

IMPIANTI DI ELABORAZIONE

tenuto da

Ing. Domenico Cotroneo
Ing. Pietro Liguori
Ing. Stefano Rosiello

Simone Dotolo M63001503

Marzo 2023



Dipartimento di Ingegneria Elettrica e delle Tecnologie
dell'Informazione
Università degli Studi di Napoli Federico II

Indice

1	Introduzione	1
2	Benchmark	2
2.1	Raccolta dei dati	2
2.2	Valutazione della dimensione campionaria	3
2.3	Test visivo	4
2.4	Test parametrico	5
3	Workload Characterization	7
3.1	Raccolta e Preprocessing dei dati	7
3.2	Principal Component Analysis	8
3.3	Clustering	8
3.4	Calcolo della devianza persa	10
3.5	Considerazioni finali	11
4	Web Server	12
4.1	Setup sperimentale	12
4.2	Capacity Test	13
4.2.1	Capacity Test e Performance Analysis	13
4.2.2	Bottleneck Performance	16
4.2.3	Fairness Index	17
4.2.4	Considerazioni finali	19
4.3	Workload Characterization	20
4.3.1	Creazione del Workload reale	20
4.3.2	Workload Characterization HL	21
4.3.3	Workload Characterization LL	22
4.3.4	Creazione Workload sintetico	24
4.3.5	Workload Characterization LL'	25
4.3.6	Confronto tra LL_c e LL'_c	27
4.4	Design of Experiment	31
4.4.1	Analisi dell'importanza	32
4.4.2	Analisi della significatività	33
5	Regressione	36
5.1	Dataset EXP1	36
5.2	Dataset EXP2	39
5.3	Dataset OS1	42
5.4	Dataset OS2	45
5.5	Dataset OS3	48
5.6	Dataset VmRes1	51
5.7	Dataset VmRes2	54
5.8	Dataset VmRes3	57
6	Reliability	60
6.1	Esercizio 1: Reliability Block Diagram	60
6.2	Esercizio 2: Confronto tra sistemi	62
6.3	Esercizio 3: Skip Ring Network	64

6.4	Esercizio 4: Confronto tra sistemi	66
6.4.1	Confronto 1	66
6.4.2	Confronto 2	67
6.4.3	Confronto 3	68
6.4.4	Confronto 4	69
6.5	Esercizio 5: Helicopter Processing System	70
7	Field Failure Data Analysis	72
7.1	Calcolatore Mercury	72
7.1.1	Logging Collection e Filtering	72
7.1.2	Data Manipulation	73
7.1.3	Data Analysis	74
7.2	Calcolatore Blue Gene/L	75
7.2.1	Logging Collection e Filtering	75
7.2.2	Data Manipulation	76
7.2.3	Data Analysis	77
7.3	Confronto tra Mercury e Blue Gene/L	79
7.4	Analisi finestra di coalescenza	79
7.4.1	Nodi Mercury	79
7.4.2	Categorie di errore Mercury	82
7.4.3	Nodi Blue Gene/L	85
7.5	Reliability del sistema vs Reliability dei nodi	88
7.5.1	Calcolatore Mercury	88
7.5.2	Calcolatore Blue Gene/L	90
7.6	Reliability Bottlenecks	93
7.6.1	Calcolatore Mercury	93
7.6.2	Calcolatore Blue Gene/L	94
7.7	Reliability di nodi simili	94
7.7.1	Calcolatore Mercury	94
7.7.2	Calcolatore Mercury	95
7.8	Relazione tra tipi di errore e nodi	96

1 INTRODUZIONE

Il seguente elaborato è una raccolta degli homework assegnati durante il corso di Impianti di Elaborazione A.A 2023/2024.

Gli homework realizzati sono:

1. Benchmark
2. Workload Characterization
3. Web Server
4. Regression
5. Reliability
6. FFDA

Per automatizzare il processo di raccolta ed analisi dei dati, sono stati sviluppati degli script in *Python*.
Tutto il materiale utilizzato per la realizzazione dell'elaborato è presente sulla repository GitHub.
Dove non specificato, per i test d'ipotesi, si considera $\alpha = 0.05$.



2 BENCHMARK

Si effettua il confronto tra due sistemi di elaborazione differenti utilizzando il **Benchmark *nbody***. L'obiettivo è quello di dimostrare che i tempi di esecuzione valutati durante l'esecuzione indipendente di *nbody* sui due sistemi, risultino statisticamente differenti al fine di diversificarli. Di seguito le configurazioni dei due sistemi:

Configurazione PC fisso	
CPU	i5-3470
Core	4
RAM	8 GB
S.O.	Ubuntu 22.04

Table 1: **Caratteristiche macchina fisica.**

Configurazione Laptop	
CPU	AMD Ryzen 5700u
Core	16
RAM	16 GB
S.O.	Ubuntu 22.04

Table 2: **Configurazione Laptop.**

2.1 RACCOLTA DEI DATI

La prima fase dell'analisi è la raccolta dei dati tramite delle operazioni di misurazione diretta. La condizione fondamentale da garantire è l'indipendenza tra le osservazioni raccolte, per questo, si procede allo spegnimento e alla riaccensione delle macchine per ogni misurazione diretta. Nello specifico si è effettuata tale operazione inizialmente per 10 volte e ogni volta è stato lanciato il seguente comando:

```
bash ./launch nbody.sh -r 5 -n ***
```

dove:

- **-r**: indica il numero di ripetizioni (5).
- **-n**: indica il numero di corpi da simulare (10000, 1000000).

Di seguito le 10 osservazioni preliminari raccolte sui due sistemi di elaborazione, per 10000 e 1000000 corpi simulati.

Osservazioni PC (ms)		Osservazioni Laptop (ms)	
10000 corpi	1000000 corpi	10000 corpi	1000000 corpi
3382.8	299180.8	2957.4	218973.4
3981.4	302526.8	3010.8	223506.6
4300.2	298758.4	2473.6	217941.8
4486.2	303138.8	2378.8	221713.2
4377.2	302412.8	2150	222334
4247.4	302406	2899	218650.6
4502.8	303688	2725.8	217433.8
4692.6	301799.4	2822.6	218238.6
4306	300727.8	2807.6	222302.6
4103.4	304562	2644.6	224741

Table 3: **10 osservazioni.**

Successivamente sono state analizzate le osservazioni raccolte per verificare se provengono da una distribuzione normale. Sono stati realizzati dunque dei plot quantile-quantile ed il test di *Shapiro-Wilk* per ogni campione.

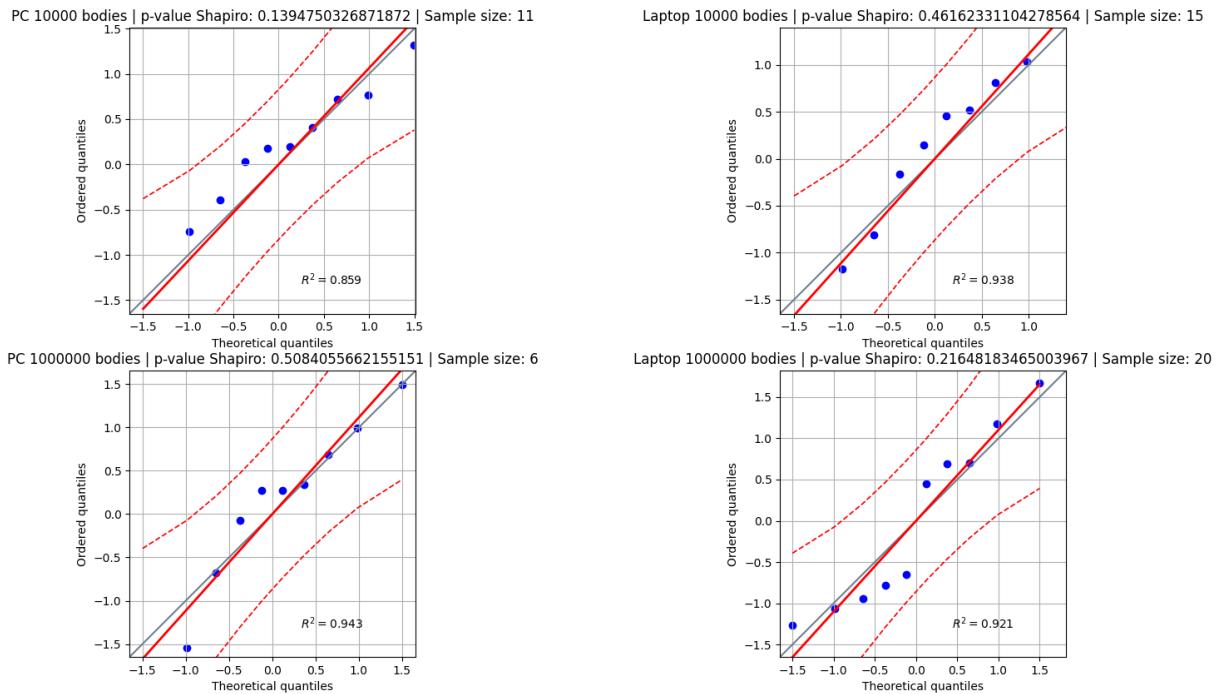


Figure 1: **Plot quantile-quantile per le 10 osservazioni.**

Sia dal test visivo che dal test non parametrico, è possibile concludere che le osservazioni provengono da una distribuzione normale.

2.2 VALUTAZIONE DELLA DIMENSIONE CAMPIONARIA

Si procede dunque al calcolo della dimensione campionaria. Dato un errore E ed un intervallo di confidenza del $(100 - \alpha)\%$:

$$n = \left(\frac{z_{\alpha/2} * \sigma}{E} \right)^2 \quad (1)$$

Nel nostro caso, non avendo a disposizione la deviazione standard σ della popolazione, utilizzeremo la deviazione standard s dei campioni, moltiplicata per 2, avendo raccolto poche osservazioni. Utilizzando un intervallo di confidenza del 95%, un errore del 10% per 10000 corpi ed un errore dell'1% per 1000000 corpi otteniamo le seguenti dimensioni campionarie.

PC (Dimensione Campionaria)	Laptop (Dimensione Campionaria)
10000 corpi	1000000 corpi
11	6
15	20

Table 4: Dimensioni campionarie.

2.3 TEST VISIVO

Per verificare che i campioni delle rispettive macchine provengono dalla stessa o diversa popolazione ci si avvale inizialmente di un test visivo. Si è provveduto al calcolo degli intervalli di confidenza della media μ considerando anche in questo caso un livello di confidenza del 95%, utilizzando la distribuzione t ($n < 30$):

$$\bar{x} - \frac{t_{\frac{\alpha}{2}, n-1} * s}{\sqrt{n}} \leq \mu \leq \bar{x} + \frac{t_{\frac{\alpha}{2}, n-1} * s}{\sqrt{n}} \quad (2)$$

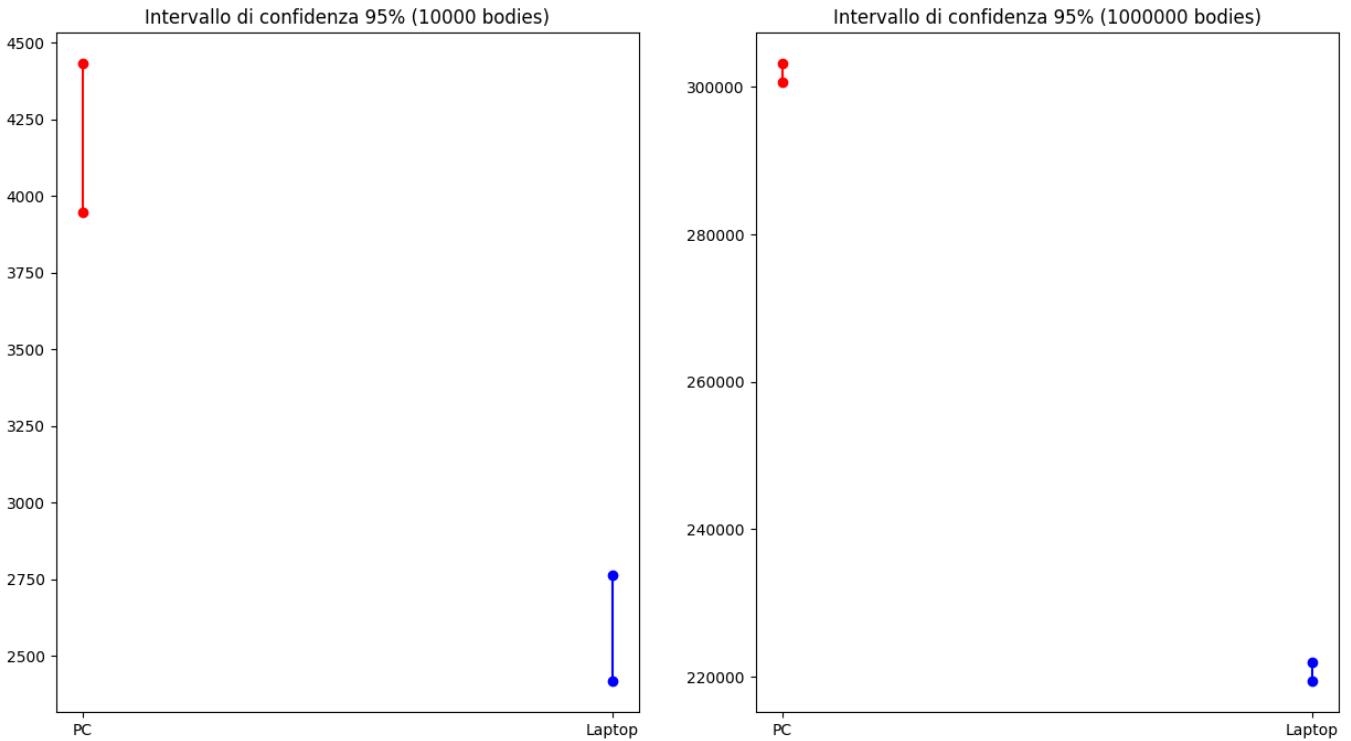


Figure 2: Intervalli di confidenza.

Il test visivo ci suggerisce che i campioni (per 10000 e 1000000 corpi) provengono da popolazioni diverse.

2.4 TEST PARAMETRICO

Infine viene eseguito un test d'ipotesi per determinare se la differenza tra i tempi di esecuzione dei due sistemi è statisticamente significativa. Dati i campioni ottenuti dalle misurazioni dirette, si procede a verificarne la normalità e l'omoschedasticità. Per verificare la normalità dei dati, vengono realizzati dei plot quantile-quantile e a supporto del test visivo, il test di *Shapiro-Wilk*. Per verificare l'omoschedasticità dei dati, oltre al test visivo, viene effettuato il test di *O'Brien*.

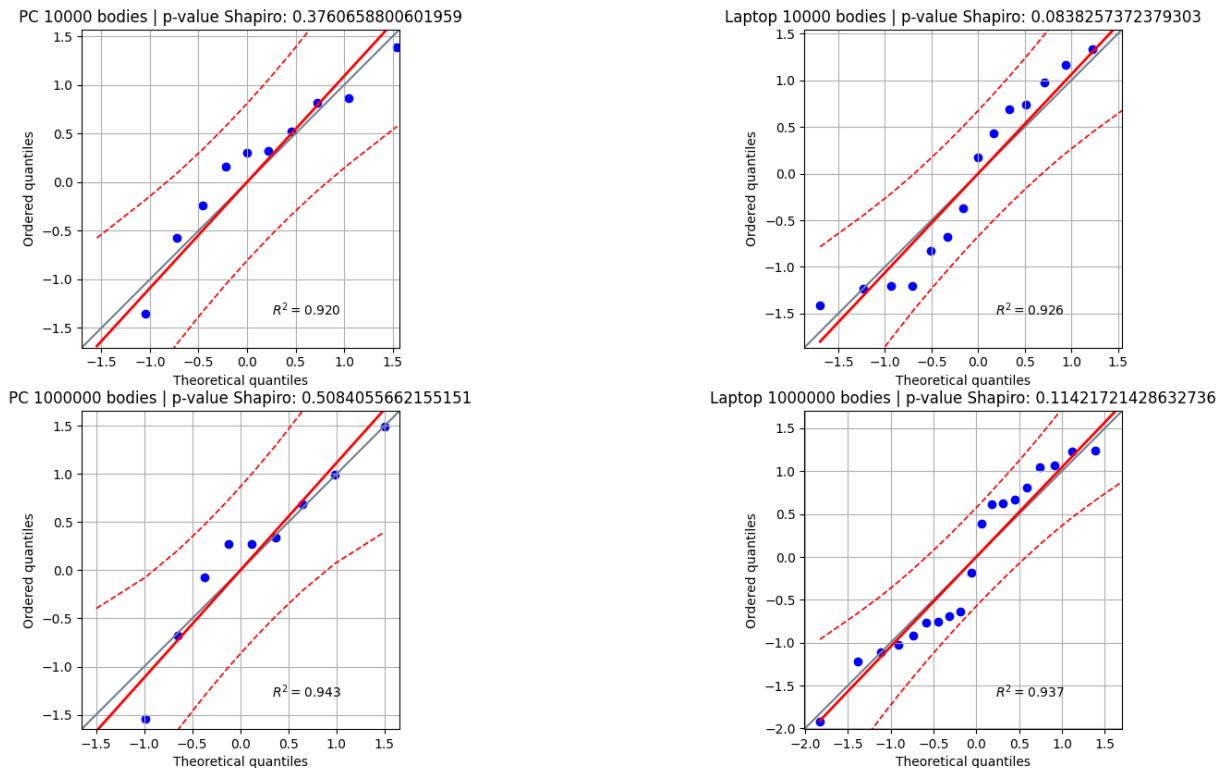


Figure 3: Plot quantile-quantile dei campioni.

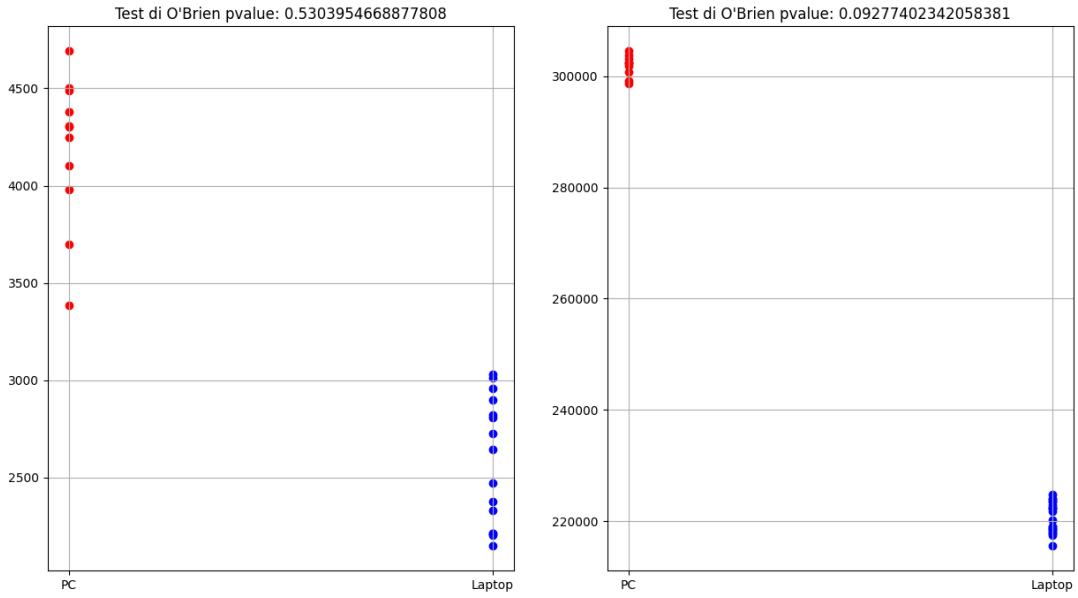


Figure 4: **Verifica di omoschedasticità.**

In entrambi i casi, sia i test visivi che non parametrici, suggeriscono che i campioni sono normali ed omoschedastici. Per questo motivo si procede con un **Two-Sample t-Test**.

```
***** Two-sample t-test con stessa varianza (10000 bodies) *****
pvalue: 2.7953784678803683e-11 | Ipotesi nulla rigettata
***** Two-sample t-test con stessa varianza (1000000 bodies) *****
pvalue: 3.025987260880871e-35 | Ipotesi nulla rigettata
```

Figure 5: **Risultati del test'ipotesi.**

Dai risultati è possibile osservare che sia in entrambi i casi (10000 corpi e 1000000 corpi), il *pvalue* è molto piccolo, dunque viene rigettata l'ipotesi nulla (H_0 : i due campioni provengono da due popolazioni con la stessa media).

3 WORKLOAD CHARACTERIZATION

Il processo di Workload Characterization ha lo scopo di generare un Workload sintetico a partire da un Workload reale. Le tecniche di analisi dei dati che verranno utilizzate durante il processo di Workload Characterization sono la **Principal Component Analysis (PCA)** ed il **Clustering**. La PCA è una tecnica che permette di ridurre la dimensionalità di un insieme di dati. Il clustering, d'altra parte, è una tecnica di apprendimento non supervisionato che mira a raggruppare dati simili tra loro in insiemi detti cluster.

3.1 RACCOLTA E PREPROCESSING DEI DATI

Il Workload reale è composto da **24 feature** (colonne del Dataset) e da **4995 osservazioni** (righe del Dataset) relative all'utilizzo di memoria da parte dei processi eseguiti su una distribuzione Linux e raccolti tramite il filesystem virtuale "proc" gestito dal kernel Linux.

È stato effettuato un filtraggio preliminare dei dati, eliminando le colonne costanti, le colonne con correlazione elevata, ed effettuando la standardizzazione dei dati.

Sono state dunque eliminate le colonne *AnonPages*, *avgLatency* ed *Errors* perchè costanti. Successivamente sono state eliminate le colonne *VmPTE*, *Dirty* e *Writeback* perchè correlate con altre colonne ($\rho > 0.99$). Infine sono state standardizzate le colonne rimanenti come segue:

$$z = \frac{x - \bar{x}}{\sigma} \quad (3)$$

Il Dataset avrà dunque **18 colonne** e **4995 righe**.

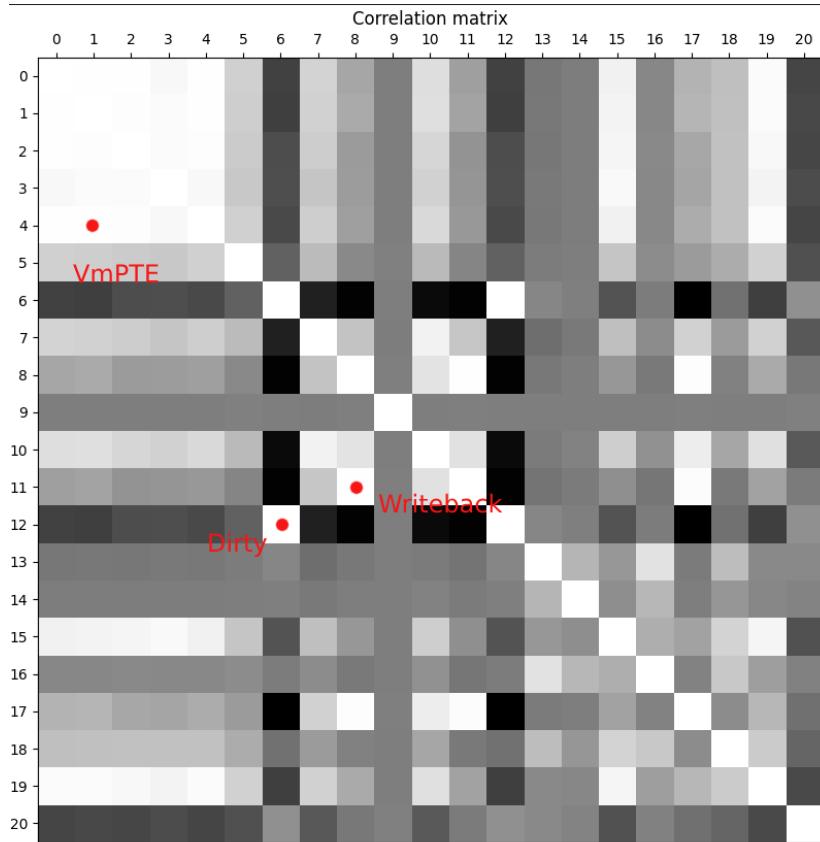


Figure 6: Matrice di Correlazione.

3.2 PRINCIPAL COMPONENT ANALYSIS

La **PCA** (Principal Component Analysis, o Analisi delle Componenti Principali) è un metodo di riduzione della dimensionalità utilizzato per analizzare e visualizzare dati multidimensionali. L'obiettivo della PCA è quello di individuare le direzioni di varianza massima nei dati e di proiettare i dati su un nuovo insieme di direzioni ortogonali, chiamate componenti principali. La prima componente principale, descrive la maggior parte della varianza; viceversa, l'ennesima descrive la varianza minore. Tale trasformazione è utile perché riusciamo a capire quante e quali sono le componenti tramite cui riduciamo la dimensionalità del problema tenendo traccia della varianza persa. Di seguito i risultati ottenuti:

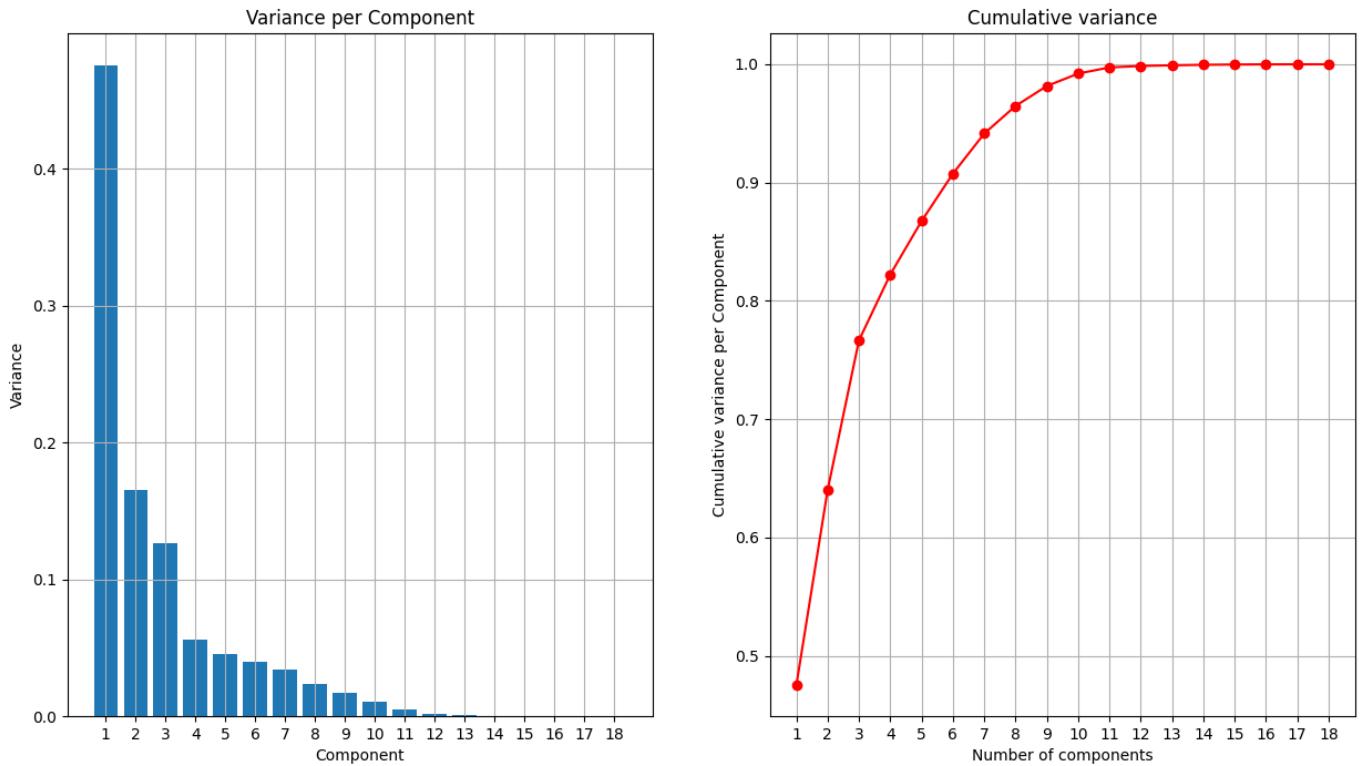


Figure 7: Principal Component Analysis.

In figura è possibile visualizzare la varianza spiegata per ogni componente, e quella cumulativa. È dunque possibile osservare che con 6 componenti principali è possibile preservare oltre il 90% della varianza totale. Nei paragrafi successivi verranno effettuati esperimenti anche con un numero diverso di componenti.

3.3 CLUSTERING

Utilizzando il **Clustering** è possibile raggruppare le righe del Dataset in Cluster, e scegliere poi una riga rappresentativa per ogni Cluster. In questo modo il numero di righe del Dataset si riduce al numero di Cluster individuati. In particolare, nel caso in esame, è stato utilizzato un clustering gerarchico con approccio agglomerativo, detto di **metodo di Ward**. Lo scopo è quello di minimizzare la variabilità interna ad ogni Cluster e massimizzare quella tra Cluster.

Attraverso la cronologia di Clusterizzazione è possibile effettuare l'analisi di sensitività ed individuare dunque il numero di Cluster più appropriato.

Numero di cluster	Distanza	Leader	Subordinato
30	10.469129	9946	9957
29	11.106636	9852	9908
28	11.724925	9937	9938
27	11.800936	9896	9942
26	11.974465	83	9940
25	12.174351	9919	9947
24	12.996562	9934	9939
23	13.763525	9941	9950
22	14.576951	9935	9944
21	17.442343	9949	9966
20	18.445545	9948	9967
19	18.476973	9945	9959
18	19.818356	4139	9956
17	23.872131	9960	9961
16	23.944824	9965	9968
15	25.653276	9904	9969
14	27.557009	9962	9973
13	30.700186	9954	9970
12	32.241632	9951	9972
11	33.817826	9971	9975
10	34.893440	9964	9978
9	35.082659	90	9963
8	46.875452	9953	9976
7	60.464241	9977	9981
6	91.743912	4942	9982
5	95.405996	89	9980
4	125.329655	9974	9979
3	135.962054	9984	9985
2	193.150345	9958	9983
1	234.466141	9986	9987

Figure 8: Cronologia di Clusterizzazione.

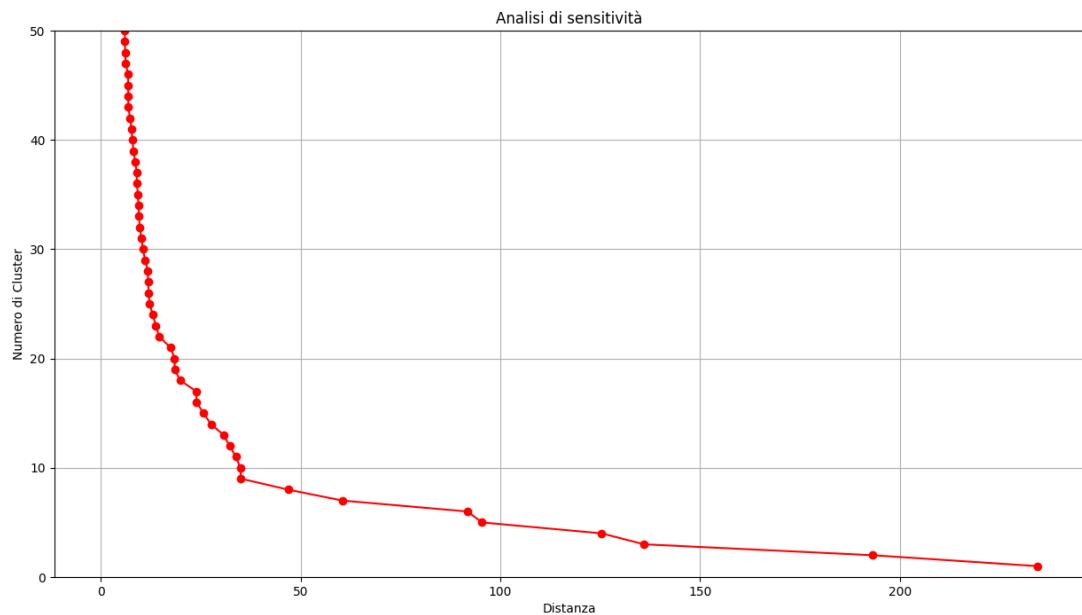


Figure 9: Analisi di sensitività.

Inizialmente sono stati scelti 9 Cluster, dei quali si visualizza l'istogramma. Successivamente verranno effettuati esperimenti anche con un numero diverso di Cluster. La scelta è stata attuata, considerando che per ottenere un numero di Cluster minore, bisogna unire Cluster molto distanti tra di loro.

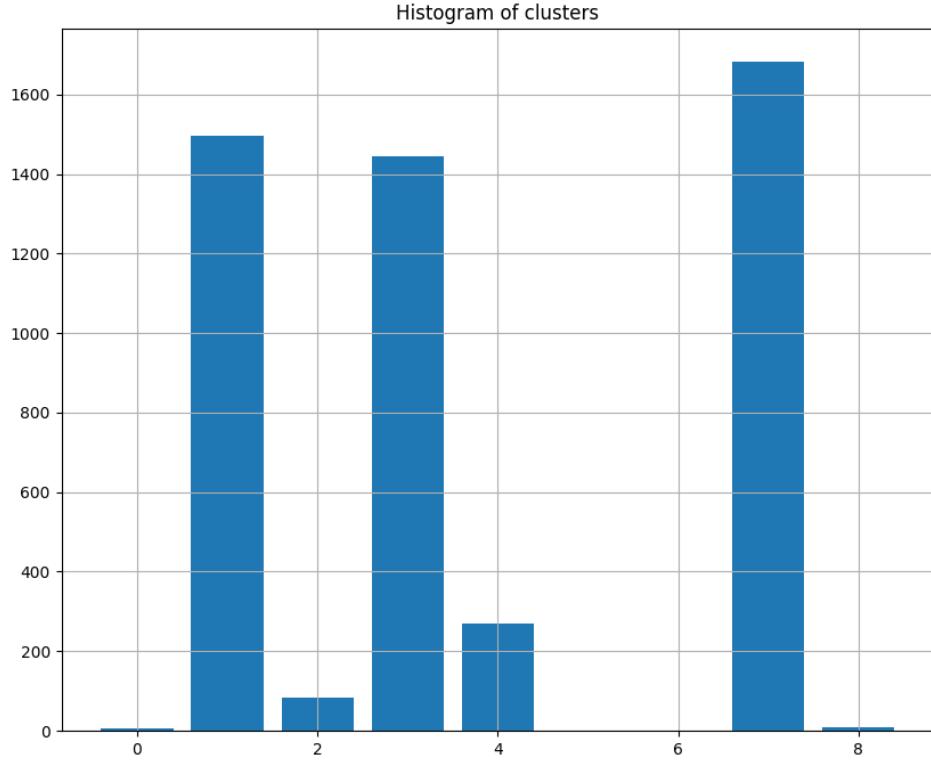


Figure 10: **Istogramma dei Cluster.**

3.4 CALCOLO DELLA DEVIANZA PERSA

Dopo aver fissato il numero delle componenti principali (6) e dei Cluster (9), si è deciso di andare ad analizzare la devianza persa dopo le operazioni precedenti.

La devianza del Dataset, prima della Principal Component Analysis è stata calcolata come segue:

$$Devianza_{TOT} = \sum_{i=1}^N \|x_i - \bar{x}\|^2 = 89910 \quad (4)$$

Con 6 componenti principali, riusciamo a preservare oltre il 90% della devianza.

$$Devianza_{PCA} = \sum_{i=1}^N \|x_{PCA_i} - \bar{x}_{PCA}\|^2 = 81601 \quad (5)$$

$$\frac{Devianza_{PCA}}{Devianza_{TOT}} = \frac{81601}{89910} = 0.907 \simeq 91\% \quad (6)$$

Una volta individuati i Cluster, è possibile calcolare la devianza all'interno di ogni Cluster, e tra i Cluster.

$$Devianza_{intra} = \sum_{i=1}^K \sum_{i=1}^{N_k} \|x_i - \bar{x}_k\|^2 = 6677 \quad (7)$$

$$Devianza_{inter} = \sum_{i=1}^K N_k \|\bar{x}_k - \bar{x}\|^2 = 74924 \quad (8)$$

$$Devianza_{PCA} = Devianza_{intra} + Devianza_{inter} \quad (9)$$

È possibile infine calcolare la devianza persa dopo le operazioni precedenti:

$$Devianza_{loss} = \left(1 - \frac{Devianza_{PCA}}{Devianza_{TOT}}\right) + \frac{Devianza_{PCA}}{Devianza_{TOT}} * \frac{Devianza_{intra}}{Devianza_{PCA}} = 0.167 \simeq 16.7\% \quad (10)$$

3.5 CONSIDERAZIONI FINALI

Infine sono stati effettuati degli esperimenti considerando varie combinazioni tra numero di componenti principali e numero di Cluster.

Numero di Cluster	Numero di Componenti Principali		
	5	6	7
5	30.0%	29.7%	28.9%
9	20.1%	16.7%	15.6%
13	17.6%	14.1%	12.5%

Table 5: Devianza persa.

Si è dunque scelto di utilizzare **6 componenti principali** e **9 Cluster** per generare il Workload sintetico, in modo da ottenere un buon trade-off tra dimensione del Workload e devianza persa.

A valle di tutte le operazioni effettuate, è stato dunque ricavato il Workload Sintetico, andando a selezionare, per ogni Cluster, un suo elemento rappresentativo, in maniera randomica.

	88	4588	63	560	2814	89	4942	1758	3002
VmPeak	170344	186284	153484	172388	211260	170344	172388	172392	170344
VmSize	170340	182620	151436	170340	211256	170340	170340	170340	170340
VmHWM	31144	41320	26612	32696	58528	31144	32696	33228	31916
VmRSS	31140	38220	26468	32000	58432	31140	31460	32012	31912
VmPTE	200	240	160	200	316	200	200	200	200
Threads	64	76	30	42	161	52	33	54	9
MemFree	4902916	4560892	5601960	4630120	4530748	4655664	4635532	4575784	4699488
Buffers	84164	194972	35192	136272	204020	89480	132072	188580	98052
Cached	864428	1118016	334676	1117224	1118384	1090572	1117192	1117864	1084804
Active	0	0	0	0	0	0	111	0	0
Inactive	454384	481040	142716	415160	524264	541220	412036	462820	386696
Dirty	624920	922508	305880	922676	908856	773332	921028	927980	882460
Writeback	4902916	4560892	5601960	4630120	4530748	4655664	4635532	4575784	4699488
Mapped	142552	176	44	152	156	324400	164	152	107284
Slab	0	0	0	0	0	1760	0	0	0
PageTables	130788	90600	78736	84340	110752	134500	83800	84356	86152
CommittedAS	28324	23848	23772	23740	23848	29956	23740	23844	24092
NumOfAllocFH	95252	113584	27808	109612	112728	106788	109580	111104	109392
procfd	8024	7248	7196	7208	7324	8028	7208	7208	7620
avgThroughput	366024	327288	298140	315244	352456	365264	316560	316320	319664
avgElapsed	2040	1020	1530	1020	1020	2040	1530	1530	1020

Figure 11: Workload sintetico.

4 WEB SERVER

Il seguente esercizio ha come obiettivo, l'analisi delle prestazioni di un Web Server.

4.1 SETUP SPERIMENTALE

Il Setup sperimentale utilizzato, prevede Web Server e Client sulla stessa macchina fisica.

Caratteristiche macchina fisica	
CPU	AMD Ryzen 5700u
Core	16
RAM	16 GB
S.O.	Ubuntu 22.04

Table 6: **Caratteristiche macchina fisica.**

In particolare, Web Server e Client sono installati su due macchine virtuali con caratteristiche differenti.

Configurazione Server	
CPU	AMD Ryzen 5700u
Core	2
RAM	512 MB
S.O.	Ubuntu 22.04 (VirtualBox)

Table 7: **Configurazione Server.**

Configurazione Client	
CPU	AMD Ryzen 5700u
Core	4
RAM	8 GB
S.O.	Ubuntu 22.04 (VirtualBox)

Table 8: **Configurazione Client.**

Sul Server è stato installato il Web Server *Apache 2.4* e successivamente sono state posizionate delle risorse di dimensioni differenti per simulare le ipotetiche richieste del Client al Server. Sul Client è stato installato *Apache JMeter 5.6* per generare le richieste HTTP.

Risorse Server		
Risorsa	Tipologia	Dimensione
lorem.txt	Testo	98 KB (Small)
random.txt	Testo	101 KB (Small)
fall.png	Immagine	1.5 MB (Medium)
space.jpg	Immagine	1.2 MB (Medium)
flower.jpg	Immagine	12 MB (Large)
wallpaper.jpg	Immagine	12 MB (Large)

Table 9: Risorse Server.

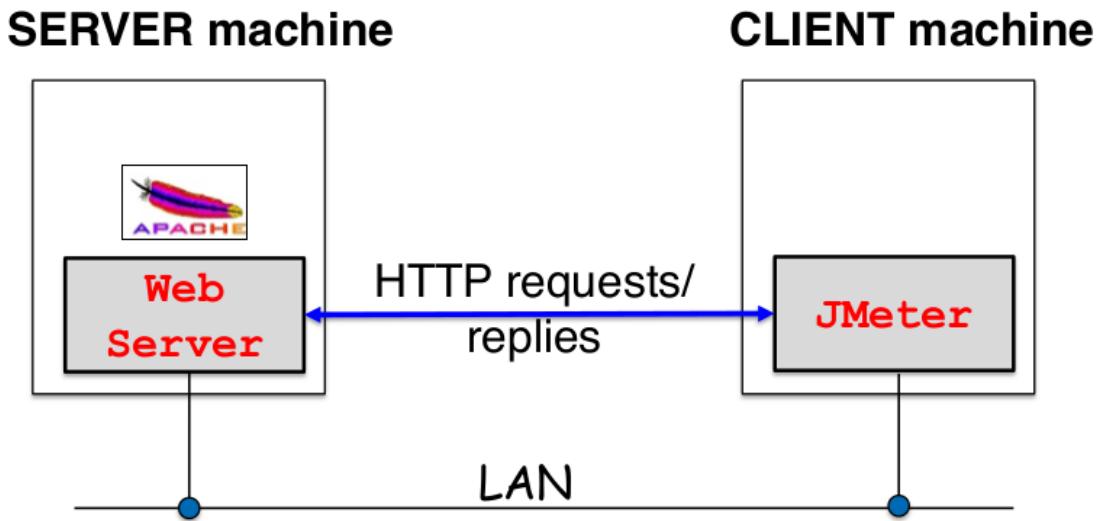


Figure 12: Setup sperimentale.

4.2 CAPACITY TEST

Un **Capacity Test** è una forma di test che mira a determinare la performance di un sistema (Web Server). Il test consiste nell'invio di un gran numero di richieste al Server in un breve periodo di tempo, al fine di determinare il punto in cui il server inizia a mostrare segni di sovraccarico e le prestazioni iniziano a diminuire.

4.2.1 CAPACITY TEST E PERFORMANCE ANALYSIS

Le metriche che vengono utilizzate per caratterizzare le performance del SUT in esame sono:

- **Response time:** intervallo di tempo che intercorre tra quando il Client effettua la richiesta e quando il Server ritorna il risultato della richiesta.
- **Throughput:** il numero di richieste per unità di tempo che il Server è in grado di soddisfare.

Date queste due metriche di performance si andranno a ricercare due punti notevoli per il funzionamento del Web Server:

- **Knee Capacity:** punto di funzionamento ideale del sistema, corrisponde al punto in cui il rapporto tra Throughput e Response Time (detto Power) è massimo. Oltre questo punto, il Response Time cresce rapidamente col crescere del carico, mentre c'è un incremento del Throughput meno significativo.

- **Usable Capacity:** punto in cui si ha il Throughput massimo, senza superare i vincoli sul tempo di risposta.

Sul Client, per effettuare il test, utilizzando il tool JMeter, si costruisce un Test Plan.

- **Thread Group**

- **Number of Threads:** 50.
- **Ramp-up period (seconds):** 50.
- **Duration (seconds):** 300.

- **HTTP Request:** sono state aggiunte le richieste specificate in precedenza.

- **Random Controller**

- **Constant Throughput Timer (CTT):** (250, 750, 1250, 1750, 2250, 2750, 3250, 3750, 4250).

- **Simple Data Writer:** fornisce l'output del Test Plan come file .csv.

Il Test Plan viene lanciato 3 volte per ogni CTT e tra un esperimento ed un altro viene resettato il Server al fine di evitare fenomeni di caching. Per ogni esecuzione vengono poi calcolati i valori di Throughput, Response Time e Power come segue:

$$\text{Throughput} = \frac{\text{Total number of requests correctly served}}{\text{Duration (seconds)}} \quad (11)$$

$$\text{Response Time} = \text{mean}(\text{Elapsed}) \quad (12)$$

$$\text{Power} = \frac{\text{Throughput}}{\text{Response Time}} \quad (13)$$

Di seguito, i risultati ottenuti:

CTT (req/min)	Throughput (req/s)			Mean Throughput (req/s)
	Oss. 1	Oss. 2	Oss. 3	
250	4.334047	4.352038	4.331603	4.339229
750	12.673601	12.655104	12.650228	12.659644
1250	20.982763	20.957547	20.963655	20.967988
1750	29.260436	29.279985	29.247665	29.262695
2250	37.595740	37.562568	37.578741	37.579016
2750	45.862779	45.773321	45.914677	45.850258
3250	54.099656	53.968984	53.708977	53.925872
3750	60.123660	62.155631	59.81595	60.698414
4250	48.146765	54.620392	55.443656	52.736937

Table 10: **Throughput.**

CTT (req/min)	Response Time (ms)			Mean Reponse Time (ms)
	Oss. 1	Oss. 2	Oss. 3	
250	26.316361	24.314488	27.456505	26.029117
750	26.373686	29.215547	23.609443	26.399558
1250	27.306900	29.790307	28.524893	20.967988
1750	28.657892	27.796154	36.293387	30.915810
2250	37.165499	41.433505	45.607774	41.402259
2750	36.955134	46.821434	42.429764	42.068777
3250	61.843072	72.982083	80.343950	71.723035
3750	148.117252	70.576393	86.036176	101.576607
4250	588.562998	456.202991	441.770841	495.512276

Table 11: Response Time.

CTT (req/min)	Power [(req/s)/(ms)]			Mean Power [(req/s)/(ms)]
	Oss. 1	Oss. 2	Oss. 3	
250	0.1646902	0.1789895	0.1577623	0.1671473
750	0.4805396	0.4331634	0.5358122	0.4831717
1250	0.7684052	0.7035022	0.7349249	0.7356108
1750	1.0210254	1.0533826	0.8058676	0.9600919
2250	1.0115764	0.9065747	0.8239547	0.9140353
2750	1.2410394	0.9776147	1.0821337	1.1002626
3250	0.8747893	0.7394826	0.6684881	0.76092
3750	0.4059194	0.8806859	0.6952419	0.6606157
4250	0.0818039	0.1197283	0.1255032	0.1090118

Table 12: Power.

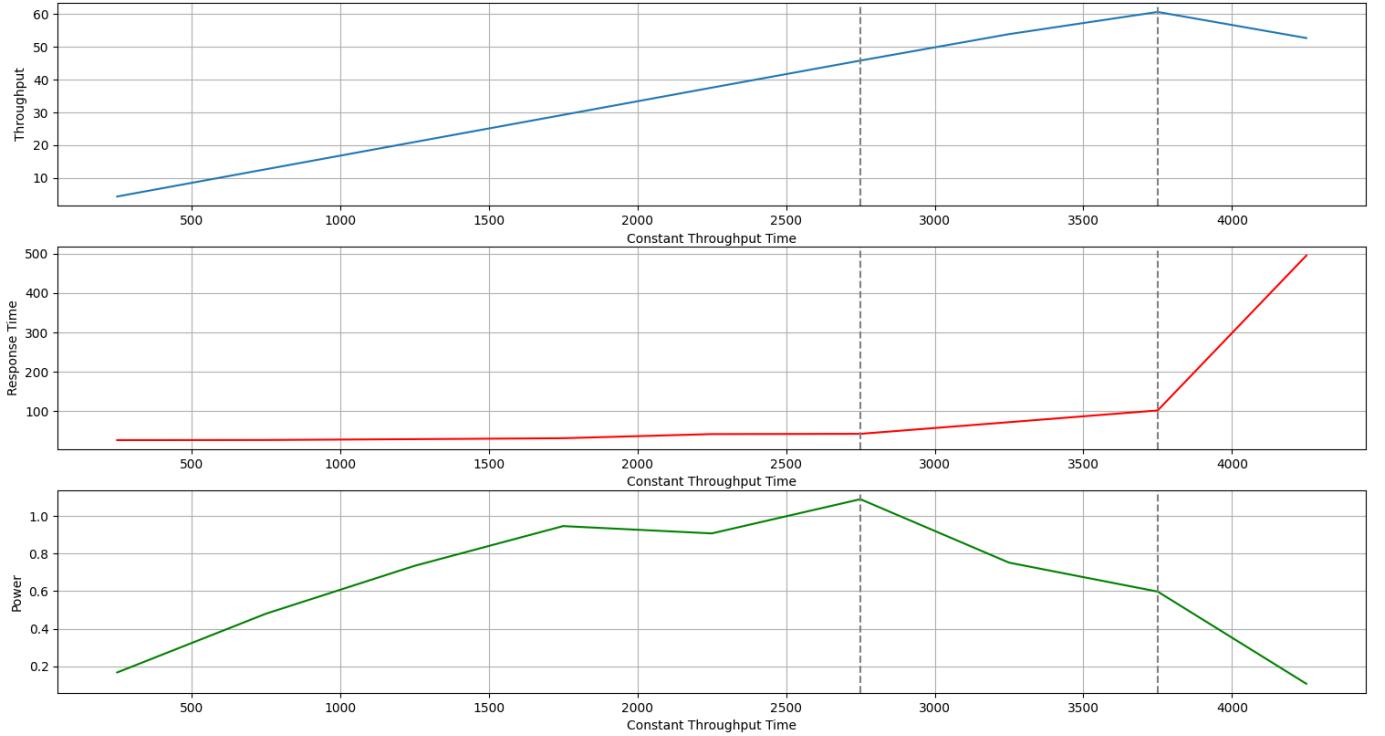


Figure 13: Grafici Capacity Test.

Dai grafici ottenuti è possibile individuare il punto di **Knee Capacity (2750 req/min)** e di **Usable Capacity (3750 req/min)** rispettivamente come il punto in cui la Power è massima, ed il punto in cui il Throughput è massimo ed oltre il quale quest'ultimo diminuisce.

4.2.2 BOTTLENECK PERFORMANCE

Durante gli esperimenti effettuati in precedenza, sono stati raccolti dei dati lato Server mediante **vmstat**. In particolare sono stati misurati i seguenti parametri:

- **r**: questo valore indica il numero di task in esecuzione ed in attesa di risorse del processore. Quando questo numero è maggiore del numero di CPU del Server, è presente un bottleneck.
- **b**: questo valore indica il numero di task in attesa di input/output. Questa colonna dovrebbe contenere il valore 0 per la maggior parte del tempo.
- **si**: quantità di memoria swappata dal disco.

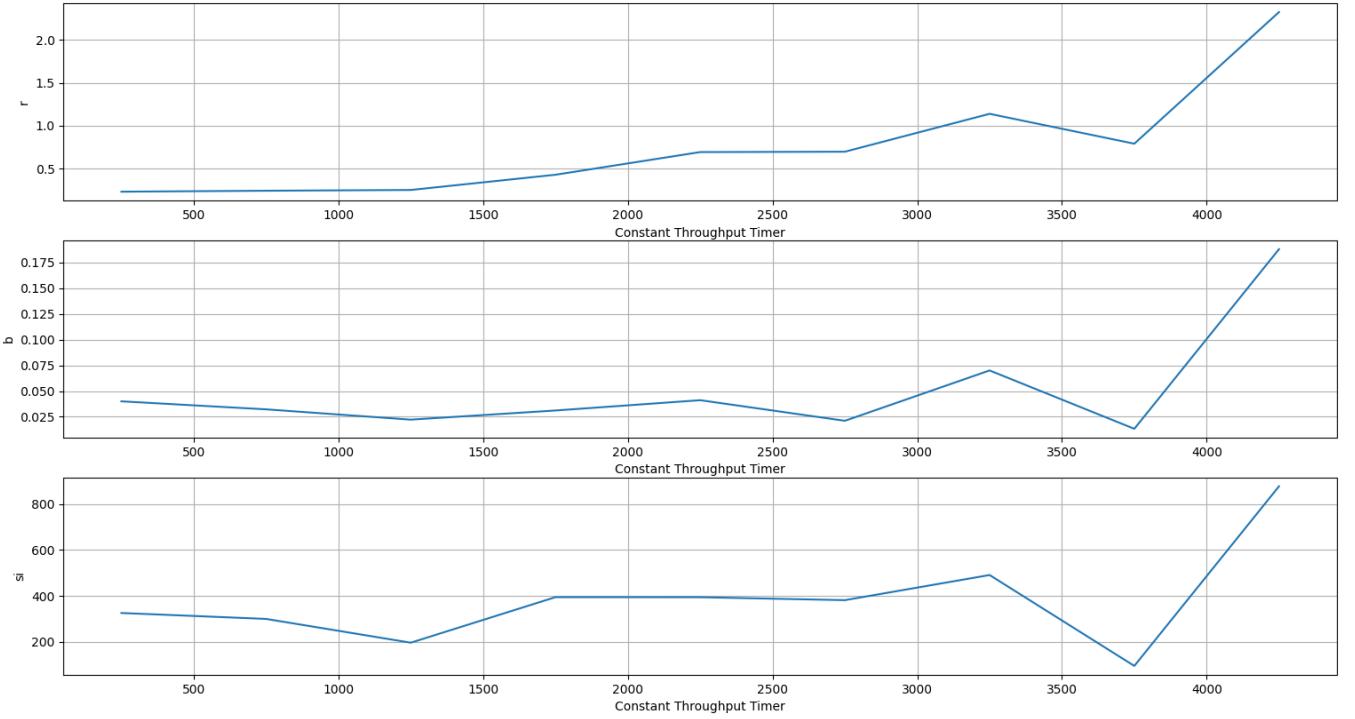


Figure 14: Dati raccolti con vmstat.

4.2.3 FAIRNESS INDEX

Successivamente è stato creato un Test Plan, con diversi Thread Group che richiedono in maniera concorrente le stesse risorse su un server. L'obiettivo è quello di calcolare il **Fairness Index**. Sono stati effettuati 4 esperimenti, ognuno con 3 Thread Group con 5, 15 e 30 Thread, ogni misurazione è stata ripetuta 3 volte.

Thread Group			
CTT Totale	5 Threads	10 Threads	15 Threads
1000	100	300	600
2000	200	600	1200
3000	300	900	1800
4000	400	1200	2400

Table 13: CTT per Thread Group.

La valutazione del Fairness Index è così fatta:

$$fairness = f(x_1, x_2, \dots, x_n) = \frac{(\sum_{i=1}^n x_i)^2}{n \sum_{i=1}^n x_i^2} \quad (14)$$

dove:

$$x_i = \frac{T_i}{O_i} \quad (15)$$

è il Normalized Throughput, con:

- T_i : Throughput misurato.
- O_i : Fair Throughput (specificato attraverso il Constant Throughput Timer).

Di seguito i risultati della valutazione del Fairness Index:

	Thread Group		
	5 Threads	10 Threads	15 Threads
CTT	100	300	600
Fair Throughput	1.666	5	10
Measured Throughput	1.681	5.047	10.095
Normalized Throughput	1.008	1.009	1.009
Fairness Index		0.999	

Table 14: CTT Totale 1000.

	Thread Group		
	5 Threads	10 Threads	15 Threads
CTT	200	600	1200
Fair Throughput	3.333	10	20
Measured Throughput	3.349	10.046	20.080
Normalized Throughput	1.005	1.004	1.004
Fairness Index		0.999	

Table 15: CTT Totale 2000.

	Thread Group		
	5 Threads	10 Threads	15 Threads
CTT	300	900	1800
Fair Throughput	5	15	30
Measured Throughput	5.013	15.030	30.017
Normalized Throughput	1.002	1.002	1.000
Fairness Index		0.999	

Table 16: CTT Totale 3000.

	Thread Group		
	5 Threads	10 Threads	15 Threads
CTT	400	1200	2400
Fair Throughput	6.666	20	40
Measured Throughput	6.093	18.254	36.501
Normalized Throughput	0.914	0.912	0.912
Fairness Index		0.999	

Table 17: **CTT Totale 4000.**

4.2.4 CONSIDERAZIONI FINALI

Nella prima parte dell'esercizio sono stati identificati i punti di **Knee Capacity** e di **Usable Capacity** in corrispondenza dei carichi **2750 req/min** e **3750 req/min**. Il rapporto tra i rispettivi punti è inferiore al 75%:

$$\frac{\text{Knee Capacity}}{\text{Usable Capacity}} = \frac{2750}{3750} = 0.73 < 0.75 \quad (16)$$

Questo garantisce che il punto di Knee Capacity sia sufficientemente lontano dal punto di Usable Capacity, oltre il quale, iniziano a degradare le prestazioni del Server.

Successivamente sono stati analizzati i dati raccolti lato Server. È possibile osservare che in prossimità del punto di Usable Capacity, aumenta velocemente l'utilizzo delle risorse da parte del Server, in particolare il valore **r** supera il numero di CPU presenti sul Server, indice della presenza di un Bottleneck. Inoltre i valori **b** e **si** crescono rapidamente, suggerendo che la memoria fisica sia satura.

Infine, mediante la realizzazione di un apposito Test Plan, è stata analizzato il **Fairness Index**. Si può notare che per ogni carico scelto, il Fairness Index è prossimo ad 1, segno che il sistema sta distribuendo il Throughput in maniera equa tra i vari Thread Group. Una possibile spiegazione, è aver scelto un Fair Throughput per ogni Thread Group proporzionale al numero di Thread presenti.

4.3 WORKLOAD CHARACTERIZATION

Si sperimenta come un Workload reale applicato al SUT possa essere caratterizzato al fine di effettuare degli esperimenti con un Workload sintetico che è caratterizzato dalla gran parte della variabilità del dataset, ma con una dimensionalità ridotta. Per effettuare questo processo di Workload Characterization sono state eseguite le seguenti fasi descritte in figura:

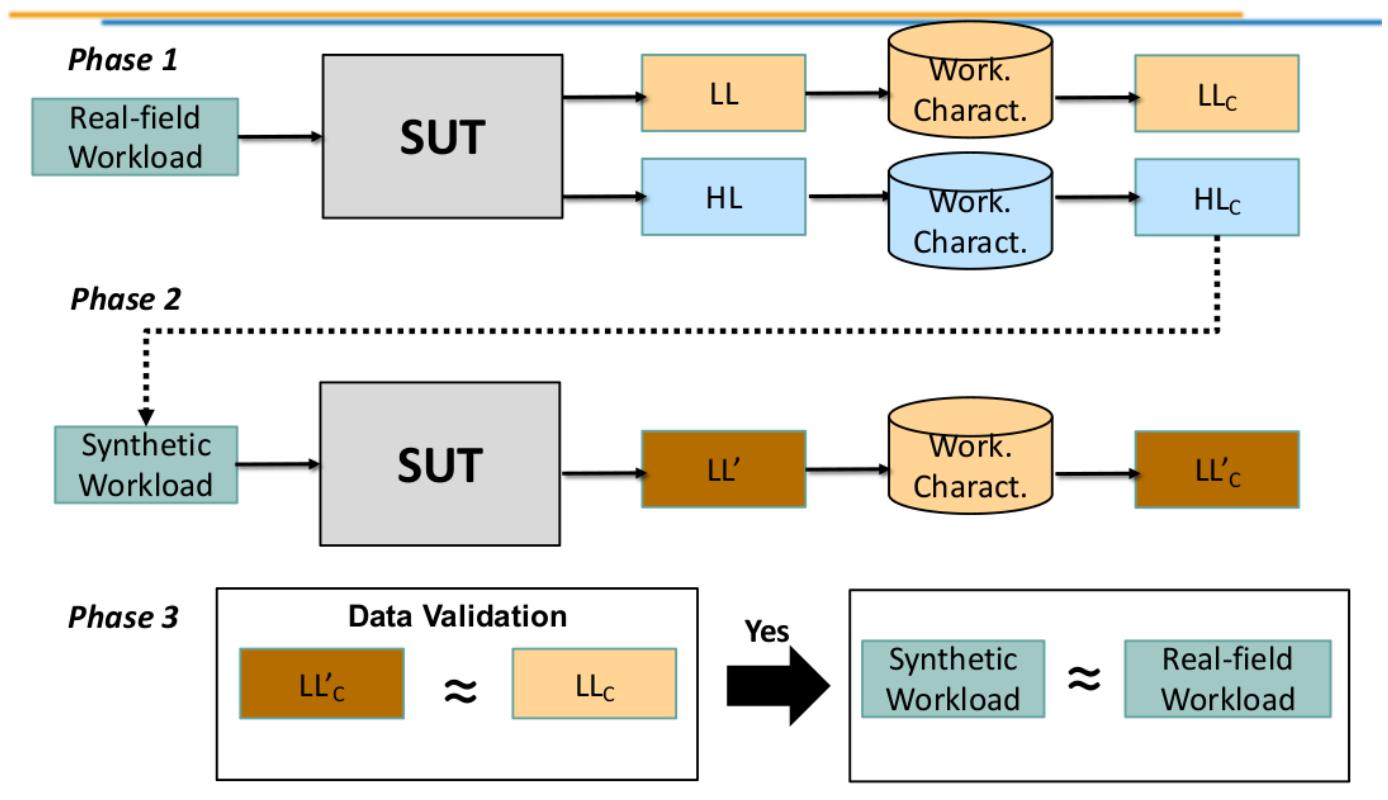


Figure 15: Workload Characterization.

4.3.1 CREAZIONE DEL WORKLOAD REALE

Nella prima fase viene generato il Workload reale mediante la creazione di un Test Plan su *Jmeter*. In particolare il Test Plan comprende la creazione di **5 Thread Group**, i quali richiederanno le **6 risorse** presenti sul Server (specificate nei paragrafi precedenti). Ogni Thread Group avrà inoltre un *Random Controller* ed un *Constant Throughput Timer* per gestire le richieste da effettuare. Abbiamo in totale **30 tipologie di richieste**, ognuna caratterizzata dalla coppia (*Thread Group*, *Risorsa richiesta*). I parametri di alto livello verranno collezionati lato Client mediante un *Simple Data Writer* su *Jmeter*, mentre lato Server mediante *vmstat*.

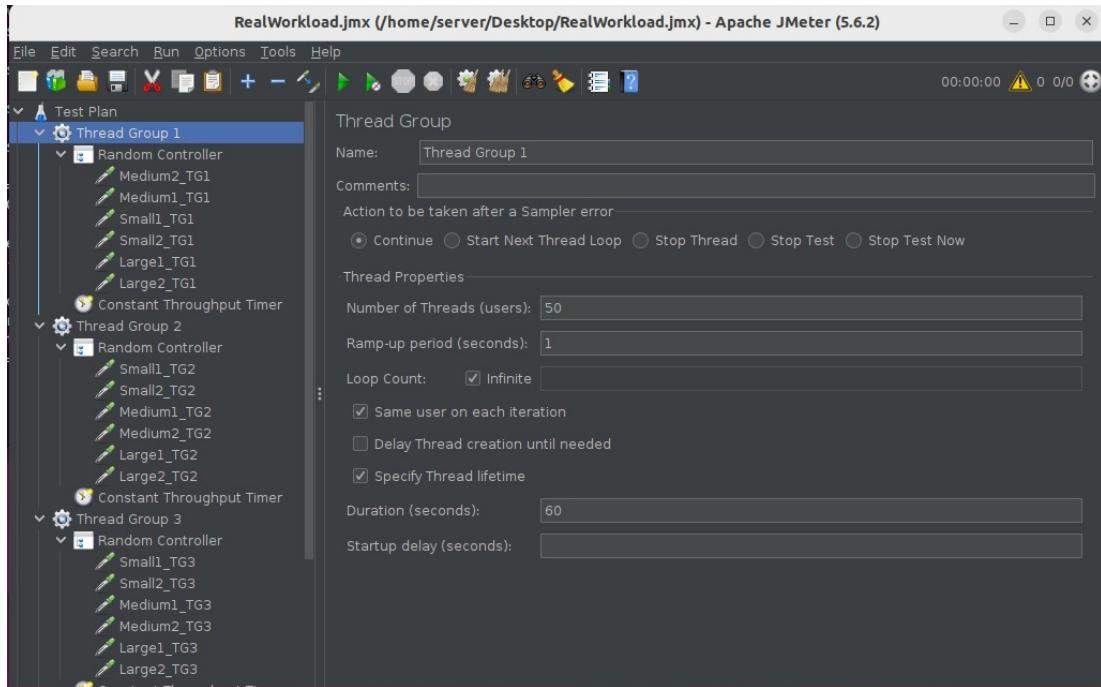


Figure 16: Test Plan.

4.3.2 WORKLOAD CHARACTERIZATION HL

Una volta eseguito il Test Plan, e dopo aver collezionato i parametri di alto livello mediante un *Simple Data Writer*, viene effettuata la caratterizzazione del Workload sulla falsa riga del Capitolo 2 dell'elaborato.

Inizialmente è stato effettuato un filtraggio preliminare dei dati, eliminando le colonne non numeriche, quelle costanti e con correlazione elevata. Successivamente è stata realizzata la standardizzazione dei dati.

In seguito è stata effettuata l'analisi delle componenti principali, di cui vengono riportati i risultati:

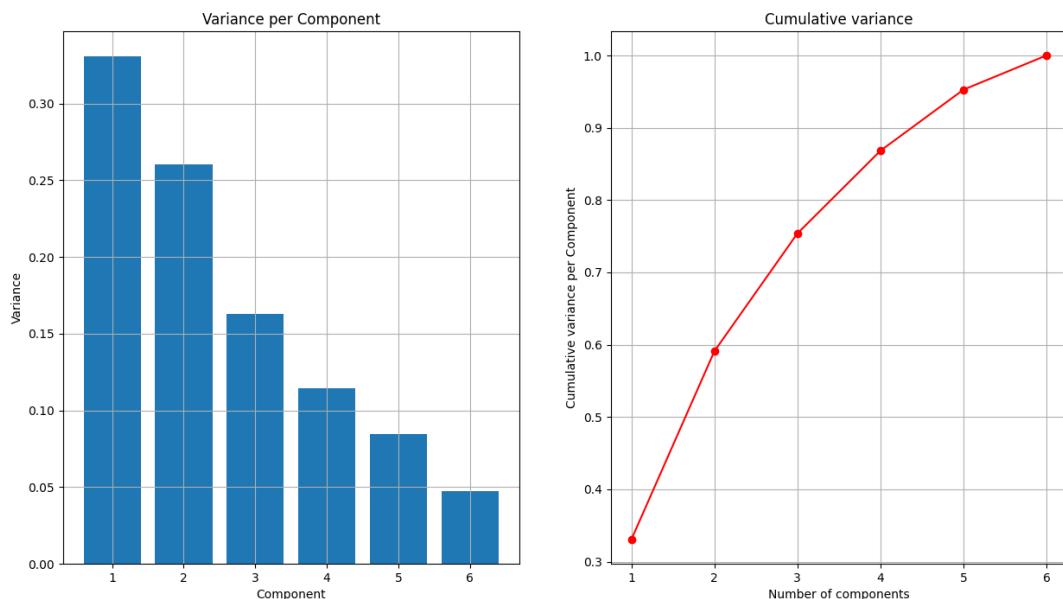


Figure 17: HL Principal Component Analysis.

Sono state selezionate dunque **5 componenti principali** in modo da preservare il 95% della

varianza complessiva.

Dopo la PCA, è stata effettuata la Clusterizzazione dei dati. Dalla cronologia di Clusterizzazione, dal grafico ottenuto e da diversi esperimenti effettuati, sono stati selezionati **15 Cluster**.

Numero di cluster	Distanza	Leader	Subordinato
30	10.476046	14288	14332
29	10.963382	14335	14352
28	11.029982	14305	14349
27	11.161854	14334	14350
26	11.628657	14309	14330
25	11.630018	14279	14347
24	12.346915	14331	14361
23	13.346451	14315	14358
22	15.285408	14318	14319
21	15.885836	14351	14353
20	17.902627	14348	14364
19	18.535522	14345	14346
18	19.035073	14357	14367
17	20.849592	14324	14362
16	23.435063	14354	14359
15	24.632194	14339	14355
14	24.680533	14340	14363
13	24.912127	14356	14368
12	35.180283	14365	14374
11	35.690788	2644	14360
10	46.447288	14376	14378
9	47.515621	14370	14371
8	48.914408	14372	14373
7	55.683381	14375	14377
6	59.915493	14381	14382
5	81.263485	14366	14380
4	82.678167	14369	14379
3	90.497400	14292	14385
2	119.074880	14383	14386
1	146.069660	14384	14387

Figure 18: **HL Cronologia di clusterizzazione.**

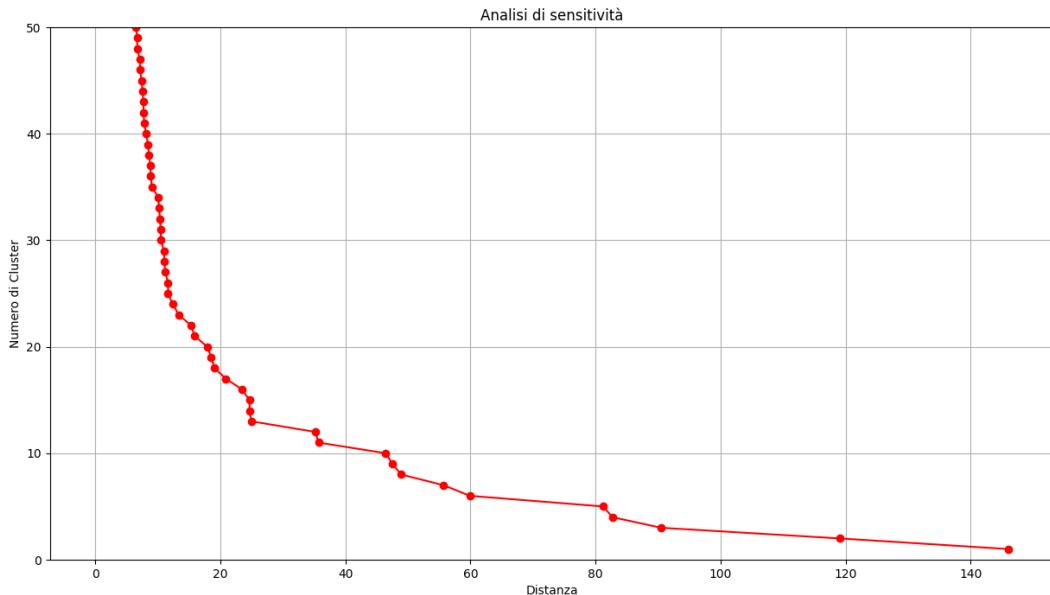


Figure 19: **HL Analisi di sensitività.**

Complessivamente è stato perso il 13.8% della devianza totale.

4.3.3 WORKLOAD CHARACTERIZATION LL

Lato Server sono stati raccolti i parametri mediante *vmstat*.

Anche qui è stato effettuato un filtraggio preliminare dei dati, eliminando le colonne non numeriche, quelle costanti e con correlazione elevata. Successivamente è stata realizzata la standardizzazione dei

dati.

In seguito è stata effettuata l'analisi delle componenti principali, di cui vengono riportati i risultati:

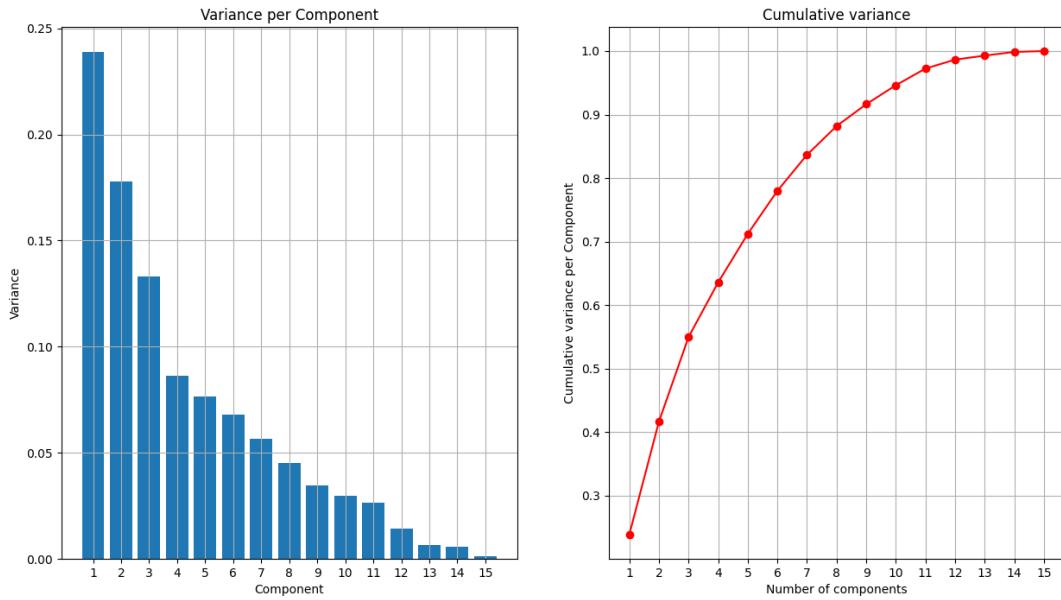


Figure 20: LL Principal Component Analysis.

Sono state selezionate dunque **10 componenti principali** in modo da preservare il 95% della varianza complessiva.

Dopo la PCA, è stata effettuata la Clusterizzazione dei dati. Dalla cronologia di Clusterizzazione, dal grafico ottenuto e da diversi esperimenti effettuati, sono stati selezionati **25 Cluster**.

Numero di cluster	Distanza	Leader	Subordinato
30	5.686289	516	541
29	5.864847	128	550
28	5.980539	524	560
27	6.263388	1	180
26	6.460578	534	552
25	6.616100	548	570
24	6.808523	559	565
23	6.985781	543	558
22	7.016852	554	557
21	7.127386	546	564
20	7.390716	490	555
19	7.730022	530	567
18	7.922009	553	575
17	9.538619	571	577
16	9.584566	556	569
15	9.890541	572	573
14	10.158623	563	576
13	11.186334	580	582
12	11.791597	579	585
11	12.554391	566	581
10	12.572756	578	583
9	17.110639	562	584
8	18.181827	568	589
7	19.729235	574	587
6	24.476071	0	590
5	25.809228	591	592
4	26.094970	588	594
3	29.965522	593	595
2	32.371828	586	596
1	34.449262	16	597

Figure 21: LL Cronologia di clusterizzazione.

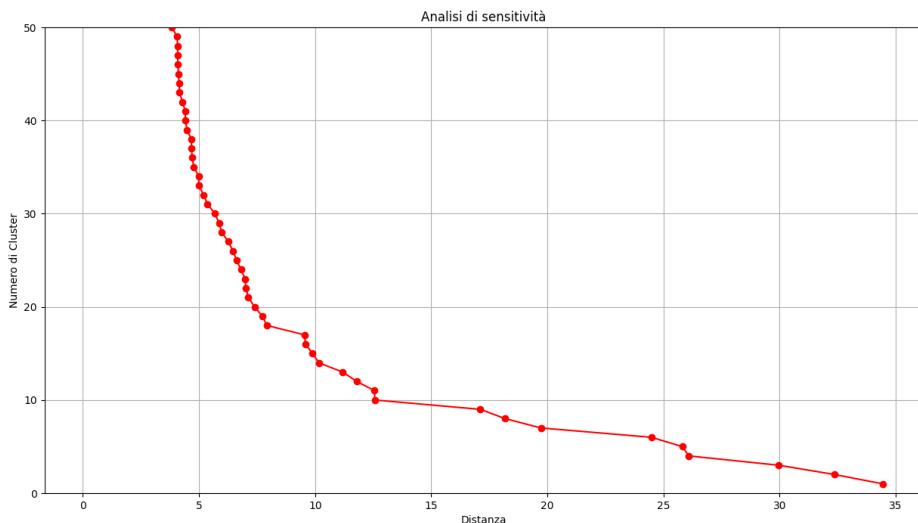


Figure 22: LL Analisi di sensitività.

Complessivamente è stato perso il 17.5% della devianza totale.

4.3.4 CREAZIONE WORKLOAD SINTETICO

Per la realizzazione del Workload sintetico, sono state analizzate per ogni Cluster, le percentuali di risorse presenti nel Cluster. Per ogni Cluster è stata selezionata la risorsa più presente, o se già scelta per un Cluster, la seconda più presente e così via.

Risorsa	Cluster														
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Large1_TG1	0.00	0.16	0.00	0.00	0.00	0.00	0.00	0.00	0.59	0.00	0.00	0.00	0.00	0.00	0.02
Large1_TG2	0.00	0.20	0.00	0.00	0.00	0.00	0.00	0.38	0.00	0.00	0.00	0.03	0.00	0.01	
Large1_TG3	0.00	0.14	0.49	0.00	0.00	0.00	0.00	0.02	0.46	0.00	0.00	0.07	0.00	0.01	
Large1_TG4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.02	0.00	0.46
Large1_TG5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.51
Large2_TG1	0.25	0.20	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.05	0.00	0.00	
Large2_TG2	0.18	0.16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.02	0.00	0.00
Large2_TG3	0.01	0.14	0.51	0.00	0.00	0.00	0.00	0.54	0.00	0.00	0.03	0.00	0.00	0.00	0.00
Large2_TG4	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.00	0.00	0.00	
Large2_TG5	0.31	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.17	0.00	0.00	0.00	
Medium1_TG1	0.00	0.00	0.00	0.14	0.00	0.00	0.19	0.00	0.00	0.00	0.10	0.00	0.05	0.06	0.00
Medium1_TG2	0.00	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.02	0.04	0.00	
Medium1_TG3	0.00	0.00	0.00	0.00	0.00	0.25	0.24	0.00	0.00	0.20	1.00	0.03	0.11	0.00	
Medium1_TG4	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Medium1_TG5	0.00	0.00	0.00	0.01	0.12	0.00	0.00	0.50	0.00	0.00	0.00	0.03	0.02	0.00	
Medium2_TG1	0.00	0.00	0.00	0.14	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.05	0.00	0.00	
Medium2_TG2	0.00	0.00	0.00	0.10	0.01	0.00	0.00	0.00	0.00	0.04	0.00	0.08	0.04	0.00	
Medium2_TG3	0.00	0.00	0.00	0.01	0.00	0.26	0.19	0.00	0.00	0.14	0.00	0.02	0.06	0.00	
Medium2_TG4	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	
Medium2_TG5	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	
Small1_TG1	0.00	0.00	0.00	0.15	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.02	0.06	0.00	
Small1_TG2	0.00	0.00	0.00	0.08	0.00	0.00	0.05	0.00	0.00	0.04	0.00	0.02	0.07	0.00	
Small1_TG3	0.00	0.00	0.00	0.00	0.25	0.14	0.00	0.00	0.16	0.00	0.00	0.17	0.00	0.00	
Small1_TG4	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
Small1_TG5	0.00	0.00	0.00	0.00	0.14	0.00	0.00	0.00	0.00	0.00	0.05	0.02	0.00	0.00	
Small2_TG1	0.00	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.02	0.07	0.00	
Small2_TG2	0.00	0.00	0.00	0.00	0.00	0.25	0.05	0.00	0.00	0.10	0.00	0.02	0.09	0.00	
Small2_TG3	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.00	0.00	0.00	0.03	0.04	0.00	0.00	
Small2_TG4	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.00	0.00	0.02	0.00	0.02	0.02	0.00	
Small2_TG5	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.00	0.00	0.02	0.00	0.02	0.02	0.00	

Figure 23: **Workload sintetico.**

Pertanto, il Workload sintetico risulta così composto:

Cluster	Richiesta
0	Large2_TG5
1	Large1_TG2
2	Large2_TG3
3	Small1_TG1
4	Small1_TG5
5	Medium2_TG3
6	Medium1_TG1
7	Medium1_TG5
8	Large1_TG1
9	Large1_TG3
10	Small1_TG3
11	Medium1_TG3
12	Medium2_TG2
13	Small2_TG1
14	Large1_TG5

Table 18: **Workload sintetico.**

Una volta individuato il Workload sintetico, è stato creato un Test Plan e sono stati raccolti i parametri lato Server mediante l'utilizzo di *vmstat*.

4.3.5 WORKLOAD CHARACTERIZATION LL'

Per la caratterizzazione di LL' è stato utilizzato lo stesso workflow utilizzato per LL.

È stato effettuato un filtraggio preliminare dei dati, eliminando le colonne non numeriche, quelle costanti e con correlazione elevata. Successivamente è stata realizzata la standardizzazione dei dati. In seguito è stata effettuata l'analisi delle componenti principali, di cui vengono riportati i risultati:

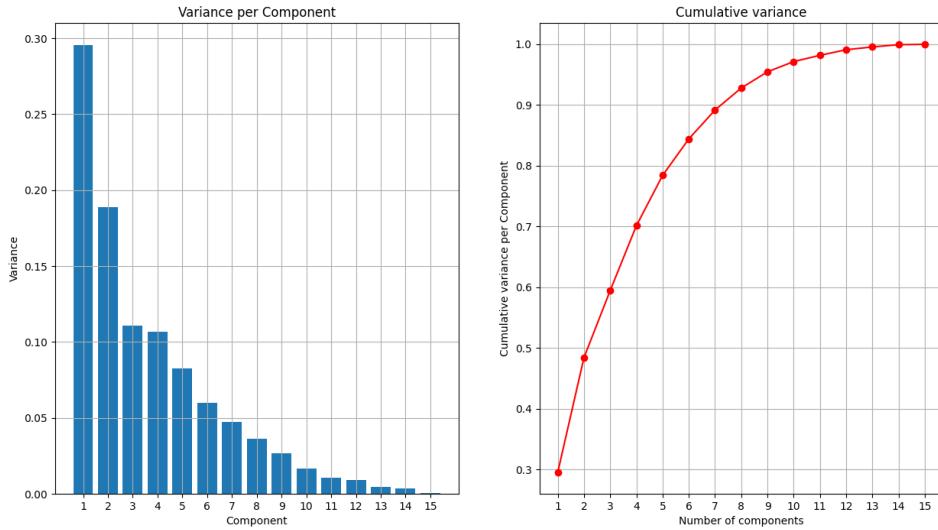


Figure 24: LL' Principal Component Analysis.

Sono state selezionate dunque **10 componenti principali**, come per LL, preservando il 95% della varianza complessiva.

Dopo la PCA, è stata effettuata la Clusterizzazione dei dati. Sono stati selezionati **25 Cluster**.

Numero di cluster	Distanza	Leader	Subordinato
30	4.564445	430	434
29	4.617870	416	424
28	4.767813	401	437
27	4.788902	427	450
26	4.860454	58	157
25	5.283980	112	118
24	5.454300	399	431
23	5.495408	127	449
22	5.515886	408	442
21	5.992304	2	445
20	6.153999	405	455
19	6.436482	117	447
18	6.758750	410	451
17	6.882817	119	454
16	7.091342	440	444
15	8.308155	453	460
14	8.323868	456	461
13	8.396661	443	458
12	9.369551	18	466
11	11.159638	175	464
10	12.073237	448	463
9	14.744641	452	459
8	15.812095	457	469
7	17.541645	465	471
6	19.961269	468	472
5	22.005998	0	462
4	23.503371	1	474
3	26.960484	470	473
2	31.085529	467	476
1	39.439296	475	477

Figure 25: LL' Cronologia di clusterizzazione.

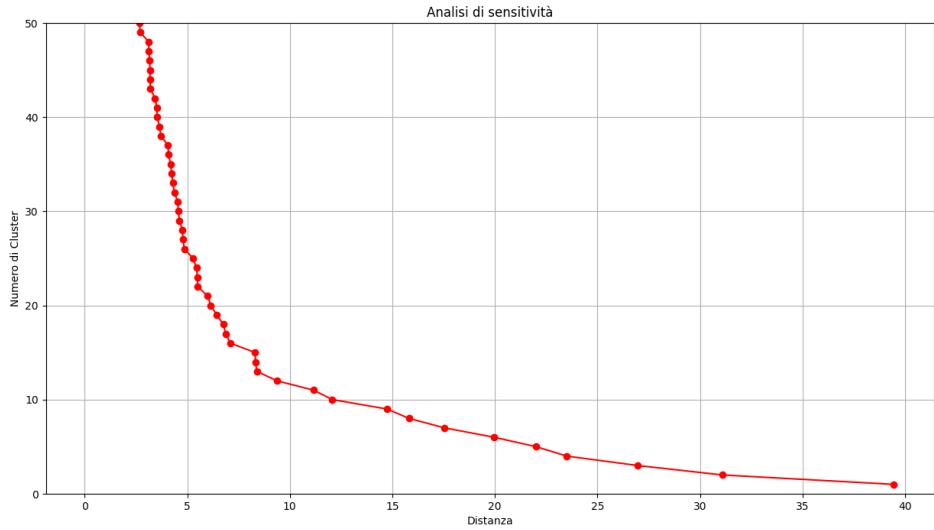


Figure 26: LL' Analisi di sensitività.

Complessivamente è stato perso l' 11.5% della devianza totale.

4.3.6 CONFRONTO TRA LL_c E LL'_c

Come ultimo passaggio, si verifica che il Workload LL' sia statisticamente rappresentativo del Workload LL . Al fine di effettuare la Data Validation dei due Workload, si segue il seguente pseudocodice:

Algorithm 1: Procedimento per la validazione di LL'_c e LL_c

```

for ogni componente principale do
    controlla la normalità dei dati ( $LL'_c$  e  $LL_c$ )
    if i dati non sono distribuiti normalmente then
        applicare un test non parametrico (rank-sum test)
    else
        verifica se i dati hanno la stessa varianza (omoschedasticità)
        if la varianza è uguale then
            Applicare two sample t-test
        else
            Applicare two sample t-test con varianza diversa
        end if
    end if
end for

```

In particolare per verificare la normalità dei dati, è stato effettuato un controllo visivo utilizzando un **plot quantile-quantile**, ed il **test di Kolmogorov-Smirnov**. Per le colonne normali, l'omoschedasticità è stata verificata utilizzando un test visivo (**Boxplot**).

Componente Principale	Normale	Omoschedasticità	Test
1	No	-	rank-sum
2	Sì	No	t-test (Varianza diversa)
3	Sì	No	t-test (Varianza diversa)
4	Sì	Sì	t-test (Varianza uguale)
5	Sì	Sì	t-test (Varianza uguale)
6	Sì	No	t-test (Varianza diversa)
7	Sì	No	t-test (Varianza diversa)
8	Sì	No	t-test (Varianza diversa)
9	Sì	No	t-test (Varianza diversa)
10	No	-	rank-sum

Table 19: Validazione dei dati.

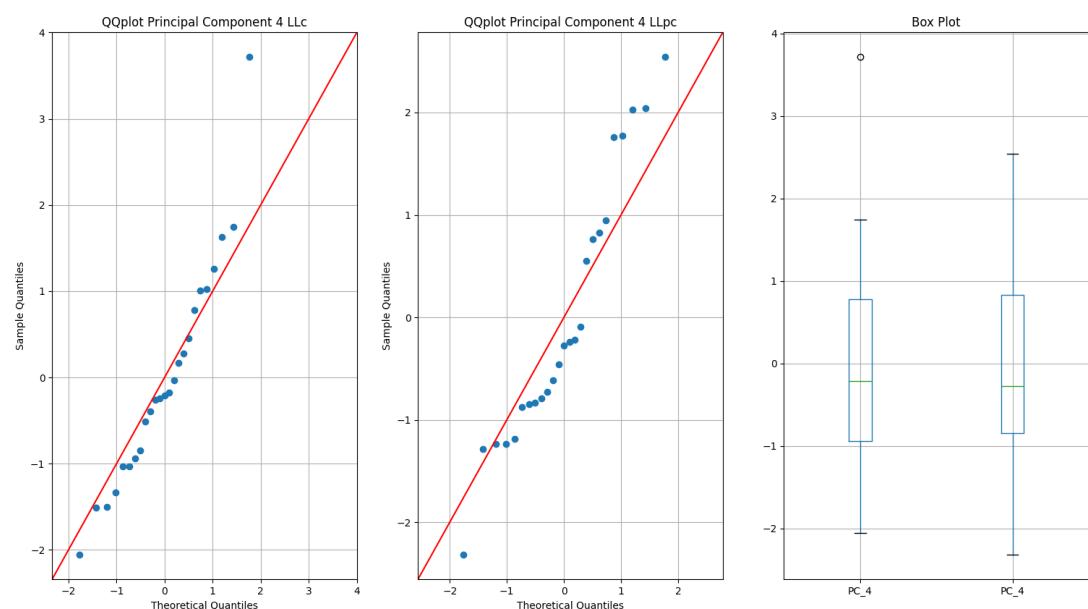


Figure 27: Componente Principale 4.

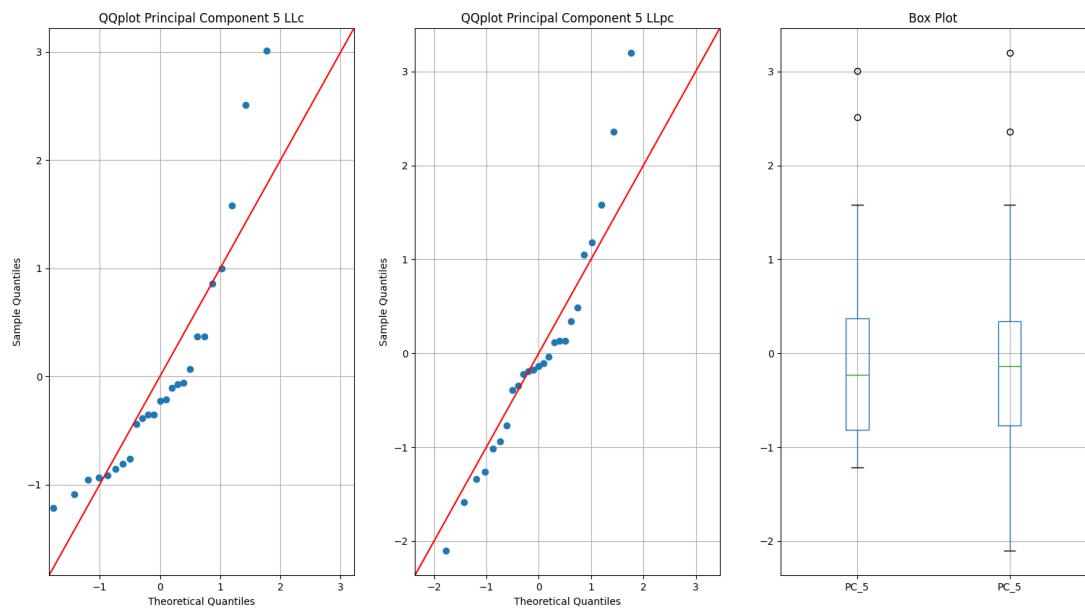


Figure 28: Componente Principale 5.

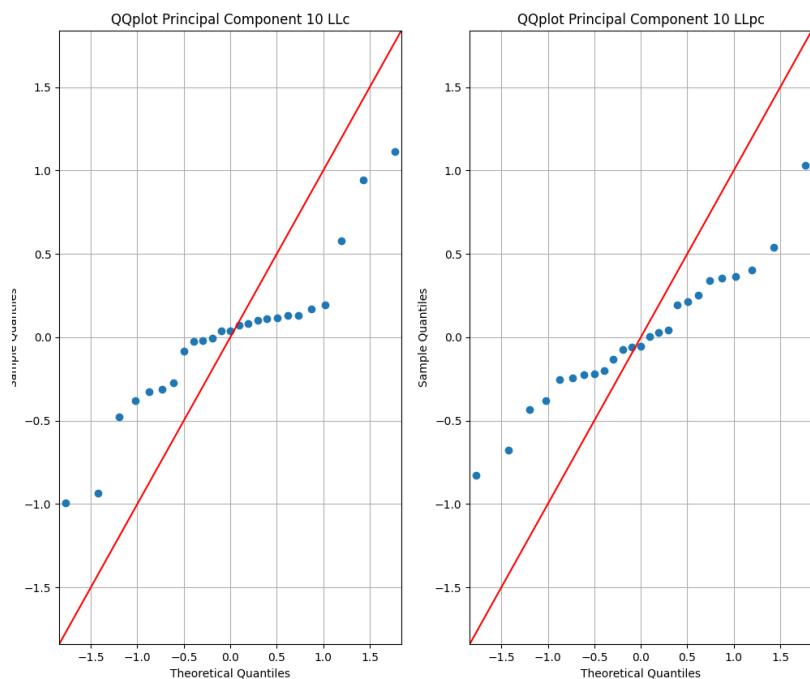


Figure 29: Componente Principale 10.

Di seguito i risultati dei test:

```
*****
Componente principale 1
Non normali. Wilcoxon rank-sum test:
H0 non rigettata, p-value: 0.7784486169673078
*****
Componente principale 2
Normali.
Stessa diversa. Two-sample t-test con varianza diversa:
H0 non rigettata, p-value: 0.9999999999999999
*****
Componente principale 3
Normali.
Stessa diversa. Two-sample t-test con varianza diversa:
H0 non rigettata, p-value: 0.9999999999999998
*****
Componente principale 4
Normali.
Stessa varianza. Two-sample t-test con varianza uguale:
H0 non rigettata, p-value: 1.0
*****
Componente principale 5
Normali.
Stessa varianza. Two-sample t-test con varianza uguale:
H0 non rigettata, p-value: 0.9999999999999997
*****
Componente principale 6
Normali.
Stessa diversa. Two-sample t-test con varianza diversa:
H0 non rigettata, p-value: 0.9999999999999998
*****
Componente principale 7
Normali.
Stessa diversa. Two-sample t-test con varianza diversa:
H0 non rigettata, p-value: 0.9999999999999998
*****
Componente principale 8
Normali.
Stessa diversa. Two-sample t-test con varianza diversa:
H0 non rigettata, p-value: 0.9999999999999998
*****
Componente principale 9
Normali.
Stessa diversa. Two-sample t-test con varianza diversa:
H0 non rigettata, p-value: 0.9999999999999998
*****
Componente principale 10
Non normali. Wilcoxon rank-sum test:
H0 non rigettata, p-value: 0.9613120861932752
```

Figure 30: **Risultati del test di ipotesi.**

In definitiva, non rigettando l'ipotesi nulla, non è possibile affermare che il Workload Sintetico generato a partire dal Workload Reale, produce effetti statisticamente differenti rispetto a quello reale, sul sistema in analisi.

4.4 DESIGN OF EXPERIMENT

Il **Design of Experiment (DoE)** è un metodo statistico utilizzato per identificare i fattori che influiscono su una determinata risposta (output del sistema) e di determinare come i fattori in input interagiscono tra loro. Una corretta analisi degli esperimenti aiuta anche a separare gli effetti dei vari fattori che potrebbero influenzare le prestazioni, determinando se un fattore è significativo o/e importante oppure se le differenze osservate dipendono semplicemente da variazioni casuali dovute ad errori di misura e/o parametri non controllabili.

Di seguito la configurazione del DoE (**Full Factorial Design**).

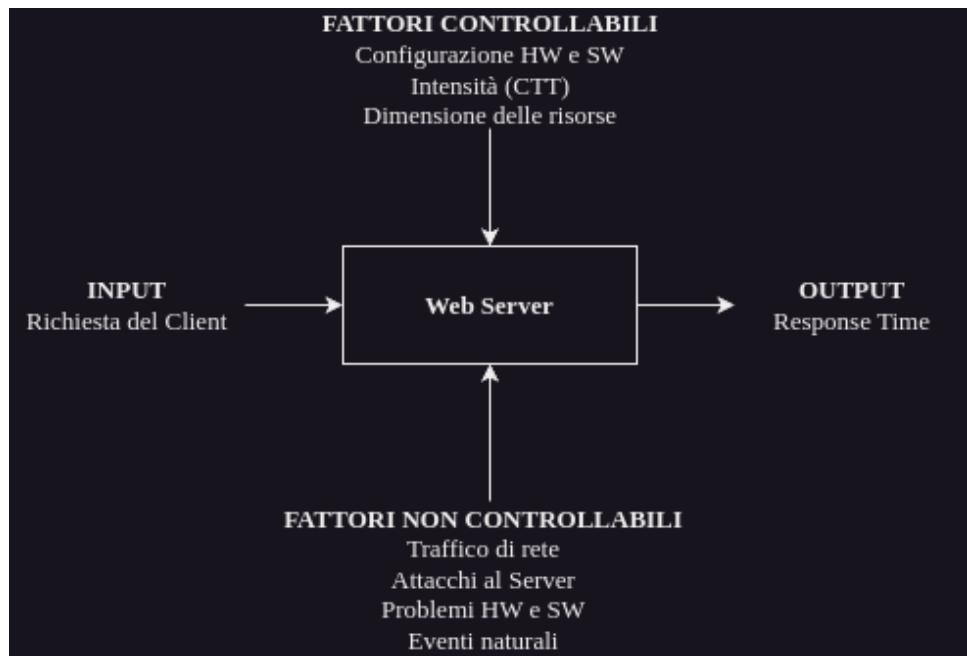


Figure 31: Design of Experiment.

I fattori presi in considerazione, sono l'**Intensità** e la **Dimensione delle risorse** richieste, ognuno con 3 livelli:

- **Intensità:**

- Livello 1: 25% della Usable Capacity (950 CTT)
- Livello 2: 50% della Usable Capacity (1900 CTT)
- Livello 3: 75% della Usable Capacity (2850 CTT)

- **Dimensione Risorsa:**

- Livello 1: Small (98 KB)
- Livello 2: Medium (1.2 MB)
- Livello 3: Large (12 MB)

Come detto in precedenza è stato effettuato un **Full Factorial Design**, e per ogni combinazione dei fattori sono stati eseguiti 5 esperimenti, per un totale di **45 esperimenti** (3 livelli * 3 livelli * 5 ripetizioni), ognuno dalla durata di 1 minuto. La risposta di interesse è il **Response Time** (calcolato come la media dell'Elapsed Time). L'obiettivo è determinare l'influenza di tali fattori sulla risposta di interesse. In particolare si vuole verificare l'importanza e la significatività dei fattori. Di seguito i risultati di alcuni esperimenti.

Risorsa	Intensità	Ripetizione	Response Time
Small	950	1	3.065
Small	950	2	2.518
Small	950	3	3.473
Small	950	4	2.370
Small	950	5	3.396
Small	1900	1	2.063
Small	1900	2	2.576
Small	1900	3	2.095
Small	1900	4	2.305
Small	1900	5	1.940
Small	2850	1	2.373
Small	2850	2	2.218
Small	2850	3	2.548
Small	2850	4	1.631
Small	2850	5	2.468
Medium	950	1	8.790
Medium	950	2	12.850
Medium	950	3	11.359
Medium	950	4	11.709
Medium	950	5	12.493
Medium	1900	1	12.504
Medium	1900	2	11.990
Medium	1900	3	10.669
Medium	1900	4	10.381
Medium	1900	5	9.195
Medium	2850	1	10.972
Medium	2850	2	12.833
Medium	2850	3	11.611
Medium	2850	4	8.377
Medium	2850	5	10.922
Large	950	1	122.458
Large	950	2	149.590
Large	950	3	112.352
Large	950	4	153.365
Large	950	5	357.577
Large	1900	1	1259.550
Large	1900	2	1305.834
			1218.155

Figure 32: **Piano fattoriale.**

4.4.1 ANALISI DELL'IMPORTANZA

In questa sezione verrà analizzata l'**importanza** di ciascun fattore. Un fattore è importante se esprime una buona percentuale di variazione rispetto alla variazione totale del sistema.

Dopo aver calcolato l'effetto di ciascun fattore, è stata calcolata la *Sum of Squares* totale, quella di ogni fattore, della loro interazione e degli errori. L'importanza di un fattore è poi calcolata come il rapporto tra la *Sum of Squares* del fattore e quella totale.

	DF	SS	Importanza
Origine			
Modello	44	12447248.287	1.000
Errore	36	104449.420	0.008
Risorsa	2	8263196.469	0.664
Intensità	2	1356823.776	0.109
IntensitàxRisorsa	4	2722778.621	0.219

Figure 33: **Analisi dell'importanza.**

Dalla figura è possibile osservare che il fattore con importanza maggiore è la **Dimensione della risorsa** che spiega il 66.4% della varianza totale, il fattore **Intensità** spiega il 10.9% della variazione, mentre il fattore d'interazione il 21.9%. I termini di errore spiegano all'incirca il 0.8% della variazione complessiva.

4.4.2 ANALISI DELLA SIGNIFICATIVITÀ

In questa sezione verrà analizzata la **significatività** di ciascun fattore. Un fattore è significativo se esprime una buona percentuale di variazione rispetto a quella relativa agli errori del medesimo fattore. Un fattore è significativo, quindi, se i suoi effetti sul sistema non sono dovuti alla casualità. Per studiare la significatività dei fattori considerati si applica il metodo **ANOVA (ANalysis Of VAriance)**. Il metodo da applicare dipende da alcune caratteristiche della distribuzione degli errori, in particolare la normalità e l'omoschedasticità.

Normale	Omoschedasticità	Test
Sì	Sì	F-test
No	Sì	Kruskal-Wallis test
Sì	No	Welch's Test
No	No	Kruskal-Wallis test o Friedman test

Table 20: **Scelta del test.**

Per verificare la normalità degli errori si ricorre ad un test visivo (plot quantile-quantile) e ad un test non parametrico (Test di *Shapiro-Wilk*).

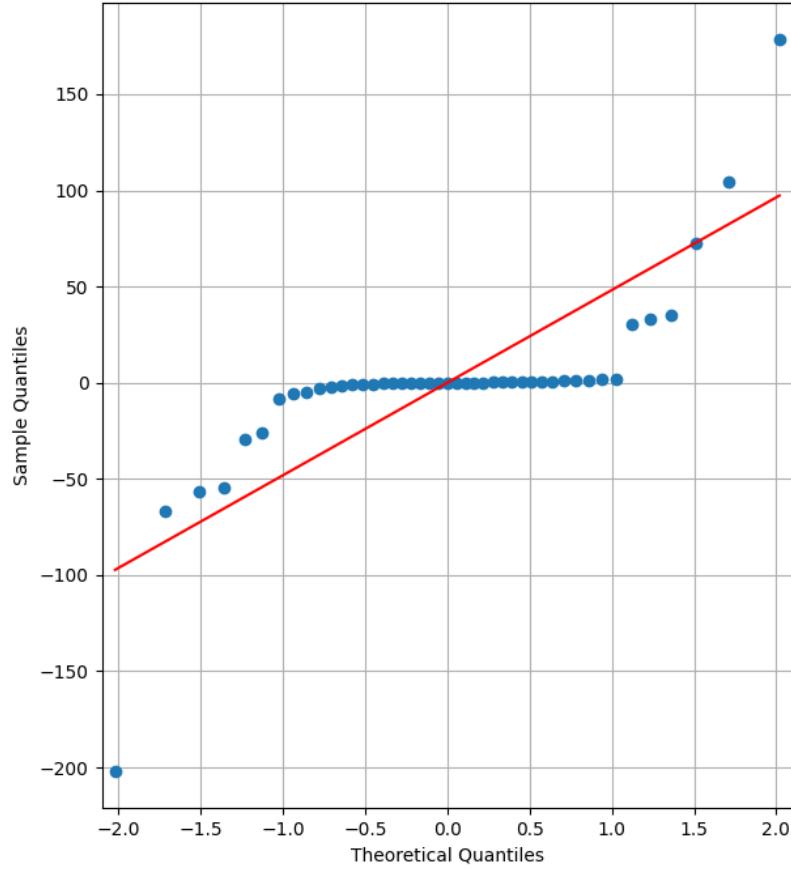


Figure 34: Plot quantile-quantile dei residui.

```
Shapiro-Wilk test p-value: 4.554813415325043e-09
```

Figure 35: Test di Shapiro-Wilk.

Sia dal test visivo, che dal test non parametrico, possiamo osservare che i dati non sono distribuiti normalmente. In particolare, il *pvalue* del test è molto basso, viene dunque rigettata l'ipotesi nulla (H_0 : i dati provengono da una distribuzione normale).

Per verificare l'omoschedasticità degli errori si ricorre ad un test visivo e ad un test non parametrico (Test di *O'Brien*).

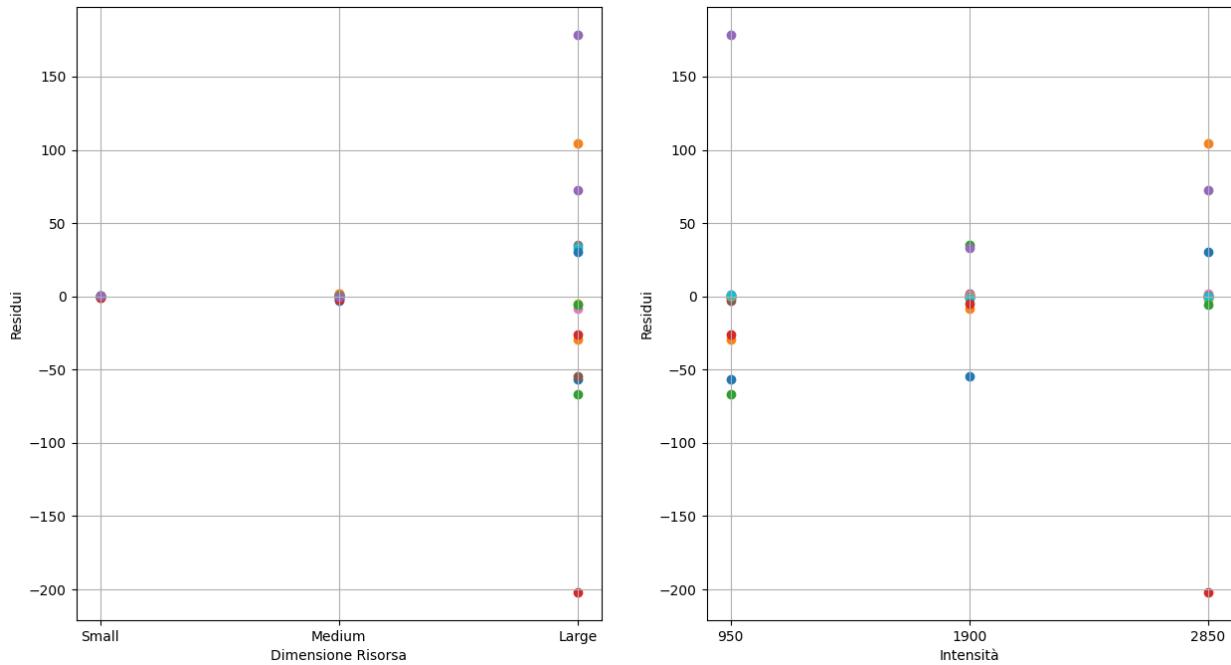


Figure 36: Valutazione omoschedasticità.

```
O'Brien test per Risorsa, p-value: 0.018446089221671153
O'Brien test per Intensità, p-value: 0.5234246364197823
```

Figure 37: Test di O'Brien.

Sia dal test visivo, che dal test non parametrico, possiamo osservare l'eteroschedasticità dei residui relativi al fattore Dimensione della risorsa, e l'omoschedasticità dei residui relativi al fattore Intensità. In particolare, nel primo caso, il *pvalue* del test è sufficientemente basso, e viene dunque rigettata l'ipotesi nulla (H_0 : i dati provengono da distribuzioni con uguale varianza).

Si procede dunque con il test di **Kruskal-Wallis** per verificare quale fattore sia statisticamente significativo.

```
Kruskal-Wallis test per Risorsa, p-value: 1.3110307577805143e-08
Kruskal-Wallis test per Intensità, p-value: 0.9999941883103266
```

Figure 38: Test di Kruskall-Wallis.

Dai risultati dei test è possibile osservare che l'unico fattore significativo è la **Dimensione della risorsa**. Questo perchè il *pvalue* del test per la Dimensione della risorsa è molto basso rigettando l'ipotesi della non significatività statistica. Mentre l'Intensità delle risorse risulta non statisticamente significativo in quanto il *pvalue* indica che l'ipotesi nulla non è stata rigettata.

5 REGRESSIONE

Un **modello regressivo lineare** è una tecnica che cerca di stabilire una relazione tra una variabile dipendente (risposta o stimata) e una o più variabili indipendenti (di predizione). La relazione viene stabilita attraverso una funzione lineare che descrive il modello. Di seguito verranno analizzati diversi dataset utilizzando dei modelli regressivi lineari per individuare eventuali trend.

5.1 DATASET EXP1

Il dataset presenta le seguenti colonne:

- Variabili di risposta:
 - *Nmail*
 - *Byte rec*
 - *Byte sent*
- Variabili di predizione:
 - *Observation*

Viene applicato il modello regressivo lineare la cui risposta è una funzione lineare del predittore. Nella stima viene indicato l' R^2 , ovvero il coefficiente di determinazione, che indica la percentuale di varianza spiegata dal modello rispetto a quella totale. Di seguito i risultati per il dataset corrente:

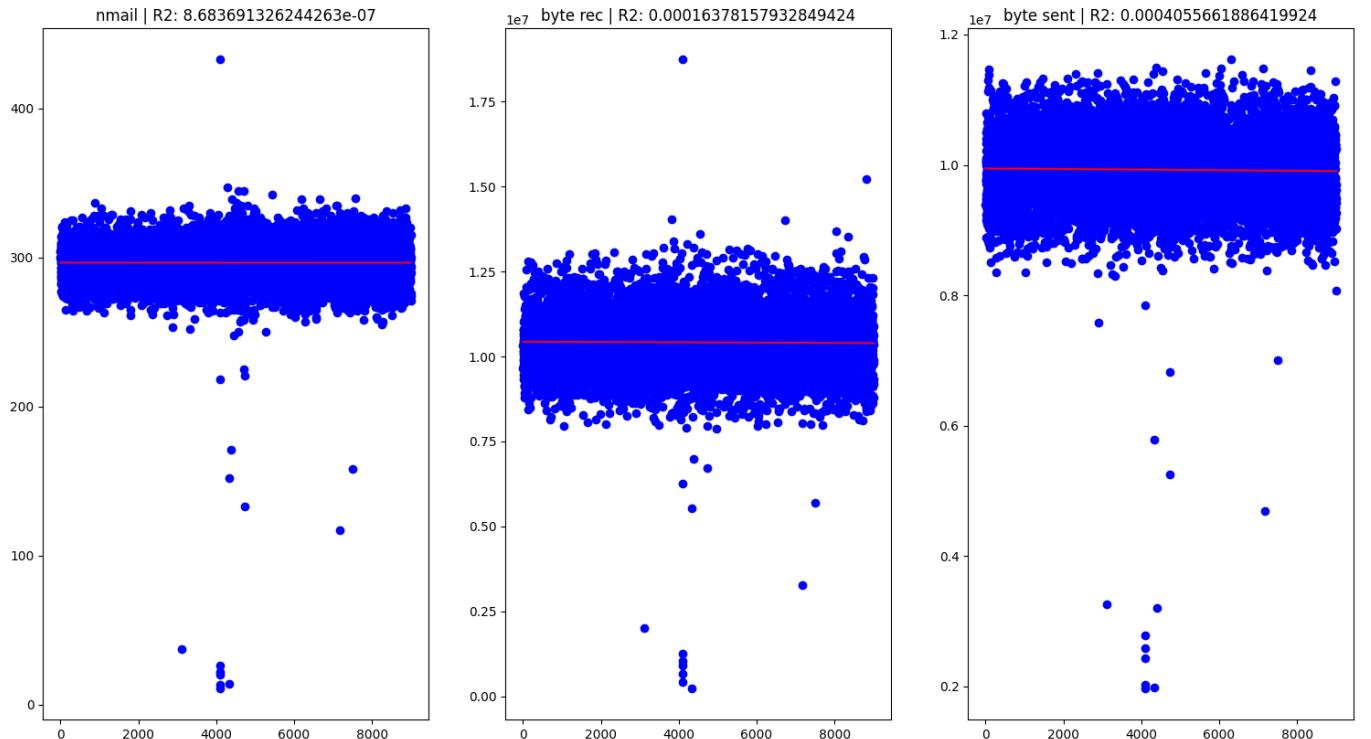


Figure 39: Regressione lineare e R^2 .

Come è possibile osservare dalla figura, l' R^2 assume un valore molto basso a dimostrazione del fatto che il modello regressivo lineare usato ha uno scarso adattamento ai dati, e dunque, le prestazioni predittive sono insufficienti.

Successivamente si procede a verificare la significatività dei parametri della retta regressiva, tramite un opportuno test statistico. Per scegliere il test da utilizzare si verifica la normalità dei residui delle rispettive variabili:

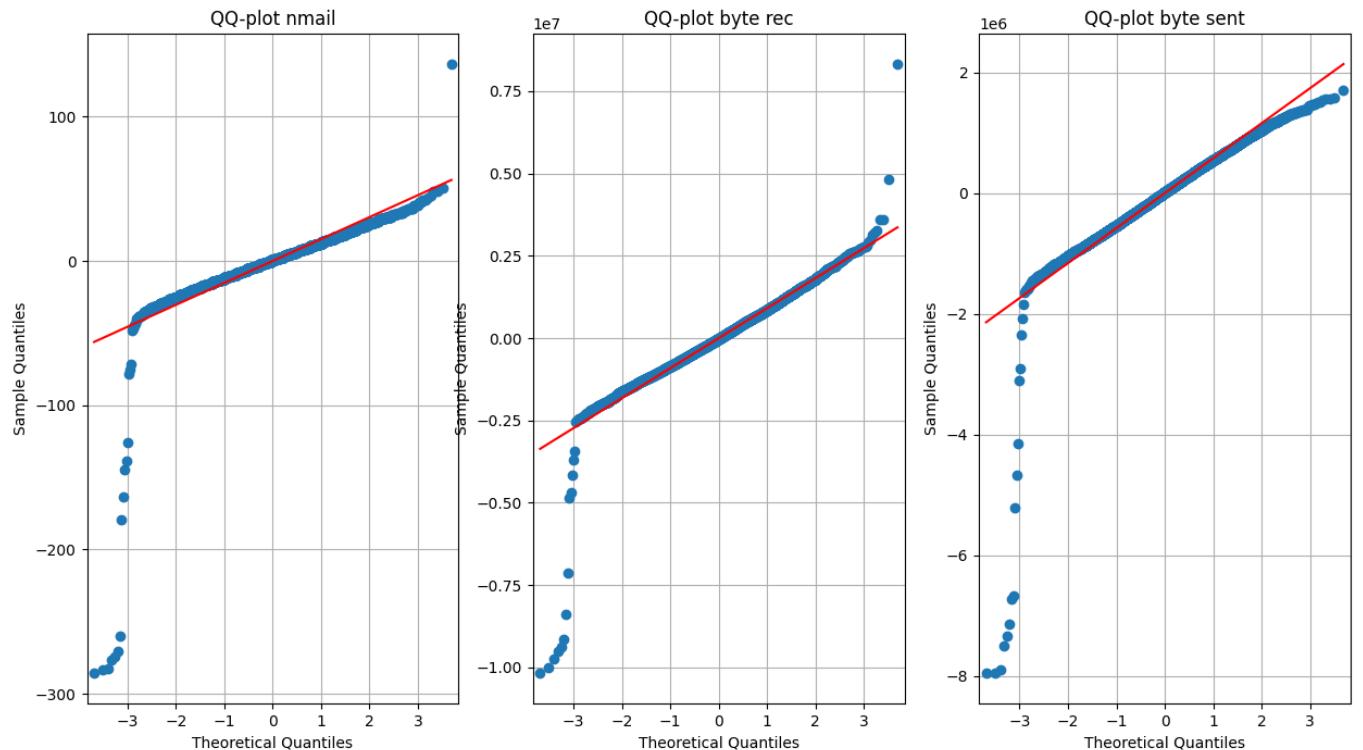


Figure 40: Plot quantile-quantile dei residui.

In tutti e 3 i casi le distribuzioni non sono normali, si procede dunque con il test non parametrico di Mann-Kendall:

```
Test di Mann-Kendall (nmail), pvalue: 0.716381531864104
Test di Mann-Kendall (byte rec), pvalue: 0.20018889443014742
Test di Mann-Kendall (byte sent), pvalue: 0.043498948902028704
```

Figure 41: Test di Mann-Kendall.

Dai risultati ottenuti è possibile osservare che solo per la variabile *byte sent* è presente un trend ($pvalue < 0.05$, H_0 : non è presente un trend monotono).

Si valutano dunque, il coefficiente angolare e l'intercetta della retta di regressione utilizzando la procedura di Theil-Sen. Questa procedura è una regressione lineare robusta progettata per fornire risultati più precisi e affidabili in presenza di outlier nei dati.

```
Regressione Lineare Robusta (Theil e Sen) per byte sent
Slope: -4.514783992985356
Intercept: 9959247.527968435
Low Slope: -8.891050583657588
High Slope: -0.13067767792081614
```

Figure 42: Procedura di Theil e Sen.

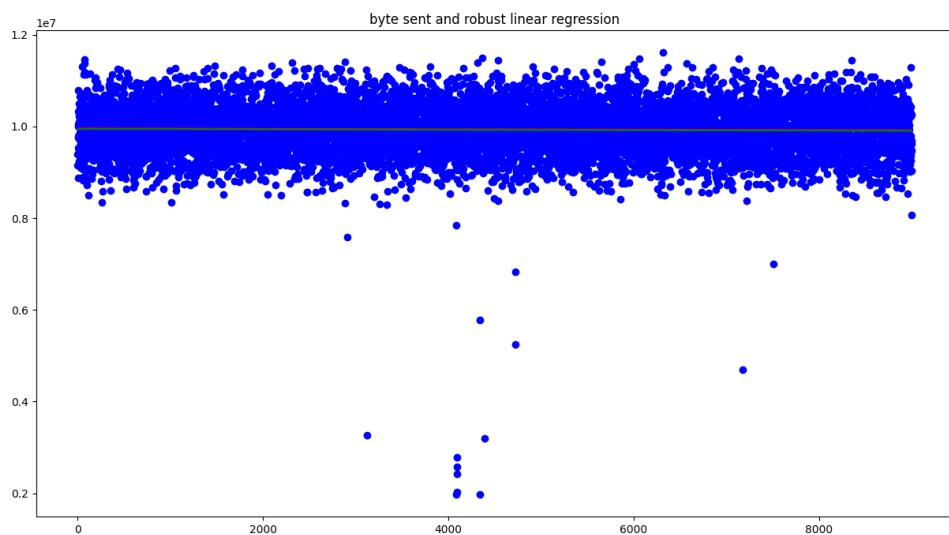


Figure 43: **Regessione robusta.**

5.2 DATASET EXP2

Il dataset presenta le seguenti colonne:

- Variabili di risposta:

- N_{mail}
- $Byte\ rec$
- $Byte\ sent$

- Variabili di predizione:

- $Observation$

Viene applicato il modello regressivo lineare la cui risposta è una funzione lineare del predittore. Di seguito i risultati per il dataset corrente:

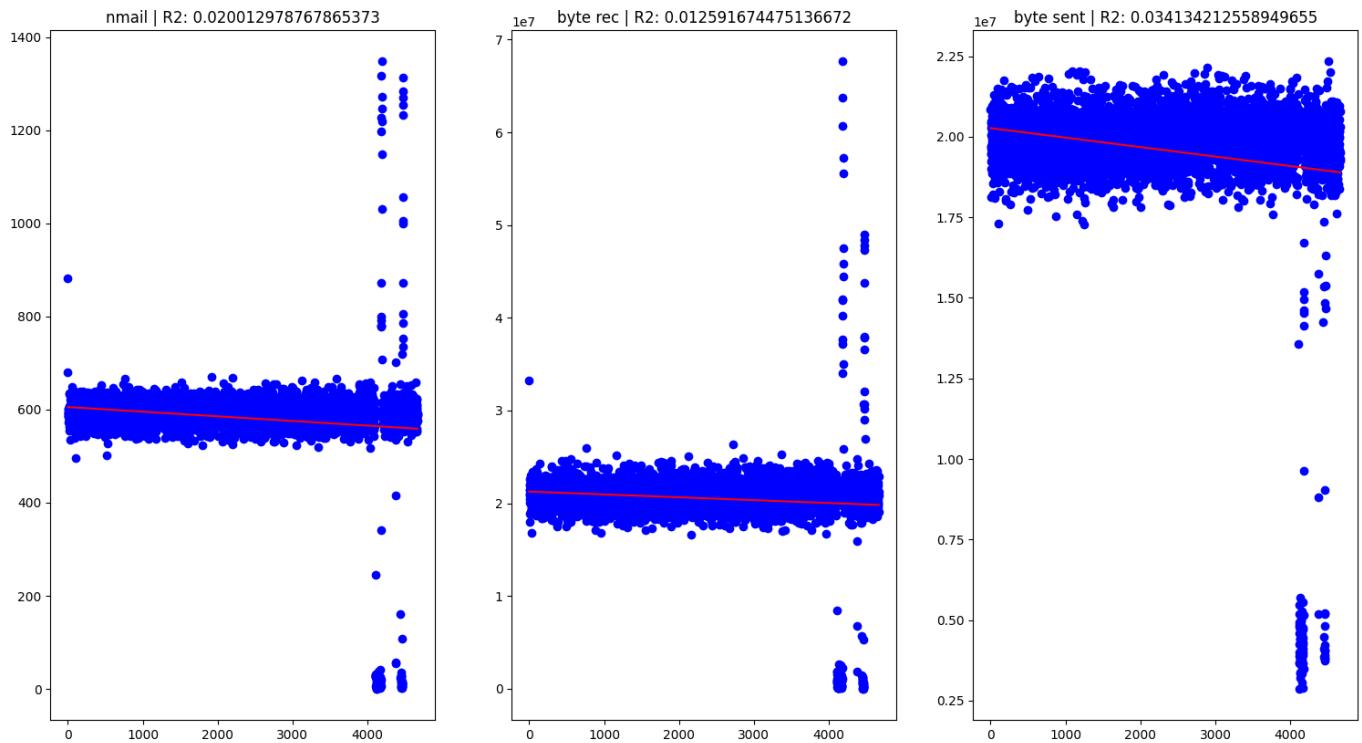


Figure 44: **Regessione lineare e R^2 .**

Come è possibile osservare dalla figura, l' R^2 assume un valore molto basso a dimostrazione del fatto che il modello regressivo lineare usato ha uno scarso adattamento ai dati, e dunque, le prestazioni predittive sono insufficienti.

Successivamente si procede a verificare la significatività dei parametri della retta regressiva, tramite un opportuno test statistico. Per scegliere il test da utilizzare si verifica la normalità dei residui delle rispettive variabili:

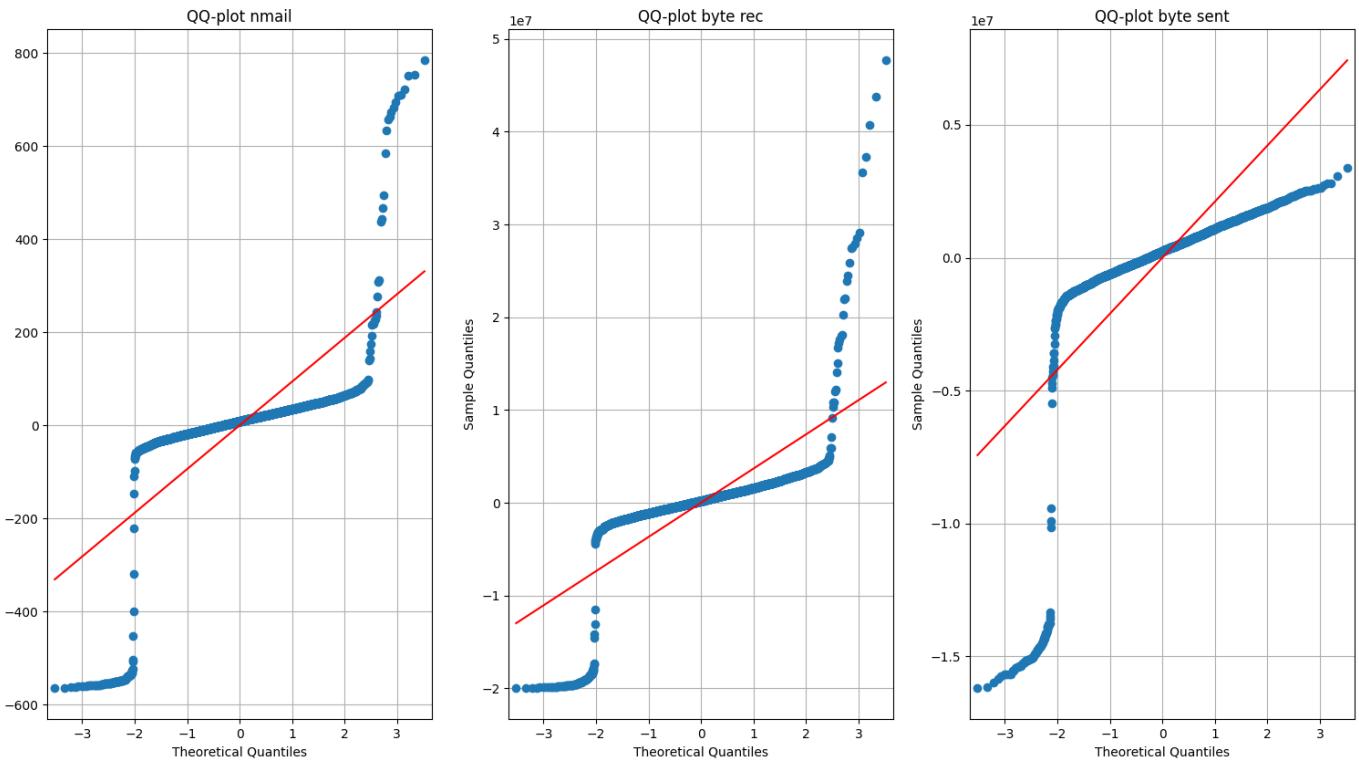


Figure 45: Plot quantile-quantile dei residui.

In tutti e 3 i casi le distribuzioni non sono normali, si procede dunque con il test non parametrico di Mann-Kendall:

```
Test di Mann-Kendall (nmail), pvalue: 0.013846454604669158
Test di Mann-Kendall (byte rec), pvalue: 0.001217360770161631
Test di Mann-Kendall (byte sent), pvalue: 5.847156844915903e-05
```

Figure 46: Test di Mann-Kendall.

Dai risultati ottenuti è possibile osservare che per tutte le variabili è presente un trend ($pvalue < 0.05$, H_0 : non è presente un trend monotono).

Si valutano dunque, il coefficiente angolare e l'intercetta della retta di regressione utilizzando la procedura di Theil-Sen.

```

Regressione Lineare Robusta (Theil e Sen) per nmail
Slope: -0.0005973715651135006
Intercept: 592.3936678614098
Low Slope: -0.0011248593925759281
High Slope: 0.0
Regressione Lineare Robusta (Theil e Sen) per byte rec
Slope: -48.29360967184802
Intercept: 20890131.991364423
Low Slope: -77.54773269689737
High Slope: -18.993146773272414
Regressione Lineare Robusta (Theil e Sen) per byte sent
Slope: -34.06305208650625
Intercept: 19903064.10051782
Low Slope: -50.672672672672675
High Slope: -17.374087591240876

```

Figure 47: Procedura di Theil e Sen.

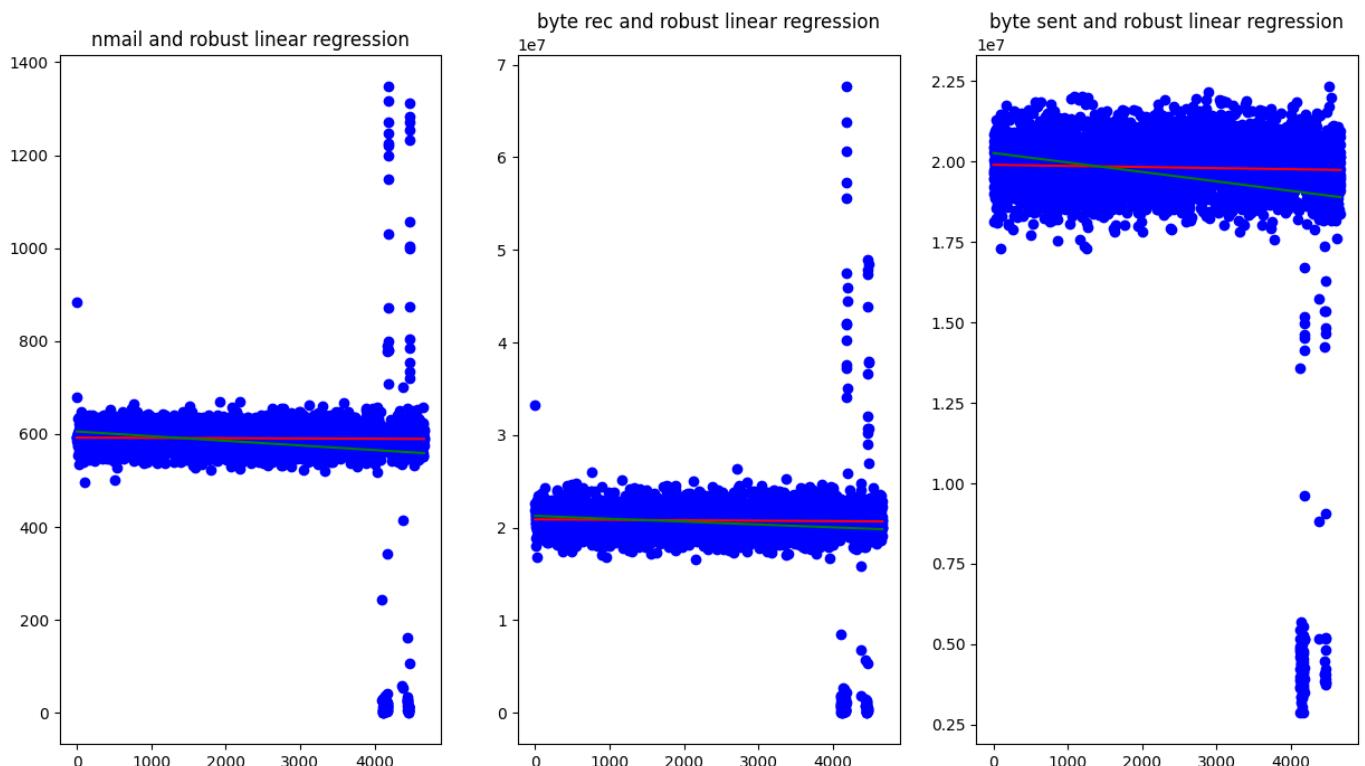


Figure 48: Regressione robusta.

5.3 DATASET OS1

Il dataset presenta le seguenti colonne:

- Variabili di risposta:
 - $VmSize$
 - $VmData$
 - RSS
 - $Byte_letti_IO$
 - $Byte_letti_IO1$

- Variabili di predizione:
 - $Time$

Viene applicato il modello regressivo lineare la cui risposta è una funzione lineare del predittore. Nella stima viene indicato l' R^2 , ovvero il coefficiente di determinazione, che indica la percentuale di varianza spiegata dal modello rispetto a quella totale. Di seguito i risultati per il dataset corrente:

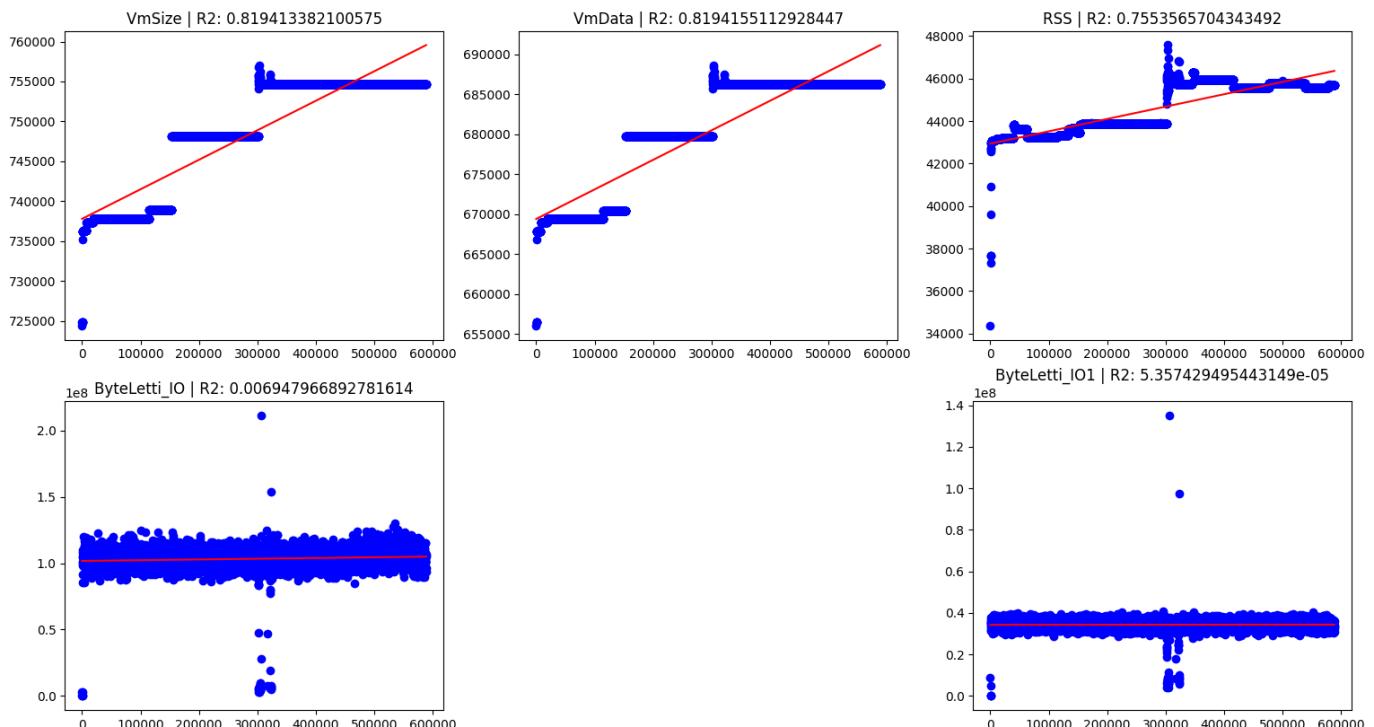


Figure 49: Regressione lineare e R^2 .

Come è possibile osservare dalla figura, l' R^2 assume un valore molto basso negli ultimi 2 casi. Successivamente si procede a verificare la significatività dei parametri della retta regressiva, tramite un opportuno test statistico. Per scegliere il test da utilizzare si verifica la normalità dei residui delle rispettive variabili:

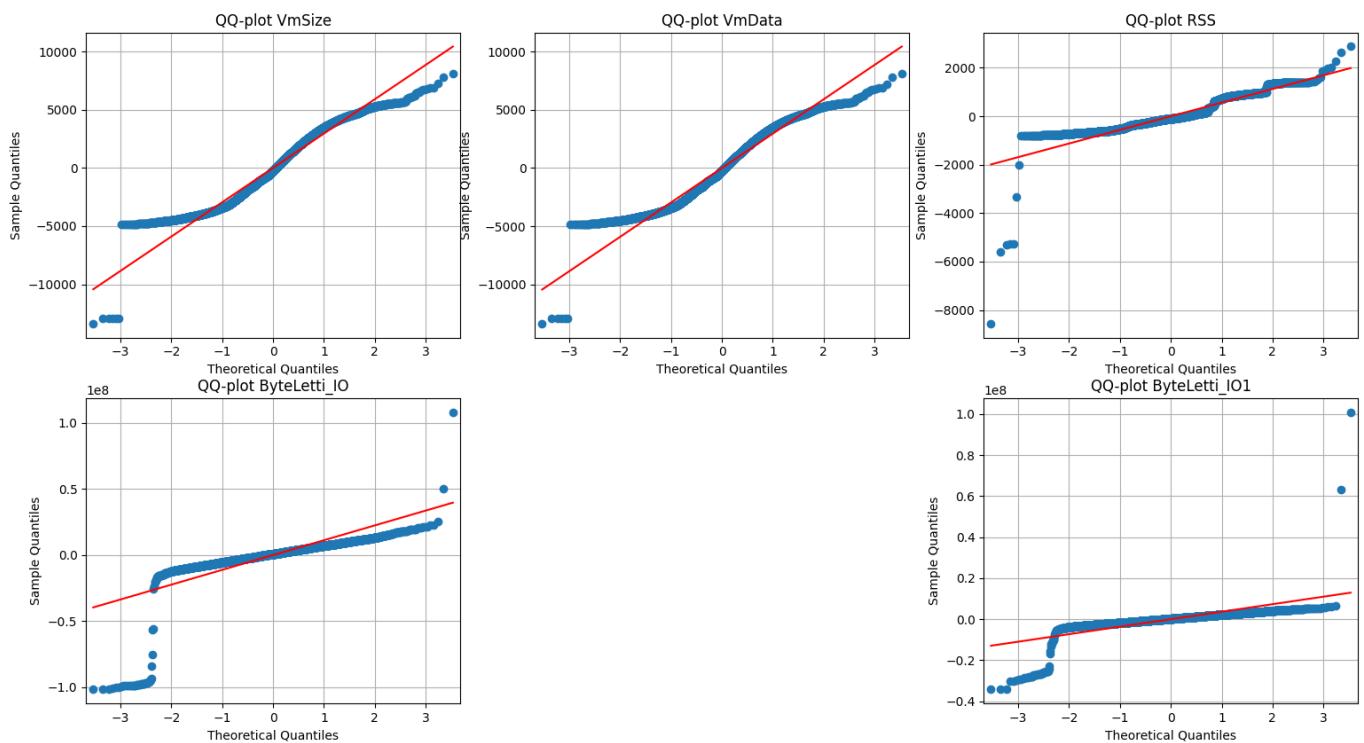


Figure 50: Plot quantile-quantile dei residui.

In tutti e 5 i casi le distribuzioni non sono normali, si procede dunque con il test non parametrico di Mann-Kendall:

```
Test di Mann-Kendall (VmSize), pvalue: 0.0
Test di Mann-Kendall (VmData), pvalue: 0.0
Test di Mann-Kendall (RSS), pvalue: 0.0
Test di Mann-Kendall (ByteLetti_IO), pvalue: 5.361557518034273e-18
Test di Mann-Kendall (ByteLetti_IO1), pvalue: 0.8554762957272513
```

Figure 51: Test di Mann-Kendall.

Dai risultati ottenuti è possibile osservare che solo per la variabile *Byte_letti_IO1* non possiamo rigettare l'ipotesi nulla ($pvalue < 0.05$, H_0 : non è presente un trend monotono).

Si valutano dunque, il coefficiente angolare e l'intercetta della retta di regressione utilizzando la procedura di Theil-Sen. Questa procedura è una regressione lineare robusta progettata per fornire risultati più precisi e affidabili in presenza di outlier nei dati.

```

Regressione Lineare Robusta (Theil e Sen) per VmSize
Slope: 0.03157224220623501
Intercept: 738854.3947841726
Low Slope: 0.030715952172645087
High Slope: 0.03232097662647015
Regressione Lineare Robusta (Theil e Sen) per VmData
Slope: 0.03157224220623501
Intercept: 670450.3947841726
Low Slope: 0.030715952172645087
High Slope: 0.03232097662647015
Regressione Lineare Robusta (Theil e Sen) per RSS
Slope: 0.004970760233918129
Intercept: 42428.80701754386
Low Slope: 0.004876649454962708
High Slope: 0.005055788005578801
Regressione Lineare Robusta (Theil e Sen) per ByteLetti_IO
Slope: 4.737578125
Intercept: 102763229.503125
Low Slope: 3.6739155483258776
High Slope: 5.80459886547812

```

Figure 52: Procedura di Theil e Sen.

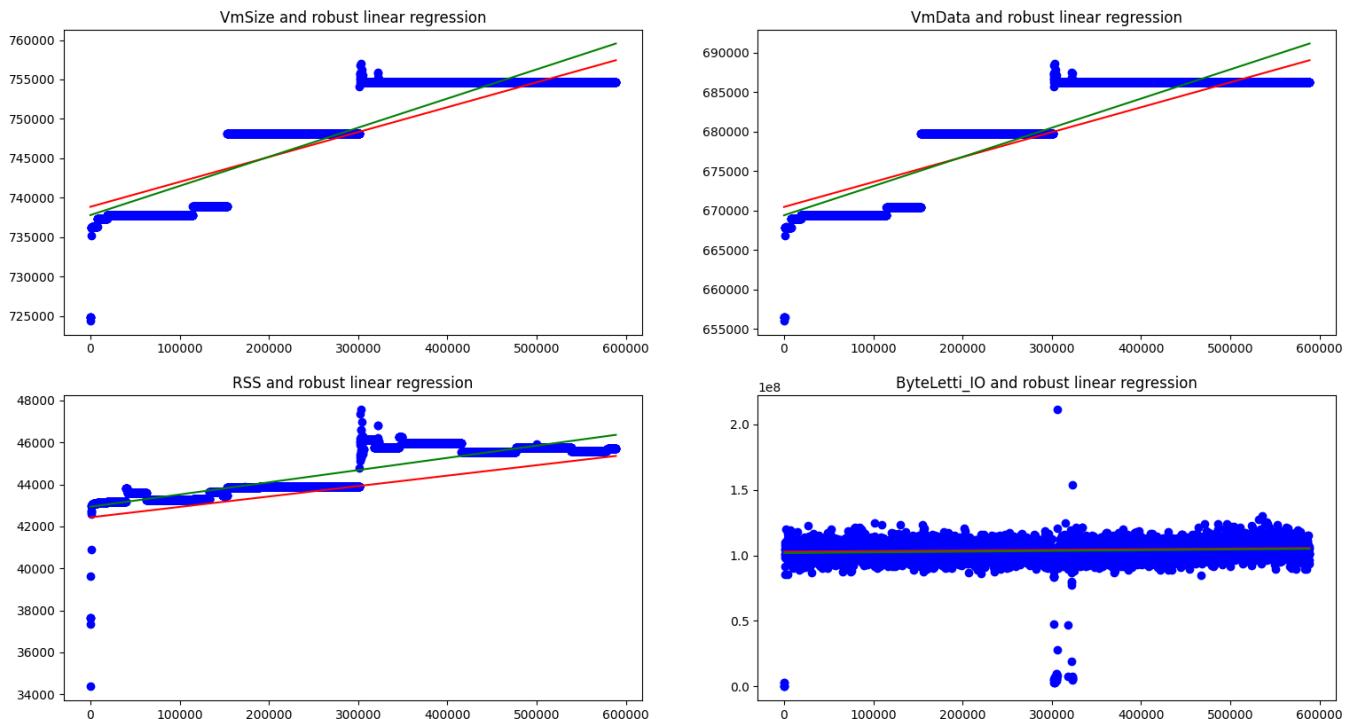


Figure 53: Regressione robusta.

5.4 DATASET OS2

Il dataset presenta le seguenti colonne:

- Variabili di risposta:
 - $VmSize$
 - $VmData$
 - RSS
 - $Byte_letti_sec$
 - $Byte_scritti_sec$

- Variabili di predizione:
 - $Time$

Viene applicato il modello regressivo lineare la cui risposta è una funzione lineare del preditore. Nella stima viene indicato l' R^2 , ovvero il coefficiente di determinazione, che indica la percentuale di varianza spiegata dal modello rispetto a quella totale. Di seguito i risultati per il dataset corrente:

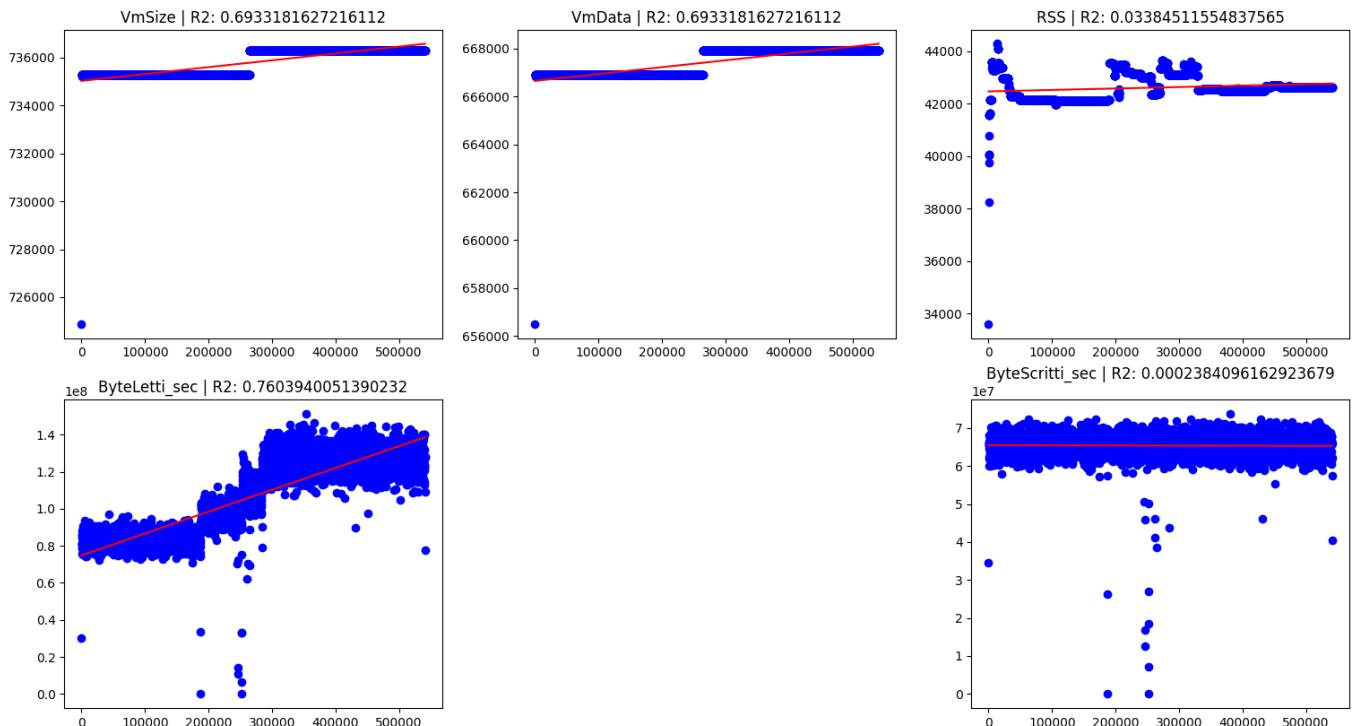


Figure 54: **Regressione lineare e R^2 .**

Come è possibile osservare dalla figura, l' R^2 assume un valore molto basso per *RSS* e *Byte_scritti_sec*. Successivamente si procede a verificare la significatività dei parametri della retta regressiva, tramite un opportuno test statistico. Per scegliere il test da utilizzare si verifica la normalità dei residui delle rispettive variabili:

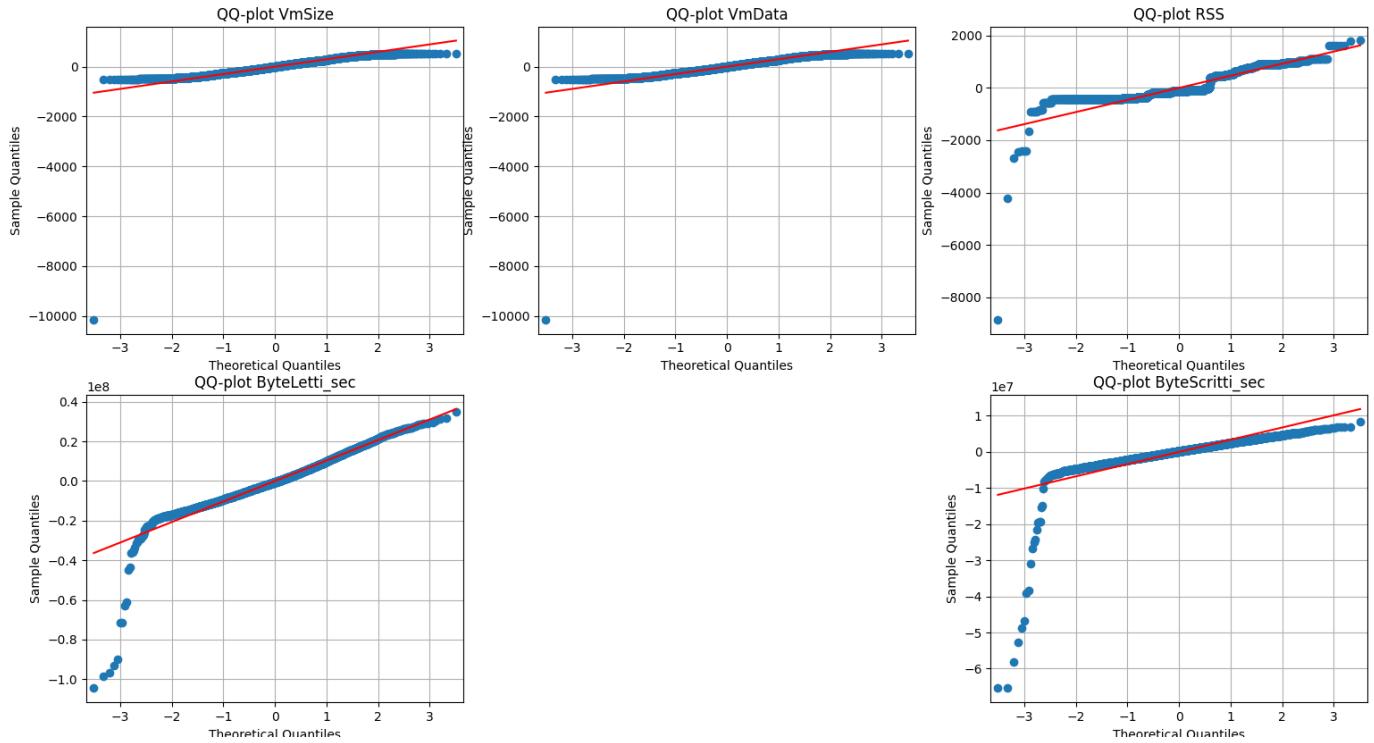


Figure 55: Plot quantile-quantile dei residui.

In tutti e 5 i casi le distribuzioni non sono normali, si procede dunque con il test non parametrico di Mann-Kendall:

```
Test di Mann-Kendall (VmSize), pvalue: 0.0
Test di Mann-Kendall (VmData), pvalue: 0.0
Test di Mann-Kendall (RSS), pvalue: 1.9089467139330274e-67
Test di Mann-Kendall (ByteLetti_sec), pvalue: 0.0
Test di Mann-Kendall (ByteScritti_sec), pvalue: 0.034196958896100235
```

Figure 56: Test di Mann-Kendall.

Dai risultati ottenuti è possibile osservare che per tutte le variabili è presente un trend ($pvalue < 0.05$, H_0 : non è presente un trend monotono).

Si valutano dunque, il coefficiente angolare e l'intercetta della retta di regressione utilizzando la procedura di Theil-Sen. Questa procedura è una regressione lineare robusta progettata per fornire risultati più precisi e affidabili in presenza di outlier nei dati.

```

Regressione Lineare Robusta (Theil e Sen) per VmSize
Slope: 0.0019128745423298214
Intercept: 735790.3761488455
Low Slope: 0.0
High Slope: 0.002085369827305311
Regressione Lineare Robusta (Theil e Sen) per VmData
Slope: 0.0019128745423298214
Intercept: 667414.3761488455
Low Slope: 0.0
High Slope: 0.002085369827305311
Regressione Lineare Robusta (Theil e Sen) per RSS
Slope: 0.0006324666198172874
Intercept: 42384.85453267744
Low Slope: 0.0005555555555555556
High Slope: 0.0007445442875481386
Regressione Lineare Robusta (Theil e Sen) per ByteLetti_sec
Slope: 114.55350983796296
Intercept: 77733384.23784722
Low Slope: 112.73359327217125
High Slope: 116.37659764826176
Regressione Lineare Robusta (Theil e Sen) per ByteScritti_sec
Slope: -0.48861262488646684
Intercept: 65693615.57629428
Low Slope: -0.9390005359656806
High Slope: -0.036319163292847505

```

Figure 57: Procedura di Theil e Sen.

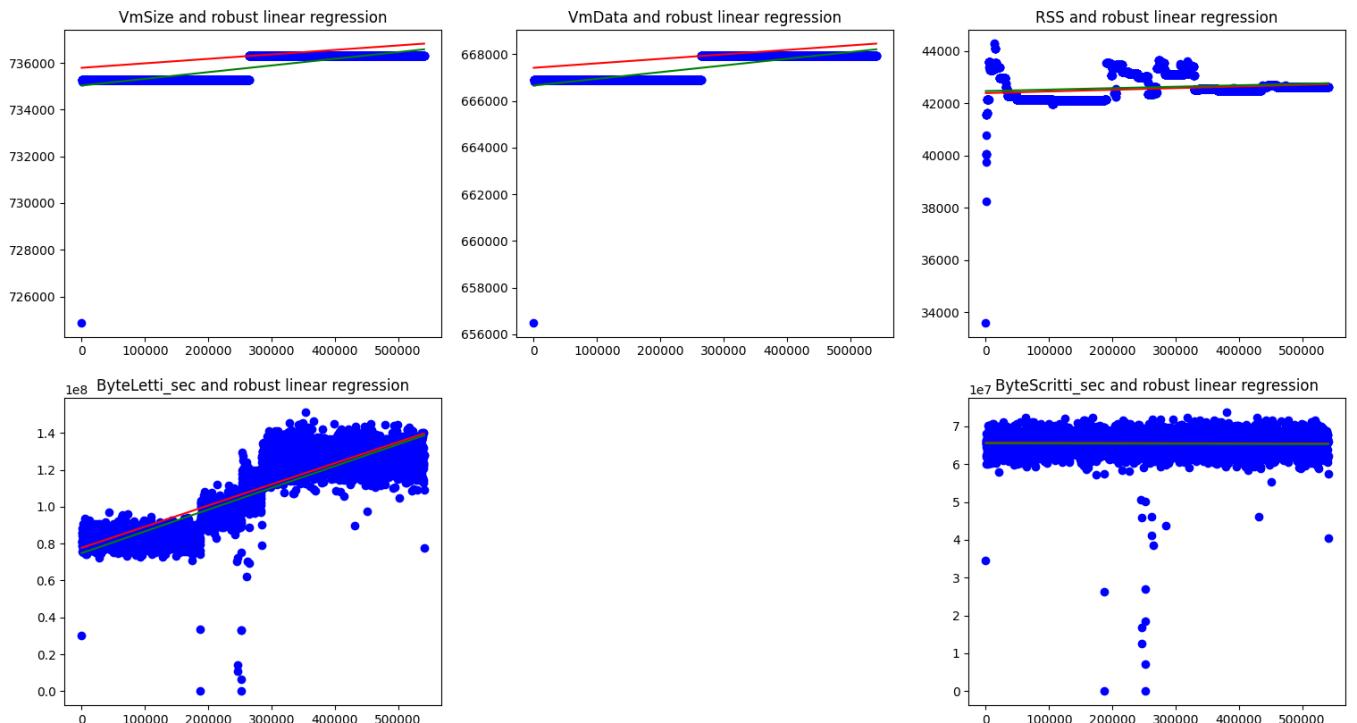


Figure 58: Regressione robusta.

5.5 DATASET OS3

Il dataset presenta le seguenti colonne:

- Variabili di risposta:
 - $VmSize$
 - $VmData$
 - RSS
 - $Byte_letti_sec$
 - $Byte_scritti_sec$

- Variabili di predizione:
 - $Time$

Viene applicato il modello regressivo lineare la cui risposta è una funzione lineare del preditore. Nella stima viene indicato l' R^2 , ovvero il coefficiente di determinazione, che indica la percentuale di varianza spiegata dal modello rispetto a quella totale. Di seguito i risultati per il dataset corrente:

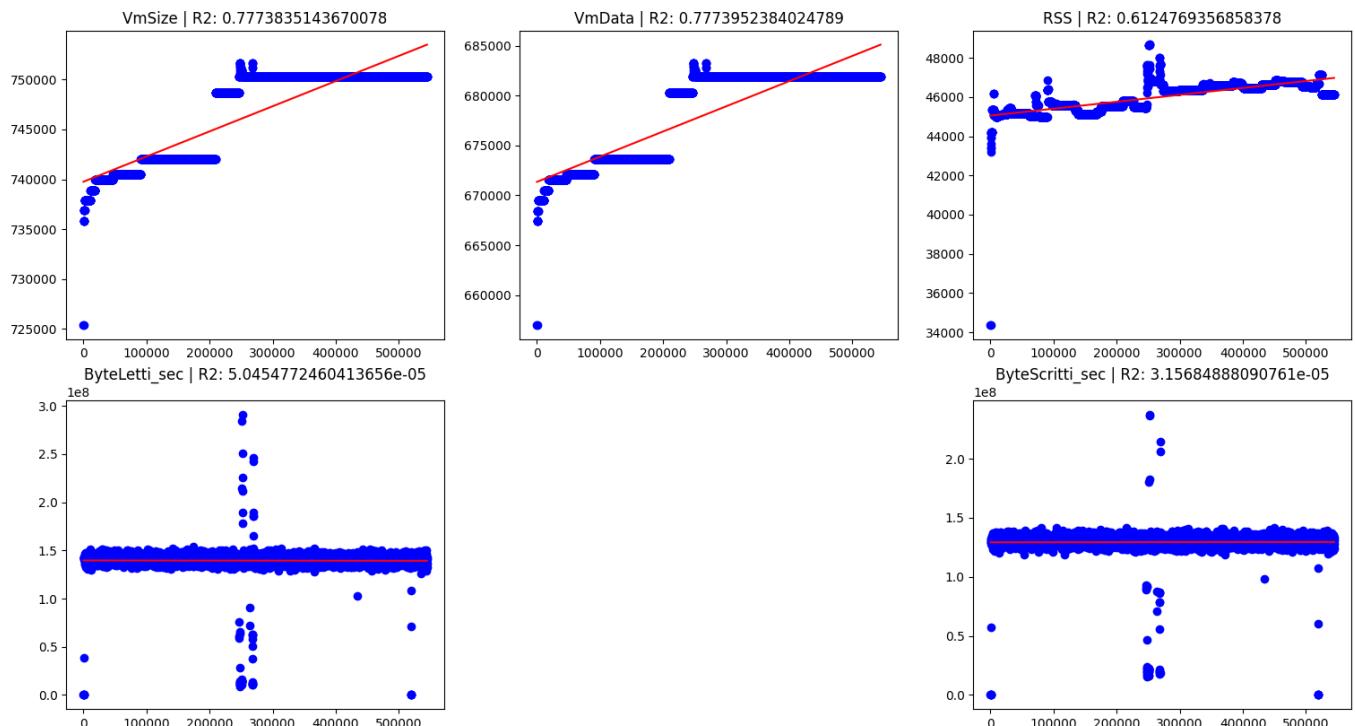


Figure 59: Regressione lineare e R^2 .

Come è possibile osservare dalla figura, l' R^2 assume un valore molto basso per *Byte_letti_sec* e *Byte_scritti_sec*.

Successivamente si procede a verificare la significatività dei parametri della retta regressiva, tramite un opportuno test statistico. Per scegliere il test da utilizzare si verifica la normalità dei residui delle rispettive variabili:

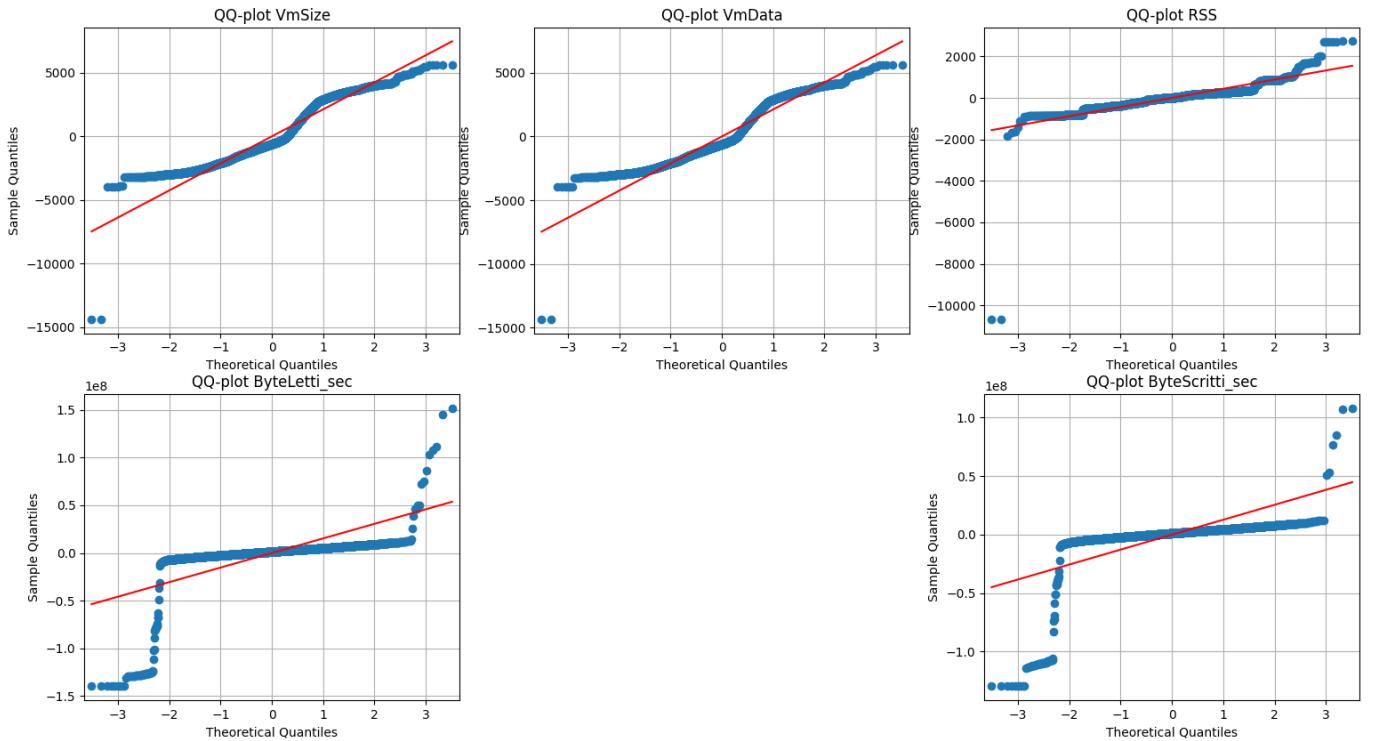


Figure 60: Plot quantile-quantile dei residui.

In tutti e 5 i casi le distribuzioni non sono normali, si procede dunque con il test non parametrico di Mann-Kendall:

```
Test di Mann-Kendall (VmSize), pvalue: 0.0
Test di Mann-Kendall (VmData), pvalue: 0.0
Test di Mann-Kendall (RSS), pvalue: 0.0
Test di Mann-Kendall (ByteLetti_sec), pvalue: 3.270057610707417e-07
Test di Mann-Kendall (ByteScritti_sec), pvalue: 0.08412024115628608
```

Figure 61: Test di Mann-Kendall.

Dai risultati ottenuti è possibile osservare che per tutte le variabili tranne *Byte_scritti_sec* è presente un trend ($pvalue < 0.05$, H_0 : non è presente un trend monotono).

Si valutano dunque, il coefficiente angolare e l'intercetta della retta di regressione utilizzando la procedura di Theil-Sen. Questa procedura è una regressione lineare robusta progettata per fornire risultati più precisi e affidabili in presenza di outlier nei dati.

```

Regressione Lineare Robusta (Theil e Sen) per VmSize
Slope: 0.021659324522760644
Intercept: 744374.5991189427
Low Slope: 0.02127364220024293
High Slope: 0.02202594299368493
Regressione Lineare Robusta (Theil e Sen) per VmData
Slope: 0.021659324522760644
Intercept: 675970.5991189427
Low Slope: 0.02127364220024293
High Slope: 0.02202594299368493
Regressione Lineare Robusta (Theil e Sen) per RSS
Slope: 0.0034579439252336447
Intercept: 45386.471028037384
Low Slope: 0.0033884297520661156
High Slope: 0.0035222894881673087
Regressione Lineare Robusta (Theil e Sen) per ByteLetti_sec
Slope: -1.891048815359477
Intercept: 141291080.77144608
Low Slope: -2.617641979071421
High Slope: -1.16775956284153

```

Figure 62: Procedura di Theil e Sen.

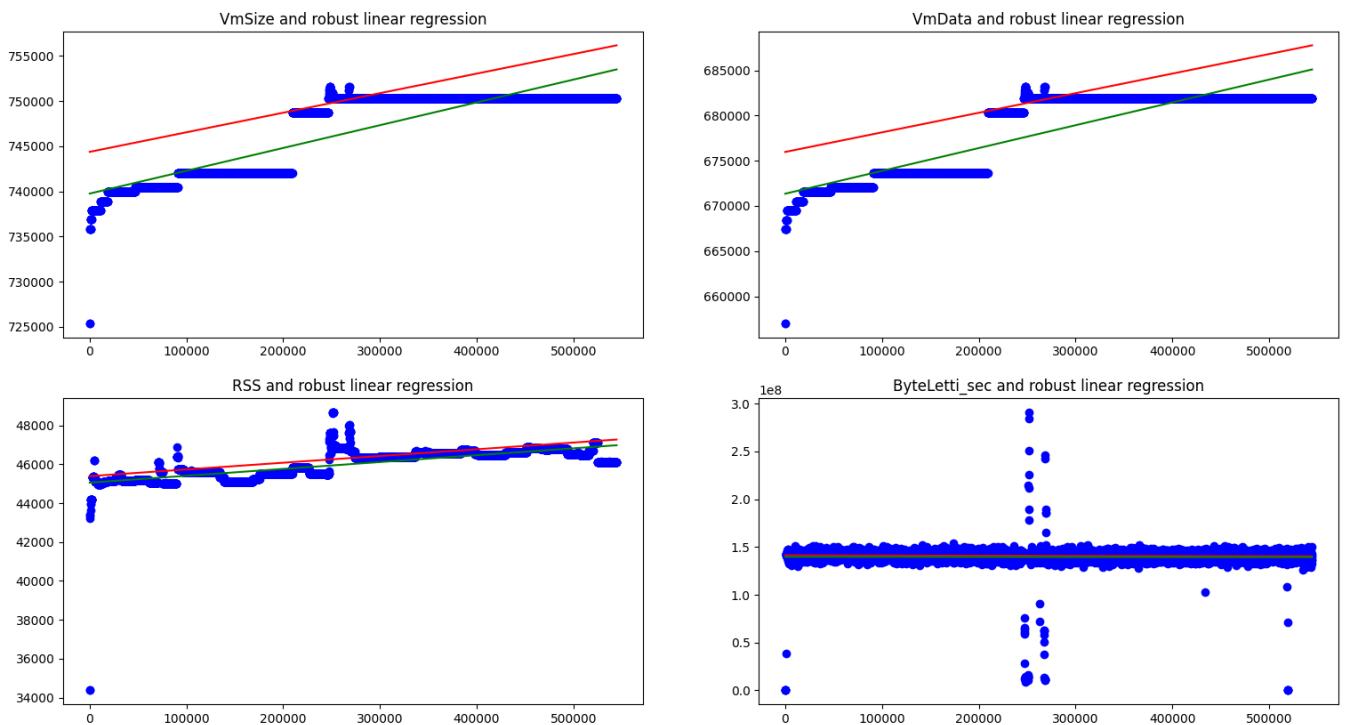


Figure 63: Regressione robusta.

Dall’analisi dei trend dei dati raccolti nei tre dataset relativamente all’occupazione di memoria di una macchina virtuale (VM), si è riscontrato che:

- Tutti i dataset presentano un trend crescente nell’utilizzo della memoria (VmSize, VmData e VmRSS), suggerendo una maggiore attività della VM, che potrebbe portare alla necessità di una maggiore quantità di memoria.
- Un trend crescente nel tasso di errore (riduzione di Byte letti sec rispetto a Byte scritti sec) potrebbe indicare problemi con la VM, come ad esempio problemi di compatibilità con il sistema operativo o problemi di configurazione; in quest’ultimo caso per effettuare delle valutazioni per prevedere andamenti futuri, è necessario prendere in considerazione il secondo dataset in quanto le variabili di risposta interessate per questa metrica presentano un trend significativo solo in quell’insieme di dati.

5.6 DATASET VMRES1

Il dataset presenta le seguenti colonne:

- Variabili di risposta:
 - *Allocated Heap*
- Variabili di predizione:
 - $T(s)$

Viene applicato il modello regressivo lineare la cui risposta è una funzione lineare del predittore. Nella stima viene indicato l' R^2 , ovvero il coefficiente di determinazione, che indica la percentuale di varianza spiegata dal modello rispetto a quella totale. Di seguito i risultati per il dataset corrente:

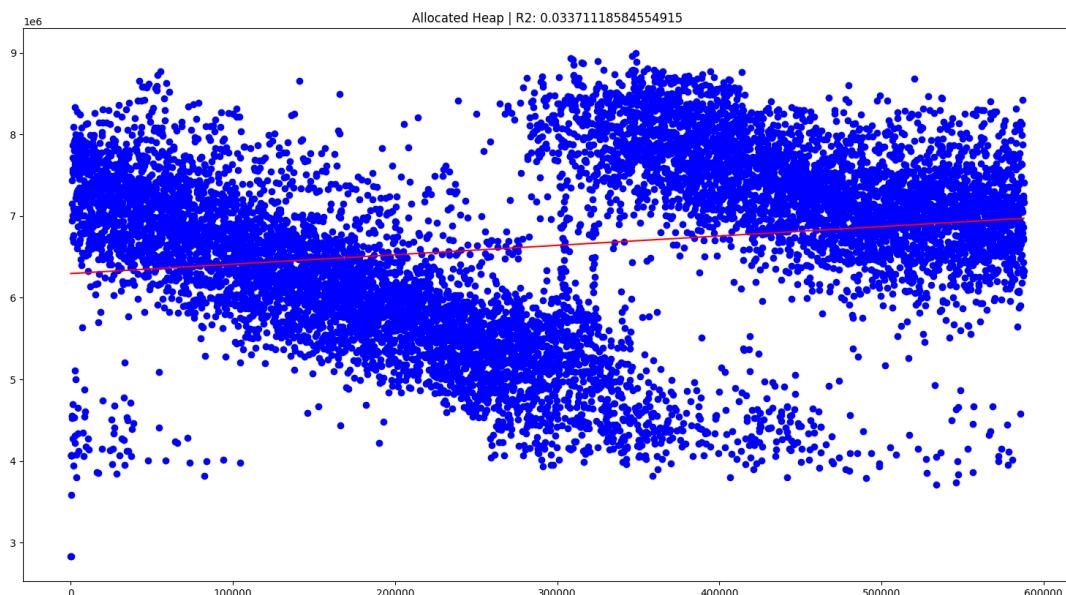


Figure 64: Regressione lineare e R^2 .

Come è possibile osservare dalla figura, l' R^2 assume un valore molto basso.

Successivamente si procede a verificare la significatività dei parametri della retta regressiva, tramite un opportuno test statistico. Per scegliere il test da utilizzare si verifica la normalità dei residui delle rispettive variabili:

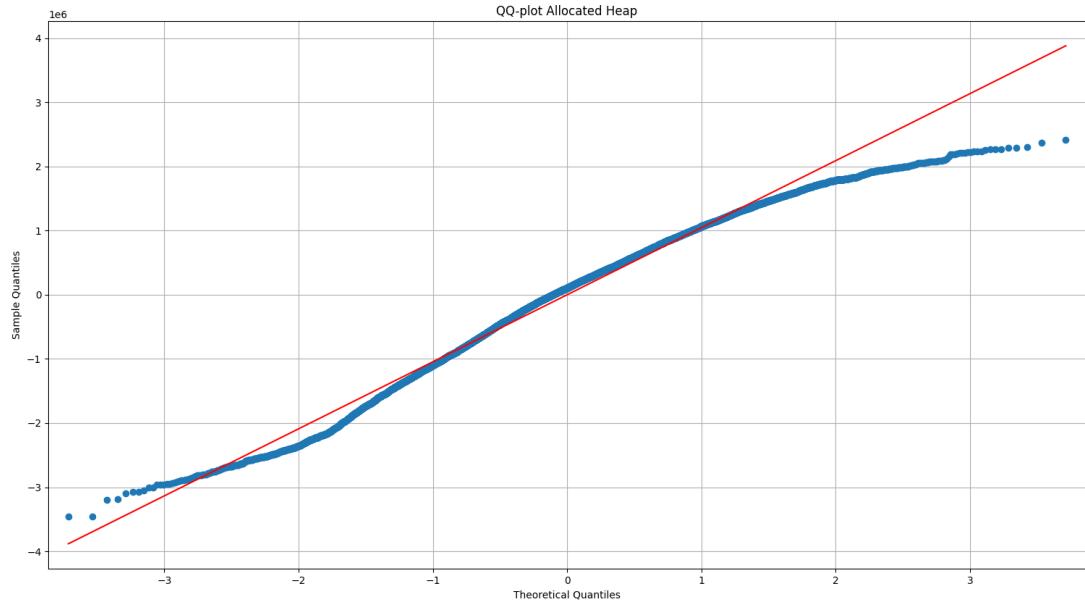


Figure 65: Plot quantile-quantile dei residui.

La distribuzione non è normale, si procede dunque con il test non parametrico di Mann-Kendall:

```
Test di Mann-Kendall (Allocated Heap), pvalue: 2.5761869168798684e-25
```

Figure 66: Test di Mann-Kendall.

Dai risultati ottenuti è possibile osservare che è presente un trend ($pvalue < 0.05$, H_0 : non è presente un trend monotono).

Si valutano dunque, il coefficiente angolare e l'intercetta della retta di regressione utilizzando la procedura di Theil-Sen. Questa procedura è una regressione lineare robusta progettata per fornire risultati più precisi e affidabili in presenza di outlier nei dati.

```
Regessione Lineare Robusta (Theil e Sen) per Allocated Heap
Slope: 0.6911224682945296
Intercept: 6562275.461669506
Low Slope: 0.563777264562484
High Slope: 0.8178240089963452
```

Figure 67: Procedura di Theil e Sen.

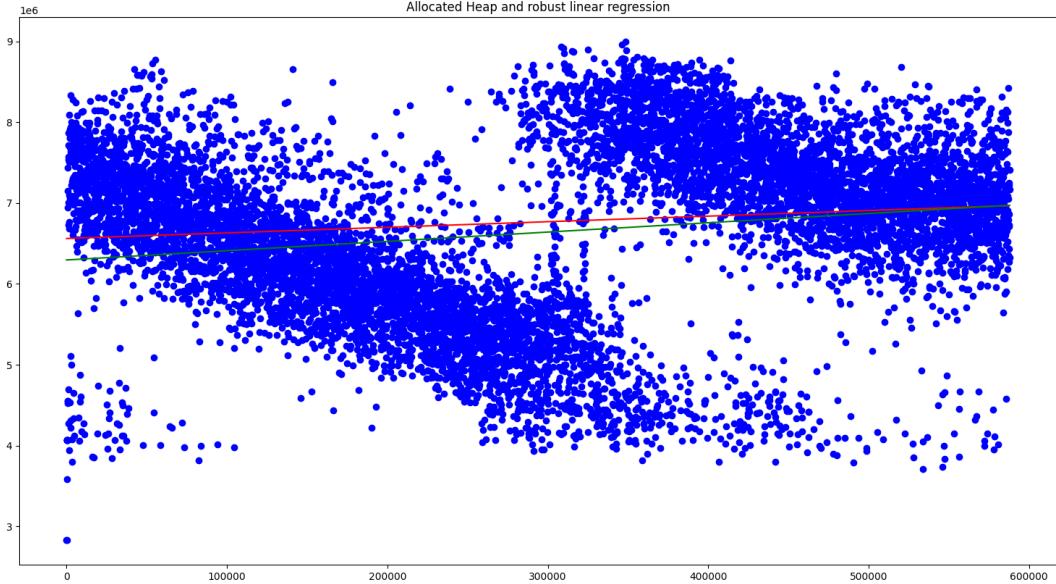


Figure 68: **Regessione robusta.**

Supponendo di avere un limite massimo alla memoria heap di 1GB, utilizzando il modello appena ottenuto, si stima il tempo di saturazione:

$$y = mx + q \quad (17)$$

dove:

- y : Allocated Heap
- x : T(s)
- m : 0.69 ± 0.13
- q : 6562275.46

$$\text{Trova } \bar{x} : \bar{y} = m\bar{x} + q = 1GB \quad (18)$$

$$\bar{x} = \frac{\bar{y} - q}{m} = 50 \pm 10 \text{ anni} \quad (19)$$

5.7 DATASET VMRES2

Il dataset presenta le seguenti colonne:

- Variabili di risposta:
 - *Allocated Heap*
- Variabili di predizione:
 - $T(s)$

Viene applicato il modello regressivo lineare la cui risposta è una funzione lineare del predittore. Nella stima viene indicato l' R^2 , ovvero il coefficiente di determinazione, che indica la percentuale di varianza spiegata dal modello rispetto a quella totale. Di seguito i risultati per il dataset corrente:

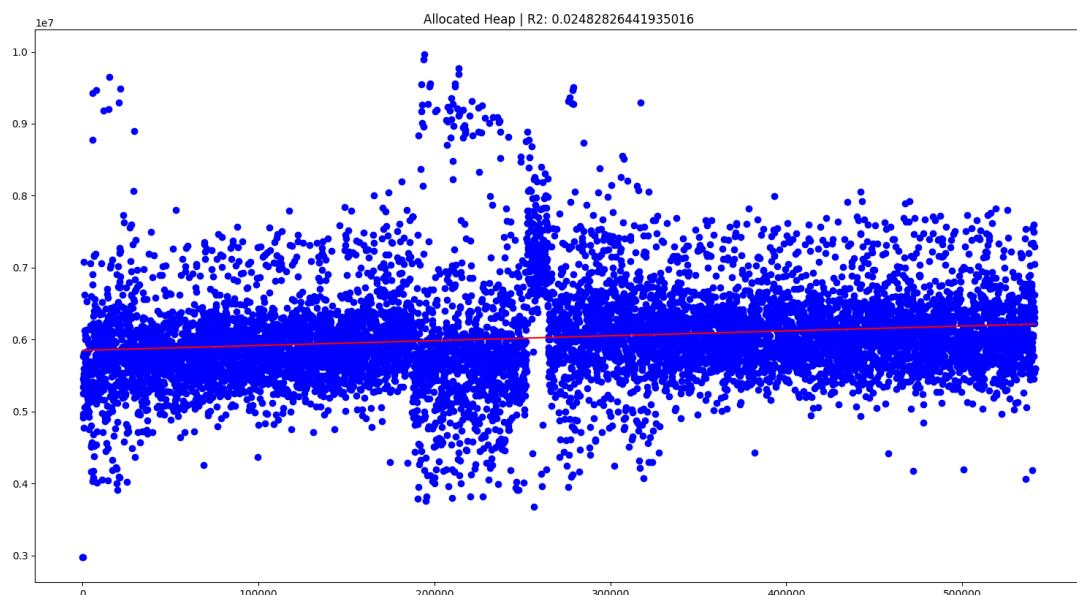


Figure 69: **Regressione lineare e R^2 .**

Come è possibile osservare dalla figura, l' R^2 assume un valore molto basso.

Successivamente si procede a verificare la significatività dei parametri della retta regressiva, tramite un opportuno test statistico. Per scegliere il test da utilizzare si verifica la normalità dei residui delle rispettive variabili:

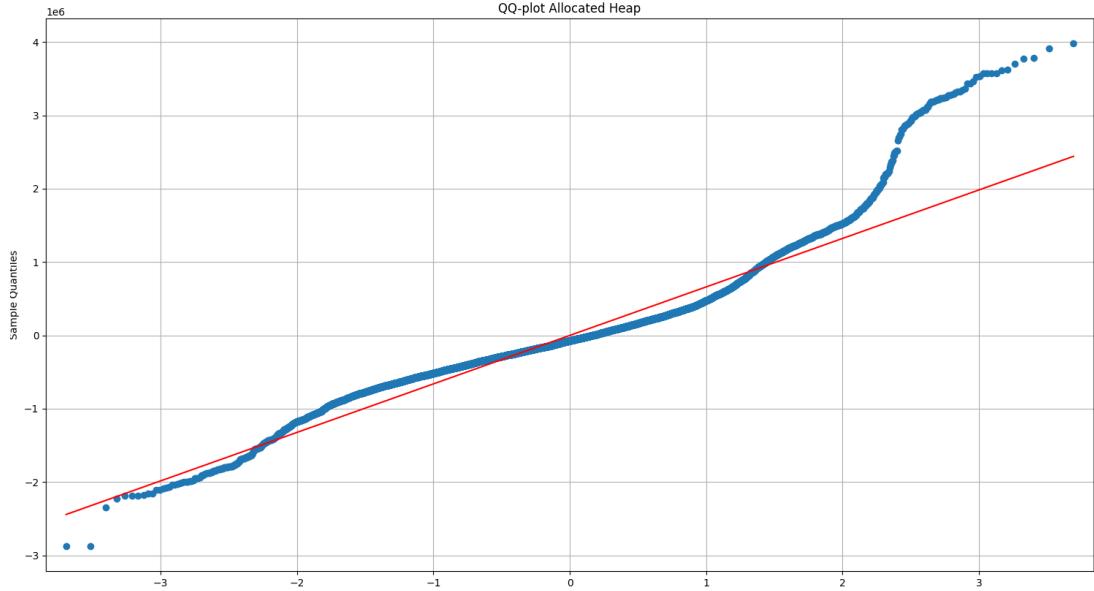


Figure 70: Plot quantile-quantile dei residui.

La distribuzione non è normale, si procede dunque con il test non parametrico di Mann-Kendall:

```
Test di Mann-Kendall (Allocated Heap), pvalue: 7.966380693966573e-86
```

Figure 71: Test di Mann-Kendall.

Dai risultati ottenuti è possibile osservare che è presente un trend ($pvalue < 0.05$, H_0 : non è presente un trend monotono).

Si valutano dunque, il coefficiente angolare e l'intercetta della retta di regressione utilizzando la procedura di Theil-Sen. Questa procedura è una regressione lineare robusta progettata per fornire risultati più precisi e affidabili in presenza di outlier nei dati.

```
Regessione Lineare Robusta (Theil e Sen) per Allocated Heap
Slope: 0.6709499304572171
Intercept: 5780180.563324536
Low Slope: 0.6048638415610893
High Slope: 0.7366538131962297
```

Figure 72: Procedura di Theil e Sen.

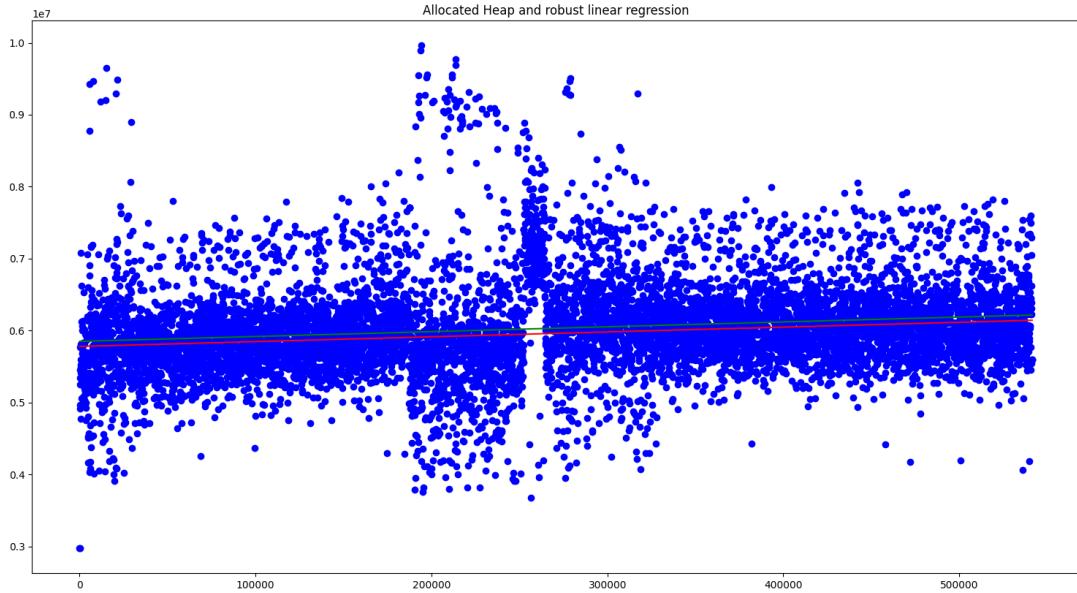


Figure 73: **Regressione robusta.**

Supponendo di avere un limite massimo alla memoria heap di 1GB, utilizzando il modello appena ottenuto, si stima il tempo di saturazione:

$$y = mx + q \quad (20)$$

dove:

- y : Allocated Heap
- x : T(s)
- m : 0.67 ± 0.06
- q : 5780180.56

$$\text{Trova } \bar{x} : \bar{y} = m\bar{x} + q = 1GB \quad (21)$$

$$\bar{x} = \frac{\bar{y} - q}{m} = 50 \pm 5 \text{ anni} \quad (22)$$

5.8 DATASET VMRES3

Il dataset presenta le seguenti colonne:

- Variabili di risposta:
 - *Allocated Heap*
- Variabili di predizione:
 - $T(s)$

Viene applicato il modello regressivo lineare la cui risposta è una funzione lineare del predittore. Nella stima viene indicato l' R^2 , ovvero il coefficiente di determinazione, che indica la percentuale di varianza spiegata dal modello rispetto a quella totale. Di seguito i risultati per il dataset corrente:

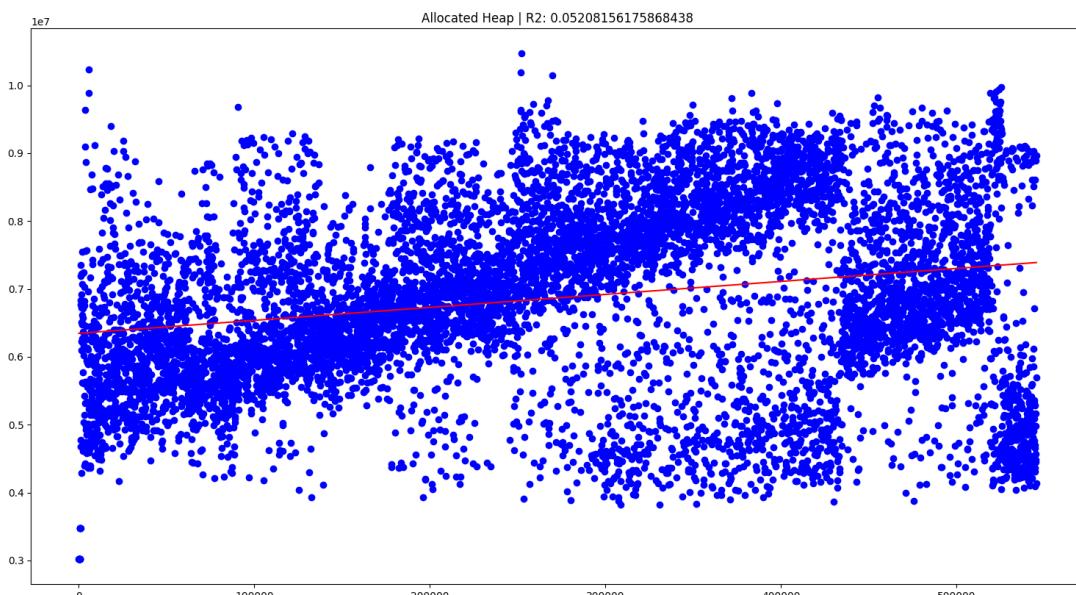


Figure 74: Regressione lineare e R^2 .

Come è possibile osservare dalla figura, l' R^2 assume un valore molto basso.

Successivamente si procede a verificare la significatività dei parametri della retta regressiva, tramite un opportuno test statistico. Per scegliere il test da utilizzare si verifica la normalità dei residui delle rispettive variabili:

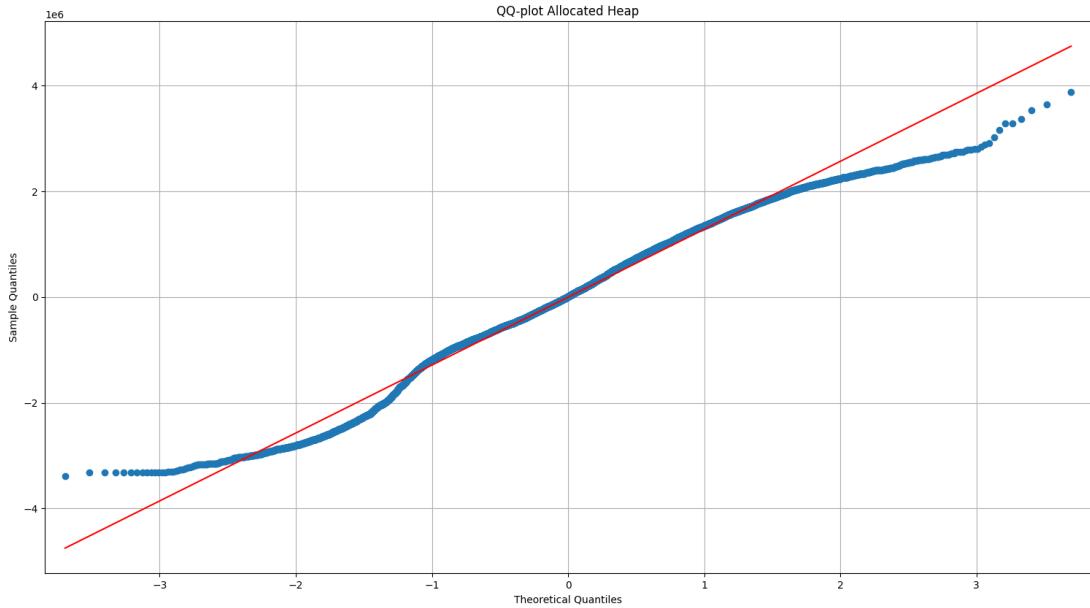


Figure 75: Plot quantile-quantile dei residui.

La distribuzione non è normale, si procede dunque con il test non parametrico di Mann-Kendall:

```
Test di Mann-Kendall (Allocated Heap), pvalue: 2.0833905055590785e-185
```

Figure 76: Test di Mann-Kendall.

Dai risultati ottenuti è possibile osservare che è presente un trend ($pvalue < 0.05$, H_0 : non è presente un trend monotono).

Si valutano dunque, il coefficiente angolare e l'intercetta della retta di regressione utilizzando la procedura di Theil-Sen. Questa procedura è una regressione lineare robusta progettata per fornire risultati più precisi e affidabili in presenza di outlier nei dati.

```
Regressione Lineare Robusta (Theil e Sen) per Allocated Heap
Slope: 2.9030932760062105
Intercept: 6041585.556431387
Low Slope: 2.714759535655058
High Slope: 3.0923529411764705
```

Figure 77: Procedura di Theil e Sen.

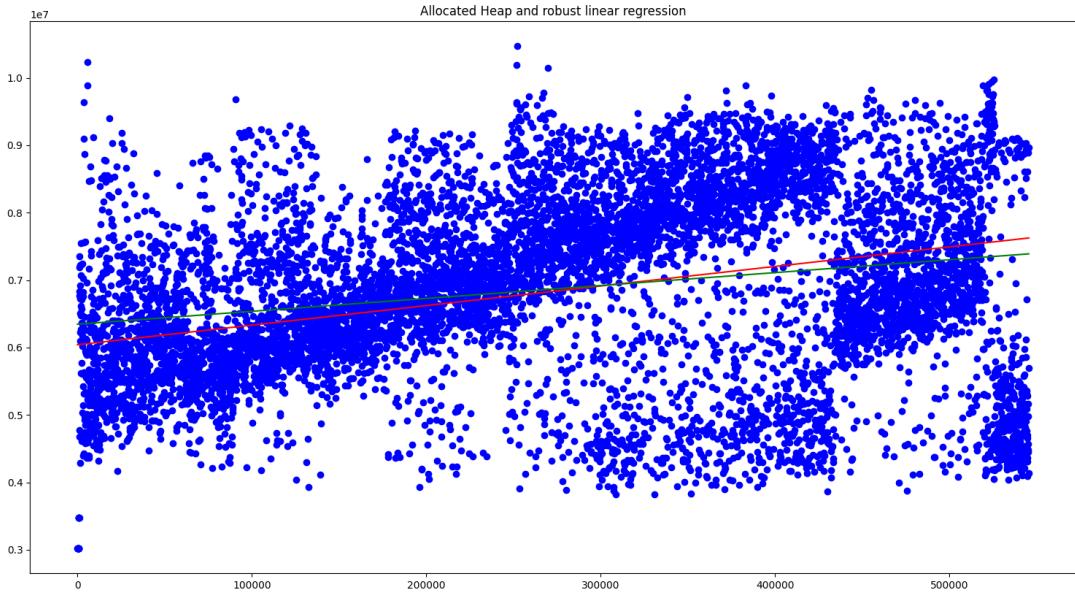


Figure 78: **Regessione robusta.**

Supponendo di avere un limite massimo alla memoria heap di 1GB, utilizzando il modello appena ottenuto, si stima il tempo di saturazione:

$$y = mx + q \quad (23)$$

dove:

- y : Allocated Heap
- x : T(s)
- m : 2.90 ± 0.19
- q : 6041585.55

$$\text{Trova } \bar{x} : \bar{y} = m\bar{x} + q = 1GB \quad (24)$$

$$\bar{x} = \frac{\bar{y} - q}{m} = 12 \pm 1 \text{ anni} \quad (25)$$

6 RELIABILITY

In questo paragrafo verranno svolti degli esercizi riguardanti l'analisi di Reliability di alcuni sistemi.

6.1 ESERCIZIO 1: RELIABILITY BLOCK DIAGRAM

Dato il seguente sistema, bisognerà calcolare la Reliability ed il MTTF, supponendo che tutti i componenti siano identici e con $R = e^{-\lambda t}$.

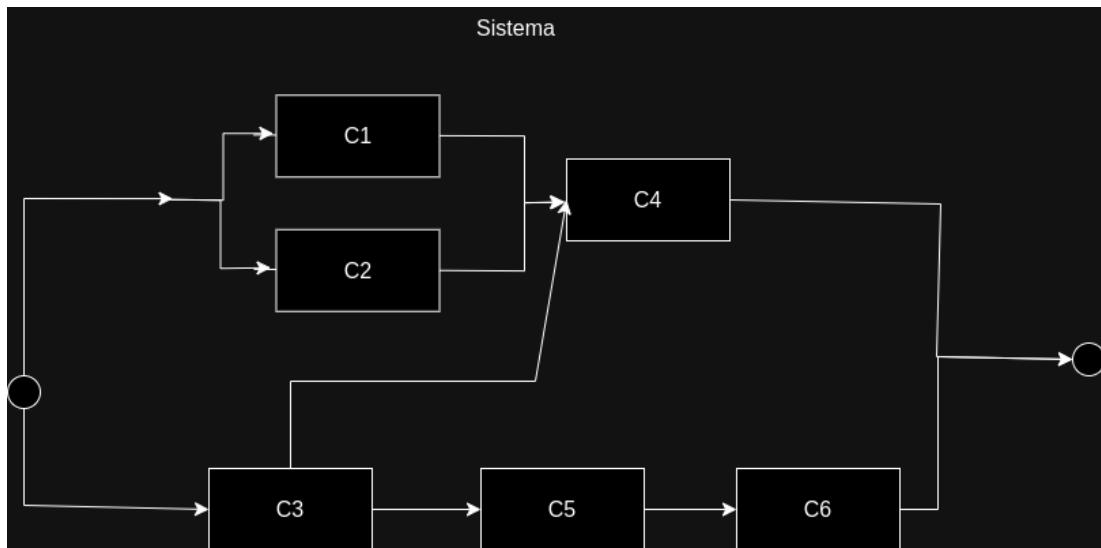


Figure 79: Reliability Block Diagram.

Dalla figura è possibile osservare che C_1 e C_2 sono in parallelo, mentre C_5 e C_6 sono in serie. Tuttavia i componenti rimanenti non sono in serie-parallelo.

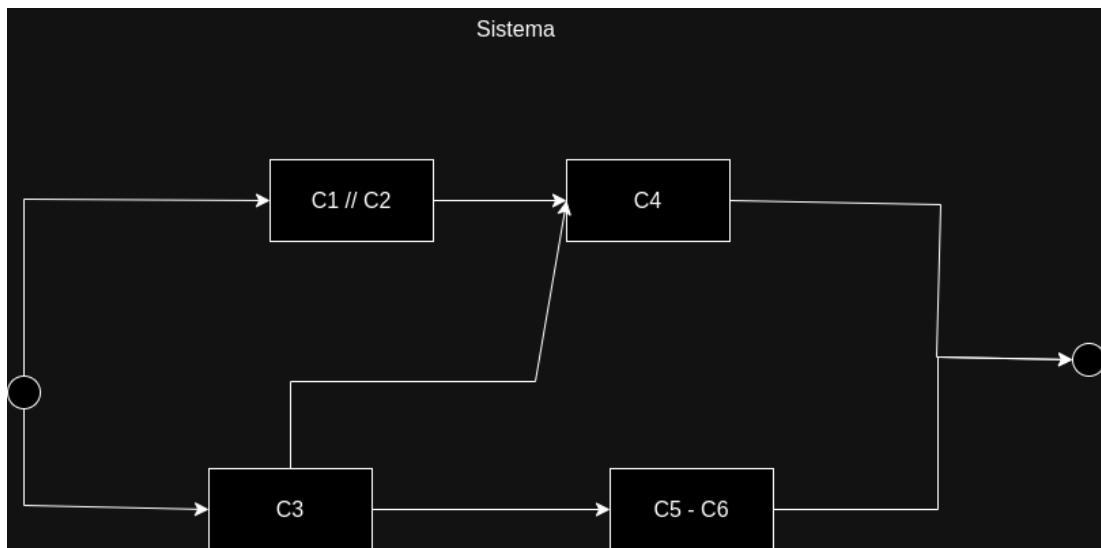


Figure 80: Reliability Block Diagram semplificato.

Viene analizzato il sistema rimanente con la **tecnica del Conditioning** e con il **Teorema dell'Upper Bound**.

Per utilizzare il **Teorema dell'Upper Bound** bisogna enumerare i Success Path e posizionarli in parallelo; un Upper Bound per il sistema di partenza, è la Reliability del sistema ottenuto posizionando i Success Path in parallelo.

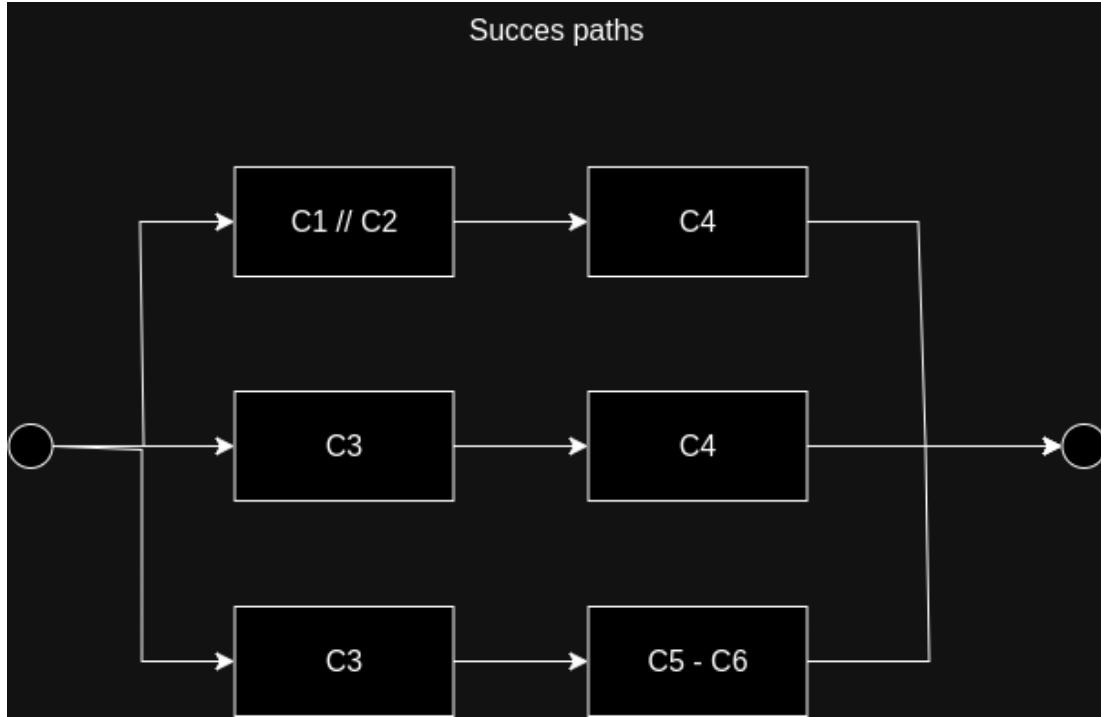


Figure 81: Teorema dell'Upper Bound.

$$R_{sistema} \leq 1 - (1 - R(1 - (1 - R)^2))(1 - R^2)(1 - R^3) \quad (26)$$

$$R_{sistema} \leq 3R^2 - 2R^4 - 2R^5 + R^6 + 2R^7 - R^8 \quad (27)$$

Utilizzando la **tecnica del Conditioning** invece, si procede a condizionare il sistema rispetto al sottosistema C_3 .

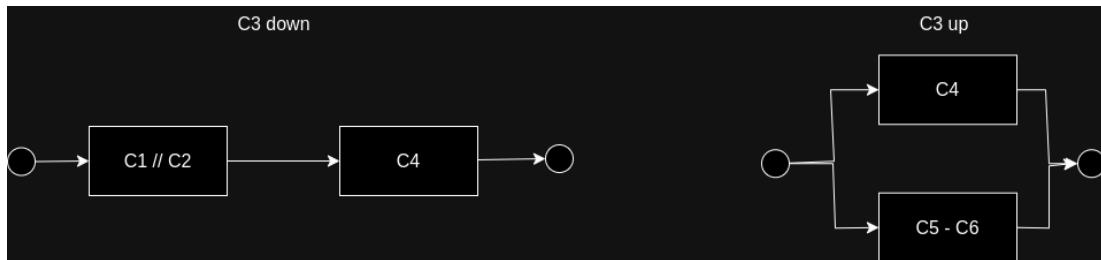


Figure 82: Tecnica del Conditioning.

$$R_{sistema} = R \cdot P(\text{system works} | C_3 \text{ works}) + (1 - R) \cdot P(\text{system works} | C_3 \text{ does not work}) \quad (28)$$

$$R_{sistema} = R(1 - (1 - R)(1 - R^2)) + (1 - R)(R(1 - (1 - R)^2)) \quad (29)$$

$$R_{sistema} = -2R^3 + 3R^2 \quad (30)$$

$$MTTF = \int_0^{+\infty} R_{sistema}(t)dt = \int_0^{+\infty} -2e^{-3\lambda t} + 3e^{-2\lambda t} dt = \frac{5}{6\lambda} \quad (31)$$

6.2 ESERCIZIO 2: CONFRONTO TRA SISTEMI

Si confrontano le reliability di due sistemi composti dagli stessi sottosistemi, ma con una topologia differente.

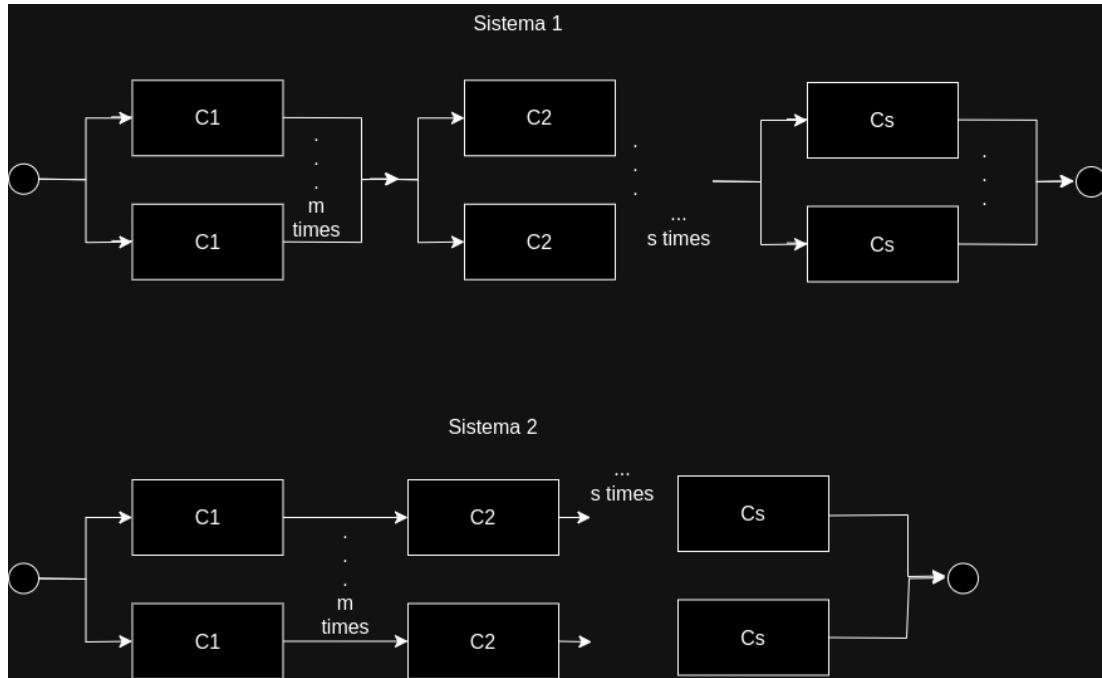


Figure 83: **Sistemi da confrontare.**

$$R_{sistema1} = (1 - (1 - R)^m)^s \quad (32)$$

$$R_{sistema2} = 1 - (1 - R^s)^m \quad (33)$$

Considerando $m = 2$, $s = 4$, $MTTF = 800h$ e $R = e^{-\lambda t}$, si confrontano i due sistemi:

$$R_{sistema1} = (1 - (1 - R)^2)^4 \quad (34)$$

$$R_{sistema2} = 1 - (1 - R^4)^2 \quad (35)$$

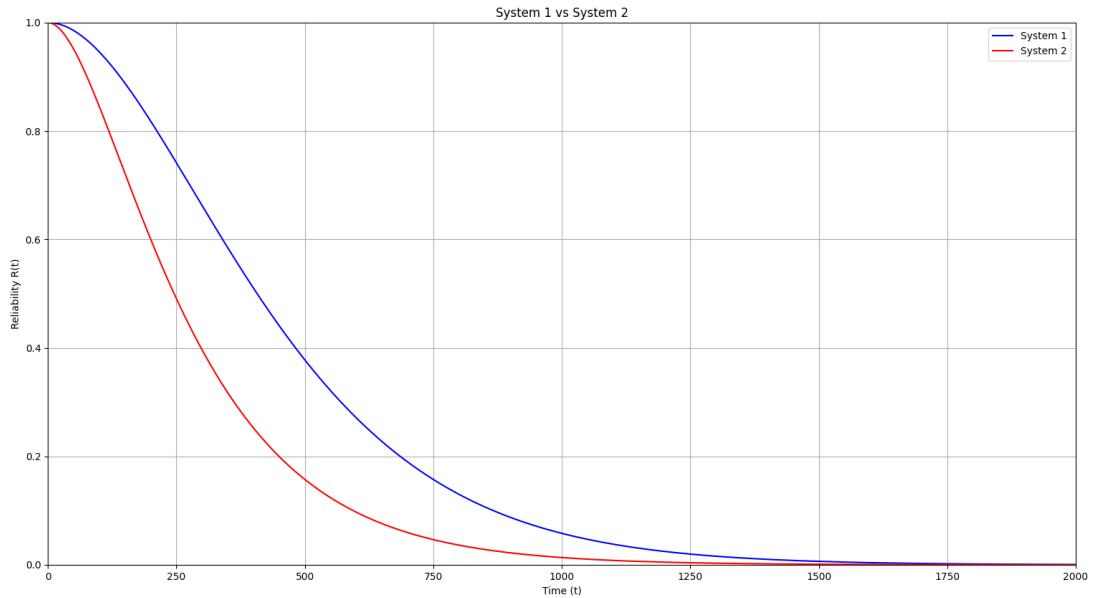


Figure 84: Confronto tra i due sistemi.

È possibile osservare dunque che $R_{sistema1} > R_{sistema2}$.

Successivamente si modifica il secondo sistema affinchè per un mission time di 800h, questo abbia la stessa Reliability del primo sistema, aumentando i collegamenti in parallelo.

$$R_{sistema1}(800) = 13\% \quad (36)$$

$$R_{sistema2}(800) = 3.6\% \quad (37)$$

$$R_{sistema2*}(800) = 1 - (1 - R^4)^m = 1 - (1 - e^{-4})^m = 0.13 \quad (38)$$

$$m = \frac{\ln 0.87}{\ln (1 - e^{-4})} = 7.53 \Rightarrow m = 8 \quad (39)$$

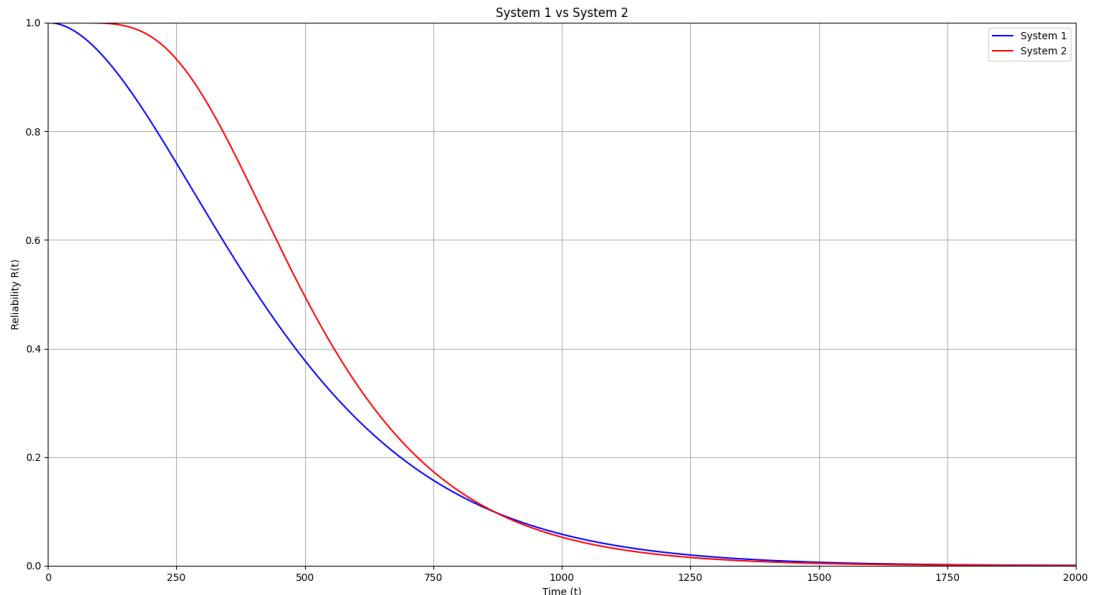


Figure 85: Confronto tra i due sistemi.

6.3 ESERCIZIO 3: SKIP RING NETWORK

Si valuta la Reliability della seguente configurazione, assumendo come Reliability di ogni singolo nodo $R_m = e^{-\lambda t}$ e $\lambda = 0.005$: Questa configurazione è costruita in modo tale da garantire che il

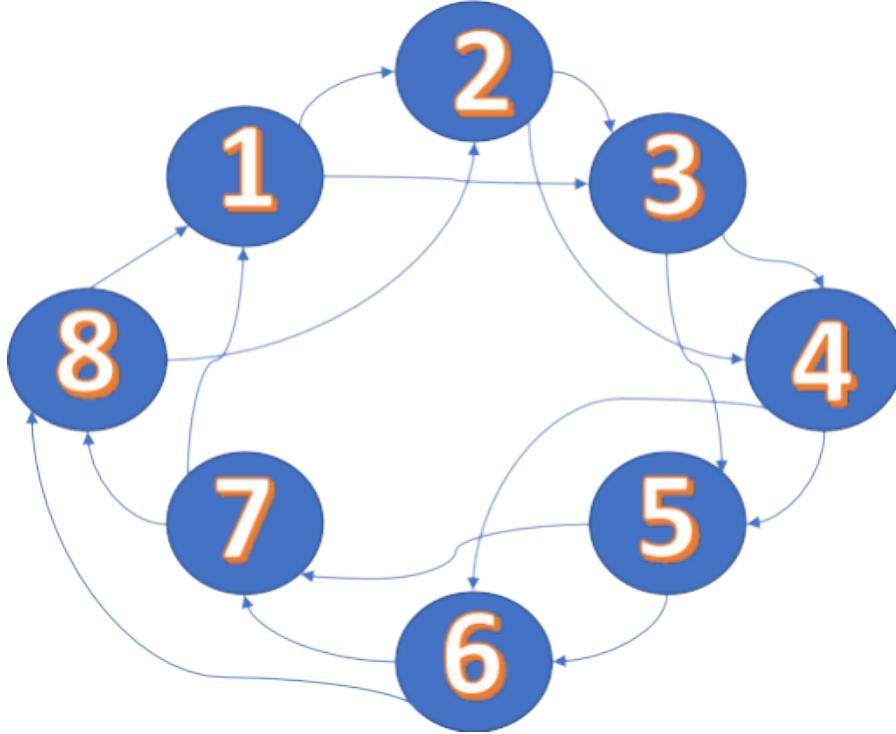


Figure 86: Skip Ring.

fallimento di un nodo non limiti la possibilità degli altri nodi di comunicare, infatti, ogni nodo è connesso con i due nodi seguenti. Si prova a valutare le combinazioni per cui il sistema non funziona calcolando la Reliability del sistema con la seguente formula (M-out-of-N-Systems) dove M è il numero di sottosistemi fuori uso e N è il numero totale di sottosistemi in una rete.

$$R_{M,N} = \sum_{i=0}^{N-M} \binom{N}{i} R_m^{N-i} (1-R_m)^i \quad (40)$$

Si valutano i seguenti casi:

- $i = 0$: Tutti i nodi funzionano

$$R_0 = \binom{8}{0} R_m^8 = R_m^8 \quad (41)$$

- $i = 1$: Un nodo fallisce, ma il sistema continua a funzionare

$$R_1 = \binom{8}{1} R_m^7 (1-R_m) = 8R_m^7 (1-R_m) \quad (42)$$

- $i = 2$: Due nodi falliscono ($\binom{8}{2}$ possibili configurazioni), ma il sistema continua a funzionare se i nodi non sono adiacenti (8 configurazioni in cui sono adiacenti)

$$R_2 = (\binom{8}{2} - 8) R_m^6 (1-R_m)^2 = 20R_m^6 (1-R_m)^2 \quad (43)$$

- **i = 3:** Tre nodi falliscono ($\binom{8}{3}$ possibili configurazioni), il sistema fallisce se i nodi sono tutti adiacenti (8 configurazioni) o se 2 sono adiacenti e l'altro no (32 configurazioni)

$$R_3 = \left(\binom{8}{3} - 8 - 32 \right) R_m^5 (1 - R_m)^3 = 16 R_m^5 (1 - R_m)^3 \quad (44)$$

- **i = 4:** Quattro nodi falliscono, il sistema continua a funzionare solo se i nodi non funzionanti si alternano (2 configurazioni)

$$R_4 = 2 R_m^4 (1 - R_m)^4 \quad (45)$$

- **i = 5, 6, 7, 8:** In questi casi, il sistema fallisce

$$R_{sistema} = R_m^8 + 8R_m^7(1 - R_m) + 20R_m^6(1 - R_m)^2 + 16R_m^5(1 - R_m)^3 + 2R_m^4(1 - R_m)^4 \quad (46)$$

$$R_{sistema}(48) = 0.72888 \quad (47)$$

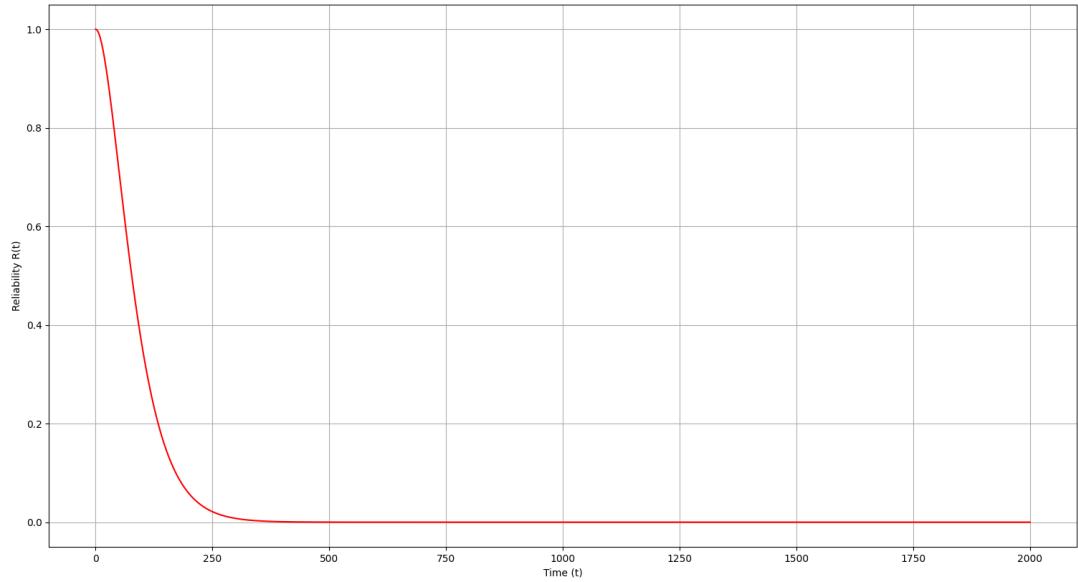


Figure 87: Andamento della Reliability.

6.4 ESERCIZIO 4: CONFRONTO TRA SISTEMI

Siano 4 coppie di sistemi composti da 3 sottosistemi (A,B,C) caratterizzati da $R = e^{-\lambda t}$ e dai seguenti MTTF:

- $MTTF_A = 500$ h
- $MTTF_B = 9000$ h
- $MTTF_C = 10000$ h

6.4.1 CONFRONTO 1

La prima coppia di sistemi da confrontare è:

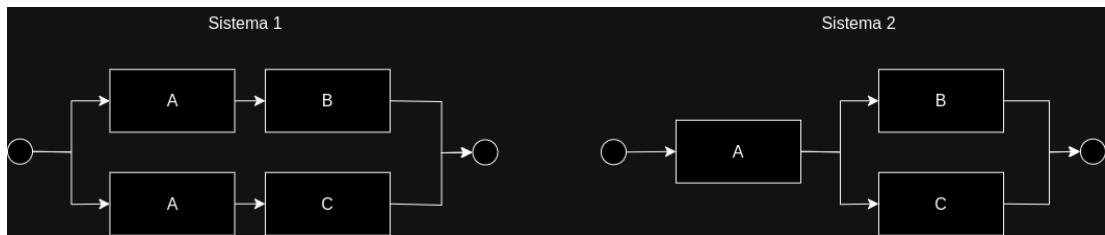


Figure 88: **Confronto 1.**

Entrambi i sistemi presentano sottosistemi collegati tra loro in serie-parallelo, dunque è possibile calcolare in maniera semplice la Reliability di ogni sistema:

$$R_{sistema1} = 1 - (1 - R_A R_B)(1 - R_A R_C) \quad (48)$$

$$R_{sistema2} = R_A(1 - R_B R_C) \quad (49)$$

Successivamente vengono plottati gli andamenti delle Reliability:

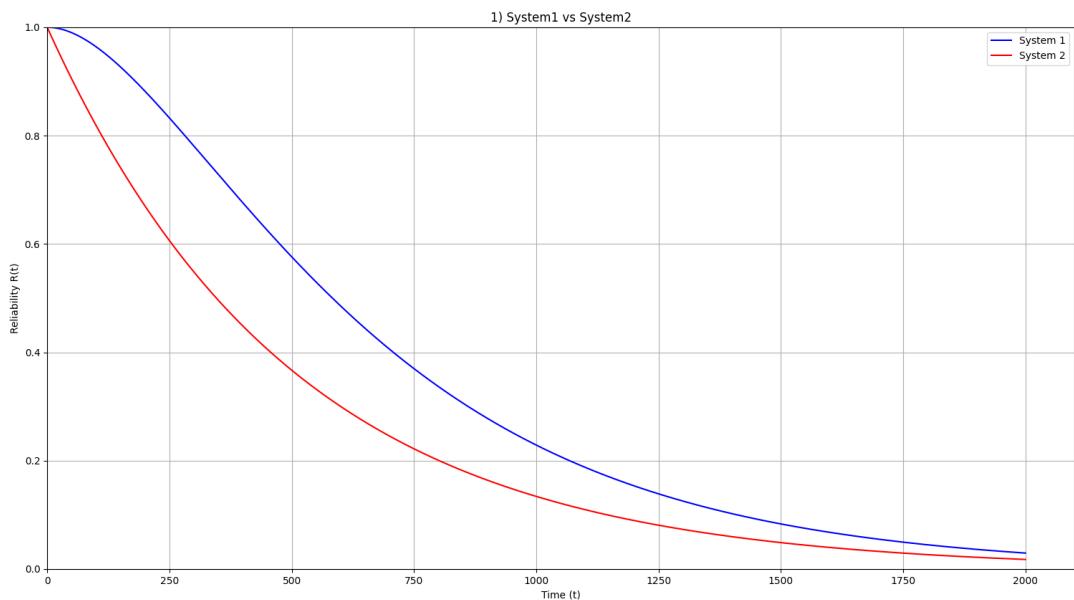


Figure 89: **Confronto 1.**

È possibile osservare dunque che $R_{sistema1} > R_{sistema2}$.

6.4.2 CONFRONTO 2

La seconda coppia di sistemi da confrontare è:

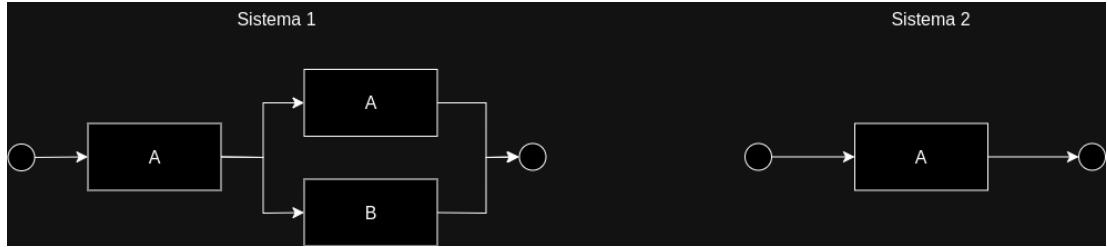


Figure 90: **Confronto 2.**

Entrambi i sistemi presentano sottosistemi collegati tra loro in serie-parallelo, dunque è possibile calcolare in maniera semplice la Reliability di ogni sistema:

$$R_{sistema1} = R_A(1 - (1 - R_A)(1 - R_B)) \quad (50)$$

$$R_{sistema2} = R_A \quad (51)$$

Successivamente vengono plottati gli andamenti delle Reliability:

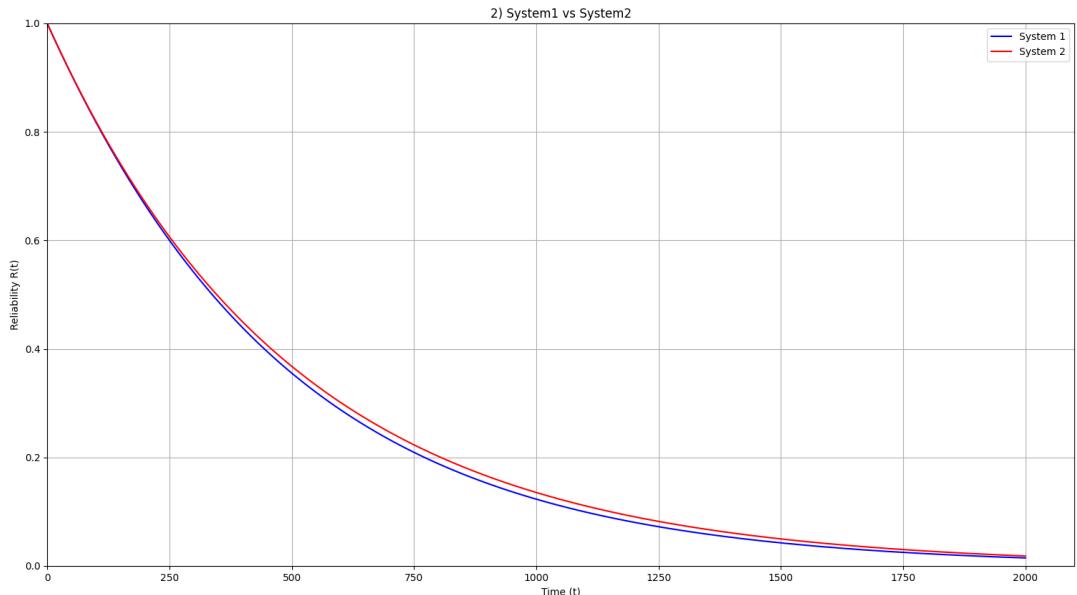


Figure 91: **Confronto 2.**

È possibile osservare dunque che $R_{sistema2} > R_{sistema1}$.

6.4.3 CONFRONTO 3

La terza coppia di sistemi da confrontare è:

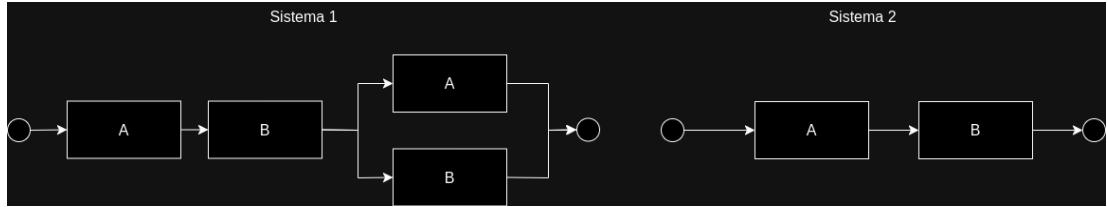


Figure 92: **Confronto 3.**

Entrambi i sistemi presentano sottosistemi collegati tra loro in serie-parallelo, dunque è possibile calcolare in maniera semplice la Reliability di ogni sistema:

$$R_{sistema1} = R_A R_B (1 - (1 - R_A)(1 - R_B)) \quad (52)$$

$$R_{sistema2} = R_A R_B \quad (53)$$

Successivamente vengono plottati gli andamenti delle Reliability:

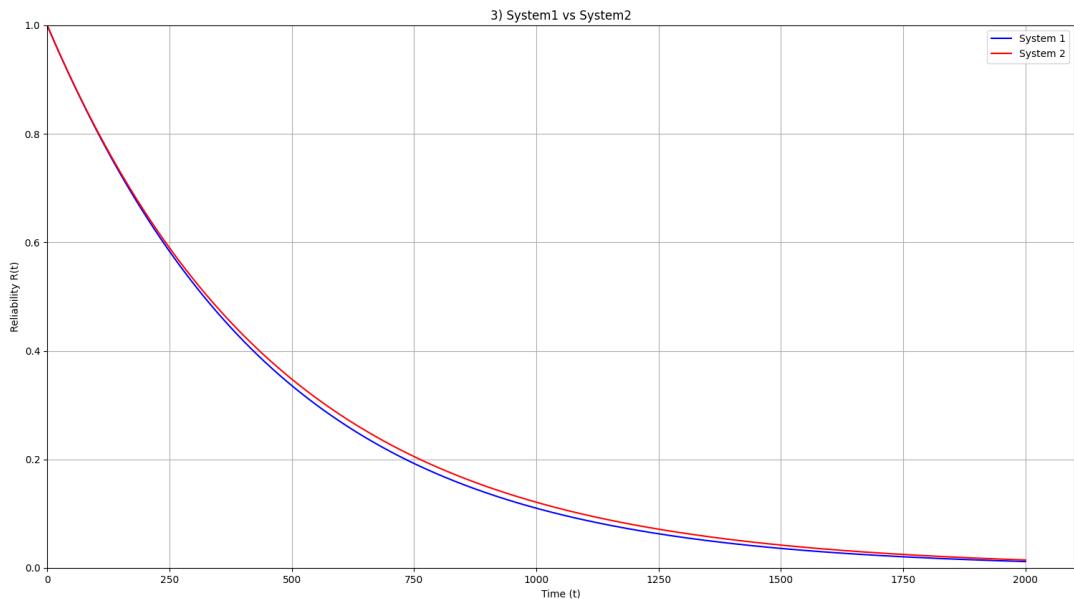


Figure 93: **Confronto 3.**

È possibile osservare dunque che $R_{sistema2} > R_{sistema1}$.

6.4.4 CONFRONTO 4

La quarta coppia di sistemi da confrontare è:

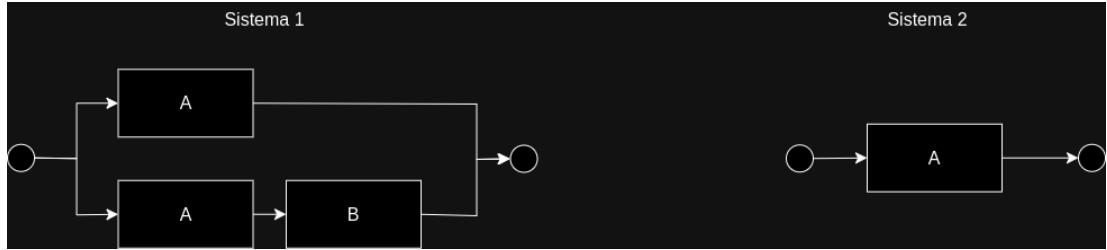


Figure 94: **Confronto 4.**

Entrambi i sistemi presentano sottosistemi collegati tra loro in serie-parallelo, dunque è possibile calcolare in maniera semplice la Reliability di ogni sistema:

$$R_{sistema1} = 1 - (1 - R_A)(1 - R_A R_B) \quad (54)$$

$$R_{sistema2} = R_A \quad (55)$$

Successivamente vengono plottati gli andamenti delle Reliability:

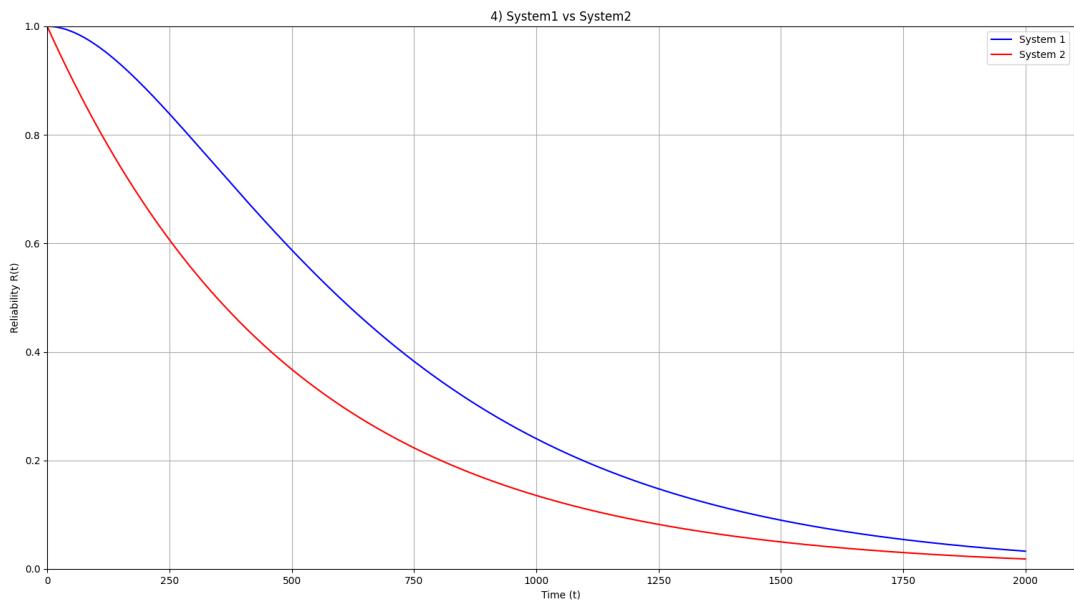


Figure 95: **Confronto 4.**

È possibile osservare dunque che $R_{sistema1} > R_{sistema2}$.

6.5 ESERCIZIO 5: HELICOPTER PROCESSING SYSTEM

Si modella e analizza, tramite RBD e FT, un sistema di controllo di un elicottero.

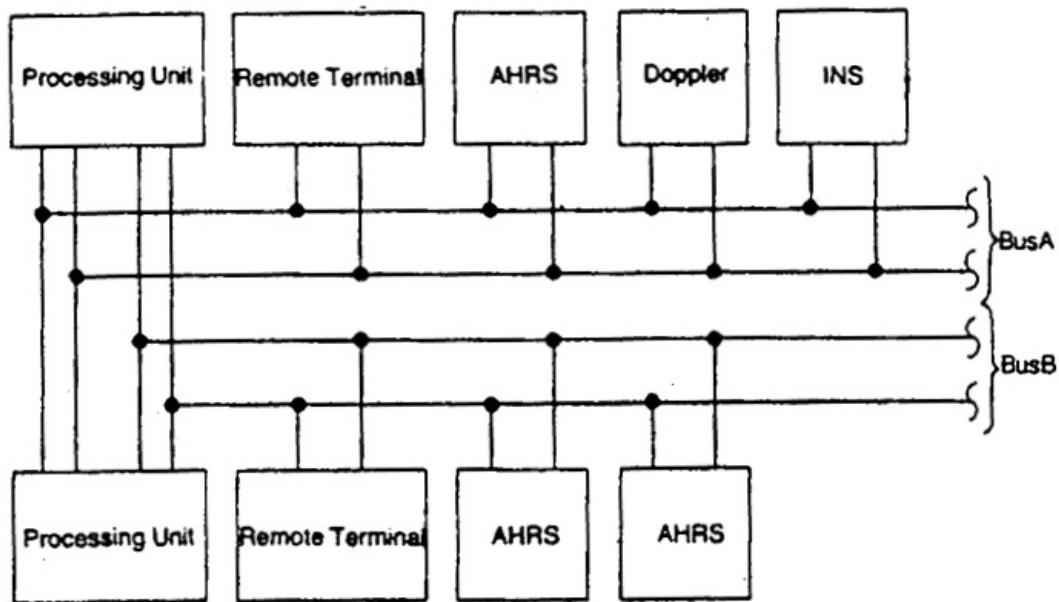


Figure 96: **Helicopter Processing System.**

Dalla descrizione del sistema sappiamo che:

- Sono presenti due processori ridondanti
- Sono presenti due terminali remoti ridondanti
- Sono presenti due bus (A e B) entrambi ridondanti
- L'elicottero può navigare utilizzando il sistema INS oppure, in caso di emergenza, il sistema Doppler ed il sistema AHRS (di cui sono presenti 3 unità)

Possiamo dunque modellare il sistema mediante un RBD:

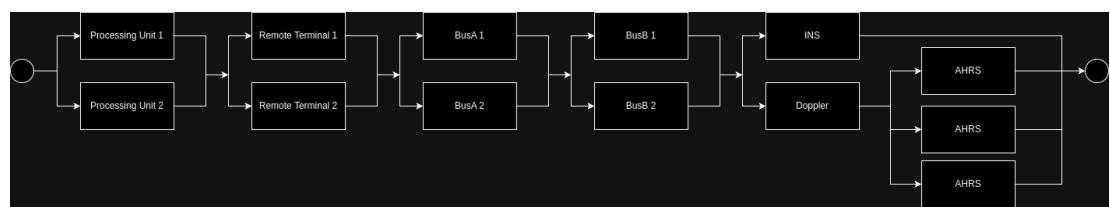


Figure 97: **Reliability Block Diagram.**

$$R_{sistema} = (1 - (1 - R_{ProcessingUnit})^2)(1 - (1 - R_{RemoteTerminal})^2)(1 - (1 - R_{BusA})^2)(1 - (1 - R_{BusB})^2) \\ (1 - (1 - R_{INS})(1 - R_{Doppler}(1 - (1 - R_{AHRS})^3))) \quad (56)$$

Equipment	MTTF (hr)
Processing Unit	10000
Remote Terminal	4500
AHRS	2000
INS	2000
Doppler	500
Bus	60000

Table 21: **MTTF dei sottosistemi.**

$$R_{sistema}(1) = 0.9999989 \quad (57)$$

Di seguito la modellazione del sistema mediante FT, in cui sono evidenziati i minimal cutsets:

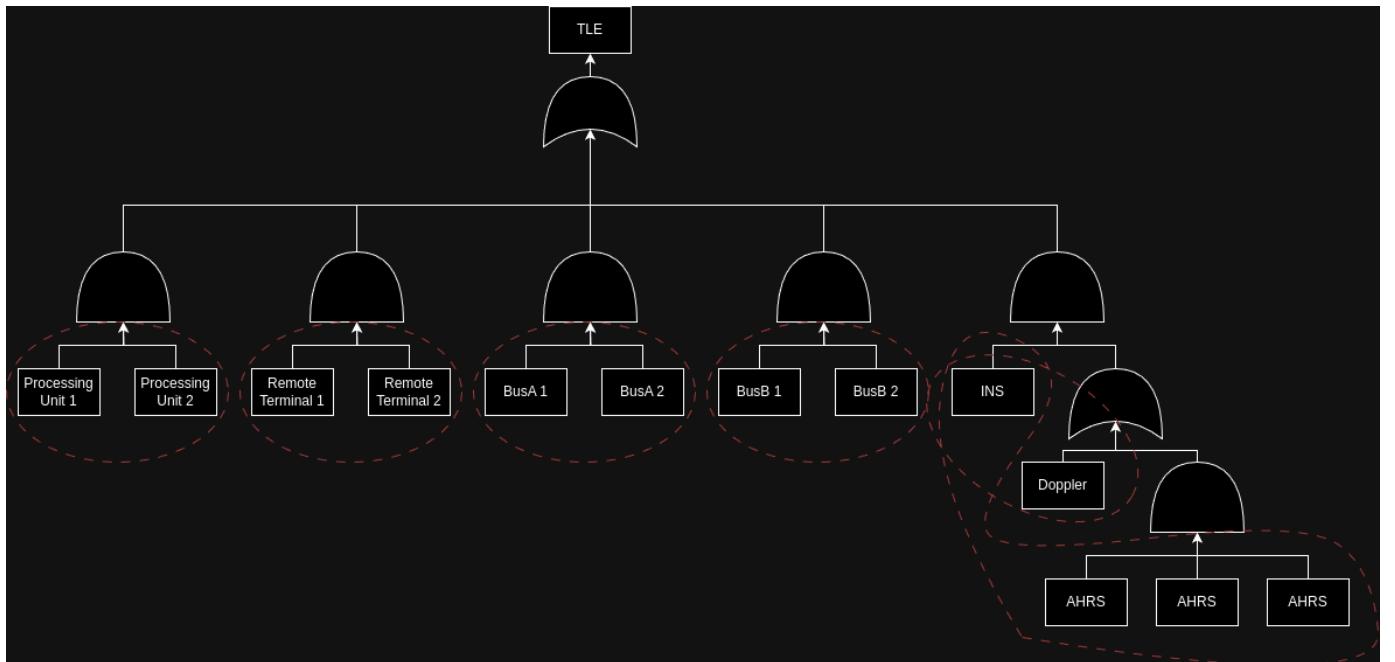


Figure 98: **Fault Tree.**

Infine, si considera un fattore di Coverage C attribuito alle CPU del sistema. Si procede al calcolo di C affinchè la Reliability (considerando un mission time di 1h) sia 0.99999:

$$R^* = (1 - (1 - R_{RemoteTerminal})^2)(1 - (1 - R_{BusA})^2)(1 - (1 - R_{BusB})^2) \\ (1 - (1 - R_{INS})(1 - R_{Doppler}(1 - (1 - R_{AHRS})^3))) \quad (58)$$

$$R_{sistema} = (R_{ProcessingUnit} + C(1 - R_{ProcessingUnit})R_{ProcessingUnit}) \cdot R^* = 0.99999 \quad (59)$$

$$C = \frac{\frac{0.99999}{R^*} - R_{ProcessingUnit}}{(1 - R_{ProcessingUnit})R_{ProcessingUnit}} = 0.9105732657648757 \quad (60)$$

7 FIELD FAILURE DATA ANALYSIS

La **Field Failure Data Analysis (FFDA)** è una tecnica di Data Analysis utilizzata per misurare attributi di Dependability del sistema, quando è sottoposta a Workload reali.

La FFDA è suddivisa nelle seguenti fasi:

- **Logging Collection:** raccolta e memorizzazione dei dati relativi al sistema mentre è in funzione.
- **Filtering:** si effettua un filtraggio dei dati raccolti per concentrare l'analisi solo sui dati significativi, tale processo è realizzato sulla base del log di interesse mediante approcci White Listing o Black Listing.
- **Data Manipulation:** i dati sono aggregati per mezzo di un processo di coalescenza. Tale operazione permette di raggruppare failure relativi allo stesso guasto all'interno di uno stesso elemento detto tupla per ridurre in maniera ulteriore e notevole la dimensione del dataset iniziale.
- **Data Analysis:** i dati sono sottoposti ad analisi, per differenti scopi, che possono essere quello di trovare la curva di reliability del sistema, fino all'identificazione della distribuzione del Time to Failure.

7.1 CALCOLATORE MERCURY

In questa sezione verrà effettuata l'analisi dei log del calcolatore Mercury.

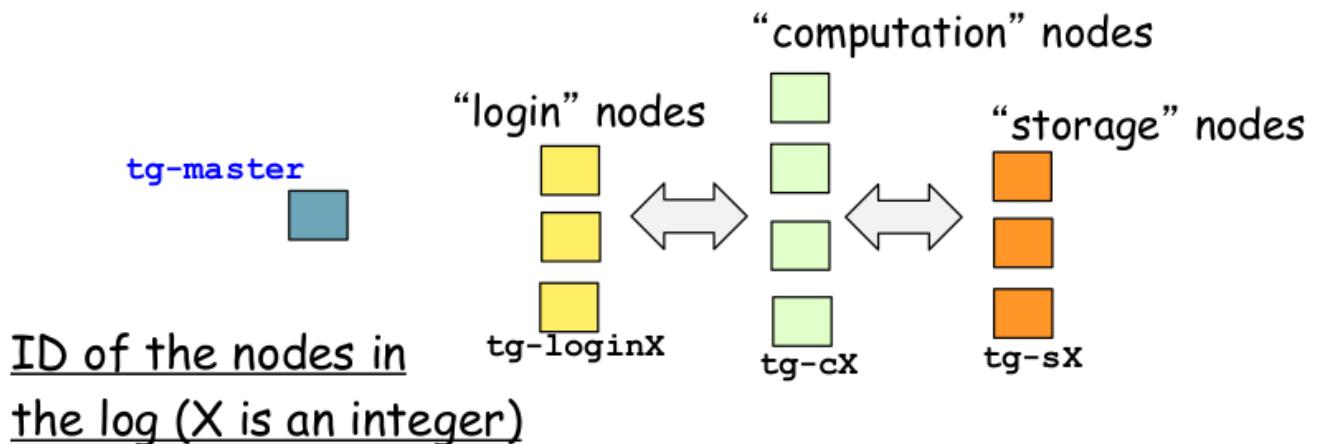


Figure 99: Mercury.

7.1.1 LOGGING COLLECTION E FILTERING

I dati sono raccolti in un file di log con 80854 entry (solo errori fatali del sistema). Il formato dei log è il seguente:

- **Timestamp**
- **Nodo di origine**
- **Categoria di errore**
- **Messaggio di failure**

7.1.2 DATA MANIPULATION

Nella fase di manipulation si identificano, a partire dal log del sistema, i fallimenti del sistema. Per identificare i fallimenti di un sistema si utilizza la tecnica della Coalescenza in grado di rilevare la correlazione tra le entry del file di log. Essa ci permette di ridurre l'insieme delle informazioni raccolte durante la fase di logging collection. Si vanno a raggruppare tutti i fallimenti relativi al medesimo guasto in un unico evento riducendone la lista dei log. In particolare si utilizza la tecnica della coalescenza temporale valutata sulla base della seguente condizione:

$$\text{IF } t(X_{i+1}) - t(X_i) < W \text{ THEN ADD } X_{i+1} \text{ to the tuple} \quad (61)$$

Le entry dei log sono dunque raggruppate in una tupla se la loro distanza è minore di W (finestra di coalescenza). Per determinare W si effettua dunque un'analisi di sensitività.

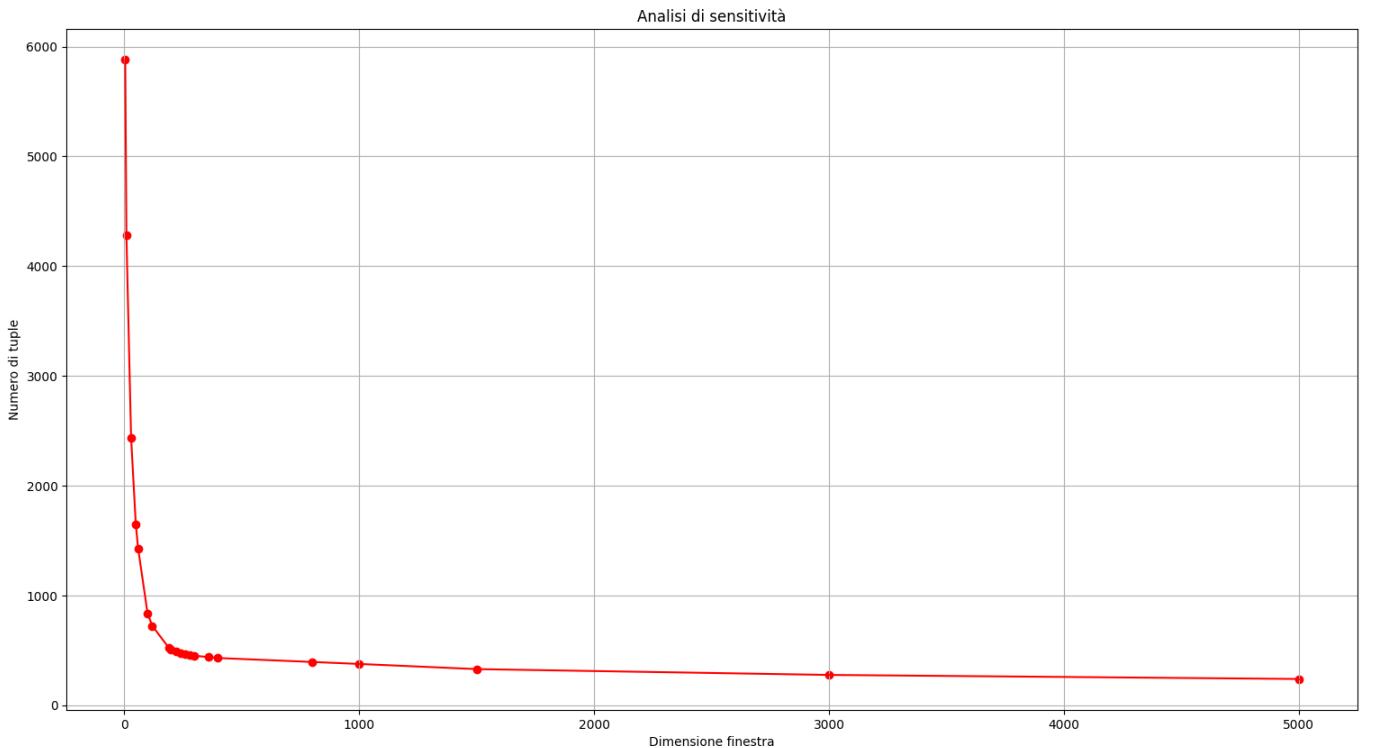


Figure 100: Analisi di sensitività (Mercury).

Si sceglie come valore di W , quello immediatamente successivo al valore corrispondente al ginocchio della curva. In questo caso è stata scelta una **finestra di coalescenza della durata di 200s**, che ci permetterà di ottenere **508 tuple**. Utilizzando uno script Python è stato analizzato il numero di troncamenti e collisioni, in particolare per questa W sono presenti 233 troncamenti (45.8%) e 47 collisioni (9.2%).

7.1.3 DATA ANALYSIS

Nell'ultima fase dell'FFDA viene effettuato uno studio sulla Reliability del sistema. Grazie al calcolo degli interarrivi (tempo che intercorre tra la prima entry di una tupla e l'ultima entry della tupla precedente), è possibile ottenere una distribuzione empirica del Time To Failure del sistema (Unreliability) e quindi anche della Reliability.

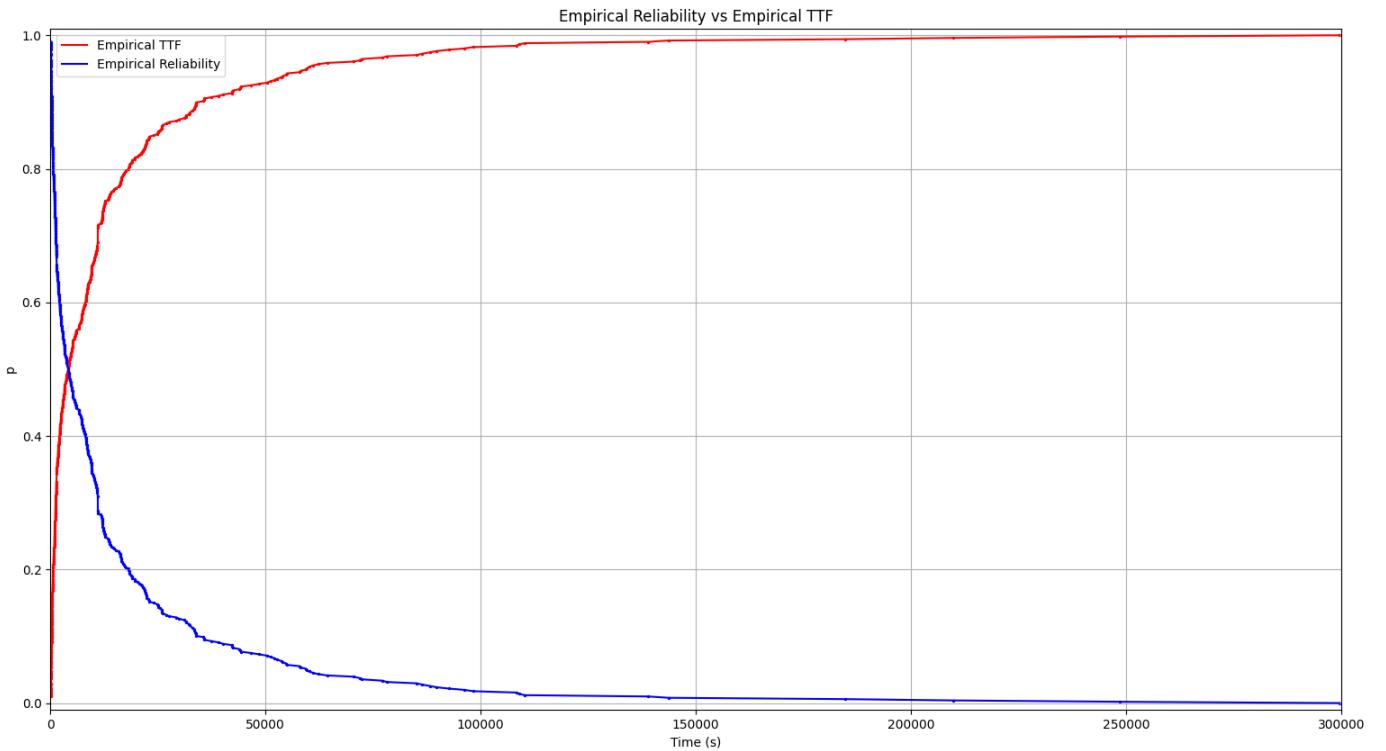


Figure 101: Unreliability e Reliability (Mercury).

Successivamente, si effettua il fitting della Reliability empirica con delle distribuzioni note. Sulla base della distribuzione che effettua il fitting migliore, è possibile dedurre il tipo di fallimento presente nei log. Verranno utilizzate tre distribuzioni:

- **Esponenziale:** relativa a guasti di natura isolata (**memoryless**), tipicamente moduli hardware indipendenti.

$$f(x, a, b) = \begin{cases} ae^{bx} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (62)$$

- **Iperesponenziale:** relativa a fallimenti causati da più moduli hardware distinti ed indipendenti.

$$f(x, a, b, c, d) = \begin{cases} ae^{bx} + ce^{dx} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (63)$$

- **Weibull:** relativa ad errori di accumulazione e failure di sistemi che sono in esecuzione per molto tempo causando degradazione dei componenti.

$$f(x, a, b) = \begin{cases} e^{-bx^a} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (64)$$

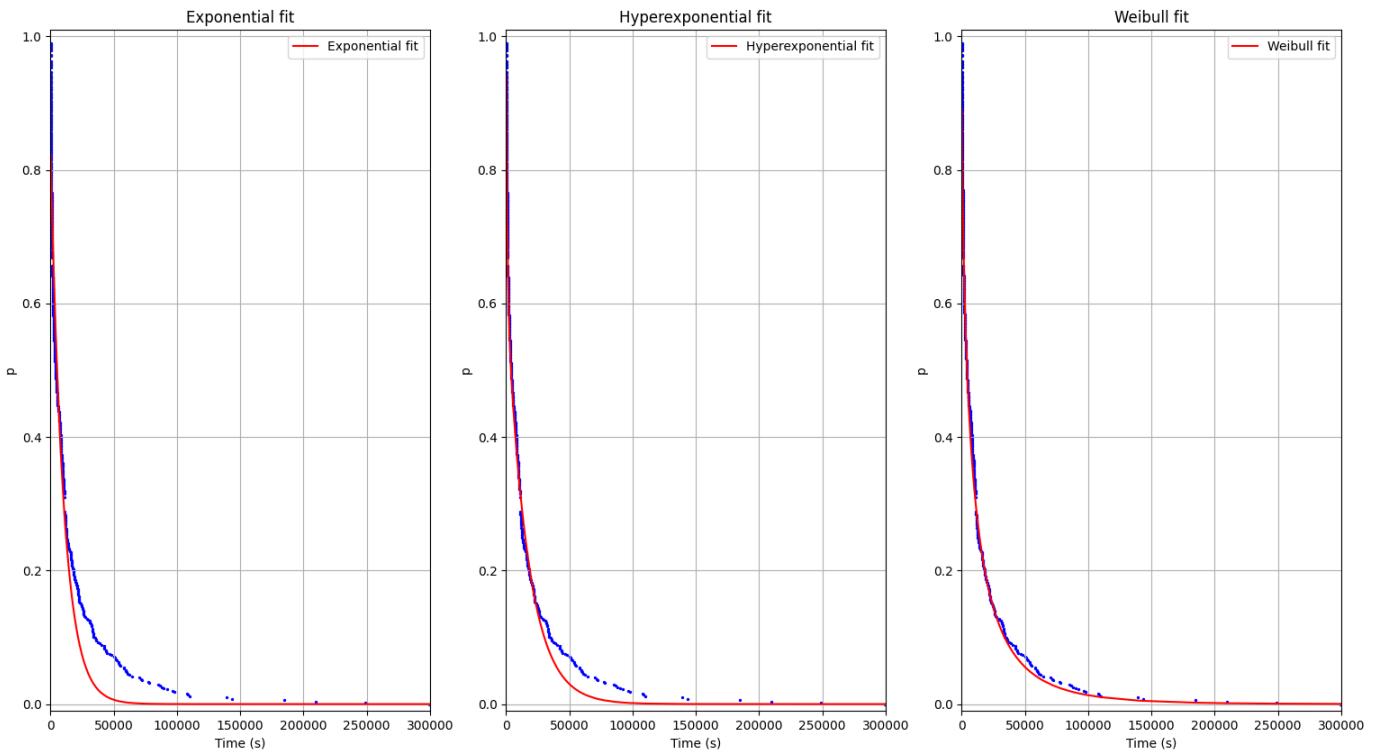


Figure 102: **Fitting della Reliability (Mercury).**

Per verificare la bontà dei fitting, per ogni distribuzione viene calcolato l' R^2 e viene effettuato il test di Kolmogorov-Smirnov.

```

Exponential fit pvalue: 3.888272486303904e-05
Exponential fit R2 score: 0.9534321903656312
Hyperexponential fit pvalue: 0.6772343435309346
Hyperexponential fit R2 score: 0.9961159745446708
Weibull fit pvalue: 0.07644653364060022
Weibull fit R2 score: 0.9917923348350616

```

Figure 103: **Goodness of Fit (Mercury).**

Dai risultati ottenuti è possibile osservare che tutte le distribuzioni presentano un R^2 molto alto. Il test di Kolmogorov-Smirnov rigetta l'ipotesi nulla per la distribuzione esponenziale. Tra la distribuzione iperesponenziale e quella di Weibull, quella iperesponenziale risulta essere il modello migliore. Questo ci porta a pensare che la Reliability del sistema Mercury dipende principalmente da fallimenti che sono riconducibili all'hardware.

7.2 CALCOLATORE BLUE GENE/L

In questa sezione verrà effettuata l'analisi dei log del calcolatore Blue Gene/L.

7.2.1 LOGGING COLLECTION E FILTERING

I dati sono raccolti in un file di log con 125624 entry (solo errori fatali del sistema). Il formato dei log è il seguente:

- **Timestamp**
- **Nodo di origine**

- Card di origine
- Messaggio di failure

7.2.2 DATA MANIPULATION

Come per il calcolatore Mercury, si procede ad individuare la dimensione della finestra di coalescenza.

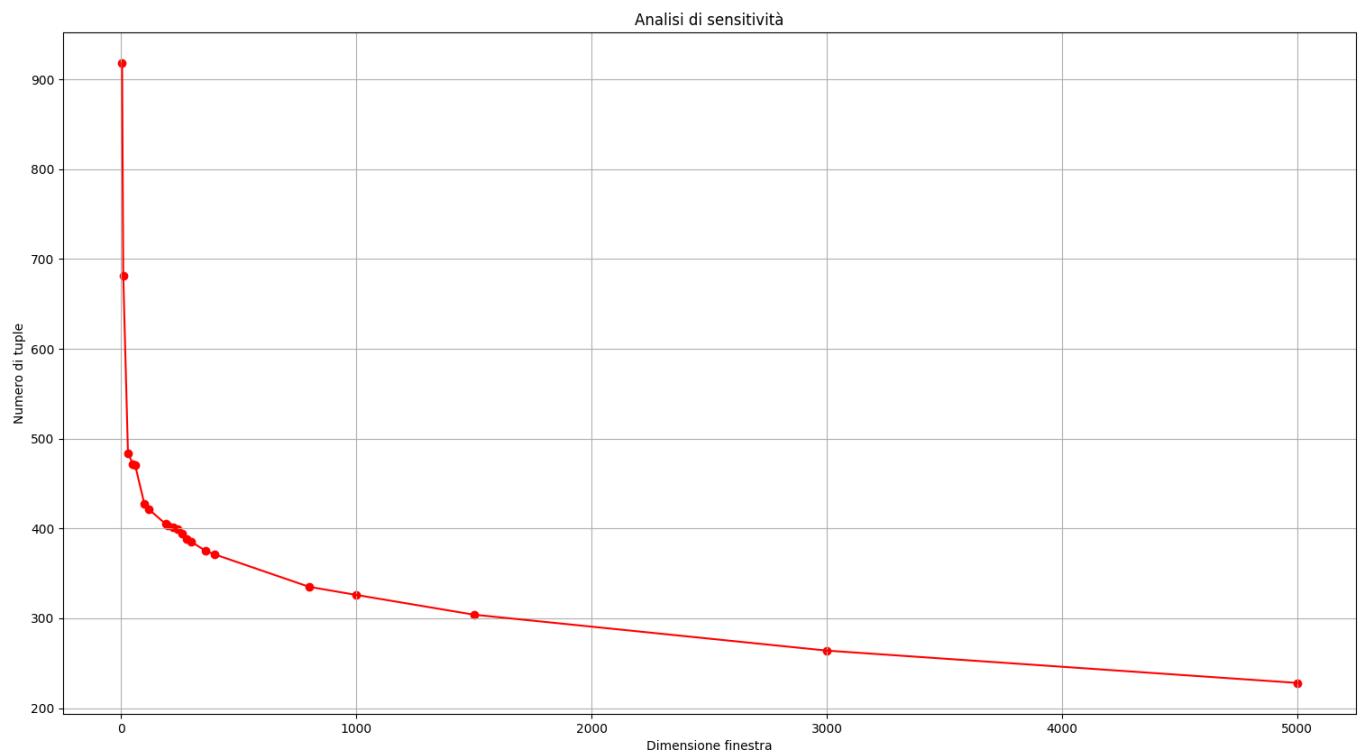


Figure 104: Analisi di sensitività (Blue Gene/L).

Si sceglie come valore di W, quello immediatamente successivo al valore corrispondente al ginocchio della curva. In questo caso è stata scelta una **finestra di coalescenza della durata di 200s**, che ci permetterà di ottenere **403 tuple**. Utilizzando uno script Python è stato analizzato il numero di troncamenti e collisioni, in particolare per questa W sono presenti 21 troncamenti (5.2%) e 39 collisioni (9.8%).

7.2.3 DATA ANALYSIS

Nell'ultima fase dell'FFDA viene effettuato uno studio sulla Reliability del sistema. Grazie al calcolo degli interarrivi (tempo che intercorre tra la prima entry di una tupla e l'ultima entry della tupla precedente), è possibile ottenere una distribuzione empirica del Time To Failure del sistema (Unreliability) e quindi anche della Reliability.

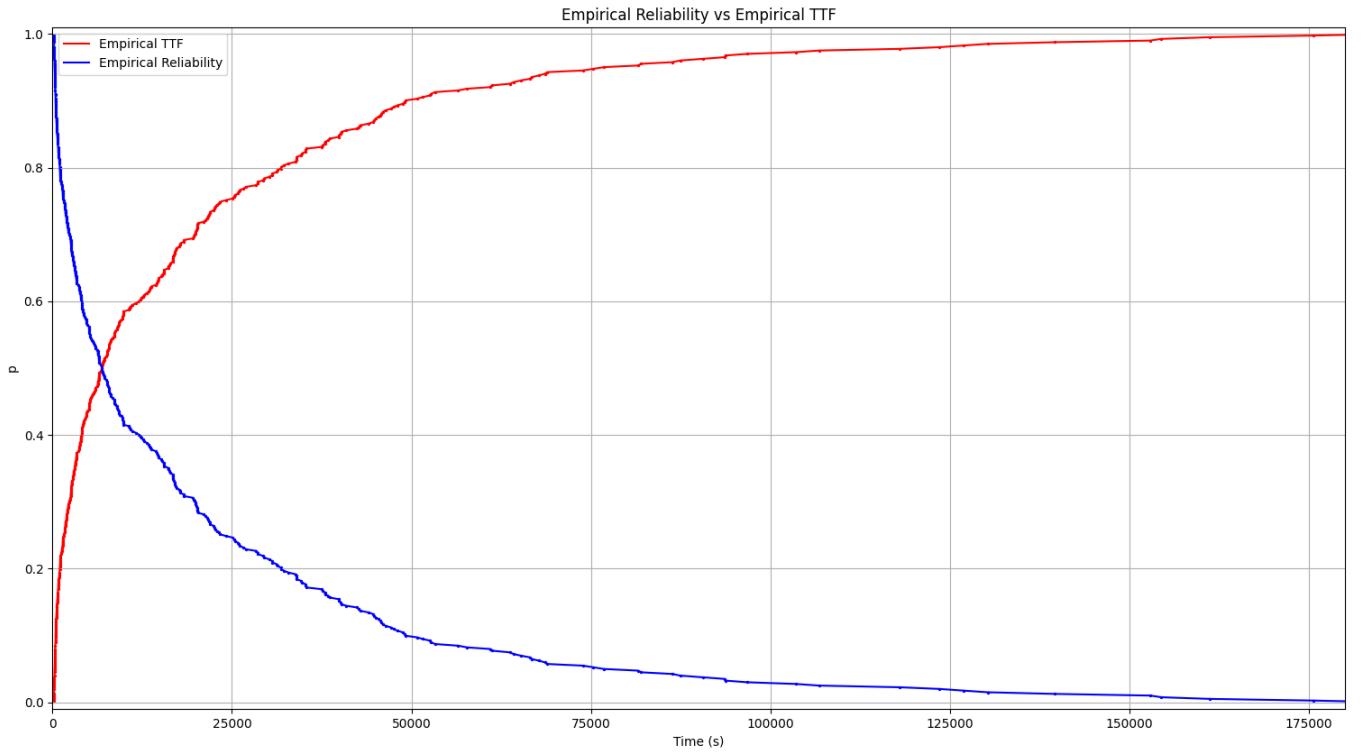


Figure 105: Unreliability e Reliability (Blue Gene/L).

Successivamente, si effettua il fitting della Reliability empirica con delle distribuzioni note. Sulla base della distribuzione che effettua il fitting migliore, è possibile dedurre il tipo di fallimento presente nei log. Verranno utilizzate tre distribuzioni:

- **Esponenziale:** relativa a guasti di natura isolata (**memoryless**), tipicamente moduli hardware indipendenti.

$$f(x, a, b) = \begin{cases} ae^{bx} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (65)$$

- **Iperesponenziale:** relativa a fallimenti causati da più moduli hardware distinti ed indipendenti.

$$f(x, a, b, c, d) = \begin{cases} ae^{bx} + ce^{dx} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (66)$$

- **Weibull:** relativa ad errori di accumulazione e failure di sistemi che sono in esecuzione per molto tempo causando degradazione dei componenti.

$$f(x, a, b) = \begin{cases} e^{-bx^a} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (67)$$

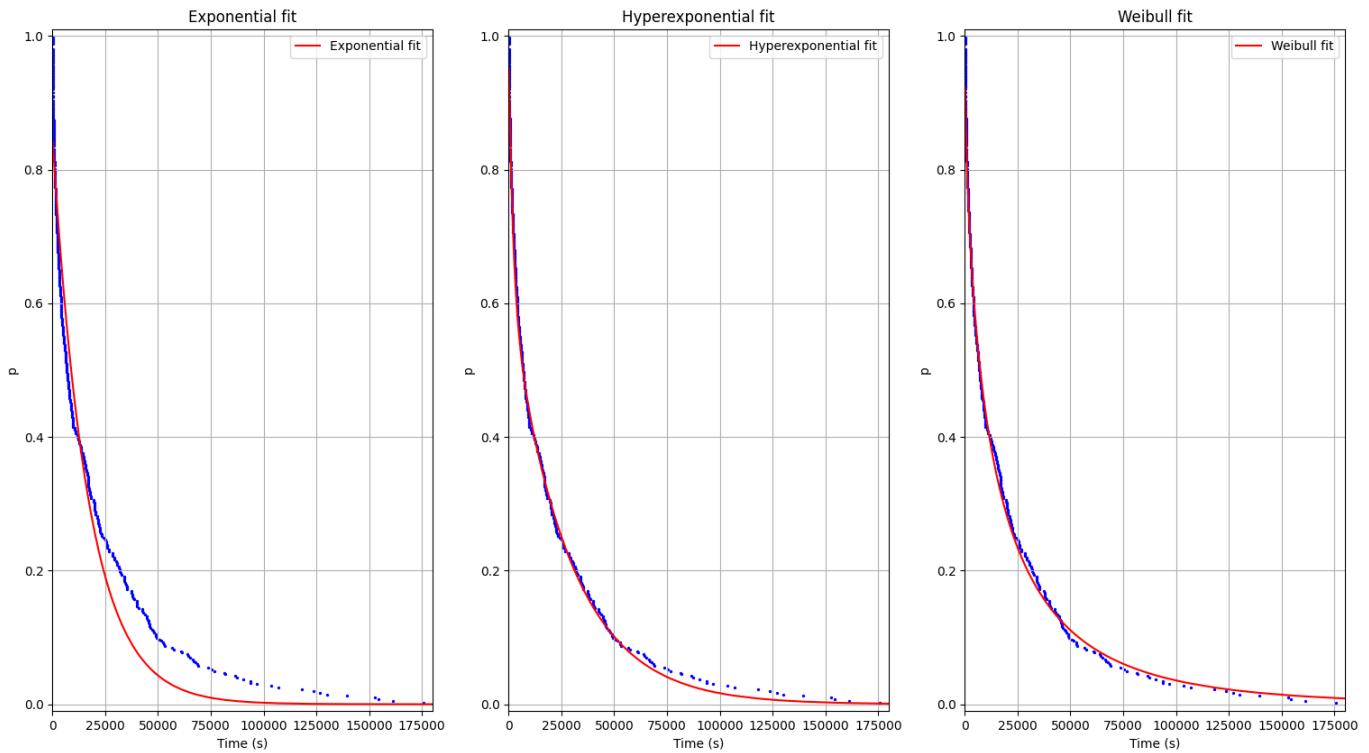


Figure 106: **Fitting della Reliability (Blue Gene/L).**

Per verificare la bontà dei fitting, per ogni distribuzione viene calcolato l' R^2 e viene effettuato il test di Kolmogorov-Smirnov.

```
Exponential fit pvalue: 0.0005149260780329968
Exponential fit R2 score: 0.956336021391305
Hyperexponential fit pvalue: 0.9019408413449534
Hyperexponential fit R2 score: 0.9978395920358024
Weibull fit pvalue: 0.27297826743702264
Weibull fit R2 score: 0.9955985796472769
```

Figure 107: **Goodness of Fit (Blue Gene/L).**

Dai risultati ottenuti è possibile osservare che tutte le distribuzioni presentano un R^2 molto alto. Il test di Kolmogorov-Smirnov rigetta l'ipotesi nulla per la distribuzione esponenziale. Tra la distribuzione iperesponenziale e quella di Weibull, quella iperesponenziale risulta essere il modello migliore. Questo ci porta a pensare che la Reliability del sistema Blue Gene/L dipende principalmente da fallimenti che sono riconducibili all'hardware.

7.3 CONFRONTO TRA MERCURY E BLUE GENE/L

È possibile confrontare le Reliability dei due calcolatori: È possibile inoltre calcolare il MTTF:

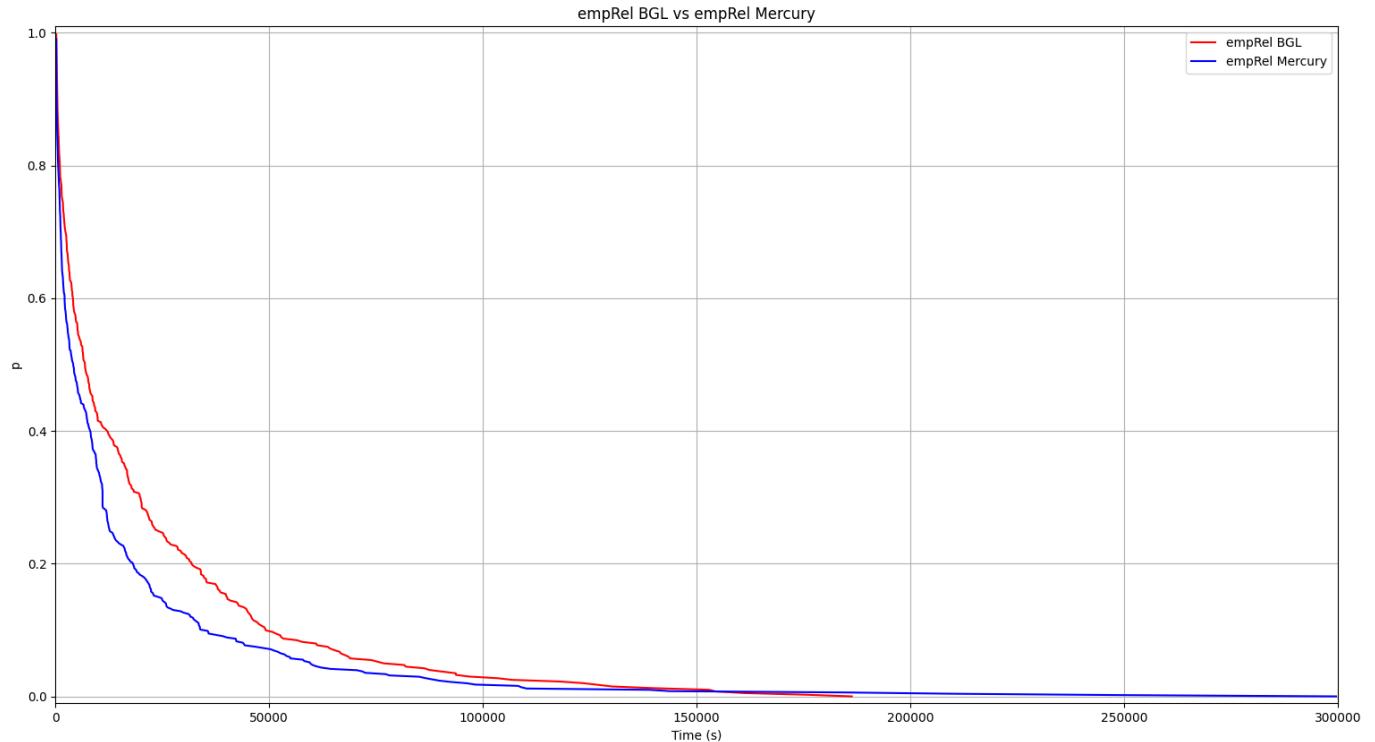


Figure 108: Confronto Reliability Mercury e Blue Gene/L.

$$MTTF_{Mercury} = \int_0^{+\infty} R_{Mercury}(t)dt = 1.35 \cdot 10^4 s \quad (68)$$

$$MTTF_{BlueGene} = \int_0^{+\infty} R_{BlueGene}(t)dt = 1.88 \cdot 10^4 s \quad (69)$$

Sia graficamente che numericamente, il calcolatore Blue Gene risulta essere più affidabile del calcolatore Mercury.

7.4 ANALISI FINESTRA DI COALESCENZA

In questa sezione viene effettuata un'analisi relativa alle finestre di coalescenza dei singoli nodi per ogni calcolatore. Inoltre per il calcolatore Mercury vengono analizzate le finestre di coalescenza relative alle varie categorie di errore. Verranno analizzati solamente i nodi (e le categorie di errore) più frequenti, estratti tramite uno script Python. Una volta filtrati i log, la dimensione delle finestre di coalescenza viene ottenuta come in precedenza, mediante l'analisi della sensitività.

7.4.1 NODI MERCURY

Vengono paragonate le finestre di coalescenza dei nodi tg-c401, tg-c238, tg-c572, tg-s044 e tg-master del calcolatore Mercury.

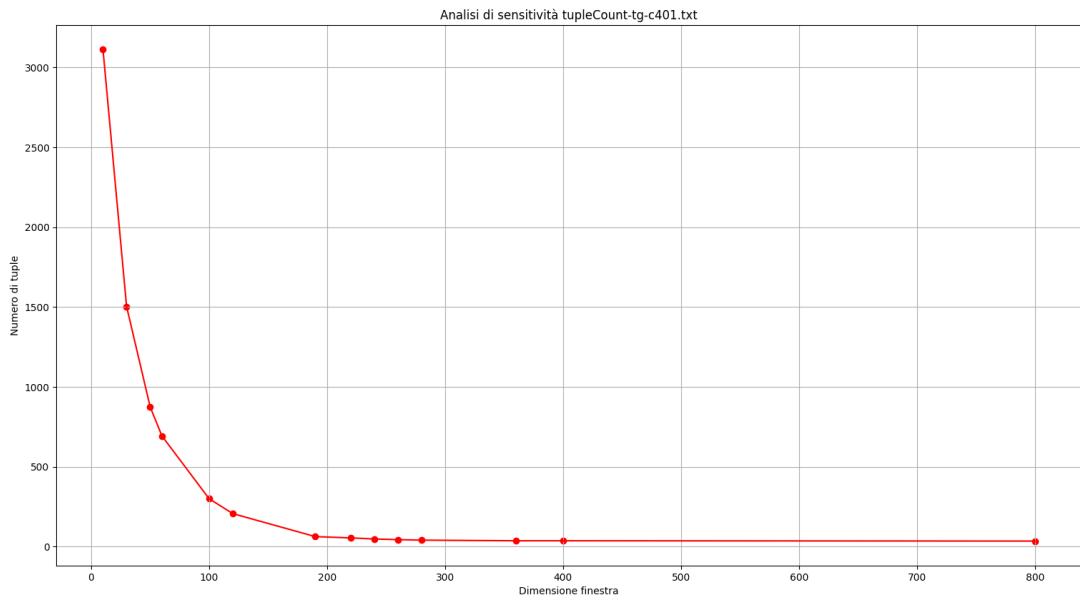


Figure 109: Analisi di sensitività tg-c401.

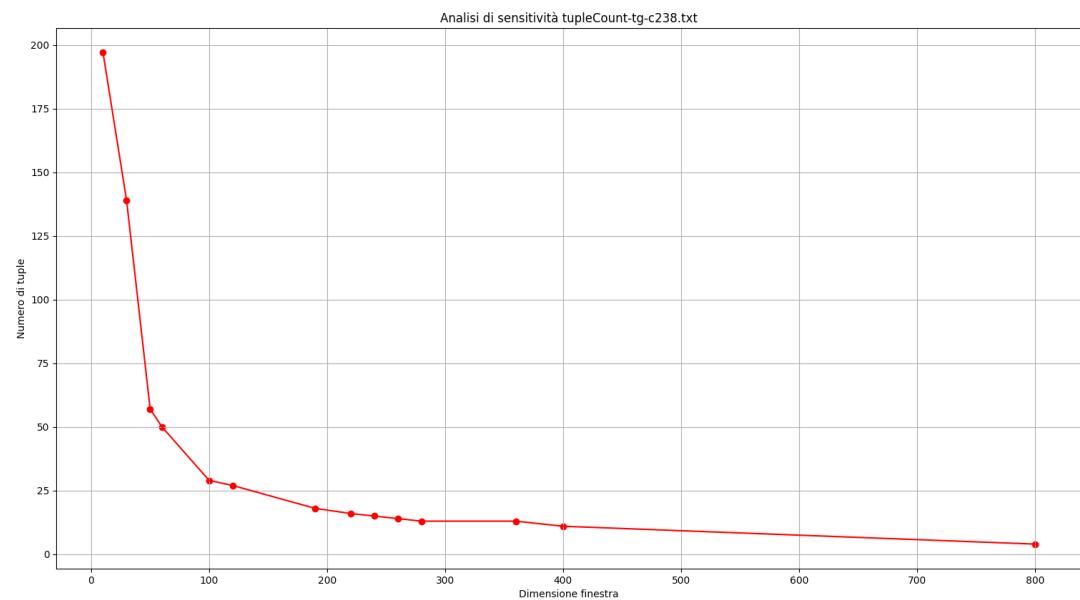


Figure 110: Analisi di sensitività tg-c238.

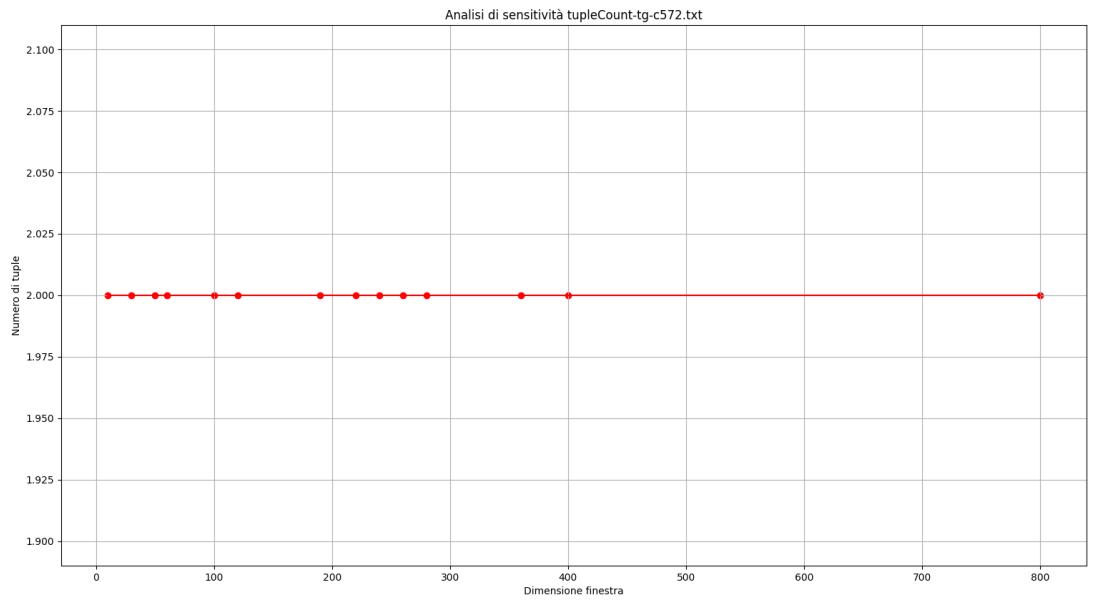


Figure 111: Analisi di sensitività tg-c572.

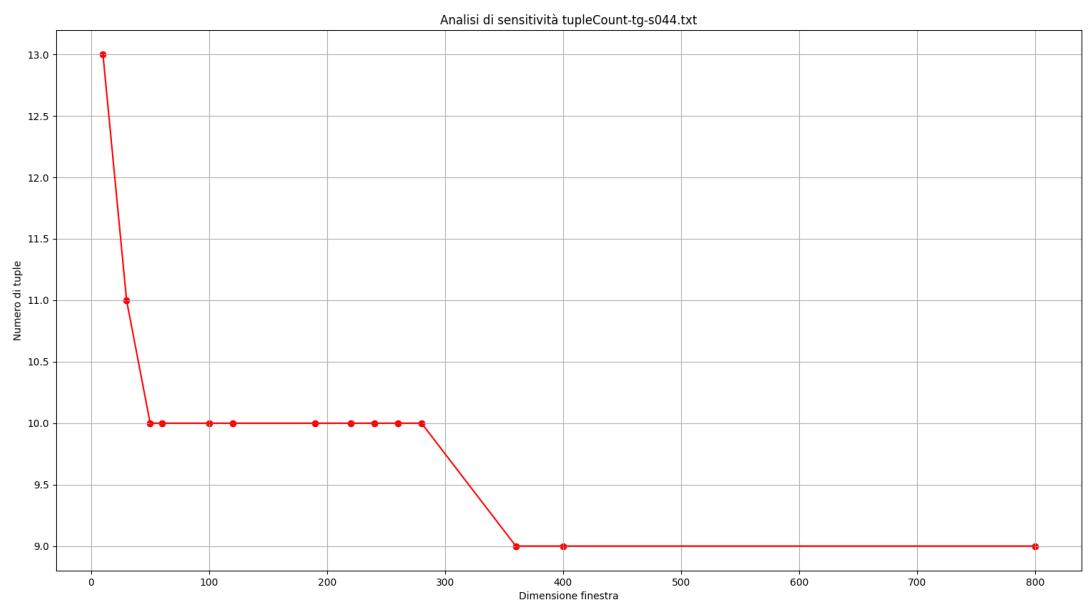


Figure 112: Analisi di sensitività tg-s044.

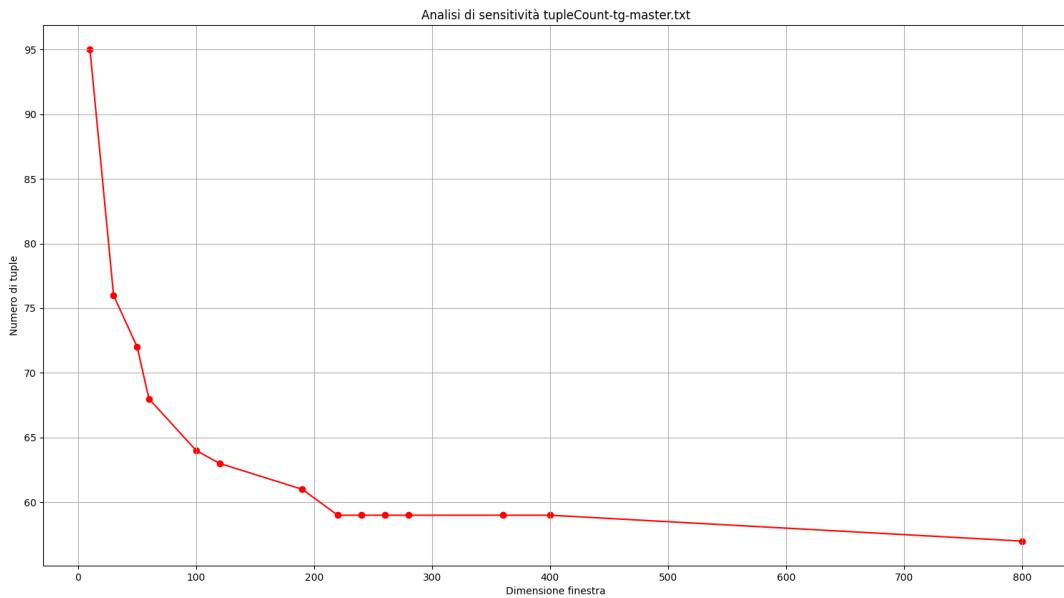


Figure 113: **Analisi di sensitività tg-master.**

Come è possibile osservare dai risultati ottenuti, per i nodi computazionali (tg-cX, ad eccezione del tg-c572 dove la W non ha importanza) e per il nodo tg-master, è possibile utilizzare una finestra di coalescenza di dimensione pari a quella utilizzata per l'intero sistema, mentre per i nodi di storage (tg-sX), la scelta ricade su una W minore.

7.4.2 CATEGORIE DI ERRORE MERCURY

Vengono paragonate le finestre di coalescenza delle categorie di errore DEV, MEM, I-O, NET, PRO ed OTH del calcolatore Mercury.

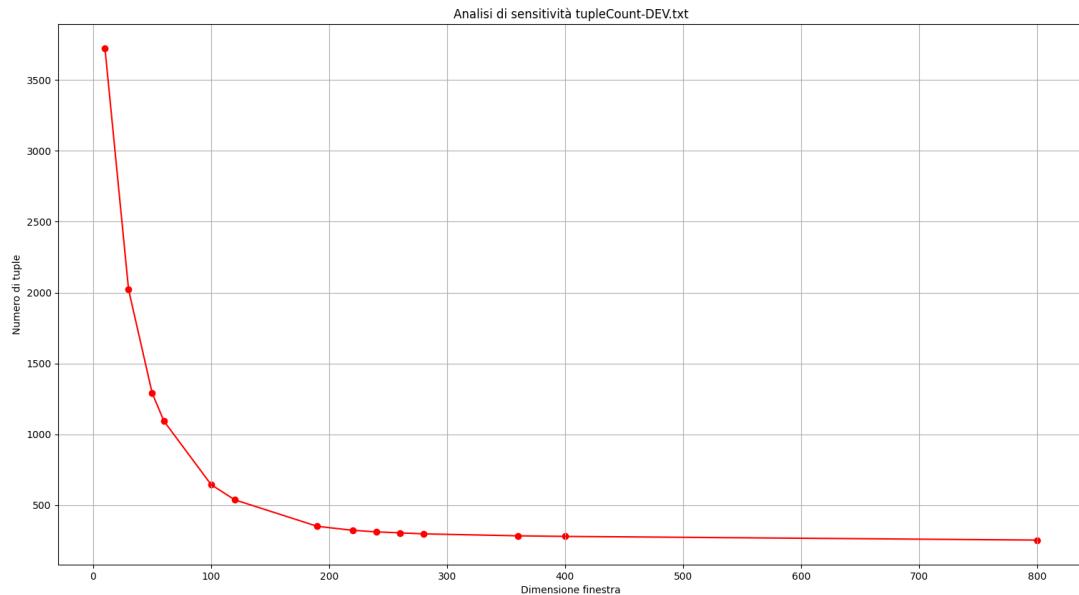


Figure 114: **Analisi di sensitività DEV.**

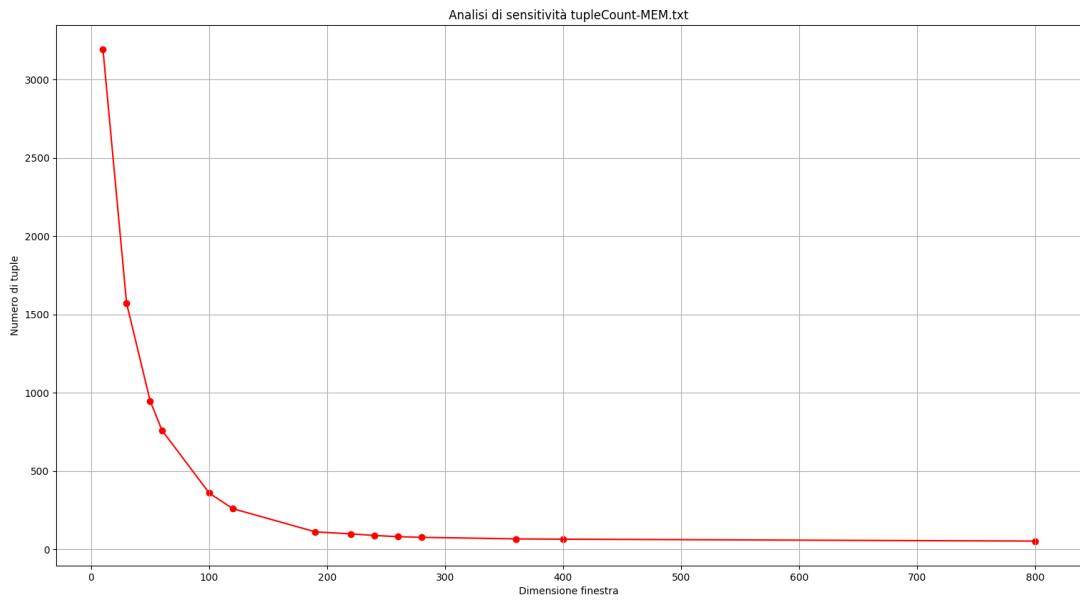


Figure 115: Analisi di sensitività MEM.

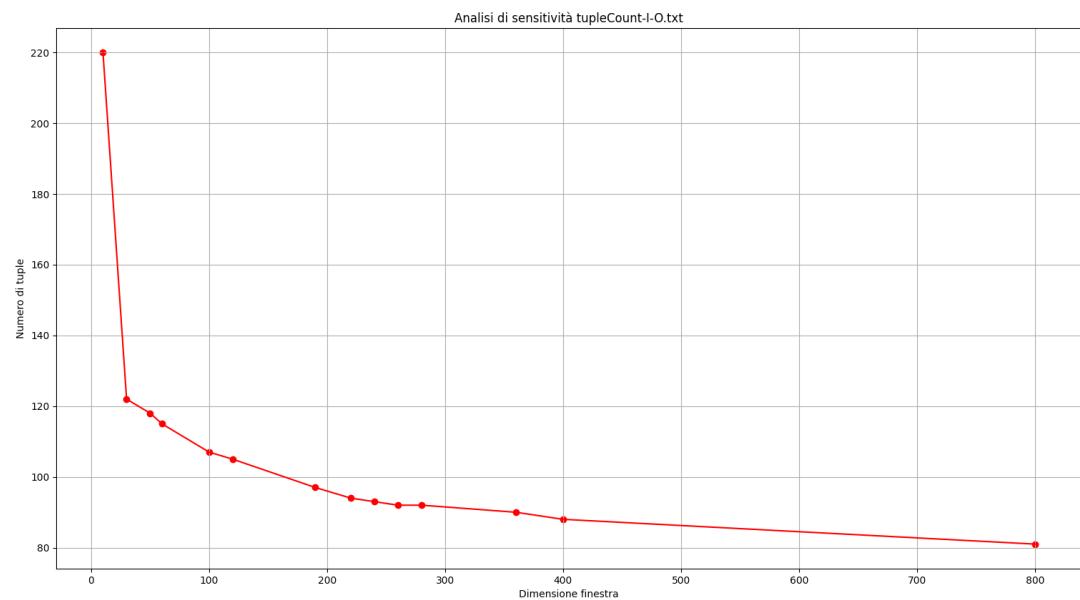


Figure 116: Analisi di sensitività I-O.

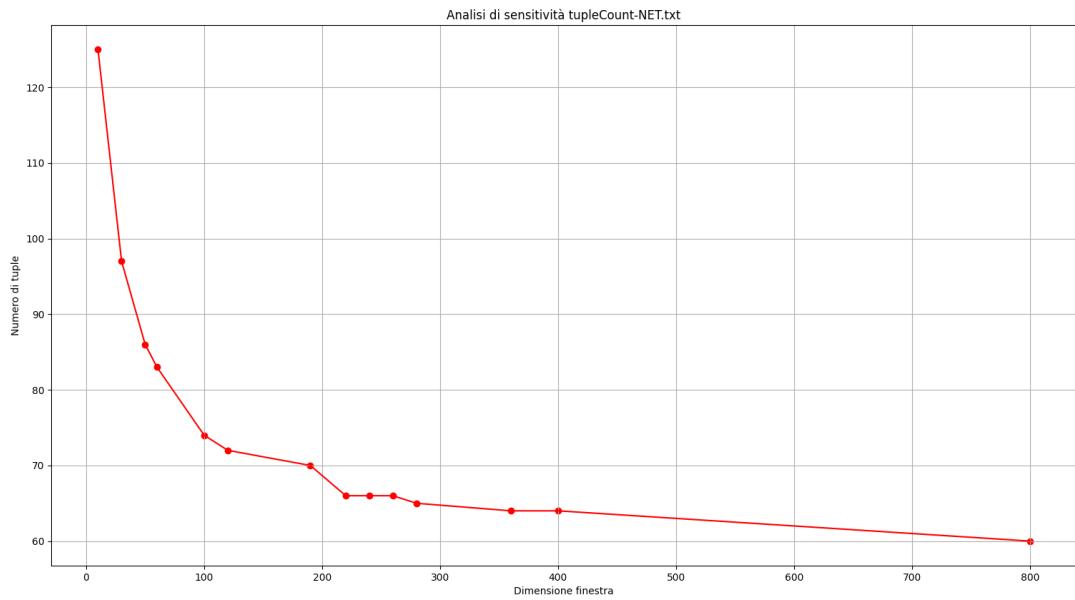


Figure 117: Analisi di sensitività NET.

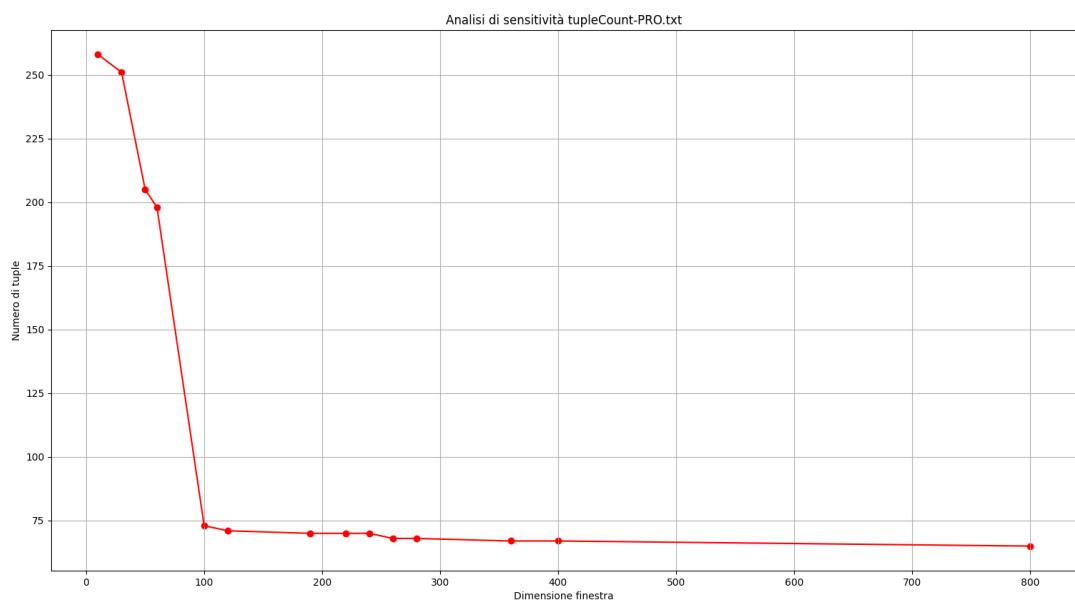


Figure 118: Analisi di sensitività PRO.

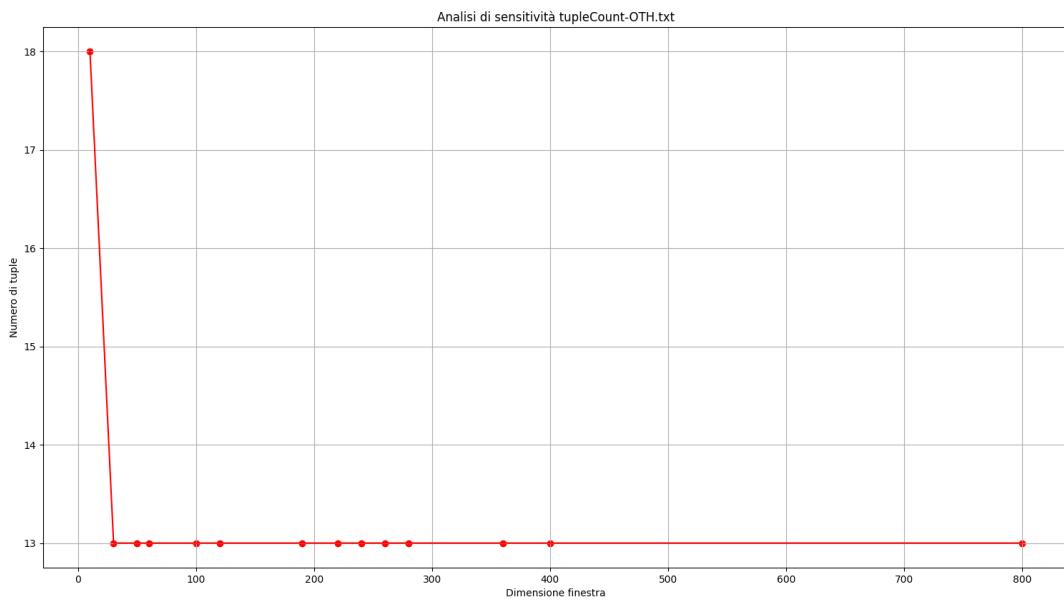


Figure 119: **Analisi di sensitività OTH.**

Dai risultati ottenuti è possibile osservare che per la categorie di errore DEV, MEM, I-O e NET è possibile utilizzare la stessa finestra di coalescenza di quella del sistema. Per OTH e PRO (le classi meno frequenti), la scelta ricade su finestre di coalescenza più brevi.

7.4.3 NODI BLUE GENE/L

Vengono paragonate le finestre di coalescenza dei nodi R71-M0-N4, R12-M0-N0, R63-M0-N2, R63-M0-N0, R03-M1-NF del calcolatore Blue Gene.

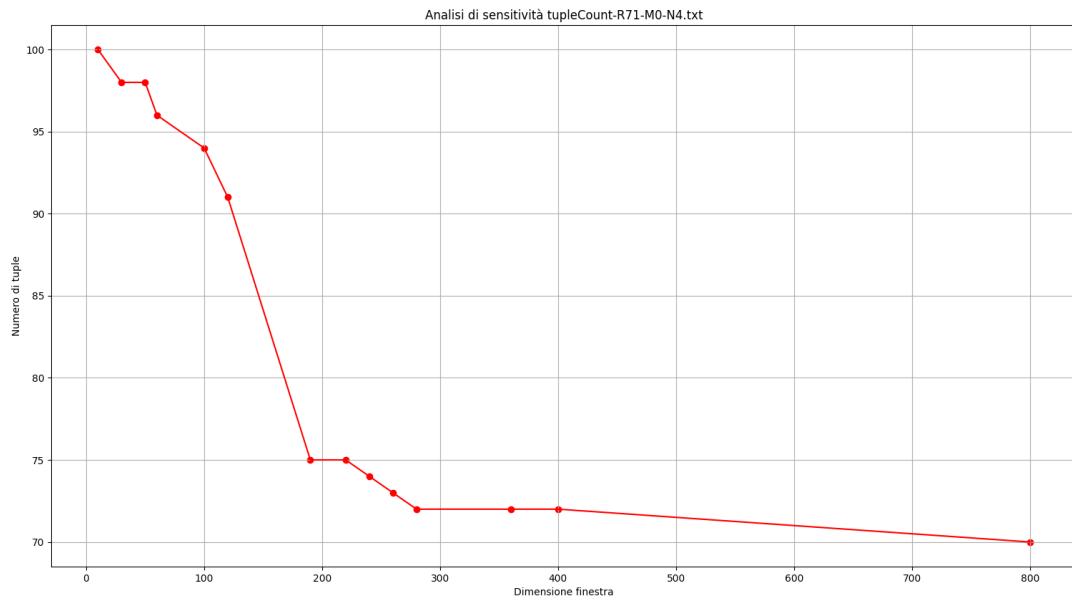


Figure 120: **Analisi di sensitività R71-M0-N4.**

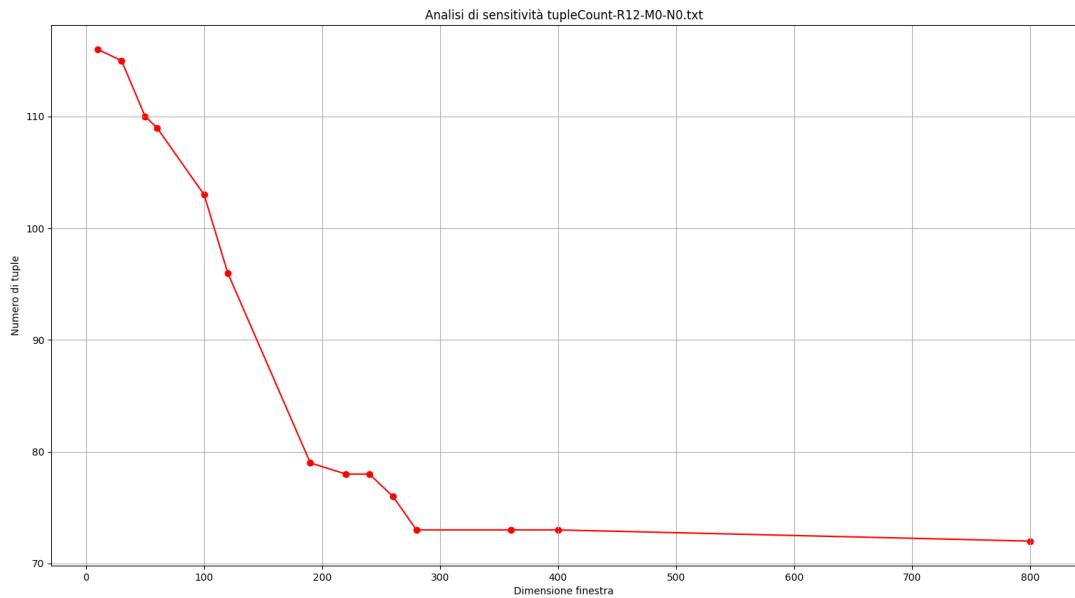


Figure 121: Analisi di sensitività R12-M0-N0.

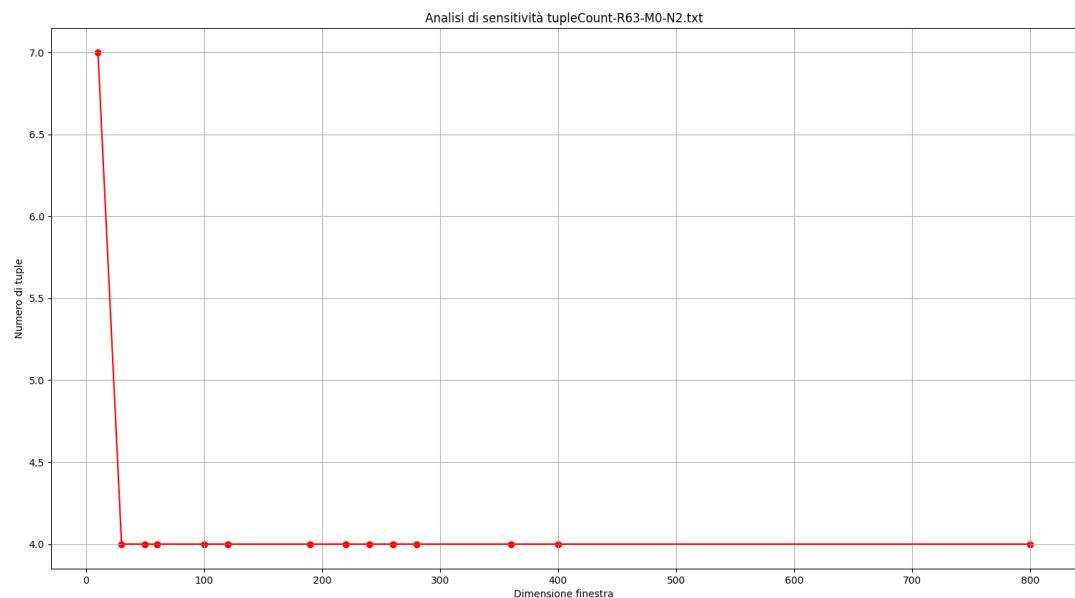


Figure 122: Analisi di sensitività R63-M0-N2.

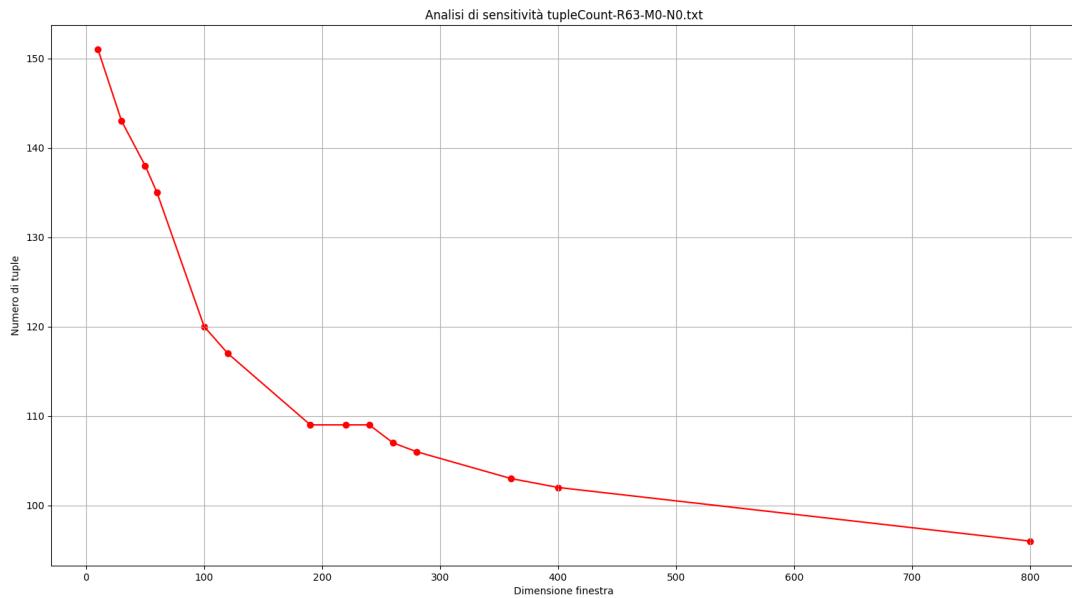


Figure 123: **Analisi di sensitività R63-M0-N0.**

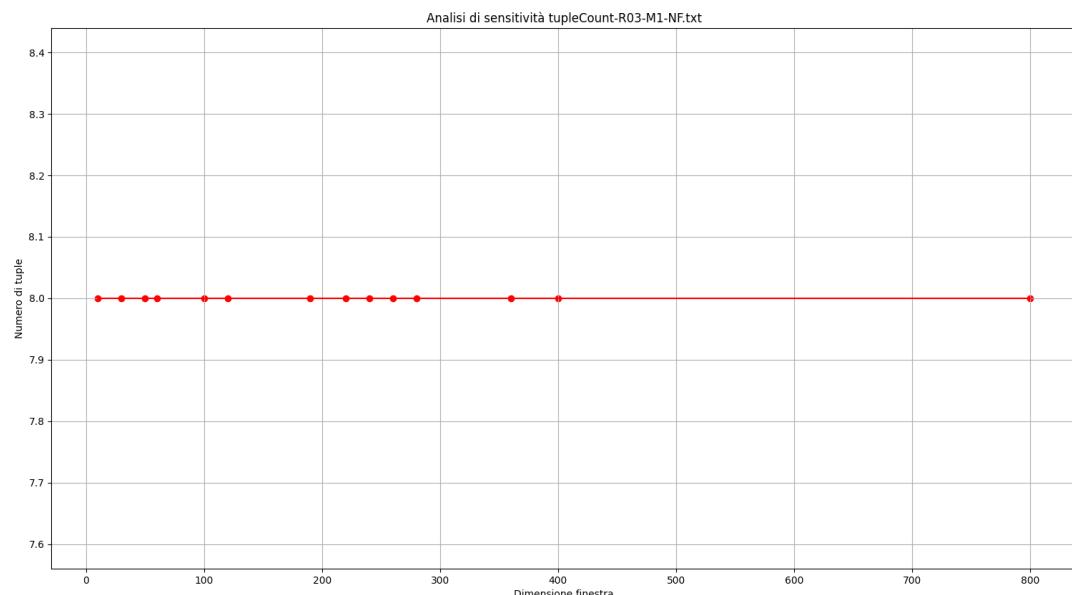


Figure 124: **Analisi di sensitività R03-M1-NF.**

Dall’analisi dei risultati ottenuti è possibile osservare che per i nodi R71-M0-N4, R12-M0-N0 e R63-M0-N0, la dimensione della finestra di coalescenza è leggermente superiore a quella del sistema. Per il nodo R03-M1-NF la dimensione della finestra di coalescenza è irrilevante mentre per R63-M0-N2, la scelta ricade su una finestra notevolmente più piccola.

7.5 RELIABILITY DEL SISTEMA VS RELIABILITY DEI NODI

In questa sezione si confronteranno le Reliability dei singoli nodi e la Reliability del sistema complessivo, sia per il calcolatore Mercury che per il calcolatore Blue Gene. La Reliability dei singoli nodi, è calcolata come quella del sistema in precedenza.

7.5.1 CALCOLATORE MERCURY

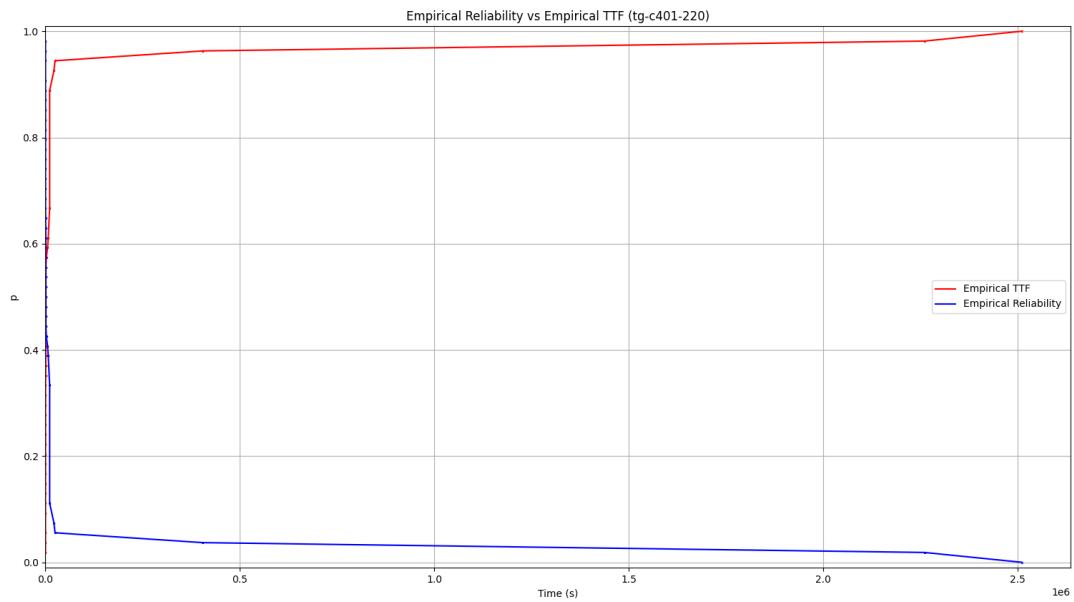


Figure 125: Reliability tg-c401.

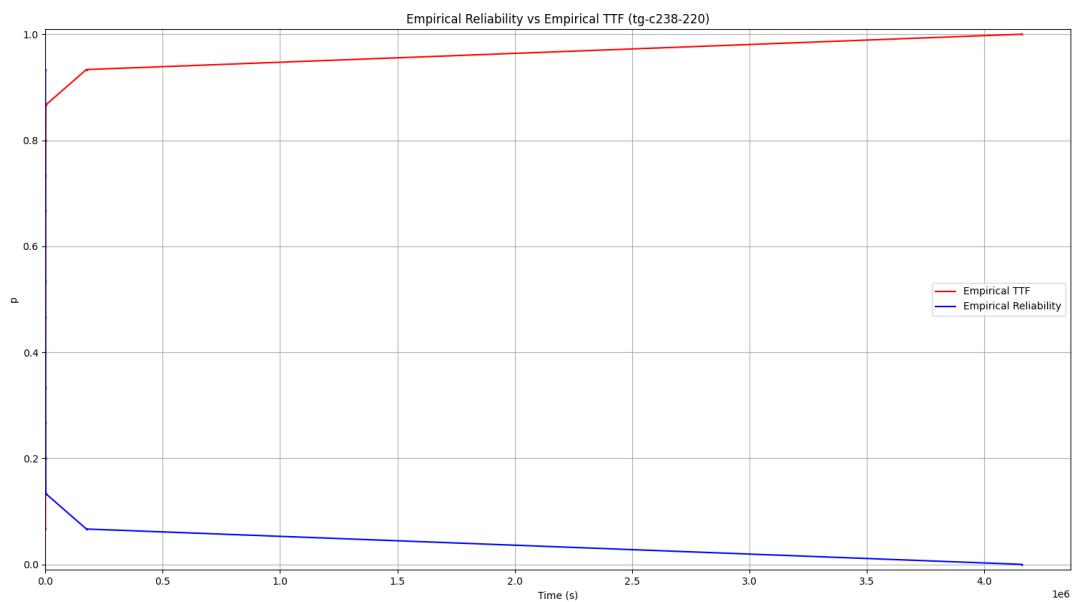


Figure 126: Reliability tg-c238.

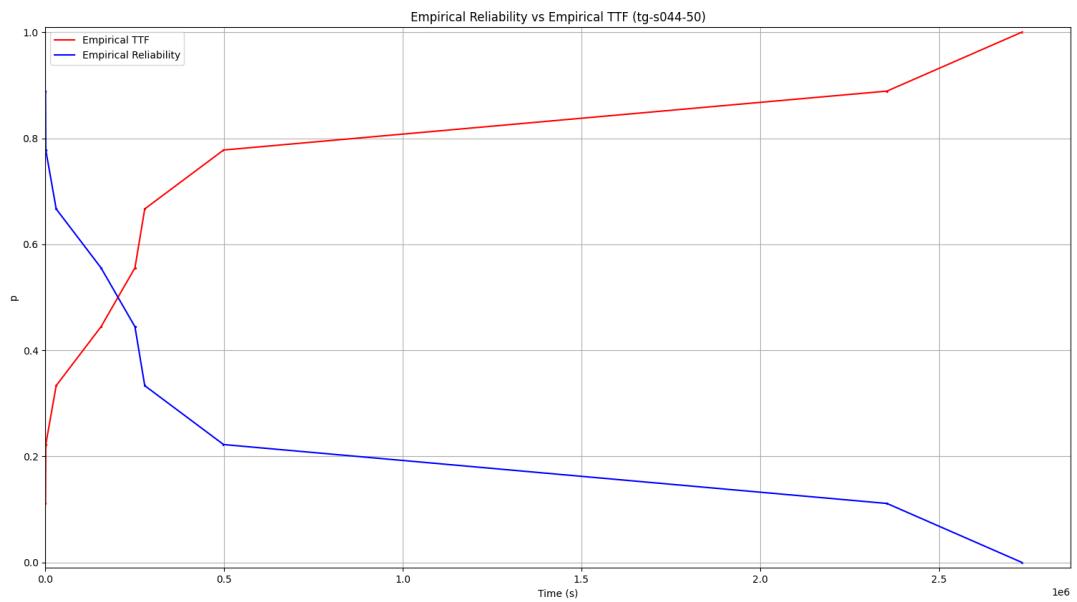


Figure 127: **Reliability tg-s044.**

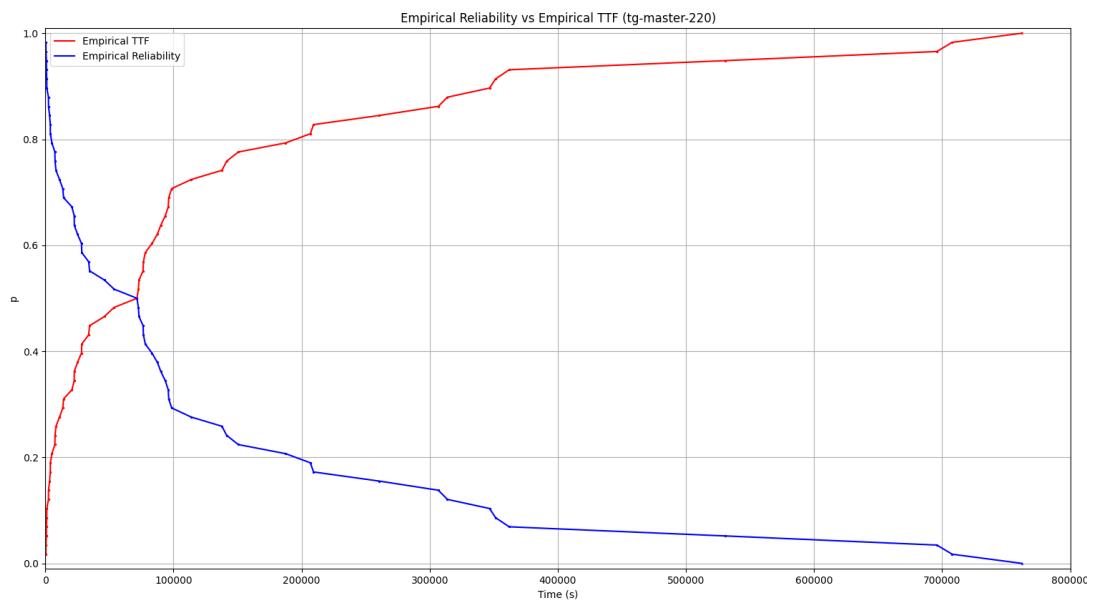


Figure 128: **Reliability tg-master.**

Infine vengono confrontate complessivamente le Reliability dei nodi con quella del sistema.

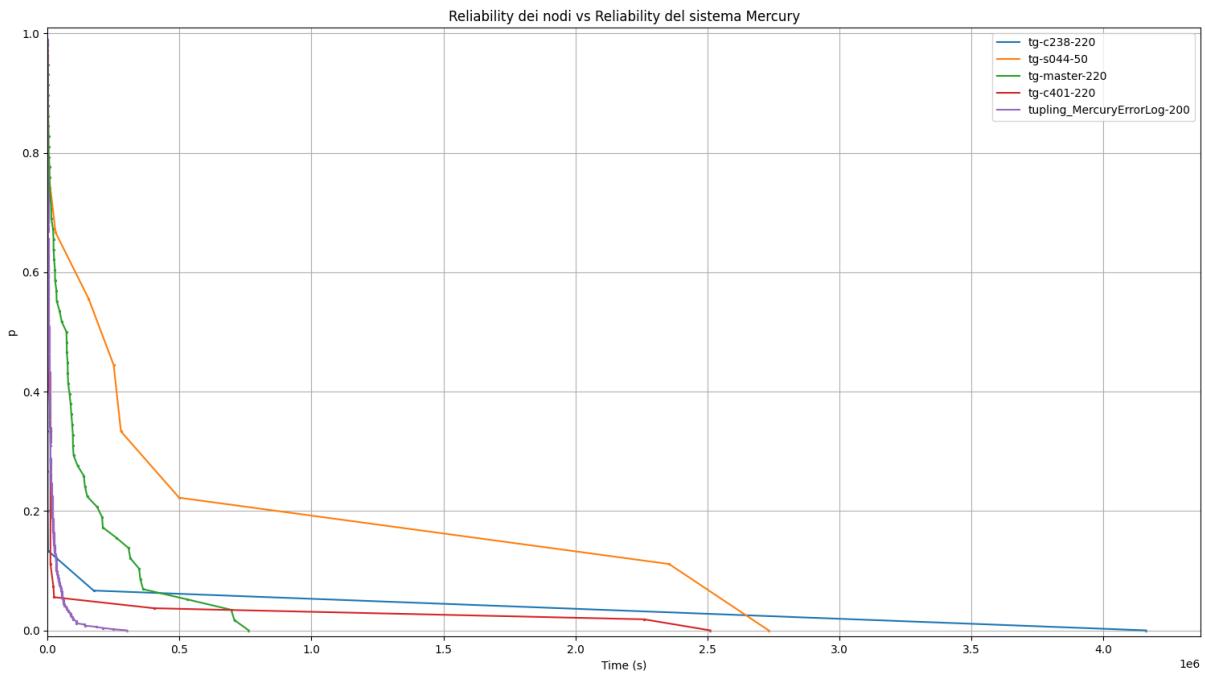


Figure 129: **Reliability a confronto.**

Com'è possibile osservare dai risultati ottenuti, la Reliability dei singoli nodi è maggiore di quella del sistema. Inoltre i nodi computazionali hanno una Reliability il cui andamento è simile a quella del sistema.

7.5.2 CALCOLATORE BLUE GENE/L

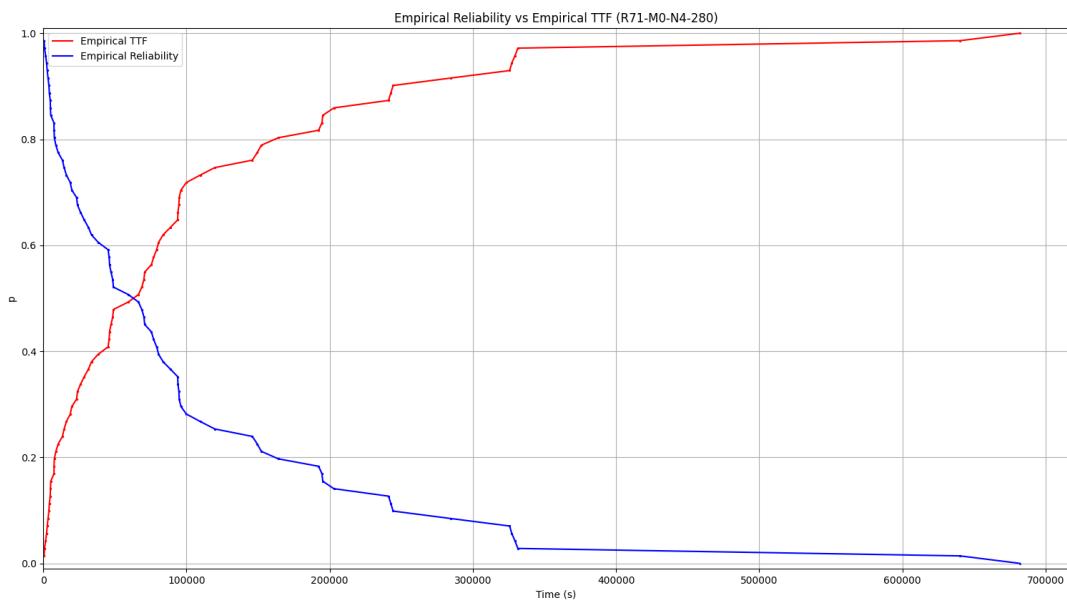


Figure 130: **Reliability R71-M0-N4.**

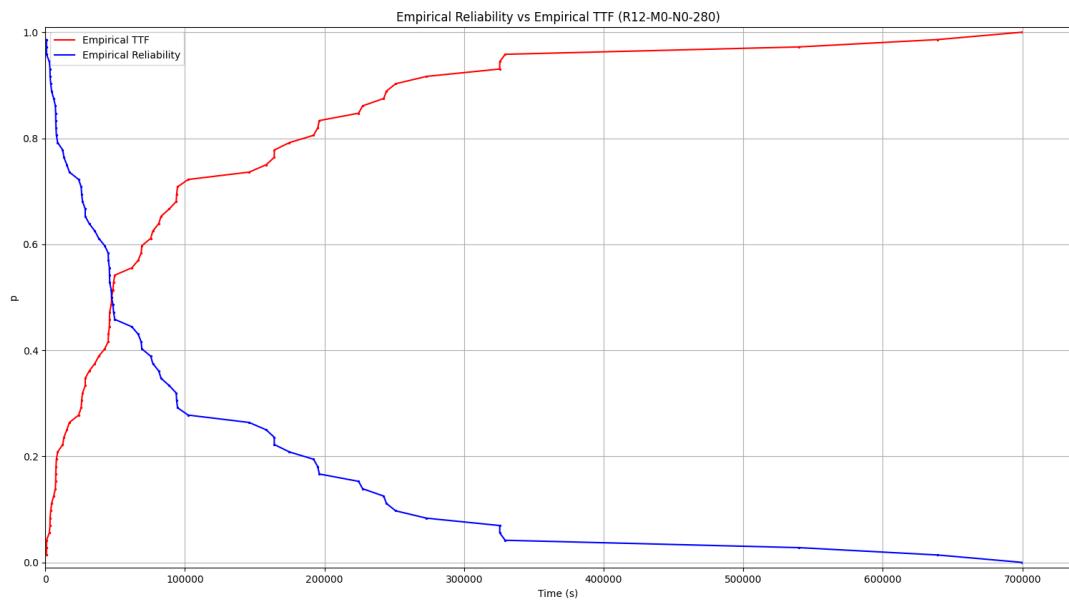


Figure 131: Reliability R12-M0-N0.

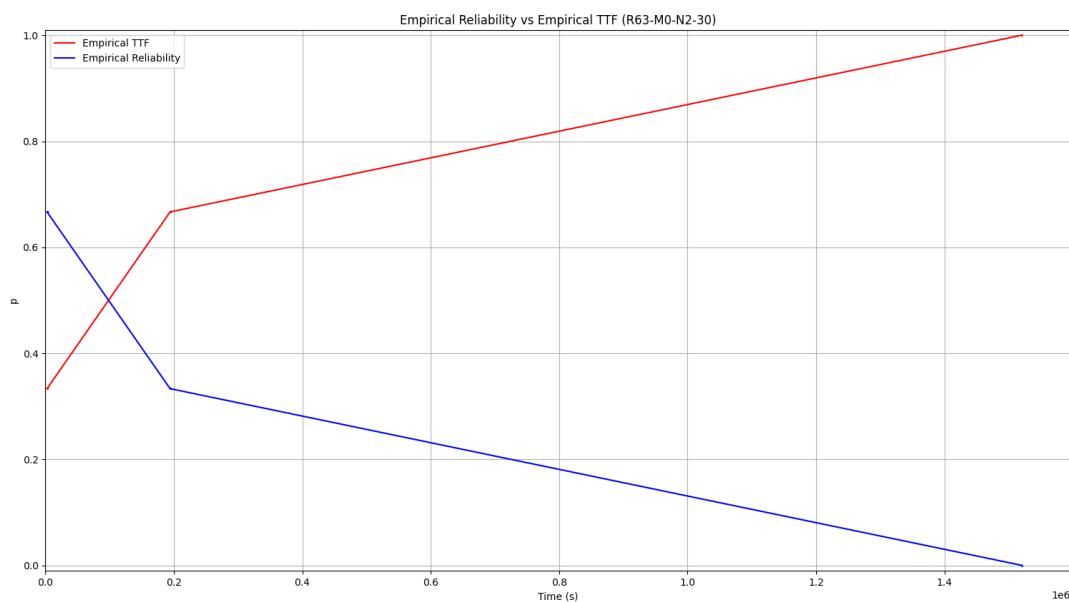


Figure 132: Reliability R63-M0-N2.

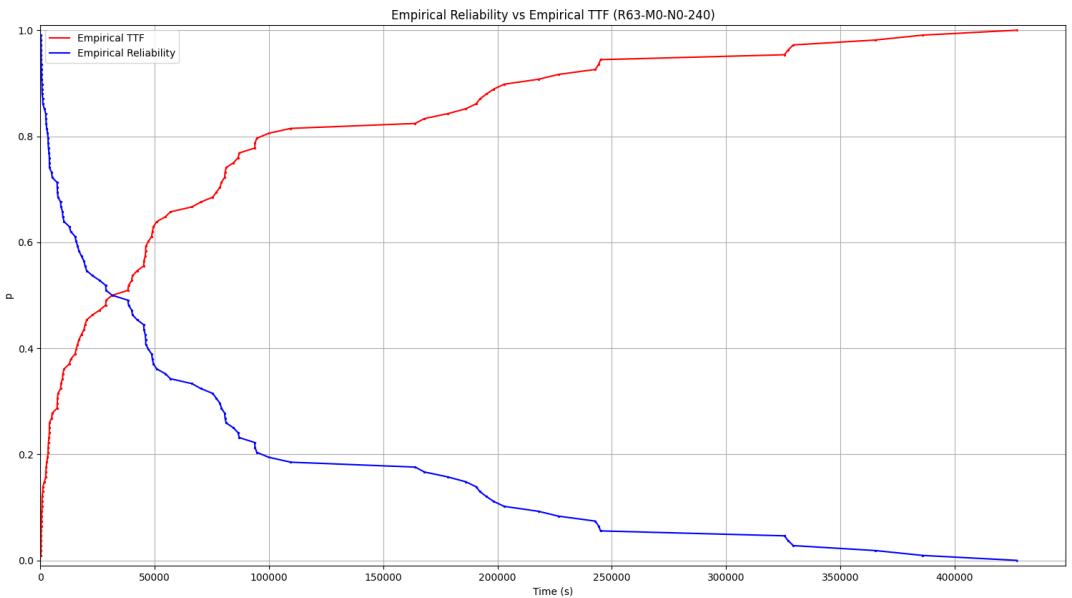


Figure 133: Reliability R63-M0-N0.

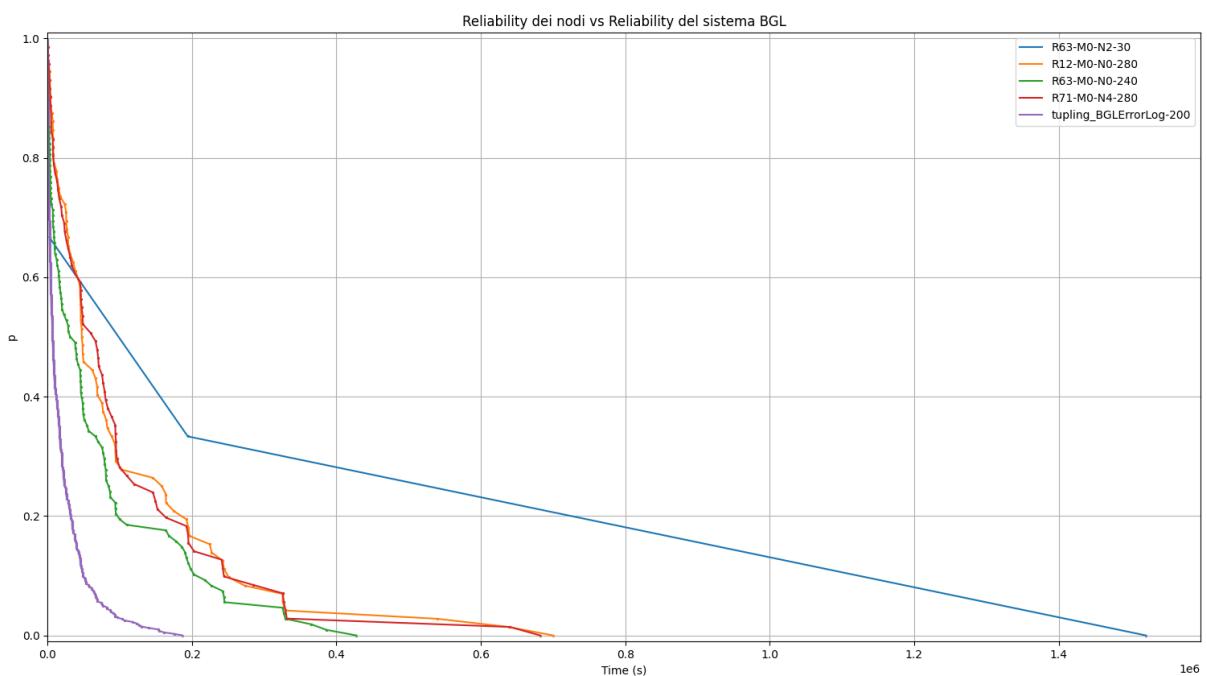


Figure 134: Reliability a confronto.

Com'è possibile osservare dai risultati ottenuti, la Reliability dei singoli nodi è maggiore di quella del sistema.

7.6 RELIABILITY BOTTLENECKS

In questa sezione vengono individuati eventuali Bottleneck del sistema. Mediante l'utilizzo di uno script Python è possibile individuare i nodi che causano più failure all'interno di ciascun calcolatore.

7.6.1 CALCOLATORE MERCURY

	Nodo	# Entry
1	tg-c401	62340
2	tg-master	4098
3	tg-c572	4030
4	tg-s044	3224
5	tg-c238	1273
6	tg-c242	1067
7	tg-c648	643
8	tg-login3	382
9	tg-c117	268
10	tg-c669	267

Table 22: **Bottleneck Mercury (nodi)**.

	Errore	# Entry
1	DEV	57248
2	MEM	12819
3	I-O	3702
4	NET	3702
5	PRO	1504
6	OTH	34

Table 23: **Bottleneck Mercury (errori)**.

Dalle tabelle è possibile osservare che il nodo tg-c401 è il nodo predominante nei log, e quindi è un Bottleneck per l'intero sistema. Inoltre, ragionando in maniera analoga per le categorie di errore, si osserva che le categorie DEV e MEM, sono quelle predominanti, costituendo un Bottleneck per il sistema.

7.6.2 CALCOLATORE BLUE GENE/L

	Nodo	# Entry
1	R71-M0-N4	1716
2	R12-M0-N0	1563
3	R63-M0-N2	976
4	R03-M1-NF	960
5	R63-M0-N0	791
6	R36-M1-N0	788
7	R62-M0-N4	515
8	R63-M0-NC	460
9	R63-M0-N8	454
10	R63-M0-N4	452

Table 24: **Bottleneck Blue Gene (nodi).**

Per il calcolatore Blue Gene non si individua nessun Bottleneck, non essendoci un nodo particolarmente frequente rispetto gli altri.

7.7 RELIABILITY DI NODI SIMILI

In questa sezione verranno confrontate le Reliability di nodi funzionalmente simili.

7.7.1 CALCOLATORE MERCURY

Per il calcolatore Mercury sono stati selezionati i nodi tg-c401 e tg-c238, entrambi nodi computazionali. Si inizia col paragonare le finestre di coalescenza e successivamente le loro Reliability.

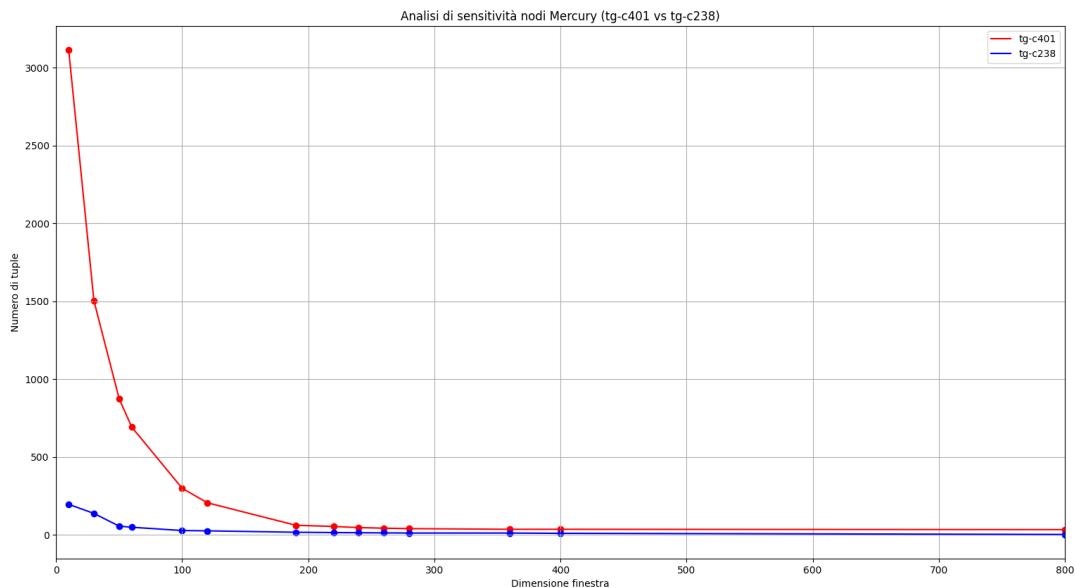


Figure 135: Confronto finestre di coalescenza tg-c401 e tg-c238.

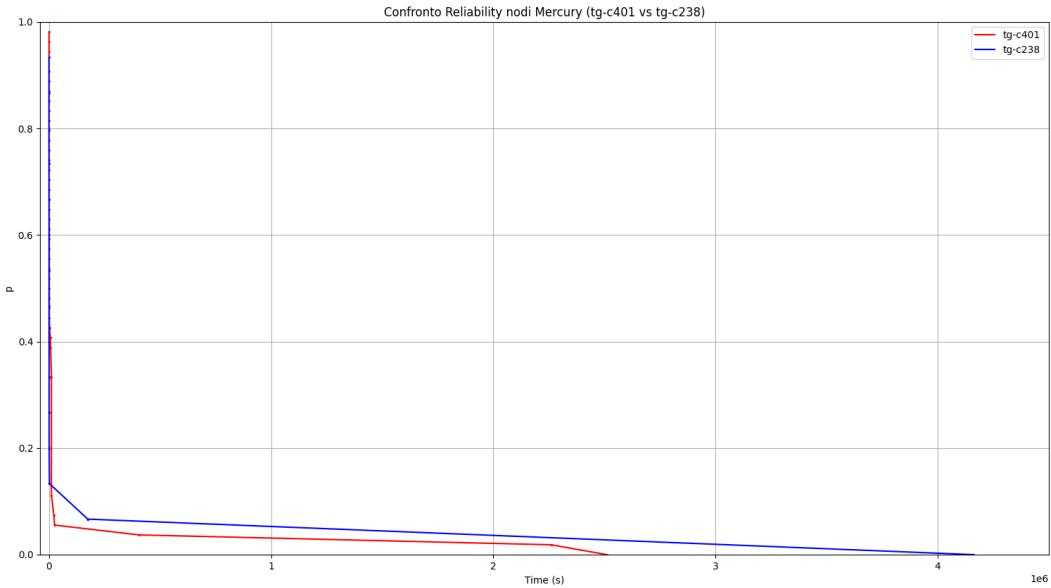


Figure 136: **Confronto Reliability tg-c401 e tg-c238.**

I due nodi hanno dimensioni della finestra di coalescenza comparabili, così come la Reliability. Dunque è possibile concludere che presentano parametri di affidabilità simili.

7.7.2 CALCOLATORE MERCURY

Per il calcolatore Blue Gene sono stati selezionati i nodi R71-M0-N4 e R12-M0-N0, entrambi nodi computazionali. Si inizia col paragonare le finestre di coalescenza e successivamente le loro Reliability.

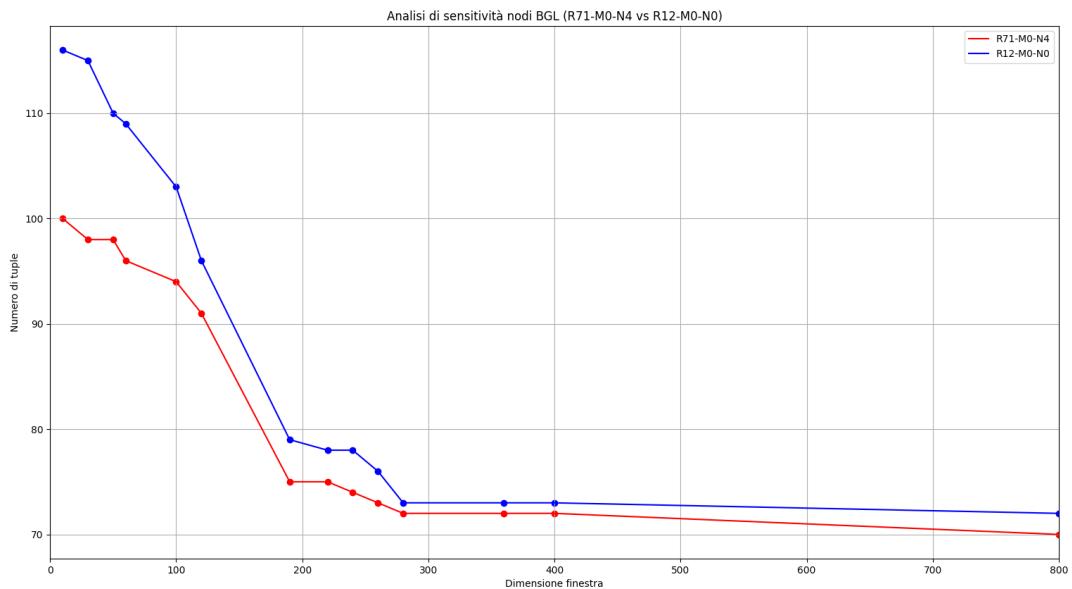


Figure 137: **Confronto finestre di coalescenza R71-M0-N4 e R12-M0-N0.**

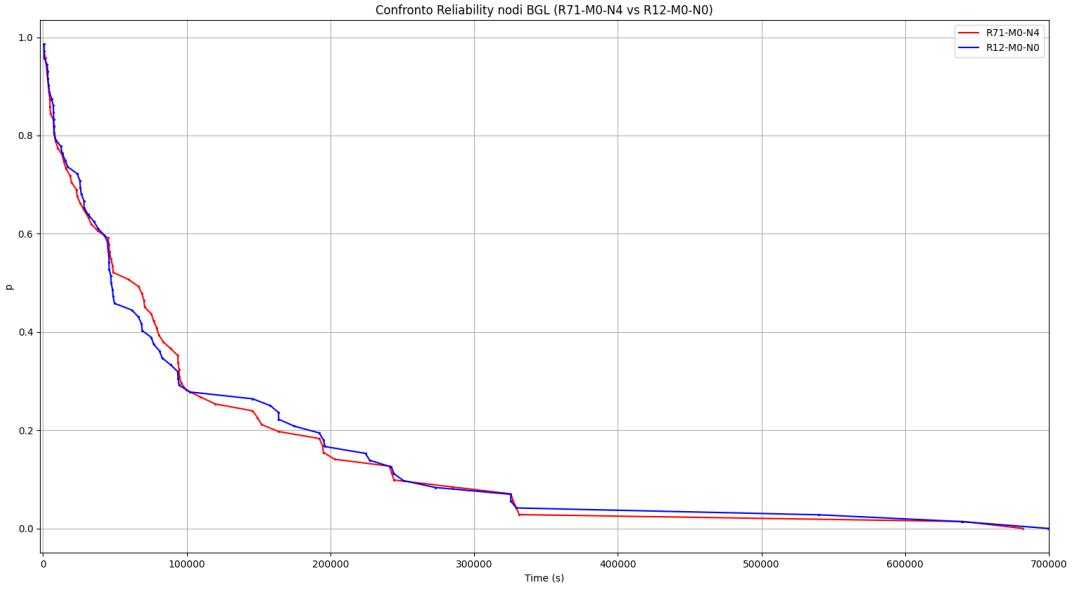


Figure 138: **Confronto Reliability R71-M0-N4 e R12-M0-N0.**

I due nodi hanno dimensioni della finestra di coalescenza comparabili, così come la Reliability. Dunque è possibile concludere che presentano parametri di affidabilità simili.

7.8 RELAZIONE TRA TIPI DI ERRORE E NODI

In questa sezione verranno individuate eventuali correlazioni tra le varie tipologie di errori ed i nodi del calcolatore Mercury. Filtrando i log a disposizione si ottengono dunque le varie tipologie di errore per ogni nodo (considerando solo i nodi più frequenti).

Nodo	DEV	MEM	I-O	NET	PRO	OTH
tg-c401	50782	11558	0	0	0	0
tg-master	0	0	452	3639	0	0
tg-c572	3176	845	0	0	0	0
tg-s044	0	0	3208	2	0	14
tg-c238	1071	0	230	3	0	0
tg-c242	918	149	0	0	0	0
tg-c648	643	0	0	0	616	0
tg-login3	0	0	381	1	0	0
tg-cc117	263	5	0	0	0	0
tg-c669	257	10	0	0	0	0

Table 25: **Relazione tra nodi e tipologie di errore.**

Tipologia Nodo	DEV	MEM	I-O	NET	PRO	OTH
Calcolo	80%	18.2%	0	0	1.8%	0
Master	0	0	12%	88%	0	0
Storage	0	0	99%	0.3%	0	0.7%
Login	0	0	99.7%	0.3%	0	0

Table 26: **Relazione tra nodi e tipologie di errore (%)**.

Dalle frequenze di tipologie di errore per ogni nodo è possibile osservare che per i nodi computazionali, le principali categorie di errore sono DEV e MEM, per il nodo master la principale categoria di errore è NET, mentre per i restanti nodi (nodi di storage e di login) la principale categoria di errore è I-O.