Execução Paralela do Algoritmo Genético Aplicado na Estimação de Parâmetros em Cluster Beowulf

Tiago Luiz Ferreira[1], Gideon V. Leandro[1], Robinson F. de Camargo[1], Sergio L. Dill[1], Edson L. Padoin[1]

UNIJUÍ – Universidade Regional do Noroeste do Estado do Rio Grande do Sul
[1]DeTec – Departamento de Tecnologia – Ijuí, RS

{tiago, gede, dill, padoin, robinson.camargo}@unijui.edu.br

1. Introdução

Os modelos matemáticos que simulam situações reais do sistema são bastante estudados e aplicados ao processamento de alto desempenho, por demandarem alto poder de processamento e pela possibilidade das equações possuírem um alto grau de paralelismo. Pela flexibilidade, simplicidade de implementação será paralelizado nesse trabalho um algoritmo genético cuja aplicação está difundida nas mais diversas áreas da ciência.

2. Definição do Problema

Algoritmos Genéticos (AG) são algoritmos de procura baseados nas mecânicas de seleção e genética natural onde os indivíduos mais fortes e aptos têm probabilidade maior de sobreviver e evoluir. A partir de uma amostra da população procura-se evoluir conforme o desempenho de seus indivíduos, privilegiando os mais aptos [MICHALEWICZ, 1999].

A estrutura básica de um AG é mostrada na Figura 1 [GOLDBERG, 1989].

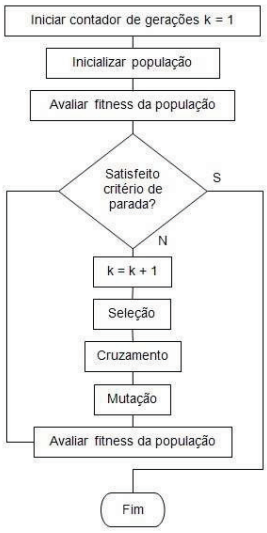

Figura 1 – Fluxograma de um Algoritmo Genético Simples.

- Inicialização da População - normalmente a população inicial é gerada aleatoriamente, procurando cobrir todo o espaço de busca do problema.

- Função de Avaliação – a função de avaliação, função objetivo ou função custo é quem determina o grau de aptidão (*fitness*) de um cromossomo (possível solução). Nos AG todos os cromossomos, em cada geração, são submetidos à função objetivo para determinar o valor de *fitness* de cada indivíduo que definirá a probabilidade de seleção e reprodução para a geração seguinte.

- Operadores Genéticos – São responsáveis pelas mudanças que ocorrem na população, ao passar de uma geração para outra, mudando algumas de suas características, os quais são: Seleção, Cruzamento (aritmético ou aritmético com os extremos) e Mutação (Uniforme ou não-uniforme).

- Critério de Parada - os AG são métodos iterativos de busca estocástica e, como tal, necessitam de um critério de parada para informar ao programa o momento de parar. Esse critério pode ser o número máximo de iterações ou gerações da população, ou ainda, verificar a evolução do melhor indivíduo após determinado número de gerações.

Nesse algoritmo, a família inicial de cromossomos é gerada aleatoriamente e em seguida submetida à função de avaliação para determinar a aptidão ou fitness de cada cromossomo. Depois é verificado se foi atingido o critério de parada, que normalmente está relacionado a precisão da resposta e ao número máximo de gerações. Caso não seja satisfeito esse critério, entra-se no laço onde ocorrem o incremento das gerações e a aplicação dos operadores genéticos de reprodução, cruzamento e mutação para, em seguida, os cromossomos serem novamente submetidos à função de avaliação[GOMES, 2005].

O ponto de partida nos AG é a representação de cada possível solução no espaço de busca (universo de soluções) como uma seqüência de símbolos pertencentes a um alfabeto finito $U = \{s_1, s_2, ..., s_n\}$. A cada seqüência de *s* corresponde a um cromossomo e cada elemento de *s* equivale a um gene [GOMES, 2005].

3. Ambiente de Execução Paralela

O cluster computacional foi constituído no padrão Beowulf com computadores do tipo PC interligados por um switch central Fast Ethernet 100baseT Full Duplex, onde uma das máquinas é o nó mestre e os demais são os nós escravos dedicados a execução das tarefas solicitadas pelo nó mestre.

O cluster é formado por 19 máquinas homogêneas com processadores Intel® Core™ 2 Duo E4600 2.40GHz com 2Mb de memória cache compartilhada, 2 GB de memória principal e adaptador de rede Gigabit Ethernet 1000baseT Full Duplex. Utilizou-se o sistema operacional GNU/Linux distribuição OpenSuse 10.2, com kernel versão 2.6.18.2-34-default x86, compilador GCC na versão 4.1.2 e o middleware MPICH na versão 1.2.7p1.

4. Implementação Paralela

A versão paralela do algoritmo foi desenvolvida na linguagem de programação C com base no algoritmo desenvolvido em MATLAB por [GOMES, 2005] e fazendo uso da biblioteca de troca de mensagens MPI. Basicamente utilizou-se matrizes alocadas em tempo de execução do tipo double ** para armazenar as informações, sendo necessário dessa forma a conversão das informações de double ** para double * para serem enviadas através das funções de troca de mensagens.

A execução do AG no cluster começa pelo nó mestre com a carga dos parâmetros de entrada, a construção da matriz de repressores e a definição do valor de normalização. Após definido esses parâmetros, as informações são replicadas para cada nó do cluster através de funções de comunicação coletivas.

Na seqüência cada nó do cluster gera aleatoriamente uma população inicial e a avalia, concentrando ao final desse processo as informações no nó mestre, para que nele sejam selecionados somente os indivíduos mais aptos. A classificação do indivíduo quanto à aptidão é feita através de seu erro com relação aos dados de entrada, então o indivíduo que tiver menor erro esse será mais apto a continuar na evolução.

Essa população selecionada então é dividida para que cada nó do cluster dentro do laço das gerações aplique os operadores genéticos sobre os seus indivíduos. Cada indivíduo ou cromossomo representa uma linha na matriz da população, dessa forma cada nó será responsável por processar um determinado número de linhas da matriz.

Ao final de cada iteração do laço das gerações, cada processo avalia sua população de cromossomos e envia para o nó mestre seu melhor indivíduo. O nó mestre novamente avalia dentre todos os resultados recebidos qual é o melhor cromossomo e de posse desse o armazena como valor daquela geração, que mais tarde ao fim do laço das gerações todos os dados armazenados formarão o resultado do algoritmo genético.

Caso o melhor indivíduo dessa geração seja pior que o da anterior, então o cromossomo da geração anterior é copiado para a atual e o pior individuo é eliminada da população sendo substituído pelo da geração anterior.

Dessa forma ao fim do laço das gerações tem-se armazenados os melhores cromossomos evoluídos ao longo do tempo, formando a saída estimada do sistema que dever ser o mais próximo possível da saída real do modelo.

5. Resultados

A análise de desempenho da implementação paralela do AG foi realizada através da execução paralela em diferentes números de nodos processadores, buscando obter a melhor relação de divisão de trabalho entre os nodos do cluster. Os tempos representam uma média de cinco execuções e correspondem a configuração do AG com um população inicial de 2000 indivíduos e 200 gerações.

No Gráfico 1, 2 e 3 são apresentados respectivamente os tempos de execução em segundos, o speed-up e a eficiência obtidos variando-se o número de nodos do cluster.

Analisando os tempos de execução do algoritmo paralelo apresentados nos gráficos, observa-se um ganho significativo de tempo até as execuções com 10 nós. A partir de 11 nós o tempo começa a se estabilizar havendo pouca diferença com acréscimo de mais nós. Devido à limitação de 19 máquinas que compunham o cluster e ao tempo de execução do algoritmo ter diminuído até essa quantidade de nodos, não foi possível definir com quantos nós o tempo de execução começaria a aumentar devido ao tempo gasto com a comunicação.

O menor tempo de execução observado é de 17,5 segundos na utilização de 19 nodos. Tomando como base o tempo do algoritmo seqüencial que é de 253,5 segundo, percebemos um uma redução de tempo muito próximo à décima quarta parte do tempo seqüencial, ou seja, um *speed-up* de 14,4 e uma *eficiência* de 76%.

O algoritmo se mostrou mais eficiente nas execuções em 8 e 9 nodos, atingindo um valor ideal de 100%, o que quer dizer que o algoritmo nessas situações utilizou

100% do processamento disponível em cada nó explorando ao máximo o paralelismo do algoritmo.

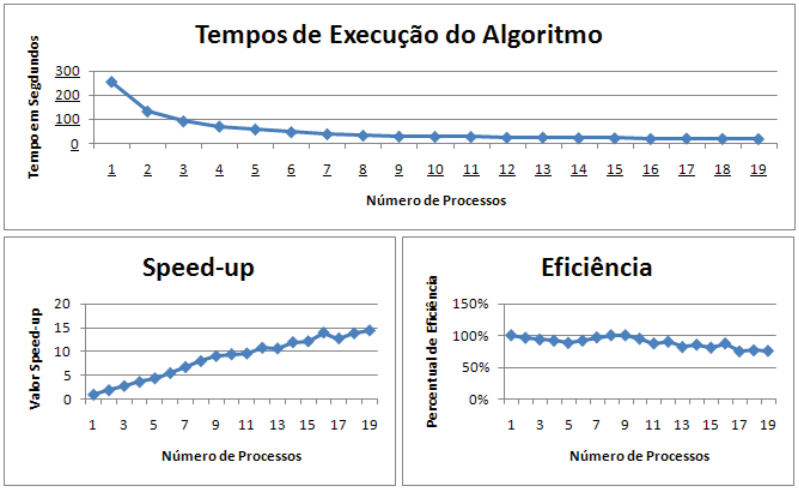

Gráfico 1 – Tempos; Gráfico 2 – Speed-up; Gráfico 3 – Eficiência.

6. Conclusão e Trabalhos Futuros

Através dos experimentos realizados comprova-se que o uso de cluster de baixo custo pode ser uma alternativa viável para a resolução de problemas que necessitam de uma resposta com maior precisão em menos tempo. Observou-se também a alta granularidade dos algoritmos genéticos quando paralelizados com uso da biblioteca de troca de mensagem *MPI*. A comprovação apresenta-se na nos tempos que em alguns casos o algoritmo chegou a um grau ótimo de paralelização e utilização dos nodos.

Como trabalhos futuros têm-se a possibilidade de ajustar o AG para que possa ser executado em Grids Computacionais, que dão a possibilidade de submeter Jobs do algoritmo genético através da web para serem processados em máquinas geograficamente distribuídas.

Referências

GOLDBERG, David E. **Genetic algorithms in search, optimization, and machine learning**. Addison Wesley Longman, Inc, 1989.

GOMES, Moacir Carlos. **Identificação de sistemas utilizando algoritmos genéticos para estimação de parâmetros**. Dissertação de Mestrado em Modelagem Matemática – Universidade Regional do Noroeste do Estado do Rio Grande do Sul, Ijuí, 2005.

MICHALEWICZ, Z. **Genetic Algorithms + Data Structures = Evolution Programs**. Springer Verlag, 1999.