ARCANOS Y FUTUROS DE LA COMPUTACIÓN

Juan Carreón es profesor titular de Sistemas Inteligentes en la FI de la Universidad Nacional Autónoma de México (UNAM). Se le puede seguir en http://twitter.com/carreonG

Una hipótesis productiva para comprender el cómputo, es que su aspecto central tiene que ver con la programación (software), y ésta con los lenguajes de programación.

Esa hipótesis se comprueba en cierta medida al considerar el estancamiento actual de la velocidad de reloj de los chips de microprocesadores de PC (en la cercanía de los 3.0GHz., o tres mil millones de "ticks" por segundo) a lo largo de los cuatro años recientes.

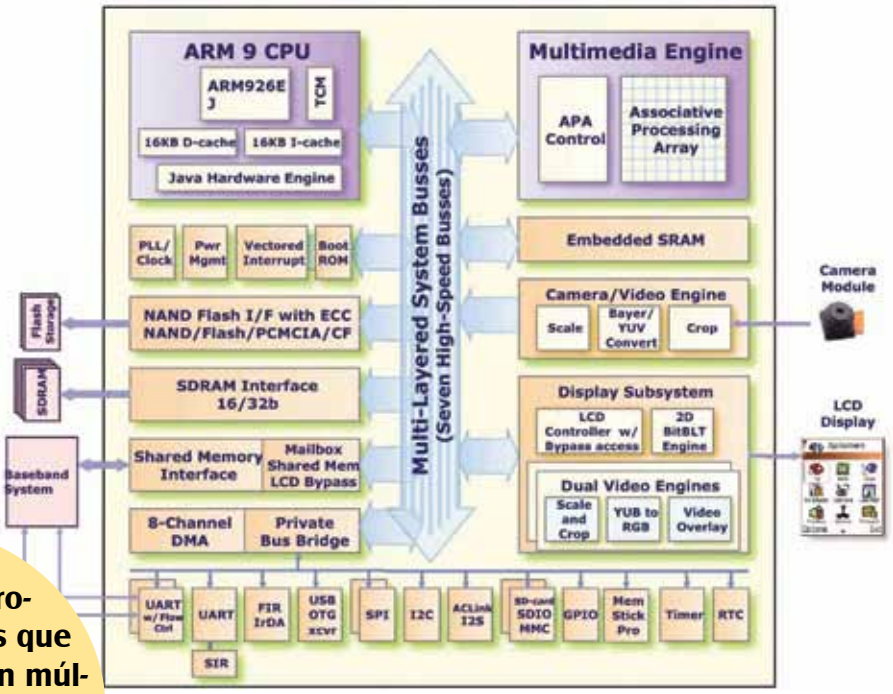
El logro de programas eficientes que corran paralelos en múltiples núcleos ha resultado una tarea más difícil de lo que se pensaba; ¿entonces?

Si previamente a lo largo de varias décadas la velocidad de esos chips se había estado duplicando cada dos años aproximadamente, en años recientes la dificultad de enfriar los procesadores y su pérdida de eficiencia energética ha hecho imposible los incrementos adicionales de velocidad.

Tal hecho se ha buscado superar empacando múltiples máquinas de procesamiento en un solo chip, de ahí que las PC y laptops actuales típicamente dispongan de procesadores dobles (p. e., Intel Core i3) o cuádruples, entre otros.

Esto no significa que la velocidad de realización de tareas empleando dichas computadoras se duplique o cuadruplique, ya que desafortunadamente la mayoría del software se desarrolla para correr en chip con un solo núcleo.

No porque no existan lenguajes de programación para el diseño de programas que corran en paralelo en múltiples núcleos, sino debido a que lograr programas paralelos eficientes y factibles de ser corregidos y compilados ha resultado una tarea más difícil de lo que inicialmente se pensaba.



El procesamiento paralelo es clave para la televisión digital móvil y para otros dispositivos multimedia de próxima generación (Imagen: Pen Computing Magazine, http://ow.ly/6e9AN)

Tres enfoques

Un enfoque es desarrollar en paralelo máquinas de alto desempeño que, al mismo tiempo que sean más poderosas, sean más fáciles de programar, así como herramientas de apoyo a la programación más eficientes.

Otra perspectiva ha sido adaptar lenguajes existentes como C++ y Fortran al desarrollo de código paralelo, así como herramientas de apoyo de este enfoque, camino seguido por Intel y Microsoft.

Otro enfoque ha surgido de lenguajes empleados más en ambientes académicos, típicamente desde un enfoque funcional apoyado en un estilo de programación matemático, asimilando los programas a funciones matemáticas, con lenguajes como Erlang, Haskell, Scala y Scheme*.

Culturas en choque

Este último enfoque diferente al tradicional de los lenguajes imperativos, con base en comandos, choca con la cultura que ha sido hasta ahora la convencional para la mayoría de los programadores.

Sin embargo, este último enfoque ha sido adecuado para la programación paralela de grandes equipos de telecomunicaciones, o para manejar el chat en Facebook; Scala que combina aspectos de los lenguajes funcionales con el de los tradicionales, es empleado en Twitter, LinkedIn y Foursquare, entre otros.

Conclusión

Algunos sostienen que lo más probable es que las técnicas de programación que finalmente se vuelvan dominantes en la programación paralela sean una combinación de los tres enfoques mencionados**.

En todo caso en los arcanos de los lenguajes de programación reside la computación del futuro.

*The Design and Implementation of a Dialect of Scheme for Parallel Processing on the GPU, http://www.engineeringtv.com/video/The-Design-and-Implementation

**Parallel bars, The Economist, Technology Quarterly: Q2 2011.