**Henrique de S. Q. dos Santos, NUSP 10819029**

**Witor M. A. de Oliveira, NUSP 10692190**

**Gustavo T. Mastrobuono, NUSP**

**Atividade 1 - Projeto PCAM**

1. Introdução

De acordo com as notas de aulas disponibilizadas pelo professor Paulo S. L. de Souza, bem como o material da disciplina por ele ministrada, definimos o projeto PCAM da nossa versão do algoritmo de busca concorrente pelo maior elemento em um vetor.

1. Projeto PCAM
   1. P - Particionamento

Essa etapa define o “particionamento do problema em tarefas concorrentes”. O nosso programa era dividido nas etapas de preenchimento do vetor e busca pelo maior elemento deste vetor. Assim, nessas duas partes, utilizamos a programação concorrente para que o processo fosse agilizado. Pras duas etapas - preenchimento e busca - utilizamos a mesma forma de **particionamento por bloco de dados**: considerando o resto da operação de divisão entre o número de elementos do vetor e o número de threads utilizadas no programa, quando o resto da operação era zero efetuamos a divisão do vetor em T blocos iguais, onde T é o número de Threads utilizadas no programa, e dividimos o vetor em 2 blocos iguais quando o resto era diferente de zero. Tanto no preenchimento do vetor, quanto na busca pelo maior elemento, cada thread ficou unicamente responsável por um bloco desse vetor, isto é, uma thread sempre iria trabalhar em apenas um dos blocos.

* 1. C - Comunicação

Essa etapa é responsável pela “definição da interação das tarefas: comunicação e sincronização”. Para as duas operações supracitadas, não houve uma comunicação. Porém, houve sincronização entre as etapas: o processo de busca pelo maior elemento no vetor era iniciado apenas quando todas as threads finalizassem o preenchimento do mesmo.

* 1. A - Aglomeração

Nessa fase, devemos “aglomerar tarefas concorrentes em processos concorrentes”. A estratégia de aglomeração que utilizamos foi atribuir a cada thread uma tarefa. No nosso caso, cada thread ficou responsável por preencher ou buscar pelo maior elemento em cada bloco do vetor passado na criação da thread. E o número de threads criadas foram de acordo com o número de cores do computador.

* 1. M - Mapeamento