Risk-driven Proactive Fault-tolerant Operation of IaaS Providers

Jordi Guitart*[†], Mario Macias*[†], Karim Djemame[‡], Tom Kirkham [‡], Ming Jiang [‡] and Django Armstrong[‡]
*Barcelona Supercomputing Center and [†]Universitat Politecnica de Catalunya - Barcelona Tech
Jordi Girona 31, 08034 Barcelona, Spain
{jguitart, mario}@ac.upc.edu
[‡]School of Computing, University of Leeds, Leeds LS2 9JT, UK
{K.Djemame, T.Kirkham, M.Jiang, een4dja}@leeds.ac.uk

***Abstract*—In order to improve service execution in Clouds, the management of Cloud Infrastructure has to take measures to adhere to Service Level Agreements and Business Level Objectives, from the application layer through to how services are supported at the lowest hardware levels. In this paper a risk model methodology and holistic management approach is developed specific to the operation of the Cloud Infrastructure Provider and is applied through improvements to SLA fault tolerance in Cloud Infrastructure. Risk assessments are used to analyse execution specific data from the Cloud Infrastructure and linked to a business driven holistic management component that is part of a Cloud Manager. Initial results show improved eco-efficiency, virtual machine availability and reductions in SLA failure across the whole cloud infrastructure by applying our combined risk based fault tolerance approach.**

I. Introduction

Fault tolerance mechanisms in Cloud Infrastructures enhance the well-known reactive mechanisms that allow recovering the execution of services upon infrastructure failures with proactive capabilities. In our model we link the proactive capabilities to profit that can be gained by the application of risk assessment tools to apply preventive measures before the actual failure occurs. We propose proactive management for IP providers in order to deal with Virtual Machine (VM) and host failures before they actually happen. If these failures take place before we can pro-actively avoid them, our approach reverts to the reactive recovery mechanism.

In any case, our fault tolerance mechanism suggests recovery actions (reactively and pro-actively), which are evaluated by the VM manager with respect to the overall IP objectives. Although carrying out these actions will surely improve the availability and reliability of the provider, it could happen that their impact on other metrics is not compliant with the provider's goals. For instance, some recovery actions can increase the cost or reduce the ecoefficiency of the operation. For this reason, suggested recovery actions are evaluated against the provider's global objectives and carried out only if they contribute to accomplish these goals.We demonstrate this by analysing the impact of risk based proactive management of the Cloud Infrastructure in relation to the eco-efficiency of the Cloud.

Business objectives that dictate the proactive action based on the level of risk are also derived from the risk management approach. Risk in clouds spans the need to support various parties involved in making informed decisions regarding contractual agreements. Consider an infrastructure provider that wishes to offer use of its resources as a pay-per-use service. Interactions between the provider and an end-user (a service consumer or a broker acting on their behalf) can then be governed through a Service Level Agreement (SLA), contractually defining the infrastructure provider's obligations, the price the end-user must pay and the penalty the provider needs to pay in the event that it fails to fulfil its obligations. Consequently a provider may be unwilling to implement such an approach without effective risk assessment that allows to analyze the impact of such failures before they actually occur [1].

Therefore, there is a clear need to offer tools to efficiently manage the full life cycle of Cloud services. These tools will provide for simplified construction of services, and for making informed deployment and runtime management decisions based on risk assessment models for evaluation of providers and will permit the appropriate establishment of fault tolerance mechanisms [2].

The paper presents two main contributions: 1) a novel approach to fault tolerance and how this can be combined with risk. Experiments in the paper reveal how VM availability and eco-efficiency can be improved using risk; and 2) the development of the Risk model and its application on a Cloud testbed and the results of the assessments and subsequent proactive management based on real data. The paper focuses on the risk assessment tools an Infrastructure Provider makes use of during service operation and additionally, the fault-tolerance mechanisms put in place for an optimized service management.

The rest of the paper is as follows. Section II presents the related work. Section III introduces the risk models used in this work. Section IV describes our approach for fault-tolerant operation of IPs. Section V presents the evaluation results. Finally, Section VI concludes the paper.

II. Related Work

Risk as a basis for proactive or reactive service management in Clouds can be seen within the domain of information security / privacy [3]. Information security is suited to the management of risk in clouds as risk can be defined and linked to existing ways of expressing security policy [4]. In these cases an organisation or user can associate events expressed in policy which can be measured using risk assessment, proactive and reactive action can then added to the process to act upon the risk [5]. Risk as a management concept has a significant background in the concept of systems auditing and third party insight into systems [6].

Risk from a third party service as an extension to risk assessment mechanisms has also been explored in cloud environments [7]. In our proposed model we monitor risk in terms of computing resource behaviour within the domain of the Cloud provider and present interfaces for third party auditing and control. Our focus is with respect to internal threats to service execution based on analysis of historic and current data from the infrastructure.

Risk assessment in distributed computing has been researched in various related projects. These projects have included objectives ranging from information protection, evaluation /prediction of QoS and probability of SLA failures [8], [9], [10]. In terms of wider resource failure a wide range of studies exist for distributed computing environments [11], [12].

Several works aim for fault tolerance in computing systems. Remus [13] provides completely transparent recovery from fail-stop failures of a single physical host. The authors encapsulate protected software in a virtual machine, which is replicated asynchronously. This allows the virtual machine to continue executing speculatively. Remus uses timeouts to detect failures, so this is a completely reactive approach. Although it uses virtual machines, Remus does not consider individual VM failures, only physical host failures.

Jung et al. [14] propose a component placement and resource allocation scheme that, in reaction to hardware failures, regenerates the affected software components and reconfigures the set of applications on the resource pool, by either migrating the VM to another host or changing the CPU share allocated to the VM on its current host, to maintain a user-defined availability level while minimizing response time degradation. Again, this approach is reactive and considers only hardware host failures.

Fu [15] presents a self-reconfiguration system for HPC virtualized environments and proposes a set of strategies to select the nodes to execute the tasks. In addition to performance states of candidate nodes, these strategies also consider their reliability status, which is estimated by forecasting when the next failure will occur in that node. Fu's approach is proactive as he proposes migrating a VM when the deadline of task is after the predicted occurrence time of the next failure in its running node. However, this work assumes at most one failure that might occur in the lifetime of a job task. In addition, it is oriented only to node failures.

Nagarajan et al. [16] promote a proactive approach where processes automatically migrate from unhealthy nodes to healthy ones. They monitor the health of physical nodes via periodic sampling of values from the given set of sensors and compare them with threshold values. In case any of the thresholds is exceeded, a target node is selected to migrate the guest VM to. In this aspect, this work is similar to our approach. However, it only considers host failures. In addition, their threshold values refer to low-level parameters such as the safe temperature range for the CPU, which is far from the knowledge of a common user. Instead of this, we deal with risk level categories.

Alonso et al. [17] present a framework which provides a transparent and predictive rejuvenation solution to web services on virtualized platform. They use Machine Learning to estimate the time to crash of a service due to software aging. When this time is below a predefined threshold, software rejuvenation for the service is triggered. This is coordinated with the reconfiguration of the platform to maximize the number of services running simultaneously while ensuring that every service deployed has the resources requested by the service owner. Although this is a proactive approach, it only considers software aging failures, not physical node failures.

Notably, none of the above works proposes a holistic solution including complementary proactive-reactive approaches to tolerate both physical and VM failures, as we do in this paper. In order to influence risk assessment along any business / third party grounds we have to take the management and the setting of the risk assessment outside of the Cloud Fabric into a service level [18]. In the OPTIMIS project we have developed the Holistic Management service in order to achieve this [2]. This service links directly to the OPTIMIS cloud toolkit and components including Cloud Optimisation and Cloud Risk Assessment services. Holistic Management ensures proactive and reactive risk management linked to higher / service level goals.

III. Risk Models

A. Elements of Risk

In order to assess the risk associated with the Cloud resources based on (real-time) data provided by the Cloud Monitoring Infrastructure (CMI), it is essential to know what data is required by the risk assessor, and how it is going to be analysed to estimate the actual risk. For this purpose, the *risk inventory* is populated with:

Assets: Virtual Machine (VM), physical host, Service Level Agreement (SLA) with a description of their characteristics. Risk events will be assessed in terms of these.

Incidents / Risk Scenarios: aim to describe any event, condition or their combination that has the potential to reduce the capacity or availability of an asset. They are composed of:

Vulnerabilities: describe inherent weaknesses of the asset (e.g. a faulty hardware) and their impact reflects the possibility of a risk incident, e.g. violations of the Quality of Service (QoS), and SLA indicators, inherent to the assets.

Threats: represent the other side of the risk which depends on factors independent to the asset, e.g. loss of connectivity of a physical host.

Adaptive capacity: description of the mitigating strategies in place for the specific asset, e.g. server replication

Impact/Consequence of a risk incident, e.g. failure of a physical host, and is defined using as degraded performance, loss of data, or unavailability. The evaluation is performed according to the indicators selected to describe the asset as well as associated costs, e.g. of not meeting predefined service levels.

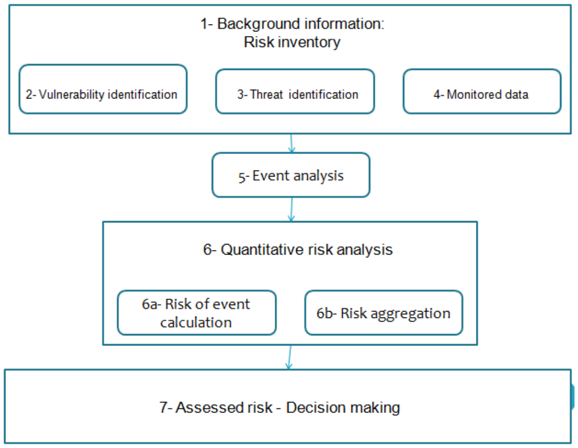
1- Background information:
Risk inventory
2- Vulnerability identification
3- Threat identification
4- Monitored data
5- Event analysis
6- Quantitative risk analysis
6a- Risk of event calculation
6b- Risk aggregation
7- Assessed risk - Decision making

Fig. 1: Generic Risk Assessment Model

B. Process

A quantitative risk assessment approach is then applied to estimate the level of risk attached to VMs, Physical Hosts, and SLAS thanks to the data gathered by the CMI. Therefore, an identification of the elements of risk in the risk inventory for VMs, Physical Hosts, and SLAS becomes important. It should be noted, that the nature of risks may differ thus, the quantitative risk estimation too.

Figure 1 shows the generic risk assessment model, which divides the risk assessment process into the following stages:

Risk Inventory. At this stage, requirements analysis is performed to identify how the risk inventory is populated.

Vulnerability identification. A vulnerability is considered as a weakness or flaw in system procedures, design or internal, management controls that can be accidentally triggered or intentionally exploited. Let each vulnerability be represented as a single bit in the vulnerability vector:

$\vec{V}$ = $\{V_i\}$ = 1,0 $\forall i$, i=1,2, . . . n

where V_i represents an individual vulnerability. The value 1 indicates the presence of this vulnerability in the system under assessment, otherwise 0.

Threat identification. During a threat analysis process potential threat sources and actions that may exploit system vulnerabilities are identified. Information about threats can be gathered from experts, the Cloud provider's historical database or log files. Let each threat be represented as a single bit in the threat vector:

$\vec{T}$ = $\{T_j\}$ = 1,0 $\forall j$, i=1,2, . . . m

where T_j represents an individual threat. The value 1 indicates the presence of this threat, otherwise 0.

Data Monitoring. At this stage, the data requirements that need support from the CMI are identified.

Event Analysis. An event can be defined as a pair: a vulnerability and it matching threat. Events can be identified from historical facts, which took place in a specific context. In order to identify the possibility of an event occurring, the likelihood should be estimated considering factors of threat-source motivation and capability, and nature of the vulnerability. Therefore, the likelihood of threat acting over vulnerability is defined as : L_{ji} = $\langle T_j$,$V_i\rangle$

Quantitative Risk Analysis. Risk is defined as the likelihood of an event and its consequence. After potential events and their likelihood have been identified the quantitative risk assessment approach is applied to estimate a level of risk for VMs, Physical Hosts, SLAs and the IP.

C. Model

Individual risks associated with each event (vulnerability, threat) are first calculated and then an aggregated risk for enhancing knowledge based on these individual risks is estimated. Within the general risk assessment model, several elements of risk are identified:

$$R_{j,i} = L_{ji} \cdot I_i$$

Thereafter, the risk for an individual element within an asset under context specific environment can be calculated as follows:

$$R_E = 1 - \prod_{j=1}^{m} (1 - R_{ji})$$

where E = 1,2, . . . is an individual element of risk within the asset. The formula only applies in cases when an element has threats and vulnerabilities associated.

The aggregated risk consists of all individual risks within an asset and is defined as:

$$R_{agg} = 1 - (R_{E_1} \cdot R_{E_2} \cdot \ldots R_{E_k})$$

IV. Fault-tolerant IP Operation

Our approach for fault-tolerant VM management in an IP considers the synergistic operation of several components:

Risk Assessor, which is able to assess the risk level of a virtual machine or a physical host according to the current status and forecast their risk level according to foreseen status. In addition, it can assess/forecast the overall risk level for the provider, which adds SLA failure and legal risks to the aforementioned VM and host failure risks. Risk-level assessments and forecasts are derived using the models described in previous section.

Fault Tolerance Engine (FTE), which is responsible for self-healing infrastructure operation, in particular, it enables tolerance to physical host[1] and virtual machine failures. To detect such failures, it monitors the state of physical hosts and virtual-IT infrastructure. The engine processes this monitoring information and decides whether any corrective action is required during services' operation, such as restarting a recently failed VM.

The engine combines two complementary approaches to enable fault tolerance, namely reactive and proactive. The reactive approach is able to detect already occurred failures and initiate recovery actions, as described in Section IV-A. The proactive approach is able to anticipate foreseen failures

[1] We consider the failure of a whole host, not individual components within a host.

and initiate preventive recovery actions before the actual failure occurs, as described in Section IV-B. Note that if a host or VM fails without prior notice from the proactive fault tolerance mechanism, the described reactive approach takes place.

Note that this component only deals with failures at the infrastructure level. This means that software failures occurred during the execution of services within the VMs are not considered at this level, because this is a responsibility of the SP. Additionally, it assumes stateless services. Therefore, no transaction/check pointing mechanism is needed to handle recovery of services execution upon failures.

However, the engine does not actuate directly. It notifies holistic management components that corrective actions are needed, suggesting what these actions can be. The acceptance of those suggestions and the details of how to carry out necessary actions are left to those higher-level components. In particular, this is the role of **Cloud Manager (CM)**, which decides at global level about fault tolerance recommendations, whether to accept them or not, and if accepted, where to restart/migrate the virtual machine (i.e. in the provider's local infrastructure or bursting it to an external provider). This decision is taken aiming to fulfil the BLOs of the IP, as described in Section IV-C.

A. Reactive Fault Tolerance

To detect the failure of virtual machines, the engine periodically checks if virtual machines are online by pinging them. When a virtual machine does not respond, the engine checks also its CPU consumption and its state. When a VM failure is detected, the CM is asked to restart the failing VM.

To detect the failure of physical hosts, the engine periodically checks if physical hosts are online by pinging them. If any physical resource does not respond within a given time, it is considered to be down, and the CM is asked to restart the VMs running in that host in other hosts.

B. Proactive Fault Tolerance

Besides this reactive behaviour, the engine is also able to anticipate failures in a proactive way. This is done is collaboration with the risk assessor described previously. The Fault Tolerance Engine can configure the risk assessor with the aim to receive proactive notifications when the risk level of VM or host failure is above a given threshold. The rationale used to set this risk level is two-fold. On one side, it can be set for all the VMs running in the provider and for all its physical hosts if the provider's business strategy includes a risk level constraint or an aim to optimize the overall risk level during its operation. On the other side, it can be set individually for the VMs composing a specific service, if a risk level constraint has been specified in the service manifest [2].

Once the engine receives an alert from risk assessor that a physical host is going likely to fail, it informs the CM about this situation by suggesting the migration of all the VMs running in that host to other hosts.

If the engine is alerted of a potential VM failure, the process is the same than the reactive behaviour: that is ask the CM to restart that VM.

C. Holistic Management

This section describes the algorithm used by CM to decide about fault tolerance recommendations coming from the Fault Tolerance Engine, namely whether to accept them or not, and if accepted, where to restart/migrate the virtual machine.

The algorithm is based on multi-faceted management and considers four high-level facets of the IP, namely Trust level, Risk level, Eco-efficiency, and Cost. Using high-level facets allows an easily alignment with the provider's BLOs. The IP's administrator can define these BLOs, specifying in this way the provider's interests. The BLOs can include a high-level facet to optimize and a number of constraints over these high-level facets. For instance, minimize the risk level while keeping the cost under a given value.

When CM receives a request from the engine to restart or migrate a VM, it builds four vectors (one for each high-level facet) of size 3, where each cell represents the forecast of the overall trust, risk, eco-efficiency, or cost for the IP for a corresponding action, namely reject the suggestion to restart/migrate the VM, accept this suggestion and deploy the new VM in the local infrastructure of the IP, and accept the suggestion and burst the VM to another IP.

One vector corresponds to the objective function defined in the BLOs and all of them can be used to check constraints by eliminating values over/below the threshold in the respective vector. The goal is to select the allocation that max/minimizes the objective function in the first vector and fulfils the constraints in the rest.

V. Evaluation

A. Fault-tolerant IP Operation

Our approach for fault-tolerant IP operation has been evaluated by means of a synthetic 24-hour web workload, with a peak of workload during afternoon and an off-peak during late night. In between the peak and off-peak phases, there are intermediate workloads. The statistical description of the workload is as follows. The average service deployment frequency is 1 service every minute during peak and 1 service every 10 minutes during off-peak. The running time of the services follows a Normal distribution, where mean is 6 minutes and standard deviation is 5 minutes, with a maximum of 30 minutes. The number of Virtual Machines per service follows also a Normal distribution, where mean is 1 VM and standard deviation is 2 VMs, with a maximum of 4 VMs.

To have statistically sound data, we have evaluated an environment with a high rate of failures. This rate is unrealistic compared to usual IP operation, but it may reflect some situations such as general outages. Two types of failures have been considered: VM failure and Node failure. Using the Reactive mode, after a VM failure, the VM can be restarted for getting it operational after some seconds or minutes. When a physical node fails, the VMs that are running there are restarted in another node, if there is enough capacity. If there is not enough capacity in other nodes, VMs are restarted in the node that crashed after it gets restarted. The Proactive mode can restart VMs or migrate VMs before the actual failures occur. The placement of the VMs is decided by the Virtual Machine component, that is configured in the same mode as CM

(Energy Efficiency Maximization or Risk Minimization) and, depending on the configuration, will distribute or consolidate the deployed VMs.

We have compared the same workload in a testbed of 4 nodes with 32 CPUs each node. As shown in Table I, for each evaluation, different configurations have been considered for Cloud Manager (CM) and Fault Tolerance Engine (FTE). CM can work in *energy efficiency maximization* or in *risk minimization* mode. FTE can be configured in *reactive* mode (only notifies after VMs or nodes fail), *proactive mode with a low risk threshold* (notifies CM when the probability of VM or node failure is 50%), *proactive mode with a medium risk threshold* (probability of failure is 70%), or *proactive mode with a high risk threshold* (probability of failure is 90%). In our experiments, we evaluate 3 metrics:

Availability: the amount of time in which the VM runs the services, and these are available to the client. We consider that the VM is not available when it (or the physical node that contains it) has crashed, when the VM is being restarted, or when the VM is being migrated. In our experiments, we are not considering software failures.

IP risk level. Calculated using the IP failure risk model.

IP energy efficiency. To simplify the evaluation, the energy efficiency values are shown in percentages. 100% of energy efficiency means that the physical nodes that are switched on are using all their CPUs. Nodes that have no workload are switched off, and they will automatically switch on when the running nodes do not have enough resources for the incoming requests. 0% of energy efficiency would mean that all the nodes are switched on and there are no running tasks.

CM mode	FTE mode	Avail.	Risk	Energy Eff.
Risk Min	Reactive	84%	3.5	30%
Risk Min	Proactive (low)	81%	4.2	37%
Risk Min	Proactive (medium)	89%	3.4	26%
Risk Min	Proactive (high)	94%	3.6	28%
EE Max	Reactive	79%	5.8	60%
EE Max	Proactive (low)	81%	5.6	55%
EE Max	Proactive (medium)	86%	5.7	54%
EE Max	Proactive (high)	88%	5.5	54%

TABLE I: Results of the fault tolerance experiments

Table I shows how *risk minimization* policies provide better % of availability of the VMs, since they tend to distribute the workload and minimize the probability of errors due to resources overload. For proactive mode, the higher the threshold, the higher the % availability. The explanation for this is that a low threshold would trigger migration actions prematurely and, since the VM is unavailable during migration, more migrations will entail less availability. That explains that reactive mode provides more availability than proactive mode with low threshold.

Energy efficiency maximization policies provide lower % of availability because they tend to consolidate all the workloads in the nodes, and this increases the probability of failure due to overload. Reactive mode only restarts VMs when they fail, so there are no preventive actions to avoid VMs to be unavailable when Risk Assessor detects that a physical node is about to fail. Proactive mode is able to increase the % of availability of the VMs because it can anticipate the host failures and migrate many VMs when possible.

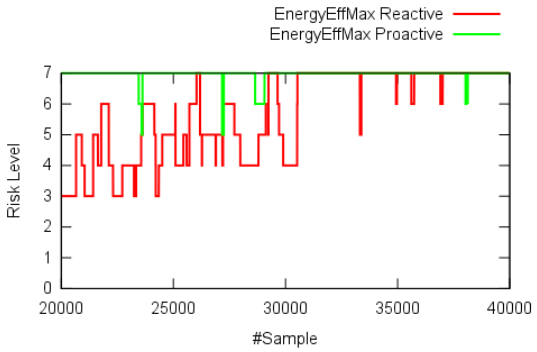

Fig. 2: IP risk during off-peak hours for energy-efficiency maximization policies

As expected, *energy efficiency maximization* policies provide better energy efficiency. However, proactive mode actually decreases energy efficiency when compared with reactive mode. This is because proactive mode tends to distribute VMs when the risk of failure is high to improve the availability, which reduces the consolidation. It is important to remember the main goal of fault-tolerant IP operation is to avoid the maximum number of failures, so this behavior is compliant with this goal.

Note that *proactive risk minimization* with low threshold provides higher energy efficiency than its other *risk minimization* counterparts because it favours the consolidation of VMs when they are migrated from a node that is about to fail to other nodes that already host running VMs. The consolidation caused by the migrations is also increasing the risk of failure. In any case, *risk minimization* policies provide lower risk than *energy efficiency maximization* ones.

The effectiveness of the policies is different depending on the overall load of the physical hosts. This can be appreciated in the following figures. Note that proactive mode in the figures refer to the version with medium threshold. In particular, as shown in Figure 3), the difference in terms of risk between proactive and reactive modes for *energy efficiency maximization* is minimum during peak hours because of the overload of hosts, that minimize the possibilities of redistributing VMs for consolidating them. As shown in Figure 2, during off-peak hours the higher risk level for proactive mode must be considered a proof of the effectiveness of such policy: the consolidation of VMs is always at its maximum possible. Figure 4 and Figure 5 show that *risk minimization* policies have similar results in terms of risk in both proactive and reactive modes (the important fact is that proactive mode helps increasing the availability rate, as explained later).

In terms of energy efficiency, Figure 6 and Figure 7 show that proactive mode provides lower energy efficiency, as mentioned previously. Both reactive and proactive modes are more stable during off-peak hours with respect to peak hours because there are less deployments and failures, so Fault Tolerance Engine is triggered less frequently. A similar behaviour can be observed in Figure 8 and Figure 9, but with

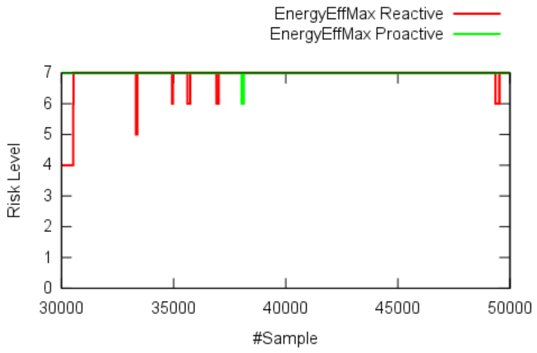

Fig. 3: IP risk during peak hours for energy-efficiency maximization policies

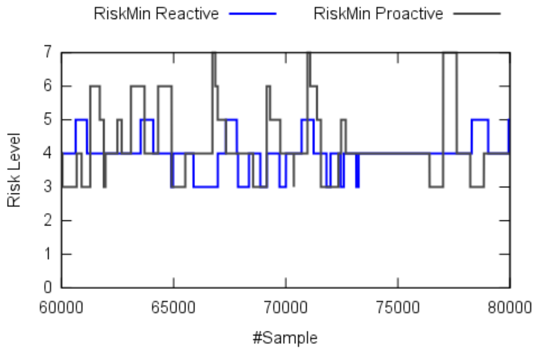

Fig. 4: IP risk during off-peak hours for risk minimization policies

a lower rate of energy efficiency caused by *risk minimization* policies.

For measuring VM availability over time, we have measured one sample each 20 minutes. Each sample measures the average availability of all the VMs deployed in the system during these 20 minutes. According to this, peak hours correspond to samples 30:50 and off-peak to samples 0:30 and 50:80. Figure 10 and Figure 11 show that, as opposite to energy efficiency and risk, the availability of the VMs is kept stable during the whole experiment especially for *risk minimization* in proactive mode, both in off-peak and peak hours.

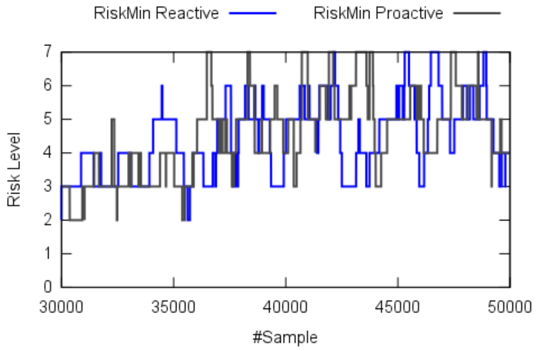

Fig. 5: IP risk during peak hours for risk minimization policies

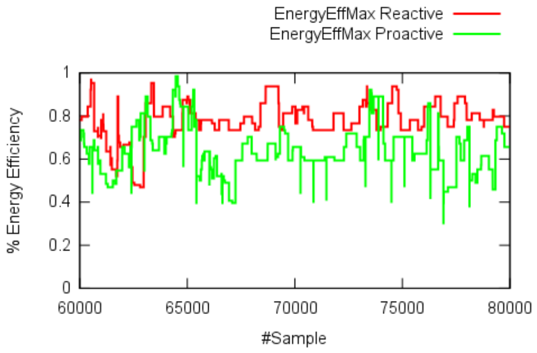

Fig. 6: Energy-efficiency during off-peak hours for energy-efficiency maximization policies

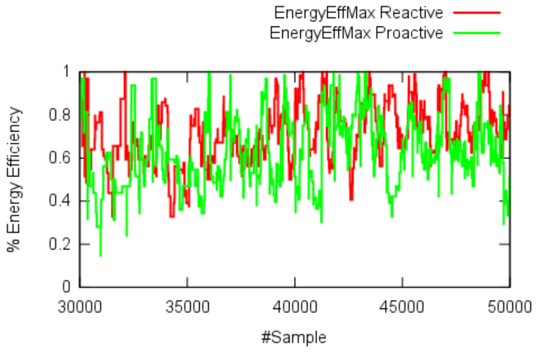

Fig. 7: Energy-efficiency during peak hours for energy-efficiency maximization policies

B. *Early Experiments of Risk Models in Cloud Testbed*

1) Cloud testbed description: The risk infrastructure was deployed as part of the wider OPTIMIS toolkit on a Cloud Infrastructure provided by the University of Umea, Sweden. The infrastructure consisted of two physical hosts each containing 64G of RAM and 32 (2.4 Ghz) cores each. A VM consisted

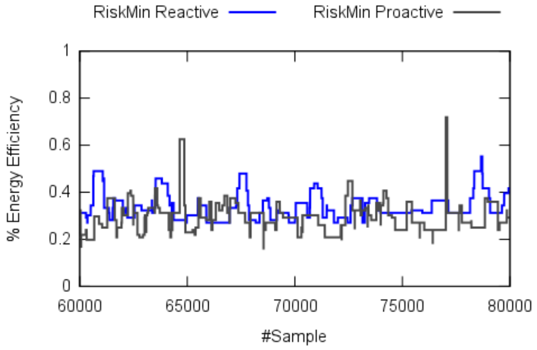

Fig. 8: Energy-efficiency during off-peak hours for risk minimization policies

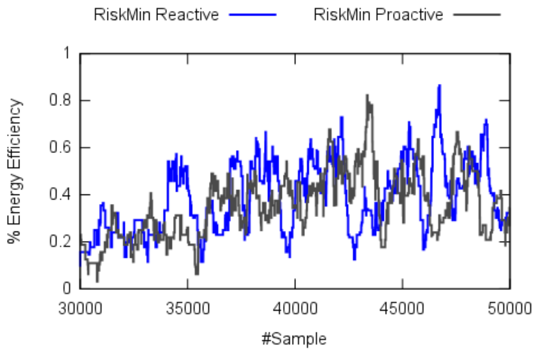

Fig. 9: Energy-efficiency during peak hours for risk minimization policies

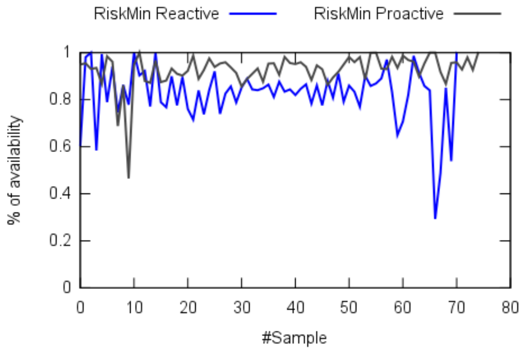

Fig. 10: Availability of VMs for risk minimization policies

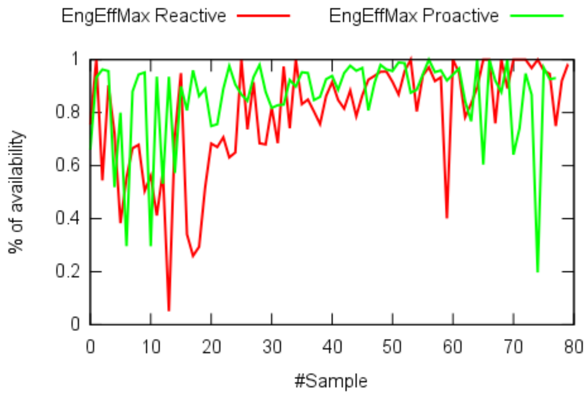

Fig. 11: Availability of VMs for energy-efficiency maximization policies

of 3 cores each. We conducted the experiments with 8 VMs deployed on the hosts, 6 on one and 2 on the other.

The experiment looked at the proactive management of risk using the OPTIMIS holistic management and the risk assessor component using live monitored metrics from the test bed. The concept being that in order to reduce risk to the system based upon the risk levels the CM would proactively react to reduce risk in order to protect the service execution. Risk was primarily measured using a variety of metrics and the primary metrics of concern consisted of CPU and Memory capacity for the physical and virtual hosts. For the SLA the combined risk of the Virtual and Physical Hosts was taking into account. The IP risk included the SLA risk but also the impact of the current computing capacity within the infrastructure using the Xentop metric.

The experiment was conducted using a standard service manifest. The manifest deployed a VM with a service that would gradually increase CPU load up to a certain level on the VM. This has the effect of reducing its capacity for service execution and thus increasing the VMs risk. We detected the increased CPU load via the monitoring tools on the infrastructure which we processed to express on a scale of 0 to 1. The closer the score to 1 relates to the increased probability of the service on the VM or Physical host failing. We completed the risk calculation by multiplying the value by the impact set on the same scale. Risk Bands are then applied to the resultant value. The risk scale of 1-7 was visualised in the Risk UI and presented in the graphical representation of the results. Once the risk of the IP reached 4 the deployment of the VMs was changed in order to reduce risk. The VM with high CPU was given extra capacity or migrated onto a more powerful host.

2) Results: The experimental results were recorded from the service deployment phase in order to show increased risk as the service came online and computation increased. The results were captured to include the proactive management phase which shows a reduce in risk as the processing load is redistributed in the IP by the CM. Table II shows the Risks (and main contributory threats / vulnerabilities) that were recorded for the four main risk types of Physical Host, Virtual Machine, SLA and IP risk at the time of CM notification.

Risk	Vulnerabilities	Threats
Virtual Machine	Unresponsive VM Inadequate Memory Inadequate CPU	CPU Usage Memory Usage VM State OS Release
Physical Host	Slow Network Unstable Host Inadequate Memory Inadequate CPU	Capacity CPU Usage Memory Usage Network performance Blacklisted Domain Name Reboot Needed
SLA	SLA Failure	VM Risk Physical Host Risk
IP	Capacity of IP Exceeded SLA Breach	Capacity SLA Risk

TABLE II: Risks, Threats and Vulnerabilites during Service Execution

The results show the impact on the risk of the proactive holistic management. The CM sets a threshold at which to receive notifications of risk from the risk assessment engine.

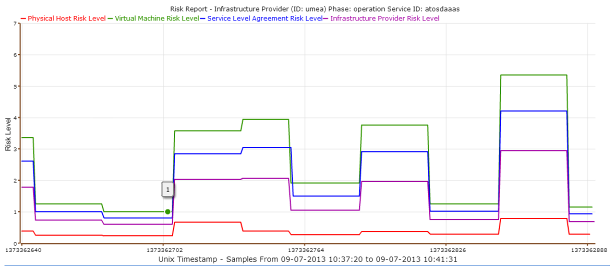

Fig. 12: Risk levels before and after proactive management with threshold at IP 2

In Figure 12 the IP risk is set at a level of 2. The experiment was run over a longer period of time. The risk can be seen to reduce once the threshold is reached. However in this case a clear breach of the threshold can be seen in the final peak. The breach demonstrates a weakness in our proactive approach to the risk levels in that in cases where risk increases quickly and the threshold is at a low level our model can respond too slowly. However, the risk level can be seen to reduce and not go anymore that one point higher.

VI. Conclusion

The management of Cloud Infrastructures in order to improve the delivery of Business Level Objectives such as cost and eco-efficiency needs the development of sensitive tools to link business objectives with the management of the infrastructure. By using risk driven proactive / reactive fault tolerance mechanisms this paper has illustrated a novel approach to delivering a greater level of holistic management in the Cloud. The application of these approaches on both a synthetic and real Cloud testbed has produced results that show increased availability of VMs along with improvements in eco-efficiency and reduction in SLA breaches. However, by lowering the threshold on proactive risk the availability is decreased as more VM migrations take place. Increases in energy efficiency have also been observed using a proactive risk approach due to greater VM consolidation occurring. The initial experiments on a live Cloud testbed shows that proactive risk management is capable of reducing overall IP risk, although when the threshold is low the ability for the threshold to be breached temporarily increases. In terms of future work we are investigating the use of risk forecasting to better aid the proactive management of risk and how Cloud providers can react to events that threaten service execution from various perspectives.

Acknowledgment

This work is supported by the European Commission under FP7-ICT-2009-5 contract 257115 (OPTIMIS), by the Ministry of Science and Technology of Spain and the European Union (FEDER funds) under contract TIN2007-60625, and by the Generalitat de Catalunya under contract 2009-SGR-980.

References

[1] K. Djemame, D. Armstrong, M. Kiran, and M. Jiang, “A risk assessment framework and software toolkit for cloud service ecosystems,” in *Proceedings of the Second International Conference on Cloud Computing, GRIDs, and Virtualization*, Rome, Italy, Sep 2011.

[2] A. J. Ferrer, F. Hernandez, J. Tordsson, E. Elmroth, A. Ali-Eldin, C. Zsigri, R. Sirvent, J. Guitart, R. M. Badia, K. Djemame, W. Ziegler, T. Dimitrakos, S. K. Nair, G. Kousiouris, K. Konstanteli, T. Varvarigou, B. Hudzia, A. Kipp, S. Wesner, M. Corrales, N. Forg, T. Sharif, and C. Sheridan, “Optimis: A holistic approach to cloud service provisioning,” *Future Generation Computer Systems*, vol. 28, no. 1, pp. 66 – 77, 2012.

[3] P. Srivastava, S. Singh, A. A. Pinto, S. Verma, V. K. Chaurasiya, and R. Gupta, “An architecture based on proactive model for security in cloud computing,” in *Recent Trends in Information Technology (ICRTIT), 2011 International Conference on*. IEEE, 2011, pp. 661–666.

[4] C. Chen, W. Han, and J. Yong, “Specify and enforce the policies of quantified risk adaptive access control,” in *Computer Supported Cooperative Work in Design (CSCWD), 2010 14th International Conference on*. IEEE, 2010, pp. 110–115.

[5] S. Pearson, “Toward accountability in the cloud,” *Internet Computing, IEEE*, vol. 15, no. 4, pp. 64–69, 2011.

[6] L. Willcocks and H. Margetts, “Risk assessment and information systems,” *European Journal of Information Systems*, vol. 3, pp. 127–127, 1994.

[7] B. S. Kaliski Jr and W. Pauley, “Toward risk assessment as a service in cloud environments,” in *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*. USENIX Association, 2010, pp. 13–13.

[8] K. Djemame, I. Gourlay, J. Padgett, G. Birkenheuer, M. Hovestadt, O. Kao, and K. Voss, “Introducing risk management into the grid,” in *e-Science and Grid Computing, 2006. e-Science’06. Second IEEE International Conference on*. IEEE, 2006, pp. 28–28.

[9] A. Morali and R. Wieringa, “Risk-based confidentiality requirements specification for outsourced it systems,” in *Requirements Engineering Conference (RE), 2010 18th IEEE International*. IEEE, 2010, pp. 199–208.

[10] K. Djemame, J. Padgett, I. Gourlay, and D. Armstrong, “Brokering of risk-aware service level agreements in grids,” *Concurrency and Computation: Practice and Experience*, vol. 23, no. 7, 2011.

[11] B. Schroeder and G. A. Gibson, “A large-scale study of failures in high-performance computing systems,” *Dependable and Secure Computing, IEEE Transactions on*, vol. 7, no. 4, pp. 337–350, 2010.

[12] T. J. Hacker, F. Romero, and C. D. Carothers, “An analysis of clustered failures on large supercomputing systems,” *Journal of Parallel and Distributed Computing*, vol. 69, no. 7, pp. 652–665, 2009.

[13] B. Cully, G. Lefebvre, D. T. Meyer, M. Feeley, N. C. Hutchinson, and A. Warfield, “Remus: High availability via asynchronous virtual machine replication. (best paper),” in *5th USENIX Symposium on Networked Systems Design & Implementation, NSDI 2008, April 16-18, 2008, San Francisco, CA, USA*, 2008, p. 161.

[14] G. Jung, K. R. Joshi, M. A. Hiltunen, R. D. Schlichting, and C. Pu, “Performance and availability aware regeneration for cloud based multitier applications,” in *2010 IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2010, Chicago, IL, USA, June 28 - July 1 2010*, 2010, pp. 497–506.

[15] S. Fu, “Failure-aware construction and reconfiguration of distributed virtual machines for high availability computing,” in *9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGrid 2009, Shanghai, China, 18-21 May 2009*, 2009, pp. 372–379.

[16] A. B. Nagarajan, F. Mueller, C. Engelmann, and S. L. Scott, “Proactive fault tolerance for hpc with xen virtualization,” in *21th Annual International Conference on Supercomputing, ICS 2007, Seattle, Washington, USA, June 17-21, 2007*, 2007, pp. 23–32.

[17] J. Alonso, I. Goiri, J. Guitart, R. Gavaldà, and J. Torres, “Optimal resource allocation in a virtualized software aging platform with software rejuvenation,” in *IEEE 22nd International Symposium on Software Reliability Engineering, ISSRE 2011, Hiroshima, Japan, November 29 - December 2, 2011*, 2011, pp. 250–259.

[18] J. O. Fitó, M. Macías, and J. Guitart, “Toward business-driven risk management for cloud computing,” in *Network and Service Management (CNSM), 2010 International Conference on*. IEEE, 2010, pp. 238–241.