

# ORIENTAÇÃO A OBJETOS E PROGRAMAÇÃO PARALELA APLICADOS A PROBLEMAS DE COMPUTAÇÃO CIENTÍFICA

Flavio Henrique do Nascimento Moreira (Bolsista PIBIC/CNPq)

Aluno da UNITAU

Orientador: Airam Jonatas Preto, Pesquisador Titular, LAC - INPE

Historicamente, as aplicações científicas, principalmente as que estão voltadas ao processamento de alto desempenho, têm sido desenvolvidas em linguagem Fortran. Isso levou ao desenvolvimento de compiladores Fortran extremamente eficientes para a otimização de código.

Por outro lado o surgimento da metodologia orientada a objetos (OO) tornou possível o desenvolvimento de programas com facilidade para o tratamento de entidades num alto nível de abstração, que tem sido uma motivação crescente para a utilização da programação OO na Computação Científica. Considere-se ainda o fato do paradigma OO ser baseado no conceito de objetos que interagem enviando e recebendo mensagens: esta característica permite que um conjunto de objetos seja mapeado para um grupo de processadores, para que seja explorado o paralelismo nas suas computações.

Assim, o objetivo desse projeto foi implementar uma classe Array de duas dimensões, utilizando a linguagem C++, que encapsulasse as operações matemáticas efetuadas em matrizes (como soma, subtração, multiplicação por escalar e por outra matriz), de forma a conseguir operações paralelas transparentes durante a programação.

Para isso a abordagem utilizada foi o Paralelismo de Dados: essa é aplicada a um grande conjunto de dados que pode ser dividido entre os vários processadores de um sistema paralelo, para serem processados por rotinas iguais sobre todos os processadores.

Nesse caso específico a implementação foi feita sobre uma estrutura de computação paralela construída a partir de estações Intel x86 rodando Linux com kernel 2.2, utilizando uma biblioteca para comunicação por mensagens, MPI (message passing interface); sua implementação específica para OO, conhecida como OOMPI para C++.

A implementação dessa classe foi feita de forma que, ao instanciar-se essa classe, os dados deste objeto fossem espelhados sobre todos os processadores e distribuídos conforme uma diretiva interna definida por um método dessa classe. Ao efetuar-se operações com esses objetos cada processador atua apenas sobre os dados que lhe são concernentes. Isso é ilustrado na figura abaixo:

