Markov Decision on Data Backup Scheduling for Big Data

K. Madasamy
Department of Computer Applications
Madurai Kamaraj University
Madurai, India

C. Elango
Research Center for Mathematical Sciences
Cardamom Planters' Association College, Bodinayakanur, India

M. Ramaswami
Department of Computer Applications
School of Information Technology
Madurai Kamaraj University
Madurai, India
mrswami123@gmail.com

Abstract- **Surfing over the data deluge is an unavoidable phenomena of the day, managing and protecting the massive amount of data is a vital task. Data backup operation is one of the most essential areas of Information Technology. In this article, we study a backup processing model using Markov Decision Process (MDP) for dealing with Big data. As backup storage operation of huge voluminous of data is a tedious task, we scheduled this process optimally using the versatile tool Markov Decision Process (MDP). A Numerical example is provided to illustrate the suggested data processing and backup model a viable one.**

KeyWords : Markov Decision Process (MDP), Big Data Analytics, Optimal scheduling, Data backup.

1. INTRODUCTION

In the present information technology scenario, the problems of data storage and retrievel processes are prevailing everywhere. They are not only very complex, risky and tedious task, but also pose heavy expenditure for the various organisations doing data storage processing. Data are just like in a crude form and it is imperative for organisations to refine them so as to make them valuable and they should be broken down and analysed for using them in profitable ways. Again one can take data as investment and protect them without any loss. Hence, Clive Humby a UK Mathematician rightly observed as "Data is the new oil!" (Michael Palmer 2006).

Primarily, four reasons are indentified for losing data and they are hardware failures, software bugs, human activities and natural disasters. As hardware devices are electronic devices which may prone to failure. One may make use of hard disk in computer system for storing voluminous data for a prolonged period of time. Backup processing is a shrewd method of protecting the huge data. Owing to several replicas of original data which leads to destroying the data wantonly, one should think about the ultimate cost of the restoration of the lost data from the replicas which are in our hand. Accordingly, backups are radically performed to serve three purposes : disaster recovery, operational backup, and archival (Gnanasundram,S. and Shrivastava 2009).Hence backup tasks are very essential.

To reap the real fruits of backup tasks they should be properly performed at the right instance and with the right hardware devices (Lars Wirzenius, Joanna Oja, Stephen Stafford and Alex Weeks). Full backup, Incremental backup, Differential backup, Mirror backup, Local backup, Offsite backup, Online backup, Remote backup, Cloud backup and FTP backup are the important methods of backup. Each method has its own merits and demerits which are beyond the scope of this paper. Rajeev Agrawal et.al., (Agrawal, R. & Nyamful, C 2016) recommended that full backup as a favourable choice over the other backup schemes. Full backup guarantees speedy recovery, even though it takes a significant amount of time to backup a large dataset. Applying full backup to large datasets may increase the rate of data block repetition. Data deduplication technology, significantly reduces the volume of stored data blocks for every single full backup, and allow users to backup, and recover data within a relatively short period of time. Ruofan Xia et.al., (Ruofan Xia, Fumio Machida, Kishor Trivedi 2014) have suggested the applications of Markov Decision Process (MDP) for data back up scheduling process. The framework modeled by them permits the translation of several data and system related requirements into an MDP instances and the solution to the instance provides the optimal schedule which ultimately reduces the system downtime. Manali Raje et.al., (Manali Raje, Debajyoti Mukhopadhyay 2014) presented a survey study on backup of data on remote Servers using Cloud. They suggested that large amounts of data is stored securely on the remote servers using the “Seed block” algorithm. Ruofan Xia et.al., (Ruofan Xia, Fumio Machida, Kishor Trivedi 2015) have further investigated a scalable optimization framework for storage backup operations using MDP. They suggested an approximation method to deal with the scalability issues.

Loss of data is very costly and it is a great menace for any business organizations. In particularly, its financial impact is very fatal. Therefore such situations shold be seriously taken in to account because finance is the life blood of business. Laszlo Toka et.al., (Laszlo Toka, Mathes,D. and Inchiarodi 2010) have conducted a study on Peer-Assisted approach for Online data backup process. They concluded that a peer-assisted approach does not discriminate peers in terms of performance, but associates a storage cost to peers contributing with little resources (Laszlo Toka, Mathes,D. and Inchiarodi 2010).

We organize the paper as follows: Section II gives an overall picture of Big Data Analytics. Section III deals with a brief outline about the Markov Decision Process. Section IV describes the proposed MDP model formulation. Section V covers an analysis part of the model and a methods for optimization. Section VI discusses a detailed algorithm of MDP. Section VII, provides a brief quantitative investigation to validates the feasibility of the proposed model. Section VIII outlined the conclusion and suggestion for future enhancement of the present research.

2. BIG DATA ANALYTICS

Big data is a catchword that defines data sets or combinations of data sets whose size (volume), complexity (variability), and rate of growth (velocity) make them difficult to be captured, managed, processed or analyzed by conventional technologies and tools, such as relational databases, desktop statistical tools and visualization packages. While the size used to determine whether a particular data set is considered as big data is not firmly defined and continues to change over time. Many analysts and practitioners currently refer to data sets from 30-50 Tera bytes to multiple Petabytes as big data.

3. MARKOV DECISION PROCESS

Markov Decision Process (MDP for short) is a popular mathematical framework for sequential decision making under uncertainty. In recent years, the field has seen explosive growth because of new application areas thrown up by advances in technology. These have not only stretched the limits of the existing theory, but have also brought about novel methodologies to handle problems that do not fit the existing theoretical constructs. An offspring of the operations research boom of the post-world war years, it quickly blossomed into a major sub discipline, not only of operations research, but also of control engineering and mathematical statistics. By the seventies, it already accounted for a vast number of articles, texts and surveys. But it did not get fossilized like some of its siblings from the boom years because of the continuous input of new problems thrown up by the emerging application areas. In recent years, such an impetus as come from the technological advances in communication networks and flexible manufacturing systems (Vivek S. Borkar, Mrinal, K. Ghosh 1995). Markov Decision Process is a special kind of sequential decision making nodes which are symbolically represented in Figure 1.

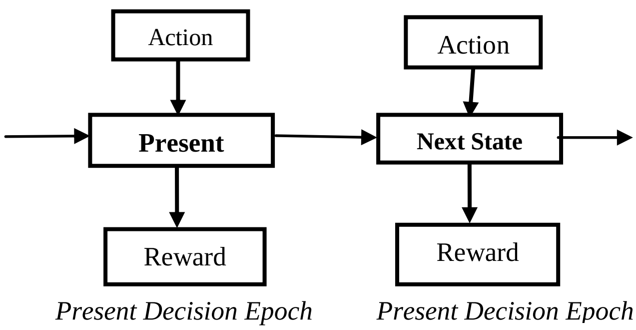

Figure 1. Markov Decision Process

At special kinds of time points called decision epoch the agent or a decision maker observes the state of a system. Based on the states, a specific action is selected from an action set with deterministic or probability rule. The decision maker received an immediate reward (or an immediate cost) and the system evolves to a new state according to the probability distributions and the actions choice.

a) Let T denote the set of decisions epochs (discrete or continuous intervals).
b) Let S denote the set of states occupied by the system (may be finite or countable or continuous).
c) Let us denote the action set with all possible action A_S corresponding to the state s ϵ S.
d) As a result of choosing actions aϵA in state s at decision epoch t, the decision maker receives a reward rt(s,a) and
e) The system state at the next decision epoch is determined by the transition probability distribution Pt(.|s,a). Here rt(s,a) is a real valued function defined for each sϵS and aϵA denote the value at time t of the reward received in period t.

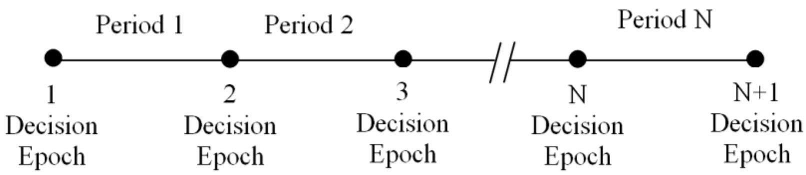

Figure.2 Decision epoch at different time slots

f) MDP (T,S, As Pt(.|s,a), rt(s,a)) is a process in which states moves in a path according to Markov rule to next state depends on the present state and the decision (action) implemented, not on the previous history of the states movement (evolution) over the period of time. Figure 2 shows the decision epoch at different point of time intervals. The theory of MDP reveals that it is sufficient to locate a stationary policy to achieve optimality (Puterman,M 2009). Thus the objective of solving MDP is to generate a mapping from the states to their actions. The applied solution for an MDP is obtained by maximizing or minimizing the total reward (gain or loss) due to the specific criteria (Puterman,M 2009). The expected total reward criterion is the most commonly applied criteria. In this study we used this criteria to solve the MDP problem.

Consider a typical system environment which is being dedicated for storing high volume of data, preferably in terms of Petabytes (~1015 bytes) measures. One of the major problem in big data analytics is its volumness and varities. This problem can be solved by classification and cleaning of data. After cleaning of data the volume of data will be reduced, only the significant and necessary data will be stored and finally data with noise will be removed. The three random quantities involved in this process are quantity of data entered in to the system, quantity of data stored after cleaning, quantity of data backed up. This concept can be represented in Figure 3.

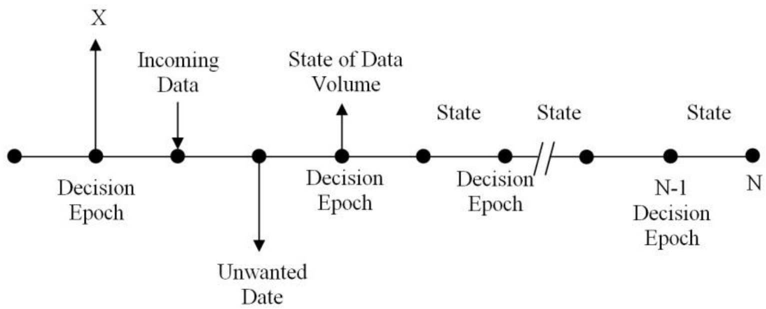

Figure.3 Incoming Data with Decision Epoch

Let X_t denote the quantity of data stored after backup process at the begining of period t, a_{t-1} being the amount of data received up to the end of t period, u_t , the quantity of data removed by cleaning process during the period t and X_{t+1} is the quantity of data stored after backup or non-backup decision is implemented. Figure 4 represents the backup activity along with their specific time interval..

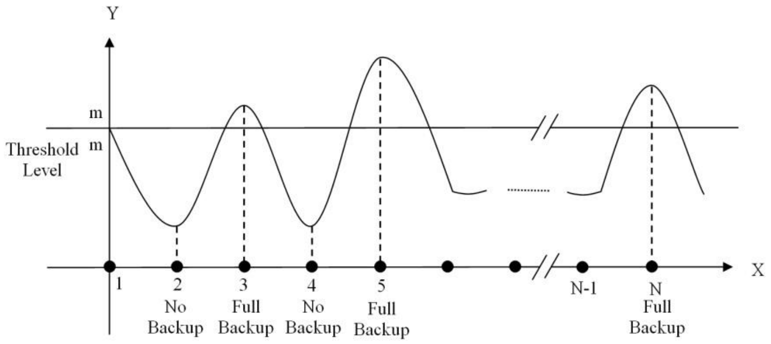

Figure.4 Back up at different state of time

The functional relation between the X_{t+1} and X_t of the system is given by equation (1) and is shown in figure 5.

$$X_{t+1} = X_t + a_t - u_t \tag{1}$$

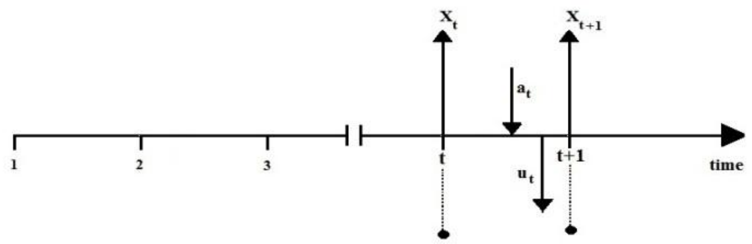

Figure 5. Periodic review of data storage

The decision policy is as follows: Whenever the quantity of data pool $X_t \geq m$, a full backup (entire data) is done, otherwise no backup will be done. $\{X_t: t \geq 0\}$ is a stochastic process with state space $E = \{0,1,2, \ldots\ldots N\}$, where N is a maximum storage capacity $N < \infty$. Hence the action space is A={0,1,2} , 0 – No backup, 1-full backup and 2 – compulsory backup.

4. MDP MODEL FORMULATION

We consider the problem on MDP having five components (5 tuples)

$(T, S, A, P_t(.|), r_t(.,.))$

Decision Epochs:

T={0,1,2,3…}

States : E – Number of units of data considered for backup.

$E = \{0,1,2, \ldots\ldots N\}$

Actions : Quantity of data (in discrete) units backed is the decision variable.

$A_0 = \{0\}, A_s = \{0,1,2\}$ for s=1,2,3,.....,N

$A = A_0 \cup A_s, s = 1,2,3, \ldots\ldots, N$

Cost : $c_t(s,a) = \begin{cases} \text{K+sb, where K is the setup cost and b is the backup cost per unit data} \\ 0 \quad \text{Otherwise} \end{cases}$

5. ANALYSIS

The one step cost are given by the equation (2).

$$c_t(s,a),\ s \in E \tag{2}$$

Let $\{X_t : t = 0,1,2....\}$ denote the state of the system at decision epochs t.

Assume the stationary policy R and hence the transition probability becomes as given in the equation (3)

$$\mathrm{P}_t\left(s^{'}.\,|\,\mathrm{s,a}\right) = P_r\left\{X_{t+1} = s^{'} \mid \mathrm{X}_t = s,a\right\},\ \ s,s^{'} \in E. \tag{3}$$

regardless of the past history of the system up to time epoch t.

Clearly $\{X_t : t = 0,1,2....\}$ is a Markov Chain (MC), with state space $E = \{0,1,2,.....,N\}$. The t-step transition probability of the MC under policy R is given by the equation (4).

$$P_t(s^{'}|s)\,(R) = P_r\{X_t = s^{'}|X_0 = s\}, s^{'}, s \in E, \mathrm{t} \in T \tag{4}$$

Let $V_t(s,R)$, denote the total expected cost over the first t decision epochs with states $s \in E$, when policy R is adopted. It is given by the equation (5)

$$V_t(s,R) = \sum_{k=0}^{t-1} \sum_{s^{'} \in E} P_{ss^{'}}{}^{(k)}\,(R).\,C_{s^{'}}\big(R_{s^{'}}\big), t, s \in E \tag{5}$$

where $C_s(R)$ setup cost + backup cost, K + b $\times$ s, where K-setup cost, b-the backup cost and s the present quantity of pool of data at time epoch t.

5.1. Cost Analysis

The average cost function $g_s(R)$ is given by$g_s(R) = lim \frac{1}{t} V_t(s,R)$, where $s \in E$.

Theorem 5.1

For all $s'.\,s \in E, \lim\limits_{t\to\infty} \frac{1}{t} \sum_{k=1}^{t-1} P_t \ \ (s'|s.\,R)$ always exists and for any $s' \in E$

$$\sum_{k=0}^{t} P^{(k)} \ \ (s^{'}|s) = \begin{cases} 1/\mu(s^{'}) \text{ if s}^{'} \text{ is recurrent} \\ 0 \qquad\quad \text{if s}^{'} \text{ is transient} \end{cases} \tag{6}$$

where $\mu_{s'}$, denote the mean recurrent time from state $s'$to itself; since the MC $\{X_t : t = 0,1,2....\}$ is a unichain which is inducible, all its states are ergodic and have an unique equillibrium distribution. Thus, by equation (7) it is given as,

$$\pi_{s^{'}}(R) = \lim_{t\to\infty} \frac{1}{t} \sum_{u=1}^{t} P^{(k)}(s^{'}|s)\,(R), s, s^{'} \in E \tag{7}$$

exists and is independent of initial state, such that $\pi P = \pi$ and $\sum_{s\in E} \pi(s) = 1$.

5.2 Optimal Policy – Algorithm

A stationary policy $R^*$is said to be an average cost optimal policy if $g_s(R^*) \leq g_s(R)$. For each stationary policy R uniformly in the initial state $s \in E$.

By constructing the new policy R^*such that for each $s \in E$, the equation (8) is framed as given below.

$$C_s({R_s}^*) - g(R^*) + \sum_{s^{'} \in E} P_{ss^{'}}({R_s}^*)\ \nu_{s^{'}} \leq \nu_s, \tag{8}$$

We obtain the improved policy R^*, such that $g(R^*) \leq g(R)$.

The optimal policy is obtained by minimizing the cost function (Michael Palmer 2006) by selecting actions $a \in A_s$, such that $a^* \in \underset{a \in A_s}{argmin}\{C_1(a) - g(R) - \sum_{s' \in E} P_t(s'|s,a) v_s(R)\}$

6. ALGORITHM

Step 0: (Initialization) Choose a stationary policy for the periodic review based data backup storage system.

Step 1: (Value determination step) Compute the unique solutions $(g(R), v_s(R))$ for the current policy , having the equations:

$$v_s = C_s(R_s) - g + \sum_{s^1 \in E} P_t(s'|s)\, v_{s'}\ s \in E \tag{9}$$

$v_i = 0$, where i is an arbitrarily chosen state in E.

Step 2: (Policy improvement) For each state $s \in E$, determine the actions yielding as in the equation 10,

$$\underset{a \in A_s}{argmin}\{C_s(a) - g + \sum_{s' \in E} P_t\,(s'|s,a) v_{s'}(a)\} \tag{10}$$

The new stationary policy R^* is obtained by choosing $R_s{}^* = a_s$.

Step 3: (Convergence Test) If the new policy $R^* = R$, the old one, then the process of searching stops with the policy R ; Otherwise go to step 1 with R replaced by R^*.

7. QUANTITATIVE INVESTIGATION

Consider a MDP formulation of a Dedicated Backup System capable of storing data in units of Tera bytes(TB) or Peta bytes(PB). The backup decisions are taken at time epochs equidistant from each other. Decisions for full backup of data units are taken. The policy adopted is m-policy: whenever the backed data level is ≥ m units, backup is done immediately. The transition probabilities $P(s'|s())$ are given in the table 1.

Table: 1 Transition Probability

S \ S’	0	1	2	3	4	5
0	0.1	0.6	0.2	0.1	0.0	0.0
1	0.0	0.1	0.6	0.25	0.05	0.0
2	0.0	0.0	0.15	0.55	0.25	0.05
3	0.0	0.0	0.0	0.25	0.65	0.1
4	0.0	0.0	0.0	0.0	0.1	0.9
5	0.0	0.0	0.0	0.0	0.0	1.0

Table 1 gives the estimated probability obtained from the real-time system data (*assuming Poission arrival distribution*). Arrival of units of data stream to the system follows a probability distribution $p\{\alpha_n = j\} = p_n$. By the natural generation of huge data, the arrival of data units to the systems leads to the transition probabilities given above. For the system, we assume the following parameters. N=5, the back-up costs at each level $s \in E$ are respectively. $c_0 = 0, c_1 = 2, c_2 = 2.5, c_3 = 3, c_4 = 3.5$ and $c_f = c_5 = 20$ and setup cost K=5.

7.1 Computational Procedure

The following computational procedure is implemented using the symbolic processing software Maple V Release 4.00a.

First, we initialize the policy $R^{(0)} = (0,0,0,0,0,2)$ prescribing a compulsory backup be done when s=5, with the following assumed cost structure as in the equation 11.

$c0$=0, $c1$=2, $c2$=2.5, $c3$=3, $c4$=3.5 and cf=$c5$=2 and setup cost K=5

$$T_s(a, R) = C_s(a) - g(R) + \sum_{s' \in S} P_t(s'|s,a) v_{s'}(a) \tag{11}$$

$$where\ T_s(a, R) = v_s(R) \text{ for a=R}_i.$$

For any given policy R, the policy improvement quantity is given by

Iteration 1:
For the current policy $R^{(1)} = (0,0,0,0,0,2)$the linear equation involving, $g(R)$ and $v_i$are

$$
\begin{aligned}
v_0 &= 0 - g + 0.1v_0 + 0.6v_1 + 0.2v_2 + 0.1v_3 \\
v_1 &= 0 - g + 0.1v_1 + 0.6v_2 + 0.25v_3 + 0.05v_4 \\
v_2 &= 0 - g + 0.15v_2 + 0.55v_3 + 0.25v_4 + 0.05v_5 \\
v_3 &= 0 - g + 0.25v_3 + 0.65v_4 + 0.1v_5 \\
v_4 &= 0 - g + 0.1v_4 + 0.9v_5 \\
v_5 &= 20 - g + v_5.
\end{aligned}
$$

By assuming $v_0 = 0$and solving the system of equation we get,

$$g = 20,\ v_0{}^{(R^{(1)})} = 0,\ v_1{}^{(R^{(1)})} = 15.25377229, v_2{}^{(R^{(1)})} = 28.59994747,\ v_3{}^{(R^{(1)})} = 51.27747132,\ v_4{}^{(R^{(1)})} = 74.98117503.$$

The test quantity $T_s(a, R())$has the comparable values.

$T_{10}(a, R^{(1)}) = 15.25377229$ $\quad$ $T_{11}(1, R^{(1)}) = 10.0$
$T_{20}(a, R^{(1)}) = 24.85088848$ $\quad$ $T_{21}(1, R^{(1)}) = 9.0$
$T_{30}(a, R^{(1)}) = 51.27747085$ $\quad$ $T_{31}(1, R^{(1)}) = 14$
$T_{40}(a, R^{(1)}) = 74.98117075$ $\quad$ $T_{41}(1, R^{(1)}) = 19.0$
$T_{00} = 0$ (default) and $T_{01} = 0$ (impossible)

The new optimal policy will be, $R^{(2)} = (0,1,1,1,1,2)$. (Table 2).
It is observed that the new policy $R^{(2)}$is different from$R^{(1)}$, then we proceed to the next iteration.
In Figure 6, the first row (series 1) contains cost corresponding to decision **0** (*i.e., "No Backup"*) and the second row (series 2) contains cost corresponding to decision **1** (*i.e., "Backup"*).

Iteration 2:
For the policy $R^{(2)}$, the next set of linear equations involving the variable $g(R^{(2)})$ and $v_s(R^{(2)})$are given by

$$
\begin{aligned}
v_0 &= 0 - g + 0.1v_0 + 0.6v_1 + 0.2v_2 + 0.1v_3 \\
v_1 &= 7 - g + 0.1v_1 + 0.6v_2 + 0.25v_3 + 0.05v_4 \\
v_2 &= 10 - g + 0.15v_2 + 0.55v_3 + 0.25v_4 + 0.05v_5 \\
v_3 &= 14 - g + 0.25v_3 + 0.65v_4 + 0.1v_5 \\
v_4 &= 19 - g + 0.1v_5 + 0.9v_5 \\
v_5 &= 20 - g + v_5
\end{aligned}
$$

Table: 2 Storage Levels and their corresponding Operating costs for iteration-1

Storage Level	Decision	Operating Cost	Optimum Value
0	0	0.0	0.0
	1	0.0	
1	0	15.253	10.0
	1	10.0	
2	0	24.85	9.0
	1	9.0	
3	0	51.277	14.0
	1	14.0	
4	0	74.981	19.0
	1	19.0	

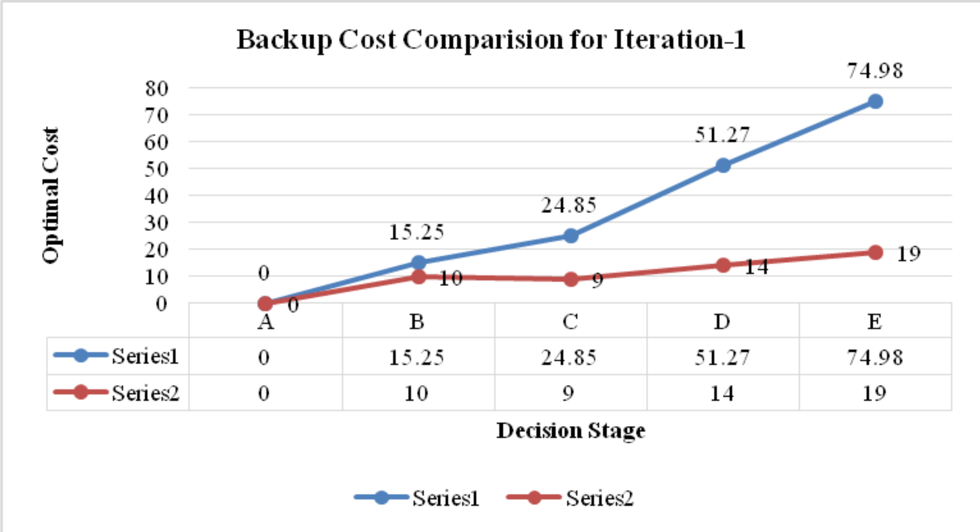

Figure.6 Optimal cost analysis for iteration -1

By assuming $v_0 = 0$ arbitrarily, and solving we get

$$g = 20, \quad v_0 = 0, \quad v_1 = 17.55281207, v_2 = 28.58500767,$$
$$v_3 = 37.51311224, \; v_4 = 45.36496409$$

The test quantity $T_s(a, R())$ has the comparable values.

$$T_{10}(a, R^{(2)}) = 10.55281208 \qquad T_{11}(1, R^{(2)}) = 7$$
$$T_{20}(a, R^{(2)}) = 14.06274509 \qquad T_{21}(1, R^{(2)}) = 10.0$$
$$T_{30}(a, R^{(2)}) = 23.54157992 \qquad T_{31}(1, R^{(2)}) = 14$$
$$T_{40}(a, R^{(2)}) = 26.62117321 \qquad T_{41}(1, R^{(2)}) = 19.0$$

The new optimal policy will be $R^{(3)} = (0,1,1,1,1,2)$. (Table 3)

In figure.7, the first row (series 1) contains cost corresponding to decision **0** (*i.e., "No Backup"*) and the second row (series 2) contains cost corresponding to decision **1** (*i.e., "Backup"*).

Since $R^{(2)} = R^{(3)}$, the policy sequence converges in the second iteration. This indicates that the optimal policy for backup operations in data storage problem is obtained. That is for each state of the system with positive quantity of data accumulation, it is wise and optimal to store (backup) the data at the beginning of the period.

Table: 3 Storage Levels and their corresponding Operating costs for iteration-2

Storage Level	Decision	Operating Cost	Optimum Value
0	0	0.0	**0.0**
	1	0.0	
1	0	10.552	**7.0**
	1	7.0	
2	0	14.062	**10.0**
	1	10.0	
3	0	23.541	**14.0**
	1	14.0	
4	0	26.621	**19.0**
	1	19.0	

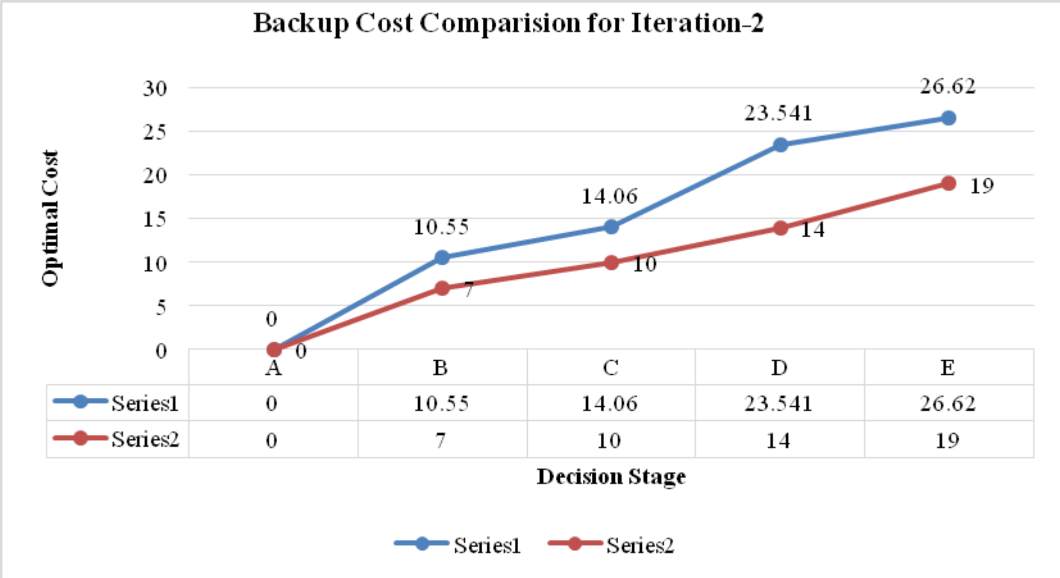

Figure.7 Optimal cost analysis for iteration -2

6. CONCLUSION

In this article we have presented the strategy for performing optimal data backup operation with relevance to big data application . A novel attempt has been made in this paper to backup the big data source using Markov Decision Process. As data backup process is a vital and critical task for any organization, it should be scheduled periodically and done in a proper way. This paper attempts to suggest a mathematical model that helps any user when and how much of data is subject to go for backup operations by utilizing optimal cost constraints in an elegant way. Numerical examples are provided to illustrate the concepts behind the proposed model. As an extension work, the same study could be implemented for the storage of real-time streaming data using Continuous Time Markov Chain model (CTMC) which is a semi- Markov decision Model.

REFERENCES

Gnanasundram,S. and Shrivastava (2009), Information storage and Management, Wiley Publishsing.

Laszlo Toka, Mathes,D. and Inchiarodi (2010), P. Online data backup A peer assisted approach, In Proc of 10th International conference on P2P computing, (1 – 10).

Puterman,M (2009). Markav Decision Process: discrete stochastic dynamic programming, volume 414, wiley.com.

Agrawal, R. & Nyamful, C (2016). Challenges of big data storage and management, Global Journal of Information Technology. 6(1), 01-10.

Ruofan Xia, Fumio Machida, Kishor Trivedi (2014), A Markov Decision Process Approach for Optimal Data Backup Scheduling, 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks.

Manali Raje, Debajyoti Mukhopadhyay (2014), A Survey on backup of data on Remote Server,International Journal of Science and Research, Volume 3 (12).

Vivek S. Borkar, Mrinal, K. Ghosh (1995), Recent trends in Markov decision Processes, J. Indian inst. Sci., 75, 5-24.

Ruofan Xia, Fumio Machida, Kishor Trivedi (2015), A Scalable Optimization Framework for Storage Backup Operations using Markov Decision Process, 21st Pacific International Symposium on Dependable Computing, IEEE.

Derman.C. (1970), Finite State Markovian Decision Processes, Academic Press, New York.

Michael Palmer (2006), Data is the New Oil, WWW. ana.blogs.com, Online article published.

Lars Wirzenius, Joanna Oja, Stephen Stafford and Alex Weeks, Linux System Administrators Guide, Chapter 12.

http:// https://www.maplesoft.com.