

# Memory management bits in arch/

Mike Rapoport  
[<rppt@linux.ibm.com>](mailto:rppt@linux.ibm.com)



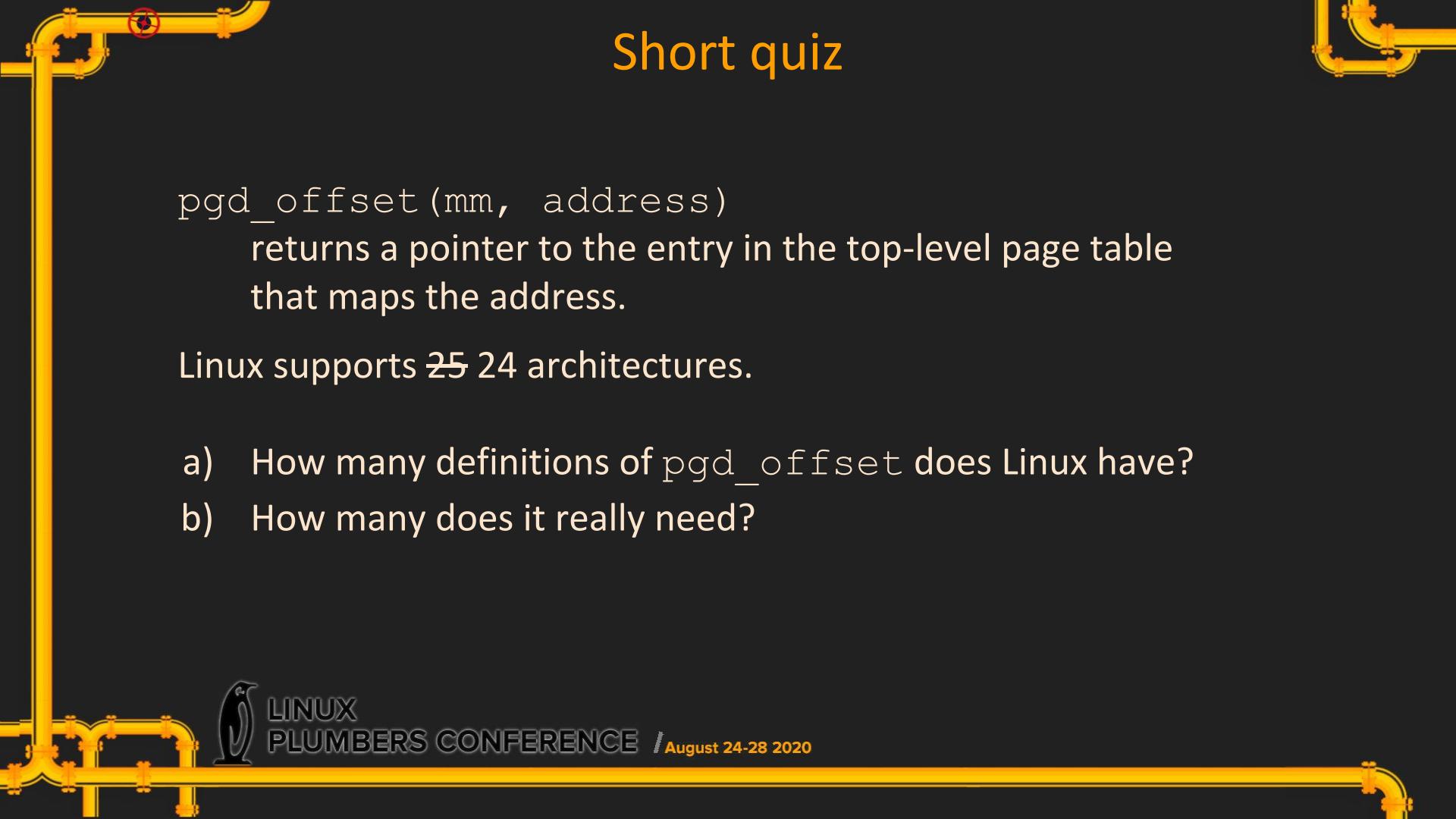
This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 825377





# When arch/ meets mm/

- TLB management
- Page table manipulations
- Memory models
- Memory detection and initialization
  - Cold and hot (un)plug



# Short quiz

`pgd_offset(mm, address)`

returns a pointer to the entry in the top-level page table  
that maps the address.

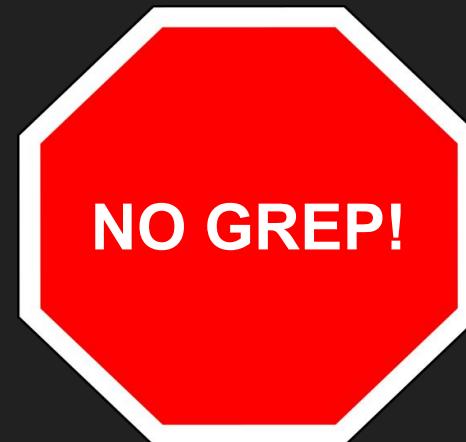
Linux supports ~~25~~ 24 architectures.

- a) How many definitions of `pgd_offset` does Linux have?
- b) How many does it really need?



# Short quiz

- A. 31
- B. 24
- C. 2
- D. 1

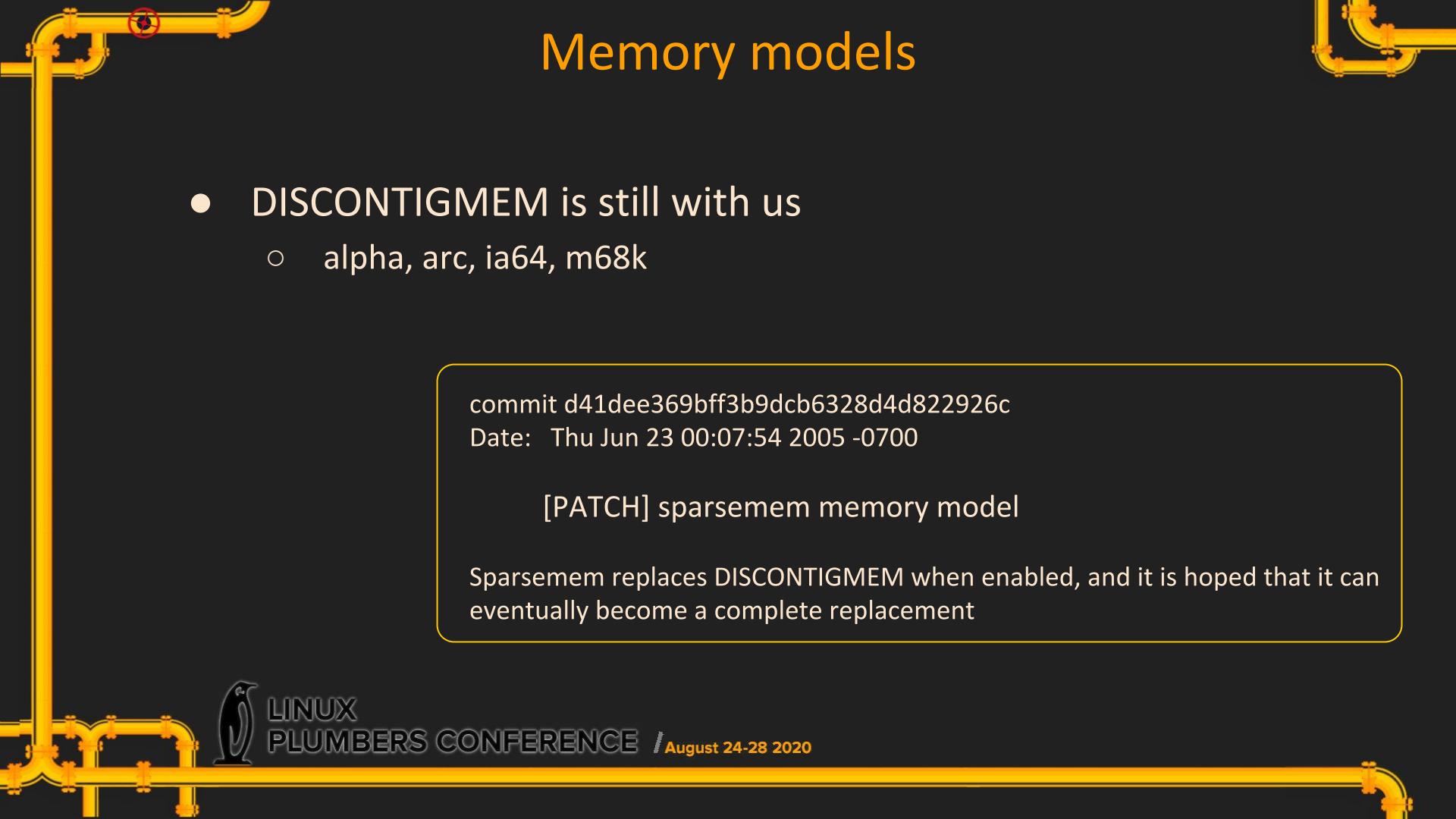




# Page table manipulation



- Folding is neat, but...
  - Lots of “empty” lines
  - Tree-wide updates for each new level (once in couple of years)
- Possible alternatives:
  - Use page walk
  - Completely new interface, e.g.  
`vpte_for_each(vpte, start, end, flags)`
- Split and clean `asm/page.h` and `asm/pgtable.h`
  - For instance, like x86...



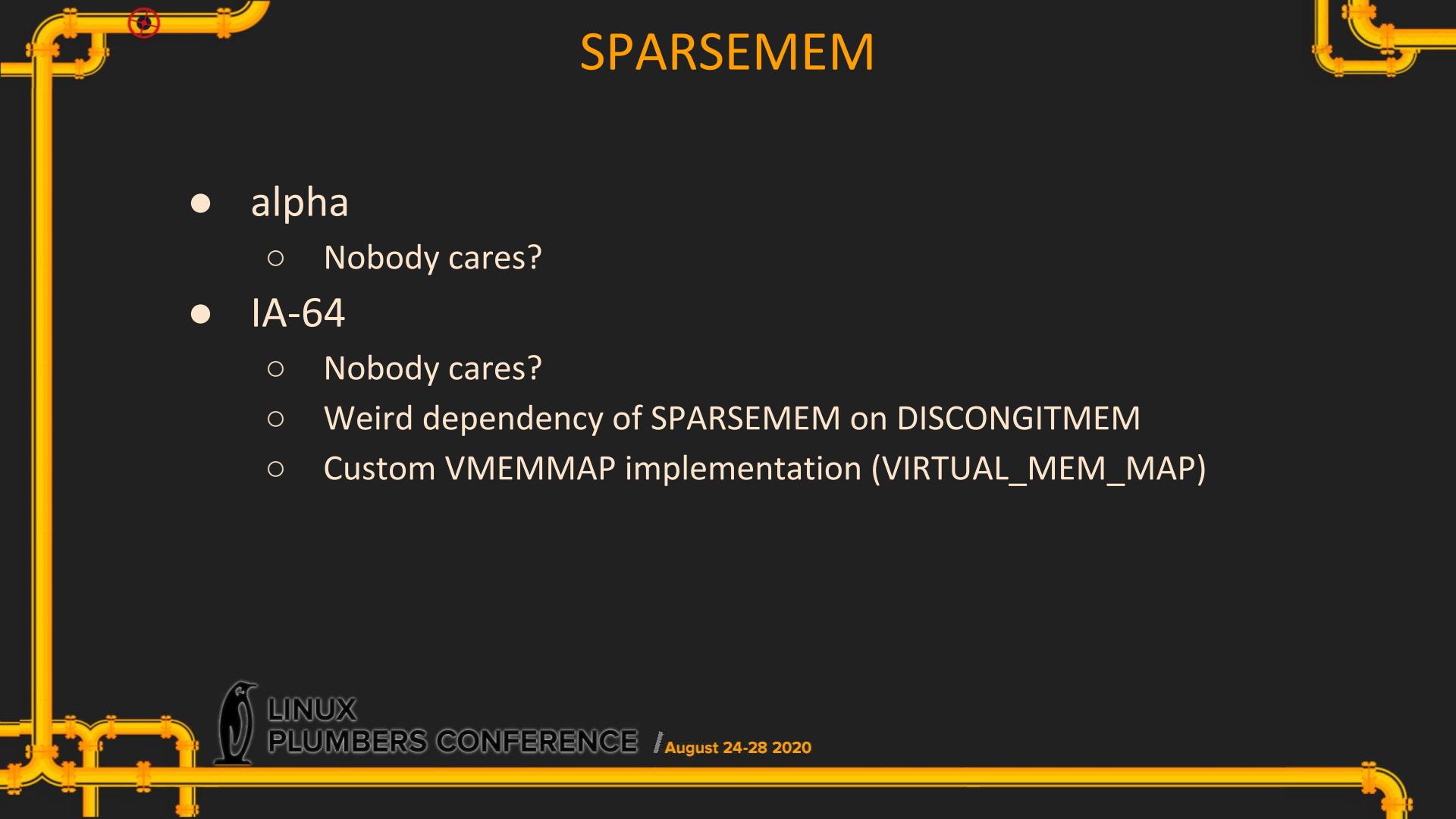
# Memory models

- DISCONTIGMEM is still with us
  - alpha, arc, ia64, m68k

```
commit d41dee369bff3b9dcb6328d4d822926c  
Date: Thu Jun 23 00:07:54 2005 -0700
```

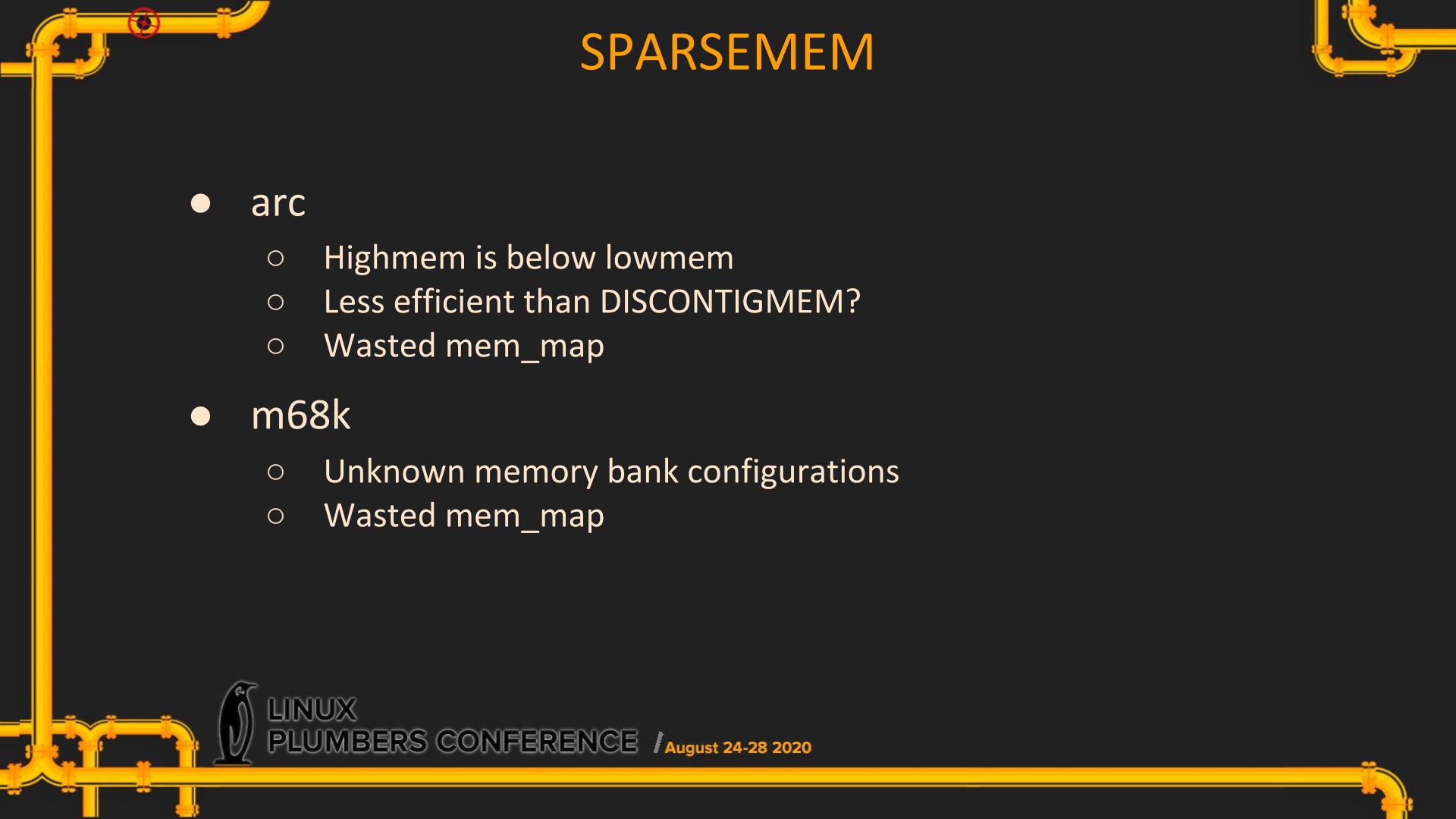
[PATCH] sparsemem memory model

Sparsemem replaces DISCONTIGMEM when enabled, and it is hoped that it can eventually become a complete replacement



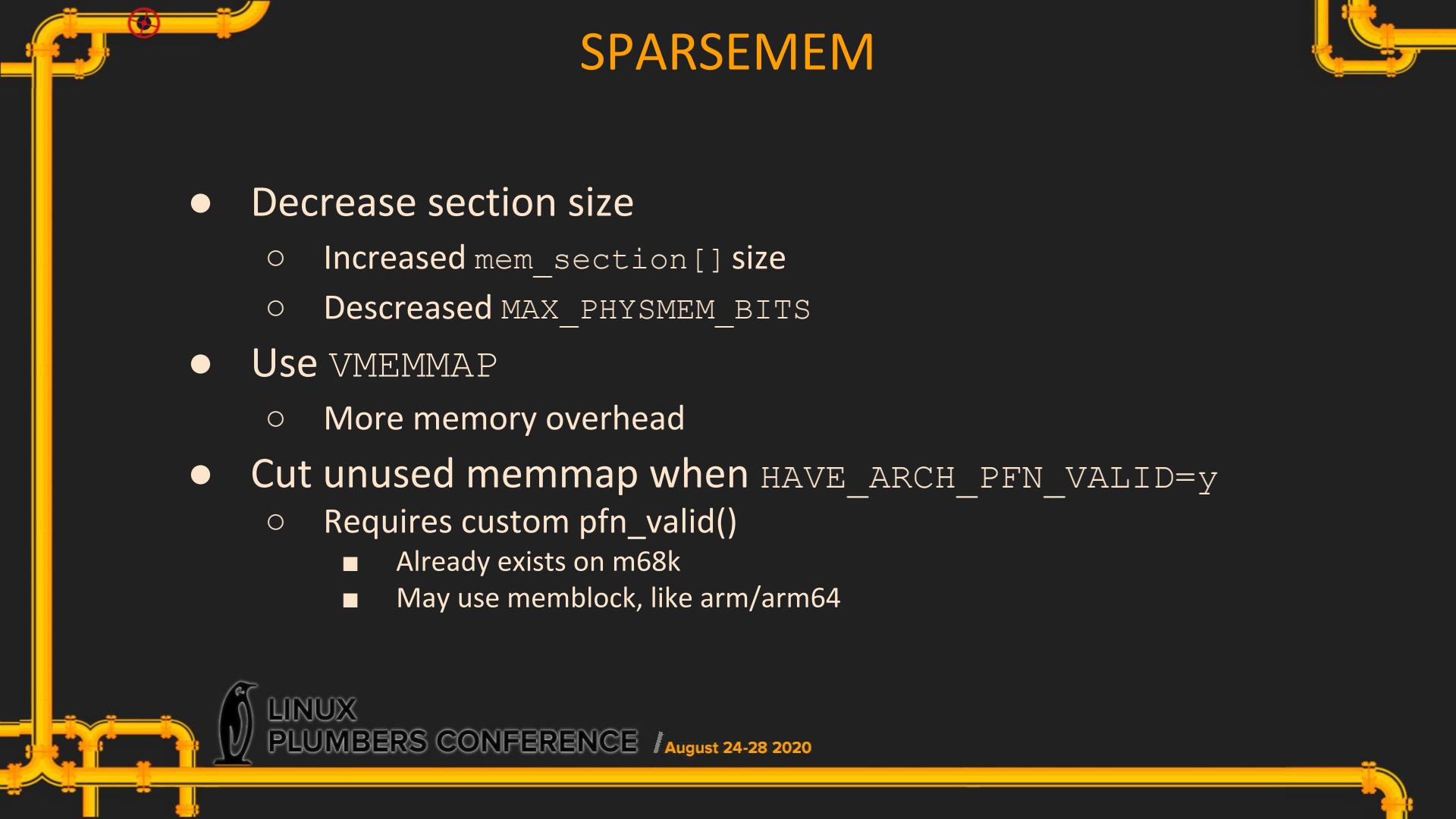
# SPARSEMEM

- alpha
  - Nobody cares?
- IA-64
  - Nobody cares?
  - Weird dependency of SPARSEMEM on DISCONGITMEM
  - Custom VMEMMAP implementation (VIRTUAL\_MEM\_MAP)



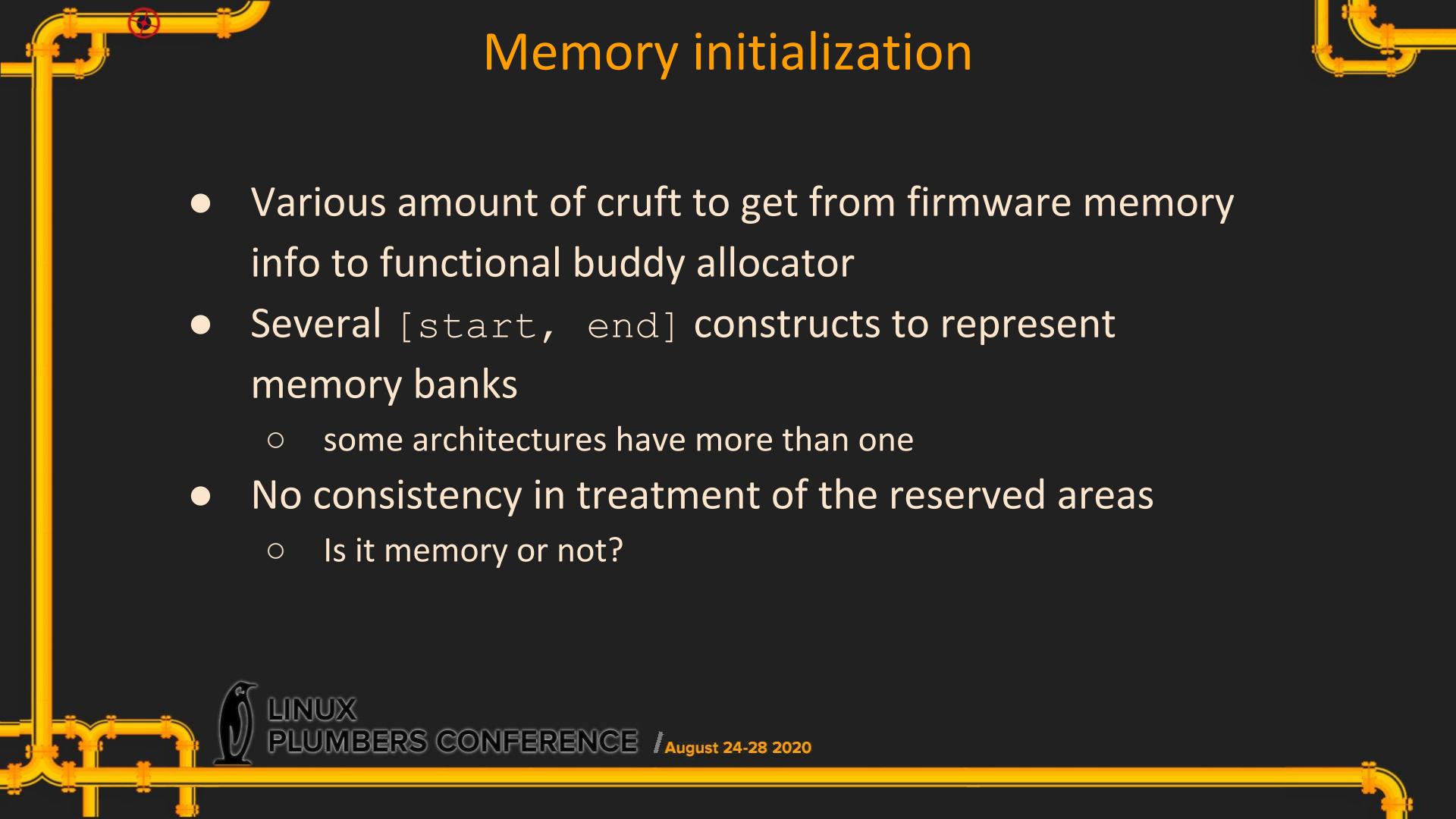
# SPARSEMEM

- arc
  - Highmem is below lowmem
  - Less efficient than DISCONTIGMEM?
  - Wasted mem\_map
- m68k
  - Unknown memory bank configurations
  - Wasted mem\_map



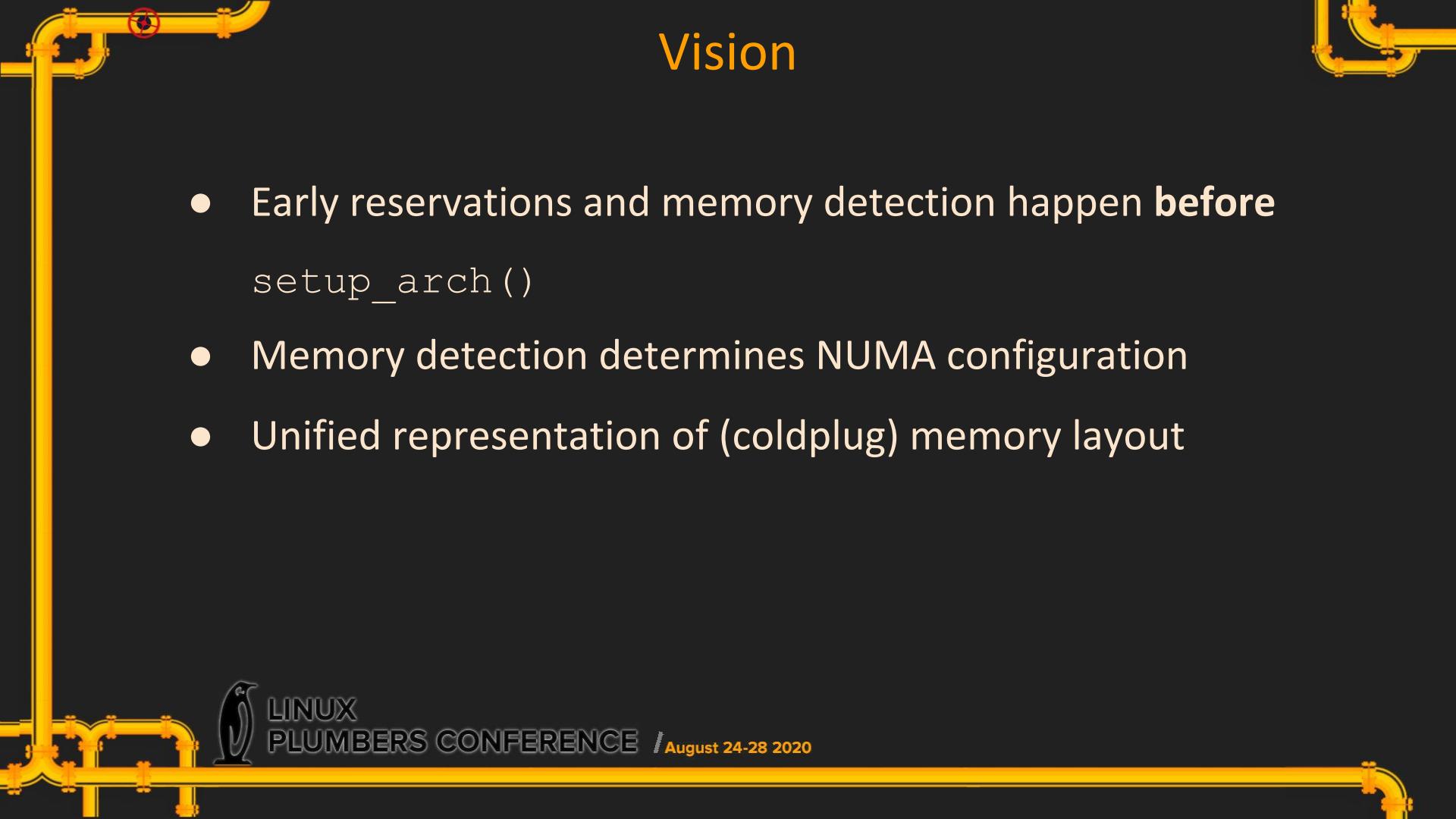
# SPARSEMEM

- Decrease section size
  - Increased `mem_section[] size`
  - Descreased `MAX_PHYSMEM_BITS`
- Use `VMMEMMAP`
  - More memory overhead
- Cut unused memmap when `HAVE_ARCH_PFN_VALID=y`
  - Requires custom `pfn_valid()`
    - Already exists on m68k
    - May use memblock, like arm/arm64



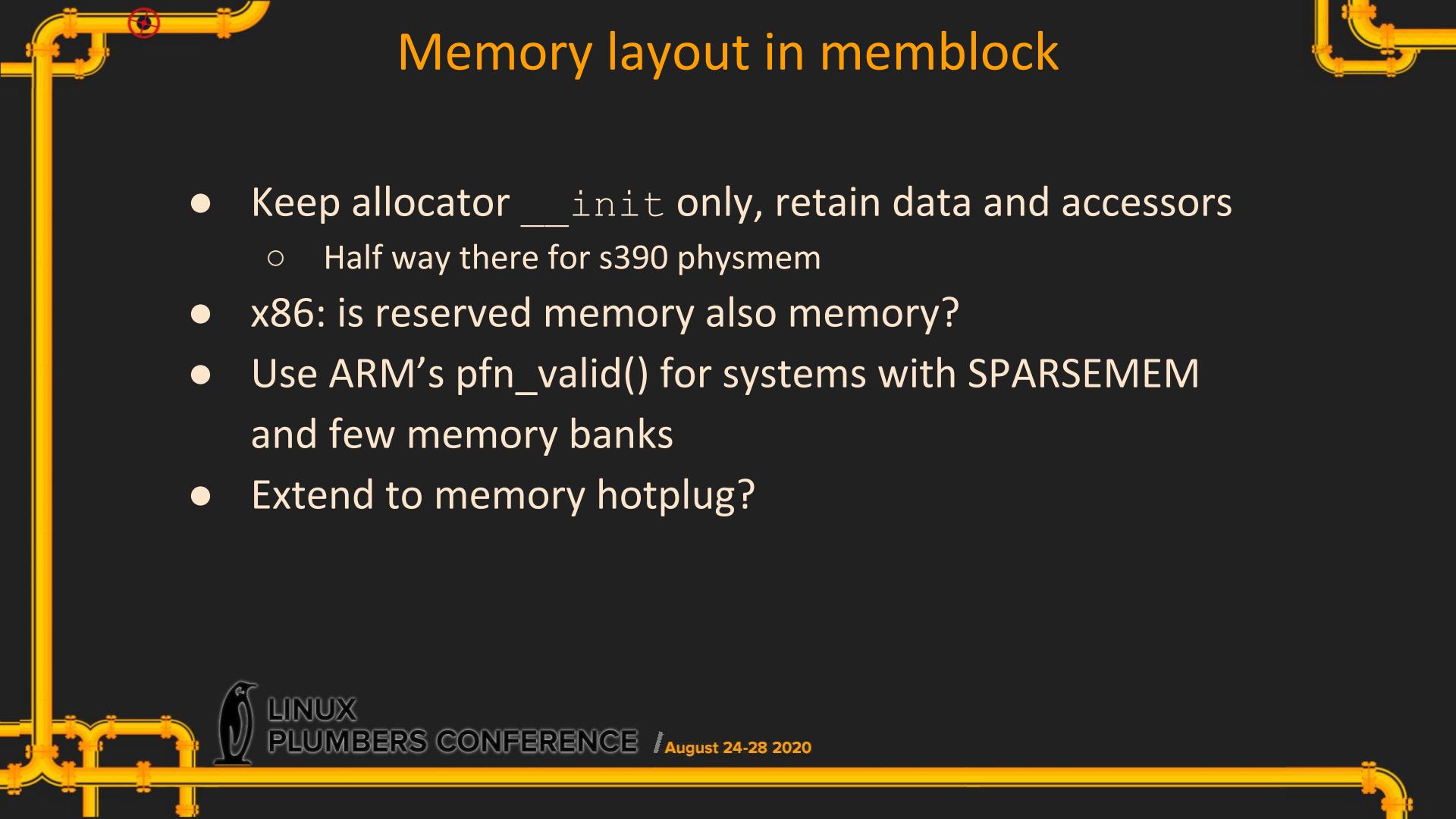
# Memory initialization

- Various amount of cruft to get from firmware memory info to functional buddy allocator
- Several [start, end] constructs to represent memory banks
  - some architectures have more than one
- No consistency in treatment of the reserved areas
  - Is it memory or not?



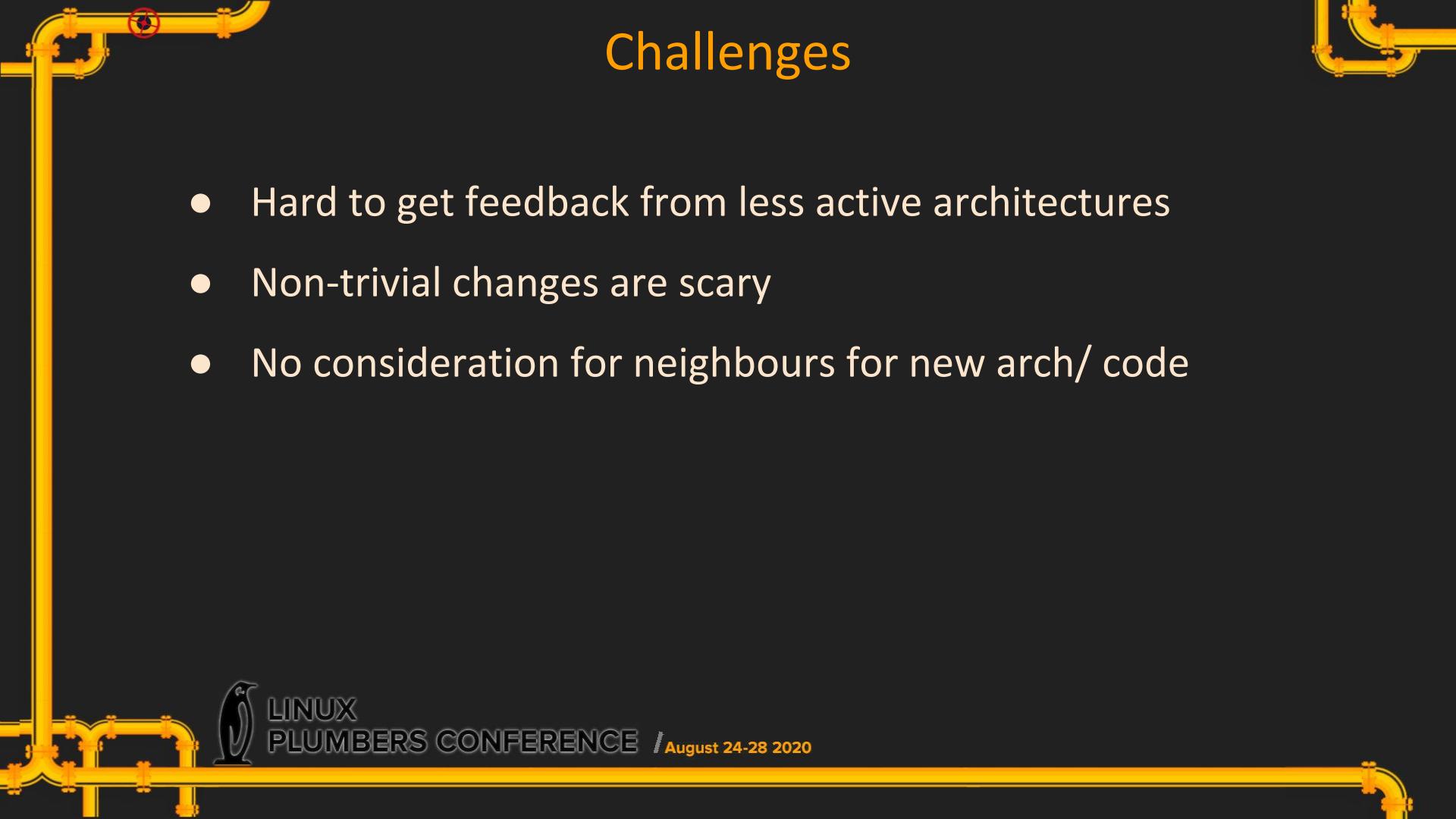
# Vision

- Early reservations and memory detection happen **before** `setup_arch()`
- Memory detection determines NUMA configuration
- Unified representation of (coldplug) memory layout



# Memory layout in memblock

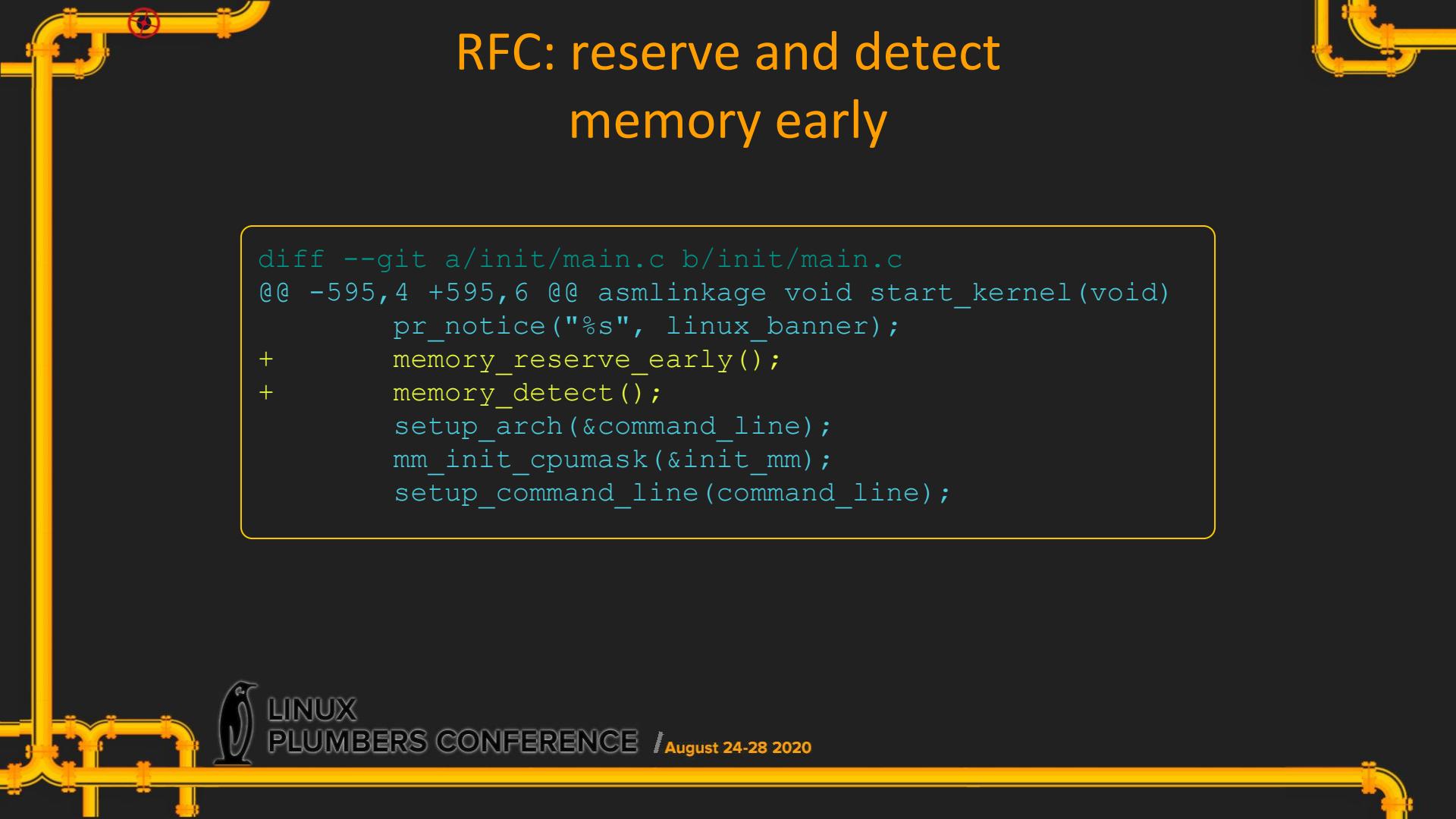
- Keep allocator `__init` only, retain data and accessors
  - Half way there for s390 physmem
- x86: is reserved memory also memory?
- Use ARM's `pfn_valid()` for systems with SPARSEMEM and few memory banks
- Extend to memory hotplug?

A decorative border on the left and right sides of the slide consists of stylized yellow pipes with orange valves and fittings, resembling a plumbing system.

# Challenges

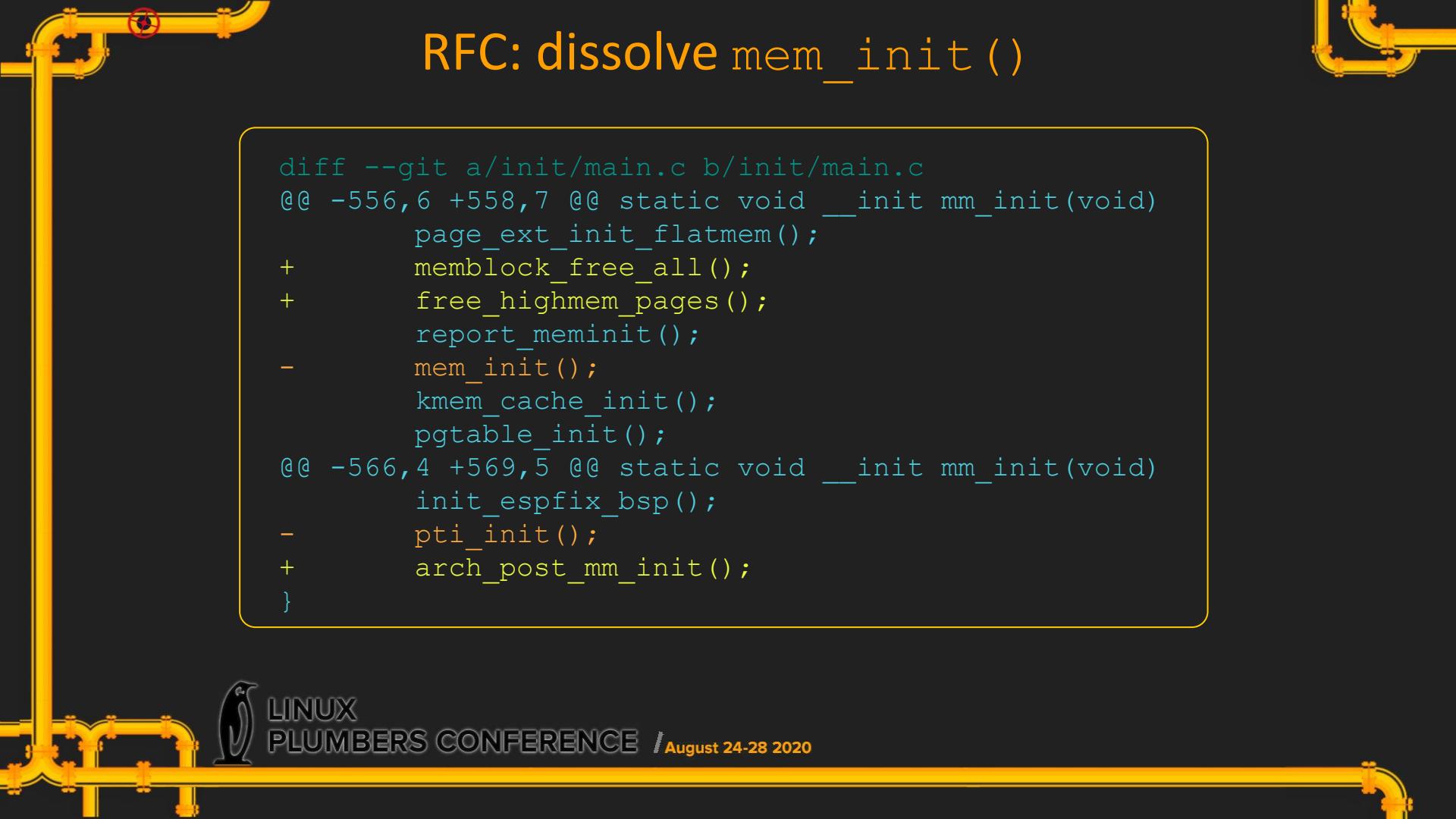
- Hard to get feedback from less active architectures
- Non-trivial changes are scary
- No consideration for neighbours for new arch/ code

Thank you!



# RFC: reserve and detect memory early

```
diff --git a/init/main.c b/init/main.c
@@ -595,4 +595,6 @@ asmlinkage void start_kernel(void)
        pr_notice("%s", linux_banner);
+
+    memory_reserve_early();
+
+    memory_detect();
    setup_arch(&command_line);
    mm_init_cpumask(&init_mm);
    setup_command_line(command_line);
```



# RFC: dissolve mm\_init()

```
diff --git a/init/main.c b/init/main.c
@@ -556,6 +558,7 @@ static void __init mm_init(void)
        page_ext_init_flatmem();
+
+    memblock_free_all();
+
+    free_highmem_pages();
+
+    report_meminit();
-
-    mem_init();
-
-    kmem_cache_init();
-
-    pgtable_init();
@@ -566,4 +569,5 @@ static void __init mm_init(void)
        init_espfix_bsp();
-
-    pti_init();
+
+    arch_post_mm_init();
}
```