

1. Proponente

Paulo Bisch – Instituto de Biofísica - UFRJ

2. Título do projeto

GridGene – Pesquisa Genômica no Grid Computacional

3. Objetivos

Geral: Apoiar a análise de genomas e estudos pós-genômicos, agilizando tais pesquisas através do desenvolvimento de software paralelo e distribuído capaz de executar em Grids Computacionais.

Específico: Paralelizar software que permita (i) a comparação de biosseqüências, (ii) data mining em cima dos resultados da comparação de biosseqüências, e (iii) a simulação da dinâmica molecular de proteínas e ácidos nucléicos. Tal software paralelo será projetado para executar em Grids Computacionais, o que possibilita níveis de desempenho muito superiores aos atingidos atualmente, a um custo inferior às soluções disponíveis.

4. Metas

Nossas metas são (i) desenvolvimento de uma ferramenta para viabilizar a execução em paralelo do software BLAST (Basic Local Alignment Search Tool) em Grids Computacionais [ALTSCHUL90,97], (ii) paralelização de uma ferramenta para simulação de dinâmica molecular, (iii) paralelização de algoritmos de data mining como Sequential Patterns [HAN2001] e Clustering [WITTEN2000], com o objetivo de sintetizar os resultados obtidos através da comparação de biosseqüências, e (iv) desenvolvimento de um Sistema Gerente de Banco de Dados para Grids Computacionais, que tornará eficiente a transferência de dados no Grid, viabilizando a execução eficiente das aplicações acima descritas no Grid.

Grids Computacionais

Grids Computacionais são sistemas de suporte a execução de aplicações paralelas que acoplam recursos heterogêneos distribuídos, oferecendo acesso consistente e barato aos recursos, independente de sua posição física. A tecnologia de Grids Computacionais possibilita agregar recursos computacionais variados e dispersos em um único “supercomputador virtual”, grandemente acelerando a execução de várias aplicações paralelas

[FOSTER01]. Grids se tornaram possíveis nos últimos anos devido a grande melhora em desempenho e redução em custo tanto de redes de computadores, quanto de microprocessadores.

Vantagens de grids computacionais sobre plataformas para computação paralela tradicionais incluem (i) integração de recursos computacionais geograficamente distribuídos, (ii) disponibilização de uma plataforma computacional distribuída de alto desempenho, (iii) maximização da utilização de recursos computacionais, e (iv) integração de grupos de pesquisadores.

Grids são construídos como um agrupamento de serviços básicos independentes. Um aspecto essencial dos serviços de Grid é que estes estão disponíveis uniformemente através dos ambientes distribuídos na Grid. Os serviços são agrupados em um sistema integrado, também chamado de middleware. Exemplos de ferramentas atuais de Grid incluem Globus [FOSTER98], Legion [LEWIS96], OpenGrid [CIRNE01], AppLeS [BERMAN97]. Recentemente o termo Data Grid tem sido utilizado como forma de descrever o middleware e os serviços para aplicações de dados intensivos na Grid.

Do ponto de vista da aplicação, o Grid é uma coleção de serviços de middleware que fornecem para as aplicações uma visão uniforme dos recursos distribuídos em conjunto com um mecanismo para agrupá-los em um único sistema. Alguns dos serviços disponibilizados para a aplicação são escalonamento, monitoração de recursos, diretórios de dados globais, autenticação, serviços/bibliotecas de comunicação, etc. Entretanto, do ponto de vista do implementador da Grid, estes serviços resultantes precisam interagir com um grupo de plataformas heterogêneas, o que pode envolver múltiplas camadas da infra-estrutura de comunicação e de computação.

Aplicações Genômicas

Comparação de biosseqüências – BLAST (Basic Local Alignment Search Tool) é uma ferramenta de software que compara biosseqüências a fim de encontrar trechos semelhantes entre elas, sendo atualmente a aplicação mais usada para este fim. A comparação de biosseqüências é fundamental no processo de análise de genomas.

Ferramentas de simulação da dinâmica molecular – Estes sistemas calculam de forma iterativa o movimento de átomos em moléculas, o que é fundamental na determinação da estrutura tridimensional de proteínas e ácidos nucléicos. Existem vários sistemas disponíveis para simulação da dinâmica molecular. Alguns exemplos são: (i) CHARMM [BROOKS83], desenvolvido na universidade de Harvard, (ii) Namd, Universidade de Illinois em Urbana-Champaign, e (iii) Thor, Universidade Federal do Rio de Janeiro. Nosso objetivo é analisar os diversos sistemas disponíveis para determinar qual deles se adapta mais naturalmente ao modelo de computação/comunicação associado a um Grid de computadores.

Data Mining – Devido à grande (e crescente) quantidade de biosseqüências disponíveis, a comparação de biosseqüências gera resultados muito extensos, de difícil uso pelo biólogo. Soluções de data mining extraem tendências, correlações e padrões de grandes volumes de dados, mesmo sem a prévia formulação de hipóteses. Portanto, data mining se apresenta como fundamental no processo de pesquisa genômica, pois possibilita ao biólogo sintetizar a enorme massa de dados produzida pelas ferramentas de comparação de biosseqüências.

5. Metodologia

Em um primeiro momento nos concentraremos em adaptar o BLAST para execução no Grid. Tal decisão reflete a grande urgência em agilizar seqüenciamento genômico, o primeiro passo na análise de genomas e que também obviamente precede estudos pós-genômicos.

Além disso, data mining dos resultados da comparação de biosseqüências requer que um BLAST de alta performance esteja disponível para gerar os dados de entrada para o data mining. Desta forma, o trabalho em BLAST naturalmente precede o trabalho em data mining.

Mais ainda, o BLAST é de paralelização relativamente simples. De fato, a comparação de uma determinada seqüência de consulta pode ser feita de forma paralela por diferentes computadores em diferentes partições do banco de dados de biosseqüências. A maior simplicidade da paralelização do BLAST reforça a decisão de atacá-lo primeiro, pois os resultados estarão disponíveis mais rapidamente, atendendo a necessidades de pesquisadores na área.

Após a geração do BLAST Paralelo (versão do BLAST para execução em clusters de workstations), iniciaremos esforços em data mining e também na simulação da dinâmica molecular.

A seguir, temos uma descrição detalhada de cada um dos subprojetos que compõe este projeto:

Blast

A adaptação do BLAST para execução em Grids Computacionais acontecerá em duas fases. Primeiro, nos concentraremos na paralelização do BLAST voltada para clusters de estações de trabalho. O objetivo é prover ao pesquisador a mesma interface utilizada no BLAST seqüencial. A diferença é que vários processadores serão utilizados, reduzindo assim o tempo de execução da aplicação. Note que a paralelização do BLAST precisa considerar a carga dos processadores disponíveis, de forma a gerar uma boa distribuição de trabalho, evitando assim que processadores lentos e/ou sobrecarregados degradem o tempo de execução do BLAST paralelo. Como resultado desta primeira fase, obteremos uma versão do BLAST que executará em clusters de workstations, já gerando um impacto positivo no dia-a-dia dos pesquisadores.

Na segunda fase, será desenvolvido o Grid BLAST, um sistema BLAST para execução em Grids computacionais. Sistemas de computação *Grid* atualmente necessitam de uma extensiva fase de ajustes finos a fim de apresentar bom desempenho em um ambiente heterogêneo [SPRING98]. Um dos nossos objetivos principais a nível de pesquisa é fazer o escalonamento de recursos transparente no Grid computacional para o usuário final, dispensando a fase de ajustes/adaptação de aplicações necessária em outros sistemas de gerenciamento de recursos heterogêneos. De forma a ter informações mais precisas sobre a carga de trabalho sendo executada nos diferentes recursos computacionais que formam a grade, nós pretendemos utilizar as técnicas de caracterização de carga de trabalho propostas em [SILVA01][SILVA00-1][SILVA00-2] no subsistema de monitoração e previsão. Estas técnicas fazem uso de estimadores bayesianos e elementos de lógica difusa para fazer uma análise qualitativa precisa da carga de trabalho que estará sendo executada em diferentes plataformas. De posse dessas informações e da distribuição dos dados no sistema, o sistema de escalonamento pode tomar uma decisão mais inteligente em relação à alocação de novos processos/sistemas, de forma a maximizar tanto a utilização dos recursos computacionais como o desempenho do sistema como um todo.

No caso da implementação para um Grid Computacional, bancos de dados de biosseqüências que podem chegar a centenas de megabytes deverão estar disponíveis em diferentes plataformas, que podem estar geograficamente distribuídas em função da partição do processamento a ser executado. Portanto, um dos problemas cruciais a ser resolvido em uma implementação voltada para o Grid Computacional é a definição de uma arquitetura de distribuição de dados que minimize o tempo total de execução da aplicação.

Alem disso, a execução de diversos processos BLAST no Grid gera uma grande quantidade de arquivos de saída que pôr sua vez deverão também ser utilizados para futuras análises. Desta forma, o sistema de Grid requer um Sistema de Entrada e Saída de Dados Paralelo/Distribuído que possibilite um acesso global, transparente e eficiente para as bases de dados do sistema pelas aplicações do Grid. Para que os sistemas arquivos de saída do sistema possam vir a ser utilizados para futuras análises de forma eficiente é importante que este sistema seja acoplado a um sistema gerente de banco de dados paralelo/ distribuído. Este sistema devera ser desenvolvido em paralelo com o desenvolvimento da ferramenta para a adaptação para Grids Computacionais do software BLAST.

Para isto pretendemos desenvolver um sistema de armazenamento de dados mais rápido do que o sistema de armazenamento tradicional em disco, que chamaremos de Sistema de Entrada/Saída BLAST. Neste sistema, pretendemos aplicar a experiência desenvolvida em [osthoff et al.] para desenvolver um sistema de Entrada e Saída de dados compatível e/ou integrado com os padrões de armazenamento de dados para Grid atualmente em desenvolvimento tal como o sistema Distributed Parallel Storage System [DPSS] e o Image Server Sistem (ISS)

Um aspecto importante da implementação do Grid Blast consiste da escolha de middleware para Grid a ser utilizado. Middlewares para Grid provêm serviços básicos como autenticação, diretório e controle remoto de tarefas que simplificam o desenvolvimento de aplicações para o Grid. Middlewares para Grid incluem Globus, Legion e o OpenGrid. Quanto do início do desenvolvimento do Grid BLAST, investigaremos as possibilidades disponíveis no momento, para a escolha da mais adequada à nossa aplicação.

Gerenciador de Banco de Dados Paralelo e Distribuído no Grid

Pretendemos investigar as necessidades de um sistema gerente de banco de dados para poder se integrar a um sistema de Grid. Iremos adotar como plataforma de desenvolvimento o Sistema Gerente de Banco de Dados GOA++ (SGBD GOA++) baseado nas experiências anteriores obtidas com a paralelização do SGBD GOA++ [andre2000] [osthoff2000]. Desta forma pretendemos adaptar o conhecimento de paralelismo adquirido na paralelização do SGBD GOA++ atualmente em desenvolvimento na COPPE/UFRJ ao sistema de Grid a ser desenvolvido.

É importante ressaltarmos que é fundamental que se disponha de um software de SGBD com características avançadas para que novas tecnologias de banco de dados sejam avaliadas experimentalmente. Com uma ferramenta destas o desenvolvedor não precisa construir um SGBD inteiro e sim construir seu experimento a ser integrado ao SGBD. Entretanto, se os pesquisadores não têm acesso ao fonte, o único uso que podem dar ao SGBD é o de construção de aplicações sobre o SGBD e não o de desenvolvimento de módulos dentro do SGBD que oferecem as novas tecnologias. Ainda que os fontes estejam disponíveis, se a arquitetura do *software* não é aberta, flexível e documentada, o desenvolvimento de novas tecnologias também fica comprometido.

Atualmente, com exceção do sistema SGBD GOA++ [MAURO 1997], não temos no Brasil um *software* de SGBD aberto, com o fonte disponível. Internacionalmente, existem poucos *softwares* nessa linha, e que não satisfazem as nossas necessidades de desenvolvimento, pois a maioria das bases de genoma está armazenada em formato compatível com o padrão XML, que é uma das formas de armazenamentos permitidos pelo sistema SGBD GOA++.

Simulação de Dinâmica Molecular

Uma vez concluída a adaptação do BLAST para execução em Grids computacionais, focaremos nossos esforços em disponibilizar para o Grid uma ferramenta de simulação de dinâmica molecular. A primeira fase deste esforço consistirá naturalmente na definição de qual ferramenta utilizar. Para tal, investigaremos a estrutura interna das ferramentas CHARMM, Namd e Thor para determinar sua adequação a execução no Grid. Uma vez definida a

ferramenta, passaremos a fase de paralelização, seguida à adaptação ao Grid, tal qual feito com o BLAST.

Aplicações de simulação de dinâmica molecular podem ser classificadas como irregulares e dinâmicas. Irregulares porque o número de operações por iteração pode variar para um mesmo passo, e dinâmicas porque o número de operações pode variar entre passos. Entre as versões paralelas disponíveis atualmente, a que demonstrou ter melhor escalabilidade em um cluster de estações de trabalho e a que faz uma decomposição espacial do conjunto de átomos da molécula. Na decomposição espacial, os átomos são distribuídos em cubos em função da localização espacial. Se necessário, um átomo pode migrar de um cubo para outro. Nos pretendemos investigar a fundo esta questão de modo a maximizar a escalabilidade e o tempo de resposta deste tipo de aplicação tanto em um cluster de workstations como no Grid.

Data Mining

Diversas técnicas de análise de padrões de seqüência de dados e de similaridade entre grupos de dados têm sido desenvolvidas em data mining. Em consequência, data mining torna-se uma tecnologia potencialmente poderosa, podendo contribuir substancialmente para a análise genômica de diversas maneiras. Comentaremos brevemente duas técnicas: Sequential Patterns e Clustering.

Um dos mais importantes problemas em pesquisa genética é a busca de similaridade e comparação entre seqüências de DNA. Seqüências de genes isolados de tecidos orgânicos saudáveis e doentes podem ser comparados para identificar diferenças críticas entre as duas classes de genes. Por exemplo, seqüências ocorrendo mais freqüentemente em amostras doentes que em amostras sãs podem indicar os fatores genéticos da doença. A técnica de data mining conhecida por Sequential Pattern Analysis [HAN2001] pode ajudar na análise de similaridade e diferença de seqüências genéticas.

Uma doença pode ser o resultado de uma combinação de genes atuando juntos. Técnicas de data mining baseadas em análise de grupos de dados — Clustering [WITTEN2000] — podem ajudar na descoberta de grupos de genes e no estudo das interações e relações entre eles.

Nossa pesquisa se concentrará no desenvolvimento de versões paralelas de algoritmos de data mining para Sequential Pattern Analysis e Clustering, e sua aplicação à análise de dados genômicos.

6. Justificativa

Sistemas de comparação de biosseqüências, data mining dos resultados obtidos em tal comparação, e ferramentas de simulação dinâmica de moléculas são computacionalmente intensivos e fundamentais para a pesquisa genômica. Uma vez que as características destes sistemas os tornam passíveis de execução em Grid computacionais, este projeto colocará o Brasil na liderança em processamento genômico, tanto no que diz respeito ao desempenho absoluto dos sistemas, quando ao custo/desempenho dos mesmos.

O BLAST é computacionalmente intensivo porque lida com quantidades enormes de dados. Como tais dados tem apresentado uma taxa de crescimento exponencial, soluções eficientes para a execução do BLAST são necessárias para manutenção do forte ritmo de pesquisa genômica que temos presenciado nos últimos anos.

Algoritmos de data mining são igualmente computacionalmente intensivos porque lidam com bancos de dados volumosos, como os bancos genéticos. Algoritmos de data mining têm a vantagem de ser `escaláveis`, i.e. seus tempo de execução tendem a crescer linearmente com o tamanho do banco de dados. Entretanto, os bancos de dados genéticos tendem a crescer exponencialmente, portanto aumentando exponencialmente a demanda computacional para algoritmos de data mining usados sobre dados genéticos. Daí a importância de paralelizar os algoritmos de data mining. Com as versões paralelas desses algoritmos que serão desenvolvidas, pode-se esperar um alto desempenho das atividades de data mining sobre bancos de dados genéticos.

Ferramentas de simulação da dinâmica molecular, por sua vez, calculam de forma iterativa a energia potencial total, forcas e coordenadas para cada átomo do sistema a cada passo de simulação. Devido à alta freqüência de vibração das ligações entre os átomos, o passo de simulação pode chegar a um femtosegundo. A ordem de grandeza do tempo de simulação pode ser nanosegundos ou maior. Além disso, moléculas de interesse podem ter centenas ou milhares de átomos, o que caracteriza esta aplicação como computacionalmente intensiva. É possível particionar o conjunto de átomos de forma a limitar a comunicação entre as diversas partições, o que torna esta aplicação adequada para execução no Grid.

Além da importância das aplicações alvo deste projeto, a utilização de tecnologia de Computação em Grids é também de grande importância para o país. A continuada evolução em Redes e Microprocessadores indica que Grids Computacionais serão a plataforma dominante para Computação de Alta-Performance em um futuro próximo.

7. Indicadores de Avaliação do Projeto

Uma vez que este projeto se propõe ao desenvolvimento de soluções de software, o acompanhamento do mesmo naturalmente se dará pela disponibilização para comunidade científica dos sistemas aqui propostos. Tal disponibilização será feita através da Web, para que o maior número possível de pesquisadores seja beneficiados. A Tabela 1 mostra o cronograma de disponibilização dos softwares descritos na Sessão 5

Software	Mês											
	2	4	6	8	10	12	14	16	18	20	22	24
BLAST Paralelo	X	X	X	X								
Grid BLAST					X	X	X	X	X	X		
MD Paralelo					X	X	X	X				
Grid MD									X	X	X	X
Data Mining												
Parallel Sequential Pattern Algorithm					X	X	X	X				
Parallel Clustering Algorithm									X	X	X	X
E/S Blast e Grid SGBD	X	X	X	X	X	X	X	X	X	X	X	X

Tabela 1 – Cronograma de Disponibilização de Software

8. Infra-estrutura Física e Competência Existente

Este projeto vai utilizar como plataforma de teste os clusters de estações de trabalho/máquinas paralelas disponíveis nas instituições participantes, além das máquinas clusters que estão sendo pedidos especificamente para o projeto. A equipe de pesquisadores com a respectiva área de atuação está detalhada na tabela a seguir:

Pesquisador	Instituição	Área
-------------	-------------	------

Fabrcio Silva	LNCC	Grid Computing
Carla Osthoff	LNCC	Grid Computing
Walfredo Cirne	UFPB/Campina Grande	Grid Computing
Francisco Brasileiro	UFPB/Campina Grande	Grid Computing
Marcus Sampaio	UFPB/Campina Grande	Data Mining
Paulo Ferreira	UFRJ/Bioquímica	Aplicação
Orlando Martins	UFRJ/Bioquímica	Aplicação
Paulo Bisch	UFRJ/Biofísica	Aplicação
Marta Matoso	UFRJ/COPPE	Banco de dados
Cláudio Amorim(?)	UFRJ/COPPE	Arquiteturas
Nelson Ebecken	UFRJ/COPPE	Data Mining
Sergio Lifschitz (?)	PUC/RJ	Banco de dados
Wim Degrave	Fiocruz	Aplicação

9. Articulações Interinstitucionais

Este projeto permitira a colaboração e integração de pesquisadores das seguintes instituições:

- LNCC
- UFRJ
- UFPB
- FioCruz
- PUC/RJ

10. Orçamento

Equipamento

<i>Quant</i>	<i>Descrição</i>	<i>Preço</i>	<i>Total</i>
	30 PC Pentium III 1GHz 512MB RAM 60 GB Disco	4000,00	120000,00

4 Switch Ethernet 10/100	1000,00	4000,00
2 Roteadores CISCO 2500 (?)	4000,00	8000,00
6 Estações de trabalho	8000,00	48000,00
		180000,00

Viagens

<i>Quant</i>	<i>Descrição</i>	<i>Preço</i>	<i>Total</i>
14	Viagem Internacional	5500,00	77000,00
16	Viagem Nacional	1400,00	22400,00
10	Viagem Nacional	1400,00	14000,00
			113400,00

Serviços de Terceiros

<i>Quant</i>	<i>Descrição</i>	<i>Preço</i>	<i>Total</i>
1	Lab UFPB	10000,00	10000,00
1	Lab LNCC	10000,00	10000,00
1	Lab Bioquímica	10000,00	10000,00
1	Lab COPPE	10000,00	10000,00
1	Lab Biofísica	10000,00	10000,00
100	Consultor/hora	100,00	10000,00
			60000,00

Custeio

<i>Quant</i>	<i>Descrição</i>	<i>Preço</i>	<i>Total</i>
1	Lab UFPB	5000,00	5000,00
1	Lab LNCC	5000,00	5000,00

1 Lab COPPE	5000,00	5000,00
1 Lab UFRJ/Biofísica	5000,00	5000,00
1 Lab UFRJ/Bioquímica	5000,00	5000,00
		25000,00
Total Geral		378400,00

11. Viabilidade Técnica

Nosso objetivo é dividir os subprojetos descritos anteriormente entre os membros da equipe, em função da sua área de atuação. Em alguns casos, alunos de graduação/pós-graduação também irão se envolver nos subprojetos. Nossa expectativa é de terminar o projeto em 24 meses, como detalhado na seção **Indicadores de Avaliação dos Projetos**.

12. Propriedade Intelectual

A intenção é distribuir o software desenvolvido livremente na Web. Entretanto, para evitar que outros venham a obter vantagens comerciais sem nossa autorização, patentearmos tecnologias importantes que venham a ser geradas durante o projeto.

13. Referencias

[ALTSCHUL90] S. F. Altschul et al. **A Basic Local Alignment Search Tool**. J.of Molecular Biology 215, pp 403-410, 1990

[ALTSCHUL97] S. F. Altschul et al. **Gapped Blast and Psi Blast: a new generation of Protein Database Search Programs**. Nucleic Acids Research 25(17), pp 3389-3402, 1997.

[BERMAN97] Fran Berman and Rich Wolski. **The AppLeS Project: A Status Report** , Proceedings of the 8th NEC Research Symposium, Berlim, Germany, May 1997.

[BROOKS83] B. R. Brooks et al., **CHARMM: A Program for Macromolecular Energy, Minimization, and Dynamics Calculations**, J. Comp. Chem. 4, 187-217, 1983

[CIRNE01] Walfredo Cirne and Keith Marzullo. **Open Grid: A User-Centric Approach for Grid Computing**. Proceedings of the 13th Symposium on Computer Architecture and High Performance Computing, September 2001

[FOSTER98] Ian Foster and Carl Kesselman, **The Globus Project: A Status Report**, Proceedings Heterogeneous Computing Workshop, IEEE Press, 1998

[FOSTER01] Ian Foster, Carl Kesselman and Steven Tuecke. **The Anatomy of the Grid – Enabling Scalable Virtual Organizations**. To appear in the International Journal of Supercomputing Applications

[HAN2001] Jiawei Han, Micheline Kamber. **Data Mining: Concepts and Techniques**. Morgan Kaufmann Publishers, 2001.

[LEWIS96] Michael J. Lewis, Andrew Grimshaw **The Core Legion Object Model**. Proceedings of the Fifth IEEE International Symposium on High Performance Distributed Computing, IEEE Computer Society Press, Los Alamitos, California, August 1996.

[MAURO1997] Mauro, R.C. Mattoso. M.L.Q. et al. **“GOA++: Tecnologia, implementação e extensões aos serviços de gerência de objetos”**, Anais do XII Simpósio Brasileiro de Banco de Dados, Fortaleza, outubro, 1997, pp.272-286.

[OSTHOFF 1999] Osthoff C.,R. Bianchini, M. Mattoso,C. Seidel and C.Amorim. **“Explorando Conceitos e Mecanismos de Memória Compartilhada Distribuída em E/S Paralela”**. XV Simpósio Brasileiro de Arquitetura de Computadores- Computação de Alto Desempenho, Natal, outubro, 1999.

[OSTHOFF 2000] Osthoff C. **Proposta e Avaliação de Mecanismos de Software de Memória Compartilhada Distribuída para E/S Paralela**, Tese de D.Sc., COPPE/UFRJ, RJ Brasil,2000.

[OSTHOFF 2000a] Osthoff C.,Seidel C., Bianchini R.,Mattoso M e Amorim C., **“Evaluating Cache Coherence in DSMIO System “**, SBAC-PAD'2000 Symposium on Computer Architectures and High Performance Computing. – São Pedro 2000 .

[OSTHOFF 2000b] Osthoff C., Mattoso M.,Seidel C.,Bianchini R. e Amorim C., **“O Algoritmo de coerência de cache de disco DSMIO”**, XV Brazilian Symposium on Databases (SBBD2000), ACM SIGMOD,2000.

[SILVA01] Fabricio Silva and Isaac D. Scherson, **Efficient Parallel Job Scheduling Using Gang Service**, To appear in the International Journal of Foundations of Computer Science.(June 2001)

[SILVA00-1] Fabricio Silva and Isaac D. Scherson, **Improving Throughput and Utilization in Parallel Machines Through Concurrent Gang Scheduling**, Proceedings of the IEEE International Parallel and Distributed Processing Symposium 2000, Cancun, Mexico, May 2000.

[SILVA00-2] Fabricio Silva and Isaac D. Scherson, **Improving Parallel Job Scheduling Using Runtime Measurements**, Job Scheduling Strategies for Parallel Processing, ed.: Dror G. Feitelson and Larry Rudolph, Lecture Notes on Computer Science 1911

[SPRING98] Neil Spring and Rich Wolski, **Application Level Scheduling of Gene Sequence Comparison on Metacomputers**, Proceedings of of the 12th ACM International Conference on Supercomputing, Melbourne, Australia, July, 1998.

[WITTEN2000] Ian H. Witten, Eibe Frank. **Data Mining: Practical Machine Learning Tools**. Morgan Kaufmann Publishers, 2000.