Université Pierre et Marie Curie

Rapport Archi 4

TP4 MJPEG

Auteurs :
BITAM Massine
TOUMLILT Ilyas

Encadrant :
MEUNIER quentin

4 janvier 2016

Question 1

Il serait plus efficace que la tâche split ne soit pas “gênée” par une attente sur le canal split->demux, et donc il serait plus intéressant d’augmenter la profondeur de ce dernier pour optimiser le parallèlisme, la profondeur la plus efficace aurait donc une valeur équivalente à celle de la taille de son banc mémoire, c’est à dire 32. D’ailleurs en testant, on confirme bien qu’une profondeur de 32 est 52% plus efficace qu’une profondeur de 2.
Pour les fifos idct0->libu et idct1->libu,on considère que libu s’exécute plus rapidement que la moitié du temps d’exécution d’un pipeline. Il n’est donc pas nécessaire de changer la profondeur des canaux idct->libu.

Question 2

Configuration 1 :

```
mapper.map( "tg_split", buffer = "cram0_0", status = "cram0_0")
mapper.map( "libu_ramdac", buffer = "cram3_0", status = "cram3_0")
#PIPE 1
mapper.map("split_demux1", buffer = "cram0_0", status = "cram0_0")
mapper.map( "demux_vld1", buffer = "cram1_0", status = "cram1_0")
mapper.map( "vld_iqzz1", buffer = "cram1_0", status = "cram1_0")
mapper.map( "iqzz_idct1", buffer = "cram3_0", status = "cram3_0")
mapper.map( "idct_libu1", buffer = "cram1_0", status = "cram1_0")
mapper.map( "huffman1", buffer = "cram1_0", status = "cram1_0")
mapper.map( "quanti1", buffer = "cram1_0", status = "cram1_0")
#PIPE 2
mapper.map("split_demux2", buffer = "cram0_0", status = "cram0_0")
mapper.map( "demux_vld2", buffer = "cram2_0", status = "cram2_0")
mapper.map( "vld_iqzz2", buffer = "cram2_0", status = "cram2_0")
mapper.map( "iqzz_idct2", buffer = "cram3_0", status = "cram3_0")
mapper.map( "idct_libu2", buffer = "cram2_0", status = "cram2_0")
mapper.map( "huffman2", buffer = "cram2_0", status = "cram2_0")
mapper.map( "quanti2", buffer = "cram3_0", status = "cram3_0")
# mapping the "prod0" and "cons0" tasks
mapper.map("split", run = "cpu0_0", stack = "cram0_0", desc = "cram0_0")
#1
mapper.map("demux1", run = "cpu1_0", stack = "cram0_0", desc = "cram0_0")
mapper.map("vld1", run = "cpu1_1", stack = "cram1_0", desc = "cram1_0")
mapper.map("iqzz1", run = "cpu1_0", stack = "cram1_0", desc = "cram1_0")
mapper.map("idct1", run = "cpu1_1", stack = "cram1_0", desc = "cram1_0")
#2
mapper.map("demux2", run = "cpu2_0", stack = "cram0_0", desc = "cram0_0")
mapper.map("vld2", run = "cpu2_1", stack = "cram2_0", desc = "cram2_0")
mapper.map("iqzz2", run = "cpu2_0", stack = "cram2_0", desc = "cram2_0")
mapper.map("idct2", run = "cpu2_1", stack = "cram2_0", desc = "cram2_0")

mapper.map("libu", run = "cpu3_0", stack = "cram3_0", desc = "cram3_0")
```

Configuration 2 :

```
mapper.map( "tg_split", buffer = "cram0_0", status = "cram0_0")
mapper.map( "libu_ramdac", buffer = "cram3_0", status = "cram3_0")
#PIPE 1
mapper.map("split_demux1", buffer = "cram0_0", status = "cram0_0")
mapper.map( "demux_vld1", buffer = "cram1_0", status = "cram1_0")
mapper.map( "vld_iqzz1", buffer = "cram1_0", status = "cram1_0")
mapper.map( "iqzz_idct1", buffer = "cram2_0", status = "cram2_0")
mapper.map( "idct_libu1", buffer = "cram3_0", status = "cram3_0")
mapper.map( "huffman1", buffer = "cram0_0", status = "cram0_0")
mapper.map( "quanti1", buffer = "cram1_0", status = "cram1_0")
#PIPE 2
mapper.map("split_demux2", buffer = "cram0_0", status = "cram0_0")
mapper.map( "demux_vld2", buffer = "cram1_0", status = "cram1_0")
mapper.map( "vld_iqzz2", buffer = "cram1_0", status = "cram1_0")
mapper.map( "iqzz_idct2", buffer = "cram2_0", status = "cram2_0")
mapper.map( "idct_libu2", buffer = "cram3_0", status = "cram3_0")
mapper.map( "huffman2", buffer = "cram0_0", status = "cram0_0")
mapper.map( "quanti2", buffer = "cram1_0", status = "cram1_0")
# mapping the "prod0" and "cons0" tasks
mapper.map("split", run = "cpu0_0", stack = "cram0_0", desc = "cram0_0")
#1
mapper.map("demux1", run = "cpu0_0", stack = "cram0_0", desc = "cram0_0")
mapper.map("vld1", run = "cpu1_1", stack = "cram1_0", desc = "cram1_0")
mapper.map("iqzz1", run = "cpu1_0", stack = "cram1_0", desc = "cram1_0")
mapper.map("idct1", run = "cpu2_0", stack = "cram1_0", desc = "cram1_0")
#2
mapper.map("demux2", run = "cpu1_0", stack = "cram0_0", desc = "cram0_0")
mapper.map("vld2", run = "cpu2_1", stack = "cram2_0", desc = "cram2_0")
mapper.map("iqzz2", run = "cpu2_0", stack = "cram2_0", desc = "cram2_0")
mapper.map("idct2", run = "cpu3_0", stack = "cram2_0", desc = "cram2_0")

mapper.map("libu", run = "cpu3_0", stack = "cram3_0", desc = "cram3_0")
```

Nombres de cycles obtenus :
Config1 :260000000
Config2 :285000000

Question 3

Les performances sont meilleures quand les tâches communiquantes et leurs canaux de communication sont sur un même cluster.
De plus, il est préférable dans un même cluster, d'éviter de mettre deux taches qui se suivent séquentiellement sur un même CPU.