

„Software“ ist keine „Weiche Hardware“ !

Reiner Hartenstein

Daß man „soft“ mit „weich“ übersetzt ist irreführend, denn „Software“ ist das katastrophale Gegenteil von weicher Hardware und kommt aus der Katastrophe nicht heraus. Die „Software-Krise“ [F. L. Bauer, 1968] wird

Critique of the von Neumann Model

Nathan's Law: Software is a gas. „even fills the internet“
It expands to fill all its containers ... and the clouds

The Next Fifty Years of Software: The Crisis Continues [ACM97]
 Nathan Myhrvold, Microsoft Ex-CTO

Wirth's Law
 "software is slowing faster than hardware is accelerating"

"The von Neumann Syndrome"
 "RAM" 2007; Ed Adams: "Why Software Still Stinks (It's STILL Insecure!!)" David Parnas

| year | system | SLOC (millions) |
|------|----------------|-----------------|
| 2001 | Windows XP | 40 |
| 2005 | MAC OS X 10.4 | 86 |
| 2007 | SAP Net Weaver | 238 |

incompetent programmers

Encyclopedia of things considered harmful: "All software sucks" Peter G. Neumann

Software Disaster Reports:
 N. N. 1995: THE STANDISH GROUP REPORT
 Robert N. Charette 2005: Why Software Fails; IEEE Spectrum
 Anthony Berglas 2008: Why it is Important that Software Projects Fail

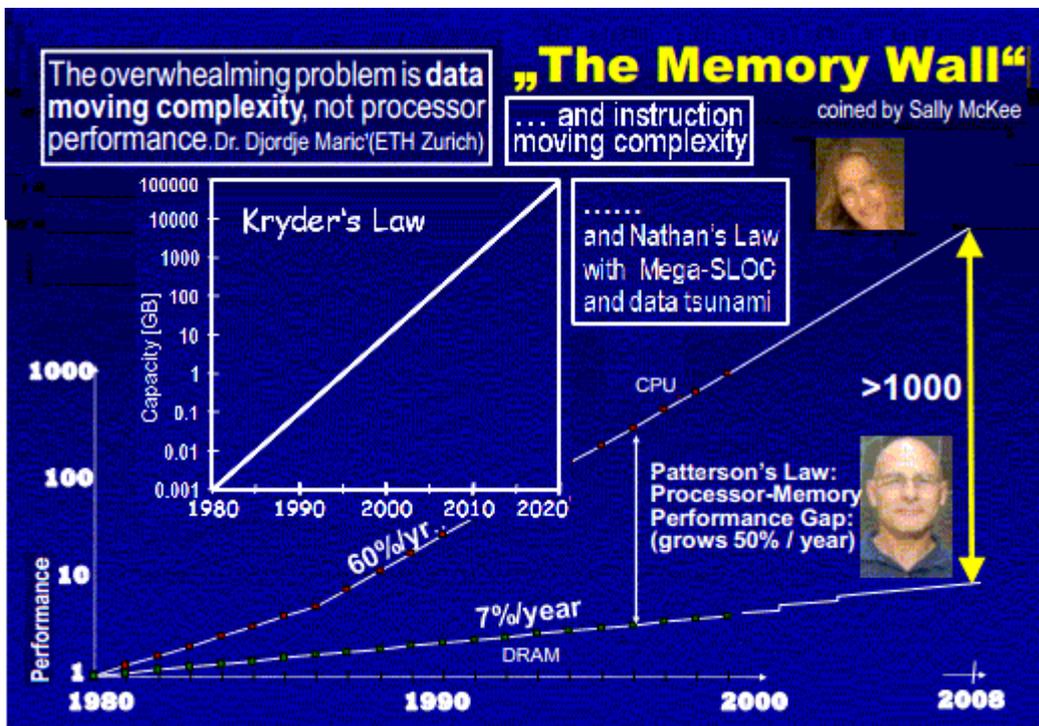
Critique of von Neumann is not new:
 E. Dijkstra 1968; J. Backus 1978; Arvind, 1983; R. Hartenstein, G. Koch 1975; L. Savain 2006;
 Peter G. Neumann 1985-2003; L. Savain 2006. The universal Bus considered harmful Why Software is bad

1985-2003: 216x "Inside Risks" 18 years in back cover of Comm. ACM

wegen „Nathan's Law“ stets immer schlimmer, u. a. auch wegen des von-Neumann-Syndroms. In dem nebenstehenden Bild wird von der großen Zahl von Kritikern nur ein kleiner Teil erwähnt.

Folgende Ursachen sind schuld an der über alle Maßen ungläublichen Ineffizienz von Software. Das Bild nennt eine Anzahl davon. Es ist die ungeheure Größe von Software-Paketen. Siehe Nathan's Law und die Tabelle der SLOC (Source Lines of Code). Dazu kommt die oft sehr beklagenswerte Qualität der Ergebnisse der Bemühungen der meisten Software Teams. Hinzu kommt die wegen der Größe sehr ineffizienten Speicher-Hierarchien. S. nächstes Bild.

Die Zugriffszeit zu Speicherchips außerhalb des Chips des Mikroprozessors (CPU) ist mehr als um den Faktor Tausend langsamer als zum Speicher auf dem Mikroprozessor-Chip selbst Dazu kommt noch eine weitere



organisatorische Ebene: die Festplatten, deren Zugriffszeit noch viel langsamer ist. Kryder's Law zeigt, was für riesige Mengen von Software und Daten sich auf dem langsameren Medium heute befinden können.

Seit etwa dem Jahr 2005 liefern Firmen wie intel nur noch Chips mit mehreren langsameren Prozessoren wobei nicht nur Programme parallel laufen müssen, sondern auch die Speicher. Dabei können einzelne Daten- oder Befehls-Ströme sich gegenseitig behindern. Der Zwang zu Parallelität mindert nicht nur die Produktivität der Softwareleute, sondern erschwert auch Anforderungen an Programmierer, Betriebssystem und Compiler.

Man sieht, daß das Bündel der dramatischen Behinderungs-Ursachen sich zu einer unglaublich massiven Ineffizienz hochmultipliziert. Dies ist typisch für Systeme, die nach den von-Neumann-Prinzipien organisiert sind. Im Gegensatz hierzu stehen Systeme, die vollständig auf der sogenannten „weichen Hardware“ basieren. Deren um Größenordnungen bessere Effizienz (siehe unter „Speed-up-Faktoren“) beruht u. a. darauf, daß diese zur Laufzeit keine Befehlsströme benötigen, auch nicht für hochparallele Anwendungen.