

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

```

01  /**
02   * dma_contiguous_reserve() - reserve area(s) for contiguous memory handling
03   * @limit: End address of the reserved memory (optional, 0 for any).
04   *
05   * This function reserves memory from early allocator. It should be
06   * called by arch specific code once the early allocator (memblock or bootmem)
07   * has been activated and all other subsystems have already allocated/reserved
08   * memory.
09   */
10  void __init dma_contiguous_reserve(phys_addr_t limit)
11  {
12      phys_addr_t selected_size = 0;
13      phys_addr_t selected_base = 0;
14      phys_addr_t selected_limit = limit;
15      bool fixed = false;
16
17      pr_debug("%s(limit %08lx)\n", __func__, (unsigned long)limit);
18
19      if (size_cmdline != -1) {
20          selected_size = size_cmdline;
21          selected_base = base_cmdline;
22          selected_limit = min_not_zero(limit_cmdline, limit);
23          if (base_cmdline + size_cmdline == limit_cmdline)
24              fixed = true;
25      } else {
26#ifdef CONFIG_CMA_SIZE_SEL_MBYTES
27          selected_size = size_bytes;
28#elif defined(CONFIG_CMA_SIZE_SEL_PERCENTAGE)
29          selected_size = cma_early_percent_memory();
30#elif defined(CONFIG_CMA_SIZE_SEL_MIN)
31          selected_size = min(size_bytes, cma_early_percent_memory());
32#elif defined(CONFIG_CMA_SIZE_SEL_MAX)
33          selected_size = max(size_bytes, cma_early_percent_memory());
34#endif
35      }
36
37      if (selected_size && !dma_contiguous_default_area) {
38          pr_debug("%s: reserving %ld MiB for global area\n", __func__,
39                  (unsigned long)selected_size / SZ_1M);
40      }

```

```

41 | dma_contiguous_reserve_area(selected_size, selected_base,
42 |                             selected_limit,
43 |                             &dma_contiguous_default_area,
44 |                             fixed);
45 | }
46 | }

```

Add the area equal to the memory size requested by kernel parameters or kernel options to the Contiguous 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

```

01 | /**
02 |  * dma_contiguous_reserve_area() - reserve custom contiguous area
03 |  * @size: Size of the reserved area (in bytes),
04 |  * @base: Base address of the reserved area optional, use 0 for any
05 |  * @limit: End address of the reserved memory (optional, 0 for any).
06 |  * @res_cma: Pointer to store the created cma region.
07 |  * @fixed: hint about where to place the reserved area
08 |  *
09 |  * This function reserves memory from early allocator. It should be
10 |  * called by arch specific code once the early allocator (memblock or bo
11 |  * otmem)
12 |  * has been activated and all other subsystems have already allocated/re
13 |  * served
14 |  * memory. This function allows to create custom reserved areas for spec
15 |  * ific
16 |  * devices.
17 |  *
18 |  * If @fixed is true, reserve contiguous area at exactly @base. If fals
19 |  * e,

```

```

16  * reserve in range from @base to @limit.
17  */
18  int __init dma_contiguous_reserve_area(phys_addr_t size, phys_addr_t base,
19                                          phys_addr_t limit, struct cma **res_cma,
20                                          bool fixed)
21  {
22      int ret;
23
24      ret = cma_declare_contiguous(base, size, limit, 0, 0, fixed, res_cma);
25      if (ret)
26          return ret;
27
28      /* Architecture specific contiguous memory fixup. */
29      dma_contiguous_early_fixup(cma_get_base(*res_cma),
30                                cma_get_size(*res_cma));
31
32      return 0;
33  }

```

- 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

```

1  void __init dma_contiguous_early_fixup(phys_addr_t base, unsigned long size)
2  {
3      dma_mmu_remap[dma_mmu_remap_num].base = base;
4      dma_mmu_remap[dma_mmu_remap_num].size = size;
5      dma_mmu_remap_num++;
6  }

```

- 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

```

01  /*
02  * Default global CMA area size can be defined in kernel's .config.
03  * This is useful mainly for distro maintainers to create a kernel
04  * that works correctly for most supported systems.
05  * The size can be set in bytes or as a percentage of the total memory
06  * in the system.
07  *
08  * Users, who want to set the size of global CMA area for their system
09  * should use cma= kernel parameter.
10  */
11  static const phys_addr_t size_bytes = CMA_SIZE_MBYTES * SZ_1M;
12  static phys_addr_t size_cmdline = -1;
13  static phys_addr_t base_cmdline;
14  static phys_addr_t limit_cmdline;
15
16  static int __init early_cma(char *p)
17  {
18      pr_debug("%s(%s)\n", __func__, p);
19      size_cmdline = memparse(p, &p);
20      if (*p != '@')
21          return 0;
22      base_cmdline = memparse(p + 1, &p);
23      if (*p != '-') {
24          limit_cmdline = base_cmdline + size_cmdline;
25          return 0;
26      }
27      limit_cmdline = memparse(p + 1, &p);
28
29      return 0;
30  }
31  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

```

1  struct dma_contig_early_reserve {
2      phys_addr_t base;
3      unsigned long size;
4  };

```

Global Variables

arch/arm/mm/dma-mapping.c

```
1 | static struct dma_contig_early_reserve dma_mmu_remap[MAX_CMA_AREAS] __in  
  | itdata;  
2 | 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\(Contiguous Memory Allocator\) \(http://jake.dothome.co.kr/cma/\)](http://jake.dothome.co.kr/cma/)

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

Munc Blog (2015 ~ 2024)