Resumen de John Hennessy and David Patterson 2017 ACM A.M. Turing Award Lecture

Dalia Camacho García Formentí

March 2019

La conferencia dada por Henessey y Patterson, ganadores del premio Turing 2017, se divide en tres partes. En la primera hacen referencia a la historia de los sets de instrucciones que abarca desde las guerras por generar un número mayor de microinstrucciones, la creación de RISC y sus beneficios y el fracaso de VLIW que llevó nuevamente a la arquitecura RISC. En la segunda parte hablan sobre los retos actuales en arquitectura de computadoras que tienen que ver con el final de la ley de Moore, un alto consumo energético, problemas de seguridad y el interés de seguir empujando los límites de eficiencia. La última parte está enfocada en RISC-V que es un set de instrucciones de tipo *open source* y cómo tener una arquitectura abierta permite obtener mejoras más rápidamente, ya que se aprovecha la experiencia de muchos colaboradores. En las siguientes secciones describiré más puntualmente lo que se abordó en caa parte de la plática.

Historia

A principios de 1960 IBM tenía cuatro líneas de computadoras incompatibles, para resolver este problema se buscó crear un set de instrucciones basado en microprogramación que lograra resolver el problema. Este set de instrucciones fue el IBM 360. A partir de esto y gracias a la ley de Moore se buscó tener cada vez más microinstrucciones que generaban sets de instrucciones más grandes y complicados dando lugar a las arquitecutras tipo CISC. La arquitectura VAX, por ejemplo tenía más de 5,000 microinstrucciones.

Al utilizar la RAM para el área de control se podían agregar microoperaciones, ya que el área de control era escribible (*Writable Control store*). Con esto la microprogramación ganó importancia en el ámbito científico. La computadora Xerox Alto creada en 1973 fue la más relevante en términos de microprogramación ya que fue la primera en tener interfaz gráfica y ethernet y estaba completamente progrmada con microcódigos.

Aumetar el número de instrucciones complicadas tuvo sentido en ese momento, ya que se programaba en lenguaje ensamblador y esto hacía más sencilla la programación.

Intel intentó crear un set de instrucciones muy ambicioso, el cual esperaban que fuera el set que se utilizaría durante muchis años, sin embargo el set era

tan complicado que tenía un mal rendimiento y las instrucciones no cabían en un solo chip. Fue por esto que tuvieron que recurrir a un plan de emergencia en el que en un año crearon el ISA 8086 que era compatible con el 8 bit 8080. Aunque el proyecto de Motorola fue mejor IBM eligió el 8086 de Intel ya que lo entregaron en tiempo.

Cuando se empezó a programar en lenguajes de alto nivel se le empezó a dar importancia a los compiladores y John Cocke obtuvo ejecuciones 3 veces más rápidas utilizando únicamente instrucciones sencillas en vez de usar las instrucciones complicadas.

Otro problema de CISC era la dificultad que se tenía para encontrar errores en el microcódigo, a partir de esto se va dando el salto de CISC a RISC. Además de que eran instrucciones más sencillas requerían menos espacio en SRAM para el microcódigo, con esto pudieron usar ese espacio en los chips para agregar caches y también se mejoró la eficiencia en el uso de los registros.

Los grupos de Henessey en Stanford y de Patterson en Berkley impulsaron el desarrollo de arquitecturas más sencillas la MIPS y la RISC respectivamente. La eficiencia de estas arquitecturas en comparación con CISC puede comprobarse mediante la ley del acero, ya que a pesar de que CISC requiere menos instrucciones, éstas requieren más ciclos por instrucción. En total se tenía que RISC era 4 veces más rápido que CISC. Además lograron que un chip desarrollado por estudiantes fuera mejor que aquellos desarrollados por Intel. Debido a esto Intel se encargó de traducir las instrucciones del x86 en instrucciones en RISC.

En 1994 Intel junto con HP buscaron cambiar la arquitectura RISC por una arquitectura tipo VLIW para una arquitectura de 64 bits. Una motivación para hacerlo era que AMD no podría tener esta arquitectura y en cambio AMD sí podía hacer uso de las instrucciones x86. Esta nueva arquitectura tendría el nombre de EPIC y creyeron que podría ser la nueva gran arquitectura, sin embargo fue un fracaso ya que había problemas con los compiladores y no se podían predecir los *branches* ni los *misses* en caché. Los compiladores que hubiera necesitado VLIW para funcionar adecuadamente no podían ser programados.

Actualmente no se utilizan arquitecturas tipo CISC, las arquitecturas tipo VLIW se utilizan sólo en casos específicos en que se saba que funcionan bien, pero para propósito general RISC sigue siendo la arquitectura predominante.

Retos actuales

En el panorama actual es difícil seguir avanzando por medio de avances puramente tecnológicos, ya que la ley de Moore llegó a su fin y uno de los mayores problemas es el consumo de energía y poder necesarios los cuales también tienen un límite. Es por eso que se deben tener ideas nuevas que se puedan explotar, ya que las ideas que ayudaron a mejorar las arquitecturas como los caches llegaron a un límite en que tener más no mejora el rendimiento.

Se menciona como ha sido la evolución en cuanto a rendimiento a lo largo de los años y de haber tenido crecimientos de 22% anual, 52% anual y 23% anual,

actualmente el crecimiento es únicamente de un 3% anual.

Otro problema que consideran relevante es el de mejorar la seguridad, para esto proponen retomar ideas de 1970 que en su momento se descartaron, por que no había arquitecturas que permitieran llevarlas a cabo. Además se sabe que la verificación formal no se da como se debería y que el software está lleno de errores, por lo que habría que abordar el problema de seguridad desde el hardware. Más aún se debe considerar la seguridad ya que hay ataques que actúandirectamente sobre la arquitectura.

En la plática se abordan posibles soluciones para continuar avanzando, dentro de estas se encuentran tener una programación de software más cuidadosa y más enfocada hacia la eficiencia, generar arquitecturas específicas para aplicaciones específicas. Ejemplos de esto último son arquitecturas enfocadas en aprendizaje de máquina y en gráficos. Se menciona la *tensor processing unit* (TPU) que es una arquitectura específica para aprendizaje de máquia en la que el manejo de memoria es distinto y tiene un área del chip destinada a multiplicación de matrices.

También da énfasis en generar benchmarks a partir de los cuales se puedan comparar las distintas arquitecturas específicas. Además recalca la necesidad de generar una mayor comunicación entre los que hacen software y los que hacen hardware para así avanzar de la mejor manera, ya que los retos actuales requieren de una integración entre las distintas áreas de la computación.

Arquitecturas abiertas

En esta parte se habla sobre la arquitectura RISC-V, que aunque en un principio la idea principal era simplemente tener una arquitectura que no estuviera controlada por Intel, distintos grupos comenzaron a utilizarla por lo que cambió la forma en que se desarrolló en un principio para convertirse en una arquitectura abierta.

Las características principales de RISC-V son las siguientes

- Sencilla.
- Se empezó desde cero evitando errores del pasado.
- La base es sencilla, pero tiene lugar para posibles extensiones.
- Se puede especializar ya que hay códigos de operación disponibles.
- Se crea en comunidad.
- Los cambios son orientados a mejoras técnicas y no hacia marketing o ventas.

Se busca que RISC-V se convierta en el "Linux de los procesadores".

Además se ha logrado generar un flujo de trabajo distinto en el que se avanza más rápido a partir de desarrollar prototipos de procesadores intermedios, obtener retroalimentación y continuar con el desarrollo. Esto último se hace ahora en vez de pasar años en la fase del desarrollo.

A partir de esta nueva forma de trabajo y los retos actuales le llaman al momento ctual la nueva era dorada para la arquitectura en computadoras, ya que hay muchas posibilidades para desarrollar nuevas ideas e impulsar el avance en la computación.

References

[1] Association for Computing Machinery (ACM). John hennessy and david patterson 2017 acm a.m. turing award lecture, Jun 2018.