUSÃO

Nesta dissertação, apresentamos o *TLP-Allocator*, uma estratégia para otimizar a execução de aplicações paralelas em ambiente de nuvem heterogêneo baseada no grau de TLP + arquitetura alvo ideal, implementada em cima da tecnologia *Kubernetes*. O *TLP-Allocator* é completamente transparente para usuários finais e clientes. Dado um conjunto de aplicações paralelas serem executados, ele encontra grau de TLP ideal para cada aplicação e realiza *match* entre aplicação e arquitetura ideal, para melhor aproveitamento dos recursos disponíveis.

Demonstramos que o *TLP-Allocator* alcança um melhor EDP do que a execução padrão de aplicações paralelas executando em ambiente de nuvem heterogênea, onde as aplicações são executas com o número de *threads* igual ao número de cores na arquitetura, nos também mostramos que essa abordagem não necessariamente fornecera o melhor desempenho, economia de energia e resultado de EDP. Também mostramos que o *match* entre aplicação ideal e arquitetura alvo ideal podem ter um ganho significado de desempenho e EDP pois não considera as características de grau de TLP e a arquitetura.

Embora não tenha sido abordado nesta dissertação, é importante ressaltar que o *TLP-Allocator* pode ser utilizado em ambientes homogêneos. Ainda que a principal motivação desta dissertação tenha sido a busca por um melhor desempenho e eficiência energética em ambientes heterogêneos, o *TLP-Allocator* é capaz de encontrar o grau de paralelismo ideal para as aplicações paralelas executando em ambientes homogêneos.

Como trabalho futuros, pretendemos integrar *TLP-Allocator* em diferentes arquiteturas heterogêneas (*e.g.*, CPU, GPU, FPGAs e Aceleradores) para que aplicações que possuem diferentes características que possam se beneficiar das diferentes arquiteturas. Assim como explorar as diversas interfaces de programação paralelas (*e.g.*, OpenMP, OpenACC, OpenCL e Intel OneAPI). Além disso, também pretendemos melhorar ainda mais nossa estratégia de aprendizado para considerar as diferentes arquiteturas heterogêneas durante a fase de aprendizado.

6.1 LISTA DE PUBLICAÇÕES

- Simpósio em Sistemas Computacionais de Alto Desempenho (WSCAD/2022) (LIMA et al., 2022).
- Artigo publicado no *journal Concurrency and Computation Practice and Experience* (SILVA et al., 2021).
- Um artigo em processo de revisão para a IEEE *Transactions on Parallel and Distributed Systems* (TPDS), em que na primeira revisão o artigo recebeu *major review*.
- Adicionalmente, os resultados da dissertação estão sendo preparados para submissão ao *Journal of Parallel and Distributed Computing* (JPDC).

REFERÊNCIAS

ABADI, M. et al. Tensorflow: Large-scale machine learning on heterogeneous systems. *arXiv preprint arXiv:1603.04467*, 2015. Citado na página 58.

ABBASI, S. I. et al. Affinity-based task scheduling on heterogeneous multicore systems using cbs and qbictm. *Applied Sciences*, MDPI, v. 11, n. 12, p. 5740, 2021. Citado na página 42.

BAILEY, D. H. et al. The nas parallel benchmarks and summary and preliminary results. In: *ACM/IEEE SC*. USA: ACM, 1991. p. 158–165. ISBN 0-89791-459-7. Citado na página 45.

BARHAM, P. et al. Xen and the art of virtualization. *SIGOPS Oper. Syst. Rev.*, Association for Computing Machinery, New York, NY, USA, v. 37, n. 5, p. 164–177, out. 2003. ISSN 0163-5980. Citado na página 30.

BI, J. et al. Deep neural networks for predicting task time series in cloud computing systems. In: *2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC)*. [S.l.: s.n.], 2019. p. 86–91. Citado 2 vezes nas páginas 24 e 42.

CHAI, L.; GAO, Q.; PANDA, D. K. Understanding the impact of multi-core architecture in cluster computing: A case study with intel dual-core system. In: IEEE. *Seventh IEEE international symposium on cluster computing and the grid (CCGrid'07)*. [S.l.], 2007. p. 471–478. Citado na página 31.

CHAPMAN, B.; JOST, G.; PAS, R. V. D. *Using OpenMP: portable shared memory parallel programming*. [S.l.]: MIT press, 2007. Citado na página 28.

CHARR, J.-C. et al. Energy consumption reduction with dvfs for message passing iterative applications on heterogeneous architectures. In: IEEE. *2015 IEEE International Parallel and Distributed Processing Symposium Workshop*. [S.l.], 2015. p. 922–931. Citado na página 40.

CHE, S. et al. Rodinia: A benchmark suite for heterogeneous computing. In: *IEEE Int. Symp. on Workload Characterization*. DC, USA: IEEE Computer Society, 2009. p. 44–54. ISBN 978-1-4244-5156-2. Citado na página 45.

CHO, Y.; GUZMAN, C. A. C.; EGGER, B. Maximizing system utilization via parallelism management for co-located parallel applications. In: *Proceedings of the 27th International Conference on Parallel Architectures and Compilation Techniques*. [S.l.: s.n.], 2018. p. 1–14. Citado na página 35.

CHOI, R. Y. et al. Introduction to machine learning, neural networks, and deep learning. *Translational Vision Science & Technology*, The Association for Research in Vision and Ophthalmology, v. 9, n. 2, p. 14–14, 2020. Citado 2 vezes nas páginas 35 e 36.

CUI, Z.; CHEN, W.; CHEN, Y. Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995*, 2016. Citado na página 37.

CULLER, D.; SINGH, J. P.; GUPTA, A. *Parallel computer architecture: a hardware/software approach*. [S.l.]: Gulf Professional Publishing, 1999. Citado na página 27.

CURTIS-MAURY, M. et al. Online power-performance adaptation of multithreaded programs using hardware event-based prediction. In: *Proceedings of the 20th annual international conference on Supercomputing*. [S.l.: s.n.], 2006. p. 157–166. Citado na página 39.

DEELMAN, E. et al. Pegasus: A framework for mapping complex scientific workflows onto distributed systems. *Scientific Programming*, IOS Press, v. 13, n. 3, p. 219–237, 2005. Citado na página 56.

Docker. *Docker - Building, Shipping, and Running Distributed Applications*. 2023. <https://www.docker.com/>. Citado na página 31.

DONGARRA, J.; HEROUX, M. A.; LUSZCZEK, P. Hpcg benchmark: A new metric for ranking high performance computing systems. *Knoxville, Tennessee*, 2015. Citado na página 46.

ETINSKI, M. et al. Utilization driven power-aware parallel job scheduling. *Computer Science-Research and Development*, Springer, v. 25, n. 3, p. 207–216, 2010. Citado 2 vezes nas páginas 39 e 43.

FOUNDATION, P. S. *Python*. 2022. Disponível em: <https://www.python.org/>. Citado na página 58.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT press, 2016. Citado na página 25.

GUERRERO, C.; LERA, I.; JUIZ, C. Genetic algorithm for multi-objective optimization of container allocation in cloud architecture. *Journal of Grid Computing*, Springer, v. 16, p. 113–135, 2018. Citado na página 41.

HACKENBERG, D. et al. Power measurement techniques on standard compute nodes: A quantitative comparison. In: *IEEE ISPASS*. [S.l.: s.n.], 2013. p. 194–204. Citado 2 vezes nas páginas 46 e 48.

HAM, T. J. et al. Disintegrated control for energy-efficient and heterogeneous memory systems. In: *IEEE HPCA*. [S.l.: s.n.], 2013. p. 424–435. ISSN 1530-0897. Citado na página 29.

HARRIS, T.; MAAS, M.; MARATHE, V. J. Callisto: Co-scheduling parallel runtime systems. In: *Proceedings of the Ninth European Conference on Computer Systems*. [S.l.: s.n.], 2014. p. 1–14. Citado na página 40.

HASTIE, T. et al. *The elements of statistical learning: data mining, inference, and prediction*. [S.l.]: Springer, 2009. v. 2. Citado na página 37.

HOTTA, Y. et al. Profile-based optimization of power performance by using dynamic voltage scaling on a pc cluster. In: IEEE. *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*. [S.l.], 2006. p. 8–pp. Citado na página 39.

HUANG, J.; XIAO, C.; WU, W. Rlsk: A job scheduler for federated kubernetes clusters based on reinforcement learning. In: *2020 IEEE International Conference on Cloud Engineering (IC2E)*. [S.l.: s.n.], 2020. p. 116–123. Citado 2 vezes nas páginas 42 e 43.

IDE, H.; KURITA, T. Improvement of learning for cnn with relu activation by sparse regularization. In: IEEE. *2017 international joint conference on neural networks (IJCNN)*. [S.l.], 2017. p. 2684–2691. Citado na página 57.

IOSUP, A. et al. Performance analysis of cloud computing services for many-tasks scientific computing. *IEEE Transactions on Parallel and Distributed systems*, IEEE, v. 22, n. 6, p. 931–945, 2011. Citado na página 32.

KAPUR, R. A workload balanced approach for resource scheduling in cloud computing. In: IEEE. *2015 eighth international conference on contemporary computing (IC3)*. [S.l.], 2015. p. 36–41. Citado na página 33.

KIRK, D. B.; WEN-MEI, W. H. *Programming massively parallel processors: a hands-on approach*. [S.l.]: Morgan kaufmann, 2016. Citado na página 28.

KOVÁCS, Z. L. *Redes neurais artificiais*. [S.l.]: Editora Livraria da Fisica, 2002. Citado na página 36.

KRZYWANIAK, A.; CZARNUL, P.; PROFICZ, J. Depo: A dynamic energy-performance optimizer tool for automatic power capping for energy efficient high-performance computing. *Software: Practice and Experience*, Wiley Online Library, v. 52, n. 12, p. 2598–2634, 2022. Citado na página 29.

Kubernetes. *Kubernetes - A Plataforma de Orquestração de Contêineres*. 2023. <https://kubernetes.io/pt-br/>. Citado 4 vezes nas páginas 31, 32, 34 e 60.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group UK London, v. 521, n. 7553, p. 436–444, 2015. Citado na página 36.

LIMA, E. C. de et al. Otimizando a execuçao de aplicaçoes paralelas em ambiente de nuvem heterogênea. In: SBC. *Anais do XXIII Simpósio em Sistemas Computacionais de Alto Desempenho*. [S.l.], 2022. p. 181–192. Citado na página 67.

LIU, F. et al. *NIST Cloud Computing Reference Architecture: Recommendations of the National Institute of Standards and Technology*. USA: CreateSpace Independent Publishing Platform, 2012. ISBN 1478168021, 9781478168027. Citado na página 30.

LUCKERT, M.; SCHAEFER-KEHNERT, M. *Using machine learning methods for evaluating the quality of technical documents*. 2016. Citado na página 35.

MAGHSOUD, Z.; NOORI, H.; MOZAFFARI, S. P. Peps: Predictive energy-efficient parallel scheduler for multi-core processors. *The Journal of Supercomputing*, Springer, v. 77, n. 7, p. 6566–6585, 2021. Citado 2 vezes nas páginas 42 e 43.

MAKRANI, H. M. et al. Energy-aware and machine learning-based resource provisioning of in-memory analytics on cloud. In: *Proceedings of the ACM Symposium on Cloud Computing*. [S.l.: s.n.], 2018. p. 517–517. Citado 2 vezes nas páginas 41 e 43.

MáRQUEZ, G.; VILLEGAS, M. M.; ASTUDILLO, H. A pattern language for scalable microservices-based systems. In: *ECSA*. NY, USA: ACM, 2018. ISBN 9781450364836. Citado na página 30.

MARREIROS, A. C. et al. Population dynamics: variance and the sigmoid activation function. *Neuroimage*, Elsevier, v. 42, n. 1, p. 147–157, 2008. Citado na página 57.

MASANET, E. et al. Recalibrating global data center energy-use estimates. *Science*, American Association for the Advancement of Science, v. 367, n. 6481, p. 984–986, 2020. Citado na página 23.

MOLNAR, C. *Interpretable machine learning*. [S.l.]: Lulu. com, 2020. Citado 2 vezes nas páginas 36 e 37.

PATTERSON, D. A.; HENNESSY, J. L. *Computer organization and design ARM edition: the hardware software interface*. [S.l.]: Morgan kaufmann, 2016. Citado na página 27.

PATTERSON, D. A.; HENNESSY, J. L.; GOLDBERG, D. *Computer architecture: a quantitative approach*. [S.l.]: Morgan Kaufmann San Mateo, CA, 1990. v. 2. Citado na página 27.

PUSUKURI, K. K.; GUPTA, R.; BHUYAN, L. N. Thread reinforcer: Dynamically determining number of threads via os level monitoring. In: IEEE. *2011 IEEE International Symposium on Workload Characterization (IISWC)*. [S.l.], 2011. p. 116–125. Citado na página 40.

QASAIMEH, M. et al. Comparing energy efficiency of cpu, gpu and fpga implementations for vision kernels. In: IEEE. *2019 IEEE international conference on embedded software and systems (ICESS)*. [S.l.], 2019. p. 1–8. Citado na página 29.

QIAO, Y. et al. Fpga-accelerated deep convolutional neural networks for high throughput and energy efficiency. *Concurrency and Computation: Practice and Experience*, Wiley Online Library, v. 29, n. 20, p. e3850, 2017. Citado na página 35.

SAHNI, J.; VIDYARTHI, D. P. A cost-effective deadline-constrained dynamic scheduling algorithm for scientific workflows in a cloud environment. *IEEE Transactions on Cloud Computing*, IEEE, v. 6, n. 1, p. 2–18, 2015. Citado na página 33.

SANGEETHA, S. B. et al. Resource management framework using deep neural networks in multi-cloud environment. *Operationalizing Multi-Cloud Environments: Technologies, Tools and Use Cases*, Springer, p. 89–104, 2022. Citado 2 vezes nas páginas 24 e 42.

SILVA, V. S. da et al. Smart resource allocation of concurrent execution of parallel applications. *Concurrency and Computation: Practice and Experience*, Wiley Online Library, p. e6600, 2021. Citado na página 67.

STEEN, M. V.; TANENBAUM, A. S. *Distributed systems*. [S.l.]: Maarten van Steen Leiden, The Netherlands, 2017. Citado na página 33.

STRATTON, J. et al. Parboil: A revised benchmark suite for scientific and commercial throughput computing. *Center for Reliable and High-Performance Computing*, 2012. Citado na página 46.

SULEMAN, M. A.; QURESHI, M. K.; PATT, Y. N. Feedback-driven threading: Power-efficient and high-performance execution of multi-threaded workloads on cmps. *SIGARCH Comput. Archit. News*, ACM, NY, USA, v. 36, n. 1, p. 277–286, 2008. ISSN 0163-5964. Citado 5 vezes nas páginas 23, 29, 39, 43 e 50.

TAFFONI, G. et al. Towards exascale: Measuring the energy footprint of astrophysics hpc simulations. In: IEEE. *2019 15th International Conference on eScience (eScience)*. [S.l.], 2019. p. 403–412. Citado na página 29.

TAKOUNA, I.; DAWOUD, W.; MEINEL, C. Energy efficient scheduling of hpc-jobs on virtualize clusters using host and vm dynamic configuration. *ACM SIGOPS Operating Systems Review*, ACM New York, NY, USA, v. 46, n. 2, p. 19–27, 2012. Citado na página 40.

THURGOOD, B.; LENNON, R. G. Cloud computing with kubernetes cluster elastic scaling. In: *ICFNDS*. NY, USA: ACM, 2019. ISBN 9781450371636. Citado na página 31.

TUNCER, O. et al. Diagnosing performance variations in hpc applications using machine learning. In: SPRINGER. *High Performance Computing: 32nd International Conference, ISC High Performance 2017, Frankfurt, Germany, June 18–22, 2017, Proceedings 32*. [S.l.], 2017. p. 355–373. Citado 3 vezes nas páginas 24, 41 e 43.

VOGADO, L. H. et al. Rede neural convolucional para o diagnóstico de leucemia. In: SBC. *Anais do XIX Simpósio Brasileiro de Computação Aplicada à Saúde*. [S.l.], 2019. p. 46–57. Citado na página 37.

VYDYANATHAN, N. et al. An integrated approach for processor allocation and scheduling of mixed-parallel applications. In: IEEE. *2006 International Conference on Parallel Processing (ICPP'06)*. [S.l.], 2006. p. 443–450. Citado 2 vezes nas páginas 39 e 43.

WANG, W.; DAVIDSON, J. W.; SOFFA, M. L. Predicting the memory bandwidth and optimal core allocations for multi-threaded applications on large-scale numa machines. In: IEEE. *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. [S.l.], 2016. p. 419–431. Citado na página 40.

XAVIER, M. G. et al. Performance evaluation of container-based virtualization for high performance computing environments. In: IEEE. *2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*. [S.l.], 2013. p. 233–240. Citado 2 vezes nas páginas 31 e 32.

ZHOU, N. et al. Container orchestration on hpc systems. In: IEEE. *2020 IEEE 13th International Conference on Cloud Computing (CLOUD)*. [S.l.], 2020. p. 34–36. Citado na página 33.

ZHOU, Z.-H. *Machine learning*. [S.l.]: Springer Nature, 2021. Citado na página 35.

ZVARA, Z. et al. Optimizing distributed data stream processing by tracing. *Future Generation Computer Systems*, Elsevier, v. 90, p. 578–591, 2019. Citado na página 32.

APÊNDICE A – ARTEFATOS PRODUZIDOS

A.1 CRIAÇÃO DO AMBIENTE DE EXECUÇÃO DE UMA APLICAÇÃO COM *CONTAINERS DOCKER*

Para criar um ambiente de execução, é necessário ter alguns pré-requisitos, conhecimento das tecnologias *Docker* e *Kubernetes*. O primeiro deles é o sistema operacional Linux instalado em sua máquina. Além disso, é necessário instalar o Docker, uma plataforma que permite a criação e o gerenciamento de *containers*, e o *Kubernetes*[1], uma ferramenta de orquestração de *containers* que permite a criação e gerenciamento de *clusters* de *containers*. Com esses pré-requisitos instalados, é possível criar o ambiente de execução. **O processo de execução é composto por quatro etapas:**

- Criação da imagem *Docker*.
- *pull* da imagem no repositório *Docker* Hub[2]
- Criação do arquivo descritor *kubernetes*.
- Despacho para a execução.

A.2 CRIAÇÃO DE UMA IMAGEM *DOCKER*

O encapsulamento de uma aplicação paralela ocorre por meio da criação de uma imagem *Docker*. Essa imagem é um pacote contendo todos os recursos necessários para executar a aplicação em um ambiente isolado. A imagem *Docker* pode ser vista no algoritmo 1 é um exemplo de um *Dockerfile* que contém o sistema operacional, as bibliotecas, as dependências e o código-fonte da aplicação. Após a criação da imagem *Docker*, ela pode ser usada para criar um ou mais *containers*, que são instâncias em execução da imagem. Cada *container* é um ambiente isolado, com seu próprio sistema de arquivos e rede. O encapsulamento em imagem *Docker* é uma técnica importante para garantir a portabilidade, a segurança e a facilidade de gerenciamento de aplicações paralelas.

Algorithm 1 Dockerfile

```
1: FROM gcc : latest                          ▷ Sistema operacional base ja com o gcc
2: COPY ./home      ▷ Copia da aplicação binários e bibliotecas para o /home do container
3: WORKDIR /home                              ▷ Cria o diretório /home
4: CMD [ "./script.sh" ]                      ▷ script que chama a aplicação para a execução
```

[1] https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/
[2] https://hub.docker.com/

A.3 ARQUIVO DESCRITOR

O arquivo *manifest* do *Kubernetes* algoritmo 2, é um arquivo *YAML* que define as especificações de um recurso no *cluster*, como um *pod*, um serviço ou um *deployment*. Para a criação de um *pod*, o arquivo *manifest* deve incluir informações como o nome do *pod*, a imagem *docker* que será usada, os recursos de hardware alocados para o *pod*, entre outras informações relevantes. Além disso, é possível especificar configurações de rede e de armazenamento para o *pod*. O arquivo *manifest* também permite a definição de regras de escalonamento e tolerância a falhas, permitindo que o *Kubernetes* gerencie automaticamente o ciclo de vida do *pod*. O comando para realizar o despacho do arquivo descrito para gerar um pod: ***kubectl apply -f arquivo.yaml***.

Algorithm 2 arquivo.yaml

```
1: apiVersion : v1                                  ▷ versão da api kubernetes
2: kind : Pod                                       ▷ tipo do objeto a ser criado kubernetes
3: metadata :
4:     name : name_pod                              ▷ nome que o pod vai receber
5:     namespace : default                          ▷ nome do namespace que deseja alocar o pod
6: spec :
7:     nodeName : name_nodo_work                    ▷ nome do nodo que vai executar o pod
8:     containers
9:     −name : name_container                       ▷ nome do container dentro do pod
10:      image : name_image                         ▷ nome da imagem no repositório https://hub.docker.com/
11:      command : ["path/./app.sh"]                ▷ caminho dentro da imagem para a aplicação
12:      securityContext :
13:          privileged : true                      ▷ modo privilegiado a acessos aos recurso computacionais
14:     restartPolicy : Never                       ▷ não realizar restar automático do pod
```

A.4 EXECUÇÃO DAS APLICAÇÕES COM DIFERENTES NÚMEROS DE *THREADS* NAS ARQUITETURAS AMD-16, AMD-24 E AMD-64

Nesta Seção, é apresentado os anexos da dissertação, que contêm os resultados detalhados da execução das 24 aplicações em três diferentes arquiteturas: AMD16, AMD24 e AMD64. Cada aplicação foi executada com diferentes números de *threads*, e os resultados foram registrados em tabelas que contêm nove colunas: número de *threads*, IPC (Instruções Por Ciclo), Hit-L2 (acertos na cache L2), Miss-L2 (erros na cache L2), Hit-L3 (acertos na cache L3), Miss-L3 (erros na cache L3), tempo em (segundos), energia em (joules) e EDP (*Energy-Delay Product*). As tabelas fornecem informações valiosas sobre o desempenho de cada aplicação em diferentes cenários de execução, permitindo uma análise mais aprofundada do impacto da configuração de *threads* e dos níveis de cache no desempenho geral do sistema. Isto pode ser visto nas Tabela 6, 7 e 8.

Tabela 6 – Execução das aplicações com diferentes números de *threads* na arquitetura *AMD-16*.

Aplicação FFT-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.83	0.29	0.71	0.86	0.14	88.316	2719.316	240159.1119
4	0.58	0.78	0.22	0.92	0.08	83.748	2635.491	220717.1003
6	0.58	0.60	0.40	0.86	0.14	82.29	2528.721	208088.4511
8	0.50	0.65	0.35	0.71	0.29	82.521	2513.398	207408.1164
10	0.48	0.43	0.57	0.88	0.12	83.325	2546.707	212204.3608
12	0.43	0.21	0.79	0.82	0.18	83.871	2580.454	216425.2574
14	0.41	0.52	0.48	0.58	0.42	84.142	2604.765	219170.1366
16	0.40	0.76	0.24	0.88	0.12	85.517	2657.943	227299.3115
Aplicação NB-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.39	0.48	0.52	0.70	0.30	37.42	1190.433	44550.76459
4	1.30	0.63	0.37	0.73	0.27	24.98	1005.467	25116.56566
6	1.26	0.51	0.49	0.59	0.41	36.29	1594.818	57872.75558
8	1.16	0.45	0.55	0.94	0.06	41.59	1797.451	74763.17689
10	1.12	0.56	0.44	0.91	0.09	47.18	2197.426	103674.5587
12	0.90	0.34	0.66	0.94	0.06	55.04	2468.781	135881.7062
14	0.76	0.79	0.21	0.89	0.11	70.30	3081.308	216606.7085
16	0.64	0.25	0.75	0.76	0.24	94.16	4112.079	387205.6949
Aplicação HPCG-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.37	0.36	0.64	0.65	0.35	28.19	1073.623	30262.2115
4	0.83	0.58	0.42	0.73	0.27	27.28	1096.424	29907.15745
6	0.73	0.39	0.61	0.64	0.12	26.00	1015.856	26414.28771
8	0.53	0.50	0.50	0.79	0.21	25.92	1001.709	25965.29899
10	0.50	0.40	0.60	0.80	0.20	26.38	1002.998	26454.07225
12	0.43	0.64	0.36	0.80	0.20	26.84	1032.044	27700.06096
14	0.38	0.36	0.64	0.83	0.17	26.55	1052.061	27930.11543
16	0.30	0.49	0.51	0.02	0.98	26.78	1062.473	28455.15189
Aplicação PO-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	2.00	0.46	0.54	0.70	0.30	168.41	6542.504	1101823.099
4	1.37	0.45	0.55	0.91	0.09	134.831	5906.299	796352.2005
6	2.99	0.45	0.55	0.91	0.09	41.234	2600.725	107238.2947
8	3.04	0.73	0.27	0.82	0.18	34.477	2260.344	77929.88009
10	2.58	0.64	0.36	0.81	0.19	37.029	2436.806	90232.48937
12	2.15	0.73	0.27	0.83	0.17	34.123	2264.803	77281.87277
14	2.15	0.40	0.60	0.85	0.15	28.205	2017.037	56890.52859
16	2.09	0.55	0.45	0.76	0.24	25.877	1871.91	48439.41507
Aplicação BT-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	2.13	0.30	0.70	0.85	0.15	77.61216	2932.613998	227606.5069
4	2.13	0.20	0.80	0.85	0.15	47.66524	2377.699509	113333.6177
6	2.06	0.44	0.56	0.73	0.27	37.6377	1957.338531	73669.72045
8	1.85	0.44	0.56	0.91	0.09	33.41182	1727.627502	57723.17914
10	1.55	0.34	0.66	0.84	0.16	40.4074	1943.303635	78523.84729
12	1.29	0.62	0.38	0.87	0.13	36.99216	1931.307999	71443.2545
14	1.11	0.67	0.33	0.89	0.11	34.5146	1942.666168	67050.34573
16	1.00	0.70	0.30	0.90	0.10	33.01792	1858.834045	61374.8338
Aplicação EP-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.42	0.15	0.85	0.79	0.21	43.39536	1185.528839	51446.45076
4	0.47	0.39	0.61	0.87	0.13	22.72253	760.2179871	17274.07602
6	0.45	0.54	0.46	0.86	0.14	16.12355	518.6235046	8362.052008
8	0.49	0.38	0.62	0.88	0.12	12.6659	450.6258545	5707.58201
10	0.49	0.54	0.46	0.83	0.17	10.17354	366.7866363	3731.518516
12	0.47	0.64	0.36	0.75	0.25	8.50906	345.8696289	2943.025425
14	0.49	0.68	0.32	0.76	0.24	7.30764	305.5804901	2233.072213
16	0.49	0.57	0.43	0.67	0.33	6.47515	277.2163086	1795.017181

Aplicação JA-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.72	0.50	0.50	0.85	0.15	277.073	9845.122	2727817.488
4	0.44	0.41	0.59	0.84	0.16	276.136	10308.777	2846624.446
6	0.38	0.21	0.79	0.78	0.22	268.548	9617.948	2582880.7
8	0.27	0.41	0.59	0.85	0.15	271.474	9916.802	2692153.906
10	0.27	0.47	0.53	0.72	0.28	274.201	10083.561	2764922.51
12	0.22	0.51	0.49	0.73	0.27	279.59	10061.339	2813049.771
14	0.20	0.51	0.49	0.73	0.27	279.283	10331.914	2885527.938
16	0.15	0.34	0.66	0.62	0.38	284.898	10962.853	3123294.894
Aplicação ST-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.36	0.53	0.47	0.74	0.26	173.739	5954.301	1034494.301
4	0.20	0.28	0.72	0.85	0.15	178.84853	6535.867706	1168930.332
6	0.23	0.53	0.47	0.73	0.27	174.181	6296.227	1096683.115
8	0.14	0.47	0.53	0.81	0.19	175.654	6019.653	1057376.128
10	0.18	0.70	0.30	0.84	0.16	177.597	6396.198	1135945.576
12	0.14	0.47	0.53	0.81	0.19	181.032	6332.929	1146462.803
14	0.13	0.49	0.51	0.72	0.28	180.922	6380.454	1154364.499
16	0.09	0.54	0.46	0.80	0.20	183.548	6367.872	1168810.17
Aplicação CG-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.02	0.59	0.41	0.89	0.11	21.09408	815.3904877	17199.91218
4	0.82	0.65	0.35	0.86	0.14	13.21657	654.5345764	8650.702047
6	0.73	0.65	0.35	0.90	0.10	11.44295	543.897171	6223.788133
8	0.54	0.50	0.50	0.87	0.13	10.99203	503.6460113	5536.092066
10	0.56	0.59	0.41	0.94	0.06	11.45171	557.1226502	6380.007024
12	0.52	0.73	0.27	0.88	0.12	11.08801	575.6073303	6382.339835
14	0.50	0.58	0.42	0.86	0.14	11.0204	566.4029083	6241.986611
16	0.42	0.67	0.33	0.74	0.26	11.16242	563.4384918	6289.33709
Aplicação FT-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	2.01	0.49	0.60	0.85	0.15	12.22289	500.1402283	6113.158995
4	1.77	0.05	0.95	0.88	0.12	7.81693	386.1312714	3018.361119
6	1.66	0.73	0.27	0.79	0.21	6.49505	321.3464661	2087.161364
8	1.29	0.44	0.56	0.66	0.34	6.09077	295.6169128	1800.534624
10	1.23	0.54	0.46	0.69	0.31	6.21652	309.3085175	1922.822585
12	0.99	0.43	0.57	0.86	0.14	5.92201	310.3704834	1838.017106
14	0.92	0.62	0.38	0.71	0.29	5.56134	307.8078155	1711.823917
16	0.81	0.16	0.84	0.77	0.23	5.53979	304.0655518	1684.459303
Aplicação LU-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.34	0.66	0.34	0.74	0.26	54.97266	2083.601364	114541.1094
4	1.33	0.81	1.01	0.88	0.12	32.99929	1548.683502	51105.45601
6	1.48	0.81	0.19	0.90	0.10	22.53474	1141.21463	25716.97497
8	1.20	0.36	0.64	0.67	0.33	20.98404	1037.055649	21761.61722
10	1.04	0.46	0.54	0.72	0.28	20.44333	1063.142441	21734.17175
12	0.94	0.31	0.69	0.82	0.18	18.93643	1020.797638	19330.26302
14	0.88	0.53	0.47	0.83	0.17	18.02296	1008.963623	18184.51102
16	0.83	0.47	0.53	0.72	0.28	17.50156	979.3419647	17140.01216
Aplicação MG-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.56	0.34	0.66	0.79	0.21	10.81297	412.8506012	4464.141165
4	0.95	0.73	0.27	0.84	0.16	9.84656	415.0951233	4087.259037
6	0.92	0.46	0.54	0.79	0.21	8.98037	377.6195068	3391.162891
8	0.56	0.41	0.59	0.85	0.15	9.16474	363.0803528	3327.537032
10	0.58	0.61	0.39	0.70	0.30	9.87445	411.1864624	4060.240164
12	0.49	0.62	0.38	0.66	0.34	10.48878	428.2266083	4491.574684
14	0.39	0.36	0.64	0.85	0.15	10.51908	426.9020233	4490.616535
16	0.34	0.34	0.66	0.72	0.28	11.0015	433.6996155	4771.34632

Aplicação SP-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.54	0.34	0.66	0.78	0.22	43.00153	1729.283218	74361.82419
4	1.09	0.35	0.65	0.88	0.12	37.40919	1690.298416	63232.69461
6	1.08	0.50	0.50	0.77	0.23	33.82561	1554.40062	52578.54914
8	0.70	0.46	0.54	0.76	0.24	34.10614	1483.290115	50589.30034
10	0.64	0.28	0.72	0.72	0.28	39.76828	1789.807632	71177.57107
12	0.51	0.74	0.26	0.79	0.21	49.28516	2087.844696	102899.7599
14	0.39	0.50	0.50	0.83	0.17	51.30767	2165.403137	111101.7896
16	0.27	0.42	0.58	1.00	0.00	56.93472	2278.473892	129724.2731
Aplicação UA-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.97	0.36	0.64	0.67	0.33	62.97615	2153.730225	135633.6377
4	0.86	0.52	0.48	0.71	0.29	44.43938	1707.948104	75900.15481
6	0.88	0.47	0.53	0.86	0.14	37.49541	1550.741074	58145.67236
8	0.72	0.51	0.49	0.67	0.33	32.74709	1322.000549	43291.67097
10	0.75	0.80	0.20	0.78	0.22	31.95105	1426.965988	45593.06164
12	0.66	0.55	0.45	0.77	0.23	30.37421	1381.242889	41954.16158
14	0.55	0.46	0.54	0.89	0.11	28.55086	1291.24379	36866.12066
16	0.47	0.64	0.36	0.73	0.27	27.90022	1226.048477	34207.02224
Aplicação HS-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.73	0.45	0.55	0.82	0.18	579.239	18791.23	10884613.27
4	0.70	0.43	0.57	0.84	0.16	345.858	13045.647	4511941.38
6	1.10	0.53	0.47	0.72	0.28	164.151	7596.322	1246943.853
8	1.21	0.31	0.69	0.74	0.26	125.033	6217.181	777352.792
10	1.14	0.79	0.21	0.80	0.20	100.899	5487.787	553712.2205
12	1.10	0.78	0.22	0.73	0.27	93.46	5112.026	477769.95
14	1.22	0.49	0.51	0.75	0.25	72.877	4360.12	317752.4652
16	1.01	0.31	0.69	0.83	0.17	80.02	4758.28	380757.5656
Aplicação SC-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.08	0.60	0.40	0.86	0.14	254.916	8484.156	2162747.111
4	1.15	0.51	0.49	0.67	0.33	145.505	5956.342	866677.5427
6	1.14	0.40	0.60	0.85	0.15	111.65	4827.362	538974.9673
8	1.13	0.65	0.35	0.75	0.25	102.366	4359.167	446230.4891
10	0.98	0.53	0.47	0.86	0.14	100.004	4302.711	430288.3108
12	0.84	0.53	0.47	0.86	0.14	99.382	4298.871	427230.3977
14	0.74	0.31	0.69	0.84	0.16	99.836	4308.473	430140.7104
16	0.63	0.62	0.38	0.73	0.27	102.004	4357.751	444508.033
Aplicação LUD-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.80	0.37	0.63	0.72	0.28	99.889	3782.364	377816.5576
4	1.98	0.49	0.51	0.89	0.11	58.943	2886.22	170122.4655
6	1.65	0.51	0.49	0.87	0.13	48.031	2309.921	110947.8156
8	1.61	0.83	0.17	0.83	0.17	43.22	2109.47	91171.2934
10	0.60	0.41	0.59	0.76	0.24	39.148	2035.543	79687.43736
12	1.33	0.36	0.64	0.87	0.13	35.049	1920.412	67308.52019
14	1.26	0.56	0.44	0.88	0.12	32.679	1828.308	59747.27713
16	1.11	0.52	0.48	0.79	0.21	33.035	1851.805	61174.37818
Aplicação BFS-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.62	0.46	0.54	0.85	0.15	5.826	245.776	1431.890976
4	0.91	0.47	0.53	0.75	0.25	2.259	71.831	162.266229
6	0.87	0.26	0.74	0.75	0.25	3.671	134.156	492.486676
8	1.09	0.34	0.66	0.77	0.23	4.1	148.687	609.6167
10	0.99	0.34	0.66	0.76	0.24	4.868	190.589	927.787252
12	0.92	0.44	0.56	0.79	0.21	5.098	205.131	1045.757838
14	0.85	0.47	0.53	0.82	0.18	5.757	245.86	1415.41602
16	0.85	0.55	0.45	0.86	0.14	6.095	258.5	1575.5575

Aplicação MRI-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.51	0.31	0.69	0.85	0.15	9.595	253.692	2434.17474
4	0.25	0.74	0.26	0.72	0.28	17.322	550.554	9536.696388
6	0.22	0.35	0.65	0.65	0.35	26.811	876.551	23501.20886
8	0.17	0.33	0.67	0.72	0.28	36.216	1188.882	43056.55051
10	0.14	0.26	0.74	0.81	0.19	49.553	1595.942	79083.71393
12	0.14	0.65	0.35	0.74	0.26	60.938	2001.859	121989.2837
14	0.13	0.45	0.55	0.68	0.32	81.589	2625.252	214191.6854
16	0.09	0.27	0.73	0.80	0.20	98.2269	3097.876312	304294.7867
Aplicação LBM-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.87	0.44	0.56	0.87	0.13	270.424	11385.828	3079001.151
4	1.78	0.51	0.49	0.75	0.25	159.589	8070.282	1287928.234
6	1.96	0.54	0.46	0.74	0.26	127.644	6742.775	860674.7721
8	1.33	0.38	0.62	0.77	0.23	134.772	6739.999	908363.1452
10	1.53	0.53	0.47	0.91	0.09	156.681	7227.82	1132462.065
12	1.35	0.40	0.60	0.87	0.13	139.169	6914.387	962268.3244
14	1.12	0.52	0.48	0.81	0.19	138.781	7104.488	985967.9491
16	0.88	0.52	0.48	0.75	0.25	142.655	7222.263	1030291.928
Aplicação CUTCP-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.52	0.69	0.31	0.80	0.20	26.35843	902.4840241	23788.06197
4	1.55	0.26	0.74	0.82	0.18	14.48191	607.3667602	8795.830759
6	1.53	0.56	0.44	0.70	0.30	10.33636	445.0505829	4600.203043
8	1.44	0.37	0.63	0.78	0.22	8.43093	391.2770386	3298.829323
10	1.28	0.14	0.86	0.83	0.17	8.10874	397.2685394	3221.347296
12	1.21	0.51	0.49	0.80	0.20	7.22698	362.0821228	2616.76026
14	1.17	0.60	0.40	0.74	0.26	6.31495	339.9113617	2146.523254
16	1.14	0.42	0.58	0.82	0.18	5.82691	318.9631805	1858.569746
Aplicação SGEMM-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.88	0.14	0.86	0.79	0.21	3.85878	129.8704681	501.141565
4	1.26	0.75	0.25	0.81	0.19	2.43049	97.6060791	237.2305992
6	1.45	0.47	0.53	0.77	0.23	2.07851	81.96934509	170.3741035
8	1.35	0.47	0.53	0.71	0.29	1.84482	74.75024414	137.9007454
10	1.07	0.76	0.24	0.79	0.21	2.42281	92.29864502	223.6220801
12	1.04	0.45	0.55	0.84	0.16	2.20705	90.39572144	199.507877
14	0.90	0.38	0.62	0.80	0.20	2.06507	86.25067139	178.113674
16	0.82	0.32	0.68	0.79	0.21	2.0159	85.72276306	172.8085181
Aplicação SPMV-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.24	0.46	0.54	0.84	0.16	1.96211	55.51976013	108.9358765
4	1.13	0.41	0.59	0.82	0.18	1.72541	50.49412537	87.12306885
6	1.56	0.31	0.69	0.73	0.27	1.58225	47.20895386	74.69636724
8	1.36	0.59	0.41	0.84	0.16	1.60217	46.59962463	74.66052059
10	1.14	0.63	0.37	0.80	0.20	1.63205	49.68716431	81.09193651
12	1.04	0.51	0.49	0.84	0.16	1.58919	45.94914246	73.02191771
14	1.02	0.79	0.21	0.76	0.24	1.53157	45.85083008	70.22375583
16	1.07	0.57	0.43	0.84	0.16	1.53504	46.30685425	71.08287355
Aplicação TPACF-AMD16								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.78	0.48	0.52	0.79	0.21	2.57878	86.75950623	223.7336795
4	0.61	0.48	0.52	0.81	0.19	3.81286	129.4683685	493.6447636
6	0.62	0.47	0.53	0.84	0.16	9.4352	321.4776459	3033.205884
8	0.72	0.33	0.67	0.80	0.20	10.09186	373.5765381	3770.082122
10	0.62	0.51	0.49	0.77	0.23	9.5614	374.3434753	3579.247705
12	0.54	0.41	0.59	0.82	0.18	9.03102	376.3561859	3398.880242
14	0.33	0.53	0.47	0.84	0.16	11.46898	423.9050446	4861.758478
16	0.21	0.49	0.51	0.85	0.15	15.58943	513.0625305	7998.352405

Tabela 7 – Execução das aplicações com diferentes números de *threads* na arquitetura *AMD-24*.

Aplicação FFT-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.71	0.59	0.41	0.83	0.17	90.72285	4111.711853	373026.2177
4	0.63	0.40	0.60	0.81	0.19	82.9602	4239.216995	351686.2898
6	0.53	0.37	0.63	0.89	0.11	82.66023	4705.361023	388946.2244
8	0.57	0.45	0.55	0.91	0.09	76.08639	4630.894485	352348.0439
10	0.58	0.04	0.96	0.89	0.11	72.92199	4796.862701	349796.7739
12	0.51	0.80	0.20	0.90	0.10	73.14221	5198.373596	380220.5332
14	0.51	0.49	0.51	0.91	0.09	72.09354	4996.356934	360205.0584
16	0.50	0.58	0.42	0.81	0.19	77.75142	5519.404785	429141.5596
18	0.42	0.40	0.60	0.88	0.12	75.93499	5482.50174	416313.7148
20	0.46	0.32	0.68	0.88	0.12	79.44717	5862.678787	465773.2383
22	0.44	0.38	0.62	0.87	0.13	78.9173	5924.136566	467516.8626
24	0.43	0.56	0.44	0.87	0.13	72.4788	5407.932526	391960.4599
Aplicação NB-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.58	0.73	0.27	0.94	0.06	35.94585	1771.095901	63663.54761
4	1.46	0.24	0.76	0.95	0.05	32.42176	2129.721558	69049.32121
6	1.36	0.62	0.38	0.93	0.07	35.18992	2933.578537	103232.394
8	1.34	0.55	0.45	0.94	0.06	53.76281	5407.87674	290742.6496
10	1.28	0.39	0.61	0.76	0.24	67.03973	7958.420593	533530.3678
12	1.36	0.36	0.64	0.95	0.05	72.18437	9902.365189	714795.9926
14	1.17	0.37	0.63	0.74	0.26	98.78712	14046.09045	1387572.823
16	0.99	0.28	0.72	0.89	0.11	125.37463	17586.39197	2204887.386
18	0.88	0.55	0.45	0.93	0.07	142.39902	20077.78893	2859057.467
20	0.86	0.52	0.48	0.88	0.12	150.76628	21889.24887	3300160.624
22	0.78	0.55	0.45	0.93	0.07	166.35728	24466.07205	4070109.199
24	0.73	0.51	0.49	0.91	0.09	193.10489	28637.78795	5530096.892
Aplicação HPCG-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.00	0.46	0.54	0.95	0.05	31.70915	1792.713959	56845.43582
4	0.85	0.53	0.47	0.74	0.26	27.69813	1854.864243	51376.27092
6	0.56	0.40	0.60	0.93	0.07	26.88379	2099.818512	56451.07991
8	0.71	0.53	0.47	0.95	0.05	22.30784	1958.397003	43687.607
10	0.67	0.57	0.43	0.94	0.06	18.48211	1857.094238	34323.01999
12	0.50	0.19	0.81	0.93	0.07	18.21701	2068.407471	37680.19958
14	0.48	0.17	0.83	0.93	0.07	18.92655	2058.804733	38966.07072
16	0.46	0.59	0.41	0.95	0.05	20.13685	2129.605484	42883.54619
18	0.38	0.26	0.74	0.90	0.10	21.02785	2263.465775	47595.81879
20	0.35	0.35	0.65	0.95	0.05	19.0901	2153.355698	41107.7756
22	0.31	0.42	0.58	0.83	0.17	18.55499	2195.963791	40746.08618
24	0.44	0.28	0.72	0.93	0.07	18.99295	2355.712021	44741.92063
Aplicação PO-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.20	0.50	0.50	0.82	0.18	182.39453	10086.26709	1839679.945
4	2.01	0.57	0.43	0.84	0.16	64.54871	6027.49411	389066.9693
6	3.03	0.39	0.61	0.95	0.05	36.61979	4498.831421	164746.2619
8	3.10	0.42	0.58	0.90	0.10	27.65616	4376.488449	121036.8648
10	3.21	0.59	0.41	0.94	0.06	22.54604	3967.65094	89454.8168
12	2.83	0.32	0.68	0.92	0.08	20.15907	3540.609512	71375.395
14	2.71	0.43	0.57	0.87	0.13	22.84244	4045.618362	92411.79471
16	2.51	0.37	0.63	0.82	0.18	20.8125	3691.646057	76832.38356
18	2.46	0.09	0.91	0.94	0.06	18.97403	3360.392273	63760.1838
20	2.18	0.60	0.40	0.95	0.05	17.54042	3107.221878	54501.97677
22	2.08	0.41	0.59	0.95	0.05	16.07604	2843.954697	45719.52946
24	2.09	0.22	0.78	0.88	0.12	15.42516	2669.765854	41181.56546

Aplicação BT-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	2.12	0.62	0.38	0.88	0.12	74.86137	4539.926544	339865.1208
4	2.21	0.55	0.45	0.91	0.09	42.56631	3658.888931	155745.4005
6	2.02	0.30	0.70	0.92	0.08	32.29274	3403.068283	109894.3993
8	2.07	0.12	0.88	0.92	0.08	25.7583	3232.66449	83267.94173
10	1.99	0.38	0.62	0.89	0.11	21.28079	3085.791199	65668.07448
12	1.80	0.53	0.47	0.94	0.06	19.56703	2873.916	56234.0006
14	1.65	0.46	0.54	0.92	0.08	26.68764	3237.934464	86412.82931
16	1.37	0.50	0.50	0.90	0.10	25.4758	3322.072922	84632.46534
18	1.24	0.50	0.50	0.89	0.11	23.35215	3138.174591	73283.12378
20	1.14	0.05	0.95	0.91	0.09	22.70888	3242.567627	73635.07913
22	1.06	0.23	0.77	0.91	0.09	20.83429	3104.070862	64671.11252
24	1.03	0.48	0.52	0.91	0.09	20.37602	3017.593048	61486.5363
Aplicação EP-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.47	0.53	0.47	0.85	0.15	41.20996	1913.345428	78848.88857
4	0.49	0.58	0.42	0.79	0.21	21.06269	1337.865662	28179.04969
6	0.51	0.13	0.87	0.91	0.09	14.23077	1145.144302	16296.28518
8	0.48	0.43	0.57	0.83	0.17	10.74518	1053.412933	11319.11158
10	0.48	0.09	0.91	0.87	0.13	8.55058	980.0774078	8380.230282
12	0.52	0.40	0.60	0.89	0.11	7.16056	934.1839905	6689.280515
14	0.49	0.75	0.25	0.78	0.22	6.20224	841.5283356	5219.360704
16	0.48	0.25	0.75	0.91	0.09	5.50033	776.8392181	4272.872057
18	0.47	0.61	0.39	0.86	0.14	4.8465	722.9112244	3503.589249
20	0.48	0.29	0.71	0.90	0.10	4.33916	589.3168945	2232.842068
22	0.48	0.69	0.31	0.74	0.26	3.99863	557.3999786	2228.836277
24	0.49	0.58	0.42	0.77	0.23	3.61417	617.8021698	2232.842068
Aplicação JA-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.52	0.55	0.45	0.92	0.08	300.07874	15705.31955	4712832.502
4	0.44	0.34	0.66	0.83	0.17	277.99307	17142.26308	4765430.339
6	0.28	0.45	0.55	0.91	0.09	274.61319	20083.558	5515209.929
8	0.45	0.55	0.45	0.95	0.05	206.93051	16849.57906	3486691.987
10	0.45	0.43	0.57	0.95	0.05	249.67111	22907.97169	5719458.721
12	0.30	0.32	0.68	0.84	0.16	139.44255	16142.76218	2250987.922
14	0.32	0.19	0.81	0.94	0.06	163.168	15911.60612	2596264.948
16	0.31	0.67	0.33	0.94	0.06	279.39908	26026.27809	7271718.155
18	0.23	0.35	0.65	0.94	0.06	196.89748	19104.60953	3761649.472
20	0.22	0.38	0.62	0.94	0.06	177.59714	18088.63556	3212489.942
22	0.35	0.07	0.93	0.92	0.08	235.82416	24881.97281	5867770.337
24	0.56	0.49	0.51	0.95	0.05	148.80101	18443.33264	2744386.525
Aplicação ST-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.29	0.61	0.39	0.95	0.05	187.98711	9189.668182	1727539.163
4	0.28	0.25	0.75	0.96	0.04	182.19154	10424.02985	1899170.051
6	0.14	0.49	0.51	0.94	0.06	179.11188	12457.96463	2231369.466
8	0.29	0.36	0.64	0.93	0.07	135.6855	10400.57831	1411207.668
10	0.28	0.20	0.80	0.90	0.10	148.78513	13,554.30190	2016678.57
12	0.17	0.58	0.42	0.94	0.06	195.52008	18218.71608	3562124.825
14	0.22	0.21	0.79	0.94	0.06	107.14826	10464.22447	1121223.444
16	0.23	0.50	0.50	0.80	0.20	117.90543	10683.84633	1259683.495
18	0.19	0.40	0.60	0.85	0.15	127.17326	11671.45602	1484297.112
20	0.18	0.21	0.79	0.95	0.05	114.89906	11289.26208	1297125.602
22	0.15	0.34	0.66	0.95	0.05	104.89923	11382.74274	1194040.948
24	0.11	0.34	0.66	0.93	0.07	97.24726	11310.3338	1099898.972

Aplicação CG-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.52	0.37	0.63	0.92	0.08	20.52156	1281.277634	26293.81584
4	0.63	0.57	0.43	0.84	0.16	13.03694	1104.37529	14397.67439
6	0.41	0.44	0.56	0.96	0.04	11.37341	1133.638794	12893.3388
8	0.65	0.43	0.57	0.87	0.13	8.80374	1055.332413	9290.872175
10	0.62	0.23	0.77	0.91	0.09	7.09508	1016.407974	7211.49589
12	0.52	0.57	0.43	0.95	0.05	6.08788	1004.484573	6115.181544
14	0.54	0.18	0.82	0.77	0.23	7.12477	1100.23526	7838.923173
16	0.49	0.47	0.53	0.78	0.22	7.52511	1134.322815	8535.903958
18	0.40	0.48	0.52	0.92	0.08	7.88203	1174.61528	9258.352877
20	0.38	0.25	0.75	0.93	0.07	7.04032	1143.414215	8050.001967
22	0.34	0.29	0.71	0.92	0.08	6.44893	1112.616379	7175.185144
24	0.31	0.51	0.49	0.83	0.17	5.98555	1029.417358	6161.62907
Aplicação FT-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.68	0.47	0.53	0.92	0.08	12.5588	762.3713989	9574.469925
4	1.73	0.23	0.77	0.92	0.08	7.66872	641.9283447	4922.768736
6	1.42	0.51	0.49	0.89	0.11	6.04863	619.0876312	3744.632019
8	1.66	0.14	0.86	0.90	0.10	4.69418	561.4751587	2635.66546
10	1.64	0.59	0.41	0.88	0.12	3.99246	520.054184	2076.295527
12	1.49	0.63	0.37	0.84	0.16	3.60865	485.7369537	1752.854658
14	1.47	0.53	0.47	0.93	0.07	4.09391	519.5069428	2126.814668
16	1.07	0.42	0.58	0.93	0.07	4.01388	529.1797028	2124.063825
18	1.08	0.23	0.77	0.90	0.10	4.14598	558.2304077	2314.412106
20	1.01	0.57	0.43	0.92	0.08	3.68652	501.6214905	1849.237657
22	0.95	0.53	0.47	0.90	0.10	3.62025	497.2937164	1800.327577
24	0.90	0.46	0.54	0.89	0.11	3.34168	463.9548492	1550.388641
Aplicação LU-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.15	0.47	0.53	0.71	0.29	58.72623	3211.817795	188617.9505
4	1.62	0.42	0.58	0.85	0.15	26.96771	2229.570023	60126.39779
6	1.38	0.50	0.50	0.93	0.07	20.3576	2101.038544	42772.10226
8	1.46	0.16	0.84	0.94	0.06	15.66782	1981.351715	31043.46203
10	1.34	0.32	0.68	0.94	0.06	13.09472	1932.911774	25310.93846
12	1.18	0.43	0.57	0.90	0.10	12.33207	1930.112808	23802.28626
14	1.05	0.50	0.50	0.84	0.16	13.45907	2080.631317	28003.36254
16	0.94	0.32	0.68	0.78	0.22	13.24287	2004.643433	26547.23237
18	0.71	0.71	0.29	0.94	0.06	12.70162	1906.640518	24217.42334
20	0.65	0.20	0.80	0.90	0.10	11.75847	1784.338013	20981.08499
22	0.60	0.49	0.51	0.85	0.15	11.25278	1745.519485	19641.94676
24	0.55	0.27	0.73	0.85	0.15	11.3754	1750.185257	19909.05737
Aplicação MG-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.16	0.52	0.48	0.94	0.06	11.57646	659.6838379	7636.803562
4	1.00	0.56	0.44	0.77	0.23	8.97222	639.7378235	5739.868495
6	0.66	0.39	0.61	0.77	0.23	8.6661	713.1418152	6180.158285
8	0.99	0.65	0.35	0.85	0.15	6.57289	622.6864166	4092.849321
10	0.96	0.44	0.56	0.82	0.18	5.39022	590.859314	3184.861691
12	0.70	0.41	0.59	0.77	0.23	4.48362	598.7833404	2684.716961
14	0.75	0.55	0.45	0.90	0.10	5.91462	623.4820099	3687.659165
16	0.68	0.40	0.60	0.85	0.15	6.69192	671.8417816	4495.911455
18	0.54	0.68	0.32	0.87	0.13	7.67976	780.3611908	5992.986659
20	0.49	0.56	0.44	0.89	0.11	6.901	755.1002045	5210.946511
22	0.41	0.78	0.22	0.77	0.23	6.32964	756.5370331	4788.607066
24	0.33	0.58	0.42	0.89	0.11	5.9826	790.809021	4731.094049

Aplicação UA-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.95	0.70	0.30	0.93	0.07	62.13268	3016.980728	187453.0981
4	0.90	0.29	0.71	0.77	0.23	40.28617	2592.98732	104461.528
6	0.69	0.55	0.45	0.83	0.17	33.26629	2588.619644	86113.77178
8	0.96	0.44	0.56	0.91	0.09	24.57939	2345.748596	57657.06959
10	1.02	0.38	0.62	0.88	0.12	18.50643	2168.616364	40133.34693
12	0.93	0.36	0.64	0.88	0.12	15.85308	2053.383102	32552.44659
14	1.01	0.34	0.66	0.93	0.07	18.10669	2461.500885	44569.63346
16	0.92	0.51	0.49	0.90	0.10	18.38469	2491.362442	45802.92617
18	0.79	0.63	0.37	0.95	0.05	18.53858	2456.597733	45541.83359
20	0.73	0.51	0.49	0.81	0.19	16.04745	2300.827393	36922.41254
22	0.66	0.52	0.48	0.93	0.07	14.08511	2122.684189	29898.2403
24	0.60	0.45	0.55	0.78	0.22	13.25822	2008.700104	26631.78789
Aplicação HS-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.68	0.72	0.28	0.81	0.19	625.49988	29783.89362	18629821.88
4	1.02	0.41	0.59	0.84	0.16	247.85519	18087.2404	4483016.406
6	1.12	0.44	0.56	0.90	0.10	131.07712	13146.47906	1723202.614
8	1.15	0.41	0.59	0.74	0.26	104.58408	12780.38875	1336625.199
10	1.19	0.34	0.66	0.89	0.11	88.26805	12732.19556	1123846.074
12	1.14	0.44	0.56	0.77	0.23	77.56865	12875.19379	998711.4005
14	1.12	0.73	0.27	0.75	0.25	71.12768	12285.27084	873822.8133
16	1.05	0.44	0.56	0.94	0.06	67.93745	11550.92075	784740.1006
18	1.14	0.48	0.52	0.91	0.09	60.13392	9590.145355	576693.0336
20	1.11	0.38	0.62	0.92	0.08	55.2612	9139.133484	505039.4833
22	1.12	0.47	0.53	0.76	0.24	50.6728	8591.003967	435330.2258
24	0.96	0.40	0.60	0.91	0.09	50.02054	8598.920609	430122.6523
Aplicação SP-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.35	0.45	0.55	0.88	0.12	47.50801	2756.813675	130970.7316
4	1.17	0.43	0.57	0.91	0.09	34.94442	2601.162048	90896.09911
6	0.80	0.51	0.49	0.81	0.19	33.14436	2878.707611	95412.9214
8	1.10	0.65	0.35	0.94	0.06	26.28245	2649.851974	69644.60203
10	1.06	0.50	0.50	0.95	0.05	21.32851	2484.79512	52996.97757
12	0.84	0.32	0.68	0.91	0.09	18.35911	2509.535049	46072.83002
14	0.88	0.60	0.40	0.95	0.05	24.24613	2725.415848	66080.78695
16	0.79	0.46	0.54	0.84	0.16	29.60911	3033.718796	89825.71353
18	0.56	0.67	0.33	0.79	0.21	35.10139	3570.325958	125323.4039
20	0.53	0.58	0.42	0.84	0.16	33.08521	3562.892914	117879.0603
22	0.43	0.18	0.82	0.83	0.17	29.37959	3507.659439	103053.5962
24	0.35	0.47	0.53	0.89	0.11	24.24613	2725.415848	66080.78695
Aplicação SC-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.54	0.50	0.50	0.95	0.05	248.03586	12565.12082	3116600.548
4	0.82	0.41	0.59	0.87	0.13	136.18149	9436.293503	1285048.509
6	0.88	0.68	0.32	0.94	0.06	99.83293	8454.185196	844006.0789
8	0.94	0.47	0.53	0.91	0.09	80.5335	8005.942001	644746.5302
10	1.00	0.36	0.64	0.93	0.07	70.16489	7812.249542	548145.6298
12	0.99	0.54	0.46	0.95	0.05	63.88191	7666.845688	489772.7462
14	0.97	0.44	0.56	0.72	0.28	61.8998	7521.071777	465552.8388
16	0.89	0.54	0.46	0.93	0.07	64.27124	7551.179123	485323.6457
18	0.83	0.40	0.60	0.91	0.09	66.86698	7865.278046	525927.3898
20	0.73	0.54	0.46	0.93	0.07	61.18842	7520.433929	460163.4699
22	0.69	0.42	0.58	0.95	0.05	59.37855	7564.025879	449140.8889
24	0.63	0.61	0.39	0.93	0.07	57.40179	7693.712097	441632.8461

Aplicação LUD-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.72	0.60	0.40	0.72	0.28	98.18794	5621.455719	551959.1568
4	1.74	0.49	0.51	0.93	0.07	59.15953	4639.595169	274476.2696
6	1.48	0.51	0.49	0.95	0.05	48.40018	4547.353439	220092.725
8	1.70	0.60	0.40	0.83	0.17	35.89128	4097.528763	147065.5521
10	1.14	0.44	0.56	0.92	0.08	32.05934	4097.01889	131347.7216
12	0.85	0.20	0.80	0.95	0.05	31.28492	4252.80191	133048.5675
14	0.74	0.44	0.56	0.94	0.06	28.25522	4150.5271	117274.0563
16	0.69	0.48	0.52	0.75	0.25	26.31193	3936.962418	103589.0795
18	0.64	0.25	0.75	0.94	0.06	24.69841	3686.049591	91039.56408
20	0.60	0.54	0.46	0.74	0.26	24.23557	3811.947617	92384.7233
22	0.57	0.41	0.59	0.95	0.05	23.68193	3787.706131	89700.19145
24	0.55	0.31	0.69	0.96	0.04	25.77585	3982.680603	102656.9778
Aplicação BFS-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.27	0.30	0.70	0.95	0.05	2.07628	87.04502869	180.7298522
4	1.16	0.46	0.54	0.89	0.11	3.98335	229.5466309	914.364572
6	1.10	0.36	0.64	0.93	0.07	4.02329	288.6898956	1161.48317
8	1.09	0.24	0.76	0.94	0.06	5.35898	487.5167389	2612.592453
10	1.17	0.37	0.63	0.95	0.05	6.45847	704.8749542	4552.413746
12	1.20	0.50	0.50	0.94	0.06	6.92319	876.3334198	6067.022769
14	1.21	0.46	0.54	0.94	0.06	9.33469	1310.335114	12231.57208
16	1.10	0.48	0.52	0.93	0.07	9.79071	1394.304626	13651.23225
18	1.02	0.32	0.68	0.83	0.17	10.55021	1530.414566	16146.19506
20	0.91	0.29	0.71	0.90	0.10	11.82315	1745.220474	20634.00345
22	0.90	0.33	0.67	0.95	0.05	12.31191	1837.848175	22627.42132
24	0.89	0.81	0.19	0.96	0.04	13.58216	2062.737839	28016.43536
Aplicação MRI-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.44	0.56	0.44	0.77	0.23	9.20511	9.20511	84.73405011
4	0.27	0.34	0.66	0.91	0.09	16.74074	16.74074	280.2523757
6	0.23	0.24	0.76	0.93	0.07	25.55168	25.55168	652.8883508
8	0.28	0.62	0.38	0.94	0.06	43.86671	43.86671	1924.288246
10	0.26	0.79	0.21	0.91	0.09	54.44503	54.44503	2964.261292
12	0.30	0.45	0.55	0.92	0.08	70.02682	70.02682	4903.755519
14	0.26	0.39	0.61	0.95	0.05	99.23514	99.23514	9847.613011
16	0.22	0.47	0.53	0.93	0.07	98.28333	98.28333	9659.612956
18	0.20	0.29	0.71	0.94	0.06	103.58268	103.58268	10729.3716
20	0.18	0.40	0.60	0.93	0.07	113.1152	113.1152	12795.04847
22	0.16	0.55	0.45	0.96	0.04	119.42033	119.42033	14261.21522
24	0.14	0.55	0.45	0.91	0.09	123.41433	123.41433	15231.09685
Aplicação LBM-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	2.11	0.39	0.61	0.94	0.06	264.577	15587.03088	4123962.7
4	2.60	0.71	0.29	0.84	0.16	139.895	12132.58461	1697282.222
6	1.83	0.33	0.67	0.94	0.06	139.245	12942.10072	1802121.78
8	2.08	0.38	0.62	0.94	0.06	100.872	11904.31075	1200816.871
10	2.25	0.40	0.60	0.93	0.07	80.811	11212.06723	906055.4498
12	2.01	0.79	0.21	0.84	0.16	70.208	10731.42343	753436.7127
14	1.97	0.33	0.67	0.94	0.06	91.641	11639.85773	1066691.461
16	1.75	0.53	0.47	0.87	0.13	90.772	11721.98273	1064027.464
18	1.36	0.43	0.57	0.94	0.06	99.349	12804.91327	1272159.938
20	1.27	0.39	0.61	0.92	0.08	90.468	11828.73846	1070125.269
22	1.13	0.34	0.66	0.90	0.10	83.514	11769.74542	982941.8147
24	0.81	0.42	0.58	0.94	0.06	76.807	11393.12794	875071.4084

Aplicação CUTCP-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.26	0.68	0.32	0.92	0.08	25.51647	1297.751373	33114.03398
4	1.31	0.51	0.49	0.86	0.14	25.53931	1294.487671	33060.32192
6	1.31	0.23	0.77	0.92	0.08	9.18795	856.3790894	7868.368254
8	1.34	0.34	0.66	0.81	0.19	7.39652	816.0586243	6035.993936
10	1.39	0.62	0.38	0.90	0.10	5.79753	776.6668243	4502.749214
12	1.25	0.47	0.53	0.92	0.08	5.61987	785.5628357	4414.761013
14	1.28	0.53	0.47	0.90	0.10	4.87552	733.2759399	3575.101511
16	1.19	0.40	0.60	0.92	0.08	4.91264	738.091156	3625.976137
18	1.11	0.28	0.72	0.92	0.08	4.42737	681.5348968	3017.407156
20	1.00	0.44	0.56	0.93	0.07	4.00667	639.2357483	2561.206696
22	1.02	0.55	0.45	0.92	0.08	3.99173	630.4155579	2516.448695
24	1.04	0.44	0.56	0.92	0.08	3.9511	609.0565338	2406.443271
Aplicação SGEMM-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.15	0.55	0.45	0.92	0.08	4.09795	234.9405518	962.7746341
4	1.10	0.71	0.29	0.90	0.10	2.55458	186.8576355	477.3427785
6	1.15	0.59	0.41	0.84	0.16	2.0658	172.3741608	356.0905413
8	1.12	0.35	0.65	0.92	0.08	1.8392	166.1278992	305.5424322
10	1.23	0.62	0.38	0.93	0.07	1.70489	151.6643372	258.5710118
12	1.09	0.51	0.49	0.90	0.10	1.56113	132.9300995	207.5211662
14	1.01	0.40	0.60	0.88	0.12	1.92991	151.4596252	292.3034453
16	0.85	0.11	0.89	0.78	0.22	1.95021	152.2575531	296.9342026
18	0.83	0.45	0.55	0.87	0.13	1.77998	153.505661	273.2370065
20	0.71	0.53	0.47	0.90	0.10	1.69393	149.4284515	253.1213369
22	0.71	0.39	0.61	0.74	0.26	1.67789	144.0671387	241.7288113
24	0.74	0.07	0.93	0.94	0.06	1.58026	137.0581055	216.5874418
Aplicação SPMV-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.72	0.57	0.43	0.90	0.10	1.92281	84.01545715	161.5457612
4	1.29	0.54	0.46	0.92	0.08	1.82682	80.99546814	147.9641411
6	1.28	0.31	0.69	0.78	0.22	1.72487	78.53044128	135.4548023
8	1.20	0.44	0.56	0.82	0.18	1.5594	73.14175415	114.0572514
10	1.29	0.41	0.59	0.92	0.08	1.50413	72.99238586	109.7900373
12	0.95	0.53	0.47	0.77	0.23	1.49285	72.80023193	108.6798262
14	1.21	0.34	0.66	0.94	0.06	1.54305	75.03718567	115.7861293
16	1.09	0.49	0.51	0.89	0.11	1.49149	72.85266113	108.6590155
18	1.01	0.53	0.47	0.92	0.08	1.4904	73.27693176	109.2119391
20	0.98	0.27	0.73	0.88	0.12	1.46631	72.46925354	106.2623912
22	0.86	0.53	0.47	0.88	0.12	1.47785	72.82121277	107.6188293
24	0.94	0.60	0.40	0.92	0.08	1.44075	70.61361694	101.7365686
Aplicação TPACF-AMD24								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.61	0.53	0.47	0.88	0.12	2.46386	111.1647492	273.8943788
4	0.54	0.61	0.39	0.81	0.19	4.9576	277.5486908	1375.97539
6	0.69	0.50	0.50	0.90	0.10	8.65784	585.9344635	5072.926835
8	0.80	0.39	0.61	0.94	0.06	12.65665	1079.775742	13666.34364
10	0.80	0.36	0.64	0.91	0.09	11.97552	1204.866302	14428.9005
12	0.87	0.50	0.50	0.92	0.08	12.63705	1462.475479	18481.37575
14	0.80	0.77	0.23	0.80	0.20	13.07728	1572.132446	20559.2162
16	0.75	0.36	0.64	0.91	0.09	16.53514	1974.583435	32650.01354
18	0.69	0.60	0.40	0.92	0.08	18.95845	2210.388779	41905.54514
20	0.58	0.37	0.63	0.86	0.14	17.37841	2094.225067	36394.30185
22	0.46	0.41	0.59	0.89	0.11	17.00306	2153.058548	36608.58367
24	0.40	0.61	0.39	0.85	0.15	18.52367	2307.243805	42738.62285

Tabela 8 – Execução das aplicações com diferentes números de *threads* na arquitetura *AMD-64*.

Aplicação FFT-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.90	0.40	0.60	0.25	0.75	126.786	8530.339	1081527.56
4	0.53	0.36	0.64	0.39	0.61	126.25	8546.136	1078949.67
6	0.44	0.26	0.74	0.30	0.70	126.652	8733.167	1106073.067
8	0.33	0.50	0.50	0.45	0.55	127.55	8939.917	1140286.413
10	0.31	0.41	0.59	0.21	0.79	128.018	9054.812	1159178.923
12	0.33	0.53	0.47	0.47	0.53	128.695	9214.042	1185801.135
14	0.19	0.49	0.51	0.52	0.48	129.693	9370.518	1215290.591
16	0.24	0.16	0.84	0.67	0.33	130.024	9530.985	1239256.794
18	0.23	0.42	0.58	0.30	0.70	130.279	9569.111	1246654.212
20	0.24	0.48	0.52	0.59	0.41	130.616	9671.243	1263219.076
22	0.24	0.33	0.67	0.35	0.65	130.944	9798.166	1283011.049
24	0.22	0.40	0.60	0.52	0.48	131.146	9906.443	1299190.374
26	0.20	0.44	0.56	0.54	0.46	131.355	9993.96	1312756.616
28	0.20	0.41	0.59	0.32	0.68	131.494	10054.015	1322042.648
30	0.21	0.42	0.58	0.26	0.74	131.626	10193.731	1341760.037
32	0.19	0.32	0.68	0.39	0.61	131.824	10337.61	1362745.101
34	0.19	0.44	0.56	0.47	0.53	131.954	10403.811	1372824.477
36	0.18	0.36	0.64	0.29	0.71	132.08	10478.712	1384028.281
38	0.20	0.46	0.54	0.79	0.21	132.259	10514.166	1390593.081
40	0.19	0.55	0.45	0.76	0.24	132.414	10664.217	1412091.63
42	0.19	0.53	0.47	0.85	0.15	132.752	10770.57	1429814.709
44	0.19	0.35	0.65	0.49	0.51	132.918	10867.537	1444491.283
46	0.19	0.40	0.60	0.34	0.66	133.028	10968.258	1459085.425
48	0.20	0.27	0.73	0.45	0.55	133.297	11065.628	1475015.016
50	0.19	0.34	0.66	0.65	0.35	133.556	11183.573	1493633.276
52	0.19	0.36	0.64	0.20	0.80	133.574	11250.24	1502739.558
54	0.18	0.34	0.66	0.39	0.61	133.909	11378.898	1523736.852
56	0.19	0.43	0.57	0.65	0.35	134.226	11460.664	1538319.086
58	0.19	0.44	0.56	0.82	0.18	134.509	11625.107	1563681.517
60	0.20	0.39	0.61	0.61	0.39	134.954	11775.842	1589196.981
62	0.20	0.39	0.61	0.40	0.60	136.141	11937.343	1625161.813
64	0.20	0.44	0.56	0.30	0.70	136.12	12044.939	1639557.097
Aplicação NB-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	2.41	0.22	0.78	0.46	0.54	30.00423	2634.429276	79044.0219
4	1.99	0.32	0.68	0.51	0.49	19.01344	1835.78038	34904.50011
6	0.96	0.39	0.61	0.48	0.52	28.39677	2755.312622	78241.97881
8	0.55	0.39	0.61	0.40	0.60	31.61031	3165.047363	100048.1283
10	0.49	0.48	0.52	0.35	0.65	39.73557	4098.896454	162871.987
12	0.38	0.33	0.67	0.41	0.59	45.65706	4833.498932	220683.3507
14	0.31	0.45	0.55	0.53	0.47	56.63024	6176.64386	349784.8242
16	0.27	0.46	0.54	0.50	0.50	63.00729	7109.202438	447931.5797
18	0.23	0.42	0.58	0.51	0.49	74.78778	8677.996811	649008.1163
20	0.20	0.43	0.57	0.29	0.71	82.04938	9819.518097	805685.3718
22	0.17	0.44	0.56	0.24	0.76	88.50082	10941.51971	968333.466
24	0.16	0.45	0.55	0.39	0.61	96.12775	12222.32449	1174904.553
26	0.14	0.37	0.63	0.56	0.44	102.51505	13432.88255	1377072.627
28	0.14	0.33	0.67	0.39	0.61	110.19016	14906.7484	1642576.991
30	0.13	0.44	0.56	0.28	0.72	118.00873	16421.53355	1937884.319
32	0.13	0.47	0.53	0.41	0.59	124.1927	17871.61024	2219523.53
34	0.12	0.42	0.58	0.35	0.65	142.06998	20842.91579	2961152.629
36	0.12	0.34	0.66	0.45	0.55	143.27843	21456.61951	3074270.756
38	0.12	0.53	0.47	0.42	0.58	182.97552	28112.24866	5143853.316
40	0.11	0.51	0.49	0.73	0.27	205.44383	32316.40935	6639206.908
42	0.11	0.34	0.66	0.45	0.55	233.99454	37593.02803	8796563.301
44	0.11	0.36	0.64	0.27	0.73	254.81821	41744.24638	10637194.14
46	0.11	0.40	0.60	0.55	0.45	262.5408	44240.40126	11614910.34
48	0.11	0.41	0.59	0.50	0.50	315.66531	29190.83195	9214533.018
50	0.11	0.32	0.68	0.35	0.65	329.83069	30943.01421	10205955.73
52	0.11	0.49	0.51	0.40	0.60	338.71362	32230.60063	10916943.41
54	0.10	0.31	0.69	0.40	0.60	359.61783	34873.04962	12540970.43
56	0.10	0.42	0.58	0.35	0.65	354.30884	1530.706985	542343.0164
58	0.10	0.51	0.49	0.31	0.69	340.47714	286.013504	97381.05985
60	0.10	0.39	0.61	0.49	0.51	352.62949	4132.325485	1457179.828
62	0.10	0.52	0.48	0.43	0.57	373.00476	9300.463593	3469117.19
64	0.10	0.28	0.72	0.30	0.70	366.70737	8284.847855	3038114.768

Aplicação HPCG-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.51	0.36	0.64	0.61	0.39	42.943	3814.131	163790.2275
4	0.75	0.50	0.50	0.80	0.20	43.592	4068.161	177339.2743
6	0.54	0.54	0.46	0.75	0.25	44.208	4334.415	191615.8183
8	0.37	0.37	0.63	0.52	0.48	45.084	4623.094	208427.5699
10	0.34	0.16	0.84	0.36	0.64	45.066	4743.456	213768.5881
12	0.27	0.58	0.42	0.75	0.25	45.525	5027.894	228894.8744
14	0.23	0.40	0.60	0.46	0.54	45.7	5250.807	239961.8799
16	0.19	0.57	0.43	0.60	0.40	46.354	5639.926	261433.1298
18	0.20	0.27	0.73	0.45	0.55	45.977	5678.851	261096.5324
20	0.18	0.50	0.50	0.51	0.49	46.125	5843.46	269529.5925
22	0.15	0.57	0.43	0.67	0.33	45.948	6093.368	279978.0729
24	0.14	0.37	0.63	0.79	0.21	46.156	6281.843	289944.7455
26	0.14	0.45	0.55	0.72	0.28	45.844	6430.191	294785.6762
28	0.12	0.28	0.72	0.83	0.17	45.907	6601.994	303077.7386
30	0.12	0.28	0.72	0.54	0.46	45.809	6874.698	314923.0407
32	0.11	0.35	0.65	0.59	0.41	45.925	7106.988	326388.4239
34	0.10	0.52	0.48	0.57	0.43	45.626	7228.606	329812.3774
36	0.11	0.47	0.53	0.61	0.39	45.503	7546.05	343367.9132
38	0.10	0.59	0.41	0.55	0.45	45.495	7536.766	342885.1692
40	0.09	0.32	0.68	0.37	0.63	45.537	7753.804	353084.9727
42	0.09	0.24	0.76	0.36	0.64	45.423	7925.199	359986.3142
44	0.09	0.40	0.60	0.37	0.63	45.668	8002.647	365464.8832
46	0.09	0.35	0.65	0.39	0.61	45.21	8154.516	368665.6684
48	0.09	0.48	0.52	0.29	0.71	45.239	8272.071	374220.22
50	0.09	0.46	0.54	0.37	0.63	45.065	8386.651	377944.4273
52	0.08	0.50	0.50	0.43	0.57	45.025	8578.146	386231.0237
54	0.08	0.42	0.58	0.54	0.46	44.91	8796.657	395057.8659
56	0.08	0.38	0.62	0.37	0.63	44.943	8933.302	401489.3918
58	0.08	0.49	0.51	0.38	0.62	44.744	9070.102	405832.6439
60	0.08	0.36	0.64	0.48	0.52	44.404	9135.049	405632.7158
62	0.08	0.41	0.59	0.39	0.61	44.408	9248.745	410718.268
64	0.07	0.37	0.63	0.61	0.39	44.49	9050.216	402644.1098
Aplicação PO-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	4.45	0.48	0.52	0.35	0.65	79.61812	8005.837357	637409.7194
4	4.44	0.49	0.51	0.36	0.64	41.13789	5013.309891	206236.9908
6	4.63	0.48	0.52	0.45	0.55	26.39344	3740.765717	98731.67549
8	4.59	0.24	0.76	0.46	0.54	26.86789	2378.406982	63902.77718
10	4.55	0.58	0.42	0.23	0.77	15.70821	2882.297241	45275.73035
12	4.52	0.37	0.63	0.63	0.37	13.15376	2647.310303	34822.08437
14	4.43	0.30	0.70	0.77	0.23	11.64622	2465.844864	28717.77177
16	4.36	0.41	0.59	0.22	0.78	10.39357	2308.694885	23995.5819
18	4.24	0.25	0.75	0.60	0.40	9.55891	2271.262039	21710.78942
20	4.19	0.57	0.43	0.49	0.51	8.73739	2207.632858	19288.94926
22	3.95	0.42	0.58	0.51	0.49	8.27507	2189.213074	18115.89143
24	3.80	0.47	0.53	0.46	0.54	7.69138	2138.755814	16449.98369
26	3.41	0.18	0.82	0.75	0.25	8.2083	2221.779892	18237.03589
28	3.30	0.31	0.69	0.34	0.66	7.62965	2127.812195	16234.46231
30	2.99	0.32	0.68	0.38	0.62	9.80481	1272.600189	12477.60306
32	3.04	0.48	0.52	0.40	0.60	7.77884	2125.426727	16533.35444
34	2.57	0.07	0.93	0.70	0.30	78.0928	7903.576447	617212.4147
36	2.46	0.42	0.58	0.26	0.74	7.81568	2013.98349	15740.65048
38	2.38	0.40	0.60	0.68	0.32	8.55022	2111.56337	18054.33136
40	2.34	0.28	0.72	0.33	0.67	8.18169	1990.428345	16285.06768
42	2.06	0.47	0.53	0.32	0.68	9.12467	2113.064972	19281.02056
44	1.98	0.19	0.81	0.34	0.66	9.2314	2079.327789	19195.10655
46	1.74	0.42	0.58	0.56	0.44	9.90861	2150.489624	21308.36299
48	1.78	0.51	0.49	0.39	0.61	9.95087	2139.411469	21289.0054
50	1.71	0.53	0.47	0.49	0.51	9.76978	2084.553925	20365.63324
52	1.56	0.50	0.50	0.59	0.41	10.52609	1450.796371	15271.21318
54	1.46	0.49	0.51	0.37	0.63	9.8892	2050.953522	20282.28957
56	1.44	0.44	0.56	0.56	0.44	9.86877	2034.788361	20080.85833
58	1.38	0.54	0.46	0.27	0.73	10.05124	2054.678604	20652.06777
60	1.35	0.41	0.59	0.34	0.66	9.90248	2019.611862	19999.16607
62	1.31	0.27	0.73	0.37	0.63	10.31746	2142.882187	22109.10125
64	1.18	0.50	0.50	0.65	0.35	10.87266	2153.739746	23416.87999

Aplicação BT-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	2.55	0.23	0.77	0.48	0.52	95.27634	6561.962692	625199.7885
4	2.54	0.35	0.65	0.53	0.47	54.35186	3961.469604	215313.2413
6	2.23	0.37	0.63	0.49	0.51	41.33298	3195.8414	132093.6487
8	1.95	0.38	0.62	0.38	0.62	35.27199	2876.567993	101462.2775
10	1.86	0.48	0.52	0.40	0.60	29.91241	2541.003906	76007.55066
12	1.62	0.44	0.56	0.60	0.40	28.52207	2473.181137	70540.24551
14	1.51	0.28	0.72	0.82	0.18	26.17558	2328.337723	60945.59033
16	1.39	0.14	0.86	0.47	0.53	24.91849	2265.386658	56450.01478
18	1.41	0.29	0.71	0.45	0.55	22.19036	2083.327744	46229.79263
20	1.33	0.46	0.54	0.70	0.30	20.80095	1999.967773	41601.22966
22	1.34	0.45	0.55	0.59	0.41	18.89009	1885.828918	35623.47799
24	1.24	0.39	0.61	0.32	0.68	18.82276	1880.885834	35403.46264
26	1.29	0.35	0.65	0.45	0.55	16.12171	1683.009766	27132.99537
28	1.30	0.36	0.64	0.42	0.58	15.4957	1648.666428	25547.24036
30	1.31	0.33	0.67	0.46	0.54	14.5319	1592.446884	23141.27888
32	1.28	0.41	0.59	0.65	0.35	13.77728	1535.27829	21151.95888
34	1.42	0.38	0.62	0.34	0.66	11.58459	1362.963226	15789.37016
36	1.38	0.48	0.52	0.48	0.52	11.44188	1365.024246	15618.44362
38	1.36	0.45	0.55	0.73	0.27	10.99891	1336.696869	14702.20856
40	1.31	0.34	0.66	0.36	0.64	10.72256	1324.748962	14204.70023
42	1.29	0.24	0.76	0.60	0.40	10.61232	1325.841721	14070.25661
44	1.26	0.38	0.62	0.38	0.62	10.29898	1306.557663	13456.21124
46	1.24	0.38	0.62	0.32	0.68	10.07443	1296.494263	13061.44069
48	1.19	0.20	0.80	0.38	0.62	9.82987	1282.55275	12607.3268
50	1.36	0.43	0.57	0.50	0.50	8.18058	1138.902939	9316.886603
52	1.37	0.42	0.58	0.66	0.34	8.01648	1130.461563	9062.322511
54	1.24	0.44	0.56	0.53	0.47	8.0992	1149.835602	9312.748506
56	1.26	0.37	0.63	0.45	0.55	8.08113	1158.089645	9358.672976
58	1.19	0.41	0.59	0.57	0.43	8.08948	1168.325287	9451.144042
60	1.16	0.40	0.60	0.71	0.29	8.17175	1186.823837	9698.427692
62	1.12	0.38	0.62	0.65	0.35	8.15598	1193.379257	9733.177354
64	1.05	0.40	0.60	0.62	0.38	8.87705	1277.818039	11343.25462
Aplicação EP-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.56	0.29	0.71	0.66	0.34	76.92881	3063.793549	235693.9918
4	0.56	0.50	0.50	0.58	0.42	103.45152	7077.547562	732183.0531
6	0.56	0.36	0.64	0.41	0.59	68.96918	4890.5979	337300.5269
8	0.55	0.26	0.74	0.69	0.31	51.66233	3828.878128	197808.7654
10	0.56	0.42	0.58	0.40	0.60	41.39863	3177.799698	131556.5539
12	0.56	0.42	0.58	0.72	0.28	34.42508	2737.484924	94238.13752
14	0.56	0.46	0.54	0.35	0.65	29.58817	2414.633087	71444.57427
16	0.56	0.38	0.62	0.52	0.48	25.8909	2171.817429	56230.30786
18	0.56	0.37	0.63	0.79	0.21	23.00829	1994.793213	45896.78073
20	1.79	0.39	0.61	0.51	0.49	20.70713	1849.437103	38296.53452
22	0.56	0.50	0.50	0.37	0.63	18.81591	1726.148727	32479.0591
24	0.56	0.33	0.67	0.52	0.48	17.26506	1619.921295	27968.03836
26	0.49	0.35	0.65	0.56	0.44	15.94276	1537.400589	24510.40861
28	0.56	0.32	0.68	0.42	0.58	14.83703	1487.609772	22071.71081
30	0.56	0.33	0.67	0.42	0.58	14.97652	1490.509888	22322.65114
32	0.56	0.47	0.53	0.55	0.45	12.96882	1368.717148	17750.64632
34	0.55	0.38	0.62	0.49	0.51	12.20072	1317.712555	16077.04192
36	0.56	0.44	0.56	0.52	0.48	11.53636	1278.035416	14743.87665
38	0.56	0.30	0.70	0.44	0.56	11.39635	1269.590836	14468.70152
40	0.56	0.51	0.49	0.38	0.62	11.24076	1261.321838	14178.21607
42	0.56	0.35	0.65	0.50	0.50	10.7001	1229.159088	13152.12516
44	0.56	0.60	0.40	0.92	0.08	9.4808	1134.912186	10759.87545
46	0.56	0.27	0.73	0.41	0.59	9.03627	1113.896378	10065.46842
48	0.56	0.33	0.67	0.49	0.51	8.75483	1087.497681	9520.85732
50	0.55	0.55	0.45	0.46	0.54	9.01202	1101.088013	9923.027192
52	0.52	0.46	0.54	0.62	0.38	8.00293	1034.241913	8276.965632
54	0.56	0.58	0.42	0.51	0.49	7.88717	1037.493347	8182.886403
56	0.56	0.42	0.58	0.75	0.25	7.44529	997.7214508	7428.325541
58	0.56	0.43	0.57	0.59	0.41	7.57916	1017.490921	7711.726489
60	0.56	0.35	0.65	0.51	0.49	7.52348	1023.197479	7698.005771
62	0.56	0.49	0.51	0.52	0.48	7.28001	1004.774612	7314.769226
64	0.56	0.35	0.65	0.44	0.56	7.0444	987.2722778	6954.740834

Aplicação JA-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.78	0.27	0.73	0.37	0.63	448.47952	31396.9924	14080908.08
4	0.38	0.35	0.65	0.48	0.52	460.34787	32687.03908	15047408.82
6	0.27	0.45	0.55	0.76	0.24	465.99469	34086.83914	15884286.04
8	0.19	0.40	0.60	0.46	0.54	472.76926	35086.20573	16587679.52
10	0.17	0.55	0.45	0.44	0.56	475.27579	36041.78854	17129789.52
12	0.13	0.50	0.50	0.51	0.49	539.13037	41314.45435	22273877.06
14	0.11	0.45	0.55	0.42	0.58	487.36804	38026.5379	18532919.25
16	0.09	0.24	0.76	0.70	0.30	492.36157	38370.81371	18892314.08
18	0.09	0.46	0.54	0.95	0.05	492.40442	38923.40965	19166058.95
20	0.08	0.51	0.49	0.24	0.76	494.44009	39406.45296	19484130.15
22	0.07	0.27	0.73	0.24	0.76	498.11502	40747.18193	20296783.34
24	0.06	0.42	0.58	0.32	0.68	498.69766	41894.61255	20892745.24
26	0.06	0.32	0.68	0.42	0.58	499.62494	42432.30479	21200237.74
28	0.06	0.34	0.66	0.56	0.44	499.72833	42947.33847	21462001.73
30	0.06	0.43	0.57	0.24	0.76	499.33826	43917.71989	21929797.84
32	0.05	0.38	0.62	0.20	0.80	499.60211	44943.90094	22454067.74
34	0.05	0.39	0.61	0.30	0.70	500.23267	45235.67876	22628364.36
36	0.05	0.45	0.55	0.25	0.75	499.20135	45711.66046	22819322.61
38	0.05	0.43	0.57	0.32	0.68	499.92358	46131.22078	23062085.04
40	0.05	0.39	0.61	0.41	0.59	500.74222	47089.10095	23579500.95
42	0.04	0.42	0.58	0.40	0.60	502.16968	47789.79852	23998587.83
44	0.04	0.52	0.48	0.31	0.69	501.36053	48454.94247	24293395.64
46	0.04	0.29	0.71	0.37	0.63	501.28763	48899.17632	24512552.2
48	0.04	0.33	0.67	0.30	0.70	502.00748	49512.37892	24855584.57
50	0.04	0.36	0.64	0.58	0.42	502.83624	50281.01981	25283118.94
52	0.04	0.34	0.66	0.74	0.26	503.26187	51069.81171	25701488.94
54	0.04	0.45	0.55	0.73	0.27	503.38211	52054.23242	26203169.35
56	0.04	0.30	0.70	0.87	0.13	503.4184	35367.8588	17804830.89
58	0.03	0.37	0.63	0.43	0.57	504.9967	53555.30246	27045251.01
60	0.03	0.38	0.62	0.42	0.58	505.83331	54268.03085	27450577.67
62	0.03	0.25	0.75	0.71	0.29	507.40125	54430.45148	27618079.12
64	0.04	0.39	0.61	0.44	0.56	507.44482	55276.0238	28049531.95
Aplicação ST-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.35	0.44	0.56	0.87	0.13	283.33368	19695.97336	5580532.613
4	0.17	0.50	0.50	0.92	0.08	293.75537	20628.82503	6059828.129
6	0.12	0.45	0.55	0.94	0.06	297.86481	21503.81081	6405228.52
8	0.08	0.42	0.58	0.95	0.05	303.46854	30408.77536	9228106.662
10	0.08	0.57	0.43	0.93	0.07	305.16052	22718.41881	6932764.497
12	0.07	0.42	0.58	0.93	0.07	308.94058	33675.20311	10403636.78
14	0.05	0.48	0.52	0.93	0.07	311.21671	23974.23999	7461184.095
16	0.04	0.44	0.56	0.93	0.07	314.78262	37639.84404	11848368.72
18	0.05	0.41	0.59	0.80	0.20	311.5831	24604.99422	7666500.374
20	0.04	0.45	0.55	0.94	0.06	315.07343	39955.24126	12588834.91
22	0.04	0.62	0.38	0.94	0.06	307.58841	25050.62669	7705282.434
24	0.03	0.41	0.59	0.92	0.08	364.09625	50098.3651	18240626.86
26	0.04	0.41	0.59	0.95	0.05	321.56497	26932.40735	8660518.761
28	0.03	0.36	0.64	0.92	0.08	328.52121	47237.78517	15518614.34
30	0.03	0.43	0.57	0.93	0.07	318.65573	28178.91859	8979373.875
32	0.03	0.41	0.59	0.93	0.07	307.23749	27205.8232	8358648.832
34	0.06	0.44	0.56	0.94	0.06	277.32855	24947.53383	6918663.383
36	0.04	0.42	0.58	0.91	0.09	274.02225	25021.25203	6856379.779
38	0.04	0.46	0.54	0.91	0.09	244.22418	41117.51367	10041891.06
40	0.04	0.49	0.51	0.93	0.07	234.66705	22041.01053	5172298.92
42	0.05	0.44	0.56	0.94	0.06	204.40697	19405.67314	3966654.848
44	0.04	0.53	0.47	0.90	0.10	192.33723	18545.82733	3567053.057
46	0.05	0.11	0.89	0.43	0.57	165.45966	16142.93089	2671003.857
48	0.05	0.41	0.59	0.92	0.08	151.14049	14947.9975	2259247.666
50	0.06	0.50	0.50	0.94	0.06	127.46652	12770.40187	1627798.685
52	0.06	0.43	0.57	0.93	0.07	117.13755	11949.83234	1399774.083
54	0.07	0.50	0.50	0.90	0.10	94.81336	9839.884521	932952.5135
56	0.07	0.50	0.50	0.90	0.10	84.42946	8897.305481	751194.6972
58	0.08	0.19	0.81	0.90	0.10	72.44395	7760.288345	562185.9409
60	0.08	0.48	0.52	0.90	0.10	58.08655	6363.618546	369640.6468
62	0.09	0.51	0.49	0.82	0.18	48.46329	10610.24338	514207.3018
64	0.10	0.42	0.58	0.90	0.10	42.72093	4853.835129	207360.3508

Aplicação CG-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.49	0.30	0.70	0.74	0.26	25.58631	1816.156998	46468.75595
4	1.03	0.36	0.64	0.63	0.37	18.57765	1368.65683	25426.42755
6	0.70	0.31	0.69	0.54	0.46	18.48726	1398.604523	25856.36545
8	0.52	0.54	0.46	0.42	0.58	18.68952	1450.900818	27116.63985
10	0.43	0.34	0.66	0.58	0.42	18.64668	1474.818024	27500.45975
12	0.36	0.08	0.92	0.93	0.07	18.79116	1507.852829	28334.30377
14	0.31	0.40	0.60	0.74	0.26	18.51271	1513.530777	28019.55635
16	0.26	0.42	0.58	0.54	0.46	19.16455	1589.333282	30458.85716
18	0.25	0.32	0.68	0.43	0.57	18.11058	1526.199493	27640.35802
20	0.22	0.37	0.63	0.50	0.50	19.01292	1617.282928	30749.27094
22	0.22	0.40	0.60	0.40	0.60	17.66306	1532.230316	27063.87601
24	0.19	0.49	0.51	0.37	0.63	18.52658	1628.127518	30163.63471
26	0.20	0.39	0.61	0.52	0.48	16.59794	1486.470673	24672.35104
28	0.18	0.48	0.52	0.31	0.69	16.70387	1515.973602	25322.62598
30	0.21	0.41	0.59	0.51	0.49	14.16397	1315.054901	18626.39817
32	0.19	0.33	0.67	0.49	0.51	13.54307	1281.800232	17359.51027
34	0.21	0.51	0.49	0.68	0.32	11.54746	1118.314178	12913.68824
36	0.23	0.55	0.45	0.52	0.48	10.6869	1052.865723	11251.87069
38	0.26	0.49	0.51	0.62	0.38	8.40137	856.5329285	7196.050049
40	0.27	0.29	0.71	0.48	0.52	8.15811	841.1046905	6861.824587
42	0.32	0.55	0.45	0.46	0.54	5.85965	635.4142456	3723.305084
44	0.33	0.45	0.55	0.50	0.50	5.21317	579.7826996	3022.505776
46	0.40	0.29	0.71	0.71	0.29	4.61535	530.1525879	2446.839747
48	0.40	0.37	0.63	0.68	0.32	3.82087	454.9841614	1738.435333
50	0.48	0.37	0.63	0.40	0.60	3.73826	450.5748291	1684.365861
52	0.50	0.39	0.61	0.57	0.43	2.5425	331.3177033	842.3752605
54	0.55	0.29	0.71	0.51	0.49	2.95388	375.9908752	1110.631927
56	0.64	0.52	0.48	0.43	0.57	2.53993	335.8345489	852.9962459
58	0.53	0.41	0.59	0.39	0.61	2.28469	312.9785156	715.0588849
60	0.60	0.47	0.53	0.45	0.55	2.23635	309.8123932	692.8489455
62	0.55	0.22	0.78	0.50	0.50	2.27642	317.0992432	721.8510591
64	0.52	0.35	0.65	0.57	0.43	2.6367	348.997757	920.2023858
Aplicação FT-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	2.44	0.39	0.61	0.55	0.45	16.32483	1130.579636	18456.52035
4	1.95	0.43	0.57	0.47	0.53	10.23039	750.4311066	7677.202888
6	1.55	0.35	0.65	0.73	0.27	8.55901	658.4017334	5635.26702
8	1.27	0.27	0.73	0.53	0.47	8.16976	639.5609894	5225.059789
10	1.10	0.44	0.56	0.71	0.29	8.06005	641.8624268	5173.443253
12	0.89	0.33	0.67	0.45	0.55	8.06922	651.0158691	5253.190272
14	0.77	0.43	0.57	0.48	0.52	8.0527	659.1500855	5307.937893
16	0.67	0.56	0.44	0.54	0.46	8.04225	668.1246338	5373.225336
18	0.62	0.46	0.54	0.85	0.15	8.05658	674.7997284	5436.577996
20	0.57	0.41	0.59	0.55	0.45	8.02814	678.0524445	5443.499951
22	0.51	0.27	0.73	0.49	0.51	8.03095	684.9429779	5500.742808
24	0.46	0.41	0.59	0.58	0.42	8.05469	695.5918274	5602.776536
26	0.43	0.43	0.57	0.81	0.19	8.03357	701.4694977	5635.304312
28	0.41	0.32	0.68	0.56	0.44	8.0673	713.4270477	5755.430022
30	0.38	0.46	0.54	0.46	0.54	8.0301	719.3886414	5776.762729
32	0.34	0.37	0.63	0.57	0.43	8.05014	732.7520752	5898.756791
34	0.35	0.43	0.57	0.77	0.23	8.04834	737.6843414	5937.134393
36	0.33	0.31	0.69	0.60	0.40	8.07434	744.0095978	6007.386456
38	0.30	0.37	0.63	0.57	0.43	8.02817	748.9690399	6012.850777
40	0.30	0.39	0.61	0.55	0.45	8.04521	755.8043518	6080.604729
42	0.29	0.43	0.57	0.59	0.41	8.06276	765.0428925	6168.357232
44	0.28	0.14	0.86	0.68	0.32	8.05907	775.1430206	6246.931863
46	0.27	0.33	0.67	0.74	0.26	8.02421	783.7208557	6288.740728
48	0.25	0.33	0.67	0.49	0.51	8.03439	791.0321503	6355.460798
50	0.24	0.39	0.61	0.46	0.54	8.01581	796.4584961	6384.259978
52	0.26	0.11	0.89	0.59	0.41	8.02104	802.4994812	6436.880439
54	0.23	0.39	0.61	0.50	0.50	7.97854	805.2490387	6424.711665
56	0.22	0.37	0.63	0.58	0.42	8.00026	815.3805695	6523.256555
58	0.22	0.44	0.56	0.69	0.31	7.96507	821.0848846	6539.998582
60	0.20	0.47	0.53	0.79	0.21	8.00003	833.1579285	6665.288422
62	0.20	0.48	0.52	0.56	0.44	7.98074	839.2216644	6697.609906
64	0.19	0.60	0.40	0.67	0.33	8.2693	870.0890808	7195.027636

Aplicação LU-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	2.17	0.23	0.77	0.40	0.60	58.3969	3991.646851	233099.802
4	1.79	0.43	0.57	0.55	0.45	36.05655	2601.708771	93808.64238
6	1.53	0.22	0.78	0.50	0.50	27.81089	2119.128326	58934.84478
8	1.32	0.40	0.60	0.62	0.38	24.02554	1890.258621	45414.48411
10	1.22	0.35	0.65	0.63	0.37	20.95115	1714.732559	35925.61906
12	1.05	0.32	0.68	0.63	0.37	20.12794	1674.917511	33712.63917
14	0.99	0.41	0.59	0.81	0.19	18.56931	1583.53093	29405.07673
16	0.89	0.46	0.54	0.58	0.42	17.91309	1556.287659	27877.9209
18	0.88	0.38	0.62	0.50	0.50	16.48085	1465.648331	24155.13029
20	0.85	0.31	0.69	0.59	0.41	15.22345	1381.651001	21033.49493
22	0.82	0.28	0.72	0.57	0.43	14.34353	1332.12413	19107.36243
24	0.78	0.44	0.56	0.79	0.21	13.84588	1307.155533	18098.71865
26	0.76	0.46	0.54	0.53	0.47	13.23828	1277.19519	16907.86755
28	0.73	0.25	0.75	0.59	0.41	12.85518	1262.823227	16233.81989
30	0.70	0.36	0.64	0.55	0.45	12.39371	1254.767136	15551.22
32	0.68	0.18	0.82	0.88	0.12	12.06619	1221.922562	14743.94979
34	0.65	0.41	0.59	0.57	0.43	11.93292	1225.912384	14628.71441
36	0.61	0.56	0.44	0.50	0.50	12.02996	1248.365753	15017.79008
38	0.58	0.34	0.66	0.76	0.24	12.01681	1267.26564	15228.49042
40	0.56	0.49	0.51	0.66	0.34	12.00608	1273.776016	15293.05675
42	0.53	0.42	0.58	0.59	0.41	12.04549	1292.583038	15569.79606
44	0.51	0.34	0.66	0.59	0.41	12.02978	1306.998627	15722.90594
46	0.49	0.50	0.50	0.59	0.41	12.01583	1319.784805	15858.30986
48	0.47	0.40	0.60	0.74	0.26	11.97362	1329.793823	15922.44592
50	0.46	0.38	0.62	0.74	0.26	11.68077	1307.803009	15276.14615
52	0.45	0.62	0.38	0.63	0.37	11.70969	1328.609512	15557.60552
54	0.43	0.29	0.71	0.68	0.32	11.78997	1345.512863	15863.55629
56	0.42	0.33	0.67	0.76	0.24	11.78993	1371.578964	16170.81998
58	0.40	0.40	0.60	0.76	0.24	11.83033	1394.36731	16495.82541
60	0.39	0.31	0.69	0.74	0.26	11.82488	1409.842773	16671.22161
62	0.38	0.42	0.58	0.75	0.25	11.90111	1439.689835	17133.90709
64	0.36	0.66	0.34	0.60	0.40	12.16517	1477.372772	17972.49093
Aplicação MG-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.82	0.38	0.62	0.60	0.40	13.8788	969.8923798	13460.94236
4	0.96	0.43	0.57	0.48	0.52	13.33583	948.7565918	12652.45662
6	0.66	0.40	0.60	0.72	0.28	13.05426	962.2242737	12561.12585
8	0.49	0.49	0.51	0.68	0.32	13.15456	984.4442596	12949.93108
10	0.42	0.45	0.55	0.58	0.42	13.09361	999.8122559	13091.15175
12	0.33	0.23	0.77	0.72	0.28	13.23221	1023.875214	13548.13184
14	0.28	0.35	0.65	0.46	0.54	13.29822	1043.328201	13874.40795
16	0.24	0.36	0.64	0.52	0.48	13.46932	1069.655975	14407.53862
18	0.23	0.36	0.64	0.37	0.63	13.46395	1081.684967	14563.75231
20	0.20	0.48	0.52	0.54	0.46	13.46012	1092.505707	14705.25791
22	0.19	0.12	0.88	0.64	0.36	13.50263	1112.013199	15015.10278
24	0.17	0.43	0.57	0.55	0.45	13.60487	1136.834122	15466.48044
26	0.16	0.33	0.67	0.69	0.31	13.66842	1158.710098	15837.73628
28	0.15	0.35	0.65	0.47	0.53	13.57365	1168.684143	15863.30952
30	0.14	0.49	0.51	0.54	0.46	13.3077	1161.740631	15460.0958
32	0.13	0.45	0.55	0.59	0.41	13.49134	1195.140152	16124.04214
34	0.13	0.34	0.66	0.70	0.30	13.28016	1191.01265	15816.83855
36	0.12	0.51	0.49	0.60	0.40	13.44061	1221.77301	16421.37454
38	0.12	0.36	0.64	0.59	0.41	13.15222	1207.973267	15887.53016
40	0.12	0.50	0.50	0.71	0.29	13.43992	1246.578354	16753.91335
42	0.12	0.35	0.65	0.69	0.31	13.213	1249.161392	16505.16948
44	0.11	0.31	0.69	0.59	0.41	13.34534	1276.034515	17029.11446
46	0.11	0.36	0.64	0.63	0.37	12.97666	1253.550629	16266.9003
48	0.11	0.49	0.51	0.45	0.55	13.04066	1270.723084	16571.06769
50	0.11	0.37	0.63	0.64	0.36	12.78256	1265.951782	16182.10461
52	0.11	0.43	0.57	0.68	0.32	12.74608	1274.405609	16243.67585
54	0.11	0.59	0.41	0.77	0.23	12.00526	1221.337173	14662.47032
56	0.11	0.36	0.64	0.70	0.30	11.65883	1204.215744	14039.74664
58	0.11	0.31	0.69	0.71	0.29	10.99671	1142.420258	12562.86427
60	0.11	0.19	0.81	0.57	0.43	10.74828	1130.55928	12151.5677
62	0.11	0.50	0.50	0.56	0.44	10.24933	1093.348572	11206.09032
64	0.11	0.32	0.68	0.53	0.47	10.28806	1065.702606	10964.01235

Aplicação UA-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.30	0.37	0.63	0.73	0.27	71.10899	4832.161484	343610.1226
4	0.91	0.32	0.68	0.55	0.45	51.55061	3649.031265	188109.7876
6	0.86	0.41	0.59	0.59	0.41	37.00473	2710.23613	100291.5562
8	0.71	0.37	0.63	0.63	0.37	33.18122	2492.588654	82707.13248
10	0.76	0.34	0.66	0.79	0.21	25.04849	1946.524826	48757.50764
12	0.68	0.43	0.57	0.67	0.33	23.01073	1825.025726	41995.17423
14	0.75	0.34	0.66	0.77	0.23	17.99687	1478.598648	26610.14765
16	0.70	0.38	0.62	0.61	0.39	16.78327	1403.018478	23547.23794
18	0.81	0.39	0.61	0.71	0.29	13.10305	1137.420547	14903.6783
20	0.83	0.29	0.71	0.76	0.24	11.72966	1043.007019	12234.11771
22	0.92	0.37	0.63	0.71	0.29	9.50763	876.0282745	8328.952704
24	0.95	0.42	0.58	0.61	0.39	8.44538	802.7343903	6779.396965
26	1.05	0.40	0.60	0.70	0.30	7.07577	700.351532	4955.526359
28	1.08	0.44	0.56	0.62	0.38	6.35832	650.3565369	4135.174976
30	1.11	0.14	0.86	0.50	0.50	5.77589	610.4768982	3526.047411
32	1.12	0.54	0.46	0.69	0.31	5.35625	579.8555908	3105.851508
34	1.08	0.45	0.55	0.61	0.39	5.23356	572.2364044	2994.833557
36	1.03	0.39	0.61	0.69	0.31	5.04326	563.8030395	2843.405317
38	0.99	0.43	0.57	0.71	0.29	4.97201	562.3835449	2796.176609
40	1.00	0.43	0.57	0.70	0.30	4.78923	554.3456726	2654.888926
42	0.94	0.30	0.70	0.62	0.38	4.82002	564.6743927	2721.741866
44	0.91	0.30	0.70	0.62	0.38	4.81604	572.5953827	2757.642267
46	0.88	0.36	0.64	0.62	0.38	4.73735	570.5460358	2702.876263
48	0.85	0.41	0.59	0.76	0.24	4.70124	572.8820343	2693.255935
50	0.81	0.33	0.67	0.65	0.35	4.74293	583.6558227	2768.238711
52	0.79	0.41	0.59	0.58	0.42	4.67564	582.6248169	2724.143899
54	0.74	0.17	0.83	0.69	0.31	4.82458	603.2108307	2910.23891
56	0.72	0.40	0.60	0.75	0.25	4.79633	610.3160248	2927.277059
58	0.67	0.48	0.52	0.75	0.25	4.89441	625.377594	3060.85435
60	0.65	0.40	0.60	0.60	0.40	5.01431	642.1468506	3219.923374
62	0.63	0.42	0.58	0.57	0.43	5.17	664.5770721	3435.863463
64	0.56	0.34	0.66	0.64	0.36	5.38183	671.3953705	3613.335747
Aplicação HS-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	2.37	0.35	0.65	0.39	0.61	323.65051	22127.29851	7161511.447
4	2.26	0.29	0.71	0.85	0.15	146.98145	10681.66478	1570006.578
6	1.93	0.40	0.60	0.64	0.36	88.78159	6888.934158	611610.528
8	1.71	0.48	0.52	0.56	0.44	69.52864	5730.699234	398447.724
10	1.48	0.50	0.50	0.58	0.42	58.26169	5094.806015	296832.0087
12	1.27	0.46	0.54	0.53	0.47	49.99081	4587.557449	229335.7128
14	1.09	0.36	0.64	0.70	0.30	43.99925	4195.028519	184578.1086
16	0.94	0.34	0.66	0.84	0.16	39.76873	3890.399109	154716.2318
18	0.87	0.44	0.56	0.54	0.46	36.36126	3673.924789	133588.5345
20	0.78	0.41	0.59	0.71	0.29	34.10598	3599.990936	122781.2189
22	0.74	0.47	0.53	0.81	0.19	31.37945	3339.085266	104778.6592
24	0.65	0.41	0.59	0.77	0.23	30.02757	3239.992584	97289.10412
26	0.65	0.48	0.52	0.76	0.24	28.02765	3093.940399	86715.87863
28	0.61	0.44	0.56	0.76	0.24	27.42099	3070.790649	84204.11969
30	0.55	0.50	0.50	0.75	0.25	25.81816	2965.773499	76570.81471
32	0.50	0.32	0.68	0.80	0.20	24.71695	2893.299606	71513.5417
34	0.53	0.43	0.57	0.76	0.24	27.10789	3189.805511	86468.89693
36	0.52	0.38	0.62	0.68	0.32	29.73326	3427.162262	101900.7066
38	0.48	0.40	0.60	0.85	0.15	28.95909	3410.244064	98757.56478
40	0.45	0.43	0.57	0.83	0.17	29.01763	3432.387527	99599.75129
42	0.47	0.38	0.62	0.87	0.13	30.32207	3592.997971	108947.136
44	0.45	0.45	0.55	0.91	0.09	29.08992	3528.000671	102629.2573
46	0.42	0.41	0.59	0.76	0.24	29.82129	3647.624847	108776.8784
48	0.41	0.50	0.50	0.82	0.18	29.47194	3606.11232	106279.1259
50	0.42	0.48	0.52	0.67	0.33	28.0684	3516.34494	98698.17632
52	0.41	0.46	0.54	0.81	0.19	29.23657	3672.753616	107378.7182
54	0.41	0.44	0.56	0.70	0.30	28.29398	3612.582581	102214.3393
56	0.38	0.51	0.49	0.65	0.35	29.66499	3834.238785	113742.6552
58	0.36	0.32	0.68	0.86	0.14	31.55962	4029.648071	127174.1619
60	0.35	0.44	0.56	0.68	0.32	30.41909	3965.978195	120641.4477
62	0.36	0.40	0.60	0.83	0.17	33.37392	4328.259476	144450.9855
64	0.34	0.55	0.45	0.64	0.36	23.57021	3288.433945	77509.07865

Aplicação SP-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.96	0.49	0.51	0.65	0.35	59.60001	4177.452484	248976.2098
4	1.16	0.39	0.61	0.67	0.33	50.34163	3613.821533	181925.6665
6	0.83	0.37	0.63	0.52	0.48	46.9758	3488.511169	163875.603
8	0.61	0.20	0.80	0.59	0.41	48.58145	3715.305084	180494.9082
10	0.54	0.51	0.49	0.94	0.06	44.28872	3429.93811	151907.5686
12	0.44	0.34	0.66	0.57	0.43	45.41972	3560.937469	161736.7828
14	0.42	0.25	0.75	0.73	0.27	40.94308	3288.532822	134642.6624
16	0.36	0.37	0.63	0.76	0.24	41.28819	3353.979843	138479.757
18	0.38	0.39	0.61	0.66	0.34	36.41536	3015.957962	109827.1949
20	0.33	0.46	0.54	0.69	0.31	36.03621	3020.277695	108839.3613
22	0.37	0.43	0.57	0.68	0.32	31.55038	2700.558151	85203.63588
24	0.34	0.43	0.57	0.57	0.43	30.46328	2649.207504	80703.54998
26	0.36	0.16	0.84	0.52	0.48	26.94568	2398.839249	64638.35477
28	0.35	0.43	0.57	0.68	0.32	25.45456	2309.529282	58788.05167
30	0.37	0.35	0.65	0.58	0.42	23.16323	2147.161499	49735.19565
32	0.35	0.36	0.64	0.56	0.44	22.00757	2068.972031	45533.04679
34	0.36	0.38	0.62	0.60	0.40	19.87192	1905.251968	37861.0147
36	0.36	0.35	0.65	0.51	0.49	19.71504	1911.649368	37688.24376
38	0.36	0.47	0.53	0.60	0.40	18.74468	1847.535049	34631.45329
40	0.35	0.48	0.52	0.71	0.29	18.65241	1861.143539	34714.81237
42	0.35	0.22	0.78	0.63	0.37	17.91047	1815.242325	32511.8432
44	0.34	0.38	0.62	0.54	0.46	17.57553	1805.957642	31740.66271
46	0.34	0.45	0.55	0.72	0.28	17.20084	1787.61203	30748.42851
48	0.32	0.44	0.56	0.69	0.31	17.02142	1789.891846	30466.50086
50	0.31	0.16	0.84	0.80	0.20	16.73052	1777.276733	29734.76393
52	0.30	0.38	0.62	0.52	0.48	16.46208	1768.398956	29111.52509
54	0.30	0.41	0.59	0.65	0.35	16.41892	1784.391098	29297.77469
56	0.29	0.42	0.58	0.79	0.21	16.40688	1805.646133	29625.01943
58	0.28	0.43	0.57	0.60	0.40	16.48531	1833.505768	30225.91097
60	0.28	0.52	0.48	0.66	0.34	16.1923	1825.28717	29555.59745
62	0.27	0.50	0.50	0.65	0.35	16.3205	1858.298782	30328.36528
64	0.26	0.50	0.50	0.75	0.25	16.6468	1879.409073	31286.14695
Aplicação SC-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.34	0.20	0.80	0.91	0.09	337.77582	22976.9493	7761057.889
4	1.29	0.49	0.51	0.36	0.64	144.75774	14160.94255	2049906.04
6	1.03	0.48	0.52	0.88	0.12	151.45427	11200.53197	1696368.393
8	0.78	0.49	0.51	0.93	0.07	149.52676	11359.31514	1698521.589
10	0.73	0.38	0.62	0.92	0.08	150.57939	11589.91307	1745202.04
12	0.60	0.49	0.51	0.91	0.09	146.43613	16566.54204	2425940.304
14	0.50	0.50	0.50	0.85	0.15	152.6629	12128.67558	1851598.788
16	0.42	0.40	0.60	0.93	0.07	154.74443	12442.9837	1925482.421
18	0.43	0.52	0.48	0.93	0.07	149.38524	18716.43399	2795958.984
20	0.39	0.42	0.58	0.93	0.07	156.05289	12868.51506	2008168.965
22	0.33	0.46	0.54	0.93	0.07	158.27643	13257.85669	2098406.226
24	0.30	0.45	0.55	0.92	0.08	161.66792	13735.8187	2220641.238
26	0.31	0.49	0.51	0.93	0.07	164.44489	14140.44774	2325324.373
28	0.28	0.43	0.57	0.94	0.06	153.72701	21989.88768	3380439.683
30	0.26	0.54	0.46	0.93	0.07	171.16283	15162.08865	2595186.003
32	0.24	0.43	0.57	0.89	0.11	175.55754	15856.03784	2783646.998
34	0.22	0.41	0.59	0.93	0.07	180.33331	16557.62126	2985890.648
36	0.22	0.44	0.56	0.93	0.07	185.1814	17150.78323	3176006.05
38	0.21	0.59	0.41	0.93	0.07	156.11131	25518.22052	3983682.834
40	0.21	0.47	0.53	0.94	0.06	196.72034	18734.54407	3685465.879
42	0.20	0.52	0.48	0.94	0.06	203.8289	19676.10864	4010559.581
44	0.20	0.47	0.53	0.92	0.08	208.69571	20422.36166	4262059.267
46	0.19	0.49	0.51	0.94	0.06	157.62996	28165.60268	4439742.823
48	0.19	0.48	0.52	0.95	0.05	215.22104	21655.1853	4660651.502
50	0.19	0.44	0.56	0.90	0.10	217.23047	22124.00009	4806006.938
52	0.19	0.48	0.52	0.83	0.17	219.36292	22676.75113	4974438.344
54	0.18	0.44	0.56	0.94	0.06	157.98659	30445.06902	4809912.636
56	0.17	0.44	0.56	0.85	0.15	225.85275	24027.33968	5426640.741
58	0.16	0.40	0.60	0.93	0.07	231.869	24978.62056	5791767.771
60	0.17	0.47	0.53	0.93	0.07	237.935	25875.72206	6156739.929
62	0.16	0.38	0.62	0.93	0.07	159.27599	32739.76578	5214658.607
64	0.16	0.49	0.51	0.91	0.09	168.06239	17760.90585	2984940.286

Aplicação LUD-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	2.37	0.33	0.67	0.78	0.22	136.44362	9172.627945	1251546.562
4	2.26	0.43	0.57	0.48	0.52	71.73212	5288.966919	379388.8097
6	1.93	0.46	0.54	0.50	0.50	56.62048	4270.274658	241785.0009
8	1.71	0.49	0.51	0.68	0.32	47.64098	3787.30957	180431.1395
10	1.48	0.36	0.64	0.49	0.51	43.89862	3590.247726	157606.9206
12	1.27	0.50	0.50	0.73	0.27	43.03886	3572.081635	153738.3214
14	1.09	0.46	0.54	0.75	0.25	42.7725	3606.998886	154280.3599
16	0.94	0.32	0.68	0.61	0.39	42.95131	3651.87825	156852.9548
18	0.87	0.36	0.64	0.37	0.63	42.61683	3656.344208	155821.7995
20	0.78	0.44	0.56	0.78	0.22	42.78196	3790.35556	162158.84
22	0.74	0.43	0.57	0.44	0.56	42.51099	3709.363739	157688.7248
24	0.65	0.47	0.53	0.87	0.13	42.7017	3765.029343	160773.1535
26	0.65	0.47	0.53	0.49	0.51	42.26617	3764.337891	159104.1452
28	0.61	0.43	0.57	0.62	0.38	42.28935	3803.720215	160856.8555
30	0.55	0.34	0.66	0.71	0.29	41.97054	3828.196487	160671.4738
32	0.50	0.41	0.59	0.89	0.11	42.04092	3887.445389	163431.7806
34	0.53	0.40	0.60	0.69	0.31	41.54115	3870.426498	160781.9677
36	0.52	0.39	0.61	0.63	0.37	41.43175	4011.323929	166196.1702
38	0.48	0.38	0.62	0.84	0.16	40.95646	3881.973831	158991.9059
40	0.45	0.49	0.51	0.36	0.64	40.9836	3918.624634	160599.3445
42	0.47	0.41	0.59	0.56	0.44	40.53535	3912.684357	158602.0298
44	0.45	0.35	0.65	0.76	0.24	40.63953	3957.402145	160826.9632
46	0.42	0.40	0.60	0.89	0.11	40.17358	3948.105911	158609.5487
48	0.41	0.48	0.52	0.75	0.25	40.0607	3983.637726	159587.3158
50	0.42	0.39	0.61	0.72	0.28	39.50403	3970.760788	156861.0533
52	0.41	0.45	0.55	0.81	0.19	39.49419	4014.654724	158555.5365
54	0.41	0.44	0.56	0.77	0.23	38.86213	3989.825089	155053.1013
56	0.38	0.41	0.59	0.81	0.19	38.91495	4033.182266	156951.0862
58	0.36	0.43	0.57	0.72	0.28	38.34538	4010.364929	153778.9671
60	0.35	0.49	0.51	0.86	0.14	38.32727	4062.544617	155706.2444
62	0.36	0.50	0.50	0.86	0.14	37.78133	4050.945526	153050.1097
64	0.34	0.50	0.50	0.90	0.10	37.99408	4082.548721	155112.6827
Aplicação BFS-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.59	0.50	0.50	0.46	0.54	1.76336	147.1783142	259.5283521
4	0.71	0.35	0.65	0.69	0.31	1.79956	154.1945038	277.4822612
6	0.54	0.44	0.56	0.48	0.52	3.2548	282.4822845	919.4233398
8	0.37	0.43	0.57	0.45	0.55	3.52683	317.9732819	1121.43771
10	0.41	0.36	0.64	0.39	0.61	4.43386	402.5302582	1784.762811
12	0.35	0.44	0.56	0.68	0.32	4.30462	415.5261078	1788.681994
14	0.32	0.45	0.55	0.55	0.45	4.94697	491.1673584	2429.790187
16	0.30	0.41	0.59	0.51	0.49	4.98157	513.7072144	2559.068448
18	0.30	0.39	0.61	0.47	0.53	5.46286	576.8613434	3151.312758
20	0.26	0.22	0.78	0.53	0.47	5.92579	647.6173859	3837.644629
22	0.25	0.48	0.52	0.75	0.25	6.10055	680.3555298	4150.542927
24	0.23	0.36	0.64	0.54	0.46	6.60572	744.354126	4916.994937
26	0.25	0.41	0.59	0.48	0.52	6.83908	812.7575226	5558.513718
28	0.21	0.42	0.58	0.64	0.36	6.71053	814.0010223	5462.37828
30	0.22	0.50	0.50	0.50	0.50	7.56671	949.8762665	7187.438244
32	0.24	0.44	0.56	0.69	0.31	7.84255	1007.874329	7904.304816
34	0.22	0.34	0.66	0.71	0.29	8.81254	1178.901978	10389.12083
36	0.22	0.35	0.65	0.77	0.23	8.97367	1218.260361	10932.26645
38	0.23	0.40	0.60	0.58	0.42	9.54471	1327.763519	12673.11774
40	0.21	0.57	0.43	0.60	0.40	9.93315	1417.372299	14078.97165
42	0.54	0.48	0.52	0.60	0.40	1.76119	146.1863709	257.4619745
44	0.22	0.33	0.67	0.55	0.45	10.79133	1606.236298	17333.42595
46	0.19	0.47	0.53	0.45	0.55	11.24195	1705.857971	19177.17002
48	0.19	0.39	0.61	0.63	0.37	11.52279	1779.746445	20507.64454
50	0.20	0.29	0.71	0.55	0.45	11.96828	1896.224686	22694.54798
52	0.20	0.37	0.63	0.68	0.32	12.45243	2009.143799	25018.72251
54	0.17	0.22	0.78	0.66	0.34	12.7825	2106.021851	26920.22431
56	0.17	0.38	0.62	0.69	0.31	13.12611	2206.482132	28962.52718
58	0.17	0.31	0.69	0.63	0.37	13.66436	2335.796326	31917.16188
60	0.18	0.50	0.50	0.70	0.30	14.29306	2494.960098	35660.61438
62	0.19	0.30	0.70	0.76	0.24	14.52155	2588.509354	37589.168
64	0.19	0.42	0.58	0.65	0.35	15.39169	2786.569534	42890.01444

Aplicação MRI-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.53	0.50	0.50	0.46	0.54	15.06195	976.5475617	14708.71055
4	0.21	0.41	0.59	0.72	0.28	27.53515	1786.914444	49202.95725
6	0.17	0.38	0.62	0.69	0.31	34.51257	2278.701141	78643.83265
8	0.10	0.43	0.57	0.43	0.57	42.37773	2847.337891	120663.7163
10	0.10	0.43	0.57	0.56	0.44	43.50781	2970.354218	129233.6069
12	0.08	0.46	0.54	0.75	0.25	48.79546	3396.83252	165750.0053
14	0.07	0.41	0.59	0.57	0.43	49.06414	3470.735214	170288.6385
16	0.06	0.39	0.61	0.47	0.53	53.18146	3826.548828	203501.4534
18	0.06	0.37	0.63	0.60	0.40	54.74349	4103.790482	224655.8132
20	0.09	0.44	0.56	0.57	0.43	55.46403	4072.970825	225903.376
22	0.10	0.41	0.59	0.55	0.45	57.41413	4260.789993	244629.5506
24	0.10	0.48	0.52	0.45	0.55	58.62734	4370.551254	256233.7944
26	0.11	0.26	0.74	0.61	0.39	63.5133	4824.036789	306390.4958
28	0.11	0.26	0.74	0.66	0.34	64.85816	4986.011291	323383.5181
30	0.11	0.37	0.63	0.50	0.50	65.20608	5054.479584	329582.8001
32	0.10	0.53	0.47	0.54	0.46	70.18056	5527.870102	387949.0194
34	0.10	0.34	0.66	0.44	0.56	66.99041	5295.081345	354719.6703
36	0.11	0.44	0.56	0.55	0.45	71.04467	5709.792938	405650.3551
38	0.12	0.30	0.70	0.54	0.46	71.34155	5899.189255	420857.3052
40	0.12	0.42	0.58	0.47	0.53	71.77402	5858.793488	420509.161
42	0.12	0.32	0.68	0.46	0.54	72.64706	5948.307739	432127.0692
44	0.12	0.40	0.60	0.45	0.55	78.63112	6552.284607	515213.4772
46	0.12	0.37	0.63	0.59	0.41	86.50724	7305.982773	632020.4052
48	0.13	0.44	0.56	0.39	0.61	83.46583	7124.799072	594677.2682
50	0.12	0.38	0.62	0.60	0.40	79.66631	6803.282974	541992.4504
52	0.14	0.30	0.70	0.51	0.49	87.43603	7605.386063	664984.7639
54	0.14	0.47	0.53	0.39	0.61	91.48648	8055.483765	736967.8543
56	0.15	0.43	0.57	0.49	0.51	88.9461	8107.695953	721147.935
58	0.16	0.42	0.58	0.37	0.63	92.16657	8324.018188	767196.205
60	0.15	0.34	0.66	0.73	0.27	86.65453	7843.838211	679704.1136
62	0.17	0.45	0.55	0.42	0.58	88.5723	8086.794922	716266.0259
64	0.17	0.53	0.47	0.57	0.43	93.1463	8438.366272	786002.5963
Aplicação LBM-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	3.39	0.18	0.82	0.45	0.55	333.9545	22548.65347	7530224.296
4	3.15	0.37	0.63	0.52	0.48	153.46367	16655.82372	2556063.834
6	2.63	0.29	0.71	0.51	0.49	141.00656	10901.66319	1537206.025
8	1.74	0.30	0.70	0.50	0.50	160.67197	12423.53162	1996113.299
10	1.65	0.46	0.54	0.38	0.62	149.84821	11817.63438	1770851.359
12	1.17	0.46	0.54	0.45	0.55	163.29991	12920.89502	2109980.994
14	1.08	0.37	0.63	0.44	0.56	155.05157	12516.46857	1940698.102
16	0.86	0.38	0.62	0.48	0.52	164.40276	13321.259	2190051.747
18	0.84	0.38	0.62	0.65	0.35	159.24802	13064.86021	2080553.121
20	0.70	0.27	0.73	0.46	0.54	166.82582	13747.63034	2293459.705
22	0.68	0.39	0.61	0.46	0.54	162.64108	13572.15117	2207389.324
24	0.57	0.37	0.63	0.39	0.61	167.93892	14120.65651	2371407.804
26	0.57	0.35	0.65	0.48	0.52	164.58408	13993.57765	2303120.104
28	0.51	0.47	0.53	0.41	0.59	170.30711	14655.83081	2495992.19
30	0.50	0.44	0.56	0.76	0.24	165.99641	14442.17342	2397348.94
32	0.43	0.36	0.64	0.50	0.50	171.98592	15163.46278	2607902.097
34	0.44	0.41	0.59	0.56	0.44	166.28424	14838.00723	2467326.756
36	0.42	0.52	0.48	0.38	0.62	171.17421	15313.94653	2621352.7
38	0.43	0.20	0.80	0.72	0.28	164.8699	14928.31342	2461229.54
40	0.40	0.18	0.82	0.31	0.69	165	27655.85794	4565772.515
42	0.42	0.33	0.67	0.59	0.41	157.79529	14592.63136	2302648.498
44	0.39	0.59	0.41	0.64	0.36	157.59033	14795.34468	2331603.251
46	0.41	0.33	0.67	0.53	0.47	146.82881	26188.44907	3845218.812
48	0.39	0.41	0.59	0.56	0.44	145.85817	14034.89508	2047104.113
50	0.43	0.31	0.69	0.72	0.28	136.21423	13257.002	1805792.319
52	0.41	0.35	0.65	0.60	0.40	132.4126	25069.74385	3319549.965
54	0.44	0.41	0.59	0.46	0.54	122.88482	12338.56528	1516222.373
56	0.42	0.43	0.57	0.44	0.56	118.8495	12137.79716	1442571.124
58	0.45	0.38	0.62	0.62	0.38	109.29048	22621.06415	2472266.959
60	0.43	0.34	0.66	0.37	0.63	106.68141	11305.97356	1206137.2
62	0.46	0.36	0.64	0.50	0.50	98.71228	10590.82346	1045444.33
64	0.44	0.42	0.58	0.41	0.59	95.91451	10481.61507	1005338.973

Aplicação CUTCP-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.84	0.37	0.63	0.63	0.37	30.39394	2138.128387	64986.14592
4	1.77	0.36	0.64	0.76	0.24	15.99672	1183.853973	18937.78053
6	1.81	0.38	0.62	0.51	0.49	10.80471	841.5218658	9092.399719
8	1.72	0.36	0.64	0.61	0.39	8.45507	684.5896606	5788.253502
10	1.78	0.49	0.51	0.84	0.16	6.6824	564.0208435	3769.012885
12	1.51	0.36	0.64	0.53	0.47	6.02299	523.1703796	3151.049965
14	1.80	0.59	0.41	0.50	0.50	4.8096	439.5198669	2113.914752
16	0.72	0.33	0.67	0.43	0.57	4.49777	423.004364	1902.576338
18	1.78	0.42	0.58	0.78	0.22	3.84961	380.2183228	1463.692257
20	1.75	0.44	0.56	0.61	0.39	3.49121	357.1336365	1246.828523
22	1.61	0.39	0.61	0.47	0.53	3.32716	344.1437378	1145.021279
24	1.52	0.43	0.57	0.75	0.25	3.28756	349.0379028	1147.483048
26	1.30	0.53	0.47	0.75	0.25	3.05925	332.8104248	1018.150292
28	1.71	0.27	0.73	0.77	0.23	2.64809	299.7610931	793.7943531
30	1.68	0.41	0.59	0.49	0.51	2.54281	292.1629944	742.9149837
32	1.61	0.39	0.61	0.57	0.43	2.48085	290.1560059	719.8335271
34	1.56	0.30	0.70	0.44	0.56	2.43372	288.4722137	702.060596
36	1.45	0.43	0.57	0.90	0.10	2.32799	281.7212982	655.844365
38	1.39	0.37	0.63	0.81	0.19	2.34814	285.7537079	670.9897116
40	1.28	0.43	0.57	0.66	0.34	2.1598	270.9595184	585.2183679
42	1.20	0.50	0.50	0.83	0.17	2.03178	263.1294098	534.6210722
44	1.14	0.41	0.59	0.39	0.61	2.01655	262.3387756	529.019258
46	1.56	0.35	0.65	0.75	0.25	1.88484	252.7775421	476.4452225
48	1.06	0.41	0.59	0.77	0.23	2.02923	270.9069214	549.7324521
50	1.00	0.46	0.54	0.83	0.17	2.02917	276.7549286	561.5827984
52	1.40	0.43	0.57	0.81	0.19	1.93225	265.362381	512.7464606
54	1.49	0.54	0.46	0.44	0.56	1.73925	247.0599365	429.6989946
56	1.51	0.41	0.59	0.80	0.20	1.64481	238.8083954	392.7944368
58	1.46	0.28	0.72	0.82	0.18	1.70832	246.9516144	421.8723819
60	1.27	0.01	0.99	0.57	0.43	1.87353	265.9189758	498.2071788
62	1.22	0.34	0.66	0.81	0.19	1.92788	273.5496521	527.3709033
64	1.18	0.37	0.63	0.84	0.16	1.91994	274.1671905	526.3845558
Aplicação SGEMM-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	2.02	0.48	0.52	0.84	0.16	3.8853	278.4538269	1081.876654
4	1.79	0.21	0.79	0.72	0.28	2.7176	200.7374115	545.5239895
6	1.46	0.41	0.59	0.80	0.20	2.19547	166.0068359	364.4630281
8	1.66	0.36	0.64	0.58	0.42	2.02628	154.3450775	312.7463437
10	1.73	0.35	0.65	0.83	0.17	1.86894	144.1542206	269.415589
12	1.76	0.40	0.60	0.88	0.12	1.83647	141.5051117	259.8698925
14	1.80	0.18	0.82	0.74	0.26	1.70507	132.6796417	226.2280767
16	1.70	0.46	0.54	0.58	0.42	1.66923	130.5626526	217.9390966
18	1.60	0.46	0.54	0.74	0.26	1.62749	127.6916351	207.8168593
20	1.23	0.35	0.65	0.84	0.16	1.59124	125.7918243	200.1649826
22	1.22	0.39	0.61	0.86	0.14	1.56285	124.1503601	194.0283903
24	1.00	0.43	0.57	0.78	0.22	1.64242	128.6719666	211.3334113
26	1.22	0.44	0.56	0.57	0.43	1.56243	122.8672028	191.9714036
28	1.05	0.38	0.62	0.51	0.49	1.52799	120.7894745	184.5651091
30	1.14	0.36	0.64	0.55	0.45	1.51276	119.0090485	180.0321281
32	1.21	0.43	0.57	0.75	0.25	1.53336	121.6986389	186.607825
34	1.11	0.38	0.62	0.57	0.43	1.52337	119.9141388	182.6736016
36	1.07	0.64	0.36	0.86	0.14	1.45897	115.4525909	168.4418666
38	1.12	0.33	0.67	0.77	0.23	1.52102	120.6468964	183.5063423
40	1.08	0.30	0.70	0.75	0.25	1.45842	115.7607422	168.8277816
42	1.02	0.40	0.60	0.70	0.30	1.44851	115.4409027	167.217302
44	1.01	0.39	0.61	0.72	0.28	1.45742	116.6052399	169.9428087
46	1.03	0.44	0.56	0.93	0.07	1.46032	115.6911621	168.9461179
48	1.09	0.52	0.48	0.54	0.46	1.43614	115.4148254	165.7518474
50	1.07	0.47	0.53	0.75	0.25	1.43347	115.3863983	165.4029404
52	1.04	0.47	0.53	0.71	0.29	1.43759	114.493454	164.5946445
54	1.04	0.16	0.84	0.64	0.36	1.47671	118.8503876	175.5075558
56	1.01	0.31	0.69	0.70	0.30	1.45886	115.6780548	168.758087
58	1.03	0.36	0.64	0.76	0.24	1.49462	117.3463287	175.3881699
60	1.00	0.38	0.62	0.62	0.38	1.41497	112.1994171	158.7588092
62	1.05	0.35	0.65	0.52	0.48	1.42966	112.7055817	161.1306619
64	1.09	0.30	0.70	0.73	0.27	1.43723	113.8007507	163.557853

Aplicação SPMV-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	1.78	0.33	0.67	0.54	0.46	2.42292	167.6382751	406.1741296
4	1.44	0.46	0.54	0.65	0.35	2.07622	144.2846832	299.566745
6	1.01	0.39	0.61	0.51	0.49	1.96013	136.5323486	267.6211525
8	1.66	0.42	0.58	0.56	0.44	1.91922	133.6481781	256.5002564
10	1.09	0.46	0.54	0.58	0.42	1.89381	131.9497528	249.8877614
12	0.93	0.40	0.60	0.84	0.16	1.82444	127.6941071	232.9702367
14	0.89	0.47	0.53	0.51	0.49	1.80646	126.4477081	228.4227268
16	1.38	0.34	0.66	0.65	0.35	1.7772	124.1780701	220.6892661
18	0.93	0.48	0.52	0.84	0.16	1.8472	131.5270538	242.9567738
20	0.91	0.52	0.48	0.56	0.44	1.75685	128.8557129	226.3801592
22	0.95	0.43	0.57	0.82	0.18	1.76443	125.2449493	220.985946
24	0.95	0.34	0.66	0.55	0.45	1.73605	123.3010712	214.0568246
26	0.87	0.36	0.64	0.89	0.11	1.74874	124.1567535	217.1178812
28	0.92	0.45	0.55	0.69	0.31	1.74128	122.8614502	213.936186
30	0.88	0.46	0.54	0.73	0.27	1.71457	121.3103332	207.9950581
32	1.48	0.36	0.64	0.45	0.55	1.72537	122.6273804	211.5776033
34	0.91	0.38	0.62	0.73	0.27	1.70381	120.6663361	205.59251
36	0.91	0.40	0.60	0.48	0.52	1.70923	120.6194	206.1662971
38	0.86	0.41	0.59	0.69	0.31	1.70453	120.7980804	205.9039521
40	1.74	0.39	0.61	0.41	0.59	1.69679	120.1762848	203.9139183
42	0.89	0.38	0.62	0.69	0.31	1.69297	121.3524933	205.4461306
44	0.86	0.35	0.65	0.83	0.17	1.68391	120.5139313	202.934614
46	0.83	0.11	0.89	0.43	0.57	1.70021	121.8166657	207.1139131
48	0.79	0.50	0.50	0.71	0.29	1.70051	121.1074066	205.944356
50	0.79	0.33	0.67	0.83	0.17	1.68864	120.4784851	203.4447891
52	0.81	0.35	0.65	0.78	0.22	1.70794	121.6290588	207.7351348
54	0.76	0.55	0.45	0.55	0.45	1.70275	122.1945343	208.0667433
56	0.70	0.40	0.60	0.56	0.44	1.68028	120.4135284	202.3284436
58	0.75	0.31	0.69	0.77	0.23	1.67344	120.6895599	201.9667372
60	0.75	0.39	0.61	0.34	0.66	1.69054	120.0543976	202.9567613
62	0.75	0.35	0.65	0.74	0.26	1.69602	121.3880463	205.8765542
64	0.76	0.43	0.57	0.79	0.21	1.67815	119.2794342	200.1687825
Aplicação TPACF-AMD64								
	IPC	Hit-L2	Miss-L2	Hit-L3	Miss-L3	Tempo	Energia	EDP
2	0.58	0.42	0.58	0.58	0.42	2.86825	196.7819367	564.4197898
4	0.25	0.32	0.68	0.75	0.25	4.49108	310.5513306	1394.71087
6	0.15	0.37	0.63	0.74	0.26	6.39661	446.9516907	2858.975654
8	0.13	0.34	0.66	0.75	0.25	7.14175	501.0870972	3578.638776
10	0.11	0.47	0.53	0.80	0.20	8.43835	606.7138214	5119.663575
12	0.11	0.50	0.50	0.59	0.41	8.66864	632.1841583	5480.176882
14	0.10	0.39	0.61	0.70	0.30	9.19188	682.2213898	6270.897148
16	0.10	0.33	0.67	0.71	0.29	9.15253	688.0179138	6297.104597
18	0.10	0.55	0.45	0.62	0.38	9.47056	726.3231659	6878.687122
20	0.10	0.34	0.66	0.79	0.21	9.44524	723.8221741	6836.674151
22	0.10	0.39	0.61	0.57	0.43	9.70072	757.0190735	7343.630067
24	0.10	0.23	0.77	0.50	0.50	9.7008	763.9290008	7410.722451
26	0.10	0.46	0.54	0.62	0.38	9.97237	800.3322601	7981.209421
28	0.09	0.25	0.75	0.74	0.26	9.98174	813.0036774	8115.191327
30	0.12	0.43	0.57	0.80	0.20	10.32978	854.0625915	8822.278677
32	0.09	0.39	0.61	0.59	0.41	10.35981	863.2888031	8943.507975
34	0.09	0.41	0.59	0.84	0.16	10.87413	917.3258972	9975.121059
36	0.09	0.37	0.63	0.81	0.19	10.88578	928.8059845	10110.77761
38	0.09	0.44	0.56	0.55	0.45	11.23809	973.9412842	10945.23981
40	0.09	0.46	0.54	0.84	0.16	11.22318	982.096817	11022.24935
42	0.09	0.44	0.56	0.87	0.13	11.59981	1028.118454	11925.97872
44	0.09	0.25	0.75	0.74	0.26	11.58947	1038.780685	12038.91759
46	0.09	0.40	0.60	0.75	0.25	11.94703	1081.540955	12921.20223
48	0.09	0.42	0.58	0.93	0.07	11.99189	1100.639481	13198.74758
50	0.09	0.35	0.65	0.70	0.30	12.2847	1142.05043	14029.74692
52	0.09	0.37	0.63	0.78	0.22	12.30816	1151.633789	14174.49294
54	0.09	0.32	0.68	0.50	0.50	12.57152	1193.601563	15005.38592
56	0.09	0.37	0.63	0.63	0.37	12.68941	1219.409439	15473.58633
58	0.09	0.30	0.70	0.89	0.11	12.87311	1253.774002	16139.97064
60	0.09	0.03	0.97	0.73	0.27	13.03392	1286.500183	16768.14047
62	0.09	0.37	0.63	0.67	0.33	13.32722	1325.787872	17669.06665
64	0.09	0.33	0.67	0.53	0.47	14.69689	1450.207459	21313.53949