TITLE hadoop DATE 2019.07.05 (금)

빅데이터

정확히 Big data 가 아닌 Big Value from data (분석)

① 빅데이터 시대가 도래하면서 누구나 빅데이터를 분석할 수 있어야하는 능력이 요구되었다
② 누구나 분석하기 위해서는 플랫폼이 필요했고, 사용하기 쉬워야 했다
③ 그렇게 하둡이라는 플랫폼이 탄생했다

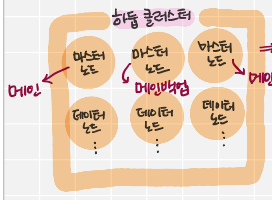
하둡 에코시스템
빅데이터 기술 + 분석도구(R)
빅데이터 플랫폼 → 하둡

데이터 처리과정
데이터 수집 → 저장 → 분석 → 시각화 → 서비스

빅데이터의 활용 ① 예측 ② 추천
ex) 고객(검색), 아이템(쇼핑몰), 영화(추천)
추천시스템 과정
① 예측
② 분류
③ 클러스터링 (공통분모)
④ 연관규칙 (연관성)
⑤ 협업필터링

하둡 에코시스템

2005년 즈음부터 하둡은 기술적인 한계에 봉착한다 (열 발생 ↑↑)
한 대의 컴퓨터를 돌리니 열발생 & 과부하때문에 Cluster방식 (N대의 컴퓨터연결) 탄생
슈퍼컴퓨터 ① 1000노드 이상 ② 속도 Top500 (매년 발표)
하둡 클러스터 : 분산파일시스템 + 분산병렬처리시스템



⇒ 마스터노드는 1개만 존재해야 한다. 하지만 데이터 중요할 경우를 대비해 백업이 있으면 좋다. 백업 마스터 노드가 1개이면 데이터노드와 자리싸움을 하기때문에 마스터노드는 2개가 가장 이상적이다

2008년에 하둡이 슈퍼컴퓨터보다 빨랐다. 하둡은 다른 컴퓨터의 일은 신경쓰지않고 각자 열심히하는 (클러스터) 2개의 컴퓨터만 각자 잘 돌아가게 해줬다
슈퍼컴퓨터는 모든 처리과정이 잘 진행되는지 끊임없이 서로 확인하기때문에 완성과 응답을 계속해서 속도가 하둡에 비해 느림
="남 신경쓰지말고 네 일만 잘하라" → 하둡이 빠른속력이 되는 유일한 이유 → 분산처리

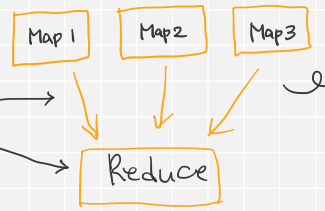
하둡 분산파일시스템 HDFS

분산시스템의 유일한 단점은 어디에 뭐가 저장되었는지 모른다는 것이다 ㅇㅇ!
저장하려는 파일이 128mb 이상이면 128mb씩 자른다
128mb 이하이면 원본 그대로 유지한다

네임노드가 데이터노드의 상태를 보고 어디다 저장할 것인지 결정한다

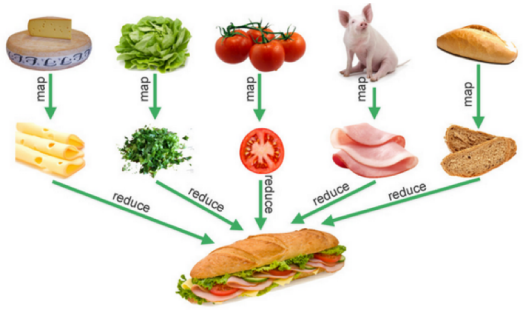
Map Reduce

맵리듀스
① Split
② Map()
③ 셔플
④ Reduce()
⑤ Output



데이터가 다 가는게 아니라 결과값만 보내는게 더욱 효과적이므로 셔플을 이용한다

Map 조계
+
Reduce 정계



맵리듀스의 원리

	정렬시간	Read/Write 시간
백만건	1분	0.1분
천만건	100분	1분
억건	10,000분	10분

Map 셔플 Reduce

100분 10분 10분 → 120분

* 셔플을 하게되면 꼭 오버헤드(20%~30%)가 발생한다

Shuffle (모양별로 수집) Reduce (모양별 Merge Sort)

ex)

Map개수	Reduce개수	Map시간	Shuffle시간	Reduce 시간
10	1	100분	10분	10분
10	4	100/4분	10/4분	10/4분

컴퓨터 10대로 1000배의 속도를 낼 수 있다