클라우드 기반 대규모 데이터 처리 및 관리 기술

Big Data Processing and Management Service on Cloud

클라우드 컴퓨팅 특집

이미영 (M.Y. Lee) 데이터베이스연구팀 팀장

목 차

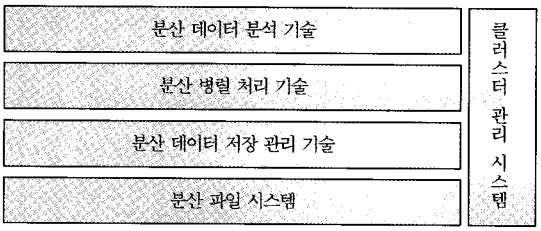
인터넷 서비스 데이터량의 지속적인 증가로 대량의 원시 데이터로부터 정보를 가공 처리하는 과정, 체계화된 정보의 저장 관리 및 유용한 정보를 추출하기 위한 분석 등에 분산 컴퓨팅 기술을 적용하는 움직임이 활발히 진행되고 있다. 기존의 RDBMS 기술, MPI 분산 처리 기술 등은 대규모 데이터 처리 환경에 적용하기에는 운영 환경, 기능/성능면에서 확장성 혹은 고비용 문제가 따른다. 그러므로 저가의 서버들로 구성된 대규모 클러스터 환경을 기반으로 분산 컴퓨팅 기술을 적용한 새로운 시스템들이 대규모 데이터 처리를 요하는 인터넷 서비스 응용에 이용되고 있다. 이를 기반으로 바이오인포매틱스, 과학 시뮬레이션, 비즈니스 인텔리전스 등 다른 응용 영역으로 확대하여 클라우드 서비스로 제공하려는 비즈니스 모델이 제시되고 있다. 본 논문에서는 이와 같은 분산 컴퓨팅 기술을 적용한 대규모 데이터 저장 관리 및 처리 기술 동향을 조사하고 클라우드 기반 서비스로의 발전 방향을 서술한다.

Ⅰ. 서론

1. 분산 컴퓨팅 플랫폼의 등장

인터넷 서비스 데이터는 지속적으로 증가하므로, 이를 서비스하는 시스템의 비용과 확장성이 인터넷 서비스 업체의 경쟁력 확보에 중요한 요소가 되고 있다. 구글, 야후 등 글로벌 인터넷 서비스 업체들은 인터넷 서비스 플랫폼의 중요성을 인식하고 자체 연구 개발을 수행, 저가 상용 노드를 기반으로 한 대규모 클러스터 기반의 분산 컴퓨팅 플랫폼 기술을 개발 활용하고 있다[1]-[3]. 특히 대규모 인터넷 서비스 데이터의 가공, 저장, 검색, 분석 등에 분산 처리 기술을 적용하고 있다.

대규모 데이터 서비스를 위한 분산 컴퓨팅 플랫폼은 (그림 1)처럼 분산 환경에 대한 모니터링 및 스케줄링을 제공하는 클러스터 관리 시스템의 지원 하에 대규모 데이터 저장을 위한 분산 파일 시스템 기술, 대규모 데이터 검색 및 변경을 지원하는 분산 데이터 저장 관리 기술, 대규모 데이터 분석 처리를 지원하기 위한 분산 병렬 처리 및 분석 기술 등이 통합되어 활용되고 있다.



(그림 1) 인터넷 서비스용 분산 컴퓨팅 플랫폼

2. 클라우드 컴퓨팅 플랫폼으로 발전

분산 컴퓨팅 플랫폼 기술은 가상화 기술과 함께 클라우드 컴퓨팅 기술의 핵심이 되고 있다. 클라우드 컴퓨팅 기술은 사용자의 IT 자원 서비스 요구에 대해 유연하게 동적으로 대처함으로써 전체 시스템의 활용성을 증진시켜 IT 비용 절감을 가능하게 한다[4]. 클라우드 컴퓨팅 기술은 최근에 각광을 받고 있는데, 이는 데이터 센터 규모의 빠른 확장, 브로드밴드 인터넷의 확산, x86 하드웨어 및 LAMP 소프트웨어(Linux, Apache, MySQL과 Perl, PHP, Python 등 스크립트 언어로 구성된 솔루션)와 같이 공통 플랫폼의 등장 등으로 인터넷을 통해 IT 자원을 유틸리티 서비스로 제공하는 것이 현실적으로 가능하게 되었기 때문이라고 볼 수 있다[5].

클라우드 서비스는 크게 컴퓨팅 인프라를 서비스로 제공하는 IaaS, 응용을 개발 및 운영할 수 있는 소프트웨어 플랫폼을 서비스로 제공하는 PaaS, 개인이나 기업에서 필요로 하는 소프트웨어를 서비스로 제공, 이용할 수 있게 하는 SaaS 등으로 구분된다[4].

PaaS 서비스는 응용 분야에 따라 소프트웨어 플랫폼의 구성 요소가 달라진다. 현재 PaaS 서비스로 논의되고 있는 서비스 중 하나가 대규모 데이터 관리 및 처리 서비스이다. 이는 데이터량이 지속적으로 증가하고 있어 많은 응용에서 대규모 데이터 처리 및 관리가 필요해지고 있는 상황이기 때문이다.

대규모 데이터 처리 및 저장 관리가 필요한 대표적인 응용으로는 인터넷 서비스 분야뿐만 아니라 바이오 정보 처리, 과학 시뮬레이션, 비즈니스 인텔리전스 등 대량의 데이터를 취급하는 분야이다. 이들 분야에서는 대량의 원시 데이터를 가공하여 정제된 데이터를 만들어 저장 관리하고, 대량의 데이터로부터 새로운 정보를 찾아내기 위한 분석 작업 등을 수행하게 된다. 이와 같은 작업을 수행하기 위해 자체 IT 환경을 구축, 운영하는 것 보다는 클라우드 기반 서비스를 이용하는 것이 더 쉽고 빠르게 처리가 가능하며, 초기 구축 및 운영 비용 절감이 가능하다.

본 논문에서는 이와 같은 클라우드 기반의 대규모 데이터 처리 및 관리 서비스를 가능하게 하는 기술에 대해 조사한다. Ⅱ장에서는 대규모 데이터 저장 관리 기술에 대하여, Ⅲ장에서는 대규모 데이터 분석을 위한 분산 병렬 처리 기술에 대하여 서술하고, Ⅳ장에서는 이들 기술을 활용한 클라우드 기반의 서비스 동향을 조사하고, 마지막으로 결론을 맺기로 한다.

Ⅱ. 대규모 데이터 저장 관리 기술

1. 대규모 데이터 저장 관리 개요

인터넷 서비스 응용은 검색, 업로드와 같이 간단한 트랜잭션 처리와 소량의 데이터 접근이 필요한 온라인 서비스 분야와 키워드 추출, 로그 분석 등과 같이 대량의 데이터 접근이 필요한 배치 서비스 분야가 있다. 기존 데이터베이스 기술은 인터넷 서비스 데이터 규모 및 동시 사용자 수에 대한 확장성 문제로 인해 비용 문제나 성능 문제가 야기된다. 또한 인터넷 서비스 응용에서 필요로 하는 기능보다 너무 많은 기능을 제공하거나, 응용에 맞게 최적화 할 수 있는 유연성이 부족하다[3].

예를 들어 대표적인 인터넷 서비스인 flickr 서비스에서 발생하는 데이터베이스 연산을 보면, 사진을 업로드 및 태깅 서비스가 주요 변경 연산이고, 질의 연산은 내 사진 검색 혹은 태그 기반의 키워드 검색 등 간단한 연산으로 구성된다. 그러나 기존 RDBMS에서는 트랜잭션 처리, 조인 기능 등 너무 많은 기능이 탑재되어 있어 도리어 성능을 저하하는 면이 있다.

또한 분석 연산에 활용하기에는 저장된 데이터를 유연하게 활용할 수 있는 방법을 제공하지 않고 시스템이 제공하는 인터페이스를 통해서만 데이터에 접근하게 한다. 또한 분석 업무에 적합한 데이터 저장 모델 선택 등 최적화를 위한 튜닝 여지를 별로 제공하지 않는다.

그러므로 인터넷 서비스용의 특화된 데이터 저장 관리 시스템이 필요하며, 주요 고려사항은 다음과 같다.

- 데이터 및 사용자 수 증가에 따른 확장성 제공
- 서비스 중단을 최소화하기 위한 고가용성 지원
- 인터넷 서비스 응용에서 요구하는 기능 및 성능에 적합한 데이터 관리 기능 제공

구글, 야후, 아마존 등 글로벌 인터넷 서비스 업체에서는 이와 같은 요구사항을 고려하여 자체적으로 데이터 저장 관리 시스템을 개발 활용하고 있다.

구글은 인터넷 온라인 서비스 및 배치 서비스의 인프라로 Bigtable이란 분산 저장 관리 시스템을 개발하였고[6], 아마존, 야후에서도 Dynamo[7], PNUTS[8] 등을 개발하였으며, 각각의 시스템은 지원하는 수준이 다음과 같이 약간씩 다르다.

- Bigtable: 테이블, 행, 컬럼 등 데이터 구조 모델링이 가능한 분산 구조 데이터 저장 시스템
- Dynamo: key-value 레코드 구조의 분산 데이터 저장 시스템
- PNUTS: 데이터 구조 모델링뿐만 아니라 고급 수준의 처리 언어를 제공하는 분산 데이터베이스 관리 시스템

2. 분산 구조 데이터 저장 시스템

구글에서는 2006년 구조 데이터를 분산 저장 관리하는 Bigtable이란 시스템을 발표하였고, 이후 오픈 소스 프로젝트에서 이와 유사한 기술을 개발하려는 활동이 진행중이다. ASF의 Hadoop 오픈 소스 프로젝트[9]에서 수행되는 Hbase[10]가 있고, 또한 Hypertable[11]이란 오픈 소스도 개발되고 있다.

가. 구글의 Bigtable

Bigtable은 구글에서 웹 페이지 관리를 위해 개발되어 구글의 다양한 서비스(Personalized Search, Google Earth, YouTube 등)에 활용이 되고 있다. 2008년 3월 200개 이상의 Bigtable 클러스터가 운영되고 있으며, 가장 큰 클러스터는 3,000개 이상의 노드로 구성되어 있고, 6 페타바이트 이상의 데이터를 관리한다고 한다[2].

Bigtable은 인터넷 온라인 및 배치 서비스 지원을 위해 필수적으로 요구되는 확장성을 제공하기 위

● 용 어 해 설 ●

Hadoop: 아파치 소프트웨어 파운데이션(Apache Software Foundation)에 구성되어 있는 오픈 소스 프로젝트 명으로, 구글의 인터넷 서비스 플랫폼 기술을 공개 소프트웨어로 개발하고 있는 커뮤니티이다.

해 인터넷 서비스 환경에 특화된 시스템이다. Bigtable의 주요 특징은 다음과 같다[6].

• 다차원 정렬 맵 모델 제공

행, 컬럼, 타임스탬프로 데이터를 구조화하고, 이를 정렬 관리하여 키(행 키: 컬럼 명: 타임스탬프) 기반의 접근을 제공하는 다차원 정렬 맵 모델을 지원한다. 데이터를 정렬 관리하고, 키 기반의 접근으로 제약함으로써 데이터의 빠른 접근이 가능해진다.

• 컬럼 기반의 저장 모델 제공

기존 RDBMS에서 일반적으로 사용하는 행 기반의 저장 모델 대신 컬럼 기반의 저장 모델을 제공한다. 즉 관련된 컬럼들을 그룹핑한 컬럼 그룹별로 별도의 파일에 저장하는 방식을 사용한다. 컬럼 기반의 저장 모델 제공으로 특정 컬럼 기반의 전체 데이터 분석 작업시 빠른 접근이 가능하다.

• 메모리내 정보 관리

다수 사용자의 빠른 변경 및 검색을 지원하기 위해 메모리에서만 데이터를 변경하고, 일괄 디스크 반영을 택함으로써 디스크 I/O 수를 줄이고 데이터 클러스터링을 유지한다. 또한 읽기 전용 버퍼를 관리하여 빠른 검색을 제공한다.

• 데이터 분산 관리

행 키 기반으로 정렬된 테이블을 구간별로 파티션하여 별도로 저장하고, 각 파티션을 노드에 분산 배치하여 서비스함으로써 데이터 증가에 대한 확장성을 제공한다.

• 단일 행 기반의 트랜잭션 모델

인터넷 서비스 응용의 사용 패턴에 맞는 단일 행 기반의 트랜잭션을 지원한다. 행 잠금으로 데이터의 일관성을 유지하고, 메모리내 변경 및 일괄 반영이므로 로그에 수행 내역을 기록하고, 고장이 발생시에는 로그를 이용한 redo를 수행하여 복구한다.

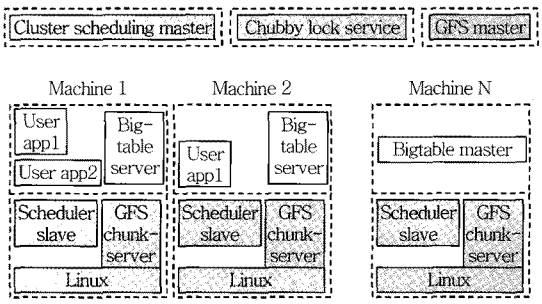
• 데이터 가용성을 위한 복제

노드나 네트워크 고장으로부터 복구를 위한 데이터 복제는 분산 파일 시스템인 GFS[12]에 의존한다. GFS는 크기가 큰 파일 저장용으로 개발된 분산 파일 시스템으로 3개의 복제본을 유지하여 데이터의 고가용성을 제공한다.

• Master/slave 구조

Bigtable은 master, tablet server와 클라이언트 라이브러리로 구성되며, master는 메타데이터 관리 및 tablet server 간의 부하 분산을 담당하고, tablet server는 할당된 테이블 파티션의 데이터 서비스를 담당한다.

Bigtable은 (그림 2)에서 보듯이 분산 파일 시스템인 GFS, 잠금 서비스 시스템인 Chubby[13] 및 클러스터 스케줄링 시스템을 기반으로 동작한다[2].



<자료>: Google, 2008.

(그림 2) Bigtable 운영 환경

나. Hadoop의 Hbase

Hbase는 구글의 Bigtable과 유사한 시스템으로 Java로 개발되고 있는 오픈 소스이다. Hadoop의 서브 프로젝트로 Hadoop의 분산 파일 시스템을 이용한다. 2007년 5월 Powerset에서 최초 버전을 제공한 이래, 2009년 6월 기준 0.19.3 버전이 나온 상태로 아직 Bigtable에서 제공하는 기능 및 성능에 미치지 못하고 있다[10].

Hbase는 Adobe, Powerset, 야후 등 여러 기업에서 실제 활용되고 있다. Powerset에서는 자사의 위키피디아를 구축, 위키피디아 각 페이지를 구성하는 문서를 저장 관리하는 데 Hbase를 이용하고 있다. 또한 야후에서는 유사한 문서를 검출하기 위해

문서의 핑거프린트를 저장하는 데 Hbase를 이용하고 있다[10].

상기 Hbase 활용 사례에서 보듯이 메타 데이터를 RDBMS에 관리하고 원시 데이터는 별도의 파일로 관리하는 기존 방식과 달리, Hbase를 이용하는 응용에서는 문서와 문서에 대한 메타 데이터를 같이 관리하고 있다. 이는 RDBMS와 파일 시스템을 분리하여 사용시 응용에서 두 시스템을 연동하여 사용하는 오버헤드 및 대량의 파일 생성으로 야기되는 파일 시스템의 문제(파일 개수의 제약 및 성능 저하) 등을 Hbase를 통해 해결하고 있다.

Hbase와 유사하게 Hadoop의 분산 파일 시스템을 기반으로 개발되고 있는 또 다른 오픈 소스인 Hypertable은 C++ 버전으로, Zvents라는 검색회사와 중국의 포털업체 Baidu에서 지원하고 있다[11].

3. 분산 데이터 저장 시스템

아마존에서는 e-commerce 플랫폼을 제공하면서 얻은 경험을 통해, 다음의 특징을 도출하였다.

대부분의 e-commerce 응용이 데이터베이스 사용시 primary 키 기반의 접근 등 단순 연산 위주이며, 변경 요구는 100% 꼭 수행이 되도록 지원하여야 하며, 대부분의 연산이 수백 ms 안에 수행되어야 한다는 것이다. 이와 같은 특성을 고려하여 Dynamo라는 고가용성, 고확장성의 분산 데이터 저장 시스템을 자체 개발하였다. Dynamo를 실제 e-commerce에 활용한 결과 연말 쇼핑 시즌 등 과부하가 발생하는 기간중에도 동시 사용자 수십만 관리 등 서비스가 가능하다고 한다.

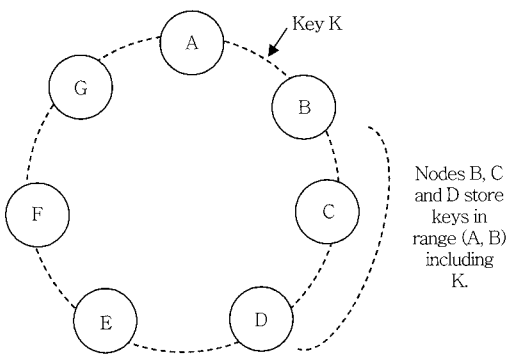
Dynamo의 특징은 다음과 같다[7].

• (key, value) 기반의 저장 모델 제공

Dynamo는 크기가 적은 데이터 관리가 목적으로 데이터를 (key, value)쌍으로 구성된 집합으로 관리하며, 연산도 (key, value)를 저장하는 put(), 키에 의해 값을 얻는 get()처럼 간단한 연산을 지원한다.

• Consistent 해시에 의한 데이터 분산 관리

데이터 분산을 위해 (그림 3)처럼 노드들이 서비스할 데이터 결정을 consistent 해시를 이용한다(해시 값의 범위를 원으로 구성 관리). 노드별로 해시 값에 따른 위치를 부여한 다음, 키에 해시를 적용하여 얻은 값의 다음 위치에 해당하는 노드에서 해당 데이터를 관리한다. 또한 데이터의 분배가 특정 노드로 몰리지 않고 균일하게 분포시키기 위해 가상 노드 개념을 이용한다. 해시에 의해 가상 노드를 선택하고, 실제 가상 노드는 여러 개의 노드로 구성된다. 즉 하나의 노드가 여러 해시 값을 갖는 데이터를 서비스하게 한다.



<자료>: Amazon, 2007.

(그림 3) 데이터 분산 및 복제

• 데이터 가용성을 위한 데이터 복제

데이터의 가용성을 위해 n개의 복제를 유지한다. 해시에 의해 노드가 선택되면, 이 노드에서 일차 데이터를 저장하고, (그림 3)처럼 시계 방향으로 자신의 위치로부터 n-1 안에 있는 노드들에 복제한다.

• Eventually consistent 기반 트랜잭션 관리

Dynamo는 데이터 서비스의 가용성을 높이기 위해 데이터의 일관성은 eventually consistent 유지 정책을 따른다. 즉, 데이터 변경을 복제본에 반영시 낙관적인 정책에 따라 비동기로 수행되며, 충돌이 발생 시에는 나중에 중재하여 최종적으로 데이터의 일관성을 유지한다. 충돌 중재 시기도 read 연산일 때 함으로써, write는 항상 가능하도록 한다. 충돌시

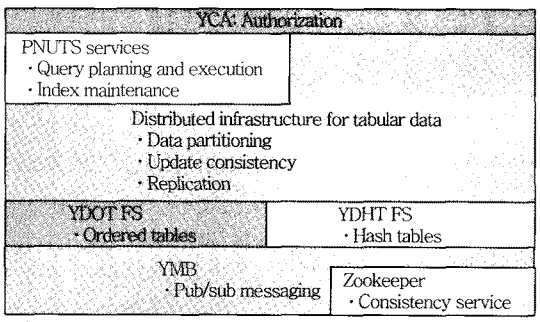
최종 선택을 위한 중재 방법은 응용에서 선택할 수 있게 한다.

• 분산 peer-to-peer 구조

Dynamo는 peer-to-peer 구조로 모든 노드들은 동일한 역할과 책임을 가지므로, 중앙 통제형이 갖는 고장의 위험성을 배제한다.

4. 분산 데이터베이스 관리 시스템

야후에서는 데이터 서비스 플랫폼으로 Sherpa를 개발하고 있으며, (그림 4)처럼 메시지 publish/subscribe 시스템인 YMB, 정렬 테이블로 저장 관리하는 YDOT FS와 해시 테이블로 데이터를 저장하는 YDHT FS 등 두 가지 형태의 데이터 저장 시스템과 잠금 관리 시스템인 Zookeeper를 기반으로 PNUTS 시스템이 구축되어 있다[3]. Zookeeper는 Hadoop에서 구글의 Chubby 대응으로 개발하고 있는 시스템이다.



<자료>: Yahoo, 2008.

(그림 4) Sherpa 구성도

PNUTS는 flickr, del.icio.us 서비스 등 자사의 웹 응용 구축을 위한 호스티드 데이터 관리 서비스를 제공하기 위해 개발되어 실제 활용되고 있다. PNUTS는 온라인 웹 응용을 지원하기 위한 것으로 배치 서비스는 지원 대상이 아니다. PNUTS의 주요 특징은 다음과 같다[8].

• 단순화된 릴레이션 모델 제공

행, 컬럼으로 구성된 릴레이션 모델을 따르며, 스키마 구조에 유연성을 제공하여 같은 테이블에 속하는 모든 행이 모든 컬럼을 가질 필요가 없다. 그리고 연산은 단일 테이블 기반으로 수행되며, SQL 기반 질의어를 제공한다.

• 데이터 접근을 위한 인덱스 제공

정렬 타입과 해시 타입의 저장 관리를 제공하고 사용자가 테이블별로 선택 가능하다.

• 데이터 분산 관리

행 키 기반으로 키 범위 혹은 키에 해시를 적용한 해시 값의 범위에 의해 데이터를 파티션하여 분산 관리한다.

• 데이터 센터간 복제 제공

전세계 분산된 사용자에게 빠른 접근 및 장애 대처를 위해 데이터 센터간 비동기 데이터 복제를 제공한다. 데이터 센터간 복제는 YMB 시스템을 이용한다. 지역적으로 분산된 시스템에 복제는 수백 ms 이상이 소요되고, 사용자 요청은 100 ms 이내에 서비스되어야 하므로 비동기 방식을 이용한다.

• Relaxed consistent 트랜잭션 관리 제공

웹 응용은 일반적으로 한 번에 하나의 레코드를 대상으로 하며, 다른 레코드에 대한 연산은 다른 지역의 요청에 의한 것이라는 분석에 따라 serializability 유지와 eventually consistent 모델의 중간에 해당되는 per record timeline consistency를 지원한다. 즉, 특정 레코드에 변경이 발생시 복제본에 변경 반영을 동일한 순서로 수행함으로써 write 연산은 항상 최신 버전 기반이지만, read 연산은 과거 데이터 기반일 수 있다.

• Master/slave 구조

Tablet controller, router와 storage unit으로 구성된다. Tablet controller는 파티션을 어떻게 분배할지 결정하고 이에 대한 정보와 파티션 구성 정보를 유지한다. Router는 파티션 분배 정보를 메모리내 캐시하여 서비스하고, storage unit은 실제 파티션들의 서비스를 담당한다.

5. 대규모 데이터 저장 관리 기술 비교

대규모 인터넷 서비스용으로 개발된 3개의 데이터 저장 관리 시스템을 <표 1>에 비교한다. 목표로 하는 응용 분야, 데이터 모델, 트랜잭션 처리 및 분산 관리 등의 측면에서 비교한다.

<표 1> 대규모 데이터 저장 관리 기술 비교

분류	Bigtable(GFS)	Dynamo	PNUTS (YDOT, YDHT)
수준	구조 데이터 분산 저장 시스템	분산 저장 시스템	분산 데이터베이스 관리 시스템
응용	온라인, 배치 인터넷 포털 서비스	e-commerce	온라인 인터넷 포털 서비스
시스템 구조	Master/tablet server	Peer-to-peer	Tablet controller/router/storage unit
모델	다차원 맵, 스키마 존재	(key-value) 집합, 구조 없음	단순한 릴레이션
트랜잭션	consistent	eventually consistent	Relaxed consistent
분산 배치	정렬키 범위 기반 파티션	해시키 기반 파티션	키 기반 파티션 (정렬키, 해시키)
복제	3 노드에 동기 복제(GFS 담당)	N 노드에 비동기 복제	데이터 센터간 비동기 복제
DBL	없음	없음	SQL

Ⅲ. 대규모 데이터 처리 기술

1. 대규모 데이터 처리 기술 개요

인터넷 서비스를 제공하기 위해서는 여러 단계의 작업이 필요하다. 예를 들어 웹 포털 검색을 제공하기 위해서는 웹 페이지 자동 수집기에 의해 수집된 웹 페이지로부터 키워드를 추출하고, 이로부터 인덱스를 구축, 저장 관리하며, 이를 이용하여 사용자 검색 요구를 처리하게 된다. 또한 개인 맞춤형 서비스를 제공하기 위해 혹은 회사의 새로운 서비스 전략 수립 등을 위해서는 누적된 데이터를 분석하여 의미있는 정보를 추출하는 과정이 필요하다.

이와 같은 데이터 가공, 분석을 수행하는 배치성의 데이터 처리 업무는 인터넷 데이터의 규모가 지속적으로 증가함에 따라 처리 시간이 수 일 혹은 수십 일 등 소요되므로 이의 단축이 필요하다. 이를 위해 각 응용에서는 독자적으로 병렬 분산 처리를 활용하여 개발해 오다가 이를 공통 플랫폼으로 제공하기 시작했다. 구글에서 2004년 MapReduce 병렬 처리 시스템[14]을 발표 후, 이를 기반으로 한 기술들이 야후, Hadoop 등에서 개발되어, 현재는 대규모 데이터 처리 분야에서는 MapReduce 병렬 처리 모델이 거의 사실 표준이 되고 있다.

본 장에서는 대규모 데이터 분석 처리를 위한 기술개발 현황에 대하여 알아보기 위해, 먼저 MapReduce 분산 병렬 처리 기술에 대하여 조사하고, 이를 기반으로 개발된 데이터 분석용의 데이터플로 언어 처리 기술, 그리고 분산 데이터웨어하우스 기술에 대하여 서술한다.

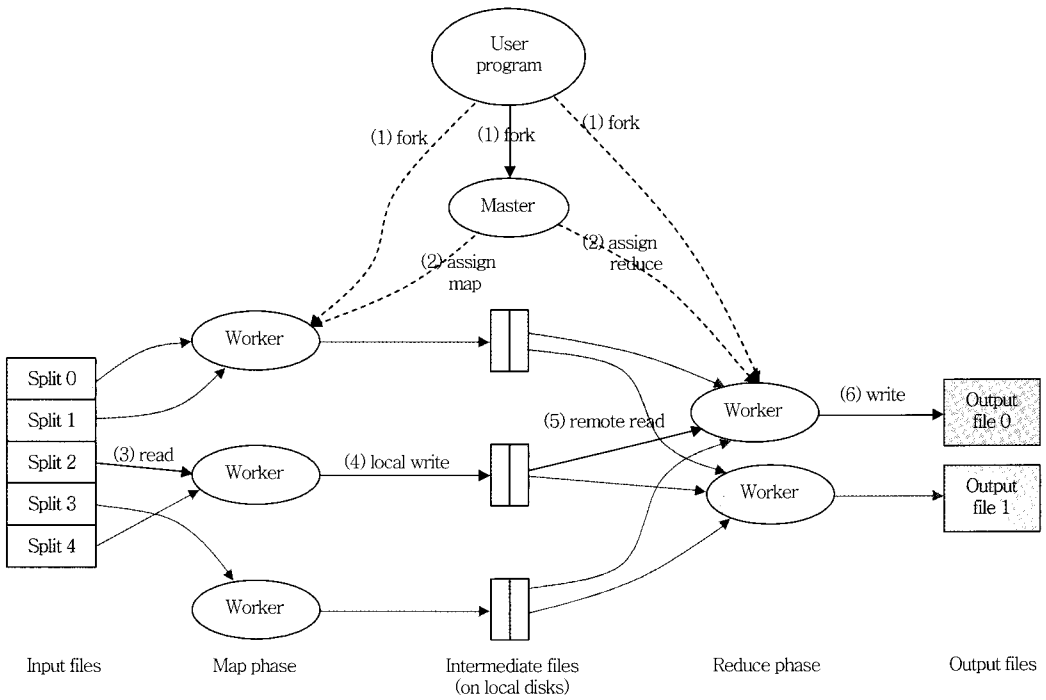
2. MapReduce 분산 병렬 처리 기술

가. 구글의 MapReduce

MapReduce는 인터넷 서비스를 위한 배치 업무들을 빠르게 수행하기 위해 제안된 병렬 처리 모델로 구글의 다양한 서비스(Earth, News, Analytics, 검색, 인덱싱 등)에 적용되고 있다.

MPI와 같은 기존 병렬 처리 기술은 고성능의 컴퓨팅을 요하는 분야의 응용들을 빨리 처리하기 위한 HPC에 초점을 맞추고 있어 데이터량이 많은 경우엔 적용에 문제가 있다. 대규모 데이터 병렬 처리를 위해서는 데이터 증가에 따른 확장성을 제공할 수 있어야 하며, 노드간 데이터 이동에 따른 네트워크 트래픽을 최소화 할 수 있도록 업무 분산이 필요하다. MapReduce는 이와 같은 요구사항을 고려하여 만들어진 시스템이다.

MapReduce는 (key, value) 기반의 데이터를 병렬 처리하는 모델로, (그림 5)에서와 같이 입력 데이터 소스를 기반으로 Map 태스크를 수행하여 중간 결과를 생성하고 이를 입력으로 Reduce 태스



<자료>: Google, 2004.

(그림 5) MapReduce 실행 모델

크를 수행하여 최종 결과를 산출하는 2단계로 구성된다[14].

입력 데이터는 여러 개로 분할되어 분할된 일부 데이터를 대상으로 여러 노드에서 동시에 Map 태스크가 수행된다. 각각의 Map 태스크는 자신에게 할당된 입력 데이터를 대상으로 처리한 결과를 각 노드의 로컬 파일 시스템에 저장한다. Reduce 태스크는 여러 노드에 저장되어 있는 중간 결과를 입력으로 받아 통합 처리하여 최종 결과를 제공한다.

태스크 분배는 네트워크 트래픽 발생을 최소화하기 위해 가능한 한 데이터가 위치한 노드에서 수행하도록 분배한다. 이를 위해 입력 데이터 분할시 데이터의 저장 상태 및 위치를 고려하여 분할한다.

나. Hadoop의 MapReduce

Hadoop에서는 구글의 MapReduce와 동일한 시스템을 개발하고 있다[15]. Lucene라는 텍스트 정보 검색 시스템을 개발하면서 병렬 처리 필요성이 제기되었고, 동일한 목적으로 개발된 구글의 MapReduce 발표를 계기로 이를 오픈 소스로 개발 추진한 것이다. 야후는 Hadoop 프로젝트를 적극 지원하고 있으며 이를 자사 인터넷 서비스에 적극 활용하고 있다. Hadoop의 MapReduce는 IBM, 구글이 대학과 공동으로 구축한 클라우드 컴퓨팅 테스트베드에도 탑재되어 활용되고 있으며, 대학에서도 이를 이용한 분산 병렬 처리 프로그래밍 교육이 이루어지고 있다.

구글의 MapReduce나 Hadoop의 MapReduce는 하나의 Job을 처리하는 데 적합하게 구성되어 있으므로, 여러 Job간 분산 병렬 처리 노드 자원의 공유가 필요한 경우에는 별도의 스케줄러를 활용하거나(구글에서는 WorkQueue라는 Job 스케줄러 이용) HoD[16]처럼 Job별로 가상의 클러스터를 구성 제공하는 기술을 활용한다.

3. 데이터플로 언어 처리 기술

MapReduce 병렬 처리 모델은 키 기반 정렬, 키

기반 결과 Merge 기능과 Map 태스크와 Reduce 태스크로 기본 업무를 정의하는 기능 외 나머지 기능은 사용자가 직접 코딩하여야 한다.

데이터플로 언어 기반 병렬 처리 시스템은 데이터 처리에서 많이 활용되는 기능 등을 시스템이 미리 라이브러리로 구축하고 이를 고급 수준의 언어로 제공함으로써, 사용자는 고급 언어를 이용하여 프로그래밍하면 자동으로 병렬 처리 업무를 생성, 수행하는 시스템이다. 즉, 대규모 데이터 분석 처리를 위해 일반적으로 사용되는 필터링, 집계, 그룹핑, 정렬 등의 기능을 고급 수준의 언어로 제공하고, 사용자는 이 언어를 이용하여 쉽게 데이터 분석 업무를 개발하면 인터프리터나 컴파일러에 의해 분산 병렬 처리가 가능한 업무가 자동 생성되어 수행되는 것이다.

가. 구글의 Sawzall

Sawzall은 이와 같은 대규모 데이터에 대한 분석 처리를 위해 구글에서 개발한 병렬 프로그램 언어로 인터프리트 방식에 의해 언어를 해석하여, 자동으로 일련의 MapReduce 업무를 생성, 수행한다[17].

Sawzall은 데이터 구조 선언문은 Pascal 구문과 유사하고, 제어문, 표현식 등은 C를 기반으로 만들어진 스크립트 언어이다. Sawzall 언어는 MapReduce 모델처럼 외부 입출력 데이터의 형식을 정의하는 기능(프로토콜 버퍼 형식이라고 함)과 이를 기반으로 개개의 입력 데이터에 대해 처리하는 기능을 정의하는 기능, 그리고 처리 결과를 중간 결과로 방출하는 기능, 결과를 집계 처리하는 기능으로 구성된다.

예를 들어 다음 프로그램은 웹 도메인마다 각각 가장 높은 페이지랭크를 갖는 페이지를 구하는 예제이다.

(1) *proto* "document.proto"
(2) max_pagerank_url:
(3) *table maximum*(1) [domain: *string*] *of* url: *string*
(4) *weight* pagerank: *int*;
(5) doc: *Document* = *input*;
(6) *emit* max_pagerank_url[domain(doc.url)] <- doc.url *weight* doc.pagerank;

(1) 라인이 파일로 입출력할 데이터 형식을 정의,
(2)-(4) 라인이 최종 결과를 어떻게 집계할지 정의하는 부분으로 입력 데이터 중 url 필드 값을 집계하는데 도메인별로 pagerank 필드 값이 가장 큰 것들만 집계하라는 의미
(5) 각 입력 데이터 레코드를 읽어 들이라는 의미
(6) 입력 레코드를 처리하여 중간 결과를 방출하라는 의미로 url 필드로부터 도메인 값을 얻고, 도메인별로 url을 중간 결과로 방출하라는 의미이다.
(5)-(6) 라인이 Map 태스크로 구현된 것과 같은 의미이고, (2)-(4)에 정의된 집계방법에 의해 Reduce 태스크가 수행되는 것이다.

나. Hadoop의 Pig

Pig는 Sawzall과 마찬가지로 데이터플로 언어를 지원하는 플랫폼으로 오픈 소스이다[18]. 야후에서 공헌하여 2008년 10월 Hadoop의 서브 프로젝트로 승인된 프로젝트로, 야후에서 수행되는 Hadoop 업무의 30% 이상이 이를 기반으로 개발 활용된다고 한다.

Pig에서 다루는 데이터는 Data Atom, Tuple, Data Bag 혹은 Map으로 구성된다. Bag은 중첩 릴레이션과 유사하고, Map은 (key, value)로 구성된 정보에 대해 key 기반 연산을 제공하는 타입이다. Map 타입은 sparse 릴레이션 데이터(널 값이 많은 테이블), XML와 같은 반구조 데이터(데이터에 따라 구성 집합의 차이가 큰 데이터)에 유용하다.

Pig의 데이터 플로 언어인 Pig Latin은 다음과 같은 5개의 기본 골격으로 구성된다[19].

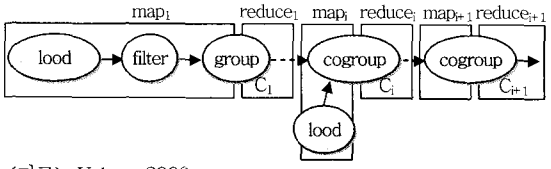
- LOAD: 파일로부터 데이터를 구조화하여 로드하는 방법을 정의하는 것으로 사용자가 정의한 함수 이용도 가능하다.
- FILTER: 로드된 데이터 중 필요없는 데이터를 프레디키트 및 부울린 조건을 주어 제거 가능하다.

- COGROUP: 데이터를 그룹핑하는 것으로 특정 필드 값들을 기준으로 그룹핑, 조인에 의한 그룹핑이 가능하다.
- FOREACH/GENERATE: 데이터를 변환하여 새로운 데이터를 생성한다. 변환 기능으로 필드 값 추출, 함수 호출, 중첩된 Bag 데이터를 평준화하는 flattening, 정렬 기능 등을 제공한다.
- STORE: 결과를 외부 데이터 소스에 저장하는 것으로 저장 방법 명시가 가능하다.

이외에도 두 릴레이션에 대한 catesian product, union 등의 기능을 제공한다.

Pig Latin은 Java 프로그램에 내장하여 사용할 수도 있으며, registerQuery(), openIterator(), Store() 등을 이용하여 Pig Latin을 실행시킬 수 있다.

현재 개발된 Pig 인프라는 (그림 6)에서처럼 Pig Latin을 일련의 MapReduce 업무들로 변환하고 이를 순서대로 실행해 준다[19].



<자료>: Yahoo, 2008.

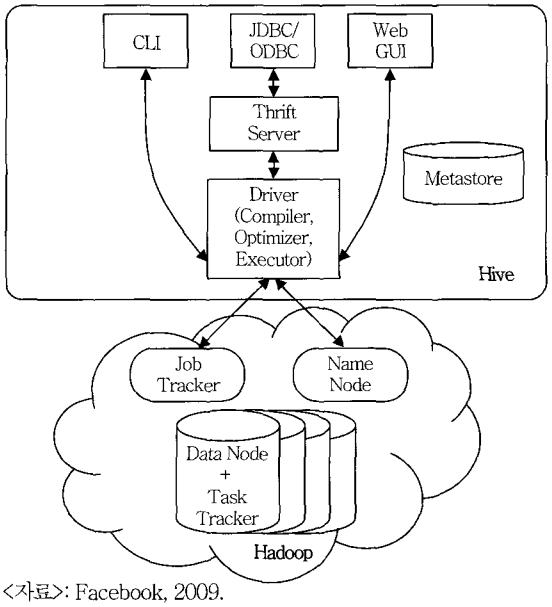
(그림 6) Pig Latin의 컴파일

4. 분산 데이터웨어하우스 기술

데이터웨어하우스 기술도 대규모 데이터 분석을 위해 분산 기술을 적용한다. Hive는 대량의 데이터에 대한 집계, 질의, 분석 등을 쉽게 할 수 있도록 개발되고 있는 데이터웨어하우스용의 오픈 소스이다[20]. Hive 오픈 소스는 Facebook에서 제안한 Hadoop의 서브 프로젝트이다.

Hive는 릴레이션 모델처럼 스키마 구조에 따라 테이블에 데이터를 저장하고, SQL 유사한 HiveQL이라는 언어로 DDL, DML을 지원하고, select, project, join, aggregate, union 등을 지원한다.

Hive에서 데이터는 테이블, 파티션, 버킷으로 계층화되어 저장 관리된다. 파티션은 Bigtable의 키처럼 특정 컬럼들의 값을 기준으로 분할이 되고, 버킷은 파티션 내에서 컬럼의 해시 값에 의해 분할된다. 버킷은 하나의 파일로 대응되어 저장되고, 이들이 모인 파티션이 디렉토리가 되고, 다시 n개의 파티션을 모아 놓은 테이블이 상위 디렉토리가 된다. 데이터를 이와 같이 분할 저장함으로써 이를 기반으로 여러 노드에서 동시 처리가 가능하다.



<자료>: Facebook, 2009.

(그림 7) Hive 구조

Hive는 (그림 7)에서 보듯이 다음의 구성 요소를 갖는다[21].

- 외부 인터페이스: 사용자 인터페이스로 명령어 라인 인터페이스(CLI) 및 웹 기반의 GUI와 JDBC/ODBC와 같은 API 제공
- Thrift server: 다양한 언어에서 사용할 수 있도록 HiveQL 문을 실행하는 API를 제공
- Driver: 사용자 세션 관리 및 질의를 접수
- Compiler: 질의를 파싱, 의미 분석 등을 통해 MapReduce 업무들의 DAG로 구성된 실행 계획을 생성
- Execution Engine: Compiler가 생성한 실행 계획에 따라 실행
- Metastore: Hive에 저장되어 있는 테이블에 대한 메타데이터를 갖는 시스템 카탈로그로, 데이

터가 저장되어 있는 파일에서 데이터를 읽고, 쓰는 데 사용하는 함수, 컬럼에 대한 정보 등 테이블의 구조 정보가 저장 관리된다.

5. 대규모 데이터 처리 기술 비교

대규모 데이터 처리 기술인 MapReduce 병렬 처리 기술, 데이터플로 언어 처리 기술, 분산 데이터웨어하우스 기술에 대하여 <표 2>에서 비교한다.

〈표 2〉 대규모 데이터 처리 기술 비교

분류	MapReduce	Sawzall, Pig	Hive
수준	분산 병렬 데이터 처리	데이터플로 언어 처리	데이터웨어 하우스
대상	파일 데이터베이스	파일	구조를 갖는 파일
데이터 모델	(key, value)로 구성된 Map	구조를 갖는 Bag, Map 등	영속적인 구조를 갖는 Bag
데이터 처리 기능	정렬, 키 기반 그룹핑	정렬, 그룹핑, 조인, 필터링	정렬, 그룹핑, 조인, 필터링
인터페이스	API	Data Flow Language	SQL

Ⅳ. 클라우드 기반 서비스

1. 클라우드 기반 데이터 관리 서비스

클라우드 컴퓨팅 플랫폼은 하나의 기관만 이용하는 전용 클라우드와 여러 기관이 공동으로 사용하는 공용 클라우드로 나눌 수 있다. 앞장에서 설명한 구글의 Bigtable, 야후의 PNUTS 등은 전용 클라우드 기반 데이터 관리 서비스의 활용 사례가 될 수 있다. 구글, 아마존 등 글로벌 인터넷 포털 업체에서는 공용 클라우드 서비스로 데이터 관리 서비스를 제공한다.

클라우드 기반 데이터 관리 서비스는 DBMS를 설치 및 운영 관리하지 않고도 데이터량에 대한 확장성 및 데이터의 가용성을 시스템이 알아서 저비용으로 제공한다. 클라우드 기반 데이터 관리 서비스는 다음과 같이 크게 두 가지로 구분된다.

- 웹 서비스로 데이터 관리 서비스를 제공하는 인프라로서의 서비스
- 데이터베이스 응용을 개발할 수 있도록 프로그래밍 언어로 지원하는 개발 플랫폼으로서의 서비스

클라우드 기반 데이터 관리 서비스는 웹 서비스나 특정 API를 이용하여 이들 기능을 이용하는 응용을 구축하게 되므로, 표준화가 선행되지 않으면 개발된 응용이 특정 플랫폼에 종속되는 결과를 야기하므로 표준화 제정이 필요하다.

가. 인프라 서비스로서의 데이터 관리 서비스

아마존에서는 베타 서비스로 SimpleDB 서비스[22]를 제공한다. 웹 서비스로 데이터 저장 관리 인터페이스를 제공하고, 이를 이용하여 사용자가 아마존 클라우드 플랫폼에 데이터를 저장, 검색 등 관리할 수 있게 해준다. SimpleDB 서비스는 데이터베이스 관리 서비스를 사용자가 직접 사용하는 것이 목적이므로 IaaS 클라우드 서비스인 스토리지 인프라 서비스의 일종이다. 아마존의 S3 스토리지 서비스[23]가 파일 개념의 저장 관리 서비스라면 SimpleDB는 구조화된 레코드 단위의 저장 관리 서비스라는 차이를 갖는다.

SimpleDB 서비스는 데이터 저장을 위해 스키마를 미리 정의할 필요가 없으며, 인덱스 구축도 시스템이 알아서 수행한다. 그러므로 데이터 구조 변화가 자유로우며, 데이터 모델링, 인덱스 구축, 성능 튜닝 등의 DBA 업무가 필요 없다. SimpleDB에서 데이터는 어트리뷰트와 값으로 구성된 아이템의 집

〈표 3〉 SimpleDB 웹 서비스 종류

웹 서비스 명	기능 설명
• 데이터 구축 서비스	
CreateDomain	도메인 생성
DeleteDomain	도메인 삭제
ListDomains	사용자가 정의한 도메인 리스트
DomainMetadata	도메인 생성일자 등 정보 제공
• 데이터 관리 서비스	
Put	데이터 변경 혹은 삽입
Batch Put	일시에 여러 아이템 값 변경 및 삽입
Delete	데이터 삭제
• 데이터 검색 서비스	
GetAttributes	특정 어트리뷰트의 값을 추출
Select	도메인에서 검색 조건에 맞는 데이터만 추출

합인 도메인들로 구성되고, 검색 범위는 특정 도메인을 대상으로 한다.

SimpleDB에서 제공하는 웹 서비스 종류는 <표 3>과 같다.

나. 개발 플랫폼으로서의 데이터 관리 서비스

구글은 클라우드 기반 응용 개발 플랫폼으로 App Engine을 제공하고 있다[24]. 개발자가 App Engine을 이용하여 응용을 구현, 시험 및 운영할 수 있게 한다. App Engine에 통합되어 제공되는 datastore는 데이터 관리 서비스 기능을 제공한다. Datastore에서 제공하는 기능을 Java 혹은 Python 프로그래밍 언어에서 이용하여 개발하고, 이를 구글 플랫폼에 설치, 운영할 수 있게 한다. Datastore 서비스는 Bigtable을 기반으로 개발된 것으로 데이터베이스 언어로 GQL을 제공하고 있다[25].

2. 클라우드 기반 데이터 처리 서비스

앞에서 언급한 것과 같이 대량의 데이터 분석 처리를 위해서는 많은 컴퓨팅 노드가 필요하므로 이를 클라우드 기반 서비스로 활용하는 것이 효과적이다.

아마존에서는 Elastic MapReduce라는 웹 서비스를 제공한다[26]. Elastic MapReduce 서비스는 아직은 베타 서비스로 아마존 EC2[27], S3 인프라 위에서 동작하는 Hadoop 프레임워크를 제공한다. Job을 수행할 EC2 인스턴스 수와 타입(small, large, extra large 등) 정보, S3에 있는 데이터, 응용의 위치 정보를 알려 주면, Elastic MapReduce가 알아서 시스템을 준비하여 실행한다. 제공되는 웹 서비스는 다음과 같다.

- RunJobFlow: 수행 환경을 설정하고 Job Flow를 실행하라는 뜻으로 EC2 인스턴스가 준비되어 가동되면서 처리를 시작한다. Job Flow는 앞의 수행 결과를 받아 다른 단계가 수행되는 일련의 MapReduce 업무들을 의미한다.
- DescribeJobFlow: Job Flow의 상태 정보를 제공한다.
- AddJobFlowSteps: 수행되고 있는 Job Flow에 새로운 업무를 추가할 수 있다.
- TerminateJobFlows: 수행중인 Job Flow를 종료하고 모든 EC2 인스턴스의 가동을 중지한다.

아마존 외에도 HP, 인텔, 야후 등이 주축이 되어 구축하고 있는 글로벌 개방형 클라우드 컴퓨팅인 Open Cirrus 테스트베드에서도 대규모 데이터 처리 서비스에 필요한 기술 검증 등을 위해 Hadoop 기반의 분산 컴퓨팅 플랫폼을 탑재 운영하고 있다[28]. 현재 Hadoop/DFS, MapReduce, Pig 등을 기본 소프트웨어로 운영하고 있다.

이외에도 미국의 클라우드데라[29], 국내 넥스알이라는 회사는 대규모 데이터 처리 응용 분야에 Hadoop 분산 컴퓨팅 플랫폼 기술을 이용한 서비스 구축 기술 및 컨설팅을 비즈니스 모델로 하고 있다.

V. 결론

구글, 야후, 아마존 등 글로벌 인터넷 서비스 업체는 포털, 쇼핑몰 등 특정 영역의 서비스 업체에서 클라우드 서비스 업체로 변모하고 있다. 이는 인터넷 서비스를 지원하면서 축적된 대규모 인터넷 서비스 플랫폼 기술을 활용한 자연스런 서비스 영역의 확대라고 볼 수 있다. 구글은 GFS, Bigtable, MapReduce, Sawzall 등 대규모 인터넷 서비스를 지원하기 위해 개발된 분산 데이터 처리 플랫폼을 기반으로 웹 응용 개발을 지원하는 PaaS 서비스를 제공하기 시작했다. 야후도 오픈 소스 Hadoop(DFS, MapReduce, Pig)과 PNUTS 기술을 이용하여 전용 클라우드 컴퓨팅 플랫폼을 구축, 서비스를 수행하고 있다.

클라우드 기반 대규모 데이터 처리 서비스는 현재 가시화된 인터넷 서비스 외에도 기업의 비즈니스 인텔리전스 서비스, 바이오 데이터 처리 서비스, 시뮬레이션 서비스 등 지속적인 확장이 예상된다.

국내에서도 포털 업체, 데이터센터 업체 등에서 클라우드 컴퓨팅에 대한 많은 관심을 보이고 있다.

또한 인터넷 서비스 플랫폼 기술(GLORY 사업)을 개발하고 있는 ETRI에서는 연구 결과물을 Open Cirrus 클라우드 컴퓨팅 테스트베드에 적용하려고 하고 있다. 아직 클라우드 컴퓨팅 플랫폼의 표준 모델이 정해진 것이 없으므로, 국내에서도 조기 서비스 적용으로 국내 기술의 활성화가 요구된다.

● 용 어 해 설 ●

웹 서비스: 네트워크상에서 시스템간 상호 연동이 가능하도록 설계된 소프트웨어 시스템을 말하는 것으로, 일반적으로 웹상에서 HTTP 프로토콜을 이용하여 호출이 가능한 API를 제공하고, 호출된 서비스는 호스트하는 원격 시스템에서 수행되는 서비스이다.

약어 정리

API	Application Programming Interface
ASF	Apache Software Foundation
DAG	Directed Acyclic Graph
DBA	DataBase Administrator
DBL	DataBase Language
DDL	Data Definition Language
DFS	Distributed File System
DML	Data Manipulation Language
HoD	Hadoop On Demand
IaaS	Infrastructure as a Service
MPI	Message Passing Interface
PaaS	Platform as a Service
RDBMS	Relational DataBase Management System
S3	Simple Storage Service
SaaS	Software as a Service
SQL	Structured Query Language
XML	Extensible Markup Language
YMB	Yahoo! Message Broker

참 고 문 헌

[1] Stephen E. Arnold, "The Google Legacy," infonortics, 2005.

[2] Jeff Dean, "Handling Large Datasets at Google: Current Systems and Future Directions," Data-Intensive Computing Symposium 발표자료, Mar. 2008.

[3] Raghu Rmakrishnan, "Sherpa: Cloud Computing of the Third Kind," Data-Intensive Computing Symposium 발표자료, Mar. 2008.

[4] 정제호, "클라우드 컴퓨팅의 현재와 미래, 그리고 시장전략," 한국소프트웨어진흥원, Oct. 2008.

[5] Armando Fox, "Is Cloud Computing in My Future?," Open Cirrus Summit 발표자료, June 2009.

[6] Fay Chang 외 8인, "Bigtable: A Distributed Storage System for Structured Data," *Seventh Symp. on Operating System Design and Implementation*, Seattle, USA, Nov. 2006.

[7] Giuseppe DeCandia 외 8인, "Dynamo: Amazon's Highly Available Key-value Store," *SOSP'07*, Washington, USA, Oct. 2007.

[8] Brian F. Cooper 외 8인, "PNUTS: Yahoo!'s Hosted Data Serving Platform," *VLDB'08*, Auckland, New Zealand, Aug. 2008.

[9] http://hadoop.apache.org/

[10] http://hadoop.apache.org/hbase

[11] http://www.hypertable.org/

[12] Sanjay Ghemawat 외 2인, "The Google File System," *19th ACM Symp. on Operating Systems Principles*, New York, USA, Oct. 2003.

[13] Mike Burrows, "The Chubby lock service for loosely-coupled distributed systems," *Seventh Symp. on Operating System Design and Implementation*, Seattle, USA, Nov. 2006.

[14] Jeffrey Dean and Sanjay Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Sixth Symp. on Operating System Design and Implementation*, San Francisco, USA, Dec. 2004.

[15] http://hadoop.apache.org/core

[16] http://hadoop.apache.org/core/docs/r.17.2/hod.html

[17] Rob Pike 외 3인, "Interpreting the Data: Parallel Analysis with Sawzall," *Scientific Programming Journal Special Issue on Grids and Worldwide Computing Programming Models and Infrastructure*, Vol.13, No.4, pp.227-298.

[18] http://hadoop.apache.org/pig/

[19] Christopher Olston 외 4인, "Pig Latin: A Not-

So-Foreign Language for Data Processing," *SIGMOD'08*, Vancouver, Canada, June 2008.
[20] http://wiki.apache.org/hadoop/Hive/
[21] Ashish Thusoo 외 8인, "Hive - A Warehousing Solution Over a Map-Reduce Framework," VLDB'09 발표 예정, Lyon, France, Aug. 2009.
[22] http://aws.amazon.com/simpledb/
[23] http://aws.amazon.com/s3/
[24] http://code.google.com/intl/en/appengine/
[25] Joe Gregorio, "Under the Hood of the App Engine Datastore," Cloud Computing Conference & Expo 발표자료, New York, USA, Apr. 2009.
[26] http://aws.amazon.com/elasticmapreduce/
[27] http://aws.amazon.com/EC2/
[28] http://www.cloudera.com
[29] http://www.opencirrus.org