

## Een wet van Moore voor software?

*Jan Heering*

Jan Karel heeft zijn sporen verdiend in de (discrete) algoritmiëk. De constatering in het PITAC rapport [1] en elders dat betere algoritmen en betere hardware in gelijke mate hebben bijgedragen aan de fenomenale ontwikkeling van computermodellering was dan ook koren op zijn molen. Twee elkaar versterkende exponentiële trends. Een indrukwekkende onderstreping van het belang van algoritmiëk en voor Jan Karel een ideale kapstok om zijn veelvuldige pitches over het belang van computational science aan op te hangen.

Sinds de eerste chips rond 1960 ten tonele verschenen heeft de evolutie van hardware de wet van Moore gevolgd. Gezien het bovenstaande geldt er blijkbaar ook een soort wet van Moore voor algoritmiëk. Deze heeft geen naam en verdient nog minder een wet genoemd te worden dan die van Moore. De mogelijkheid (maar niet de praktische realisatie, laat staan het tempo van realisatie) van de wet van Moore kan herleid worden tot de schalingswetten van Dennard et al. voor MOSFETs, de op chips meest gebruikte transistors. Ik kan me vergissen maar het lijkt me kras dat een vergelijkbaar eenvoudig principe aan de basis ligt van een wet van Moore voor algoritmiëk. Daarvoor is het veld te groot en te divers. Het zijn veeleer algoritmische uitvindingen en verbeteringen over een breed front en in snel tempo die samen een op de wet van Moore lijkend exponentieel verloop opleveren. De exponentiële ontwikkeling van hardware heeft een navenante ontwikkeling van software mogelijk gemaakt. Zie het groeitempo van Windows en Linux [2]. Opnieuw een soort wet van Moore, maar nu voor het aantal regels code van softwaresystemen. Het eind is niet in zicht. Onder het motto *scale changes everything* waagt het ULS rapport [3] zich aan een poging het onafwendbaar naderbij komen van ultragrootschalige systemen van miljarden regels code onder ogen te zien. Moed en optimisme kunnen de auteurs niet ontzegd worden. Geestverruimende lectuur.

Het ziet er naar uit dat de wet van Moore (de echte) zijn langste tijd gehad heeft. Maar lithografie is een kat met zeven levens. Hoeveel winst er, voorbij de grenzen van de lithografie, op moleculaire, atomaire of zelfs nucleaire (waarom niet?) schaal uiteindelijk nog te behalen zal zijn, valt niet te zeggen. Het enthousiasme voor moleculaire elektronica van mijzelf en anderen in de tachtiger jaren was, achteraf gezien, prematuur [4]. Dertig jaar later is moleculaire elektronica nog steeds niet meer dan een kandidaat-opvolger waaraan mogelijk geen behoefte zal blijken te bestaan.

Hoe dan ook, welke invloed zal een vertraging in de evolutie van hardware hebben op de algoritmie? Het antwoord ligt voor de hand. Algoritmie zal verder aan belang winnen naarmate de vooruitgang op hardwaregebied langzamer gaat. De versterkte inzet op parallelle algoritmen om de huidige multicoreprocessors effectief te kunnen benutten is pas het begin. Uiteindelijk komt de algoritmie er alleen voor te staan. Dat zal wennen zijn voor een hardware-enthousiast als ik. Maar Jan Karel zal er geen enkele moeite mee hebben. En misschien loopt het allemaal heel anders.

[1] PITAC, *Computational Science: Ensuring America's Competitiveness*, 2005.

[http://www.nitrd.gov/pitac/reports/20050609\\_computational/computational.pdf](http://www.nitrd.gov/pitac/reports/20050609_computational/computational.pdf)

[2] James R. Larus, Spending Moore's dividend, Technical Report MSR-TR-2008-69, Microsoft Research, 2008. <http://research.microsoft.com/pubs/70581/tr-2008-69.pdf>

[3] Linda M. Northrop et al., *Ultra-Large-Scale Systems: The Software Challenge of the Future*, Carnegie Mellon Software Engineering Institute, 2006.

<http://www.sei.cmu.edu/uls/>

[4] Hyungsub Choi and Cyrus C. M. Mody, The long history of molecular electronics: Microelectronics origins of nanotechnology, *Social Studies of Science* **39** 1 (February 2009).

<http://www.owl.net.rice.edu/~cyrus.mody/MyPubs/Molecular%20Electronics.pdf>