

GPU 기반 대규모 시계열 데이터 처리 시스템의 효율적 동작을 위한 스케줄링 기법

김성섭, 이승우, 송민환, 권영민, 이상신
한국전자기술연구원

e-mail:sskim@keti.re.kr, seungwoo.lee@keti.re.kr, mhsong@keti.re.kr,
ymkwon@keti.re.kr, sslee@keti.re.kr

Scheduling Technique for Efficient Computing of GPU-based Large-scale Time Series Data Processing Systems

Seongseop Kim, Seungwoo Lee, Min-Hwan Song, Youngmin Kwon, Sang-Shin
Lee
Korea Electronics Technology Institute

요약

GPU 기반 시스템에서 대규모 시계열 데이터를 효율적으로 처리하기 위해 본 연구에서는 그래프 기반 최단 경로 탐색 알고리즘을 이용한 스케줄링 기법을 제안한다. 제안된 방법은 생성된 IoT 빅데이터를 활용하여 GPU 기반 시스템에서 분석되었다. 그 결과, 제안한 스케줄링 알고리즘이 시간 및 처리의 계층구조로 인해 복잡한 대규모 시계열 데이터 처리 시스템의 최적 스케줄링에 효율적으로 사용될 수 있음을 확인하였다.

1. 서론

사물 인터넷에서 발생하는 센서 데이터의 경우 많은 데이터의 양으로 인해 많은 연산량이 요구된다. 특히, 항만 등 특수한 환경에서의 IoT 시스템의 여러 이중 시스템에서 얻어지는 시계열 데이터의 경우 연산량뿐만 아니라, 서버 측으로 전송되는 데이터의 시점 및 연산 방법이 각기 달라 각 시스템에서 얻어지는 데이터의 최적 연산 방법을 찾는 데 어려움이 존재한다.

본 연구에서는 이런 문제를 해결하고자 여러 시스템에서 얻어진 시계열 데이터의 효율적인 처리를 위한 GPU 기반 시스템의 대규모 데이터 처리 스케줄링 기법을 제안한다. 실험을 위해 시간에 따라 계층적인 구조를 가지도록 시스템별 모의 센서 데이터 및 GPU에서 특정 결과를 얻기 위한 모델을 생성하였고 제안하는 알고리즘의 결과를 분석하였다. 그 결과 제안하는 구조의 스케줄링 결과로부터 GPU 기반의 대규모 시계열 데이터 처리 시스템의 효율적 동작을 찾을 수 있음을 확인하였다.

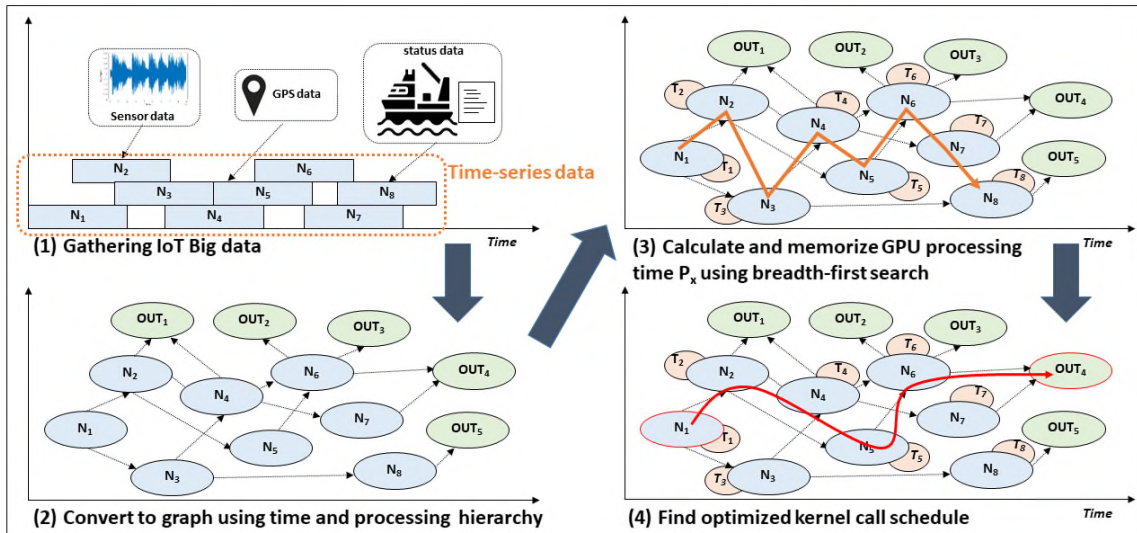
2. 본론

사물 인터넷에서 얻어지는 데이터를 처리하기 위한 자원 분배 연구들은 꾸준히 진행되었으며[1], GPU를 사용하는 시

스템의 컴퓨팅 자원 분배 연구들도 진행되었다[2]. 본 논문에서는 사물 인터넷 시스템에서의 효율적인 연산을 위해 GPU를 포함하는 시스템의 효율적인 연산을 위한 스케줄링 기법을 제안한다. 일반적으로 GPU를 사용한 연산의 경우 반복되는 연산에 장점을 가져 대규모 센서 데이터 등 대규모 처리에 적합하나 GPU 메모리로 데이터를 복사하는 오버헤드가 존재한다. 따라서, 본 논문에서는 특정 결과를 얻기 위해 최적의 연산 순서를 정하여 불필요한 데이터의 복사 및 연산을 줄이는 구조를 제안한다. 여러 시스템에서 얻어진 대규모 시계열 데이터들을 시간, 처리 방법의 계층구조로부터 그래프 구조로 변환시켜 각 데이터 집합 별 처리 속도를 추정하여 특정 결과를 얻기 위한 최적의 연산 순서를 찾는 GPU 커널 스케줄링 구조를 구현하였고 그 동작을 분석하였다.

3. 구현

제안하는 연산량 탐색 기반 GPU 시스템의 최적 스케줄링 구조는 그림 1. (a)와 같다. 여러 분산 시스템에서 얻어진 빅데이터들을 서버 측으로 모은 후, 해당 데이터 집합의 수집 시간, 처리를 위한 계층구조 등을 토대로 그래프 구조로 변환한다. 그 후, 모든 노드 N_x 를 탐색하며 데이터 집합 별 GPU 커널 호출 시의 실행 시간 T_x 를 저장한다. 저장된 실행 시간 T_x 만을 바탕으로 특정 출력 OUT_x 를 얻기 위한 최적의 처리



[그림 1] 제안하는 구조

```

Algorithm 1 Proposed scheduling method
for Node do
  Queue = Enqueue(Node_start)
  while Queue do
    Node_x ← Dequeue(Queue)
    for k = 1, 2, 3, ..., i ← block index, m ←
    thread index do
      Kernel <<< i, m >>> for Node_x
      syncthreads();
      T_x = T_k + T_c,
      (where, T_k = kernel elapsed time, T_c =
      Memory copy time (device, host) for Node_x)
    end for
    while Node_x → next do
      Enqueue(Node_x → next)
    end while
  end while
end for
for T_x do
  S_x = shortest processing time in Node
end for
    
```

[그림 2] 동작 알고리즘

4. 결론

GPU 기반 시스템에서의 효율적인 대규모 시계열 데이터 처리를 위해 본 논문에서는 그래프 기반 최단 경로 탐색을 통한 스케줄링 기법을 제안하였다. 제안하는 기법을 모의 데이터와 실제 GPU를 포함하는 시스템에서 동작을 확인하였고 분석하였다. 그 결과 제안하는 구조를 통해 복잡한 시간-연산 계층구조를 가지는 대규모 시계열 데이터 시스템의 최적 스케줄링에 충분히 사용 가능함을 확인하였다. 추후 연구로는 실제 대규모 시계열 데이터를 사용하여 그 동작을 분석하고 보완해나갈 예정이다.

ACKNOWLEDGEMENT

이 논문은 2020년 해양수산부 재원으로 해양수산과학기술진흥원의 지원을 받아 수행된 연구임(스마트 항만 IoT 인프라 구축 기술 개발)

참고문헌

[1] K. Nishii, Y. Tanigawa and H. Tode, "A Search Method of Large-Scale Resources for Providing Efficient Computing on a Participating Fine-Granular Cloud Computing Platform," 2018 IEEE 7th International Conference on Cloud Networking (CloudNet), pp. 1-4, 2018

[2] Z. Tasoulas and I. Anagnostopoulos, "Kernel-Based Resource Allocation for Improving GPU Throughput While Minimizing the Activity Divergence of SMs," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, no. 2, pp. 428-440, 2020

방법을 최적 경로 탐색을 통해 찾아내어 GPU 사용의 최적화를 구하도록 하였다. 제안하는 동작의 알고리즘은 그림 1. (b)와 같으며, 상세 동작은 아래와 같다.

연산 계층 구조를 변환한 그래프 구조의 탐색을 위해 너비 우선 탐색을 사용하였으며, 탐색 과정 중 GPU 커널을 실제 호출하며 CUDA를 이용한 GPU 실행 시간 T_x 를 노드 별로 저장하도록 하였다. 그 후, 저장된 T_x 와 그래프의 연결 구조를 바탕으로 출력을 얻기 위한 스케줄링 방법을 최단 경로 탐색 알고리즘인 Dijkstra 알고리즘을 통해 찾도록 구현하였다. 구현한 구조의 실험을 위해 대규모 모의 센서 데이터, 시스템의 상태 데이터 등을 사용하여 Geforce GTX 1080 Ti 기반의 시스템에서 그 동작을 확인하였고, 특정 출력을 얻기 위한 최적의 GPU 커널 스케줄링이 가능함을 확인하였다.