

맵-리듀스의 처리 속도 향상을 위한 데이터 접근 패턴에 따른 핫-데이터 복제 기법

A Hot-Data Replication Scheme Based on Data Access Patterns for Enhancing Processing Speed of MapReduce

손인국, 류은경, 박준호, 복경수, 유재수
충북대학교 정보통신공학부

Ingook Son(dlsnr94422@nate.com), Eunkyung Ryu(lyk1728@hanmail.net),
Junho Park(junhopark@chungbuk.ac.kr), Kyoungsoo Bok(ksbok@chungbuk.ac.kr),
Jaesoo Yoo(yjs@chungbuk.ac.kr)

요약

최근 대규모 데이터의 처리와 관리를 위한 분산 저장 및 처리 시스템의 연구 및 활용이 중요해지고 있다. 대표적인 분산 저장 및 처리 프레임워크로써 하둡(Hadoop)이 널리 활용되고 있다. 하둡 분산 파일 시스템을 기반으로 수행되는 맵-리듀스에서의 태스크 할당은 데이터의 지역성(locality)를 고려하여 최대한 가깝게 할당한다. 하지만 맵-리듀스에서의 데이터 분석 작업에서 작업 형태에 따라 빈번하게 요청되는 데이터가 존재한다. 이러한 경우, 해당 데이터의 낮은 지역성으로 인해 수행시간 증가 및 데이터 전송의 지연의 문제점을 야기 시킨다. 본 논문에서는 맵-리듀스의 처리 속도 향상을 위한 데이터 접근 패턴에 따른 핫-데이터 복제 기법을 제안한다. 제안하는 기법에서는 데이터 접근 패턴에 따라 높은 접근 빈도를 보이는 핫-데이터에 대한 복제본 최적화 알고리즘을 활용하여 데이터 지역성을 향상시키고 결과적으로 작업 수행시간을 감소시킨다. 성능평가 결과, 기존 기법에 비해 접근 빈도의 부하가 감소하는 것을 확인하였다.

■ 중심어 : | 분산 컴퓨팅 | 맵-리듀스 | 하둡 | 로컬리티 | 핫-데이터 |

Abstract

In recently years, with the growth of social media and the development of mobile devices, the data have been significantly increased. Hadoop has been widely utilized as a typical distributed storage and processing framework. The tasks in Mapreduce based on the Hadoop distributed file system are allocated to the map as close as possible by considering the data locality. However, there are data being requested frequently according to the data analysis tasks of Mapreduce. In this paper, we propose a hot-data replication mechanism to improve the processing speed of Mapreduce according to data access patterns. The proposed scheme reduces the task processing time and improves the data locality using the replica optimization algorithm on the high access frequency of hot data. It is shown through performance evaluation that the proposed scheme outperforms the existing scheme in terms of the load of access frequency.

■ keyword : | Distributed Computing | MapReduce | Hadoop | Data Locality | Hot-Data |

* 본 논문은 2013 춘계종합학술대회에서 '맵-리듀스의 처리 속도 향상을 위한 데이터 접근 패턴에 따른 핫-데이터 복제 기법'의 제목으로 발표된 논문을 확장한 것임.

* 이 논문은 2012년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(2012R1A1A2A10042015)과 미래창조과학부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(NIPA-2013-H0301-13-4009)

접수일자 : 2013년 08월 01일

심사완료일 : 2013년 08월 11일

수정일자 : 2013년 08월 09일

교신저자 : 유재수, e-mail : yjs@chungbuk.ac.kr

I. 서론

최근 소셜 미디어의 성장과 모바일 장치와 같은 디지털 기기가 삶의 전반에 배치되면서 일상 속에서 다양한 종류의 대규모 데이터가 급속히 생성, 유통, 저장되고 있다. 이와 함께 방대한 데이터를 축적하고 분석하려는 노력이 이어지고 있다. 디지털로 축적된 다양한 데이터의 분석을 통해 기존에 보지 못했던 새로운 가치들이 체계적으로 도출되고 있기 때문이다. 몇 해 전부터 꾸준히 이슈화된 빅 데이터를 본격적으로 활용하기 위한 연구가 본격적으로 궤도에 올라서고 있는 모양새다.

이렇게 기하급수적으로 늘고 있는 데이터를 상용 솔루션으로 감당하기엔 그에 따른 비용 부담 또한 기하급수적으로 늘어날 수밖에 없다[1][2]. 그러므로 대규모 데이터 처리와 관리를 위한 분산 저장 및 처리 시스템의 연구 및 활용이 중요해지고 있으며, 최근 그 해답을 오픈소스 솔루션에서 얻고 있다. 하둡(Hadoop)[3]은 대규모 자료의 저장 및 처리를 위한 분산 응용 프로그램을 지원하는 대표적인 오픈소스 소프트웨어 프레임워크이다. 하둡은 페타바이트 이상의 대규모 데이터를 클러스터 환경에서 저장하기 위한 하둡 분산 파일 시스템(HDFS : Hadoop Distributed File System)[4]과 이를 기반으로 병렬 처리를 지원하기 위한 맵-리듀스(MapReduce)[5] 프레임워크로 구성된다. 하둡 분산 파일 시스템과 맵-리듀스는 실제 사례에서 확인된 것과 같이 높은 활용성으로 인해 많은 주목을 받고 있으며, 추가적인 성능 향상을 위한 연구가 활발하게 진행되고 있다[6][7].

맵-리듀스 프레임워크는 함수형 프로그래밍에서 일반적으로 활용되는 맵(Map)과 리듀스(Reduce) 함수 기반으로 구성된다. 맵 단계에서는 함수의 정의에 따라 청크(데이터 블록)를 읽어서 가공된 데이터를 키-값의 형태로 변화하는 작업을 수행하고, 리듀스 단계에서는 맵 단계의 결과를 병합하여 출력한다.

하둡 분산 파일 시스템은 데이터의 유실이나 장애 발생을 고려하여 기본적으로 3개의 청크 복제본을 노드, 랙, 오프-랙에 저장한다. 하둡 분산 파일 시스템을 기반으로 수행되는 맵-리듀스에서의 태스크 할당은 데이터

의 로컬리티를 고려하여 최대한 가깝게 할당한다. 다시 말해, 태스크 수행에 필요한 청크가 위치한 노드에 태스크를 할당하거나 해당 노드가 다른 태스크를 수행 중일 경우 노드가 위치한 랙의 다른 노드로, 노드가 위치한 랙의 다른 노드도 모두 다른 태스크를 수행 중일 경우 다른 랙의 태스크를 수행 중이지 않은 노드로 할당한다. 이 때, 처리 대상 데이터가 다른 노드에 위치할 경우, 처리 노드로 해당 청크를 불러오기 위한 데이터 검색, 접근, 전송 등의 추가적인 작업을 필요로 하고, 이는 해당 작업 수행 시간의 증가를 야기한다. 뿐만 아니라, 다수의 노드로 구성되는 네트워크상의 한정된 대역폭을 점유함으로써 다른 데이터의 전송 역시 지연시키는 문제가 발생한다.

맵-리듀스에서의 데이터 분석 작업에서 작업 형태에 따라 빈번하게 요청되는 데이터가 존재한다. 예를 들어, 특정 사전 데이터와 비교를 수행할 경우, 기준이 되는 사전 데이터는 대부분의 노드에서 필요로 하게 된다. 데이터 비교를 수행할 경우 사전 데이터의 로컬리티가 낮은 상태에서 수행시간 증가 및 데이터 전송의 지연의 문제점을 야기한다[8].

이러한 문제점을 고려하여, 본 논문에서는 맵-리듀스의 처리 속도 향상을 위한 데이터 접근 패턴에 따른 핫-데이터 복제 기법을 제안한다. 제안하는 기법에서는 데이터 접근 패턴에 따라 높은 접근 빈도를 보이는 핫-데이터에 대한 복제본 최적화 알고리즘을 활용하여 기존 하둡 프레임워크에서 빈번히 요청되는 비교데이터를 보관하는 노드가 접근 빈도가 높아 노드 부하가 생기는 문제점을 해결 한다. 그 결과 데이터 로컬리티를 향상시킴에 따라 작업 수행시간을 감소시키는 것이 가능하다.

본 논문의 구성은 다음과 같다. 제 2절에서는 기존에 제안된 하둡 프레임워크의 문제점과 연구 목적을 설명한다. 제 3절에서는 제안하는 맵-리듀스의 처리 속도 향상을 위한 데이터 접근 패턴에 따른 핫-데이터 복제 기법을 기술한다. 제 4절에서는 기존 기법과의 성능 평가를 통해 제안하는 기법의 우수성을 보이며, 마지막 제 5절에서는 결론과 향후 연구 방향을 제시한다.

II. 관련 연구

대용량 데이터를 처리하기 위한 기법들이 활발하게 연구되고 있다. 이러한 대용량 데이터를 처리하기 위한 대표적인 프레임워크인 하둡은 대량의 자료를 처리할 수 있는 분산 응용 프로그램을 지원한다. 하둡은 구글에서 활용되는 기반 시스템인 구글 파일 시스템(GFS: Google File System)[9]을 이용한 하둡 분산 파일 시스템과 이를 기반으로 하는 데이터 분산 처리 프레임워크인 맵-리듀스를 구현한 기술이다.

맵-리듀스는 맵과 리듀스의 2가지 함수를 조합하여 데이터를 처리한다. 맵은 데이터의 집합을 받아들여 사전에 정해진 처리를 통해 새로운 데이터를 생성하는 프로세스이며, 리듀스는 맵에 의해 만들어진 데이터를 모아 최종적으로 원하는 결과로 만들어 내는 프로세스이다.

[그림 1]과 같이 하둡 프레임워크는 위에 설명한 맵-리듀스와 하둡 파일 시스템으로 구분된다. 마스터(Master) 노드의 잡 트래커는 각 슬레이브(Slave) 노드의 태스크 트래커(Task Tracker)에게 잡 수행시간과 데이터 노드의 위치를 고려하여 잡을 할당한다. 그리고 네임노드는 초기 데이터가 입력될 때 각 청크 데이터의 위치를 포함한 메타데이터를 관리하므로, 잡 트래커가 잡을 할당할 때 활용한다. 그리고 데이터의 초기 배치는 랜덤하게 배치하며 데이터의 유실 또는 장애 발생을 고려하여 기본적으로 3개의 청크 복제본을 유지한다.

하지만 맵-리듀스에서의 데이터 분석 작업에서 작업 형태에 따라 빈번하게 요청되는 데이터가 존재한다. 예를 들어, 특정 사전 데이터와 비교를 수행할 경우, 기준이 되는 사전 데이터는 대부분의 노드에서 필요로 하게 된다. 데이터 비교를 수행할 경우 사전 데이터의 로컬리티가 낮은 상태에서 수행시간 증가 및 데이터 전송의 지연의 문제점을 야기시킨다. 그리고 데이터를 데이터 노드에 청크 크기로 배치를 할 때 각 청크 데이터를 3개의 복제본을 위치를 고려하지 않고 랜덤으로 배치하여 같은 데이터가 유사한 위치에 쏠림(Skewed) 현상이 나타난다.

이러한 문제점을 고려하여, 접근 빈도에 따른 핫-데이터를 추출 후 청크 복제본을 생성하여 노드의 부하를

감소하며 로컬리티를 고쳐한 복제본 배치로써 데이터의 쏠림 현상을 해결하는 연구가 필요하다.

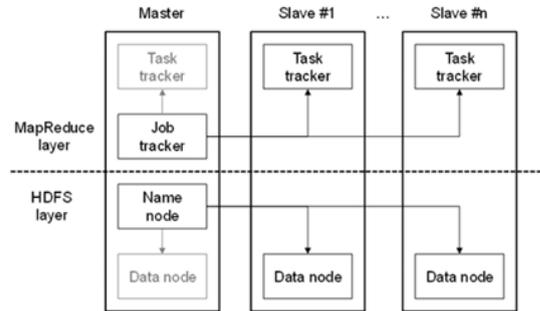


그림 1. Hadoop의 물리적인 구조

III. 제안하는 기법

3.1 제안하는 시스템의 구조

본 절에서는 맵-리듀스 처리 속도 향상을 위한 데이터 접근 패턴을 고려한 핫-데이터 복제 기법 알고리즘을 제안한다. 제안하는 기법에서는 기존 맵-리듀스에서의 데이터 분석 작업 형태에 따른 노드 부하 문제와 데이터 배치의 쏠림 현상을 해결하기 위해 네임노드에서 데이터 접근 관리 테이블을 생성 한 후 접근 빈도에 따른 핫-데이터 추출 후 복제본 수를 산출한다. 또한 데이터의 쏠림 현상을 해결하기 위해 로컬리티를 최대한 고려한 복제본 배치를 한다.

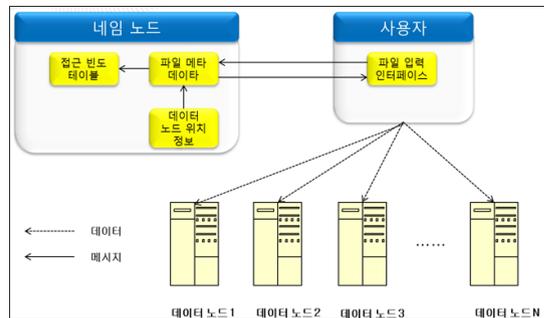


그림 2. 제안하는 시스템 구조

[그림 2]는 제안하는 데이터 알고리즘을 위한 시스템

구조를 나타낸다. 제안하는 기법의 시스템 구조는 기존 기법의 시스템 구조와 유사하지만 하둡 프레임워크의 네임노드를 수정하였다. 네임노드는 파일 메타데이터를 통해 각 청크 데이터의 위치를 기록하고, 접근 빈도 테이블을 통해 데이터의 접근 빈도를 기록하여 핫-데이터 추출에 이용한다. 그리고 데이터 노드 위치 정보를 통해 로컬리티를 최대한 고려한 복제본을 추가 배치한다.

3.2 핫-데이터 추출

제안하는 기법에서는 핫-데이터를 추출하기 위해 네임 노드(NameNode)에서 데이터의 접근 패턴을 기록한다. 맵-리듀스 프레임워크에서는 클라이언트로부터 제출된 작업은 잡 트래커에 의해 태스크로 분할되어 태스크 트래커에 할당된다. 잡 수행에 필요한 데이터를 찾기 위해 태스크 트래커는 하둡 분산 파일 시스템에 접근하여 메타 데이터를 활용한다. 제안하는 기법의 수행을 위해, 데이터를 관리하는 네임노드(NameNode) 상에 태스크의 데이터 접근 빈도를 파악하기 위한 모듈을 탑재하고, [표 1]과 같은 접근 패턴 저장 구조를 생성한다. 파일 이름은 원본 파일의 위치를 포함한 이름이며, 복제본 리스트는 복제본의 위치를 포함한 이름의 리스트, 블록 아이디는 데이터 노드에서의 복제본이 저장된 블록의 위치를 나타내며 접근 수는 각 복제본에 접근된 수를 나타낸다.

표 1. 데이터 접근패턴 테이블

Filename	Replica_list	block_id	accessCount	storage Time
파일 이름	복제본 리스트	블록 아이디	접근 수	데이터 저장시간

접근 빈도에 따른 최적의 복제본의 수를 도출하기 위해 평균 태스크 처리 시간당 접근 빈도를 사용한다. 이 때, 평균 태스크 처리 시간은 식 (1)을 기반으로 산출된다.

$$Time_{avg} = \frac{Total\ time\ of\ Jobprocess}{Number\ of\ Totaltasks} \quad (1)$$

식(1)은 하둡 프레임워크가 클라이언트로부터 잡

처리 시간당 모든 노드에 할당된 전체 태스크를 의미한다.

```
//Replication Scheme Based on Data Access Patterns
BEGIN //데이터 입력 시작

FOR EACH chunk i( 0 ≤ i < n )
  metadata[i] of NameNode record:
  DataNode Arrangement;
END FOR EACH

END
BEGIN //질의 입력 시작

FOR EACH query i( 0 ≤ i < n )
  Access Table record:
   $Time_{avg} = \frac{Total\ time\ of\ Jobprocess}{Number\ of\ Totaltasks}$  ;
//평균 태스크 처리 시간 계산
 $Replica_{count} = \frac{access\ Count \times Time_{avr}}{storage\ Time}$  ;
//복제본 수 산출
 $Distance_{ij} = |Chunk\_IP_i - Chunk\_IP_j|$ ;
if(Distanceij < arri[num])
  arri[num+1] = arri[num];
  arri[num] = Chunk_IPi;
  num++;
else
  arri[num+1] = Chunk_IPj;
  num++;
//다른 노드와의 거리 연산
FOR EACH Replica j( 0 ≤ j < n )
if(j != 0)
  j Replica position = arri[arri.size - j];
else
  j Replica position = arri[0];
END FOR EACH
// 복제본 배치 위치 지정
END
```

그림 3. 제안하는 시스템 알고리즘

이를 통해, 태스크 당 평균 처리 시간을 기반으로 최적의 복제본의 수를 산출하는 것이 가능하다. 식 (2)는 평균 태스크 처리 시간, 접근 수 그리고 데이터 저장 시간을 사용하여 각 데이터의 복제본의 수를 산출하는 연산을 나타낸다.

$$Replica_{count} = \frac{access\ Count \times Time_{avr}}{storage\ Time} \quad (2)$$

3.3 복제본 배치 알고리즘

제안하는 기법에서는 최적의 복제본의 수의 산출 후, 복제본을 임의로 배치하는 것이 아닌 로컬리티를 최대

한 고려한 배치 전략을 취한다. 하둡 프레임워크는 [그림 4]와 같이 내부적으로 거리에 따른 IP 주소를 할당 정책을 활용한다. 세부적으로 주소 앞부분부터 일치할 수록 근거리 노드라고 할 수 있다. 이러한 거리에 따른 IP 주소 할당을 이용하여 복제본을 배치한다. 복제본을 균등하게 배치하기 위해 복제본의 배치는 기존 데이터 배치 노드의 주소를 검사하여 데이터가 존재하지 않은 곳에 배치한다. 예를 들어, 10.0.1.1 노드에만 데이터가 존재한다면 10.1.Y.Y에 복제본을 우선 배치한다. 이러한 복제본 배치 정책으로 데이터의 로컬리티를 최대화하는 것이 가능하다.

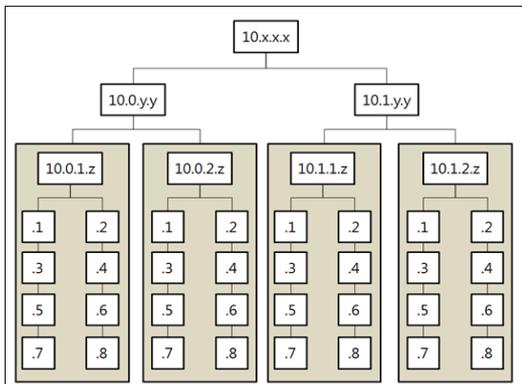


그림 4. 거리에 따른 주소 체계

IV. 성능 평가

4.1 실험 환경

본 절에서는 제안하는 기법의 우수성을 입증하기 위해, 기존 하둡 프레임워크[3]와의 시뮬레이션을 통한 성능 평가를 수행하였다. 본 시뮬레이션은 자체 Java 시뮬레이터를 구현하여 수행하였다. [표 2]은 성능평가 환경을 나타낸다. 클러스터 내의 전체 노드 수 100개, 전체 체크 데이터를 1000개로 고정된 후, 태스크 수행에 필요한 비교 데이터를 50~200개로 변경하며 전체 노드의 접근 빈도의 표준편차를 비교 평가하였다. 여기서 비교 데이터는 질의 처리 시 다른 데이터와의 비교를 필요로 하여 데이터의 접근이 빈번이 일어나는 데이터를 말한다. 또한, 네트워크 대역폭의 변화에 따른 질의

처리시간을 비교 평가하였다.

표 2. 성능평가 환경

파라미터	값
전체 노드의 수(EA)	100
전체 체크 데이터(개)	1000
비교데이터(개)	50 ~ 200
대역폭(Mb/s)	4 ~ 32

4.2 실험 결과

[그림 5]는 접근 빈도의 분포를 기존 하둡 프레임워크의 접근빈도와 비교 평가한 결과이다. 본 성능 평가에서는 비교 데이터에 따른 각 노드의 접근 빈도를 비교 평가 하였다. 표준편차는 접근빈도의 분포를 의미한다. 기존 하둡 에서 자주 요청되는 비교 데이터의 경우 기존 3개의 복제 데이터만을 사용하므로 노드의 접근이 많은 노드가 발생한다. 하지만 제안하는 기법은 요청되는 빈도에 따라 복제본의 수를 산출하여 데이터를 균등하게 배치를 하여 각 노드에 데이터 요청의 빈도가 모든 노드에게 균등하게 되는 것이 가능하다. 성능 평가 결과, 기존 하둡 프레임워크에 비해 제안하는 기법의 접근 빈도의 부하가 평균 약 8% 감소하였다.

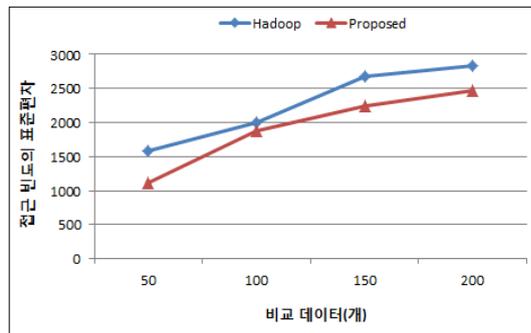


그림 5. 비교 데이터 수에 따른 접근 빈도 표준편차

[그림 6]은 대역폭에 따른 질의 처리 시간을 비교 평가한 결과이다[10]. 본 성능 평가에서는 클러스터를 구성하는 노드는 100개, 전체 체크 데이터의 수는 1000개 및 비교 대상 데이터는 200개로 설정하였다. 기존 기법에서는 자주 요청되는 체크 데이터를 보유한 노드에 많은 부하가 생겨 질의 처리 시간이 높지만, 제안하는 기

법에서는 복제본의 수를 증가시킴으로써 노드 부하를 감소시키고, 결과적으로 질의처리 시간을 최소화하는 것이 가능하다. 성능 평가 결과, 기존 Hadoop에 비해 제안하는 기법에서의 질의 처리 시간이 평균 약 55% 감소하였다.

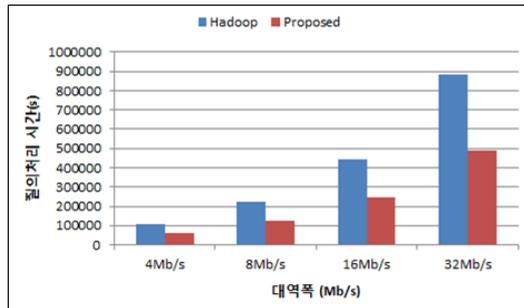


그림 6. 대역폭에 따른 질의 처리시간

본 논문에서는 기존 하둡 프레임워크와 비교 데이터의 수에 따른 노드 접근 빈도의 표준편차와 대역폭을 변화에 따른 질의 처리 시간을 비교하였다. 결과적으로 비교데이터를 보관하는 노드에 집중되는 접근 부하가 다른 노드에도 분산 되는 것을 확인하였으며, 그 결과 질의 처리속도가 향상된 것을 확인하였다.

V. 결론 및 향후 연구

본 논문에서는 기존 기법의 문제점을 분석하고 맵-리듀스의 처리 속도 향상을 위한 데이터 접근 패턴에 따른 핫-데이터 복제 기법을 제안하였다. 제안하는 기법에서는 데이터 접근 패턴에 따라 높은 접근 빈도를 보이는 핫-데이터에 대한 복제본 최적화 알고리즘을 활용하여 데이터 로컬리티를 향상시키고 작업 수행 시간을 감소시켰다. 성능평가 결과, 제안하는 기법은 기존 기법에 비해 접근 빈도의 부하가 평균 약 8% 감소하는 것을 확인하였으며, 전체 질의 처리 시간이 평균 약 55% 감소하는 것을 확인하였다. 향후 연구로는 실제 하둡 환경에서의 다양한 성능 평가를 통해 제안하는 기법의 우수성을 입증하는 것이다.

참고 문헌

- [1] J. Dittrich and J. Quiané-Ruiz, "Efficient Big Data Processing in Hadoop MapReduce," Proc. of the VLDB Endowment, Vol.5, No.12, pp.2014-2015, 2012.
- [2] J. Cohen, J. Dolan, M. Dunlap, J. Hellerstein, and C. Welton, "MAD Skills: New Analysis Practices for Big Data," Proc. of the VLDB Endowment, Vol.2, No.2, pp.1481-1492, 2009.
- [3] <http://hadoop.apache.org>.
- [4] K. Shvachko, H. Huang, S. Radia, and R. Chansler, "The Hadoop Distributed File System," Proc. of the IEEE Symposium on Massive Storage Systems, pp.1-10, 2010.
- [5] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Communication of the ACM, Vol.81, No.1, pp.107-113, 2008.
- [6] F. N. Afrati and J. D. Ullman, "Optimizing Joins in a Map-reduce Environment," Proc. of the International Conference on Extending Database Technology(EDBT '10), pp.99-110, 2010.
- [7] I. Hwang, K. Jung, K. Im, and J. Lee, "Improving the Map/Reduce Model through Data Distribution and Task Progress Scheduling," Journal of the Korea Contents Association, Vol.10, No.10, pp.78-85, 2010.
- [8] H.-C. Yang, A. Dasdan, R.-L. Hsiao, and D. S. Parker, "Map-Reduce-Merge: Simplified Relational Data Processing on Large Clusters," Proc. of the ACM SIGMOD International Conference on Management of Data, pp.1029-1040, 2007.
- [9] S. Ghemawat, H. Gobioff, and S. Leung. "The Google File System," Proc. of ACM Symposium on Operating Systems Principles, pp.29-43, 2003.
- [10] H. Zhao, S. Yang, Z. Chen, S. Jin, H. Yin, and

L. Li, "MapReduce Model-Based Optimization of Range Queries," Proc. of the International Conference on Fuzzy Systems and Knowledge Discovery(FSKD '12), pp.2487-2492, 2012.

저 자 소 개

손 인 국(Ingook Son)

정회원



- 2012년 2월 : 충북대학교 정보통신공학과 공학사
- 2012년 3월 ~ 현재 : 충북대학교 정보통신공학과 석사과정

<관심분야> : 모바일 네트워크, 무선 센서 네트워크, 위치 기반 서비스, 빅 데이터, 클라우드 컴퓨팅 등

류 은 경(Eunkyung Ryu)

정회원



- 2012년 2월 : 충북대학교 정보통신공학과(공학사)
- 2012년 3월 ~ 현재 : 충북대학교 정보통신공학과 석사과정

<관심분야> : 센서 네트워크, 빅데이터, 맵-리듀스, 데이터베이스 시스템 등

박 준 호(Junho Park)

정회원



- 2008년 2월 : 충북대학교 정보통신공학과(공학사)
- 2010년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2010년 3월 ~ 현재 : 충북대학교 정보통신공학과 박사과정

<관심분야> : 무선 센서 네트워크, 데이터베이스 시스템, RFID, 차세대 웹, LMS/LCMS, 바이오인포매틱스 등

복 경 수(Kyoungsoo Bok)

중신회원



- 1998년 2월 : 충북대학교 수학과(이학사)
- 2000년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2005년 2월 : 충북대학교 정보통신공학과(공학박사)

- 2005년 3월 ~ 2008년 2월 : 한국과학기술원 전산학과 Postdoc
 - 2008년 3월 ~ 2011년 2월 : (주)가인정보기술 연구소
 - 2011년 3월 ~ 현재 : 충북대학교 정보통신공학과 초빙부교수
- <관심분야> : 데이터베이스 시스템, 자료저장 시스템, 위치기반서비스, 모바일 P2P 네트워크, 센서네트워크 및 RFID 등

유 재 수(Jaesoo Yoo)

중신회원



- 1989년 2월 : 전북대학교 컴퓨터공학과(공학사)
- 1991년 2월 : KAIST 전산학과(공학석사)
- 1995년 2월 : KAIST 전산학과(공학박사)

- 1995년 3월 ~ 1996년 8월 : 목포대학교 전산통계학과 (전임강사)
 - 1996년 8월 ~ 현재 : 충북대학교 정보통신공학부 및 컴퓨터정보통신연구소 교수
 - 2009년 3월 ~ 2010년 2월 : 캘리포니아주립대학교 방문교수
- <관심분야> : 데이터베이스시스템, 빅데이터, 센서네트워크 및 RFID, 소셜네트워크서비스, 분산객체컴퓨팅, 바이오인포매틱스 등