
Subject: [RFC][PATCH] memory controller per zone patches take 2 [6/10]
remember reclaim priority for memory c
Posted by [KAMEZAWA Hiroyuki](#) on Fri, 16 Nov 2007 10:23:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

Define function to remember reclaim priority (as zone->prev_priority)

Signed-off-by: KAMEZAWA Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com>

include/linux/memcontrol.h | 23 +++++
mm/memcontrol.c | 20 +++++
2 files changed, 43 insertions(+)

Index: linux-2.6.24-rc2-mm1/mm/memcontrol.c

```
=====
--- linux-2.6.24-rc2-mm1.orig/mm/memcontrol.c
+++ linux-2.6.24-rc2-mm1/mm/memcontrol.c
@@ -133,6 +133,7 @@ struct mem_cgroup {
    */
    spinlock_t lru_lock;
    unsigned long control_type; /* control RSS or RSS+Pagecache */
+ int prev_priority; /* for recording reclaim priority */
    /*
     * statistics.
     */
@@ -452,6 +453,25 @@ int mem_cgroup_reclaim_imbalance(struct
    return active / (inactive + 1);
}

+/*
+ * prev_priority control...this will be used in memory reclaim path.
+ */
+int mem_cgroup_get_reclaim_priority(struct mem_cgroup *mem)
+{
+ return mem->prev_priority;
+}
+
+void mem_cgroup_note_reclaim_priority(struct mem_cgroup *mem, int priority)
+{
+ if (priority < mem->prev_priority)
+ mem->prev_priority = priority;
+}
+
+void mem_cgroup_record_reclaim_priority(struct mem_cgroup *mem, int priority)
+{
+ mem->prev_priority = priority;
+}
+
```

```
unsigned long mem_cgroup_isolate_pages(unsigned long nr_to_scan,
    struct list_head *dst,
    unsigned long *scanned, int order,
```

Index: linux-2.6.24-rc2-mm1/include/linux/memcontrol.h

=====

--- linux-2.6.24-rc2-mm1.orig/include/linux/memcontrol.h

+++ linux-2.6.24-rc2-mm1/include/linux/memcontrol.h

```
@@ -67,6 +67,11 @@ extern void mem_cgroup_page_migration(st
extern int mem_cgroup_calc_mapped_ratio(struct mem_cgroup *mem);
extern int mem_cgroup_reclaim_imbalance(struct mem_cgroup *mem);
```

```
+extern int mem_cgroup_get_reclaim_priority(struct mem_cgroup *mem);
+extern void mem_cgroup_note_reclaim_priority(struct mem_cgroup *mem,
+    int priority);
+extern void mem_cgroup_record_reclaim_priority(struct mem_cgroup *mem,
+    int priority);
```

```
#else /* CONFIG_CGROUP_MEM_CONT */
```

```
@@ -150,6 +155,24 @@ static inline int mem_cgroup_reclaim_imb
    return 0;
}
```

```
+static inline int mem_cgroup_get_reclaim_priority(struct mem_cgroup *mem,
+    int priority)
```

```
+{
+    return 0;
+}
```

```
+
+static inline void mem_cgroup_note_reclaim_priority(struct mem_cgroup *mem,
+    int priority)
```

```
+{
+    return 0;
+}
```

```
+
+static inline void mem_cgroup_record_reclaim_priority(struct mem_cgroup *mem,
+    int priority)
```

```
+{
+    return 0;
+}
```

```
#endif /* CONFIG_CGROUP_MEM_CONT */
```

```
#endif /* _LINUX_MEMCONTROL_H */
```

Containers mailing list

Containers@lists.linux-foundation.org

