

Assignment 3b Report

Introduction

Little's Law is a theory that relates the average number of items in a system (L), the average arrival rate of items (λ), and the average time an item spends in the system (W). The law is expressed as:

$$L = \lambda \times W$$

In the context of web servers and concurrent requests, Little's Law can help us understand the relationship between the three.

Observations from the Results

The provided results show the total time taken (in milliseconds) for different numbers of threads (100 and 1000) with varying `maxThreads` settings on Tomcat (200, 100, and 10).

In terms of the Little's Law:

1. $L = \text{maxThreads}$ setting on Tomcat
2. $\lambda = \text{Number of Threads} / \text{Total Time taken}$ (assuming arrival rate is constant)
3. $W = \text{Total Time taken}$

Below are the key observations:

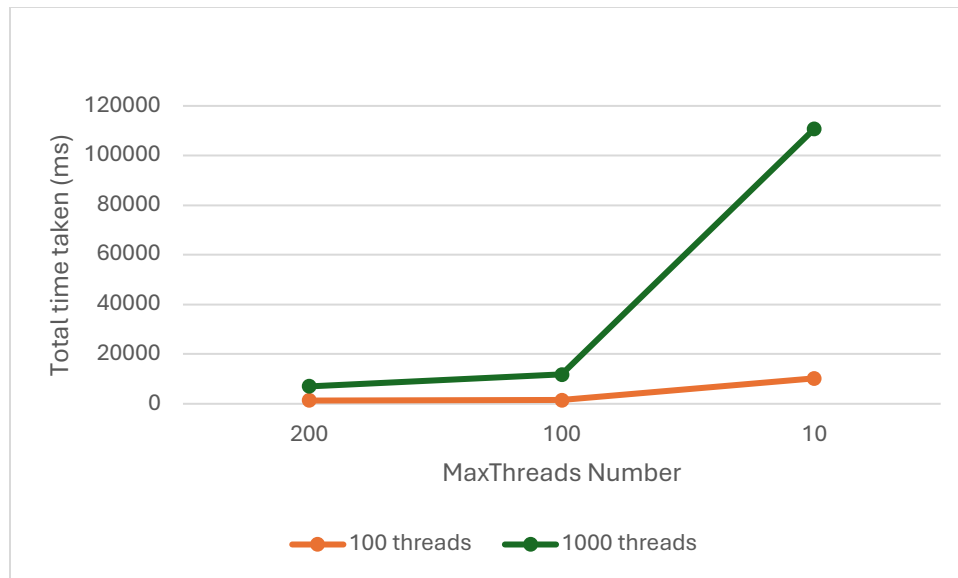
1. 100 threads

<code>maxThreads</code> number	Total time taken (ms)
200	1307
100	1424
10	10256

2. 1000 threads

<code>maxThreads</code> number	Total time taken (ms)
200	5701
100	10365
10	100510

3. A plotted graph on the combined results:



Analysis

1. 100 Threads:

- `maxThreads = 200`: The system can handle all 100 threads concurrently, so the total time is relatively low (1307 ms).
- `maxThreads = 100`: The system can handle up to 100 threads concurrently, which matches the number of threads, resulting in a slightly higher total time (1424 ms).
- `maxThreads = 10`: The system can only handle 10 threads at a time, leading to a significant increase in total time (10256 ms) due to queuing.

2. 1000 Threads:

- `maxThreads = 200`: The system can handle 200 threads concurrently, but with 1000 threads, there is significant queuing, resulting in a higher total time (5701 ms).
- `maxThreads = 100`: The system can handle up to 100 threads concurrently, leading to even more queuing and a higher total time (10365 ms).
- `maxThreads = 10`: The system can only handle 10 threads at a time, causing a massive increase in total time (100510 ms) due to extensive queuing.

Conclusion

The results clearly demonstrate the impact of the `maxThreads` setting on the total time taken to process requests. As the number of concurrent threads increases beyond the `MaxThreads` limit, the system experiences queuing, leading to longer total processing times. This aligns with Little's Law, which predicts that increasing the number of concurrent requests (L) without a corresponding increase in the processing capacity (λ) will result in longer average times in the system (W).