

Compressed Sparse Row 기반 그래프 처리 성능 향상을 위한 재정렬 기법 분석

여인혁, 오상운
아주대학교

{cambria96, syoh}@ajou.ac.kr

Analysis of reordering effect on processing Compressed Sparse Row data

Inhyeok Yeo, Sangyoon Oh
Ajou University

요 약

본 논문에서는 그래프 데이터 표현에 가장 많이 사용되는 Compressed Sparse Row(CSR) 포맷과 CSR 기반의 그래프 데이터 처리의 성능향상을 위한 기법을 분석한다. CSR을 사용한 그래프 데이터 처리는 CSR 포맷의 특성에 따라서 데이터에 불규칙적으로 접근하게 되며, 이에 따라 전체 그래프 처리 성능 저하 문제가 발생하게 된다. 이를 해결하기 위해 재정렬 기법이 많이 사용되고 있으며, 본 논문에서는 이 재정렬 기법 적용에 따른 효과를 분석한다.

I. 서 론

그래프 데이터는 과학 데이터, 산업 데이터, 트위터, 페이스북과 같은 소셜 네트워크 데이터, 그리고 통신망과 같은 네트워크를 표현하기 위해 사용된다. 그래프 데이터 처리는 연결성을 가지는 그래프 데이터의 특성으로 인해 상대적으로 난이도가 높으며, 처리해야 하는 그래프 데이터의 크기가 큰 경우 데이터를 분산해서 처리해야 하기 때문에 메모리와 같은 하드웨어 자원이 제한됨에 따라 처리 시간이 증가하는 문제가 발생한다.

이처럼 그래프 처리는 다른 데이터 처리에 비해 상대적으로 어렵기 때문에 이를 지원하는 다양한 프레임워크(예: nvGRAPH[1], Gunrock[2], Apache Graph[3], Totem[4])가 존재한다. 그래프 데이터는 해당 데이터의 성격, 그리고 처리해야 하는 그래프 알고리즘에 따라 적합한 처리 방식이 달라지기 때문에 그래프 처리 프레임워크들은 더 높은 처리 성능을 달성하고 다양한 환경에서 사용이 가능한 높은 확장성을 가지기 위해 서로 다른 그래프 데이터의 분산 처리 방식, 처리 모델(예: vertex-centric, edge-centric), 및 데이터 표현 방식들을 적용하고 있다. 이 프레임워크들에서 보편적으로 지원되는 그래프 데이터 형식으로는 CSR(Compressed Sparse Row)이 있다. 이 방식은 메모리 등 자원 사용에 있어 높은 효율을 가지지만, 사용에서 나타나는 불규칙적인 메모리 접근과 이에 따른 성능 저하 문제[5]가 발생한다.

II. CSR 기반 그래프 데이터 지역성 향상과 재정렬

본 논문에서는 CSR 기반 그래프 데이터 처리 성능

향상을 위해 사용하는 Reordering 기법의 효과에 대해 분석을 진행한다. Graph 데이터에 대한 Reordering 기법은 Rabbit Order[6]와 RADAR [7] 등의 기존 연구들에서 이미 고려된 기법이며, 본 논문에서는 기존 연구들의 제안에서 사용된 중복 제거(deduplication), 재정렬(reordering)들과 같은 다양한 기법 중 재정렬 기법을 중심으로 효과를 분석한다. CSR format의 Reorder 기법은 offset array(OA)와 coordinate array(CA)의 값을 정렬해 temporal locality와 spatial locality를 개선하는 방법으로서, 이를 통해 그래프 처리 성능을 향상시킬 수 있다.

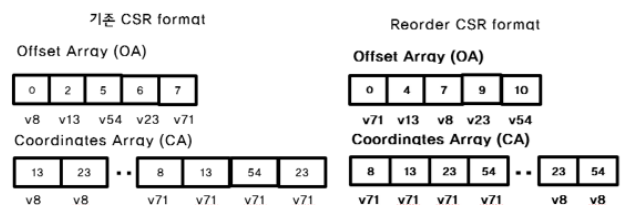


Figure 1 기존 CSR format과 reorder CSR format 비교

기존 CSR format의 경우에는 Figure 1과 같이 OA와 CA의 값이 정렬되어 있지 않아, 해당 배열 값을 메모리에서 접근할 때 불규칙적으로 접근해 cache locality 문제가 발생한다. 하지만 CSR format에 Reordering 을 적용하게 되면, 각 vertex의 degree에 따라 OA의 값을 정렬하고 CA값들은 각 vertex의 인접 vertex들 내에서 vertex id에 따라 정렬되게 된다. 예를 들어, vertex 71에 대한 CA값은 해당 vertex의 인접한 vertex인 8, 13, 23, 54이다. 이 값에 대해 기존 CSR

format을 적용시킨다면 vertex 71에 대한 값이 8, 13, 54, 23로 정렬이 되지 않은 상태로 메모리에 접근하게 되어 cache locality 문제가 발생한다. 하지만 Reorder CSR format을 사용해 vertex들을 id에 따라 정렬한다면 8, 13, 23, 54로 정렬이 되어 순차적으로 메모리에 접근할 수 있게 된다.

이처럼 OA를 정렬한다면 vertex degree가 큰 vertex부터 메모리에 접근하므로 동일한 vertex를 메모리에서 접근할 확률을 최대화하여 temporal locality를 향상시킬 수 있다. 또한 CA의 정렬에 의해 메모리에서의 특정 vertex data 구간이 cache에 존재할 때 cache hit rate를 최대화 할 수 있어 spatial locality를 향상시킬 수 있다.

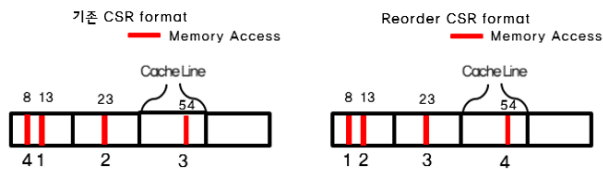


Figure 2 메모리 접근 비교

Figure 2를 보면 array 값 정렬을 한 후 그래프 알고리즘에 의해 각 vertex들을 메모리에서 접근할 때 접근 순서가 변경되어 특정 vertex data들이 cache line에 존재할 때 해당 vertex를 접근할 수 있어 cache locality를 향상시킬 수 있다.

만약 첫 번째 cache 구간이 vertex 8과 vertex 13이 해당되는 구간일 경우 기존 CSR format은 vertex 13을 접근할 때 cache 존재하므로 cache hit가 발생하지만 vertex 8은 마지막에 접근해야 하므로 해당 vertex는 cache hit를 하지 못한다. 하지만 Reorder CSR format의 경우 vertex id에 따라 값을 정렬한 상태이기 때문에 vertex 8을 메모리에 먼저 접근해 cache hit가 발생하고 두 번째로 vertex 13을 메모리에 접근하기 때문에 cache hit가 발생할 확률이 높아지게 된다. 이와 같이 CA값 정렬을 통해 vertex id를 정렬한다면 cache hit rate가 증가해 spatial locality를 개선 시킬 수 있다.

III. 고 찰

그래프 데이터의 representation 방식으로 가장 많이 사용되는 CSR format는 처리 성능과 관련된 데이터 지역성 문제를 가지고 있다. 성능 향상을 위해 다양한 기법들이 제안되고 있으며, 이 제안들에서는 재정렬(reordering) 기법이 적용되고 있어 이에 대한 이해를 기반으로 효과를 분석하는 것이 필요하다. Reordering, 즉 재정렬 기법을 적용한 CSR format은 기존 CSR format 방식을 유지하면서, 재정렬된 그래프 정보 배열(OA 및 CA)의 접근에 대해 성능향상을 기대할 수 있다. 성능향상의 상충요소로서 재정렬을 위한 오버헤드가 존재하기 때문에 이를 개선하기 위한 요소가 재정렬 기법의 적용에 고려되어야 한다.

ACKNOWLEDGMENT

"본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학사업의 수행결과로 추진되었음"(2015-0-00908)

참 고 문 헌

- [1] nvGRAPH, JUN. 2020, [online] Available: <https://developer.nvidia.com/nvgraph>
- [2] Wang, Y., Davidson, A., Pan, Y., Wu, Y., Riffel, A., & Owens, J. D. (2016, February). Gunrock: A high-performance graph processing library on the GPU. In Proceedings of the 21st ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming. 2015 pp. 1-12.
- [3] Apache Giraph, JUN. 2020, [online] Available: <http://giraph.apache.org/>.
- [4] Gharaibeh, ES-N. Abdullah, Lauro Beltrao Costa, and M. Ripeanu. "Totem: Accelerating graph processing on hybrid cpu+ gpu systems." GPU Technology Conference. 2013.
- [5] Vignesh Balaji, Brandon Lucia, "When is Graph Reordering an Optimization?" Carnegie Mellon University, 2018.
- [6] J. Arai, H. Shiokawa, T. Yamamuro, M. Onizuka, and S. Iwamura, "Rabbit order: Just-in-time parallel reordering for fast graph analysis," in Proceedings of the International Parallel and Distributed Processing Symposium, 2016
- [7] Vignesh Balaji, Brandon Lucia, "Combining data duplication and graph reordering to accelerate parallel graph processing," in Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing. 2019. p. 133-144.