

Título: Acelerando Aplicações de Uso Intensivo da Memória através de Processamento Perto dos Dados: Desde o Desenho Arquitetural até à Simulação de Sistemas Completos

Nome: João Miguel Morgado Pereira Vieira

Doutoramento em: Engenharia Eletrotécnica e de Computadores

Orientador: Doutor Pedro Filipe Zeferino Aidos Tomás

Co-Orientadores: Doutor Nuno Filipe Valentim Roma
Doutor Gabriel Falcão Paiva Fernandes

Resumo:

O aparecimento de novos algoritmos com uso intensivo de dados em múltiplos domínios tornou mais evidente a discrepância de desempenho entre as *Central Processing Units* (CPUs) e hierarquias de memória, resultando em baixa utilização dos recursos das CPUs. Consequentemente, os aceleradores especializados tornaram-se um meio de processamento alternativo popular, já que oferecem melhorias significativas de desempenho em relação às CPUs por dois motivos: (1) estão otimizados para um conjunto específico de aplicações, tirando máximo partido das suas características subjacentes, como padrões de acesso a dados e oportunidades de paralelização; e (2) podem ser instalados próximos da memória, tornando-se *Near-Data Accelerators* (NDAccs), concedendo-lhes uma maior largura de banda e menor latência para a memória do que a CPU.

No entanto, as arquiteturas de NDAccs dedicadas a aplicações intensivas em dados de processamento de sinal ainda são escassas na literatura. Além disso, não existem atualmente metodologias formais para desenvolver esses dispositivos sem recorrer a métodos tradicionais, como implementação em *Register Transfer Level* (RTL) e integração em *System on Chip* (SoC), que são proibitivamente complexos e demorados. Isto é particularmente verdadeiro nas fases iniciais de desenvolvimento, onde abordagens baseadas em simulação são frequentemente mais práticas e atrativas. Assim, esta tese concentra-se em duas contribuições principais: uma nova arquitetura de NDAcc adaptada a aplicações de processamento de sinal e uma cadeia de ferramentas de simulação especificamente projetada para avaliação desses dispositivos em fases iniciais de desenvolvimento.

O NDAcc proposto, o *Cache Compute System* (CCS), acelera tarefas de processamento de sinal, como *Convolutional Neural Networks* (CNNs) e algoritmos de *clustering*, ao combinar processamento próximo da cache com um dispositivo de computação especialmente otimizado para operações de *map* e *reduce*, comuns neste tipo de aplicações. O CCS, modelado e testado no simulador gem5, alcança melhorias de desempenho significativas, destacando-se um aumento de desempenho de até 40.6× em relação a uma CPU ARM Cortex-A53.

Adicionalmente, em resposta à falta de ferramentas de desenvolvimento eficazes para

NDAccs, esta tese também apresenta o NDPmulator, uma estrutura de simulação abrangente construída sobre a plataforma gem5. O NDPmulator permite modelar, simular e avaliar arquiteturas de NDAcc em cenários de sistema completo, considerando interações entre CPUs e a memória. A ferramenta simplifica o desenvolvimento ao automatizar a integração e os testes de desempenho de NDAccs, demonstrando resultados promissores tanto em configurações bare-metal quanto baseadas em *Operating System* (OS).

Com estas contribuições, o trabalho desta tese aborda limitações chave impostas pelos sistemas de processamento tradicionais em aplicações intensivas em dados, apresentando NDAccs como alternativas eficientes às CPUs e até mesmo às *Graphics Processing Units* (GPUs) para o processamento desses workloads. Adicionalmente, a estrutura de simulação proposta estabelece uma base para futuras explorações de design em arquiteturas de *Near-Data Processing* (NDP). O NDPmulator é capaz de acelerar o tempo de simulação em até 81.75× (quando comparado à simulação em RTL), apresentando uma precisão média impressionante de 95.7%.

Palavras Chave: Processamento Perto dos Dados, Desempenho Limitado pela Memória, Hierarquias de Memória de Múltiplos Níveis, *Framework* de Desenvolvimento de *Hardware*, Simulação de Sistemas Completos.

Title: Accelerating Memory-Bound Applications with Near-Data Processing:
From Architectural Design to Full-System Simulation

Abstract:

The continuous growth of data-intensive algorithms in multiple application domains has widened the performance gap between modern high-performance Central Processing Units (CPUs) and their memory hierarchies, resulting in underperforming systems and low utilization of CPU resources. As a result, an opportunity was created for specialized Near-Data Accelerators (NDAccs), which offer significant performance improvements over CPUs for two distinct reasons: (1) they are specifically optimized for given application classes, taking full advantage of their underlying characteristics, such as data access patterns and parallelization opportunities; and (2) they are installed close to where data is actually stored, granting them a much higher bandwidth and lower latency to memory than the CPU (also, spending less energy across wires).

However, NDAcc architectures targeting critical data-intensive signal-processing applications remain scarce in the literature. Moreover, no formal methodologies currently exist for developing such devices without relying on traditional methods, such as Register Transfer Level (RTL) implementation and System on Chip (SoC) integration, which are prohibitively complex and time-consuming. This is particularly true during early development stages, where simulation-based approaches are often more practical and attractive. Accordingly, this thesis focuses on two main contributions: a new NDAcc architecture tailored for signal-processing applications, and a simulation toolchain specifically designed for early-stage evaluation of these devices.

The proposed NDAcc, the Cache Compute System (CCS), accelerates signal-processing tasks, such as Convolutional Neural Networks (CNNs) and clustering algorithms, by combining near-cache processing with a compute device specially optimized for map and reduce operations, which are common in this type of applications. The CCS, modelled and tested in the gem5 simulator, achieves significant performance improvements, notably up to 40.6× speedup over an ARM Cortex-A53 CPU.

Additionally, in response to the lack of effective development tools for NDAccs, the thesis also introduces NDPmulator, a comprehensive simulation framework built on top of the gem5 platform. NDPmulator allows designers to model, simulate, and evaluate NDAcc architectures in full-system scenarios, considering CPU and memory interactions. The devised toolchain simplifies development by automating NDAcc integration and performance testing, showing promising results in both bare-metal and Operating System (OS)-based configurations.

With these contributions, the work of this thesis addresses key limitations imposed by tra-

ditional processing systems to data-intensive applications, presenting NDAccs as efficient alternatives to CPUs and even Graphics Processing Units (GPUs) to process these workloads. Additionally, the proposed simulation framework sets a foundation for future design exploration in Near-Data Processing (NDP) architectures. NDPmulator is capable of simulation time speedups of up to 81.75× (when compared to RTL simulation), while showing an impressive average accuracy of 95.7%.

Keywords: Near-Data Processing, Memory-bound, Multi-Level Memory Hierarchies, Hardware Development Framework, Full System Simulation.

Título: Acelerando Aplicações de Uso Intensivo da Memória através de
Processamento Perto dos Dados: Desde o Desenho Arquitetural
até à Simulação de Sistemas Completos

Resumo Alargado:

O surgimento de novos algoritmos com uso intensivo de dados em aplicações de vários domínios (por exemplo, processamento de imagem, bioinformática, física, finanças e inteligência artificial) tornou mais evidente a discrepância de desempenho entre as *Central Processing Units* (CPUs) de alto desempenho modernas, com vários núcleos, e as suas hierarquias de memória. De forma a tentar mitigar este problema, os subsistemas de memória tornaram-se cada vez mais complexos, levando ao consumo de uma quantidade significativa de energia nos barramentos que pode chegar a 64 % da energia global gasta pelo sistema (enquanto apenas 1 % da energia total é usada pelas unidades funcionais no processamento de dados). Além disso, apesar dos avanços tecnológicos, as tecnologias de memória atuais ainda não conseguem acompanhar o desempenho das CPUs. Este problema torna-se ainda mais evidente aquando do envolvimento de grandes quantidades de dados não estruturados, resultando na degradação do desempenho e baixa utilização da CPU.

Consequentemente, os aceleradores especializados tornaram-se um meio de processamento alternativo popular, já que oferecem melhorias significativas de desempenho em relação às CPUs por dois motivos distintos: (1) estes dispositivos são otimizados especificamente para uma determinada aplicação (ou um pequeno conjunto de aplicações), aproveitando ao máximo as suas características subjacentes, tais como padrões de acesso aos dados e oportunidades de paralelização; e (2) estes aceleradores podem ser instalados perto do local onde os dados são armazenados (ou seja, dispositivos de memória), tornando-se dispositivos de *Near-Data Processings* (NDPs), ou *Near-Data Accelerators* (NDAccs), o que lhes confere uma largura de banda muito maior e menor latência à memória do que a CPU.

O trabalho desta tese é dedicado a explorar tipologias, técnicas de design e otimizações e tecnologias de desenvolvimento direcionadas especificamente para NDAccs. Numa primeira fase, uma arquitetura de *Processing Near Memory* (PNM) otimizada para aplicações de processamento de sinais—o *Cache Compute System* (CCS)—foi projetada e prototipada utilizando o simulador arquitetural gem5. A arquitetura proposta foi especificamente desenhada para explorar o paralelismo intrínseco dos padrões de processamento *map* e *reduce*, comuns em aplicações de processamento de sinais. A unidade foi integrada com a *cache* de último nível de um sistema de processamento convencional, tirando partido da localidade dos dados e da largura de banda disponível a esse nível (podendo processar até uma linha de *cache* por ciclo de operação). Adicionalmente, por ser instalado ao nível da *cache*, o

CCS usufrui ainda dos mecanismos de coerência de dados do sistema de *cache*, garantindo sempre a obtenção dos operandos atualizados assim como a disponibilidade dos resultados produzidos para serem utilizados em operações posteriores da CPU.

Os resultados experimentais obtidos mostraram que o sistema desenvolvido é capaz de acelerar operações comumente utilizadas em *Convolutional Neural Networks* (CNNs) e aplicações de *clustering* até 40,6×, comparado com uma CPU de alto desempenho ARM Cortex-A53. Além disso, o sistema proposto também revela ser uma alternativa competitiva às *Graphics Processing Units* (GPUs) sempre que o desempenho é limitado pela largura de banda da memória.

No entanto, o processo de prototipagem do CCS revelou um problema mais profundo: praticamente não existem ferramentas de apoio ao desenvolvimento e avaliação de dispositivos de NDP. Ou seja, não existe um mecanismo padrão para implementar tais dispositivos e testá-los adequadamente enquanto integrados num sistema de processamento convencional e hierarquia de memória correspondente.

A avaliação de desempenho de NDAccs é criticamente diferente de aceleradores convencionais, visto que o comportamento dos dispositivos de memória e a cooperação com a CPU afeta significativamente o desempenho de dispositivos de NDP. Em adição, a prototipagem destes dispositivos com recurso a tecnologias clássicas—tais como o desenvolvimento *Register Transfer Level* (RTL)—é proibitivamente complexa e demorada, principalmente quando o objetivo é apenas avaliar o potencial de uma ideia ou ajustar parâmetros arquiteturais de um dispositivo ainda em desenvolvimento. Além disso, ao não integrarem os NDAccs com o restante sistema de processamento, estas metodologias de desenvolvimento clássicas podem levar à obtenção de resultados de desempenho demasiado optimistas (e irrealistas), visto que não consideram os *overheads* de controlo e sincronização existentes entre os NDAccs e a CPU e a contenção de dados gerada devido à existência de vários dispositivos a acederem em paralelo à hierarquia de memória.

Adicionalmente, a maioria das ferramentas utilizadas em trabalhos anteriores não foram projetadas com o intuito de desenvolver NDAccs. Deste modo, estes trabalhos baseiam-se sobretudo em alterações heurísticas destas ferramentas para suportar o desenvolvimento das respetivas arquiteturas de NDP, ou dependem de passos adicionais para combinar os resultados de desempenho dos NDAccs com o restante sistema (resultando na impossibilidade de emular o funcionamento simultâneo do CPU e dos NDAccs).

Em consequência, torna-se fundamental a criação de ferramentas de simulação especificamente desenhadas para dispositivos de NDP que permitam o desenvolvimento e a validação e avaliação rigorosas destes dispositivos de forma expedita sem comprometer a qualidade dos resultados. Este tema foi abordado na segunda parte desta tese de doutoramento,

sendo proposta uma plataforma de simulação destinada à investigação de soluções de NDP intitulada NDPmulator.

A solução proposta consiste numa plataforma de simulação baseada no famoso simulador arquitetural gem5 composta por três partes principais: (1) uma *framework* arquitetural que serve como base para o desenvolvimento de novas arquiteturas de NDP, lidando automaticamente com os protocolos de comunicação internos complexos do gem5 para que o utilizador se possa focar exclusivamente no desenvolvimento dos NDAccs; (2) um conjunto de *scripts* que permitem a integração dos NDAccs com o restante sistema de processamento e a simulação dos mesmos; e (3) um conjunto de ferramentas de software escritas em C que possibilitam a utilização dos NDAccs por parte de aplicações executadas na CPU, tais como bibliotecas, modelos de rotinas de processamento e *drivers*.

Adicionalmente, a solução apresentada permite simular uma aplicação isolada—modo *System Emulation* (SE)—ou um sistema convencional incluindo um *Operating System* (OS)—modo *Full System* (FS). O NDPmulator permite ainda emular a operação simultânea da CPU e dos NDAccs e é compatível com vários *Instruction Set Architectures* (ISAs), tais como x86, ARM e RISC-V.

O NDPmulator foi testado e validado exaustivamente através da simulação de NDAccs propostos em trabalhos anteriores, demonstrando ser capaz de reproduzir os resultados obtidos nas avaliações experimentais originais. Em adição, o desempenho de simulação do NDPmulator provou ser significativamente superior ao de outras ferramentas de desenvolvimento, com velocidades de simulação superiores em duas ordens de grandeza.

Tanto quanto sabemos, a solução proposta é a primeira ferramenta de simulação formal especificamente desenhada para desenvolver, validar e testar NDAccs, cobrindo não só a conceção de arquiteturas de NDP mas também a sua integração com o restante sistema de processamento e a execução em tempo real de aplicações com *kernels* de NDP.

Em resumo, esta tese de doutoramento visa explorar o potencial das técnicas de NDP como alternativa à utilização exclusiva de CPUs de uso geral para acelerar e otimizar a execução de um universo em constante crescimento de aplicações que fazem uso intensivo da memória. Em particular, propõe-se uma nova arquitetura multi-uso de PNM destinada a aplicações proeminentes de processamento de sinal, como CNNs. Além disso, é apresentado ainda um conjunto de ferramentas para apoiar o desenvolvimento desta classe de dispositivos, que ainda são escassos, visando contribuir para o *know-how* da comunidade no desenvolvimento, validação e avaliação de dispositivos de NDP, e encorajando a investigação de soluções de NDP como um paradigma alternativo de computação de alto desempenho. As valiosas contribuições do trabalho desta tese foram reconhecidas pela comunidade com a publicação de cinco artigos em conferências e revistas internacionais de

prestígio.

Palavras Chave: Processamento Perto dos Dados, Desempenho Limitado pela Memória, Hierarquias de Memória de Múltiplos Níveis, *Framework* de Desenvolvimento de *Hardware*, Simulação de Sistemas Completos.