29

ZARAH The Central Computing Facility for the ZEUS Experiment

Lothar A.T. Bauerdick, Oleg Derugin^a Dave Gilkinson,
Matthias Kasemann, Olaf Mańczak

Deutsches Elektronen Synchrotron, Notkestrasse 85, 22603 Hamburg, Germany

In this paper we would like to present the most important technical issues concerning the design and implementation of ZARAH – the central computing facility for physics analysis for the ZEUS experiment. We would like to present to the HEP community our achievements, open problems, and ideas for the future concerning transparent access to hierarchical mass storage, indexed data access methods, data compression, ORACLE based data bookkeeping and data processing in a high performance distributed “batch” environment.

1 Introduction

During summer 1992 the HERA collider at DESY delivered its first luminosity and the ZEUS detector recorded its first data. At that time the ZEUS computing environment consisted of some few dozens of workstations (DEC and SGI), a parallel reconstruction facility running on multiprocessor SGI machines and there was a strong commitment in favor of computing solutions based on inexpensive RISC/UNIX technology. Nevertheless, a large fraction of the physics analysis and software development was done on the IBM 3090 mainframe and on the central DESY VAX cluster. Also the IBM mainframe with its STK tape library acted as our central repository for experimental and Monte-Carlo data.

Over the last three years, the architecture of the ZEUS *offline* computing evolved gradually into the *real* implementation of an *open* and scalable client-server environment with dedicated network-connected servers (storage-servers, powerful computing-servers, etc.), virtually unlimited hierarchical mass-storage and state-of-the-art network infrastructure ¹. This implementation has proven to be truly suitable for efficient physics analysis and processing huge volumes of experimental data at the ZEUS experiment. In this paper we describe the history of this evolution, our motivation and experiences.

2 History

This evolution of the ZEUS *offline* computing happened in three major steps:

- In early 1993 we implemented ZARAH (*Zentrale RechenAnlage für HERA Physik*), our central batch facility for ZEUS physics analysis at DESY. In this first implementation ZARAH consisted of a multiprocessor SGI Power Series 460 machine ($\approx 6 \times 30$ SPECint92) equipped with some 100 GB of fast, directly connected disk storage, a direct connection to an AMPEX tape robot (256 $\times$ 25 GB cartridges) and the possibility to access STK silos connected to the IBM mainframe via dedicated a ULTRANET connection ².

^aOn leave of absence from Moscow State University, Institute of Nuclear Physics

SW9607

- By the end of 1993 we significantly increased the computing resources of ZARAH and the data reconstruction facility to match increasing CPU requirements by upgrading to two SGI Challenge XL SMP machines equipped with 18 R4400 processors ($\approx 18 \times 90$ SPECint92), one machine dedicated to data reconstruction and the other to batch analysis.
- In early 1995 we increased the disk storage capacity of ZARAH to 650 GB to provide fast access to the complete 1994 data sample and we increased our batch computing resources with one more SGI Challenge machine equipped with 16 processors. The disk storage upgrade included a new dedicated file-server (SGI Challenge DM with 4 processors) and all the ZARAH machines were connected with HIPPI links (800 Mbps) to a high performance switch. In addition we now have the possibility to access STK silos with the Open Storage Manager bypassing the IBM mainframe. In addition, we moved the "online" data link from the IBM to the reconstruction machine, effectively eliminating any dependence on the IBM.

Figure 1: The ZARAH environment in 1995

To give a realistic impression about the scale and usage of this environment we would like to present the following numbers:

- Batch computing power of about 4000-5000 SPECint92 (under normal conditions we use altogether 34 R4400 processors but after reconstruction is finished we can use up to 12 additional processors) running up to 70 jobs simultaneously and processing about 700 jobs per day
- Data storage consisting of 650 GB of fast disks, 6 TBytes of directly accessible AMPEX tape storage, and fast access to STK silos. We handle above 1000 tape requests per day
- Sustained transfer rates of data from our main fileserver above 10 MB/s without visible overhead caused by accessing remote data.
- Around 150 physicists every given week using ZARAH for their analysis.

3 Motivation

In 1993 we gained some experience about the advantages and disadvantages of inexpensive workstation based solutions for offline computing vs powerful, but significantly more expensive symmetric-multiprocessor (SMP) machines.

At that time ZEUS physics analysis was done by means of an offline cluster consisting of some two dozens of DECstations 5000 ($20 \times 20 - 24$ SPECint92) and a single- backplane SGI Power Series 460 SMP machine (6×30 SPECint92). DECstations were equipped with some 2-3 GB of disk space each interconnected via Network File System (NFS) while the SGI machine had some 100 GB of fast direct disk storage and a dedicated ULTRANET connection to the IBM-STK tape library where the data was kept.

We found inexpensive distributed workstation solutions very useful for CPU-bound computing (e.g. editing, histogramming, mail), but found workstation based data analysis to be painful and slow. Major bottlenecks were:

- The necessity of shipping large datasets from the remote data repository to workstation disks over the network; with average file sizes of 200 MB, an effective transfer rate of 100 KB/s (an Ethernet backbone was shared by 20 workstations) and a limited number of drives in the STK silos, users could often wait over an hour for a dataset to arrive.
- Datasets stored temporarily on a distributed network filesystem were accessible at similar transfer rates so that a typical analysis job, which running on a 20-25 SPECint92 workstation able to process 7-8 events/s (35 KB per event), was blocked by insufficient network I/O rates.

Meanwhile, analysis done on the SGI machine avoided these bottlenecks and gave the possibility to perform a pass through the whole disk data sample within one night. Given these observations, we decided to separate the functions of workstations and SMP machines. The ZEUS workstations would in future be used for interactive analysis involving editing, histogramming, *background* Monte-Carlo production³ etc., and the SMP machines would be used strictly as batch machines providing fast access to data and efficient use of resources. Below we discuss some of the methods we have developed to reach our present configuration.

4 The ZARAH Batch Environment

For historical reasons, we use the *Network Queuing System* (NQS) as a batch control system. Additionally, we have implemented a job spooling system and a set of simple client- server tools, whereby the user can submit a job to the batch system directly from his workstation, monitor its execution and retrieve results from the batch system to his workstation.

This approach has proven to be very efficient and allows us to minimize interactive usage of the batch platform. Interactive use is not restricted but "*not recommended*". We have learned that it is highly reasonable to give users some freedom concerning usage of the batch system and monitor that this freedom is not misused rather than preemptively restrict *user rights*.

In addition, as part of the batch package, we have invented smart SIGSEGV and SIGBUS *signal handler* routines which can spawn the *dbx* debugger, execute a debugger command file which prints a useful debug report, and finally calls a user exit routine in order to save results. These signal handlers eliminate the need for running long jobs through the debugger and provide useful diagnostics for the user.

5 Data Handling

The problem of data duplication is a well known issue in HEP data handling. Essentially, there are always two *wishes* causing this problem:

- select smaller subsample of tape-data to keep it on *faster* media (e.g. disk) to reduce access time,
- select subsample of “*interesting*” events from a large data volume to avoid sequential processing of all the data.

We aimed to organize the hierarchy of the storage in a proper way, to use fast and randomly accessible *disks* efficiently and to eliminate data duplication. Therefore,

- we created MINI-DST datasets with “*smaller*” events (35 KB vs. 100-150 KB in reconstructed datasets) containing only the most important (but sufficient for most analysis) information to store more data on disks,
- we compressed MINI-DST datasets (on a *per-event* basis) to reduce the amount of disk storage required for *one year’s data sample* to some 400-500 GB,
- and we invented the *ZEUS event directories* i.e. an indexed data access method which provided us with random access to any given event.

6 Data compression

We have investigated various possibilities of efficient data compression. We found that predictive algorithms based on the knowledge of event structure are difficult in implementation and they have to be changed if the event structure changes. Results obtained with the Lempel-Ziv (LZ77) non-predictive dictionary compression algorithm [LZ] (we have used implementation derived from the GNU gzip program) were much better. We achieved an average compression factor (i.e. ratio of compressed size to uncompressed size) of 65% with relatively small penalty in the data processing rate.

7 Tape staging

At ZEUS, large volumes of data are created only by a few privileged users during data reconstruction and reprocessing and most of the users use this data as read-only. The frequency and method of accessing given datasets depends on the type of data they contain. The ZEUS *data* can be classified as:

- raw data – about 2 TBytes a year
- reconstructed data (RDST) – about 2 TBytes a year
- MINI-DST – about 0.5 TBytes a year (compressed)
- Monte-Carlo – sample of several TBytes

The raw data is accessed during reconstruction and reprocessing and then access is rather rare. The RDST datasets are accessed only if analysis requires information which has been stripped and can't be found on MINI-DSTs. The RDST datasets are usually accessed only within the first 1.5 years. Access to MINI-DST datasets can be easily optimized by a permanent disk storage dedicated to the complete last year's sample and small subsamples of selected events (runs) from previous years. We found as well that it is virtually impossible to create this kind of disk *cache* for Monte-Carlo datasets since the amount of randomly accessed data has a size of several TBytes.

In order to make high-volume tape-based data analysis efficient it is essential to guarantee sufficient I/O bandwidth and reasonably short access time. These requirements can be easily translated into the number of simultaneous staging streams (*drives*) and transfer rates per stream. Our experience with accessing IBM-STK silos via ULTRANET (14 streams but low transfer rates) and AMPEX tape robot (transfer rate up to 10 MB/s but only one stream) showed that both these requirements are very important. The OSM system has greatly improved reliability and performance of our tape staging system.

We believe as well that it is crucial for efficient *tape-based* data processing to make the staging process *fault-tolerant* and *transparent* to the user. For this purpose, we are about to implement an additional layer — a *virtual meta-data filesystem*, visible to the user through an NFS-compatible interface, and an *open server*, performing the actual file staging and returning to the user process an opened file handle. The user can access the *virtual meta-data filesystem* via standard Sun Microsystems NFS protocol but returned information is generated by the *meta-data database* server and points either to an existing *dataset* available in a temporary staging pool or to an open server named socket. The *open server* retrieves information from the *meta-data database* about the size and physical location of the dataset, allocates the required disk space, transfers from tape to disk and performs all the necessary error handling. By this method staging is hidden from the user process and looks very much like access to regular disk files (an emulation of the open system call, aware of the open server, is implemented in an additional dynamically linked run-time library). A prototype of this system has been already implemented and we expect to introduce it into full operation soon.

8 Documentation and Databases

Having fast, powerful and efficient computing resources is certainly a necessary condition for physics analysis at ZEUS. However, all this is wasted if no one can use it.

We have found it extremely useful to have good documentation available for both the new and experienced users. New users are provided with working examples of programs and instructions about what to do when their results don't match those documented. Experienced users are kept up to date on new developments and changes and have access to a FAQ about the system, software and data. Our initial hardcopy manual has been superceded by a HTML version which allows us to do "real time" updating, something that is often neglected with text documents.

Information about data is also an important topic. The ZEUS data contain several millions experimental and Monte-Carlo events and consists already of some 70.000 datasets. In order to use and maintain in a consistent way information about relations between runs and datasets, about parameters use for generation of Monte-Carlo datasets etc, we have designed and implemented, using the *ORACLE* RDBMS, some relational databases. The database information is updated automatically during data reconstruction and production of new Monte-Carlo datasets and is available to all the ZEUS physicists with user friendly client-server tools. Our initial experience with *ORACLE* has been quite good and we intend to continue our database development in this direction.

9 Future plans

Aside from ongoing efforts mentioned in the previous sections there is a need (and some ideas how to do it) to implement transparent job *checkpointing*. We believe very much that an ability to suspend any given batch job at any time and later resume the process, possibly on a different machine, is very desirable. *Checkpointing* is one of the most important issues concerning reliable and distributed batch processing.

In addition, we plan to increase our disk space capacity by approximately 100 GB a year in order to keep selected data taken in the former years.

10 Conclusions

The joint efforts of ZEUS and the DESY central computing group ⁴ have built a state-of-the-art facility for batch analysis. Our recipe consists of simple design, an open technology with scalable architecture, some *thinking* about efficient ways of data processing and some effort to put it all together. ZARAH, with virtually unlimited hierarchical mass-storage, powerful computing resources and easy access from the interactive desktop computing facilities on the net, has become a useful and important tool for the efficient physics analysis at the ZEUS experiment.

We believe that our motivation and experience in developing this highly successful system can be useful for experiments of the year 2000 and beyond.

References

1. L.A.T.Bauerdick et al., "The Physics Analysis Environment of the ZEUS Experiment", in *Proceedings of CHEP '95 Conference*, 1995.
2. O. Mańczak, "The Computing Environment for Physics Analysis for ZEUS Experiment at HERA", in *Proceedings of CHEP '94 Conference*, 1994.
3. B. Burow, "Funnel: Towards Comfortable Event Processing", in *Proceedings of CHEP '95 Conference*, 1995.
4. M. Ernst, "Putting all that (HEP-) Data to work..." in *Proceedings of CHEP '95 Conference*, 1995.