Luis Miguel Benitez Ruiz

994361

Tarea 3

Analisis de rendimiento.

Es muy importante evaluar el rendimiento de un algoritmo paralelo, puesto que no siempre valdrá la pena el esfuerzo de su implementación si la mejora en el tiempo de ejecución no es la esperada. La mejora que se obtiene por paralelizar un algoritmo se define de forma sencilla:

$$Mejora = \frac{Tiempo\ de\ ejecucion\ secuencial}{Tiempo\ de\ ejecucion\ paralelo}$$

Esto representa la proporción en la que un algoritmo paralelo es más rápido con respecto a la versión secuencial. Una mejora de 2, por ejemplo, significa que el algoritmo paralelo es dos veces más rápido que el algoritmo secuencial.

Sin embargo, esta medida es demasiado optimista, pues no considera que hay costos inherentes a la paralelización de la ejecución del programa, pues tanto los procesos concurrentes como las unidades de procesamiento que se utilicen deben ser coordinados y comunicarse entre ellos.

Una fórmula más completa es la siguiente:

$$\psi(n,p) = \frac{\sigma(n) + \varphi(n)}{\sigma(n) + {}^{\varphi(n)}/_{p} + \kappa(n,p)}$$

$\psi(n,p)$ representa la mejora al paralelizar un problema de tamaño n utilizando p procesos. $\sigma(n)$ representa el tiempo de ejecución de los segmentos secuenciales del algoritmo, y $\varphi(n)$ el tiempo de ejecución de los segmentos paralelizables, y $\kappa(n,p)$ es el tiempo utilizado para comunicar y coordinar los procesos. El denominador representa el tiempo de ejecución en paralelo, donde se aprecia que conforme p (el número de procesos) aumenta, la sección paralelizable del código se hará en cada vez menos tiempo, sin embargo, $\kappa(n,p)$ aumentara conforme aumente p, por lo que se puede llegar a un punto en que aumentar procesadores acabe por reducir la mejora obtenida.

A partir de esta expresión puede derivarse una métrica muy común, llamada la "Ley de Amdhal", que permite estimar la proporción de mejora en tiempo que se tendría al paralelizar un algoritmo.

$$\psi \leq \frac{1}{f + {}^{(1-f)}/_{p}}$$

Donde $f = \frac{\sigma(n)}{\sigma(n)+\varphi(n)}$ representa la fracción secuencial de toda la ejecución.

La desigualdad surge del hecho de que $\kappa(n,p) \geq 0$ por lo que

$$\psi(n,p) \leq \frac{\sigma(n) + \varphi(n)}{\sigma(n) + {}^{\varphi(n)}/_{p}}$$

La medida obtenida por la ley de Amdhal expresa la mejora máxima posible que se puede obtener para un problema si se ejecutara en p procesadores. También puede utilizarse para obtener el límite de la mejora conforme aumentan los procesadores. Hay que considerar que para poder obtener esta mejora en la implementación real del algoritmo, habría que tener cero costos de comunicación, pues esta expresión no los toma en cuenta. También es importante notar que esta expresión ya no depende de n, pues f es solo un porcentaje, expresado entre 0 y 1, de la totalidad del tiempo de ejecución.

Existe otra medida de rendimiento llamada la Ley de Gustafson-Barsis. En contraste con la ley de Amdhal, esta métrica evalúa el rendimiento de un programa paralelo en base al tamaño del problema, y trata el tiempo de ejecución como una constante. Se utilizan dos expresiones mas, que representan el tiempo que se utiliza en ejecutar tareas secuenciales durante una ejecución paralela (s) y el tiempo utilizado en tareas paralelas durante una ejecución paralela (1-s)

$$s = \frac{\sigma(n)}{\sigma(n) + {}^{\varphi(n)}/_{p}}$$

$$1 - s = \frac{{}^{\varphi(n)}/_{p}}{\sigma(n) + {}^{\varphi(n)}/_{p}}$$

Y

$$\sigma(n) = (\sigma(n) + {}^{\varphi(n)}/_{p})s$$

$$\varphi(n) = \left(\sigma(n) + {}^{\varphi(n)}/_{p}\right)(1 - s)p$$

Substituyendo en la expresión de la desigualdad de $\psi(n, p)$

$$\psi(n, p) \leq \frac{(\sigma(n) + {}^{\varphi(n)}/_{p})(s + (1 - s)p)}{\sigma(n) + {}^{\varphi(n)}/_{p}}$$

$$\psi(n, p) \leq p + (1 - p)s$$

Como se explicó anteriormente, s representa la fracción de tiempo de ejecución utilizado en ejecutar tareas secuenciales durante una ejecución paralela con p procesadores. Mientras que la ley Amdhal expresa cuanto se podría acelerar un algoritmo paralelo con respecto a uno secuencial, esta métrica expresa como mejora un algoritmo ya paralelizado conforme el número de procesadores aumenta.

Una métrica que si toma en cuenta el costo de comunicación entre procesadores es la metrica de Karp-Flatt. Este sirve para encontrar la limitante más importante para la mejora en rendimiento del algoritmo paralelo contra el algoritmo secuencial. Partimos de la definición del tiempo total de ejecución en paralelo con p procesadores:

$$T(n,p) = \sigma(n) + {}^{\varphi(n)}\!/_{p} + \kappa(n,p)$$

Un proceso serial con 1 procesador no tiene costos de comunicación por lo que:

$$T(n,1) = \sigma(n) + \varphi(n)$$

Entonces la medida de mejora se expresa:

$$\psi(n,p) = {}^{T(n,1)}\!/_{T(n,p)}$$

Y se define la 'fraccion serial experimentalmente determinada':

$$e = {}^{\sigma(n) + \kappa(n,p)}\!/_{T(N,1)}$$

$$T(n,1)e = \sigma(n) + \kappa(n,p)$$

La función

$$T(n,p) = T(n,1)e + {}^{T(n,1)(1-e)}\!/_{p}$$

Utilizando $T(n,1) = T(n,p)\psi(n,p)$ se puede simplificar en:

$$e = \frac{{}^{1}\!/_{\psi} - {}^{1}\!/_{p}}{1 - {}^{1}\!/_{p}}$$

Utilizando los datos obtenidos experimentalmente para ψ y p, pueden usarse las expresiones de la fracción serial para evaluar el impacto l costo de comunicaciones y del tiempo necesario para la ejecución de las tareas secuenciales en la mejora obtenida por el algoritmo.

Como puede observarse, hay diferentes formas de medir el rendimiento de un algoritmo paralelo, y cada una se enfoca en aspectos diferentes pero importantes de la ejecución del algoritmo paralelo, y sirven para propósitos distintos, por lo que es importante evaluar la métrica adecuada para lo que se quiera lograr.

Finalmente, exista una métrica sencilla llamada eficiencia, que indica que tan bien se están aprovechando los procesadores paralelos.

$$Eficiencia = \frac{Mejora\ en\ tiempo}{Numero\ de\ procesadores}$$

Referencias

Quinn, M. J. (2003). *Parallel Programming in C With MPI and OpenMP.* International Edition: McGraw Hill Higher Education.