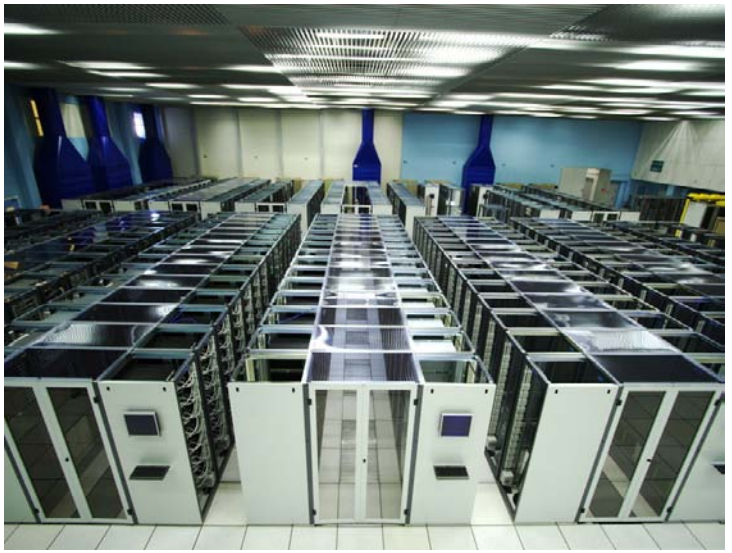
Multicore Panel

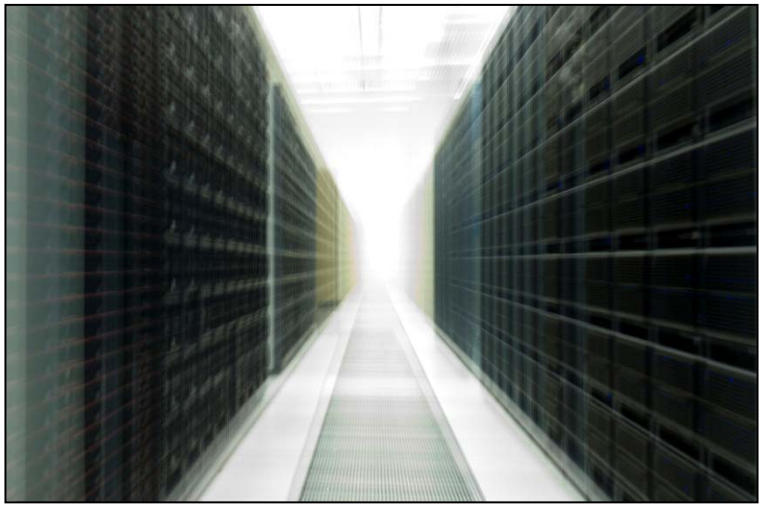
Introduction



Sverre Jarp

CERN openlab

CERN



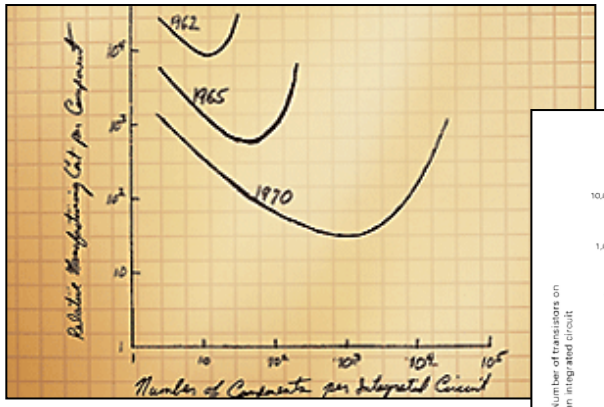
ACAT 2010 – 25 February 2010

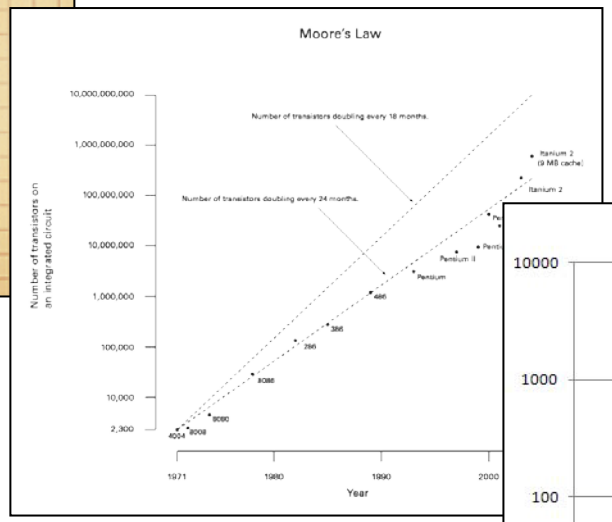
Panelists:

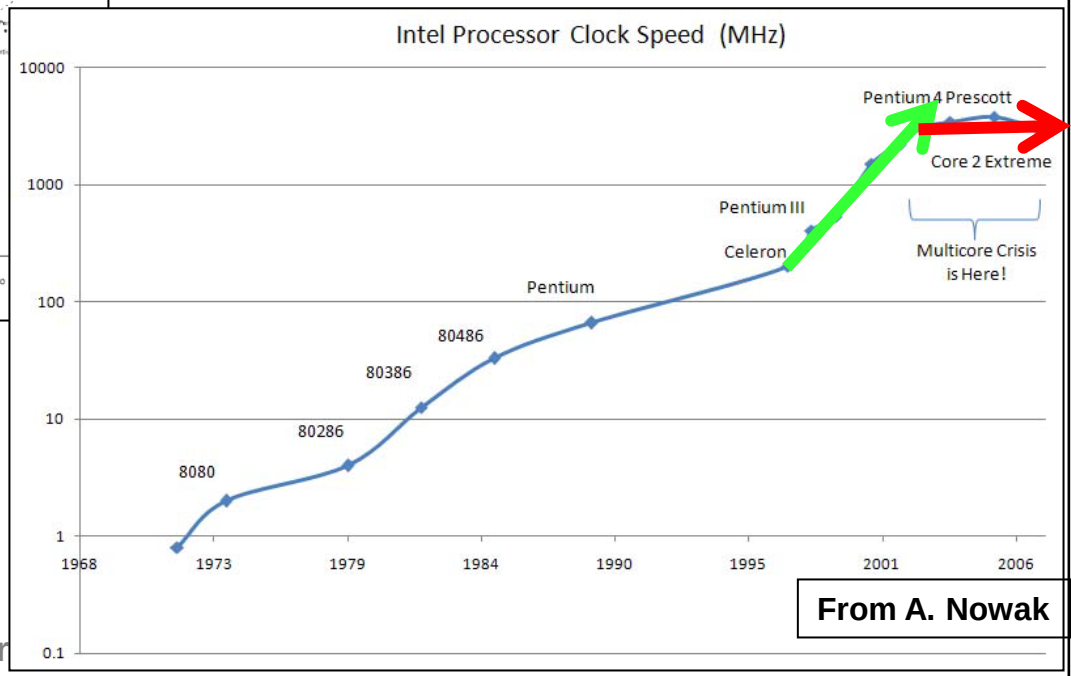
- **Sverre Jarp/CERN: CPU**
- **Mohammad Al-Turany/GSI: GPU**
- **Alfio Lazzaro/CERN: Applications**
- **Mukesh Gangadhar: Vendor tools**

Moore's law

- **We continue to double the number of transistors every other year(*)**
  - The consequence
    - Single core → Multicore → Manycore







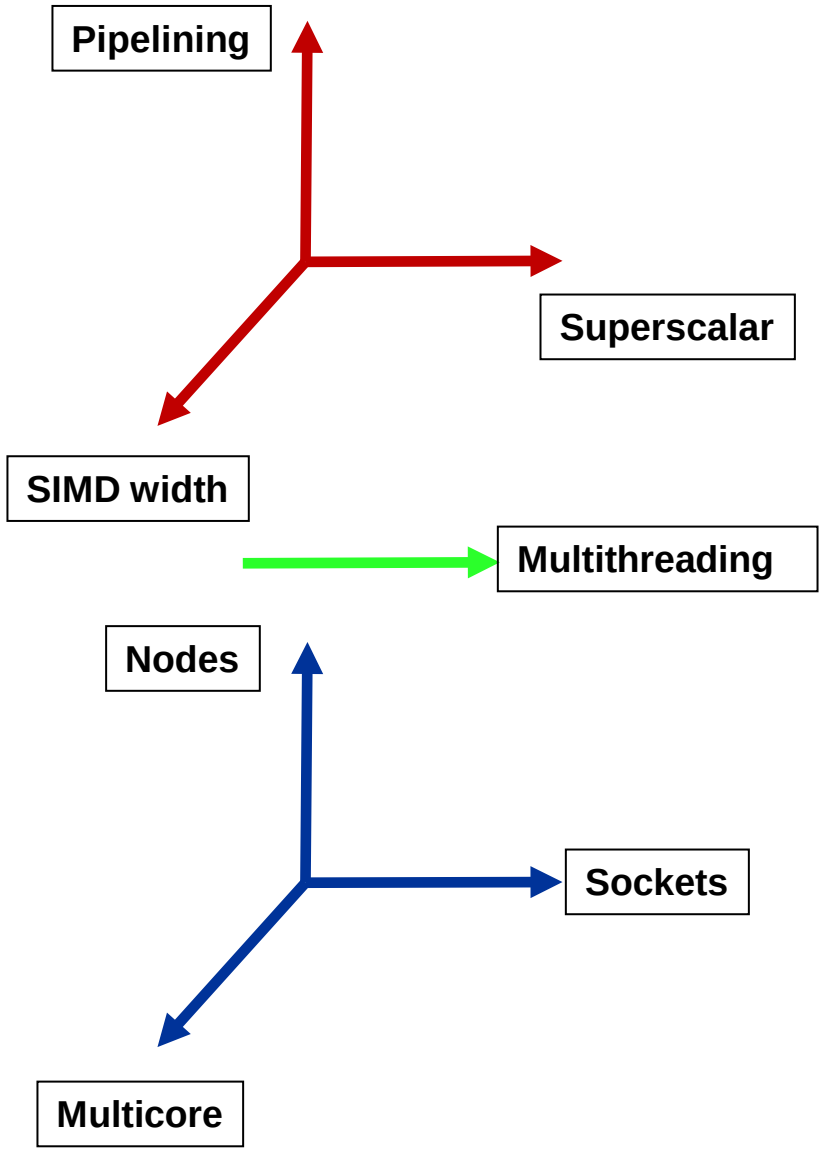
(*) But, the derivative "law" which stated that the frequency would also double is no longer true!

Real consequence of Moore's law

- **We are being "snowed under" by transistors:**
  - More (and more complex) execution units
  - Longer SIMD vectors
  - More and more cores
  - More hardware threading
- **In order to benefit we need to "think parallel"**
  - Data parallelism
  - Task parallelism
- **We also need to think forward scalability**

7 dimensions of CPU performance

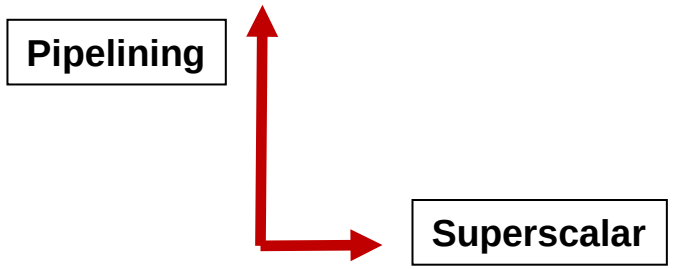
- **First three dimensions:**
  - Superscalar
  - Pipelining
  - Vector/SIMD width
- **Next dimension is a "pseudo" dimension:**
  - Hardware multithreading
- **Last three dimensions:**
  - Multiple cores
  - Multiple sockets
  - Multiple compute nodes



SIMD = Single Instruction Multiple Data

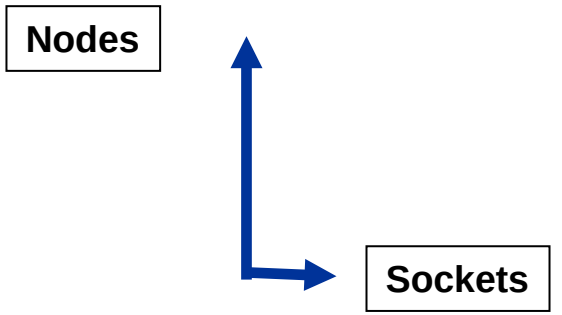
In the days of the Pentium (1995)

- First three dimensions:
  - Superscalar (only two ports)
  - Pipelining (OK)
  - No vectors



- Next dimension is a “pseudo” dimension:
  - No hardware multithreading

- Last three dimensions:
  - No cores
  - Hardly any dual socket systems
  - Multiple compute nodes (OK)

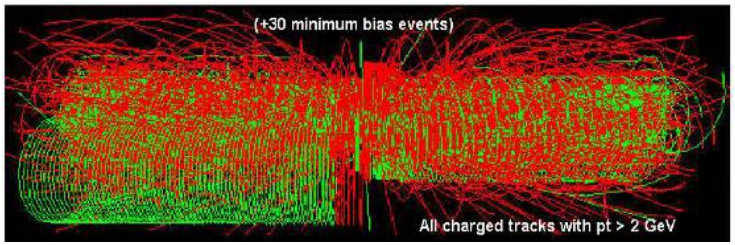


The move to many-core systems

- **Examples of "slots": Sockets * Cores * HW-threads**
  - Basically what you observe in "cat /proc/cpuinfo"
  - Conservative:
    - Dual-socket Intel quad-core Nehalem: 2 * 4 * 2 = 16
    - Quad-socket Intel Dunnington server: 4 * 6 * 1 = 24
  - Aggressive:
    - Quad-socket AMD Magny-Cours (12 core) 4 * 12 * 1 = 48
    - Quad-socket Nehalem-EX "octo-core": 4 * 8 * 2 = 64
    - Quad-socket Sun Niagara (T2+) processors w/8 cores and 8 threads: 4 * 8 * 8 = 256
    - Radeon ATI/AMD GPU (w/Stream Processors): 1600
- **When planning new software: Thousands !!**

Concurrency in HEP

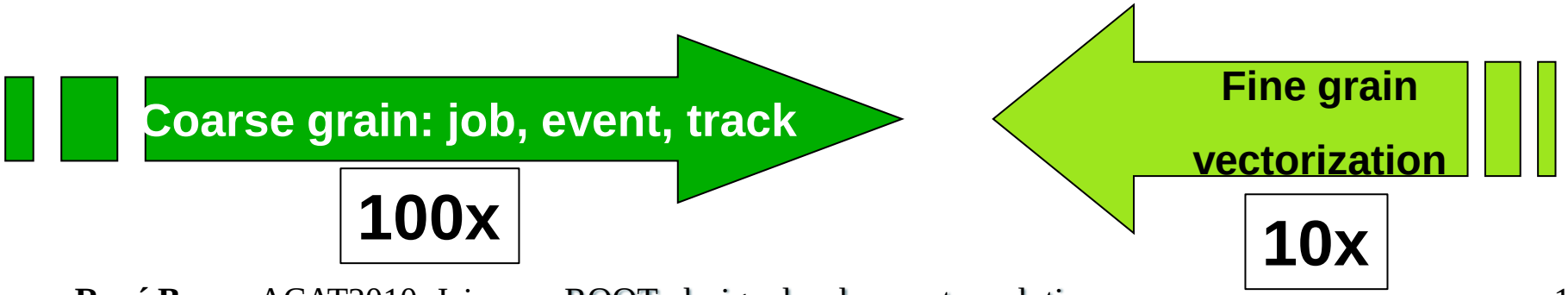
- **We are "blessed" with lots of it:**
  - Entire events
  - Particles, tracks and vertices
  - Physics processes
  - I/O streams (ROOT trees, branches)
  - Buffer handling (also data compaction, etc.)
  - Fitting variables
  - Partial sums, partial histograms
  - and many others …..



- **Usable for both data and task parallelism!**

Design for Parallelism

- **The GRID is a parallel engine. However it is unlikely that you will use the GRID software on your 32-core laptop.**
- **There is a lot of work to be done within ROOT to use parallelism internally or making ROOT-based applications more easily parallelizable.**
- **Think Top→Down and Bottom→Up**



Conclusions

- **Think “parallel”**
  - This may often require new/modified algorithms
- **Think ”forward scalability”**
- **GPUs will help us think “vector”**