Documentation Ipanema

Baptiste Pires

April 18, 2024

1 Introduction

2 Structures

Dans cette section nous allons voir les différentes structures utilisées dans Ipanema et leurs utilités

2.1 `struct sched_ipanema_entity`

Nous allons commencer par la structure qui représente une entité d'ordonnancement. La structure est déclarée de cette façon :

```
struct sched_ipanema_entity {
  union {
    struct rb_node node_runqueue;
    struct list_head node_list;
  };

  /* used for load balancing in policies */
  struct list_head ipa_tasks;

  int just_yielded;
  int nopreempt;

  enum ipanema_state state;
  struct ipanema_rq *rq;

  /* Policy-specific metadata */
  void *policy_metadata;

  struct ipanema_policy *policy;
};
```

Les deux premiers champs, `struct rb_node node_runqueue` et `struct list_head node_list` sont utilisés pour maintenir une tâche dans une runqueue. Dû à l'union, on peut être dans une seule runqueue à un moment donner (soit dans une runqueue qui se sert d'un arbre rouge-noir comme CFS, ou une runqueue qui utilise une liste comme un algorithme FIFO par exemple).

Le champ `struct list_head ipa_tasks` est utilisé par les différentes politiques d'ordonnancement pendant l'équilibrage de charge. Par exemple, si on migre des tâches d'un coeur C_1 à un coeur C_2, on a besoin de retirer les tâches de la runqueue de C_1 en ayant le verrou sur celle-ci. Quand on retire les tâches on peut les ajouter à la liste `ipa_tasks`. Une fois qu'on a retirée toutes les tâches qu'on souhaite migrer, on prend le verrour sur la runqueue de C_2 et on peut itérer sur la liste que l'on vient de construire.

`int just_yield` permet de savoir si on vient d'appeler `sched_yield()` ou non.

`int nopreempt`
`enum ipanema_state state` représente l'état d'une entitté Ipanema . Les états possibles sont les suivants :

```
enum ipanema_state {
  IPANEMA_NOT_QUEUED, /* La tache n'est pas dans une runqueue */
  IPANEMA_MIGRATING, /* La tache est en train de migrer d'un coeur a un autre */
  IPANEMA_RUNNING, /* La tache est dans une runqueue et s'execute */
  IPANEMA_READY, /* La tache est dans une runqueue mais ne s'execute pas */
  IPANEMA_BLOCKED, /* La tache est bloque (I/O, lock, etc...) */
  IPANEMA_TERMINATED, /* La tache meurt (?) */
  /*
   * A special state that is equivalent to IPANEMA_READY, but makes it
   * possible to figure out that the state change came from tick().
   */
  IPANEMA_READY_TICK
};
```

`struct ipanema_rq *rq` c'est un pointeur vers la `struct ipanema_rq` sur laquelle la tâche s'exécute (ou va s'exécuter).

3 Ecrire un scheduler dans Ipanema

Dans cette section nous allons voir comment écrire un scheduler sous la forme d'un module en utilisant Ipanema .