UA-1 SOFTWARE and COMPUTING

Experience and Projections

Denis Linglin (LAPP, Annecy, France)

1) Introduction

The UA-1 experiment is installed in the long straight section 5 (LSS5) of the CERN SPS collider. Preparation started early in 1978 (committee approval was in June), the detector was installed in 1981 (although not complete) and by then was able to record its first data.

Five data taking periods occured so far :

run dates	$\int \mathcal{L} dt$ (nb^{-1}) (on tape)	$\sqrt{s}$ (GeV)	Comments
Oct-Dec 81	0.02	540	" jet run"
Oct-Dec 82	18	540	$\mathcal{L}_{max} \simeq 2\text{-}5\ 10^{28}$ " W run "
Apr-Jun 83	108	540	$\mathcal{L}_{max} \simeq 0.5\text{-}2\ 10^{29}$ " Z^0 run " start using 168E's (emulators)
Oct-Dec 84	270	630	$\mathcal{L}_{max} \simeq 1\text{-}3\ 10^{29}$ 168E's reject events
Oct-Dec 85	400	630	$\mathcal{L}_{max} \simeq 2\text{-}5\ 10^{29}$ VME acquisition

By the end of this year, we will have $\simeq 800\ nb^{-1}$ on tape ($\geq 1000\ nb^{-1}$ delivered). The next $\bar{p}p$ period is scheduled for April-May 1986, hopefully with the micro-vertex detector in. Then, $\simeq 2$ years later, we hope to start with ACOL ($\mathcal{L}_{max} \simeq 3\ 10^{30}$) and a new UA1 calorimeter.

As it is well known, UA-1 is presently one of the largest particle physics experiments worldwide, with ≃ 180 physicists from 20 different institutes in Austria (1), Finland (1), France (3), Germany (2), Italy (2), UK (4), the Netherlands (1), Canada (1) and USA (4), plus CERN of course.

The detector looks like many of the present all-purpose detectors installed at colliders. It is formed of a set of complementary detectors surrounding completely the interaction region from a fraction of a degree upwards, except for a few cracks here and there. A particle emerging from a collision will possibly traverse a large central detector ("Image Chambers"), EM Calorimeters (64 "Gondolas", and 2 "Bouchons" divided into 32 petals each), Hadron Calorimeters (so-called C's and I's, because of their shape, and located behind the gondolas and bouchons respectively), more shielding iron, instrumented with Iarocci tubes, and large muon drift chambers (typically 4x6 meters, each chamber consists of 2 sets of 4 planes, 50cm apart, with 15x4cm section drift tubes). Forward detectors complement this central coverage. The detector is weighing ≃ 3000 tons and, if one excludes the forward parts, forms a box of about 12 - 15 meters in any direction.

Contrary to e^+e^- detectors, the UA-1 magnet (7 kG in 80 m^3) is a dipole, to provide good forward momentum analysis.

The detector is installed on a large chariot to shuttle back and forth between the "garage" and the "Xptal area" position. It is followed by a Mobile Electronic Chariot ("MEC"), 3-story high. In the MEC is located the electronics that cannot be installed too far away, in the UA-1 control room, a hundred meters of cable length away.

2) The 1985 RUN

A few key numbers summarize this 3-month data taking period, ending by Xmas :

$\mathcal{L}_{max}$	$\simeq 4\ 10^{29}$
event rate ($\sigma_{inelastic}$ = 50mb)	20 KHz
Rates for $\mathcal{L}_{max}$ before emulators :	
(VME bus) – Max allowed	30 - 40 Hz
– Actual	6 - 8 Hz
Rates after the 6 168E's :	
– "Normal tapes" (30-40% of input)	2 - 3 Hz
– "Special tapes" (3-4%)	≃ 0.25 Hz
Event size – before reduction in MEC	≃ 1.7 Mbyte
– on tape	≃ 120 KBytes
Number of events per raw data tape (depends on trigger type & mixture)	1200-1500

During this $\bar{p}p$ period, the standard daily scenario was more or less the following : Every late afternoon, SPS engineers are delivering a dense shot of antiprotons, after a few adjustments with pilots. This shot is lasting ≃15 - 18 hours and is dumped sometimes in the morning, depending on beam life-time (and hence decreasing luminosity). One can expect from 4 to 15 nb^{-1} per such dense shot, with 30 to 120 normal tapes and 2 to 10 special tapes written overnight.

The overall amount of data collected is written on 3000-4000 tapes (≃ 4-5 10^6 events), including some cosmics, Min Bias events, ... Similar numbers were obtained in 1983 and 1984. They are mainly fixed by the maximum tape speed, given a reasonable dead time.

Given the tapes used for production and analysis, needs have been at the level of ≃ **10000 tapes per year** over the past 3 years.

3) On-line computers

As with many other modern experiments, UA-1 relies on a network of several hundred mini- and micro-computers and every operation can be controlled by tasks activated from terminals. There is in principle no switch to play with, only terminals. Here follows a list of the various computers and microprocessors of the experiment :

A) The general support of UA-1 is based on **2 Norsk Data ND 100/500**, with 2 Mbytes of central memory each, plus **one ND100**. Each 100/500 system is in fact bi-processor, with the 16-bit ND100 (multi-user, multi-task) front-end of the 32-bit ND500 (multi-user).
Overall capacity is about ≃ 4 VAXs.
As a guideline, one can assume hereafter the following rough computing speed ratios (taking the VAX 780 as the reference unit, which corresponds by the way to almost one Mips) :

Computer type	relative speed
VAX 11/780	1
168E emulator	2
ND 100/500	2
IBM 168 or 3032	4
3081E emulator	4
VAX 8600	4
CYBER 175	8
IBM 3081K	20
SIEMENS 7880	20
SIEMENS 7890	40
IBM 3090	40

B) Six **168E emulators** make the level-3 trigger. Each one has 256 KB program memory, 512 KB data memory and runs Fortran programs, previously tested on IBM machines. With the ND500s, they provide altogether the computing capacity of ≃15 VAXs.
The first two **3081E's**, with 2 to 3 MB memory each, have now arrived in the UA-1 control room. Six such units will replace the 168E's in the future, with the additional possibility to run off-line production.

C) 20 (Super) **CAVIARs.** microcomputers are used for equipment test and control. For instance, there is one to control the PM high voltage system, two to control the central detector, one to control the emulators, etc... The CAVIAR is a DACQ-oriented microprocessor, developed at CERN. Based on the Motorola 6800 and a floating point processor AMD9511FP, it has 256 KB RAM, 84 KB EPROM and a fast CAMAC interface. It is programmable in a custom-made BASIC, called Bambi, and is available with a large I/O and histogramming library.

D) More than 20 Macintosh, with the VME interface McVee (developed at CERN), are replacing older "dum" or "intelligent" terminals. They are used for tests, monitoring and software developments.

E) More than 60 VME CPU boards are used to monitor, sample and control the data along the VME data acquisition bus (parallel readout and event builder processors). Each such "CPUA1" consists of a M68010, NS16001FP, 256 KB RAM, 32 KB EPROM, runs Fortran programs and is equivalent to half a VAX. In the future, the VME CPU's and Macintoshs should replace the CAVIARs.

F) Various other types of microprocessors :
- The ≃ **110 ROPs** ("Read-Out Processors") ar used for CD data reduction in the MEC. Each ROP is based on 2 μP, a MC68B00 for the readout electronics control, and a Signetics 8x300 for data reduction and formatting.
- The 7 M68010 and M68020 processors of the second level muon trigger should be soon ready.
- Other μPs, M6800 or equivalent, are also used here and there, like in GPMCs ("General Purpose μP Controllers") or the MUMMs. There is also a μP to drive a voice synthesizer, etc.

G) Graphics : A few kilometers away from the experiment, three 3-D interactive graphics systems **Megatek** (colour & B/W), controlled by a VAX 11/780, complement the list above with the possiblity of on-line complex analysis and scanning.
11 more Megatek systems, driven by IBM, PRIME or VAX, are today installed in various institutions within the collaboration (Birmingham and Rutherford in UK, Annecy, Paris and Saclay in France, UC Riverside, Wisconsin, MIT and Harvard in the US, Rome, Aachen).
Three institutes in addition are using Apollos for graphics applications : Saclay, NIKHEF and Victoria (BC).

In the list above, it seems obvious one cannot evenly add all the Mips and compare for instance the result to the off-line usage. On-line computing power is used only a fraction of the time, roughly between 10 and 30% depending on application. Hence the numbers quoted for on-line "low cost" Mips, expressed for example in VAX equiv. units, correspond to peak power, while off-line "high cost" Mips correspond to actual usage.

4) Off−line Saga

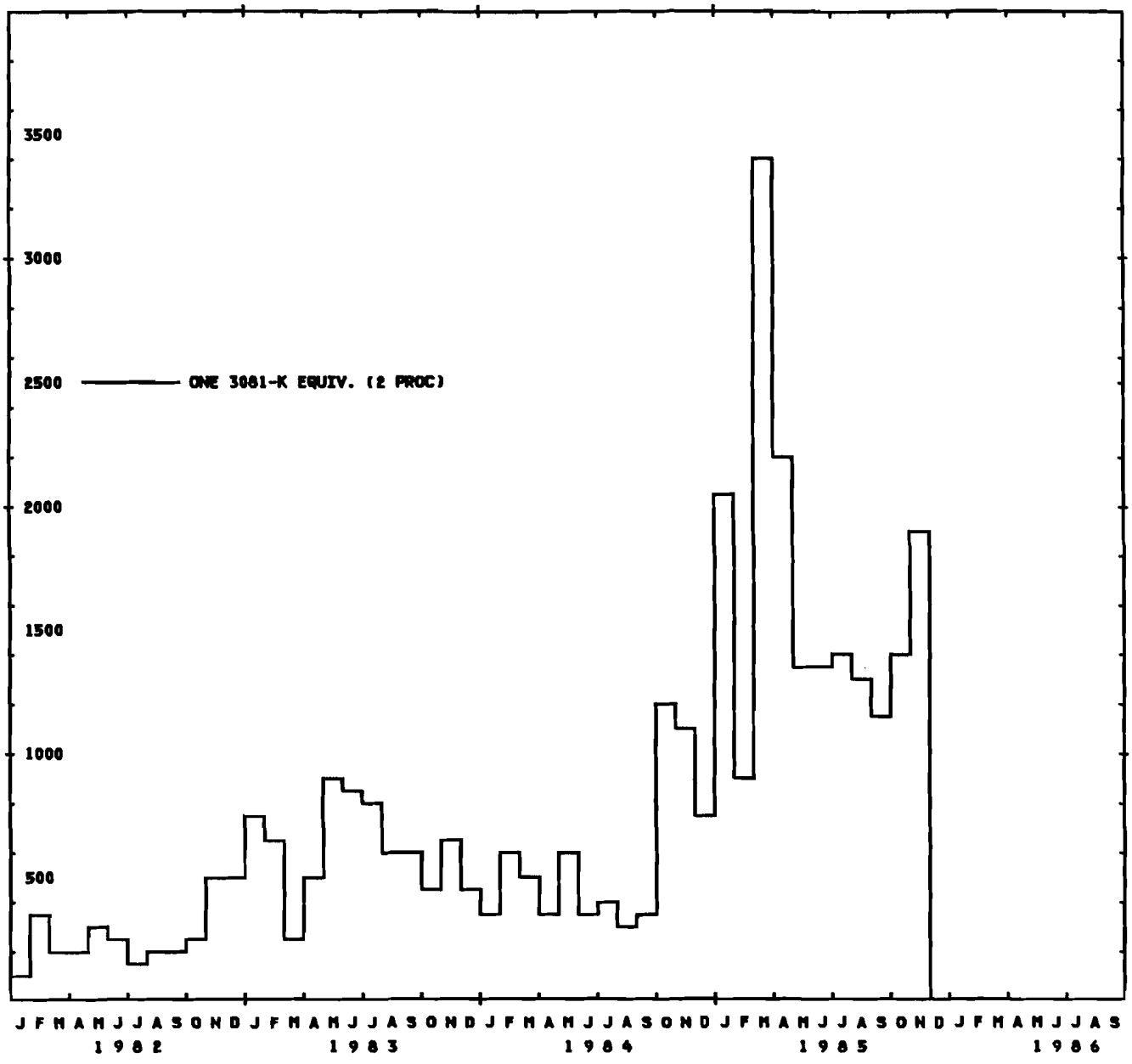
4.1) Overview

When UA-1 software activities started, in early 1978, there were only half-a-dozen full-time physicists involved, then it increased with time, to reach more than 30 in 1981. With the analysis of the data, there are now more than 140 people active on the CERN computer centre every week (and 240 registered today). Difference active-registered has to do with people either working mainly at their home

UA-1

DECEMBER 1ST, 1985
D. LINGLIN

CERN COMPUTER CENTRE (IBM/SIEMENS) : UA-1 USAGE
(CP HOURS PER MONTH - IBM 168 EQUIV.)



APPROX. EQUIV. : 1 SIEMENS 7890 = 2 IBM 3081K = 5 CYBER 175 = 10 IBM/168 = 40 VAX 780
IBM 3090 VAX 8600

computer centre and/or using CERN computers only for documentation, electronic mail, networking, etc...

end of year	active users	registered users
1977	5	10
1979	40	70
1981	60	100
1983	90	150
1985	140	240

At the beginning, the IBM 168 and 3032, with **Wylbur/MVS,** were adequate for all the development work (memory size, Wylbur friendliness, etc..). Later, the replacement of the 3032 with the IBM 3081K (1981), then of the 168 with the Siemens 7880 (1982), was made without any change to the user, except the larger CP capacities needed for production and analysis. The same scenario repeated itself with the switch to a Siemens 7890 (January 85) and to an IBM 3090 (January 86) :

Machine	system	From ⇒ To
IBM 168	Wylbur	1976 – 1982
IBM 3032	Wylbur	1978 – 1981
IBM 3081K	Wylbur → VM	1981 – 1985
Siemens 7880	Wylbur	1982 – 1984
Siemens 7890	Wylbur	1985 –
IBM 3090	VM	1986 –

4.2) UA-1 central computing at CERN

The following **figure** displays the monthly CP consumption of UA-1 on the CERN IBMs (our CYBER usage has always been negligible). I must say that the low level of the load until 1982 was partly due to a tight schedule. No extensive simulation was performed until the first data showed up, and there was no noticeable "spike" over that period. However, a small Monte Carlo production was run on several computers of the Collaboration, to test the chain of reconstruction programs with simulated events. This was done in the first half of 1981, a few months before recording the first real data. **This exercise has proven to play a key role in getting ready on time.**

The "bumps" in our CERN usage, as can be seen on the figure, depend only on two parameters : Run periods (with ≃ one-month delay) and acquisition of new computers (especially the 7890 in January 85). Numbers quoted are for Batch Jobs only. These bumps are of course due to the production and run related activities, they extend three to four months after each run period, before reaching a plateau. Plateau levels correspond to physics analysis and program development, the corresponding load occur essentially during prime shifts : the computer centre peak power needs to be considerably higher in order to cope with the bursts without too much delay.

The group disk space is presently 1.0 Gbyte. In addition, we have ≃ 3 Gbytes of private disks for tape staging (production) and 7.5 Gbytes of mass storage (MSS). More disk and MSS space is permanently welcome. This has to be compared to the ≃ 10000 tapes (1500 Gbytes = 1.5 Terabyte) used to store our yearly data base of raw + processed events.

With almost 1000 jobs submitted on the average per day (and peaks at 1800), computer resources are sometimes hard to manage and may bring the CERN Computer Centre to its limits.

4.3) CERN versus outside Computing

With the tight schedule to build UA-1 (3 years), the priority given by CERN to the $\bar{p}p$ project and, at the time, the limited network connections with outside institutions, it was decided from the early beginning that most of the development work would be made at CERN. One could have developed the software outside more than what was done, without much problem. But, as most of the physicists were physically at CERN anyway for hardware construction, beam tests, installation, on-line developments, etc... the result was that only a small fraction of the code (a few "HYDRA processors") was written outside. In this case no particular problem emerged (some explanations about HYDRA are given in the next sections).

Outside CERN, computer centres have been mainly used for production. More recently, analysis and development are also taking place. The main outside locations so far have been : RAL (IBM 3081K with VM), Saclay (IBM 3081K with MVS), Paris/CCPN (IBM 3081K with VM). The latter centre will move to Lyons next summer and switch by then to an IBM 3090.

The computing ratio CERN/(all UA-1), which was 100% until 1982, has then decreased to almost 50% and is now more like 2/3. It is sensitive to the relative improvements in capacity of the centres involved. Almost all the large computers used so far in UA-1 were IBMs or compatible (Siemens/Fujitsu, Atlas). Small amounts of production were also performed on UNIVAC (Rome).

4.4) UA-1 PUBLIC CODE :

The basic decisions and rules, agreed from the very beginning, were to use :

- **FORTRAN IV.** Fortran 77 was not available on CERN IBMs until 1982 (and was still shaky two years later). But it was available however on many other machines like the VAX, NORDs, UNIVAC, used by the collaboration. Non standard statements were forbidden, except a few well defined cases, like format delimiters, to ensure compatibility between the various computers accessible to the collaboration (IBM, CYBER, UNIVAC, VAX, NORD, 168E). A few further restrictions were imposed in view of the emulators.
  FORTRAN 77 is now the UA-1 standard (from June 1985).

- **PATCHY** for file handling, code maintenance and code transfer (CETA format) from computer to computer. Actually, only a small subset of PATCHY was used, as modern editors make part of it rather out-of-date.
  For PATCHY, the file unit is called a PAM file (typically 5 to 15 K-lines of code in UA-1), divided in PATCHES, each patch being divided in DECKS (each deck corresponding roughly to one subroutine).

- **HYDRA** for memory manager (Banks, tree-like bank structure, reference links, dynamic bank storage, etc...), I/O packages (computer independent file formats), debugging, free format titles, recording facilities, etc... Here again, all the features offered by HYDRA were not used. The concept of **processor**, with defined interfaces between pieces of code, was also very useful.
  Let me mention here that the two CERN-supported systems ZBOOK (for medium-size programs) and HYDRA (for larger programs) will be replaced by a single and more advanced system, called ZEBRA, which is already partially available.

- **HBOOK** as standard histogramming package.

- **SCRIPT** as text processing package (documentation, publications). Later, a good fraction of the work was transferred to dedicated word processors (mainly `Macintosh` today).
  This report, for instance, is using SCRIPT on the CERN IBM, with the APA6670 laser printer.

As the "human cost" to write the software of a large experiment like UA-1 is very high, one has to take great care of making the best choice that can improve productivity. For instance one can use some kind of structured programming, as offered by HYDRA. Also, the "human factor" being so important, it is a dream to hope using ADA, LISP or even PASCAL, as long as more than 95% of the people only know FORTRAN.
As a matter of fact, it has not been completely obvious to decide on HYDRA. This custom-made system on top of FORTRAN was chosen, after some lively discussions, by a nucleus of convinced, motivated people who had used it before. Once started, there was almost no way to keep away from it, for someone who wanted to contribute to the software. Teaching sessions were regularly set up for newcomers.
An attempt was made, however, by some people to have DSTs without HYDRA. It led to too much complication after 3 to 4 years, and we switched smoothly to HYDRA DSTs for the 1984 data processing.

CODE HANDLING

There are presently about 380,000 lines of public code, to hold mainly Fortran programs and detector basic constants (not including on-line). This number has increased steadily with time, as shown in the following table :

year	No of K-lines
1977	0
1979	$\simeq$ 20
1981	$\simeq$ 100
1983	$\simeq$ 200
1985	$\simeq$ 400

As usual (but for how long will HEP experiments continue to do so !?), UA-1 software was started from scratch.

Three main tools / features have played an important role to keep all this code under control :

1) PATCHY – Code is divided, through PATCHY, into $\simeq$ 75 independent pieces ("PAM files"), 6 or 8 at the beginning, split several times. Each one exists at any time under one or several different cycles. The subdivision is made according to several criteria :

- the BCD version of a PAM file must not be larger than the maximum allowed by the editors (typically 15,000 lines for Wylbur).
- Not too many people can work on developing a given PAM, typically not more than 5. Optimum is 2 ± 1 .
- Each PAM must only contain code or information related to a dedicated task, eg. CD reconstruction, calorimeter simulation, muon chamber description titles, etc...
  For example, there are presently 11 PAMs for Central Detector, 8 PAMs for muon detector.

Each PAM is under the responsibility of one person (updates).
A few conventions are used to ease the work of everybody, some examples follow :

- Mnemonics are used for the 3-digit PAM name, and cycle incremental values depend on the importance of the modifications or on similar mods in other PAMs.
- There is an "history" patch at the beginning of each PAM. No change can be made without a short description, a date and who does it.
- PATCHY corrections, like +ADD, +DEL, +REP, ... are forbidden inside a PAM, except temporarily in a special "bugs" patch.

Compiled (binary) libraries are made for every cycle of most of the PAM files. Old useless cycles are discarded.

2) **HYDRA** also plays a key role : it defines common rules of programming and, with the help of a few simple naming conventions, imposes to write "readable" code, a must when more and more often, the code has to be understood or modified by people who did not write it in the first place.
To take only one example, from statements like `LTKD = IQ(LVX-2) ; Q(LTKD+9) = 1/P ;` where one fills the ninth data word of bank TKD, and from the UA-1 bank description (a 20,000-lines book, but each bank description is also available directly at the terminal), one knows it is the fitted momentum^{-1} of one of the central detector tracks, associated with the primary vertex.
Vice-versa, there exists documentation tables to give as well the routine (PAM, Patch, Deck) where a given bank is created and filled.
An attempt to summarize very briefly what is HYDRA can be found in the Appendix.

3) **Wylbur** command files (" **Exec files** ") provide friendly basic facilities for file handling, networking, UA-1 documentation, job submission, electronic mail and all kind of distributed information. This was very important, to spread a minimal information without meeting all the time. It also allowed me to spy on the software activities of UA-1 people and provide more cohesion and coordination to the system.

In addition to the above three technical tools, there was a good starting nucleus of physicists, almost all of them based at CERN.

MAIN UA-1 PROGRAMS :

Basically, one can divide the production programs into few categories :

- SIMULATION. It consists of a chain of 3 to 5 different programs. The full detector simulation of one Monte Carlo event takes about 10−20 sec. CP (IBM-168 units). UA-1 is now using various event generators (ISAJET, EUROJET, COJET, etc ..).

- PREPROCESSING. This single program reads raw data tapes, applies calibration constants, converts data to HYDRA format (Banks) and possibly filters the events through partial reconstruction. It needs 1.0 to 1.5 sec./event.

- RECONSTRUCTION. The so-called **BINGO** program performs partial or full reconstruction of the event, after preprocessing. One typically needs 15 sec. CP / event for a high Pt trigger (half of it for Minimum Bias events). This value is mainly due to Central Detector reconstruction, where time is shared evenly between track finding and track fitting. The high number of points per track and the constant field keep the necessary amount of CP time within reasonable limits. Remember, an average high pt event yields 60-70 charged tracks in the CD, of which about two thirds are connected to the primary vertex.

- CALIBRATION. It consists of many programs, running on NORDs, emulators or IBMs. Only the Central Detector calibration needs sizeable amounts of CP.

- Processing of "Normal tapes". The so-called **BIGMAC** program performs preprocessing and filtering through partial reconstruction. It separates successful events into several streams, according to trigger selections, and then goes on with the full reconstruction. Usually, BIGMAC is run at CERN without the CP-intensive BINGO, filtered tapes are then shipped to outside computer centres for full reconstruction.

- DST and ANALYSIS programs allow data reduction and basic analysis facilities.

- GRAPHICS. These programs are mainly VAX/Megatek oriented (scanning, debugging, physics analysis, etc...).

The memory size needed to run any of these programs is typically in the range 1 to 3 Mbytes. About 4 Mbytes are needed for interactive graphics. The memory manager of HYDRA has helped to keep the above numbers to rather low values. These values scale more or less with the size of the events.

As far as CP sharing between all these programs is concerned, a rough estimate yields :

40 - 50 %	Production (calibration, preprocessing, reconstruction)
30 - 35 %	Physics analysis (including Monte Carlo studies)
20 - 25 %	Program development.

Only production can be done on "low cost" Mips (emulator farms, dedicated CPUs), analysis and development demand "high cost" Mips, as provided by large computers, and now starting, by personal workstations.

TAPES :

The magnetic tapes in use at any level are always written at 6250 bpi density ($\simeq$ 150 Mbytes/tape). Typical maximum numbers of events per tape are the following :

- Raw data	1200	(120 KBytes/event average)
- after Preprocessing	1200	
- after Reconstruction	750	(200 KBytes/event, if preprocessed data are not copied)
- DSTs	5000	(consist of one of the HYDRA structures of the fully reconstructed events)

4.5) OFF-LINE ACTIVITIES DURING DATA TAKING

Over the 15-18 hours of each shot, the related off-line activities, carried out by a set of 20 people (one "Xpress-line" team of two people every 8 hours), were mainly the following :

1) 168E checks : emulator results were checked against the same program / algorithm run on the NORDs or/and the IBMs. A few minutes CP per job, run on the first hundreds events of every 10 normal tapes.

2) Calibration of calorimeters : programs are run before each shot, using 168Es and NORDs. The outcome is a calibration file, stored on both computers. A copy is sent to MSS via CERNET and installed in the UA-1 constant base.

3) Calibration of Central Detector : It is still performed on IBM every one or two shots. It consists of various jobs run on the first 15 normal tapes of the shot. It delivers, after 4 hours CP (IBM-168 equiv.), a calibration file valid for the next one or two shots, that is installed as a separate element of the constant base.

Altogether, the various calibration files generated for a shot had a total size of a few hundreds KBytes, that is, of the order of 50 Mbytes for a 3-month running period. Hence, the size of this constant base is not a major problem.

Only when the above calibration procedure is performed can one proceed with the "Xpress-line" production. Only special ZF tapes were processed at this level (2 to 10 tapes per shot).

4) Preprocessing : calibration constants are applied + etc... This takes slightly over one second per event, hence a few hours CP.

5) Reconstruction (about 12 to 15 sec. per event, IBM-168 units) : 10 - 40 hours per shot.

6) Megatek selection : Higher cuts and CD track matching to select electron and double muon candidates were performed on reconstructed tapes. Cuts were set up such as to yield not too many events for the round-the-clock team of scanning physicists.

OTHER PRODUCTIONS :

Besides " Xpress-line " activities and analysis, other productions were going on as well in parallel, some with a few weeks delay :

- Processing of cosmic data, for calibration purposes.

- Processing of Minimum Bias data for physics, but also for calibration and monitoring.

- Processing of normal tapes (BIGMAC) :
  - calibration, preprocessing, filtering at CERN
  - Reconstruction mainly in outside computing centres.

All these productions were scheduled, not to overload the centre and to avoid competition with Xpress-line. To keep tape drive usage at an affordable level, one uses over 15 tapes of disk space as tape staging.

OVERALL CONSUMPTION :

Altogether, summing up the above changing numbers, $\simeq$ 30 hours per shot were used for Xpress-line, 0-40 hours/day for other productions, and a background of 10-40 hours/day for other activities of development and physics analysis, hence a total of 60 - 80 hours/day on the average, with peaks above 100 hours, over the past 2 months.

The overall consumption can be seen on the computing usage histogram.

4.6) FORECASTS UNTIL 1988

In 1985, UA-1 is using $\simeq$ 20,000 CP hours (IBM-168 equiv.), equivalent to 15 VAXs or so. One can foresee that the same amount will be used for each of the next two years and outside computing will add $\simeq$ 10 VAXs or so.

Let me repeat here that, except for production, most of the computing usage is requested during prime shifts.

If needed, a factor two increase should still be possible, given the capacity of present computer centres available in the Collaboration. But a factor five would be much more difficult, without using a number of dedicated processors (Emulators ..). In 1986, such 3081E emulator farms, for off-line production, will start in Saclay, CERN and Harvard−MIT. Rome is looking for official approval of their farm early in January 86.

Because of the difficulties to export the constant base and large numbers of tapes, preprocessing (+ filter) of the 3000 normal tapes will be made at CERN, as explained above, while most of the reconstruction will be performed outside, mainly at Rutherford, Paris (then Lyons) and Saclay.

The UA-1 trends, for the two years ahead, can be listed as follows :

- Hardware :
  - full VME + 65 CPUs (0.5 VAX each)
  - Six 3081E's (4 VAX each)
  - Magnetic tapes ⇒ optical disks (?)
  - More "intelligent" terminals, à la Macintosh (+ McVee)
  - New generations of graphics workstations.

- Software :
  - No basic change is expected, but the " **upstream move** " will continue. For example, the Central Detector and other calibrations as well as the full preprocessing could go on-line in VME CPUs. This way, we could have events calibrated and preprocessed, then filtered by emulators before being recorded.
  - Off-line production could be made with 3081E's, at CERN and in a few other places.
  - UA-1 could switch from Wylbur to VM on CERN machines. This is not a trivial task.

5) Projections

In spring 1984 was held in Lausanne an LHC (" Juratron ") workshop, where I had to coordinate the working group on data acquisition and data processing. Participants of this group were essentially from UA-1, UA-2 and ISR R-807. This chapter is a summary of the group report and since the situation does not change so rapidly, it can still be considered as an up-to-date extrapolation of these experiments.

Two main trends were found to emerge from present large experiments :

- The so-called " upstream move " will certainly continue in the years ahead, given its many advantages. Already now, many decision tasks, monitoring or calibration tasks, and even partial reconstruction tasks, are performed locally, either near the detector or in the control room (dedicated processors or large minis).
- On the other hand, it is crucial to have an excellent flexibility in order to get ready for many scenarios, whether it be the existence of surprising event topologies, the demand of increasing luminosity, etc... For example, triggers should be flexible enough to set up for any combination of " elementary " constituents (quarks or gluons → jets, leptons, photons). Also, the CPU capacity of the high-level triggers must be flexible, to cope with increasing luminosities and/or level-2 trigger rates.

Let us take as granted the canonical numbers usually quoted of a 1 KHz maximum rate and a 1 MByte average event size at the exit of level-2 trigger. The following proposal, which solves these trends reasonably well, is based on three main ideas :

– A high speed Data Acquisition Bus,

between the detector area and the control room, typically over a distance of 50 to 100 meters. This bus also links level-2 and level-3 triggers. It should be able to sustain a rate of 1 Gbyte/s (1KHz ⊗ 1Mbyte). Presumably this will only be feasible with several (between, say, 10 and 25) parallel branches and possibly with optical fibres.
For comparison, VME can reach ≃10 Mbytes/s and FASTBUS can theoretically reach a maximum transfer rate of ≃ 20 to 40 Mbytes/s over short distances. The one-Gbyte/s quoted above represents 25 parallel branches with a speed not so different from today's maximum value for Fastbus, apart from the larger distance involved and N-branch coordination problems.
Although the goal of 1 Gbyte/s does not seem unrealistic, we feel that Research and Development will be desired in this field.

– A single level-3 trigger system with very large CPU capacity,

up to 1000 processors, each one equivalent in speed and memory to a present large mainframe.
The event information is still in N separate pieces when it arrives in the control room at the end of the DACQ bus. Only here, at level-3, does a single processor have access to the full event information, ready for recording.
It is proposed to install at this stage a "stack" made of a large (50 → 1000) number of processor units (as for the 3081E emulator of today), as shown on the following figure. Each unit of this stack has a typical CPU speed of at least 10 Mips or Mflops and 16 Mbytes of central memory. This is roughly the speed and memory sizes of the large computers we are using today in our computer centres. We assume that the computer industry will be able to deliver such processors in a volume equivalent to one (or a few) CAMAC or VME crate slots, at a typical price of 1,000$. This means one rack could hold 5 crates with 5 to 20 processors each, plus its (optical disk) recording unit. A 1000-processor system would then be accomodated in 10 to 40 racks, at a typical price of one Mega-$ and a peak power of 10 Kilo-VAXs.

Each incoming event selects the first unit available and, depending upon the bits set by lower level triggers, starts one of the fast filter programs. This can be, for instance, a refined level-2 trigger, with the final calibration constants, or an elaborate jet finding algorithm, with for example an improved Et cut or a multi-jet effective mass selection. If the event passes the test, one starts a second, more elaborate, selection program. Thanks to CPU power, this can even be the reconstruction of tracks from the central detector for additional rejection power.
Other selection programs may follow, increasingly elaborate as the remaining events decrease, with consequently more CPU time available per event.
Possibly, selected events can be fully reconstructed before being recorded.
With enough memory, each unit can hold all the filter and reconstruction programs and play the role of "several - in - one" high level triggers.

Moreover, the scheme allows :
- a flexible number of microprocessor units, to match increasing luminosity, level-2 rates or decreasing costs per unit,
- an easy implementation of new algorithms (the development of which depends mainly on the off-line analysis of previous data). It is very important that algorithms run with the most up-to-date information and calibration constants.

– Data storage and processing mainly at the experiment.

RECORDING : The best choice foreseen as a recording medium seems to be the optical disk (although magtapes may have not yet given their last word). 12" optical disks are now arriving on the market, with reasonable prices, although one must admit that none has been delivered yet to customers. Advertisements already propose 2-Gbytes capacity disks (1 Gbyte per side), that is more than ten 6250bpi tapes, with typical writing speeds of 0.4 Mbyte/s, only limited by laser power. Capacity and recording speeds should increase, and prices should decrease, in the next few years. With rather cheap disk systems available in 10 years from now, one can choose between a good "juke-box", or a disk pack system, or 5-20 independent disk drives as shown on the next figure.

Although it would be possible to record rates as high as 10-50 Hz (eg. with 20 independent disk drives), we feel that one should aim at a standard rate of $\simeq 1$ Hz, with peak values of $\simeq 5$ Hz. Otherwise, the high level trigger programs in the stack would not be fully efficient.
For an average 1 Hz trigger rate of events, with a typical 1 Mbyte length, we would need to change a 1 Gbyte side of a disk every 15-20'.

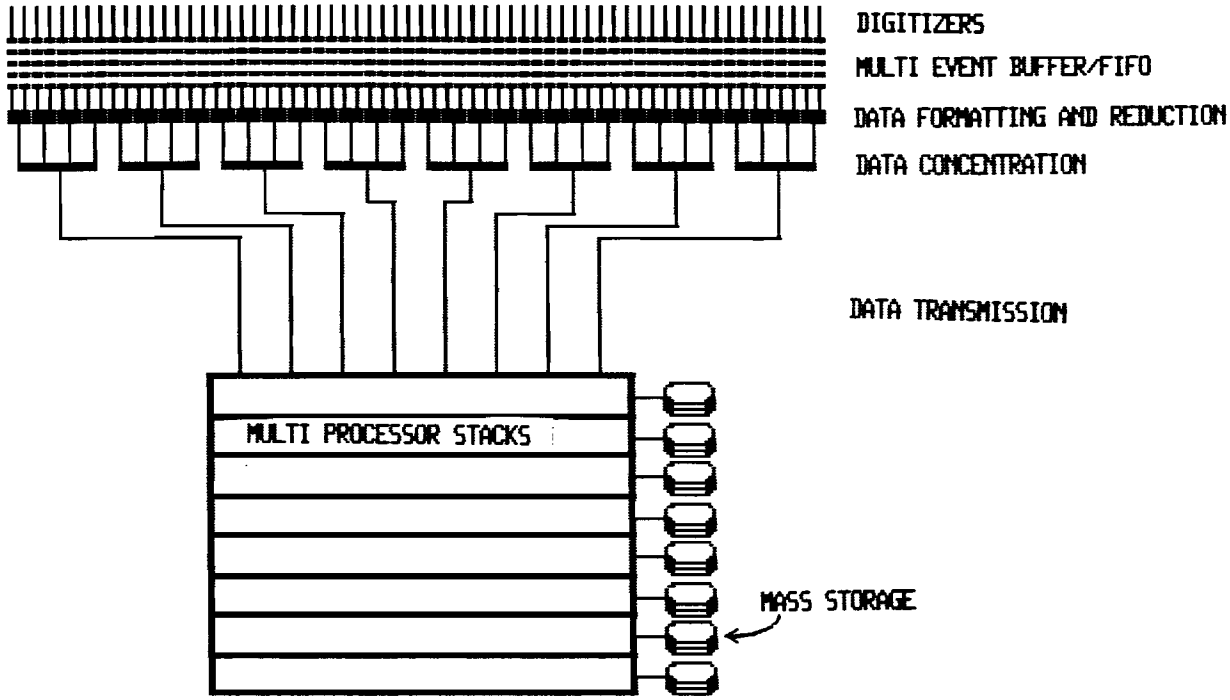
DATA PROCESSING : Since the full experimental data base is available in the control room, and having in addition a large CPU capacity there, the reconstruction and data reduction tasks should be made with the multiprocessor stack. This can be performed off-line or even on-line when one has enough confidence in the programs. This would ease all the administrative aspects such as bookkeeping, calibration constant base, etc.., and also the present tape handling bottleneck, which presently afflict large experiments. Also, the random access facility of the optical disks must ease many of the data processing activities.

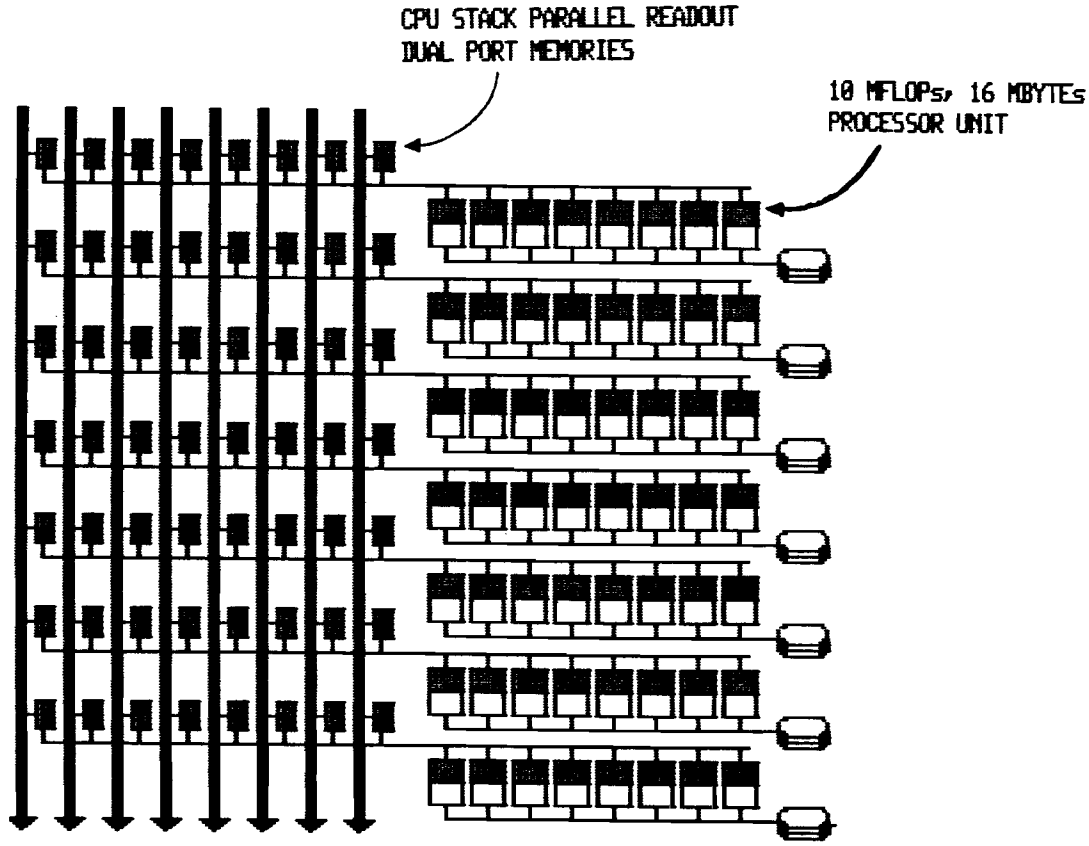
With the large filtering power of the stack, every recorded event should be interesting enough for further analysis. Each 1000-hours period of useful data taking should then yield, for 1 Hz average recording speed, 3.6 10^6 events of 1 Mbyte each. With the processing, one would typically need $\simeq$ 10 Tbytes, stored for instance on a thousand 10-Gbyte disks. Assuming 20 to 50" CPU time to fully process an event in any given processor of the stack, this means between 20 and 1000 hours for 50→1000 processor units. Such computing power allows for several complete processings of the same events, whenever wanted.

ANALYSIS and analysis development should be done on private workstations or on large mainframes because of the niceties which are not available with the multiprocessor stack.
To provide data information to any external laboratory, disks can be copied and shipped. Individual events can also flow from the control room through inter-computer networks.
However, for bulk analysis, the best scheme would be to connect private workstations to the on-line computer ("supervisor") and from there, use the stack and the data base.

6) Conclusion

This short paper is an attempt to summarize the computer related features of UA-1 that can be of interest to any future large experiment. Although the computer world is moving very rapidly and situations are not the same, one should emphasize a few points not to be forgotten : they are called networks, graphics, centralized Mass storage, large memory and CPU capacities, ... and last but not least, a good set of well organized, motivated people.





MULTI PROCESSOR STACKS

Appendix : What is HYDRA ?

Introduction

The HYDRA system primarily is a set of service routines to give the FORTRAN programmer facilities for managing data-structures in a dynamic store located in the blank common. This set of routines is called the MQ package, or **Memory manager.** Basic HYDRA consists of this package, plus a series of data-structure utility routines, plus the exception-handling features needed to make it a fully operational package.
Given the existence of the MQ package and hence of data-structures in a defined format called banks, further general packages have been written which rely on the dynamic store and the services of the MQ package. A number of such packages for common problems are provided, like input/output of data-structures, debug-aids, histogramming, program-flow statistics, processor handling, and others.
The name "HYDRA system" covers the basic MQ package and also these second level packages. Some of these are vital in a full data-processing environment to the extent that the user would have to reinvent them in one form or another if they did not exist. This is particularly true for input/output and debugging. Yet these packages are not integrated parts of the system, they are strictly independent of each other.
All HYDRA system packages rely on the KERNLIB subset of the CERN Program library.

The MQ package (Memory manager)

The basic unit of the data structure is a thing called a " **BANK** ". All banks are held in one FORTRAN common array Q called the "dynamic store". Each bank consists of an identifier, data information and link information. If the bank ID is stored at Q(L), L is the bank address, data start at Q(L+1) upwards whereas link information starts at IQ(L-1) backwards (EQUIVALENCE (Q,IQ)). For example you may have a bank carrying the parameters of a particular track, a bank describing the properties of a particular detector module, or a bank carrying all the parameters and counters of an histogram, etc...

The simplest structure to be built from these units is the " linear structure ", a collection of several banks of the same kind, for example all the tracks of a $\bar{p}p$ collision or all the similar cells of a calorimeter. To materialize this grouping of like banks, each such bank has a reserved word (IQ(L-1) in this case), where is stored the pointer (a **link**) to the next bank. This pointer is the address in Q of the next bank ID.
For many types of information, this simplest structure may be enough. But for many other cases, one needs a more sophisticated structure, allowing relations between banks of different kinds. The fundamental relation " dependence of existence " is defined in the data-structure by a " **structure link** ", and supported throughout the HYDRA system routines.

In addition, HYDRA allows for pointers from any bank to any other bank to be handled by the user as he needs to. They are called **reference links**, in opposition to the two kinds of structure links described above ("next of the same type" and "dependent in existence") and used to interconnect the banks in a data-structure.

With a description of all possible banks in a given program, that is the collection of data plus all their logical inter-relations, you always know which piece of data a piece of code is dealing with and in a few lines, you have access to any information in any of your structures, as shown in the example of section 4.4 :

LFE	=	IQ (LEVT − 7)	Final Event structure address is at −7 in bank EVT
LVX	=	IQ (LFE − 2)	Primary vertex bank address is at −2 in bank FE
LTKD	=	IQ (LVX − 2)	Track bank address is at −2 in bank VX
Q(LTKD + 9)	=	1/P	data word 9 is filled with track momentum^{-1}

Many utilities are of course supplied to help the user : lifting and dropping of banks, garbage collection, etc...

The JQ package (Processors)

The MQ package helps the user in organising his data. The purpose of the JQ package is to assist him in structuring his program. It allows to formalize the concept of "program module" or **processor** beyond the mere FORTRAN subroutine, and provides the back-up service for these modules. A program becomes then a collection of processors. The art consists in designing processors with interfaces as simple and logical as possible, and entirely documentable. In practice, a processor usually consists of a few subroutines (one of them calling the others) which perform a well defined task, like event track finding, outer muon to central detector matching, etc.... It is controlled essentially by a parameter list passed in a special "call bank", containing reference links plus data words and used for both input and output. A processor may call other processors, although the fewer levels one can do with, the better of course.

The JQ package includes as well for each processor separately, the idea and the associated handling of :
- Processor constants, initialized by default within the processor and that can be over-ruled by the calling code, if wanted.
- Processor conditions : How many times each user-defined condition occurs ?
- **working space** : a local area in Q that you can define and use temporarily all along a given processor and is released at the end. This includes facilities to have both data and relocatable links in the working space, and to save / restore working space of a processor calling another processor.
- list of processor flags (eg. to define the amount of debug you want in a given processor).
- statistics of processor usage, like number of times entered and CP time spent.

I know, from experience, that processors written independently (eg. in outside labs) usually had no problem to be plugged in an existing program, once its task and its call-bank were defined.

Other HYDRA packages

1) I/O transportability (FQ package) : HYDRA handles reading and writing of data-structures in a format acceptable by any computer that supports HYDRA. This means that at any stage of a program on machine A, one can write the results (CALL XXXOUT ...), reread the tape or the file on machine B (CALL XXXIN ...), and continue as if on machine A.

2) Debugging facilities (DQ package), in particular dumps of the dynamic store.

3) Recording and trapping of program conditions (RQ package). Trapping allows to jump in principle from any point of the code to some predefined areas, independently of the FORTRAN statements CALL or RETURN.

4) Title handling (TQ package) : input data, parameters or constants can be read in a free-format form and be later addressed as "Title banks".

5) Program flow monitoring (SQ package).

6) Simple histogramming (HQ package). For a more elaborate facility, people use the HBOOK histogramming package, based on the ZBOOK memory manager.

Etc... The HYDRA system is able to sustain structures with a few hundreds to a few thousands banks at a time, with a negligible or small overhead (but the question of overhead, often mentioned, is a false problem : with any program of such complexity you need a data and program management system or your productivity will be degraded).

In short, HYDRA has been up to now a good answer to the weakness of FORTRAN, which has no high-level concepts in data organisation. In addition, it provides facilities that do not exist off-the-shelf in another language and would then need to be developed should a starting experiment decide on another language.

The new memory manager **ZEBRA** has more capabilities and is more suited to present computers with large real and virtual memory, but the general philosophy remains the same.