



**Universidade:  
presente!**

**UFRGS**  
PROPEAQ



**XXXI SIC**

21. 25. OUTUBRO • CAMPUS DO VALE

<b>Evento</b>	Salão UFRGS 2019: SIC - XXXI SALÃO DE INICIAÇÃO CIENTÍFICA DA UFRGS
<b>Ano</b>	2019
<b>Local</b>	Campus do Vale - UFRGS
<b>Título</b>	Exploração de espaço de projeto para otimização de aplicações paralelas
<b>Autor</b>	LETÍCIA DOS SANTOS
<b>Orientador</b>	ANTONIO CARLOS SCHNEIDER BECK FILHO

## **Exploração de espaço de projeto para otimização de aplicações paralelas**

**Aluna:** Letícia dos Santos

**Orientador:** Antonio Carlos Schneider Beck Filho

UFRGS - Instituto de Informática

O uso de programação paralela permite dividir as tarefas de uma aplicação entre os múltiplos núcleos disponíveis nos sistemas atuais, de forma que estas sejam executadas simultaneamente em um tempo de execução reduzido. De maneira geral, há dois modelos de programação paralela, sendo que a principal diferença entre eles é a forma de comunicação entre as threads/processos: memória compartilhada e troca de mensagens. No primeiro, o mais adequado é utilizar a comunicação por variáveis compartilhadas. Entre as interfaces de programação paralela existentes, a mais utilizada neste modelo é OpenMP. Na segunda forma de comunicação, geralmente usada em sistemas com memória distribuída (i.e. espaços de endereçamento distintos), a comunicação precisa ser realizada por meio de troca de mensagens. Neste caso, MPI é a interface mais usada. Considerando que os sistemas de HPC (high-performance computing) atuais são compostos por múltiplos processadores com múltiplos núcleos, combinar os dois modelos, usando uma abordagem híbrida OpenMP/MPI, possibilita utilizar todos os recursos disponíveis para obter o melhor desempenho. Em contrapartida, a potência consumida poderá maior, podendo gerar um aumento no consumo total de energia.

Para entender como os modelos de programação paralela funcionam, foram paralelizados dois programas: (i) cálculo do valor de pi por meio de uma integral; e (ii) multiplicação de matrizes. Esses algoritmos foram escolhidos porque integrais e multiplicação de matrizes são amplamente utilizadas em diversos domínios de aplicação. Além disso, o cálculo de pi é uma aplicação com intensa carga de trabalho para o processador, enquanto a multiplicação de matrizes realiza muitas consultas e escritas na memória. Os experimentos foram realizados inicialmente em um sistema com processador quad-core Intel Core i5-2400 com 3.10GHz de frequência. As seguintes configurações foram avaliadas: 4 threads (versão OpenMP) e 4 processos (versão MPI). Durante os experimentos, foram coletados dados de tempo de execução. A versão com OpenMP foi 2,7% mais rápida para o cálculo de pi e 53,8% mais rápida para multiplicação de matrizes. Já era esperado que o OpenMP fosse melhor, pois a máquina utilizada possui memória compartilhada. A grande diferença entre a melhora dos algoritmos se deve a comunicação entre os processos ou threads. Na multiplicação de matriz há mais comunicação, tornando o overhead do MPI mais visível, enquanto que no OpenMP a comunicação entre threads é mais barata, pois ocorre por meio da memória compartilhada.

A versão híbrida OpenMP/MPI dos algoritmos está sendo implementada. Os experimentos com a versão híbrida serão realizados em um cluster com 2 nós computacionais, onde cada nó é composto por 2 processadores octa-core Intel Xeon E5-2630 (Sandy Bridge) com hyper-threading, totalizando 32 threads por nó. O consumo de energia também será medido para que possa ser identificado a configuração da quantidade de processos e threads que ofereça o melhor balanceamento entre desempenho e consumo de energia.