Subject: [PATCH 3/4] The block devices layer changes
Posted by Pavel Emelianov on Thu, 07 Feb 2008 12:59:48 GMT
View Forum Message <> Reply to Message

They are the same as for the character layer, but
the good news is that there are no caching in this
case.

So this patch is smaller and easier to understand
as compared to the previous one.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

```
diff --git a/block/genhd.c b/block/genhd.c
index 5e4ab4b..6f9ef48 100644
--- a/block/genhd.c
+++ b/block/genhd.c
@@ -8,6 +8,7 @@
 #include <linux/kdev_t.h>
 #include <linux/kernel.h>
 #include <linux/blkdev.h>
+#include <linux/devscontrol.h>
 #include <linux/init.h>
 #include <linux/spinlock.h>
 #include <linux/seq_file.h>
@@ -195,6 +196,57 @@ void unlink_gendisk(struct gendisk *disk)
          disk->minors);
 }

+#ifdef CONFIG_CGROUP_DEVS
+int bdev_add_to_map(struct kobj_map *map, dev_t dev, int all, mode_t mode)
+{
+ int tmp;
+ struct kobject *kobj;
+ struct device *d;
+ struct gendisk *disk;
+
+ kobj = kobj_lookup(bdev_map, dev, NULL, &tmp);
+ if (kobj == NULL)
+  return -ENODEV;
+
+ d = kobj_to_dev(kobj);
+ disk = dev_to_disk(d);
+ tmp = kobj_remap(map, dev, mode, all ? MINORBITS : 1, NULL,
+   exact_match, exact_lock, disk);
+ if (tmp < 0) {
```

```
+  put_disk(disk);
+  return tmp;
+ }
+
+ return 0;
+}
+
+int bdev_del_from_map(struct kobj_map *map, dev_t dev, int all)
+{
+ int tmp;
+ struct kobject *kobj;
+ struct device *d;
+ struct gendisk *disk;
+
+ kobj = kobj_lookup(bdev_map, dev, NULL, &tmp);
+ if (kobj == NULL)
+  return -ENODEV;
+
+ d = kobj_to_dev(kobj);
+ disk = dev_to_disk(d);
+ kobj_unmap(map, dev, all ? MINORBITS : 1);
+
+ put_disk(disk);
+ put_disk(disk);
+ return 0;
+}
+
+void bdev_iterate_map(struct kobj_map *map,
+  int (*fn)(dev_t, int, mode_t, void *), void *x)
+{
+ kobj_map_iterate(map, fn, x);
+}
+#endif
+
 /**
  * get_gendisk - get partitioning information for a given device
  * @dev: device to get partitioning information for
@@ -202,10 +254,18 @@ void unlink_gendisk(struct gendisk *disk)
  * This function gets the structure containing partitioning
  * information for the given device @dev.
  */
-struct gendisk *get_gendisk(dev_t devt, int *part)
+struct gendisk *get_gendisk(dev_t devt, mode_t *mode, int *part)
 {
- struct kobject *kobj = kobj_lookup(bdev_map, devt, part);
- struct device *dev = kobj_to_dev(kobj);
+ struct kobj_map *map;
+ struct kobject *kobj;
+ struct device *dev;
+
+ map = task_bdev_map(current);
+ if (map == NULL)
+  map = bdev_map;
+
+ kobj = kobj_lookup(map, devt, mode, part);
+ dev = kobj_to_dev(kobj);

  return  kobj ? dev_to_disk(dev) : NULL;
 }
@@ -356,10 +416,20 @@ static struct kobject *base_probe(dev_t devt, int *part, void *data)
  return NULL;
 }

+struct kobj_map *bdev_map_init(void)
+{
+ return kobj_map_init(base_probe, &block_class_lock);
+}
+
+void bdev_map_fini(struct kobj_map *map)
+{
+ kobj_map_fini(map);
+}
+
 static int __init genhd_device_init(void)
 {
  class_register(&block_class);
- bdev_map = kobj_map_init(base_probe, &block_class_lock);
+ bdev_map = bdev_map_init();
  blk_dev_init();

 #ifndef CONFIG_SYSFS_DEPRECATED
diff --git a/fs/block_dev.c b/fs/block_dev.c
index 55295a4..03b1b5e 100644
--- a/fs/block_dev.c
+++ b/fs/block_dev.c
@@ -1129,16 +1129,25 @@ static int do_open(struct block_device *bdev, struct file *file, int for_part)
  struct module *owner = NULL;
  struct gendisk *disk;
  int ret = -ENXIO;
+ mode_t mode;
  int part;

  file->f_mapping = bdev->bd_inode->i_mapping;
  lock_kernel();
- disk = get_gendisk(bdev->bd_dev, &part);
```

```
+ disk = get_gendisk(bdev->bd_dev, &mode, &part);
  if (!disk) {
   unlock_kernel();
   bdput(bdev);
   return ret;
  }
+
+ if ((file->f_mode & mode) != file->f_mode) {
+  unlock_kernel();
+  bdput(bdev);
+  put_disk(disk);
+  return -EACCES;
+ }
+
  owner = disk->fops->owner;

  mutex_lock_nested(&bdev->bd_mutex, for_part);
diff --git a/include/linux/genhd.h b/include/linux/genhd.h
index 1dbea0a..4a92b65 100644
--- a/include/linux/genhd.h
+++ b/include/linux/genhd.h
@@ -240,7 +240,15 @@ extern int get_blkdev_list(char *, int);
 extern void add_disk(struct gendisk *disk);
 extern void del_gendisk(struct gendisk *gp);
 extern void unlink_gendisk(struct gendisk *gp);
-extern struct gendisk *get_gendisk(dev_t dev, int *part);
+extern struct gendisk *get_gendisk(dev_t dev, mode_t *mode, int *part);
+
+struct kobj_map;
+extern int bdev_add_to_map(struct kobj_map *, dev_t dev, int all, mode_t mode);
+extern int bdev_del_from_map(struct kobj_map *map, dev_t dev, int all);
+extern void bdev_iterate_map(struct kobj_map *map,
+  int (*fn)(dev_t, int, mode_t, void *), void *x);
+extern struct kobj_map *bdev_map_init(void);
+extern void bdev_map_fini(struct kobj_map *map);

 extern void set_device_ro(struct block_device *bdev, int flag);
 extern void set_disk_ro(struct gendisk *disk, int flag);
```

____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers