Subject: [RFC][for -mm] memory controller enhancements for NUMA [6/10] record reclaim priority
Posted by KAMEZAWA Hiroyuki on Wed, 14 Nov 2007 08:50:19 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
@@ -125,6 +125,7 @@ struct mem_cgroup {
   */
  struct list_head active_list;
  struct list_head inactive_list;
+ int    prev_priority; /* used in memory reclaim (see zone's one)*/
  /*
   * spin_lock to protect the per cgroup LRU
   */
@@ -429,6 +430,25 @@ 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