

# 데이터 스트림 연속 처리 태스크의 병렬 처리 방법

양경아, 이대우, 김기현  
ETRI 부설연구소 미래연구센터  
{kayang, neigh, khkim}@nsr.re.kr

## The Parallel Processing Method of the Continuous Data Stream Processing Tasks

Kyungah Yang, Daewoo Lee, Kiheon Kim  
Future Research Center, The Attached Institute of ETRI

### 요 약

데이터 스트림 환경에서 연속 질의를 처리하기 위한 데이터 스트림 처리 시스템이 개발되었다. 데이터 스트림 처리 시스템에서 질의를 처리하는 태스크에 과도한 데이터가 발생할 경우 일반적으로 데이터 스트림을 선별적으로 버리는 load shedding 방법을 이용하지만 이러한 방법은 처리 결과의 정확도가 저하될 수 있다. 따라서, 본 논문은 이를 해결하는 방법으로 분산 데이터 스트림 처리 시스템에서 데이터 스트림 분할을 통한 데이터 스트림 연속 처리 태스크의 병렬 처리 방법을 제시한다. 이를 위해 분산 데이터 스트림을 처리하기 위한 기준을 제시 및 데이터 분할 방법에 대해서 언급한다.

### 1. 서론

금융이나 제조공정 관리, 시스템 모니터링, 소셜 네트워크 등 다양한 분야에서 빠르게 연속적으로 새로운 실시간 데이터가 무수히 발생하고 있다. 이런 데이터 스트림 환경에서 연속 질의를 처리하기 위해 데이터 스트림 처리 시스템이 개발되었다. 데이터 스트림 처리 시스템은 연속적으로 입력되는 데이터를 처리하기 위한 태스크(또는 연산)로 구성된다. 특히, 비주기적으로 입력되는 대용량 데이터 스트림을 처리하기 위해 다수의 노드를 이용하여 질의를 분산 처리할 수 있는 분산 데이터 스트림 처리 시스템이 개발되었다. 분산 데이터 스트림 처리 시스템은 데이터 스트림에 대한 질의 처리를 위해 질의를 구성하는 여러 태스크를 하나 이상의 노드를 활용하여 분산하여 처리한다 [1].

분산 데이터 스트림 처리 시스템으로 입력되는 데이터 스트림의 폭증으로 인해 특정 태스크를 단일 노드에서 처리하기 힘든 경우가 발생하며, 이로 인해 연속 질의 처리가 지연되거나 오류가 발생하거나 최악의 경우 분산 데이터 스트림 처리 시스템이 정지하는 상황이 발생할 수 있다 [2].

이러한 문제를 처리하기 위해 기존의 데이터 스트림 처리 시스템에서는 데이터 스트림을 선별적으로 버리는 load shedding 방법을 사용하고 있지만 해당 방법은 처리 결과의 정확도를 저하시킬 수 있다[3].

따라서, 본 논문은 이를 해결하는 방법으로 분산 데이터 스트림 처리 시스템에서 데이터 스트림 분할을 통한 데이터 스트림 연속 처리 태스크의 병렬 처리 방법을 제시한다. 이를 위해 병렬 처리에 앞서 병렬 처리의 적용 여부를 결정하는 기준과 데이터 분할 방법에 대해 소개한다.

본 논문의 구성은 다음과 같다. 2장에서는 분산 데이터 스트림 처리 시스템에서 데이터 스트림 분할을 통한 연속 처리 태스크의 병렬 처리 방법의 기준에 대해 기술

한다. 3 장에서는 데이터 스트림을 병렬 처리하기 위한 분할 방법에 대해 설명한다. 4 장에서는 결론 및 향후 연구에 대해서 기술한다.

### 2. 분산 데이터 스트림 처리

분산 데이터 스트림 처리 시스템에서 단일 노드로 입력되는 대용량의 데이터 스트림 처리는 해당 노드의 메모리 과 부하 및 처리 지연 등으로 인하여 전체 질의 처리 성능을 크게 저하 시킬 수 있다. 이런 경우, 데이터를 분할하여 병렬 처리하는 것이 필요하다. 데이터 스트림 연속 처리 태스크의 병렬 처리에 앞서 우선 병렬 처리가 필요한지 판단해야 한다.

병렬 처리의 필요 여부 판단은 데이터 손실이 없어야 되는 경우, 즉 최대한 모든 데이터를 처리해야 하는 경우와 비용 상에 이득이 존재하는 경우로 판단할 수 있다. 비용 상의 이득이 있을 경우는 일정한 데이터 량에 대해 특정 태스크를 단일 노드에서 처리하는 비용보다 다수의 노드에서 병렬 처리하는 비용을 비교하여 판단한다.

$$\sum_{k=1}^n D' + \sum_{k=1}^n C' + A' > \sum_{k=1}^n C$$

위의 수식에서 보듯이, 데이터 스트림 연속 처리 태스크의 병렬 처리는 일정한 데이터량(n)을 처리 할 때, 단일 노드에서 해당 데이터를 처리하는 비용보다 다수의 노드에서 병렬 처리하는 비용이 작은 경우에 수행된다.

수식에서 단일 노드의 처리 비용은 해당 노드의 데이터 처리 비용 C이다. 이때, C는 단일 노드에서 처리로 인한 메모리 과부하 및 처리 지연 등으로 인한 소모되는 모든 비용을 포함한다. 수식에서 다수 노드에서의 병렬 처리 비용은 해당 노드의 데이터 분할 전송 비용, D' 해

당 노드들의 데이터 처리 비용  $C'$ , 처리 결과를 통합하는 비용  $A'$  을 합산한 비용이다.

데이터 스트림 연속 처리 태스크의 병렬 처리가 필요하다고 판단되는 경우, 정의된 데이터 스트림 분할 요소, 데이터 스트림 분할 방법, 데이터 스트림 분할 배치 방법, 연속 처리 태스크 할당 방법, 처리 결과 전달 방법에 따라 분산 데이터 스트림 처리 시스템에서 데이터 스트림 연속 처리 태스크의 병렬 처리를 수행하고 그 결과를 전달한다.

분산 데이터 스트림 처리 시스템에서 데이터 입력 시마다 데이터 스트림 연속 처리 태스크에 대한 병렬 처리의 지속 여부를 판단할 수 있으나, 병렬 처리의 지속 여부에 대한 판단을 내리기 위해 필요한 연산 비용이 클 수 있으므로 분산 데이터 스트림 처리 시스템의 스케줄링 및 최적화를 담당하는 부분에서 주기적으로 판단하여 병렬 처리가 필요 없을 때까지 다음 입력 데이터에 대해 연속적으로 병렬 처리 및 전달을 수행할 수 있도록 한다.

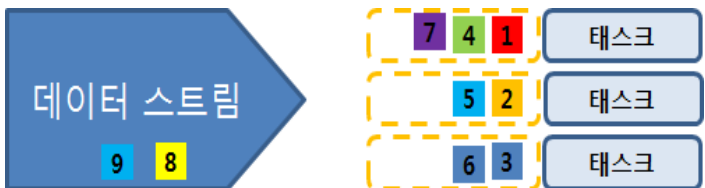
### 3. 데이터 스트림 분할 방법

데이터 스트림 분할 요소는 데이터 스트림 분할의 최소 단위를 의미한다. 데이터 스트림 분할 요소는 입력되는 데이터 스트림의 최소 단위인 레코드나 태스크 처리의 기본 단위인 윈도우를 기준으로 분할 가능하다. 이때, 레코드의 경우는 해당 태스크가 레코드 단위의 처리가 가능하게 작성된 경우로 제한한다.

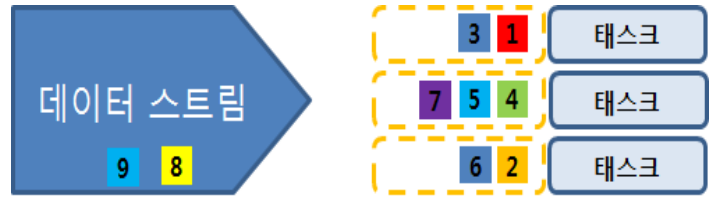
데이터 스트림 분할 배치 방법은 데이터 스트림의 분할 요소를 실제 수행할 태스크로 배치하는 순서를 지정하는 것으로써 이는 향후 태스크의 처리 결과 전달에 영향을 줄 수 있다. 연속 처리 태스크 할당 방법은 분할된 데이터 스트림을 수행할 병렬 태스크를 선정 및 할당하기 위한 방법을 결정하는 것이다.

병렬 처리를 위한 연속 처리 태스크 할당 방법은 태스크를 미리 분산 배치한 후 분할된 데이터 스트림을 전달하는 선배치 방법과 분할된 데이터 스트림을 저장한 후 병렬 태스크를 배치하는 후배치 방법이 있다.

데이터 스트림 분할 선배치 방법은 (그림 1)과 같이 데이터 스트림을 입력된 순서대로 병렬 태스크에 배치하는 방법과 (그림 2)과 같이 입력된 순서와 상관없이 병렬 태스크에 배치하는 방법이 있다. 데이터 스트림 분할 배치 방법은 태스크의 특성에 따라 태스크를 작성한 사용자가 정의한다.

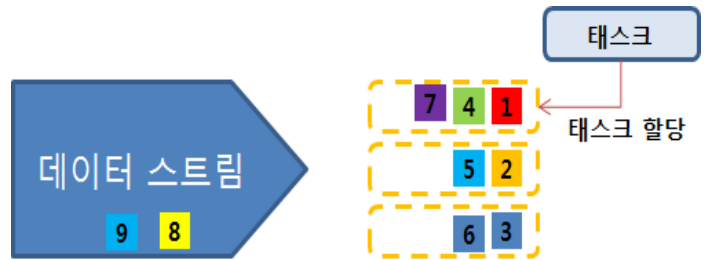


(그림 1) 입력순서 기반 선배치 방법



(그림 2) 입력순서와 무관한 선배치 방법

(그림 3)은 후배치 방법을 보여준다. 후배치 방법은 윈도우 단위로 처리되는 스트림에서 윈도우가 가득차거나 윈도우의 특정 크기에 도달했을 때 태스크를 배치하여 처리한다.



(그림 3) 후배치 방법

선배치 방법은 후배치 방법에 비해 구현이 간단하며, 처리 속도가 빠르다. 후배치 방법은 태스크 배치 전에 분할된 데이터 스트림을 저장하고 태스크를 배치하여 처리해주는 부분에 대한 고려가 필요하나, 선배치 방법에 비해 확장성이 뛰어나며 노드의 자원의 활용도가 높다. 연속 처리 태스크의 할당 방법은 자원 활용도 및 처리 속도 등을 고려하여 사용자가 정의한다.

### 4. 결론 및 향후 연구

본 논문에서는 분산 데이터 스트림 처리 시스템에서 특정 노드에 할당된 연산에 업무가 과중한 경우 해당 데이터 스트림 분할을 통한 태스크를 병렬 처리하는 방법을 제시함으로써 대용량 데이터 및 질의의 과부하로 인하여 특정 연산에 집중되는 부하를 분산할 수 있다. 이러한 부하 감소로 인하여 실시간 데이터 스트림의 실시간 처리 보장 및 데이터 shedding으로 인한 데이터 손실을 줄일 수 있는 이점이 있다. 향후에는 시험을 통해 성능을 검증이 필요하다.

### 5. 참고문헌

[1] 이현호, 이원석, 데이터 스트림 상에서 다중 연속질의 처리를 위한 속성기반 접근 기법, 정보처리학회논문지, 제 14권 5호, pp.459-470 2007.  
 [2] 한승호, 김명진, 최운, 이한구, 클라우드 환경에서의 분산 미디어 서버를 위한 스트리밍 태스크 분배 알고리즘, 한국정보과학회 학술대회 논문집, pp.1337-1339 2013.  
 [3] Tatbul, E.N. Load Shedding Techniques for Data Stream Management Systems, Ph.D. thesis, Brown University, 2007.