



특집 02

빅데이터 분석을 위한 빅데이터 처리 기술 동향



이미영·최 완 (한국전자통신연구원)

목 차 »

1. 개요
2. 빅데이터
3. 빅데이터 분석 플랫폼
4. 빅데이터 처리 기술
5. 결론

1. 서론

스마트폰, CCTV, RFID, 센서 등의 출현은 사람, 사물, 환경의 스마트화를 가속시키고 있으며, 데이터의 생산, 유통, 소비 체계에 큰 변화를 주고 있다. 데이터의 폭증으로 데이터가 경제적 자산이 되는 빅데이터 시대를 맞이하고 있으며, 빅데이터의 활용 및 이를 위한 새로운 IT 기술에 대한 요구가 커지고 있다.

시장 조사 분석에 의하면^[1], 2020년 전 세계 디지털 데이터량은 2009년 대비 44배 증가한 35제타바이트에 달할 것으로 예측하고 있으며, 이는 생산되는 스토리지 용량보다 더 많은 양의 데이터일 것으로 전망하고 있다. 또한 생성 데이터의 90% 이상이 텍스트, 영상, 음성, 비디오 등 비정형 데이터이며, 센서 데이터, 트윗 등 실시간으로 발생하는 스트림 데이터가 폭증하고 있다.

데이터의 폭증, 비정형 데이터의 급증 및 빠른 데이터 생성은 데이터 활용 방법에도 영향을 미

치고 있다. 수많은 이중 데이터안에 숨어 있는 가치를 찾아 활용하고자 하는 욕구가 커지고 있고, 최신 정보 기반의 빠른 서비스를 요구하고 있다. 빅데이터는 기존의 전형적인 데이터 처리, 저장 관리 및 분석 기술에 큰 변화를 요구하고 있다.

본고에서는 빅데이터 시대에 대처하기 위한 대규모 데이터 처리 기술 동향에 대해 소개한다. 2장에서 빅데이터의 속성 및 활용 방법의 변화에 대해 알아보고, 3장에서는 빅데이터 활용을 지원하기 위한 빅데이터 분석 플랫폼에 대해 간략히 기술하고, 4장에서는 빅데이터 분석 플랫폼을 구성하는 핵심 기술인 빅데이터 처리 기술 동향에 대해 기술하고, 마지막으로 결론을 맺는다.

2. 빅데이터

2.1 빅데이터 속성

빅데이터는 기존의 IT 방식으로는 저장/관리/

분석이 어려울 정도로 데이터의 생성 속도가 빠르고, 다양한 종류로 구성된 큰 규모의 데이터를 의미한다. 빅데이터의 속성은 크게 데이터 규모, 데이터 다양성, 데이터 생성/처리 속도로 정의하고 있다. 이 외에도 데이터 활용 환경의 복잡성까지 포함하여 빅데이터 속성을 정의하기도 한다^[2]. 빅데이터를 구성하는 속성은 상호 밀접하게 연관되어 있으며, 속성의 정도가 심화될수록 현재 기술로는 지원이 불가능하므로 기술 개선 및 혁신이 필요하다.

- 데이터 규모(Volume): 인터넷, 모바일폰 및 센싱 장치 등의 활용 확산으로 데이터 생성 속도가 가속화되고 있다. 2009년 약 800억사바이트의 데이터가 생성 및 복제되었고, 연평균 40% 성장률로 데이터가 증가되어 2020년에는 2009년 대비 44배 증가한 35제타바이트에 달할 것으로 예측하고 있다^[1].
- 데이터 다양성(Variety): 감시 카메라에서 생성되는 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 정형 데이터뿐만 아니라 다양한 비정형 데이터가 생성되고 있다.
- 데이터 생성/처리 속도(Velocity): 자동으로 데이터를 생성하는 센싱 장치, 스마트폰 등과 같이 데이터 생성 및 유통 채널의 변화로 지속적으로 생성되는 스트림 데이터가 증가하고 있으며, 데이터 생성 속도가 빨라지고 있다. 데이터 생성 속도의 가속화는 이를 활용하기 위한 처리 속도의 가속화도 요하고 있다.
- 데이터 활용 환경의 복잡성(Complexity): 데이터 종류에 따라 적용되는 표준, 도메인별

규칙, 저장 형식 등이 서로 다르다. 데이터안에 숨어 있는 가치있는 정보를 얻기 위해서는 이와 같은 이질의 데이터를 통합하여 활용할 필요성이 커지고 있다.

2.2 빅데이터 활용 방식

빅데이터 시대가 되면서 정보의 홍수 속에서 가치있는 지식을 찾아내기가 점점 더 어려워지고 있다. 또한 빅데이터 속에 어떤 유용한 정보가 숨어 있는지, 무엇을 찾아야 할지를 미리 알지 못하므로 정보를 활용하는 방법도 변하고 있다. 즉, 찾고자 하는 것이 무엇인지를 미리 알고 이를 찾는 정보 검색에서, 무슨 정보가 숨어 있는지 알 수 없는 상황에서 정보를 분석하여 유용한 지식을 찾아 활용하는 방향으로 변하고 있다.

또한 정보를 분석하여 활용하는 방법에도 있어서도 예측 및 대응하는 방향으로 발전하고 있다. 기존에는 이미 발생한 현상에 대한 정보를 저장 관리하고, 이로부터 왜 발생했는지 이유를 확인하는 방식이었지만, 지금은 현재 어떤 일이 발생하고 있으며, 왜 발생하는지를 실시간으로 파악, 활용하고자 하는 요구가 커지고 있다. 앞으로는 미래에 발생할 일에 대해 미리 예측하고 대응 방안을 마련하기 위한 분석 방법으로 발전할 것이다^[3].



출처: Gartner 2011

(그림 1) 빅데이터 활용 방식의 변화

3. 빅데이터 분석 플랫폼

데이터로부터 지능을 발굴하여 활용하는 과정은 기본적으로 데이터 수집, 데이터 전처리, 정보 저장관리, 정보 처리/분석, 지능 발굴 및 가시화를 통해 활용하는 6단계로 볼 수 있다.

다양한 데이터 소스로부터 데이터를 수집하고, 필요한 경우 수집된 데이터로부터 필요 없는 데이터를 필터링 혹은 적절한 형태로 가공하는 전처리 단계를 거친 다음, 정보를 저장 관리하면서 이로부터 유용한 정보를 추출 혹은 분석 과정을 통해 지능을 발굴하고 이를 가시화하여 활용하는 단계로 정의할 수 있다. 경우에 따라서는 정보 저장 관리 단계 전에 유용한 정보를 추출 혹은 분석 작업이 선행된 후 필요시 저장 관리한다.

데이터를 수집, 지능을 발굴하여 활용하기까지의 각 과정은 빅데이터 환경이 되면서 기술적인

개선/혁신이 필요하다. 데이터 규모, 데이터 생성 및 처리 속도, 데이터 다양성, 데이터 활용 환경의 복잡성 등 빅데이터 속성을 고려한 기술 지원이 되어야 한다.

빅데이터 활용을 지원하기 위한 빅데이터 분석 플랫폼의 주요 기술은 크게 이기종 데이터 융합 기술, 빅데이터 저장 관리 기술, 빅데이터 처리 기술, 빅데이터 분석 기술 및 지능 가시화 기술로 나눌 수 있다.

- 이기종 데이터 융합 기술: 빅데이터 분석을 위해 다양한 데이터 소스에 접근, 데이터 수집 및 데이터 융합 처리를 지원하는 이기종 멀티데이터 융합 기술
- 빅데이터 저장 관리 기술: 폭증하는 다양한 형식의 데이터를 저장 관리할 수 있는 고확장성의 분산 저장 관리 기술⁵⁾ 및 실시간 분



(그림 2) 데이터에서 지능 발굴 활용 단계



(그림 3) 빅데이터 분석 플랫폼 구성

- 빅데이터 처리 기술 : 지속적으로 발생하는 스트림 데이터 및 대규모 저장 데이터의 적시 분석 처리를 지원하기 위한 분산 병렬 데이터 처리 기술
- 빅데이터 분석 기술 : 빅데이터에 내재된 가치를 추출하기 위해 필요한 대규모 데이터 및 스트림 데이터기반 통계 처리, 데이터 마이닝 등 공통 분석 기술 및 기계 학습, 인공지능 기법을 적용한 고급 분석 기술
- 지능 가시화 기술 : 분석된 결과를 쉽게 직관적으로 이해할 수 있도록 도표, 이미지, 애니메이션 등을 활용하여 가시화하여 제공하는 기술

4. 빅데이터 처리 기술

4.1 빅데이터 처리 기술의 발전

빅데이터에서 유용한 정보 및 숨어 있는 지식을 찾아내기 위한 데이터 가공 및 분석 과정을 지원하는 빅데이터 처리 기술은 빅데이터 속성인 데이터 규모, 데이터 생성/처리 속도, 데이터 다양성에 대한 고려가 필요한 기술이다. 대규모 데이터 처리를 위한 확장성, 데이터 생성 및 처리 속도를 해결하기 위한 처리 시간 단축 및 실시간 처리 지원, 비정형 데이터 처리 지원 등이 필요하다.

대표적인 빅데이터 처리 기술인 MapReduce 기술은^[5] 배치성의 데이터 처리 업무에 적합한 기술로 기술의 장단점에 대한 많은 논쟁이 있어 왔으나^[6], 현재는 오픈 소스 Hadoop의 성공으로 분산 병렬 데이터 처리 기술의 사실 표준이 되었다. MapReduce 기술은 일반 범용 서버로 구성된 클러스터 시스템을 기반으로 key-value 기반의

입력 데이터 분할 처리 및 처리 결과 통합 기술, Job 스케줄링 기술, 태스크 분배 기술, 장애 대처를 위한 태스크 재수행 기술 등이 통합된 분산 컴퓨팅 기술이다.

MapReduce 기술이 확산되면서 새로운 하드웨어 시스템에 최적화된 데이터 처리 기술 연구, 반복/연속 처리 지원 및 유연한 데이터 흐름을 표현하기 위한 프로그래밍 모델 개선 등에 대한 연구가 수행되고 있다. 또한 데이터 활용 방식의 변화로 현재 발생하는 상황 파악 및 발생 원인의 실시간 분석에 대한 중요성이 커짐에 따라 대규모 스트림 데이터 처리 기술에 대한 연구가 수행되고 있다.

4.2 매니코어기반 MapReduce 기술

전력 소비 대비 성능 측면을 고려하여 소수의 코어로 구성된 노드들을 다수 연결하여 구성하는 것보다, 단일 칩에 다수의 코어를 집적시킨 매니코어가 탑재된 노드를 소수 연결한 클러스터 시스템에 대한 관심도가 높아지고 있다. 현재 NVIDIA의 GPGPU(General Purpose computing on Graphic Processing Units)를 가속 장치로 널리 활용하고 있으며, 인텔에서는 MIC(Many Integrated Core)라는 매니코어기반 가속 장치를 개발하고 있다. 또한 Tiler는 매니코어 CPU로 구성된 서버를 출시하고 있다.

대규모 데이터 처리 시스템도 이와 같은 매니코어 시스템에서 효율적으로 운영될 수 있도록 확장 개선이 필요하다. 매니코어 환경에 적합한 태스크 운영, 코어들간에 데이터 전달 방법, 그리고 가속 장치로 활용되는 GPGPU처럼 장치내 메모리 용량을 벗어나는 데이터 처리에 대한 고려가 필요하다. 또한 GPGPU 환경에서는 메모리 계층이 모든 코어에서 접근 가능한 글로벌 메모리

와 프로세서내 코어들만이 접근 가능한 공유 메모리가 존재하므로 이에 대한 최적 활용이 필요하다^[7].

이와 같은 매니코어 환경을 고려한 MapReduce 기술 개발 주요 사례는 다음과 같다.

- Mars^[8]: GPGPU를 활용하여 병렬 처리하기 위해, 데이터를 GPGPU 메모리 공간으로 옮기고, GPGPU의 SIMD 기술을 활용하여 처리하는 MapReduce 시스템이다. GPGPU에서 동적 메모리 할당을 지원하지 않는 문제를 해결하기 위해 1단계에서 소요 메모리 공간 계산을 위해 처리할 데이터 량에 대한 계산 후 2단계에서 실제 태스크를 수행하는 방법으로 확장하였다.
- MapCG^[9]: 데이터 처리 응용을 CPU, GPU를 혼용하여 처리하고 있으며, 메모리 할당 모듈의 자체 개발을 통해 GPU의 동적 메모리 관리 제약을 극복하고 있다. 또한 GPU의 메모리 크기를 초과하는 대량 데이터 처리를 위해 데이터를 분할하여 각 슬라이스 별로 처리 후, 결과를 저장 관리하고 마지막에 Merge 과정을 거쳐 최종 결과를 만들어내는 구조로 설계되었다. 또한 Map에서 각 Reduce 태스크 별로 나뉘질 데이터를 분류하는데 해

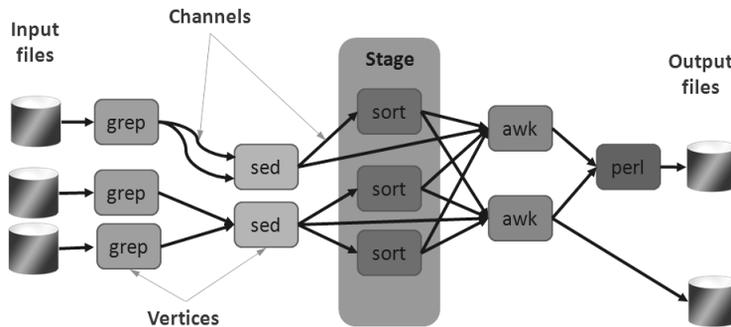
쉬 테이블 구조를 도입하여 정렬 과정을 생략하였다.

- GPMR^[10]: 단일 GPU에서 제공하는 메모리 공간을 초과하는 데이터 처리를 위해서 GPU 클러스터 환경을 기반으로 데이터를 chunk로 나누어 처리할 수 있도록 제공하고 있으며, GPU 간 네트워크 통신을 최소화하기 위해 Map 처리 후 Combiner와 결과 누적 과정을 추가하여 태스크간 데이터 전송량을 줄이는 방법을 제안하고 있다.

이밖에도 Map 과정에서 데이터 입출력시 GPU 공유메모리를 효과적으로 활용하기 위한 연구도 수행되었다^[11].

4.3 DAG 기반 데이터 처리 기술

MapReduce 모델외에도 DAG(Directed Acyclic Graph) 기반의 프로그래밍 모델을 제공하는 배치형 데이터 처리 시스템으로 Microsoft에서 개발한 Dryad^[12]가 있다. Dryad는 데이터 처리 로직을 표현하는 Vertex와 입출력 데이터나 Vertex 간의 데이터 전송 채널을 표현하는 Edge로 구성된 그래프 기반의 Job 정의를 제공한다. Vertex는 여러 개의 입력 채널과 출력 채널을 가질 수 있으며,



(그림 4) DAG 모델 기반 데이터 처리

파일, TCP 파이프, 공유 메모리 FIFO 중에서 Vertex 간의 데이터 전송 방법을 설정할 수 있다.

Dryad 시스템은 범용 서버들을 이용한 대규모 클러스터 환경뿐만 아니라, 향후 로컬 컴퓨팅 파워의 발전 방향이 단일 코어(single core)에서의 instruction-level의 병렬 처리화나 단일 칩(chip) 상에서 코어 수의 증가가 예측됨을 고려하여 설계되었기 때문에 다중 코어를 소유한 단일 컴퓨터에서부터 수천대의 컴퓨터로 구성된 클러스터에 모두 운영이 가능하다.

4.4 데이터 반복/점진 처리 기술

데이터를 가공, 분석하는 업무는 많은 경우에 서로 다른 데이터 집합에 대해 동일한 로직을 적용하여 처리한다. 이와 같이 데이터만 달리하여 반복적으로 처리하는 업무를 위해 실행 환경을 설정하고 나면 반복적으로 사용할 수 있도록 확장함으로써, 프로세스 재구동 등 수행 환경 설정에 소요되는 시간 및 자원 낭비를 막을 수 있다. 또한 데이터가 변경되어도 변화하지 않는 데이터를 재활용할 수 있도록 지원함으로써 데이터 처리 성능 개선이 가능하다. MapReduce 기반의 데이터 반복 처리 기술에 대한 연구는 다음과 같다.

- Twister^[13]: 반복적인 MapReduce 처리를 효과적으로 처리하기 위하여, 한번 설정후 반복적으로 사용할 수 있도록 지원하고 있으며, 태스크간 데이터 전달을 분산 메모리 기반으로 publish/subscribe 메시징 기술을 이용하도록 확장되었다.
- HaLoop^[14]: Twister와 마찬가지로 반복 작업의 효율을 높이기 위하여, 동일 데이터를 사용하여 처리하는 태스크를 같은 노드에 배치하고, 변경이 없는 데이터를 캐쉬에 관리하

여 공유함으로써 IO 비용을 줄일 수 있도록 확장하였다.

버클리대학에서 개발한 MapReduce Online^[15]은 Map 단계의 처리 결과 데이터를 Reduce 단계로 전달하는 과정에 일정량의 데이터 처리가 완료되면 Reduce에게 전달하는 파이프라인 기능을 적용하여 점진적인 처리가 가능하도록 확장하였다. MapReduce Online은 최종 처리 결과에 대한 응답 시간 단축 및 주기적으로 중간 처리 결과 제공이 가능하며, 연속 처리 환경 구축도 가능하다.

4.5 실시간 데이터 처리 기술

지속적으로 발생하는 데이터 스트림에 대한 실시간 처리를 위해, 데이터 모델에 따라 정형 이벤트 스트림 처리 연산을 제공하는 CEP(Complex Event Processing) 기술^[16]이 주로 활용되고 있으나, 비정형 스트림 및 대규모 스트림 처리를 위해 데이터 처리 로직을 자유롭게 정의하여 사용할 수 있는 분산 스트림 병렬 처리 기술이 나오기 시작했다. 대표적인 오픈 소스 시스템으로 SNS의 트윗 실시간 처리를 위해 만들어진 트위터의 STORM, 초당 수 천건의 질의 요청을 받는 웹 검색 서비스에서 사용자의 검색 요구를 실시간으로 분석하는 환경 구축을 위해 만들어진 Yahoo!의 S4(Simple Scalable Streaming System) 등이 있다. 또한 IBM에서는 InfoSphere Streams^[17] 제품을 기반으로 실시간 스트림 데이터 분석을 지원하고 있고, HStreaming^[18]에서는 Hadoop 기반으로 배치 처리 및 실시간 처리를 모두 지원하는 시스템을 내고 있다.

- S4^[19]: 저지연 속도로 스트림 처리 제공을 위

해 클러스터 시스템에서 동작하도록 개발된 시스템으로, 연산을 수행하는 PE(Processing Element)와 PE 사이에 전달되는 메시징인 이벤트 데이터로 구성된다. PE는 병렬 처리가 가능하도록 여러 개의 인스턴스를 구동할 수 있으며, PE 인스턴스가 처리할 이벤트 데이터 분배 방식은 데이터 타입 및 키 값에 의해 결정된다. 즉, PE는 특정 타입에 속하는 이벤트들 중 특정 키 값을 갖는 이벤트들만을 처리하는 연산의 기본 단위이다. 시스템 구조는 모든 노드에 동일하게 PN(Processing Node)이 수행되는 형태이며, PN이 실제 여러 개의 PE 수행을 담당한다.

- STORM^[20]: 필드들로 구성된 튜플 스트림 입력에 대해 그래프로 데이터 처리 흐름을 표현하고, 그래프의 각 노드에 해당하는 처리 로직이 클러스터 시스템에 배치되고 나면 연속적으로 실행되는 구조이다. 그래프 노드들의 처리 로직은 병렬로 수행되며 하나의 처리 로직에 대해서도 다양한 입력 데이터 분배 방식을 통해 병렬로 처리 가능하다. 그래프를 구성하는 노드는 데이터 소스로부터 데이터를 읽어 스트림을 출력하는 역할을 담당하는 Spout와 입력 스트림을 읽어, 처리후 결과 스트림을 출력하는 기능을 제공하는 Bolt로 구성된다. 장애에 대처하기 위해 입력 스트림에 대한 처리가 정해진 시간내에 완료되지 않으면 재수행하는 방법을 사용한다. 시스템 구조는 MapReduce와 마찬가지로 Master/Slave 구조로 전체 시스템의 배치 및 동작 상황을 관리하는 Nimbus와 처리 로직 수행을 담당하는 Supervisor들로 구성된다.

실시간 데이터 처리 기술은 빠른 처리를 위해

메모리 기반 처리를 하고 있으며, 대규모 스트림 처리를 위한 확장성을 위해 입력 데이터의 분할 병렬 처리, 태스크의 병렬 처리 등 분산 병렬 처리 기술이 활용되고 있다. 앞으로 실시간 데이터 처리 기술의 활용이 확대되면 더 큰 메모리에 대한 요구 및 메모리 소비 전력 절감에 대한 필요성이 커질 것으로 예상되므로, 요구되는 처리 속도 및 처리 로직 등을 고려하여 DRAM뿐만 아니라 차세대 비휘발성 메모리 활용에 대한 고려가 필요하다.

5. 결론

빅데이터 활용은 개인, 기업, 정부 등 스마트한 사회를 이루기 위한 필수적인 요건이 되고 있다. 또한 과학 연구를 위한 수단으로도 빅데이터를 분석하여 원리 발굴 및 문제 해결을 시도하는 데이터 탐구 과학 시대가 되었다. 빅데이터를 얼마나 잘 다룰 능력이 있느냐에 국가의 경쟁력이 좌우되는 시대를 맞이하여, 이를 지원하는 빅데이터 분석 플랫폼 기술 확보가 필수적이다.

이에 따라 빅데이터의 속성 및 활용 방식의 변화 방향에 대해 알아보고, 빅데이터 활용을 지원하는 빅데이터 처리 기술의 연구 동향을 조사 분석하며, 연구 방향에 대해 알아보았다. 빅데이터의 속성 중 데이터 규모, 데이터 생성 및 처리 속도, 데이터 다양성의 정도가 가속화될수록 현재의 IT 기술이 비용대비 효과나 성능 제약 문제가 대두될 것으로 보인다.

차세대 메모리, 매니코어 등 하드웨어 발전 방향을 고려한 빅데이터 처리 기술 개발을 통해 앞으로 더욱 경쟁이 치열할 빅데이터 플랫폼 시장에서 국내 기술이 경쟁력을 가질 수 있기를 기대한다.

참 고 문 헌

- [1] McKinsey Global Institute, Big data: The next Frontier for Innovation, competition, and productivity, McKinsey & Company, 2011.
- [2] Mark A. Beyer, Anne Lapkin, Nicholas Gall, Donald Feinberg, Valentin T. Stribar, Big Data Is Only the Beginning of Extreme Information Management, Gartner, 2011.
- [3] Yvonne Genevese, Ian Bertram, "Information 2020: Senario for Business Intelligence & Information Management," Gartner, 2011.
- [4] 이미영, "클라우드 기반 대규모 데이터 처리 및 관리 기술," 전자통신동향분석, 2009.
- [5] Jeffrey Dean, Sanjay Ghemawat, "MapReduce: Simplified Data processing on large clusters," OSDI, 2004.
- [6] Kyong-Ha Lee, Yoon-Joon Lee, Hyunsik Choi, Yon Dohn Chung, Boogki Moon, "Parallel Data Processing with MapReudce: A Survey," SIGMOD Record, 2011.
- [7] NVIDIA's Next Generation CUDA Compute Architecture: Fermi, NVIDIA, 2009.
- [8] Bingsheng He, Wenbin Fang, Naga K. Govindaraju, Qiong Luo, Tuyong Wang, "Mars: a MapReduce framework on graphics processors," PACT, 2008.
- [9] Chuntao-Hong, Dehao Chen, Wenguang Chen, Weimin Zheng, Haibo Lin, "MapCG: Writing Parallel Program Portable between CPU and GPU", PACT, 2010.
- [10] Jeff A. Stuart, John D. Owens, "Multi-GPU MapReduce on GPU Clusters," IPDPS, 2011.
- [11] Feng Ji, Xiaosong Ma, "Using Shared Memory to Accelerate MapReduce on Graphics Processing Units," IPDPS, 2011.
- [12] M. Isard, Mihai Budiu, Yuan Yu, Andrew Birrell, Dennis Fetterly, "Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks," EuroSys, 2007.
- [13] Jaliya Ekanayake, Hui Li, BingJing Zhang, Thilina Gunarathne, Seung-Hee Bae, "Twister: A Runtime for Iterative MapReduce," HPDC, 2010.
- [14] Yingyi Bu, Bill Howe, Magdalena Balazinska, Michael D. Ernst, "HaLoop: Efficient Iterative Data Processing on Large Clusters," VLDB, 2010.
- [15] Tyson Condie, Neil Conway, Peter Alvaro, Joseph M. Hellerstein, Khaled Elmeleegy, Russell Sears, "MapReduce Online," NSDI, 2010.
- [16] Gianpaolo Cugola, Alessandro Margara, "Processing Flows of Information: From Data Stream to Complex Event Processing," DEBS, 2011.
- [17] Roger Rea, Krishna Mamidipaka, "IBM InfoSphere Streams, Enabling complex analytics with ultra-low latencies on data in motion," IBM, 2009.
- [18] <http://www.hstreaming.com/>
- [19] Leonard Neumeyer, Bruce Robbins, Anish Nair, Anans Kesari, "S4: Distributed Stream Computing Platform," KDCloud, 2010.
- [20] <https://github.com/nathanmarz/storm/wiki/>

저 자 약 력



이 미 영

이메일 : mylee@etri.re.kr

- 1981년 서울대학교 식품영양학과(학사)
- 1983년 서울대학교 계산통계학과(석사)
- 2005년 충남대학교 컴퓨터공학과(박사)
- 1983년~1985년 한국전자통신연구소/연구원
- 1988년~현재 한국전자통신연구원/책임연구원
- 관심분야: 데이터베이스 시스템, 데이터 스트림 처리 기술, 분산 병렬 처리 기술



최 완

이메일 : wchoi@etri.re.kr

- 1981년 경북대학교 전자공학과(학사)
- 1983년 KAIST 전산학과(석사)
- 1988년 정보처리 기술사
- 1985년~현재 한국전자통신연구원/부장
- 관심분야: 클라우드 컴퓨팅, 슈퍼 컴퓨팅 기술