4
Conclusão

Este trabalho teve como objetivo o estudo de métodos que facilitem o manejo de dados sísmicos de grande porte. Analisamos alguns problemas que a dimensão destes dados oferecem e apresentamos dois estudos independentes que nos levaram a atingir este objetivo.

Com a reorganização dos dados diminuímos o custo da leitura dos dados mantidos em disco durante o trabalho. Com a compressão permitimos baixar o custo da transferência e do armazenamento dos mesmos.

A reorganização apresentou resultados bastante satisfatórios. Os casos testados mostraram que o ganho obtido com esta estratégia compensa sua implementação. Embora não tenhamos conseguido provar a existência de um tamanho ou formato de sub-volume ótimos acreditamos que um ponto deste tipo exista para cada par máquina e volume, assim como não acreditamos que haja um único ponto ideal para todas as combinações. A determinação deste ponto, ou de um ponto próximo o suficiente, pode ser feita, para cada par máquina e volume, a partir da análise gráfica de um conjunto de leituras com subdivisões variantes semelhantes as realizadas nos testes deste trabalho.

O estudo da reorganização também deixa bem claro que é importante conhecer o formato de uso dos dados para a escolha de sua estrutura de armazenamento. Para o caso particular trabalhado, onde partimos da premissa que são buscados conjuntos de dados com coerência espacial, a proposta é adequada.

Tão importante quanto a redução dos tempos necessários para a leitura de fatias é a redução da influência da localização das amostras desejadas nestes tempos. Para aplicativos interativos este fator é importante, no caso estudado, leitura de fatias, por exemplo, a resposta pode ser a mesma, ou semelhante, independentemente da direção selecionada pelo usuário.

O método de compressão proposto e estudado aqui é fundamentado em técnicas de agrupamento. O trabalho não teve como objetivo encontrar a melhor forma de comprimir dados sísmicos e sim estudar como o agrupamento pode ser aplicado para tais fins.

O estudo foi bastante abrangente na comparação entre as duas técnicas de agrupamento apresentadas, mostrando a relação entre os critérios utilizados e as vantagens e desvantagens de cada estratégia.

Ainda completamos o estudo com o aproveitamento da natureza do dado para a codificação. Conseguimos observar três maneiras de aproveitar a coerência espacial dos dados, a diferença lateral, a diferença segundo a curva de Hilbert e o PPM, todas apresentando bons resultados. Da comparação entre estas três acreditamos que o melhores casos ocorrem quando a diferença lateral é usada em conjunto com o PPM. Finalmente apresentamos nossa proposta baseada nesta estratégia cujos resultados são satisfatórios e atingem o objetivo.

4.1. União entre reorganização e compressão

Os estudos sobre a reorganização e a compressão dos dados foram feitos inteiramente separados. Agora investigamos como poderíamos unir as duas partes e obter um novo formato que melhore, em comparação com o formato tradicional, o armazenamento, a transferência e a leitura.

Antes, vamos considerar o argumento de que esta união não é necessária pois uma vez comprimidos os dados cabem em memória. Isto, porém não é necessariamente verdade. A Tabela 16 mostrou um dado que foi reduzido a 6,75% de seu tamanho, para um dado de 20Gb esta redução é suficiente para poder trazê-lo para a memória, porém para dados de 50Gb não é. Ainda temos que levar em conta o erro da compressão, em nossos testes não temos restrições quanto a seu limite, porém é natural esperar que na prática a compressão seja sacrificada em prol de minimizar a perda de informação. Isto dito, podemos continuar com o estudo da união entre a compressão e a reorganização.

Podemos, inicialmente, pensar em duas maneiras diferentes de juntar organização e compressão. A primeira consiste em comprimir todo o dado e depois reorganizá-lo. A segunda é o oposto, primeiramente reorganiza-se e em seguida comprime-se cada parte individualmente. Em todos os casos a compressão consiste em um agrupamento, diferença lateral e uma codificação que pode ser PPM, Huffman dinâmico ou qualquer método estático.

Considerando primeiro a hipótese de realizar toda a compressão e posteriormente reorganizá-lo, vamos analisar como esta seqüência afeta a

descompressão. Suponha que conseguimos ler um bloco comprimido. Se este bloco foi codificado com um método dinâmico, a decodificação de cada símbolo depende da decodificação do símbolo que o precede e conseqüentemente a decodificação da primeira amostra do bloco depende da decodificação da ultima amostra do bloco anterior na seqüência. Sem conhecer os dados precedentes o decodificador não pode montar a tabela de freqüências e conseqüentemente fazer a decodificação correta. Neste momento torna-se necessário a leitura de outro bloco e tantos quantos forem precisos para se conseguir a primeira amostra. A leitura deixa de ser independente e a estratégia de reorganização perde o sentido. Agora vamos supor que usemos uma codificação estática. Obter o valor de cada símbolo passa a ser apenas uma operação de busca em uma tabela comum para todo o volume de dados. A decodificação é feita mas os valores originais ainda não são recuperados pois os símbolos são na verdade o resultado da diferença lateral. Para desfazer a operação da diferença caímos no mesmo problema da decodificação de um método dinâmico. A obtenção do valor original de uma amostra depende do valor original da que a precede. Concluímos, então que comprimir todo o dado e depois reordená-lo não é uma boa estratégia.

Vamos considerar, agora, o inverso. Podemos dividir todo o volume nos blocos e depois comprimir cada um deles individualmente. Embora não seja esperado que a compressão das partes menores seja tão eficiente quanto a compressão do dado todo não vamos levar este fato em consideração por enquanto, vamos apenas afirmar que se cada uma das compressões atingirem a taxa desejada, a compressão total também estará nessa faixa.

Na etapa de quantização, além do dado deve ser armazenada a tabela de valores. Esta tabela pode ter tantas entradas quantos forem o número de grupos, como no caso do agrupamento por K-Medianas ou apenas os dois limiares do intervalo no caso do agrupamento por intervalos uniformes. Uma tabela de códigos também deve ser armazenada para cada bloco se for escolhida uma codificação estática. Como a codificação é feita após a etapa da diferença lateral esta tabela tem *k*2-1* entradas.

Como exemplo vamos ver o que ocorre se optarmos por agrupamento por 256 Medianas com codificação dinâmica. Neste caso uma tabela deve ser guardada para cada bloco. Para a reorganização utilizamos blocos de lado 25 conforme se mostrou adequado nos testes do Capitulo 2, então temos 15.625 amostras de 4 bytes por bloco. Supondo um dado comprimido com 10% do tamanho original este bloco ocupa 15.625 x 4 x 0,1 = 6.250 bytes. Se a

quantização tem 256 índices a tabela deve ocupar 256 x 4 = 1.024 bytes. Isso indica que para cada 6.250 bytes acrescentamos 1024 para a tabela. Este acréscimo corresponde a 16% do bloco comprimido mas representa, para a compressão final, um acréscimo de apenas 1.6%, valor baixo porém ainda significativo. O acréscimo final é o mesmo para qualquer fator de compressão atingido, mas varia de acordo com o tamanho do bloco e da tabela.

Para a quantização por intervalos uniformes com codificação dinâmica bastam apenas dois valores i.e. oito bytes para cada bloco, um acréscimo insignificante. Isto poderia ser justificativa para escolher o agrupamento por intervalos uniformes sobre as K-Medianas, porém não resolve outro problema proveniente desta estratégia de união entre reorganização e compressão. A existência de uma tabela para cada bloco pode gerar associações diferentes para amostras de valores idênticos se estas estiverem em blocos distintos. É um problema de descontinuidade não desejado que pode prejudicar tanto análises visuais quanto cálculos matemáticos. Um exemplo análogo é o formato JPEG [7,17] onde a compressão é aplicada de maneira independente em sub-quadrados 8x8 da imagem.

Nenhuma das duas estratégias se mostrou ideal porque tentamos manter todas as etapas da compressão seguidas. O que podemos fazer é separá-las pela reorganização. A compressão é composta da quantização, da diferença lateral e da codificação. A quantização é feita no arquivo todo deixando uma única tabela para o dado inteiro. Daí é feita a reorganização. A diferença lateral e a codificação são então aplicadas em cada bloco individualmente. A codificação escolhida pode ser tanto dinâmica por bloco ou estática para o volume todo, dependerá do caso comparar se a tabela compensa ou não. Como os blocos são pequenos pode ser que métodos como o PPM não tenham uma boa adaptação. O importante é que a decodificação possa ser feita individualmente pra cada bloco.

Com a compressão aplicada os sub-volumes não mais têm, garantido, o mesmo tamanho em byte, embora tenham o mesmo número de amostras. Isto implica modificações no algoritmo de leitura. Para saber quais bytes carregar para memória precisamos saber como identificar o início e o fim de cada bloco. Um mapa de índices em memória é uma solução simples, com ele a identificação dos blocos continua sendo feita em tempo constante. O mapa, contudo apresenta um acréscimo no tamanho do dado já que este deve ser armazenado junto com o mesmo. Este acréscimo será proporcional ao número de sub-volumes e independe do fator de compressão. Seguindo o exemplo

anterior e utilizando blocos de lado 25, temos um bloco, i.e. uma entrada no mapa para cada 15.625 amostras. Novamente supondo um dado comprimido com 10% do tamanho original este bloco ocupa 6.250 bytes e com cada entrada no mapa ocupando 64 bits para endereçamentos acima de 4Gb, então o acréscimo na compressão é de 8 bytes pra cada 6.250. O acréscimo, nesse exemplo é menor que 0,01%. Mesmo com variações no tamanho do bloco o fator de acréscimo tende a continuar pouco significativo.

Com base na discussão acima propomos um novo formato de armazenamento dos dados: uma das técnicas de agrupamento é realizado no conjunto todo que em seguida é dividido em sub-volumes, a diferença lateral e o PPM são aplicados a cada bloco. Desta maneira não há necessidade de tabelas de informação adicional por bloco, a tabela do agrupamento e o mapa de índice são únicos para o dado todo. Neste formato deve-se apenas levar em consideração o caráter compensatório que reorganização e compressão apresentam, se os sub-volumes forem pequenos demais o PPM pode ser comprometido porém poucos blocos grandes podem não ser suficientes para tornar a leitura de dados eficiente.

4.2. Trabalhos futuros

Para a aplicação da reorganização ainda é preciso estudar as variações que podem ser necessárias. Muitas vezes os volumes de dados sísmicos não formam paralelepípedos e sim poliedros irregulares. Podemos estender nossos estudos para estas variações. É necessário organizar os sub-volumes dentro destes poliedros de forma que mantenham a performance da leitura. A determinação do tamanho do sub-volume ideal também ficou em aberto neste trabalho. Trabalhos mais avançados devem ser feitos para entender exatamente como *hardware*, sistema operacional tamanho e formato do dado influenciam nesta escolha.

Também não foi abordado aqui a possibilidade de utilizar os sub-volumes para um sistema de *cache*. Os aplicativos que oferecem as ferramentas de leitura podem implementar uma estratégia que mantenha uma quantidade dos sub-volumes em memória prevendo sua utilização.

O trabalho sobre agrupamento pode ser estendido em vários aspectos. Pode-se trabalhar com novas funções de custo em conjunto com outros critérios de avaliação do erro e da qualidade; e ainda podemos estudar como o erro se

propaga com a aplicação dos muitos procedimentos matemáticos pelos quais os dados costumam passar na análise sísmica.