하둡 분산 컴퓨팅 환경에서 페이로드 시그니쳐 기반 응용 트래픽 분류

심규석, 이수강, 김성민, 김명섭

고려대학교 컴퓨터정보학과

{kusuk007, sukanglee, gogumiking, tmskim}@korea.ac.kr

요 약

네트워크 발전이 급속도로 성장함에 따라 네트워크 트래픽 사용량 또한 폭발적으로 증가하면서 빅데이터 시대에 가까워 지고 있다. 그럼에도 불구하고 네트워크 트래픽 관리방법은 트래픽 사용량의 증가추세에 맞게 발전되지 못하고 있는 것이 현실이다. 향후 네트워크 트래픽 사용량의 증가추세를 고려했을 때 많은 양의 트래픽 데이터에 적합한 네트워크 관리방법은 반드시 고려되어야 한다. 따라서 본 논문에서는 빅데이터를 분석하는 플랫폼 중 하나인 Hadoop 분산 처리 시스템에서의 페이로드 시그니처 기반 응용 트래픽 분류 시스템을 제안한다. 또한, 본 논문에서 제안한 Hadoop 기반 응용 트래픽 분류 시스템과 기존의 응용 트래픽 분류 시스템의 성능을 비교함으로써 가능성 및 타당성을 증명한다.

1. 서론

네트워크는 사용자에게 더욱 빠른 속도와 다양한 서비스를 제공하기 위해 급속도로 발전하고 있다. 그에 따른 네트워크의 트래픽 사용량도 폭발적으로 증가하고 있다. 그럼에도 불구하고 네트워크 트래픽 관리방법은 트래픽 사용량의 증가추세에 맞게 발전되지 못하고 있는 것이 현실이다.

일반적인 페이로드 시그니쳐를 이용한 응용 분류 방법은 대부분의 경우 하나의 머신에서 동작한다. 그러나 하나의 머신에서 많은 양의 트래픽 데이터를 처리하면 실행속도가 늘어나거나 저장공간이 부족하게 되는 문제점이 있다. 하지만 Hadoop 시스템은 트래픽 데이터를 여러 대의 머신에 분산 저장소와 분산 연산 기능을 지원 하기 때문에 기존의 방법보다 처리속도와 메모리 비율, 그리고 저장공간에 많은 이점을 보인다.

따라서 네트워크 트래픽 관리 시스템에 분산처리 플랫폼인 Hadoop 을 사용하는 방법에 대하여 제안한다. Hadoop 은 병렬도가 굉장히 높은 단순 작업에 사용하기 적합하기 때문에 페이로드와 시그니쳐를 비교하여 응용을 분류하는 작업을 고안했다. 페이로드 시그니쳐를 이용한 응용 분류 방법은 Packet 에 포함되어 있는 페이로드, Port 번호, IP, Protocol 번호를 이미 정의한 시그니쳐와 비교하여 응용을 분류하는 방법이다. 페이로드는 Packet 이 전송하고자 하는 데이터의 한 블록이고, 문자열로 구성된다. 시그니쳐는 특정 응용을 파악할 수 있는 정보이다. 트래픽의 관리에서 시그니쳐 기반 응용분석을 Hadoop 기반에서 구현하여 처리속도와 확장성, 가용성을 높이는데 치중하였다.

본 논문의 구성은 다음과 같다. 2 장에서는 기존 관련연구에서 Hadoop 에 대한 간단한 설명과 기존 연구와의 차이점에 대해 언급하고 본 연구에서 Hadoop 이 하는 역할에 대해서 제시한다. 또한, 시그니쳐 기반 응용분석을 Hadoop 분산 컴퓨팅 환경과 기존 환경에 대해 비교 실험을 통해 Hadoop 분산 컴퓨팅 환경의 이점에 대해 나타내고, 현재 구축되어 있는 Hadoop 실험환경에 대해 기술한다. 본 실험에 사용된 MapReduce 의 Key, Value 값을 사용하여 응용분석을 사용하는 시스템 구조에 대해 제안하고, 응용 분석 실험 결과를 통하여 본 시스템에 대한 장점과 마지막으로 향후 연구에 대해 언급한다.

2. 관련 연구

Hadoop 은 분산 연산 기능과 분산 저장소를 모두 지원하는 플랫폼이다. 분산 연산기능은 MapReduce 로 지원하고, 분산 저장소는 HDFS (Hadoop Distributed File System)로 지원한다. Hadoop 은 2005 년 더그 커팅과 마이크 캐퍼렐라가 개발하였다. Hadoop 은 구글의 분산 파일 시스템 논문이 공개된 후 그 구조에 MapReduce 를 대응하는 체계로 개발되었다. 현재 Hadoop 은 아파치 재단으로 넘어가 공개 소프트웨어로 개발되고 있다.[5]

Hadoop 에서 가장 핵심적인 개념인 MapReduce 는 분산 연산 기능을 수행한다. MapReduce 의 동작과정은 다음 그림 1 과 같다. Input 파일의 크기에 따라 Map 으로 분할되어 입력된다. Default 로 64MB 단위로 분할되게 되어 있지만, 사용자 임의로 설정할 수 있다. 그림 1 에서 볼 수 있듯이 Map 으로 입력 되면 데이터가 3 개로 복제되어서 각각의 노드로 입력이 된다. 똑같은 Input 파일이 3 개로 복제 되는 것은,

이 논문은 2012 년 정부(교육과학기술부)의 재원으로 한국연구재단(2012R1A1A2007483) 및 2013 년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대정보.컴퓨팅기술개발사업(2010-0020728)의 지원을 받아 수행된 연구임

만약 하나의 머신 즉, Slave 노드가 불량이 되더라도 다른 Slave 노드에서 처리할 수 있도록 데이터 손실을 막기 위함이다. 여러 개의 Map 이 분산 연산기능을 하고, 분산된 데이터를 다시 합치는 것이 Reduce 가 하는 일이다. Reduce 는 Map 에서 분리되어 분산적으로 처리했던 데이터을 다시 합쳐서 사용자에게 원하는 처리 결과를 보여주는 역할을 한다. 따라서 Hadoop 의 분산처리 기술은 MapReduce 의 역할이 핵심적이다.

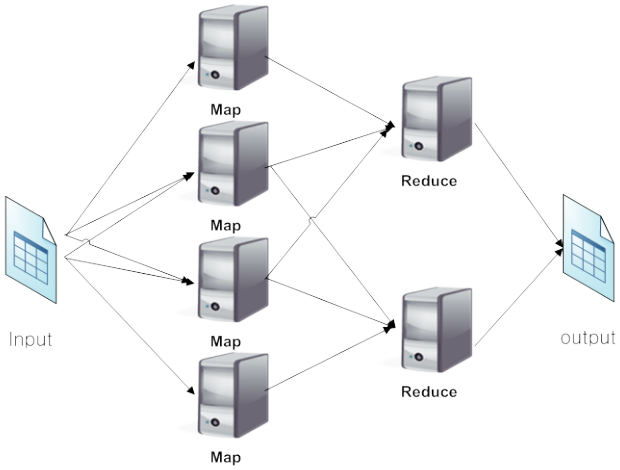

그림 1. MapReduce 의 동작 과정

기존 Hadoop 분산 컴퓨팅 환경에서 네트워크 트래픽 분석 관련 연구는 많은 발전을 했지만 응용 트래픽 분류 관련 연구[1,4] 는 많이 진행되지 않았다. 응용 트래픽 분류는 네트워크 관리자 입장에서 중요한 분야 중 하나이다. 특히, 시그니처 기반 응용 트래픽 분류 연구는 높은 정확도를 보이지만처리속도의 단점을 가지고 있다. 처리속도를 높이기 위한 여러 연구가 진행되고 있지만, 분산 처리가 아닌 알고리즘 향상 부분의 연구가 대부분이다. [3,6]

Hadoop 분산 처리 환경에서 트래픽을 분석하는 도구에 대한 연구는 이미 진행되고 있다.[7] 그러나 이미 진행되고 있는 연구는 트래픽의 양에 대한 분석 또는 트래픽 형태에 대한 분석이다. 본 논문에서는 Hadoop 분산 처리 환경에서 응용 레벨 트래픽 분류 방법을 제안한다.

본 연구에서는 Hadoop 분산 컴퓨팅 환경에서 페이로드 시그니처 기반 응용 트래픽 분류를 제안한다. 시그니처 기반 응용 트래픽 분류는 처리속도의 단점이 있지만 분산 처리로 인해 처리속도를 감소할 수 있다. 또한 많은 양의 데이터를 분석할 때 기존의 시스템 보다 처리속도, 메모리, 저장공간 등 많은 이점이 있다.

3. Hadoop 기반 트래픽 분석

본 장에서는 Hadoop 기반의 응용 분석 시스템 구조를 제안한다. 응용 분석 시스템이란 트래픽을 수집하고, 트래픽의 정보를 확인하여 어떤 응용인지를 분류하는 시스템이다. 트래픽의 Flow 정보들이 입력되고, 특정 응용에서 발생한 트래픽의 Flow, Packet, Byte 양을 출력한다. 또한, 응용 분류 시스템의 분석율을 파악하기 위해 전체 Flow, Packet, Byte 양도 확인한다. 먼저 트래픽 데이터를 수집하는 과정을 언급하고, Hadoop 기반으로 입력되는 Input 파일에 대한 설명을 한다. 또한, Hadoop 분산 처리 환경에서의 응용 트래픽 분류 과정을 제시한다. 트래픽이 수집되는 과정부터 출력되기까지의 전체적인 과정은 본 그림 2 와 같다.

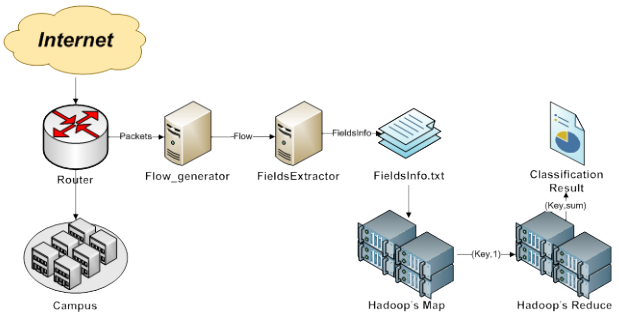

그림 2. Hadoop 기반 응용 트래픽 분석 시스템

3.1 트래픽 수집 및 실험 환경 구현

먼저, 학내에서 발생하는 트래픽을 Packet 단위로 수집한다. 수집된 Packet 들은 Flow 생성기를 통해 Flow 형태의 파일로 변환된다. 여기서 Flow 는 5-tuple (source IP, source port, protocol number, destination IP, destination port)이 같은 packet 의 집합으로 정의한다. Flow 파일에는 5-tuple, Packet 수, Byte 양 그리고 각 Packet 의 페이로드 정보를 포함하여 구성된다.

Flow 파일을 이용하여 Fields Extractor 에서 세 가지 조건에 의해 FieldsInfo.txt 를 출력한다.

· HTTP 트래픽에 한해 분석한다. HTTP 트래픽이 아닌 암호화된 트래픽에서는 페이로드를 분석하는데 한계가 있기 때문이다.

· 분석된 Flow 는 출력하지 않는다. Flow 는 1 분 단위로 생성한다. 그러나 예를 들어 특정 응용에서 트래픽이 3 분동안 발생했을 때 처음 발생한 1 분에서 Flow 가 생성되고, 이 후 Flow 가 같은 응용에 대해서 생성되면 해당 Flow 는 그 전에 생성되었던 Flow 와 같은 의미를 가지게 된다. 따라서 처음 생성된 Flow 에 한해서 분석한다.

· Flow 의 첫 번째 Request Packet 페이로드만 출력한다. Flow 의 첫 번째 Request Packet 에서 해당 응용의 정보를 파악할 수 있기 때문에 나머지 Packet 의 페이로드는 생략한다.

3.2 입력 파일 구성

위의 세 가지 조건에 맞춰서 Flow 파일을 Fields Extractor 을 통하여 FieldsInfo.txt 로 출력한다. FieldsInfo.txt 파일은 하나의 Flow 에서 “sourceIP, sourcePort, protocol number, destinationIP, destinationPort, First Request Packet Payload, pkts, bytes” 로 구성된다. 먼저 Flow 정보인 5-tuple 과 첫 번째 Packet 의 페이

로드를 명시한다. 그리고 해당 Flow 의 Packet 수와 Byte 양은 분별된 Flow 의 Packet 수와 Byte 양을 총 합하기 위해 출력한다.

3.3 MapReduce 를 이용한 응용 분류 방법

FieldsInfo.txt 파일을 이용하여 Hadoop 시스템의 Map 과 Reduce 에서 응용 트래픽 분류하는 과정은 다음과 같다. 먼저 하나의 Flow 정보가 Map 으로 입력된다. 하나의 Flow 에는 8 개의 정보가 있는데 8 개의 정보 중 페이로드를 찾고, 각 응용의 시그니쳐와 해당 Flow 의 페이로드를 비교한다. 만약 기존의 시그니쳐와 해당 페이로드가 일치하면, 해당 응용의 이름을 반환하여 Map 에서 응용의 이름을 출력한다. 예를 들면 Naver 의 Flow 에 대해서 Map 은 Key 값으로 “s-portal-naver”, value 값으로 “1”을 Reduce 로 전송하게 된다. 모든 Flow 정보를 분석함으로써 Map 이 종료되면, Reduce 가 실행된다. Reduce 에서는 Map 에서 생성된 Key 값에 대한 value 값을 모두 더하게 되어 특정 응용에 대한 Flow 개수를 알 수 있다. 그러나 페이로드를 분석하고 응용의 이름을 출력하여 Flow 개수를 파악하는 것과 해당 응용의 Packet, Byte 양를 Map 에서 출력하는 과정은 차이가 있다. 해당 응용의 Packet, Byte 양을 출력하는 과정은 그림 4 와 같다.

```
      Send to Reduce the byte and packet information from Map.
      f = a flow
      f.pay = Payload data of the flow f
      f.pkt = the number of packets in flow f
      f.byte = the number of bytes in flow f
      s, S = a signature(s) in the signature set (S)
      s.name = application name of the signature s
-------------------------------------------------------------------------------
 1:   The count the number of Packets, Bytes
 2:   Input : a flow f
 3:          (srcIP, srcPort, Prot, dstIP, dstPort, pay, pkt, byte )
 4:   Map {
 5:     Find s such that f.pay ⊃ s
 6:
 7:     if ( s exists ) Name = s.name
 8:     else            Name = “unknown”
 9:
10:     Key = Name + “_packet”
11:     Send to Reduce ( Key, f.pkt )
12:     Key = Name + “_byte”
13:     Send to Reduce (Key, f.byte )
14:  }
```

그림 4. 응용 탐지 분산 처리 Map 알고리즘

해당 응용의 Packet, Byte 출력 알고리즘은 먼저 하나의 Flow 정보인 8 개의 정보(srcIP, srcPort, protocol number, dstIP, dstPort, Payload, the number of packets, the number of bytes)를 Map 에서 한번에 입력받는다. 8 개의 정보 중 페이로드를 찾아서 저장되어 있는 시그니쳐와 비교한다. 시그니쳐에는 Port 정보 및 IP 정보를 Payload 와 같이 비교해야 할 때는 비교해준다. 비교한 후 해당 Application 의 시그니쳐가 Flow 의 페이로드와 일치한다면 해당 Application 의 이름을 출력한 후 Packets 의 양을 Map 의 value 값으로 출력하고, key 값은 해당 Application 의 Packet 을 명시한 후 출력한다. 예를 들면, 만약 임의의 Flow 가 naver 포탈 에 의해 발생했다면, Key 값은 “s-portal-naver-packets”이고, Value 값은 해당 Flow 에 포함된 Packet 의 양이 된다. 따라서 Reduce 로 전달되었을 때 s-portal-naver-Packets 은 하나로 합쳐져서 naver 에 해당하는 Packet 들의 합을 구할 수 있다. Byte 의 양도 Packet 의 양을 출력할 때와 마찬가지이다. FieldsInfo.txt 에 출력된 모든 Flow 를 읽은 후 Reduce 가 실행되는데, 이때 Reduce 는 Map 에서 출력되었던 Key 값이 같은 것만 해서 한번 실행된다. 따라서, 해당 Application 의 Packet 이라고 명시되어 있는 value 값을 모두 더하면서, 해당 Application 의 Packet 양을 구할 수 있다.

4. 실험 및 결과

앞에서 Hadoop 에 대해 소개했듯이 Hadoop 은 분산 처리 시스템을 지원하는 플랫폼이다. 네트워크 트래픽 양이 많아지고, 데이터의 양이 넘쳐나는 현 시대에 적용해야만 하는 시스템이라고 생각할 수 있다. 따라서 본 연구실에서는 Hadoop 플랫폼 환경을 구성하였다. 총 6 대의 컴퓨터로 Master 노드 1 대, Slave 노드 5 대로 구성되어 있으며, 각 노드는 듀얼 코어, RAM 1G, HDD 200G 로 구성되어있다. 향후 더 나은 스팩의 하드웨어로 설치가 되면, Hadoop 시스템의 장점이 더욱더 부각될 것으로 기대한다.

본 장에서 Hadoop 분산 처리 환경에서 응용 분류된 결과를 통하여 가능성을 증명한다. 응용 트래픽 분류 실험은 대표 포탈 사이트인 naver, nate, daum, google, facebook 그리고 학교포탈사이트의 정의된 시그니쳐를 사용하여 진행하였다. 또한 Hadoop 분산 처리 환경에서 페이로드 시그니쳐 기반 응용 트래픽 분석 시스템의 처리시간과 기존 하나의 머신에서 페이로드 시그니쳐 기반 응용 트래픽 분석 처리시간을 비교하여 Hadoop 에 대한 장점을 제시한다.

Hadoop 분산 처리 환경에서 페이로드 시그니쳐 기반 응용 분석 실험은 위에서 언급했듯이 6 개의 응용을 지정하여, 모든 HTTP 트래픽 중 전체, 분류된 트래픽, 미 분류된 트래픽의 Flow, Packet, Byte 의 양을 분석하고, 분류된 트래픽은 대표 포탈 사이트의 각각의 Flow, Packet, Byte 양을 조사하였다.

표 1 은 Hadoop 분산 처리 환경에서 페이로드 시그니쳐 기반 응용 트래픽 분석 실험 결과이다. 표 1 에서 Time 은 트래픽 수집 시간이다. 1 분부터 24 시간까지 다양하게 트래픽을 수집하여 명확한 결과를 얻는다. Type 은 Flow, Packet, Byte 별로 나타내기 위해 구분하고, Total 은 트래픽 수집 시간 동안 발생한 Flow, Packet, Byte 의 양을 분석한 결과이다. Classification 은 6 개의 응용으로부터 분류된 양, Non-Classification 은 미 분류된 양을 나타낸다.

표 1 에서 볼 수 있듯이 실험은 6 개의 특정 응용 트래픽에 대해서만 분석을 하기 때문에 분석율이

30%~40%로 높지 않은 것을 볼 수 있다. 향후 분석 대상 응용이 증가하면 분석율은 증가할 것이다.

표 1. Hadoop 분산 처리 환경에서 페이로드 시그니쳐 기반 응용 트래픽 분석 결과

Time	Type	Total	Classification	Non-Classification
1m	Flow	50	20	30
	Packet	942	230	712
	Byte(MB)	0.53	0.11	0.42
0.5h	Flow	3,638	1,003	2,635
	Packet(×1000)	157	37	120
	Byte(MB)	134.49	30.01	104.48
1h	Flow	5,989	1,852	3,837
	Packet(×1000)	334	57	277
	Byte(MB)	285	43.61	241.39
3h	Flow	9,966	3,750	6,216
	Packet(×1000)	657	118	539
	Byte(MB)	555.09	87.42	467.67
6h	Flow	13,985	5,144	8,841
	Packet(×1000)	1,145	212	933
	Byte(MB)	993.61	171.42	822.19
9h	Flow	26,993	7,250	19,743
	Packet(×1000)	1,691	298	1,393
	Byte(GB)	1.44	0.24	1.2
12h	Flow	82,736	28,505	54,231
	Packet(×1000)	6,068	1,289	4,779
	Byte(GB)	5.1	1.02	4.08
18h	Flow	252,683	71,861	180,822
	Packet(×1000)	15,268	3,235	12,033
	Byte(GB)	12.7	2.56	10.14
24h	Flow	331,610	94,852	236,758
	Packet(×1000)	20,094	4,376	15,718
	Byte(GB)	16.74	3.48	13.26

다음으로 Hadoop 분산 처리 환경과 하나의 머신 환경에서의 처리속도에 대해 분석한다. 그림 5 는 트래픽 수집 시간대 별 처리 속도 시간에 대해 비교한 그래프이다.

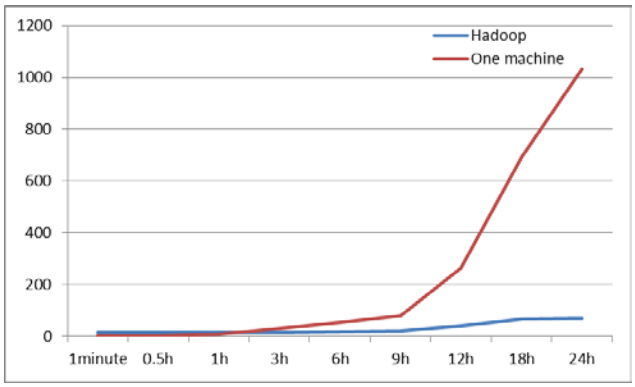

그림 5. 트래픽 수집 시간 별 처리속도

그림 5 의 X 축은 트래픽 수집 시간이고, Y 축은 처리 시간이며 초단위로 나타낸다. 즉, 트래픽 수집 시간이 많을수록 처리해야 할 Flow 데이터가 증가하면서 처리속도가 증가할 것이다. 실험 결과에서 1 시간의 트래픽을 분석할 때까지 기존 하나의 머신 환경에서 분석하는 것보다 Hadoop 환경에서 분석하는 것이 보다 효과적이다. Hadoop 분산 처리 시스템은 데이터를 분할하고, 결합하는 과정인 Map Reduce 과정에서 소요되는 시간이 있기 때문에 적은 데이터 양 분석은 오히려 기존 하나의 머신 환경이 효과적이다. 그러나 3 시간 트래픽을 분석할 때부터는 Hadoop 분산처리 환경에서의 처리속도가 더 빠르다. 또한 기존 하나의 머신 환경은 9 시간의 트래픽 수집 시간부터 처리속도가 급격하게 증가한다.

하지만 Hadoop 시스템과 기존시스템은 실험환경에 차이가 있다. Hadoop 환경은 FieldsExtractor 로 추출된 Text 파일을 분석하고, 기존환경에서는 직접 Binary 형태의 Flow 파일을 분석한다. 따라서 처리시간에 정확한 결과를 기대할 순 없지만 기존 시스템의 분석결과를 분석했을 때, 트래픽 수집 시간이 증가할수록 처리속도는 급격하게 증가하는 것을 파악할 수 있다. 그러나 Hadoop 분산 처리 환경에서는 트래픽 수집 시간이 증가하여도, 처리시간은 크게 증가하지 않는다. 따라서 향후 Hadoop 분산 처리 환경에서 Binary 파일을 사용하더라도, 그림 5 의 결과와 크게 차이 나지 않을 것으로 예상한다.

5. 결론 및 향후 과제

본 논문에서는 Hadoop 기반 페이로드 시그니쳐를 이용한 응용 분석 시스템에 대해 제안하였다. Hadoop 으로 페이로드 시그니쳐를 이용한 응용 분석은 시작단계라서 많은 단점들이 있지만, 가능성을 높일 수 있다는 것을 확인하였다. 또한 기존 응용 트래픽 분류 시스템과 비교를 통해 처리속도에 대한 장점을 증명하였다. Hadoop 은 분석 데이터 양이 적으면 오히려 효과적이지 않지만, 데이터 양이 증가하면 처리속도, 메모리비율, 저장공간 등에서의 부분에서 큰 이점을 보인다.

본 시스템의 분석율에 대한 단점이 있지만, 분석 대상을 증가시키면 또한 분석율도 증가할 것으로 예상한다. 또한, 기존의 방법처럼 Flow 파일을 직접 입력해서 분석하는 방법이 아니라 FieldsExtractor 로 알맞은 형태로 바꾼 후 입력 방법이라는 단점이 있지만, 향후 직접 Flow 파일로 직접 페이로드 시그니쳐를 분석하는 분석기개발하기 위한 초석이 될 것이다. 따라서 본 논문에서 소개하는 Hadoop 기반 페이로드 시그니쳐를 이용한 응용 분석 시스템은 상당한 가치가 있다고 판단된다.

향후 연구로는 처리속도와 편의성과 처리속도를 높이기 위해 Binary 파일을 Hadoop 에서 처리하는 시스템을 구현하여 Text 파일을 만드는 작업을 생략해야 한다. 그리고 Hadoop 환경의 응용 트래픽 분류 시스템과 기존 하나의 머신으로 응용 트래픽 분류 시스템의 비교 실험환경을 완벽히 구현하여, 두 시스템을 비교할 때 Hadoop 환경에서 효과적으로 실험할 수 있는 데이터 양에 대한 최정점을 찾아야 한다. 또한 향후 네트워크 트래픽의 양은 지속적으로 증가될 것으로 예상되기 때문에 응용 트래픽 분

류뿐만 아니라 네트워크 관리의 많은 분야는 Hadoop 기반으로 시도 되어야 한다.

참고 문헌

[1] Zhao-wen LIN, Yan MA “Research of Hadoop-based data Flow management system, Volume 18, Supplement 2, Dec 2011, pp164-168

[2] 박준상, 윤성호, 박진완, 이현신, 이상우, 김명섭, "페이로드 시그니처 기반 트래픽 분석 시스템의 성능 향상", 통신학회논문지 Vol.35 No.9, , Sep. 2010, pp.1287-1294.

[3] 박준상, 박진완, 윤성호, 이현신, 김명섭, "페이로드 시그니처 기반 트래픽 분석 시스템의 성능 향상", Proc. of the 20th Joint Conference on Communications and Information (JCCI) 2010, 충남 안면도 오션캐슬, Apr. 28-30, 2010, pp. 148.

[4] 강원철, 이연희, 이명석 “다중 사용자를 위한 Hadoop 기반 트래픽 분석 시스템 구조”, 한국컴퓨터종합학술대회 Vol.38 No.1 ,2011

[5] Apache Hadoop. http://hadoop.apache.org/

[6] Fnag Yu, Zhifeng Chen, Yanlei Dino, T. V. Lakshman, Randy H. Katz, “Fast and memory Efficient Regular Expression Matching for Deep Packet Inspection” ANCS 2006, December , 2006, San jose, California USA.

[7]Y H Lee, Y S Lee, “Toward scalable internet traffic measurement and analysis with Hadoop” ACM SIGCOMM Computer Communication Review table of contents archive Vol 43 Issue 1, Jan 2013 NewYork, USA.