

Trabalho OpenMP - Algoritmo Monte Carlo

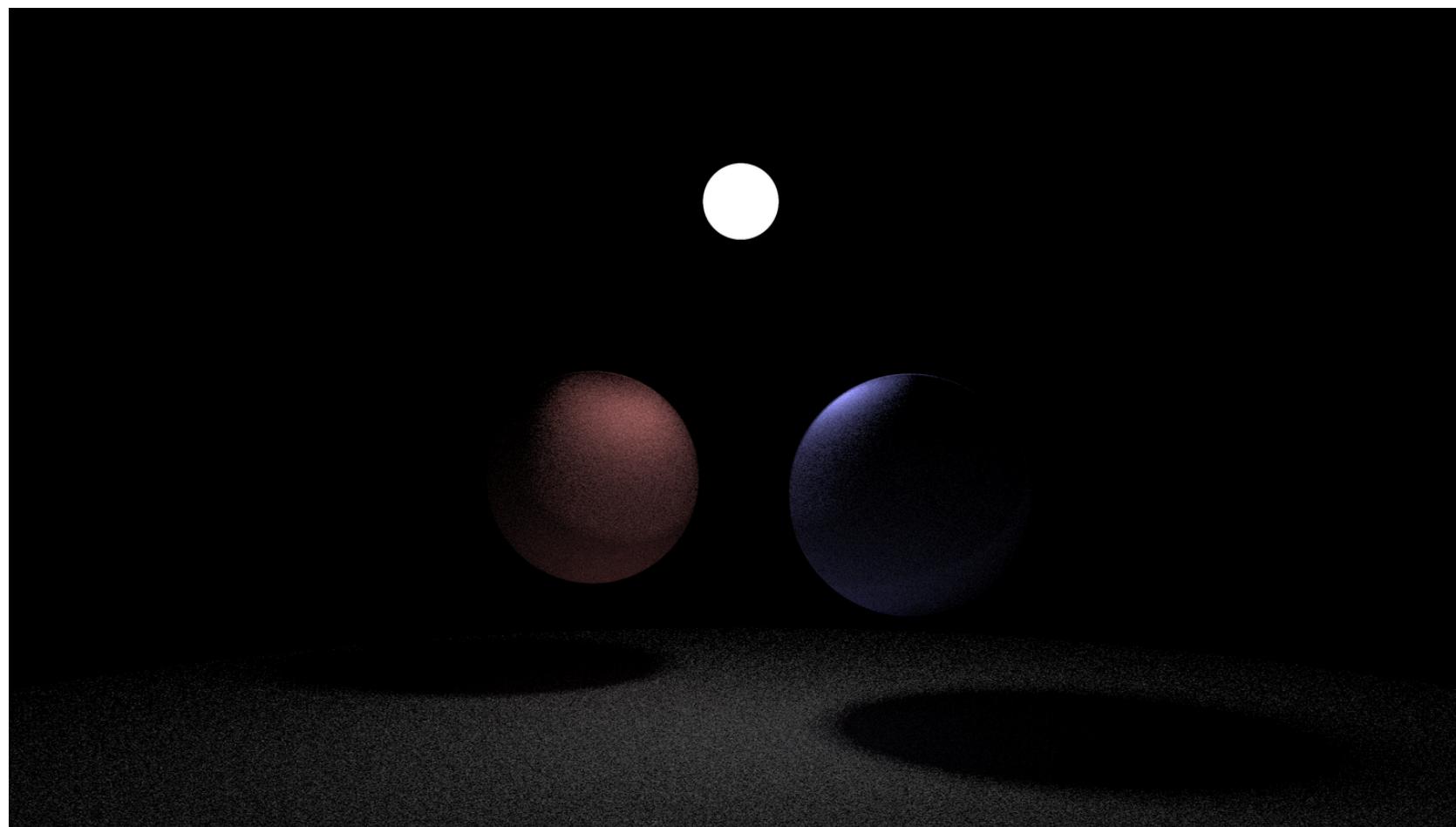
Leonardo Heisler

Thiago Gonçalves

Vinicius Spadotto

Algoritmo e Paralelismo

- Algoritmo escolhido: Pathtracing para renderização de imagens 3D, feito com o algoritmo Monte Carlo
- Paralelismo explorado: A nível de Samples, a nível de Tasks, e com tarefas OpenMP
- Número de Threads pra comparação: {1, 20, 40}



**exemplo de imagem criada com o algoritmo

Experimentos Realizados e Fatores

Resolution	Otimization	Samples	Threads	Blocks
3840x2160	collapse	30	20	0.1
3840x2160	samples	30	1	0.1
640x480	collapse	30	1	0.1
640x480	tasks	900	20	0.1
640x480	tasks	900	1	0.1
3840x2160	tasks	900	40	0.1
3840x2160	tasks	30	20	0.1
640x480	samples	30	40	0.1
640x480	collapse	900	20	0.1
3840x2160	samples	900	40	0.1
640x480	samples	900	20	0.1
3840x2160	collapse	30	40	0.1
3840x2160	samples	900	1	0.1
3840x2160	samples	900	20	0.1
640x480	collapse	900	40	0.1

Samples: Quantidade de amostras utilizadas pelo algoritmo

Threads: Quantidade de Threads utilizadas

Resolution: Resolução da Imagem final

Optimization: Otimização do utilizada

Resultados dos Experimentos: Uso dos recursos do sistema

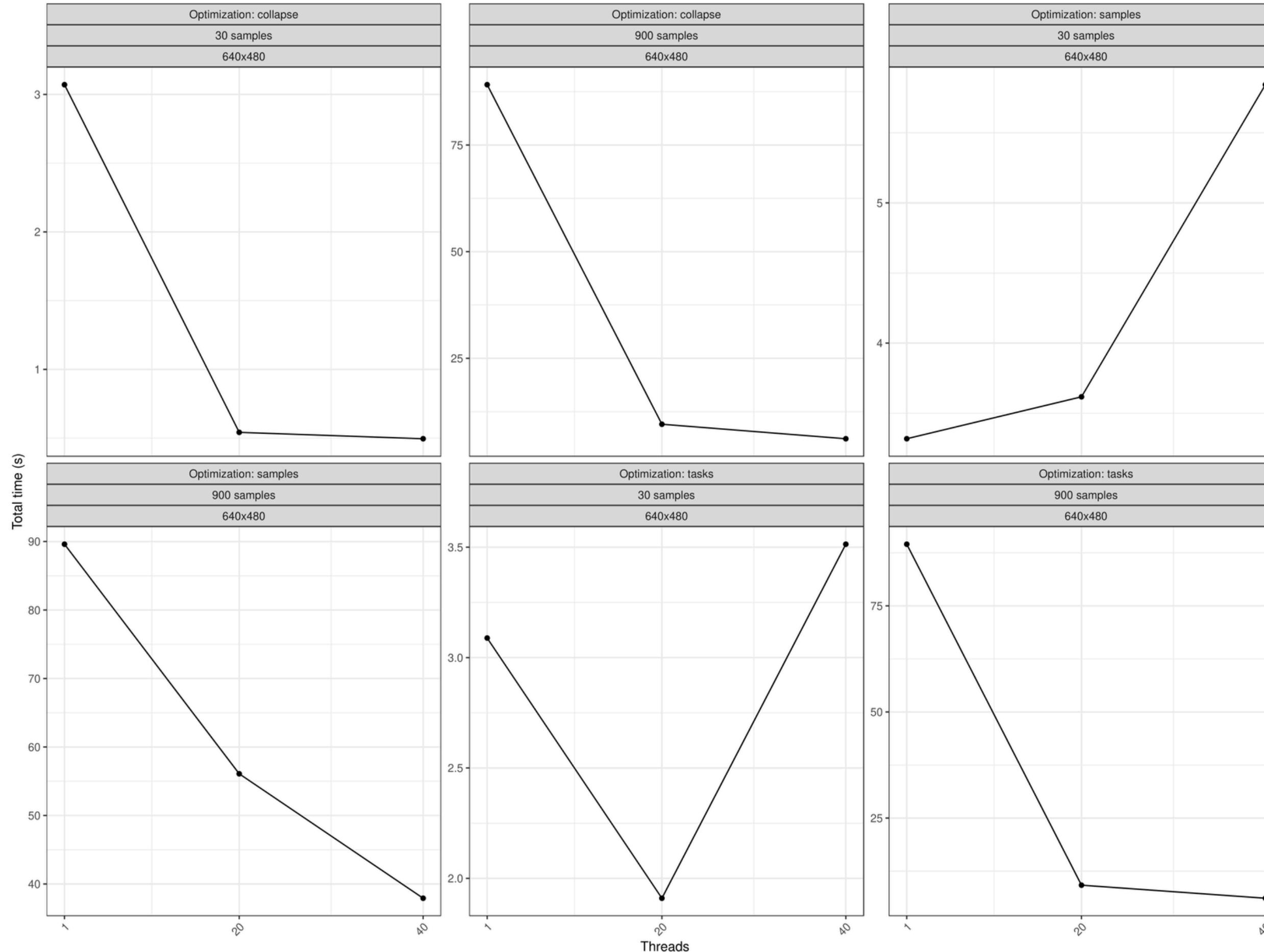
Resolution	Metric	Mean	Standard deviation	Min	Max	Median
3840x2160	CPI	1.825	1.864	0.475	6.166	0.663
3840x2160	CPU	2.733	0.182	2.594	2.989	2.607
3840x2160	Calc. time	597.734	844.91	5.157	2290.334	147.786
3840x2160	LC	0.415	0.359	0.025	0.99	0.436
3840x2160	PC	0.573	0.412	0.049	1	0.649
3840x2160	Total time	600.517	845.01	7.848	2293.335	150.722
640x480	CPI	1.779	1.782	0.478	6.017	0.669
640x480	CPU	2.732	0.181	2.594	2.989	2.609
640x480	Calc. time	23.136	32.965	0.274	89.609	5.816
640x480	LC	0.397	0.352	0.024	0.988	0.359
640x480	PC	0.542	0.399	0.049	0.989	0.607
640x480	Total time	23.261	32.965	0.384	89.745	5.973

**testes realizados na máquina hype5

PC e LC: min correspondente a single core e max correspondente ao paralelismo total

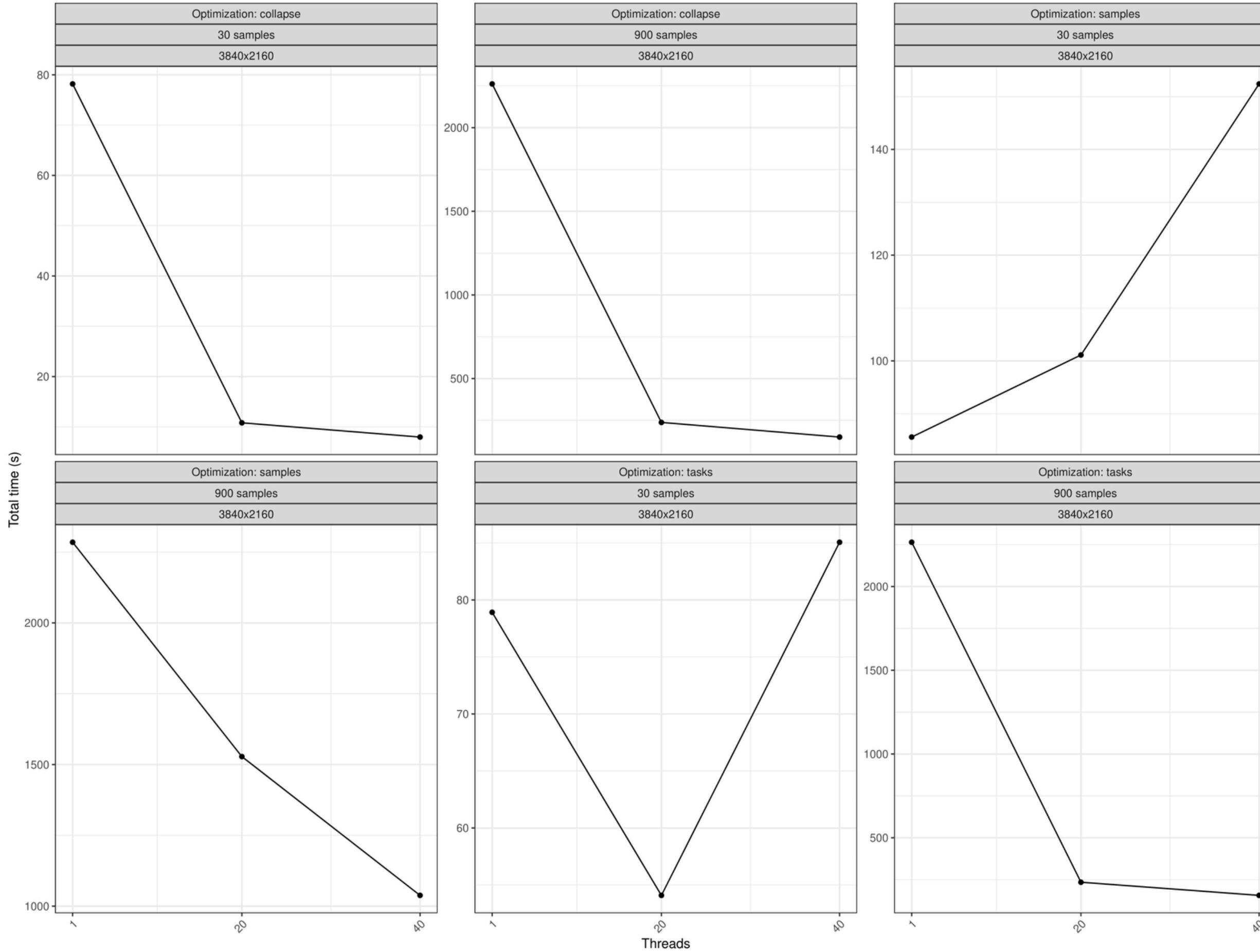
Tempo de execução: Variabilidade alta pelo uso do paralelismo

Resultados dos Experimentos: Resolução 640x480



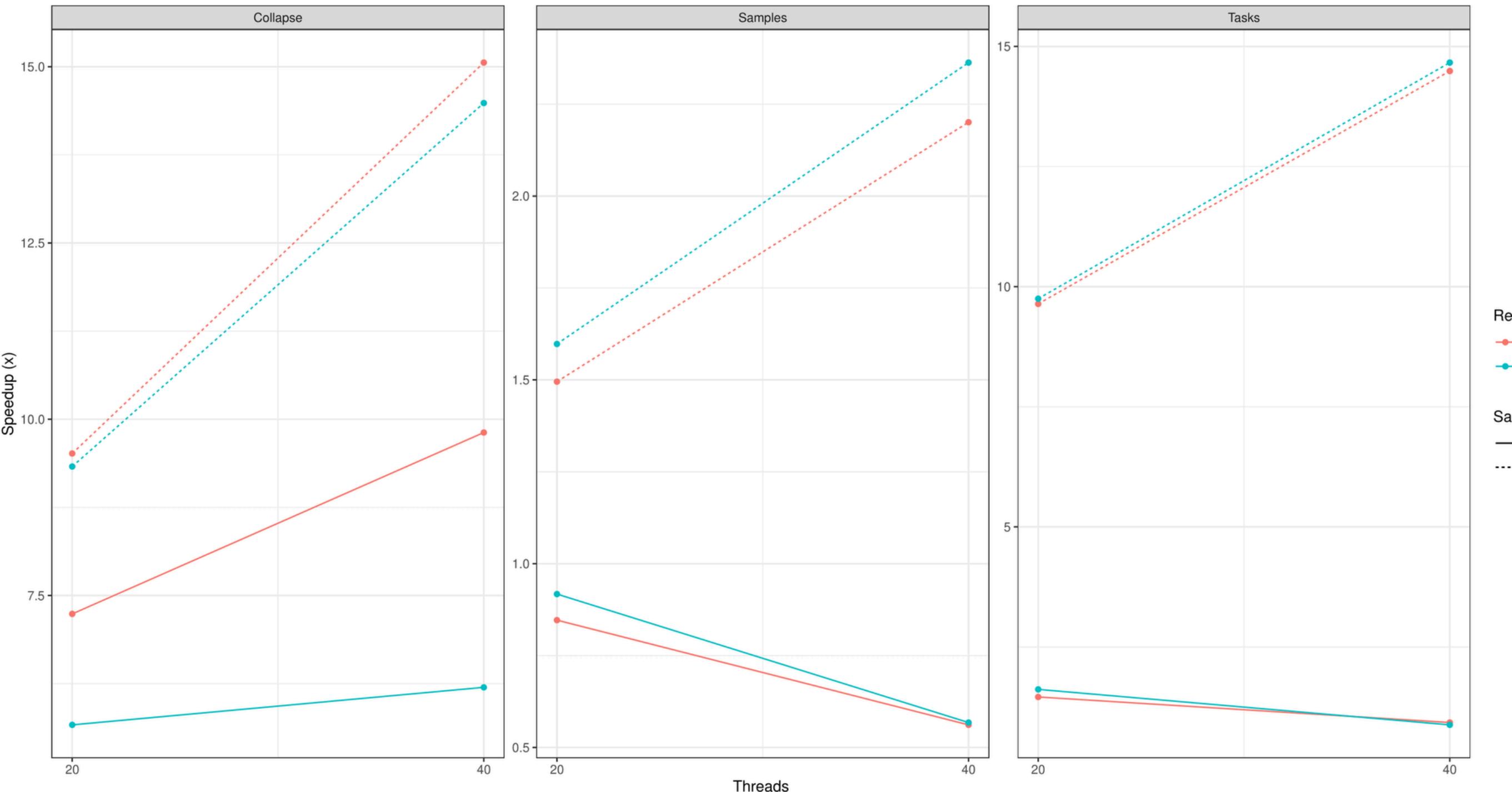
Collapse e Task demonstraram uma diminuição no tempo de execução enquanto **Samples** executa em mais tempo por conta do overhead do OpenMP.

Resultados dos Experimentos: Resolução 3840 × 2160



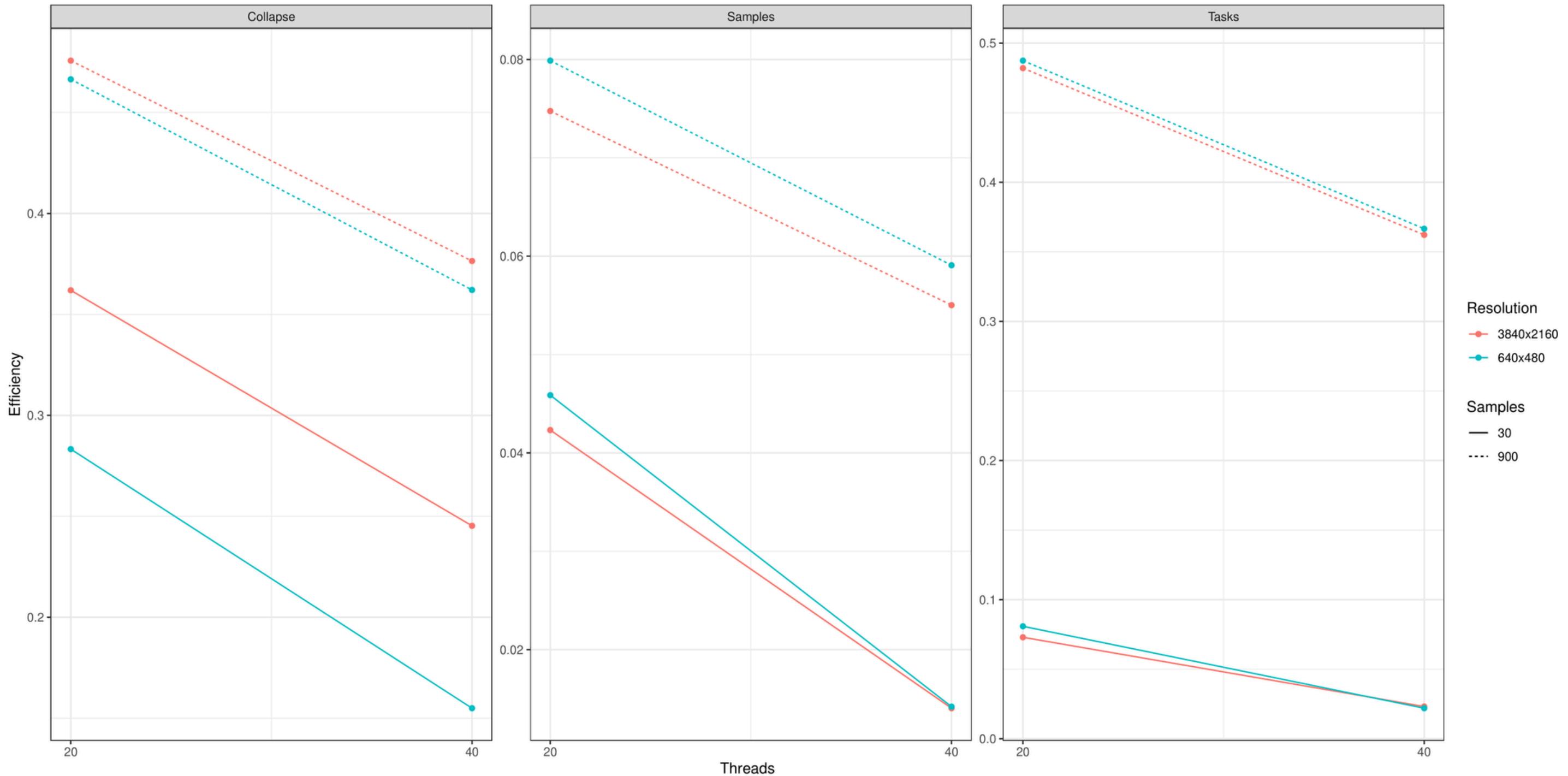
Resultados semelhantes à resolução menor; ganho no paralelismo mais evidente, levando **2000 segundos a menos para completar a tarefa quando comparado à execução sequencial.**

Resultados dos Experimentos: SpeedUp



SpeedUp de até **15x** em **Tasks** e **Collapse**, sendo maior nas instâncias de maior resolução - carga de trabalho mais expressiva. **Samples** pouco melhorou o custo de tempo.

Resultados dos Experimentos: Eficiência



Eficiência de **até 50%** nos casos onde a carga de trabalho é maior para **Collapse** e **Tasks**; Ao usarmos 40 Threads a Eficiência cai para **30%** pelo custo das trocas de contexto

Conclusões

- **Melhores Estratégias:** Paralelismo a nível de imagem (collapse) e a nível de tarefas (tasks) apresentaram desempenho similar e foram as mais eficientes.
- **Pior Estratégia:** Paralelismo a nível de amostra (samples) foi ineficiente, provavelmente devido ao alto overhead causado pela granularidade fina.
- **Escalabilidade:** O desempenho escala bem com o aumento da carga de trabalho (resolução e número de amostras), tornando a paralelização mais vantajosa.
- **Limitações:** A eficiência não é perfeita e diminui com mais threads, indicando gargalos sequenciais ou sobrecarga de sincronização.