(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2014년10월31일
(11) 등록번호 10-1455293
(24) 등록일자 2014년10월21일

(51) 국제특허분류(Int. Cl.)
G06F 21/55 (2013.01)
(21) 출원번호 10-2013-0033316
(22) 출원일자 2013년03월28일
심사청구일자 2013년03월28일
(65) 공개번호 10-2014-0118070
(43) 공개일자 2014년10월08일
(56) 선행기술조사문헌
KR1020100049470 A
KR100745613 B1
KR101009482 B1
전체 청구항 수 : 총 14 항

(73) 특허권자
고려대학교 산학협력단

(72) 발명자
이희조
박현도

(74) 대리인
특허법인엠에이피에스

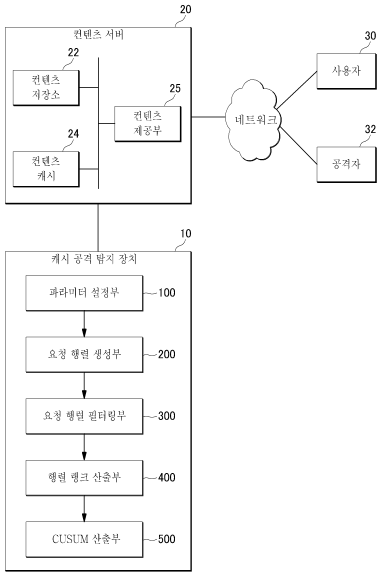
심사관 : 구본재

(54) 발명의 명칭 **캐시 공격 탐지 장치 및 방법**

(57) 요 약

본 발명은 각 모니터링 시점에서, 컨텐츠 제공 요청을 받은 컨텐츠에 대한 요청 표지를 포함하는 요청 행렬을 생성하는 요청 행렬 생성부; 및 상기 요청 행렬의 랭크(rank)를 산출하는 행렬 랭크 산출부;를 포함하되, 상기 랭크를 통해 상기 요청 행렬의 무작위성(randomness)을 조사하여, 컨텐츠 제공 요청 분포가 균일 분포(uniform distribution)에 근접해졌는지 여부에 따라 컨텐츠에 대한 요청을 조작하는 방식의 서비스 거부 공격을 탐지하는 캐시 공격 탐지 장치를 제공한다.

대 표 도 - 도1



이 발명을 지원한 국가연구개발사업
과제고유번호 WR080951
부처명 서울특별시
연구관리전문기관 서울통상산업진흥원
연구사업명 세계유수연구소유치지원사업
연구과제명 4차년도 [1-B] Blended Services Applications
기 여 율 1/1
주관기관 고려대학교 산학협력단
연구기간 2011.12.01 ~ 2012.11.30

특허청구의 범위

청구항 1

캐시에 대한 공격을 탐지하는 장치에 있어서,

각 모니터링 시점에서, 컨텐츠 제공 요청을 받은 컨텐츠에 대한 요청 표지를 포함하는 요청 행렬을 생성하는 요청 행렬 생성부; 및

상기 요청 행렬의 랭크(rank)를 산출하는 행렬 랭크 산출부;를 포함하되,

상기 랭크를 통해 상기 요청 행렬의 무작위성(randomness)을 조사하여, 컨텐츠 제공 요청 분포가 균일 분포(uniform distribution)에 근접해졌는지 여부에 따라 컨텐츠에 대한 요청을 조작하는 방식의 서비스 거부 공격을 탐지하며, 상기 랭크가 랭크 임계값을 넘어서는 경우, 서비스 거부 공격이 있다고 판단하는 캐시 공격 탐지 장치.

청구항 2

제 1 항에 있어서,

상기 요청 행렬 생성부는

컨텐츠 제공을 위해 컨텐츠 객체가 저장되는 캐시(cache)를 조사하여 상기 요청 행렬을 생성하는 캐시 공격 탐지 장치.

청구항 3

삭제

청구항 4

제 1 항에 있어서,

상기 캐시 공격 탐지 장치는

상기 요청 행렬 중 연속으로 컨텐츠 제공 요청을 받은 컨텐츠에 대해 표시된 요청 표지를 소거하는 요청 행렬 필터링부;를 더 포함하는 캐시 공격 탐지 장치.

청구항 5

제 1 항에 있어서,

상기 캐시 공격 탐지 장치는

상기 행렬 랭크 산출부가 산출한 랭크에 대한 CUSUM(cumulative sum)을 추적하는 CUSUM 산출부;를 더 포함하며,

상기 CUSUM의 편차가 횟수 임계값 이상 연속으로 편차 임계값을 넘어서는 경우, 서비스 거부 공격이 있다고 판단하는 캐시 공격 탐지 장치.

청구항 6

제 1 항에 있어서,

상기 행렬 랭크 산출부는

가우시안 소거법을 사용하여 상기 랭크를 산출하는 캐시 공격 탐지 장치.

청구항 7

제 1 항에 있어서,

상기 캐시 공격 탐지 장치는

컨텐츠 개수에 기초하여 상기 요청 행렬의 크기를 결정하고,

상기 요청 행렬의 크기에 기초하여 모니터링 시간 단위를 결정하는 캐시 공격 탐지 장치.

청구항 8

제 1 항에 있어서,

상기 요청 행렬은

컨텐츠 제공 요청을 받지 않은 컨텐츠에 대응되는 요소는 0으로 설정되고, 컨텐츠 제공 요청을 받은 컨텐츠에 대응되는 요소는 1로 설정되는 캐시 공격 탐지 장치.

청구항 9

제 1 항에 있어서,

상기 요청 행렬 생성부는

상기 컨텐츠 제공 요청을 받은 컨텐츠의 이름을 해시(hash)하여 상기 요청 행렬의 요소에 대응시키는 캐시 공격 탐지 장치.

청구항 10

캐시 공격 탐지 장치를 사용하는 캐시 공격 탐지 방법에 있어서,

각 모니터링 시점에서, 컨텐츠 제공 요청을 받은 컨텐츠에 대한 요청 표지를 포함하는 요청 행렬을 생성하는 단계; 및

상기 요청 행렬의 랭크(rank)를 산출하는 단계;를 포함하며,

상기 랭크를 통해 상기 요청 행렬의 무작위성(randomness)을 조사하여, 컨텐츠 제공 요청 분포가 균일 분포(uniform distribution)에 근접해졌는지 여부에 따라 컨텐츠에 대한 요청을 조작하는 방식의 서비스 거부 공격을 탐지하되, 상기 랭크가 랭크 임계값을 넘어서는 경우, 서비스 거부 공격이 있다고 판단하는 캐시 공격 탐지 방법.

청구항 11

제 10 항에 있어서,

상기 요청 행렬의 랭크에 대한 CUSUM(cumulative sum)을 추적하는 단계;를 더 포함하며,

상기 CUSUM의 편차가 횟수 임계값 이상 연속으로 편차 임계값을 넘어서는 경우, 서비스 거부 공격이 있다고 판단하는 캐시 공격 탐지 방법.

청구항 12

제 10 항에 있어서,

컨텐츠 개수에 기초하여 상기 요청 행렬의 크기를 결정하고, 상기 요청 행렬의 크기에 기초하여 모니터링 시간 단위를 결정하는 단계;를 더 포함하는 캐시 공격 탐지 방법.

청구항 13

제 10 항에 있어서,

상기 요청 행렬 중 연속으로 컨텐츠 제공 요청을 받은 컨텐츠에 대해 표시된 요청 표지를 소거하는 단계;를 더 포함하는 캐시 공격 탐지 방법.

청구항 14

제 10 항에 있어서,

상기 요청 행렬을 생성하는 단계는

컨텐츠 제공을 위해 컨텐츠 객체가 저장되는 캐시(cache)를 조사하여, 상기 컨텐츠 제공 요청을 받은 컨텐츠의 이름을 해시(hash)함으로써 상기 요청 행렬의 대응되는 요소의 인덱스를 산출하는 캐시 공격 탐지 방법.

청구항 15

제 10 항에 있어서,

상기 요청 행렬은

컨텐츠 제공 요청을 받지 않은 컨텐츠에 대응되는 요소는 0으로 설정되고, 컨텐츠 제공 요청을 받은 컨텐츠에 대응되는 요소는 1로 설정되는 캐시 공격 탐지 방법.

명 세 서

기 술 분 야

[0001] 본 발명은 서비스 거부 공격의 일종인 캐시 오염 공격(cache pollution attack)을 탐지하는 장치 및 방법에 관한 것이다.

배 경 기 술

[0002] 최근 사용자에 의해 만들어진 컨텐츠들이 폭발적으로 늘어남에 따라, 컨텐츠가 사용자에게 효율적으로 전달될 수 있도록 캐시 서버 사용이 늘어나고 있으며, 각 라우터에 캐시를 구현하는 형태의 네트워크 아키텍처인 컨텐츠 중심 네트워킹(CCN: Content-Centric Networking)도 주목받고 있다.

[0003] 이에 따라, 컨텐츠 서버 또는 캐시 서버를 공격하기 위한 신종 서비스 거부 공격(DDoS: Distributed Denial of Service)들도 등장하고 있다. 서비스 거부 공격은 최근 등장하고 있는 다양하고 위협적인 보안 위협 중에서도 치명적인 피해를 끼칠 수 있는 위험한 공격이다. 따라서 신속하게 탐지하고 대응하는 것이 중요하다.

[0004] 캐시 오염 공격(cache pollution attack)은 캐시 내 컨텐츠의 지역성(locality)이 위반되도록 비인기 컨텐츠를 계속 요청하여, 인기 컨텐츠가 캐시 내에 저장되어 있지 못하게 만들어, 사용자에게 컨텐츠 제공 서비스를 제대로 제공할 수 없게 만드는 방식의 서비스 거부 공격이다.

[0005] 사용자의 컨텐츠 제공 요청은 일반적으로 Zipf 분포를 따른다. 즉, 인기 컨텐츠에 대한 제공 요청이 전체 컨텐츠 제공 요청에서 차지하는 비중이 굉장히 높다. 따라서 공격이 없을 때는 인기 컨텐츠가 캐시에 이미 저장되어 있을 확률이 높고 사용자가 해당 인기 컨텐츠를 요청할 확률 또한 높으므로, 캐시 히트율(hit ratio)이 높을 것이다. 그러나 캐시 오염 공격 때문에 인기 컨텐츠가 캐시에서 쫓겨나면, 사용자가 해당 인기 컨텐츠를 요청할 확률은 여전히 높지만 캐시에는 해당 컨텐츠가 저장되어 있지 못하므로, 도 2에 도시되어 있는 예와 같이, 캐시 히트율이 낮아진다.

[0006] 도 2는 적법한 컨텐츠 제공 요청에 대한 공격, 즉 오염된 컨텐츠 제공 요청의 비율이 높아질 때 캐시 히트율이 낮아짐을 보여주고 있다. 인기 컨텐츠를 캐시로 다시 저장하더라도 다른 컨텐츠에 의해 다시 캐시에서 쫓겨나기 때문에 여전히 캐시에는 해당 컨텐츠가 존재하지 않게 된다. 따라서 캐시 오염 공격이 성공하면, 인기 컨텐츠를 캐시에 불러들이는 동작을 반복하더라도, 사용자에게는 해당 인기 컨텐츠가 원할하게 제공되지 못하게 된다.

[0007] 그러므로 이러한 유형의 서비스 거부 공격을 효과적으로 탐지할 수 있는 장치 및 방법이 필요하다.

[0008] 이와 관련하여 미국등록특허 US7930428호("Verification of DNS accuracy in cache poisoning")에는 DNS 캐시 포이즌 공격(DNS cache poisoning attack)을 막는 방법이 개시되어 있다.

[0009] 또한, 미국등록특허 US7656840호("Method of reducing denial-of-service attacks and a system as well as an access router therefor")에는 모바일 IP 환경에서 서비스 거부 공격을 감소시키는 방법이 개시되어 있다.

발명의 내용

해결하려는 과제

[0010] 본 발명은 전술한 문제를 해결하기 위한 것으로서, 그 목적은 저비용으로 정확하게 서비스 거부 공격을 탐지하는 캐시 공격 탐지 장치 및 방법을 제공하는 것이다.

과제의 해결 수단

[0011] 상기와 같은 목적을 달성하기 위한 본 발명의 제 1 측면에 따른 캐시에 대한 공격을 탐지하는 장치는 각 모니터링 시점에서, 컨텐츠 제공 요청을 받은 컨텐츠에 대한 요청 표지를 포함하는 요청 행렬을 생성하는 요청 행렬 생성부; 및 상기 요청 행렬의 랭크(rank)를 산출하는 행렬 랭크 산출부;를 포함하되, 상기 랭크를 통해 상기 요청 행렬의 무작위성(randomness)을 조사하여, 컨텐츠 제공 요청 분포가 균일 분포(uniform distribution)에 근접해졌는지 여부에 따라 컨텐츠에 대한 요청을 조작하는 방식의 서비스 거부 공격을 탐지하는 것을 특징으로 한다.

[0012] 상기와 같은 목적을 달성하기 위한 본 발명의 제 2 측면에 따른 캐시 공격 탐지 장치를 사용하는 캐시 공격 탐지 방법에 있어서, 각 모니터링 시점에서, 컨텐츠 제공 요청을 받은 컨텐츠에 대한 요청 표지를 포함하는 요청 행렬을 생성하는 단계; 및 상기 요청 행렬의 랭크(rank)를 산출하는 단계;를 포함하며, 상기 랭크를 통해 상기 요청 행렬의 무작위성(randomness)을 조사하여, 컨텐츠 제공 요청 분포가 균일 분포(uniform distribution)에 근접해졌는지 여부에 따라 컨텐츠에 대한 요청을 조작하는 방식의 서비스 거부 공격을 탐지하는 것을 특징으로 한다.

발명의 효과

[0013] 본 발명은 캐시 공격 탐지 장치 및 방법에 있어, 저비용으로 빠르고 정확하게 서비스 거부 공격을 탐지하는 효과를 얻는다.

[0014] 트래픽이 아니라 캐시 내 존재하는 컨텐츠를 조사하며, 해당 컨텐츠들에 대한 제공 요청의 분포를 행렬로 표현하고, 가우시안 소거법이라는 단순한 연산을 사용하여 행렬 랭크의 변화를 감시하는 간단한 방법을 사용하므로 노력과 자원이 적게 들고 탐지 성능이 좋다.

[0015] 이에 더해 XOR이라는 단순한 연산을 통해 행렬에서 인기 컨텐츠를 미리 소거하므로 효율이 더욱 높다.

[0016] 또한 CUSUM을 추적하여 랭크 추적만으로 놓칠 수 있는 공격까지 탐지하여 탐지 성능이 더욱 높다.

도면의 간단한 설명

[0017] 도 1은 본 발명의 일실시예에 따른 캐시 공격 탐지 장치의 구조를 도시함.

도 2는 캐시 오염 공격이 있을 때 캐시 히트율 추이를 도시함.

도 3은 본 발명의 일실시예에 따른 캐시 공격 탐지 방법의 흐름을 도시함.

도 4는 본 발명의 일실시예에 따른 캐시 공격 탐지 방법의 파라미터 설정 단계의 흐름을 도시함.

도 5는 본 발명의 일실시예에 따른 캐시 공격 탐지 방법의 요청 행렬 생성 단계의 흐름을 도시함.

도 6은 본 발명의 일실시예에 따른 캐시 공격 탐지 방법의 요청 행렬 필터링 단계의 흐름을 도시함.

도 7은 본 발명의 일실시예에 따른 캐시 공격 탐지 방법의 행렬 랭크(rank) 산출 단계의 흐름을 도시함.

도 8은 본 발명의 일실시예에 따른 캐시 공격 탐지 방법의 CUSUM(cumulative sum) 산출 단계의 흐름을 도시함.

도 9는 캐시 내를 조사하였을 때의 랭크 추이와 트래픽을 조사하였을 때의 랭크 추이를 비교 도시함.

발명을 실시하기 위한 구체적인 내용

[0018] 아래에서는 첨부한 도면을 참조하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 본 발명의 실시예를 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.

[0019] 명세서 전체에서, 어떤 부분이 다른 부분과 "연결"되어 있다고 할 때, 이는 "직접적으로 연결"되어 있는 경우뿐 아니라, 그 중간에 다른 소자를 사이에 두고 "전기적으로 연결"되어 있는 경우도 포함한다. 또한 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.

[0020] 도 1은 본 발명의 일실시예에 따른 캐시 공격 탐지 장치의 구조를 도시하고 있다.

[0021] 본 발명의 일실시예에 따른 캐시 공격 탐지 장치(10)는 컨텐츠 서버(20)에 대한 서비스 거부 공격의 일종인 캐시 오염 공격(cache pollution attack)을 탐지하기 위해 컨텐츠 서버(20)에 포함되거나 연결되도록 구성될 수 있다.

[0022] 컨텐츠 서버(20)는 컨텐츠가 저장되는 컨텐츠 저장소(22) 및 컨텐츠 저장소(22)에 저장되어 있는 컨텐츠가 사용자(30)에게 제공되기 위해 임시 저장되는 컨텐츠 캐시(24), 및 사용자(30)로부터 컨텐츠 제공 요청을 수신하고, 컨텐츠 저장소(22) 및 컨텐츠 캐시(24)를 사용하여 사용자(30)에게 컨텐츠를 제공하는 컨텐츠 제공부(25)를 포함한다. 본 명세서에서 사용자(30) 및 공격자(32)는 하드웨어나 소프트웨어 등의 개체(entity)를 지칭하는 개념으로 사용된다. 컨텐츠 서버(20)는 사용자(30)와 다양한 종류의 네트워크를 통해 연결될 수 있으며, 본명세서에서 지칭하는 네트워크의 종류에는 제한이 없다.

[0023] 컨텐츠 서버(20)는 하나 이상의 별도의 장치가 네트워크를 통해 서로 연결되도록 구성된 것일 수 있다. 예를 들어, 컨텐츠 서버(20)는 컨텐츠 저장소(22)를 포함하는 하나 이상의 서버와 컨텐츠 캐시(24)를 포함하는 하나 이상의 서버(예: 캐시 서버)가 서로 연결되어 서버군이 형성된 것일 수 있다. 일실시예에서, 컨텐츠 서버(20)는 컨텐츠 중심 네트워크(CCN: content contric network)를 구성하는 요소로 사용될 수 있다. 이러한 실시예에서는 각 라우터에 컨텐츠 캐시(24)가 포함된다. 또는 일실시예에서 컨텐츠 서버(20)는 캐시 서버만을 포함할 수 있으며, 캐시 공격 탐지 장치(10)는 캐시 서버에만 포함되거나 연결된 것일 수 있다.

[0024] 본 발명의 일실시예에 따른 캐시 공격 탐지 장치(10)는 컨텐츠 서버(20)에 대한 서비스 거부 공격을 탐지하기 위하여, 파라미터 설정부(100), 요청 행렬 생성부(200), 요청 행렬 필터링부(300), 행렬 랭크 산출부(400), 및 CUSUM 산출부(500)를 포함한다.

[0025] 앞서 도 2를 통해 살펴본 바와 같이, 캐시 오염 공격은 캐시에 저장되는 컨텐츠의 지역성(locality)을 조작하여 캐시의 히트율을 낮추어 인기 컨텐츠가 원할하게 서비스되지 못하도록 하는 일종의 서비스 거부 공격이다.

[0026] 캐시 오염 공격은 인기 컨텐츠를 제외한 비인기 컨텐츠에 대해 제공 요청을 계속 보내는 방법으로 수행될 수 있다. 가짜 지역성 공격(false-locality attack)이라고 불리는 이 방법에서, 공격자(32)는 적법한 사용자(30)가 요청하지 않는 비인기 컨텐츠들을 파악하고, 컨텐츠 서버(20)에 해당 비인기 컨텐츠들에 대한 요청을 반복적으로 보낸다. 이 방법을 사용하려는 공격자(32)는 인기 컨텐츠의 전체 분포를 알고 있어야 하는데, 인기 컨텐츠는 시간에 따라 캐시의 위치에 따라 달라지므로 현실적인 방법은 아니다.

[0027] 대신 공격자(32)는 모든 컨텐츠에 대해 균일 분포(uniform distribution)로 컨텐츠 제공 요청을 보내는 방법으로 캐시 오염 공격을 수행할 수 있다. 지역성 혼란 공격(locality-disruption attack)이라 불리는 이 방법에서, 공격자(32)는 요청할 컨텐츠를 무작위로(randomly) 선정하면 되므로, 컨텐츠의 인기도를 파악하지 않고도 컨텐츠 서버(20)를 효과적으로 공격할 수 있다.

[0028] 본 발명의 일실시예에 따른 캐시 공격 탐지 장치(10)는 후자의 방법에 대해 대응한다. 이를 위해 캐시 공격 탐지 장치(10)는 컨텐츠 제공 요청이 Zipf 분포를 따르지 않고 균일 분포에 근접해졌는지를 감시한다. 즉, 컨텐츠 제공 요청의 무작위성(randomness)를 감시한다.

[0029] 사용자(30)가 무작위로 컨텐츠를 요청하는 경우는 실제 상황에서 발생하지 않는다. 적법한 사용자(30)만이 존재한다면, 어떠한 상황에서도 인기있는 컨텐츠와 인기 없는 컨텐츠가 존재하게 되고, 인기 있는 컨텐츠에 대해서는 반복된 요청이 컨텐츠 서버(20)에 들어오게 된다. 이러한 반복된 요청은 많은 서로 다른 사용자(30)에게서 들어오는 요청이다. 따라서 요청된 컨텐츠의 분포가 Zipf 분포를 따르지 않고 무작위한 균일 분포를 따른다면, 실생활에서 나올 수 없는 분포이므로 공격으로 판단할 수 있다.

[0030] 이를 위해 본 발명의 일실시예에 따른 캐시 공격 탐지 장치(10)는 무작위성이 높은 이진 행렬은 높은 랭크 값을 갖는다는 특징을 활용한다. 따라서 행렬의 랭크 값이 소정의 임계값을 넘어서면 컨텐츠 제공 요청이 균일 분포에 근접하였다고 판단할 수 있다. 본명세서에서 이 임계값은 랭크 임계값으로 지칭되며, 바람직한 실시예에서 랭크 임계값은 4이다. 이는 수학적으로 증명된 값이다. 이 값의 이상의 랭크값이 나오면, 99.999%이상의 확률로 행렬의 원소들의 분포가 랜덤해짐이 증명되었다.

[0031] 컨텐츠 제공 요청의 분포를 표현할 수 있는 요청 행렬을 생성하고, 요청 행렬의 랭크(rank)의 값과 변화를 추적하기 위해, 본 발명의 일실시예에 따른 캐시 공격 탐지 장치(10)는 파라미터 설정부(100), 요청 행렬 생성부(200), 요청 행렬 필터링부(300), 행렬 랭크 산출부(400), 및 CUSUM 산출부(500)를 포함한다.

[0032] 요청 행렬 생성부(200)는 각 모니터링 시점에서 컨텐츠 제공 요청을 받은 컨텐츠에 대한 요청 표지를 포함하는 요청 행렬을 생성한다. 요청 행렬은 0과 1로 이루어진 이진(binary number) 행렬이 사용될 수 있다. 컨텐츠 제공 요청을 받지 않은 컨텐츠에 대응되는 요소는 0으로 설정되고, 컨텐츠 제공 요청을 받은 컨텐츠에 대응되는 요소는 1로 설정된다. 즉, 요청 행렬은 컨텐츠 제공 요청을 받은 컨텐츠에 대한 요청 표지로 1을 사용한다.

[0033] 이를 위해 요청 행렬 생성부(200)는 영행렬을 생성한 후 컨텐츠 제공 요청을 받은 컨텐츠에 대응되는 요소를 1로 설정할 수 있다. 컨텐츠 제공 요청을 받은 컨텐츠를 요청 행렬의 요소에 대응시키는 데에는 해당 컨텐츠의 이름의 해시(hash)가 사용될 수 있다. 자세한 내용은 후술한다.

[0034] 행렬 랭크 산출부(400)는 행렬 랭크를 산출한다. 행렬 랭크는 가우시안 소거법을 사용하여 산출할 수 있다. 가우시안 소거법을 행렬에 적용하면, 행렬은 상삼각행렬(대각행렬 기준으로 아래는 모두 0인 행렬)이 된다. 그 중에 모두 0으로 구성되지 않은 행의 개수가 랭크값이다.

[0035] 행렬 랭크 산출부(400)가 행렬 랭크를 산출하기 전에, 요청 행렬 필터링부(300)가 요청 행렬에서 인기 컨텐츠를 미리 걸러내어 효율을 높일 수 있다. 요청 행렬 필터링부(300)는 요청 행렬 중 연속으로 컨텐츠 제공 요청을 받은 컨텐츠에 대해 표시된 요청 표지를 소거한다. 요청 행렬의 1인 요소 중 이전 모니터링 시점에 생성된 요청 행렬에서도 1인 요소를 0으로 소거하기 위해 XOR 연산을 사용한다. 자세한 내용은 후술한다.

[0036] CUSUM 산출부(500)가 요청 행렬의 랭크에 대한 CUSUM(cumulative sum)을 추적할 수 있다. CUSUM의 편차가 횟수 임계값 이상 연속으로 편차 임계값을 넘어서는 경우, 서비스 거부 공격이 있다고 판단할 수 있다.

[0037] CUSUM을 통해 랭크의 변화까지 추적하는 이유는 아주 느린 공격 속도를 가진 공격(low-rate attack)까지 탐지하기 위함이다. 이러한 낮은 속도의 공격은 랭크만으로는 탐지할 수 없을 수 있으나, CUSUM을 함께 조사하면, 낮은 공격 속도를 가진(정상에 비해 1/10의 속도를 가진) 공격도 실시간으로 탐지가 가능하다. 자세한 내용은 후술한다.

[0038] 본격적으로 모니터링을 시작하기 전에 파라미터 설정부(100) 요청 행렬의 크기 및 모니터링 단위 시간 등을 포함하는 파라미터를 설정할 수 있다. 자세한 내용은 후술하되, 도 9를 통해, 바람직한 실시예에서 요청 행렬은 컨텐츠 객체가 저장되는 캐시(cache)를 조사하여 생성됨을 먼저 살펴본다.

[0039] 도 9는 캐시 내를 조사하였을 때의 랭크 추이(첫번째 도면)와 트래픽을 조사하였을 때의 랭크 추이(두번째 도면)를 비교 도시하고 있다.

[0040] 컨텐츠에 대한 요청은 컨텐츠 캐시(24) 내부를 조사하여 파악될 수도 있고, 컨텐츠 서버(20)가 수신하는 요청을 컨텐츠 캐시(24) 외부에서 조사하여 파악될 수도 있을 것이다. 이에 대해 실험한 결과, 도면에 도시되어 있는 바와 같이, 캐시 내를 조사하였을 경우가 Zipf 분포일 때와 균일 분포일 때의 랭크 값의 추이를 더 잘 추적하는 것으로 나타났다. 따라서 본 발명의 일실시예에 따른 요청 행렬 생성부(200)는 요청 행렬 생성시 컨텐츠 캐시(24) 내부를 조사한다.

[0041] 도 3은 본 발명의 일실시예에 따른 캐시 공격 탐지 방법의 흐름을 도시하고 있다. 각 단계를 대략적으로 살펴보고 자세한 내용은 후술한다.

[0042] 파라미터를 설정한다(S100). 컨텐츠 개수에 기초하여 요청 행렬의 크기를 결정하고, 요청 행렬의 크기에 기초하여 모니터링 시간 단위를 결정한다.

[0043] 각 모니터링 시점에서(S200), 컨텐츠 제공 요청을 받은 컨텐츠에 대한 요청 표지를 포함하는 요청 행렬을 생성한다(S300).

[0044] 다음, 요청 행렬 중 연속으로 컨텐츠 제공 요청을 받은 컨텐츠에 대해 표시된 요청 표지를 소거하여 요청 행렬을 필터링한다(S400).

[0045] 다음, 가우스 소거법을 사용하여 요청 행렬의 랭크(rank)를 산출한다(S500).

[0046] 다음, 행렬 랭크의 CUSUM을 산출한다(S700).

[0047] 행렬 랭크가 랭크 임계값을 넘어서거나(S600), CUSUM 편차가 편차 임계값을 넘어서면(S800), 공격이 탐지된 것이다(S900). 도면은 랭크 임계값과 편차 임계값을 별도의 단계에서 추적하는 바람직한 실시예를 도시하고 있으나, 한꺼번에 추적하여도 된다.

[0048] 공격이 탐지되면 공격이 탐지되었음을 알리고 대응 조치를 취한다. 공격이 탐지되지 않았을 때는 상기 단계들

(S200~S800)을 반복하여 모니터링을 계속한다.

[0049] 도 4는 본 발명의 일실시예에 따른 캐시 공격 탐지 방법의 파라미터 설정 단계의 흐름을 도시하고 있다.

[0050] 컨텐츠 수를 산출한다(S110). 예를 들어, 컨텐츠 객체 수 총합(So) 및 캐시 내 객체 수 평균(Sc)가 산출될 수 있다. 캐시 내 객체 수 평균을 산출하는 이유는 도 9를 통해 전술한 바와 같이, 본 발명의 일실시예에 따른 캐시 공격 탐지 방법은 요청 행렬 생성시 컨텐츠 캐시(24) 외부가 아닌 컨텐츠 캐시(24) 내부를 조사하기 때문이다.

[0051] 이에 기초하여 요청 행렬의 크기를 결정한다(S120). 요청 행렬로 (nxn) 행렬이 사용된다면 n은 도시되어 있는 바와 같이, 캐시 내 객체 수 평균에 기초하여 산출될 수 있다. 예를 들어, 평균 파일 크기가 1MB이고 컨텐츠 캐시(24)의 크기가 1TB라면, n은 대략 1,000이 될 수 있다.

[0052] 요청 행렬의 크기가 너무 작으면 요청 분포의 무작위성을 제대로 탐지하지 못할 것이고, 너무 크면 공간 자원 낭비가 심할 것이므로, 적절한 크기를 산출하는 것은 중요하다.

[0053] 다음 모니터링 단위 시간 크기를 결정한다(S130). 모니터링 단위 시간은 실제 시간 단위를 사용할 수도 있으나, 컨텐츠 제공 요청 건수에 따라 모니터 시간을 정하는 것이 효율적일 수 있다. 예를 들어, 바람직한 실시예에서 3n개의 요청마다 모니터링되도록 설정될 수 있다.

[0054] 도 5는 본 발명의 일실시예에 따른 캐시 공격 탐지 방법의 요청 행렬 생성 단계의 흐름을 도시하고 있다.

[0055] t 시점의 요청 행렬 Mt를 생성한다(S310). Mt는 (nxn) 크기의 영행렬이다. 컨텐츠 제공 서비스를 요청받은 컨텐츠에 대해 1을 요청 표지로서 설정할 것이므로, 요청 행렬 Mt는 이진 행렬이 된다.

[0056] 요청 표지를 설정할 행렬의 요소를 찾기 위해, 컨텐츠 제공 서비스를 요청받은 컨텐츠 객체 이름(c)에 대응되는 인덱스 (i, j)를 산출한다(S320). 전술한 바와 같이 해시(hash) 연산이 사용되었다. 해시 연산은 입력으로 컨텐츠 이름(c)을 받아서, 행렬의 인덱스 (i, j)를 출력한다. 해시와 나머지연산(mod)이라는 간단한 연산을 사용하므로 도 5의 단계는 효율이 높다.

[0057] (i, j) 번째 요소를 1로 설정한다(S330). 결국 요청 행렬은 요청된 컨텐츠들에 대한 부분만 1이 되고, 요청받지 않은 부분은 모두 0으로 남아있게 된다.

[0058] 도 6은 본 발명의 일실시예에 따른 캐시 공격 탐지 방법의 요청 행렬 필터링 단계의 흐름을 도시하고 있다.

[0059] t, t-1, t-2 시점의 요청 행렬들을 읽는다(S410).

[0060] XOR 연산을 사용하는 연산 단계들(S420, S430)을 통해, 인기 객체가 필터링된 요청 행렬인 Mt''를 출력한다(S440).

[0061] Mt''는 도시된 벤 다이어그램에서 색칠된 부분이다. 벤다이어그램에서 색칠이 되지 않은 부분이 인기있는 컨텐츠인 반면, 색이 칠해진 부분이 비인기 컨텐츠를 나타낸다. 즉, 인기 객체가 필터링된 요청 행렬은 연속된 세번의 시간 구간에서 중복되지 않는 부분들에 대한 내용들만을 표현하게 된다.

[0062] 인기있는 컨텐츠는 연속된 세번의 모니터링 시점에서 지속적으로 출몰하는 컨텐츠이다. 인기가 있으므로, 많은 사용자들로부터 반복적으로 컨텐츠 요청이 들어오기 때문이다. Mt에서 색칠된 부분은 이전의 Mt-1이나 Mt-2에서 한번도 요청된적이 없는 파일들이며, Mt-1에서 색칠된 부분은 Mt혹은 Mt-2에서 한번도 요청되지 않은 파일들이기 때문에 인기 없는 컨텐츠 객체들만 추출되게 된다.

[0063] XOR과 AND라는 간단한 연산을 사용하므로 도 6의 단계는 효율이 높다.

[0064] 도 7은 본 발명의 일실시예에 따른 캐시 공격 탐지 방법의 행렬 랭크(rank) 산출 단계의 흐름을 도시하고 있다.

[0065] 인기 객체가 필터링된 요청 행렬 Mt''를 가우스 소거법을 사용하여 상삼각 행렬로 변환하여(S510), 0이 아닌 요소가 있는 행의 개수인 행렬 랭크 r 산출한다(S520).

[0066] 가우스 소거법이라는 간단한 연산을 사용하므로 도 7의 단계는 효율이 높다.

[0067] 도 8은 본 발명의 일실시예에 따른 캐시 공격 탐지 방법의 CUSUM(cumulative sum) 산출 단계의 흐름을 도시하고 있다.

[0068] t 시점의 랭크 관찰값(rt)을 읽는다(S710). 즉, 랭크 관찰값은 각 모니터링 시점에 행렬 랭크 산출부(400)에 의

해 산출된 랭크이다.

[0069] t 시점의 랭크 기대값 평균(Et)을 산출한다(S720). 수식을 보면, t 시점의 랭크 기대값 평균은 이전 시점(t-1 시점)의 랭크 기대값 평균과 t 시점의 랭크 관찰값을 기초로 가중치(w)를 사용하여 산출됨을 알 수 있다. 가중치 w는 EWMA(exponentially weighted moving average) 가중치일 수 있다.

[0070] t 시점의 랭크 기대값 평균의 편차(deviation, gt)를 산출한다(S730). 단, 도시되어 있는 바와 같이, g0=0로 미리 설정되고, gt가 0보다 작거나 같으면 gt=0으로 재조정된다.

[0071] 본 발명의 일 실시예는 컴퓨터에 의해 실행되는 프로그램 모듈과 같은 컴퓨터에 의해 실행가능한 명령어를 포함하는 기록 매체의 형태로도 구현될 수 있다. 컴퓨터 판독 가능 매체는 컴퓨터에 의해 액세스될 수 있는 임의의 가용 매체일 수 있고, 휘발성 및 비휘발성 매체, 분리형 및 비분리형 매체를 모두 포함한다. 또한, 컴퓨터 판독 가능 매체는 컴퓨터 저장 매체 및 통신 매체를 모두 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함한다. 통신 매체는 전형적으로 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈, 또는 반송파와 같은 변조된 데이터 신호의 기타 데이터, 또는 기타 전송 메커니즘을 포함하며, 임의의 정보 전달 매체를 포함한다.

[0072] 전술한 본 발명의 설명은 예시를 위한 것이며, 본 발명이 속하는 기술분야의 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 쉽게 변형이 가능하다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다. 예를 들어, 단일형으로 설명되어 있는 각 구성 요소는 분산되어 실시될 수도 있으며, 마찬가지로 분산된 것으로 설명되어 있는 구성 요소들도 결합된 형태로 실시될 수 있다.

[0073] 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

부호의 설명

[0074] 20: 컨텐츠 서버

22: 컨텐츠 저장소

24: 컨텐츠 캐시

25: 컨텐츠 제공부

30: 사용자

32: 공격자

10: 캐시 공격 탐지 장치

100: 파라미터 설정부

200: 요청 행렬 생성부

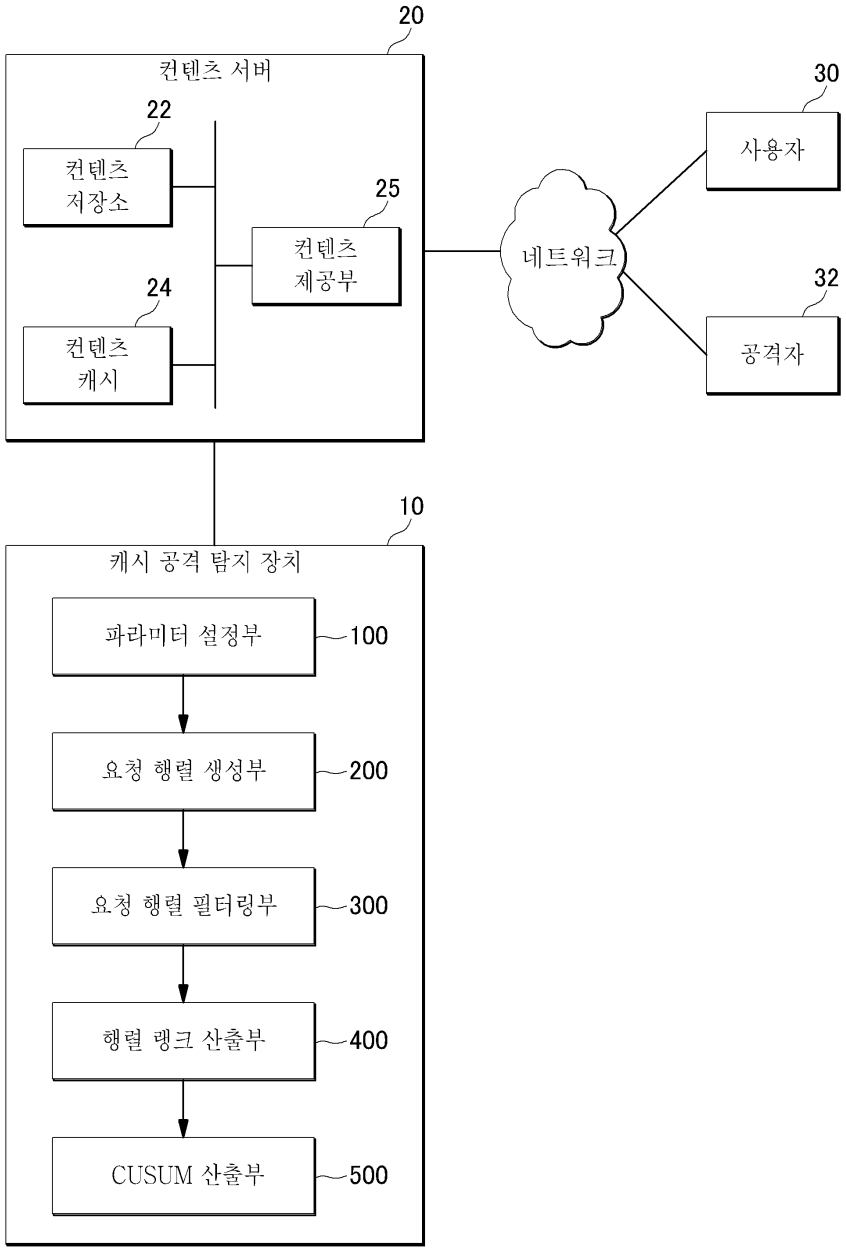
300: 요청 행렬 필터링부

400: 행렬 랭크 산출부

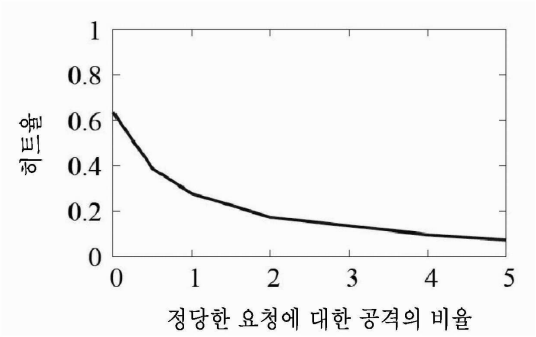
500: CUSUM 산출부

도면

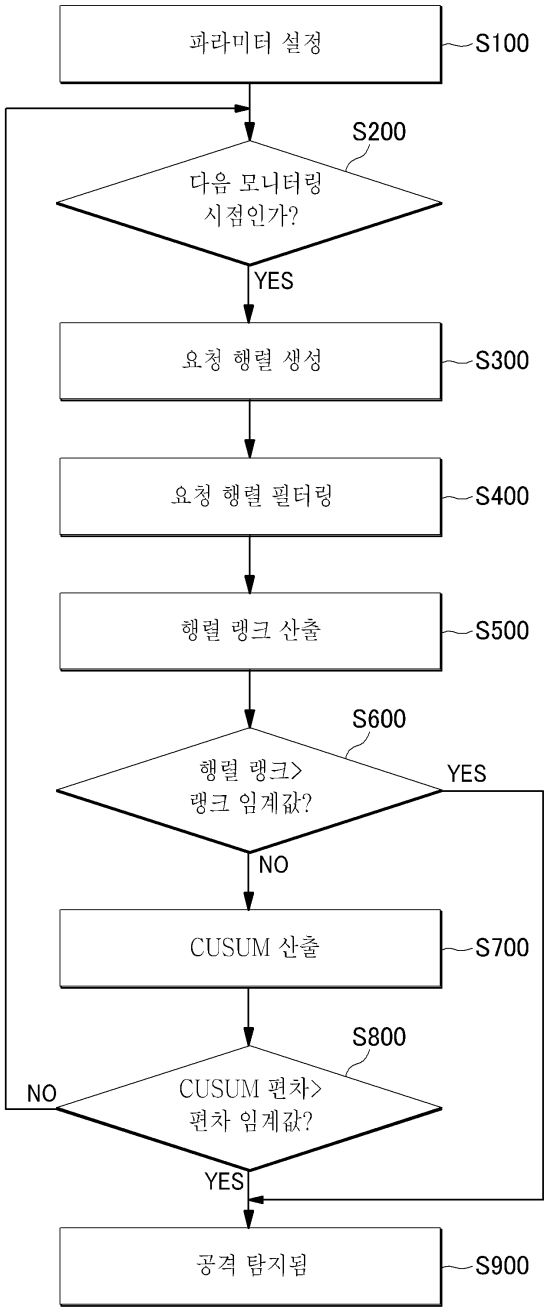
도면1


20
컨텐츠 서버
22
컨텐츠
저장소
25
컨텐츠
제공부
24
컨텐츠
캐시
30
사용자
네트워크
32
공격자
10
캐시 공격 탐지 장치
파라미터 설정부
100
요청 행렬 생성부
200
요청 행렬 필터링부
300
행렬 랭크 산출부
400
CUSUM 산출부
500

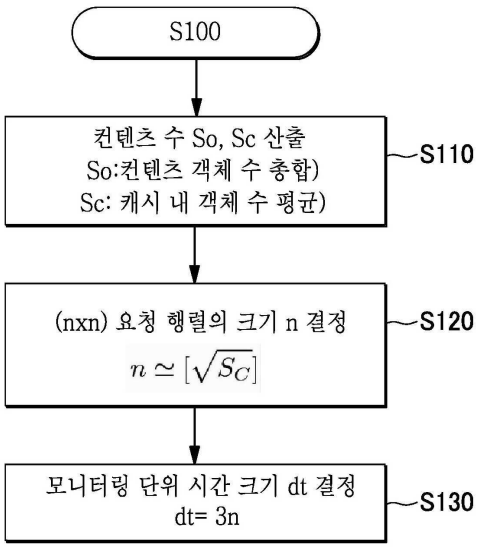
도면2



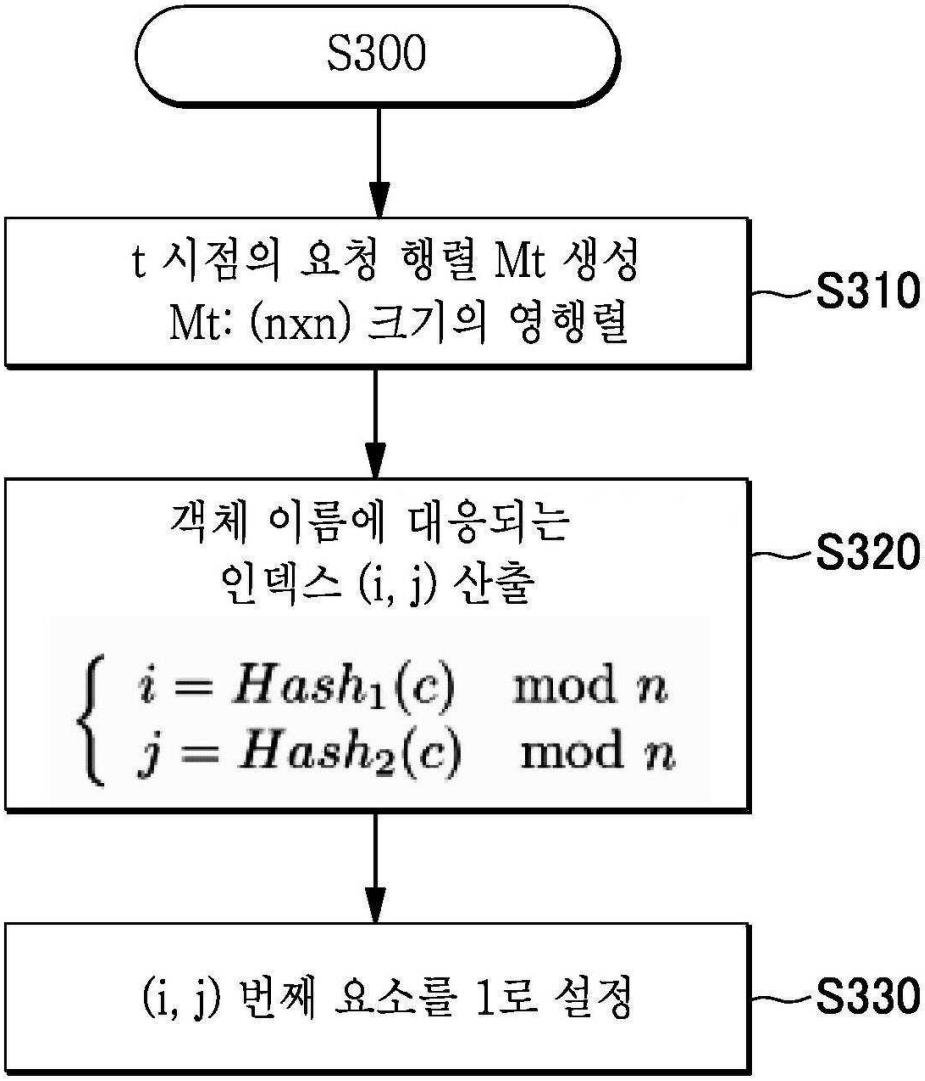
도면3


파라미터 설정
S100
S200
다음 모니터링 시점인가?
YES
요청 행렬 생성
S300
요청 행렬 필터링
S400
행렬 랭크 산출
S500
S600
행렬 랭크> 랭크 임계값?
YES
NO
CUSUM 산출
S700
S800
CUSUM 편차> 편차 임계값?
NO
YES
공격 탐지됨
S900

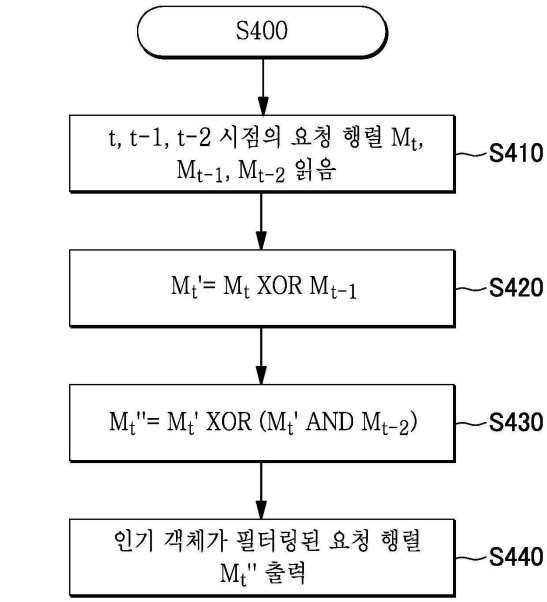
도면4

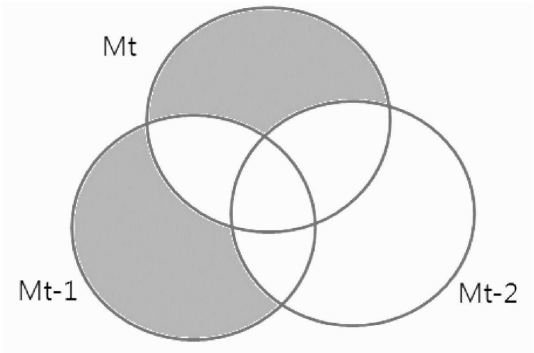

S100
컨텐츠 수 So, Sc 산출
So:컨텐츠 객체 수 총합)
Sc: 캐시 내 객체 수 평균)
S110
(nxn) 요청 행렬의 크기 n 결정
$n \simeq [\sqrt{S_C}]$
S120
모니터링 단위 시간 크기 dt 결정
dt= 3n
S130

도면5

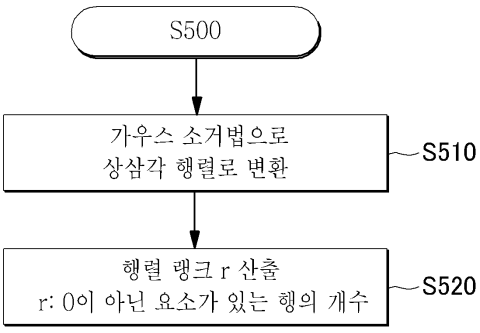

S300
t 시점의 요청 행렬 Mt 생성
Mt: (nxn) 크기의 영행렬
S310
객체 이름에 대응되는
인덱스 (i, j) 산출
$\begin{cases} i = Hash_1(c) \mod n \\ j = Hash_2(c) \mod n \end{cases}$
S320
(i, j) 번째 요소를 1로 설정
S330

도면6

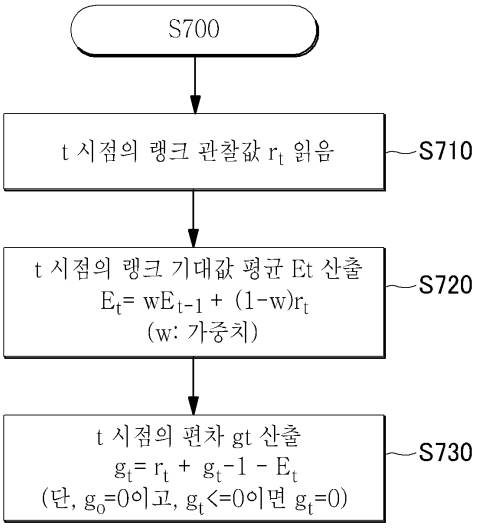

S400
t, t-1, t-2 시점의 요청 행렬 M_t, M_{t-1}, M_{t-2} 읽음
S410
$M_t' = M_t$ XOR M_{t-1}
S420
$M_t'' = M_t'$ XOR (M_t' AND M_{t-2})
S430
인기 객체가 필터링된 요청 행렬 M_t'' 출력
S440


Mt
Mt-1
Mt-2

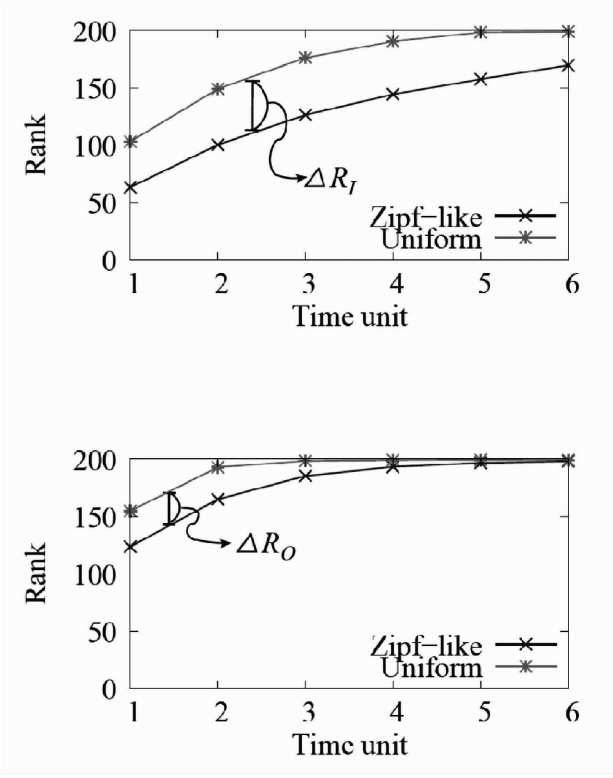
도면7


S500
가우스 소거법으로 상삼각 행렬로 변환
S510
행렬 랭크 r 산출
r: 0이 아닌 요소가 있는 행의 개수
S520

도면8


S700
t 시점의 랭크 관찰값 r_t 읽음
S710
t 시점의 랭크 기대값 평균 Et 산출
$E_t = wE_{t-1} + (1-w)r_t$
(w: 가중치)
S720
t 시점의 편차 gt 산출
$g_t = r_t + g_t - 1 - E_t$
(단, g_0=0이고, g_t<=0이면 g_t=0)
S730

도면9


Rank
Time unit
ΔR_I
Zipf-like
Uniform
ΔR_O