UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA
DEPARTAMENTO DE MATEMÁTICA APLICADA

Alcides Goldoni Junior

Aplicação e Comparação dos Sistemas de Arquivo Ext 4 com XFS e ZFS em Processamentos Sísmicos

Campinas
13/07/2022

Alcides Goldoni Junior

Aplicação e Comparação dos Sistemas de Arquivo Ext 4 com XFS e ZFS em Processamentos Sísmicos

Monografia apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos para obtenção de créditos na disciplina Projeto Supervisionado, sob a orientação do Pesquisador Dr. Jorge Henrique Faccipieri Junior.

Resumo

Os sistemas de arquivos são parte importante da computação de alto desempenho, sendo este responsável por um dos aspectos mais básicos da informática, a organização e manipulação de dados. Nesse sentido, otimizações são realizadas para evitar que uma simples busca e ou manipulação de dados provoque gargalos de desempenho ou inconsistências nos dados. Sendo assim, sistemas de arquivos mais complexos, escaláveis e ágeis estão em constante desenvolvimento e otimizações.

Este trabalho teve como objetivo avaliar um algoritmo criado pelos funcionários do *High-Perfomance Geophysics Lab* (HPG Lab), sediado no Centro de Estudos de Energia e Petróleo (Cepetro), em sistemas de arquivos mais modernos visando encontrar potenciais melhorias em custo-benefício em relação ao tempo de processamento, tamanho dos arquivos armazenados e impacto qualidade do resultado final. A técnica utilizada foi o Empilhamento *Common Reflection Point* (CRP) que estima atributos cinemáticos relacionados a inclinações e velocidades relativos a eventos sísmicos presentes no dado analisado. Foram considerados dois dados sísmicos reais, um da Bacia do Tacutu e outra da Bacia do Jequitinhonha. Tais dados foram armazenados em RAID 1 (Espelhamento). Quanto aos sistemas de arquivos considerados nesse estudo, temos: Ext 4, padrão na grande maioria dos sistemas operacionais GNU/Linux; XFS ("X"File System) e ZFS (Zettabyte File System).

Nesse estudo não foi observado nenhum ganho significativo para a técnica de processamento avaliada. Com respeito aos resultados obtidos, dados de saída, todos os resultados obtidos tiveram aproximadamente o mesmo tamanho e pouca diferença quantitativa em relação a referência. Em conclusão, a técnica de Empilhamento CRP desenvolvido pelo HPG Lab foi indiferente ao uso de qualquer sistemas de arquivos considerados nesse estudo.

Abstract

File systems are an important part of high-performance computing, being responsible for the most basic concepts of informatics which is organizing and manipulating of data. In this sense, optimizations can be done in order to avoid that a simple search and/or data manipulations cause bottlenecks in performance, data loss or corruption. Thus, file systems are in constant development to ensure scalabilty and performance.

The objective of this work is to evaluate an algorithm created by the High-Performance Geophysics Lab (HPG Lab), based at the Centro de Estudos de Energia e Petróleo (Cepetro), considering modern file systems in order to find the best cost-benefit in terms of processing time, size and quality of the final data. The technique considered was the Common Reflection Point (CRP) Stacking which estimates cinematic attributes related to slopes and velocities of seismic events. Two real seismic datasets were considered, one from the Tacutu basin and another from Jequitinhonha basin, both stored on RAID 1 (mirroring). The file systems considered were: Ext 4, standard for most GNU/Linux distributions; XFS ("X"File System) and ZFS (Zettabyte File System).

In conclusion, we could not find any significant gain in terms of execution time, file size and quality of the obtained results by changing the file systems to XFS and ZFS.

Conteúdo

1 Introdução

Nas últimas décadas ocorreram significativos avanços computacionais e com eles, o volume de dados criado diariamente aumentou de maneira expressiva. Atualmente, todos temos um pequeno computador nos bolsos, gerando e compartilhando cada vez mais informações. Para trabalhar de maneira eficiente com esse grande volume de dados, foram desenvolvidas diversas soluções em HPC (*High-Performance Computing*). Tais avanços ocorrem em relação ao paralelismo de processamento nas CPU's (*Central Processor Unit*) e em GPU's (*Graphics Processing Unit*). Além disso, os dispositivos de armazenamento também ficaram mais rápidos e com maiores capacidades como os SSD's (*Solid State Drive*) e os NVMe (*Non-Volatile Memory Express*), que hoje podem ser encontrados com capacidades da ordem de *Terabytes*. Estes avanços fizeram com que a computação de alto desempenho venha sendo cada vez mais utilizada nas mais diversas aplicações, como *Big Data* e *Machine Learning*, entre outros. Essas aplicações demandam grande quantidade de leitura e escrita de dados que ficam organizados em diretórios que podem estar no computador local ou não. Dessa forma, é evidente que a organização das estruturas de dados nessas unidades de armazenamento influenciam na busca de arquivos, como por exemplo quando troca-se o um dispositivo mecânico como um HD (*Hard Disk*) por um SSD ou NVMe.

A maneira na qual estes dados são armazenados e organizados em arquivos e diretórios é responsabilidade do sistema operacional e feita através do sistema de arquivos. Cada sistema operacional pode utilizar um ou vários tipos de sistema de arquivos, por exemplo em sistemas baseados em Windows, temos o padrão NTFS (*New Techonology File System*) (Microsoft, 2009). Já em sistemas baseados em GNU/Linux temos como padrão, na grande maioria das distribuições, o Ext 4 (4^o *Extended File System*) (Kernel, 2019). No entanto, existem vários outros tipos como FAT (*File Allocation Table*) (Microsoft, 2008), XFS (*"X" File System*) (RedHat, 1999), BeeGFS (Thinkparq, 2019), ZFS (*Zettabyte File System*) (Oracle, 2010) entre outros, cada um tendo uma característica própria que melhora o gerenciamento, armazenamento e recuperação de dados.

Nesse sentido, esse trabalho tem como objetivo comparar o sistema de arquivo Ext4 com os sistemas XFS e ZFS através da aplicação de uma técnica empilhamento

de dados sísmicos usada em uma das etapas envolvidas no processamento de um dado sísmico. O processamento sísmico compõe um grande conjunto de ferramentas, aplicadas em sequência para preparar os dados sísmicos para posterior interpretação.

2 Metodologia

Com base em uma revisão de bibliográfica, foi encontrado um trabalho que realizou a comparação entre os sistemas de arquivo Ext 3 e o XFS. Nesse trabalho, verificou-se que o XFS apresentou desempenho superior de leitura e escrita para tarefas single-thread e multi-thread utilizando o sotware de benchmark IOzone, Meiling (2017). Hoje o Ext 3 ainda pode ser encontrado, porém, o padrão que diversas distribuições GNU/LINUX utilizam é sua versão melhorada, o Ext 4. Dessa forma, nesse estudo foi considerada essa versão mais atual do sistema de arquivos na comparação com o XFS e com o ZFS. Para essa comparação, foi utiliza uma técnica de processamento sísmico, desenvolvido no *High-Performance Geophysics Lab* (HPG Lab), sediado no Centro de Estudos de Energia e Petróleo (Cepetro) da Universidade Estadual de Campinas (Unicamp). Essa técnica foi aplicada em dados sísmicos reais, públicos, da Bacia do Tacutu, de aproximadamente 75 MB, e da Bacia do Jequitinhonha, com aproximadamente 785MB.

Foram realizados comparações entre esses sistemas de arquivo de modo a avaliar potenciais ganhos em relação ao tempo de execução, ao tamanho desses dados e se há alterações na qualidade do resultado final depois da aplicação da técnica. Para isso, foram utilizados dois computadores um como nó de processamento e um servidor de arquivos, com as seguintes configurações:

Nó de processamento:

- Processador Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz
- Nvidia TITAN X (Pascal)
- 16 GB de memória RAM
- Sistema Operacional Ubuntu 16.05.6 LTS

Servidor de arquivos:

- Processador Intel(R) Xeon(R) CPU E5-2640 v2 @ 2.00GHz
- 94 GB de memória RAM
- 2 x 2TB Seagate Barracuda 7200rpm em RAID 1
- Sistema Operacional CentOS 7.9

É importante mencionar que a conexão entre o nó de processamento e o servidor de arquivos foi feito via cabo *gigabit ethernet* de 1Gbps. Além disso, a maneira com que o nó de processamento acessa o disco no servidor de arquivos foi feita via NFS (*Network File System*) (RedHat, 2010). Com isso, o nó de processamento pode acessar o dado como se ele estivesse localmente no computador.

Em relação aos discos, estes foram configurados em RAID 1 (espelhamento), ou seja, os dados são replicados nas duas unidades de disco para tolerar eventuais falhas em uma das unidades, não conferindo nenhum ganho de desempenho.

Nosso objetivo é analisar o desempenho desses diferentes tipo de sistemas de arquivo, coletando os seguintes dados: Tempo de execução, taxa de leitura e escrita durante a execução da técnica e a qualidade do resultado final em relação a uma referência. O tempo de execução foi calculado para um conjunto de 3 execuções, das quais foram computadas médias. A taxa de leitura foi monitorada utilizando-se de uma ferramenta *open source*, iostat (IBM, 2022). Com relação os tamanhos dos arquivos, foi utilizado como referência o dado no sistema de arquivos Ext 4.

2.1 O Empilhamento CRP

O desempenho dos sistemas de arquivos considerados nesse trabalho foram avaliados durante a execução de uma técnica de processamento sísmico desenvolvida no HPG Lab do Cepetro. Essa técnica, denominada Empilhamento *Common Reflection Point* (CRP), é capaz de atenuar ruídos nas amplitudes de dados sísmicos preservando sua geometria original (posição de fontes e receptores). Mais detalhes sobre a técnica podem ser encontrados em Coimbra (2014), porém uma breve explicação do método e seu contexto de aplicação é apresentada a seguir.

O método CRP tem como dado de entrada um dado sísmico, este é composto pelo registro da resposta do meio geológico (em diversos receptores) para a detonação de uma fonte sísmica. Para cada detonação da fonte sísmica, temos um conjunto de diversos registros em tempo, para cada receptor, chamados traços, cujas posições são conhecidas. Nesse volume de dados, compostos de traços, podemos avaliar o ajuste de possíveis superfícies de tempos de trânsito (tempos de resposta) às amplitudes presentes no dado sísmico através de uma medida de coerência, a Semblance (Neidell, 1971). No Empilhamento CRP, essa superfície de tempo de trânsito é parametrizada por dois parâmetros, um relacionado a velocidade média do meio geológico (V), e outro associado as inclinações dos eventos sísmicos (A) presentes no dado. Quando as amplitudes do dado se ajustam a superfície CRP para determinados valores de A e V (isto é, temos um valor alto de coerência) isso indica que temos um evento sísmico naquela região e sendo assim, esses valores são registrados ou salvos para aquela posição do dado sísmico. Os valores de A e V são testados utilizando-se uma heurística de busca (*Differential Evolution*) (Barros and Lopes, 2015).

Uma vez que a aplicação do Empilhamento CRP envolve passos de leitura, para avaliação das amplitudes do dado, e de escrita, para o registro dos parâmetros e valores de coerência medidos, o objetivo do trabalho foi verificar essas etapas teriam alguma influência do sistema de arquivos utilizado.

2.2 RAID

O RAID (*Redundant Array of Independent Disks*) (Gupta, 2002) é uma tecnologia que utiliza recursos de vários discos combinados de forma que estes possam ser manipulados como se fossem apenas um único dispositivo, facilitando assim, a utilização e gerenciamento do mesmo pelo o usuário.

Existem vários tipos de RAID com diferentes funcionalidades. Nesse trabalho, faremos a utilização do RAID 1, também conhecido como espelhamento (ou *mirroring*). Para isso, são necessários pelo menos duas unidades de disco onde os dados serão espelhados, ou seja replicados, em ambos os discos. Dessa forma, em caso de perda de um dos discos, não se corre o risco de perder o dado processado ou em processamento (Intel, 2017).

2.3 Sistemas de arquivos

Os sistemas de arquivos são os responsáveis por organizar, armazenar e permitir acesso aos dados registrados em um ou mais discos de forma efetiva e integra. Os sistemas de arquivo podem possuir formas distintas de realizar esses processos, sendo mais ou menos eficientes dependendo da aplicação executada e também do objetivo visado em seu desenvolvimento, por exemplo, um sistema que faz a criptografia em seus arquivos dificilmente será tão rápido quanto um sistema de arquivos voltado a computação de alto desempenho. Porém, todos eles tem o mesmo objetivo de permitir o acesso via *software* aos dados gravados no *hardware*.

2.3.1 Ext 4

O Ext 4 (*4º Extended File System*) é a versão estável de sistema de arquivo utilizada desde a versão 2.6.28 do kernel do GNU/Linux. Em muitos casos, esse é o sistema de arquivos padrão das distribuições, para mais informações ver Kernel (2019). As características relevantes para esse trabalho são:

- Capacidade de armazenamento: Seu antecessor Ext 3 suportava um tamanho máximo de 16TB, o Ext 4 suporta 1024 PB (*Petabytes*) e arquivos de até 16 TB.
- Alocação múltipla de blocos: Ao invés de alocar um bloco de 4KB por vez, ele aloca múltiplos blocos de 4KB evitando múltiplas chamadas para uma mesma gravação de dados, como forma de evitar sobrecargas.
- Atraso na alocação: O sistema mantém o dado na memória cache o maior tempo possível de forma com que mesmo que o aquivo cresça, ele só será gravado no disco quando for necessário. Trata-se de uma forma de evitar fragmentação e melhorar a carga de trabalho.
- Persistência na Pré-alocação: Evita, por exemplo, que uma aplicação fique sem espaço durante um processo de gravação.

2.3.2 XFS

O XFS (*"X" File System*) é um sistema de arquivo que suporta arquivos de larga escala em um único computador, tem como vantagem ser escalável (ou seja, capaz de manter o desempenho independente do tamanho do dado em operação), robusto e de alto desempenho. O XFS já validado em testes com computação de larga escala, ou seja, com múltiplas CPU's, múltiplos periféricos e com carga de trabalho de leitura e escrita multi-tarefas, apresentando bom desempenho nessas condições RedHat (1999). Atualmente, a partir do Red Hat Enterprise Linux 8, é o sistema de arquivo padrão para esse sistema. O XFS teve seu desenvolvimento iniciado nos anos 90 pela *Silicon Graphics*, empresa que foi comprada pela *Hewlett Packard Enterprise* (HPE) no ano de 2016, a qual tem histórico de trabalhar em servidores com armazenamento de larga escala (RedHat, 1999). As principais características desse sistema de arquivo para este trabalho são:

- Escalabilidade: Tamanho do sistema de arquivo suportado até 1024 TB com capacidade de suportar grande número de operações simultâneas.
- Desempenho: Atua com boa performance para sistemas com altas cargas de trabalho (CPU's e periféricos).
- Atraso na alocação: O sistema mantém o dado na memória cache o maior tempo possível de forma com que mesmo que o aquivo cresça, ele só será gravado no disco quando for necessário. Trata-se de uma forma de evitar fragmentação e melhorar a carga de trabalho.
- Persistência na Pré-alocação: Evita, por exemplo, que uma aplicação fique sem espaço durante um processo de gravação.

2.3.3 ZFS

O ZFS (*Zettabyte File System*) apresenta uma ideia diferente na forma com que os dados são armazenados. Ele introduz o conceito de *storage pool*, uma agregação de capacidade de discos físicos . Historicamente, os sistemas de arquivos eram construídos sob um único dispositivo físico. Para considerar múltiplos dispositivos e também garantir redundância, foi então introduzido o conceito de *volume manager*, de forma que ele

representava um único dispositivo. Isso demanda modificações nos sistemas de arquivos, inserindo mais uma camada de complexidade, visto que, o sistema de arquivo não tinha controle sobre o dispositivo físicos dos dados do *volume manager*.

A diferença do ZFS em relação a outros sistemas de arquivos considerados nesse trabalho é que ele não trabalha com o gerenciamento de volumes. O ZFS simplesmente elimina essa etapa, agregando os novos dispositivos em um *pool*. Dessa forma, o sistema de arquivo não está mais restrito a dispositivos individuais, permitindo que se compartilhe o sistema de arquivo entre todos os dispositivos do *pool*. Para mais detalhes sobre esses conceitos ver Oracle (2010). Esse sistema de arquivo tem como principais características:

- Escalabilidade: Suporte na ordem do Zettabytes de armazenamento.
- Redundância de dados: Uma forma de garantir redundância é através de um *snapshot* que é uma cópia com permissão apenas de leitura que pode ser criada de maneira fácil e rápida, não gerando dados adicionais em disco.

3 Análise

Com base nas informações coletadas, os dados foram separados em dois pontos principais. O primeiro, considera o tempo de execução ou processamento da aplicação, verificando se algum dos sistemas de arquivos seria capaz de apresentar alguma redução. O segundo diz respeito ao dado de saída, verificando a qualidade e o tamanho do mesmo em relação ao dado de saída no sistema de arquivos de referência, o Ext 4.

3.1 Tempo de processamento

Durante o processamento do dado é esperado que o disco seja usado durante todo o tempo do processamento, tanto para leitura quanto para escrita, como pode ser visto nas Figuras 1 a 6. Com base nessas informações, referentes as taxas de leitura e escrita, pode-se garantir que o dado realmente foi lido e escrito no disco e que não se encontrava totalmente em memória RAM ou memória cache.

Essas mesmas figuras mostram o comportamento das taxas de leitura e escrita, em MB/s, no disco durante todo o processamento. Cada sistema de arquivos apresenta

um comportamento diferente, uma vez que possuem diferentes características na forma de ler e escrever os dados. Um exemplo dessas diferenças são os tamanhos dos blocos de alocação. No Ext 4, esse tamanho varia de 1 a 64 KB, sendo tipicamente utilizado 4 KB. Para o XFS de 512 bytes a 64 KB e para o ZFS de 128 KB, podendo ser divido em 32 setores de 4 KB ou 256 setores de 512 bytes. Um ponto importante a ser mencionado é que todos esses sistemas de arquivo fazem alocação múltiplas de bloco, por exemplo, para alocar 100 MB, sem a alocação múltipla de blocos, o sistema operacional precisaria realizar 25 mil chamadas de 4 KB em série, executando uma a uma, o que causaria um enorme *overhead*.

Nas Figuras 1, 2, 4, 5 e 6 podemos observar que para um determinado intervalo de tempo de processamento, a taxa de escrita, representada pela linha vermelha, cresce para todos os sistemas de arquivo, mostrando mais uma vez que o disco está sendo utilizado e que os dados não ficaram em memórias, seja a RAM ou cache, tanto do nó de processamento quanto do servidor de arquivos. Sobre a taxa de leitura, representada pela linha azul, como o processamento trabalha com porções do dado de cada vez, a quantidade de dado acessada é suficiente para que o processamento aconteça sem a necessidade de aumento dessa taxa.

Já na Figura 2, o comportamento é um pouco diferente, mantendo estável durante todo o tempo de processamento, o que pode ser justificado pelos tamanho de bloco que o sistema de arquivo utiliza.

As Figuras 7 e 8 exibem a média dos tempos totais de processamento em três execuções do Empilhamento CRP para os dois dados avaliados. Nesse experimento, verificou-se pouca diferença nos tempos de execução.

Comparando os tempos, para o Tacutu, a diferença entre o Ext4 e o XFS foi nula. Para o Ext 4 e o ZFS, temos uma diferença absoluta de 19 segundos. Essa diferença representa uma perda de desempenho de aproximadamente $9,5\%$ em relação o Ext 4, que neste contexto pode ser considerada irrelevante. Já para o Jequitinhonha, a diferença foi de 75 segundos entre Ext 4 e XFS e de 193 segundos entre o Ext 4 e o ZFS, aproximadamente 0.3% e $0,79\%$, respectivamente. Novamente, podemos considerar essa diferença desprezível.

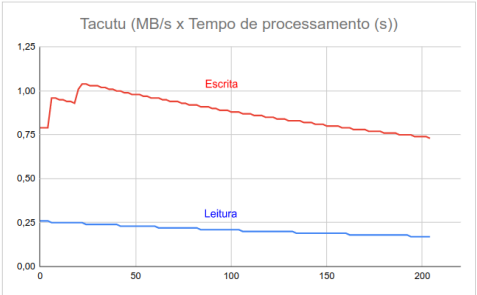

Figura 1: Taxa de processamento em Ext 4 no Tacutu.

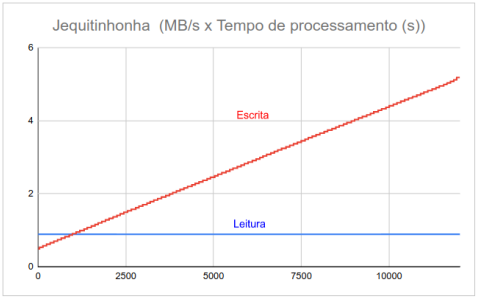

Figura 2: Taxa de processamento em Ext 4 no Jequitinhonha.

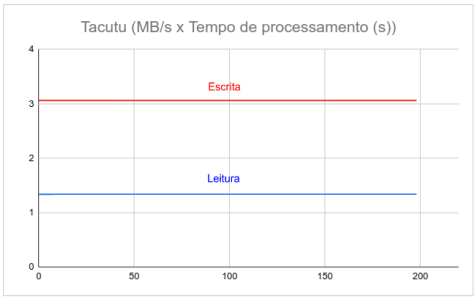

Figura 3: Taxa de processamento em XFS no Tacutu.

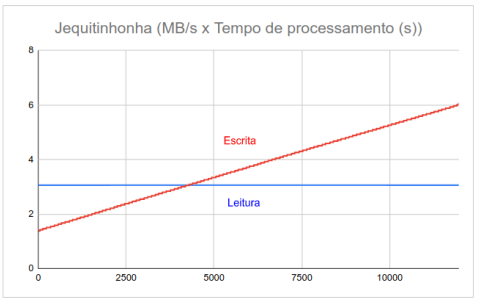

Figura 4: Taxa de processamento em XFS no Jequitinhonha.

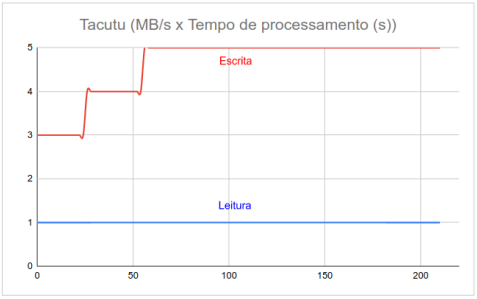

Figura 5: Taxa de processamento em ZFS no Tacutu.

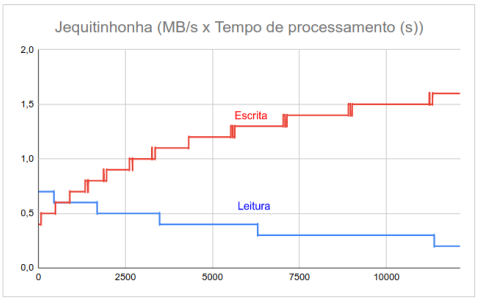

Figura 6: Taxa de processamento em ZFS no Jequitinhonha.

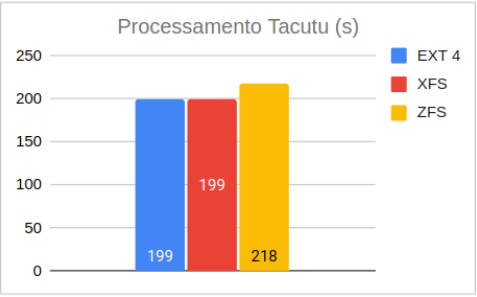

Figura 7: Tempo de processamento para o Tacutu.

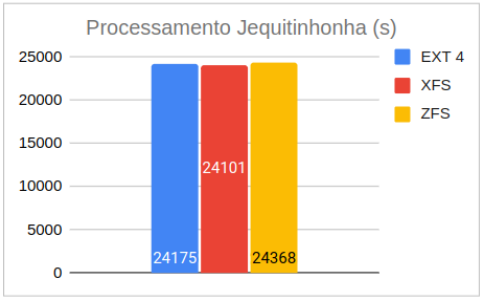

Figura 8: Tempo de processamento para o Jequitinhonha.

3.2 O dado de saída

Para fins de comparação entre os resultados obtidos pela técnica nos diferentes sistemas de arquivo, será considerado como referência o dado gerado pelo sistema de arquivo Ext 4.

Valores de erro médio e de amplitude para os resultados obtidos para o dado do Tacutu estão apresentados na Tabela 1. Note que para os valores encontrados para o parâmetro A, temos erros médios muito próximos. Agora, levando em consideração a amplitude, que representa a diferença entre o maior e o menor valor encontrado no dado de saída, temos uma pequena diferença para o caso do ZFS em relação aos outros sistemas de arquivos. Para o parâmetro V, os valores encontrados para os três sistemas de arquivos testados foram similares e representam uma diferença significativa.

Para fins ilustrativos, a Figura 9 (a), (b), (c) e (d) mostram o atributo A para diferentes subconjuntos de dado, denominados CDP's, para o sistema de arquivo Ext 4. As Figuras 10 e 11 (a), (b), (c) e (d) representam o mesmo parâmetro A para os sistemas de arquivos XFS e ZFS, respectivamente. Analogamente, a Figura 12 (a), (b), (c) e (d) mostram o atributo V para os mesmos CDP's considerados na Figura 9 para o sistema de arquivo Ext 4. As Figuras 13 e 14 (a), (b), (c) e (d) representam o mesmo parâmetro V para os sistemas de arquivos XFS e ZFS, respectivamente. Essas imagens mostram o que foi exposto na Tabela 1 de que não há diferenças significativas. As imagens foram geradas usando o *software* SU (*Seismic Unix*) (Stockwell, 1999).

	Tacutu			
	Erro médio absoluto de A	Amplitude	Erro médio absoluto V	Amplitude
Ext 4	0	1.9000E-03	0	3.0000E+03
XFS	1.3532E-04	1.9000E-03	7.8982E+02	3.0000E+03
ZFS	1.3570E-04	1.8999E-03	7.9143E+02	3.0000E+03

Tabela 1: Tabela com os valores de erro médio e amplitude para o Tacutu.

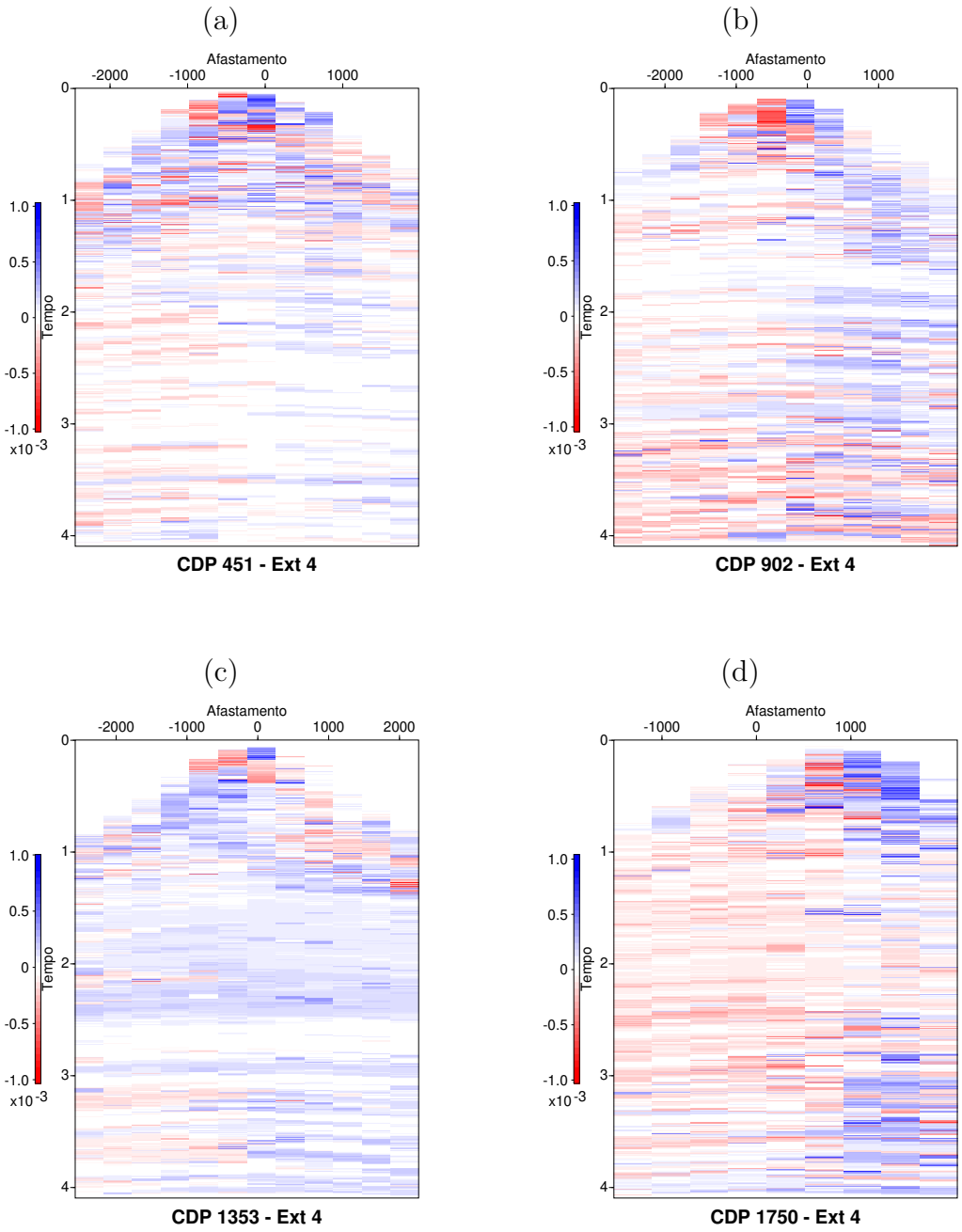

Figura 9: Dado de saída usando o sistema de arquivos Ext 4. Parâmetro A - Tacutu. CDP's (a) 451 (b) 902 (c) 1353 (d) 1750

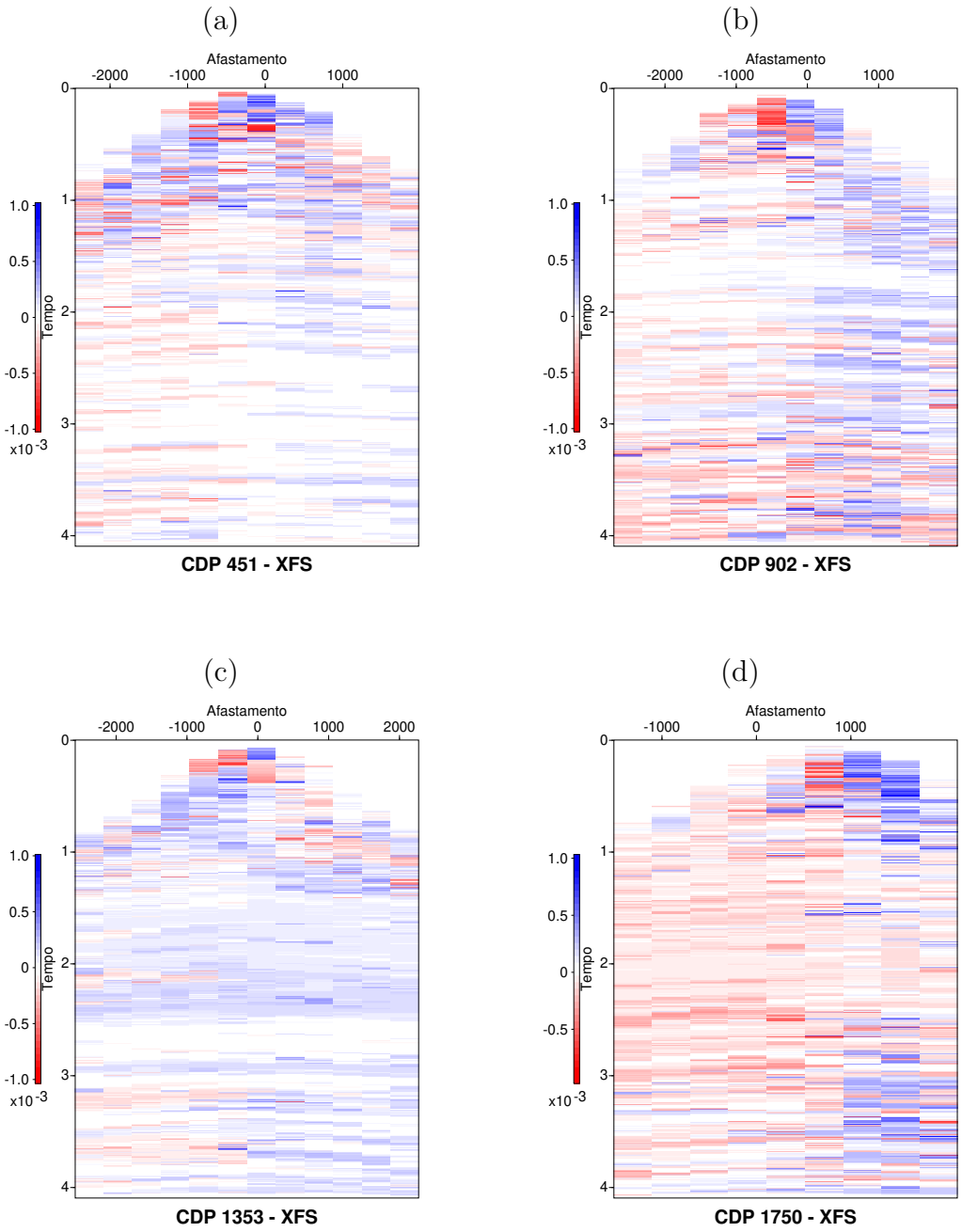

Figura 10: Dado de saída usando o sistema de arquivos XFS. Parâmetro A - Tacutu. CDP's (a) 451 (b) 902 (c) 1353 (d) 1750

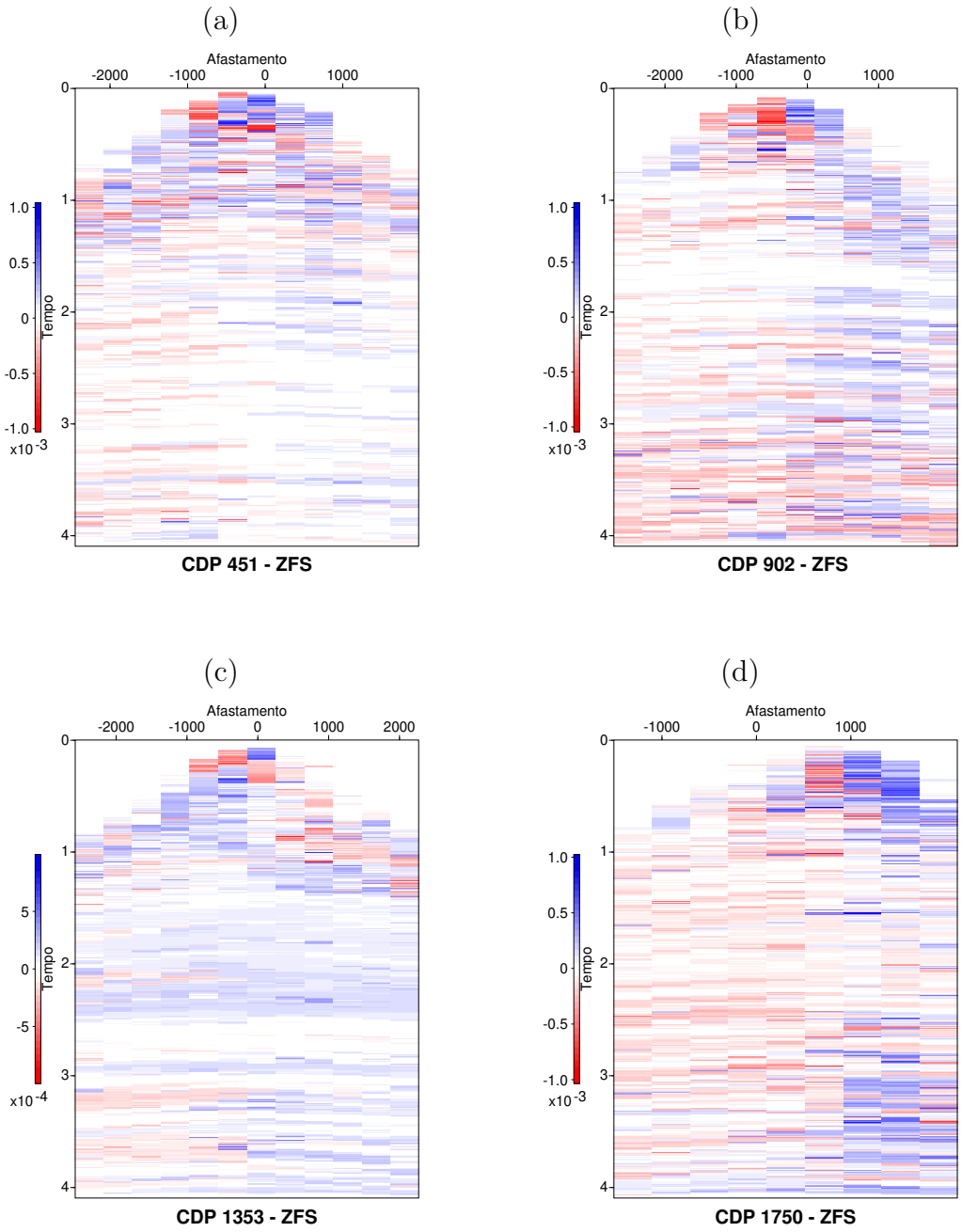

Figura 11: Dado de saída usando o sistema de arquivos ZFS. Parâmetro A - Tacutu. CDP's (a) 451 (b) 902 (c) 1353 (d) 1750

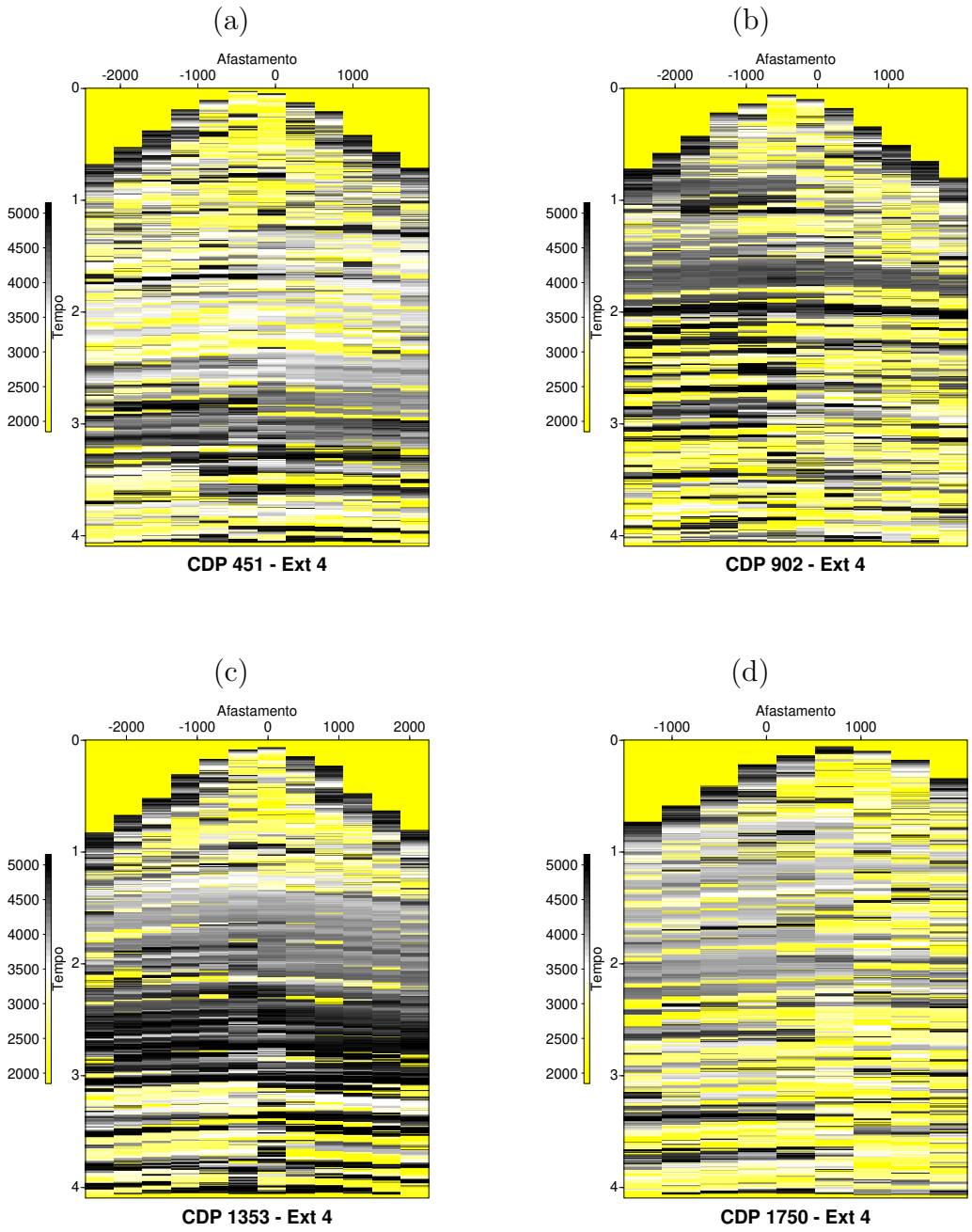

Figura 12: Dado de saída usando o sistema de arquivos Ext 4. Parâmetro V - Tacutu. CDP's (a) 451 (b) 902 (c) 1353 (d) 1750

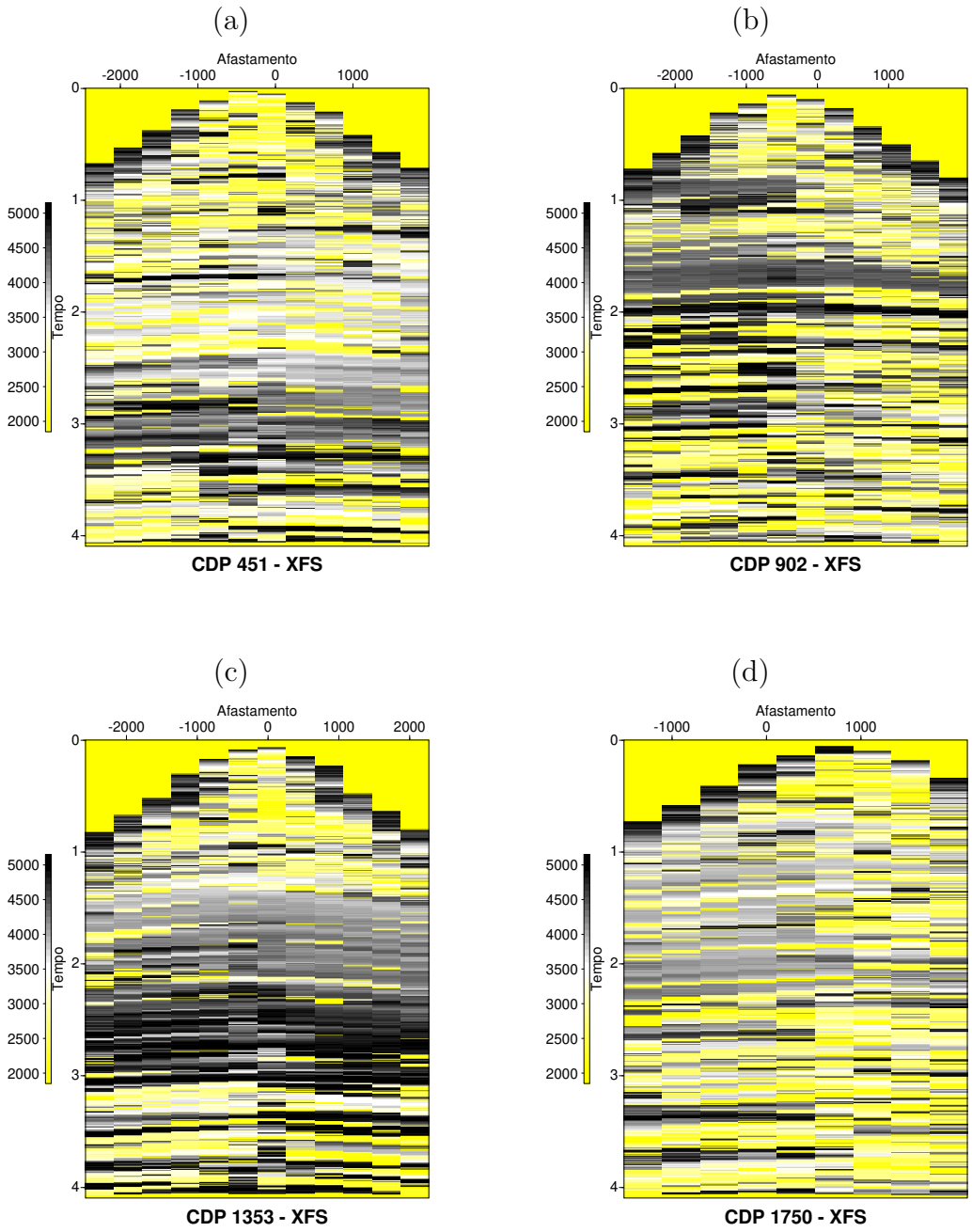

Figura 13: Dado de saída usando o sistema de arquivos XFS. Parâmetro V - Tacutu. CDP's (a) 451 (b) 902 (c) 1353 (d) 1750

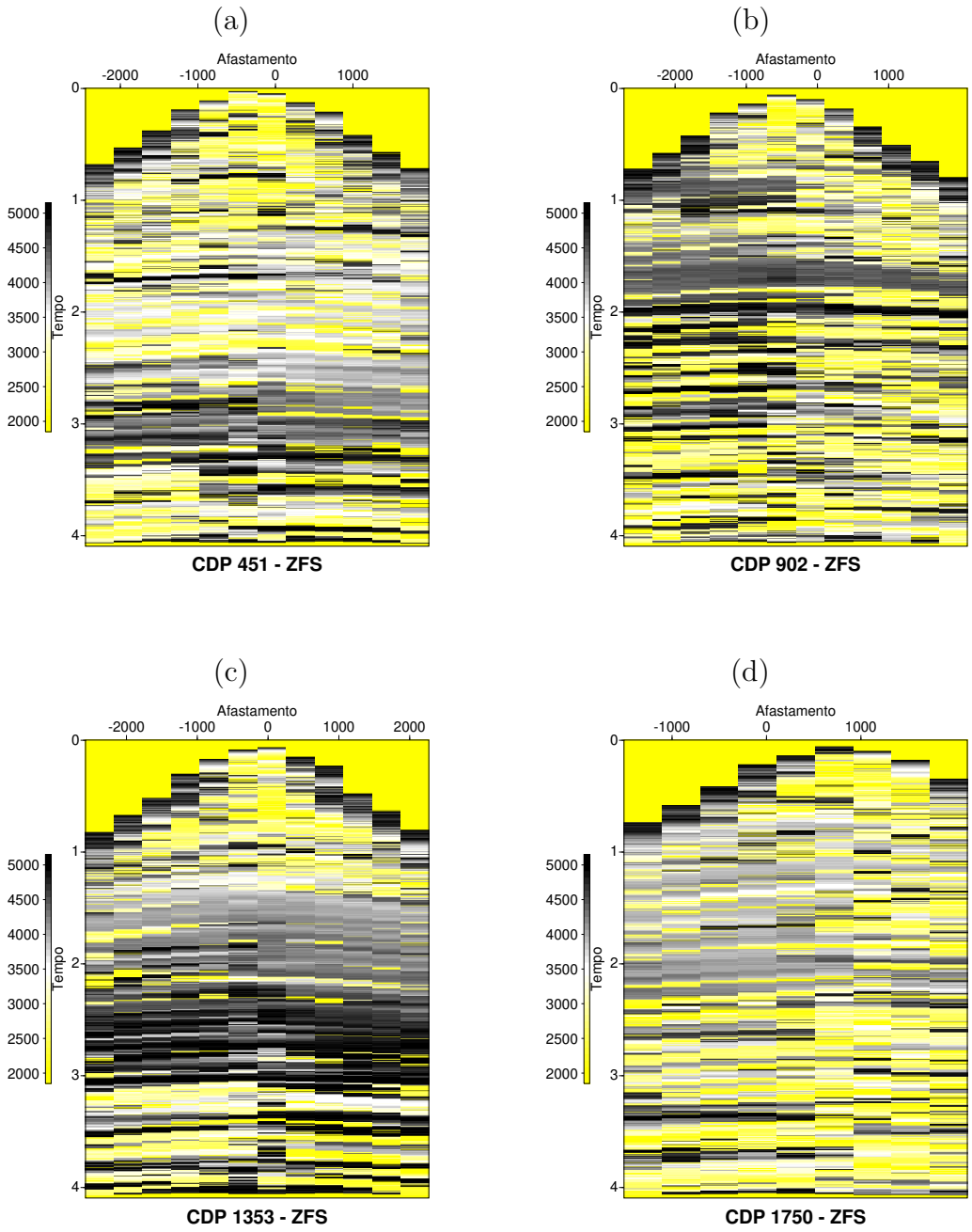

Figura 14: Dado de saída usando o sistema de arquivos ZFS. Parâmetro V - Tacutu. CDP's (a) 451 (b) 902 (c) 1353 (d) 1750

No caso do dado Jequitinhonha, os valores de erro médio e amplitude podem ser vistos na Tabela 2. Note que os valores exibidos para as amplitudes dos parâmetros A e V idênticos. As Figuras de 15 a 20 que exibem as mesmas comparações de parâmetros realizados anteriormente para o caso do Tacutu para cada sistema de arquivo. Note que mesmo com os erros, a dispersão dos valores para cada ponto é igual para os três sistemas de arquivos. Dessa forma, concluímos que também não há diferença significativa para esse dado.

A pequena diferença encontrada, tanto em relação ao erro quanto em relação a amplitude, ocorre devido a variações no conjunto de valores de inicialização da heurística de busca (*Differential Evolution*) (Barros and Lopes, 2015) utilizada pelo empilhamento CRP. Esse processo de inicialização tem caráter aleatório e não determinístico entre execuções. Um processo de busca exaustiva (*brute force*) eliminaria essas diferenças e seria a opção mais indicada para o tipo de investigação realizada nesse trabalho, porém não é uma opção viável por tornar o tempo de execução da aplicação inviável. Além disso, o objetivo deste trabalho era verificar se a versão de produção software, em seu estágio atual, seria impactada pelo sistema de artigo.

Em relação ao tamanho dos dados, a Tabela 3 mostra em Bytes o tamanho dos dados para cada sistema de arquivo. Podemos observar que para o Tacutu temos uma diferença muito sutil, i.e., em torno de 4 Bytes. Já para o Jequitinhonha todos os dados apresentam o mesmo tamanho. Isso pode ser atribuído a diferença de tamanho dos blocos alocados em cada sistema de arquivos, visto que esse tamanho varia de de 1 a 64 KB, sendo tipicamente utilizado 4KB para o Ext 4; de 512 bytes a 64KB para o XFS e de 128KB, podendo ser divido em 32 setores de 4 KB ou 256 setores de 512 bytes.

Jequitinhonha				
	Erro médio absoluto de A	Amplitude	Erro médio absoluto V	Amplitude
Ext 4	0	1.9000E-03	0	3.0000E+03
XFS	6.4255E-05	1.9000E-03	2.2125E+02	3.0000E+03
ZFS	6.3024E-05	1.9000E-03	2.1551E+02	3.0000E+03

Tabela 2: Tabela com os valores de erro médio e amplitude para o Jequitinhonha.

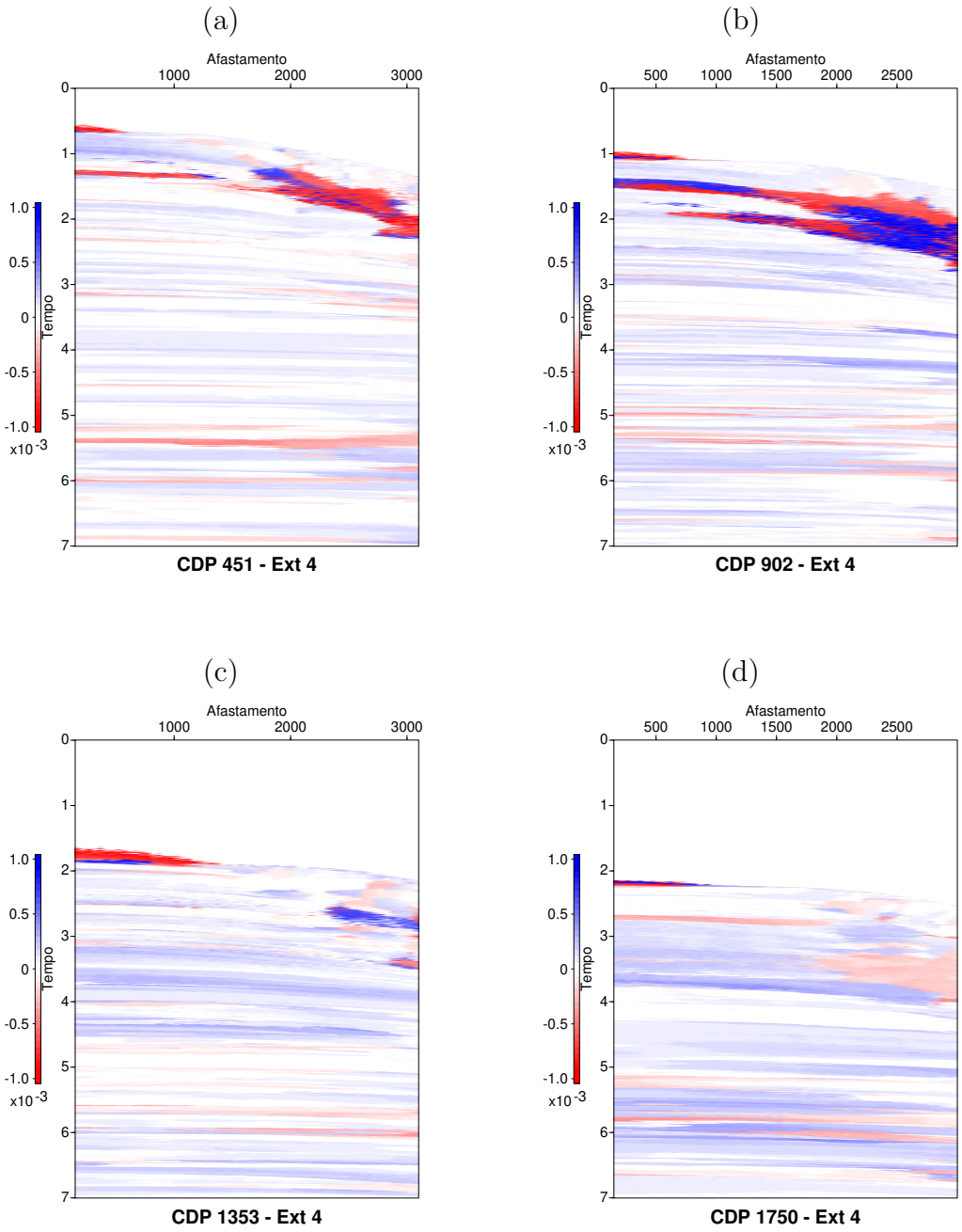

Figura 15: Dado de saída usando o sistema de arquivos Ext 4. Parâmetro A - Jequitinhonha. CDP's (a) 451 (b) 902 (c) 1353 (d) 1750

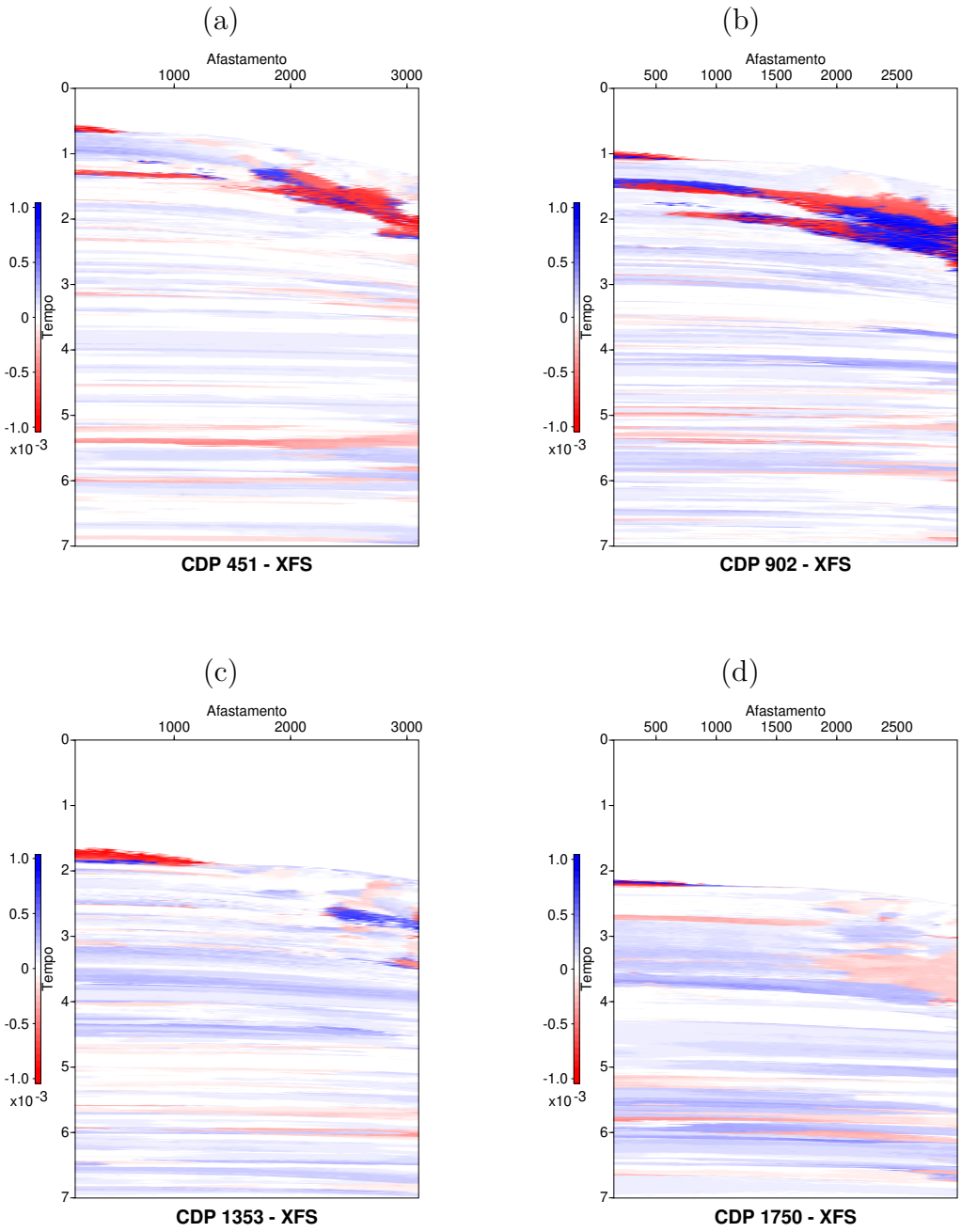

Figura 16: Dado de saída usando o sistema de arquivos XFS. Parâmetro A - Jequitinhonha. CDP's (a) 451 (b) 902 (c) 1353 (d) 1750

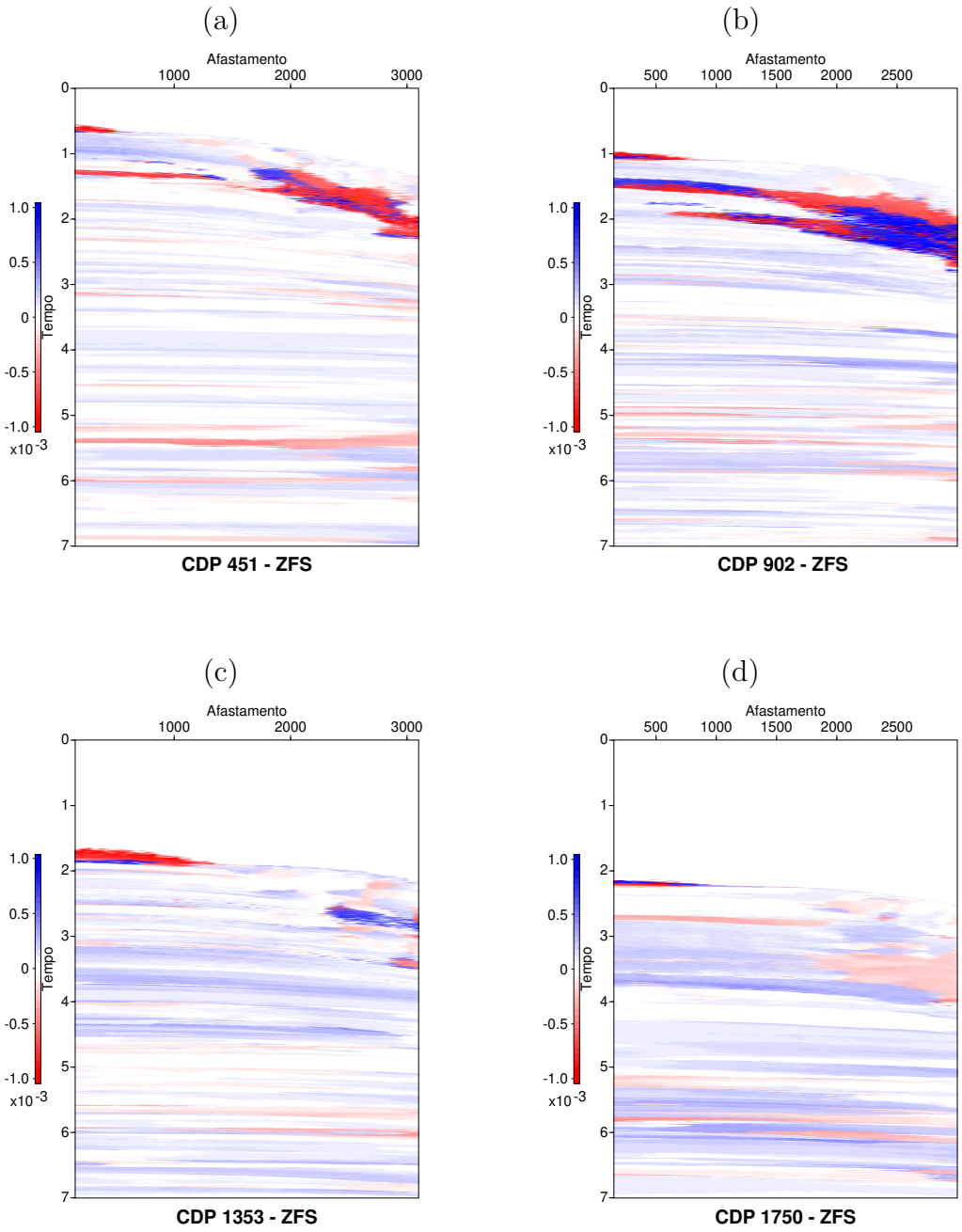

Figura 17: Dado de saída usando o sistema de arquivos ZFS. Parâmetro A - Jequitinhonha. CDP's (a) 451 (b) 902 (c) 1353 (d) 1750

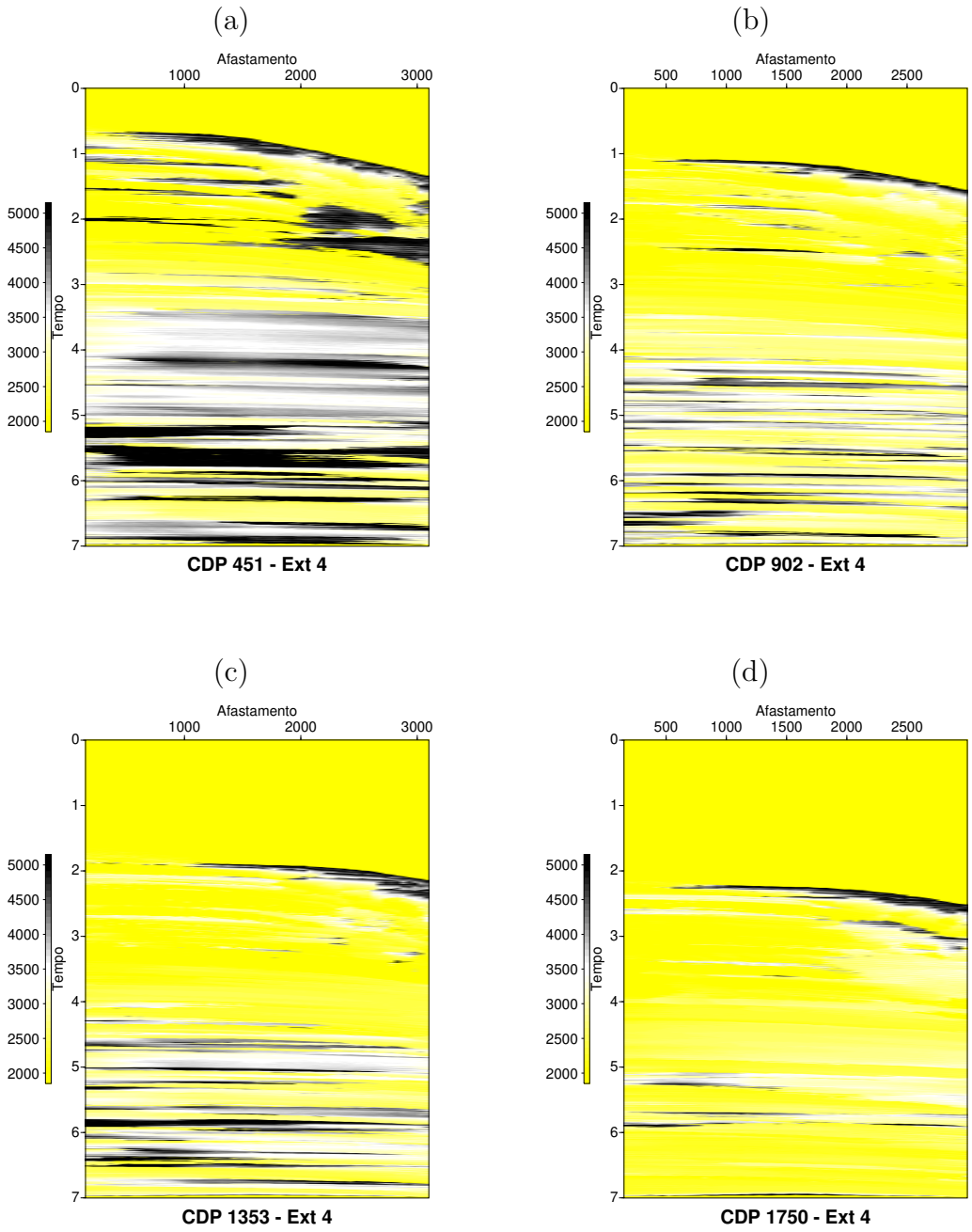

Figura 18: Dado de saída usando o sistema de arquivos Ext 4. Parâmetro V - Jequitinhonha. CDP's (a) 451 (b) 902 (c) 1353 (d) 1750

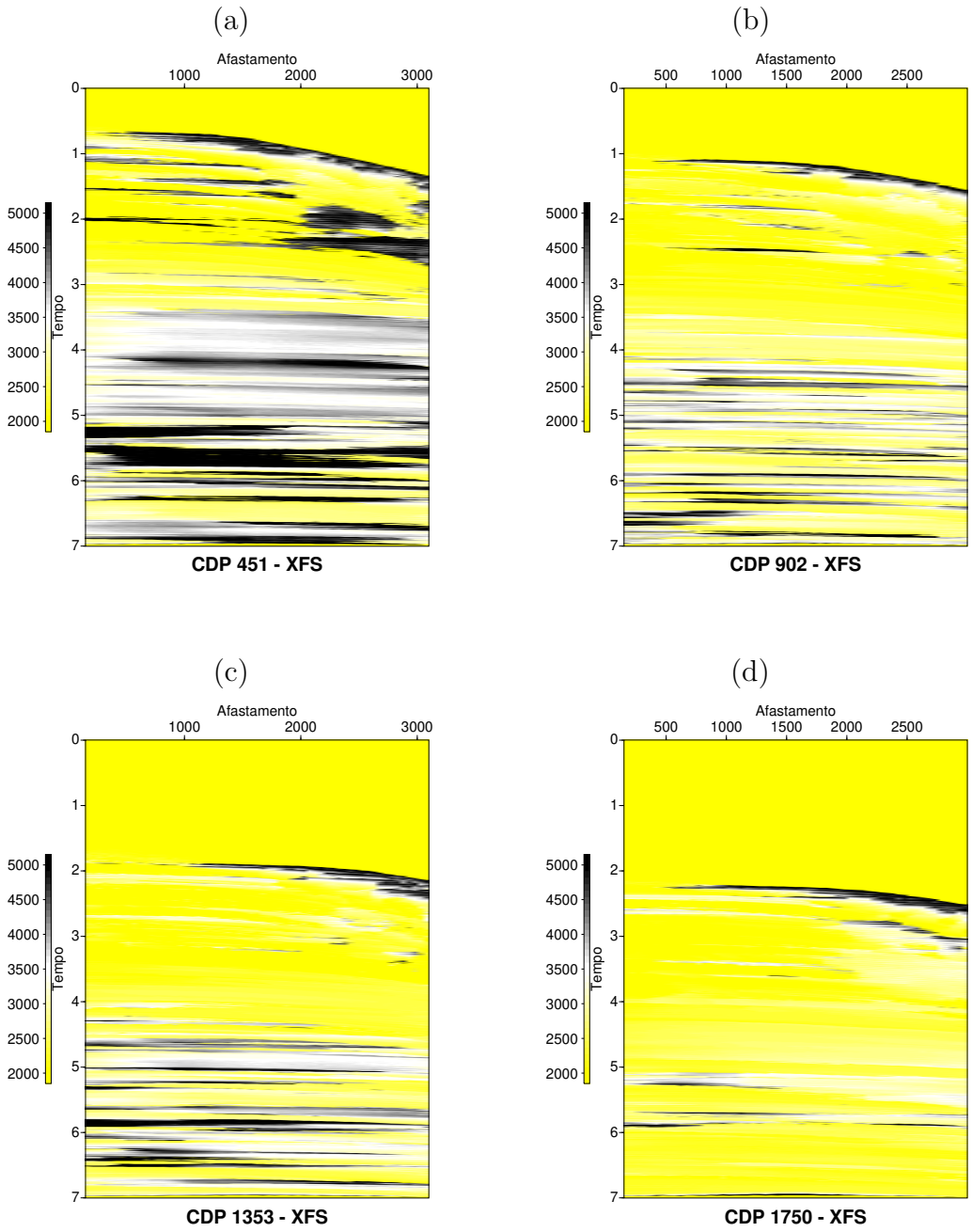

Figura 19: Dado de saída usando o sistema de arquivos XFS. Parâmetro V - Jequitinhonha. CDP's (a) 451 (b) 902 (c) 1353 (d) 1750

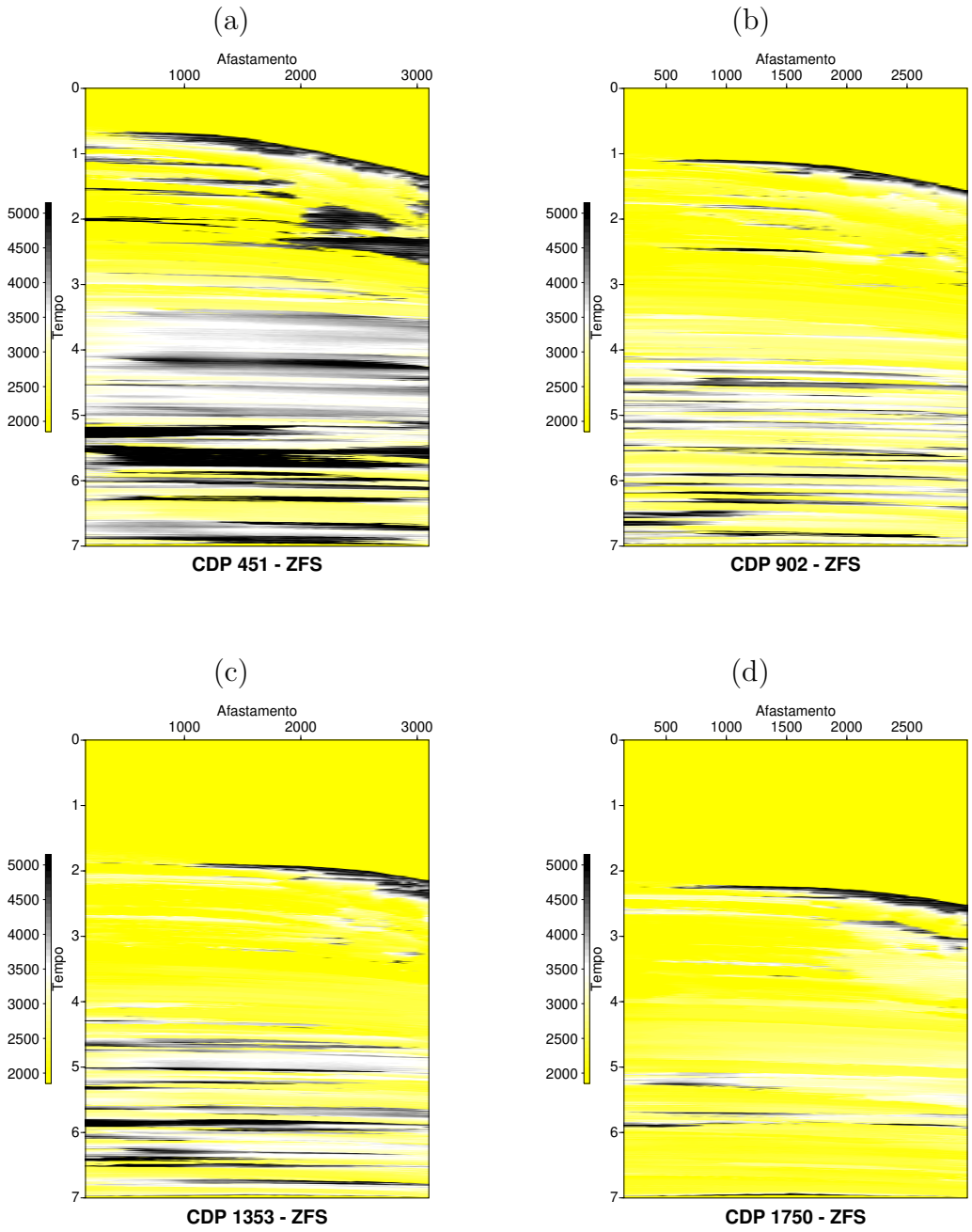

Figura 20: Dado de saída usando o sistema de arquivos ZFS. Parâmetro V - Jequitinhonha. CDP's (a) 451 (b) 902 (c) 1353 (d) 1750

	Tacutu (em Bytes)			Jequitinhona (em Bytes)		
	Entrada	Saída A	Saída V	Entrada	Saída A	Saída V
Ext 4	76432	76432	76432	803388	803388	803388
XFS	76428	76428	76428	803388	803388	803388
ZFS	76428	76432	76428	803388	803388	803388

Tabela 3: Tamanhos dos arquivos para os diferentes sistemas (em Bytes).

4 Conclusão

Neste estudo uma técnica de processamento sísmico foi aplicada a dois dados sísmicos reais. Durante sua execução foram coletados dados referentes ao comportamento dos discos no servidor de arquivos e estes foram comparados com o tempo total de processamento, tomando como referência o sistemas de arquivos mais comumente encontrado nas distribuições GNU/Linux, o Ext 4 com o XFS e ZFS. Após o processamento, os dados de saída foram comparados em relação a sua qualidade e tamanho em disco conforme apresentado nas seções anteriores. A partir da análise dos dados coletados, foi possível concluir que o tempo de processamento teve diferenças desprezíveis, mesmo havendo muita leitura e escrita durante todo o processamento sísmico. Dessa forma, concluímos que os sistemas de arquivos não tiveram uma influencia significativa. Em relação ao dado final obtido, grande parte dos valores encontrados foram similares, visto que os parâmetros não foram buscados por força bruta e sim por heurística não-determinística. Além disso, no caso do Tacutu a mudança é muito sutil em relação ao tamanho do dado final tanto para o parâmetro A quanto para V. No dado do Jequitinhonha o dado de saída tem o mesmo tamanho para todos os sistemas de arquivo. Dessa forma, podemos concluir que o sistema de arquivo não produz resultado significativo para a técnica utilizada.

Diante desses resultados, como trabalhos futuros, vislumbra-se verificar questões de compatibilidade do algoritmo CRP para execução na família Red Hat Enterprise Linux 8 a fim de usufruir de potenciais ganhos utilizando um outro sistema de arquivos encontrado durante a execução desse projeto, o BeeGFS. Esse sistema de arquivos parece promissor no contexto de dados sísmicos, visto que foi projetado para ambientes de HPC no contexto do setor de óleo e gás (Thinkparq, 2019). Além disso, também será considerada a utilização de RAID 0 (*striping*) de forma a melhorar a velocidade de leitura e escrita, visto que essa tecnologia divide o acesso aos discos que fazem parte do RAID, aumentando sua velocidade, porém, sem garantias à falha (Intel, 2017).

Referências

Barros, T., R. F. R. K. and Lopes, R. (2015). Differential evolution-based optimization procedure for automatic estimation of the common-reflection surface travel. *Geophysics*, 80(6):189–200.

Coimbra, T. A. A. (2014). *Operação para continuação do afastamento: Operador diferencial, comportamento dinâmico e empilhamento multi-paramétrico*. PhD thesis, Podunk IN.

Gupta, M. (2002). *Storage Area Network Fundamentals*. Cisco Press, Cambridge MA.

IBM (2022). iostat command.

Intel (2017). Definindo volumes raid para tecnologia de armazenamento intel® rapid.

Kernel (2019). Ext4 howto.

Meiling, S. (2017). Research and application of xfs file system in seismic data process. *Society of Exploration Geophysicists - SEG*.

Microsoft (2008). File systems.

Microsoft (2009). Ntfs technical reference.

Neidell, N. S.; Taner, M. T. (1971). Semblance and other coherency measures for multichannel data. *Geophysics*, 36(3):482–497.

Oracle (2010). Oracle solaris zfs administration guide.

RedHat (1999). O sistema de arquivo xfs.

RedHat (2010). Storage administration guide.

Stockwell, J. J. W. (1999). Cwp/su: Seismic un*x package. Technical report.

Thinkparq (2019). The leading parallel file system.