Desenvolvimento e Avaliação de Algoritmos de Hashing Perceptual para Vídeos em Ambiente Paralelo

MATEUS CORDOVA DA SILVA SANTOS; RAFAEL BURLAMAQUI AMARAL; MARILTON SANCHOTENE DE AGUIAR

[1] *Universidade Federal de Pelotas – mcdssantos@inf.ufpel.edu.br*
[2] *Universidade Federal de Pelotas – rafael.amaral@inf.ufpel.edu.br*
[3] *Universidade Federal de Pelotas – marilton@inf.ufpel.edu.br*

1. INTRODUÇÃO

Na era do avanço tecnológico e da ampla acessibilidade à internet, o consumo de mídias digitais experimentou um notável aumento nas últimas décadas. O tráfego de vídeo e áudio, em particular, tem dominado o consumo de dados da internet por anos. De acordo com o *Cisco Annual Internet Report 2018-2023*, estima-se que até o final de 2023, aproximadamente 70% da população global terá acesso à internet móvel, intensificando ainda mais a demanda por mídias digitais.

No entanto, o crescimento no consumo de conteúdo digital levanta novas preocupações para os criadores de conteúdo digital em relação à propriedade intelectual e aos direitos autorais.

A *blockchain*, com suas características de imutabilidade e descentralização, surge como uma solução eficaz para registrar e proteger ativos digitais.

Conforme Laurence (2019), a *blockchain* é uma estrutura de dados distribuída e imutável, que opera por meio de um registro descentralizado de transações ou informações. Essa tecnologia utiliza criptografia avançada para garantir a segurança e a autenticidade dos dados registrados, tornando-os praticamente incorruptíveis e resistentes a alterações não autorizadas.

O uso de algoritmos de *hashing* perceptual, como o proposto neste artigo, permite a autenticação e verificação de vídeos, assegurando sua integridade e originalidade. O *hash* de um vídeo é uma sequência alfanumérica única e fixa gerada a partir do conteúdo do vídeo por meio de algoritmos de *hash* (DONOHUE, 2014)

O algoritmo proposto é uma implementação eficiente para gerar "impressões digitais" de vídeos através de um processo de *hashing* perceptual, com o foco no escalonamento do algoritmo para processo paralelo distribuído.

Seu diferencial está na capacidade de escalar o processamento para ambientes paralelos e distribuídos, permitindo o processamento simultâneo de grandes volumes de dados.

2. METODOLOGIA

Os experimentos foram conduzidos no Laboratório de Sistemas Ubíquos e Paralelos (LUPS)[1] da Universidade Federal de Pelotas, como parte do projeto de pesquisa Nuvem IaaS. Esse projeto investiga um modelo de serviço de computação em nuvem que disponibiliza recursos de infraestrutura virtualizados por meio da Internet.

O ambiente de teste utilizado foi composto por sete computadores interconectados, com configurações de hardware idênticas, assegurando

[1] https://lups.inf.ufpel.edu.br/about/doku.php?id=inicio Acesso em: 10/10/2024

uniformidade de desempenho e facilidade de gerenciamento. Cada máquina é equipada com um processador Intel Core i5 de 4ª geração, 16 GB de memória RAM e um SSD de 120 GB.

Todos os computadores operam com o sistema Ubuntu Server, escolhido por sua ampla variedade de bibliotecas e frameworks, suporte à virtualização e ampla adoção em ambientes de computação em nuvem.

O algoritmo foi desenvolvido em Python, utilizando as seguintes dependências: *OpenCV2* para processamento de vídeo, *Numpy* para manipulação de mosaicos, *ImageHash* para geração de hash dos mosaicos e Dask para distribuir as tarefas entre as máquinas do cluster.

Para o desenvolvimento do experimento, três algoritmos foram criados e adaptados com o objetivo de otimizar o desempenho e evidenciar, de forma mais clara, as diferenças de *performance* entre suas diferentes versões.

A primeira versão do algoritmo desenvolvido para o experimento foi para reduzir o algoritmo original de multi-thread para single-thread com a intenção de demonstrar o uso de multi-thread e do processamento paralelo.

Para o algoritmo de processamento paralelo, utilizamos um total de sete máquinas: uma dedicada à orquestração do agendamento de tarefas e seis destinadas ao processamento paralelo.

Nesse contexto, empregamos o Dask, uma biblioteca de computação paralela em Python que possibilita o processamento eficiente de grandes volumes de dados. Dask distribui tarefas entre múltiplos núcleos de CPU ou clusters de máquinas, sendo especialmente útil para trabalhar com conjuntos de dados que não cabem na memória. Além disso, sua interface semelhante à do Pandas[2] facilita a transição para usuários já familiarizados com a manipulação de dados.

3. RESULTADOS E DISCUSSÃO

O foco principal deste documento é demonstrar a escalabilidade do algoritmo em relação ao original e evidenciar de forma prática a necessidade do processamento paralelo distribuído quando se trata de grandes volumes de dados.

Todos os três algoritmos podem ser encontrados no repositório GitHub[3] do autor. Eles foram executados com os mesmos parâmetros sobre o mesmo arquivo de vídeo, garantindo uma comparação justa e equivalente entre eles. Dessa forma, todos os algoritmos devem retornar exatamente o mesmo hash para o mesmo vídeo.

É importante observar que, para os algoritmos de processamento paralelo ou multi-thread, a sequência de hashes gerada pode apresentar ordens diferentes. Isso ocorre porque as threads e o processamento paralelo podem ter tempos de execução variados para cada mosaico. Por exemplo, dependendo da carga de trabalho e do estado de cada núcleo de CPU no momento da execução, alguns mosaicos podem ser processados mais rapidamente do que outros. Isso pode resultar em uma variação na ordem dos hashes, mesmo que os valores individuais permaneçam consistentes.

Portanto, enquanto os hashes finais devem ser idênticos, a sequência em que são apresentados pode diferir entre as execuções, refletindo a natureza do processamento concorrente.

---

[2] https://pandas.pydata.org/ Acesso em: 10/10/2024
[3] https://github.com/thehatb0y/VideoToHash Acesso em: 10/10/2024

Os resultados obtidos demonstram uma significativa melhoria no tempo de execução ao se utilizar o processamento paralelo. Na Tabela 1, observamos que o algoritmo single-thread levou 207 segundos para processar o vídeo, enquanto o algoritmo multi-thread reduziu esse tempo para 120 segundos.

A implementação em cluster, utilizando sete máquinas, apresentou o melhor desempenho, com um tempo de execução de apenas 27 segundos.

Esses resultados ressaltam a eficiência do processamento paralelo em cenários de alta demanda, onde o tempo de resposta é crucial.

Além disso, a análise de consumo de recursos revela que a utilização de múltiplas threads e máquinas não apenas acelera o processamento, mas também permite uma melhor utilização do hardware disponível. O algoritmo multi-thread, por exemplo, teve a utilização média de 97% no uso da CPU, enquanto o cluster alcançou um consumo médio de 99%. Esses dados indicam que, ao distribuir a carga de trabalho, é possível otimizar não apenas o tempo de execução, mas também a eficiência no uso dos recursos computacionais.

Tipo	Avg Proc %	Tempo
Sigle-Thread	42%	207s
Multi-Thread	97%	120s
Cluster - 7	99%	27s

Tabela 1 - Consumo de Recursos e Tempo de execução

4. CONCLUSÕES

Os resultados obtidos com a implementação do algoritmo proposto confirmam sua viabilidade e eficiência na autenticação de vídeos por meio do *hashing* perceptual. A principal contribuição deste trabalho foi a criação de um sistema escalável, capaz de gerar "impressões digitais" únicas de vídeos, sem a necessidade de armazenamento de grandes arquivos, utilizando técnicas de processamento paralelo e distribuído.

Ao comparar o desempenho dos três algoritmos (single-thread, multi-thread e em cluster), observamos que o processamento paralelo não só reduz significativamente o tempo de execução, mas também melhora o uso dos recursos computacionais disponíveis. Como evidenciado pelos resultados, o tempo de processamento foi reduzido de 207 segundos no algoritmo single-thread para 27 segundos no ambiente de cluster, que utilizou sete máquinas para distribuir a carga de trabalho. Essa redução de tempo representa uma melhoria de quase 90%, demonstrando a eficácia do uso de múltiplos núcleos e máquinas em tarefas de grande demanda computacional.

O algoritmo em cluster, ao utilizar a biblioteca Dask, mostrou-se particularmente eficaz na distribuição das tarefas de processamento de mosaicos entre diferentes máquinas. Além disso, a análise de consumo de recursos revelou que o uso de múltiplas threads e máquinas levou a uma utilização quase total da capacidade de CPU, com a implementação em cluster atingindo um uso médio de 99%. Esse resultado não apenas acelera o processamento, mas também maximiza a eficiência do hardware, tornando-o uma solução viável para ambientes de alto desempenho.

Além disso, o uso de *hashing* perceptual combinado com o processamento paralelo mostrou-se uma ferramenta poderosa para a autenticação de vídeos. Esta abordagem oferece uma solução escalável para a proteção de direitos autorais e verificação de integridade de conteúdo digital, sendo potencialmente aplicável em diversos contextos, como sistemas de verificação de vídeos em redes sociais e plataformas de streaming.

Futuras pesquisas podem explorar a aplicação desse método em outros tipos de mídia, como imagens ou áudio, além de investigar técnicas adicionais de otimização, como o uso de GPUs para acelerar ainda mais o processo. Outro campo promissor é a integração com tecnologias de *blockchain*, garantindo a autenticidade e integridade dos vídeos de maneira descentralizada e imutável. A contínua evolução do consumo de mídias digitais torna o aprimoramento desses sistemas uma área de grande relevância.

5. REFERÊNCIAS BIBLIOGRÁFICAS

CISCO. **Cisco Annual Internet Report (2018–2023)**. Cisco, 2019. Disponível em: Cisco Annual Internet Report (2018–2023) White Paper. Acesso em: 25 jul. 2023.

DONOHUE, Brian. Hash: o que são e como funcionam. Blog Kaspersky Daily, 4 abr. 2014. Disponível em: Hashs criptográficos usados para armazenar senhas, Detecção de Malware | Blog oficial da Kaspersky. Acesso em: 24 jul. 2023.

LAURENCE, Tiana. **Blockchain para Leigos**. 2. ed. Rio de Janeiro: Alta Books, 2019.

Debate mostra desafios da propriedade intelectual frente mudanças trazidas pela pandemia. Gov.br, 30 out. 2020. Notícias. Disponível em: Debate mostra desafios da propriedade intelectual frente mudanças trazidas pela pandemia. Acesso em: 14 ago. 2023.

RODECK, David. **Understanding Blockchain Technology**. Forbes, 23 mai. 2023. Disponível em: Understanding Blockchain Technology. Acesso em: 30 jul. 2023.