

## 2 Uma proposta para reorganização de dados sísmicos

A maneira tradicional de organizar dados sísmicos consiste em seqüenciar as amostras variando apenas uma coordenada de cada vez exceto quando se chega no fim de um eixo. Desta maneira temos que o  $n$ -ésimo valor da seqüência, tem índice  $i=n-1$  e corresponde ao valor da posição

$$(x, y, z) = \left( \left\lfloor \frac{i}{l_z \cdot l_y} \right\rfloor, \frac{i \bmod (l_z \cdot l_y)}{l_z}, i \bmod (l_z \cdot l_y) \bmod l_z \right) \quad (1)$$

onde  $l_x$ ,  $l_y$  e  $l_z$  são as dimensões, em número de amostras das arestas do paralelepípedo, alinhados aos eixos, x, y e z respectivamente. Dizemos que esta organização prioriza um eixo. A Figura 3 ilustra este seqüenciamento.

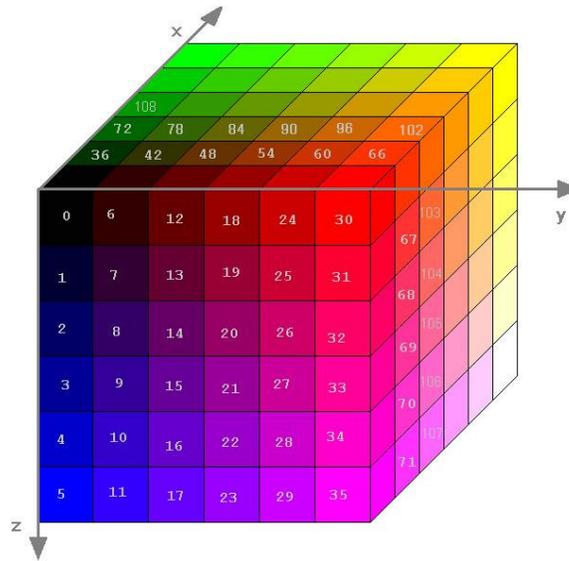


Figura 3: Índices de cada amostra.

Em geral o eixo z aponta para dentro da terra correspondendo a profundidade ou ao tempo de resposta do sinal. Os outros eixos, comumente chamados de *line* e *crossline*, montam juntos um plano paralelo a superfície.

## 2.1. Seleção e tempo de leitura

Os arquivos de dados sísmicos com os quais se deseja trabalhar são muito grandes. É inviável para uma pessoa comum ver e analisar todos os dados ao mesmo tempo. Aplicativos como v3o2, gOcad e Geoprobe®, portanto, oferecem ferramentas para a seleção de uma sub-área sendo, a mais comum delas, um semi-plano paralelo a uma das faces do volume.

A estratégia básica de seleção para a organização tradicional já explicada tenta aproveitar ao máximo a estrutura do arquivo. São lidas o máximo de amostras consecutivas pertencentes à fatia e depois salta-se para a próxima posição que contenha amostras de interesse. O processo é repetido até que toda a fatia tenha sido lida. Um exemplo numérico do procedimento pode ser acompanhado na Figura 4.

Para analisar esta estratégia realizamos uma bateria de testes. Nesta bateria cronometramos o tempo decorrido para a leitura de fatias das três direções de um volume cúbico armazenado em disco. O ambiente utilizado para os testes é indicado na Tabela 2.

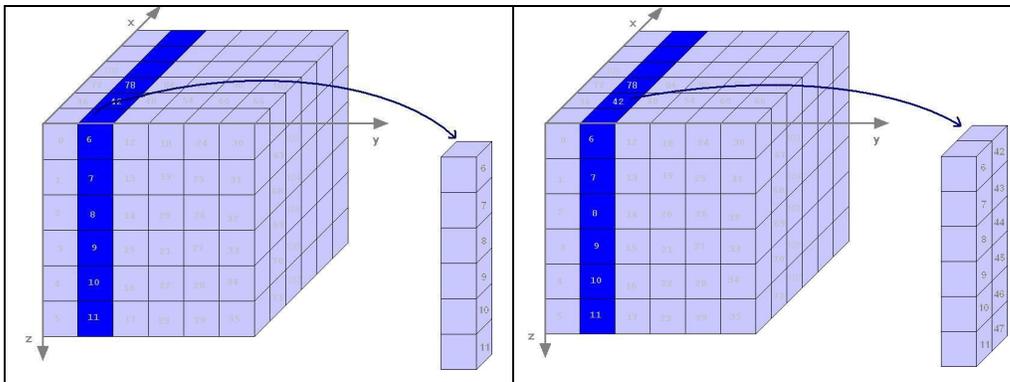


Figura 4: Exemplo das etapas do procedimento de leitura atual. O cubo tem lado 6 e deseja-se ler a fatia  $y=1$ . Primeiro lê-se as amostras de índice 6 a 11 em bloco, daí salta-se para a amostra 42 e lê-se as amostras 42 a 47 também em bloco. O processo é repetido até que se tenha lido o último bloco com as amostras de índice 181 a 191.

Dimensão	600x600x600 Amostras
Tipo da amostra	Ponto Flutuante 4 bytes
Tamanho Total do Arquivo	864.000.000 bytes (823,97 Mb)
Processador	AMD Sempron, 1.6Ghz, 32 bits
Memória RAM	512 Mb

Tabela 2: Ambiente de testes.

Fatia	Tempo (ms)	Fatia	Tempo (ms)	Fatia	Tempo (ms)
X=05	0	Y=05	3453	Z=05	51125
X=37	62	Y=37	3453	Z=37	43203
X=87	47	Y=87	3297	Z=87	48141
X=35	63	Y=35	3422	Z=35	47219
X=99	32	Y=99	3453	Z=99	44984
X=243	47	Y=243	3781	Z=243	45343
X=182	31	Y=182	3328	Z=182	42922
X=125	32	Y=125	3329	Z=125	40844
X=333	31	Y=333	3297	Z=333	40734
X=470	31	Y=470	3343	Z=470	40344

Tabela 3: Testes de leitura de fatias de dados armazenados na forma tradicional.

Os resultados da Tabela 3 mostram uma grande disparidade entre o tempo de leitura das fatias do tipo  $x=cte$  para as fatias do tipo  $z=cte$ . Analisando a estrutura dos dados percebe-se que para se ler uma fatia  $x=cte$  basta uma única leitura de  $l_y * l_z$  dados começando na posição  $d * l_y * l_z$  do arquivo. Para ler uma fatia  $y=cte$  devem ser feitas  $l_x$  leituras de  $l_z$  dados e entre cada leitura um salto de  $(l_y - 1) * l_z$  posições à frente. O ultimo e pior caso ocorre para lermos uma fatia  $z=cte$ , nesta devem ser feitas  $l_x * l_y$  leituras de tamanho 1.

Uma vez que o total de dados trazidos do disco é o mesmo nos três casos identificamos que o gargalo da operação está nos posicionamentos. Quando falamos de transferência de dados de memória para memória o tempo é proporcional ao número de bytes trazidos. Como se trata de leitura de disco para memória temos que considerar outros fatores.

Cada operação de leitura passa por três etapas [16]. Primeiro o controlador recebe a chamada e prepara a leitura, depois a cabeça do disco deve ser movida para a posição correta e por último os bytes são lidos e transferidos através de um barramento para a memória. O tempo do controlador é constante, o tempo do posicionamento é uma função,  $p(m)$ , da distância  $m$  percorrida e a transferência uma função,  $T(n)$ , dependente do número de bytes  $n$  transferidos. Podemos calcular o tempo decorrido através da seguinte fórmula

$$t = c + p(m) + T(n) \quad (2)$$

onde  $c$  é o tempo do controlador,  $m$  a distância, em bytes, percorrida pela cabeça do disco no posicionamento e  $n$  o número de bytes lidos.

Como nos sistemas operacionais multitarefa não podemos garantir a localização da cabeça do disco no momento da chamada prefere-se utilizar o tempo médio. Substitui-se, então, ( 2 ) por

$$t = c + \hat{p} + T(n) \quad (3)$$

Como  $c$  e  $\hat{p}$  são constantes temos a equação final

$$t = O + T(n) \quad (4)$$

descrevendo a leitura de disco como uma função do número de bytes lidos somada a um *overhead*.

Uma solução direta da análise dos testes acima seria ter três cópias do arquivo, variando o eixo principal da ordenação. Esta solução, porém, implica um grande desperdício de memória, já que em um ambiente como o de estudos de dados sísmicos, a coletânea é de centenas ou até milhares de arquivos cujos tamanhos individuais vão de dezenas a centenas de gigabytes. Triplicar esta capacidade é custoso, portanto, nosso desafio consiste em reduzir estes tempos mantendo o mesmo consumo de memória.

## 2.2. Estratégia de multi-fatias

A fim de identificar a magnitude do *overhead* de leitura observado nos testes da estratégia atual foi elaborada uma nova maneira para ler as fatias do tipo  $z=cte$ . Nesta estratégia lemos um bloco grande de dados do arquivo de modo a garantir a presença de mais de uma amostra da fatia  $z=cte$  desejada. Deste bloco extraímos, em operação de memória para memória, os valores de interesse e repetimos o processo para um novo conjunto de dados até que toda a fatia tenha sido lida. A Figura 5 ilustra o procedimento.

Os blocos foram determinados como um conjunto de  $k$  fatias  $x=j$  consecutivas, aproveitando a natureza da estrutura do arquivo de dados. O número de blocos a serem lidos será sempre  $\frac{l_x}{k}$ . O tempo total de leitura foi medido para vários tamanhos de sub-blocos, desde que, divisores de 600 para que não se precisasse tratar casos especiais. O tempos de leitura de um único bloco e da extração das amostras dos mesmos também foram medidos. Os resultados podem ser vistos nos gráficos da Figura 6 e da Figura 7.

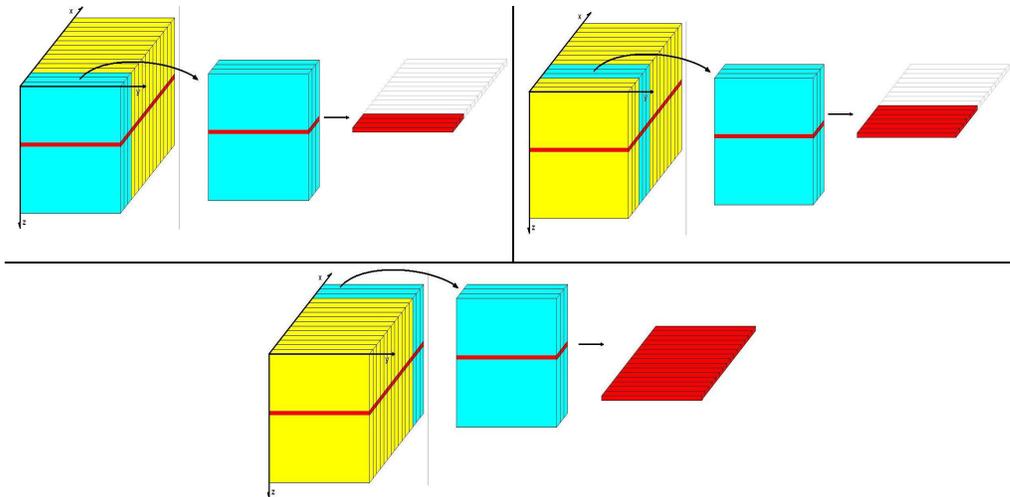


Figura 5: Estratégia de multi-fatias.

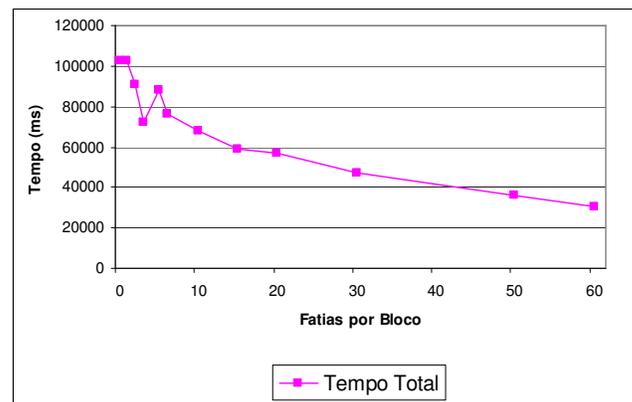


Figura 6: Gráficos mostrando o tempo total de leitura em função do tamanho do bloco.

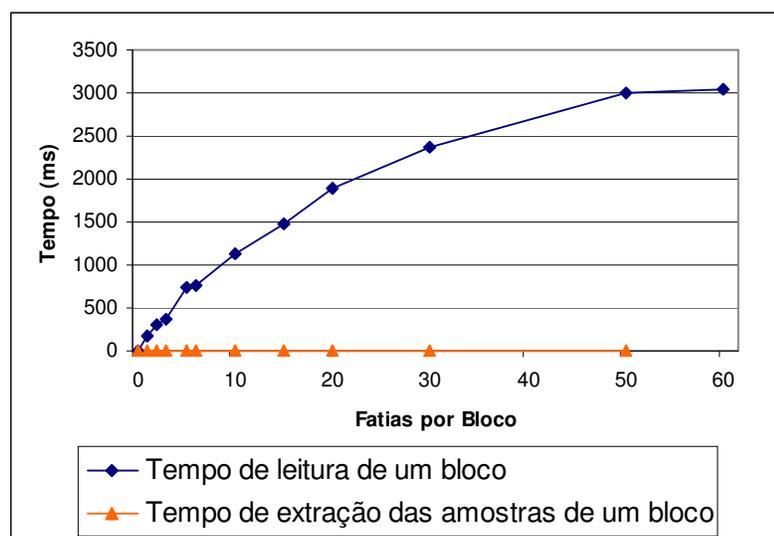


Figura 7: Gráficos mostrando o tempo de leitura de um bloco e o tempo de seleção das amostras no bloco lido em função do tamanho do bloco.

Para o volume e máquina utilizados a estratégia de leituras de fatias  $z=cte$  começou a compensar, ou seja, superar o tempo obtido pela estratégia corrente mostrado na Tabela 3, quando ultrapassamos a marca das 30 fatias de cada vez. O teste mostra que de modo geral o tempo final é decrescente com o tamanho do bloco. Isto acontece porque embora o tempo de leitura de cada bloco seja maior a redução no número de chamadas de leitura e conseqüentemente no número de *overheads* compensa.

O fato relevante destes resultados é a comprovação de que mesmo lendo mais amostras que o desejado é possível melhorar o tempo total. Na verdade, a observação do decaimento do gráfico da Figura 6, nos leva a crer que a estratégia ideal seria trazer o volume inteiro para memória e fazer todas as operações neste ambiente. Porém esta memória é limitada e, em geral, os volumes com os quais queremos trabalhar ultrapassam sua capacidade. É necessário que a estratégia mantenha o dado em disco.

### **2.3. Estratégia de sub-volumes**

Um problema identificado na estratégia de blocos feita para ler fatias do tipo  $z=cte$  é que, independentemente do tamanho do bloco, no final todos os valores do volume terão sido lidos. A razão entre as amostras descartadas para as amostras aproveitadas está muito alta. Um desperdício de leitura muito grande ocorre.

Se pudermos dividir o volume em sub-volumes tais que nenhuma fatia, em nenhuma direção, corte todos os sub-volumes podemos obter cada fatia selecionando suas amostras apenas dentro dos sub-volumes que ela intercepta.

Como exemplo podemos dividir cada direção em três partes iguais, gerando  $3^3 = 27$  sub-volumes. Reorganizamos o arquivo agrupando as amostras de um mesmo sub-volume em uma seqüência contínua de dados. A Figura 8 ilustra a organização das amostras neste novo formato. A organização anterior continua sendo usada dentro de cada sub-volume e também para ordenar os sub-volumes no arquivo todo.

Qualquer fatia, em qualquer direção, atravessa apenas nove desses sub-volumes. A estratégia de leitura passa a ser, então, identificar os sub-volumes por onde passa a fatia, carregar tais sub-volumes e deles copiar, com operação de memória para memória, as amostras de interesse. O total de amostras lidas passa a ser um terço do total do volume e o espaço necessário é de um vinte e

sete avos do volume mais o espaço necessário para uma fatia, já que após ler cada sub-volume podemos descartá-lo e copiar o próximo na mesma área de memória. A restrição de dividir todos os eixos e impor que nenhuma fatia, em nenhuma direção corte todos os sub-volumes serve para impedir a existência de casos onde o arquivo inteiro deva ser carregado para a leitura de uma única fatia.

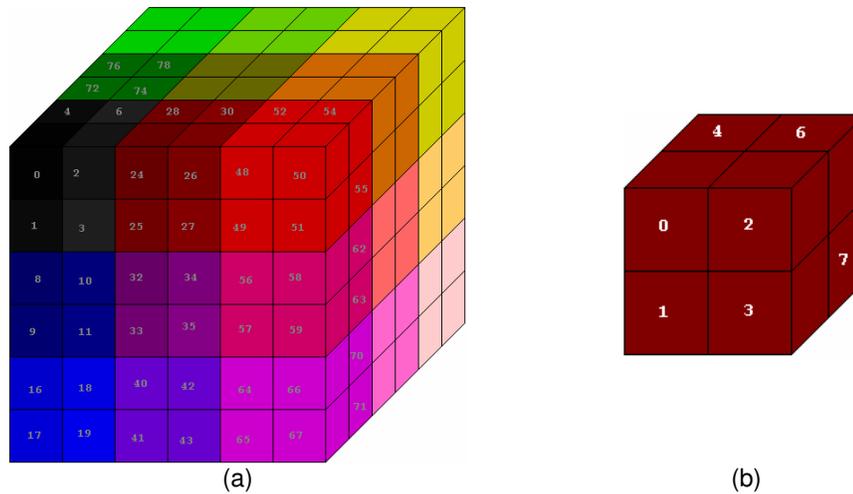


Figura 8: (a) Exemplo da indexação de acordo com a posição. ( $l_x=l_y=l_z=6$ )  
 (b) Dentro de um sub-volume a antiga ordenação é utilizada.

Uma estimativa do ganho gerado com esta organização pode ser feita a partir dos tempos de leitura e extração de amostras obtidos nos testes anteriores. No exemplo de 27 sub-volumes, uma fatia é carregada com 9 leituras de blocos, uma a menos que as necessárias para ler blocos de 60 fatias no volume de dimensão 600x600x600. Ainda utilizando este volume e subdivisão, temos um bloco de tamanho 200x200x200=8.000.000 com 200x200=40.000 amostras de interesse. Dos resultados mostrados na Figura 7 temos que a leitura de um bloco de tamanho 10.800.000 (trinta fatias de lado 600) leva aproximadamente 2.300 ms e a extração de 36.000 amostras de interesse não passa de 2ms. O tempo final ficaria sendo aproximadamente  $9 \times 2.300\text{ms} + 9 \times 2\text{ms} = 20,718\text{s}$ . Este valor esperado é menor que o melhor tempo obtido na estratégia de multi-fatias.

### 2.3.1. Algoritmo de leitura de uma fatia

A leitura completa passa por duas etapas, carregar blocos e extrair as amostras destes. Dentro de cada uma destas etapas estão, respectivamente, os processos de identificar quais blocos devem ser carregados e quais amostras são de interesse.

Vamos supor um volume de dimensão  $l_x \times l_y \times l_z$  cujos lados tenham sido divididos em  $f_x$ ,  $f_y$  e  $f_z$  partes. Com esta divisão passamos a ter  $f_x * f_y * f_z$  sub-volumes de tamanho  $d_x * d_y * d_z$  onde  $d_e = l_e / f_e$  para qualquer eixo  $e$ .

Para identificar quais sub-volumes devem ser lidos utilizamos o mesmo algoritmo de leitura existente atualmente. O que varia é que o volume total é tratado como um volume de dimensões  $f_x \times f_y \times f_z$  onde cada amostra, em vez de 4 bytes, tem o tamanho de um sub-volume. Para ler uma fatia  $e=j$  precisamos obter os sub-volumes da superfatia  $E=j/d_e=J$ .

Uma vez em memória o sub-volume é tratado como um volume independente de dimensões  $d_x \times d_y \times d_z$ , suas amostras são indexadas a partir do zero e é feita a leitura das amostras pertencentes a sub-fatia  $e=j \bmod d_e$ . Um exemplo numérico é mostrado na Figura 9.

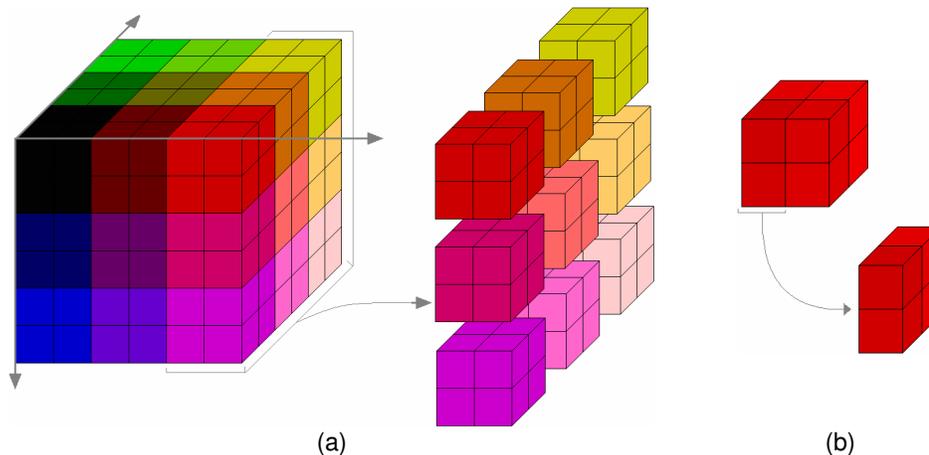


Figura 9: (a) Para ler a fatia  $y=4$  primeiro são lidos os sub-volumes  $Y=4/(6/3) = 2$   
(b) Em cada um dos sub-volumes lê-se a fatia  $y=4 \bmod (6/3) = 0$

### 2.3.2. Experimentos computacionais

A estratégia de sub-volumes precisa ser testada com várias subdivisões até que se possa entender o *trade-off* entre trazer amostras não desejadas e o tempo gasto para encontrar, em disco, as amostras de interesse.

São dois os problemas que ocorrem ao se trazer amostras a mais: um deles é o tempo acrescido devido à transferência de dados para a memória e o outro é o tempo de seleção das amostras em memória. O número total de amostras selecionadas será sempre o mesmo, já que estas são as amostras de interesse e já vimos, nos testes anteriores, que este tempo de seleção é praticamente desprezível quando comparado ao tempo de leitura do disco

Na bateria de testes realizada, as mesmas fatias, nas três direções, foram lidas para cada opção de subdivisão. Para que fosse comparável o efeito da reorganização em cada direção utilizamos um volume cúbico e os fatores de divisão eram sempre iguais nos três eixos. O cubo de teste tem lado 600 e o ambiente continua sendo o da Tabela 2. Para cada opção de divisão do lado lemos doze fatias em cada direção e a média dos tempos de leitura gera uma entrada no gráfico da Figura 10.

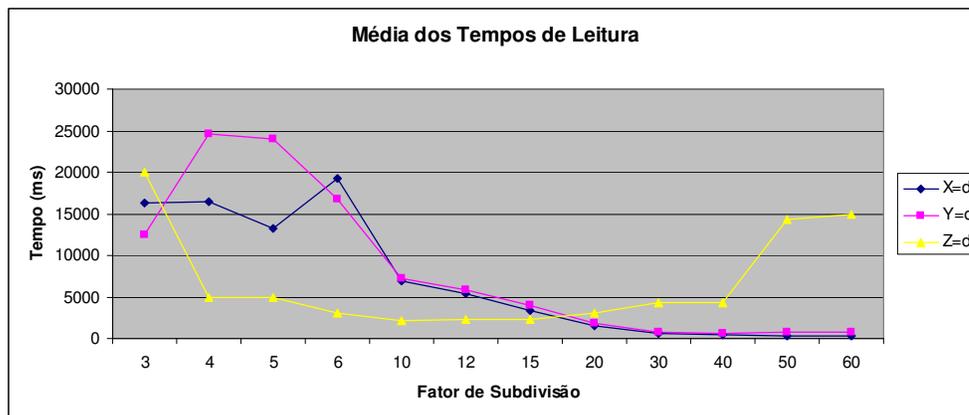


Figura 10: Gráfico mostrando o tempo médio da leitura de uma fatia em função do fator de subdivisão.

Como nas outras estruturas dos lados e estratégias de leitura existe uma clara diferença de comportamento entre as direções. Um ponto de equilíbrio no tempo das três direções é visto, para este cubo, quando o fator de subdivisão está entre 15 e 20. Isto significa ter de 3375 a 8.000 sub-volumes com número de amostras por volume entre 64.000 e 27.000. Considerando que as três direções de leitura são igualmente importantes e ocorrem com igual frequência

podemos tirar o máximo dos tempos médios de todas as direções, conforme mostrado no gráfico da Figura 11, e encontrar o ponto ótimo em 20. Neste ponto podemos garantir que para qualquer leitura feita, independente da direção, o tempo decorrido estará em torno de 3 segundos.

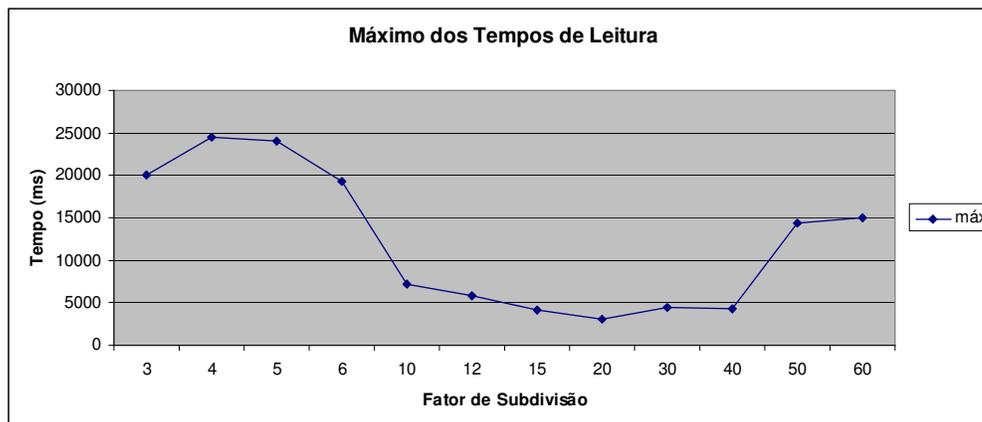


Figura 11: Gráfico mostrando o tempo máximo, entre as três direções, para a leitura de uma fatia em função do fator de subdivisão.

Embora os testes tenham mostrado uma condição em que os resultados são satisfatórios, mais estudos devem ser realizados. Já era esperado que os tempos de leitura das fatias  $z=cte$  fossem melhores que os tempos destas mesmas fatias nas estratégias anteriores, porém não está claro porque estes tempos são melhores que das direções  $x=cte$  e  $y=cte$ .

O mesmo teste foi feito também para volumes cúbicos de lado 800 e 1000. O gráfico da Figura 12 abaixo mostra como o tamanho do bloco influencia no tempo de leitura. Para blocos muito grandes a razão entre amostras carregadas e de interesse é muito alta, enquanto que para blocos pequenos muito tempo é gasto com posicionamento da cabeça do disco. Nos três volumes testado o tempo ótimo ocorre para sub-volumes de lado entre 25 e 30 amostras.

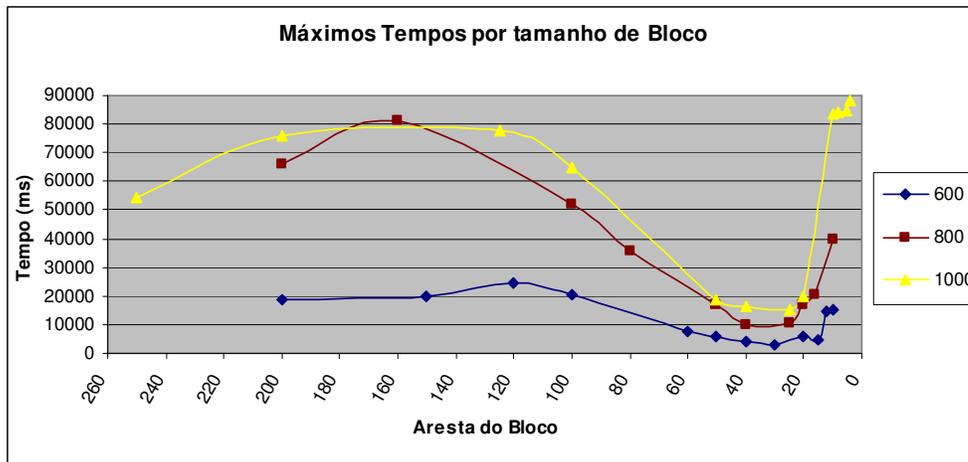


Figura 12: Gráfico mostrando o tempo máximo, entre as três direções, em função do tamanho da aresta do sub-volume para três volumes diferentes.

### 2.3.3. Otimizações

Com a nova organização o tempo médio foi reduzido consideravelmente, com uma melhora substancial na leitura de fatias  $z=cte$ , porém se examinarmos apenas fatias do tipo  $x=cte$  e  $y=cte$  temos uma piora. Como a solução mexe na estrutura do dado, e não apenas no algoritmo, não há como usar a estratégia antiga nestes dois casos. Existe ainda, contudo, uma maneira de tentar melhorar o resultado das fatias  $x=cte$  e  $y=cte$ .

Como o sub-volume guarda as amostras com a organização original podemos obter uma fatia  $x=cte$  sem ler todo o sub-volume, mas lendo apenas as amostras que pertençam a fatia de interesse. Tais amostras estarão todas contíguas em disco e, portanto, reduzimos o número de bytes carregados sem aumentar o número de leituras.

A mesma idéia pode ser aplicada para fatias do tipo  $y=cte$ , mas neste caso o número de leituras cresce e o resultado, então, depende dos tamanhos de volume e sub-volume. Testamos esta idéia para volumes cúbicos de lado 600, 800 e 1000, os resultados são mostrados nas três tabelas a seguir.

fatia subdivisão		Tempo (ms)			
		x = cte	x = cte otimizado	y = cte	y = cte otimizado
4		16.102,75	46,92	20.150,92	18,17
5		13.342,33	10,50	24.390,67	27,50
6		20177	5,17	16.946,67	27,33
10		7.455,83	6,42	7.537,58	59,97
12		5.675,92	6,42	5.696,50	92,25
15		3.240,83	10,33	3.984,50	124,83
20		1.545,50	15,58	1.772,17	134,25
30		570,25	12,92	722,58	226,50
40		4.477,83	11,67	2.187,50	301,00
50		351,67	23,58	795,67	450,33
60		296,83	18,47	823,25	501,17

Tabela 4: Comparação entre os tempos de leitura das estratégias de sub-volumes com e sem as otimizações para x=cte e y=cte. Volume de dimensões 600x600x600

fatia subdivisão		Tempo (ms)			
		x = cte	x = cte otimizado	y = cte	y = cte otimizado
4		65.820,42	23,42	49.535,00	27,42
5		81.515,50	10,42	62.368,50	41,67
8		51.933,75	19,50	50.062,42	72,83
10		33.768,25	19,58	35.865,92	121,08
16		17.113,33	5,33	15.977,67	174,50
20		10.222,5	23,42	9.912,75	265,67
32		3.546,92	24,75	2.478,00	386,67
40		2.508,92	6,58	2.270,83	459,67
50		1.937,50	22,08	1.968,67	529,92
80		1.304,75	19,58	1.764,42	855,50

Tabela 5: Comparação entre os tempos de leitura das estratégias de sub-volumes com e sem as otimizações para x=cte e y=cte. Volume de dimensões 800x800x800

fatia subdivisão	Tempo (ms)			
	x = cte	x = cte otimizado	y = cte	y = cte otimizado
5	75.617,83	8,50	65.452,42	22,75
8	68.622,33	7,87	77.831,33	43,54
10	54.518,25	9,70	64.806,04	65,17
20	17.939,42	16,29	18.557,92	192,83
25	11.051,46	20,88	16.475,38	275,91
40	2.215,46	15,63	3.535,75	416,79
50	1.724,71	20,25	2.561,25	543,58
100	866,67	15,67	2.365,29	1.183,63
125	714,04	34,54	2.634,17	1.590,33
200	447,83	39,08	3.228,42	2.645,88

Tabela 6: Comparação entre os tempos de leitura das estratégias de sub-volumes com e sem as otimizações para x=cte e y=cte. Volume de dimensões 1.000x1.000x1.000

Observando as tabelas vemos a diferença na evolução dos resultados com e sem otimização. Enquanto que o tempo para a técnica mais simples, em geral, decresce com o fator de sub-divisão para a técnica otimizada a relação é crescente. Isto está de acordo com o fato de que na otimização não ocorre leitura de amostras adicionais, assim o único fator de influência é o número de leituras. Ao comparar os tempos vemos que para estes tamanhos de volumes a estratégia compensa.

Considerando, então, todas as direções e a melhor estratégia para cada uma delas, vemos que é possível encontrar uma organização que seja melhor ou igual que a tradicional. Para o volume de lado 600x600x600, por exemplo, podemos escolher dividir em sub-volumes de lado 30 e fazer uma comparação final entre as estratégias como mostra a tabela a seguir. Os tempos de leituras nas direções y e z são claramente melhores, porém para as fatias na direção x estes tempos são muito baixos e sujeitos a imprecisão dos testes.

Fatia	Organização Tradicional	Organização de sub-volumes de lado 30 com leitura otimizada para fatias x e y
x=cte	0,037 s	0,016 s
y=cte	3,415 s	0,134 s
z=cte	44,485 s	3,071 s

Tabela 7: Comparação entre os tempos de leitura originais e com a estratégia eleita melhor para o volume e máquina utilizados.

### 2.3.4. Paralelepípedos

Para concluir este estudo estendemos a técnica apresentada para dados em paralelepípedos uma vez que, em sísmica, estes são mais comuns que dados cúbicos. Aqui vamos mostrar que a estratégia também se aplica e analisar algumas variações que podem ser consideradas.

Primeiramente fizemos a leitura de fatias em volumes organizados da maneira tradicional. Novamente notamos a disparidade entre os tempos de leitura das fatias do tipo  $z=cte$  e as demais. Mesmo quando o eixo mais longo é o eixo  $z$  e, portanto, o número de amostras de interesse é menor a leitura das fatias  $z=cte$  é mais demorada. A Tabela 8 mostra os resultados dos testes.

fatia dimensões	Tempo (ms)		
	X=cte	Y=cte	Z=cte
300 x 300 x 900	9.75	445.25	12839.88
300 x 900 x 300	0.00	252.00	14283.13
900 x 300 x 300	0.00	2162.00	14347.75
300 x 600 x 1050	76.25	1081.88	28937.63
600 x 300 x 1050	37.13	2517.38	29543.00
300 x 1050 x 600	64.50	644.50	30792.88
600 x 1050 x 300	33.25	2314.38	33779.38
1050 x 300 x 600	2.00	6906.25	29671.88
1050 x 600 x 300	21.50	6650.50	32308.50

Tabela 8: Média dos tempos de leitura de diversas fatias em volumes não cúbicos.

Para aplicar a estratégia proposta em tais conjuntos de dados consideramos duas dentre as inúmeras maneiras possíveis de dividir o dado. A primeira utiliza sub-volumes cúbicos. Neste caso o número de blocos que deve ser carregado para a memória varia de acordo com a direção da fatia de interesse. A segunda divide todos os lados pelo mesmo fator e os sub-volumes são paralelepípedos com lados proporcionais aos lados do volume total. A diferença principal entre as duas estratégias é que na primeira a razão entre o número de amostras carregadas e o número das amostras de interesse é constante para fatias na três direções. Na segunda o número de sub-volumes carregados é que é mantido nos três tipos de leitura.

Um exemplo das duas estratégias é mostrado na Figura 13 onde vemos um volume de dimensões 8 x 16 x 12 dividido em 24 sub-volumes cúbicos de

lado 4; e dividido em 64 sub-volumes de dimensões 2 x 4 x 3. Neste último caso os sub-volumes são proporcionais ao volume original:  $2/8 = 4/16 = 3/12$ .

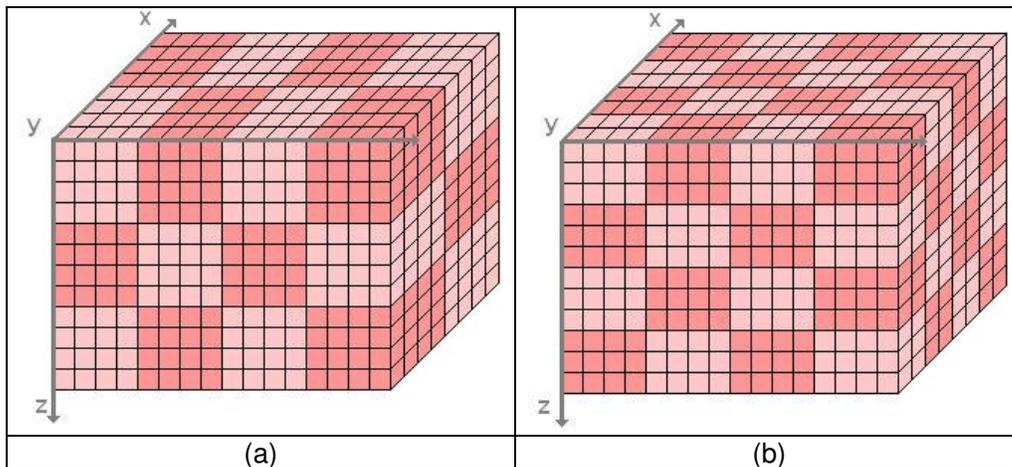


Figura 13: Um mesmo volumes sub-dividido de duas maneiras diferentes. Em (a) vemos a subdivisão em cubos. Em (b) vemos a subdivisão em paralelepípedos proporcionais ao volumes principal.

Mostramos nos gráficos a seguir os resultados dos tempos de leitura para fatias das três direções em função da dimensão do sub-volume. Separamos em gráficos diferentes as leituras feitas com sub-volumes cúbicos e não cúbicos.

#### Volume 300x300x900

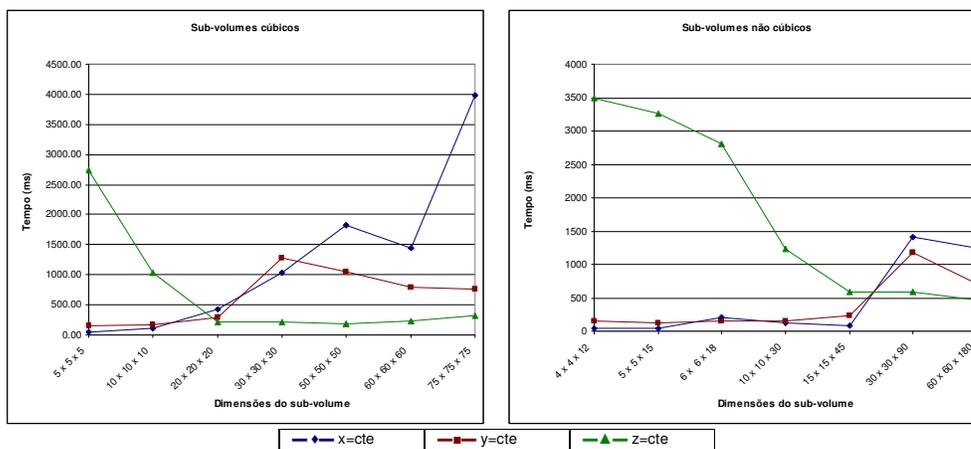


Figura 14: Gráficos mostrando os tempos de leitura de fatias dos tipos x=cte, y=cte e z=cte para as duas estratégias de divisão em um volume cuja dimensão do eixo z é superior as dimensão dos eixos x e y .

### Volume 300 x 900 x 300

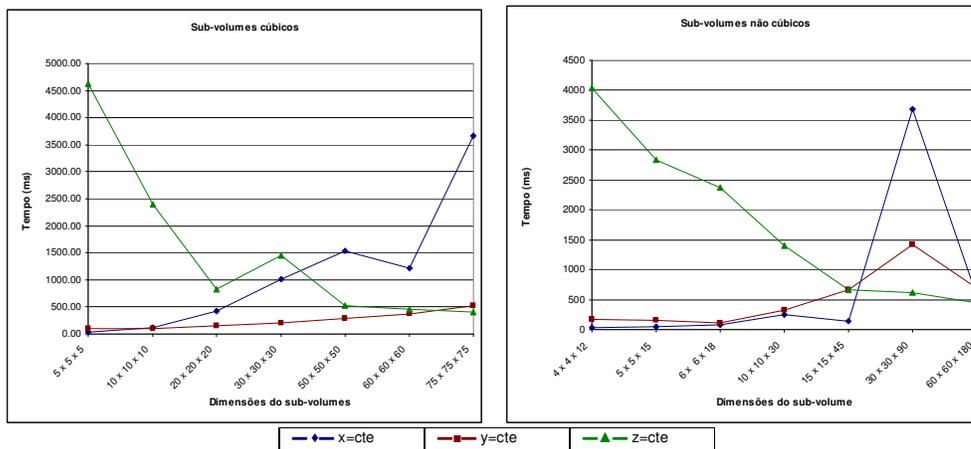


Figura 15: Gráficos mostrando os tempos de leitura de fatias dos tipos  $x=cte$ ,  $y=cte$  e  $z=cte$  para as duas estratégias de divisão em um volume cuja dimensão do eixo  $y$  é superior as dimensão dos eixos  $x$  e  $z$

### Volume 900 x 300 x 300

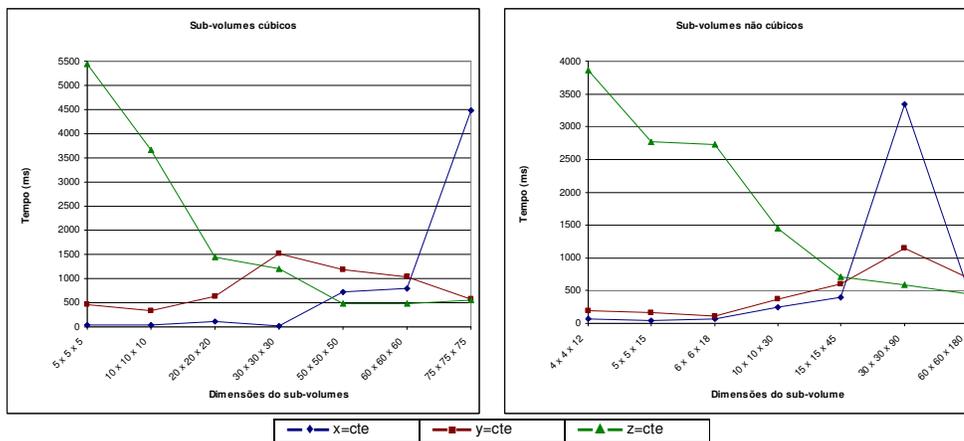


Figura 16: Gráficos mostrando os tempos de leitura de fatias dos tipos  $x=cte$ ,  $y=cte$  e  $z=cte$  para as duas estratégias de divisão em um volume cuja dimensão do eixo  $x$  é superior as dimensão dos eixos  $y$  e  $z$ .

Nos gráficos da Figura 14 até a Figura 16 vemos que para qualquer dos blocos testados o tempo máximo de leitura não ultrapassou seis segundos em todos os casos. Percebe-se a existência de um tamanho que permita manter os tempos médios inferiores a um segundo. Quando comparados com os dados da Tabela 8 onde o tempo de leitura pra fatias  $z=cte$  chegou a 14 segundos concluímos que a estratégia de sub-volumes também apresenta ganho em dados organizados em paralelepípedos não cúbicos.

### Volume 300 x 600 x 1050

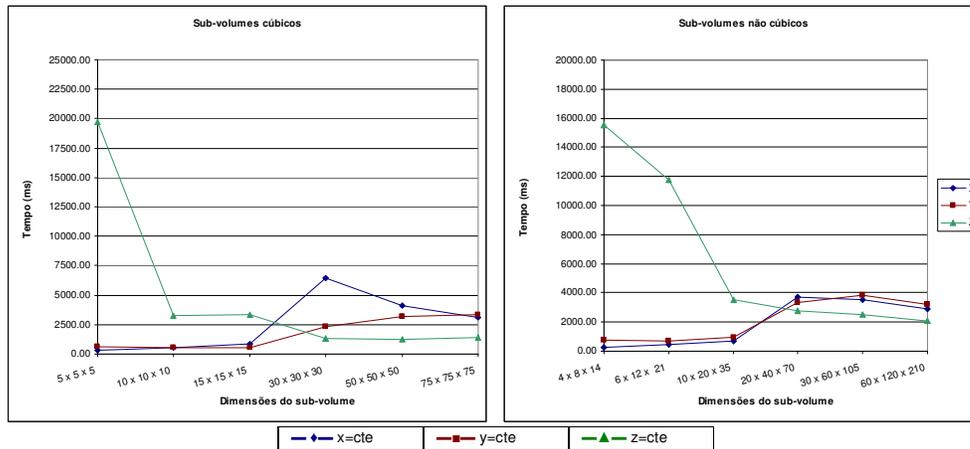


Figura 17: Gráficos mostrando os tempos de leitura de fatias dos tipos x=cte, y=cte e z=cte para as duas estratégias de divisão em um volume cuja dimensão do eixo z é superior a dimensão do eixo y; que por sua vez é superior a dimensão do eixo x.

### Volume 600 x 300 x 1050

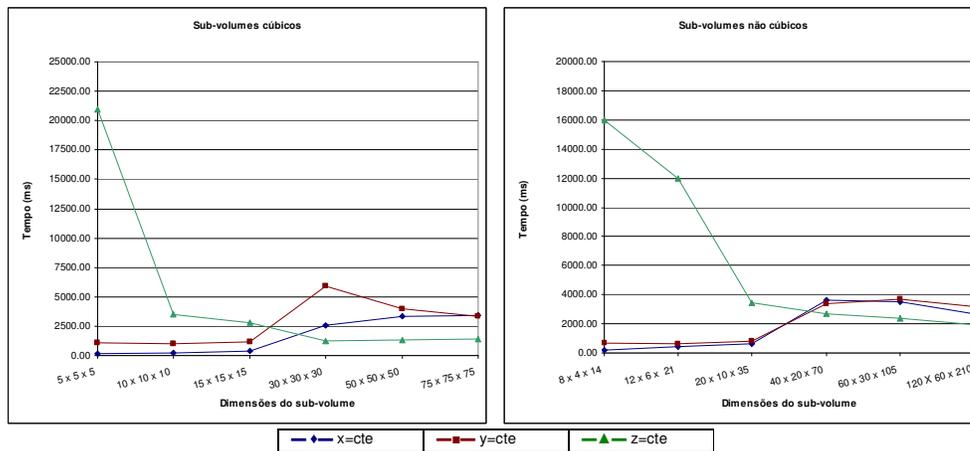


Figura 18: Gráficos mostrando os tempos de leitura de fatias dos tipos x=cte, y=cte e z=cte para as duas estratégias de divisão em um volume cuja dimensão do eixo z é superior a dimensão do eixo x; que por sua vez é superior a dimensão do eixo y.

### Volume 300 x 1050 x 600

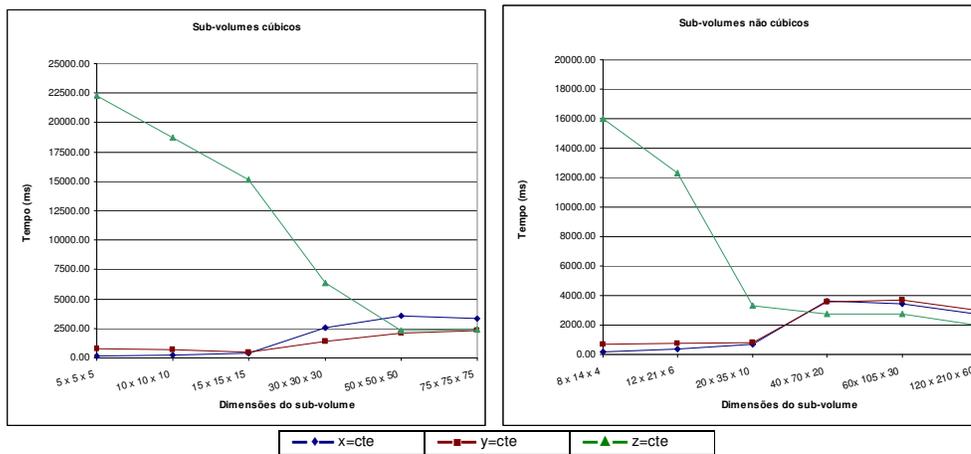


Figura 19: Gráficos mostrando os tempos de leitura de fatias dos tipos x=cte, y=cte e z=cte para as duas estratégias de divisão em um volume cuja dimensão do eixo y é superior a dimensão do eixo z; que por sua vez é superior a dimensão do eixo x.

### Volume 600 x 1050 x 300

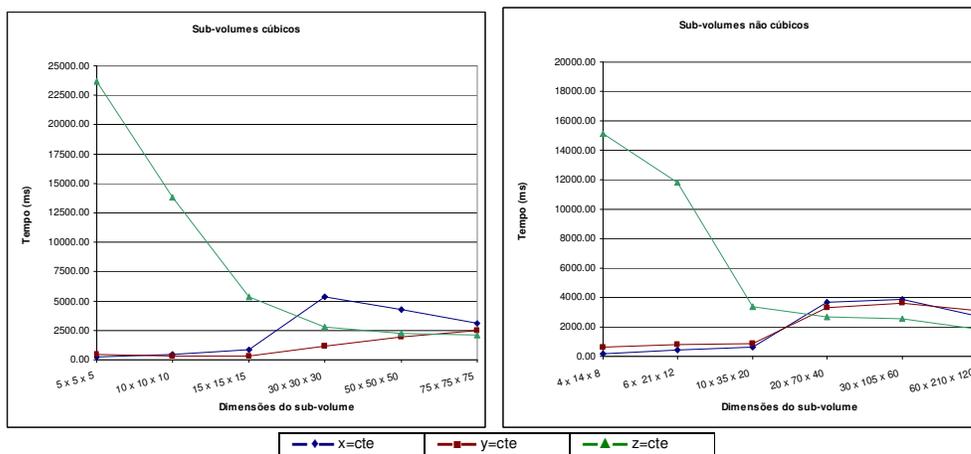


Figura 20: Gráficos mostrando os tempos de leitura de fatias dos tipos x=cte, y=cte e z=cte para as duas estratégias de divisão em um volume cuja dimensão do eixo y é superior a dimensão do eixo x; que por sua vez é superior a dimensão do eixo z.

### Volume 1050 x 300 x 600

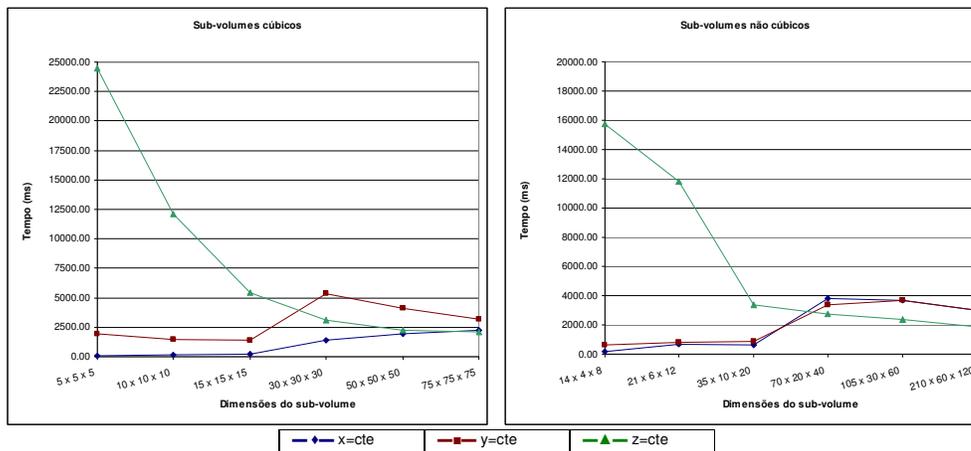


Figura 21: Gráficos mostrando os tempos de leitura de fatias dos tipos  $x=cte$ ,  $y=cte$  e  $z=cte$  para as duas estratégias de divisão em um volume cuja dimensão do eixo  $x$  é superior a dimensão do eixo  $z$ ; que por sua vez é superior a dimensão do eixo  $y$ .

### Volume 1050 x 600 x 300

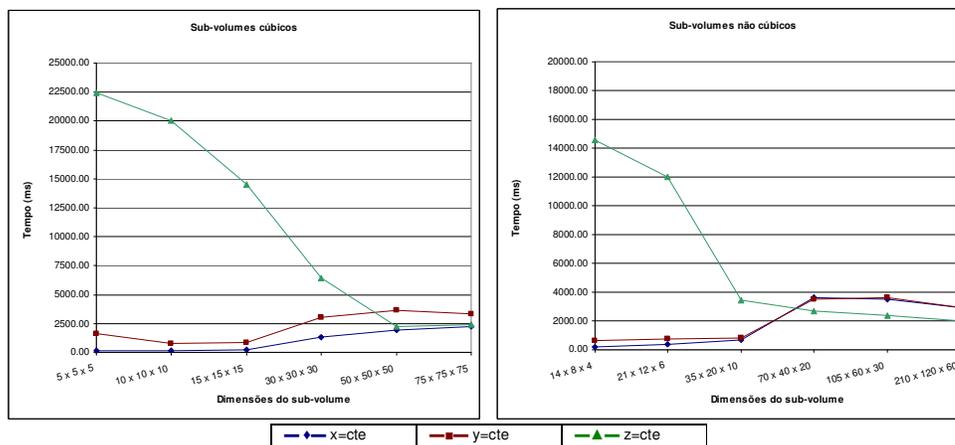


Figura 22: Gráficos mostrando os tempos de leitura de fatias dos tipos  $x=cte$ ,  $y=cte$  e  $z=cte$  para as duas estratégias de divisão em um volume cuja dimensão do eixo  $x$  é superior a dimensão do eixo  $y$ ; que por sua vez é superior a dimensão do eixo  $z$ .

Continuando nossa análise nos doze últimos gráficos novamente vemos o excelente resultado do uso desta estratégia. Em todos os casos existe um ponto cujo tempo máximo não ultrapassa 3s, tempo este que representa apenas 10% do tempo máximo na leitura da organização tradicional. Os testes realizados conseguem comprovar a validade da reorganização em sub-volumes para arquivos de dados com dimensões não cúbicas. Bons resultados foram encontrados tanto para divisões cúbicas quanto não cúbicas, porém para obter a melhor maneira de dividir os volumes é necessário um estudo mais aprofundado dos casos possíveis. Os testes nos levam a crer que o tamanho do sub-volume é

o fator mais importante a se considerar ao gerar a divisão, mas acreditamos que o formato ainda pode ser diferencial principalmente quando a disparidade entre as dimensões em cada direção for mais acentuada.

## 2.4.

### Observações finais

Os testes deste trabalho limitaram-se a leitura de fatias, porém acreditamos que a estratégia de sub-volumes seja eficiente para qualquer aplicação de sub-seleção em dados sísmicos. Esta crença se baseia no fato que tais sub-seleções, assim como as fatias, possuem uma coerência espacial, i.e. não são escolhidas amostras isoladas no espaço e sim amostras que pertencem a uma região contínua. Região esta que pode cortar o espaço de inúmeras maneiras diferentes não necessariamente tendo alguma relação com a organização do dado em disco. Um exemplo muito comum é o horizonte sísmico. O horizonte sísmico é um conjunto de pontos tais que cada par  $(x,y)$  só ocorre uma única vez e a cada um deles é associado um valor de  $z$ . Normalmente pontos adjacentes no par  $(x,y)$  tem valores próximos de  $z$  permitindo um caminhamento suave ao longo da superfície. O horizonte sísmico é uma representação aproximada da interface entre camadas do sub-solo formada ao longo dos anos. A Figura 23 mostra um exemplo de horizonte sísmico. O horizonte lembra a organização de uma fatia do tipo  $z=cte$ , justamente o pior caso de leitura na organização tradicional.

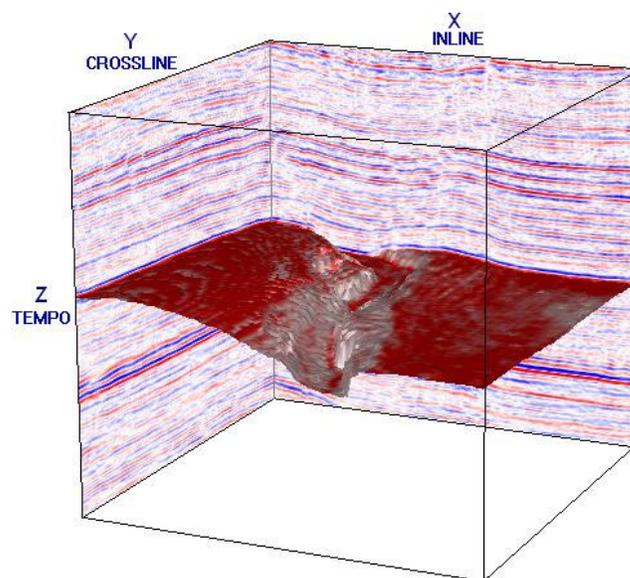


Figura 23: Exemplo de horizonte sísmico cortado por duas fatias.

Finalmente não podemos esquecer de comentar possíveis adaptações da estratégia para casos particulares que podemos prever. Em algum momento é provável que nos deparemos com volumes de lados de tamanho primo ou lados cujos divisores geram sub-volumes de tamanho muito distante do ótimo.

Para tratar este problema podemos estender as dimensões do volume até um tamanho próprio para as divisões. O dado seria preenchido com um valor especial. Esta idéia, porém, pode fazer com que o dado cresça muito e o número de amostras lidas e descartadas, em alguns casos, crescerá igualmente.

Também podemos considerar a possibilidade de dividir o dado pelo fator ótimo e formar, nos cantos, blocos menores com lados de tamanho equivalente ao resto da divisão. Os aplicativos teriam de tratar este caso especial o que não é difícil conhecendo as dimensões do dado e o fator de divisão.