Computer and Information Science; Vol. 9, No. 2; 2016
ISSN 1913-8989 E-ISSN 1913-8997
Published by Canadian Center of Science and Education

A Cloud-Based Adaptive Disaster Recovery Optimization Model

Omar H. Alhazmi[1]

[1] Department of Computer Science, Taibah University, Medina, Saudi Arabia

Correspondence: Omar H Alhazmi, Department of Computer Science, Taibah University, Medina, Saudi Arabia. E-mail: ohhazmi@taibahu.edu.sa

Received: March 1, 2016 Accepted: March 13, 2016 Online Published: March 28, 2016

doi:10.5539/cis.v9n2p58 URL: http://dx.doi.org/10.5539/cis.v9n2p58

Abstract

Disaster recovery and business continuity plans are essential to make sure businesses keep on going. However, many small and medium businesses feel that these plans can cost them a lot. Moreover, the issues of cost and operation overhead prevent them from having solid disaster recovery plans. However, with the spread of cloud computing and pay-as-you-go and pay-for-what-you-use models, issues of operational overhead and expensive investment in extra storage and extra infrastructure are significantly minimized. On the other hand, as it becomes more affordable, businesses want to make sure they get the most optimal solution for minimum cost and overhead. In this work, we propose an adaptive cloud-based disaster recovery model that will be flexible in protecting data and applications with different plans by considering changes in risk levels and at the same time managing the costs and billing issues associated with the cloud. Therefore, this suggests an adaptive model to manage resources on the go while keeping costs as planned with the best possible protection.

Keywords: disaster recovery plan, disaster recovery as a service, business continuity

1. Introduction

1.1 Introduction to Disaster Recovery Optimization

Business Continuity Plans and Disaster Recovery Plans are becoming standard requirements as part of any organization's IT department, sometimes as a government regulation and sometimes as a standard, for example, ISO 22301 (ISO, 2011) for business continuity and ISO 24762 for disaster recovery (ISO & IEC, 2008). Hence, this is the case in big organizations because they realize the benefits; moreover, they have the capacity to dedicate some of their resources to this purpose. However, many medium and small businesses (SMBs) believe that cost is too high. A study by Semantic shows that 57% of small businesses and 47% of medium businesses do not have a disaster recovery plan (Semantic, 2011). Recently, a study by Cisco and Fortune showed that about 49% of small businesses owners do not have disaster recovery plan, fearing extra costs of remote sites and operational overhead (Cisco, 2015). Fortunately, advances in cloud computing and pay-as-you-go plans made startup costs significantly lower; thus, more medium and small organizations are able to afford disaster recovery plans than at any time before. However, SMBs need to plan their budgets to make sure that they spend funds in the best optimized way.

1.2 Importance of Disaster Recovery Optimization

We need to provide SMBs some adaptive model to cover their disaster recovery and business continuity and at the same time direct the resources in the right direction and stick with a pre-assigned limited budget.

In order to design a suitable disaster recovery plan (DRP) for a SMB, a business analyst must make a cost-benefit analysis of the technology used; for example, how much does one hour of downtime cost? How much does the loss of 1MB of data cost? How much can the business tolerate? If the business has more than one application or database, each might need separate analysis. Moreover, the indirect costs of downtime and loss of data should also be considered, such as loss of reputation, loyalty and customer confidence. Hence, the outcome will be setting the right recovery time objective (RTO) and recovery point objective (RPO). These can be set for the whole system or, even better, for each independent part of their system.

1.3 Related Work

When cloud computing became an important option for organizations, the cloud providers offered different forms, such as platform as a service (PaaS), infrastructure as a service (IaaS), or software as a service (SaaS).

However, not until later was disaster recovery as a service introduced and discussed by researchers such as (Wood et al. 2010), where they have shown how the cloud can be an excellent alternative for disaster recovery sites. Others (Alhazmi & Malaiya, 2013) have compared co-location disaster recovery and cloud-based disaster recovery. Now the term disaster recovery as a service (DRaaS) is becoming a common type of cloud service.

The issue of optimizing resources and trying to find the best allocation with some constraints is not a new topic. For example, (Buyya et al. 2001) has proposed an algorithm to allocate heterogeneous resources on the global grid. They have considered a time-budget constrained model. Also, recently, (Wientraub & Cohen, 2015) have proposed a combination of IaaS, PaaS and SaaS to optimize resource allocation. Moreover, (Manikandaprabhu1 & Senthil, 2013) have discussed some dynamic provisioning plans to reduce cost based on reserving resources by using on-demand vs. pre-reserved resources. They have shown some promising results. However, their models are based on multi-service providers and are for generic cloud resource allocation, while our work is specifically for disaster recovery purposes and our model does not take service providers into account.

(Banu & Saravanan, 2013) have proposed a two-phase algorithm to allocate resources at minimum cost and improved quality of service in the cloud. With similar objectives, (Poobalan & Selvie, 2013) have proposed better management of resources by utilizing an optimal cloud resource provisioning algorithm (OCRP). All have considered the pre-reserved and the on-demand options provided by infrastructure as a service (IaaS) cloud providers. The service providers offer the pre-reserved option, which costs less but needs to be reserved earlier. However, the on-demand option is more expensive but more convenient by not requiring prior reservation.

1.4 The Proposed Model

In this work, we present an adaptive cost optimizing algorithm based on a constraint of cost or budget and some other factors given by the user. However, the business is responsible for setting their RPO and RTO and also for identifying potential risks and changing levels of risk over time, which is sometimes predictable and can be set according to alerts and external threats, such as weather, or internal threats, such as scheduled maintenance. Here, the optimization model will allocate more resources and increase protection during these periods; this makes the optimization model flexible.

The goal of this model is to have a flexible adaptive optimization disaster recovery model that manages the resources according to some quality of service goals, financial limitations and internal and external threats, yet is applicable and practical and has very little management overhead. Moreover, the proposed model is vendor independent and can be applied to various cloud service providers, not limited to some billing and service plans. Therefore, in its current form it is generic and can be tailored to the specific needs of a SMB with some changes to accommodate the requirements of cloud service providers (CSPs).

In the following section which will give the main parameters, inputs, outputs of the model and the main algorithm. Next, in section 3, the optimization model is simulated using a scenario to illustrate how the model works, how it can be applied, and what will happen in a real operational environment; finally, in section 4 we will present a conclusion and remarks about the advantages and disadvantages of the model, identify strengths and limitations of the model, suggest possible improvements and briefly discuss some future research directions.

2. The Adaptive Cloud Based Disaster Recovery Model

Here, we preview the proposed model; in the next section we will preview the parameters used in the model; in section 2.2 we will preview the outline; and in section 2.3, the algorithm is previewed.

2.1 The Main Factors

Here we are going to overview the factors that impact the proposed system, which are: Criticality levels, Risk levels, Disaster Recovery tiers and Cost. Here we shall explain each factor.

The first factor is *criticality level,* which represents the level of importance of the data or an application. It can be assigned to a certain segment of data or an application. This is usually based on the business need and business analysis. The analysis should set the required Recovery Time Objective (RTO) and Recovery Point Objective (RPO). For simplification, we will assume three levels of criticality (high, medium and low); and segments with higher level of criticality have more priority when they are allocated in the cloud.

The second factor is *risk level*. A study by (Qurium, 2013) shows that although disaster recovery is often associated with natural disasters, nature is only responsible for about 5% (See Table 1), while the other 95% are mainly due to hardware and software failures and human errors. This can be helpful for system administrators to take care of their disaster recovery system during specific periods of time, particularly during software and hardware installations, upgrades and reconfiguration. With increased risks during these times, an optimized disaster recovery system can be elevated to allocate more resources during these times than during other times and thus utilize resources more efficiently.

Table 1. Ranking of Causes of Outages (Qurium, 2013)

Cause Rank	Cause	Percentage
1	Hardware failure	55%
2	Human error	22%
3	Software failure	18%
4	Natural	5%

Here, we will also assume that risk changes over time and we can assume three levels of risk (high, medium and low). The risk level of a disaster happening can be determined, although some natural disasters come without warning, such as earthquakes or fires. However, some of them come with a short notice, like floods and storms, which can be predicted by weather forecasts. Moreover, some disasters actually happen during prescheduled system upgrades and system maintenance. Those are ideal for this system because by raising the risk level during these activities, more resources are allocated to protect the data in a better way. Here, Figure 1 shows a hypothetical change of risk level over time:

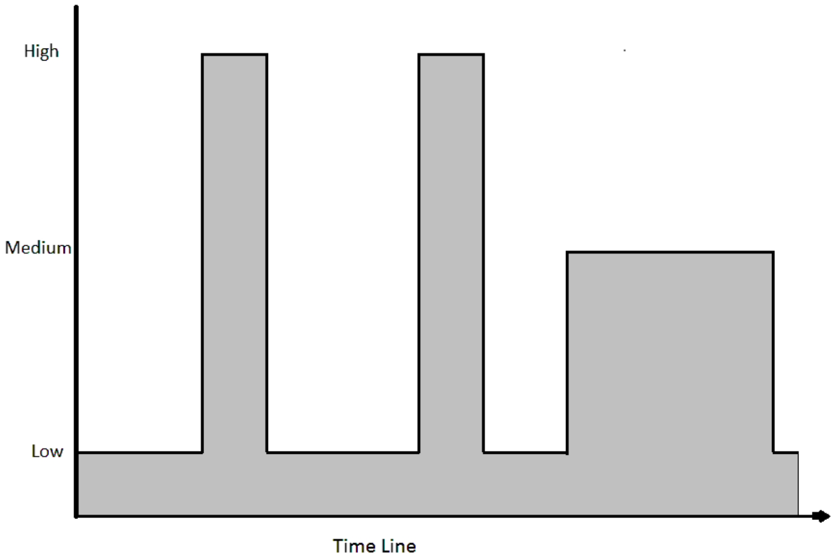

Figure 1. Risk level changes over a period of time

The third factor is the disaster recovery tier. There are several disaster recovery tier schemes previewed in (Alhazmi & Malaiya, 2012). However, in the cloud, we can prefer a scheme designed for the cloud; hence, we will choose Firdhous' (Firdous, 2014) classification of Disaster Recovery as a Service (DRaaS). Firdhous has proposed three levels of DRaas: CDDRaaS, WMDRaaS and HTDRaaS. However, Firdhous does not give full descriptions for these levels; thus, we are defining these levels using (Carroll, 2013).

Table 2. Firdhous classification (Firdhous, 2014)

DRaaS Level	Description	RTO	RPO
No DRaaS	No disaster recovery	-	
Cold site (CDDRaaS)	Only data is backed up; recovery systems need to be installed and configured and data needs to be loaded	Hours to days	Hours to days
Warm site (WMDRaaS)	Systems are preloaded and preinstalled preconfigured; however, data is not fully or partially	Minutes to hours	Minutes to hours

	ready		
Hot site (HTDRaaS)	Disaster recovery system is fully mirrored with the original system and fully synchronized and ready to take over operation	Less than 5 minutes	Less than 5 minutes

The fourth factor is *cost,* a main constraint here, because many small businesses don't want to spend more on extra storage and recovery systems. A study conducted by Forrester has shown that cost prevents many SMBs from owning a disaster recovery; for example, 49% worry about cost of a remote site, while 46% are concerned about cost of extra hardware in the recovery site. Moreover, 42% answered that the cost of implementation of a DR system is the main reason they don't have one (Cisco, 2015). Hence, the cost constraint is very important. Cloud technology and DRaaS could reduce the upfront cost and allow more companies and small businesses to own disaster recovery systems.

2.2 The Cloud-Based Adaptive Disaster Recovery Optimization Model (CBADROM)

The Cloud-Based Adaptive Disaster Recovery Optimization Model (CBADROM) assumes that data can be divided into independent segments so for each S_i ($S_1, S_2, S_3, ..., S_n$), there is a criticality level associated with it. Based on business analysis and business need, these levels can be in m levels ($C_1, C_2, ..., C_m$) (see Table 4) and there are also *k* risk levels for the whole system. Moreover, the cost is strongly linked to the disaster recovery tiers and the billing policy of the disaster recovery service provider. In other words, CBADROM takes four inputs: two direct inputs which are cost and current level of risk, and two inputs that need configuration, which are criticality/risk/disaster recovery tier matrix (see Table 3) and segment criticality map (see Table 4). Figure 4, illustrates the model's main outlines and how the components interact.

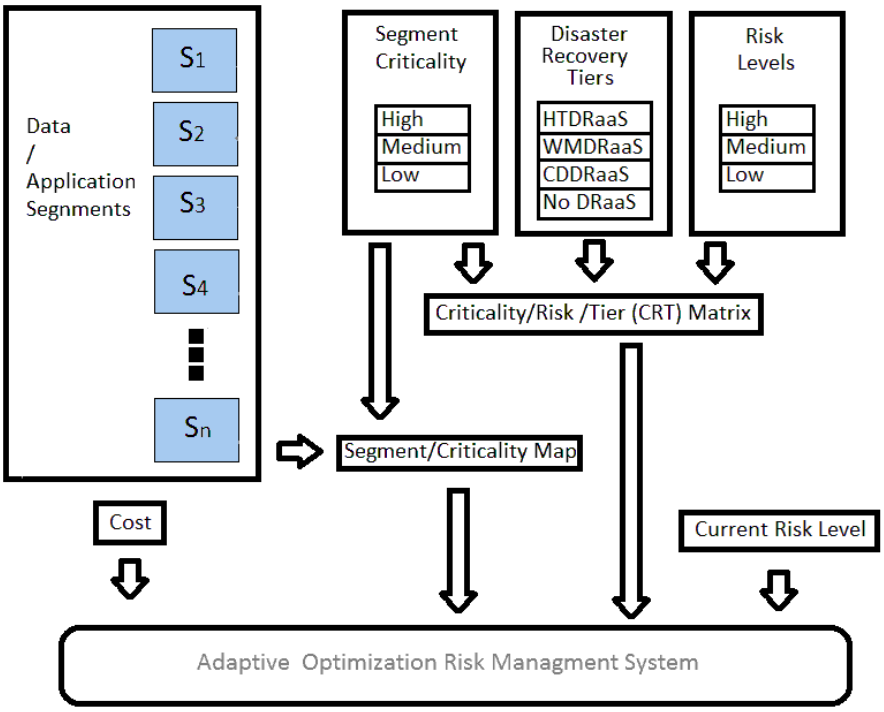

Figure 2. Outline of the Cloud-based Disaster Recovery Model

The CRT matrix (see Table 3) suggests multiple modes of operation and resource scheduling based on risk. We will assume the role of a business owner and build a criticality requirement table. This should be built by the business analyst to reflect the actual business needs of an organization. Table 3, below, shows an example of how a system analyst must build a criticality/risk tier for each criticality level and risk level, the desired recovery time objective (RTO) (how much time does the system need to recover) and a recovery point objective (RPO) (how much data loss can be tolerated). This can be achieved by analyzing loss impact on the business and must take into account direct, indirect, short term and long term impacts. After the CRT is built, segments must be given the appropriate criticality level (Table 4).

Table 3. Criticality/Risk Matrix/Tier matrix

Risk Level	Segment Criticality Level		
	Low	Medium	High
Low	CDDRaaS	CDDRaaS	WMDRaaS
Medium	CDDRaaS	WMDRaaS	HTDRaaS
High	WMDRaaS	HTDRaaS	HTDRaaS

2.3 The Cloud-Based Adaptive Disaster Recovery Optimization Model (CBADROM) Algorithm

The algorithm is given in Figure 3, below. The four inputs are shown in the top box. The main part is also shown on the bottom part. The system will call allocate_all() every period of time; here, the default is one day. It can be extended or shortened as desired.

```
Input Disaster_recovery_tiers RC [risk_levels] [criticality_levels],
Segment [num_segments], cost, risk; // the four factors
Output Allocated[num_segments] [tier]; // the allocation matrix
---------------------------------------------------------------
Main( )
Begin
        //each 24 hours the resources will be allocated based on current risk level
While (TRUE)
    Begin
    Risk ← Current_Risk_level; // current risk level for the day
    //defined by user or external factors
    Allocate_All (Risk, Cost, RC, Segment); // allocate the DR system on the cloud
    wait_for_one_day (); // wait for some time default:24 hours
    end while
end
```

Figure 3. The main algorithm

Figure 4 shows the algorithm for the functions. Cost() will estimate the cost of allocating a segment on a specific tier, based on size. Allocate_all() will go on all segments, starting with the higher criticality level, and allocate them based on this priority and then to the lower criticality level and so on. Of course, all of this is based on criticality-risk-tier matrix and segment-criticality map.

```
Into Cost(Segment S,Tier T)
Begin
Return Segment.size*T.cost_per_gigabyte;// return the actual cost
end
```

```
Int Allocate_All ()
Begin
for j← 1 to 3 // for criticality level 3 to 2 to 1
         for i ← 0 to num_segments      //for all segments 0 to i
                 Begin
                     if (Segment[i]. Criticality ==j)
                        result = Allocate (Segment[i], RC [Risk, Segment[i]. Criticality]);
                       if (result == 0) return 0; //if can not allocate , fail
                   end for
Return (number_of_segments);
// this will allocate segment i to the appropriate disaster recovery tier
End
--------------------------------------------------------------------------------
Allocate (segment S, Tier)
Begin
  If (Budget-Cost (S,Tier) < 0 ) return 0 //    if budget is too low return fail
  Else Budget ← Budget – Cost (S,T); //deduct the cost from budget
             return 1;
End
//allocated and cost deducted from Budget
```

Figure 4. Functions to be called by the main program

3. Applying the Cloud-based Disaster Recovery Optimzation Model

Here we examine a hypothetical scenario, as an assumption we will consider the criticality-risk-tier matrix given by Table 3. We shall see that if the company decides to spend only $100 for disaster recovery, the company has a set of systems and applications; their business analyst determined that their system can be divided into six segments as shown in Table 4. They assigned different a criticality level for each segment, and the size of each segment is also shown in the last column.

Table 4. A hypothetical system for simulation

Segment	Criticality	Size (Gb)
S_1	High	50
S_2	Low	80
S_3	Low	20
S_4	Medium	40
S_5	High	40
S_6	Medium	60
	Total	290

This company has obtained a list of prices from their cloud service provider (CSP), along with its RTO/RPO specifications, all illustrated in Table 5:

Table 5. Sample pricing for cloud-based disaster recovery system

DRaaS Level	Price	RTO	RPO
No DRaaS	0.00$	-	
Cold site (CDDRaaS)	0.03$ per day	Hours to days	Hours to days
Warm site (WMDRaaS)	0.05$ per day	Minutes to hours	Minutes to hours
Hot site (HTDRaaS)	0.09$ per day	Less than 5 minutes	Less than 5 minutes

From Tables 3, 4, and 5, we have produced Table 6, which shows the daily cost of the whole system for different risk levels.

Table 6. Applying the Criticality-Risk on the given example

		Low Risk		*Medium Risk*		*High Risk*	
Segments	*Size(Gb)*	*Cost (GB/day)*	*Segment cost($)*	*Cost (GB/day)*	*Segment cost($)*	*Cost (GB/day)*	*Segment cost($)*
S_1	50	0.09	4.5	0.09	4.5	0.09	4.5
S_2	80	0.03	2.4	0.03	2.4	0.05	4
S_3	20	0.03	0.6	0.03	0.6	0.05	1
S_4	40	0.03	1.2	0.05	2	0.09	3.6
S_5	40	0.09	3.6	0.09	3.6	0.09	3.6
S_6	60	0.03	1.8	0.05	3	0.09	5.4
Total disaster recovery cost per day			14.1		16.1		22.1

The company's analyst decides that for some reason risk will be medium on the third day and high on the fifth day and for all other days it is low.

Figures 5 and 6, show how the model allocates the appropriate disaster recovery tier to the segment based on the criticality/risk matrix given in Table 3. Here is the scenario:

- Days 1 and 2 are low risk days; therefore, S1 and S5 are allocated to HTDRaaS while S2, S3, S4 and S6 are allocated to CDDRaaS.
- In day 3, risk is elevated to medium. Based on the criticality/risk matrix in Table 4, only S4 and S6 need to be upgraded to WMDRaaS, while the rest of them stay unchanged.
- In day 4, risk is demoted to low, causing the downgrade of S4 and S6 back to CDDRaaS.
- In day 5, risk is elevated again but this time to high, causing an upgrade to S4 and S6 to HTDDRaaS and S2 and S3 to WMDRaaS.
- In day 6, risk is demoted to low, causing a downgrade of S2, S3, S4 and S6 to CDDRaaS.
- In day 7, the funds are about to finish (the balance is only about $5.40), causing a downgrade to most of the segments. Only S1 is allocated at the cost of $4.50; here, the remaining balance is too low, so the model had to skip S5, S4, S6 and then S2, to allocate only S3, as it is the first segment that is allocatable with the available funds.

Given this scenario in Figures 5 and 6, it is clear that the company needs to allocate more funds to disaster recovery to avoid the drop in the disaster recovery service that occurred on the seventh day. This occurred due to the lack of planning of the system but can be remedied by increasing the planning from one day to a longer period of time, thus avoiding the drop of DR for all segments.

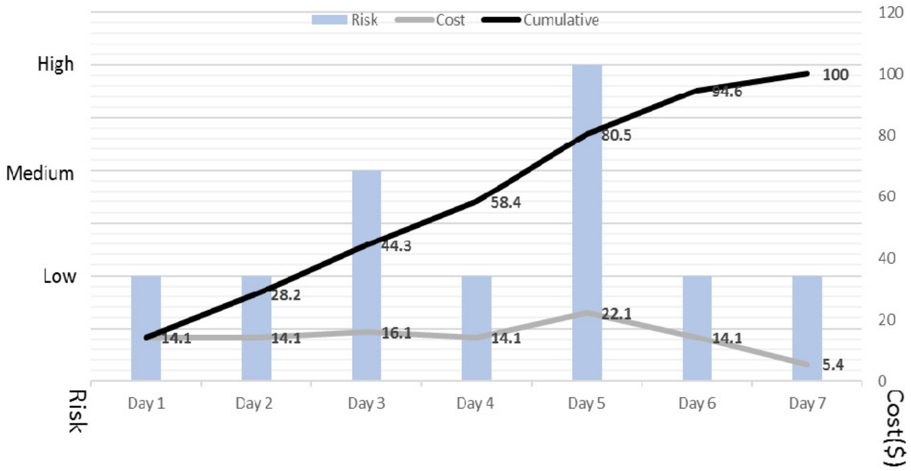

Figure 5. The bars represent the risk, the gray shows the daily cost, and the black shows the cumulative cost

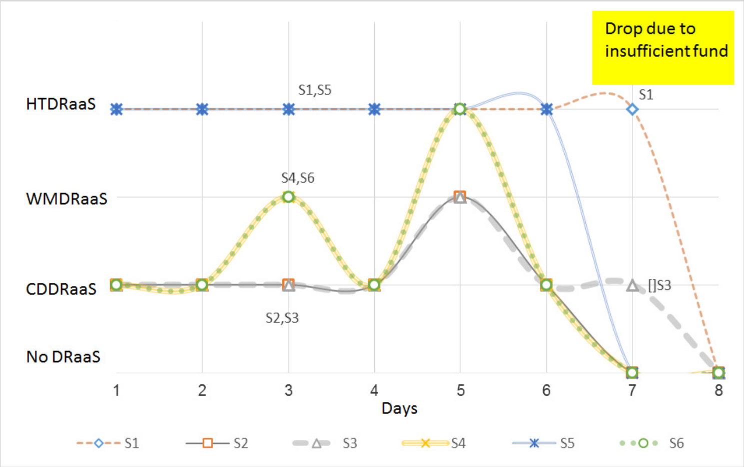

Figure 6. Segments allocation for the 8-days using 1-day planning

For the previous example, we have used day-by-day planning, and we have seen that at the seventh day most segments could not be allocated and on the eight day all of them – even critical segments – are left without coverage. Therefore, we shall try a four-day allocation system. We expect that this way, the results will change and the most critical segments will be favored over others. Let us repeat the scenario but with four-day planning (see Figure 7):

- The first four days will cost a total of $58.40 to allocate all segments. Thus, the remaining balance is $41.60.

- The estimated cost of the next four days is 22.10+14.10+14.10+14.01= $64.40, which is too high; therefore, the most critical parts will be allocated first.

- The first critical job S1 will be allocated for four days, costing 4.5*4=$18. The remaining balance is 41.60-18=$23.60.

- The next highly critical job S5 will be allocated, costing $14.40 for the four days; here the remaining balance will be 23.60-14.40=$9.40.

- Then, the system will look at the next job S4 which costs 1.20*3+2=$5.60 and will be allocated for four days. The remaining balance will drop to 9.40-5.60=$3.80.

- Then, the system will go to S6, only to discover that the needed amount is $8.40 while the balance is enough for two days at a low tier: 3.80-3.60=$0.20.

- Here, we notice that S2 and S3 are not allocated anymore for the second through fourth days because they have low criticality status, as illustrated by Figure 7.

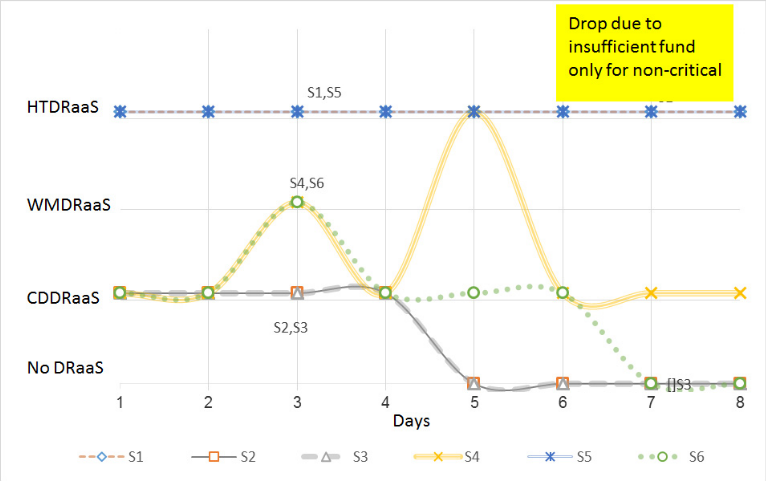

Figure 7. Segment Allocation using 4-day planning

4. Conclusions

Cloud computing has given the area of disaster recovery boost, as it has significantly reduced the upfront costs and thus increased interest in disaster recovery systems. Moreover, it has given disaster recovery systems more flexibility and made them dynamic and scalable. However, for small and medium businesses the issue of cost is still a concern, especially in times of slow economies. Here, we have presented a model to optimize resources based on available budget constraints. The model needs a business strategy as an input, which is the criticality-risk-tier matrix and segment-criticality map, cost and current risk, in order to manage resources in real time and give more priority to critical resources and change allocation based on changing risk levels.

We have demonstrated how the model can be applied on some hypothetical use case examples, and we have shown that the system prioritizes segment allocation based on criticality and that the system successfully re-allocates segments based on changes in risk levels. The success of the model is based on the correct business analysis and representing the analysis on the criticality-risk-tier matrix correctly, because this matrix is the main reference for the model and shows business priorities.

We have tested the model with two modes, a one-day planning mode and a four-day planning mode. We have found that longer term planning causes the model to enforce priorities more strictly and shorter term planning is quite the opposite.

Further work needs to be done; for example, the model can be implemented and tested in a simulation or experimental environments. Furthermore, longer term planning is a good area for experiments. Moreover, the model should be tested in a real environment and feedback should be collected and analyzed.

References

Alhazmi, O. H., & Malaiya, Y. K. (2013). *Evaluating Disaster Recovery Plans Using the Cloud*, Proc. Reliability and Maintainability Symposium (RAMS 2013), 37-42.

Alhazmi, O. H., & Malaiya, Y. K. (2012). *Assessing Disaster Recovery Alternatives: On-site, Colocation or Cloud*, Proc. 23rd IEEE Int. Symposium on Software Reliability Engineering Workshop, pp. 19-20.

Banu, U., & Saravanan, K. (2013). Optimizing the Cost for Resource Subscription Policy in IaaS Cloud. *International Journal of Engineering Trends and Technology, 6*(5). 296-301.

Buyya, R., Murshed, M., & Abramson, D. (2001). *A Deadline and Budget Constrained Cost Time Optimisation Algorithm for Scheduling Task Farming Applications on Global Grids*, in Int. Conf. on Parallel and Distributed Processing Techniques and Applications, pp.

Carroll, B. (2013). The Three Stages of Disaster Recovery Sites. Retrieved from http://www.seguetech.com/blog/2013/11/20/three-stages-disaster-recovery-sites

Cisco (2015). Are SMBs Taking Disaster Recovery Seriously Enough, white paper. Retrieved from http://www.cisco.com/c/dam/en_us/solutions/trends/open_network_environment/docs/draas-whitepaper.pdf

Firdhous, M. (2014). *A Comprehensive Taxonomy for the Infrastructure as a Service in Cloud Computing*, Fourth International Conference on Advances in Computing and Communications (ICACC), pp.158-161, 27-29 Aug. 2014.

ISO & IEC 24762 (2008). Guidelines for Information and Communications Technology Disaster Recovery Services. Retrieved from http://www.iso.org/iso/catalogue_detail?csnumber=41532

ISO 22301 (2012). International Standard Organization: Business Continuity. Retrieved from http://www.iso.org/iso/news.htm?Refid=Ref1602

Manikandaprabhu1, M., & Senthil, R. S. (2013). *Resource Provisioning Cost of Cloud Computing by Adaptive Reservation Techniques*, International Journal of Computer Trends and Technology, Volume 4, Number 5. Pp. 1118-1124.

Poobalan, A., & Slevi, V. (2013). Optimization of Cost in Cloud Computing Using OCRP Algorithm. *International Journal of Engineering Trends and Technology, 4*(5), 2105-2107.

Qurium (2013). Disaster Recovery Report, Q1-2013, White paper. Retrieved from http://oneclick.quorum.net/rs/position2quorum/images/Disaster-recovery-report-Q1-2013-1031.pdf

Symantic (2011). SMB Disaster Recovery Preparedness: Global Results. Retrieved from http://www.symantic.com/about/news/resources.jsp

Wientraub, E., & Cohen, Y. (2015). Cost Optimization of Cloud Computing Services in a Networked

Environment. *International Journal of Advanced Computer Science and Applications, 6*(4), 148-157.

Wood, T., Cecchet, E., Ramakrishnan, K. K., Shenoy, P., Van Der Merwe, J., & Venkataramani, A. (2010). *Disaster Recovery as a Cloud Service: Economic Benefits & Deployment Challenges*, 2nd USENIX Workshop on Hot Topics in Cloud Computing, p8.

Copyrights