Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007

Predictive E-Mail Server Performability Analysis Based on Fuzzy Arithmetic

Guillermo Navarro[1], Milos Manic[2]

Abstract –The performability of disk arrays systems has been studied before. However, in the case of imprecise data, a fuzzy model can be the base for the performability analysis. This paper presents a performability analysis of an MSExchange-like e-mail server. The analysis is based on a Markov Reward model. The performability analysis is accomplished through the use of fuzzy arithmetic. Unlike traditional Markov Chains, Fuzzy Markov Chains can successfully handle uncertain, imprecise probabilities. In cases where the failure rates, repair rates, or the workload parameters are uncertain, Markov Chains enhanced with fuzzy arithmetic provide means for comprehensive predictive performability analysis of a system. This performability analysis provides a valuable guideline regarding required resources such as the number of mailboxes, and therefore, the number of users the mail server can support with regards to the reliability and performance of the disk array used by the mail server. The fuzzy arithmetic helps in better visualization and estimation of the range of number of users the mail server is capable of servicing over long periods of time.

1. INTRODUCTION

Since the introduction of RAID systems in the Patterson's seminal paper in the late eighties [1], disk arrays have been an active area of research. The analysis of the reliability and the performance of the RAID systems and the different RAID levels have been studied since then [2-6,10,11].

The performability concept introduced in the late seventies [8,9] has been applied to different technologies including the disk arrays [10,11]. The disk arrays have been used as servers for databases since their inception. Although the primary purpose of disk arrays still remains the storage of data, specific types of data impose significantly different requirements on the set up and configuration of disk arrays. For example, disk arrays can used in support of video servers, web servers, or other type of data base servers. A study on performability analysis on usage such as disk array as a video server is described in [11].

The integrated modeling of the performance and reliability aspects of computer and communication systems has dramatically increased in the last decade. This approach has been referred to as the performability modeling and evaluation [18]. Typically, Reward Markov Models (RMM) have been used to analyze and evaluate the performability of various systems [18,19]. The reward function used in the RMM usually comes from performance analysis based on queueing theory [20]. Fuzzy arithmetic represents one of the essential sub domains of fuzzy logic [12]. The use of fuzzy logic for the analysis of reliability and performability has been done before [13,14,22]. The concept of crisp numbers and crisp vectors can be generalized to fuzzy numbers and fuzzy vectors, respectively. The fuzzy number can be realized in different forms, such as LR fuzzy numbers, discretized fuzzy numbers, or decomposed fuzzy numbers [12]

In this paper we present the performability analysis of a disk array used as a mail server. We base the analysis on some of the rules of thumb for the configuration of an MSExchange Server 2003 [15-17].

The authors do not claim to be doing an absolute performability study of a MSExchange e-mail server. Rather, based on selected number of MSExchange configuration recommendations, the authors demonstrate the following proof of concept: performability analysis enhanced by fuzzy arithmetic can be effectively used for predictive e-mail server performability analysis.

The organization of this paper is as follows: Section 2 lists the concepts used for the performability analysis and the modeling of the performability of a disk array. Section 3 shows the parameters used for an example of performability analysis of a mail server and explains the results obtained. Section 4 presents the conclusions.

2. PERFORMABILITY MODELING

2.1. Markov Model of a Disk Array

For testing purposes, a disk array with a total of N disks divided in groups of G disks is considered. The Markov Model (MM) used for the reliability analysis of this configuration is shown in Fig. 1.

This Markov Chain (MC) does not consider the failure of other components of a disk array, such as controller failures. RAID reliability studies with the consideration of failure of components besides disks can be found in literature [2].

When a single disk fails, the disk array goes to the non-optimal state S_1. This implicates the loss of the data redundancy. But the data is still not completely lost, since it is available on one of the $G-1$ disks that are still working in the group. The data lost on the failed disk must be then rebuilt from the redundant data. This is the feature that makes the disk array fault-tolerant. The repair rate μ is referred to as the *rebuild rate*. After the time $1/\mu$, the disk array completes the rebuild of the redundancy and the disk array goes back to state S_0 (back to the state with G working disks).

If during the time $1/\mu$ while the disk array is in state S_1 another disk within the disk group with the non-redundant data fails, the data is lost. In this case the disk array goes to

[1] Guillermo Navarro is with Hewlett-Packard.
Boise, ID 83714, USA. Email: guillermo.navarro@hp.com
[2] Dr. Milos Manic is with the University of Idaho at Idaho Falls.
Idaho Falls, 83402, USA. Email: misko@uidaho.edu

failure state S_2. If this event occurs, the user must restore the data using the backup on tape or some other media. The MC shown in Fig. 1 is for a disk group with G disks and one parity disk. That is why it has three states. For RAID levels with two parity disks, like R6 [3], the number of states would be four.

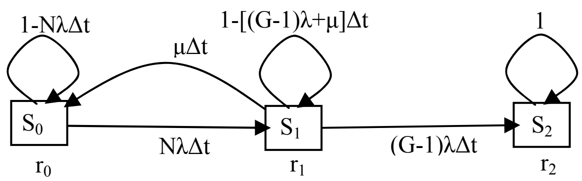

Fig. 1. Markov Chain (MC) of a Disk Array.

The derivation of this Markov Model comes from the traditional RAID models studied in literature. The reader is referred to the list of references for the review of the classical RAID Markov Models and equations used for reliability analysis of RAID systems. The model from Fig.1 is based on the Markov Model and MTTF equations derived in Shooman [7] and Patterson [1].

The system of differential equations used for the Markov Model of the reliability of a disk array as shown in Fig. 1 is described via probabilities of being in state S_0, S_1 and S_2:

$$\begin{aligned} \frac{dP_{S0}(t)}{dt} &= -N\lambda P_{S0}(t) + \mu P_{S1}(t) \\ \frac{dP_{S1}(t)}{dt} &= N\lambda P_{S0}(t) - [(G-1)\lambda + \mu]P_{S1}(t) \\ \frac{dP_{S2}(t)}{dt} &= (G-1)\lambda P_{S1}(t) \end{aligned} \quad (1)$$

The change made to the classical model is minor. We replaced the transition from S_0 to S_1 with $N\lambda$. This gives rise to the following system of equations using the Laplace transform:

$$\begin{aligned} P_{S0}(s) &= \frac{s + (G-1)\lambda + \mu}{s^2 + [(N+G-1)\lambda + \mu]s + N(G-1)\lambda^2} \\ P_{S1}(s) &= \frac{N\lambda}{s^2 + [(N+G-1)\lambda + \mu]s + N(G-1)\lambda^2} \\ P_{S2}(s) &= \frac{N(G-1)\lambda^2}{s\{s^2 + [(N+G-1)\lambda + \mu]s + N(G-1)\lambda^2\}} \end{aligned} \quad (2)$$

With the system of equations (2), we find that the reliability of the disk array represented by the Markov Model from Fig. 1 is as (3):

$$R(s) = \frac{s + (G-1)\lambda + \mu + N\lambda}{s^2 + [(N+G-1)\lambda + \mu]s + N(G-1)\lambda^2} \quad (3)$$

By applying (4) we get:

$$MTTF_{GROUP} = \lim_{s \to 0} R(s) \quad (4)$$

We derive an equation that we can use as the $MTTF_{RAID}$. The newly derived equation gives rise to a slightly different Markov Model for disk array reliability and performability estimation, compared to the one traditionally used (as illustrated by Fig. 1):

$$MTTF_{RAID} = \frac{(N+G-1)\lambda + \mu}{N(G-1)\lambda^2} \quad (5)$$

The newly derived equation (5) improves the traditional approach (6)

$$MTTF_{GROUP} = \frac{(2G-1)\lambda + \mu}{G(G-1)\lambda^2} \quad (6)$$

by introducing a term N that better describes a scenario with total of N disks divided in G groups.

The rebuild process is performed automatically. Certainly, the failed disk must be manually replaced at some point [1].

Equation (5) can be verified against the equation proposed by Chen in [3]. If we have a high lambda, like λ=500,000 and a disk array with N=200 disks using RAID1, so G=2, and with a rebuild time of 8 hours. We have:

$(N+G-1)\lambda$ = (200+2-1) * (1/500,000) = 0.000402

And μ = 1/8hrs. = 0.125.

It is easy to see that $(N+G-1)\lambda \ll \mu$ and we can again make the same consideration made in Shooman [7] and remove the $(N+G-1)\lambda$ term. This turns (5) in

$$MTTF_{RAID} = \frac{\mu}{N(G-1)\lambda^2} \quad (7)$$

Equation (7) is the classical $MTTF_{RAID}$ estimation proposed by Patterson and Chen in [3]. This is a verification of the MM proposed in this paper. We can use the Markov Model shown in Fig. 1 for the reliability estimation of the disk array.

If we consider a lower lambda, like λ=10,000 and again, a disk array with N=200 disks using RAID1, so G=2, and with a rebuild time of 8 hours. We have:

$(N+G-1)\lambda$ = (200+2-1) * (1/10,000) = 0.0201

And μ = 1/8hrs. = 0.125.

It is easy to see that in this case $(N+G-1)\lambda \ll \mu$ does not hold and we would have to use the (5) with all its terms for the estimation of $MTTF_{RAID}$. This is the same consideration as the MTTF equation obtained by Shooman in [7].

In order to estimate the system reliability we need to estimate the probability of the Markov Model being in state S_i at time t. This probability is designated as $P_{Si}(t)$ and can be estimated by means of the initial probability vector $PS(0)$ = [$P_{S0}(0), P_{S1}(0), \ldots, P_{Sm}(0)$] of the $(m+1)$ states and the *state transition probability matrix* (TPM) of the Markov Model of the disk array. The transition probabilities among states S_0, S_1 and S_2 are shown in Fig. 1 and can be translated into the TPM matrix (8):

$$TPM = \begin{array}{c} \\ S_0 \\ S_1 \\ S_2 \end{array} \begin{array}{c} \begin{array}{ccc} S_0 & S_1 & S_2 \end{array} \\ \begin{bmatrix} 1 - N\lambda\Delta t & N\lambda\Delta t & 0 \\ \mu\Delta t & 1 - [(G-1)\lambda + \mu]\Delta t & (G-1)\lambda\Delta t \\ 0 & 0 & 1 \end{bmatrix} \end{array} \quad (8)$$

The initial probability of S_0 is $P_{S0}(0)$=1 while the initial probabilities for S_1 and S_2 are $P_{S2}(0)$=0, and $P_{S3}(0)$=0.

Therefore, the initial probability vector is $PS(0)$=[1,0,0]. Failure rate (λ) and repair rate (μ) are assumed to be constant during the life of the mail server.

The estimation of probabilities of the states was done during discrete iterations of time. Thus, the time t at which the probabilities of all states (S_0, S_1, S_2) was evaluated was using a value n that ranged from 0 to certain maximum value, i.e. $n = (0,1,2,\ldots,n_{max})$. The time t was obtained by multiplying this value n by a time increment Δt (one hour delta for the example in this paper). In other words, we estimated the reliability of the system every hour from 0 through n_{max} hours. The criterion to choose the hour-based discretization steps is consistent with disk manufacturers that provide their failure rates in hours. The equation to compute the time at which the probabilities were estimated is:

$$t = n\Delta t \tag{9}$$

The probabilities of all states $PS(t) = [P(t)_{S0}, P(t)_{S1}, P(t)_{S2}]$ at some time $t=n\Delta t$ was estimated using:

$$PS(n\Delta t) = (TPM)^n PS(0) \tag{10}$$

Once the probabilities $PS(t)$ are calculated, the reliability of the RAID system can be obtained as:

$$R(n\Delta t) = P_{S0}(n\Delta t) + P_{S1}(n\Delta t) \tag{11}$$

2.2. Performability Model of Disk Arrays

The two performance measures used for the performability evaluation of the disk array were: 1) the throughput in IOPS (I/Os per second) and 2) the number of mailboxes the mail server can support based on the performance and reliability.

The throughput that a disk array can deliver depends on three factors: 1) the total number of IOPS that can be delivered by the disks installed in the disk array; 2) the RAID level used, and 3) the ratio of reads and writes.

In order to estimate the IOPS a disk array with N disks can yield, a model for the throughput of a single disk must be used first. The model used to estimate the throughput of a disk is based on [6] with some modifications. The average disk service time (τ_d) per I/O is estimated using the equation:

$$\tau_d = S_t + R_t + \frac{B_s}{\chi_b} \tag{12}$$

where S_t is the average seek time, R_t is the average rotational latency, B_s is the size of the transferred block of data, and χ_b is the bandwidth of the bus that connects the disk with the disk array controller. We are considering the same S_t for both reads and writes. Although in reality disks have different average seek times S_t for reads and writes, for the purposes of this analysis this simplification was made.

The inverse of the τ_d time gives us the throughput of one disk (χ_d) in IOPS:

$$\chi_d = \frac{1}{\tau_d} \tag{13}$$

This is another simplification, since the throughput of a disk also depends on the internal seek reordering algorithms [6]. The throughput of N disks is then:

$$\chi_d(N) = N\chi_d \tag{14}$$

The equations shown so far can be used to calculate the number of IOPS we can get from the disks in a disk array without considering the RAID level. The derivation of the equation to compute the disk array throughput is the next step. For this, the RAID level used to store the data must be considered. For this paper a RAID1 and a RADI5 disk array is assumed. If RAID1 is used the data must be mirrored and G=2. If RAID5 is used, then G=5 is used.

For RAID1 we have to consider that every data write is translated into two writes to different disks. Therefore, for RAID1 writes, the total number of IOPS that can be delivered by the disks must be divided by two. For the RAID1 reads it is only required to read the data from one disk. Thus, the number of IOPS that can be delivered by the disks is the number of IOPS for the reads. The ratio of reads R_p is also a factor that determines the disk array throughput (χ_{DA}) in IOPS. Thus, the equation to estimate the RAID1 disk array throughput is:

$$\chi_{DA}^{R1}(N) = R_p \chi_d(N) + (1 - R_p)\frac{\chi_d(N)}{2} \tag{15}$$

The reward r_0 of the optimal state S_0 for a RAID1 disk array is therefore:

$$r_0^{R1} = \chi_{DA}^{R1}(N) \tag{16}$$

For RAID5 we have to consider the kind of writes were used for the analysis. In our case we used the typical small 4KB accesses that an Exchange 2003 Server performs. The RAID5 level suffers from what is known as the "read-modify-writes" [1]. Every write is translated into two reads and two writes. Therefore, for RAID5 writes, the total number of IOPS that can be delivered by the disks must be divided by four. For the RAID5 reads it is only required to read the data from one disk. Thus, the number of IOPS that can be delivered by the disks is the number of IOPS for the reads. Again, the ratio of reads R_p is also a factor that determines the disk array throughput (χ_{DA}) in IOPS. Thus, the equation to estimate the RAID5 disk array throughput is:

$$\chi_{DA}^{R5}(N) = R_p \chi_d(N) + (1 - R_p)\frac{\chi_d(N)}{4} \tag{17}$$

The reward r_0 of the optimal state S_0 for a RAID5 disk array is therefore:

$$r_0^{R5} = \chi_{DA}^{R5}(N) \tag{18}$$

The reward r_1 for S_1, the non-optimal state, can be estimated by two factors: 1) One disk failed so we now have the throughput of $N-1$ disks. 2) The disk array is also copying the data that was stored on the failed disk on other disk besides servicing user requests. Besides estimating the throughput for the case of $N-1$ disks we need to add a factor that will drop the throughput a little more. We introduced a factor, R_f, with a value from [0,1]. This factor was the same for RAID1 and RAID5. This factor expressed the drop in throughput in a percentage form. For example, if the drop in performance caused by the rebuild is 5%, we assign Rf =

0.05. This is another simplification. If more accuracy is needed, all it has to be done is to introduce two factors, one for RAID1 and one for RAID5. So, the reward estimated for r_2 is:

$$r_2^{R1} = (1 - Rf)\chi_{DA}^{R1}(N-1) \quad (19)$$

$$r_2^{R5} = (1 - Rf)\chi_{DA}^{R5}(N-1) \quad (20)$$

Finally, of course, the reward for $r_3 = 0$, since the disk array is the failed state.

The transient performability (TP) is a concept defined in [18] as the expected reward r at time t. Equation (8) gives us the probability of each state and with that we can o estimate the performability of the disk array for every n_{th} iteration of Δt time by using:

$$TP^{R1}(n\Delta t) = P_{S1}(n\Delta t) r_1^{R1} + P_{S2}(n\Delta t) r_2^{R1} \quad (21)$$

$$TP^{R5}(n\Delta t) = P_{S1}(n\Delta t) r_1^{R5} + P_{S2}(n\Delta t) r_2^{R5} \quad (22)$$

We then used (19) and (20) to estimate the disk array performability in IOPS.

Now we need to come up with a way to estimate the performability of the mail server in number of users based on the performability in IOPS. We base the analysis on some of the recommendations for the configuration of an Exchange Server 2003 [16].

User Type	Database Volume IOPS	Send/Receive per day	Mailbox Size
Light	.5	20 sent/50 received	50 MB
Average	.75	30 sent/75 received	100 MB
Heavy	1.0	40 sent/100 received	200 MB
Large	1.5	60 sent/150 received	500 MB

Table 1. User profiles and corresponding usage patterns.

The formula to estimate the performability in mailboxes, i.e., users the mail server can support is based on three factors: 1) user profiles shown in Table 1; 2) the formula shown on the web page [16]; and 3) the fact that 90% of the IOPS are user interaction and the other 10% go to the logs maintained by the mail server. The formulas are:

$$PM^{R1}(n\Delta t) = \frac{0.9(TP^{R1})}{UT_{Type}} \quad (23)$$

$$PM^{R5}(n\Delta t) = \frac{0.9(TP^{R5})}{UT_{Type}} \quad (24)$$

where *UType = (Light,Average,Heavy,Large)*

3. EXPERIMENTAL RESULTS ON FUZZY PERFORMABILITY ANALYSIS OF THE MAIL SERVER

The intention of applying the fuzzy arithmetic to the performability analysis is to deal with uncertainty in a better way. For the purpose of this example the authors decided, based on [21], to use a λ=1/10000 hrs. Some of the parameters do not have a crisp value but a fuzzy value expressed in discretized form [12]. The discretized representation of fuzzy numbers used to deal with the Markov Chain model of performability can be expressed as fuzzy sets with five tuples $(x_i, \mu(x_i))$ where x_i is the value of the number and $\mu(x_i)$ is the corresponding membership value of x_i.

$$\tilde{P}^* = [(x_1,0),(x_2,0.5),(x_3,1),(x_4,0.5),(x_5,0)] \quad (25)$$

The fuzzy parameters for the this analysis are shown in a more concise form, where the $\mu(x_i)$ is omitted for brevity:

$$\tilde{p} = [x_1, x_2, x_3, x_4, x_5] \quad (26)$$

The parameters for this analysis were the following:

The life span of the mail server is 43,800 hours (5 years).

G for R1 = 2, Number of disks for a R1 group

G for R5 = 5, Number of disks for a R5 group

N = 200, Total number of disks

λ=[0.3x10^4, 0.5x10^4, 1x10^4, 2x10^4, 3x10^4] Failure rate

μ=[1/24, 1/16, 1/8, 1/4, 3/8] Repair Rate

R_p = [0.55, 0.6, 0.65, 0.7, 0.75], Percentage of Reads

R_t = [0.002,0.002,0.002,0.002,0.002] , Time for a rotation

S_t = [0.0038,0.0039,0.004,0.0041,0.0042], Time for a seek

B_s = [4096,4096,4096,4096,4096], Block size

χ_b = [2x10^8, 2x10^8, 2x10^8, 2x10^8, 2x10^8], Transfer rate

R_f = [0.03,0.04,0.05,0.06,0.07], Rebuild impact on reward

The resulting performability estimation is a fuzzy number with five values. For every iteration of the time t given by (9), a fuzzy performability result is generated. The solid line is the central value of the fuzzy result. The dotted lines are values in between the central and both boundaries. The low (high) dashed line is the lowest (highest) boundary of the fuzzy result.

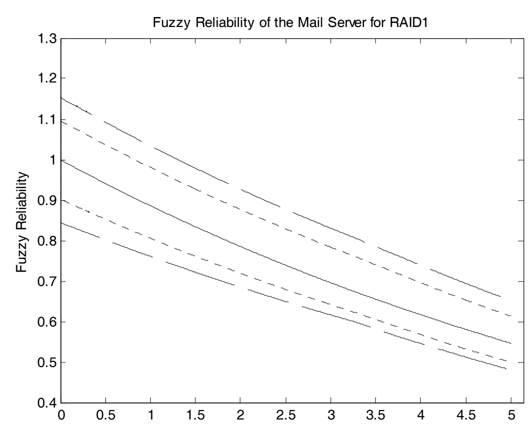

Fig. 2. Family of curves for fuzzy reliability RAID1. The low (high) dashed line is the lowest (highest) boundary of the fuzzy result.

Figure 2 shows the fuzzy RAID1 reliability of the mail server. It can be seen that there is a linear drop from 1 to 0.5

after 5 years of use. This is an indication that the mail server most likely will not have any problems at the beginning of its life. At the end of its life there should be some provisions in case of failure.

Figure 3 shows the fuzzy RAID5 reliability of the mail server. It can be seen that there is a linear drop from 1 to 0.1 after 5 years of use. This is an indication that the mail server most likely will fail as it gets closer to the end of its life. Here it is clear that provisions must be made to counter the fact that this server will most likely fail at the end of its life. Like for example, a backup server should be considered or budgeted within the next 5 years in case the "main" mail server fails.

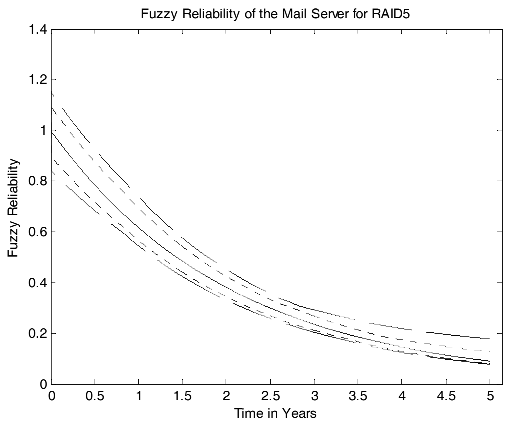

Fig. 3. Family of curves for fuzzy reliability RAID5. The low (high) dashed line is the lowest (highest) boundary of the fuzzy result.

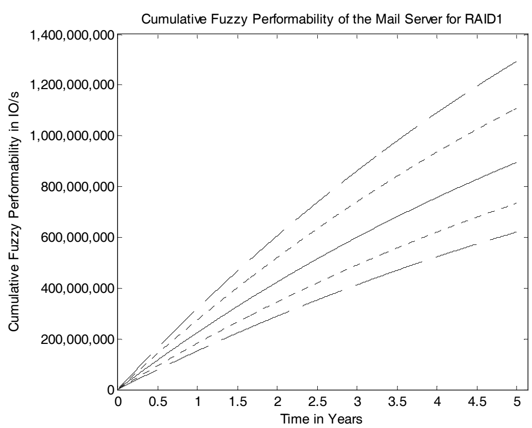

Fig. 4. Family of curves for fuzzy performability RAID1 in IOPS. Same interpretation of curves as Fig. 2 and 3.

Figure 4 shows the fuzzy RAID1 performability of the mail server. It can be seen that the IOPS range from around 40,000 to 20,000 at the beginning of the life of the mail server. The performability analysis tells us that after five years we can have throughputs in the order of 10,000 to 25,000 IOPS considering the reliability of the server. Depending on what level of service is expected in the next five years, plans should be made to either adjust the amount of service the mail server will provide.

Figure 5 shows the fuzzy RAID5 performability of the mail server. It can be seen that the IOPS range from around 35,000 to almost 15,000 at the beginning of the life of the mail server. The performability analysis tells us that after five years we can have no throughput. Here is very clear that if backup plans should be put in place to counter this future problem.

Figures 6 and 7 show the cumulative performability over the life of the mail server. This measure can serve to plan for the total amount of work the system can yield. In this case IOPS we can obtain from the mail server in its life span. We see the big difference between RAID1 and RAID5 in terms of the amount of work they can deliver due to the difference in reliability.

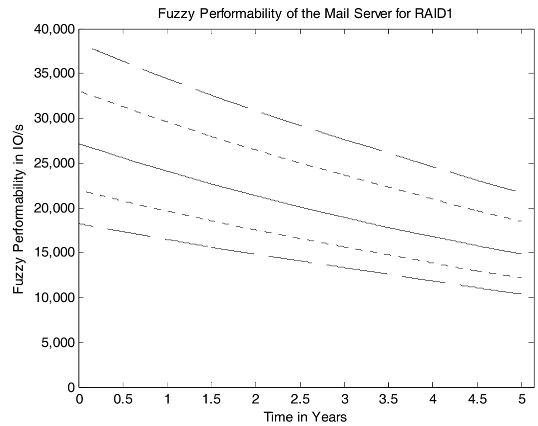

Fig. 5. Family of curves for fuzzy performability RAID5 in IOPS. Same interpretation of curves as Fig. 2 and 3.

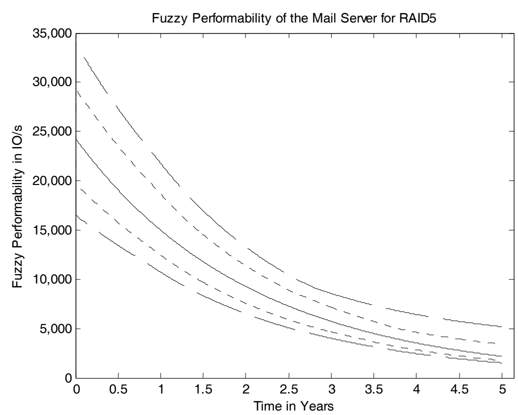

Fig. 6. Family of curves for fuzzy cumulative performability RAID1 in IOPS. The low (high) dashed line is the lowest (highest) boundary of the fuzzy result.

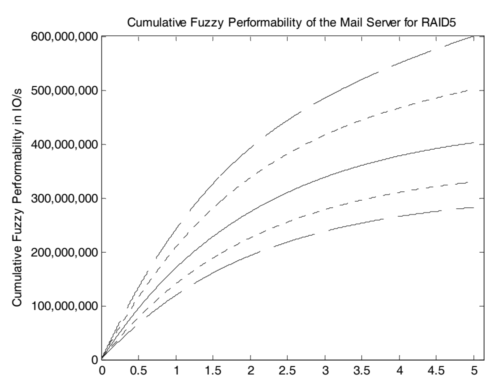

Fig. 7. Family of curves for fuzzy cumulative performability RAID5 in IOPS. The low (high) dashed line is the lowest (highest) boundary of the fuzzy result.

Figure 8 and 9 show the performability in number of users over the life of the RAID1 mail server. This measure can serve to plan for the amount of service the system can yield. As we can see, at the beginning of the life of the mail server it can serve up to 30,000 light users or around 23,000 of the heavy users.

If we want to keep this number of users constant we need to plan for the performability in the entire life of the product. In real life, figures 8 and 9 can be used to make the decision to use either RAID1 or RAID5 very easy based on the amount of service a business wants to provide.

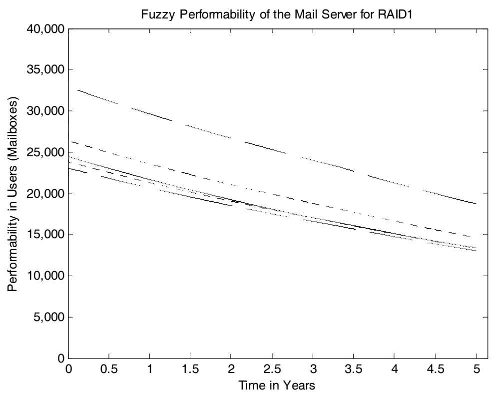

Fig.8. Family of curves for fuzzy performability in Users (mailboxes) R1. The low (high) dashed line is the lowest (highest) boundary of the fuzzy result.

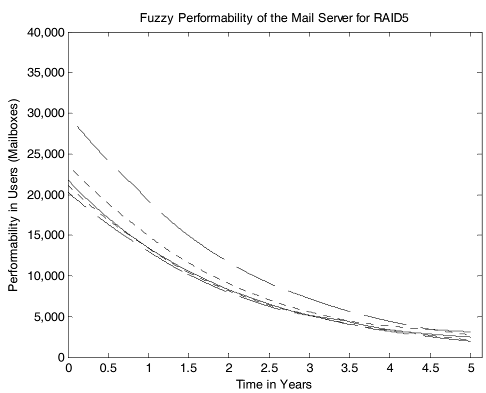

Fig. 9. Family of curves for fuzzy performability in Users (mailboxes) R5. The low (high) dashed line is the lowest (highest) boundary of the fuzzy result.

4. CONCLUSIONS

The performability analysis is a powerful tool in the analysis of the future capacity of service a computer system, and by extension, a business can provide. In this paper, the authors have shown how performability based on specific fuzzy arithmetic approach can be a tool for planning the future ahead so a business can keep the quality of service that may promise to customers. Not only that, but performability helps in the planning of the future needs based on the both probabilistic and possibilistic behavior of the systems.

Using the fuzzy arithmetic approach, all assets of presented model were taken as they were – uncertain. By employing fuzzy arithmetic, aggregated inherent uncertainties of such a RAID system were modeled in one run. Compared to traditional, numerical approaches, to obtain the family of curves such as presented in previous section, one would need to execute multiple runs of a model while choosing different possible model parameters each time. Savings in computational time, ability to provide an immediate response of the model, and elaborate depiction of multitude of performability curves make this approach suitable for near real time applications. Also, extreme system performability behaviors illustrated by boundary curves paint an immediate picture on what are the worst and best case scenarios under given system parameter uncertainties. The approach of performability modeling based on fuzzy arithmetic therefore provide a powerful tool for the effective design and business planning.

5. REFERENCES

[1] David A. Patterson, Garth Gibson, Randy H. Katz, "A case for redundant arrays of inexpensive disks (RAID)", ACM, 1988.

[2] Schulze, M.; Gibson, G.; Katz, R.; Patterson, D.A., "How reliable is a RAID?", COMPCON, 1989.

[3] D.A. Patterson, P.M. Chen, E.K. Lee, G.A. Gibson, R.H. Katz, "RAID: High-Performance, Reliable Secondary Storage", ACM Computing Surveys, 1994.

[4] Burkhard, W.A.; Menon, J, "Disk array storage system reliability", Fault-Tolerant Computing, 1993. FTCS-23.

[5] Ganger, G.R.; Worthington, B.L.; Hou, R.Y.; Patt, Y.N., "Disk arrays: high-performance, high-reliability storage subsystems", Computer, Volume 27, March 1994.

[6] Patterson David A., Hennessy John L., "Computer Architecture, a quantitative approach", Morgan Kaufman Publishers, 2003.

[7] Martin L. Shooman, "Reliability of Computer Systems and Networks", John Wiley & Sons, 2002.

[8] J. F. Meyer, "On Evaluating the Performability of Degradable Computing Systems", IEEE, 1978.

[9] M. Danielle Beaudry, "Performance Related Reliability Measures for Computing Systems", FTCS-7, 1977.

[10] Islam, S.M. Rezaul, "Performability Analysis of Disk Arrays", IEEE Circuits and Systems, 1993.

[11] S. A. Barnett, G. J. Anido, "Performability of disk-array-based video servers", Multimedia Systems, 1998.

[12] M. Hanss, "Applied Fuzzy Arithmetic",Springer-Verlag, 2005.

[13] M. Dumitrescu, T. Munteanu, A. P. Ulmeanu, "Fuzzy Logic in Power System Performability", ICIS, June 2004.

[14] P.S. Cugnasca, "Safety and reliability assessment using fuzzy theory applied to a subway system", Computer in Railways VIII, WIT Press, 2002.

[15] http://technet.microsoft.com/en-us/library/bb123872.aspx

[16] http://technet.microsoft.com/en-us/library/bb125019.aspx

[17] http://download.microsoft.com/download/e/b/a/eba1050f-a31d-436b-9281-92cdfeae4b45/subsys_perf.doc

[18] B. R. Haverkort, R. Marie, G. Rubino, K. Trivedi, "Performability Modeling: Technique and Tools", John Wiley & Sons, Ltd, 2001.

[19] K. S. Trivedi, S. P. Woolet, B. R. Haverkort, "Composite performance and dependability analysis", Elsevier, 1992.

[20] L. Kleinrock, "Queuing Systems", John Wiley & Sons, 1975.

[21] IDEMA, "Specifications of Hard Drive Reliability", R1-98.

[22] G. Navarro, M. Manic, "Fuzzy Performability Analysis of Disk Arrays", ISIE, 2006.