CNPq – Conselho Nacional de Desenvolvimento Científico

Fortalecimento das Áreas de Computação de Alto Desempenho e Inteligência Computacional do Programa de Pós-Graduação em Informática da UFES

Linhas de Pesquisa em Computação de Alto Desempenho e Inteligência Computacional do Programa de Pós-Graduação em Informática da UFES

1. Introdução

A presente proposta tem como objetivo central fortalecer e incrementar as interações entre as linhas de pesquisa de Inteligência Computacional e Computação de Alto Desempenho do Programa de Pós-graduação em Informática da UFES, contando para isso com o apoio de grupos de pesquisa de programas de pós-graduação já consolidados da COPPE/UFRJ e da USP/São Carlos. O grupo de pesquisadores da COPPE/UFRJ irá apoiar e interagir fundamentalmente com os pesquisadores da linha de Computação de Alto Desempenho do programa não consolidado, enquanto que o grupo de pesquisa da USP/São Carlos irá interagir e apoiar os pesquisadores da linha de Inteligência Computacional.

Na linha de Computação de Alto Desempenho serão realizados trabalhos conjuntos entre a equipe da UFES e da COPPE nos seguintes temas: (i) estudo e implementação paralela de novas técnicas adaptativas no tempo e no espaço de representação de configurações de malha de elementos finitos aplicados a problemas de escoamentos em meios porosos; (ii) novos paradigmas de programação adequados para futuros sistemas paralelos com dezenas ou centenas de processadores implementados através de circuitos integrados *multi-core*. Na linha de Inteligência Computacional serão realizados trabalhos na área de descoberta de conhecimento em bases de dados, envolvendo técnicas de mineração de dados, reconhecimento de padrões e aprendizado automático. Especificamente, serão realizados estudos nos seguintes temas: (i) seleção de atributos; (ii) adaptações das técnicas clássicas de classificação ao caso de dados com variáveis temporais; (iii) integração de classificadores simbólicos utilizando sistemas multi-agentes.

O elemento integrador dos trabalhos de pesquisa a serem realizados nesse projeto será o estudo e o desenvolvimento de técnicas, algoritmos, metodologias, hardware e software para aplicação da computação paralela na etapa de aprendizado dos processos de mineração de dados, visando, com isso, tornar esses processos mais eficazes. A etapa de aprendizado dos processos de mineração de dados demanda um esforço computacional intenso em virtude da necessidade de se avaliar inúmeros modelos de aprendizado ao longo de um processo que requer a manipulação e análise de uma grande quantidade de dados. Para se ter uma idéia, considere que um único modelo de rede neural (um dos métodos mais usados para aprendizado) pode levar semanas para ser construído e treinado. Em muitos casos, é preciso avaliar inúmeras variações de modelos de redes neurais de modo a encontrar aquele que melhor se ajuste à aplicação. Adicionalmente, cada um desses modelos precisa ser verificado em bases de testes. Muitas vezes, o tempo excessivo requerido para a realização de todas essas tarefas acaba por impedir uma busca mais efetiva pelo modelo mais apropriado, impactando na qualidade do aprendizado. Possibilitar que esse processo seja realizado paralelamente por várias máquinas em um cluster certamente pode torná-lo mais eficaz. Contudo, o trabalho de paralelização pode requerer diferentes abordagens para diferentes modelos de aprendizado. Assim, neste projeto será investigado como esses modelos de aprendizado podem ser paralelizados e serão avaliadas quais as abordagens mais apropriadas para serem aplicadas a cada modelo.

2. Programas de Pós-Graduação Participantes

O presente projeto envolve, no programa não consolidado (Programa de Pós-Graduação em Informática da UFES), pesquisadores associados às Linhas de Pesquisa em Computação de Alto Desempenho e Inteligência Computacional. Pesquisadores de dois programas de pós-graduação consolidados colaborarão para o desenvolvimento do projeto: da Área Interdisciplinar de Computação de Alto Desempenho, formada pelos Programas de Pós-Graduação em Engenharia Civil, Sistemas e Computação e Elétrica da COPPE/UFRJ; e do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC) do ICMC/USP. A seguir, apresentamos breve histórico dos programas, considerando os critérios de avaliação e conceito da CAPES. Para cada um dos programas também é apresentada uma listagem das áreas de concentração e linhas de pesquisa, projetos em andamento e trabalhos científicos, tecnológicos e de inovação já desenvolvidos pelos grupos, evidenciando suas experiências prévias e

potencialidades na área/tema do projeto. Ao fim desta seção, apresentamos, planilha contendo informações sobre os pesquisadores dos grupos de pesquisa participantes.

Programa de Pós-Graduação em Informática da UFES (PPGI/UFES)

O Programa de Pós-Graduação em Informática (PPGI) da Universidade Federal do Espírito Santo (UFES), através do seu curso de Mestrado em Informática (MI) foi criado em 1994 com os seguintes objetivos:

- Formar, no Estado do Espírito Santo, professores e pesquisadores para atuarem nas instituições de ensino e de pesquisa;
- Contribuir no desenvolvimento científico e tecnológico na área de Informática do estado;
- Oferecer formação tecnológica altamente especializada para empresas (e seus funcionários) que atuam no estado, como a Companhia Vale do Rio Doce (CVRD), a Companhia Siderúrgica de Tubarão (CST), a Aracruz Celulose, a Empresa de Energia Elétrica do Espírito Santo (ESCELSA), a Samarco Mineração, dentre outras, como também para as instituições públicas localizadas no Estado;
- Agregar o corpo de pesquisadores em Ciência da Computação da UFES para desenvolvimento de pesquisa em diversas sub-áreas da Ciência da Computação.

Nos seus sete primeiros anos, ou seja, até o ano de 2000, o corpo docente do PPGI era formado por cerca de 12 professores. O corpo discente era relativamente reduzido, tendo sido formados 34 (trinta e quatro) mestres até dezembro de 2000. Graças a um esforço do Departamento de Informática no final dos anos 90 para a capacitação em doutoramento do seu corpo docente, a partir de 2001 o PPGI cresceu significativamente, tendo destaque os seguintes itens:

- Ampliação do corpo docente para 23 professores doutores contratados em regime de dedicação exclusiva;
- Ampliação da oferta de vagas, sendo 20 alunos em 2001, 36 alunos em 2002, 17 em 2003, 40 em 2004 e 25 em 2005 (vários professores se afastaram para pós-doutorado em 2005).

Como conseqüência, foram defendidas 21 dissertações em 2001, 13 em 2002, 21 em 2003, 31 em 2004, e 21 em 2005, elevando o total de dissertações defendidas desde sua criação até hoje (2006) para 145.

Muito embora o PPGI/MI tenha iniciado em 1994, efetivamente nos últimos cinco anos é que o mesmo se estruturou como um programa de pós-graduação, passando a contar com um corpo docente (formado em sua maioria por recém-doutores) com formação em diversas sub-áreas da Ciência da Computação, permitindo assim melhor atender às demandas existentes. Vale destacar que a área de influência do programa também cresceu: O programa tem recebido regularmente alunos provenientes do leste de Minas Gerais, norte do Rio de Janeiro, sul da Bahia e região Norte do Brasil. Cabe ainda ressaltar que o Mestrado em Informática do PPGI é o único curso de mestrado em computação do Estado.

Nos últimos anos, várias parcerias têm sido estabelecidas com empresas locais e com os fundos setoriais, tais como: ESCELSA, CST, CT-INFRA, ANEEL, ANP dentre outros, destinando bolsas de graduação e de pesquisador, além de recursos para equipamentos/laboratórios. O Departamento de Informática (DI) e o PPGI vêm recebendo recursos para infra-estrutura física de pesquisa e ensino de graduação e pós-graduação provenientes de um convênio UFES-Petrobras. No biênio 2003/2004 o total de recursos repassados foi de R$30 mil para o PPGI, R$80 mil para o DI e R$400 mil para construção de um novo prédio de salas de aula/laboratórios para o DI/PPGI. Para o biênio 2005/2006, o total de recursos previstos é de R$112 mil para o PPGI e R$340 mil para o DI. Especificamente no caso dos recursos destinados ao PPGI, estes têm sido prioritariamente alocados na infra-estrutura dos laboratórios de pesquisa.

Os recursos obtidos através de parcerias e a inserção dos novos doutores no programa têm contribuído para a consolidação dos diferentes grupos temáticos de pesquisa agregados ao programa e para o desejo de criação de um curso de Doutorado do PPGI. Com este objetivo, em 2003 o programa passou por uma reforma curricular e definiu uma única Área de Concentração: Ciência da Computação. Para agrupar professores experientes e recém-doutores considerando suas formações específicas, definiu-se as seguintes linhas de pesquisa: Computação de Alto

Desempenho, Inteligência Computacional, Otimização e Sistemas de Informação. Esta divisão por linhas de pesquisa possibilitou uma melhor estruturação do PPGI nos seguintes aspectos:

- Melhor integração do seu corpo docente com as linhas de pesquisa;
- Reorganização das disciplinas do MI, com divisão por área e por tipo (regulares, eventuais e tópicos especiais), e com uma melhor e constante oferta anual;
- Surgimento de vários projetos de pesquisa, com maiores vínculos entre projetos/produção intelectual e com trabalhos de conclusão;
- Reformulação e investimentos em laboratórios temáticos de pesquisa.

Nos anos 2003/2004, os principais esforços do PPGI para seu fortalecimento foram:

- A promoção de uma reforma curricular, visando principalmente a integração de seus de docentes em linhas de pesquisa;
- A diminuição do tempo médio de formação, através de um melhor acompanhamento dos alunos;
- O estabelecimento de um procedimento interno (credenciamento) para a entrada e permanência dos professores no quadro docente do programa (permanentes e colaboradores), valorizando e incentivando as publicações qualificadas.

Com todos os esforços citados, espera-se que o PPGI no atual triênio de avaliação (2004-2006) tenha um melhor desempenho, avançando na direção de seu objetivo de criar um curso de doutorado. Contudo, no que se refere à produção bibliográfica, três componentes ainda comprometem seus resultados:

- O reduzido número de bolsas concedidas para o mestrado (apenas sete da CAPES e uma do CNPq em 2004);
- A inexistência de um curso de doutorado; e
- A ainda pequena cooperação entre os docentes das diferentes linhas de pesquisa e do programa com outros programas de pós-graduação consolidados.

Certamente, com mais bolsas disponíveis o programa teria mais alunos em dedicação exclusiva; com o curso de doutorado, mais prazo para um maior aprofundamento das pesquisas desenvolvidas; e com maior cooperação, o preenchimento de lacunas específicas nas atuais áreas de expertise e a potencialização das capacidades do corpo docente. Este projeto busca ampliar a cooperação entre as linhas de pesquisa em Computação de Alto Desempenho e Inteligência Computacional, e a interação do PPGI com outros programas consolidados, de modo a ampliar a produção científica qualificada do PPGI e fortalecer o programa. O PPGI possui hoje conceito 3 na CAPES.

Programa de Pós-Graduação em Engenharia Civil da COPPE/UFRJ, área interdisciplinar de Computação de Alto Desempenho

A COPPE/UFRJ – Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia da Universidade Federal do Rio de Janeiro – iniciou suas atividades em 1963, passando em 1965 a ser Órgão Suplementar da UFRJ, com a finalidade de promover e coordenar os cursos de pós-graduação strictu sensu, nos níveis de mestrado e doutorado nas diversas áreas da engenharia. A COPPE conta hoje com cerca de 300 professores, todos doutores em regime de dedicação exclusiva, mais 170 técnicos de nível superior e um corpo discente de 3000 alunos de mestrado e doutorado.

O padrão de excelência da COPPE se reflete na produção acadêmica da Instituição. Anualmente são defendidos cerca de 150 teses de doutorado e 300 dissertações de mestrado, índices comparáveis aos principais centros de ensino e pesquisa do mundo. A produção científica da COPPE alcança uma média anual de 1400 artigos em congressos e revistas nacionais e internacionais.

A organização dos estudos em nível de pós-graduação na COPPE se dá, institucionalmente e regimentalmente, através de diversos programas concentram suas atividades nas áreas de Engenharia Civil, Química, Mecânica, Elétrica, Metalúrgica e de Materiais, Produção, Oceânica, Nuclear, Biomédica, Sistemas e Computação, Transportes e Planejamento Energético. Em anos recentes, em vários ramos do conhecimento científico e tecnológico, ficou evidenciada a necessidade de uma maior interação entre as várias áreas da Engenharia, caracterizada pela complementaridade e interdisciplinaridade.

O Núcleo de Computação Paralela, criado em 1988, atualmente denominado Núcleo de Atendimento em Computação de Alto Desempenho (NACAD) da COPPE representa uma iniciativa pioneira no Brasil na área de processamento de alto desempenho e vem contribuindo substancialmente para o desenvolvimento da informática nacional, nas áreas de sistemas computacionais para engenharia, metodologias de programação, linguagens e compiladores, algoritmos, sistemas operacionais e arquitetura de computadores. O curso de Mestrado e Doutorado na Área Interdisciplinar de Computação de Alto Desempenho, formado pelos Programas de Pós-Graduação em Engenharia Civil (nível 7-CAPES), Programa de Engenharia de Sistemas e Computação (nível 6-CAPES) e pelo Programa de Engenharia Elétrica (nível 7-CAPES), objetiva oferecer uma visão ampla e atualizada das bases e formas de desenvolvimento tecnológico na área de Computação Científica. Do ponto de vista acadêmico/institucional, a criação deste curso consolidou e ampliou as atividades do Núcleo de Computação de Alto Desempenho, facilitando e incrementando as articulações com os diversos Programas da COPPE atuantes nesta área.

Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional do ICMC/USP (PPG-CCMC/USP)

Mesmo antes da criação do Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC/USP), São Carlos já tinha um bem conceituado Programa de Pós-Graduação em Matemática, com cursos de Mestrado e Doutorado, patrocinado pelo Departamento de Matemática da Escola de Engenharia de São Carlos - USP.

Esse departamento foi o embrião do ICMC-USP, fundado em 1971. A Pós-Graduação do ICMC-USP existe oficialmente desde essa data. Na poca, já existia o Programa de Matemática com os cursos de Mestrado e Doutorado. Em 1983 foi criado o Programa de Ciências de Computação, aproveitando os recursos humanos oriundos da extinta Pós-Graduação Interunidades que envolvia os Campi de São Carlos e de Ribeirão Preto da USP, existente desde 1974 para o curso de mestrado. Em 1995 foi iniciado o Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional com o curso de Doutorado. Assim, atualmente o Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC) do ICMC/USP é responsável por um curso de Mestrado e um curso de Doutorado, os quais são desenvolvidos dentro das seguintes linhas de pesquisa: Banco de Dados; Computação Bioinspirada; Computação Gráfica e Processamento de Imagens; Engenharia de Software e Sistemas de Informação; Hipermídia; Inteligência Computacional; Mecânica dos Fluidos Computacional; Otimização e Modelos Estocásticos e Sistemas Distribuídos e Programação Concorrente.

Até maio de 2006, o programa formou 645 Mestres e 69 Doutores. Atualmente o programa conta com 130 alunos de Mestrado e 87 alunos de Doutorado matriculados. O programa é qualificado pela CAPES com o conceito 5[1].

O grupo de pesquisadores que participa deste projeto já tem uma larga experiência no tema de descoberta de conhecimento em bases de dados, como pode ser comprovado pela lista de publicações e trabalhos orientados dos professores participantes, apresentada em seus currículos Lattes.

[1] Segundo o Item 1.3.3 do Edital, programas da região Sul e Sudeste podem ser considerados consolidados quando houver carência de programas com conceito 6 e 7 na área. Este certamente é o caso da área de Ciência da Computação, a qual conta com apenas um programa nível 6 (o programa de Engenharia de Sistemas e Computação da COPPE/UFRJ, que colabora com a Área Interdisciplinar de Computação de Alto Desempenho da COPPE) e um programa nível 7 (o programa de Informática da PUC do Rio de Janeiro).

Equipe dos Grupos de Pesquisa Participantes

Nome	Titulação	Bolsista CNPq, Categoria-nível	Curso / Instituição	Função
Alberto Ferreira De Souza[2]	Doutor	II	PPGI / UFES	Coordenador
Andréa Maria Pedrosa Valli[3]	Doutor	não	PPGI / UFES	Pesquisador Doutor
Lucia Catabriga3	Doutor	não	PPGI / UFES	Pesquisador Doutor
Flávio Miguel Varejão3	Doutor	não	PPGI / UFES	Pesquisador Doutor
Thomas Walter Rauber3	Doutor	não	PPGI / UFES	Pesquisador Doutor
Sérgio Antônio Andrade de Freitas3	Doutor	não	PPGI / UFES	Pesquisador Doutor
Idílio Drago	Graduado	não	PPGI / UFES	Aluno de Mestrado
Suelen Loureiro Marconsini	Graduado	não	PPGI / UFES	Aluno de Mestrado
Gabriela Trazzi Perim	Graduado	não	PPGI / UFES	Aluno de Mestrado
Letícia Rosseti Margotto	Graduado	não	PPGI / UFES	Aluno de Mestrado
Stiven Schwanz Dias	Graduado	não	PPGI / UFES	Aluno de Mestrado
Felipe Thomaz Pedroni	Graduado	não	PPGI / UFES	Aluno de Mestrado
Jose Jeronimo Camata	Graduado	não	PPGI / UFES	Aluno de Mestrado
Alvaro Luiz Gayoso de Azeredo Coutinho3	Doutor	1-A	PEC/COPPE/UFRJ	Vice-Coordenador
José Luís Drummond Alves3	Doutor	1-D	PEC/COPPE/UFRJ	Pesquisador Doutor
Marcos A. D. Martins3	Doutor	não	NACAD/COPPE/UFRJ	Pós-doutorando
Renato Nascimento Elias	Mestre	não	PEC/COPPE/UFRJ	Aluno Doutorado
Denis Araujo Filgueiras de Souza	Mestre	não	NACAD/COPPE/UFRJ	Aluno Doutorado
Maria Carolina Monard3	Doutor/Livre Docente	não	PPG-CCMC ICMC/USP	Vice-Coordenador
Solange Oliveira Rezende3	Doutor/Livre Docente	não	PPG-CCMC ICMC/USP	Pesquisador Doutor
Alneu de Andrade Lopes3	Doutor	não	PPG-CCMC ICMC/USP	Pesquisador Doutor
Ronaldo Cristiano Prati3	Doutor	não	PPG-CCMC ICMC/USP	Pós-doutorando
Edson Matsubara	Mestre	não	PPG-CCMC ICMC/USP	Aluno de Doutorado
Jean Metz	Mestre	não	PPG-CCMC ICMC/USP	Aluno de Doutorado
Mariza Ferro	Mestre	não	PPG-CCMC ICMC/USP	Aluno de Doutorado
Maria Fernanda Moura	Mestre	não	PPG-CCMC ICMC/USP	Aluno de Doutorado
Veronica Oliveira de Carvalho	Mestre	não	PPG-CCMC ICMC/USP	Aluno de Doutorado
Walter Aioma Nagai	Mestre	não	PPG-CCMC ICMC/USP	Aluno de Doutorado
Daniel de Faveri Honorato	Graduado	não	PPG-CCMC ICMC/USP	Aluno de Mestrado

[2] O Prof. Alberto Ferreira De Souza declara aqui que foi indicado formalmente pelo Programa de Pós-Graduação em Informática da UFES como coordenador desta proposta.
[3] O(A) Pesquisador(a) concordou formalmente por escrito em participar deste projeto.

3. Caracterização dos Problemas que Serão Investigados

O fortalecimento do Programa de Pós-Graduação em Informática da UFES será evidenciado não somente pelo aumento do número de publicações em revistas científicas de qualidade, mas também pelo aumento do número de projetos de pesquisa realizados em conjunto com grupos de pesquisa de outros programas de pós-graduação consolidados e do intercâmbio de orientação e co-orientação com outros programas. Hoje, o programa sofre pala falta de um aprofundamento de parcerias com outros programas consolidados e por falta de financiamento para interação com tais programas. Com este projeto espera-se estabelecer laços de cooperação para a investigação de questões científicas relevantes nas áreas de Computação de Alto Desempenho e Inteligência Computacional. A seguir, discutimos estas questões.

Computação de Alto Desempenho

As questões científicas da área de Computação de Alto Desempenho que serão investigadas estão ligadas aos seguintes temas: (i) novas técnicas adaptativas no tempo e no espaço de representação de configurações de malha de elementos finitos aplicados a problemas de escoamentos em meios porosos; (ii) novos paradigmas de programação adequados para futuros sistemas paralelos com dezenas ou centenas de processadores implementados através de circuitos integrados *multi-core*. Os pesquisadores que investigarão estes temas já possuem cooperação científica relevante, como é evidenciado pelas publicações em conjunto listadas em seus currículos Lattes.

Novas técnicas adaptativas no tempo e no espaço de representação de configurações de malha de elementos finitos aplicados a problemas de escoamentos em meios porosos

A simulação numérica de escoamentos complexos demanda algoritmos eficientes e plataformas computacionais velozes. O uso de técnicas adaptativas permite ajustar a discretização de acordo com as necessidades da análise, mas gera dificuldades para o desenvolvimento de um programa paralelo. São necessárias técnicas de implementação computacionais eficientes de um método adaptativo que consiga representar as novas configurações da malha de elementos finitos e que, ao mesmo tempo, seja capaz de lidar com a distribuição do domínio nos diversos processadores. Um dos objetivos desse projeto é o estudo e implementação de técnicas adaptativas para o método dos elementos finitos aplicado a problemas de escoamentos em meios porosos, com computação de alto desempenho e utilizando a biblioteca LibMesh (*C++ Framework for the Numerical Simulation of Partial Differential Equations on Serial and Parallel Platforms* (http://libmesh.sourceforge.net). Esta biblioteca está sendo desenvolvida principalmente pelo *Computational Fluid Dynamics Laboratory* (CFDLAB) do *Institute for Computational Engineering and Sciences* (ICES) da *University of Texas at Austin*, Estados Unidos. Dois pesquisadores que participam deste projeto estão envolvidos com este desenvolvimento.

A principal característica da LibMesh é oferecer o desenvolvimento de modernos algoritmos de elementos finitos usando aspectos sofisticados de estimativas de erros e malhas adaptativas. Podem ser destacadas as seguintes funcionalidades: suporta problemas uni, bi e tridimensionais; possibilita refinamento local da malha, incluindo diferentes estratégias de refinamento adaptativo baseadas em indicadores de erros locais e estimativas de erros; e algoritmos rápidos para resolver problemas com milhões de graus de liberdade. O uso da biblioteca LibMesh permite liberar o programador das aplicações dos aspectos de gerenciamento da malha e do refinamento.

A solução aproximada obtida pelo método dos elementos finitos é dependente da discretização do domínio. Portanto, tem-se necessidade de determinar o tamanho dos elementos e a ordem de interpolação que proporcionem uma solução aproximada aceitável em uma determinada norma [Carey99]. Para atingir uma precisão desejada, é necessário adaptar a discretização da malha às características da solução. Esta adaptação baseia-se em estimadores de erro e na utilização de técnicas adequadas sobre os elementos com maior erro de aproximação. Na técnica tipo h-adaptativa, refina-se os elementos construindo seqüências de malhas de forma dinâmica e automática. Em outra técnica, tipo p-adaptativa, o que é modificado não é o tamanho dos elementos, mas sim o grau dos polinômios no espaço aproximante. Uma outra possibilidade é refinar as malhas e adaptar a ordem de interpolação, simultaneamente, conduzindo à idéia de hp-adaptatividade. Qualquer que seja a técnica escolhida é necessário definir estimadores de erro confiáveis associados à equação diferencial, além de dispor de algoritmos flexíveis que permitam

modificar a malha e/ou a ordem dos espaços aproximantes. Tipicamente, os estimadores de erro são do tipo a-priori e a-posteriori. Um estimador de erro a-priori é utilizado para prever o comportamento do erro de acordo com os parâmetros das malhas. Por outro lado, uma estimativa de erro a-posteriori é utilizada para avaliar o erro da solução aproximada computada. Neste projeto pretende-se estudar estimadores de erro a-posteriori capazes de tratar com eficiência as dificuldades envolvidas nos problemas de escoamentos complexos, tais como: a presença de frentes com altos gradientes, o caráter predominantemente convectivo das equações governantes e as heterogeneidades do meio [Coutinho01a, Coutinho01b , Catabriga02, Catabriga04a].

Por outro lado, o uso da técnica de decomposição de domínios está baseado na idéia de dividir para conquistar. Um problema original pode ser reformulado com uma família de sub-problemas de tamanho reduzido definidos em sub-domínios. Uma das principais vantagens da decomposição de domínios é a aplicação natural do paralelismo na solução dos sub-problemas. Neste projeto pretende-se estudar a aplicabilidade das ferramentas disponíveis, tais como o METIS [Zomaya96, Lima04], para efetuar não somente a partição inicial do domínio, mas os reparticionamentos subsequentes provocados pela evolução da solução adaptada no tempo e no espaço.

Dentre os métodos iterativos não estacionários mais populares na solução de sistemas de grande porte oriundos de discretizações de elementos finitos, destacam-se: o método dos gradientes conjugados para sistemas simétricos e o método dos resíduos mínimos generalizados (GMRES) para sistemas não simétricos [Kelley95, Saad03], Oliker02]. Recentemente foi introduzido o método das direções conjugadas à esquerda (LCD) [Yuan04, Dai04] como uma alternativa ao método GMRES. Testes iniciais do método LCD [Valentim04, Catabriga04b, Catabriga06a] demonstram que o método LCD possui convergência mais rápida, mas tende a ser mais caro computacionalmente por iteração.

Pesquisadores deste projeto vêm desenvolvendo pesquisas em conjunto com o grupo de pesquisadores do CFDLAB vinculado ao projeto de cooperação internacional CAPES-UT NO11/04 APFE (Elementos Finitos Adaptativos com Processamento de Alto Desempenho). Os resultados iniciais obtidos até o momento estao reportados em [Camata05, Catabriga06b, Camata06].

Novos paradigmas de programação adequados para futuros sistemas paralelos com dezenas ou centenas de processadores implementados através de circuitos integrados multi-core

O continuo aumento do número máximo de transistores que pode ser colocado em um único circuito integrado (CI, ou chip) tem permitido a implementação, em um único CI, de sistemas com múltiplos processadores (ou sistemas *multi-core*) cada vez mais elaborados e poderosos. Hoje, computadores do tipo PC (*personal computer*) para uso doméstico com dois processadores (*dual-core*) já estão disponíveis no mercado brasileiro e sistemas *multi-core* estão entre os tópicos de pesquisa mais importantes para os grandes fabricantes mundiais de processador (Intel, AMD, Sun, etc.). Isso porque, muito embora o aumento de desempenho dos processadores encapsulados em um único CI esteja aumentando continuamente, e não existam evidências de que esta tendência vá se modificar nos próximos 2-4 anos (Figura 1, [Olukotun05]), o aumento do desempenho advindo da exploração do paralelismo no nível de instrução (paralelismo de grão fino) em sistemas monoprocessador tem sido cada vez menor (Figura 2, [Olukotun05]). Além disso, o aumento do consumo de energia necessário para o continuo aumento de desempenho de sistemas monoprocessador está tornando inviável a implementação de monoprocessadores com mais paralelismo no nível de instrução e/ou com *clock* mais rápido (Figura 3 [Olukotun05]). Por essa razão, os grandes fabricantes mundiais de processador estão se voltando para a única opção aparentemente possível: sistemas *multi-core*.

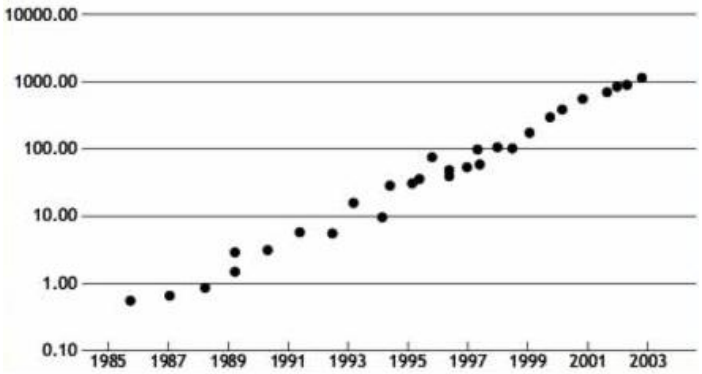


Figura 1: Desempenho relativo - processadores Intel

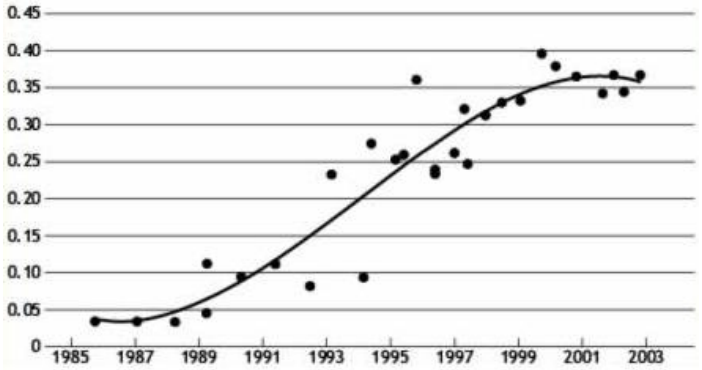


Figura 2: Instruções por ciclo - processadores Intel

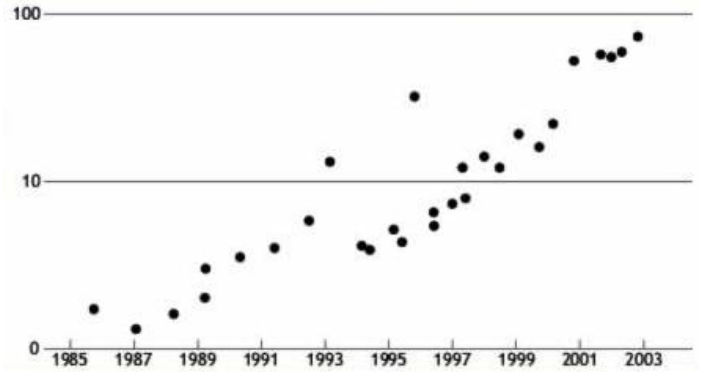


Figura 3: Consumo em Watts - processadores Intel

Sistemas *multi-core* tiram proveito do número de transistores disponível em um único CI, que continua a crescer a taxas históricas (segundo a Lei de Moore [Moore65]), para disponibilizar hardware próprio para a exploração do paralelismo de grão grosso (paralelismo disponível no nível de processo ou *thread*) existente nas aplicações atuais. Contudo, embora sistemas multiprocessados estejam entre nós desde a década de 1960 e tenham sido muito estudados nas décadas de 1980 e 1990 [Hennessy03], novos paradigmas de programação próprios para os sistemas *multi-core* que estão por vir nos próximos anos ainda são necessários (estima-se que em 2015 existam CIs *multi-core* em computadores do tipo PC com mais de 128 processadores). Neste projeto propomos investigar novos paradigmas de programação para exploração do paralelismo de hardware que estará disponível em futuros sistemas *multi-core*. Mais especificamente, estudaremos paradigmas voltados para a codificação da solução de problemas numericamente

intensivos. Estudaremos também paradigmas voltados para a codificação do processamento de vídeo e de dados de sensores baseados em laser em tempo real, que designamos aqui de processamento de *streams*, e de algoritmos e heurísticas para a descoberta de conhecimento em grandes bases de dados. Nós usaremos clusters de computadores como plataforma experimental de emulação dos sistemas *multi-core* futuros.

Inteligência Computacional

A descoberta de conhecimento em bases de dados [Fayad96] é um campo de pesquisa em Inteligência Artificial (IA) que tem crescido rapidamente. A grande motivação para esse crescimento está ligada, principalmente, à existência de uma poderosa tecnologia para coleta e gerenciamento de grande quantidade de dados. Considerando-se que, em geral, as bases contém muitos dados a ponto de tornar inviável a análise manual, tornou-se necessário o desenvolvimento de processos de análise automática. Contudo, foi percebido que, mesmo os dados estando armazenados e gerenciados por um sistema gerenciador de bancos de dados, eles podem conter campos que não são considerados na análise, como aqueles preenchidos com informações textuais e dados não estruturados. Além disso, essas informações não estão presentes apenas em tabelas, mas também em diversos documentos disponibilizados, armazenados e acessados com grande freqüência. Como a área de pesquisa de extração de conhecimento de base de dados é relativamente nova, ela possui poucas soluções efetivas e um amplo campo de investigação, envolvendo áreas como bancos de dados, estatística, aprendizado de máquina, extração de informação textual, e gestão do conhecimento e da informação.

Este projeto contempla o descobrimento de conhecimento utilizando Mineração de Dados (MD) [Rezende03, Witten00, Weiss98]. MD envolve uma série de atividades, que compreendem desde a definição do problema até a utilização dos resultados obtidos. Como muitas dessas atividades são aplicadas repetidamente em um ciclo, mineração de dados é na realidade um processo inerentemente interativo e iterativo: não se pode obter algum tipo de conhecimento útil simplesmente submetendo uma certa quantidade de dados a um sistema de MD. O usuário de um sistema de MD deve ter um sólido entendimento do domínio da aplicação para poder selecionar os subconjuntos de dados adequados para descrever padrões, bem como definir critérios apropriados para avaliá-los.

Tipicamente, os processos de MD são compostos por três etapas: pré-processamento, extração de padrões e pós-processamento. A etapa de pré-processamento envolve as tarefas de definição dos atributos que serão utilizados, extração e integração dos dados provenientes de diferentes fontes, transformação dos dados para formatos mais apropriados para a extração de conhecimento, limpeza e composição de atributos e redução do conjunto de dados.

A extração de padrões é a etapa na qual os padrões interessantes e que ocorrem mais freqüentemente são descobertos a partir dos dados. Nessa etapa pode-se utilizar vários métodos de extração de padrões, tais como análise estatística, modelagem matemática, visualização e, também, vários métodos de aprendizado de máquina supervisionado, semi-supervisionado e não-supervisionado [Mitchell97, Breiman84, Michalski83, Michie94].

O processo de MD não termina quando os padrões são encontrados. É necessário realizar um pós-processamento desses padrões [Hand97]. Em outras palavras, os padrões descobertos devem ser entendidos, avaliados, confrontados com o conhecimento já existente, ordenados, filtrados e traduzidos em termos inteligíveis para os usuários.

Deve ser observado que existem muitos problemas a serem solucionados em cada uma das etapas do processo de MD. O objetivo desta proposta é investigar e propor soluções a diversos problemas em aberto nessas etapas, bem como solucionar problemas específicos utilizando bases de dados reais. Alguns dos temas de interesse já identificados pelos membros da equipe, e nos quais vários dos membros da equipe já atuam, para desenvolver pesquisas conjuntas e formação de recursos humanos, são descritos a seguir.

Seleção de Atributos

Há diversas razões para a realização da seleção de atributos em um processo de mineração de dados. Dentre elas pode-se citar: a manipulação de um conjunto menor de atributos é menos custosa; a qualidade do processo de aprendizado pode ser aumentada quando se trabalha apenas com o subconjunto de atributos mais relevantes de uma base de dados; o modelo gerado é mais simples; a identificação dos atributos mais relevantes pode auxiliar no entendimento e no estudo do problema que está sendo analisado.

As vantagens em se utilizar a seleção de atributos se dão, principalmente, porque a grande quantidade de atributos pode trazer informações redundantes, contraditórias ou totalmente irrelevantes. Em aprendizado supervisionado, a indução de um modelo *h* é influenciado pelos valores dos atributos do conjunto de exemplos. Teoricamente, o uso de um maior número de atributos para descrever os exemplos deveria fornecer um maior poder de discriminação para aproximar a verdadeira função $y = f(x)$ desconhecida, onde x representa um exemplo do conjunto de dados, composto por N exemplos. Porém, isso pode não ocorrer na presença de atributos irrelevantes e/ou redundantes, os quais, freqüentemente, confundem o algoritmo de aprendizado. A área de Seleção de Atributos (SA), explorada não somente em estatística mas também em aprendizado de máquina e mineração de dados [Liu98], tem como objetivo determinar, segundo algum critério, quais atributos são importantes. Os resultados obtidos, tanto teórica quanto experimentalmente, mostram que a SA melhora a predição de modelos e reduz a complexidade de *h.*

Os diversos modelos de SA propostos na literatura podem ser categorizados nos modelos wrapper e filtro [Liu98]. Além dos atributos irrelevantes, tem sido observado que atributos redundantes também afetam a precisão dos classificadores induzidos e, portanto, deveriam ser eliminados [Koller96, Hall00]. Considera-se que dois atributos são redundantes entre si quando seus valores estão correlacionados, parcial ou completamente. Entretanto, o número de subconjuntos de atributos cresce exponencialmente com o número de atributos disponíveis e encontrar o subconjunto ótimo de atributos pode ser NP [Kohavi97]. Esse alto custo computacional pode torná-los ineficientes para conjuntos de dados com alta dimensionalidade. Neste projeto serão investigados modelos que possibilitem diminuir o custo computacional da busca por um subconjunto que aproxima o subconjunto ótimo. Em particular, modelos de SA adaptados para uso de computação paralela são bastante promissores.

Adaptações das Técnicas Clássicas de Classificação ao Caso de Dados com Variáveis Temporais

As principais técnicas de aprendizado supervisionado têm como característica em comum o fato de que um exemplo é classificado em uma classe, baseado em suas informações, independentemente de questões temporais que podem influenciar o valor deste registro. Em sistemas em que se deseja inferir uma classe em função de comportamentos que se alteram no decorrer do tempo, outras técnicas sensíveis às características temporais dos dados devem ser utilizadas. Aplicações desse tipo de mineração de dados estão disponíveis em [Rong02, Bellazzi05].

Para tratamento deste tipo especial de problema em mineração de dados, algumas abordagens têm sido estudadas. Uma abordagem comum encontrada na literatura é aplicar técnicas de transformação e extração de informações estáticas de séries temporais para criação de novas características, que são submetidas a sistemas de aprendizado convencionais. Essas abordagens tratam o problema como uma parte da fase de pré-processamento dos dados. Algumas técnicas aplicáveis a este contexto podem ser encontradas em [Daubechies92, Pollock99].

Outras abordagens tratam o problema também na fase de extração de padrões. Por exemplo, uma abordagem possível é aplicar técnicas de classificação especializadas ao caso temporal diretamente aos dados originais. Em [Roddick02] e [Antunes01] é apresentado um resumo das diversas técnicas de mineração de dados temporais disponíveis. Em [Elfeky05] são apresentadas técnicas de detecção de periodicidades em séries temporais. Em [Hoppner02] é discutido uma técnica de busca de dependência em intervalos de séries temporais distintas, com o objetivo de produzir regras de variação similar às normalmente criadas pela análise humana. Em [Yang04] uma técnica clássica de redução do espaço de características em um problema de classificação é adaptada para medição de similaridade entre duas curvas distintas - uma abordagem semelhante à encontrada em sistemas de aprendizado não supervisionado.

O estudo de adaptações das técnicas clássicas de classificação ao caso de dados com variáveis temporais é um possível campo de pesquisa a ser explorado. Diversos problemas práticos apresentam esta característica, por exemplo, a identificação de fraudes a partir de dados mensais dos clientes de prestadoras de serviços públicos (como telefonia e eletricidade), a previsão e o acompanhamento de valores de indicadores com variação temporal etc. Nesse contexto, este projeto busca também aprofundar os estudos neste ramo da mineração de dados por meio de intercâmbio de pesquisa, estudo e aplicação de técnicas existentes em problemas

reais e desenvolvimento de novas técnicas em cada uma das fases da mineração de dados. Cabe destacar que também aqui é possível a investigação do uso da computação paralela nas etapas de pré-processamento de dados (especificamente, na seleção de atributos) e de extração de padrões (especificamente, nos algoritmos que identificam padrões e similaridades entre séries temporais).

Integração de Classificadores Simbólicos utilizando Sistemas Multi-Agentes

De acordo com [Freitas98], o processo de mineração de dados distribuída consiste em particionar uma base de dados muito grande em bases menores, que são mineradas por múltiplos processadores, aplicando o mesmo ou diferentes algoritmos de aprendizado na fase de MD relacionada a extração de padrões. Posteriormente, é necessário integrar os resultados dos classificadores gerados nas diferentes bases de conhecimento distribuídas.

O objetivo desta pesquisa é estudar formas de integrar essas bases de regras de maneira eficiente. Este processo de integração pode ser visto como um tipo de resolução de problemas no qual se busca encontrar, em um espaço de possibilidades, um conjunto reduzido e consistente de regras que representa um bom modelo para todas as bases de dados utilizadas. Entretanto, o espaço de regras a ser pesquisado pode ser bastante grande, o que inviabiliza a utilização de técnicas de busca simples. Devido à distribuição inerente do problema e ao tamanho do espaço de possibilidades, pode-se investigar a utilização de Resolução Distribuída de Problemas (RDP) [Ferber99] para a solução do problema de integração de classificadores à base de regras.

A RDP é uma subárea da Inteligência Artificial Distribuída (IAD). Ela engloba as técnicas que permitem a um conjunto de entidades independentes (agentes) dividir e compartilhar conhecimentos correspondentes ao estado atual da solução de um problema particular. A interação entre essas entidades representa uma busca global em um espaço de soluções.

Dessa forma, pretende-se desenvolver técnicas baseadas em agentes para integrar classificadores baseados em regras atuando de forma paralela. Essas técnicas de integração permitirão que grandes bases de dados sejam mineradas a partir de amostras, reduzindo o tempo de processamento e possivelmente aumentado à precisão dos modelos, produzindo uma solução escalável para a mineração de grandes bases de dados.

4. Infra-Estrutura das Instituições Participantes

O Programa de Pós-Graduação em Informática da UFES possui um Laboratório de Computação de Alto Desempenho (LCAD – www.lcad.inf.ufes.br). O LCAD conta com uma variedade de sistemas computacionais para atender as necessidades de usuários de Computação de Alto Desempenho. O principal sistema de computação do LCAD é um cluster de 64+1 processadores com desempenho teórico máximo de 204 Gflop/s. Além do Cluster, o LCAD conta com várias estações de trabalho para dar apoio às tarefas de desenvolvimento e pesquisa de aplicações de Alto Desempenho.

Totalmente operacional desde janeiro de 2003, o Cluster do LCAD é composto de 64 nós de processamento ATHLON's XP 1800, com 256 MB de RAM, 20 GB de disco e 01 nó de administração (host) com 512 MB de RAM, 90 GB de disco. O cluster totaliza 16 GB de memória RAM e 1,2 TB de capacidade de armazenamento em disco.

O LCAD possui 10 (dez) estações de trabalho: 1 ATHLON 64 X2 3800 *dual-core* 2.4GHz, 4GB RAM DDR, 160GB de disco; 1 ATHLON 64 3700 2.2GHz, 4GB RAM DDR, 160GB de disco; 4 ATHLON XP 1800, 256 MB RAM, e 20 GB de disco; 2 Pentium III 800MHz, 192 RAM, 20 GB de disco; 1 SUN Ultra 5, 64 MB RAM, SunOS; e 1 AlphaStation DIGITAL 4/266. O LCAD conta também com um servidor de backup ATHLON 64 3700 dual-core 2.2GHz, 1GB RAM DDR e 1 TB de disco.

O grupo de Inteligência Computacional da UFES conta com um laboratório de pesquisa (Ninfa - Núcleo de Inferência e Algoritmos) composto por 8 estações de trabalho adquiridas pela própria Universidade. O grupo de pesquisa tem desenvolvido trabalhos em conjunto com empresas locais, que colaboram com a infra-estrutura do laboratório cedendo 6 estações de trabalho ao grupo.

Na COPPE, o presente projeto será desenvolvido no Núcleo de Atendimento em Computação de Alto Desempenho (NACAD). As instalações do NACAD, em uma área de 275m2 no Complexo I2000 do Centro de Tecnologia da UFRJ, possuem toda a infra-estrutura básica necessária, como energia elétrica estabilizada, rede local de comunicação de dados em fibra ótica, acesso por fibra

ótica a RNP, ar condicionado central, sistema eletrônico de segurança, biblioteca, sala de seminários e sala especial para clientes industriais. As máquinas atualmente disponíveis no NACAD são um computador SGI Altix 350, um InfoServer Itautec, um cluster de PC´s PIV e um cluster Itanium 2, integrados por um sistema de arquivos NetApp.

Os docentes, pesquisadores e alunos de pós-graduação em Inteligência Computacional do ICMC da USP contam com os recursos do Laboratório de Inteligência Computacional (LABIC), que possui várias estações de trabalho, conseguidas, através de diversos auxílios concedidos pela FAPESP, FINEP/RECOPE, PROTEM III-CC/CNPq, USP e CNPq. Além disso, o laboratório conta com um cluster de computadores com 12 nós de processamento interligados por Switch Fast Ethernet, sendo que cada nó é composto por 2 processadores Xeon de 2.4Ghz com 1 Giga de memória RAM DDR 266 e HD IDE de 40 Giga. O sistema operacional em uso no cluster é o Linux. Atualmente está sendo realizada uma reformulação do cabeamento de 24 pontos de rede estruturados, já que os cabos atuais não suportam a tecnologia Gigabit Ethernet.

Toda a infra-estrutura apresentada passará a ser usada de forma cooperativa dentro do escopo deste projeto.

5. Objetivos e Metas

O principal objetivo deste projeto é o fortalecimento das áreas de Computação de Alto Desempenho e Inteligência Computacional e o conseqüente fortalecimento do Programa de Pós-Graduação da UFES. Outros objetivos deste projeto são: o aumento do número de publicações em revistas científicas relevantes, o aumento do envolvimento dos alunos em publicações e participação em eventos científicos, o aumento da produção de patentes e outros instrumentos de propriedade intelectual.

As metas deste projeto são:

1. Orientar 2 alunos de doutorado
2. Orientar 6 alunos de mestrado
3. Orientar 6 bolsistas de iniciação científica
4. Publicar 6 artigos nacionais em parceria
5. Publicar 6 artigos em revistas científicas internacionais relevantes em parceria

6. Metodologia

Essa seção descreve detalhes específicos das metolodogias aplicadas para o desenvolvimento das pesquisas em Computação de Alto Desempenho e Inteligência Computacional.

Computação de Alto Desempenho

Novas técnicas adaptativas no tempo e no espaço de representação de configurações de malha de elementos finitos aplicados a problemas de escoamentos em meios porosos

No estudo das técnicas adaptativas pretende-se aprofundar os estudos sobre quais os estimadores de erros a-posteriori são mais indicados para a análise adaptativa de diversos problemas de escoamento. Essa etapa é importante, já que em alguns tipos de problemas podem existir algumas dificuldades inerentes, tais como a presença de frentes com alto gradiente que se movem e materiais heterogêneos, isto é, propriedades como porosidade, viscosidade e permeabilidade são dependentes da posição [Coutinho99]. Além disso, pretende-se investigar a extensão das técnicas adaptativas de seleção do passo de tempo baseadas em teoria de controle, propostas para malhas fixas, para o caso de malhas adaptativas [Valli05].

Malhas de elementos finitos são tipicamente não-estruturadas e portanto, permitem a geração automática de malhas para geometrias complexas provenientes da discretização de problemas da natureza. Além disso, a habilidade de adaptar dinamicamente tais malhas para a solução eficiente de problemas envolvendo a evolução de grandezas físicas como saturação, concentração, pressão etc fornece uma ferramenta poderosa de análise [Carey97]. Porém, a paralelização do

remalhamento adaptativo é uma tarefa difícil, devido principalmente ao desbalanceamento de carga criado pela dinâmica de refinamentos/desrefinamentos sucessivos das malhas. A ferramenta de particionamento de malhas mais usada na área de elementos finitos é o METIS, que emprega um algoritmo multi-nível em três níveis [Karypis95]. O primeiro nível restringe o grafo a ser particionado. O segundo nível particiona o grafo reduzido, enquanto que o terceiro nível projeta o grafo particionado no grafo original. Conseqüentemente o processamento cai na categoria de algoritmos irregulares, o que coloca dificuldades adicionais. Além disso, deve-se levar em conta a forma dos grafos gerados pelo refinamento/desrefinamento espacial com e sem enriquecimento do espaço de interpolação (refinamento h, p e hp) [Bose99].

Problemas gerais de escoamento possuem equações governantes com acoplamento de variáveis de interesse envolvendo pressão, temperatura e velocidade. O processo de solução envolve aproximações no tempo e no domínio considerado. Considerando o preocesso de solução baseado em discretizar o problema no domínio pelo método dos elementos finitos e no tempo por um algoritmo preditor-multicorretor baseado no método das diferenças finitas [Catabriga02]. Em cada passo do algoritmo de aproximação no tempo é necessário resolver sistemas lineares para obter a aproximação das variáveis de interesse. Considerando a implementação paralela baseada no método de decomposição de dados aplicada ao método dos elementos finitos tem-se em cada passo de tempo matrizes decompostas nos diversos sub-domínios [Lima04a, Lima04b]. Pretende-se aplicar essa técnica para estudar aplicações de multifísica e multi-escala usando elementos finitos adaptativos em Cluster de PC.

Novos paradigmas de programação adequados para futuros sistemas paralelos com dezenas ou centenas de processadores implementados através de circuitos integrados multi-core

O processamento de *streams* compreende um amplo conjunto de problemas que, na sua maioria, são caracterizados por possuir uma grande quantidade de paralelismo de grão grosso e aplicação comercial crescente. Exemplos de aplicações que requerem o processamento de *streams* de vídeo em tempo real incluem o reconhecimento de faces em áreas de grande movimento, reconhecimento de padrões de comportamento humano a partir de seqüências de movimentos capturados em vídeo e por laser range scan, reconhecimento de padrões em imagens capturadas por câmeras e scans de laser em ambientes industriais, entre outras. Todas estas aplicações requerem um grande poder computacional que hoje está disponível apenas em supercomputadores e grandes clusters de PCs. Contudo, com o avanço dos sistemas *multi-core*, logo este poder computacional estará disponível em computadores de baixo custo, ampliando significativamente o espectro de aplicações envolvendo o processamento de *streams*. Para ilustrar este fato, destacamos que o processador *multi-core* Cell [Paulson05, Kahle05], que equipará os novos consoles de vídeo game Playstation III, possui poder computacional bruto superior ao do Cluster Enterprise do Departamento de Informática da UFES, que figurava em 48º lugar da lista de clusters mais poderosos do mundo (www.top500cluster.com) em 2003 (Cell 256 GFlop/s x Enterprise 196 GFlop/s).

Para explorar o paralelismo de grão grosso presente no processamento de *streams*, tirando proveito do grande número de processadores que estarão disponíveis em futuros CIs, propomos a utilização do conceito de memória compartilhada distribuída implementada por software (*Software Distributed Shared Memory - S-DSM*) [Hennessy03, Amza96].

Em sistemas *multi-core* atuais é possível implementar memória compartilhada em hardware via coerência de cache. Contudo, assim como tem se mostrado inviável a implementação eficiente de coerência de cache em sistemas multiprocessados implementados com grande número de CIs monoprocessador (128 ou mais), acreditamos que existirão dificuldades na implementação de mecanismos de compartilhamento de memória via hardware em sistemas *multi-core* futuros.

Quando comparados com sistemas multiprocessados atuais implementados com CIs monoprocessador, sistemas *multi-core* apresentam latência de comunicação entre processadores menor e taxa de transferência de dados maior, o que beneficia a implementação de sistemas DSM no hardware de sistemas *multi-core*. No entanto, estas mesmas características são também benéficas para a implementação de sistemas S-DSM em *multi-cores*. Vale destacar que o CI Cell, por exemplo, possui nove processadores mas apenas um deles possui caches; os outros oito possuem apenas blocos privativos de memória de 256Kbytes sem mecanismos de coerência entre eles implementados em hardware – a movimentação dos dados entre estes blocos de memória deve ser orquestrada pelo programador [Kahle05]. Alternativas que disponibilizem os benefícios de

sistemas com memória compartilhada, mas mantenham a simplicidade dos sistemas *multi-core* sem coerência de cache, serão atrativas e, acreditamos, devem ser examinadas. Um dos objetivos deste projeto de pesquisa é investigar novos mecanismos de implementação de S-DSM apropriados para o processamento de *streams* e de fácil e eficiente implementação em futuros CIs multi-core.

Em sistemas S-DSM atuais, tipicamente existe uma única memória compartilhada. As principais desvantagens deste arranjo são: (i) a necessidade de se manter a coerência desta memória compartilhada através de um sistema de sincronização único que perpassa todos os processadores; e (ii) o desperdício de memória provocado pela necessidade da existência de cópia de toda a memória compartilhada em todos os processadores, mesmo quando alguns deles acessam apenas parte dela. Nos sistemas S-DSM que investigaremos durante este trabalho de pesquisa, ao invés de apenas uma memória global compartilhada, existirão regiões de memória compartilhada, e estas regiões serão compartilhadas apenas por grupos de processadores ao invés de por todos os processadores. Além disso, as regiões de memória compartilhada serão produzidas ou (exclusivo) consumidas por cada processador: um produtor poderá ler ou escrever em suas regiões de memória compartilhada; mas um consumidor poderá apenas ler de suas regiões compartilhadas. Este novo arranjo permite um melhor uso da memória disponível e tem o potencial de simplificar e, conseqüentemente, tornar mais veloz os sistemas S-DSM propostos quando comparados aos existentes, especialmente para aplicações que envolvam o processamento de *streams*.

A Figura 4 mostra, através de um diagrama de blocos, como tipicamente é feito o processamento de um fluxo de imagens em um sistema monoprocessador. Nos sistemas S-DSM que investigaremos, o processamento poderá ser distribuído entre vários processadores como apresentado na Figura 5. No sistema mostrado na Figura 5, as estruturas de dados sendo produzidas e consumidas podem ser replicadas de modo que, enquanto uma está sendo produzida/consumida, outras possam estar sendo consumidas/produzidas. Note que as estruturas de dados podem também ser particionadas e distribuídas entre processadores adicionais eventualmente disponíveis.

Para que uma aplicação seqüencial possa ser paralelizada para tirar proveito dos novos sistemas S-DSM que serão estudados, tipicamente será preciso apenas identificar que estruturas de dados devem ser processadas em quais processadores. Barreiras de sincronização e atualização global não serão necessárias – a verificação da disponibilidade da estrutura de dados pelos produtores/consumidores será usada para sincronizar a computação paralela. Buscar-se-á simplicidade na definição da interface entre os sistemas S-DSM que serão investigados e as linguagens de programação (seqüenciais) existentes. Esta interface estará associada à alocação dos dados (em C, a função *malloc*) e à mecanismos de controle de acesso aos mesmos apenas.

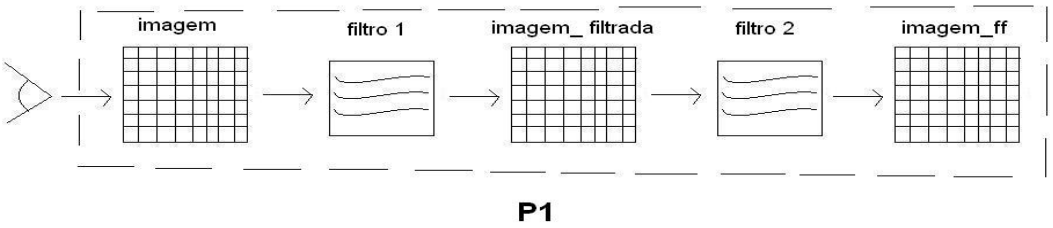


Figura 4: Processamento de imagens em um monoprocessador

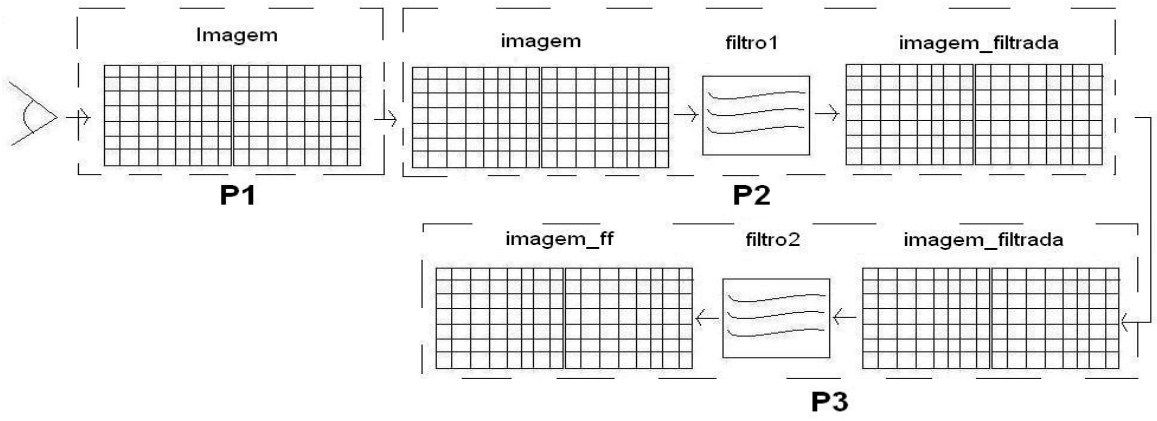


Figura 5: Processamento de imagens em um multiprocessador com o sistema S-DSM proposto

Inteligência Computacional

Como pode ser observado, os pesquisadores integrantes da equipe realizaram trabalhos relacionados aos temas propostos neste plano. A metodologia a ser seguida para a sua execução contempla os seguintes aspectos:

2. Atualização bibliográfica;
3. Realização de seminários e cursos intensivos de curta duração para os alunos de pós-graduação que participam do projeto, ministrados pelos pesquisadores responsáveis em suas áreas de especialidade;
4. Contatos periódico dos pesquisadores por meio de teleconferências, bem como intercâmbio de experiências e dos resultados parciais alcançados durante o desenvolvimento do projeto;
5. Avaliação das funcionalidades dos sistemas descritos e avaliação geral do desempenho dos sistemas utilizando casos de teste escolhidos para essa finalidade;
6. Avaliação e publicação de resultados.

Quanto à análise de resultados, deve ser observado que na etapa de extração de padrões no processo de mineração, são utilizados algoritmos de aprendizado de máquina. Aprendizado de máquina é uma área naturalmente empírica [Dietterich90], na qual pesquisadores geralmente conduzem estudos experimentais para avaliar o desempenho do método ou algoritmo proposto [Langley00]. Para uma boa avaliação, é fundamental desenhar criteriosamente os experimentos, identificando os critérios de avaliação e variáveis independentes, além da (possível) neutralização de variáveis aleatórias que fogem ao controle do experimentador.

Tipicamente, o critério de avaliação está relacionado a uma medida de desempenho (por exemplo a taxa de erro de classificação), mas outras métricas, tais como características do modelo induzido também são utilizadas. Quanto à identificação das variáveis independentes, a abordagem freqüentemente empregada inclui a utilização de um ou mais métodos de controle - geralmente um algoritmo ou método estabelecido e utilizado pela comunidade - para a comparação com novos métodos ou algoritmos propostos. Ambas as abordagens (controle e novas propostas) são executados nos mesmos domínios. Esses domínios são, geralmente, conjuntos de dados *benchmark* disponíveis em repositórios públicos de dados, tais como os da UCI.

A maioria desses estudos tenta estabelecer se o método proposto é melhor que o método de controle e, para tanto, tratam os domínios como variáveis aleatórias, pois estamos interessados no comportamento médio do método e não no desempenho em um conjunto de dados em particular. Por esses motivos, é importante que no estudo sejam incluídos conjuntos de dados variados.

Outros fatores, tais como a acomodação do algoritmo a um conjunto de dados em particular, são geralmente tratados pela repetição da execução do algoritmo várias vezes em diferentes (sub) amostras dos dados. Nesse sentido, os métodos mais aplicados são validação cruzada e amostragem bootstrap [Kohavi95], além de se testar um modelo em uma amostra de dados

diferente daquela utilizada na sua indução. Outro aspecto muito importante que deve ser levado em conta no aprendizado supervisionado e semi-supervisionado é o desbalanceamento entre as classes [Prati04a,Prati04b, Batista05, Batista04, Matsubara06], o qual é freqüente em situações reais. Finalmente, é importante ressaltar que para se tirar conclusões a respeito de estudos em aprendizado de máquina é muito importante a utilização de testes de hipóteses [Demsar06], para se determinar o grau de confiança em que as conclusões foram tiradas. Todos esses aspectos serão rigorosamente levados em consideração no desenvolvimento deste trabalho.
Além da análise experimental com conjuntos de dados *benckmark*, também é necessário a validação dos métodos propostos em dados reais. Para a construção de modelos, serão utilizados algoritmos de aprendizado simbólico proposicionais, tais como [Clark91], C4.5 e C4.5 Rules [Quinlan93], Roccer [Prati05], relacionais [Ferro05] ou *ensembles* de classificadores simbólicos [Bernardini05]. Deve ser ressaltada a importância dos modelos simbólicos [Hand97, Monard03, Monard05], assim como a extração de atributos importantes [Lee06a, Lee06b], já que muitas vezes os especialistas requerem a explicação de novas classificações para validar o conhecimento induzido. Também serão utilizados algoritmos de aprendizado semi-supervisionado [Matsubara04] e de interpretação de clusters [Metz05] para auxiliar o especialista na rotulação de um número adequado de casos para o uso de algoritmos de aprendizado supervisionado.

7. Cronograma

Atividade	**Descrição**	**Bimestre Início**	**Bimestre Término**
1	Atualização Bibliográfica	1	3
2	Pesquisa e desenvolvimento de novas arquiteturas de alto desempenho, de técnicas de mineração de dados e da integração entre computação de alto desempenho e mineração de dados	1	10
3	Realização de experimentos	3	11
4	Elaboração de relatórios, artigos e patentes	8 e 12	-

	Bimestre											
Atividade	1	2	3	4	5	6	7	8	9	10	11	12
1	■	■	■									
2	■	■	■	■	■	■	■	■	■	■		
3			■	■	■	■	■	■	■	■	■	
4								■				■

8. Orçamento

Viabilidade Financeira e Técnica

Para a realização deste trabalho serão necessárias pelo menos 2 bolsas de doutorado, 6 bolsas de mestrado e 6 bolsas de iniciação científica por ano, um cluster de pelo menos 32 processadores, câmeras de vídeo firewire, laser range scan, estações de trabalho, impressoras, livros (material bibliográfico), software, material de consumo e recursos para participação em congressos. O cluster será utilizado para a realização dos experimentos, as estações de trabalho serão utilizadas para edição de relatórios e trabalhos científicos, e impressoras para impressão dos relatórios e dos trabalhos científicos a serem escritos.

Dos itens necessários já estão disponíveis um cluster de 64 processadores (www.lcad.inf.ufes.br), parte das estações de trabalho, câmeras de vídeo firewire, impressoras, parte do material bibliográfico, software e parte do material de consumo. O Programa de Pós-Graduação em Informática da UFES têm recebido bolsas de mestrado em número suficiente para as áreas de Computação de Alto Desempenho e Inteligência Computacional, e os programas consolidados, bolsas de doutorado suficientes para o sucesso deste projeto. Bolsas de iniciação científica têm sido também obtidas pelos pesquisadores destas áreas no PIBIC-UFES e no CNPq. Contudo, para garantir a viabilidade técnica e financeira deste projeto de pesquisa serão necessários recursos para aquisição de equipamentos adicionais (estações de trabalho e um laser range scan), material bibliográfico, material de consumo, além de recursos para viabilizar viagens para reuniões científicas e de apresentação de trabalhos em conferências nacionais e internacionais.

Recursos Solicitados ao CNPq

Na Tabela 1, abaixo, são detalhados os investimentos que precisam ser feitos no Programa de Pós-Graduação em Informática da UFES para viabilizar a execução deste projeto. As passagens aéreas e diárias nacionais visam permitir a apresentação de trabalhos em congressos e a realização de visitas técnicas aos grupos dos programas consolidados. As passagens aéreas e diárias internacionais visam permitir a apresentação de trabalhos em congressos no exterior. O notebook e o laser range scan serão utilizados nos trabalhos do grupo de Computação de Alto Desempenho e a impressora laser, um notebook e os seis microcomputadores serão utilizados pelo grupo de Inteligência Computacional (os notebooks serão importantes nas viagens de intercâmbio entre os grupos). Os recursos destinados ao material de consumo serão utilizados na compra de toners, papel, dvds e outros materiais auxiliares. Os recursos destinados ao material bibliográfico serão utilizados na aquisição de livros mais recentes relacionados aos temas do projeto.

Tabela 1: Investimentos no Programa de Pós-Graduação em Informática da UFES

Item	Descrição	Quantidade	Valor Unit.	Valor
1	Passagem aérea nacional	25	R$ 800,00	R$ 20.000,00
2	Diárias no Brasil	125	R$ 187,83	R$ 23.478,75
3	Passagem aérea internacional	8	R$ 4.050,00	R$ 32.400,00
4	Diárias no exterior	40	R$ 594,00	R$ 23.760,00
5	Notebook	2	R$ 5.400,00	R$ 10.800,00
6	Impressora Laser	1	R$ 2.000,00	R$ 2.000,00
7	Microcomputadores	6	R$ 3.000,00	R$ 18.000,00
8	Laser range scan	1	R$ 18.900,00	R$ 18.900,00
9	Material de consumo	-	R$ 6.000,00	R$ 6.000,00
10	Material bibliográfico	-	R$ 8.000,00	R$ 8.000,00
11	Serviços de terceiros	-	R$ 3.000,00	R$ 3.000,00
			Total	R$ 166.338,75

Na Tabela 2, abaixo, são detalhados os investimentos que precisam ser feitos no Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional do ICMC/USP para viabilizar a execução deste projeto. As passagens aéreas e diárias nacionais visam permitir a realização de 4 cursos intensivos de curta duração (um por semestre) para os pesquisadores e alunos de pós-graduação do programa não consolidado. As passagens aéreas e diárias internacionais visam permitir a apresentação de trabalhos em congressos no exterior. Os recursos destinados ao material de consumo serão utilizados na compra de 7 toners de impressora. Os recursos destinados a serviços de terceiros serão aplicados na instalação de equipamentos, softwares e afins. O microcomputador será usado para experimentos e para a realização de vídeo conferências, e os dois monitores substituirão monitores antigos de 15". Os recursos destinados ao material bibliográfico serão utilizados na aquisição de livros mais recentes relacionados aos temas do projeto.

Tabela 2: Investimentos no Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional do ICMC/USP

Item	Descrição	Quantidade	Valor Unit.	Valor
1	Passagem aérea nacional	4	R$ 800,00	R$ 3.200,00
2	Diárias no Brasil	20	R$ 187,83	R$ 3.756,60
3	Passagem aérea internacional	2	R$ 4.050,00	R$ 8.100,00
4	Diárias no exterior	10	R$ 594,00	R$ 5.940,00
5	Microcomputador	1	R$ 4.000,00	R$ 4.000,00
6	Monitor SVGA 17”	2	R$ 1.000,00	R$ 2.000,00
7	Material de consumo	-	R$ 3.000,00	R$ 3.000,00
8	Material bibliográfico	-	R$ 2.000,00	R$ 2.000,00
9	Serviços de Terceiros	-	R$ 2.000,00	R$ 2.000,00
			Total	R$ 33.996,60

Na Tabela 3, abaixo, são detalhados os investimentos que precisam ser feitos no NACAD-COPPE/UFRJ para viabilizar a execução deste projeto. Assim como no caso da colaboração com o ICMC/USP, as passagens aéreas e diárias nacionais visam permitir a realização de 4 cursos intensivos de curta duração (um por semestre) para os pesquisadores e alunos de pós-graduação do programa não consolidado. As passagens aéreas e diárias internacionais visam permitir a apresentação de trabalhos em congressos no exterior, produzidos dentro do esforço de colaboração. Os recursos destinados ao material de consumo serão utilizados na compra de toners e cartuchos para impressoras Laser e Laser a cores disponíveis no NACAD. Os recursos destinados a serviços de terceiros serão aplicados nas instalações de equipamentos, softwares e afins. Os monitores novos substituirão os monitores antigos de 15" da equipe. Tais monitores são necessários para suportar a visualização dos resultados das simulações efetuadas. Os recursos destinados ao material bibliográfico serão utilizados na aquisição de livros mais recentes relacionados aos temas do projeto.

Tabela 3: Investimentos na Área Interdisciplinar de Computação de Alto Desempenho da COPPE/UFRJ

Item	Descrição	Quantidade	Valor Unit.	Valor
1	Passagem aérea nacional	4	R$ 800,00	R$ 3.200,00
2	Diárias no Brasil	20	R$ 187,83	R$ 3.756,60
3	Passagem aérea internacional	2	R$ 4.050,00	R$ 8.100,00
4	Diárias no exterior	10	R$ 594,00	R$ 5.940,00
5	Monitor SVGA 17”	6	R$ 1.000,00	R$ 6.000,00
5	Material de consumo	-	R$ 3.000,00	R$ 3.000,00
7	Material bibliográfico	-	R$ 2.000,00	R$ 2.000,00
8	Serviços de Terceiros	-	R$ 2.000,00	R$ 2.000,00
			Total	R$ 33.996,60

9. Resultados Esperados

Esperamos atingir resultados práticos quantitativos e qualitativos com este trabalho de pesquisa. Como resultados quantitativos deste trabalho esperamos:

1. Orientar 2 alunos de doutorado
2. Orientar 6 alunos de mestrado
3. Orientar 6 bolsistas de iniciação científica
4. Publicar 6 artigos nacionais em parceria
5. Publicar 6 artigos internacionais em revistas científicas internacionais relevantes em parceria

Os principais resultados qualitativos deste trabalho serão: estender o estado da arte em programação de sistemas paralelos; estender o estado da arte em técnicas temporais de mineração de dados, seleção de atributos e em classificadores simbólicos integrados por sistemas multi-agentes; estender o estado da arte na aplicação de computação paralela ao processo de mineração de dados; e a disponibilização local, nacional e internacional das ferramentas de programação paralela e dos códigos paralelos desenvolvidos.

Além destas contribuições, este trabalho contribuirá ainda com a formação de especialistas em computação de alto desempenho e inteligência computacional no nível de doutorado, mestrado e de graduação.

10.Referências Bibliográficas

[Amza96] C. Amza, A.L. Cox, S. Dwarkadas, P. Keleher, H. Lu, R. Rajamony, W. Yu, and W. Zwaenepoel, "TreadMarks: Shared Memory Computing on Networks of Workstations", IEEE Computer, Vol. 29, No. 2, pp. 18-28, February 1996.

[Antunes01] Antunes, C. M. and A. L. Oliveira (2001). Temporal data mining: an overview. Lecture Notes in Computer Science 1, KDD Workshop on Temporal Data Mining.

[Batista04] Batista, G. E. A. P. A., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explorations, 6(1): 20–29.

[Batista05] Batista, G. E. A. P. A., Prati, R. C., and Monard, M. C. (2005). Balancing strategies and class overlapping. In International Symposium on Intelligent Data Analysis (IDA'2005), volume 3646 of Lecture Notes in Computer Science, pages 24–35, Madrid (Spain). Springer-Verlag.

[Bernardini05] Bernardini, F. C., Monard, M. C., and Prati, R. C. (2005). Constructing ensembles of symbolic classifiers. In International Conference on Hybrid Intelligent Systems — HIS 2005, volume 1, pages 315–320, Rio de Janeiro. California: IEEE Computer Society.

[Bose99] Bose A, Carey GF A class of data structures and object-oriented implementation for finite element methods on distributed memory systems COMPUT METHOD APPL M 171 (1-2): 109-121 MAR 26 1999

[Breiman84] Breiman, L., J. Friedman, R. Olshen, and C. Stone (1984). Classification and Regression Trees. Pacific Grove, CA: Wadsworth & Books.

[Camata05] J.J. Camata, L. Catabriga, A.M.P. Valli, A.L.G.A. Coutinho and G. Carey, Numerical Evaluation of the LCD method implemented in the LibMesh Library, CILAMCE 2005, Guarapari, Brasil.

[Camata06] J.J. Camata,A.M.P. Valli, L. Catabriga,A.L.G.A. Coutinho, G.F. Carey, Simultaneous Space-Time Adaptive Finite Element Simulations on Fluid Applications, 2nd LNCC Meeting on Computational Modeling, August 8-11, 2006 (accepted).

[Carey97] G. F. Carey, Computational Grids: Generations, Adaptation and Solution Strategies, Taylor - Francis, 1997.

[Carey99] Carey GF, McLay R, Bicken G, et al., Parallel finite element solution of three-dimensional Rayleigh-Benard-Marangoni flows, INT J NUMER METH FL 31 (1): 37-52 SEP 15 1999.

[Catabriga02] L. Catabriga and A.L.G.A. Coutinho, Implicit SUPG Solution of Euler Equation Using Edge-Based Data Structures, Computer Methods in Applied Mechanics and Engineering, 191/32:3477-3490, 2002.

[Catabriga04a] L. Catabriga, A.L.G.A.Coutinho and T.E. Tezduyar, Finite element SUPG parameters computed from local edge matrices for compressible flows, Computational Fluid Dynamics Journal, 13(3):62, pp.540-459, 2004.

[Catabriga04b] L. Catabriga, A. L. G.A. Coutinho, and L.P. Franca, Evaluating the LCD algorithm for solving linear systems of equations aring from implicit SUPG formulation of compressible flows, International Journal for Numerical Methods in Engineering, Vol. 60, pp. 1513—1534, 2004.

[Catabriga06a] L. Catabriga, A.M.P. Valli, B.Z. Melotti, L.M. Pessoa, A.L.G.A.Coutinho, Comparison between GMRES and LCD iterative methods in the finite element and finite difference solution of convection-diffusion equations, Communications in Numerical Methods in Engineering, Vol. 22, pp 643-656, 2006

[Catabriga06b] L. Catabriga, J.J. Camata, A.M.P. Valli, A.L.G.A. Coutinho, G.F. Carey, Reordering Effects on Preconditioned Krylov Methods in AMR solutions of Flow and Transport, WCCM06, Los Angeles ,16-21 Julho (Accepted).

[Clark91] Clark, P. and Boswell, R. (1991). Rule induction with CN2: Some recent improvements. In Kodratoff, Y., editor, Proc. of the 5th European Conference EWSL 91, pages 151–163. Springer-Verlag.

[Coutinho01a] Coutinho, A. L. G. A., Martins, M.A.D., Alves, J. L. D., Landau, L., Moraes, A. Edge-Based Finite Element Techniques for Nonlinear Solid Mechanics Problems. International Journal for Numerical Methods in Engineering, v.50, n.9, p.2053-2068, 2001.

[Coutinho01b] Coutinho, A. L. G. A., Sampaio, Paulo A. B., A natural derivation of discontinuity capturing operator for convection-diffusion problems. Computer Methods in Applied Mechanics and Engineering, 190, p.46-47, 2001.

[Coutinho99] Coutinho, A. L. G. A., Parsons, I. D., Finite Element Multigrid for Two-Phase Immiscible Flow in Heterogeneous Media. Communications in Numerical Methods in Engineering, 15, p.1-7, 1999.

[Dai04] Y. Dai, and J. Yuan, Study on semi-conjugate direction methods for non-symmetric systems, International Journal for Numerical Methods in Engineering, Vol. 60, pp. 1383—1399, 2004.

[Daubechies92] Daubechies, I. (1992). Ten Lectures on Wavelets. SIAM.

[Demsar06] Demsar, J. (2006). Statistical comparison of classifiers over multiple data sets. Journal of Machine Learning Research, 7(1):1–30.

[Dietterich90] Dietterich, T. G. (1990). Exploratory research in machine learning. Machine Learning, 5(1):5–10.

[Elfeky05] Elfeky, M. G., W. G. Aref, and A. K. Elmagarmid (2005). Periodicity detection in time series databases. IEEE Transactions on Knowledge and Data Engineering 17 (7), 875–887.

[Fayad96] Fayyad, U. M., G. Platestsky-Shapiro, and P. Smyth (1996). Advances in Knowledge Discovery and Data Mining, Chapter From Data Mining to Knowledge Discovery: An Overview, pp. 1–30. Menlo Park, CA: American Association for Artificial Intelligence (AAAI Press).

[Ferber99] Ferber, J. (1999). Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. Addison-Wesley.

[Ferro05] Ferro, M. and Monard, M. C. (2005). Uma ferramenta para converter exemplos de aprendizado no formato atributo-valor para o formato apropriado ao uso de algoritmos de programação em lógica indutiva. In Anais da XXXI Conferência Latinoamericana de Informática (CLEI2005), pages 109–120, Cali Colombia.

[Freitas98] Freitas, A. A. and S. Lavington (1998). Mining very large databases with Parallel Processing. Kluwer Academic Publishers.

[Hall00] Hall, M. A. (2000). Correlation-based feature selection for discrete and numeric class machine learning. In Proc. of the 17th Int. Conf. on Machine Learning, San Francisco, CA, pp. 359–366. Morgan Kaufmann.

[Hand97] Hand, D. J. (1997). Construction and Assessment of Classification Rules. John Wiley and Sons.

[Hennessy03] J. L Hennessy and D. A. Patterson, “Computer Architecture: A Quantitative Approach, Third Edition”, Morgan Kaufmann Publishers, Inc., 2003.

[Hoppner02] Hoppner, F. (2002). Learning dependencies in multivariate time series. In Proceedings of the ECAI’02 Workshop

[Kahle05] J. A. Kahle, M. N. Day, H. P. Hofstee, C. R. Johns, T. R. Maeurer, and D. Shippy, "Introduction to the Cell multiprocessor", IBM Journal of Research and Development, Vol. 49, No. 4/5, pp. 589-604, July/September 2005.

[Karypis95] G. Karypis and V. Kumar, Parallel Multilevel Graph Partitioning, Technical Report TR 95-036, Department of Computer Science, University of Minnesota, Minneapolis, 1995.

[Kelley95] C.T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, Philadelphia, 1995.

[Kohavi95] Kohavi, R. and Li, C.-H. (1995). Oblivious decision trees, graphs, and top-down pruning. In Proc. of the 14th IJCAI, pages 1071–1077, Montreal, Canada.

[Kohavi97] Kohavi, R. and G. H. John (1997). Wrappers for feature subset selection. Artif. Intell. 97 (1-2), 273–324.

[Koller96] Koller, D. and M. Sahami (1996). Toward optimal feature selection. In Proc. of the 13th Int. Conf. on Machine Learning, Bari, Italy, pp. 284–292.

[Langley00] Langley, P. (2000). Crafting papers on machine learning. In Proc. 17th International Conf. on Machine Learning, pages 1207–1212. Morgan Kaufmann, San Francisco, CA.

[Lee06a] Lee, H. D., Monard, M. C., and Chung, W. F. (2006a). A fractal dimension based filter algorithm to select features for supervised learning. In Brazilian Symposium on Artificial Intelligence (SBIA/IBERAMIA '2006), Lecture Notes in Artificial Intelligence. in print.

[Lee06b] Lee, H. D., Monard, M. C., Voltolini, R. F., Prati, R. C., and Chung, W. F. (2006b). A simple evaluation model for feature subset selection algorithms. In Eighth Argentine Symposium on Artificial Intelligence (ASAI 2006). submetido.

[Lima04a] L. M. Lima, B. Z. Melotti, L. Catabriga and, A.M.P. Valli, Estratégias de Armazenamento para Implementações Paralelas do Método dos elementos Finitos, WSCAD 2004.

[Lima04b] L.M. Lima, B.Z. Melotti, L. Catabriga and A.M.P. Valli, Uma implementação paralela eficiente do método dos elementos finitos para a equação de advecção e difusão, CILAMCE 2004, Recife, Brasil.

[Liu98] Liu, H. and H. Motoda (1998). Feature Selection for Knowledge and Data Mining. Massachusetts: Kluwer Academic Publishers.

[Matsubara04] Matsubara, E. T. (2004). O algoritmo de aprendizado semi-supervisionado co-training e sua aplicação na rotulação de documentos. Dissertação de Mestrado, ICMC-USP. http: //www.teses.usp.br/teses/disponiveis/55/55134/tde-19082004-092311/.

[Matsubara06] Matsubara, E. T., Monard, M. C., and Prati, R. C. (2006). On the class-distribution labelling-step sensitivity of co-training. In IFIP World Computer Congress - TC12 Artificial Intelligence in Theory and Practice (IFIP AI 2006). in print.

[Metz05] Metz, J. and Monard, M. C. (2005). Clustering hierárquico: uma metodologia para auxiliar na interpretação dos clusters. In V Encontro Nacional de Inteligencia Artificial (ENIA'2005), pages 1170–1173, São Leopoldo (RS). Sociedade Brasileira de Computação (SBC).

[Michalski83] Michalski, R. S., J. G. Carbonell, and T. M. Mitchell (1983). Machine Learning: An Artificial Intelligence Approach. Tioga Publishing Company.

[Michie94] Michie, D., D. J. Spiegelhalter, and C. C. Taylor (Eds.) (1994). Machine Learning, Neural and Statistical Classification. Ellis Horwood.

[Mitchell97] Mitchell, T. M. (1997). Machine Learning. McGraw-Hill.

[Monard03] Monard, M. C. and Baranauskas, J. A. (2003). Indução de Regras e Árvores de Decisão, volume 1 of 1, chapter 5, pages 115–140. Editora Manole, Barueri, SP, Brasil, 1 edition.

[Monard05] Monard, M. C. and Prati, R. C. (2005). Aprendizado de máquina simbólico para mineração de dados. In Castineira, M. I. and Schuhmacher, V. R. N., editors, XIII Escola Regional de Inform´atica (ERI 2005), pages 1–26. WRC´opias.

[Moore65] G. E. Moore, "Cramming more components onto integrated circuits", Electronics, pp. 114-117, April 2005.

[Oliker02] Oliker, L, Li, X, Husbands, P and Biswas, R. Effects of Ordering Strategies and Programming Paradigms on Sparse Matrix Computations, SIAM Review, 44(3): 373-393, 2002.

[Olukotun05] K. Olukotun and L. Hammond, "The Future of Microprocessors", ACM Queue, Vol. 3, No. 7, pp. 26-34, September 2005.

[Paulson05] L. D. Paulson, "Squeezing Supercomputers onto a Chip", IEEE Computer, Vol. 38, No. 1, pp. 21-23, January 2005.

[Pollock99] Pollock, D. S. G. (1999). A Handbook of Time-Series Analysis, Signal Processing and Dynamics (San Diego ed.). New York: Academic Press.

[Prati04a] Prati, R. C., Batista, G. E. A. P. A., and Monard, M. C. (2004a). Class imbalance versus class overlaping: an analysis of a learning system behaviour. In Mexican International Conference on Artificial Intelligence (MICAI'2004), volume 2972 of Lecture Notes in Computer Science, pages 312–321, Mexico City (Mexico). Springer-Verlag.

[Prati04b] Prati, R. C., Batista, G. E. A. P. A., and Monard, M. C. (2004b). Learning with class skews and small disjuncts. In Brazilian Symposium on Artificial Intelligence (SBIA'2004), volume 3171 of Lecture Notes in Artificial Intelligence, pages 296–306, S˜ao Luiz (BR). Springer.

[Prati05] Prati, R. C. and Flach, P. A. (2005). Roccer: An algorithm for rule learning based on ROC analysis. In Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI'2005), pages 823–828, Edinburgh, Scotland, UK.

[Quinlan93] Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann. California.

[Roddick02] Roddick, J. F. and M. Spiliopoulou (2002). A survey of temporal knowledge discovery paradigms and methods. IEEE Transactions on Knowledge and Data Engineering 14 (4), 750–767.

[Rong02] Rong, J., H. Tagaris, A. Lachsz, and M. Jeffrei (2002). Wavelet based feature extraction and multiple classifiers for electricity fraud detection. In 2002 Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES, Volume 3, pp. 2251–2256.

[Saad03] Yousef Saad, Iterative Methods for Sparse Linear Systems, Second Edition, Society for Industrial - Applied Mathematics, 2003.

[Valentim04] E.C. Valentim, L.M. Pessoa, B.Z. Melotti, A.M.P. Valli e L. Catabriga, Comparações entre os métodos GMRES e LCD na implementação do métodos de Newton inexato em problemas de diferenças finitas, 10th Brazilian Congress of Thermal Engineering and Sciences (ENCIT 2004), Rio de Janeiro, RJ, Brasil.

[Valli05] A.M.P. Valli, G.F. Carey and A.L.G.A. Coutinho, Control Strategies for Timestep Selection in Simulation of Incompressible Flows and Coupled Reaction-Convection-Diffusion Processes, International Journal for Numerical Methods in Fluids, v.47, pp. 201-231, 2005

[Weiss98] Weiss, S. M. and N. Indurkhya (1998). Predictive Data Mining: A Practical Guide. San Francisco, CA: Morgan Kaufmann.

[Witten00] Witten, I. H. and E. Frank (2000). Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann.

[Yang04] Yang, K. and C. Shahabi (2004). A pca-based similarity measure for multivariate time series. In MMDB '04: Proceedings of the 2nd ACM international workshop on Multimedia databases, New York, NY, USA, pp. 65–74. ACM Press.

[Yuan04] J.Yuan, G. Golub, R. Plemmons, and W. A. G., Cecílio, Semiconjugate direction methods for real positive definite systems, BIT (http://www-sccm.stanford.edu/urap/pub-tech.html/SCCM-02-02), (accepted), (2003)

[Zomaya96] A.Y.H. Zomaya, Parallel and distributed computing handbook, McGraw-Hill, Inc., 1996.