LESSON 1

Wordcount TopN(후처리)

Wordcount TopN(후처리)

문제점

- 단순 단어별 빈도수만 나열
  - 특정한 단어가 몇 번 나왔는지 파악하는 데 문제없음

- 어떤 단어가 가장 많이 나왔는지, 많이 나온 단어를 파악하려면?
  - SQL에서의 order by, top/limit/rownum과 같은 작업
  - R과 같은 통계 툴에서 처리
  - 워드카운트의 결과가 아주 많다면?
    - 하둡 자체에서 처리해야 함

Wordcount TopN(후처리)

해결책

- 데이터의 정렬은 복잡하면서 시스템에 부하를 줌
  - O(NlogN) ~ O(N^2)
- 제일 많이/적게 나온 단어를 찾는 것은 O(N)
- 자바의 PriorityQueue를 사용해
  - 가장 많이 나오는 빈도의 단어 개수를 파악
  - O(N)에 근사
- 워드카운트의 결과 값을 받아 많이 나오는 빈도의 단어들의 리스트를 구하는 프로그램

Wordcount TopN(후처리)

프로그래밍 수행방식

- 두 개의 하둡 잡으로 분리
  - 워드카운트를 수행하고 결과를 TopN으로 수행
  - WordCount -> TopN
  - 워드카운트와 TopN을 하나의 프로그램으로 통합
  - WordCount + TopN
  - Oozie(워크플로우)사용
    - 스크립트 작성
    - WordCount -> TopN

LESSON 2

빅데이터와 통계(R)처리
결합방식

빅데이터와 통계(R)처리 결합방식

- 하둡의 결과값(주로 csv)을
  - CSV로 변환해 R의 입력으로
- R기능을 통합한 하둡 커스터마이즈 버전을 사용
  - R-Hadoop / R-Hive / MapR
  - 통일된 방식은 존재하지 않음
  - 각각 처리한 후 이 결과를 다음 툴에 넘겨주는 방식이 대세