Subject: [PATCH v5 01/18] move slabinfo processing to slab_common.c
Posted by Glauber Costa on Fri, 19 Oct 2012 14:20:25 GMT
View Forum Message <> Reply to Message

This patch moves all the common machinery to slabinfo processing
to slab_common.c. We can do better by noticing that the output is
heavily common, and having the allocators to just provide finished
information about this. But after this first step, this can be done
easier.

Signed-off-by: Glauber Costa <glommer@parallels.com>
Acked-by: Christoph Lameter <cl@linux.com>
CC: Pekka Enberg <penberg@cs.helsinki.fi>
CC: David Rientjes <rientjes@google.com>

```
---
 mm/slab.c        | 72 ++++++++++----------------------------------------------------------------
 mm/slab.h        |  8 +++++++
 mm/slab_common.c | 70 +++++++++++++++++++++++++++++++++++++++++++++++++++++++
 mm/slub.c        | 51 ++++----------------------------------------
 4 files changed, 96 insertions(+), 105 deletions(-)

diff --git a/mm/slab.c b/mm/slab.c
index eafef58..e35970a 100644
--- a/mm/slab.c
+++ b/mm/slab.c
@@ -4263,7 +4263,7 @@ out:

 #ifdef CONFIG_SLABINFO

-static void print_slabinfo_header(struct seq_file *m)
+void print_slabinfo_header(struct seq_file *m)
 {
  /*
   * Output format version, so at least we can change it
@@ -4286,28 +4286,7 @@ static void print_slabinfo_header(struct seq_file *m)
  seq_putc(m, '\n');
 }

-static void *s_start(struct seq_file *m, loff_t *pos)
-{
- loff_t n = *pos;
-
- mutex_lock(&slab_mutex);
- if (!n)
-  print_slabinfo_header(m);
-
- return seq_list_start(&slab_caches, *pos);
-}
```

```
-
-static void *s_next(struct seq_file *m, void *p, loff_t *pos)
-{
- return seq_list_next(p, &slab_caches, pos);
-}
-
-static void s_stop(struct seq_file *m, void *p)
-{
- mutex_unlock(&slab_mutex);
-}
-
-static int s_show(struct seq_file *m, void *p)
+int slabinfo_show(struct seq_file *m, void *p)
 {
  struct kmem_cache *cachep = list_entry(p, struct kmem_cache, list);
  struct slab *slabp;
@@ -4404,27 +4383,6 @@ static int s_show(struct seq_file *m, void *p)
  return 0;
 }

-/*
- * slabinfo_op - iterator that generates /proc/slabinfo
- *
- * Output layout:
- * cache-name
- * num-active-objs
- * total-objs
- * object size
- * num-active-slabs
- * total-slabs
- * num-pages-per-slab
- * + further values on SMP and with statistics enabled
- */
-
-static const struct seq_operations slabinfo_op = {
- .start = s_start,
- .next = s_next,
- .stop = s_stop,
- .show = s_show,
-};
-
 #define MAX_SLABINFO_WRITE 128
 /**
  * slabinfo_write - Tuning for the slab allocator
@@ -4433,7 +4391,7 @@ static const struct seq_operations slabinfo_op = {
  * @count: data length
  * @ppos: unused
  */
```

```
-static ssize_t slabinfo_write(struct file *file, const char __user *buffer,
+ssize_t slabinfo_write(struct file *file, const char __user *buffer,
          size_t count, loff_t *ppos)
 {
  char kbuf[MAX_SLABINFO_WRITE + 1], *tmp;
@@ -4476,19 +4434,6 @@ static ssize_t slabinfo_write(struct file *file, const char __user *buffer,
  return res;
 }

-static int slabinfo_open(struct inode *inode, struct file *file)
-{
- return seq_open(file, &slabinfo_op);
-}
-
-static const struct file_operations proc_slabinfo_operations = {
- .open  = slabinfo_open,
- .read  = seq_read,
- .write  = slabinfo_write,
- .llseek  = seq_lseek,
- .release = seq_release,
-};
-
 #ifdef CONFIG_DEBUG_SLAB_LEAK

 static void *leaks_start(struct seq_file *m, loff_t *pos)
@@ -4617,6 +4562,16 @@ static int leaks_show(struct seq_file *m, void *p)
  return 0;
 }

+static void *s_next(struct seq_file *m, void *p, loff_t *pos)
+{
+ return seq_list_next(p, &slab_caches, pos);
+}
+
+static void s_stop(struct seq_file *m, void *p)
+{
+ mutex_unlock(&slab_mutex);
+}
+
 static const struct seq_operations slabstats_op = {
  .start = leaks_start,
  .next = s_next,
@@ -4651,7 +4606,6 @@ static const struct file_operations proc_slabstats_operations = {

 static int __init slab_proc_init(void)
 {
- proc_create("slabinfo",S_IWUSR|S_IRUSR,NULL,&proc_slabinfo_operations);
 #ifdef CONFIG_DEBUG_SLAB_LEAK
```

```
  proc_create("slab_allocators", 0, NULL, &proc_slabstats_operations);
 #endif
diff --git a/mm/slab.h b/mm/slab.h
index 35b60b7..4156d21 100644
--- a/mm/slab.h
+++ b/mm/slab.h
@@ -72,4 +72,12 @@ static inline struct kmem_cache *__kmem_cache_alias(const char *name, size_t siz

 int __kmem_cache_shutdown(struct kmem_cache *);

+struct seq_file;
+struct file;
+void print_slabinfo_header(struct seq_file *m);
+
+int slabinfo_show(struct seq_file *m, void *p);
+
+ssize_t slabinfo_write(struct file *file, const char __user *buffer,
+         size_t count, loff_t *ppos);
 #endif
diff --git a/mm/slab_common.c b/mm/slab_common.c
index 0e2b8e3..11ecab4 100644
--- a/mm/slab_common.c
+++ b/mm/slab_common.c
@@ -13,6 +13,8 @@
 #include <linux/module.h>
 #include <linux/cpu.h>
 #include <linux/uaccess.h>
+#include <linux/seq_file.h>
+#include <linux/proc_fs.h>
 #include <asm/cacheflush.h>
 #include <asm/tlbflush.h>
 #include <asm/page.h>
@@ -196,3 +198,71 @@ int slab_is_available(void)
 {
  return slab_state >= UP;
 }
+
+#ifdef CONFIG_SLABINFO
+static void *s_start(struct seq_file *m, loff_t *pos)
+{
+ loff_t n = *pos;
+
+ mutex_lock(&slab_mutex);
+ if (!n)
+  print_slabinfo_header(m);
+
+ return seq_list_start(&slab_caches, *pos);
```

```
+}
+
+static void *s_next(struct seq_file *m, void *p, loff_t *pos)
+{
+ return seq_list_next(p, &slab_caches, pos);
+}
+
+static void s_stop(struct seq_file *m, void *p)
+{
+ mutex_unlock(&slab_mutex);
+}
+
+static int s_show(struct seq_file *m, void *p)
+{
+ return slabinfo_show(m, p);
+}
+
+/*
+ * slabinfo_op - iterator that generates /proc/slabinfo
+ *
+ * Output layout:
+ * cache-name
+ * num-active-objs
+ * total-objs
+ * object size
+ * num-active-slabs
+ * total-slabs
+ * num-pages-per-slab
+ * + further values on SMP and with statistics enabled
+ */
+static const struct seq_operations slabinfo_op = {
+ .start = s_start,
+ .next = s_next,
+ .stop = s_stop,
+ .show = s_show,
+};
+
+static int slabinfo_open(struct inode *inode, struct file *file)
+{
+ return seq_open(file, &slabinfo_op);
+}
+
+static const struct file_operations proc_slabinfo_operations = {
+ .open  = slabinfo_open,
+ .read  = seq_read,
+ .write          = slabinfo_write,
+ .llseek  = seq_lseek,
+ .release = seq_release,
+};
+
+static int __init slab_proc_init(void)
+{
+ proc_create("slabinfo", S_IRUSR, NULL, &proc_slabinfo_operations);
+ return 0;
+}
+module_init(slab_proc_init);
+#endif /* CONFIG_SLABINFO */
diff --git a/mm/slub.c b/mm/slub.c
index f50c5b2..55304ed 100644
--- a/mm/slub.c
+++ b/mm/slub.c
@@ -5394,7 +5394,7 @@ __initcall(slab_sysfs_init);
  * The /proc/slabinfo ABI
  */
 #ifdef CONFIG_SLABINFO
-static void print_slabinfo_header(struct seq_file *m)
+void print_slabinfo_header(struct seq_file *m)
 {
  seq_puts(m, "slabinfo - version: 2.1\n");
  seq_puts(m, "# name            <active_objs> <num_objs> <object_size> "
@@ -5404,28 +5404,7 @@ static void print_slabinfo_header(struct seq_file *m)
  seq_putc(m, '\n');
 }

-static void *s_start(struct seq_file *m, loff_t *pos)
-{
- loff_t n = *pos;
-
- mutex_lock(&slab_mutex);
- if (!n)
-  print_slabinfo_header(m);
-
- return seq_list_start(&slab_caches, *pos);
-}
-
-static void *s_next(struct seq_file *m, void *p, loff_t *pos)
-{
- return seq_list_next(p, &slab_caches, pos);
-}
-
-static void s_stop(struct seq_file *m, void *p)
-{
- mutex_unlock(&slab_mutex);
-}
-
-static int s_show(struct seq_file *m, void *p)
```

```
+int slabinfo_show(struct seq_file *m, void *p)
 {
  unsigned long nr_partials = 0;
  unsigned long nr_slabs = 0;
@@ -5461,29 +5440,9 @@ static int s_show(struct seq_file *m, void *p)
  return 0;
 }

-static const struct seq_operations slabinfo_op = {
- .start = s_start,
- .next = s_next,
- .stop = s_stop,
- .show = s_show,
-};
-
-static int slabinfo_open(struct inode *inode, struct file *file)
-{
- return seq_open(file, &slabinfo_op);
-}
-
-static const struct file_operations proc_slabinfo_operations = {
- .open  = slabinfo_open,
- .read  = seq_read,
- .llseek  = seq_lseek,
- .release = seq_release,
-};
-
-static int __init slab_proc_init(void)
+ssize_t slabinfo_write(struct file *file, const char __user *buffer,
+         size_t count, loff_t *ppos)
 {
- proc_create("slabinfo", S_IRUSR, NULL, &proc_slabinfo_operations);
- return 0;
+ return -EIO;
 }
-module_init(slab_proc_init);
 #endif /* CONFIG_SLABINFO */
--
1.7.11.7
```