

Tesina di Impianti di Elaborazione

Gruppo X

Pafundi Vincenzo - Mat. XX/XXXX Riccio Marco - Mat. XX/XXXX
Russo Davide - Mat. M63/820

17 gennaio 2019

Indice

1	Benchmark	1
2	PCA & Clustering	2
2.1	Traccia	2
2.2	Soluzione	2
2.2.1	Analisi preliminare	2
2.2.2	Principal Component Analysis	7
2.2.3	Clustering	8
2.3	Calcolo analitico della varianza persa	9
2.3.1	Devianza post-PCA	9
2.3.2	Devianza post-clustering	9
2.3.3	Conclusione	10
2.4	Costruzione del workload sintetico	10
3	Capacity Test	12
4	Dependability	13
4.1	Esercizio 1	13
4.1.1	Traccia	13
4.1.2	Soluzione	13
4.2	Esercizio 2	15
4.2.1	Traccia	15
4.2.2	Soluzione	16
4.3	Esercizio 3	17
4.3.1	Traccia	17
4.3.2	Soluzione	18
4.4	Esercizio 4	19
4.4.1	Traccia	19
4.4.2	Soluzione	19
4.5	Esercizio 5	21
4.5.1	Traccia	21
4.5.2	Soluzione	23
4.5.2.1	Punto 1	23
4.5.2.2	Punto 2	23
4.5.2.3	Punto 4	26

5	FFDA	28
---	------	----

Capitolo 1

Benchmark

Capitolo 2

PCA & Clustering

2.1 Traccia

Analizzare i dati contenuti all'interno del file fornito, ottenuti dal monitoraggio di un file system di Unix. Si vuole ridurre il dataset ed ottenere un workload sintetico che sia rappresentativo di quello reale, preservando gran parte della varianza.

2.2 Soluzione

Per rispondere al problema si è scelto di utilizzare il tool JMP per l'analisi del dataset al fine di ridurre il numero di dati. Dopo un'analisi preliminare sui dati, la riduzione effettiva del dataset avviene utilizzando due tecniche: Principal Component Analysis (PCA) e clustering.

2.2.1 Analisi preliminare

L'analisi preliminare prevede, dopo aver importato il file nel tool, di analizzare le feature presenti, al fine di eliminare quelle prive di contenuto informativo. A tale scopo si è proceduto all'analisi delle distribuzioni di tali attributi osservando, in particolare, il coefficiente di variazione (CV) di ognuna. Il coefficiente di variazione è la normalizzazione della varianza con la media. Se il coefficiente di variazione è nullo, il parametro misurato è costante, e quindi si sceglie di non includere quella feature nelle successive analisi.

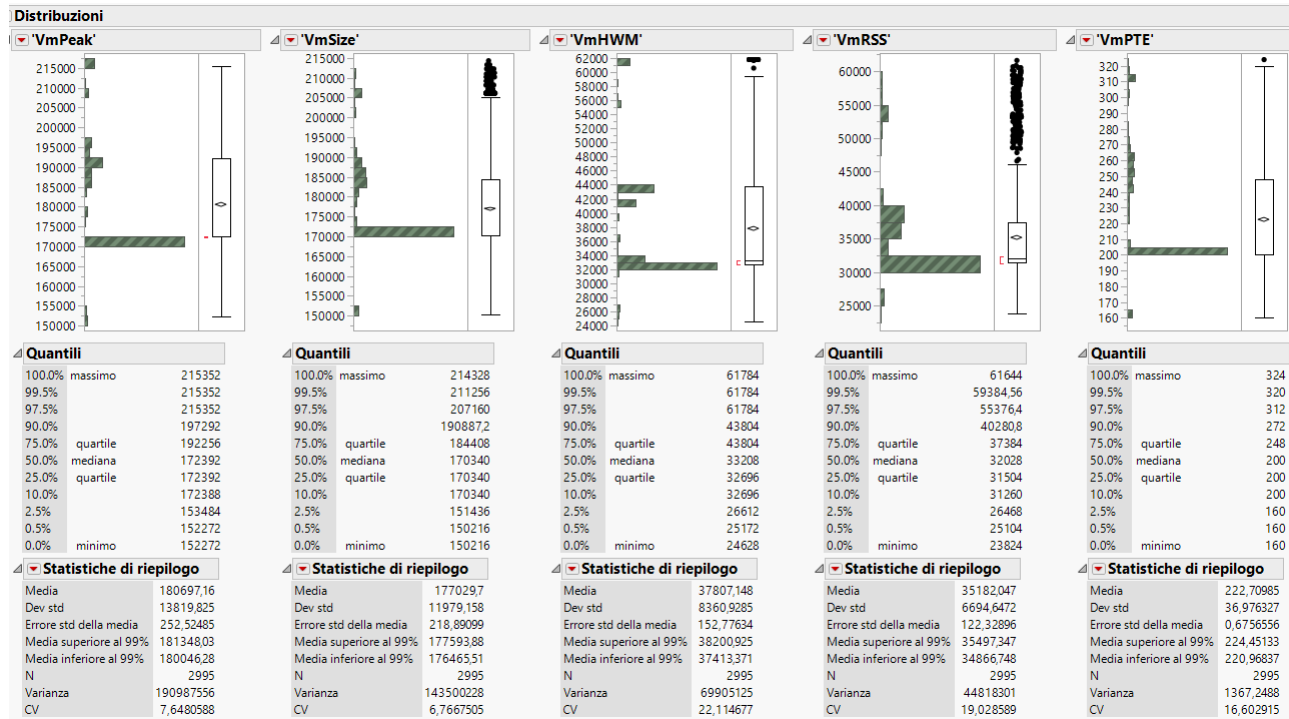


Figura 2.1: Distribuzioni delle feature

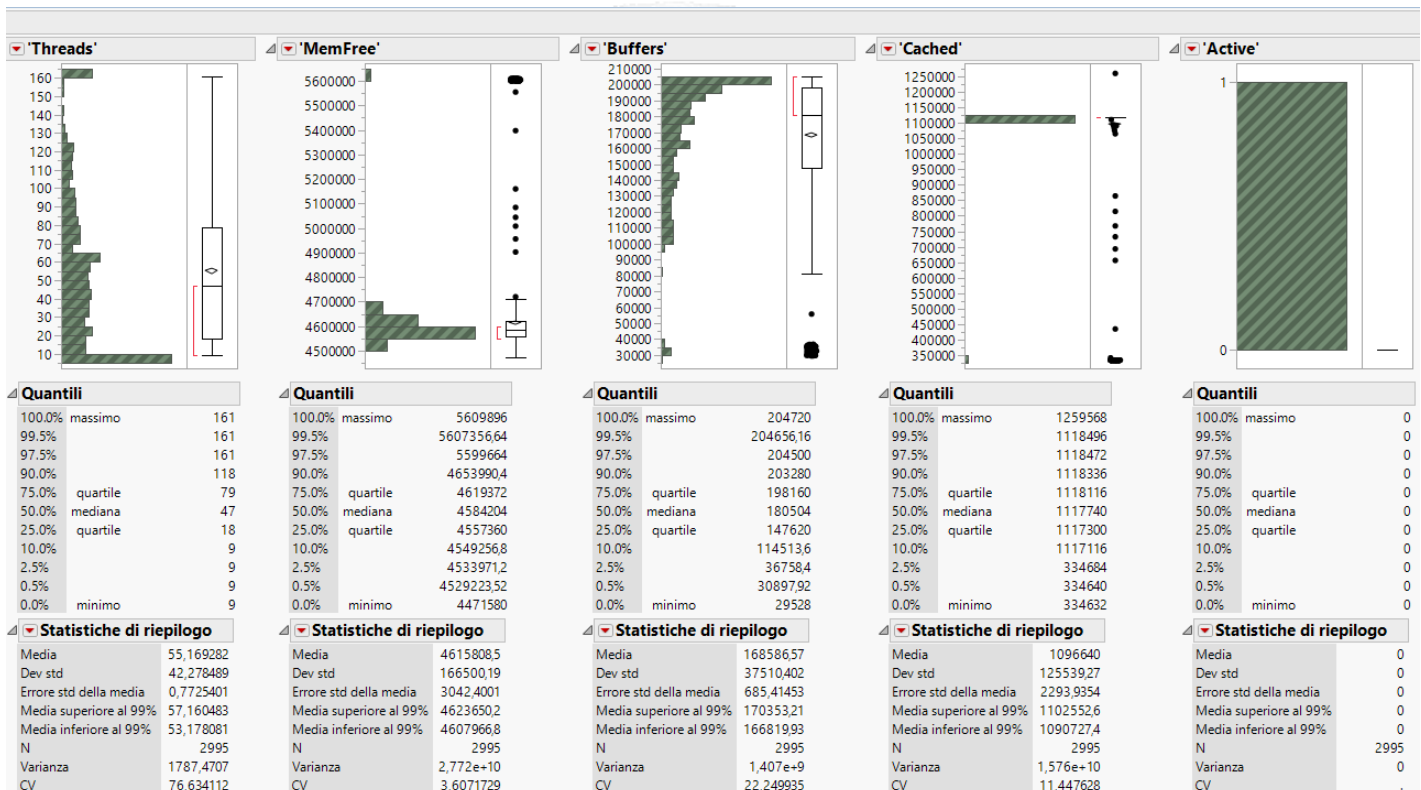


Figura 2.2: Distribuzioni delle feature

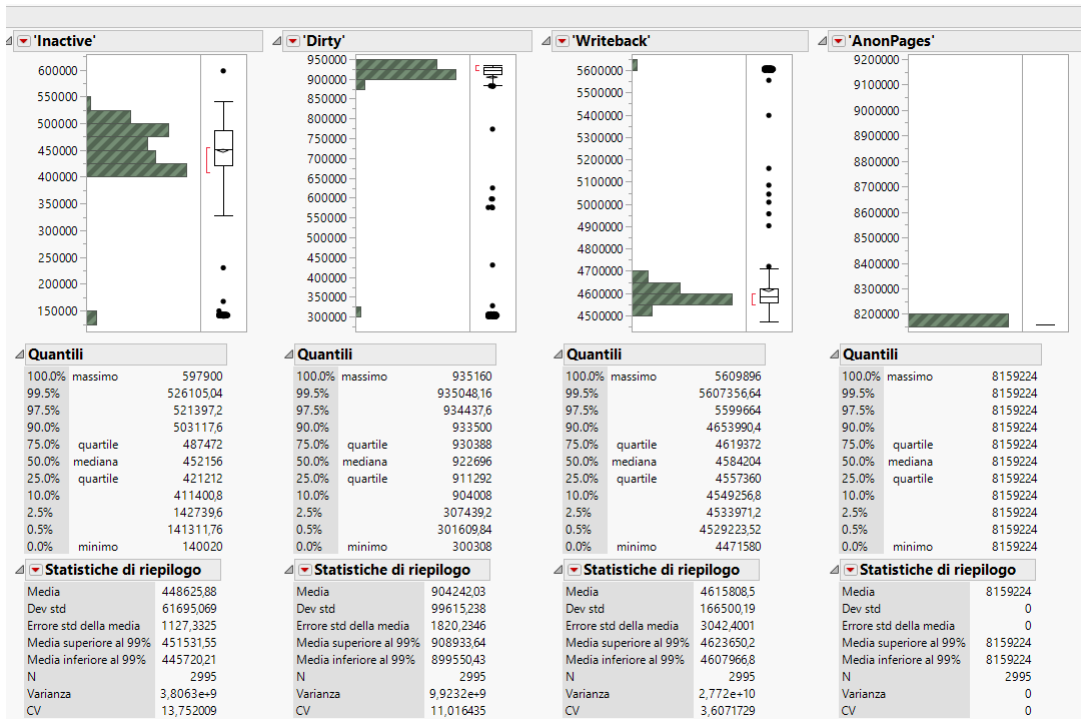
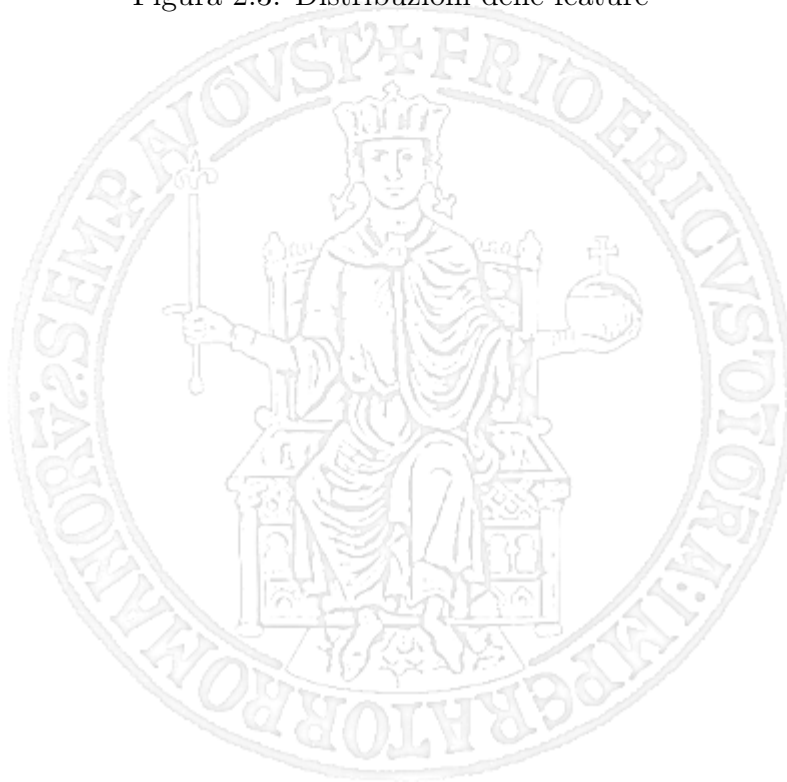


Figura 2.3: Distribuzioni delle feature



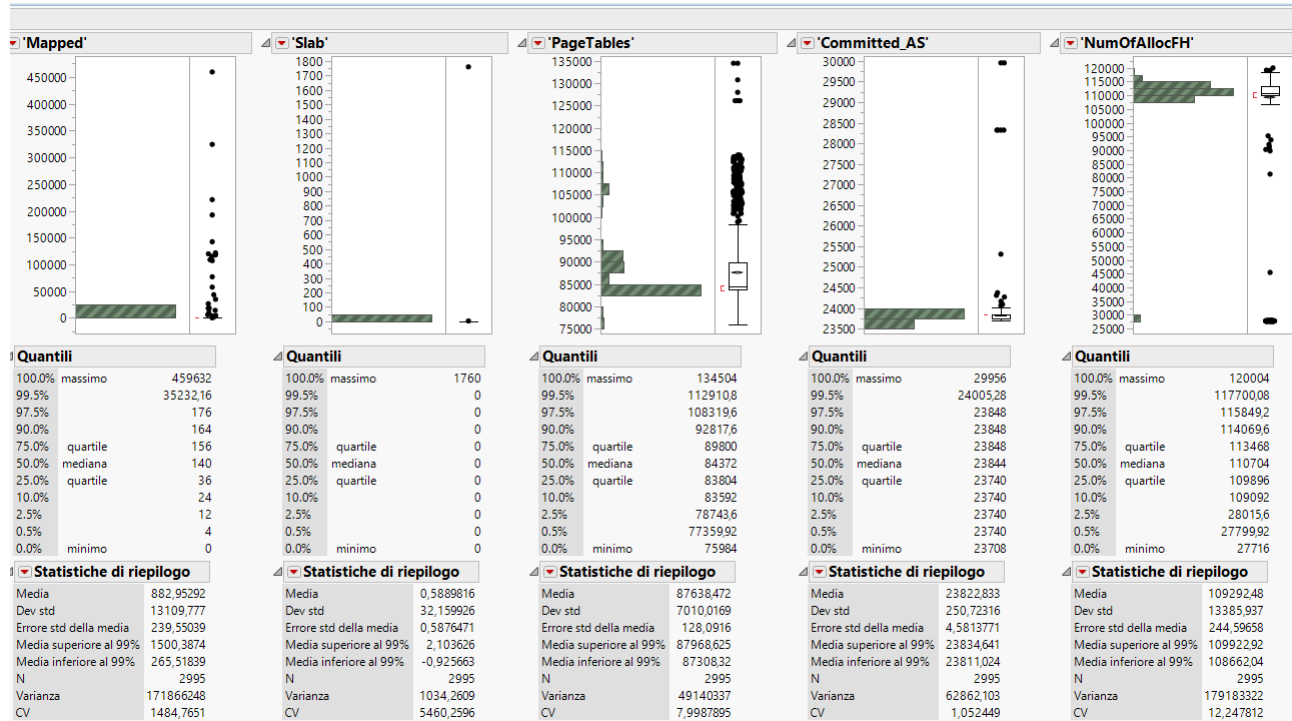


Figura 2.4: Distribuzioni delle feature

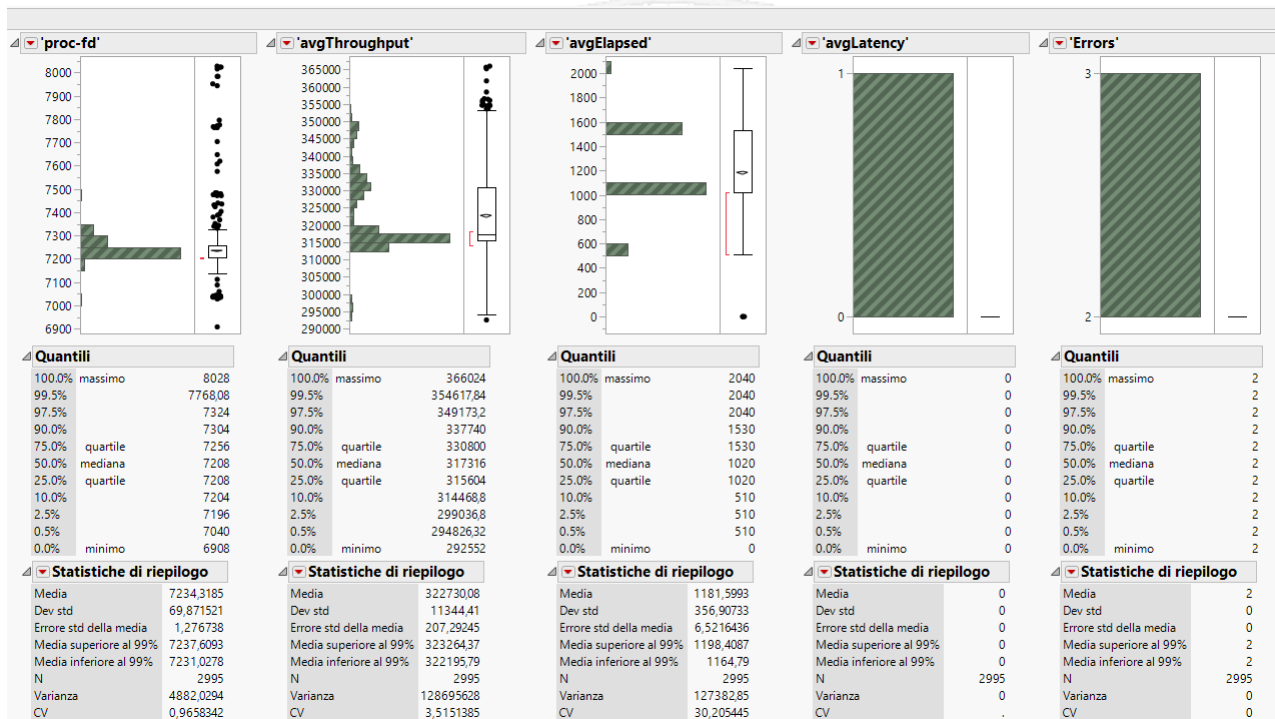


Figura 2.5: Distribuzioni delle feature

Come si può osservare dalle distribuzioni mostrate, gli attributi che presentano CV nullo sono: *active*, *anonpages*, *avglatency* ed *errors*.

Alle colonne già eliminate, si aggiunge la colonna *slab* che risulta essere non significativa ai fini dell'analisi in quanto, nonostante la varianza diversa da zero, contribuisce poco alla discriminazione dei punti rispetto alle altre feature.

Un'ulteriore scrematura delle feature si può effettuare osservando colonne che presentano stessi valori. Ad occhio, le colonne *memfree* e *writeback* risultano essere uguali. Per esserne certi, si è utilizzata la funzionalità del tool che permette di applicare delle formule sui dati delle colonne.

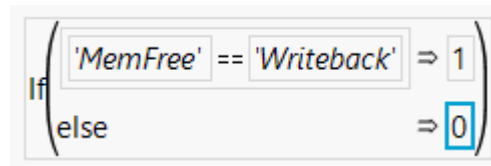


Figura 2.6: Formula confronto colonne

Del risultato prodotto, si è analizzata la distribuzione verificando che questa abbia coefficiente di variazione nullo.

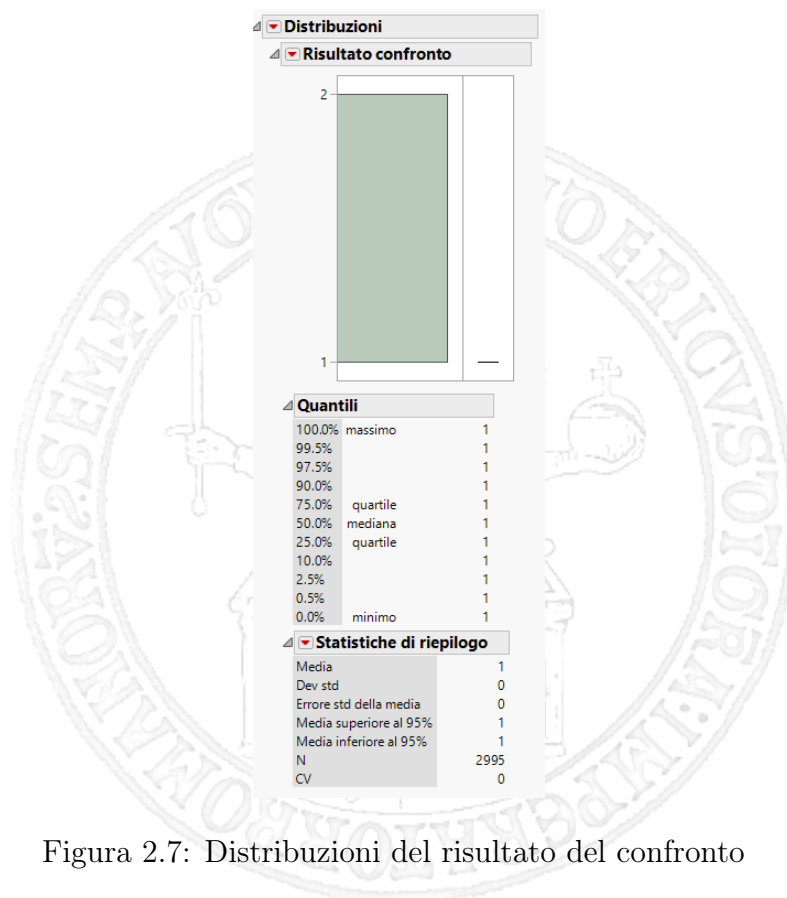


Figura 2.7: Distribuzioni del risultato del confronto

Il risultato di questa fase è una selezione di 18 feature a partire dalle 24 iniziali.

2.2.2 Principal Component Analysis

A questo punto si è proceduto ad effettuare un'analisi delle componenti principali, allo scopo di ridurre ulteriormente il numero di feature conservando la maggior parte della varianza. Ci si pone come obiettivo di conservare almeno il 95% della varianza totale.

Il risultato può essere osservato in forma grafica tramite uno score plot e loading plot.

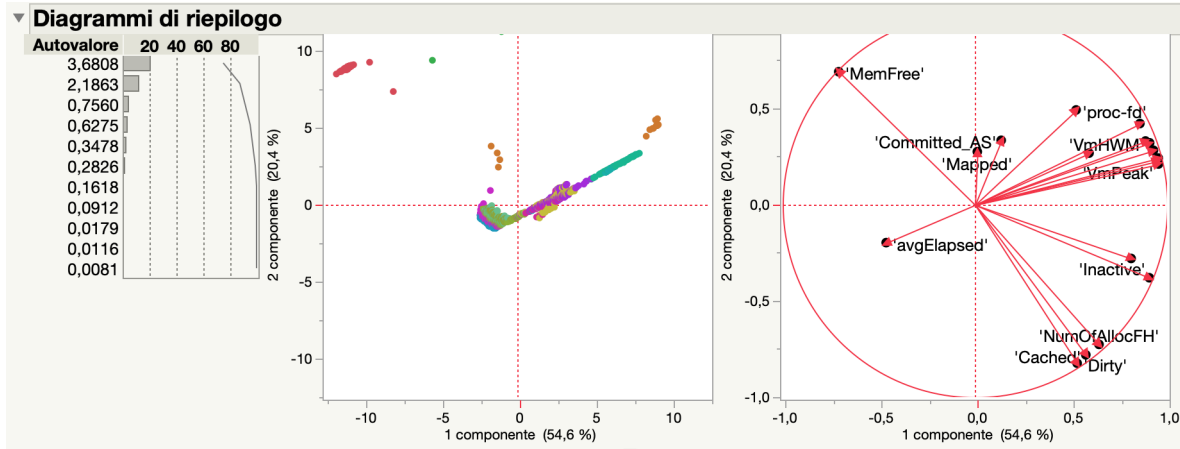


Figura 2.8: Score plot e loading plot generate dal tool

Il tool genera una vista degli autovalori della matrice di correlazione con la relativa percentuale di varianza per ogni componente ottenuta.

Autovalori					Percentuale cumulativa
Numero	Autovalore	Percentuale	20	40	
1	9,8192	54,551			54,551
2	3,6808	20,449			75,000
3	2,1863	12,146			87,146
4	0,7560	4,200			91,346
5	0,6275	3,486			94,832
6	0,3478	1,932			96,764
7	0,2826	1,570			98,334
8	0,1618	0,899			99,233
9	0,0912	0,507			99,740
10	0,0179	0,100			99,839
11	0,0116	0,065			99,904
12	0,0081	0,045			99,949
13	0,0062	0,035			99,983
14	0,0014	0,008			99,991
15	0,0009	0,005			99,996
16	0,0007	0,004			100,000
17	0,0000	0,000			100,000
18	0,0000	0,000			100,000

Figura 2.9: Autovalori della matrice di correlazione

Per rispondere all'obiettivo, si sceglie un numero di componenti principali pari a 6, rappresentativi del 96.764% della varianza totale.

2.2.3 Clustering

A valle della PCA effettuata, si vuole ridurre ulteriormente il dataset, con la differenza di voler diminuire il numero di istanze. A tale scopo si utilizza la tecnica del clustering di tipo gerarchico sulle componenti principali individuate, tramite la quale, scegliendo come metrica la distanza di Ward, si vuole individuare un trade-off tra la necessità di conservare una buona percentuale di varianza e quella di avere un numero accettabile di cluster.

Il risultato della fase di clustering è apprezzabile tramite il dendrogramma, prodotto dal tool.

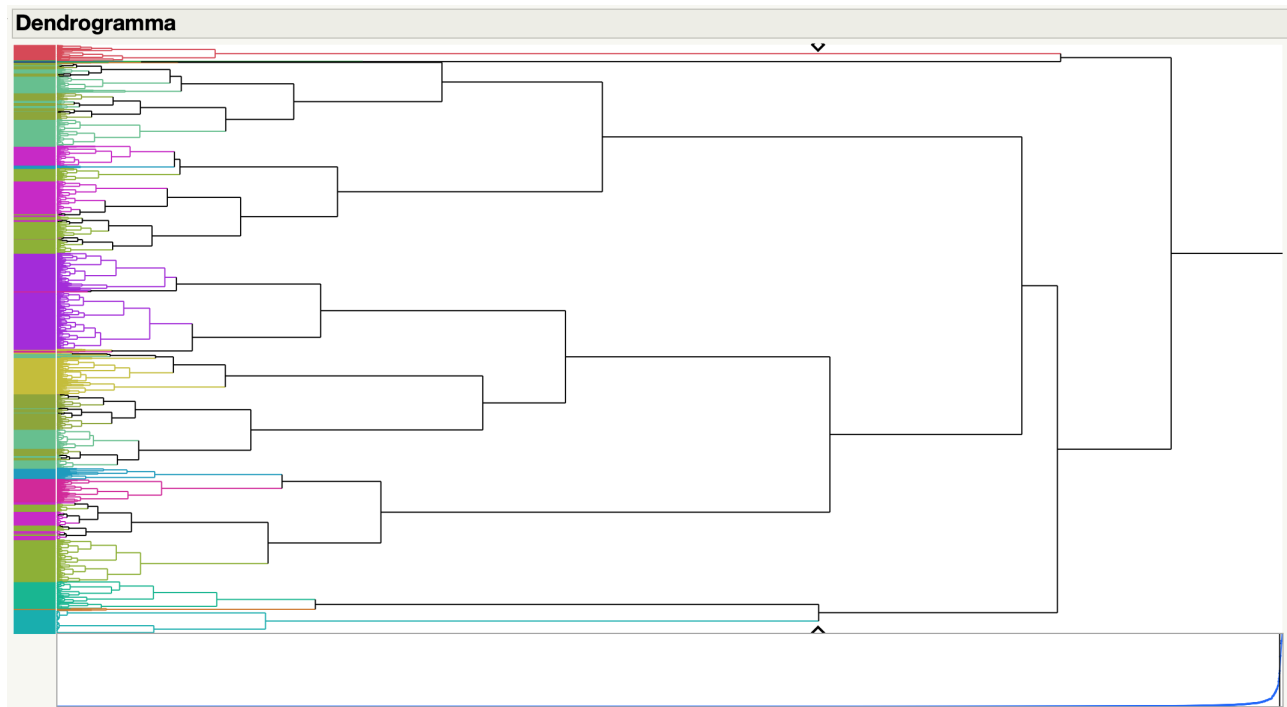


Figura 2.10: Dendrogramma

Inoltre il tool produce anche una tabella contenente, per ogni partizione, il relativo valore della distanza di Ward.

Numero di cluster	Distanza
42	4,53727383
41	4,64824107
40	4,71544410
39	4,73740514
38	4,78214101
37	4,82168691
36	5,11585070
35	5,27519441
34	5,39485020
33	5,53006180
32	5,69106202
31	5,74292374
30	5,83968358
29	5,93161593
28	6,01314292
27	6,07815214
26	6,61466470
25	7,73131312
24	7,81110869
23	8,22772672
22	8,25055421
21	8,97756907
20	9,47509536
19	10,19087495
18	10,31583926
17	10,99907676
16	11,57187362
15	12,62391905
14	12,88568385
13	13,70796307
12	14,94532692
11	15,82795378
10	18,80875084
9	20,80268710
8	24,84198796
7	26,64132735
6	37,20575450
5	37,75534548
4	47,12952556
3	48,86654204
2	49,02494378
1	54,42090668

Figura 2.11: Cronologia di clusterizzazione

Al fine di scegliere il numero di cluster, non avendo informazioni sulla realtà sperimentale, né relative al budget disponibile per condurre gli esperimenti, si è optato per una consistente riduzione del dataset ad un numero di cluster pari a 7; tale valore è stato ottenuto andando ad analizzare i valori delle distanze fornite da JMP (mostrate nella figura 11) e notando che la differenza maggiore tra tali valori si riscontrava nel passaggio da 6 a 7 cluster.

2.3 Calcolo analitico della varianza persa

A questo punto si è proceduto al calcolo preciso della varianza persa a seguito prima della fase di PCA e poi di clusterizzazione. A tal fine, è stato utilizzato il tool di data analysis Knime.

2.3.1 Devianza post-PCA

Per calcolare il valore di tale devianza sono stati dapprima ottenuti i valori di varianza per ogni componente principale, e successivamente sommati tra di loro. Il valore risultante, che sappiamo spiegare il 96.764% della varianza totale, è di **52148.055**.

2.3.2 Devianza post-clustering

La devianza post-clustering può essere espressa tramite la seguente somma:

$$\text{DEVIANZA TOTALE} = \text{DEVIANZA INTRA-CLUSTER} + \text{DEVIANZA INTER-CLUSTER}$$

Per calcolare la devianza intra-cluster, sono state isolate le istanze in base al cluster di appartenenza. Per ogni cluster si è proceduto nel seguente modo:

- è stata calcolata la varianza per ogni componente principale (ogni colonna);
- tali valori di varianza sono stati moltiplicati poi per $N-1$ (con N cardinalità del cluster) per ottenere la devianza;
- i valori di devianza ottenuti sono stati poi sommati per ottenere la devianza complessiva *intra-cluster*.

Per calcolare la devianza inter-cluster, si è dapprima ottenuta da JMP la tabella contenente i centroidi dei cluster individuati; poi:

- di ogni colonna, relativa alle componenti principali di tali punti, rappresentativi dei diversi cluster, è stata ottenuta la varianza;
- tali valori di varianza sono stati poi sommati e moltiplicati per $N-1$ (con N numero di cluster) per ottenere la devianza *inter-cluster*.

Una volta ottenuti tali valori è stata ottenuta la devianza totale:

$$Dev_{TOT} = Dev_{intra} + Dev_{inter} = 8715.526682054657 + 968.24685619697 \simeq 9683.78$$

2.3.3 Conclusione

Con i valori di devianza ottenuti è possibile a questo punto calcolare la percentuale di varianza persa a valle della PCA e del clustering.

$$\text{VARIANZA PERSA} = \text{DEVIANZA POST CLUSTERING} / \text{DEVIANZA POST PCA}$$

$$Var_{persa}(\%) = \frac{9683.78}{52148.055} \cdot 100 = 18.57\%$$

Quindi, del 96.764% di varianza spiegata dalla PCA, il clustering ne fa perdere il 18.57% (ne spiega quindi l'81.43%).

2.4 Costruzione del workload sintetico

Un possibile workload a questo punto può essere ottenuto con un random sampling dai cluster individuati.

'VmPeak'	'VmSize'	'VmHWM'	'VmRSS'	'VmPTE'	'Threads'	'MemFree'	'Buffers'	'Cached'	'Active'	'Inactive'	'Dirty'	'Writeback'	'AnonPages'	'Mapped'	'Slab'
152272	150216	25172	25108	160	55	5607348	30900	334636	0	141312	301612	5607348	8159224	52	0
170344	170340	31144	31140	200	43	4471580	94884	1259568	0	597900	891332	4471580	8159224	459632	0
170344	170340	31920	31912	200	25	4668808	98624	1117028	0	407620	892240	4668808	8159224	26764	0
178608	174512	36808	36160	208	65	4566668	192424	1117940	0	473252	925612	4566668	8159224	144	0
172392	170340	32696	31340	200	9	4602368	164352	1117452	0	432948	932540	4602368	8159224	28	0
215352	211256	61784	58896	312	161	4529564	204232	1118440	0	525496	908416	4529564	8159224	24	0
215352	206136	61784	53204	312	9	4539096	204664	1118496	0	521232	907432	4539096	8159224	36	0

Figura 2.12: Esempio di workload sintetico

'PageTables'	'Committed_AS'	'NumOfAllocFH'	'proc-fd'	'avgThroughput'	'avgElapsed'	'avgLatency'	'Errors'	Cluster
77364	23772	27764	7196	294240	1530	0	2	1
134504	29956	116216	8024	365692	1530	0	2	2
84208	23740	109296	7208	316832	1530	0	2	3
88500	23844	112872	7212	319412	1020	0	2	4
83740	23844	110316	7204	317048	1530	0	2	5
111240	23848	112956	7320	351388	510	0	2	6
105548	23848	110164	7320	345176	1020	0	2	7

Figura 2.13: Esempio di workload sintetico



Capitolo 3

Capacity Test



Capitolo 4

Dependability

4.1 Esercizio 1

4.1.1 Traccia

Trovare la $R(t)$ e l'MTTF per il sistema di cui viene fornito l'RBD. Nel calcolo dell'MTTF, assumere che tutti i componenti siano identici e falliscano randomicamente con failure rate λ .

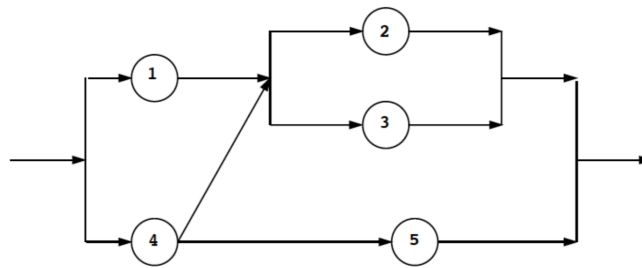


Figura 4.1: RBD

4.1.2 Soluzione

Da un'analisi del diagramma fornito, è possibile notare che i componenti 2 e 3 sono disposti in parallelo; possono essere dunque ridotti ad un unico blocco con reliability $R_{2||3} = 1 - (1 - R_2)(1 - R_3)$.

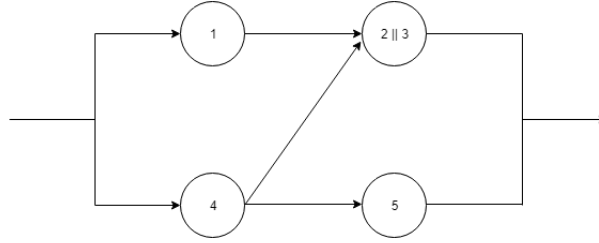


Figura 4.2: Sistema ridotto

Si osservi che non è possibile effettuare ulteriori riduzioni, per cui il diagramma che ne risulta è di tipo non-serie-parallelo. Per questi tipi di diagrammi, effettuiamo l'analisi dei success path; sappiamo infatti che la reliability del sistema risulta essere minore o uguale di quella del parallelo dei success path, che mostriamo di seguito:

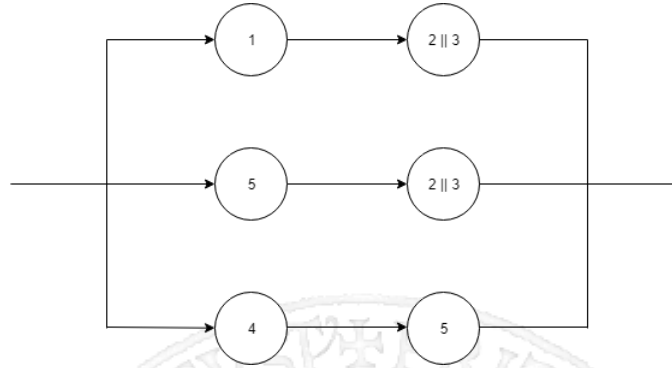


Figura 4.3: Success Path

Il limite superiore che individuiamo è il seguente:

$$R_{SYS} \leq 1 - (1 - R_1 R_{2||3})(1 - R_4 R_{2||3})(1 - R_4 R_5)$$

Per avere un valore preciso di Reliability, facciamo ricorso alla tecnica del conditioning, condizionando appunto il funzionamento del sistema a quello di un nodo. In particolare, si è scelto il nodo 4, in quanto si può notare che nel caso in cui esso funzioni (considerandolo dunque come corto circuito) o meno (considerandolo circuito aperto), è possibile effettuare una riduzione del sistema rispettivamente ad un parallelo ed una serie, che sono di più semplice trattazione.

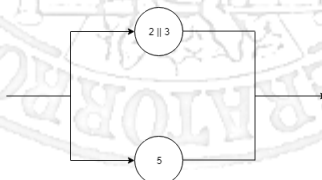


Figura 4.4: Caso in cui 4 funzioni

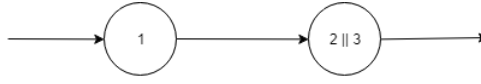


Figura 4.5: Caso in cui 4 non funziona

Calcoliamo dunque le probabilità condizionate:

$$P(\text{system works} \mid 4 \text{ works}) = 1 - (1 - R_{2||3})(1 - R_5)$$

$$P(\text{system works} \mid 4 \text{ doesn't work}) = R_1 \cdot R_{2||3}$$

Sfruttando il teorema della probabilità totale, la reliability totale del sistema sarà data dalla somma della prima probabilità, moltiplicata per la reliability del componente 4, e la seconda, moltiplicata per l'unreliability del componente 4.

$$R_{SYS} = (1 - (1 - R_{2||3})(1 - R_5)) \cdot R_4 + (R_1 \cdot R_{2||3}) \cdot (1 - R_4)$$

Da traccia, tutti i nodi hanno la stessa reliability; supponiamo sia R .

$$R_{2||3} = 1 - (1 - R_2)(1 - R_3) = 1 - (1 - R)(1 - R) = 1 - (1 - R)^2 = -R^2 + 2R$$

$$R_{SYS} = (1 - (1 - (2R - R^2))(1 - R)) \cdot R + (R \cdot (2R - R^2))(1 - R) = 2R^4 - 6R^3 + 5R^2$$

Ora, sfruttando nuovamente l'ipotesi che tutti i componenti del sistema siano identici, ed assumendo che i fallimenti seguano andamento esponenziale con failure rate pari a λ ($R(t) = e^{-\lambda t}$) possiamo calcolare:

$$MTTF_{SYS} = \int_0^{\infty} [2R^4 - 6R^3 + 5R^2] dt = \int_0^{\infty} [2e^{-4\lambda t} - 6e^{-3\lambda t} + 5e^{-2\lambda t}] dt = \frac{1}{2\lambda} - \frac{2}{\lambda} + \frac{5}{2\lambda} = \frac{1}{\lambda}$$

4.2 Esercizio 2

4.2.1 Traccia

Vogliamo confrontare due diversi modi di utilizzare la ridondanza per incrementare la reliability di un sistema. Supponiamo che il sistema abbia bisogno di s componenti identici in serie per funzionare in maniera appropriata. Supponiamo inoltre che ci vengano dati $m \times s$ componenti. Tra i due schemi, quale fornisce una maggiore reliability? Supposta r la reliability del singolo componente, derivare le espressioni delle reliability delle due configurazioni. Comparare le espressioni per $m=3$ e $s=3$.

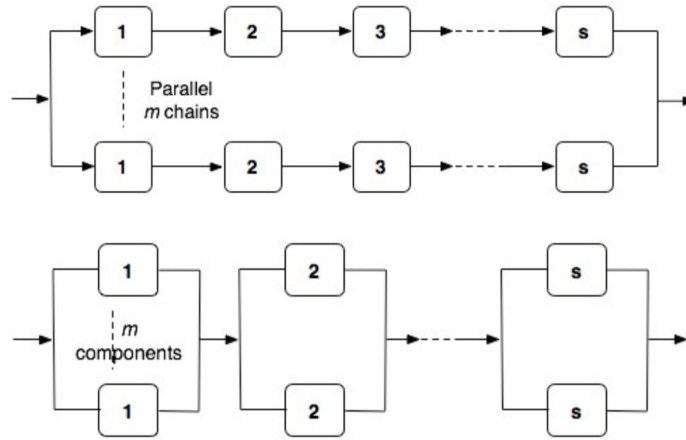


Figura 4.6: Schemi da comparare

4.2.2 Soluzione

L'obiettivo è quello di ricavare le espressioni per i due differenti schemi. Si può notare che il primo è costituito da serie di s componenti disposte su m linee collegate in parallelo. Abbiamo dunque un parallelo di serie, la cui espressione della reliability è la seguente:

$$R_{PoS} = 1 - (1 - r^s)^m$$

Il secondo, invece, è composto da s "blocchi" in serie, ognuno dei quali costituito dal parallelo di m componenti. Ci troviamo quindi di fronte ad un sistema serie di paralleli, la cui espressione della reliability è la seguente:

$$R_{SoP} = (1 - (1 - r)^m)^s$$

Andando a settare $s=m=3$, si sono plottate le due funzioni con l'aiuto di MATLAB; il risultato è il seguente.

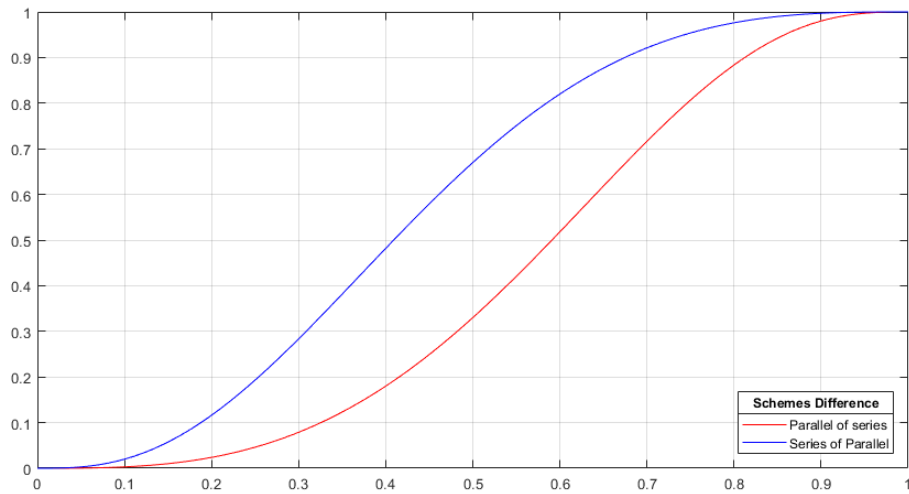


Figura 4.7: Confronto tra i due schemi

È possibile notare che la reliability del secondo schema è migliore.

4.3 Esercizio 3

4.3.1 Traccia

In figura è mostrata l'architettura di una rete di computer in un sistema bancario. L'architettura è chiamata “skip-ring network” and è progettata per permettere ai processori di comunicare anche dopo l'avvenimento di un failure in un nodo. Ad esempio, se il nodo 1 fallisce, il nodo 8 può bypassare il nodo fallito instradando i dati sul link alternativo che collega il nodo 8 con il 2. Assumendo che i link siano perfetti e i nodi abbiano ognuno una reliability R_m , derivare l'espressione per la reliability della rete. Se R_m segue la legge di fallimento esponenziale e il failure rate di ogni nodo è di 0.005 failure all'ora, determinare la reliability del sistema alla fine di un periodo di 48 ore.

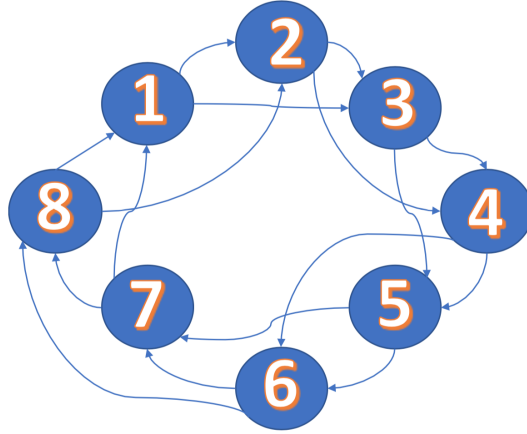


Figura 4.8: Rete del sistema bancario

4.3.2 Soluzione

Per il tipo di rete in esame, ci chiediamo quando il sistema può dichiararsi fallito. La particolarità di tale rete è che un nodo fallito può essere bypassato: il nodo raggiunto tramite il link alternativo dovrà necessariamente funzionare, pena il fallimento dell'intero sistema. Si può quindi dedurre che i fallimenti non devono riguardare nodi adiacenti. Fatta questa assunzione, quanti fallimenti possono avvenire contemporaneamente? Dati 8 nodi, il sistema continua a funzionare con al massimo 4 fallimenti, limite oltre il quale essi saranno per forza di cose adiacenti.

Lo schema è dunque un M-out-of-N System, in particolare un 4-out-of-8. La reliability in questo caso può essere espressa nel seguente modo:

$$R_{48} = \sum_{i=0}^4 \binom{8}{i} R_m^{8-i} (1 - R_m)^i$$

Dal coefficiente binomiale presente nella sommatoria, bisogna però escludere tutte le combinazioni non ammissibili di nodi.

$$R_{48} = \sum_{i=0}^4 ((\binom{8}{i}) - e_i) R_m^{8-i} (1 - R_m)^i$$

Dove e_i rappresenta il numero di configurazioni da escludere.

i	e_i	Nota
0	0	Nessun nodo fallisce
1	0	Tutte le configurazioni sono ammissibili in quanto non esistono nodi adiacenti che falliscono
2	8	Bisogna escludere tutte le coppie di nodi adiacenti che falliscono, che sono 8
3	40	Bisogna escludere tutte le configurazioni in cui falliscono 3 nodi, di cui 2 adiacenti più un terzo che ruota ($6 \cdot 8 = 48$ configurazioni). Da queste bisogna escludere però le ripetizioni di coppie di nodi che hanno un nodo in comune (8)
4	68	Le uniche due configurazioni ammissibili sono 1-3-5-7 e 2-4-6-8

Tabella 4.1: Configurazioni

Una volta fatto ciò è possibile calcolare R_{48}

$$R_{48} = R_m^8 + 8 \cdot R_m^7 \cdot (1 - R_m) + (28 - 8) \cdot R_m^6 \cdot (1 - R_m)^2 + (56 - 40) \cdot R_m^5 \cdot (1 - R_m)^3 + (70 - 68) \cdot R_m^4 \cdot (1 - R_m)^4$$

Tenendo conto che R_m segue una legge di fallimento esponenziale tale che: $R_m = e^{-\lambda t}$, con $\lambda = 0.005$, abbiamo $R_m(48) \simeq 0.7866$, e la reliability del sistema in un periodo di 48h risulterà pari a:

$$R_{48} \simeq 0.728822$$

4.4 Esercizio 4

4.4.1 Traccia

Un'applicazione richiede in un sistema multiprocessore almeno tre processore debbano essere disponibili con probabilità maggiore del 99%. Il costo di un processore con una reliability dell'80% è di 1000\$, e ogni incremento del 10% di reliability costerà 500\$. Determinare il numero di processori (n) e la reliability (p) di ogni processore (assumendo che i processori abbiano la stessa reliability) che minimizzano il costo totale del sistema.

4.4.2 Soluzione

Dall'analisi del problema, si deduce che il sistema descritto è di tipo m-out-of-n. In particolare, affinché il sistema funzioni, è necessario il funzionamento di almeno 3 processori. L'obiettivo è di trovare il numero di processori per la configurazione a costo minimo con reliability maggiore di 0.99.

La reliability in questo tipo di sistema si calcola nel seguente modo:

$$R_{MN} = \sum_{i=0}^{N-M} \binom{N}{i} R_m^{N-i} (1 - R_m)^i$$

Con R_m reliability del singolo processore, uguale per tutti i processori.

Al fine di trovare la configurazione a costo minimo, è stato individuato il costo di ogni processore al crescere della reliability.

Reliability	Costo (\$)
0.8	1000
0.9	1500
1.0	2000

Tabella 4.2: Costi dei processori

Per ogni configurazione possibile sono stati valutati, dato il numero minimo di processori funzionanti (M), l'affidabilità e il costo delle varie configurazioni al variare del numero dei processori del sistema (N). Di queste configurazioni, tra quelle che presentano reliability maggiore del 99%, si sceglie quella a costo minimo. A tale scopo è stato realizzato il seguente script MATLAB.

```

1 function [R,C,c_min,r_min] = reliabilityFunction(M,N)
2 R = zeros;
3 C = zeros;
4 p=[0.80, 0.88, 0.96];
5 for j=1:3
6     for n=3:1:N
7         acc=0;
8         for i=0:n-M
9             acc = acc + nchoosek(n,i)*(p(j)^(n-i))*((1-p(j))^i);
10        end
11        R(n-2,j) = acc;
12        if(abs(p(j)-0.80)<eps)
13            C(n-2,j) = 1000*n;
14        end
15        if(abs(p(j)-0.88)<eps)
16            C(n-2,j) = 1500*n;
17        end
18        if(abs(p(j)-0.96)<eps)
19            C(n-2,j) = 2000*n;
20        end
21    end
22 end
23 c_min = min(C(R>0.99));
24 r_min = R(C==min(C(R>0.99)));
25 end

```

Codice Componente 4.1: Calcolo reliability e costi

I risultati sono riportati di seguito.

#processori	Reliability singolo processore		
	0.8	0.88	0.96
3	0.51200002	0.68147200	0.88473600
4	0.81919998	0.92680192	0.99090433
5	0.94208002	0.98568112	0.99939781
6	0.98303998	0.99745691	0.99996400
7	0.99532801	0.99957657	0.99999797
8	0.99876863	0.99993271	0.99999988
9	0.99968612	0.99998969	1
10	0.99992210	0.99999845	1

Tabella 4.3: Risultati di reliability

#processori	Costi delle configurazioni		
	0.8	0.88	0.96
3	3000	4500	6000
4	4000	6000	8000
5	5000	7500	10000
6	6000	9000	12000
7	7000	10500	14000
8	8000	12000	16000
9	9000	13500	18000
10	10000	15000	20000

Tabella 4.4: Risultati costi

La soluzione che garantisce il costo minimo e affidabilità maggiore del 99% è quella con numero di processori pari a 7, ognuno con reliability di 0.8. Il sistema complessivo presenta reliability pari a 0.99532801 con costo pari a 7000\$.

4.5 Esercizio 5

4.5.1 Traccia

Il sistema mostrato in figura è un sistema di elaborazione per un elicottero. Il sistema ha una ridondanza duale sia per i processori che per le unità di interfacciamento. Vengono utilizzati due bus, ed ogni bus è replicato. La parte interessante del sistema è l'equipaggiamento di navigazione. Il velivolo può essere completamente guidato usato l'“Inertial Navigation System” (INS). Se tale INS

fallisce, il velivolo può essere guidato attraverso una combinazione di Doppler e AHRS (Altitude Heading Reference System). Di quest'ultima unità, ne sono presenti 3, delle quali una sola è necessaria. I dati dal Doppler e un AHRS possono essere usati in sostituzione del componente INS se esso fallisce. A causa di altri sensori e strumentazioni, sono richiesti entrambi i bus affinché il sistema funzioni in maniera appropriata, indipendentemente dal sistema di navigazione utilizzato.

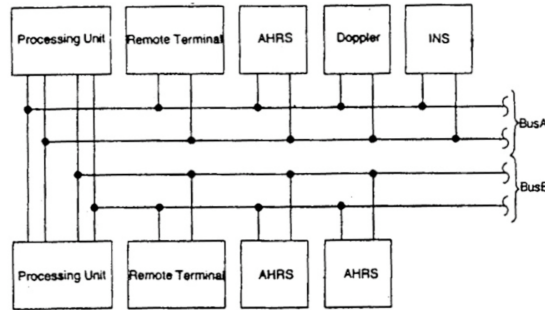


Figura 4.9: Architettura del sistema

1. Disegnare l'RBD del sistema
2. Disegnare il Fault Tree del sistema ed analizzare i minimal cutset
3. Calcolare la reliability per un'ora di volo utilizzando i valori di MTTF in tabella. Assumere che venga applicata la legge di fallimento esponenziale e che la fault coverage sia perfetta

Equipment	MTTF (hr)
Processing Unit	5000
Remote Terminal	2500
AHRS	1000
INS	1000
Doppler	300
Bus	10000

Tabella 4.5: MTTF dei componenti

4. Ripetere il punto precedente, ma stavolta incorporare un fattore di coverage per la fault detection e riconfigurazione delle unità di elaborazione. Usando gli stessi dati di fallimento, determinare il valore approssimativo di fault coverage richiesto per ottenere (alla fine dell'ora) una reliability di 0.99999

4.5.2 Soluzione

4.5.2.1 Punto 1

Dalla descrizione fornita, è possibile dedurre come ciascuna unità funzionale sia necessaria al corretto funzionamento del sistema. L'RBD sarà pertanto composto da una serie di componenti. In particolare, in corrispondenza di componenti replicati, all'interno della serie essi saranno disposti in parallelo tra di loro. Unica eccezione è fatta per il sistema di navigazione, in cui è necessario il funzionamento dell'INS, o in alternativa del Doppler in coppia con un AHRS.

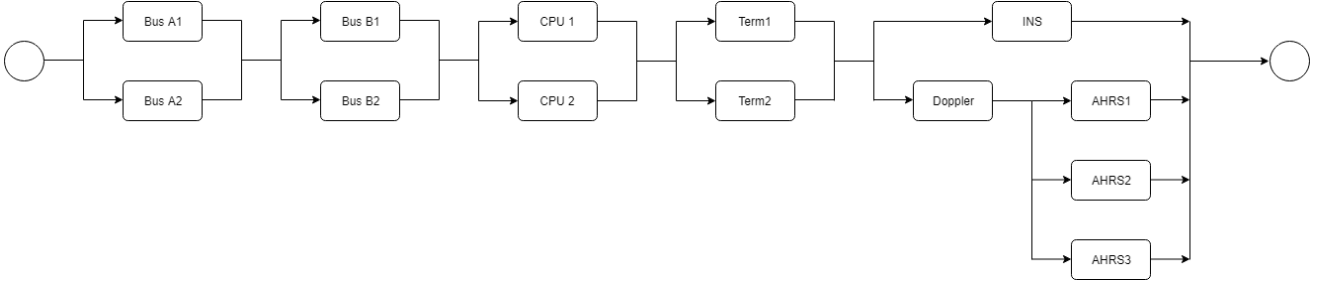


Figura 4.10: RBD del sistema

4.5.2.2 Punto 2

Per costruire il fault tree bisogna considerare i fallimenti del sistema. Si osserva dunque che il sistema fallisce in presenza di una delle seguenti condizioni:

- entrambi i bus di tipo A falliscono;
- entrambi i bus di tipo B falliscono;
- entrambe le CPU falliscono;
- entrambi i terminali falliscono;
- entrambi i sistemi di navigazione falliscono: ciò implica il fallimento dell'INS e o del Doppler o di tutti e 3 gli AHRS.

Date queste condizioni, è possibile ricavare la formula del fallimento:

$$(BusA1 \wedge BusA2) \vee (BusB1 \wedge BusB2) \vee (CPU1 \wedge CPU2) \vee (Term1 \wedge Term2) \vee (INS \wedge (Doppler \vee (AHRs1 \wedge AHRs2 \wedge AHRs3)))$$

Da questa espressione booleana è possibile ricavare il fault tree.

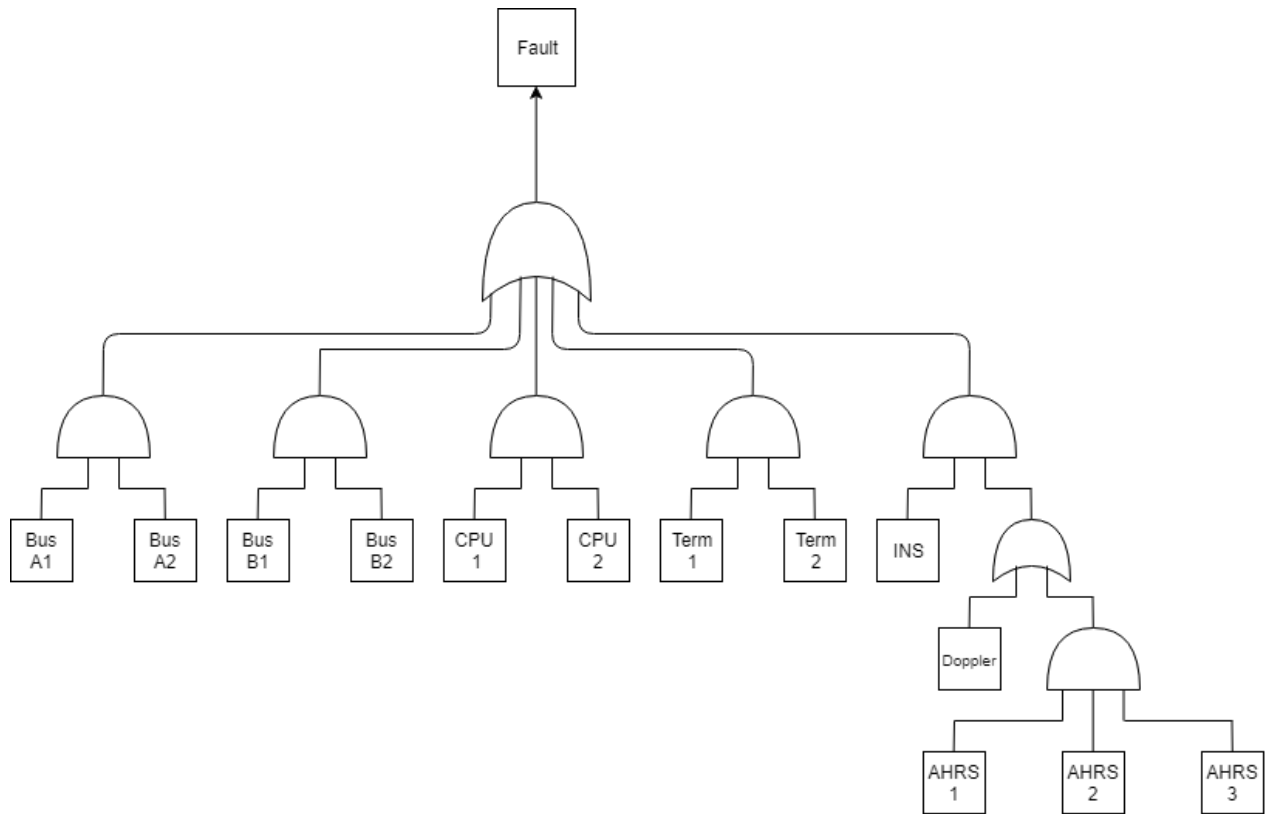


Figura 4.11: Fault Tree del sistema

Analizzando la figura si individuano i seguenti minimal cutset:

- {Bus A1, Bus A2}
- {Bus B1, Bus B2}
- {CPU 1, CPU 2}
- {Term 1, Term 2}
- {INS, Doppler}
- {INS, AHRS 1, AHRS 2, AHRS 3}

0.0.2.3 Punto 3

Si va adesso ad analizzare la reliability del sistema dopo un'ora di volo. È possibile calcolare tale valore a partire dalle reliability dei singoli componenti; si inizia quindi a trovare il parametro λ relativo ad ogni componente a partire dalla tabella con i diversi MTTF.

Equipement	MTTF (hr)	λ
Processing Unit	5000	0.0002
Remote Terminal	2500	0.0004
AHRS	1000	0.001
INS	1000	0.001
Doppler	300	0.003
Bus	10000	0.0001

Tabella 4.6: MTTF dei componenti

A questo punto è possibile esprimere le diverse reliability dei componenti del sistema come:

$$R_{BUSAEQ} = 1 - (1 - R_{BUS})^2$$

$$R_{BUSBEQ} = 1 - (1 - R_{BUS})^2$$

Le quali rappresentano le reliability dei due sistemi formati dai paralleli dei Bus A e B.

$$R_{CPUEQ} = 1 - (1 - R_{CPU})^2$$

$$R_{TermEQ} = 1 - (1 - R_{BUS})^2$$

$$R_{AHRSEQ} = 1 - (1 - R_{AHRS})^3$$

Rappresentano gli altri sottosistemi formati dalla replicazioni di componenti uguali in parallelo (CPU, Terminal, AHRS).

$$R_{ALT} = R_{Doppler} R_{AHRSEQ}$$

$$R_{CS} = 1 - (1 - R_{INS})(1 - R_{ALT})$$

Le precedenti due reliability rappresentano la serie tra il Doppler e il parallelo dei tre componenti AHRS, quindi il sistema di controllo alternativo, e il parallelo di tale sistema con il sistema INS, in maniera tale da ottenere la reliability totale del sottosistema dedicato al controllo. A questo punto è possibile calcolare la reliability totale del sistema tramite la serie dei diversi sottosistemi sopra analizzati:

$$R_{sys} = R_{BUSAEQ} R_{BUSBEQ} R_{CPUEQ} R_{TermEQ} R_{CS}$$

Il risultato è stato ottenuto tramite il seguente script MATLAB:

```

1 format long
2
3 ycpu = 0.0002;
4 yterm = 0.0004;
5 yAHRS = 0.001;
6 yins = 0.001;
7 ydoppler = 0.003;
8 ybus = 0.0001;
9
10 t = 1;
11
12
13 Rbus = exp(-(ybus.*t));
14 Rcpu = exp(-(ycpu.*t));
15 Rterm = exp(-(yterm.*t));
16 R_AHRS = exp(-(yAHRS.*t));
17 Rdoppler = exp(-(ydoppler.*t));
18 Rins = exp(-(yins.*t));
19
20 RbusA = 1-(1-Rbus).^2;
21 RbusB = 1-(1-Rbus).^2;
22 Rcpu_eq = 1-(1-Rcpu).^2;
23 Rterm_eq = 1-(1-Rterm).^2;
24 Ralt = Rdoppler.*(1-(1-R_AHRS).^3);
25 R_cs = 1-(1-Rins).*(1-Ralt);
26 Rsys = RbusA*RbusB*Rcpu_eq*Rterm_eq*R_cs;
27 Rsys

```

Codice Componente 4.2: Reliability totale del sistema

Si ottiene quindi il valore della reliability totale del sistema $R_{sys} = 0.999996786066414$

4.5.2.3 Punto 4

Si supponga a questo punto di introdurre un circuito di detection per la rilevazione del fallimento delle CPU. La probabilità di tale circuito di rilevare un fallimento è indicata con il parametro c :

$$R_{sys} = R_1 + c(1 - R_1)R_2$$

Quindi nel caso del sottosistema formato dalle due CPU in parallelo avremo che:

$$c = \frac{(R_{detection} - R_{CPU})}{R_{CPU}(1 - R_{CPU})}$$

Si consideri adesso la reliability del sistema non includendo le CPU:

$$R_{sysNOCPU} = R_{BUSAEQ}R_{BUSBEQ}R_{TermEQ}R_{CS}$$

A tal proposito, si vuole trovare la probabilità c che ci permetta di avere una reliability del sistema almeno pari a 0.99999; per farlo scriviamo la relazione imponendo tale vincolo:

$$0.99999 = R_{\text{detection}} R_{\text{sysNOCPU}}$$

$$R_{\text{detection}} = \frac{0.99999}{R_{\text{sysNOCPU}}}$$

Una volta trovato tale valore lo si sostituisce all'interno della formula che esprime il valore di c .

Il procedimento illustrato è stato implementato nel seguente script MATLAB:

```

1  format long
2
3  ycpu = 0.0002;
4  yterm = 0.0004;
5  yAHRs = 0.001;
6  yins = 0.001;
7  ydoppler = 0.003;
8  ybus = 0.0001;
9
10 t = 1;
11
12
13  Rbus = exp(-(ybus.*t));
14  Rcpu = exp(-(ycpu.*t));
15  Rterm = exp(-(yterm.*t));
16  R_AHRs = exp(-(yAHRs.*t));
17  Rdoppler = exp(-(ydoppler.*t));
18  Rins = exp(-(yins.*t));
19
20  RbusA = 1-(1-Rbus).^2;
21  RbusB = 1-(1-Rbus).^2;
22  Rcpu_eq = 1-(1-Rcpu).^2;
23  Rterm_eq = 1-(1-Rterm).^2;
24  Ralt = Rdoppler.*(1-(1-R_AHRs).^3);
25  R_cs = 1-(1-Rins).*(1-Ralt);
26
27  Rsys_noCPU = RbusA*RbusB*Rterm_eq*R_cs;
28  Rsys_noCPU
29
30  Rcpu_detection = 0.99999/Rsys_noCPU;
31  C = (Rcpu_detection - Rcpu)/(Rcpu*(1-Rcpu));
32  C

```

Codice Componente 4.3: Reliability totale del sistema

Il risultato è che il valore minimo per il parametro c deve essere: $c \simeq 0.96606$

Capitolo 5

FFDA

