The slab allocators of past, present, and future

Monday, 12 September 2022 13:05 (25 minutes)

• A summary of how we got to have SLAB, SLOB and SLUB.
• The strengths and weaknesses of each - performance, debugging, memory overhead.
• The issues with having three implementations.
• Code complexity and bitrot
• Other features having to implement for each variant or limit choice (kmemcg, PREEMPT_RT...)
• Imperfect common code, recent attempts to unify it more
• API improvement issues - we would like kfree() to work on kmem_cache_alloc() objects, but SLOB would have to adapt and increase memory overhead.
• Can we drop SLOB and/or SLAB? What changes would SLUB need in order to replace their use cases?

I agree to abide by the anti-harassment policy

Yes

Primary author: BABKA, Vlastimil (SUSE Labs)
Presenter: BABKA, Vlastimil (SUSE Labs)
Session Classification: Kernel Memory Management MC
Track Classification: LPC Microconference: Kernel Memory Management MC