GDLL: A Scalable and Share Nothing Architecture Based Distributed Graph Neural Networks Framework

2세부

경희대학교 KYUNG HEE UNIVERSITY

연구 동기

- 최첨단 솔루션들은 확장성이 좋은 MapReduce, 그래프 신경망의 배치 학습, 내결함성과 같은 기존의 shared-nothing 분산 아키텍처 활용에 한계를 가짐
- 기존 프레임워크들은 그래프 학습 모델의 학습에 더 중점을 두지만, 시스템 무결성 및 일반화 가능성을 간과함

연구 목표

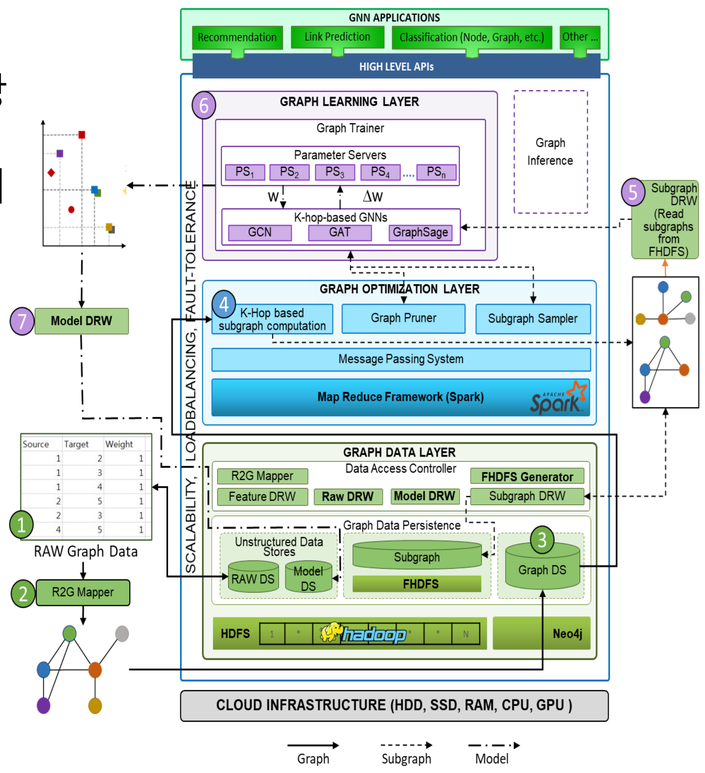
- 초거대 그래프를 관리할 수 없었던 독립형 고성능 머신에 그래프 데이터의 메모리 내 저장 방식을 연구
- 그래프 저장소와 파라미터 서버 사이의 대규모 통신을 유도할 수 있는 맞춤형 그래프 저장소의 개발

관련 기술

- AGL
  - 하둡 에코 시스템을 활용하여 메시지 전달 체계를 기반으로 함
  - MapReduce 환경에서 그래프 신경망을 학습하며, 각 노드에 대한 “정보 완전성 기반 k-홉 서브 그래프”를 생성함
  - 100만 노드의 거대 그래프에 대한 2계층 그래프 어텐션 신경망(GAT) 학습에 14시간이 소요됨
- DistDGL
  - 동기식 학습 접근을 통해 에고 네트워크가 미니 배치를 형성함
  - 최소 에지 분할과 METIS 분할 알고리즘을 사용함
  - 미니 배치 사이에 최적의 로드 균등 분할 방안이 부족함

GDLL의 전체 동작 구조도

- GDL
  - 1단계: 원본 그래프 데이터는 RAW DS로 저장
  - 2단계: R2G 매퍼는 원본 그래프를 지정 그래프 형식으로 매핑
  - 3단계: 매핑 된 그래프를 그래프 DS에 저장
- GOL
  - 4단계: 분산 인 메모리(Spark RDD)에 그래프를 로드하여 정보 완전성을 갖춘 k-홉 서브 그래프를 계산
  - 5단계: 서브 그래프를 F-HDFS에서 인덱싱
- GLL
  - 6단계: 모델을 생성하고, 파라미터 서버를 활용하여 분산 학습 수행
  - 7단계: 학습된 모델은 추론을 위해 모델 DS에 저장

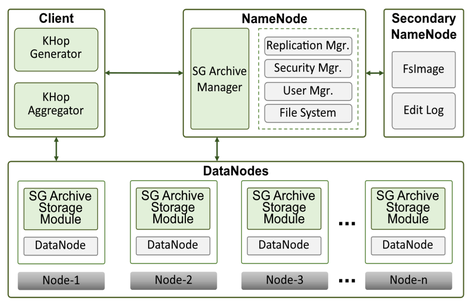


GDLL의 주요 동작 과정

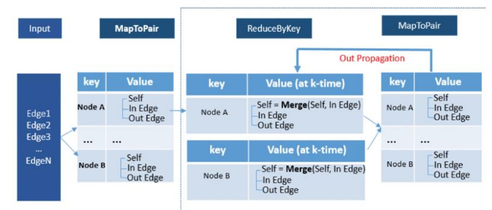
① 그래프 학습 데이터 매니저(GDL)
- 신경망 학습 라이프사이클 전반에 걸쳐 그래프 데이터 관리함
- 그래프 액세스 컨트롤러를 통해 중간 생성물 입출력을 수행
- K-HDFS를 통해 서브 그래프를 배낭에 병합하여 HDFS에 저장하고, 다단계 인덱싱을 통해 접근을 용이하게 함

② 그래프 학습 처리 매니저(GOL)
- 그래프 분할, 샘플링 및 인덱싱 등 중간 단계의 그래프 처리함
- 아파치 스파크를 사용하여 독립 K-홉 서브 그래프 생성
- 서브 그래프들을 인덱싱하여 학습 과정에서 사용
- 샘플링 및 가지치기를 통해 그래프 뒤틀림 문제 해결

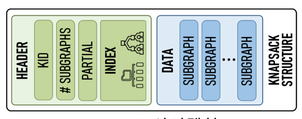
③ 그래프 머신러닝/딥러닝 매니저(GLL)
- 여러 그래프 신경망 모델 구현, 그래프 분산 학습 제공함
- GCN, GraphSAGE, GAT에 대한 효율적인 구현 제공
- 각 파라미터 서버(PS)로 균등하게 서브 그래프를 분배



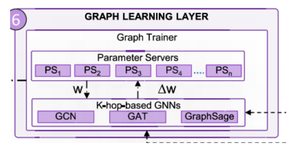
제안 K-HDFS 아키텍처



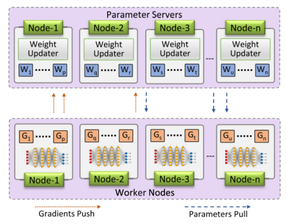
그래프 학습 처리 매니저 세부 구조도



Knapsack 아키텍처



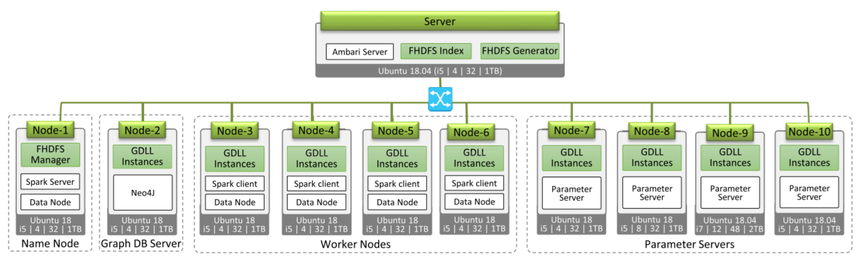
그래프 머신러닝/딥러닝 매니저 세부 구조도



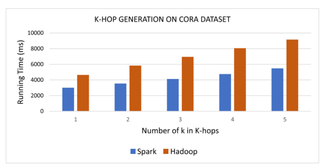
파라미터 서버

성능 평가

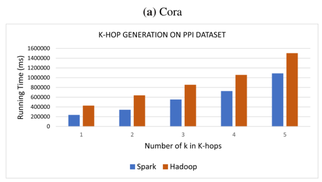
- 실험 설정
  - 11대의 노드 클러스터를 구축하여 HDP 실행
  - 모든 노드는 4개 코어, 32GB 램, 1TB 디스크, Core i5, Ubuntu 18 사용



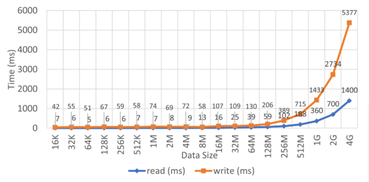
- Cora, PPI 데이터 세트를 이용한 k-홉 서브그래프 생성 평가
- K-HDFS 읽기 성능(서브그래프 당 시간)

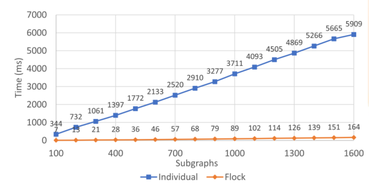


(a) Cora



(b) PPI





- GDLL, AGL 비교
  - OGBN-product의 분산 학습
  - 확장성 검사를 위해 “d”로 대표되는 피처 주입