

Apache Tajo 프로젝트 소개 및 최신 기술동향



손지훈

발표자 소개

- 손지훈 (Jihoon Son)
 - 박사 과정 수료 (Computer Science & Engineering, 2010.3 ~)
 - Apache Tajo PMC and Committer (2014.5.1 ~)
 - Mentor of Google Summer of Code (2013)
- 연락처
 - Email: jihoonson AT apache.org
 - LinkedIn: <https://www.linkedin.com/in/jihoonson>
 - Twitter: @jihoonson

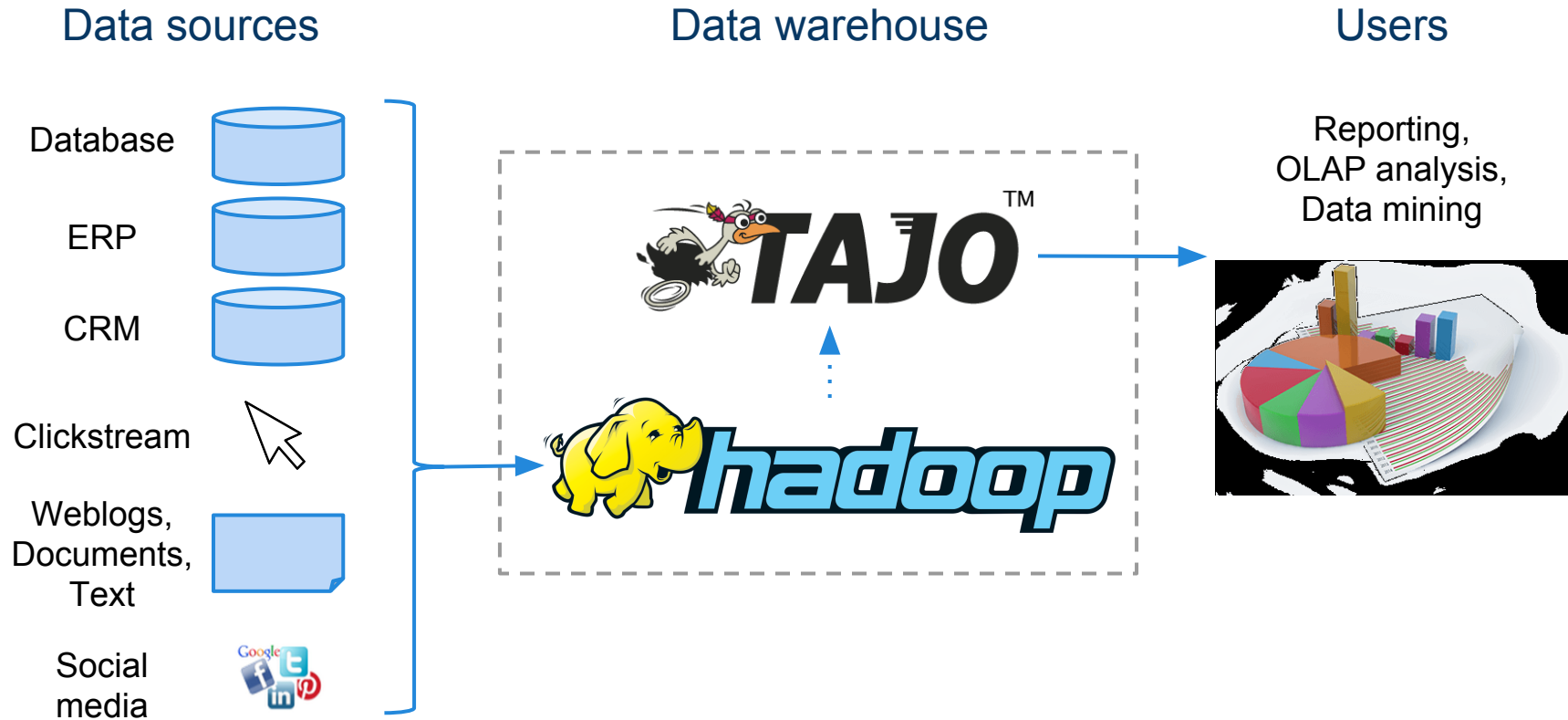
Contents

- Tajo란?
- Tajo의 특징
- Tajo 응용의 예
- 적용 사례
- 앞으로의 발전 방향

Tajo란?

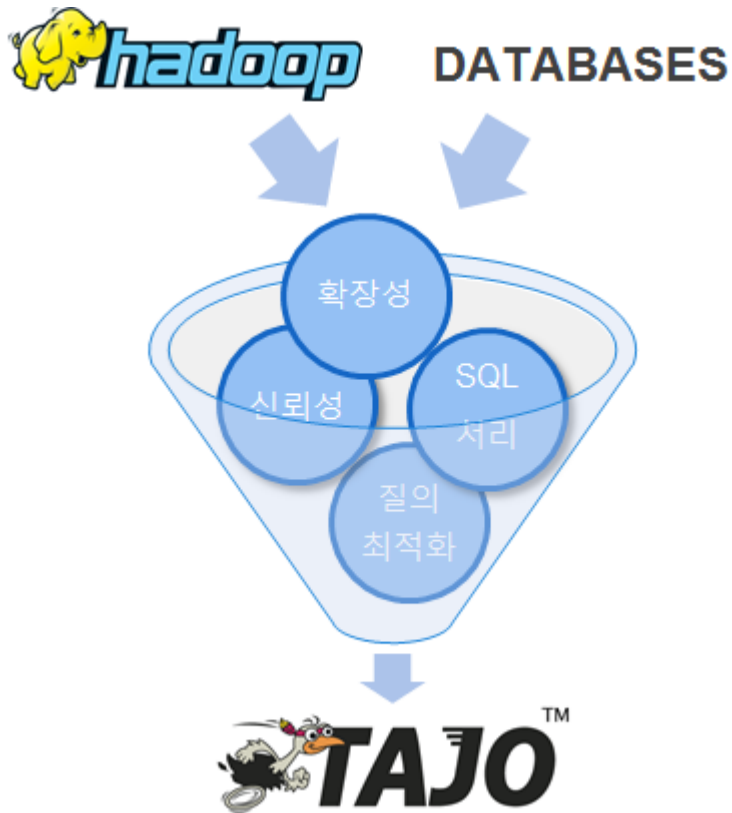
- 빅데이터 분석을 위한 데이터웨어하우스 시스템
- SQL-on-Hadoop
 - 지속적으로 Hadoop에 저장되는 데이터를 관리
 - Hadoop에 저장된 대용량의 데이터에 대해 효과적으로 SQL 질의를 처리

Tajo란?



Tajo의 특징

- 하둡과 데이터베이스의 기술적 융합
 - Hadoop의 분산 처리 기술
 - 확장성, 신뢰성
 - 데이터베이스의 SQL 질의 처리 기술
 - SQL 질의 처리 엔진, 질의 최적화



Tajo의 특징

- 풍부한 SQL 기능 지원
- 효율적인 SQL 질의 처리
- 신뢰성 있는 질의 처리
- 그 외
 - 확장성, 이종성

Tajo의 특징

- 풍부한 SQL 기능 지원
 - 표준 호환 SQL 지원
 - 다양한 Join 지원
 - INNER, CROSS, LEFT/RIGHT/FULL OUTER, LEFT/RIGHT SEMI, LEFT/RIGHT ANTI JOINS
 - Window function 지원
 - 명시된 window frame에 대해, 각각의 window frame에 포함되는 데이터에 대한 결과값을 계산

Tajo의 특징

- 풍부한 SQL 기능 지원

window frame

```
SELECT age, avg(salary) as avg_sal  
OVER (ORDER BY age)  
FROM empsalary;
```

age	avg_sal
30	4000
32	4500
...	

- Window function 지원

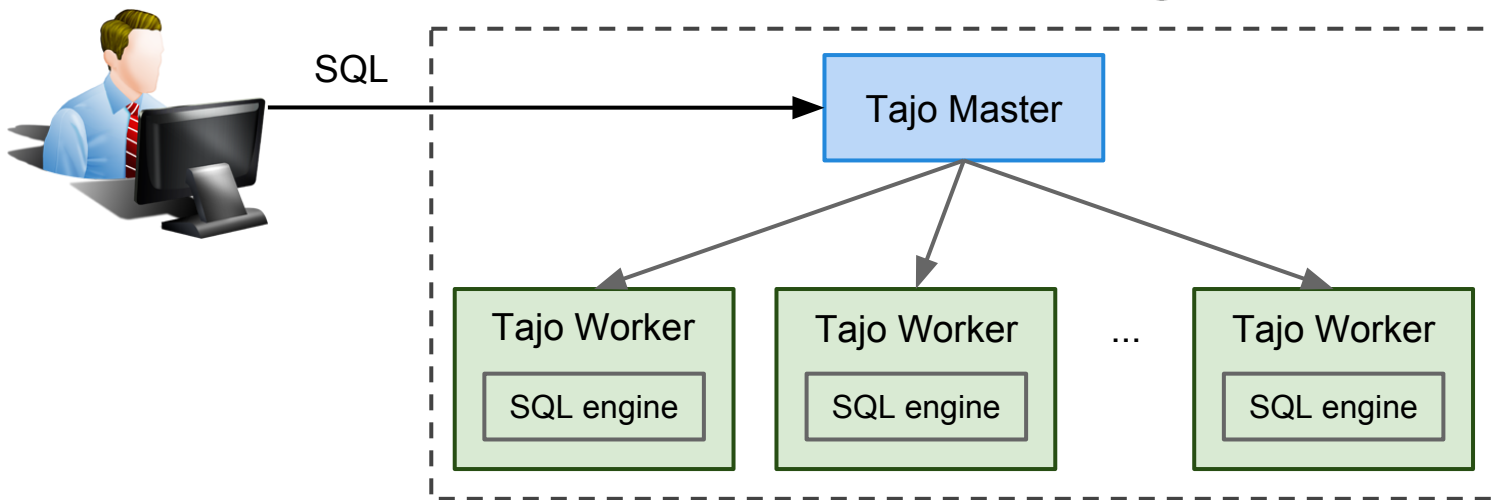
- 명시된 window frame에 대해, 각각의 window frame에 포함되는 데이터에 대한 결과값을 계산

Tajo의 특징

- 효율적인 SQL 질의 처리
 - 고유의 분산 질의 처리 엔진
 - Hadoop의 MapReduce를 사용하지 않음
 - 관계형 질의 처리 엔진
 - SQL 처리에 최적화
 - 대용량의 데이터를 빠르게 처리
 - 대용량의 데이터를 여러 클러스터 노드가 함께 처리

Tajo의 특징

- 효율적인 SQL 질의 처리
 - 고유의 분산 질의 처리 엔진

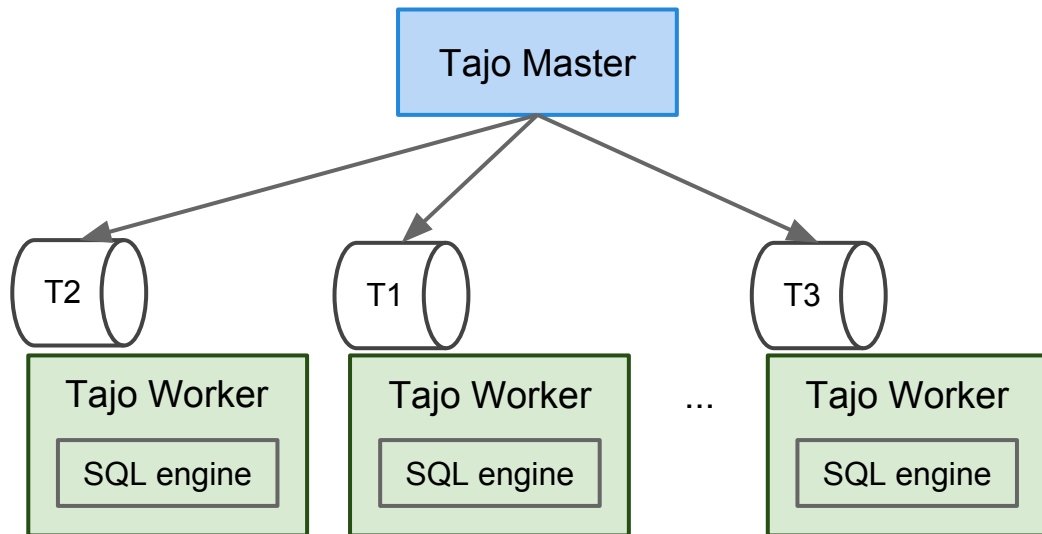


Tajo의 특징

- 효율적인 SQL 질의 처리
 - 발전된 질의 처리 최적화
 - 데이터베이스의 첨단 질의 최적화 기법 적용
 - Join 순서 최적화, In-memory 질의 처리 기법 등
 - 클라우드 환경에 적합한 새로운 질의 처리 최적화 기법 적용
 - 동적 로드 밸런싱 (Dynamic Load Balancing)
 - 질의 수행 중 동적으로 워커들의 로드를 조절
 - 점진적 질의 최적화 기법 (Progressive Query Optimization)
 - 질의를 수행하면서 점진적으로 수행 중인 질의를 최적화
 - 시간이 오래 걸리는 질의 처리에 효율적

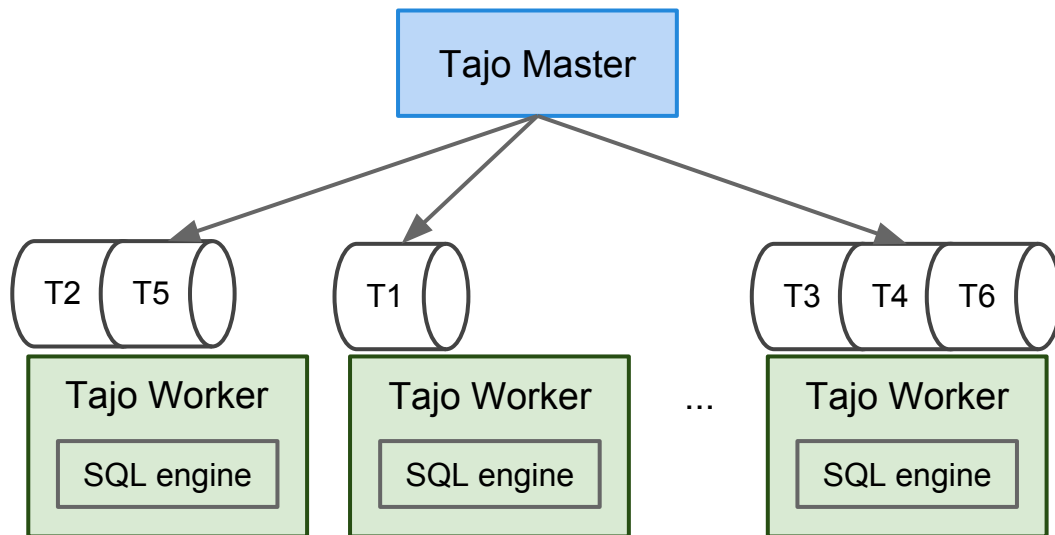
Tajo의 특징

- 효율적인 SQL 질의 처리
 - 동적 로드 밸런싱



Tajo의 특징

- 효율적인 SQL 질의 처리
 - 동적 로드 밸런싱

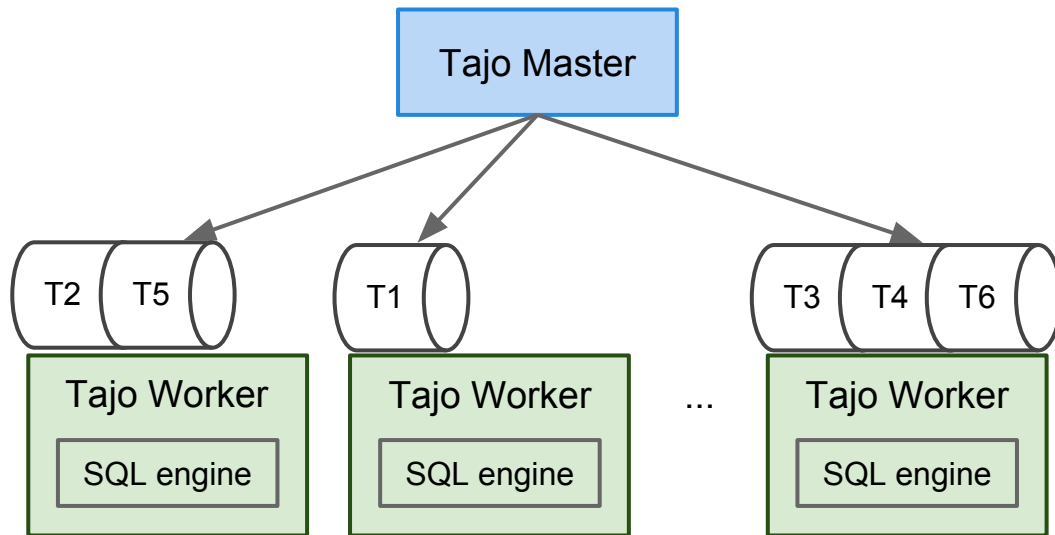


Tajo의 특징

- 신뢰성 있는 질의 처리
 - 장애 허용 질의 처리
 - 질의 처리 중, 일부 클러스터 노드에 장애가 발생해도 질의를 끝까지 마칠 수 있음
 - 높은 가용성 (High availability)
 - Tajo 클러스터를 총괄하는 Master 노드에 장애가 발생하면 자동으로 백업 Master가 역할을 대신함
 - 단일 장애점 (Single Point of Failure) 문제 해결

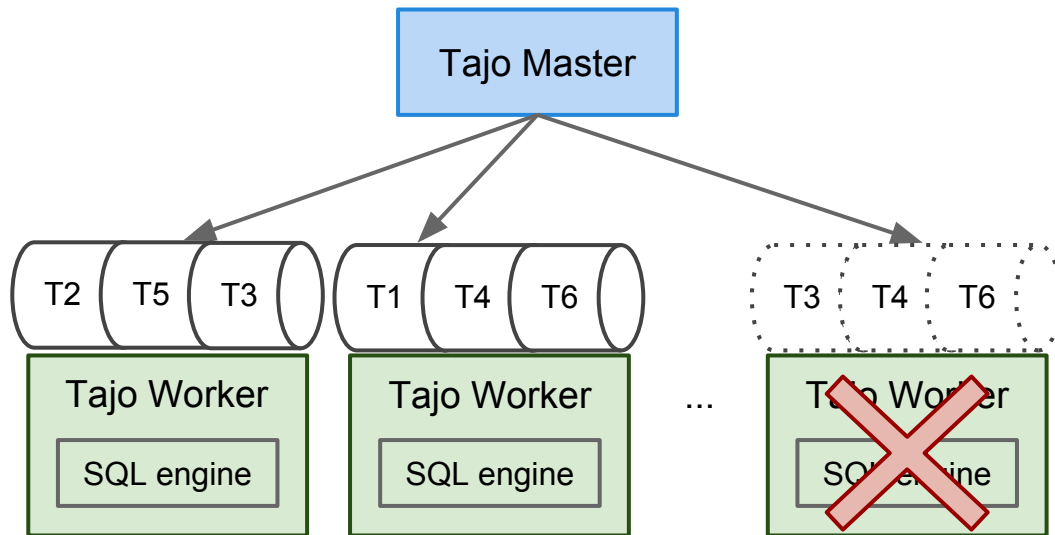
Tajo의 특징

- 신뢰성 있는 질의 처리
 - 장애 허용 질의 처리



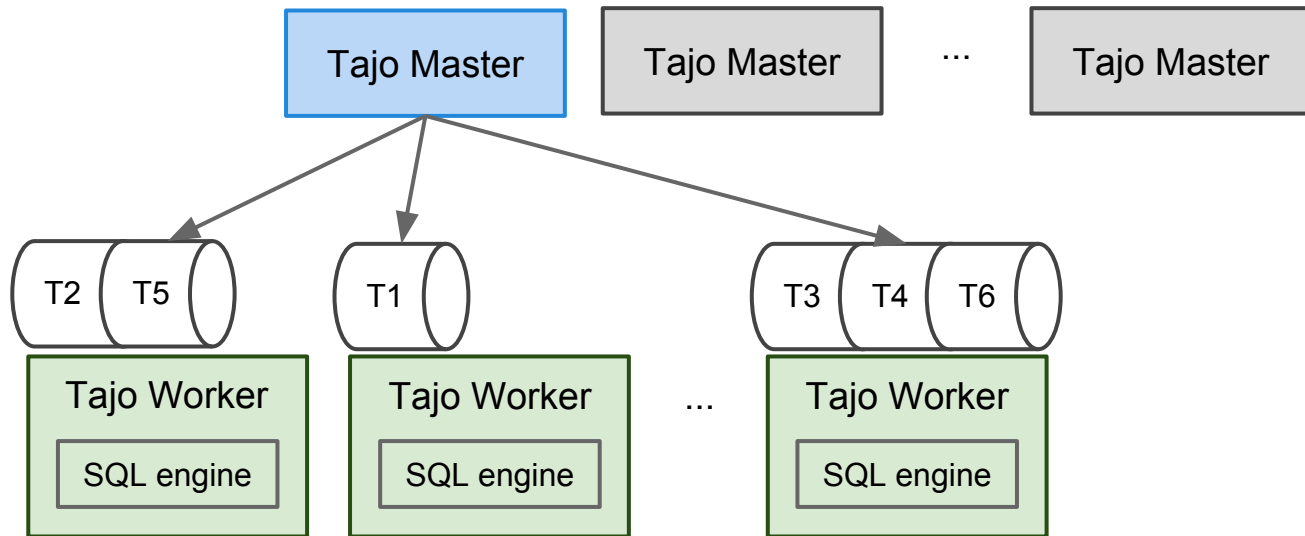
Tajo의 특징

- 신뢰성 있는 질의 처리
 - 장애 허용 질의 처리



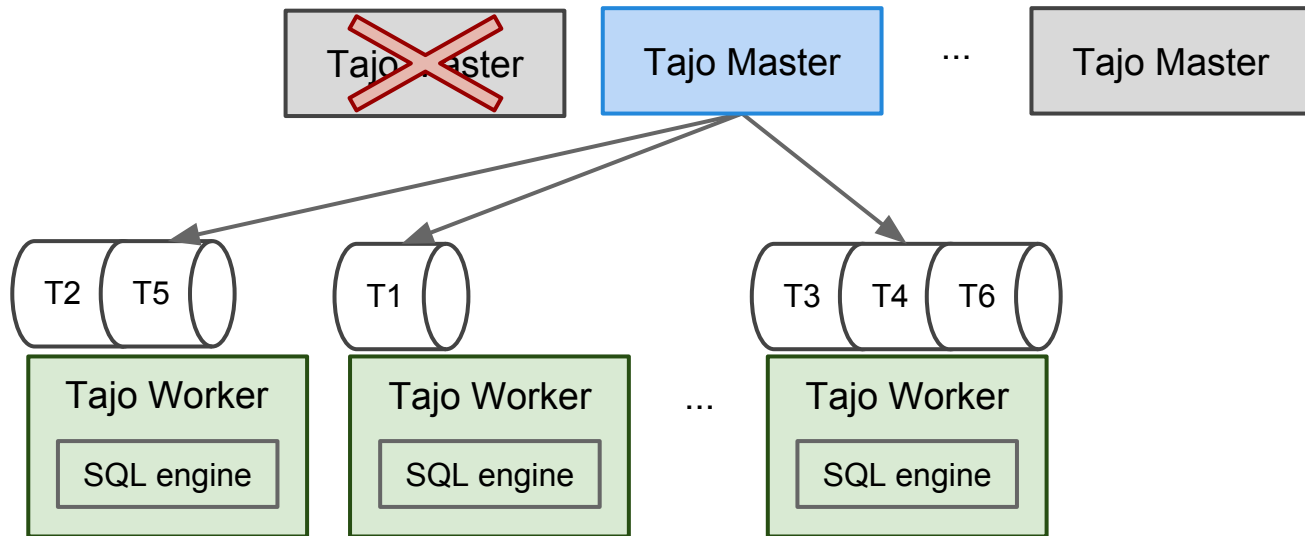
Tajo의 특징

- 신뢰성 있는 질의 처리
 - 높은 가용성 (High availability)



Tajo의 특징

- 신뢰성 있는 질의 처리
 - 높은 가용성 (High availability)



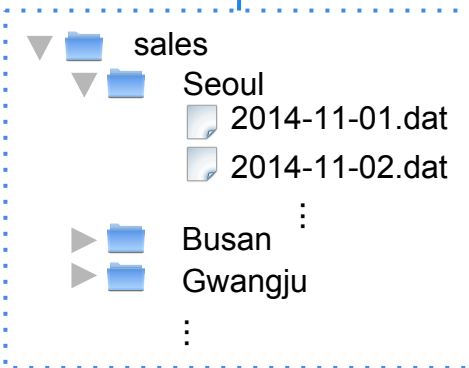
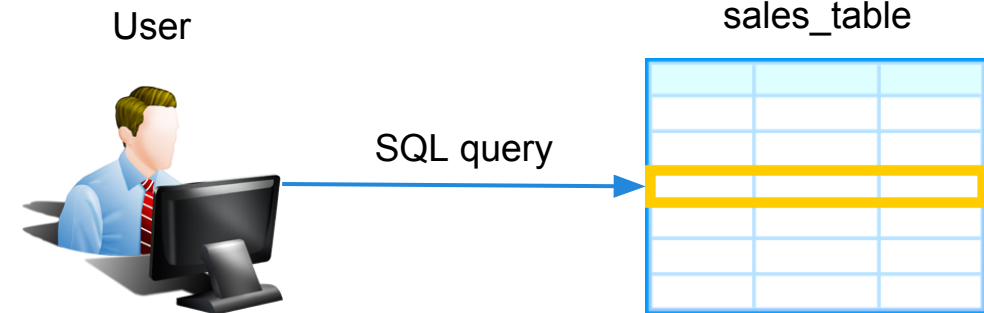
Tajo의 특징

- 확장성
 - 복잡한 과정 없이, 클러스터에 노드를 단순히 추가함으로써 클러스터 확장 가능
- 이종성
 - 서로 다른 컴퓨팅 능력을 가진 노드들로 구성된 클러스터에서도 잘 동작
 - 분산 질의 실행 시, 각 노드들의 컴퓨팅 능력을 고려하여 작업 할당

Tajo 응용의 예

- 다양한 소스로부터 지속적으로 추가되는 데이터 관리에 용이
 - 하나의 Hadoop 디렉토리에 저장된 데이터를 하나의 테이블로 인식
 - **Partitioned table** 지원
 - 테이블을 저장할 때, 일정한 기준에 따라 테이블을 나누어 저장
 - 예) 상품 판매 데이터를 저장할 때, 지점 별로 다른 디렉토리에 나누어 저장 가능
 - 관리하기 용이하며, 질의 성능 향상에 사용될 수 있음
 - 현재 Hive와 호환되는 **column partitioning** 지원

Tajo 응용의 예



Tajo 응용의 예

- 배치 분석
 - 큰 데이터를 한 번에 처리
 - 시간이 오래 걸림
 - 일반적으로 처리할 질의가 미리 정의되어 있음
 - 사용자가 많지 않은 시간에 주로 수행
 - 예) "올해와 작년의 수익 차를 분기 별, 월 별, 지점 별로 보여라"

Tajo 응용의 예

- 대화형 분석
 - 사용자가 시스템과 대화하듯 분석 질의를 수행
 - 일반적으로 시간이 짧게 걸림
 - 사용자가 임의의 질의를 수행
 - 탐험적 분석
 - 예) "지난 한시간 동안 서울 지점에서 가장 많이 판매된 물품 10개를 카테고리 별로 출력하라"

적용 사례

- SK Telecom

- 목적: 통신망 로그 분석
- 클러스터 크기: 50대 이상
- 일일 데이터 처리량: 200 TB 이상
- 관련 자료
 - News: <http://news.imaso.co.kr/123151>
 - Slide: <http://www.slideshare.net/devview/2a3big-data-launching-episodes>

적용 사례

- Gruter
 - 목적: 소셜 네트워크 서비스 분석
 - 클러스터 크기: 약 30대
 - 일일 데이터 처리량: 수백 기가
- LOEN 엔터테인먼트
 - 목적: MelOn 서비스 로그 분석
 - 클러스터 크기: 약 50대

Tajo Releases

- 버전 0.2.0
 - 릴리즈 날짜: 2013.11.20
 - 주요 내용: 더 많은 SQL 기능 지원
- 버전 0.8.0
 - 릴리즈 날짜: 2014.05.01
 - 주요 내용: 안전성 개선

Tajo Releases

- 버전 0.9.0
 - 릴리즈 날짜: 2014.10.21
 - 주요 내용: 더 많은 SQL 기능 지원 및 질의 처리 성능 향상
- 버전 0.9.1
 - 릴리즈 계획: 2014.11
 - 주요 계획
 - 비정형 파일 포맷 (JSON) 지원
 - 네트워크 비용 감소로 인한 질의 처리 성능 향상

앞으로의 발전 방향

- 근사 질의 처리 (Approximate query processing)
 - 빅데이터에서 반드시 정확한 결과 값을 얻을 필요 없음
 - 정확도가 떨어지는 대신, 아주 빠른 시간 안에 결과 값 도출 가능
 - 예) BlinkDB

```
SELECT avg(sessionTime)
FROM Table
WHERE city='San Francisco'
WITHIN 2 SECONDS
```

Queries with Time Bounds

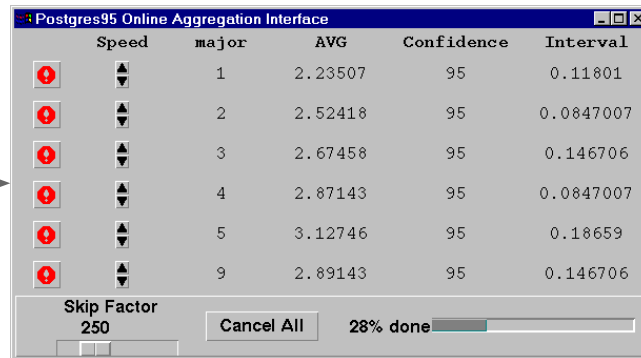
```
SELECT avg(sessionTime)
FROM Table
WHERE city='San Francisco'
ERROR 0.1 CONFIDENCE 95.0%
```

Queries with Error Bounds

앞으로의 발전 방향

- 근사 질의 처리 (Approximate query processing)
 - 빅데이터에서 반드시 정확한 결과 값을 얻을 필요 없음
 - 정확도가 떨어지는 대신, 아주 빠른 시간 안에 결과 값 도출 가능
 - 예) Online aggregation

```
SELECT major, avg(score)
FROM students
GROUP BY major
```



Speed	major	AVG	Confidence	Interval
1	1	2.23507	95	0.11801
2	2	2.52418	95	0.0847007
3	3	2.67458	95	0.146706
4	4	2.87143	95	0.0847007
5	5	3.12746	95	0.18659
9	9	2.89143	95	0.146706

Skip Factor
250

Cancel All 28% done

앞으로의 발전 방향

- 질의 처리 성능 향상
 - 최신 데이터베이스 질의 처리 기법 적용
 - Vectorized query processing, runtime code generation
 - 질의 최적화 기법 향상
 - 리소스를 효율적으로 활용하는 병렬 처리 기법
 - 색인의 효율적인 활용
- 다양한 타입의 저장소 지원
 - HBase, MongoDB, local file system, ...

앞으로의 발전 방향

- Database connectivity 지원
 - ODBC
- Yarn 리소스 스케줄러 지원
 - Hadoop의 Yarn 리소스 스케줄러와 호환
 - Yarn을 이용하여 Tajo의 리소스를 스케줄링
- 사용자 정의 함수 (UDF) 지원
 - CREATE/DROP FUNCTION 구문 지원

Join Us!

<http://tajo.apache.org/>