

Discurso de aceptación

21 de septiembre de 2021

David A. Patterson, galardonado en la categoría de Tecnologías de la Información y la Comunicación (XIII edición)

John Hennessy y yo estamos muy agradecidos a la Fundación BBVA y al jurado de colegas de nuestra profesión que nos ha seleccionado para este prestigioso premio. Nos ha emocionado mucho recibirlo.

Para entender nuestra contribución, explicaremos algo de jerga. Cuando el software habla con el hardware, utiliza un vocabulario llamado “conjunto de instrucciones”. Ejemplos de palabras o “instrucciones” de este vocabulario son las teclas de una calculadora: sumar, multiplicar, etc. Los programas se componen de millones de estas sencillas instrucciones ejecutadas miles de millones de veces.

En 1980, cuando Hennessy y yo éramos assistant professors, la opinión más extendida era que muchos problemas de la tecnología de la información se debían a que los vocabularios de los ordenadores eran de muy bajo nivel. Estos conjuntos de instrucciones al parecer representaban una carga considerable para los programadores, lo que daba lugar a errores de software y proyectos de software fallidos. La filosofía predominante era elevar el nivel de abstracción y crear vocabularios de instrucciones complejas para reducir la distancia entre las personas y las máquinas.

En la década de 1970, solo había microprocesadores en los electrodomésticos. Hennessy y yo creíamos en la ley de Moore, la extraordinaria predicción de que el número de transistores en cada chip se duplicaba cada año o cada dos años. Estábamos convencidos de que los microprocesadores se convertirían en la base de toda la computación. La pregunta era cuál era el mejor vocabulario o conjunto de instrucciones para estos microprocesadores en rápido progreso. ¿Debía ser diferente al de las grandes computadoras que impulsaban la industria en los años 70?

Para responder esta pregunta crítica sobre la interfaz entre el software y el hardware, necesitamos más contexto. Era tan pesado programar en el lenguaje de máquina de bajo nivel que los pioneros de la computación enseguida inventaron lenguajes de programación de alto nivel y uso más fácil, y además programas llamados “compiladores” que traducían de los lenguajes de alto nivel al lenguaje de máquina. Aunque los lenguajes de alto nivel gozaban de gran aceptación para muchas tareas, los compiladores eran considerados demasiado ineficientes para ciertos softwares, como los sistemas operativos. A finales de los años 70, el éxito del sistema operativo UNIX, escrito en un lenguaje de alto nivel, convenció a la gente de lo contrario.

21 de septiembre de 2021

Respecto a los conjuntos de instrucciones de los microprocesadores, la carga que suponía para los programadores escribir en lenguaje de máquina de bajo nivel ya no era un problema. ¿Pero cuál era la dificultad de los compiladores para producir programas de calidad en lenguaje máquina?

A las grandes computadoras con vocabularios acordes a la extendida preferencia por los “ordenadores con conjunto de instrucciones complejas” los denominamos CISC para abreviar, pronunciado “sisk”. Propusimos que los microprocesadores utilizaran en su lugar vocabularios con instrucciones sencillas, a los que denominamos “ordenadores con conjunto de instrucciones reducido”, abreviado RISC y pronunciado “risk”. Piensen que CISC tiene muchas palabras polisílabas, mientras que las palabras de RISC son monosilábicas. Hennessy y yo pensamos que sería más fácil construir microprocesadores que entendieran vocabularios RISC que vocabularios CISC.

La pregunta entonces era cuál de ellos, RISC o CISC, era el diseño más rápido. Aunque los programas tuvieran que leer menos instrucciones con CISC, al ser esas instrucciones más complejas, la instrucción media exigiría más tiempo para leerse y ejecutarse que una instrucción RISC. La analogía es que una página llena de palabras polisílabas podría tardar más en leerse que una página de palabras casi todas monosílabas.

Esta pregunta científica, aparentemente clara, estaba cargada de emoción en el mundo del diseño de ordenadores, lo que llamamos la “arquitectura de los ordenadores”. Los partidarios de CISC creían que RISC era un paso equivocado y que añadiría complejidad al software. Los partidarios de RISC veían obsoleto este argumento y pensaban que los compiladores podrían ocultar esas diferencias de forma que los programadores no las notaran. Aunque Berkeley y Stanford son instituciones rivales —como el Barcelona y el Real Madrid en fútbol—, Hennessy y yo comprendimos que debíamos unir fuerzas para defender la arquitectura RISC. En un congreso internacional celebrado en Silicon Valley en 1982 se organizó el primero de varios debates sobre el diseño RISC frente al CISC. Las acaloradas conversaciones continuaban en los pasillos y se prolongaban hasta la noche en aquel evento, al que siguieron varios debates más a lo largo de los dos años siguientes.

Max Plank dijo que la verdad científica no triunfa convenciendo a los oponentes y haciéndoles ver la luz, sino que la ciencia avanza a golpe de entierros. Afortunadamente, en la arquitectura de ordenadores hay una industria que nos acompaña y que pone a prueba nuestras ideas en el mercado, con lo que no tenemos que esperar a ningún entierro para avanzar el campo.

Para responder cuál es mejor, RISC o CISC, necesitábamos descubrir las proporciones entre el promedio de instrucciones que un programa necesita leer —para CISC debería ser menor que para RISC— frente al promedio de tiempo para leer una instrucción —las instrucciones de RISC deberían ser más rápidas que las de CISC—. Acabamos descubriendo que RISC lee algunas instrucciones más, pero es tan rápido que el resultado neto es que su velocidad es de 3 a 4 veces mayor que la de CISC. Es más, los microprocesadores



21 de septiembre de 2021

RISC necesitaban menos hardware que los CISC. Estas ventajas se confirmaron en el mercado informático, y los procesadores RISC dieron muy buenos resultados a finales de los años 1980 y principios de los 1990.

Intel disponía de una arquitectura CISC impulsada por el ordenador personal de IBM; una valiosa franquicia. También tenían la mejor fabricación de semiconductores del mundo, al menos dos años por delante de todos los demás, lo que implicaba al menos el doble de transistores, según la Ley de Moore. Es reseñable que utilizaran esos transistores adicionales de sus microprocesadores para traducir internamente en el hardware miles de millones de veces por segundo las instrucciones de CISC a instrucciones de RISC, con lo que podían ejecutar los programas antiguos, pero a velocidades de RISC. El diseño RISC de Intel vestido de CISC dominó el mercado de los ordenadores personales y después el de las computadoras de mayor tamaño durante la década de 2000.

El iPhone de 2007, basado en RISC, abrió un nuevo mercado que creció más que el de los ordenadores personales. Cuando la potencia se convirtió en el principal limitador de los chips y la ley de Moore empezó a ralentizarse, la ventaja de RISC en eficiencia (tanto en el uso de transistores como en la potencia) que llevó a utilizarlo en el iPhone se agrandó aún más, aupándolo a una posición todavía más hegemónica. En la última década y en el futuro previsible, el 99% de los microprocesadores utilizan RISC.

Para concluir nuestra historia, la arquitectura de ordenadores es un deporte de equipo, por lo que el premio BBVA supone un reconocimiento a nuestros maravillosos colaboradores y antiguos alumnos de Berkeley y Stanford tanto como a nosotros. También agradecemos a nuestras esposas y familias sus 50 años de apoyo, desde la escuela de postgrado y durante nuestras largas carreras de investigación; de ninguna manera habríamos podido hacer estas contribuciones sin ellas a nuestro lado.

Mi colaborador y buen amigo de tantos años John Hennessy contará el resto de nuestra historia, que es cómo el avance de RISC llevó a la creación de empresas startup e inspiró una revolución en la forma de enseñar la arquitectura de los ordenadores.