http://dx.doi.org/10.14400/JDC.2015.13.9.235

하둡 환경에 적합한 클러스터 그룹 기반 속성 정보를 이용한 빅 데이터 관리 기법

한군희*, 정윤수**
백석대학교 정보통신공학과*, 목원대학교 정보통신융합공학부**

Big Data Management Scheme using Property Information based on Cluster Group in adopt to Hadoop Environment

Kun-Hee Han*, Yoon-Su Jeong**
Dept. of Information Communication & Engineeringe, Baeseok University*
Dept. of Information Communication & Engineeringe, Mokwon University**

요 약 소셜 네트워크 기술이 발달하면서 빅 데이터 서비스에 대한 관심이 증가하고 있다. 그러나, 중앙 서버가 아닌 분산 서버에 저장된 데이터를 손쉽게 검색 및 추출하기 위한 기술은 부족한 실정이다. 본 논문에서는 빅 데이터 서비스를 제공하는 컨텐츠 서버와 관리 서버에서 사용자가 원하는 정보의 처리시간을 최소화하기 위한 빅 데이터 관리 기법을 제안하다. 제안 기법은 빅 데이터의 종류, 기능, 특성에 따라 데이터를 그룹으로 분류한 후 분류된 그룹 내 데이터를 속성정보와 연계하여 해쉬체인에 적용한다. 또한, 분산 서버에 저장된 데이터를 최단 시간에 추출하기 위해서 데이터 인덱스 정보(DII, Data Index Information)를 그룹화하여 데이터에 부여된 다중의 속성 정보를 분류하여 데이터의 처리 속도를 향상시킨다. 실험 결과, 클러스터 그룹 수에 따른 데이터의 평균 검색 시간은 평균 14.6% 향상되었고, 키워드 수에 따른 데이터 처리시간은 평균 13% 단축되었다.

주제어 : 빅 데이터, 하둡 환경, 다중 속성, 데이터 관리, 데이터 인덱스 정보

Abstract Social network technology has been increasing interest in the big data service and development. However, the data stored in the distributed server and not on the central server technology is easy enough to find and extract. In this paper, we propose a big data management techniques to minimize the processing time of information you want from the content server and the management server that provides big data services. The proposed method is to link the in-group data, classified data and groups according to the type, feature, characteristic of big data and the attribute information applied to a hash chain. Further, the data generated to extract the stored data in the distributed server to record time for improving the data index information processing speed of the data classification of the multi-attribute information imparted to the data. As experimental result, The average seek time of the data through the number of cluster groups was increased an average of 14.6% and the data processing time through the number of keywords was reduced an average of 13%.

Key Words : Big Data, Hadoop Environment, Multi Attribute, Data Management, Data Index Information

Received 19 July 2015, Revised 21 August 2015
Accepted 20 September 2015
Corresponding Author: Yoon-Su Jeong(Mokwon University)
Email: bukmunro@mokwon.ac.kr

ISSN: 1738-1916

1. 서론

소셜 네트워크의 대중화로 인하여 이질적인 환경에 존재하는 데이터를 손쉽게 검색하기 위한 빅 데이터 서비스가 증가하고 있다[1,2]. 현재, 소셜 네트워크를 통해 이질적인 환경에 저장되고 있는 데이터는 소셜 네트워크의 기술 발달과 함께 남녀노소를 가리지 않고 손쉽게 사용되고 있다[3]. 과거에는 대용량의 데이터를 중앙 서버에서 주로 처리하였지만 최근에는 서버의 과부하 및 피해를 줄이기 위해서 소규모 용량의 데이터를 저장할 수 있는 분산 서버로 운영되고 있는 현실이다. 분산 서버에 저장된 데이터를 서비스하기 위해서는 다양한 종류의 데이터를 생성, 수집, 분석, 표현이 가능해야 하고, 사회 변화에 대한 정확한 예측 가능 정보가 맞춤 형태로 제공되어야 한다[4,5].

하둡 기반의 정보 방법은 크게 노드링크(NL, node-link) 접근방법[6], 행렬 그래프(MAT, matrix graph) 접근방법[7], 노드 링크와 행렬 그래프의 혼합형(hybrid of NL and MAT) 접근방법[8,9] 등 3가지 방법으로 분류된다. 노드 링크 접근방법은 네트워크 구조를 시각화할 수 있는 장점을 가지지만 시각화되는 노드들이 서로 겹치고 에지들이 서로 엇갈리는 문제를 가지고 있다. 행렬 그래프 기반 방법은 노드 링크 기반방법에 비해서 쉽게 노드를 파악할 수 있고, 노드들 간의 경로도 역시 쉽게 파악할 수 있는 장점은 있지만, 행렬 형태로 표현되는 노드들이 행렬 상에 희소행렬(sparse)로 표시되어 많은 공간을 사용하기 때문에 공간에 대한 낭비가 심할 뿐만아니라 행렬 그래프로 표현되는 노드들이 이해하기 힘든 문제점이 있다[6,7]. 혼합형 접근방법은 노드 링크 방법의 에지를 행렬 그래프 방법에 겹쳐서 행렬 그래프 방법의 성능을 향상시키지만 행렬에 연결된 링크들이 복잡하게 구성되어 있고 행렬 상에 나타나는 노드의 관계가 이해하기 어려운 단점이 있다[8,9].

본 논문에서는 빅 데이터 서비스의 처리시간을 최소화하기 위해서 클러스터 기반의 속성정보를 통한 빅 데이터 관리 기법을 제안하다. 제안기법은 빅 데이터의 종류, 기능, 특성에 따라 데이터를 그룹으로 분류한 후 분류된 그룹 내 데이터에 속성정보를 연계하여 해쉬체인 기법에 적용한다. 이 같은 이유는 여러 지역에 분산된 데이터를 손쉽게 추출함으로써 최단 시간에 사용자가 요청한 데이터를 검색할 뿐만 아니라 데이터 인덱스 정보(DII, Data Index Information)를 그룹화 된 데이터에서 생성한 후 데이터에 부여된 다중의 속성 정보를 분류하여 데이터의 처리 속도를 향상시킨다.

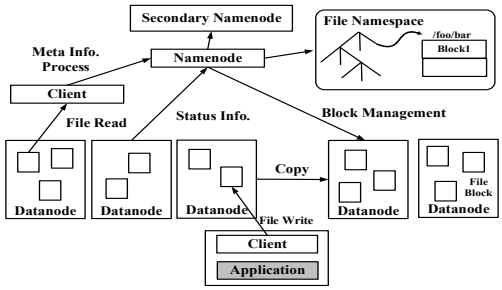
이 논문의 구성은 다음과 같다. 2장에서는 하둡 시스템의 정의 및 특징과 하둡 시스템의 기존 정보 관리 기법해서 알아본다. 3장에서는 분산된 빅 데이터의 검색 속도 및 안정성을 보장하는 속성기반 클러스터 데이터 관리 기법을 제안하고, 4장에서는 제안 기법의 성능평가를 분석하고 마지막으로 5장에서 결론을 맺는다.

2. 관련연구

2.1 하둡 시스템

대용량의 빅 데이터를 분석 처리하기 위한 대표적인 오픈소스 프레임워크인 하둡 시스템은 구글 분산 파일 시스템(GFS, Google File System)[2]과 맵리듀스(MapReduce)를 구현되었다[1,3]. 하둡 시스템의 구성은 구글과 많은 부분에서 유사하게 설계되었으며, 대용량 데이터를 분산 저장하고 관리하는 하둡 분산 파일 시스템과 대용량 데이터의 분석을 수행하는 하둡 맵리듀스(Hadoop Map-Reduce)로 구성된다[4,5].

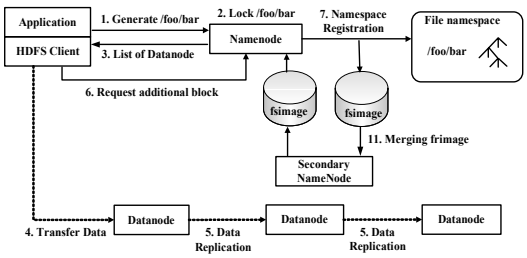
하둡 시스템의 가장 큰 특징은 파일 분산 처리 기술이다. 하둡 시스템은 대규모 데이터를 분산 저장 및 관리하기 위해서, 저장하고자 하는 파일을 블록 단위(기본적으로 64MB로 설정)로 나누어 분산된 서버에 저장한다. 데이터가 블록 단위(64MB)로 나누어 떨어지지 않는 경우, 남은 부분을 그대로 블록으로 저장한다[10,11,12].



[Fig. 1] Construction of Hadoop Distribution File System

[Fig. 1]은 하둡 분산 파일 시스템의 구성을 나타주고 있다. 하둡 분산 파일 시스템은 네임노드(NameNode), 보조 네임노드(Secondary NameNode)와 다수의 데이터노드(DataNode)로 구성된다[5,6]. 네임노드는 하둡 분산 파일 시스템의 모든 메타데이터(디렉토리명, 파일명, 파일 블록 등에 대한 트리 형태의 네임 스페이스)를 관리하며, 클라이언트가 하둡 분산 파일 시스템 상에서 파일 읽기 및 저장을 요청할 때 메타데이터를 기반으로 데이터노드에 저장된 블록 위치를 조회하거나 저장될 파일 복사본의 데이터 노드를 결정한다. 데이터노드는 블록 단위로 나눠진 데이터를 저장하는 데이터 서버로서, 네임노드와 클라이언트의 데이터 입출력 요청을 관리한다[13].

데이터노드는 주기적으로 자신의 상태 정보를 포함한 하트비트(heartbeat) 메시지와 블록의 목록을 담은 블록 리포트(block report)를 네임노드에게 전송한다. 네임노드는 하트비트와 블록 리포트를 통하여 데이터노드의 정상 작동 여부와 데이터노드 내의 모든 블록 목록을 확인하고, 네임노드와 클라이언트의 파일 읽기 및 저장 요청 시 활용한다[14,15].



[Fig. 2] Operation Process of Hadoop Distribution File System

[Fig. 2]은 하둡 분산 파일 시스템의 전체 동작과정을 나타내고 있다. [Fig. 2]처럼 모든 서버들은 완전 연결(fully connected)되어 있으며, TCP(Transmission Control Protocol)기반 프로토콜을 이용하여 통신한다.

2.2 기존 연구

현재 하둡 기반 바이오인포매틱스 정보의 접근방법은 노드링크(NL, node-link) 접근방법[6], 행렬 그래프(MAT, matrix graph) 접근방법[7], 노드 링크와 행렬 그래프의 혼합형(hybrid of NL and MAT) 접근방법[8,9] 등이 주로 연구되고 있다. 노드 링크를 이용한 방법의 경우 네트워크의 전체 구조를 시각화하는데 유용하다. 그러나 시각화되는 노드들이 서로 겹치고 에지들이 서로 엇갈리는 문제를 가지고 있다. 즉, 노드 링크를 이용하여 표현하는 사용자간의 관계가 많아질수록 사용자의 관계를 파악할 수 없는 문제가 있다. 이러한 문제를 해결하기 위해서 샘플링, 필터링, 군집과 같은 후처리를 통하여 필요한 부분만을 정제하여 해결할 수 있는 다양한 방법이 제안되었으나 시각회되는 결과가 이해하기 어렵거나 비용이 많이 드는 문제를 가지고 있다.

행렬 그래프 기반 방법은 노드 링크 기반 방법의 시각화 결과에 대한 가독성 문제를 해결하기 위해서 제안되었다[7]. 행렬 그래프 기반 방법은 노드 링크 기반방법에 비해서 쉽게 노드를 파악할 수 있으며, 노드들 간의 경로도 역시 쉽게 파악할 수 있다. 그러나 행렬 형태로 표현되는 노드들이 행렬 상에 희소행렬(sparse)로 표시되어 많은 공간을 사용하기 때문에 공간에 대한 낭비가 발생되며, 노드 링크 기반방법과 마찬가지로 경로 탐색이 어려운 문제를 가지고 있고, 행렬 그래프로 표현되는 노드들이 이해하기 힘든 문제를 가지고 있다[6].

혼합형 접근방법은 노드 링크 방법의 에지를 행렬 그래프 방법에 겹쳐서 행렬 그래프 방법의 성능을 향상시키고 있다[8,9]. 그러나 이 방법 역시 행렬에 연결된 링크들이 복잡하게 구성되어 있고 행렬 상에 나타나는 노드의 관계가 이해하기 어려운 단점이 있다.

3. 클러스터 기반 속성 정보를 통한 빅 데이터 관리기법

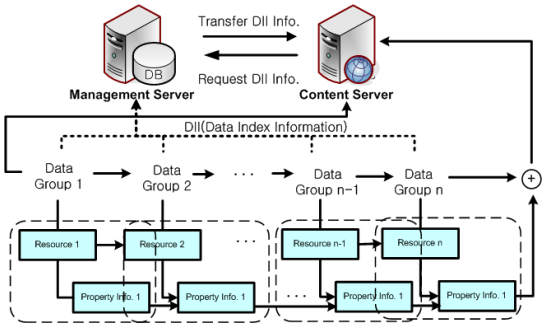
이 절에서는 사용자 요청에 의한 빅 데이터 정보 추출 시간을 최소화하기 위해서 데이터의 종류, 기능, 특성에 따라 데이터를 분류하고, 각 데이터에 속성정보와 연계하도록 해쉬체인을 구성하여 사용자가 안전하게 데이터를 서비스 받을 수 있도록 데이터를 분산 처리한다.

3.1 개요

최근 빅 데이터 서비스는 소셜 네트워크와 의료 서비스를 중심으로 사용량이 증가하고 있어 빅 데이터의 종류 및 특성에 따라 데이터를 관리할 필요성이 높아지고

있다. 빅 데이터 서비스는 사용자의 요구사항에 맞는 데이터를 정확하면서도 신속하게 서비스되어야 한다.

제안 기법에서는 소셜 네트워크나 의료 서비스를 중심으로 관리되고 있는 빅 데이터의 처리 시간을 단축하기 위한 데이터 관리 기법을 제안한다. 제안 기법은 사용자가 원하는 데이터를 신속하게 제공하기 위해서 데이터가 가지고 있는 고유 정보 이외에 복잡한 특성을 다양하게 반영하도록 정보를 연계하도록 구성한다. 제안 기법의 구성요서 중 관리서버는 연계정보에 대한 데이터의 DII(Data Index Information) 정보를 추가로 등록하여 서비스 하는 역할을 담당한다. DII 정보는 관리서버와 컨텐츠 서버간 공유된 정보로써 데이터 관리를 위해 사용된다. 관리서버와 컨텐츠 서버는 관리자가 설정한 설정정보(시간정보)에 따라 주기적으로 동기화가 이루어진다.



[Fig. 3] System Constructure and Process of Proposed Scheme

[Fig. 3]은 빅 데이터 서비스의 처리 시간을 최소화하면서 사용자가 원하는 정보를 정확하게 찾기 위한 제안 기법의 시스템 구성 및 동작과정을 보여주고 있다. [Fig. 3]처럼 제안기법은 데이터와 데이터의 속성정보를 분산 처리 및 저장 관리할 수 있도록 데이터를 블록 단위로 구성한다. [Fig. 3]에서 제안 모델은 빅 데이터와 그 데이터에 속한 속성값을 해쉬체인을 통해 계층적으로 구성하며, DII 정보를 데이터의 검색 키워드로 사용하여 데이터 처리시간을 단축한다.

제안 기법은 데이터의 추출 시간을 단축하기 위해서 크게 5가지 특징을 가진다. 첫째, 빅 데이터 서비스에 사용되는 데이터는 64비트 블록 단위로 데이터를 나누어 분산 서버에 저장한다. 둘째, 데이터 블록은 n개로 구성되며, n-1번째 블록 데이터와 n-2번째 데이터의 속성정보와 연계하여 n-1번째 속성 정보를 생성한다. 셋째, 특성 및 기능이 동일한 데이터 및 속성 정보는 그룹화 시키며, 데이터 그룹 정보는 DII를 생성하여 관리 서버에 전달한다. 넷째, n번째 그룹 정보는 n번째 속성정보와 연계하여 컨텐츠 서버에 전달한다. 다섯째, 시스템 장애가 발생할 경우, 제안 기법은 DII 정보를 이용하여 장애 복구 능력을 갖고 있는 보조 네임노드(secondary namenode)가 데이터 접근을 용이하도록 연결 정보로 활용함으로써 빅 데이터의 접근 제어를 분산 처리한다.

3.2 용어 정의

<Table 1>은 제안 기법에서 사용하는 용어에 대한 설명이다.

〈Table 1〉 Notations

Notation	Definition
D_n	Group data about data block of number of n
$\vec{p_i}$	Property value through data charater
w_i	Keyword for data search
DII_i	index information of Group data
HI_i	i^{th} Hash information such as $H(D_i, \vec{p_{i-1}})$
$H()$	one-way hash chin

3.3 해쉬 체인 기반 분산 환경의 데이터 처리 기법

이 절에서는 데이터의 사용목적에 따라 데이터의 속성정보(종류, 기능, 특성)를 그룹화하여 DII와 연계함으로써 데이터 처리 시간을 단축한다. 특히, 제안 기법에서 데이터들을 그룹화함으로써 발생하는 복잡도는 데이터 정보 집합 정보와 속성값 정보만을 해쉬함수에 적용함으로써 데이터 그룹화에 발생하는 복잡도를 최소화하였다.

3.3.1 데이터 인덱스 정보 생성 과정

이 절에서는 데이터와 속성정보를 사용목적에 따라 그룹화한 데이터 그룹을 생성하여 빅 데이터 환경에 분산된 데이터를 계층적으로 관리할 수 있도록 데이터 인덱스 정보를 생성하는 과정을 5단계로 구성한다.

- 단계 1 : 64비트 단위로 데이터를 블록으로 나누어 분산될 수 있도록 데이터를 식 (1)처럼 생성한다. 식 (1)

에서 d_i는 빅 데이터에서 생성되는 블록 단위의 최소 데이터를 의미한다. 또한, D_n는 n개로 구성된 데이터 블록에 대한 그룹 데이터를 의미한다.

$$D_n=\{d_i \in Z \mid d_1, ..., d_i\},\ 1 \leq i \leq n \bmod 64 \quad (1)$$

• 단계 2 : 식 (1)에서 생성된 데이터 블록에 대한 그룹 데이터 D_n는 데이터 특성에 따라 속성값 $\vec{p_i}$을 식 (2)처럼 생성한다.

$$\vec{p_i} = (p_1, p_2, ..., p_n) = \{p_i \in d_i | 1 \leq i \leq n \bmod 64\} \quad (2)$$

여기서 n은 생성된 데이터 블록의 크기를 의미하고 $\vec{p_i}$는 데이터 블록에 대한 속성값을 의미한다.

• 단계 3 : 데이터 블록으로 구성된 데이터 정보 집합 D_i과 속성값 $\vec{p_{i-1}}$을 식 (3)~식 (4)처럼 해쉬함수에 적용한다.

$$H: \{0,1\} \rightarrow Z_N \quad (3)$$

$$HI_i = H(D_i, \vec{p_{i-1}}),\quad 1 \leq i \leq n \quad (4)$$

여기서, H_i: $\{0,1\} \rightarrow Z_N$는 안전한 해쉬함수를 의미한다.

• 단계 4 : 데이터 그룹 간 생성된 그룹 데이터 인덱스 정보 DII_i는 식 (5)처럼 해쉬함수에 적용하여 생성한다. 생성된 데이터 인덱스 정보 DII_i는 관리 서버에 등록한다.

$$DII_i = H(D_i) \in L_{w_i},\quad 1 \leq i \leq n \quad (5)$$

여기서, w_i는 데이터 검색을 위한 키워드를 의미하며, L_{w_i},는 키워드 w_i를 설정하여 관리 서버에 등록하여 데이터 인덱스 정보 DII_i와 함께 데이터 추출에 사용된다.

• 단계 5 : 데이터 블록을 적용한 해쉬함수 HI_i 정보와 데이터 인덱스 정보 DII_i를 식 (6)처럼 연접하여 컨텐츠 서버에 전달한다.

$$HI_i \oplus DII_i,\quad 1 \leq i \leq n \quad (6)$$

3.3.2 데이터 정보 추출 과정

이 과정은 관리 서버를 통해 컨텐츠 서버에 등록되어 있는 빅 데이터 정보를 추출하여 서비스를 제공하기 위한 과정이다.

• 단계 1 : 관리서버는 데이터 정보 추출과 관련하여 컨텐츠 서버에게 식 (7)과 같은 정보 쌍을 보내어 서비스를 요청한다.

$$Transfer \quad (w_i, HI_i, D_i, DII_i) \quad (7)$$

• 단계 2 : 컨텐츠 서버는 데이터베이스에 저장되어 있는 정보 중 데이터 검색을 위한 키워드 w_i와 데이터 정보 집합 D_i을 이용하여 데이터의 속성 정보를 확인한다.

$$Check \quad \vec{p_{i-1}} \text{ and } Compare \quad HI_i \cong HI_i' \quad (8)$$

• 단계 3 : 콘텐츠 서버는 데이터 정보 집합 D_i'에 대해서 속성 정보 $\vec{p_{i-1}}$와 일치하는 데이터의 종류, 기능, 특성에 따라서 데이터 인덱스 정보 DII_i' 생성한다.

$$DII_i' = H(D_i') \in L_{w_i}',\quad 1 \leq i \leq n \quad (9)$$

• 단계 4 : 컨텐츠 서버는 생성된 데이터 인덱스 정보 DII_i'와 관리 서버가 요청한 데이터 인덱스 정보 DII_i를 비교하여 동일하면 컨텐츠 서버에게 데이터 정보를 전달한다.

$$Compare \quad DII_i = DII_i' \quad (10)$$

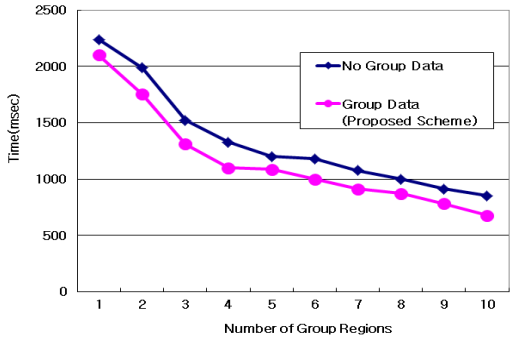
$$Transfer \quad New\ Data\ Group\ infor. = \sum_{i=1}^{n} D_i \quad (11)$$

4. 성능 평가

이 절에서는 정보 검색 평균 검색 시간, 키워드 수에 따른 처리 시간, 통신 지연시간 등으로 평가한다.

4.1 평균 정보 검색 시간

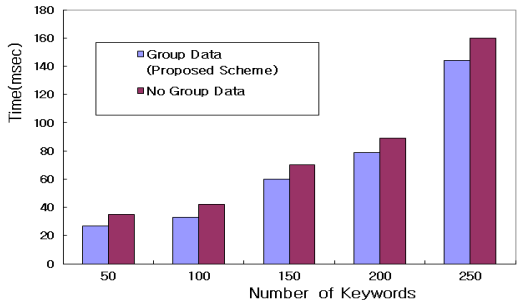
[Fig. 4]는 키워드를 통해 사용자의 정보 검색 요청시 키워드와 동일한 1000개의 정보를 그룹 크기로 랜덤하게 정보를 검색할 때의 평균 정보 검색 시간을 나타내고 있다.



[Fig. 4] Average Information Search Time thourgh Number of Group Regions

[Fig. 4]의 결과, 데이터를 속성과 키워드로 그룹화한 경우와 그룹화하지 않는 경우 모두 그룹 수가 증할 수 록 비례적으로 평균 정보 검색 시간이 줄어들었지만, 데이터를 속성과 키워드로 그룹화한 경우가 그룹화하지 않는 경우보다 평균 정보 검색 시간은 14.6% 낮게 나타났다. 이 같은 결과는 속성과 키워드로 데이터를 분류하여 유사 데이터를 검색할 경우 데이터의 정확도 및 처리시간이 빠르기 때문에 나타난 결과이다.

4.2 키워드 수에 따른 데이터 처리시간

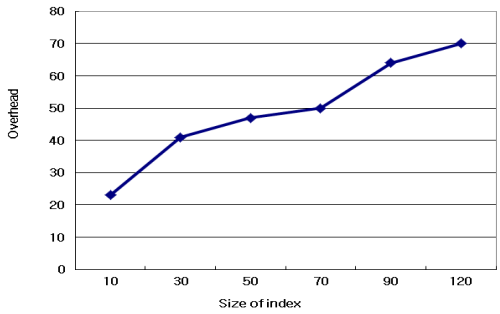


[Fig. 5] Data Process time thourgh Number of Keywords

[Fig. 5]은 동일 크기의 그룹 데이터에서 키워드 수에 따른 정보의 업데이트 시간을 보여주고 있다. [Fig. 5]의 결과, 키워드 수가 증가할수록 데이터 그룹과 데이터 비그룹의 데이터 처리 시간은 모두 비례적으로 증가하였다. 그러나, 데이터 그룹의 데이터 처리시간이 데이터 비그룹의 처리시간보다 13% 단축되었으며, 데이터를 추가하거나 삭제할 때도, 데이터 그룹은 데이터를 추가할 경우 평균 6.7% 단축되었고 데이터를 삭제할 경우에는 7.6% 단축되었다. 이 같은 결과는 데이터를 인덱스 정보와 속성정보를 해쉬체인하여 데이터를 그룹관리하기 때문에 나타난 결과이다.

4.3 데이터 인덱스 크기에 따른 오버헤드

[Fig. 6]은 데이터 인덱스의 크기에 따른 관리 서버의 오버헤드를 보여주고 있다.



[Fig. 6] Overhead thourgh Size of Data Index

[Fig. 6]의 실험결과, 데이터 인덱스의 크기가 증가할수록 오버헤드 또한 증가하는 결과를 얻었다. 특히, 데이터 인덱스의 크기가 30~70사이에는 증가율이 낮았지만 10~30과 70~120 사이의 데이터 인덱스 크기의 오버헤드 증가율은 높았다. 이 같은 결과는 데이터 관리가 계층적으로 이루어지면서 초기와 마지막에 데이터 처리시간이 일정하게 증가하기 때문에 나타난 결과이다.

5. 결론

최근 소셜네트워크와 같은 SNS 기술이 발달하면서 빅 데이터 서비스 개선 요구사항이 증가하고 있다. 본 논문에서는 대용량의 빅 데이터 중 사용자가 요구하는 데이터를 서버의 부하없이 데이터를 추출하는 시간을 줄이는 데이터 관리기법을 제안하였다. 실험 결과, 데이터를 속성과 키워드로 그룹화한 경우가 그룹화하지 않는 경우보다 평균 정보 검색 시간은 14.6% 낮았고, 데이터 그룹

의 데이터 처리시간이 데이터 비그룹의 처리시간보다 13% 단축되었다. 또한, 데이터 인덱스의 크기가 증가할수록 오버헤드 또한 증가하는 결과를 얻었다. 향후 연구에서는 본 연구의 결과를 빅 데이터 시스템에 실제 적용하여 성능평가를 수행할 계획이다.

REFERENCES

[1] H. Hu, Y. Wen, T. S. Chua, X. Li, “Toward Scalable Systems for Big Data Anaqlytics: A Technology Tutorial”, IEEE Access, vol. 2, pp. 652-687, 2014.

[2] P. Russom, “Big Data Analytics”, TDWI Research Fourth Quarter, pp. 6, Dec. 2011.

[3] V. Gadepally, J. Kepner. “Big data dimensional analysis”, 2014 IEEE High Performance Extreme Computing Conference(HPEC) pp. 1-6, Sep. 2014.

[4] Y. Demchenko, C. De Laat, P. Membrey, “Defining architecture components of the Big data Ecosystem”, 2014 International conference on Collaboration Technologies and Systems(CTS), pp.104-112, May, 2014.

[5] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, A. H. Byers, “Big Data: The Next Frontier for Innovation, Competition and Productivity”, Mckinsey Global Institute, pp. 1-137. 2011.

[6] S. Abdul-Rahman, A. A. Bakar, Z. -A, Mohamed-Hussein, "Optimizing Big Data in Bioinformatics with Swarm Algorithms", 2013 IEEE 16th International Conference on Computational Science and Engineering (CSE), pp. 1091-1095, Dec. 2013.

[7] A. Chong, T. D. Gedeon, L. T. Koczy, "Hierarchical fuzzy classifier for bioinformatics data", 2003. Proceedings. Seventh International Symposium on Signal Processing and Its Applications, pp. 45-48, July 2003.

[8] E. Ahmed, "Resource capability discovery and description management system for bioinformatics Data and service Integration - an experiment with gene regulatory networks", 2008. ICCIT 2008. 11th International Conference on Computer and Information Technology, pp. 56-61, Dec. 2008.

[9] Jiang Peiyong, Sun Xiaoxi, E.Z. Chen, Sun Kun, R. W. K. Chiu, Y. M. D. Lo, Sun Hao, "Methy-Pipe: An integrated bioinformatics data analysis pipeline for whole genome methylome analysis", 2010 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW), pp. 585-590, Dec. 2010.

[10] A. Katal, M. Wazid, R. H. Goudar, “Big data: Issues, challenges, tools and Good practices ”, 2013 Sixth International Conference on Contemporary Computing(IC3), pp. 404-409, Aug. 2013.

[11] Y. C. Jung. “Big Data revolution and media policy issues”, KISDI Premium Report, Vol. 12, No. 2, pp. 1-22, 2012.

[12] S. H. Kim, N. U. Kim, t. M. Chung, “Attribute Relationship Evaluation Methodology for Big Data Seucrity”, 2013 International Conference on IT Convergence and Security(ICITCS), pp. 1-4, Dec. 2013.

[13] S. Y. Son, “Big data, online marketing and privacy protection”, KISDI Premium Report, Vol. 13, No. 1, pp.1-26, 2013.

[14] J. T. Kim, B. J. Oh, J. Y. Park, “Standard Trends for the BigData Technologies”, 2013 Electronics and Telecommunications Trends, Vol. 28, No. 1, pp. 92-99, 2013.

[15] M. Paryasto, A. Alamsyah, B. Rahardjo, Kuspriyanto, “Big-data security management issues”, 2014 2nd International Conference on Information and Communication Technology(ICoICT), pp. 59-63, May, 2014.

한 군 희(Han, Kun Hee)

- 2000년 2월 : 충북대학교 컴퓨터공학과(공학박사)
- 2001년 3월 ~ 현재 : 백석대학교 정보통신학부 교수
- 관심분야 : 멀티미디어, 정보보호
- E-Mail : hankh@bu.ac.kr

정 윤 수(Jeong, Yoon Su)

- 2000년 2월 : 충북대학교 전자계산학과 이학석사
- 2008년 2월 : 충북대학교 전자계산학과 이학박사
- 2009년 8월 ~ 2012년 2월 : 한남대학교 산업기술연구소 전임연구원
- 2012년 3월 ~ 현재 : 목원대학교 정보통신융합공학부 조교수
- 관심분야 : 센서 보안, 암호이론, 정보보호, Network Security, 이동통신보안
- E-Mail : bukmunro@mokwon.ac.kr