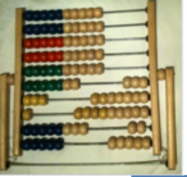
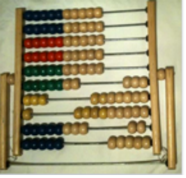


Future (?) of HEP computing

ACAT2019 Saas-Fee

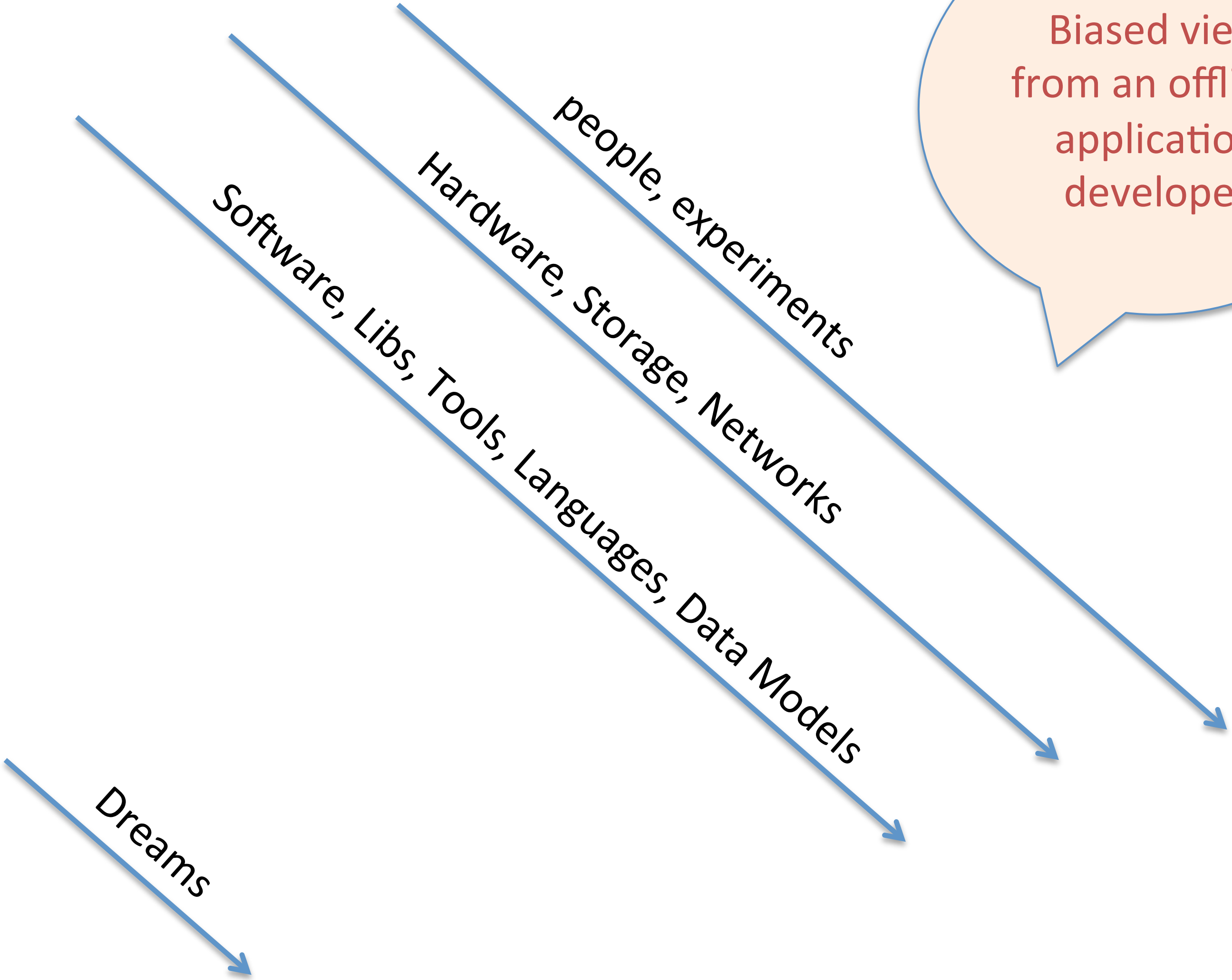
13 March 2019

René Brun /CERN



Plan of talk

- -1974:
- -1979:
- -1984:
- -1989:
- -1994:
- -1999:
- -2004:
- -2009:
- -2014:
- -2019:
- -2024:



people, experiments

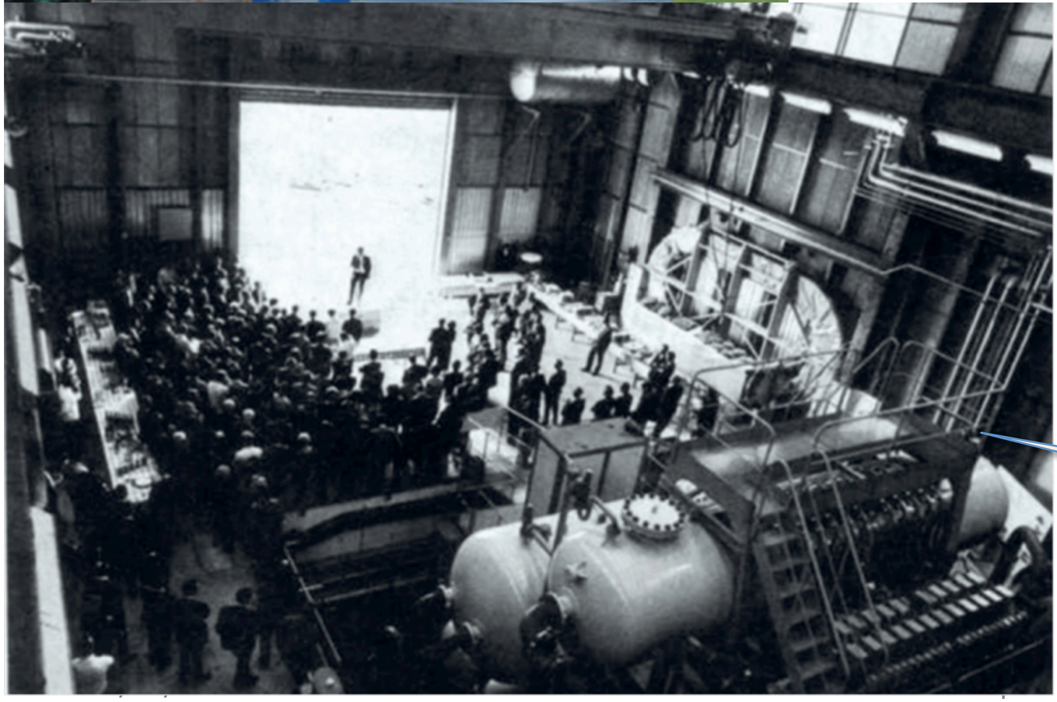
Hardware, Storage, Networks

Software, Libs, Tools, Languages, Data Models

Dreams

Biased view from an offline/ application developer

1974: Bubble chambers

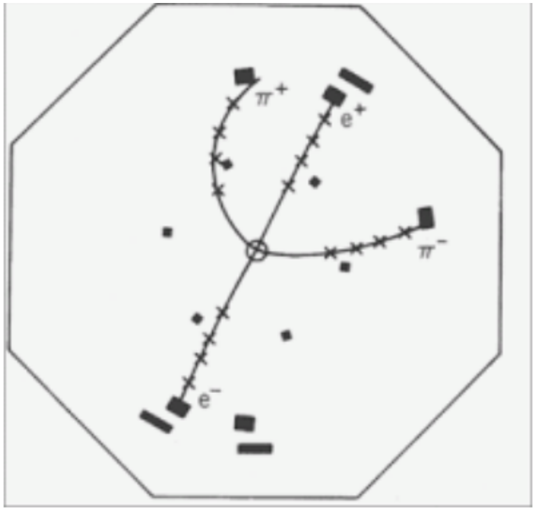


Neutral currents

50 to 100 people
Life in HEP

1974: cont

- SLAC/BNL : J/Psi
- ISR, 1st large collider, R602,R603, R704,etc
- PS: many small exp, Omega

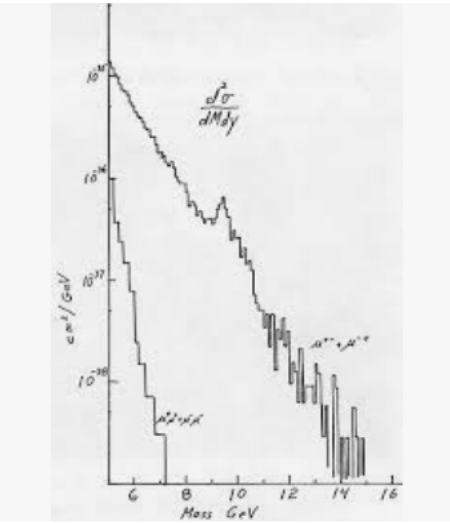


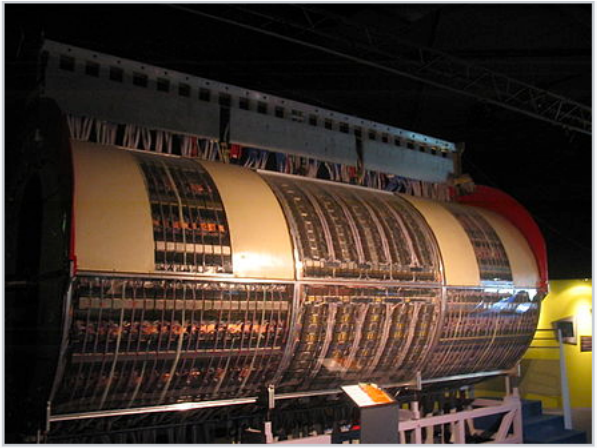
Most physicists build
and operate detectors
100% HEP career

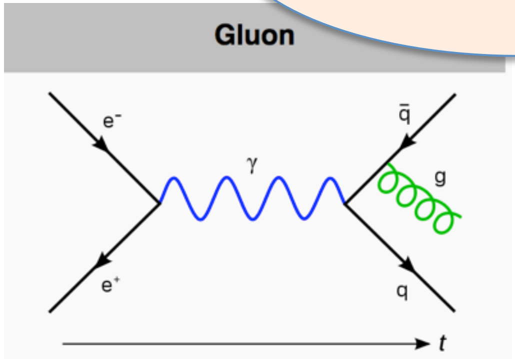
1979

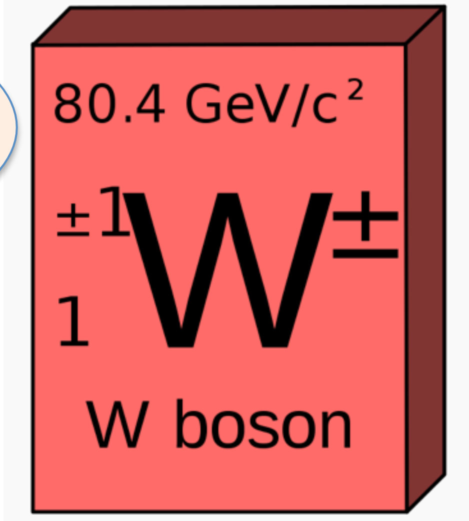
- FNAL : Upsilon
- SPS : W,Z
- DESY/Petra: gluons

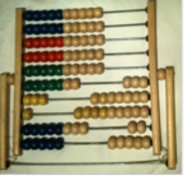
50<people<100
100% HEP career











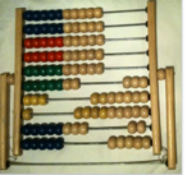
1984, 1989, 1994

100<people<200
100% HEP career

- 1984: LEP preparation
- 1989: LEP start, SSC: GEM, SDC
- 1994: SSC down,LHC up,Babar up,FNAL: top quark

WEB is born

300<people<1000
80% HEP career



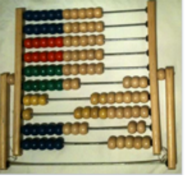
1999-2014

200<people<1000
70% HEP career

- 1999: LEP2,FNAL run 2,HERA,Neutrino oscillations exp, LHC design, construction
- 2004,FNAL,BNL,LEP, HERA,LHC
- 2009: LHC start
- 2014: After Higgs, precision measurements

1000<people<3000
50% HEP career

1000<people<3000
30% HEP career



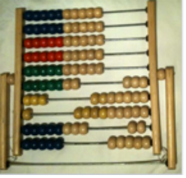
2019

- LHC shutdown
- Neutrinos, astro
- ILC, FCC, CHEP, CLIC???

Moving Out of Academia To ..

ure Esteveny , Head of CERN Alumni Relations, IR

1000<people<3000
10% HEP career



~3'600 network members

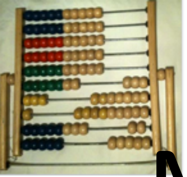
83% alumni or
alumni within a year

71% MPAs

42% USERS

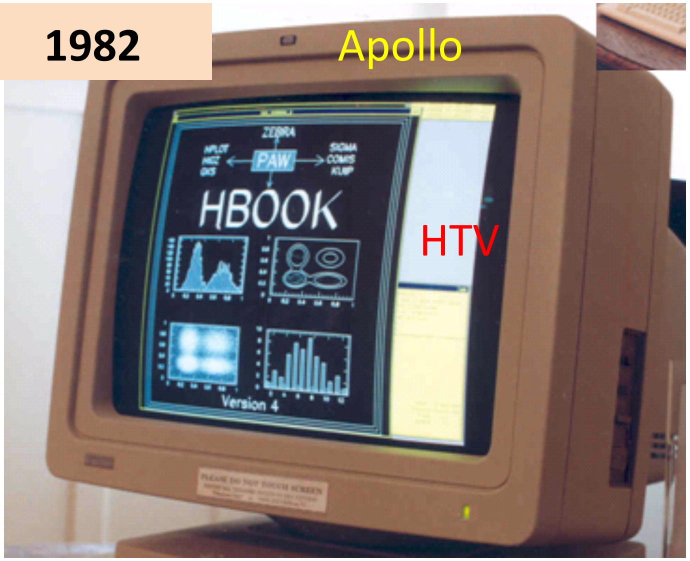
Technology
Environment
Manufacturing
Tanks
Research
Policy
Practice
Media
Design
Defense
Containers
Medical
Financial
Gambling
Apparel
Nanotechnology
Administration
Fashion
Internet
Fund
Information
Marketing
Advertising
Network
Aerospace
Space
Higher
Public
Energy
Affairs
Railroad
Raising
Travel
Packaging
Government
Cosmetics
Education
Industrial
Jewelry
Casinos
Law
Think
Legal
Resources
Computer
Manufacture
Semiconductors
Machinery
Networking
Security
Oil
Chemicals
Banking
Telecommunications
Consumer
Consulting
Luxury
Biotechnology
Utilities
Services
Accounting
Spirits
Mechanical
Software
Electronic
Games
Health
Electrical
Broadcast
Tourism
Nonprofit
Human
Sports
Management
Secondary
Aviation
Retail
Organization
Goods
Hospital
International
Automotive
Wellness
Investment
Care
Pharmaceuticals
Fitness
Primary
Engineering
Online
Learning
Airlines
Market
Renewables
Devices
Wine
Leisure
Insurance
Tobacco
Translation
Localization
Automation

Seniors + Teen-agers++ staying 5 to 8 years in HEP



Mainframes & workstations &&++

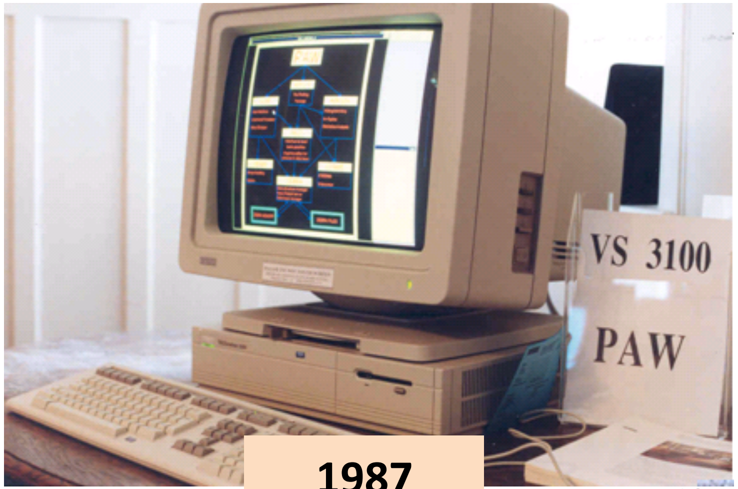
1982



Apollo

HTV

1980



1987

Computers Hardware/word-length

60 CDC6400

56 BESM6

16 PDP11

16/32/48 NORD10/50

16/32 CII

36 UNIVAC

60 CDC7600

32 VAX780

PERQ

Transputers

64 CYBER205

64 CRAY

32/64 IBM3090

32 Apollo

32 DEC

SGI 32/64

32/64 CM5

32 SUN

32/64 HP

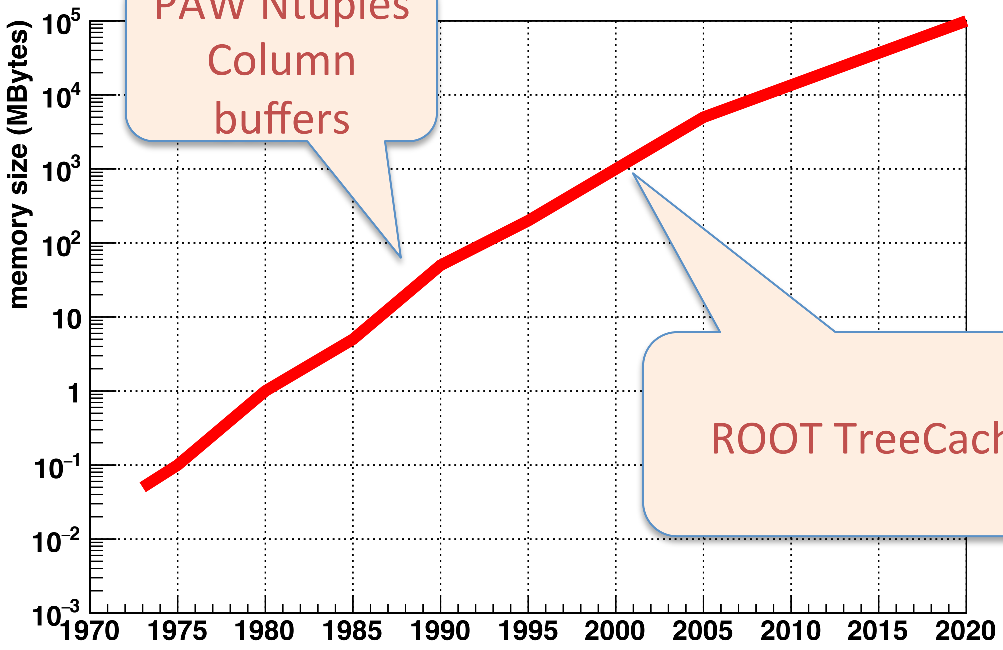
IBM 32/64

32/64 Pentium

32/64 INTELyy

32/64 INTELzz

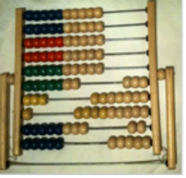
32/64 GPUs, FPGA,etc



PAW Ntuples Column buffers

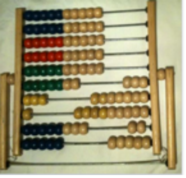
ROOT TreeCache

PM/ML Classification cache



Mainframes->Grids->HPCs

- Large HPCs are now appearing with more than 10000 cores and associated processors (GPUs, FPGA, ASIC,…
- Moving from pure CPUs to CPUs + adds is going to change very soon the software infrastructure, in particular considering the fantastic grow of ML possibilities in HEP, etc
- Petabytes are local. Dcache, Xrootd
- Other caches

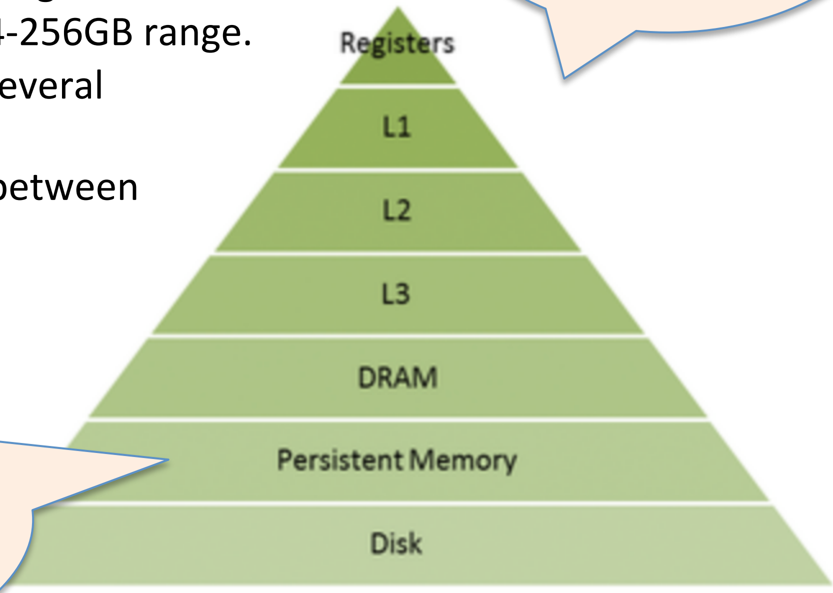


Persistent Memory

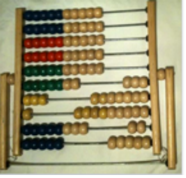
Performance (throughput and latency) is much, much better than disk but slower than DRAM.Targeted to be cheaper than DRAM but more expensive than disk or flash.

PM is bit addressable, which means you can perform I/O to it as if it were memory. In terms of capacity, persistent memory will be about an order magnitude larger than DRAM. Typical compute nodes are in the 64-256GB range. Persistent memory will be on the order of several terabytes at first.
Consequently, it could be a good trade-off between performance, cost, and memory capacity.

Probably a large impact on ML applications



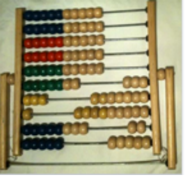
Hello ROOT,
We had branch buffers(kBytes),
TreeCache (30 Mbytes)
Please be ready for PM/ML caches (100 Gbytes)



Quantum Computers

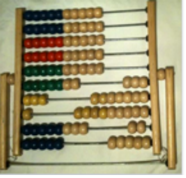
It’s the 1940s again: IBM’s Scott Crowder on the infancy of quantum computers
https://www.digitaltrends.com/computing/ibms-scott-crowder-on-the-infancy-of-quantum-computers/

- A real quantum leap
- “Absolutely,” Crowder replied when we asked him whether IBM expected to see quantum and classical computers co-exist going forward. “Absolutely, for much longer than my lifetime.”
- “What classical computers do well is store and process lots of data, and they do that much more efficiently than quantum computing does today, and for as far as I can see in the future,” he explained. “Theoretically, universal quantum computers can do any function that a classical computer does, but in my opinion, they’re not as good at processing lots of data. What quantum computers are good at is exploring large problem spaces.”
- IBM isn’t looking at quantum computers as the next generation of classical computers. It’s an entirely new category of hardware that has its own strengths, its own weaknesses, and the potential for some very powerful applications.
- Of course, it’s still immature technology. Researchers all over the world have made massive advances toward a working large-scale universal quantum computer in recent years, but less headway has been made in terms of hashing out potential applications. It’s something of a vicious cycle.
- “With a roughly 20-25 qubit system, you can still simulate that quantum system on your laptop,” said Crowder. “When you start getting to 40s, and around 50, you can’t even really explore that entire possibility space on the world’s largest supercomputer. It’s an interesting discontinuity.
- “You actually need to have access to a quantum system to really understand how you build algorithms and use cases for a quantum system,” he added. “You can’t simulate it on a classical system anymore.”



Software, Libs, Tools, Languages, Data Models

- -1974: Fortran IV, HBOOK, GD3, HYDRA, CERNLIB, EGS1, SIGMA
- -1979: Fortran 77, Pascal?, CERNLIB, GEANT2, GEANT3, EGS3, Tatina
- -1984: ADA?, OCCAM?, ZEBRA, GEANT3, GEP, Gheisha, PAW, GKS, X11, Vectorization
- -1989: AIHEP/ACAT, CHEP, PHIGS, MOTIF, Fortran90?, Modula, C, Pearl,
- Column-Wise ntuples, Oracle, WEB!?
- -1994: OO, Smalltalk, Eiffel, ObjectiveC, C++, Objectivity, MOOSE, GEANT4, ROOT, MOSAIC
- -1999: JAS, ROOT+, R, Mathematica, Netscape, TMVA, ROOFIT, PROOF, XROOTD, GPUs/graphics
- -2004: Google, Python
- -2009: Julia, LLVM
- -2014: GPUs/AI,ML, GO, RUST, CLING, MasterClasses
- -2019: This conference, user support



Programing languages

Efficiency

Simplicity

SIGMA COMIS CINT CLING

Modula2 PERL Python

ADA

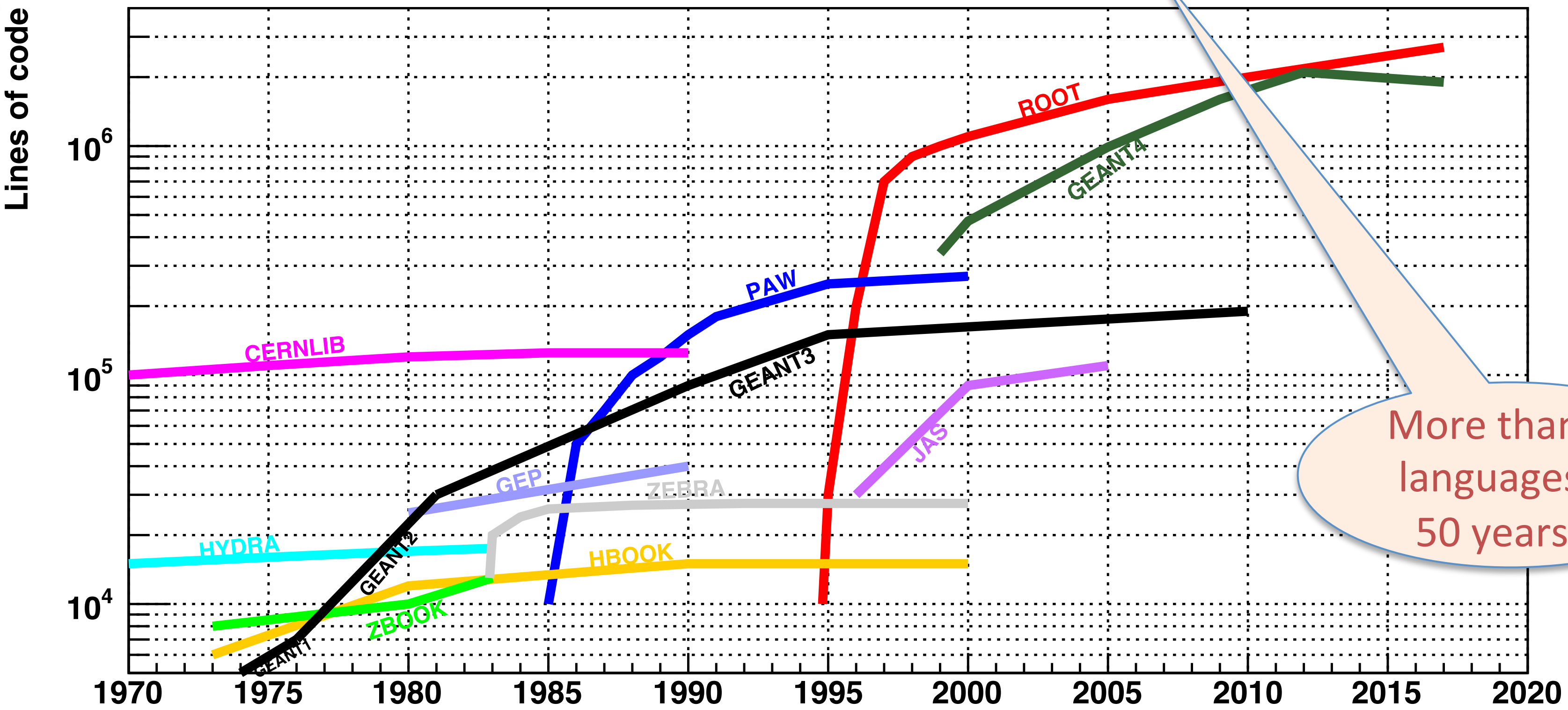
Algol Mortran OCCAM Eiffel Csharp Ruby Javascript

Oberon

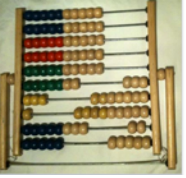
Pascal

Fortran 4,77,90 ObjectiveC Java GO CUDA

Assembler Compass PL1 C Prolog Basic C++ Julia



More than 30 languages in 50 years !!

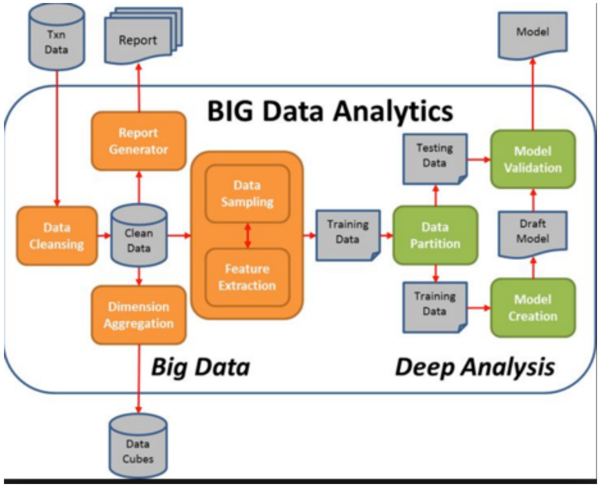


Languages problems

- No introspection (yes in Java, maybe in C++21)
- Pointers are references only
  - Memory leaks, double delete, garbage collection
  - Graphs only. Hard to draw/document structures
  - We need structure pointers
- Collections object-wise, eg Avector(T)
  - Duplicate structures for vectorization
  - Not good for array-oriented systems

All these problems reported to the C++ committee in 1995->2000

The I/O challenge



Txn Data

Report

Model

BIG Data Analytics

Report Generator

Testing Data

Model Validation

Data Sampling

Data Cleansing

Clean Data

Training Data

Data Partition

Draft Model

Feature Extraction

Training Data

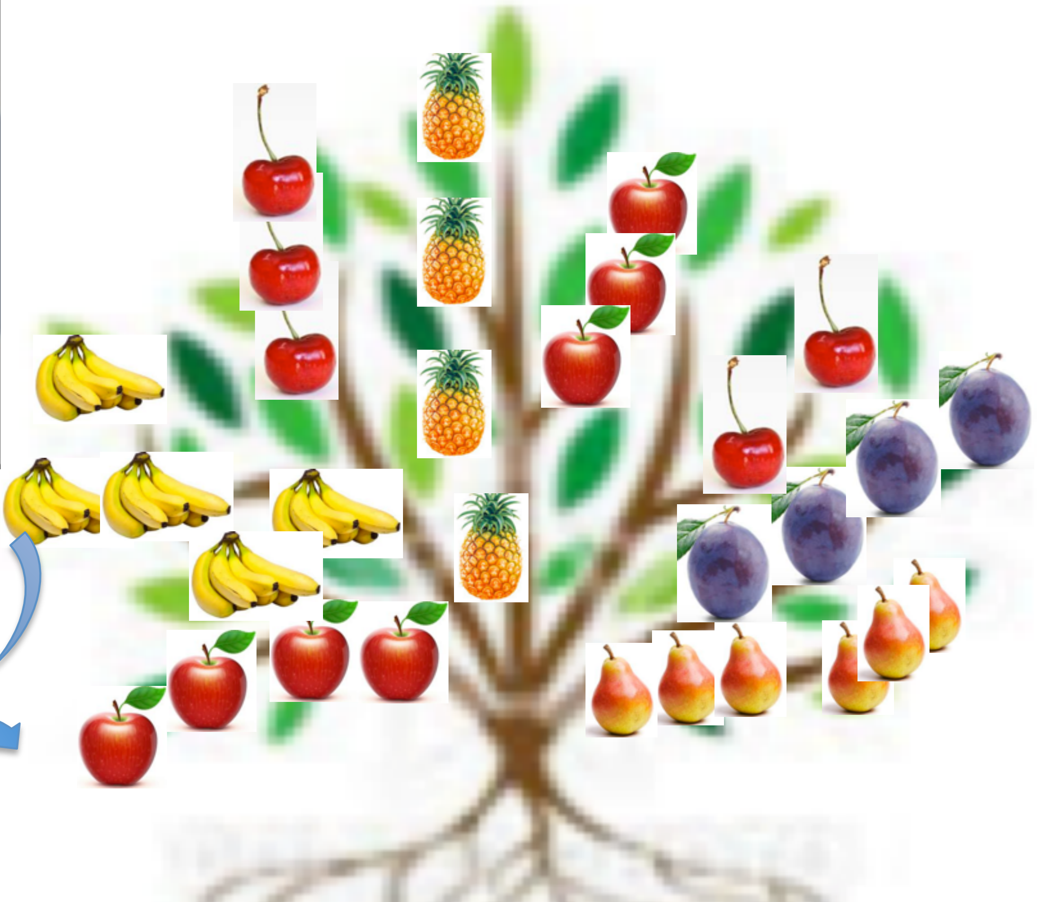
Model Creation

Dimension Aggregation

Big Data

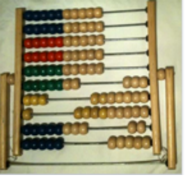
Deep Analysis

Data Cubes



Self describing files, dictionaries

Automatic schema evolution



Caches are important

- Network file access: Dcache, Xrootd
- File structure: Member-wise streaming
- Memory buffers: TreeCache

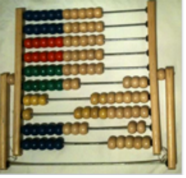
File type Cache on/off	Local file	LAN =0.3ms 1 Gbit/s	WLAN = 3ms 100 Mbits/s	ADSL=72ms 8 Mbits/s

Atlas orig
CA=OFF
	RT=720s
	TR=1328587
		RT=1156s
		NT=400+12s
			RT=4723s
			NT=4000+120s
				RT > 2 days
				NT>2 days

Atlas orig
CA=ON
	RT=132s
	TR=323
		RT=144s
		NT=0.1+12s
			RT=253s
			NT=1+120s
				RT=1575s
				NT=223+1200s

Atlas flushed
CA=OFF
	RT=354s
	TR=486931
		RT=485s
		NT=120+11s
			RT=1684
			NT=1200+110s
				RT=
				NT>1 day

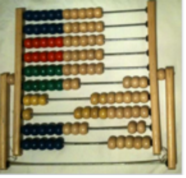
Atlas flushed
CA=ON
	RT=106s
	TR=45
		RT=117s
		NT=0.03+11s
			RT=217s
			NT=0.1+110s
				RT=1231s
				NT=25+1100s



Vectorization

- GEANT3: CrayXmp in 1983 (factor 2)
- GEANT3:Cyber205 in 1987 (factor 3)
- GEANT4:Intel in 2012-19 (factor 2)

Is the effective gain worth the manpower investment when considering long term support ?

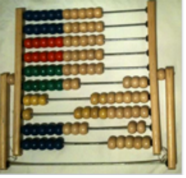


Parallelism

- Real Time vs Throughput problem
- Multi-process (PIAF, PROOF,…)
- Multi-Threading
  - Gain in I/O
  - Gain in memory (or loss)
- Main CPU + GPUs
  - GEANT ?
  - ML: Tensorflow, Python tools
- HPCs

My system uses in average
10 out of 12 cores
A success?

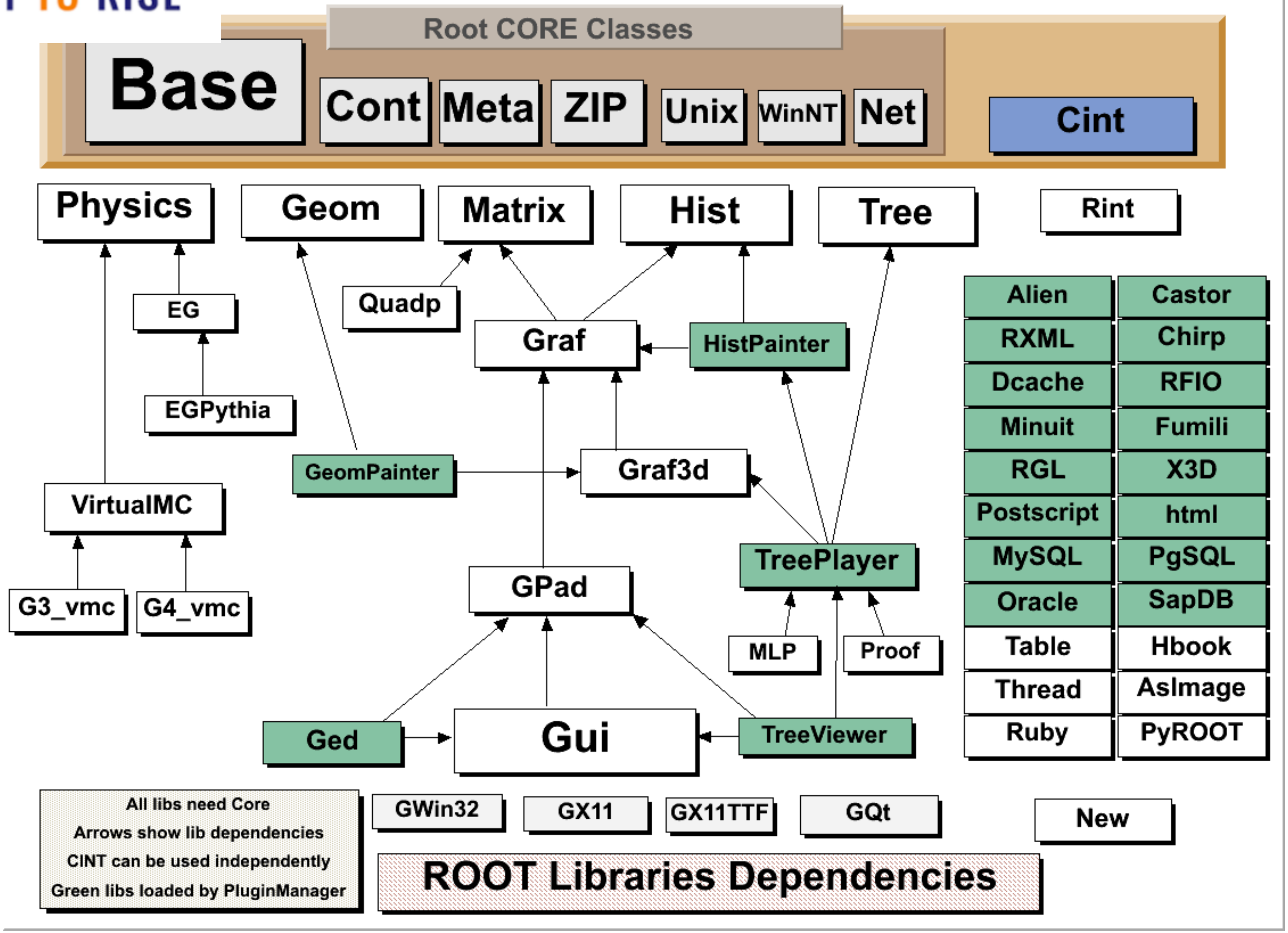
I have a dedicated machine
with 72 cores.
In one night I can do what
other people do in one month



Modularity & Coherence

- In a growing and more diverse software world we see a graph/tree of related components.
- What are the limits of a project?
- How do you maintain and distribute the components?

Current ROOT structure & libs (2002)



Root CORE Classes

Base

Cont

Meta

ZIP

Unix

WinNT

Net

Cint

Physics

Geom

Matrix

Hist

Tree

Rint

EG

EGPythia

Quadp

Graf

HistPainter

GeomPainter

Graf3d

VirtualMC

G3_vmc

G4_vmc

GPad

TreePlayer

MLP

Proof

Ged

Gui

TreeViewer

Alien

Castor

RXML

Chirp

Dcache

RFIO

Minuit

Fumili

RGL

X3D

Postscript

html

MySQL

PgSQL

Oracle

SapDB

Table

Hbook

Thread

AsImage

Ruby

PyROOT

All libs need Core
Arrows show lib dependencies
CINT can be used independently
Green libs loaded by PluginManager

GWin32

GX11

GX11TTF

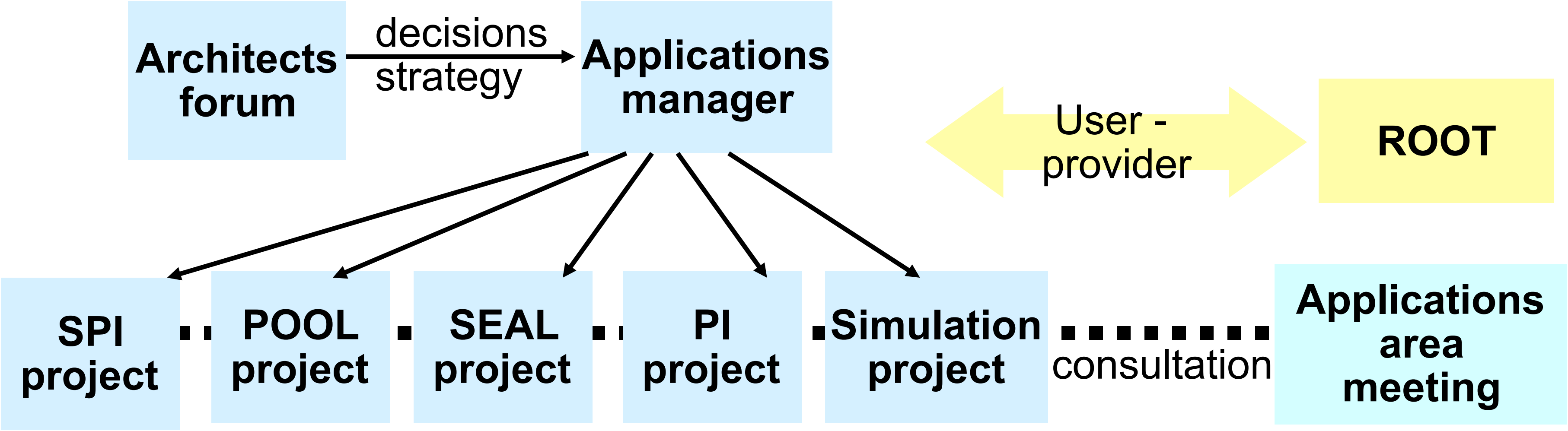
GQt

New

ROOT Libraries Dependencies

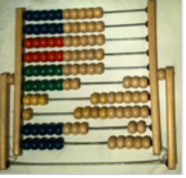
"Every problem has its root in failure of a relationship"

Applications Area Organization in 2002



We are currently discussing with our colleagues in LCG/AA to see if a convergence on key items is possible in the medium & long term.
With SEAL a possible cooperation is envisaged for

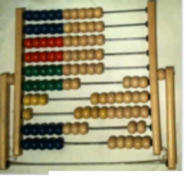
- a common Dictionary approach
- the design/implementation of a MATHLIB

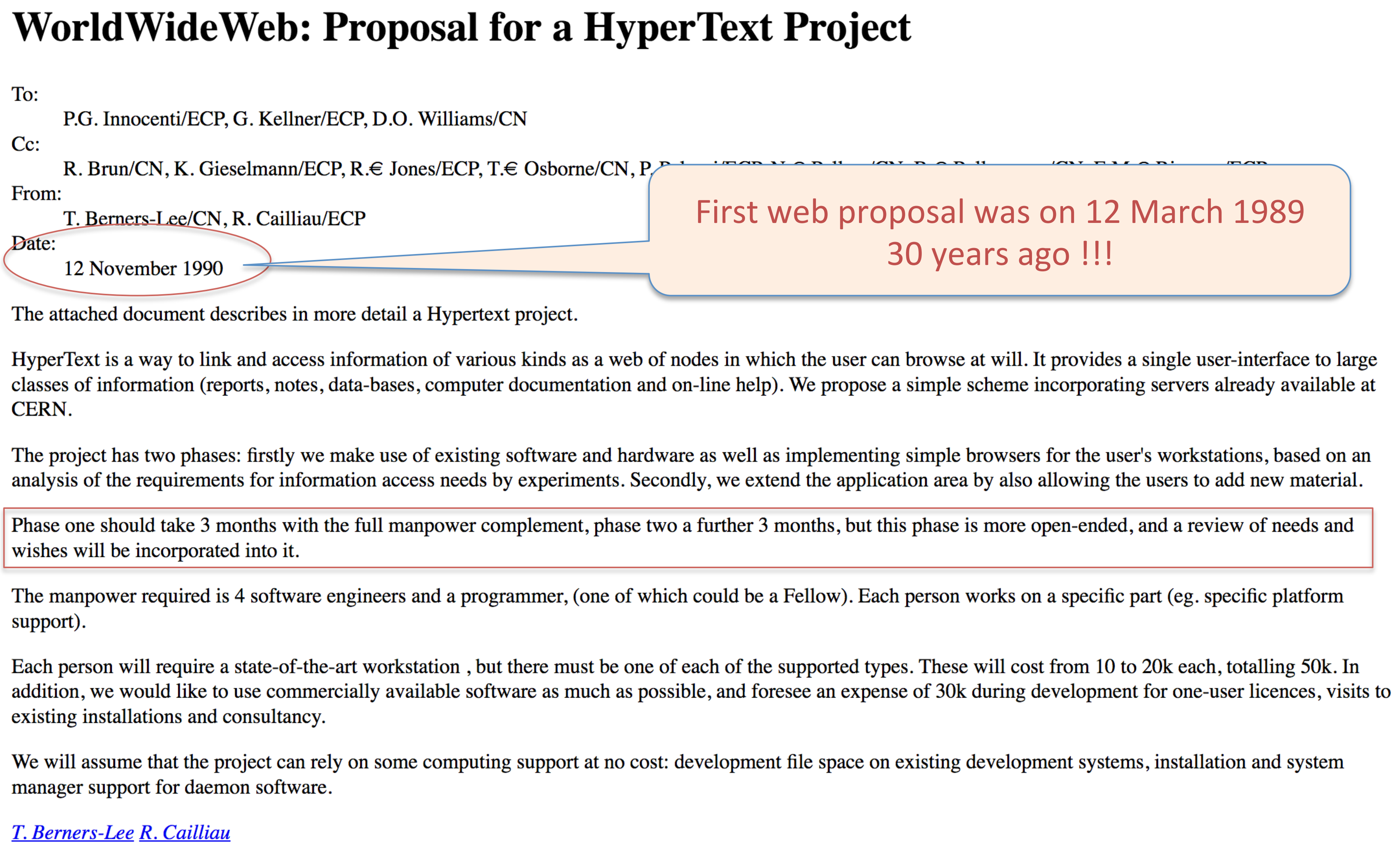


The Future(?) of HEP Computing

- One can predict some aspects of the future based on extrapolations of the past.
- However, surprises, including big surprises are always possible!!
- Eg, it is amusing a posteriori, to read the concluding talks of the CHEP97, CHEP2000 conferences.

And probably reading this talk in 2030





WorldWideWeb: Proposal for a HyperText Project

To:

P.G. Innocenti/ECP, G. Kellner/ECP, D.O. Williams/CN

Cc:

R. Brun/CN, K. Gieselmann/ECP, R.€ Jones/ECP, T.€ Osborne/CN, P.

From:

T. Berners-Lee/CN, R. Cailliau/ECP

Date:

12 November 1990

First web proposal was on 12 March 1989
30 years ago !!!

The attached document describes in more detail a Hypertext project.

HyperText is a way to link and access information of various kinds as a web of nodes in which the user can browse at will. It provides a single user-interface to large classes of information (reports, notes, data-bases, computer documentation and on-line help). We propose a simple scheme incorporating servers already available at CERN.

The project has two phases: firstly we make use of existing software and hardware as well as implementing simple browsers for the user's workstations, based on an analysis of the requirements for information access needs by experiments. Secondly, we extend the application area by also allowing the users to add new material.

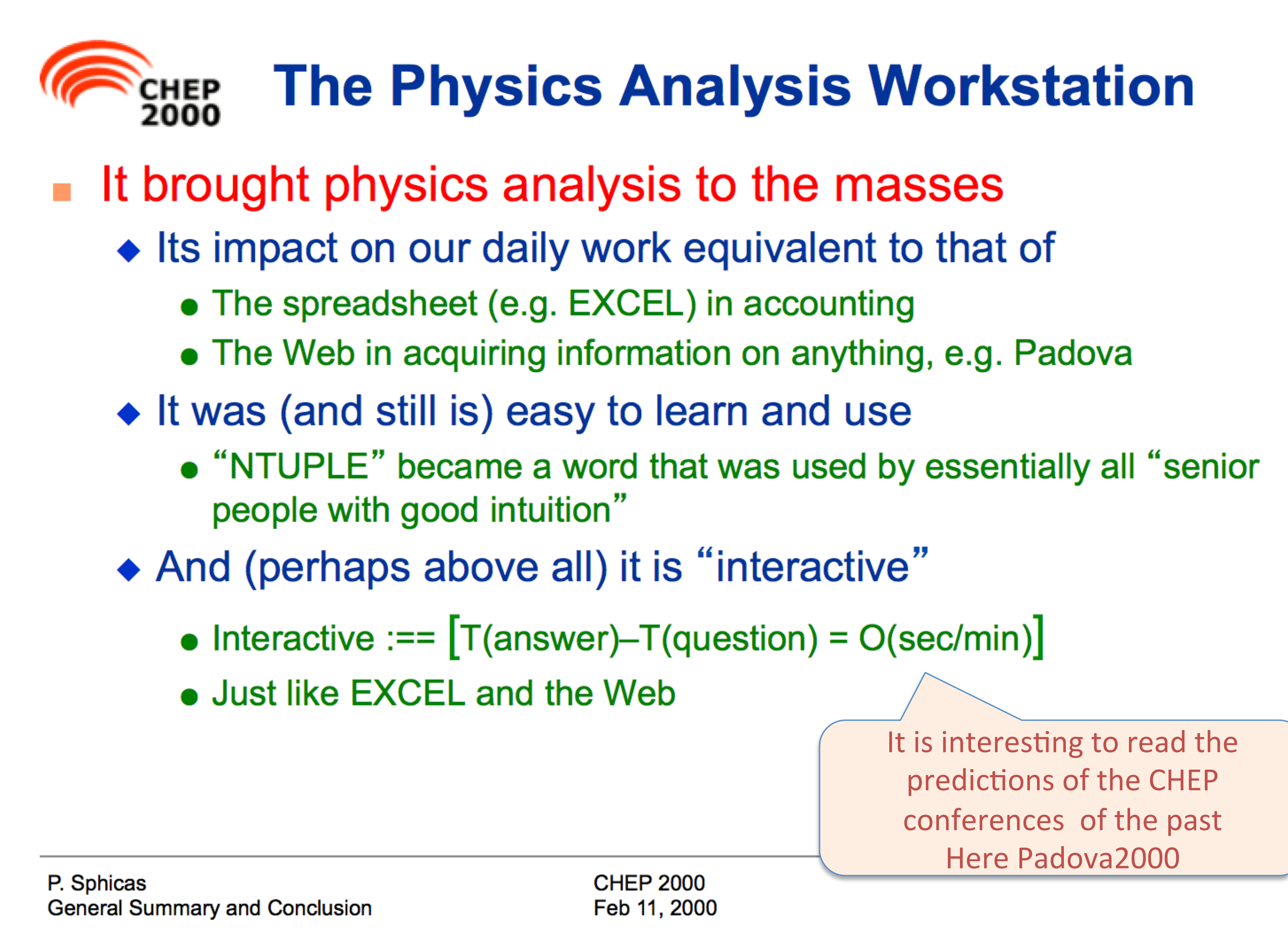
Phase one should take 3 months with the full manpower complement, phase two a further 3 months, but this phase is more open-ended, and a review of needs and wishes will be incorporated into it.

The manpower required is 4 software engineers and a programmer, (one of which could be a Fellow). Each person works on a specific part (eg. specific platform support).

Each person will require a state-of-the-art workstation , but there must be one of each of the supported types. These will cost from 10 to 20k each, totalling 50k. In addition, we would like to use commercially available software as much as possible, and foresee an expense of 30k during development for one-user licences, visits to existing installations and consultancy.

We will assume that the project can rely on some computing support at no cost: development file space on existing development systems, installation and system manager support for daemon software.

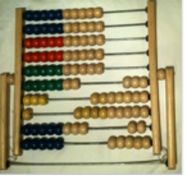
T. Berners-Lee *R. Cailliau*



The Physics Analysis Workstation

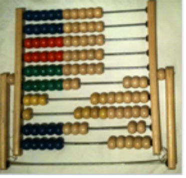
- It brought physics analysis to the masses
  - Its impact on our daily work equivalent to that of
    - The spreadsheet (e.g. EXCEL) in accounting
    - The Web in acquiring information on anything, e.g. Padova
  - It was (and still is) easy to learn and use
    - "NTUPLE" became a word that was used by essentially all "senior people with good intuition"
  - And (perhaps above all) it is "interactive"
    - Interactive :== [T(answer)–T(question) = O(sec/min)]
    - Just like EXCEL and the Web

It is interesting to read the predictions of the CHEP conferences of the past
Here Padova2000



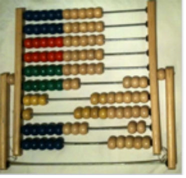
Wishes & Dreams

- As a senior++ with 45 years spent in HEP software, I see fantastic opportunities for teen-agers++ for launching new projects or important improvements in existing projects.
- I will just mention a few in the coming slides.
- None of the successful projects in the past has been launched by a committee.
- The main idea of a new proposal must be implemented in a few months (not years).
- Of course, existing projects continue their linear development in functionality and performance.



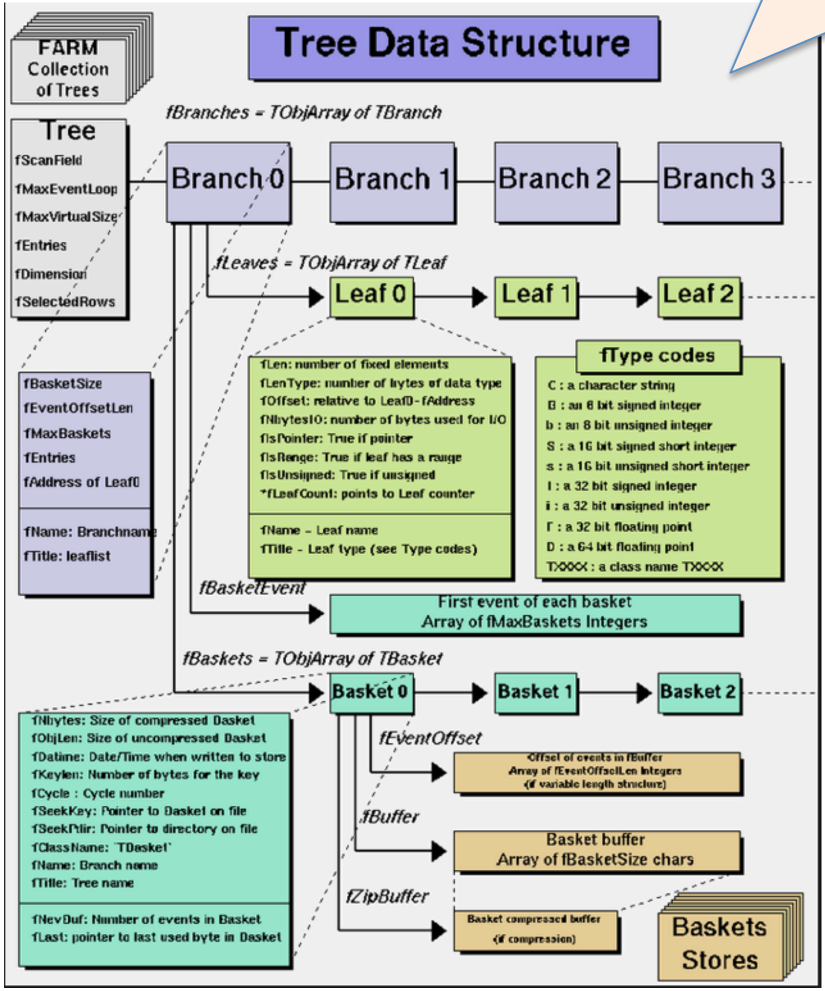
Project1

- Simplify user interface for most users
- Facilitate learning & evaluation of new languages
- Interactive navigation in data structures and programs structure with associated documentations (resurrect TTreeViewer)



Document Data Structures: a MUST

These 2 diagrams were produced
in 1995 and are still valid



fBasketSize
fEventOffsetLen
fMaxBaskets
fEntries
fAddress of Leaf0

fName: Branchname
fTitle: leaflist

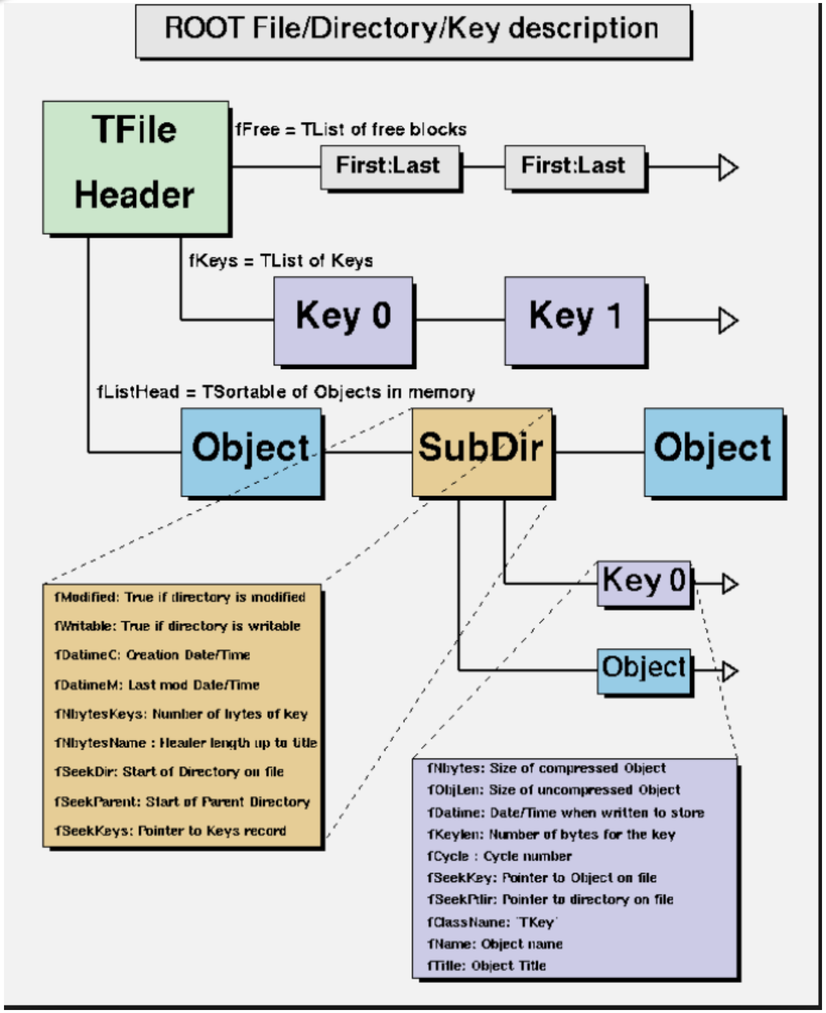
fLen: number of fixed elements
fLenType: number of bytes of data type
fOffset: relative to Leaf0-fAddress
fNbytesIO: number of bytes used for I/O
fIsPointer: True if pointer
fIsRange: True if leaf has a range
fIsUnsigned: True if unsigned
*fLeafCount: points to Leaf counter

fName – Leaf name
fTitle – Leaf type (see Type codes)

C : a character string
B : an 8 bit signed integer
b : an 8 bit unsigned integer
S : a 16 bit signed short integer
s : a 16 bit unsigned short integer
I : a 32 bit signed integer
i : a 32 bit unsigned integer
F : a 32 bit floating point
D : a 64 bit floating point
TXXXX : a class name TXXXX

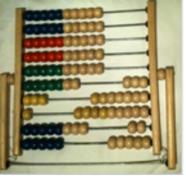
fNbytes: Size of compressed Basket
fObjLen: Size of uncompressed Basket
fDatime: Date/Time when written to store
fKeylen: Number of bytes for the key
fCycle : Cycle number
fSeekKey: Pointer to Basket on file
fSeekPdir: Pointer to directory on file
fClassName: 'TBasket'
fName: Branch name
fTitle: Tree name

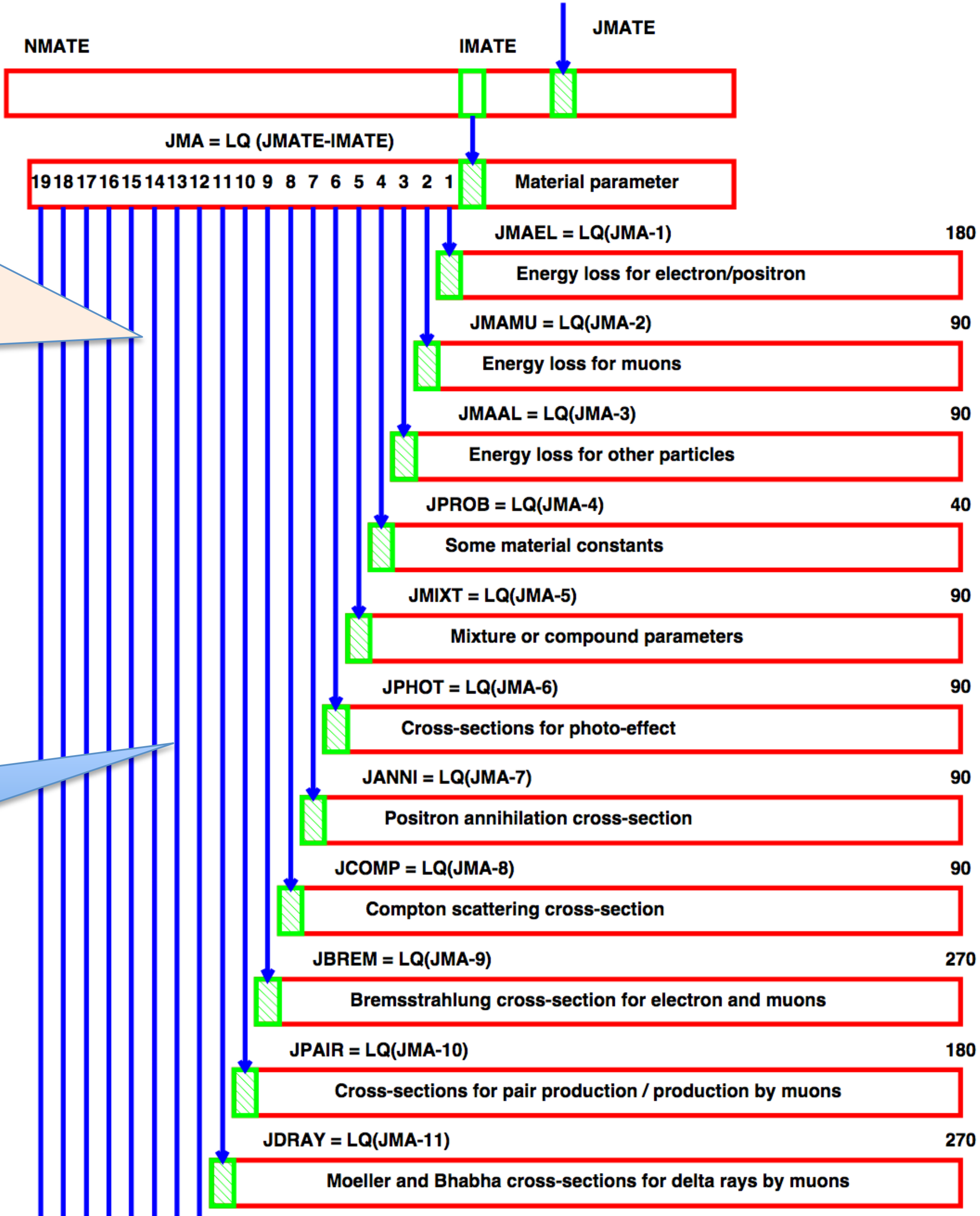
fNevBuf: Number of events in Basket
fLast: pointer to last used byte in Basket


ROOT File/Directory/Key description
TFile
Header
fFree = TList of free blocks
First:Last
First:Last
fKeys = TList of Keys
Key 0
Key 1
fListHead = TSortable of Objects in memory
Object
SubDir
Object
Key 0
Object

fModified: True if directory is modified
fWritable: True if directory is writable
fDatimeC: Creation Date/Time
fDatimeM: Last mod Date/Time
fNbytesKeys: Number of bytes of key
fNbytesName : Header length up to title
fSeekDir: Start of Directory on file
fSeekParent: Start of Parent Directory
fSeekKeys: Pointer to Keys record

fNbytes: Size of compressed Object
fObjLen: Size of uncompressed Object
fDatime: Date/Time when written to store
fKeylen: Number of bytes for the key
fCycle : Cycle number
fSeekKey: Pointer to Object on file
fSeekPdir: Pointer to directory on file
fClassName: 'TKey'
fName: Object name
fTitle: Object Title





NMATE

IMATE

JMATE

JMA = LQ (JMATE-IMATE)

19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

Material parameter

JMAEL = LQ(JMA-1) 180

Energy loss for electron/positron

JMAMU = LQ(JMA-2) 90

Energy loss for muons

JMAAL = LQ(JMA-3) 90

Energy loss for other particles

JPROB = LQ(JMA-4) 40

Some material constants

JMIXT = LQ(JMA-5) 90

Mixture or compound parameters

JPHOT = LQ(JMA-6) 90

Cross-sections for photo-effect

JANNI = LQ(JMA-7) 90

Positron annihilation cross-section

JCOMP = LQ(JMA-8) 90

Compton scattering cross-section

JBREM = LQ(JMA-9) 270

Bremsstrahlung cross-section for electron and muons

JPAIR = LQ(JMA-10) 180

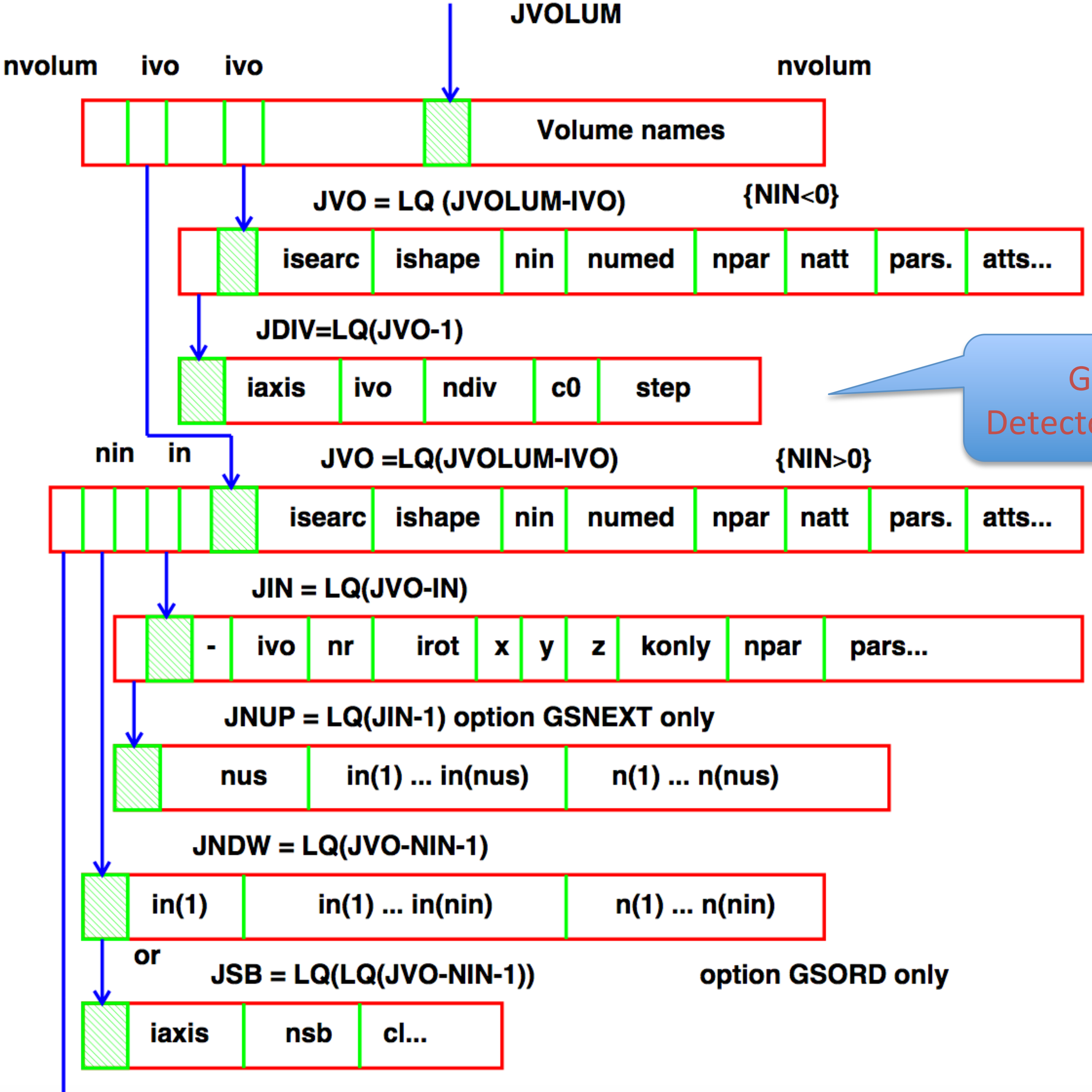
Cross-sections for pair production / production by muons

JDRAY = LQ(JMA-11) 270

Moeller and Bhabha cross-sections for delta rays by muons

GEANT3
ZBOOK→ZEBRA
DZDOC
Automatic
documentation
1983

GEANT3
materials



JVOLUM

nvolum ivo ivo nvolum

Volume names

JVO = LQ (JVOLUM-IVO) {NIN<0}

| isearc | ishape | nin | numed | npar | natt | pars. | atts... |

JDIV=LQ(JVO-1)

| iaxis | ivo | ndiv | c0 | step |

nin in

JVO =LQ(JVOLUM-IVO) {NIN>0}

| isearc | ishape | nin | numed | npar | natt | pars. | atts... |

JIN = LQ(JVO-IN)

| - | ivo | nr | irot | x | y | z | konly | npar | pars... |

JNUP = LQ(JIN-1) option GSNEXT only

| nus | in(1) ... in(nus) | n(1) ... n(nus) |

JNDW = LQ(JVO-NIN-1)

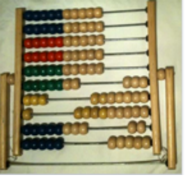
| in(1) | in(1) ... in(nin) | n(1) ... n(nin) |

or

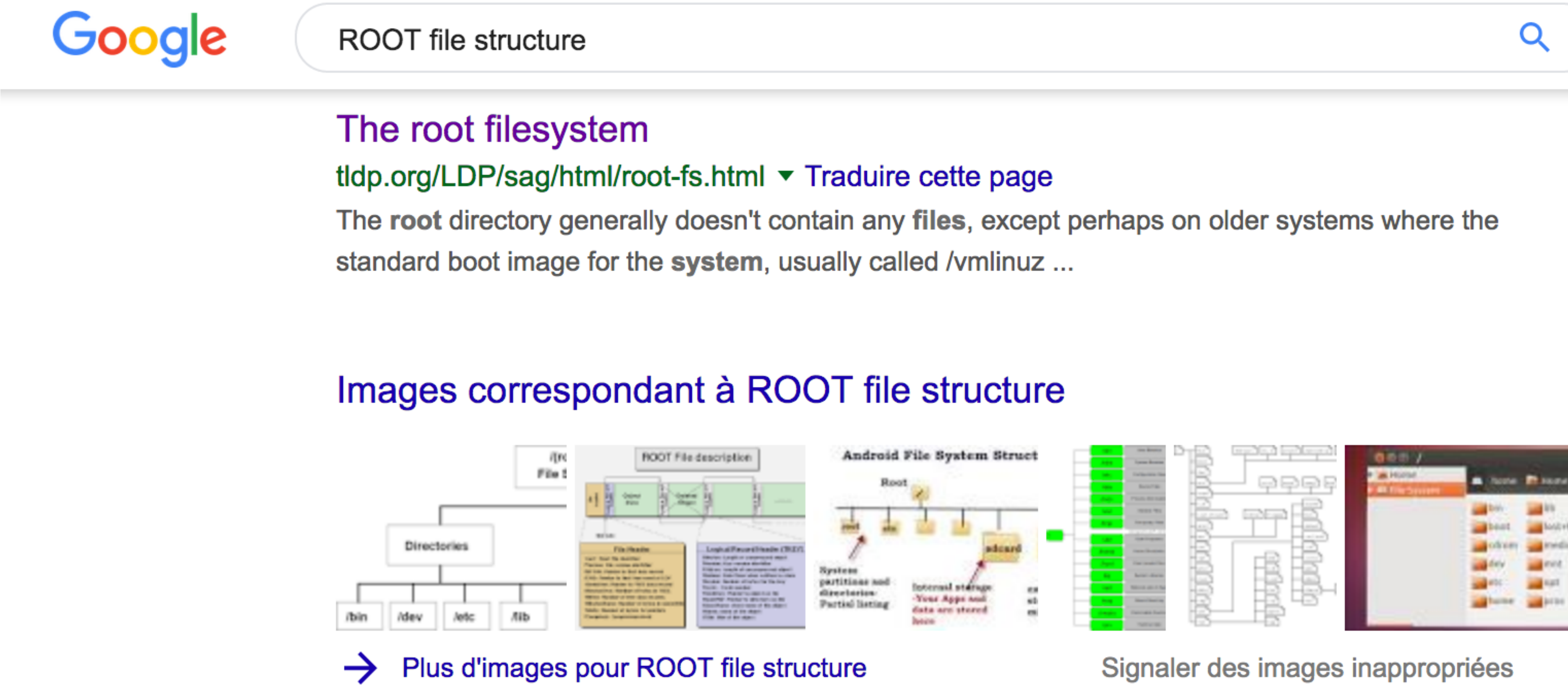
JSB = LQ(LQ(JVO-NIN-1)) option GSORD only

| iaxis | nsb | cl... |

GEANT3
Detector geometry



I want a Google-Earth and Street-View for my data structures



Google

ROOT file structure

The root filesystem

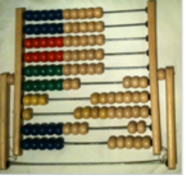
tldp.org/LDP/sag/html/root-fs.html ▾ Traduire cette page

The **root** directory generally doesn't contain any **files**, except perhaps on older systems where the standard boot image for the **system**, usually called /vmlinuz ...

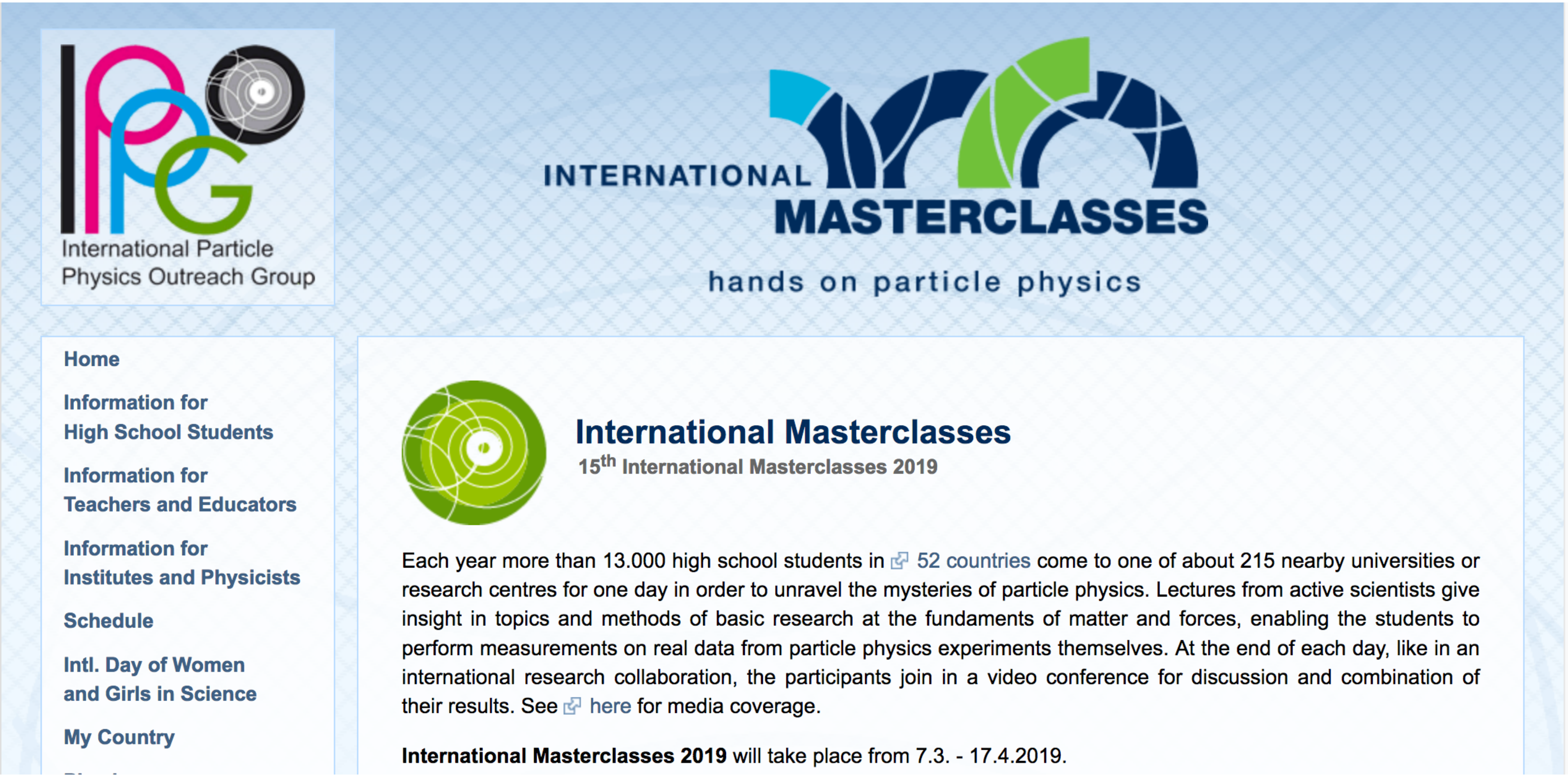
Images correspondant à ROOT file structure

→ Plus d'images pour ROOT file structure

Signaler des images inappropriées



MasterClasses



International Particle Physics Outreach Group
INTERNATIONAL MASTERCLASSES
hands on particle physics

- Home
- Information for High School Students
- Information for Teachers and Educators
- Information for Institutes and Physicists
- Schedule
- Intl. Day of Women and Girls in Science
- My Country

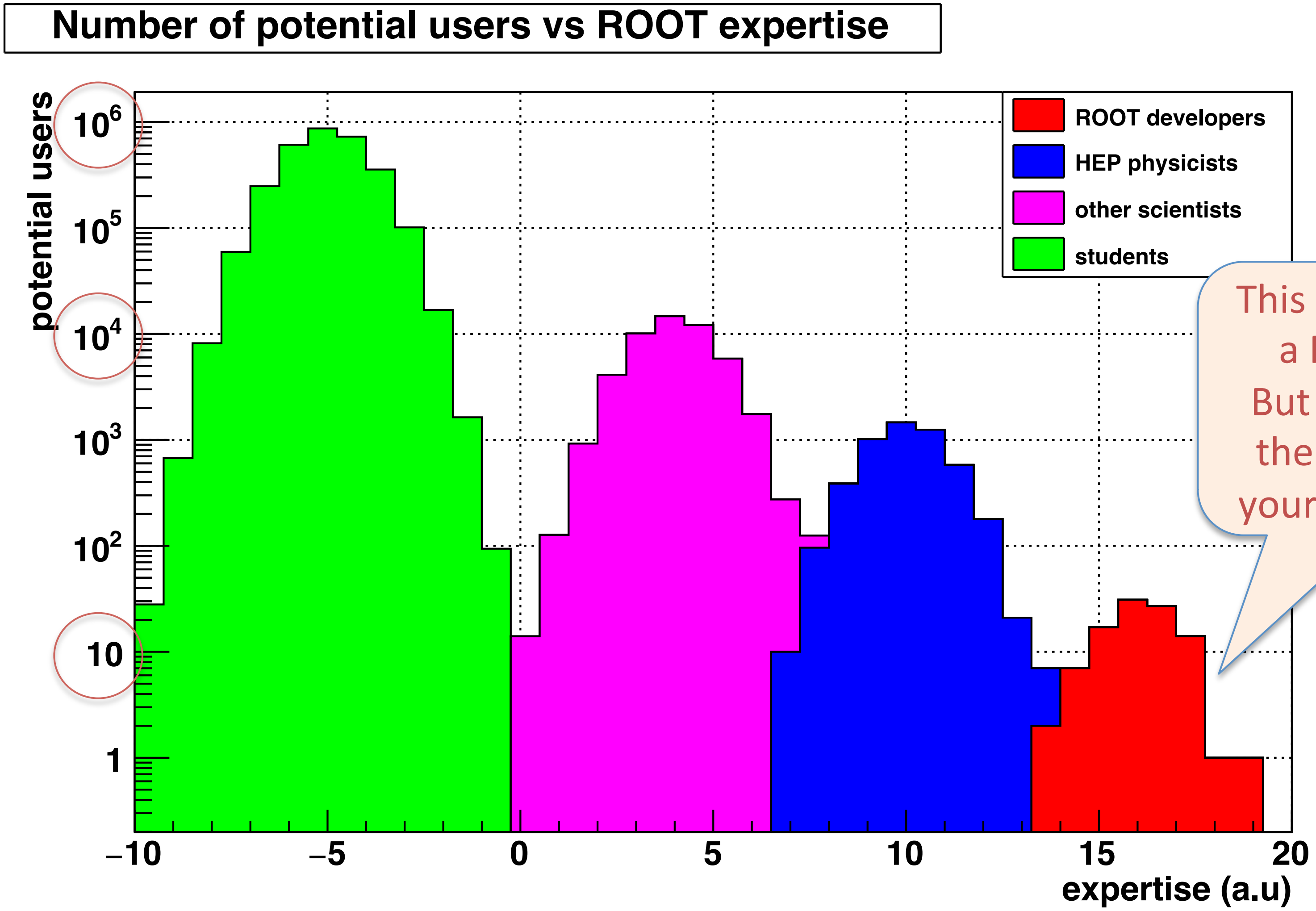
International Masterclasses

15th International Masterclasses 2019

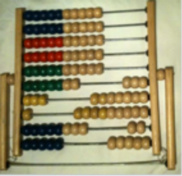
Each year more than 13.000 high school students in 52 countries come to one of about 215 nearby universities or research centres for one day in order to unravel the mysteries of particle physics. Lectures from active scientists give insight in topics and methods of basic research at the fundaments of matter and forces, enabling the students to perform measurements on real data from particle physics experiments themselves. At the end of each day, like in an international research collaboration, the participants join in a video conference for discussion and combination of their results. See here for media coverage.

International Masterclasses 2019 will take place from 7.3. - 17.4.2019.

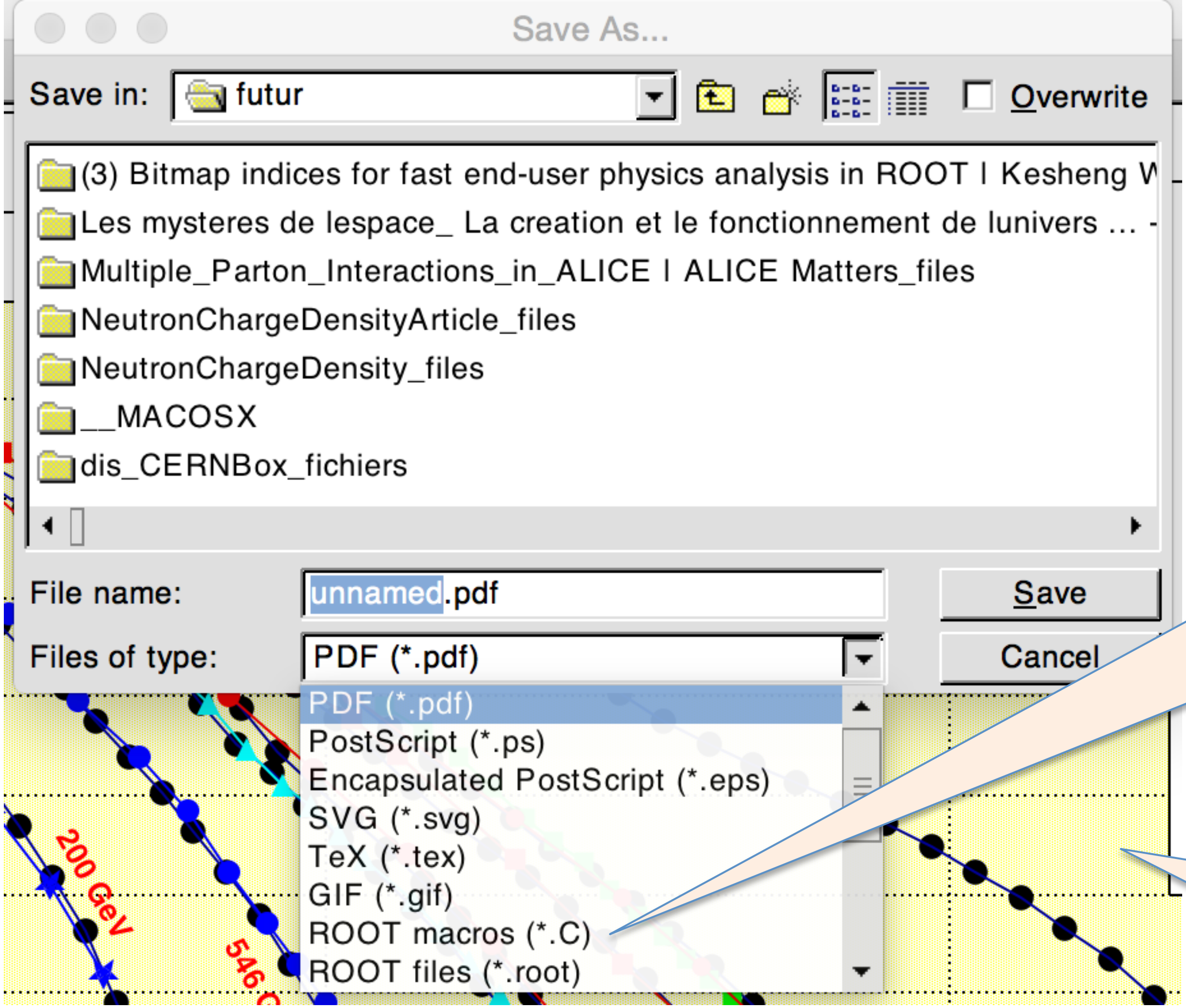
People profiles



This plot as seen by a ROOT expert
But you can make the same plot for your own expertise



Generation of a script from a canvas



Save As...

Save in: futur

□ Overwrite

(3) Bitmap indices for fast end-user physics analysis in ROOT | Kesheng W

Les mysteres de lespace_ La creation et le fonctionnement de lunivers ... -

Multiple_Parton_Interactions_in_ALICE | ALICE Matters_files

NeutronChargeDensityArticle_files

NeutronChargeDensity_files

__MACOSX

dis_CERNBox_fichiers

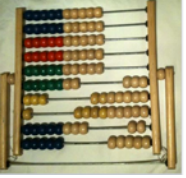
File name: unnamed.pdf

Files of type: PDF (*.pdf)

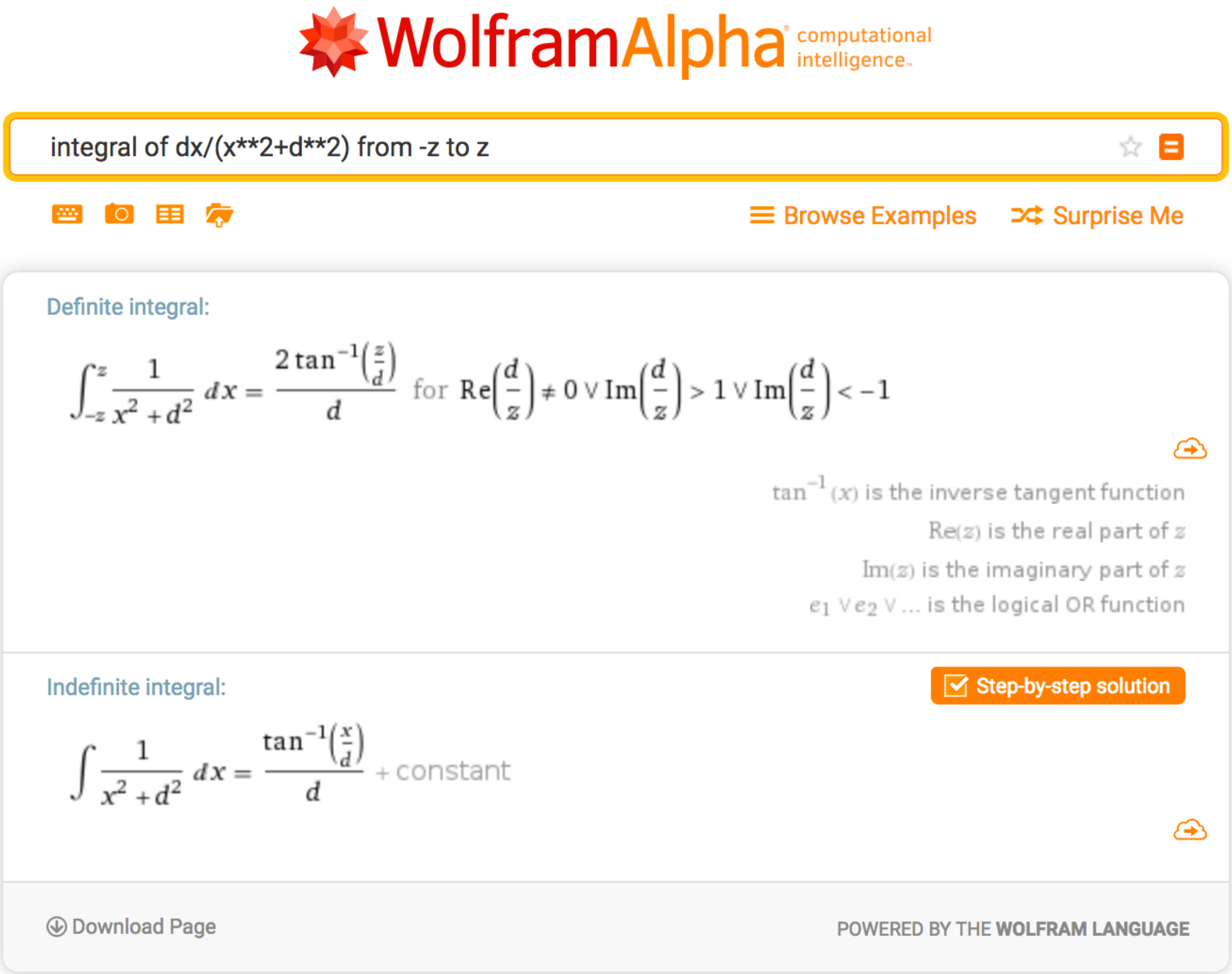
or TreeViewer
or any graphics UI

Generate script.C
Could be
script.py
script.go
etc

+ Google style
help/query



More task oriented UI



WolframAlpha computational intelligence.
Browse Examples
Surprise Me
Step-by-step solution
Download Page
POWERED BY THE WOLFRAM LANGUAGE

integral of dx/(x**2+d**2) from -z to z

Definite integral:

$$\int_{-z}^{z} \frac{1}{x^2+d^2}\,dx = \frac{2\tan^{-1}\left(\frac{z}{d}\right)}{d} \quad \text{for } \operatorname{Re}\left(\frac{d}{z}\right) \neq 0 \lor \operatorname{Im}\left(\frac{d}{z}\right) > 1 \lor \operatorname{Im}\left(\frac{d}{z}\right) < -1$$

$\tan^{-1}(x)$ is the inverse tangent function

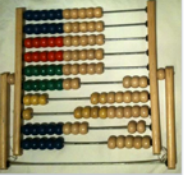
$\operatorname{Re}(z)$ is the real part of z

$\operatorname{Im}(z)$ is the imaginary part of z

$e_1 \lor e_2 \lor \ldots$ is the logical OR function

Indefinite integral:

$$\int \frac{1}{x^2+d^2}\,dx = \frac{\tan^{-1}\left(\frac{x}{d}\right)}{d} + \text{constant}$$



Project2

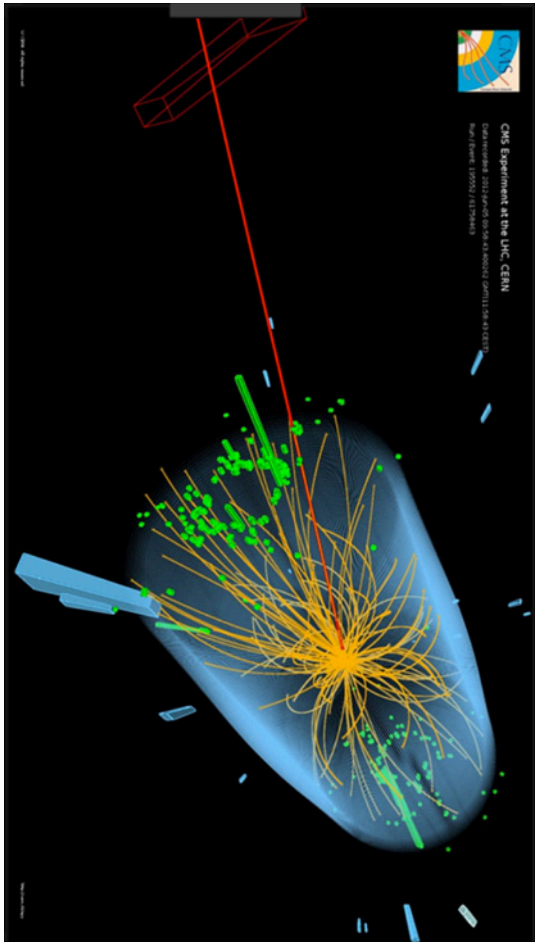
- Implement ML in GEANT transport
  - At the track level, but event view
  - At the jets level
  - In the showers machinery

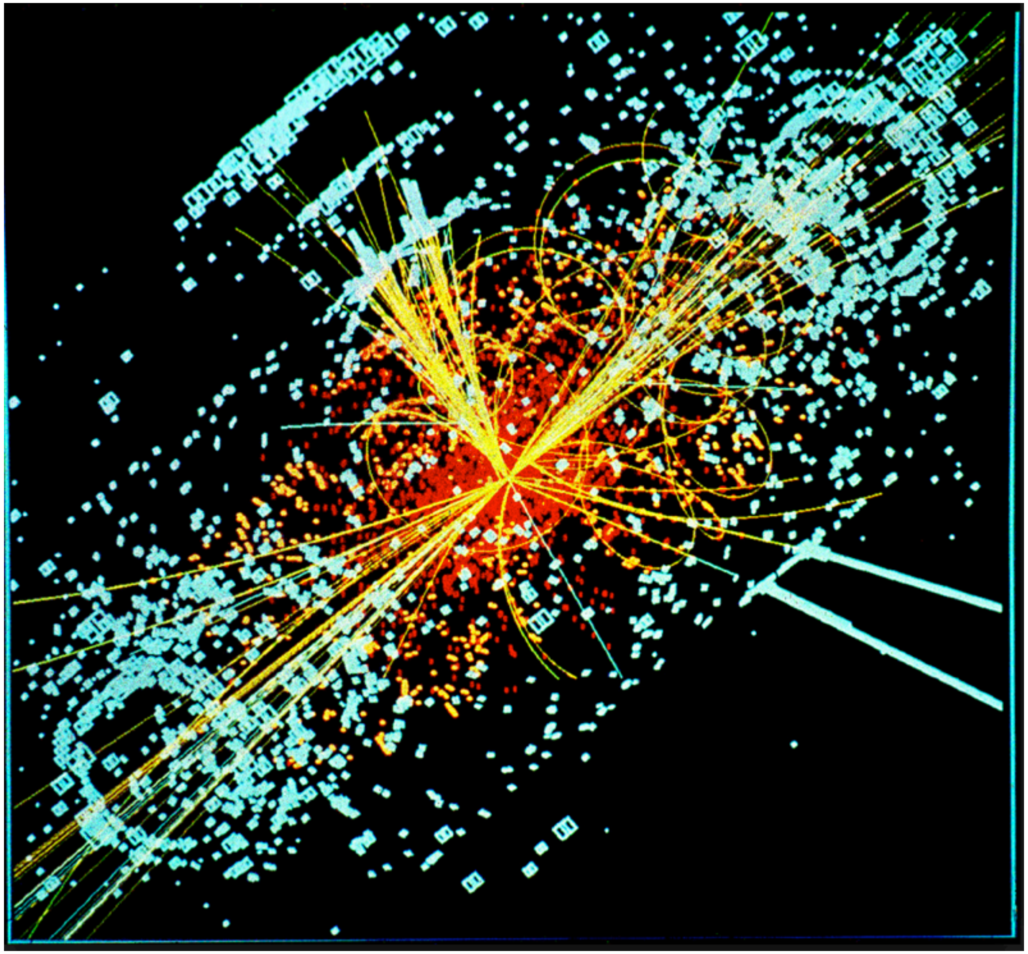
Target gain
5 to 10

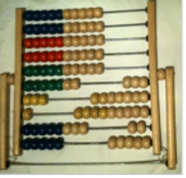
- At the event level (tracking + digits + recons)
  - Given an input list (event gen) predict the tracks and jets reconstruction

Target gain
100 to 1000

But important time spent in training and understanding errors


CMS
CMS Experiment at the LHC, CERN





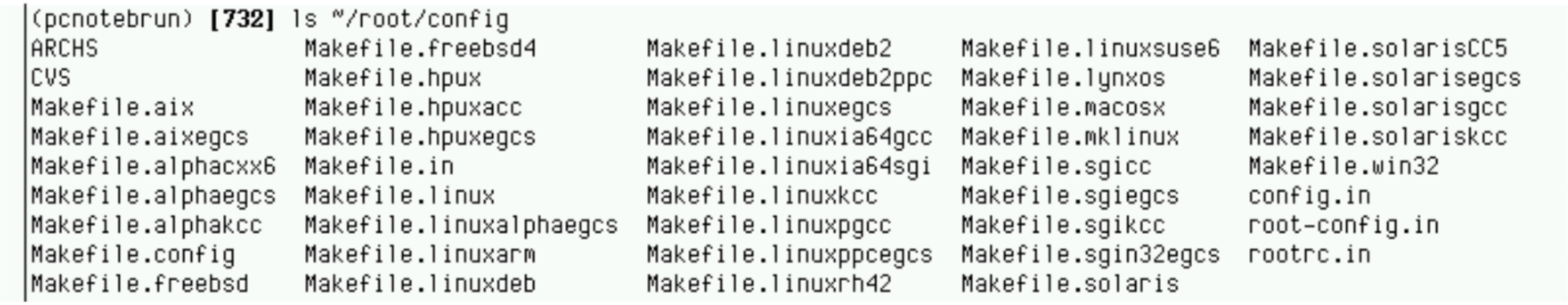
Project3

- Simplify the installation of large systems like GEANT and ROOT for the vast majority of users
- Same idea as downloading an app on your mobile phone.
- Rely on the fact that hundred/thousands of users (cmake experts) have already created, eg ROOTxx.yy on systemX with compiler ZZ, and have published their files on a central point.

Makefiles

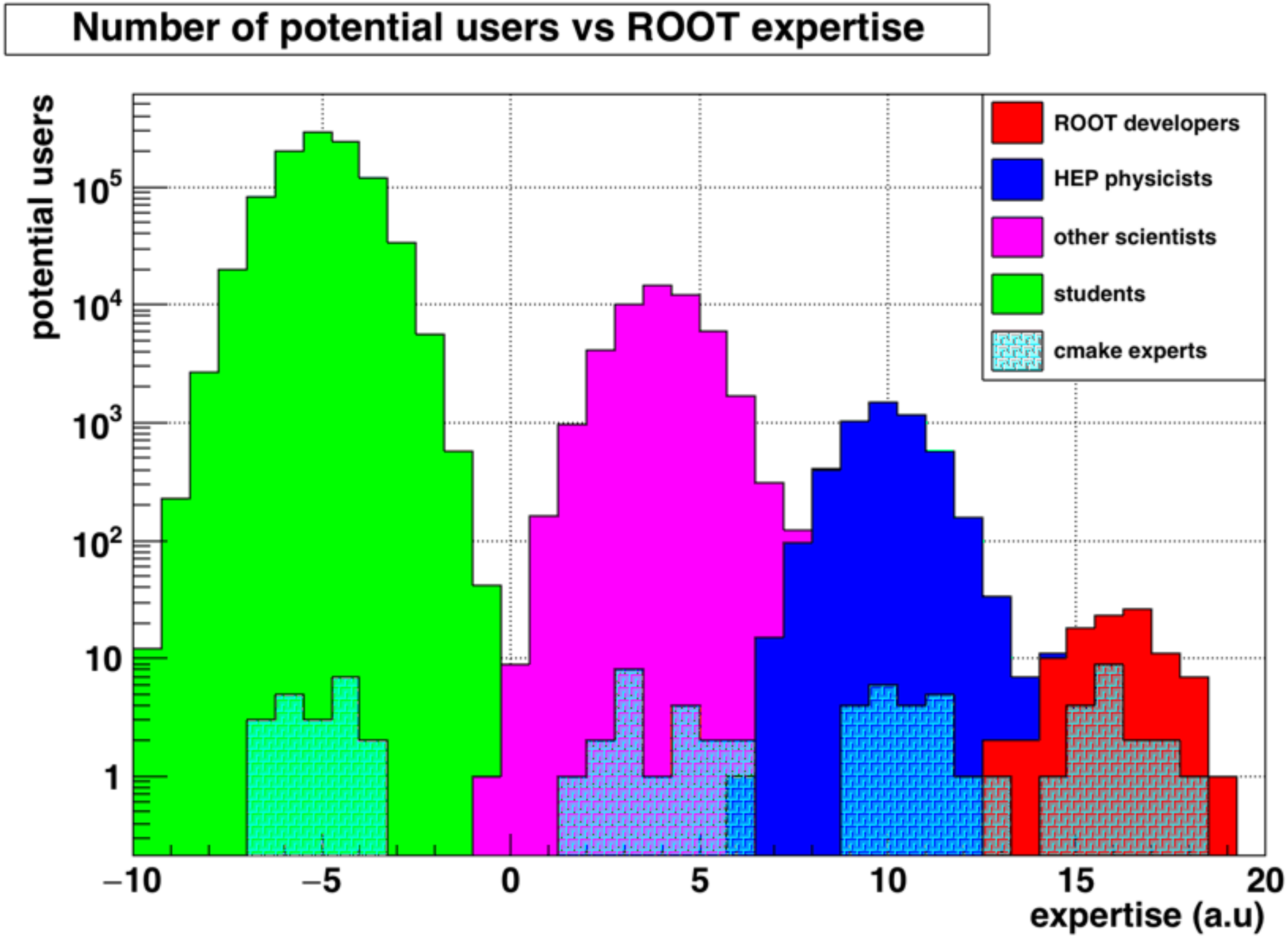
We moved to CVS
In 2001

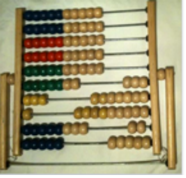
- 3 major OS (Unix, Windows, Mac OS/X)
- 10 different compilers
  - gcc with many flavors on nearly all platforms,
  - Solaris:CC4,5, HPUX:CC:aCC, SGI:CC, AIX:xlC
  - Alpha:CXX6, Windows:VC++6
  - KAI on SGI, Linux, Solaris
- 37 Makefiles



```
(pcnotebrun) [732] ls ~/root/config
ARCHS              Makefile.freebsd4       Makefile.linuxdeb2      Makefile.linuxsuse6  Makefile.solarisCC5
CVS                Makefile.hpux           Makefile.linuxdeb2ppc   Makefile.lynxos      Makefile.solarisegcs
Makefile.aix       Makefile.hpuxacc        Makefile.linuxegcs      Makefile.macosx      Makefile.solarisgcc
Makefile.aixegcs   Makefile.hpuxegcs       Makefile.linuxia64gcc   Makefile.mklinux     Makefile.solariskcc
Makefile.alphacxx6 Makefile.in             Makefile.linuxia64sgi   Makefile.sgicc       Makefile.win32
Makefile.alphaegcs Makefile.linux          Makefile.linuxkcc       Makefile.sgiegcs     config.in
Makefile.alphakcc  Makefile.linuxalphaegcs Makefile.linuxpgcc      Makefile.sgikcc      root-config.in
Makefile.config    Makefile.linuxarm       Makefile.linuxppcegcs   Makefile.sgin32egcs  rootrc.in
Makefile.freebsd   Makefile.linuxdeb       Makefile.linuxrh42      Makefile.solaris
```

People profiles (2)





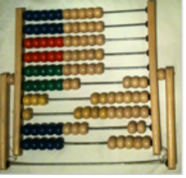
Project3(2)

- Time to install should be < 30s
- Possibility to install a subset only (eg only the most frequently used (say 50) of the 150 shared libs of ROOT.
- Possibility to go back to an earlier version in case of bugs in new versions.

Conditions for success

- Top priority: Instantaneous user support
- Understand & prioritize users requirements
- Project members must follow all branches
- Stability & continuity even with revolutions
- Creativity with short term validation
- Code quality: dash boards, coverity, etc
- Do not duplicate user interfaces
- Simplify installation
- Tools for beginners
- Last but not least

Nothing great
was ever achieved without
enthusiasm.

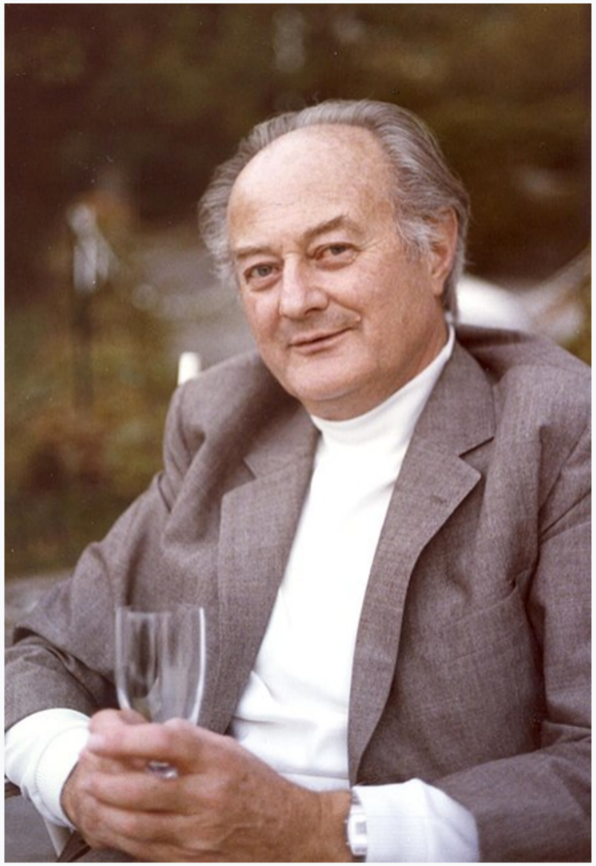


Spare slides

Sigma Language

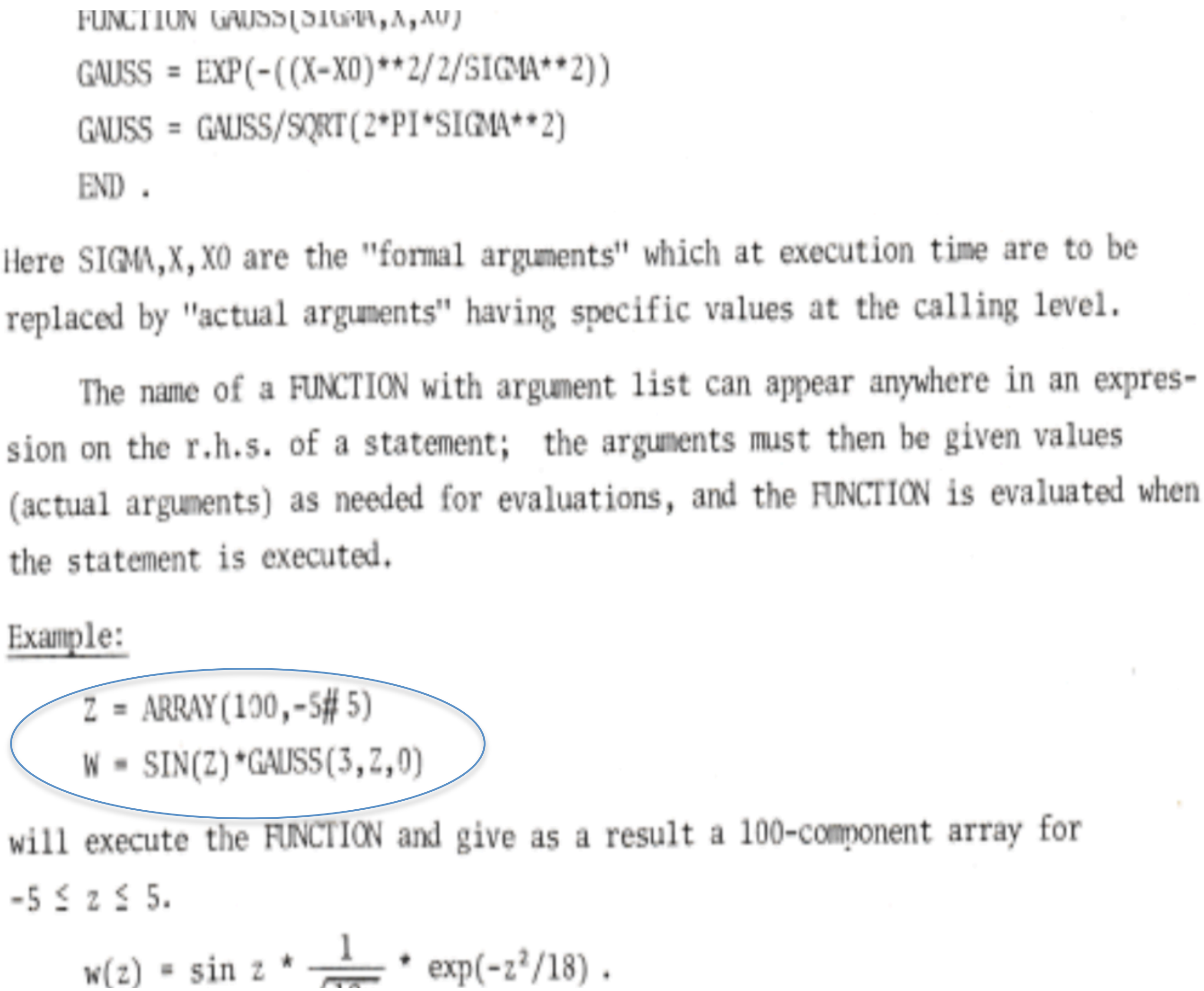
- SIGMA was an interactive, array-oriented system, designed by Rolf Hagedorn in 1978.
- A tiny fraction was included in HTV, then PAW in 1985 by C.Vandoni.
- <u>It was a big mistake not to include it in ROOT !!!</u>

Rolf Hagedorn



R.Hagedorn, J.Reinfelds, C.Vandoni, L.Van Hove: CERN 78-06 (1978)
A new language for interactive array-oriented computing

Functions, arrays, dataframes, Math



```
FUNCTION GAUSS(SIGMA,X,X0)
GAUSS = EXP(-((X-X0)**2/2/SIGMA**2))
GAUSS = GAUSS/SQRT(2*PI*SIGMA**2)
END .
```

Here SIGMA,X,X0 are the "formal arguments" which at execution time are to be replaced by "actual arguments" having specific values at the calling level.

The name of a FUNCTION with argument list can appear anywhere in an expression on the r.h.s. of a statement; the arguments must then be given values (actual arguments) as needed for evaluations, and the FUNCTION is evaluated when the statement is executed.

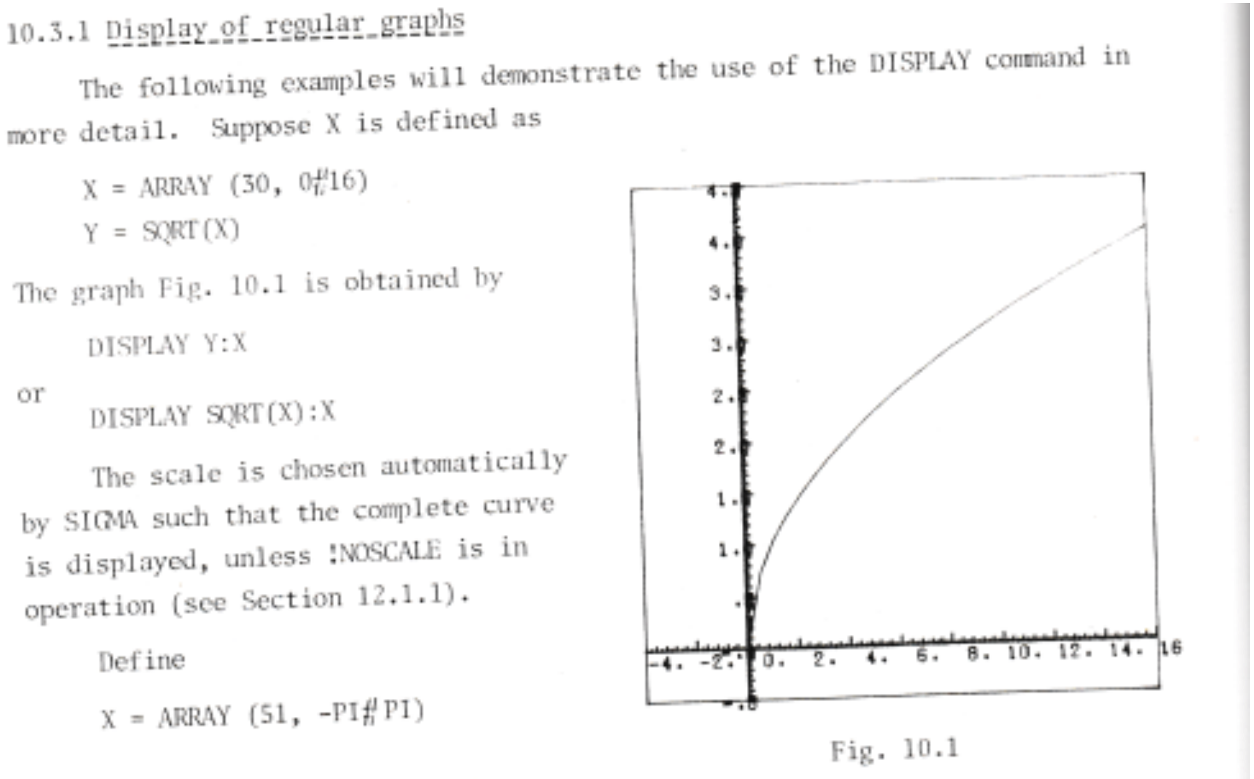
Example:

```
Z = ARRAY(100,-5#5)
W = SIN(Z)*GAUSS(3,Z,0)
```

will execute the FUNCTION and give as a result a 100-component array for $-5 \leq z \leq 5$.

$$w(z) = \sin z * \frac{1}{\sqrt{[illegible]}} * \exp(-z^2/18) .$$

SIGMA early graphics



10.3.1 Display of regular graphs

The following examples will demonstrate the use of the DISPLAY command in more detail. Suppose X is defined as

```
X = ARRAY (30, 0#16)
Y = SQRT(X)
```

The graph Fig. 10.1 is obtained by

```
DISPLAY Y:X
```

or

```
DISPLAY SQRT(X):X
```

The scale is chosen automatically by SIGMA such that the complete curve is displayed, unless !NOSCALE is in operation (see Section 12.1.1).

Define

```
X = ARRAY (51, -PI#PI)
```

Fig. 10.1