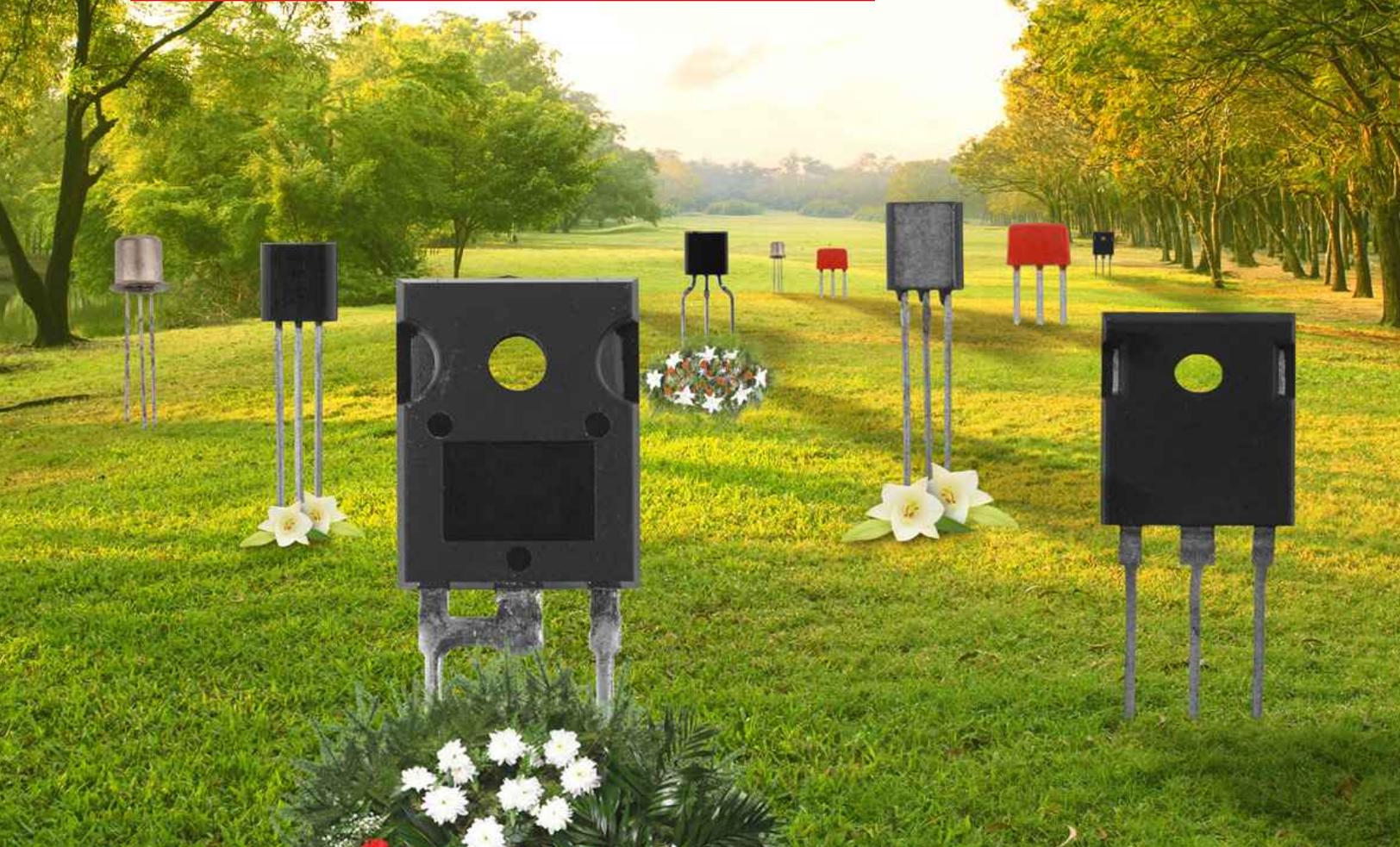


EL FIN DE MOORE Y EL SOFTWARE

Parece increíble lo que ha cambiado la tecnología
en los últimos veinte años...





JOSÉ MIGUEL PIQUER

Director de Servicios de Tecnologías de Información y Comunicaciones (STI), Universidad de Chile. Profesor Asociado Departamento de Ciencias de la Computación, Universidad de Chile. Director Técnico de NIC Chile y socio fundador de la consultora IT-Talk Ltda. y de E-voting Chile SpA, dedicada a servicios de votación electrónica. Entre 2011 y 2013, fue el encargado del desarrollo de proyectos científicos de transferencia en INRIA Chile, centro de excelencia internacional, donde se busca establecer un puente entre la investigación y el desarrollo productivo.
jpiquer@uchile.cl

En 1996 habían 22 servidores web en todo Chile, un supercomputador era apenas capaz de servir 100 videos simultáneos en una red local, nos conectábamos a Internet por teléfono a 28.8 Kbps y sacar 1 Kbyte/s era considerado óptimo. El servidor de medios del DCC tenía 100 Gbytes de almacenamiento y 2 procesadores de 100 MHz.

Hoy todos esos números son ridículos. A modo de ejemplo, bajar una película común hoy en día, de unos 2 Gbytes, habría tomado 2.000.000 de segundos, es decir casi un mes en 1996. Los chicos de hoy están acostumbrados a que eso demore unos minutos. Demás está decir que un celular en la época apenas lograba llamar por teléfono.

Por otro lado, todas estas mejoras se multiplican a veces entre ellas, o al menos trabajan juntas en lograr cambiar radicalmente lo que puedo hacer con ellas: si no tengo ancho de banda, procesador, almacenamiento y pantalla con la resolución suficiente, no puedo ver películas en alta definición. Esto hace que lo que hoy podemos lograr con la tecnología disponible no tiene ninguna relación con lo que soñábamos veinte años atrás. En ese tiempo llegué a escribir que yo no pensaba que fuera viable que Internet compitiera con una tienda de videos (¡afortunadamente no aclaré si lo pensaba para 1996 o para siempre!).

Desgraciadamente, no siento que hayamos logrado seguir ese ritmo en el lado nuestro: el software. En 1996 apareció Java, como una revolución en la forma de programar aplicaciones. Hoy aparecen y desaparecen lenguajes de programación por todos lados, pero no siento que logremos estar a la altura de lo que se requiere. Incluso buenos nuevos lenguajes como Scala, son básicamente un re-empaquetamiento de los lenguajes funcionales de los años sesenta. Me preocupa que sigamos programando como artistas y artesanos en un mundo en que tantas vidas dependen de que el software funcione.

Los sistemas operativos también se sienten un poco anticuados. En 1996 ya usábamos Linux como un sistema moderno. Es cierto que Linux hoy no

tiene mucho código que haya sobrevivido desde 1996, pero el modelo y el paradigma de base son los mismos.

Me da la impresión que el vertiginoso avance del hardware nos ha impedido mejorar el software a la velocidad adecuada. La ley de Moore nos hizo conformistas: De alguna forma era más barato y simple esperar a que Moore arreglara un programa ineficiente a intentarlo nosotros mismos. Incluso reiniciar el computador se hizo tan rápido que ni siquiera es demasiado molesto que se nos caiga el sistema. Por otro lado, la primacía de una, o a lo más dos arquitecturas, hace que no debamos preocuparnos demasiado por la portabilidad.

Pero esos tiempos han terminado, Moore ya no es lo que era.

El primer síntoma lo tuvimos cuando los procesadores comenzaron a multiplicarse en vez de aumentar su velocidad, y eso nos ha hecho revisar muchos programas que no eran paralelos y tener que reescribirlos. Pero ahora parece que será mucho peor. Llegando a la escala atómica, los procesadores se enfrentan a cambios tecnológicos mayores, y volveremos a un mundo complejo, con arquitecturas diferentes y paralelas, donde el software tendrá que adaptarse en forma mucho más difícil que antes. Es bastante probable que aparezcan arquitecturas más especializadas y no sea tan fácil escribir software para todas y portable entre todas.

Ojalá estos desafíos sirvan para que repensemos el software. ¿Cómo debíamos programar ahora en la nube? ¿Cómo manejamos múltiples computadores distribuidos por la red? ¿Cómo manejamos la movilidad con conectividad y contexto?

Me parece que es un muy buen momento para tomar lo que hemos aprendido en veinte años de programación para generar nuevos paradigmas, nuevas propuestas que pongan nuevamente al software a la par de la innovación y cambios radicales que nos ha aportado más bien el hardware todos estos años. ■