

El impacto de la nueva generación de Microprocesadores en la Ingeniería de Software

M. en C. Miguel A. Partida Tapia
Subdirector Académico y de Investigación del CINTEC-IPN.

La Ingeniería de Software ha tenido una evolución muy importante durante los últimos diez años; sin embargo, en relación al desarrollo del Hardware, se mantiene y en algunos casos se amplía un margen de retraso de por lo menos 5 años. Este desfasamiento se profundizó principalmente por la llegada de las computadoras personales y su alta demanda en las diversas áreas del conocimiento y por el tipo de usuarios.

Los principales avances del Hardware se han enfocado al desarrollo de microprocesadores tipo RISC ("Reduced Instruction Set Code", Conjunto Reducido de Instrucciones), Coprocesadores Gráficos y Almacenamiento Masivo de Datos. Un ejemplo de esto es el microprocesador i860 de Intel Corp., el cual integra una unidad de procesamiento tipo RISC, coprocesador numérico y coprocesador gráfico y, adicionalmente, capacidad para un alto ancho de banda de "bus" para transferencia de datos. Estas características de los microprocesadores han generado nuevas alternativas de procesamiento de datos y comunicaciones, principalmente el desarrollo de Multimedia, y en particular la capacidad de realizar comunicaciones de voz e imagen en tiempo real; como con-

secuencia, la comunicación vía satélite y la encriptación de la información en tiempo real es un hecho común en nuestros días.

La capacidad de estos microprocesadores ha quedado de manifiesto, sin lugar a dudas; sin embargo, todas estas aplicaciones son lo que se denomina, "Soluciones Integrales", donde los Ingenieros de Hardware y de Software no tienen más que aplicar esta solución y esperar los resultados. De este modo, la solución esperada estará acotada en las ventajas y limitaciones definidas por los diseñadores de estas "Soluciones Integrales". Para el presente análisis se partirá del caso en donde la arquitectura de la "Solución Integral" permita que diseñadores de Software desarrollen aplicaciones para resolver un tipo de problema en particular de la institución receptora.

A continuación se muestran algunas de las características que se considera limitan el desarrollo de Software:

a).- De la Arquitectura de Computadoras.

- o Arquitectura del microprocesador.
- o Tiempo de Acceso a memoria.
- o Ancho de banda del "bus".
- o Ubicación en la arquitectura de los periféricos.

- o Características de los dispositivos periféricos.
- o Capacidad de procesamiento de la unidad de punto flotante.
- o Capacidad de manejo de memoria principal y virtual.

b).- Del Compilador.

- o Capacidad de migración de código a otros sistemas.
- o Capacidad de compactación de código.
- o Capacidad de generación de código utilizando las características del microprocesador.

c).- Del Sistema Operativo.

- o Capacidad de administración de recursos.
- o Capacidad de administración de dispositivos periféricos.
- o Velocidad de respuesta en sus rutinas de atención.

Con lo anterior se puede concluir que todo desarrollo de Software en una computadora específica se encuentra en dependencia directa de los alcances y limitaciones de su arquitectura, del compilador en que se desarrolló la aplicación y del sistema operativo en que trabajará dicho Software.

Un estándar en los diseñadores de Software se ha establecido en el lenguaje C de programación, mismo que se encuentra en el mer-

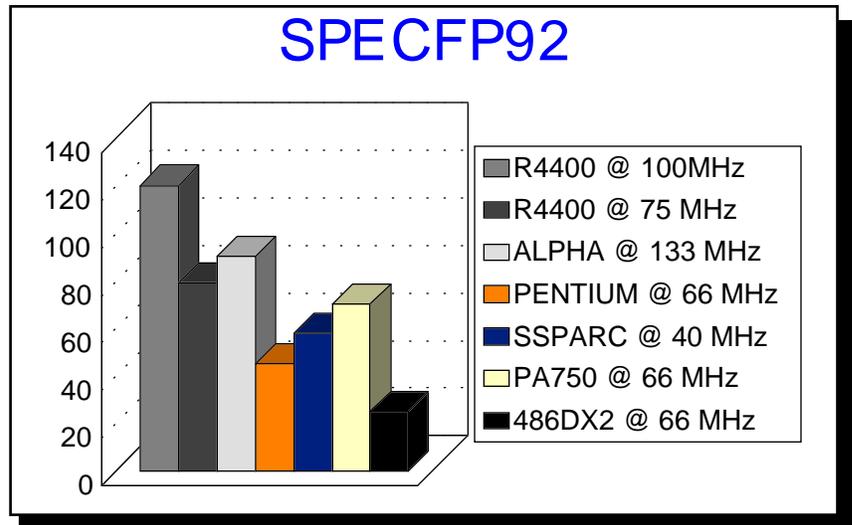
cado con diversas facetas, desde un seudo lenguaje de alto nivel (como originalmente fue propuesto), hasta aplicaciones en Programación Orientada a Objetos. Este lenguaje tiene su fundamento en la transportabilidad, sin embargo, solo ocurre en aquellas funciones que han sido normalizadas (ANSI). Dicha estandarización, en el mejor de los casos, alcanza hasta un 40% de las funciones que comprende el compilador, el resto se sujeta a funciones que cada fabricante ha definido por su cuenta y que tienen cierta similitud a otros compiladores que existen en el mercado, pero no son compatibles.

La mayoría de los compiladores mantienen un esquema de compatibilidad hacia "atrás" (versiones anteriores), obligando con esto a manejar criterios de compilación y generación de código bajo este esquema. El compilador generará código para un 8088 igual que para un 80486, con solo cambiar una bandera en la solicitud de compilación, sin embargo, esto no basta, sino que se hace necesaria una optimización del mismo para no limitar el rendimiento ("Performance") del microprocesador. No obstante, hasta el momento no existen aún, en la línea Intel, compiladores que optimicen suficientemente el código como para poder establecer un máximo rendimiento de la nueva generación de microprocesadores.

Como anteriormente se expuso, la característica de los nuevos microprocesadores RISC establece la posibilidad de ejecutar la mayoría de sus instrucciones en uno o dos ciclos de reloj y realizar una o más instrucciones en este mismo ciclo; dicho de otra manera, se tiene la posibilidad de paralelizar ciertas operaciones de alto nivel. Lo anterior implica la necesidad de

que, al diseñar el Software, sea analizado desde un punto de vista en el que la arquitectura del microprocesador es óptima, esto es, que cada procedimiento pudiese ser paralelizable en mayor medida. En las gráficas que se muestran a continuación, se establece un análisis

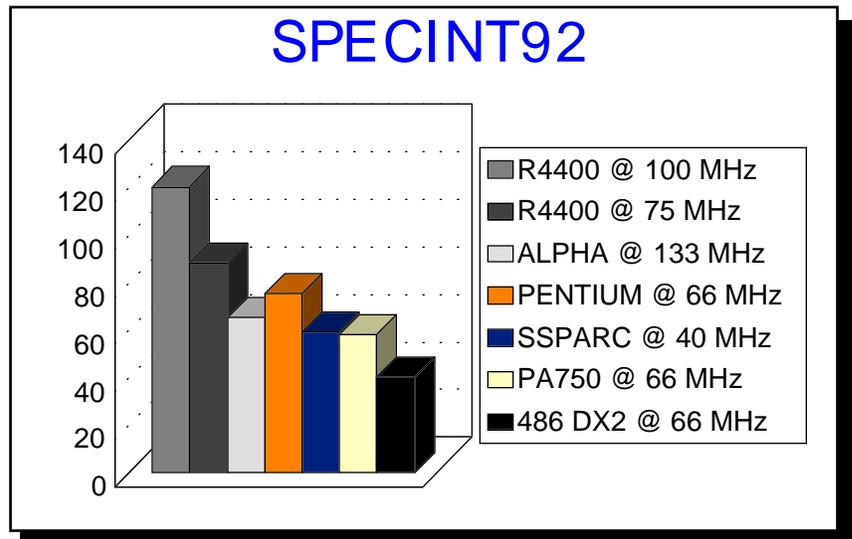
(SPECINT92). Además, se usó el sistema operativo UNIX de Silicon Graphics y la versión más reciente de NT WINDOWS. En algunos casos, como los microprocesadores R4400 y ALPHA, sus resultados fueron caracterizados por emulación, dado que algunas rutinas, prin-



de los diversos tipos de microprocesadores, donde se puede observar la relación entre precio/rendimiento. Dicha evaluación se efectuó con Software realizado por la Standard Performance Evaluation Corp., el cual mide las capacidades de rendimiento en operaciones de punto flotante (SPECFP92) y enteros

principalmente de NT WINDOWS, solo son nativas en microprocesadores de INTEL Corp. En este caso la pérdida de rendimiento fue ajustada para la tabla comparativa (1).

De lo anteriormente descrito, se puede expresar que el rendimiento de un diseño de Software se en-



cuentra definido por la siguiente expresión:

$$R_{SW} = 1 - (L_{AC} + L_{CMP} + L_{SO}).$$

donde R_{SW} se define como el rendimiento de diseño de Software.

L_{AC} = Las limitaciones de la arquitectura de la computadora donde va a ser ejecutada la aplicación,

L_{CMP} = La limitación del compilador en obtener el máximo rendimiento en la generación de código y

L_{SO} = Las limitaciones del sistema operativo para entregar y administrar los recursos requeridos por la aplicación.

El cálculo de rendimiento para una arquitectura de computadora dada se podrá determinar con el siguiente ejemplo. Sea:

T_{AC} = El tiempo total en segundos de una rutina completa, medida en base a los tiempos reportados por instrucción a ejecutar y

T'_{AC} = El tiempo total en segundos de una rutina completa ejecutada, medida en tiempo real.

$$R_{AC} = 1 - (T'_{AC} - T_{AC}) / T_{AC}$$

donde:

$$L_{AC} = 1 - R_{AC} = (T'_{AC} - T_{AC}) / T_{AC}$$

El rendimiento R_{SW} del Software, independientemente de las variables anteriormente expuestas, se verá afectado por variables intrínsecas al Ingeniero diseñador de Software, como son:

- o Metodología de diseño empleada.
- o Modularidad.
- o Dependencia de datos.
- o Verificación de transferencias entre módulos.
- o Criterios de Auditoría Informática a establecer en el sistema.
- o Requerimiento de uso de rutinas de "tiempo real"
- o Requerimiento de almacenamiento masivo.

- o Velocidad de las transferencias en el almacenamiento de datos.
- o Menú de interface.
- o etc.

De la misma forma pero en sentido contrario, estas variables definen los criterios particulares del tipo de arquitectura de la computadora, compilador y sistema operativo que habrá de seleccionarse.

Bibliografía

- [1] Child, J. "Pentium price/performance clouded by Software issues", Computer Design, May 1993, pp 28-30.
- [2] Intel Corp. "Procesador Pentium, Descripción General de Rendimiento", Notas Técnicas, Versión 2, Agosto 1993.
- [3] White, B. "Programming Techniques for Software Development", Stanford University, ISBN 0-442-29187-6, Edit. Van Nostrand Reinhold, 1989.