Frequent failure monitoring and reporting in virtualisation environment using backing algorithm technique

S. Anthoniraj*

Department of Computer Science and Engineering, Manonmaniam Sundarnar University, Tirunelveli – 627 012, India
Email: anthonirajcse@gmail.com
*Corresponding author

S. Saraswathi

Department of Information Technology, Pondicherry Engineering College, Pondicherry – 605 015, India
Email: swathi@pec.edu

Abstract: In virtual platforms, disaster management yields to the platform oriented service which is precisely taught using virtual machines. Due to user, execution comes under various verities and simplified structure becomes a complex task. The automated system called backing technique method is introduced, in which proficient performance administration and simplified constitution are guaranteed. But, in the proposed technique, the concept of backing method is presented with additional features, which, persuades the structure so that intricacy diminution is made feasible. Analysis shows that malfunction of Exe, Win-x, kernel, and host can be detrimental for an individual virtual machine. On the basis of this, a new algorithm is developed called backing algorithm which is one of the finest in its concert. The backing algorithm is intelligent to afford continuous monitoring and can accomplish failure reporting proficiently. In addition to backing algorithm concept, we extensively build virtual machine solution provider (VMSP) to sustain this system by frequent monitoring using virtual machine monitor (VMM). These results are provided, backing algorithm with VMM gives that the competent precaution technique when the crash or system boot failure occurs and also it provides the best alert solution for data recovery process at the time of failure.

Keywords: virtual machine; VM; physical machine PM; virtual machine monitor; VMM; virtual machine solution provider; VMSP; Xen; virtual power sub controller; VPSC; virtual disk sub controller; VDSC; virtual network sub controller; VNSC.

Biographical notes: S. Anthoniraj received his BE in Computer Science Engineering from Anna University Chennai, ME in Computer Science Engineering from Vinayaka Mission University Salem and pursing his PhD in Manonmaniam Sundarnar University Tirunelveli, Tamilnadu. He is working as an Associate Professor in Computer Science Engineering at Paavai College of Engineering, Namakkal, Tamilnadu. His area of interest includes virtualisation, Java programming, internet programming, and computer networks, open source tools and components.

S. Saraswathi received her BTech in Computer Science and Engineering from Pondicherry Engineering College, Puducherry, MTech in Computer Science and Engineering from Pondicherry University, Puducherry and PhD in Speech Processing from Anna University Chennai. She is working as a Professor in Information Technology Pondicherry Engineering College, Puducherry. Her research area includes speech processing, natural language processing, and artificial intelligence and wireless networks.

1 Introduction

Virtualisation is a technology with the collective virtual machines (VMs) of various aspects. It can run on a different operating system in a single server machine. It relatively helps to do the work in shared resources on quick aspects of time. Virtualisation includes various technologies like Xen, VMware, OpenVZ and Linux VServer. The multiple operating systems in a same VM helps in doing a process faster by sharing the same kernel space for an individual VM in different platform. Here we propose the work by using Xen environment (Bourguiba et al., 2012).

In a Xen environment, the driver domain hosts the physical device drivers. In addition to actual device drivers, virtual drivers are implemented in both Dom0 and guest domains. The Xen hypervisor acts as a popular open source for virtualisation software. It deploys with their major components such as the hypervisor (which is also known as VMM), VMs (which is known as domains) and the applications. A guest in a Xen is called unprivileged domain and also called as DomU. The special privileged domain which is called as Domain0 (Dom0) is responsible for managing the other guest VMs like creating, migrating and destroying. Virtual machine monitor (VMM) acts as a software layer that administers resource sharing among the synchronised VMs and guaranties the various applications run in secluded environments. A computer on which a hypervisor is running one or more VMs is defined as a host machine. Each VM is called a guest machine. The hypervisor presents the guest operating systems with a virtual operating platform and manages the execution of the guest operating systems. Software that runs in a layer between a hypervisor or host operating system and one or more VMs provides the VM abstraction to the guest operating systems.

Multiple instances have a variety of operating systems, which may share the virtualised hardware resources. With full virtualisation, the VMM exports a VM abstraction identical to a physical machine. So these standard operating systems (e.g., Windows 2000, Windows Server 2003, Linux, etc.) can run just as they would on physical hardware.
In this paper, we focus on the performance evaluation and analysis of frequent failure monitoring in hardware assisted virtualised platform. At the same time as analysing the existing technologies they hold up certain factors. As a result, they are not able to transmit finest accomplishment. But our backing concept bypasses all these margins and acts as a self learning system. The VMM acts as an examiner and validates each process. At the same time the VM service provider provides support to the monitor in such a way that it resolves the issues occurred and provides detailed information to the end user about the instance. In all points of view it does not make compromises to performance, time management and the continuity. However, the effective nature yields to the evolution of a new effective system that encourages users to interact with virtual environments.

The remainder of this paper is organised as follows: in Section 2 discusses literature survey concerning rollback and backup recovery. Next, Section 3 explains backing algorithm implementation in virtualisation environment, Section 4 contains the performance evaluations and we conclude in Section 5.

2 Literature survey

File system recovery (FSR) is a kind of recovery facility that allows users to roll back the file system state to a previous state. FSR, as a kind of undo operation on a computer system, often involves with backup and restore operations, these operations do not only have serious influence on system performance, but also lead to low recovery speed. Despite a large amount of research efforts in computer security, there is no such thing as an unbreakable system. An information system may be damaged due to malicious attacks or honest human errors, and it is fundamentally difficult either to stop all security breaches or to completely prevent mistakes. So the best thing is to restore the damaged systems back to their functional state as soon as possible. Thus, instead of aiming at an absolutely secure system with infinite mean time between breaches, one should strive to reduce the mean time to repair the system. Our solution provides a way to this target. The issue of FSR has been studied by several authors over many years. In paper (Liang and Guan, 2006), the article presents a high efficiency virtual disk environment (VDE) to implement FSR. It can be used to recover the file system quickly even when the computer system suffers the serious disaster such as system crash or boot failure. The VDE is same as virtual disk in the VM environment in some way, but it can be applied to the environment without VM supports. The VDE is very suitable to use in those fields where no data need to be saved, such as PC terminals, internet bars and other similar situations. Beside, these fields, the VDE can also be used in daily usage where a user’s data must be saved. This can be achieved through a simple configuration, for instance, we can use the VDE to protect primary partition where only operating system and applications resident and user’s data are saved to other unprotected partitions. In VDE, the recovered user’s data are saved to unprotected partition in the operating system. The backing algorithm technique prevents the system from thriving by frequent VM monitoring and in case of failure, recovery data gets abstracted from the client user.
Checkpoint-recovery-based (Gerofi et al., 2012) VM replication provides high availability for the entire software stack executed in the VM, and it runs on commodity hardware. Checkpoint recovery is based on fault tolerance captures snapshots of the running VM. The creation of checkpoint in malicious cases is not practical and the screen shot during running status will affect the performance scale. The backing algorithm technique does not capture snapshots. It stores the data directly to the server VM with the help of virtual machine service provider (VMSP). It is important to highlight that a system fails due to the consequences of aging effects accumulated over time. For example, a given aged application fails due to insufficiency of available physical memory caused by the accretion of memory leaks. In this case, the software fault causing memory leaks is a defect in the program code that prevents the use of previously allocated memory, thus the memory leak is the observed effect of an aging-related fault being activated (Machida et al., 2012).

All rejuvenation strategies, in general, have the common fact that the rejuvenation mechanism usually involves stopping the aged software to refresh its internal state. During this process, it is not uncommon to expect service downtime when the rejuvenation is being carried out. Hence, many papers in this field have concentrated on reducing (Grottke et al., 2006; Castelli et al., 2001; Tai et al., 1997) or even avoiding (Alonso et al., 2007; Matias and Filho, 2006; Silva et al., 2007) service downtime during a software rejuvenation execution. Server rejuvenation is often performed by rebooting the operating system which clears all of the internal states that may cause software aging (Huang et al., 1995).

Aging-related bugs in operating systems are reported in previous literature and are known to cause various adverse effects such as resource depletion, performance deprivation and system failure (Cotroneo et al., 2010; Vaidyanathan and Trivedi, 1999). Server rejuvenation issued in the cluster environment as well and the rejuvenation triggers can be controlled by a commercial tool (Castelli et al., 2001). Time-based rejuvenation for VM was modelled as a continuous time-Markov chain (CTMC) for virtualised systems composed of two physical servers and clustered VM sin (Thein and Park, 2009) but it did not incorporate VMM rejuvenation. VMM failure and VMM rejuvenation were taken into an account in the availability model for a virtualised system in Rezaei and Sharifi (2010). A simplified model for migrate-VM rejuvenation was used to make a comparison with the other two techniques. The high-availability function of server virtualisation has been popular in practical system configurations. VMware (2007) high availability feature and Xen server high availability feature Citrix (2009) enable VMs, upon failure of the hosting server, to restart automatically on another hosting server. More recently (Le and Tamir, 2011) is presented as a novel failure recovery mechanism for a hypervisor in which all the states of running VMs are preserved during the hypervisor recovery (Denehy et al., 2002; Howard et al., 1988). Software rejuvenation is a promising technique for improving from the failures caused by software aging by both VM and underlying VMM. It uses the monitoring tool and separate network attached storage system after the system failure occurs. It can postpone or prevent failures by using the concept of proposed, backing algorithm in VM monitoring (Manel and Kamel, 2012).
3 Backing algorithm implementation in virtualisation environment

In a virtual environment, since many VMs are used as a guest OS in the single virtualised machine, the failures occur in the guest OS due to various reasons like system crashes, boot failure, device driver failure, execution file corruptions and application corruption. These will lead to process discontinuation. So many disaster recovery plans for process continuity have been discussed in various fields of virtual environment. In this paper to solve disaster recovery in virtual environment, a new algorithmic technique named backing algorithm is introduced to resolve such problem of failure in a guest VM and also it provides the alert solution as a precaution before failure occurs. The task of a VMM is to monitor each guest VM in the virtual environment. If any crashes occur in a particular guest VM, backing technique will apply in VMM to recover, and it provides an alert to the base machine by intimating the reason for crashes. After the alert intimation received by the base VM, it gives the recovery solution to the particular crashed machine to continue it process without any distraction. As soon as the alert intimation is received, the consequent data of fault VM is stored (copy and pasted) in a partition of a base VM.

As in Figure 1 each operating system installed on hypervisor uses allocation of processor, memory and other resources from the host hardware machine. The partitioning of a system can be measured as the preliminary phase of performance assessment. At this point, the hypervisor formulates it as an effective certainty. As a result, it makes a connection firm with the server. For the successful concert we make use of the ‘Xen’ environment which enables the working path for various operating systems. With this, VMs are connected along with the client systems which act upon the precise tasks and the consequential is kept in the specified locality in the server. The hypervisor supports various platforms and various cloud environments, and the physical machine is the one which maintains entire system and it is mainly used to enhance all the hardware and software. The hypervisor helps the software to be fertilised and also helps the software to be divided into many partitions. As per the monitoring system, it monitors each process in the VM and returns the solution. During the instance of error occurrence, the solution provider ensures the uninterrupted access to the specified process or information.

VMM and VM solution (or) service providers are the integrates part of the hypervisor and it is a program that allows multiple operating systems to share on the single hardware host machine. Also VMSP is a service provider which integrates the individual VMs through virtual controllers like virtual power sub controller (VPSC), VDSC, and virtual network sub controller (VNSC). If the fault is predicted in any of the VMs, it will be monitored by VMSP through virtual controllers by the help of VMM mechanism which induces backing technique mechanism. The backing technique mechanism gets the alert signal with the help backing sub system which indicates that the machine got some disruptions. Immediately the backing sub system creates the backing physical storage of faulted VM, and provides the clear solution through VMSP and makes the process flow smoothly. As compared to earlier check point recovery technique takes less recovery time and also it gets prevented before the fault occurs.
**Backing algorithm implementation:**

\begin{verbatim}
Vmsp – Virtual machine service provider;
V_m – Virtual machine;
P_m – Physical machine;
VMM – Virtual machine monitoring;
Pm – physical machine;
Conn – Connection;

1 Begin
2 Backing ()
3 \forall V_m \in P_m
4 If ((Conn==true) && (V_m[i] ==active) do
5 Set Vmm= Check V_m[i]
6 If ((Vm[i] ==corrupt) && (fault==present)) do
7 Backing algorithm get activated
8 Set Vmsp=active
9 P_m ← Backup _Restore
10 Vmsp send ALERT message to V_m[i]
11 End if
12 If (Vm[i] receives alert message)
13 Relocate the fault in V_m[i]
14 End if
15 End if
16 End
\end{verbatim}

Fault lenience is a complicated assignment according to the typical data security factors, at the same time it transpires a vital factor in prevailing the eminence of the system. Present structures are not capable of predicting the character and disasters. So the prerequisite of an innovative proposal is crucial. As per norms the of backing algorithm, It affords the event and causes the adverse event through virtual monitors and hence the forwarded requests enable the elucidation through the solution provider. The solution precautions can be in the form of backups. The consequential guarantee is the safety and integrity of the specified system. The subsiding of system results inadequate in booting progression and storage systems. The recovery system can be considered as an exposition which resolves all this anxiety. Here kernel has an indispensable role behind the running status of a variety of systems. It facilitates operating systems at disrupted functioning, and the failure of kernel affects the system defectively.

The current mechanism enables the system to take action before it goes to dead state, so that it provides the introduction of a powerful system as shown in Figure 1. VMs can be considered as the customised environment for users. Along with that, their liability of backing algorithm converges to the evaluation of a new intelligent system. According to
this gifted system, a unit containing scheduled VMs undergoes monitoring by an invigilator machine. The whole process of other machines can be traced out by the invigilator. If there arises any instance of failure, then the smart machine would be proficient to analyse the cause and the threads behind it. It is talented to guarantee finest deductions. Thereafter VMMs forward the registers with detailed information about the fault instance to the solution provider.

Figure 1  VMSP architecture (see online version for colours)

Notes: (1) Fault detected VM, (2) service monitoring VMSP, (3) revert backup data through backing technique, (4) backing data physical storage.

4 Performance evaluations

Virtualisation and data security turn into a discussing issue in real world. So the assessment procedure should be revised on the basis of the performance scale. While analysing the Table 1 we are capable to examine leading technologies that leads in the management of the VMs. The prologue of the VDE is towards the recovery under the failure circumstances. But the user’s data are not secure and hence it is against user’s seclusion. Backing algorithm resolves this concern by the use of VMM. In check point recovery, the performance and the practical issues pull back this mechanism whereas the new one yields to the threaded mechanism.
Table 1  Comparative analysis: fault recovery system

<table>
<thead>
<tr>
<th>Ref. no</th>
<th>Authors</th>
<th>Title</th>
<th>Techniques</th>
<th>Con’s</th>
<th>Backing algorithm in virtualisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Jinquan Liang and Xiaohong Guan</td>
<td>A virtual disk environment for providing file system recovery</td>
<td>High efficiency virtual disk environment is used to implement file system recovery in case of system crash or boot failure occurs</td>
<td>Recovered user’s data saved to unprotected partition in the operating system</td>
<td>Backing algorithm prevents from system failure by frequent VM monitoring and in case of failure, recovery data gets abstracted from the client user</td>
</tr>
<tr>
<td>12</td>
<td>Balazs Gerofi, Zoltan Vass and Yutaka Ishikawa</td>
<td>Utilizing memory content similarity for improving the performance of highly Available virtual machines</td>
<td>Check-point recovery based fault tolerance captures snapshots of the running VM</td>
<td>The creation of checkpoint in malicious cases is not practical and the screenshot during running status will affect the performance scale</td>
<td>Backing algorithm does not captures snapshots, it stores the data directly to the server virtual machine</td>
</tr>
<tr>
<td>33</td>
<td>Fumio Machidaa, Dong Seong Kim and Kishor S. Trivedi</td>
<td>Modeling and analysis of software rejuvenation in a server virtualized system with live VM migration</td>
<td>Software rejuvenation is a promising technique for improving from the failures caused by software aging by both VM and underlying VMM.</td>
<td>It uses the monitoring tool and separate network attached storage system after the system failure occurs</td>
<td>It can postpone or prevent failures by using concept of proposed, backing algorithm in virtual machine monitoring</td>
</tr>
</tbody>
</table>

Software rejuvenation technique enables the failure tolerance, but the resultant could not optimise like backing technique. On the whole, backing mechanism manages the concerns related to the virtual environment and assures most favourable performance. The methodical competence of the solution provider is appreciable. On behalf of the error request, the solutions can be in such a way that the reallocation of crashed items in the living state. After the reallocation process solution provider verifies the performance of the specimen and ensures the sustainability. Thereafter the monitoring and solution procedures persist and hence provides uninterrupted service establishment to the system. Here the configuration is enabled with the help of Xen environment. In Figure 2, multiple operating systems provide customised platforms and user friendly environment. It enables the threaded mechanism in such a way that the executions in various platforms will not make any interruption in the accomplishment of platform oriented services.
Frequent failure monitoring and reporting in virtualisation environment

Figure 2  Virtual configurations with multiple operating systems (see online version for colours)

Figure 3  Backup images in virtual manager (see online version for colours)
Guaranteeing data security by declaration cannot be reasonable. The convenient experience makes the user confident in using such systems. Here in our new system the user is capable to trace the backup process and also the system user would be honoured with the user-friendly views. As per Figure 3 in this mechanism it displays the details about the backup location, size and the essential details about the data items. Hence it becomes effortless that user can self evaluate the strength and performance of the specified backing technique.

CPU utilisation can be considered as the direct evaluation technique for performance evaluation. Hereafter, recovery, VDE could not make optimal performance since it is having some problems concerned with the storage of data in unprotected environment. But backing algorithm is not ready to make negotiation in its performance. Figure 4 shows that it is ready to face any circumstances and provides the projected outcome to the users. Hence by analysing CPU utilisation, it is proved that backing algorithm is smarter than the virtual disk so that system can act as a good performer.

Figure 4  CPU utilisation (see online version for colours)

Processing time takes a main role in deciding the accuracy of the precise system. Failure of a system can take place due to various factors. They cannot be completely tolerated. So we might have to concentrate on the continuity process after a failure. A checkpoint is a familiar technique for the data recovery. But it is not able to convey the time administration. According to Figure 5, it lags in providing the precision in recovery time and process continuity. At the same time backing algorithm is proficient to reduce the recovery time, as a result the process continuity becomes an immediate one. So backing technique allows the user to gain maximum benefits in minimum time which suits as an intelligent system. Fault occurrence and recovery are the two faces of a coin. But if it takes place simultaneously, it will affect the process continuity of the system. As an intellectual system, backing algorithm remembers the previous faults and avoids its future existence. At the same time according to Figure 6 our new technique avoids the other faults through effective monitoring. Hence it provides optimal performance continuity and implies its need of subsistence. It also provides incessant services to the end users.
5 Conclusions

In this work, we have analysed the collision of precision of failure predictions on the performance of the proposed strategies. Currently, we only consider cases in which the predicted occurrence time of future failures precedes the observed time. A VM mapping is not suitable in those cases. A backing algorithm mechanism is helpful instead. Although failure prediction of client machine is possible, the prediction results are useful in reducing the failure frequency. As a future work, we will investigate the VM failure and backup recovery to enhance the productivity of the VMs.

References


