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)

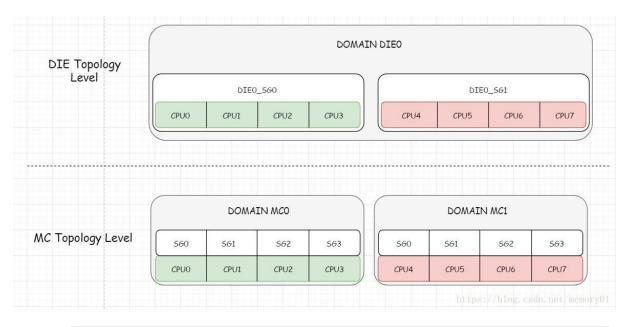
cpu最小级别的就是超线程处理器的一个smt核,次小的一级就是一个多核cpu的核,然后就是一个物理cpu封装,再往后就是cpu阵列,根据这些cpu级别的不同,Linux将所有同一级别的cpu归为一个"调度组",然后将同一级别的所有的调度组组成一个"调度域"cpu最小级别的就是超线程处理器的一个smt核,次小的一级就是一个多核cpu的核,然后就是一个物理cpu封装,再往后就是cpu阵列,根据这些cpu级别的不同,Linux将所有同一级别的cpu归为一个"调度组",然后将同一级别的所有的调度组组成一个"调度域"

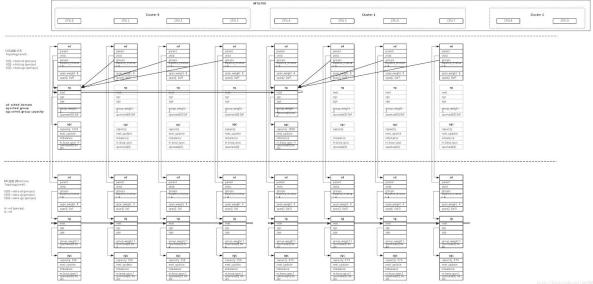
对于ARM 架构,目前由于不支持超线程技术,只有DIE和MC两个Topology Level

```
* 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





需要明白下面几个结构体元素的含义:

};