

## Task Stealing em Runtimes para Task Parallelism

Guilherme Covre Pupio\*, Divino César Soares Lucas, Guido De Araujo Costa.

### Resumo

Este projeto procurou encontrar novas heurísticas para adquirir melhor speed-up em programas paralelizados, a partir da análise de diversos benchmarks em runtimes de OpenMP 4.0. Foram estudados parâmetros como dependência de entrada ou saída, bem como entrada e saída das tasks, dentre outras características. Os resultados dos experimentos e análise destes parâmetros serão usados para projetar novas heurísticas para task stealing, as quais permitam aumentar o speed-up da execução dos programas paralelizados.

### Palavras-chave:

Task parallelism, Runtimes, Parallelism.

### Introdução

Em trabalho anterior desenvolvido no Laboratório de Sistemas de Computação (LSC) do IC-UNICAMP, foram realizados alguns experimentos preliminares (figura 1), pelo aluno de Doutorado Cesar Lucas, que mostram o potencial de speed-up que se pode atingir aplicando-se task stealing quando executando no runtime MTSP (Multi-core Task Scheduling Platform) desenvolvido naquele laboratório. Estes experimentos preliminares indicam que existe um ótimo potencial de mais ganhos de speed-up.

Dessa forma, o projeto teve por intuito ampliar e buscar novas heurísticas para adquirir melhor speed-up em programas paralelizados, tendo como base, a análise de variados benchmarks em runtimes de OpenMP 4.0. Foram estudados parâmetros como dependência de entrada ou saída, bem como entrada e saída das tasks, dentre outros parâmetros. Os resultados obtidos serão utilizados para projetar novas heurísticas para task stealing que permitam aumentar o speed-up da execução dos programas paralelizados.

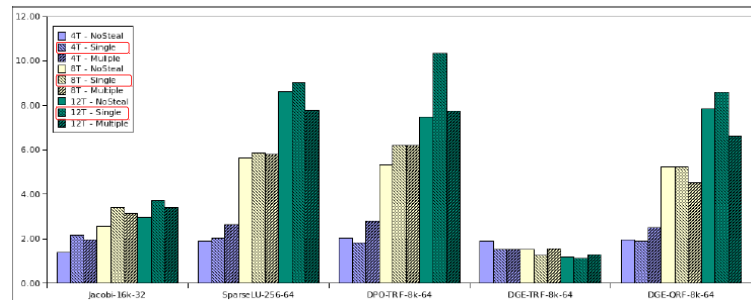


Figura 1. Melhoria da speed-up ao executar benchmarks em formas distintas: sem task stealing, apenas uma run queue e multiplas run queues, ambos casos avaliados em 4, 8 e 12 Threads.

Tabela 1. Tempo médio de execução das tasks de um benchmark, para um dado runtime. Média tomada para a versão serial e paralela do mesmo benchmark.

	task name	Serial (us)	Parallel (us)	Overhead
Benchmark	task 1	62	106	72,39%
	task 2	92,126	123,821	34,40%
	task 3	93,016	134,459	44,55%

### Resultados e Discussão

O experimento foi realizado em uma máquina HP, com processador Intel i5, quad-core, RAM de 4GB, clock de 2800 MHz e sistema operacional Ubuntu 16.04.

Ao analisar os tempos médios de execução e tempo médio do agendamento de cada tarefa, para cada task presente em uma série de benchmarks, notou-se que as tasks, quando executadas em serial, apresentam menor tempo médio de execução que as mesmas tasks executadas em paralelo. A tabela 1 mostra um pequeno exemplo de um dos resultado obtidos.

Durante o decorrer do projeto, notou-se que o padrão se mantinha para ambos os runtimes estudados. A última coluna da tabela 1, indica a proporção da diferença entre os tempos de execução de cada task em sua respectiva versão.

### Conclusão

Assim, tem-se que a execução de programas paralelizados, embora seja, no todo, mais veloz em relação a versão serial, o tempo médio de execução de cada task paralelizada é maior que em sua respectiva versão serial, para ambos os runtimes analisados.

Por fim, nota-se que, no geral, a execução dos benchmarks em arquiteturas Single (como indicado na figura 1), que utilizam apenas uma run queue, levam a maiores speed-ups. Também é notável a influência da quantidade de threads no speed-up final, pois este foi proporcional ao número de threads.

### Agradecimentos

Agradeço ao Divino César, por seu grande auxílio e serenidade ao longo do projeto.