1. Introdução

Algoritmos paralelos são projetados segundo dois modelos de programação: paralelismo de função (ou de controle) e paralelismo de dados. No primeiro modelo, um programa paralelo é composto por um conjunto de tarefas diferentes, que implementam funções distintas, executadas concorrentemente nos diversos processadores. No modelo paralelismo de dados, um programa paralelo é caracterizado por um mesmo processo executando em diversos processadores sobre diferentes conjuntos de dados. Desta forma, o que define uma tarefa neste modelo é a partição dos dados sobre a qual o código comum deve ser executado.

O modelo paralelismo de dados, quando associado a arquiteturas MIMD, é conhecido como SPMD (Single Program, Multiple Data), no qual o mesmo programa é executado, sem sincronismo no nível de instrução, nos diferentes processadores, manipulando conjuntos distintos de dados (tarefas).

A adequada distribuição de tarefas nos diversos processadores é fundamental para a minimização do tempo de execução do programa paralelo e a maximização da utilização dos recursos computacionais disponíveis. A identificação desta distribuição ideal de tarefas caracteriza o problema de **balanceamento de carga**.

Um algoritmo de balanceamento de carga de uma aplicação paralela tem como objetivo, então, distribuir entre os processadores as tarefas que compõem a aplicação de forma a tentar minimizar o seu tempo de execução. Para tanto, o algoritmo de balanceamento de carga deve controlar o efeito de diferentes fatores de desequilíbrio, que ocorrem em diversas aplicações e arquiteturas, tais como: a heterogeneidade da arquitetura paralela adotada, o desconhecimento da quantidade de processamento envolvida em cada tarefa, a criação dinâmica de tarefas, a migração de tarefas, além da heterogeneidade e da variação da carga externa à aplicação nos diversos processadores de um ambiente paralelo não dedicado, adaptativo e não uniforme ou instável.

No caso do modelo SPMD, a estratégia de balanceamento de carga, freqüentemente, aparece inserida na própria aplicação, o que exige do projetista uma preocupação com a escolha da estratégia de balanceamento adequada. Se, por um lado, o mo delo SPMD tem atraído a atenção de muitos projetistas e pesquisadores por oferecer uma estrutura mais simples e clara de programação paralela, por outro, exige um esforço na identificação do algoritmo de balanceamen to de carga ideal à aplicação sendo desenvolvida.

O objetivo deste trabalho é avaliar o desempenho dos algoritmos descritos na seção 2 em aplicações específicas.

Na seção 3 apresentaremos os testes a serem realizados. A modelagem será apresentada na seção 4. A seção 5 visa fazer uma estimativa de com se comporta cada algoritmo com o aumento do número de processadores. Finalmente, são apresentadas na seção 6 as conclusões.

2. Algoritmos da Biblioteca

De acordo com a taxonomia, em relação à política de distribuição, uma estratégia de balanceamento de carga pode apresentar três comportamentos. Em uma estratégia estática, há uma distribuição inicial das tarefas seguida de sua execução. Em uma estratégia dinâmica sob demanda, a distribuição das tarefas ocorre ao longo da execução da aplicação de acordo com o comportamento dos processadores, tentando contornar os fatores dinâmicos de desequilíbrio. Em estratégias dinâmicas por migração, há uma distribuição inicial das tarefas com possíveis transferências destas ao longo da execução que visam desfazer os efeitos dos fatores dinâmicos de desequilíbrio. Estas últimas são caracterizadas por sua política de transferência.

Os algoritmos da biblioteca de balanceamento de carga são implementados em duas partes: a que se refere à política de distribuição e aquela que implementa a política de transferência (no caso de estratégias dinâmicas por migração).

2.1. Algoritmo #1 - Estático

Política de Distribuição

O processador mestre, que contém o conjunto inicial de tarefas, as distribui igualmen te entre todos os processadores, incluindo o próprio.

Política de Transferência

Não há política de transferência em estratégias estáticas. Após a distribuição das tarefas, estas são executadas sem que nenhuma ação visando o balanceamento de carga seja considerada.

2.2. Algoritmo #2 - Distribuído, Síncrono, Global, Por Even to, Coletivo e Isolado

Política de Distribuição

O processador mestre, que contém o conjunto inicial de tarefas, as distribui igualmente entre todos os processadores, incluindo o próprio.

Política de Transferência

Após a distribuição das tarefas, inicia-se a sua execução. Sempre que um processador encerra as sua tarefas, envia uma mensagem aos outros processadores solicitando a execução síncrona do balanceamen to de carga. Todos os processadores, então, inclusive o que solicitou o balanceamento de carga, enviam a todos os outros o seu índice de carga interna, no caso, a quan tidade de tarefas ainda não executadas.

Em seguida, todos os processadores executam, ao mesmo tempo, o algoritmo que define as transferências necessárias ao equilíbrio de carga. Calcula-se a média das cargas (**md**) de todos os processadores. O algoritmo continua somente se a carga de algum dos processadores encontra-se fora da faixa definida pelos valores **md-delta** e **md+delta**. Caso contrário, considera-se o balanceamento de carga desnecessário, pois a carga encontra-se relativamente equilibrada. O valor **delta** é definido a partir de um

parâmetro de ativação da aplicação. No início da aplicação, define-se que **delta** equivale a **P**% do número inicial de tarefas em cada processador. **P** é o parâmetro de ativação.

Todos os processadores definem então, diante da necessidade do balanceamento de carga, as transferências necessárias ao equilíbrio exato, ou seja, para que todos passem a ter a mesma carga interna. Finalmente, cada processador executa as transferências nas quais está envolvido.

Em aplicações SPMD nas quais tarefas são geradas dinamicamente, o parâmetro **P** deve ser definido com o valor 0, pois a proximidade do fim da aplicação não pode ser prevista pela carga aparente nos processadores.

Parâmetro de ativação: o valor de **P** .

2.3. Algoritmo #3 - Centralizado, Global, Por Evento, Coletivo e Isolado

Política de Distribuição

O processador mestre, que contém o conjunto inicial de tarefas, as distribui igualmente entre todos os processadores, incluindo o próprio.

Política de Transferência

Após a distribuição das tarefas, inicia-se a sua execução. Sempre que um processador encerra suas tarefas, envia uma mensagem aos outros processadores solicitando que estes enviem ao processador mestre o índice de carga interna; no caso, a quantidade de tarefas ainda não executadas. Este executará o algoritmo que define as transferências necessárias ao equilíbrio de carga.

Após receber o índice de carga de todos os processadores, inclusive do que solicitou o balanceamento de carga, o processador mestre inicia o algoritmo que define as transferências necessárias ao equilíbrio de carga. A média das cargas (**md**) é calculada. A execução do algoritmo continua apenas se a carga de algum dos processadores encontra-se fora da faixa definida pelos valores **md-delta** e **md+delta**. Caso contrário, considera-se o balanceamento de carga desnecessário, pois a carga encontra-se relativamente equilibrada. Neste caso, o mestre avisa aos escravos que não haverá transferência de tarefas. O valor **delta** é definido a partir de um parâmetro de ativação da aplicação. No início da aplicação, define-se que **delta** equivale a **P**% do número inicial de tarefas em cada processador. **P** é o parâmetro de ativação.

O mestre define então as transferências de carga necessárias ao equilíbrio exato da carga em todos os processadores e informa aos escravos as transferências nas quais estes devem participar. Em seguida, cada processador executa as transferências indicadas.

Em aplicações SPMD nas quais tarefas são geradas dinamicamente, o parâmetro **P** deve ser definido com o valor 0, pois a proximidade do fim da aplicação não pode ser prevista pela carga aparente nos processadores.

Parâmetro de ativação: o valor de **P** .

2.4. Algoritmo #4 - Distribuído, Síncrono, Global, Periódico, Coletivo e Isolado

Política de Distribuição

O processador mestre, que contém o conjunto inicial de tarefas, as distribui igualmente entre todos os processadores, incluindo o próprio.

Política de Transferência

Após a distribuição das tarefas, inicia-se a sua execução. O balanceamento de carga é ativado, periodicamente, em todos os processadores de forma síncrona. O período é definido pela quantidade de tarefas executadas em um dos processadores, no caso, o mestre. Este valor é um parâmetro de ativação da aplicação.

O mestre inicia o balanceamento de carga após executar a quantidade de tarefas definida como período, enviando uma mensagem aos outros processadores solicitando a execução síncrona da estratégia. Todos os processadores, então, inclusive o mestre, enviam a todos os outros o seu índice de carga in terna, no caso, a quan tidade de tarefas ainda não executadas.

Em seguida, todos os processadores executam, ao mesmo tempo, o algoritmo que define as transferências necessárias ao equilíbrio de carga. Calcula-se a média das cargas (**md**) de todos os processadores. O algoritmo continua somente se a carga de algum dos processadores encontra-se fora da faixa definida pelos valores **md-delta** e **md+delta**. Caso contrário, considera-se o balanceamento de carga desnecessário, pois a carga encontra-se relativamente equilibrada. O valor **delta** é definido a partir de um parâmetro de ativação da aplicação. No início da aplicação, define-se que **delta** equivale a **P**% do número inicial de tarefas em cada processador. **P** é o parâmetro de ativação.

Todos os processadores definem então, diante da necessidade do balanceamento de carga, as transferências necessárias ao equilíbrio exato, ou seja, a que todos passem a ter a mesma carga interna. Em seguida, cada processador executa as transferências nas quais está envolvido.

Em aplicações SPMD nas quais tarefas são geradas dinamicamente, o parâmetro **P** deve ser definido com o valor 0, pois a proximidade do fim da aplicação não pode ser prevista pela carga aparente nos processadores.

Parâmetros de ativação: o valor de **P** e a quantidade de tarefas que define o período.

2.5. Algoritmo #5 - Distribuído, Síncrono, Local, Particionado, Por Evento, Coletivo e Isolado

Neste algoritmo, as transferências de tarefas ocorrem somente entre processadores de um mesmo grupo. Cada processador pertence somente a um grupo, sendo então todos disjuntos. Os grupos são definidos pelo usuário em um arquivo, cujo nome é informado como parâmetro de ativação da aplicação. A primeira linha do arquivo contém o número de grupos - **n** . Cada grupo possui um identificador inteiro - 0..**n**-1. Considere **p** pro cessadores, com identificadores 0..**p**-1. A segunda linha do arquivo contém o identificador do grupo do processador 0 . A terceira linha, o identificado do grupo do processador 1 . A **i**-ésima linha iden tifica então o grupo do processador **i**-2 .

O conteúdo do arquivo exemplificado a seguir especifica que os oito processadores estão organizados em dois grupos. Os quatro primeiros, no grupo 0 , e os demais, no grupo 1.

2\n
0\n
0\n
0\n
0\n
1\n
1\n
1\n
1\n

Política de Distribuição

O processador mestre, que contém o conjunto inicial de tarefas, as distribui igualmente entre todos os processadores, incluindo o próprio.

Política de Transferência

Após a distribuição das tarefas, inicia-se a sua execução. Sempre que um processador conclui as suas tarefas, envia uma mensagem aos outros processadores do mesmo grupo, solicitando a execução síncrona do balanceamento de carga. Estes processadores, então, inclusive o que solicitou o balanceamento de carga, enviam a todos os outros do grupo o seu índice de carga interna, no caso, a quantidade de tarefas ainda não executadas.

Em seguida, todos os processadores do grupo executam, ao mesmo tempo, o algoritmo que define as transferências necessárias ao equilíbrio de carga. Calcula-se a média das cargas (**md**) dos processadores do grupo. O algoritmo continua somente se a carga de algum encontra-se fora da faixa definida pelos valores **md-delta** e **md+delta**. Caso contrário, considera-se o balanceamento de carga desnecessário, pois a carga encontra-se relativamente equilibrada. O valor delta é definido a partir de um parâmetro de ativação da aplicação. No início da aplicação, define-se que delta equivale a **P**% do número inicial de tarefas em cada processador. **P** é o parâmetro de ativação.

Os processadores do grupo definem então, diante da necessidade do balanceamento de carga, as transferências necessárias ao equilíbrio exato, ou seja, a que todos passem a ter a mesma carga interna. Em seguida, cada processador executa as transferências nas quais está envolvido.

Em aplicações SPMD nas quais tarefas são geradas dinamicamente, o parâmetro **P** deve ser definido com o valor 0, pois a proximidade do fim da aplicação não pode ser prevista pela carga aparente nos processadores.

Parâmetros de ativação: o nome do arquivo que contém a definição do agrupamento e o valor de **P**.

3. Experimentos

Serão apresentados nesta seção os testes que foram realizados utilizando o computador IBM/SP2 do Laboratório Nacional de Computação Científica (LNCC) com 4, 8 e 16 processadores.

Serão analisados os cinco algoritmos apresentados na seção 2 quando aplicados a dois problemas. Este problemas são:

- ***Problema 1***: Solução da integral $\int_a^b x^2$ através do método da quadratura;
- ***Problema 2:*** Solução da integral $\int_a^b e^x$ através do método da quadratura.

O método da quadratura será apresentado a seguir.

3.1. Método da Quadratura

Seja $f(x)$ uma função contínua e não-negativa. Sejam l e r dois valores tal que $l < r$. O problema da quadratura consiste em estabelecer a área limitada por $f(x)$, o eixo x, e as linhas verticais $x = l$ e $x = r$. Daí, o problema é atingir, aproximadamente, a integral de $f(x)$ de l a r.

A maneira típica de alcançar a área aproximada sob uma curva é dividir o intervalo $[l, r]$ em uma série de subintervalos. Em seguida utiliza-se um trapézio para conseguir a área aproximada de cada subintervalo. Em particular, seja $[a, b]$ o subintervalo. Então, a área aproximada sob f de a a b é a área do trapézio com base b-a e lados de altura $f(a)$ e $f(b)$.

O problema da quadratura pode ser resolvido tanto estaticamente quanto dinamicamente. Quando resolvido estaticamente, é usado um número fixo de intervalos de tamanhos iguais. Daí, computa-se a área de cada subintervalo e em seguida soma-se os resultados. Este processo é então repetido – tipicamente, dobrando o número de intervalos – até duas aproximações sucessivas serem próximas o bastante para serem aceitas.

Quando feita dinamicamente, inicia-se com um intervalo de l a r e computa-se o ponto médio m entre l e r. Calcula-se, então, as áreas de três trapézios.

1.O maior, limitado por l, r, $f(l)$ e $f(r)$;
2.O limitado por l, m, $f(l)$ e $f(m)$;
3.O limitado por m, r, $f(m)$ e $f(r)$.

Em seguida, compara-se a área do trapézio maior com a soma das áreas dos dois trapézios menores. Se estas áreas forem próximas o suficiente (diferença menor que um determinado ε), aceita-se a área do trapézio maior como uma boa aproximação da área sob f. Caso contrário, o processo é repetido recursivamente até que a solução de cada subproblema seja aceitável. Soma-se, então, a resposta de cada subproblema para se obter a resposta final.

Tanto com a maneira estática quanto com a maneira dinâmica, todo subproblema é independente dos outros. Assim, a aplicação pode ser facilmente paralelizada. Neste trabalho, utilizou-se a maneira dinâmica para nossa aplicação. Inicia-se a aplicação com um único processo p_1 que calcula as três áreas descritas acima. Se a área do trapézio maior for próxima o suficiente à soma das áreas dos trapézios menores, então chegamos ao fim da aplicação com a resposta igual a área do trapézio maior. Caso contrário, dá-se

início a um novo processo p_2. Este processo será responsável pela área aproximada sob f de m a r, onde m é o ponto médio entre l e r. O processo p_1 agora fica responsável pela área aproximada sob f de l a m. Esse processo é repetido recursivamente até que as áreas de cada intervalo sejam aceitáveis.

Uma característica presente nesta aplicação é a indeterminação do número de processos, ou seja, não é possível prever quantos processos serão gerados no decorrer da aplicação.

3.2. Definição dos testes

Na análise de desempenho da aplicação da quadratura, será utilizado um fator de desequilíbrio presente na aplicação. Este fator é a criação dinâmica de processos.

Nos experimentos pretende-se obter a valor das seguintes variáveis: y_{lm}, onde:

- y_{lm} é o tempo gasto pelo algoritmo l para resolver a instância m da aplicação de quadratura. Onde $1 \leq l \leq 5$ e $1 \leq m \leq 3$(número de níveis do fator processadores).

4. MODELAGEM

Nesta seção será realizada a análise dos dados obtidos através dos experimentos realizados.

4.1. Análise do Primeiro Experimento

Este experimento visa resover a integral $\int_a^b x^2$ através do método da quadratura. Os parâmetros utilizados foram os seguintes:

- a : 0;
- b : 100000
- ε : 0.0000001 (parâmetro da quadratura)

4.1.1. Modelo

O modelo que será utilizado para esta análise será o *fatorial completo para dois parâmetros sem repetição*.

O modelo matemático usado é representado pela seguinte fórmula:

$y_{ij} = \mu + \alpha_j + \beta_i + e_{ij}$, onde

- y_{ij} = efeito da observação com o fator número de processadores no nível j e com o fator algoritmo no nível i.
- μ = média das possíveis observações de y_{ij}.
- α_j = efeito do fator número de processadores no nível j.
- β_i = efeito do fator algoritmo no nível i.
- e_{ij} = efeito do erro.

4.1.2. Cálculo dos Estimadores do Modelo

Os valores dos estimadores do modelo devem ser calculados de forma que a média dos erros seja nula. Para isso, serão calculados:

- para cada linha, a média das observações ($\breve{y}_i$);
- para cada coluna, a média das observações ($\breve{y}_j$);
- a média total ($\breve{y}$);
- a diferença, $\breve{y}_i - \breve{y}$, que representa o efeito da linha correspondente (β_i);
- a diferença, $\breve{y}_j - \breve{y}$, que representa o efeito da coluna correspondente (α_j).

Consideraremos que $\mu = \bar{y}$.

A tabela a seguir apresenta os resultados.

	Número de Processadores				
Algoritmos	4	8	16	Média	Efeito(β)
1	106	53	28	63.33	-176.8
2	162	163	248	191	-48.13
3	160	82	43	95	-144.13
4	1050	704	483	745.67	506.53
5	154	112	39	101.67	-137.47
Média	326.4	222.8	168.2	239.13	
Efeito(α)	87.27	-16.33	-70.93		

A tabela acima pode ser interpretada da seguinte maneira:

- Média Total ($\bar{y}$): nos experimentos, foram gastos, em média, 239.13 s;
- Estimador α: Com o uso de 4 processadores foram gastos, em média, 326.4 s. Com 8 processadores, 222.8 s. E com 16 processadores, 168.2 s.
- Estimador β: Utilizando o algoritmo 1 foram gastos, em média, 63.33 s. Com o algoritmo 2, 191 s. Com o algoritmo 3, 95 s. Com o algoritmo 4, 745.67 s. E com o algoritmo 5, 101.67 s.

4.1.3. Erros Experimentais

Com os estimadores calculados, o próximo passo é calcular a resposta estimada esperada do experimento, que é dada pela seguinte fórmula:

$yEsp_{ij} = \mu + \alpha_j + \beta_i$.

A diferença entre a resposta estimada e a resposta medida é atribuída a erros experimentais. Ou seja,

$e_{ij} = y_{ij} - yEsp_{ij} = y_{ij} - (\mu + \alpha_j + \beta_i)$.

A tabela de erros é apresentada a seguir.

	Número de Processadores		
Algoritmos	4	8	16
1	-43.6	7	36.6
2	-116.27	-11.67	127.93
3	-22.27	-11.67	127.93
4	217.07	-25.33	-191.73
5	-34.93	26.67	8.27

A variância dos erros pode ser estimada através da Soma dos Quadrados dos Erros (*SSE*). Calculando *SSE*, tem-se:

$SSE = \sum_{i=1}^{5}\sum_{j=1}^{3} e_{ij}^2$, logo:

$SSE = 120697.1$

4.1.4. Variância e Coeficiente de Determinação

A variância total de *y* pode ser atribuída aos dois fatores e ao erro experimental. Para isto, eleva-se ao quadrado os dois lados da equação do modelo e soma-se todas as observações. Assim, tem-se:

$$\sum_{ij} y_{ij}^2 = 15\mu^2 + 5\sum_{j}\alpha_j^2 + 3\sum_{i}\beta_i^2 + \sum_{ij} e_{ij}^2 .$$

Ou seja,

$SSY = SS\mu + SS\alpha + SS\beta + SSE.$

A variação total (*SST*) pode então ser especificada como:

$SST = SSY - SS\mu = SS\alpha + SS\beta + SSE.$

Logo, a variação total pode ser dividida em partes explicadas pelos fatores α (processadores) e β (algoritmos), e uma parte que é devido a erros experimentais. Então:

$SSY = \sum_{ij} y_{ij}^2 = 2032505$

$SS\mu = 15 * (239.13)^2 = 857771.3.$

$SS\alpha = 5\sum_{j}\alpha_j^2 = 64568.93.$

$SS\beta = 3\sum_{i}\beta_i^2 = 989467.7.$

$SST = 1174734.$

Pode-se, agora, calcular a porcentagem de variação explicada por cada fator. A porcentagem explicada pelo número de processadores é:

$100 * (SS\alpha / SST) = 5.50\%$

A porcentagem de variação explicada pelo fator algoritmo é:

$100 * (SS\beta / SST) = 84.23\%$

A porcentagem de variação não explicada pelo modelo é:

$100 * (SSE / SST) = 10.27\%$

O coeficiente de determinação (R^2) pode ser então calculado.

$R^2 = 100 * [1 - (SSE / SST)] = 89.73\%$

Analisando estas porcentagens, percebe-se que o fator algoritmo é importante para um bom tempo da aplicação. O que não foi percebido na análise do fator processadores. Também nota-se que o modelo apresentou um coeficiente de determinação elevado, indicando que o modelo está adequado para a análise.

4.1.5. Análise das Variâncias

Para testar estatisticamente a significância de um parâmetro, primeiramente deve-se dividir a sua soma de quadrados pelo seu grau de liberdade, obtendo, assim, a média dos quadrados (*MQ*). Então:

$MS\alpha = SS\alpha / (3-1)$,

$MS\beta = SS\beta / (5-1)$,

$MSE = SSE / ((3-1)\times(5-1))$.

Para testar a significância de um parâmetro, utilizamos o teste de *F* (Distribuição *F*). Que é definido como:

$F_\alpha = MS\alpha / MSE \quad e \quad F_\beta = MS\beta / MSE.$

Um parâmetro é considerado significante se o seu *F* calculado for maior que o *F* obtido de uma tabela padronizada de variações de *F* (tabelas A6, A7 e A8 do livro do Jain). Assim:

Parämetro	S.S.	%Var	G.L	M.S.	F
y	2032505				
$\check{y}$	857771.3				
y - $\check{y}$	1174734	100%	14		
Nº de processadores	64568.93	5.50%	2	32284.47	2.14
Algoritmo	989467.7	84.23%	4	247366.90	16.40
Erro	120697.1	10.27%	8	15087.13	

De acordo com os dados da tabela acima, podemos perceber que o fator algoritmo é significante para o modelo a um grau de 99%, e o fator processador tem uma significância inferior a 90%, pois:

$2.14 < 3.11 \Rightarrow F_\alpha < F_{[\,90\%,\,2,\,8]}$ e

$16.40 > 7.01 \Rightarrow F_\beta > F_{[\,99\%,\,4,\,8]}$.

4.1.6. Intervalos de Confiança

Os intervalos de confiança dos efeitos do modelo podem ser calculados usando os valores de *t* da distribuição de Student para os graus de liberdade do erro. Primeiramente, necessita-se calcular as variâncias e os desvios padrões dos estimadores. Então:

$S_e = \sqrt{MSE} = 122.83$

Parâmetro	Estimador	Variância	Desvio-padrão
μ	$\check{y}$	$S_e^2 / 15 = 1005.81$	31.71
α_j	$\check{y}_j - \check{y}$	$2 * S_e^2 / 15 = 2011.62$	44.85
β_i	$\check{y}_i - \check{y}$	$4 * S_e^2 / 15 = 4023.24$	63.43

O valor da variável *t* de Student para 8 graus de liberdade é:

$t_{[0.95,\,8]} = 1.86$

Os intervalos de confiança são calculados através da seguinte fórmula:

$Ic = \gamma \pm \tau \,.\, t$, onde

Ic = intervalo de confiança,

γ = parâmetro,

τ = desvio-padrão do parâmetro.

A tabela abaixo apresenta o intervalo de confiança de cada parâmetro.

Parâmetro	Média	Desvio-padrão	Intervalo de Confiança
μ	239.13	31.71	[180.14; 298.12]
Número de processadores			
4	326.40	44.85	[242.12; 409.82]

8	222.80	44.85	[139.37; 306.22]
16	168.20	44.85	[84.77; 251.62]
Algoritmo			
1	62.33	63.43	[-55.64; 180.31]
2	191	63.43	[73.02; 308.98]
3	95	63.43	[-22.98; 212.98]
4	745.67	63.43	[627.69; 863.64]
5	101.67	63.43	[-16.31; 219.64]

4.2. Análise do Segundo Experimento

Este experimento visa resover a integral $\int_a^b e^x$ através do método da quadratura. Os parâmetros utilizados foram os seguintes:

- a : 0;
- b : 30
- ε : 0.00000001 (parâmetro da quadratura)

4.2.1. Modelo

O modelo que será utilizado para esta análise será o *fatorial completo para dois parâmetros sem repetição*.

O modelo matemático usado é representado pela seguinte fórmula:

$y_{ij} = \mu + \alpha_j + \beta_i + e_{ij}$, onde

- y_{ij} = efeito da observação com o fator número de processadores no nível j e com o fator algoritmo no nível i.
- μ = média das possíveis observações de y_{ij}.
- α_j = efeito do fator número de processadores no nível j.
- β_i = efeito do fator algoritmo no nível i.
- e_{ij} = efeito do erro.

4.2.2. Cálculo dos Estimadores do Modelo

Os valores dos estimadores do modelo devem ser calculados de forma que a média dos erros seja nula. Para isso, serão calculados:

- para cada linha, a média das observações ($\breve{y}_i$);

- para cada coluna, a média das observações ($\bar{y}_j$);
- a média total ($\bar{y}$);
- a diferença, $\bar{y}_i - \bar{y}$, que representa o efeito da linha correspondente (β_i);
- a diferença, $\bar{y}_j - \bar{y}$, que representa o efeito da coluna correspondente (α_j).

Consideraremos que $\mu = \bar{y}$.

A tabela a seguir apresenta os resultados.

	Número de Processadores				
Algoritmos	4	8	16	Média	Efeito(β)
1	928	922	461	770.33	285.13
2	399	301	271	323.67	-161.53
3	203	289	150	214	-271.20
4	664	732	701	699	213.80
5	747	315	195	419	-66.20
Média	588.20	511.80	355.60	485.20	
Efeito(α)	103	26.60	-129.60		

A tabela acima pode ser interpretada da seguinte maneira:

- Média Total ($\bar{y}$): nos experimentos, foram gastos, em média, 485.20 s;
- Estimador α: Com o uso de 4 processadores foram gastos, em média, 588.20 s. Com 8 processadores, 511.80 s. E com 16 processadores, 355.60 s.
- Estimador β: Utilizando o algoritmo 1 foram gastos, em média, 770.33 s. Com o algoritmo 2, 323.67 s. Com o algoritmo 3, 214 s. Com o algoritmo 4, 699 s. E com o algoritmo 5, 419 s.

4.2.3. Erros Experimentais

Com os estimadores calculados, o próximo passo é calcular a resposta estimada esperada do experimento, que é dada pela seguinte fórmula:

$$yEsp_{ij} = \mu + \alpha_j + \beta_i.$$

A diferença entre a resposta estimada e a resposta medida é atribuída a erros experimentais. Ou seja,

$$e_{ij} = y_{ij} - yEsp_{ij} = y_{ij} - (\mu + \alpha_j + \beta_i).$$

A tabela de erros é apresentada a seguir.

	Número de Processadores		
Algoritmos	4	8	16
1	54.67	125.07	-179.73
2	-27.67	-49.27	76.93
3	-114	48.40	65.60
4	-138	6.40	131.60
5	225	-130.60	-94.40

A variância dos erros pode ser estimada através da Soma dos Quadrados dos Erros (*SSE*). Calculando *SSE*, tem-se:

$SSE = \sum_{i=1}^{5}\sum_{j=1}^{3} e_{ij}^2$, logo:

$SSE = 192683.7$

4.2.4. Variância e Coeficiente de Determinação

A variância total de *y* pode ser atribuída aos dois fatores e ao erro experimental. Para isto, eleva-se ao quadrado os dois lados da equação do modelo e soma-se todas as observações. Assim, tem-se:

$$\sum_{ij} y_{ij}^2 = 15\mu^2 + 5\sum_{j}\alpha_j^2 + 3\sum_{i}\beta_i^2 + \sum_{ij} e_{ij}^2 .$$

Ou seja,

$SSY = SS\mu + SS\alpha + SS\beta + SSE.$

A variação total (*SST*) pode então ser especificada como:

$SST = SSY - SS\mu = SS\alpha + SS\beta + SSE.$

Logo, a variação total pode ser dividida em partes explicadas pelos fatores α (processadores) e β (algoritmos), e uma parte que é devido a erros experimentais. Então:

$SSY = \sum_{ij} y_{ij}^2 = 4557642.$

$SS\mu = 15 * (239.13)^2 = 3531286.$

$SS\alpha = 5\sum_{j}\alpha_j^2 = 140563.60.$

$SS\beta = 3 \sum_i \beta_i^2 \;\; = \;\; 693109.10.$

$SST = \;\; 1026356.$

Pode-se, agora, calcular a porcentagem de variação explicada por cada fator. A porcentagem explicada pelo número de processadores é:

$100 * (SS\alpha / SST) = \;\; 13.69\%$

A porcentagem de variação explicada pelo fator algoritmo é:

$100 * (SS\beta / SST) = \;\; 67.53\%$

A porcentagem de variação não explicada pelo modelo é:

$100 * (SSE / SST) = 18.77\%$

O coeficiente de determinação (R^2) pode ser então calculado.

$R^2 = 100 * [\; 1 - (SSE / SST)] = 81.23\%$

Analisando estas porcentagens, percebe-se que o fator algoritmo é importante para um bom tempo da aplicação. O que não foi percebido na análise do fator processadores. Também nota-se que o modelo apresentou um bom coeficiente de determinação, indicando que o modelo está adequado para a análise.

4.2.5. Análise das Variâncias

Para testar estatisticamente a significância de um parâmetro, primeiramente deve-se dividir a sua soma de quadrados pelo seu grau de liberdade, obtendo, assim, a média dos quadrados (*MQ*). Então:

$MS\alpha = SS\alpha / (3\text{-}1),$

$MS\beta = SS\beta / (5\text{-}1),$

$MSE = SSE / ((3\text{-}1)\times(5\text{-}1)).$

Para testar a significância de um parâmetro, utilizamos o teste de *F* (Distribuição *F*). Que é definido como:

$F_\alpha = MS\alpha / MSE \qquad e \qquad F_\beta = MS\beta / MSE.$

Um parâmetro é considerado significante se o seu *F* calculado for maior que o *F* obtido de uma tabela padronizada de variações de *F* (tabelas A6, A7 e A8 do livro do Jain). Assim:

Parämetro	S.S.	%Var	G.L	M.S.	F
y	4557642				
$\breve{y}$	3531286				
y - $\breve{y}$	1026356	100%	14		
Nº de processadores	140563.60	13.69%	2	70281.80	2.92
Algoritmo	693109.10	67.53%	4	173277.30	7.19
Erro	192683.70	18.77%	8	24085.47	

De acordo com os dados da tabela acima, podemos perceber que o fator algoritmo é significante para o modelo a um grau de 99%, e o fator processador tem uma significância inferior a 90%, pois:

$2.92 < 3.11 \Rightarrow F_\alpha < F_{[\,90\%,\,2,\,8]}$ e

$7.19 > 7.01 \Rightarrow F_\beta > F_{[\,99\%,\,4,\,8]}$.

4.2.6. Intervalos de Confiança

Os intervalos de confiança dos efeitos do modelo podem ser calculados usando os valores de *t* da distribuição de Student para os graus de liberdade do erro. Primeiramente, necessita-se calcular as variâncias e os desvios padrões dos estimadores. Então:

$S_e = \sqrt{MSE} = 155.19$

Parâmetro	Estimador	Variância	Desvio-padrão
μ	$\breve{y}$	$S_e^2 / 15 = 1605.70$	40.07
α_j	$\breve{y}_j - \breve{y}$	$2 * S_e^2 / 15 = 3211.40$	56.67
β_i	$\breve{y}_i - \breve{y}$	$4 * S_e^2 / 15 = 6422.79$	80.14

O valor da variável *t* de Student para 8 graus de liberdade é:

$t_{[0.95,\,8]} = 1.86$

Os intervalos de confiança são calculados através da seguinte fórmula:

$Ic = \gamma \pm \tau \,.\, t$, onde

Ic = intervalo de confiança,

γ = parâmetro,

τ = desvio-padrão do parâmetro.

A tabela abaixo apresenta o intervalo de confiança de cada parâmetro.

Parâmetro	Média	Desvio-padrão	Intervalo de Confiança
μ	485.20	40.07	[410.67; 559.73]
Número de processadores			
4	588.20	56.67	[482.79; 693.60]
8	511.80	56.67	[406.39; 617.20]
16	355.60	56.67	[250.19; 461.00]
Algoritmo			
1	770.33	80.14	[621.27; 919.40]
2	323.67	80.14	[174.60; 472.73]
3	214	80.14	[64.94; 363.06]
4	699	80.14	[549.93; 848.06]
5	419	80.14	[269.94; 568.06]

5. ESTIMANDO RESPOSTAS

Esta seção visa definir uma função do tipo:

$$y = b_0 + b_1 x,$$

para cada agoritmo da análise do primeiro e do segundo experimento apresentados na seção anterior. Esta reta serviria como estimativa do tempo *y* gasto por um algoritmo para resolver uma das aplicações usando *x* processadores.

É mostrado (livro do Jain – Regressão Linear) que os coeficientes da reta podem ser calculados como descrito a seguir:

$$b_1 = \frac{\sum xy - n\overline{xy}}{\sum x^2 - n(\overline{x})^2} \text{ e}$$

$b_0 = \overline{y} - b_1 \overline{x}$, onde

$$\overline{x} = \frac{1}{n}\sum_1^n x_i ,$$

$$\overline{y} = \frac{1}{n}\sum_1^n y_i ,$$

$$\sum xy = \sum_{i=1}^{n} x_i y_i \text{ e}$$

$$\sum x^2 = \sum_{i=1}^{n} x_i^2 .$$

O coeficiente de determinação (R^2) indica o quanto a reta obtida representa o comportamento do algoritmo. Ele pode ser calculado pela seguinte fórmula:

R^2 = (SST – SSE) / SST, onde

$$\text{SST} = \sum_{i=1}^{n} (y_i - \overline{y})^2 \text{ e}$$

$$\text{SSE} = \sum y^2 - b_0 \sum y - b_1 \sum xy .$$

A tabela a seguir apresenta as retas e o coeficiente de determinação para cada algoritmo analisado.

Primeiro Experimento		
Algoritmo	Reta	R^2
1	$y = 118.50 – 6.02x$	0.85
2	$y = 119.50 + 7.66x$	0.90
3	$y = 179.50 – 9.05x$	0.86
4	$y = 1160.50 – 44.45x$	0.90
5	$y = 190.50 – 9.52x$	0.99
Segundo Experimento		
Algoritmo	Reta	R^2
1	$y = 1158.50 – 41.59x$	0.89
2	$y = 414.00 – 9.68x$	0.78
3	$y = 272.50 – 6.27x$	0.30
4	$y = 679.50 + 2.01x$	0.25
5	$y = 807.00 – 41.57x$	0.76

6. CONCLUSÕES

Os resultados obtidos mostraram que o fator algoritmo teve uma grande participação no tempo da aplicação. A escolha de um bom algoritmo pode acarretar em uma excelente economia de tempo.

Já o fator processador, para estas aplicações, não mostrou uma grande significância no tempo da aplicação. Talvez, se fossem feitos testes em uma aplicação com tarefas mais duradouras, os resultados poderiam ser diferentes.

No primeiro experimento, os processadores já ficavam naturalmente balanceados, o que favoreceu o algoritmo estático (1) que não gastava tempo com comunicação. Já os outros algoritmos gastavam um certo tempo verificando se a aplicação estava balanceada.

No segundo experimento, os processadores ficavam completamente desbalanceados, o que favoreceu os algoritmos que faziam balanceamento dinâmico.