



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0059607
(43) 공개일자 2023년05월03일

(51) 국제특허분류(Int. Cl.)
G06F 11/30 (2006.01) G06F 11/07 (2006.01)
G06F 11/34 (2006.01) G06F 9/455 (2018.01)
H04L 41/00 (2022.01)
(52) CPC특허분류
G06F 11/301 (2013.01)
G06F 11/0712 (2013.01)
(21) 출원번호 10-2021-0143954
(22) 출원일자 2021년10월26일
심사청구일자 2021년10월26일

(71) 출원인
포항공과대학교 산학협력단
경상북도 포항시 남구 청암로 77 (지곡동)
(72) 발명자
홍원기
경상북도 포항시 남구 지곡로 319, 328동 304호
홍지범
서울특별시 도봉구 도봉로136길 28, 512동 1801호
(뒷면에 계속)
(74) 대리인
특허법인이상

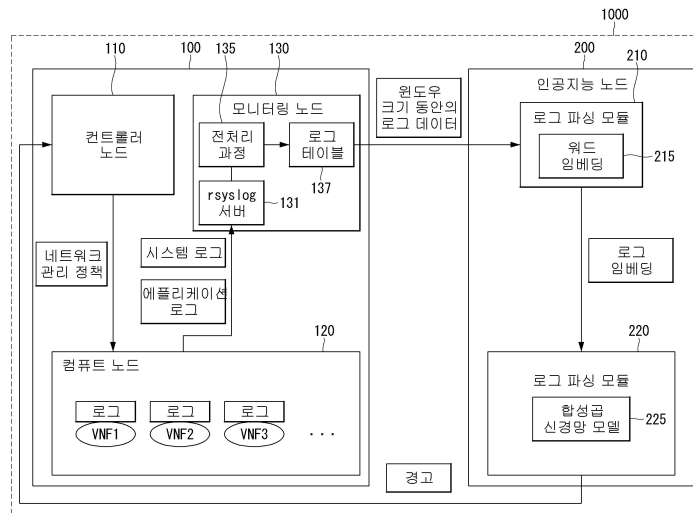
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법, 장치 및 시스템**

(57) 요약

본 발명의 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법은, (a) 가상 머신을 서버 상에서 운영하고 가상 네트워크 기능(VNF: Virtual Network Functions)을 가상 머신에서 운영하여 사용자(client)에게 네트워크 기능을 제공하는 단계, (b) 가상 머신 및 가상 머신을 동작시키는 서버로부터 발생하는 로그를 수집하는 단계, (c) 수집한 로그를 기계학습의 입력인 로그 임베딩 행렬(log embedding matrix)로 변환하는 단계 및 (d) 변환된 행렬을 기초로 가상 머신 및 서버의 고장을 예측하는 단계를 포함한다.

대표도



(52) CPC특허분류

G06F 11/3065 (2013.01)
 G06F 11/3466 (2013.01)
 G06F 9/45558 (2013.01)
 H04L 41/0813 (2022.05)
 H04L 41/0893 (2022.05)
 G06F 2009/45591 (2019.08)

남석현

경상북도 포항시 남구 효성로15번길 9-8, 101호

(72) 발명자

유재형

서울특별시 송파구 올림픽로 135, 211동 1303호

이 발명을 지원한 국가연구개발사업

과제고유번호	1711125937
과제번호	2018-0-00749-004
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	방송통신산업기술개발(R&D, 정보화)
연구과제명	인공지능 기반 가상 네트워크 관리기술 개발
기 여 율	1/1
과제수행기관명	포항공과대학교 산학협력단
연구기간	2021.01.01 ~ 2021.12.31

명세서

청구범위

청구항 1

클라우드 컴퓨팅 관리 시스템을 이용하여 구축하는 네트워크 기능 가상화 인프라(NFVI: Network Function Virtualization Infrastructure)를 사용하여 구축되는 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 시스템에 있어서,

결정된 네트워크 관리 정책(policy)을 컴퓨트 노드(Compute node)로 전달하고, 가상 네트워크 기능(VNF: Virtual Network Functions)들을 별개의 가상 머신 상에서 동작시켜 데이터를 수집하여, 가상 머신 및 가상 머신을 동작시키는 서버로부터 발생하는 로그를 수집하고 전처리 과정을 통해 로그 테이블을 생성하는 네트워크 기능 가상화(NFV: Network Function Virtualization) 구현부; 및

합성곱 신경망 모델의 출력값이 사전에 정의한 임계치를 넘기면 일정 시간 뒤에 해당 가상 머신 혹은 서버에서 고장이 발생할 것에 대해 네트워크 기능 가상화(NFV: Network Function Virtualization) 구현부에 경고를 보내는 인공지능 노드; 를 포함하는,

로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 시스템.

청구항 2

청구항 1에 있어서, 네트워크 기능 가상화(NFV: Network Function Virtualization) 구현부는,

결정된 네트워크 관리 정책(policy)을 컴퓨트 노드(Compute node)로 전달하여 기존 가상 네트워크 기능(VNF: Virtual Network Functions) 또는 서비스 기능 체인(SFC: Service Function Chaining)의 구성 및 배치를 변화시키는 컨트롤러 노드(Controller node);

가상 네트워크 기능(VNF: Virtual Network Functions)들을 별개의 가상 머신 상에서 동작시키는 컴퓨트 노드(Compute node); 및

네트워크 기능 가상화(NFV: Network Function Virtualization) 환경의 컴퓨트 노드(Compute node)에서 수집 가능한 데이터를 모니터링하는 모니터링 노드(Monitoring node); 를 포함하는,

로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 시스템.

청구항 3

청구항 2에 있어서, 컴퓨트 노드(Compute node)는,

가상 네트워크 기능(VNF: Virtual Network Functions)들을 이용하여 서비스 기능 체인(SFC: Service Function Chaining)을 배치하고, 이를 통해 사용자에게 네트워크 기능 가상화(NFV: Network Function Virtualization) 기반의 네트워크 기능을 제공하는,

로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 시스템.

청구항 4

청구항 2에 있어서, 모니터링 노드(Monitoring node)는,

고장 예측을 위하여 컴퓨트 노드(Compute node)에서 동작하는 각 가상 머신들과 가상 머신을 구동하는 서버들에서 발생하는 로그 데이터를 수집하고,

로그 수집을 위하여 로그 수집 유틸리티인 rsyslog(The rocket-fast syslog server)를 활용하여 rsyslog 서버를 동작시키고,

컴퓨트 노드(Compute node)의 각 가상 머신에서 rsyslog를 동작시키고 로그를 rsyslog 서버로 전송하도록 설정하여 각 가상 머신에서 발생하는 로그를 rsyslog 서버를 통해 수집하는,

로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 시스템.

청구항 5

청구항 2에 있어서, 모니터링 노드(Monitoring node)는,

가상 머신에서 동작하는 가상 네트워크 기능(VNF: Virtual Network Functions)에서 출력시키는 애플리케이션 로그(application log)뿐만 아니라 가상 머신에서 발생하는 시스템 로그(system log), 커널 로그(kernel log)를 수집하고,

가상 머신들을 구동시키는 서버에서도 rsyslog 설정을 통해 모든 로그를 모니터링 노드로 전송시키고,

수집된 로그에 전처리 과정을 거친 후 로그 테이블에 저장하는,

로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 시스템.

청구항 6

청구항 2에 있어서, 인공지능 노드는,

로그를 고장 예측 모듈의 입력인 로그 임베딩으로 변화시키는 모듈인 로그 파싱 모듈; 및

합성곱 신경망의 학습으로 사전에 정의된 시간차(gap) 뒤에 고장이 발생할지를 예측하는 고장 예측 모듈; 을 포함하는,

로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 시스템.

청구항 7

청구항 6에 있어서, 로그 파싱 모듈은,

모니터링 노드로부터 타임스탬프를 기준으로 사전에 정해진 윈도우 크기(window size) 동안의 로그 데이터를 로그 테이블에서 추출한 데이터인 로그 데이터를 전달받고,

네트워크 기능 가상화 인프라(NFVI: Network Function Virtualization Infrastructure)에서 사전에 수집한 대량의 로그 데이터를 워드투벡터 라이브러리의 입력데이터로 사용하여 사전에 생성되어 학습된 워드 임베딩 모듈을 포함하는,

로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 시스템.

청구항 8

청구항 6에 있어서, 고장 예측 모듈은,

수학식 1의 로스 함수(loss function, 목적 함수)에 따르는 합성곱 신경망 모델을 포함하는,

로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 시스템.

(수학식 1)

$$loss = y_{true} \times \log\left(\frac{y_{true}}{y_{pred}}\right) \times Class\ Weight_1 + (1 - y_{true}) \times \log\left(\frac{1 - y_{true}}{1 - y_{pred}}\right) \times Class\ Weight_0$$

단, 우변의 각 항들은 y_{true} 는 정답 출력값,

y_{pred} 는 합성곱 신경망에서 예측한 결과,

$Class\ Weight_0$ 과 $Class\ Weight_1$ 은 각각 0과 1에 대한 클래스 가중치(class weight)임.

청구항 9

청구항 8에 있어서, 고장 예측 모듈은,

로스 함수에 수학적 2의 클래스 가중치 방식을 적용한 합성곱 신경망 모델을 포함하는,
로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 시스템.

(수학적 2)

$$Class\ Weigh_0 = \frac{TotalNumSamples}{NumSamplesInClass0}, Class\ Weigh_1 = \frac{TotalNumSamples}{NumSamplesInClass1}$$

단, *TotalNumSamples*는 전체 학습용 데이터 수,

*NumSamplesInClass1*과 *NumSamplesInClass0*는 각각 전체 학습용 데이터 수 중 출력값이 1로 태깅된 데이터의 수(고장 데이터의 수), 그 외의 데이터의 수(정상 데이터의 수)임.

청구항 10

클라우드 컴퓨팅 관리 시스템을 이용하여 구축하는 네트워크 기능 가상화 인프라(NFVI: Network Function Virtualization Infrastructure)를 사용하여 구축되는 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 장치에 있어서,

프로세서(processor); 및

프로세서를 통해 실행되는 적어도 하나의 명령이 저장된 메모리(memory); 를 포함하고,

상기 적어도 하나의 명령은 상기 프로세서가:

가상 머신을 서버 상에서 운용하고 가상 네트워크 기능(VNF: Virtual Network Functions)을 가상 머신에서 운용하여 사용자(client)에게 네트워크 기능을 제공하는 단계 (a);

가상 머신 및 가상 머신을 동작시키는 서버로부터 발생하는 로그를 수집하는 단계 (b);

수집한 로그를 기계 학습의 입력인 로그 임베딩 행렬(log embedding matrix)로 변환하는 단계 (c); 및

변환된 행렬을 기초로 가상 머신 및 서버의 고장을 예측하는 단계 (d); 를 수행하도록 구성되는,

로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 장치.

청구항 11

(a) 가상 머신을 서버 상에서 운용하고 가상 네트워크 기능(VNF: Virtual Network Functions)을 가상 머신에서 운용하여 사용자(client)에게 네트워크 기능을 제공하는 단계;

(b) 가상 머신 및 가상 머신을 동작시키는 서버로부터 발생하는 로그를 수집하는 단계;

(c) 수집한 로그를 기계학습의 입력인 로그 임베딩 행렬(log embedding matrix)로 변환하는 단계; 및

(d) 변환된 행렬을 기초로 가상 머신 및 서버의 고장을 예측하는 단계; 를 포함하는,

로그 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법.

청구항 12

청구항 11에 있어서, 로그 임베딩 행렬은,

가상 머신과 서버에서 수집한 로그를 통해 미리 학습된 워드 임베딩 모델 (Word embedding model)을 통해 생성되는,

로그 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법.

청구항 13

청구항 11에 있어서, 가상 머신 및 서버의 고장을 예측하는 단계는,

미리 학습된 합성곱 신경망을 통해 수행되는,

로그 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법.

청구항 14

청구항 13에 있어서, 합성곱 신경망은,

로스 함수(loss function, 목적 함수)인 수학식 1에 따르는,

로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법.

(수학식 1)

$$loss = y_{true} \times \log\left(\frac{y_{true}}{y_{pred}}\right) \times Class\ Weight_1 + (1 - y_{true}) \times \log\left(\frac{1 - y_{true}}{1 - y_{pred}}\right) \times Class\ Weight_0$$

단, 우변의 각 항들은 y_{true} 는 정답 출력값,

y_{pred} 는 합성곱 신경망에서 예측한 결과,

$Class\ Weight_0$ 과 $Class\ Weight_1$ 은 각각 0과 1에 대한 클래스 가중치(class weigh)임.

청구항 15

청구항 14에 있어서, 합성곱 신경망은,

로스 함수에 클래스 가중치 방식을 적용한 수학식 2를 따르는,

로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법.

(수학식 2)

$$Class\ Weigh_0 = \frac{Total\ Num\ Samples}{Num\ Samples\ In\ Class_0}, Class\ Weigh_1 = \frac{Total\ Num\ Samples}{Num\ Samples\ In\ Class_1}$$

단, $Total\ Num\ Samples$ 는 전체 학습용 데이터 수,

$Num\ Samples\ In\ Class_1$ 과 $Num\ Samples\ In\ Class_0$ 는 각각 전체 학습용 데이터 수 중 출력값이 1로 태깅된 데이터의 수(고장 데이터의 수), 그 외의 데이터의 수(정상 데이터의 수)임.

청구항 16

청구항 13에 있어서, 상기 방법은,

인공 신경망의 학습 과정에 사용할 데이터의 생성 과정에서는 고장 전 태깅(pre-failure tagging) 기법을 사용하는,

로그 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법.

청구항 17

청구항 13에 있어서, 상기 방법은,

가상 머신 및 서버의 고장을 감지하는 경우, 인공 신경망이 로그 임베딩 행렬을 기초로 가상 머신 및 서버에 대한 점수를 산정하여 산정된 점수를 설정 임계값과 비교하도록 수행되고, 고장 발생 전에 조치를 취하는 단계를 더 포함하는,

로그 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법.

청구항 18

청구항 17에 있어서, 고장 발생 전에 조치를 취하는 단계는,

고장의 처리 방법을 결정하는 경우, 고장 발생이 예측되는 가상 머신을 새로운 가상 머신으로 이전(VM migration)하거나 고장 발생이 예측되는 서버에서 동작하는 가상 머신들을 새로운 서버로 이전하도록 수행되는, 로그 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법.

청구항 19

청구항 13 내지 청구항 18 중 어느 한 항의 로그 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법을 구현하기 위한 컴퓨터 판독 가능한 기록매체에 저장된 컴퓨터 프로그램.

청구항 20

청구항 13 내지 청구항 18 중 어느 한 항의 로그 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법의 프로그램을 구현하기 위한 컴퓨터 판독 가능한 기록매체.

발명의 설명

기술분야

[0001] 본 발명은 네트워크의 서버 고장 예측 방법, 장치 및 시스템에 관한 것으로, 상세하게는 소프트웨어 정의 네트워크 및 네트워크 기능 가상화 환경에서 가상 머신과 서버의 고장을 예측하는 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법, 장치 및 시스템에 관한 것이다.

배경기술

[0002] 소프트웨어 정의 네트워크(SDN: Software-Defined Network)는 네트워크 장치의 제어 평면(Control plane)과 데이터 평면(Data plane)을 분리하고, 복잡한 네트워크 제어 기능을 범용 컴퓨터에 소프트웨어로 구현하여 중앙 집중 방식으로 다수의 데이터 평면을 제어하도록 하는 기술이다. 소프트웨어 정의 네트워크를 네트워크 기능 가상화(NFV: Network Function Virtualization) 기술과 함께 사용하면 값비싼 미들박스(Middle box) 장비를 가상 서버 상에 소프트웨어로 구현하여 운용비용과 투자비용을 절감하고, 새로운 기능을 구현하는 시간을 단축할 수도 있다는 장점을 가진다. 이러한 기술은 5G 코어 네트워크와 공중 및 사설 클라우드 데이터 센터(Cloud Data center)를 비롯한 다양한 환경에서 활발히 적용되고 있다.

[0003] 그러나, 이러한 소프트웨어 정의 네트워크(SDN: Software-Defined Network)와 네트워크 기능 가상화(NFV: Network Function Virtualization)기술의 도입은 네트워크 구조를 더 복잡하게 만들었으며, 네트워크와 서버에 가상화 기술이 도입됨에 따라 네트워크와 서버의 장애를 진단하고 관리하는 것이 더 어려워지고 있다. 특히 소프트웨어 정의 네트워크(SDN: Software-Defined Network)와 네트워크 기능 가상화(NFV: Network Function Virtualization) 환경에서 실제 사용자 응용 서비스를 처리하는 서버 및 가상 머신(VM: Virtual Machine)의 고장은 즉시 대응하지 못하면 서비스 가용도(availability)의 저하로 이어져 고객의 불만을 초래한다는 문제를 안고 있다. 기존의 네트워크와 서버 관리 기술은 고장을 실시간으로 탐지하고 사후 조치를 빠르게 하는 것에 초점을 맞추고 있었지만, 최근 5G 서비스 및 고가용도를 요구하는 서비스(예: 금융 서비스)의 확산에 따라 품질 보장을 위해 고장을 미리 예측하고 사전 조치(예: 서버 이전: server migration)하는 방향으로 연구개발이 진행되는 추세에 있다.

[0004] 고장 예측은 서버가 상시 만들어 내는 다양한 로그(log) 데이터를 분석하여 고장의 징후를 찾아내는 방식으로 수행할 수 있지만 가상 머신과 서버가 출력하는 로그의 양이 매우 많기 때문에 네트워크 관리자가 이를 직접 분석하여 예측하기에는 어려움이 따른다.

선행기술문헌

비특허문헌

[0005] (비특허문헌 0001) Sukhyun Nam, Jibum Hong, Jae-Hyoung Yoo, James Won-Ki Hong, "Virtual Machine Failure Prediction using Log Analysis," The 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS 2021), Tainan, Taiwan, Sep. 8-10, 2021.

발명의 내용

해결하려는 과제

[0006] 상기와 같은 문제점을 해결하기 위한 본 발명의 목적은 합성곱 신경망(Convolution Neural Network, CNN)을 이용하여 서버 및 가상 머신이 출력하는 로그를 자동으로 분석하고 학습하여 서버 및 가상 머신의 고장을 미리 예측하는 방법을 제공하기 위한 것이다.

과제의 해결 수단

[0007] 상기 목적을 달성하기 위한 본 발명의 일 실시예에 따른 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 시스템은, 클라우드 컴퓨팅 관리 시스템을 이용하여 구축하는 네트워크 기능 가상화 인프라(NFVI: Network Function Virtualization Infrastructure)를 사용하여 구축되는 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 시스템에 있어서, 결정된 네트워크 관리 정책(policy)을 컴퓨트 노드(Compute node)로 전달하고, 가상 네트워크 기능(VNF: Virtual Network Functions)들을 별개의 가상 머신 상에서 동작시켜 데이터를 수집하여, 가상 머신 및 가상 머신을 동작시키는 서버로부터 발생하는 로그를 수집하고 전처리 과정을 통해 로그 테이블을 생성하는 네트워크 기능 가상화(NFV: Network Function Virtualization) 구현부; 및 합성곱 신경망 모델의 출력값이 사전에 정의한 임계치를 넘기면 일정 시간 뒤에 해당 가상 머신 혹은 서버에서 고장이 발생할 것에 대해 네트워크 기능 가상화(NFV: Network Function Virtualization) 구현부에 경고를 보내는 인공지능 노드; 를 포함할 수 있다.

[0008] 네트워크 기능 가상화(NFV: Network Function Virtualization) 구현부는, 결정된 네트워크 관리 정책(policy)을 컴퓨트 노드(Compute node)로 전달하여 기존 가상 네트워크 기능(VNF: Virtual Network Functions) 또는 서비스 기능 체인(SFC: Service Function Chaining)의 구성 및 배치를 변화시키는 컨트롤러 노드(Controller node); 가상 네트워크 기능(VNF: Virtual Network Functions)들을 별개의 가상 머신 상에서 동작시키는 컴퓨트 노드(Compute node); 및 네트워크 기능 가상화(NFV: Network Function Virtualization) 환경의 컴퓨트 노드(Compute node)에서 수집 가능한 데이터를 모니터링하는 모니터링 노드(Monitoring node); 를 포함할 수 있다.

[0009] 컴퓨트 노드(Compute node)는, 가상 네트워크 기능(VNF: Virtual Network Functions)들을 이용하여 서비스 기능 체인(SFC: Service Function Chaining)을 배치하고, 이를 통해 사용자에게 네트워크 기능 가상화(NFV: Network Function Virtualization) 기반의 네트워크 기능을 제공할 수 있다.

[0010] 모니터링 노드(Monitoring node)는, 고장 예측을 위하여 컴퓨트 노드(Compute node)에서 동작하는 각 가상 머신들과 가상 머신을 구동하는 서버들에서 발생하는 로그 데이터를 수집하고, 로그 수집을 위하여 로그 수집 유틸리티인 rsyslog(The rocket-fast syslog server)를 활용하여 rsyslog 서버를 동작시키고, 컴퓨트 노드(Compute node)의 각 가상 머신에서 rsyslog를 동작시키고 로그를 rsyslog 서버로 전송하도록 설정하여 각 가상 머신에서 발생하는 로그를 rsyslog 서버를 통해 수집할 수 있다.

[0011] 모니터링 노드(Monitoring node)는, 가상 머신에서 동작하는 가상 네트워크 기능(VNF: Virtual Network Functions)에서 출력시키는 애플리케이션 로그(application log)뿐만 아니라 가상 머신에서 발생하는 시스템 로그(system log), 커널 로그(kernel log)를 수집하고, 가상 머신들을 구동시키는 서버에서도 rsyslog 설정을 통해 모든 로그를 모니터링 노드로 전송시키고, 수집된 로그에 전처리 과정을 거친 후 로그 테이블에 저장할 수 있다.

[0012] 인공지능 노드는, 로그를 고장 예측 모듈의 입력인 로그 임베딩으로 변화시키는 모듈인 로그 파싱 모듈; 및 합성곱 신경망의 학습으로 사전에 정의된 시간차(gap) 뒤에 고장이 발생할지를 예측하는 고장 예측 모듈; 을 포함할 수 있다.

[0013] 로그 파싱 모듈은, 모니터링 노드로부터 타임스탬프를 기준으로 사전에 정해진 윈도우 크기(window size) 동안의 로그 데이터를 로그 테이블에서 추출한 데이터인 로그 데이터를 전달받고, 네트워크 기능 가상화 인프라(NFVI: Network Function Virtualization Infrastructure)에서 사전에 수집한 대량의 로그 데이터를 워드투벡터(word2vec) 라이브러리의 입력데이터로 사용하여 사전에 생성되어 학습된 워드 임베딩 모델을 포함할 수 있다.

[0014] 고장 예측 모듈은, 수학적 1의 로스 함수(loss function, 목적 함수)에 따르는 합성곱 신경망 모델을 포함할 수 있다.

- [0015] 고장 예측 모듈은, 로스 함수에 수학적 2의 클래스 가중치 방식을 적용한 합성곱 신경망 모델을 포함할 수 있다.
- [0016] 본 발명의 다른 목적을 달성하기 위한 클라우드 컴퓨팅 관리 시스템을 이용하여 구축하는 네트워크 기능 가상화 인프라(NFVI: Network Function Virtualization Infrastructure)를 사용하여 구축되는 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 장치에 있어서, 프로세서(processor); 및 프로세서를 통해 실행되는 적어도 하나의 명령이 저장된 메모리(memory); 를 포함하고, 상기 적어도 하나의 명령은 상기 프로세서가: 가상 머신을 서버 상에서 운용하고 가상 네트워크 기능(VNF: Virtual Network Functions)을 가상 머신에서 운용하여 사용자(client)에게 네트워크 기능을 제공하는 단계 (a); 가상 머신 및 가상 머신을 동작시키는 서버로부터 발생하는 로그를 수집하는 단계 (b); 수집한 로그를 기계 학습의 입력인 로그 임베딩 행렬(log embedding matrix)로 변환하는 단계 (c); 및 변환된 행렬을 기초로 가상 머신 및 서버의 고장을 예측하는 단계 (d); 를 수행하도록 구성될 수 있다.
- [0017] 본 발명의 또 다른 목적을 달성하기 위한 본 발명의 일 실시예에 따른 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법은, (a) 가상 머신을 서버 상에서 운용하고 가상 네트워크 기능(VNF: Virtual Network Functions)을 가상 머신에서 운용하여 사용자(client)에게 네트워크 기능을 제공하는 단계; (b) 가상 머신 및 가상 머신을 동작시키는 서버로부터 발생하는 로그를 수집하는 단계; (c) 수집한 로그를 기계 학습의 입력인 로그 임베딩 행렬(log embedding matrix)로 변환하는 단계; 및 (d) 변환된 행렬을 기초로 가상 머신 및 서버의 고장을 예측하는 단계; 를 포함한다.
- [0018] 로그 임베딩 행렬은, 가상 머신과 서버에서 수집한 로그를 통해 미리 학습된 워드 임베딩 모델(Word embedding model)을 통해 생성될 수 있다.
- [0019] 가상 머신 및 서버의 고장을 예측하는 단계는, 미리 학습된 합성곱 신경망을 통해 수행될 수 있다.
- [0020] 합성곱 신경망은, 로스 함수(loss function, 목적 함수)인 수학적 1에 따를 수 있다.
- [0021] 합성곱 신경망은, 로스 함수에 클래스 가중치 방식을 적용한 수학적 2를 따를 수 있다.
- [0022] 인공 신경망의 학습 과정에 사용할 데이터의 생성 과정에서는 고장 전 태깅(pre-failure tagging) 기법을 사용할 수 있다.
- [0023] 상기 방법은, 가상 머신 및 서버의 고장을 감지하는 경우, 인공 신경망이 로그 임베딩 행렬을 기초로 가상 머신 및 서버에 대한 점수를 산정하여 산정된 점수를 설정 임계값과 비교하도록 수행되고, 고장 발생 전에 조치를 취하는 단계; 를 더 포함할 수 있다.
- [0024] 고장 발생 전에 조치를 취하는 단계는, 고장의 처리 방법을 결정하는 경우, 고장 발생이 예측되는 가상 머신을 새로운 가상 머신으로 이전(VM migration)하거나 고장 발생이 예측되는 서버에서 동작하는 가상 머신들을 새로운 서버로 이전하도록 수행될 수 있다.
- [0025] 전술한 항 중 어느 한 항의 로그 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법을 구현하기 위한 컴퓨터 판독 가능한 기록매체에 저장된 컴퓨터 프로그램일 수 있다.
- [0026] 전술한 항 중 어느 한 항의 로그 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법의 프로그램을 구현하기 위한 컴퓨터 판독 가능한 기록매체일 수 있다.

발명의 효과

- [0027] 본 발명의 일 실시예에 따르면, 서버 및 가상 머신에 고장이 발생하기 전에 고장을 예측하여 가상 머신 및 서버의 고장을 사전에 방지할 수 있는 효과가 있다.
- [0028] 본 발명에서는 자연 언어 처리(NLP: Natural Language Process) 분야의 워드 임베딩과 합성곱 신경망을 통해 서버 및 가상 머신에서 발생하는 대량의 로그 메시지를 사람이 분석하지 않고 자동으로 에러와 관련된 메시지를 파악할 수 있다.
- [0029] 또한, 본 발명에서 제안하는 고장 전 태깅 기법을 통해 기계 학습의 정확도를 더 높일 수 있다는 장점이 있다.

도면의 간단한 설명

- [0030] 도 1은 본 발명의 일 실시예의 네트워크 기능 가상화 환경에서 로그 데이터를 실시간으로 분석하여 고장을 예측

하는 시스템 구조를 설명하는 구성도이다.

도 2는 본 발명의 일 실시예의 모니터링 노드(130)에서 로그에 수행하는 전처리 과정의 예시를 나타낸 모식도이다.

도 3은 본 발명의 일 실시예의 인공지능 노드(200)의 로그 파싱 모듈에서 로그 데이터를 입력으로 받아 로그 임베딩을 생성하는 과정을 설명하기 위한 모식도이다.

도 4는 본 발명의 일 실시예의 인공지능 노드(200)의 고장 예측 모듈(220)에서 사용하는 합성곱 신경망의 생성 및 학습 과정을 나타내는 순서도이다.

도 5는 본 발명의 일 실시예의 합성곱 신경망의 입력 및 출력을 나타내는 모식도이다.

도 6은 본 발명의 일 실시예의 합성곱 신경망의 출력 데이터 생성을 위한 상태 확인 과정을 설명하기 위한 서버들의 구성도이다.

도 7은 본 발명의 일 실시예의 합성곱 신경망의 출력 태깅에 사용하는 고장 전 태깅 기법의 예시를 나타내는 모식도이다.

도 8은 본 발명의 일 실시예의 고장 예측 모듈(220)에서 사용하는 합성곱 신경망의 구조를 나타내는 모식도이다.

도 9는 본 발명의 일 실시예의 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법의 순서도이다.

도 10은 본 발명의 일 실시예의 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 장치(1000)의 구성도이다.

발명을 실시하기 위한 구체적인 내용

- [0031] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세한 설명에 상세하게 설명하고자 한다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다. 각 도면을 설명하면서 유사한 참조부호를 유사한 구성요소에 대해 사용하였다.
- [0032] 제1, 제2, A, B 등의 용어는 다양한 구성요소들을 설명하는 데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. "및/또는"이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.
- [0033] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다.
- [0034] 본 출원에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, "포함하다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0035] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가지고 있다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥 상 가지는 의미와 일치하는 의미를 가지는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

- [0037] 이하, 본 발명에 따른 바람직한 실시예를 첨부된 도면을 참조하여 상세하게 설명한다.
- [0039] 도 1은 본 발명의 일 실시예의 네트워크 기능 가상화 환경에서 로그 데이터를 실시간으로 분석하여 고장을 예측하는 시스템 구조를 설명하는 구성도이다.
- [0040] 도 1을 참조하면, 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 시스템은 클라우드 컴퓨팅 관리 시스템을 이용하여 구축하는 네트워크 기능 가상화 인프라(NFVI: Network Function Virtualization Infrastructure)를 사용하여 구축하며, 로그 메시지 분석을 통한 가상 머신 및 서버의 네트워크 기능 가상화(NFV: Network Function Virtualization) 구현부(100)와 인공지능 노드(200)를 포함한다.
- [0041] 로그 메시지 분석을 통한 가상 머신 및 서버의 네트워크 기능 가상화(NFV: Network Function Virtualization) 구현부(100)는 컨트롤러 노드(Controller node)(110), 컴퓨트 노드(Compute node)(120) 및 모니터링 노드(Monitoring node)(130)를 포함한다.
- [0042] 컨트롤러 노드(Controller node)(110)에서는 결정된 네트워크 관리 정책(policy)을 컴퓨트 노드(Compute node)로 전달하여 기존 가상 네트워크 기능(VNF: Virtual Network Functions) 또는 서비스 기능 체인(SFC: Service Function Chaining)의 구성 및 배치를 변화시킨다.
- [0043] 컴퓨트 노드(Compute node)(120)에서는 가상 네트워크 기능(VNF: Virtual Network Functions)들을 별개의 가상 머신 상에서 동작시킨다. 또한, 가상 네트워크 기능(VNF: Virtual Network Functions)들을 이용하여 서비스 기능 체인(SFC: Service Function Chaining)을 배치하고, 이를 통해 사용자에게 네트워크 기능 가상화(NFV: Network Function Virtualization) 기반의 네트워크 기능을 제공한다.
- [0044] 모니터링 노드(Monitoring node)(130)는 네트워크 기능 가상화(NFV: Network Function Virtualization) 환경의 컴퓨트 노드(Compute node)에서 수집 가능한 데이터를 모니터링하는 노드이다. 본 발명에서는 모니터링 노드(130)는 고장 예측을 위하여 컴퓨트 노드(Compute node)(120)에서 동작하는 각 가상 머신들과 가상 머신을 구동하는 서버들에서 발생하는 로그 데이터를 수집한다. 모니터링 노드(130)는 로그 수집을 위하여 로그 수집 유틸리티인 rsyslog(The rocket-fast syslog server)를 활용하여 rsyslog 서버를 동작시킨다. 컴퓨트 노드(Compute node)(120)의 각 가상 머신에서 rsyslog를 동작시키고 로그를 rsyslog 서버로 전송하도록 설정하여 각 가상 머신에서 발생하는 로그를 모니터링 노드의 rsyslog 서버를 통해 수집한다. 가상 머신에서 동작하는 가상 네트워크 기능(VNF: Virtual Network Functions)에서 출력시키는 애플리케이션 로그(application log) 뿐 아니라 가상 머신에서 발생하는 시스템 로그(system log), 커널 로그(kernel log) 등을 모두 수집한다. 가상 머신들을 구동시키는 서버에서도 마찬가지로 rsyslog 설정을 통해 모든 로그를 모니터링 노드로 전송시키도록 한다. 모니터링 노드는 수집된 로그에 전처리 과정을 거친 후 로그 테이블에 저장한다.
- [0045]
- [0046] 도 2는 본 발명의 일 실시예의 모니터링 노드(130)에서 로그에 수행하는 전처리 과정의 예시를 나타낸 모식도이다.
- [0047] 도 2를 참조하면, 모니터링 노드(130)는 시각 정보를 읽어 들어 로그에 대한 타임스탬프(timestamp)로 변환하여 저장한다. 또한, 가상 머신 이름을 추출하여 정보 분류에 활용하며, 그 외의 로그는 숫자 및 기호, 애플리케이션 이름을 제거한 후 모든 대문자는 소문자로 바뀌어 저장한다. 이러한 정보 추출은 규칙 기반(rule-based)으로 진행한다. 도2의 예시에서는 공백을 기준으로 3번째 공백 이전은 시각 정보로, 3번째 공백과 4번째 공백 사이는 가상 머신의 이름으로, 기호 ':'을 기준으로 5번째 ':' 이후의 로그를 최종 로그로 추출할 수 있으며, 이러한 로그 생성 규칙은 로그를 출력하는 시스템에 따라 상이하기 때문에 고장 예측 시스템 설치 전 설정하는 단계가 필요하다. 모니터링 노드(130)에서는 로그 데이터를 가상 머신 별로 가상 머신 로그 테이블의 형태로 저장하며, 서버와 관련된 로그 또한 서버 별 서버 로그 테이블에 저장한다.
- [0048] 다시 도 1로 돌아가서, 도 1의 인공지능 노드(AI node: Artificial Intelligence node)(200)는 모니터링 노드(130)로부터 로그 데이터를 전달받는다. 이때 전달되는 로그 데이터는 타임스탬프를 기준으로 사전에 정해진 윈도우 크기(window size) 동안의 로그 데이터를 로그 테이블에서 추출한 데이터이다. 인공지능 노드(200)는 로그 파싱 모듈(210)과 고장 예측 모듈(220)을 포함한다. 로그 파싱 모듈(210)은 로그를 고장 예측 모듈(220)의 입력인 로그 임베딩으로 변화시키는 모듈로서, 사전에 학습된 워드 임베딩 모델(215)을 포함한다.
- [0049] 워드 임베딩은 자연 언어 처리에서 사용하는 기술로서, 문장 내에서 단어 사이의 거리를 기반으로 각 단어의 유

사도를 파악하여 단어를 벡터(vector)로 변환시키도록 학습하는 기술이다. 생성되는 벡터의 차원(size)의 경우 사용자가 일정한 크기로 사전에 설정할 수 있다. 워드 임베딩은 말뭉치(corpus) 속의 각 단어 별로 초기화된 같은 크기의 벡터들을 생성한 후 학습을 진행한다. 학습 과정에서는 인공신경망이 각 단어의 벡터를 입력으로 받아 문장 내에서 단어를 예측하고 예측 결과의 정확도를 기반으로 각 단어 벡터를 수정하며 문장 내의 단어를 잘 예측할 수 있도록 단어 벡터를 만들어나간다.

[0050] 본 발명에서 사용하는 워드투벡터(Word2Vec)는 구글(Google)에서 개발한 파이썬(Python) 오픈소스 라이브러리(open-source library)로서, 말뭉치를 입력으로 넣어 워드 임베딩 모델(215)을 생성할 수 있다. 워드투벡터의 경우, 학습 과정에서 주변에 있는 단어들을 가지고 중간에 있는 단어를 예측하며 학습하는 CBOW(Continuous Bag of Words) 방법과 중심 단어를 통해 주변 단어를 예측하며 학습하는 스킵 그램(Skip-gram) 방법중 사용자가 선택할 수 있다.

[0051] 본 발명에서는 네트워크 기능 가상화 인프라(NFVI: Network Function Virtualization Infrastructure)에서 사전에 수집한 대량의 로그 데이터를 워드투벡터 라이브러리의 입력데이터로 사용하여 워드 임베딩 모델(Word embedding model)(215)을 사전에 생성한다. 워드투벡터를 사용할 때 학습 방식, 벡터 차원, 최소 언급 횟수, 학습 윈도우 크기를 설정하여야 한다. 본 발명에서는 가장 일반적으로 사용하는 설정인 학습 방식으로 스킵 그램 방식을, 벡터 차원으로는 100을, 학습 윈도우 크기로는 5를 사용하였다. 또한 로그에서 프로세스 ID와 같이 필요 없는 단어는 워드 임베딩 모델(215)에서 제외되도록 하기 위해 말뭉치에서 3번 이상 언급된 단어에 대해서만 워드 임베딩이 생성되도록 최소 언급 횟수를 3으로 설정한다. 생성된 워드 임베딩 모델(215)은 각 단어들에 대한 100차원의 벡터를 저장하고 있으며, 워드투벡터 라이브러리를 통해 읽어 들일 수 있는 형태의 파일로서 저장된다. 로그 파싱 모듈(210)은 워드임베딩 모델을 읽어 들여 입력 데이터의 각 단어들에 100차원의 단어 벡터를 매칭 시킨 후 합침으로서(concatenate) 입력 로그를 행렬(matrix)로 변환시킬 수 있다. 이 때, 생성된 행렬은 행 수가 입력 로그의 단어 개수이며 열 수가 단어 벡터의 차원인 100개인 행렬이며, 로그 데이터의 특성을 나타내는 행렬이다. 행 수가 입력 로그의 단어 수이기 때문에 로그 임베딩 행렬은 매번 그 크기가 변화하지만, 이는 후술할 합성곱 신경망의 맥스 풀링 층을 거치며 일정한 크기의 벡터로 변환된다.

[0052]

[0053] 도 3은 본 발명의 일 실시예의 인공지능 노드(200)의 로그 파싱 모듈에서 로그 데이터를 입력으로 받아 로그 임베딩을 생성하는 과정을 설명하기 위한 모식도이다.

[0054] 도 3을 참조하면, 모니터링 노드(130)로부터 윈도우 크기 동안의 로그 데이터가 전달되면 각 단어별로 해당되는 벡터를 매칭 시켜 로그 임베딩을 생성하는 것을 나타낸다. 이 때, 워드 임베딩 모델(215)의 학습과정에서 말뭉치에서 3번 이상 언급된 단어에 대하여만 벡터를 생성하였기 때문에 로그 중에 워드 임베딩 모델(215)에 포함되지 않은 단어가 존재할 수 있다. 이러한 경우에는 'Unknown'을 의미하는 '<UNK>'로 태그되는데, 자연 언어 처리 분야에서는 일반적으로 <UNK>를 없는 단어로 취급하거나, 미리 정의해둔 방법에 따라 <UNK>에 해당하는 벡터를 생성하여 매칭시킨다. 로그 데이터에서 워드 임베딩에 없는 단어는 크게 두 가지 원인에 의한 것으로, 첫 번째는 도3의 'U0ixfW'와 같이 뜻이 없으며 프로세스 ID 등의 시스템과 관련되어 매번 변하여서 워드 임베딩 학습용 말뭉치에는 없는 단어이고, 두 번째는 뜻이 존재하는 단어이지만 워드 임베딩 학습에 존재하지 않았던 로그가 발생한 경우로서, 평소에 잘 발생하지 않는 로그가 발생한 경우이다. 만약 첫 번째 경우만 존재한다면 <UNK>에 대해 시스템과 관련된 단어임을 뜻하도록 사전에 정의해둔 벡터를 태깅하거나 없는 단어로 취급하면 되지만, 이러한 경우 두 번째 경우에 해당하는 로그를 분석할 때 대해서 시스템과 관련된 ID의 나열로 인식되거나 로그가 사라지기 때문에 부적절하다. 본 발명에서는 <UNK>를 매번 100차원의 임의 벡터(random vector)를 매칭시킨다. <UNK>에 매칭하는 벡터는 Out-of-Vector (OOV)로 표기한다.

[0055] 생성된 로그 임베딩 행렬은 사전에 학습된 고장 예측 모듈(220)의 입력으로써 사용되어 합성곱 신경망이 사전에 정의된 시간차(gap) 뒤에 고장이 발생할지를 출력한다. 고장 예측에 사용되는 윈도우 크기와 시간차는 서버의 구동 목적에 맞도록 적절한 값을 사전에 설정할 수 있으며, 단위로는 분(minutes)을 사용한다.

[0057] 도 4는 본 발명의 일 실시예의 인공지능 노드(200)의 고장 예측 모듈(220)에서 사용하는 합성곱 신경망의 생성 및 학습 과정을 나타내는 순서도이다.

[0058] 도 4를 참조하면, 해당 과정은 본 발명에서 제안하는 고장 예측 시스템의 생성 과정에서 기계 학습 모델의 최적화를 위하여 사전에 거치는 과정이며, 예측 시스템 생성 후 실제 활용 과정에는 사용하지 않는다. 도 4에서 임

력값과 출력값의 자세한 설명은 도 5에 자세히 나타나 있고, 고장 기록 데이터의 생성 과정은 도 6에서 자세히 나타나 있다.

- [0059]
- [0060] 도 5는 본 발명의 일 실시예의 합성곱 신경망의 입력 및 출력을 나타내는 모식도이다.
- [0061] 도 5를 참조하면, 합성곱 신경망은 윈도우 크기로 생성된 입력 윈도우 동안의 로그 데이터와 입력 윈도우로부터 일정한 시간차 이후의 고장 여부를 나타내는 0에서 1 사이의 값을 각각 입력과 출력으로 사용한다. 입력 윈도우와 예측 대상은 1분 단위로 로그 테이블 및 고장 기록 데이터를 이동하며 학습용 데이터를 생성한다. 학습용 데이터 수집 단계에 모니터링 노드는 가상 머신 및 서버 별로 로그 테이블과 고장 기록 데이터를 수집해야 한다. 로그 테이블은 고장 예측 시스템의 운용 과정에서와 마찬가지로 수집하며, 고장 기록 데이터 수집을 위한 고장 태깅 과정을 추가로 가져야 한다.
- [0062]
- [0063] 도 6은 본 발명의 일 실시예의 합성곱 신경망의 출력 데이터 생성을 위한 상태 확인 과정을 설명하기 위한 서버들의 구성도이다.
- [0064] 도 6을 참조하면, 모니터링 노드(130)는 데이터 수집 단계에 타임스탬프가 존재하는 로그 데이터와 타임스탬프가 존재하는 고장 기록 데이터를 저장한다. 고장 여부 확인을 위하여 데이터 수집 단계에서는 컴퓨트 노드(Compute node)(120)의 각 서버별로 상태 확인 가상 머신(상태 확인 VM)을 운용한다. 상태 확인 VM은 주기적으로 같은 서버 내에서 가상 네트워크 기능(VNF: Virtual Network Functions)을 동작시키는 가상 머신들에게 핑(ping) 메시지를 보내고 응답을 받아 고장을 판단한다. 핑 메시지에 대한 응답이 없는 가상 머신을 고장 상태로 판단하고 고장이 발생한 시각을 기록한다. 모니터링 노드(130)는 상태확인 VM으로부터 고장 여부를 전달받아 각 가상 머신에 대한 고장 기록 데이터를 생성한다. 고장이 발생한 가상 머신이 추후 핑 메시지에 대해 응답이 다시 발생하면 고장 상태가 끝나고 정상 상태로 태깅된다.
- [0065] 모니터링 노드(130)는 동일한 서버의 가상 머신들이 동시다발적으로 고장을 일으키는 경우 해당 서버를 고장 상태로 태깅한다. 동시다발적 고장의 기준은 사전에 설정한다. 예를 들어, 같은 서버의 가상 머신 3개가 1분 이내의 차이를 두고 동시에 고장 상태가 되면 서버의 고장으로 태깅한다. 이러한 경우에는 가상 머신의 고장 원인이 서버이기 때문에 고장과 관련된 로그는 가상 머신 로그가 아닌 서버 로그에 존재할 것이므로, 고장을 일으킨 가상 머신은 정상 상태로 태깅한다.
- [0066] 가상 머신 및 서버 상태 확인 과정은 이후 고장 예측 시스템을 포함한 네트워크의 운용 단계에서는 사용하지 않으며, 상태확인 VM 또한 운용단계에는 사용하지 않는다. 또한 학습용 데이터의 수집 단계에서는 일반적인 상황에서 고장 상태가 잘 발생하지 않기 때문에 기계 학습의 성능이 떨어질 수도 있으므로 학습에 적합한 데이터 생성을 위하여 정상 트래픽을 생성하면서 동시에 고장 상태 유발을 위한 악의적 공격 트래픽을 생성하며 데이터를 생성하여 고장 예측 모듈(220)의 성능을 높인다. 데이터 수집 단계가 길어지면 추후 고장 예측 시스템의 성능이 높아질 것이며, 적절한 성능의 고장 예측 시스템을 위해서는 약 한 달 이상의 학습용 데이터 수집 단계가 필요하다.
- [0067] 다시 도 4로 돌아가서, 모니터링 노드(130)는 타임스탬프를 기준으로 정해진 윈도우 크기 동안의 로그 데이터를 입력 데이터로, 정해진 시간차 이후의 고장 여부를 출력 데이터로 생성하여 도 5의 입력과 출력에 해당하는 데이터를 합성곱 신경망 모델(225)의 학습에 사용되도록 전달시킨다. 이때, 고장 여부는 고장이 난 경우 1을, 고장이 나지 않은 정상 상태는 0으로 변환시켜서 사용한다. 이 때, 고장이 나지 않은 정상 상태 중에서도 일부는 고장 전 태깅 기법에 따라 0이 아닌 숫자가 태깅된다.
- [0069] 도 7은 본 발명의 일 실시예의 합성곱 신경망의 출력 태깅에 사용하는 고장 전 태깅 기법의 예시를 나타내는 모식도이다.
- [0070] 도 7을 참조하면, 이는 입력 데이터의 순서에 상관없이 가장 중요한 정보를 학습하는 합성곱 신경망의 특성에 따라 기계 학습 모델의 성능을 높이기 위해 제안하는 기법이다. 이는 고장이 발생하기 전 고장 전 태깅 크기(pre-failure tagging size) 동안의 윈도우는 일정 시간차 이후의 상태가 고장 상태가 아니더라도 고장 전 상태를 나타내는 고장 전 태깅 값(pre-failure tagging value)을 출력값으로 갖도록 하는 기법이다.

[0071] 도 7은 고장 전 태깅 크기가 4인 예시로서, 윈도우 5의 출력값은 고장 상태를 의미하는 1이 되지만, 윈도우 1부터 윈도우 4까지의 4개의 윈도우는 시간차 이후가 정상 상태임에도 출력값이 0이 아닌, 고장 전 태깅 값이 된다. 이때 고장 전 태깅 값은 정상을 나타내는 0과 고장을 나타내는 1 사이의 값이다. 고장 전 태깅 값은 하나의 정해진 값으로 설정하여도 되지만, 고장에 가까울수록 값이 커지는 선형 함수 등을 활용하여서 각 윈도우 별로 다르게 설정할 수도 있다. 고장 전 태깅 기법은 여러 로그 메시지가 일정 시간차 이후의 상태가 정상인 윈도우에 포함되는 경우에도 합성곱 신경망이 해당 메시지에 대하여 학습할 수 있도록 하여 예측 성능을 높인다. 고장 전 태깅 크기와 고장 전 태깅 값은 합성곱 신경망 모델(225)의 학습 과정에서 실험을 통해 성능을 더 높일 수 있는 값으로 설정한다.

[0072]

[0073] 도 8은 본 발명의 일 실시예의 고장 예측 모듈(220)에서 사용하는 합성곱 신경망의 구조를 나타내는 모식도이다.

[0074]

도 8을 참조하면, 본 합성곱 신경망 모델(225)은 1개의 합성곱 층과 1개의 풀링 층을 갖는 단순한 모델이다. 단순한 형태여서 학습이 굉장히 빠름에도 불구하고 대량의 데이터로부터 가장 중요한 데이터를 추출할 수 있는 장점을 통해 대량의 문장으로부터 주제와 관련된 단어들을 찾아낼 수 있으며, 이 때문에 자연 언어 처리의 한 분야인 문장 분류 문제(sentence classification problem)에서 가장 많이 사용하는 기계 학습 모델이다. 해당 합성곱 신경망 모델(225)은 로그 임베딩 행렬을 입력으로 받아서 0에서 1 사이의 값을 출력한다.

[0075]

도 8의 각 층을 설명하자면, 합성곱 층은 필터 크기가 3, 4, 5의 3가지 종류의 필터를 사용하는 합성곱 층이다. 합성곱 층은 정해진 필터 크기의 필터 100개가 입력 행렬이 입력 행렬을 1칸씩 이동(striding)하면서 합성곱을 수행한다. 학습 과정에서 합성곱 신경망 모델(225)은 각 필터의 값들을 변화시키며 학습을 수행한다. 입력 로그의 양이 많아지면 필터의 종류를 더 늘려서 사용할 수 있다. 풀링층에서는 이전 층의 결과에서 가장 큰 값만 남기는 맥스 풀링(Max pooling)을 사용한다. 맥스 풀링층은 각 필터의 연산 결과별로 가장 큰 값을 남기기 때문에 필터의 개수만큼의 값을 가지는 벡터를 생성한다. 예를 들어, 필터 크기가 3, 4, 5인 경우에는 필터가 300개며, 각 필터의 연산 결과로부터 가장 높은 값들만 남겨 300차원의 벡터를 출력한다. 이 때문에 입력 데이터의 크기가 아무리 커져도 맥스 풀링 층 이후에는 동일한 크기의 벡터가 생성되어 완전 연결 계층(FC layer: Fully Connected layer)에서 문제없이 학습을 진행할 수 있다. 풀링 층 다음에는 시그모이드(Sigmoid)를 활성 함수로 갖는 완전 연결 계층(FC layer)을 통해 0에서 1 사이의 값을 출력하도록 학습한다. 합성곱 층과 맥스 풀링 층은 입력 데이터에서 중요한 값들을 추출해내도록 학습이 진행된다면, 완전 연결 계층(FC layer)에서는 맥스 풀링 층의 결과로부터 고장 여부를 계산하도록 학습된다.

수학식 1

[0077]

$$loss = y_{true} \times \log\left(\frac{y_{true}}{y_{pred}}\right) \times Class\ Weight_1 + (1 - y_{true}) \times \log\left(\frac{1 - y_{true}}{1 - y_{pred}}\right) \times Class\ Weight_0$$

[0079]

수학식 1은 합성곱 신경망의 학습에서 사용하는 로스 함수(loss function, 목적 함수)를 나타낸다. 로스 함수는 기계 학습의 학습 과정에서 줄이는 목표가 되는 값인 로스(loss)값을 계산하는 함수이다. 우변의 각 항들은 y_{true} 는 정답 출력값, y_{pred} 는 합성곱 신경망에서 예측한 결과, $Class\ Weight_0$ 과 $Class\ Weight_1$ 을 각각 0과 1에 대한 클래스 가중치(class weighth)이다. 해당 로스 함수는 본 합성곱 신경망의 학습에 사용하는 데이터의 정답 출력값이 0과 1뿐 아니라 0과 1사이의 값을 가지고 있기 때문에 0과 1 사이의 값을 갖는 y_{true} 와 y_{pred} 의 차이를 측정하기 위해 두 확률 분포의 차이를 측정할 때 사용하는 쿨백-라이블러 발산(KL divergence: Kullback-Leibler divergence) 함수에 기반하여 만든 함수이다. 해당 로스 함수는 y_{true} 와 y_{pred} 의 값이 동일하면 0이 되고, y_{pred} 가 y_{true} 로부터 멀어질수록 더 커진다. 또한 고장 예측 실험 수집 데이터의 특성상, 학습용 데이터는 클래스가 불균형(imbalanced)하게 수집된다. 이 때문에 대부분의 학습용 데이터가 y_{true} 값이 0을 나타내며, y_{true} 값이 1일 때

이에 대해 민감하게 학습이 진행되도록 하지 않으면 대부분의 예측을 0으로 하여 고장 예측의 성능이 낮아진다. 이 때문에 데이터의 수에 반비례하게 1과 0에 대해 가중치를 주는 클래스 가중치 방식을 로스 함수에 적용시키며, $Class\ Weight_0$ 과 $Class\ Weight_1$ 을 다음 수학적 식 2에 따라 설정한다.

[0080]

수학적 식 2

$$Class\ Weight_0 = \frac{TotalNumSamples}{NumSamplesInClass0}, Class\ Weight_1 = \frac{TotalNumSamples}{NumSamplesInClass1}$$

[0081]

여기서, $TotalNumSamples$ 는 전체 학습용 데이터 수, $NumSamplesInClass1$ 과 $NumSamplesInClass0$ 는 각각 전체 학습용 데이터 수 중 출력값이 1로 태깅된 데이터의 수(고장 데이터의 수), 그 외의 데이터의 수(정상 데이터의 수)를 나타낸다. 고장 전 태깅이 된 데이터도 $NumSamplesInClass0$ 에 포함시킨다. 클래스 가중치를 통하여 y_{true} 값이 고장 상태를 나타내는 1인데 y_{pred} 가 이를 제대로 예측하지 못한 경우에 대해 로스 값이 더 크게 생성되어 y_{true} 값이 1인 경우에 대해 더 잘 학습될 수 있도록 한다.

[0083]

다시 도 1로 돌아가서, 인공지능 노드(200)는 합성곱 신경망 모델(225)의 출력값이 사전에 정의한 임계치를 넘기면 일정 시간 뒤에 해당 가상 머신 혹은 서버에서 고장이 발생할 것에 대해 컨트롤러 노드에 경고를 보낸다. 임계치는 네트워크 관리자가 설정할 수 있으며, 경고가 발생할 경우의 컨트롤러 노드의 행동 또한 네트워크 관리자가 설정한다.

[0084]

도 9는 본 발명의 일 실시예의 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법의 순서도이다.

[0086]

본 발명의 일 실시예의 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법은, 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 시스템 및 장치에 의해 수행되며 S1000 단계 내지 S4000 단계를 포함한다.

[0087]

S1000 단계는 가상 머신을 서버 상에서 운용하고 가상 네트워크 기능(VNF: Virtual Network Functions)을 가상 머신에서 운용하여 사용자(client)에게 네트워크 기능을 제공하는 단계이다.

[0088]

S2000 단계는 가상 머신 및 가상 머신을 동작시키는 서버로부터 발생하는 로그를 수집하는 단계이다.

[0089]

S3000 단계는 수집한 로그를 기계 학습의 입력인 로그 임베딩 행렬(log embedding matrix)로 변환하는 단계이다.

[0090]

S4000 단계는 변환된 행렬을 기초로 가상 머신 및 서버의 고장을 예측하는 단계이다.

[0091]

본 발명의 일 실시예의 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법에서, 로그 임베딩 행렬은 가상 머신과 서버에서 수집한 로그를 통해 미리 학습된 워드 임베딩 모델 (Word embedding model)을 통해 생성될 수 있다.

[0092]

본 발명의 일 실시예의 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법에서, 가상 머신 및 서버의 고장을 예측하는 단계는, 미리 학습된 합성곱 신경망을 통해 수행될 수 있다.

[0093]

여기서, 합성곱 신경망은 로스 함수(loss function, 목적 함수)인 수학적 식 1에 따를 수 있다.

[0094]

[0096] (수학식 1)

$$loss = y_{true} \times \log\left(\frac{y_{true}}{y_{pred}}\right) \times Class\ Weight_1 + (1 - y_{true}) \times \log\left(\frac{1 - y_{true}}{1 - y_{pred}}\right) \times Class\ Weight_0$$

[0097]

[0098] 단, 우변의 각 항들은 y_{true} 는 정답 출력값, y_{pred} 는 합성곱 신경망에서 예측한 결과, $Class\ Weight_1$ 과 $Class\ Weight_0$ 은 각각 0과 1에 대한 클래스 가중치(class weight)이다.

[0100] 여기서, 합성곱 신경망은 로스 함수에 클래스 가중치 방식을 적용한 수학식 2를 따를 수 있다.

[0102] (수학식 2)

$$Class\ Weight_0 = \frac{Total\ Num\ Samples}{Num\ Samples\ In\ Class\ 0}, Class\ Weight_1 = \frac{Total\ Num\ Samples}{Num\ Samples\ In\ Class\ 1}$$

[0103]

[0104] 단, $Total\ Num\ Samples$ 는 전체 학습용 데이터 수, $Num\ Samples\ In\ Class\ 1$ 과 $Num\ Samples\ In\ Class\ 0$ 는 각각 전체 학습용 데이터 수 중 출력값이 1로 태깅된 데이터의 수(고장 데이터의 수), 그 외의 데이터의 수(정상 데이터의 수)이다.

[0106] 본 발명의 일 실시예의 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법에서, 인공 신경망의 학습 과정에 사용할 데이터의 생성 과정에서는 고장 전 태깅(pre-failure tagging) 기법을 사용할 수 있다.

[0107] 본 발명의 일 실시예의 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 방법은, 가상 머신 및 서버의 고장을 감지하는 경우, 인공 신경망이 로그 임베딩 행렬을 기초로 가상 머신 및 서버에 대한 점수를 산정하여 산정된 점수를 설정 임계값과 비교하도록 수행되고, 고장 발생 전에 조치를 취하는 단계가 더 포함될 수 있다.

[0108] 바람직하게는, 고장 발생 전에 조치를 취하는 단계는 고장의 처리 방법을 결정하는 경우, 고장 발생이 예측되는 가상 머신을 새로운 가상 머신으로 이전(VM migration)하거나 고장 발생이 예측되는 서버에서 동작하는 가상 머신들을 새로운 서버로 이전하도록 수행될 수 있다.

[0110] 도 10은 본 발명의 일 실시예의 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 장치(1000)의 구성도이다.

[0111] 도 10을 참조하면, 본 발명의 일 실시예의 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 장치(1000)는, 프로세서(1100), 메모리(1200), 송수신 장치(transceiver, 1300), 입력 인터페이스 장치(1400), 출력 인터페이스 장치(1500), 저장 장치(1600) 및 버스(bus)(1700)를 포함하여 구성될 수 있다.

[0112] 본 발명의 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 장치(1000)는, 클라우드 컴퓨팅 관리 시스템을 이용하여 구축하는 네트워크 기능 가상화 인프라(NFVI: Network Function Virtualization Infrastructure)를 사용하여 구축되는 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 장치에 있어서, 프로세서(processor)(1100) 및 프로세서(1100)를 통해 실행되는 적어도 하나의 명령이 저장된 메모리(memory)(1200)를 포함하되, 적어도 하나의 명령은 상기 프로세서(1100)가, 가상 머신을 서버 상에서 운용하고 가상 네트워크 기능(VNF: Virtual Network Functions)을 가상 머신에서 운용하여 사용자(client)에게 네트워크 기능을 제공하는 단계(S1000), 가상 머신 및 가상 머신을 동작시키는 서버로부터 발생하는 로그를 수집하는 단계(S2000), 수집한 로그를 기계학습의 입력인 로그 임베딩 행렬(log embedding matrix)로 변환하는 단계(S300) 및 변환된 행렬을 기초로 가상 머신 및 서버의 고장을 예측하는 단계 (S4000) 를 수행하도록 구성된다.

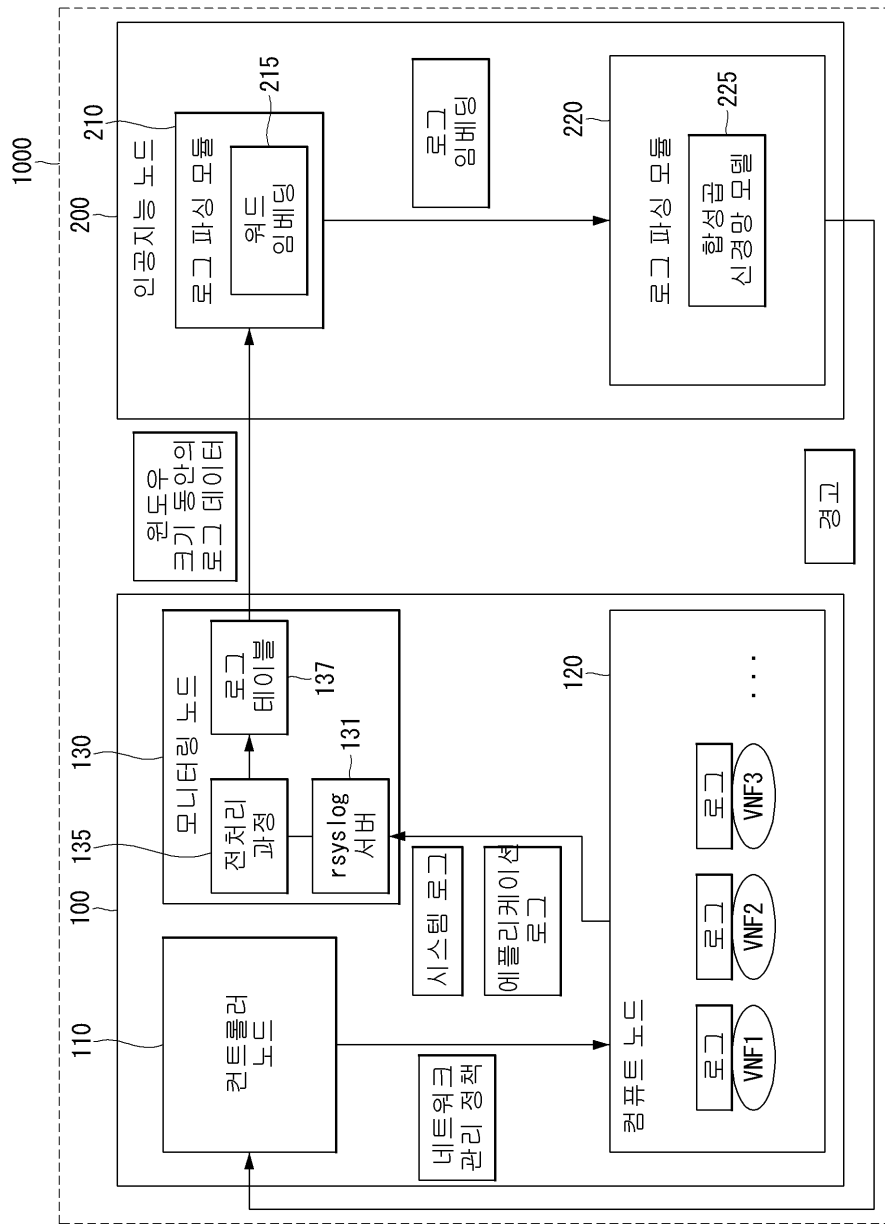
[0113] 프로세서(1100)는 중앙 처리 장치(central processing unit, CPU), 그래픽 처리 장치(graphics processing

unit, GPU), 또는 본 발명의 실시예들에 따른 방법들이 수행되는 전용의 프로세서를 의미할 수 있다.

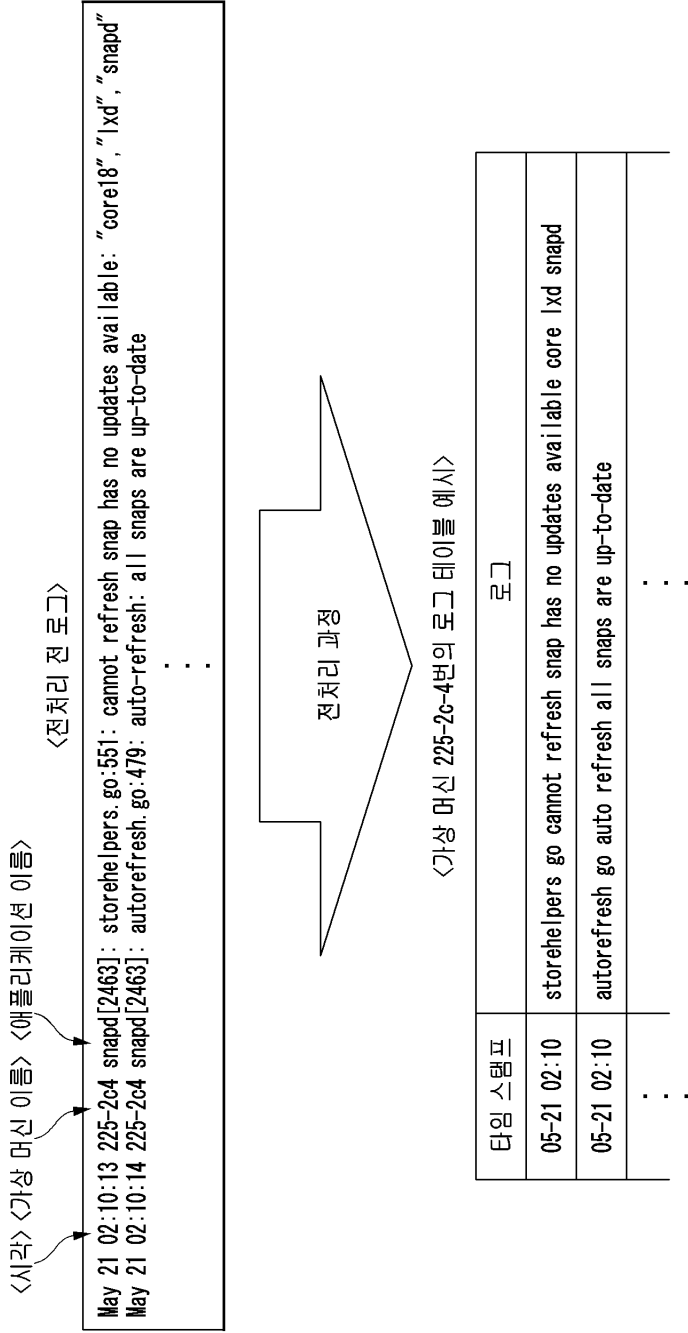
- [0114] 메모리(1200) 및 저장 장치(1600) 각각은 휘발성 저장 매체 및 비휘발성 저장 매체 중에서 적어도 하나로 구성될 수 있다. 예를 들어, 메모리(1200)는 읽기 전용 메모리(read only memory, ROM) 및 랜덤 액세스 메모리(random access memory, RAM) 중에서 적어도 하나로 구성될 수 있다.
- [0115] 또한, 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 장치(1000)는 무선 네트워크를 통해 통신을 수행하는 송수신 장치(transceiver)(1300)를 포함할 수 있다.
- [0116] 또한, 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 장치(1000)는 입력 인터페이스 장치(1400), 출력 인터페이스 장치(1500), 저장 장치(1600) 등을 더 포함할 수 있다.
- [0117] 또한, 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 장치(1000)에 포함된 각각의 구성 요소들은 버스(bus)(1700)에 의해 연결되어 서로 통신을 수행할 수 있다.
- [0118] 본 발명의 로그 메시지 분석을 통한 가상 머신 및 서버의 고장 예측 자동화 장치(1000)의 예를 들면, 통신 가능한 데스크탑 컴퓨터(desktop computer), 랩탑 컴퓨터(laptop computer), 노트북(notebook), 스마트폰(smart phone), 태블릿 PC(tablet PC), 모바일폰(mobile phone), 스마트 워치(smart watch), 스마트 글래스(smart glass), e-book 리더기, PMP(portable multimedia player), 휴대용 게임기, 네비게이션(navigation) 장치, 디지털 카메라(digital camera), DMB(digital multimedia broadcasting) 재생기, 디지털 음성 녹음기(digital audio recorder), 디지털 음성 재생기(digital audio player), 디지털 동영상 녹화기(digital video recorder), 디지털 동영상 재생기(digital video player), PDA(Personal Digital Assistant) 등일 수 있다.
- [0120] 본 발명의 실시예에 따른 방법의 동작은 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 프로그램 또는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의해 읽혀질 수 있는 정보가 저장되는 모든 종류의 기록장치를 포함한다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어 분산 방식으로 컴퓨터로 읽을 수 있는 프로그램 또는 코드가 저장되고 실행될 수 있다.
- [0121] 또한, 컴퓨터가 읽을 수 있는 기록매체는 롬(rom), 램(ram), 플래시 메모리(flash memory) 등과 같이 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치를 포함할 수 있다. 프로그램 명령은 컴파일러(compiler)에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터(interpreter) 등을 사용해서 컴퓨터에 의해 실행될 수 있는 고급 언어 코드를 포함할 수 있다.
- [0122] 본 발명의 일부 측면들은 장치의 문맥에서 설명되었으나, 그것은 상응하는 방법에 따른 설명 또한 나타낼 수 있고, 여기서 블록 또는 장치는 방법 단계 또는 방법 단계의 특징에 상응한다. 유사하게, 방법의 문맥에서 설명된 측면들은 또한 상응하는 블록 또는 아이템 또는 상응하는 장치의 특징으로 나타낼 수 있다. 방법 단계들의 몇몇 또는 전부는 예를 들어, 마이크로프로세서, 프로그램 가능한 컴퓨터 또는 전자 회로와 같은 하드웨어 장치에 의해(또는 이용하여) 수행될 수 있다. 몇몇의 실시예에서, 가장 중요한 방법 단계들의 하나 이상은 이와 같은 장치에 의해 수행될 수 있다.
- [0123] 실시예들에서, 프로그램 가능한 로직 장치(예를 들어, 필드 프로그래머블 게이트 어레이)가 여기서 설명된 방법들의 기능의 일부 또는 전부를 수행하기 위해 사용될 수 있다. 실시예들에서, 필드 프로그래머블 게이트 어레이는 여기서 설명된 방법들 중 하나를 수행하기 위한 마이크로프로세서와 함께 작동할 수 있다. 일반적으로, 방법들은 어떤 하드웨어 장치에 의해 수행되는 것이 바람직하다.
- [0124] 이상 본 발명의 바람직한 실시예를 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구의 범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

도면

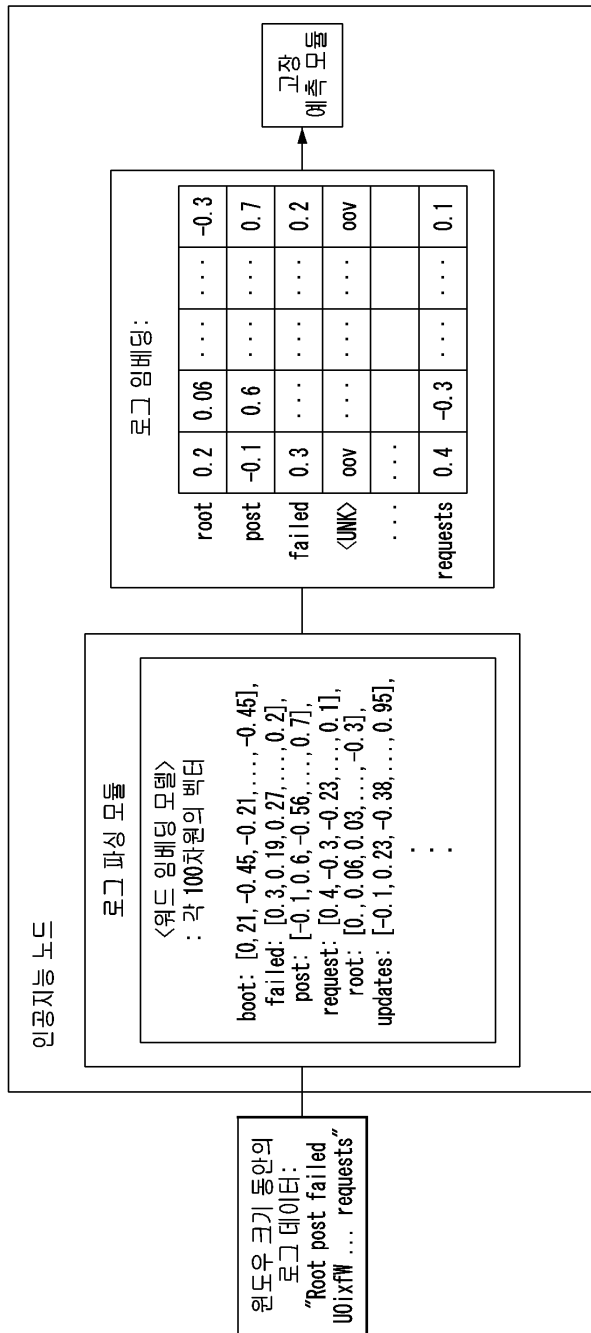
도면1



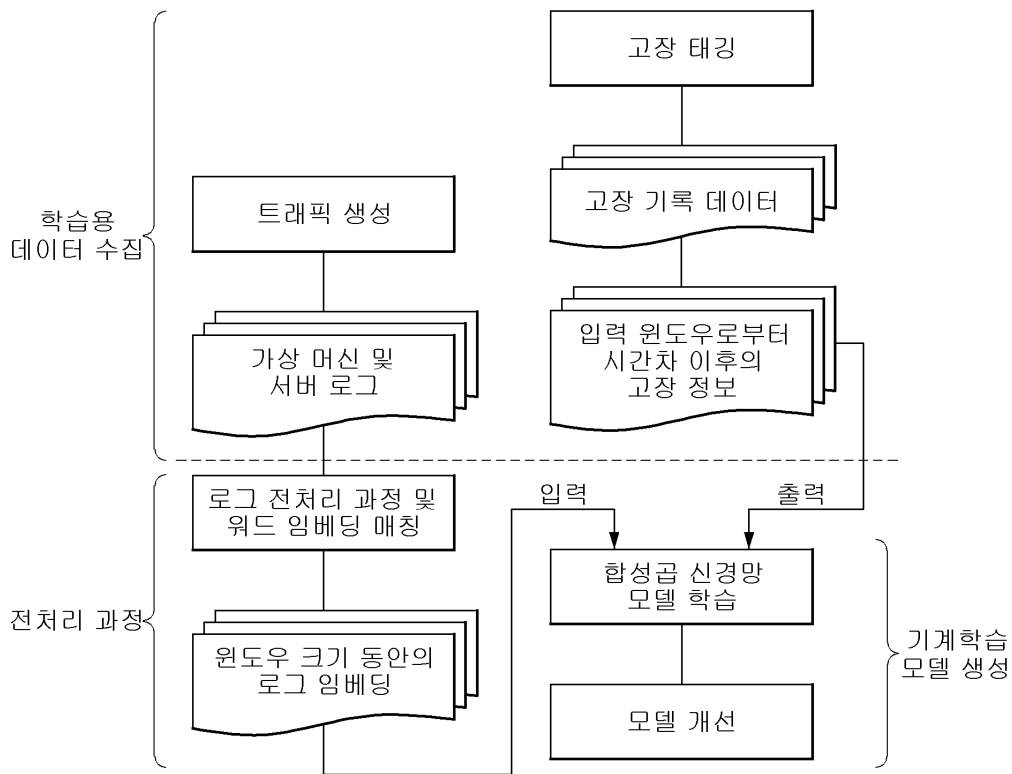
도면2



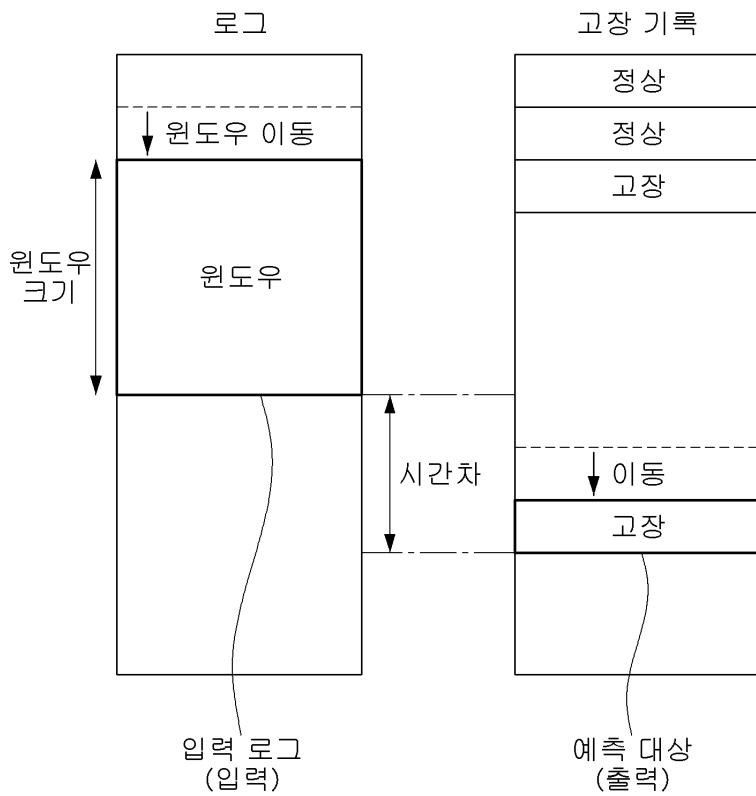
도면3



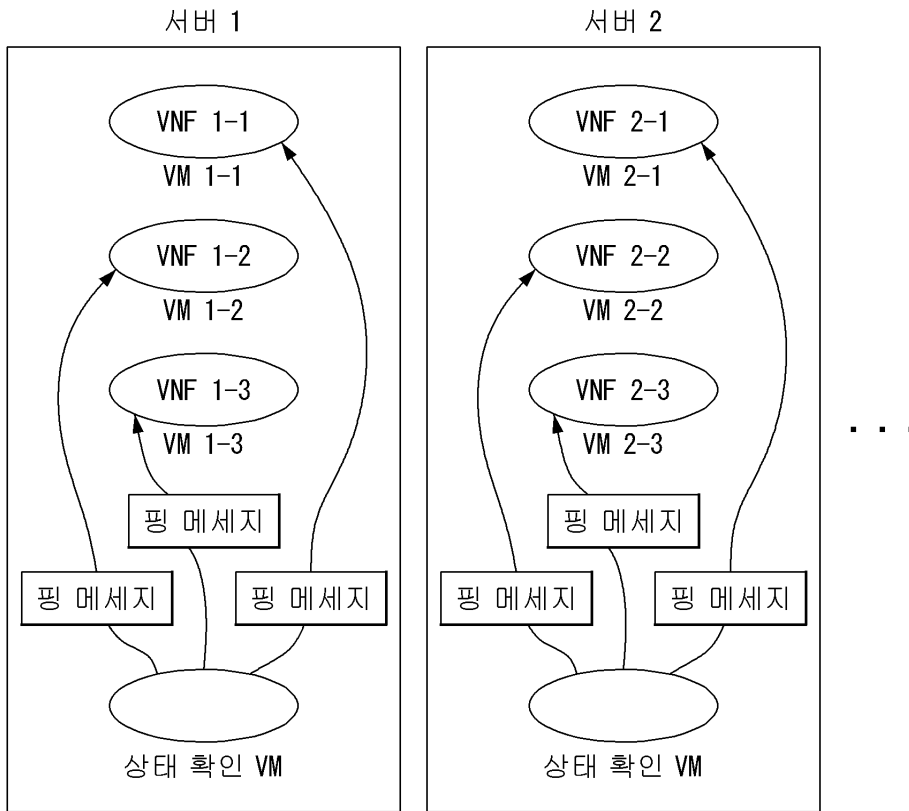
도면4



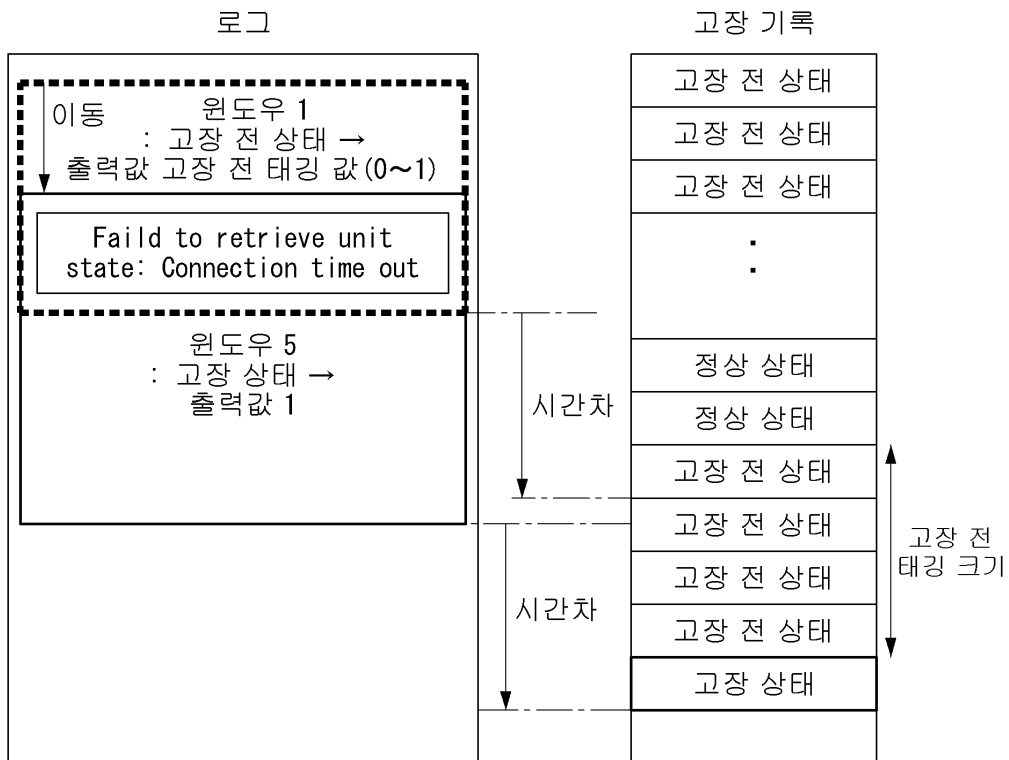
도면5



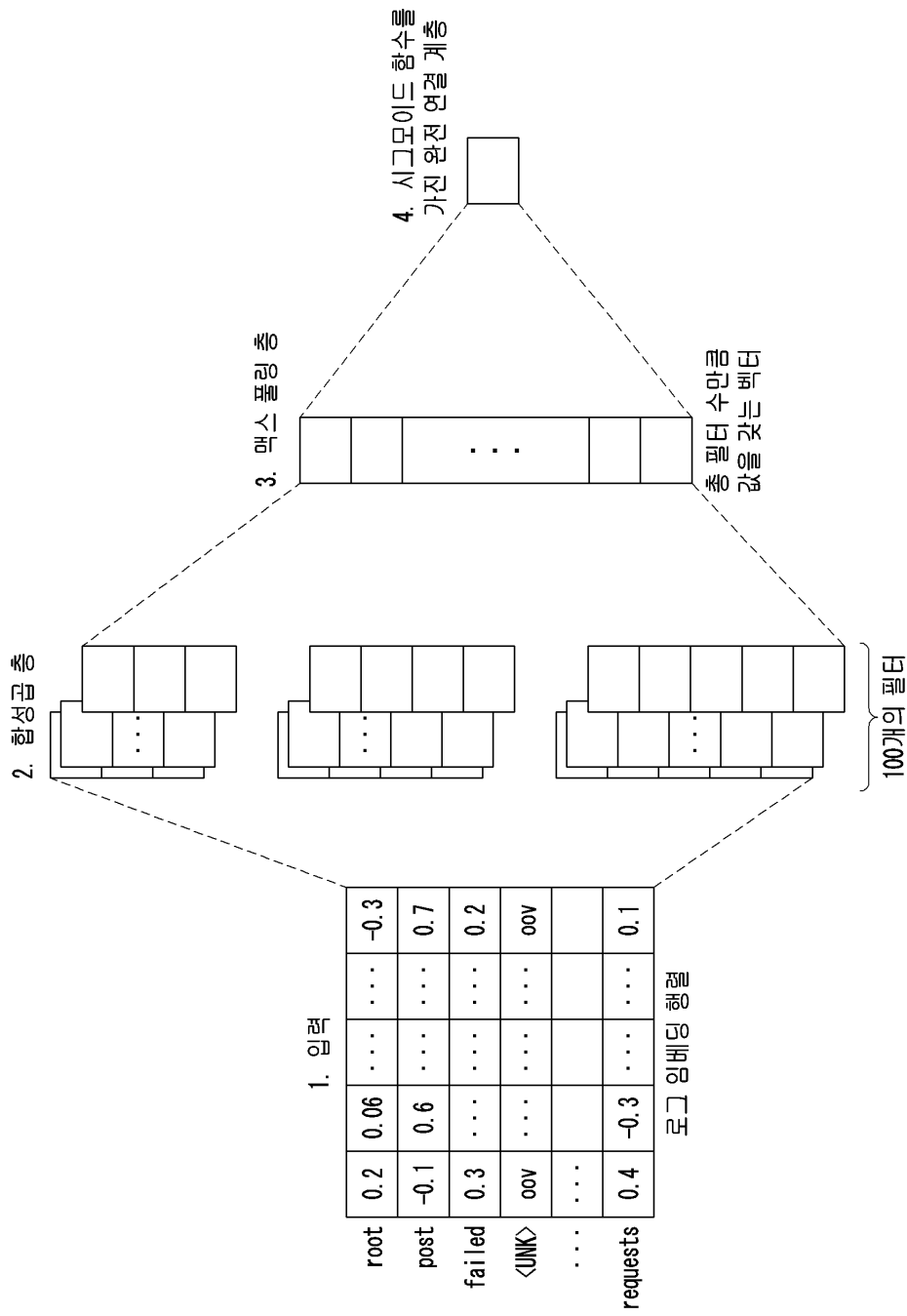
도면6



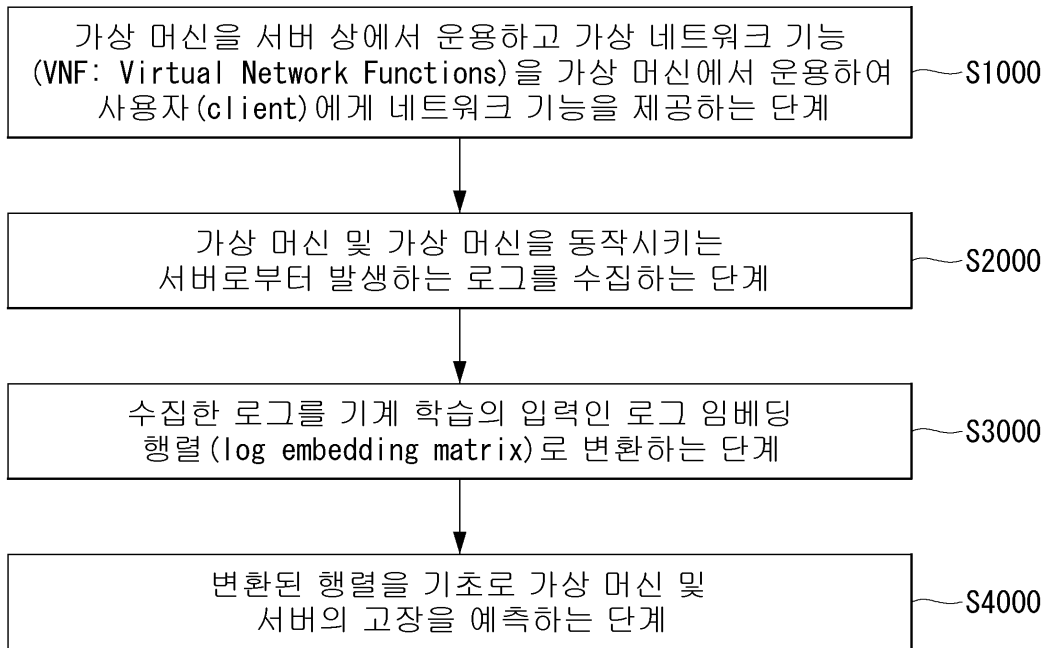
도면7



도면8



도면9



도면10

