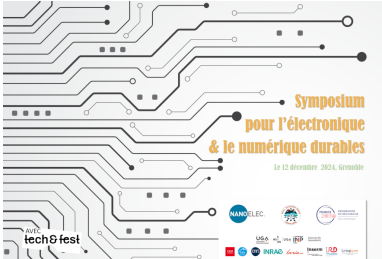
Analyse de cycle de vie adapté au logiciel : outillages et recommandations

Mettre des chiffres sur la loi de Wirth

Henri-Pierre CHARLES

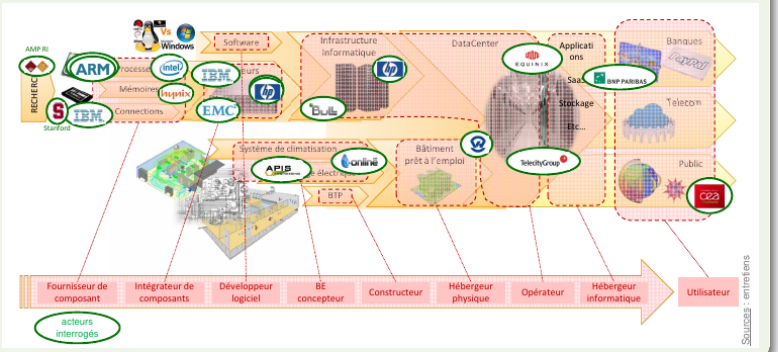
CEA HDR Fellow
Univ. Grenoble Alpes, CEA, List, F-38000 Grenoble, France

jeudi 12 décembre 2024
16:55 - 17:10



Introduction : Data Center Value Chain

Data Center Value Chain



Missing Value : SW Stack

- Server : LAMP (Linux / Apache / PHP MySQL)
- Client : browser, Javascript engine / HTML / CSS
- Value : user data

Software has no “value” but is responsible of hardware obsolescence

Introduction : Niklaus Wirth (15 February 1934, 1 January 2024)

Niklaus Wirth in 2005. Niklaus Wirth

Wirth Projects

- Pascal 1968-1972 Pascal2 / P-Code - UCSD - TurboPascal
- Modula2 1973-76
- Oberon 1977-1980
- Lilith 1977-1981

Books / Articles

- “The Pascal User Manual and Report”
- Algorithms + Data Structures = Programs
- Wirth’s law (1995) **“Software is getting slower more rapidly than hardware is becoming faster.”**
  Article "A Plea for Lean Software"

Introduction : Wirth's Law

Wirth's law

A Plea for Lean Software

Niklaus Wirth
ETH Zürich

Memory requirements of today's workstations typically jump substantially—from several to many megabytes—whenever there's a new software release. When demand surpasses capacity, it's time to buy add-on memory. When the system has no more extensibility, it's time to buy a new, more powerful workstation. Do increased performance and functionality keep pace with the increased demand for resources? Mostly the answer is no.

- February 1995 DOI : 10.1109/2.348001
- "Software is getting slower more rapidly than hardware is becoming faster"
- Is the the Moore Law's to the minkind benefit ?

Illustrations

- Windows boot speed
- Android "killer" application
- Visual Studio Code
- Android Studio

Reasons : Business model

- Deliver fast, not reliable
- Tricking functionnality : not only provide a functionnality, push advertisment, get data from user,
- Add functionnality, do not choose. The text editor nightmare : where is the menu option to activate a given functionnalyty ?

Introduction : Amhdal Law's ?

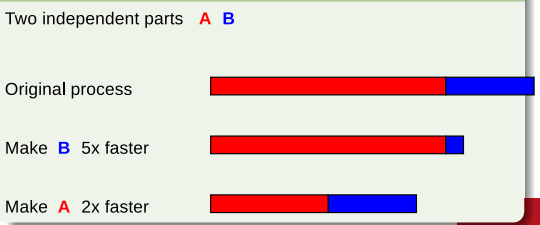
Ahmdal law's: "Speedup is limited by the sequential part"

- Acceleration limited by the "X" part : $S = \frac{1}{1-p}$
- X $\in$ {Parallel, Vectorized, using IP bloc, ...}

Programmer approach

- "This part is parallel" let's optimize !
- It's better to fight for a small x2 than for a big x5 !

What to optimize



Old Hardware Examples : SharpPC1500

Sharp PC-1500

- Release : 1981
- Lifespan : 4 years
- Software update : 0
- Programming language : basic

No user evolution : no obsolescence

Illustration

BloatWareExamples : Android Applications

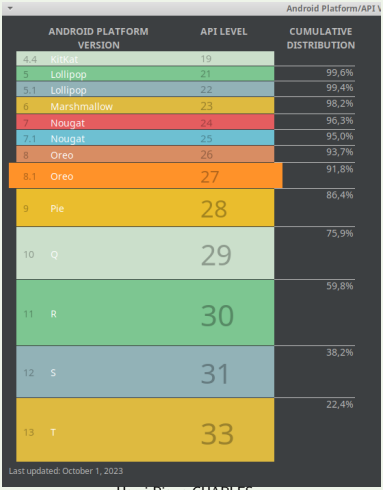
Android Application Funding

- Mainly with advertising
- Application implication
  - % of screen surface
  - % of user time
- In App purchases : e.g. android games
- Paid plugins : e.g. web software platforms (wordpress)

Value chain

Software Value is the “Invisible elephant” in the value chain

Programmer dilemna



Software Updates : The Wave

Why update ? Why refuse ?

- Operating system : for “security” reasons. Who can refuse a “new old” unsecure platform ?
- Applications : for new functionnality. Who can refuse a “better” software ?
- Processor microcode : for security reasons. Who can refuse a “new old” unsecure platform ?

What is a software ?

- An instance ?
  - LibreOffice release 24.8.3,
  - firefox 133.0
- An evolving object with features ?
  - LibreOffice (since 2011),
  - Mozilla Firefox (since 2002),
  - GNU Emacs (1984)

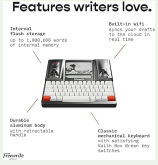
Is it possible to resist innovation ?

La_Grande_Vague_de_Kanagawa

Software Updates : Exceptions

Hardware Markets

- There is a (small) market for “distraction free” hardware :
  - e-readers : no colors, no animated images
  - Type writer :



  - Distraction free cell phones : Nokia 225, The Light Phone, . . .

Users Exceptions

- George R. R. Martin, “Game of Thrones” scenarist use wordstar text editor (from 1978):

  "I hate some of these modern systems where you type a lower case letter and it becomes a capital. I don’t want a capital. If I’d wanted a capital, I’d have typed a capital. I know how to work the shift key. Stop fixing it."

  `https://www.bbc.com/news/technology-27407502`
- All scientific writer using LaTeX

Project : Initial Guess

"Give me the exact number" (Brazil)

Is the software bloat a realty ?

- Who to mesure this realty ?
- How to compare to the hardware evolution ?
- Are the two evolution parallel ?
- Could we use recent software on 10 years old hardware ?
- Could we give advice to programmers doing a new software release ?

Planned experimentation scenarios

- Software candidate :
  - Text Editor,
  - Web Browser,
  - Web Server
- Hardware candidate :
  - User Laptop with a long life
  - Gather hardware statistics

Project : Methodology

Study software evolution

"Software are forever"

- Software life span is larger than hardware
- Focus on end user software

Study hardware evolution

Hardware

- Emulate hardware by software and gather
  - Instruction count
  - Memory accesses
  - Disk accesses
  - Network access

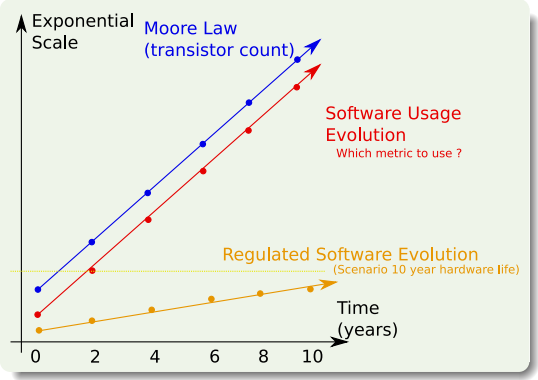
Study HW / SW interaction

- Automatize usage scenarios
  - Text editor
  - Web browser client
  - Web browser server side

How

- Count how many instructions are needed to achieve a given task
  - Write a simple document
  - Browse (render) a web page (client side)
  - Generate a web page (server side)

Project : Expected Results (Conclusion)



Targets

- Prove that Wirth's Law is true
- Give metrics (M insns / task ?)
- Design programming tools
- Be part of a normalization regulation project
- Provide tools to achieve those results

Output : Regulation

- Master software complexity evolution with new metrics (Giga instruction cycles ? Joules ?)
- Provide programming tools
- Force (Gently suggest) software editor to keep release evolution under a complexity threshold evolution.
  (Similarity with web accessibility ?)
  - LibreOffice future release should work under a X instruction count for a given scenario