**AIX Enterprise/Scalability White Paper**

8/06/94

Submitted and approved by the UNIX Cluster of Share Inc.


Chair:

Richard Alexander
RPI - ITS
Troy, NY 12180
(518) 276-6785
alexar@rpi.edu

Editor:

Pete Nielsen
UCLA/OAC
5628 MSA
405 Hilgard Ave.
Los Angeles, CA 90025-1557
(310) 825-2921
csyspcn@oac.ucla.edu

Cluster Manager:

Harold Pritchett
University Computing
Computing Services Anex
University of Georgia
Athens GA 30602-1911

**Executive Summary:**

It has often been observed that the price/performance of single CPU RISC boxes is doubling every 12 to 18 months. The systems themselves and the uses to which they are put are rapidly scaling into the realm that was previously considered "enterprise computing"

This paper identifies several areas where such growth may "break" the current system, and where lack of infrastructure in the current offering may make AIX inappropriate for deployment as an enterprise solution.

The focus is on a single box, running may concurrent users. Cluster and parallel concerns are not addressed.

1) **Scaling issues.**

There are several structures and algorithms in traditional UNIX which are already becoming so slow that they are being replaced at many customer sites with home grown alternatives. Others work now, but are inadequate for the types of work that our users which to run.

2) **Infrastructure.**

It has been observed that because of it's historical background UNIX is the result of an accretion of the work of disparate groups of enthusiasts. The result is that system monitoring is done inconsistently if at all. We need the tools for accounting/tuning/capacity planning that one associates with an enterprise level operating system.

3) **Security, Integrity, and Reliability**

As the systems become large so does the cost of losing data or having proprietary information compromised.

We feel IBM should make a statement of integrity for AIX, and accept integrity APAR's against it, as they do for MVS.

The "standard" model of UNIX security is inadequate for our needs.

We understand the direction of building systems on commodity hardware, but feel that HACMP and fail over strategies are inadequate to provide the level of reliability at the application layer that we require.

1) **Scaling issues**.
We are often told that the cost performance of AIX systems is doubling every 12 to 18 months. The high end of IBM's offerings are already comparable to yesterdays mainframes. As the capabilities of these systems grow, we are concerned that the AIX operating system not run into a wall due to its small systems heritage.

Scaling issues have been a long-standing topic of discussion between IBM and it's customers with respect to AIX/ESA. As customers we hope that this corporate wisdom is transferable to the Power arena.

a) ***/etc/passwd.***
One of the participants in this discussion had 26,000 users on one system.Many have found that in order for password changes etc, to be accomplished in a reasonable length of time, they have had to bypass the supplied tools and directly manipulate the password structures. Several of the participating installations had gone to using a database to contain the password information, and generate and distribute the relevant files from that controlling source.

b) **that may not beThe base OS needs a thorough review of algorithms appropriate in a "large systems" context.**
Subsystems within AIX which deal with process scheduling real and virtual memory management, and I/O handling are the first likely areas of improvement. When properly configured we expect that a large AIX system to run thousands of simultaneous processes which exhibit a vast variety of load on the system. We would like to be able to run both large cpu intensive jobs, I/O intensive jobs, and interactive work and have all receive reasonable levels of service. The site must have tools to implement installation policies about which tasks get what levels of service.

On AIX/ESA, an improved scheduler has recently become available. Initial reports are favorable. Because this scheduler represents a good start, we recommend that IBM consider using this scheduler for AIX4. Of course CPU scheduling is inter-dependent on real storage management and virtual memory management. Feedback loops need to be implemented among these sub-systems to create a site-tunable environment. On a system with large, heterogeneous load, we need a greater degree of control over where system resources are allocated.

Under the general concern of lack of Scalability, We are also concerned by interlocks which become a bottleneck on a busy system. One of the participating installations had a system with 1300 simultaneous users, and found that because many of their tools and packages used explicit paths to access files (As is recommended for security reasons.) the lock on the root inode became a critical bottleneck in the system.

c) **Resource Limits.**
Standard UNIX controls resources on a process basis. While controls are needed at this level, they are also needed at the "session" or "job" level as well as at the user level. The proposal in the current draft of Posix 1003.1a describes a beginning in adequate resource limits.

d) **Checkpoint/restart.**
One large systems feature that is needed in AIX is the facility to checkpoint a user "session" or job and then restart it at a later time, either on the same processing node or a different node. While there are challenges to accomplishing this, several UNIX implementations have succeeded (Unicos). Posix 1003.1 has a checkpoint / restart proposal in draft form. The need to migrate work will be a continuing issue in "large" UNIX systems.

e) **File size requirements.**.
Some users at many sites have single file size requirements much greater than 2 GB. AIX/ESA provides a solution to this continual issue. Perhaps it is appropriate to AIX4.

f) **Global system limits.**.
Many global system limits are historical and do not control the system in the way a large site needs. An obvious example is maxusers. Others examples may be developed.

2) **Infrastructure.**
We are concerned that tools and capabilities that we have come to take for granted in large enterprise operating systems be available for AIX.

a) **Openness.**
Although the individual systems are getting larger they are increasingly found in complex heterogeneous networks. It is no longer sufficient that a system perform reliably as a stand alone resource. It must now be able to interoperate both as a client and a server.

We would like to see OSF compliance. In addition we feel that the information flows between clients and servers, and data storage formats, as well as API's be documented and made available for system services.

An example is the information flows between the pieces of LoadLeveler. If we wish to use this IBM distributed job scheduling system, it is not enough that IBM provide implementations on the most popular platforms. We must also be able to implement compatible function on non-IBM supported platforms that happen to be critical to our environments. To do that, we need to know the information flows, and we need access to any informational resources (such as the SP1 resource manager) which LoadLeveler may use to provide its function.

b) **Diagnostics, performance, and capacity planning tools.**.
We need a comprehensive set of tools to diagnose problems, to tune our systems, and to predict future resource shortages for capacity planning purposes.

This need is especially acute for AIX, as so many of the common UNIX measurements are inaccurate, wrap too soon, are measured at inappropriate intervals, and are often interdependent.

At a recent SHARE a UNIX tuning expert observed that historically, UNIX was a collection of the work of independent enthusiasts. Because of this system monitoring values were implemented at the whim of the subsection developer. It is not uncommon for a value to measure something other than it's name implies, and or for it to be calculated and/or collected inconsistently across subsections of the kernel. If this is true of AIX, then it needs to be fixed.

We need to be able to do accounting, tuning, and capacity planning on these systems.

c) **DCE/DFS.**
We would like to see these included as part of the base operating system. If DCE/DFS is inappropriate for smaller systems, then it should be fully integrated into the medium and larger versions. We want third party vendors and all parts of IBM to be able to assume that they are an available base on which to implement function. As long as DCE/DFS are optional, most vendors will not depend on them.

As an example of what happens when this is not the case. On SPx machines IBM's LoadLeveler supports the Andrew file system, but IBM's PVMe does not. If DCE/DFS cannot be assumed to freely available on all systems, then this will continue to happen.

As we move to Kerberos etc. for security, we cannot afford to have the lack of "kerborized" software be a reason for maintaining the old methodologies.

The current DCE/DFS product modifies neither standard login, ftp, nor telnet. The current state is that every user will login to the AIX system and then, separately, authenticate him/herself. This is not a seamless presentation of the environment.

d) **Removable Media Support.**
The discussion was primarily about tapes, but applies equally well to CD Roms, or other removable media.

i) **Standard Label support.**
We need to be able to know that what has been mounted is what was asked for.

ii) **Operator support.**
There needs to be a way for mount requests, etc, to be presented to and acted upon by our operators.

iii) **Catalog/library support.**
We need support for tape Catalogs/libraries. It would be highly desirable if AIX could share Catalogs/libraries with MVS.

e) **Software installation.**
We must not restrice software installation to the sysadmins. It should be possible for anyone with write permissions in the temporary build location to build LPP software. This would allow the installation to separate the LPP application software installation and sysadmin functions.

3) **Security, Integrity, and Reliability**
As the size of these systems grows larger, so does the cost of a "disaster".

a) **Integrity**.
We feel that IBM should make a statement of integrity for AIX, and accept security APAR's against the system. As the system grow larger, and work is right sized on these platforms, integrity is a paramount concern. We feel that a statement of integrity would be a marketing advantage for IBM.

b) **Security.**
"Standard" UNIX security is inadequate for large system use. Too often you have to provide root access or nothing. OSF's approach of ACL's has been found cumbersome and difficult to manage by some who have tried to use it.

What we need is:

i) A way to authorize specific functions to operators so that they can operate the system without being able to destroy it.

ii) A read everything but not write authority, for use by sysadmin's in research mode, and for consultants in help desk mode.

iii) Several people needed the ability to allow specific people to shutdown the system, but have no other special privileges.

iv) There needs to be a way to allow sysadmins to build LPP's but not install them. I.E.: To build them into some temporary location other than the production libraries.

c) **RAS.**

As the number of users on the system becomes large the need to offer 24x7 availability increases. We need to be able to maintain/upgrade the systems with minimal outages, and we need to be able to back out, new code quickly and reliably.

It is tempting to think that fail-over strategies such as those employed in HACMP will be adequate to this need. However, it is not just the OS, but the applications which must exhibit 25x7x356 availability. The unit of loss on a failover of an entire node seems to be too large, especially if a single application can be distributed across those nodes.

AIX/ESA had a substantial amount of work invested in making it more fail soft. It showed in a much more graceful reaction to system resource shortages. As the systems grow larger, they will increasingly encounter job mixes which produce temporary system resource shortages. The OS must survive this and other stresses without necessitating a panic and fail-over. It should be clear that a stress induced failure is unlikely to be relieved by the removal of a nodes worth of resources from the complex.

d) **Licensing.**

With a large number of users, it becomes important that the installation be able to license the software for the use of a few users, even though those few users may run on a large box

Similarly it is important that license-managed software impose limits gracefully, and that additional licenses can be obtained with no delay. We have heard examples of an IBM product which checked for the existence of three license manager daemons every 5 seconds. If it lost communication with any of the daemons it failed all work in progress. A site with intermittent network congestion found that this made the product essentially unusable.

It is also unacceptable to have a batch job fail on a weekend because the number of instantaneous interactive users of a package had driven the license manager to its maximum legal value. A better solution is to have a the license manager keeps giving out access for a fixed period of time, say a month. In the meantime, it notifies system administrators of each occurrence of a request that exceeds the legal limit. During this notification time, the expectation is that the site will buy more licenses and update the license manager.

A license server must be highly repliable and provide good perfomance in a large system. In order to provide seamless availability, sufficient administrative tools should be provided to drain a license server and shift licenses to another server without interruption to the users of the licenced software.