

# 비용 효율적 맵리듀스 처리를 위한 클러스터 규모 설정

류우석\*

Scaling of Hadoop Cluster for Cost-Effective Processing of MapReduce Applications

Woo-Seok Ryu\*

요약

본 논문에서는 하둡 플랫폼에서 비용 효율적 빅데이터 분석을 수행하기 위한 클러스터 규모의 설정 방안을 연구한다. 의료기관의 경우 진료기록의 병원 외부 저장에 따라 클라우드 기반 빅데이터 분석 요구가 증가하고 있다. 본 논문에서는 대중적으로 많이 사용되고 있는 클라우드 서비스인 아마존 EMR 프레임워크를 분석하고, 비용 효율적으로 하둡을 운용하기 위해 클러스터의 규모를 산정하기 위한 모델을 제시한다. 그리고, 다양한 조건에서의 실험을 통해 맵리듀스의 실행에 영향을 미치는 요인을 분석한다. 이를 통해 비용 대비 처리시간이 가장 효율적인 클러스터를 설정함으로써 빅데이터 분석시 효율성을 증대시킬 수 있다.

ABSTRACT

This paper studies a method for estimating the scale of a Hadoop cluster to process big data as a cost-effective manner. In the case of medical institutions, demands for cloud-based big data analysis are increasing as medical records can be stored outside the hospital. This paper first analyze the Amazon EMR framework, which is one of the popular cloud-based big data framework. Then, this paper presents a efficiency model for scaling the Hadoop cluster to execute a Mapreduce application more cost-effectively. This paper also analyzes the factors that influence the execution of the Mapreduce application by performing several experiments under various conditions. The cost efficiency of the analysis of the big data can be increased by setting the scale of cluster with the most efficient processing time compared to the operational cost.

키워드

Hadoop, Cluster, Efficiency, Mapreduce, Cloud  
하둡, 클러스터, 효율성, 맵리듀스, 클라우드

## 1. 서론

빅데이터를 처리하기 위한 시스템 중 가장 널리 알려져 있는 하둡(Hadoop)은 대량 생성된 빅데이터를 수백 또는 수천 개 이상의 분산 노드에 저장하고 맵

리듀스 프레임워크(MapReduce Framework)를 이용하여 분산 저장된 데이터를 병렬 처리할 수 있는 오픈 소스 기반 플랫폼이다. 하둡은 급속히 증가하는 데이터를 효과적으로 처리하기 위해 그 활용 범위를 넓혀가고 있으며[1], 의료 분야에서도 유전체 분석, 정밀의

\* 부산가톨릭대학교 병원경영학과 (wsryu@cup.ac.kr) • Received : Oct. 22, 2019, Revised : Dec. 19, 2019, Accepted : Feb. 15, 2020  
\* 교신저자 : 부산가톨릭대학교 병원경영학과 • Corresponding Author : Woo-Seok Ryu  
• 접수일 : 2019. 10. 22 Dept. of Health Care Management, Catholic University of Pusan,  
• 수정완료일 : 2019. 12. 19 Email : wsryu@cup.ac.kr  
• 게재확정일 : 2020. 02. 15

료, 진료 빅데이터 등 널리 활용되고 있다[2][3].

국내 대부분을 차지하는 2차병원 이하 중소병원의 경우 진료데이터를 활용한 빅데이터 분석의 필요성은 꾸준히 제기되고 있음에도 불구하고 비용 등 다양한 문제로 인해 빅데이터 분석 플랫폼을 직접 구축하기에는 어려움이 있다. 2016년 8월 의료법 시행규칙이 개정되고 전자의무기록의 관리·보존에 필요한 시설과 장비에 관한 기준이 시행됨에 따라 전자의무기록이 의료기관 내부는 물론 클라우드[4][5] 등 병원 외부 장소에서도 보관이 가능하게 되었다. 이에 중소병원에서도 빅데이터 분석 플랫폼을 자체적으로 구축하는 대신 클라우드 환경에서 저렴한 비용으로 빅데이터 분석이 가능하게 되었다[6].

국내외 주요 클라우드 플랫폼은 네이버비즈니스플랫폼의 네이버 클라우드(Naver Cloud), 아마존 웹서비스(AWS), 마이크로소프트 매저(Microsoft Azure), 구글 클라우드(Google Cloud) 등이 있다[7][8]. 그중 전세계적으로 가장 많이 사용되고 있는 클라우드 플랫폼인 아마존 웹서비스의 경우 아마존 EMR 프레임워크를 통해 HaaS(Hadoop-as-a-Service) 형태의 빅데이터 분석 프레임워크를 제공하고 있으므로 중소병원에서 별도의 플랫폼 구축 없이 편리하게 빅데이터 분석을 수행할 수 있는 장점이 있다[9].

클라우드 서비스의 경우 사용 비용은 사용하는 가상 노드의 성능 및 개수, 그리고 사용 시간에 따라 책정되는 특징이 있다. 클라우드 서비스를 비용 효율적으로 이용하기 위해서는 사용할 컴퓨팅 자원의 수와 사용 시간을 줄이는 것이 필요하다. 이를 하둡 플랫폼에 적용하면 주어진 맵리듀스 어플리케이션을 비용 효율적으로 수행하기 위한 최적의 하둡 클러스터의 규모를 산정하는 것으로 정의내릴 수 있다. 본 논문에서는 선행 연구[10]를 확장하여 맵리듀스 프레임워크의 비용 효율성 모델을 제시하고, 실험을 통해 성능에 영향을 미치는 요인을 분석하고자 한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서 관련 연구로서 아마존 EMR 프레임워크를 분석하고, 3장에서는 하둡에서 맵리듀스 실행을 위한 효율성 모델을 제시한다. 4장에서는 하둡 플랫폼에서의 클러스터 규모에 따른 맵리듀스 실행 성능을 비교 분석하고 5장에서 결론 및 향후 연구를 기술한다.

## II. 아마존 EMR 아키텍처

아마존 EMR(Elastic MapReduce)은 아마존 클라우드 환경에서 대량의 데이터를 비용 효율적으로 처리하기 위한 하둡 프레임워크이다[9]. 아마존 EMR에서는 하둡 이외에도 아파치 스파크, HBase, Hive 등의 다양한 오픈소스 분산 프레임워크를 지원함으로써 오픈소스 기반 빅데이터 분석을 클라우드 환경에서 그대로 실현할 수 있는 특징이 있다.

아마존 EMR에서 맵리듀스를 처리하는 방법은 크게 두가지로 구분된다. 첫 번째는 아마존 EC2(Elastic Compute Cloud 2)를 활용하는 방법이다. EC2는 크기조정이 가능한 분산 컴퓨팅 리소스이며 EC2 클러스터에 다수의 EC2 인스턴스를 할당받은 후 HDFS(: Hadoop Distributed File System) 및 맵리듀스 프레임워크를 설치하여 하둡을 운용할 수 있다. 다만 이 방법은 EC2가 운용되는 동안 지속적으로 과금이 발생하는 특징이 있으며 클러스터를 종료하는 경우 HDFS에 저장된 데이터가 삭제되는 특징이 있다.

두 번째 방법은 데이터의 스토리지로 HDFS 대신 EMRFS(: Elastic Map Reduce File System)를 이용하는 방법이다. EMRFS는 아파치에서 제공하는 확장 가능한 저렴한 클라우드 기반 데이터 저장소인 아마존 S3를 램핑하는 특징이 있다. 아마존 S3는 내구성, 확장성이 뛰어나면서도 상대적으로 요금이 저렴한 스토리지 서비스이다. 즉 데이터는 S3에 장기 저장 유지하면서 분석이 필요할 때 선택적으로 아마존 EMR 클러스터를 구동시키고 분석이 종료될 때 바로 클러스터를 종료할 수 있다. 중소병원과 같이 대량의 데이터를 지속적으로 저장하되, 필요에 따라 간헐적으로 데이터 분석을 수행해야 하는 경우에는 EC2 노드들에 HDFS를 장기간 유지하는 것보다 S3에 데이터를 저장하고 데이터 분석이 필요할 때에만 아마존 EC2 노드로 클러스터를 구성하여 빠른 시간 내에 분석을 종료하는 것이 보다 비용 효율적이다.

아마존 EMR의 요금 정책은 노드당 사용시간에 따른 초당 요금으로 계산된다. 즉, 하둡 클러스터의 규모와 사용시간에 비례하여 요금이 책정된다. 예를 들어, 10개의 EC2 노드를 1시간 사용하는 비용과 노드가 2개인 클러스터를 5시간 사용하는 비용이 동일한 특징이 있다.

### III. 맵리듀스 실행의 효율성 모델

본 연구에서는 임의의 데이터셋에 대해 하둡에서 맵리듀스를 실행시킬 때 가장 비용이 적게 소요되는 클러스터의 규모를 산출하기 위한 효율성 모델을 제시하고자 한다. 이때 실행 비용은 아마존 EMR의 과금 모델과 동일하게 클러스터에 포함된 노드의 수와 총 실행시간의 곱으로 계산하고자 한다.

하나의 맵 태스크와 리듀스 태스크의 실행시간을 각각  $t_m$ ,  $t_r$ 이라고 가정하면 M개의 맵 태스크와 R개의 리듀스 태스크를 포함하는 하나의 잡을 N개의 노드에서 실행시키는 시간  $T_N$ 는 간략히 식(1)과 같이 표현할 수 있다.

$$T_N = \lceil M/N \rceil \times t_m + \lceil R/N \rceil \times t_r \quad (1)$$

단, 위 식은 수식의 간략화를 위해 몇가지 가정을 설정하였다. 위 식은 맵 태스크 결과의 서플 및 정렬로 인해 발생하는 네트워크 부하는 고려하지 않았으며, 각 노드별로 동시에 실행 가능한 태스크는 1개로 가정하였다. 그리고, 모든 노드의 맵 태스크가 모두 종료된 후 리듀스 태스크가 시작된 것을 가정하고 있으나 실제로는 맵 태스크의 일부 종료 후 리듀스 태스크가 시작될 수 있으므로 다소 차이가 있다.

맵리듀스 실행시 주로 병렬로 처리되는 태스크는 맵 태스크이며 위 식에서 M 값은 입력 데이터의 크기에 따라 스플릿된 블록 개수이며, 이때 블록당 하나씩 생성된 맵 태스크가 자동으로 생성된다. 반면에 R 값은 사용자가 지정하는 값으로서 맵 태스크의 결과 임시 파일을 R 값에 맞게 파티셔닝한 후 리듀스 태스크의 입력으로 전달된다. 만일 R 값을 1로 지정한다면 식(1)은 식(2)와 같이 간략화 될 수 있다.

$$T_N = \lceil M/N \rceil \times t_m + t_r \quad (2)$$

컴퓨팅 노드 1개에 대해 1초당 사용 비용을  $c$ 라고 정의하고  $t$ 초당 사용 비용을  $C(t)$ 라고 정의하면  $C(t)$ 는  $c \times t$ 를 만족한다. N개의 노드로 구성된 클러스터를 총  $t$ 초간 사용한 클러스터의 총 실행 비용  $C_N$ 은 위 가정에 따라 식(3)과 같이 산출된다.

$$\begin{aligned} C_N &= N \times C(T_N) \\ &= C(N \times T_N) \\ &= C(N \times \lceil M/N \rceil \times t_m + N \times t_r) \end{aligned} \quad (3)$$

클러스터의 노드 개수가 증가할수록 N이 커지게 되어 병렬 수행이 되므로  $T_N$ 은 감소하나 클러스터의 규모에 따른 실행 비용  $C_N$ 은 N이 커질수록 증가하는 특성이 있다. 잡의 실행시간  $T_N$ 은 N이 1일 때 최대가 되며 N이 M이상이 되면 최소가 된다. 반면에 총 실행비용  $C_N$ 은 N이 1일 때 최소가 되며 N이 커질수록 증가하게 된다. 이때, 클러스터 규모의 효율성은 비용 증가 대비 시간 감소의 이득이 가장 클 때 가장 효율적이라고 할 수 있다. N개의 노드에서 실행시킬 때의 실행 효율  $E_N$ 은 식(4)와 같이 정의할 수 있다.

$$\begin{aligned} E_N &= (T_1/T_N) / (C_N/C_1) \\ &= (T_1 \times C_1) / (T_N \times C_N) \\ &= (T_1 \times C(T_1)) / (T_N \times N \times C(T_N)) \\ &= (T_1/T_N)^2 / N \end{aligned} \quad (4)$$

이를 식(2)와 결합하면 아래 식(5)와 같이 표현될 수 있다.

$$E_N = ((M \times t_m + t_r) / (\lceil M/N \rceil \times t_m + t_r))^2 / N \quad (5)$$

식(5)에 따르면 이론적으로 맵리듀스 실행에서 효율이 높을 때는 M이 N의 배수가 되도록 N을 설정하는 것이다. 즉, M이 8일 때 N을 2, 4, 8 등으로 설정할 때 효율을 높일 수 있다. 하지만, 실제 맵리듀스에서는 스케줄러에 따라 균등하게 분배되지 않을 수 있으며, 특히 EMRFS이 아니라 HDFS를 사용하는 경우라면 맵 태스크의 생성이 데이터의 지역성(data locality)에 영향을 받아서 분산 노드에 맵 태스크가 골고루 생성되지 않거나 데이터 블록을 네트워크로 전송하는 비용이 추가되는 특징이 있다.

## IV. 실 험

### 4.1 실험환경 설정

이 장에서는 클러스터에서 노드의 개수가 변경될

때 발생하는 맵리듀스의 실행 성능 변화를 분석하고자 한다. 실험 환경으로 2-코어 펜티엄 프로세서와 4GB 메모리가 장착된 10개의 PC를 1기가비트 스위칭 허브로 연결한 클러스터를 이용하였다. 각 시스템의 소프트웨어로는 우분투 14와 하둡 2.7.4를 이용하였으며, 스케줄러, 데이터 블록 크기와 같은 하둡 설정은 배포판의 기본 설정을 이용하였다.

클러스터 규모에 따른 맵리듀스의 성능 분석을 위해 다양한 개수의 슬레이브 노드로 구성된 하둡 클러스터를 구성하였다. 총 10대의 노드 중 하나의 노드는 마스터 노드로 설정하였으며, 슬레이브 노드는 1에서 9대로 총 9가지 구성의 클러스터를 설정하였다.

클러스터 규모에 따른 성능 비교를 위한 하둡 잡(Job)은 하둡 배포판에 포함되어 있는 wordcount 예제 프로그램을 이용하였으며, 성능 측정은 yarn 명령어로 wordcount 예제 프로그램을 실행할 때의 전체 시간을 시스템 시간을 통해 측정하였고 동일한 조건에서 5회 실행 후 평균 실행 시간을 산출하였다. 이때 맵 태스크의 개수는 데이터의 크기를 블록 사이즈(128MB)로 나눈 블록의 개수로 자동 설정되며, 리듀스 태스크의 개수는 기본값인 1로 설정하였다.

본 실험에서 데이터는 두 가지 종류로서 하나는 리듀스 태스크의 입력을 최소화하기 위해 300개의 단어만을 이용해서 용량을 늘린 텍스트 데이터이다. 표 1의 데이터셋1은 용량은 256MB, 512MB, 1GB, 2GB, 4GB 다섯 가지로 구성하였으며 256MB의 경우 맵 태스크 수행 후 결과 데이터는 8,916바이트에 불과하여 리듀스 태스크의 부하가 매우 적은 특징이 있다. 데이터셋2는 데이터셋1과 마찬가지로 다섯 가지 사이즈의 데이터로 구성되어 있으나 리듀스 태스크의 부하를 늘리기 위해 임의 단어로 생성한 텍스트 데이터이다. 데이터셋2는 256MB의 경우 맵 태스크 수행 후 결과 데이터가 오히려 원본 데이터보다 큰 용량을 가지고 있으며, 리듀스 태스크의 입력 레코드 수도 2.5M 가량 될 만큼 리듀스 태스크의 부하가 많이 발생하는 특징이 있다.

표 1. 실험 데이터셋  
Table 1. Dataset for the experiment

Data-set	Data size	Map output materialized bytes	Reduce input records	Reduce output records
1	256MB	8,916	608	304
	512MB	17,832	1,216	304
	1GB	35,664	2,432	304
	2GB	71,328	4,864	304
	4GB	147,108	10,032	304
2	256MB	294.8M	2.5M	2.5M
	512MB	589.8M	5.0M	2.5M
	1GB	1,179.5M	10.0M	2.5M
	2GB	2,358.8M	20.0M	2.5M
	4GB	4717.6M	40.1M	2.5M

#### 4.2 실험 결과

그림 1은 데이터셋1을 사용하여 프로그램을 실행하였을 때의 슬레이브 노드의 수에 따른 실행 시간을 도시한 결과이다. 데이터셋1의 경우 리듀스 태스크의 부하가 거의 없음에 따라 슬레이브 노드의 개수가 늘어남에 따라 실행 성능이 반비례하는 것을 확인할 수 있다. 데이터의 크기가 256MB인 경우에는 데이터 블록이 2개이며 맵 태스크가 두 개가 생성되는데, 슬레이브 노드가 1개일 때와 9개일 때가 성능 차이가 거의 발생하지 않는다. 하나의 노드가 2개의 코어를 가지고 있으며 이에 두 개의 맵 태스크를 병렬 처리하게 되므로 1개의 슬레이브 노드만으로도 충분한 성능을 낼 수 있다는 것을 의미한다.

데이터의 크기가 1G인 경우에는 데이터 블록이 8개이며 총 8개의 맵 태스크가 생성된다. 이때에는 이론상으로는 4개의 노드일 때 최상의 맵 태스크 실행 성능을 보여야 하나 실험 결과로는 슬레이브 노드가 1개에서 2개로 늘어날 때 성능이 두배로 증가한 후 노드를 계속 늘려갈 때의 성능 향상은 미미하였다. 그림 2에 도시된 바와 같이 슬레이브 노드의 개수가 맵 태스크 개수의 1/4이 될 때 가장 비용이 낮은 것으로 확인되었으며, 데이터의 크기가 2GB, 4GB인 경우 슬

레이브 노드의 개수는 각각 4개, 8개일 때 가장 비용이 낮은 것을 확인할 수 있다. 이론적 모델과 실제의 차이는 데이터 블록의 복제 계수에 따른 데이터 지역성(data locality)[11][12]으로 인해 맵 태스크가 노드에 균등 분산되지 않아서 발생하는 것으로 판단된다.

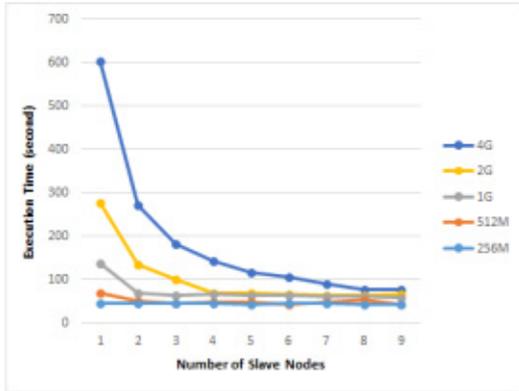


그림 1. 데이터셋1의 맵 리듀스 실행 성능  
Fig. 1 Performance of Mapreduce using dataset1

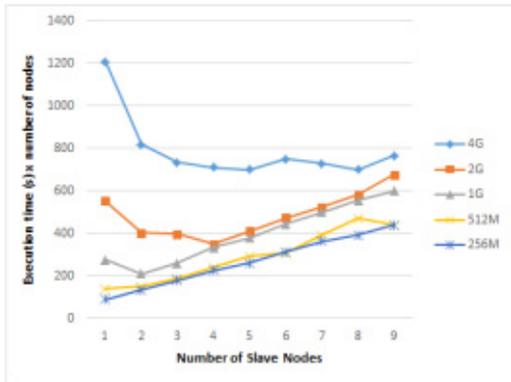


그림 2. 데이터셋1의 맵 리듀스 실행 비용  
Fig. 2 Execution Cost of Mapreduce using dataset1

그림 3은 데이터셋2를 이용하여 동일한 실험을 수행한 결과이다. 데이터의 크기가 256MB, 512MB, 1GB인 경우에는 데이터셋1과 비교하여 큰 차이점이 발견되지 않으나 슬레이브 노드가 6개 이상으로 늘어날 때에 오히려 실행 성능이 떨어지는 것을 확인할 수 있다. 이는 노드 개수가 늘어남으로 인해 네트워크 전송 부하가 더욱 커진 것으로 판단된다. 2GB 데이터의 경우 데이터셋1과 비교하여 볼 때 실행 시간이 보

다 점진적으로 감소하여 노드의 개수가 6개일 때 성능이 가장 높은 것을 확인할 수 있다. 하지만 그림 4에 따르면 비용은 노드의 개수가 2개 이상일 때부터 지속적으로 증가하게 되므로, 이때 가장 효율적인 클러스터의 규모는 슬레이브 노드의 수가 2일 때로 한정할 수 있다. 데이터의 크기가 4GB인 경우 병렬 수행에 따른 실행시간의 이익이 매우 한정적이며 이에 비용은 지속적으로 상승하는 것을 확인할 수 있다.

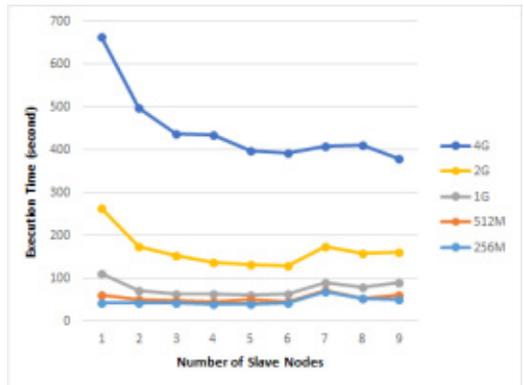


그림 3. 데이터셋2의 맵 리듀스 실행 성능  
Fig. 3 Performance of Mapreduce using dataset2

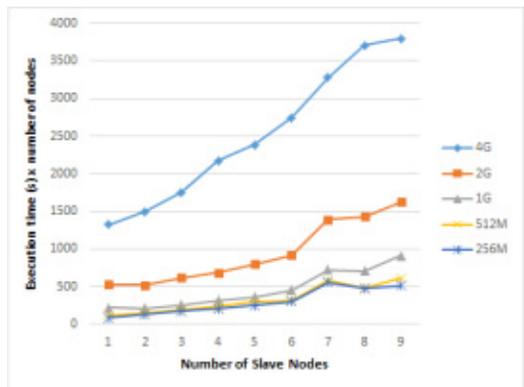


그림 4. 데이터셋2의 맵 리듀스 실행 비용  
Fig. 4 Execution Cost of Mapreduce using dataset2

실험 결과를 종합하여 볼 때 클러스터 규모에 따른 맵리듀스의 실행 시간 및 비용은 리듀스 태스크에 매우 큰 영향을 받음에 따라 효율적인 클러스터의 규모를 산정하기 위해서는 맵 단계에서 병렬화를 최대화하기 위한 맵태스크 위주의 맵리듀스 프로그램 개발

이 선행될 필요가 있다. 리듀스 단계의 부하가 적은 경우 본 실험에서는 맵 태스크 개수의 1/4만큼 슬레이브 노드를 구성하였을 때 비용을 최소화할 수 있었음을 확인할 수 있었으며, 아마존 EMR과 S3이 조합된 클라우드 환경에서는 데이터 지역성이 고려되지 않으므로 3장에서 제시한 효율성 모델이 적합할 것으로 판단된다.

## V. 결 론

본 논문에서는 하둡에서 맵리듀스를 실행시킬 때의 수행 비용을 최소화하기 위한 클러스터 효율성 모델을 제시하고 실험을 통해 이를 검증하였다. 최적의 클러스터 규모는 특히 클라우드 환경에서 중요한 문제로 사용될 노드의 수가 사용 비용에 직결되는 특징이 있다. 최적의 클러스터 규모를 산정하기 위해서는 먼저 맵 태스크 단계에서의 병렬 수행을 최적화하여 상대적으로 병렬 수행이 어려운 리듀스 태스크 단계의 부하를 줄여야 하는 점을 실험을 통해 입증하였으며, 병렬 수행이 충분한 경우 제안한 모델을 통해 맵리듀스의 실행시간과 비용을 동시에 줄일 수 있는 방안을 제시하였다. 이를 통해, 보다 비용 효율적으로 빅데이터 분석이 가능함에 따라 중소병원과 같은 작은 규모의 기관에서도 클라우드 환경을 기반으로 한 빅데이터 분석이 확산될 수 있을 것이다. 추후 연구로는 본 실험을 아마존 EMR에 실제 적용하여 성능과 비용을 평가함으로써 클라우드 환경에서의 보다 정밀한 비용 모델을 산출하는 것이다.

### 감사의 글

이 논문은 2017년도 부산가톨릭대학교 교내연구비에 의하여 연구되었음

## References

- [1] Y. Ding and K. Kim, "A Customized Tourism System Using Log Data on Hadoop," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 13, no. 2, Apr. 2018, pp. 397-404.
- [2] E. Nazari, M. H. Shahriari, and H. Tabesh, "Big Data Analysis in Healthcare: Apache Hadoop, Apache spark and Apache Flink," *Frontiers in Health Informatics*, vol. 8, no. 1, 2019, pp. 92-101.
- [3] J. Choi, "Utilization value of medical Big Data created in operation of medical information system," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 10, no. 12, Dec. 2015, pp. 1403-1410.
- [4] Y. Ahn and H. Cho, "Hospital System Model for Personalized Medical Service," *J. of the Korea Convergence Society*, vol. 8, no. 12, Dec. 2017, pp. 77-84.
- [5] S. Kim and D. Kim, "The Design and Implementation of the Fire Spot Display System Using s Smart Device," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 13, no. 6, Dec. 2018, pp. 1287-1292.
- [6] M. Lee, "Considerations for the Migration of Electronic Medical Records to Cloud Based Storage," *J. of Korean Library and Information Science*, vol. 47, no. 1, Mar. 2016, pp. 149-173.
- [7] M. Copeland, J. Soh, A. Puca, M. Manning, and D. Gollob, *Microsoft Azure*. Berkeley: Apress, 2015.
- [8] T. Gunarathne, T. Wu, J. Qiu, and G. Fox, "MapReduce in the Clouds for Science," In *Proc. the IEEE Cloud Computing Technology and Science*, Indianapolis, USA, 2010, pp. 565-572.
- [9] S. Mathew, "Overview of Amazon Web Services," *Amazon Whitepapers*, Nov. 2014.
- [10] W. Ryu, "Cost-Effective MapReduce Processing in the Cloud," In *Proc. the Conf. on Korea Information and Communication Engineering*, vol. 22, no. 2, Oct. 2018, pp. 114-115.
- [11] A. Sharma and G. Singh, "A Review on Data locality in Hadoop MapReduce," In *2018 Fifth Int. Conf. on Parallel, Distributed and Grid Computing*, Solan Himachal Pradesh, India,

Dec. 2018, pp. 723-728.

- [12] S. Kim, Y. Kim, and W. Kim, "The Design of Method for Efficient Processing of Small Files in the Distributed System based on Hadoop Framework," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 10, no. 10, Oct. 2015, pp. 1115-1122.

## 저자 소개



### 류우석(Woo-Seok Ryu)

1997년 부산대학교 컴퓨터공학과  
졸업 (공학사)

1999년 부산대학교 대학원 컴퓨터  
공학과 졸업(공학석사)

2012년 부산대학교 대학원 컴퓨터공학과 졸업(공학  
박사)

2013년 ~현재 부산가톨릭대학교 병원경영학과 부교수

※ 관심분야 : 의료정보, 빅데이터, 하둡 플랫폼

