빅 데이터 분석 분야
Stack 통합 테스트 결과보고서
[Hive]

2012. 11.

nipa 정보통신산업진흥원 National IT Industry Promotion Agency
nipa 공개SW역량프라자 Open Source Software COMPENTENCE PLAZA

목 차

I. Stack 통합 테스트 개요

공개SW Stack 통합테스트는 여러 공개SW들의 조합으로 시스템 Stack을 구성한 후 Stack을 구성하는 공개SW의 상호운용성에 중점을 두고 기능 및 성능테스트 시나리오를 개발하여 테스트를 진행한다.

본 통합테스트를 통해 안정된 Stack 정보를 제공하여 민간 및 공공 정보시스템 도입 시 활용될 수 있도록 한다.

1. 목적

□ 공개SW Stack 통합 테스트 수행 목적

- o 공개SW로 구성된 Stack이 유기적으로 잘 동작함을 확인
- o 다양한 Stack 구성에 기반을 둔 테스트를 통해 안정된 Stack 조합 규명
- o 공개SW 시스템 도입을 위한 Stack 참조모델의 신뢰성 정보로 활용
- o 공개SW의 신뢰성과 범용성에 대한 사용자 인식 제고

II. 빅 데이터 소개

1. 빅 데이터(Big data)

빅 데이터란 기존 데이터베이스[1] 관리도구의 데이터 수집, 저장, 관리, 분석의 역량을 넘어서는 대량의 정형[2] 또는 비정형[3] 데이터 세트 및 이러한 데이터로부터 가치를 추출하고 결과를 분석하는 기술을 의미한다.

기존 빅 데이터의 개념이 단순히 데이터의 양이 많은 것을 의미했다면, 최근의 일반적인 빅 데이터의 개념은 기존 데이터에 비해 너무 방대해 일반적으로 사용하는 방법이나 도구로 수집, 저장, 검색, 분석, 시각화 등을 하기 어려운 정형 또는 비정형 데이터세트를 의미한다.

다양한 종류의 대규모 데이터의 생성, 수집, 분석, 표현을 특징으로 하는 빅 데이터 기술의 발전은, 다변화된 현대 사회를 정확하게 예측하여 효율적으로 작동하게 하고, 개인화된 현대 사회 구성원 마다 맞춤형 정보를 제공, 관리, 분석을 가능하게 하며, 과거에는 불가능 했던 기술을 실현시키기도 한다.

이같은 빅 데이터는 정치, 사회, 경제, 문화, 과학 기술 등 전 영역에 걸쳐 사회와 인류에게 가치있는 정보를 제공하며, 그 중요성 또한 부각되고 있다.

(출처 : 위키백과)

1) 여러 응용 시스템들의 통합된 정보들을 저장하여 운영할 수 있는 공용 데이터의 묶음

2) 구조가 있는 데이터의 집합에 대하여 정확한 질의어를 사용하여 조건에 맞는 결과의 집합을 도출

3) 구조가 없는 데이터에 대하여 자유로운 형식의 모호성이 있는 질의어를 사용하여 관련된 결과 목록을 도출

□ 빅데이터 분석 기술

대부분의 빅 데이터 분석 기술과 방법들은 기존 통계학과 전산학에서 사용되던 데이터 마이닝[4], 기계 학습[5], 자연 언어 처리[6], 패턴 인식[7] 등이 해당된다. 특히 최근 소셜 미디어 등 비정형 데이터의 증가로 인해, 분석 기법들 중에서 텍스트 마이닝, 평판 분석, 소셜네트워크 분석, 군집 분석 등이 주목 받고 있다.

o Text Mining
비정형 또는 반정형 텍스트 데이터에서 자연 언어 처리 기술에 기반하여 유용한 정보를 추출, 가공하는 것을 목적으로 하는 기술

o 평판 분석(Opinion mining)
소셜미디어 등의 정형 또는 비정형 텍스트의 긍정, 부정, 중립의 선호도를 판별하는 기술

o 소셜 네트워크 분석(Social network analysis)
소셜 네트워크 연결구조 및 연결강도 등을 바탕으로 사용자의 명성 및 영향력을 측정하는 기술

o 군집 분석(Cluster Analysis)
비슷한 특성을 가진 개체를 합쳐가면서 최종적으로 유사 특성의 군을 발굴하는데 사용

대규모의 정형 및 비정형 데이터를 처리하는 데 있어 가장 기본적인 분석 인프라로 하둡[8]이 있으며, 데이터를 유연하고 더욱 빠르게 처리하기 위해 NoSQL 기술이 활용되기도 한다.

(출처 : 위키백과)

---

4) 대규모로 저장된 데이터 안에서 체계적이고 자동적으로 통계적 규칙이나 패턴을 도출

5) 인공 지능의 한 분야로, 컴퓨터가 학습할 수 있도록 하는 알고리즘과 기술을 개발하는 분야

6) 인간이 발화하는 언어 현상을 기계적으로 분석해서 컴퓨터가 이해할 수 있는 형태로 만드는 자연 언어 이해 혹은 그러한 형태를 다시 인간이 이해할 수 있는 언어로 표현하는 제반 기술

7) 계산이 가능한 기계적인 장치가 어떠한 대상을 인식하는 문제를 다루는 인공지능의 한 분야

8) 대량의 자료를 처리할 수 있는 컴퓨터 클러스터에서 동작하는 분산 응용 프로그램을 지원하는 자유 자바 소프트웨어 프레임워크

2. 빅 데이터 분석 분야 주요 공개SW

[표 II-1. 주요 공개SW]

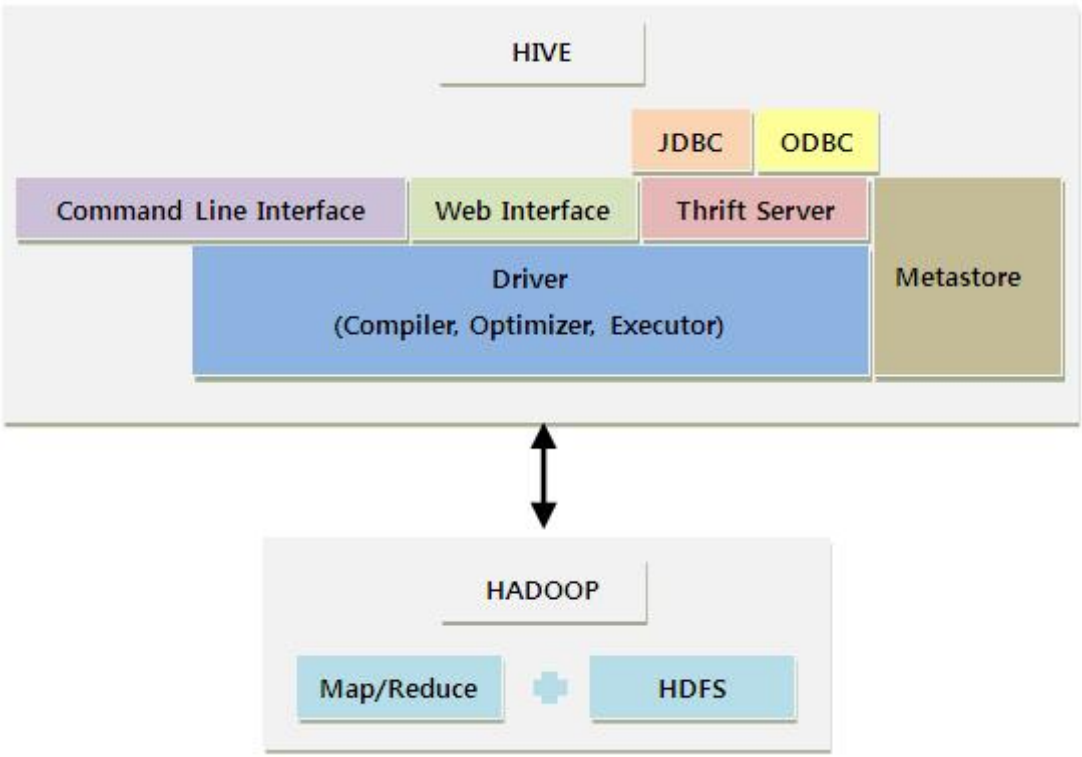
제품명	Stack 환경		홈페이지	비고
Hive	OS	Cross-platform	http://hive.apache.org	Apache License 2.0
	JAVA	JAVA 1.6.0 이상		
Pig	OS	Cross-platform	http://pig.apache.org/	Apache License 2.0
	JAVA	JAVA 1.6.0 이상		
R	OS	Cross-platform	http://www.r-project.org/	GNU GPL
MapReduce	OS	Cross-platform	http://hadoop.apache.org/	Apache License 2.0
	JAVA	JAVA 1.6.0 이상		

III. 테스트 대상 소개

1. Hive 소개

Hive는 데이터 요약, 쿼리, 분석을 제공하는 하둡 기반의 데이터 웨어하우스 시스템으로 데이터의 분석 및 요약을 좀더 쉽게 사용하기 위하여 설계 되었다. Hive는 Hive QL이라는 SQL 베이스의 쿼리를 제공한다.

Hive는 온라인 트랜젝션 프로세싱을 위하여 디자인 되지 않았다. 그리고 실시간 쿼리와 열단위 업데이트를 제공하지 않는다. Hive는 웹로그 같은 큰 데이터로, 비정형 데이터를 배치로 수행하여 결과를 얻을 때 가장 좋다.



[그림 III-1. Hive 구조]

IV. Stack 통합 테스트

1. 테스트 환경

□ Hive 환경

[표 IV-1. Hive 환경]

모듈	Version
Hive	0.9.0

□ Stack 환경

[표 IV-2. Stack 환경]

구성	OS	Java	Hadoop
Master Node	CentOS 6.3(64bit)	1.6.0_35	1.0.3
Slave Node 1	CentOS 6.3(64bit)	1.6.0_35	1.0.3
Slave Node 2	CentOS 6.3(64bit)	1.6.0_35	1.0.3
Slave Node 3	CentOS 6.3(64bit)	1.6.0_35	1.0.3

□ HW 환경

[표 IV-3. HW 환경]

제조사	모델명	CPU	MEM	Disk	NIC
IBM	X3550M2	Intel Xeon(R)CPU 2.40GHz * 4	8GB	320GB	Gigabit 1Port

※ 동일 사양의 HW Stack 구성

2. 주요 테스트 방법

□ 시나리오 테스트

시나리오 테스트 기법은 단일 기능에 대한 결함 여부를 확인하는 것이 아니라, 서로 다른 컴포넌트 사이의 상호작용과 간섭으로 발생할 수 있는 결함을 발견하기 위한 기법이다.

본 테스트에서는 사용자 시나리오 테스트 기법을 적용하여 Hive를 사용하는 사용자들이 사용할 수 있는 항목 중 Data Definition, Data Manipulation, Queries에 대한 사용자 시나리오를 도출하였다. 각각의 항목에서 도출한 세부 시나리오는 사용자가 일반적으로 수행할 수 있는 시나리오를 추출하여 테스트케이스로 작성하였다.

□ 상호 운용성 기반 테스트

상호 운용성은 서로 다른 기술로 이루어진 제품이나 서비스가 상호작용 상의 오류가 없는지 검증하는 기법으로, 본 테스트에서는 애플리케이션이 지원하는 Stack을 구성하여 애플리케이션과 Stack 환경 사이의 상호작용 상의 동작여부를 검증하였다.

3. 기능 테스트 수행 결과

기능 테스트 수행 관련 세부 시나리오는 별첨「Hive 테스트 케이스」문서를 참고한다.

□ 테스트 시나리오 현황

[표 IV-4. 테스트 시나리오 현황]

기능	테스트 시나리오	테스트 케이스
Data Definition	7	29
Data Manipulation	2	6
Queries	3	12
합 계	12	47

□ 테스트 결과

기능 테스트 시나리오를 통한 테스트 수행 결과 Data Definition, Data Manipulation, Queries 시나리오 상의 모든 기능이 예상 결과와 동일하게 동작함을 확인하였다.

[표 IV-5. 테스트 결과]

분류		PASS	FAIL	N/A
기능	개수			
Data Definition	29	29	0	0
Data Manipulation	6	6	0	0
Queries	12	12	0	0

4. 성능 테스트 수행 결과

성능 테스트의 경우 하드웨어 사양뿐 아니라, OS 및 애플리케이션 환경 구성에 따라 성능 측정 결과가 상이하므로, 실제 운영 시스템 환경에 따라 테스트 결과가 다를 수 있다.

본 성능 테스트는 Hive 시스템이 가동되는 상황에서 부하 시나리오를 재현하여, 데이터의 용량에 따른 데이터 분석 속도 차이와 Slave Node의 개수에 따른 데이터 분석 속도의 차이 및 Server의 자원 사용률을 측정하였다.

□ 테스트 시나리오

[표 IV-6. 테스트 시나리오]

시나리오 ID	시나리오
SC_BDA_H_1	데이터의 용량에 따른 분석 속도 측정
SC_BDA_H_2	Slave Node의 개수에 따른 분석 속도 측정
SC_BDA_H_3	데이터 분석 시 Slave Node 개수에 따른 Server 자원 사용률 측정

□ 서버 구성

[표 IV-7. 서버 설정 정보]

구분		항목
HADOOP	hdfs	dfs.block.size = 67108864(64MB)
	mapred	mapred.map = Xmx1024m(1G) mapred.reduce = Xmx1024m(1G)

□ 측정항목

[표 IV-8. 측정항목]

항목	내용
분석 속도	데이터 용량에 따른 Hive 데이터 분석 속도를 측정
	Slave Node 개수에 따른 Hive 데이터 분석 속도를 측정
CPU 사용률	프로세스에서 CPU(Central Processing Unit)를 사용한 비율(%)
메모리 사용률	Physiscal 메모리를 사용량

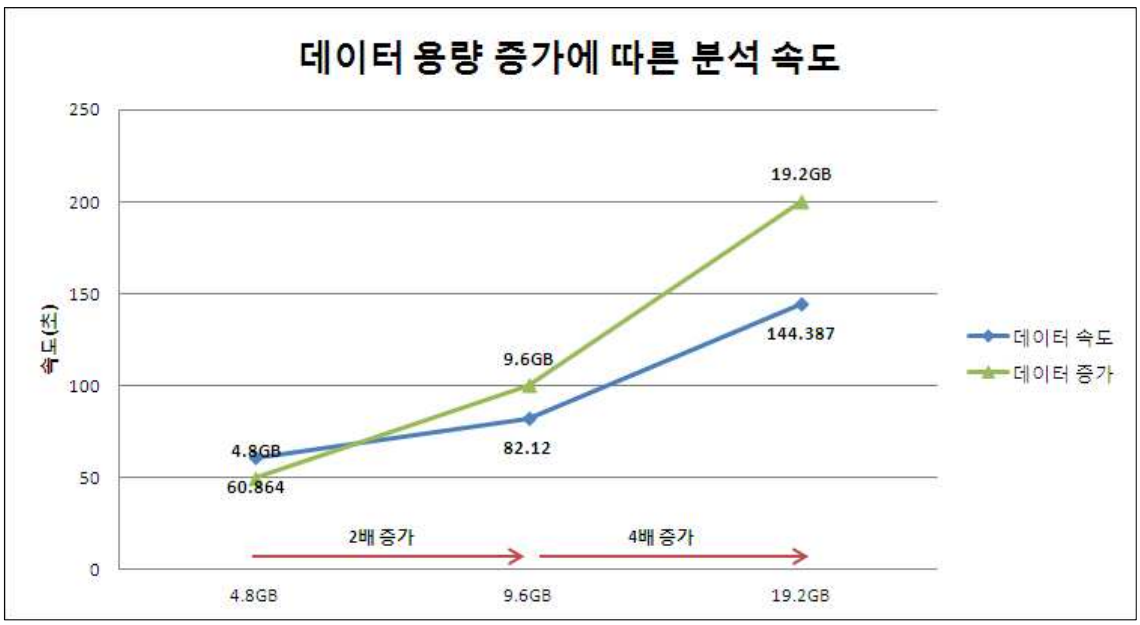
□ 테스트 결과

o 데이터의 용량에 따른 Hive의 분석 속도 측정

- 수행정보

수행 조건	- 4.8GB의 28509544행, 16열의 Data 준비 - 9.6GB의 57019088행, 16열의 Data 준비 - 19.2GB의 114038176행, 16열의 Data 준비 - Hive에서 table 생성 후 데이터 load - Master Node 와 Slave Node 1, 2, 3 연동

- 전체 데이터의 행 수와 3번째 열 값의 평균을 분석 요청

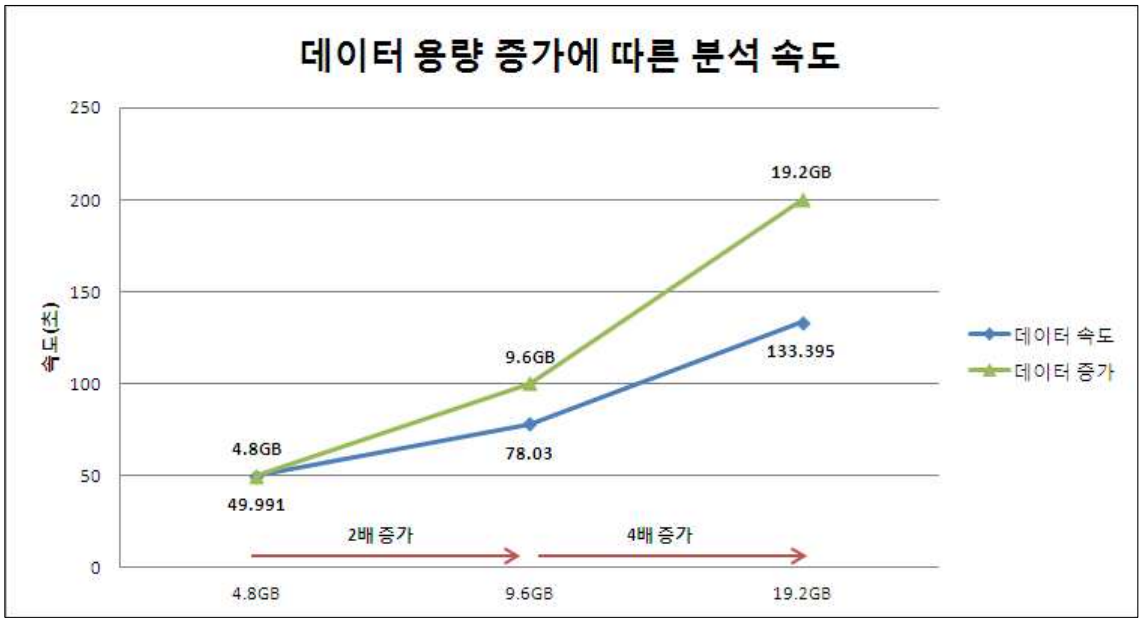


[그림 IV-1. 데이터의 양 증가에 따른 데이터 행 수 및 열 값의 평균 분석]

(2배) 데이터의 양이 2배 증가 시 데이터의 속도는 1.349 증가

(4배) 데이터의 양이 4배 증가 시 데이터의 속도는 2.372 증가

- 데이터의 3번째 열의 A 값과 4번째 열의 B값과 일치하는 값을 조회



[그림 IV-2. 데이터의 양 증가에 따른 특정 값 조회]

(2배) 데이터의 양이 2배 증가 시 데이터의 속도는 1.560 증가

(4배) 데이터의 양이 4배 증가 시 데이터의 속도는 2.668 증가

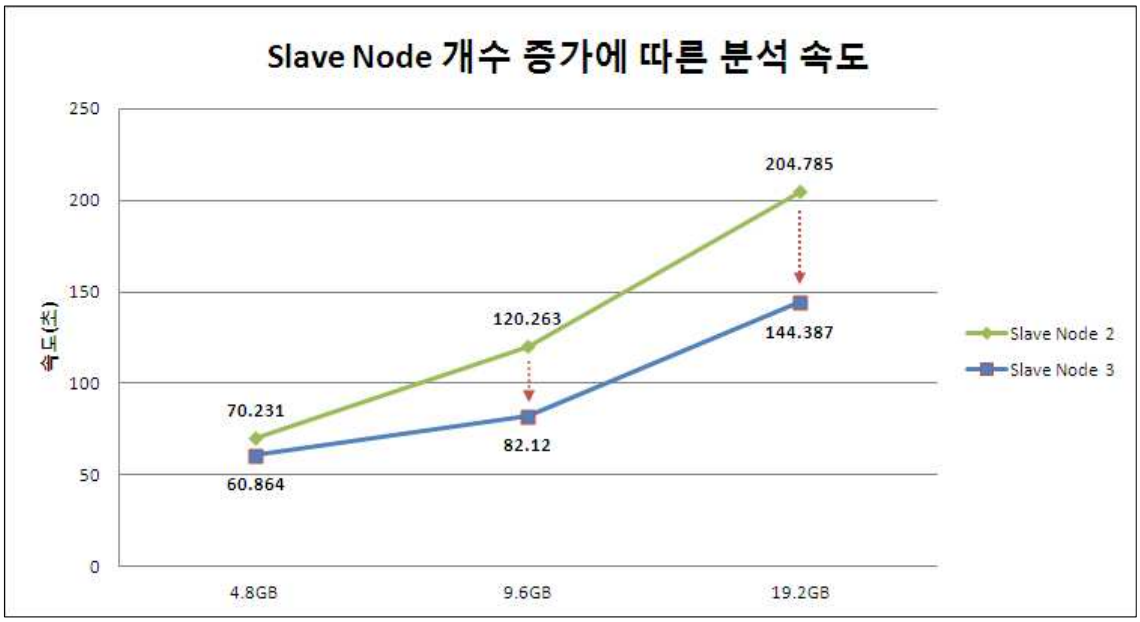
o Slave Node 개수에 따른 Hive의 데이터 분석 속도 측정

Slave Node의 개수가 2개일 때보다 3개일 때 데이터의 분석 속도가 증가한다. 또한 분석 되는 데이터의 양이 증가할수록 Slave Node 개수에 따른 데이터의 분석 속도가 더욱 증가하는 것을 알 수 있다.

- 수행정보

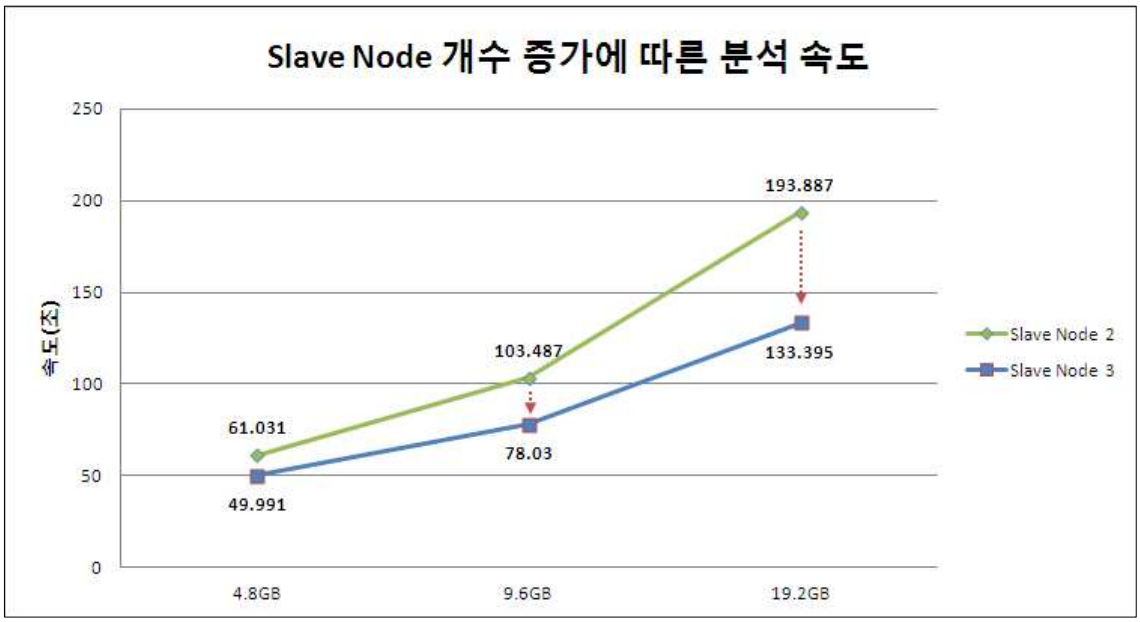
수행 조건	- 19.2GB의 114038176행, 16열의 Data 준비 - Hive에서 table 생성 후 데이터 load - Master Node 와 Slave Node 1, 2 연동 후 분석 요청 - Master Node 와 Slave Node 1, 2, 3 연동 후 분석 요청

- 전체 데이터의 행 수와 3번째 열 값의 평균을 분석 요청



[그림 IV-3. Slave Node 증가에 따른 데이터 행 수 및 열 값의 평균 분석]

- 데이터의 3번째 열의 A 값과 4번째 열의 B값과 일치하는 값을 조회



[그림 IV-4. Slave Node 증가에 따른 특정 값 조회]

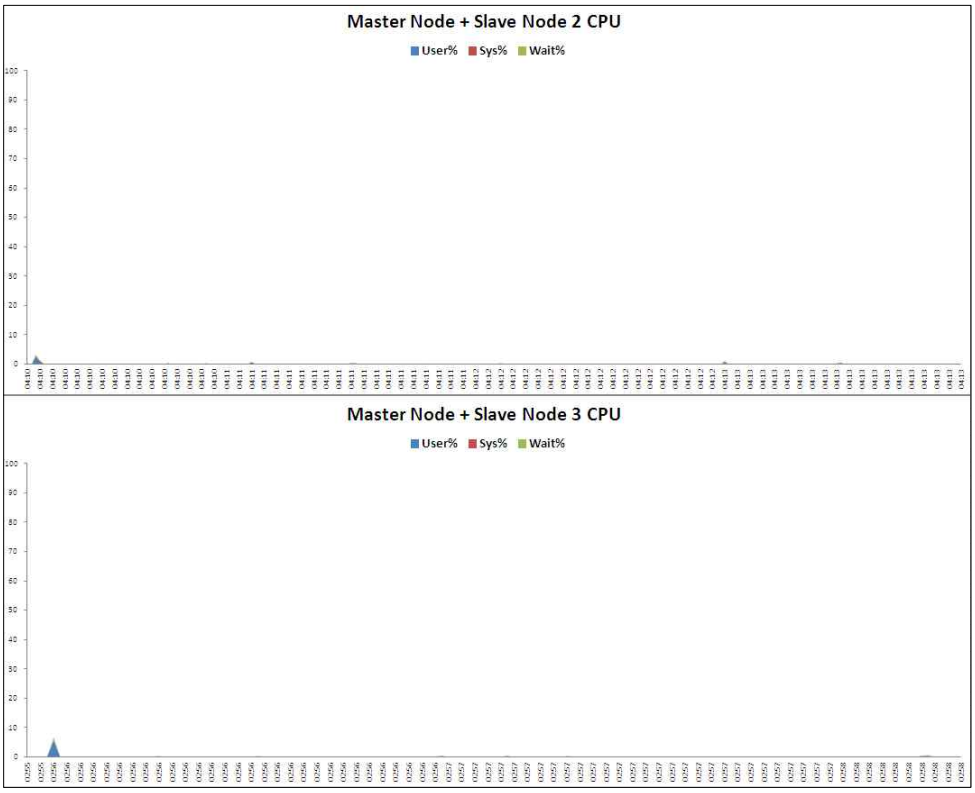
o 데이터 분석 시 Slave Node의 개수에 따른 Server의 자원 사용률 측정

- 수행정보

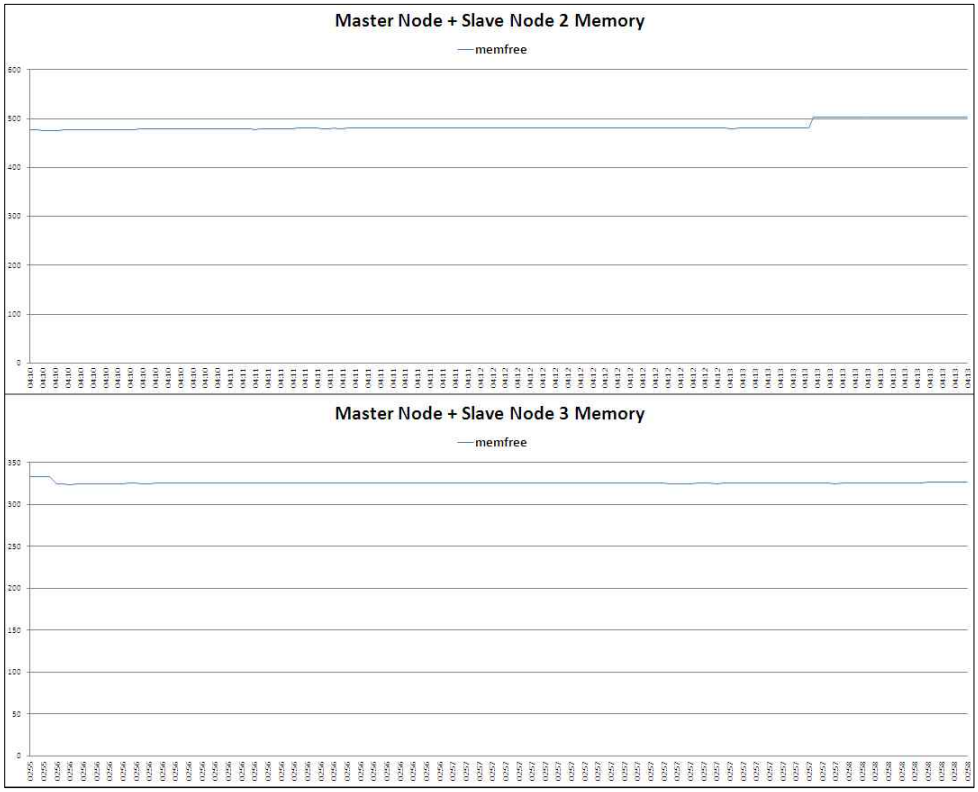
수행 조건	- 19.2GB의 114038176행, 16열의 Data 준비 - Hive에서 table 생성 후 데이터 load - Master Node 와 Slave Node 1, 2 연동 후 분석 요청 - Master Node 와 Slave Node 1, 2, 3 연동 후 분석 요청

- Master Node 측정 결과

서버 자원 사용률 측정 결과 Master Node 에서는 CPU 및 Memory 자원을 거의 사용하지 않는 것으로 나타남

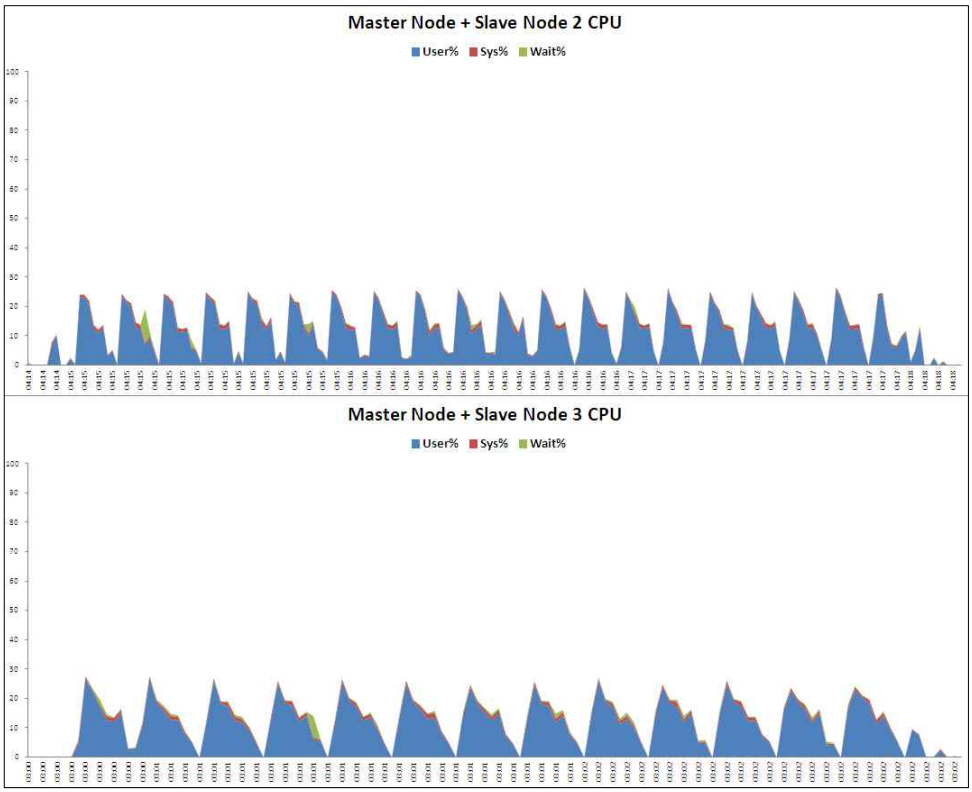


[그림 IV-5. CPU 자원사용률]

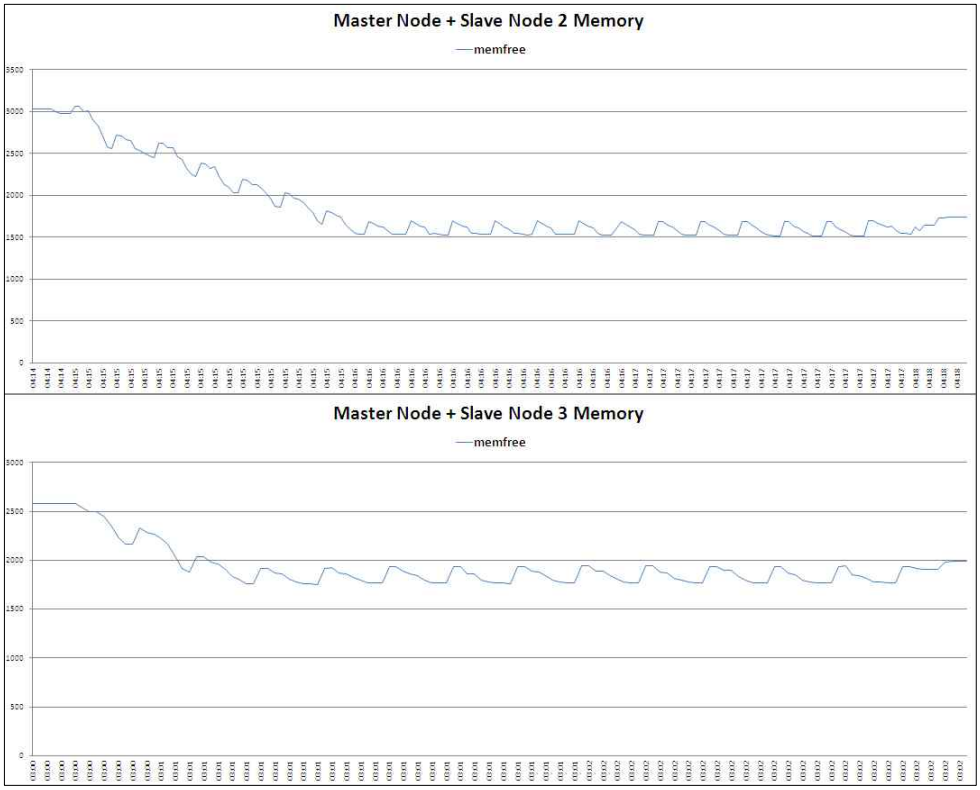


[그림 IV-6. Memory 자원사용률]

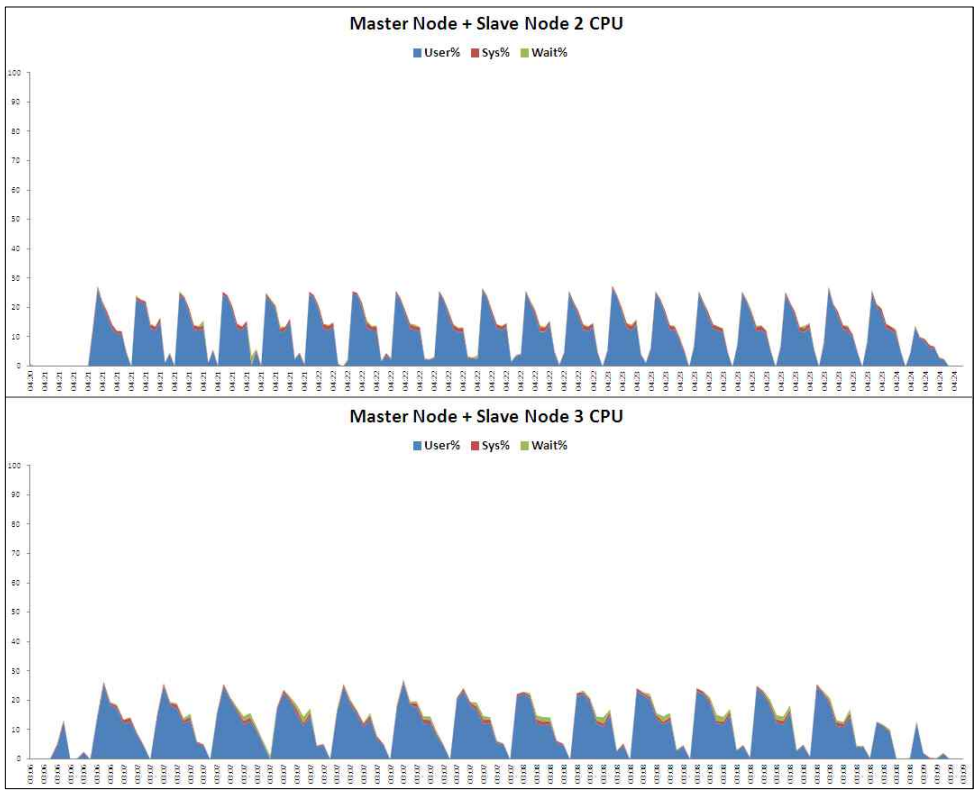
- Slave Node 측정 결과



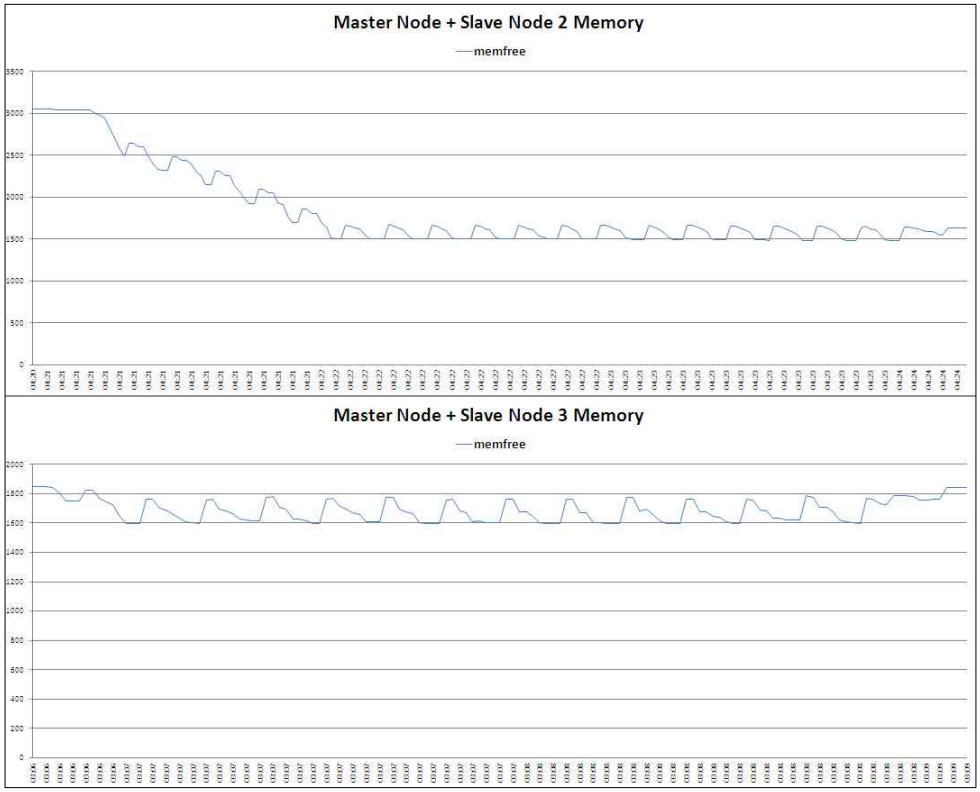
[그림 IV-7. Slave Node 1 CPU 자원사용률]



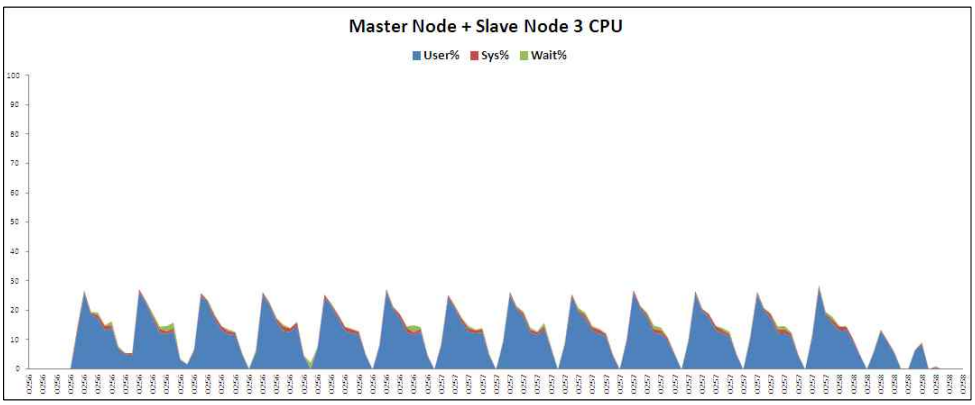
[그림 IV-8. Slave Node 1 Memory 자원사용률]



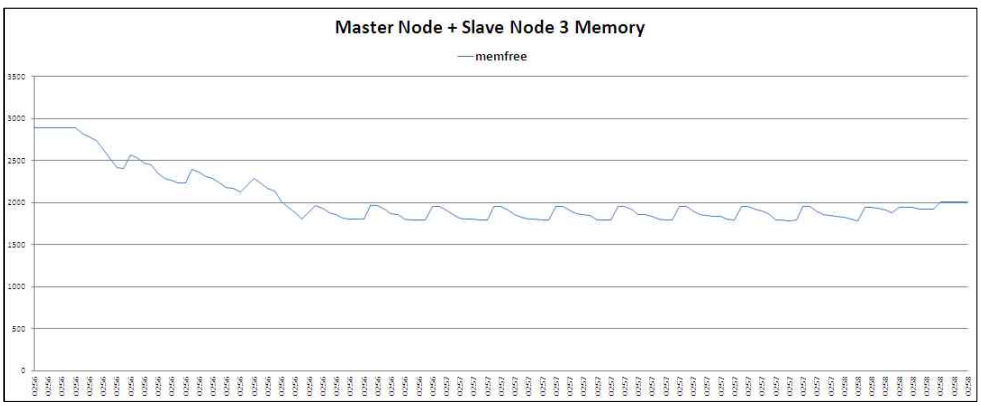
[그림 IV-9. Slave Node 2 CPU 자원사용률]



[그림 IV-10. Slave Node 2 Memory 자원사용률]



[그림 IV-11. Slave Node 3 CPU 자원사용률]



[그림 IV-12. Slave Node 3 Memory 자원사용률]

Ⅴ. 종합

□ Hive 기능 테스트 수행 결과 공개SW로 구성된 Stack 상에서 각 기능 시나리오 수행 시 치명적 오류 또는 심각한 장애가 발생하지 않았으며, Stack을 구성하는 각 공개SW가 유기적으로 동작함을 확인하였음

□ Hive 성능 테스트 수행 결과 데이터의 양 또는 Slave Node의 개수가 증가할수록 분석 속도가 증가 하는 것을 알 수 있다. 또한 데이터 분석 시 Server의 자원사용률을 측정한 결과 Master Node의 성능은 거의 사용하지 않으나, Slave Node의 Memory 사용률에 대해서는 Slave Node의 개수에 따라 차이가 발생하였다. 이에 Hive 도입 시 사용 중인 시스템 환경에서 발생하는 데이터의 양과 Slave Node 개수에 따른 데이터 분석 속도와 Slave Node 개수에 따른 Memory 사용률을 분석해야 할 것으로 판단됨

※ 참고 자료

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

[2] http://hive.apache.org/

[3] http://www.java.com/

[4] http://www.centos.org/

[5] http://en.wikipedia.org/wiki/

[6] http://ko.wikipedia.org/wiki/