

# SisA<sup>3</sup>: Uma arquitetura HW/SW para Auditoria em Armazéns de Granéis \*

Wagner G. Al-Alam<sup>1</sup>, Renata H. S. Reiser<sup>1</sup>, Adenauer C. Yamin<sup>1</sup>, Maurício L. Pilla<sup>2</sup>

<sup>1</sup> Programa de Pós-Graduação em Informática Mestrado em Ciência da Computação  
Universidade Católica de Pelotas – RS – Brasil  
{alalam, reiser, adenauer}@ucpel.tche.br

<sup>2</sup> Departamento de Informática/IFM  
Universidade Federal de Pelotas – RS – Brasil  
pilla@ufpel.edu.br

**Abstract.** Current audit processes in the content of grain stores are frequently based on emptying the store. Besides, they are also often error prone. The SisA<sup>3</sup> Automated Audit System intends to improve this process while keeping the grain in its current storage. It is comprised of: (i) a digital equipment to measure points in the non-uniform relief of the stored grain; and (ii) a volume calculation software (iCone-SisA<sup>3</sup>) which combines numeric interpolation of measured points with iterative volume calculation. In this paper, the architecture of SisA<sup>3</sup> is presented, as well as a parallel version of iCone-SisA<sup>3</sup> in the Distributed Geometric Machine (D-GM) environment. The latter provides formalisms in order to model and execute correctly the parallel steps and the required synchronization. The software prototype for iCone-SisA<sup>3</sup> is functionally validated and its performance results are presented. The SisA<sup>3</sup> will allow the fast and precise assessment of silos, with a broad application in the market.

**Keywords:** Architecture of Computing Systems; Numeric Methods; Parallel Processing

## 1 Introdução

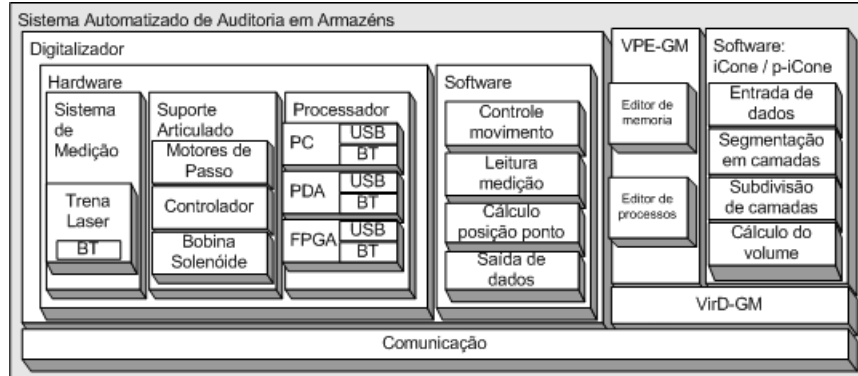
Este trabalho introduz o Sistema Automatizado de Auditoria em Armazéns denominado SisA<sup>3</sup>, consistindo numa ferramenta integrada de hardware e software, organizada em dois subsistemas: o sistema digitalizador *Digital SisA<sup>3</sup>* e o método numérico caracterizando o sistema *iCone-SisA<sup>3</sup>*, provendo tanto automação baseada em digitalização na coleta de dados obtidos por processo de medição a *laser*, como a solução computacional para o problema de cálculo do volume em superfícies não uniformes. A consolidação do protótipo p-iCone visa o ganho de desempenho pela exploração do paralelismo intrínseco à aplicação iCone, adicionando recursos que possibilitam a explicitação do paralelismo com validação de parâmetros via representação visual e o controle de execução dos processos.

\* Projeto parcialmente financiado pelo CNPq, referente aos processos: Edital Universal 476933/2007-2 e Apoio Técnico 502999/2008-0.

Este artigo está organizado da seguinte forma: na Seção 2, apresenta-se a arquitetura integrada de software e hardware do sistema *SisA*<sup>3</sup> e segue-se a organização do sistema *Digital SisA*<sup>3</sup> na Seção 3. Na Seção 4, descreve-se o modelo matemático incluindo a discretização e ajustes dos dados de entrada. A Seção 5 apresenta a solução numérica para cálculo do volume do relevo, com modelagem no VPE-GM e execução gerenciada pelo VirD-GM. Logo após, os resultados são avaliados na Seção 6, incluindo as etapas de testes e de análise do erro associado ao desenvolvimento dos protótipos. Finalmente, conclusões e trabalhos futuros são discutidos na Seção 7.

## 2 Apresentação do Sistema *SisA*<sup>3</sup>

O Sistema Automatizado de Auditoria em Armazéns (*SisA*<sup>3</sup>) consiste numa proposta para viabilizar a avaliação do volume de estoques da produção agrícola armazenada em silos (cônicos ou horizontais). De acordo com a Figura 1, a arquitetura proposta considera a integração de dois produtos: (i) o método numérico *iCone-SisA*<sup>3</sup>, desenvolvido sob a ótica de software para desempenhar a tarefa de cálculo do volume estocado; e (ii) o equipamento digitalizador *Digital SisA*<sup>3</sup> responsável por efetuar medições no relevo e gerar a entrada de dados para execução do método *iCone-SisA*<sup>3</sup>. A integração destes produtos possibilita o cálculo computacional do volume sem que exista contato com os produtos armazenados, disponibilizando uma ferramenta comercial para auxiliar empresas e órgãos de controle do poder público ou privado na realização de auditorias de estoque.



**Figura 1.** Visão Arquitetural do Sistema *SisA*<sup>3</sup>.

Considera-se o projeto D-GM (*Distributed Geometric Machine*), disponibilizando o ambiente de desenvolvimento VPE-GM (*Visual Programming Environment for the Geometric Machine Model*) [1] e o de execução distribuída VirD-GM (*Virtual Distributed Geometric Machine Model*) [2][3] para suporte ao desenvolvimento do sistema *SisA*<sup>3</sup>.

O VPE-GM adiciona recursos que possibilitam a modelagem e seleção das etapas de paralelização, incluindo representação visual, validação de parâmetros e controle dos processos na etapa de desenvolvimento, de forma a satisfazer as necessidades do protótipo. O VirD-GM é responsável por receber os arquivos descritores exportados pelo ambiente VPE-GM, coordenando a leitura das estruturas de processos, de memória

e dos parâmetros para execução, e gerenciando o controle da execução da computação baseado nos dados obtidos junto às matrizes de adjacências. Neste contexto, o protótipo p-iCone consiste em um caso de estudo que colabora com a validação do ambiente de execução VirD-GM.

O desenvolvimento da metodologia científica para modelagem computacional e cálculo do volume de relevos em superfícies não-uniformes consiste num dos principais desafios deste trabalho. Instigados por esta questão, consideram-se estudos concentrados em duas grandes áreas de pesquisa: (i) a computação científica, baseada em técnicas, métodos e processos do cálculo numérico para o desenvolvimento de software numérico no âmbito científico e tecnológico [4,5,6,7], fundamentando a modelagem matemática para o problema de cálculo do relevo em superfícies não-uniformes e considerando a análise dos erros para os métodos numéricos empregados; (ii) o processamento paralelo e distribuído, que na continuidade ao projeto iCone [8], introduz uma solução mais disponível e acessível para simulação do cálculo do relevo em superfícies não-uniformes, consolidando o projeto p-iCone pela melhoria de desempenho no processamento numérico caracterizado como intensivo e de elevada complexidade.

Baseada na metodologia desenvolvida, o protótipo aplica o cálculo analítico do volume identificado a partir de uma matriz de representação, onde aproximações de medições intermediárias são obtidas através da interpolação de pontos, utilizando *splines* cúbicas. A partir do acesso aos pontos desta matriz, torna-se possível a modelagem do relevo e a aplicação de algoritmos para simulação do cálculo do volume. Neste caso, na computação do volume, consideram-se desde parâmetros relativos a distâncias, ângulos, áreas e volumes parciais até a obtenção do resultado final.

### 3 Descrição do Digitalizador

Descreve-se, nesta seção, a organização do sistema *Digital SisA<sup>3</sup>* a partir de suas componentes de hardware e de software.

A infra-estrutura física de suporte ao *Digital SisA<sup>3</sup>* está organizada por três sistemas de hardware: (i) *Sistema de Medição*, contendo uma trena eletrônica a *laser*; (ii) *Sistema de Processamento*, que pode ser representado por diferentes dispositivos computacionais (PC, PDA ou FPGA) providas de interface para conexões (USB, Bluetooth); (iii) *Sistema de Suporte Articulado*, composto de controladores e atuadores, sob forma de bobina solenóide e motores de passo responsáveis pelo deslocamento angular preciso e incluindo bloqueio de posição.

Os recursos de hardware são mapeados para quatro abstrações básicas, modelando a implementação dos programas utilizadas na composição do ambiente de software para o *Digital SisA<sup>3</sup>*: (i) controle de movimento; (ii) leitura de medição; (iii) determinação do ponto representando a posição de medida; e (iv) formatação de dados para o execução do método numérico.

A partir de um ponto inicial e com a trena na posição vertical, efetua-se uma primeira medição de distância. Após, aplicam-se movimentos angulares sobre a trena no sentido transversal ao armazém até o limite lateral, previamente parametrizado. Os valores de leitura da distância e do correspondente ângulo são armazenados juntamente com parâmetros medidos na etapa de calibração para, posteriormente, se determinar na matriz de representação o posicionamento de cada ponto no plano tridimensional. Assim, cada linha da matriz de representação do relevo está associada a um passo de

medição. O reposicionamento do equipamento em diversos pontos do armazém garante uma representação completa do relevo.

Outros equipamentos, por exemplo, *AR4000* da *Acuity* [9], apresentam funcionamento semelhante mas com custo relativamente elevado. Neste sentido, o sistema *SisA<sup>3</sup>* consiste numa proposta automatizada, confiável e de custo competitivo no mercado, com alcance compatível com a estrutura de grande porte dos armazéns comerciais. Estas vantagens decorrem da integração entre a metodologia científica (interpolação numérica por *spline*) e a tecnologia usada na construção do equipamento de medição (atuadores para movimentação de trena *laser*).

#### 4 Modelagem Matemática do Relevo

A entrada de dados para modelagem matemática do relevo consiste na matriz tridimensional, cujos valores das coordenadas são calculadas a partir das medidas obtidas pelo equipamento digitalizador. Os pontos são armazenados a partir de precisão definida pelo equipamento de digitalização e são referenciados de acordo com o espaço euclidiano (3D), com três planos ( $XY$ ,  $YZ$ ,  $XZ$ ) perpendiculares dois a dois, e três eixos coordenados: (i) no eixo  $\overrightarrow{OX}$ , indica-se o sentido das medições; (ii) no eixo  $\overrightarrow{OZ}$ , a evolução temporal do processo de medição está representada; (iii) e no eixo  $\overrightarrow{OY}$ , representa-se a altura dos pontos coletados.

Neste trabalho, consideram-se classes de funções que não apresentam uniformidade em sua construção gráfica, com curvatura irregular ou compostas por sucessivos segmentos de curva. Neste contexto, o armazenamento de todas as coordenadas dos pontos reais que compõem grande parte de uma superfície não-uniforme é computacionalmente inviável. Visando a preservação de propriedades matemáticas e geométricas que determinam a fidelidade da superfície modelada, consideram-se as funções de aproximação obtidas pela técnica de interpolação polinomial cúbica por *splines* [4], geradas a partir de pontos obtidos no processo de digitalização. Na técnica de modelagem por interpolação, a curva passa, obrigatoriamente, pelos pontos do domínio.

O termo *spline* refere-se à longa e flexível barra de metal utilizada pelos desenhistas e engenheiros para delinear as superfícies de aviões e navios. A equivalente expressão matemática a esta barra, *spline cúbica natural*, consiste na continuidade da polinomial cúbica a qual considera a posição, a inclinação e a curvatura como parâmetros que interpolam os pontos de controle. As *splines* são formas suaves de modelagem de curvas paramétricas cúbicas [10] [5] [7].

Seja  $Q(u) = (X(u), Y(u))$  uma curva bi-dimensional e parametrizada, tal que  $X(u)$  e  $Y(u)$  são funções definidas pelo parâmetro  $u$ .  $X(u)$  e  $Y(u)$  são conhecidas através de pontos na curva  $Q(u)$ , para qualquer valor de  $u$ , respectivamente.

$X(u)$ ,  $Y(u)$  são funções polinomiais, logo são consideradas computacionalmente eficientes e de fácil manipulação. Porém, nem sempre é possível modelar a curva  $Q(u)$  aplicando somente uma função. Para modelar esta possível irregularidade, considera-se o particionamento da curva em segmentos, onde  $u$  é o parâmetro que varia do ponto inicial  $u_{min}$  ao ponto final  $u_{max}$ . Cada parâmetro  $u$ , determinado pelas correspondentes funções polinomiais  $X(u)$  e  $Y(u)$ , é denominado de nó e corresponde à ligação entre um segmento e o seu sucessor, com exceção dos nós extremos,  $u_{min}$  e  $u_{max}$ . Salienta-se também que, a sequência de nós não pode ser decrescente, ou seja:

$$u_0 \leq \dots \leq u_j = u_{min} \leq \dots \leq u_l = u_{max} \leq \dots \leq u_n.$$

Assim, a sequência de nós  $\{u_k\}_{0 \leq k \leq n} = [u_0, \dots, u_j, \dots, u_l, \dots, u_n]$  é denominada de vetor de nós. A componente  $X(u)$  está segmentada por um conjunto de funções polinomiais, cobrindo todo o intervalo de definição do parâmetro  $u$ , considerando os segmentos variando de  $u_{min}$  a  $u_{max}$ . O mesmo ocorre no componente  $Y(u)$ . Além disto, cada componente satisfaz condições de continuidade nas junções entre sucessivos segmentos polinomiais, garantindo também a existência e a unicidade para a função diferencial de várias ordens.

Dado  $m + 1$  pontos,  $P_0, \dots, P_m$ , tem-se a definição da curva por  $m$  polinômios cúbicos, onde cada par de pontos  $(P_i, P_{i+1})_{0 \leq i \leq m-1}$  determina o  $i$ -ésimo segmento e está conectado por um polinômio  $X_i$ . Em geral, as coordenadas  $X(u)$  dos pontos da curva são determinados somente pelas coordenadas  $x_0, \dots, x_m$ , ocorrendo o mesmo com os outros eixos, ou seja, o cálculo de cada polinômio é obtido de forma independente em cada coordenada.

Cada segmento  $x_i, \dots, x_{i+1}$  possui uma função polinomial cúbica no parâmetro  $u$  dada pela expressão  $X_i(u) = a_i + b_i u + c_i u^2 + d_i u^3$ , e satisfazendo as seguintes condições:

$$X_i(0) = x_i = a_i \quad \text{e} \quad X_i(1) = a_i + b_i + c_i + d_i. \quad (1)$$

Considerando os quatro coeficientes  $a_i, b_i, c_i, d_i$ , os quais determinam completamente a função  $X_i(u)$ , e a continuidade das funções nos pontos dos segmentos inicial e final, aplica-se a primeira derivada  $D_i$  de  $X(u)$  em cada nó  $u_i$  e obtém-se:

$$X_i^{(1)}(0) = D_i = b_i \quad X_i^{(1)}(1) = D_{i+1} = b_i + 2c_i + 3d_i \quad (2)$$

Pelas equações em (1) e (2), as coordenadas de  $X(u)$  correspondem a solução do Sistema (3):

$$\begin{cases} a_i = x_i, & c_i = 3(x_{i+1} - x_i) - 2D_i - D_{i+1} \\ b_i = D_i, & d_i = 2(x_i - x_{i+1}) + D_i + D_{i+1} \end{cases} \quad (3)$$

De forma análoga são obtidos os coeficientes referentes a  $Y(u)$ . Portanto, ao final, tem-se a identificação de todos os pontos intermediários do intervalo de interpolação.

Pela interpolação por *splines* é gerada uma quantidade de pontos interpolados inúmeras vezes superior à quantidade de pontos coletada pelo instrumento digitalizador, pelo qual se define a matriz de entrada.

## 5 Solução Numérica para Cálculo do Volume do Relevo

A metodologia proposta, além de considerar todos os detalhes do relevo, apresenta a possibilidade de emprego de interpolação para melhoria de precisão e desempenho sem a necessidade de nova medição nos dados de entrada. Entretanto, tem-se neste caso, sua influência direta no tempo de execução do algoritmo.

Nos estudos para modelagem deste trabalho, verificou-se que grande parte da descarga de produtos nos armazéns do tipo horizontais está direcionada ao centro do armazém, a partir de uma esteira localizada na parte superior. Nesta modelagem, considera-se a variação somente no sentido do comprimento do armazém. Assim, os estoques dos produtos tendem a formar cones, figura geométrica uniforme e com fórmula definida para o cálculo de seu volume, a partir de medidas precisas como a altura e o raio da base. Entretanto, na formação dos cones geralmente ocorrem sobreposições em algumas posições ou intervalos no eixo  $\overrightarrow{OZ}$ , e intersecções entre sucessivos cones introduzem dificuldades quando do cálculo analítico do volume. Pela simetria na forma do relevo gerada na distribuição longitudinal (eixo  $\overrightarrow{OZ}$ ) dos grãos no armazém horizontal, e visando a otimização da solução numérica, na geração dos dados da matriz

de entrada foram considerados apenas os semicones e correspondentes superposições e/ou deformações, quando da estratégia de modelagem e desenvolvimento do protótipo i-Cone.

### 5.1 Prototipação Sequencial: i-Cone

Na abordagem sequencial, considera-se o método de integração do volume da face das áreas de camadas. Na modelagem do algoritmo para o protótipo i-Cone, caracterizam-se quatro etapas para o cálculo do volume total, as quais são descritas a seguir.

Na etapa de **Processamento da Entrada**, ocorre a leitura da matriz de entrada e os dados são convertidos para o formato de uma matriz tridimensional de pontos. Depois da construção da matriz tridimensional, com as coordenadas dos pontos, calcula-se os polinômios de interpolação  $X_i(u)$ , cujos coeficientes são armazenados em nova estrutura matricial. Os polinômios de interpolação  $X_i(u)$ , referentes a cada camada, correspondem às linhas desta estrutura matricial.

Na etapa de **Segmentação da Matriz de Dados** são geradas camadas com dois intervalos de medição. Como as medições dos pontos em cada iteração apresentam os mesmos valores para  $Z(i)$ , com  $i$  indicando o índice do passo de medição, cada camada corresponde ao intervalo entre sucessivas medições,  $Z(i)$  e  $Z(i+1)$ , definido os parâmetros para cálculo dos coeficientes dos polinômios de interpolação linear. Após etapa de segmentação da matriz tridimensional, cada camada pode ser modelada de forma independente, pois deixam de existir as dependências, viabilizando a exploração do paralelismo.

Na **Divisão das Camadas** são tratados os erros de modelagem que resultam na formação de degraus durante a segmentação da matriz e geração das camadas, conforme a Figura 2. Para minimizar este erro, a quantidade de camadas é incrementada por técnicas de interpolação: (i) polinomial linear, aplicada no eixo  $\overrightarrow{OY}$ ; e (ii) polinomial cúbica, por *splines*, aplicada sobre o eixo  $\overrightarrow{OX}$ , introduzindo desta forma camadas intermediárias, e, por consequência, a geração de nova matriz. Neste caso, tem-se um polinômio interpolador para cada sequência de nós  $\{u_k\}_{0 \leq k \leq n}$ :

$$X_i(u_k) = a_i + b_i u_k + c_i u_k^2 + d_i u_k^3, \quad 0 \leq i \leq m. \quad (4)$$

A quantidade de subdivisões é definida por um parâmetro que informa o comprimento máximo das novas subcamadas. O algoritmo identifica qual o divisor que deverá ser aplicado a cada camada. A interpolação linear no eixo  $\overrightarrow{OZ}$  define a profundidade associada a cada subcamada gerada.

A não dependências de dados entre as camadas viabiliza a implementação sequencial ou concorrente, considerando um ou vários nodos processadores na sua execução. O usuário seleciona a estratégia de implementação e, dependendo da quantidade de pontos coletados e da precisão desejada, a execução sequencial pode tornar-se inviável, justificando a versão concorrente.

**Cálculo do Volume das Subcamadas:** O cálculo da área é efetuado a partir da face (poligonal) de cada subcamada, considerando-se os dados dos pontos sob a visão bidimensional (Eq. (4)), ignorando as coordenadas do eixo  $\overrightarrow{OZ}$ :

$$S_i = \sum_{i=0}^m (x_{i+1} - x_i) * \frac{1}{2} * [X_i(u_0) + 2X_i(u_1) + \dots + 2X_i(u_{n-1}) + X_i(u_n)] \quad (5)$$

A Eq. (5) considera a integração trapezoidal para o cálculo da área, conforme mostra a Figura 3. Para aproximar a área limitada pelas sessões transversais geradas por um plano vertical sobre um cone, modelado por curvas planas denominadas hipérboles, introduziu-se um processo de redivisão dos trapézios, otimizando o algoritmo. A única exceção, não significativa, refere-se ao centro do cone. Com o valor da área da face da camada já conhecida, multiplica-se pela profundidade  $z_i$ , ou ainda, pela distância entre sucessivos pontos no eixo  $\vec{OZ}$ , obtendo-se assim os volumes parciais referentes às camadas. Os valores dos volumes parciais são entradas do programa somador volume.camada\_i, indicado pela expressão:  $V_i = S_i * z_i$ .

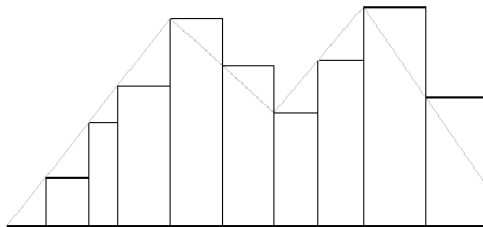


Figura 2. Aproximação da altura

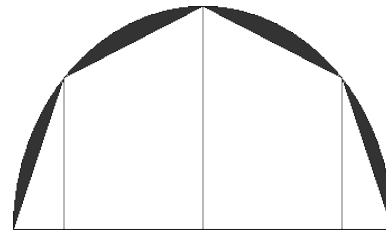


Figura 3. Corte na matriz de pontos.

**Geração do Resultado Final:** Efetua-se o somatório dos volumes das subcamadas, retornando o resultado final:  $V = \sum_{l=0}^{mn} V_l$ .

Salienta-se que o protótipo foi desenvolvido utilizando a linguagem Java, a qual conta com as bibliotecas de suporte para a modelagem prevista. A modelagem do algoritmo sequencial para o protótipo i-Cone assegura a não dependências de dados entre as camadas, e desta forma viabiliza a implementação em paralelo, considerando um ou vários nodos processadores na sua execução. A busca por melhor desempenho justifica uma execução paralela.

## 5.2 Prototipação Paralela: p-iCone

A prototipação paralelizada do software iCone, denominada p-iCone foi desenvolvida sobre a arquitetura VirD-GM (*Virtual Distributed Geometric Machine Model*), a qual implementa abstrações da versão distribuída do modelo GM.

A D-GM foi concebida para dar semântica aos algoritmos da computação científica e considera a indexação dos processos e da memória global e distribuída, possivelmente infinita, para interpretação do paralelismo síncrono. Baseada na construção indutiva, tem-se a operação de composição de construtores sobre processos elementares para geração de novos processos.

O componente de software VirD-GM é responsável por criar e gerenciar um ambiente físico para processamento de aplicações, promovendo a execução, sob este ambiente, das especificações desenvolvidas no ambiente de programação visual VPE-GM. Estas aplicações são, por natureza, paralelizáveis e direcionadas ao estudo de algoritmos paralelos para a computação científica [3].

No desenvolvimento do ambiente de execução do VirD-GM, utiliza-se o *middleware* EXEHDA (*Execution Environment for High Distributed Applications*) [11] para dar suporte aos mecanismos de distribuição, comunicação e gerência das computações

distribuídas. Neste contexto, definem-se processos elementares por funções que, ao serem executadas, alteram apenas uma posição da memória e caracterizam uma unidade de tempo computacional [2]. Para transformação de várias posições de memória, considera-se a composição de processos elementares, executados sequencial ou paralelamente.

Cada aplicação para ser executada na VirD-GM necessita de dois argumentos, o arquivo de entrada de dados e o arquivo descritor de processos.

A memória de entrada é uma matriz tridimensional com os pontos que inicializam o algoritmo. Na Figura 4, apresenta-se no ambiente visual do VPE-GM a edição do processo referente à aplicação  $SisA^3$ , onde tem-se: (a) a aquisição da matriz de entrada, que está indicada pelo processo elementar ( $invert^1$ ); (b) o cálculo dos polinômios de interpolação na etapa de segmentação, modelado pelo processo ( $segm^2$ ); (c) o paralelismo é explorado por processos síncronos que aplicam a função  $subdiv^k$  em distintas posições de memória ( $k = 3(1)7$ ); (d) o execução do somatório após finalizar a sincronização dos processos; e (e) o cálculo do volume total pelo processo  $sumC^8$ , atribuindo o valor resultante à posição de memória 8.

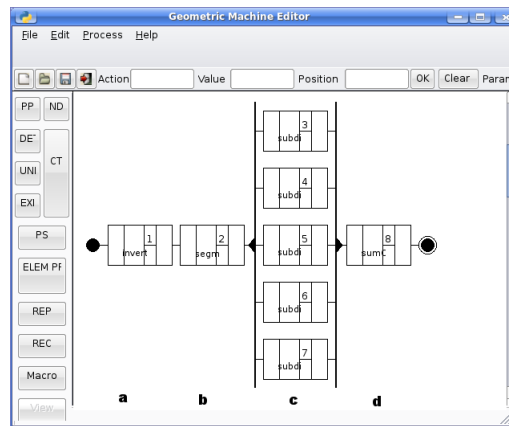


Figura 4. Modelagem do protótipo utilizando o VPE-GM

Salienta-se que o protótipo  $p-iCone$  pode ser executado em processadores multicore, utilizando os vários núcleos do processador. Utiliza-se o VirD-GM como ambiente de execução, o qual define um processo paralelizável para cada núcleo, quando disponível.

## 6 Avaliação dos Resultados

Na validação do algoritmo e da metodologia proposta para o  $iCone$  e o  $p-iCone$ , considera-se um cone com 20 metros de altura e 5 metros de raio. Na versão sequencial, na aplicação da fórmula analítica  $V = (\pi/3 \times r^2 \times h)$ , onde  $r$  e  $h$  correspondem a medida do raio da base e altura do cone, respectivamente, obtém-se o volume real do cone  $V = 261,7993878 \times 2$ .

A Tabela 1, relativa ao semicone, mostra os resultados das iterações do algoritmo e resume a análise de erros.



**Tabela 1.** Resultados da análise de erros do estudo de caso iCone (sequencial)

n	Volume sem dividir area sub-camada	Volume com 10 divisões da area	Profundidade sub-camada em milímetros	Erro Absoluto	Erro Relativo
1	211,4418	212,2856	100,00	49,5138	23,32%
5	251,4418	252,0962	20,00	9,7032	3,85%
10	256,4418	257,0808	10,00	4,7186	1,84%
100	260,9418	261,5770	1,00	0,2224	0,09%

Pela análise sobre os resultados, identificou-se que o número  $n$  de iterações por camada, a partir de  $n = 10$ , gera resultados que, tanto pelo cálculo direto quanto pelo o processo de subdivisão dos trapézios, mostram-se próximos do valor real. No segundo caso, já se tem um pequeno ganho na aproximação do volume real. Com o aumento da quantidade de subdivisões por camada, o resultado converge para o valor real, conforme pode ser visualizado na última linha da tabela, quando se divide uma camada em  $n = 100$  subcamadas (1.0 cm de profundidade). Com base na metodologia atualmente utilizada por parte das empresas de armazenagem em silos, onde as medições e os cálculos são efetuados manualmente, e na qual se considera apenas como parâmetro a altura do centro do armazém, o erro relativo obtido se mostra satisfatório.

A execução paralela no ambiente VirD-GM foi aplicada a, no máximo, 5 unidades de um cluster, considerando um conjunto homogêneo de nodos selecionados. De acordo com a definição da arquitetura da VirD-GM, um dos nodos distingue-se dos demais, caracterizando o nodo VirD-base, responsável tanto pelo escalonamento das tarefas quanto pelo controle do fluxo de dados. A diferenciação na execução dá-se pelo número de nodos, considera-se que os parâmetros de memória e do programa, são idênticos para todos os testes.

Na avaliação da versão paralela do algoritmo para o protótipo p-iCone, considera-se a mesma precisão da versão sequencial, obtendo-se o mesmo resultado numérico mas com melhor desempenho. Na sequência, tem-se a seleção dos testes e a correspondente média dos resultados obtidos.

A Figura 5, com 100 subdivisões em cada área calculada, tem-se a seguinte classificação: o tipo **a** com 100 subcamadas (baixa complexidade), o tipo **b** com 1000 subcamadas (media complexidade), e tipo **c** com 10000 subcamadas (Alta complexidade). Resumem-se os resultados em termos de *speedup* da aplicação referente ao cálculo do volume do relevo, mostrando que o teste com maior complexidade (carga computacional) se refere ao tipo **c** com variação na quantidade de nodos de 1 a 5 e, assim, obtendo o máximo *speedup* de 3,697.

## 7 Conclusão

Neste trabalho, obteve-se uma visão integrada em diferentes áreas que possibilitaram desenvolvimento do sistema /Sis.A<sup>3</sup>, brevemente descritos: (i) avaliação da oportunidade de interação com empresas agrícolas da região, visando a melhoria no gerenciamento e armazenamento de granéis; (ii) contribuição para o desenvolvimento de aplicações distribuídas que utilizam as ferramentas implementadas na continuidade do projeto D-GM; (iii) aproximação da área teórica de fundamentos e matemática da

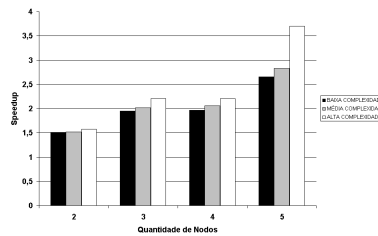


Figura 5. Gráfico de desempenho p-iCone

computação da área de soluções tecnológicas promovendo uma metodologia para o problema de modelagem computacional e o cálculo de volumes em relevos de superfícies não uniformes; *(iii)* desenvolvimento de protótipo sequencial e paralelo contribuindo em áreas de gestão de recursos naturais, aplicados em atividades como monitoramento, controle e auditoria em estoques da produção agrícola; *(iv)* modelagem do *Digital Sis.A<sup>3</sup>*, equipamento *scanner* para coleta e digitalização de dados em tempo hábil.

Na continuidade, tem-se a busca por soluções tecnológicas considerando a área de sensoriamento, pela qual serão adquiridos os parâmetros de entrada, incluindo envio e acesso de resultados aos diversos setores empresariais e a geração dinâmica de processos paralelos no protótipo p-iCone.

## Referências

1. Prestes, D., Reiser, R., Costa, A., Cardoso, C.: Estendendo o modelo de máquina geométrica a um ambiente de programação visual. In: CLEI2005 XXXI Conferência Latinoamericana de Informática, Cali, Universidad Javeriana (2005) 1–10
2. Munhoz, F.N.: Expandindo o vird-gm para suporte a novos construtores do modelo d-gm. Monografia de graduação 1, Pelotas-RS (2008)
3. FONSECA, V.S.: Vird-gm: Uma contribuição para o modelo de distribuição e paralelismo do projeto d-gm. Dissertação para a obtenção do grau de mestre em ciência da computação, Pelotas-RS (2008)
4. Foley, J.D.: Computer Graphics: Principles and Practice. Addison Wesley, NY (1990)
5. Kreyszig, E.: Advanced Engineering Mathematics. John Wiley and Sons, NY (1993)
6. Kirehner, F.F., Filho, A.F., Scolforo, J.R.S., Machado, S., Mitishita, E.A.: O uso de funções splines no cálculo de volume de árvores. Technical report, Revista Floresta, Curitiba/PR (1989)
7. Silva, L.K.M.: Um sistema de modelagem geométrica usando splines. Technical report, Universidade Católica de Pernambuco, Recife/PE (2007)
8. Al-Alam, W.G., Reiser, R.S., Pilla, M.L.: Estudo de metodologias para cálculo numérico do volume de produtos em armazéns. In: VIII Encontro Regional de Matemática Aplicada e Computacional, Pelotas / RS (2008)
9. Distage: General - Laser Distance Meter. Distage, <http://www.distage.com/gequ.html>. (2003)
10. Bartels, R.H., Beatty, J.C., Barsky, B.A.: Introduction to Splines in Computer Graphics and Geometric Modeling. Morgan-Kaufmann (1987)
11. Yamin, A.: Arquitetura para um Ambiente de Grade Computacional Direcionado às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva. PhD thesis, Instituto de Informática/UFRGS - Porto Alegre (2004)