State of futex2

LCA 2022

André Almeida

Kernel Developer

andrealmeid@collabora.com

Open First

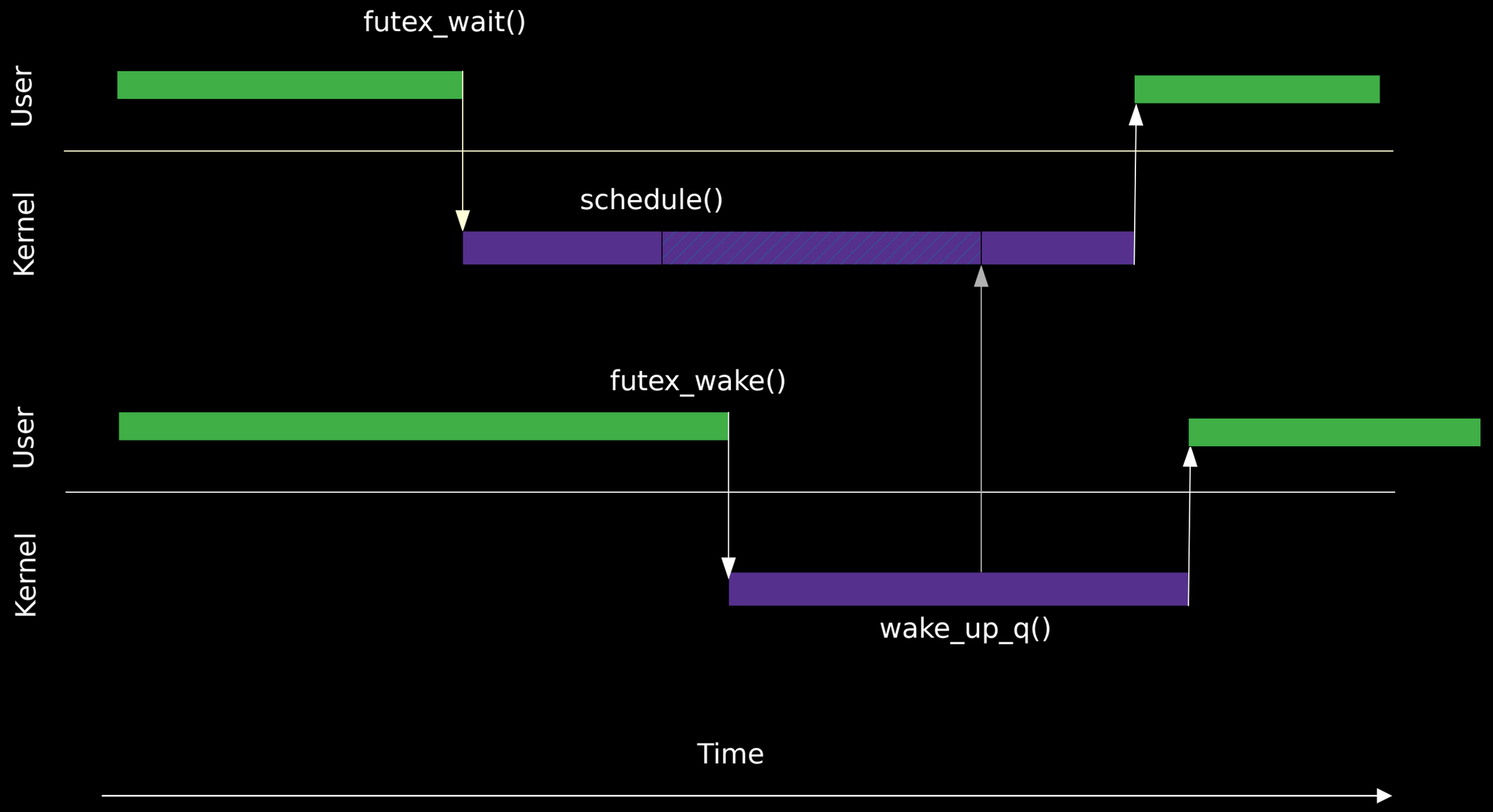
Overview

- What’s futex?
- Why do we need futex2?
- Current status
- Next steps

Fast Userspace Mutex

- System call for creating sync primitives in userspace (e. g. mutexes, semaphores)
- Kernel just provide ways to sleep/wake thread, all the logic is done in userspace

Fast Userspace Mutex



Why do we need a new futex API?

- Limitations: wait on a single futex, only 32bit sized futexes, no NUMA awareness
- Can’t add those features in the old multiplexed syscall:

```
futex(uint32_t *uaddr, int futex_op, uint32_t val,
      const struct timespec *timeout,   /* or: uint32_t val2 */
      uint32_t *uaddr2, uint32_t val3);
```

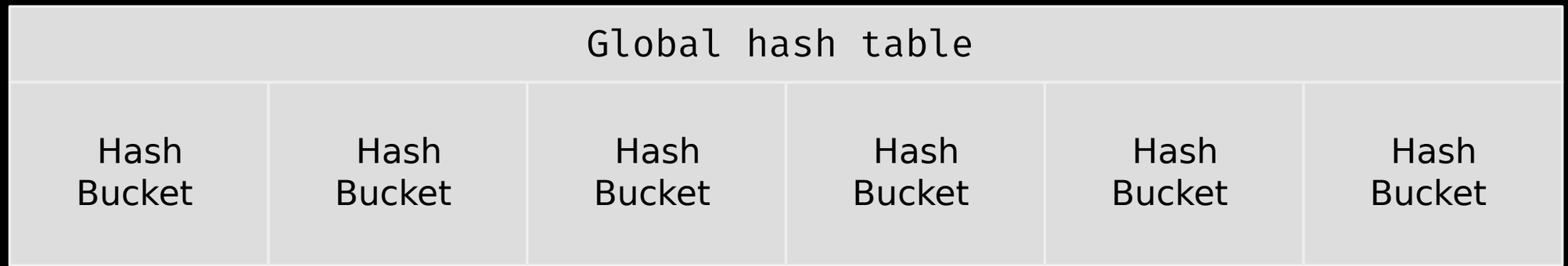
Wait on multiple futexes

- Wait on multiple futexes at the same time, wake on the first one that triggers a wake
- Similar operations can be find in other OS's, in particular WinAPI's `WaitForMultipleObjects()`
- Useful for game engine's loads, we used this in Steam's Proton
- `futex_waitv()` merged at 5.16!

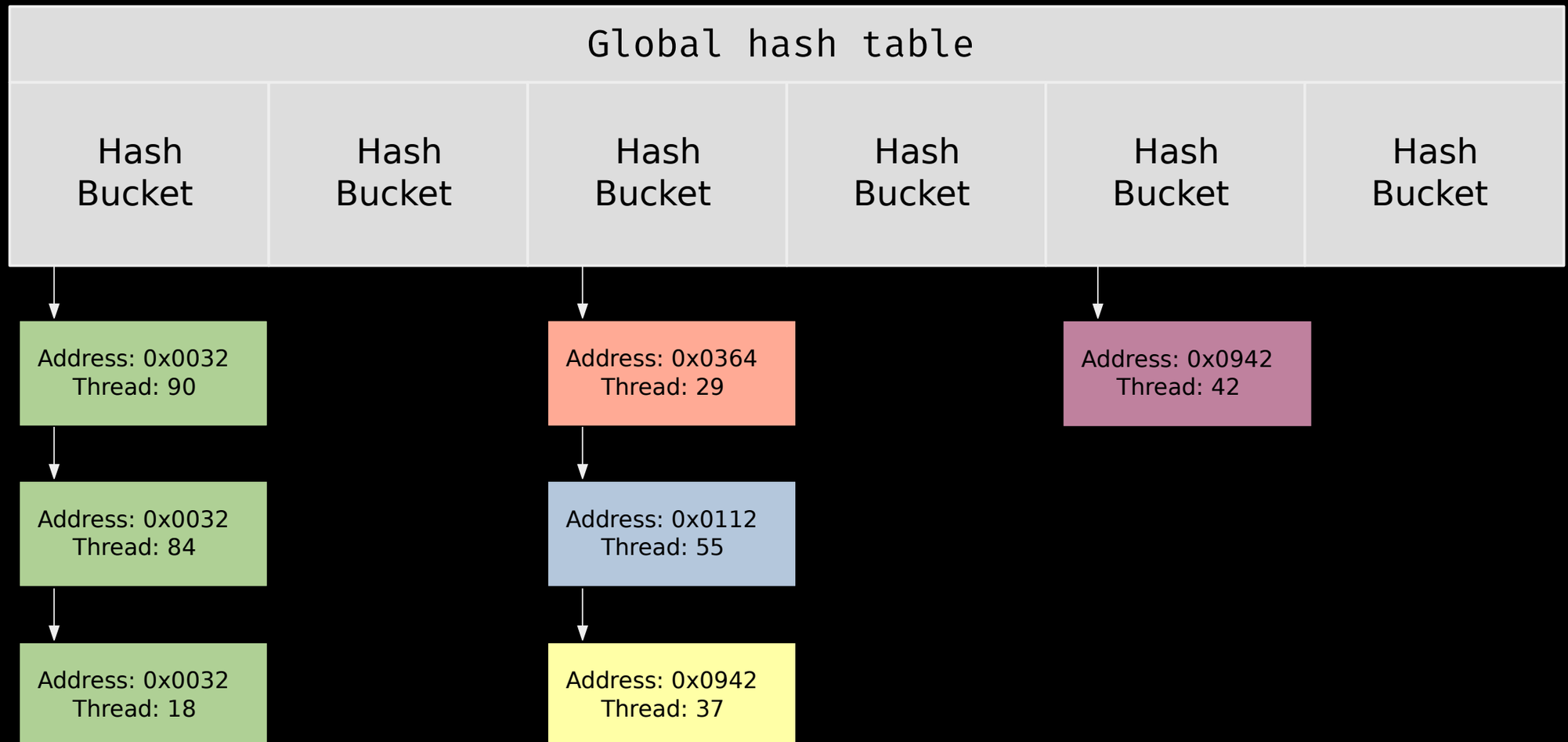
Variable sized futexes

- Current interface only supports 32-bit values
- Uses cases are related to atomic operations
- Userspace atomic primitives implementation
- 64-bit can be also useful to wait in a pointer value
- Sizes flags: `FUTEX_8`, `FUTEX_16`, `FUTEX_32`, `FUTEX_64`

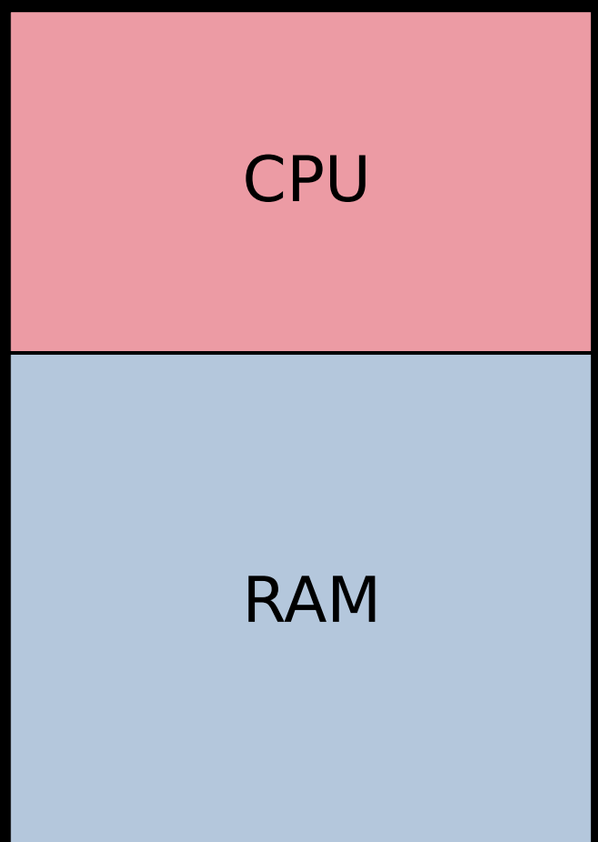
NUMA awareness



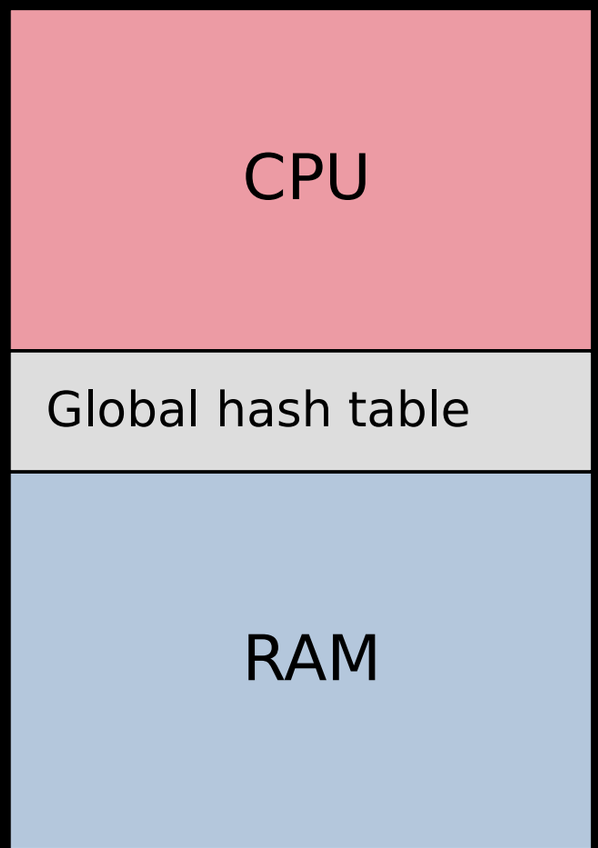
NUMA awareness



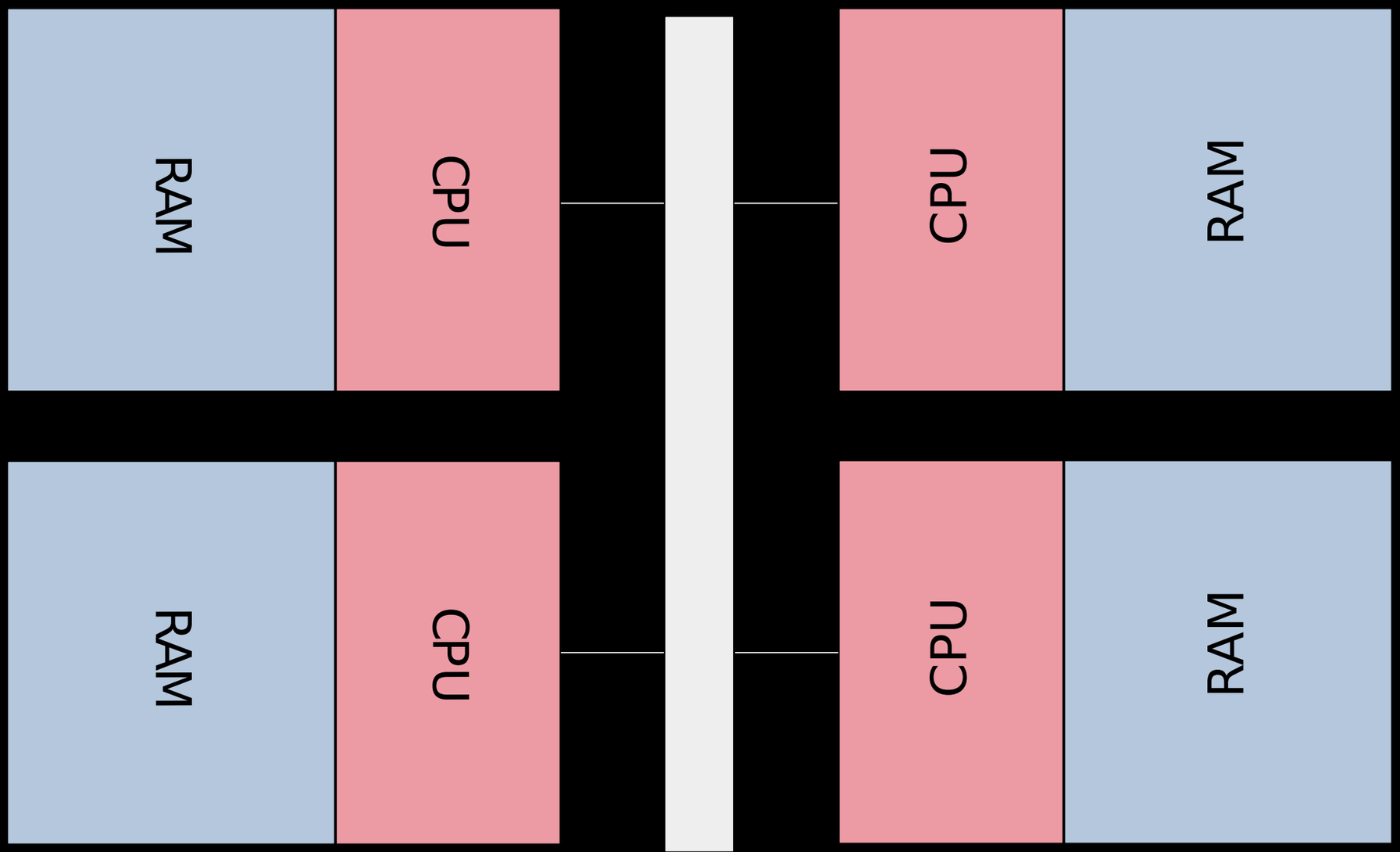
NUMA awareness



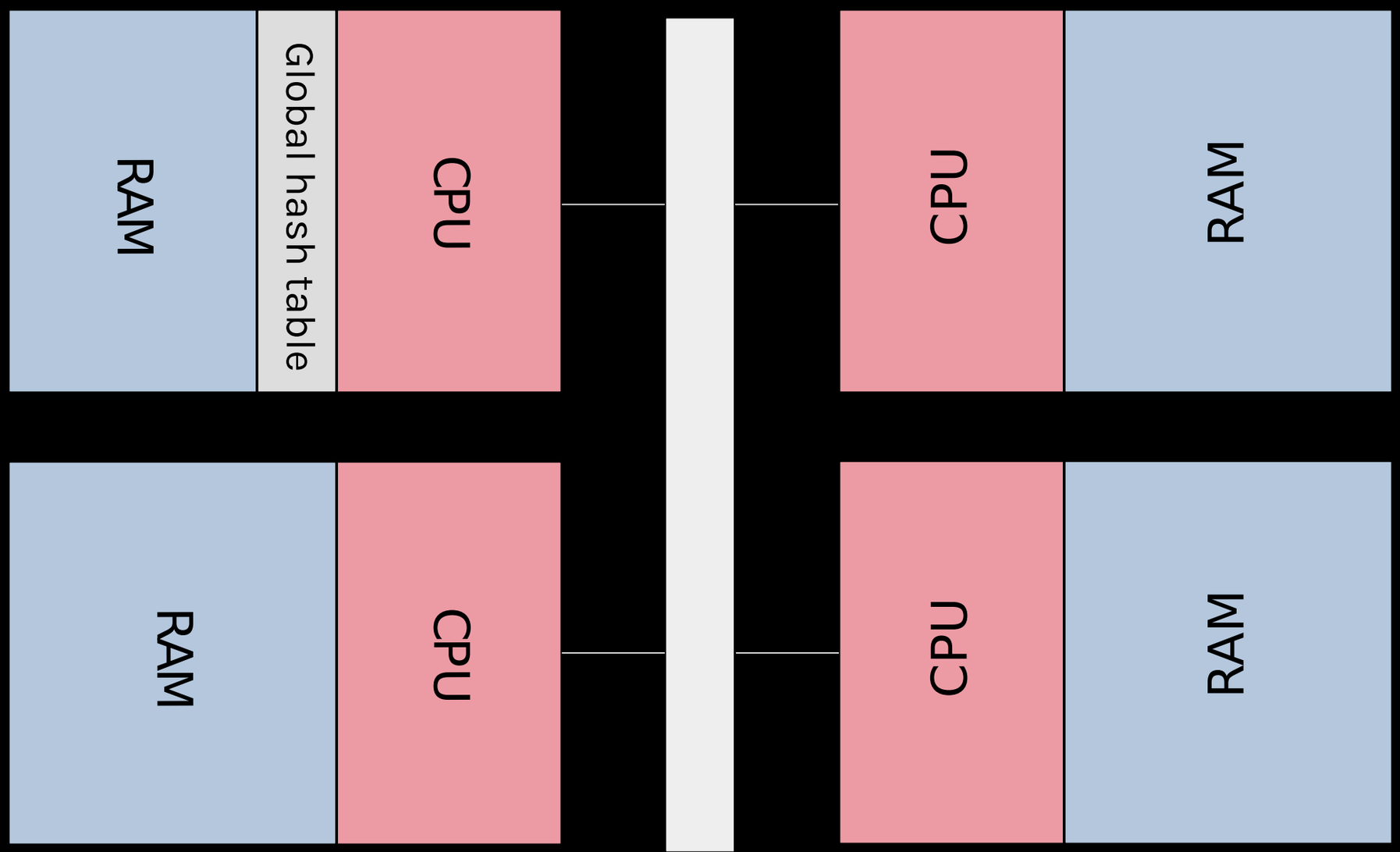
NUMA awareness



NUMA awareness



NUMA awareness



Next steps

- Figure out NUMA syntax

```
Flag: FUTEX_NUMA_FLAG

void *uaddr:
struct futex32_numa {
    __u32 value;
    __s32 hint;
};

value → expected value
hint → [0, MAX_NUMA_NODE] for NUMA to operate, -1 to current node
```

Thank you

```
Message {
  config {
    priority: "high"
    body: "Collabora is hiring"  // Many open positions
    recipient: "you"             // Please join us
    calltoaction: "http://col.la/join"
  }
}
```