

[PATCH] percpu: Introduce DEFINE_PER_CPU_PAGE_ALIGNED() macro

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-03/msg12044.html>

- *From:* Eric Dumazet <dada1@xxxxxxxxxxxxxx>
 - *Date:* Mon, 31 Mar 2008 17:22:22 +0200
-

While examining holes in percpu section I found this :

```
c05f5000 D per_cpu__current_task
c05f5000 D __per_cpu_start
c05f5004 D per_cpu__cpu_number
c05f5008 D per_cpu__irq_regs
c05f500c d per_cpu__cpu_devices
c05f5040 D per_cpu__cyc2ns
```

<Big Hole of about 4000 bytes>

```
c05f6000 d per_cpu__cpuid4_info
c05f6004 d per_cpu__cache_kobject
c05f6008 d per_cpu__index_kobject
```

<Big Hole of about 4000 bytes>

```
c05f7000 D per_cpu__gdt_page
```

This is because gdt_page is a percpu variable, defined with a page alignment, and linker is doing its job, two times because of .o nesting in the build process.

I introduced a new macro DEFINE_PER_CPU_PAGE_ALIGNED() to avoid wasting this space. All page aligned variables (only one at this time) are put in a separate subsection .data.percpu.page_aligned, at the very beginning of percpu zone.

Before patch , on a x86_32 machine :

```
# size -A vmlinux.old vmlinux | grep percpu
.data.percpu 30232 3227471872
.data.percpu 22168 3227471872
```

Thats 8064 bytes saved for each CPU.

Patch against x86.git/latest

Note : This will collide with Christoph efforts to put pda at offset 0 of percpu zone...

[PATCH] percpu: Introduce DEFINE_PER_CPU_PAGE_ALIGNED() macro

Signed-off-by: Eric Dumazet <dada1@xxxxxxxxxxxxxx>

```
arch/x86/kernel/cpu/common.c | 2 +-
arch/x86/kernel/vmlinux_32.lds.S | 1 +
include/asm-generic/vmlinux.lds.h | 1 +
include/linux/percpu.h | 7 ++++++
4 files changed, 10 insertions(+), 1 deletion(-)
```

```
diff --git a/arch/x86/kernel/cpu/common.c b/arch/x86/kernel/cpu/common.c
```

```
index b186047..c665dfc 100644
```

```
--- a/arch/x86/kernel/cpu/common.c
```

```
+++ b/arch/x86/kernel/cpu/common.c
```

```
@@ -21,7 +21,7 @@
```

```
#include "cpu.h"
```

```
-DEFINE_PER_CPU(struct gdt_page, gdt_page) = { .gdt = {
```

```
+DEFINE_PER_CPU_PAGE_ALIGNED(struct gdt_page, gdt_page) = { .gdt = {
```

```
[GDT_ENTRY_KERNEL_CS] = { { { 0x0000ffff, 0x00cf9a00 } } },
```

```
[GDT_ENTRY_KERNEL_DS] = { { { 0x0000ffff, 0x00cf9200 } } },
```

```
[GDT_ENTRY_DEFAULT_USER_CS] = { { { 0x0000ffff, 0x00cfa00 } } },
```

```
diff --git a/arch/x86/kernel/vmlinux_32.lds.S b/arch/x86/kernel/vmlinux_32.lds.S
```

```
index ce5ed08..0f7c29a 100644
```

```
--- a/arch/x86/kernel/vmlinux_32.lds.S
```

```
+++ b/arch/x86/kernel/vmlinux_32.lds.S
```

```
@@ -189,6 +189,7 @@ SECTIONS
```

```
. = ALIGN(PAGE_SIZE);
```

```
.data.percpu : AT(ADDR(.data.percpu) - LOAD_OFFSET) {
```

```
__per_cpu_start = .;
```

```
+ *(.data.percpu.page_aligned)
```

```
*(.data.percpu)
```

```
*(.data.percpu.shared_aligned)
```

```
__per_cpu_end = .;
```

```
diff --git a/include/asm-generic/vmlinux.lds.h b/include/asm-generic/vmlinux.lds.h
```

```
index f054778..69e5c11 100644
```

```
--- a/include/asm-generic/vmlinux.lds.h
```

```
+++ b/include/asm-generic/vmlinux.lds.h
```

```
@@ -348,6 +348,7 @@
```

```
. = ALIGN(align); \
```

```
__per_cpu_start = .; \
```

```
.data.percpu : AT(ADDR(.data.percpu) - LOAD_OFFSET) { \
```

```
+ *(.data.percpu.page_aligned) \
```

```
*(.data.percpu) \
```

```
*(.data.percpu.shared_aligned) \
```

```
} \
```

```
diff --git a/include/linux/percpu.h b/include/linux/percpu.h
```

```
index 1ac9697..1c0cc0d 100644
```

```
--- a/include/linux/percpu.h
```

```
+++ b/include/linux/percpu.h
```

```
@@ -18,12 +18,19 @@
```

[PATCH] percpu: Introduce DEFINE_PER_CPU_PAGE_ALIGNED() macro

[PATCH] percpu: Introduce DEFINE_PER_CPU_PAGE_ALIGNED() macro

```
__attribute__((__section__(".data.percpu.shared_aligned"))) \
PER_CPU_ATTRIBUTES __typeof__(type) per_cpu_##name \
__cacheline_aligned_in_smp
+
+#define DEFINE_PER_CPU_PAGE_ALIGNED(type, name) \
+ __attribute__((__section__(".data.percpu.page_aligned"))) \
+ PER_CPU_ATTRIBUTES __typeof__(type) per_cpu_##name
#else
#define DEFINE_PER_CPU(type, name) \
PER_CPU_ATTRIBUTES __typeof__(type) per_cpu_##name

#define DEFINE_PER_CPU_SHARED_ALIGNED(type, name) \
DEFINE_PER_CPU(type, name)
+
+#define DEFINE_PER_CPU_PAGE_ALIGNED(type, name) \
+ DEFINE_PER_CPU(type, name)
#endif

#define EXPORT_PER_CPU_SYMBOL(var) EXPORT_SYMBOL(per_cpu_##var)
```