527326-13-3E AID: 258164| 31/05/2020

When we doubling the block size more than there performance at that time the blocks make bigger and this can arise some issue due to there internal fragmentation. A file can only grow in increments of all blocks. If the sizes of files are random, we would expect on the average that half of the last block of a file is wasted. If most files are many blocks large, waste is small, but if the block size is large compared to the size of a typical file, half a block per file is significant. In fact, if files are very small (compared to the block size), the problem is even worse. If, for example, we choose a block size of 8k and the average file is only 1K bytes long, we would be wasting about 7/8 of the disk.

In Unix system most of the files are very small. The Berkeley researchers made a list of the sizes of all files on a typical disk and did some calculations of wasted space by various block sizes. Simply rounding the size of each file up to a multiple of 512 bytes resulted in wasting 4.2% of the space. Including overhead for inodes and indirect blocks, the original 512-byte file system had a total space overhead of 6.9%. Changing to 1K blocks raised the overhead to 11.8%. With 2k blocks, the overhead would be 22.4% and with 4k blocks it would be 45.6%. Would 4k blocks be worthwhile? The answer depends on economics. In those days disks were very expensive, and a wasting half the disk seemed extreme. These days, disks are cheap, and for many applications people would be happy to pay twice as much per byte of disk space to get a disk that was twice as fast.

The UNIX system has share all available resources to the greatest extent possible. In the file system all the available space can be allocate to any single user. But in most of certain environments this is unacceptable. Consequently, a quota mechanism has been added for restricting the amount of file system resources that a user can obtain. The quota mechanism sets limits on both the number of inodes and the number of disk blocks that a user may allocate. A separate quota can be set for each user on each file system. Resources are given both a hard and a soft limit. When a program exceeds a soft limit, a warning is printed on the users terminal; the offending program is not terminated unless it exceeds its hard limit. The idea is that users should stay below their soft limit between login sessions, but they may use more resources while they are actively working. To encourage this behavior, users are warned when logging in if they are over any of their soft limits. If users fails to correct the problem for too many login sessions, they are eventually reprimanded by having their soft limit enforced as their hard limit.