A NUMA interface for futex2

André Almeida

LPC 2022
futex2

Ongoing effort to solve futex issues

futex_waitv() merged
For each FUTEX_WAIT operation, there's an entry in a kernelside hash table. And there's a single hash table, in one node. This creates a cost for the other nodes to access the table.
futex2 design

- A single syscall per operation (no multiplex)
- Designed to solve a bunch of issues at once (waitv, variable size, NUMA)
futex2 design

futex_wait(void *uaddr, unsigned int val, unsigned int flags, struct timespec *timo)

futex_wake(void *uaddr, unsigned long nr_wake, unsigned int flags)
futex2 design

```c
futex_waitv(struct futex_waitv *waiters,
    unsigned int nr_futexes,
    unsigned int flags,
    struct timespec *timeout,
    clockid_t clockid)

struct futex_waitv {
    __u64 val;
    __u64 uaddr;
    __u32 flags;
    __u32 __reserved;
};
```
futex2 design

```c
futex_requeue(struct futex_requeue *rq1,
               struct futex_requeue *rq2,
               unsigned int nr_wake,
               unsigned int nr_requeue,
               u64 cmpval, unsigned int flags)

struct futex_requeue {
    void *uaddr;
    unsigned int flags;
};
```
struct futexX_numa {
    __uX value;
    __sX hint;
};

struct futex32_numa f = {.value = 0, hint = -1};
FUTEX_NUMA_FLAG

```c
struct futexX_numa {
    __uX value;
    __sX hint;
};
```

- **value** is the futex value
- **hint** can be `[0, MAX_NUMA_NODES)` to specify a node or `-1` for the current node
```c
struct futexX_numa {
    __uX value;
    __sX hint;
};

struct futex32_numa f = {.value = 0, hint = -1};

futex_wait(&f, 0, FUTEX_NUMA | FUTEX_32, NULL);

// getting the lock
f.value = 1;
```
**FUTEX_NUMA_FLAG**

```c
struct futex32_numa f = {.value = 0, hint = 2};

futex_wait(&f, 0, FUTEX_NUMA, NULL);  // T1, N3
futex_wait(&f, 0, FUTEX_NUMA, NULL);  // T2, N0
futex_wait(&f, 0, FUTEX_NUMA, NULL);  // T3, N2
futex_wake(&f, 2, FUTEX_NUMA);        // T4, N1
```
Thanks!