Computación Paralela y Distribuída

Repaso Examen Parcial

Pregrado

2022-II

Profesor: José Fiestas

1.01

Indicaciones específicas:

- Esta evaluación contiene 12 páginas (incluyendo esta página) con 5 preguntas. El total de puntos son 20.
- El tiempo límite para la evaluación es 120 minutos.
- El exámen deberá ser respondida en un solo archivo pdf. Si es foto pueden ser varios archivos
- Deberá subir estos archivos directamente a https://www.gradescope.com
- Se pide activar cámaras durante el exámen. En caso no ser posible, enviar un correo de justificación

Competencias:

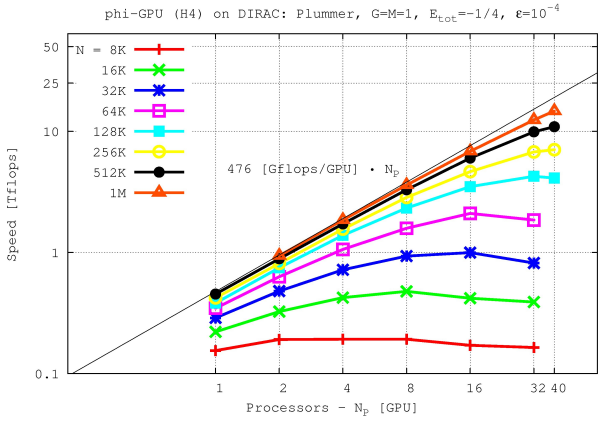
- Aplica conocimientos de computación apropiados para la solución de problemas definidos y sus requerimientos en la disciplina del programa. (nivel 3)
- Resuelve problemas de computación y otras disciplinas relevantes en el dominio (nivel 3)
- Analiza y valora el impacto local y global de la computación sobre las personas, las organizaciones y la sociedad (nivel 3)
- Reconoce la necesidad del aprendizaje autónomo (nivel 2)

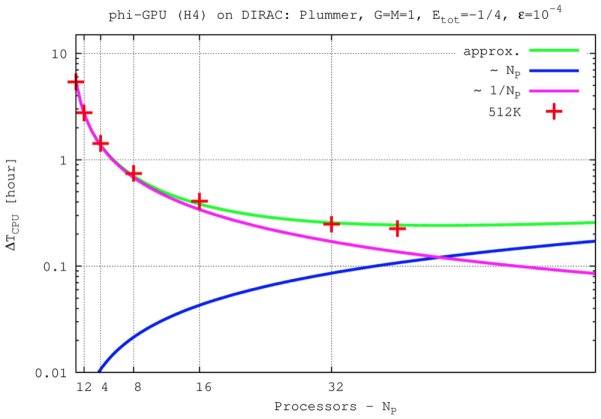
Calificación:

Tabla de puntos (sólo para uso del professor)

Question	Points	Score
1	4	
2	4	
3	4	
4	4	
5	4	
Total:	20	

1. (4 points) Las siguientes gráficas representan velocidad en FLOPs vs. número de nodos de un código de N-cuerpos para N=512K así como el tiempo de ejecución en horas vs. número de nodos.





- Utilice la complejidad del algoritmo de N-cuerpos para generar una fórmula que contenga las componentes principales del tiempo de ejecución.
  Respuesta: $T = T_{comp} + T_{comm} = O(n^2/p) + O(\frac{n}{p}p) = O(n^2/p)$
- Derive una formula para calcular la velocidad en TFLOPs. Calcule el speedup y compárelo con la formula de velocidad. Calcule la eficiencia teórica para este problema
  Respuesta: multiplicando los flops por iteración, por n^2 y dividiendo entre el tiempo $S_{TFLOPs} = \frac{FLOPs \times n^2}{T \times 10^{-9}} \propto p$ debido a que $T \propto 1/p$. Por otro lado, $S = n^2/(n^2/p) = p$ (mantiene la misma proporcionalidad con p que la velocidad), $E = 1$
- Nombre dos posibles factores causantes de overhead para este problema
  Respuesta: Comunicacion punto a punto de partículas en procesos vecinos, gather de resultados para actualizar posicion de partículas en cada paso temporal.

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Método o algoritmo	Describe al algoritmo de solución del problema planteado en forma adecuada (2 pts)	Algoritmo con algunos errores que no afectan el resultado (1.5 pts).	Algoritmo con errores que afectan minimamente el resultado (0.5 pt).	EAlgoritmo con errores, que afectan significativamente el resultado (0 pts)
Resultados	Solución correcta usando un método adecuado (1 pt)	Errores mínimos en el método que no afectan el resultado (0.6 pts)	Errores en el método que afectan el resultado (0.3 pts)	No aplica el método ni llega a la solución correcta (0 pts).
Optimización	Solución original y optimizada (1 pt)	Solución parcialmente optimizada (0.6 pts)	Solución original pero no optimizada (0.3 pts)	Resultado encontrado no está optimizado (0 pts).

2. (4 points)
Responda a las siguientes preguntas con respecto al modelo PRAM del método de Suma de Prefijos mostrado abajo.
a) Determine ($T(n)$) y trabajo ($W(n)$)
b) ¿Qué paradigma se aplica a este algoritmo (PRAM)?
c) Un algoritmo se considera WT-óptimo cuando $W(n) = T^*(n)$, siendo $T^*(n)$ el algoritmo secuencial óptimo. ¿Es éste algoritmo WT-óptimo ?

Entrada: $x_1, ..., x_n$, $(n = 2^k)$
Salida: $s_1, ..., s_n$, $s_i = x_1 \circ x_1 \circ ... \circ x_i \forall i \in \mathbb{N}$, donde $\circ$ es una operación asociativa

---

```
if n=1 then
    s_1 := x_1
else
    for i ∈ {1,...,n/2} pardo
        y_i := x_{2i-1} ∘ x_{2i}
    (z_1,...,z_{n/2}) := Prefix-sum(y_1,...,y_{n/2})
    for i ∈ {1,...,n} pardo
        if i = 1 then
            s_1 := x_1
        else
            if i mod 2 = 1 then
                s_i := z_{(i-1)/2} ∘ x_i
            else
                s_i := z_{i/2}
            endif
        endif
endif
```

Respuesta (a): T(n):
Primer condicional (if): O(1)
Primer pardo: O(1)
Recursividad: O(log n) , ya que T(n/2)+c = T(n/4) + 2c = T(1) + (log n) c
Segundo pardo: O(1)
T(n)=O(log n)
W(n):
Primer condicional (if): O(1)
Primer pardo:O(n/2)
Recursividad: O(n) , ya que W(n) $\leq$ W(n/2) + dn = W(n/4) + d n/2 + dn = d 2n = O(n)
Segundo pardo: O(n)
W(n)=O(n)

Respuesta (b): se aplica memoria compartida. Para memoria distribuida, necesitamos especificar operaciones de comunicación entre procesos
Respuesta (c): Ya que la suma de prefijos secuencial es $T^* = O(n)$, W(n) = $T^*(n)$

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Método o algoritmo	Describe al algoritmo de solución del problema planteado en forma adecuada (2 pts)	Algoritmo con algunos errores que no afectan el resultado (1.5 pts).	Algoritmo con errores que afectan minimamente el resultado (0.5 pt).	EAlgoritmo con errores, que afectan significativamente el resultado (0 pts)
Resultados	Solución correcta usando un método adecuado (1 pt)	Errores mínimos en el método que no afectan el resultado (0.6 pts)	Errores en el método que afectan el resultado (0.3 pts)	No aplica el método ni llega a la solución correcta (0 pts).
Optimización	Solución original y optimizada (1 pt)	Solución parcialmente optimizada (0.6 pts)	Solución original pero no optimizada (0.3 pts)	Resultado encontrado no está optimizado (0 pts).

3. (4 points)
Desarrolle un algoritmo PRAM para el siguiente procedimiento de prueba de conectividad en una red de procesos:
Cada proceso envía un mensaje a otro proceso, escogido en forma aleatoria. Este último vuelve a enviar el mismo mensaje de retorno, al proceso original. En el momento en que el primer proceso recibe la señal de confirmación, envía un mensaje a otro proceso, escogido en forma aleatoria. De esta forma, todos los p procesos deben enviar n mensajes (y la correspondiente confirmación).
Cuando el procedimiento ha terminado, es decir, todos los procesos enviaron y recibieron su mensaje, el proceso 0 determina el tiempo que ha tomado el procedimiento.
Comente que complejidad se espera del procedimiento de prueba de la red.

Respuesta:
Entrada: M[i]=0, con $1 < i < p$, contando con p procesos.
Salida: proceso 0 calcula el tiempo de ejecución

```
1. k=p
    for i=1 to p
        M[i]=0
    time0=time()
2. for i = 1 to p pardo {
    while (k>1) {
        j= random entre 1 y p
        if (i!=j && M[i]==0) {
            send to j
            recv from i
            k=k-1
            } // if
    } // while
    M[i]=1
    } // for
3. if (p==1) time=time() - time0
```

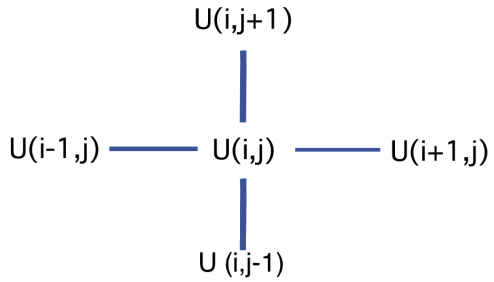
Si cada proceso tiene n mensajes, tenemos una cantidad total de mensajes enviados/recibidos de n(p-1), con p procesos. En paralelo, cada proceso debe enviar un mensaje a la vez al resto, pero entre procesos es paralelo. Obtenemos O(n)

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Método o algoritmo	Describe al algoritmo de solución del problema planteado en forma adecuada (2 pts)	Algoritmo con algunos errores que no afectan el resultado (1.5 pts).	Algoritmo con errores que afectan minimamente el resultado (0.5 pt).	EAlgoritmo con errores, que afectan significativamente el resultado (0 pts)
Resultados	Solución correcta usando un método adecuado (1 pt)	Errores mínimos en el método que no afectan el resultado (0.6 pts)	Errores en el método que afectan el resultado (0.3 pts)	No aplica el método ni llega a la solución correcta (0 pts).
Optimización	Solución original y optimizada (1 pt)	Solución parcialmente optimizada (0.6 pts)	Solución original pero no optimizada (0.3 pts)	Resultado encontrado no está optimizado (0 pts).

4. (4 points)

Un algoritmo de suavizado en procesamiento gráfico utiliza los valores de los pixels vecinos para modificar el valor de cada pixel de la imagen, de acuerdo al siguiente gráfico y ecuación discretizada



$$U_{i,j}^{t+1} = \frac{U_{i-1,j}^{t} + U_{i+1,j}^{t} + U_{i,j-1}^{t} + U_{i,j+1}^{t}}{4}$$

- Formule un algoritmo PRAM (memoria distribuída), con las operaciones de comunicación necesarias, para particionar el dominio y resolver el problema de suavizado, si cada proceso recibe n/p pixels. Asuma que la operación de suavizado tiene T= O(1), y las operaciones de comunicacion tienen una complejidad O(p) . Describa el tratamiento de las condiciones de frontera.
  Entrada: Matriz U$[n][m]$, procesos P$[r][s]$, $n = 2^k, m = 2^l, r = 2^t, s = 2^u$ $l, k, l, t, u \in \mathbb{Z}$
  Salida: Matriz U$[n][m]$ con celdas suavizadas

```
scatter(U,0 to all)
if(P00) send(P00, P01), send(P00, P10)
else if(Pp-1,p-1) recv(Pp-1,p-1, Pp-1,p-2),recv(Pp-1,p-1, Pp-2,p-1)
else{
send(Ps,r, Ps,r+1), send(Ps,r, Ps+1,r),
send(Ps,r, Ps-1,r), send(Ps,r, Ps,r-1)
recv(Ps,r, Ps-1,r),recv(Ps,r, Ps,r-1),
recv(Ps,r, Ps+1,r),recv(Ps,r, Ps,r+1)
}

for i ∈ {1,. . .,n} , j ∈ {1,. . .,m} pardo
     Ui,j = (Ui-1,j + Ui+1,j + Ui,j-1 + Ui,j+1)/4
gather(U,all to 0)
```

- Describa la complejidad de computo y comunicacion de este algoritmo. ¿Es éste algoritmo escalable?
  Respuesta: $T_{comp} = O(4 \cdot nm/p)$, $T_{comm} = O(4 \cdot (p-1))$. $S = O(\frac{nm}{nm/p+p})$, $E = O(\frac{nm/p}{nm/p+p}) = O(\frac{1}{1+p^2/nm})$. Es decir, E=O(1), si $nm \propto p^2$.

- ¿Cuánto tiempo (cómputo) necesitaría el cluster Khipu (Intel-Xeon, con un número efectivo de 80 threads por nodo, 2.1 GHz de frecuencia) para procesar una imagen Ultra HD (3840 x 2160 pixel) en paralelo? ¿Cuánto tiempo si se ejecuta el algoritmo secuencialmente? Considere un total de 4 nodos disponibles en Khipu y que se estiman 2 ciclos por operacion de coma flotante.
  Respuesta: $T_{comp} = \frac{3840 \times 2160 \times 2}{320 \times 2 \times 10^9} \approx 2 \times 10^{-5}$ s

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Método o algoritmo	Describe al algoritmo de solución del problema planteado en forma adecuada (2 pts)	Algoritmo con algunos errores que no afectan el resultado (1.5 pts).	Algoritmo con errores que afectan minimamente el resultado (0.5 pt).	EAlgoritmo con errores, que afectan significativamente el resultado (0 pts)
Resultados	Solución correcta usando un método adecuado (1 pt)	Errores mínimos en el método que no afectan el resultado (0.6 pts)	Errores en el método que afectan el resultado (0.3 pts)	No aplica el método ni llega a la solución correcta (0 pts).
Optimización	Solución original y optimizada (1 pt)	Solución parcialmente optimizada (0.6 pts)	Solución original pero no optimizada (0.3 pts)	Resultado encontrado no está optimizado (0 pts).

5. (4 points)
Corrija el siguiente código, tal que el proceso maestro sea capaz de calcular el máximo global con información de los demás procesos. No se requiere completar todas las funciones sino lo necesario dentro de la sección de código mostrada. Asuma que las variables/funciones están previamente definidas y MPI correctamente inicializado

```
long localmax;
if (rank != 1) {
for (int i = 0; i < nproc-1; i=i+2) {
MPI_IRecv( &localmax, 1, MPI_LONG, i, 123, MPI_COMM_WORLD, &
   request);
}
MPI_Wait(&request, &status);
globalmax = findMax(localmax, globalmax);
un_calculo(globalmax);
} else {
MPI_Send(&localmax, 1, MPI_LONG, 1, 123, MPI_COMM_WORLD, &
   request);
otro_calculo();
MPI_Wait(&request, &status);
}

}
```

Respuesta:

```
long localmax;
if (rank == 0) {
for (int i = 1; i < nproc-1; i++) {
MPI_IRecv(&localmax, 1, MPI_LONG, i, 123, MPI_COMM_WORLD, &
   request);}
globalmax = findMax(localmax, globalmax);
un_calculo(globalmax);
MPI_Wait(&request, &status);
} else {
MPI_Isend(&localmax, 1, MPI_LONG, 0, 123, MPI_COMM_WORLD, &
   request);
otro_calculo();
MPI_Wait(&request, &status);
}

}
```

Reformule el código anterior con directivas MPI de comunicación colectiva, tal que las operaciones entre procesos estén sincronizadas. Y compare ambos métodos indicando cuál considera es mejor.

```
for(i=0;i<10;i++){
    A[i]=rand()%100;
}
int local_max=A[0];

for(i=0;i<10;i++){
    if(A[i]>local_max){
        local_max=A[i];
    }
}

int global_max;
MPI_Reduce(&local_max, &global_max, 1, MPI_INT, MPI_MAX,
    0,MPI_COMM_WORLD);
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Método o algoritmo	Describe al algoritmo de solución del problema planteado en forma adecuada (2 pts)	Algoritmo con algunos errores que no afectan el resultado (1.5 pts).	Algoritmo con errores que afectan minimamente el resultado (0.5 pt).	EAlgoritmo con errores, que afectan significativamente el resultado (0 pts)
Resultados	Solución correcta usando un método adecuado (1 pt)	Errores mínimos en el método que no afectan el resultado (0.6 pts)	Errores en el método que afectan el resultado (0.3 pts)	No aplica el método ni llega a la solución correcta (0 pts).
Optimización	Solución original y optimizada (1 pt)	Solución parcialmente optimizada (0.6 pts)	Solución original pero no optimizada (0.3 pts)	Resultado encontrado no está optimizado (0 pts).