Available online at www.sciencedirect.com

ScienceDirect

Nuclear and Particle Physics Proceedings 273–275 (2016) 138–147

www.elsevier.com/locate/nppp

Computing in high-energy physics

Richard P. Mount

SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, California 94025, USA

Abstract

I present a very personalized journey through more than three decades of computing for experimental high-energy physics, pointing out the enduring lessons that I learned. This is followed by a vision of how the computing environment will evolve in the coming ten years and the technical challenges that this will bring. I then address the scale and cost of high-energy physics software and examine the many current and future challenges, particularly those of management, funding and software-lifecycle management. Finally, I describe recent developments aimed at improving the overall coherence of high-energy physics software.

Keywords: Computing; Technology evolution; Software; Software lifecycle

1. A personal view of four decades of computing

The personalized approach ensures that my account is rigidly founded on reality, albeit on a limited reality. It might also be hoped that it contributes readability that offsets the inevitable distortion with respect to wider truth.

1.1. In the beginning

As I was leaving high school (or in the language of the place and time, "grammar school"), already knowing that I would study physics at Oxford University, I accepted the farsighted advice of my Oxford tutor to "find out something about computers". I got a summer job at a computer services start-up in a manufacturing town in Yorkshire. I learned to program fast in bad COBOL, fast and accurate punching on a totally manual 12-key punch, and networking (using Ford van technology).

As an undergraduate, apart from running some least squares fits to laboratory experiment data on a PDP-8, I didn't encounter computers, but as a Cambridge University graduate student I was immediately surrounded by the computing paraphernalia of bubble-chamber physics. I visited CERN and joined in taking hundreds of thousands of pictures of the 2-metre hydrogen bubble chamber which I analysed with the aid of a human trigger (scanners), a PDP-8/S data acquisition system, bicycle-powered networking, and analysis via opportunistic use of the astronomer's IBM 360/44. I learned FORTRAN and OS/360 JCL. I also found out that people (scanners) work better when motivated, and that high-energy physics (HEP) already involved a lot of arcane software.

1.2. The European Muon Collaboration (EMC)

My experiences as a postdoc on the EMC experiment were to shape my view of the role of computing. EMC, with 99 physicists, was the largest collaboration of its day, building and then exploiting a massive and complex detector at the end of a purpose-built 2-km muon beamline at CERN. As a

http://dx.doi.org/10.1016/j.nuclphysbps.2015.09.017

flagship experiment of its day, EMC ran for a large fraction of each year writing raw-data tapes at a rate of up to one tape every ten minutes. The resultant 10,000 tapes per year is numerically comparable to the tape use of an LHC experiment.

In spite of this expected data rate, computing resource planning could be summarised as "we hope we can process the data somewhere." As a result, even first-pass reconstruction was often delayed by over a year, major detector deterioration was not discovered until dangerously late, and physics analysis had to fight large systematic effects. I saw responsible physicists trying for almost two years to ascribe an unexpected physics result [1] to systematics, before finally giving in and publishing.

One of my personal contributions to EMC was to invent the technique for beam normalisation (luminosity measurement). I proposed exploiting the installed beam-muon tracking system in a novel way [2] that required randomly triggering during the beam burst and writing out large numbers of very small events. "Can't do that, you will fill up the tapes with inter-record gaps" was the response of my colleagues. So I implemented a new I/O layer on a number of architectures and developed and operated a 10,000 tape/year data-management system that separated EMC data into physics and trigger streams, including, of course, my own luminosity trigger.

By the time I moved on from EMC, I was incurably convinced that computing resource planning was essential, that data management mattered, and that software quality mattered even more than software efficiency. The latter conviction came from fighting to understand code that had been shoehorned into the CDC 7600's Small Core Memory by heavy re-use of variables within FORTRAN.

In many ways, EMC set the course of my future by offering graphic and often painful demonstrations that software and computing matter in HEP.

1.3. L3 at CERN's LEP acclerator

After EMC, I joined Harvey Newman in planning computing for the L3 experiment and in creating, right down to making cables, the "LEP3NET" US-CERN network that is the ancestor of today's transatlantic component of LHCOPN[1]. Our 1983 computing planning, six years ahead of LEP start-up, estimated CPU that turned out to be low by a factor of 1000 and planned to use disk only for tape staging. However, the cost estimate was approximately correct! Criticism from CERN management that our requests were irresponsibly greedy was taken as some measure of validation, especially as we were not expecting CERN to provide most of the resources.

I learned that, not only in war, *"plans are worthless, but planning is everything"* [3]. I also learned that "what are our requirements" is the wrong question in computing for HEP. The right question is something like "what will affordable technology be able to do for our physics productivity," where technology includes CPU, disk, tape and networks.

2. BaBar

BaBar was close to being the entirety of the SLAC HEP program in 1997 when I joined the laboratory to take charge of computing. I found a laboratory totally committed to the success of BaBar, but struggling to plan for BaBar computing. The 1995 Technical Design Report called for 17,500 MIPS of CPU, 5 TB of disk and planned all data movement between centres to be on tape. Data taking started in 1999 and within two years BaBar had 1,700,000 MIPS, 80 TB of disk, and had moved data around the world using only the network. Fortunately SLAC management had guessed that huge resources would be needed and I only had to confirm and quantify this viewpoint.

While the resource-planning experience served as a good endorsement of the Eisenhower quotation, BaBar also provided an experience from which the lessons are harder to extract. In 1997, with some encouragement from me, BaBar took the bold step of choosing an object database management system, Obectivity/DB, to manage the storage of, and access to, its event data. BaBar's Objectivity database was scaled to close to 1 PB between 1999 and 2003, when it was abandoned with much rejoicing on the part of the BaBar physicists. An analysis of this experience would fill more space than I am allowed, so I will reduce the message to my personal distillation: making strategic software decisions is hard!

I will avoid the temptation to attempt to write wise words about my next adventure, computing for the ATLAS LHC experiement, and move almost immediately to an attempt to look into the future, somewhat informed by past experience.

[1] The LHC Optical Private Network