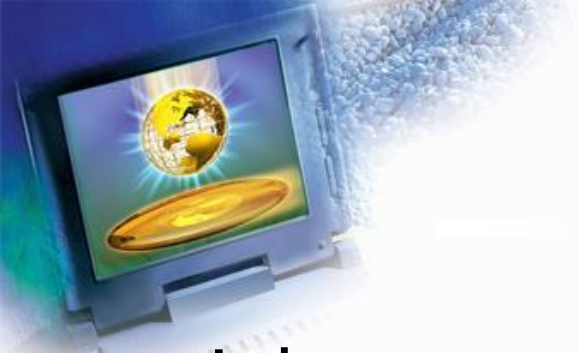
하둡 맵리듀스

워드 카운트 처리

- /home/vagrant/hadoop/share/hadoop/mapreduce
  - hadoop-mapreduce-examples-2.6.1.jar 을 통해 예제인 워드 카운트를 처리할 수 있다.
    - 실행권한 설정
      - chmod 744 hadoop-mapreduce-examples-2.6.1.jar
- 입력 : 테스터할 hdfs에서 folder를 생성과 파일 copy
  - hdfs dfs -put /etc/hadoop/hadoop-env.sh 입력할폴드경로
  - ex)
  - [vagrant@master ~]$ hdfs dfs -put hadoop/etc/hadoop/hadoop-env.sh /user/vagrant/conf
- 출력 : 해당 입력 파일로 결과 folder 생성..
- 명령 : wordcount 자바 파일 실행
  - yarn jar hadoop-mapreduce-examples-2.6.1.jar wordcount 입력폴드 출력폴드

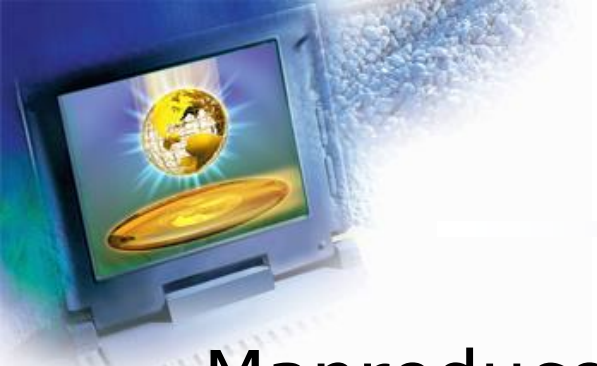
파일내용 확인

- hdfs dfs -cat /user/vagrant/output/part-r-00000 | tail -5



MapReduce

- job
  - 데이터를 처리하는 mapper와 reducer 들로 이루어진 프로그램 단위
  - ex) 20개의 파일들을 입력으로 word count 실행
- task
  - mapper나 reducer의 각각의 실행 인스턴스
  - ex) 20개의 입력 파일들을 20개의 map task들과 다수의 reducer task들을 의미
- task attempt
  - 한 머신에서 실행되는 특정 task
  - ex) 최소 20개의 task attempt

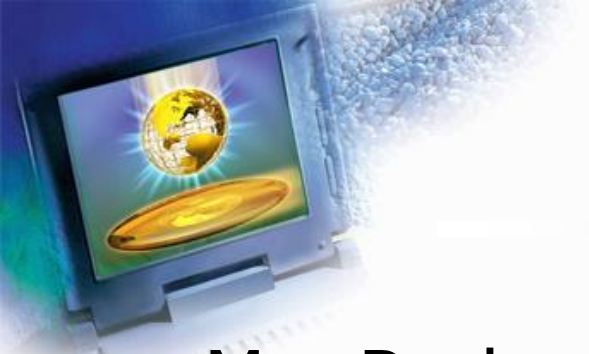


job 데이터 분배

- Mapreduce job이 실행될 때 실행하는 jar과 옵션정보 xml파일이 hdfs에 저장되고, 이 파일의 다운로드 위치 정보를 task tracker들에게 알려준다.
- task tacker 내부적으로 이 binary들을 다운 받아서 실행할 수 있는 환경을 구성

입력 데이터 분산

- 모든 mapper 들은 기본적으로 동일한데,
- 처리해야 할 데이터 블록이 있는 곳에서 mapper를 실행하면 네트워크 트래픽을 줄일 수 있다.
- reducer는 필연적으로 네트워크 트래픽이 발생하면 hdfs가 처리.



Configuration 클래스

- MapReduce 프로그램은 많은 설정 옵션을 가짐
- Configuration 객체는 (키, 값)쌍의 현태로 설정 옵션을 관리
  - "mapred.map.tasks" ==> 20
  - Configuration 객체는 클러스터내의 모든 테스크들에게 직렬화 되어 배포 됨

Mapper 생성

- Mapper 클래스는 사용자가 작성
  - MapReduceBase 인터페이스를 상속
- Task단위로 사용자의 Mapper 인스턴스가 초기화 됨
  - mapper의 모든 인스턴스들은 서로 다른 프로세스 내에 존재함. 데이터 공유가 없음..
- 코드

```
void map(WritableComparable key, Writable value, Context context)
```

Writable 클래스

- 문자(Text), 정수(IntWritable, LongWritable), 실수(FloatWritable) 등의 데이터를 표현 하는 컨테이너 클리스
- 모든 key
  - Writable의 인스턴스
- 모든 value
  - WritableComparable의 인스턴스
- 더 좋은 cache 사용을 위해서 메모리 주소를 재사용하는 공유 컨테이너 모델

데이터 읽기

- 데이터 집합들을 InputFormat 클래스에 의해 지정됨
  - 디렉토리등 입력 데이터 지정
  - 데이터를 분할하여 map의 갯수만큼 inputSplit을 만들어 냄
  - 입력 소스로부터(키, 값) 쌍으로 데이터를 읽을 RecordReader 객체를 생성

FileInputFormat 상속 클래스

- TextInputFormat은 파일 내의 '\n'로 끝나는 라인들을 값(value)으로 처리
- KeyValueTextInputFormat은 파일내의 '\n'로 끝나는 라인들을 (키 구분자 값)의 형식으로 처리
- SequenceFileInputFormat은 (키,값) 쌍으로 구성된 바이너리 포맷 파일
- SequenceFileAsTextInputFormat은 위에서 텍스트라는 것만 차이.

Input Split 크기

- FileInputFormat은 큰 파일들을 여러 개의 chunk로 나눔
  - 옵션, 블록사이즈
    - "mapred.min.split.size", "mapred.max.split.size"
- RecordReader
  - 파일명, 시작주소(offset), 읽을량(chunk크기) 인자로 받음
  - chunk 크기는 함수를 override 해서 바꿀 수 있음.

Partitioner 클래스

- int getPartition(key, val,numPartitions)
  - 주어진 키에 대한 파티션 번호
  - 하나의 파티션번호는 하나의 Reduce 태스크와 Mapping
- HashPartitioner 객체가 기본
  - 키의 해쉬값을 Reducer의 개수로 modular 연산해서 얻은 값을 반환
- Job 객체에서 Partitioner 클래스를 지정할 수 있다.

Thank You !

www.themegallery.com