데이터 분배와 통신 패턴 최적화를 통한 대규모 그래프 파티셔닝

박영진, 정재윤, 안성배, 오상윤*
아주대학교

jacob717@ajou.ac.kr, jyjung9910@ajou.ac.kr, aqua1756@ajou.ac.kr, *syoh@ajou.ac.kr

Mitigating Communication Bottleneck in Large-scale Graph Partitioning via 2D Data Distribution and Persistent Communication

Youngjin Park, Jaeyoon Cheong, Seongbae An, Sangyoon Oh*
Ajou Univ.

요 약

대규모 그래프 처리는 비정형 데이터 접근과 노드 간 빈번한 데이터 교환, 전역 동기화를 수반하므로 시스템 확장성을 확보하기 위해 효율적인 데이터 분배와 통신 기법의 선택이 필수적이다. 본 연구는 분산 메모리 환경에서 대규모 그래프 파티셔닝 알고리즘을 수행할 때 발생하는 통신 병목 현상을 완화하고자 데이터 분배 방식과 MPI 통신 패턴을 비교 분석하였다. 이를 바탕으로 MPI 기반 Label Propagation 모델을 구현하여 프로세스 증가에 따른 통신 오버헤드 완화 효과와 그에 따른 확장성을 종합적으로 평가하였다.

Ⅰ. 서론

소셜 네트워크, 생물 정보학, 웹 그래프 등에서 생성되는 방대한 규모의 그래프 데이터는 단일 컴퓨터 노드의 메모리 용량을 훨씬 초과하며, 분산 처리 환경에서 노드 간 빈번한 데이터 교환은 통신 병목의 주된 요인이 된다. 이러한 대규모 그래프를 효율적으로 하기 위한 Label Propagation(LP)[1]나 Multi-level Partitioning[2] 기반의 분산 알고리즘이 반드시 필요하지만, 분산 환경에서의 노드 수 증가에 따라 통신 비용이 기하급수적으로 늘어나게 되어 전체 성능이 저하되는 문제를 야기한다.

그래프 처리 알고리즘은 데이터 접근과 통신 패턴이 매우 불규칙하다는 특성을 지닌다.[3] 이러한 불규칙한 패턴은 분산 컴퓨팅 환경에서 노드 간의 빈번한 데이터 이동과 동기화 문제, 그리고 프로세스 증가에 따른 통신 비용 급증을 야기하며, 이는 결국 전체 시스템의 성능 저하로 이어진다. 따라서 통신 병목 현상을 최적화하는 것은 전체 실행 시간을 단축할 뿐만 아니라, 시스템 규모 확장에 따른 성능 저하를 억제하여 확장성을 향상시키는 데 필수적이다.

본 연구에서는 이러한 그래프 파티셔닝의 특성을 고려하여, 분산 처리 과정 중 발생하는 통신 오버헤드를 완화하기 위한 기법들을 비교 분석한다. 데이터 분배 측면에서 1 차원 분배 방식과 2 차원 분배 방식을 통해 통신 범위에 따른 오버헤드 변화를 분석하고, 통신 패턴 측면에서 Neighborhood Collective Communication 대비 Persistent Neighborhood Communication 이 지니는 통신 병목 완화 성능을 평가한다. 최종적으로는 분석된 결과를 바탕으로 Message Passing Interface (MPI) 기반 LP 모델을 구현하여, 통신 오버헤드 완화 효과와 확장성을 정량적으로 검증한다.

Ⅱ. 데이터 분배 및 통신 패턴 비교 분석

i. 데이터 분배

분산 그래프 처리에서 데이터 분배 방식은 통신량과 프로세스 간 부하 균형을 결정하는 핵심 요소 중 하나이다. 그래프의 정점을 기준으로 각 프로세스에 분할하는 방식과 같은 1 차원(1D) 분배 방식은 구현이 직관적이라는 장점이 존재하나 프로세서 수가 증가함에 따라 각 프로세스가 통신해야 하는 대상이 선형적으로 증가하게 되어, 대규모 클러스터에서는 통신 오버헤드가 급증하는 단점이 있다. 프로세스를 논리적인 격자 구조로 배치하고, 격자 크기를 이용하여 2 차원 좌표로 변환한 정점 ID 기준으로 분배하는 2 차원(2D) 분배 방식은 통신 범위를 이웃 프로세스들로 제한함으로써 오버헤드를 완화한다.

ii. MPI 통신 패턴

2 차원 직사각형 격자로 분할된 그래프 데이터는 각 프로세스가 통신해야 할 이웃 프로세서가 토폴로지 상에서 고정되는 특징을 가진다. 이러한 구조적 특성을 활용한 Neighborhood Collective Communication (NCC)은 프로세스 간 통신 범위를 가상 토폴로지 내의 이웃 노드로 국한함으로써 불필요한 데이터 전송을 차단하여 통신 병목을 완화할 수 있다.

LP 의 파티션 경계 확장과 같이 동일한 과정이 반복되는 그래프 연산 특성을 고려할 때, NCC 은 매 단계마다 통신 객체를 초기화하고 해제하는 과정을 반복하며 고정적인 오버헤드를 누적시키는 한계가 존재한다. 이러한 구조적 문제를 극복하기 위한 통신 패턴으로는 Persistent Neighborhood Communication (PNC)이 있다. PNC 은 실행 중 변하지 않는 프로세스 간 이웃 관계를 활용하여 통신 객체를 사전에 정의하고 재사용함으로써 반복 호출 오버헤드를 제거한다.

Ⅲ. MPI 기반 효과적인 Label Propagation 모델 구현

앞서 분석한 2D 분배 방식과 PNC 을 적용하여 MPI 기반 LP 모델을 구현하였다. 먼저 그래프 데이터를 2D 분배 방식에 따라 CSR 구조로 로드하며, 각 프로세서는 2D 격자 토폴로지와 자신이 소유한 정점의 인접 정보를 참조하여 MPI 분산 그래프 생성자를 통해 프로세스 간 토폴로지를 정의한다. 이어 FSNS[4] 기법을 적용하여 각 파티션의 중심이 되는 시드(seed) 노드를 선정하고, 부하 균형을 위해 라운드 로빈(round-robin) 방식으로 각 프로세스에 할당한다. 반복적인 라벨 갱신 및 원격 노드 요청이 수반되는 파티션 경계 확장 단계에서는 통신 부하를 줄이기 위해 통신 특성에 따라 기법을 구분하였다. 정적 크기의 메타데이터 교환에는 PNC 을, 가변 크기의 실 데이터 교환에는 NCC 을 활용하였으며, 알고리즘 종료 조건 확인에는 비동기 통신을 적용하여 통신 오버헤드를 완화하였다.

Ⅳ. 실험

실험 환경은 2 개의 Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz 과 32GB RAM, 그리고 Open MPI 5.0.9 버전을 사용하여 구성하였다. 실험에 사용한 데이터셋은 LP 알고리즘의 복잡도를 낮추고, 구현의 편의성을 위해 모두 무방향으로 변환하였으며, 표 1 과 같다.

그래프명	|Node|	|Edge|
arabic-2005	22,744,080	553,903,073
enwiki-2025	6,961,383	164,532,437
eu-2015-host	11,264,052	260,943,053
indochina-2004	7,414,866	150,984,819
uk-2002	18,520,486	261,787,258

표 1. Dataset

모든 실험에서 그래프의 파티션 수는 24, 프로세서 수는 부하 균형을 위하여 파티션 수의 약수 개로 설정하고 실험을 진행하였다.

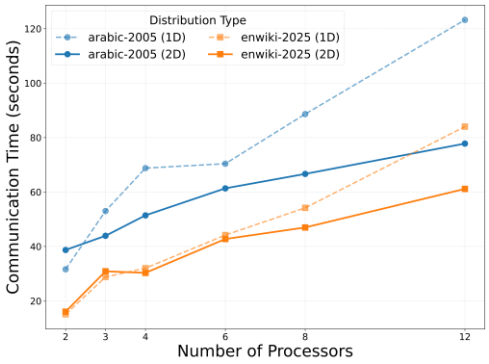

그림 1. 데이터 분배 방식에 따른 통신 오버헤드

그림 1 은 데이터 분배 방식에 따른 통신 시간 차이로, 2D 방식에서 최대 37% 감소하였으며, 별도로 측정한 통신량 또한 1D 방식 대비 낮은 증가율을 보였다.

그림 2 는 NCC 대비 PNC 의 오버헤드 절감 효과로, 대규모 그래프일수록 개선 효과가 뚜렷하다. 비슷한 방식으로 측정한 처리량 또한 PNC 가 더 높은 모습을 보인다.

그림 3 은 구현한 모델에 대하여 고정된 크기의 그래프 데이터에 대해 프로세서 수를 증가하며 성능을 측정했을 때 Speedup 에 대한 그래프로, 프로세스 증가에 따라 필연적으로 통신 오버헤드 또한 증가하기 때문에 특정 성능에 수렴하는 모습을 보인다.

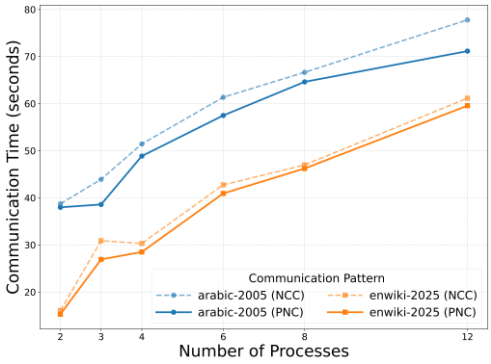

그림 2. 통신 패턴에 따른 통신 시간

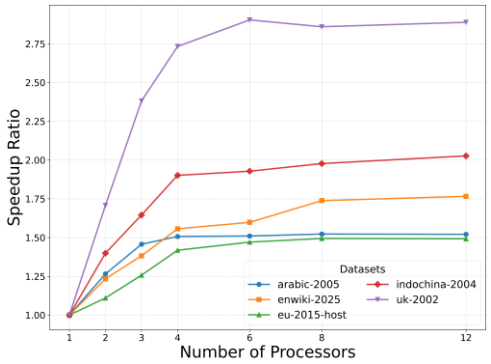

그림 3. 프로세스 증가에 따른 모델 성능

Ⅴ. 결론

본 연구에서는 대규모 그래프 처리 과정에서 발생하는 통신 병목을 완화하기 위해 데이터 분배 방식과 통신 패턴을 비교 분석하였다. 그 결과, 1D 분배 방식보다 통신 범위가 줄어드는 2D 분배 방식이 통신 오버헤드 완화에 우위를 점하고, 반복 연산 상황에서는 객체 초기화 및 해제 오버헤드를 제거한 PNC 가 NCC 보다 더 효과적임을 입증하였다. 이를 토대로, 분산 환경에서 Label Propagation 모델을 구현한 후, 모델의 통신 오버헤드 감소 효과와 확장성을 정량적으로 평가하였다.

ACKNOWLEDGMENT

본 연구는 2026 년 과학기술정보통신부 및 정보통신기획평가원의 SW 중심대학사업(2022-0-01077) 및 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(RS-2023-00283799).

참 고 문 헌

[1] Slota, George M., et al. "Partitioning trillion-edge graphs in minutes." *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2017.

[2] Hendrickson, Bruce, and Robert W. Leland. "A Multi-Level Algorithm For Partitioning Graphs." SC 95.28 (1995): 1-14.

[3] Li, Dongsheng, et al. "TopoX: Topology refactorization for efficient graph partitioning and processing." *Proceedings of the VLDB Endowment* 12.8 (2019): 891-905.

[4] Lee, JeongTae et al. "FSNS: Fast seed node selector using Lorenz curve and Landmarks." Proceedings of the Korean Society of Computer Information Conference, vol. 33, no. 2, 2025, 1-4.