

TEST N.1

Nel test N.1 profiliamo un'applicazione fissa in condizioni di transizione tra due app e a regime, variando il numero di core assegnati ai disturbi e all'app da profilare.

Nello specifico, immaginiamo il dispositivo Android avente due display:

l'app da profilare rimarrà fissa sul display 2(virtuale), mentre sul display 1 si alternano varie app, I "disturbi".

Con il programma simpleperf (di Android) andremo a profilare l'applicazione principale nelle seguenti condizioni:

- Transitorio (switch tra una due app secondarie, disturbi)
- Regime (disturbo in foreground sul display 1 da almeno 1 secondo)

Ogni istanza di simpleperf, sia in transitorio che in regime, ha durata di 1 secondo.

Ricordo che in Android, le app possono trovarsi in 3 stati principali:

- Foreground (visibile sullo schermo, no interazione con l'utente (niente tocco sul display))
- Background (non visibile all'utente)
- Top-App (interazione con l'utente)

In base allo stato in cui si trova, all'app è assegnato un CPU Set omonimo, per cui vengono assegnati rispettivamente

- 6 core (little+Medium) per Foreground
- 4 core (little) per Background
- 8 core (tutti) per Top-App

Faremo vari esperimenti in cui andremo ad alterare I Core assegnati ai disturbi e all'app da profilare. Per avere più controllo sull'app da profilare, data la natura dinamica degli stati delle app in base all'interazione, ho creato un nuovo CPU Set, in cui l'app da profilare rimane per tutta la durata dell'esperimento.

Un CPU Set è una struttura dati del kernel. P.es per creare un nuovo Set di nome "Custom", basta creare una cartella in /dev/cpuset/ di nome Custom. La cartella verrà popolata con tutti I file necessari, tra cui il file "cpus" che determina I core assegnati ad un processo di tipo "Custom"

Con ' echo "4-7" > /dev/cpuset/Custom/cpus ' assegno I core 4-7 alle app di tipo Custom.

Con ' echo PID > /dev/cpuset/Custom/tasks ' invece assegno un task di pid PID al cpuset Custom.

Questa è la tecnica usata per assegnare in modo stabile all'app da profilare determinati core.

Per assegnare I core ai disturbi durante l'esperimento, basta cambiare le cpu assegnate ai Sets Top-app e Foreground.

P.es se voglio che I disturbi abbiano solo I primi 6 core, scrivo "0-5" nel file "cpus" del set Top-app e Foreground, così che, in ogni caso, gli vengano assegnati solo 6 core, sia che ci sia interazione con il tocco o no. (' echo "0-5" > /dev/cpuset/top-app/cpus ')

Faccio inoltre un underclock per avere risultati affidabili, con clock fisso nel tempo e dispositivo esente da thermal throttling. L'efficacia dell'underclock è stata verificata dalle tracce registrate con Perfetto.

Underclock:

-freq max little 1197MHz

-max medium 1836 MHz

-max big 1826 MHz

Inoltre nel seguente test, le app in background, i "disturbi", possono avere il seguente numero di Core

- 8

- 4 (Medium + BIG core)

- 2 (BIG)

- 4 (little)

L'App principale da profilare sarà Minecraft (MC).

8 CORE A MC E 6 CORE (litte + Medium) ai Disturbi

Media dati di 21 esecuzioni nel Transitorio

Cache Miss = 12097080

Instructions = 1095673412

CPU Cycles = 1737751157

Istruzioni per ciclo: 0.6305122615435554

BP Miss = 5824525

dTLB Walks = 3446941

iTLB Walks = 958701

Media dati di 21 esecuzioni a Regime

Cache Miss = 12093133

Instructions = 1102243846

CPU Cycles = 1813592692

Istruzioni per ciclo: 0.6077681338605658

BP Miss = 5841520

dTLB Walks = 3821997

iTLB Walks = 1085514

8 CORE A MC E 4 CORE (Medium + BIG) ai Disturbi

Media dati di 21 esecuzioni nel Transitorio

Cache Miss = 11621773

Instructions = 843937240

CPU Cycles = 1927372182

Istruzioni per ciclo: 0.43786936839788837

BP Miss = 6404868

dTLB Walks = 2875303

iTLB Walks = 966963

Media dati di 21 esecuzioni a Regime
Cache Miss = 12312899
Instructions = 1094601469
CPU Cycles = 2022756981
Istruzioni per ciclo: 0.5411433401450216

BP Miss = 6340615
dTLB Walks = 3593321
iTLB Walks = 1041016

8 CORE MC, 2 CORE (BIG) ai Disturbi

Media dati di 21 esecuzioni nel Transitorio
Cache Miss = 13366729
Instructions = 1110855394
CPU Cycles = 2201837614
Istruzioni per ciclo: 0.5045128609561486

BP Miss = 6584369
dTLB Walks = 4069180
iTLB Walks = 1458827

Media dati di 21 esecuzioni a Regime
Cache Miss = 13619468
Instructions = 11452919422
CPU Cycles = 2137971341
Istruzioni per ciclo: 0.5356909702373789

BP Miss = 6594781
dTLB Walks = 3298328
iTLB Walks = 1075237

8 CORE MC, 4 CORE (little) ai Disturbi

IRREALISTICO, I DISTURBI sono veramente POCO PERFORMANTI

Si nota in questo caso che in regime l'app ha un IPC minore rispetto al transitorio, perché facendo swtich tra 2 Disturbi, i core little vengono occupati dall'app in avvio e quindi MC va sui core BIG e Medium che ha a disposizione. (In regime invece MC si “”spalma”” sui core little meno performanti)

Questo ovviamente succede solo quando i disturbi hanno assegnati solo i core little, mentre MC ha i core più performanti. con le CPU scariche MC si distribuisce su tutti gli 8 core, mentre se i little core sono occupati da disturbi in avvio, MC va solo sui Medium e BIG

Media dati di 21 esecuzioni nel Transitorio
Cache Miss = 11889481
Instructions = 1042033597

CPU Cycles = 1599752265
Istruzioni per ciclo: 0.6513718528787331

BP Miss = 5684443
dTLB Walks = 3780750
iTLB Walks = 1199767

Media dati di 21 esecuzioni a Regime
Cache Miss = 12838224
Instructions = 1120283872
CPU Cycles = 1825160334
Istruzioni per ciclo: 0.6138002514797146

BP Miss = 6036021
dTLB Walks = 4047769
iTLB Walks = 1211779

8 CORE MC 8 CORE Disturbi

Media dati di 21 esecuzioni nel Transitorio
Cache Miss = 11967453
Instructions = 922426191
CPU Cycles = 1901364762
Istruzioni per ciclo: 0.48513899564948393

BP Miss = 6559637
dTLB Walks = 3620424
iTLB Walks = 1179062

Media dati di 21 esecuzioni a Regime
Cache Miss = 12738853
Instructions = 1092364195
CPU Cycles = 1950434523
Istruzioni per ciclo: 0.5600619667661615

BP Miss = 6232228
dTLB Walks = 3921615
iTLB Walks = 1111333

###NUOVO TEST, assegno a Minecraft solo 6 core (Little+Medium)

6 CORE MC 8 CORE Disturbi

Media dati di 21 esecuzioni nel Transitorio
Cache Miss = 11977943
Instructions = 907541163
CPU Cycles = 2051922009

Istruzioni per ciclo: 0.44228833211954693

BP Miss = 6490363
dTLB Walks = 3653073
iTLB Walks = 1186491

Media dati di 21 esecuzioni a Regime
Cache Miss = 12619976
Instructions = 1094730450
CPU Cycles = 2159587745
Istruzioni per ciclo: 0.5069164022321306

BP Miss = 6776806
dTLB Walks = 4329539
iTLB Walks = 1326240

6 CORE MC, 4 CORE (Medium BIG) disturbi

Media dati di 21 esecuzioni nel Transitorio
Cache Miss = 11135456
Instructions = 717072438
CPU Cycles = 1960387052
Istruzioni per ciclo: 0.3657810518940318

BP Miss = 6363446
dTLB Walks = 2729106
iTLB Walks = 948908

Media dati di 21 esecuzioni a Regime
Cache Miss = 12730341
Instructions = 1099683719
CPU Cycles = 2266309626
Istruzioni per ciclo: 0.48523101450216405

BP Miss = 7040261
dTLB Walks = 3941421
iTLB Walks = 1222337

Le prestazioni sono molto basse perché durante lo switch, i core 4-7 vengono prelazionati e dati ai disturbi. MC allora è costretto a stare solo sui 4 core little (nel transitorio)

6 CORE MC, 2 CORE DISTURBi (BIG)

Media dati di 21 esecuzioni nel Transitorio
Cache Miss = 13016254
Instructions = 1184601974

CPU Cycles = 2218645509
Istruzioni per ciclo: 0.5339302602396947

BP Miss = 6681076
dTLB Walks = 4114342
iTLB Walks = 1341313

Media dati di 21 esecuzioni a Regime
Cache Miss = 13127599
Instructions = 1204525801
CPU Cycles = 2262178153
Istruzioni per ciclo: 0.5324628387037561

BP Miss = 7014559
dTLB Walks = 4053932
iTLB Walks = 1308209

prestazioni equilibrate, non c'è "lotta" per le risorse: Disturbi e MC
sono su core differenti e dunque avrò simili prestazioni (in termini di IPC) tra
regime e transitorio.

6 Core MC DISTURBI 4 Core(Little)

Media dati di 21 esecuzioni nel Transitorio
Cache Miss = 11605756
Instructions = 1027484272
CPU Cycles = 1900398213
Istruzioni per ciclo: 0.5406678794851087

BP Miss = 5910603
dTLB Walks = 4089805
iTLB Walks = 1263795

Media dati di 21 esecuzioni a Regime
Cache Miss = 12790706
Instructions = 1116437955
CPU Cycles = 2168570600
Istruzioni per ciclo: 0.5148266581682883

BP Miss = 6541650
dTLB Walks = 4344551
iTLB Walks = 1337225

MC è praticamente sempre sui core Medium.
sui core BIG "liberi" da MC e Disturbi, ci sono task del kernel e sonde di Perfetto per la
profilazione e registrazione delle tracce.

Come riferimento, seguiranno I risultati della profilazione dell'app principale senza la presenza di disturbi.

NO DISTURBI, SOLO MC CON 8 CPU

```
#      count event_name  # count / runtime
392,006,538 cache-misses # 10.524 M/sec
35,784,614,089 instructions # 961.013 M/sec
56,257,077,841 cpu-cycles  # 1.511283 GHz
ISTRUZIONI PER CICLO 0.63589
```

```
185,551,272 branch-misses # 5.824 M/sec
121,861,422 raw-dtlb-walk # 3.825 M/sec
37,244,349 raw-itlb-walk # 1.169 M/sec
```

NO DISTURBI, SOLO MC CON 6 CPU

```
#      count event_name  # count / runtime
419,573,076 cache-misses # 11.070 M/sec
36,651,652,347 instructions # 967.087 M/sec
73,208,642,174 cpu-cycles  # 1.931804 GHz
ISTRUZIONI PER CICLO 0.50061
```

```
208,785,016 branch-misses # 5.128 M/sec
145,974,998 raw-dtlb-walk # 3.586 M/sec
45,364,353 raw-itlb-walk # 1.115 M/sec
```

TEST N.2

Nel secondo test avremo due applicazioni, APP1 e APP2, che si susseguono una dopo l'altra tra Foreground e Background (Ho un solo schermo). Misuro nei transitori i cali di prestazioni dell'intero Sistema, e non della singola app. (Profilo dunque l'intero sistema).

Il test durerà 30 secondi, nei quali ci saranno 20 swipe tra le 2 applicazioni.
Le due app profilate sono il Gestore di File e il Dialer

Anche in questo test verrà usato il precedente Underclock.
Inoltre le app avranno un comportamento "di serie", dunque durante lo swipe saranno Top-App (ho interazione, 8 core assegnati), mentre sono sullo schermo sono Foreground (6 core little+Medium), e se non lo sono, sono Background (4 core Little).

PROVA DI 20 CICLI DI SWIPE TRA LE 2 APPLICAZIONI

###Ricordo che le statistiche riguardano l'intero sistema e non le singole app###

STATISTICHE VERSO APP 1 (FILES → DIALER)

Media dati di 20 esecuzioni del Transitorio App2 → APP1

Cache Miss = 10291300

Instructions = 757450207

CPU Cycles = 1582750635

Istruzioni per ciclo: 0.47856572617960125

BP Miss = 9225329

dTLB Walks = 1480982

iTLB Walks = 512464

Media dati di 20 esecuzioni a Regime dell'App 1 (DIALER)

Cache Miss = 20125868

Instructions = 1762777331

CPU Cycles = 3004833993

Istruzioni per ciclo: 0.5866471609102301

BP Miss = 16883841

dTLB Walks = 2568645

iTLB Walks = 863252

STATISTICHE APP 2 (DIALER → FILES)

Media dati di 20 esecuzioni del Transitorio APP1 → APP2

Cache Miss = 27420905

Instructions = 2193380641

CPU Cycles = 4212301774

Istruzioni per ciclo: 0.520708334464168

BP Miss = 18773721

dTLB Walks = 3188968

iTLB Walks = 1168315

Media dati di 20 esecuzioni a Regime APP2 (Files)

Cache Miss = 16340186

Instructions = 1406287674

CPU Cycles = 2376478832

Istruzioni per ciclo: 0.5917526615696781

BP Miss = 9406056

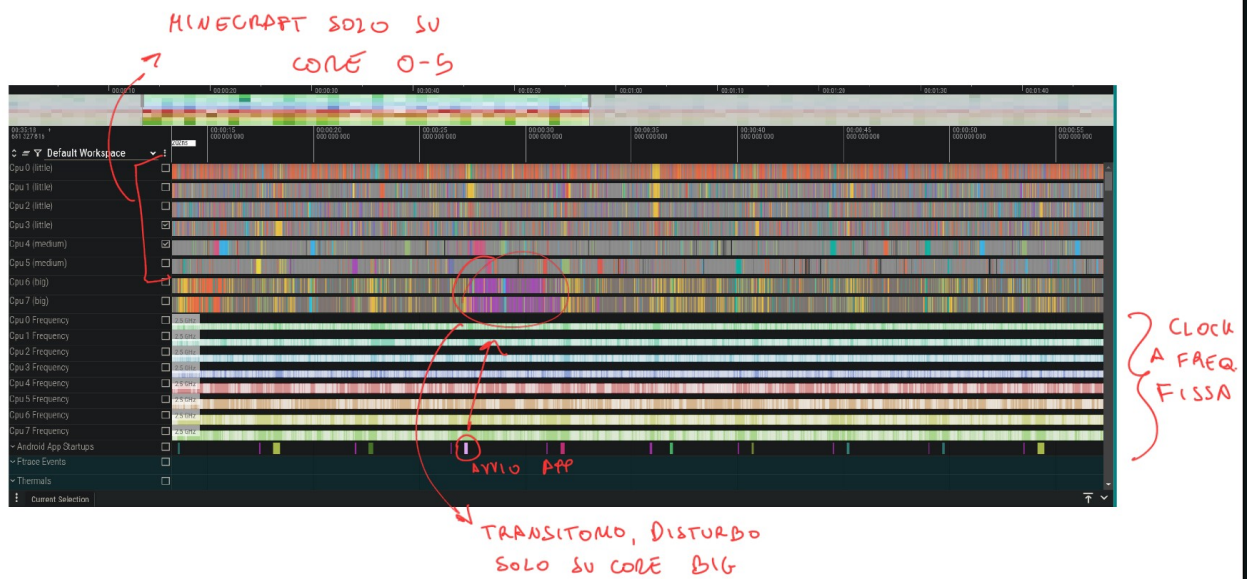
dTLB Walks = 1502386

iTLB Walks = 521166

TUTTI I RISULTATI PRECEDENTI DI OGNI TEST SONO STATI VERIFICATI CONTROLLANDO PER OGNI ESECUZIONE LA CORRISPONDENTE TRACCIA CON PERFETTO.

In particolare è stata verificata la corretta assegnazione dei task ai rispettivi core, il clock fisso dei singoli core e confermato il comportamento atteso di ogni singolo test.

Esempio di Traccia Test 1 – MC sui core 0-5 e Disturbi sui core 6-7



In primis notiamo che I Clock dei core rimangono ad una frequenza fissa e stabile.
I Disturbi sono solo sui core 6-7, mentre Minecraft (Grigio) solo sui core 0-5.
Possiamo dalla traccia confermare il comportamento atteso.