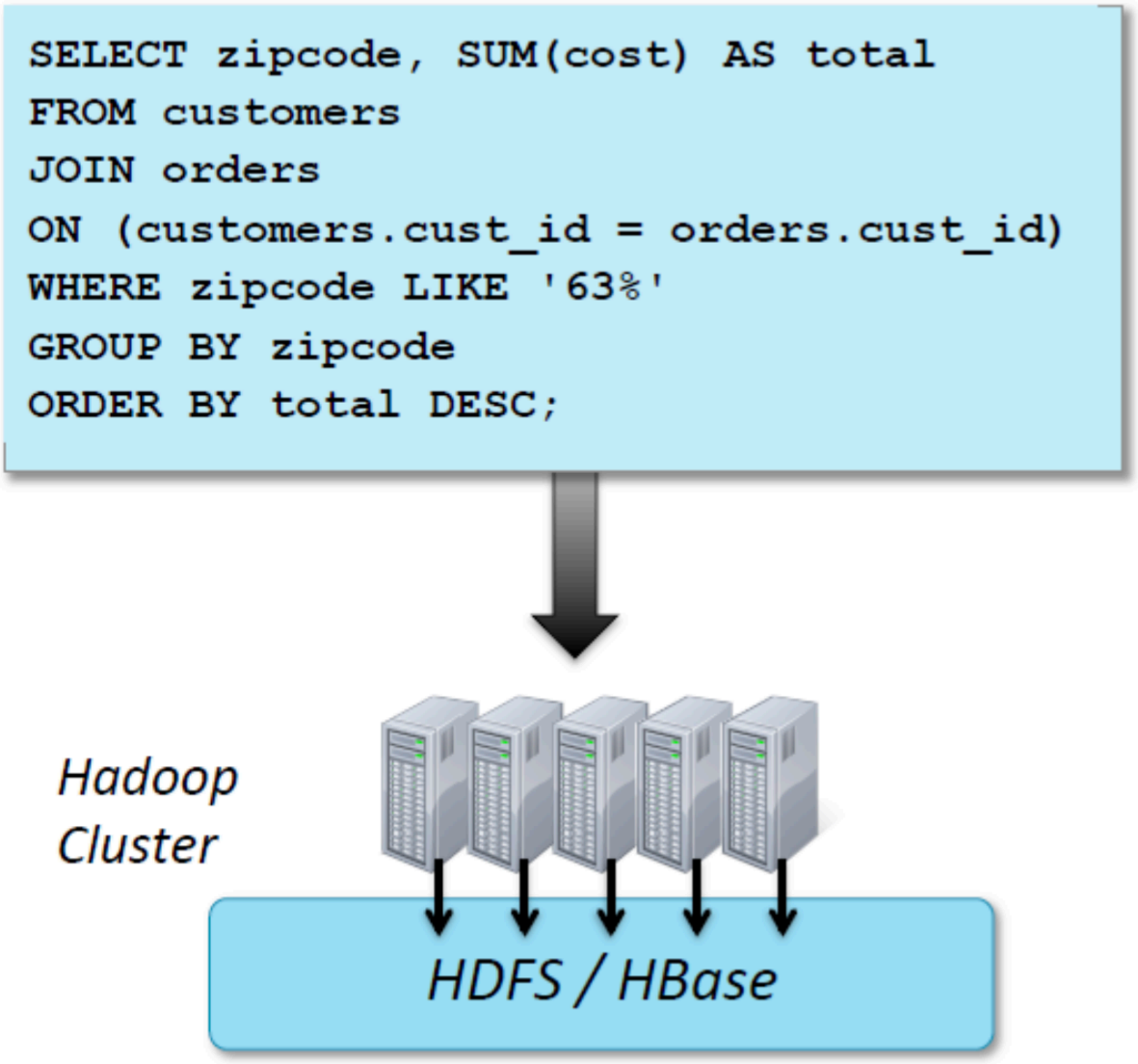
impala와 하이브

- 생긴 이유 : 스쿱에서 파일 시스템에서는 데이터 찾기 (데이터 접근 어려움)
- 사용자들이 데이터 접근을 쉽게하기 위해서 SQL에 대응되는 새로운 layer를 얹은 것
- 내부적으로 변환작업을 거쳐서 파일 시스템에 대한 파일에 대한 접근을 SQL을 통해서 파일을 구조화된 정보로 보이게 해주고 찾게도 해줌



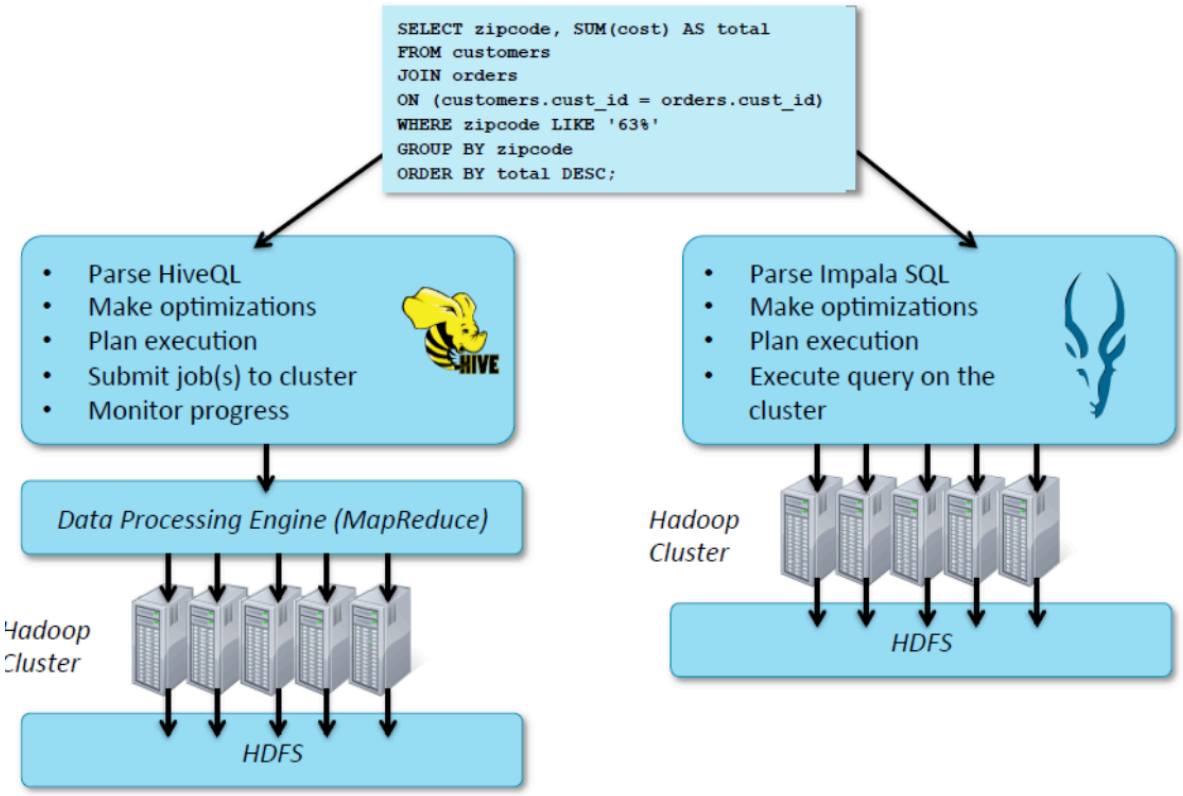
Apache Hive

- High level abstraction on top of Mapreduce
  - 스파크에서 맵리듀스 형태로 변환
  - 맵리듀스를 사용하는 것에 있어서 강점
  - 다양한 기능을 제공함
  - HiveQL 사용
  - Facebook에서 개발 이후 Open source가 됨

Cloudera Impala

- High performance dedicated SQL engine
    - HDFS에 직접 접근
    - Impala가 hive보다 빠르다. 성능 효율적
    - uses Impala SQL

진행과정



- Hive는 각 노드들이 맵리듀스작업을 효율적으로 분배되게 처리함
- Impala는 hdfs 수준에서 파일 시스템을 직접 접근할때의 최적화

왜 사용하는가?

- 빅데이터를 다루기 위해서 하둡 파일시스템으로 옮겼을때 기존 SQL을 그대로 사용할 수 있다.
- MapReduce로 프로그래밍을 해도 200줄 이상 넘어가는데 (적어도) 하이브를 사용하면 SQL문 5줄 정도면 된다.
    - 찾고자하는 대상을 선언만 (what) - declaritive language
    - MapReduce 프로그래밍에서는 일일이 해줘야된다.

특수한 경우마다 최적화된 결과를 뽑아낼 수는 없지만 보편적인 rule을 가지고 변환해주는데 micro 하게 내어플리케이션 적합한 MapReduce까지 변환은 안된다고 본다. 추가적인 Optimization을 위해서는 MapReduce단 직접 작업 필요.

- 사용자가 SQL문을 쓰는 게 편리한 것처럼, 다른 시스템과의 프로토콜을 정의하기에 굉장히 쉬움 - interoperability with other system (business intelligence tools(BI))
- use case

- log data를 table로 확인할 수 있음
- Sentiment Analytics (긍정이냐 부정이냐 를 파악하는 것들)
- BI : 시각화, 시간을 기준으로 grouping, 내가 보고싶어하는 기준을 대입해서 여러가지 분석을 할 수 있음
    - BI를 구현하는데 하이브 임팔라 차이점
        - 하이브는 기능이 많은 반면에 실시간으로 이뤄지는 분석은 어울리지 않음 (성능상의 오버헤드가 많을 수 있음)
        - 실시간데이터를 다룰 때는 임팔라를 사용하는게 더 효율적임

Impala 사용법

Impala-shell : localhost 서버로 접속

Impala-shell -i server host : Impala 서버가 깔린 다른 서버에 접속

shell : 리눅스 멸영어 사용할 수 있게 해줌

SQL이 길경우 파일로 보관을 하는데 -f 옵션을 주면 sql 파일을 sql로 줄수 있음

hive로 import를 해도, impala로 import를 해도 데이터를 공유함

Clent-server database management systems

- very fast response time
- support transaction
- allow modification of existing records
- can serve thousands of simultaneous clients

Hadoop is not an RDBMS

- Mapreduce HIveQL -> limition of HDFS and MapReduce still apply
- impala is faster but not intended for OLTP db
- no transaction support (update, delete 지원안함)
    - 내부적으로 data consistency를 보장하지 않음 (분산환경에서 하기 어렵다. 여러노드에 복제되기때문에 data consistency를 보장하기 어렵다.)

	Relational Database	Hive	Impala
Query language	SQL (full)	SQL (subset)	SQL (subset)
Update individual records	Yes	No	No
Delete individual records	Yes	No	No
Transactions	Yes	No	No
Index support	Extensive	Limited	No
Latency	Very low	High	Low
Data size	Terabytes	Petabytes	Petabytes

위의 단점에도 불구하고 마지막 한 로우로인해서 빅데이터에 하이브와 임팔라가 쓰임.

strong consistency : weak consistency