

Estructura del Computador

Fonaments dels Computadors – Grau en Intel·ligència Artificial

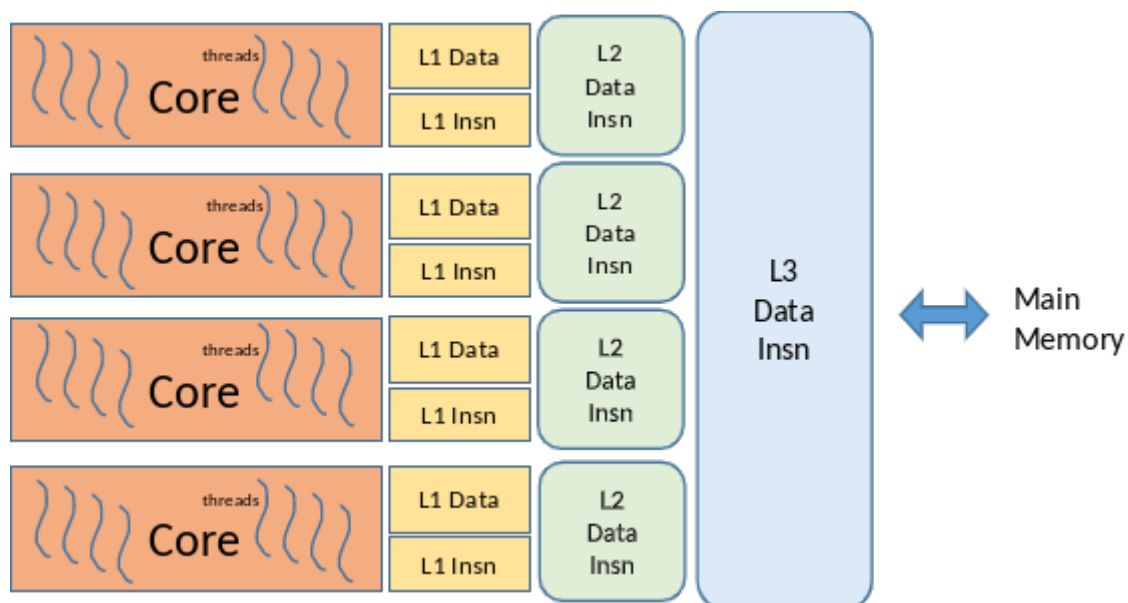
Facultat d'Informàtica de Barcelona – Departament d'Arquitectura de Computadors

Aquest tema de laboratori introdueix diversos conceptes relacionats amb el maquinari dels computadors. Encara que el GIA no sigui un grau orientat a arquitectura de computadors, per tal d'usar els computadors eficientment és necessari tenir experiència en la seva estructura i funcionament bàsic, així com en l'obtenció d'informació interna detallada sobre la seva composició, cosa que ajuda a entendre com l'estructura dels computadors té un impacte en el rendiment de les aplicacions que hi executem.

En aquesta sessió de laboratori treballarem amb algunes comandes que proporcionen informació sobre les característiques de l'ordinador on estem treballant. Nosaltres us introduïm aquestes comandes, però també serà bo que vosaltres consulteu les pàgines del seu manual ("man"), per completar la informació sobre el seu ús i les possibilitats que donen. També us proporcionem una col·lecció d'aplicacions de prova, que podreu utilitzar per analitzar l'impacte de diferents maneres d'executar-les sobre el seu rendiment.

El Processador i els seus Components

A les sessions de teoria hem vist que el processador té diversos components molt interrelacionats que executen o donen suport a l'execució de les aplicacions:



Els *hardware threads* (fluxos) executen les instruccions. Quan parlem a nivell hardware, els hardware threads són el conjunt de recursos hardware bàsics per executar instruccions. Mentre que a nivell software, els software threads són la seqüència d'instruccions que executaran els hardware threads. Habitualment comparteixen alguns dels recursos del *core* (nucli), per exemple les unitats que implementen l'aritmètica de coma flotant. Per accedir a memòria els threads passen per la jerarquia de memòria, formada per diversos nivells de memòria *cache* (memòria cau). Els més propers (L1 Data / L1 insn) són més petits i més ràpids. Els més llunyans (L3) són més grans i més lents. Finalment, se surt del xip per anar a memòria principal, que té més quantitat de memòria, però encara més lenta.

Els sistemes operatius compatibles amb Linux tenen la informació del sistema recopilada en un directori (“/proc” - process information pseudo-filesystem). Aquest directori conté moltes dades sobre les característiques de l’entorn d’execució, i sobre l’estat actual del que està passant en el sistema. Pots trobar informació detallada de la informació que proporciona el “/proc”, mirant la seva pàgina de manual (man 5 proc).

Com a primer pas, mirarem la informació relacionada amb el processador, que podeu trobar al fitxer “/proc/cpuinfo”. Observeu el seu contingut (podeu fer servir la comanda “less”). Observeu que presenta les característiques dels fluxos del processador. Entre elles veureu el número de fluxos (“siblings”), que es refereix al número total de hardware threads que estan en el mateix xip. Cada flux pot executar una seqüència d’instruccions (un software thread) independent d’un programa.

També podràs veure el camp “cpu cores”, que es refereix al número de cores integrat en el mateix xip. En alguns processadors, el número de “siblings” és el doble del número de “cpu cores”, i això significa que cada nucli (“core”) té dos fluxos. Si el número de “siblings” és igual que el número de “cpu cores” vol dir que només tenim un hardware thread per nucli (en un processador d’Intel, pot significar que no suporta hyperthreading, o que aquest no està habilitat).

Hi ha una comanda en particular, “lscpu”, que presenta informació addicional, a banda de la que dona el “/proc/cpuinfo”. La sortida pot utilitzar una nomenclatura una mica diferent... per exemple:

- CPU(s): 8 que indica el número de fluxos (“siblings”)
- Thread(s) per core: 2 el número de fluxos per nucli
- Core(s) per socket: 4 el número de nuclis per xip
- Socket(s): 1 el número de xips que té l’ordinador

NOTA: si el teu sistema Linux no inclou la comanda “lscpu”, pots instal·lar-la amb aquesta comanda. Recorda que només pots instal·lar software en el teu ordinador, on tens accés com a administrador. Als ordinadors de la Facultat, la comanda “lscpu” hauria d’estar instal·lada. En les següents comandes, recorda que els caràcters “\$>” representen el símbol de l’interpret de comandes (prompt). Un cop feta la instal·lació, prova un altre cop a executar “lscpu”:

```
$> sudo apt-get update
$> sudo apt-get install util-linux
```

NOTA: si el teu sistema és un Mac M1 o M2, és molt probable que aquesta comanda presenti problemes. En aquest cas, els exercicis relacionats amb Linux recomanem que els facis en els ordinadors del laboratori de la FIB. Encara que utilitzis màquina virtual, hem detectat problemes.

Exercici 1

Com a les sessions prèvies, obre un nou fitxer “respostes.txt”, com a part de l’entrega que faràs d’aquesta sessió. Indica en aquest fitxer la següent informació del teu sistema. Pots fer servir el teu Linux, el Linux de la màquina virtual o el Suse Linux dels ordinadors de la FIB:

- Arquitectura, Model del processador
- Número de xips (sockets), número de nuclis per xip, número de fluxos per nucli
- Freqüència del processador i l’ordre dels bytes (“byte order” o “endianness”)

Pots obtenir la mateixa informació d’un sistema operatiu Windows fent servir les següents comandes de PowerShell. Si el teu ordinador està executant Windows nativament, pots provar-les fàcilment:

Característiques del processador

```
$ Get-CimInstance -class CIM_Processor | Select-Object * | more
```

Informació sobre la memòria

```
$ Get-CimInstance -class CIM_Chip | Select-Object * | more
```

Informació de la cache

```
$ Get-CimInstance -class CIM_CacheMemory | Select-Object * | more
```

També podeu obtenir la informació en els sistema operatiu Mac/OS, usant les eines de l'Escriptori. Si el teu ordinador està executant Mac/OS nativament, pots fer servir aquests passos:

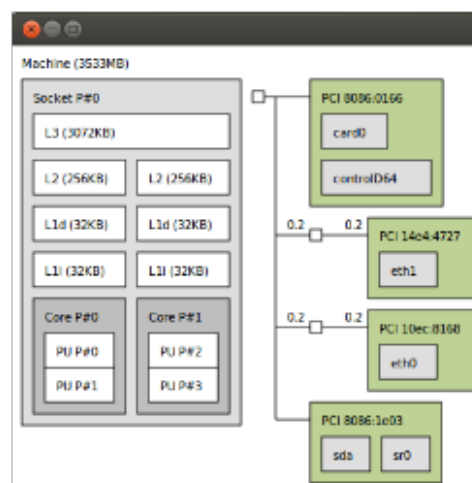
- Ves al menú de la poma del Mac/OS, i selecciona “About This Mac”. Veuràs la versió del sistema Mac/OS, el model del processador i la quantitat de memòria que té el sistema.
- En el mateix diàleg, prem el botó “More Info...”. El nou diàleg et mostrarà més informació:
 - Model de la màquina, número de fluxos, nuclis...
 - Discos instal·lats
 - Altres dispositius disponibles...

A Mac/OS també hi ha la comanda “system_profiler”, que pots fer servir des de l'interpret de comandes.

A Linux, hi ha una altra comanda, “lstopo”, que mostra una representació gràfica dels elements del computador i com estan connectats. Aquesta comanda, inclosa en el paquet “hwloc”, no està instal·lada per defecte en molts Linux. Si estàs fent servir la imatge d'Ubuntu en màquina virtual, pots instal·lar-la fent:

```
$> sudo apt-get install hwloc
```

Un cop instal·lat el paquet, prova la comanda “lstopo”, executant-la des del terminal. Veuràs una representació gràfica de la màquina virtual, semblant a aquesta:



NOTA: Les caixetes amb inicials “PU” dins dels cores fan referència a “Processing Unit”, és a dir hardware thread.

Amb aquesta eina podràs veure l'organització dels components de qualsevol ordinador amb Linux. Pots aconseguir la mateixa informació en format text de dues formes diferents:

```
$> lstopo -v  
$> lstopo-no-graphics
```

Exercici 2

Respon en el teu fitxer “respostes.txt” indicant, a partir de la sortida d'aquestes dues comandes (`lstopo -v` i `lstopo-no-graphics`), quin és el component de maquinari que està connectat a cada ranura del bus PCI a la teva màquina o màquina virtual. **NOTA:** per poder contestar has d'investigar-ho analitzant les etiquetes que et mostra la sortida de les línies de comandes, així com els components destacats en les transparències del Part 1 del Tema 2 de teoria.

La Jerarquia de Memòria

Amb les comandes que hem explicat a la secció anterior, també podràs trobar informació sobre la jerarquia de memòria.

Exercici 3

A partir de la sortida que obtens amb l'`lstopo`, indica en el fitxer “respostes.txt”:

- Els nivells de memòria cau que té el teu ordinador (propri o el node de la FIB que fas servir)
- Per a què es fa servir cada nivell de la memòria cau (instruccions, dades o les dues coses)
- Les mides que té cada nivell
- Si el nivell de memòria cau és privat per a cada flux o compartit pels fluxos dins el core

Hi ha una altra eina interessant, “numactl” (NUMA control), per veure i actuar sobre la política d'ús de la memòria, quan l'ordinador té diversos xips (sockets), cadascun amb el seu mòdul de memòria proper. Pel següent exercici, et suggerim que utilitzis la comanda:

```
$> numactl -s # (show) mostra la informació sobre la política de memòria que fa servir el procés  
$> numactl -H # (hardware) mostra la informació de l'estructura dels nodes de memòria, mides...
```

Et permetrà comprovar la informació dels mòduls de memòria. És molt possible que en la teva màquina en tinguis un de sol. Si no tens la comanda “numactl” disponible en el teu sistema, pots instal·lar-la amb:

```
$> sudo apt-get install numactl
```

Exercici 4

Indica quants nodes de memòria trobes en el teu ordinador, la seva mida en MB, i quanta memòria te el sistema lliure (no utilitzada encara, “free”). Aquest serà el resum de l'estat de la memòria del teu sistema. Escribe la informació en el teu fitxer “respostes.txt”.

Dispositius Externs basats en USB

Veurem ara la comanda “lsusb” (list USB). Aquesta comanda es fa servir per obtenir informació del subsistema USB i els dispositius que hi tenim connectats, que poden ser el teclat, el ratolí, el lector

d'empremptes digitals, el lector de targetes SD... i molts més. Invoca-la sense arguments, o amb l'argument "-v" (verbose), que et donarà més informació.

També tenim disponible la comanda "lshw" (list hardware), que llista els dispositius que tenim en el sistema, incloent la placa base, processadors, memòria, connectors PCI, pantalla, USB, discos... Si no el tens disponible pots instal·lar-lo fent: `$> sudo apt-get install lshw`. Llavors prova:

```
$> lshw
```

Exercici 5

Inclou en el teu fitxer "respostes.txt", la llista de dispositius USB que has trobat en el sistema. No et preocupis dels detalls amb els que no siguis familiar. Fixa't en els dispositius que hem comentat a classe (processador, caches, memòria, pantalla, discos, xarxa...), i intenta trobar el significat d'alguna de les seves característiques (la mida dels discos, o el tipus de connexió de xarxa...).

Discos

La comanda "df" (disk free) et donarà informació sobre els discos, les seves particions i quin sistema de fitxers té cadascuna, i l'espai de cadascuna i quant d'espai encara està disponible. Et suggerim invocar-la així:

```
$> df -H
```

(on H indica que la sortida sigui en format "humà" i usant les unitats del Sistema Internacional - SI). Veuràs la informació de les particions que té el teu disc o discos, si n'hi ha més d'un a l'ordinador. Per a cada partició veuràs (les dades entre parèntesi són exemples):

- el nom del dispositiu (/dev/sda6)
- la mida total en KBytes (247G)
- la quantitat de dades en el disc en KBytes (144G)
- la quantitat d'espai lliure (103G)
- el percentatge d'ocupació (100% indica que està ple)
- el directori on el trobem disponible pel sistema (on està muntat, /users)

(estudiarem més detalls de com funcionen i es tracten els discos i els sistemes de fitxers en temes posteriors durant el curs)

Hi ha una altra comanda, "hdparm", que mostra les característiques del disc dur indicat. Per executar aquesta comanda cal ser superusuari. Per tant, l'executem amb "sudo":

```
$> sudo hdparm -I <disc>
```

D'aquesta manera, amb la comanda sudo, i indicant la nostra paraula de pas, obtenim els permisos necessaris per executar la comanda "hdparm" amb els arguments -I (i llatina majúscula, per obtenir informació detallada del disc) i el "disc" a examinar. "hdparm" mostrarà les característiques del disc. Habitualment, per referir-nos als discos fem servir el nom amb els que el sistema operatiu els coneix. En les nostres màquines habituals serà un fitxer especial de l'estil "/dev/sda", que significa el primer disc ("sda") que ha reconegut el sistema en iniciar-se. Aquests noms els ha mostrat abans la comanda "df", el podeu treure del seu llistat. Amb "hdparm" només podrem veure les característiques dels discos reals. "df" mostra diversos discos virtuals (udev, tmpfs...) dels que no mostrarà res, justament perquè són virtuals, no tenen un disc físic associat.

Exercici 6

Intenta identificar les característiques dels discos que hem comentat a classe (model, mida del sector, mida total) a la sortida de la comanda “hdparm”, i descriu el que trobes en el teu fitxer “respostes.txt”.

Què passa si...?

En aquesta secció treballarem amb diversos aspectes de l'entorn d'execució. Per fer-ho, baixa el fitxer d'exemples que tenim a la web:

https://docencia.ac.upc.edu/FIB/GIA/FC/documents/Lab/S7/S7-FC_GIA.tar.gz

Extreu el contingut d'aquest fitxer. Pots usar la comanda:

```
$> tar xvf S7-FC_GIA.tar.gz
```

veuràs que en el nou directori hi ha 3 fitxers en C, el Makefile que ens permet compilar-los, i un fitxer de comandes pel “bash” (shell script, “launch.sh”). En els següents exercicis, si estàs executant en màquina virtual, intenta que la màquina ofereixi almenys dos nuclis o fluxos al sistema Linux.

Exercici 7

Compila els codis d'exemple amb la comanda “make”. Com a resultat, tindràs 3 fitxers executables: “floats”, “integers” i “mems”. Escriu en el teu fitxer “respostes.txt” la comanda i els arguments que has fet servir per compilar-los els 3 de cop.

El fitxer script “launch.sh” conté una sèrie de comandes que interpretarà l'interpret de comandes “bash” en el teu terminal. Per executar-lo, aquest fitxer rep parells d'arguments. De cada parell, el primer argument indica en quin flux (hardware thread) s'executarà la comanda indicada en el segon argument del parell. Per exemple, si teclegem:

```
$> ./launch.sh 0 ./integers 1 ./floats 1 ./mems
```

“launch.sh” executarà el programa “./integers”, en el hardware thread 0, i els programes “./floats” i “./mems”, en el hardware thread 1.

NOTA: *aquelles persones que utilitzeu un portàtil Mac M1 o M2 podeu tenir problemes per a realitzar correctament aquesta vinculació. Per això, en aquests casos, s'aconsella fer-ho en els terminals del laboratori de la FIB.*

Et recomanem que obris un segon terminal per poder executar de forma separada la comanda “top”. Aquesta comanda mostra l'estat actual del sistema i els processos que hi estan corrent. La seva sortida s'actualitza automàticament cada 2 segons (per defecte). La freqüència d'actualització es pot canviar prement el caràcter ‘s’ (sleep) i entrant-li llavors el nou temps d'espera entre actualitzacions. Aquest temps es pot donar amb un nombre en coma flotant.

Per sortir del “top”, només cal prémer el caràcter ‘q’ (quit).

Un cop el tinguis executant-se pots fixar-te en la columna que indica el percentatge de processador que rep cada procés (columna %CPU). Un cop engeguis les execucions a l'altre terminal amb la comanda “./launch.sh”, hauries de veure que els processos apareixen en execució a la finestra del top, amb el seu percentatge d'ús del processador.

Al terminal on executis el “./launch.sh”, quan cada programa acabi la seva execució veuràs també que dona informació sobre el temps d'execució que ha trigat, per exemple:

```
3.20user 0.00system 0:04.12elapsed 99%CPU ...
4.93user 0.01system 0:05.04elapsed 99%CPU...
```

Fixa't en el temps "elapsed", 4.12 i 5.04 segons, i el percentatge de CPU, 99% en la sortida anterior. El temps indica en minuts i segons el temps invertit per cada programa en executar-se.

La distinció entre el temps mostrat per "user" i "system" l'aclarirem en l'últim tema del curs (està relacionat amb el sistema operatiu). Ara bé, sense entrar en aquests detalls, els dos mesuren el temps que han estat executant instruccions en la CPU. La suma d'aquests dos temps pot ser igual o inferior al temps mostrat per "elapsed". Aquest temps inclou tant el període de temps que el programa ha estat executant instruccions en la CPU, així com el temps que NO ho ha fet. Per tant, abarca des de que ha sigut llençat a executar fins que ha acabat. El percentatge de CPU indica el consum mig que ha fet de processador (de hardware thread), en aquesta execució concreta. Mentre duren les execucions dels programes, intenteu no fer altres execucions a l'ordinador, de forma que no alterem els resultats de les mesures de temps i ús del processador.

NOTA: per fer correctament els següents exercicis, has d'esperar a que sortin tantes línies de sortida amb el temps d'execució com número de programes hagi llençat a executar amb la línia de comandes del "launch".

Exercici 8

Respon a les següents preguntes en el teu fitxer "respostes.txt":

- a) *La línia de comanda que fas servir per executar una sola instància del binari "./integers" en qualsevol hardware thread (pista: mireu com fer-ho al manual del "taskset")*
- b) *El temps d'execució "elapsed" i el %CPU obtinguts per l'execució del punt a) seran els que farem servir de rendiment base obtingut per aquest programa*
- c) *La comanda que fas servir per executar una sola instància de "./floats" en qualsevol hardware thread*
- d) *El temps "elapsed" i el %CPU obtinguts seran el rendiment base per aquest programa*
- e) *La comanda que fas servir per executar una sola instància de "./mems" en qualsevol hardware thread*
- f) *El temps "elapsed" i el %CPU obtinguts seran el rendiment base per aquest programa*

Si el teu ordinador executa aquests programes molt de pressa, diguem en menys del 10 segons, pots modificar els codis en C, per fer que el número d'iteracions del bucle extern (el bucle "j") sigui 10 vegades més, de forma que repetirà la feina 10 cops, i trigarà més.

Contenció en l'Ús de Recursos de l'Ordinador

Ara analitzarem què passa quan executem diverses instàncies del mateix programa en fluxos hardware diferents, cadascun en un nucli. Useu la informació que heu obtingut durant la primera part d'aquesta sessió de laboratori (de l'"lstopo") per saber com poder assignar dues execucions en fluxos hardware que no siguin del mateix nucli.

Exercici 9

Inclou en el teu fitxer "respostes.txt" la informació següent per cada tipus de programa (e.g. "./integers", "./floats", "./mems"), tot i vinculant els programes en fluxos en diferents nuclis:

- a) *La línia de comanda per executar cada experiment*
- b) *La mitjana dels temps "elapsed" i %CPU obtinguts*

- c) *Analitza i explica breument si hi ha cap diferència amb els temps base que vas apuntar a l'Exercici 8*

El següent pas consisteix en lligar les instàncies del mateix programa en el mateix nucli, però en diferent hardware thread. En el cas que la teva màquina virtual tingui un sol flux per nucli, pots optar per a) reconfigurar-la (has d'apagar-la prèviament sense mantenir l'estat de la màquina virtual. Després entrar a "paràmetres" -> "sistema" -> "Processador") per incloure els dos fluxos per nucli de la màquina nativa a dins de la màquina virtual (confirma si després del canvi ha creat 2 hardware threads en el mateix core o dos cores amb un hardware thread cadascun), o b) executar els dos programes en el mateix flux.

Exercici 10

En el teu fitxer "respostes.txt", respon a les següents preguntes per cada experiment:

- a) *La comanda que fas servir per executar cada experiment per cada programa*
- b) *La mitjana dels temps "elapsed" i %CPU obtinguts*
- c) *Analitza i explica breument si hi ha cap diferència amb els temps base que vas apuntar pel programa corresponent en l'Exercici 8. Indica també si trobes cap patró de comportament en el camp de "%CPU" que dona la comanda "top" durant l'execució dels experiments*

Finalment, executa experiments barrejant diferents tipus de programes, lligant-los al mateix nucli, però no al mateix hardware thread. Com a l'exercici 10, pots provar de reconfigurar la teva màquina virtual per que tingui disponibles més nuclis i hardware threads. Si no pots disposar d'aquesta configuració, no és necessari que facis l'exercici 11.

Exercici 11

Respon en el teu fitxer "respostes.txt" les següents preguntes per a cada experiment:

- a) *La línia de comanda que fas servir per executar cada parella de programes*
- b) *Els temps d'execució "elapsed" i %CPU individuals que obtens. En aquest cas, com que els programes són diferents, no podem fer les mitjanes*
- c) *Analitza i explica breument si les execucions mixtes de programes causen algun impacte en els temps d'execució o percentage de CPU, comparant els resultats amb les dades base de cada programa que tens de l'Exercici 8. Indica també si trobes cap patró de comportament en el camp de "%CPU" que dona la comanda "top" durant l'execució dels experiments*

Entrega de la pràctica

Per empaquetar tots el fitxers que heu modificat i el vostre "respostes.txt" en un sol fitxer per entregar, feu:

```
$ tar czvf sessio7.tar.gz respostes.txt Makefile *.c
```

I ara podeu anar al Racó i enviar aquest fitxer "sessio7.tar.gz" a la pràctica corresponent.