

INVESTIGAÇÃO DE PARALELISMO NA MIGRAÇÃO SÍSMICA

*Silvio Sinedino Pinheiro*¹

*Jairo Panetta*²

*Cláudio Luís de Amorim*³

RESUMO

Este trabalho visa avaliar a viabilidade do uso de paralelismo no processamento sísmico. Para isto foi escolhida uma aplicação representativa (migração sísmica), resolvida por um algoritmo comumente empregado em produção (migração ω -x). Sua implementação foi experimentada em cinco máquinas paralelas com três arquiteturas distintas. Os resultados experimentais foram constantemente bons e coerentes com trabalhos correlatos recentes, o que nos leva a concluir pela viabilidade do uso do paralelismo no processamento sísmico.

ABSTRACT

This work intends to evaluate the feasibility of applying paralelism into seismic processing, by selecting a representative application (seismic migration), a well established algorithm (ω -x migration) and testing implementations on five parallel machines under three different architectures. Experimental results were consistently good, in accordance to recent related work, supporting the conclusion that is feasible to apply paralelism in seismic processing.

1- INTRODUÇÃO

A fase inicial da prospecção do petróleo busca definir locais promissores para a perfuração dos poços. Isto exige um conhecimento profundo da disposição das camadas geológicas em subsuperfície, na área de interesse.

Para isto são utilizados os métodos indiretos de prospecção geofísica como a gravimetria, a magnetometria e a sísmica de reflexão. Dentre eles destaca-se o Método Sísmico de Reflexão. Este método compreende três etapas: Aquisição, Processamento e Interpretação Sísmica.

Na Aquisição Sísmica [Duarte85] são geradas, na superfície, ondas acústicas que se propagam nas camadas em subsuperfície. Ao encontrar uma mudança de meio

¹MSc (COPPE – 1993), *Analista de Sistemas do Depto. de Exploração da PETROBRÁS*, email: s090@c53000.petrobras.anrj.br

²PhD (Purdue – 1985), *Pesquisador Adjunto do IEAv/CTA*, email: panetta@ieav.cta.br

³PhD (Imperial College – 1984), *Professor Adjunto da COPPE – Sistemas (UFRJ)*, email: amorim@rio.cos.ufrj.br

(doravante denominada um *evento*) parte da energia é refletida e o restante é refratada. A energia refletida é captada na superfície por dispositivos especiais e é gravada digitalmente em meio magnético. O processo de construção de imagens representativas da geologia em subsuperfície a partir desses dados requer a aplicação de diversos processos computacionais, constituindo o que é conhecido como Processamento Sísmico.

O Processamento Sísmico [Hatton86], apesar de largamente utilizado, é limitado pela capacidade computacional disponível. O volume de dados a processar num levantamento típico em duas dimensões (2-D) é grande (da ordem de centenas de megabytes) e alguns processos são computacionalmente muito intensos ($O(n^3)$). Assim, muitas vezes, estes processos tornam-se praticamente inviáveis, sendo postergados para fases avançadas do processamento, quando o volume de dados já se encontra bastante reduzido. Nestes casos, não é todo o potencial do processo que pode ser aproveitado pela sua aplicação tardia.

A Interpretação dos Dados Sísmicos é a análise dos dados coletados e processados, com base no conhecimento geológico da região, procurando identificar nas camadas geológicas em subsuperfície a localização de situações mais favoráveis à ocorrência de jazidas de óleo e gás.

São exatamente os intérpretes, com suas cabidas exigências de uma maior quantidade de informações nos dados processados, que requerem a utilização de técnicas mais complexas e computacionalmente intensivas que demandam o uso de computadores progressivamente mais poderosos. Assim, nem os atuais supercomputadores onde é executado o processamento sísmico tornam factível a aplicação de tais processos.

O surgimento das primeiras máquinas paralelas comerciais traz a indagação do seu desempenho no Processamento Sísmico. "Será viável a utilização das máquinas paralelas no Processamento Sísmico ?"

Ajudar a responder esta pergunta é a motivação deste trabalho.

Para isto, escolheu-se um processo representativo da intensidade computacional do Processamento Sísmico que é a Migração Sísmica [Loewenthal76]. Em seguida, escolheu-se um algoritmo amplamente utilizado, a migração ω -x e desenvolveu-se uma formulação paralela deste algoritmo que foi implementada em cinco máquinas paralelas: INTEL/iPSC, NCP-I/COPPE, IBM 3090/600VF, IBM 9021/820 e uma rede de estações IBM RS-6000 funcionando como uma máquina paralela sob PVM (Parallel Virtual Machine).

Após grande quantidade de experimentos resultando em ganhos consistentemente favoráveis em todo o espectro de máquinas testadas, concluiu-se pela viabilidade do uso de paralelismo no Processamento Sísmico.

Este trabalho está organizado do seguinte modo: a seção 2 introduz a migração sísmica e apresenta o método de migração ω -x. A seção 3 apresenta o algoritmo e discute o paralelismo da migração ω -x. As seções 4, 5 e 6 descrevem os experimentos nas

máquinas hipercúbicas, de memória central e na rede de estações, respectivamente. A seção 7 apresenta as conclusões.

2- A MIGRAÇÃO SÍSMICA

A Aquisição Sísmica obtém fotografias da subsuperfície, apresentadas na forma de uma matriz denominada *seção sísmica*, com as colunas representando profundidade e as linhas deslocamento na superfície. Cada elemento da matriz é denominado uma *amostra* e cada coluna é denominada um *traço*.

As seções sísmicas são obtidas gerando-se, na superfície, ondas acústicas que, ao se propagarem pela subsuperfície, são parcialmente refletidas nas interfaces das camadas geológicas (os eventos). As reflexões são registradas, na superfície, por microfones específicos. Conseqüentemente, a Aquisição Sísmica gera uma imagem da subsuperfície obtida na superfície. As estruturas geológicas mostradas nessa seção não estão na posição correta em subsuperfície. Isto ocorre por diversos motivos, alguns inerentes à técnica de aquisição.

A Migração Sísmica [Rosa93] é o passo principal do processo que transforma a imagem recebida na superfície na imagem real do que existe em subsuperfície. A figura 1 mostra uma seção sísmica antes e após a aplicação da migração sísmica. Observe-se que a migração moveu o evento B (em (a)) para a posição A (em (b)) e colapsou a difração D (em (a)) para o seu ápice P (em (b)).

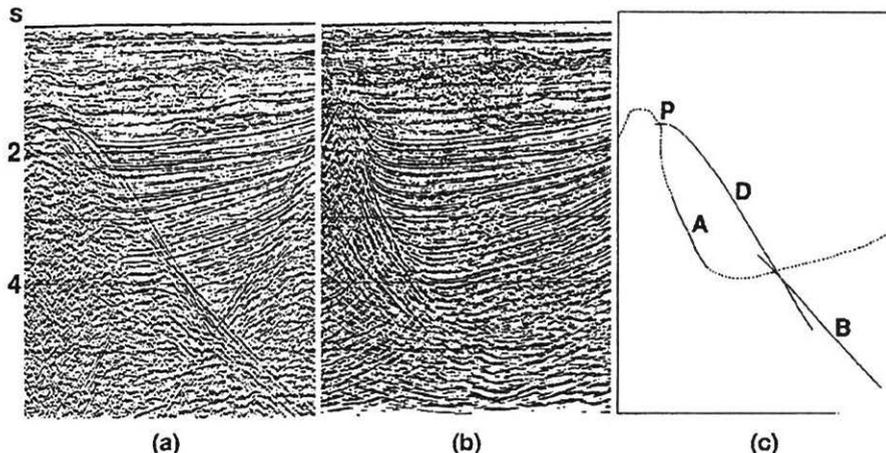


Figura 1. Efeito da migração sísmica: Seção sísmica antes (a) e após (b) a migração sísmica. Em (c) vê-se o desenho esquemático de uma difração proeminente D migrada para a posição P e um evento mergulhante antes (B) e após (A) a migração. (Adaptação de [Yilmaz87]).

Conseqüentemente, a migração sísmica corrige a forma dos eventos e reposiciona-os. A forma de um evento é um dos indicadores da possibilidade da estrutura armazenar

óleo. O reposicionamento, conforme mostram dados reais, pode alterar o local das estruturas (e conseqüentemente, o ponto de perfuração) em até 4 km.

3- O ALGORITMO SEQUENCIAL E SUA VERSÃO PARALELA

A propagação de ondas em um meio acústico, homogêneo e bidimensional é modelada pela equação da onda:

$$\frac{1}{V^2} \frac{\partial^2 P}{\partial t^2} = \frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial z^2}$$

onde $P(x,z,t)$ representa o campo de ondas e V é a velocidade de propagação no meio. No sistema de coordenadas cartesianas adotado, x representa o deslocamento ao longo da superfície, z representa a profundidade em subsuperfície e t representa o tempo. O problema da migração é, dada a seção registrada na superfície ($P(x,0,t)$) encontrar a configuração da subsuperfície ($P(x,z,0)$) capaz de gerar esse registro. Logo, trata-se de um problema inverso.

O algoritmo da migração ω - x resolve essa questão em dois passos. O primeiro passo é extrapolar o campo de ondas registrado na superfície para profundidades sucessivamente crescentes, até a profundidade máxima desejada. O segundo passo é obter, no campo extrapolado, as amostras que compõem a subsuperfície em cada profundidade. Cada passo do processo de extrapolação corrige as distorções introduzidas pela propagação de ondas durante a coleta de dados, para a seção inteira mas apenas na profundidade atual. O segundo passo é imediato, pois as amostras que compõem a subsuperfície em uma profundidade z são $P(x,z,0)$.

A extrapolação é efetuada no domínio da frequência, aproveitando as propriedades matemáticas desse domínio. Para tanto, a seção é convertida do tempo para a frequência apenas uma vez, no instante inicial da computação, e nunca mais retorna ao domínio do tempo. Apenas as amostras que compõem a subsuperfície, em cada profundidade, devem voltar ao tempo. Mesmo assim, esta operação é extremamente simples (apenas uma acumulação) pois aproveita o fato que retorna-se para $t = 0$.

A implementação segue estritamente a sugerida em [Claerbout85], conforme a discussão em [Pinheiro93]. O domínio é discretizado com n_x traços, n_t amostras por traço, n_ω frequências e n_z profundidades. Como n_t e n_ω são parâmetros derivados de n_x ($n_z = n_t$ e $n_\omega = n_t/2$), bastam n_t e n_x para determinar o tamanho do problema. A estrutura do algoritmo é fornecida a seguir:

```

Obtém a seção em frequência (múltiplas FFTs)
Para  $i_z = 2$  até  $n_z$  faça      (Laço em profundidade)
  Para  $i_\omega = 2$  até  $n_\omega$  faça    (Laço em frequência)
    Extrapola as amostras da frequência  $i_\omega$  para a profundidade  $i_z$ ;
    Acumula as amostras de interesse nesta profundidade
  fim faça      (Fim do laço em frequência)
fim faça      (Fim do laço em profundidade)
  
```

O laço central é executado exatamente $(n_z - 1)(n_\omega - 1)$ vezes, o que corresponde à $(n_t - 1)(n_f/2 - 1)$. A complexidade de cada um dos passos do aninhamento é de $O(n_x)$. Conseqüentemente, a complexidade do aninhamento é $O(n_t^2 n_x)$, ou seja, o tempo de execução do aninhamento cresce linearmente com o número de traços e quadraticamente com o número de amostras por traço. Como testes isolados demonstraram que o tempo gasto nas FFTs é desprezível face aos outros componentes da computação, concluímos que o tempo de execução é dominado pelo aninhamento central.

Uma simples análise de dependências detectou múltiplas estratégias de paralelização do algoritmo. Destas, selecionamos a que particiona o espaço de frequências pelos processadores, ou seja, cada processador recebe as amostras relativas a um conjunto de frequências e as extrapola da superfície até o fundo da seção. A acumulação é realizada em dois passos. Localmente, cada processador acumula as amostras das frequências sob seu controle. Globalmente, um processador utiliza as acumulações parciais de todos os outros processadores para realizar a acumulação final.

O tempo de execução seqüencial, derivado da análise da complexidade, é aproximadamente $k_0 n_t^2 n_x$. Já o tempo de execução paralela é composto por três parcelas: o custo da transmissão inicial da seção em frequência, o custo da computação local a cada processador e o custo da acumulação final.

Admitindo-se que as FFTs são computadas por um único processador, o custo da transmissão inicial é de $k_1 n_t n_x$, pois toda a seção em frequência (de tamanho $n_t n_x$) deve ser transmitida de um único processador para os demais.

A computação local é idêntica à computação seqüencial, apenas realizada sobre um número menor de frequências. Indicando por p o número de processadores e admitindo que p divide exatamente n_ω , o número de frequências por processador será n_ω/p . Como estas frequências serão tratadas simultaneamente pelos processadores, o custo da computação local será $(k_0 n_t^2 n_x)/p$.

O custo da acumulação final abrange tanto a transmissão das p acumulações parciais quanto a acumulação final propriamente dita. Cada transmissão movimenta $n_t n_x$ dados, que também é o número de somas necessárias à incorporação destes dados na acumulação final. Como há p transmissões (na realidade, $p - 1$, que serão aproximadas por p) e armazenando em k_2 o custo da transmissão e da soma de cada dado, obtém-se o custo de acumulação final de $k_2 p n_t n_x$. Como $n_z = n_t$, atinge-se $k_2 p n_t n_x$.

Conseqüentemente, o custo da execução paralela será $k_1 n_t n_x + (k_0 n_t^2 n_x)/p + k_2 p n_t n_x$, e o ganho será

$$S(n_t, n_x, p) = \frac{k_0 n_t^2 n_x}{k_1 n_t n_x + \frac{k_0 n_t^2 n_x}{p} + k_2 p n_t n_x}$$

Simplificando-se esta fórmula e utilizando k_3 para representar o quociente de k_1 por k_0 e k_4 para representar o quociente de k_2 por k_0 atinge-se

$$S(n_i, n_x, p) = \frac{pn_i}{k_3p + n_i + k_4p^2}$$

Esta formulação permite três conclusões interessantes. Primeiro, o ganho independe de n_x , ou seja, do número de traços por seção. Segundo, para um valor fixo de p , o ganho tende a p quando n_i tende a infinito. Isto significa que, para qualquer número de processadores, haverá um problema de tamanho suficiente para obter ganho ótimo. Terceiro, para um valor fixo de n_i , o ganho tende a 0 quando p tende a infinito. Conseqüentemente, para um problema de tamanho fixo, não adianta acrescentar processadores à máquina acima de um certo valor, pois o ganho cairá.

Esse algoritmo foi implementado e experimentado em diversas máquinas paralelas, visando obter dados reais que corroborem a discussão teórica recém apresentada. Os resultados desse esforço são apresentados nas próximas seções.

4- EXPERIMENTOS EM MÁQUINAS HIPERCÚBICAS

Os representantes das arquiteturas hipercúbicas são as máquinas disponíveis na COPPE (INTEL/iPSC e NCP-I/COPPE). São máquinas de memória real com 8 e 2 megabytes de memória por processador, respectivamente. O iPSC possui oito processadores medianamente poderosos (da ordem de dezenas de megaflops) e o NCP possui oito processadores pouco poderosos (da ordem de megaflops). O tempo necessário para executar seqüencialmente uma migração sísmica nessas máquinas em função do tamanho do problema está retratado nas tabelas a seguir.

	64 tracos	128 tracos	256 tracos	512 tracos	1024 tracos
Numero de Amostras					
64	3,73	7,37	14,65	29,50	58,80
128	14,63	29,38	58,88	119,29	238,03
256	58,43	117,09	236,42	479,96	957,13
512	234,46	466,88	944,90	1.924,29	3.840,75
1024	935,08	1.864,50	3.760,41	7.702,57	

Tabela 1 - INTEL/iPSC - Tempos de execução (s) seqüencial.

	64 tracos	128 tracos	256 tracos	512 tracos	1024 tracos
Numero de Amostras					
64	35,85	71,56	141,78	281,71	561,57
128	145,17	289,70	574,57	1.140,17	.
256	583,70	1.165,80	2.318,44	.	.
512	2.342,07	4.680,28	.	.	.
1024	9.385,05

Tabela 2 - NCP I/COPPE - Tempos de execução (s) seqüencial.

Os resultados comprovam a análise de complexidade seqüencial (dependência linear com o número de traços e quadrática com o número de amostras). O poderio desta classe de máquinas não reside na potência de cada processador, mas sim na capacidade ilimitada de agregar processadores, conseqüentemente, esperam-se bons resultados na implementação paralela do algoritmo. As lacunas nas tabelas são fruto da baixa quantidade de memória disponível e da inexistência de memória virtual.

A implementação paralela manteve a estratégia anteriormente apresentada, distribuindo as freqüências pelos processadores. Embora utilizando o mesmo algoritmo, cada máquina requereu implementação dedicada, devido a variações na sintaxe das primitivas de comunicação. As tabelas a seguir contém os ganhos em função do tamanho do problema.

	64 tracos	128 tracos	256 tracos	512 tracos	1024 tracos
Numero de Amostras					
64	5,40	6,09	6,54	6,25	6,23
128	6,84	6,68	7,02	7,27	7,22
256	7,35	7,52	7,64	7,61	7,61
512	7,77	7,78	7,81	7,78	.
1024	7,96	7,94	7,91	.	.

Tabela 3 - INTEL/iPSC - Ganhos com 8 processadores.

	64 tracos	128 tracos	256 tracos	512 tracos	1024 tracos
Numero de Amostras					
64	6,07	6,22	6,29	6,30	6,30
128	6,93	6,98	7,00	7,00	.
256	7,40	7,42	7,42	.	.
512	7,66	7,66	.	.	.
1024	7,79

Tabela 4 - NCP I/COPPE - Ganhos com 8 processadores.

Os resultados são excelentes, com ganhos próximos ao ótimo nos problemas de maior porte nas duas máquinas. Observa-se também o aumento do ganho com o número de amostras e uma ligeira variação do ganho com o número de traços, notadamente quando o tamanho do problema, *por processador*, é pequeno. Estas constatações atestam a correção das previsões decorrentes da análise de complexidade.

5- EXPERIMENTOS EM MÁQUINAS DE MEMÓRIA CENTRAL

Os representantes das arquiteturas paralelas de memória central são o IBM 3090/600VF (6 processadores) e o IBM 9021/820 (4 processadores), ambos disponíveis na PETROBRÁS. Essas máquinas apresentam processadores vetoriais poderosos, com desempenho de centenas de megaflops por processador. Tratam-se de máquinas preparadas para resolver problemas que requeiram a manipulação de grandes volumes de dados. Conseqüentemente, espera-se obter um desempenho seqüencial superior. A tabela a seguir apresenta o tempo de execução seqüencial em função do tamanho do problema (com número de traços idêntico ao número de amostras).

Dimensao	3090	9021
64 x 64	1,24	0,53
128 x 128	8,90	3,32
256 x 256	67,68	25,60
512 x 512	520,71	188,71

Tabela 5 - Tempos de execução seqüencial (s).

O número de experimentos foi reduzido em função da grande utilização das máquinas testadas (são as máquinas de produção científica da PETROBRÁS) e da necessidade de acesso exclusivo para realizar os experimentos. Mesmo assim, os resultados confirmam as previsões iniciais. Cada processador do 3090 é aproximadamente 4 vezes mais veloz que o do iPSC e 34 vezes mais veloz que o do NCP. Por sua vez, cada processador do 9021 é 10 vezes mais veloz que o do iPSC e 92 vezes mais veloz que o do NCP. Embora estes números aparentem ser exagerados, eles não são justos para

com o 3090 e o 9021, pois a implementação utilizada apresenta grau de vetorização muito inferior à média das aplicações sísmicas.

A exploração do paralelismo segue a mesma estratégia das máquinas hipercúbicas, distribuindo as frequências pelos processadores. Como as máquinas são de memória central, não é mais necessário que a seção em frequência, calculada em um processador, seja transmitida à memória dos demais processadores. Do mesmo modo, não há mais necessidade de comunicar as amostras de interesse aos demais processadores. Entretanto, como todos os processadores vão acumular os seus resultados numa estrutura comum, é necessário evitar *corridas*⁵ entre os processadores. Para isto, basta garantir exclusividade de acesso às posições concorridas, instalando um semáforo para cada profundidade. As tabelas abaixo apresentam os ganhos obtidos por este esquema para as duas máquinas.

Dimensao	numero de processadores					
	1	2	3	4	5	6
64X64	1,000	1,363	1,219	1,195	1,057	0,923
128X128	1,000	1,826	2,518	3,036	3,279	3,651
256X256	1,000	1,973	2,877	3,776	4,581	5,250
512X512	1,000	1,997	3,005	3,986	4,954	5,687

Tabela 6 - IBM 3090/600VF - Ganhos obtidos.

Dimensao	numero de processadores			
	1	2	3	4
64X64	1,000	0,956	0,771	0,645
128X128	1,000	1,636	1,725	1,683
256X256	1,000	1,960	2,671	3,202
512X512	1,000	1,975	2,901	3,742

Tabela 7 - IBM 9021/820 - Ganhos obtidos.

Os resultados permitem observar o aumento do ganho com o tamanho do problema, para um número fixo de processadores. Também apresentam limitações do paralelismo, como nos problemas de tamanho fixo em que o ganho não aumenta com o número de processadores. Por exemplo, no caso 64×64 nas duas máquinas, aumentar o número de processadores acima de um valor ótimo diminui o ganho e, conseqüentemente, aumenta o tempo de execução. No 9021, este valor ótimo é um processador, enquanto

⁵*race conditions*

no 3090 é dois processadores. Um outro exemplo é o caso 128×128 no 9021, que é executado mais rapidamente com três processadores.

Por outro lado, o custo do paralelismo que já era alto no IBM 3090 é ainda maior no IBM 9021. Observe-se que ganhos bons (acima de 80% do ótimo) para o número máximo de processadores, que eram atingidos com a dimensão 256×256 no IBM 3090, só são atingidos no IBM 9021 com a dimensão 512×512 . Isto nos leva a especular que o custo de paralelismo é diretamente proporcional ao desempenho das CPUs testadas, ou seja, quanto mais poderosa a CPU, maior o custo de associar múltiplas CPUs em um único programa.

6- EXPERIMENTOS EM REDES DE ESTAÇÕES DE TRABALHO

Avaliamos a paralelização da migração ω -x em uma rede com seis estações IBM RS-6000 320H, pertencentes à PETROBRÁS, conectadas fisicamente em rede Token-Ring e formando uma máquina paralela virtual através do PVM (Parallel Virtual Machine) [Sunderam90]. Tratam-se de estações de trabalho com desempenho da ordem de dezenas de Megaflops e processadores superescalares. Espera-se bom desempenho escalar (em função da arquitetura superescalar) mas o desempenho paralelo é imprevisível. A tabela a seguir apresenta os tempos de execução da versão seqüencial.

	64 tracos	127 tracos	256 tracos	512 tracos
Numero de Amostras				
64	1,45	2,89	5,64	11,17
128	5,88	11,69	23,00	44,95
256	23,64	46,97	93,00	181,26
512	94,99	187,37	376,01	736,49

Tabela 8 - RS-6000 320H - Tempos de execução (s) seqüencial.

O desempenho escalar é muito bom, coerente com o esperado. São processadores da ordem de 25 vezes mais rápidos que os do NCP, 3 vezes mais rápidos que os do iPSC, ligeiramente mais lentos que o 3090 e quatro vezes mais lentos que o 9021.

A estratégia de paralelização é idêntica à das máquinas hipercúbicas, com a mesma distribuição de tarefas pelos processadores. Assim, todas as comunicações envolvem o primeiro processador, quer distribuindo a seção em frequência gerada no início da computação, quer recebendo as acumulações parciais para totalização. As tabelas a seguir contém os ganhos em função do tamanho do problema.

	64 tracos	127 tracos	256 tracos	512 tracos
Numero de Amostras				
64	1,99	2,43	2,64	2,82
128	3,06	3,57	3,83	3,95
256	4,20	4,60	4,63	4,69
512	4,98	5,18	5,30	5,34

Tabela 9 - RS-6000/PVM - Ganhos com 6 estações.

Observam-se ganhos crescentes com o tamanho do problema e próximos ao ótimo para as maiores dimensões testadas. Notam-se também dependências de ganhos com n_x não previstas na análise de complexidade, o que merece atenção especial. A investigação desse fato levou este trabalho a novas conclusões.

Tal investigação iniciou-se instrumentando o programa de forma a medir separadamente os tempos de comunicação inicial, os tempos de processamento e os tempos de comunicação de resultados. As únicas discrepâncias observadas ocorrem nos tempos de comunicação inicial em problemas pequenos, que são muito superiores aos estimados. Em seguida, um conjunto de experimentos específicos demonstrou que o tempo de comunicação para mensagens pequenas é dominado pela latência, que não foi considerada na análise de complexidade.

Modelando o tempo de comunicação de uma mensagem de n bytes por $t(n) = c_0 + c_1 n$ e realizando experimentos específicos para medir as constantes c_0 e c_1 resultam em $c_0 = 3.90 \times 10^{-3}$ e $c_1 = 5.86 \times 10^{-7}$. Conseqüentemente, a latência domina o tempo de comunicação para mensagens de até centenas de elementos. Nestas condições, é necessário alterar a análise de complexidade, introduzindo a latência nas comunicações. Resulta a expressão a seguir:

$$S = \frac{k_0 n_t^2}{0.5c_0 \frac{n_t}{n_x} + 4c_1 n_t + \frac{k_0 n_t^2}{p} + k_3(p-1)n_t}$$

Nessa expressão, há uma única parcela do ganho que depende de n_x , o que não acontecia na expressão anterior. Conseqüentemente, o ganho passa a variar com n_x . Observe que a relevância dessa parcela tende a diminuir com o aumento do problema, pois é a parcela que varia mais lentamente na expressão (assumindo $n_t = n_x$, é constante no tamanho do problema enquanto as demais parcelas são linear e quadrática). De forma equivalente, a parcela diminui de importância com c_0 , ou seja, diminuindo a latência (o que ocorre substituindo a rede por hipercubos).

Por outro lado, as duas outras conclusões oriundas da análise da expressão anterior mantêm-se na nova formulação, ou seja, para qualquer número de processadores,

umentando o tamanho do problema obtém-se ganho ótimo, mas, para um problema de tamanho fixo, aumentar o número de processadores acima de um certo valor acarreta perda de desempenho. A rigor, é possível computar o número de processadores que produz ganho máximo, para um problema de tamanho fixo. Supondo $n_x = n_t = n$, pode-se escrever o ganho S como uma função de n e p e computar o ganho máximo em função de p para um valor fixo de n :

$$\frac{\partial S}{\partial p} = 0 \Rightarrow p = \sqrt{\frac{k_0}{k_3}} n$$

A disponibilidade da rede de estações permitiu avaliar, experimentalmente, os valores de k_0 e k_3 e, conseqüentemente, encontrar o número de processadores e o ganho máximo para cada tamanho de problema. A tabela a seguir contém estes valores:

n	p	S
64	9	3
128	13	6
256	19	9
512	27	13
1500	46	23

Tabela 10 - Número de processadores (p) que fornece ganho máximo (S) em função do tamanho do problema (n)

Infelizmente, não foi possível observar estes valores experimentalmente, pela falta de processadores.

7- CONCLUSÕES

A grande quantidade de experimentos realizados em múltiplas arquiteturas paralelas, produzindo ganhos consistentemente altos, demonstram a viabilidade do utilização do processamento paralelo na Migração Sísmica. Os principais argumentos são os ganhos máximos obtidos: 7,78 nas máquinas hipercúbicas com 8 processadores, 3,74 nas máquinas de memória central com 4 processadores e 5,34 nas redes com 6 estações.

Entretanto, este trabalho também apresenta limites e observações para o paralelismo:

- Fixo o número de processadores, existe sempre um problema suficientemente grande para o qual o ganho será ótimo.
- Fixo o tamanho de um problema existe um número de processadores para o qual o ganho é máximo, i.e., a partir de um certo número de processadores o ganho diminui.

A atualidade deste trabalho é confirmada por trabalhos correlatos como [Bell92], também em 2-D, e em [Black92a], [Black92b] e [Lynn92] em aplicações 3-D, que são

análogas. Os resultados em todos os casos são coerentes, mesmo utilizando múltipla gama de processadores, máquinas paralelas e algoritmos.

AGRADECIMENTOS

À PETROBRÁS pelo apoio dado a este trabalho especialmente no uso das máquinas não hipercúbicas.

REFERÊNCIAS BIBLIOGRÁFICAS

- [Bell92] Almasi, G.S., McLuckie, T., Bell, J., Gordon, A., and Hale, D. Parallel distributed seismic migration. Future Generation Computer Systems, V. 8, p 9-26, 1992.
- [Black92a] Black, J.L., Su, C.B. Networked parallel seismic computing. In: Offshore Technology Conference, 24, 1992. Houston. Proceedings ... Richardson, TX, Offshore Technology Conference, 1992. p. 169-176.
- [Black92b] Black, J.L., Su, C.B. Performance of parallel downward continuation. In: Annual SEG International Meeting, 62, 1992. New Orleans. Expanded Technical Program Abstracts. p. 326-329.
- [Claerbout85] Claerbout J.F. Imaging the earth's interior. Oxford, Blackwell Scientific Publications, 1985. 398 p. il.
- [Duarte85] Duarte, O.O. Processamento de reflexão sísmica Rio de Janeiro, Petrobrás, 1985. (apostila).
- [Hatton86] Hatton, L., Worthington, M.H., and Makin, J. Seismic data processing: theory and practice. Oxford, Blackwell Scientific Publications, 1986. 177p. il.
- [Loewenthal76] Loewenthal, D., Lu, L., Roberson, R., and Sherwood, J. The wave equation applied to migration. Geophysical Prospecting, V. 24, p. 380-399, 1976.
- [Lynn92] Lynn, W.S., Perkins, W., Cabrera J., and French, W.S. 3-D prestack imaging on massively parallel computers. In: Offshore Technology Conference, 24, 1992. Houston. Proceedings ... Richardson, TX, Offshore Technology Conference, 1992. p. 177-179.
- [Pinheiro93] Pinheiro, S.S. Investigação de Paralelismo na Migração W-X. Dissertação (Mestrado em Engenharia de Sistemas e Computação) - Coordenação dos Programas de Pós-Graduação em Engenharia, Universidade Federal do Rio de Janeiro, 1993.
- [Rosa93] Rosa, A.L.R. Migração de dados sísmicos. Rio de Janeiro, Petrobrás, 1993. (apostila).

- [Sunderam90] Sunderam, V.S. PVM: a framework for parallel distributed computing. Concurrency: practice and experience, V. 2, n. 4, p. 315-339, 1990.
- [Yilmaz87] Yilmaz, O. Seismic data processing. Tulsa, Society of Exploration Geophysicists, 1987. 526 p. il.