CMA(Contiguous Memory Allocator) for DMA

2016-03-10 (http://jake.dothome.co.kr/cma-dma/) Moon Young-il (http://jake.dothome.co.kr/author/admin/) Linux Kernel (http://jake.dothome.co.kr/category/linux/)

The DMA area used by the device driver is registered with the Contiguous Memory Allocator (CMA), a reserve memblock is added, and memory allocation management of the area is performed.

dma_contiguous_reserve()

drivers/base/dma-contiguous.c

```
/**
 * dma_contiguous_reserve() - reserve area(s) for contiguous memory handling
 * @limit: End address of the reserved memory (optional, 0 for any).
 *
 * This function reserves memory from early allocator. It should be
 * called by arch specific code once the early allocator (memblock or bootmem)
 * has been activated and all other subsystems have already allocated/reserved
 * memory.
 */
void __init dma_contiguous_reserve(phys_addr_t limit)
{
        phys_addr_t selected_size = 0;
        phys_addr_t selected_base = 0;
        phys_addr_t selected_limit = limit;
        bool fixed = false;

        pr_debug("%s(limit %08lx)\n", __func__, (unsigned long)limit);

        if (size_cmdline != -1) {
                selected_size = size_cmdline;
                selected_base = base_cmdline;
                selected_limit = min_not_zero(limit_cmdline, limit);
                if (base_cmdline + size_cmdline == limit_cmdline)
                        fixed = true;
        } else {
#ifdef CONFIG_CMA_SIZE_SEL_MBYTES
                selected_size = size_bytes;
#elif defined(CONFIG_CMA_SIZE_SEL_PERCENTAGE)
                selected_size = cma_early_percent_memory();
#elif defined(CONFIG_CMA_SIZE_SEL_MIN)
                selected_size = min(size_bytes, cma_early_percent_memory());
#elif defined(CONFIG_CMA_SIZE_SEL_MAX)
                selected_size = max(size_bytes, cma_early_percent_memory());
#endif
        }

        if (selected_size && !dma_contiguous_default_area) {
                pr_debug("%s: reserving %ld MiB for global area\n", __func__,
                         (unsigned long)selected_size / SZ_1M);

                dma_contiguous_reserve_area(selected_size, selected_base,
                                            selected_limit,
                                            &dma_contiguous_default_area,
                                            fixed);
        }
}
```

Add the area equal to the memory size requested by kernel parameters or kernel options to the Contigugou Memory Allocator (CMA) as a management item, but only once it is requested.

- When the early_cma() routine is called, it knows the size_cmdline, base_cmdline, limit_cmdline, etc.
    - If the request area is the same as limit, set fixed to true so that it can be assigned to the exact base (start address).
    - If fixed is false, the request is made from base to limit size.
- if (size_cmdline != -1) {
    - If you know size_cmdline information, use the parsed global variable values.
- CONFIG_CMA_SIZE_SEL_MBYTES
    - If you don't know the size_cmdline information, use the value set as a kernel option.
    - size_bytes = CMA_SIZE_MBYTES * SZ_1M
        - CMA_SIZE_MBYTES = CONFIG_CMA_SIZE_MBYTES
- if (selected_size && !dma_contiguous_default_area) {
    - When the size is greater than zero and the CMA management area is first set up (null)
- dma_contiguous_reserve_area(selected_size, selected_base, selected_limit, &dma_contiguous_default_area, fixed);
    - Add management items to the CMA with the memory start address, size, limit, and fixed values to configure the CMA, and also configure the remapping information for future remapping.
    - It is added to the CMA management section and to the Reserve Memblock.

dma_contiguous_reserve_area()

drivers/base/dma-contiguous.c

```
/**
 * dma_contiguous_reserve_area() - reserve custom contiguous area
 * @size: Size of the reserved area (in bytes),
 * @base: Base address of the reserved area optional, use 0 for any
 * @limit: End address of the reserved memory (optional, 0 for any).
 * @res_cma: Pointer to store the created cma region.
 * @fixed: hint about where to place the reserved area
 *
 * This function reserves memory from early allocator. It should be
 * called by arch specific code once the early allocator (memblock or bootmem)
 * has been activated and all other subsystems have already allocated/reserved
 * memory. This function allows to create custom reserved areas for specific
 * devices.
 *
 * If @fixed is true, reserve contiguous area at exactly @base.  If false,
 * reserve in range from @base to @limit.
 */
int __init dma_contiguous_reserve_area(phys_addr_t size, phys_addr_t base,
                                       phys_addr_t limit, struct cma **res_cma,
                                       bool fixed)
{
        int ret;

        ret = cma_declare_contiguous(base, size, limit, 0, 0, fixed, res_cma);
        if (ret)
                return ret;

        /* Architecture specific contiguous memory fixup. */
        dma_contiguous_early_fixup(cma_get_base(*res_cma),
                                cma_get_size(*res_cma));

        return 0;
}
```

- ret = cma_declare_contiguous(base, size, limit, 0, 0, fixed, res_cma);
    - Add management entries to the cma_areas[] and set the base_pfn, count, bitmap assignment, alignment, and order_per_bit with the memory size information to manage: base, size, limit, alignment, and order_per_bit.
        - The bitmaps used by CMA allow each bit to represent one page.
        - The alignment unit of the domain is made to one page.
        - The area is also registered in the reserve memblock.
    - If it fails, it returns an error.
    - The entries added are initialized by calling them when the CMA driver is loaded.
- dma_contiguous_early_fixup(cma_get_base(*res_cma), cma_get_size(*res_cma));
    - Add the area of memory to be managed by CMA to the dma_mmu_remap[] array.
    - This simply adds the information to the array and later calls the functions in the order setup_arch() -> paging_init() -> dma_contiguous_remap() to remap it.

dma_contiguous_early_fixup()

arch/arm/mm/dma-mapping.c

```
void __init dma_contiguous_early_fixup(phys_addr_t base, unsigned long size)
{
        dma_mmu_remap[dma_mmu_remap_num].base = base;
        dma_mmu_remap[dma_mmu_remap_num].size = size;
        dma_mmu_remap_num++;
}
```

- dma_mmu_remap[] array.

CMA Memory Early Setup Routine

early_cma()

"cma=" is called and used by the kernel parameter.

drivers/base/dma-contiguous.c

```c
/*
 * Default global CMA area size can be defined in kernel's .config.
 * This is useful mainly for distro maintainers to create a kernel
 * that works correctly for most supported systems.
 * The size can be set in bytes or as a percentage of the total memory
 * in the system.
 *
 * Users, who want to set the size of global CMA area for their system
 * should use cma= kernel parameter.
 */
static const phys_addr_t size_bytes = CMA_SIZE_MBYTES * SZ_1M;
static phys_addr_t size_cmdline = -1;
static phys_addr_t base_cmdline;
static phys_addr_t limit_cmdline;

static int __init early_cma(char *p)
{
        pr_debug("%s(%s)\n", __func__, p);
        size_cmdline = memparse(p, &p);
        if (*p != '@')
                return 0;
        base_cmdline = memparse(p + 1, &p);
        if (*p != '-') {
                limit_cmdline = base_cmdline + size_cmdline;
                return 0;
        }
        limit_cmdline = memparse(p + 1, &p);

        return 0;
}
early_param("cma", early_cma);
```

Here's an example:

- cma=4M
  - Within the 0x0000_0000 ~ arm_dma_limit (DMA size) area, 4M of space is allocated for CMA.
- cma=4M@0x00c00000
  - Within the 0x00c0_0000 ~ arm_dma_limit (DMA size) area, 4M of space for CMA is allocated.
- cma=4M@0x10000000-0x40000000
  - Within the 0x1000_0000 ~ 0x4000_0000 area, 4M of space is allocated for CMA.
- cma=4M@0x10000000-0x10400000
  - ALLOCATE 0 METERS OF SPACE FOR CMA AT EXACTLY 1000x0000_4 POSITION. (fixed location)

Structs and Global Variables

dma_contig_early_reserve Struct

arch/arm/mm/dma-mapping.c

```c
struct dma_contig_early_reserve {
        phys_addr_t base;
        unsigned long size;
};
```

Global Variables

arch/arm/mm/dma-mapping.c

```c
static struct dma_contig_early_reserve dma_mmu_remap[MAX_CMA_AREAS] __initdata;
static int dma_mmu_remap_num __initdata;
```

- dma_mmu_remap[]
  - MAX_CMA_AREAS is a CONFIG_CMA_AREAS kernel option + 1 value.
- dma_mmu_remap_num
  - Number of items to remap

consultation

- CMA(Contiguous Memory Allocator) (http://jake.dothome.co.kr/cma) | 문c
- dma_contiguous_remap()

LEAVE A COMMENT

Your email will not be published. Required fields are marked with *

Comments

name *

email *

Website

WRITE A COMMENT

< CMA(Contiguous Memory Allocator) (http://jake.dothome.co.kr/cma/)

build_mem_type_table() ❯ (http://jake.dothome.co.kr/build_mem_type_table/)

Munc Blog (2015 ~ 2023)