대규모 정형 데이터를 위한 맵리듀스 기반 고속 가명 처리 시스템 개발*

유사라[1], 장은조[1], 이기용[2]
[1]숙명여자대학교 컴퓨터과학과
[2]숙명여자대학교 소프트웨어학부

{rrrr4ra, wkddmswh99, kiyonglee}@sookmyung.ac.kr

Development of a High-Speed Pseudonym Data Processing System Based on MapReduce for Large-Scale Structured Data

Sara Yu[1], Eunjo Jang[1], Ki Yong Lee[2]
[1]Department of Computer Science, Sookmyung Women's University
[2]Division of Computer Science, Sookmyung Women's University

요 약

정보화 사회의 개인 정보의 경제적 가치가 증대됨에 따라 대규모 데이터의 수집, 분석 및 활용이 급증하고 있다. 이러한 데이터는 종종 민감한 개인 정보를 포함하고 있어, 적절한 가명 처리가 필수적이다. 그러나 기존의 가명 처리 시스템은 처리 시간 지연, 컴퓨팅 자원의 과다한 요구, 확장성 부족 등의 문제를 안고 있다. 따라서 본 논문은 MapReduce를 기반으로 한 효율적인 대규모 가명 처리 시스템을 제안한다. 제안하는 시스템은 통계 정보를 획득하는 작업 Job1과 실질적으로 가명 처리를 수행하는 작업 Job2로 구성하여 8가지 가명 처리 기능을 구현한다. 본 연구에서는 16대의 노드 클러스터 환경에서 대규모 정형 데이터(500, 1000, 1500, 2000GB) 가명 처리를 Job1, Job2 각각의 수행 시간을 측정하여 효율성과 실용성을 입증한다. 성능 측정 결과 2TB 기준으로 약 60분 만에 수행됨을 확인하였고, 이는 작은 작업 단위로 분할하여 병렬 처리함으로써 빠른 연산 속도를 달성하고, 다양한 가명 처리 기능을 통합하여 개인정보 보호를 강화하여 대규모 데이터의 활용성 증대를 기대할 수 있다.

1. 서 론

다양한 산업 분야에서 대규모 데이터의 수집, 분석 및 활용이 급증하면서, 정보의 경제적 가치가 증대됨에 따라 개인 정보의 수집 범위가 크게 증가하였다. 개인 정보란 살아있는 개인에 관한 정보로 성명, 주민등록번호, 주소 등 개인을 알아볼 수 있는 정보를 의미한다. 최근 정보의 민감성 및 개인정보보호법[1]의 강화로 인해 개인 정보의 안전한 처리 방안 연구가 중요해지고 있다. 가명 정보란 추가 정보 없이는 특정 개인을 식별할 수 없는 정보를 의미한다. 그림 1은 개인을 특정할 수 있는 정보를 가명 처리 한 예시이다. 이는 개인 정보 유출 피해를 예방하며 데이터의 안전한 활용을 보장한다. 그러나 데이터의 양이 지속적으로 증가함에 따라, 단일 컴퓨터로 대용량 개인 정보를 가명 처리를 하기에는 한계가 존재한다. 단일 시스템에서 메모리에 로드하기 조차 어려우며, 무한한 자원을 가정하더라도 높은 수행 시간으로 상당한 지연을 초래한다. 따라서 지속적으로 증가하는 개인 정보의 양은 시스템의 확장성에 대한 요구를 촉진하며, 관리와 보안 분야의 발전을 위해 대용량 데이터의 가명 처리 시스템 개발은 필수적인 실정이다.

맵리듀스(MapReduce)는 빅데이터 처리에 널리 사용되는 하둡 기반의 분산 컴퓨팅 프레임워크로, 컴퓨터를 클러스터 환경으로 구성하여 작업을 작은 단위로 분산처리를 수행함으로써 빠른 처리 속도의 장점을 갖는다.

* 이 논문은 2023년 과학기술정보통신부의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2021-0-00634)

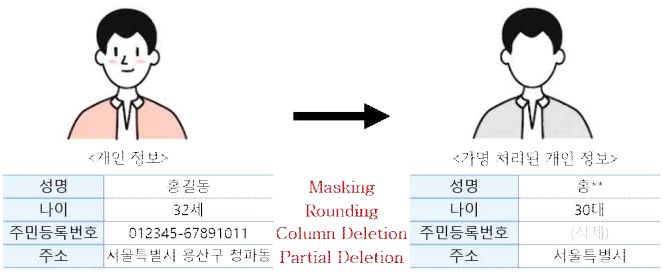

그림 1. 개인 정보와 가명 처리된 개인 정보

따라서 본 논문에서는 대용량 데이터의 가명 처리를 위해 맵리듀스(MapReduce)기반 고속 가명 처리 방법을 제안하고 그 시스템을 설계한다. 제안하는 시스템은 마스킹, 부분 삭제, 열 삭제, 라운딩, 상하단 코딩, 부분 총계, 난수 생성, 암호화 총 8개의 대표적인 가명 처리 기능을 포함한다. 본 연구에서는 직접 구축한 16대의 하둡 클러스터 환경에서 테라바이트(TB)급 가상 개인 정보 데이터에 대한 가명 처리 수행 시간을 측정하여 성능을 평가하였다.

본 논문은 다음과 같이 구성된다. 2장에서는 관련된 사전 지식을 살펴보고, 3장에서는 제안하는 시스템의 구조와 기능, 그리고 동작 원리에 대해 상세히 설명한다. 4장에서는 실제 시스템 성능의 평가 결과를 제시하여 제안된 시스템의 유효성을 입증하고, 5장에서 결론을 맺는다.

2. 하둡

하둡(Hadoop)은 대규모 데이터 세트를 분산 처리하기 위한 프레임워크로 [2], 여러 대의 컴퓨터 클러스터에서 작동한다. 이 시스템은 단일 서버에서부터 수천 대의 서버로 구성된 클러스터까지 확장될 수 있으며, 하둡 분산 파일 시스템(HDFS)과 맵리듀스(MapReduce) 두 가지 주요 구성 요소로 이루어져 있다.

2.1 하둡 분산형 파일 시스템(HDFS)

HDFS[3]는 마스터 노드(Master Node)와 데이터 노드(Data Node) 두 가지 유형의 노드로 구성된다.

- 마스터 노드: 각 데이터 블록이 어떤 데이터 노드에 저장되어 있는지에 대한 정보와 파일 구조, 데이터 블록의 위치 정보가 포함된 메타 데이터를 관리 및 저장한다.
- 데이터 노드: 마스터 노드의 지시에 따라 데이터를 저장, 검색, 삭제, 복제 등의 작업을 처리한다. 파일 시스템의 실제 데이터를 담당하며, 클러스터 내의 다른 데이터 노드와 함께 작업하여 데이터의 신뢰성과 가용성을 보장한다.

2.2 맵리듀스(MapReduce)

맵리듀스[4]는 분산 컴퓨팅 원칙을 기반으로, 큰 대용량 데이터를 작은 조각으로 나누어 여러 노드에서 동시에 처리할 수 있는 프레임워크이다. 이는 맵(Map)과 리듀스(Reduce) 두 가지 주요 작업으로 나뉜다.

- 맵 : 입력 데이터를 여러 하위 조각으로 나누어 각 노드에서 병렬 처리된다. 맵 단계에서는 중간 결과로 키(key)와 값(value) 쌍을 출력한다.
- 리듀스 : 중간 키-값 쌍을 처리하여 다시 합쳐 결과를 생성한다. 리듀스 단계는 최종 결과를 하둡 분산 파일 시스템에 저장한다.

3. 분산 병렬 처리 기반의 가명 처리 시스템

본 논문에서는 대규모 정형 데이터를 위한 맵리듀스 기반 가명 처리 시스템을 제안한다. 표 1은 제안하는 시스템에서 제공하는 8가지 가명 처리 기능 상하단 코딩, 부분 총계, 난수화, 열 삭제, 마스킹, 라운딩, 부분 삭제, 암호화 기능 정의이다.

표 1. 구현한 개인정보 가명 처리 기능

개인 정보 가명 처리 기능 정의		Job1	Job2
상하단 코딩 (Top&bottom coding)	숫자형 열에서 μ ± σ 바깥의 데이터를 μ로 대체 (μ: 평균, σ: 표준편차)	O	O
부분 총계 (Micro-aggregation)	데이터 중 일부 레코드의 지정된 열이 특정 값을 가지는 레코드에 대해 지정된 다른 열의 값을 해당 열의 평균값으로 대체		
난수화 (Randomization)	숫자형 열 : [최솟값, 최댓값] 사이의 난수로 대체, 문자형 열: [최소길이, 최대길이] 사이의 길이를 가지는 임의의 문자열로 대체		
열 삭제 (Column deletion)	지정된 열 전체를 삭제	X	
마스킹 (Masking)	지정된 열에서 특정 부분을 마스킹 문자로 대체 ('*' 등)		
라운딩 (Rounding)	숫자형 열에 대해 반올림, 올림, 또는 버림 수행 (자릿수 지정 가능)		
부분 삭제 (Partial deletion)	지정된 열에서 특정 부분만 부분적으로 삭제 (시작 부분과 끝 부분 지정 가능)		
암호화 (Encryption)	지정된 열을 SHA-256 알고리즘을 사용하여 암호화		

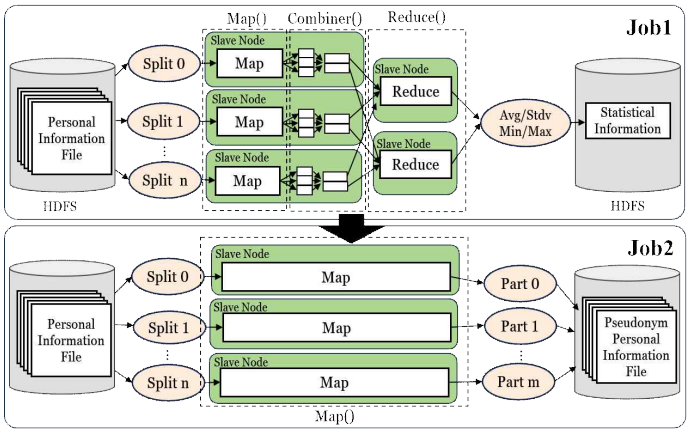

그림 2. 제안하는 시스템의 전반적인 흐름도

그림 2는 제안하는 시스템의 전반적인 흐름도를 나타낸다. 시스템은 열의 통계 정보를 얻는 Job1과 실제 가명 처리를 수행하는 Job2 두 단계로 나눠서 작업을 수행하며, 각 작업은 HDFS에 저장된 데이터를 분할하여 각각의 노드에 전달하여 MapReduce 과정을 수행하고 그 결과를 HDFS에 저장한다. 이때, 분할된 작업을 각각의 노드에서 분산 병렬 처리를 수행한다.

(1) Job1 : 가명화 연산을 수행하기 위해 통계 정보를 획득하는 작업으로 Map, Combiner, Reduce 함수로 구성하며 각 단계의 역할은 다음과 같다.

- Map() : Map 함수는 입력 데이터에서 특정 조건에 맞는 데이터를 필터링하여 추출한다.
- Combiner() : 네트워크 트래픽과 디스크 I/O 감소를 위해 Map 단계의 중간 결과를 각 서버의 로컬에서 우선 집계한 후, Reduce로 전송한다. 이는 Map의 출력을 합계, 제곱합, 개수, 최솟값, 최댓값 등으로 집계하여 전송하므로 Reduce로 전송하는 데이터의 크기와 개수를 줄인다.
- Reduce() : Reduce 함수는 Combiner 함수에서 전달받은 각 서버의 출력값을 통해 열의 평균/표준편차, 최솟값/최댓값을 계산한다.

(2) Job2 : 8가지 가명 처리를 실질적으로 수행하는 작업으로 불필요한 Reduce 단계를 생략한 Map-only 작업으로 수행한다.

- Map() : 레코드 별로 한 줄씩 읽어 가명 처리에 해당하는 열일 경우 연산을 수행한다. Map 함수의 결과값은 최종적으로 가명 처리가 완료된 데이터로 HDFS에 저장된다.

표 2. 제안 시스템 작업별 구분

Job 구분		해당 가명 처리 기능
Job 1	Map	상하단 코딩, 부분 총계 [평균, 표준편차] 난수화 [최솟값, 최대값]
	Combiner	
	Reduce	
Job 2	Map	8가지 기능 수행

표 2는 각 작업별로 해당하는 가명 처리 기능이다. 상하단 코딩, 부분 총계, 난수화는 가명 처리에 필요한 통계 정보를 계산하기 위해 Job1이 필요하고, 그 외 열 삭제, 마스킹, 라운딩, 부분삭제, 암호화 총 5가지 연산은 Job1을 생략하고 바로 Job2를 수행한다.

4. 성능 평가

본 장에서는 제안하는 MapReduce 기반 고속 가명 처리 시스템의 성능 평가를 제시한다.

4.1 실험 환경

성능 검증은 표 3의 시스템 환경으로 직접 구축한 Hadoop 클러스터에서 수행한다.

표 3. 시스템 환경

노드 수	네임 노드 : 1대	
	데이터 노드 : 15대	
노드 성능	CPU: Intel(R) Xeon(R) Silver@ 2.10GHz	
	Memory: 64GB	
	SSD	446.6GB
	HDD	2TB
네트워크 대역폭	1Gbps	
운영체제	Linux	
Hadoop	hadoop-3.3.5	

4.2 실험 데이터

실험 데이터는 ㈜이지서티에서 제공한 주민등록번호, 성별, 혈액형, 주거래은행, 근무년수, 연봉 등으로 구성된 가상 개인정보를 사용하였다. 수치형 8개, 문자형 22개로 총, 30개의 열로 구성된 대규모 정형 데이터로 500GB, 1000GB, 1500GB, 2000GB로 4가지 크기로 실험을 진행하였다. 이는 2000GB 기준 약 60억 행으로 구성된 대규모 정형 데이터이다.

4.3 맵리듀스 성능 최적화 파라미터

하둡은 설정 파라미터에 따라 수행 시간과 속도가 크게 영향을 받는다. 최적 파라미터는 그리드 서치(Grid Search) 방식으로 모니터링과 실험을 거쳐 최적화된 값을 확인했으며, 표 4는 설정한 최적 파라미터이다. 설정은 각 노드에 일관되게 적용하였다.

표 4. Hadoop 설정 파라미터 최적값

매개변수	설명	값
yarn.nodemanager. resource.memory-mb	클러스터의 각 노드마다 컨테이너 운영에 할당 가능한 총 메모리량	40GB
yarn.nodemanager. resource.cpu-vcores	클러스터의 각 노드에서 실행 중인 컨테이너에 할당 가능한 가상 CPU 개수	12
yarn.scheduler. maximum-allocation-mb	클러스터에서 각 컨테이너에 할당 가능한 메모리의 최대값	38GB
yarn.scheduler. minimum-allocation-mb	하나의 컨테이너에 할당할 수 있는 메모리의 최소값	2GB
yarn.app.mapreduce.am.resour ce.mb	ApplicationMaster(AM)이 사용할 수 있는 메모리 양 제한	4GB
dfs.block.size	HDFS에서 파일을 블록으로 나눌 때 각 블록의 크기	128MB
mapreduce. reduce.memory.mb	MapReduce 작업의 Reduce 태스크가 사용할 수 있는 메모리 양	8GB
mapreduce. map.memory.mb	MapReduce 작업의 Map 태스크가 사용할 수 있는 메모리 양	4GB
mapreduce.job.maps	Map 단계에서 실행할 맵 태스크 개수	200
mapreduce.job.reduces	Reduce 단계에서 실행할 리듀스 태스크 개수	100
mapreduce.input. fileinputformat.split.minsize	하나의 입력 파일이 분할 가능한 최소 크기	128MB
mapreduce.input. fileinputformat.split.maxsize	하나의 입력 파일이 분할 가능한 최대 크기	128MB
mapred.compress.map.output	Map 단계 출력의 압축 여부	True

4.4 수행 시간 측정 결과

성능평가는 500, 1000, 1500, 2000GB로 데이터 규모에 따른 3장에서 제안한 Job1과 Job2에 대한 수행 소요 시간을 측정하였다.

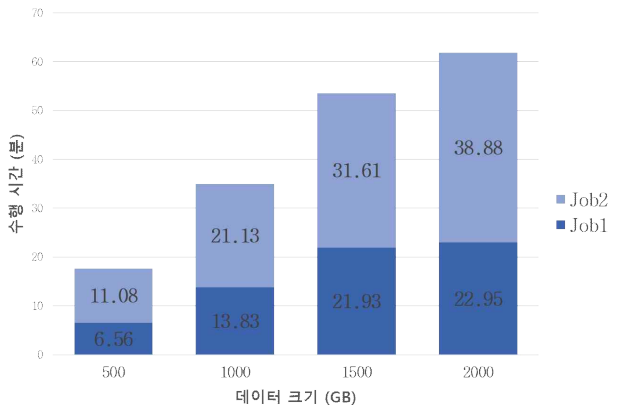

그림 3. 제안하는 시스템의 처리 수행 시간

그림 3은 개인정보 데이터를 제안하는 시스템의 데이터의 크기에 따른 가명 처리 수행 시간을 측정한 결과이다. 표 4에서 제시한 파라미터 최적값을 모든 데이터에 동일하게 적용하여 데이터 크기를 500GB부터 2000GB까지 500GB 간격으로 증량하며 실험을 진행하였다. HDFS에 있는 데이터를 읽어 분할하고 Job1과 Job2를 거쳐 가명 처리를 수행하는데 500GB 17.64분, 1000GB 34.96분, 1500GB 53.54분 2000GB 61.83분이 소요됨을 확인하였다. 이는, 단일 노드 처리 대비 효율적인 처리 수행 시간으로 제안하는 시스템이 실제 산업에서도 대규모 정형 데이터 가명 처리에 실용적으로 적용할 수 있음을 보여준다.

5. 결론

본 논문에서는 대용량 데이터에 대한 맵리듀스 기반의 가명 처리를 제안하였다. 또한, 실험을 통해 제안 방법이 대용량 데이터의 가명 처리 수행 시간을 측정하여 효율적으로 처리함을 보였다. 제안하는 시스템은 정보화 사회에 지속적으로 증가하는 규모의 데이터를 효율적으로 처리하여 개인정보를 보호하고, 최종적으로 데이터 활용에 긍정적인 영향을 미칠 것으로 기대된다.

참고문헌

[1] 개인정보보호위원회. 가명정보 처리 가이드라인 2022.04

[2] J. Dean and S. Ghemawat., “MapReduce: simplified data processing on large clusters,” Communications of the ACM, 51(1), 107-113, 2008.

[3] D. Borthakur., “HDFS architecture guide,”. Hadoop apache project, 53(1-13), 2, 2008.

[4] J. H. Kwak, J. W. Yoon, Y. H. Jung, J. g. Hahm, D. I. Park, "Large-scale Data Analysis based on Hadoop for Astroinformatics", Journal of KIISE : Computing Practices and Letters, vol. 17, no. 11, pp.587-591, Nov. 2011.