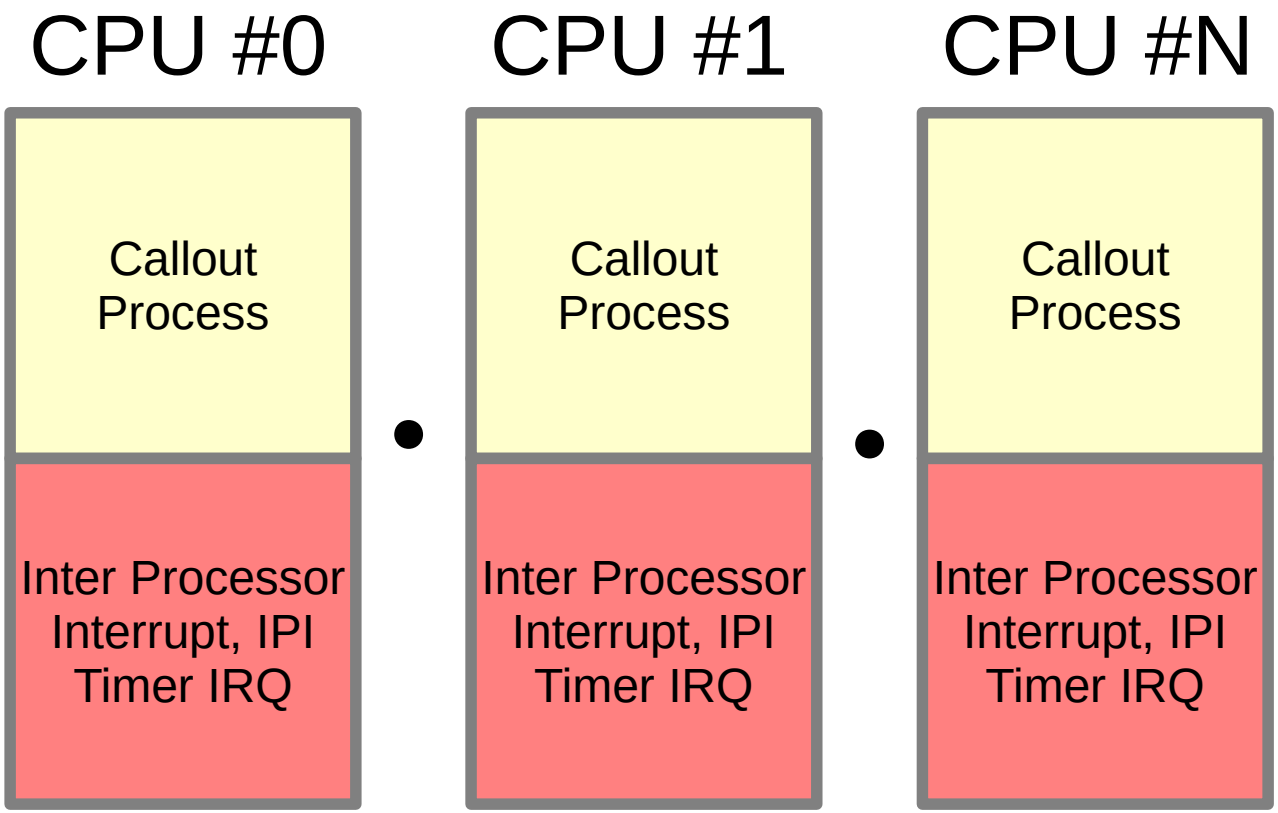
Callouts in FreeBSD

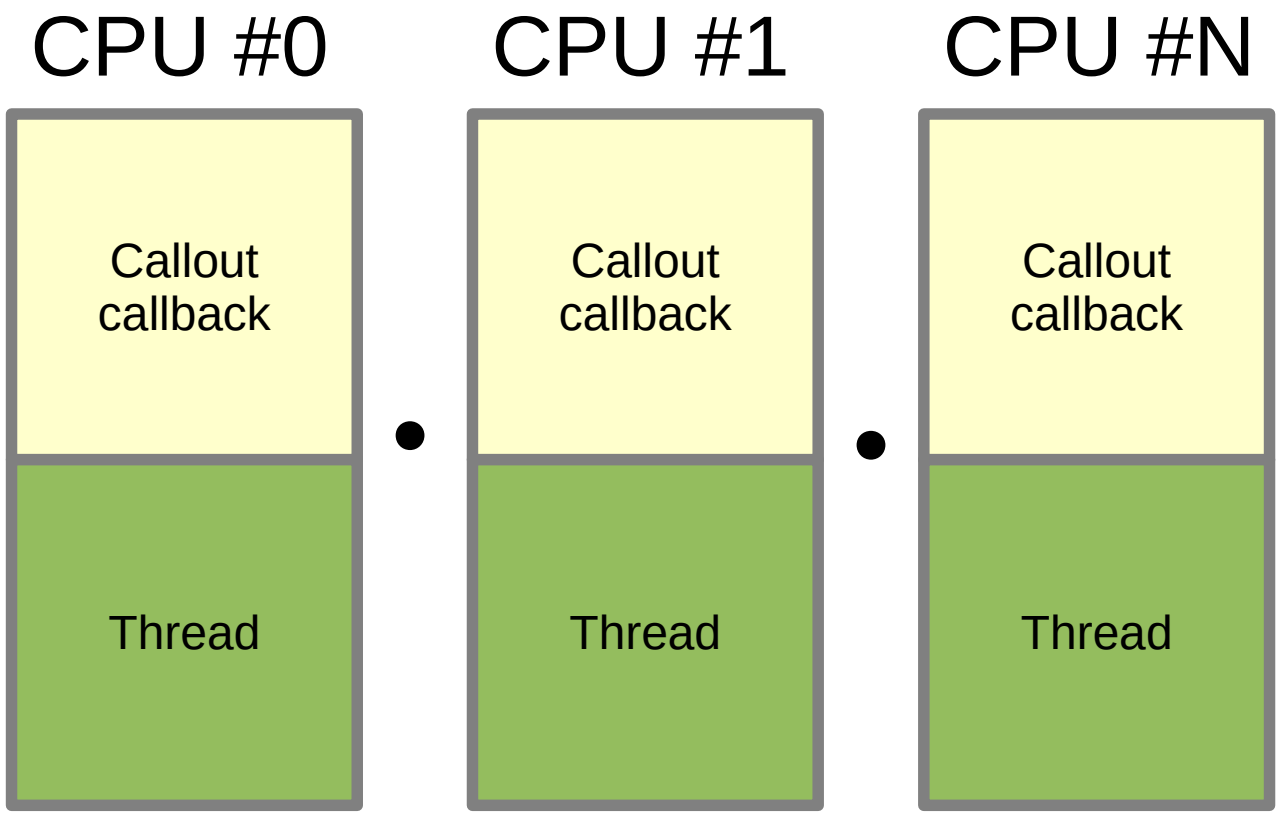
by hselasky @ freebsd .org

1) Introduction
2) Callouts at fast level
3) Callouts at slow level
4) Callouts and locking
5) Stopping callouts
6) Draining callouts w/o locks
7) Draining callouts w/ locks
8) Callout drain async
9) Interaction with «struct thread»
10) About projects/hps_head
11) Questions / Discussion

Callouts at fast level



Callouts at slow level



Callouts and locking

Lock type	Properties
None	FAST and SLOW context
SPIN *	FAST and SLOW context
MUTEX	SLOW context only
RW **	SLOW context only
RM **	SLOW context only

* new proposed feature
** shared locks can cause problems when stopping callouts

Stopping callouts

```
static void
softclock_call_cc()
{
        /* control code */

        SPIN UNLOCK()

        /* unlocked */

        XXX CO LOCK();

        /* callback code */

        XXX CO UNLOCK();

        /* unlocked */

        SPIN LOCK()
}
```

```
int
callout_stop()
{
        SPIN LOCK()
        /* control code */
        SPIN UNLOCK()

}
```

- callout_drain_async()
- callout_drain()

Draining callouts w/o locks

```
r = callout_reset(c);
 ...
 s = callout_stop(c);
 if (r && s) {
   /* Free resources used in callout callback */
   free_res();
 } else {
   /* Defer free resources used in callout callback */
   defer_free_res();
 }
```

Below code with mpsafe callout is right, and less tricky:

```
if (callout_drain_async(c, defer_free_res)) {

  /* Free resources used in callout callback */
  free_res();
}
```

Draining callouts w/ locks

Non-blocking alternative:

```
if (callout_drain_async(c, defer_free_res)) {

  /* Free resources used in callout callback */
  free_res();
}
```

Blocking alternative:

```
callout_drain(c) ;
free_res();
```

Callout drain async

- Makes tearing down lockless callouts less tricky (Julien Charbon)
- Allows fully non-blocking protocol implementations using callouts (HPS)

Interaction w/ «struct thread»

- cv_timedwait() invokes callout_reset()
- callout_drain() invokes msleep_spin()
- LOR between
  - callout spinlock
  - thread spin lock
- FreeBSD has own thread waiting state, TDF_TIMOFAIL

About projects/hps_head

- Removes the need for TDF_TIMOFAIL (uses spinlock to make callout stop atomic)
- Greatly simplifies callout_drain()
- Implements callout_drain_async()
- Adds support for SPIN locks as callback locks
-

Questions / Discussion

?