

가상 디스크 결함에 의한 가상 머신 생성 실패 진단 및 분석 도구

구민오*, 민덕기*

A Tool for Analyzing VM Creation Failure caused by Virtual Disk Faults

Mino Ku*, Dugki Min*

요약

본 논문에서는 가상화 기술 기반의 가상 컴퓨팅 환경에서 가상 디스크의 내·외부적 결함에 의한 가상 머신 생성 실패 감지 및 원인 분석 기능을 제공하는 도구 (VMBootFailMonitor)를 제시한다. 또한, 본 논문에서는 이러한 가상 디스크 결함으로 초래되는 가상 머신 생성 실패 및 실패 원인의 사례를 제공하며, Xen 기반의 가상화된 시스템 상에서 이를 분석하는 도구의 설계와 일련의 결함 분석 방법을 제공한다. 특히, VMBootFailMonitor는 가상 디스크 분석 모듈, 가상화된 시스템 분석 모듈 및 시스템 로그 분석 모듈을 기반으로 VM 생성 실패에 대한 원인 분석 결과를 제공한다. 그리고 본 논문에서는 가상 머신 운영체제의 정상 부트 시간과 본 논문에서 제시하는 도구를 통한 부트 실패 감지의 시간적 측면에서의 비교 결과를 제공한다. 본 실험 결과, 정상 가상 디스크 기반의 가상 머신 부트 완료 시간 (8~16초) 보다 짧은 3~6초대의 가상 머신 부트 실패 및 가상 디스크 오류 감지 결과를 나타내었다.

▶ Keywords : 가상화, VM 생성, 가상 디스크 결함, 가상 머신, 클라우드 컴퓨팅

Abstract

In this paper, we present a tool (named VMBootFailMonitor) to detect and analyze a failure of a VM boot creation caused by faults on virtual disks of a Xen-based VM. Also, we presents an architecture and detail analysis process of the virtual disk faults in our tool. Especially, VMBootFailMonitor provides a causal analysis result for a case of VM creation failure based on

detection times of VM creation based on abnormal virtual disks. At result, our tool detects VM boot failures (3~6 seconds) within normal VM boot times (8~16 seconds).

▶ Keywords : Virtualization, VM Creation, Virtual Disk Fault, Virtual Machine, Cloud Computing

I. 서론

최근 시스템 가상화 기술은 고성능화 되어가는 컴퓨팅 시스템의 비효율적 자원 활용 및 저비용 고효율 서비스 제공 등의 문제를 해결하기 위한 대안으로써 폭넓은 인지도 형성과 함께 적용 확대가 이루어지고 있다. 특히, 시스템 가상화 기술은 클라우드 컴퓨팅 (Cloud Computing) [1] 플랫폼을 구현하기 위한 기술 중 하나로써 각광을 받고 있으며, 가상화된 독립 컴퓨팅 환경을 제공하는 VDI (Virtual Desktop Infrastructure) 서비스의 핵심 기반 기술로써 자리매김하고 있다. 이러한 시스템 가상화 기술은 Xen [2], KVM [3], VMWare ESX [4], MS Hyper-V [5], OpenVZ [6] 등 다양한 가상 머신 모니터 (Virtual Machine Monitor) 또는 하이퍼바이저 (Hypervisor)로 불리는 가상화 소프트웨어를 통해 서비스 가능한 가상화된 시스템의 구축을 가능케 하는 수준에 도달했다. 더욱이 가상화 기술은 고성능화 되어가는 하드웨어 성능으로 인해 임베디드 시스템에서까지 적용되고 있다 [29,30]. 하지만, 이러한 가상화된 시스템 상에서 구동되는 가상 머신은 생성에서 소멸까지 일정한 주기를 가지고 있으며, 본 가상 머신 운용 주기 중 생성 (Creation) 단계는 향후 운용 (Operation) 단계의 가능 여부를 결정짓는 중요한 단계이다. 그럼에도 불구하고, 가상 머신 생성 단계는 다양한 원인에 의해 실패할 수 있으며, 다양한 실패 원인 중 가상 디스크(Virtual Disk)와 관련된 내·외부적 문제 (사용자에 의한 문제 또는 해킹, 비정상적인 파일 I/O 작업 등)로 인해 발생할 수 있다. 더욱이 VMM 또는 특권 도메인 (Privileged Domain)의 일시적인 오류로 인한 문제뿐 아니라 가상 머신 생성의 지속적인 실패를 초래할 수 있는 연속적 문제까지 치명적인 문제들이 산재해 있다. 하지만, 기존의 가상 머신의 관리 미들웨어 및 도구들에서는 가상 컴퓨팅 자원

영속적으로 발생할 수 있는 가상 머신 생성 실패 문제 발생 시 관리자의 수동적 대응 및 원인 파악은 관리 비용의 급격한 증가를 야기할 수 있다.

따라서 본 연구에서는 Xen 하이퍼바이저를 기반으로 한 가상화된 시스템 상에서 가상 디스크의 내·외부적 오류 및 결합 요인으로 인한 가상 머신 생성 실패의 감지 및 원인 분석을 가능케 하는 도구 (VMBootFailMonitor)의 설계 및 구현을 제시한다. 앞서 언급한바와 같이, 가상 머신 생성 실패 문제는 컴퓨팅 요소의 물리적, 비 물리적 요소들에 의해 발생할 수 있다. 본 논문에서는 이 중 가상 머신 생성 실패를 야기할 수 있는 대표적인 가상 디스크 오류 및 결합 예를 제공하며, 이에 대한 일련의 자동화된 분석 절차에 대해 기술하고 있다.

본 논문의 2장에서는 본 연구와 관련된 가상 컴퓨팅 자원 생성 및 관리에 관한 연구와 가상 컴퓨팅 환경의 신뢰성을 위한 감지 및 분석에 관한 연구 및 물리 디스크 에러에 관한 연구를 제공하며, 3장에서는 가상 머신 생성의 영속적 실패를 초래할 수 있는 원인에 대해 설명하고 있다. 4장에서는 본 논문에서 제시하는 VMBootFailMonitor의 설계에 대해 언급한다. 5장에서는 본 도구의 구현 및 실험에 대해 언급을 하며, 마지막으로 6장에서는 본 논문의 결론을 제시한다.

II. 관련 연구

2.1. 가상 컴퓨팅 자원 생성 및 관리에 관한 연구

VM 기반의 가상 컴퓨팅 환경 구성 단계는 필연적으로 VM 생성 과정을 수반하게 된다. 일반적으로 VM 배치 (VM Deployment 또는 VM Placement) 단계는 일정한 자원 할당 정책에 기반을 두어 특정 가상화된 시스템에 가상 머신을 생성하는 단계를 포함하고 있다. 이러한 VM 배치에 관한 연구는 지금까지 VM 기반의 대규모 가상 클러스터 컴퓨팅 및

기반의 xl, xm과 같은 도구들을 제공하고 있다. 특히, 앞서 언급된 두 도구 (xl, xm)는 OpenNebula에서 가상 머신 생성 및 소멸, 동적 컴퓨팅 자원 변경 등을 위해 활용한다. 이외에도 Laura Grit는 가상 컴퓨팅 자원의 효율적이고 자동화된 관리를 위해 Shirako 시스템[11]을 제시하였다. Rodrigo N. Calheiros는 클라우드 컴퓨팅 환경에서의 Seamless한 모델링, 시뮬레이션, 실험 환경 제공 및 서비스 관리를 위한 CloudSim 프레임워크 [12]를 제시하였다. Michael Isard는 마이크로 소프트사의 자동화된 데이터 센터 관리를 위한 소프트웨어 프로비저닝 및 배포의 자동화, 시스템 모니터링, 소프트웨어 및 하드웨어 고장에 대한 자동화된 복구 기능등을 제공하는 Autopilot [13]을 제시하였다. Nikhil Bhatia는 동적 로드밸런싱, 선점적 노드 관리, VM의 라이브 마이그레이션 (Live Migration) 기능에 특화된 XCM (Xen Cluster Manager) [14]를 제시하였다. Delei Weng은 [15]에서 가상화된 시스템의 관리 정책 기반 자동화된 자원 관리 기법을 소개하였다.

위와 같은 선행된 연구들은 가상 컴퓨팅 자원으로 대표되는 V의 관리 시스템, 도구, 프레임워크 등을 제시하였으나 모두 VM 부트 실패에 대해 문제를 고려하지 않았다. 예를 들어, 앞서 언급된 Xen 기반 가상화된 시스템 관리를 위한 xl 및 xm 도구는 VM 설정 파일 상에 가상 머신 생성을 위한 필수 구성 요소들이 적시되어 있고 문법적 오류가 없다면 가상 머신 생성을 진행하여 가상 머신 생성 자체는 성공적으로 수행하나 가용 여부에 대해서는 보장하지 않는다. 따라서 본 연구에서는 신뢰성 있는 가상 컴퓨팅 자원 생성을 위해 가상 디스크 결합으로 초래된 부트 과정 실패 감지 및 원인 진단 기능을 제공한다.

2.2. 가상 컴퓨팅 환경의 신뢰성을 위한 감지 및 분석에 관한 연구

가상 컴퓨팅 환경에서 컴퓨팅 시스템의 내·외적 고장 및 오류 상황에 대해 신뢰성 있는 컴퓨팅 환경을 제공하기 위한 다양한 연구들이 이루어져 왔다. George W. Dunlap는 시스템의 다양한 공격 및 비 정상적 실행에 관한 분석을 위해 명령어 수준 (Instruction Level)의 VM 모니터링 및 로깅

위한 VMscope[18]를 제시하였다. 본 시스템은 기존의 허니 팻 시스템의 한계점으로 거론된 내부 감시의 한계를 극복하고 시스템 외부에서의 시스템 호출 이벤트 모니터링 기능을 제공한다. Jim Chow는 Aftersight[19]을 통해 가상 컴퓨팅 환경에서 일반적인 응용 프로그램 및 시스템 활동 분석을 위해 시스템 상태 정보를 명령어 수준의 로깅 및 재생성을 지원하며, 특히, 로깅 환경과 분석을 위한 재생 환경의 이원화를 통해 보다 높은 수준의 분석 환경적 유연성을 제공한다. Dan Pelleg는 가상 머신의 구동 중 결합 감지 및 분석을 위해 기계 학습 (Machine Learning) 기법을 적용한 Vigilant [20]를 제시하였다. Hui Kang은 가상 머신간의 상관 특성을 통한 정준상관분석 (Canonical Correlation Analysis)을 활용한 VM 구동 결합 감지와 진단하는 PeerWatch [21]을 제시하였다. Bikash Sharma는 클라우드 컴퓨팅 환경에서 빈번하게 발생하는 VM 동적 자원 규모 변경 (VM Resizing), VM 실시간 마이그레이션 (VM Live Migration) 등과 같은 가상 컴퓨팅 자원 관리상의 문제의 감지 및 진단할 수 있는 CloudPD[22]를 제시하였다.

이 중 Virgilant와 PeerWatch는 결합 감지를 위한 모니터링 요소들을 가상 머신 내에서 구동하지 않는 특징을 가지고 있다. 이러한 구조적 특징은 본 논문에서 제시하고자 하는 가상 머신 생성 실패 진단 및 분석 도구에서도 채용하고 있는 방식과 유사하다. 하지만, Vigilant 및 PeerWatch의 경우, 가상 머신의 자원 활용 및 상태 수집을 위해 Xenmon[23]을 기반으로 하고 있다. 또한 CloudPD의 경우, 가상화 계층으로써 VMWare EXS를 채용하였다. 그리고 VM 수준의 자원 활용 정보를 획득하기 위해 sar를 사용하였으며, 서버의 Cache Miss 및 CPU/메모리 사용 정보를 획득하기 위하여 VMWare vmkperf[24], powercli cmdlets[25]를 활용하였다.

본 논문에서 제시하는 도구에서는 Xenstore 및 Xenstat를 통해 VMM 수준의 가상 머신의 컴퓨팅 자원 정보 수집을 하며, 가상 머신 생성 과정 중 가상 하드웨어 생성 및 조합 이후 생성된 가상 컴퓨팅 환경에서의 부트 과정 실패에 초점을 맞추었다. 이는 앞서 소개된 신뢰성 있는 가상 컴퓨팅 환경에서의 다양한 운영상의 관점에 초점을 맞춘 연구와는 차별성을 가지게 된다.

오류 (Disk Access Error)에 관한 연구는 오랜 시간동안 광범위하게 연구되어 왔다. Sandeep Shah와 Jon G. Elerath는 [26]을 통해 오늘날 디스크 드라이브에서 발생될 수 있는 대표적인 결함의 메커니즘에 대해 기술하고 있다. L. N. Bairavasundaram는 [27]을 통해 대표적인 디스크 접근 에러로써 지연 섹터 오류 (Latent Sector Errors), 비준비 상태 에러 (Not-Ready-Condition Errors), 복구 에러 (Recovered Errors)로 나누었다. 이 중 지연 섹터 오류의 경우, 디스크 섹터의 일부분의 읽기/쓰기 작업 실패 혹은 수정 불가능한 ECC (Error Correcting Codes) 에러 발생 시 나타나는 오류의 유형으로써, 데이터 접근 및 사용의 신뢰성에 손상을 입히는 중대한 문제이다. 더욱이 지연 섹터 오류는 오류가 발생한 섹터의 데이터 접근하기 전까지 감지 할 수 없다 [28].

본 논문에서 제공하는 가상 디스크의 결함에 관한 정보 중 중요 디스크 블록 손상에 의한 결함의 경우, 임의적인 파티션 정보 및 디스크 메타 정보 삭제 혹은 이러한 정보가 기록된 디스크 드라이브 특정 섹터 오류로 인한 접근 불가로 인해 유발될 수 있는 문제와 높은 관련성을 가지고 있다. 따라서 본 논문에서 제시하는 도구에서는 가상 머신 OS의 루트 파티션을 담고 있는 디스크 드라이브의 파티션 정보 획득 불가로 인한 문제를 감지하는 기능을 제공하고 있다. 더 나아가 가상 디스크의 결함은 매칭 된 물리 디스크의 물리적 혹은 기계적 오류를 넘어 VM 설정 오류 혹은 마운트 작업 실패 등으로 인해 유발될 수 있으므로 물리 디스크 오류로 인해 전개되는 가상 디스크 결함만으로 한정되지 않는다.

III. 가상 디스크 인식 실패 원인

가상 머신의 가상 디스크는 가상 머신 생성 시 대표적인 4가지 원인들(VM 커널 오류, VM 설정 오류, 비 마운트의 루프백 디바이스의 가상 디스크화, 디스크 정보 손상)로 인해 인식 과정에 문제를 야기하며 이로 인해 가상 머신 생성 단계를 실패할 수 있는 주요한 요인이 될 수 있다. 더욱이 이러한 가상 디스크 인식 실패를 야기하는 문제의 원인들은 관리자, 사용자, 불특정 침입자 등에 의해 고의적 혹은 비고의적인 행위에 의해 발생할 수 있다. 본 장에서는 앞서 소개한 4가지 대

류가 발생할 경우, 가상 머신 생성 및 운용 장애를 초래할 수 있다. 예를 들어, 가상 컴퓨팅 자원 관리자가 가상 머신의 가상 디스크 기술 부분을 실수로 기입하지 않게 되거나 테스트 목적으로 인해 주석 처리를 하게 될 경우 (Omitting Error), 가상 머신 자체의 생성은 정상적으로 이루어지나 루트 파티션의 확인 불가로 인해 가상 머신의 부트 과정이 실패하게 된다.

3.2. VM 커널 오류

가상 머신의 경우 일반 물리적 머신과 동일하게 부트 과정 수행하기 위해서는 이에 적합한 운영체제 커널이 수반되어야 한다. 하지만, 커널 파일이 손상되거나 혹은 가상 디스크 인식을 위한 파일 시스템 모듈과 같은 커널 모듈을 내제하지 않았을 경우, 루트 파티션 인식 불가로 인한 가상 머신 부트 과정의 실패가 초래된다. Xen 기반의 가상화된 시스템의 경우, PVOPS의 경우, VM을 생성하기 위해 VM 외부적으로 제공되는 커널이 비정상적이거나 파일 시스템 모듈 부재일 경우 부트 과정에 실패하게 된다. 이에 반해 PVGrub 혹은 PyGrub 기반의 VM 부트 커널 호출 방법을 사용할 경우, VM 내부에 적재된 커널 및 커널 모듈의 손상 및 커널 모듈 부재에 따라 부트 과정 실패가 초래된다. 그림 1은 가상 디스크 중 루트 파티션 파일 시스템 모듈이 부재하였을 경우 커널 패닉 (Kernel Panic) 결과를 나타낸 화면이다.

```

0.024288] Kernel panic - not syncing: VFS: Unable to mount root fs on unknown
-block(0,0)
0.024293] Pid: 1, comm: swapper Not tainted 2.6.32-5-amd64 #1
0.024297] Call Trace:
0.024305] <ffffffff812af169> ? panic+0x86/0x143
0.024313] <ffffffff8100a221> ? xen_restore_fl_direct_end+0x0/0x1
0.024320] <ffffffff8151d032> ? mount_block_root+0x2e/0x29a
0.024325] <ffffffff8151d291> ? prepare_namespace+0x169/0x169
0.024332] <ffffffff8151c6bd> ? kernel_init+0x19e/0x1ae
0.024340] <ffffffff81011baa> ? child_rip+0xa/0x20
0.024345] <ffffffff81010611> ? int_ret_from_sys_call+0x7/0x1b
0.024351] <ffffffff810151d9> ? ret_int_restore_args+0x5/0x6
0.024357] <ffffffff81011ba9> ? child_rip+0x0/0x20
    
```

그림 1. 파일 시스템 모듈 부재로 인한 커널 패닉 상태
 Fig. 1. A kernel panic situation during a VM boot process caused by omitting kernel modules.

3.3. 비 마운트의 루프백 디바이스의 가상 디스크화

가상 머신의 가상 디스크화 가능한 매체는 파일, 물리적 파티션, 루프백 디바이스, NFS 기반 외부 스토리지 등 다양

혹은 xm 상에서 VM 설정 오류가 발생되지 않는다. 따라서 상기 가상 컴퓨팅 자원 관리 도구는 가상 머신 생성 시 야기될 오류가 없다고 판단하여 가상 머신 생성 과정을 진행하게 된다.

3.4. 디스크 정보 손상

가상 디스크는 물리적인 시스템에서 사용하는 물리 디스크와 동일하게 파티션 및 파일 시스템에 관한 정보 등을 포함하고 있다. 특히 파일 시스템 타입, 파일 사이즈, 상태 정보를 포함하는 파일 시스템 메타 정보를 저장한 슈퍼 블록(Superblock)과 같은 디스크의 특정 섹터 손상 시 정상적인 가상 머신 생성 및 구동이 불가능하다. 최근 2011년 발생한 농협 해킹 사건의 경우, dd 명령어에 의한 디스크 핵심 정보를 비롯한 디스크 내용이 손상된 대표적인 해킹에 의한 디스크 정보 파괴 사건이다. 더욱이, 디스크 전체 데이터의 삭제는 시간이 다소 걸릴 수 있으나, 디스크 구조상 파티션 정보 및 슈퍼 블록과 같은 민감한 데이터 블록의 손상은 더욱 빠른 시간 안에 이루어질 수 있으며 간단한 파괴적인 행위 혹은 작동 또는 운영 실수로 인한 파괴 행위에 의해 발생할 수 있으며, 영속적인 가상 머신 생성 및 구동이 불가 문제를 야기한다.

IV. 설계

4.1. 설계

그림 2는 본 논문에서 제시하는 VMBootFailMonitor의 아키텍처를 도식화한 것으로써, 본 도구는 크게 테스트 VM에 대한 가상 컴퓨팅 자원 활용 상태를 모니터링 및 조사를 담당하는 “가상 컴퓨팅 자원 모니터링 및 조사 계층”과 VM 부트 실패를 분석하는 “VM 부트 실패 분석 계층”으로 나뉘어진다. “가상 컴퓨팅 자원 모니터링 및 조사 계층”은 세부적으로 가상 컴퓨팅 자원을 모니터링 하는 “VCR Monitor (Virtual Computing Resource Monitor)” 모듈, 테스트를 진행할 가상화된 시스템 상에 테스트 VM의 존재 여부 확인 등의 테스트 환경에 대한 사전 검사를 수행하는 “Pre-Checking” 모듈, VM 생성을 감지하는 “VM Creation

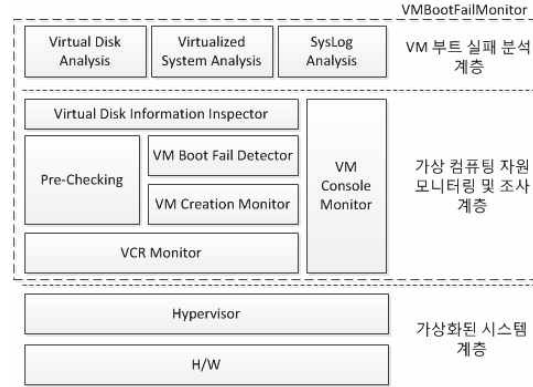


그림 2. VMBootFailMonitor 아키텍처
Fig. 2. Architecture of VMBootFailMonitor

VM 부트 실패 분석 계층은 가상 디스크 상태를 분석하는 “Virtual Disk Analysis” 모듈, VM을 생성을 관장하는 가상화된 시스템에 대한 분석을 수행하는 “Virtualized System Analysis” 모듈과 VM이 부트 과정 중에서 발생하는 콘솔 로그를 분석하는 “SysLog Analysis” 모듈로 구성되어 있다.

4.2. VMBootFailMonitor 실행 프로세스

그림 3은 VMBootFailMonitor의 분석 단계 이전의 전체적인 실행 프로세스를 도식화한 그림이다. 가상 먼저, VMBootFailMonitor는 테스트하고자 하는 VM이 구동 중인지 여부를 검사한다. 만약, 테스트 VM이 구동중이지 않은 상태일 경우, VM을 생성함과 동시에 테스트 시간 (Test Time)을 측정하기 위한 타이머를 구동 및 VM 생성 감지를 시작하게 된다. 테스트 VM의 생성됨을 감지하였을 경우, 해당 VM의 콘솔 출력 결과를 모니터링을 하며, 가상 디스크 역할을 하게 될 가상 블록 디바이스 (Virtual Block Device) 상태를 모니터링 한다. 상기 두 모니터링 결과를 바탕으로 VM의 부트 성공 여부를 검사하게 된다. VM의 부트 성공 여부는 생성된 테스트 VM 운영체제의 제어대기 상태 인지 여부에 따라서 VM 부트 성공 혹은 실패를 판단하게 된다. 만약, VM 생성이 성공적일 경우, 본 도구는 재검사시 생성된 VM인지 여부를 검사하게 된다. 재검사시 생성된 VM의 검

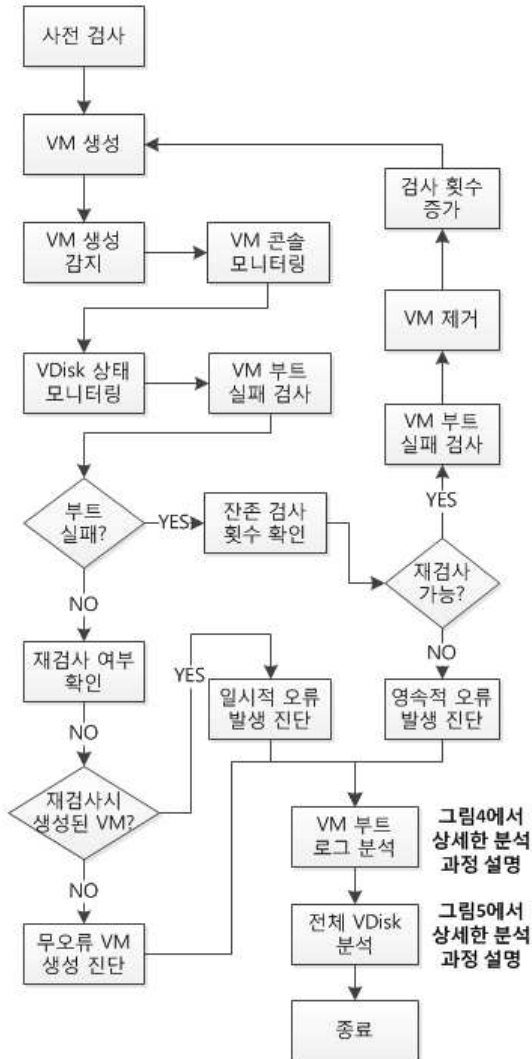


그림 3. VMBootFailMonitor 실행 프로세스
Fig. 3. Execution process of VMBootFailMonitor

VM의 부트 실패 검사 시 부트 실패 상태가 발생하였을 경우, 잔존 검사 횟수에 대해 확인을 한다. 본 도구는 기본적인 반복 검사 횟수를 3으로 하였으며, 실행 시 검사 횟수에 대해 사용자가 지정할 수 있다. 잔존 검사 횟수 확인 시 재검사가

급한 성공적 가상 머신 생성 진단 혹은 일시적 오류 발생 진단과 더불어 VM 부트 로그 분석 단계 및 전체 가상 디스크에 대한 분석 단계에 진입하게 된다.

4.3. 가상 디스크 결함 분석

본 절에서는 VMBootFailMonitor의 부트 실패 및 가상 디스크 결함 감지 방법과 가상 디스크 결함 요인으로 인한 부트 실패 원인을 분석하는 절차를 기술한다. 그림 4는 테스트 VM 생성 시 발생한 부트 로그의 분석 절차를 표현한 그림이며, 그림 5는 테스트 VM의 가상 디스크에 관한 결함 분석 절차를 표현한 그림이다.

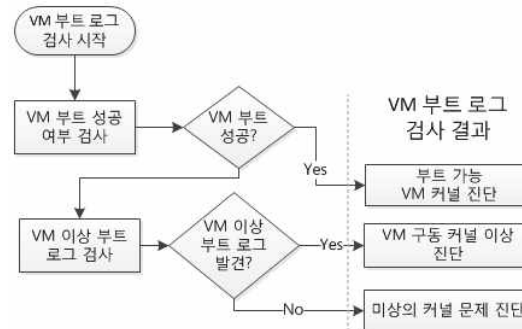


그림 4. VM 부트 로그 분석 과정
Fig. 4. Process of VM boot log analysis

4.3.1 VM 이상 부트 로그 검사 단계

VM은 일반 물리적인 머신과 동일하게 커널 상에 루트 파티션을 마운트 할 수 있는 파일 시스템 모듈을 포함하거나 램 디스크 (Ramdisk) 형태로 제공되어야 한다. 따라서 본 단계에서는 앞서 VM 부트가 성공적으로 이루어졌는지 여부를 검사하고, VM 부트 과정에서 제공된 커널 오류 메시지를 바탕으로 정상 구동 가능 커널인지 여부를 검사한다. VM 부트의 성공 또는 실패 여부는 사용자의 접근이 허락되는 콘솔 로그가 존재하는지 여부를 기준으로 판단한다. 개발 및 서버 운영체제로써 폭 넓게 활용되고 있는 Debian, CentOS, Ubuntu, NetBSD, Suse 등의 운영체제에서는 운영체제 부트 과정 수행 후 사용자 접근이 허락되는 시점에서는 "login:"

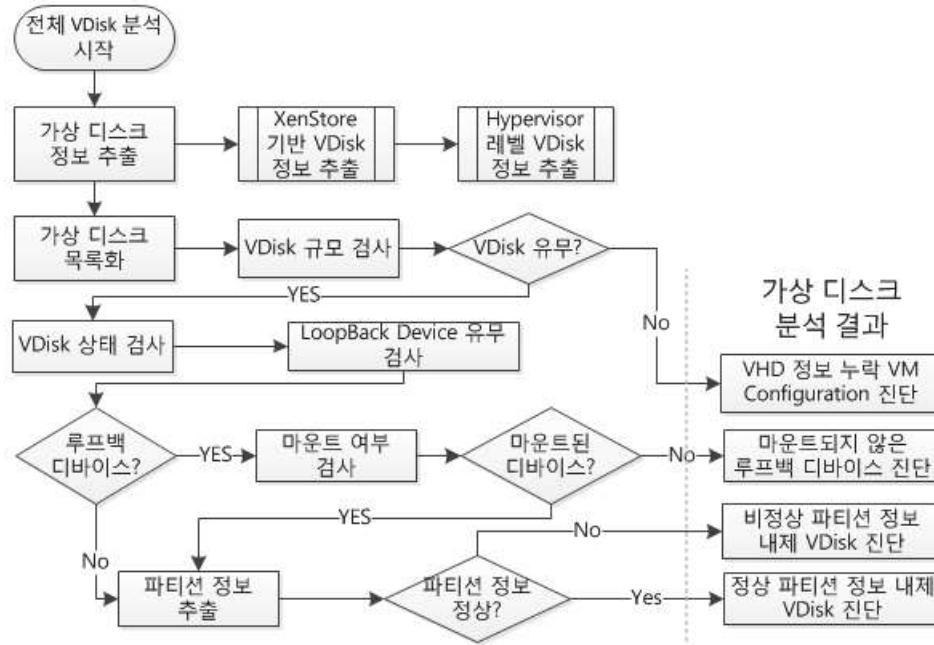


그림 5. 가상 디스크 결합 분석 과정
 Fig. 5. Process of virtual disk fault analysis

부트 실패는 다양한 요인에 의해 발생할 수 있다. 따라서 본 도구에서 다루지 못한 요인에 의해 VM 운영체제 부트 실패가 발생할 경우, 본 도구는 “미상의 커널 문제로 인한 부트 실패”로 진단한다.

4.3.2 VM 설정 오류 검사 단계

본 분석 단계는 VM 설정 오류 중 가상 디스크 설정 부분을 제외하였을 경우 발생하는 VM 구동 실패 유형을 검사하기 위한 단계이다. 본 유형의 오류를 내제한 VM 생성 시 VM은 운영체제를 구동하기 위한 루트 파티션 (Root Partition)을 가진 가상 디스크를 감지할 수 없다. 따라서 VM은 루트 파티션을 가진 가상 디스크를 감지하기 위하여 무한 대기 상태 (Hang 상태)에 빠지게 되므로 지정된 테스트 제한 시간을 초과하더라도 부트 성공 또는 실패 여부를 알 수 없다. 이러한 특징을 바탕으로 “Vdisk 규모 검사” 단계에서는 루트 파티션을 찾기 위하여 VM의 부트 프로세스가 대기 중인지 여

4.3.3 비 마운트의 루프백 디바이스의 가상 디스크화

검사 단계

만약 마운트 되지 않은 루프백 디바이스를 기반으로 VM 생성을 시도할 경우 VM은 앞서 4.3.2 절에서 설명된 “VM 설정 오류”와 유사하게 무한 대기 상태에 빠지게 된다. 하지만, VM 설정 오류 발생 시 나타나는 증상과의 가장 큰 차이점은 가상 머신 모니터에 인식되는 가상 디스크의 규모가 1개 이상이라는 점이다. 그럼에도 불구하고, 인식된 가상 디스크들은 마운트가 되지 않은 장치이므로 구동 가능한 상태가 아니며 가상 머신에서 발생하는 가상 블록 디바이스의 쓰기 및 읽기 상태는 0을 지속하게 된다. 이러한 특징을 바탕으로 가상 디스크가 루프백 디바이스인지 여부를 판단하는 “LoopBack Device 유무 검사” 단계를 수행한다. 만약 본 검사에서 루프백 디바이스로 판단될 경우 마운트 여부를 검사하는 “마운트 여부 검사” 단계를 수행한다. 만약, 마운트되지 않

```
Measurement Time Limit : 15 seconds
Target VM : testVM

Testing Time : 1 second
Testing Time : 2 seconds
Testing Time : 3 seconds
Testing Time : 4 seconds
Testing Time : 5 seconds
Testing Time : 6 seconds

)) Root Partition Detection Fail!
)) log : (initramfs)
```

그림 6. VMBootFailMonitor 구동 화면
Fig 6. A screenshot of VMBootFailMonitor Operation

만약 파티션 정보를 추출할 수 있을 경우, 정상 가상 디스크로 진단한다. 하지만, 파티션 정보를 추출할 수 없을 경우 해당 디스크가 정상적으로 활용 가능한 가상 디스크가 아니므로 “비정상 파티션 정보를 내제한 가상 디스크”로 진단한다. 추가적으로, VMBootFailMonitor에서는 루트 파티션이 존재하는 메인 가상 디스크 외에도 테스트 VM의 전체 가상 디스크에 대한 파티션 정보 손상 혹은 누락 여부를 검사하여 사용자에게 제공한다.

V. 구현 및 실험

5.1. 구현

본 논문에서 제시하는 VMBootFailMonitor는 C 언어를 기반으로 제작이 되었으며, 가상 디스크의 결함 감지 및 정보 (파티션 정보 등) 획득을 위해 libparted를 사용하였다. 그리고 가상 디스크 목록 및 위치 정보 등은 Xenstore를 기반으로 수집하였다. 가상화 소프트웨어로서 Xen 하이퍼바이저 (Ver. 4.1.2)를 기반으로 제작하였다. 그림 6은 본 논문에서 제시하는 VMBootFailMonitor의 구동화면을 나타내며, 그림 7은 본 논문에서 제시한 가상 디스크 결함 유형 중 “비 마운트의 루프백 디바이스의 가상 디스크화” 결함에 의해 발생한

언급한 가상 디스크 결함에 의한 부트 실패 감지 후 실패 원인에 대한 가상 디스크 결함 차원에서의 분석 결과를 나타낸 화면이다. 그림 7에서 볼 수 있듯이, 부트 실패 원인이 할당되지 않은 (마운트 되지 않은) 루프백 디바이스를 기반으로 VM 생성이 원인이었음을 나타낸다.

```
Entire Testing Time : 6
Complete Loop (1).

Virtual hard disk(s) of your test VM have a permanent problem.
VM creation test has failed 2 times.

----- VM Boot Result -----
VM Boot : Failed (2 times)

----- Virtual Disk Diagnosis -----

VBD (1) : /dev/loop6
vdisk path : /dev/loop6
[+ Virtual Disk Diagnosis Result +]
> This virtual disk is not allocated device

-----

Try to destroy Test VM (testVM)

Virtual Hardware Creation Time : 3 seconds
Complete!

root@VTI:/data/lab/development/VMBootFailDetector#
```

그림 7. 부트 실패 감지 후 가상 디스크 결함 원인 분석 화면
Fig 7. A screenshot of virtual disk fault analysis after detecting VM boot failure

5.2.1 실험 환경

표 1은 본 논문에서 제시하는 VMBootFailMonitor의 가상 디스크의 대표적인 4가지 결함 감지를 수행하기 위한 가상화 시스템 및 가상 머신 사양을 나타낸다. 가상화 시스템은 AMD Hexa 코어 시스템이며, 각각의 테스트 가상 머신은 6개의 코어 중 1개를 할당하였다.

표 1. 가상화 시스템 및 가상 머신 사양
Table 1. Specifications of test virtualized system and VM

시스템 자원	가상화 시스템	가상 머신
CPU	AMD Phenom II X6 1055T	Core 1 할당
메모리	16G	1G
운영체제	Debian Linux (Wheezy) 64bit	Debian (Wheezy) Ubuntu (Quantal) Server Edition

본 실험에서는 총 4개의 운영체제를 제공하며, Debian 리눅스와 Ubuntu 리눅스는 운영체제 차원에서 제공하는 반 가상화 머신 (Para-virtualized VM, PVM) 커널을 기반으로 운영체제 환경을 구축하였다. Fedora와 CentOS 운영체제의 경우, 운영체제 차원에서 반 가상화 머신을 위한 운영체제 커널을 제공하지 않았으므로 HVM (Hardware-assisted VM)으로 운영체제 환경을 구축한 뒤, PVM으로 전환 하여 구동 하였다. 가상 머신의 운영체제 서비스는 가상 머신의 부트 시간(Boot Time)에 영향을 미치는 요소 중 하나로써 본 실험을 위한 가상 머신 운영체제 설치 시 운영체제 서비스가 최소화 될 수 있도록 "최소 설치" 옵션을 기반으로 하였다. 테스트 VM의 생성 방법은 가상화 시스템 커널을 가상 머신의 부트 커널로 활용하는 PVOPS 기법으로 생성하여 테스트 하

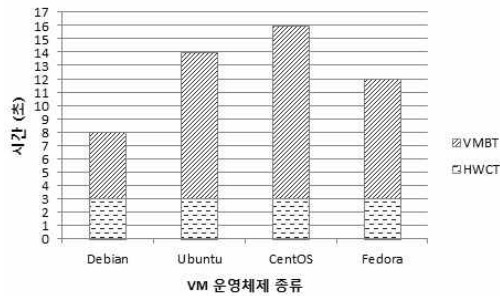


그림 8. 운영체제별 가상 머신 생성 및 부트 시간
Fig 8. VM creation and boot time according to the VM operating system

였다. 가상 디스크의 유형은 RAW 디스크 이미지 형태로 제작하였다. 정상 VM의 부트 시간을 측정하기 위하여 표1에서 표기된 총 4가지의 운영체제를 VM의 운영체제로 선정하여 실험하였다. 반면 본 논문에서 제시하는 가상 디스크 결합 감지는 VM의 운영체제와 상관없이 공통적으로 감지될 수 있으므로 Debian 리눅스를 기준으로 가상 디스크 결합 감지 실험을 실시하였다.

본 도구의 가상 디스크 결합 감지 실험을 위해 결합 감지 항목 중 "디스크 정보 손상"을 아래와 같이 dd 명령어를 사용하여 디스크의 파티션 정보 및 슈퍼 블록 정보를 모두 0으로 채움으로써 결합을 생성하였다.

VM 설정 파일에서 램 디스크를 지정하지 않았다.

5.2.2 실험 결과

가상머신을 생성하기 위해 VMM은 가상 하드웨어를 생성 및 조성을 위한 과정과 생성된 가상 하드웨어를 기반으로 운영체제 부트 과정을 수행하는 VM 부트 과정을 수행하게 된다. 그림 8은 정상 구동 가능한 가상 머신의 가상 머신 운영체제 별 가상 머신의 하드웨어 생성 시간 (Hardware Creation Time, HWCT)와 부트 시간 (VM Boot Time, VMBT)을 나타낸다. 본 실험 결과 모든 운영체제에서 가상 하드웨어 시간은 3초를 기록하였으며, 이러한 가상 하드웨어 생성 시간은 VMM에 의해 결정된다. 하지만 가상 머신의 운영체제에 따라서 8초~16초까지로 다양하게 기록하였다.

표 2는 가상 디스크 결합 유형에 대한 약어표이며, 그림 9는 테스트 VM 생성 후 결합 감지까지의 시간을 나타낸 그림이다. 결합 감지 시간 (Tf)은 아래의 수식과 같이 가상 하드웨어 생성 시간 (Th) 시간과 결합 감지 시점까지의 지연 시간(Td)으로 구성된다.

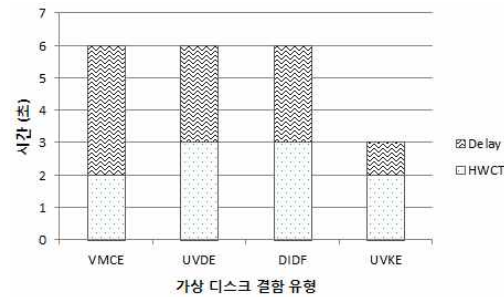


그림 9. 가상 디스크 결합 유형에 따른 결합 감지 시간
Fig. 9. Fault detection time by virtual disk fault types

$$T_f = T_h + T_d$$

본 결합 감지 실험 결과 하드웨어 생성 시간은 2~3초 정도 경과하였으며, 결합 감지 지연 시간(Td)은 1~4초까지 다양하게 나타났다. 또한 앞서 언급한 바와 같이 가상 하드웨어 생성 시간(Th)은 VMM에 의해 결정이 되는 뒤에도 불구하고 2~3

표 2. 결함 감지 시간
Table 2. Fault Detection Time

약어	결함 유형
VMCE	VM 설정 오류 (VM Configuration Error)
UMDE	비 마운트의 루프백 디바이스의 가상 디스크화 (Unmounted Virtual Disk Error)
DIDF	디스크 정보 손상 (Disk Information Damage Fault)
UMKE	VM 커널 오류 (Unsuitable VM Kernel Error)

V. 결론

VM은 소프트웨어 프로세스와 유사하게 생성에서부터 소멸까지 일정한 주기를 가지고 있다. 하지만, VM 생성은 다양한 원인으로부터 파생된 오류와 결함으로 인해 생성 과정을 완료할 수 없는 문제에 직면할 수 있다. 특히, VMM 혹은 특권 관리 도메인의 일시적인 오류로 발생할 수 있는 일시적 결함뿐 아니라 반복적인 시행에도 영속적인 VM 생성 실패를 유발하는 영속적 결함이 있다. 이 중, 사용자의 실수 혹은 해킹 등의 자의적·타의적 오류 또는 결함으로 발생한 가상 디스크 결함은 VM 생성의 영속적 실패를 유발할 수 있는 원인이 된다. 따라서 본 논문에서는 영속적 VM 생성 실패를 유발할 수 있는 가상 디스크의 내·외부적 오류 및 결함 요인들을 제공하며, 이러한 오류 및 결함의 감지와 분석 기능을 제공하는 VMBootFailMonitor를 제시하였다.

본 논문에서 제시한 VMBootFailMonitor는 Xen 하이퍼바이저 기반의 가상화된 시스템 상에서 반 가상화된 가상 머신에 대한 상기 제시된 결함 감지 및 분석 기능을 제공한다. 본 논문에서는 VMBootFailMonitor를 통해 보편적으로 사용되는 VM 운영체제들을 기반으로 한 정상 VM의 부트 시간과 가상 디스크의 내·외부적 결함을 내제한 VM의 부트 실패 및 결함 감지 시간을 비교하였다. 비교 결과, 가상 디스크의 결함으로 인한 부트 실패 및 결함 감지 시간은 정상 부트 시간 내에 감지가 가능했다. 이는 가상 디스크 결함으로 인한 VM 생성 실패 여부가 감지되는 기준 시간을 정상 VM 부트

적화를 용이하게 한다.

더욱이 본 논문에서 제시한 VMBootFailMonitor를 통해 대규모 가상 컴퓨팅 자원 생성 시 발생할 수 있는 일부 VM의 생성 실패에 대해 관리자는 정확하게 문제 발생 여부를 인지할 수 있도록 함으로써 관리 효율성을 향상 시킨다. 더 나아가, 원인을 알 수 없는 새로운 유형의 문제에 대해서도 콘솔 로그를 사용자에게 제공함으로써 문제 발생의 재현 작업 없이 문제의 원인 분석과 해결을 더욱 용이하게 하였다.

영속적 VM 생성 실패는 가상 컴퓨팅 자원 활용을 무력화시키는 가장 근원적이며 치명적인 원인이다. 이러한 치명적 성격과는 상이하게 영속적 문제는 그 문제에 대한 발생 원인이 규칙성을 가지게 된다. 따라서 향후, 이러한 규칙성을 사용자가 손쉽게 추가/제거/수정 할 수 있는 구조적·기능적 개선을 이룰 것이며, 더욱 다양한 VM 생성 실패 문제와 원인들을 분석하여 규칙성을 보강할 것이다.

참고문헌

- [1] P. Mell and T. Grance, "NIST definition of cloud computing," National Institute of Standards and Technology, Oct. 2009.
- [2] B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, I. Pratt, A. Warfield, P. Barham, R. Neugebauer, "Xen and the Art of Virtualization," In Proceedings of Symposium on Operating Systems Principles, pp. 164-177, Oct. 2003.
- [3] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori, "KVM: The Linux Virtual Machine Monitor," In Proceedings of the 2007 Ottawa Linux Symposium, pp. 225 - 230, Jun. 2007.
- [4] VMware ESXi, <http://www.vmware.com/esx/>
- [5] Microsoft Hyper-V, <http://www.microsoft.com/en-us/server-cloud/hyper-v-server>
- [6] OpenVIZ, <http://www.openviz.com>
- [7] OpenStack Cloud Software.

- pp. 124 - 131, May. 2009.
- [9] B. Sotomayor, R. Montero, I. Llorente, and I. Foster, "Resource Leasing and the Art of Suspending Virtual Machines," In Proceedings of the 11th IEEE International Conference High-Performance Computing and Communications (HPCC 09), pp. 59 - 68, Jun. 2009.
- [10] Nimbus, <http://www.nimbusproject.org>
- [11] L. Grit, D. Irwin, A. Yumerefendi and J. Chase, "Virtual Machine Hosting for Networked Clusters," In Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing, Nov. 2006.
- [12] R. N. Calheiros, R. Ranjan, C. A. F. D. Rose, and R. Buyya, "Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services," Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Vol. abs/0903.2525, Mar. 2009.
- [13] M. Isard, "Autopilot: automatic data center management," Operating Systems Review, Vol. 42, No. 2, Apr. 2007.
- [14] N. Bhatia and J. Vetter, "Virtual Cluster Management with Xen," In Proceedings of the Euro-Par 2007 Workshops, LNCS 4854, pp. 185-194, Aug. 2007.
- [15] D. Weng and M. A. Bauer, "Using Policies to Drive Autonomic Management of Virtual Systems," In Proceedings of the 6th International Conference on Network and Service Management, IEEE CNSM 2010, pp. 258-261, Oct. 2010.
- [16] G. W. Dunlap, S. T. King, S. Cinar, M. Basrai, and P. M. Chen, "ReVirt: Enabling Intrusion Analysis through Virtual-Machine Logging and Replay," In Proceedings of the 5th Symposium on Operating Systems Design and Implementation, USENIX Technical Conference, pp. 1 - 15, Apr. 2005.
- [18] X. Jiang and X. Wang, "Out-of-the-Box monitoring of VM-based high-interaction honeypots," In Proceedings of the International Symposium on Recent Advances in Intrusion Detection, pp. 198-218, Sep. 2007.
- [19] J. Chow, T. Garfinkel, and P. M. Chen, "Decoupling dynamic program analysis from execution in virtual environments," In Proceedings of USENIX 2008 Annual Technical Conference on Annual Technical Conference, pp.1-14, Jun. 2008.
- [20] D. Pelleg, M. Ben-Yehuda, R. Harper, L. Spainhower and T. Adeshiyan, "Vigilant: out-of-band detection of failures in virtual machines," ACM SIGOPS Operating Systems Review, Vol. 42, No. 1, Jan. 2008.
- [21] H. Kang, H. Chen, and G. Jiang, "PeerWatch: A Fault Detection and Diagnosis Tool for Virtualized Consolidation Systems," In Proceedings of the 7th international conference on Autonomic computing, pp. 119-128, Jun. 2010.
- [22] Bikash Sharma, Praveen Jayachandran, Akshat Verma, and Chita R. Das, "CloudPD: Problem Determination and Diagnosis in Shared Dynamic Clouds," CSE Technical Report, Pennsylvania State University, May. 2012.
- [23] D. Gupta, R. Gardner, and L. Cherkasovah, "Xenmon: Qos monitoring and performance profiling tool," Technical Report HPL-2005-187, HP Labs, Oct. 2005.
- [24] VMware vmkperf. Monitoring tool, <http://labs.vmware.com/download/143>
- [25] VMware Powercli cmdlets, <http://www.vmware.com/support/developer/PowerCLI/index.html>

Latent Sector Errors in Disk disks," In Proceedings of the International Conference on Measurements and Modeling of Computer Systems, pp. 289-300, Jun. 2007.

- [28] B. Schroeder, S. Damouras, and P. Gill, "Understanding latent sector errors and how to protect against them," ACM Transactions on Storage, Vol. 6, No. 3, Sep. 2010.
- [29] Dongha Shin, Jiyeon Kim, "Implementation of Hypervisor for Virtualizing uC/OS-II Real Time Kernel," Journal of the Korea Society of Computer and Information, Vol. 12, No. 5, pp.103-112, Nov. 2007.
- [30] Sunghoon Son, Jaehyeon Lee, "Design and Implementation of Virtual Machine Monitor for Embedded Systems," Journal of the Korea Society of Computer and Information, Vol. 14, No. 1, pp.57-64, Jan. 2009.

저 자 소 개



구 민 오
 2008 : 건국대학교
 컴퓨터공학부 공학사
 2010 : 건국대학교
 컴퓨터공학과 공학석사
 현 재 : 건국대학교
 컴퓨터정보과 박사과정
 관심분야 : 클라우드 컴퓨팅,
 시스템 가상화, 클러스터 컴퓨팅,
 분산/병렬 시스템
 Email : happykus@konkuk.ac.kr



민 덕 기
 1986 : 고려대학교
 산업공학과 공학사
 1991 : Michigan State University
 Computer Science 공학석사
 1995 : Michigan State University
 Computer Science 공학박사
 현 재 : 건국대학교
 컴퓨터공학부 교수
 관심분야 : 분산/병렬 시스템,
 사이버 피지컬 시스템,
 유비쿼터스 컴퓨팅,
 모바일 클라우드 컴퓨팅,
 미들웨어, 모텔기반 SW 개발
 Email : dkmin@konkuk.ac.kr