

[CSE 5441](#)  
[Lab 3 Report](#)  
[Rajarshi Biswas](#)  
[biswas.91@osu.edu](mailto:biswas.91@osu.edu)

## Table of Contents

Run time of OpenMP programs for different threads: .....	1
Comparative run-time analysis between pthread and OpenMP:.....	2
Best Result: .....	3
Implementation effort:.....	3
Threading mechanism for similar application: .....	3
When to use pthreads over OpenMP? .....	3
Surprises encountered in this exercise:.....	3

## Run time of OpenMP programs for different threads:

Time measured using UNIX time command on stdlinux using testgrid\_400\_12206 test file.

For both persistent and disposable thread (checked in each iteration) model I have observed the actual number of threads created and the requested number of threads are same.

Epsilon – 0.03, Affect rate – 0.03.

Dissipation converged in 460838 iterations; max DSV = 0.0850069 and min DSV = 0.0824567.

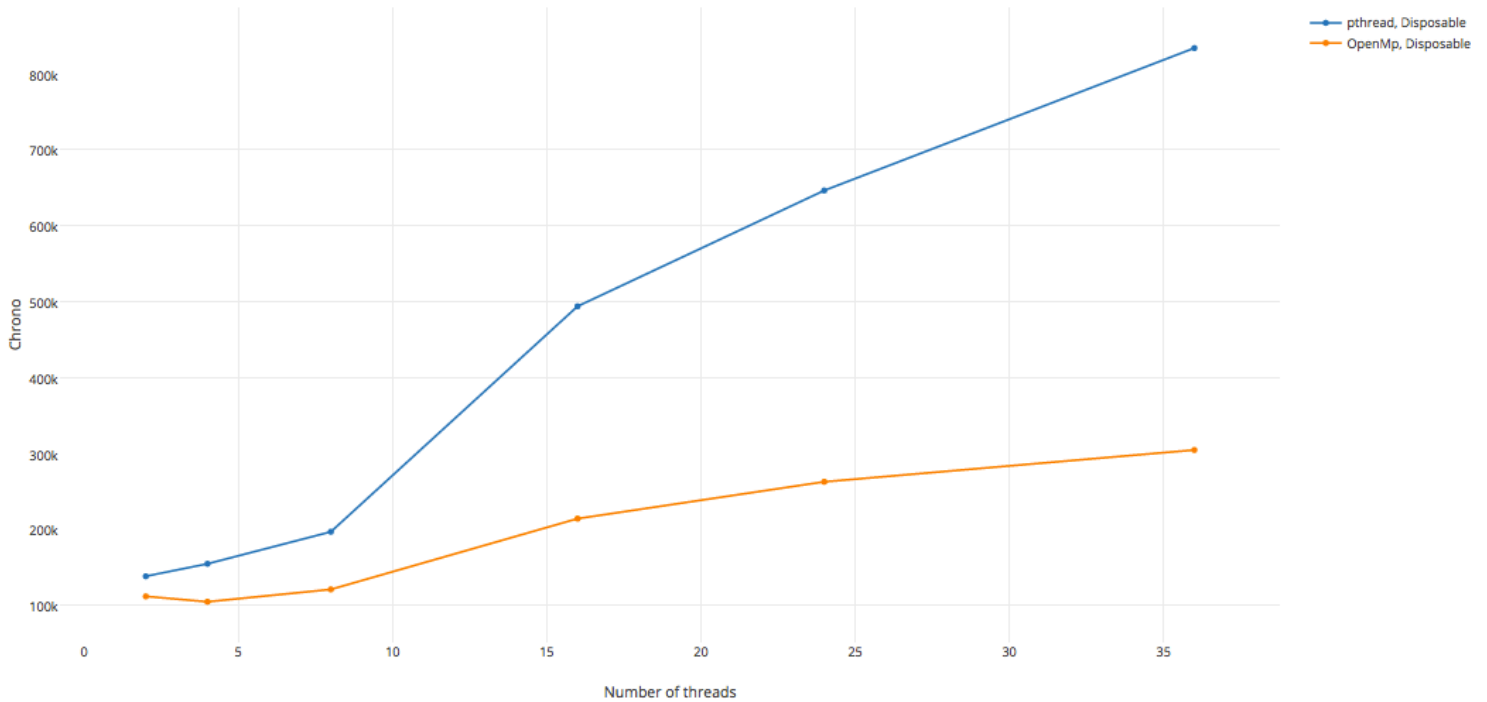
Number of threads	Disposable Thread Model			Persistent Thread Mode		
	Real	User	Sys	Real	User	Sys
2	1m36.360s	3m12.067s	0m0.394s	1m35.819s	3m11.361s	0m0.193s
4	1m4.712s	4m15.826s	0m0.587s	1m3.595s	4m12.060s	0m0.559s
8	1m37.427s	3m38.974s	0m40.687s	1m34.720s	3m38.818s	0m37.501s
16	2m44.623s	3m56.775s	0m53.311s	2m29.081s	3m59.522s	0m58.448s
24	3m35.016s	4m11.456s	1m6.973s	3m18.619s	4m12.245s	1m14.686s
36	4m21.096s	4m25.029s	1m30.429s	4m12.935s	4m31.499s	1m44.747s

Clearly, persistent thread model has performed better in compared to disposable thread model.

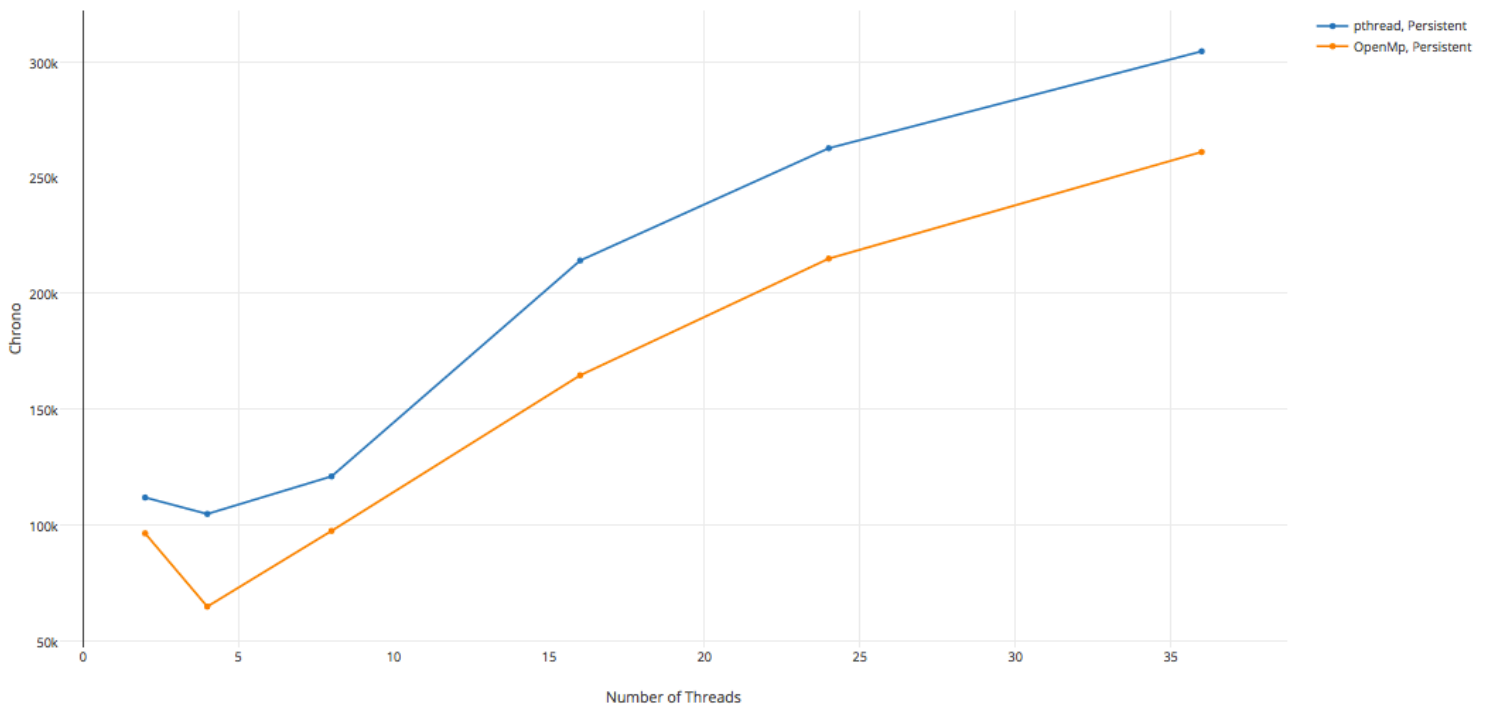
## Comparative run-time analysis between pthread and OpenMP:

Time method used Chrono on stdlinux using testgrid\_400\_12206 test file.; Epsilon – 0.03, Affect rate – 0.03; Dissipation converged in 460838 iterations; max DSV = 0.0850069 and min DSV = 0.0824567.

Comparison of pthread and OpenMP programs (Disposable Thread Model)



Comparison of pthread and OpenMP programs (Persistent Thread Model)



Clearly, OpenMP threading (both disposable and persistent model) mechanism provided better performance.

### Best Result:

OpenMP threading mechanism provided the best results compared to pthread for both disposable and persistent threading model.

### Implementation effort:

OpenMP threading mechanism was the easiest to implement.

### Threading mechanism for similar application:

I would most likely choose the OpenMP threading mechanism for a similar application.

### When to use pthreads over OpenMP?

pthreads offers a much greater range of primitive functions that provide finer-grained control over threading operations. So, in applications in which threads have to be individually managed, using pthreads would be preferable.

### Surprises encountered in this exercise:

I have seen a significant improvement in wall clock time for OpenMP disposable thread version compared to pthread disposable thread version.