



INSTITUTO FEDERAL Brasília

Instituto Federal de Brasília

Campus Brasília

Curso Superior de Tecnologia em Sistemas para Internet

A STAR DE ALTO DESEMPENHO NA WEB

Por

CAIO GOMES FLAUSINO

Tecnólogo

BRASÍLIA
2021

Caio Gomes Flausino

A STAR DE ALTO DESEMPENHO NA WEB

Trabalho apresentado ao Programa de Curso Superior de Tecnologia em Sistemas para Internet da Instituto Federal de Brasília como requisito parcial para obtenção do grau de Tecnólogo em Sistemas de Internet .

Orientador: Dr. Daniel Sundfeld Lima

Co-Orientador: MSc. Diego Cesar Florencio de Queiroz

BRASÍLIA

2021

Caio Gomes Flausino

A Star de alto desempenho na web/ Caio Gomes Flausino. – BRASÍLIA, 2021-
43 p. : il. (algumas color.) ; 30 cm.

Orientador Dr. Daniel Sundfeld Lima

Tecnólogo – Instituto Federal de Brasília, 2021.

1. Palavra-chave1. 2. Palavra-chave2. I. Orientador. II. Universidade xxx. III.
Faculdade de xxx. IV. Título

CDU 004

Caio Gomes Flausino

A Star de alto desempenho na web

Trabalho de conclusão de curso de graduação apresentado a Coordenação do Curso Superior de Tecnologia em Sistemas de Internet do Instituto Federal de Brasília – Campus Brasília, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas de Internet.

Aprovado em: _____ de _____ de _____.

BANCA EXAMINADORA

Prof. Dr. Daniel Sundfeld Lima
Computação/IFB

Prof.^a Dr.^a Primeira Membro da Banca
Computação/IFB

Prof. Dr. Segundo Membro da Banca
Computação/IFB

Prof.^a Dr.^a Terceira Membro da Banca
Computação/IFB

BRASÍLIA
2021

Dedico este trabalho à minha família.

Agradecimentos

Agradeço ao meu orientador Prof. Dr. Nome do Orientador, pela sabedoria com que me guiou nesta trajetória.

Aos meus colegas de sala.

A Secretaria do Curso, pela cooperação.

Gostaria de deixar registrado também, o meu reconhecimento à minha família, pois acredito que sem o apoio deles seria muito difícil vencer esse desafio.

Enfim, a todos os que por algum motivo contribuíram para a realização desta pesquisa.

Elemento opcional.
*Espaço destinado à epígrafe (elemento opcional). Nesta folha, o autor usa
uma citação, seguida de indicação de autoria e ano, relacionada com a
matéria tratada no corpo do trabalho.*
—NOME DO AUTOR

Resumo

SOBRENOME, Prenome do Autor do Trabalho. Título do trabalho: subtítulo (se houver). 2018. 65 f. Trabalho de Conclusão de Curso (Graduação) – Tecnólogo em Sistemas para Internet. Instituto Federal de Brasília – Campus Brasília. Brasília/DF, 2018.

Elemento obrigatório, constituído de uma sequência de frases concisas e objetivas, fornecendo uma visão rápida e clara do conteúdo do estudo. O texto deverá conter no máximo 500 palavras e ser antecedido pela referência do estudo, com exceção do resumo inserido no próprio documento. Também, não deve conter citações. O resumo deve ser redigido em parágrafo único, espaçamento simples e seguido das palavras representativas do conteúdo do estudo, isto é, palavras-chave, em número de três a cinco, separadas entre si por ponto e finalizadas também por ponto. Usar o verbo na terceira pessoa do singular, com linguagem impessoal (pronome SE), bem como fazer uso, preferencialmente, da voz ativa.

Palavras-chave: Primeira palavra. Segunda palavra. Terceira palavra. Quarta palavra. Quinta-palavra.

Abstract

SOBRENOME, Prenome do Autor do Trabalho. Título do trabalho: subtítulo (se houver). 2018. 65 f. Trabalho de Conclusão de Curso (Graduação) – Tecnólogo em Sistemas para Internet. Instituto Federal de Brasília – Campus Brasília. Brasília/DF, 2018.

Elemento obrigatório, constituído de uma sequência de frases concisas e objetivas, fornecendo uma visão rápida e clara do conteúdo do estudo. O texto deverá conter no máximo 500 palavras e ser antecedido pela referência do estudo, com exceção do resumo inserido no próprio documento. Também, não deve conter citações. O resumo deve ser redigido em parágrafo único, espaçamento simples e seguido das palavras representativas do conteúdo do estudo, isto é, palavras-chave, em número de três a cinco, separadas entre si por ponto e finalizadas também por ponto. Usar o verbo na terceira pessoa do singular, com linguagem impessoal (pronome SE), bem como fazer uso, preferencialmente, da voz ativa.

Keywords: Keyword. Second keyword. Third keyword. Keyword.

Lista de Figuras

Lista de Algoritmos

Lista de Tabelas

A.1	List of conferences on which the searches were performed.	41
A.2	List of journals in which the searches were performed.	42
A.3	Search string per Search Engine.	43

Lista de Acrônimos

Sumário

1	Introdução	25
1.1	Tema	26
1.2	Problema	26
1.2.1	Objetivo geral	26
1.2.2	Objetivos específicos	26
1.3	Estrutura do TCC	26
1.3.1	Classificação da Pesquisa	27
2	Conceitos gerais e revisão da literatura	29
2.1	Tecnologias principais desse trabalho	29
2.1.1	Linguagem de programação C++	29
2.1.2	Contêineres Linux	29
2.1.3	Kubernetes	29
2.2	Trabalhos relacionados	29
3	Metodologia	31
3.1	Uma seção	31
3.2	Uma outra seção	31
4	Apresentação e Análise dos Resultados	33
5	Conclusões e Trabalhos Futuros	35
	Referências	37
	Apêndice	39
A	Mapping Study's Instruments	41

1

Introdução

Atualmente, *softwares* têm sido portados para o ambientes web cada vez mais, contudo com uma séries de limitações como a falta de acesso direto ao *hardware* e suporte a apenas algumas linguagens de programação. Para combater esses impedimentos e ainda permitir alto desempenho, deve-se implementar estratégias que possivelmente transpõem as barreiras do navegador.

Mesmo sendo ainda restrito pelo *sandbox* do *browser*, uma possível abordagem seria fazer uso do WebAssembly, um formato binário capaz de ser executado nativamente pelo navegador e que pode ser utilizado como alvo de compilação para linguagens como C/C++ ou Rust. Embora atraente, a completa impossibilidade de acessar o hardware diretamente e a perda de performance em relação ao código nativo demonstram a necessidade de outra abordagem. (W3C Community Group, 2021)

Outra maneira simples de executar código nativo no ambiente web é usá-lo em *server-side* e depois apenas implementar um mecanismo de comunicação com o cliente. Dessa maneira, conseguimos executar um código nativo sem qualquer restrição imposta pelo navegador.

Então, para isolar e distribuir com facilidade aplicativos em *server-side* seria necessário utilizar contêineres, os quais são particularmente interessantes por isolarem inclusive as dependências do programa. Visto que, os contêineres tanto resolvem a questão do isolamento quanto facilitam a distribuição das aplicações nativas, eles possuem um papel chave no desenvolvimento moderno de *software* no lado do servidor.

Um dos algoritmos de busca geralmente associados ao alto desempenho é o *A Star*, que a partir de heurísticas tenta reduzir o custo computacional de sua execução. Uma de suas grandes vantagens é poder parametrizá-lo e fazê-lo agir como outros algoritmos: procurando o mais rápido possível um caminho ou tentando procurar somente pelo melhor caminho. Sendo um algoritmo extremamente útil, não é difícil notar seus usos em diversas áreas, desde *pathfinding* em jogo digitais até o sequenciamento de cadeias biológicas. Logo, um excelente alternativa para se estudar o desempenho em ambientes web.

1.1 Tema

Este trabalho visa demonstrar como podemos acelerar o algoritmo do *A Star* executando a parte computacionalmente intensiva em *server-side*, fazendo uso de recursos nativos da máquina.

1.2 Problema

A pergunta norteadora é: como executar o *A Star* no ambiente web e simultaneamente manter alto desempenho da aplicação? Para responder a essa questão, faz-se necessário entender as tecnologias disponíveis para o contexto da *world wide web*.

Tipicamente aplicações as quais funcionam pelo navegador são divididas em duas partes principais: lado do cliente e lado do servidor. O lado do cliente é executado diretamente no navegador e por esse motivo, fica limitado a apenas as tecnologias ofertadas por ele, como por exemplo: WebAssembly, JavaScript e WebGL.

Já o lado do servidor, não possui qualquer uma das restrições anteriores, podendo usar qualquer tecnologia disponível do hardware em que se está sendo executado. Sendo assim, o servidor pode fazer uso de qualquer linguagem de programação, usar uma diversidade de bibliotecas para GPUs, além de poder executar binários diretamente.

Portanto, mesmo o lado do cliente suportando mecanismos como WebAssembly e WebGL, o lado do servidor é bem mais interessante quanto às possibilidades de otimizações de performance. Nesse sentido, este trabalho irá demonstrar como acelerar uma aplicação baseada em *A Star* usando o lado do servidor e realizando a comunicação com o cliente por meio de interfaces de rede.

1.2.1 Objetivo geral

Desenvolver um *software* web baseado no algoritmo *A Star* com alto desempenho.

1.2.2 Objetivos específicos

Os objetivos específicos deste trabalho:

- Descrever otimizações do *A Star*;
- Desenvolver um *web app* capaz de utilizar tecnologias nativas com alto desempenho;
- Compreender mecanismos de troca de mensagens.

1.3 Estrutura do TCC

Neste item você vai descrever como está constituída a monografia, indicando o que será encontrado em cada uma das sessões seguintes.

1.3.1 Classificação da Pesquisa

Neste item será apresentada a classificação da pesquisa quanto aos objetivos (exploratória, descritiva ou explicativa); aos procedimentos (Pesquisa bibliográfica, Pesquisa documental, Pesquisa experimental, Estudo de caso controle, Levantamento, Estudo de caso ou Estudo de campo) e ao método de investigação científica (qualitativa ou quantitativa).

2

Conceitos gerais e revisão da literatura

2.1 Tecnologias principais desse trabalho

2.1.1 Linguagem de programação C++

2.1.2 Contêineres Linux

2.1.3 Kubernetes

2.2 Trabalhos relacionados

Pesquisas atuais no algoritmo *A Star* mostram o potencial de aceleração com GPUs (*Graphics Processing Unit*) em diversas áreas da ciência. Também houve um foco crescente em tornar esse algoritmo escalável nos mais variados ambientes de *hardware*, fazendo uso de múltiplas GPUs por exemplo.

A tese de Lima (2017) investiga problemas relacionados ao alinhamento de sequências biológicas, tirando proveito do poder computacional de GPUs e do algoritmo *A Star* a fim de encontrar o resultado ótimo no menor tempo possível.

Ao longo desse trabalho, verificamos a utilização da metodologia comparada, a partir de uma série de análises entre três principais soluções: PA-Star, Foldalign 2.5 e CUDA-Sankoff. Dessa forma, a conclusão foi de que o algoritmo CUDA-Sankoff conseguiu o maior ganho de performance por explorar propriedades específicas de GPUs.

Pettersson (2018), em sua dissertação, avalia uma implementação do algoritmo *A Star* em um *host* com múltiplas GPUs, a fim de demonstrar que esse *software* é possível e quais são os pontos mais problemáticos nesse ambiente.

Foi utilizado uma metodologia experimental, ou seja, várias soluções diferentes foram executadas em um ambiente controlado e seus respectivos resultados avaliados.

Ao final da pesquisa, dois principais problemas foram identificados: sobrecarga de término e de comunicação. Dessa forma, trazendo à tona a “sobrecarga de término”, um dos principais gargalos no ambiente de múltiplas GPUs, pois elimina a escalabilidade horizontal ao aumentar o tempo de execução com adição de novas GPUs no sistema.

O artigo de He et al. (2021) busca explorar pontos críticos na otimização de performance

do *A Star* em sistemas com múltiplas GPUs, objetivando aprimorar métodos de particionamento de grafos e de comunicação entre GPUs. Além de propor um novo algoritmo baseado no *A Star*, o *DA** para computação paralela com múltiplas GPUs.

Utilizou-se uma metodologia comparativa e experimental, mostrando as diferentes vantagens entre cada um das implementações do algoritmo *A Star*.

Dessa maneira, a solução dos pesquisadores alcançou em até 3x mais performance, em comparação à solução com uma única GPU. Também foi proposto o algoritmo *DG**, capaz de coordenar múltiplas GPUs e de particionar os grafos de acordo com seu tipo, trazendo também estratégias para lidar com a sobrecarga de múltiplas listas de prioridade.

3

Metodologia

Aqui conterão os métodos e procedimentos adotados no desenvolvimento do trabalho. Esta é uma das sessões mais importantes pois demonstra o poder científico que foi utilizado para a pesquisa. Sem uma boa metodologia a pesquisa pode perder a validade. O pesquisador deve utilizar métodos ou técnicas aceitas pela comunidade científica na busca de provar suas hipóteses.

A metodologia escolhida deve ser aquela que mais se adéqua ao seu objeto de estudo e à abordagem aplicada. Há dois métodos principais: 1) quantitativo, que é o uso de instrumental estatístico, de dados numéricos; e 2) qualitativo, que se caracteriza pela qualificação dos dados coletados, durante a análise do problema.

3.1 Uma seção

Texto.

3.2 Uma outra seção

Texto.

4

Apresentação e Análise dos Resultados

Toda pesquisa deve apresentar uma análise sobre a investigação que foi realizada através da metodologia que foi aplicada. Nesta sessão é interessante inserir tabelas, gráficos, imagens que mostrem os resultados, análise de dados coletados, etc.

É interessante que nessa sessão o autor compare os seus resultados com os resultados de outros trabalhos existentes. Essa comparação aumenta a qualidade do trabalho e demonstra a relevância do mesmo.

5

Conclusões e Trabalhos Futuros

A conclusão deve conter os principais aspectos e contribuições de forma a finalizar o trabalho apresentado. Deve-se apresentar o que era esperado do trabalho através dos objetivos inseridos inicialmente e mostrar o que foi conseguido.

Não deve-se inserir um novo assunto na conclusão. Aqui o autor apresentará as próprias impressões sobre o trabalho efetuado.

É importante também que sejam identificadas limitações e problemas que surgiram durante o desenvolvimento do trabalho e quais as consequências do mesmo.

Os trabalhos futuros devem conter oportunidades de expansão do trabalho apresentado, bem como, novos projetos que puderam ser vislumbrados a partir do desenvolvimento do trabalho

HE, X. et al. Efficient parallel A* search on multi-GPU system. **Future Generation Computer Systems**, [S.l.], v.123, p.35–47, 2021.

LIMA, D. S. **Alinhamento primário e secundário de sequências biológicas em arquiteturas de alto desempenho**. 2017. Tese (Doutorado em Ciência da Computação) — Universidade de Brasília, Brasília.

PETTERSSON, H. **Multi-GPU sliding tile puzzle solving with GA*, Groute and Abstract Zobrist Hashing**. 2018. Dissertação (Mestrado em Ciência da Computação) — Norwegian University of Science and Technology, Norwegian.

W3C Community Group. **WebAssembly**. Acesso em: 2021-12-05, <https://webassembly.org/>.

Apêndice

A

Mapping Study's Instruments

Tabela A.1 List of conferences on which the searches were performed.

Acronym	Conference
APSEC	Asia Pacific Software Engineering Conference
ASE	IEEE/ACM International Conference on Automated Software Engineering
CSMR	European Conference on Software Maintenance and Reengineering
ESEC	European Software Engineering Conference
ESEM	International Symposium on Empirical Software Management and Measurement
ICSE	International Conference on Software Engineering
ICSM	International Conference on Software Maintenance
ICST	International Conference on Software Testing
InfoVis	IEEE Information Visualization Conference
KDD	ACM SIGKDD International Conference on Knowledge Discovery and Data Mining
MSR	Working Conference on Mining Software Repositories
OOPSLA	Object-Oriented Programming, Systems, Languages and Applications
QSIC	International Conference On Quality Software
SAC	ACM Symposium on Applied Computing
SEAA	EUROMICRO Conference on Software Engineering and Advanced Applications
SEDE	19th International Conference on Software Engineering and Data Engineering
SEKE	International Conference on Software Engineering and Knowledge Engineering

Tabela A.2 List of journals in which the searches were performed.

Journal title
ACM Transactions on Software Engineering and Methodology
Automated Software Engineering
Elsevier Information and Software Technology
Elsevier Journal of Systems and Software
Empirical Software Engineering
IEEE Software
IEEE Computer
IEEE Transactions on Software Engineering
International Journal of Software Engineering and Knowledge Engineering
Journal of Software: Evolution and Process
Software Quality Journal
Journal of Software
Software Practice and Experience Journal

Tabela A.3 Search string per Search Engine.

Search Engine	Search String
Google Scholar	bug report OR track OR triage “change request” issue track OR request OR software OR “modification request” OR “defect track” OR “software issue” repositories maintenance evolution
ACM Portal	Abstract: "bug report"or Abstract:"change request"or Abstract:"bug track"or Abstract:"issue track"or Abstract:"defect track"or Abstract:"bug triage"or Abstract: "software issue"or Abstract: "issue request"or Abstract: "modification request") and (Abstract:software or Abstract:maintenance or Abstract:repositories or Abstract:repository
IEEEExplorer (1)	(((((((((("Abstract": "bug report") OR "Abstract": "change request") OR "Abstract": "bug track") OR "Abstract": "software issue") OR "Abstract": "issue request") OR "Abstract": "modification request") OR "Abstract": "issue track") OR "Abstract": "defect track") OR "Abstract": "bug triage") AND "Abstract": software)
IEEEExplorer (2)	(((((((((("Abstract": "bug report") OR "Abstract": "change request") OR "Abstract": "bug track") OR "Abstract": "software issue") OR "Abstract": "issue request") OR "Abstract": "modification request") OR "Abstract": "issue track") OR "Abstract": "defect track") OR "Abstract": "bug triage") AND "Abstract": maintenance)
IEEEExplorer (3)	(((((((((("Abstract": "bug report") OR "Abstract": "change request") OR "Abstract": "bug track") OR "Abstract": "software issue") OR "Abstract": "issue request") OR "Abstract": "modification request") OR "Abstract": "issue track") OR "Abstract": "defect track") OR "Abstract": "bug triage") AND "Abstract": repositories)
IEEEExplorer	(((((((((("Abstract": "bug report") OR "Abstract": "change request") OR "Abstract": "bug track") OR "Abstract": "software issue") OR "Abstract": "issue request") OR "Abstract": "modification request") OR "Abstract": "issue track") OR "Abstract": "defect track") OR "Abstract": "bug triage") AND "Abstract": repository)
Citeseer Library	(abstract: "bug report"OR abstract:"change request"OR abstract:"bug track"OR abstract:"issue track"OR abstract:"defect track"OR abstract:"bug triage"OR abstract: "software issue"OR abstract: "issue request"OR abstract: "modification request") AND (abstract:software OR abstract:maintenance OR abstract:repositories OR abstract:repository)
Elsevier	("bug report"OR "change request"OR "bug track"OR "issue track"OR "defect track"OR "bug triage"OR "software issue"OR "issue request"OR "modification request") AND (software OR maintenance OR repositories OR repository)
Scirus	("bug report"OR "change request"OR "bug track"OR "issue track"OR "defect track"OR "bug triage"OR "software issue"OR "issue request"OR "modification request") AND (software maintenance OR repositories OR repository) ANDNOT (medical OR aerospace)
ScienceDirect	("bug report"OR "change request"OR "bug track"OR "issue track"OR "defect track"OR "bug triage"OR "issue request"OR "modification request") AND LIMIT-TO(topics, "soft ware")
Scopus	("bug report"OR "change request"OR "bug track"OR "issue track"OR "defect track"OR "bug triage"OR "software issue"OR "issue request"OR "modification request") AND (software maintenance OR repositories OR repository)
Wiley	("bug report"OR "change request"OR "bug track"OR "issue track"OR "defect track"OR "bug triage"OR "software issue"OR "issue request"OR "modification request") AND (software maintenance OR repositories OR repository)
ISI Web of Knowledge	("bug report"OR "change request"OR "bug track"OR "issue track"OR "defect track"OR "bug triage"OR "software issue"OR "issue request"OR "modification request") AND (software maintenance OR repositories OR repository) ANDNOT (medical OR aerospace)
SpringerLink	("bug report"OR "change request"OR "bug track"OR "issue track"OR "defect track"OR "bug triage"OR "software issue"OR "issue request"OR "modification request") AND (software maintenance OR repositories OR repository) ANDNOT (medical OR aerospace)