Jornada de Iniciação Científica e Tecnológica do LNCC

Petrópolis, 25 de setembro de 2015.

Laboratório Nacional de Computação Científica – LNCC

Diretor
Pedro leite da Silva Dias

Coordenação de Administração - CAD
Anmily Paula dos Santos Martins

Coordenação de Ciência da Computação - CCC
Jauvane Cavalcante de Oliveira

Coordenação de Matemática Aplicada - CMA
Frédéric Gerard Christian Valentin

Coordenação de Mecânica Computacional - CMC
Márcio Arab Murad

Coordenação de Sistemas e Controle - CSC
Carlos Emanuel de Souza

Coordenação de Sistemas e Redes – CSR
Wagner Vieira Léo

Programa Institucional de Bolsas de Iniciação Científica &
Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação
Renato Portugal

Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq

Presidente
Glaucius Oliva

Coordenadora Geral do PIBIC
Lucimar Batista de Almeida

Jornada de Iniciação Científica e Tecnológica do LNCC

Comissão Interna do PIBIC/PIBITI-LNCC

Renato Portugal
Eduardo Lucio Mendes Garcia
Helio José Correia Barbosa
Jack Baczynski

Avaliador Externo

Demerson Nunes Gonçalves
CEFET

Apresentação

Apresentamos neste volume a relação dos trabalhos da Jornada de Iniciação Científica e Tecnológica do LNCC, desenvolvidos pelos bolsistas no período de agosto de 2014 a julho de 2015. Estes trabalhos foram apresentados em sessão pôster para, juntamente com as fichas de avaliação do bolsista, serem utilizados como elementos de avaliação do PIBIC/PIBITI-LNCC. Nesta avaliação contamos com o Prof. Dr. Demerson Nunes Gonçalves - CEFET, que em função das entrevistas e exame deste volume preparou e enviou ao CNPq relatório de avaliação do Programa.

Agradecemos o empenho dos Professores Orientadores e dos Bolsistas na preparação das fichas de avaliação e dos pôsteres e da disponibilidade da presença para entrevista, para que nosso programa fosse avaliado. É preciso mencionar que temos sido bem avaliados, estando nosso PIBIC/PIBITI bem colocado entre os melhores do país, conforme divulgação feita pelo CNPq.

Aproveitamos a ocasião para agradecer ao CNPq pelas bolsas concedidas, a Direção do LNCC pelo apoio e a Comissão Interna deste programa no LNCC.

Renato Portugal
Coordenador do PIBIC/PIBITI - LNCC

Índice

Bolsistas PIBIC com 1 ano ou mais

Bolsistas PIBIC com menos de 1 ano

Bolsistas PIBITI com 1 ano ou mais

Gerenciamento do Motor Gerador de *Containers* para Nuvens Computacionais

Allan Matheus M. dos Santos, Bruno Schulze

Diversos estudos têm sido realizados nos últimos anos dedicados a verificar qual o nível dos efeitos que a camada de virtualização traz ao desempenho de ambientes computacionais. Motivado por esses estudos, este trabalho descreve uma abordagem que substitui o uso dos virtualizadores mais comuns, como o KVM, por LXC (*LinuX Containers*). Embora, atualmente, a tecnologia de *containers* não ofereça todas as funcionalidades dos *hypervisors* tradicionais, podemos verificar o aumento do uso dessa tecnologia como alternativa para as infraestruturas de nuvens computacionais devido ao ganho em performance que os *containers* oferecem em relação às máquinas virtuais.

Este trabalho é fruto de uma análise das infraestruturas de nuvem computacional existentes e consiste no desenvolvimento de uma ferramenta de controle para o VirtualIS (*Virtual Infrastructure for Science*) com objetivo de ampliar o controle da infraestrutura computacional de nuvem em apoio às aplicações científicas hospedadas pelo ComCiDis e INCT-MACC. A ferramenta *rufus-core* é um motor gerador de *containers*, responsável pela criação de *containers* e por disponibilizá-los em forma de *templates* pré-configurados dedicados às aplicações científicas.

As seguintes funcionalidades estão disponíveis atualmente na API: exibir informações sobre o *host*, os *containers*, os *templates* e as imagens de *container* disponíveis; criar, clonar, alterar o estado e destruir *containers*; disponibilizar *templates* de aplicações científicas criados através de *containers* pré-configurados; e executar aplicações em paralelo. Para essa última, são criados *N* novos *containers* a partir de clonagem utilizando a técnica de *Copy On Write*.

A utilização da tecnologia de virtualização por *containers* em apoio as nuvens computacionais amplia as possibilidades de utilização deste tipo de arquitetura em especial por aplicações que demandam alta escalabilidade e muito capacidade de processamento. Nesse sentido, a principal contribuição das ferramentas apresentadas é explorar essas possibilidades permitindo que pesquisadores possam se beneficiar das vantagens oferecidas pela tecnologia de *container* no ambiente de nuvem. Dessa forma, facilitando para pesquisadores, o processo de execução de fluxos aplicações em ambientes virtualizados.

Simulações de um modelo epidemiológico aplicado à Dengue

Prates, D.B.[1], SILVA, J. M.[1], Kritz, M. V.[2]

1- Instituto de Ciência, Engenharia e Tecnologia, ICET, UFVJM.
2- Laboratório Nacional de Computação Científica, LNCC.

RESUMO

A modelagem matemática aborda diversas áreas do conhecimento tornando-se uma importante ferramenta para interpretação e descrição de diversos fenômenos naturais. Dentre as diversas áreas destaca-se o estudo epidemiológico. As epidemias são historicamente responsáveis por inúmeras mortes no mundo, o que as tornam um assunto de saúde pública. O estudo das epidemias é realizado através de modelos compartimentais baseados em equações diferenciais, onde há análises e estudos das características endêmicas facilitando a compreensão do comportamento dinâmico de epidemias em determinadas populações.

O modelo SIR proposto por Kermack e McKendrick em 1927, foi o pioneiro na modelagem epidemiologia e propõe dividir a população em três classes: suscetíveis, infectados e removidos. Com a inserção de indivíduos infectados na população, a epidemia pode ser analisada apresentando resultados conclusivos. O modelo é constantemente modificado para simular as características intrínsecas de determinada doenças.

A Dengue é uma doença viral transmitida pelos mosquitos Aëdes aegypti e Aëdes albopictus e apresenta a peculiaridade de ser causada por até quatro tipos virais. A disseminação da dengue está diretamente relacionada a fatores ambientais, como temperatura e precipitação pluviométrica. Esta doença é caracterizada por epidemias recorrentes, principalmente em grandes centros urbanos, e ações de controle têm se mostrado ineficientes frente ao crescimento populacional desorganizado e aos problemas de saneamento básico.

Modificações nos modelos epidemiológicos tradicionais através da inserção de parâmetros e classes que simulam a sazonalidade, a presença de mais de uma variante do vírus e a imunidade cruzada adquirida depois da primeira infecção foram realizadas visando simular condições reais. A partir desse modelo modificado foi realizado estudo sobre a influência dos parâmetros associados à epidemia no comportamento epidemiológico da população.

Referências

[1] Aguiar, M., Stollenwerk, N., Kooi, B.W., (2009). Torus bifurcations, isolas and chaotic attractors in a simple dengue model with ADE and temporary cross immunity.Int. J. Comput. Math. 86, 1867–1877.

[2] Prates, D. B. ; Jardim, C. L. T. F. ; Figueiredo, L. A. ; Silva, J. M. ; Kritz, M. (2015) . Simulations of a epidemic model with parameters variation analysis for the dengue fever. Journal of Physics. Conference Series, 1742-6596.

[3] Prates, D. B. ; Gomes, J. L. ; Kritz, M. ; Silva, J. M. (2015) . An epidemiological model with vaccination strategies. AIP Conference Proceedings, 1551-7616.

PARALELISMO E TOLERÂNCIA A FALHAS NO AMBIENTE SPiNMe

Emiliano Medeiros de Oliveira Neto (Bolsista PIBIC-CNPq), Antônio Tadeu Azevedo Gomes (Orientador)

Resumo

Neste projeto busca-se a compreensão e aperfeiçoamento de funcionalidades do simulador numérico MHM (Multiscale Hybrid Mixed) desenvolvido nas linguagens Erlang e C++. O objetivo é propor e implementar melhorias de desempenho, suporte a tolerância a falhas, maior escalabilidade e flexibilidade, e melhor aproveitamento possível da infraestrutura disponível para a execução desse simulador.

Durante o primeiro ano deste projeto, as atividades concentraram-se nos módulos desenvolvidos em Erlang, que têm como foco o tratamento da distribuição dos processos C++ e o suporte a tolerância a falhas nesses processos. Das contribuições para aumento do desempenho na execução do simulador, destacam-se a análise e implementação realizadas para a redução dos custos computacionais, diminuindo a frequência com que novos processos C++ são criados e terminados em cada nó processador na infraestrutura. Com a substituição do antigo método de criação e terminação dos processos C++ pela troca de mensagens entre esses processos e os processos correspondentes em Erlang, observou-se uma redução significativa no tempo de execução para um cenário com processos C++ de baixo custo computacional, em ambiente de execução local.

Além disso, duas funcionalidades foram incluídas no código do simulador. A primeira delas diz respeito ao suporte no simulador à resolução em paralelo de problemas transientes. A segunda funcionalidade foi a flexibilização do simulador para funcionamento em cloud. Nesta segunda funcionalidade, ainda em desenvolvimento, busca-se implementar módulos Erlang que permitam a execução dos processos C++ tanto em ambiente de cluster como de cloud sem modificação do código C++ correspondente, permitindo a evolução desacoplada e simultânea de ambos os códigos C++ e Erlang.

Palavras-chave: *Multiscale, Hybrid, Mixed, SPiNMe*

Uma abordagem de *re-scoring* através do potencial estatístico DOPE

Laboratório Nacional de Computação Científica (LNCC)

Frederico J. R. Carlos[1], Gregorio K. Rocha[2], Fábio L. Custódio[2], Laurent E. Dardenne[2]
fjose, gregorio, flc, dardenne@lncc.br

Palavras Chave: Predição de Estrutura de Proteínas, Modelagem Molecular, GAPF, DOPE.

A predição de estrutura de proteínas tem por objetivos elucidar o arranjo tridimensional a partir de sua sequência de aminoácidos. Na natureza, as sequências de aminoácidos se enovelam em um único estado nativo em um curto período de tempo. Prever a estrutura 3D de uma proteína implica em alto custo computacional, enquadra-se como um dos maiores desafios da biologia computacional, tendo uma diversidade de aplicações biotecnológicas.

O Grupo de Modelagem Molecular de Sistemas Biológicos (GMMSB/LNCC) vem desenvolvendo um programa para predição de estruturas de proteínas chamado *Genetic Algorithm for Protein Folding* (GAPF). O GAPF utiliza um algoritmo genético de múltiplos mínimos com *crowding* fenotípico para a busca das melhores conformações na superfície de energia, aliado à uma função de energia obtida com um campo de força clássico (GROMOS96). Ao final da predição, os modelos de melhor energia nem sempre correspondem aos modelos de melhor RMSD (considerados os melhores modelos). Assim, esse trabalho busca uma abordagem para selecioná-los dentre toda a população final.

Um conjunto teste com 50 proteínas de diversos tamanhos (de 18 até 95 resíduos) foi utilizado. As predições são realizadas em 30 rodadas independentes, com uma população de 200 indivíduos em cada, resultando em um total de 6000 modelos por proteína, dentre os quais os melhores modelos devem ser escolhidos.

Após a geração dos 6000 modelos finais pelo GAPF, os mesmos são submetidos a um *re-scoring* utilizando o potencial do *Discrete Optimazed Protein Energy (DOPE)*, o qual faz uso de um potencial estatístico baseado em distâncias interatômicas.

Ao final do *re-scoring,* são selecionadas através do potencial do DOPE as 10 estruturas de melhor energia de cada proteína. O modelo de melhor RMSD dentre essas estruturas foi calculado em relação à estrutura nativa de referência (retirada do PDB), e comparado com os melhores modelos obtidos utilizando somente a função de energia do GAPF (i.e., sem *re-scoring*).

O *re-scoring* através do DOPE não promoveu melhoras significativas na seleção dos melhores modelos, incluvise piorando a qualidade dos modelos selecionados em mais de 50% dos alvos em relação ao GAPF sem *re-scoring*. Acredita-se que o DOPE possui uma boa capacidade de discriminação de estruturas somente quando as mesmas encontram-se com as conformações bem definidas, o que nem sempre é encontrado nas populações finais do GAPF.

Interação Tátil e Renderização 2D de Simulador de Cirurgia por Videolaparoscopia

Gabriel Pereira Guarisa, Jauvane Cavalcante de Oliveira

Dado um sistema nominado LapVR, que permite que um usuário tenha toda a sensação tátil relacionada com uma operação cirúrgica de videolaparoscopia. Este equipamento possui um software de treinamento de pequenos procedimentos que tem como objetivo a aquisição de destreza e coordenação visual-tátil. Entretanto, não há um procedimento cirúrgico completo.

Um dos principais motivos que não possibilitam o desenvolvimento de um sistema de cirurgia mais complexo deve-se ao fato de que o aparelho não tem capacidade de processamento e de renderização que seja suficiente para processar e apresentar cenas complexas ao usuário. Por este motivo a ideia apresentada era de separar o processamento de interações táteis, realizadas no LapVR em si, de responsabilidade do aluno redator do presente documento; e a parte da interface gráfica do usuário, que seria realizada por outra máquina, de responsabilidade de outro aluno do laboratório.

Uma API foi cedida pela empresa responsável pela fabricação do LapVR, para poder iniciar o desenvolvimento do sistema. A API era responsável somente pela parte tátil da máquina, fornecendo dados sobre os periféricos, e permitindo uma resposta de força e feedback nas mesmas.

Inicialmente um tempo foi tomado para estudo da API, para o conhecimento de suas funções e delimitação do que era possível realizar com a mesma. Para testar as possibilidades, e desenvolver as aptidões necessárias para o desenvolvimento do projeto, pequenos protótipos foram pensados e desenvolvidos, utilizando a linguagem de programação C++. Inicialmente, um programa teste de captura dos dados foi-me apresentado, feito por outro aluno do laboratório. Utilizando tal programa como base, e após alguns testes e algumas versões, o sistema se mostrou estável.

Para um desenvolvimento mais fluído um sistema foi pensado, criando uma subdivisão no projeto inicial, dado que o sistema tátil e o gráfico já eram duas partes independentes. A primeira subdivisão é responsável pela captura dos dados dos periféricos manipulados pelo usuário; a segunda parte é composta, basicamente, por um socket responsável por enviar os dados coletados por uma rede, restrita entre as duas máquinas, para a parte gráfica; a terceira consistia em uma thread responsável por aguardar uma eventual resposta da parte gráfica, indicando se houve alguma colisão, e, por fim, a quarta parte, caso tenha ocorrido alguma colisão, aplica uma resposta de força e feedback nos instrumentos utilizados.

Logo após estabelecer o sistema e seus módulos o desenvolvimento foi iniciado. Já na parte de coleta dos dados alguns problemas começaram a surgir, por exemplo, o uso excessivo do processador do LapVR, que posteriormente foi tratado com uma otimização do código. Outro problema era um erro constante em acessar o periférico responsável por simular a câmera, problema que já estava presente no programa teste apresentado para estudo, após uma análise mais detalhada do código, e algumas mudanças na estrutura do algoritmo, o problema foi solucionado.

Em busca de uma otimização no envio dos dados, uma estrutura foi criada para padronizar o envio de informações ao computador rodando a parte gráfica. A estrutura é dividida em elementos para representar cada periférico. Tal padronização permitiu ao desenvolvedor da parte

gráfica ter uma noção maior dos dados disponíveis que teriam que ser utilizados. Uma "suavização" dos dados enviados foi feita tirando a média dos últimos valores capturados, pois, ao aplicar na parte gráfica, oscilavam de dois a três décimos (na escala utilizada pelo LapVR), gerando um desconforto visual.

Feito isto, a parte seguinte foi iniciada, uma thread foi utilizada para o socket aguardar uma resposta da parte gráfica, e assim ter independência da parte do código responsável por enviar os dados. Tal parte é responsável por aguardar uma eventual resposta, indicando que houve uma colisão entre os objetos representados e exibidos na tela.

Por fim, a última etapa, no desenvolvimento do sistema, teve início, porém muitas dificuldades foram identificadas, a começar pela falta de funções específicas na API, que serviriam para determinados tipos de resposta (force-feedback). Apesar dos contratempos, um módulo-protótipo responsável pela resposta tátil foi desenvolvido, porém não houve tempo hábil para testes do mesmo.

Rufus: portal web de gerenciamento de recursos computacionais e submissão de tarefas científicas

Jonatan Gall Delgado de Souza – ComCiDis - LNCC, Bruno Schulze

O uso de aplicações científicas para a solução de problemas complexos requer uma demasiada quantidade de recursos computacionais. Com base neste fato, o processamento dessas aplicações em um ambiente local, mesmo com tarefas em paralelo torna o tempo alvo bastante inviável. Alternativas como *clusters* computacionais e virtualização têm se tornado cada vez mais fundamentais para a execução de tarefas científicas que demandam uma elevada quantidade de recursos computacionais. Porém, a configuração de um ambiente distribuído pode se tornar uma tarefa onerosa e de alta complexidade.

Diante deste cenário, o trabalho tem por finalidade o desenvolvimento de portal *web* capaz de gerenciar recursos computacionais e tornar possível a submissão de tarefas científicas. Denominado *Rufus-web,* o portal em questão será capaz de fornecer um *cluster* computacional gerenciável a partir de uma integração com uma API geradora de *templates*, e ainda prover um ambiente onde o pesquisador pode submeter suas tarefas com uma interface intuitiva, facilitando assim a elaboração e submissão de uma aplicação específica ou *workflow* científico.

Ao acessar o portal Rufus-web, o sistema verifica se este está autenticado no Argus (Autenticador do portal VirtualIS). Uma vez autenticado, o portal inicia uma seção com o usuário, caso contrário, este usuário e redirecionado ao autenticador. Ao primeiro acesso ao portal Rufus-web, é criado um diretório com o e-mail do usuário obtido pelo autenticador. Neste diretório, o usuário pode fazer upload de seus arquivos. Os arquivos serão exportados através do serviço NFS para a API geradora de templates responsável pelos clusters. No ambiente de gerência de templates os administradores poderão configurar templates base com as diretivas necessárias para a execução de sua aplicação.

Uma vez que os templates base estejam configurados e com os arquivos a serem processados pela aplicação no diretório do usuário, pode-se utilizar a área de submissão de tarefas e workflows científicos Este ambiente possui uma interface gráfica amigável, onde é possível elaborar workflows científicos através de diagramas.

Ao ser submetido, o workflow é serializado em JSON e é submetido para a API geradora de templates executando a tarefa de acordo com as configurações feitas. Ao término da tarefa, o portal salva os resultados no diretório do usuário. Estes resultados podem ser obtidos através de download.

A utilização do portal tornará possível ao pesquisador um foco direcionado à pesquisa propriamente dita e não mais nos recursos computacionais, tendo em vista a necessidade apenas da criação de um template base e configuração deste para formação do cluster.

Controle de Deformação e Resistencia de vigas homogêneas

Monique Ribeiro da Costa, Jaime E. Muñoz Rivera

Modelaremos as deformações de uma viga, através de uma equação de segunda ordem, de coeficientes constantes. Estes coeficientes dependem das características do material. Nosso objetivo é encontrar a máxima carga que pode ser colocada sobre a viga, de tal forma que a deformação dela esteja dentro dos padrões de segurança. O método que usaremos será o método de equações diferenciais parciais e a teoria de controle ótimo para funcionais quadráticos.

Faremos a discretização por diferenças finitas do problema.

Controle e Simulação de Sistemas Sujeitos a Saltos Markovianos

Oscar Neiva Eulálio Neto, Marcos Garcia Todorov

Introdução

Este trabalho é dedicado ao estudo de sistemas sujeitos a saltos Markovianos e questões relacionadas a cadeias de Markov. A pesquisa desenvolvida visa o aprendizado de aspectos teóricos fundamentais relacionados a cadeias de Markov, teoria de controle, e simulação de problemas distribuídos.É de particular interesse o problema de cálculo do PageRank através de resultados recentes da literatura de sistemas sujeitos a saltos markovianos, em especial aqueles reportados no artigo[1]. Alguns artigos recentes[2] ilustram o interesse crescente da comunidade de sistemas e controle no tema. Diversas dificuldades são encontradas no cálculo do PageRank, dada a complexidade deste problema: os problemas são de grande dimensão, e a variabilidade da web torna necessária a atualização frequente do cálculo. Ademais, a possibilidade do cálculo ser efetuado através de recursos computacionais distribuídos evidencia o caráter desafiador deste problema.

Sistemas Lineares com Saltos Markovianos

Considere $\Theta = \{\Theta(k), k = 0,1,...\}$ uma cadeia de Markov homogênea no espaço de estados discreto $N = \{1, 2, \dots, M\}$, isto é, tal que:

$$P(\Theta(k+1) = j \mid \Theta(k) = i, \Theta(k-1), ..., \Theta(0)) = P(\Theta(k+1) = j \mid \Theta(k) = i),$$

que é denominada propriedade de Markov. Tal processo estocástico tem aplicação em diversas áreas da ciência como, por exemplo, computação, engenharia, economia e biologia [3].

O interesse neste trabalho é no estudo de sistemas dinâmicos governados pela seguinte equação:

$$x(k+1) = A_{\theta(k)}\, x(k), \quad k = 0, 1, 2,$$

que são frequentemente chamados Sistemas Lineares com Saltos Markovianos. Existe atualmente uma vasta literatura dedicada ao estudo de tais sistemas, devido à crescente complexidade das aplicações modernas, impulsionadas em grande parte pelos avanços em telecomunicações e em computação distribuída.Uma referência recente no tema é o livro [4].

PageRank

O PageRank foi criado em 1998 para atribuir valores para páginas de um conjunto de documentos interligados. A idéia por trás do algoritmo é de atribuir pesos às páginas da world wide web, em que recebem os maiores pesos as mais visitadas.

Definição

A análise do PageRank tipicamente modela a web como um grafo orientado contendo N nós, que representam as páginas, e uma coleção de arestas E que representam os links correspondentes. A navegação é então representada através de uma cadeia de Markov com espaço de estados discreto de dimensão N. A hipótese usual é de que todas as arestas que saem de um dado nó têm o mesmo peso, e portanto a matriz de transição da cadeia de Markov é a seguinte:

$$p_{ij} = \begin{cases} \frac{1}{N_i}, \text{caso } (i,j) \in E, \\ 0, \quad \text{caso contrário,} \end{cases}$$

onde N_i é o número de links que saem da página i.

O PageRank de um conjunto de N páginas da web é definido como o vetor $x \in R^{N\times 1}$ que satisfaz as seguintes equações (1):

$$x = Ax \ , \ x \geq 0, \quad \sum_{j=1}^{N} x^j = 1, \quad (1)$$

onde $A \in R^{N\times N}$ é a transposta da matriz de transição da cadeia de Markov.

Um método usual para obtenção do PageRank é o chamado *power method* (2), que busca aproximar o PageRank através da seguinte iteração:

$$x(k+1) = Ax(k), k \geq 0, \quad \text{com } x(0) = x_0 \quad (2)$$

onde $x_0 \in R^{N\times 1}$ é uma condição inicial positiva de soma igual a um.

Modelo Teleportation

Muito embora a simplicidade do power method o torne atraente, em geral ele não fornece uma garantia de convergência. O *teleportation model* é uma estratégia reconhecida para que,

através de uma pequena modificação na matriz A, o método convirja globalmente para o PageRank.

O modelo considera os casos em que um navegador que passeia aleatoriamente por páginas, após um certo tempo, fica entediado, e deixa de seguir a estrutura de Hiperlink anteriormente descrita. E então ele salta para uma página não diretamente conectada a que ele se encontra.

Matematicamente o teleportation model é representado como uma combinação convexa de duas matrizes. Em que,

- m é um parâmetro, tal que m ϵ (0,1);
- e a matriz link modificada é dada por M ϵ $R^{n \times n}$, definida por,

$$M := (1 - m)\, A + (m/n)\, 11^T,$$

onde $11^T \in R^{N \times N}$ é a matriz cujas entradas são todas iguais a um.

Algoritmos Distribuídos

A fim de tornar o cálculo do PageRank menos custoso, e melhor explorar os recursos computacionais dos servidores disponíveis na web, uma alternativa é o emprego de algoritmos distribuídos. Neste caso um conjunto específico de servidores web calcula, localmente, uma estimativa do PageRank, e através de comunicações eventuais os cálculos locais são trocados para fins de atualização.

Uma classe de algoritmos distribuídos recentemente proposta na literatura de controle [5] assume que, a cada instante de tempo, os servidores da web realizam duas operações: (i) envia a estimativa de seu PageRank para as páginas para quais tem links de saída, e (ii) requisita estimativas para os servidores a ele conectados. Para fins de paralelismo, supõe-se que tal comunicação é feita de forma assíncrona e que seu início é determinado de forma aleatória.

A agregação de tais experimentos aleatórios faz então com que o power method se torne um sistema linear com saltos Markovianos do seguinte tipo:

$$x(k+1) = A(\theta_k)x(k), k \geq 0, \quad \text{com} \quad x(0) = x_0.$$

Para as matrizes links distribuídas A_i Є R, i = 1, 2, … , n, tem-se que elas são definidas da seguinte forma:

- A i-ésima coluna de A_i coincide com a i-ésima coluna de A.
- A j-ésima diagonal entrada de A_i é igual a um, para j = 1,...,n, j ≠ i.
- Todas as outras entradas a_{ij} são zero.

De forma similar ao que ocorre no power method, este método pode apresentar problemas de convergência, que são resolvidos pela seguinte modificação do teleportation model:

$$x(k+1) = (1-\hat{m})A(\theta_k)x(k) + \frac{\hat{m}}{N}11^T, k \geq 0, \quad \text{com} \quad x(0) = x_0$$

onde $\hat{m}$ é definido em (3) através da seguinte fórmula:

$$\hat{m} = (2m)/(n - m(n - 2)). \quad (3)$$

Esse algoritmo converge, conforme mostrado em (H. Ishii and R. Tempo, 2014), no sentido da média quadrática:

$$\lim_{k\to\infty} E[\|y(k) - x \quad \|^2] = 0,$$

onde y(k) é a média do conjunto de amostras x(0), ... , x(k) definida como:

$$y(k) = \frac{1}{k+1}\sum_{i=0}^{k} x(i)$$

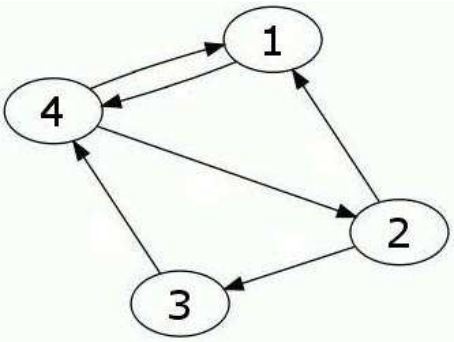
Essa média é conhecida como média de Polyak ou média de Cèsaro e sua forma recursiva é dada por:

$$y(k+1) = [y(k).(k+1)/(k+2)] + [1/(k+2).x(k+1)].$$

Simulações

Durante o período de iniciação científica foram realizadas simulações computacionais de algoritmos implementados em linguagem Matlab. A seguir são apresentados os resultados das simulações feitas para cada um dos modelos apresentados anteriormente. As simulações foram feitas considerando-se um conjunto de quatro páginas, ou seja, uma cadeia de Markov com quatro

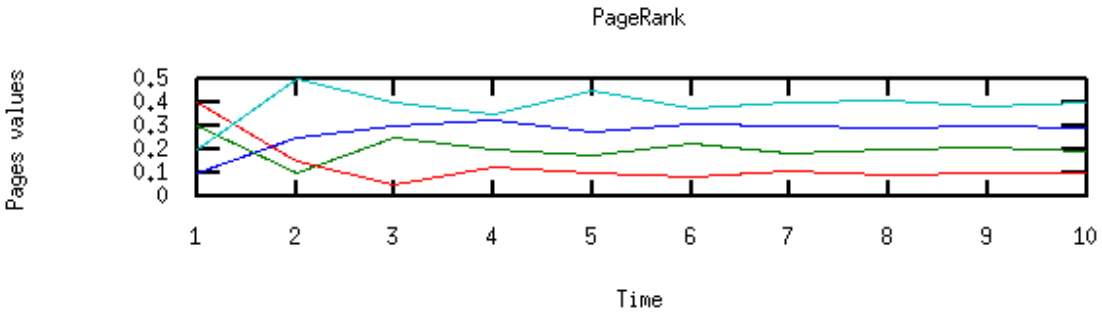
estados. Foram feitas simulações dos três modelos descritos anteriormente, e uma simulação usando-se algoritmo de Monte Carlo.



Simulação do PageRank

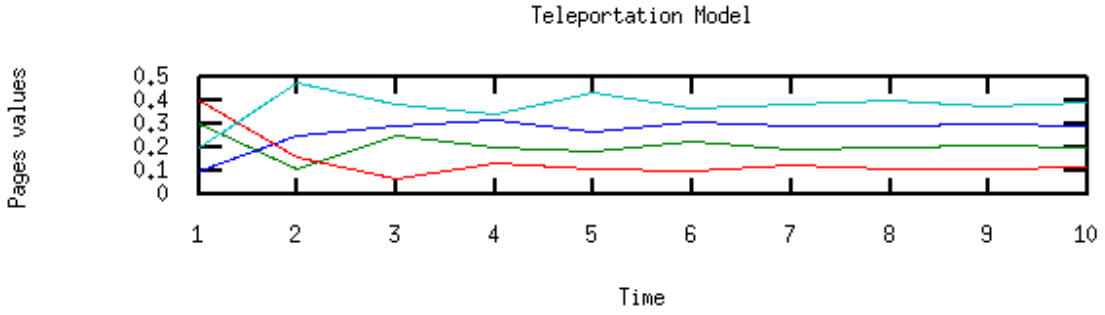
Na simulação do modelo do PageRank em sua forma mais simples, observa-se uma rápida convergência para cada um dos valores do vetor de estados, isso ocorre devido a dimenção do problema usado na simulação.

Nesta simulação tem-se para os valores das quatro páginas: a cor azul escuro para a página 1, a cor verde para a página 2, a cor vermelha para a página 3 e a cor azul claro representa os valores de estado da página 4. Assim pode-se observar que a página de número quatro foi a que recebeu um maior peso neste conjunto de links. Ao mesmo tempo que é a página que está associada a um maior número de links, tanto de entrada quanto de saída.



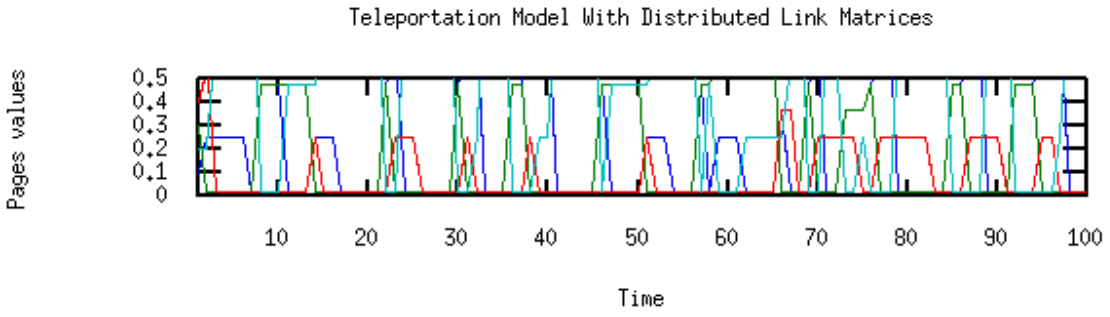
Simulação do Teleportation Model

Na simulação do Teleportation Model, observa-se uma grande semelhança com os resultados da simulação anterior, isso ocorre pois o exemplo simulado é bem comportado, em situações um pouco mais mal-condicionadas, com estados absorventes ou classes fechadas, haveria uma melhor distinção entre os dois modelos. Além disso, o modelo de Teleport considera saltos do navegador para uma página desconexa a que ele se encontra,devido a quase todas as páginas estarem conexas, nos resultados deste simples modelo não é possível ser feita tal observação.

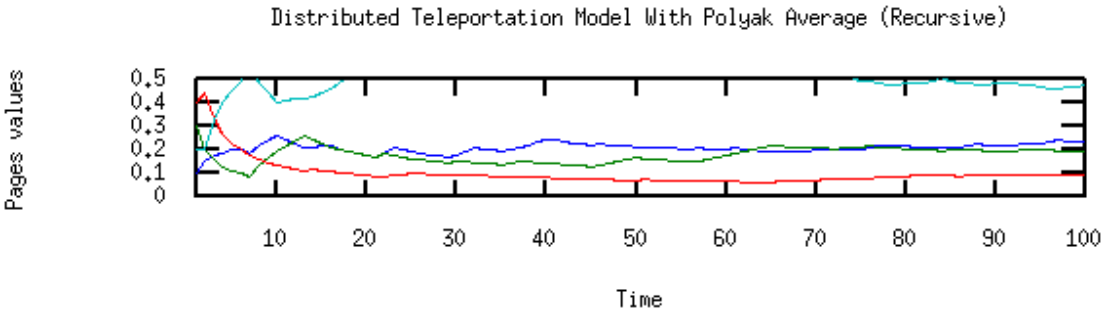


Simulação do Modelo Distribuído

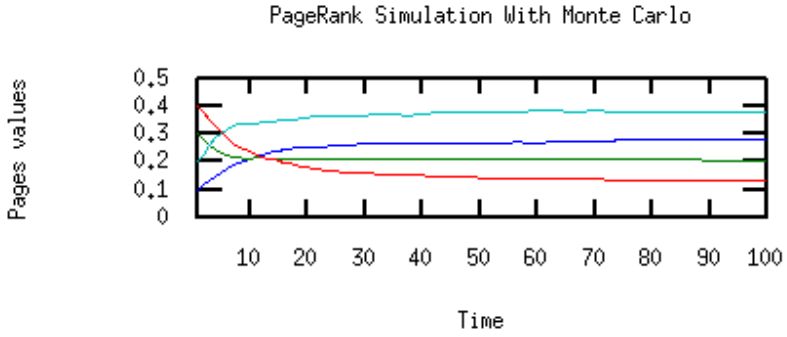
Em uma primeira simulação do modelo distribuído observa-se uma oscilação dos estados sem que ao longo do tempo estacionem em algun valor, este problema é tratado por meio de uma média dos valores dos estados da cadeia de Markov, conhecida como média de Polyak ou média de Cèsaro.



Assim, feita a média dos valores da cadeia no modelo distribuído, chega-se a valores finais próximos aos valores anteriormente encontrados para cada um dos estados.



Ao final fez-se uma simulação usando método de Monte Carlo, de forma que é feita uma média dos estados obtidos na simulação do modelo distribuído do Teleportation Model após os estados passarem também pela média de Polyak. Por fim obtém-se resultados com uma transição de estados mais suavizada e que convergem de forma mais precisa para os resultados obtidos nas simulações do modelo simples do PageRank.



Perspectivas Futuras

No que concerne a aplicações, este trabalho tem especial atenção ao estudo do algoritmo PageRank através de resultados da teoria de sistemas lineares com saltos Markovianos. Neste projeto pretende-se ainda trabalhar com outros métodos de controle e simulação de sistemas estocásticos, que também sejam válidos na simulação do PageRank. Simulações essas como as de

modelos agregados de cadeias de Makov e dentro do contexto do PageRank abordagens também de problemas de consenso.

Bibliografia

[1] H. Ishii, R. Tempo, EW. Bai, PageRank computation via a distributed randomized approach with lossy communication, Elsevier, 2012.

[2] J. Lei, HF. Chen, Distributed Randomized PageRank Algorithm Based on Stochastic Approximation, Fellow, IEEE ,2015.

[3] P. Brémaud. Markov Chains, Gibbs Fields, Monte Carlo Simulation, and Queues, volume 31 of Texts in Applied Mathematics. Springer, New York, 1999.

[4] O. L. V. Costa, M. D. Fragoso, and M. G. Todorov, Continuous-Time Markov Jump Linear Systems. Probability and Its Applications. Springer-Verlag, Heidelberg, 2013.

[5] H. Ishii and R. Tempo. The PageRank problem, multiagent consensus, and web aggregation: a systems and control viewpoint. IEEE Control Syst. Mag., 34(3):34–53, 2014.

Avaliação de desempenho do paralelismo de um problema fortemente acoplado em arquiteturas Multicore e Manycore

Rafael Lourenço Stanislau, Carla Osthoff, Mariano Pereira Silva

Resumo:

O Jogo da Vida é um autômato celular, desenvolvido por John Conway, com o objetivo de observar o comportamento em grupos de seres vivos. Foi-se escolhido esta aplicação para avaliação, por ser de fácil implementação e paralelização. Foram utilizados três processadores e um coprocessador: Intel Core i7 Nehalem com 6 cores, 12 GB RAM, 3.33 GHz e 12 MB de cache; Intel Xeon Westemere (1ª geração) com 12 cores, 24 GB RAM, 2.4 GHz e 12 MB de cache; Intel Xeon Sandy Bridge (2ª geração) com 16 cores, 184 GB RAM, 3.1 GHz e 20 MB de cache e Intel Xeon Phi com 61 cores, 16 GB RAM, 1.238 GHz e 30.5 MB de cache. O Jogo da Vida, implementado em Fortran, foi paralelizado com OpenMP e rodou com uma configuração de 2000x2000 células a 5000 gerações (time steps). Efetuaram-se testes de um thread até a quantidade de cores físico de cada processador e 244 threads no Xeon Phi. Todos os testes foram avaliados com o Hyper-Threading desabilitado e utilizaram-se, também, as instruções FMA e AVX no Xeon Sandy Bridge. Avaliaram-se as métricas de tempo de execução, Speed Up e Eficiência. A versão paralela do Jogo da Vida obteve o menor tempo de execução no Xeon Sandy Brigde, melhor Speed Up no Xeon Westemere, e melhores índices de Eficiência no Core i7. Utilizou-se o Intel VTune para observar o comportamento deste problema em relação ao hardware. Efetuou-se a análise General Exploration, nas arquiteturas de CPU e viu-se que haviam altas taxas de LLC Miss. O Intel Core i7 consegue os melhores resultados em Eficiência, pois ele possui maior quantidade de cache por core do que os demais processadores. Sendo assim, ocorrem menos operações de cache miss. Ao compararmos o desempenho da arquitetura Xeon em relação ao desempenho do Xeon Phi, que possui uma quantidade de cores ainda maior e consequentemente uma quantidade de memória cache por core, ainda menor, obteve-se uma eficiência ainda menor. Como o Jogo da Vida é uma aplicação que possui uma limitação de paralelismo, o uso das instruções AVX e FMA apresentou melhora no desempenho, mas não o suficiente para superar o uso do OpenMP.

MHOLline 2.0 – A evolução de um *workflow* científico para problemas em Bioinformática e Biologia Estrutural

Victor Crisóstomo Cruz Reis (v.crisostomo.reis@gmail.com)[1], Priscila V. Z. Capriles Goliatt (capriles@lncc.br)[1,2], Laurent E. Dardenne (dardenne@lncc.br)[2]
1 – Departamento de Ciência da Computação – Universidade Federal de Juiz de Fora – UFJF/MG
2 – Grupo de Modelagem Molecular de Sistemas Biológicos – Laboratório Nacional de Computação Científica – GMMSB/LNCC

As bases para a primeira versão *web* do *workflow* científico MHOLline foram apresentadas em 2004 na tese de doutorado de Shaila C. S. Rössle[1] defendida pela UFRJ. Através de um conjunto de *softwares* voltados para a análise sequencial e estrutural de proteínas, o MHOLline busca auxiliar em pesquisas nas áreas de Bioinformática e de Biologia Computacional. Em 2010, a parceria da UFRJ com o Laboratório Nacional de Computação Científica proporcionou melhorias ao MHOLline[2], sendo lançada sua primeira versão *web* no sítio eletrônico www.mholline.lncc.br[3-4].

O *workflow* foi definido como um ambiente computacional dividido em: (i) MHOLcore, código responsável pela geração, processamento e manutenção de dados; (ii) MHOLweb, interfaceia a submissão dos processos (ou *jobs*), a exibição de seu progresso e os resultados gerados; (iii) MHOLdb, base de dados usada para armazenamento e consulta de dados de entrada, de resultados e do controle do sistema.

O MHOLline foi desenvolvido para ambiente Linux. No MHOLcore foram utilizadas as linguagem de programação Perl, ShellScript, C e Python, sendo os três últimos usados para a compatibilização e comunicação entre algumas ferramentas e o servidor. O MHOLdb utiliza o SGBD MySQL. Já a estrutura do MHOLweb foi desenvolvida com as linguagens PHP e JavaScript, além de HTML e CSS.

Existem dois tipos básicos de acesso: com ou sem login. O usuário não logado pode submeter novos *jobs*, porém sendo limitado à até 50 sequências de aminoácidos. O usuário logado pode ser classificado como: (i) Simples, pode somente submeter e gerenciar seus *jobs* (sem restrições); (ii) Mantenedor, tem alguns privilégios administrativos para auxiliar na manutenção de pequenas tarefas do sistema; (iii) Administrador, possui a disposição todas as ferramentas de manutenção do servidor, da base de dados e das bases de consulta para manter o MHOLline funcionando e atualizado. Conta também com ferramentas de análises estatísticas.

Para a submissão de um novo *job* o usuário deve inicialmente selecionar os módulos que deseja executar. Algumas ferramentas possuem interdependência, portanto, ao selecionar uma delas automaticamente as outras serão selecionadas. Após a seleção, deve-se fazer upload de um

arquivo no formato FASTA contendo as sequências de aminoácidos a serem analisadas (*e.g.*, desde uma única sequência até todo um genoma). Um usuário não logado poderá opcionalmente indicar seu endereço de e-mail para que receba um comunicado da conclusão de seu *job*. Caso não informe seu e-mail o endereço fornecido pelo navegador deverá ser guardado.

O primeiro módulo a ser executado é o BLAST[5]. Ele busca por proteínas evolutivamente relacionadas, através da comparação do alinhamento local das sequências submetidas com o as que possuem estrutura tridimensional (3D) conhecida e depositada no PDB (Protein Data Bank)[6]. O programa BATS foi desenvolvido para classificar as sequências identificadas pelo BLAST em quatro grupos (G0, G1, G2, G3) de acordo com sua pontuação obtida após a análise do resultado da comparação das sequências. Somente as sequências em G2 são destinadas aos próximos módulos do *workflow*. A ferramenta FILTERS foi desenvolvida para classificar as sequências selecionadas pelo BATS em sete grupos distintos, separando de acordo com a qualidade do modelo (de *Very High* até *Very Low*). O programa ECNGet foi desenvolvido para fazer a atribuição de ao menos um número *Enzime Comission* (EC) para cada sequência a ser modelada cujas proteínas de referência possuem pelo menos uma função enzimática conhecida.

A construção do modelo 3D é realizada pelo programa MODELLER[7], lembrando que apenas as sequências pertencentes a G2 chegam até este estágio. Os modelos construídos são avaliados de acordo com sua qualidade estereoquímica pelo programa PROCHECK[8]. O MHOLline conta ainda com o programa HMMTOP[9] para a identificação de regiões transmembranares em proteínas e pode ser executado independente dos demais.

Ao final da execução de cada módulo selecionado pelo usuário dentro do *workflow*, são disponibilizados arquivos com seu resultado padrão (*Output Files*), o resumo de execução (*Resume Files*) e os novos arquivos no formato FASTA (*FASTA Files*). Adicionalmente, o MHOLline gera um arquivo de resumo (*Summary*) contendo uma descrição das informações produzidas em cada etapa do *workflow*.

Em 2013 deu-se início ao desenvolvimento da versão 2.0 do *workflow* científico MHOLline, que consiste mais detalhadamente em: (i) atualizar os módulos já pertencentes ao *workflow*; (ii) adionar novas ferramentas de Biologia Computacional; (iii) refatorar seu código, além de sua padronização e documentação; (iv) conferir mais segurança ao sistema buscando soluções para vulnerabilidades diversas; (v) criar ferramentas para visualização *online* dos resultados; (vi) criar uma interface de reprocessamento de resultados (selecionados pelo usuário); (vii) além da elaboração de uma nova interface *web,* levando em conta a compatibilidade de *layout* e de funcionamento em cada um dos principais navegadores, de acordo com cada conceito de IHC.

Até o presente momento, os programas advindos da versão 1.0 já foram atualizados e novos programas já foram acoplados, como o MOLPROBITY[10], para a validação estereoquímica das estruturas 3D modeladas, o SIGNALP[11], para identificação de regiões de peptídeo sinal em

sequências proteicas, o TMHMM[12], para a predição de regiões de hélices em transmembrana e o PSIPRED[13], para a predição de estrutura secundária.

À medida que o sistema é refatorado e novas funcionalidades são adicionadas, o código gerado é documentado e padronizado e as correções de vulnerabilidades são feitas. Diagramas UML (*Unified Modeling Language*) foram gerados para ajudar no entendimento de cada parte do *workflow*, desde o MHOLcore até o MHOLweb. A interface *web* do *workflow* foi reescrita e um novo *layout* de resultados está sendo elaborado, ambos para melhorar a experiência de uso do usuário.

Os próximos passos são: adicionar mais ferramentas de Biologia Computacional; reescrever a área administrativa, adaptando-a às novas ferramentas e necessidades do sistema; finalizar a refatoração, padronização da documentação de todo o código do *workflow*; realização de testes de carga; lançar uma versão 2.0-beta do

MHOLline para testes de usabilidade; coletar e processar os relatórios de bugs e sugestões de usuários; e por fim dar como finalizada a versão 2.0 do *workflow* científico MHOLline.

Referências

[1] Rössle, SCS (2004). **Desenvolvimento de um sistema computacional para modelagem comparativa em genômica estrutural: Análise de seqüências do genoma da** *Gluconacetobacter diazotrophicus*. Tese de Doutorado, Instituto de Biofísica Carlos Chagas Filho – Universidade Federal do Rio de Janeiro / Brasil.

[2] Goliatt, PVZC. (2007). **Técnicas de Bioinformática e Modelagem Computacional Aplicadas ao Estudo do Genoma de** *Trypanosoma cruzi* **e de Enzimas Consideradas de Interesse no Tratamento da Doença de Chagas: Estudo Particular das Cruzipaínas 1 e 2**. Dissertação de Mestrado, Laboratório Nacional de Computação Científica.

[3] Capriles, PVSZ; Guimarães, ACR; et al. (2010). BMC Genomics, 11:610. doi:10.1186/1471-2164-11-610.

[4] Guimarães, ACR; Capriles, PVSZ; et al. (2013). **In: iConcept Press Ltd. Genomics II Bacteria, Viruses and Metabolic Pathways.** ISBN: 978-1-480254-145.

[5] Altschul S.F., Gish W., Miller W., Myers E.W. and Lipman D.J. (1990) **Basic local alignment search tool.** J. Mol. Biol. 215: 403-410.

[6]H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne (2000) **The Protein Data Bank Nucleic Acids Research**, 28: 235-242.

[7] N. Eswar, M. A. Marti-Renom, B. Webb, M. S. Madhusudhan, D. Eramian, M. Shen, U. Pieper, A. Sali. **Comparative Protein Structure Modeling With MODELLER.** Current Protocols in Bioinformatics, John Wiley & Sons, Inc., Supplement 15, 5.6.1-5.6.30, 2006.

[8] Laskowski R A, MacArthur M W, Moss D S, Thornton J M (1993). **PROCHECK - a program to check the stereochemical quality of protein structures.** J. App. Cryst., 26, 283-291.

[9] Tusnady GE, Simon I. (2001) **The HMMTOP transmembrane topology prediction server**. Bioinformatics 17:849-50.

[10] Vincent B. Chen, W. Bryan Arendall III, Jeffrey J. Headd, Daniel A. Keedy, Robert M. Immormino, Gary J. Kapral, Laura W. Murray, Jane S. Richardson and David C. Richardson (2010) **MolProbity: all-atom structure validation for macromolecular crystallography.** Acta Crystallographica D66: 12-21.

[11] **SignalP 4.0: discriminating signal peptides from transmembrane regions** Thomas Nordahl Petersen, Søren Brunak, Gunnar von Heijne & Henrik Nielsen Nature Methods, 8:785-786, 2011

[12] Lo A., Chiu Y.Y., Rødland E.A., Lyu P.C., Sung T.Y., and Hsu W.L. (2009) **Predicting helix-helix interactions from residue contacts in membrane proteins, Bioinformatics** (2009) 25: 996-1003. doi:10.1093/bioinformatics/btp114

[13] Jones, D.T. (1999) **Protein secondary structure prediction based on position-specific scoring matrices**. J. Mol. Biol. 292:195-202.

Técnicas de Otimização sem Uso de Derivadas

Bolsista: Viviane de Jesus Galvão

Orientador: Helio José Corrêa Barbosa

Pode-se encontrar nas Ciências Exatas e Engenharias problemas de otimização com restrições, em espaço de busca contínuo, que possuem análise de sua diferenciabilidade inviável e sua avaliação com alto custo computacional. Estas dificuldades induzem o uso de uma classe de métodos apropriada: os métodos de Otimização sem Derivadas.

O trabalho aqui apresentado propõe a combinação de métodos estocásticos e determinísticos para resolver os tipos de problemas citados. Como métodos estócasticos foram estudadas as metaheurísticas de Otimização por Enxame de Partículas e Estratégia Evolutiva. A Otimização por Enxame de Partículas é um método que foi inspirado no comportamento social de bandos de pássaros a procura de alimentos, simulando os mecanismos de inteligência coletiva. A Estratégia Evolutiva é um método baseado no modelo biológico de processos de adaptação e evolução de espécies, para o qual tem-se uma população de indivíduos sujeitos à mutação e seleção, onde os mais aptos tem mais chance de sobreviver. A estratégia determinística adotada aqui é uma Busca Padrão, um método que não faz uso das derivadas da função objetivo e que opera movimentos exploratórios através de dois passos: o passo de sondagem do espaço e o passo de avaliação, que é executado se o passo de sondagem falhar. É proposto também modificar o passo de avaliação da Busca Padrão adotada aqui com o intuito de realizar menos cálculos da função objetivo, fazendo a memorização da última coordenada na qual se obteve sucesso e, na sua próxima execução, a busca começa pela coordenada posterior.

A proposta da combinação dessas duas classes de métodos induz a junção das características positivas de tais, com o intuito de usar menos cálculos da função objetivo e procurando obter os melhores resultados possíveis.

Experimentos computacionais são feitos para a resolução de problemas-testes disponíveis na literatura e os resultados encontrados são utilizados para analisar o impacto no desempenho pela incorporação da Busca Padrão nas metaheurísticas consideradas aqui. Com base nos experimentos realizados, os resultados da combinação das metaheurísticas com a Busca Padrão são bastante competitivos em relação aos resultados das metaheurísticas puras.

Uso do Teste de Hipótese de Granger na Investigação de Causalidade entre Variáveis Escolares e Evasão[1]

Aluno: Weslei Peter de Oliveira (Faeterj)
Orientador: Dr. José Karam Filho (LNCC)
Coorientador: Dr. Fabiano Saldanha Gomes de Oliveira (IMS/UERJ)

1. Introdução

Com o avanço constante da tecnologia é fácil perceber como a informação passou a se transmitir de forma mais rápida, permitindo agilidade e praticidade em atividades que antes eram realizadas manualmente. Hoje muitas dessas atividades são executadas a partir de sistemas ou ferramentas. A tecnologia pode ser utilizada em qualquer área: saúde, transporte, esporte, finança, dentre outras.

Na área da educação não é diferente, muitas instituições de ensino estão se atualizando e aprimorando tecnologicamente, fazendo com que os processos realizados dentro de um cenário acadêmico se tornem mais simples e fáceis, sem perder a confiabilidade da informação.

A Instituição FAETERJ – Petrópolis, antes chamada Instituto Superior de Tecnologia em Ciência da Computação de Petrópolis, existe desde 2002 e, desde então, ocorreu naturalmente o aumento do número de alunos matriculados. Devido a isto o número de dados acadêmicos aumentou consideravelmente.

O presente relatório mostra um sistema desenvolvido para organizar e prover dados, de forma a subsidiar estudos sobre o desenvolvimento de instituições de educação utilizando ferramentas e metodologias de tecnologia da informação; utilizando o sistema desenvolvido, aqui são analisados três possíveis causas de evasão em escola superior.

2. Objetivos

O trabalho aqui desenvolvido faz uso de métodos numéricos e modelagem computacional [1], aplicados no desenvolvimento de metodologias quantitativas que permitam de uma maneira inovadora entender a fenomenologia que atua nos processos educacionais, com estudo de caso específico de evasão [2].

[1]Este trabalho faz parte dos temas do projeto Sistema Inteligente de Gestão da Educação LNCC/ISTCC/FAETEC, desenvolvido no LMCA/LNCC.

3. Metodologia

Para a realização dos objetivos deste trabalho, foram realizados no período deste relatório estudos e implementações computacionais como:

- Implementação de um banco de dados com informações acadêmicas
- Visualização de dados
- Análise de causalidade

O banco de dados foi desenvolvido em Java e Mysql.

O modelo para investigar a causalidade usado neste trabalho baseia-se no teste de Granger [4]. A ideia central deste teste é medir a correlação temporal de duas variáveis para identificar se uma delas influencia significativamente a outra.

Sejam as variáveis aleatórias *U* e *Y*. Suponha que temos uma amostra de tamanho *T* destas variáveis ao longo do tempo. Logo *U=u(1), u(2), ..., u(T) e Y=y(1), y(2),..., y(T)*. As variáveis estão relacionadas pela equação a diferenças de grau p:

$$y(k+1)=\sum_{i=1}^{p}\alpha_i y(k-i+1)+\sum_{i=1}^{p}\beta_i u(k-i+1)+e_1(k). \quad (1)$$

A hipótese nula H_0 do teste é dada por: H_0: $\beta_i=0$, $i=1,2,...,p$. Ou seja, u não afeta y até o atraso de valor *p* intervalos. Para determinar o parâmetro *p*, que define quantos valores passados de y influenciam o seu valor presente, usamos a equação do tipo auto regressiva:

$$y(k+1)=\sum_{i=1}^{p}\gamma_i y(k-i+1)+e_0(k). \quad (2)$$

Usamos a metodologia de Box & Jenkins [5] para estimar os coeficientes de (1) e (2).

Para calcular os resíduos ou erros quadráticos de (1) e (2), sejam R_0 e R_1 as somas dos erros quadráticos, assim:

$$R_0=\sum_{i=1}^{T}e_0^2(i) \quad \text{e} \quad R_1=\sum_{i=1}^{T}e_1^2(i). \quad (3)$$

O teste de hipótese de Granger se baseia na estatística representada pela variável aleatória *g*, que segue uma distribuição *F* de Fisher para precisão do teste de 90%. Ou seja F(0,1;p;T-2p-1). A função F(0,1;p;T-2p-1) será chamada daqui em diante de *F(p,T -2p -1)*. Para uma precisão maior do teste a

amostra se mostrou inconclusiva. Logo para obter uma maior precisão precisamos de mais dados. Portanto para a amostra disponível temos:

$$g = \frac{\frac{R_0 - R_1}{p}}{\frac{R_1}{T - 2p - 1}} \sim F(p, T - 2p - 1). \quad (4)$$

Se *g* é maior que o valor tabelado para *F* com precisão de 90%, rejeitamos a hipótese nula e a variável *u* não causa *y*. Caso contrário, temos uma forte evidência de que *U* causa *Y*. O teste de Granger compara o erro da equação (1), uma equação de um modelo auto regressivo a médias móveis ou ARMA, com a equação (2) de um modelo auto regressivo ou AR. Se o teste indicar forte evidência de que os coeficientes β na equação (1) valem zero, então a variável aleatória *Y* depende apenas de valores do seu passado e não é influenciada pela variável aleatória u. Caso contrário, u influencia ou causa y. Na prática valores grandes de g (tipicamente ≥ 1), indicam forte evidência de causalidade ou rejeição da hipótese nula.

4. Resultados Obtidos

Os dados utilizados neste estudo foram cedidos pelas instituições FAETERJ Petrópolis e CPTI petrópolis. A FAETERJ possui um curso de Tecnólogo em Tecnologia da Informação de nível superior e o CPTI um curso de nível médio concomitante em informática.

Usamos como teste a análise do efeito sobre a variável quantidade de evasão no semestre, do curso superior da FAETERJ Petrópolis (*Y)*, medido separadamente das variáveis:

1- média por semestre do desempenho dos alunos no curso superior da FAETERJ Petrópolis (U_1);
2- quantidade de alunos, por semestre, que entram para o mercado de trabalho antes do término do curso superior (U_2);
3- quantidade de alunos por semestre , que concluiu o curso técnico do CPTI e ingressou no curso superior da FAETERJ Petrópolis (U_3).

Desde o ano de 2004 até a realização deste estudo, os gestores se perguntaram como cada um desses efeitos influencia o fenômeno evasão do curso superior de tecnólogo da FAETERJ Petrópolis. Os resultados mostrados a seguir respondem a esta questão.

A variável evasão funciona como a variável *Y* da Eq. 2. As variáveis testadas foram:

U_1: desempenho do aluno no curso superior;
U_2: partir para o mercado de trabalho antes de concluir o curso superior;

U_3: ter integralizado o curso técnico de informática do CPTI.

A amostra possui observações das variáveis *Y, U_1 e U_2* desde o primeiro semestre de 2004 até o primeiro semestre de 2015. A variável U_3 foi observada desde o segundo semestre de 2008 até o primeiro semestre de 2015. Testamos a causalidade desta variável com a série no mesmo período da variável *Y*.

Testamos respectivamente e separadamente como cada uma das variáveis *U_1, U_2 e U_3* influenciam *Y*. Realizamos três testes de hipótese para responder as perguntas (hipóteses nulas ou H_0):

1. O desempenho do aluno do curso superior causa evasão?
2. Partir para o mercado de trabalho, antes de terminar o curso superior causa evasão?
3. Não fazer o curso técnico do CPTI.

A seguinte tabela foi obtida:

séries	Valor de p	Tamanho da série (T)	Resultado da estatística do teste (g)	Valor da estatística F	Decisão tomada com 90% de certeza
U_1 e Y	3	22	0,4402	0,5754	**Aceitar H_0** ou forte evidência de que **U_1 não causa Y**
U_2 e Y	3	22	1,3100	0,5754	**Rejeitar H_0** ou forte evidência de que **U_2 causa Y**
U_3 e Y	2	14	0,0038	0,0059	**Aceitar H_0** ou forte evidência de que **U_3 não causa Y**

Os resultados dos testes mostraram que:

1- *U_1 não influencia Y.* A estatística g é menor que F(0,1,22,22-6-1) ou aceitar a hipótese nula. Há uma forte evidência (90% de confiança), de que no curso superior, o desempenho dos alunos não causa evasão.

2- *U_2 influencia Y.* A estatística g é maior que F(0,1,22,22-6-1) ou rejeitar a hipótese nula. Há uma forte evidência (90% de confiança) de que o aluno que procura emprego antes de se formar no curso superior evade.

3- *U_3 não influencia Y.* A estatística g é menor que F(0,1,14,14-4-1) ou aceitar a hipótese nula. Há uma forte evidência (90% de confiança) de que os alunos que terminaram o curso técnico do CPTI e ingressaram no curso superior da FAETERJ integralizam o curso superior. Não evadem.

Referências

[1] Karam F., J. e Almeida, R. C., Introdução à Modelagem Matemática, Publicações da Pós-Graduação do LNCC, LNCC, 2003.

[2] Revista Rio Pesquisa Ano III, Nº. 9, publicada pela FAPERJ em Dezembro de 2009.

[3] Sommerville, J.; Engenharia de Software 9ª edição, Pearson 2011.

[4] Cabrera, J.B.; Lewis, L.; Qin, X.;Lee, W.; Prasanth, R.K.; Ravichandran, B.; Mehra, K.; Proactive Detection of Distributed Denial of Service Attacks using MIB Traffic Variables - A Feasibility Study, Proceedings of the 7th IFIP/IEEE International Symposium on Integrated Network Management, Seattle, WA - May 14-18, 2001.

[5] Box, G.E.P.; Jenkins, G.C.R.; Time Series Analysis, Wiley 1970.

Modelagem da rede complexa dinâmica formada pela malha aérea domésticabrasileira com Grafos MultiAspectos (MAGs)

Bolsista: Bernardo B. A. da Costa (CEFET/RJ – Petrópolis) - bantunes@lncc.br
Orientador: Artur Ziviani (LNCC) - ziviani@lncc.br
Colaboradores:
João Victor M. Bechara (CEFET/RJ – Petrópolis) - joaovmb@lncc.br
Klaus Wehmuth (LNCC) - klaus@lncc.br
Laura Assis (CEFET/RJ – Petrópolis) - laura.assis@gmail.com

Grafos MultiAspectos (MAG) foram introduzidos como uma generalização de grafo [1,2], capaz de representar redes complexas dinâmicas com múltiplas camadas, variantes no tempo, ou com ambas as características. Um conjunto de vértices, camadas, instantes de tempo ou qualquer outra característica independente presente na rede complexa pode ser considerado como um aspecto do MAG.

Este trabalho apresenta a modelagem da rede complexa dinâmica formada pela malha aérea domésticabrasileira através de um MAG. A estrutura resultante dessa modelagem consegue representar com naturalidade diferentes aspectos da malha aérea doméstica brasileira [3], de modo a facilitar futuras análises sob diferentes perspectivas bem como conseguir expressar de forma simples e compacta as diversas características dessa rede complexa.

Com base em dados publicamente disponíveis no sítio eletrônico da Agência Nacional de Aviação Civil (ANAC) no dia 3 de junho de 2015, pode-se representar amalha aérea domésticabrasileira referente ao período de uma semana com todos os voos em vigor nessa data. A malha aérea obtida é formada por 110 aeroportos, onde7empresas aéreas (Azul, Avianca, Gol, MAP, Passaredo, TAM, Sete) operam voos diariamente. São ao todo 19.601 voos distribuídos nos 7 dias da semana.

O modelo baseado em MAG proposto neste trabalho possui três aspectos (aeroportos, empresas aéreas, instantes de tempo). No primeiro aspecto trataremos os 110 aeroportos. No segundo aspecto, são representadas as 14 camadas referentes às 7 empresas e suas respectivas áreas de trânsito (cada empresa possui uma camada que representa seus voos e outra camada para representar conexões entre seus voos).O terceiro aspecto registra todos os 9.296 instantes de

tempo, equivalentes aos momentos de ocorrência de cada um dos eventos (p.ex. partida de voo, chegada de voo, embarque, desembarque). A rede complexa dinâmica resultante é composta por um MAG com 71.346 vértices e 91.881 arestas.

O modelo construído a partir das atividades do bolsista possibilitou a representação dos voos da malha aérea doméstica brasileira de forma mais realística e a inclusão deinformações comoaeroportos de origem e de destino, horários de embarque e desembarque, duração dos voos, assim como da ciclicidade dos dias da semana, possibilitando também a representação decontinuidade no tempo. O modelo permite analisar toda a rede complexa dinâmica, que é multi-camada e variante no tempo, referente à malha aérea doméstica brasileira. Além da possibilidade de análise integrada da malha aérea, o modelo permite a análise sob diferentes perspectivas, taiscomo a malha aérea de cada empresa aérea, a malha aérea de cada dia da semana ou ainda a malha aérea de cada empresa aérea por dia da semana. Essa visão sob diferentes perspectivas é uma contribuição em relação aos trabalhos anteriores encontrados na literatura que analisam malhas aéreas[3,4,5]. É importante ressaltar que a adoção de um MAG para a representação e modelagem da malha aérea doméstica brasileira permite todas essas análises sob diferentes perspectivas utilizando-se um único objeto matemático.

Referências

[1] On MultiAspect Graphs, K. Wehmuth, E. Fleury, A. Ziviani, 2014, arXiv 1408.0943

[2] MultiAspect Graphs: algebraic representation and algorithms, K. Wehmuth, E. Fleury, A. Ziviani, 2015, arXiv 1504.07893

[3] Structural Properties of the Brazilian Air Transportation Network, G. S. Couto; A. P. C. da Silva; L. B. Ruiz; F. Benevenuto, Aceito para publicação nos Anais da Academia Brasileira de Ciências (ABC), 2015.

[4] The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles, Guimera et al. Proceedings of the National Academy of Sciences (PNAS), vol. 201, no. 22, May 2005.

[5] Decomposing multilayer transportation networks using complex network analysis: A case study for the Greek aviation network, D. Tsiotas, S. Polyzos, Journal of Complex Networks. 2015.

Apoio à descoberta de objetos escondidos em Big Data

João Guilherme Rittmeyer, Fabio Porto

Big Data tem sido um termo extensamente discutido recentemente, tanto na indústria quanto na academia. A disponibilização de conjuntos de dados cada vez maiores impõe desafios tecnológicos e de pesquisa. Este trabalho faz parte da dissertação de mestrado do aluno Amir Khatibi que pretende desvendar objetos escondidos por entre grandes volumes de dados [1].

A intuição do problema está em que em grandes volumes de dados o interesse deixa de ser nos objetos individuais e passa a ser em objetos complexos obtidos a partir da composição de objetos mais simples. Em [2] Khatibi et al apresentaram uma implementação paralela usando o sistema Hadoop[2] para implementar a funcão de descoberta de objetos em Big Data. Neste trabalho, investigamos novos algoritmos para composição de objetos e sua implementação no framework paralelo Spark [3].

Os frameworks Spark e Hadoop processam grandes volumes de dados e possuem algumas diferenças e características em comum. O Hadoop foi desenvolvido pela apache após o paradigma MapReduce da Google, foi adotado no mercado e no meio científico pelo desempenho e escalabilidade. O Spark teve a mesma concepção, mas basea-se no uso extensível de memória RAM para troca de dados entre funções.

O Spark apresenta a técnica resilient distributed dataset (RDD) que considera uma coleção particionada entre os nós de um cluster de elementos imutáveis que podem ser operados em paralelo. Utiliza as linguagens de programação Python e Scalla além do Java e usa o sistema de arquivos Hadoop Distributed File System (HDFS). Funções escritas em Spark podem ser executadas na memória RAM, assim como, em memória secundaria com uma diferença no desempenho. Os processos executados em memória RAM, por ser uma memória rápida, apresentam um resultado superior em relação a execução na memória secundária.

A implementação da descoberta de objetos em grandes volumes de dados utilizando o Spark permitirá que a implementação escale para conjuntos de dados bastante volumosos. Consideramos igualmente, integrar essa solução com o sistema QEF[4], permitindo que possamos combinar sequência centralizadas com opercnoes distribuídas.

Bibliografia

- [1] Khatibi, A., Porto, F., Unveiling Objects in Big data Using similarity approach, Many Faces of Distance Workshop, Campinas, SP, 2014.
- [2]White, T. (2015). *Hadoop: The Definitive Guide*. O'Reilly Media, Inc., 4st edition.
- [3] Karau, H., Konwinski, A., Wendell, P and Zaharia, M (2015). Learning Spark: "*Lightning-Fast Data Analysis*". O'Reilly Media, Inc.
- [4] Porto, F., Tajmouati, O., Silva, V. F. V. D., Schulze, B., and Ayres, F. V. M. (2007). Qef - supporting complex query applications. In *CCGRID '07: Proceedings of the Seventh IEEE*

International Symposium on Cluster Computing and the Grid, pages 846–851, Washington, DC, USA. IEEE Computer Society.

Modelagem do Crescimento da Bactéria Patogênica Staphylococcus aureus N315 com a Análise de Balanço de Fluxo (FBA)

Leonardo Carvalho Gall Ilharco Morgado, Marcelo Trindade dos Santos,
Marisa Fabiana Nicolás, Maiana de Oliveira Cerqueira e Costa

Staphylococcus aureus é uma bactéria Gram-positiva responsável por diversas infecções adquiridas tanto em hospitais como em comunidades. A emergência de isolados resistentes a diferentes antimicrobianos responsáveis por infecções com altas taxas de mortalidade (*S. aureus* resistentes à meticilina [MRSA]), assim como a capacidade de expressar uma grande variedade de fatores de virulência são uma preocupação mundial no que concerne à saúde pública. Sendo assim, a compreensão dos processos infecciosos e a possibilidade de elencar novos alvos moleculares para o desenvolvimento de drogas torna-se uma necessidade urgente.

A reconstrução de modelos metabólicos patogênicos e de humano têm sido empregados para o desenvolvimento de novas drogas para combater infecções bacterianas com o mínimo efeito colateral para o hospedeiro. O metabolismo é um processo de transdução de energia que ocorre através de uma rede complexa e coordenada de reações bioquímicas[1]. Em uma determinada condição ambiental, a célula expressa um conjunto específico de enzimas que produzem uma distribuição de fluxo particular na rede conhecido como "fenótipo metabólico" [2]. A reconstrução de modelos quantitativos detalhados é limitada pela dificuldade de se obter dados dos parâmetros cinéticos das enzimas, uma vez que muitos são desconhecidos e difíceis de serem medidos precisamente. Além disso, a complexidade e grande quantidade de reações, em uma rede de escala genômica, torna os modelos quantitativos difíceis devido a grande quantidade de dados e complexidade computacional necessária.

A ausência de informação detalhada e a dificuldade em aplicar os modelos quantitativos em escala genômica levou ao desenvolvimento de um método alternativo denominado de reconstrução e análise baseado em restrições (COBRA - do inglês *Constraint-based reconstruction and analysis*). O mesmo baseia-se em conceitos fundamentais como a imposição de restrições que limitam os fenótipos metabólicos possíveis, a identificação e descrição matemática de pressões seletivas evolutivas e uma perspectiva em escala genômica do metabolismo celular [3]. Com apenas a estequiometria e as restrições nos fluxos de entradas e saídas da rede, é possível a realização de análises *in silico* no modelo como a Análise de Balanço de Fluxo (FBA - do inglês *Flux balance analysis*). Sendo assim, simulações do fluxo metabólico através da rede podme fornecer predições do efeito de cada produto gênico sobre uma função específica da rede metabólica [3].

O modelo utilizado no projeto foi retirado do banco de dados *BIGG 2*(Biochemical Genetic and Genomic knowledgebase), que reúne modelos publicados de reconstruções metabólicas em escala genômica. O modelo possui a anotação de genes disponível no banco *NCBI*(*National Center for Biotechnology Information*) do genoma da bactéria *Staphylococcus aureus* N315. O arquivo disponível no banco é do tipo SBML (*System Biology Markup Language*) utilizado para representar modelos biológicos de redes metabólicas. O mesmo contém a lista de reações, metabólitos, sua relação estequiométrica e parâmetros cinéticos e de restrição de cada reação.

O balanço de fluxo foi calculado utilizando a ferramenta COBRA *Toolbox* integrada ao *Matlab.* A otimização no COBRA *Toolbox* é feita utilizando-se programação linear, onde é definida uma função objetivo e a partir das restrições da matriz estequiométrica a mesma é otimizada. A reação utilizada como função objetivo encontra-se no modelo e representa todos os compostos considerados essenciais para o crescimento da bactéria. Para identificar possíveis alvos para drogas, foram feitas restrições de fluxo em zero para reações de vias específicas da rede. Se uma reação inibida interfere no fluxo de crescimento da bactéria, anulando o fluxo de biomassa, um fármaco capaz de inibir a mesma reação é um possível candidato para o tratamento de infecções provocadas pela mesma. As reações inibidas foram selecionadas baseadas em vias essenciais da bactéria, sendo selecionados os genes da bactéria identificados nas vias de aminoácidos e carboidratos. Foi utilizado o banco *Uniprot para a identificação dos genes de S.aureus*, de modo a comparar com os genes encontrados no modelo, permitindo então a seleção de reações associadas às vias desejadas.

Para o metabolismo de aminoácidos foram identificados 52 genes no banco *Uniprot*, porém apenas 49 também foram encontrados no modelo de *S.aureus*. Dos 49 genes identificados verificou-se que 47 os mesmos estavam associados à enzimas, indicando que o modelo apresenta 47 reações de síntese de aminoácidos. Para testar se as enzimas são essenciais para o crescimento da bactéria, foi desenvolvido um script que restringe o fluxo de cada uma para zero, e que em seguida calcula a otimização da função de crescimento. Das enzimas testadas 37 bloquearam o fluxo da função objetivo, significando que essas reações são essenciais para o crescimento da bactéria.

O próximo passo na identificação das enzimas que poderão ser utilizadas como alvo para drogas, é verificar quais delas não são homólogas em humanos. Para realizar essa comparação buscou-se o arquivo fasta de cada uma das enzimas seguindo sua referência no banco NCBI. Foi feito então, para cada uma das enzimas anteriormente, um *Protein Blast* contra o genoma humano, utilizando o próprio banco NCBI. Foi selecionado apenas enzimas com um *E-value* maior que 1e-10, obtendo um conjunto de 22 enzimas como possíveis alvos para fármacos. Dos resultados obtidos destacam-se as enzimas utilizadas na via do *shikimato*, que é uma rota metabólica para a

síntese de fenilalanina, tirosina e triptofano. Essa via não é encontrada em humanos e algumas de suas enzimas são estudadas como possíveis alvos para drogas[4].

Como trabalho futuro, será realizado um aprimoramento dos dados, é de interesse testar as similaridades das enzimas com bactérias comensais. Obter um alvo com baixa semelhança entre bactérias comensais resultaria em uma droga mais específica às infecções de *S.aureus,* e com menor efeitos colaterais em humanos.Além disso será realizado uma análise filogenética das enzimas encontradas para verificar sua semelhança com outros organismos.

Referências Bibliográficas

[1] Jan Schellenberger, Junyoung O Park, Tom M Conrad, and Bernhard O Palsson. BiGG: a Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions. **BMC Bioinformatics**, 11(213), 2010.

[2] Amit Varma & Bernhard O. Palsson. Metabolic Flux Balancing: Basic Concepts, Scientific and Practical Use. **Nature Biotechnology**, 12(10):994–998, 1994.

[3] Nathan E Lewis, Harish Nagarajan, and Bernhard O Palsson. Constraining themetabolic genotype-phenotype relationship using a phylogeny of in silico methods.**Nature Reviews Microbiology**, 10(4):291–305, 2012.

[4] Coracini, J. D., and W. F. de Azevedo. "Shikimate kinase, a protein target for drug design." **Current medicinal chemistry** 21.5 (2014): 592-604.

Um Estudo Sobre Métodos Numéricos Aplicados à Simulação de Reservatórios Petrolíferos

Luís H. C. da Cunha[1], Cristiane O. Faria[1], Sandra M. C. Malta[2]

[1] Universidade do Estado do Rio de Janeiro, Departamento de Análise Matemática – IME, Rua São Francisco Xavier, 524, 6o andar, Bloco F, CEP:20550-900, Rio de Janeiro, RJ, Brazil

[2]Laboratório Nacional de Computação Científica, Av. Getúlio Vargas 333, CEP: 25651-075, Petrópolis, RJ, Brazil

RESUMO

Simulação de reservatórios vem se estabelecendo como a principal ferramenta utilizada por empresas de perfuração para tomada de decisões quanto à viabilidade de investimentos para a exploração de novos poços. A possibilidade de reproduzir o comportamento de certas características físicas dos reservatórios de petróleo a um custo relativamente muito mais baixo do que em testes de campo, prevendo performances futuras destes, torna esta ferramenta muito atrativa. No entanto, escoamentos de fluidos em meios porosos quando são modelados matematicamente usando equações diferenciais parciais (EDPs) devem levar em conta a presença das características físicas relevantes e resultam em modelos extremamente complexos, e para serem resolvidos é necessário a aplicação de diversas técnicas matemáticas sofisticadas.

Essas soluções numéricas são baseadas no conceito de discretização onde o meio é dividido em células ou pontos, derivadas são aproximadas por diferenças, e as EDPs se transformam em um sistema de equações. No caso de um escoamento monofásico incompressível em um meio poroso, o problema modelado pela equação de conservação de massa e a lei de Darcy, juntos formam o que chamamos de sistema de Darcy, que relaciona a velocidade do fluido no meio poroso (u) com o gradiente da pressão (p) através do tensor de permeabilidade. Considerando-se, em particular, o caso unidimensional, onde o domínio é o intervalo $[0, L] \in R$, com as propriedades $\phi(x) = \phi$(porosidade), $\rho(x) = \rho$(densidade) e μ(viscosidade) constantes, os efeitos gravitacionais desprezados ($\mathbf{g}$=0) e o tensor de permeabilidade $\frac{k(x)}{\mu} = k$, obtém-se o seguinte sistema:

$$u' = f em [0, L] \in R$$
$$u = -kp' em [0, L] \in R$$

Para resolver esse sistema, vários métodos podem ser aplicados e encontrados na literatura. Em particular, neste trabalho consideraremos dois métodos numéricos: o método de Diferenças Finitas (DF) (LeVeque, 2007; Biezuner, 2007), onde sua essência é a discretização do contínuo tornando o problema "finito", e isto o faz computável. Inicialmente, deve-se discretizar o domínio, ou gerar uma malha. Este procedimento consiste em aproximar o domínio contínuo por pontos discretos contidos nele. Usando um espaçamento constante Δx, entre os pontos, o intervalo [0, L] é subdividido em n subintervalos de comprimento $\Delta x = L/n$ e o domínio discreto é definido como $x_0 = 0, x_1 = \Delta x, \ldots, x_i = i\Delta x, \ldots, x_n = n\Delta x = L$. As aproximações serão calculadas nos pontos desta malha. O outro método é o de Volumes Finitos (VF) (Rodrigues, 2015; Fortuna, 2000), em que a discretização é feita dividindo o domínio em Volumes de Controle (VC) (ou células da malha). Aqui, foi utilizada a malha centrada na célula uniforme, onde são definidas as fronteiras do VC de modo similar ao utilizado em diferenças finitas, e pontos x_i são colocados no ponto médio de cada VC $\left(\frac{(x_i + x_{i+1})}{2}\right)$. Onde x_i é a posição da fronteira do VC e $\Delta x = x_i - x_{i-1}$ (esquematizado na Figura 1). A partir dessa discretização, integra-se em cada VC.

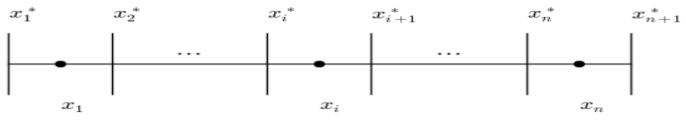


Figura 1: Representação dos volumes de controle (VC) em uma malha utilizada no método de VF.

Neste trabalho é apresentado um estudo comparativo onde avaliam-se a influência do número de pontos da malha, a escolha das aproximações, a aplicação de diferenças atrasada ou central na aproximação da condição de contorno e a influência da permeabilidade (meio heterogêneo) na obtenção das soluções aproximadas.

REFERÊNCIAS

Biezuner, R. J., 2007. Métodos Numéricos para Equações Diferenciais Parciais Elípticas - Notas de Aula.

Fortuna, A. O., 2000. Técnicas computacionais para dinâmica dos fluidos: conceitos básicos e aplicações. Edusp.

LeVeque, R. J., 2007. Finite Difference Methods for Ordinary and Partial Differential Equations Steady-State and Time-Dependent Problems. SIAM.

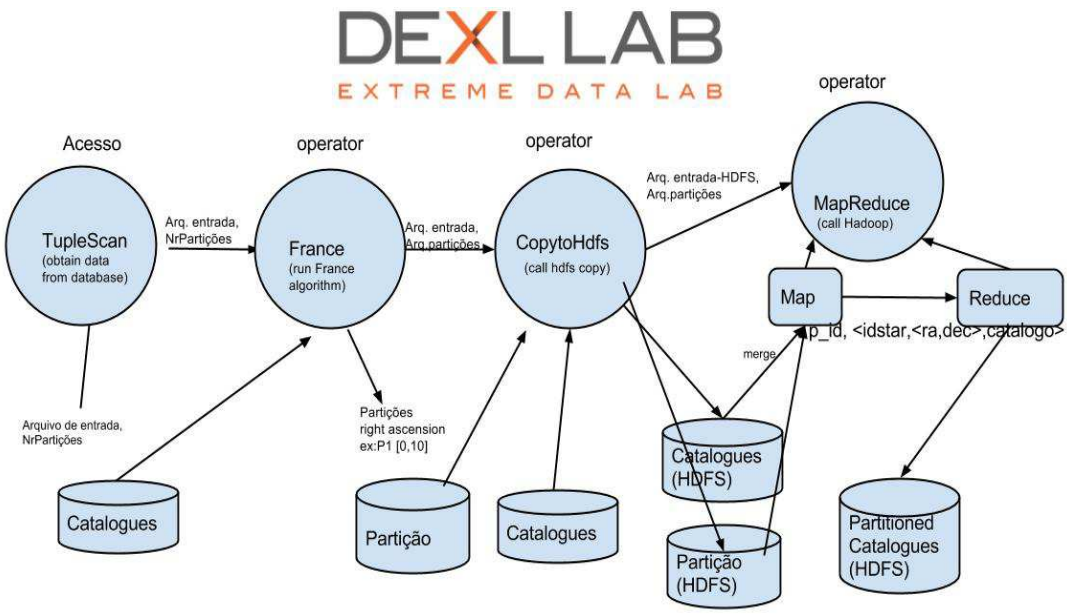
Rodrigues, J. R., 2015. Introdução à Simulação de Reservatórios Petrolíferos - Programa de Verão, LNCC 2015.

Particionamento de dados da astronomia utilizando os sistemas QEF e Hadoop

Rodrigo Botelho, Fabio Porto

Vivemos um período de grande aumento na capacidade de armazenamento de dados eletronicamente, estamos na era dos "Zettabytes". Contudo, esse aumento da capacidade de armazenamento não foi acompanhado pelo aumento da capacidade de leitura desses dados, acarretando um fator altamente desigual, pois em grandes volumes de dados perdemos muito tempo no processo de leitura, isso se utilizando de técnicas de processamento concentrado. Para contornamos este problema utilizaremos técnicas e ferramentas de processamento distribuído.

O objetivo deste trabalho está focado em técnicas para particionar, processar e, finalmente, integrar os dados, tornando seu resultado final integro. Ocorre que estratégias de particionamento de dados tipicamente agem de duas formas: (i) divisão física, em blocos de dados de um certo tamanho, (ii) particionamento de valores em uma chave de identificação. Em vários cenários, no entanto, o critério de particionamento se apresenta de forma mais complexa.. O particionamento per si, pode ser visto como um processo em duas etapas: a definição do critério de particionamento, e identificação dos dados com as respectivas partições; e a alocação das partições sobre o sistema distribuído. Desta forma, este trabalho apresenta um dataflow para particionamento e alocação de dados em um sistema distribuído.



Com este cenário em mente, estamos implementando o dataflow descrito acima.. Destaca-se em nossa implementação o framework QEF(Query Evalution Framework) , atuando de maneira à gerenciar a execução do dataflow. Faremos, igualmente, uso da framework Hadoop e o do seu sistema de arquivos distribuidos HDFS(Hadoop file system), no qual os dados serão atribuídos.

Sobre a implementação deste dataflow, iniciamos com as configurações do Framework QEF, onde alimentamos as informações de controle dentro do Query Execution Plan (QEP, arquivo xml, onde estão informações do plano de execução). no QEF, um “request” é um grafo de operações algébricas definidas pelo usuário, comunicando uns com os outros e com o objetivo de produzir um resultado. Estas operações estão representadas dentro do plano de execução. Posteriormente, implementamos nossas classes em java com tarefas específicas dentro da aplicação, estas classes serão nossa base dentro do plano de execução e irão acionar todas as tarefas de processamento dentro da aplicação.

AVALIAÇÃO DO TAMANHO DA POPULAÇÃO E DOS OPERADORES DE MUTAÇÃO DA EVOLUÇÃO DIFERENCIAL EM PROBLEMAS DE OTIMIZAÇÃO COM RESTRIÇÕES.

Samantha Vilaça de Almeida Souza[1 2], Eduardo Krempser[1 4], Helio J. C. Barbosa [2 3]

1 - Faculdade de Educação Tecnológica do Estado do Rio de Janeiro, Petrópolis, Rio de Janeiro, Brasil.
2 - Laboratório Nacional de Computação Científica, Petrópolis, Rio de Janeiro, Brasil.
3 - Universidade Federal de Juiz de Fora, Juiz de Fora, Minas Gerais, Brasil.
4 - Fundação Oswaldo Cruz, Rio de Janeiro, Rio de Janeiro, Brasil.

1. OTIMIZAÇÃO

Em geral a abordagem de problemas de otimização, presentes em diversas áreas do conhecimento humano, envolve a minimização ou maximização de uma função objetivo. Embora existam métodos que não façam uso de uma função objetivo, estes em geral são inferiores aos que façam a sua utilização (STORN e PRICE, 1997).

A função objetivo a ser minimizada ou maximizada está relacionada às variáveis utilizadas para definir o ponto ótimo, denominado de variáveis de decisão. Variáveis estas sobre as quais se tem controle e que serão modificadas com o objetivo de otimização. Os possíveis valores para as variáveis de decisão, por conseguinte, podem ainda ser delimitados por um conjunto de restrições impostas sobre as mesmas, formando assim um conjunto de soluções factíveis de um determinado problema (PAIVA, 2001).

Otimização é a busca pela melhor solução (valor ótimo) para um determinado problema dentro de um conjunto finito ou infinito de possíveis soluções.

Técnicas voltadas para obtenção de valores ótimos são implementadas apenas quando não há a aplicação de uma solução simples e diretamente calculável. Isso ocorre geralmente quando a estrutura do problema é complexa, ou possui diversas soluções.

O conceito de valor ótimo está diretamente ligado ao problema que se deseja otimizar. Por exemplo, em uma determinada situação “A” modelada matematicamente por uma única função

f(A) há a necessidade de obter-se um valor ótimo tal que f(A) seja mínimo, ou ainda, uma situação "B" cujo modelo matemático seja expresso por n funções f(Bn) (n = 1,2, 3,...,n) onde se pretende maximizar algumas e minimizar as demais. Neste caso, pode-se ter uma única solução, um conjunto de soluções ou ainda não haver solução que satisfaça todas as funções do sistema.

Portanto, ao aumentar o número de funções e variáveis do sistema, a dificuldade e complexidade de um determinado conjunto de soluções ótimas também aumentam. Com isso, surge à necessidade de desenvolver técnicas matemáticas e computacionais para aprimorar o processo de otimização.

Os algoritmos de otimização podem ser classificados em duas categorias: método determinísticos e meta-heurístico (ALMEIDA, 2013).

Métodos determinísticos são capazes de gerar uma sequência determinística de possíveis soluções, requerendo, na maioria das vezes, o uso de pelo menos de uma primeira derivada da função objetivo em relação às variáveis de projeto. Desta forma, a função objetivo deve ser contínua e diferencial em seu espaço de busca.

Além disso, sua aplicação a problemas multimodais (possui várias inflexões na sua superfície, o que a caracteriza com múltiplos pontos ótimos (OLIVEIRA, 2001)) pode ser desaconselhada, devido ao fato destes métodos possuírem formulações que lhes permitem convergência para uma solução ótima não necessariamente global (AMEIDA, 2013). Em alguns métodos determinísticos, a qualidade da solução obtida pode ser dependente do ponto de partida da busca, reafirmando assim, a limitação destes métodos quanto a sua aplicabilidade a problemas multimodais.

Já os processos de meta-heurísticas são classificados como métodos de alto nível que utilizam procedimentos de baixo nível para obter soluções consideradas satisfatórias para um determinado problema de otimização. Este processo também não garante obtenção do ótimo global, uma vez que muitos dos métodos pertencentes a esta classe utilizam-se de processos

estocásticos, fazendo com que o resultado dependa totalmente do conjunto de variáveis pseudoaleatórias geradas (ALMEIDA, 2013).

Segundo (FERREIRA, 1999), heurística é definida como sendo uma metodologia ou algoritmo responsável por resolver problemas que embora não rigorosos, refletem diretamente no conhecimento humano, proporcionando assim, uma solução satisfatória.

Os algoritmos pertencentes aos processos de meta-heurísticas apresentam diversas técnicas, dentre elas encontram-se os Algoritmos evolutivos, que são métodos computacionais fundamentadas em conceitos biológicos, especialmente aqueles relacionados à evolução e à genética. O principal conceito relacionado a esta técnica é a Teoria Sintética da Evolução, também conhecida como neodarwinismo (RIDLEY, 1996), a qual possibilita que através dos principais fatores evolutivos haja um incremento na aptidão dos indivíduos constituintes de uma população ao longo das gerações. Tendo como objetivo manter uma população de indivíduos que indiquem soluções candidatas para problemas que evoluem ao longo de gerações através de um processo de competição natural, onde os mais aptos obtenham maiores chances de sobrevivência e reprodução, ou seja, representem a solução ótima.

Desde DARWIN (1859), a teoria da evolução vem sendo a principal ideia unificadora nas mais diversas áreas da biologia, pois a seleção natural é a força propulsora que distingue os sistemas biológicos dos demais sistemas físicos e químicos.

A teoria da seleção natural não prevê apenas a ocorrência de variações sucessivas junto aos indivíduos de uma dada espécie, mas também indica o tipo de variação, as quais devem necessariamente conduzir o organismo a uma melhor adaptação ao meio.

2. ALGORITMOS EVOLUTIVOS

Algoritmos Evolutivos são métodos probabilísticos que imitam o processo de evolução natural (processo condutor do surgimento de novas estruturas mais complexas e bem adaptadas).

Estes algoritmos são empregados para resolução de problemas complexos desde a década de 80(BACK, 1997).

São baseados em população, na qual um indivíduo pode ser afetado por outros, assim como pode ser afetado pelo ambiente. Quanto mais adaptado for o individuo nestas condições, maior será a chance dele sobreviver por mais tempo e consequentemente gerar descendentes, que

por sua vez, herdarão suas informações genéticas. No curso da evolução, este cenário leva à disseminação de informação genética de indivíduos acima da média.

A estrutura de um algoritmo evolutivo pode ser representada conforme o algoritmo abaixo:

t ← 1;

inicializa P(t);

avalia P(t);

While not condição_de_parada **do**

Pd(t) ← variação[P(t)];

Avalia [Pd(t)];

P(t + 1) ← seleciona[Pd(t) U Q);

t ← t + 1;

end while

No algoritmo, P(t) simboliza a população de indivíduos na geração t. Q é o conjunto especial de indivíduos que devem ser considerados para a seleção. Uma população de descendentes Pd(t) é gerada por meio de operadores de variação como mutação e/ou recombinação aplicados à população P(t). A população de descendentes Pd(t) é avaliada calculando-se o valor da função objetivo para cada uma das possíveis soluções representadas pelos indivíduos de Pd (t).

Em relação ás diferenças entre os diversos Algoritmos evolucionistas existentes, podemos identifica-las principalmente pela forma com que os indivíduos são efetivamente representados, dos operadores utilizados e do mecanismo de seleção. Dentre as principais técnicas vinculadas a esta classe de algoritmos, podemos citar: Algoritmos Genéticos, Programação Genética, Sistemas Classificadores e Evolução Diferencial.

3. EVOLUÇÃO DIFERENCIAL

A Evolução Diferencial do inglês Differential Evolution (DE) é um processo meta-heurística estocástica simples, baseada nos mecanismos da evolução natural das espécies e da genética de populações, que utiliza operadores de mutação, cruzamento e seleção natural para gerar novos indivíduos mais aptos.

Foi desenvolvida por Rainer Storn e Kenneth Price em meados da década de 90 a partir de tentativas de resolução do problema de ajuste polinomial de Chebychev (STORN e PRICE, 1997).

A DE tem gradualmente recebido crescente interesse e se tornado um algoritmo cada vez mais popular nos últimos anos, sendo utilizado em muitos casos práticos, principalmente por demonstrar boas propriedades de convergência e de fácil entendimento (PAIVA, 2011).

Seu sucesso deve-se principalmente a seu mecanismo de busca, o qual usa a diferença entre dois vetores, escolhidos aleatoriamente das soluções candidatas, para produzir novas soluções. À medida que a população evolui, a direção e o tamanho do passo da busca na mutação mudam ajustando-se de acordo com a distribuição da população no espaço.

Um fator importante sobre a DE é que ela é eficiente mesmo voltada para: populações pequenas, funções descontínuas, multimodais e não lineares, pois possui simplicidade, rápida convergência e precisão.

A literatura cita que em contraste com a maioria dos algoritmos evolutivos, onde muitos parâmetros precisam ser ajustados, a evolução diferencial apresenta poucos parâmetros, os quais são definidos como: POP (Número de vetores/indivíduos mantidos na população), GEN (Número

de gerações utilizadas), F (Valor de ponderação da diferença empregada na mutação) e CR (Probabilidade de ocorrência da recombinação).

Após definirmos todos os parâmetros apresentados anteriormente, torna-se necessário gerar uma população inicial de POP indivíduos, de maneira totalmente aleatória (ALMEIDA, 2013).

Em seguida, ao longo de GEN gerações, cada um dos indivíduos pertencentes à população, denominados vetores alvo, competirão por uma posição na geração subsequente com um novo indivíduo gerado através dos operadores de recombinação e mutação. A geração de um novo

indivíduo, chamada de vetor teste, inicia-se com a escolha de indivíduos pertencentes à população, e que serão utilizados no processo de mutação. Após essa escolha, a mutação é realizada para cada uma das dimensões do problema, representadas por cada valor contido nos indivíduos.

A premissa principal da mutação é a geração de novos indivíduos, denominados de vetores modificados ou doadores, pela adição da diferença ponderada entre dois indivíduos aleatórios da população a um terceiro indivíduo, a princípio também selecionado aleatoriamente.

Os componentes do indivíduo doador são misturados com as componentes de um indivíduo escolhido aleatoriamente (denotado vetor alvo), resultando no vetor experimental. O processo de misturar os parâmetros é referido frequentemente como "cruzamento" na comunidade dos algoritmos evolutivos (OLIVEIRA e SARAMAGO, 2015).

Se o vetor experimental resultar em um valor da função objetivo menor que o vetor alvo, então o vetor experimental substitui o vetor alvo na geração seguinte. Este processo é denominado de seleção.

Em seu modelo básico, a DE emprega um operador de recombinação, o qual realiza a intercalação de valores gerados pela mutação com valores presentes em um indivíduo alvo. Para isso, determina-se uma probabilidade CR de ocorrência de recombinação. Na geração de um novo indivíduo, a mutação é realizada com probabilidade CR, caso contrário, o valor presente no indivíduo alvo é copiado diretamente ao novo indivíduo gerado. Para garantir a realização da mutação em ao menos uma variável, seleciona-se anteriormente à geração de um novo indivíduo, em um dos componentes do mesmo, onde a mutação é realizada independentemente da probabilidade CR. Desta maneira, evita-se que o vetor teste seja uma cópia do vetor alvo. Tal modelo de recombinação é denominado binomial, uma vez que o número de valores copiados diretamente do vetor alvo para o vetor teste possui uma distribuição aproximadamente binomial (ALMEIDA, 2013).

Cabe ressaltar que o critério para seleção dos indivíduos e o número de diferenças empregadas na mutação, bem como o modelo de recombinação utilizado, poderão variar, cada uma dessas possíveis combinações é denominada variante do DE.

Por fim, o vetor teste é submetido à função objetivo e o valor produzido é comparado ao do vetor alvo. O indivíduo com menor valor (problemas de minimização) ou maior valor (problemas de maximização) é alocado na população pertencente à geração subsequente, sendo esta operação denominada substituição. Este processo é repetido até que o número de gerações máximas ou um critério de parada pré-estabelecido seja alcançado (ALMEIDA, 2013).

A figura abaixo representa um algoritmo básico de DE (KREMPSER, 2014):

Figura 01 - Algoritmo de Evolução Diferencial.

Algorithm 1: Algoritmo DE/rand/1/bin

input : NP (tamanho da população), GEN (número de gerações), F (ponderação da mutação), CR (taxa de recombinação)

```
1  G ← 0;
2  Cria_População_Inicial_Aleatória(NP);
3  for i ← 1 to NP do
4  |  Evaluate f(x⃗_{i,G}) ;        /* x⃗_{i,G} é um indivíduo na população */
5  for G ← 1 to GEN do
6  |  for i ← 1 to NP do
7  |  |  Seleciona_Aleatoriamente(r1, r2, r3) ;      /* r1 ≠ r2 ≠ r3 ≠ i */
8  |  |  jRand ←RandInt(1, N) ;      /* N é o número de variáveis */
9  |  |  for j ← 1 to N do
10 |  |  |  if Rand(0, 1) < CR or j = jRand then
11 |  |  |  |  u_{i,j,G+1} = x_{r3,j,G} + F.(x_{r1,j,G} − x_{r2,j,G});
12 |  |  |  else
13 |  |  |  |  u_{i,j,G+1} = x_{i,j,G};
14 |  |  if f(u⃗_{i,G+1}) ≤ f(x⃗_{i,G}) then
15 |  |  |  x⃗_{i,G+1} = u⃗_{i,G+1};
16 |  |  else
17 |  |  |  x⃗_{i,G+1} = x⃗_{i,G};
```

4. CONFIGURAÇÃO DE PARÂMETROS

A definição da representação das soluções e da função objetivo são itens que formam a ponte entre o contexto original do problema que se deseja tratar e o método de resolução. Quando este método é um algoritmo evolutivo, precisamos definir seus componentes, como operadores de variação que sejam adequados à representação, o mecanismo de seleção natural e a população

inicial. Cada um destes componentes deve possuir parâmetros, em princípio: a probabilidade da mutação, o número de indivíduos envolvidos na seleção e o tamanho da população. Os valores destes parâmetros determinam fortemente a qualidade das soluções encontradas e o tempo gasto para encontra-las (EIBEN, et al., 1999).

Normalmente a escolha dos parâmetros é feira por tentativa e erro numa bateria de testes preliminares, que por sua vez demandam um tempo considerável. Para tratar esta questão métodos de configuração automática de parâmetros vem sendo desenvolvidos. Estes métodos podem ser classificados de três formas (EIBEN, et al., 1999):

Determinísticos (também conhecidos como dinâmicos): utilizam regras determinísticas para modificar os parâmetros. Este processo é realizado sem a utilização de *feedback* do processo de busca (ALMEIDA, 2013).

Adaptativos: incorporam algum tipo de *feedback* do processo de busca para guiar o ajuste dos parâmetros de controle (ALMEIDA, 2013).

Auto-adaptativos: os parâmetros a serem configurados são codificados diretamente na representação do indivíduo. Assim, a qualidade dos parâmetros codificados será responsável por determinar a qualidade das soluções geradas. Estas soluções, por sua vez, possuem uma maior chance de sobreviver e produzir descendentes, propagando estes bons valores de parâmetros (ALMEIDA, 2013).

OBJETIVO

O objetivo do trabalho será analisar o impacto dos diferentes parâmetros envolvidos na aplicação da Evolução Diferencial, sendo inicialmente avaliado o comportamento da técnica para diferentes tamanhos da população, incluindo a adaptação desse valor durante o processo evolutivo. O correto ajuste desse parâmetro ou sua adaptação podem refletir diretamente na otimização, reduzindo os custos computacionais ao diminuir a população necessária para a obtenção de valores ótimos ou ampliar a busca em momentos de estagnação.

Na literatura existem modelos propostos para definição do tamanho da população, os quais na maioria das vezes divergem sobre as regras a serem adotadas (ALMEIDA, 2013).

Em (STORN; PRICE, 1995), é indicado que um valor adequado para o número de indivíduos na população deve ser escolhido entre 5-D e 10-D, onde D representa a dimensionalidade do problema.

Já em (GAMPERLE; MULLER; KOUMOUTSAKOS, 2002), utilizando-se as funções *Sphere*, *Rosenbrock* e *Rastrigin*,diferentes configurações de parâmetros são analisadas. Onde os resultados referentes ao tamanho da população estarão entre 3-D a 8-D.

Em (ZAHARIE, 2002), o relacionamento entre os parâmetros de controle e a variância da população é utilizado para a elaboração de um modelo responsável por fornecer valores que evitem a convergência prematura através da manutenção da diversidade da população.

REFERÊNCIAS

ALEXANDRE M. OLIVEIRA - Tcc (2001). *Algoritmos evolutivos para problemas de otimização numérica com variáveis reais.*

ANA O. PAIVA – Dissertação de Mestrado (2011). *Aplicação do método de evolução diferencial.*

BACK, T. HAMMEL, U. e SCHWEFEL, H. –P (1997). Evolutionary Compotation: *Comments on the history and current state*. IEEE TRANSACTIONS ON EVOLUTIONARU COMPUTATION.

EDUARDO KREMPSER DA SILVA – Tese de Doutorado (2014). *Uso de Metamodelos na Evolução Diferencial para Problemas Envolvendo Simulação de Alto Custo Computacional.*

EIBEN, A. E. et al. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 1999.

FERREIRA, A. B. H. *Novo Aurélio Século XXI: o dicionário da língua portuguesa*. [S.l.]: Editora Nova Fronteira S. A., 1999.

GAMPERLE, R.; MULLER, S. D.; KOUMOUTSAKOS, P. A parameter study for differential evolution. In: *WSEAS Int. Conf. on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*. [S.l.]: Press, 2002. p. 293–298.

OLIVEIRA e SARAMAGO - 15° POSMEC. FEMEC/UFU, Uberlândia-MG, 2005. *Estratégias De Evolução Diferencial Aplicadas a Problemas de Otimização Restritos.*

RIDLEY, M. *Evolution*. 2. ed. Cambridge, MA: Blackwell Science, Inc., 1996.

RODRIGO C. P. SILVA – *Tcc* (2010). *Um estudo sobre a auto-adaptação de parâmetros na evolução diferencial.*

STORN, R.; PRICE, K. *Differential Evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces*. 1995.

STORN, R.; PRICE, K., 1997, *"Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", Journal of Global Optimization*, n. 11, pp. 341-359.

VINICIUS K. ALMEIDA - *Tcc* (2013). *Análise comparativa dos tratamentos de limites das variáveis e da adaptação de parâmetros na evolução diferencial.*

ZAHARIE, D. Critical values for the control parameters of differential evolution algorithm. In: MATOUEK, R.; OMERA, P. (Ed.). *Proceedings of MENDEL 2002, 8th International Mendel Conference on Soft Computing, Brno, Czech Republic*. [S.l.: s.n.], 2002.

Análise da expressão de genes codificados nos plasmídeos de Klebsiella pneumoniae subsp. pneumoniae Kp13 conferindo resistência cruzada a antibióticos em resposta à elevadas concentrações de colistina B.

Thiago Cardoso Pereira Carneiro[1],Marisa Fabiana Nicolás.[2]

1-Universidade Estácio de Sá(thgcardoso40@gmail.com)

2-Laboratório Nacional de Computação Científica.(marisa@lncc.br)

Resumo

Se tem conhecimento da resistência a agentes químicos e físicos por bactérias desde o inicio da era microbiana. Pode-se ressaltar ainda o uso clínico em 1941 da penicilina onde foi revelado que a resistência de bactérias a antimicrobianos poderia ser tanto uma característica natural da espécie quanto adquirida por cepas individuais.[1]

Essa resistência é conferida através de genes de resistência, que quando não estão presentes originalmente no DNA das bactérias, podem ser importados através de resistência transferível (transdução,conjugação e genes contidos em plasmídeos e transposons). Esses genes por sua vez contém a informação para expressão de mecanismos bioquímicos que impedem a ação antimicrobiana, como exemplo temos: i) a produção de enzimas que hidrolisam componentes do antibiótico, ii) produção de enzimas que modificam seus componentes, iii) mudanças nos receptores da membrana e iv) mudanças em sua carga.[1]

Tendo em vista que hoje a polimixina B (ou colistina B) é cada vez mais utilizadas como linha ultima de tratamento contra Bactérias Gram-negativas multirresistentes (incluindo a espécie *Klebsiella pneumoniae*), e que as mesmas utilizam mecanismos moleculares complexos para se protegerem da ação antimicrobiana, é importante entender a resposta desta resistência bacteriana. Assim, o presente estudo tem como objetivo analisar o perfil dos genes codificados nos plasmídeos de *Klebsiella pneumoniae* subsp. *pneumoniae* Kp13 (KP13) que foram expressos em altas concentrações de colistina B afim de compararar a expressão dos genes de resistência nas condições de estudo [2]

Os dados analisados de KP13 foram gerados em sequenciamento de nova geracao (NGS) através de plataforma illumina Hiseq (www.illumina.com), em condição induzida e não induzida. A analise do transcriptoma foi feita através de ferramentas de bioinformatica. Os dados recém gerados precisaram ser trimados e para isso foi usado o programa skewer [3], os dados dos plasmideos foram mapeados contra os dados sequenciados nas condições escolhidas usando o programa

Tophat (http://ccb.jhu.edu/software/tophat/index.shtml). Após o mapeamento, os dados foram ordenados usando o programa samtools (www.samtools.sourceforge.net), com os dados ordenados foi feita a contagem dos reads mapeados usando arquivos gtf com o programa Htseq-count(http://www-huber.embl.de/users/anders/HTSeq/doc/count.html). Em uma etapa complementar, os arquivos FASTA dos plasmideos foram submetidos contra bancos de dados de resistência afim de se encontrar genes de resistência conhecidos. O uso do programa MEME(www.meme-suite.org) foi aplicado para encontrar motivos regulatórios para alguns genes. Para a analise de expressão diferencial foram usados os pacotes do programa R conforme protocolo [4]. Finalizando foram criados diagramas de venn contendo os genes up e down regulados entre as condições de expressão avaliadas.

Palavras Chave: Bioinformática, Klebsiella pneumoniae, KP13, colistina B.

Referencias:

1 - TAVARES, Walter. Bactérias gram-positivas problemas: resistência do estafilococo, do enterococo e do pneumococo aos antimicrobianos. Rev. Soc. Bras. Med. Trop. [online]. 2000, vol.33, n.3 [cited 2015-09-15], pp. 281-301 . Available from: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0037-86822000000300008&lng=en&nrm=iso>. ISSN 1678-9849. http://dx.doi.org/10.1590/S0037-86822000000300008.

2- Meredith S. Wright,a Yo Suzuki,a Marcus B. Jones,a Steven H. Marshall,b Susan D. Rudin,b David van Duin,c Keith Kaye,d Michael R. Jacobs,e Robert A. Bonomo,b,f Mark D. Adamsa ,Genomic and Transcriptomic Analyses of Colistin-Resistant Clinical Isolates of Klebsiella pneumoniae Reveal Multiple Pathways of Resistance .

3- Jiang H1,Lei R, Ding SW, Zhu S. Skewer: a fast and accurate adapter trimmer for next-generation sequencing paired-end reads. BMC Bioinformatics. 2014. 12;15:182.

4- Count-based differential expression analysis of RNA sequencing data using R and Bioconductor Simon Anders1, Davis J McCarthy2,3, Yunshun CChen4,5, Michal Okoniewski6, Gordon K Smyth4,7, Wolfgang Huber1 & Mark D Robinson8,9.

Estudo, teste e implementação de Bibliotecas para visualização 3D

Laboratório Nacional de Computação Científica (LNCC)

Alexsandro de Paula Miranda[1], Jauvane C. de Oliveira[2]

Palavras Chave: CAVE, Realidade Virtual, Simulação, Ambientes virtuais.

1. Introdução

Devido ao avanço da tecnologia hoje é possível, por exemplo, ampliar a produção e reduzir os custos nas indústrias, aumentar a produtividade de funcionários, e elaborar sofisticados equipamentos para a medicina. A computação tem agregado uma evolução significativa na vida do homem moderno. Entre tantas tecnologias que surgiram com o passar dos anos a Realidade Virtual, um conceito que surgiu na década de 60 com Morton Heilig [1], chegou em áreas comerciais como uma ferramenta de auxílio para treinamentos e simulações. O objetivo deste trabalho é apresentar o desenvolvimento de uma aplicação que permite usar a infraestrutura de um ambiente imersivo de custo relativamente baixo, conhecido como ambiente CAVE. Este ambiente é construído com a finalidade de possibilitar a interação do usuário com o cenário em tempo de execução, e também auxiliar na redução de custos com treinamentos, e isentar os funcionários de possíveis riscos que poderiam ocorrer em ambientes reais. Para o desenvolvimento desta aplicação foi necessário o uso da IDE Worldviz Vizard que já vem com o seu próprio compilador Python, e do ambiente CAVE para a visualização da aplicação.

2. Ambiente CAVE do LNCC

O Laboratório Nacional de Computação Científica (LNCC) que se encontra na cidade de Petrópolis, no estado do Rio de Janeiro, possui um ambiente CAVE em uma de suas instalações. Atualmente a ferramenta de desenvolvimento do sistema de controle para o ambiente CAVE é o Instant Reality, arquitetura que combina vários recursos, fornecendo uma interface para desenvolvedores de aplicações de Realidade Virtual. O objetivo desta pesquisa é encontrar bibliotecas de visualização 3D que possam servir como uma segunda via para o ambiente CAVE que só possui o Instant Reality como sistema de controle.

Projetado pelo engenheiro François Maltric e construído pela equipe do laboratório ACiMA, o sistema CAVE do LNCC é um projeto de baixo custo, porém de alta qualidade. De acordo com o Coordenador do CCC do LNCC, Jauvane de Oliveira, o custo inicial esperado para a construção do ambiente CAVE era de 2 milhões de reais, mas o orçamento final do projeto foi de um pouco mais de 200 mil reais. Os componentes principais da estrutura do ambiente CAVE do LNCC estão representados na figura 2.1. Toda a estrutura foi projetada com PVC, projetores DLP's

comuns, computadores normais com bom desempenho de vídeo, materiais encontrados facilmente em qualquer metrópole. A opção de mudar uma estrutura de alumínio para uma de PVC é um exemplo de como foi possível construir o sistema de visualização automático com 10 por cento do valor médio de uma construção de um sistema deste tipo. Os componentes numerados de 1 a 8 são referentes aos projetores, e os numerados de 9 a 12 são referentes aos espelhos.

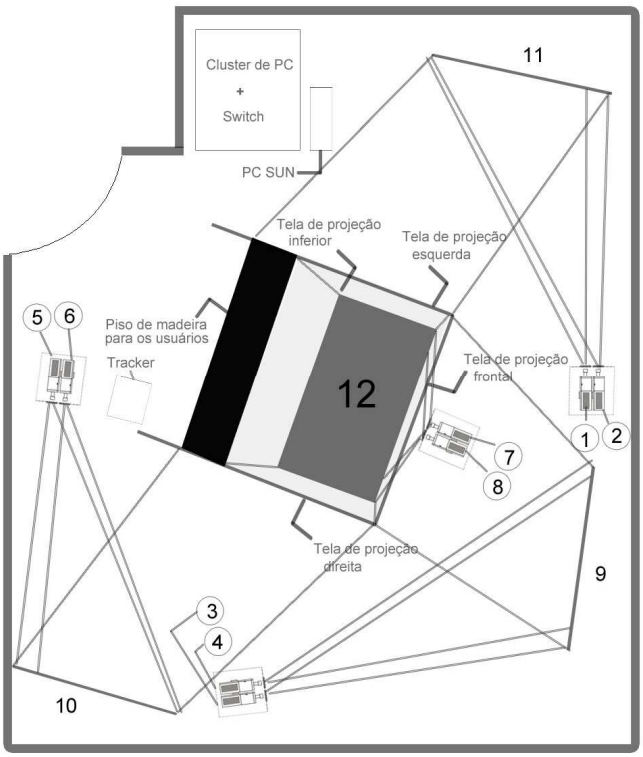


Fig 2.1: Planta baixa do ambiente CAVE do LNCC

3.1 Escolhendo, testando e implementando

Nas primeiras implementações realizadas utilizou-se o OGRE 3D (Object-oriented Graphics Rendering Engine), um motor gráfico open source escrito em C++ que permite a manipulação de objetos 3D. Inicialmente havia uma motivação para o uso do Ogre, devido a compatibilidade com bibliotecas de simulação de física que é um recurso desejável para uso em aplicações do ambiente CAVE. No entanto há um grau mais elevado na curva de aprendizado e há pouco material bibliográfico disponível, o que é um aspecto negativo para o uso desta ferramenta. Começou-se então uma nova implementação utilizando a arquitetura Vizard que também é um motor gráfico escrito em Python, não é open source, e possui uma imensa biblioteca disponível a fim de facilitar a implementação. Essas bibliotecas influenciaram na evolução dos projetos que outrora eram mais complexos de se desenvolver.

4. Os resultados

Sabe-se que as lâmpadas dos projetores do ambiente CAVE possuem tempo de vida útil. Para evitar um consumo desnecessário dessas lâmpadas utilizou-se dois monitores de 20 polegadas para iniciar as implementações, pois assim haveria um melhor aproveitamento das telas quando todas as viewports fossem geradas, simulando as paredes do CAVE. Com o ambiente computacional preparado devidamente, distribuiu-se todas as viewports pela tela com tamanho equivalente as paredes do CAVE, assim ao criar uma viewport com X de altura e Y de largura na parede do CAVE, no monitor o seu tamanho seria devidamente proporcional e vice-versa. Configurou-se também a manipulação dessas viewports pelo teclado para que fosse possível obter o tamanho exato em tempo real, assim elas não ultrapassariam as arestas do ambiente CAVE. Feito isso, foi necessário alinhar o objeto dentro do ambiente CAVE, de modo que, quando um objeto fosse movido ele passaria de uma viewport para outra, dando uma sensação de continuidade.

5. Conclusão

Hoje, com o auxílio do software Vizard que tornou possível utilizar o ambiente virtual automático CAVE do LNCC, é possível que novos desenvolvedores implementem suas aplicações para o ambiente CAVE sem a necessidade do conhecimento específico que ele exige. A partir do desenvolvimento apresentado neste trabalho é possível manipular projeções e compilar qualquer tipo de aplicação imersiva no ambiente CAVE.

Análise das vias metabólicas dos genes plasmidiais de *Klebsiella pneumoniae* subsp. *pneumoniae* Kp13 em resposta à polimixina B, a partir de dados de RNA-seq

Laboratório Nacional de Computação Científica (LNCC)

Gisele Lucchetti da Silva[1], Marisa Fabiana Nicolás[2]

1 Universidade Estácio de Sá (giselelucchetti@gmail.com)
2 Laboratório Nacional de Computação Científica LNCC (marisa@lncc.br)

Resumo

O aumento da resistência aos antibióticos nas bactérias Gram-negativas, em particular *Klebsiella pneumoniae* é um desafio para a medicina global. A polimixina B (ou colistina B) estão sendo cada vez mais utilizadas como opções terapêuticas de último recurso para tratamento de infecções causadas por bactérias MDR (do inglês: *Multidrug Resistance*). As taxas de resistência são alarmantes para as polimixinas que tem sido relatado entre a KPC (*Klebsiella pneumoniae* Carbapenemase).[1,2]

A resistência aos antibióticos acontece através de diferentes mecanismos e representa uma resposta desenvolvida pela população bacteriana a uma pressão seletiva imposta pela ação da droga. Com isso, os indivíduos que conseguem sobreviver ao contato com a droga possuem mais chances de ser selecionado, e assim dentro de um contexto evolutivo se desenvolvem em populações resistentes ao antibiótico. Estudos de procura de novos alvos moleculares para novas drogas são necessários para o controle de patógenos que representam ameaças a saúde humana. Bactérias da espécie *Klebsiella pneumoniae* são uma das grandes responsáveis de surtos hospitalares em todo o mundo, uma vez que podem se tornar resistentes a maioria dos antibióticos.[3]

O objetivo principal do trabalho é analisar o conjunto completo de transcritos codificados nos plasmideos do genoma da *K. pneumoniae* subsp. *pneumoniae* Kp13 (KP13) (RAMOS ,2012 e RAMOS e col,2012) a partir de dados de RNA-seq utilizando como condições duas bibliotecas de cDNA (não induzido e induzido com polimixina B).

Para o sequenciamento foi utilizado à plataforma de Illumina Hiseq, para que a partir do resultado obtido pode-se se gerar resultados de expressão gênica.

Como primeira etapa foi realizado o pré-processamento dos dados de RNA-Seq utilizando o programa Skewer[4]. Na segunda etapa foi gerado o alinhamento das leituras do RNA-seq contra o genoma da KP13 usando o programa TopHat [5], para se obter a quantificação da expressão dos genes para a determinação da expressão diferencial. Para este último, foi utilizado a plataforma R (https://www.r-project.org/)e o pacote edgeR [6], onde observa-se quais genes foram mas expressos nas condições estudadas. Com esta abordagem foi possível obter os dados para as posteriores análises, através da ferramenta de análises de Bioinformática como *Kegg pathway reconstruction* (http://www.genome.jp/kegg/pathway.html). Com isto foi possível realizar a análise das vias metabólicas dos genes expressos

Palavras chaves: *Klebsiella pneumoniae*, KP13, polimixina B, RNA-seq

Referencias

1 - Ramos PI., Picão RC, Vespero EC, Pelisson M, Zuleta LFG, Almeida LGP, Gerber AL, Vasconcelos ATR, Gales AC, Nicolás MF: Pyrosequencing-based analysis reveals a novel capsular gene cluster in a KPC-producing Klebsiella pneumoniae clinical isolate identified in Brazil. *BMC Microbiology* 2012, 12:173.

2 - Ramos PI.: **Análise genônica comparativa e reconstrução metabolica de Klebsiella pneumoniae Kp13**. Laboratorio Nacional de Computação Científica; 2012:164f. Dissertação (Mestrado) - Laboratório Nacional de Computação Científica.

3 - Davies J, Davies D: Origins and evolution of antibiotic resistance. *Microbiology and Molecular Biology Reviews* 2010, 74:417–33.

4 - Jiang H[1], Lei R, Ding SW, Zhu S.: Skewer: a fast and accurate adapter trimmer for next-generation sequencing paired-end reads. BMC Bioinformatics. 2014 Jun 12;15:182. doi: 10.1186/1471-2105-15-182.

5 - Daehwan Kim, Geo Pertea, Cole Trapnell, Harold Pimentel, Ryan Kelley, and Steven L Salzberg: TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. Journal List Genome Biol v.14(4); 2013 PMC4053844

6 - Davis J McCarthy, Yunshun Chen, Michal Okoniewski, Gordon K Smyth, Wolfgang Huber & Mark D Robinson: Count-based differential expression analysis of RNA sequencing data using R and Bioconductor

Comparação de desempenho de rotinas de multiplicação de matrizes densas em arquiteturas multi-core e many-core

Mateus Silva de Melo (Bolsa de Iniciação Tecnológica – PIBITI/LNCC), Roberto Pinto Souto (Coordenação de Sistemas e Redes – CSR/LNCC), Ícaro Fontes Moreira de Castro (Programa de Estágio - LNCC)

Uma parcela significativa de aplicações científicas fazem uso da álgebra linear computacional. Por este motivo, existe um grande esforço no desenvolvimento de bibliotecas especializadas em álgebra linear, implementadas especialmente para arquiteturas paralelas.

Este trabalho trata-se de um estudo comparativo de desempenho dessas bibliotecas, utilizando três arquiteturas paralelas: CPU e MIC, utilizando a rotina SGEMM da biblioteca Intel MKL, e também a GPU, utilizando a rotina cuBLAS SGEMM da biblioteca NVIDIA cuBLAS.

Foram utilizadas matrizes densas quadradas de tamanho 4096, 8192, 16384 e 20480, com rodadas de 1, 2, 4, 8, 16 threads na CPU, e de 8, 16, 32, 64, 128 e 240 threads na MIC. Na GPU as configurações de execução são determinadas internamente pela rotina cuBLAS SGEMM.

Através de algumas métricas de avaliação, tais como ganho de desempenho (*speed-up)*, eficiência paralela, e desempenho sustentado, foi observado que a rotina SGEMM da Intel MKL possui uma boa escalabilidade nas arquiteturas CPU e MIC. Porém a biblioteca NVIDIA cuBLAS, sendo executada na GPU, foi a que apresentou o melhor desempenho.

Estudo de uma versão paralela em arquitetura manycore do processo de classificação de metagenomas

Micaella Coelho, Carla Osthoff e Fabrício Vilasbôas
Laboratório Nacional de Computação Científica

O desenvolvimento tecnológico na área de bioinformática tem gerado uma grande quantidade de dados a serem processados, porém por outro lado o advento de tecnologias tais como GPU têm possibilitado um processamento cada vez maior dos dados.

Nesse projeto temos como objetivo apresentar técnicas de otimização utilizando uma arquitetura manycore em GPU para a classificação de metagenomas. O algoritmo utilizado é o K-mer que se dá pela contabilização da frequência de repetição de todas as possíveis combinações de k nucleotídeos em uma sequência. Este estudo, apresenta a primeira parte da otimização que consiste na implementação do K-mer em GPU, onde demostramos que possui uma capacidade de processamento de dados muito superior à implementação em CPU.

Os dados utilizados para os testes são dados reais provenientes do sequenciamento de uma amostra da microbiota intestinal humana. O sequenciamento foi feito utilizando o equipamento da nova geração de sequenciadores, o Illumina, que gera uma grande quantidade de dados com alta fidelidade. Equipamentos como este necessitam de novas tecnologias, com capacidade de processamento de dados muito superior aos utilizados para o processamento dos dados gerados pelos sequenciadores da geração anterior.

Modelagem Geométrica e Otimização de objetos 3D para ambientes virtuais colaborativos

Laboratório Nacional de Computação Científica (LNCC)

Stephane Deserto Vasconcelos[1], Jauvane C. de Oliveira[2]

Palavras Chave: 3D, Otimização.

A implementação de sistemas de Realidade Virtual Colaborativa somente é possível devido a existência de objetos tridimensionais, estes são representados a partir da combinação de polígonos, juntamente com texturas, luzes, etc., a fim de criar representações de uma infinidade de objetos para compor o mundo virtual da forma mais realista possível.

Para a utilização de modelos 3D de alta resolução, é necessário a aquisição de hardwares poderosos e caros, o que acaba por limitar a quantidade de usuários que poderão usufruir dos mesmos. Visando um maior alcance de usuários, o laboratório ACiMA, desenvolve soluções alternativas que utilizam alto desempenho, porém, com custos relativamente baixos.

É importante lembrar que só é possível utilizar os modelos nos sistemas atuais de baixo custo, depois que eles passam por um processo de otimização, onde, a otimização da malha, em outras palavras, consiste em diminuir a quantidade de polígonos dos objetos em questão sem que haja grandes alterações em seu visual, tornando assim os modelos geométricos mais "leves" e computacionalmente tratáveis.

Nas figuras 1 e 2, é possível observar a diferença entre um fígado não otimizado e um fígado otimizado.

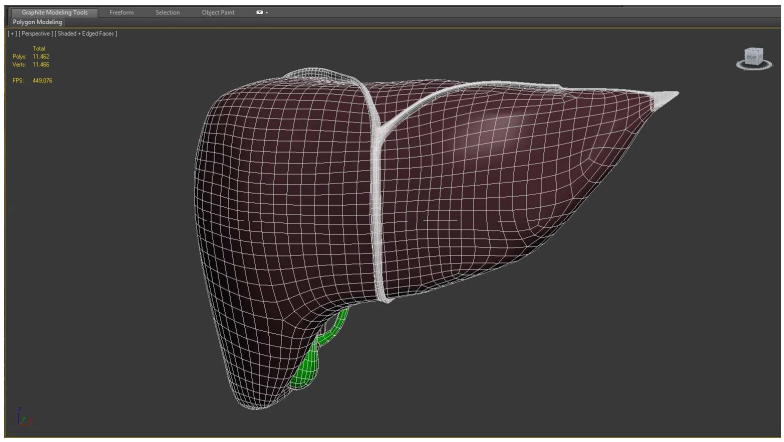


Figura 1: Fígado não otimizado com 11.462 polígonos.

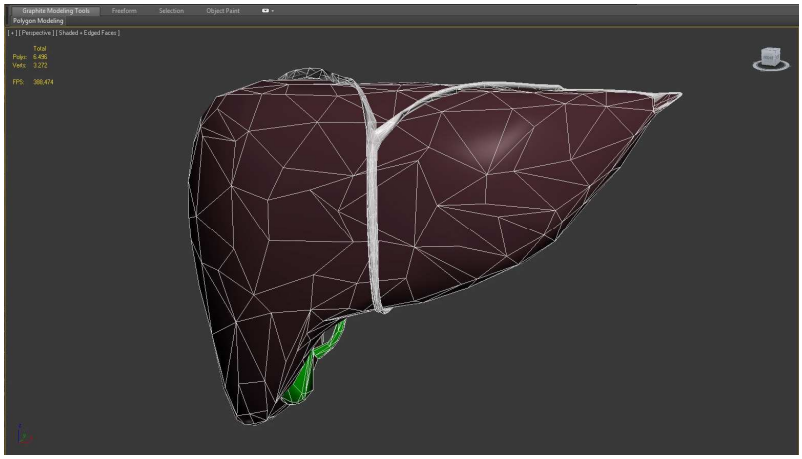


Figura 2: Fígado otimizado com 6.496 polígonos.

Para a criação dos modelos 3D são utilizados softwares específicos para modelagem 3D, como 3DS Max e o Maya.

Objetos mais complexos, como partes do corpo humano, tendem a ser criados com uma quantidade alta de detalhes, e posteriormente são otimizados, diminuindo os polígonos em locais específicos e, portanto, dosando os pontos que necessitam ser expostos com uma maior minúcia. Já objetos mais simples e menos importantes na cena, são criados com poucos polígonos desde o início, dispensando a etapa de otimização.

Após a criação do modelo, é iniciada a etapa de coloração e texturização, nela é dado cor ou textura aos modelos criados, deixando-os mais realistas. Para a aplicação de textura de forma coerente, o modelo passa por dois processos o *UV unwrapping* e o *UV mapping*, onde o *UV unwrapping* é o desembrulhar de um objeto 3D, tornando possível a criação de sua textura para a aplicação do *UV mapping*. Já o *UV mapping* é o processo de representação de uma imagem 3D através de uma imagem 2D, deste modo, em contraste com o X, Y e Z, que são coordenadas originais do objeto tridimensional no ambiente de modelagem, U e V são as coordenadas do objeto já transformado. A imagem é então retransformada (*wrapped*) sobre a superfície do objeto 3D.

Na figura 3, podem ser observados os processos de UV unwrapping e UV mapping.

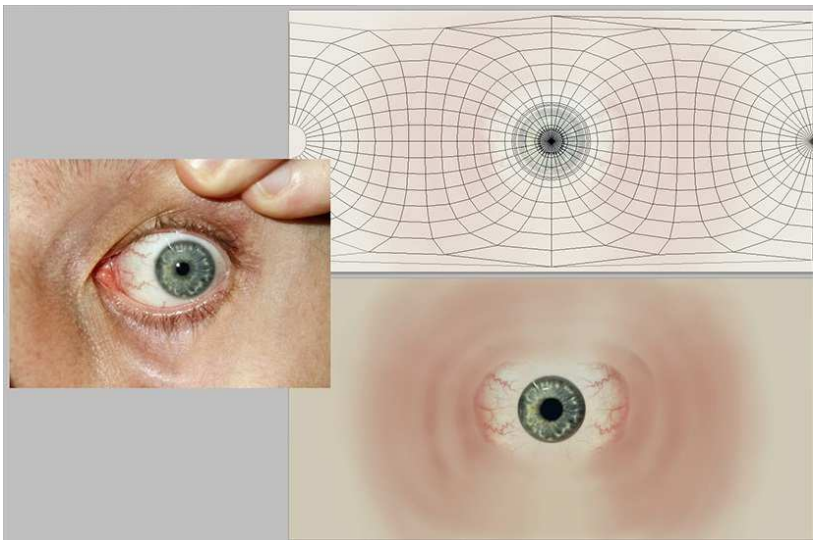


Figura 3: Representação dos processos *UV unwrapping* e *UV mapping*.

Na figura 4, pode ser observado o resultado final após a etapa de texturização e os processos de UV *unwrapping* e *UV mapping*.

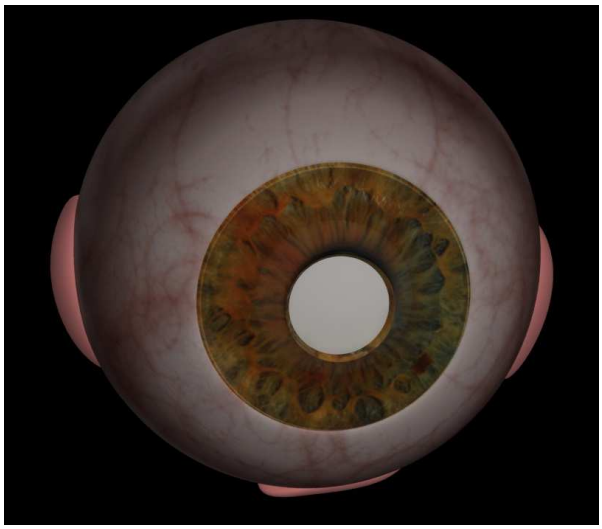


Figura 4: Olho finalizado, após modelagem e texturização.

Ao final desta etapa os modelos são exportados, em extensões predeterminadas pelos programadores, e utilizados para ilustrar e dar um maior realismo às aplicações criadas.