

Re: [PATCH 1/5] generic __remove_pages() support

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-03/msg02881.html>

- *From:* Badari Pulavarty <pbadari@xxxxxxxxxxx>
 - *Date:* Thu, 06 Mar 2008 15:37:20 -0800
-

Here is the latest version, addressing Randy and Dave's comments.

Thanks,
Badari

Generic helper function to remove section mappings and sysfs entries for the section of the memory we are removing. `offline_pages()` correctly adjusted zone and marked the pages reserved.

Issue: If `mem_map`, `usemap` allocation could come from different places – `kmalloc`, `vmalloc`, `alloc_pages` or `bootmem`. There is no easy way to find and free up `bootmem` allocations.

Signed-off-by: Badari Pulavarty <pbadari@xxxxxxxxxxx>

include/linux/memory_hotplug.h | 6 +++-
mm/memory_hotplug.c | 55 +++
mm/sparse.c | 45 ++++++++++++++++++++++++++++++++++++++-----
3 files changed, 102 insertions(+), 4 deletions(-)

Index: linux-2.6.25-rc3.save/mm/memory_hotplug.c

```
=====
--- linux-2.6.25-rc3.save.orig/mm/memory_hotplug.c 2008-03-05 10:44:30.000000000 -0800
+++ linux-2.6.25-rc3.save/mm/memory_hotplug.c 2008-03-06 15:08:45.000000000 -0800
@@ -102,6 +102,21 @@ static int __add_section(struct zone *zo
return register_new_memory(__pfn_to_section(phys_start_pfn));
}

+static int __remove_section(struct zone *zone, struct mem_section *ms)
+{
+ int ret = -EINVAL;
+
+ if (!valid_section(ms))
+ return ret;
+
+ ret = unregister_memory_section(ms);
+ if (ret)
```

Re: [PATCH 1/5] generic __remove_pages() support

```
+ return ret;
+
+ sparse_remove_one_section(zone, ms);
+ return 0;
+}
+
+/*
+ * Reasonably generic function for adding memory. It is
+ * expected that archs that support memory hotplug will
+ * @@ -135,6 +150,46 @@ int __add_pages(struct zone *zone, unsigned
+ * }
+ EXPORT_SYMBOL_GPL(__add_pages);
+
+/**
+ * __remove_pages() – remove sections of pages from a zone
+ * @zone: zone from which pages need to be removed
+ * @phys_start_pfn: starting pageframe (must be aligned to start of a section)
+ * @nr_pages: number of pages to remove (must be multiple of section size)
+ *
+ * Generic helper function to remove section mappings and sysfs entries
+ * for the section of the memory we are removing. Caller needs to make
+ * sure that pages are marked reserved and zones are adjust properly by
+ * calling offline_pages().
+ */
+int __remove_pages(struct zone *zone, unsigned long phys_start_pfn,
+ unsigned long nr_pages)
+{
+ unsigned long i, ret = 0;
+ int sections_to_remove;
+ unsigned long flags;
+ struct pglist_data *pgdat = zone->zone_pgdat;
+
+ /*
+ * We can only remove entire sections
+ */
+ BUG_ON(phys_start_pfn & ~PAGE_SECTION_MASK);
+ BUG_ON(nr_pages % PAGES_PER_SECTION);
+
+ release_mem_region(phys_start_pfn << PAGE_SHIFT, nr_pages * PAGE_SIZE);
+
+ sections_to_remove = nr_pages / PAGES_PER_SECTION;
+ for (i = 0; i < sections_to_remove; i++) {
+ unsigned long pfn = phys_start_pfn + i * PAGES_PER_SECTION;
+ pgdat_resize_lock(pgdat, &flags);
+ ret = __remove_section(zone, __pfn_to_section(pfn));
+ pgdat_resize_unlock(pgdat, &flags);
+ if (ret)
+ break;
+ }
+ return ret;
+}
```

Re: [PATCH 1/5] generic __remove_pages() support

```
+EXPORT_SYMBOL_GPL(__remove_pages);
+
static void grow_zone_span(struct zone *zone,
unsigned long start_pfn, unsigned long end_pfn)
{
Index: linux-2.6.25-rc3.save/mm/sparse.c
=====
--- linux-2.6.25-rc3.save.orig/mm/sparse.c 2008-03-05 10:44:30.000000000 -0800
+++ linux-2.6.25-rc3.save/mm/sparse.c 2008-03-06 15:15:18.000000000 -0800
@@ -198,12 +198,13 @@ static unsigned long sparse_encode_mem_m
}

/*
- * We need this if we ever free the mem_maps. While not implemented yet,
- * this function is included for parity with its sibling.
+ * Decode mem_map from the coded memmap
*/
-static __attribute__((unused))
+static
struct page *sparse_decode_mem_map(unsigned long coded_mem_map, unsigned long pnum)
{
+ /* mask off the extra low bits of information */
+ coded_mem_map &= SECTION_MAP_MASK;
return ((struct page *)coded_mem_map) + section_nr_to_pfn(pnum);
}

@@ -363,6 +364,28 @@ static void __kfree_section_memmap(struc
}
#endif /* CONFIG_SPARSEMEM_VMEMMAP */

+static void free_section_usemap(struct page *memmap, unsigned long *usemap)
+{
+ if (!usemap)
+ return;
+
+ /*
+ * Check to see if allocation came from hot-plug-add
+ */
+ if (PageSlab(virt_to_page(usemap))) {
+ kfree(usemap);
+ if (memmap)
+ __kfree_section_memmap(memmap, PAGES_PER_SECTION);
+ return;
+ }
+
+ /*
+ * TODO: Allocations came from bootmem - how do I free up ?
+ */
+ printk(KERN_WARNING "Not freeing up allocations from bootmem "
+ "- leaking memory\n");
+}

```

Re: [PATCH 1/5] generic __remove_pages() support

```
+
+/*
+ * returns the number of sections whose mem_maps were properly
+ * set. If this is <=0, then that means that the passed-in
@@ -415,4 +438,20 @@ out:
+ }
+ return ret;
+ }
+
+void sparse_remove_one_section(struct zone *zone, struct mem_section *ms)
+{
+ struct page *memmap = NULL;
+ unsigned long *usemap = NULL;
+
+ if (ms->section_mem_map) {
+ usemap = ms->pageblock_flags;
+ memmap = sparse_decode_mem_map(ms->section_mem_map,
+ __section_nr(ms));
+ ms->section_mem_map = 0;
+ ms->pageblock_flags = NULL;
+ }
+
+ free_section_usemap(memmap, usemap);
+}
#endif
Index: linux-2.6.25-rc3.save/include/linux/memory_hotplug.h
=====
--- linux-2.6.25-rc3.save.orig/include/linux/memory_hotplug.h 2008-03-05 10:44:30.000000000 -0800
+++ linux-2.6.25-rc3.save/include/linux/memory_hotplug.h 2008-03-06 15:02:13.000000000 -0800
@@ -8,6 +8,7 @@
struct page;
struct zone;
struct pglist_data;
+struct mem_section;

#ifdef CONFIG_MEMORY_HOTPLUG
+/*
@@ -64,9 +65,11 @@ extern int offline_pages(unsigned long,
+/* reasonably generic interface to expand the physical pages in a zone */
extern int __add_pages(struct zone *zone, unsigned long start_pfn,
unsigned long nr_pages);
+extern int __remove_pages(struct zone *zone, unsigned long start_pfn,
+ unsigned long nr_pages);

+/*
+ * Walk through all memory which is registered as resource.
+ * Walk through all memory which is registered as resource.
+ * arg is (start_pfn, nr_pages, private_arg_pointer)
+ */
extern int walk_memory_resource(unsigned long start_pfn,
@@ -188,5 +191,6 @@ extern int arch_add_memory(int nid, u64
```

Re: [PATCH 1/5] generic __remove_pages() support

```
extern int remove_memory(u64 start, u64 size);
extern int sparse_add_one_section(struct zone *zone, unsigned long start_pfn,
int nr_pages);
+extern void sparse_remove_one_section(struct zone *zone, struct mem_section *ms);

#endif /* __LINUX_MEMORY_HOTPLUG_H */
```

--

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@xxxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>