Bigalloc

At the moment, the default size of a block is 4KiB, which is a commonly supported page size on most MMU-capable hardware. This is fortunate, as ext4 code is not prepared to handle the case where the block size exceeds the page size. However, for a filesystem of mostly huge files, it is desirable to be able to allocate disk blocks in units of multiple blocks to reduce both fragmentation and metadata overhead. The bigalloc feature provides exactly this ability.

The bigalloc feature (EXT4_FEATURE_RO_COMPAT_BIGALLOC) changes ext4 to use clustered allocation, so that each bit in the ext4 block allocation bitmap addresses a power of two number of blocks. For example, if the file system is mainly going to be storing large files in the 4-32 megabyte range, it might make sense to set a cluster size of 1 megabyte. This means that each bit in the block allocation bitmap now addresses 256 4k blocks. This shrinks the total size of the block allocation bitmaps for a 2T file system from 64 megabytes to 256 kilobytes. It also means that a block group addresses 32 gigabytes instead of 128 megabytes, also shrinking the amount of file system overhead for metadata.

The administrator can set a block cluster size at mkfs time (which is stored in the s_log_cluster_size field in the superblock); from then on, the block bitmaps track clusters, not individual blocks. This means that block groups can be several gigabytes in size (instead of just 128MiB); however, the minimum allocation unit becomes a cluster, not a block, even for directories. TaoBao had a patchset to extend the "use units of clusters instead of blocks" to the extent tree, though it is not clear where those patches went-- they eventually morphed into "extent tree v2" but that code has not landed as of May 2015.