



AWS09:A/UX Buffer Cache Q & As on AWS95

Written by: Sandhya Vora
Revised by: Chuq Von Rospach

August, 1993
May, 1994

This technical note discusses the buffer cache on the Apple Workgroup Server 95, how it affects performance and how to tune your buffer cache to maximize performance.

Q: What is Unix buffer cache and how is it related to performance?

A: Typically, the I/O buffers, or call it cache, store the most recently fetched information from block I/O devices, primarily disk drives. So in general, increasing the number/size of cache increases the chance that commonly accessed data will be in the cache, thereby increasing performance. However, it also reduces the free memory in the system, which increases the probability that the system will run out of physical memory and begin paging or swapping.

If the system commonly has free memory available, and/or if the applications have SMALL memory requirements, the cache size can be increased quite a bit.

If the application is NOT dependent on performance or A/UX is being used as a workstation then, the application should probably use the default cache size, though the cache could be adjusted up or down depending on many factors such as the nature of application, number of processes, file system size etc.

Adding more system RAM definitely gives more flexibility and freedom in terms of using more RAM to cache and thereby affecting the performance, by increasing the probability that the items will be in the cache.

Q: How can proper cache sizing be achieved?

A: A/UX buffer cache sizing on A/UX 3.0.1 and later releases may be achieved through Unix or by the Macintosh memory CDEV. There has been an enhancement made to the Macintosh Memory Control Panel Device (CDEV) to permit viewing and sizing of A/UX buffer cache, the Unix analogue of Macintosh's disk cache, in A/UX Version 3.0.1.

Using this facility, you can set the A/UX buffer cache in the Memory CDEV, located in the Control Panels folder under the Apple Menu. The Cache Size shows the current buffer cache size in use by the running kernel. If the user changes the cache size with the slide control, it will not be changed in the running kernel, until the memory CDEV is closed and the system is rebooted.

The traditional UNIX way, checking and changing the buffer cache size which is the function of the kernel's NBUF and SBUFSIZE parameters, requires you to use commands such as `kconfig -a -n /Unix | grep NBUF`. Since using *kconfig* can be confusing and prone to errors if *kconfig* adjustments are done incorrectly, this enhancement enables you to

change the buffer cache through the memory Control Panel, and the system ensures the buffer cache size stays within acceptable limits.

In addition, A/UX 3.0.1 supports static and dynamic cache sizing.

In dynamic cache sizing, the kernel uses an algorithm to size the buffer cache as it boots, based on the value of NBUF=0 or -1 and SBUFSIZE.

NBUF=0

Set the cache to 10% of memory available at boot time.

NBUF=-1

See AWS08 for details on how the buffers are allocated.

In static cache sizing, the buffer cache is set to $\text{NBUF} * \text{SBUFSIZE}$ bytes, based on the value of $\text{NBUF} > 0$, unless such a cache would be greater than 90% of available memory then it resets to 10% of available memory.

Q: What is the cache setting for the default File/Print as well as database configuration?

A: The AWS95 software comes in two flavors, one for the file/print server configuration and the other for database server configuration.

The manual “Tuning Server Performance and Adding Internal Hard Drives” shipped with release 3.1 has details on how to configure the server.

Q: What A/UX cache settings will optimize server performance?

A: As a default configuration, the file/print server version of A/UX sets NBUF to -1 and SBUFSIZE to 4096, and the database server version of A/UX sets NBUF to 0 and SBUFSIZE to 4096.

In general, an A/UX system acting as a file/print server, should have a fairly large buffer cache. A/UX systems running only database applications that access disks through raw devices should have a small buffer cache. More detail can be found on this in technical note AWS03.