RZ 2887 (#90835) 12/02/96
Computer Science / Mathematics 47 pages

Research Report

Efficient Support Structures for Distributed Computing

E.Rothauser*, J. Gustafsson, R.F. Hauser, A. Meier, M. Niksch, M. Rappoport, A. Tanner and A. Wespi

IBM Research Division
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

contact:

Dr. E. Rothauser

Höflistraße
CH-8864 Reichenburg Sz
Switzerland

Tel: +41(55)444 12 42
Fax: +41(55)464 11 44
Mail: e.rothauser@spectraweb.ch

contact:

Ing. B. Lassy

A-3464 Pettendorf 78
Austria

Tel: +43(699)10056327
Fax: +49(89)2443-14677
Mail: bernhard.lassy@chello.at
http://members.chello.at/bernhard.lassy

*E. Rothauser managed the Distributed Computing Support project at IBM Zurich Research Laboratory from 1st start in 1990 until his early retirement in April 1995.

Research Division
Almaden ● T.J. Watson ● Tokyo ● Zurich

Efficient Support Structures for Distributed Computing

E. Rothauser[1], J. Gustafsson, R.F. Hauser, A. Meier, M. Niksch, M. Rappoport, A. Tanner and A. Wespi

IBM Research Division, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland

Abstract

This paper summarizes the scope, strategies, approaches, and some of the results and experiences of providing distributed computing support to the various users at the IBM Zurich Research Laboratory from 1990 to early 1995. Given the wide spectrum of conceivable support strategies from the "classical glasshouse" to "full freedom for every individual user," we selected together with management and user representatives a strategy that was to minimize total costs of distributed computing for the laboratory while achieving excellent user satisfaction. In abstract terms our strategy was based on "Occam's razor," which states that "entities must not be multiplied beyond necessity". This means that we should make all aspects of the total distributed system with its hundreds of workstations as simple as possible and avoid needless duplications. This also applies to all related tasks from system planning over system administration, hot-line support and user involvement to accounting for and recycling of resources. Specifically, this strategy requires that all key services to be provided on a single operating system platform that is sufficiently open to meet the heterogeneous needs of the users in a research laboratory. Only a multi-user, multi-tasking operating system can meet all these requirements. Hence, AIX was selected to carry all our backbone services. This does not preclude the usage of PCs to meet special user needs. Given a mix of users ranging from experienced programmers to occasional administrative users, we have established a nearly self-explanatory, application-oriented graphical AIX user interface which provides sufficient freedom to experienced users but hides all intricacies of the underlying operating system from naive users. This allowed us to confirm that there is no truth in the statement that only computer scientists can happily work in an *IX environment. Depending on their needs, we have converted some of the administrative users from their old PC environments to AIX-based applications. Their learning time was minimal, and they have been very satisfied, especially regarding the ease of teamwork without impacting the high level of data security they need. Our experience shows that, for large environments, distributed computing with centralized control does indeed achieve an overall optimal cost/performance ratio. In contrast to general expectations, practically all our *IX (AIX) users as well as a large proportion of our PC users are very satisfied with such a centralized environment, not least because it also allowed us to offer very effective hot-line help facilities.

[1] E. Rothauser managed the Distributed Computing Support project at the IBM Zurich Research Laboratory from its start in 1990 until his early retirement in April 1995. His present e-mail address is: e.rotbauser@spectraweb.ch .

1 History and Environment

In 1990, we assigned two people to introduce AIX to an environment of host users (VM & MVS) and PC users (DOS, Windows & OS/2) at the IBM Zurich Research Laboratory (ZRL). This group, called Distributed Computing Support (DCS), had grown to around a dozen people serving the entire laboratory by the end of 1994. Over time, local host support has been drastically reduced to selected core VM services, whereas local MVS support was terminated in mid-1993.

AIX has now become the strategic backbone system with a large community of direct AIX users. PC users reach AIX-based core services via X-windows support.

The ZRL user community consists of about 300 people including research staff members, visiting scientists, temporaries, and all others. This results in a wide spectrum of users from

- "naive" users,
- administrative users,
- computer scientists, to
- "special" users, e.g. those who do numerically intensive computing, and/or require advanced visualization tools or design systems.

All users need

- office systems support including printing,
- e-mail, and . general information services.

User expectations concerning the desired level of support vary from

- full support for a user-specific environment to
- freedom to try things out, but, if something goes wrong, excellent support for recovery.

At the end of 1994, there were about 300 PCs, 100 Xstations, 160 RISC Systems/6000, and a 1 6-node SP-2, plus a 2-processor VM host at ZRL. Workstations are distributed fairly evenly between ZRL's two sites, which are about 5 km apart and interconnected by an 8 Mb/s link. Some 80 PCs were installed as home terminals.

2 Basic Dilemmas of Providing Computing Support

Studying many existing computer clusters and their support structures it becomes clear that there is no single optimal set of goals and strategies to govern a computing support effort:

- Experience with the fully centralized "glass house" approach has confirmed its superiority for host environments.

- Sophisticated UNIX programmers often work successfully in large, fully decentralized environments.

These opposing positions do not seem to allow for a reasonable compromise. The issue is further complicated

- by naive managers who have found that many large host programming efforts have become unnecessary with the introduction of PCs and their suites of commercial application program packages. Some of these managers tend to extrapolate linearly and, given their past experience, hope that computing support costs can be reduced to almost zero simply by giving everybody an inexpensive PC and by cutting support staff, and/or

- by naive users who have their own PC at home and who feel they have sufficient expertise to administer their own PC at the office.

The present paper outlines the approach taken until early 1995 to optimize the computing cost/performance ratio at ZRL. The decision to take this approach was reached together with management and a committee of user representatives. There were several other valid approaches under consideration, such as assisting in IBM software development efforts by testing specific pre-release software packages by practical use, or by encouraging users to resolve general computing problems by themselves, hoping that this would entice these users to learn more about working with computers. Obviously, the decision to take a specific approach to computing support had to be made at the beginning of our project, but it will have to be reconsidered and revised whenever general conditions and requirements change.

3 Key Goals and Strategies

3.1 Key Goal

The key goal of the DCS project was to establish and maintain a distributed computing environment for the entire laboratory, and to achieve excellent user productivity and user satisfaction at minimum costs.

This goal required further specifications before the correct strategy could be defined. We assumed that

- "minimum costs" include the efforts of users to establish and maintain their working environment,
- a standard set of application programs would emerge and slowly change over time depending on user requirements, rather than being rigidly predefined from the beginning, and
- standard I BM hardware including related system support products should be used wherever practical.

The first two specifications had a crucial effect on our selection of a general strategy out of a wide range of conceivable options, whereas the third specification is fairly obvious for an IBM laboratory.

3.2 General Strategy

Our general strategy was based on "Occam's razor", which states that "Entities must not be multiplied beyond necessity".

Given the technical need for a few hundred workplaces with information system support, we should then make all aspects of the total system cluster, its parts, and the related support operations as simple and standard as possible (but not simpler...), and avoid needless duplications. This includes all related tasks, such as hardware and software planning, structuring the actual system, performing installations, administration, and maintenance, conducting all user involvement's (instruction, working, system tailoring), maintaining a help desk, accounting, and recycling resources. "Occam's razor" should also be applied to limit the number of experts who understand and can modify our environment.

3.3 Full User Support with Computing Services

Users receive from DCS a directly usable set of computing services, including the necessary hard-ware and software components, as well as assistance with these operational tools. They are not simply offered a collection of hardware and software packages with little further support.

4 Derived Concepts and Strategies

The above three key concepts can be broken down into a number of sub concepts including the following.

4.1 General Concepts

1. We attempt to establish a stable "open systems environment", even if desirable building blocks are not yet available in product form, and "open" even if our system will be composed mostly of IBM components.

2. All users should receive computing support for all general tasks. Users should be able to concentrate on their own work, and should not be burdened with general support tasks, such as hardware and software planning and acquisition, . workstation installation and setup, design and maintenance of the general user computing environment including networking, designing programming/support structures for work-intensive, repetitive tasks, such as letter writing, or team-based activities, data backup and archiving, recycling and redeployment of computing resources, or . installation of SW used by more than one user.

3. Standardization of components in our ZRL distributed computing environment is limited by user acceptance and user needs, especially in the case of cooperation with outside research groups. However, there will be a growing need to put business records and work reports into suitable unified data banks for record keeping and information retrieval.

4. Users should be able to obtain special hardware and software as needed for their work. Justifications and funds for such acquisitions have to be provided by the user's project or department.

5. Users should be permitted to log in at any workstation, and find, as far as possible, their own computing environment. This includes a largely standardized graphical user interface (Single System Image SSI) providing access to nearly all services a user may need without ever having to resort to the AIX command line.

6. Users should access applications via some graphical user interface without having to deal with the underlying OS, or its command line interface. There should be no need for users to understand how to modify their system environment.

7. Help Desk services must be so good and responsive, that even "hackers" rely on the majority of the centrally supported standard services, rather than setting up their own solutions. This requires that the overall structure of the computing environment be kept simple and manageable as seen from the help desk, and problem diagnosis and correction can usually be made on the spot by DCS system experts without leaving their "help desk".

8. Appropriate levels of data security & privacy must be ensured. Even internal "hackers" should not be able to gain unauthorized access to protected files or systems.

9. Many users occasionally need very large amounts of computing power. Idle workstations utilized as temporary servers as well as special compute servers can satisfy such needs. The same concept can be reformulated to state that most users need only personal resources to provide appropriate "presentation services", or that networked personal workstations need not meet individually the peak computing requirements of their current users.

10. As part of a research laboratory we should to some extent understand how also our solutions could be applied to the needs of IBM customers.

4.2 Networking Concepts

1. Networking is based on TCP/IP. Proprietary protocols would not be appropriate for our open systems environment, and will be used only where absolutely necessary.

2. LAN-based networking services must be so reliable that they can be a stable and undisputed basis for distributed computing.
3. Home terminals and mobile terminals require operation also in disconnected mode. Updating and resynchronization between such terminals and their central servers will have to become easier during connect phases. Supporting automatic resynchronization is becoming an increasingly important goal for the near future.

4.3 OS-Related Concepts

1. It would be nice to provide full parallel support for several OS-platforms, including the necessary bridges between similar applications running on these different platforms. Following our key strategy, we have had to select a single platform that fits most of our needs and supports a wide range of systems from laptops to such large multi-processor systems as SP-2's, all desired backbone services, and especially the requirements stated in points 2, and 7-9 in Sec. 4.1. *Only a multi-user multi-tasking system can satisfy these requirements. Hence, AIX was selected as our preferred system platform to carry all backbone services.*

2. *Other OSs, like OS/2, are needed especially for present laptops and to run certain applications* for some of our users. *We* have tried to integrate PC-type workstations running such OSs into our "open systems" environment by providing them with access to centralized services on AIX. Additionally, we have tried to find native solutions in the AIX domain for important applications.

3. Achieving the desired levels of security and privacy for business data or for classified mail in a large international company exceeds the capabilities of an intrinsically nonsecure file system like NFS. Therefore, we have used the *Andrew File System (AFS) with its Kerberos protection scheme for all user data in the AIX domain.* OS/2 machines can also store their files in AFS via an (experimental) OS/2 client.

4.4 Application-Related Concepts

1. *All backbone services have been implemented in the AIX environment.* This includes info, mall, file and print servers, data bases, and many others. Moving even conservative host users to this new environment is an ongoing task.

2. Given the AIX backbone system, we have been confronted with an ever-increasing number of new PC application packages and with users who are excited about the new programs they have "just installed very easily" on their home PC. As a support group, DCS has to negotiate a careful course of
   - granting users enough freedom to acquire and use *the* applications *they need, but*
   - without DCS having to support a multitude of incompatible graphical user interfaces, *serv*ers, and *d*ata formats, and
   - without users expecting DCS to resolve the ensuing often unresolveable compatibility issues which users typically appreciate much too late.

   Therefore, it has been our policy to offer users first an AIX-based application before endorsing a PC-only solution.

3. *Application support is given at different levels:*
   - full *support* for a few representative *groups* of *users, so* that *we* may learn how to support users with similar needs and can set up better tailored applications for all users,
   - full support for key applications of general interest such as mail and letter writing,
   - support for important general application packages,

- maintenance of certain special-purpose application packages, and
- installation only, or help with installation of packages needed by just a few users.

In summary, an application is only installed once, usually by the support group, and may then be used by any user, as far as licensing agreements permit.

Hardware and Software Administration

1. *Given the ever increasing needs of cutting-edge users, workstations must be replaced by newer and better ones relatively frequently. In order to prolong the useful life of such equipment, workstations are redistributed as new ones arrive. Equipment that has reached the end of the recycling queue of the lab can be phased out. This gives "normal" users an opportunity to improve their equipment without having to place an order. This is an important contributor to overall user satisfaction.*

2. *Within reason, all software should be available and executable on all nodes where it might be run. This is fairly straightforward in the continuously interconnected AIX environment with its multi-user node capabilities. The PC environments face some drawbacks in this respect. For some applications floating licenses are not readily available nor practical, and PCs with their limited internal hard disks are not necessarily always connected to large reliable file servers. Obviously, many of these problems disappear when PC users access the AIX backbone via X-Windows support.*

5 Solutions for Critical Issues in the AIX Domain

5.1 Management of the Workstation Cluster

5.1.1 The Environment

User Spectrum and User Expectations

Some 350 persons hold currently an AIX user-id. Our user spectrum ranges from "UNIX specialists" to "naïve" users. Obviously there is a large community of experienced computer users migrating slowly from other environments to AIX. These people have not only very specific expectations, but usually also lots of data and application programs which they need to take along. Many people in a research laboratory like ours need numerically intensive computations (NIC), visualization, or design systems. However, there is no distinction made between these researchers and administrative users regarding office applications. Everybody needs such basic services as a calendar system, mail, printing, and some office support programs including text processing, access to administrative data, spreadsheets, and even desktop publishing to prepare foils or reports. Nearly all our users are involved in some cooperative efforts for which they need communication networks and distributed file servers. And especially our research staff members demonstrate an insatiable appetite for more compute power and new advanced applications.

Development of Our Computing Support

The structure of the computing support in our laboratory developed like in many other similar places. After MVS and VM we had to add DOS, OS/2 and in 1990 also AIX for scientific applications. Given the manpower limitations of a small laboratory on the one hand, and minimal group sizes for supporting an operating system environment plus the extra support required for applications crossing the boundaries of two or more operating systems on the other hand, it became clear that we must concentrate on a single operating system environment in order to avoid inordinate complexities. With NIC being one of the most important computing requirements in our laboratory, the decision for AIX was made and is supported by all our departments. Obviously, this does not mean that support for other operating systems can be discontinued immediately and completely. In that sense we have been in a "transition phase" for several years. The "easy things" have been accomplished, like moving the physicists with their NIC applications to AIX, or offering "better solutions" like a group calendar, or good desktop publishing support. Local MVS host services were discontinued in 1994, and in 1996, our local VM host system is

planned to become only a server without direct login access for users. However, there are several projects that need specific application programs and operating systems for their work. Such situations will have to be handled on a case-by-case basis. Our decision to concentrate on AIX and on an open systems environment should not be understood to reflect value judgments on other operating systems. It is solely dictated by the scientific computing requirements of our laboratory and its manpower limitations.

Development of Our LAN Environment
As mentioned above, reliable LAN-based networking services are one of the prerequisites for distributed computing. These services are provided by a separate small project headed by W. Frei.
The LAN project started in early 1987. Its objective was to install a LAN hardware infrastructure to allow commercially available Local Area Network products to be used. We also hoped to gain experience with LANs, which would enable us to support future user needs and provide them with a test bed for local development efforts and research activities. Our strategy was to support mainly TokenRing LANs via the IBM cabling system (STP) within ZRL as well as Ethernet for special cases in its lab environments. The LAN consisted of a Centralized Backbone ring with subrings for each building. 8228 ('Non-intelligent') Multi Station Access Units were used in the wiring closets.

Initially, only bridges were used to connect the various Token-Ring segments in a transparent manner and were mainly used for mainframe access from LAN-connected workstations via PVM/PC, SNA and TCP/IP gateways. Connectivity to Remote sites was established via our VM host.

In 1990, the LAN environment had to be redesigned and expanded for several reasons:

- The new AIX/DCE environment required TCP/IP protocol support.
- A second laboratory site was opened in Adliswil about 5 km away.
- The migration from 4 to 16 Mb/s Token-Rings started.
- 'Intelligent' Token-Ring Concentrators/Hubs became available.

For availability, performance, and administrative reasons we decided to utilize CISCO Routers for better handling of the IP protocol for LAN and WAN connections. The IP LAN topology at each site changed from a Token-Ring backbone to a collapsed router backbone configuration via the multi-port routers. In order to improve token-ring availability, we replaced most of the 8228 MSAUs with the "intelligent" IBM 8230 Homogeneous Token-Ring Hub. Before that, we had several outages because of users attaching PCs set to 4 instead of 16 Mb/s to the faster rings.

The basic structure of the ZRL LAN as of mid 1994 is shown in Figure 1.

Size of Current AIX Support Structures
By the end of 1994, about six professionals were dedicated to AIX support. They provide support for some 160 RISC Systems/6000, an SP-2, and 100 XStations, as well as for AIX on most of the various PS/2s via XWindows emulators, like eXceed. The systems are installed at our two locations, which are about 5 km apart, and which are connected by four 2 Mb/s fiber channels.

5.1.2 Basic Strategies for the AIX Support

When we started the Distributed Computing Support Project in 1990, it was clear to us that we should find an optimal strategy between the following two extremes: Users install, maintain, and own their workstations. The classic, fully centralized (host or glasshouse) system approach with its unresponsiveness to the needs of individual users.

The positive reactions of our users indicate that we seem to have followed a very reasonable set of basic strategies:

Distributed Computing with Centralized System Management
Users can have "their" workstations completely installed and integrated into the system complex. We promptly also install new application programs in response to user requests. Therefore, users do not need and do not have root access.

Single System Image
All AIX workstations display a Single System Image (SSI) using a shared file system, Korn shell, and X Windows/Motif. Nearly all machines run currently under AIX 3.2.5. SSI permits users to login at any workstation within their reach, and to call the root menu for access to all currently installed applications without knowing or having to type specific AIX instructions.
Non-programmers almost never have to resort to the AIX command line. SSI has also drastically reduced the number of requests to move workstations to various workplaces compared to laboratories that use "personal workstations", and that have to accept the resulting significant work to relocate such workstations, and the additional efforts ensuing from transmission system problems.

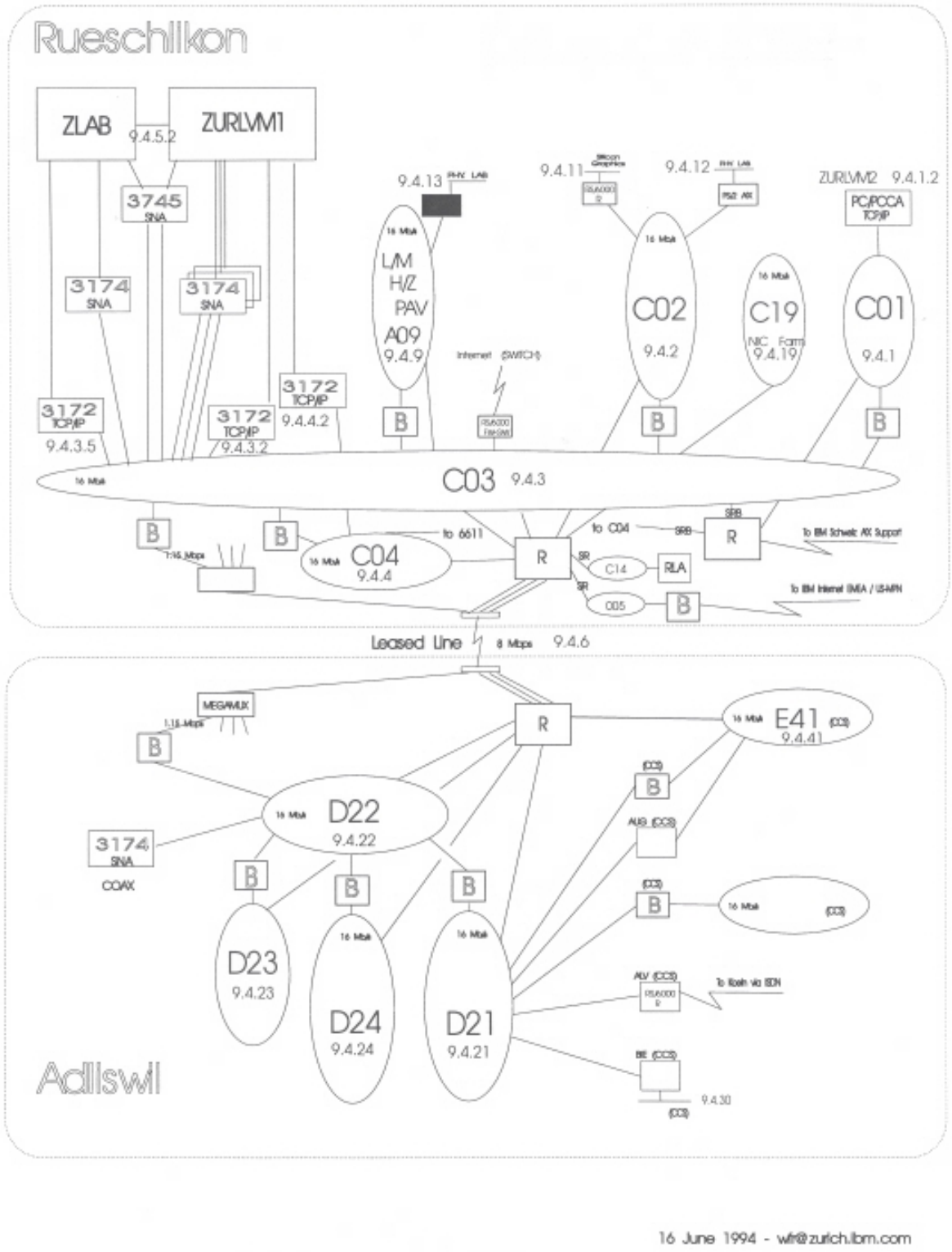


Figure 1

Global File System Based on AFS

In the interest of file security we have decided to use the Andrew File System (AFS) from Transarc instead of NFS. This decision is based mostly on its advanced authentication and file protection features using Kerberos. AFS allows the easy management of volumes. They can be backed up or moved while in use. Replication of data for increased availability is supported. The main drawback of AFS for us has been that a product which is continually improved cannot at the same time also be stable and error-free. Our overall rating of AFS is, however, very satisfactory. Now we are monitoring the development of the next-generation file system DFS developed by IBM, and will

switch to it and to AIX 4 when these new software packages have become sufficiently stable and suitable for our production environment.

Standard User Environment
Users are provided with a fully operational workstation plus services, so that even beginners can become "immediately" productive. The root menu includes all basic applications, like mail, text processing, and host access, as well as various help functions. A subset of the root menu can be customized by advanced users to fit their needs. This whole approach allows us to introduce new applications for all users without inconveniencing them in any way. Starting with the root menu, users can accomplish most of the things they may want to do without ever having to resort to the AIX command line.

Easy Migration Path for New Users
Many of our users come from other older systems that offered within their respective limits quite satisfactory services. Such users will not be willing to learn Emacs, Korn shell and C before they can become productive again. They want to get their first positive results "within a day", and we must endeavor to keep them satisfied in the new environment. Initially, most of them will enjoy a 19" screen for the first time, and all will find new and interesting applications, while having host services available as before. The next steps become more difficult:

- **Adapting to Filing Structures**
  Users must learn how to use file trees most effectively and how to manage the corresponding Access Control Lists.
- **Computing**
  Our NIC users could not simply switch from MVS to AIX, but had to migrate slowly. This included intermediate steps like keeping the data on the host and running Fortran programs in the AIX environment. All this required special support.
- **Mail**
  In the IBM environment one cannot merely switch to, say, Xmh and ftp and forget everything else. We have to interface to standard VNET services via VM hosts, and most users have to be able to handle mail also from their DOS-OS/2 home terminal, or maybe from old 3270's during business trips. We had to come up with workable solutions for all these difficult cases, while providing very good mall services in the AIX environment.
- **Education**
  The standard view that all AIX workstation users must know AIX, Emacs, etc., is certainly not true for our environment where system management is done centrally. Users can concentrate on working with their applications in a desktop fashion, unless they do their own programming. Hence, software training is still important, but we suspect it is much less of a problem than at other sites where innocent beginners can play havoc with their root privileges.

Extensions for Advanced Users
We have found that even "expert UNIX hackers" want to stay within the confines of our SSi, unless they really need a stand-alone laboratory machine. The ground rules are simple:

- Users can append to our standard **Path**. However, if they change it, they are completely on their own, and we will not support them if they run into problems.
- Users can only modify a designated part of the root menu, otherwise they are again on their own.
- Individual Restructuring of the file system is not possible. Users can put programs of public interest into /zurich/6k/contrib/bin. All other program installations are made by us centrally, which is usually much faster than if users did it themselves.
- Users do not get **root access**. Where necessary we provide **sudo** routines to allow users to make "legal" system modifications that would otherwise require root access.

Support "from the Desk"
Our most precious resources are our system experts. It was clear from the beginning that they should not have to run around at two sites fixing problems. Given our centralized support

philosophy coupled with the possibilities of a multi-user operating system, "maintenance from the desk" becomes possible. We can even completely reinstall systems in our complex without visiting the machine. Our hotline support profits from such possibilities. Most user problems can be resolved "from the desk" while the user is still on the phone.

Centralized Hardware and Software Administration
Whereas it makes good business sense to have various departments and projects order the components they need for their work out of their own budgets, there are limitations and disadvantages to this approach:

- Users or their managers should not be forced to understand the rapidly changing hardware and software offerings on the market, as well as their interoperabilities and constraints.
- A significant number of users in our research projects need "leading-edge" workstations which have to be replaced much faster by newer models than in more conservative environments. The redeployment of such nominally outdated systems cannot be resolved economically within project boundaries.

Therefore we have instituted central ordering and administration of all workstation hardware and software to optimize capital investments and expenses throughout the laboratory. This effort includes statistics on system utilization by our users. However, new capital investments have finally still to be justified and made by the respective departments and projects.

5.1.3 Current Tools

For cost reasons it would be very desirable to have only a non-overlapping set of application programs and tools to cover all accepted user needs. We have learned that, at least for a laboratory like ours, this is not a workable approach.

- We have to buy and install programs that have not yet reached maturity and are often superceded by others. Some users are then not ready to switch, and consequently our program portfolio grows.
- Even after a user committee has formally evaluated, say, two seemingly mature competitive software products and has decided to get one of them, a few users will inevitably present a valid case for installing the other product, too.
- For optimal productivity, visiting scientists require the tools they are used to at their home location. After they leave us, programs they developed at ZRL requiring such "private" tools usually remain, and they can even "infect" some of our permanent users, meaning that our program portfolio grows again.

After experiencing all these divergent interests we have decided to adopt a somehow relaxed attitude. We try to support fully within our manpower constraints those software packages which we consider to belong to a minimum set of core tools and applications. We maintain a few more, but without trying to understand them in detail, and finally there are quite a number of packages that we only install without giving any further service. Currently installed tools and application programs include the following:

Scientific Computing

- Matlab, Maple, Mathematica, and Axiom
- ESSL, IMSL libraries
- LoadLeveler and PVM for large parallel computations

Basic Office & Administration Support

- X3270
- Applixware (word processing, spread sheets, drawing,..)
- Z-Mail
- Framemaker, Interleaf, CorelDraw, LaTeX

- Netscape

Program Development

- Emacs, xcdb, dbx
- Fortran, C, C++
- AIX Tools like SDE, CMVC, AIC, and others

Hotline Support
The biggest contributor to the success of our support work seems to be the hotline desk with immediate second-level support as a rotating task among group members. This has two positive effects. First, users get immediate attention and qualified help. Second, all members of the group become familiar with the whole system, understand which problems really need to be fixed, and have to contribute personally to user satisfaction.

Our Users are Well Behaved
It is a frequent concern of people accustomed to a restrictive host environment that users in an *IX environment might misuse the many additional degrees of freedom provided by such systems. For example, at present we do not yet have tools installed that would allow us to manage critical system resources very closely. It turns out that this is not one of our most urgent problems. Once our users understand that some system resource is becoming constrained, they are cooperative, well behaved, and help establish fair access patterns. Therefore we can limit restrictions largely to root access in the interest of system and file security.

Required Level of Service
Whereas the availability of a single host system can be expressed in a single percentage value, availability in a distributed system becomes a much more complex concept. Laboratory machines may (intentionally) crash very frequently, but file servers should work continually without any grace periods for "periodically scheduled maintenance". All other system nodes have their own availability goals between those extremes. Providing the correct level of service with respect to availability and many other similar service characteristics is an important concept for optimizing overall cost/performance.

Limitations of Long-Range Planning
We all know that the planning cycles for new computing equipment and software are shrinking. Sometimes even the useful lifetimes of interesting products seem to become shorter than one year. This makes efficient long-range planning in terms of specific boxes and software packages very difficult. Again this calls for a centralized approach to the problem, which seems to be the only way to keep the variety of components in our environment down to a manageable level.

Optimization of Cost/performance
Part of the central administration of all our workstation hardware and software entails collecting data on system usage

- for specific system components ranging from the utilization of network nodes down to paging rates on processors of interest.
- by individuals, projects, and departments. This helps us understand usage patterns and user requirements. It will also enable us to issue future computing system charges to our customers. However, the goal of this effort would by no means be an exercise in penny-pinching. It is hoped that reasonable inputs on where and for what our computing budgets are spent will help
- us optimize our services,
- our users operate in a more cost-conscious way, and
- management steer overall project expenditures.

Some specific results of data collection in our system will be shown in a later section.

5.2 Security

Traditionally, UNIX systems have been considered "insecure", a favorite target of hackers around the world, and not to be trusted with sensitive information. The introduction of distributed computing into commercial environments made security a high-priority issue.

From the start our problem was to provide a secure system while preserving a user-friendly AIX environment for our researchers.

Security issues related to networking would go beyond the scope of this paper. Let us here just assume that our network provides a solid and reliable basis for interconnecting the components of our system cluster.

Even though the evolution of the AIX system at ZRL was not planned in detail, it was clear from the start that the system would have some special characteristics affecting system security:

- We would have a "true" distributed system where all user machines in the system would be "equal". They would not have a particular "owner", and users should be allowed to access and use any such machine in the system. Thus, all users within the system would be considered "regular employees".
- All machines, including file servers and service machines, would be managed centrally, i.e. there would be a group responsible for managing the total system.
- The system should try to offer security comparable to RACF in the host environment. Secure storage and handling of IBM Confidential data must be possible.
- The system should work well in a global environment, providing network access to the IBM internal network, access to data from outside, etc.

When we examined the requirements listed above, it became clear that many of the security issues would be different from the traditional host environment.

From our requirements, some good and bad points regarding security can be extracted. On the positive side:

- All machines in the system are managed centrally. This has allowed us to implement a uniform security policy throughout the system.
- Root (privileged) access to a user workstation is granted only in special cases. "Normal users" do not have root access.
- Machines are installed and maintained by the central support group, and a normal user cannot do any of this. The machines are physically locked, cannot be booted from a local disk, and the keys needed for making any changes are kept by the support group.

On the negative side:

- As all user machines are equal and a user can login to any such machine in the system, if security can be compromised on any one machine, it can then be compromised on all user machines.
- As users are not allowed to and cannot make system modifications, they must rely on the central systems group to do a good job.

The next subsection describes some of the solutions implemented in Zurich, where we have tried to take advantage of the positive aspects described above while minimizing the impact of the negative aspects.

5.2.1 Our Approach

This subsection describes some of the procedures and tools that we use for management of the Zurich AIX system. In looking at the following items one has to remember that a secure system is not created by a single solution, but by a synergy of mechanisms that covers all recognized potential exposures.

5.2.1.1 Andrew File System

The largest problem in a distributed environment is controlled access to data. Typically, user data will not reside on the local user machine, but somewhere on central file servers. This means that the user must ask another machine to send such data over the network. This creates a potential security problem.

We decided early to use the Andrew File System (AFS) from Transarc. AFS is a true distributed file system designed with efficiency and security in mind. Apart from all other advantages of using AFS, it offers the following features with respect to security:

- **Secure authentication**, i.e. users are really who they claim to be. AFS implements this by using a separate authentication server (a modified version of Kerberos). At login time, the authentication server provides a temporary token which is later used by various servers, e.g. file servers, to determine whether a given user process has the necessary access rights.
- **Access Control List** (ACL), i.e. a list with users and their access rights. An individual ACL controls the access to every directory, e.g.:

  ```
  Joe   read write
  Jeff  read write administer
  anyuser    read
  ```

  Here, Joe can read and write files, Jeff can do the same but can also change the ACL, and "anyuser" can only read files.
- **No visible passwords**. AFS hides the passwords in itself, and they are not visible to normal users, not even in an encrypted form.

AFS has proved to be the most important component in our security policy, and several of the tools below were implemented by using (and bending) AFS.

5.2.1.2 sudo

Traditionally, system management of an AIX machine is done by a person with root (system) privileges, i.e. to perform system management, one has to login as user "root". The root user has all privileges on a machine, and of course, the password for root must be protected carefully.

Because we manage all AIX machines centrally, it would be necessary to keep the root password the same on all machines and the root password would also have to be shared among the members of the systems group.

There are several disadvantages to this:

- A shared password is never good because the password will be known by more and more people over time.
- Some users may need root access to a small set of machines to be able to do their work. A good example are researchers doing systems prototyping in their lab. A shared root password would again not be a good solution for them.
- It is very difficult to limit system access; once logged in as root, there is virtually no limitation to what one can do.

On the other hand every user needs some of the functions normally provided by root. For example, only root can close a system down in a controlled fashion (shutdown), or only root can cancel print jobs on a public print server. We soon realized that if we could not use a shared root password, something better was needed. In the UNIX literature, we found an elegant solution: *sudo.* It stands for "super-user do", and offers the following:

- The sudo program allows selected users to execute specific commands as root. The commands are listed in a file maintained by the system support group.
- A special C program (sudo) reads the file with commands, determines whether the user is privileged to issue a given command, and then executes the command if privileged.

- The sudo program keeps a log of the commands executed, when they were executed and by whom.

The sudo command allowed us to solve the problem of shared root passwords: we simply do not have root passwords! To gain root access to a machine, we issue the command:

```
sudo ksh
```

which gives a shell for system administration requiring a separate login. Hence, trying to guess the root password of a machine is meaningless because no one can login as root. The only access is through sudo.

5.2.1.3 Liberté, egalité, fraternité

As we decided that all our user machines should be equal, all users can login and use all machines. If users have successfully logged into one machine, they can then login to other machines without typing a password. This has been done by using an .rhosts file in the home directory of each user containing all machines that support our Single System Image.

Normally, the use of **. rhosts** is discouraged, because it may create a security exposure. However, this precaution does not apply to our environment, as access to .rhosts is only possible for users authenticated by AFS and its ACL's. On the contrary, this file has proved to be most valuable for us. Using it, we can give our users transparent access to all machines without requesting them to type the password each time they want to access an additional node in our system cluster.

Of course, there is no rule without exceptions. We have some machines with restricted access, and all machines have restricted access for root. Only members of the system group can login on a few special machines; these are our trusted/secure machines. Only coming from a trusted machine is it possible to rlogin as root on another machine.

5.2.1.4 Central Management/Installation

As all machines are delivered to the user with the necessary software, it is easy to make sure that all necessary security fixes have been installed as well. Upgrades of machines are done centrally. This allows security holes etc. to be plugged quickly as they are reported. We also know that we deliver clean systems, because the install images we use would be very difficult to tamper with.

It has also been of great value to have one person responsible for security. Reading forums, scanning information for fixes, and implementing them as necessary can be quite time-consuming and we cannot require that everybody do this in a timely fashion. For example, more than one year ago we received an official warning about a serious security exposure for most UNIX-type installations late on a Friday night. One of our system administrators saw the message a few hours later on Saturday morning, and he had fixed the problem for our whole system cluster in less than two hours. Obviously, installations with "personally owned" machines could not react that quickly. In that case one would have to first inform all users, ask them to fix their machines, and, finally, one might have no way to make sure whether or when all machines had really been correctly updated.

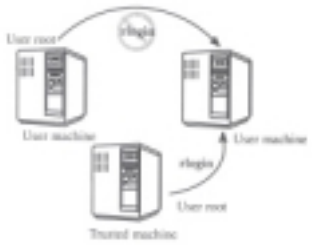


Figure 2

5.2.1.5 Big Brother

Even if we deliver "clean" systems to our users, it is also necessary to supervise and check the running systems. Things that might "happen" while a machine is running are:

- File protections are changed by accident, e.g. `/etc/passwd` becomes writeable for "anyuser".
- Setuid programs are installed by "mistake".
- User accounts without passwords are created.
- etc. etc.

There is an obvious need for a "Big Brother" to watch the systems. In the public domain, there are several packages available that do this, the best-known package being called COPS. All of these packages must be tailored to a given system as one might want to perform checks differently than those available in the package.

In our environment we use a locally developed package called *BigBrother.* BigBrother is submitted from one of our trusted machines to the machine to be checked. While the check routines run, a log is written, listing potential system problems, or "unexpected user behavior". The log is compared with a database containing information about the current "clean" system version and user authorizations. If any differences are found, the support group is notified.

5.2.1.6 Mail

Mail in traditional *IX systems has not been very secure. Implementing secure mail in a distributed system using standard tools is not easy, but we have at least been able to achieve secure storage/access to mail. Between our local AIX users we can even guarantee secure mail delivery, i.e., mail will

correctly identify its sender, and will only be accessible to authorized parties.

The only way to have secure storage/access of mail in a distributed environment is to have secure data access. AFS gives us the secure data access, if the mailboxes (`/usr/spool/mail`) are kept within AFS. Moving the directories was easy, but the standard AIX tools like `sendmail` and `bellmail` complicated things. These programs are not well documented and their behavior made them difficult to integrate into AFS.

The figure below shows how mail delivery is done in the ZRL system. The drawing is not comprehensive, but illustrates the principle of our mail delivery system:

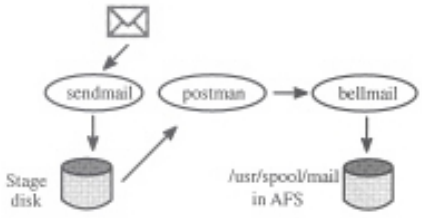


Figure 3

Authenticated mall can be sent within our local AFS cell relying on AFS file protection. In this case mail gets an additional header tag, like

```
Subject: change password
Date: Thu, 18 Mar 93 09:33:24 +0100
X-Afs-Authentication: er@zurich.ihm.com
```

5.2.1.7 File Sharing

Many of our users come from a host environment (VM/MVS) and were not used to the *IX concepts of sharing data. On the host, one has very few shared files, and if one wants to pass,

say, some foils to a colleague, one sends the corresponding file as mail or with the `sendfile` command. By doing this, one need not grant other users access to files because one "gives away" only what one wants the recipient to see.

In AIX, it is much easier to share files (especially with a global file system). Shared files offer the advantage that several people can work with the same file. If changes are made to the file, the file does not need to be redistributed to several users.

Shared files are more complicated with respect to security. In AIX, if you control a file and want to share it with several people, you must set the correct file protection on the *file.* In AFS, you must define an ACL for the *directory* in which the file resides.

For several of our users, sharing files looked too cumbersome, so we came up with a temporary solution, the idea of "file mailboxes".

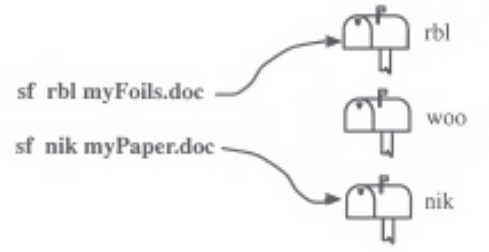


Figure 4

Any user (within the Zurich system) can now put a file in the directory `rbl.` The user rbl can read and delete the file. We have written a small tool that allows a user to send a file easily:

```
sf rbl myFoils.doc
```

A file mailbox takes advantage of a feature of the AFS ACLs called the *insert* privilege. This privilege allows users to create a file in a directory but not to read it afterwards. We defined directories (or mailboxes) for all users with the permissions:

```
Directory rbl: rbl insert read delete authuser insert
```

(authuser is the AFS term for local users).

The above solution is one of several attempts to render previous host users immediately productive in their new AIX environment. In parallel we try to encourage the utilization of all the possibilities of an advanced distributed file system, like AFS, within our user community by

- Generating a Public directory in the home directory of every user. There one can put files that should be accessible to all users in our local AFS cell.
- Generating subdirectories accessible only to fixed sets of users, like projects, departments or any other defined set of users. We encourage users to benefit from such shared facilities, because they are easier to administer by the projects and by us, especially if users leave, or move to other projects. Additionally, these shared facilities result in significant overall savings of disk space.
- Offering tutorials on effective utilization of ACLs for organizing data structures belonging to teams of users.

It will take some time before all our users understand the potential of file sharing compared to sending files to each other.

5.2.1.8 Uniform User Environment

Since our goal was to hide the complexities of security as much as possible from inexperienced users, each user initially gets a standard environment with reasonable defaults.
The first time a user logs in, he or she has to select whether the home directory should have restricted access. A home directory with restricted access does not allow access by other users. If the home directory is kept "open", other users can list the contents of directories, read configuration files, but cannot read any personal files.

We also provide some directories with special access rights. For example:

- Confidential: No access for other users.
- Mail: No access for other users.
- Public: Read access for "anyuser".

5.2.1.9 Tools and Education

Based on the principle that no chain is stronger than its weakest link, it is important that our users have a fundamental understanding of the AIX system and the tools at their disposal. There should also be documentation available that describes the security aspects of the system, provides helpful hints, etc.

Some of the tools that we provide are: Access check: An easy way to find out the ACL settings on directories. AFS ACL: An X-Windows tool that makes it easier to list and change AFS ACLs. Password checker: Checks whether a password complies with IBM's internal security rules.

Regarding documentation, we provide security guidelines that describe the most important security issues. We also try to provide basic user education at regular intervals.

5.2.1.10 Internet Connectivity

Our users show a strong and growing interest in accessing the worldwide non-IBM Internet using mail, ftp, telnet, or WWW services. In order to achieve the necessary level of separation between the internal IBM network domain and the external Internet we use a firewall machine running IBM NetSP software, and a proxy server on our internal IP network to handle external requests. All this allows our users to access Internet resources without too many security restrictions, and it guarantees security from outside intruders.

Given the dependencies of our users on outside connections we keep a second firewall machine in stand-by mode as a spare in case of hardware problems.

5.2.2 User Groups with Special Security Requirements

Some groups have special needs regarding data security, data availability, and data integrity. A good example is our personnel department, which requires that access to their data, their backup records, and to certain processes is limited and tightly controlled. We have achieved this by setting up a separate subsystem for this group, such that even our system administrators cannot directly access any of the personnel data. The group owns a set of machines which only group members can access and utilize. Not a single byte leaves the trusted environment unencrypted which consists of a network on its own, workstations, printers, and a router. Backup of data is also done in their encrypted form, which can then be stored together with all regular backups on tapes. To restore personnel data into readable form, a member of the personnel group must enter the decrypt password into the system. DCS administrators can only access these machines as a regular users without special rights. Only the personnel manager may authorize system maintenance work by providing a temporary password. During maintenance work the Administrator is either supervised by a member of Personnel, or unsupervised maintenance on a machine may begin only after all available sensitive data has been encrypted on that particular machine. Obviously, our approach is still not completely secure against all conceivable forms of attack, but it seems to offer better privacy and security than the old host environment, and, most importantly, it satisfies the expectations of our most sensitive local groups of users.

5.2.3 Software Management

Centralized installation and maintenance of software products is an important requirement of efficient software management, and especially system security.

5.2.3.1 Operating System

In a system cluster with centralized control one would like to be able to install or reinstall machines via the network. As such a feature is not supported on AIX levels up to 3.x, we wrote our own OS installation tools which permit rapid and largely automatic Remote installation via NFS. First-time installations require the target machine to be started with a few standard diskettes, and reinstallation's can be done completely in Remote mode. Equivalent facilities are becoming available as part of AIX 4.

5.2.3.2 Application Programs

Although we install the operating system and closely related software for performance reasons locally on all machines, most application software is installed by the support group into the AFS filespace. This allows controlled installation, maintenance, and license control, and provides different versions of application programs. Response time problems due to network delays can be minimized, because AFS permits replication of data on several file servers and supports local caching of user data.

5.2.4 User Reactions

When implementing security in a system, it is important to remember that security for its own sake is meaningless; it must serve some purpose. Ultimately, security must serve the users; too much or to little is not good, it should be "just right".

During the evolution of the ZRL AIX system, we have added several security features to the system such as AFS, no root password, password expiration etc. How did our users and researches react to this?

- **Different needs:** We have found that the users most concerned about security are "nonhacker" users, like secretaries, etc. The users with extensive know-how about computers normally find their own ways, but other users need more support. We have received requests from about 10% of our users for the AIX support group to man age access control to their data. This might be an indication that our information and documentation still needs to be improved.
- **New environment:** Most of our users came from a host environment (VM/MVS) much different from the AIX environment. The "culture shock" made it difficult, as there were so many changes at the same time. Most users were busy adjusting to the new environment, and were not too interested in security issues. As they became more comfortable in their new environment, security became again an issue for them.
- **No need to worry:** Most users do not want to worry about security, it should just "be there". Users appreciate that security for "basic AIX" is handled centrally. Most users define their environment only once, and find our default user environment to be of great value.
- **Little feedback:** We have done quite a bit of work regarding security, but we have received surprisingly little feedback from our users. The optimistic interpretation is that we have done a good job, and have been able to hide most of the tedious sides of security ("If no one complains, it's good."). Several "friendly" audits have not found any major flaws.
- **Poor tools and documentation:** The complaints that we do still receive pertain to tools and their documentation. We hear questions like: *"How can I check that my data is secure".* Unfortunately, we cannot say: *"For that, you use tool XYZ"* and *"It is documented in ABC".* There are few good tools and documentation at a suitable level for our users dealing with security in a distributed system. Most of the tools we use, with the exception of AFS, are "home-grown", and only briefly documented. Hopefully, this will improve over time.

5.2.5 Lessons Learned

What have we learned about security in a large distributed AIX-based system with centralized control?

- **Security is necessary and possible:** Driven by user expectations and IBM-internal security guidelines our system offers adequate or better security or privacy to meet prevailing requirements. We have no illusions that our AIX system is "perfectly" secure, but its security seems to us more than adequate for our commercial environment. However, it is important to remember that security is one of those areas that is never complete. To provide a secure system requires one to constantly improve monitoring tools, to keep abreast of new developments, add security fixes or changes to the operating system, monitor system activities for possible violations, and so on.
- **Central system management is necessary:** We do not see how one could provide a secure, distributed AIX system without central management. Central management may have a number of disadvantages, especially as far as the emotional reactions of certain users are concerned, but when we consider security, it is the only way to go. Such statements may not apply to single-user machines connected by a network for transporting only mall between them. However, as soon as data bases and computing resources have to be shared on a discretionary basis, like for supporting teamwork between users, centralized management becomes a necessity.
- **New approaches are required:** A distributed system requires a different approach from a stand-alone system like a host. Literature on security in distributed systems and good tools is only slowly becoming available. Although commercial tools would be highly desirable, their non-availability cannot be construed as an excuse for not providing a distributed system cluster with adequate security levels.
- **Optimization of resources:** Basic system security cannot and should not be the concern of all users. The only way to deal with security in a distributed environment is to have one or two people responsible for system security. Only then can one implement a uniform security policy with reasonable amounts of human resources. Obviously, individual users still have to take responsibility for administering their own data within the general environment.
- **Listen to your users:** Security exists for users and it is there to serve them. Listen to your users, and they will tell you what is important.
- **Security is still visible and not complete:** We have tried to hide as many security issues as possible from our users, but have not always succeeded. Some security issues are still visible and sometimes cause unnecessary work for our users.
- **Don't tell users about exposures:** It is generally better not to broadcast security exposures. If a hole has been detected that cannot be immediately fixed, it is often better to keep it a secret from normal users. Security is a sensitive area and good judgment is crucial.

5.2.6 Conclusions

- Security in a distributed AIX environment is possible and realistic.
- By managing security issues centrally, it is possible to achieve substantial resource savings while improving the security level of the system.
- Implementing security is an ongoing process that is never completed. When allocating resources, one should not forget the security effort!
- When speaking of security, one has to consider the whole picture, and not just some specific details. A secure AIX system includes network security, user education and acceptance, user administration, procedures, etc. Former host administrators have to learn that the complete picture for a distributed system is significantly larger and more complex than for a single host!

5.3 E-Mail Services

5.3.1 Basic Considerations for ZRL E-Mail

Within IBM, e-mail is one of the most important applications probably used by every user. Therefore it is essential to provide reliable, easy-to-use e-mail support in the new environment if we want to convert from the "classical" VM host solutions to a workstation-based distributed environment. During the transition phase it is also important to make sure that our users find an easy migration path, and that the new environment can coexist with the old one locally, as well as with Remote environments inside and outside IBM.

5.3.1.1 Different User Environments

Basically we were faced with three different kinds of environments:

- In a classical IBM environment, e-mail services are provided as a VM application, and RSCS is used for mall and file exchanges between VM nodes. As the majority of the ZRL population are experienced computer users, they use the note exec directly. PROFS is not supported at ZRL.

- For personal computers with Intel-based operating systems several e-mall systems have evolved. However, none of their message-handling protocols have become a generally accepted standard. Without a strong incentive to move to a PC-based solution most of our users with OS/2-DOS workstations have continued to access e-mail services on VM via 327~1ike terminal emulators.

- On AIX workstations the situation has been different. The standard UNIX mail handler and SMTP functionality come with the basic operating system. In the non-IBM world SMTP is a standard solution used for worldwide e-mall exchange on the Internet. SMTP usage is clearly gaining against the RSCS-based BITNET. However, the usual way of running e-mall services on AIX systems within IBM has been strongly influenced by the way these systems were managed. At most IBM sites AIX environments evolved around personal activities. Users simply bought a machine for their own use, and they also installed and administered it on their own. Quite naturally, these users wanted their e-mall to be sent or forwarded to their personal machines, and not to be handled by some central server of dubious reputation. As there usually was no standard mail service adequately supported, users chose out of a multitude of offerings whatever happened to appeal to them.

5.3.1.2 Generic Addresses

Originally the mail address of a user consisted of a user-id and the name of the VM host node supporting this user, like
USER at ZURLVM1.
With the increasing number of hosts and workstations it became clear that there were conflicting requirements. Users wanted a stable address at which they could always be reached, even if the processor node changed for operational or other reasons, such as a case of a user deciding to get mail no longer in the VM domain but in the AIX domain. This led to the introduction of generic addresses. For ZRL the obvious choice for generic addresses is
USER at ZURICH
as the address visible from all RSCS systems, both inside and outside IBM, and
user~zurich. ihm. com
as the address visible from all SMTP systems, both inside and outside IBM.

In view of our centralized administration we have decided that normal users shall have only one unique user-id within our local domain. From any system in the lab it should be possible to send e-mail to any user by just typing his/her unique user-id, and the mail should be delivered to the place where the recipient is expecting it. From outside the lab, both inside and outside IBM, all ZRL users should be reachable by the same user-id plus a generic site address. For consistency and to make the 'reply' function work correctly, all e-mail sent by a ZRL user should appear to come

from a fully qualified generic address. On their business cards all ZRL users should only have to publish their generic address.

Using the same scheme throughout IBM would result in a truly two-dimensional namespace (USER,SITE). This appears to be the proper granularity for IBM's size and allows independent site administrations. Considering the increasing number of XAGENT installations following that idea, it appears that this view is widely shared throughout IBM.

5.3.1.3 XAGENT

To allow users on AIX systems to communicate with the rest of IBM still using VM, the SMTP and RSCS worlds had to be linked together. As most VM sites even refused (and sometimes still refuse) to install TCP/IP on their machines, conversion between SMTP mail and RSCS NOTE format had to be done at the AIX users' own sites. XAGENT was developed by David Singer and others at the IBM Almaden Research Center to serve this purpose. Because the RSCS world did not support names with more than 8 characters, it was also necessary to map possible SMTP addresses like `michael.niksch@spitzhornli.zurich.ibm.com` to RSCS addresses like `NIK at ZURICH.`

5.3.1.4 Accessing AIX System Mailboxes

As the ZRL AIX environment was based on AFS rather early, it was a natural choice to put all mailboxes into AFS as well. On all our machines, /var/spool/mail is a symbolic link to /afs/zurich.ibm.com/ spool/mail. AFS provides the necessary security for our mall. If desired, ACLs can be used to grant others access to incoming mail, e.g. while on vacation. System mailboxes are also accessible from AFS clients at other sites. Having system mailboxes in AFS does not rule out use of a POP server. In fact, any AFS client can be set up as a POP server. Only a slight modification to the server code is necessary to make it accept an AFS password and obtain an AFS token. We have started experimenting with this, but we do not use it in production yet.

The complicated part of using AFS is mall delivery, because it requires write access to the system mailbox. For a local or an NFS filesystem the delivery program can work *setuid root* or *setgid mall* to obtain necessary access privileges. However, an AFS token is required to write to an AFS-protected system mailbox. To solve this problem we have created a special shell script called *zurimailer,* which is invoked by sendmail.cf as the local mailer. This script places the mail as a new file into an AFS spool directory with *system:anyuser li* rights. Another script called *zurimallerd* runs as *root* on a trusted server machine. It has access to an AFS password authenticating it as a member of the *system:postman* AFS group. This script periodically checks the AFS spool directory for new mall and delivers it to the appropriate system mailbox.

5.3.1.5 External Gateways

Already at times when using only VM-mail, many users at IBM sites had increasing needs to communicate with partners outside IBM. Many different mail gateways to non-lBM networks were installed at various places.

The earliest gateways were probably those between the IBM VNET and the external RSCS-based BITNET and EARN, BITNET's European counterpart. Fortunately, these external networks became linked to each other, and it became unnecessary to distinguish between them and run independent gateways to each of them. Special derivatives of NOTE EXEC showed up to communicate with these networks. With external non-RSCS networks evolving, derivatives of NOTE EXEC were even taught to understand non-RSCS recipient addresses.

External gateways created a need to map IBM addresses to external addresses. The simplest approach to achieve this was to give gateway users an ID on the gateway node. However, as an increasing number of IBMers from different sites requested access to the gateway, it became obvious that mapping the USER at NODE namespace within IBM to a completely flat namespace seen from outside was bound to fail for a company of IBM's size.

Using fully qualified names like
`user0zurich.ihm.com`
together with appropriate gateways to the internet resolves these now historical problems.

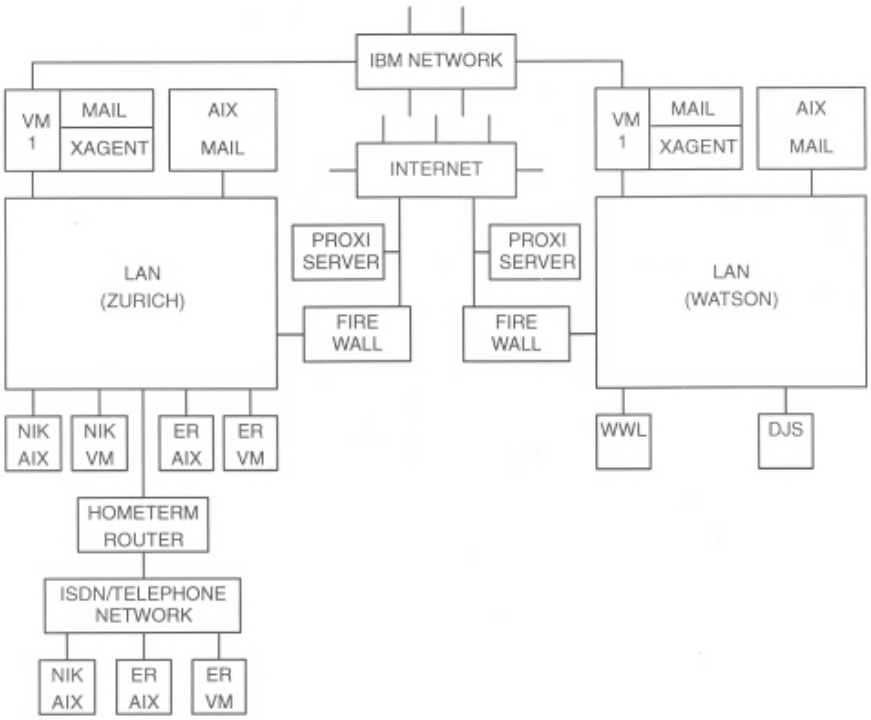


Figure 5

5.3.2 Basic Structure of ZRL E-Mail Network

In the old host environment the conceptual structure of the e-mail network was quite simple (Figure 6). The Watson Lab in Yorktown Heights, N.Y., is shown as a second site, because our gateway to the outside world has historically been there.

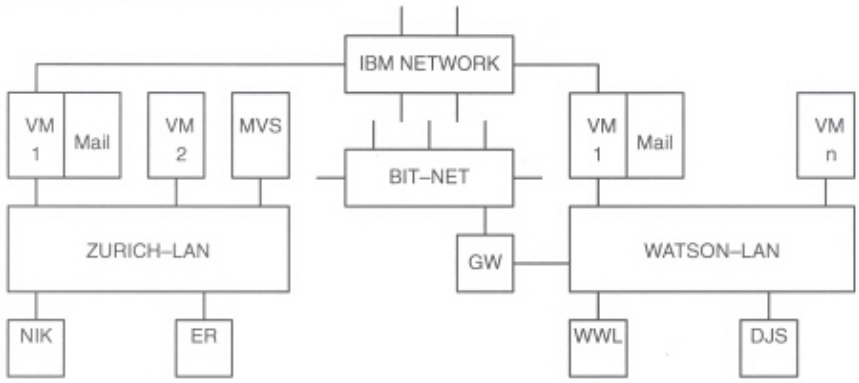


Figure 6

Our new environment requires mail support for users of VM, AIX, OS/2, and DOS/Windows. Currently, users of the latter two operating systems handle their mall either via 3270 emulators in the VM domain or via an X-Windows client in the AIX domain.

5.3.3 Routing of E-Mail

5.3.3.1 Routing of Notes

- **VM to VM**

The path of VM notes can be followed in Figure 5 or Figure 6. If NIK in ZURICH sends a note to DJS at WATSON, the note goes first to NIK's VM host ZURLVM1, and then via the IBM network to the attached VM host at Watson. There the generic address DJS at WATSON is resolved to the real address, say, DJS at YKTVMV. The same procedure applies to a reply message to NIK, which would be resolved in Zurich to NIK at ZURLVM1. The resolution of generic host names is always done locally by modifications of the VM `note exec`. Generic and real node names are all unique in the IBM network to avoid misroutings. If a user wants to send mail explicitly to a VM port, he or she has to specify a real VM node name.

- **VM to/from AIX**

As mentioned before, all our local user-ids are unique and identical for VM and AIX. In accordance with the Single System Image philosophy AIX users have to be able to see their mail on any AIX machine they happen to be logged into. Additionally, users should be free to decide whether they want to handle their mail in the VM or AIX domain. This goal was achieved with the help of XAGENT shown in Figure 5.

XAGENT is installed on our local ZURLVM1 system and does the essential part for the RSCSstyle generic addresses by handling the ZURICH VNET node. Where necessary, it converts between RSCS and SMTP format and rewrites *user~zurich.ibm.com* to *USER at ZURICH* sender addresses when mall originating on our AIX system is being routed to an RSCS network. Using a modified NOTE EXEC makes e-mail originating on our local ZURLVM1 system appear to come from the ZURICH node. The same NOTE EXEC can also be used to address e-mail to SMTP addresses both inside and outside IBM. Our local VM system acts as a gateway between VNET and EARN/BITNET, and the ZURICH node has also been registered externally. A special Internet mail gateway provides the *zurich.ibm.com* generic node visible from the Internet. For historical reasons, this gateway is still run on a VM system located at the IBM Watson Research Center. Technically, this gateway acts similarly to the *vnet.ibm.com* gateway widely used via the INOTE facility. Of course, users also obtain their unique user-ids on the Internet gateway.

All machines in the Zurich AIX environment use the same, heavily modified sendmall.cf file. It takes care of rewriting sender and local recipient addresses to the fully qualified generic *user~zurich.ibm.com* format. On the other hand, all machines consider mail to *user~zuric-h.ibm.com* local. A common mail.aliases database takes care of redirecting such mail to the user's preferred destination, i.e. it redirects it to *user~zurivm1* if the user has chosen to receive mall on VM, whereas mail to be delivered on AIX is treated in a way described below Our sendmail.cf file also handles automatic RSCS sending, i.e. it treats mail to *user~node* as if it had been explicitly addressed to *user~node.rscs* when *node* is known to be a VNET or BITNET node. In addition, it accepts the form *user~node.vnet.ibm.com.* Mall to non-lBM SMTP addresses is routed to the *zurich.ibm.com* Internet gateway. As we found no way of routing mall addressed to *nickname~vnet.ihm.comto* the *vnet.idm.com* gateway via the internal network, such mall is also sent to the outbound *zurich.ibm.com* Internet gateway.

Within the IBM IP network one of our servers acts as the *zurich.ibm.com* generic node. As all our AIX machines accept mail to this node as local, any machine can be used to serve this purpose. Machines that are not part of our Single System Image can use *zurich.ibm.com* as a relay node for all outgoing mail.

To make sure our generic addresses are used by our partners within IBM, the IBM CallUp data-base lists all Zurich users as *USER at ZURICH.* However, it has proven a seemingly neverending task to get all our users and all our partners at other locations to replace all occurrences of *USER at ZURLVM1* in their NAMES and aliases files by our generic addresses.

XAGENT allows a VM user to change the destination for mail arriving at the *ZURICH* generic RSCS node. A local tool allows a user to simply type

```
ZURIMAIL TO AIX
or
ZURIMAIL TO VM
```

to get mail delivered to the system of his or her choice. Unfortunately, we do not currently have a way of changing the destination via an equivalent AIX command. So, for many new users it might be the only time in their life to login to a VM system when they have to type *ZURIMAIL TO AIX.*

To make sure mail arriving at the *zurich.ibm.com* generic SMTP node is routed to the same system, the mail.aliases map is rebuilt nightly from the MAILBOX XAGENT file. As mail arriving at the *zurich.ibm.com* Internet gateway is still routed via the *ZURICH* RSCS node, no special action has to be taken to direct it to the proper system.

If the sender has a reason to send mail to a particular system, this is still possible by addressing it to *USER at ZURLVM1* or *user~maller.zurich.ibm.com.* Whereas the former is obvious, the latter is achieved by trapping the *mailer.zurich.ibm.com* pseudo node in sendmail.cf and have mall addressed to it avoid being redirected via the mail.aliases map.

In addition it is still technically possible to address mail to *user~hostname.zurich.ibm.com,* in which case the mail is routed to the specified host via SMTP. Although we are not actually using this currently, it would allow us to deliver e-mall to machines that are not part of our Single System Image.

- **Local AIX to AIX**

From the above it should have become clear that local notes going from an AIX sender to an AIX receiver simply need to specify the destination user-id as a valid address; fully qualified addresses are filled in by AIX MAIL. However all these notes would still have to pass XAGENT, because the potential receiver may have switched his or her mailbox to VM. In order to avoid this unnecessary detour for AIX-AIX mall, the MAILBOX XAGENT database is shadowed on AIX in the mail.aliases map. Mail originating on AIX and addressed to *user[@zurich.ibm.com] is* therefore never converted, but delivered immediately on the AIX system if the addressee has chosen to receive mall on AIX.

Selecting a standard AIX mail handler turned out to be a difficult task. Although the UNIX world knows a multitude of mall handlers and corresponding user interfaces, we wanted again to support only one solution. Some mall handlers are available at no additional cost, e.g. the UNIX *mall* program and the *mh/xmh* system. However, they do not satisfy all our requirements. Some office applications come with a nice built-in mail handler. However, if one decides to rely on one of these, it may happen that the rest of the application package turns out to be inadequate, and one is stuck with supporting it just because of its mall part.
Therefore, we have decided to go with a modern, dedicated mail application. We have bought a site license for Z-Mail from Z-Code Software Corporation. Z-Mail is an advanced and easyto-use mail handler providing a nice Motif user interface and MIME functionality. It is highly configurable by both the system administrator and the end user. This allows us to provide a default environment meeting typical needs, whereas users can easily add personal configurations and additions.

An example of local customization relates to Z-Mail's ability to invoke an address lookup program whenever an address is typed into a *To, Cc,* or *Bcc* header field, or just before a message is actually sent. We are using this feature to automatically expand local user-ids with information from the mail.aliases map (so that it is even available in the copy kept by the sender). Names that are not local user-ids are looked up as real-world names first in the local, then in the global IBM CallUp database. It is also possible to force a global CallUp search. We are using the AIX CallUp implementation developed by John Bissell at IBM Austin for this purpose.

In the VM domain users have access to alias lists via personal NAMES files, and via general system files which contain aliases for, say, all employees who report to a particular manager, belong to a particular department, or have their office in a particular building. Corresponding information is kept in the AIX domain by rebuilding the mail.aliases map nightly from XAGENT information.

As we have the system mailboxes in an area shared by all users, we must make sure that users actually move incoming notes into their own home directories. Our Z-Mail configuration therefore hides the user's system mailbox and instead provides a *NewMail* button to move all incoming mail from the system mailbox into the user's private *mbox* before the user can read it. Though this is equivalent to the *inc* function of *mh/xmh,* we would rather live without this nonstandard usage of the system mailbox and the *mbox.*

Z-Mall supports read-only folders with external indices, which we are also exploiting for our Zurich Mail Bulletin Board. Its folderformat is compatible with UNIX *mail.* Unfortunately, *mh/ xmh* use a different folder format. Z-Mall can also act as a POP client. There is a command line interface and a full-screen terminal interface. Moreover, a WINDOWS interface is being developed by Z-Code. Approximately 1/3 of all ZRL users currently receive their mail on AIX. About 80 percent of them have chosen Z-Mail, and the rest is still using *xmh.* We believe that our investment in a Z-Mail site license will certainly pay off for our users.

- **Local to Remote AIX**

Figure Figure 5 shows two different possible paths for routing notes to Remote IBM locations. One goes via the IBM Network by specifying a destination address like DJS at WATSON. The other one goes via external networks like Internet through the two firewall machines at either end. The necessary corresponding destination address is dj s@watson. ibm. com. In any case a local master controller has to make sure that notes take the desired path. In our VM/AIX environment it has been the simplest solution to use XAGENT in the VM domain for this purpose, in a strict client/server environment a separate server machine could also have assumed this task.

5.3.3.2 Routing of Files

- **VM to VM**

Mailing of data and program files is a routine procedure in the VM domain. Whereas it is possible to create shared collections of files for "public" access or for specific user groups in a VM RACF environment, the setup of such facilities is a task reserved for authorized VM system administrators. VM users have learned to avoid this often cumbersome approach and mail files to each other using the sendmail facilities, even if the maintenance of multiple copies of larger files may cost significant amounts of harddisk storage.

- **AIX to AIX**

File transmission within the AIX domain is usually not the best and only solution to utilize some-body else's files. AFS permits a user to give access rights for specific subdirectories very readily to any other group of local users, and files readable for 'anyuser' can be accessed also from other AFS cells. DFS will expand these facilities even further. Therefore, files can usually be copied, or simply linked, and need not be transmitted. Obviously, it is also possible to transmit files using the standard UNIX ftp mechanism.

- **VM to/from AIX**

In an environment where users are gradually migrating from VM to AIX, one of the key problems is to correctly handle file transfers from a VM user to an AIX user. After ail, the sender may not even know that the receiver operates in the AIX domain. Whereas a VM user can accept transferred VM files directly, the AIX user has to specifically download them in binary or ASCII mode, and needs a backup in case the wrong mode is chosen. For this purpose we have written an AIX-extension to the sendfile command which satisfies these requirements and permits the user to remail the file to his or her VM mailbox if the file can only be utilized in the VM domain anyway. We have also succeeded in reducing the need to logon to VM for AIX users by modifying our printing subsystem so it can accept the printing of LIST3820 files in the AIX domain as well.

5.3.3.3 Handling of MIME Attachments

Like several other modern mail systems, Z-Mail allows users to send a file attachment together with a note. Such attachments can be correctly interpreted by a receiver that also conforms to the MIME standard. This feature allows, for example, any of the following to be sent together with the note: an image, a formatted document, a spreadsheet, an audio recording, or other information in form of a file which can be directly interpreted and immediately presented by a properly configurated receiver station. Obviously, this requires that sender and receiver have MIME compatibility, and that they both are able to run the appropriate application programs to generate, say, a spreadsheet, and to present it again at the receiver side. Therefore, it is hopeless to try to use a VM-based receiver to correctly interpret such compound messages.

5.3.3.4 Authentication

The implementation of business procedures without requiring hardcopy signatures necessitates the reliable authentication of the senders of messages created in the course of such a business procedure.

Sender authentication in the VM domain is fairly reliable, but it is still one of the drawbacks of SMTP that the recipient has no real proof that mail was indeed sent by the person named in the *From* header. This applies also to all mall received from the TCP/IP network via SMTP.

Only the local delivery mechanism in our AIX cell gives us currently fully reliable AFS authentication. When a local AIX user calls *sendmail* to send mail to another local user, *sendmail* calls the *zurimailer* program. When *zurimaller inserts* the mail as a new file into the AFS spool directory, this is done with the sending user's AFS token. Therefore, the sending user becomes the owner of the new file. When *zurimailerd* picks up the file for delivery into the recipient's system mailbox, it checks the owner of the file and inserts an *X-Afs-Authentication* header. If mail is received from the network or from an unauthenticated user, or if the mail was queued intermediately, the owner of the file is AFS' *anyuser (UID* 32766) and authentication is marked as *NONE.*

5.3.4 Remote Access to Personal Mailbox

- **via Home Terminal**

Many of our approximately 80 home terminal users with their PCs running OS/2 or DOS access the ZRL computing environment via secure call-back facilities and login only into VM and handle essentially only their mail there. This is not a good solution for the growing number of AIX users who receive their mail at the lab in the AIX domain. They can login via a separate path as remote LAN users with optional call-back. Then they login to our ZRL AIX environment, and are granted access to all services including Z-Mail. The only drawback they have to face are the slow response times caused by the limited bandwidth offered by the dial-up telephone channels. However, users with ISDN connections and PCs in the'486 class and above can work almost as if they were attaching locally at ZRL.

- **via 3270-like Terminal**

There is practically no difference between one of our users who has access only to a 3270-like terminal at some other IBM location during a business trip, and the home terminal users above using a 3270-emulator. Both can login to our local VM host and handle their mail there. For the Z-Mail users among them we have written a CMS front end based on REXX sockets which supports a minimal set of functions, so that Z-Mail users can access their mail via an automated TELNET connection. It should be understood that this front end is merely a survival tool for travelers to look at their mailbox which should contain only a few recent notes.

5.3.5 Problem Areas

5.3.5.1 Bulletin Boards and Fora

Z-mail allows access to folders in read-only mode. If a user 'deletes' a message in a read-only folder, this information is stored in an *exteMal index* file. When the user re-opens the folder, the message will be hidden as it is marked deleted. This feature allows a simple implementation of

mailing lists and bulletin boards. The Zurich Mail Bulletin Board is a directory containing a number of read-only folders accessible by all Z-Mall users. These folders are the system mailboxes of special dummy userids. These userids can act as a native AiX replacement for local VM FORA because any user can send mail to such a user-id to have it appended to a publicly readable folder. They can also be subscribed to mailing lists outside Zurich or even outside IBM. If several people at the lab are interested in the same information, it can thus be avoided that multiple copies of the mail are received. Users have the advantage of a clean personal mailbox. They can easily 'subscribe' and 'unsubscribe' by simply opening or closing the read-only folder. If they find a message important, they can easily copy it into one of their personal folders. It is also possible to shadow VM FORA in the Zurich Mail Bulletin Board. This is important to make local as well as some other FORA available to all our users who login to VM less and less frequently. A hook in the *zurimallerd* script reformats incoming FORA append headers for better readability in the bulletin board. The configurability of Z-Mail has allowed us to automatically use an appropriate template when 'composing' or 'replying' in a bulletin board folder. A new append is thus automatically sent to the proper address and contains the necessary tags for the TOOLNOTE facility.

In fact, the Zurich Mail Bulletin Board is a special implementation of a News facility. Though we do have an NNTP News server at the Zurich lab, many of our users find accessing the Zurich Mail Bulletin Board via the Z-Mail graphical user interface much more convenient than any of the common News readers.

The reason for listing our Mail Bulletin Board under Problems is quite simple. In its current form it is adequate for following only a few fora, otherwise its response times become too long. The problem is with the expectations of former VM users who are only willing switch to a new approach if it is demonstrably better than their old one. Presently we do have this nice solution unified with Z-Mail, but its performance is much worse than the equivalent mechanism in the VM domain where a user can keep track of hundreds of fora without noticing any access delays.

5.3.5.2 Additional Support of Other Mail Facilities

There are always users who would like us to support additional mail facilities, e.g. those that exist in OS/2, Windows, or as parts of application packages. Usually these users do not realize that the resulting support costs are higher than the mere installation costs. We are reluctant to start such activities for the following reasons:

- Basic mail support can be expected to grow with the square of the number of mail systems supported. As we cannot yet discontinue VM mail, the next system would be the third one to support. Calling the three systems A, B, and C, we would have to support all transfer possibilities, like AA, AB, AC, BA, BB, and so on. The total number of these combinations of two elements is n=(3)2 =9 Correspondingly, for four systems we would have 16 such combinations. This means roughly a duplication of our basic support effort with each additional mail system.

- Any mail system for professional use has to build on a reliable file system including reliable back-up. This is true for VM and for our AIX with AFS. Mail systems in the PC domain rely on their own special file servers. Currently we recommend that our OS/2 users keep their files in AFS where we offer adequate back-up facilities. It looks very improbable that we could also find a way to support the file system of one of these mail systems under AFS. Therefore, the actual support cost of one of these additional mail systems would also require separate support and back-up of its file system.

- Modern mail systems are usually not completely stand-alone facilities. There are ties to other applications and consequences because of three reasons:

    - A mall system requires a minimum set of ties to other applications. Examples are ties to address data banks and alias files, or to editors and templates for composing notes or fax

messages.
- All this work has to be repeated for every new mail system.

- In order to benefit from standard MIME compatibility, sender and receiver must use the same application supporting the transferred MIME attachment.
  - Permitting more than one mail system and one set of office applications would be counterproductive.
- A manufacturer may decide to closely intertwine all office functions including mall on top of a special file system, such that it becomes very difficult for a competitor to sell a specific office function product for that environment.
  - Permitting more than one mall system and one set of office applications would be "counterstrategic", and, hence, give rise to all kinds of difficulties.

5.3.5.3 User Mobility

We tend to assume that a given user will always use only one mail environment. However, this assumption is often wrong. We have already seen that Z-Mail users may occasionally be forced to access their mail via 3270 terminals. Similarly, they might wish to use at home a PC-based mall solution of their choice. In the previous subsection we have outlined the problems caused by supporting different mall systems for disjoint groups of users. Here we point to an additional and very difficult problem: Users may wish to switch, sometimes even on a daily basis, among different mall systems, and expect then not only to read or send a few messages, but would like to see their complete mail environment including all their mail folders and related information. This expectation seems unrealistic if users have to use or insist on using different mail systems for each environment. We have tried to do this for Z-Mall users switching occasionally into the VM domain. For reasons mentioned above even in spite of considerable support work on our side and personal user involvement to prepare the Z-Mail equivalent of a VM NAMES file, it takes long conversion times every time the mail system is switched. Even so, as mentioned above, not all Z-Mail features are implementable in the VM domain, so the ultimate result is not much more than a survival tool.

We would prefer the only sensible solution of standardizing a single mail system usable on all AIX workstations and PCs for ZRL, like the former VM mail. Currently, this would be Z-Mall, as there is now also a Z-Mall client available for the Windows and OS/2-Windows domains besides full emulators for X-Windows.

5.4 Printing

Printing is the second basic function needed by virtually every user.

Normally printers are attached locally to workstations. As not every user can justify having a personal printer, we have installed public printers within easy reach of all users. This allows almost exclusively high-speed Postscript printers to be used as shared resources. There are a number of users who can justify having a printer in their own office. These printers are also treated like public printers and can receive input from one of the four print servers of our AIX complex. The print servers do not only provide print job queuing, but also the automatic translation of input print streams including EBCDIC, ASCII, PostScript, AFPDS (Advanced Function Printing Data Stream), tersed or untersed, black&white or color. Access to the print facility is achieved via the remote printing protocol provided by the TCP/IP suite. The output stream ultimately sent to our printers is either encoded in PostScript' PostScript Level 2 or destination-specific.

This allows us to meet most expectations of our users:

- Printing operations are as far as possible the same for all applications we support.
- All printers can be addressed directly from all workstations, regardless of machine type or operating system. Even "hardcopy mail" becomes possible, at least when directed to people with their own printers.

- In agreement with the SSI philosophy users can print on any printer in the vicinity of the workstation they just happen to use. Additionally, users may specify for each of the two sites and for home a separate set of default printers (simplex, duplex, and color).
- Color printers can be utilized in the same way by all users, even if only very few of these special printers are needed at each of our two locations, and
- The concept of public printers has even been extended to home terminals running X-Windows. A11 public printers at ZRL can be addressed from there, and if the home terminal itself has a Postscript printer, this can also be used as a public printer while the whole home setup is working actively. All this is a prerequisite if a home user wants to run an AIX application remotely at ZRL, but to print results at home.

5.5 General Information Services

The third basic function needed by every user relates to all kinds of general information. We attempt to provide this via our computing services.

There are two kinds of information in our environment:

- "General information" which may be actively sought by our users: This ranges from local bulletin board announcements to all kinds of IBM internal information to public information, as it can be found via the Internet. Obviously, a description of our computing environment also has to be available.

- Passive information: Our users need to receive certain information relating to computing services automatically without their active intervention. For example, certain sets of users must be informed when a specific service has been scheduled to become temporarily unavailable. Otherwise they might start long-running batch jobs on a compute server which will be turned off before their jobs can terminate. Another example are messages directed to a specific user, like: "Your print job is finishing on printer x". Independently, this same messaging service has also been made available for sending discretionary messages between people, like "Please, call me asap at xxx" to and from any user logged in.

5.5.1 Searching Actively for Information

Over the years, general information services were developed in the VM environment. Various such services were provided by many independent groups and individuals. Most information was provided as flat files, but in some cases programs were also started to write information directly to the screen. ZRL information and programs were maintained locally, but not centrally. There were also corporatewide information and programs which were automatically distributed to all IBM-internal VM information servers. The system had two important advantages: It was very easy to set up for information providers because it could be based on simple flat files, and it was very fast. However it had several disadvantages:

- There were many local and IBM internal information providers who operated independently. Information was created not only by humans, but sometimes also by disjoint periodically restarted programs.
- A significant part of the available information was structured in a rigid decimal classification system. The problem was that users quickly learned to use the decimal classification of desired information as a fastpath to access it, e.g., "info 3.9.9". The resulting disadvantage was that users resented all the necessary extensions and modifications of the classification scheme to keep it abreast of current user needs and expectations. This disadvantage hides an even bigger one: A single stable classification system looks very enticing to a conservative organizer of classical information structures, but it cannot serve the needs of modern users who have to seek direct and simple answers in the huge, complex, and short-lived masses of sometimes conflicting information which is growing on a dally basis.

- The user interface was not standardized; even the function keys often had different meanings.
- Certain information was also routinely distributed for easier access as VM printouts by hardcopy mail.
- Information structures were not standardized.
- The information itself was not stored in one central data bank, but it was kept in many different places, like on the personal minidisks of specific information providers.
- No support was available to produce graphic features, colors, and images.
- By its very conception the system was limited to VM.
- Some information items were redundant. In such cases providers found it simpler or better to create a new package than to find and/or extend an existing one. This happened frequently, especially if an information was needed in a strictly formal way as an input to programs, and at another time and place the same information had to be presented to human readers. Besides the overhead created by redundancies, the main disadvantage was that this redundant information was almost certainly not updated synchronously.
- Keeping information up-to-date is a nontrivial requirement. Some items must be always current, like employee listings, or bus schedules. Other items merely lose importance over time, like local news or talk announcements. Other information is explicitly dated, like technical reports.
- As there were many separate and independent information providers, even at a local level no single person was responsible for making sure that all updates were made in a correct and timely fashion.

When we started the AIX environment, our first information service supplied an introductory description of this AIX environment for users. There we tried an approach based on hypertext and good search facilities hoping to avoid some of the disadvantages of the previous VM solutions. With the growing acceptance of distributed computing it became clear that all information services should be migrated from the host to the new environment.

The moving of all information services into the distributed computing environment offered a chance for significant clean-ups. Among several possibilities we looked for a new standard approach that would avoid most of the disadvantages listed above for the old VM solutions without introducing too many new ones. Finally we decided to use approaches and solutions as they were coming into existence in the environment of the worldwide web (WWW). Our main reasons were:

- Many of our users were, or would soon be interested or required to access, and sometimes even to post information in the WWW. All of these people would save time and effort if they had to learn only one standard approach.

- Many of our users have strong interactions with outside institutions and colleagues in Europe. The amount of traffic made it advisable to install a local gateway into the Internet via a firewall machine as shown in Fig. 4.2. This included a proxy server to support not only mail, but also exchanges of files. Given this setup, it was easy for us to support upcoming user requests for presenting selected ZRL information to outside parties in the WWW environment.

- All of the system-related disadvantages listed above for the VM-based information services can be overcome in the WWW environment.

- Following "Occam's razor" it was then the best course of action to follow the WWW paradigm for local information services as well, instead of selecting an additional standard.

  Besides performing the actual migration of our information services, the only additional cost for us was to add a local WWW-style server on the internal side of our firewall machine.

The three most notable disadvantages which new users normally find in the WWW environment are:

- Response times in the WWW are slow, especially if a new page of information contains much image material and if the link between the user and the WWW has only limited channel capacity.

  Experience has shown that this disadvantage does not apply to our local information pages, because response times within our cluster are sufficiently short.

- The information available in WWW is huge, steadily growing, chaotic, and changes its structure and availability in a seemingly erratic fashion. Nevertheless, users who have learned to cope with this environment find it very useful for their work.

  We can make our local information services much more user-friendly, because we are setting them up in a centralized fashion. This allows us to keep structural changes to a reasonable minimum and to make sure that all this local information is timely and valid.

- Information providers have to learn to supply information in HTML format. The learning effort to do this is shrinking, as an increasing number of word processors and desktop publishing systems are becoming able to create output in HTML format.

Several new requirements should be met in providing our local information services:

- Information must be ordered "naturally". The web makes it easier than our old VM menu system to structure various topics. Definition lists and others allow us to group and subgroup items belonging together. If users might expect a topic in another place, we can simply add a link to the same topic. We also try to keep the structure as stable as possible and to avoid fancy but unclear titles. We saw in other places titles like "Tour de Research", "Datastream", "CyberJournal" and "Research enclaves". Where would you look there for, say, job offerings?

- Information must be kept up-to-date:
  We consequently try to keep all information only in one place in AFS where we can easily update
  it, and derive other usages from there. Such a master can be:
  - a file:
    The shuttle bus schedule mentioned earlier is now kept as an HTML file, and it is sourced into other HTML documents via server includes.
  - a database table:
    Certain information in database tables, like conference room reservations, is obtained directly from the respective data base using cgi scripts, while other HTML files are regenerated only every night, like organigrams and staff listings.

  - a program and its data:
    Simple cgi scripts can call other programs and reformat the output to HTML. All our locally available on-line language dictionaries are made available this way.

- Old information must be eliminated from the system: We have various mechanisms to feed bulletin-board like information into the web. Simple cgi scripts can create HTML lists which allow an item to be viewed. The creator of such an item usually knows how long this information is valid. We therefore store the expiration date with the file name, and the cgi script shows only those items that are not yet expired.

Our workstation users are apparently very satisfied with the way we manage local information in the WWW environment. They have learned to search for specific information using key words

rather than some possibly outdated classification number, and usually find quickly what they are looking for. To them it looks like a fairly stable island in the middle of a huge, chaotic, dynamic ocean.

Unfortunately, there are still users and situations requiring the support of local information services in the VM domain. As the migration of the basic services to AIX is now fairly complete, we primarily export information from the workstation domain back into VM to satisfy these needs as far as possible, and as long as necessary.

5.5.2 Receiving Information Passively

There are several cases where our users should receive information without actively seeking it:

- Users of our computing facilities should receive relevant information about temporary bottlenecks or outages of service components so they can plan their own work accordingly. Similarly they should receive automatically generated operational messages, like "Your print job is finishing on printer x", so they can pick up possibly confidential printouts. Such messages are distributed using the Zephyr shareware package from MIT. These messages also appear on the screen of a previously unavailable user whenever he or she logs in again or unlocks the screen. All such messages are stored automatically in the mail directory of the recipient.

- The same Zephyr mechanism can be used to send short discretionary messages, like "Please call me asap at xxx".

- The need of management to keep their groups informed about upcoming events or new developments do not require novel technical solutions. Besides electronic bulletin board announcements there will always be personal meetings, as well as electronic and hardcopy mall.

- One of the potentially most useful services desired by many of our users is unfortunately not yet available: to filter efficiently the mass of information on the WWW to obtain only relevant information that may impact one's own work. Our users have currently only two options:

    - Ordering periodical literature searches following relatively coarsely defined interest profiles utilizing commercial technical abstract services, or

    - Monitoring more or less actively selected news groups, or specific pages on the WWW.

  In the near future we do not expect to be able to help our users very much beyond this. The problem looks more like a long-range research topic than a short-range support task.

5.6 Office Applications

The forth and last basic function needed by every user is some form of office support, like the generation and handling of letters, documents, reports, spreadsheets, or faxes.

At first glance it looks easy to support such office functions, because there exists a booming market offering a multitude of more or less suitable, but continuously improved and modified application packages. Indeed, a single user working alone has no problems to find a set of software packages satisfying his or her needs. Our situation as a support group is different. Even if it were possible to support all the competing application packages selected by our users, there would be no way to support teamwork among groups of users relying on different packages. Although many of these packages have a fairly similar look and feel for implementing fairly similar tasks, their data formats are sufficiently different to make conversions of personal files from one domain into another one nearly impossible. Usually little more than the ASCII character input

strings can be correctly moved from one domain to the other. Full compatibility is available at best at the level of postscript output files and printed hardcopy.

With AIX as our primary computing platform, we decided to support standard office tools in this domain. After evaluating different commercially available offerings we selected Applixware from Applix Inc. for supporting the majority of the office tasks.

The reasons for selecting Applixware included

- Integrated support for word processing, drawings, spreadsheets, data base access, and mall,
- A relatively easy to learn user interface, especially for users of other popular products for word processing, spreadsheets and graphics,
- A powerful high-level macro language (ELF) which makes customization easy, and
- One of the best sets of conversion routines to accept inputs from other application packages.

An important first step for us was to create a set of templates that allowed users to write letters, memos, and faxes with IBM letterhead. This included modifications of the standard Applixware user profiles to reflect our local preferences (metric, printers, languages, etc.)

Next we implemented modular linkage with our printing and mall environments. Here we decided not to support the Applixware mail subsystem, but to rely on Z-Mall as our standard mail system because of its superior capabilities.

While creating the templates, we ran into the need to repeatedly change many of them after feedback from our test users. As many templates showed only minor differences such as the style of the letterhead or the right/left positioning of the destination address, these multiple changes led to annoying inconsistencies between templates, requiring extra time and effort.

These difficulties led to the use of Applixware's ELF to assemble all templates from standardized components as required for each kind of document. Thus, a typical template may consist of a letterhead component, followed by a heading component, a body, closing and footing components, etc., combined into a single document. To modify a template, it is usually only necessary to modify a single component. This we can easily modify and then "recompile" all the templates that use it, and release them again to users in a clean fashion within minutes.

Based on our experience with ELF to build documents, we gradually advanced to the next step of offering as an optional convenience a dialog-box front end to help users prepare letters and various standard documents.

Today we offer users the ability to generate letters and faxes carrying the letterhead of their choice with many parts of such documents generated automatically. The ELF macro language allowed us also to establish links to the address listings maintained by different user departments and stored in DB2/6000 databases. These steps towards a convenient letter writing system have become well accepted; many users including secretaries and managers can now easily prepare and administer their own correspondence.
For the computer-based sending and receiving of faxes we had to select an additional package, because Applixware and Z-Mall did not offer adequate facilities. Currently we are using VSI-FAX for that purpose.

- Faxes are prepared in the Applixware domain and sent out via the VSI-FAX backend also installed in the AIX environment. We offer the same fax channel as a pseudo printer to support other applications, such as our desktop publishing systems Framemaker and Interleaf.

- Incoming faxes have still to be rerouted by fax coordinators, such as department secretaries. We hope that SwissTelecom will soon offer the necessary hardware and services, so that faxes can be routed directly via our computing network to their intended destinations.

By standardizing, centralizing, and imbedding our office applications support into our AIX computing environment, we are in a position to provide:

- Excellent support for teamwork and as well as for personal work
- Standard formats and structures for business documents
- Immediate propagation of improvements and bug fixes to all users
- Very high level of user support with minimal manpower, as all users use the exact same basic environment, and individual files can be viewed and fixed from a Remote site.
- All benefits of already available services, like printing, or backup.

5.7 Load Management

Introduction

By end of 1994, about 350 people had an AIX user-id. An adequate number of workstations and other RISC-systems had to be installed and maintained to fulfill the continuously growing computing requirements of our users. Figure 7 lists the increase in compute power for 1993 and 1994. The SPECmark89 values are calculated by summing up the SPECmark89 values of all RISC systems available during each week. Load management tries to assign the available resources to users in an optimum and dynamic fashion and to plan ahead for meeting future needs.

Approach

Three basic principles are considered when distributing workstations among users.

- Give each user the hardware he or she needs. Based on the accounting data collected in the past, the user's work plans, and the system administrator's knowledge of the user's activities, it is decided what type of workstation has to be given to a user. Heavy users are provided with faster workstations, whereas a slower machine or a PC running X-Windows may be appropriate for casual AIX users. Recycling of workstations-is another important aspect in our environment. A workstation that does no longer fulfill a heavy user's needs, and therefore has to be replaced, may constitute an upgrade for a more casual user. Thus the useful lifetime of workstations may be extended.

- Maintain a server infrastructure with a performance adapted the needs of the whole workstation cluster.

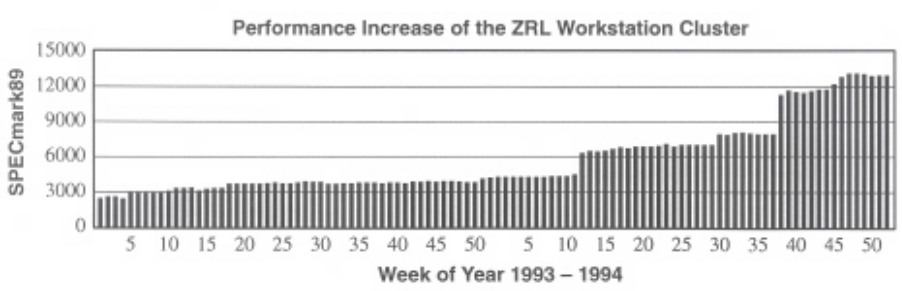


Figure 7

- Make use of idle cycles. It is known that the actual utilization of personal workstations by their owners rarely exceeds more than 10-15%. This is also true for our environment. Making the idle cycles available for batch job processing is a challenge addressed with emphasis in our environment. About two years ago, we introduced

LoadLeveler, IBM's batch job scheduling system for distributed workstation environments as our tool to make idle CPU cycles available for batch job processing.

Workstation Utilization
To describe the workstation utilization in our environment more precisely, we differentiate between four classes of workstations:

- Farm
  This set of dedicated compute servers includes an SP2 installed in the fall of 1994.

- Single-user workstations
  These are desktop workstations for very active users.

- Multi-user workstations
  Xstation servers or application servers are designated as multi-user workstations.
- Servers
  These are dedicated systems like file or mail servers. Direct user access to these systems is not possible.

For each class and week, Figure 8 shows the total available computing power and consumed CPU cycles by interactively started processes as well as jobs executed under the control of LoadLeveler. Again, to take into account the workstations' relative performance, the data are given in SPECmark89.

The approach to integrate all user workstations into LoadLeveler to make use of idle CPU cycles is quite unique. Therefore, the next paragraph describes this approach in more detail.

Experience with LoadLeveler
LoadLeveler's architecture is depicted in Figure 8. The *central manager* receives all the relevant system information. The *workstation info* data base contains the information to be entered by a system administrator, e.g. the definitions of when a workstation is considered idle or busy, or the type of jobs allowed to run on this workstation. The *workstation usage* data base is periodically updated and contains the current workstation utilization data gathered by a *status* daemon running on every workstation. The status daemon monitors the CPU utilization and the time elapsed since the last keyboard or mouse event. When a job is submitted, it is placed first in a central *job queue.* A *job description* file lists the minimum resources the job requires. These include the architecture the job should run on, the memory size needed, and the free temporary disk space or special software packages the job requires. Whenever there is a job in the central queue, the central facility determines, based on the information in the workstation info and status data base as well as the job description file, which workstation the job should run on. Once a job is submitted to a workstation, a local *control* daemon supervises the job execution. The control daemon suspends or resumes batch job execution based on the activity of interactive users.

First, we started using LoadLeveler to manage our compute farm, a cluster of about 10 RISC System/6000 models 550 and 580. After introducing Loadleveler, the farm's overall utilization increased from about 80 to nearly 100%.

After LoadLeveler had proven its usefulness for dedicated compute servers, we integrated all our workstations with the obvious exception of dedicated servers into LoadLeveler. We stated rules that whenever a workstation is interactively used, a possibly running LoadLeveler job is stopped and only resumed after the workstation has become idle again for a certain time period.

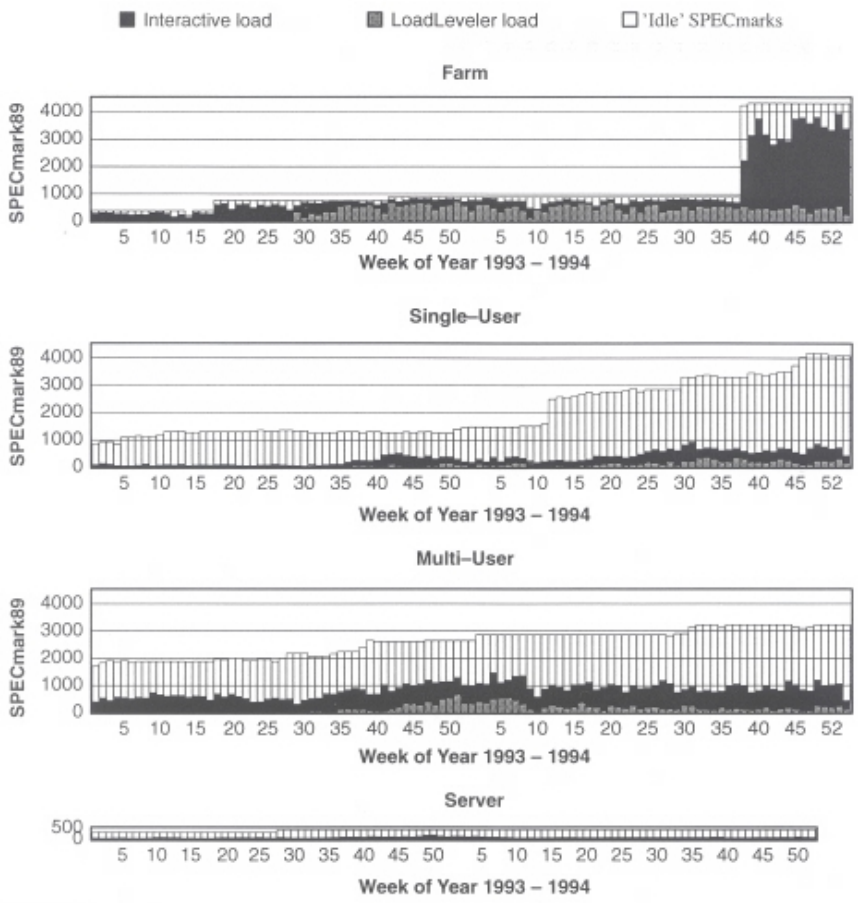
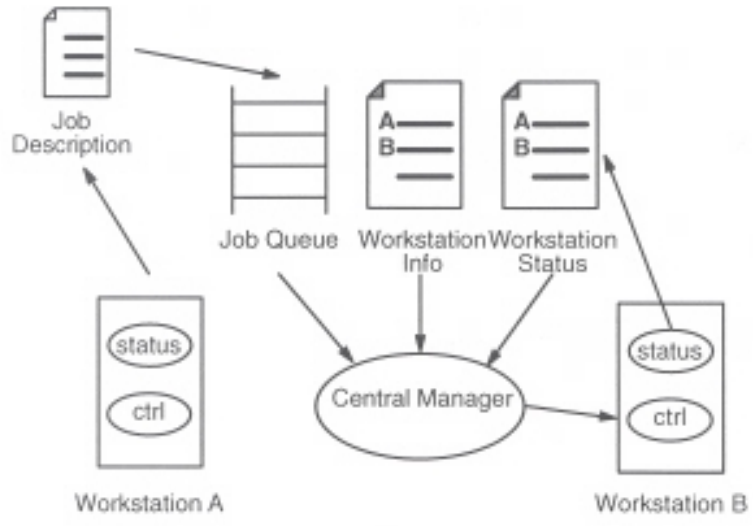


Figure 8



Figure

It has been shown that maintaining LoadLeveler in our centrally managed distributed computing environment is easy for the following reasons:

- Our Single System Image approach has the advantage that a user submitting a job to LoadLeveler need not bother about the availability of the needed files on the target machine.

- As every user is allowed to login to every user workstation, we do not have to deal with access restrictions.

- Every workstation may have its own profile stating under which conditions a machine is available for batch job processing. These profiles can be administered centrally, allowing the system administrator to take into consideration machine-specific usage profiles. Additionally, LoadLeveler does not have to care about the readiness of users to submit "their workstation" to the LoadLeveler regime. After all, given LoadLeveler all users can expect to obtain as much temporary computing power on short notice as they need.

Our experience with LoadLeveler is quite positive:

- Users accept that there may be another user's LoadLeveler job running on their workstation. Users may notice LoadLeveler jobs only when they restart interactive work after more than 15 minutes since their last machine interaction. Then any LoadLeveler job is stopped immediately, but the user's memory pages may also have been moved to paging space. Restoring the environment of the user back into physical memory may take at most a few seconds.

- Although in our environment the demand for batch compute power is far below the amount that can be delivered by LoadLeveler, the additional processing capacity provided by LoadLeveler corresponds roughly to the processing capacity of our dedicated compute farm without SP-2.

- In our environment, there are frequent demands to run memory-intensive batch jobs. This fact can be taken into account when buying new workstations. They may be ordered with more memory than needed by their interactive users, because this will allow memory-intensive batch jobs to be run whenever these machines are idle. The resulting configurations are highly attractive for our heavy interactive users as well as for those users who want to run batch jobs.

5.8 VM Host Access

Within IBM practically all users still need access to some data and services provided only via host systems. Although new data and services have been migrated into the workstation environment, there are several good reasons to see migration as a gradual process:

- There are important large data banks and information repositories that would cost more to move than to utilize as they are.

- Users are accustomed to working with host services. Many of them do not consider the transition to novel workstation-based services justified in view of incompatibilities between both environments. A classical example are e-mall services where certain features are not supported in the new environment, although many other new ones are added. Conservative users tend to see only what they lose, and they are not interested in new features that they had not consciously missed in the past.

- Our users and other people within IBM may need Remote access to our ZRL environment over the IBM network. The least common denominator is still a 3270-like interface.

Therefore, a 3270-type interface is an important element of our SSI. Normal IBM visitors without a local user-id are offered only such a 3270-type screen image for logging in at their home base.

5.9 Direct User Support

It would be a gross error to underestimate the importance and the efforts that go into direct user support. Providing the right amount of direct user support does more for user productivity and user satisfaction than spending equivalent amounts of money for additional hardware and software.

- Education
  During the introductory phase of our workstation services it was necessary to invite all users to attend basic courses. We can now essentially restrict our course offerings to special groups of users who wish to learn newly introduced application software.

- On-Line Help This seems to be the most important contributor to user satisfaction. Whenever users face a problem related to computing services they tend to expect immediate help because otherwise they "cannot continue" with their work. In our centralized and standardized SSI environment it is fortunately usually very easy to provide help from the desk when the user calls our hotline. Users first reach a trained operator who can resolve many complaints, and only the more difficult calls are passed on to one of our system administrators currently on hotline duty. We have found it much better to rotate this responsibility within the support group on a dally basis than to assign a specialist permanently to this job. In our model all members of the group get first-hand experience with selling our efforts to users and with recognizing issues that need fixing on a broader base.

- Introducing New Products The introduction of new products in itself is usually not as formidable a problem as tackling some of the issues around it. Introducing workstation-based e-mail may again be a good example. At first, experienced UNIX users had no problems with this at all. However, they soon asked for help in migrating their host mail repositories into their new e-mail environment, and requested us to add specific features, and so on. Users without UNIX experience wanted all these migration tools to be in place before they would even consider moving. We have learned that we must give appropriate tools and tailored help to migrate users from established applications to new ones, regardless of the promises of these new applications. Now, given the continuously changing situation in the software applications market, it becomes a sizable task to guide and accompany users in the never-ending quest for a computing environment with optimum cost-performance.

- Support for Representative User Groups In contrast to the promises of application vendors and the hopes of even experienced users, most efforts of groups of users to organize their computer-based teamwork fall, unless such efforts are set up by an expert. Looking back, it was quite clear that in a host environment user support had to provide a complete set of tailored screen images to support a specific application. Today it is usually possible for a user working alone to utilize an application package "out of the box". Unfortunately this is not true for setting up optimum structures to support teamwork. Therefore, we have started to work with a few groups of users in order to help them with their specific problems. Examples are the above-mentioned Personnel group with their specific security and privacy requirements, or a pool of part-time secretaries who should provide seemingly full-time support to their department and its manager. We hope that such pilot efforts will also help us continuously improve the services offered to all our users.

5.10 Compatibility of Various Desktop Applications

In accordance with our concept of setting up an Open Systems environment we had hoped to find a sufficient set of application packages which can either run on all our platforms of interest, or would at least offer clients that run natively on these platforms. This wish is not merely to comply with a philosophy, but to avoid the sometimes monumental or even hopeless tasks of converting data between application packages of different vendors. These conversions become necessary to support teamwork of inhomogeneous groups of users.

Although we tried to stay away from routine conversions of data between similar application packages, we had to invest quite some effort in establishing at least a modicum of corporate image for letters and documents generated with the help of such diverse packages.

As we cannot hope that the diversity of operating system platforms needed in a research laboratory will decrease significantly in the near future, we have to remain alert to find "standard" application packages that support as many of our platforms as possible.

5.11 Backup

AFS, the Andrew File System, is the production file system in which access rights and backups must guarantee that even critical data is always accessible to the owner and at the same time never accessible to any unauthorized user. An AFS cell contains a number of AFS volumes, which in turn contain files and data. AFS volumes are easily manageable areas whose size can be adjusted during operation. Because of the high number of files, the backup is oriented according to volumes. In early 1995, we had a total of around 180 Gbyte of file space, of which more than one third was regularly backed up.

Every week a full backup is performed of all volumes to be backed up. In addition, during the days in between, an incremental backup is performed on the file level, i.e. every file that has changed since the last backup is again backed up. The last incremental backup is always available on hard drives from an AFS read-only volume, where every individual user can access his or her files. Incremental backups from previous days are kept on tape.

The weekly full backup is done on tapes, which are kept for one month. The last full backup of the month is kept for the rest of the quarter. Every end-of-the-quarter backup has so far been kept indefinitely. A small data base is kept of the performed backups so that it is possible to find the data again. Our backup solution might at first glance seem manually simplistic for such a complex environment, but it has been our experience that because all tapes are written with standard AIX commands, one is always able to get to the information on the tapes no matter what happens to the database. This might not be the case with a shrink-wrapped solution where the data might not be lost but forever inaccessible because another part of the backup program crashed. Admittedly, our solution does not scale arbitrarily, but it has proved extremely reliable, and so far we have been able to restore everything ever requested.

Restoring anything other than yesterday's files involves systems administrators and reading from tape, which is labor intensive. Fortunately, such request occur only very seldom.

6 Solutions for Critical Issues in the OS/2 and Windows Domains

6.1 Overview of the Intel Environment at ZRL

6.1.1 Inventory

The total number of Intel-based machines in the Zurich laboratory is approximately double the number of employees. Their variety is growing and reflects the rapid change in the PC market.

As previously stated in earlier sections, our goal is to achieve an Open Systems environment. This is also true for the OS/2 workstations, which are exclusively used as clients in the open environment. This means that we have no pure OS/2 servers, and that all OS/2 clients rely on the AIX environment for their infrastructure. All networked PCs use TCP/IP like the AIX workstations. An important reason for insisting on an open environment is that it is a requirement for teamwork and collaboration between people working on heterogeneous computer platforms.

Our experience shows that this works well and according to expectations, but because of OS/2's nature (single-user system, no system-level protection etc.), it requires a computer-literate person to maintain the OS/2 systems. As most OS/2 users are not sufficiently qualified for this, someone

from DCS has to do that job, which means frequent visits to those customers. As our resources are limited we cannot fully support a large community of such users. We can only recommend them to switch to our AIX environment where we can support them much more easily.

6.1.2 Users of the Intel Environment

6.1.2.1 Types of Users

In the environment of a research laboratory all possible levels of user background and computer knowledge exist:

- Naive users who have no experience maintaining a PC and who know nothing about Windows or OS/2. At best they know how to use a few basic functions in the applications they use daily.

- Experienced users who do their work on the computer, but do not know how to maintain the system and, knowing their limitations, leave the system alone. From DCS's point of view, they are the ideal customers.
- "Know-it-all" users who are actually quite knowledgeable, who are able to install a PC at home and to run it. Unfortunately, in the office they also think they know better than the support experts, even in a networked environment. Consequently they experiment, break their system, and then expect instant repairs so they can continue to experiment.
- "True" PC experts who understand in principle how to install a networked environment for client/server computing. There are many such users in a computer research lab. Unfortunately again, they have no actual experience with implementing such systems in a real environment, nor do they know what it takes to operate such systems serving real users. It is difficult to satisfy these people because they do not accept that support people have to spend more time with users currently in distress than building beautiful technical solutions for tomorrow.

6.1.2.2 User Requirements

Our Intel environment has emerged based on particular user needs for which the AIX environment has proven to be not so well suited given the special requirements that may occur in a research laboratory. While the AIX environment is well able to satisfy regular office and computing needs in a very economic fashion, PC-type solutions are frequently needed because of interactions with a variety of academic, company-internal, and business partners. Likewise, joint research programs may impose special software requirements. Other examples are lab automation and portable computers, which currently seem to fit better into the realm of PCs. Beyond clear-cut business reasons there is also a strong emotional motivation of many PC users. Seeing all the feverish sales activities in the PC marketplace and the flood of novel software they want to have, they forget that for their own optimum productivity they would be better off with a reasonably stable environment than with a constantly changing environment that drains their time and energy.

6.1.3 Standard PC Configurations

6.1.3.1 PC Hardware

In 1994, there were active efforts to reduce the number of 386-based machines and to buy new 486 and Pentiums. In 1995, only Pentiums were bought, except for portables. There are still some old 286-basod machines used for lab automation, but even they are gradually eliminated in view of their relatively high maintenance costs. The pace at which new models keep arriving in the Intel domain is ever increasing, which makes streamlining the inventory extremely difficult. There seems to be only one survival strategy for our support group: Whenever a set of new machines has to be bought, we buy as many as possible of the presently best machine available with the biggest disks affordable, with as much memory that one might need for the next five years, and with all the additional features which might be needed. Then we keep buying that same model for as long as it is available from that manufacturer, and resist ordering "insignificantly" changed models. Usually, these "insignificant changes" cause enormous additional effort and anguish later. Based on the above we recommend to our users only three standard PC configurations: a current style laptop and a deskside PC for office use, and an expandable server model for lab

applications. In early 1995, all machines had to have 32 MBytes of memory and more than 750 MBytes of diskspace. It is not always easy to convince some of our users that they should order, in spite of tight badgets, such "luxury" models with more memory and diskspace than they consider absolutely necessary. However, again and again, we see that a top-of-the-line machine is the best solution because it survives the longest. Not buying top-of-the-line, shortens the useful lifetime of the machine, with the consequence that one conducts selective upgrades usually with unsatisfactory results. We still have to do this occasionally, but we nearly always regret it. It is a lot of work, yet is not what we want it to be, and, additionally, the stability of the upgraded machine is weakened for the rest of its lifetime.

6.1.3.2 OS/2

At given point in time we deploy the latest version of OS/2 together with selected communications products and desktop applications. In March 1995, we installed TCP/IP, Communications Manager for VM host access, Exceed/W as an X-server for UNIX/AIX access, and LAN Distance for Remote access of portables and home workstations via analog or ISDN connections. Applications offered were Lotus Smartsuite, Framemaker, CorelDraw and Winword, all running under Win-OS/2.

6.1.3.3 Windows

On older machines an in special cases we have to install DOS and MS-Windows 3.11. For this we also provide TCP/IP and X-servers in the default setup.

6.1.4 OS/2 Systems Administration

6.1.4.1 ZRL PC Approach

In accordance with our Open Systems approach we try to give PC users the feeling that their PCs not only give them access to all the core services on AIX, but satisfy their more special needs as well. Core services were outlined in the previous main section on AIX and include mail, printing, and info services.

We try to limit specific PC applications to run only on the individual user machines themselves. We do not provide other networking protocols beyond TCP/IP, or servers not running under AIX.

An important problem area for us is the time we have to spend at the user's PC for installation, maintenance, and trouble shooting. Some of our approaches will be discussed in the following. While they help reduce the total service time per PC, we have not yet found a fully satisfactory solution to bring total service time for PCs down to that of AIX workstations.

6.1.4.2 Initial Basic Installations of OS/2 and Windows

For initial installations of OS/2 we are using CID (Configuration, Installation and Distribution), which is a standard OS/2 approach that permits the installation of applications via CD-ROM, network or hard drive, thereby eliminating the cumbersome task of installing an empty PC from diskettes. We have extended the CID procedures to a point where complete machine installations can be made totally unattended. As long as an administrator prepares a few necessary parameters (network addresses etc.) a technician/student is able to perform an installation by simply inserting two diskettes subsequently into the PC, and typing the name of the PC. Then the PC will perform the rest of the installation automatically. The two diskettes contain minimum OS/2 support plus enough network software to build a connection to the CID server and then to load the desired software onto the machine.
In addition, we have also been using snapshots, or images, of frequently used machines. This means that we first install one PC of a common type and perfect its installation "by hand". A copy is made of its complete software, which can then be replicated onto other machines of the same kind. "Of the same kind" is key here, and in an environment of many different types of machines this convenient way of installing machines is not possible. Just a few different combinations of display chips, hard drive types, and network cards multiply into more images than one has time to prepare unless an image can be installed on many machines of the same kind.

The same method of using two boot diskettes and images is also used to install DOS or MS-Windows where needed.

6.1.4.3 System(S/W) Upgrades

As mentioned above we try to avoid hardware upgrades by always buying the best available machines. For software this is more difficult because new releases appear in short intervals. The best we can do in this respect is to launch a major new release as soon as it becomes available and keep it running until the next major release becomes available. We provide a standard set of operating system plus applications, as described above in the section on standard configurations, which we install and let run, with minimum alterations, until we have a new standard set available. This implies that incremental upgrades are not well accommodated. Only completely new installations are provided. This is of course a controversial decision, because it is easy to imagine users who, after having received their workstations, perform a number of customizations which will be lost when the next supported upgrade comes along. The alternative would be to fully support users with individual upgrades. This, however, would lead to unpredictable adventures; experience shows that this approach would place an unacceptable load on the support group without guaranteeing user satisfaction.

Our approach of reinstalling complete systems rather than providing incremental backups does not only simplify support, is also discourages users from reinstalling old applications which are no longer needed. We backup all user data before reinstalling machines. It is at the discretion of the individual user how much of these data and old programs he or she wants to reinstall.

A problem limiting the individual freedom of upgrading may arise with certain applications where, when one person upgrades, everyone else in the same working group may have to follow suit in order to avoid incompatibilities. This is another disadvantage which occurs only in personally administered systems. Obviously, a centrally administered cluster of workstations like our AIX environment does not have this problem at all.

6.1.4.4 Installation of Applications

Generally there are two approaches to provide PC applications in a networked environment. Either they can be run on a Remote server, or a local copy of the program is installed from a server. As running Remote PC servers would also have required us to support other transmission protocols besides TCP/IP, we have decided to support only the second approach. As an increasing number of our users need to be mobile with their laptops, we provide the facilities to install applications locally. The application packages are available from an NFS partition on an AIX server. With our open systems strategy we can simply rely on the AIX infrastructure. The installations are menu-driven: one starts a menu, selects the desired application, and it will be installed from the NFS server over the network. Certain applications need licenses; this is managed via a special NFS export list which allows downloads only for authorized machines.

6.1.4.5 Remote User Aid

While we are almost always able to provide Remote help for users in the multi-user environment of AIX, this is usually not the case in the PC environment. Given its constraints we have found a limited solution to provide at least some Remote help. The standard install images contain a small executable which permits the user to temporarily open his or her machine to a Remote system administrator using a telnet daemon with a password known to the system administrators. A few additional AIX-like tools, including ps, allow the administrator once logged in to look at files, to see which processes are running, to stop undesired processes, and to start desired ones. This access facility would be highly insecure if it did not work only when a user explicitly asks the system administrator to run a Remote check-up on the PC. While this is in fact a very primitive tool, it helps, if nothing else, to decide whether we have to visit the machine personally.

In the past year, several new tools have been released to manage OS/2 systems remotely, such as DCAF, Netview/DM, Netfinity, and Systemview, for none of which we have had as yet the resources to test and launch in our environment.

6.1.4.6 Central Configuration Inventory

The executable described above also contains a procedure that uploads a few files every time the PC boots. Those files are config.sys, os2.ini, os2sys.ini, and a file containing the present hardware configuration of the PC produced by QSYSTEM, which is public domain "IBM employee written software". These files are backed up every night, so that several recent copies of them are available. The two ini-files are mainly used as backup for users who have had a system crash and lost their desktop. The config.sys backup is frequently used to help those users who "played" a bit too much until they cannot get their machine to boot again. For the systems administrator, it is also used to see whether the user played at all, and which additional applications were installed. The hardware-configuration is perhaps the most useful file for the administration, because the collection of these files from all PCs constitutes a detailed and current inventory. One can see how many machines are installed, who owns them, how much memory they have, whether machines are running out of hard disk space, or when one needs more memory, which chips should be ordered. We have repeatedly been happy to have this file centrally stored for reasons we did not initially anticipate.

Finally, we can also see when a machine was last booted, and if it has not booted for weeks, one can expect that something must be wrong, or that the machine was removed.

This central repository of important files is probably the best one can do in a research environment like ours. While it would be convenient for us to protect the machines completely from user alterations, this would be unacceptable in our laboratory. The best we can do is to track what is happening and maybe to offer a solution when something goes wrong.

6.1.5 System Components and Solutions

6.1.5.1 Access to the AIX Environment

AIX access is very important regarding our strategy of having AIX as the backbone system. In order to benefit from the savings of providing only one infrastructure and one backbone, all users must have access to it. Examples of services we provide only once are: a shared diary system (Xdiary), a messaging system for person-person and machine-person interactions (Zephyr), E-mail (Z-Mail), facilities for all users to set up batch jobs, access to Internet and WWW, and general local information services.

PCs can have two different links into AIX, either by emulating an X-station under OS/2, or by accessing AFS via an OS/2 client. As X-servers we provide both Exceed and PMX. Exceed is the wellknown X-server from Hummingbird. It exists in both a Windows and an OS/2 version. We have found that the Windows version is superior to the OS/2 version, even under OS/2, with the only drawback that the Windows version only works in Win-OS2 full-screen mode. PMX is an IBM product and works well under OS/2, but not under Windows.

An X-server is the simplest way from a support point of view to provide a full SSI environment to PC users. This, of course, also applies for home users who can access the AIX-SSI environment via LAN Distance plus an analog modem connection, or an ISDN connection.

6.1.5.2 File Servers

As pointed out above, our philosophy is to have a variety of clients, but only one basic infrastructure. Therefore, the Andrew File System AFS which we use for AIX has to serve also as the central data repository for OS/2. This rules out otherwise quite interesting approaches like NetBios or the OS/2 LAN server. The AFS environment offers an experimental OS/2 client, which we found extremely useful. It gives our OS/2 users not only access to secure and backed-up file space, but also to a global file system shared with AIX clients. For users who must keep their data locally, we still recommend that they use AFS at least for data backup.

6.1.5.3 Printing Services

In our Open Systems environment it is natural to use Postscript for printing. As outlined in the AIX section on printing we utilize almost exclusively public Postscript printers which are within easy reach of all users. These printers are connected to the network over so-called LAN boxes, and do not need a local server next to them. The four print servers controlling them are located somewhere on the network. These print servers are also available to the PCs, which can send print jobs to them via TCP/IP. We tried to deploy a few printers with individual PC print servers, but found this solution very unsatisfactory because, whenever anything goes wrong in this server/printer combination, an expert has to visit it and fix it.

Consequently we have tried to avoid locally attached printers, especially non-postscript ones. Local printers are installed only for users who have a real business need, and who can usually help themselves with their setup if something goes wrong.

An important reason for using AIX print servers is to be able to, upon completion of a print job (which is recognized by the respective print server), send a message to the user, irrespective of the system platform, that the job has finished. This facility is a requirement for the printing of confidential material, which must be picked up within half an hour of completion.

Printing from OS/2 is done directly via the AIX print servers, not via a LAN server, which is a common solution in some environments. This is possible because we install TCP/IP also on all OS/2 machines, and set up printing to be performed via the TCP/IP command "lprmon". Lprmon is a process that runs in background mode and forwards all print jobs onto a certain "lpt:-port" on the Intel machine, to a certain printer on its specific print server. It is of course possible to have several lpt-ports configured and to print to many different printers. The available TCP/iP command also allows a user to cancel jobs and query the queue status. The above of course only applies to local print job on the OS/2 machine. Shouid a user be logged in via an X-server, as described above, printing from that environment works completely within AIX.

6.1.5.4 E-mail

Our basic approach is to always have e-mail services fully supported under centralized control in the AIX and VM environments. PC users have to log in there through an X-server or a host emulator to access their e-mail.

In 1994, a Windows client for Z-Mail first became available which also runs under Win-OS/2. This allows the advantages of a local and a centralized solution to be combined. While the Z-Mail clients can run locally, incoming mail is still stored on a central server, and can be downloaded to the client using the POP protocol. If all mail folders and even the mailbox file are kept in AFS, one is free to switch between doing e-mail either on a PC or through AIX without losing anything.

Even today, Z-Mail seems to be the only mail system that offers clients for nearly all important platforms, and that is compatible with SMTP and MIME.

6.1.5.5 VM Host Access

In spite of the growing importance of distributed computing, host services still play a very important role within IBM. Hence, host access facilities are still considered necessary for most workstations.
Our PCs use the standard product Personal Communications 3270, which works well in the TCP/IP environment.

6.1.5.6 Backup

Experience shows that most workstation users do not care about backups. They do not expect data losses, or even a disk crash in their own nice computer. While this may be a tolerable attitude for a private user, in a commercial environment backups are mandatory.

In our environment we have decided to take two separate approaches for avoiding loss of important data and for restoring PCs after a catastrophic failure like a disk crash:

- Data files for which backup is considered necessary should reside in AFS, or should be copied into there. For data stored in AFS we can guarantee reasonable backup, security, and confidentiality.

- Restoration after a catastrophic failure requires a complete backup of the hard disks. For this purpose we have set up an ADSM server with a tape robot at both of our sites, each capable of holding about 1 Terabyte of data on 8-mm tapes. Four tape drives reduce the bottleneck of writing to the relatively low-speed drives. In addition disk caches reduce the peaks of arriving data at the robots if multiple workstations are performing a backup in parallel. An automated scheduler attempts to prevent such situations to reduce network and write congestion by assigning individual backup start times for each workstation. No user interaction is required to trigger the backup. A fixed schedule is not possible for computers that are mobile or are switched off for longer time periods. The scheduler tries to reach these machines and, as soon as a contact can be established, the backup operation starts. In our environment a typical backup is done incrementally, meaning that only files that have changed since the last backup are stored on the tapes. Generally, more than one version of a file is available for restores. Files that have been removed are only marked as inactive but are kept on the tapes until an expire process removes surplus backup versions.

  A PC can be completely restored after a hard disk change in one to two hours depending on the processor power and network situation. We are currently supporting backup for about 100 PCs. Our present approach appears to scale up to 1000 PCs.

  ADSM can also be utilized for discretionary backup of individual files. Some of our more experienced PC users have started to use this capability to their satisfaction.

6.1.6 Desktop and Other Personal Applications

Computer users agree readily that "Everyone should use the same application so that we can communicate, profit from each other's knowledge and need only streamlined support". The problem lies in agreeing on one streamlined set of applications.

We have tried several times to establish recommendations for a standard set of applications, either based on technical reasons or on committee decisions together with user representatives. We have learned that, at least in a research laboratory with all its external commitments and constraints, it is a nearly impossible task to enforce such a standard set of PC applications. While everybody agrees that one should look for applications that run on all our workstation platforms, there are many groups who absolutely need just one of those applications which we had hoped to rule out. As a support group with limited resources we cannot hope to fully help our users with this seemingly unlimited range of special applications. We have adopted the strategy of providing, besides the AIX-based applications, all PC applications that are widely used at ZRL, but with a varying degree of support. Fully supported at this point are: Framemaker, CorelDraw, the Lotus Suite, and Winword. In all other cases we would still buy the applications centrally to keep track of licenses. However, we only make sure the latest versions are available, know how to install them and distribute them to users who have a business need for them. Any further support is a user responsibility. They may call external help-line numbers or ask their colleagues, but our internal help desk will not get involved.

This strategy has been surprisingly successful, and most of our users are satisfied with it.

6.1.7 Off-Site Computing

6.1.7.1 Off-Site User Requirements

Portability and computing anywhere are becoming an increasingly important issue. For many years we have had a home terminal program, where employees have been able to take a PC

home and access the office via an asynchronous connection. Portability is simply a generalization of this established home terminal program. For off-site computing let us examine a variety of scenarios in which the user is connected through networks of various qualities.

We have:

- the disconnected user working only with locally installed applications and data,
- the "Remote network" user attached to the Internet with a connection to his or her office LAN (for instance an IBMer on a business trip to another IBM site),
- "on the road" connections via an asynchronous dial-up telephone line, e.g. from a hotel,
- "home" users attaching via a reliable 64 kb/s ISDN connection.

In all of the above cases, the user can be one of the following types:

- an AIX SSI user and who does not rely on a local PC environment,
- a PC user sitting at a PC that is not his or her usual environment,
- a PC user sitting at an "other" PC, with a familiar environment, but needing the files available on his or her office PC, and
- · a PC user using his or her portable office laptop.

6.1.7.2 Access Possibilities

Remote stations would usually log into a host or server at the desired destination. Together with the LAN project we have decided to use LAN Distance, an IBM product which attaches the Remote PC to the LAN like any other PC back in the office, with the obvious exception of the speed at which the PC can communicate with the LAN. The major advantage of this approach is that everything works just as is does in the office, especially in the case of the portable office laptop. In our environment LAN Distance supports all common modem speeds up to 2-channel ISDN connections.

The link speed has a direct impact on the optimum working style. With a slow modem connection one would most probably work with local applications, and exchange data only as necessary over the link. With an ISDN connection one can comfortably work also with the X-server in the SSI environment.

7 Results and Conclusions

7.1 Main Result

We have approached the original goal of the DCS project to establish and maintain a distributed computing environment that satisfies our users as well as our management. Users have largely accepted our two key strategies of

- achieving simplicity at all levels through centralized control of distributed computing resources, and
- providing users as far as possible with fully supported computing environments.

In early 1995, most users including "UNIX experts" seemed to be fully satisfied with their environment, whereas there was still a fraction of host and PC users who are not yet fully convinced that "there is no better way".

7.2 Estimated Cost Benefits of Our Centralized Approach

1. *Savings in working time over our total user population exceed most probably 10-20%.*Experience shows that it takes several specialists to master the intricacies of setting up an environment with hundreds of workstations and keeping it up-to-date . Even if one disregards the time those specialists have to spend supporting individual workstations and users, the remaining workload is probably still too much for the equivalent of two or three people. Obviously, a single user cannot achieve this level of expertise. Many PC users at ZRL could and probably have justified (at least to themselves) devoting 10-20% and more of their working time to "improving" the computing environment of their own PC. They simply overlook the high probability that an integrated environment designed and tested by a group of experts will permit much higher productivity gains than a patchwork of individual solutions, which may well serve users individually but is not optimized for teamwork on a larger scale. Savings of the working time of managers, their staff, and their "experts" in planning and ordering new hardware and software will also be substantial, but are very difficult to quantify. As all system planning, establishing of standard configurations, and ordering is done centrally by DCS, line managers usually simply have to pick from a list of standard configurations whatever their group needs within given budget limitations.

2. *Savings in hardware and software investments are estimated to exceed 10-20%.* The estimate of 10-20% is based on our experiences with centralized planning of workstations and related equipment, centralized ordering of most hardware and software including floating licenses, installation and maintenance, the necessary inventory of spare parts, and a centralized administration that endeavors to make optimal use of the available resources. Additional savings become possible in a cluster of workstations running under a multi-user operating system. All our idle workstations are available for any user to run batch jobs under LoadLeveler. Although this capability has not yet been used extensively, it has already saved about the equivalent computing power of 20% of the "better" workstations. As some users have a seemingly insatiable appetite for conveniently accessible computing power, there is room for further growth considering that the usage of personal workstations and PCs seldom exceeds 30% over time.

3. *Manpower savings were achieved in the Support Group through a streamlined environment and support from "the Desk".* The support group would have needed several additional members, and could still not have achieved the same level of service, if we had not been supporting but one file system (AFS), the Single System Image (SSI), a single mail server and only one info server in the AIX domain.
   Although all workstations must occasionally be visited by support personnel, we found that the time spent with PCs is several times more than that with AIX-based systems. There are three reasons for this:
   - Their single-user operating system limits problem diagnosis and correction from an on-line help desk, and requires personal visits to the PC.
   - Although graphical user interfaces can be very convenient for experienced users, they become very awkward when the help-desk person tries to instruct naive users over the telephone on how to diagnose and fix problems in their systems "by simply clicking with their mouse"
   - Users can inadvertently modify their local systems in ways that would never happen in a centrally controlled environment.

7.3 Conclusions

- Close cooperation between users, management, and DCS has led to a satisfactory and continuously improving computing environment at our laboratory.
- We confirmed that there is no truth to the statement that only computer scientists can happily work in an *IX environment. We have converted a few groups of administrative users from their old PC environments to AIX-based applications. Their learning time was minimal, and they have been very satisfied, especially about the ease of teamwork without impacting the high level of data security they need.

8 Final Remarks

The present paper is a retrospective description of a successful implementation of a centrally managed, AIX-based distributed computing environment. Since early 1995, two major developments have strongly impacted the way I/S services are provided and used in the IBM Zurich Research Laboratory:

- A marked shift towards new Intel-based portable IBM ThinkPad machines, which are now being used not only for computing at home, on the road, and to access the I/S infrastructure from outside the laboratory, but together with docking stations as office computers.

- With Lotus Corporation now being part of IBM, Lotus Notes is very rapidly becoming the standard multi-purpose system for supporting e-mail and teamwork in general within IBM. Lotus Notes is platform-independent, and its elegant database replication function makes it ideally suited for cost-efficient Remote access to the infrastructure. Lotus Notes is also well suited for data sharing among local users and within the organization, or even with external organizations. It is actually a new paradigm that will transform our way of working with computers. Obviously, the Lotus Smart Suite office type applications will also play a central role in word processing, business graphics, spreadsheeting, etc.

These changes cause a noticeable shift from *IX-type workstations toward PC-type machines, which increase the amount and the complexity of our support efforts. Our centrally managed AIX system, as described in this paper, was demonstrated to be a very solid, minimum-cost backbone for providing distributed computing services. The new paradigm featuring Lotus Notes and IBM ThinkPads definitely requires more support, but has begun to increase the personal efficiency and convenience of our users.

9 Acknowledgments

Our special thanks go to past and present members of the Distributed Computing Support project, especially R. Camm, R. Larsson, B. Kopelman, B. Lassy, H. Neuwirth, S. Wood, and S. Zanini. We also appreciate the help and cooperation received from W. Frei and P. Sturzenegger of the LAN project, and from colleagues at other IBM laboratories, especially D. Singer (Almaden), G. Puhak (Yorktown), and J. Biseli (Austin). Thanks are also due to the laboratory management, who established and fostered this project, and to the users who relied on the evolving support efforts, and who gave us not only valuable feedback, but in a few cases even contributed directly to our efforts.

10 Recommended reading

In addition to the large amount of standard product-related manuals and literature we want to draw

attention to the following three references:

[1] S. Garfinkel and G. Spafford, *Practical Unix Security* (O'Reilly & Associates, Inc., 1991)
[2] Elements of AIX Security R3.1. IBM GG24-3622~1 (1991)

[3] Elizabeth D. Zwicky, *Are We There Yet? Evaluating Your Site's System Administration.* http:/lwww. usenix.o rg/sage/lisa/lisa8/invited-talks. html