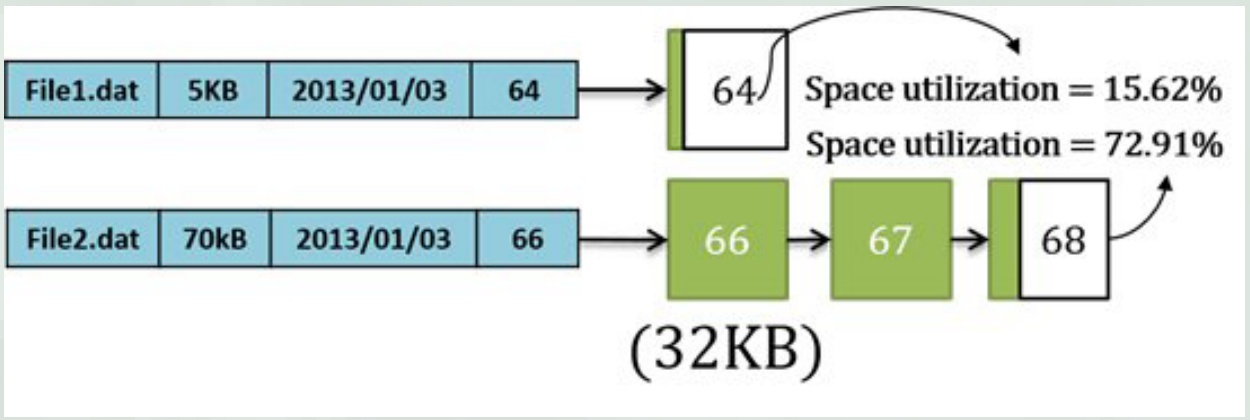
Optimizing Space Utilization of Embedded File Systems

Dynamic Tail Packing Scheme Saves Space up to 64%

Yuan-Hao Chang
Associate Research Fellow



The problem of low space utilization (Fig. 1).

Embedded/mobile computing systems are usually battery-powered devices and are widely adopted in various application domains. Due to cost and energy considerations, they usually have limited computing power, RAM space, and storage capacity. However, recent improvements in embedded/mobile computing systems' computing ability have allowed some embedded computing systems to adopt embedded file systems to simplify the complexity of managing their data . For example, Android file systems no longer use a log-based file system (e.g., yaffs2). Instead of utilizing a log-based file system, some embedded/mobile computing systems, such as mobile phones and embedded consumer electronics, manage their data in flash storage devices with a (simple) file system. However, existing file systems usually allocate storage space in the unit of a cluster, whose size often reaches several kilobytes. This leads to low space utilization in the storing of the tail data of (small) files, and becomes a critical issue in the design of embedded storage systems (i.e., storage systems in embedded computing systems), which usually have a limited storage capacity.

Most file systems in embedded computing systems allocate space for storage of a file in the unit of a cluster — no matter how small the file. Well-known examples are FAT and ext3/ext4 file systems; this is because FAT is simple enough to be used in resource-limited embedded systems and because ext4 is the default file system of Linux operating systems, which are widely used in many embedded systems. However, the cluster-based allocation adopted in many embedded file systems seriously decreases the space available for storage, especially in applications such as sensor nodes and control systems that need to store small data files. This situation is exacerbated in some file systems when their cluster size is increased in proportion to the storage capacity. For example, the cluster size of FAT32 is proportional to the storage capacity, and reaches 32KB when the storage capacity is larger than 32GB. As a result, embedded file systems have low space utilization in the large size of a basic allocation unit because the size of files is usually very small. For example, when the cluster size is 32KB, a 5KB small file would result in the waste of 27KB of space, while a 70KB file would waste 26KB of space, because the file system still allocates one cluster for the small tail of the file (see Figure 1 for details), where the tail of a file indicates the file's last part that cannot completely fill a cluster.

To solve this space-utilization problem, we propose a dynamic tail packing (DTP) scheme to resolve the space utilization issue of embedded file systems (i.e., the file systems used in embedded computing systems). It has two main objectives: (1) to optimize the space utilization of storage systems by packing the tail data of files together with limited performance overhead, and (2) to minimize the internal/external fragmentation issues that exist in the existing tail packing techniques. In order to achieve these objectives, the proposed DTP scheme defines a new type of cluster, called a micro cluster or mCluster, and dynamically packs the tail data of files in mClusters. An mCluster is divided into two parts, the data area and the data table. The data area is used to store the tail data of files, and the data table is used to maintain the start address and the size of the tail data