



Tecnológico de Monterrey

Act 1.5

Periodo Semestral Agosto-Diciembre 2022

Fecha de entrega | *Septiembre 5, 2022*

Multiprocesadores TE3061.1

Profesor:

Emmanuel Torres Rios

Abril Jimenez Alatraste | *A01732412*

Instituto Tecnológico de Estudios Superiores de Monterrey

Instrucción

Desarrollar los programas necesarios para medir:

- 1) Velocidad de lecto-escritura de la memoria RAM de su equipo mediante threads en comparación con la cache
- 2) Velocidad de lecto-escritura del disco duro de su equipo mediante threads en comparación con la cache
- 3) La operación matemática(exponencial, potencia, trigonométrica, etc.) más lenta en comparación con la suma.

Debe entregar como evidencia de la actividad un reporte discutiendo la forma y los resultados obtenidos

Introducción

La memoria de acceso aleatorio (RAM) es la memoria a corto plazo de una computadora, donde los datos que el procesador está usando actualmente se almacenan temporalmente, ahora bien la memoria caché es un componente de computadora basado en un chip que hace que la recuperación de datos de la memoria de la computadora sea más eficiente. Actúa como un área de almacenamiento temporal de la que el procesador de la computadora puede recuperar datos fácilmente.

Al tener esto en mente debemos entender que estos problemas son para medir la capacidad de la laptop HP OMEN 015.

Marco Teórico

Debemos realizar la paralelización del programa para así obtener la información con respecto de la RAM ya que esa memoria usaremos para los archivos txt.c

Con un pointer de tipo doble la obtención de datos nos devolverá la matriz generada en la misma función.

Finalmente los datos se almacenarán en una tabla.

```

double** euler_method(int N){
    printf("Numero de pasos:%d Atendido por thread:%d\n", N,omp_get_thread_num());
    double w0=0.5,a=0,b=2;
    int i;

    double** data = malloc(sizeof(double*) * 2);
    double* w = malloc(sizeof(double) * N);
    double* t = malloc(sizeof(double) * N);
    double h,ab;

    h=(b-a)/N;
    w[0] = w0;
    t[0] = a;

    data[0] = calloc(1,sizeof(double));
    data[1] = calloc(1,sizeof(double));

    for(i=0;i<N;i++){
        t[i]=a+(h*i);
        // printf("%lf\t", t[i]);
        w[i]=w[i-1]+h*(1+t[i-1]*t[i-1]-w[i-1]);
        // printf("%lf\n", w[i]);
    }
    // printf("\n");
    *(data) = t; *(data+1) = w;
    // printArr(data);
    return data;
}

```

Para paralelizar este programa llevaremos a cabo una suma de 1000000 de puntos y haciendo uso de 6 threads, finalmente se deberá definir el numero de threads, el tiempo de inicio, la parte de paralelización, el tiempo final y las escrituras en texto de de las tablas generadas

```

int main(){
    omp_set_num_threads(NUM_THREADS);
    const double startTime = omp_get_wtime();
    #pragma omp parallel
    {
        #pragma omp sections
        {
            #pragma omp section
            tab1 = euler_method(60);
            #pragma omp section
            tab2 = euler_method(600);
            #pragma omp section
            tab3 = euler_method(50);
            #pragma omp section
            tab4 = euler_method(500);
            #pragma omp section
            tab5 = euler_method(40);
            #pragma omp section
            tab6 = euler_method(400);
        }
    }
}

```

```

void writeArr(double** arr, FILE *fp){
    size_t size = _msize(*arr)/sizeof arr[0][0];
    // printf("%d\n", size);
    fprintf(fp, "Datos que encuentra el metodo de Euler:\nN\t\tX\t\t\t\t Y\n");
    for(int j = 0; j < size; j++){
        fprintf(fp, "%03i\t%.4lf\t%.4lf\n", j, arr[0][j], arr[1][j]);
    }
    printf("\n");
}

```

COMPARACION Caché, RAM y ROM

Compararemos y obtendremos los speedups obteniendo que lo mas rapido en mi caso es mi memoria RAM.

pues cache se obtuvo 0.011000

RAM 0.004000

ROM 0.400000

Comparación con diferentes operaciones matemáticas

la función a evaluar para la RAM es $f(y) = x^2 - y + 1$ por lo que $f(y)$ hay que variar a las siguientes operaciones

$$f(y) = (x^2 - y + 1)^2$$

0.02400

$$f(x) = \sqrt{x^2 - y + 1}$$

0.01100

$$f(x) = \text{sen}(x^2 - y + 1)$$

0.012000

$$f(x) = \text{atan}(x^2 - y + 1)$$

0.017000

$$f(x) = \log(x^2 - y + 1)$$

0.016000

$$f(x) = e^{x^2 - y + 1}$$

0.035000

Conclusión

Se concluye que apesar de sus limitaciones una RAM de 8 es suficientemente rapida.