

# Avaliação das Estruturas de Arquivo para Processamento de Dados Sísmicos com Alto Desempenho na Nuvem Computacional

Murilo G. Andrietta<sup>1</sup>, Charles B. Rodamilans<sup>1</sup>, Antonio Guimarães<sup>2</sup> e Edson Borin<sup>1,2</sup>

<sup>1</sup>Centro de Estudo de Petróleo (CEPETRO)  
Universidade Estadual de Campinas (UNICAMP) – Campinas, Brasil

{mgandrietta, charles}@ggaunicamp.com

<sup>2</sup>Instituto de Computação  
Universidade Estadual de Campinas (UNICAMP) – Campinas, Brasil

{antonio.guimaraes, edson}@ic.unicamp.br

**Abstract.** *Applications in the seismology field rely on the processing of up to hundreds of terabytes of data and their performance may be strongly affected by IO operations. In this paper, we generalize the main file structures currently used to store seismic data and evaluate their performance. We present a theoretical analysis of data loading operations and a benchmark on the AWS public cloud, using three different storage technologies (HDD, SSD, and EFS). We show that an adequate choice of the file structure for a typical use case enables an up to 193 times reduction on the amount of data read and 139 times speedup in time. Our results also indicate that the use of more expensive cloud instances presents negligible effects on the performance of network storage, despite featuring enhanced network transmission capacity.*

**Resumo.** *As aplicações computacionais na área de sismologia processam dados até a ordem de centenas de terabytes e seus desempenhos podem ser fortemente afetados pelas operações de Leitura e Escrita. Este artigo generaliza as principais estruturas de arquivos para armazenamento de dados sísmicos e avalia seus desempenhos. São apresentadas uma análise teórica do carregamento de dados na memória e uma análise de desempenho na nuvem computacional, utilizando diferentes tecnologias de armazenamento (HDD, SSD e EFS). A partir delas, obteve-se que a escolha adequada da estrutura de arquivo para um caso de uso típico permite uma redução de até 193 vezes na quantidade de dados lidos. Observou-se também que a melhor estrutura avaliada desempenha até 139 vezes mais rapidamente do que a estrutura adotada pelo formato SEG-Y, usado como padrão pela Agência Nacional de Petróleo do Brasil. Por fim, nos experimentos com armazenamento em rede, concluiu-se que o uso de instâncias mais custosas, mas com maior capacidade de transmissão de dados, não traz benefícios significativos.*

## 1. Introdução

Métodos de armazenamento e leitura de dados são partes fundamentais na determinação do desempenho de diversas aplicações científicas e de engenharia. Na área de sismologia,

---

Os autores agradecem à Petrobras, à FAPESP e ao CNPq pelo apoio financeiro à pesquisa sob os processos nº 2013/08293-7, 2019/12783-6 e 313012/2017-2.

o processo para se determinar as propriedades do subsolo envolve aplicações computacionais que trabalham com dados até a ordem de centenas de *terabytes* e cujo desempenho pode ser fortemente afetado pela eficiência das operações de escrita e leitura de dados de arquivos. Tal processo tem grande importância nas áreas de exploração de hidrocarbonetos (*e.g.* petróleo e gás) e requer uso intensivo de ferramentas da Computação de Alto Desempenho para o processamento dos dados.

Diferentes subáreas da sismologia adotam estruturas de dados distintas para armazenamento de dados similares obtidos do subsolo [Krischer et al. 2016]. Na sismologia passiva, algumas das estruturas de arquivos em uso tendem a ser bastante complexas, com adoção de índices e tabelas relacionais que aceleraram o acesso aos dados [Krischer et al. 2016]. Nela, o objetivo é observar e registrar ondas sonoras geradas por eventos naturais, como terremotos. Já na sismologia ativa, as ondas captadas do subsolo são artificialmente geradas na região em que se tem interesse em conhecer as características do subsolo. Neste domínio, adotou-se o uso de formatos mais simples, porém mais apropriados à extensão de arquivos por operações de concatenação (em inglês, *appending*). Tal fato tinha grande importância numa época em que escrita sequencial era fortemente valorizada devido ao uso de fitas magnéticas. Considerando os meios de armazenamento modernos, entretanto, tal vantagem precisa ser reavaliada.

A recente tendência de utilização da nuvem computacional para processamento de dados sísmicos [Brandsberg-Dahl 2017] traz novos fatores a serem considerados na estruturação de formatos de arquivo. Na nuvem, os diferentes métodos de armazenamento disponíveis apresentam particularidades que afetam fortemente não só o desempenho, mas também o custo da execução de aplicações. Considerando isso, este artigo faz uma comparação entre três padrões de estruturas de arquivos para dados sísmicos e apresenta as seguintes contribuições:

- A definição de três estruturas que generalizam aquelas utilizadas pelos formatos mais populares na sismologia;
- Uma análise teórica do desempenho de leitura das estruturas considerando os mecanismos de paginação dos sistemas operacionais modernos;
- Uma medição do desempenho das estruturas para um caso de uso típico na sismologia ativa, mostrando que as estruturas modernas utilizadas na sismologia passiva são até 139 vezes mais rápidas;
- E uma análise experimental do impacto das estruturas no desempenho de diferentes recursos de armazenamento na nuvem computacional.

A partir da análise teórica, foi possível demonstrar que a escolha de uma estrutura de arquivos adequada às características do sistema de armazenamento subjacente permite uma leitura até 193 vezes mais eficiente em um caso de uso típico na sismologia. Já a partir da análise experimental realizada, observou-se que algumas das estruturas modernas utilizadas na sismologia passiva apresentam desempenho, medido em tempo de leitura, de 12 a 139 vezes melhor do que aquelas utilizadas pelos formatos mais populares da sismologia ativa. O maior ganho médio de desempenho obtido resultou dos experimentos realizados com a utilização de Unidades de Estado Sólido (SSD, do inglês *Solid-State Drive*), enquanto o menor foi resultado de experimentos com armazenamento sobre a rede (utilizando o Amazon EFS [AWS 2016]). Esperava-se que o desempenho de leitura sobre a rede fosse limitado pela tecnologia de rede e que o uso de instâncias com maior capacidade de transmissão de dados reduzisse o tempo de leitura, mas isso não foi observado.

O artigo está organizado da seguinte forma: a Seção 2 descreve o processo de aquisição sísmica e os formatos de arquivos utilizados em sismologia; a Seção 3 apresenta a generalização das estruturas de arquivos; a Seção 4 descreve a metodologia dos experimentos; a Seção 5 explica os resultados experimentais; a Seção 6 lista os trabalhos relacionados; por fim, a Seção 7 apresenta as conclusões do trabalho.

## 2. Processo de Aquisição e Formatos de Arquivos de Sísmicos

### 2.1. Aquisição e representação de dados sísmicos

Na sismologia ativa, o processo de aquisição de dados sísmicos é iniciado através da emissão de ondas (geralmente geradas por explosões) que se deslocam de um certo ponto da superfície em direção ao interior da Terra, conforme apresentado na Figura 1 (a). Ao longo desse caminho, as ondas atravessam diferentes camadas geológicas, com propriedades físicas que afetam a velocidade de propagação da onda. Ao atravessar a interface entre duas camadas, parte da energia da onda é refletida e se desloca em direção à superfície, ao encontro de geofones (sensores) posicionados para a captação das oscilações da onda. Por fim, os dados coletados são armazenados para serem analisados posteriormente.

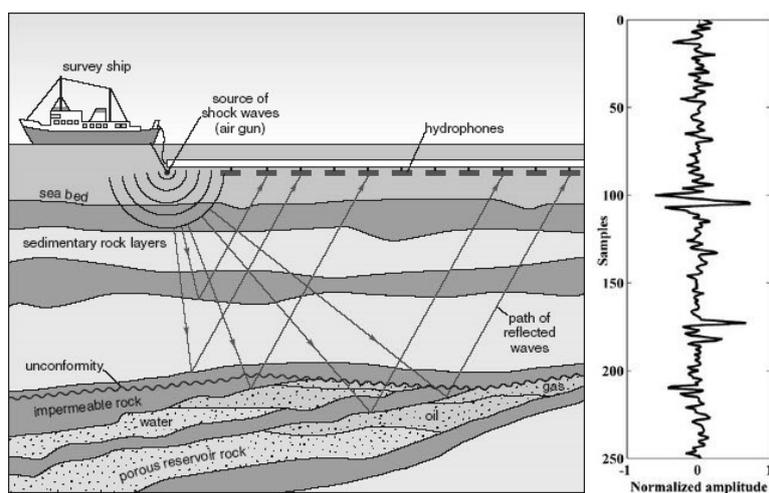


Figura 1. (a) Processo de aquisição sísmica marítima (Fonte: [Open University 2011]); e (b) oscilações registradas por um geofone.

A sequência de oscilações registradas por um geofone após uma explosão, ilustradas na Figura 1 (b), compõem as amostras de um traço sísmico. Além das amostras, também é importante registrar metainformações a respeito da coleta que foi realizada, como, por exemplo, a posição geográfica da fonte que gerou a explosão e do geofone que registrou as oscilações. Estas metainformações são geralmente agrupadas em uma estrutura chamada de cabeçalho do traço sísmico. Dessa forma, um traço sísmico possui um cabeçalho com atributos que armazenam as metainformações da coleta sísmica e uma sequência de valores que representam as amostras registradas.

Após a aquisição, os dados são processados para gerar imagens que auxiliam na interpretação das camadas de interesse. As técnicas Common Midpoint (CMP) [Mayne 1962] e Full-Waveform Inversion (FWI) [Tarantola 1984] são exemplos de métodos usados no processamento de dados sísmicos. Comumente, tais métodos não precisam de todas as informações capturadas no processo de aquisição. Isso ocorre, pois uma aquisição coerente apresenta dados para todos os métodos, mas cada método necessita de apenas um subconjunto das informações coletadas. Um exemplo está na aplicação

do método CMP, que utiliza apenas algumas poucas variáveis, como os quatro atributos de localização (posições espaciais dos geofones e das fontes geradoras de ondas), o atributo *Common Depth Point (CDP)* e a quantidade de amostras presente em cada traço sísmico, fazendo com que a maior parte dos atributos armazenados no cabeçalho (que podem chegar a centenas [Yang et al. 2000]) não sejam necessárias durante o processamento.

## 2.2. Formatos de arquivos para dados sísmicos

Diversas ferramentas comerciais implementam os métodos citados na Seção 2.1 e a maioria delas utiliza estruturas de dados internas bastante eficientes para representar os dados sísmicos. Durante a aquisição, para o armazenamento e para o compartilhamento, entretanto, convencionou-se principalmente o uso do formato SEG-Y e de seus derivados.

O formato SEG-Y foi apresentado inicialmente em 1975 [Barry et al. 1975], sendo posteriormente revisado para adição de suporte à aquisição 3D, armazenamento em mídias de alta capacidade, entre outras melhorias. Vale mencionar que este formato é atualmente utilizado pela Agência Nacional de Petróleo do Brasil como padrão para armazenamento e compartilhamento de dados sísmicos. Os principais derivados do SEG-Y em uso atualmente são os formatos Seismic Unix [John W. Stockwell and Cohen 2008] e RSF [Claerbout 1991]. O primeiro foi apresentado em 1997 como uma simplificação do SEG-Y. Ele eliminou os cabeçalhos globais até então utilizados, enquanto manteve os cabeçalhos de cada traço sísmico, agora com 240 *bytes*. A Figura 2 ilustra sua estrutura e a Tabela 1 apresenta exemplos de atributos do seu cabeçalho. O segundo foi criado em 1991 com propósito de “maximizar a conveniência, transparência e flexibilidade” no tratamento dos dados sísmicos [Fomel et al. 2013] utilizando representações que permitem a interpretação dos dados, ainda que ineficiente, sem o uso de ferramentas computacionais.



**Figura 2. Estrutura organizacional de arquivos su.**

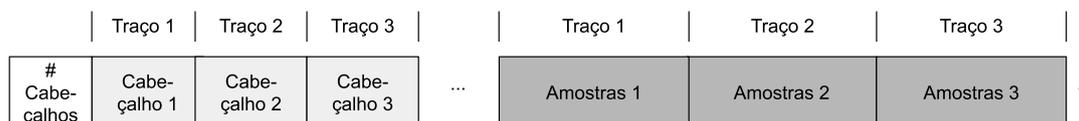
**Tabela 1. Exemplos de atributos do cabeçalho SU.**

Palavra Chave	Descrição	Tamanho ( <i>bytes</i> )
ns	número de amostras no traço	2
sx, sy	coordenadas bidimensional (x, y) da fonte	4 (cada)
gx, gy	coordenadas bidimensional (x, y) do receptor	4 (cada)
cdp	Número do conjunto ( <i>ensemble</i> ): CDP, CMP, CRP...	4

Em comum aos formatos mencionados, está a estrutura ilustrada pela Figura 2, com o cabeçalho precedendo o conjunto de amostras de cada traço sísmico. Além de nem todos os campos de um cabeçalho serem necessários, alguns dos métodos de processamento sísmico mais utilizados trabalham apenas sobre um subconjunto específico dos traços que é selecionado posteriormente à análise dos cabeçalhos. Por exemplo, o método CMP precisa ordenar os traços sísmicos pelo parâmetro CDP antes de selecionar os traços de interesse. Dessa forma, torna-se necessária a leitura completa dos cabeçalhos quando assim representados, mesmo que o interesse esteja apenas em um pequeno subgrupo.

Já na sismologia passiva, o primeiro formato a ser amplamente adotado como padrão pela comunidade foi o SEED [Ahern et al. 2007] (do inglês *Standard for the Exchange of Earthquake Data*), em 1987. Em suas primeiras versões, apesar de já possuir

divisões lógicas mais adequadas aos meios de armazenamento modernos, ele ainda seguia uma estrutura similar àquela do formato SEG-Y e SU. Em 1990, com sua versão 2.1, o SEED definiu um novo paradigma para construções de formatos de arquivos sísmicos ao adotar uma estrutura que mantinha os cabeçalhos separados das amostras que caracterizam. A Figura 3 ilustra esse tipo de estrutura.



**Figura 3. Estrutura organizacional de arquivos com cabeçalho na frente (HF).**

Formatos mais recentes, além de manterem a separação entre cabeçalhos e amostras, passaram também a adotar estruturas mais complexas para o armazenamento de dados. Os principais exemplos disso estão no uso do modelo de dados hierárquico HDF5 [Kozioł and Editor Padua 2011] e de métodos de indexação de dados para obter melhor desempenho. O uso de estruturas de índices nos cabeçalhos, entretanto, não é novidade na sismologia passiva. Em 1990, a versão 3.0 do formato CSS-DBMS [J. Anderson 1990] (do inglês *Center for Seismic Studies' Database Management System*) já apresentava tabelas relacionais nos índices e seguia o modelo de separação entre cabeçalho e amostras.

Diversas foram as tentativas de se padronizar um formato de arquivo para as subáreas da sismologia. Destaca-se, notavelmente, os esforços de IRIS/PASSCAL Data Group com o desenvolvimento do formato PH5 em 2012 [IRIS/PASSCAL Data Group 2012] e de Krischer *et al.* com a criação do formato ASDF em 2016 [Krischer et al. 2016]. Em ambos os casos, buscou-se manter a compatibilidade com as principais aplicações de toda sismologia, enquanto se aproveitava da estrutura interna mais complexa dos formatos utilizados na sismologia passiva para obter melhor desempenho. Embora suportados por ferramentas de grande utilização, tais formatos não se popularizaram, especialmente entre as aplicações da sismologia ativa.

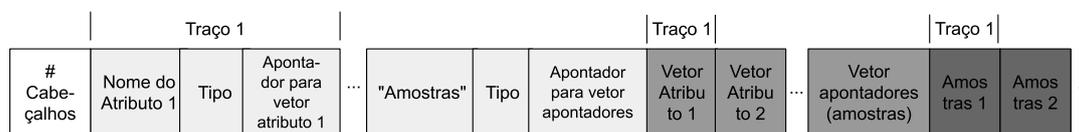
### 3. Generalização das estruturas de dados utilizadas na sismologia

Com base nos formatos de arquivos levantados na Seção 2, propõe-se aqui uma generalização de suas estruturas para facilitar a avaliação dos seus desempenhos em operações de leitura. Estas estruturas foram selecionadas para abranger os formatos com relevante popularidade e são descritas a seguir.

**Cabeçalho-Amostras (HS, do inglês *Header-Sample*):** O conjunto de amostras sucede imediatamente o cabeçalho que o caracteriza. A estrutura se define como apresentado pelo formato Seismic Unix e ilustrado pela Figura 2. Nela se classificam, além do próprio formato Seismic Unix, os formatos SEG-Y, RSF e SEED até a versão 2.0.

**Cabeçalho na frente (HF, do inglês *Header First*):** Cabeçalhos e amostras ficam em seções diferentes do arquivo. Os primeiros são reunidos ao início, enquanto os últimos são reunidos ao fim, agrupando as informações em dois grandes blocos como ilustrado pela Figura 3. Nota-se, entretanto, que não se faz o uso de estruturas de indexação avançadas para acelerar o acesso ao dado. Pelo contrário, os cabeçalhos se mantêm similares aos da estrutura anterior (com exceção da adição de um meta-cabeçalho indicando o número de traços). Classificam-se nesta estrutura alguns dos formatos mais tradicionais da sismologia passiva, como o SEED a partir da versão 2.1.

**Indexado com cabeçalho na frente (HFI, do inglês *Header First Indexed*):** Este modelo tem por intenção generalizar os formatos de arquivo que, além de apresentarem a divisão descrita pela estrutura HF, utilizam estruturas de índices complexas para acelerar o acesso ao dado. A Figura 4 descreve sua estrutura. O meta-cabeçalho é composto pelo número de traços, seguido pela estrutura de indexação em forma de triplas no formato (*nome do atributo, tipo do atributo, apontador para o vetor de atributos*). Finalizando o meta-cabeçalho, tem-se uma tripla relacionada a amostra ("*Amostras*", *tipo, apontador para vetor de apontadores para amostras*). A seguir, tem-se uma estrutura de vetores com os valores de cada atributo (*vetor atributo 1, vetor atributo 2, ..., vetor atributo N*), seguido pelo vetor apontadores de amostras e, por fim, pelas amostras com seus valores (*Amostra 1, Amostra 2*). Classificam-se nesta estrutura alguns dos formatos mais modernos utilizados na sismologia passiva, como o PH5 e o ASDF, assim como alguns dos formatos tradicionais como CSS-DBMS.



**Figura 4. Estrutura organizacional de arquivos Indexado com cabeçalho na frente (HFI).**

#### 4. Metodologia Experimental

Esta seção detalha os experimentos realizados para analisar o desempenho de leitura das diferentes estruturas de arquivos. O experimento envolve a leitura de atributos dos cabeçalhos dos traços sísmicos que são necessários pela técnica *Common Midpoint* (CMP), clássica na sismologia ativa. Esta técnica agrupa e processa traços sísmicos com ponto médio (i.e. o ponto espacial localizado entre a fonte que produziu a onda e o geofone que registrou as amostras) em comum. Para isto, são necessários especialmente os atributos dos cabeçalhos apresentados na Tabela 1.

Para a análise de desempenho, foram utilizados três dados sísmicos distintos, com tamanhos de 16, 64 e 256 GB. Para cada um deles, gerou-se três arquivos, um para cada estrutura apresentada (HS, HF e HFI). O desempenho é medido a partir da leitura dos seis atributos descritos na Tabela 1 do cabeçalho de cada traço sísmico. Os cabeçalhos dos traços contêm os mesmos campos do formato SU, totalizando 94 atributos e 240 bytes.

Considerando a tendência de uso da nuvem computacional para o processamento de dados sísmicos [Brandsberg-Dahl 2017], todos os experimentos ocorreram em máquinas virtuais da *Amazon Web Services* (AWS)[AWS 2019]. Os tipos de instâncias utilizadas, seus custos e suas características são apresentados na Tabela 2. As instâncias h1.2xlarge e i3.xlarge utilizam armazenamento de instância [AWS 2016] Disco Rígido (HDD, do inglês *Hard Disk Drive*) e Unidade de Estado Sólido (SSD, do inglês *Solid-State Drives*), ou seja, utilizam armazenamento local, sem uso da rede. As instâncias i3.xlarge, i3.8xlarge e c5n.4xlarge utilizam o armazenamento *Elastic File System* (EFS).

O *Elastic File System* (EFS) [AWS 2016] é um sistema de arquivo elástico baseado no *Network File System version 4* (NFSv4). A elasticidade está na capacidade de alteração do tamanho do sistema de arquivos de forma dinâmica e gerenciada pelo provedor. Este dimensionamento ajusta-se dinamicamente ao tamanho dos dados armazenados e o valor cobrado está diretamente relacionado ao tamanho do sistema de arquivos.

**Tabela 2. Instâncias utilizadas na execução dos experimentos.**

Instância	Custo Instância (Dólar/hora)	Armazenamento	Memória (GB)	Desempenho Rede (Gbps)
h1.2xlarge	0,468	Local HDD	32,0	Não se aplica
i3.xlarge	0,312	Local SSD	30,5	Não se aplica
	0,312	EFS	30,5	Até 10
i3.8xlarge	2,496	EFS	244,0	10
c5n.4xlarge	0,864	EFS	42,0	Até 25

No EFS, existem dois tipos de taxa de transferência de dados [AWS 2016]: (a) modo provisionado e (b) modo de intermitência. No modo provisionado, a taxa de transferência é contínua, independe do tamanho total dos arquivos armazenados, e possui um valor cobrado mais elevado (do que o modo de intermitência). No modo de intermitência, a taxa de transferência é variável, está diretamente relacionada ao tamanho total do armazenamento e existe uma alta taxa de transferência (*e.g.* 100 MB/s) que dura enquanto houverem créditos de *burst*. Após o término deles, utiliza-se a taxa de transferência base, que é menor (*e.g.* 12,5 MB/s). Os créditos são restaurados diariamente. O consumo destes créditos é um dos motivos pelo qual é importante reduzir a leitura de dados dos arquivos.

Quando se cria um sistema de arquivos, é fornecido um crédito de *burst* inicial. Estes créditos serão utilizados antes dos seus créditos de armazenamento. Para aproveitarmos os créditos iniciais em nossos experimentos, foi criado um EFS e monitorada a sua utilização de créditos. Quando não havia crédito para um próximo experimento, era criado um novo EFS. Dessa forma, evitou-se que a utilização dos créditos influenciasse os resultados finais e padronizou-se a taxa de transferência para todos os experimentos.

Cada tipo de instância continha três arquivos com os tamanhos mencionados, de forma que no momento da execução do experimento, cada instância teria um total de  $256 + 64 + 16 \text{ GB} = 336 \text{ GB}$  na sua respectiva tecnologia de armazenamento (local HDD, local SSD ou EFS). Para cada experimento - execução da técnica CMP utilizando uma estrutura de arquivo, um tipo de instância/tipo armazenamento - foram realizadas 3 repetições e para cada repetição a cache da memória principal foi limpa.

As operações de leitura nas diferentes estruturas de arquivos (HS, HF e HFI) possuem um acesso sequencial e foram realizadas de forma síncrona. Foi utilizado o *mmap* [Rubini and Corbet 2001] para realizar o mapeamento dos arquivos em uma região de memória e carregar na memória somente as regiões dos arquivos que estão sendo utilizados no momento que a aplicação necessitar. O sistema operacional utilizado foi o Ubuntu 18.04.1 LTS e o compilador foi o `gcc` versão 7.4.0.

## 5. Análise dos Resultados

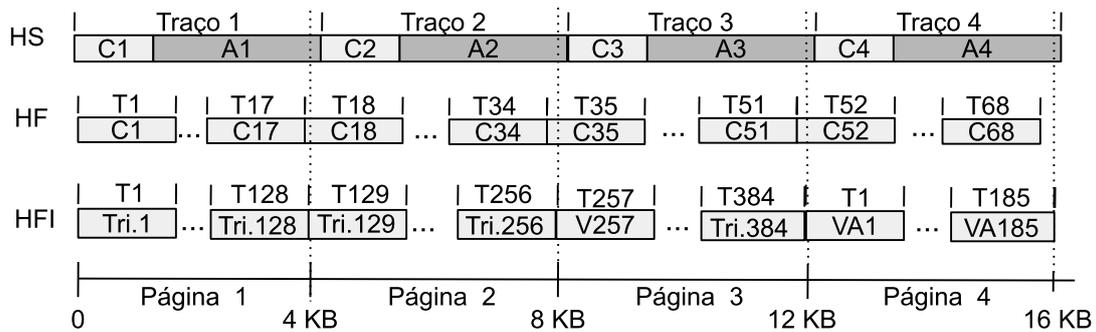
### 5.1. Análise Teórica

O objetivo desta análise é determinar a quantidade de *bytes* que serão levados à memória principal ao realizar a leitura dos 6 atributos de interesse dos traços dos dados sísmicos em cada uma das três estruturas de arquivos apresentadas. A quantidade de amostras por traço depende da taxa de amostragem (quantidade de amostras por segundo), do equipamento utilizado e do tempo da aquisição sísmica. A taxa de amostragem e o tempo de aquisição podem variar de acordo com a tecnologia utilizada, sendo que a tendência é que a taxa de amostragem e o tempo aumentem à medida que a tecnologia evolui e o imageamento de camadas cada vez mais profundas seja necessário, como é o caso do pré-sal no Brasil.

Na análise teórica, utilizou-se inicialmente um dado de 4 TB que possui 1000 amostras por traço. Como cada amostra é representada por um número na representação ponto-flutuante de precisão simples (4 bytes) e, no formato SU, o cabeçalho do traço possui 240 bytes (94 atributos com diferentes tamanhos), cada traço ocupa  $(240+1000 \times 4) = 4\,240$  bytes. Consequentemente, o dado possui  $(4\text{ TB} / 4\,240\text{ bytes})$  1 037 275 121 traços.

Com o intuito de explorar a localidade espacial e melhorar o desempenho do sistema, quando um dado é lido de um arquivo, o sistema operacional busca e armazena na memória, além do dado lido, os dados vizinhos. Dessa forma, quando um dado vizinho for solicitado, ele poderá ser recuperado diretamente da memória. De forma geral, os sistemas computacionais modernos permitem que os dados sejam organizados na memória em páginas, que podem ser de 4 KB, 2 MB ou outro tamanho, a depender da arquitetura. Nos sistemas baseados em processadores x86 e ARM, os sistemas operacionais organizam, geralmente, os dados em páginas de 4 KB e trazem para a memória uma página inteira toda vez que o dado solicitado não está na memória. Dessa forma, se o programa ler o primeiro byte de um arquivo, os primeiros 4 096 bytes serão carregados na memória.

A Figura 5 ilustra como os atributos e amostras de traços sísmicos com 4 240 bytes são organizados em páginas de 4 KB. Note que, na estrutura HS, quando um atributo do cabeçalho C1 é lido, a página 1 é lida e, como consequência, todos os atributos (C1) e quase todas as amostras (A1) do traço T1 são lidos, mesmo que não sejam necessários. Dessa forma, se lermos 22 bytes de cada cabeçalho do dado sísmico, quase todas as páginas do arquivo serão carregadas na memória, ou seja, 4 TB de dados serão lidos do arquivo. Já no caso da estrutura HF, como os cabeçalhos são agrupados no início do arquivo, se lermos 22 bytes de cada cabeçalho, apenas os 1 037 275 121 cabeçalhos, ou seja  $\lceil \frac{1\,037\,275\,121 \times 240}{4096} \rceil$  60 777 839 páginas, serão lidos, resultando em 231,85 GB lidos do arquivo. Este montante corresponde a 5,66% de 4 TB.



**Figura 5. Distribuição de dados em páginas de 4 KB nas estruturas HS, HF e HFI.**

Por fim, no caso da estrutura HFI, teríamos que ler as páginas que contém as 6 triplas que descrevem os atributos de interesse e depois ler as páginas que contém os valores destes atributos. Dessa forma, no pior caso, onde as triplas não estão na mesma página, teríamos que ler 6 páginas para recuperar as 6 triplas, 5 064 815 páginas para recuperar os 5 atributos de 4 bytes cada e 506 482 páginas para recuperar o atributo de 2 bytes de todos os traços. Consequentemente, neste caso, seriam lidos 21,25 GB de dados do arquivo, o que corresponde a 0,52% de 4 TB.

A Tabela 3 resume as informações dos dados sísmicos utilizados na análise acima ( $DS_1$ ) e daqueles utilizados nos experimentos práticos ( $DS_2$ ,  $DS_3$  e  $DS_4$ ). A Tabela 4 mostra um resumo da eficiência da leitura dos 6 atributos de interesse utilizando as estru-

turas HS, HF e HFI. A coluna 2 da Tabela 4 mostra a quantidade de dados necessária para acessar os 6 atributos (22 *bytes*) de todos os cabeçalhos, enquanto que as colunas 3, 4 e 5 mostram a quantidade de dados efetivamente lida do arquivo em cada uma das estruturas. Por fim, as colunas 6, 7 e 8 mostram a eficiência da leitura, ou seja, a quantidade de dados necessária dividida pela quantidade de dados lidos para cada uma das estruturas.

**Tabela 3. Descrição dos dados sísmicos utilizados.**

Dado	Tamanho em <i>bytes</i>	# de amostras por traço	# total de traços
$DS_1$	4 TB	1 000	1 037 275 121
$DS_2$	256 GB	700	88 301 137
$DS_3$	64 GB	700	22 605 092
$DS_4$	16 GB	700	5 651 273

**Tabela 4. Comparação teórica da eficiência de leitura dos 6 atributos de interesse para os diferentes dados sísmicos em cada estrutura (HS, HF e HFI).**

Dado	Leitura Necessária (GB)	Leitura Realizada HS (GB)	Leitura Realizada HF (GB)	Leitura Realizada HFI (GB)	Aprov. de Leit. HS (%)	Aprov. de Leit. HF (%)	Aprov. de Leit. HFI (%)
$DS_1$	21,25	4 096,00	231,85	21,25	0,52%	9,17%	100%
$DS_2$	1,81	256,00	19,74	1,85	0,72%	9,17%	100%
$DS_3$	0,46	64,00	5,05	0,46	0,72%	9,17%	100%
$DS_4$	0,12	16,00	1,26	0,12	0,72%	9,17%	100%

Supondo-se que apenas o atributo CDP (4 *bytes*) fosse necessário para realizar a indexação dos traços, então a eficiência da estrutura HS cairia de 0,52%/0,72% para 0,09%/0,13% e a eficiência da estrutura HF cairia de 9,17% para 1,67% enquanto que a eficiência da estrutura HFI permaneceria em torno de 100%.

## 5.2. Resultados Experimentais

Esta seção apresenta os resultados obtidos dos experimentos de leitura dos atributos dos dados sísmicos em máquinas da AWS. As Tabelas 5, 6 e 7 trazem os tempos e as taxas de leitura de arquivos com diferentes tamanhos ( $DS_2$ ,  $DS_3$  e  $DS_4$ ) nas estruturas estudadas.

**Tabela 5. Tempos e taxas de leitura dos atributos do arquivo  $DS_2$  (16 GB).**

Instância (Armazenamento)	HS		HF		HFI	
	Tempo (s)	Taxa (MB/s)	Tempo (s)	Taxa (MB/s)	Tempo (s)	Taxa (MB/s)
i3.xlarge (SSD)	16,9 ± 0,2	950	1,1 ± 0,2	1150	0,1 <sup>a</sup>	850
h1.2xlarge (HDD)	82,7 ± 0,6	190	6,6 ± 0,1	180	0,7 ± 0,1	160
i3.xlarge (EFS)	162,1 ± 0,1	100	11,9 ± 0,1	110	1,5 ± 0,5	80
i3.8xlarge (EFS)	162,1 ± 0,3	100	11,9 ± 0,1	110	1,5 ± 0,5	80
c5n.4xlarge (EFS)	162,4 <sup>a</sup>	100	11,9 <sup>a</sup>	110	1,5 ± 0,6	70

<sup>a</sup> Desvio padrão omitido por não apresentar valor significativo ( $< 0, 1$ ).

Como pode-se observar nas Tabelas 5, 6 e 7, a leitura dos atributos de arquivos que implementam a estrutura HS atinge um desempenho bem inferior quando comparado ao desempenho da leitura nos arquivos que usam a estrutura HFI. O tempo de leitura dos 6 atributos de um arquivo de 256 GB (Tabela 7) armazenado no EFS é de aproximadamente 2 600 segundos quando o arquivo faz uso da estrutura HS e de apenas 19,7 segundos

**Tabela 6. Tempos e taxas de leitura dos atributos do arquivo  $DS_3$  (64 GB).**

Instância (Armazenamento)	HS		HF		HFI	
	Tempo (s)	Taxa (MB/s)	Tempo (s)	Taxa (MB/s)	Tempo (s)	Taxa (MB/s)
i3.xlarge (SSD)	71,2 ± 0,3	900	5,3 ± 0,2	960	0,5 <sup>a</sup>	860
h1.2xlarge (HDD)	333,5 ± 1,7	190	26,7 ± 0,1	190	2,6 ± 0,3	180
i3.xlarge (EFS)	653,1 ± 0,6	100	51,3 ± 0,1	100	5,6 <sup>a</sup>	80
i3.8xlarge (EFS)	652,1 ± 0,2	100	51,3 ± 0,1	100	5,6 <sup>a</sup>	80
c5n.4xlarge (EFS)	653,7 ± 0,3	100	51,5 <sup>a</sup>	100	5,6 <sup>a</sup>	80

<sup>a</sup> Desvio padrão omitido por não apresentar valor significativo ( $< 0, 1$ ).

**Tabela 7. Tempos e taxas de leitura dos atributos do arquivo  $DS_4$  (256 GB).**

Instância (Armazenamento)	HS		HF		HFI	
	Tempo (s)	Taxa (MB/s)	Tempo (s)	Taxa (MB/s)	Tempo (s)	Taxa (MB/s)
i3.xlarge (SSD)	285,8 ± 0,5	900	21,3 ± 0,2	950	2,1 <sup>a</sup>	870
h1.2xlarge (HDD)	1364,6 ± 1,0	190	115,6 ± 1,0	170	9,8 <sup>a</sup>	190
i3.xlarge (EFS)	2614,4 ± 1,3	100	205,7 ± 0,2	100	19,6 <sup>a</sup>	90
i3.8xlarge (EFS)	2623,1 ± 0,6	100	205,6 ± 0,3	100	19,7 <sup>a</sup>	90
c5n.4xlarge (EFS)	2617,5 ± 0,2	100	206,2 <sup>a</sup>	100	19,7 <sup>a</sup>	90

<sup>a</sup> Desvio padrão omitido por não apresentar valor significativo ( $< 0, 1$ ).

quando o arquivo faz uso da estrutura HFI. Ao calcularmos a razão entre os tempos, fixado em uma certa instância, obtemos uma variação entre 7,3% e 8,5% para HF/HS e entre 0,7% e 0,8% para HFI/HS, o que é condizente com a eficiência observada na análise teórica (Tabela 4). Em suma, a leitura dos 6 atributos de arquivos que fazem uso da estrutura HFI é de 104 a 139 vezes mais rápida do que a mesma leitura em arquivos com a estrutura HS. Já a leitura em arquivos que usam a estrutura HF atinge um desempenho de 11 a 15 melhor do que a leitura em arquivos que usam a estrutura HS.

Os resultados também mostram que a tecnologia de armazenamento EFS apresenta os maiores tempos para a execução dos experimentos, enquanto que os SSDs oferecem o melhor desempenho. Enquanto o EFS é capaz de oferecer uma taxa de média de leitura da ordem de 100 MB/s (entre 70 e 110 MB/s nos experimentos), o SSD foi capaz de oferecer taxas que variam de 850 a 1 150 MB/s, e o HDD ofereceu taxas que variam de 160 a 190 MB/s. Nota-se também que o desempenho de leitura de arquivos armazenados no EFS não depende do desempenho da rede das máquinas virtuais. Observe que as instâncias do tipo i3.xlarge, i3.8xlarge e c5n.4xlarge possuem desempenho da rede bem distintos (coluna 5 da Tabela 2), no entanto, o desempenho observado nos experimentos com o EFS não variou em função disso. Por fim, nota-se que o desempenho observado com o EFS é coerente com o desempenho divulgado pela AWS (100 MB/s).

## 6. Trabalhos Relacionados

As dificuldades relacionadas ao armazenamento e à leitura de dados sísmicos são um problema bastante abordado na sismologia. A maior parte da literatura, entretanto, foca em minimiza-lo através de técnicas de compressão e algoritmos de busca considerando apenas as propriedades matemáticas do dado em si. Nickerson *et al.* [Nickerson 1999] utilizaram uma árvore quaternária modificada [Samet 1984] para melhorar o desempenho da busca de informações em arquivos SEG-Y. Assim, eles foram capazes de reduzir em

21,6% o tempo da busca 2D, enquanto utilizavam apenas 2,74% do tamanho original dos arquivos para indexação. Li *et al.* [Li 2017] apresentaram algoritmos para compressão de arquivos SEG-Y com taxa de compressão recorde de 75 - 80%. Tais trabalhos poderiam ser vistos como uma alternativa para se buscar desempenho na área de sismologia ativa sem passar pela alteração estrutural nos arquivos. Entretanto, em sua maioria, eles apresentam contribuições genéricas o bastante para também serem aplicadas às estruturas de arquivos da sismologia passiva. Mesmo na definição de formatos de arquivo modernos, como o ASDF [Krischer et al. 2016], aspectos relacionados ao sistema computacional subjacente, como os que abordamos neste trabalho, não costumam ser tratados.

## 7. Conclusões

Este trabalho definiu 3 estruturas de arquivos que generalizam os formatos mais populares utilizados na sismologia para a manipulação de dados sísmicos. A partir disso, foi realizada uma análise teórica do desempenho de leitura de dados de cada estrutura, que indicou uma possível redução de até 193 vezes na quantidade de dados lidos em um caso de uso típico da sismologia. Foi realizada também, uma análise experimental do desempenho de leitura com diferentes formas de armazenamento disponíveis na nuvem computacional. Tal análise demonstrou que é possível executar a leitura de dados, ainda para um caso de uso típico, até 139 vezes mais rapidamente a depender da estrutura empregada e da tecnologia de armazenamento escolhida.

Dentre as estruturas avaliadas, a estrutura *Header First Indexed* (HFI) é mais eficiente para a leitura de dados, tendo apresentado tempos de leitura uma ordem de grandeza menores do que com a utilização da estrutura *Header Sample* (HS). Esta, por sua vez, também apresenta ganho em torno de uma ordem de grandeza sobre a estrutura *Header First* (HF). Embora já fosse esperada tal influência da organização interna dos arquivos no desempenho da leitura de dados, os experimentos permitiram a sua quantização considerando as diferentes tecnologias de armazenamento disponível na nuvem. Demonstrou-se, por exemplo, que métodos de armazenamento local, como SDDs e HDDs, são os que sofrem maior influência da estrutura de dados escolhida, enquanto o armazenamento sobre a rede é ligeiramente menos afetado. Sobre este último, notou-se ainda que a escolha de instâncias com maior capacidade de transmissão de dados não apresenta influência significativa nos tempos de leitura dos arquivos.

Os resultados apresentados dão subsídio ao desenvolvimento de novos formatos para manipulação de dados sísmicos, bem como à melhoria dos já existentes. A área da sismologia sofre de padrões legados que, como também demonstrado neste artigo, não se adaptam bem às tecnologias de armazenamento modernas. Mudanças, entretanto, não seriam inéditas, como demonstra a literatura da sismologia passiva [Ahern et al. 2007]. Pretende-se, como trabalho futuro: a definição de um modelo de desempenho derivado da análise teórica apresentada; a extensão dos experimentos realizados para mais tamanhos de arquivos e para diferentes tamanhos de páginas de memória; a reprodução das análises em outras arquiteturas computacionais de interesse; e, por fim, a identificação ou criação de formatos de arquivo para dados sísmicos que melhor considerem os aspectos funcionais dos sistemas computacionais modernos e da nuvem computacional.

## Referências

Ahern, T., Casey, R., Barnes, D., Benson, R., and Knight, T. (2007). Seed standard for the exchange of earthquake data reference manual format version 2.4. *Incorporated*

- Research Institutions for Seismology (IRIS), Seattle.*
- AWS (2016). *AWS Storage Services Overview - Whitepaper*. Acesso em: 05/08/2019.
- AWS (2019). *Amazon Web Service (AWS)*. Acesso em: 05/08/2019.
- Barry, K., Cavers, D., and Kneale, C. (1975). Recommended standards for digital tape formats. *Geophysics*, 40(2):344–352.
- Brandsberg-Dahl, S. (2017). *High-performance computing for seismic imaging; from shoestrings to the cloud*, pages 5273–5277.
- Claerbout, J. (1991). Introduction to seplib and sep utility software. *SEP-70: Stanford Exploration Project*, 413:436.
- Fomel, S., Sava, P., Vlad, I., Liu, Y., and Bashkardin, V. (2013). Madagascar: Open-source software project for multidimensional data analysis and reproducible computational experiments. *Journal of Open Research Software*.
- IRIS/PASSCAL Data Group (2012). Introduction to Active Source Data Archiving Utilizing PH5 as the Archive Format. Technical Report 2012336, IRIS/PASSCAL Instrument Center.
- J. Anderson, W.E. Farrell, K. G. J. G. H. S. (1990). Center for Seismic Studies version 3 database: Schema reference manual. Technical Report C90-01, Science Applications International Corp, Center for Seismic Studies.
- John W. Stockwell, J. and Cohen, J. K. (2008). *The New SU User's Manual*. Último acesso em: 05/08/2019.
- Koziol, Q. and Editor Padua, D. (2011). *HDF5*, pages 827–833. Springer US, Boston, MA.
- Krischer, L., Smith, J., Lei, W., Lefebvre, M., Ruan, Y., de Andrade, E. S., Podhorszki, N., Bozdağ, E., and Tromp, J. (2016). An Adaptable Seismic Data Format. *Geophysical Journal International*.
- Li, Huailiang, T. X. S. T. H. M. J. e. C. J. (2017). *An improved lossless group compression algorithm for seismic data in SEG-Y and MiniSEEDfile formats*. *Computers and Geosciences*, pages 41 – 45.
- Mayne, W. H. (1962). Common reflection point horizontal data stacking techniques. *Geophysics*, 27(6):927–938.
- Nickerson, Bradford G., J. P. A. e. M. L. A. (1999). *Data structures for fast searching of SEG-Y seismic data*. *Computers and Geosciences*, pages 179 – 190.
- Open University (2011). *File:Seismic acquisition cartoon marine.jpg*. Licença CC-BY-SA-NC. Último acesso em: 18/08/2019.
- Rubini, A. and Corbet, J. (2001). *Linux device drivers*. "O'Reilly Media".
- Samet, H. (1984). The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)*.
- Tarantola, A. (1984). Inversion of seismic reflection data in the acoustic approximation. *Geophysics*, 49(8):1259–1266.
- Yang, X., McLaughlin, K., and North, R. (2000). User's Guide to the CMR Seismic/Hydroacoustic/Infrasonic Data Products.