

Parallel Computing: Unplugging to Learn

Anderson Corrêa de Lima¹, Daniel Bailo¹, Thiago Carvalho¹, José Filipe Rozeno Rodrigues¹,
Plabiany Rodrigo Acosta¹, Wellington Marques de Aquino¹

¹*Campus de Ponta Porã (CPPP)*

Universidade Federal de Mato Grosso do Sul

Ponta Porã, Brasil

anderson.correa.lima@gmail.com, danbailoufms@gmail.com, thiago.carvalho@aluno.ufms.br,
josefiliperdrigues@gmail.com, plabiany@gmail.com, w_linton@hotmail.com

Abstract—Technological devices programmed through parallel computing are increasingly present in people’s daily lives. They are responsible for processing images that we see on computers, televisions, cell phones and games. It is a fact that parallel processing is a reality, so computer professionals need to know the peculiarities of this form of programming. However, teaching parallel computing is not a simple task, the computational thinking parallel is quite different from that for the sequential processing. This work, then proposes a new methodology for teaching parallel computing for young beginners in programming. Our technique is based on principles of unplugged computing, like to learn doing and playing. We believe that unplugged computing can be very useful in teaching the difficult concepts of parallel computing¹.

Index Terms—computing, parallelism, high-performance, challenges, teaching.

I. INTRODUÇÃO

Desde a metade do século passado o mundo tem experimentado o impacto e as consequências da terceira revolução industrial: a revolução da Tecnologia da Informação. Desde então, o novo universo proporcionado pela computação foi sendo incorporado não só no meio acadêmico e científico, como também na vida pessoal dos indivíduos de nossa sociedade.

Porém, apesar de tanto envolvimento, o conhecimento do que a computação pode oferecer de melhor é bem limitado, mesmo dentro do ambiente educacional. De acordo com Vieira [1], habitualmente, as escolas de ensino básico propiciam o contato com a informática por meio da exposição de alguns conceitos iniciais e enfatizam atividades práticas, as quais requerem uso dos computadores em laboratórios de informática. Já no ensino superior ou técnico nas áreas de computação, a proposta curricular é geralmente direcionada para o ensino de habilidades com foco na tradicional programação sequencial.

No entanto, o ensino apenas de programação sequencial não condiz mais com a realidade dos níveis tecnológicos existentes na computação hoje, principalmente após 2003, quando surgiu um novo paradigma para computação que se baseou na utilização de mais de uma unidade de processamento, conhecidas como núcleos (*cores*), em cada chip, com o propósito de aumentar o poder de processamento [2]. Era o início da computação paralela com alto desempenho,

que catapultou a capacidade de processamento de diversos dispositivos tecnológicos, muitos deles ubíquos [2].

A Computação Paralela e Distribuída (*Parallel and Distributed Computing* - PDC) de Alto Desempenho já faz parte do dia-a-dia de muitas pessoas, mesmo que elas não saibam. Ela está presente no processamento de seus *smartphones*, *websites* de redes sociais e muito mais. Tornou-se imperativo então transmitir e ensinar as particularidades do pensamento computacional paralelo em vários níveis da estrutura educacional [3]. Isto é primordial e urgente, visto que o advento da computação paralela de alto desempenho parece não admitir retrocesso. Entretanto, ensinar o pensamento computacional “em paralelo”, principalmente para novos programadores, não é uma tarefa simples. Muitas habilidades precisam ser treinadas. Na educação em computação, seja ela a nível técnico, tecnológico ou superior, este parece ser um grande desafio. Certamente, não é mais suficiente que os futuros programadores adquiram somente habilidades da programação sequencial convencional [4].

É possível encontrar várias iniciativas fora do Brasil buscando incentivar que os conceitos da computação paralela tenham um maior alcance na sociedade acadêmica e tecnológica. Um exemplo dessas iniciativas foi a proposta apresentada pelo Comitê Técnico do IEEE sobre Processamento Paralelo (TCPP), junto com a *National Science Foundation* (NSF) [4]. A proposta descreve uma nova estrutura curricular para os cursos de Ciência da Computação e Engenharia da Computação, de forma que os graduandos possam adquirir um certo nível de habilidades no estudo de computação paralela. Aprender as técnicas e tecnologias da computação paralela tem se tornado um desafio tanto para os profissionais já na área de desenvolvimento, como também para os que agora se iniciam no estudo da computação, seja ela no ensino técnico ou superior [5].

Tendo em vista a necessidade crescente de pessoas habilitadas em técnicas de computação paralela, tanto no ambiente acadêmico quanto no empresarial, nós realizamos uma breve pesquisa nas ementas curriculares de instituições de ensino técnico federal do Brasil (Institutos Federais de Educação, Ciência e Tecnologia), para verificar se haviam disciplinas voltadas para o desenvolvimento do pensamento computacional paralelo. Procurou-se selecionar, no mínimo, uma instituição

¹Trabalho em andamento como descrito na chamada desta conferência.

federal por região do Brasil. Entretanto, das grades curriculares de oito instituições pesquisadas, nenhuma oferece ou descreve qualquer disciplina relacionada ao ensino de computação paralela, mesmo com a urgência do aprendizado desta.

Devido a diversidade de técnicas e tecnologias direcionadas à computação paralela, que vem se ampliando dentro de universidades estrangeiras renomadas e do mercado de trabalho, e percebendo também o estado de carência que o ensino brasileiro voltado a tecnologia da informação possui em relação ao paralelismo, este trabalho propõe uma nova forma de ensinar computação paralela. Para desmistificar conceitos que podem parecer de difícil aprendizado, nós iremos fazer uso da teoria da computação desplugada e de protótipos reais desenvolvidos a partir de garrafas pet. É nosso objetivo oferecer oficinas sobre computação paralela por meio da computação desplugada, para instituições públicas de ensino médio com técnico em informática. Pretendemos, com isso, despertar e motivar o pensamento computacional paralelo em futuros profissionais de tecnologia da informação no Brasil.

II. FUNDAMENTAÇÃO TEÓRICA

Esta seção descreve a fundamentação teórica para este trabalho. São discutidos brevemente os seguintes temas: Educação em computação com ênfase para programação. Definição de computação paralela. Desafios no ensino de computação paralela e por fim discute-se sobre a técnica de computação desplugada.

A. Educação em computação: programação

Alguns países vêm reconhecendo a necessidade de atualizar os seus sistemas educacionais no que se refere à educação em computação. Nos Estados Unidos, o modelo que está sendo experimentado chama atenção para conteúdos que permitem o alcance da chamada Educação Imperativa, que é aquela onde mais importante do que aprender temas ligados à tecnologia está a capacidade de desenvolver nos estudantes o pensamento computacional e a sua autonomia para a resolução de problemas [6]. Na academia, muitas revisões da literatura evidenciam a necessidade de ampliar os esforços dos pesquisadores no que se refere ao conhecimento que se possui sobre o porquê de ensinar computação nas escolas, de como os conteúdos deveriam ser ensinados, que tópicos deveriam ser lecionados e para quem esta educação poderia ser significativa [6]. Particularmente, estes temas ainda geram muitas discussões, principalmente quando consideramos a constante evolução tecnológica e seus reflexos no ensino de computação.

As disciplinas de computação e informática são compostas de diferentes e interligadas áreas de conhecimento desenvolvendo diferentes competências, habilidades e conhecimentos tanto na área básica quanto na área tecnológica. Entre as competências mais difíceis de serem desenvolvidas estão as relacionadas com o desenvolvimento de algoritmos e programas [7]. Esta é uma forte razão para incluir tal temática no ensino médio – assim, os futuros alunos dos cursos de computação teriam menos dificuldade nessa área. E aqueles que se dirigissem para outros ramos profissionais teriam desenvolvido

competências de resolução de problemas e formalização dos mesmos que seriam úteis em suas respectivas áreas do conhecimento [7].

B. Definindo um computador paralelo

Segundo Ian Foster, um computador paralelo é um conjunto de processadores interconectados, capazes de trabalhar de forma simultânea e cooperativa a fim de resolver um mesmo problema computacional [8]. Essa definição é suficientemente ampla para incluir supercomputadores paralelos, que englobam centenas ou milhares de computadores, redes locais de computadores e estações de trabalho com múltiplos processadores. A ideia principal da computação paralela consiste em executar tarefas simultaneamente.

C. Ensinar: Um desafio para a computação paralela

Na última década, a criação de microprocessadores com múltiplos núcleos tornou-se uma tendência, principalmente quando a indústria percebeu que o aumento da frequência de processamento estava criando mais problemas do que resolvendo, devidos entre outros fatores aos problemas de superaquecimento dos chips com poucos *cores* [2]. Seguindo esta tendência computadores em um contexto geral passaram a suportar múltiplas *threads* (*multithreading*) ou múltiplos núcleos (*multi/many core*). Hoje em dia, os hardwares paralelos compostos de múltiplos processadores, já estão presentes nos principais equipamentos ou plataformas computacionais. Estações de trabalho, servidores, supercomputadores ou até mesmo sistemas embarcados já fazem uso desta arquitetura [2].

A presença de mais processadores aumentou drasticamente o poder computacional de diversos dispositivos e isto possibilitou o tratamento de problemas, que há bem pouco tempo eram impraticáveis. Novas abordagens paralelas, proporcionadas pelas novas arquiteturas de múltiplos núcleos, ofereceram então novas propostas para solução de antigos problemas. O desafio de explorar estas novas abordagens sinaliza ser bastante interessante, dada a possibilidade de ganho de tempo, porém, antes de qualquer jornada neste caminho, é necessário compreender e utilizar as novas arquiteturas de multiprocessamento da melhor maneira possível. Os obstáculos se iniciam precocemente. Pensar em paralelo, por si só exige maior esforço do programador. É preciso tratar de problemas de concorrência e sincronização, para os quais, a depuração é um processo bastante custoso e as vezes manual. Logo, percebe-se que mesmo com as arquiteturas *multi/many core* sendo uma realidade, a programação paralela continua a ser uma tarefa complexa. Além da dependência e disponibilidade de ferramentas e ambientes de programação adequados para memória compartilhada ou distribuída, enfrenta-se uma série de problemas não existentes em programação sequencial: código paralelo, concorrência, comunicação, sincronização, granularidade e balanceamento de carga, estão entre eles. Ademais, podemos enumerar uma série de outros questionamentos a cerca da migração de uma solução sequencial para uma solução em arquitetura paralela *multi/many core* [2].

Como visto, são diversos os obstáculos, mas se não forem criados programas que façam uso dos recursos da computação paralela multi/*many core*, o desperdício de processamento e de energia continuará [2], entretanto uma outra questão ainda mais relevante é: **Visto que a computação paralela de alto desempenho é bastante complexa, mas tornou-se imprescindível, qual a melhor forma de ensinar seus conceitos e técnicas aos futuros programadores?** A resposta para esta pergunta não é simples, pois desenvolver *software* baseado em computação paralela é uma tarefa de programação árdua de considerável esforço intelectual. Além disso, adquirir dispositivos com milhares de núcleos para o ensino de programação paralela é financeiramente custoso.

D. A Computação Desplugada

O termo Computação Desplugada tornou-se conhecido por meio da publicação do livro “*Computer Science Unplugged-off-line activities and games for all ages*”, em 1998, dos autores Tim Bell, Ian H. Witten e Mike Fellows [9]. O livro apresenta uma compilação de várias atividades e dinâmicas que apresentam, de forma lúdica, conceitos matemáticos e computacionais como números binários, padrões e ordenamento, criptografia, dentre outros.

A computação desplugada pode ser vista como uma democratização do ensino da computação, pois, algumas de suas vantagens abrangem a possibilidade de se ensinar a todas as idades [1]; com independência dos recursos de *hardware* e *software*, o que gera a possibilidade do ensino em áreas remotas e sem acesso a tecnologia; a viabilidade de ser aplicada por não especialistas na área; dentre outras [10].

Com a definição da técnica, diversos trabalhos no ensino vem sendo realizados por meio da computação desplugada. Um exemplo foi o trabalho descrito pelo artigo “Um Relato de Experiência do Uso da Técnica Computação Desplugada” [1], no qual certos conceitos de computação foram ensinados a crianças e adolescentes do ensino fundamental e médio, por meio de peças teatrais lúdicas.

III. PROPOSTA E METODOLOGIA DO TRABALHO

A proposta deste trabalho consiste em apresentar uma nova forma de ensino de tópicos da disciplina de computação paralela, por meio de técnicas de computação desplugada, de maneira mais simples, divertida e lúdica, para programadores iniciantes. Especificamente, por sua simplicidade, o primeiro tópico escolhido é o algoritmo da soma em paralelo [11].

Para alcançar o objetivo nossa metodologia foi dividida em algumas etapas. É importante ressaltar que como se trata de um trabalho em andamento, com resultados parciais, algumas etapas ainda não foram concluídas. As etapas são: Pesquisa na literatura sobre Educação em Computação voltada para a disciplina de computação paralela (concluída). Elaboração de uma oficina denominada: “Despertando para a Computação Paralela” (em andamento). Realização da oficina em Instituições de Ensino Técnico em Informática, que são parceiras do projeto (próxima etapa). Coleta e análise de dados

advindos das oficinas. Elaboração de artigos completos a partir dos dados obtidos com a pesquisa.

O formato de nossa oficina partiu de nossa constatação de que o ensino de programação sequencial, muitas vezes em compiladores linux, nesta era da internet das coisas é desafiador, logo, o ensino de programação paralela, pode ser ainda mais complexo. Este é um fato importante e para tratá-lo, nós sugerimos uma alternativa baseada na utilização da computação desplugada, uma técnica difundida mundialmente, que consiste em ensinar os fundamentos da computação, por meio de atividades, sem o uso do computador [9] [1].

O público-alvo da oficina será composto por alunos de cursos técnicos em informática de unidades do Instituto Federal, que é parceiro neste trabalho. A oficina foi idealizada em duas fases: A fase 1 (um) consiste de uma aula expositiva e dialogada, na qual serão apresentados conceitos importantes sobre a computação paralela, destacando seu início e suas possibilidades nos dias atuais, tanto na academia, quanto no mercado. Em seguida, serão apresentados aos alunos dispositivos de *hardware*, que permitem o processamento paralelo de alto desempenho, por exemplo uma GPU (*Graphics Processing Unit*), as conhecidas unidades de processamento gráfico. Logo após realizaremos uma comparação em sala, dos já antigos dispositivos de processadores únicos (*single-core*) com as modernas GPU's, discutindo as diferenças entre eles. Na oficina os alunos poderão tocar e reconhecer os dispositivos. A metodologia da fase 2 (dois) é prática. Nós vamos discutir e apresentar particularidades da programação paralela, por meio de um protótipo físico construído a partir de garrafas de PET grandes e médias, bolinhas de isopor, pranchas de isopor, tintas guache, fitas coloridas e outros itens mais acessíveis e com baixo custo. O protótipo simula a execução de um dos primeiros e principais algoritmos para o ensino inicial de computação paralela: o algoritmo da soma de um vetor numérico em paralelo, descrito em detalhes por Joseph JáJá [11]. A Figura 1 apresenta o protótipo que construímos.



Figura 1. Maquete do algoritmo da soma em paralelo.

A partir de nosso protótipo diversos conceitos importantes da programação paralela podem ser ensinados, entre eles: a quantidade e a identificação de processos para uma ta-

refa, a divisão da carga de trabalho, a computação local, a computação global, a sincronização de dados, a comunicação entre processos, etc. É importante ressaltar que estes tópicos não são de fácil compreensão, acreditamos porém que uma atividade prática envolvendo os alunos, como a utilização do protótipo, pode ensinar estes conceitos de forma mais natural. Ações assim fazem parte de princípios da computação desplugada como o “aprender fazendo de forma divertida” [9] [1].

Em nossa maquete (protótipo), cada garrafa PET representa um processo durante a execução de um algoritmo, no caso o algoritmo da soma local de valores inteiros. No protótipo, o algoritmo paralelo possui três níveis de execução para as operações de soma globais.

Inicialmente os alunos serão convidados para se posicionarem frente a uma garrafa que esteja no nível mais alto da estrutura (nível um). Um instrutor, representando um computador *single-core* irá distribuir igualmente bolinhas de cor laranja para cada um dos alunos, que estão em frente às garrafas. As bolinhas identificam valores de um vetor numérico. Além disso, o instrutor identifica-rá cada aluno com um cordão contendo um papel com o texto: **Processo** P_x , onde x é o número do processo. Em seguida, os alunos inserem as bolinhas nas garrafas PET, que representam os processos. Cada cor de bolinha representa um valor determinado. As garrafas contém três divisórias. Cada aluno irá inserir duas bolinhas de cor laranja e retirar a primeira divisória da garrafa. Logo após, juntos eles retiram a terceira divisória das garrafas azuis (nível um), delas fluem bolinhas azuis, já previamente armazenadas, representando a soma das bolinhas de cor laranja. A Figura 2 representa visões de uma garrafa-processo e suas três divisórias. A partir destas ações, conceitos importantes de programação paralela já podem ser discutidos, tais como: o número de processos escolhidos, a identificação de processos, a divisão de carga de trabalho, a computação local e a sincronização de tarefas.

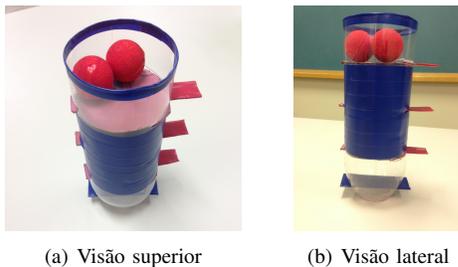


Figura 2. Garrafa-processo e suas três divisórias.

Ao observar o fluxo de bolinhas azuis deslizando pelas canaletas para as garrafas do nível inferior, alguns outros conceitos podem ser discutidos: a comunicação entre processos, a diminuição do número de processos ativos e a explicação do que é um passo de computação em um algoritmo paralelo. Além disso, por meio da maquete, os alunos passarão a perceber que o número de garrafas-processos no nível dois é menor do que do nível um (metade), com isso, alguns processos

não serão mais ativos. Os alunos que representarão processos inativos serão então convidados a deixar sua posição. As ações de manipulação das divisórias são então repetidas pelos alunos-processos restantes até que apenas um aluno-processo, que representa a última garrafa esteja sozinho. Neste instante o último aluno pode calcular o valor da soma, de acordo com as bolinhas e suas cores e dizer ao instrutor o valor final obtido.

IV. CONCLUSÕES

Este trabalho apresenta uma proposta para o ensino de computação paralela, principalmente para programadores iniciantes, baseada em princípios de computação desplugada. Nós acreditamos que a estratégia de utilização de nossa maquete física pode contribuir para desmistificar muitos conceitos da programação paralela. Nosso público-alvo será formado por alunos de cursos de informática de escolas técnicas públicas. Como trabalhos futuros, além da oficina, propomos um curso básico de programação paralela utilizando uma linguagem real, como o *Python*. *Python* é interessante, pois suporta múltiplos paradigmas de programação e é de fácil aprendizado. Acreditamos por fim, que este trabalho pode ser ampliado e utilizado para despertar e estimular o ensino de programação paralela em sala de aula. Gostaríamos de agradecer ao suporte financeiro e estrutural oferecido pela Universidade Federal de Mato do Sul e ao programa PET Fronteira do Campus de Ponta Porã da UFMS para realização deste trabalho.

REFERÊNCIAS

- [1] A. Vieira, O. Passos, and R. Barreto, “Um relato de experiência do uso da técnica computação desplugada,” *Anais do XXI WEI*, pp. 670–679, 2013.
- [2] D. Kirk and W.-M. Hwu, *Programando para processadores paralelos: uma abordagem prática à programação de GPU*. Elsevier Brasil, 2010.
- [3] S. Prasad, A. Gupta, K. Kant, A. Lumsdaine, D. Padua, Y. Robert, A. Rosenberg, A. Sussman, and C. Weems, “Literacy for all in parallel and distributed computing: guidelines for an undergraduate core curriculum,” *The New Zealand Journal of Applied Computing and Information Technology*, 01 2013.
- [4] S. K. Prasad, A. Y. Chtchelkanova, S. K. Das, F. Dehne, M. G. Gouda, A. Gupta, J. JáJá, K. Kant, A. L. Salle, R. LeBlanc, M. Lumsdaine, D. A. Padua, M. Parashar, V. K. Prasanna, Y. Robert, A. L. Rosenberg, S. Sahni, B. Shirazi, A. Sussman, C. C. Weems, and J. Wu, “Nsf/ieee-tcpp curriculum initiative on parallel and distributed computing: core topics for undergraduates,” in *SIGCSE*, 2011.
- [5] A. E. A. M. . M. J. C. S. A. S. R. Cesar, Eduardo; Cortés, “Introducing computational thinking, parallel programming and performance engineering in interdisciplinary studies,” *Journal of Parallel and Distributed Computing*, pp. 116–126, 01 2017.
- [6] A. C. Cambraia and P. D. Scaico, “Os desafios da educação em computação no brasil: um relato de experiências com projetos pibid no sul e nordeste do país,” *Revista Espaço Acadêmico*, vol. 13, no. 148, pp. 01–09, 2013.
- [7] J. Júnior, C. E. Rapkiewicz, C. Delgado, and J. A. M. Xexeo, “Ensino de algoritmos e programação: uma experiência no nível médio,” in *XIII Workshop de Educação em Computação (WEI 2005)*. São Leopoldo, RS, Brasil, 2005.
- [8] I. Foster, *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995.
- [9] T. C. Bell, I. H. Witten, and M. Fellows, *Computer Science Unplugged: Off-line activities and games for all ages*. Citeseer, 1998.
- [10] T. Bell, I. H. Witten, M. Fellows, R. Adams, and J. McKenzie, “Ensinando ciência da computação sem o uso do computador,” *Computer Science Unplugged ORG*, 2011.
- [11] J. JáJá, *An introduction to parallel algorithms*. Addison-Wesley Reading, 1992, vol. 17.