

Linux CPU 拓扑结构之调度域 | 调度组 - 以8核ARM big.Little架构处理器为例

CPU拓扑结构简介

- SMT Level 超线程处理器的一个核心
- MC Level 多核CPU的一个核心
- DIE Level 一个物理CPU的晶片（注意不是package，package是封装好了的，肉眼看到的CPU处理器）

cpu最小级别的就是超线程处理器的一个smt核，次小的一级就是一个多核cpu的核，然后就是一个物理cpu封装，再往后就是cpu阵列，根据这些cpu级别的不同，Linux将所有同一级别的cpu归为一个“调度组”，然后将同一级别的所有的调度组组成一个“调度域”cpu最小级别的就是超线程处理器的一个smt核，次小的一级就是一个多核cpu的核，然后就是一个物理cpu封装，再往后就是cpu阵列，根据这些cpu级别的不同，Linux将所有同一级别的cpu归为一个“调度组”，然后将同一级别的所有的调度组组成一个“调度域”

对于ARM 架构，目前由于不支持超线程技术，只有DIE和MC两个Topology Level

```
/*
 * Topology list, bottom-up.
 */
static struct sched_domain_topology_level default_topology[] = {
#ifdef CONFIG_SCHED_SMT
    { cpu_smt_mask, cpu_smt_flags, SD_INIT_NAME(SMT) },
#endif
#ifdef CONFIG_SCHED_MC
    { cpu_coregroup_mask, cpu_core_flags, SD_INIT_NAME(MC) },
#endif
    { cpu_cpu_mask, SD_INIT_NAME(DIE) },
    { NULL, },
};

static struct sched_domain_topology_level *sched_domain_topology =
    default_topology;

#define for_each_sd_topology(tl) \
    for (tl = sched_domain_topology; tl->mask; tl++)

void set_sched_topology(struct sched_domain_topology_level *tl)
{
    sched_domain_topology = tl;
}
```

拓扑结构图

以一个现在典型的8核心，4小核4大核的ARM架构的处理器为例

