How To Save Half Your Disk Space Automatically !

Paper number: 1029
Paul Wang
SolutionSoft Systems, Inc.
2350 Mission College Blvd. Suite 715
Santa Clara, CA 95054
paulwang@netcom.com
408.988.7378

Abstract

You know your company's critical assets are the files on the systems: customer profiles, e-mails, payrolls, order entries, inventories, and account receivables, etc. Without having all your information readily available, you won't stay competitive for long. So, one thing is for certain. You can't let your systems run out of disk space!

It is intriguing to note that on average 30 to 60 percent of disk space on a MPE/iX system is all wasted: Some free space is too small to be useful. Some disk space is allocated, but never utilized. Some disk space is allocated and utilized, but will either never or unlikely to be accessed again.

This paper explains how disk space is wasted. How those wasted disk space can be reclaimed automatically and systematically. Finally, it discusses the correct way to add new disks when the time comes.

Introduction

Companies around the world rely more and more on information systems to run their operations. This has led to an explosion of the data storage on computer systems. There is no avoiding it: the demand for disk space rarely keeps up with the requirement for it.

On one hand, new technologies such as Data Warehousing, Workflow Automation, Document Imaging, and graphics and multimedia all use large amounts of disk space and ensure that storage requirements will continue to grow at unprecedented rates. On the other hand, the price-per-megabyte of disk space continues to drop, providing more disk space for the same price and leading applications to use more data and keep it around longer. Finally, government and agency auditing requirements often mandate 5, 7 or 15 years worth of operational data must be kept available. This further magnifies the data explosion problem.

Adding more disk space is just a temporary solution for a permanent problem. Disk space will continue to grow, consume hardware budgets, and become more difficult to manage and more time-consuming to back them up.

A big part of the solution is to better manage your disk space automatically. Once those automated procedures are setup, you can be sure your MPE/iX disk space would continue to be saved and better utilized.

Tips To Save Disk Space Automatically

MPE/iX systems allocate disk space in extents. An extent is a chunk of contiguous disk space. The size of an extent is always in multiple of 4 kbytes and anywhere from 4 kbyte to 4 gigabytes. A file may consist of zero, one, or unlimited number of extents.

Due to the file system allocation algorithm and the way some subsystems allocate disk space, some free space is too small to be useful and some disk space is allocated, but never utilized. In addition, just by the nature of the data, some disk space is allocated and utilized, but will either never or unlikely to be accessed again. All those types of disk space are wasted disk space and they adds up really fast. It is intriguing to note that on average 30 to 60 percent of disk space on a MPE/iX system is all wasted.

The good news are most of those wasted disk space can be saved automatically. The magic word here is automatically, which means you can set and forget. Just spend some time set it up once, and you can forget about it and count on the disk space savings will keep on coming.

Let's examine how disk space are wasted and how those techniques can reclaim them back.

Defragment Disks Regularly

Since extents are of different sizes, as extents are allocated and deallocated, the average size of free space tends to become smaller and very tiny fragments of free space may be created. This is called disk fragmentation (see figure 1). Severe disk fragmentation not only decrease performance, but also waste disk space. Any free fragment less than 64 kbytes are wasted, since file system tries to allocate disk space in at least 64 kbyte chunks. It is common to see tens of megabytes of such free space are wasted on user systems.

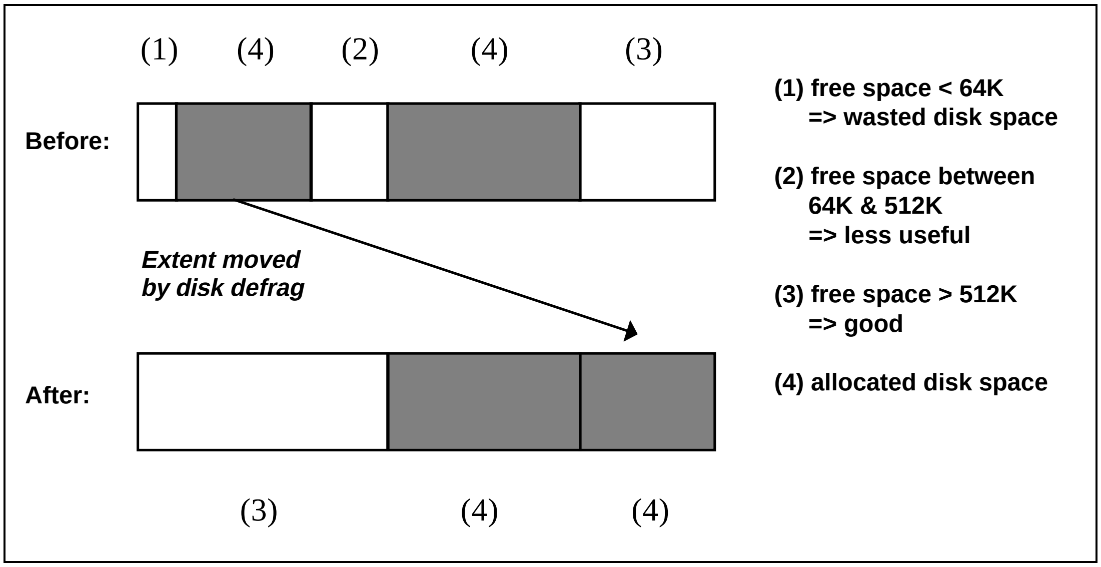

Figure 1. Disk defragmentation creates bigger and more useful free fragments

Free fragment between 64 and 512 kbytes can be utilized by the file system, but they are less useful. When the allocation request is bigger than 16 megabytes, the file system makes sure each extent allocated is at least 512 kbytes. On a severely fragmented system, this can cause premature out of disk space error where the allocation fails despite the system has much more free space than the request! This is one of the most common mystery in the user community. In this case, the system is not out of free space. Rather, it is out of free fragments bigger than 512 kbytes.

The solution is to proactively defragment the disks. Rather than waiting for the fragmentation to accumulate and wasting disk space, we always keep the disks clean. It is recommended to defragment all disks at least once each week. This can be easily automated into daily or weekly batch processing.

Truncate Files Regularly

It is common to have wasted disk space beyond the file EOF due to file system dynamic allocation (extent fault) and EOF cutback. Since the file system has no idea where the EOF will be, it just allocates disk space as it goes. The EOF usually will be set somewhere in the middle of the last extent. The potential wastage this way per file is 4 to 512 Kbytes. Users could also cut back EOF so that any disk space allocated between the old and new EOF are wasted (see figure 2). For example, file opened with write access will reset the EOF to zero! If a huge work file or scratch file (say with a million records) is reused this way with tiny data (say ten records), lots of valuable disk space is wasted. The potential wastage per file is up to the file limit. Wasted disk space adds up really fast. In my experience, it is not uncommon to see thousands or even millions of sectors wasted this way on a production system!

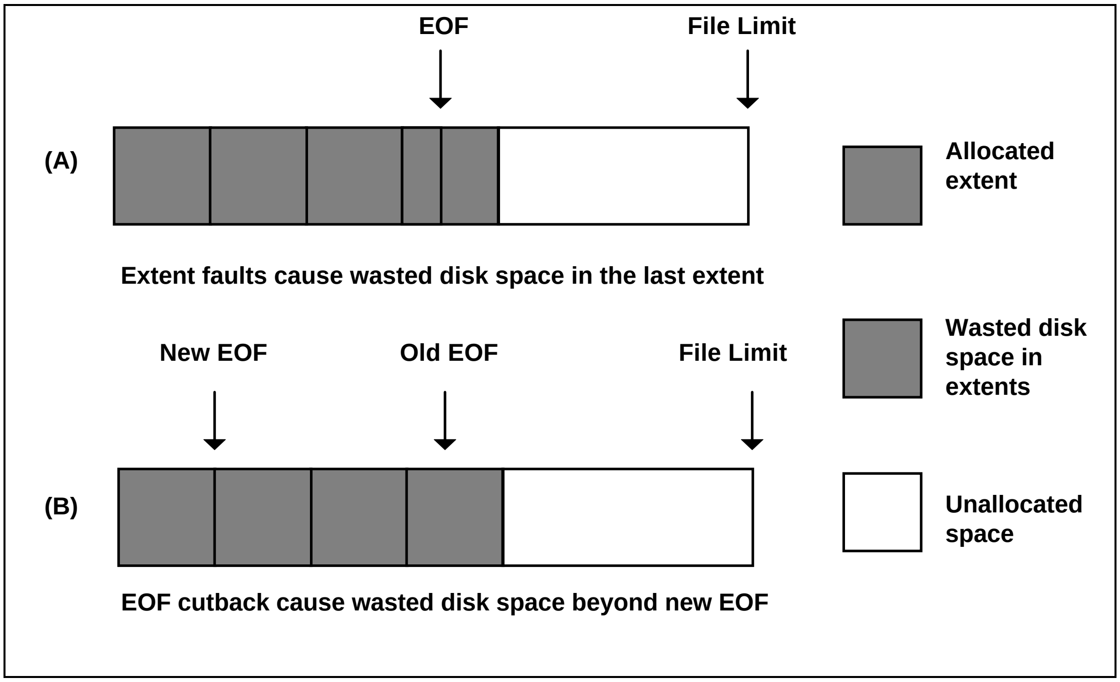
EOF
File Limit
(A)
Allocated extent
Extent faults cause wasted disk space in the last extent
New EOF
Old EOF
File Limit
Wasted disk space in extents
(B)
Unallocated space
EOF cutback cause wasted disk space beyond new EOF

Figure 2. Truncation reclaims wasted disk space beyond EOF

File truncation is probably the easiest and most effective way to reclaim wasted disk space. However, It does have the side effect of increase disk fragmentation. It is recommended that users truncate all files periodically. The best way to achieve this is to incorporate file truncation and disk defragmentation (in this order) into nightly or weekly batch jobs. The disks will be proactively kept clean automatically this way.

Truncate IMAGE/SQL Datasets Periodically

The regular truncation beyond EOF has no effects on IMAGE/SQL files, since IMAGE/SQL subsystem never allocate any disk space beyond EOF. However, any disk space allocated beyond the high-water mark are wasted. They can add up really quickly.

Without Dynamic Detail dataset eXpansion (DDX) enabled, all disk space allocated between the high-water mark and the file limit (or full capacity) are wasted. Since users usually leave room for 10 to 35 percent of capacity, the amount of wasted disk space for IMAGE/SQL datasets could be tremendous.

Enabling DDX with a moderate, incremental size will dramatically reduce the amount of wasted space. Even though disk space allocated beyond the high water mark for a DDX enabled file is still wasted, it is limited up to the incremental size (see figure 3). Please note that Jumbo dataset does not support DDX. As a result, gigabytes worth of disk space may be wasted just by one Jumbo dataset!

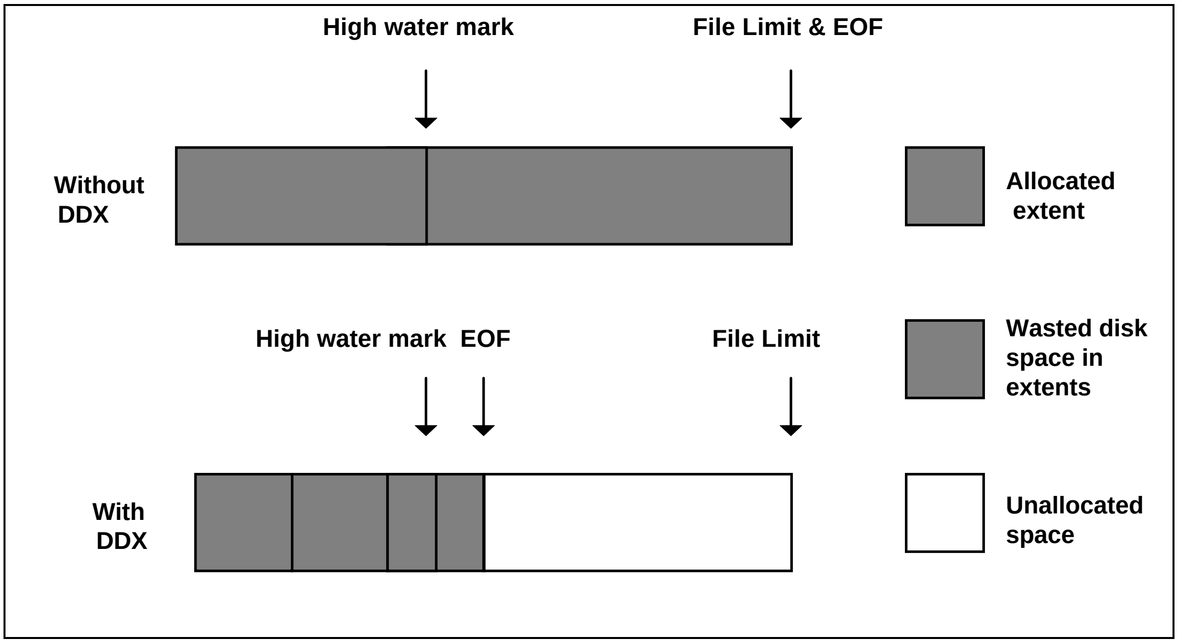

Figure 3. Truncation of IMAGE/SQL datasets reclaims wasted space beyond high water mark

If you can live with a DBPUT could produce an out of disk space error (the trade-off for not preallocating all the disk space), then truncate all IMAGE/SQL files periodically to high-water mark will reclaim all those wasted disk space back. This can be easily automated into weekly or monthly batch processing.

Compress Infrequently-used Files Periodically

The amount of disk space allocated for infrequently-used files is enormous. Just take a look how many files have not been accessed for a month and you will know what I mean. It is the 80/20 rule at work here: 80 percent of the processing is done on only 20 percent of the data. This means that 80 percent of your data is less often used or not at all (see figure 4).

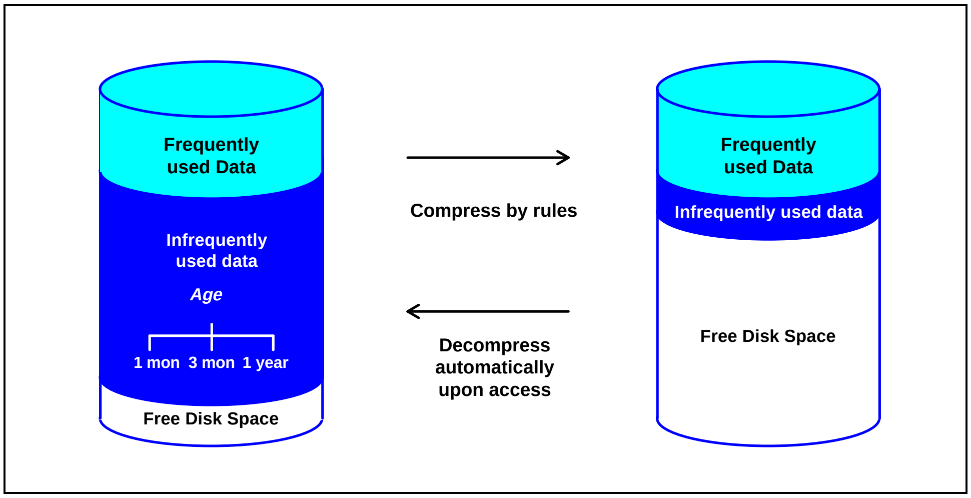

Figure 4. Compress infrequently used data by rules save disk space

Half, or possibly more, of the wasted disk space could be reclaimed if those infrequently-used files were compressed. The actual compression ratio of course depends on the data. In general, MPE/iX files enjoy very high compression ratio. The file system and database subsystem tends to pad blanks or zeros into big fixed length records. In addition, many files are not 100% full in capacity. It is very common to see 50 to 95 percent compression ratio for ASCII, KSAM, IMAGE/SQL, ALLBASE/SQL and Oracle files. On the other hand, program and library files are much harder to compress. Object codes have less repeating pattern and dead space, the compression ratio for those files is 10 to 50 percent.

The best way to do this is to establish compression policies on various accounts (based on last accessed date, for example) and then compress files in batch periodically and automatically according to the policies. Compression not only saves disk space, but also speeds up backups since less data are stored. If your data is 50 percent compressed, then a full backup will be 50 percent faster! The same benefits applies to restore, reload and disaster recovery. If your data is 50 percent compressed, then a reload after a disk crash is 50 percent faster! That is less system downtime.

Of course, when users need to access those compressed files, they must be decompressed first. This could become a big management headache if it occurs frequently and system manager/operator/help desk must be involved each time. The best solution is to utilize an online archiving tool that supports auto-decompression so that compressed files are automatically decompressed as soon as they are accessed. Since decompression speed is very fast (typically 1 to 2 megabytes per second), in most cases, users won’t even notice they are accessing a compressed file.

Archive Or Purge Unused Files Periodically

The amount of disk space allocated for unused files is also quite large. There are probably many files that have not been accessed for more than a year!

Any disk space allocated for unused files is wasted. It is unrealistic to expect all users will review and clean up unused files all the time. Unless global clean-up procedures are in place, those unused files will never go away. Global clean-up procedures may involve defining the clean-up policies for various account, or file types. After certain idle time, those files may be cleaned up by purging and/or archiving to tape.

Certain types of files can be safely purged, such as temporary work files. For other files, it is best to archive (i.e. purge and store) them on tapes. If they are needed later on, just restore them. This could become a big management headache if it occurs frequently and system manager/operator/help desk must be involved each time. The best solution is to utilize an online archiving tool that supports auto-dearchiving so that archived files are automatically dearchived as soon as they are accessed.

How To Add A New Disk

Most users just add a new disk whenever the amount of free disk space is low. With the above automated techniques, you will be able to save lots of disk space and delay the new disk purchases. However, you will probably still have to do it at some point. It is important to do it in the right way so that it won’t become an I/O bottleneck!

MPE/iX extent placement algorithm places new extents on the disk with most free space. When a new disk is added, it will be THE place where new extents are allocated. Due to data locality, those new data tend to be the most active data. Worse yet, if any performance critical databases are restored or reorganized, then they will be allocated solely onto the new disk (see figure 5)! Clearly the I/O demand for the new disk may easily exceed its bandwidth. As a result, the system throughput and response time may suffer.

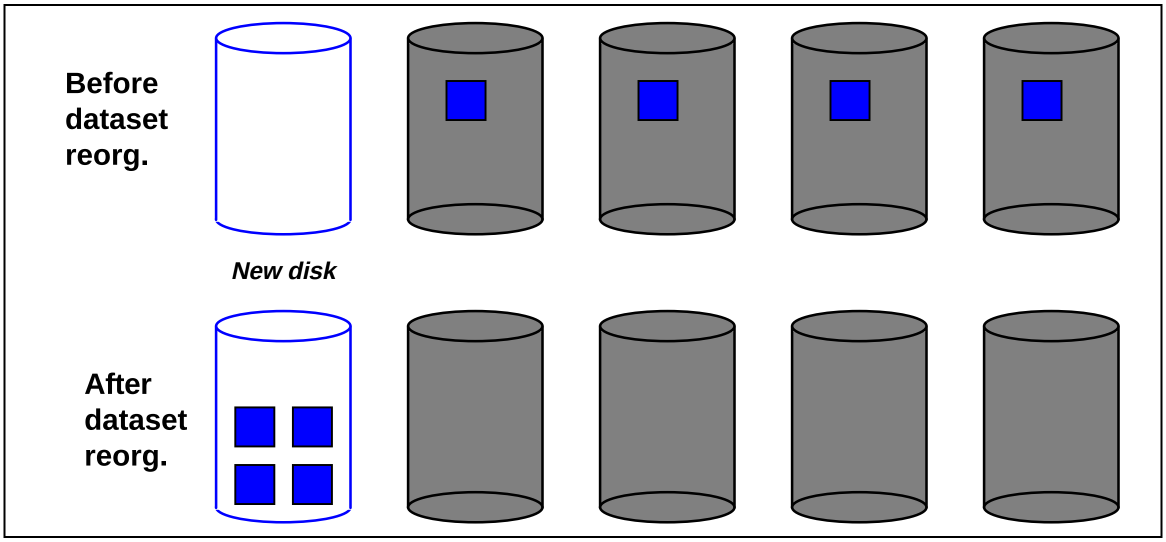

Figure 5. Adding a new disk can easily become an I/O bottleneck

It is of critical importance to populate the volume set to rebalance the disk allocation immediately after adding a new disk. This will not only prevent the new disk from becoming an I/O bottleneck, but also improve system performance by increasing I/O concurrency across all disks in the volume set.

Another potential performance problem is the I/O channel. When add a new disk, do make sure the number of disks and other devices on the channel does not exceed the channel bandwidth.

Conclusion

Data explosion will continue to generate ever more data to be kept available. As a result, disk space will continue to grow, consume hardware budgets, and become more difficult to manage and more time-consuming to backup.

A big part of the solution is to better manage your disk space automatically. By truncating files and defragment disks regularly, you proactively keep your disks clean. The unused space allocated for IMAGE/SQL datasets can be huge, an automated truncation process will keep them well-behaved. Finally, compress and archive files according to established archiving policies is the best way to save disk space and speedup backup/restore.

From my experience, 50 percent of disk space saving is very achievable if all the automated techniques are utilized. I think you will be pleasantly surprised on how easy and how much disk space can be saved on a regularly basis. Keep up with the good work and keep me posted on your results.

Biographic

Paul Wang is the president of SolutionSoft Systems, Inc. He is a software developer and consultant, specialized in transaction management, system performance, file system internals, data base, disk space management and On-line transaction processing. Previously, he was an internal architect of transaction management in HP's Core MPE/iX lab.