

Computacion paralela Trabajo 01

henry angel pacco zuvieta

09 Mayo 2024

1 Realizar la impresion de los numeros pares entre 50000 y 100000

1.1 Algoritmo paralelo

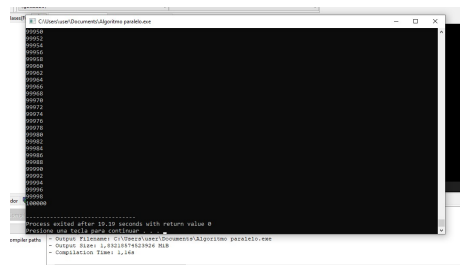


Figure 1: Este algoritmo paralelo imprime los números pares entre 50000 y 100000 utilizando OpenMP para la programación en paralelo. Sin embargo, al ejecutar el código, se observa que el tiempo de ejecución es de 19.19 segundos

Listing 1: Código en C++

```
#include <iostream>
#include <omp.h>

int main() {
    int inicio = 50000;
    int fin = 100000;

    #pragma omp parallel for
    for (int i = inicio; i <= fin; i++) {
        if (i % 2 == 0) {
            #pragma omp critical
            std::cout << i << std::endl;
        }
    }
}
```

```

    }

    return 0;
}

```

1.2 Algoritmo concurrente

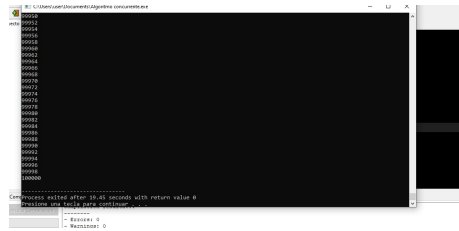


Figure 2: Este algoritmo concurrente imprime los números pares entre 50000 y 100000 de manera secuencial. Sin embargo, al ejecutar el código, se observa que el tiempo de ejecución es de 19.45 segundos

Listing 2: Código en C++

```

#include <iostream>

int main() {
    // Imprimir nmeros pares entre 50000 y 100000
    for (int i = 50000; i <= 100000; i++) {
        if (i % 2 == 0) {
            std::cout << i << std::endl;
        }
    }
    return 0;
}

```

DESCRIPCIÓN

presentamos dos programas para imprimir números pares entre 50000 y 100000. El primer programa utiliza OpenMP para la programación paralela, mientras que el segundo programa es una implementación secuencial. Aunque ambos programas funcionan correctamente, el programa paralelo tiene un tiempo de ejecución de 19.19 segundos, mientras que el programa secuencial tiene un tiempo de ejecución de 19.45 segundos. Esto sugiere que, en este caso, la programación paralela ofrece un mejor rendimiento, aunque marginalmente.