Sketches of Padova
Impressions from the first LHC Computing Workshop (Padova, 11-13 June 1996)

Manuel Delfino *IFAE, Universitat Autònoma de Barcelona, Spain*

Abstract

The first LHC Computing Workshop was organised recently to explore the problems and possible solutions to the computing of the next generation of experiments at CERN. Although the requirements are quite large, technological evolution is expected to provide the necessary hardware and infrastructure components. The real problems lie in the development and maintenance of crucial, very complex software by thousands of geographically distributed people over the next decade.

Keywords: Particle physics experiments, computing, software development

1 Introduction

The Large Hadron Collider will be the next large accelerator at CERN. Suffice it to say here that the nature of the proton-proton collision physics that will be investigated in the LHC experiments is such that massive amounts of computer power will be needed to extract meaningful answers. Interesting collisions will be buried by factors of at least one to one million in background. With expected event sizes of at least 1 MB, recording all the collisions for subsequent analysis is prohibitive, as the data rates would exceed 1 GB/s creating millions of Terabytes of stored data. Therefore, it is expected that the data will be filtered online, utilising complex event reconstruction software which will absorb millions of Mips of CPU power, in order to recognise potentially interesting collisions and reduce the recording rate to a few Terabytes per year. These datasets will then be analysed further by hundreds of physicists simultaneously. The first LHC Computing Workshop was organised to discuss these problems, and took place at the University of Padova on 11-13 of June, 1996. Clearly, it is impossible to thoroughly summarise three full days of presentations and discussion in one hour, so I chose to give my impressions focusing on the problems for which clear solutions are not yet evident.

2 The Workshop's Programme

These were the titles of the sessions, each lasting several hours:

- Overview by ATLAS, CMS and ALICE (these are the 3 approved LHC experiments)
- Technology trends
- Computing model
- Data Storage
- Worldwide collaboration
- Software Development
- Software Process - part II
- Summary session

And this is my paraphrasing into simple but direct questions that need to be answered:

- LHC physics requires a lot of computing
- What will technology provide us with ?
- How do we arrange the computing ?
- How do we access large sparse data pools ?
- How do thousands of people collaborate ?
- How do we design complex software ?
- How do we construct and maintain software ?

Although this was the first formal workshop on this subject, a huge amount of work has already been done by the experiments in feasibility and R&D studies, and in preparing their letters of intent and proposals. It is not surprising, then, that one of the most interesting discussions took place at the end of the very first session. But to understand why it was so interesting, you will have to read on until you reach the epilogue.

3 Hardware and Infrastructure evolution

When one reads any of the technical proposals of the LHC experiments, one is struck by the fact that it would be impossible to fulfil their computing needs with the technology of a decade ago. Furthermore, although one could employ today's technology to fulfil these needs, it would be unafordable. Therefore, the whole success of these designs is based on the assumption that computing technology will evolve in the next decade, while the detectors are being built, and that an affordable solution will be available when needed. CERN has setup a number of Technology Tracking Teams recently and there were preliminary reports from most of them.

Very briefly, over the next decade we can expect an evolution of the basic computing components as:

- Individual memory chips 4 Gb (500 MB) for $200, which will allow us to put enough memory per node
- Individual processors 20 times more power than today at 1/200 of today's cost/power-unit, which will allow us to afford all the power needed and, just as important, to install and manage it in the form of a few thousand nodes. The latter will be helped by the fact that simple but effective parallel architectures, such as Symmetric Multi-Processing of dozens of nodes, will be common.
- Disk space at prices below $10/GB, or better $10 million/Petabyte, making it possible to store on disk a reasonable subset of data. Less conservative estimates coming from some disk manufacturers would predict an additional cost reduction by a factor of 10, making it possible to keep essentially all data on disk.
- High speed, high aggregate throughput switched networks, such as Asynchronous Transfer Mode, will be available and affordable, making it possible to interconnect many processors and to build the system to extract the information from the individual detecting elements.

Certainly, the expected trends in these components are good news for the experiments. A fourth component, which has traditionally been very important for particle physics experiments, is tape storage. Unfortunately, the physical tape transport involves delicate mechanical parts, and is not expected to evolve as rapidly as the other components. If one is optimistic about disk prices and pessimistic about tape drive prices in the future, then perhaps the tape system will be overshadowed by large, highly reliable disk arrays.

Regardless of where they are stored, users analyse these data looking at it collision by collision, and therefore they should all be individually accessible in a simple way. The field of automated storage management has not provided us with suitable solutions so far, so developments should be followed closely. Also, recent success with storing and retrieving collision data in Object Oriented Database Management systems may change the type of storage management that is needed.

Given the distributed nature of the particle physics community, an essential tool is high quality and performant Wide Area Networks. Evolution of these can be quite unpredictable, as they depend on the complex environment of progressively deregulated telephone providers in various countries and the progressive implantation of global communications. It is hoped that the current, worldwide Internet craze will fuel the market and provide in a decade WAN connections at today's LAN speeds at affordable prices, but this remains to be seen. This uncertainty in WAN costs causes large uncertainties in the design of the overall computing infrastructure for the experiments, for example whether most computing should be concentrated on the CERN site or, on the contrary, whether regional centres should be setup all over the world. In fact, assumptions about future WAN costs may well be responsible for widely different schemes and differing architecture simulation results that were presented at the workshop.

In summary, there is confidence that the experiments will have good quality components available at reasonable prices. This does not mean that all hardware and infrastructure problems are solved, however, and they will have to be smart in assembling and managing these components to do what is needed.

4 The key problems:

I define a key problem as one for which I currently don't see that steps have been taken to provide a high quality, maintainable and affordable solution.

Amazingly, acquiring data from thousands of sources at 1 GB/s or storing Petabytes and accessing sparse datasets are not key problems, since we can foresee solutions based on technological evolution as we have seen above. Not surprisingly, the key problems lie in areas which depend less on technology and more on our human resources. I will concentrate on two of these key problems:

- Designing and building the software we need
- Using our geographically distributed human resources effectively

5 Software development

Each experiment estimates that developing the essential part of their software is a 1000 person-year effort. This figure should impress us, as it is several times the effort needed to develop a sophisticated high-tech operating system, such as the Macintosh OS from the 1980s. Because much of this software will be used to select data online, it must be robust and well understood. Furthermore, the long operational time of the experiments make it essential to provide software that can be understood and maintained by people other than those who wrote it originally. To put it another way, we cannot leave understanding of the software to "reading the code," as has almost always been done so far in particle physics.

In the past, development of software for experiments has often been done by concentrating, both geographically and temporally, a team of people at the host laboratory. This

concentrated effort could be helped by a disproportionately high contribution from the host laboratory itself, either through permanent staff or temporary fellowships. It is unlikely that this will be the case for the LHC experiments, and current estimates show at most 15% of the software effort to be based at the host laboratory, i.e. at CERN. Also, it is unlikely that the base software can be developed in less than 5 years, simply because of the complexity involved. Software development using teams which are much more distributed in space and time, as well as the requirements outlined above, immediately lead us to require a much more rigorous approach to the problem, using Software Engineering techniques. In fact, the engineering approach to software development for these experiments should be on a par with the engineering approaches to build their mechanical and electronic components. A small difficulty, however, is that Software Engineering is a younger field, and its techniques are not as "standardised" as in other engineering fields, and there were several presentations about the various methods, though everyone agrees that Object Oriented modelling is the general approach.

One severe problem is that the Software Engineering field has recently evolved rapidly, at least by human career standards. Most senior people working on or managing the resources for these development are either not familiar with Software Engineering techniques at all, or know rather archaic ones. This can lead to various problems, such as misassignment of resources or inappropriate decisions triggered by vociferous and enthusiastic, but inexperienced, younger staff. This was very well said in a slide from D. Quarrie, when describing the experience of the BaBar experiment at the Stanford PEP-II B factory, and which I have taken the liberty of annotating in italics:

- You will have religious fights
- Get them over and done with early on
- Accept decisions and move on
- Non-involvement is better than active *often uninformed or misinformed* obstruction
- Experience has to be gained early
- Expect -*Insist !*- to throw away early designs & prototypes
- Most of the problems are sociological

Another problem, very acute amongst people working against deadlines for technical proposals who write and use programs (notice I say write programs and not develop software) to get quick results, is the absolute opposition to throw away any of their existing code. After attending the workshop, these member of our community should reflect on the fact that any possible savings that preserving these poor quality codes would bring are vastly offset by the problems created by throwing away the knowledge of their postdocs and graduate students by recording it as completely ununderstandable machine code.

Two breaths of fresh air came to our relief at this point of the workshop. For the first time, several of the experiments presented schedules where a clean start for proper development of the base software is indicated (around the year 1998-2000), hopefully absorbing all the knowledge accumulated up to that point but none of the poorly written code. The second good point is that the GEANT4 project, about which there are several papers in these proceedings, is well on its way to creating the next generation of general-purpose detector simulation software, applying proper engineering techniques. This provides the "existence proof" which is so important in our community.

6 Collaboration at a distance:

The second key problem is that of using our geographically distributed resources effectively. In a similar fashion to the effort on the mechanical and electronic parts of the detectors, it is impossible (and probably unwise) to concentrate all the resources for software development at CERN. Although part of the success of the GEANT4 project is that they seem to have learned from D. Quarrie's statements (reproduced above), getting over their "religious fights" over methodologies early on, etc., a non-negligible part of their progress is due to effective worldwide collaboration.

The problem is not just one of harvesting person-hours wherever they may be located. The concepts behind the needed software are quite complex, and it is the necessary knowledge that needs to be harvested. It is necessary to create the right environment for this, and to realise that literally dozens of people may need to be aware of any given fact. Just as we cannot depend on "reading the code" for understanding the software, we cannot depend on having coffee at the CERN cafeteria to collaborate amongst ourselves. Properly organising how we collaborate, including setting up clear projects and responsibilities, and proper reporting and review paths, is an important part of the much talked about "Software Process." But there are other parts, such as continuing education, easy to use tools (a rapidly evolving field due to the explosion of the Internet in the world) and, most important of all, the will of each individual to make the extra effort needed to participate in worldwide collaboration. I refer the reader to the many excellent presentations on these subjects.

7 Reflections and conclusions

Although the program of the workshop was quite ambitious, there were several topics that were not discussed, but which I consider important. Operating systems, the facilities they provide and their comparative difficulties of setup and maintenance, should be discussed in the future, as one key component of any LHC experiment will be computing systems comprised of thousands of processors. Component software, either of the CORBA or the OLE type, should probably be considered as a tool to break-up very large and complex programs into more manageable pieces. Perhaps the workshop's silence on these subjects was a reflection of the real-time part of the community, which although present was mostly silent.

A different thing I found striking was that, with only 15% of the software development effort expected to come from CERN, 60% of the participants listed CERN as their institute and about 75% of speakers were CERN employees. Those of us from universities and other research laboratories should certainly reflect on this, and examine our own resources and leadership. Of course, CERN itself has a dual role as host laboratory and collaborating institute. It must help to create an environment that allows outsiders to contribute, as well as learn how to do its share of work alongside the rest of the institutes.

In order to construct a path to solve the key problems I outlined above, I would conclude that we need:

- to adjust our spending patterns to match today's needs of HEP computing, e.g. commercial software, software development and collaboration tools, professional software engineers and programmers
- to encourage continuing education in order for physicists to learn enough about software engineering to participate effectively and interact with engineers
- to protect our cumulative knowledge by expressing it in a code-independent manner

8 Epilogue: The problem is in the middle

I mentioned early on that one of the discussions that I found most interesting was after the very first session of the workshop. As the discussion progressed, I noted on the blackboard of the lecture hall the key points. At the end it read:

- Common Solutions: How ? Define.
- 1000 person-year
- Training
- Costs ? CERN vs. Others - U.S. vs. European accounting methods
- Expertise $\leftrightarrow$ 85% persons outside of CERN

Note that nothing whatsoever on this blackboard refers specifically to computing. This is what propelled me into concentrating on development of complex concepts (expressed as software) and collaboration at a distance as the key problems.

I mentioned earlier that many senior people lack knowledge about modern software engineering, and that under the pressure of today's deadlines are reluctant to allocate the time and effort needed for "software development" rather than "code writing". The question is whether this affects negatively the LHC experimental effort. On this subject, I would like to leave here the voice of a young software engineer, temporarily working at CERN. She made the following comment after my presentation: *We keep trying to explain to our bosses that to do a proper job on this complex software, we need an engineering approach and the proper support tools. At high levels of management, and from talks like yours, we hear the same message. But the problem is in the middle.* Food for thought.

Acknowledgements:

I would like to thank J. Harvey and M. Mazzucato for inviting me to summarise the workshop. This work was supported in part by the Spanish Comisión Interministerial de Ciencia y Tecnología and by the CERN School of Computing.