

# ESPER 엔진 기반의 CEP 모델 설계 및 평가

김환\*, 황부현\*  
 \*전남대학교 전자컴퓨터공학부  
 e-mail: sentual7@gmail.com

## CEP Model Design and Revaluation Using ESPER Engine

Hwan Kim\*, Buhyun Hwang\*  
 \*Dept. of Electronics and Computer Engineering, Chonnam National University

### 요 약

IT 기술이 성장함에 따라 발생하는 데이터의 양이 많아지고 기존의 방식으로 처리하기 힘든 빅 데이터가 이슈가 되고 있으며 이에 따른 대용량 데이터 처리기술 또한 발전하고 있다. 이 논문에서는 실시간 대용량 데이터 처리를 위한 Complex Event Processing을 소개하고 ESPER 엔진 기반의 Complex Event Processing 모델을 설계하고 이에 대한 성능을 평가했다.

### 1. 서론

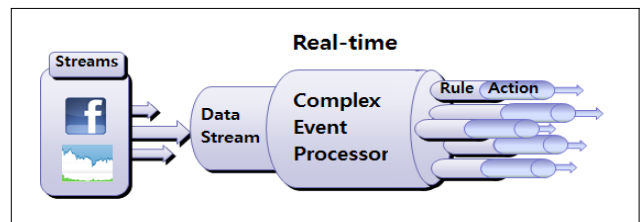
최근 IT 기술의 발전과 스마트폰, 태블릿 PC 등의 사용 증가로 모바일 환경이 확대되고 있다. 이로 인해 기업에서 생성하는 데이터양 뿐 아니라 소셜 네트워크 서비스(SNS : Social Network Service)를 통해 생성되는 데이터 양이 폭발적으로 증가하고 있다[1]. SNS 중 하나인 Twitter의 경우 매일 약 1억 1,000만 개의 메시지가 발송되고 있다.

빅 데이터란 데이터의 크기가 크고, 빠르게 변하거나 구조화하기 힘들어 기존의 방식으로 처리하기 힘든 데이터이다[2]. 빅 데이터를 처리하기 위한 방식으로 맵 리듀스(Map Reduce)와 하둡(Hadoop)과 같은 분산 파일 시스템이 활용되고 있다. 하지만 실시간 대용량 데이터 분석 기술이 주목받으면서 CEP 엔진 또한 주목받고 있다. 이 논문에서는 ESPER 기반의 CEP 모델을 설계하고 성능을 평가했다. 논문의 구성은 다음과 같다. 2장에서는 CEP 기술을 소개하고 3장에서는 ESPER 기반의 CEP 모델을 제시한다. 4장에서는 CEP 엔진 성능 평가 환경과 결과를 기술하고 5장에서는 결론 및 향후과제를 기술한다.

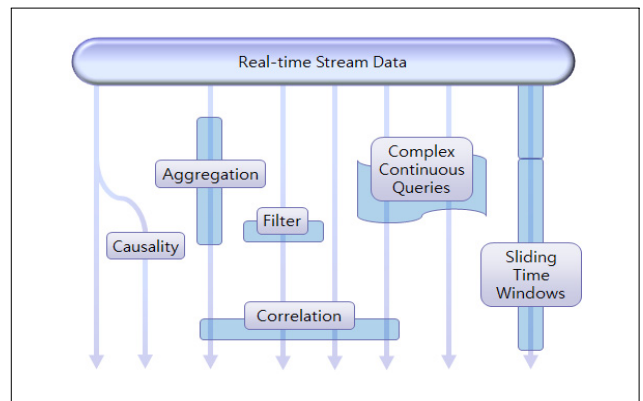
### 2. CEP(Complex Event Processing)

CEP(Complex Event Processing)는 복합적인 Event에 대한 실시간 대용량 데이터 처리 기술이다. CEP는 (그림 1)과 같이 입력되는 스트림 데이터를 미리 등록된 질의에 따라 이벤트 처리를 수행한다.

CEP는 (그림 2)와 같은 이벤트 처리가 가능하다. 특정 조건을 만족하는 항목을 통과시키는 Filter 기능. 입력 값들에 대한 Aggregate 함수를 이용한 결과 값 생성. 구간별 윈도우를 생성하는 Sliding Time Windows. 입력 값들 사이의 관계를 찾아내는 Correlation 등의 데이터 처리가 가능하다.



(그림 1) CEP 개념도



(그림 2) CEP의 이벤트 처리 종류

시스템에 질의(Query)를 미리 등록하기 위해 사용하는 CQL(Continuous Query Language)는 기존의 SQL 구문과 유사한 구조를 가지고 있지만 아직 표준은 정해져 있지 않다[3][4].

### 3. CEP 모델 설계

실제 응용 시스템에서 발생할 수 있는 스트림 데이터의 이벤트 처리를 위해 ESPER 엔진 기반의 CEP모델을 설계했다.

3.1 데이터 생성

데이터는 (symbol, price, timeStamp) 3가지의 속성 값을 가지고 있으며 Java의 Random generator를 사용하여 생성하였다.

3.2 질의(Query) 등록

조건을 만족하는 데이터에 대한 이벤트 처리를 위해 ESPER 엔진에서 사용하는 CQL인 EPL(Event Processing Language)을 사용하여 질의(Rule)를 등록하였다. 성능 평가에 사용된 질의는 다음과 같다[5].

```
"select * from "
+ StockTick(symbol='Z').win:length(2) "
+ "having avg(price) > X";
```

Z라는 품목에 대해 최근 두 항목의 price 평균이 X보다 클 경우 이벤트 처리를 수행

```
"select * from "
+ StockTick(symbol='Z').win:length(1) "
+ "having price < Y";
```

Z라는 품목에 대해 현재 price가 Y 보다 작을 경우 이벤트 처리를 수행

3.3 이벤트 처리

등록된 질의에 만족하는 데이터가 감지되면 해당 데이터에 대한 이벤트 처리(Action)를 수행하는 monitor 함수를 정의하였다. 성능 평가에 사용된 이벤트 처리는 등록된 질의를 만족시키는 데이터의 발생 횟수를 출력한다.

4. CEP 모델 평가

4.1 성능 평가 환경

ESPER 엔진을 활용하여 CEP의 데이터 처리 성능을 평가하기 위해 <표 1>의 시스템을 사용하였다.

항 목	사 양
C P U	Intel(R) Core(TM)2 Quad
메 모 리	4GB
운영체제	Windows7 Ultimate K
프로그래밍	Java 1.6.0_30

<표 1> 성능 측정 시스템 환경

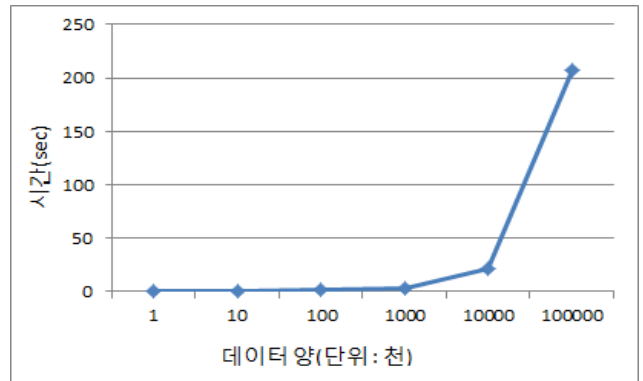
4.2 성능 평가

성능 평가는 데이터의 수와 이벤트 처리를 위한 질의의 수를 늘리면서 프로그램 수행시간을 측정했다. 수행시간은 초 단위로 나타내고 소수점 셋째자리에서 반올림하여 둘째자리 까지 표기하였다.

4.2.1 입력 데이터 양 증가

입력되는 데이터양을 1000, 10000, 100000, 1000000, 10000000, 100000000로 10배씩 증가시키면서 프로그램 수행시간을 측정했다. (그림 3)는 데이터양이 증가함에 따라 이벤트 처리를 수행하는 시간을 나타내고 있다. 1~1000천

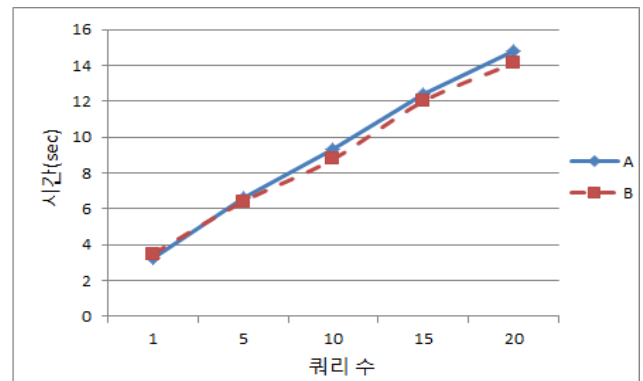
개 데이터 사이에서는 1~3초로 짧은 시간에 이벤트 처리가 가능하였다. 1억 개의 데이터 경우 수행 시간이 급격하게 증가했지만 초당 약 483512개의 데이터를 처리가 가능했다.



(그림 3) 데이터양에 따른 이벤트 처리 시간

4.2.2 질의 수 증가

시험 A는 1000000개의 데이터를 기준으로 다양한 이벤트 처리를 위해 질의의 수를 늘리며 수행시간을 측정하였다. 시험 B는 미리 20개의 질의를 등록해 두고 입력 데이터의 이벤트 처리에 영향을 미치는 질의의 수를 늘리며 수행시간을 측정하였다.



(그림 4) 질의 수에 따른 이벤트 처리 시간

(그림 4)에서 A는 입력 데이터에 대해 다양한 이벤트 처리를 위해 등록된 질의의 수를 늘려감에 따라 수행 시간이 유사한 기울기로 증가함을 볼 수 있다. B에서는 미리 20개의 질의를 등록해 두었지만 데이터 이벤트 처리에 관여하는 질의가 증가함에 따라 수행시간이 A와 유사하게 증가하였다. 즉, 입력되는 데이터의 이벤트 처리에 관여하는 질의의 수가 증가함에 따라 수행시간이 영향을 받을 수 있다.

5. 결론

실시간 대용량 데이터 처리를 위해 ESPER 엔진 기반의 CEP(Complex Event Processing) 모델을 설계하고 성능을 평가해 보았다. 실제 응용환경에서 사용자가 필요한 정보가 한정되어 있음에 따라 등록되는 질의의 수도 제한

될 것이다. 또한 4.2.2의 시험 결과에 따라 데이터의 이벤트 처리에 관여하는 질의 만이 수행시간에 영향을 주었기 때문에 실제 응용 분야에서 빠르고 효율적인 대용량 데이터 처리가 가능할 것이다. 이 논문의 CEP 모델에서는 단일 스트림 데이터에 대한 처리를 다루었다. 향후 다양한 종류의 복합적인 스트림 데이터를 처리할 수 있는 CEP 모델 분석 및 설계가 필요하다.

### 참고문헌

- [1] 이만재. “빅 데이터와 공공 데이터 활용”, Internet and Information Security, 제2권, 제3호, pp.47~64, 2011.
- [2] 채승병 “정보홍수 속에서 금맥 찾기 : 빅 데이터(Big Data) 분석과 활용” 삼성경제 연구소.
- [3] 강만모 “차세대\_웹 환경에서 Complex Event Processing 엔진을 이용한 대용량데이터 처리” 정보과학회 논문지 2010.
- [4] S. R. Madden, M. A. Shah, J. M. Hellerstein, and V. Raman. “Continuously Adaptive Continuous Queries over Streams”, Proc. of ACM SIGMOD 2002, Madison, Wisconsin, United States, 2002.
- [5] ESPER EPL Reference :  
<http://esper.codehaus.org/esper-4.5.0/doc/reference/en/html>