



Contribution ID: 103

Type: **not specified**

## Memory management bits in arch/\*

*Tuesday, September 10, 2019 10:00 AM (45 minutes)*

There is a lot of similar and duplicated code in architecture specific bits of memory management.

For instance, most architectures have

```
\#define PGALLOC\GFP (GFP\KERNEL | \_\GFP\ZERO)
```

for allocating page table pages and many of them use similar, if not identical, implementation of `pte_alloc_one*`).

But that's only the tip of the iceberg.

There are several `early_alloc()` or similarly called routines that do

```
if (slab\is\available())
    return kzalloc();
else
    return memblock\_alloc();
```

Some other trivial examples are `free_initmem()`, `free_initrd_mem()` which were nearly identical across many architectures until very recently.

More complex cases are per-cpu initialization, passing of memory topology to the generic MM, reservation of crash kernel, mmap of vdso etc. They are not really duplicated, but still are very similar in at least several architectures.

While factoring out the common code is an obvious step to take, I believe there is also room for refining arch <-> mm interface to avoid adding extra `HAVE_ARCH_NO_BOOTMEM^w^wWHAT_NOT` and then searching for the ways to get rid of them.

### I agree to abide by the anti-harassment policy

Yes

### I confirm that I am already registered for LPC 2019

**Primary author:** RAPOPORT, Mike (IBM)

**Presenter:** RAPOPORT, Mike (IBM)

**Session Classification:** Kernel Summit Track

**Track Classification:** Kernel Summit talk