

Distribuidor de Tarefas para uma plataforma de Computação Voluntária utilizando WebAssembly

Miguel F. S. Vasconcelos , Daniel Cordeiro

¹Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)
São Paulo – SP – Brazil

{miguel.vasconcelos, daniel.cordeiro}@usp.br

Abstract. *Volunteer Computing has profoundly changed scientific computing by allowing the use of idle computing resources from collaborators geographically dispersed around the world. Nevertheless, the most popular platforms require the installation of specific software on the collaborator’s machine to obtain computational resources, which can hinder its adoption by new volunteers. In order to simplify the use of Volunteer Computing, this work proposes the use of software compiled for WebAssembly — a new assembly-like language that can be executed by modern browsers — and a Web application to support the distribution and execution of tasks, as well the validation and retrieval of results.*

Resumo. *A Computação Voluntária revolucionou a computação científica ao permitir o uso de recursos computacionais ociosos provenientes de colaboradores geograficamente espalhados pelo mundo. Entretanto, as plataformas mais populares requerem a instalação de softwares específicos na máquina do colaborador para, então, obter recursos computacionais, o que pode gerar um obstáculo na adesão de novos voluntários. Este trabalho propõe o uso de software compilado para WebAssembly — um novo formato de linguagem de montagem que pode ser executado por navegadores modernos — e de uma aplicação Web para a distribuição, execução de tarefas, validação e recuperação dos resultados como forma de simplificar o uso de Computação Voluntária.*

1. Introdução

A computação voluntária é um tipo de computação distribuída, cujos elementos computacionais são constituídos por dispositivos de colaboradores que doam recursos ociosos, em termos de poder de processamento e armazenamento, para serem utilizados na execução de aplicações, que, em grande parte, possuem fins científicos. Essa abordagem mostrou-se uma forma eficiente para conseguir recursos computacionais. No ano de 2015, BOINC (Berkeley Open Infrastructure for Network Computing) possuía poder de processamento computacional superior ao supercomputador classificado na primeira posição do ranking TOP 500 [Chorazyk et al. 2017].

No que se refere ao processo de adesão dos usuários às plataformas de computação voluntária, muitas dessas plataformas necessitam a instalação de determinados softwares, como o Folding@Home, uma plataforma que realiza simulações biofísicas para investigar doenças como Alzheimer, Parkinson e outras [Larson et al. 2009]. Essa restrição poderia impor um obstáculo na adesão de novos voluntários.

Uma abordagem que pode facilitar o processo de adesão de novos colaboradores é o desenvolvimento de plataformas web de computação voluntária, em que bastaria ao usuário acessar a página da plataforma para doar seus recursos computacionais. Entretanto, o desempenho de tecnologias web é inferior a de softwares que são executados diretamente nas máquinas dos voluntários. Nesse contexto, o uso de WebAssembly é uma estratégia a ser considerada.

O WebAssembly é uma tecnologia que visa ser “um compilador para a web”, em outras palavras, as aplicações serão executadas em velocidade quase nativa diretamente nos navegadores web, possuindo grande interoperabilidade e portabilidade [Mozilla 2017]. Embora ainda esteja em desenvolvimento, a tecnologia já foi incorporada ao W3C, significando que será adotada como padrão por todos os navegadores de internet [W3C 2017]. Atualmente, é possível gerar binários WebAssembly através da compilação de programas nas linguagens de programação *JavaScript*, *C*, *C++* e *Rust* e os principais navegadores de internet já oferecem suporte a tecnologia, como é possível observar na Tabela 1.

Tabela 1. Suporte para WebAssembly nos navegadores de internet

Navegador	Versão	Mês de Lançamento
<i>Chrome</i>	57	03/2017
<i>Firefox</i>	52	03/2017
<i>Edge</i>	16	10/2017
<i>Safari</i>	11	09/2017
<i>Safari para iOS</i>	11	09/2017
<i>Chrome para Android</i>	62	10/2017

O presente trabalho propõe a adoção dessa estratégia no desenvolvimento de um protótipo de sistema distribuidor de tarefas, um dos principais componentes para plataformas de computação voluntária, ao qual serão submetidas aplicações em WebAssembly para validar se a tecnologia atende os principais requisitos de computação voluntária, visando usufruir os aspectos de interoperabilidade, portabilidade e desempenho que o WebAssembly apresenta, assim como executar as aplicações diretamente nos navegadores de internet torna mais prático a adesão de novos colaboradores.

2. Trabalhos Relacionados

Bayanihan é uma plataforma web de computação voluntária desenvolvida em JAVA, cuja denominação deriva de um termo filipino referente a uma tradição que envolve a cooperação e união de indivíduos para resolver problemas complexos [Sarmenta and Hirano 1999]. O projeto visava adotar essa filosofia na computação, através do desenvolvimento de uma plataforma WEB que possibilitasse indivíduos que não necessariamente tivessem alto nível de conhecimento técnico contribuir com recursos computacionais para projetos científicos [Sarmenta and Hirano 1999].

O CrowdCL é outro exemplo de plataforma web de computação voluntária. Essa plataforma é *open source* e utiliza a tecnologia OpenCL de placas de vídeo para obter recursos computacionais. Os projetos que adotam a plataforma executam códigos na linguagem JavaScript e utilizam da API WebCL — versão WEB da tecnologia OpenCL [MacWilliam and Cecka 2013].

Outro exemplo de esforços para produzir uma plataforma web de computação voluntária foi a plataforma *open source* que utiliza *Web Workers* [Chorazyk et al. 2017]. As aplicações submetidas a plataforma devem ser desenvolvidas na linguagem de programação JavaScript e, conforme Web Workers são executados em plano de fundo, o colaborador pode continuar a usar o navegador de internet [Chorazyk et al. 2017].

3. Metodologia

O desenvolvimento deste trabalho, consistiu, basicamente, de duas partes: integração da tecnologia WebAssembly com o arcabouço de desenvolvimento Ruby on Rails; e desenvolvimento da arquitetura do sistema distribuidor de tarefas. Em relação a primeira parte, realizamos levantamento de informações em páginas de fóruns na internet, páginas oficiais com as documentações das tecnologias, páginas de eventos e conferências de tecnologia e repositórios de códigos, visando encontrar exemplo de aplicações desenvolvidas em WebAssembly e exemplos de integração com arcabouços de desenvolvimento web, em especial o Ruby on Rails, para desenvolver o sistema da maneira mais elegante.

No que se refere à arquitetura do sistema, nos baseamos em componentes da arquitetura do BOINC, visto que atualmente é a plataforma com maior número de dispositivos de colaboradores — mais de 4 milhões [BOINCstats 2018], principalmente no mecanismo de validação de dados, denominado *persistent redundant computing* [Anderson et al. 2005].

Após realizar um levantamento sobre as funcionalidades básicas de plataformas de computação voluntária, identificamos os seguintes requisitos que foram utilizados no desenvolvimento do sistema: submeter aplicações ao sistema; disponibilizar essas aplicações para possíveis colaboradores; enviar e receber dados para as aplicações; e validar os resultados.

4. Resultados

Como resultado da realização deste trabalho, foi desenvolvido um protótipo de sistema distribuidor de tarefas hospedado em determinada página WEB que permite a submissão de aplicações escritas nas linguagens de programação C/C++ e compiladas para WebAssembly, assim como arquivos relacionados a dados que deverão ser processados. O protótipo também apresenta um mecanismo simples para consulta e validação de resultados do processamento das aplicações.

O código produzido no desenvolvimento deste sistema está disponível publicamente em <https://github.com/migvasc/vcwasm>. Além do código, foi disponibilizado um modelo de código de aplicação em C/C++ que pode ser utilizado com a plataforma para que os pesquisadores possam ter uma diretriz sobre como desenvolver aplicações compatíveis com o sistema distribuidor de tarefas desenvolvido.

5. Conclusão

Este trabalho propõe um sistema distribuidor de tarefas que possui requisitos mínimos para ser adotado por plataformas de computação voluntária. O sistema utiliza a arquitetura WEB e as aplicações que são submetidas ao sistema devem estar compiladas em WebAssembly. Esses requisitos podem facilitar o processo de adesão de novos voluntários

na plataforma, visto que é executada diretamente em uma página WEB, apresenta grande portabilidade e interoperabilidade. O fato de as aplicações estarem compiladas em WebAssembly também garante um desempenho próximo ao nível nativo.

O WebAssembly é uma tecnologia muito nova que acaba de ser incorporada pelos navegadores web mais modernos. A falta de documentação ainda atrapalha sua adoção e utilização em sistemas web. Para auxiliar na superação desta dificuldade na utilização do sistema desenvolvido neste trabalho, existe um modelo de aplicação no repositório da ferramenta de versionamento de código utilizada. Acreditamos que essa dificuldade será sanada com o tempo. O suporte oferecido pelos principais navegadores, conforme apresentado na Tabela 1, também fomentará uma maior disseminação da tecnologia e, juntamente com a incorporação ao W3C, ressaltam o grande potencial do WebAssembly, que permitirá que aplicações dos mais diversos tipos sejam executadas diretamente nos navegadores de internet com desempenho próximo ao nativo.

Por fim, dentre os possíveis trabalhos futuros relacionados ao projeto é possível listar: i) integração com a API IndexedDB para criação de *checkpoints*; ii) aperfeiçoamento do mecanismo de validação de dados; iii) avaliação de desempenho do sistema; e iv) desenvolvimento de mecanismos que atendam aos quesitos de segurança para evitar execução de aplicações maliciosas.

Referências

- Anderson, D. P., Korpela, E., and Walton, R. (2005). High-performance task distribution for volunteer computing. In *e-Science and Grid Computing, 2005. First International Conference on*, pages 8–pp. IEEE.
- BOINCstats (2018). Boincstats. Disponível em: <https://boincstats.com/>. Acessado em: 15 de Fevereiro de 2018.
- Chorazyk, P., Godzik, M., Pietak, K., Turek, W., Kisiel-Dorohinicki, M., and Byrski, A. (2017). Lightweight volunteer computing platform using web workers. *Procedia Computer Science*, 108:948–957.
- Larson, S. M., Snow, C. D., Shirts, M., and Pande, V. S. (2009). Folding@ home and genome@ home: Using distributed computing to tackle previously intractable problems in computational biology. *arXiv preprint arXiv:0901.0866*.
- MacWilliam, T. and Cecka, C. (2013). Crowdcl: Web-based volunteer computing with webcl. In *High Performance Extreme Computing Conference (HPEC), 2013 IEEE*, pages 1–6. IEEE.
- Mozilla (2017). Webassembly. Disponível em: <https://developer.mozilla.org/pt-BR/docs/WebAssembly>. Acessado em: 15 de Fevereiro de 2018.
- Sarmenta, L. F. and Hirano, S. (1999). Bayanihan: Building and studying web-based volunteer computing systems using java. *Future Generation Computer Systems*, 15(5):675–686.
- W3C (2017). Webassembly. Disponível em: <https://www.w3.org/community/webassembly/>. Acessado em: 15 de Fevereiro de 2018.