



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2012-0085400
(43) 공개일자 2012년08월01일

(51) 국제특허분류(Int. Cl.)
H04L 12/56 (2006.01) H04L 12/26 (2006.01)
G06F 15/16 (2006.01)
(21) 출원번호 10-2011-0006691
(22) 출원일자 2011년01월24일
심사청구일자 2011년01월24일

(71) 출원인
충남대학교산학협력단
대전광역시 유성구 대학로 99 (궁동, 충남대학교)
(72) 발명자
이영석
대전광역시 유성구 배울2로 3, 대덕테크노밸리 8
단지 금성백조아파트 809동 602호 (관평동)
이연희
대전광역시 유성구 노은로 353, 송림마을 3단지
304동 1701호 (하기동)
(74) 대리인
김원준

전체 청구항 수 : 총 6 항

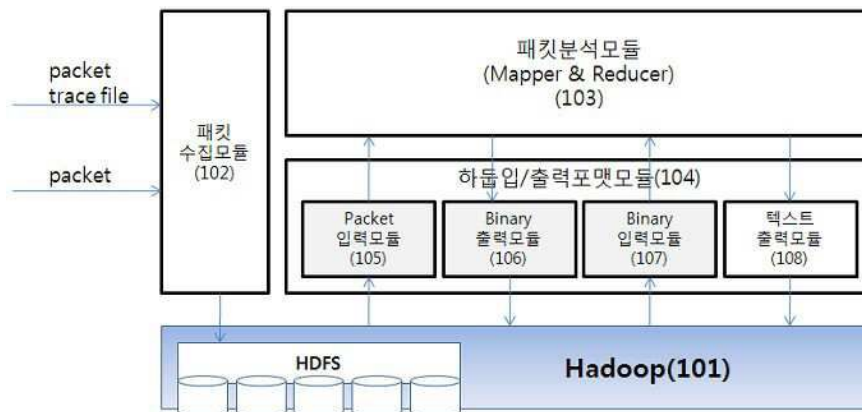
(54) 발명의 명칭 하둡 기반 병렬 연산에 의한 패킷분석 시스템 및 방법

(57) 요약

본 발명은 하둡(Hadoop)이라는 오픈 소스 분산 시스템에서 네트워크 상에서 수집되는 대량의 패킷을 각 클러스터 노드에서 병렬로 처리할 수 있도록 구현한 패킷분석 시스템 및 패킷분석 방법에 관한 것으로, 보다 상세하게는 패킷 trace를 HDFS(Hadoop분산파일시스템)에 분산 저장시키는 패킷수집모듈; 저장된 HDFS 상의 패킷 trace를 하둡의 각 클러스터 노드에서 맵리듀스 방식으로 분산하여 처리하는 패킷분석모듈; 및 HDFS에 저장된 대용량의 패킷 trace를 맵리듀스 방식으로 처리할 수 있도록 패킷분석모듈에 전달하고, 패킷분석모듈에서 맵리듀스에 의해 산출된 분석결과를 HDFS에 출력하는 하둡입/출력포맷모듈;을 포함하여 구성되는 것을 특징으로 하는 패킷분석 시스템 및 이에 의한 패킷분석 방법에 관한 것이다.

본 발명의 패킷분석 시스템 및 방법에 의하면, 하둡 클러스터 환경에서 패킷 데이터를 저장 및 분석함으로써 대량의 패킷 trace를 빠르게 처리할 수 있다.

대표도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호	2010-0263
부처명	정보통신연구진흥원
연구사업명	대학 IT 연구센터 육성지원사업
연구과제명	해킹?바이러스 대응기술 연구
주관기관	충남대학교
연구기간	2003.08.01 ~ 2011.08.31이 발명을 지원한 국가연구개발사업
과제고유번호	2010-0582
부처명	한국산업기술 평가관리원
연구사업명	IT 핵심 기술개발 사업
연구과제명	초정밀 측정 및 분석기술 연구
주관기관	충남대학교
연구기간	2008.03.01 ~ 2013.02.28

특허청구의 범위

청구항 1

하둡프레임워크를 기반으로,

패킷 trace를 HDFS(Hadoop분산파일시스템)에 분산 저장시키는 패킷수집모듈;

저장된 HDFS 상의 패킷 trace를 하둡의 각 클러스터 노드에서 맵리듀스 방식으로 분산하여 처리하는 패킷분석모듈; 및

HDFS에 저장된 대용량의 패킷 trace를 맵리듀스 방식으로 처리할 수 있도록 패킷분석모듈에 전달하고, 패킷분석모듈에서 맵리듀스에 의해 산출된 분석결과를 HDFS에 출력하는 하둡입/출력포맷모듈;

을 포함하여 구성되는 것을 특징으로 하는 패킷분석 시스템.

청구항 2

제 1 항에 있어서,

상기 패킷수집모듈은,

네트워크 상의 패킷으로부터 패킷 trace를 수집하는 패킷수집부; 및

패킷수집부에 의해 수집된 패킷 trace 또는 미리 생성된 패킷 trace 파일을 하둡파일시스템 API를 이용하여 HDFS에 저장할 수 있도록 하는 패킷저장부;

를 포함하여 구성되는 것을 특징으로 하는 패킷분석 시스템.

청구항 3

(A) 패킷 trace를 HDFS에 저장하는 단계;

(B) 하둡의 각 클러스터 노드에서 상기 HDFS에 저장된 패킷 trace를 읽어 레코드를 추출하고 맵리듀스로 전달하는 단계;

(C) 상기 전달된 레코드들을 맵리듀스 방식에 의해 분석하는 단계; 및

(D) 분석한 레코드를 HDFS로 저장하는 단계;

를 포함하여 이루어지는 것을 특징으로 하는 하둡 기반 병렬 연산에 의한 패킷분석 방법.

청구항 4

제 3 항에 있어서,

상기 (A) 단계의 패킷 trace는 패킷 trace 파일의 형태로 생성되어 있는 것을 수집한 것을 특징으로 하는 패킷분석 방법.

청구항 5

제 3 항에 있어서,

상기 (A) 단계의 패킷 trace는 네트워크 상에서 실시간으로 수집한 패킷으로부터 캡처하여 생성한 것을 특징으로 하는 패킷분석 방법.

청구항 6

제 3 항 내지 제 5 항 중 어느 한 항에 있어서,

상기 (C)단계의 맵리듀스 분석과정은,

(a) 패킷 trace로부터 레코드를 추출하여 맵리듀스 처리하는 것에 의해 플로우를 생성하고, 고정길이의 레코드를 갖는 바이너리 데이터의 형태로 HDFS에 저장하는 제1잡(job); 및

(b) 상기 생성된 플로우 데이터로부터 레코드를 추출하여 플로우를 분석하는 제2잡;

으로 이루어지는 것을 특징으로 하는 패킷분석 방법.

명세서

기술분야

[0001] 본 발명은 하둡(Hadoop)이라는 오픈 소스 분산 시스템에서 네트워크 상에서 수집되는 대량의 패킷을 각 클러스터 노드에서 병렬로 처리할 수 있도록 구현한 패킷분석 시스템 및 패킷분석 방법에 관한 것이다.

배경기술

[0002] 네트워크에서 데이터가 전송되는 양의 정도를 나타내는 네트워크 트래픽을 측정·분석하는 일은 컴퓨터 네트워크 분야를 연구하는데 있어서 기본적으로면서도 가장 중요한 분야 중 하나이다. 네트워크 트래픽의 측정은 네트워크의 운영 상태와 트래픽 특성 파악, 설계 및 계획, 유해한 트래픽의 차단, 과금, QoS(Quality of Service) 보장을 위해 필수적이다.

[0003] 네트워크 트래픽의 분석방법에는 패킷 단위의 분석과 플로우 단위의 분석이 존재한다. 초기의 트래픽 분석은 패킷 단위로 이루어졌으나, 인터넷 사용자 수 및 네트워크와 트래픽의 규모가 급속도로 증가하면서 패킷의 모음인 플로우 단위의 분석이 등장하여 널리 사용되고 있다. 플로우 기반 분석 방법에서는 모든 패킷에 대해 패킷 단위로 측정하고 분석하는 대신 공통된 특성(예로 들어, 소스 IP 주소(Source IP Address), 목적지 IP 주소(Destination IP Address), 소스 포트(Source Port), 목적지 포트(Destination Port) 및 프로토콜 ID, DSCP, 등)를 가지는 패킷들을 플로우라는 단위로 묶어서 분석한다. 이러한 플로우 기반 분석 방법은 패킷을 일정 기준에 따라 묶은 플로우를 기반으로 트래픽을 분석함으로써, 트래픽 분석 처리 지연시간을 크게 줄일 수 있으나 플로우에는 패킷에 대한 세부 정보가 결여되어 있기 때문에 패킷분석에 비해 제공되는 데이터의 수가 적다는 단점이 있다.

[0004] 대량으로 수집되는 인터넷 트래픽의 측정과 분석은 대용량의 저장공간과 처리 성능을 요구하며, 특히 패킷단위의 트래픽의 측정과 분석은 플로우 단위의 트래픽의 측정과 분석에 비해 더 많은 저장공간과 처리 능력을 요구한다. 하지만 현재 단일노드에서 실행되는 수집과 분석 툴들은 이러한 요구를 만족하기에는 한계가 있다. 이런 이유로 라우터가 각 네트워크 인터페이스를 통해 지나가는 플로우 정보를 수집하여 제공하는 Cisco NetFlow를 이용한 트래픽 분석이 제안되었다. 플로우 단위 분석방법으로는 IPFIX가 있으며, Flow-tools 등이 대표적인 분석 툴로 사용된다. 그러나 IPFIX와 같은 플로우 단위 분석 툴의 경우 단일 서버에서 작동하므로 패킷 분석보다는 높은 성능을 기대할 수 있다고 하더라도 플로우 분석 서버의 성능이 오버헤드로 작용하여 트래픽 분석 속도가 저하될 수 있다는 문제가 있다. 이러한 문제점은 수 100Mbps에서 수 10Gbps에 달하는 고속의 인터넷 망에서 대용량 트래픽을 처리하는 라우터로부터 대용량 패킷 데이터를 수집하여 분석하는 시스템에서 더욱 심각하게 발생한다. 따라서 트래픽 측정을 위하여 빠른 시간 내에 플로우 데이터를 분석하고 그 결과를 사용자에게 전달하기 위해서는 서버의 성능이 우수하여야 하므로 고성능의 서버를 필요로 하여 비용면에서 부담이 된다.

[0005] 하둡(Hadoop)은 너치(Nutch)의 분산처리를 지원하기 위해 개발된 것으로, 수백 기가바이트~테라바이트 혹은 페타바이트 크기의 데이터를 처리할 수 있는 어플리케이션을 제작하고 운영할 수 있는 기반을 제공해 주는 데이터 처리 플랫폼이다. 하둡이 처리하는 데이터의 크기가 통상 최소 수백 기가바이트 수준이기 때문에 데이터는 하나의 컴퓨터에 저장되는 것이 아니라 여러 개의 블록으로 나누어져 여러 개의 컴퓨터에 분산 저장된다. 따라서 하둡은 입력되는 데이터를 나누어 처리할 수 있도록 하는 하둡 분산 파일 시스템(HDFS: Hadoop Distributed

File System)을 포함하며, 분산 저장된 데이터들은 대용량 데이터를 클러스터 환경에서 병렬 처리하기 위해 개발된 MapReduce 과정에 의해 처리되어진다. 하둡은 대용량의 데이터 처리를 요하는 여러분야에 널리 응용되어 사용되고 있으나, 아직까지 이를 이용한 패킷분석 시스템과 방법에 대해서는 개발된 바 없다.

발명의 내용

해결하려는 과제

[0006] 상기와 같은 종래기술의 문제점을 해소하기 위한 본 발명의 목적은 대량의 패킷 데이터를 처리할 수 있는 프레임워크로서 하둡 분산시스템을 이용하여, 패킷 데이터를 다수의 서버에 분산하여 저장하고 다수의 서버에서 병렬연산에 의해 분석할 수 있는 시스템 및 방법을 제공하는 것이다.

과제의 해결 수단

[0007] 전술한 목적을 달성하기 위한 본 발명은 하둡프레임워크를 기반으로 한 패킷분석 시스템에 관한 것으로, 보다 상세하게는 패킷 trace를 HDFS(Hadoop분산파일시스템)에 분산 저장시키는 패킷수집모듈; 저장된 HDFS 상의 패킷 trace를 하둡의 각 클러스터 노드에서 맵리듀스 방식으로 분산하여 처리하는 패킷분석모듈; 및 HDFS에 저장된 대용량의 패킷 trace를 맵리듀스 방식으로 처리할 수 있도록 패킷분석모듈에 전달하고, 패킷분석모듈에서 맵리듀스에 의해 산출된 분석결과를 HDFS에 출력하는 하둡입/출력포맷모듈;을 포함하여 구성되는 것을 특징으로 한다.

[0008] 본 발명은 또한 (A) 패킷 trace를 HDFS에 저장하는 단계; (B) 하둡의 각 클러스터 노드에서 상기 HDFS에 저장된 패킷 trace를 읽어 레코드를 추출하고 맵리듀스로 전달하는 단계; (C) 상기 전달된 레코드들을 맵리듀스 방식에 의해 분석하는 단계; 및 (D) 분석한 레코드를 HDFS로 저장하는 단계;를 포함하여 이루어지는 것을 특징으로 하는 하둡 기반 병렬 연산에 의한 패킷분석 방법에 관한 것이다.

발명의 효과

[0009] 이상과 같이 본 발명의 패킷분석 시스템 및 방법에 의하면, 하둡 클러스터 환경에서 패킷 데이터를 저장 및 분석함으로써 대량의 패킷 trace를 빠르게 처리할 수 있다.

[0010] 본 발명의 바이너리 형태의 데이터 분석 방법은 하둡 시스템을 이용한 패킷의 패턴매칭과 같은 응용을 통한 침입탐지시스템의 구성이나, 이미지 데이터, 유전자 정보, 암호처리 등과 같은 바이너리 데이터를 다루는 분야의 분석에 활용될 수 있다. 또한 다수의 서버에서 패킷분석을 병렬연산에 의해 수행하기 때문에 성능이 우수한 고가의 서버를 요하지 않아 비용 절감효과를 가져올 수 있다.

도면의 간단한 설명

[0011] 도 1은 본 발명의 패킷분석 시스템과 그 내부 구성을 보여주는 블록도.
 도 2는 패킷수집모듈의 내부 구성을 보여주는 블록도.
 도 3은 본 발명의 일실시예에 의한 패킷분석 과정을 보여주는 도면.
 도 4는 본 발명의 다른 일실시예에 의한 패킷분석 알고리즘을 보여주는 도면.
 도 5는 도 3의 패킷으로부터 생성된 플로우 통계를 구하는 알고리즘을 보여주는 도면.
 도 6은 본 발명의 다른 일실시예에 의한 패킷분석 알고리즘을 보여주는 도면.

발명을 실시하기 위한 구체적인 내용

[0012] 이하 첨부된 도면을 참조하여 본 발명을 보다 상세히 설명한다. 그러나 이러한 도면은 본 발명의 기술적 사상

의 내용과 범위를 쉽게 설명하기 위한 예시일 뿐, 이에 의해 본 발명의 기술적 범위가 한정되거나 변경되는 것은 아니다. 또한 이러한 예시에 기초하여 본 발명의 기술적 사상의 범위 안에서 다양한 변형과 변경이 가능함은 당업자에게는 당연할 것이다.

- [0013] 본 발명은 하둡이라는 오픈 소스 분산 시스템에서 대량의 패킷을 각 클러스터 노드에서 병렬 처리할 수 있도록 구현한 시스템으로서, 도 1은 일 실시예에 의한 본 발명의 패킷분석 시스템과 그 내부 구성을 보여주는 블록도이다. 도 1을 참조하면, 본 발명의 패킷분석시스템은 하둡프레임워크(101)를 기반으로 하며, 패킷 trace를 HDFS(Hadoop분산파일시스템)에 분산 저장시키는 패킷수집모듈(102), 저장된 HDFS 상의 대용량의 패킷 trace를 하둡의 각 클러스터 노드에서 맵리듀스 방식으로 분산하여 처리하는 패킷분석모듈(103), HDFS에 저장된 대용량의 패킷 trace를 맵리듀스 방식으로 처리할 수 있도록 패킷분석모듈(103)에 전달하고, 패킷분석모듈(103)에서 맵리듀스에 의해 산출된 분석결과를 HDFS에 출력하는 하둡입/출력포맷모듈(104)로 구성된다. 상기 패킷 trace는 이미 패킷 trace 파일의 형태로 생성되어 있는 것일 수도 있으며, 네트워크 상에서 실시간으로 수집한 패킷으로부터 캡처하여 생성한 것일 수도 있다.
- [0014] 도 1에서는 하둡입/출력포맷모듈의 세부적인 구성으로 pcap입력포맷모듈, 바이너리출력포맷모듈, 바이너리입력포맷모듈, 텍스트출력포맷모듈을 블록도로 도시하였으나, 이는 입/출력포맷모듈의 일 예의 구성을 도시한 것으로 본 발명에서 입/출력포맷모듈이 이에 한정되는 것은 아니며 기존에 알려진 하둡입/출력포맷 또는 이후 하둡 맵리듀스를 위해 고안되는 입/출력포맷 중 분석하고자하는 목적에 따라 적절히 선택하여 사용할 수 있음은 당연하다.
- [0015] 일 예로, 상기 텍스트출력포맷은 기존에 널리 이용되고 있는 출력포맷이나, 바이너리 입/출력포맷은 고정길이의 레코드를 갖는 바이너리 데이터를 보다 효율적으로 하둡맵리듀스로 분석하기 위하여 본 발명자들이 고안한 입/출력포맷이며 이에 대해서는 특허출원 10-2011-5424로 출원한 바있다. pcap입력포맷은 가변길이의 레코드를 갖는 바이너리 패킷데이터의 하둡맵리듀스를 위해 고안한 입력포맷으로, 이에 대해서는 특허출원 10-2011-6180호로 출원하였다. 상기 바이너리입/출력포맷이나 pcap입력포맷에 의하면 바이너리 데이터를 별도의 데이터 포맷의 변환작업 없이 하둡맵리듀스로 처리가 가능하므로 보다 효율적으로 패킷데이터를 처리할 수 있다. 그러나, 시퀀스입/출력포맷이나 텍스트입/출력포맷과 같이 종래에 알려진 입/출력포맷만으로도 본 발명의 시스템을 구현하는 것이 불가능한 것은 아니다.
- [0016] 도 2는 본 발명에 의한 분산 병렬 패킷 분석시스템 중 패킷수집모듈의 내부를 구성하는 블록도이다. 상기 패킷수집모듈은 네트워크 상의 패킷으로부터 패킷 trace를 수집하는 패킷수집부, 패킷수집부에 의해 수집된 패킷 trace 또는 미리 생성된 패킷 trace 파일을 하둡파일시스템 API를 이용하여 HDFS에 저장할 수 있도록 하는 패킷저장부로 구성된다. 패킷수집모듈의 구체적인 구성을 예로 들면, 먼저 네트워크 상의 패킷을 수집하기 위해 Libpcap(201)을 사용하여 패킷을 수집한다. 수집된 패킷을 자바 기반의 하둡 시스템과의 연동을 위해 자바 기반의 캡처틀인 Jpcap(202)을 이용하여 하둡으로 전달하고, 이렇게 전달된 패킷 trace는 하둡 파일 시스템 API(203)를 사용하여 HDFS에 저장하도록 한다.
- [0017] 상기 패킷수집모듈에 의해 네트워크 상에서 이동하는 패킷을 실시간으로 수집하여 패킷 trace를 HDFS에 저장한다. 또한, 이전에 패킷 trace 파일 형태로 저장된 파일은 바로 하둡파일시스템 API에 의해 HDFS에 저장한다.
- [0018] 본 발명은 또한 상기 시스템에 의해 패킷을 분석하는 방법에 관한 것이다. 보다 구체적으로 본 발명의 패킷분석 방법은 (A) 패킷 trace를 HDFS에 저장하는 단계; (B) 하둡의 각 클러스터 노드에서 상기 HDFS에 저장된 패킷 trace를 읽어 레코드를 추출하고 맵리듀스로 전달하는 단계; (C) 상기 전달된 레코드들을 맵리듀스 방식에 의해 분석하는 단계; 및 (D) 분석한 레코드를 HDFS로 저장하는 단계로 구성된다.
- [0019] 이때 상기 (A) 단계의 패킷 trace는 이미 패킷 trace 파일의 형태로 생성되어 있는 것일 수도 있으며, 네트워크 상에서 실시간으로 수집한 패킷으로부터 캡처하여 생성한 것일 수도 있다.
- [0020] (B) 단계에서 HDFS에 저장된 패킷 trace를 읽는 것은 하둡의 입력포맷을 통하여 수행된다. 상기 입력포맷은 기존의 하둡시스템에서 제공되는 다양한 입력포맷 중 하나를 선택하여 사용할 수도 있으며 별도의 패킷 입력포맷을 구현하여 사용할 수도 있다. 입력포맷은 HDFS에 저장된 데이터 블록으로부터 레코드를 읽는 방식을 규정하

는 것으로, 적절한 입력포맷을 사용하면 더욱 효과적으로 패킷분석을 수행할 수 있다.

[0021] 즉, 가변길이의 레코드를 갖는 바이너리 패킷 데이터의 분석에는 본 발명자들이 특허출원 10-2011-6180호로 출원한 pcap입력포맷을 사용할 수 있다. pcap입력포맷은 (a) 패킷의 캡처를 수행한 시작시간과 종료시간에 대한 정보를 configuration property로부터 획득하는 단계; (b) 하둡분산파일시스템(HDFS)에 저장된 데이터 블록 중 처리해야 될 데이터 블록에서 첫 패킷의 시작점을 검색하는 단계; (c) 상기 첫 패킷의 시작점을 InputSplit의 시작점으로 하여 이전 InputSplit과 자신의 InputSplit의 경계를 설정하는 것에 의해 InputSplit을 정의하는 단계; (d) 상기에서 정의된 자신의 InputSplit 전체 영역에 대해 시작점으로부터 각 패킷의 캡처된 패킷헤더(pcap header)에 기록된 캡처된 패킷길이(capLen)만큼씩 읽는 일을 수행하는 RecordReader를 생성하고 이를 반환하는 단계; 및 (e) 상기 RecordReader를 통해 (Key, Value)를 (LongWritable, BytesWritable)의 형태로 레코드들을 추출하는 단계;를 포함하여 이루어진다.

[0022] 한편, 고정길이의 레코드를 갖는 바이너리 데이터에 대해서는 본 발명자들이 특허출원 10-2011-5424호로 출원한 바이너리입력포맷을 사용할 수 있다. 바이너리입력포맷은 (ㄱ) 바이너리 데이터의 레코드의 길이를 입력받는 단계; (ㄴ) 하둡분산파일시스템(HDFS)에 저장된 데이터 블록 중 처리해야 될 데이터 블록에서 레코드의 길이의 n배수가 되는 지점 중 블록 시작점에 가장 가까운 값을 시작점으로 이전 InputSplit과 자신의 InputSplit의 경계를 설정하는 것에 의해 InputSplit을 정의하는 단계; (ㄷ) 상기에서 정의된 자신의 InputSplit 전체 영역에 대해 시작점으로부터 레코드의 길이만큼씩 읽는 임무를 수행하는 RecordReader를 생성하고 이를 반환하는 단계; 및 (ㄹ) 상기 RecordReader를 통해 (Key, Value)를 (LongWritable, BytesWritable)의 형태로 레코드들을 추출하는 단계;를 포함하여 이루어진다.

[0023] 상기 pcap입력포맷이나 바이너리입력포맷을 사용하면 각 분산노드에 저장된 바이너리 형태의 데이터 블록에 대해 InputSplit를 정의하여 줌으로써 동시에 접근하여 처리가 가능하도록 하며, InputSplit에서 바이너리 패킷 데이터를 추출하여 맵퍼에 전달함으로써 데이터 포맷의 변환작업 없이 처리가 가능하므로, 다른 형태의 데이터에 비해 적은 저장 공간을 요하며 빠른 처리 속도를 가능하게 한다.

[0024] 상기 (C)의 데이터 분석은 분석하고자 하는 특성에 따라 적절한 (Key, Value)를 고안하여 맵리듀스를 수행하는 것에 의해 결과를 얻을 수 있다. 예를 들어, 1) 패킷으로부터 플로우를 생성하여 통계를 얻고자 하는 경우, 5 튜플(출발지 아이피, 도착지 아이피, 출발지 포트번호, 도착지 아이피 번호, 프로토콜)과 플로우 존속기간을 기준으로 패킷의 타임스탬프를 구역으로 분류한 정보를 기준으로 시간대 별 플로우의 바이트와 패킷수, 플로우의 수의 통계를 구하거나, 2) 패킷에 대한 아이피 버전과 프로토콜 별 총 바이트와 패킷, 플로우의 수의 통계를 구하고, 유일한 프로토콜 버전 별 유일한 아이피나 포트 수와 같은 통계를 구하거나, 3) 각 포트와 아이피 별 트래픽 볼륨을 얻고자 하는 경우, 각 포트나 아이피, 프로토콜을 기준으로 한 바이트 수, 패킷 수, 플로우의 수를 구하고, 시간 대별 패킷의 바이트 수, 패킷 수, 플로우 수를 구하는 단계를 진행한다.

[0025] 이를 위해 이를 위해 상기 (C)단계의 맵리듀스 분석과정은 HDFS로부터 레코드를 추출하여 맵리듀스에 의해 분석을 수행하는 잡(job)이 하나 이상 수행될 수 있다. 예를 들어, 바이너리 패킷 데이터를 읽어 5튜플로 분류하여 처리된 중간 결과로서 플로우를 생성한 후 고정 길이의 레코드를 갖는 바이너리 데이터의 형태로 다시 HDFS에 저장하고, 상기 바이너리 플로우 데이터를 읽어 플로우를 분석하는 잡을 수행할 수 있다. 앞에서 설명한 분석 항목에 대한 설명은 단지 예시일 뿐 분석하는 대상에 따라 다양한 방법의 접근이 가능하다.

[0026] 도 3~도 6은 본 발명의 일실시예에 의한 패킷분석 과정을 보여준다.

[0027] 도 3은 MapReduce를 이용한 패킷분석의 과정의 예로서, 패킷으로부터 플로우를 추출하여 시간대별 총 바이트, 총 패킷 수, 총 플로우의 수를 구하는 과정을 본 발명의 시스템 모듈과 연계하여 보여준다. 본 패킷분석 과정은 총 2개의 맵리듀스 잡으로 구성된다. 첫 번째 잡은 개별적인 패킷레코드들로부터 5튜플과 패킷의 캡처시간을 일정 시간대역으로 마스킹한 값을 키(key)로하고 패킷의 내용을 값(value)로 추출하는 Map과 이러한 키에 대한 바이트 수와 패킷의 수를 합하는 Reduce 함수로 구성하여 패킷으로부터 플로우 생성하는 일을 수행한다.

[0028] 두 번째 잡은 생성한 플로우 레코드를 읽어 키값에서 마스킹된 캡처시간을 떼어낸 5튜플을 키로 하고 플로우 수를 구하기 위해 현재 플로우의 수를 나타내는 '1'을 바이트 수와 패킷 수와 함께 값(value)으로 구성하는 Map과 이 값을 가져와 5튜플 별 총 바이트 수, 패킷 수, 플로우의 수를 합산하는 Reduce로 구성하여 최종 플로우 별

통계를 출력한다.

[0029] 상기 패킷을 이용한 플로우 별 통계는 병렬 패킷 처리의 하나의 예시이고, 이러한 잡은 분석하고자 하는 대상에 따라 필요한 맵과 리듀스 함수를 구현하여 잡을 수행할 수 있다. 또한 하나 이상의 잡으로 구성하여 이전 잡의 결과를 다음 잡의 입력으로 연결하여 좀 더 복잡하고 정제된 분석 결과를 얻을 수 있다.

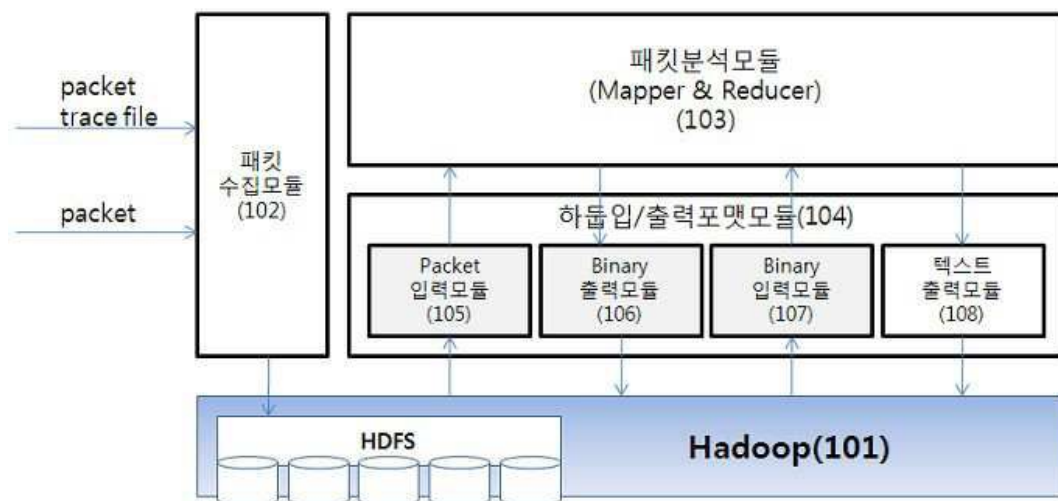
[0030] 도 4는 전체 트래픽에 대해 IP 버전 별 바이트 수와, 패킷의 수, 그리고 IP 버전 별 유일한 source와 destination IP주소의 수, port 번호의 수, 그리고 IPv4에 대한 플로우의 수를 구하기 위해 두 개의 맵리듀스 잡으로 구성하여 구현한 알고리즘을 도시한다. 첫 번째 잡에서는 Non-IP와 IPv4, IPv6를 구분하여 바이트 수와 패킷 수를 구하는 일과 유일한 IPv4의 source/destination 별 IP주소, 프로토콜 별 포트번호를 구하기 위해 각 레코드의 키와 유일 값(1)을 생성한다. 또한 IPv4에 대한 플로우의 수를 구하기 위해 패킷레코드들로부터 5튜플과 패킷의 캡처시간을 일정 시간대역으로 마스킹한 값을 키(key)를 생성한다. 두 번째 잡에서는 특정 레코드 값을 가리키는 키를 유일 값의 통계를 구하고자 하는 항목에 대한 그룹 키를 생성하여 리듀스로 전달함으로써 같은 그룹에 대한 합계를 구한다.

[0031] 도 5는 도3에서 예시한 플로우 통계를 구하는 알고리즘을 도시한다. Job에 대한 설명은 도 3에서 설명한 바와 같다.

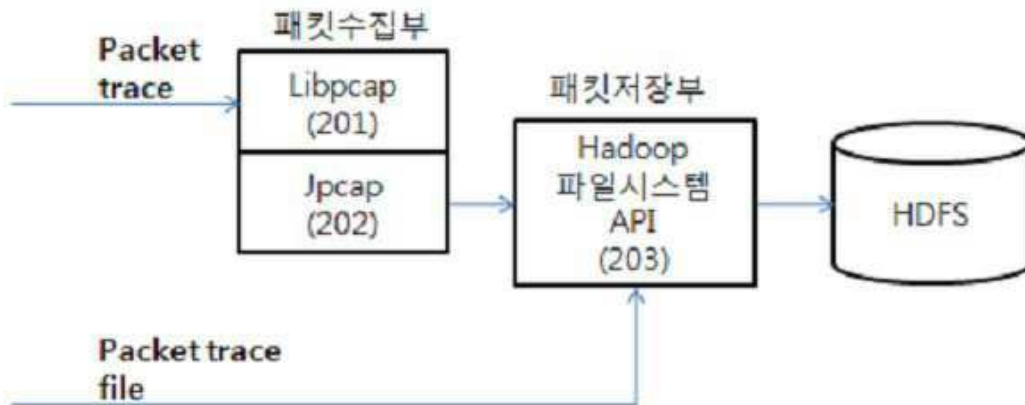
[0032] 도 6은 이전 잡에서 얻은 결과를 정렬하여 최고 값이나 최저 값을 가진 N개의 레코드만 출력하는 알고리즘을 도시한다. 맵단계에서는 이전 잡의 결과를 받아 정렬하고자 하는 기준을 Key로 생성하고 리듀스 단계에서는 이 Key로 정렬된 결과를 N개만 추출하여 출력한다.

도면

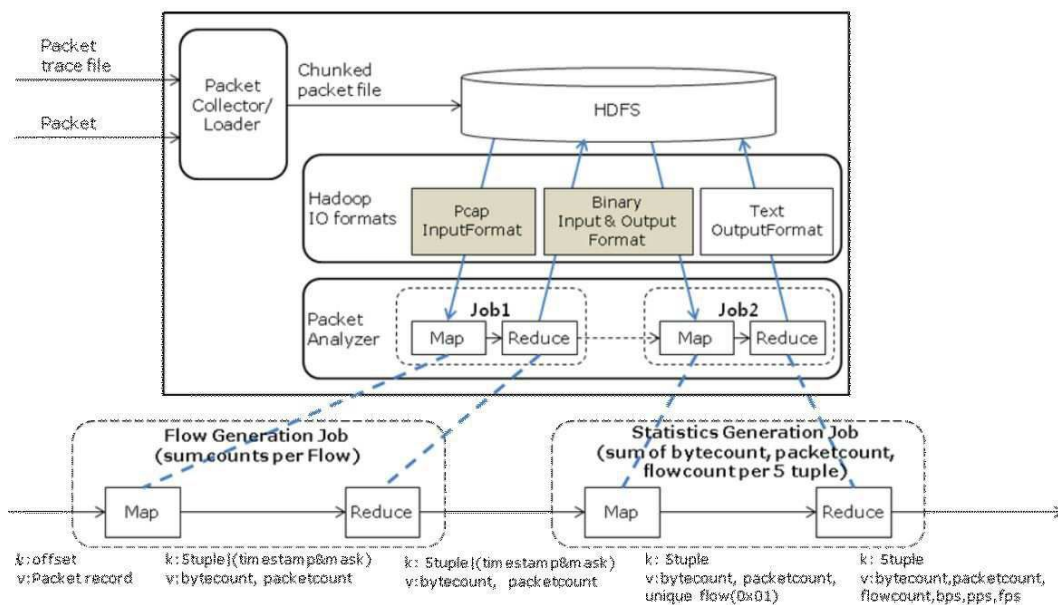
도면1



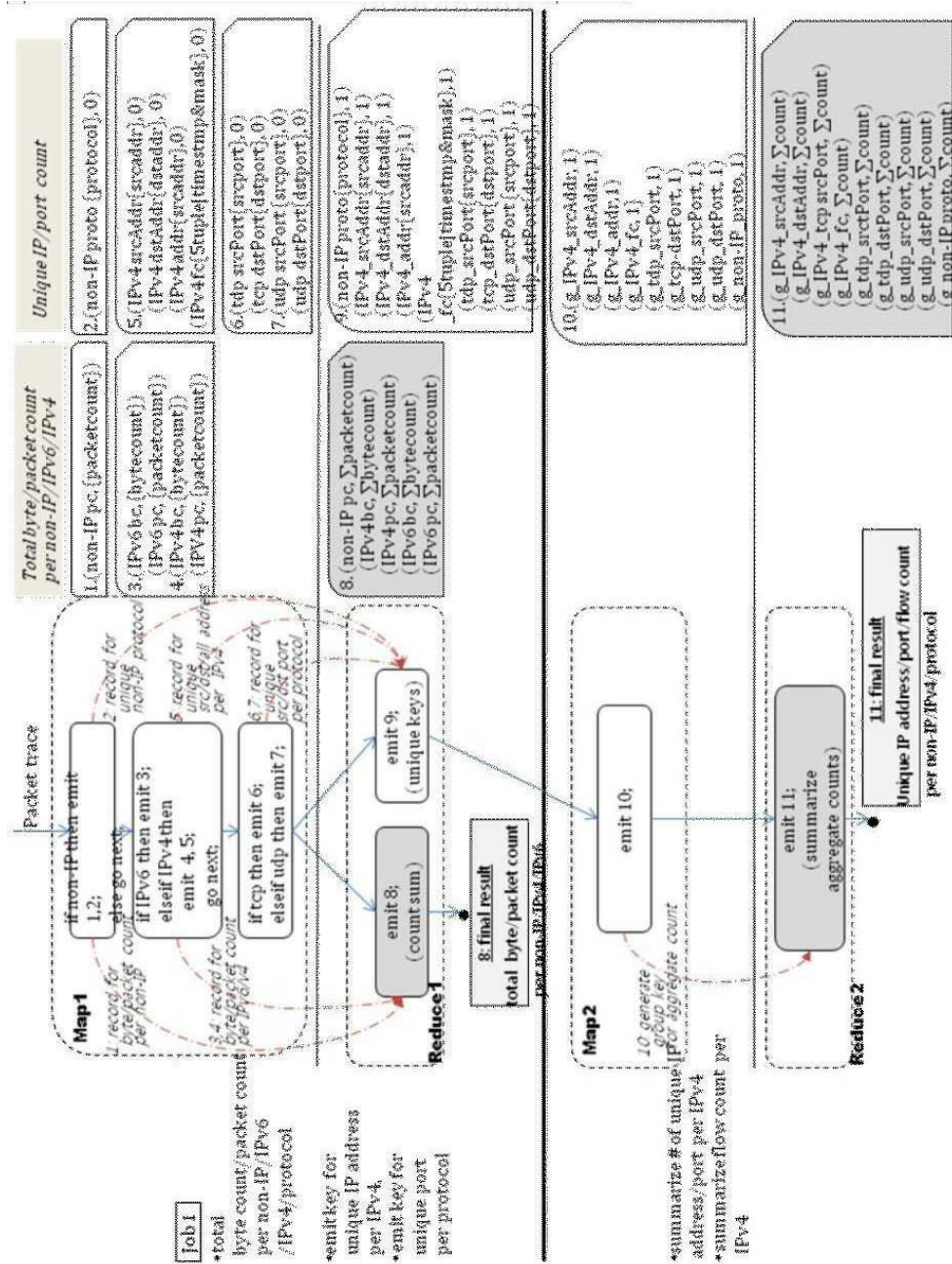
도면2



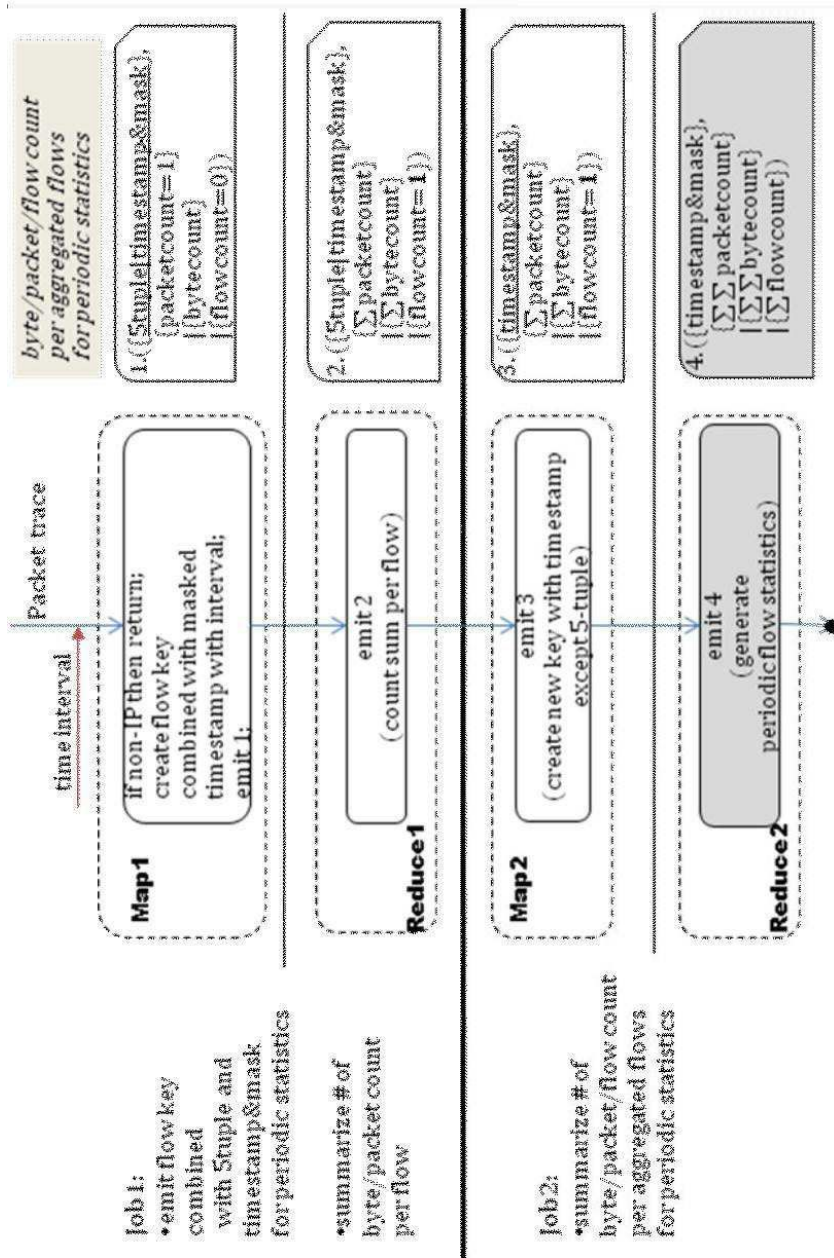
도면3



도면4



도면5



도면6

