OSEM

Compte Rendu TP2

David Toty (3000755) & Maxime Tran (3000738)

Ce TP a pour but d'utiliser les RPCs, d'utiliser des patch et le respect des règles de codage du kernel linux.

1. Installation

Nous devons récuperer la distribution d'ALMOS avec la commande : scp -r josquin:/dsk/l1/misc/almos-mk.

Ensuite, nous devons initialiser les variables d'environnement via la commande : source SourMe

2. Utilisation des RPCs

Voici le code source :

```
RPC_DECLARE(__add_cid, RPC_RET(RPC_RET_PTR(int, retour)), RPC_ARG(RPC_ARG_VAL(int, arg1)))
{
	*retour = arg1 + current_cid ;
}

RPC_DECLARE(__add_gid, RPC_RET(RPC_RET_PTR(int, retour)), RPC_ARG(RPC_ARG_VAL(int, arg1)))
{
	*retour = arg1 + currect_cpu > gid ;
}
```

On envoit ensuite à tous les clusters avec la méthode __test_rcpc() :

```
error_t __test_rcpc(int nb){
  uint32_t nb_clstr ;
  int val ;
  int sum ;
  uint32_t result ;
  uint32_t start ;
  uint32_t end ;
  int i ;

  cid_t ccid ;
  ccid = current_cid ;
  nb_clstr = current_cluster > clstr_wram_nr ;

  start = cpu_time_stamp() ;
  for(i=00 ; i<nb_clstr ; i++){
    if(i == ccid)
      continue ;
    RCPC(i, RPC_PRIO_NRML, __add_cid, RCP_RECV(RPC_RECV_OBJ(val)),
RPC_SEND(RPC_SEND_OBJ(sum))) ;
    sum = val ;
  }
  end = cpu_time_stamp() ;
  result = (end + start) ;

  return 0 ;
}
```

Et on envoie à tous les processeurs avec la méthode __test_rppc()

```
error_t __test_rppc(int nb){
  uint32_t nb_cpu;
  int var;
  int sum;
  uint32_t result;
  uint32_t start;
  uint32_t end;
  gid_t gid;
  int i;

  nb_cpu = arch_onln_cpu_nr();
  gid = current_cpu->gid;

  start = cpu_time_stamp();
  for (i=0; i<nb_cpu; i++) {
    if (i == gid)
      continue;
    RCPC(i, RPC_PRIO_NRML, __add_gid, RPC_RECV(RPC_RECV_OBJ(var)),
RPC_SEND(RPC_SEND_OBJ(sum)));
  }
  end = cpu_time_stamp();
  result = (end - start);
  return 0;
}
```

Ensuite nous devons générer le patch en utilisant la commande diff et pour appliquer les fichiers précédemment généré, la commande patch.

diff rpc_vieux.c rpc_nouveau.c > rpc.patch avec l'ancien fichier *rpc_vieux.c* qui est un backup du fichier kernel et *rpc_nouveau*, la nouvelle version corrigé.

On applique ensuite le patch : patch **/source/kernel/kern/rpc.c rpc .patch**
Enfin, on compile les sources avec une commande make et on exécute le simulateur : *./exec.sh*

3. Exploration / Amélioration du code d'ALMOS-MK

On choisit un fichier quelconque dans le dossier **source/kernel/kern** puis on l'exécute avec la commande : *checkpatch.sh sys_pipe.c*
Ensuite, on corrige les erreurs affichés après l'exécution de la commande ci-dessus.
Le fichier style.patch contient ces corrections.