表 5-4 FAT 的块大小

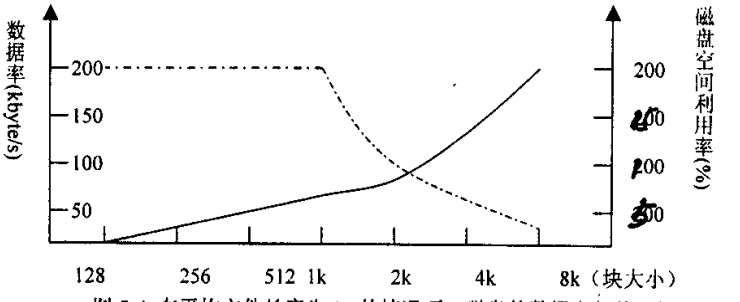

图 5-4 在平均文件长度为 1k 的情况下，磁盘的数据率与使用效率

所有这些给系统提出的要求是：根据磁盘的大小，自动寻找空间和效率的最佳结合点。我们的做法是：在不违反 FAT 的“最大簇数”规定的情况下，使得簇的大小尽量的小，以满足嵌入式系统的特殊需要。

5.2 文件系统调用和算法举例

5.2.1 文件系统调用

现在的文件系统版本提供了以下文件系统系统调用：

函数名称	函数原型	函数功能
dfs_open	int dfs_open(char *path, int flag)	打开一个已存在的文件
dfs_close	int dfs_close(int fd)	关闭一个已打开的文件
dfs_creat	int dfs_creat(char *path, int attrib)	创建一个新文件
dfs_delete	int dfs_delete(char * path)	删除指定路径下的文件
dfs_read	int dfs_read(int fd, void * buffer, unsigned int count)	函数从 fd（文件描述符）所指向的文件中读取长度为 count 的数据内容，并放到 buffer 所指向的目的缓冲区去
dfs_rename	int dfs_rename(char * path1,	修改文件名，将 path1 所在路径的文

	char * path2)	件名改为 path2 所在路径的文件名
dfs_getfsize	long dfs_getfsize(int fd)	获得一个指定文件的长度
dfs_getfoffset	long dfs_getfoffset(int fd)	获得当前文件的偏移量
dfs_splitpath	int fs_splitpath(char * fd,int * drive,char * dir,char * name,char * ext);	将该完整的文件描述符 fd，分解成盘符 drive；路径 dir；文件名 name；扩展名 ext。
dfs_getcufsize	long dfs_getcufsize(int fd)	获得一个文件当前文件的长度
dfs_setfsize	int dfs_setfsize(int fd, long size)	设置一个指定文件的长度
dfs_write	int dfs_write(int fd, void * buffer, unsigned int count)	函数把 count 个字节从 buffer 所指向的缓冲区写入到 fd 指向的文件中
dfs_lseek	long dfs_lseek(int fd, long foffset, int origin)	把 fd 文件描述符所指向的指针定位到 foffset 和 origin 所指定的位置
dfs_getfattr	int dfs_getfattr(char *name,char attrp)	获得一个指定文件的属性
dfs_setfattr	int dfs_setfattr(char *name,char attrp)	设置一个指定文件的属性
dfs_rmdir	int dfs_rmdir(char * path)	删除由 path 所指定的路径名下的目录
dfs_mkdir	int dfs_mkdir(char * dir)	在 dir 所指向的路径名下建立一个新的目录

5.2.2 实现举例

到将上面提到的函数实现去描述清楚无疑是一件繁琐的事情。在这一小节中，采用自顶向下的描述方法，扼要的介绍文件系统中最典型的调用：dfs_read。