CORRECCIÓN POR PARES

COMPILADORES E INTERPRETES

25015

Aspectos positivos

La optimización se explica de forma clara, siendo fácil de entender.

Se utilizan múltiples gráficas para representar los datos, ofreciendo el porcentaje de mejora del código optimizado respecto al no optimizado, observando como evoluciona la mejora con N.

Añade el código compilado con O3. De esta forma tenemos 3 niveles de rendimiento y podemos comparar la optimización con un código peor y uno mejor.

Aspectos negativos

En la optimización se identifica claramente un segundo factor que afecta a su rendimiento: **el factor de unroll**. No obstante, no se ha realizado ningún estudio sobre este parámetro y se ha considerado un valor de 4 para el experimento, sin explicar las razones para escoger este valor (por lo que consideramos que es arbitrario), cuando su modificación influye directamente en el rendimiento del código. En la conclusión se habla del rendimiento asociado a este factor, pero al no realizar el estudio y no razonar el porqué, solo podemos considerarlo como suposiciones.

Tampoco se ha realizado un estudio inicial sobre los tamaños de N a utilizar, escogiendo tamaños relativamente pequeños, pero el rendimiento puede no ser mejor si se aumenta el tamaño de N.

En la conclusión se dan una serie de razones del porqué de la mejora, pero no se llegan a citar todas las recogidas en el documento, como puede ser el paralelismo a nivel de instrucción. Además, el razonamiento es muy pobre a lo largo del documento y, aunque sea cierto, se echa en falta un razonamiento más con el que poder estar de acuerdo.

No se exponen las características de la máquina que ha ejecutado el código.

Puntuación del documento

Según las observaciones expuestas, se considera que el trabajo tiene una puntuación de **-1** (**ligeramente peor** con respecto al nuestro).

Comentarios adicionales

En la revisión del código ensamblador sería interesante realizar una comparación del número total de saltos presentes en el código. De esta forma se podría relacionar como el reducir el número de iteraciones en el código afecta a reducir el número de saltos y por tanto mejora el rendimiento.

Martín Suárez García
COMPILADORES E INTERPRETES