https://doi.org/10.5392/JKCA.2021.21.12.084

다양성을 지원하는 그래프 데이터베이스 벤치마킹 시스템

Graph Database Benchmarking Systems Supporting Diversity

최도진[*], 백연희[**], 이소민[*], 김윤아[**], 김남영[**], 최재용[*], 이현병[*], 임종태[*], 복경수[***], 송석일[****], 유재수[*]
충북대학교 정보통신공학과[*], 충북대학교 빅데이터학과[**], 원광대학교 SW융합학과[***],
한국교통대학교 컴퓨터공학과[****]

**Do-Jin Choi(mycdj91@cbnu.ac.kr)[*], Yeon-Hee Baek(yh100@cbnu.ac.kr)[**],
So-Min Lee(somin@cbnu.ac.kr)[*], Yun-A Kim(rud5356@naver.com)[**],
Nam-Young Kim(minstrel68@naver.com)[**], Jae-Young Choi(headmeat@naver.com)[*],
Hyeon-Byeong Lee(lhb@cbnu.ac.kr)[*], Jong-Tae Lim(jtlim@cbnu.ac.kr)[*],
Kyoung-Soo Bok(ksbok@wku.ac.kr)[***], Seok-Il Song(sisong@ut.ac.kr)[****],
Jae-Soo Yoo(yjs@cbnu.ac.kr)[*]**

요약

객체 간의 관계를 표현하기 위해 정점과 간선으로 구성된 그래프 데이터를 효율적으로 저장하고 질의 처리하기 위한 그래프 데이터베이스가 개발되었다. 그래프 데이터베이스는 질의 유형이 기존 NoSQL 데이터베이스와 매우 다른 특성을 보이기 때문에 그래프 데이터베이스의 성능을 검증하기 위해서는 그래프 데이터베이스에 알맞은 벤치마킹 도구가 필요하다. 본 논문에서는 그래프 입력과 질의에 대한 다양성을 지원하는 효율적인 그래프 데이터베이스 벤치마킹 시스템을 제안한다. 제안하는 시스템은 그래프 데이터베이스에 대한 벤치마킹을 테스트하기 위해서 OrientDB를 활용한다. 입력 그래프와 질의 그래프의 다양성을 지원하기 위해서 기존 그래프 데이터 생성 도구인 LDBC를 이용한다. 벤치마킹 결과 분석을 통해 제안하는 기법의 타당성 및 실효성을 입증한다. 성능 평가 결과 제안하는 시스템은 사용자 정의 가능한 가상 그래프 데이터가 생성이 가능하며, 생성된 그래프 데이터를 기반으로 벤치마킹이 가능함을 보였다.

■ 중심어 : | 빅데이터 | 데이터베이스 | 벤치마킹 | 워크로드 | NoSQL |

Abstract

Graph databases have been developed to efficiently store and query graph data composed of vertices and edges to express relationships between objects. Since the query types of graph database show very different characteristics from traditional NoSQL databases, benchmarking tools suitable for graph databases to verify the performance of the graph database are needed. In this paper, we propose an efficient graph database benchmarking system that supports diversity in graph inputs and queries. The proposed system utilizes OrientDB to conduct benchmarking for graph databases. In order to support the diversity of input graphs and query graphs, we use LDBC that is an existing graph data generation tool. We demonstrate the feasibility and effectiveness of the proposed scheme through analysis of benchmarking results. As a result of performance evaluation, it has been shown that the proposed system can generate customizable synthetic graph data, and benchmarking can be performed based on the generated graph data.

■ keyword : | Bigdata | Database | Benchmarking | Workload | NoSQL |

* 이 (성과)는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원(No. 2019R1A2C2084257), 중소벤처기업부 '산업전문인력역량강화사업'의 재원으로 한국산학연협회(AURI)의 지원, (2021년 기업연계형연구개발인력양성사업, 과제번호 : S3047889) 농촌진흥청 연구사업 (세부과제번호: PJ01624701)의 지원 및 과학기술정보통신부 및 정보통신기획평가원의 지역지능화혁신인재양성(Grand ICT연구센터) 사업의 연구결과로 수행되었음" (IITP-2021-2020-0-01462)

접수일자 : 2021년 08월 31일 심사완료일 : 2021년 09월 27일
수정일자 : 2021년 09월 27일 교신저자 : 유재수, e-mail : yjs@cbnu.ac.kr

I. 서 론

대량의 빅데이터가 생성되고 활용됨에 따라 수평적 확장성, 고성능, 유연성을 제공하기 위해 전통적인 관계형 데이베이스보다 보다 덜 제한적인 일관성 모델을 제공하는 NoSQL(Not Only SQL)이 활발히 활용되고 있다[1-5]. NoSQL은 데이터 모델에 따라 키-값 스토어(key-value store), 와이드 컬럼 스토어(wide-column store), 문서 스토어(document store), 그래프 스토어(graph store)로 구분된다[1-6]. OrientDB는 다중 모델 NoSQL 데이터베이스로 키-값, 문서, 그래프, 객체 데이터 모델을 단일 데이터베이스에서 모두 지원함으로써 유연한 데이터 모델링을 제공한다[6]. 그래프 데이터는 객체 간의 관계를 기반한 탐색 연산 및 질의가 수행되어야 한다. OrientDB에서는 확장된 SQL(Structured Query Language)과 Gremlin을 통해 그래프 탐색 질의를 수행한다[7].

NoSQL 데이터베이스에 대한 효율성 및 적합성 검증을 위해 YCSB(Yahoo! Cloud Serving Benchmark)와 같은 벤치마킹 도구가 활발히 활용된다[8]. YCSB는 범용적인 목적으로 다양한 NoSQL의 성능을 평가하기 위해서 사용자가 지정한 데이터베이스에 대해 CRUD(Create/Read/Update/Delete) 연산을 수행하고 연산에 대한 응답 시간을 측정한다. 그러나 YCSB는 범용성을 위해서 단순 CRUD 연산 기반의 벤치마킹을 수행하기 때문에 그래프 데이터베이스에 대한 벤치마킹을 지원하기 위해서는 그래프 데이터베이스에서 이루어지는 연산을 구현해야 한다. 또한, 데이터 입력 단계에서도 적절한 그래프 데이터 입력 방법이 제공되어야 한다.

그래프 데이터베이스는 저장하는 형태와 저장된 데이터에 대한 질의 유형이 기존 NoSQL 데이터베이스와 매우 다른 특성을 보이기 때문에 그래프 데이터베이스에 알맞은 벤치마킹 도구가 필요하다[3-5]. 기존 그래프 데이터베이스에 대한 벤치마킹을 지원하는 다양한 그래프 데이터베이스 벤치마킹 도구들이 개발되었다[9-12]. 그래프 데이터베이스 전용 벤치마킹 도구들은 대부분 사전 정의된 데이터를 활용하여 데이터를 입력하고 입력된 그래프에 대한 벤치마킹을 수행한다. 또한, 그래프 전용 질의가 구현되어있어 사용자는 질의 선택을 통해 대상 그래프 데이터베이스에 대한 벤치마킹을 수행한다. 질의는 입력 데이터와 마찬가지로 사전 정의된 형태의 질의 대상 목록을 활용하여 벤치마킹을 수행한다. 그러나 실제 응용 환경을 고려하기 위해서는 입력되는 그래프 데이터와 질의의 대상 데이터가 불규칙하고 예측 불가능한 형태이다. 따라서 실제 응용 환경과 같은 환경에서의 벤치마킹을 수행하기 위해서는 사용자 정의에 맞는 데이터를 생성하고, 질의 또한 사용자 정의에 맞게 수행되어야 한다. 더불어 기존 그래프 데이터베이스 벤치마킹 도구들은 그래프 입력에 대한 성능 측정을 수행하지 않기 때문에 YCSB와 유사하게 그래프 입력에 대한 성능 측정 결과를 제공해야 한다.

본 논문에서는 그래프 입력과 질의에 대한 다양성을 지원하는 그래프 데이터베이스 벤치마킹 시스템을 제안한다. 제안하는 시스템은 그래프 데이터베이스에 대한 벤치마킹을 테스트하기 위해서 OrientDB를 활용한다. 기존 그래프 데이터베이스 벤치마킹 도구를 활용하여 새로운 기능을 추가한 벤치마킹 도구를 설계하고 구현한다. 입력 그래프와 질의 그래프의 다양성을 지원하기 위해서 기존 그래프 데이터 생성 도구를 활용한다. 실험 결과 분석을 통해 제안하는 기법의 타당성 및 실효성을 입증한다.

본 논문의 구성은 다음과 같다. II장에서는 관련 연구를 설명한다. III장에서는 제안하는 기법의 특징과 벤치마킹 시스템에 관해 설명하고, IV장에서는 제안하는 기법의 타당성 및 실효성을 입증하기 위해 성능 평가를 수행한다. 마지막으로 V장에서는 본 논문의 결론과 향후 연구를 제시한다.

II. 관련 연구

LDBC-SNB(Linked Data Benchmark Council-Social Network Benchmark)는 소셜 네트워크 데이터를 기반으로 그래프 데이터베이스를 벤치마킹하는 벤치마킹 도구이다[9]. 벤치마킹에 사용되는 그래프 질의는 실시간 워크로드인 interactive 워크로드와 분석용 워크로드인 business intelligence 워크로드로 분류가 된다.

LDBC는 그래프 데이터베이스 전용 벤치마킹 도구로써는 가장 활발히 활용된다. 더불어서 하둡과 같은 빅데이터 처리 플랫폼을 활용하여 대용량 그래프 데이터 생성을 지원한다.

[10]은 ArangoDB 사에서 활용하는 NoSQL 성능 측정 도구이다. 오픈 소스 프로젝트를 통해 성능 평가에 대해 누구나 검증하고 재현할 수 있도록 하였다. 쉘 스크립트로 구현되어 그래프 데이터베이스에 대한 질의를 쉽게 확장 구현 할 수 있는 장점이 있다. ArangoDB, Neo4j, MongoDB, OrientDB, PostgreSQL에 대한 성능 평가가 수행 가능하다. SNAP[13]에서 제공하는 소셜 네트워크 데이터인 Pokec을 활용하여 그래프 데이터에 대한 질의 처리 성능을 측정하고 있다[14]. 정점(문서)에 대한 읽기 쓰기 연산과 더불어 집계 연산과 특정 정점의 이웃 검색, 최단 경로 질의를 지원하고 있다. 그러나 질의 대상의 목록이 미리 정의되어야 한다는 단점이 있다.

[11]은 소셜 네트워크 데이터를 다룰 수 있는 그래프 데이터베이스 4종(Titan, OrientDB, Neo4j, Sparksee)에 대한 성능을 비교한 논문[15]에서 사용된 벤치마크 분석 도구이다. 커뮤니티 감지, 대용량 삽입, 증분 삽입, 이웃 및 최단 거리 검색 질의를 활용하여 성능 평가를 수행한다. 기존 도구들과는 달리 다양한 데이터 집합을 지원하고 있으나 일부 질의는 사전 생성된 데이터 집합을 활용해야 한다.

[12]는 LDBC-SNB가 생성하는 가상 그래프 데이터를 기반으로 Neo4j, SQLGraph, Vertica에 대해 벤치마킹을 수행하는 그래프 데이터베이스 벤치마킹 도구이다. 그래프 데이터베이스 질의 언어인 Cypher를 통해 워크로드 질의를 구현하였다[16]. 100GB 크기의 대용량 그래프 데이터에 대한 성능 평가도 가능한 특징이 있다. 워크로드에 사용되는 질의는 LDBC의 Interactive 질의 중 5개를 선별하여 해당 질의에 대한 성능 평가를 수행한다. 벤치마킹 결과를 웹 UI 형태로 가시화할 수 있는 장점이 존재한다. 다른 벤치마킹 도구와 마찬가지로 벤치마킹을 수행하기 위해서 사전 정의된 질의 대상 객체 리스트가 필요하다.

LDBC는 실세계에서 활용되는 가상의 그래프와 질의를 생성하는 도구이기 때문에 본 연구에서 추구하는 사용자 정의가 가능한 입력 그래프 생성에 활용 할 수 있다. 기존 그래프 데이터베이스 벤치마킹 도구들은 입력 그래프가 사전에 정의된 경우가 많다[10][12]. 사용자가 원하는 그래프 데이터에 대한 벤치마킹을 수행하기 위해서는 입력 그래프에 대한 다양성을 지원해야한다. [11]은 입력 데이터의 다양성을 일부 지원하나 사용자가 정의하고자 하는 데이터를 생성해주지는 못한다. 입력 그래프뿐만 아니라 질의 그래프에 대한 다양성을 지원하는 기능도 매우 중요하다. 기존 벤치마킹 도구들은 질의에 대한 정의를 상세히 하고 있지만 질의 실행 횟수, 실행 순서, 질의 대상 목록을 사전 정의된 형태로만 수행되기 때문에 실세계의 질의를 그대로 구현했다고는 볼 수 없다. 본 연구에서는 이러한 단점을 보완하기 위해서 질의에 대한 조정 가능한 파라미터를 상세하게 수정 가능한 형태의 기능을 제공한다. 추가로 데이터 입력과 질의 처리를 분리하여 YCSB와 유사하게 두 단계 결과를 제공하여 벤치마킹 결과의 다양성을 제시한다.

III. 제안하는 그래프 데이터베이스 벤치마킹 시스템

1. 전체 시스템 구조

다양한 그래프 입력과 질의 다양성을 지원하는 그래프 데이터베이스 벤치마킹을 수행하기 위해서는 새로운 벤치마킹 도구가 필요하다. 본 논문에서는 사용자가 크기를 다양하게 지정 할 수 있는 그래프 데이터 입력 기능과 질의 횟수 및 비율 또한 상세하게 설정 가능한 그래프 데이터베이스 벤치마킹 도구를 설계하고 구현한다. 그래프 데이터의 다양성을 지원하기 위해서 LDBC-SNB를 활용하고, 질의 다양성을 지원하기 위해서 기존 벤치마킹 도구를 기반으로 한다. YCSB에서 제안하는 형태의 벤치마킹 단계를 차용하여 데이터 로드, 워크로드 실행 두 단계의 성능이 모두 측정 가능한 벤치마킹 시스템을 제안한다. 벤치마킹 시스템의 유효성을 검증하기 위해서 OrientDB 기반의 벤치마킹을 수행한다. 실제 응용 상황과 유사하게 구현하기 위해서 클러스터 환경을 고려한 벤치마킹을 수행한다.

[그림 1]은 제안하는 벤치마킹 시스템 구조도를 나타낸다. 사용자는 먼저 CLI(Command Line Interface)를 통해 그래프 데이터베이스에서 실행할 질의 정보가 포함된 워크로드 파일을 생성한다. 생성된 워크로드 파일을 지정하여 벤치마킹을 수행한다. 벤치마킹 수행 시 입력 그래프 데이터의 크기와 벤치마킹 대상 데이터베이스를 지정하여 수행한다. 그래프 데이터 생성기는 입력 그래프 데이터 크기의 가상 그래프 데이터를 생성한다. LDBC 시스템을 수정하여 사용자가 지정한 가상 그래프 데이터를 생성한다. 만약 사용자가 입력한 그래프 크기가 사전에 생성된 적이 있다면 해당 파일을 재활용한다. 벤치마킹은 [10]에서 제안된 기법을 일부 수정하여 수행한다. 그래프 벤치마킹 모듈은 지정한 그래프 데이터베이스 (OrientDB 클러스터)에 대해 벤치마킹을 수행하는데, YCSB와 유사하게 데이터를 데이터베이스에 입력하는 적재 단계와 적재된 데이터에 대해 질의를 수행하는 실행 단계 2단계 벤치마킹을 수행한다. 제안하는 벤치마킹 시스템에서는 실행 단계에서 정점 및 간선에 대해 CRUD 연산을 수행하는 실시간 질의와 최단 경로, 이웃 목록 질의, 집계 연산을 수행하는 분석 질의를 구분하여 수행한다. 사용자는 실시간 질의에 대한 연산 비율 및 연산 횟수를 지정할 수 있으며 분석 질의도 마찬가지로 연산의 횟수와 비율을 지정할 수 있다. 벤치마킹이 정상적으로 종료되면 벤치마킹 결과가 CSV(Comma Separated Value) 파일 형태로 생성된다.

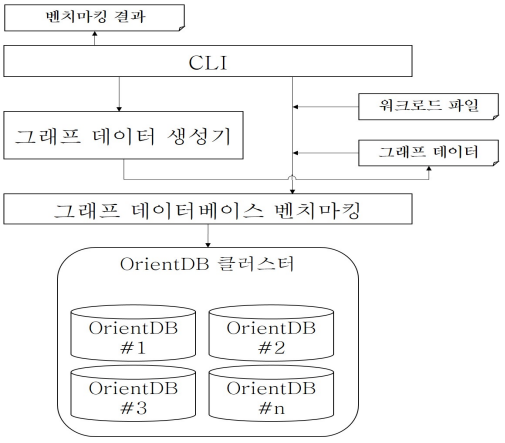


그림 1. 제안하는 벤치마킹 시스템 구조도

2. 벤치마킹 명령

사용자에게 벤치마킹의 편의성을 제공하기 위해서 CLI를 제공한다. 사용자는 CLI를 통해 질의 워크로드 관리 및 벤치마킹 수행을 할 수 있다. 워크로드 관리 기능은 워크로드 생성, 변경, 삭제, 조회가 가능하다. 워크로드 생성을 통해 워크로드 파일이 생성되며 워크로드 안에는 그래프 질의와 관련된 정보를 포함한다.

제안하는 벤치마킹 시스템은 그래프 질의에 대해 두 가지 유형을 제공한다. 첫 번째는 실시간 질의 유형으로, 사전에 입력된 그래프에 대해 정점 및 간선의 조작 연산을 수행하는 질의이다. 정점 및 간선에 대해 생성, 삭제, 갱신을 수행 할 수 있다. 두 번째 질의 유형은 분석 질의로 SSSP(Single Source Shortest Path), 집계 연산, 단순 이웃 조회, 상세 이웃 조회 질의를 수행한다. SSSP는 특정 정점을 기준으로 다른 모든 정점까지의 최단 거리를 계산하는 분석 질의이고, 집계 연산은 정점의 특정 속성을 기준으로 집계 연산을 수행하는 질의이다. 이웃 조회 질의는 특정 정점의 이웃 정보를 조회하는 질의로써, 단순 이웃 조회는 이웃의 목록을 조회하는 연산이며 상세 이웃 조회는 이웃의 목록뿐만 아니라 속성 정보를 모두 조회하는 질의를 의미한다.

[표 1]은 CLI를 통해 생성되는 워크로드 속성 정보를 나타낸다. 질의 유형은 실시간 질의와 분석 질의 두 가지 유형으로 나뉘며, 질의별로 연산 수를 정수로 지정한다. 각 연산 별로 연산 비율을 설정하여 벤치마킹 시 수행되는 사용자가 지정한 연산을 비율에 따라 수행한다. 실시간 질의 같은 경우 정점 및 간선에 관한 연산을 주로 수행하고, 분석 질의는 SSSP, 집계 연산, 단순 이웃 조회, 상세 이웃 조회 연산을 사용자가 지정한 비율에 맞추어 실행한다. 이때 질의 대상이 되는 정점은 임의로 지정한다.

표 1. 워크로드 속성 정보

질의 유형	필드 명	값 유형
실시간 질의	연산 수	정수
	정점 생성 비율	실수
	정점 갱신 비율	
	정점 삭제 비율	
	간선 생성 비율	
	간선 갱신 비율	
	간선 삭제 비율	
분석 질의	연산 수	정수

	최단 거리(SSSP)	실수
	집계 연산	
	단순 이웃 조회	
	상세 이웃 조회	

3. 그래프 데이터 생성기

그래프 데이터베이스에 대해 벤치마킹을 수행하기 위해서는 입력 그래프가 필요하다. 기존 대부분 벤치마킹 시스템은 입력 그래프를 사전에 정의된 형태의 크기의 그래프만을 사용하였다. 실제 응용에서는 다양한 크기의 그래프가 존재하기 때문에, 실제 응용에 맞는 벤치마킹을 수행하기 위해서는 사용자가 정의하는 크기에 따라 그래프가 생성되고 생성된 그래프를 입력받을 수 있어야만 한다.

[그림 2]는 그래프 데이터 생성 과정을 나타낸다. 제안하는 벤치마킹 시스템은 CLI를 통해 사용자가 원하는 형태의 입력 그래프 크기를 받고, 그래프 데이터 생성기는 해당 크기의 그래프 데이터를 생성한다. 만약 해당 크기의 그래프 데이터를 생성한 적이 있다면 이전에 생성한 그래프 데이터를 재활용한다. 그래프 데이터 생성은 기존에 자주 활용되는 LDBC 그래프 데이터 생성기를 활용한다. LDBC 그래프 생성기는 파라미터 정보가 지정된 파일을 기반으로 그래프 데이터를 생성한다. 파라미터 정보로는 데이터 전체 크기, 정점 및 간선 수, 정점의 속성 값의 범위 등을 지정한다. 현실 세계의 그래프 데이터를 재현하기 위해 다양한 분포 기반의 데이터를 생성할 수 있는데, 소셜 네트워크 서비스의 대표 응용인 페이스 북 기반의 데이터 분포를 기반으로 생성한다. 제안하는 벤치마킹 시스템은 신뢰성 있는 그래프 데이터 제공을 위해 LDBC 뿐만 아니라 SNAP에서 제공하는 그래프 데이터를 선택할 수 있게 옵션을 제공한다[13]. 그래프 데이터 생성이 완료되거나 그래프 데이터가 존재하여 생성 단계를 수행하지 않는다면 벤치마킹을 수행한다.

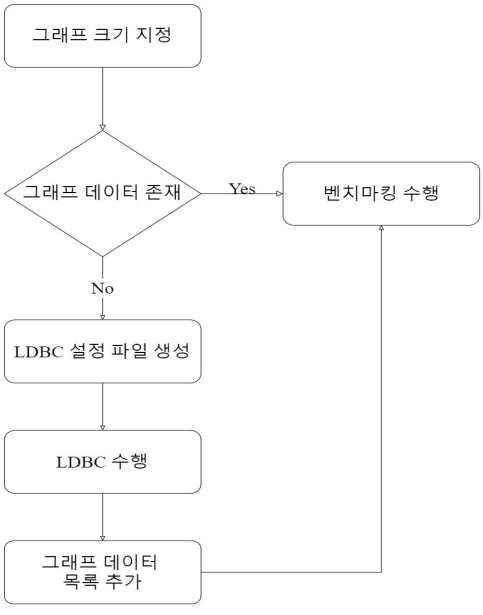


그림 2. 그래프 데이터 생성 과정

4. 벤치마킹 수행 과정

제안하는 벤치마킹 시스템은 CLI를 통해 입력받은 워크로드 및 입력 그래프 크기를 기반으로 벤치마킹을 수행한다. 추가로 벤치마킹 대상이 되는 그래프 데이터베이스를 지정해야 하는데, 본 연구에서는 OrientDB를 기반으로 벤치마킹을 수행하였다.

[그림 3]은 벤치마킹의 전체적인 수행 과정을 나타낸다. 사용자가 CLI를 통해 벤치마킹 파라미터를 입력한다. 사용자가 입력한 크기의 그래프 데이터가 존재하지 않는다면 LDBC를 통해 그래프 데이터 생성을 수행한다. 생성된 입력 그래프를 입력하는 적재 단계를 수행한다. 그 후 적재된 그래프 데이터에 대해 실시간 및 분석 질의를 수행하고 수행된 결과에 대한 파일을 생성한다. 이때 사용자가 지정한 파라미터 (시간 단위)에 따라 성능 평가 결과를 실시간으로 기록하고, 기록된 결과를 바탕으로 가시화를 수행한다.

OrientDB에서는 대용량 그래프 데이터를 효율적으로 수행하는 OETL(OrientDB Extraction-Transformation-Loading)을 제공한다[17]. OETL은 지정한 소스(혹은 파일)에서 데이터를 추출하고 데이터베이스에 저장하기 위해 포맷을 변형하고 최종적으로 데이터를 적재하는 일련의 처리를 수행하는 모듈이다. OrientDB에 서 명령어를 통해 제공해주며 OrientDB에 최적화되어 동작한다.

질의 처리 단계는 실시간 질의를 먼저 수행하고, 분석 질의를 마지막으로 수행한다. 사용자가 사전에 입력

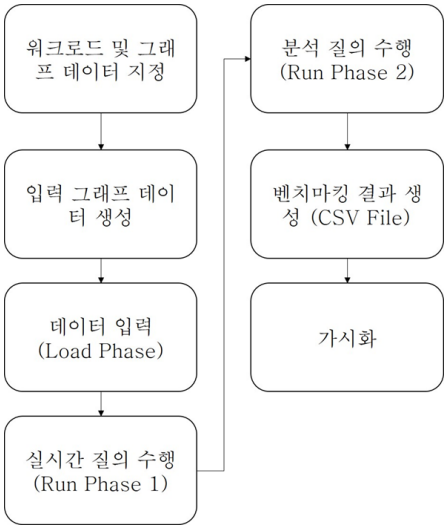


그림 3. 벤치마킹 수행 과정

한 연산 횟수만큼 각 질의를 수행한다. 기존 벤치마킹 시스템에서는 질의에 대한 정보를 사전에 파일로 생성하고 이를 기반으로 질의를 수행하였지만, 제안하는 벤치마킹 시스템은 사용자가 지정한 파라미터에 맞게 질의를 수행하기 때문에 더욱 효율적인 벤치마킹을 수행한다. 벤치마킹이 완료되면 YCSB와 마찬가지로 벤치마킹 결과가 생성된다. 결과에는 평균 지연시간, 최대 지연시간, 95%, 99%, 99.99% 지연시간이 포함된다. 또한 사용자가 지정할 수 있는 시간 단위의 연산 처리 수 및 평균 지연 시간을 같이 측정하여 CSV 파일로 생성한다. 생성된 파일을 기반으로 파이썬의 Matplotlib를 통해 가시화된 결과를 생성한다[18].

IV. 벤치마킹 결과

제안하는 벤치마킹 시스템의 타당성과 실효성을 입증하기 위해서 OrientDB 클러스터에 대한 벤치마킹을 수행한다. [표 2]는 본 논문에서 수행한 벤치마킹 환경을 나타낸다. 동일한 성능을 가진 3대의 서버를 OrientDB 클러스터로 구성하였다. 각 서버는 Intel Core i7-6700 CPU 3.4GHz 8 Core와 32GB의 메모리를 탑재하였다. 성능 평가는 LDBC에서 생성 가능한 데이터 가장 큰 크기의 그래프 데이터 2종을 기반으로 벤치마킹을 수행하였다. 그래프는 363,056개의 정점과 8,122,282개의 간선으로 이루어진다. 정점별 크기를 변경하여 10GB, 100GB 파일로 생성하였다. 질의 처리에서는 전체 그래프 크기의 5%에 해당하는 I/O 연산을 수행하였으며, 페이스 북에서 분석한 그래프 질의 처리 비율을 참고하였다[19].

표 2. 벤치마킹 환경

이름	값
CPU	Intel Core i7-6700 CPU 3.4GHz 8 Core
메모리	32GB
운영 체제	Cent OS 8.2.2004
Kernel 버전	4.9.216 x86_64
OrientDB 버전	2.2.29
LDBC 버전	0.3.2
서버 수	3
정점 수	363,056
간선 수	8,122,282
파일 크기	10GB, 100GB

[표 3]은 OrientDB 기반의 벤치마킹 수행 결과를 나타낸다. 10GB 파일의 경우 적재 시간은 약 0.92시간이 소요되었으며, 질의 처리 시간은 0.56시간이 소요되었다. 분석 질의 1,000번을 수행한 평균 지연시간은 2.02ms로 측정되었으며 최대 지연시간은 약 4.7초가 소요되었다. 100GB 파일의 경우 적재 시간은 약 169.3시간이 소요되었으며, 질의 처리 시간은 0.9시간이 소요되었다. 분석 질의 5,000번을 수행한 평균 지연시간은 0.64ms로 측정되었으며 최대 지연시간은 약 2.4초가 소요됨을 확인하였다.

표 3. 벤치마킹 결과

구분	10GB	100GB
Load Time(hr)	0.92	169.3
Run Time(hr)	0.56	0.9
Operations	1000	5000
Average Latency(ms)	2.02	0.64
Max Latency(ms)	47827	26060
95th PercentileLatency(ms)	45435.65	24757
99th PercentileLa	47348.73	25799.4

tency(ms)		
99.99 PercentileLatency(ms)	47822.2173	26057.394

[그림 4]는 연산 처리 성능에 대한 벤치마킹 결과이다. 사용자가 지정한 단위 시간당 평균 질의 처리 횟수를 나타낸다. W는 사용자가 지정한 시간 범위를 의미한다. 시간에 따른 변동 폭이 매우 크게 나타나는 것을 확인 할 수 있었다. 입력 그래프 데이터의 크기가 크지 않음을 고려하더라도 여러 노드에 데이터베이스가 분산되어 있으면 순회로 발생하는 질의 처리 오버헤드가 원인으로 보인다. [그림 5]는 연산 지연 시간 관점의 결과이다. 앞서 지정한 시간 처리 단위당 평균 지연 시간을 나타내는데 일부 이상치를 제외하고 보더라도 연산 처리 성능 결과와 마찬가지로 매우 큰 변동 폭을 나타내고 있다.

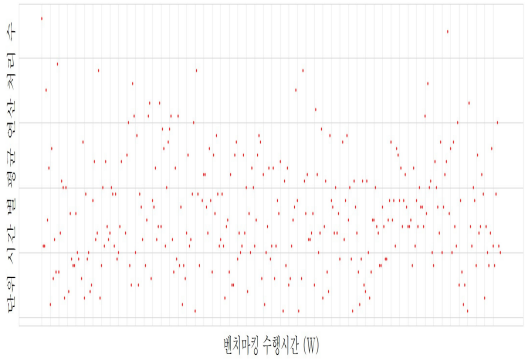


그림 4. 연산 처리 성능 벤치마킹 결과

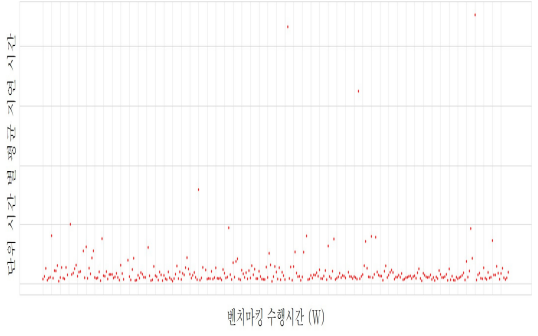


그림 5. 연산 지연시간 벤치마킹 결과

[그림 6]은 노드별 I/O 연산 수를 나타낸다. 이전과 마찬가지로 사용자가 정의한 시간 단위의 연산 수를 기록하였다. I/O 연산 수는 3개의 노드별로 확인 가능한데, 주로 노드 1번에 많은 I/O가 발생한다. I/O 연산은 읽기 연산과 쓰기 연산을 모두 확인 할 수 있는데, 그래프 질의 처리는 읽기 연산이 주로 발생하는 현상을 나타낸다. 또한, 노드 3번은 안정적인 성능을 나타내고 있다. 이러한 현상 확인을 통해 질의 처리 스케쥴링이나 로드 밸런싱을 수정하여 성능의 최적화를 도모할 수 있다.

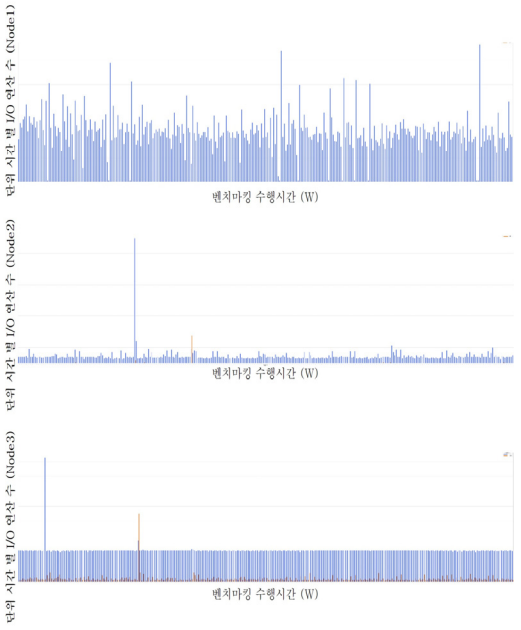


그림 6. 노드별 I/O 연산 수

[그림 7]은 노드별 GC(Garbage Collection) 수를 나타낸다. 여기서 GC는 보조 기억 장치에서의 GC를 의미한다. 노드 1번이 안정적인 GC 수를 보임으로써, 연산 수가 상대적으로 많더라도 보조 기억 장치에 부하를 크게 발생하지 않는다. 이러한 노드별 GC 성능 확인을 통해 GC를 주로 발생시키는 질의 연산을 분석해 볼 수 있다. 다만 시간 단위별 연산 정보를 기록하고 해당 질의가 GC를 발생시키는지에 대한 여부는 더욱 세밀한 인과 관계를 확인해야 한다. 벤치마킹 결과 제안하는 기법은 기존 기법과 달리 사용자가 지정 가능한

형태의 입력 그래프와 질의를 지정할 수 있음을 입증하였다. 추가로 시간 단위별 벤치마킹 결과를 제공함으로써, 질의에 따라 성능이 취약해지는 부분을 시각적으로 확인 할 수 있는 결과가 제공 가능하다.

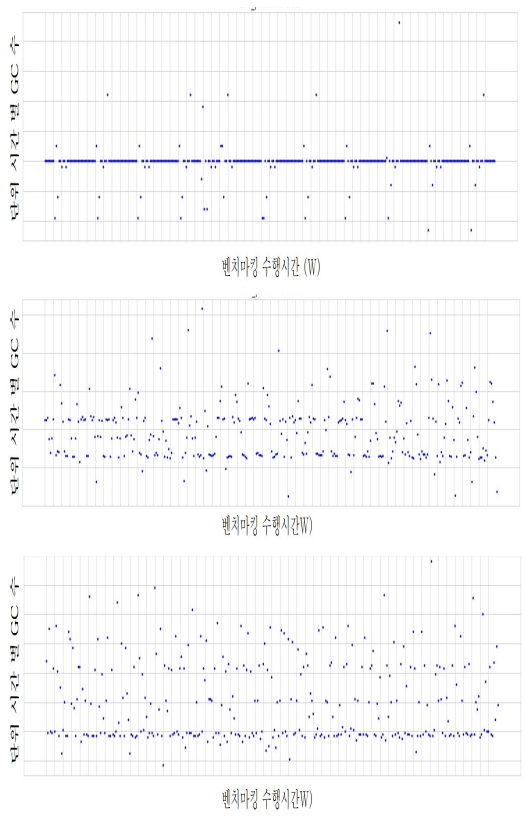


그림 7. 노드별 GC 수

V. 결 론

본 논문에서는 그래프 입력과 질의에 대한 다양성을 지원하는 그래프 데이터베이스 벤치마킹 시스템을 제안하였다. 제안하는 시스템은 그래프 데이터베이스에 대한 벤치마킹을 수행하기 위해 기존 그래프 데이터베이스 벤치마킹 도구인 YCSB를 기반으로 하였다. 제안하는 시스템은 입력 그래프와 질의 그래프의 다양성을 지원하기 위해서 CLI를 제공하며 그래프 데이터 생성기인 LDBC를 활용하여 다양한 크기의 입력 그래프를 생성한다. 다중 모델 NoSQL인 OrientDB 기반의 벤치마킹 수행을 통해 제안하는 벤치마킹 시스템의 실효성을 입증하였다. 본 연구 결과를 통해 그래프 데이터베이스 벤치마킹 도구를 일반화시킬 수 있음을 입증함으로써, 기존 그래프 데이터베이스 벤치마킹 연구에서 해결하지 못하였던 문제점들을 해결하였다. 또한 실제 그래프 데이터베이스를 활용하고 싶은 기업들에는 실제 적용 전 생길 수 있는 문제점들을 미리 확인하고 테스트를 수행할 기회를 제공하여 위험 부담을 감소시킬 수 있다. 제안하는 기법은 OrientDB 기반의 그래프 벤치마킹 결과를 제시하고 있다. 그러나 그래프 데이터베이스마다 데이터 입력 방식이 다르고, 질의 처리하는 방식도 다르다. 따라서 범용적인 그래프 데이터베이스 벤치마킹을 수행하기 위해서는 단일화된 인터페이스 설계가 필요하다. 또한 시간 단위의 벤치마킹 결과를 확인할 수 있지만 더욱 의미 있는 결과를 생성하기 위해서는 해당 시간 단위별 수행하는 연산의 통계를 같이 제시할 수 있어야만 한다. 이러한 문제점을 해결하기 위해서 향후에는 OrientDB 뿐만 아니라 범용적인 데이터베이스 인터페이스를 설계하고, 시간 단위별 다양한 벤치마킹 결과를 제시하여 다양한 시각을 제공할 수 있는 그래프 데이터베이스 벤치마킹 도구를 제안할 예정이다.

참 고 문 헌

[1] J. Han, E. Haihong, G. G. Le, and J. Du, “Survey on NoSQL Database,” In 2011 6th international conference on pervasive computing and applications, pp.363-366, 2011.

[2] R. Hecht and S. Jablonski, “NoSQL Evaluation: A Use Case Oriented Survey,” In 2011 International Conference on Cloud and Service Computing, pp.236-341, 2011.

[3] Y. Li and S. Manoharan, “A Performance Comparison of SQL and NoSQL Databases,” In 2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), pp.15-19, 2013.

[4] A. Davoudian, L. hen, and M. Liu, “A Survey on

NoSQL Stores," ACM Computing Surveys (CSUR), Vol.51, No.2, pp.1-43, 2018.
[5] D. Fernandes and J. Bernardino, "Graph Databases Comparison: AllegroGraph, ArangoDB, InfiniteGraph, Neo4J, and OrientDB," In Data, pp. 373-380, 2018.
[6] S. Ray, B. Simion, and A. D. Brown, "Jackpine: A Benchmark to Evaluate Spatial Database Performance," In 2011 IEEE 27th International Conference on Data Engineering, pp.1139-1150, 2011.
[6] L. Sfaxi and M. M. B. Aissa, "Babel: A Generic Benchmarking Platform for Big Data Architectures," Big Data Research, Vol.24, 100186, 2021.
[7] F. Holzschuher and R. Peinl, "Performance of Graph Query Languages: Comparison of Cypher, Gremlin and Native access in Neo4j," Proc. Joint EDBT/ICDT 2013 Workshops, pp.195-204, 2013.
[8] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan and R. Sears, "Benchmarking Cloud Serving Systems with YCSB," Proc. 1st ACM symposium on Cloud computing, pp.143-154, 2010.
[9] O. Erling, A. Averbuch, J. Larriba-Pey, H. Chafi, A. Gubichev, A. Prat and P. Boncz, "The LDBC Social Network Benchmark: Interactive Workload," Proc. International Conference on Management of Data, pp.619-630, 2015.
[10] https://www.arangodb.com/2018/02/nosql-performance-benchmark-2018-mongodb-postgresql-orientdb-neo4j-arangodb/, 2021.08.23
[11] https://github.com/socialsensor/graphdb-benchmarks, 2021.08.23
[12] https://github.com/Alnaimi-/database-benchmark, 2021.08.23
[13] https://snap.stanford.edu/data/, 2021.08.23
[14] https://pokec.azet.sk/, 2021.08.23
[15] S. Beis, S. Papadopoulos, and Y. Kompatsiaris, "Benchmarking Graph Databases on The Problem of Community Detection," In New Trends in Database and Information Systems II, pp.3-14, 2015.
[16] N. Francis, A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault, and A. Taylor, "Cypher: An Evolving Query Language for Property Graphs," Proc. International Conference on Management of Data, pp.1433-1445, 2018.
[17] P. Vassiliadis, A. Simitsis, and S. Skiadopoulos, "Conceptual Modeling for ETL Processes," Proc. International workshop on Data Warehousing and OLAP, pp.14-21, 2002.
[18] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," Computing in science & engineering, Vol.9, No.3, pp.90-95, 2007.
[19] T. G. Armstrong, V. Ponnekanti, D. Borthakur, and M. Callaghan, "LinkBench: A Database Benchmark based on The Facebook Social graph," Proc. International Conference on Management of Data, pp.1185-1196, 2013.

저 자 소 개

최 도 진(Do-Jin Choi) 정회원

- 2014년 2월 : 한국교통대학교 컴퓨터공학과(공학사)
- 2016년 2월 : 한국교통대학교 컴퓨터공학과(공학석사)
- 2020년 2월 : 충북대학교 정보통신공학과(공학박사)
- 2020년 3월 ~ 2020년 8월 : 충북대학교 정보통신공학과 박사후연구원 (Postdoc)

〈관심분야〉 : 연속 질의 처리, 그래프 스트림, 빅데이터

백 연 희(Yeon-Hee Baek) 준회원

- 2019년 2월 : 충북대학교 경영학부 (복수전공은 빅데이터 연계전공)(학사)
- 2021년 2월 : 충북대학교 빅데이터 협동과정(석사)

〈관심분야〉 : 소셜 네트워크, SIoV, 빅데이터 처리, 데이터베이스

이 소 민(So-Min Lee) 준회원

- 2019년 2월 : 충북대학교 정보통신공학부(공학사)
- 2019년 9월 ~ 현재 : 충북대학교 정보통신공학과(석사)

〈관심분야〉 : 그래프 처리, 연속 질의 처리, 빅데이터, 기계학습

김 윤 아(Yun-A Kim) 준회원

- 2020년 2월 : 청주대학교 통계학과(이학사)
- 2020년 3월 ~ 현재 : 충북대학교 빅데이터협동과정(석사)

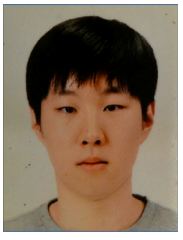
〈관심분야〉 : 소셜 네트워크, 기계학습, 빅데이터 처리, 데이터베이스

김 남 영(Nam-Young Kim) 준회원

- 2020년 2월 : 청주대학교 통계학과(이학사)
- 2020년 3월 ~ 현재 : 충북대학교 빅데이터 협동과정(석사)

〈관심분야〉 : 그래프 처리, 그래프 스트림, 빅데이터, 소셜 네트워크

최 재 용(Jae-Yong Choi) 준회원



- 2020년 2월 : 충북대학교 정보통신공학부(공학사)
- 2020년 3월 ~ 현재 : 충북대학교 정보통신공학과(석사)

〈관심분야〉 : 소셜 네트워크, 빅데이터 처리

이 현 병(Hyeon-Byeong Lee) 정회원

- 2016년 8월 : 한국교통대학교 컴퓨터공학과(공학사)
- 2018년 8월 : 한국교통대학교 컴퓨터공학과(공학석사)
- 2019년 3월 ~ 현재 : 충북대학교 정보통신공학과(박사과정)

〈관심분야〉 : 그래프 스트림, 빅데이터, 데이터베이스 시스템

송 석 일(Seok-Il Song) 종신회원

- 1998년 2월 : 충북대학교 정보통신공학과(공학사)
- 2000년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2003년 2월 : 충북대학교 정보통신공학과(공학박사)
- 2003년 7월 ~ 현재 : 한국교통대학교 컴퓨터공학과 교수

〈관심분야〉 : 데이터베이스, 센서 네트워크, 스토리지 시스템 등

임 종 태(Jong-Tea Lim) 정회원

- 2009년 2월 : 충북대학교 정보통신공학과(공학사)
- 2011년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2015년 8월 : 충북대학교 정보통신공학과(공학박사)
- 2015년 9월 ~ 2019년 8월 : 충북대학교 정보통신공학과 Postdoc.
- 2019년 10월 ~ 현재 : 충북대학교 전자정보대학 정보통신 공학부 초빙 조교수

〈관심분야〉 : 소셜 미디어, 빅데이터, 시공간 데이터베이스, 위치기반 서비스 등

복 경 수(Kyoung-Soo Bok) **종신회원**

- 1998년 2월 : 충북대학교 수학과 (이학사)
- 2000년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2005년 8월 : 충북대학교 정보통신공학과(공학박사)
- 2005년 3월 ~ 2008년 2월 : 한국과학기술원 정보전자연구소 Postdoc
- 2008년 3월 ~ 2011년 2월 : 가인정보기술 연구소 차장
- 2011년 3월 ~ 2019년 8월 : 충북대학교 전자정보대학 정보통신공학부 초빙교수
- 2019년 9월 ~ 현재 : 원광대학교 SW 융합학과 조교수

〈관심분야〉 : 데이터베이스 시스템, 이동 객체 데이터베이스, 이동 P2P 네트워크, 소셜 네트워크 서비스, 빅데이터 처리 등

유 재 수(Jae-Soo Yoo) **종신회원**

- 1995년 2월 : KAIST 전산학과(공학박사)
- 1995년 2월 ~ 1996년 8월 : 목포대학교 전산통계학과 전임강사
- 1996년 8월 ~ 현재 : 충북대학교 전자정보대학 정보통신공학부 정교수
- 2009년 3월 ~ 2010년 2월 : California State University, 방문교수
- 2019년 9월 ~ 2020년 8월 : California State University, 방문교수

〈관심분야〉 : 데이터베이스 시스템, 멀티미디어 데이터베이스, 빅데이터 등