ISSN: 0974-5823 Vol. 6 No. 3 December, 2021

International Journal of Mechanical Engineering

An Analysis on Reliability and Availability in Cluster Architectured Software

[1]Dr. Akana Chandra Mouli Venkata Srinivas

[1]Professor, Department of CSE & Principal, BVC Engineering College (Autonomous), Odalarevu, EG Dt., AP., India.

[2]Dr. T.V. Janardhana Rao,

[2]Professor, Department of ECE, BVC Engineering College (Autonomous), Odalarevu, EG Dt., AP., India.

[3]Dr. Gunamani Jena

[3]Professor and Head, Department of CSE, BVC Engineering College (Autonomous), Odalarevu, EG Dt., AP., India.

[4]Dr. P.Devabalan

[4]Professor, Department of CSE, BVC Engineering College (Autonomous), Odalarevu, EG Dt., AP., India.

Abstract- The previous decade has seen huge research embraced in the field of elite performance computing leading to the emergence of different superior computing systems. The most generally perceived systems are symmetric multi and parallel processors (extraordinarily) dispersed systems, memory access (non-uniform and cash-coherent) and clustered architectures. Among these, Clustered architecture systems is utilized in an assortment of utilizations. It includes an assortment of at least two systems, or hubs, cooperating as a solitary unit. It is a high accessibility system which continually conveys applications, information, and assets on request. This design contains a lot of completely useful independent systems interconnected to shape a solitary element, consequently giving expanded performance and accessibility. Consistency and availability of the gadget in clusters is turning a fundamental subject with distant of groups in extraordinary enterprise apps. Consequently, this look at may be beneficial to system architects in making plans and scheming clustered architecture structures with multiplied tiers of reliability, consistency, obtainability and availability.

Index Terms- software reliability, availability, clustered architecture, failure rate, repair rate, reliability analysis.

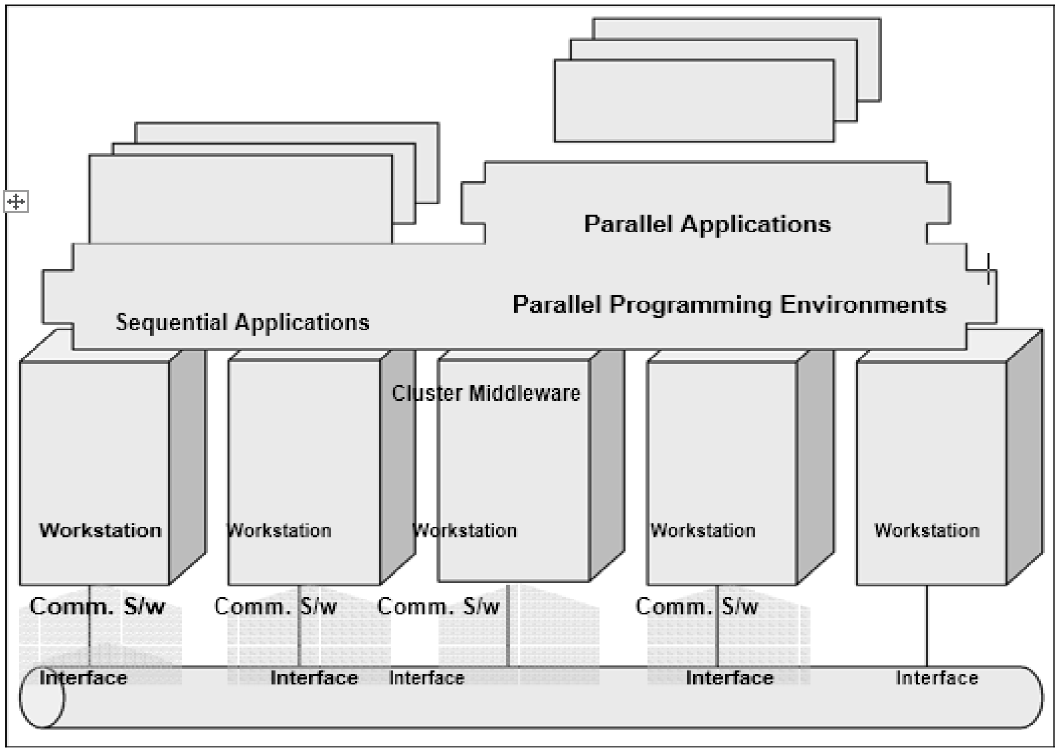
I. Introduction

The software program reliability and availability of clustered architecture systems has been tested through taking authentic fix and failure charges of different segments of the system into idea [1]. It is hard to correctly gauge a portion of the qualities inside the restricted time span through testing or may change on various client destinations because of working conditions, software utilized and repair rates [11]. The previous investigation didn't dissect the issue by thinking about vulnerability of the parameters. Eventually, in this paper, an exertion has been made to complete fuzzy consistency and accessibility evaluation to consolidate such doubtful variables. In the contemporary look at, clustered architecture systems mathematical model has been created utilizing Markov approach and its fuzzy consistency and accessibility are analyzed.

To begin with, the general structure for clustered gadget is clarified in brief. The device configuration, notations applied within the contemporary investigation and states and edges depiction are likewise delivered on this segment. Chapman-Kolomogrov mathematical analysis used to decide the consistency and accessibility of the device, is created. That is trailed with the aid of the conduct investigation of the gadget over diverse mixes of failure and repair prices of the sub-systems. The bushy reliability and availability are decided for different mixes of screw ups and fix prices for various α-reduce ranges. The doubtful variables are designed by fuzzy figures the usage of three-sided club characteristic. At long final, through examining the impact of software disasters and attach fees, on the reliability and availability of the machine, the affectability evaluation has been finished in each transient and constant country [2]. The conclusions depending on the modern-day study are at long last brought in paper.

II. General Architecture of Clustered Computing Systems

Fig. 1 outlines the primary parts of a Clustered architecture computing System: Numerous elite desktops, nation of the artwork OS, superior switches, community interface cards, a cluster middleware, ,speedy correspondence protocols and offerings, tools & applications and parallel programming environment. The cluster-nodes assume a significant function in the usage of a cluster architecture [14, 15]. The cluster hubs or nodes can work as a solitary integrated computing system, or work as individual PCs [12].



High Speed Network/ Switch

Fig. 1: Clustered planning system

The cluster intermediary state supports solitary system perplexion picture to the client. A cluster setup can also utilize n+k hubs or nodes (redundant), wherein n shows the amount of dynamic processing nodes and k demonstrates the amount of greater processing nodes inside the cluster. A.

A. Definition of States

State Zero: This state is far the acceptable nation and the state is always success state within the system. The device is viewed as working at complete restriction.

State One: it's far the faded state where errors has arisen in the system at state zero with unsuccessful rate λc and is known as error detection or identity and restoration state. In the event that the fault is recouped the machine returns to the initial state with correction rate μ_1 (anyways the system either unsuccessful & is going to fourth state with unsuccessful rate $\lambda_1(1-c_1)$ or the structure arrives any other restoration state, state 2 with unsuccessful rate $\lambda_1 c_1$.

State Two: It's miles the diminished state and an fault has arisen within the structure at state one with unsuccessful rate $\lambda_1 c_1$ and known as risky statistics recuperation state.

State Three: it is the condensed state wherein a error has passed off inside the structure at state two with unsuccessful rate $\lambda_2 c_2$ and is known as continual data recuperation state.

State Four: This state is unscusessful state.

B. Portrayal of an Edge Connecting Different States

In Table 1, the model (mendiratta [1]) portrayas pretty much all the edges interfacing unique states regarded in Fig. 1

Table 1. Source state, Destination state and Edge Weight

Source State	Destination State	Edge Weight
0	1	λ_c
0	4	$\lambda(1\text{-}c)$
1	0	μ_1
1	2	$\lambda_1 c_1$
1	4	$\lambda_1(1\text{-}c_1)$
2	0	μ_2
2	3	$\lambda_2 c_2$
2	4	$\lambda_1(1\text{-}c_2)$
3	0	μ_3
3	4	$\lambda 3$
4	0	M

III. Mathematical Modelling of the System

To accrue the consistency and accessibility of the device, system of linear differential equations (LDEs) along with Chapman-Kolomogrov equations is applied by means of utilising mnemonic rules and the system state transition diagram, shown in Fig. 1. The gadget whenever may be in desirable or decreased or unsuccessful state. State zero is the desirable state and successful state of any sort in the system. At the factor whilst a unsuccessful happens, the structure either goes to state three (reduced state or failure state). In diminished state, system presentation is corrupted and inside such a state the syatem quits working. States one, two and three are the decreased states and the fourth is the unsuccessful one [3].

A. Transient state for reliability analysis

The accompanying set of equations are determined utilizing mnemonic rule at time t, in view of the likelihood contemplations of various states in transient diagram [13] as represented in Fig. 1.

$$P_o(t)[1+\lambda c+\lambda(1-c)] = \sum_{i=1\ to\ 4}^{\cdot} [\mu_i . P_i(t)] \quad (1)$$

$$P_1^{'}(t)+\left(\mu_1+\lambda_1 c_1+\lambda_1(1-c_1)\right)P_1(t)=\lambda c\ P_0(t) \quad (2)$$

$$P_2^{'}(t)+(\mu_2+\lambda_1(1-c_2)+\lambda_2 c_2)P_2(t)=\lambda_1 c_1 P_1(t) \quad (3)$$

$$P_3^{'}(t)+(\lambda_3+\mu_3)P_3(t)=\lambda_2\, c_2\, P_2(t) \quad (4)$$

$$P_4^{'}(t)+\mu P(t)= (\lambda(1-c)P(t)+(\lambda(1-c))P(t)+(\lambda(1-c))P(t)+\lambda P(t) \quad (5)$$

With initial guess:

P_0 (0) =1 and P_j (0)=0 for (j ε 1,...,4) (6)

The underlying circumstance relies upon at the presumption that there's success in the structure closer to the start. By using the preliminary guess (6) and accepting a step size h=0.005 as 60 minutes, equations (1-5) are unraveled utilizing fourth order Runge-Kutta method. The system reliability is at last determined for different states of software failure and repair rates utilizing equation (7).

$$R(t) = P_0(t) + P_1(t) + P_2(t) + P_3(t) \quad (7)$$

From (Mendiratta, 1997 & 1998) software failures and repair rates records ongling data of the unique subsystems is taken and is considered in hourly units.

B. Steady State for availability analysis

A high availability system is consistently of most extreme significance for any association. So as to do this the consistent state likelihood of the system must be recognized. This can be gotten by forcing the $\frac{d}{dt} \to 0$, when t approaches to infinity. Using this guess, the LDEs, linear differential equations (1-5) can be changed as a set of linear equations as given underneath:

$$(\lambda(1-c)+\lambda c)P_0 = \mu_1 P_1+\mu_2 P_2+\mu_3 P_3+\mu P_4 \quad (8)$$

$$(\mu_1+\lambda_1 c_1+\lambda_1(1-c_1))P_1 = \lambda_c\ P_0 \quad (9)$$

$$(\mu_2+\lambda_1(1-c_2)+\lambda_2 c_2))P_2 = \lambda_1 c_1 P_1 \quad (10)$$

$$(\lambda_3+\mu_3)P_3 = \lambda_2\, c_2\, P_2 \quad (11)$$

$$\mu P = (\lambda(1-c)P+(\lambda(1-c))P+(\lambda(1-c))P+\lambda P \quad (12)$$

The set of linear equations (8 - 12) alongside the normalizing condition $\sum_i^p = 1$ is fathomed to locate the obscure perameter Pi (t) (

I = 0,1...,4). The availability of the system A(∞) would then be able to be assessed utilizing equation (13).

$$A(\infty) = P + P + P + P \quad (13)$$

IV. Numerical Results And Analysis

It is hard to gauge software failure and repair rate precisely inside the restricted time periods through testing and the numbers may fluctuate dependent on the working conditions at various client sites [4]. Those uncertain parameters were tested as fuzzy numbers using three-sided club feature:

$$f_{\triangle}(x\ \varepsilon\ a, b, c) = \begin{cases} \text{Zero} & x < a \text{ and } x > c. \\ (x-a)/(b-a) & x\ \varepsilon(a,b) \\ (c-x)/(c-b) & x\ \varepsilon(b,c) \end{cases}$$

The estimations of above uncertain variables are definite within the shape [L, N, H] wherein L is the maximum minimal viable estimation of the variable, the ostensible value (N) & the most increased achievable parameter (H). The different capability estimations of these variables are distinctive inside the table 2. Furthermore, table 3.

Table 2: Fuzzy Triangular Relationship Variables for different Software Failure times

Variables	λ_1 (hr)			λ_2 (hr)			λ_3 (hr)		
	L	M	H	L	M	H	L	M	H
Low	10	20	30	1500	1600	1700	80	90	100
Medium	20	30	40	1600	1700	1800	90	100	110
High	40	50	60	1700	1800	1900	100	110	120

Table 3: Fuzzy Triangular Relationship Variables for different Software Failure times

Variables	μ_1 (hr)			μ_2 (hr)			μ_3 (hr)		
	L	M	H	L	M	H	L	M	H
Low	15	30	45	1500	1600	1700	3300	3400	3500
Medium	30	45	60	1600	1700	1800	3400	3500	3600
High	45	60	75	1700	1800	1900	3500	3600	3700

In desk 2 & 3, for diverse combinations of failure and restore costs,of the consistency and accessibility of the system is calculated.

A. *Variation inside the consistency of the system with the various repair rates (μ_1).*

Impact of coorection rate μ_1 on the unwavering and Mean Time Between Failures of the system is concentrated by changing the estimation of μ1 from 15 to 75, with an addition of 15. The estimations of different parameters, for example, λ1 = 30, λ2 = 1800, λ3 = 100, c_1 = 0.9, μ2 = 1800, μ3 = 3600, λ = 0.0011, μ = 0.25, c = 0.9, c_2 = 0.9 have been used and are fixed values. The outcomes in this manner got have been presented in the Table 4.

Impact of restore rate μ1 on the unwavering reliability(consistency) and Mean Time Between Failures of the system (MTBF) of the gadget is focused by converting the estimation of μ_1 ($15 \leq \mu_1 \leq 75$), with an increment of 15. The estimations of different variables are used , for instance, fixed values, λ1 = 30, c =c_2= 0.9, λ2 = 1800, λ3 =100, μ2 = 1800, c_1 = 0.9,μ3 = 3600, λ = 0.0011, μ = 0.25. The effects on this manner were given have been provided within the table 4.

Table 4: System reliability variability with various restore rates (μ1)

t (in hrs)	μ_1=15	μ_1=30	μ_1=45	μ_1=60	μ_1=75
50	0.999958	0.999972	0.999974	0.999975	0.999976
100	0.999927	0.999949	0.999951	0.999951	0.999952
150	0.999903	0.999925	0.999927	0.999927	0.999928
200	0.999879	0.999901	0.999903	0.999904	0.999905
250	0.999855	0.999877	0.999879	0.999880	0.999881
300	0.999831	0.999853	0.999855	0.999856	0.999857
350	0.999807	0.999829	0.999831	0.999832	0.999833
400	0.999783	0.999805	0.999808	0.999809	0.99981
450	0.999760	0.999782	0.999784	0.999785	0.999886
500	0.999736	0.99958	0.99976	0.999761	0.999861
MTBF	**433.267**	**433.273**	**433.277**	**433.280**	**433.283**

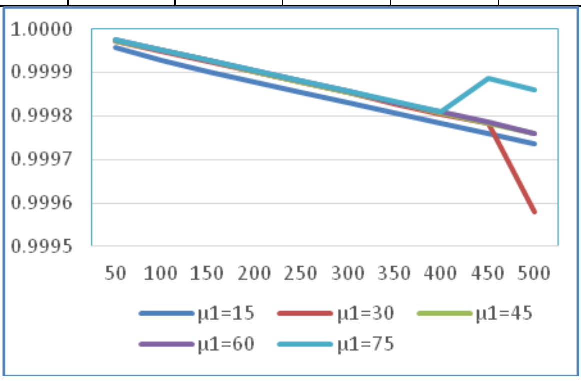


Fig. 2: System reliability variability with various repair rates (μ_1)

B. system consistency variation with the various unsuccessful rates (μ_1)

Effect of unsuccessful rate λ1 on the unwavering consistency simply as Mean Time Between Failures of the gadget is focused via differing the estimation of λ1 ($10 \leq \lambda_1 \leq 50$), with an augmentation of 10 and retaining the estimations of various variables constant: μ1 = 0.25,c_1 = 0.9, λ2 = 1800, μ = 0.25, λ = 0.0011, λ3 = 100, μ1 = 30, μ2 = 1800, μ3 = 3600, c =c_2 = 0.9. In table 5, the outcomes have been given.

Table 5: System reliability variability with the various failure rates (μ_1)

TIME (hrs)	λ_1=10	λ_1=20	λ_1=30	λ_1=40	λ_1 =50
50	0.999976	0.999975	0.999974	0.999974	0.999973
100	0.999952	0.999951	0.999951	0.999950	0.999949
150	0.999928	0.999927	0.999927	0.999926	0.999925
200	0.999905	0.999904	0.999903	0.999902	0.999902
250	0.999881	0.99988	0.999879	0.999878	0.999878
300	0.999857	0.999856	0.999855	0.999855	0.999854
350	0.999833	0.999832	0.999831	0.999831	0.999830
400	0.999809	0.999808	0.999808	0.999807	0.999806
450	0.999785	0.999784	0.999784	0.999783	0.999782
500	0.999762	0.999761	0.99976	0.999759	0.999758
MTBF	**433.2780**	**433.2776**	**433.2774**	**433.2770**	**433.2767**

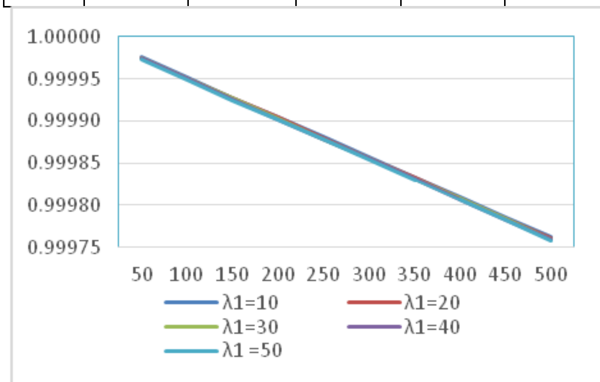


Fig. 3: system reliability variability with the varying failure rates (μ_1)

C. system reliabiloty variation with various errori dentification coverage rates, c

Impact of failure discovery coverage rate (c), on the unwavering reliability (consistency) and Mean Time Between Failures of the system is focused by using different estimations of c (c=0, c=9E-1, c=99E-2). The estimations of various every hour variables are taken as: μ = 0.25,λ1 = 30, λ2 = 1800, λ3 = 100, c_1=0.9, μ1 = 30, μ2 = 1800, μ3 = 3600, λ = 0.0011, c=0.9 and c_2=0.9. In table 6, the consequences finally received had been given.

Table 6: System reliability variation with changing c.

t (hrs)	c=0	C=9E-1	c=99E-2
50.0	0.999782	0.999958	0.999981
100.0	0.999567	0.999927	0.999957
150.0	0.999368	0.999903	0.999933
200.0	0.999178	0.999879	0.999809
250.0	0.998991	0.999855	0.999886
300.0	0.998824	0.999831	0.999862
350.0	0.998657	0.999807	0.999838

400.0	0.998498	0.999783	0.999814
450.0	0.998354	0.999760	0.999790
500.0	0.998211	0.999736	0.999780
MTBF	432.8830	433.2670	433.2730

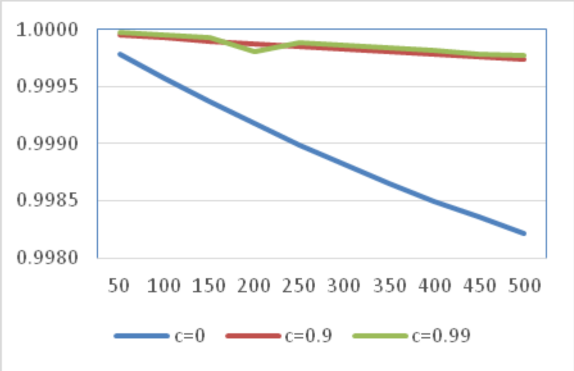


Fig. 4: Plot system reliability variation with changing values of c

D. System availability variation for Less values of unsuccessful rates and repair rates (Low (L), Medium(M) and High(H))

The obtainability of the structure has been decided for diverse capacity estimations of software program correction costs, μ_1 , μ_2 , μ_3 represented in the trio shape as: μ_1 = Low[15.0, 30.0, 45.0], Medium [30.0, 45.0, 60.0] and High [45.0, 60.0, 75.0], μ_2 = Low [1500.0, 1600.0, 1700.0], Medium [1600.0, 1700.0,1800.0] and high [1700.0, 1800.0, 1900.0], μ_3 = Low [3300.0, 3400.0, 3500.0], Medium [3400.0, 3500.0, 3600.0] and high [3500.0, 3600.0, 3700.0] and by taking constant estimations of software program repair fees: λ_1 =Low [10.0,20.0, 30.0], λ_2 =Low [1500.0, 1600.0, 1700.0], λ_3 =Low [80.0, 90.0, 100.0]. We have calculated the system accessibility for various α-cuts and the values are given in table 7.

L-L Combination 1: (Failure Rate) $_{\text{Low}}$ -(Repair Rate)$_{\text{Low}}$

L-M Combination 2: (Failure Rate) $_{\text{Low}}$ -(Repair Rate)$_{\text{Medium}}$

L-H Combination 3: (Failure Rate) $_{\text{Low}}$ -(Repair Rate)$_{\text{High}}$

Lower Bound	Low-Low Combination		Low-Medium Combination		Low-High Combination	
	Lower Limit	Upper Limit	Lower Limit	Upper Limit	Lower Limit	Upper Limit
0	0.999405	0.999439	0.999456	0.999479	0.999477	0.999505
0.1	0.999408	0.999437	0.999458	0.999478	0.999479	0.999503
0.2	0.999411	0.999435	0.999460	0.999476	0.999481	0.999501
0.3	0.999413	0.999433	0.999461	0.999475	0.999483	0 .999499
0.4	0.999415	0.999431	0.999462	0.999474	0.999484	0.999497
0.5	0.999416	0.999429	0.999463	0.999473	0.999485	0.999495
0.6	0.999418	0.999427	0.999464	0.999472	0.999486	0.999494
0.7	0.999419	0.999425	0.999465	0.999471	0.999487	0.999493
0.8	0.999420	0.999424	0.999466	0.999470	0.999488	0.999492
0.9	0.999421	0.999423	0.999467	0.999469	0.999489	0.999491
1.0	0.999422	0.999422	0.999468	0.999468	0.999490	0.999490

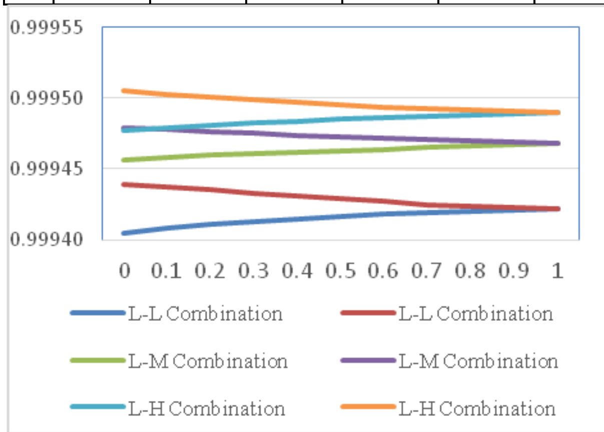


Fig. 5: System availability variation for Medium values of unsuccessful rates and repair rates (Low (L), Medium(M) and High(H))

E. System availability variation for Medium values unsuccessful rates and repair rates (Low (L), Medium(M) and High(H))

The readiness of the gadget has been determined for diverse α-cuts using the medium estimations of software programunsuucessful rate as: λ_1 = Medium[20.0, 30.0, 40.0], λ_2 = Medium[1600.0, 1700.0, 1800.0], λ_3 = Medium[90.0, 100.0, 110.0]. The estimations of various varibles are customary identical as in 4.4. The effects are shown in table 8.

M-L Combination 1: (Failure Rate) $_{\text{Medium}}$-(Repair Rate)$_{\text{Low}}$

M-M Combination 2: (Failure Rate) $_{\text{Medium}}$ -(Repair Rate)$_{\text{Medium}}$

M-H Combination 3: (Failure Rate) $_{\text{Medium}}$ -(Repair Rate)$_{\text{High.}}$

TABLE 8: System availability variation for Medium values of unsuccessful rates and repair rates (Low (L), Medium(M) and High(H))

α	Medium(M)-Low(L) Combination 1		Medium(M)-Medium (M)Combination 2		Medium(M)-High(H) Combination 3	
	Lower Limit	Upper Limit	Lower Limit	Lower Limit	Upper Limit	Lower Limit
0	0.999344	0.999396	0.999422	0.999491	0.999451	0.99952
0.1	0.99935	0.999395	0.999427	0.999486	0.999455	0.999515
0.2	0.999355	0.999394	0.999431	0.999481	0.999459	0.999511
0.3	0.99936	0.999392	0.999435	0.999477	0.999463	0.999507
0.4	0.999364	0.999391	0.999439	0.999473	0.999466	0.999503
0.5	0.999367	0.99939	0.999443	0.99947	0.999469	0.999499
0.6	0.999371	0.999388	0.999446	0.999467	0.999473	0.999495
0.7	0.999374	0.999387	0.999449	0.999464	0.999476	0.999492
0.8	0.999376	0.999385	0.999452	0.999462	0.999479	0.999489
0.9	0.999379	0.999383	0.999455	0.99946	0.999482	0.999487
1.0	0.999381	0.999381	0.999458	0.999458	0.999485	0.999485

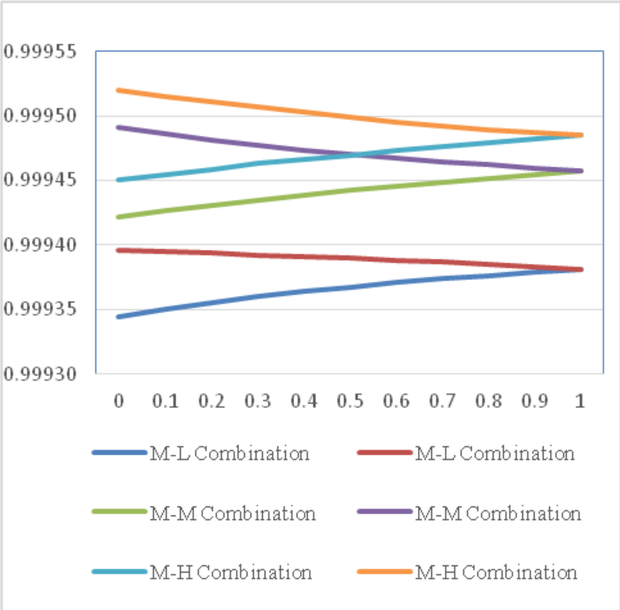


Fig. 6: System availability variation for Medium values of unsuccessful rates and repair rates (Low (L), Medium(M) and High(H))

F. System availability variation for High values of unsuccessful rates and repair rates (Low, Medium and High)

variety in the obtainability of the gadget via using excessive estimations of unsuccessful rates and occasional. In table 9, for numerous α-cuts , medium and excessive estimations of restore rates are shown. The estimations of software failure (unsuccessful) costs are taken as: λ_1 =maximum [40.0, 50.0, 60.0], λ_2 = maximum[1700.0, 1800.0, 1900.0], λ_3 = maximum [100.0, 110.0, 120.0] and the estimations of various variables are universal same as in 4.4.

TABLE 9: System availability variation for Medium values of unsuccessful rates and repair rates (Low (L), Medium(M) and High(H))

α	High(H)-Low(L) Combination 1		Hih(H)-Medium (M) Combination 2		High(H)-High (H)Combination 3	
	Lower Limit	Upper Limit	Lower Limit	Lower Limit	Upper Limit	Upper Bound
0	0.999301	0.999361	0.999367	0.999393	0.999409	0.999429
0.1	0.999306	0.999359	0.999369	0.999392	0.99941	0.999428
0.2	0.999311	0.999357	0.999371	0.999391	0.999411	0.999427
0.3	0.999316	0.999356	0.999373	0.99939	0.999412	0.999426
0.4	0.999320	0.999354	0.999374	0.999389	0.999413	0.999425
0.5	0.999324	0.999352	0.999376	0.999388	0.999414	0.999424
0.6	0.999328	0.999350	0.999378	0.999387	0.999415	0.999423
0.7	0.999331	0.999348	0.999379	0.999386	0.999416	0.999422
0.8	0.999335	0.999345	0.999381	0.999385	0.999417	0.999421
0.9	0.999338	0.999343	0.999382	0.999384	0.999418	0.99942
1.0	0.999340	0.999340	0.999383	0.999383	0.999419	0.999419

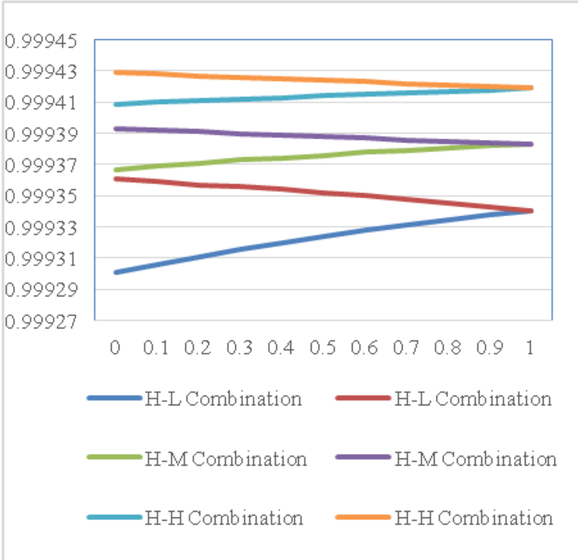


Fig. 7: System availability variation for High values of unsuccessful rates and repair rates (Low (L), Medium(M) and High(H))

V. RESULTS

From table 5. We will set off that with the expansion inside the restore rate from fifteen to seventy five, there may be an increment of 0.003% in MTBF and 0.0018% growth in the reliability of the system. with increment in time from 50 to 500 hours the reliability diminishes via roughly 0.02% [7].

Table No 6. Suggests that, a diffusion in failure rate from ten to fifty adversely influences the MTBF and reliability of the system. The MTBF diminishes by means of round 0.0003% and the reliability of the gadget additionally diminishes by using roughly 0.0003%. The reliability is moreover faded through more or less 0.02% with an expansion in time from fifty to five hundred hours [8].

Table 7. Suggests that an expansion within the estimation of c from zero to ninety nine improves the MTBF and reliability by more or less 0.09% and 0.05% separately. Even though, with a spread in time from fifty to five hundred hours, the reliability diminishes by way of round 0.1%.

Consistent with the outcomes from table 8., by using the expansion inside the product correction (repair) quotes from minimum to excessive features and by way of taking low estimations of constant software failure rates, the decrease limits of the accessibility of the system increments by more or less 0.0072% and higher limits of accessibility of the device increments by using around 0.00668% due to the expansion inside the product correction quotes from minimum to excessive features and by using minimum estimations of software unsuccessful rates, constant. Moreover, using the expansion in estimations of α (0≤ α ≤1), the decrease limits of availability of the gadget increments by means of around 0.0017% and the maximum limits of accessibility of the system diminishes via kind of 0.0017%.

From Table no 9 it is suggested that the minimum limits of accessibility of the structure increments is nearly 0.01% and maximu limits of accessibility of the structure increments is nearly 0.012% with the expansion within the product restore rates from minimum to excessive traits and with the aid of using medium estimations of software program fixed failure rates. Additionally due to the growth in estimations of α (0 ≤ α ≤1), the decrease limits of accessibility of the system increments by way of kind of 0.003% and the maximum limits of accessibility of the system diminishes nearly 0.0015%.

From table no 9, one can make some derivations. Due to the growth in the product restore rates from minimum to maximum traits, the lower limits of accessibility of the structure increments by using more or less 0.01% and the top limits (bounds) of accessibility of the system increments via around 0.0068% and via using high estimations of software program unsuccess rates, constant. Likewise due to the expansion in estimations of α (0≤ α ≤1), the decrease limits of accessibility of the device increments through kind of 0.0039% and the higher limits of availability of the device diminishes through round 0.002% [10].

VI. Conclusion

The conclusions can be produced using the research of above results that variety inside the traits the deficiency identification inclusion price impacts the reliability esteems to finest degree as evaluation with specific barriers. The estimations of deficiency area inclusion rate have to be saved as excessive by way of can be predicted beneath the situations. This can make the clustered architecture system highly consistent. Moreover, the derivations produced the usage of the Tables (8-10) may be beneficial to the system analysts in choosing an appropriate estimations of system failure and repair rates so that you can accomplish most extreme ranges of availability of the system.

References

[1] Michael R. Lyu, "s*oftware Reliability Engineering: A road map*", IEEE conference on .future of software engineering (FOSE-2007).

[2] M. Xie, "Software *Reliability Modeling",* World Scientific Publisher, Singapore (1991)

[3] J.T. Duane, "Learning curve approach to reliability Monitoring", *IEEE Trans. Aerospace,* AS-2, pp. 563-566

[4] J.D. Musa, and K. Okumoto, "A logarithmic Poisson execution time model for software reliability measurement", Proc. 7th Inf. Conf. on Software Engineering, 23@238 (1984)

[5] AL. Goel and K. Okumoto, "Time dependent error detection rate model for software reliability and other performance measures", *IEEE Trans Reliability,* R-28, pp. 206-21 1 (1979).

[6] M. Zhao and M. Xie, "Comparison of the log power NHPP model with some other two parametric Ones", *Technical Report, LiTH-IKP-R-6.53,* Linkoping, Sweden (1991)

[7] H. Ascher. and H. Feingold, *Repairable SystemsReliability: Modelling, Inference, Misconceptions and their Causes.* Marcel Dekker, New York (1983).

[8] M.R. Lyu and A. Nikora, A heuristic approach for software reliability prediction: the equally-weighted linear combination model. *Proc. Inf. Symp. On Software Reliability Engineering,* May 18-19, Austin, Texas, pp. 172-181 (1991).

[9] D.W. Noon, Practical software reliability. *IFAC Reliability of Instrumentation Systems,* The Hague, Netherlands, pp. 89-96 (1991)

[10] M. Zhao and M. Xie, Applications of the log power NHPP software reliability model. *Technical Report, LiTH-IKP-R-652,* Linkoping University, Sweden (1991)

[11] J. Lindén, A. Söderberg, U. Sellgren, "Reliability Assessment with Varying Operating Conditions", Volume 50, pp. 796-801, (2016), DOI: 10.1016/j.procir.2016.04.139.

[12] V. B. Mendiratta, "Reliability Analysis of Clustered Computing Systems", Proceedings Ninth International Symposium on Software Reliability Engineering, (4-7 Nov. 1998), Print ISSN: 1071-9458, DOI: 10.1109/ISSRE.1998.730890.

[13] N. Miskov-Zivanov, D. Marculescu, "Formal modeling and reasoning for reliability analysis", Design Automation Conference, (13-18 June 2010), pp. 531-536, DOI: 10.1145/1837274.1837406

[14] A. Fahim, "A Clustering Algorithm based on Local Density of Points", International Journal of Modern Education and Computer Science, 2017, Vol 12, pp. 9-16, DOI: 10.5815/ijmecs.2017.12.02.

[15] A. Kumar and S. Kumar, "Density Based Initialization Method for K-Means Clustering Algorithm", International Journal of Intelligent Systems and Applications, 2017, 10, pp. 40-48, DOI: 10.5815/ijisa.2017.10.05