Chapter 22
Towards Cloud Customers Self-Monitoring and Availability-Monitoring

Sameh Hussein and Nashwa Abdelbaki

Abstract As an attractive IT environment, Cloud Computing represents a good enough paradigm which governments, national entities, small/medium/large organizations and companies want to migrate to. In fact, outsourcing IT related services to Cloud technology, needs monitoring and controlling mechanisms. However, Cloud Customers cannot fully rely on the Cloud Providers measurements, reports and figures. In this book chapter, we cover the two Cloud Computing operation sides. For the first operation side, we provide advices and guidelines for Cloud layers which can be under Cloud Customer control, to allow Cloud Customer contributes in Cloud infrastructure monitoring and controlling. For second operation side, we produce our developed monitoring tool, to allow Cloud Customer contributes in service monitoring. It is for Cloud Customers to self-monitor the Availability as a metric of the outsourced IT service.

22.1 Introduction

Network management is one of the areas which is continuously evolving, widely demanded, and appeared with complex/large networks. It was one of the key components that is discovered when scientists were researching the broad subject of managing computer networks. There exist hundreds of software and hardware products that help network system admins to manage networks under their supervision [1]. Also, there is a variety of tools which guarantee full control over these networks [2]. Network management covers a wide spectrum including security, performance, reliability, class of service, etc.

S. Hussein (✉) · N. Abdelbaki
School of Communications and Information Technology, Nile University, Cairo, Egypt
e-mail: sameh.hussein@nileu.edu.eg

N. Abdelbaki
e-mail: nabdelbaki@nileuniversity.edu.eg

A. E. Hassanien et al. (eds.), *Bio-inspiring Cyber Security and Cloud Services: Trends and Innovations*, Intelligent Systems Reference Library 70,
DOI: 10.1007/978-3-662-43616-5_22, © Springer-Verlag Berlin Heidelberg 2014

Network monitoring is more strategic than its name means. It demands watching for problems on 24/7 manner. Moreover, it's also about optimizing data flow and data access in a complex and changing environment [3]. Services and tools are as numerous and varied as the environment they analyze changes.

In network management world, network monitoring is the proof of concept used to describe the monitoring system. It continuously monitors the network and notifies the network system administrator via messaging system [4]. Usually, notifications are sent in case of a device fails, lack of connectivity, or an outage occurs [5]. Notifications are through E-mails, SMS, warning messages, or alerts. However, network monitoring is performed through the use of tools and software applications [6].

The previous paragraph leads us to a very important question. What can network monitoring systems monitor? Monitoring network will not help, unless we know the right things to be monitored according to service nature, SLA/SLO metrics, and security constrains [7]. Usually, network monitoring is examining bandwidth usage, application performance and server performance. As a fundamental task, traffic monitoring is one of which network maintenance/building tasks are based on [8].

However, network monitoring systems have evolved to oversee an assortment of devices such as, switches, routers, servers, desktops, backbone devices, network nodes, cell phones, and others related. Moreover, network monitoring systems may come with auto-discovery functions, which is able to continuously log and record devices as they are joined, leaved, or undergo of configuration changes [9, 10]. Like such functions, segregate devices dynamically based on rubrics such as IP address, service, type (switch, router, etc.), and physical location [11].

It is an obvious advantage of knowing exactly (and in real time) what has been deployed and what has been automatically discovered to help monitoring. Underused hardware can provide new functions which help pinpoint problems [12]. As an example, if most of the connected devices at a given area are underperforming, then, there might be a resource management problem to be addressed [13].

On the other hand, business ability to link network monitoring with the provided services, moves the strategic interest to service monitoring instead of network monitoring. However, the deep understanding of the service provided leads to determine SLA/SLO characteristics as well as the service metrics that are necessary to be monitored [14]. For example, when we have a website as a service, some metrics are vital to be measured such as, availability, response time, performance, network connectivity, DNS records, database injections, bandwidth, and computer resources like free RAM, CPU load, disk space, and others [15].

Throughout the rest of this chapter, we visit Cloud Computing monitoring and controlling. We examine Cloud layers versus the three basic and main implementation models, address the conflict between Cloud Customer and Cloud Provider, produce recommendations and guidelines for layers under Cloud Customer control, then discuss Cloud service availability to produce our developed Availability Monitoring tool and its flow chart, we examine our tool in test environment. Finally, we conclude and expect the future.

22.2 Monitoring and Controlling Cloud Computing

Measurement climate and monitoring weather get changed once Cloud Computing becomes the atmosphere and the hardware environment of the service to be outsourced. Nothing more than Cloud Computing has different nature compared with ordinary service providers. This difference in nature is steaming from Cloud Computing characteristics like, on demand self-service, elasticity, metered service and ubiquitous access. From business prospective, hosting IT services on public, private, or hybrid cloud is troublesome without appropriate metrics measurements. Where unified visibility, control and awareness of the entire cloud infrastructure is required to monitor cloud operations.

Cloud Computing monitoring has two operational sides. The first is to monitor the core infrastructure of the cloud [16, 17]. It has benefits for the Cloud Provider like increase servers and network equipment availability, fast detection of network outages, and fast detection of Cloud Computing environment problems. The other operational side is to monitor an assortment of service related metrics, to guarantee that the delivered services are matching with the agreed quality levels.

The first operation side can be monitored and controlled via monitoring and controlling Cloud layers. One of its problem is the conflict of interests between Cloud Customers and Cloud Providers. More clarifications of the Cloud layers, conflict of interests problem, and the proposed solution are discussed within the following sections.

The second operation side can be monitored and controlled via monitoring and controlling some selected service metrics. One of its problem is that Cloud Providers sometimes report inaccurate measurements and misleading figures of the Quality of Service metrics. We have developed an Availability Monitoring tool which allows Cloud Customers monitor service availability to compare its results with the ones reported by the Cloud Provider.

However, it is like the flip coin game, as the Cloud Provider flips the Cloud to operate where the Cloud Customer would like to monitor both operation sides. Figure 22.1, represents so.

22.2.1 Monitoring and Controlling Cloud Layers

However, According to Cloud Security Alliances (CSA) work [18], Cloud Computing can be layered into seven layers. Like the rainbow, each color represents a layer in the spectrum. They are Facility (F), Network (N), Hardware (H), OS (O), Middleware (M), Application (A), and User (U). Exactly as the raining weather, a Cloud rains the layers which are allowed for Cloud Customers to monitor and Control. In fact, Cloud atmosphere which represent the implementation model (IaaS, PaaS, SaaS), decides which of these layers are under Cloud Provider control, and which are under Cloud Customer control. In Fig. 22.2, a nature scenario which implements what we were explaining.

Fig. 22.1 Flip Coin Game

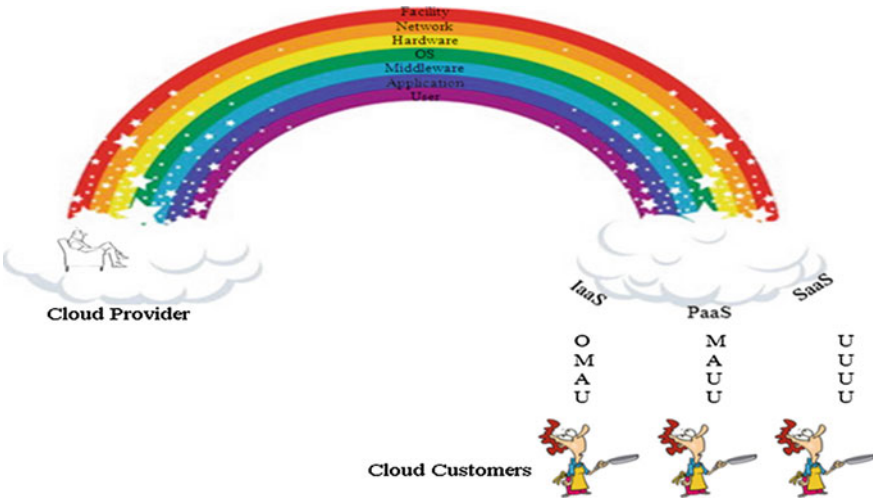

Fig. 22.2 Cloud layers, implementation models, provider versus customer control

As shown in the above Fig. 22.2, Cloud Customers are able to monitor and control the Cloud till a certain depth according to the implementation model. Each layer is linked with its previous and next layer. In SaaS, the Cloud will rain User layer for Cloud Customers to monitor and control. This because other layers are completely managed by the Cloud Provider.

In PaaS, the Cloud will rain User, Application, and Middleware layers for Cloud Customers to monitor and control. For this model, Middleware layer will be a negotiated one, where both Cloud Provider and Cloud Customer should decide who will

have hands on it. Usually, whoever will control it, layer operation recommendations should be shared with the other side. Other layers are completely managed by the Cloud Provider.

In IaaS, the Cloud will rain User, Application, Middleware, and OS layers for Cloud Customers to monitor and control. For this model, OS layer will be a negotiated one, where both Cloud Provider and Cloud Customer should decide who will have hands on it. The same as before, whoever will control it, layer operation recommendations should be shared with the other side. Other layers are completely managed by the Cloud Provider.

Regardless which layer is under Cloud Customer supervision, the Cloud Provider always sits away. Not doing nothing, but for overall management of the entire Cloud as well as remote monitoring and controlling for the left layers.

22.2.2 Cloud Customer/Provider Conflict of Interests

Day after day, Cloud Customers discover new traps and new backdoors for the Cloud technology. This pushes them to negotiate more and more with the Cloud Providers, looking for more visibility and more management over the Cloud layers. This might not be possible in the public Clouds, but for sure, it can be achieved in the private Clouds.

However, any Cloud Provider is used to be keen enough to keep as much layers under his control. On the other side, Cloud Customer is afraid having troubles. Then, Cloud Customer seeks more layers for monitoring and controlling, especially when new drawbacks get discovered. Thus, we have a conflict of interests. Usually, new traps and backdoors tumble customers business in terms of availability, accessibility, continuity, and others which will have financial influences.

The shown Fig. 22.3, represents a scenario where conflict of interests takes place. As human being, Cloud Customer will be very happy running away with the Cloud to try to serve his business. To quickly achieve so, Cloud Customer needs to have control over more layers. However, it is not that easy, the Cloud is bounded by the layers under Cloud Provider control. Also, it might be controlled by other Cloud Customers, in case of public Clouds.

Therefore, it depends on the Cloud atmosphere and its consequences, to determine the area which Cloud Customer is allowed to drive the Cloud within it. As shown in Fig. 22.3, Cloud Provider is used to get back the control of the cloud. Cloud Provider tries to bound the cloud by a wire which is fixed in the ground.

22.3 Cloud Layers Under Cloud Customer Control

As we mentioned before, we always have conflict of interests, where Cloud Customer will have control over some Cloud layers. Regardless the Cloud atmosphere, Cloud Customer will never have a control over deeper layers. At maximum, OS layer, where

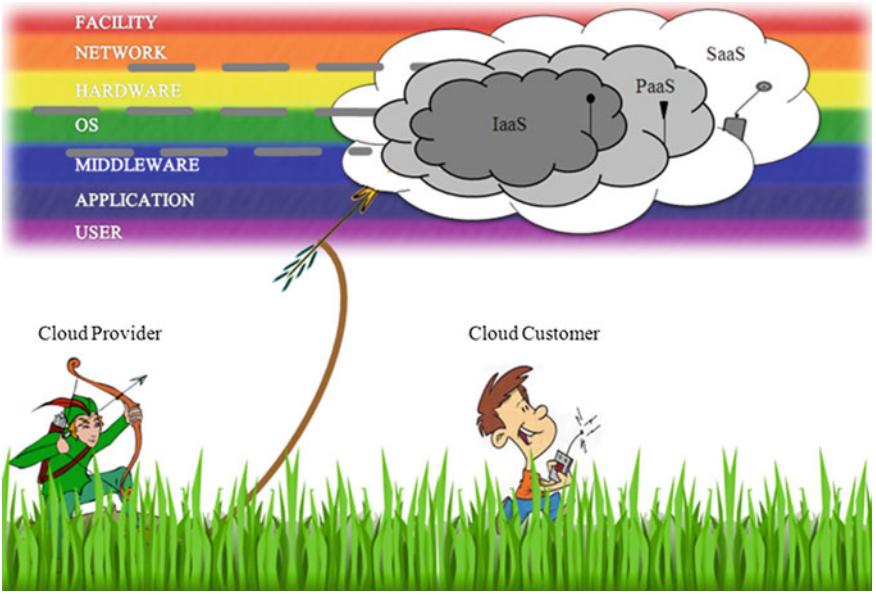

Fig. 22.3 Cloud customer/provider conflict of interests

the agreed recommendations and guidelines will be deployed. In the following we discuss the four Cloud layers which can be under Cloud Customers control within deferent implementation models.

OS, The cloud based OS were evolved to own four mature roles. It acts as well defined interfaces that hide all implementation details. It is responsible for core security services. It manages, hosts, controls, and assign resources for virtualization. It also manages the workloads to ensure quality of service and performance. Therefore, Cloud Customer should be keen to deploy highly secured and controlled OS. Fundamentally, the deployed OS should be cleaned from all additional and non-essential functions. Because only the necessary functions should operate over the OS, these functions should be checked thoroughly for backdoors and vulnerabilities before installing. Also the OS itself must be immune against compromising. However, all system calls between VMs and hardware should be controlled and monitored by the OS. Thus, OS has access to all data passing to or from the VMs, as VMs transferring and processing plaintext data. On the other hand, it also has access to all data stored on VMs, because it is stored on disks which are controlled by the OS, but data can be stored after encryption, where the OS doesn't have its key. Cloud Customer has to ensure monitoring of VMs logs and binary changes, and any offending change has to be returned into a known good state, where monitoring memory dumping and processor over utilize need to be investigated to define and configure new security policies to prevent similar incident. Reports of hardware and software regarding performance should be matched and shared with both customer and provider.

Middleware, as a term, has wide range of definitions. As a simple form, it is a software that connected computers with databases. For Cloud Computing, middleware is a floppy topic that extends from virtualization management tools, to data format conversion. It needs to run security functions for dynamic cloud architectures. Although middleware is important for Cloud Computing, it can be a significant potential weak point for customers and providers when deploying information security assurance mechanisms. Middleware as a concept, still immature layer, especially for cloud computing. However, this layer is the natural place to monitor and secure communication between various system components because it mediates between the applications and the OS, where there are various safeguards to be implemented and pitfalls to be avoided. Then, customers should ensure that middleware will accept and transmit encrypted data. When the customer takes the control over middleware, the provider should protect it against malicious manipulation. As the middleware tends to gain rights to access, manipulate and distribute data, beside specialized functions such as managing access controls, it would be damaging the OS as modifying the OS. To guarantee the avoidance of related concerns, provider should be ready for customer misconfiguration of resources and policies as well as abusing of middlewares functions. For sure, the provider need very intelligent and sophisticated monitoring system for the middleware, but it is very difficult. Also provider needs code inspection tools to scan middleware coding vulnerabilities.

Application, it represent the software hosed by the Cloud Computing. Customers should seek applications in which its source code and business logic have been carefully examined by neutral entrusted third parties for potential flaws and deficiencies. Application must be holding the standards of best practice like sanitizing of all user inputs. In traditional environment, a host based security system can monitor abnormal behaviors in the operating applications. However, in cloud environment, monitoring system should keep track of all violations for each running application. It is difficult to have so, because one instance of an application may serves multiple users simultaneously and doesnt reside on a dedicated host. An application may sit in memory to accomplish multiuser nature of cloud computing. Then, application compromising may lead to memory dump which needs corruption detection mechanisms. When the layer be under customer control, the only different from a traditional computing model is that the monitoring will also be virtualized. Then, it allows for a more costbenefit analysis of monitoring different metrics. This is due to the Cloud architecture inherent scalability and flexibility. In SaaS, providers might develop customized monitoring solutions. However, it should be able to describe those monitoring and remediation strategies in detail.

User, We have two kinds of users. First, is the stand alone users who seek cloud webpage or video services, they have little security impact. Second, users who are members of the customer organization, they should comply with organization security policies. However, both kinds for users access should be monitored and controlled against malicious behaviors. Any aberrant, abnormal or anomaly user of the service should be logged. Like such alerts and notifications should be reported to IT managers of accounts for which their organization is responsible. However, IT security party must add proscribing access to sensitive data in public areas.

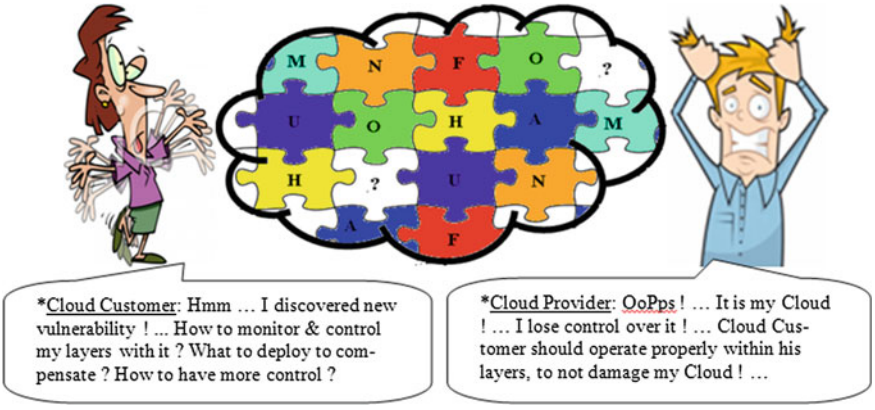

Fig. 22.4 Puzzle game (cloud customer versus cloud provider)

However, managing Cloud layers is more or less similar to the puzzle game (Fig. 22.4), where each player wants to lead putting the missing part to own it. Also, each player thinks of how to control and how to monitor, according to the new traps and backdoors. In our case, Cloud Customer is looking for mechanisms and approaches to benefit his business the most. Therefore, Cloud Provider is looking for how to make his Cloud secure and safeguard his Cloud against Cloud Customer abuse. Then, each one is playing the puzzle game based on his experience and business needs.

22.4 Cloud Service Availability

To address Cloud Computing availability, we can say it is the number one Cloud Customer priority. Since Cloud Computing became a great choice for the IT needs of large companies and organizations all over the world, Cloud Providers were faced with many obstacles that threatened to bring the development and expansion of this technology to a halt [19, 20]. The fact of the matter is that making applications highly available is very difficult. It requires highly specialized and sophisticated tools, systems and trained staff. Furthermore, it is very much expensive. Many Cloud Providers are required to run multiple data centers due to high availability requirements (usually for customers business requirements). Some Cloud Providers have data centers in a standby mode, waiting to be used in a case of a failover [21]. Other Cloud Providers are able to achieve a certain level of success with active/active data centers, where all data centers are ready for incoming user requests. Achieving high availability for services is relatively not easy, establishing a highly available database farm is far more complex. Actually, it is very complex for many companies to establish yearly tests to validate failover procedures.

Being not able to keep services available 24/7 is what all providers fear the most, as even the slightest mishap will have painful consequences on their clients business workflow and reflects on the trust. When we think about it, it is like hypothetically buying the services of Google and not being able to perform online searches.

22.4.1 Cloud Availability Notable Comments

Addressing availability as a metric to be measured, is more or less vital for Cloud Customers. As it means whether the outsourced services are alive or not. Usually Cloud Customers are looking for 100.

First is the planned outages, which can be carried out due to maintenance window, software update, equipment upgrade, install new license after renewal, site migration, adoption of new technology, service upgrade or downgrade, delayed paid installments fee, or customers ask for service suspension [22, 23].

Seconds is the unplanned outages, which can be carried out due to power failure, hardware failure, software failure, network failure, authentication failure, bad configuration, wrong setup, external and internal attacks, or security breaches. Although, there are more reasons behind service lack of availability, but monitoring it and reporting its results, should be totally independent on the actual reasons. At the end of the day, there will be a percentage of service availability, in which both parties should be keen enough to pursue [22, 23].

Raising the point of achieving or not achieving the desired availability percentage, will lead the decision makers to allow compensations and penalties terms and conditions take place according to contact clauses and SLA/SLO financial terms.

On the other hand, the measured availability percentage should be multiplied by the event severity. In other words, when a Cloud Customer experiences lack of service availability (whatever the reason is) in weekends and public bank of holydays, they will raising alerting messages to their Cloud Provider with moderate severity. However, when they suffer the same within normal business days (especially rush hours, where heavy transaction are performed), a strong and high management level channel should be held between both Cloud Customers and Cloud Provider (with very high severity) to ensure that service will be restored within a time window according to SLA/SLO/QoS terms and conditions.

Furthermore, Cloud Provider should be ready with alternatives to guarantees that customers still a live with minimum interruptions. However, like such scenarios should trigger SLA monitoring team and legal department to focus on and activate compensations and penalties terms and conditions.

22.4.2 Surveying the Existing Cloud Availability Monitoring Tools

Nowadays, hundreds of powerful tools are available. Some are for specific service metrics, and others are comprehensive. Some are for LANs, and others are for WANs.

Some are generic to operate within any platform, and others are platform dependent. Some are for general purposes, and others are for specific purposes. Some are made using standard/known programming languages, and others are using special programming languages. Some are offering basic functions, and others are offering advanced functions. However, most of the well-known monitoring tools are using PING command. In fact, it is a programmer decision, where other SNMP commands still can be used. PING command is being widely used by programmers and demanded by Cloud customers for its great benefits. We discuss these benefits through the next section.

As an example of PING monitoring tool, Ping Plotter, EMCO Ping Monitor, Ping for life, Kaseya Ping Monitoring, NirSoft Ping Info View, SoftPedia Ping Monitor, etc.

We can say, that most of these tools are using PING commend for monitoring the availability. PING is considered a type of network monitoring tool at the most basic level. Within the commercial context, other software packages can include a network monitoring system that is developed to monitor an entire business or enterprise network. Some tools and software applications are used to monitor network traffic, such as VoIP, video streaming, mail server, and others.

As common features offered by most of the availability monitoring tools, we have Connection Status Tracking, Connection Loss and Recovery Detection, Regular PING Statistics, Connection Quality Report, Configurable Event Handlers, Alerts and Notifications, Custom Event Handlers, Configurable Terminate Actions, E-Mail and SMS Notifications, Pause/Continue Button, etc.

On the other hand, we found how it is badly in need to have monitoring tools which are measuring the accumulative value of service metrics. This is because usually when Cloud Customers start self-monitoring they would need to append the previous findings. Furthermore, Cloud Customers usually seek advanced technical analysis for further investigations. This is in case of sudden or gradual changes in Quality of Service metrics. To do so, a monitoring tool needs to log all sent and received commands or replies. However, although Cloud Customers dream with the idea of having JAVA developed tool for mobility and portability purposes, it is rarely found. Both missing features has been developed to be offered through the using of our developed tool.

22.5 Our Developed Availability Monitoring Tool

Our developed Availability Monitoring tool allows Cloud Customers to have their own view and calculations over the outsourced service availability instead of total dependency on Cloud Providers reports and measurements. However, Cloud Providers still suffer lack of round-the-clock service, this actually results in frequent outages (planned or un-planned). Then, It is important to monitor the service being provided using internal or third-party tools.

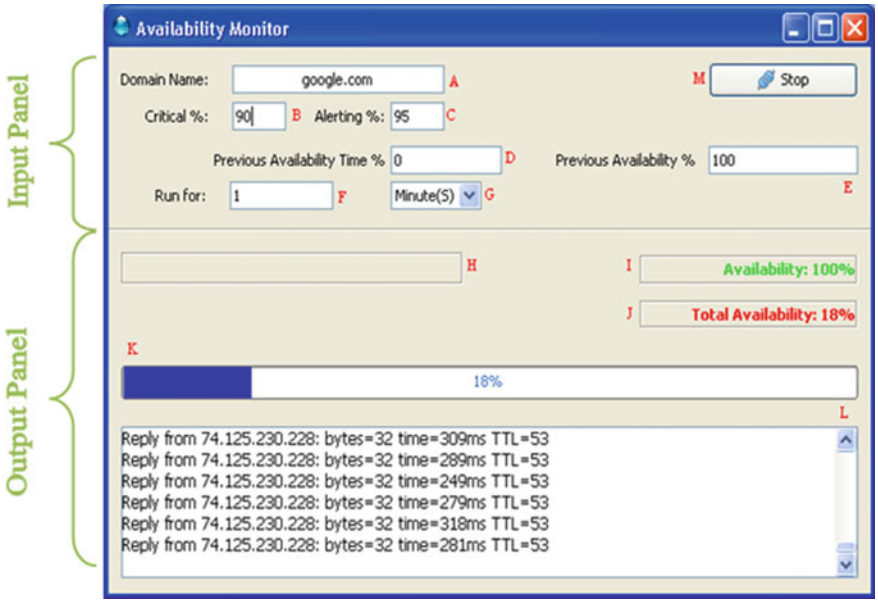

Fig. 22.5 Availability monitor tool GUI

Our developed tool performs PING command continuously for the given domain name, then it collects all replies for analysis. If the returned values of the PING command is TTL, then we consider the desired equipment/service/network is alive and available. On the other hand, if the returned values of the PING command is Timeout, then we consider the desired equipment/service/network is unavailable.

During the running of this function, the tool logs the replies history for further investigations and deep analysis. It uses some probability and statistics mathematical function to calculate and display the availability and the accumulative availability depending on user inputs.

Because we are targeting of an easy and friendly interface, we used JAVA programming languages. In fact, there are 3 billion devices are using JAVA, this clearly shows us how much our developed tool will be compatible with many operating systems.

The above shown tool GUI, Fig. 22.5, was designed to be simple, friendly and easy to use. It can be run over any platform including the recent devices mobile phones, DPAs, and mobile computers. It also can be converted to operate over smart phones like I-Phone, tablets and mobile computers. It uses standard Java classes, and needs minimum resources, in terms of memory, processing, and bandwidth. Tools GUI has two main panel. The Input Panel, for all input fields, program expects user to modify the default values, and the Output Panel, for all output fields, program show and represent the calculated values in this panel and show the progress percentage. However, the Table 22.1 is defining each field of the tool GUI.

Table 22.1 Tool GUI fields function

Letter	Field name	Function
A	Domain name	To enter either desired IP address, device name or URL
B	Critical %	To enter the defined critical range, which will red color the calculated availabilities
C	Alerting %	To enter the defined alerting range, which will orange color the calculated availabilities
D	Pervious availability time %	To enter how long the previous availability lasts
E	Pervious availability %	To enter percentage of the appended availability
F	Run for time	To enter the specified time for the tool to run
G	Time unit	A drop down list to select run time units
H	Error bar	To display errors due to wrong values entered
I	Availability result	To display the colored calculated current availability
J	Total availability result	To display the colored calculated total availability
K	Progress bar	To show how long has the tool being run
L	History log box	A text box where all returned values and replies logged
M	Start/stop button	A button where user can start and stop the tool any time

22.5.1 Flowchart of Our Tool Operation

Exactly as any developed tool, it is highly recommended to deliver the operation flowchart for tool users. Once the user starts to run the .exe file, GUI will appear and then the tool becomes operational to loop inside the flowchart. It keeps running till the user ends it by closing GUI window. The Flowchart in Fig. 22.6, represents all stages, branches, and possible scenarios of tool operation including invalid input parameters. There is only one process that can be triggered any time during the tool operation (running or idle stages), it is the green one. On the other hand, the button STOP it can be clicked any time, but the button START cannot be clicked unless all parameters are entered. Therefore we set default values for each parameter. Table 22.2, shows the default values for each fields in the input panel.

22.5.2 Tool Examination in Test Environment

In this section we will show an example for analyzing a logged history when we were monitoring its availability using our developed tool. Before we go through that, we have to set a group of assumptions and environmental factors, then show the logged history, show the 2-D graphs, then analyze and comment them:

22.5.2.1 Analysis Assumptions

1. We are measuring the availability and other quality related factors.
2. We assume a live and reachable server to perform our measurements.

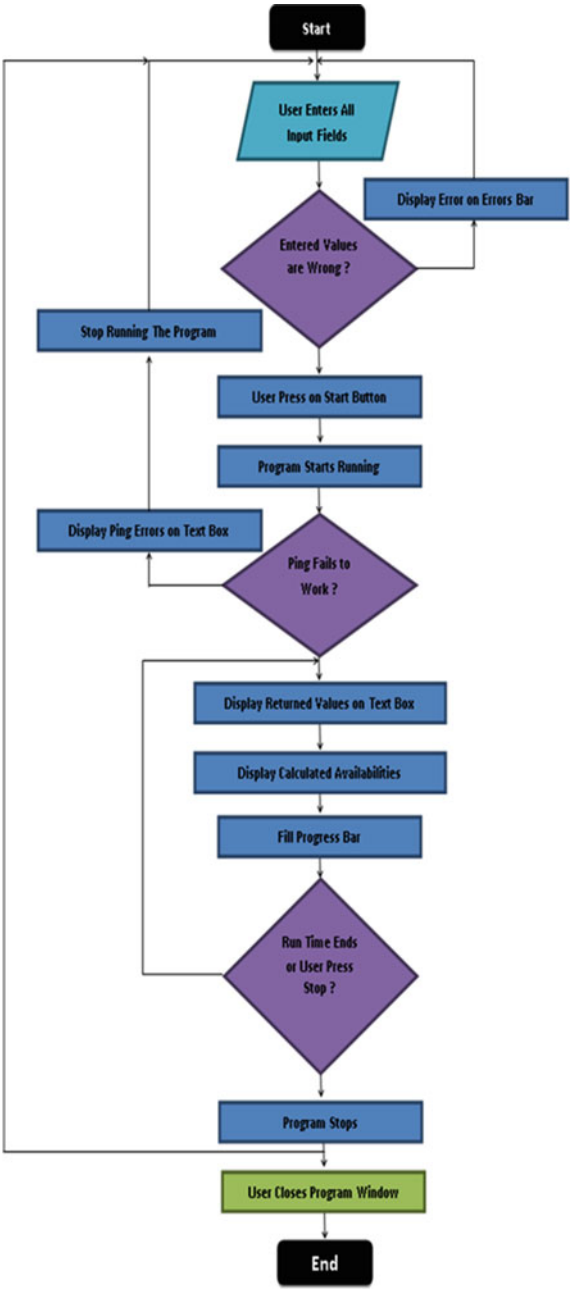

Fig. 22.6 Availability monitor tool flowchart

Table 22.2 Defaul values of input fields

Letter	Field name	Function
A	Domain name	127.0.0.1
B	Critical %	90
C	Alerting %	95
D	Pervious availability Time %	0
E	Pervious availability %	100
F	Run for time	1
G	Time unit	Minutes

3. We assume no software, hardware or networks difficulties.
4. We assume PING traffic is permitted between server and monitoring PC.
5. We assume desired server between replying and not replying.
6. We assume monitoring PC is up and running probably.
7. We assume the unaltered and integrity for the returned values.
8. We assume the monitoring tool works in healthy enough environment.
9. We assume DNS and DHCP servers are alive.
10. We assume DNS and DHCP servers are working probably.
11. We assume tool user is able to run it probably.
12. We assume no appended previous availability.
13. We assume availability thresholds are standard.
14. We assume this example as a part of long term monitoring.

22.5.2.2 Analysis Environmental Factors

1. We monitor in test environment.
2. We monitor in LAN network topology.
3. We monitor a server located within the same LAN of monitoring PC.
4. DNS, DHCP, Monitoring PC, and servers are located in same LAN.
5. Desired server has domain name: Test Server.
6. Desired server has as record in the local DNS server.
7. Desired server has an IP address of: 192.168.1.110
8. Monitoring PC has an IP address of: 192.168.1.10
9. We analyze the logged history via Microsoft Excel.
10. We will show 2-D graphs for our analysis.
11. We created a Macro program to do the same analysis in future.
12. The Macro is used to repeat analysis procedures on an Excel file.
13. The created Macro is usable for any Excel edition.
14. The created Macro is usable for any logged history.
15. We set Critical
16. We set Alerting
17. We set Previous Availability Time

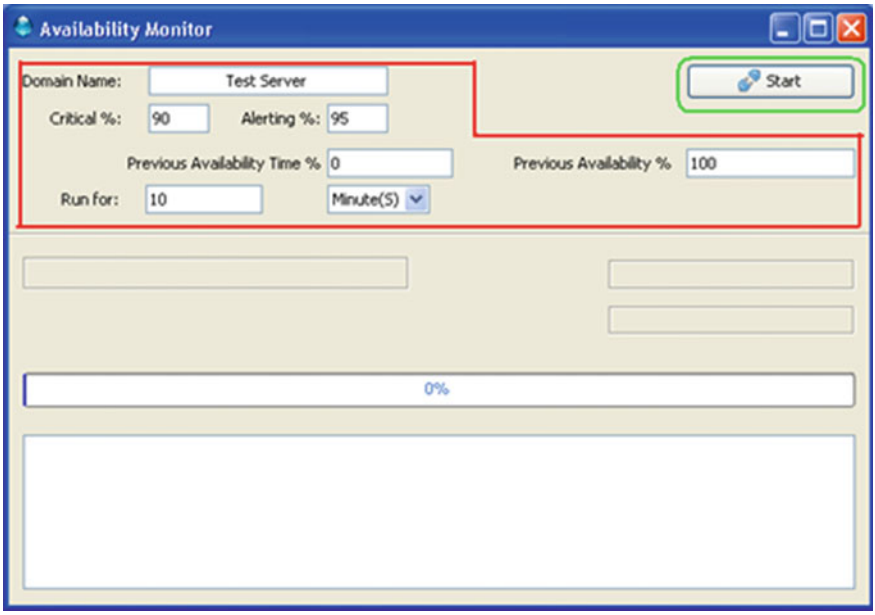

Fig. 22.7 Snapshot of the tool GUI before running

18. We set Previous Availability
19. We set Run For time = 10 min.
20. We analyze a sample of the 10 min logged history.

The above, Fig. 22.7, we show the monitoring tool with values entered according to environmental factors mentioned above, before we start running it. After we have ran the tool with such entered parameters, we will ex-tract the logged history and perform some analysis on it, to conclude some notable comments. Then, we show a sample of the logged history and not all of it, as for 10 minutes, we may have hundreds of lines, thus , we focus on a random time window of it. Figure 22.8, represents the sample.

We will extract all the logged history and perform some basic analysis on it. We opted to analyze some of the returned parameters of the PING command. They are Round Trip Time, Server Availability, Server Total Availability, and Time To Live. However, technical users may make other advanced analysis to reach to deeper concludes.

In the above graph, Fig. 22.9, it shows that round trip time varies from a sample to another. In fact, there are notable differences which indicate (more or less) network instability. These gaps are for the Request timed out replies. It means that for some PING signals, the server was not able to respond.

In the above graph, Fig. 22.10, it shows the availability percentages and how it varies according PING replies, it is crystal clear that monitored desired server suffers

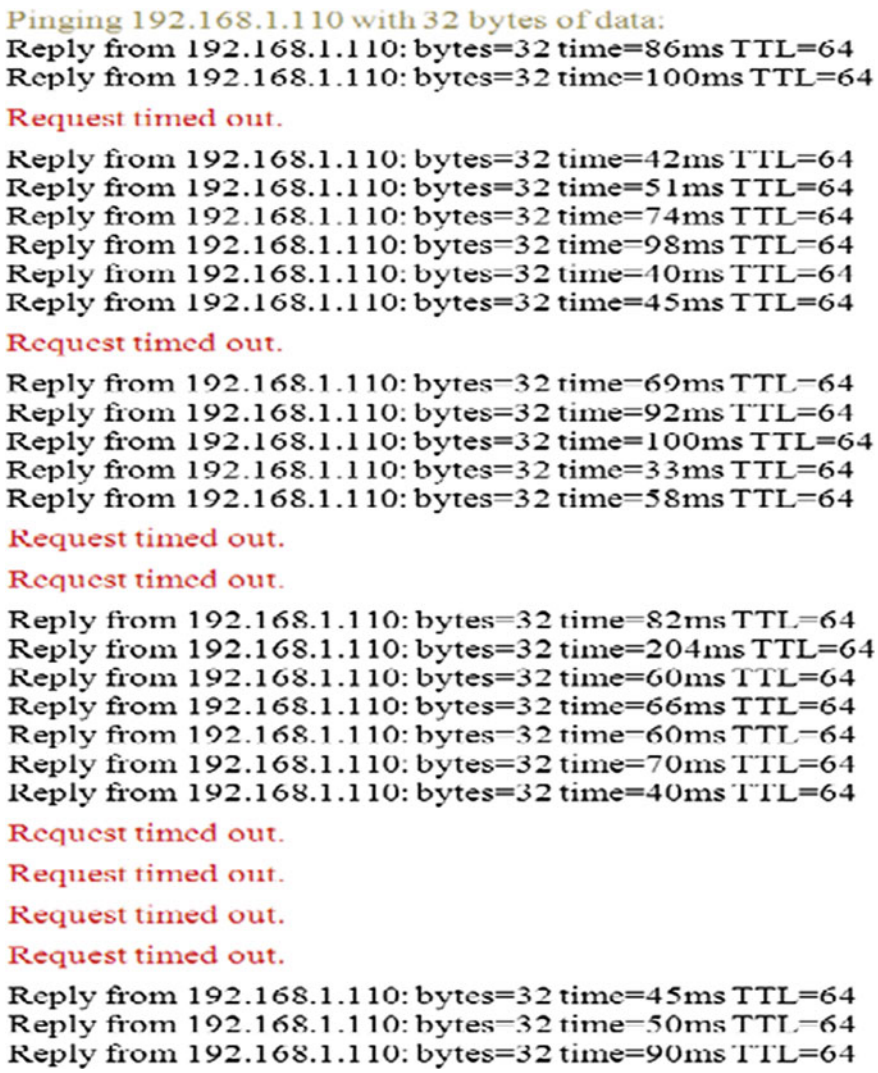

Fig. 22.8 A sample of the 10 min logged history

some problems, as some PING packets are dropped, then it might be over load-ed or suffers connection issues like high IP-Band-Width utilization. Therefore, we have average availability of 81 % which falls in critical range. Thus, a course of corrective and adaptive actions need to be addressed.

In the above graph, Fig. 22.11, it shows the total availability percentages and how it varies according PING replies. In fact, total availability always represents an increasing trend which stemming from its nature (accumulative availability, will be

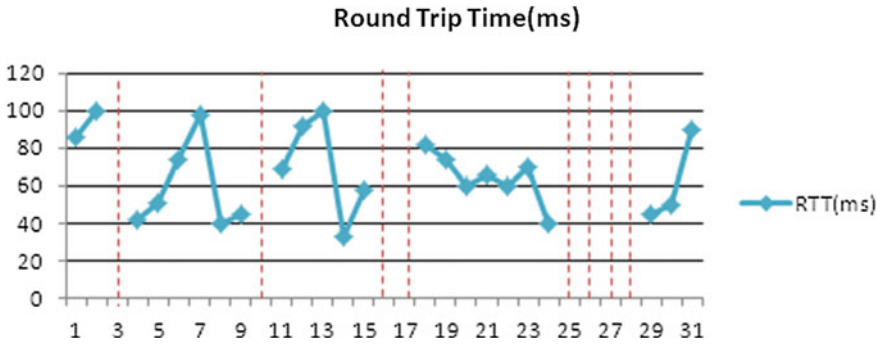

Fig. 22.9 Round trip time graph

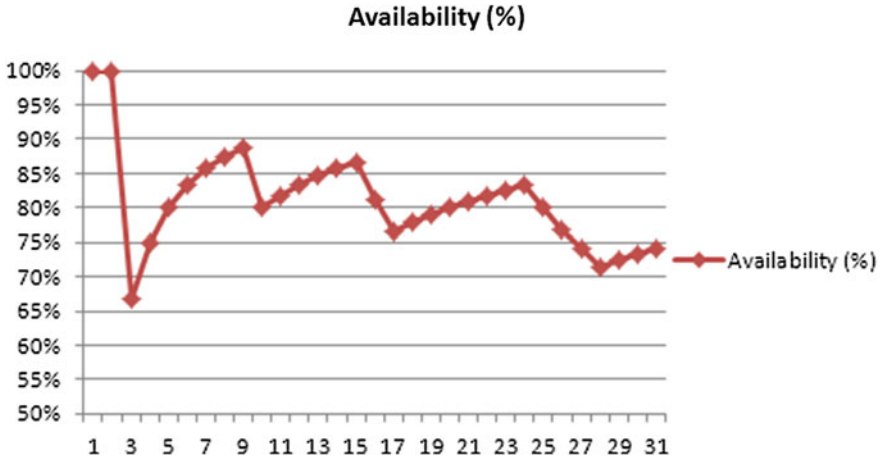

Fig. 22.10 Server availability graph

discussed in the next section). Total availability will be more and more meaningful when we add pervious availability to be appended to the one we are monitoring.

In the above graph, Fig. 22.12, it shows the time to live counts, which means that PING packets still can live 64 hop counts, a hop count means how many nodes a packet can pass through. However, it show a constant straight line because both monitored server and monitoring PC are in the same LAN, but if we going to monitor a WAN server, then networks dynamics will lead to variable TTL.

In real world, there are more and more calculations should be performed (i.e. variance, deviations, standard deviations, upper and lower control limits, upper and lower specification limits, means (averages), medians, and many others).

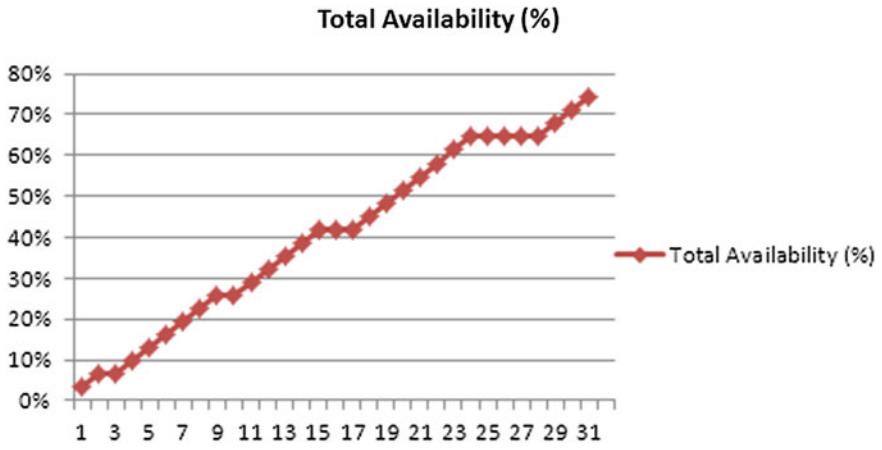

Fig. 22.11 Server total availability graph

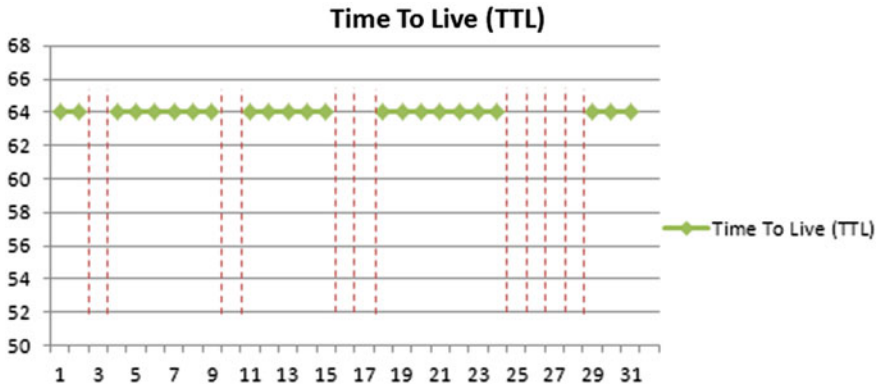

Fig. 22.12 Time to live graph

22.5.3 Tool Accumulative Function

In fact, this tool was not developed only for measuring the current availability, but it also can measure the accumulative availability (total availability). However this accumulative function comes very useful and necessary, to guarantee that at the end of the contract or at SLA termination, user will have the two side of the availability. One is for the period of time the program ran for, and the other for any additional previous availability.

For more clarifications, let us assume that user has SLA contract period of one year (12 months), user was knowing that by the end of September user had 90 % availability (reported by the Cloud Provider), then by the beginning of October user will take the lead and back in-source monitoring of the availability as one of SLO/QoS metrics.

Then, when user starts to run the developed tool, he needs to enter the previous availability = 90 % and the previous availability time = (12 − 3)/12 = 9/12 = 75 %, where the 90 % availability was distributed over 75 % of the one year contract. On the other hand, user also needs to enter run for period = 92 days! where the rest of contract period will be October (31 days) + November (30 days) + December (31 days).

After that, user will press on START button to start monitoring the availability over the remaining interval of the one year contract (last 3 months). During the monitoring, user will have the current availability percentage plus the current total availability which contains both current and previous availabilities.

At the end of the run time (end of the year), the program will stop automatically and will keep all logged history for farther analysis.

If user wants to measure the availability only for certain period of time without appending and previous experiences, then user has to enter previous availability time = 0 and no matters what value user will enter in previous availability. Thus, at the end of the program, user will have the current availability and the total accumulative availability distributed over the needed run period.

22.6 Conclusions and Future Work

As we have seen throughout of this book chapter, contracting with Cloud Provider looks easy, but it is not. It is all about the integration between Cloud Customer and Cloud Provider to achieve the agreed SLAs and meet its objectives. However, monitoring QoS and its related metrics is very much necessary for both Cloud operation sides.

Following the proposed advices, recommendations and guidelines of Cloud layers under Cloud Customer supervision, will resolve the conflict of interests with the Cloud Provider. There should be a sustained effort to allow deep cooperation and collaboration against all Cloud layers. Once held, both parties can guarantee the first Cloud Computing monitoring operational side. Moreover, this work can be intergrated with any related framework. Simply, it can be combined with our proposed IT/Legal framework which has been published before. This integration, enrich the understanding that business need to make its decision towards Cloud Computing.

However, for the seconds Cloud Computing monitoring operational side, we have our tool can be developed to tackle more aspect of the availability being monitored. Also, adding more metrics to be monitored like bandwidth utilization, adding more functions to analyze deeply the PING returned lines, adding an option to monitor more target devices in the same time and in the same program window, allow users to enter more parameters for more precise measurements, add function to draw the metric measured on a 2-D graph, add functions to let the tool send periodic and exceptional reports automatically, add function to let the tool export the deep details to excel files, add functions to provide more calculations options, enhance the GUI to run in the background then pop-up messages in warning cases, and many others which allow cloud computing customer to rely on its own findings to validate and verify the measurements reports provided by the cloud service provider.

References

1. Samaan, N., Karmouch, A.: Towards autonomic network management: an analysis of current and future research directions. Commun. Surv. Tutor. IEEE **11**(3), 22–36 (2009)
2. Chowdhury, K., Boutaba, R.: Network virtualization: state of the art and research challenges. Commun. Mag. IEEE **47**(7), 20–26 (2009)
3. Hyojoon, K., Nick, F.: Improving network management with software defined networking. Commun. Mag. IEEE **51**(2), 114–119 (2013)
4. Timothy, H., Natasha, G., Martin, C., John, M., Scott, S.: Practical declarative network management. In: WREN '09 Proceedings of the 1st ACM Workshop on Research on Enterprise Networking, pp. 1–10 (2009)
5. Jeonghwa, Y., David, H., Keith, W.: Supporting home network management through interactive visual tools. In: UIST '10 Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology, pp. 109–118 (2010)
6. Jeonghwa, Y., Keith, W.: A study on network management tools of householders. In: HomeNets '10 Proceedings of the 2010 ACM SIGCOMM Workshop on Home, Networks, pp. 1–6 (2010)
7. Danny, R., Rolf, S., Constantine, E., Mads, D.: In-Network monitoring, In: Algorithms for Next Generation Networks, Series Computer Communications and Networks Springer, pp: 287–317 (2010)
8. Jorge, V., Antonio, G., Vctor, V., Julio, B.: Ontology-based network management: study cases and lessons learned. J. Netw. Syst. Manag. (Springer) **17**(3), 234–254 (2009)
9. Ralf, W., Keith, W., Motivation: the dawn of the age of network-embedded applications. In: Network-Embedded Management and Applications, pp. 3–21, Springer (2013)
10. Johannes, W.: Secret-Sharing Hardware Improves the Privacy of Network Monitoring. Data Privacy Management and Autonomous Spontaneous Security, Series Lecture Notes in Computer Science Springer, vol. 6514, pp. 51–63 (2011)
11. Changzhong, W, Shukun, C., Yu, M.: One kind of remote monitoring network design based on open CNC system. In: Proceedings of the 2012 International Conference on Cybernetics and Informatics Springer, vol. 163, pp. 2007–2013 (2013)
12. Jiantao, G., Yan, W., Zhao, G.: Efficient network monitoring system. In: Information Computing and Applications, pp. 34–40. Springer (2012)
13. Francesco, F., Luca, D.: High speed network traffic analysis with commodity multi-core systems. In: IMC '10 Proceedings of the 10th ACM SIGCOMM Conference on Internet, Measurement, pp. 218–224 (2010)
14. David, R., Fabin, E., Zihui, G.: Crowdsourcing service-level network event monitoring, ACM SIGCOMM Computer Communication Review - SIGCOMM '10, 40(4), pp: 387–398, (2010)
15. Wuhib, F., Dam, M., Stadler, R., Clem, A.: Robust monitoring of network-wide aggregates through gossiping. Netw. Serv. Manag. IEEE Trans. **6**(2), 95–109 (2009)
16. Rad, M., Fouli, K., Fathallah, A., Rusch, A., Maier, M.: Passive optical network monitoring: challenges and requirements. Commun. Mag. IEEE **49**(2), 45–52 (2011)
17. Xi, C., Zheng, X., Hyungjun, K., Gratz, P., Jiang, H., Kishinevsky, M., Ogras, U.: In-network Monitoring and Control Policy for DVFS of CMP Networks-on-Chip and Last Level Caches, Networks on Chip (NoCS), 2012 Sixth IEEE/ACM International Symposium, pp. 43–50 (2012)
18. Cloud Security Alliance, http://cloudsecurityalliance.org/, (2013)
19. Dillon, T., Chen, W., Chang, E., Cloud Computing: Issues and Challenges, Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference, pp. 27–33 (2010)
20. Salvatore, V., Rocco, A., Beniamino, D., Massimilano, R., Dana, P.: A Cloud agency for SLA negotiation and management. In: Euro-Par 2010 Parallel Processing Workshops, Series Lecture Notes in Computer Science Springer, vol. 6586, pp. 587–594 (2011)
21. Michael, A., Armando, F., Rean, G., Anthony, D., Randy, K., Andy, K., Gunho, L., David, P., Ariel, R., Ion, S., Matei, Z.: A view of cloud computing. Mag. Commun. ACM **53**(4), 50–58 (2010)

22. Yi, W., Blake, M.: Service-oriented computing and cloud computing: challenges and opportunities. Internet Comput. IEEE **14**(6), 72–75 (2010)
23. Mladen, A., Eric, S., Patrick, D.: Integration of high-performance computing in to cloud computing services. In: Handbook of Cloud Computing, pp. 255–276. Springer (2010)