



PROGRAMAÇÃO CONCORRENTE E DISTRIBUÍDA
Rosangela Miyeko Shigenari 92334 Integral

Converta o programa serial para calc. condução calor 1D (Arquivo: "fcts.c") para executar em GPU.

OBS: Preocupe-se em converter apenas o trecho responsável pelo cálculo das linhas 41 até 46:

```
for (i=1; i<n; i++) {  
    u[i]=u_prev[i]+kappa*dt/(dx*dx)*(u_prev[i-1]-2*u_prev[i]+u_prev[i+1]);  
    x += dx;  
}  
u[0] = u[n] = 0.; /* forca condição de contorno */  
tmp = u_prev; u_prev = u; u = tmp; /* troca entre ponteiros */
```

Faça testes com diferentes configurações de grade e blocos variando a quantidade de threads por bloco da seguinte forma: 512, 256, 128, 64.

Verifique em que situação o algoritmo melhorou o desempenho.

A tabela abaixo mostra os valores, que relacionam o número de blocos, o número de threads e o respectivo tempo de execução do programa (em anexo em fcts.cu).

Número de blocos	Número de Threads	Tempo (ms) de execução
1562	64	299,5
781	128	251,2
390	256	248, 1
195	512	280,5

Conclusão:

Como pode se observar na tabela, o programa em cuda possui melhor comportamento quando o número de threads é maior para cada bloco, sendo assim, utilizar menor número. Este fato se dá porque os blocos não podem se comunicar, então utilizar um número muito grande de blocos não é a melhor alternativa. E a threads ajudam a não realizar leituras desnecessárias, devido a este fato, os programas com maiores número de threads rodam mais rápido.