Providing Statistical Reliability Guarantee for Cloud Clusters

Zhouhan Yang*, Anna Ye Du†, Sanjukta Das†, Ram Ramesh†, Thomas Furlani‡, Gregor von Laszewski§, Chunming Qiao*
*Department of Computer Science and Engineering, University at Buffalo (SUNY), Buffalo, NY, USA
†Management Science and Systems Department, University at Buffalo (SUNY), Buffalo, NY, USA
‡Center for Computational Research, University at Buffalo (SUNY), Buffalo, NY, USA
§Indiana University, Bloomington, IN, USA

***Abstract*—The ability to guarantee a high target reliability in a Service Level Agreement (SLA) for customers is critical for cloud providers to grow their business. Any SLA violation could cause revenues and reputation damage. In order to avoid such SLA violation on reliability, cloud providers typically allocate some backup Virtual Machines (VMs) beyond the number of VMs a customer requests. In the meanwhile, cloud providers also seek to minimize the operation costs, which are closely related to the number of backup VMs allocated to the customer. Thus an open question for cloud providers is how to choose the minimal number of backup VMs to guarantee the specified reliability statistically.**

In this paper, we investigate several failure scenarios in cloud data centers and build statistical models for them. Based on that, we take advantage of a tail bound theorem and propose some novel heuristic VM placement algorithms to resolve the above open problem in a nice way. We do extensive simulations to validate our results.

***Index Terms*—Data center, reliability;**

I. Introduction

As the cloud computing market continues to grow, the reliability of cloud services affected by various failures such as inaccessibility of cloud resources, becomes increasingly more critical. It is essential for cloud providers to make a guarantee for highly reliable services in the form of Service Level Agreement (SLA), which is challenging due to the complex nature of failures in a cloud cluster consisting of a number of servers, switches, etc.

In IaaS, a customer or application typically requires a certain number of Virtual Machines (VMs) to be up for a contract time. The reliability of an application is defined as the probability that the number of operational VMs remains above its requirement specified in the SLA over all possible failures [1]. Even though the cloud providers can provision new VMs when failures occur, the provision may have to go through complex scheduling system and result in long latency to meet the critical reliability requirements. Therefore, the common practice of cloud providers is pre-allocating some number of backup VMs distributed over different locations (e.g., different racks and data centers) for applications to tolerate various failures. Cloud providers want to provide a high reliability guarantee to make their customers happy; on the other hand, they try to minimize the number of backup VMs in order to reduce their operation cost which is dominated by the server costs. In reality, it is very challenging to figure out the minimal required number of backup VMs for an application and how to place them. Cloud providers need to carefully deal with the tradeoff between the increased costs due to over-provisioning backup VMs and the risk of reliability violation due to under-provisioning.

There have been some related work along this line. A large body of papers have studied the VM placement problem with multiple objectives such as energy saving [2] [4] [5], QoS [8] [12], and deal with the agile resource allocations issues as the resource requirements of applications dynamically change over time [7]. However, these studies haven't taken reliability requirements of applications into account, which may cause the applications or services to be abnormally terminated due to the server or switch failures. [1] and [6] considered pre-assigning backup VMs to applications to achieve a high reliability but they made the very limited assumption regarding the failures in a cloud cluster (e.g., they assumed the simplistic independent VM failure model.) Unfortunately, in practice, failures generally may spread from one VM to another.

In this paper, we build a new solution to better tackle VM provision and placement problems. Our work differs from all the previous work by considering concurrent server and ToR switch failures and providing the statistical models to compute the minimum number of backup VMs needed for each application given its reliability requirement. In summary,

- We build practical statistical models for various device failure scenarios;
- We adopt a nice tail bound theorem to infer the optimal number of backup VMs given a fixed reliability requirement. We use simulations to validate our results;
- We invent the heuristic VM placement algorithms to maximize the potential reliability strategically.

The rest of the paper is organized as follow. Section II formulates the reliability problem we target in this paper. In Section III, we start with a simplistic independent VM failure model where the number of minimal backup VMs can be computed directly. Then in Section IV, we move to the other two more realistic and complex failure models in which we no

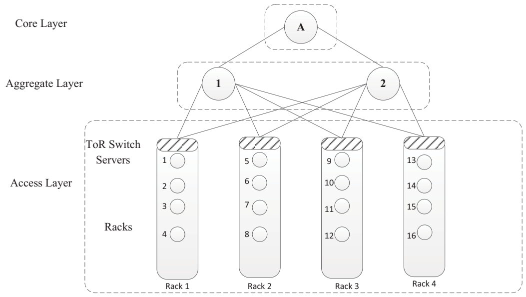


Fig. 1: Example of VM placement

longer have independent VM failures and there is no closed form solution to compute the tail probability of the failed VMs. Section V presents our simulation results. Conclusion and future work are presented in Section VI.

II. Problem Formulation

The functional layers of a cloud data center are typically built around three layers: the core layer, the aggregate layer and the access layer as in Fig. 1 [10]. The core layer is central to the data center network and provides interconnection between the aggregate layers. Typically, the core layer utilizes high performance low latency switches as Layer 3 devices. The aggregate layer acts as a services layer for the data center. Services such as load balancing, SSL optimization, firewalling, etc. are typically found at this layer. The access layer provides connectivity for the servers on different racks connected by ToR switches.

In this paper, we mainly focus on the failures in the access layer: physical server failures and ToR switch failures. The former causes all VMs on the failed server to fail while the latter makes all the VMs inside that rack to be inaccessible. We leave the failures of the core and aggregate layers and inter-cluster failures for the future work, which lead to more sophisticated failure models.

In the rest of this paper, we use n to indicate the number of VMs a customer requests and k to indicate the number of backup VMs the cloud provider offers during a given contract time, respectively. We define the reliability as follows:

Definition 1: Let V be a random variable indicating the number of failed VMs at some time instant during the contract time, where $0 \leq V \leq n+k$. The reliability in SLA denoted by α is defined to be $Pr[V \leq k] = 1 - Pr[V > k]$.

Now the problem we target is

Problem 1: minimize k subject to given n and α

Finding an optimal solution of this problem is a challenging open problem. It requires appropriate statistical modeling of server and switch failures, some strong mathematical tools on tail bound, and effective reliability-aware VM placement algorithms, etc. In the following sections, we are going to describe our approach to address the above optimization problem.

III. Independent VM Failure Model

In this section, we present the solution to Problem 1 in an unrealistic simplistic failure model. In this model, only physical servers could fail and all switches are guaranteed to work always. We assume that the failures of servers are independent and identically distributed without loss of generality [1]. We also assume that each physical server can only host at most one VM of a particular application. Under this constraint, a physical server failure will result in at most one VM failure for this application. It sheds light on more convoluted and realistic models we will investigate in Section IV.

For an application, let V denote the total number of failed VMs. Assuming the failure rate of a physical server is x. It is clear that a VM can either fail at the rate of x, or stay available at the rate of $1-x$, thus the VM failures follow binomial distribution.

According to Definition 1, we would like the probability that the number of failed VMs larger than k ($Pr[V > k]$)be smaller than $1-\alpha$. So the formulated problem as in Problem 1 becomes:

Problem 2:

$$minimize \quad k$$

subject to

$$Pr[V > k] \leq 1 - \alpha \tag{1}$$

As the VM failures follow binomial distribution, The probability that we have more than k VMs failures can be directly computed as:

$$Pr[V > k] = \sum_{i=k+1}^{n+k} C_{n+k}^{i} x^i (1-x)^{n+k-i} \tag{2}$$

We can simply iterate the possible values of k to find the minimum – we start k at some small initial value (e.g., 1) and compute the corresponding $Pr[V > k]$ using Eq. 2. We use eq. 1 to validate the value of k. If the constraint can not be met, we will increment k by 1 and iterate the above steps. We will stop once the eq. 1 is not violated. Some simulation results are shown in Section V.

IV. Correlated VM Failure Models

In this section, we are going to bring Problem 1 into some realistic failure models where we allow multiple VMs of an application allocated to a single physical server and consider the failure scenario of the ToR switches. Therefore, in these models, we no longer have independent VM failures and the number of failed VM does not follow binomial distribution any more. Thus there is no closed form solution to compute the tail probability.

In the rest of this section, we are going to present a novel approach to tackle the problem. This approach is based on the previous simplistic independent failure model and an advanced tail bound theorem which is not well known but provides

[1]It is not hard to extend our approach to handle the non-identical distributions.

a very tight tail bound for the sum of independent random variables. We will first introduce this tail bound theorem and then demonstrate how to take advantage of it to address Problem 1 in two correlated failure models with our well-designed VM placement algorithms.

A. A Tail Bound Theorem

We ground our study on a tail bound theorem originally presented in [13] which is an enhanced version of Theorem A.1.19 in [9].

Theorem 1: Given any $\theta > 0$ and $\epsilon > 0$, the following holds: Let W_j, $1 \leq j \leq m$, for some arbitrary m, be independent random variables with $E[W_j] = 0$, $|W_j| \leq \theta$ and $Var[W_j] = \sigma_j^2$. Let $W = \sum_{j=1}^{m} W_j$ and $\sigma^2 = \sum_{j=1}^{m} \sigma_j^2$ so that $Var[W] = \sigma^2$. Let $\delta = ln(1+\epsilon)/\theta$. Then for $0 < a \leq \delta\sigma$.

$$Pr[W > a\sigma] < e^{-\frac{a^2}{2}(1-\frac{\epsilon}{3})} \tag{3}$$

In the next three subsections, we are going to demonstrate how this theorem help derive the optimal number of k in Problem 1 for various failure models.

B. Correlated VM Failure Model (Server)

Now let us look at the model when we have independent server failures and reliable switches. In this model, the VM failures on a single server are correlated while the VM failures across different physical servers are still independent.

Assume m denotes the total number of servers, n_j denotes the number of VMs on the jth server and the failure rate of each physical server is still x. Let S_j denote the number of failed VMs on the jth server and S denote the total number of failed VMs among the $n+k$ VMs of an application. So we have $E[S_j] = n_j x$ and $Var[S_j] = n_j^2 x(1-x)$, thus

$$E[S] = \sum_{j=1}^{m} E[S_j] = (n+k)x \tag{4}$$

$$Var[S] = \sum_{j=1}^{m} \sigma_j^2 = \sum_{j=1}^{m} n_j^2 x(1-x) \tag{5}$$

We now describe how to apply Theorem 1 to our problem. Let W be the normalized value of S, so we have $W = S - E[S]$. And W_j be the normalized value of S_j, so $W_j = S_j - E[S_j]$. Thus, according to Definition 1 the reliability is

$$\begin{aligned} \alpha &= 1 - Pr[S > k] \qquad (6) \\ &= 1 - Pr[W + E[S] > k] \\ &= 1 - Pr[W > k - (n+k)x] \end{aligned}$$

According to 1, the formulated problem as in Problem 1 then becomes

Problem 3:

$$minimize \quad k$$

subject to

$$e^{-\frac{a^2}{2}(1-\frac{\epsilon}{3})} \leq 1 - \alpha \tag{7}$$

$$0 < a \leq \delta\sigma \tag{8}$$

$$0 < e^{\delta\theta} - 1 = \epsilon < 3 \tag{9}$$

$$a\sigma \leq k - (n+k)x \tag{10}$$

To meet the reliability requirement, we would like the probability that the total number of failed VMs larger than k ($Pr[S > k] = Pr[W > k - (n+k)x]$) to be smaller than $1 - \alpha$. According to theorem 1, $Pr[W > \alpha\sigma]$ is bounded by $e^{-\frac{a^2}{2}(1-\frac{\epsilon}{3})}$. So we bound $e^{-\frac{a^2}{2}(1-\frac{\epsilon}{3})}$ by $1 - \alpha$ as eq. 7 shows. And we need bound $Pr[W > k - (n+k)x]$ as $Pr[W > k - (n+k)x] \leq Pr[W > \alpha\sigma]$, thus we derive the constraint in eq. 10. As $e^{-\frac{a^2}{2}(1-\frac{\epsilon}{3})} < 1$, so $1 - \frac{\epsilon}{3} > 0$. We obtain $\epsilon < 3$ as shown in eq. 9.

In order to resolve the problem in this more complex failure model, We need to find the appropriate values for θ, ϵ and a to minimize k in order to bring down the server costs. $|W_j|$ is the lower bound for θ, $j = 1, 2, ..., n+k$. According to Eq. 9, large θ leads to small δ and then the value range of a narrows down according to Eq. 8. As k is related to the value of a according to Eq. 10, we would like the possible range of a to be as wide as possible in order to find the optimal k. Therefore we set θ to the largest value $|W_j|$ could be (i.e., max $n_j(1-x), 1 \leq j \leq m$).

Algorithm 1 Single level round robin placement algorithm

```
j = 1
for i = 1 to n + k do
    place VM_i on the jth server
    j += 1
    if j > m then
        j = 1;
    end if
end for
```

The optimization becomes more convoluted because $Var[W]$ is a function of the VM placement on the physical servers across racks instead of a value in the simplistic model in Section III. In order to maximize the reliability, the desirable placement strategy is distributing VMs on as many physical servers as possible. Following this strategy, we propose a novel heuristic placement algorithm in Algorithm 1 which distributes the $n+k$ VMs on m servers in a round-robin way. Following this method, we distribute the VMs as much as possible, thus the reliability violation risk can be minimized. Note that the algorithm always tries to place the VM to the first server which has enough capacity to host the VM in the order. If the servers are not able to host all the VMs of the application, the algorithm is going to reject the application. In reality, cloud providers could migrate the application to other data centers or add more racks and servers in order to accommodate customers requests.

Given Algorithm 1, we apply the following brute force approach to find the optimal k. Since the minimal k in this failure model must be larger than that (denoted by k_1) in the independent VM failure model in Section III, we apply Algorithm 1 with $k = k_1$ to obtain the values of

$n_j, 1 \leq j \leq m$. Thus we can calculate the corresponding σ. a can be computed by Eq. 10. In order to satisfy Eq. 7, we would like to minimize ϵ, which indicate the value of δ should be as small as possible. Thus δ and ϵ can be obtained by Eq. 8 and Eq. 9, respectively. We abandon all the results which violate Eq. 7. If there is no results left, we increment the value of k by 1 and iterate the above steps until Eq. 7 is satisfied. Now we have the minimal k to satisfy all the constraints. Some simulation results are shown in Section V.

C. Correlated VM Failure Model (ToR Switch and Server)

In this section, we investigate an even more complicated and realistic failure model that both physical servers and ToR switches could fail at some rates. A ToR switch failure would make all physical servers on the corresponding rack inaccessible and hence all the associated VMs on this rack. Thus the VM failures inside a rack are correlated, while the failures across different racks are independent.

Without loss of generality, we assume that the total number of racks in a cloud cluster is m_r, and each rack has m_s physical servers. Note that the adaption to the case where the number of servers is different across racks is trivial. Let y and x denote the failure rates of ToR switches and servers respectively, note that we assume the failure distribution of switches and servers are i.i.d. In addition, S_{ij} denotes the number of failed VMs on the ith server belonging to the jth rack, R_j denotes the total number of failed VMs on the jth rack, and R denotes the total number of failed VMs among the overall $n+k$ VMs. We have n_j to stand for the total number of VMs on the jth rack, and n_{ij} to stand for the total number of VMs on the ith server inside the jth rack.

so $E[R_j]$ can be computed as:

$$\begin{aligned} E[R_j] &= n_j y + (1-y)(\sum_{i=1}^{m_2} E[s_{ij}]) \\ &= n_j y + n_j x(1-y) \end{aligned}$$

and $Var[R_j]$ can be computed as:

$$\begin{aligned} Var[R_j] &= E[R_j^2] - E[R_j]^2 \\ &= (n_j^2 y) + (1-y)(E[(\sum_{i=1}^{m} s_{ij})^2]) - E[R_j]^2 \end{aligned}$$

When rack j is up, the failures of the servers inside this rack are independent. So $E[(\sum_{i=1}^{m} S_{ij})^2]$ can be computed as:

$$\begin{aligned} E[(\sum_{i=1}^{m} S_{ij})^2] &= E(\sum_{i=1}^{m} S_{ij}^2 + \sum_{i_1 \neq i_2, i_1, i_2}^{m} S_{i_1 j} S_{i_2 j}) \\ &= \sum_{i=1}^{m} E[S_{ij}^2] + \sum_{i_1, i_2 = 1, i_1 \neq i_2}^{m} E[s_{i_1} j] E[s_{i_2} j] \\ &= \sum_{i=1}^{m} n_{ij}^2 x + \sum_{i_1 \neq i_2, i_1, i_2} n_{i_1 j} n_{i_2 j} x^2 \end{aligned}$$

So both $E[R_j]$ and $Var[R_j]$ are the functions of y, x and the placement of VMs across servers on this rack (i.e., $n_{ij}, 1 \leq i \leq m_s$). And it is not hard to see that both $E[R] = \sum_j E[R_j]$ and $Var[R] = \sum_j Var[R_j]$ are also the functions of y, x and the VM placement across m_r racks.

In order to apply Theorem 1, we have $W = R - E[R]$, and $W_j = R_j - E[R_j]$. And the reliability α becomes $1 - Pr[R > k] = 1 - Pr[W > k - E[R]]$.

We need to find the appropriate values of the parameters to minimize the value of k. Similar as previous two failure models, we want to solve the following optimization problem:

Problem 4:

$$\text{minimize} \quad k$$

subject to

$$e^{-\frac{a^2}{2}(1-\frac{\epsilon}{3})} \leq 1 - \alpha \tag{11}$$

$$0 < a \leq \delta\sigma \tag{12}$$

$$0 < e^{\delta\theta} - 1 \leq \epsilon < 3 \tag{13}$$

$$a\sigma \leq k - E[R] \tag{14}$$

Algorithm 2 Two level round robin placement algorithm

$c_j = 1$ for all $1 \leq j \leq m_r$
$j = 1$
for $i = 1$ to $i = n + k$ **do**
  place VM_i on $c_j th$ server on jth rack;
  $c_j += 1$;
  if $c_j > m_s$ **then**
    $c_j = 1$;
  end if
  $j += 1$;
  if $j > m_r$ **then**
    $j = 1$;
  end if
end for

As in Section III and IV-B, we set θ to its lower bound $\max n_j(1-y), 1 \leq j \leq m_r$ to maximize the selection range of a. Since $Var[W]$ is a function of the VM placement, we must have a desirable placement to infer the optimal k. In Algorithm 2, we introduce a heuristic placement algorithm sharing the same spirit in Algorithm 1 – distributing $n+k$ VMs on m_r racks (and then m_s servers on a rack) in a round-robin way to maximize reliability.

The algorithm works as shown in follows. To place VM_i, we search the first rack which has enough capacity to host it in the order of $j, j+1, ., m_r, 1, 2, , j-1$ assuming VM_{i-1} chose $Rack_{j-1}$. The located rack is denoted as $Rack_x$. Assuming the last VM allocated on this rack choose $Server_a$, we search the first server which has enough capacity to host VM_i in the order of $a+1, a+2, , m_s, 1, 2, , a$ and place VM_i on that server. With this algorithm, we distribute the VMs as much as possible, thus the influence of correlated failures is minimized.

Since the minimal k in this failure model must be larger than that (denoted by k_2) in the simpler model in Section IV-B, we run Algorithm 2 to obtain the overall placement (i.e., $n_{ij}, 1 \leq$

$i \leq m_s, 1 \leq j \leq m_r$) starting with $k = k_2$. Then we obtain the corresponding σ and a by Eq. 14. Then δ and ϵ are obtained via Eq. 12 and Eq. 13 respectively. Then we need ensure Eq. 11 is satisfied by adjusting the current value of k and corresponding placement as necessary. If it is not, we increment the value of k by 1 and iterate the above procedure until we find the minimal viable k. Some simulation results can be found in Section V.

V. Performance Evaluation

In this section, we show some simulation results and compare them to that from Section IV. We simulate a data center topology as in Fig. 1, consisting of 50 racks, each of which has 20 servers. And each server can host at most 10 VMs. We set the homogeneous failure rates of ToR switches and physical servers to 0.05 [3] and 0.02, respectively. [2]

A. Comparison of Reliability

Table I compares the reliability achieved by our presented approaches in Section III and IV with two different simulation setups. We show the comparison results in three different failure models discussed in Section III, IV-B and IV-C (denoted by the first column of the table) when the number of requested VMs $n = 100$. In this experiment, we tune the number of backup VMs k so that the resulting reliability in our approach is equal to 95% approximately. Then given the resulting k value, the simulation 1 uses the proposed VM placement algorithms (Algorithm 1 and Algorithm 2) to place the $100+k$ VMs onto the servers/racks and then serves with the failure rate 0.02 and 0.05, respectively. We iterate these steps for $5,000$ applications, and calculate the probability of the case where the number of failed VMs is less than or equal to k as the reliability of the simulation. The simulation 2 is mostly the same as simulation 1 except that it distributes the VMs onto different racks and servers randomly instead of using our placement algorithms.

The results in Table I show that our approaches in the correlated VM failure models bound the reliability very well. Given the same number of backup VMs, the simulation 1 shows somewhat higher reliability because it is the average from $5,000$ samples while our solution bounds the worse case. The simulation 2 shows lower reliability compared to our approach, which demonstrates the effectiveness of our placement algorithms. Note that the independent VM failure model (Section III) is an exception because there is no placement issue, so we use a different approach to solve the problem.

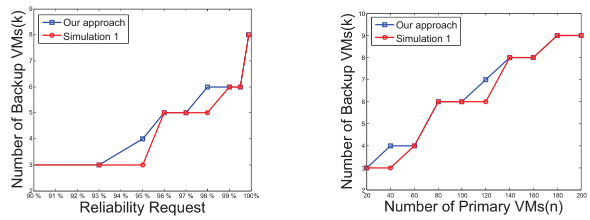
B. Comparison of Number of Backup VMs

Besides the experiments we performed in Section V-A, we performed another set of experiments which compare the minimal number of backup VMs needed among our approaches and two simulation setups given the fixed value of reliability. In the simulation 1, we use Algorithm 1 and Algorithm 2 in the corresponding failure models, gradually increase the value of k and calculate the corresponding reliability until we reach a value which is equal to or larger than the target reliability (ideally the value should be pretty close to the target, i.e. 1.01 α). The simulation 2 is mostly the same as proposed approach except that we adopt random placement instead of our well-designed algorithms. Fig. 2, Fig. 3 and Fig. 4 plot the simulation results for the three different failure models we

[2] We also extend our simulations under the situation that the failures come following some other distribution in the journal version paper, it has not been shown here due to the interest of space.

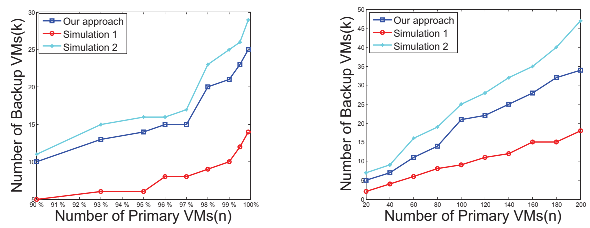
TABLE I: Comparison of reliability between Section IV and simulations

Model	k	Our approach	Simulation 1	Simulation 2
Section III	4	95%	96%	N/A
Section IV-B	14	95.9%	98.8%	93.7%
Section IV-C	37	95.3%	99.2%	92.6%



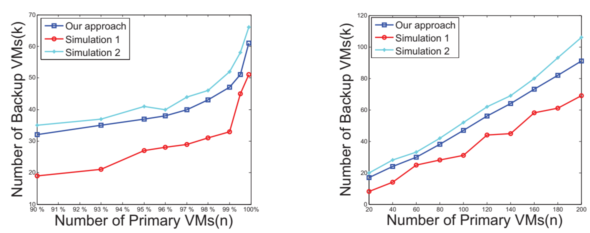
(a) Change reliability requests (b) Change number of VMs requests

Fig. 2: Results of independent VM failure model



(a) Change reliability requests (b) Change number of VMs requests

Fig. 3: Results of correlated VM failure model for server



(a) Change reliability requests (b) Change number of VMs requests

Fig. 4: Results of correlated VM failure model for ToR switch and server

have investigated in this paper.

The left figure in Fig. 2 plots the number of backup VMs for both simulation 1 and our proposed solution of independent VM failure model by varying the reliability requirements from 90% to 99.9% (the number of requested VMs is fixed at 100). It is clear that the higher the requested reliability, the more backup VMs we need. It can be seen from the figure that our approach matches the simulation results closely. Keep in mind that in this model, we don't have any placement issues. The right figure varies the number of requested VMs from 20 to 200 and plots the number of backup VMs with the fixed reliability requirement 99%. Similarly, the results of our approach match closely with the simulation. We do not plot the results of simulation 2 here because there is no placement issue in the independent VM failure model.

Fig. 3 and Fig. 4 show the simulation results for the correlated VM failure model for servers in Section IV-B and the correlated VM failure model for ToR switches and servers in Section IV-C, respectively. It can be seen from the figure that our approach servers as an upper bound of the simulation 1 results in each model as it bounds the worst case while the simulations results show the average. In addition, our approach requires much less back VMs than the simulation 2 given the same experiment configuration, which proves the effectiveness of our placement algorithms.

Fig. 4 shows the results of bi-level failure model in Section IV-C. As we can see from the results, the value of k is larger than Fig. 2 and Fig. 3, which is as expected. In this scenario, we have two levels of failures, which can cause more correlated VM failures than the previous two scenarios. In order to not violate the reliability requirement, we need allocate more backup resources than the previous two cases.

It is not hard to see that the higher the failures are correlated, the more backup VMs are needed to achieve some reliability requirements.

VI. Conclusion

In this paper, we have, for the first time, proposed a solution to directly derive the optimal number of backup VMs and VM placement strategy for a cloud cluster given the required number of VMs and reliability target as part of SLA. We investigated three different failure models from simplistic to complex and realistic failure models: 1) independent VM failure model, 2) correlated VM failure model for server, and 3) correlated VM failure model for ToR switch and server. Our solution consists of the statistical failure modeling, some novel heuristic VM placement algorithms, and the adoption of a tight tail bound theorem to infer the minimal number of backup VMs needed to guarantee the reliability SLA.

Beyond the work presented in this paper, we are working on the following aspects actively.

- Derive the solution for more complex failure model (e.g., failures on core and aggregate layers) and do more comprehensive simulations and evaluations;
- Perform resource-driven optimization: our solution has only taken reliability into consideration, thus the proposed VM placement algorithms always try to distribute the VMs among the racks and servers as much as possible to minimize the impact of correlated failures. However, this might increase the energy consumption and communication bandwidth for cloud providers and performance overhead between VMs for an application. Considering the problem in more dimensions (e.g., power consumption, bandwidth cost, etc.) leads to a more complex optimization problem we are currently working on.

VII. Acknowledgements

This research is supported by Google's Research Award, NSF CSR-1409809,National Science Foundation under grant numbers OCI-1203560, ACI-1445806, and ACI-1541215

References

[1] Yeow, W.-L., Westphal, C., and Kozat, U. C. Designing and embedding reliable virtual infrastructures. *ACM SIGCOMM Computer Communication Review 41*, 2 (2011), 57–64.

[2] Beloglazov, A., and Buyya, R. Energy efficient resource management in virtualized cloud data centers. In *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing* (2010), IEEE Computer Society, pp. 826–831.

[3] Gill, P., Jain, N., and Nagappan, N. Understanding network failures in data centers: measurement, analysis, and implications. In *ACM SIGCOMM Computer Communication Review* (2011), vol. 41, ACM, pp. 350–361.

[4] Guazzone, M., Anglano, C., and Canonico, M. Energy-efficient resource management for cloud computing infrastructures. In *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on* (2011), IEEE, pp. 424–431.

[5] Hsu, C.-H., Chen, S.-C., Lee, C.-C., Chang, H.-Y., Lai, K.-C., Li, K.-C., and Rong, C. Energy-aware task consolidation technique for cloud computing. In *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on* (Nov 2011), pp. 115–121.

[6] Koslovski, G., Yeow, W.-L., Westphal, C., Huu, T. T., Montagnat, J., and Vicat-Blanc, P. Reliability support in virtual infrastructures. In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on* (2010), IEEE, pp. 49–58.

[7] Lin, M., Wierman, A., Andrew, L., and Thereska, E. Dynamic right-sizing for power-proportional data centers. In *INFOCOM, 2011 Proceedings IEEE* (April 2011), pp. 1098–1106.

[8] Lu, K., Roblitz, T., Yahyapour, R., Yaqub, E., and Kotsokalis, C. Qos-aware sla-based advanced reservation of infrastructure as a service. In *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on* (Nov 2011), pp. 288–295.

[9] Alon, N., and Spencer, J. H. *The probabilistic method*. John Wiley & Sons, 2004.

[10] Data center switch solutions white paper, http://en.wikipedia.org/wiki/Business_logic_layer.

[11] Yang, Z., Liu, L., Qiao, C., Das, S., Ramesh, R., and Du, A. Y. Availability-aware energy-efficient virtual machine placement. In *Communications (ICC), 2015 IEEE International Conference on* (June 2015), pp. 5853–5858.

[12] Zhang, W., Zhang, H., Chen, H., Zhang, Q., and Cheng, A. Improving the qos of web applications across multiple virtual machines in cloud computing environment. In *Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW), 2012 IEEE 26th International* (May 2012), pp. 2247–2253.

[13] Zhao, Q., Xu, J., and Liu, Z. Design of a novel statistics counter architecture with optimal space and time efficiency. *ACM SIGMETRICS Performance Evaluation Review 34*, 1 (2006), 323–334.