BABEŞ-BOLYAI UNIVERSITY

FACULTY OF ECONOMICS AND BUSINESS ADMINISTRATION

DOCTORAL SCHOOL OF ECONOMICS AND BUSINESS ADMINISTRATION

DOCTORAL THESIS

-Summary-

Contributions to cloud computing service availability and data security

Scientific Advisor:

Prof. Nicolae Tomai, PhD

PhD Candidate:

Alexandru Butoi

Cluj-Napoca
2017

Summary contents

Keywords

Cloud computing, distributed systems, fault tolerance, fault trees, multi-agent systems, public cloud, data security, data confidentiality, secrete shared scheme, security protocol

Thesis contents

Publications list

1. List of publications related to the thesis contents

1. Alexandru Butoi, G. C. Silaghi, *Fault Tree - based service availability model in cloud environments. A Failure Trace Archive approach*, In Economics of Grids, Clouds, Systems, and Services, Lecture Notes in Computer Science, Editors : Altmann, Jorn, Vanmechelen, Kurt, Rana, Omer F., 2016, Springer International Publishing – presented at GECON 2016, 20-22 September 2016, Conference held at Harokopio University, Athens, Greece;

2. Alexandru Butoi, A. Stan, G. C. Silaghi, *Autonomous Management of Virtual Machine Failures in IaaS Using Fault Tree Analysis*, In Economics of Grids, Clouds, Systems, and Services, Lecture Notes in Computer Science, Editors : Altmann, Jorn, Vanmechelen, Kurt, Rana, Omer F., 2014, ISSN 978-3-319-14608-9, http://dx.doi.org/10.1007/978-3-319-14609-6_14 , Springer International Publishing – presented at GECON 2014, 16-18 September 2014, Conference held at Cardiff University, UK (2 citations according to Google Scholar at 02.06.2016);

3. Alexandru Butoi, A. Stan, G.C. Silaghi, *Reliable Management of Virtualized Resources Using Fault Trees*, Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2014 16th International Symposium on , pp.309-316, 22-25 Sept. 2014 doi: 10.1109/SYNASC.2014.49, http://dx.doi.org/10.1109/SYNASC.2014.49 IEEE Conference held at West University in Timisoara, Romania

4. Alexandru Butoi, N. Tomai, *Secret Sharing Scheme for Data Confidentiality Preserving in a Public-Private Hybrid Cloud Storage Approach*, Utility and Cloud Computing (UCC), 2014 IEEE/ACM 7th International Conference on , pp.992-997, 8-11 Dec. 2014, doi: 10.1109/UCC.2014.163, http://dx.doi.org/10.1109/UCC.2014.163 (3 citations according to Google Scholar 02.06.2016)

5. Alexandru Butoi, M. Moca, N. Tomai, *Data Confidentiality in Cloud Storage Protocol Based on Secret Sharing Scheme: A Brute Force Attack Evaluation*, In Trust Management IX, Vol. 454, 2015, ISBN: 978-3-319-18490-6, IFIP Advances in Information and Communication Technology, Editori Damsgaard Jensen, Christian and Marsh, Stephen and Dimitrakos, Theo and Murayama, Yuko, Springer International Publishing, pp. 177-184, http://dx.doi.org/10.1007/978-3-319-18491-3_13

6. Alexandru Butoi, G.A. Morar, A Ilea, *Two-Phased Protocol For Providing Data Confidentiality in Cloud Storage Environments* .In Proceedings of the BIS 2012 workshops, Lecture Notes in Business Information Systems, vol. 127, Springer, 2012, pp. 220-230, http://dx.doi.org/10.1007/978-3-642-34228-8_21

2. Other relevant publications

1. Alexandru Butoi, N. Tomai, L. Mocean, *Cloud-Based Mobile Learning*, Informatică Economică, vol. 17, no. 2/2013, pp. 27-40, 2013. ISSN 14531305/17.2.2013.03, http://dx.doi.org/10.12948/issn14531305/17.2.2013.03 (20 citations – Google Scholar la 02.06.2017)

2. Alexandru Butoi, G. C. Silaghi, *A Survey On Security And Legal Issues In Public Cloud Services,* - Proceedings of 13th International Conference on Informatics in Economy, Bucureşti, 2014

3. Nicolae Tomai, Alexandru Butoi, D. Mican, *E-Learning and M-Learning in Cloud*, - Proceedings of 13th International Conference on Informatics in Economy, Bucureşti, 2014

4. Alexandru Butoi, N. Tomai, D. Mican, G. C. Silaghi, *Designing Effective Web-Based M-Learning Systems,* in 12th International Conference on Informatics in Economy, ASE Bucureşti 2013, pp. 126-130, ISSN 2284-7472

Introduction

Cloud computing has changed and continues to change the way we do and think computation in any domains from scientific data processing and simulation to business related computing infrastructure, personal use or institutional use. In the very beginning of this new trend of cloud technologies, R. Buyya perceived cloud computing adoption as a leading factor to the creation of the 5th utility among existing ones like water, gas, electricity and telephony (Rajkumar Buyya, 2009). Nowadays, we are closer and closer to this vision, while telephony is automatically delivered with an Internet connection and sometimes with some associated cloud services like e-mail, storage or social networking. Still many challenges arise when bringing the cloud services adoption in the context of business, security and legal aspects. Adopting public cloud services at organizational level implies several questions and risks that need to be mitigated especially business continuity, data security and governance.

According to an IDCI survey among the IT executives and Chief Information Officers (Subashini & Kavitha, A survey on security issues in service delivery models of cloud computing, 2011), 74\% of subjects considered data security as the main barrier for cloud service adoption in business. Cloud service adoption to organizational or institutional level implies several security challenges. Mainly, it implies a transition from a security model based on a strong corporate data ownership to a data-centric security model where third parties gain various responsibilities in storing and managing data. Outsourcing the company's information system management to a third party e.g. the cloud provider, increases the data security risks (Dorey & A., 2011). Nowadays public clouds are recognized as an indubitable solution to address the rising cost problem of the constantly increasing data volumes that need to be stored and processed. In this entire equation, the lack of trust between cloud consumer and cloud provider represents the main drawback against public cloud adoption.

The question is how do we mitigate these aspects that generate mistrust between cloud providers and cloud adopters? Two possible and interrelated solutions arise from the literature study: Service-Level-Agreements (SLAs) and specific cloud attributes delivered "by-design". For example, if we position ourselves from an organizational perspective, organizations that rely on a cloud-based infrastructure the assurance of their business continuity is essential because their activities are dependent by the access to their cloud-stored data and cloud-deployed applications. In the case of a service shortage, their business activity will be affected, leading to financial implications. From a cloud provider perspective the business continuity assurance translates to the delivered cloud service and data availability, formally defined in SLAs as a commitment of the cloud provider to their consumer. It is also a responsibility of the cloud provider that they will keep their service and data availability to a certain minimal operable level agreed in SLAs. From a technical perspective, cloud technologies are seen as a stack of distributed hardware and software components, every component being prone to errors and failures. In this context, another question arises: How can a cloud provider commit to a SLA for these complex systems while their failure-prone factors and components cannot be fully controlled due to the probabilistic nature of the failures? A possible answer could be provided by putting in place some technical mechanisms that will assure appropriate and immediate intervention when these failures occur. The cloud provider could tackle the problem in an *availability-by-design* manner, by employing specific mechanisms that will ensure a certain service level of availability. These mechanisms could rely on replication, migration, load balancing or automatic backups. When comes to service availability, the challenge can be reduced to the manner of how these mechanisms should be combined and integrated in order to minimize the probability of service breach occurrence.

While availability of a service can be easily expressed by the percent of guaranteed service up-time from the total contracted service time, another challenge is the data security dimension generating the main concerns for the business cloud adopters. While cloud computing is a distributed computing stack, every technology, every component that is part of that stack has its own vulnerabilities and security risks. These components, if combined together results a more complex schemes of vulnerabilities and threats of the cloud stack. Moreover, the unpredictability of a security threat to strike is greater than for a failure, and the impact could be substantially greater, leading not only to data leakage and loss but also to service and data unavailability. In the case of availability and failure management the human factor can be minimized and substituted with automatic mechanisms. This is not the case for cloud data security, where the human factor represents one of the biggest threats, for example a malicious insider or a motivated attacker. On the same track, data security, privacy and data confidentiality concerns can be mitigated on SLA agreement contracts, while data privacy and confidentiality are to some extent subject to legal regulations. But still, cloud technologies are engineered to be mobile and location-transparent, meaning that data can easily cross the territorial and jurisdiction boundaries in its way to other data centers. The challenge in this context is how to specify SLAs and how to quantify the security level ensured by a provider to a customer? Again, the challenge could be tackled using on-design premises by employing the most suitable technical mechanisms to assure a provable level of data security risk as part of a security-by-design strategy. The security-by-design strategy consists in certain automatic security protocols and processes that secure data on-premises without any external trigger with the scope of minimizing the probability of unauthorized disclosure and altering. These mechanisms should always provide a provable and unconditioned level of security. In this way a cloud provider will be able to commit to certain security level SLAs by simply delivering services equipped with specific security mechanisms that are able to provide the requested data security level.

Particularly, our current work addresses these challenges of service availability and data security (especially data confidentiality) in a SLA-driven and unified manner, providing specific solutions as building blocks for a trusted-by-design and business-friendly cloud framework. The directions are mainly set by the dimension of business continuity requirement and data security with a strong emphasize on business data confidentiality in cloud storage systems. The relation between these two could be one of inclusion as the latest literature reviews consider service and data availability as a security matter when comes to critical business infrastructure.

Current work consists of two parts, one introducing a novel approach to service availability using Fault Tree Analysis and the other one tackling the problem of storing and protecting confidential data into public clouds, built using the same concepts borrowed from Fault Tree Analysis.

The main objective of the current work is to provide autonoumous and delivered-by-design approach to service availability and secure storage of high confidential data in public clouds using a unified SLA-driven manner. By "autonomous" approach we understand that the specific mechanisms are able to take decisions and trigger themselves without any external explicit intervention. By "delivered-by-design" we view our approach as integrated in the cloud framework and executed within standard cloud-specific operations and work-flows. By "unified manner" approach we consider service availability as a key attribute for a business cloud infrastructure along with data security attribute which need to be tackled together not separately as the mainstream literature tends to tackle it. By a "SLA-driven manner" we assure that the proposed models for service availability and data security provide fully quantifiable and provable SLA attributes that a cloud provider can commit in the case of implementation of the proposed models in their cloud system. For example if we consider a SLA agreement that includes a 99\% service availability commitment, applying the model for service availability should effectively provide a 99\% service availability. On the other side, if the SLA agreement includes a risk of 0.1\% for unauthorized data disclosure, applying our data security model to store data into the cloud should always provide a maximum probability of 0.001 for unwanted data disclosure. Both models are probabilistic approaches and we are using Fault Tree Analysis as the main theoretical toolbox for describing and implementing them.

The current research work uses concepts from fault tolerance domain, data security, cloud computing and is positioned at the intersection of three fields: fault tolerance applied to cloud services with the scope of availability assurance, the field of data security models, both with application in cloud computing field.

The overview of the thesis is as follows:

Chapter 1. Fault tree models applied in Computer Science presents a general overview of the Fault Tree analysis concepts with their applications in the field of computer science so far. It surveys some of the representative academic literature and describes the main applications of Fault Tree Analysis. It also overviews the fundamental concepts related to Fault Trees like minimal cut set and methods for computing these sets while identifying some challenges regarding the applicability of large trees in computing. The applicability of those is mainly limited by the state space explosion problem. Fault trees are the main instrument used in our effort for describing and implementing the models that comprise the contribution of current work.

Chapter 2. Service and Data Availability in cloud and distributed environments tackles the first track of the current work, and it overviews the existing literature, presenting the current concepts and relevant methods. The literature overview follows two main dimensions: one of a pragmatic approach to availability and one of a probabilistic approach to availability. Moreover, in the second part of the chapter we survey the most relevant approaches that tackle challenges related to cloud computing service availability. Both the academic literature point of view and the industry point of view are presented.

Chapter 3. Data and service security in cloud environments tackles the second track of the current work and provides a review of up-to-date security issues, challenges and solutions in cloud computing, followed in the second part of the chapter by a short overview of legal, non-technical and non-functional security requirements and implications of using public cloud services.

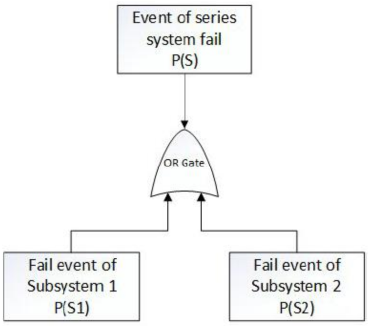
Chapter 4. Autonomous management of faults for cloud services presents the original contribution of this work included in the first track of the thesis - "service availability". Using autonomous fault tree analysis we introduce a fault agent model that is able to predict the future failure states of a service component and autonomously decide upon replication or migration operations in the scope of preserving the running state of the system and meet the SLA availability target. The approach is a probabilistic one, imposed by the usage of the fault tree analysis. In this chapter we describe our model applied for two scenarios: the independent running process and dependent running process. We describe the fault agent model that implements the fault tree, followed by the evaluation of it through empiric simulations, XEN based event traces and Failure Trace archive event traces used as input for the fault agent. The autonomous fault strategy consists in a distributed multi-agent and autonoumous fault tolerance protocol, offering passive and active replication techniques. We showed that mixing passive and active replication mechanisms in a probabilistic failure estimation approach, can significantly improve the resiliency of the system as compared to a standard passive replication benchmark without a considerable and systematic increase in resource usage.

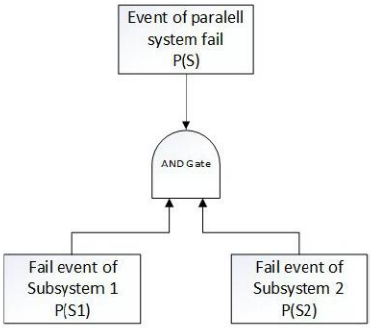
Chapter 5. Data confidentiality protection in cloud storage using fault trees tackles the original contribution of the current work on the security track by introducing a unique protocol for storing and protecting confidential business data into a private+public cloud storage setup. It does not require the classic crypto-mechanisms for protecting data from unauthorized disclosure.Our approach is based on a secret sharing scheme that has two main components: an algorithm for splitting the secret into shares and an algorithm for distributing the shares as efficient as possible from the security point of view. The secret is considered to be a file that contains confidential data which is split using two entropy-based algorithms. These algorithms will assure the splitting of shares in such manner that every share should carry minimum amount of information relative to the entire information carried by the secret. The secret sharing strategy is defined in a public+private cloud setup with multiple storage volumes at disposal, where the secret is distributed in a way that minimizes the probability of an attacker to reconstruct the secret without the distribution map that is always stored on the private cloud infrastructure. The considered attacker model is the "malicious insider". The proposed strategies for secret sharing in the cloud are in number of two: a probabilistic one and a fault tree based one - both looking to minimize the probability of unauthorized reconstruction of the secret. The model is evaluated using simulations through analysis of the probability evolution and through brute force setup simulations that provide an out-of-the box evaluation of the algorithms. Evaluation of both strategies showed that the probabilistic approach strategy provides better security level (smaller probability of data disclosure) for small files, while the fault tree based strategy provides better security level for large files. When compared with a random process of splitting and distributing files, our approach constantly minimizes the probability of reverse engineering and provides better and uniform resource utilization compared to the random process. Moreover the chunk distribution algorithms runs in polynomial time $O(n^3)$ while the splitting algorithm are of $O(n^2)$ complexity.

Conclusion and future work is the final chapter of the thesis and comprises the conclusions of our research effort, the findings, while providing an overview of the scientific contribution included with possible future work and enhancements.

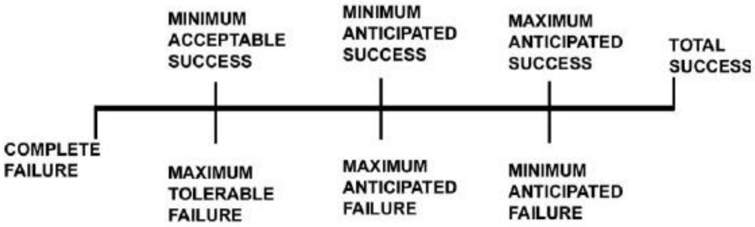
Chapter 1. Fault tree models applied in computer science

Fault tree analysis was first introduced in 1961 by H. A. Watson and later in 1975 was introduced by U.S. Nuclear Regulatory Commission as the main instrument used in their reactor safety studies. The same commission defines the fault-tree model as an analytical technique where the undesired state of the system and subsystems are specified and then the system is analysed in the context of the system's environment to asses possibilities in which the undesired event can occur \cite{haimes2005risk}. We applied the concepts of fault-tree analysis in the context of virtualized environments in a distributed and multi-agent approach. Every agent is capable to autonomously evaluate the health state of a virtual machine based on the events triggered in the environment (errors or other events) using fault trees. Fault trees can capture the reliable or unreliable state of one system. Basic fault tree analysis uses graphical tree representation of failure nodes connected together by gates (AND/ OR) resulting in new failure nodes. Every node is the equivalent of a subsystem and is characterized by the estimated probability of failure of the subsystem. Basically we use 2 simple concepts from Fault Trees: (1) the analysis procedure of the series system and (2) the analysis procedure of the paralel system :





Bill Vesely (Vesely, 2016), a NASA expert, defines Fault Tree Analysis as a "Systematic and Stylized Deductive Process" for assessing the fault risk using a fault tree which can illustrate the logical event relations in a chain of a undesired events. The same author presents some scenarios when Fault Tree Analysis(FTA) is used in safety analysis like: identifying the causes or weaknesses of system failures, safety and reliability design of a certain system, quantifying the probability of system failures.


MINIMUM ACCEPTABLE SUCCESS
MINIMUM ANTICIPATED SUCCESS
MAXIMUM ANTICIPATED SUCCESS
TOTAL SUCCESS
COMPLETE FAILURE
MAXIMUM TOLERABLE FAILURE
MAXIMUM ANTICIPATED FAILURE
MINIMUM ANTICIPATED FAILURE

We can also identify advantages and disadvantages in using FTA.

Identified advantages of FTA:

1. visual representation of cause-effect relatonships in chains of system failure events;

2. suitable for complex systems like aircrafts, space ships or complex distributed computers;

3. is a probabilistic model based on mathematics theory - scientific proven and applicable in physics, chemistry and engineering;

4. it allows to rigorously argument about the completness of a fault tree model;

5. it is tree-based representation, which can be easily implemented and applied in computer science;

6. it is versatile because it does not impose very specific maths for computing probabilistic indicators.

7. the probabilistic calculus can be easily adapted to the application domain by using the most appropriate mathematical methods that suits the best with the formalized phenomenon as we can see in the following paragraphs;

Identified disadvantages of FTA:

1. probabilistic model - it provides an estimation of the risk index;

2. incomplete or partial information - it takes into consideration only events that are relevant for the root failure event of the system;

3. the use of dynamic fault trees in complex system failure models can lead to node state explosion causing very large tree models and the usage of exponential distributions for modeling the event probabilities can decrease the modularization of the tree;

4. the tree can become very large and challenging to process when the system is very complex and is composed of many sub-systems;

5. incompleteness of information about fault events ca be a challenge in quantification of bottom fault events and probability estimation;

The main tracks for applying fault trees to computer science are:

1. Theoretical applications in computer science
2. Expert systems:
3. Software reliability studies
4. Information systems security

Chapter 2. Service availability and data security in cloud and distributed systems

The adoption of the 5th utility provides tremendous utility for the end consumers in a pay-per-use model subscription which allows immediate access to a cloud service (on-demand provisioning), immediate scaling of the resources as a response to the end-users' fast changing requirements and expectations (elasticity), efficient cost management and subscription (pay-per-use model), easy software licensing or pervasive access. On the other side, the 5th utility brings some new challenges too, like service availability, reliability or data security.

In our opinion, cloud computing services will reach to an adoption level where it will create dependency regarding the current activities, like any other utility. For example, if we consider an office building where every employee is using a computer to finish their usual tasks and a power shortage takes place in that building, the activity of the business will be severely affected while the employees cannot work on their tasks until the power will be reestablished. Now imagine a scenario in which the activity of the business is based on software and computation resources provided by a cloud provider. If the cloud services will be going down we will run in a similar situation of an entire or partial blocked activity. Moreover this faults cost money for the business and can lead to loss of data and work, and this is why, in our opinion, the availability and reliability of the cloud services are among the biggest challenges that the fifth utility adoption brings.

In the majority of cases, service availability discussions are positioned within the context of SLAs (Nabi, Toeroe, & Khendek, 2016) and we identified two main approaches of defining availability in cloud computing and distributed environments:

1. **The pragmatic approach** where availability is defined as a percentage metric expressing the total service running time from the total time in which the SLA was active. We consider that an SLA is active while it still has consumer subscribers. This approach is more used by the industry leading cloud providers like Google, Amazon or Microsoft in specifying their SLAs and is based on direct measurements and metric specifically designed for every platform or service they provide.
2. **The probabilistic approach** where availability is defined as the "probability that the system is operational when required" \cite{tl9000}. This approach is a more generic one and it can provide an estimation of the future delivered service availability. It can be expressed as a statistical estimation in form of a fraction between the Mean Time To Failure and the Mean Time Between Failures \cite[p.36]{bauer_book} \cite{toeroe_baum},

or it can be expressed mathematically using the probability theory specific concepts.

Nabi et. al. (Nabi, Toeroe, & Khendek, 2016) provide a comparison between how availability is defined by academic literature and well-known cloud providers like Amazon, Google or Microsoft. The academic literature use either the ratio between uptime and total time to quantify availability, including or excluding maintenance downtime intervals, while others have a probabilistic approach basing their availability definitions on SLA and expressing it as the "probability of providing service with respect to defined requirements". Our approach is based on the second one, using the probabilistic approach based on SLA availability requirements. Getting back to cloud provider's overview of availability quantification, the same paper \cite{Nabi201654} provides a short description of the availability SLA definition for Amazon, Microsoft and Google:

- Amazon's approach is monthly based, it differs in some aspects according to service type, and monthly availability is expressed as difference between 100\% and the percentage of service unavailability minutes per month. A commitment for guaranteed availability implies that the user to use at least two availability zone;
- in Google's approach, only downtime periods are counted, a downtime period is considered as five consecutive minutes of service unavailability and any downtime bellow five minutes is not taken into account;
- Microsoft's approach is more sophisticated by detailing the calculation of maximum of availability and unavailability time in a month to calculate the up-time percent. Generally the availability is the percent of the set time window in which the system is able to operate. The failure-prone components are also included in different update and fault domains: an update domain is defined as a set of servers in which are applied the same updates resulting different release versions. A fault domain consists in a set of servers or virtualized components that share the same resources. Deploying replicated virtual machines in different fault domains should lower the probability of total service unavailability due to power, hardware or network faults.

Chapter 3. Data and service security in cloud environments

The study of a system's information security starts with identifying the system's characteristics and behaviors followed by searching for the main challenges, risks or threats to which these systems are exposed. Cloud computing follows the same pattern but due to the fact that is composed by software and hardware stacks, the challenges and threats are present to every level of the cloud stack. Beside these, virtualization represents a key concept and technology in the context of cloud services which mainly assures the elastic provisioning of the resources, user-level isolation in some use cases of IaaS, and easy resource management, monitoring and delivery. The first definition and classification of cloud service delivery is introduced by NIST according to which Cloud computing is a "computing model for enabling ubiquitous, covenient, on demand network access to a shared pool of resources that can be rapidly provisioned and released with minimal effort or provider interaction". The definition indirectly raises some security challenges like ubiquity, network access, shared pool of resources or minimal provider interaction. The same document identifies some essential characteristics of the cloud services (on-demand self service, broad network access, resource pooling, rapid elasticity, measured service), three models of delivery(Software as a Service - SaaS, Platform as a Service - PaaS and Infrastructure as a Service - IaaS) and the four well known models of cloud deployment(private, public, community and hybrid cloud). Security in the technological context of cloud computing represents a complex issue and context-specific, every service type characteristic and delivery manner is adding new gaps and challenges from the data security point of view. Khorshed et. al. (Khorshed, Ali, & Wasimi, 2012) identified several challenges and gaps for cloud computing and classified them using a Support Vector Machine learning strategy as the best method for classifying threats in cloud computing. According to their study, the main challenges for security point of view are:

- the new features that cloud computing introduced: virtualization, multi-tenant environment, on-demand, "as a service" delivery;
- the new attributes of the computing model: autonomic, globalization, self management, distributed infrastructure;
- the new service models: SaaS, PaaS, IaaS and many more arising on the market;
- deployment models of the cloud services;
- the role definitions of the interacting parties: cloud provider, cloud broker, vendors and cloud consumer or cloud user;
- locality and location transparency of the cloud services: quality of service, SLAs and legal issues;

The gaps identified are:

- trust between parties due to minimal interactions, autonomic manner and all the externalization processes involved;
- security threats, their complexity and diversity;
- hardly quantifiable risks of technical and business nature;

We can identify here a "critical trio" formed by three key elements referring to data security and availability: **data security** "directly affects" **data availability** "directly affects" **service availability** with serious implications in the business continuity of the organizations that rely on these cloud-based services. For example, a data corruption or unauthorized deletion due to a security incident may induce unwanted behavior at the service level which uses the affected data storage resource. Further more the business processes using the affected service are delayed, restoration and investigation effort is needed as a response for the incident, implying service down-times during restore procedures and further implying human resources to be allocated to investigate and to provide a resolution for the problem..

Chapter 4. Autonoumos management of faults for cloud services

We elaborated a fault tolerance algorithmic strategy consisting in a distributed multi-agent and autonoumous fault tolerance protocol, offering passive and active replication techniques. We showed that mixing passive and active replication mechanisms in a probabilistic failure estimation approach, can significantly improve the resiliency of the system as compared to a standard passive replication benchmark without a considerable and systematic increase in resource usage. The Fault Tree based model consists in a novel approach to autonomous management of virtual machine faults suitable in both IaaS and SaaS by designing a fault agent which can reside in virtual machines or generally on service instances. Using fault tree analysis it can decide whether the service instance is reliable or not in the scope of avoiding QoS policy breach. The model uses a strategy which combines the fault tolerance achieved trough active replication and passive replication: when replication strategy fails too, the migration process of service instance is used in order to avoid general faults. We evaluated the model using a practical approach by generating virtual machine events traces from production Xen engine logs followed by using Failure Trace Archive datasets. We showed that after an error event is raised, several consecutive non-error events can have the power to reestablish the reliable status of the service instance. The presented results are for the LANL, failure traces but the model has been successfully applied on P2P, Grid, Web, DNS archives provided by the FTA, results which are also briefly depicted in appendices. The main advantage of the presented fault avoidance strategy at the service level is that the third replica is created only on demand and the required resources are not used from the start, resulting in a more efficient mechanism for resource allocation, while having the capability to provide higher service availability. The main disadvantage of this strategy would be timing: in order for the migration process to be successful, the migrated instance should be in a healthy state, implying the usage of a probabilistic provisioning system for predicting the future fault state of an instance and to be able to start migration process before the migrated instance is faulty.

Chapter 5. Data confidentiality protection in cloud storage using fault trees

The chapter presents a secret sharing scheme applicable in a hybrid private-public cloud for secure storage of data in public storage cloud services. We tackled two main challenges:

(1) splitting the ”secret file” into shares using entropy and relative entropy as metrics for security-driven optimality identification, and we proposed two versions of the same algorithm for splitting the file into chunks that have minimum informational content

relative to the entire file;

(2)propose two strategies of storing the data into the cloud that minimizes the risk of unauthorized data disclosure with an optimum usage of the available storage resources.

Both versions of the file splitting algorithms are using the same metric indicators, the Information Entropy and the Kullbach-Leibler entropy to find optimal file chunks that will represent the secret shares to be distributed among sharers. The first version of the file splitting algorithm uses a well defined and fixed search space, while the second one uses a variable search space, both versions being sub-optimal solutions of the same problem.

The second version is an enhanced one because it provides better outcome compared to the first version with limited search space.

We showed that the two splitting algorithm runs in a polynomial time of $O(n^2)$ and provide better and superior results compared to a random process of file splitting: better K-L values with better chunk sizes and lower number of chunks per file.

The distribution of the secret shares(file chunks) is tackled using a probabilistic strategy based on Bernoulli Trials computation and a Fault Tree based strategy respectively. The last one employs two fault tree models and merges them in a manner that will minimize the general fault probability. Evaluation of both strategies showed that the probabilistic approach strategy provides better security level (smaller probability of data disclosure) schemes that have less than 1000 shares (small file sizes), while the fault tree based strategy provides better security level for schemes that have more than 1000 shares (large file sizes). The modeling and evaluation of the secret shared scheme model was conducted considering a malicious insider opponent which in our opinion, is the most dangerous and severe threat for data security and confidentiality in cloud.

We showed that our distribution approach provides better results when compared with a random process for distributing files. It constantly minimizes the probability of reverse engineering a file and provides better and uniform resource utilization compared to the random process. Moreover the chunk distribution algorithms runs in polynomial time $\theta(n^3)$. There are no limitations regarding the number of chunks or the number of volumes that can be used for chunk distribution. Considering the complexity of the algorithms the running time will increase as number of chunks and volumes increase.

We can find applicability of our work in storing cloud medical data, business data or institutional data using a private-public cloud setup.

Chapter 6. Conclusions and future work

Our research effort was concentrated on two main tracks: availability - chapter 2 and 4 and data security - chapter 3 and 5. Chapter one was concentrated in introducing the main toolbox -of Fault Tree Analysis, used for tackling our challenges. Fault Tree Analysis is a powerful instrument in analyzing and estimating the failure risk of a complex system where several states combinations can appear. In the recent years, with the introduction of distributed and cloud computing technologies, Fault Tree Analysis was not yet extensively used to solve complex problems of system failures or security risk assessments. Isolated approaches start to arise in the last 2-3 years in the academic literature. One of the main original contribution of our work is that using automated Fault Tree Analysis we introduce an autonomous failure management system model, suitable for cloud and distributed computing that aims to avoid QoS breaches committed in the SLA. Also we introduce a security model based on a secret sharing scheme in which the secrets are shared using a Fault Tree - based algorithm that minimizes the probability of security breach. Moreover, on the security side we solve the problem of secret sharing both using a pure probabilistic traditional approach and using a novel approach based on fault trees.

The relation between availability track and the security track is one of inclusion, while latest literature review indicates that availability of critical infrastructure can be seen as a matter of security. This fact is driven by the business continuity requirement raised by the customers that have contracted cloud services mainly for business use. From the early stages of public cloud advertisement and introduction, the challenge of trust between the cloud provider and cloud consumer has been raised, especially when it comes to moving private and confidential information into the cloud. This type of trust was also seen as the main drawback for businesses to adopt cloud services on a large scale. The proposed solution was a better transparency between cloud provider and consumer regarding the service delivery operations and commitments. In our opinion nowadays, this transparency requirement is tackled mainly through audits and less through SLAs and bilateral commitments. Following this idea, our availability and confidentiality mechanisms provide an SLA-driven approach through "on-design premises of cloud attributes for security and availability" as follows:

1. the fault model uses the committed availability percentage in the SLA as a reference metric and employs autonomous strategies of replication and migration in a multi-agent manner, to assure the system is meeting the availability SLA requirement;

2. the data confidentiality model provides "security-by-design" mechanisms that if employed, will provide a certain provable and quantifiable level of security risk that can be guaranteed in the SLAs for certain setups and scenarios;

The main contributions of our research effort are as follows:

I. We introduce an autonomous fault tolerance model using basic Fault Tree Analysis suitable for cloud and distributed environments,using as input system-specific event logs and based on a simple Fault Tree that predicts possible failures, autonomously takes appropriate decisions in order to preserve the service delivery in the SLA-specified availability terms. We tackle the problem in a multi-agent setup with limited resource pools, while extending the CloudSim simulation environment to enable Fault Tree Analysis for clouds and to evaluate our model. The fault agent model uses as main strategy the live migration of the service when the strategy of replication presents a high probability to fail. The fault model has been evaluated in the first phase using fault events generated from an empirical Poison distribution, then we use as input the event log traces from a XEN hypervisor, continuing with the evaluation of the model based on LANL failure trace dataset provided by the Failure Trace Archive. Experiments have proven that our strategy is suitable for assuring availability in cloud and distributed environments.

II. We introduce a data confidentiality protection model built on the premises of "security-by-design" applicable in a private-public cloud data storage setup. The model consists in a secret sharing scheme strategy:

   a. the secret splitting strategy is accomplished by two versions of the same algorithm both based on Shannon Fano entropy and Kullback-Leibler entropy. It aims to split the secret(file) in shares in a manner that every share(chunk) to carry minimum information relative to the entire information carried by the secret. The solutions provided by these algorithms are sub-optimal, one version of the splitting algorithm uses fixed-limit search space strategy, while the second version of the splitting algorithm uses variable search space strategy which provides better results;

   b. the secret share distribution consists in distributing the secret shares into the cloud storage volumes in a manner that will minimize the probability of an attacker to reconstruct the secret and gain unauthorized access to the confidential data carried by the secret. We propose two strategies for achieving this: the first one is a pure probabilistic approach which minimizes the probability of secret disclosure based on Bernoulli Trials computation and the second strategy uses fault trees to model the possible attack behavior and compute the best shares distribution that provides minimum disclosure risk. The four algorithms comprise the secret sharing scheme that is evaluated through simulations and we show that the probability of unauthorized disclosure is constantly minimized. We also evaluated the algorithms outcomes through independent proxies like Levenshtein distance in evaluation of the secret splitting algorithms and brute-force attacks simulation for the distribution algorithms evaluation. We showed that the two splitting algorithm runs in a polynomial time of \$$O(n^2)$ and provide better and superior results compared to a random process of file splitting: better K-L values with better chunk sizes and lower number of chunks per file. We also showed that our distribution approach provides better results when compared with a random process for distributing files. It constantly minimizes the probability of reverse engineering a file and provides better and uniform resource utilization compared to the random process. Moreover the chunk distribution algorithms runs in polynomial time $O(n^3)$.

Other relevant contributions to Fault Tree Analysis:

1. FTA is mainly used as an analytic tool for evaluating systems in terms of safety and reliability, while we enable FTA for autonomous reasoning and decision making, extending the applicability of Fault Trees in Computer Science;
2. in our approach to FTA, the state space is a dynamic one and abstracted in form of probability of system failure inducement - the approach was imposed by the existing large number of events in complex distributed system which otherwise would lead to the state space explosion issue;
3. while classic FTA approaches takes into consideration only the "negative" states in a system - the states that describe an event of failure, in our approach, we consider not only event states that describe a failure in the system, but those event states that describe a non-failure event or a recovery event too; this is one of the aspect, in our opinion, that enables the fault tree models to be used for autonomous decision making. Moreover we do not use the method of determining the minimal cut set which can be computationally costly because the introduced model already describes a minimal cut set for a general failure in a certain setup from a cloud environment;

Other relevant contributions:

1. we use FTA implemented in a multi-agent setup for real time prediction and monitoring of system reliability - a probabilistic approach in decision making for QoS breach in cloud environments;
2. we survey FTA and Fuzzy FTA literature to provide a comprehensive and comparative overview of the applicability of these concepts with their strengths and drawbacks;

3. we overview the literature related to service availability in cloud and distributed computing following two tracks of pragmatic approach to availability along with the probabilistic one.

4. we provide a prototype for building a FTA-based versatile and inter-operable fault management system, suitable for different distributed systems implementing a hybrid solution of pro-active-reactive technique in failure mitigation;

5. we survey latest literature in data security in relation with availability. Availability has two dimensions here: a purely functional - expressed as a SLA metric for quality of service delivery and a security dimension while unavailability of services may directly affect business continuity;

6. we introduce a utility-based, SLA-driven secret sharing model that on-premises protects confidential data stored in public clouds and provide certain quantifiable and provable levels of security in different setups. The probability of unauthorized data disclosure that the secret sharing scheme is based on, can be seen as a metric for quantifying the security risk level associated to the protected data in different scenarios. From a cloud provider point of view, which employs our secret sharing scheme to protect data, these levels of security can be used for defining and committing to data confidentiality protection SLAs.

Future research directions could represent:

1. a comprehensive overview of the performance overhead that the presented solutions adds on top of an existing infrastructure implementing those models;

2. defining an associated cost model that can be associated with the implementation of these models in cloud computing infrastructures - the cost of security and the cost of availability in a pay-as-you-go fashion;

3. study and enhance the availability and data security models in a real world setup of distributed clouds or server farms;

Bibliography of the summary

1. Ortmeier , F., & Schellhorn, G. (2007). Formal fault tree analysis - practical experiences. *Electronic Notes in Theoretical Computer Science*, 139-151.

2. Batista, G. B., & et., a. (2017). A qos-driven approach for cloud computing addressing attributes of performance and security. *Future Generation Computer Systems*, 260-274.

3. Bauer, E., & Adams, R. (2012). *Service Reliability and Service Availability.* Wiley - IEEE Press.

4. Broke, P., & Paige, R. (2003). Fault trees for security system design and analysis. *Computers and Security*, 256 - 264.

5. Cha, S., & Yoo, J. (2012). A safety-focused verification using fault trees. *Future Generation Computer Systems*, 1272 - 1282.

6. Chang, K.-H., & Cheng, C.-H. (2009). novel general approach to evaluating the {PCBA} for components with different membership function. *Applied Soft Computing*, 1044 - 1056.

7. Cheraghlou, M., & et. al. (2016). A survey of fault tolerance architecture in cloud computing. *Journal of Network and Computer Applications*, 81 - 92.

8. Codetta-Raiteri, D. (2005). The conversion of dynamic fault trees to stochastic petri nets, as a case of graph transformation. *Electronic Notes in Theoretical Computer Science*, 45-60.

9. Dorey, P., & A., L. (2011). *Cloud computing a security problem or solution?*

10. Fielder, K., & Smith, C. (2016, Mai 25). *Security as a service working group.* Retrieved from https://cloudsecurityalliance.org/group/security-as-a-service/

11. Fourneau, J.-M., & Pekergin, N. (2016). Dynamic fault trees with rejuvenation: Numerical. *Electronic Notes in Theoretical Computer Science*, 27 - 47.

12. Grance, T., & Mell, P. (2014, February 21). *The nist definition of cloud computing.* Retrieved from NIST: http://dx.doi.org/10.6028/NIST.SP.800-145

13. Grunske, L., & Joyce, D. (2008). Quantitative risk-based security prediction for component-based systems with explicitly modeled attack profiles. *Journal of Systems and Software*, 1327 - 1345.

14. Haymes Y. (2005). In Haymes Y., *Risk Modeling, Assessment, and Management.* Wiley Series in Systems Engineering and Management.

15. Kabir, S., Walker, M., & Papadopoulos, Y. (2015). Quantitative evaluation of pandora temporal fault trees via petri nets. *9th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*, (pp. 20-37). Paris.

16. Kabir, S., Walker, M., & Papadopoulos, Y. (2016). Fuzzy temporal fault tree analysis of dynamic systems. *International Journal of Approximate*, 20-37.

17. Khorshed, T., Ali, S., & Wasimi, S. (2012). A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing. *Future Generation Computer Systems*, 833 - 851.

18. Labib, A., & Read, M. (2015). A hybrid model for learning from failures: The hurricane Katrina disaster. *Expert Systems with Applications*, 7869 - 7881.

19. Li, J. Z., Lu, Q., & et. al. (2013). Improving availability of cloud-based applications through deployment choices. *Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference*, (pp. 43-50).

20. Mishra, P., & et. al. (2016). Intrusion detection techniques in cloud environment: A survey. *Journal of Network and Computer*, 18 - 47.

21. Nabi, M., Toeroe, M., & Khendek, F. (2016). Availability in the cloud: State of the art. *Journal of Network and Computer Applications*, 54-67.

22. Rajkumar Buyya, C. S. (2009). Cloud computing and emerging fITg platforms: Vision, hype, and reality for delivering. *Future Generation Computer Systems*, 599 - 616.

23. Rushdi , A., & Ba-Rukab, O. (2005). Fault-tree modelling of computer system security. *International Journal of Computer Mathematics*, 806-809.

24. Singh, S., & et. al. (2016). A survey on cloud computing security: Issues, threats, and solutions. *Journal of Network and Computer Applications*, 200 - 222.

25. Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud. *Journal of Network and Computer Applications*, 1-11.

26. Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, 1-11.

27. Tekinerdogan, B., Sozer, H., & Aksit, M. (2008). Software architecture reliability analysis using failure scenarios. *Journal of Systems and Software*, 558 - 575.

28. Toeroe, M., & Tam, F. (2012). *Service availability: principles and practice.* John Wiley & Sons.

29. Tu, M., Xu, D., Xia, Z., & et. al. (2011). Reach availability modeling of replicated services. *Computer Software and Applications Conference (COMPSAC)* (pp. 688 - 693). IEEE.

30. Undheim, A., Chilwan, A., & Heegaard, P. (2011). Differentiated availability in cloud computing SLAs. *Grid Computing (GRID), 2011 12th IEEE/ACM International Conference* (pp. 126-136). IEEE/ACM.

31. Vesely, B. (2016, February 21). *Fault tree analysis (fta): Concepts and applications*. Retrieved from https://www.hq.nasa.gov/office/codeq/risk/docs/ftacourse.pdf