

Julio 6, 2010

El fin de la Ley de Moore: hora de volver a pensar

Categoría: [Sin categoría](#) — dccuchile - 3:59 pm

Por Gonzalo Navarro, profesor del Departamento de Ciencias de la Computación, FCFM, Universidad de Chile.

Gordon Moore, el cofundador de Intel, en 1965 observó el siguiente fenómeno acerca del desarrollo del hardware para computadores, el que luego se conoció como una "ley" que lleva su apellido: **el número de transistores que consigue el menor costo por transistor en un circuito integrado se duplica cada 24 meses**. Traducido al lenguaje de las consecuencias, esto decía que el poder de cómputo de los procesadores se duplicaba cada 24 meses (y no cada 18 como erróneamente se dice a veces). Esta ley se ha seguido manteniendo desde entonces con sorprendente fidelidad.



A lo largo de las décadas esta proeza bianual de la ingeniería, que nos hemos acostumbrado a aceptar como algo natural, ha modelado la forma de pensar de quienes conviven con la tecnología computacional. Es común escuchar comentarios como "no vale la pena invertir en programas más eficientes o mejores algoritmos; basta con esperar la próxima generación de chips". Esta confianza ni siquiera se ve minada por el crecimiento igualmente exponencial de la información disponible de manera digital en áreas como la Web, multimedia, bioinformática, datos astronómicos, redes de sensores, etc., además de las demandas de procesamiento cada vez más desenfadadas que conllevan.

Pues bien, este paraíso que nos ha convertido en indolentes dependientes de la tecnología parece acabar. Se está llegando a los límites fundamentales, al menos con la tecnología conocida, de la Ley de Moore. Basta prestar un poco de atención a las ofertas de PCs para notar que algo nuevo está sucediendo: se acabó la época en que los GHz de los procesadores crecían año tras año; hace ya algún tiempo que se han estancado alrededor de los 3 GHz. En cambio, cada vez hay más "cores", o unidades de procesamiento, por CPU. Esta es la nueva forma que la ingeniería ha encontrado para ofrecernos mayor poder de cómputo: más procesadores en vez de procesadores más potentes.

Este fenómeno está comenzando a cambiar drásticamente la forma en que pensamos acerca de la eficiencia computacional y cómo conseguirla. **Ya no se trata de correr el mismo software en la máquina recién comprada, sino que hay que reescribirlo para que aproveche el paralelismo al máximo (es decir, la capacidad de ejecutar acciones paralelamente en varios procesadores).** Los compiladores y sistemas operativos harán lo mejor que puedan para paralelizar automáticamente los programas, pero hay límites en esto.

El área de programación paralela se conoce desde hace mucho tiempo y nos ha mostrado que paralelizar eficientemente un programa puede ser una proeza intelectual, que no podemos esperar que se resuelva de forma automática. Cuantos más procesadores disponibles, más difícil es usarlos bien. **Por décadas los algoritmos paralelos se vieron como relativamente exóticos, enseñados en cursos muy avanzados y usados para grandes proyectos de cálculo masivo en supercomputadores. O a lo sumo, como parte del desarrollo de sistemas operativos (algo que muy pocos programadores ven en sus vidas). Hace algo de tiempo que esto dejó de ser verdad, al menos en la construcción de aplicaciones grandes.** Por ejemplo, Google y Yahoo! usan sistemas distribuidos gigantes (formados por miles de computadores) para indexar la Web y responder a las búsquedas de sus millones de usuarios en fracciones de segundo. Han desarrollado su propio lenguaje de programación para estos escenarios.

Pero la programación paralela amenaza con convertirse en lo cotidiano y cada vez con más computadores disponibles. Se está revigorizando el área de lenguajes de programación paralelos y el de algoritmos paralelos. **Pronto tendrán probablemente que repensarse los cursos de programación y de algoritmos más básicos en las carreras de Computación, mientras que los ya profesionales deberán ponerse al día.**

Pongamos un ejemplo para dar una idea de a qué nos enfrentamos. Suponga que quiere sumar un millón de números. Bueno, pues paciencia; a tomar una calculadora e ir sumando uno tras otro. Si un amigo se decide a ayudarlo, cada uno sumará quinientos mil números y luego los dos subtotalet, reduciendo así el tiempo a la mitad. Y si consiguen otros dos "digitadores", lo habrán comprimido a la cuarta parte. Pero si realmente hay muchos digitadores y calculadoras disponibles, la coordinación empieza a ser un tema relevante. Hay que decidir quiénes van a sumar qué números (nos tentaremos por darles más números a los digitadores más rápidos) y hacerles llegar su parte de la lista (no pueden estar todos al lado suyo por un problema espacial); hay que tener algún mecanismo para recolectar los resultados, seguramente en forma jerárquica (imagine que si son diez mil digitadores para sumar cien números cada uno, usted perdería más tiempo recolectando y sumando los diez mil subtotalet que ellos sumando su parte. Sería mejor entonces designar un "sumador parcial" cada cien digitadores y que esos se reporten con usted), etc. Y, según la aplicación, hasta es posible que algunos digitadores se desmayen "sin avisar" y haya que detectar estos eventos y reasignar su lista a otros.

Todo esto se refiere a un problema que es muy fácil de "paralelizar", es decir, el de sumar n números. Y los desafíos que describimos son más o menos estándares en computación paralela. Pero hay también desafíos algorítmicos; problemas que no son evidentes como paralelizar, o sea cómo aprovechar el hecho de tener

Archivos

- [El diluvio de datos](#)
- [Donald Knuth: El arte de la Computación](#)
- [Calidad: la encrucijada de la educación superior en Chile](#)
- [El fin de la Ley de Moore: hora de volver a pensar](#)
- [Educación, juventud y tecnología ¿Cuál es la calidad de nuestros estudiantes?](#)
- [Tecnología y usuarios ¿quién se adapta a quién?](#)
- [Conferencia de Computación: las tendencias en bases de datos](#)
- [Qué hace que un software sea seguro](#)
- [Monos al teclado II: Kolmogorov y la entropía del Universo](#)
- [La lógica de Twitter: ¿se puede razonar en poco espacio?](#)

Otros Blogueros



Beisario Iturra Peralta
(Noticias)



Claudio Usón
(Tecnología)



Juan Guillermo Tejeda
(Noticias)



Tomás Flores
Economista (Invertia)



Ximena Torres Cautivo
(Libros)

muchos procesadores disponibles. Por ejemplo, ¿cómo se pondría de acuerdo con alguien (o con muchos) para ordenar una lista de números? ¿Y para solamente encontrar la mediana de la lista (el elemento que caería al medio si ordenara el conjunto)? ¿Y para recorrer todos los nodos de una red moviéndose a través de conexiones entre ellos? Dele cualquiera de estos problemas sin paralelismo a cualquier alumno con un año de formación algorítmica y lo resolverá al instante. Pídale que lo resuelva con varios procesadores y seguramente lo verá bastante desconcertado.

Se acaba la era de la indolencia. Es hora de volver a pensar otra vez. Los que programan o desarrollan algoritmos tendrán que aprender a pensar en un modelo al que pocos están habituados. Y los que aplican la tecnología, a acostumbrarse al nuevo paradigma en el que lo que el pensamiento no da, esta vez la tecnología no lo prestará.

[permalink](#) [trackback](#)

3 Comentarios »

1. Totalmente cierto. Y los "computines" debemos hacernos cada vez más expertos en manejar y diseñar sistemas para hordas de computadores.

A todo esto, un pedazo de paper en el área:

<http://news.ncsu.edu/releases/wmssolinhthreads/>

Comentario by [Mauricio Monsalve](#) — Julio 6, 2010 5:20 pm

2. Conuerdo con el fondo del artículo, pero hay una imprecisión. Del artículo se desprende que la ley de Moore dejó de cumplirse, pero la verdad es que sigue vigente.

De acuerdo a la wikipedia lo que predijo Moore fue:

"The number of transistors that can be placed inexpensively on an integrated circuit has doubled approximately every two years."

http://en.wikipedia.org/wiki/Moore%27s_law

Esto se sigue cumpliendo hoy en día, aunque Moore dijo que se cumpliría al menos por los proximos 10 años, a partir del momento en que lo enunció.

Después de enunciar este principio, surgieron corolarios que *no* son atribuibles a Moore. Uno de ellos dice que la velocidad de cómputo secuencial se duplicaría cada 2 años. Esto claramente dejó de cumplirse, pero no es la ley original de Moore. Otro decía que la potencia requerida por un chip crecía en un 50% cada 2 años, lo cual significaba que por allá por el 2020 se necesitaría una central nuclear para un solo chip.

Intel dice que ellos seguirán respetando la ley de Moore por algunos años mas. Gracias a esto, será posible integrar cada vez más cores en un solo chip.

—

Pero reitero, estoy de acuerdo con el tema de fondo del artículo: hora de volver a pensar. 😊

Comentario by [Luis Mateu](#) — Julio 6, 2010 6:29 pm

3. ¡Wena profe Mateu! Pero en el pasado tiempo, la Ley de Moore era aplicable en un mundo en el que el consumo energético no era tan importante. Sin embargo, ¡ahora es una restricción en el diseño!

Otra cosa que uno puede resaltar es que esta ley también ha aplicado a otros artefactos, como consolas de juegos, PDAs, teléfonos celulares, televisores, routers, tarjetas gráficas, etc.

Comentario by [Mauricio Monsalve](#) — Julio 6, 2010 8:45 pm

[Feed RSS para comentarios en este post.](#) [TrackBack URL](#)

Dejar un comentario

Nombre (obligatorio)

Email (no será publicado) (obligatorio)

Sitio Web