實驗一

Deadline: 10/9 11:59pm

LAB1-1 Multicore architecture and multithreading

實驗目的

1. 使用 multithreading programming(Pthread)來加速程式效能，並 set affinity 來控制 thread 在不同 core 執行，並分析所帶來的差異。

實驗環境

Ubuntu 14.04 LTS(我們實驗環境，亦可使用別的版本 linux)

實驗要求

1. 程式部份(30%)
2. 實驗報告(10%)

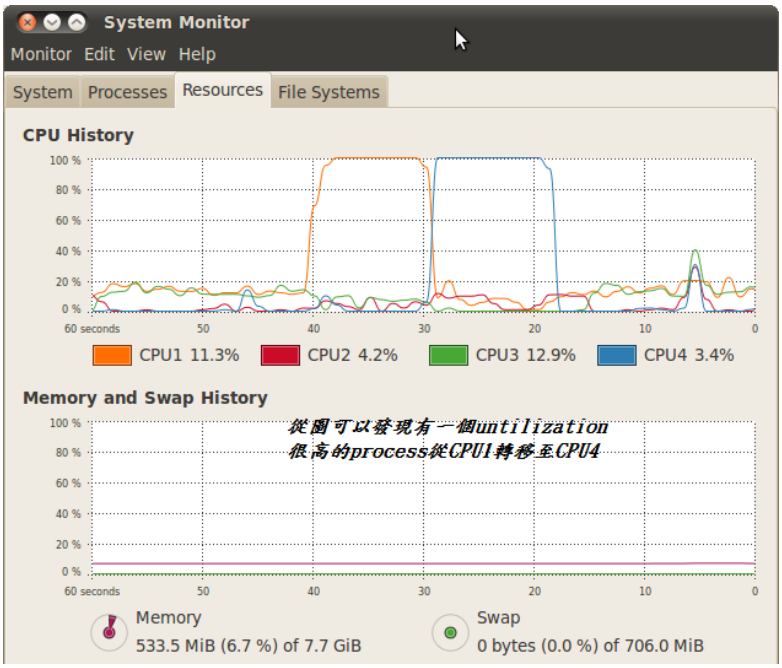
實驗步驟

1. Task1:
   利用 pthread_setaffinity_np 來分析 CPU 資源對於應用程式 pthread_setaffinity_np_test 所產生的影響。
2. 計算 pthread_setaffinity_np_test 執行時間來完成下表

項目	Time
a. 執行原始程式 pthread_setaffinity_np_test.c	
b. 修改原始程式以 create 一個 thread 並使用 pthread_setaffinity_np，讓 thread 在執行過程中切換 core 執行。<pro_1.c> <10%>	
c. 修改原始程式以 create 兩個 thread 並使用 pthread_setaffinity_np，讓兩個 thread 分別在不同兩個 core 上執行。< pro_2.c > <10%>	
d. 修改原始程式以 create 兩個 thread 並使用 pthread_setaffinity_np，讓兩個 thread 強迫在同個 core 上執行。< pro_3.c > <10%>	

3. Task1 (c,d) 兩個 threads 必須執行同份工作，意思是你必須把 waste_time function 分割給不同 thread。
4. 項目 b 需截圖附加在報告中(需要提供實驗環境, e.g. core 數,memory 大小,cache 大小)<10%>。

如：



程式架構說明

1. **Lab1-1/**
   Makefile : Makefile(make/make clean/make run)
   pthread_setaffinity_np_test.c : pthread_setaffinity_np 測試程式碼

實驗一

Deadline: 10/9 11:59pm

LAB1-2 Performance Profiling

實驗目的

1. 藉由 workload 的變化觀察系統效能。

實驗環境

1. Ubuntu 14.04 LTS(我們實驗環境，亦可使用別的版本 linux)
2. 請在系統上面安裝 iotop (詳見 Appendix A)

實驗要求

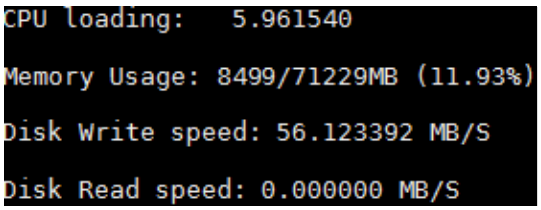
1. Workload generator (30%)
2. Monitor(30%)

實驗步驟

1. Task1 Workload:
   i. 寫一個程式讓 CPU、Memory、I/O 負載且能夠在 command line 中指定負載量。
   ii. 注意這邊指的負載量是**該程式能在系統產生**負載量。(如目前系統 cpu 使用 50%，workload 產生 10% workload，則整個系統 cpu 變為使用 60%)
   iii. 若要得知 I/O 的負載成數，需要先測試硬碟 I/O 的速度上限 (詳見 Appendix B)
   iv. 程式不一定要有輸出，但是必須能在 top/htop 中看出負載狀況
   v. 參考: top、iotop、proc 的 man page
   vi. 例 :

```
./workload cpu 10 mem 20 diskio 100
```

2. Task2 Monitor:
   寫一個能夠看到即時 CPU、Memory、I/O 狀態的程式。需要能夠顯示各項當下的狀況(必須使用/proc 底下或是系統直接提供的資料，不能直接拿別人寫好程式(如 htop,iotop)之資料)



程式架構說明

1. workload/
   Makefile(make/make clean/make run)
   workload.c(cpp)
   Usage: ./workload [cpu] [mem] [io]
2. monitor/
   Makefile : Makefile(make/make clean/make run)
   monitor.c(cpp) : Task2 的 code

Appendix A

安裝 iotop: 在 terminal 中輸入 sudo apt install iotop 即可

Appendix B

1. 可以使用 dd 來製造 disk 的最大寫入速度，並使用 iotop 觀察
   # dd if=/dev/zero of=/tmp/output conv=fdatasync; rm –f /tmp/output

上傳檔案

Lab1-1/

- pthread_setaffinity_np_test.c (cpp)
- pro_1.c
- pro_2.c
- pro_3.c
- Makefile

Lab1-2/

- workload/
  - workload.c (cpp)
  - Makefile
- monitor/
  - monitor.c (cpp)
  - Makefile

程式碼請依上述架構排好，壓縮成一個壓縮檔，檔名為 LAB1_學號_姓名.rar。

參考內容：

https://computing.llnl.gov/tutorials/pthreads/
http://www.kernel.org/doc/man-pages/online/pages/man3/pthread_setaffinity_np.3.html