Università degli Studi di Napoli "Federico II"

Scuola Politecnica e delle Scienze di Base
Corso di Laurea Magistrale in Ingegneria Informatica

Giuseppe Francesco Di Cecio - M63001211
Nicola D'Ambra - M63001223
Emma Melluso - M63001176

Elaborato di *Impianti di Elaborazione*

Università degli Studi di Napoli *Federico II*
Napoli
A.A 2020/2021

Indice

Capitolo 1

Workload Caracterization

Il dataset di partenza è composto da **3000 righe** e **24 colonne**, ciascuna delle quali rappresenta uno dei parametri del sistema oggetto di studio. Si tratta di parametri caratterizzanti l'esecuzione di vari Threads su un sistema operativo.
In particolar modo le colonne con prefisso Vm rappresentano informazioni sulla memoria virtuale occupata e utilizzata dai Threads, mentre le altre colonne rappresentano informazioni di carattere generale, come memoria libera, numero di threads, pagine inattive ecc.

1.1 Filtraggio

1.1.1 Colonne Identiche

Innanzitutto osservando il workload e effettuando un grafico delle distribuzioni ci rende conto della presenza di ben 4 colonne costanti:

- **Active**
- **AnonPages**
- **AvbLatency**
- **Error**

Essendo tali non spiegano varianza, dunque possono essere tranquillamente trascurate ai fini dell'analisi.
Osservando le distribuzioni dei parametri **WriteBack** e **MemFree** si sono notate alcune caratteristiche comuni. Per avere una maggiore chiarezza si è preferito calcolare la matrice delle correlazioni su questi due parametri.

	'MemFree'	'Writeback'
'MemFree'	1,0000	1,0000
'Writeback'	1,0000	1,0000

Figura 1.1: *Matrice di correlazione tra MemFree e WriteBack*

Osservando la matrice appare evidente che le due colonne sono esattamente identiche, fornendo quindi la stessa informazione. Per questo motivosi è deciso di trascurare una delle due, in particolare quella di WriteBack.

Le 24 colonne iniziali sono state ridotte a 19 colonne, riducendo il dataset di un numero di osservazioni pari a:

$$n_{dati} = 3.000 \times (24 - 19) = 15.000 \tag{1.1}$$

1.1.2 Outlier

Gli outliers sono valori che si discostano notevolmente dalle altre osservazioni. Non possono essere subito eliminati poiché influiscono sulle analisi statistiche in modo considerevole. Si possono però analizzare per capire se è possibile rimuoverli o meno. Osservando l'andamento dei seguenti parametri, attraverso box plot e grafici di distribuzione:

- **VmSize**, quanta memoria virtuale utilizza l'intero processo.
- **VmHWM**, di quanta RAM il processo necessita al massimo.
- **VmRSS**, quanta RAM il processo sta correntemente usando.
- **VmPTE**, quanta memoria Kernel è occupata dalle entries della tabella delle pagine.

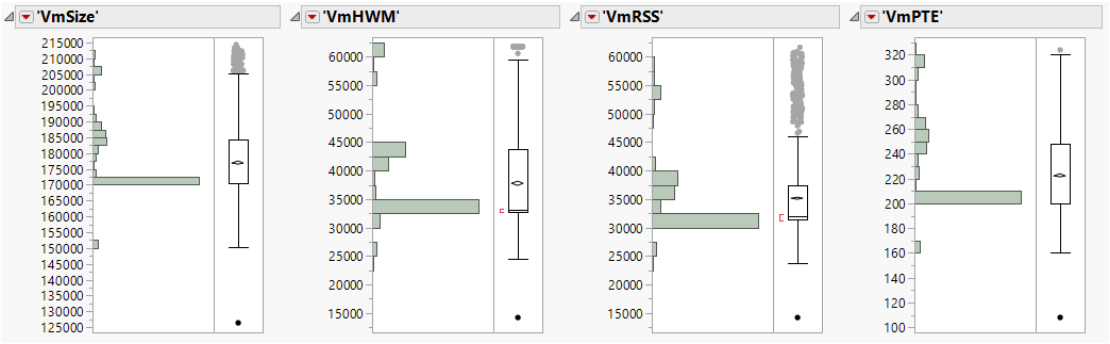

Figura 1.2: *Grafici di distrubuzione di VmSize, VmHWM, VmRSS, VmPTE*

Si è notato che essi presentano un outlier isolato (in basso ad ogni grafico) in comune associato alla prima riga del dataset. Analizzando gli altri parametri (*MemFree*, *Dirty*, *PageTables*, *Buffer*, ...) è stato possibile evidenziare che anche per la maggior parte di essi lo è, ma non è un punto isolato.
L'ipotesi fatta è che con molta probabilità le prime righe del dataset (da 0 a 100 circa), rappresentano la fase di avvio del processo e l'outlier oggetto di studio è la prima istanza di questa fase. Dato che l'obiettivo della caratterizzazione del workload è quello di analizzare le prestazioni a regime del sistema oggetto di studio (in questo caso), si è deciso di trascurare quel singolo outlier. E' bene notare che in ogni caso le informazioni riguardo questa fase di avvio non saranno del tutto perse dato che è stato rimosso un singolo punto e non tutti i punti che la rappresentano.

Un secondo outlier che può essere agevolmente rimosso è la riga 512 in cui il parametro **Slab** assume valore 4.

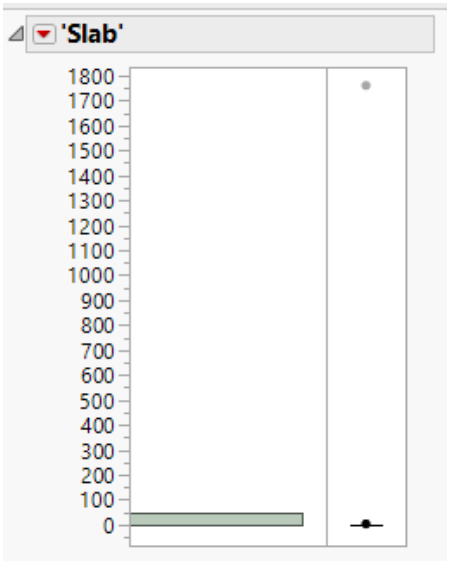

Figura 1.3: *Grafico di distrubuzione di Slab*

Oltre ad avvicinarsi molto al valore medio assunto da Slab (zero), esso risulta essere un outlier solo per il parametro stesso dato che per gli altri è un valore compreso tra i quartili. Una sua rimozione quindi non influenza gli indici di caratterizzazione sintetica dei parametri del workload complessivo.

Un terzo outlier è il valore 1760 del parametro *Slab*, associato alla riga 90 del workload.

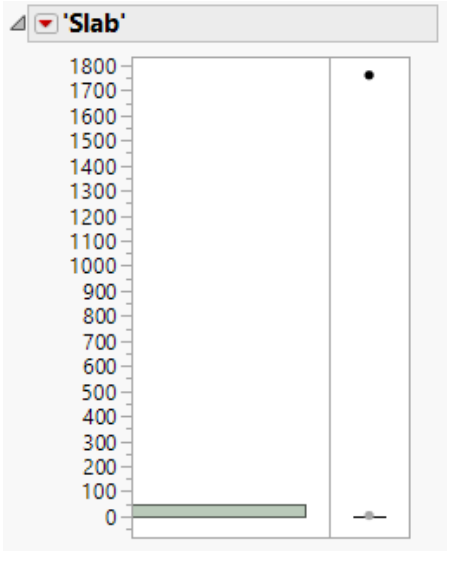

Figura 1.4: *Grafico di distrubuzione di Slab*

Rispetto al precedente, tale outlier richiede un' analisi più approfondita visto che influenza significativamente l'andamento di parametri quali *Mapped* e *PageTables*. Per descrivere meglio la dipendenza tra questi parametri si può effettuare un grafico tra il numero dell'osservazione e il valore assunto da *Mapped*, analogo discorso con *PageTables*.

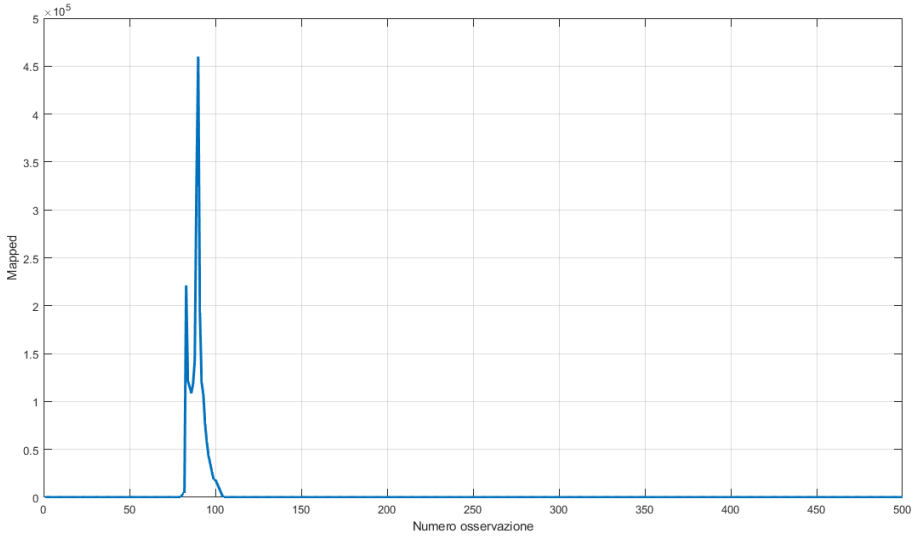

Figura 1.5: *Grafico tra numero di osservazione e valore assunto dal parametro Mapped*

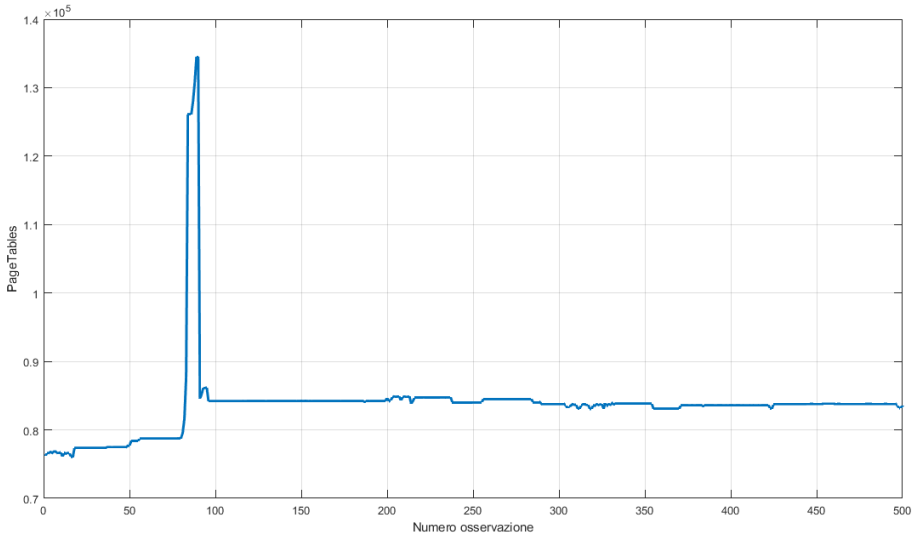

Figura 1.6: *Grafico tra numero di osservazione e valore assunto dal parametro PageTables*

Si nota che in corrispondenza (in realtà nell'osservazione appena precedente) dell'outlier del parametro *Slab* i due paramentri sopra indicati hanno un picco, durante la fase di avvio del sistema.
Lo Slab si riferisce ad un particolare meccanismo di allocazione/deallocazione della memoria nel Kernel. Dato che influenza in particolar modo altri paramentri si è preferito di non trascurarlo.

In conclusione sono stati eliminati dal dataset solo 2 outlier che corrispondo a 38 righe. Quindi il dataset è stato ridotto in totole di 15.038 elementi, provocando una diminuzione dei dati iniziali di poco più del 20%.

1.2 PCA

A seguito del filtraggio il dataset risulta ridotto grazie alla rimozione di alcune colonne e righe che rappresentano outlier.
Sul questo dataset si possono quindi iniziare a fare le prime considerazioni.
Utilizzando la tecnica della *Principal Component Analysis* il dataset può essere estremamente ridotto, sfruttando solo le *Componenti Principali* che mantengono più varianza.
Il risultato della PCA è quindi:

Numero	Autovalore	Percentuale	20 40 60 80	Percentuale cumulativa
1	9,8395	51,787		51,787
2	3,6873	19,407		71,194
3	2,4576	12,935		84,129
4	0,7782	4,096		88,225
5	0,7075	3,724		91,949
6	0,6258	3,293		95,242
7	0,3409	1,794		97,036
8	0,2658	1,399		98,435
9	0,1612	0,848		99,283
10	0,0904	0,476		99,759
11	0,0178	0,094		99,853
12	0,0114	0,060		99,913
13	0,0078	0,041		99,954
14	0,0058	0,030		99,985
15	0,0014	0,007		99,992
16	0,0009	0,005		99,997
17	0,0006	0,003		100,000
18	0,0000	0,000		100,000
19	0,0000	0,000		100,000

Figura 1.7: *PCA applicata al dataset filtrato*

La scelta del numero di componenti principali ricade in particolar modo sulla devianza che quelle componenti mantengono rispetto al dataset reale. Inoltre essa dipende anche dal tipo di osservazioni ed esperimento che è stato effettuato.
Per mantenersi in una regione di tolleranza si è scelto di utilizzare **6 Componenti Principali**, in modo da mantenere il $\sim 95\%$ della devianza totale. Andare oltre alle 6 componenti risulta svantaggioso, poiché la percentuale cumulativa aumenta di poco e per ogni componente in più vengono aggiunte circa 3000 righe (per questo particolare esperimento). Sulla base di ciò la devianza persa e mantenuta vale:

$$DEV_{PCA-PERSA} = 100 - 95\% \approx 5\% \qquad DEV_{PCA-MANTENUTA} \approx 95\%$$

Sulle queste componenti principali può essere eseguita la procedura di clustering.

1.3 Clustering

Il clustering è una tecnica che consiste nel raggruppare osservazioni "simili" tra loro. La similitudine tra un elemento e un cluster, o tra un cluster e un altro cluster, può essere

calcolata secondo varie tecniche. In questa analisi si è preferito utilizzare il **metodo di Ward**, il quale pesa la distanza tra due cluster in relazione al numero di elementi che li compongono.
Dati due cluster P e Q (un elemento non appartenente ad un cluster, può essere visto come un cluster di dimensione 1), sia $|P|$ la cardinalità di P, analogo con $|Q|$, e sia $\bar{x}_p$ il centroide di P, analogo con $\bar{x}_q$, la distanza tra P e Q viene calcolata come:

$$d(P,Q) = 2\frac{|P|\,|Q|}{|P|+|Q|}||\bar{x}_p - \bar{x}_q||^2$$

Per ogni raggruppamento viene poi scelto una singola osservazione che la rappresenta, riducendo quindi il dataset pari al numero di cluster scelti durante l'analisi.
A tal proposito il numero di cluster da scegliere può dipendere da vari fattori:

- Omogeneità dei cluster: i cluster devono raggruppare un numero di osservazioni quanto il più possibile omogeneo rispetto agli altri cluster. Avere un cluster con un numero di elementi di vari ordini di grandezza rispetto ad un altro cluster non sempre può portare a buoni risultati (in termini di devianza).
- Devianza mantenuta: a seguito della PCA parte della devianza nei dati viene persa. Dato che il clustering viene effettuato sulle *Componenti Principali* allora esso produce un'ulteriore perdita di devianza nel risultato finale.

1.3.1 Omogeneità

In una prima analisi, si possono scegliere dei cluster da valutare in modo che non ci siano troppe differenze (in termini di conteggio degli elementi) tra un cluster e un altro.
Analizzando quindi le PC con 5 cluster si ottiene:

Cluster	Conteggio
1	82
2	7
3	979
4	1929
5	1

Figura 1.8: *Numero di cluster e il relativo numero di elementi con 6 PC*

Come si nota dalla figura la differenza del numero di elementi tra un cluster e un altro è molto elevata, è preferibile quindi aumentare il numero di cluster in modo da diminuire tale fenomeno.

Cluster	Conteggio
1	82
2	7
3	152
4	711
5	116
6	1126
7	803
8	1

Cluster	Conteggio
1	82
2	7
3	152
4	497
5	214
6	116
7	639
8	487
9	803
10	1

Cluster	Conteggio
1	82
2	6
3	1
4	15
5	137
6	497
7	214
8	116
9	639
10	487
11	803
12	1

Cluster	Conteggio
1	82
2	6
3	1
4	15
5	137
6	497
7	214
8	116
9	639
10	188
11	299
12	400
13	403
14	1

Cluster	Conteggio
1	82
2	6
3	1
4	15
5	137
6	150
7	347
8	214
9	116
10	639
11	70
12	118
13	299
14	400
15	403
16	1

Cluster	Conteggio
1	82
2	6
3	1
4	4
5	11
6	137
7	150
8	347
9	214
10	54
11	62
12	639
13	70
14	118
15	299
16	400
17	403
18	1

Figura 1.9: *Numero di cluster e dimensione per diversi valori*

Eccetto qualche cluster (il cui motivo verrà spiegato in seguito), essi sono più o meno omogenei rispetto al primo caso.

1.3.2 Devianza Persa

Per questo homework si è scelto di analizzare un numero di cluster che va da 8 a 18, e valutare per ognuno di essi quanta devianza viene mantenuta con un numero di *Componenti Principali* scelto precedentemente.
Per effettuare il calcolo della devianza totale persa persa bisogna prima calcolare la devianza intra-cluster (la somma delle devianze per ogni cluster) e sulla base di questa si può calcolare la quantità richiesta.
Matematicamente, definita $DEV_{PCA-PERSA}$ la devianza persa (in termini percentuali) a causa della PCA, viceversa $DEV_{PCA-MANTENUTA}$ la devianza mantenuta dalla PCA, e DEV_{INTRA} la devianza intra-cluster, la devianza totale persa percentuale vale:

$$DEV_{PCA-LOST} + DEV_{INTRA} \times DEV_{PCA-MANTENUTA}$$

In seguto, calcolando le grandezze per ogni scenario definito, si possono racchiudere poi tutte le informazioni in una singola tabella:

8 Cluster	10 Cluster	12 Cluster	14 Cluster	16 Cluster	18 Cluster
15%	13%	12%	11%	9%	8%

1.4 Workload Sintetico