

2013 년 1 월

# Technology Inside

LG CNS R&D Journal

## '빅데이터' 플랫폼의 미래

요약

- I. 빅데이터 플랫폼의 현황
- II. 빅데이터 플랫폼의 진화 방향
- III. 시사점

저자 : 이주열 부책임연구원  
(jooyoul.lee@lgcns.com)

 LG CNS 정보기술연구원

[ 요약 ]

이른바 빅데이터의 시대다. 최근 들어 빅데이터가 주목 받는 가장 큰 이유 중 하나는 빅데이터로부터 과거에는 발견하기 어려웠던 가치 창출이 가능할 만큼 관련 기술이 성숙되었기 때문이다.

빅데이터 플랫폼은 이러한 빅데이터 기술의 집합체이자 기술을 잘 사용할 수 있도록 준비된 기술 환경이다. 기업들은 빅데이터 플랫폼을 사용하여 빅데이터를 수집하고 저장, 관리하며 처리할 수 있다. 빅데이터 플랫폼은 빅데이터를 분석하거나 활용하기 위해 필요한 필수 인프라(Infrastructure)인 셈이다.

오픈소스 하둡(Hadoop)이 빅데이터 플랫폼의 핵심 기술이자 사실상 표준으로 자리매김했으나, 하둡에는 몇 가지 한계가 존재한다. 한계를 극복하기 위해 빅데이터 플랫폼은 실시간 데이터 처리, 다양한 방법의 분산병렬 처리 및 관계형 데이터 모델 지원 등의 방향으로 진화하고 있다.

**빅데이터 플랫폼의 한계와 진화 방향**

| 빅데이터 특성                  | 한계 및 제약                                | 진화 방향   |
|--------------------------|--|---|
| <b>용량<br/>(Volume)</b>   | · 데이터센터 규모의 관리 한계                      | · 범 지구적 규모로 확장 가능<br>(지역 거점 별 데이터센터 확장)       |
| <b>다양성<br/>(Variety)</b> | · 특정 분산병렬 처리 방법 제공<br>· 관계형 데이터 모델 미지원 | · 다양한 분산병렬 처리 방법 제공<br>· 관계형 데이터베이스 및 트랜잭션 제공 |
| <b>속도<br/>(Velocity)</b> | · 일괄처리(실시간 처리 미지원)                     | · 실시간 데이터 조회와 처리                              |

“실시간 데이터 처리”는 빠르게 생산, 소비되는 빅데이터를 빠짐없이 즉각적으로 활용, 분석할 수 있도록 하며, “다양한 분산병렬 처리 방법”은 대규모의 관계도 분석 및 병렬 연산 등을 가능케 한다. 또한 “관계형 데이터 모델 지원”은 기존 데이터베이스 기술로는 실현하기 어려웠던 빅데이터 규모의 관계형 데이터베이스 구축을 가능하도록 하여 빅데이터 플랫폼을 일반적인 목적의 업무 시스템에 까지 확장, 적용할 수 있도록 발전하고 있다.

하지만 이와 같은 새로운 빅데이터 플랫폼은 오픈소스 하둡 2.0 의 발전을 중심으로 서서히 진화하고 있으며, 하둡 2.0 은 아직 개발 중에 있다.

이에 새로운 빅데이터 플랫폼을 적용하기 전에 기존 빅데이터 플랫폼의 한계를 명확히 파악해야 시행착오를 줄일 수 있고, 이를 보완할 새로운 빅데이터 플랫폼의 도입 목적과 적용 영역을 구체화할 수가 있다. 새로운 빅데이터 플랫폼은 그 규모와 기능이 성장한 만큼 시범사례 등으로 더욱 철저한 대비가 필요하다.

## I. 빅데이터 플랫폼의 현황

### 1. 빅데이터의 개념과 가치<sup>1</sup>

- 빅데이터는 형식이 다양하고 빠른 생명주기를 가져 기존 기술로는 관리, 분석하기 어려운 초대용량 데이터
- 빅데이터의 '3V' 특징
  - 초대용량(Volume) : 테라바이트(TB)~페타바이트(PB)급 규모의 데이터
  - 다양한 형태(Variety) : 통일된 구조로 정리가 어려운 비정형데이터가 80% 이상 차지 (예 : 텍스트 문서, 동영상, 이미지, 음악 등)
  - 빠른 속도(Velocity) : 데이터 생성 후 유통에서 활용까지 수분에서 수초 소요
- 빅데이터로부터 과거에는 발견하기 어려웠던 가치 창출이 가능
  - 데이터 규모와 다양성 증대에 따라 데이터 해석에서 발생할 수 있는 정보 손실과 왜곡이 줄어들고 새로운 관계 발견 등의 잠재가치 급증
  - 사건 발생 시점과 감지 시점 사이의 지연이 거의 없는 실시간 분석
  - 문제의 핵심과 가치 원천을 보다 정확하게 분석하는 것이 가능

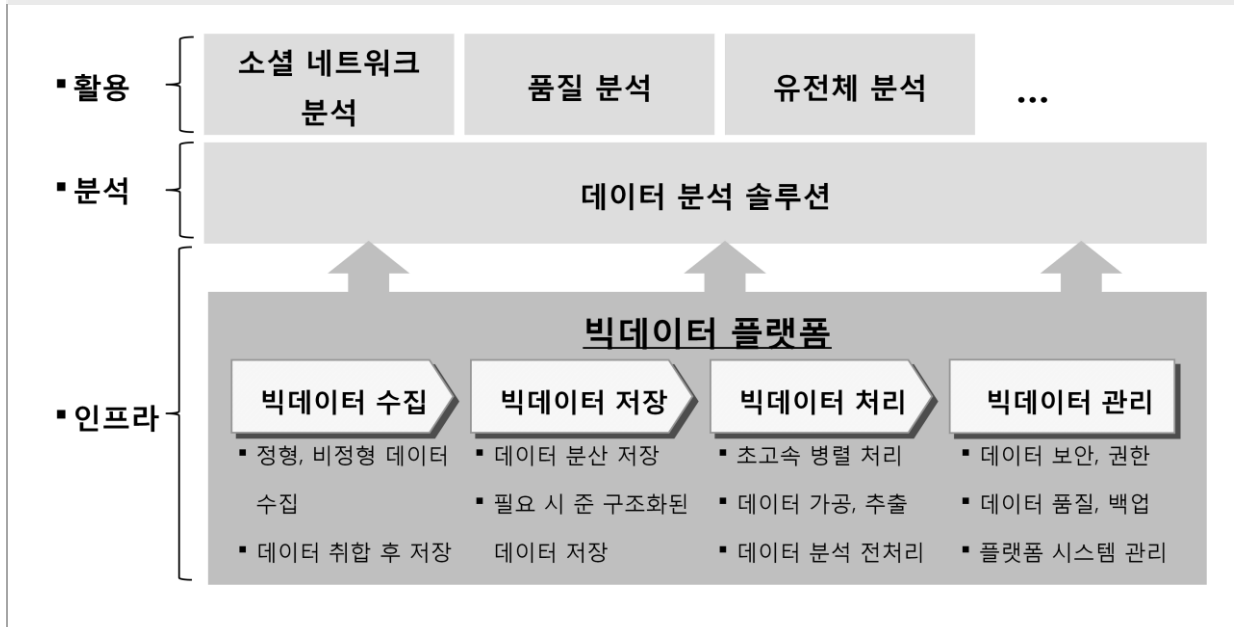
### 2. 빅데이터로부터 가치를 뽑아내는 핵심 기술환경 : 빅데이터 플랫폼

- 빅데이터 플랫폼은 빅데이터 활용과 분석을 위해 공통적으로 사용하는 빅데이터 수집, 저장, 처리, 관리 기술을 제공하는 소프트웨어
  - 빠르게 생성, 유통되는 데이터를 빠짐없이 실시간으로 수집
  - 충분한 저장 공간을 제공하여 버리는 데이터 없이 방대한 양의 데이터를 저장, 축적
  - 빅데이터를 분석 및 활용 가능한 형태로 변환하거나 정보를 추출, 가공하는 데이터 처리
  - 축적된 데이터의 파괴나 유실을 막고, 늘어나는 데이터 용량에 대응하여 확장하고 보안을 유지하는 빅데이터 관리

<sup>1</sup> SERI, '빅데이터: 산업 지각변동의 진원', 2012.05

- 빅데이터 수집부터 처리, 관리까지의 공통 기술과 기능을 일관된 소프트웨어 환경에서 통합 제공

그림-1. 빅데이터 플랫폼의 역할과 기능



□ 오픈소스 하둡은 빅데이터 활용을 가능하게 만든 빅데이터 플랫폼의 핵심 기술이자 사실 표준(De facto)

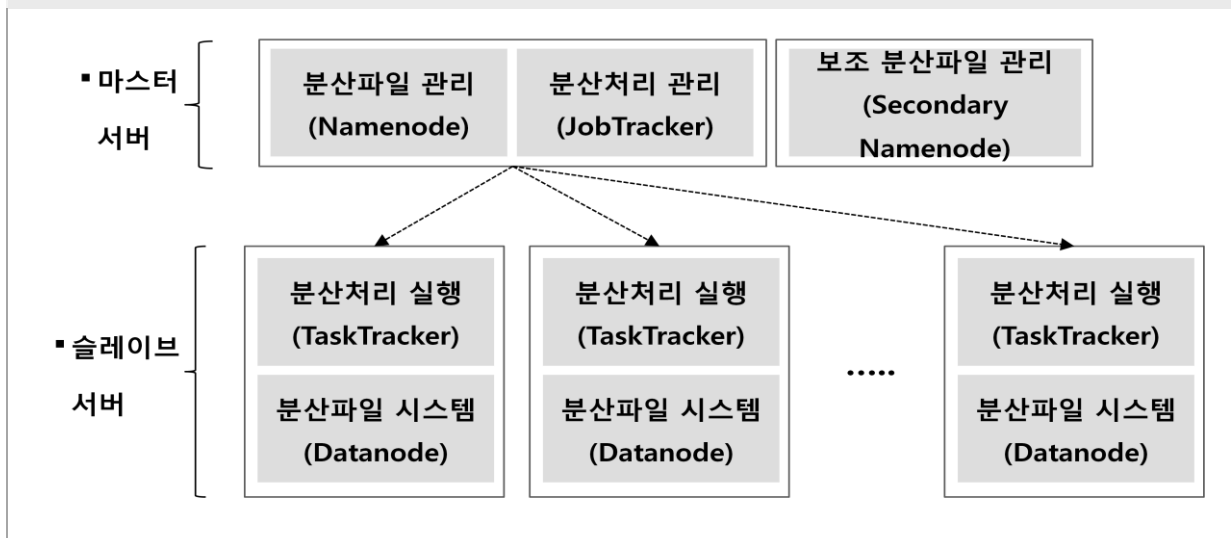
- 하둡은 빅데이터를 저장하는 분산파일 시스템(HDFS, Hadoop Distributed File System)과 분산병렬 처리하는 맵리듀스(MapReduce)로 구성
- 저가 장비(Commodity) 및 스토리지 활용으로 저비용으로 방대한 양의 데이터 저장 및 처리 가능
  - X86 아키텍처의 리눅스 서버와 SATA 디스크 활용으로 고가 장비 대비 50% 저렴
- HDFS : 다양한 형태의 초대용량 데이터를 분산 저장
  - 초대용량 데이터를 분산 저장하여 가상화된 대형 스토리지를 구성하는 분산파일 시스템 제공
  - 내고장성과 높은 확장성 보장

- MapReduce : 분산병렬 처리 방식으로 빅데이터 초고속 처리
  - 시스템의 분산 구조를 감추면서 범용 프로그래밍 언어를 이용해 병렬처리 프로그래밍이 가능한 MapReduce 기법 제공
  - MapReduce 는 비공유 구조(Shared-Nothing)의 순수 대량 병렬 처리(MPP, Massively Parallel Processing) 방식의 일종
- 하둡 에코시스템(Ecosystem) : 하둡은 빅데이터 저장과 처리의 기본적인 기능만 제공하기 때문에, 이의 부족함을 보완하는 다양한 오픈소스 소프트웨어들이 필요 (표-1 참조)

표-1. 하둡 에코시스템 현황

| 구분          | 주요 기술                                | 기술 별 주요 기능  |
|-------------|--------------------------------------|---|
| 빅데이터 수집     | · Flume<br>· Sqoop                   | · 비정형 데이터 수집<br>· 관계형 DB 로부터 데이터 가져오기                                 |
| 빅데이터 저장, 활용 | · HBase                              | · 컬럼 기반 NoSQL 데이터베이스  |
| 빅데이터 처리     | · Hive<br>· Pig<br>· Mahout          | · 유사 SQL 기반 빅데이터 처리<br>· 스크립트 언어 기반 빅데이터 처리<br>· 기계학습 알고리즘 기반 빅데이터 처리 |
| 빅데이터 관리     | · Oozie<br>· HCatalog<br>· Zookeeper | · 빅데이터 처리 과정(Process) 관리<br>· 빅데이터의 메타 정보 관리<br>· 빅데이터 서버 시스템 관리      |

그림-2. 하둡 구성도



### 3. 하둡의 단점으로 본 빅데이터 플랫폼의 기술적 한계

#### □ 실시간 데이터 처리 한계

- 하둡 MapReduce 는 일괄처리(Batch) 방식이기 때문에 실시간 데이터 처리, 조회가 안됨

#### □ 다양한 데이터 처리 한계

- 대규모 계산, 데이터간 통신 및 무결성 보장이 필요한 복잡한 연산 및 처리가 어려움

#### □ 다수의 작은 파일 관리 어려움

- 64 메가바이트(MB)<sup>2</sup> 이하의 작은 파일 저장 시 컴퓨팅 자원(메모리) 낭비 초래

#### □ 비효율적인 데이터 백업 관리

- 3 개의 복제본 파일 관리 방식으로 디스크 공간 낭비와 파일 저장에 낮은 성능
- 스냅샷(Snapshot) 방식, 재해복구(DR, Disaster Recovery) 시스템 등의 고급 데이터 백업 미지원

#### □ 단일고장점(SPoF, Single Point of Failure) 존재

- 하둡에 저장된 데이터의 메타 정보를 관리하는 마스터 서버의 이중화 미지원으로 마스터 서버 장애 발생시 하둡 전체 시스템 중단
  - 마스터 서버 장애에 대한 수동 복구는 가능

#### □ 높은 기술적 숙련도가 요구되고 유지보수가 어려움

- MapReduce 사용에 컴퓨터 프로그래밍 스킬이 요구됨
  - MapReduce 대안으로 하둡의 에코시스템 중 하나인 하이브(Hive)를 활용하여 제한적으로 유사 SQL 방식의 데이터 처리 가능
- 오픈소스 활용 시 기술지원, 유지보수를 스스로 해결해야 하며, 하둡 전문인력 확보가 필요함

---

<sup>2</sup> 하둡 설정으로 변경할 수 있으나 통상 64MB 또는 128MB를 추천함

## II. 빅데이터 플랫폼의 진화 방향

### 1. 구글은 빅데이터 플랫폼의 청사진

#### □ 구글이 발표한 빅데이터 플랫폼 기술 논문에 기초해서 오픈소스 하둡 프로젝트 시작

- 구글이 대용량 웹 데이터 검색을 비용 효율적으로 분산 처리하기 위해 고안한 분산파일 시스템(GFS, Google File System)과 MapReduce 를 각각 2003 년과 2004 년에 논문으로 발표
- 오픈소스 검색엔진<sup>3</sup> 개발자인 더그 커팅<sup>4</sup>이 구글 논문을 기초로 해서 구글 빅데이터 플랫폼의 오픈소스 버전인 하둡을 2004 년부터 개발

#### □ 구글은 오픈소스 빅데이터 플랫폼인 하둡의 지향점이자 청사진

- 구글은 지속적으로 빅데이터와 관련된 자신의 기술과 노하우를 논문으로 발표하고 오픈소스 개발자들은 이를 바탕으로 하둡을 업그레이드하거나 새로운 하둡 에코시스템을 개발
- ※ "구글이 우리에게 방향을 제시했다. 구글은 그들의 GFS 와 MapReduce 논문을 발표하기 시작했고, 우리는 재빠르게 그것을 하둡 프로젝트에 복제했다. 몇 년 동안 구글은 오픈소스 진영에 영감을 준 많은 방법들을 발표했다" (하둡 창시자, 더그 커팅)

표-2. 구글 논문과 하둡, 에코시스템 비교

| 구분     | 구글 논문            | 하둡과 에코시스템          | 설명             |
|--------|------------------|--------------------|----------------|
| 파일 시스템 | 2003 년 GFS       | 2006 년 HDFS        | 분산 파일 시스템      |
|        | 2010 년 Colossus  | -                  | GFS 단점 보완      |
| 데이터 처리 | 2004 년 MapReduce | 2006 년 MapReduce   | 분산병렬처리         |
|        | 2005 년 Sawzall   | 2008 년 Pig, Hive   | 빅데이터 쿼리(일괄처리)  |
|        | 2009 년 Pregel    | 개발 중 Giraph        | 대규모 그래프 데이터 처리 |
|        | 2010 년 Dremel    | 개발 중 Drill, Impala | 실시간 빅데이터 쿼리    |

<sup>3</sup> Lucene, <http://lucene.apache.org/core/>

<sup>4</sup> 더그 커팅 (Douglas Cutting), [http://en.wikipedia.org/wiki/Doug\\_Cutting](http://en.wikipedia.org/wiki/Doug_Cutting)



|        |        |          |        |           |                     |
|--------|--------|----------|--------|-----------|---------------------|
| 데이터베이스 | 2006 년 | BigTable | 2008 년 | HBase     | Schemaless NoSQL DB |
|        | 2012 년 | Spanner  | -      | -         | 빅데이터 트랜잭션 처리        |
| 서버 관리  | 2006 년 | Chubby   | 2008 년 | Zookeeper | 서버 락(Lock) 서비스      |

\*출처: Hadoop World 2012, Beyond Batch 참조하여 보완

## 2. 실시간 빅데이터 처리로 일괄처리 방식 극복

### □ 일괄처리 방식을 뛰어 넘는 실시간 상호 온라인(Online) 데이터

#### 쿼리(Query)로 즉각적인 빅데이터 처리와 조회

- 기존 MapReduce 는 일괄처리 방식으로 빅데이터 처리가 끝날 때까지 결과를 기다려야 하며, 따라서 실시간 빅데이터 처리에 적합하지 않음
- 실시간 빅데이터 처리는 온라인으로 연결된 상태에서 요청과 응답을 즉각 주고 받을 수 있어 다양한 분석과 피드백이 가능

### □ 임시(Ad-hoc) 분석, 트렌드 분석, 실시간 데이터 표출, 경고(Alert) 및 긴급한 데이터 검출 등에 적용

- 구글은 스팸(Spam) 분석, 웹 문서 분석, 안드로이드 마켓 데이터 분석 등 다양한 제품의 품질관리 등에 활용<sup>5</sup>

### □ 준 구조화된(Semi-Structured) 빅데이터 기반의 유사 SQL 로 실시간 처리와 조회 가능

- 준 구조화된 빅데이터를 여러 개의 컬럼 테이블로 변환, 분할 관리하고 분산 시스템 기반의 병렬 SQL 쿼리 실행으로 실시간 처리
- 구글의 Dremel, 클라우데라(Cloudera)의 Impala, 아파치 재단의 Drill 등이 대표적인 구현체이며 Impala, Drill 등은 현재 개발 중

표-3. 실시간 빅데이터 처리 구현체 비교

| 구분            | 구글 Dremel              | 클라우데라 Impala                     | 아파치 재단 Drill                              |
|---------------|------------------------|----------------------------------|---|
| 처리 가능한 데이터 크기 | 기가바이트(GB) ~ 페타바이트( PB) | 기가바이트(GB) ~ 페타바이트(PB)            | 기가바이트(GB) ~ 페타바이트(PB)                     |
| 데이터 처리 방식     | 유사 SQL                 | Impala SQL (HiveSQL 서브셋, 유사 SQL) | · DrSQL(유사 SQL)<br>· Mongo Query Language |

<sup>5</sup> Google Inc, 'Dremel: Interactive Analysis of Web-Scale Datasets', 2010

|               |                                   |                                      |   |
|---------------|-----------------------------------|--------------------------------------|---|
| 처리 가능한 데이터 모델 | 스키마를 가진 중첩 데이터 (Protocol Buffers) | 스키마를 가진 중첩 데이터 (Apache Avro, RCFile) | <ul style="list-style-type: none"> <li>스키마를 가진 중첩 데이터 (Protocol Buffers, Apache Avro, RCFile)</li> <li>스키마가 없는 데이터(JSON, BSON)</li> </ul> |
| 기반 파일 시스템     | GFS, BigTable                     | HDFS, HBase                          | HDFS, HBase   |
| 특징            | 구글 BigQuery 서비스 등에 이미 적용되어 사용 중   | Hive 의 메타스토어 활용                      | <ul style="list-style-type: none"> <li>구글 Dremel 호환</li> <li>다양한 데이터 모델과 호환</li> </ul>  |

\*출처: 각주 5 와 [www.cloudera.com](http://www.cloudera.com), <http://wiki.apache.org/incubator/DrillProposal> 참조 보완

### 3. 다양한 분산병렬 처리 방법 제공으로 단일한 빅데이터 처리 방식 개선

□ 기존 MapReduce 로 처리하기 어려운 그래프 연산, 순환 연산을 지원하여 다양한 빅데이터 처리 가능

- MapReduce 는 분할 병렬과 그 결과의 합산 방식이기 때문에 꼭지점(Vertex)와 선(Edge)을 처리하는 그래프 연산과 조건을 충족할 때 까지 특정 데이터 처리를 반복하는 순환 연산에 비효율적임
- BSP(Bulk Synchronous Parallel) 기반의 병렬 그래프 연산과 순환 연산에 MPI(Message Passing Interface) 방식의 병렬 처리 지원으로 MapReduce 의 데이터 처리 방식 보완 (표-4 참조)

□ 소셜 네트워크 관계 분석, 웹 페이지 링크 분석 등에 그래프 연산 적용, 대규모 계산 및 일반적인 고속 병렬 연산에 MPI 적용

- 구글은 웹 검색 순위를 평가하기 위해 웹 페이지 링크 분석에 기반한 PageRank<sup>6</sup> 알고리즘을 사용하며, 매일 1 조가 넘는 웹 페이지 관계를 분석하기 위해 그래프 연산 수행<sup>7</sup>

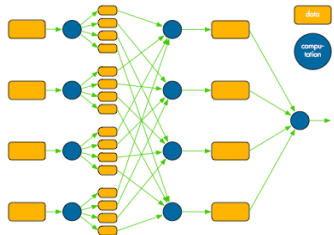
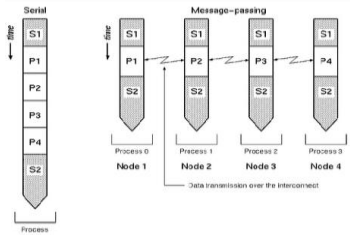
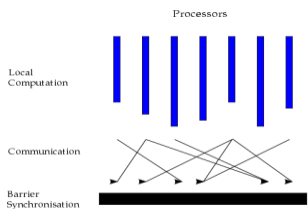
□ MapReduce 외 그래프 및 MPI 등 다양한 데이터 처리 라이브러리를 지원하도록 보편적 분산병렬 프레임워크로 진화

<sup>6</sup> 웹과 같은 링크 구조를 가지는 문서의 연결 분석으로 상대적 중요도에 따라 가중치를 부여하는 방법

<sup>7</sup> Google Inc, 'Pregel: A System for Large-Scale Graph Processing', 2009

- 그래프 연산은 아파치 재단의 Giraph<sup>8</sup>, MPI 는 오픈소스인 OpenMPI<sup>9</sup>가 대표적인 라이브러리

표-4. 분산처리 기술 비교

| 구분 | MapReduce  | MPI   | BSP  |
|----|--|---|--|
| 관점 | 대규모 데이터 처리   | 고속의 컴퓨터 연산  | 병렬 컴퓨터 연산  |
| 특징 | <ul style="list-style-type: none"> <li>· 서버에 분배되는 데이터를 주로 디스크에 저장</li> <li>· 분산 서버에 독립 프로세스가 데이터를 읽어 짧은 시간에 수행된 결과를 다시 디스크에 저장</li> <li>· 서버간 데이터 통신은 대부분 파일 I/O</li> <li>· 데이터 입/출력이 빈번하여 분산파일시스템 인프라 요구</li> </ul> | <ul style="list-style-type: none"> <li>· 각 서버에 분배되는 데이터를 주로 메모리에 저장(분산된 메모리를 한 메모리처럼 사용)</li> <li>· 수행하는 작업의 대부분은 연산 작업(CPU 소비)</li> <li>· 서버간 데이터 교환이 빈번하여 고속 네트워크 인프라 요구</li> <li>· 서버 장애 발생시 연관된 다른 노드의 프로세스 재수행에 장시간 소요, 노드간 데이터 교환 문제 빈번히 발생 가능</li> </ul> | <ul style="list-style-type: none"> <li>· 각 서버의 프로세스 마다 로컬 메모리 사용</li> <li>· 프로세스간 통신과 동기화 단계를 분리</li> <li>· 각 프로세스들 독립적으로 자신의 연산을 수행</li> <li>· 타 프로세스들간 데이터 메시지 교환</li> <li>· 모든 프로세스간의 데이터 교환이 끝나고, 동기화</li> </ul> |
| 구조 |   |   |   |
| 활용 | 대규모 데이터 처리   | 고속 네트워크 기반 대규모 계산 및 시뮬레이션   | 그래프 데이터 병렬 처리  |

\*출처: LG CNS Technology Inside 13 호 "빅데이터 처리 기술 동향과 적용사례" 참조 보완

#### 4. 관계형 데이터 모델과 대규모 업무 트랜잭션 지원으로 빅데이터 기술 적용 영역 확대

<sup>8</sup> <http://incubator.apache.org/giraph/>

<sup>9</sup> <http://www.open-mpi.org>

□ **관계형 데이터베이스에 NoSQL<sup>10</sup>의 확장성과 고성능 기능을 부여하여 빅데이터의 저장, 트랜잭션 및 SQL 처리 가능**

- 기존 빅데이터 플랫폼에서 주요하게 채용된 NoSQL 은 분산처리와 확장성이 뛰어나지만 스키마와 관계형 데이터 모델이 지원되지 않는 데이터베이스
- 일반적 목적의 트랜잭션 처리나 관계형 데이터 모델이 가능하면서도 빅데이터를 처리할 수 있도록 보완한 NewSQL 등장

□ **기존 관계형 데이터베이스나 NoSQL 이 처리 하지 못 하는 빅데이터 관리 업무에 적용 가능**

- 구글은 광고 데이터 관리와 내부적으로 사용하던 관계형 데이터베이스를 대체하고자 자체 개발한 NewSQL 인 Spanner 를 적용<sup>11</sup>

□ **분산된 데이터베이스들을 절대 시간 기준으로 동기화하여 동시성을 제어하고 분산 트랜잭션 및 두 단계 커밋(two-phase commit) 제공**

- 구글의 Spanner 는 GPS 와 원자시계(Atomic clock)를 동원하여 절대 시간을 측정하고, 각 분산 트랜잭션별 발생 시간(timestamp)을 공유, 동기화하여 데이터 읽고, 쓰기 동시성 제어

## 5. 작은 파일 관리와 데이터 백업의 효율화

□ **작은 파일(64MB 미만)들을 대규모로 저장, 관리하고 복제가 아닌 원본 데이터 복구 방식의 데이터 백업 효율화**

- 기존에는 저장하는 파일의 메타 정보(파일 이름, 위치, 크기 등)를 모두 메모리에 관리하는 방식으로 대규모 개수의 작은 파일들을 저장할 경우, 메타 정보를 관리할 마스터 서버의 메모리 공간이 부족
- 기존 데이터 백업 시 기본으로 세 개의 복제본을 추가 저장, 관리하여 스토리지 공간을 낭비

<sup>10</sup> NoSQL은 Not only SQL의 약자로, 유연한 확장성과 고가용성을 제공하지만 표준 SQL 및 엔티티간의 관계를 지원하지 않는 데이터베이스 시스템

<sup>11</sup> Google Inc, 'Spanner: Google's Globally-Distributed Database', 2012

- **작은 파일을 대량으로 관리하는 일반적인 서비스들에 적용 가능**
  - 개인이 취급하는 작은 파일들을 대량으로 저장, 관리 가능
  - 애플 iCloud, 아마존 S3, DropBox 와 같은 개인 사용자를 위한 스토리지 서비스에 적용 가능
- **저장하는 파일의 메타 정보를 분류, 분산하여 작은 파일들을 관리하고 데이터 복구 정보 기반의 효율적인 데이터 백업 관리**
  - 저장하는 파일들의 메타 정보를 분산 해시 테이블(Distributed Hash Table)로 관리
  - 자료를 일부 잃어 버려도 나머지 자료로 원본 복구가 가능한 Erasure Code<sup>12</sup> 알고리즘으로 데이터 복구 정보 관리 → 기존 방식 대비 절반 정도의 스토리지 공간 절약 가능

## 6. 빅데이터 플랫폼의 안전성 개선

- **기존에 존재하던 단일고장점(SPoF)을 시스템 이중화로 극복**
  - 저장된 파일들의 메타 정보를 관리하는 마스터 서버를 이중화(Hot Standby)하여 문제 발생시 자동으로 시스템 연속성 보장
- **외부 이중화 솔루션 없이 자체 이중화로 비용 효율화와 관리성 편의 도모**
  - 기존 마스터 서버 이중화에 DRBD<sup>13</sup>와 같은 외부 솔루션을 적용하였으나 설정과 관리에 있어 많은 공수가 소요
- **외부 컨트롤러로 시스템 상태를 모니터링하고 문제 발생시 대기 중인 시스템으로 전환, 관리**
  - 활동 중인 마스터 서버와 대기 중인 마스터 서버 각각에 감시 컨트롤러가 연결
  - 활동 중인 마스터 서버 문제 발생시 감시 컨트롤러가 즉각 인지하고 대기 중인 마스터 서버의 컨트롤러에게 제어권을 넘겨주는 방식

<sup>12</sup> [http://en.wikipedia.org/wiki/Erasure\\_code](http://en.wikipedia.org/wiki/Erasure_code)

<sup>13</sup> DRBD (Distributed Replicated Block Device): 이중화 서버 클러스터를 구축하기 위한 블럭디바이스

### III. 시사점

#### 1. 기존 빅데이터 플랫폼의 한계 파악으로 시행착오 방지

##### □ 빅데이터 플랫폼 도입 시 기존 플랫폼의 한계를 파악하여 업무 구현에 제약을 초래하는지 검토 필요

- 기능, 비기능적인 요구사항 별로 현재 빅데이터 플랫폼의 한계를 식별하고 이에 대한 검증 필요
  - 빅데이터 플랫폼은 새로운 기술이고 전문가와 전문업체가 부족한 단계인 만큼 도입 목적에 부합하는지 철저하게 개념증명(PoC, Proof of Concept) 및 테스트 필요
  - 작은 파일들의 관리나 실시간 처리와 같이 간과하기 쉬운 기능적 한계와 마스터 서버 이중화 미비에 대한 이해 필수
- 과도한 빅데이터 기술 중심적 문제 해결 지양
  - 기존 데이터베이스 기술로 해결할 수 있는 사항에 빅데이터 플랫폼을 과잉 적용하지 않도록 유의

##### □ 상용 플랫폼 또는 상용 서비스 도입 고려

- 현재 빅데이터 플랫폼은 오픈소스 하둡의 한계를 그대로 계승하나 최근에 이러한 한계를 일부 극복한 상용 빅데이터 플랫폼이 출시됨
- 하둡은 오픈소스인 만큼 누구나 사용 가능하고 해당 기술을 습득할 수 있으나 빠르게 진화하고 높은 기술 난이도 때문에 상용 서비스 및 전문 업체의 활용 고려

#### 2. 빅데이터 플랫폼의 진화 방향을 인식하고 대비

##### □ 빅데이터 플랫폼은 실시간 처리 속도, 처리 방식의 다양화, 관계형 데이터 모델 지원 등의 방향으로 진화

- 일괄 처리에 의한 결과 대기가 아니라 즉각적인 실시간 빅데이터 처리
- 분할하여 병렬 처리하고 병합하는 단순 데이터 처리 방식에서 벗어나 대규모 계산 등의 일반 연산 처리 가능

- 스키마가 없는 단순 데이터 모델이 아닌 관계형 데이터베이스(RDB) 수준의 관계형 데이터 모델을 저장하고 처리

□ **하둡 2.0 이 새로운 빅데이터 플랫폼의 핵심이나 아직 시험 단계**

- 오픈소스 하둡 2.0 버전과 Impala, Drill 등 기존의 한계를 개선한 신규 기술들은 아직 개발 중이거나 시험 테스트 중
- 하둡 2.0 의 정식 출시 일정에 따라 기타 하둡 에코시스템의 업그레이드가 지속적으로 진행
  - 하둡 2.0 은 현재 알파 버전 단계로 최종 출시까지 베타, 출시 후보(RC, Release Candidate) 등 두 단계가 남음
  - 하둡 2.0 출시 이후에 호환성이 확보된 에코시스템 업그레이드 진행이 가능

□ **하둡 1.0 부터 단계적으로 역량을 구축하여 하둡 2.0 및 새로운 빅데이터 플랫폼 대비**

- 빅데이터 플랫폼은 큰 폭으로 진화하고 있으나 기본 사상과 기술 패러다임은 변화하지 않음
- 당장 사용이 가능한 하둡 1.0 을 중심으로 빅데이터 축적, 관리, 처리 역량을 배양하고 한계를 체감
- 하둡 2.0 알파 버전 테스트와 PoC 등으로 하둡 2.0 전환 연착륙 준비

**3. 미래의 빅데이터 플랫폼은 빅데이터의 '3V' 특성을 충분하게 충족**

- **Volume** : 페타바이트(PB) 규모를 넘어 엑사바이트(EB), 제타바이트(ZB) 까지 저장, 관리하고 단일 지역이 아닌 범 지구적으로 분산된 데이터 시스템을 마치 하나의 시스템처럼 제공
- **Variety** : 정형, 비정형 데이터 처리 뿐 아니라 보편적 분산병렬처리 지원으로 빅데이터 활용 및 분석 목적에 맞는 다양한 빅데이터 처리 방법 제공
- **Velocity** : 실시간으로 생성, 유통되는 빅데이터를 실시간으로 수집하여 실시간으로 처리, 분석하는 기민성 제공

[ 참고문헌 ]

- 채승병 수석연구원 (2012). 『빅데이터 : 산업 지각변동의 진원』 SERI CEO Information 제 851 호
- 임혜순 책임연구원 (2011). 『빅데이터 처리 기술 동향과 적용사례』 LG CNS Technology Inside 13 호
- Grzegorz Malewicz et al. (2009). 『Pregel : A System for Large-Scale Graph Processing』 ACM PODC 2009
- Sergey Melnik et al. (2010). 『Dremel : Interactive Analysis of Web-Scale Datasets』 VLDB 2010
- Fikes, Andrew (2010). 『Storage Architecture and Challenges』 TechTalk, Google
- James C. Corbett et al. (2012). 『Spanner : Google's Globally-Distributed Database』 OSDI 2012
- Tom White (2012). 『Hadoop : The Definitive Guide, 3<sup>rd</sup> Edition』 O'REILLY
- Eric Sammer (2012). 『Hadoop Operations』 O'REILLY