

분산 스트림 컴퓨팅 기술 동향

Technology of Distributed Stream Computing

이미영 (M.Y. Lee) 데이터베이스연구팀 팀장

목 차

-
- I . 개요
 - II . 데이터 처리 환경의 변화
 - III . 스트림 컴퓨팅 기술
 - IV . 결론

데이터의 효과적인 활용이 경쟁력 확보에 주요한 요인이나, 데이터 폭증은 유용한 정보를 얻는데 필요한 처리 시간의 지연을 야기하고 있다. 개인 맞춤형 서비스, 방법·방재 서비스 등 모니터링 & 대응 서비스를 위해 분석할 데이터의 양이 급증하고 있으며, 텍스트, 영상, 오디오 등 비정형 데이터에 대한 실시간 분석 필요성이 증대하고 있다. 대량의 폭증하는 데이터에 대한 실시간 분석 처리 환경을 제공하기 위해 분산 병렬 컴퓨팅 기술과 데이터 스트림 연속 처리 기술이 활용되고 있다. 본 고에서는 폭증하는 데이터 스트림 처리를 위하여 확장성 및 유연한 처리 환경을 제공하는 분산 스트림 컴퓨팅 기술에 대해 소개한다.

I. 개요

스마트폰, CCTV, RFID, 센서 등 새로운 IT 환경은 사람, 사물, 환경의 스마트화를 가속시키고 있으며, 데이터 생성 및 소비에 있어서 많은 변화를 주고 있다. 데이터의 폭증을 야기하고 있으며, 실세계에서 일어나는 상황 정보가 바로 디지털화되고 있음에 따라, 최신 데이터 기반의 서비스를 요구하고 있다.

온라인 비디오, 디지털 사진 서비스, 이메일, 소셜 네트워킹 서비스 등을 통해 개인들은 엄청난 양의 디지털 정보를 생산하고 있으며, 데이터 폭증의 원인이 되고 있다. IDC와 EMC의 자료 조사에 의하면[1], 2009년 디지털 데이터는 80만 PB로 전년 대비 62% 성장했고, 2010년에는 1.2ZB에 도달할 것으로 예측하고 있으며, 생성 데이터의 대부분이 영상, 음성 등의 비정형 데이터이다.

데이터의 적재적소적시 활용이 기업의 경쟁력 확보의 주요한 요인이나, 데이터량의 증대는 유용한 정보를 얻는 데 필요한 처리 시간의 지연을 야기하고 있고, 이를 해결하기 위해 분산 병렬 컴퓨팅 기술의 활용이 커지고 있다. 또한 최신 데이터를 기반으로 전략 수립, 의사 결정 등을 수행하기 위해서 실시간으로 데이터를 처리하여 데이터 처리 지연 시간을 최소화하려는 노력들이 가속화되고 있고, 이를 가능하게 하는 기술로 데이터 스트림 연속 처리 기술이 중요해지고 있다.

본 고에서는 분산 병렬 처리 컴퓨팅 기술과 데이터 스트림 연속 처리 기술이 통합된 분산 스트림 컴퓨팅 기술에 대해 소개한다. II장에서는 데이터 처리 환경의 변화, III장에서는 스트림 컴퓨팅 기술 동향에 대해 기술하고, 마지막으로 결론을 맺기로 한다.

II. 데이터 처리 환경의 변화

1. 데이터 폭증

최근 IT 환경은 다양한 종류의 디지털 데이터를 생성하고 있으며, 이의 효율적인 활용이 요구되고 있다.

- 데이터 종류: 감시 카메라에서 생성하는 동영상, 개인이 디지털 카메라로 생성하여 웹 사이트에 올리는 사진, SNS를 통해 전달되는 메시지, 물건에 부착된 혹은 주변 환경에 설치된 센서로부터 발생하는 RFID 태그나 센서 값 등 처리해야 할 데이터의 형태가 숫자, 문자, 텍스트, 영상, 음성 등 다양해지고 있다.
- 데이터 크기: 감시 카메라에서 생성되는 수 GB 동영상, 디지털 카메라에서 생성되는 수백 KB 영상, 수십 바이트의 SNS 단문, RFID 기반 응용에서 발생하는 128bits RFID 태그 등 생성되는 데이터의 크기가 다양해지고 있다.

이와 같은 데이터들은 개인이 직접 생성 관리하기도 하고, IT 기반의 생활 환경에 따라 개인의 의지와 상관없이 자동 생성 관리되기도 한다. 개인에게 맞는 맞춤형 서비스 제공, 기업의 비즈니스 전략 수립, 공공 안전이나 재난 방지 등의 서비스에 활용 가능하며, 다양한 정보 소스로부터 얻은 정보들간의 융합을 통해 더 나은 서비스 구축이 가능하다.

2. 실시간 분석

가트너 분석 보고서에 따르면[2], 정보 분석 기술은 앞으로 5~10년 이내에 웹 콘텐츠 분석, 음성 데이터 마이닝, 대화 분석이 주요 기술이 될 것으로 예측하고 있으며, 2~5년 내에 메모리 기반 분석, SaaS 기반 분석 서비스 등이 주요 기술이 될 것이라고 예측

하고 있다. 텍스트, 영상, 음성, 동영상 등 콘텐츠 자체 분석을 위해서는 많은 양의 컴퓨팅이 필요하므로 분산 병렬 처리에 대한 요구도 증대할 것으로 예측된다. 또한 메모리 기반 분석에 대한 요구가 증대한다는 것은 그만큼 빠른 분석에 대한 필요성도 커지고 있다는 의미이다.

기업의 비즈니스 활동 모니터링(BAM), 인터넷 서비스 산업, 금융 산업, 보안 서비스 산업, 물류 유통 산업 등에서는 외부 환경 모니터링 및 대응 서비스가 널리 활용되고 있는 분야이다. 앞으로는 실시간 모니터링 & 대응이 더욱 중요해질 것이며, 한정된 환경에 대한 모니터링 & 대응이 아니라 융합을 통해 통합 환경에 대한 모니터링 서비스로 발전할 것으로 보인다.

예를 들어 서비스 콜 센터를 운영하는 기업에서는 서비스 콜 센터 업무가 정상적으로 잘 운영되고 있는지 확인하여, 문제가 있으면 이를 개선하기 위해 업무 상황을 모니터링한다. 시간 당 통화 수, 통화 시간을 분석하여, 고객의 불만이 쌓이지 않도록 서비스 업무를 실시간으로 조정한다. 뿐만 아니라 앞으로는 고객의 전화 내용 분석을 통해 어떤 문제로 전화를 하고 있는지를 파악하여, 이에 대한 조치를 빠른 시간에 반영할 수 있다[3].

SNS 업체, 인터넷 포털 검색 서비스 업체, 쇼핑몰 업체 등에서는 최신 트렌드 정보를 제공하기 위해, 웹 페이지의 광고를 클릭할 확률을 높이기 위해서 혹은 상품 구매를 촉진하기 위해 실시간 대응 체제를 구축하고자 한다. 최신 업로드 정보, 주요 클릭 페이지 등 사용자의 의도를 실시간으로 파악하여 대응 지연을 최소화하려고 하고 있다.

또한 주식 거래, 외환 거래 등 금융 분야에서는 거래 내역을 분석하여 적시에 시스템이 자동으로 거래를 수행하는 시스템 트레이드가 활용되고 있다. 이와 같은 시스템 트레이드에서도 단순한 거래 정보만 분

석하는 것이 아니라, 정치, 사회, 경제 뉴스 등 거래에 영향을 미칠 수 있는 정보들을 수집 활용하는 방향으로 발전할 것이다.

CCTV 설치 등 보안 서비스 분야에서는 국민의 안전을 위해 정부 차원에서 점차 확대 설치되고 있다. 현재는 촬영 영상을 모니터링 요원이 눈으로 확인하며, 문제 발생을 감지하거나, 문제가 발생한 후, 보관된 영상을 확인하여 사건 경위를 파악하는 용도로 활용되고 있으나, 점차 실시간으로 영상을 분석하여 위험을 사전에 감지, 대응 조치를 취할 수 있는 실시간 영상 감시 체제로 변화를 시도하고 있다.

III. 스트림 컴퓨팅 기술

지속적으로 발생하는 데이터의 실시간 처리를 지원하는 데이터 스트림 컴퓨팅 기술은 주로 정형 데이터 스트림 처리 응용 지원을 목적으로 나온 데이터 스트림 관리 시스템(DSMS) 기술과 대량의 비정형 스트림 데이터 처리 응용을 위하여 연구되고 있는 분산 스트림 처리 기술이 있다.

1. 데이터 스트림 관리 시스템

DSMS 기술은 주로 정형 데이터 스트림 처리 응용을 위하여 기존 데이터베이스 관리 기술과 차별화하여 발전한 기술이다. 데이터 저장 후 질의 처리 요구를 수행하는 기존 DBMS 방식 대신, 질의를 선등록 후 데이터가 들어오는 대로 등록된 질의를 연속 수행하는 기술이다. <표 1>에 DBMS와 DSMS에 대한 주요 차이점을 제시한다[4].

DSMS는 STREAMS, Borealis 등 대학의 연구 결과를 바탕으로 DBMS 업체들인 Oracle, Microsoft, IBM, Streambase 등에서 제공하고 있다. 주로

<표 1> DBMS와 DSMS 비교

	DBMS	DSMS
처리 대상	- 현재 저장 릴레이션	- 온라인으로 발생하는 스트림 데이터에 대한 윈도
데이터 저장소	- 용량 제한없는 디스크 - 랜덤 접근	- 용량이 제한된 메모리 - 순차적 접근
질의 형태	- 단발성 질의 - 한꺼번에 처리 - 정확한 처리 요구	- 연속 질의 - 윈도기반 점진적 처리 - 근사치 처리도 허용
데이터 변경	- 적은 편 - 트랜잭션 관리 중요	- 가변적이고 폭증 발생 - 별로 중요하지 않음
QoS	- 미지원	- 중요

<자료>: Sharma Chakravarthy 저서, 2009.

관계형 DBMS과 유사하게 관계형 데이터 모델을 기반으로 데이터 스트림에 대한 필터링, 집계, 조인 등 주요 스트림 처리 연산을 제공하며, 윈도 개념이 추가된 데이터 스트림 연속 처리 언어 등을 제공한다. SQL과 유사한 데이터 스트림 연속 처리 언어를 제공하므로 기존 DBMS 기반 응용 개발자들이 쉽게 데이터 스트림 처리 응용 구축이 가능하다는 장점이 있다. 온라인 금융 거래 정보 스트림, 센서 데이터 스트림, 위치 정보처럼 적은 크기의 정형 데이터가 지속적으로 발생하는 응용 분야에 많이 활용된다.

2. 분산 스트림 처리 시스템

텍스트, 영상 등 비정형 데이터에 대한 연속 처리가 필요한 응용 분야에서는 정해진 데이터 모델기반 연속 처리보다는 데이터 스트림 처리 로직을 응용 개발자가 자유롭게 구성 활용할 수 있어야 한다. 또한 데이터량에 대한 확장성을 제공할 수 있는 기반을 제공하기 위해 분산 컴퓨팅 인프라를 제공하는 데 초점을 맞추고 있다. 인텔, 야후 등에서 이에 대한 연구를 수행하고 있다.

분산 스트림 처리 시스템은 데이터 흐름에 따라 연속 처리, 분산 노드에 단위 업무의 분배 및 부하 분

산, 데이터 스트림 분할 및 통합에 의한 병렬 처리, 분산된 업무간의 데이터 스트림 전달 방법, 분산 노드의 장애에 대처하여 연속 서비스 제공 기술 등 분산 컴퓨팅 기술을 기반으로 스트림 연속 처리 기술을 통합하여 제공한다. 응용 분야에서는 이를 기반으로 응용에 맞는 데이터 모델 기반의 데이터 스트림 처리 연산을 통합하여 사용한다.

IBM 등 기존 DBMS 업체에서도 스트림 데이터의 폭증에 대한 확장성 제공 및 응용 로직의 유연한 통합을 위해 InfoSphere Streams라는 분산 스트림 컴퓨팅 기술을 제공하고 있다. InfoSphere Streams는 분산 스트림 처리 인프라를 기반으로 데이터 모델 기반 데이터 스트림 처리 연산이 통합되어 제공되는 분산 스트림 컴퓨팅 시스템이다[5].

가. Sprout

Sprout는 인텔에서 개발한 비디오 실시간 분석 시스템인 SlipStream의 하부 시스템으로 대량의 데이터에 대한 분산 연속 처리 시스템이다[6].

Sprout에서는 stage라는 처리 연산 단위와 stage 간의 관계를 정의하는 connector를 이용하여 데이터 스트림 처리 흐름을 DAG로 정의한다. Stage는 여러 개의 입출력을 가질 수 있고, 연산의 실행을 외부에서 제어할 수 있으며, 응용 개발자가 처리 로직을 정의한다.

Connector는 선행 stage의 출력 데이터를 후행 stage의 입력 데이터로 연결해주는 것으로, stage간의 데이터 흐름 관계를 정의한다. Connector는 데이터를 받을 stage가 어디에 있느냐에 따라 메모리 큐, TCP 전달 등이 활용된다. Connector는 데이터 폭증으로 큐가 늘어나 stage 수행 성능이 저하되는 것을 방지하기 위해, 선행 stage 수행 시기를 제어하여 데이터가 쌓이는 것을 방지한다.

```

<Application name="facedetect">
  <Module name="Scale" args=" ">
    <Inputs>In</Inputs>
    <Outputs>Out</Outputs>
    # define a continuous tuning knob with range 1-10
    <Tunable name="scale" type="continuous"
      best="1"> 1, 10</Tunable>
    <Stage class="ImageScaler">
  </Module>
  # other module specifications here

  <Connector source="FrameSource:Out"
    dest="Scale:In">
  <Connector source="Scale:Out"
    dest="Detect:In">
  # other connector specifications here

  <Servers exec="facedetect"># name of executable
    # hosts available to run this server
    <Hosts>nodeA, nodeB, nodeC</Hosts>
  </Servers>
</Application>
    
```

<자료>: Slipstream 논문

(그림 1) Sprout 응용 서비스 제출 예

데이터 분할 혹은 복제를 통해 동일한 stage의 병렬 처리가 가능하며, 병렬 처리된 stage의 입출력 데이터는 매핑 정책에 따라 connector에 매핑된다. 매핑 정책은 one-to-one, round robin, any, all 등 다양한 정책을 지원한다.

Sprout 시스템은 configuration server와 stage server들로 구성된다. Stage server는 stage 수행을 담당하고, 각각의 stage는 별도의 스레드로 동작한다. Configuration server는 응용을 수행시키는데 필요한 환경을 구성하고, 응용을 구성하는 stage들의 배치, 시작, 종료 등을 제어한다.

Sprout에서는 (그림 1)처럼 서비스 설정을 xml 형식으로 정의한다. 응용 서비스를 구성하는 모듈들, 모듈들 간의 상호 연결, 응용 서비스가 구동될 노드 정보들을 정의한 파일을 입력으로 configuration server를 구동하며, configuration server에서 해석하여 stage server를 구동 및 stage를 배치한다.

나. MapReduce Online

MapReduce Online은 버클리대학에서 대용량 데이터에 대한 일괄 처리를 지원하는 MapReduce[7],

[8]를 확장하여 개발한 대규모 데이터 연속 처리 시스템이다[9]. MapReduce Online은 Map 단계의 처리 결과 데이터를 Reduce 단계로 전달하는 과정에 파이프라인 기능을 적용한다. 파이프라인 기능은 Map 단계에서 일정량의 데이터 처리가 완료되면(Map 프로세스에 할당된 메모리 버퍼가 차면) 파일 시스템에 저장하는 것과 동시에 Push 방식으로 Reduce 단계로 전송하는 기능이다.

파이프라인 기능을 통해, 사용자는 MapReduce를 이용하는 것 보다 최종 처리 결과에 대한 응답 시간을 줄일 수 있다. 즉, Map 단계에서 처리한 마지막 메모리 버퍼의 내용만 전달되면 Reduce 단계의 다음 작업들이 진행될 수 있으므로, Map 단계가 모두 끝나고 전체 데이터를 전달하는 기존 MapReduce에 비해 처리 완료 시간이 빠르다. 또한 파이프라인 전달은 대량의 데이터 처리시 주기적으로 중간 처리 결과 제공을 가능하게 한다. 전체 데이터 처리 결과가 아니므로 정확한 정보는 아니라도 일부 데이터를 기반으로 근접 처리 결과를 미리 얻을 수 있다.

이와 같은 파이프라인 기능을 이용하여 MapReduce Online은 데이터가 계속 발생하는 환경에서 서비스를 연속적으로 수행하는 환경에 활용 가능하다.

다. S4

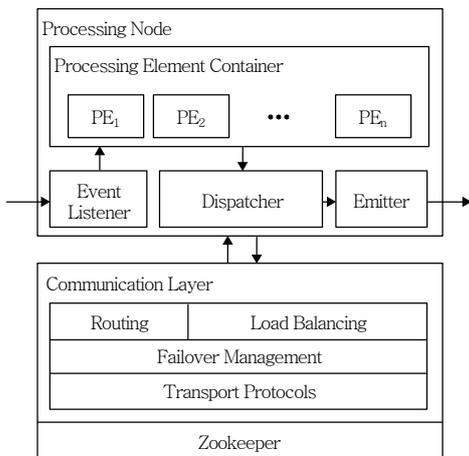
S4(Simple Scalable Streaming System)은 야후에서 개발 후 오픈소스화한 분산 스트림 처리 시스템으로 야후의 서비스 운영에 활용되고 있다[10],[11]. S4는 야후 검색 서비스 제공에 필요한 데이터 마이닝, 기계 학습 등을 지원하기 위해 개발된 시스템이다. 수백만 명의 사용자로부터 초당 수천 건의 질의 요청을 받는 웹 검색 서비스에서 사용자의 검색 요구를 실시간으로 분석하는 다양한 분석 알고리즘을 쉽게 배치 운영할 수 있으면서, 실제 현장에서 활용할

수 있도록 확장성을 제공하며 저지연 속도로 스트림 처리 제공을 목적으로 개발된 시스템으로, 클러스터 시스템 하에서 동작 가능하다.

S4는 actors 모델기반으로 연산을 수행하는 프로세싱 엘리먼트(PE)와 PE 사이에 전달되는 메시지인 이벤트 데이터로 구성된다. 이벤트 데이터는 타입을 가지며, (keys, attributes)로 구성된다. 이벤트 데이터는 키 값에 의해 전달될 PE가 결정된다. PE는 병렬 처리가 가능하도록 여러 개의 인스턴스가 구동되며, 이벤트 데이터 타입 및 이벤트 데이터 키 값에 의해 처리할 PE 인스턴스가 결정된다. 즉, PE 인스턴스는 특정 타입에 속하는 이벤트들 중 특정 키 값을 갖는 이벤트들만을 처리하는 연산의 기본 단위이다. 예를 들어 단어 발생 수를 세는 WordCountPE의 인스턴스는 단어마다 하나씩 생성되어 해당 단어 이벤트만 받아 집계한다.

S4 응용 개발자는 S4에서 제공하는 자바 API로 PE를 개발하고, Spring 프레임워크를 이용하여 PE를 응용에 연동 활용할 수 있다.

S4는 완전한 symmetric 분산 시스템으로 (그림 2)와 같은 구조를 갖는다. 각 노드에 배치된 프로세싱



<자료>: S4 논문

(그림 2) S4 시스템 구조

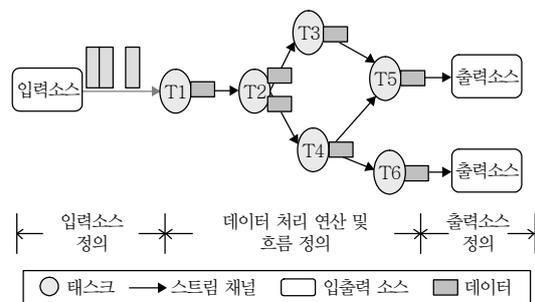
노드(PN)들은 독립적으로 동작하며, PN은 여러 개의 PE 수행을 담당하는 논리적 호스트이다. PN이 담당할 PE의 결정은 키 값에 대한 해시로 결정된다. PN은 이벤트 발생을 감지, 입력 이벤트에 대한 처리, 통신을 통해 이벤트 배치 및 결과 이벤트를 출력하는 역할을 한다.

CL는 PN과 실제 물리적 노드의 매핑, 클러스터 관리, 장애 복구 및 통신 프로토콜 연동 등을 담당한다. 클러스터 노드간의 중재는 zookeeper[12],[13]의 기능을 활용하여 제공한다.

라. iFlow

iFlow는 한국전자통신연구원에서 2010년부터 개발하고 있는 분산 스트림 처리 시스템으로 클러스터 시스템을 기반으로 폭증하는 데이터 스트림을 실시간으로 처리하는 환경을 지원하며, 시스템의 확장성과 스트림 처리 로직의 유연한 통합을 제공한다.

iFlow 시스템은 (그림 3)과 같이 태스크와 스트림 채널을 이용하여, DAG 기반 데이터 스트림 연속 처리 서비스를 구성한다. 태스크는 여러 개의 스트림 입력 소스, 처리 로직, 여러 개의 스트림 출력 싱크로 구성되고, 스트림 채널은 선행 태스크의 스트림 출력 싱크와 후행 태스크의 스트림 입력 소스간의 매핑을 담당한다. 서비스간 태스크 공유와 데이터 스트림 자동 분할을 통한 병렬 처리를 위해 태스크간 데이터 채널



(그림 3) iFlow 서비스 모델

을 동적으로 구성 가능하다. 태스크의 데이터 처리 로직은 사용자가 iFlow API를 이용하여 작성한다.

태스크는 입력 데이터 처리 단위를 윈도우를 이용하여 정의하고, 데이터 윈도우는 시간 혹은 레코드 개수의 두 가지 타입을 가지며, 윈도우 크기와 윈도우 시작 포인트로 윈도우를 지정한다.

- Time-based-window (time interval, sliding time interval)
- Event-based-window (number of event, number of sliding event)

iFlow는 갑작스런 데이터 폭증에 대응하기 위해 데이터 스트림을 분할하여 병렬 처리한다. 데이터 분할은 태스크의 윈도우에 위반되지 않도록 처리하며, 병렬 처리 태스크에 데이터 분배는 round robin 방법을 기본으로 제공하고, 사용자가 정의하여 통합 가능하다.

iFlow 시스템은 (그림 4)와 같이 서비스 관리기, 태스크 실행기, 태스크 프로세스로 구성되며, 메타 데이터 저장 및 서비스 관리기 장애에 대처하기 위해 zookeeper를 이용한다. 서비스 관리기는 iFlow 시

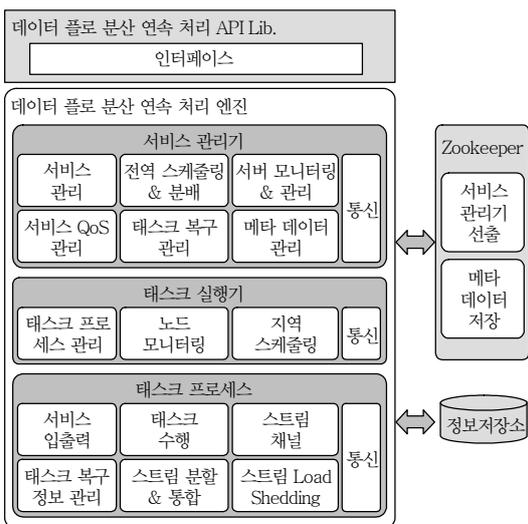
스템 제어 및 사용자 정의 서비스들을 관리하는 시스템으로 태스크 실행기 관리, 서비스에 대한 메타 정보 관리, 서비스를 구성하는 태스크들을 서비스의 QoS를 고려하여 각 노드에 배치하는 업무 등을 담당한다. 태스크 실행기는 노드별로 별도로 구동되며, 서비스를 구성하는 각 태스크들을 실행시키는 시스템으로 태스크 실행 환경 구성 및 태스크 모니터링 등을 수행하며, 태스크는 연속 실행 제어, 태스크들간의 스트림 전달 등 시스템에서 제공하는 기본 기능과 통합되어 별도의 프로세스로 동작한다.

마. 분산 스트림 처리 시스템 비교

Sprout는 DAG 기반으로 서비스 정의가 용이하며, 서비스를 구성하는 모듈의 동작을 외부에서 제어할 수 있게 한다는 장점이 있으나, 서비스별로 별도의 sprout 서버가 구동되므로 서비스간의 효율적인 자원 공유 및 서비스간 처리 결과의 공유 등을 제공하지 못한다. 또한 데이터 폭증에 대한 대처로 선행 stage의 수행을 저지하는 방식은 외부 입력 데이터의 발생이 가변적이고, 갑자기 폭증하는 응용 환경에서는 적합하지 않다.

MapReduce Online은 MapReduce 모델 및 인터페이스를 그대로 사용하면서 연속 처리가 가능하고 서비스 가용성을 제공한다는 장점이 있는 반면, 연속 처리를 위한 윈도우 정의 및 복잡한 스트림 처리 서비스 구성의 유연성(다중 입출력, DAG 모델 미지원 등) 등에 문제가 있다.

S4는 서비스를 담당하는 논리적 호스트 PN과 물리 노드를 구분함으로써 장애 대처가 효율적이라는 장점을 가지나, 키 값 기반으로 PE 인스턴스와 PN이 결정되고, PN 단위로 물리적인 노드와 매핑되므로 서비스 단위 혹은 서비스를 구성하는 태스크 단위의 부하 분산에 어려움이 있다.



(그림 4) iFlow 시스템 구조

〈표 2〉 분산 스트림 처리 시스템 비교

	Sprout	MapReduce Online	S4	iFlow
구조	응용서비스별 Master/Slave	Master/Slave	Peer	Master Cluster/Slave
서비스 모델	DAG	MapReduce	Actor	DAG
데이터 분할	Any, All, Round Robin	Key Range, Key Hash	Key 기반	Round Robin, 사용자 정의
태스크 윈도우 정의	Firing Rule	고정	출력에 대해 시간/이벤트 간격	시간/이벤트 간격
데이터 폭증 대처	일시적 폭증 미고려	실시간 발생 미고려	키 값 조정으로 가능	동적 분할, Load Shedding
부하 분산	단일 서비스내 Stage별	전체 서비스내 태스크별	PN별	전체 서비스내 태스크별
시스템 가용성	미지원	Slave 복구	PN 복구	Master/Slave 복구, 선택적 데이터 복구
API	C++	Java	Java	Java

〈표 2〉의 비교처럼 Sprout, MapReduce Online, S4 3가지 시스템 모두 대량의 동영상 혹은 텍스트를 대상으로 한 특정 응용 지원이 목적이므로 시스템 장애 발생시 데이터 스트림 복구에 대해 고려하지 않고 있다.

iFlow는 정형/비정형 데이터의 통합 분석이 요구될 것으로 예측되는 미래 서비스를 위해 유연한 서비스 정의 및 선택적 데이터 복구 지원, 데이터 분할 병렬 처리 및 load shedding 등 데이터 폭증에 대한 동적 대처, 서비스간 태스크 공유 및 태스크 단위의 분배 기능 등을 제공하여 웹 정보 검색, 지능형 영상 감시, 웹 로그 분석 및 금융 거래 정보 분석 등에 적용을 목적으로 개발되고 있다.

IV. 결론

본 고에서는 데이터 스트림 컴퓨팅 기술 동향에 대해 소개하며, 최근에 연구가 활발히 되고 있는 분산 스트림 처리 기술에 대해, 사례 시스템 및 현재 ETRI에서 개발하고 있는 iFlow 시스템에 대해 소개하였다.

분산 스트림 처리 기술은 모니터링 & 대응 서비스를 위한 기반 기술로, 데이터 처리 지연 최소화, 데이터 폭증에 대한 확장성 및 응용의 유연한 통합 환경을 제공한다. 분산 스트림 처리 기술은 데이터량의 증

가, 이로 인한 처리 시간 지연 등으로 인해 새롭게 각광받고 있는 분산 병렬 처리 기술과 데이터 처리 지연 최소화를 위해 데이터 선처리 후저장 방식의 데이터 처리 모델인 스트림 처리 기술이 통합된 기술이다. 분산 스트림 처리 기술은 컴퓨팅 오버헤드로 인해 실현되기 힘들었던 텍스트, 영상 등 대량의 비정형 데이터 기반의 실시간 서비스를 가능하게 하는 기술이다.

iFlow는 분산 스트림 처리 인프라를 개발한 다음, 텍스트 데이터 스트림 처리에 필요한 데이터 모델 및 처리 연산들을 통합 제공하여 적용 응용 분야를 확대해 나갈 계획이다.

● 용 어 해 설 ●

분산 스트림 컴퓨팅: 지속적으로 발생하는 데이터 흐름에 따라 처리 업무가 실시간으로 수행되고, 처리 업무들을 여러 노드에 분산하여 수행하는 환경

데이터 스트림 연속 처리: 저장된 데이터를 대상으로 즉시 처리하여 한꺼번에 결과를 제공하는 모델과 달리, 계속 입력되는 데이터를 대상으로 연속적으로 처리하여 결과를 지속적으로 제공하는 처리 모델

데이터 윈도우: 연속 처리 업무에서 업무를 한번 처리하는 데 필요한 데이터 영역을 의미하며, 발생한 데이터의 개수 혹은 데이터의 발생 시간을 기준으로 데이터 영역을 표현

약어 정리

BAM Business Activity Monitoring

CCTV	Closed Circuit Television
CL	Communication Layer
DAG	Directed Acyclic Graph
DBMS	DataBase Management System
DSMS	Data Stream Management System
iFlow	insight from exploding data Flow
PB	Peta Byte
PE	Processing Element
PN	Processing Node
QoS	Quality of Service
RFID	Radio Frequency Identification
SaaS	Software as a Service
SNS	Social Network Service
ZB	Zeta Byte

참고 문헌

- [1] 2010 Digital Universe Study iView content-FINAL Version: 4-26-2010, IDC & EMC.
- [2] Hype Cycle for Business Intelligence, Gartner, 2010. 8.
- [3] The Gartner Reference Architecture for Complex-Event Processing, Gartner, 2009. 6.
- [4] Sharma Chakravarthy and Qingchun Jiang, Stream Data Processing: A Quality of Service Perspective Modeling, Scheduling, Load Shedding, and Complex Event Processing, Springer, 2009.
- [5] Roger Rea and Krishna Mamidipaka, "IBM InfoSphere Streams, Enabling Complex Analytics with Ultra-low Latencies on Data in Motion," 2009.
- [6] Ming-yu Chen, Lily Mummert, Padmanabhan Pillai, Alex Hauptmann, and Rahul sukthankar, "Exploiting Multi-level Parallelism for Low-latency Activity Recognition in Streaming Video," ACM Multimedia Systems, 2010.
- [7] Jeffrey Dean and Sanjay Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," OSDI'04, 2004, pp.137-150.
- [8] Hadoop MapReduce, <http://hadoop.apache.org/mapreduce>
- [9] Tyson Condie, Neil Conway, Peter Alvaro, Joseph M. Hellerstein, Khaled Elmeleegy, and Russell Sears, "MapReduce Online," NSDI'10, 2010, pp.21-21.
- [10] Leonard Neumeyer, Bruce Robbins, Anish Nair, and Anans Kesari, "S4: Distributed Stream Computing Platform," KDCLOUD'10, 2010.
- [11] S4, <http://s4.io/>
- [12] P. Hunt, M. Konar, F.P. Junqueira, and B. Reed, "ZooKeeper: Wait-free Coordination for Internet-scale Systems," USENIXATC'10, 2010.
- [13] Hadoop ZooKeeper, <http://hadoop.apache.org/zookeeper>