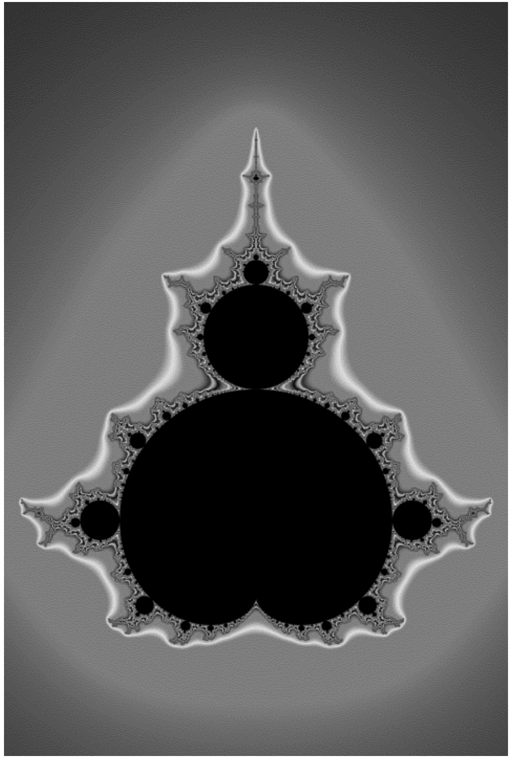
Welcome at CERN

About me

My first computer

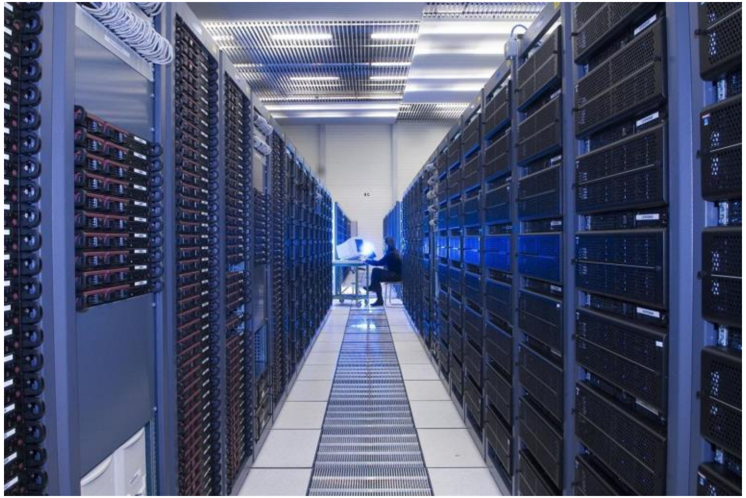
My first printer

One of my first computer programs



Hardware

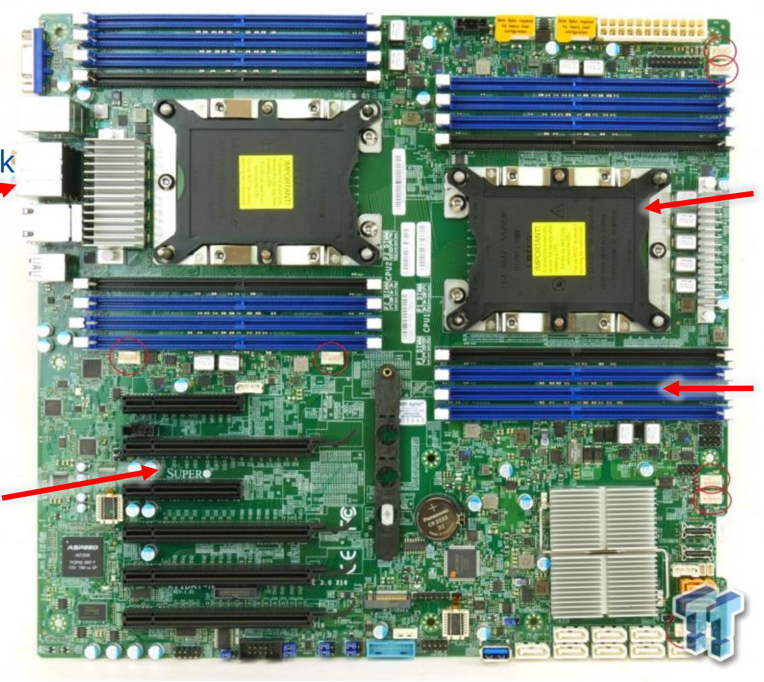
How does a computer look like?



Servers (and desktop PCs)

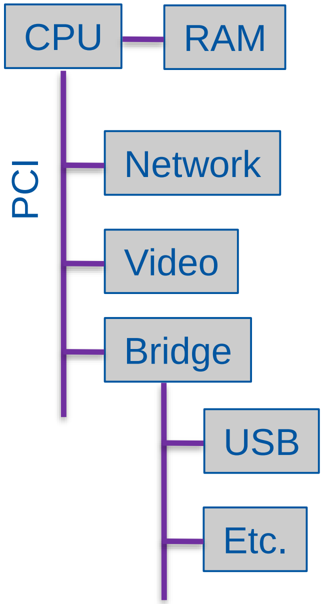
Network port

PCI slots



CPU socket for multi-core CPU

Memory bank



How much computing power has CERN got?

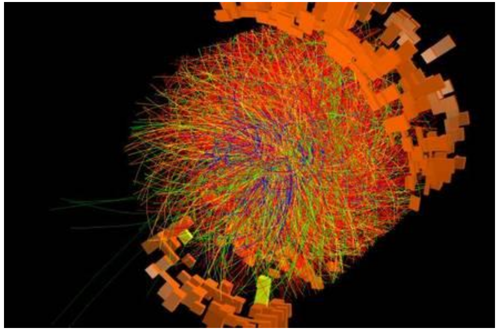
Location	Number cores
CERN computer centre	300,000
LHC experiments	4 * ~50,000
Rest of the world (LHC grid)	1,000,000

Core:
An independent processing unit that can execute a program

Should you be impressed?

- Not too much
- The large internet companies (Google, Amazon, Facebook, etc.) invest ~5-10 billions $ per year in computing H/W

Computers can also look like that….



Computing units

Storage:

Bit	= 1 or 0	= "yes" or "no"	
Byte	= 8 bits	= one character	
Kilobyte (kB)	= 1024 Bytes		= one page of text
Megabyte (MB)		= 1024 kB	= one digital image
Gigabyte (GB)		= 1024 MB	= one movie
Terabyte (TB)		= 1024 GB	= storage worth EUR 50
Petabyte (PB)		= 1024 TB	= LHC data production per week
Exabyte (EB)		= 1024 PB	= still not a number that would impress Google

Processing:

1 Hz	= one instruction per second		= my ability to add single digit n
1 KHz	= 1000 Hz		
1 MHz	= 1000 kHz	= clock frequency of the Apollo 11 computer	
1 GHz	= 1000 MHz	= clock frequency of a modern CPU	

Data archiving

Hard disk

This is what the cloud is made of

- Maximum capacity of one disk (10/2018): 12 TB
  - 3000 HD movies
  - 0.04 seconds of LHC raw data
  - 25 minutes of LHC accepted data
- CERN data archive: 132.000 HDDs

Magnetic tape

Tapes used for long term storage (backup) because:

- They are cheap
- Need no power
- Have lower risk of loosing data
- CERN data archive: 29.000 tapes

Excursion – fundamental science and technological applications

How many Nobel prizes are hidden in a hard disk?

1956: Transistor (Shockley, Bardeen, Brattain) 
But: No transistor without quantum physics
Therefore we also have to credit Bohr, Dirac, Pauli, etc. 

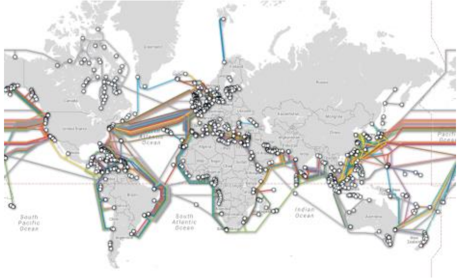
2007: Giant Magnetoresistance (Fert, Grünberg) 
Allows to increase the storage density

1964: Laser (Townes, Bassow, Prochorow) 
Used for the manufacturing process and for (future) laser assisted HDDs

…any maybe more that I am not aware of.

Networks

From  to 



Networks keep the world together

- Long distance:
  - glass fibre
  - Data rates up to 25 Gbit/s
- Short distance:
  - Copper cables
  - WiFi

Internet:

- A world wide network of computers
- Based on protocols (e.g. TCP/IP)
- Provides many services (e.g. e-mail)
- Very reliable

World Wide Web:

- An Internet service: html, http, URL

The future of computer hardware

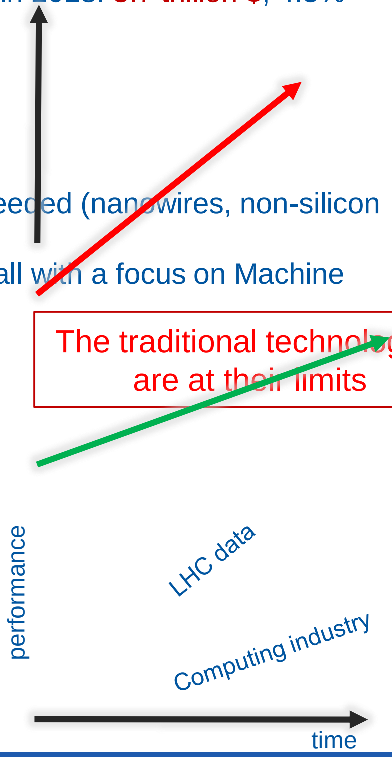
Total world wide IT spending expected in 2018: 3.7 trillion $, 4.5% growth rate

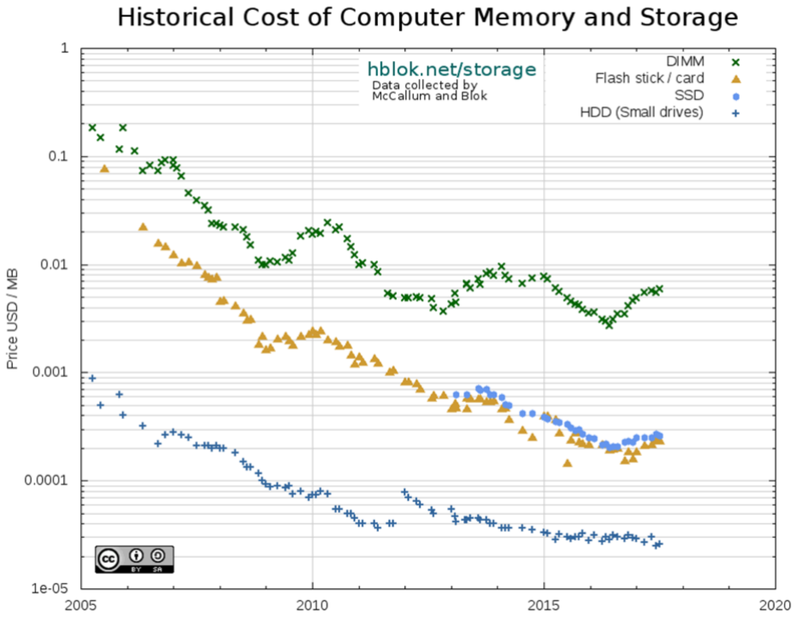
CPUs:
- Multi core CPUs
- Moore's law no longer holds
- Below 7 nm new technologies are needed (nanowires, non-silicon materials), very Expensive
- Plethora of new processor designs, all with a focus on Machine Learning

Memory:
- DRAM scaling slowed down
- Prices increasing

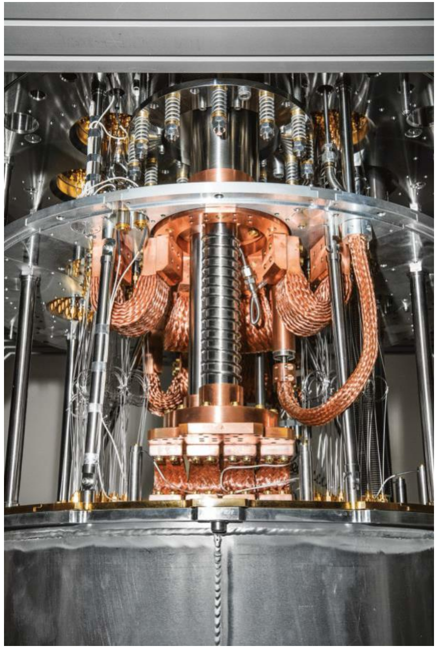
Storage:
- HDDs close to physical limits
- SSD prizes not dropping anymore

The traditional technologies are at their limits





Quantum computers?

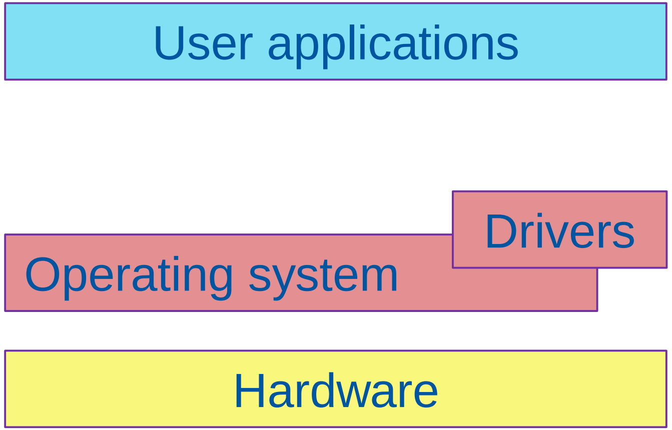


- Considerable progress during the last 2 years; number of qubits rising sharply
    - Intel 49-qubit, IBM 50, Google 72 for a quantum gate computer
- Various implementations from ion traps to silicon, focus is on silicon to re-use the fabrication process of standard chips
- Coherence time is still well below 1 ms, limits the time for quantum calculations
- Key problem is the error handling: mitigate by combining qubits
- Programming model is completely new; not clear how many algorithms can be 'converted' for a quantum computer; very, very high cost
- Prognosis: Irrelevant for HL-LHC

Despite some niche applications (e.g. cryptography) still a long way to go with many unknowns

But: CERN held a QC workshop in November 2018

From Hardware to Software



This is where 95% of the S/W development effort takes place

Extend the capabilities of the OS by custom functions

Can do (the dangerous) operations on the H/W

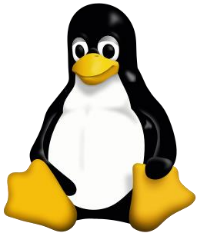
At CERN we even build our own H/W for special applications

The operating system

The basic question:

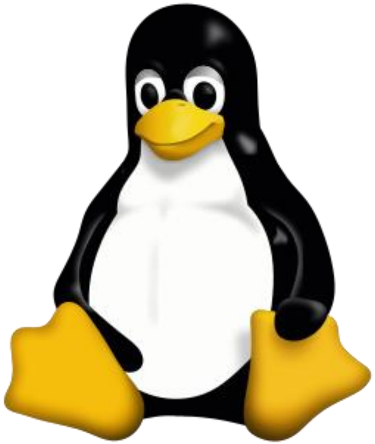
- Many users are familiar with it
- Some software only available on this platform
- Good support
- Expensive
- Proprietary (complicates driver development)

or



- Widely used in science
- Very transparent and free (you can get the source code)
- Powerful
- Not that well documented (at the kernel level)

And the „winner“ is:



Computing languages

Our (main) requirements:

- Fast! Fast! Fast! (not always)
- Procedural and object oriented
- Support for special H/W
  - e.g. GPUs, lab instruments
- Easy to learn and to debug
- Availability of 3rd party code
- Good support
- Widely used

And this language is called:

C/C++

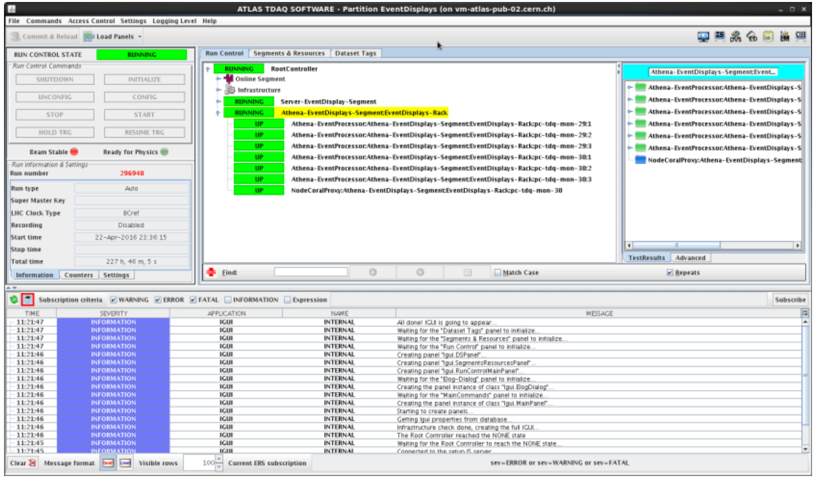
and Java

and Python (good for beginners)

and LabView

The big projects

Example 1: Data acquisition



GUI of the ATLAS DAQ system

A LHC DAQ system has to:

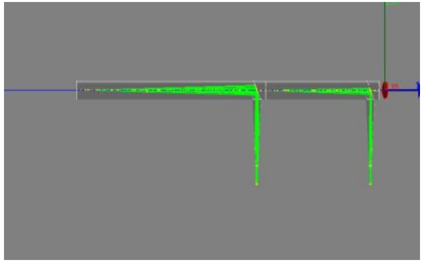
- Synchronize ~50.000 computer programs
- Handle ~80 TB/s of data
- Be scalable, fault tolerant, easy to use
- Be modular and configurable
- Be adapted to the experiment
- Be well supported
- Be fast and reliable

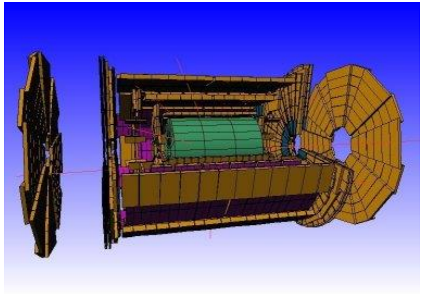
Each large experiment has developed its own solution

- 4-6 million lines of code
- Main ingredients
  - Many databases
  - State machines
  - Control trees
  - Expert systems
  - Device drivers, libraries and applications
  - Sweat and tears

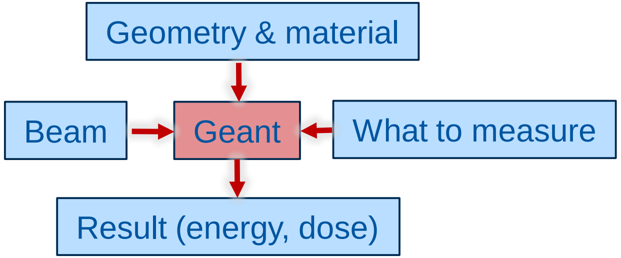
Example 2: GEANT4

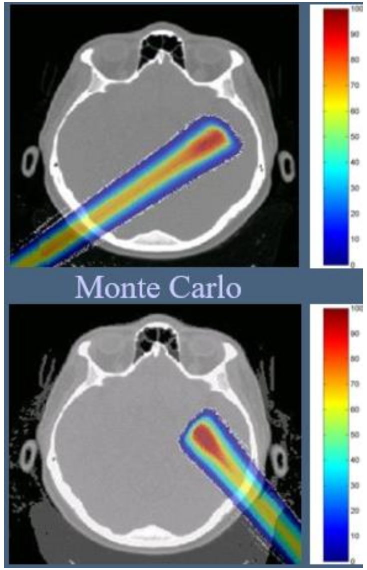
Geant4 is a platform for the simulation of the passage of particles through matter



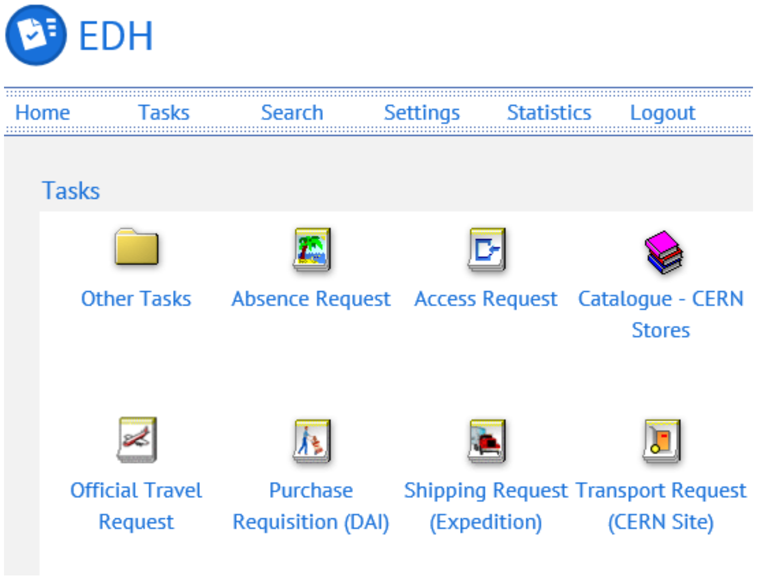


- Core Implemented in C++, 1 million lines of code
- Heavily object oriented
- Many small teams working together CERN, Japan, USA,F,I,E,UK
- Open source licence
- Works under Linux, Windows, MAC OS
- Very sensitive to performance – e.g. LHC experiment simulation uses 50% of LCG capacity





Example 3: Going on vacation

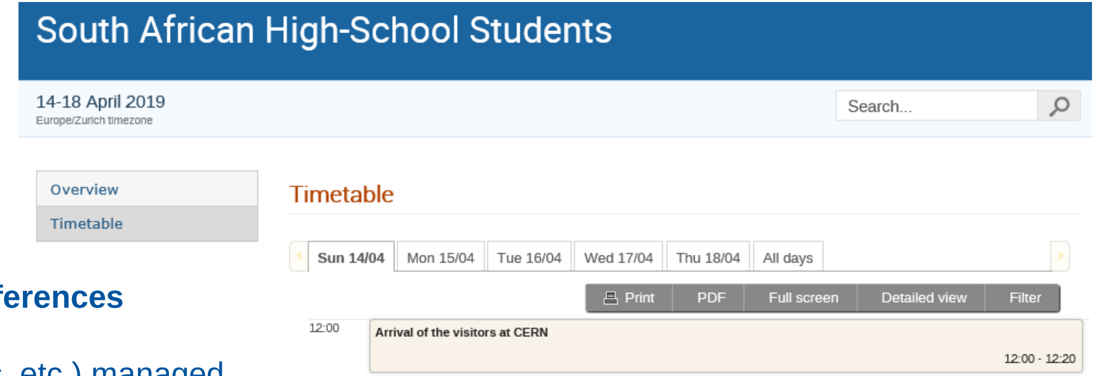


Electronic Document Handling

- Used for almost all business procedures (more than 70)
  - Orders from companies
  - Transport requests
  - Travel and vacation
  - Training requests
- 40.000 documents per month

- Java (beta user already in1996!), now Java 8, Java servlets
- JavaScript
- Web (HTML, Java applets since 1999)
- Linux Servers (virtual machines)
- Oracle WebLogic & Oracle databases -> migrating to FOSS
- Continuous integration, Jira, GIT
- IDE: JetBrains intelliJ IDEA (commercial), Eclipse
- 2.4 million lines of code

Example 4: Indico



We use Indico to organize our meetings and conferences

- More than 600.000 events (meetings, conferences. etc.) managed
- Open source stack (no commercial S/W at all)
- Publicly released under the GPLv3 (open source software license) and used by 200+ institutes and companies. E.g. UN

- Python and Java-script. No “C”.
- 150.000 lines of code (and Plug-ins for user specific functions)
- Development process: GIThub, CERN experts and external people participating
- CERN team: 4-6 FTEs

Example 5: MAlt

- Many companies heavily increase the prices for S/W licensed to CERN
  - Prices can increase by a more thane a factor of 10
- Many tools are only “free” for personal use (E.g. Dropbox)
  - Companies will be charged
- We are looking for alternatives
  - Avoid to be locked
  - Most alternatives have a license cost as well
- Often a less powerful (and cheaper) tool is acceptable
- Real FOSS S/W is rare
- “Open core”
  - Basic features are free
  - Advanced features cost money

Your take home message:
“avoid marketing traps”

CERN activities and services have been relying on commercial software, often leveraged by interesting financial conditions based on recognizing CERN’s non-profit status. Once installed, well spread and heavily used, the attractive pricing policy tends to disappear and is replaced by real business models that are considered to be unaffordable on the long term.

The CERN nature of openness targets delivering the same service to every type of CERN user, from Staff people to Users of the scientific resources. As a result the traditional business models on a per user basis drive the costs to be too high.

More importantly, the trendy Cloud approach enterprises are now focusing on, are introducing a new risk of lock-in on the data this time. It is always easy to import data in a cloud system, but always difficult to get data out.

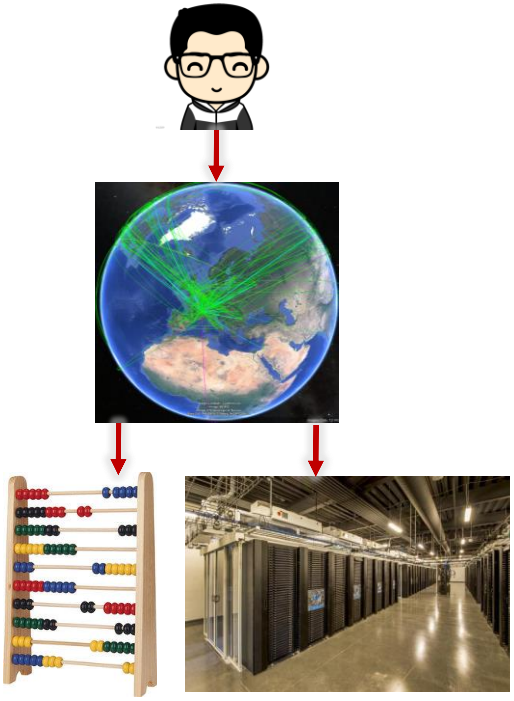
The MAlt project objective is to target if possible Open Source products, to deliver services inclusive to all CERN community.

Example 6: The LHC computing GRID

The **WLCG** is a global collaboration involving:

- 170 computing centres in 42 countries
- 2 million tasks run every day

- A federation of independent domains (grid of grids)
    - Not a centralized cloud
- Middleware: C, Java, Python Perl, etc.
- Millions of LOC
- S/W development at CERN with contributions from but also DESY, INFN, USA
    - no industrial companies involved
- All Grid computers run Linux (CC7)
- Open license Apache2.0

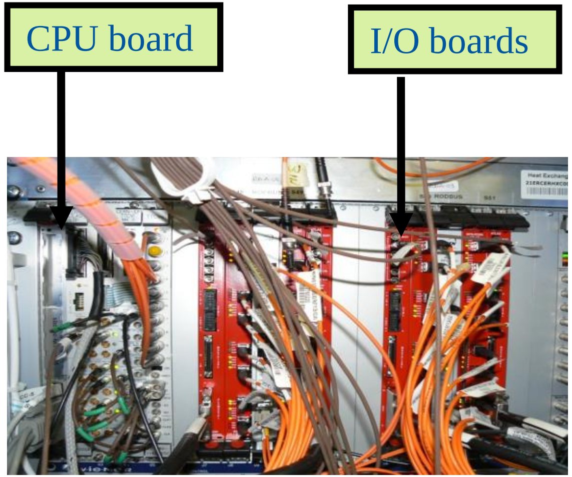


http://wlcg-public.web.cern.ch/

Join the grid: https://boinc.berkeley.edu/, https://lhcathome.cern.ch/lhcathome/

The small(er) projects

Example 1: My work (VMEbus driver)



Task given to me in 2002:
Enable the CPU board to exchange data with the I/O boards

Solution:
Develop a VMEbus device driver and library for Linux

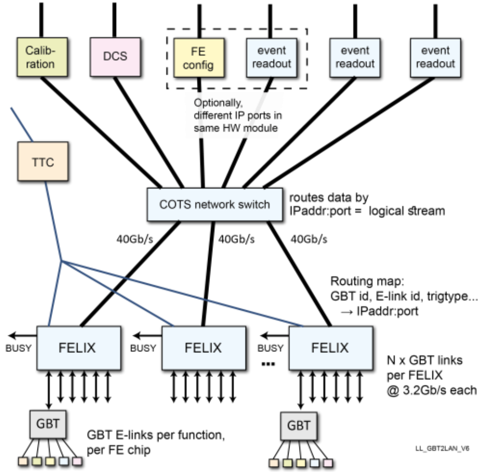
Problem:
I was not told anything about VMEbus and drivers at university. Luckily I had worked in this domain at CERN since 1993

Result (after ~1 year of work):
- Driver: 3500 lines of code (plain C)
- Library and tools: 15.000 lines of code
- Documentation: 100 pages

} ~90 lines per day

Status today:
- The driver is still used by ATLAS (~150 computers) and other experiments
- In 2018 I have found a bug that was not noticed for 16 years

Example 2: ATLAS FELIX



On of my colleagues develops S/W for moving data around in ATLAS

- His S/W (more that 10.000 LOC) receives data via a custom designed PCI card
- Data is sent to the next stage via network connections
- Performance is crucial

Therefore:

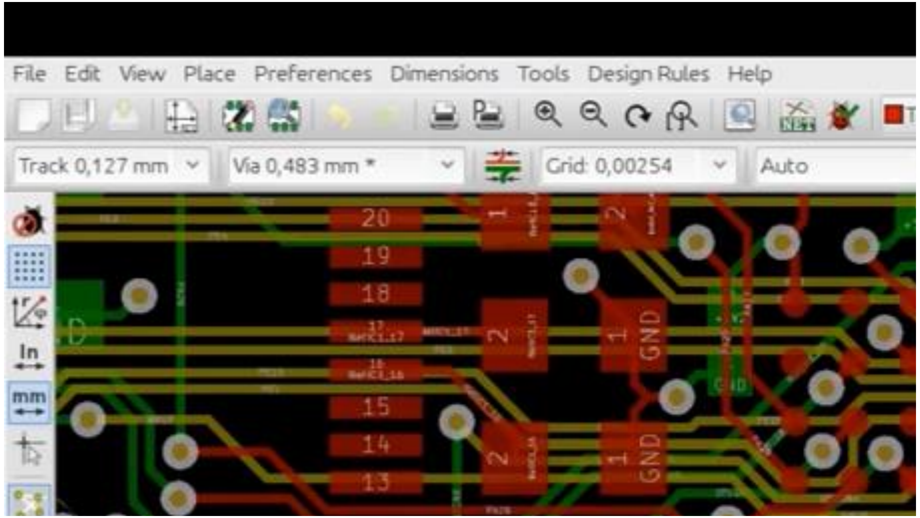
- He has to understand both the PCI card and the network in depth
- He has to understand the architecture and cache design of the computer
- He can influence the design of the PCI card and its firmware

As a consequence:

- He has to have multidisciplinary skills

CERN

Example 3: KiCad



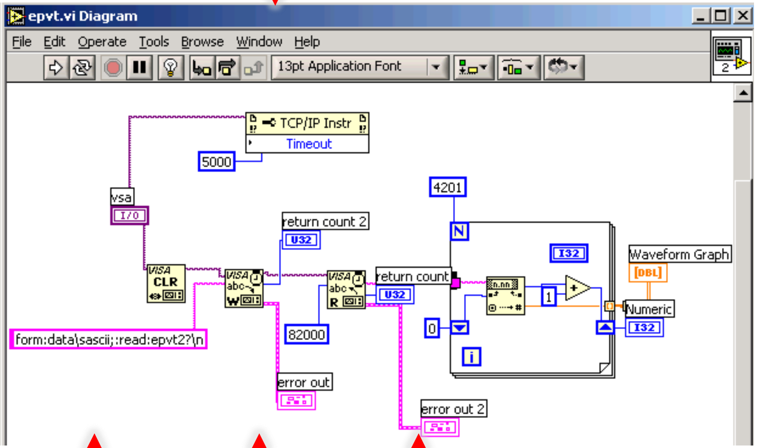
https://www.youtube.com/watch?v=CCG4daPvuVI

- KiCad is an open source software suite for Electronic Design Automation (EDA). It handles Schematic Capture and PCB Layout
- It runs on Windows, Linux and MacOS and is licensed under GNU GPL v3
- Initiated at Grenoble Institute of Technology
- A hand full of specialists at CERN have made important contributions
- Behind the scenes:
    - C++ (800.000 Lines of code)
    - Doxygen, GIT
    - A lot of mathematics (e.g. routing)
- Main challenge: A good user interface

- You can join the community

Example 4: Testing an electronics module

Card under test



Oscilloscope

Power supply

Signal generator

- Not all computer languages are text based
- LabView is a graphical programming language
- Boxes represent functions
- Lines represent the flow of data
- It can be very intuitive and productive to use this language
- LabView is very versatile and powerful
- ….but also slow, expensive and proprietary

Computer security

- CERN has a team of 7 safety experts
  - But everybody at CERN is responsible for computer safety
- Code development
  - Just some “Glue” for the Security Operations Center (SOC)
- Tools used by CERN
  - Stateful & stateless firewalling at Internet gates and between network segments
  - Antivirus: standard Microsoft Defender plus Malwarebytes “MBAM”
  - Security Operations Center (SOC): “Bro” intrusion detection, “MISP” for intelligence management, ELG, “the Hive”, “FIR”, “Cortex”, “JoeSecurity”
  - Microsoft SPAM filter plus “Fireeye EX” malicious mail filtering and attachment detonation
  - “nmap”, “OpenVAS” and “Skipfish” vulnerability scanners
- Interactions with other institutes and industry
  - Network with academic and HEP community through the world (Europe, US, Asia)
  - Law enforcement (police) & governments
  - Security vendors & analysist
- CERN is permanently under attack (directly and indirectly)
  - We seem to be ahead of the intruders

S/W life cycle

- Understand the problem and the time scales
  - Data streams, scalability, speed, etc.
  - Write a requirements document
  - Use cases vs abuse cases: how to misuse the program
- Develop a Prototype and test it
  - Get a feeling for the performance on the available H/W
  - Restart from scratch if you are not satisfied
- Implement the code and test it
  - You may want to use an IDE (e.g. Eclipse)
- Develop I/O emulators and diagnostic tools and test them
  - This may include H/W development
- Debug your code
  - You may need special tools (e.g. Valgrind, analysers, large scale systems)
- Specify and procure H/W
  - Tenders, budget constraints
- Write documentation
  - For yourself and for others
- Provide support
  - Your S/W may be used for 20 years

Note: This is not a linear process but highly iterative

The people behind the scenes…

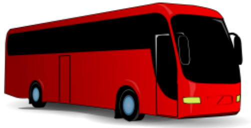
Javier Stefan Julia Pedro Jörn

Jan

- ✓ Computer scientists, engineers and physicists
- ✓ Got in contact with computers at an early age
- ✓ Acquired their basic skills at home as hobby programmers
- ✓ Have a passion for their work

John Emmanuel Markus Maarten

Do I have to learn to program?

What is the difference between  and  ?

- Programming is an essential skill for more scientists and engineers than you may think
    - Career perspectives
- The basic skills help you to better understand the world we are living in
    - “Computer literacy”

How to become a programmer?

Alternative 1: Just do it!

- Get a compiler, an editor and a bit of free S/W as well as a second hand computer
- Buy a book (e.g. “Automate the boring stuff with python”), read, program
  - You can also find good documentation on-line
- Risk: You may get used to some bad programming practices

Alternative 2: Just do it with friends!

- Same as alternative 1 but become a member of a project group (the internet is full of open source projects)
- Look at what other developers have done, understand their code, improve it
- Risk: you may get totally absorbed in the group. Make sure you don’t neglect your other duties

Alternative 3: Just study it!

- Enrol at a university and study computer science
- ….but keep programming in your free time

In case you got inspired…..

Ideas for your first (or second) step as a code developer

If you want to start with a very simple language:

- Code.org

If you want to touch or even build some H/W:

- https://www.arduino.cc
- https://www.raspberrypi.org