DOI 10.12694/scpe.v14i4.933
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 14, Number 4, pp. 279–290. http://www.scpe.org

ANALYTICAL INVESTIGATION OF AVAILABILITY IN A VISION CLOUD STORAGE CLUSTER

DARIO BRUNEO† FRANCESCO LONGO† DAVID HADAS‡ AND ELLIOT K. KOLODNER‡

Abstract. The goal of VISION Cloud, a European Commission funded project, is to design a new scalable and flexible storage cloud architecture able to provide data-intensive storage cloud services. The proposed environment employs a distributed file system on top of a set of storage rich nodes composing a cluster. Several clusters constitute a data center, while multiple geographically distributed data centers form a single storage cloud. In this paper, we focus on a single VISION Cloud storage cluster, providing a stochastic reward net model for an investigation of its availability. The proposed model is a first attempt at obtaining a quantification of the availability level of the cloud storage provided by the VISION Cloud architecture.

1. Introduction. Focusing on IT assets as commodities and on-demand usage patterns, cloud computing greatly mitigates the cost of service provisioning, through tools such as virtualization of hardware, rapid service provisioning, scalability, elasticity, accounting granularity, and cost allocation models. However, Future Internet, Internet of Things, and, in general, the rich digital environment we are experiencing nowadays pose new requirements and challenges in the Cloud area, especially with respect to the explosion of personal and organizational digital data. In fact, the strong proliferation of data-intensive services and the digital convergence of telecommunications, media, and ICT will surely amplify the explosion of raw data and the dependence on data services. System performance and dependability [3, 15], energy consumption [4], workload characterization [9] are only few examples of the Cloud related research trends that have been investigated in the last years.

VISION Cloud [11] is a European Commission Seventh Framework Programme (FP7/2006-2013) funded project. Its goal is to design a new scalable and flexible storage cloud architecture that allows the implementation of data-intensive storage cloud services, scalability and flexibility referring to the ability of the proposed architecture to deal with a large number of concurrent users and in allowing the provisioning of different kinds of storage services. Raising the abstraction level of storage, enabling data mobility across providers, allowing computational storage and content-centric access to storage and deploying new data-oriented mechanisms for QoS and security guarantees are some of the means that VISION Cloud exploits in order to achieve such a goal. With respect to QoS guarantees, reliability, availability, and fault tolerance and resiliency characteristics of the provided services are important aspects that need to be taken into consideration.

The single storage resource in the VISION Cloud reference architecture is represented by the *storage cluster* which usually includes hundreds of storage rich nodes. Such a basic element is able to store data objects and provide computational power on top of it in a transparent way. This is obtained by the use of a distributed file system installed on the storage cluster. In the prototype implementation of the architecture that the VISION Cloud project provides, the General Parallel File System for Shared Nothing Clusters* (GPFS-SNC) [8] is exploited. A high level of availability and resiliency to faults is achieved by replicating data objects across different storage clusters. VISION Cloud considers a single cloud as composed by multiple distributed data centers interconnected through dedicated networks. Each data center can be composed of multiple storage clusters.

In this paper, we provide an analytic model for the availability investigation of a storage cluster in the context of the storage cloud environment proposed by the VISION Cloud project. The model is based on stochastic reward nets (SRNs) [6], an extension of generalized stochastic Petri nets. SRNs are a graphical tool for the formal high-level representation of systems characterized by concurrency, mutual exclusion, conflict, and synchronization dynamics. Thus, such a formalism is useful in capturing the key concepts of large-scale distributed systems [5, 2] and the model we propose allows obtaining information about the reached availability level of a VISION Cloud storage cluster varying both structural and timing system parameters. Structural parameters are related to the number of nodes in the cluster, the number of disks in each node, the cluster file system metadata replication level, and similar information. Timing parameters involve information about the

†Dipartimento di Ingegneria DICIEAMA, Università degli Studi di Messina, Messina, Italy ({dbruneo,flongo}@unime.it)
‡IBM Research Labs Haifa, Haifa, Israel, ({kolodner,davidh}@il.ibm.com)
*GPFS is a trademark of International Business Machines Corp., registered in many jurisdictions worldwide.

time necessary to specific events (e.g., disk or node failure) to occur or specific operations (e.g., disk or node repair, cluster file system metadata recovery) to be performed.

Prior work deals with the performance analysis of storage cloud infrastructure [13] while little effort has been put in the context of availability analysis [20]. In this context, the majority of the work mainly considers replica placement policies [12, 19] without taking into consideration real case studies as done in our work. In fact, our model could be exploited by a VISION Cloud administrator in order to opportunely build the infrastructure accordingly to the desired availability level both from the hardware (e.g., computation, storage, network resources) and the software (e.g., replication schema, cluster file system configuration) points of view. Moreover, it could represent an useful instrument for model assisted SLA management.

The paper is organized as follows. Section 2 gives a background about Petri nets with particular reference to SRNs. Section 3 provides an overview of the VISION Cloud reference architecture and illustrates how GPFS-SNC is exploited in the reference implementation. Section 4 formally describes the considered scenario while Section 5 illustrates how such a scenario is modeled through the use of SRNs. Section 6 provides some numerical results. Finally, Section 7 concludes the paper with some final remarks on the proposed approach and on possible future work.

2. Background about Petri Nets. A Petri net (PN) [16] is a 4-tuple: $PN = (P, T, A, M)$, where P is the finite set of *places* (represented by circles), T is the finite set of *transitions* (represented by bars), A is the set of *arcs* (connecting elements of P and T) and M is the set of markings each of which denotes the number of tokens in the places of the net. Graphically, a PN is a directed bipartite graph, with two types of nodes: *places* and *transitions*. A directed arc connecting a place (transition) to a transition (place) is called an input (output) *arc* of the transition. A positive integer called multiplicity can be associated with each arc. Each place may contain zero or more tokens. A transition is *enabled* if each of its input places has at least as many tokens as the multiplicity of the corresponding input arc. A transition can *fire* when it is enabled, and upon firing, a number of tokens equal to the multiplicity of the input arcs is removed from each of the input places, and a number of tokens equal to the multiplicity of the output arcs is deposited in each of its output places. In stochastic Petri net (SPN), exponentially distributed firing times can be associated to the net transitions so that the stochastic process underlying a SPN is a homogeneous CTMC. In generalized stochastic Petri nets (GSPN) [14], transitions are allowed to be either *timed* (exponentially distributed firing time, drawn as rectangular boxes) or *immediate* (zero firing time, represented by thin black bars). Immediate transitions always have priority over timed transitions and if both timed and immediate transitions are enabled in a marking then timed transitions are treated as if they are not enabled. If several immediate transitions compete for firing, a specified probability mass function is used to break the tie. A marking of a GSPN is called *vanishing* if at least one immediate transition is enabled in it. A marking is called *tangible* otherwise. GSPN also introduces the concept of *inhibitor arc* (represented by a small hollow circle at the end of the arc) which connects a place to a transition. A transition with an inhibitor arc can not fire if the input place of the inhibitor arc contains more tokens than the multiplicity of the arc. SRNs [6] are extensions of GSPNs. In SRNs, every tangible marking of the net can be associated with a reward rate thus facilitating the computation of a variety of performance measures. Key differences with respect to GSPNs are: (1) each transition may have an enabling function (also called a guard) so that a transition is enabled only if its marking-dependent enabling function is true; (2) marking dependent arc multiplicities are allowed; (3) marking dependent firing rates are allowed; (4) transitions can be assigned different priorities; (5) besides traditional output measures obtained from a GSPN, such as throughput of a transition and mean number of tokens in a place, more complex measures can be computed by using reward functions.

3. The VISION Cloud storage environment. In this section, we provide an overview of the storage cloud environment proposed by the VISION Cloud project [1] focusing on the implemented physical infrastructure and on the data model. We also provide details about GPFS-SNC [8], and about how it is used in the reference implementation of VISION Cloud.

3.1. The proposed storage cloud environment. The goal of the VISION Cloud project is to provide efficient support for data-intensive applications. Moreover, a content-centric view of storage services is provided. Five main areas of innovation drive the VISION Cloud platform design and implementation [10]: i) content is

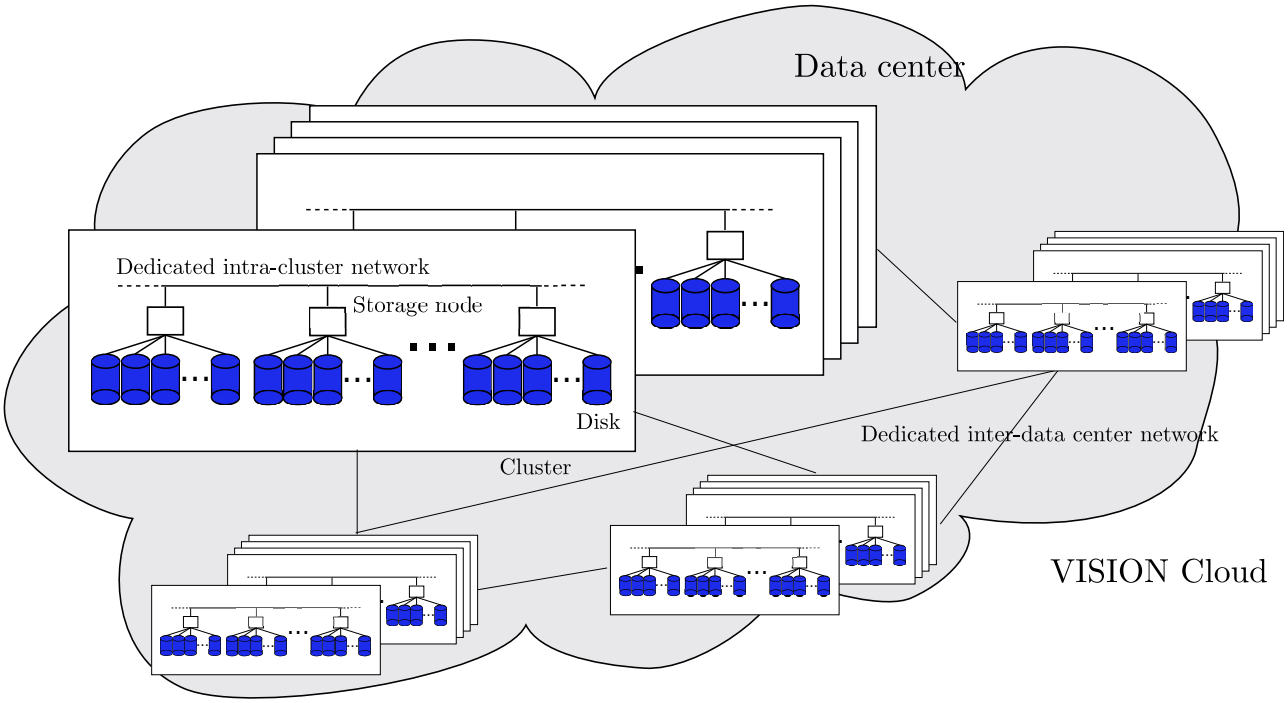

FIG. 3.1. *The VISION Cloud reference infrastructure.*

managed through data objects that can have rich metadata associated with them, ii) data lock-in is avoided by allowing migration of data across administrative domains, iii) computations are moved close to data through the use of storlets in order to avoid costly data transfers, iv) efficient retrieval of objects is allowed based on object content, properties, and relationships, and v) a high QoS level is guaranteed together with security and compliance with international regulations.

The storage cloud environment proposed by the VISION Cloud project is built on top of an infrastructure consisting of multiple data centers, potentially distributed worldwide. Each data center can be composed of one or more storage clusters containing physical resources providing computational, storage, and networking capabilities. The data centers need to be connected by dedicated high speed networks.

Each storage cluster is composed of storage rich nodes that can be built from commodity hardware and connected by commodity network devices. In fact, as common for cloud infrastructures, the storage cloud is built from low cost components and the desired reliability level is assured through the software layer. The software stack also builds advanced functionalities on top of this foundation. An example of initial hardware configuration could be 4 or 8 multiprocessor nodes with 12 to 16 GB of RAM each. Each node could have 12 to 24 high capacity direct attached disks (e.g., 2TB SATA drives). The architecture, design, and implementation of the VISION Cloud architecture supports a system with hundreds of storage clusters, where each storage cluster can have several hundred nodes and the storage clusters are spread out over dozens of data centers. Such a reference infrastructure is represented in Fig. 3.1.

The VISION Cloud data model is based on the concept of data object. A data object contains data of arbitrary type and size. It has a unique identifier that allows users to access it through the whole cloud. An object is written as a whole and cannot be partially updated, although it can be partially read. An object may be overwritten, in which case the whole content of the object is replaced. Versioning is supported. Data objects are stored in containers (with each data object residing within a single container). Containers provide easy data management, isolation, and placement policies. A rich metadata model allows system and user metadata to be associated with containers and objects. User metadata is set by the user and is transparent to cloud storage system. System metadata has concrete meaning to the cloud storage system.

The VISION Cloud data model extends traditional storage cloud models to include computation on the data objects, which is performed within the cloud storage environment through storlets. Storlets are software agents that are triggered according to specific events.

Objects may be replicated across multiple clusters and data centers. The degree of replication and placement restriction policies are defined and associated with an object's container. VISION Cloud employs a symmetric replication mechanism, where any operation on an object can be handled at any of its replicas. A storlet, when triggered, is executed once, usually at the site where the triggering condition first occurred.

3.2. GPFS-SNC as underlying distributed file system. In the storage cloud environment proposed by the VISION Cloud project, the simpler and lower level storage unit is the storage cluster. A distributed file system runs over the storage resources provided by each cluster (i.e., the servers and their direct attached disks). This allows each node to access the data objects stored in the cluster and to provide computational power on top of it by serving user requests and allowing the execution of storlets. In the current implementation of the VISION Cloud stack, the General Parallel File System for Shared Nothing Clusters (GPFS-SNC) is exploited in order to build such a distributed file system.

General Parallel File System (GPFS) [17] is a parallel file system for computer clusters providing the services of a general-purpose POSIX file system running on a single machine. GPFS supports fully parallel access to both file data and file system data structures (file system metadata). Moreover, administrative actions (e.g., adding or removing of disks) are also performed in parallel without affecting access to data. GPFS achieves its scalability through its shared storage architecture where all nodes in the cluster have access to all storage. Files are striped across all disks in the file system providing load balancing and high throughput. Large files are divided into equal sized blocks which are placed on different disks in a round-robin fashion. GPFS uses distributed locking to synchronize access to shared disks ensuring file system consistency while still allowing the necessary parallelism. As an alternative or a supplement to RAID, GPFS supports replication, storing two or more copies of each data or file system metadata block on different disks. Replication can be enabled separately for data and file system metadata.

The GPFS-SNC file system [8] builds on the existing GPFS distributed file system extending it to a shared-nothing cluster architecture. Such scenario is the one being used in the current implementation of VISION Cloud. In shared-nothing cluster architecture, every node has local disks behaving as primary server for them. If a node tries to access data and such a data is not present on a local disk, a request is sent to its primary server to transfer it.

In the reference implementation of VISION Cloud, each object is stored as a file in GPFS-SNC on a single disk. The files corresponding to objects are neither striped nor replicated within a cluster. Rather, additional object replicas are created in other VISION Cloud clusters in order to guarantee the desired level of availability. Typically a $(1+1, 1+1)$ schema is used for object replication, i.e., each object is replicated in two data centers at two storage clusters in each data center. However, other replication schema can be used changing the replication level. GPFS-SNC file system metadata is replicated with a certain level of redundancy in order to guarantee that the file system structure is preserved in the presence of faults and that it is possible to determine which object has been lost and needs to be recovered. The use of GPFS-SNC in the VISION Cloud architecture is graphically depicted in Fig. 3.2. In the remainder of the paper, we model a generic cluster file system with characteristics similar to those described above.

4. Problem formulation. In the following, we formally describe the scenario we take into consideration in the present work. Let us consider a VISION Cloud cluster composed by N nodes. Each node is associated with D directed attached storage (DAS) disks where both the distributed file system metadata and data (VISION Cloud objects) are stored. Note that, in the following we will consider only the distributed file system metadata (simply *metadata* from now on) while we ignore the system and user metadata associated with VISION Cloud objects, which are treated as files from the point of view of the cluster file system. Disks and nodes can fail. Let us suppose that the time to fail of a single disk (node) is exponentially distributed with rate λ_{df} (λ_{nf}). Disks (nodes) are repaired in an exponentially distributed time with rate μ_{dr} (μ_{nr}).

Disk and node failures are assumed to be destructive. In other words, when a disk fails the metadata and data stored in it are lost. Similarly, in order to maintain the distributed file system consistency, when a node fails metadata and data stored in all its attached disks are considered lost. VISION Cloud objects are stored

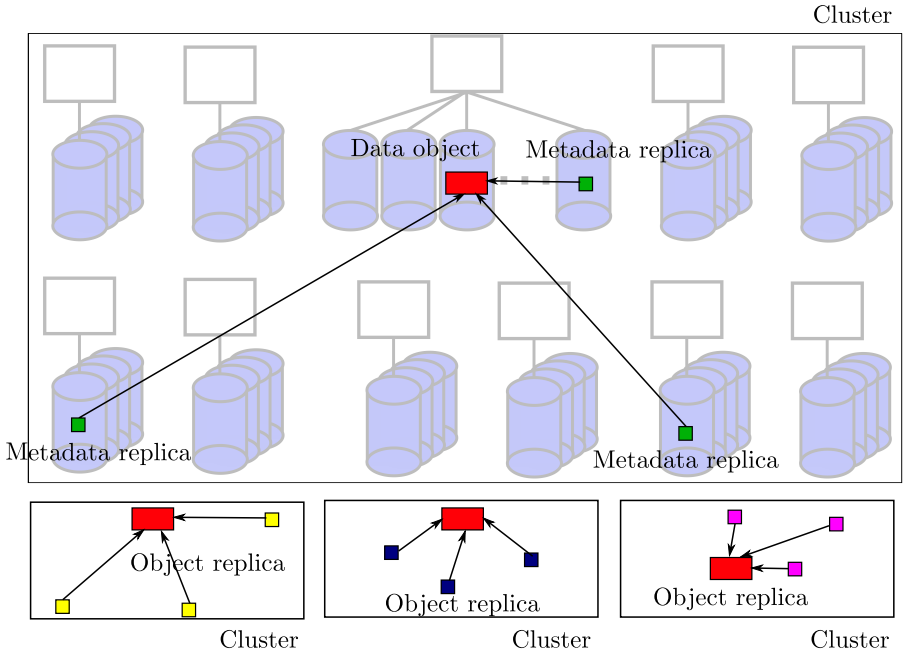

FIG. 3.2. *The use of GPFS-SNC in the VISION Cloud architecture.*

in the cluster without any striping or data replication, i.e., each object is fully contained in a single disk as a single file. On the other hand, metadata is scattered on the cluster disks and metadata records for each file are replicated on different nodes. Let us assume the level of metadata replication for each file to be R. This value is an internal parameter of the cluster file system, it is usually set during installation and it cannot be dynamically changed at runtime. When a disk fails the metadata that was present on it is replicated in a working disk in order to restore the correct level of replication. The process of metadata replication recovery takes an exponentially distributed amount of time with rate μ_{mr} to be performed.

VISION Cloud objects are replicated in other clusters. In the case of failure, the VISION Cloud Resiliency Manager (RM) is responsible for returning the storage Cloud to the proper level of resiliency. In fact, if a disk fails, a scan of the distributed file system metadata allows the RM to determine which objects were lost. Then, the RM contacts the other clusters in the Cloud (clusters in the same data center are usually queried first, since they are the closest) in order to recover the data from a replica and restore the objects into the cluster.

Let us consider a single VISION Cloud object X stored in the cluster. Objects are uniformly distributed over the cluster disks, i.e., when an object needs to be stored the target disk is randomly chosen accordingly to an uniform distribution. For such a reason, if a disk fails the probability that object X becomes unavailable (if it was still available at the failure time) is $1/x$ where x is the number of disks actually working with $0 < x \leq N \cdot D$. On the other hand, if a node fails the probability that object X becomes unavailable depends on the number of working disks that were attached to the failed node. In a first approximation, we assume that, given a VISION Cloud replication schema, at least one of the clusters in which object X was stored is always available for data recovery. Moreover, let us assume that, in order to recover an entire disk full of data, an exponentially distributed time is necessary with rate μ_{fd}. Among other factors, such a time can depend on the network bandwidth that is present between the consider cluster and the cluster from which the objects will be recovered.

Of course, given that the RM performs the data recovery as soon as possible after a disk failure, free space on other available disks is necessary in order to restore the lost objects in the cluster. Let us assume that the recovery can be performed only if there are at least K working disks in the local cluster. K can be computed considering the average disk capacity, the average object dimension, and the average number of objects in a cluster. For example, if c is the average fraction of occupied space in a disk then $K = \lceil N \cdot D \cdot c \rceil$. The time that is necessary to recover a single disk is also affected by the parameter c. In fact, the time needed to recover a

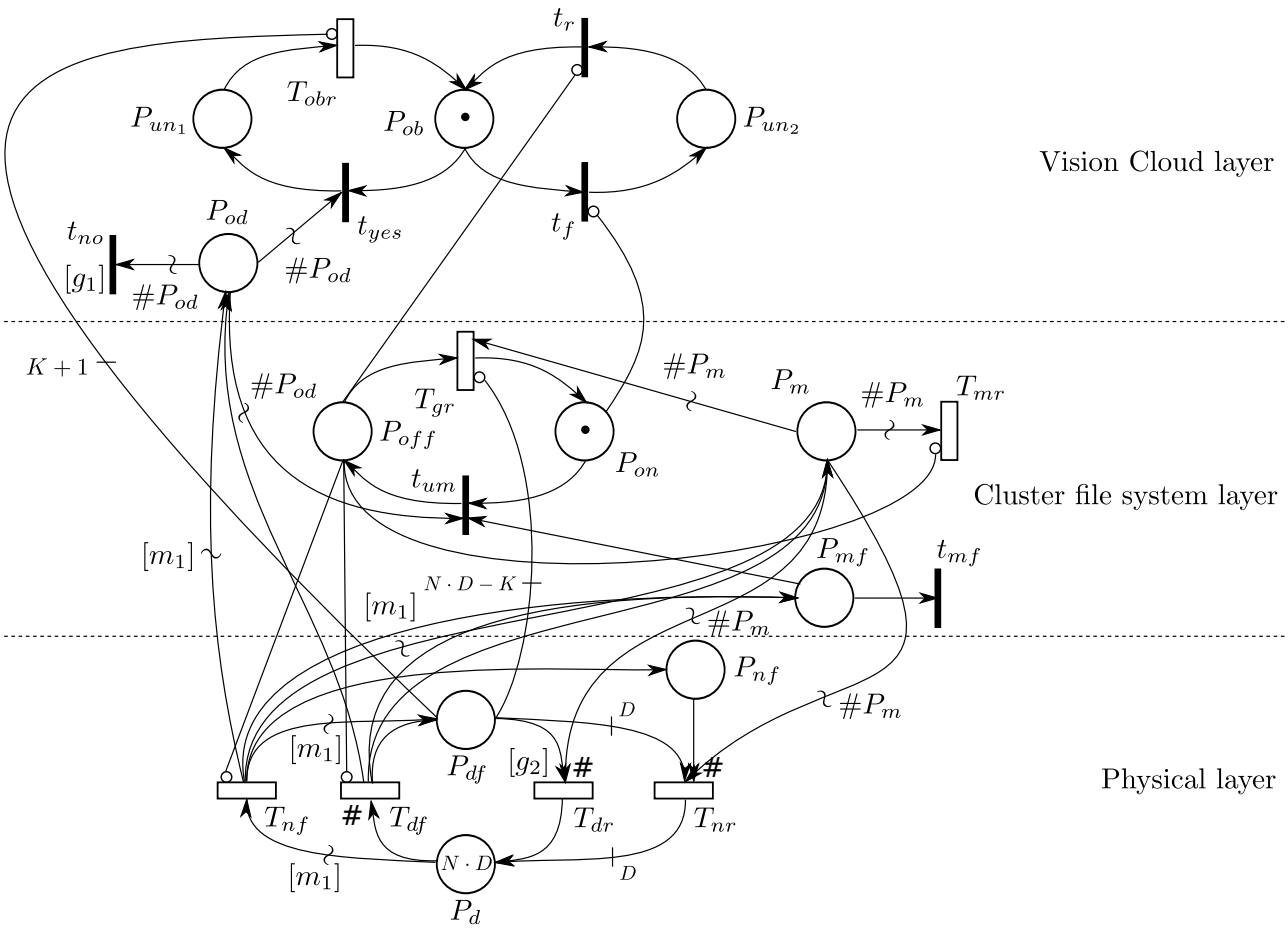

FIG. 5.1. *SRN model for a Vision Cloud cluster*

single disk (characterized by an average fraction of occupied space c) is considered as exponentially distributed with rate $\mu_{obr} = \mu_{fd}/c$.

Disk failures can affect the availability of a generic VISION Cloud object X even if it is not stored in the disk that fails. In fact, the distributed file system correctly works only until at least one metadata replica for each file is present on a working disk. If all the metadata replica for a single generic file are lost, the cluster file system is unmounted thus making object X unavailable. We suppose that when the cluster file system is unmounted no disk failures can occur and no objects recovery can be performed. The file system will be mounted again only when a sufficient number of disks are available again (we suppose such a sufficient number to be K). Moreover, all the objects that were originally present on the cluster need to be recovered. This is assumed to take an exponentially distributed time with rate μ_{gr} that can be computed as a function of μ_{fd}. In particular, if $N \cdot D$ disks are present in the cluster with an average fraction of occupied space c, then $\mu_{gr} = \mu_{fd}/(c \cdot K) = \mu_{fd}/(c \cdot \lceil N \cdot D \cdot c \rceil)$.

Note that a complete description of the VISION Cloud architecture and of its components is out of the scope of this paper. For further details please refer to [11].

5. The model. Figure 5.1 shows the SRN model for the Vision Cloud cluster described above. Three layers have been identified: physical layer (concerning node and disk failures and repairs), distributed file system layer (modeling the cluster file system metadata), and Vision Cloud layer (associated to object availability).

Places P_d and P_{df} represent working and failed disks, respectively. Place P_d initially contains $N \cdot D$ tokens while place P_{df} is initially empty. Each token represents a single disk. Transitions T_{df} and T_{dr} represent disk

failure and repair events moving tokens between places P_d and P_{df}. Rates of these transitions are considered to be dependent on the number of tokens in places P_d and P_{df}, respectively, so that the overall disk failure rate is equal to λ_{df} multiplied by the number of available disks while the overall repair rate is given by μ_{dr} multiplied by the number of failed disks. These marking dependent firing rates are represented by the # symbol near the corresponding arc.

Transitions T_{nf} and T_{nr} represent node failure and repair events. The failure of a single node is modeled as the contemporaneous failure of more than one disk by letting transition T_{nf} to move more than one token from place P_d to place P_{df}. This is obtained by associating to the arcs connecting transition T_{nf} to places P_d and P_{df} a multiplicity that depends on the actual status of the net through function $[m_1]$. In particular, the number of disks that contemporaneously fail when a node fails is assumed to be dependent on the actual number of failed nodes and disks: if nf nodes and df disks are failed, then we assume that the average number of disks that fail when a node fails is given by $(N \cdot D - df)/(N - nf)$. Considering that transition T_{nf} also puts a token in place P_{nf} at each node failure event (i.e., tokens in place P_{nf} model the number of failed nodes), we have:

$$[m_1] = \#P_d/(N - \#P_{nf})^{\dagger}.$$

The rate of transition T_{nf} also depends on the actual status of the net and, in particular, it is equal to λ_{nf} multiplied by the number of working nodes, i.e., $\lambda_{nf} \cdot (N - \#P_{nf})$. The repair of a single node is modeled as the contemporaneous repair of D disks. For such a reason, each firing of transition T_{nr} moves D tokens from place P_{df} to place P_d. Also, one token is removed from place P_{nf} in order to model a single node being repaired. The rate of transition T_{nr} depends on the number of tokens in place P_{nf} so that the overall node repair rate is equal to λ_{nr} multiplied by the number of failed nodes.

Finally, transition T_{dr} is associated with guard function $[g_2]$ that allows single disks to be repaired only if there is a sufficient number of working nodes:

$$[g_2] = \begin{cases} 1, & \text{if } \#P_{df} > D \cdot \#P_{nf} \\ 0, & \text{otherwise} \end{cases}$$

In this way, if all the failed disks correspond to failed nodes, transition T_{dr} is disabled.

Place P_m represents failed metadata replicas that need to be restored. It initially contains zero tokens. As soon as a disk fails (transition T_{df} fires) or a node fails (transition T_{nf} fires), a number of tokens equal to the number of failed disks is moved in place P_m representing the corresponding metadata replicas being lost. Transition T_{mr} represents the time necessary for the failed metadata replicas to be restored on the cluster. It is associated with a rate equal to μ_{mr} and, as soon as it fires, it flushes the content of place P_m modeling all the metadata replicas being restored. This is implemented by associating to the arc connecting transition T_{mr} to place P_m a multiplicity equal to the number of tokens in such a place.

As soon as a certain number of disks fail (either transition T_{df} or transition T_{nf} fires), a token is also put in place P_{mf} enabling the conflicting immediate transitions t_{mf} and t_{um}. Transition t_{mf} models the probability for the cluster file system to continue to work properly after the newly occurred failure conditioned to the fact that it was correctly working when the failure occurred. Such a probability depends on the actual number of working nodes and metadata replicas present in the cluster so it can be computed as a function of the current number of tokens in places P_d and P_m. As soon as transition t_{mf} fires, it removes the token from place P_{mf} leaving everything else unmodified. On the other hand, transition t_{um} models the probability for the cluster file system to be unmounted after the newly occurred failure conditioned to the fact that it was correctly working when the failure occurred. Also in this case, such a probability depends on the actual number of working nodes and metadata replicas present in the cluster and it can be computed as a function of the current number of tokens in places P_d and P_m. Given that transitions t_{mf} and t_{um} are conflicting and no other transition is contemporaneously enabled the sum of their associated probabilities needs to be equal to one. As soon as transition t_{mf} fires, a token is moved from place P_{on} to place P_{off}. Moreover, the token in place P_{mf} is removed.

Place P_{on} represents a working distributed file system while place P_{off} represents a faulty file system. When the cluster file system is down, no new metadata replica can be created (inhibitor arc from place P_{off} to transition T_{mr}) and no disks or nodes can fail (inhibitor arcs from place P_{off} to transitions T_{df} and T_{nf}).

Transition T_{gr} represents the time necessary to repair the distributed file system after a crash due to

†The notation $\#P$ indicates the number of tokens in place P.

metadata destruction, to recover all the objects from the replicas in other Vision Cloud clusters, and to create the metadata replicas. It is associated with a rate equal to μ_{gr}. Such recovery operation can be performed only after the repair of at least K disks (inhibitor arc from place P_{df} to transition T_{gr} with multiplicity $N \cdot D - K$). As soon as transition T_{gr} fires, a token is put back to place P_{on} (the cluster file system is up again) and all the tokens in place P_m are flushed modeling the recovery of all the failed metadata replicas. This is implemented by associating to the arc connecting transition T_{gr} to place P_m a multiplicity equal to the number of tokens in such a place.

A token in place P_{ob} represents the object being available. As soon as a failure occurs, a number of tokens equal to the number of failed disks is moved in place P_{od} by transitions T_{df} or T_{nf}. Such tokens enable the conflict between transitions t_{yes} and t_{no} representing the object being contained in the disks that failed or not, respectively. The probabilities associated to transitions t_{yes} and t_{no} (p_{yes} and p_{no}, respectively) depend on the system status and are given by the following functions:

$$p_{yes} = 1/(\#P_d + \#P_{od})$$

$$p_{no} = \begin{cases} 1, & \text{if } \#P_d = 0 \text{ AND } \#P_{un_1} = 1 \\ 1 - 1/(\#P_d + \#P_{od}), & \text{otherwise} \end{cases}$$

Transition t_{no} is also associated with a guard function ($[g_1]$) that prevents it to fire if the last disk failed:

$$[g_1] = \begin{cases} 0, & \text{if } \#P_d = 0 \text{ AND } \#P_{ob} = 1 \\ 1, & \text{otherwise} \end{cases}$$

If transition t_{no} fires, the object was not contained in the disks that failed and it is still available. If transitions t_{yes} fires, the object was contained in one of the disks that failed and the token in place P_{ob} is moved in place P_{un_1} modeling the object being unavailable. Transition T_{obr} represents the time necessary to recover the object from another Vision Cloud cluster where a replica of that object is present. It is associated with a rate equal to μ_{obr}. The recovery operation can be performed only when at least K disks are available (inhibitor arc from place P_{df} to transition T_{obr}). The token in place P_{ob} can also be moved in place P_{un_2} when the cluster file system is unmounted for a metadata destruction (transition t_f). A soon as the cluster file system is repaired, transition t_r fires and the object becomes available again.

5.1. Cluster file system failure probability. In order to properly set the model parameters (i.e., the probabilities associated to transitions t_{mf} and t_{um}) we need to know the probability that the cluster file system is unmounted when a new failure condition arises. Such a probability depends on the number of metadata replica as well as on the way the replica are distributed over the disks and the nodes. The problem can be formalized in the following way.

Let us start by defining a working condition where:

- n ($\leq N$) is the number of actual working nodes.
- d ($\leq D$) is the average number of working disks per node.

Indicating with MF the total number of metadata records, we are interested in the evaluation of the following probability:

- $P^{n,d,R,MF}(i)$ = Probability that, in the working condition defined by the pair (n, d), there is still one (out of the R) copy of each of the MF metadata files after i disk failures, given that the system was still working before the last failure

subjected to the following constrains:

1. Metadata are not restored.
2. If a node fails all its disk have to be considered failed, i.e., we have to consider the concurrent failure of d disks.
3. Replica are distributed so that, as long as there is a sufficient number of working nodes (i.e., $n \geq R$), two copies of the same file are not stored in the same node.

An estimation of MF can be obtained by considering the number of VISION Cloud objects actually stored in the cluster O and the number of metadata replica R as

$$MF = 1.1 \cdot O \cdot R \tag{5.1}$$

where the factor 1.1 refers to the assumption of a 10% overhead due to directory structure and VISION Cloud user and system metadata. The analytical solution of such a problem is intractable for large-scale systems [20], for this reason we solved the problem through simulation. We set up a simple simulator that starting from the values n, d, R, and MF creates a scenario by distributing metadata in an uniform way (still taking into account the constrains). Then, we iteratively introduce a failure (also in this case using an uniform distribution) until a distributed file system fault is encountered.

6. Results. The SRN model reported in Fig. 5.1 can be analytically solved by using ad-hoc tools (e.g., the SPNP tool [7]) thus allowing us to investigate the influence of system parameters on the desired performance indexes. Several powerful measures can be obtained. One interesting index is the availability A_{ob} of a generic object X formally defined as the probability that the object is fully accessible from external users at steady state. It can be obtained by computing the probability for place P_{ob} to contain one token:

$$A_{ob} = pr[\#P_{ob} = 1].$$

Similarly, the cluster availability A_{cl} (formally defined as the probability that the cluster file system is properly working at steady state) can be computed as the probability for place P_{on} to contain one token:

$$A_{cl} = pr[\#P_{on} = 1].$$

In this section, we present some preliminary results focusing on the object availability and taking into account only disk failures (i.e., considering fully reliable nodes). The relaxation of such an assumption, as well as the investigation of other performance indexes will be covered in future works.

System parameters have been set as follows. The number of nodes N has been fixed to 80 and the number of disks per node D has been fixed to 12, also considering the average fraction of occupied space in a disk c equal to 0.5. The disk mean time to failure (MTTF) $1/\lambda_{df}$ has been considered equal to 2 $years$ [18] while the mean time to repair (MTTR) $1/\mu_{dr}$ has been set to 48 h. Finally, the mean time to recover a metadata replica $1/\mu_{mr}$ has been set to 20 m. The mean time to recovery an entire disk from a remote cluster has been computed by assuming the disk dimension equal to 500 GB and by examining different scenarios with different level of bandwidth among the clusters in the same Vision Cloud. Three scenarios have been considered: HPC-like connectivity (10 Gb/sec bandwidth), high-speed WAN connectivity (100 Mb/sec bandwidth), and Internet-like connectivity (20 Mb/sec bandwidth). In the following, the three scenarios will be identified as *high*, *medium*, and *low* bandwidth scenario, respectively. Starting from the above reported assumptions, the values of the mean time to recover a disk $1/\mu_{obr}$ and the mean time to recover an entire cluster file-system $1/\mu_{gr}$ have been computed, as described in Section 4.

In the first experiment, we aim to investigate the influence of the metadata replication level R. First of all, in order to obtain the values of $P^{n,d,R,MF}(i)$ in all the working conditions, once the values for N and D have been chosen, we launched the simulator whit $n = 1, ..., N$ and $d = 1, ..., D$. The value of MF has been estimated[‡] through Eq. (5.1) by considering an average object size equal to $8MB$ (that can be considered a realistic example considering the presence of different kind of file, e.g, audio, photo, document, video files) and a corresponding number of object $O = \frac{500GB \cdot N \cdot D \cdot c}{8MB} = 30,720,000$. Data obtained for different value of R have been then collected in a file that has been used during the evaluation of the SRN model. Figure 6.1 shows the results obtained with $n = 80$, $d = 12$ and varying R from 3 to 5 (such values of R can be considered a good trade-off between redundancy and storage consumption). It can be observed that, as expected, the distributed file system failure probability increases when the number i of failed disks increases, reaching a value near to 1 when $i = 7, 22, 47$ with $R = 3, 4, 5$ respectively. Such a result highlights the influence of the replication level on the system fault tolerance. However in order to quantify the advantages obtained in terms of object availability, we solved the model using the $P^{n,d,R,MF}(i)$ values as input thus obtaining the data reported in Table 6.1. These data refer to the values of A_{ob} obtained in the three bandwidth scenarios. It can be observed that the influence of R strictly depends on the network bandwidth. In fact, in the low bandwidth scenario the object availability increases from a value of 0.95 to a value of 0.99 when R changes from 3 to 5, with a percentage gain of about 4%. On the contrary, in the high bandwidth scenario we obtain, in the same conditions, only a percentage gain of about 0.009%.

[‡]In the computation of the value of O the storage space occupied by metadata has been neglected.

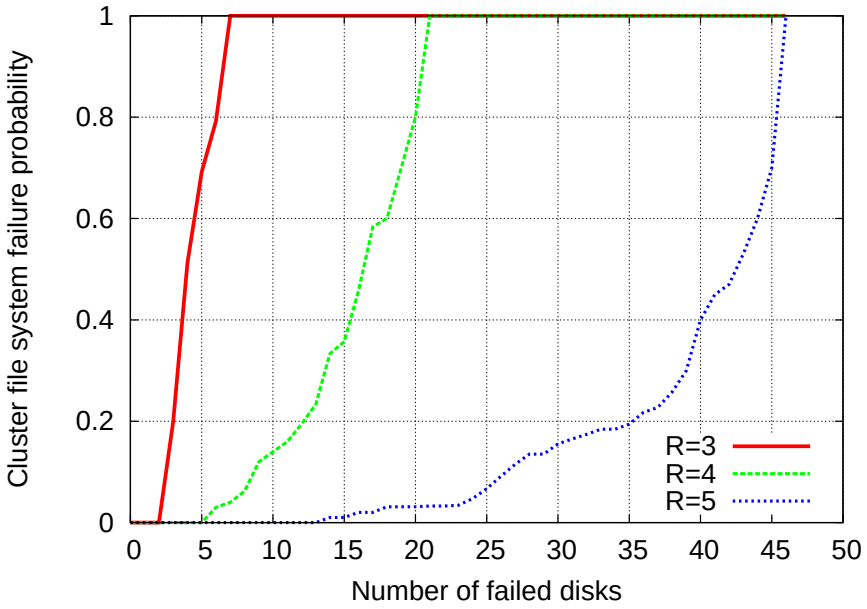

FIG. 6.1. *Cluster file system failure probability with respect to the number of disk failures* ($n = 80$, $d = 12$, $MF = 30,720,000 \cdot 1.1 \cdot R$).

	Low	Medium	High
$\boldsymbol{R = 3}$	0.9548245227	0.9906377221	0.9999077434
$\boldsymbol{R = 4}$	0.9983792830	0.9996754359	0.9999968294
$\boldsymbol{R = 5}$	0.9983795900	0.9996754973	0.9999968300

TABLE 6.1
Object availability A_{ob} varying R in three different bandwidth scenarios.

In the next experiment, we focus on the influence of the disk MTTF. Figure 6.2 shows the results obtained varying the value of the MTTF from 500 to 900 days in the medium bandwidth scenario. Such an analysis shows how increasing the disk MTTF it is possible to increment the overall object availability from a user perspective. This can be performed by choosing more reliable disks or exploiting RAID technologies with a consequent increase in the operating costs of the storage cloud infrastructure.

The above reported results give rise to interesting optimization problems. In fact, the Vision Cloud provider could take advantages of the proposed model in order to obtain useful insights during the design of a Cloud infrastructure. For example, given a certain level of desired availability and given the topological configuration of the clusters (in terms of network bandwidth), the provider can use the models to obtain the optimal number of replica to adopt and the needed disk reliability (in terms of MTTF). Similarly, per-user model-driven design could be conducted in order to optimize the placement of objects: according to the availability level requested by a single user the Cloud provider can understand in which clusters the user object replica have to be stored.

7. Conclusions. In the context of the VISION Cloud project reference architecture, we provided an SRN model for a storage cluster able to provide information about the reached availability level. Numerical results demonstrated the effectiveness of the proposed model. In fact, the model can be exploited for an assisted SLA management and a guided dimensioning of the VISION infrastructure. Future work will focus on extending the obtained results to the case of node failures and relaxing the simplifying hypothesis that we took into consideration in the present work, e.g., considering transient failures that can affect the overall object and cluster availability. Moreover, a high level methodology and a tool for the management of VISION Cloud

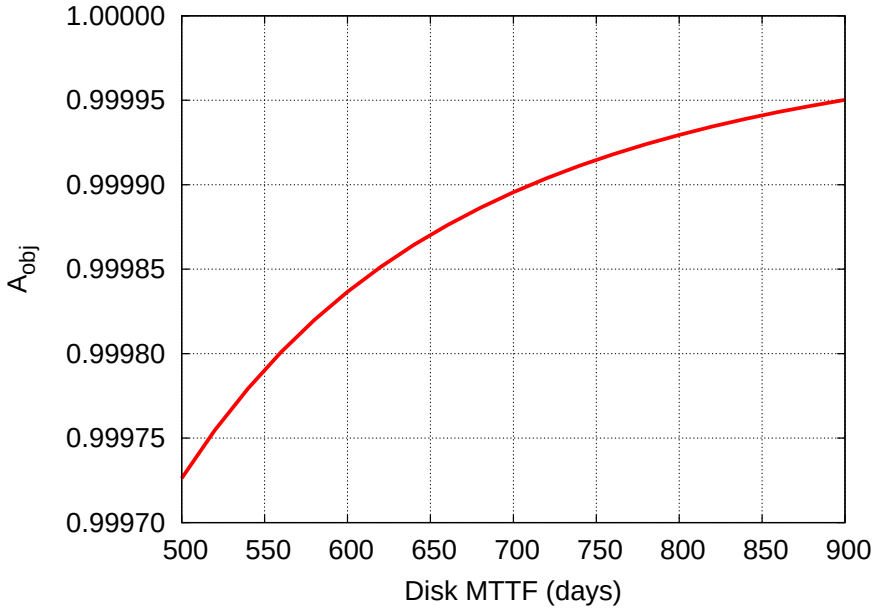

FIG. 6.2. *Object availability A_{ob} varying the disk MTTF in a medium bandwidth scenario with $R = 3$.*

storage infrastructures based on our model will be designed and implemented providing a powerful tool for both business and administrator choices. Finally, comparison of the obtained results against real world observation will be carried out in order to validate the model.

Acknowledgement. The research leading to these results has received funding from the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement number 257019.

REFERENCES

[1] *VISION Cloud Project, funded by the European Commission Seventh Framework Programme (FP7/2006-2013) under grant agreement n. 257019.* http://www.visioncloud.eu/.

[2] D. BRUNEO, *A stochastic model to investigate data center performance and qos in iaas cloud computing systems*, Parallel and Distributed Systems, IEEE Transactions on, PP (2013), pp. 1–10.

[3] D. BRUNEO, S. DISTEFANO, F. LONGO, A. PULIAFITO, AND M. SCARPA, *Workload-based software rejuvenation in cloud systems*, IEEE Transactions on Computers, 62 (2013), pp. 1072–1085.

[4] D. BRUNEO, M. FAZIO, F. LONGO, AND A. PULIAFITO, *Smart data centers for green clouds*, in Computer and Communications (ISCC), 2013 IEEE 18th International Symposium on, 2013, pp. 1–8.

[5] D. BRUNEO, M. SCARPA, AND A. PULIAFITO, *Performance evaluation of glite grids through gspns*, Parallel and Distributed Systems, IEEE Transactions on, 21 (2010), pp. 1611–1625.

[6] G. CIARDO, A. BLAKEMORE, P. F. CHIMENTO, J. K. MUPPALA, AND K. S. TRIVEDI, *Automated generation and analysis of Markov reward models using stochastic reward nets.*, IMA Volumes in Mathematics and its Applications: Linear Algebra, Markov Chains, and Queueing Models, 48 (1993), pp. 145–191.

[7] C. HIREL, B. TUFFIN, AND K. S. TRIVEDI, *SPNP: Stochastic Petri Nets. Version 6*, in International Conference on Computer Performance Evaluation: Modelling Techniques and Tools (TOOLS 2000), B. Haverkort, H. Bohnenkamp (eds.), Lecture Notes in Computer Science 1786, Springer Verlag, Schaumburg, IL, 2000, pp. 354 – 357.

[8] R. JAIN, P. SARKAR, AND D. SUBHRAVETI, *Gpfs-snc: An enterprise cluster file system for big data*, IBM Journal of Research and Development, 57 (2013), pp. 5:1–5:10.

[9] A. KHAN, X. YAN, S. TAO, AND N. ANEROUSIS, *Workload characterization and prediction in the cloud: A multiple time series approach*, in Network Operations and Management Symposium (NOMS), 2012 IEEE, 2012, pp. 1287–1294.

[10] E. KOLODNER, S. TAL, D. KYRIAZIS, D. NAOR, M. ALLALOUF, L. BONELLI, P. BRAND, A. ECKERT, E. ELMROTH, S. GOGOUVITIS, D. HARNIK, F. HERNANDEZ, M. JAEGER, E. LAKEW, J. LOPEZ, M. LORENZ, A. MESSINA, A. SHULMAN-PELEG, R. TALYANSKY, A. VOULODIMOS, AND Y. WOLFSTHAL, *A cloud environment for data-intensive storage services*, in Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on, 2011, pp. 357–366.

[11] E. K. Kolodner, A. Shulman-Peleg, D. Naor, P. Brand, M. Dao, A. Eckert, S. Gogouvitis, D. Harnik, M. Jaeger, D. Kyriazis, et al., *Data intensive storage services on clouds: Limitations, challenges and enablers*, European Research Activities in Cloud Computing, D. Petcu and JL Vazquez-Poletti, Eds. Cambridge Scholars Publishing, (2012), pp. 68–96.

[12] S. Krishnamurthy, W. Sanders, and M. Cukier, *A dynamic replica selection algorithm for tolerating timing faults*, in Dependable Systems and Networks, 2001. DSN 2001. International Conference on, 2001, pp. 107–116.

[13] ———, *Performance evaluation of a probabilistic replica selection algorithm*, in Object-Oriented Real-Time Dependable Systems, 2002. (WORDS 2002). Proceedings of the Seventh International Workshop on, 2002, pp. 119–127.

[14] M. A. Marsan, G. Balbo, and G. Conte, *A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems*, ACM Transactions on Computer Systems, 2 (1984), pp. 93–122.

[15] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, *A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing*, in Cloud Computing, vol. 34 of Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, ch. 9, pp. 115–131.

[16] C. Petri, *KommuniKation mit Automaten*, PhD thesis, University of Bonn. Germany, 1962.

[17] F. Schmuck and R. Haskin, *Gpfs: A shared-disk file system for large computing clusters*, in In Proceedings of the 2002 Conference on File and Storage Technologies (FAST, 2002, pp. 231–244.

[18] B. Schroeder and G. A. Gibson, *Disk failures in the real world: What does an mttf of 1,000,000 hours mean to you?*, in Proceedings of the 5th USENIX Conference on File and Storage Technologies, FAST '07, Berkeley, CA, USA, 2007, USENIX Association.

[19] V. Venkatesan, I. Iliadis, C. Fragouli, and R. Urbanke, *Reliability of clustered vs. declustered replica placement in data storage systems*, in Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS), 2011 IEEE 19th International Symposium on, 2011, pp. 307–317.

[20] V. Venkatesan, I. Iliadis, X.-Y. Hu, R. Haas, and C. Fragouli, *Effect of replica placement on the reliability of large-scale data storage systems*, in Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS), 2010 IEEE International Symposium on, 2010, pp. 79–88.

Edited by: Maria Fazio and Nik Bessis
Received: Nov 2, 2013
Accepted: Jan 10, 2014