(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2013-0088918
(43) 공개일자 2013년08월09일

(51) 국제특허분류(Int. Cl.)
G06F 17/30 (2006.01) *G06F 12/00* (2006.01)
G06F 11/08 (2006.01)
(21) 출원번호 10-2012-0010117
(22) 출원일자 2012년02월01일
심사청구일자 2012년02월01일

(71) 출원인
이화여자대학교 산학협력단
서울특별시 서대문구 이화여대길 52 (대현동, 이화여자대학교)
(72) 발명자
임혜숙
서울특별시 강남구 도곡2동 삼성래미안아파트 108-1102
(74) 대리인
김인철

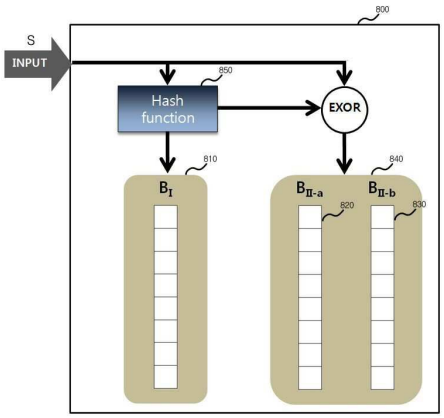
전체 청구항 수 : 총 13 항

(54) 발명의 명칭 **검증 블룸 필터를 포함하는 멀티 블룸 필터**

(57) 요 약

본 발명은 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법에 관한 것으로, 제1 블룸 필터를 이용하여 입력정보가 제1 블룸 필터에 저장한 제1 집합의 멤버에 속하는지를 확인하는 제1 확인단계; 및 상기 제1 블룸 필터의 결과가 양성(positive)인 경우, 제2 블룸 필터와 제3 블룸 필터를 포함하는 검증 블룸 필터를 이용하여 상기 양성이 거짓양성(false positive)인지를 검증하는 검증단계를 포함하되, 입력정보가 블룸 필터에 저장된 집합에 속하는 멤버인 경우, 블룸 필터의 결과는 양성을 나타내고, 입력정보가 블룸 필터에 저장된 집합에 속하지 않는 멤버인 경우, 블룸 필터의 결과는 음성을 나타내며, 상기 거짓양성은 블룸 필터의 결과가 양성임에도 불구하고 입력정보가 블룸 필터에 저장된 집합의 멤버가 아닌 경우를 표시하는 것을 특징으로 한다. 본 발명에 따르면, 블룸 필터가 지닌 간단함과 공간 효율성의 장점을 유지하면서도 블룸 필터의 성능을 향상시킬 수 있다.

대 표 도 - 도8b



이 발명을 지원한 국가연구개발사업

과제고유번호	1345148923
부처명	교육과학기술부
연구사업명	핵심연구지원사업
연구과제명	차세대 인터넷을 위한 고속 패킷 전달 기술에 관한 연구
주관기관	이화여자대학교 산학협력단
연구기간	2011.01.01 ~ 2012.02.29

특허청구의 범위

청구항 1

제1 블룸 필터를 이용하여 입력정보가 제1 블룸 필터에 저장한 제1 집합의 멤버에 속하는지를 확인하는 제1 확인단계; 및

상기 제1 블룸 필터의 결과가 양성(positive)인 경우, 제2 블룸 필터와 제3 블룸 필터를 포함하는 검증 블룸 필터를 이용하여 상기 양성이 거짓양성(false positive)인지를 검증하는 검증단계를 포함하되,

입력정보가 블룸 필터에 저장된 집합에 속하는 멤버인 경우, 블룸 필터의 결과는 양성을 나타내고, 입력정보가 블룸 필터에 저장된 집합에 속하지 않는 멤버인 경우, 블룸 필터의 결과는 음성을 나타내며, 상기 거짓양성은 블룸 필터의 결과가 양성임에도 불구하고 입력정보가 블룸 필터에 저장된 집합의 멤버가 아닌 경우를 표시하는 것을 특징으로 하는 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법.

청구항 2

제 1 항에 있어서,

상기 검증단계는, 상기 제2 블룸 필터를 이용하여 상기 입력정보가 상기 제2 블룸 필터에 저장한 제2 집합의 멤버에 속하는지를 확인하는 제2 확인단계; 및

상기 제3 블룸 필터를 이용하여 상기 입력정보가 상기 제3 블룸 필터에 저장한 제3 집합의 멤버에 속하는지를 확인하는 제3 단계를 포함하는 것을 특징으로 하는 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법.

청구항 3

제 2 항에 있어서,

상기 제1 집합은 상기 제2 집합과 상기 제3 집합의 합집합인 것을 특징으로 하는 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법.

청구항 4

제 2 항에 있어서,

상기 제2 확인단계 결과, 상기 제2 블룸 필터의 결과가 음성(negative)이고, 상기 제3 확인단계 결과, 상기 제3 블룸 필터의 결과가 음성인 경우, 상기 제1 블룸 필터의 결과가 거짓양성인 것을 특징으로 하는 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법.

청구항 5

제 2 항에 있어서,

상기 제2 집합의 멤버는 상기 제1 집합의 멤버 중 중요도, 빈도, 오류율 중 적어도 하나가 높은 멤버로 구성되는 것을 특징으로 하는 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법.

청구항 6

제 1 항에 있어서,

상기 제1 블룸 필터는 CRC 해쉬함수를 이용하여 해쉬코드를 생성하는 것을 특징으로 하는 검증 블룸 필터를 이

용한 블룸 필터의 성능 향상 방법.

청구항 7

제 6 항에 있어서,

상기 검증 블룸 필터에서 사용하는 해쉬코드는 상기 제1 블룸 필터에서 사용한 해쉬코드에 상기 입력정보를 배타적 논리합 연산(XOR 연산)하여 사용하는 것을 특징으로 하는 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법.

청구항 8

적어도 하나의 해쉬함수;

입력정보가 저장한 제1 집합의 멤버에 속하는지를 확인하는 제1 블룸 필터; 및

상기 제1 블룸 필터의 결과가 양성(positive)인 경우, 상기 양성이 거짓양성(false positive)인지를 검증하는 제2 블룸 필터와 제3 블룸 필터를 포함하는 검증 블룸 필터를 포함하되,

입력정보가 블룸 필터에 저장된 집합에 속하는 멤버인 경우, 블룸 필터의 결과는 양성을 나타내고, 입력정보가 블룸 필터에 저장된 집합에 속하지 않는 멤버인 경우, 블룸 필터의 결과는 음성을 나타내며, 상기 거짓양성은 블룸 필터의 결과가 양성임에도 불구하고 입력정보가 블룸 필터에 저장된 집합의 멤버가 아닌 경우를 표시하는 것을 특징으로 하는 검증 블룸 필터를 포함하는 멀티 블룸 필터.

청구항 9

제 8 항에 있어서,

상기 제2 블룸 필터는 상기 입력정보가 상기 제2 블룸 필터에 저장한 제2 집합의 멤버에 속하는지를 확인하고, 상기 제3 블룸 필터는 상기 입력정보가 상기 제3 블룸 필터에 저장한 제3 집합의 멤버에 속하는지를 확인하되,

상기 제1 집합은 상기 제2 집합과 상기 제3 집합의 합집합인 것을 특징으로 하는 검증 블룸 필터를 포함하는 멀티 블룸 필터.

청구항 10

제 9 항에 있어서,

상기 제2 블룸 필터의 결과가 음성(negative)이고, 상기 제3 블룸 필터의 결과가 음성인 경우, 상기 제1 블룸 필터의 결과는 거짓양성인 것을 특징으로 하는 검증 블룸 필터를 포함하는 멀티 블룸 필터.

청구항 11

제 9 항에 있어서,

상기 제2 집합의 멤버는 상기 제1 집합의 멤버 중 중요도, 빈도, 오류율 중 적어도 하나가 높은 멤버로 구성되는 것을 특징으로 하는 검증 블룸 필터를 포함하는 멀티 블룸 필터.

청구항 12

제 8 항에 있어서,

상기 해쉬함수는 CRC 해쉬함수이며, 상기 제1 블룸 필터는 상기 해쉬함수를 이용하여 해쉬코드를 생성하는 것을

특징으로 하는 검증 블룸 필터를 포함하는 멀티 블룸 필터.

청구항 13

제 12 항에 있어서,

상기 검증 블룸 필터는 상기 해쉬함수를 이용하여 생성된 해쉬코드에 상기 입력정보를 배타적 논리합 연산(XOR 연산)하여 생성된 해쉬코드를 사용하는 것을 특징으로 하는 검증 블룸 필터를 포함하는 멀티 블룸 필터.

명 세 서

기 술 분 야

[0001] 본 발명은 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법 및 검증 블룸 필터를 포함하는 멀티 블룸 필터에 관한 것으로, 더 상세하게는 검증 블룸 필터를 이용하여 블룸 필터의 거짓양성(false positive)의 비율을 줄일 수 있는 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법 및 검증 블룸 필터를 포함하는 멀티 블룸 필터에 관한 것이다.

[0002] 또한, 제1 블룸 필터에 저장한 집합을 검증 블룸 필터에 나눠서 저장함으로써, 더 작은 메모리 공간을 사용하면서 거짓양성 빈도를 줄일 수 있는 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법 및 검증 블룸 필터를 포함하는 멀티 블룸 필터에 관한 것이다.

[0003] 또한, 본 발명은 검증 블룸 필터에서 사용하는 해쉬코드는 제1 블룸 필터에서 사용한 해쉬코드에 제1 블룸 필터의 입력 정보를 배타적 논리합 연산(XOR)하여 사용함으로써 검증 블룸 필터가 추가됨에도 불구하고 종래의 블룸 필터에 비해 필요한 공간을 줄일 수 있는 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법 및 검증 블룸 필터를 포함하는 멀티 블룸 필터에 관한 것이다.

[0004] 또한, 본 발명은 검증 블룸 필터에 포함된 하나의 블룸 필터에 저장한 제2 집합의 멤버는 제1 블룸 필터에 저장한 집합의 멤버 중 중요도, 빈도, 오류율 중 적어도 하나가 높은 멤버로 구성함으로써 블룸 필터의 성능을 향상시킬 수 있는 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법 및 검증 블룸 필터를 포함하는 멀티 블룸 필터에 관한 것이다.

배 경 기 술

[0005] 개개인이 저장하고 취급하는 데이터의 양은 급격히 늘고 있으며 이에 따라 인터넷상에서의 트래픽도 증가하고 있다. 수많은 데이터를 효율적이고 안전하게 이용하기 위해 데이터를 필터링함으로써 원치않는 데이터는 차단하고 필요한 데이터는 빠르게 찾을 수 있는 애플리케이션의 개발과 사용이 증가하고 있다. 상기와 같은 목적을 위해 개발된 것이 블룸 필터이며, 블룸 필터를 이용하면 작은 메모리 공간을 이용해서 입력된 데이터가 블룸 필터가 저장하고 있는 집합의 멤버인지 아닌 지를 간단하게 알 수 있다.

[0006] 블룸 필터가 작은 메모리 공간을 이용하는 이유는 블룸 필터의 구조가 비트 벡터이고 이 비트 벡터는 블룸 필터가 저장한 집합에 포함되는 하나의 원소 당 상수 배의 비트를 할당하기 때문이다. 블룸 필터는 크기가 작으면서도 효과는 강력하다. 해쉬 함수에서 인덱스를 얻어 블룸 필터에 접근하여 상수 개의 비트를 확인하는 간단한 과정만으로 멤버쉽 쿼리를 할 수 있다.

[0007] 블룸 필터가 처음 소개된 1970년 이후로 블룸 필터는 데이터베이스 애플리케이션에서 각광을 받았고 최근에는 네트워크 분야에서도 블룸 필터의 이용이 급격히 늘고 있다. 예를 들어, 라우팅 테이블 검색, 온라인 트래픽 측량, 피어 투 피어 시스템, 웹 시스템, 방화벽 설계, 침입 탐지 등에서 블룸 필터가 다양하게 사용되고 있다. 이외에도 블룸 필터는 웹 캐쉬 공유에도 사용될 수 있는데 프록시 서버가 캐쉬된 페이지를 블룸 필터에 프로그래밍해서 이 블룸 필터를 프록시 서버를 사용하고 있는 모든 사용자에게 전달하여 자신이 저장하고 있는 페이지를 알리는 방식으로 사용될 수 있다. 상기와 같이, 블룸 필터는 멤버쉽 쿼리가 필요하다면 어디에서도 사용될 수 있기 때문에 블룸 필터의 성능 향상은 다양한 분야에 긍정적인 영향을 미칠 수 있다.

[0008] 블룸 필터의 성능은 메모리 요구량, 구조의 복잡도와 블룸 필터의 오류 비율(False Positive Rate)로 평가할 수 있는데, 블룸 필터를 이용하여 멤버쉽 쿼리를 할 경우에 발생하는 가장 큰 문제는 블룸 필터의 결과가 양성임에도 불구하고, 상기 양성이 거짓양성인 경우이다.

[0009] 즉, 블룸 필터를 이용하여 멤버쉽 쿼리를 할 경우, 블룸 필터의 결과가 양성(positive)인 경우, 이는 입력정보가 블룸 필터에 저장된 집합에 속하는 멤버임을 나타내고, 블룸 필터의 결과가 음성(negative)인 경우, 이는 입력정보가 블룸 필터에 저장된 집합에 속하는 멤버가 아님을 각각 나타내는데, 입력정보가 블룸 필터에 저장한 집합의 멤버가 아님에도 불구하고 블룸 필터에 저장한 집합의 멤버에 속하는 것으로 판단하는 거짓양성이 발생할 수 있다는 문제가 발생할 수 있다.

[0010] 상기와 같은 문제점을 극복하기 위해, 블룸 필터의 크기를 조절함으로써 거짓 양성의 비율을 줄일 수 있으나 블룸 필터가 지닌 장점인 간단함과 공간 효율성을 유지할 수 없다는 또 다른 문제를 발생시킬 수 있다.

발명의 내용

해결하려는 과제

[0011] 본 발명은 상기와 같은 문제점을 해결하기 위해 창안된 것으로서, 블룸 필터가 지닌 간단함과 공간 효율성의 장점을 유지하면서도 블룸 필터의 성능을 향상시킬 수 있는 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법 및 검증 블룸 필터를 포함하는 멀티 블룸 필터의 제공을 그 목적으로 한다.

[0012] 본 발명의 또 다른 목적은, 검증 블룸 필터를 이용하여 블룸 필터의 거짓양성의 비율을 줄일 수 있는 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법 및 검증 블룸 필터를 포함하는 멀티 블룸 필터의 제공을 그 목적으로 한다.

[0013] 본 발명의 또 다른 목적은, 제1 블룸 필터에 저장한 집합을 검증 블룸 필터에 나눠서 저장함으로써, 더 작은 메모리 공간을 사용하면서 거짓양성 빈도를 줄일 수 있는 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법 및 검증 블룸 필터를 포함하는 멀티 블룸 필터의 제공을 그 목적으로 한다.

[0014] 본 발명의 또 다른 목적은, 검증 블룸 필터에서 사용하는 해쉬코드를 제1 블룸 필터에서 사용한 해쉬코드에 제1 블룸 필터의 입력 정보를 배타적 논리합 연산(XOR)하여 사용함으로써 검증 블룸 필터가 추가됨에도 불구하고 종래의 블룸 필터에 비해 필요한 공간을 줄일 수 있는 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법 및 검증 블룸 필터를 포함하는 멀티 블룸 필터의 제공을 그 목적으로 한다.

[0015] 본 발명의 또 다른 목적은, 검증 블룸 필터에 포함된 하나의 블룸 필터에 저장한 제2 집합의 멤버는 제1 블룸 필터에 저장한 집합의 멤버 중 중요도, 빈도, 오류율 중 적어도 하나가 높은 멤버로 구성함으로써 블룸 필터의 성능을 향상시킬 수 있는 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법 및 검증 블룸 필터를 포함하는 멀티 블룸 필터의 제공을 그 목적으로 한다.

과제의 해결 수단

[0016] 상기와 같은 목적을 달성하기 위하여 본 발명에 따른 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법은, 제1 블룸 필터를 이용하여 입력정보가 제1 블룸 필터에 저장한 제1 집합의 멤버에 속하는지를 확인하는 제1 확인단계; 및 상기 제1 블룸 필터의 결과가 양성(positive)인 경우, 제2 블룸 필터와 제3 블룸 필터를 포함하는 검증 블룸 필터를 이용하여 상기 양성이 거짓양성(false positive)인지를 검증하는 검증단계를 포함하되, 입력정보가 블룸 필터에 저장된 집합에 속하는 멤버인 경우, 블룸 필터의 결과는 양성을 나타내고, 입력정보가 블룸 필터에 저장된 집합에 속하지 않는 멤버인 경우, 블룸 필터의 결과는 음성을 나타내며, 상기 거짓양성은 블룸 필터의 결과가 양성임에도 불구하고 입력정보가 블룸 필터에 저장된 집합의 멤버가 아닌 경우를 표시하는 것을 특징으로 한다.

[0017] 본 발명은, 상기 검증단계는, 상기 제2 블룸 필터를 이용하여 상기 입력정보가 상기 제2 블룸 필터에 저장한 제2 집합의 멤버에 속하는지를 확인하는 제2 확인단계; 및 상기 제3 블룸 필터를 이용하여 상기 입력정보가 상기 제3 블룸 필터에 저장한 제3 집합의 멤버에 속하는지를 확인하는 제3 단계를 포함하는 것을 특징으로 한다.

[0018] 본 발명은, 상기 제1 집합은 상기 제2 집합과 상기 제3 집합의 합집합인 것을 특징으로 한다.

[0019] 본 발명은, 상기 제2 확인단계 결과, 상기 제2 블룸 필터의 결과가 음성(negative)이고, 상기 제3 확인단계 결과, 상기 제3 블룸 필터의 결과가 음성인 경우, 상기 제1 블룸 필터의 결과가 거짓양성인 것을 특징으로 한다.

[0020] 본 발명은, 상기 제2 집합의 멤버는 상기 제1 집합의 멤버 중 중요도, 빈도, 오류율 중 적어도 하나가 높은 멤버로 구성되는 것을 특징으로 한다.

[0021] 본 발명은, 상기 제1 블룸 필터는 CRC 해쉬함수를 이용하여 해쉬코드를 생성하는 것을 특징으로 한다.

[0022] 본 발명은, 상기 검증 블룸 필터에서 사용하는 해쉬코드는 상기 제1 블룸 필터에서 사용한 해쉬코드에 상기 입력정보를 배타적 논리합 연산(XOR 연산)하여 사용하는 것을 특징으로 한다.

[0023] 상기와 같은 목적을 달성하기 위하여 본 발명에 따른 검증 블룸 필터를 포함하는 멀티 블룸 필터는, 적어도 하나의 해쉬함수; 입력정보가 저장한 제1 집합의 멤버에 속하는지를 확인하는 제1 블룸 필터; 및 상기 제1 블룸 필터의 결과가 양성(positive)인 경우, 상기 양성이 거짓양성(false positive)인지를 검증하는 제2 블룸 필터와 제3 블룸 필터를 포함하는 검증 블룸 필터를 포함하되, 입력정보가 블룸 필터에 저장된 집합에 속하는 멤버인 경우, 블룸 필터의 결과는 양성을 나타내고, 입력정보가 블룸 필터에 저장된 집합에 속하지 않는 멤버인 경우, 블룸 필터의 결과는 음성을 나타내며, 상기 거짓양성은 블룸 필터의 결과가 양성임에도 불구하고 입력정보가 블룸 필터에 저장된 집합의 멤버가 아닌 경우를 표시하는 것을 특징으로 한다.

발명의 효과

[0024] 본 발명은 다음과 같은 효과와 이점을 제공한다.

[0025] 우선, 블룸 필터가 지닌 간단함과 공간 효율성의 장점을 유지하면서도 블룸 필터의 성능을 향상시킬 수 있다.

[0026] 둘째, 검증 블룸 필터를 이용하여 블룸 필터의 거짓양성의 비율을 줄일 수 있다.

[0027] 셋째, 제1 블룸 필터에 저장한 집합을 검증 블룸 필터에 나눠서 저장함으로써, 더 작은 메모리 공간을 사용하면서 거짓양성 빈도를 줄일 수 있다.

[0028] 넷째, 검증 블룸 필터에서 사용하는 해쉬코드를 제1 블룸 필터에서 사용한 해쉬코드에 제1 블룸 필터의 입력 정보를 배타적 논리합 연산(XOR)하여 사용함으로써 검증 블룸 필터가 추가됨에도 불구하고 종래의 블룸 필터에 비해 필요한 공간을 줄일 수 있다.

[0029] 다섯째, 검증 블룸 필터에 포함된 하나의 블룸 필터에 저장한 제2 집합의 멤버는 제1 블룸 필터에 저장한 집합의 멤버 중 중요도, 빈도, 오류율 중 적어도 하나가 높은 멤버로 구성함으로써 블룸 필터의 성능을 향상시킬 수 있다.

도면의 간단한 설명

[0030] 도 1은 블룸 필터의 프로그래밍 과정을 설명하기 위한 도면이다.

도 2는 내지 도 5는 블룸 필터의 멤버쉽 쿼리를 설명하기 위한 도면이다.

도 6은 본 발명의 일실시예에 따른 멀티 블룸 필터에 포함된 해쉬 함수인 CRC 해쉬 함수의 구조를 나타낸다.

도 7은 블룸 필터의 거짓 양성 비율을 줄이기 위한 듀얼 블룸 필터의 구조를 나타내는 도면이다.

도 8은 본 발명의 일실시예에 따른 (a) 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 과정을 개략적으로 나타낸 도면, (b) 검증 블룸 필터를 포함하는 멀티 블룸 필터의 구조를 나타내는 도면이다.

도 9 내지 도 12는 본 발명에 따른 멀티 블룸 필터의 프로그래밍을 설명하기 위한 일례를 나타내고, 도 13 내지 도 15는 프로그래밍된 블룸 필터를 이용한 멤버쉽 쿼리를 설명하기 위한 일례를 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0031] 이하 첨부된 도면을 참조로 본 발명의 바람직한 실시예를 상세히 설명하기로 한다. 이에 앞서, 본 명세서 및 청구범위에 사용된 용어나 단어는 통상적이거나 사전적인 의미로 한정해서 해석되어서는 아니되며, 발명자는 그 자신의 발명을 가장 최선의 방법으로 설명하기 위해 용어의 개념을 적절하게 정의할 수 있다는 원칙에 입각하여 본 발명의 기술적 사상에 부합하는 의미와 개념으로 해석되어야만 한다. 따라서, 본 명세서에 기재된 실시예와 도면에 도시된 구성은 본 발명의 가장 바람직한 일실시예에 불과할 뿐이고 본 발명의 기술적 사상을 모두 대변하는 것은 아니므로, 본 출원시점에 있어서 이들을 대체할 수 있는 다양한 균등물과 변형예들이 있을 수 있음을 이해하여야 한다.

[0032] 특히, 본 명세서에서 정보(information)란, 값(values), 파라미터(parameters), 계수(coefficients), 성분(elements) 등을 모두 포함하는 용어로서, 경우에 따라 그 의미는 달리 해석될 수 있으므로 본 발명은 이에 한

정되지 아니한다.

[0033] 도 1은 블룸 필터의 프로그래밍 과정을 설명하기 위한 도면이다.

[0034] 블룸 필터는 비트 벡터와 해쉬 함수로 구성되며, 해쉬코드를 이용한다는 점에서는 해쉬 테이블과 유사하나 데이터를 저장하지 않고 원소의 유무만을 상수개의 비트에 저장하여 작은 공간을 차지한다는 점에서는 해쉬 테이블과 구별된다.

[0035] 해쉬 테이블의 목적은 데이터를 저장하는 것이기 때문에 일반적으로 하나의 해쉬 함수로 하나의 인덱스를 계산해서 버킷에 저장하지만 블룸 필터는 멤버쉽 쿼리가 목적이기 때문에 신뢰도를 높이기 위해서 여러 개의 해쉬 함수를 사용해서 다수의 비트에 멤버쉽을 표현할 수 있다. 블룸 필터의 해쉬 함수의 개수는 블룸 필터의 크기와 관련이 있고 블룸 필터의 성능에 영향을 미친다. 블룸 필터의 크기에 따라 최적화된 거짓 양성의 비율을 얻을 수 있는 해쉬 함수의 개수는 계산을 통해 얻을 수 있다.

[0036] 블룸 필터의 프로그래밍은 k개의 해쉬 함수를 가지는 m비트 블룸 필터 비트 벡터의 모든 비트를 0으로 초기화한 후, 저장하려는 원소의 k개의 해쉬 코드에 해당하는 k개의 블룸 필터 비트를 1로 세팅(setting)하는 과정을 집합에 포함되는 모든 원소에 대해 반복하는 과정으로 이루어진다.

[0037] 예를 들어, 도 1a에 도시된 소스 코드를 이용하여, 도 1b에 도시된 바와 같이, 입력 a_1에 대응하는 해쉬코드 1, 4에 해당하는 블룸 필터 비트를 1로 세팅하고, 입력 a_2에 대응하는 해쉬코드 6, 7에 해당하는 블룸 필터 비트를 1로 세팅할 수 있다. 즉, 입력 정보 a_1 및 a_2는 블룸 필터(100)에 저장된 집합의 멤버임을 나타낸다.

[0038] 도 2는 내지 도 5는 블룸 필터의 멤버쉽 쿼리를 설명하기 위한 도면이다.

[0039] 블룸 필터에 멤버쉽 쿼리를 하면, 입력 x가 블룸 필터에 저장한 집합의 멤버인지를 판단하여 집합에 속할 경우, 블룸 필터의 결과는 양성이고, 입력 x가 블룸 필터에 저장한 집합에 속하지 않을 경우, 블룸 필터의 결과는 음성을 나타낸다. 멤버쉽 쿼리의 과정은 상술한 프로그래밍 과정과 유사하다. 입력 x에 대하여 프로그래밍할 때와 마찬가지로 k개의 해쉬 코드를 구하고 해쉬 코드를 인덱스로 하는 k개의 비트를 확인한다. 모든 비트가 1인 경우를 양성이라고 하고 하나의 비트라도 0이면 음성이라고 한다.

[0040] 예를 들어, 도 2에 도시된 소스 코드를 이용하여 블룸 필터의 멤버쉽 쿼리를 수행할 수 있으며, 도 1 및 도 3를 참조하면, 블룸 필터(100)에 입력정보 a_1이 입력될 경우, 상술한 바와 같이 입력 a_1에 대응하는 해쉬코드는 1, 4이며, 그것에 해당하는 블룸 필터 비트가 모두 1이므로 블룸 필터(100)의 결과는 양성이다. 즉 입력정보 a_1은 블룸 필터(100)에 저장된 집합의 멤버임을 확인할 수 있다.

[0041] 도 4를 참조하면, 블룸 필터(100)에 입력정보 b가 입력될 경우, 입력 b에 대응하는 해쉬코드는 0, 7이며, 그것에 해당하는 블룸 필터 비트가 모두 1이 아니므로 블룸 필터(100)의 결과는 음성이다. 즉 입력정보 b은 블룸 필터(100)에 저장되지 않은 집합의 멤버임을 확인할 수 있다.

[0042] 블룸 필터의 오류는 멤버가 아닌 입력을 양성으로 판단하는 거짓 양성을 말하며 블룸 필터에 거짓 음성은 없다. 블룸 필터의 거짓 양성은 블룸 필터가 해쉬 함수를 사용하기 때문에 발생한다. 해쉬 함수가 함수의 입력과 출력 사이에 1 대 1 대응을 보장하지 않으면 블룸 필터의 프로그래밍과 멤버쉽 쿼리 과정에서 서로 다른 멤버, 혹은 입력이 같은 해쉬 코드를 가지고 블룸 필터의 같은 인덱스의 비트를 세팅할 수 있다. 그러므로 멤버쉽 쿼리에서 블룸 필터의 결과가 양성이어도 그것이 쿼리한 입력에 의해 프로그래밍 된 결과라고 확신할 수 없다. 블룸 필터의 결과가 양성이었으나 쿼리한 입력이 프로그래밍한 집합에 속하지 않을 때를 거짓 양성이라고 한다.

[0043] 도 5를 참조하면, 블룸 필터(100)에 입력정보 c에 대응하는 해쉬코드 4, 6에 해당하는 블룸 필터 비트가 모두 1이므로 블룸 필터(100)의 결과는 양성을 나타내었으나 입력정보 c는 블룸 필터(100)에 저장된 집합의 멤버가 아니므로 상기 양성인 결과는 거짓 양성을 나타낸다.

[0044] 블룸 필터의 프로그래밍 단계에서 집합의 모든 원소와 해쉬 코드를 1 대 1로 연결하는 해쉬 함수를 사용하여도 멤버쉽 쿼리의 입력은 집합의 크기가 매우 크거나 경우에 따라 무한집합일 수 있다. 그러나 블룸 필터가 차지하는 공간은 유한하므로 완벽한 해쉬 함수를 계산하는 것은 불가능하다.

[0045] 블룸 필터의 거짓 양성은 블룸 필터를 포함하고 있는 전체 시스템의 성능 저하를 가져올 수 있다. 즉, 블룸 필터의 결과를 이용하여 다음 작업을 수행하려고 할 때에 블룸 필터의 거짓 양성은 다음 작업에 영향을 미친다. 이를 피하기 위해서는, 블룸 필터의 양성이 참 양성인지를 확인하는 절차를 추가하여야 하는데 이러한 추가 절

차는 결국 블룸 필터에 저장한 집합의 모든 원소를 어딘가에 저장하고 비교해야 가능하기 때문에 메모리 효율성과 구조의 간단성을 모두 저하시킨다.

[0046] 상기와 같은 추가 절차를 포함하는 블룸 필터, 즉 추가적인 자료구조를 포함하는 해쉬 테이블을 사용할 경우, 추가적인 자료구조는 일반적으로 크기가 크므로 일반적으로 오프-칩 메모리를 사용하게 되는데 오프-칩 메모리는 속도가 느려 전체 시스템의 성능을 저하시키는 문제를 야기한다. 상기와 같은 문제점을 해결하기 위해서는, 전체 시스템의 성능 저하를 막기 위해 오프-칩 메모리를 사용하지 않고 거짓 양성의 빈도를 줄일 수 있어야 한다.

[0047] 간단하게는 블룸 필터의 크기를 증가시키면 거짓 양성의 개수가 줄어드나, 블룸 필터를 온칩 메모리에서 사용하기 위해서는 블룸 필터의 크기를 늘이는 것에는 한계가 있고 또 블룸 필터의 크기를 계속 늘이다 보면 블룸 필터의 크기를 늘여도 거짓 양성의 비율이 일정한 값으로 수렴하여 더 이상 줄지 않게 된다.

[0048] 해쉬 함수는 블룸 필터의 성능과 밀접한 관련이 있다. 해쉬 함수가 한 개 일 때보다 여러 개 일 때 쿼리 결과에 대한 신뢰도가 높아지지만 무분별하게 함수의 개수를 늘일 경우, 블룸 필터의 1로 세팅된 비트의 비율이 너무 높아져서 오히려 신뢰도가 떨어질 수 있다. 그러므로 블룸 필터 크기를 증가시키면서 해쉬 함수의 개수를 늘여야 거짓 양성을 줄일 수 있다. 그러나 해쉬 함수의 개수가 늘어남에 따라 구조의 복잡도도 커지는 단점이 있다.

[0049] 블룸 필터의 크기가 m, 저장한 집합의 크기가 n일 때 거짓 양성의 개수를 최소화하는 해쉬 함수의 개수 k는

$$k = \frac{m}{n} \ln 2$$

이다.

[0050] 도 6은 본 발명의 일실시예에 따른 멀티 블룸 필터에 포함된 해쉬 함수인 CRC 해쉬 함수의 구조를 나타낸다.

[0051] 도 6에 도시된 CRC 해쉬 함수는 8비트의 해쉬 코드를 생성하는 CRC-8을 나타내며, CRC 해쉬 함수는 어떠한 길이의 입력을 넣어도 정해진 길이의 해쉬 코드를 생성하며 생성된 해쉬 코드에서 임의로 일부의 비트를 추출하여 사용할 수 있다. 블룸 필터의 크기가 증가함에 따라 필요한 인덱스의 개수도 증가하지만 CRC 해쉬 함수를 사용하면 해쉬 코드에 많은 양의 비트를 할당하지 않아도 정해진 비트 내에서 추출 방법에 따라 얼마든지 인덱스를 얻을 수 있다. 본 발명의 일실시예에 따른 멀티 블룸 필터는 CRC-8 해쉬 함수를 사용하나 본 발명은 이에 한정되지 아니한다. 즉, 본 발명은 64 비트의 해쉬 코드를 생성하는 CRC-64 해쉬 함수를 사용할 수 있음은 당업자에게 자명하다.

[0052] 원소의 개수가 n인 집합을 프로그래밍하는 블룸 필터의 크기는 N'의 배수가 되며, N'은 하기 수학식 1을 통해 계산된다.

수학식 1

[0053]
$$N' = 2^{\lceil \log_2(n) \rceil}$$

[0054] 상기 수학식 1에 따라 블룸 필터를 생성하면 1N'의 블룸 필터를 사용하는 경우 집합이 포함하고 있는 하나의 원소 당 할당되는 비트 수는 최소 1개이며 2개 미만이다. 2N'의 블룸 필터를 사용하는 경우, 2비트 이상, 4비트 미만이 할당된다.

[0055] 도 7은 블룸 필터의 거짓 양성 비율을 줄이기 위한 듀얼 블룸 필터의 구조를 나타내는 도면이다.

[0056] 도 7을 참조하면, 듀얼 블룸 필터는 블룸 필터의 거짓 양성 확률을 감소시키기 위해 하나의 블룸 필터를 사용하는 기존의 방식 대신 직렬로 연결된 두 개의 블룸 필터(700, 710)를 사용한다. 첫 번째 블룸 필터(700)는 일반적인 k개의 해쉬 함수를 사용하는 m비트 비트 벡터의 블룸 필터이다. 두 번째 블룸 필터(710)는 첫 번째 블룸 필터(700)와 다른 k개의 새로운 해쉬 함수를 사용하는 m비트 비트 벡터의 블룸 필터이다.

[0057] 크기가 n인 주어진 집합에 대해 양 블룸 필터(700, 710)를 기존의 블룸 필터와 같은 방법으로 프로그래밍한다. 멤버쉽 쿼리 과정에서는 입력 S의 멤버쉽 여부를 검증하기 위해, 먼저 첫 번째 블룸 필터(700)를 사용하여 k개의 해쉬 코드를 생성하고 m비트 벡터의 해쉬 코드에 해당하는 인덱스를 갖는 k개의 비트를 확인한다. 만일 첫 번째 블룸 필터(700)가 S가 멤버가 아니라고 판단하면 멤버쉽 쿼리는 종료된다. 반대로 첫 번째 블룸 필터(70

0)가 S가 멤버가 맞다고 판단하면, S를 두 번째 블룸 필터(710)로 검증하고 그 결과에 따라 S는 블룸 필터에 속한 멤버 혹은 멤버가 아닌 것으로 판명된다.

[0058] 한편, 거짓 양성의 비율을 줄이기 위해 두 번째 블룸 필터(710)는 첫 번째 블룸 필터(700)와 전혀 다른 비트 벡터를 가져야 한다. 그러기 위해서 두 블룸 필터(700, 710)의 해쉬 함수가 달라야 블룸 필터 내에서의 비트 1의 분포가 달라진다.

[0059] 그러나 새로운 해쉬 함수를 갖는 것은 추가적인 공간과 하드웨어 자원을 요구하므로, 최소한의 구현 비용으로 새로운 해쉬 함수를 사용하기 위해 입력의 비트 0를 비트 1로, 비트 1을 비트 0으로 바꾼 다음 첫 번째 블룸 필터(700)에서 사용한 해쉬 함수로 해쉬 코드를 계산한다. 상기와 같은 방식으로 작은 메모리 공간을 추가하여 오프칩 메모리를 필요로 하지 않으면서 거짓 양성의 비율을 줄일 수 있다.

[0060] 도 8은 본 발명의 일실시예에 따른 (a) 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 과정을 개략적으로 나타낸 도면, (b) 검증 블룸 필터를 포함하는 멀티 블룸 필터의 구조를 나타내는 도면이다.

[0061] 본 발명의 일실시예에 따른 멀티 블룸 필터는 블룸 필터의 공간 효율성과 간단한 구조를 유지하면서 거짓 양성의 빈도를 줄일 수 있으며, 3개의 블룸 필터와 하나의 해쉬 함수로 이루어져 있고 다음과 같은 성질을 이용한다.

[0062] A = B ∪ C 일 때
Query(B_A, x) 가 양성 이면 Query(B_B, x)는 양성 혹은 Query(B_C, x)는 양성이다.
Query(B_A, x) 가 음성 이면 Query(B_B, x)는 음성 이고 Query(B_C, x)는 음성이다.

[0063] 여기서, A, B, C는 집합이고 B_S는 집합 S에 대한 블룸 필터를 나타내며, x는 임의의 입력을 나타낸다.

[0064] 상기 성질에 따라 Query(B_A, x)가 양성이고 Query(B_B, x)는 음성, Query(B_C, x) 또한 음성이면 B_A의 쿼리 결과는 거짓 양성으로 판명된다. 이것을 이용해 멀티 블룸 필터는 하나의 블룸 필터에서 발생하는 거짓 양성을 나머지 두 개의 블룸 필터로 검증하여 줄일 수 있다.

[0065] 도 8a에 도시된 바와 같이, 본 발명에 따른 멀티 블룸 필터는 먼저 제1 블룸 필터(810)를 이용하여 입력정보(S)가 제1 블룸 필터에 저장한 제1 집합의 멤버에 속하는지를 확인하고, 상기 제1 블룸 필터(810)의 결과가 양성인 경우, 제2 블룸 필터(820)와 제3 블룸 필터(830)를 포함하는 검증 블룸 필터(840)를 이용하여 상기 양성이 거짓양성인지를 검증한다. 즉, 제1 블룸 필터(810)의 결과가 양성임에도, 제2 블룸 필터(820)의 결과와 제3 블룸 필터(830)의 결과가 모두 음성이면 상기 제1 블룸 필터의 결과는 거짓양성으로 판명된다.

[0066] 도 8b를 참조하면, 본 발명의 일실시예에 따른 멀티 블룸 필터(800)는 적어도 하나의 해쉬함수(850), 입력정보(S)가 저장한 제1 집합의 멤버에 속하는지를 확인하는 제1 블룸 필터(810) 및 상기 제1 블룸 필터의 결과가 양성인 경우, 상기 양성이 거짓양성인지를 검증하는 제2 블룸 필터(820)와 제3 블룸 필터(830)를 포함하는 검증 블룸 필터(840)를 포함한다.

[0067] 상기 제2 블룸 필터(820)는 상기 입력정보(S)가 상기 제2 블룸 필터(820)에 저장한 제2 집합의 멤버에 속하는지를 확인하고, 상기 제3 블룸 필터(830)는 상기 입력정보(S)가 상기 제3 블룸 필터(830)에 저장한 제3 집합의 멤버에 속하는지를 확인하며, 상기 제1 집합은 상기 제2 집합과 상기 제3 집합의 합집합이다.

[0068] 멀티 블룸 필터(800)의 성능을 향상시키기 위해 검증 블룸 필터(840)는 제1 블룸 필터(810)와 다른 해쉬 함수를 사용하는 것이 바람직하다. 그러나 공간과 과정을 절약하기 위해 다른 해쉬 함수를 추가하지 않고 제1 블룸 필터(810)에서 사용한 해쉬 코드에 입력정보(S)를 배타적 논리합 연산(XOR 연산)하여 해쉬 코드로 사용할 수 있다. 서로 다른 해쉬 함수를 사용하는 경우 두 가지 해쉬 함수에서 동시에 거짓 양성이 발생해야 최종적으로 거짓 양성으로 판단되므로 그 확률이 매우 낮아진다.

[0069] 본 발명에 따른 멀티 블룸 필터(800)는 종래의 블룸 필터보다 두 개의 블룸 필터가 더 추가되지만 낮은 거짓 양성 빈도를 달성하기 위해 필요한 공간은 더 작다. 그 이유는 블룸 필터 간의 크기 차이에 있다. 즉, 본 발명에 따른 멀티 블룸 필터(800)는 제1 블룸 필터(810)에 저장한 집합을 두 집합으로 나눠서 검증 블룸 필터(840)에 저장하였다. 블룸 필터의 크기는 저장한 멤버 수에 비례하므로 제1 블룸 필터(810)의 크기는 검증 블룸 필터(840)의 두 블룸 필터(820, 830)보다 일반적으로 더 크다. 하나의 블룸 필터를 사용할 때는 거짓 양성 빈도를

줄이기 위해 크기가 큰 제1 블룸 필터의 크기를 증가시켜야하므로 블룸 필터를 2 배로 증가시킬 때마다 많은 양의 공간이 필요하다. 그러나 제1 블룸 필터(810)의 크기를 증가시키는 대신 그에 비해 크기가 작은 검증 블룸 필터(840)의 크기를 증가시킴으로써 거짓 양성의 빈도를 줄이면 더 작은 메모리 공간을 사용하면서 거짓 양성 빈도를 줄일 수 있다.

[0070] 나아가 본 발명에 따른 멀티 블룸 필터(800)는 제1 블룸 필터(810)에 저장한 집합을 두 집합으로 나누되, 제2 블룸 필터(820)에 저장한 제2 집합의 멤버는 제1 집합의 멤버 중 중요도, 빈도, 오류율 중 적어도 하나가 높은 멤버로 구성함으로써 블룸 필터의 성능을 더 향상시킬 수 있다. 예컨대, 제2 집합을 치명적인 바이러스, 자주 발견되는 바이러스 또는 거짓 양성의 원인이 되는 바이러스 중 상위 10%에 해당하는 멤버로 구성함으로써 블룸 필터의 성능을 더 향상시키는 것이 가능하다.

[0071] 도 9 내지 도 12는 본 발명에 따른 멀티 블룸 필터의 프로그래밍을 설명하기 위한 일례를 나타내고, 도 13 내지 도 15는 프로그래밍된 블룸 필터를 이용한 멤버쉽 쿼리를 설명하기 위한 일례를 나타낸다.

[0072] 도 9에 도시된 P0, P1,...,P11은 본 발명에 따른 멀티 블룸 필터(800)에 프로그래밍할 집합을 나타낸다. 블룸 필터는 IP 주소의 검색에 이용할 수 있기 때문에 간단한 IP 검색 구조로 멀티 블룸 필터의 프로그래밍과 멤버쉽 쿼리를 설명할 수 있다. 도 9a는 네트워크 상에서 노드의 주소를 나타내는 프리픽스의 집합을 나타낸다. 프리픽스의 집합으로 도 9b에 도시된 것과 같은 이진 트라이를 구성할 수 있다. 트라이의 각 노드는 도 9a에 도시된 비트로 이루어진 경로를 갖는다. 상기 경로가 멀티 블룸 필터를 프로그래밍할 때 사용할 값이다.

[0073] 본 발명에 따른 멀티 블룸 필터(800)에서는 프로그래밍 집합을 검증 블룸 필터(840)에 저장할 때 두 집합으로 나눈다. 이진 트라이의 노드는 두 가지 종류로 나눌 수 있는데 노드의 경로가 프리픽스 값과 같은 검은 노드를 프리픽스 노드라고 하고 나머지 흰색 노드를 인터널 노드라고 한다. 이진 트라이를 프리픽스 노드 집합과 인터널 노드 집합으로 나눠 각각이 검증 블룸 필터에 저장할 집합이 된다.

[0074] 멀티 블룸 필터(800)를 프로그래밍하기 위해 모든 블룸 필터를 초기화하고 제1 블룸 필터(810)에는 이진 트라이의 프리픽스 노드와 인터널 노드 모두를 프로그래밍한다. 검증 블룸 필터(840)에는 트라이의 노드를 두 집합으로 나누어 프로그래밍하는데 제2 블룸 필터(820)에는 프리픽스 노드를, 제3 블룸 필터(830)에는 인터널 노드를 프로그래밍한다. 이렇게 하면 검증 블룸 필터(840)에 저장한 집합의 합집합이 제1 블룸 필터(810)에 저장한 집합이 된다.

[0075] 멀티 블룸 필터(800)의 각 블룸 필터(810, 820, 830)에 집합의 원소들을 프로그래밍하는 과정은 기존의 블룸 필터와 같다. 이진 트라이의 전체 노드 수는 34개이고 이 중 가장 상단에 있는 노드는 경로가 없기 때문에 블룸 필터에 저장하지 않는다. 즉, 제1 블룸 필터에는 33개의 원소를 프로그래밍한다. 이진 트라이의 프리픽스 노드의 수는 도 9a에 도시된 바와 같이 프리픽스 개수와 같은 12개이다. 그러므로 제2 블룸 필터(820)에는 12개의 원소를 저장하고 제3 블룸 필터(830)에는 21개의 원소를 저장한다. 블룸 필터의 크기는 상술한 수학식 1에 따라 할당하여 각각 64비트, 16비트, 32비트이다.

[0076] 각 블룸 필터의 집합에 속하는 모든 노드를 해쉬 함수를 이용해서 프로그래밍한다. 이때 검증 블룸 필터(840)는 해쉬 함수의 해쉬 코드와 저장하려는 노드의 경로를 XOR 연산하여 나온 결과를 해쉬 코드로 사용한다.

[0077] 도 10은 이진 트라이의 노드 중 경로가 000010인 노드를 프로그래밍하는 과정을 나타낸 도면이다. 상기 000010 노드는 프리픽스 노드이므로 검증 블룸 필터(840)의 두 블룸 필터 중 제2 블룸 필터(820)에 저장한다. CRC 해쉬 함수(850)에 000010을 입력하여 생성된 해쉬 코드에서 6비트를 추출하여 제1 블룸 필터(810)의 한 비트를 1로 세팅한다. 해쉬 코드를 다시 000010과 XOR한 후 제2 블룸 필터(820)를 프로그래밍하기 위해 4비트를 추출한다. 제1 블룸 필터(810)와 제2 블룸 필터(820)의 인덱스 62와 인덱스 15 비트가 1로 프로그래밍되어 노드 000010의 프로그래밍이 끝났다.

[0078] 도 11은 인터널 노드인 노드 000010을 프로그래밍하는 과정을 나타낸 도면이다. 이 경우는 제1 블룸 필터(810)와 제3 블룸 필터(830)가 프로그래밍된다.

[0079] 도 12는 모든 노드에 대해서 위와 같은 프로그래밍 과정을 거쳐 완성된 블룸 필터를 나타낸다.

[0080] 도 13을 참조하면, 멀티 블룸 필터의 멤버쉽 쿼리는 먼저 제1 블룸 필터(810) 대한 쿼리를 진행한 후 그 결과가 양성일 때에만 검증 블룸 필터(840)로 검증하는 단계를 갖는다. 검증 블룸 필터(840)는 거짓 양성을 검증하기 위한 것이기 때문에 제1 블룸 필터(810)에서 음성의 결과가 나온 경우 검증 블룸 필터(840)는 쿼리하지 않고 그대로 음성의 결과를 최종 결과로 한다. 검증 블룸 필터(840)를 쿼리할 때에 두 블룸 필터(820, 830)는 병렬적으

로 검색한다.

[0081] 쿼리할 입력은 어떤 비트 스트림도 될 수 있다. 도 13을 참조하면, 양성의 결과가 나오는 과정을 알기 위해 프로그래밍한 집합에 속하는 "1"을 쿼리한다. 해쉬 함수(850)로부터 구한 제1 블룸 필터(810)의 인덱스는 62이다. 그런데 제1 블룸 필터(810)의 해당 비트가 1이므로 검증 블룸 필터(840)를 쿼리하는 다음 단계로 진행한다. 검증 블룸 필터(840)를 위한 해쉬 인덱스를 계산하여 인덱스 0과 0을 얻었고 두 블룸 필터(820, 830)에 접근하여 제3 블룸 필터(830)에서 양성이 확인되었다. 프로그래밍 단계에서 노드 "1"은 인터널 노드였으므로 제3 블룸 필터(830)에서만 양성의 결과가 나왔다. 최종 결과는 양성이며 인터널 노드임도 알 수 있다. 이와 같이 멀티 블룸 필터(800)를 사용하는 경우 멤버쉽의 여부뿐만 아니라 검증 블룸 필터(840)의 어떤 블룸 필터에서 양성의 결과가 나왔는가에 따라 세부적인 멤버쉽, 다시 말해 입력의 분류 또한 알 수 있다.

[0082] 도 14는 입력이 프로그래밍 집합에 속하지 않는 경우이다. 쿼리 결과 제1 블룸 필터(810)의 결과가 양성이 나왔다. 그러나 검증 블룸 필터(840)를 거친 결과 음성으로 판단되었고 처음의 양성은 거짓 양성임을 확인할 수 있다.

[0083] 도 15는 백 본 라우터 MAE-WEST1의 라우팅 테이블 데이터를 이용하여 본 발명에 따른 멀티 블룸 필터 구조의 성능을 실험한 결과를 나타내는 도면이다.

[0084] 상기 라우터에서 14553개의 프리픽스를 추출하여 이진 트라이를 구성하였으며, 이진 트라이의 전체 노드는 제1 블룸 필터에, 프리픽스 노드는 제2 블룸 필터에, 인터널 노드는 제3 블룸 필터에 각각 저장하였다. 하기 표 1은 멀티 블룸 필터의 각 블룸 필터에 저장된 집합의 크기와 블룸 필터의 크기를 보여준다.

표 1

[0085]

	집합 크기	블룸 필터(N')
제1 블룸 필터	76989	131072
제2 블룸 필터	14553	16384
제3 블룸 필터	62436	65536

[0086] 멤버쉽 쿼리의 입력은 블룸 필터에 저장되지 않은 길이가 다양한 비트 스트림 538916개를 가지고 실험하였다. 이것은 블룸 필터 프로그래밍 집합의 7배에 해당한다. 모두 저장되지 않은 비트 스트림이므로 블룸 필터 결과는 음성의 결과가 기대되며 양성의 결과가 나올 경우 거짓 양성으로 판단된다.

[0087] 도 15는 본 발명에 따른 멀티 블룸 필터(800)와 기존의 단일 블룸 필터의 성능 비교 결과를 나타낸다. 도 15a를 참조하면, 단일 블룸 필터의 크기를 2048KB까지 증가시켰을 때의 거짓 양성의 개수보다 훨씬 더 적은 거짓 양성 개수를 본 발명에 따른 멀티 블룸 필터(800)는 1664KB의 메모리만을 사용하여 얻을 수 있다. 또한, 단일 블룸 필터를 사용했을 때는 거짓 양성의 개수가 300개 이상으로 수렴하는 양상을 보이지만 본 발명에 따른 멀티 블룸 필터(800)는 3584KB를 사용하였을 때 거짓 양성의 개수가 0개가 된다. 그러므로, 본 발명에 따른 멀티 블룸 필터(800)는 단일 블룸 필터보다 더 적은 메모리 요구량으로 더 적은 거짓 양성을 만든다. 그 뿐아니라 크기를 증가시켰을 때 최종적으로 수렴하는 거짓 양성 개수 또한 단일 블룸 필터보다 적기 때문에 거짓 양성의 개수가 메모리 요구량보다 더 중요한 시스템에서 사용하기에도 적합하다.

[0088] 도 15b를 참조하면, 도 15a에서보다 많은 데이터와 정확한 수치를 확인할 수 있다. 도 15b에서는 구조에 상관없이 성능이 좋은 블룸 필터를 나타내기 위하여 거짓 양성 개수의 오름차순으로 정렬하였다. 거짓 양성의 개수가 같은 경우 메모리 요구량의 오름차순으로 정렬한다. 단일 블룸 필터를 4N' 크기로 사용하는 경우 비슷한 크기의 멀티 블룸 필터보다 거짓 양성의 개수가 적지만 거짓 양성의 개수는 2만 4천여 개로 전체 입력의 4%이다. 거짓 양성 빈도를 줄이기 위해 블룸 필터 크기를 8N', 16N', 32N'으로 증가시킬 경우 비슷한 크기의 멀티 블룸 필터가 성능이 월등히 더 좋은 것을 알 수 있다.

[0089] 상술한 바와 같이, 본 발명은 블룸 필터가 지닌 간단함과 공간 효율성의 장점을 유지하면서도 블룸 필터의 성능을 향상시킬 수 있는 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법 및 검증 블룸 필터를 포함하는 멀티 블룸 필터를 제공할 수 있다.

[0090] 또한, 본 발명은 검증 블룸 필터를 이용하여 블룸 필터의 거짓양성의 비율을 줄일 수 있는 검증 블룸 필터를 이

용한 블룸 필터의 성능 향상 방법 및 검증 블룸 필터를 포함하는 멀티 블룸 필터를 제공할 수 있다.

[0091] 또한, 본 발명은 제1 블룸 필터에 저장한 집합을 검증 블룸 필터에 나눠서 저장함으로써, 더 작은 메모리 공간을 사용하면서 거짓양성 빈도를 줄일 수 있는 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법 및 검증 블룸 필터를 포함하는 멀티 블룸 필터를 제공할 수 있다.

[0092] 또한, 본 발명은 검증 블룸 필터에서 사용하는 해쉬코드를 제1 블룸 필터에서 사용한 해쉬코드에 제1 블룸 필터의 입력 정보를 배타적 논리합 연산(XOR)하여 사용함으로써 검증 블룸 필터가 추가됨에도 불구하고 종래의 블룸 필터에 비해 필요한 공간을 줄일 수 있는 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법 및 검증 블룸 필터를 포함하는 멀티 블룸 필터를 제공할 수 있다.

[0093] 또한, 본 발명은 검증 블룸 필터에 포함된 하나의 블룸 필터에 저장한 제2 집합의 멤버는 제1 블룸 필터에 저장한 집합의 멤버 중 중요도, 빈도, 오류율 중 적어도 하나가 높은 멤버로 구성함으로써 블룸 필터의 성능을 향상시킬 수 있는 검증 블룸 필터를 이용한 블룸 필터의 성능 향상 방법 및 검증 블룸 필터를 포함하는 멀티 블룸 필터를 제공할 수 있다.

[0094] 이상과 같이, 본 발명은 비록 한정된 실시예와 도면에 의해 설명되었으나, 본 발명은 이것에 의해 한정되지 않으며 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 본 발명의 기술사상과 아래에 기재될 특허청구범위의 균등범위 내에서 다양한 수정 및 변형이 가능함은 물론이다.

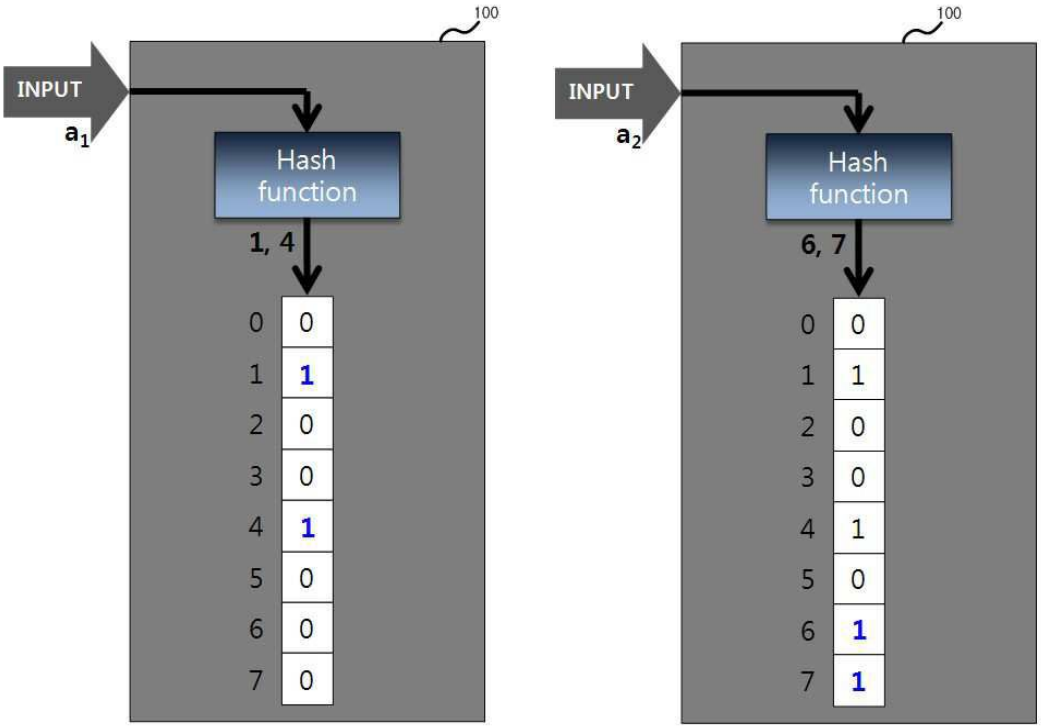
도면

도면1a

$S = \{a_1, a_2, \ldots, a_N\}$, H(x) : 해쉬 함수
B : 블룸 필터, B[x] : 블룸 필터의 x 번째 비트

```
Program(B, S)
{
        for( i is 1 to N)
        {
                B[H(a_i)_1] = 1;
                ...
                B[H(a_i)_k] = 1;
         }
}
```

도면1b

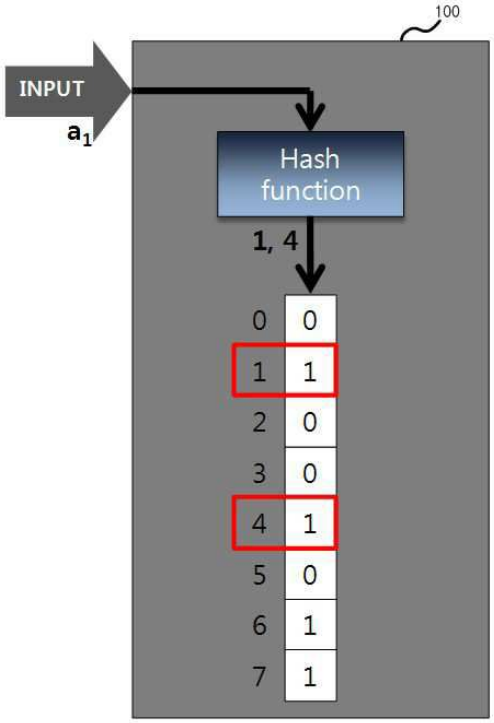


도면2

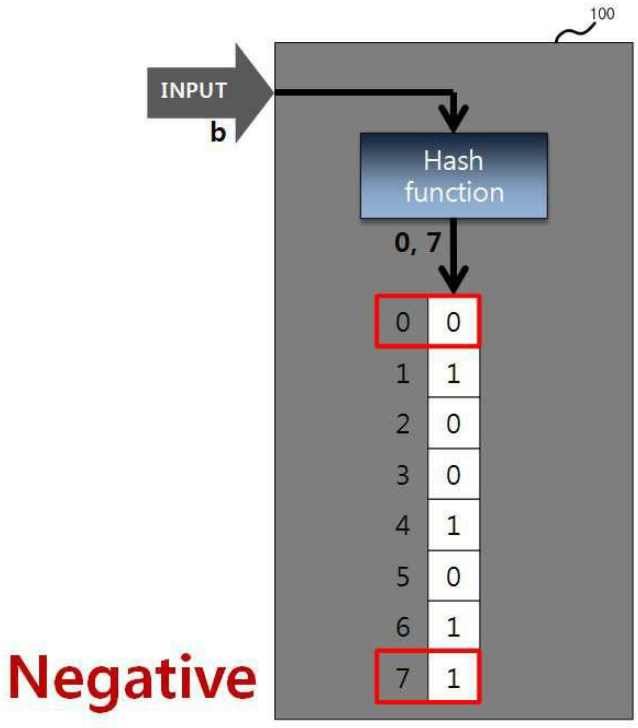
```
x : 인풋, H(x) : 해쉬 함수
B : 블룸 필터, B[x] : 블룸 필터의 x 번 째 비트

Query(B, x)
{
        if ( B[H(x)1 ] = 1 & ···& B[H(x)k] = 1)
                return Positive
        else
                return Negative
}
```

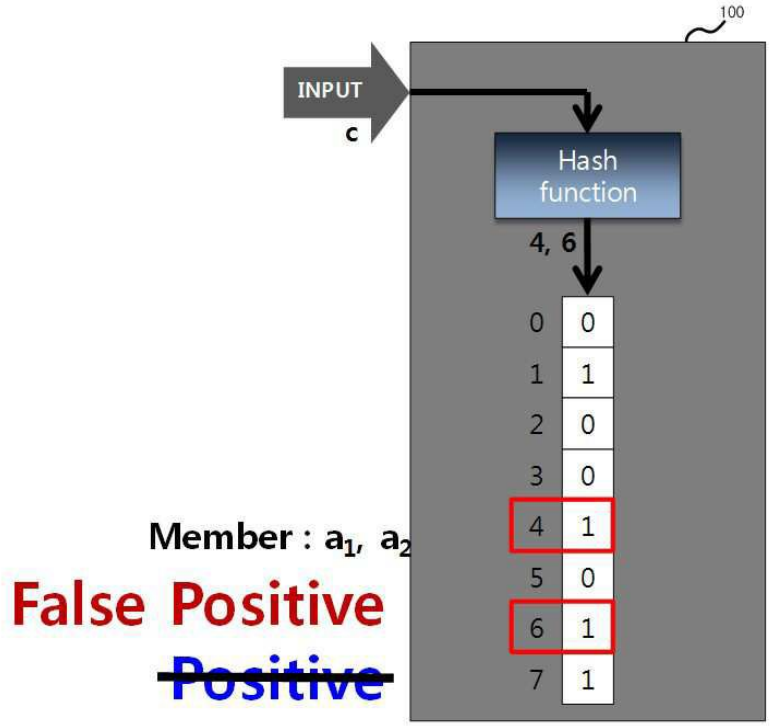
도면3


100
INPUT
a_1
Hash function
1, 4
0 0
1 1
2 0
3 0
4 1
5 0
6 1
7 1

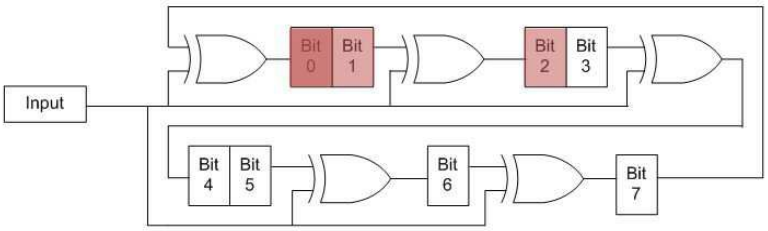
도면4


100
INPUT
b
Hash function
0, 7
0 0
1 1
2 0
3 0
4 1
5 0
6 1
7 1
Negative

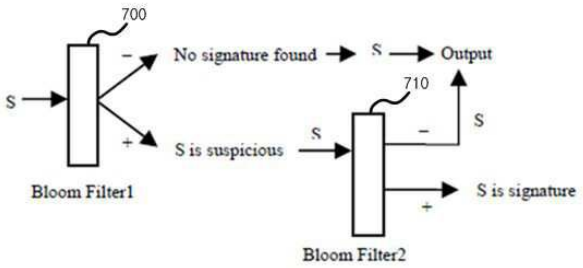
도면5


100
INPUT
c
Hash function
4, 6
0 0
1 1
2 0
3 0
4 1
5 0
6 1
7 1
Member : a_1, a_2
False Positive
Positive

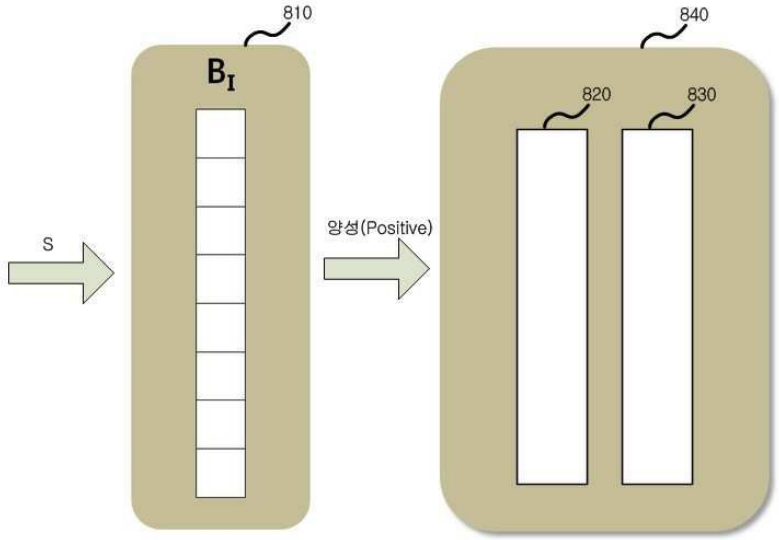
도면6


Input
Bit 0
Bit 1
Bit 2
Bit 3
Bit 4
Bit 5
Bit 6
Bit 7

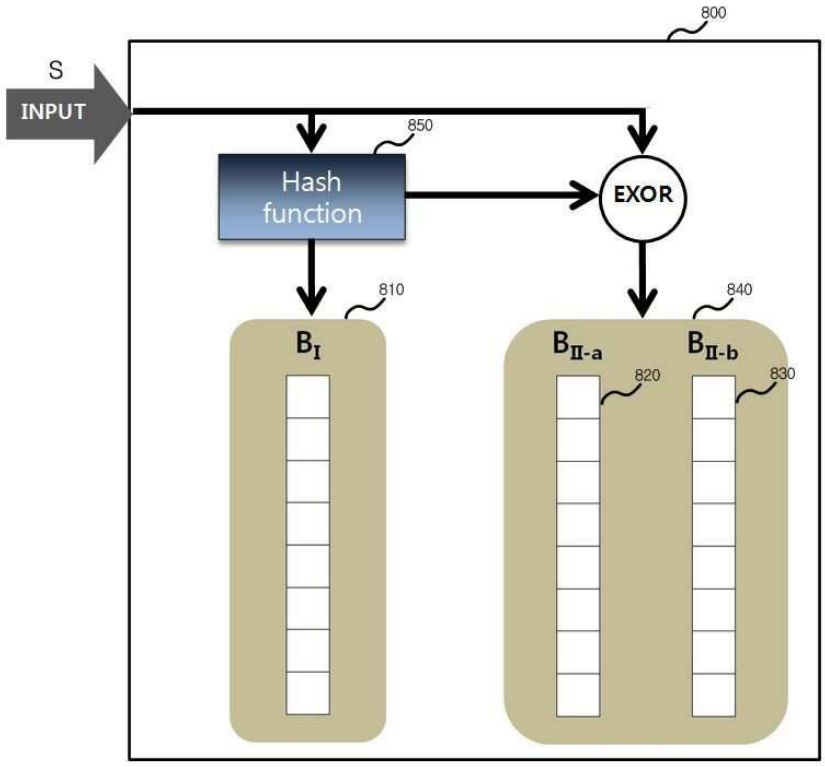
도면7


700
S
No signature found
S
Output
S is suspicious
Bloom Filter1
710
S
S
S is signature
Bloom Filter2

도면8a


810
B_I
840
820
830
S
양성(Positive)

도면8b

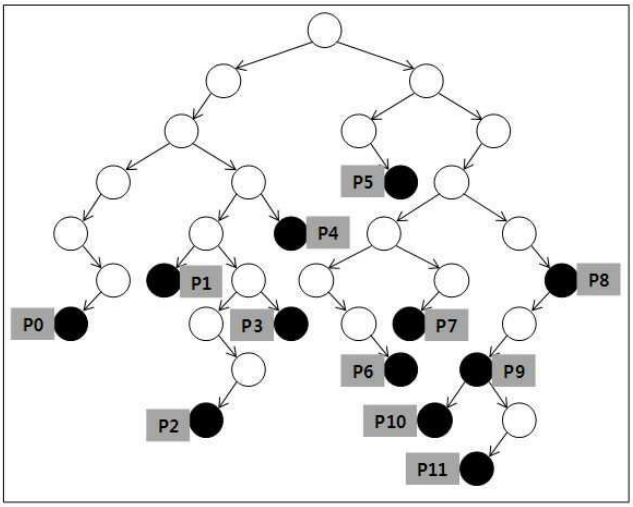

800
S
INPUT
850
Hash function
EXOR
810
B_I
840
B_{II-a}
B_{II-b}
820
830

도면9

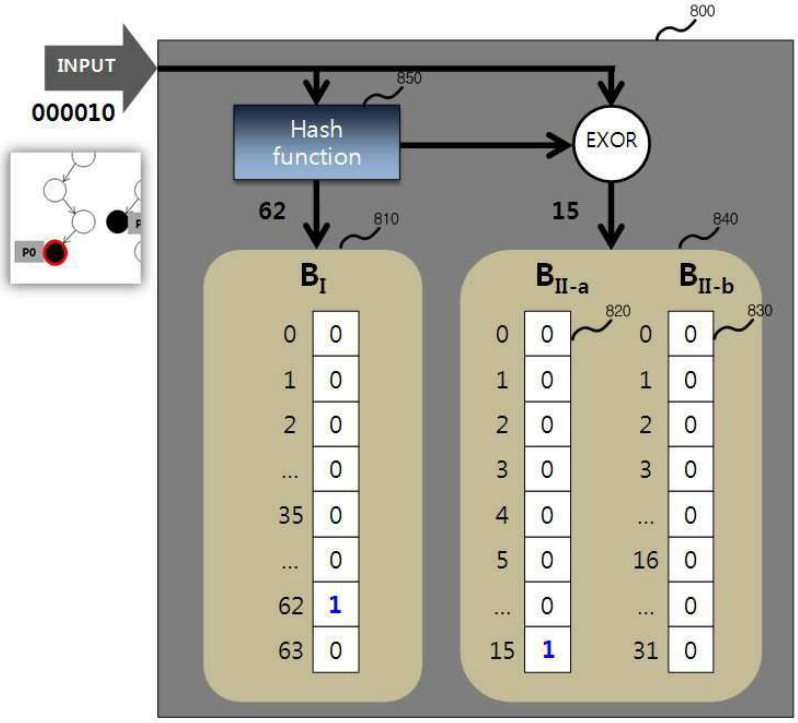
(a)

	Prefix
P0	000010
P1	00100
P2	00101010
P3	001011
P4	0011
P5	101
P6	1100011
P7	110010
P8	11011
P9	1101100
P10	11011000
P11	110110010

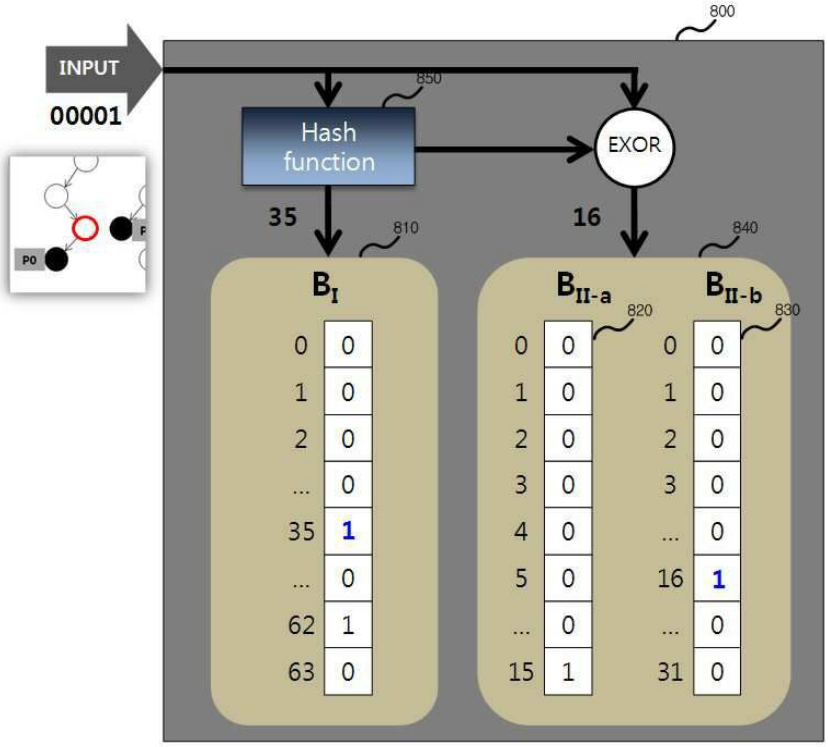
(b)



도면10



도면11



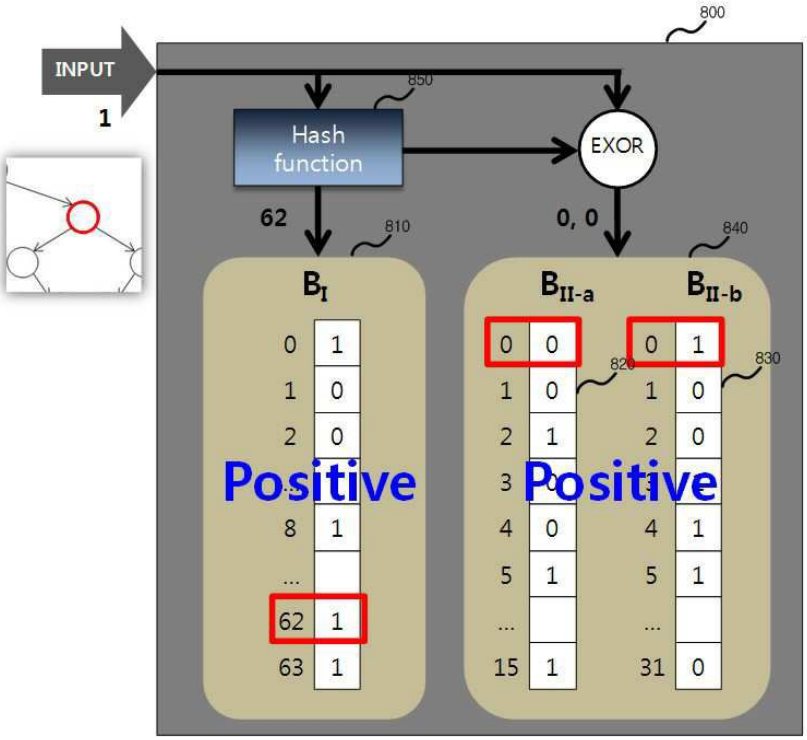
도면12

제1 블룸 필터: 10000000100010101000101001011111100110001000000000001110100100011

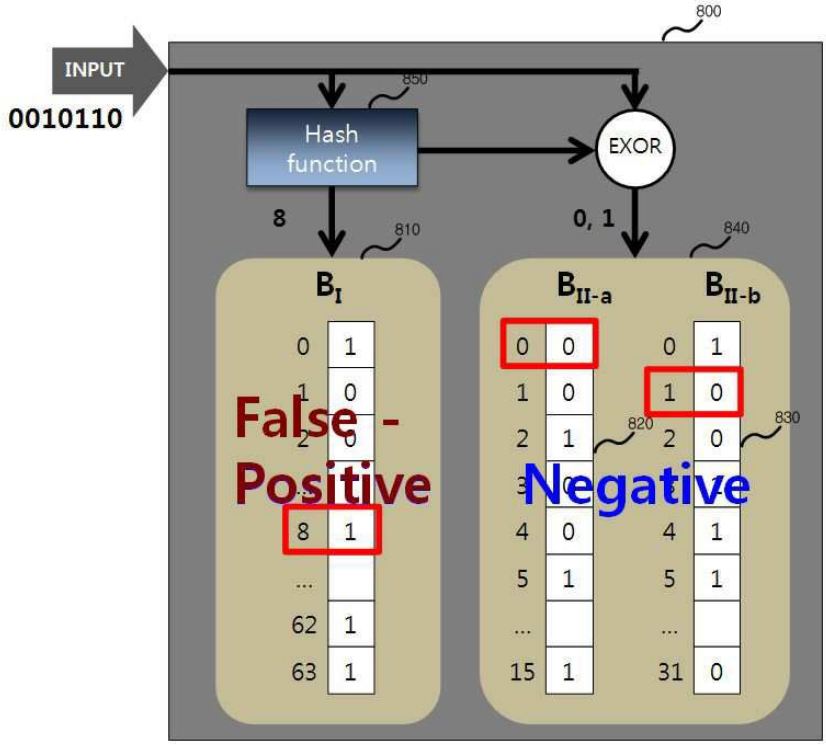
제2 블룸 필터: 0010010110110001

제3 블룸 필터: 10011110011101001010100110010101

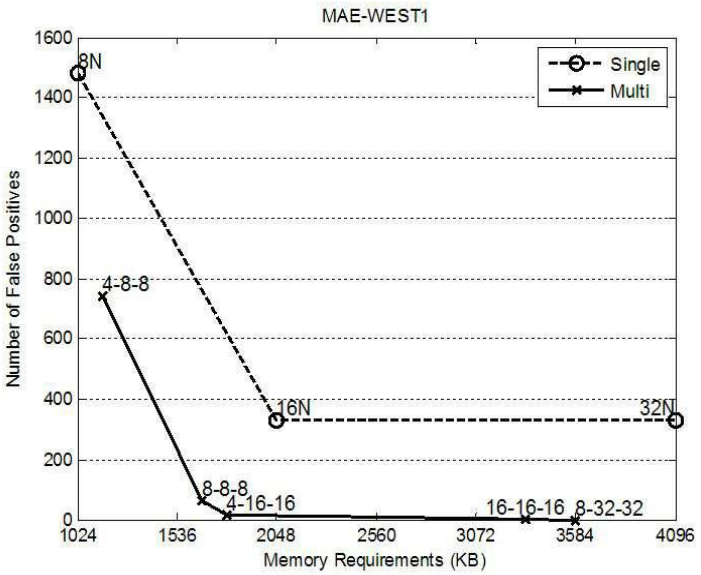
도면13


800
INPUT
1
850
Hash function
EXOR
62
0, 0
810
840
B_I
B_{II-a}
B_{II-b}
820
830
Positive
Positive

도면14


800
INPUT
0010110
850
Hash function
EXOR
8
0, 1
810
840
B_I
B_{II-a}
B_{II-b}
820
830
False - Positive
Negative

도면15a



도면15b

구조	블룸 필터의 크기 (N')	메모리 요구량 (KB)	해쉬 함수 개수 (m)	거짓 양성 개수	거짓 양성 비율
Multi	(8N', 32N', 32N')	3584	50	0	0
Multi	(16N', 16N', 16N')	3328	33	1	1.856E-06
Multi	(4N', 16N', 16N')	1792	25	18	3.340E-05
Multi	(8N', 8N', 8N')	1664	18	62	1.150E-04
Single	32N'	4096	22	328	6.086E-04
Single	16N'	2048	11	331	6.142E-04
Multi	(4N', 8N', 8N')	1152	15	740	1.373E-03
Single	8N'	1024	6	1482	2.750E-03
Multi	(2N', 8N', 8N')	896	13	4233	7.855E-03
Multi	(4N', 4N', 4N')	832	9	5560	1.032E-02
Single	4N'	512	3	24474	4.541E-02
Multi	(1N', 4N', 4N')	448	7	55693	1.033E-01
Multi	(2N', 2N', 2N')	416	3	81470	1.512E-01