



Contribution ID: 189

Type: **not specified**

## Consolidating representations of the physical memory

*Tuesday, 13 September 2022 11:00 (30 minutes)*

We have several coarse representations of the physical memory consisting of [start, end, flags] structures per memory region. There is memblock that some architectures keep after boot, there is iomem\_resource tree and “System RAM” nodes in that tree, there are memory blocks exposed in sysfs and then there are per-architecture structures, sometimes even several per architecture.

The multiplication of such structures and lack of consistency between some of them does not help the maintainability and can be a reason for subtle bugs here and there.

The layout of the physical memory is defined by hardware and firmware and there is not much room for its interpretation; single abstraction of the physical memory should suffice and a single [start, end, flags] type should be enough. There is no fundamental reason why we cannot converge per-architecture representations of the physical memory, like e820, drmmem\_lmb, memblock or numa\_meminfo into a generic abstraction.

I suggest to use memblock as the basis for such abstraction. It is already supported on all architectures and it is used as the generic representation of the physical memory at boot time. Closing the gaps between per architecture structures and memblock is anyway required for more robust initialization of the memory management. Addition of simple locking of memblock data for memory hotplug, making the memblock “allocator” part discardable and a mechanism to synchronize “System RAM” resources with memblock would complete the picture.

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

Yes

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

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

**Session Classification:** linux/arch MC

**Track Classification:** LPC Microconference: linux/arch MC