

**Universidade Católica de Brasília**  
**Ciência da Computação**  
**Atividade Avaliativa - Programação Concorrente e**  
**Distribuída**

**Análise de Desempenho no Uso de Threads para**  
**Coleta e Processamento de Dados**

**Artur Vivacqua**  
**Professor: João Robson Santos Martins**

**Brasília**  
**2024**

**ARTUR VIVACQUA**

**ANÁLISE DE DESEMPENHO NO USO DE THREADS PARA COLETA E  
PROCESSAMENTO DE DADOS**

Artigo apresentado ao curso de graduação em ciência da computação da Universidade Católica de Brasília, como requisito parcial para a obtenção do título de bacharel em computação.

Orientador: João Robson Santos Martins

Brasília  
2024

## **RESUMO**

Referência: Artur Vivacqua Brandão, Análise de Desempenho no Uso de Threads para Coleta e Processamento de Dados , 2024. nr p. Bacharelado em Ciência da Computação – UCB – Universidade Católica de Brasília, Taguatinga – DF, 2023.

Este artigo explora o conceito de threads, como elas funcionam, seu impacto no tempo de execução de algoritmos e a relação entre computação concorrente e paralela. Além disso, apresentamos um experimento que compara quatro versões de um algoritmo de coleta e processamento de dados climáticos utilizando diferentes números de threads.

## *SUMÁRIO*

<b>1. Introdução</b>	<b>5</b>
1.1 O que são Threads	5
1.2 Como Threads Funcionam Computacionalmente	5
1.3 Relação Entre Computação Concorrente e Paralela e a Performance dos Algoritmos	5
<b>2. Análise de cenário</b>	<b>6</b>
2.1 Descrição do experimento	6
2.2 Resultados Obtidos	6
<b>3. Conclusão</b>	<b>7</b>

# 1 INTRODUÇÃO

A computação moderna busca constantemente otimizar o desempenho de algoritmos, especialmente aqueles que lidam com grandes volumes de dados e operações intensivas. Uma abordagem comum para melhorar o desempenho é o uso de threads, que permite a execução paralela de tarefas. Uma thread, conforme descrito por Silberschatz et al. (2008), é uma unidade fundamental de uso da CPU e pode ser vista como um componente da execução de um programa. Embora tenha um ciclo de vida igual ao de um processo, um thread compartilha o mesmo espaço de endereçamento do processo "pai". Essa característica permite um aumento no desempenho.

## 1.1 O que são Threads?

Threads são as menores unidades de processamento que podem ser executadas de forma independente por um sistema operacional. Elas são frequentemente referidas como "caminhos de execução" dentro de um programa, permitindo que múltiplas operações sejam realizadas simultaneamente. Cada thread compartilha o mesmo espaço de memória do processo principal, mas possui seu próprio conjunto de registradores, pilha e contador de programa.

## 1.2 Como Threads Funcionam Computacionalmente

Threads são gerenciadas pelo sistema operacional e pelo hardware subjacente. Em um ambiente com múltiplos núcleos, threads podem ser distribuídas na CPU, permitindo a execução paralela real. Em sistemas com um único núcleo, o sistema operacional alterna entre as threads ( "time-slicing"), proporcionando paralelismo. O uso de threads pode reduzir significativamente o tempo de execução de algoritmos, especialmente aqueles que podem ser divididos em sub-tarefas independentes. O desempenho ótimo é alcançado balanceando o número de threads e a carga de trabalho de cada uma.

## 1.3 Relação Entre Computação Concorrente e Paralela e a Performance dos Algoritmos

A computação concorrente envolve a execução de múltiplas tarefas de forma sobreposta em um único núcleo de CPU, enquanto computação paralela envolve a execução simultânea de múltiplas tarefas em múltiplos núcleos. A programação concorrente é útil para melhorar a eficiência de programas que esperam por operações de entrada e saída, enquanto o paralelismo é essencial para tarefas que exigem alta capacidade de processamento. A escolha entre concorrência e paralelismo afeta diretamente a performance dos algoritmos.

## 2 EXPERIMENTO

### 2.1 Descrição do Experimento

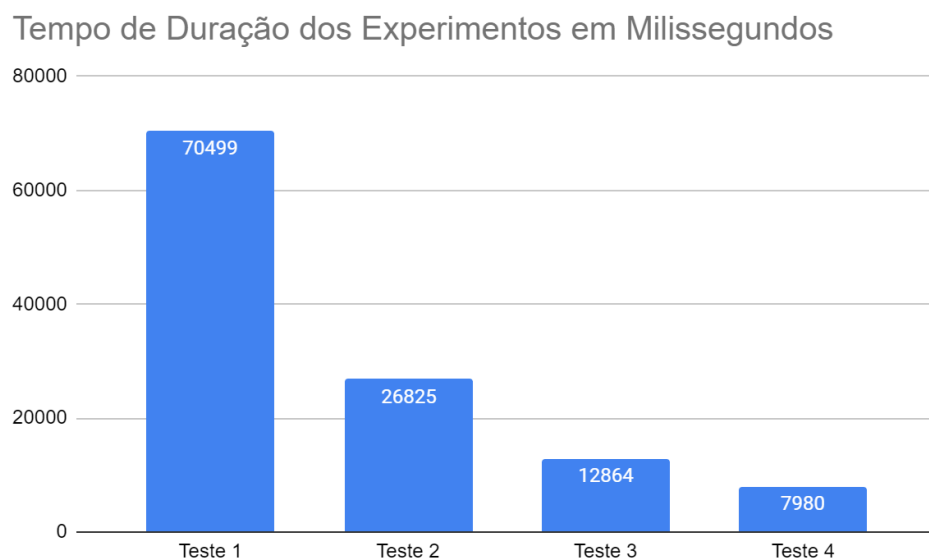
Foram realizadas quatro versões de um algoritmo para coleta e processamento de dados climáticos das 27 capitais brasileiras. O objetivo era calcular as temperaturas média, mínima e máxima por dia para cada capital. As versões variaram o número de threads usadas da seguinte maneira: 1, 3, 9 e 27 threads.

### 2.2 Resultados Obtidos

Os tempos médios de execução para cada versão do experimento foram os seguintes:

- **Experimento 1 (1 Thread):** 70499 milissegundos
- **Experimento 2 (3 Threads):** 26825 milissegundos
- **Experimento 3 (9 Threads):** 12864 milissegundos
- **Experimento 4 (27 Threads):** 7980 milissegundos

O gráfico Abaixo ilustra a relação entre o número de threads e o tempo de execução.



### 2.3 Análise dos Resultados

Os resultados mostram uma redução significativa no tempo de execução à medida que o número de threads aumenta. No experimento com uma única thread, o tempo de execução foi de aproximadamente 70 segundos. Com o aumento para 3 threads, o tempo foi reduzido para aproximadamente 27 segundos. Com 9 threads, a redução foi ainda mais acentuada, para aproximadamente 13 segundos. Finalmente, com 27 threads, o tempo caiu para cerca de 8 segundos. Observa-se uma diminuição quase exponencial no tempo de execução inicial, que

se estabiliza conforme o número de threads continua a aumentar. Este comportamento se deve ao balanceamento de carga e ao overhead de criação e gerenciamento de threads. O uso de múltiplas threads permitiu que o algoritmo processasse múltiplas requisições HTTP simultaneamente, reduzindo o tempo de espera entre as operações de entrada e saída. No entanto, a partir de um certo ponto, o ganho de desempenho diminui devido ao consumo adicional de recursos associado ao gerenciamento de um grande número de threads.

### **3 CONCLUSÃO**

Este estudo demonstrou a eficácia do uso de threads para otimizar o tempo de execução de algoritmos de coleta e processamento de dados. A escolha do número adequado de threads depende da natureza do problema e da capacidade do hardware subjacente. Para operações intensivas, como requisições HTTP, o uso de múltiplas threads pode proporcionar melhorias significativas no desempenho. Por isso que grandes empresas utilizam o sistema de threads em seu desenvolvimento, inovando a tecnologia com eficácia e agilidade.

### **REFERÊNCIAS:**

[http://www.inicepg.univap.br/cd/INIC\\_2016/anais/arquivos/0902\\_0932\\_02.pdf](http://www.inicepg.univap.br/cd/INIC_2016/anais/arquivos/0902_0932_02.pdf)