



Desarrollo ágil de aplicaciones científicas usando arquitecturas de bajo consumo energético.

Jose Maria Zamora¹

¹ Lufac Computación® S.A. de C.V.

e-mail: josema_zamora@lufac.com

En los últimos años, han emergido una gran cantidad de arquitecturas computacionales alternativas, debido al inminente alcance de la frontera en la velocidad de las unidades CPU y los altos costos de las mismas. En este sentido, las tecnologías de punta están dirigidas hacia las unidades de procesamiento gráfico (GPUs). Los GPUs requieren de APIs y librerías para poder ser utilizados por los usuarios finales, y estas implementaciones están construidas en lenguaje C/C++. Esta condición deja de lado a un sector importante de la comunidad científica que sigue desarrollando software en Fortran. Además el conocimiento sobre el uso de las funciones en lenguaje C/C++ para acceder al hardware de las GPUs, cada vez es más específico (SDK con centenares de funciones para controlar un GPU).

Así pues, acelerar un código usando GPUs, involucra varios factores: 1) conocimientos especializados en programación con C/C++ (punteros), 2) conocimiento especializado en arquitecturas de cómputo (Stream Multiprocessor, y bloques de memoria), 3) usar ingeniería de software en desarrollos científicos.

Estas condiciones motivan a que muchos investigadores abandonen la idea de acelerar sus códigos de manera rápida y eficaz usando GPUs. Es por eso, que actualmente en la empresa se implementan las herramientas más modernas para poder aliviar este alto costo en tiempo de programación y especialización, para acelerar software científico.

Nuestra propuesta parte del uso del standard de directivas OpenACC, las cuales nos permiten acelerar aplicaciones con tan solo añadir una decena de líneas de código. Durante la presentación, se detallará: 1) como funciona el **standard OpenACC** y cual es el flujo de trabajo para poder usarse, 2) se mostrarán ejemplos de programación de problemas numéricos reales usando los lenguajes **Fortran y C/C++** (plegamiento de proteínas con el model HP), 3) se **compararán** las capacidades de aceleración de CUDA y OpenACC, 4) y finalmente se mostrará como paralelizar una aplicación de **Dinámica Molecular** en muy poco tiempo y usando el mínimo de líneas de código.