Subject: [RFC 0/4] bypass charges if memcg is not used
Posted by Glauber Costa on Tue, 25 Sep 2012 08:52:49 GMT
View Forum Message <> Reply to Message

For the special case that memcg is compiled in (a quite common case) but not in use (also a common case), we are still seeing a huge performance impact. Mel Gorman demonstrated in the Kernel Summit, mm mini summit, that depending on the hardware, this can vary between 6 and 15 %.

I am proposing in this initial patch, that we take the strategy of bypassing the memcg code with static branches. We can flip it on when the first memcg gets created. Up to that moment, we'll mostly have a bunch of nops.

This patch also defers the call to page_cgroup_init() to that moment. This means that the memory used by page_cgroup structure won't be wasted until we really need it. We can do the same with the memory used for swap, if needed.

There are many edges to be trimmed, but I wanted to send this early to collect feedback. I coded this enough to get numbers out of it. I tested it in a 24-way 2-socket Intel box, 24 Gb mem. I used Mel Gorman's pft test, that he used to demonstrate this problem back in the Kernel Summit. There are three kernels:

nomemcg : memcg compile disabled.
base   : memcg enabled, patch not applied.
bypassed : memcg enabled, with patch applied.

	base	bypassed
User	109.12	105.64
System	1646.84	1597.98
Elapsed	229.56	215.76

	nomemcg	bypassed
User	104.35	105.64
System	1578.19	1597.98
Elapsed	212.33	215.76

So as one can see, the difference between base and nomemcg in terms of both system time and elapsed time is quite drastic, and consistent with the figures shown by Mel Gorman in the Kernel summit. This is a ~ 7 % drop in performance, just by having memcg enabled. memcg functions appear heavily in the profiles, even if all tasks lives in the root memcg.

With bypassed kernel, we drop this down to 1.5 %, which starts to fall in the acceptable range. More investigation is needed to see if we can claim that last percent back, but I believe at last part of it should be.

```
Glauber Costa (4):
  memcg: provide root figures from system totals
  memcg: make it suck faster
  memcg: do not call page_cgroup_init at system_boot
  memcg: do not walk all the way to the root for memcg

 include/linux/memcontrol.h  | 56 +++++++++++++++++++++++++++------
 include/linux/page_cgroup.h | 20 +++++++++---
 init/main.c                 |  1 -
 mm/memcontrol.c             | 78 ++++++++++++++++++++++++++++++++++++++-----
 mm/page_cgroup.c            | 44 ++++++++++++++++++------
 5 files changed, 165 insertions(+), 34 deletions(-)

--
1.7.11.4
```