



다운로드 모듈에 의한 부하 처리 방안



1. 부하의 원인

긴 수행시간으로 인한 요청 누적 및 여러 사용자의 동시다발적인 요청으로 인한 병목현상

오디오 다운로드

아프리카TV 에서 스트리밍된 영상의 오디오를 다운받는 경우 1시간당 약 5분 소요된다. 스트리밍 영상의 길이가 보통 2~8시간이므로 오디오 분석하기 전 다운로드 작업에만 긴 시간이 소요된다.

채팅 다운로드

트위치의 경우 스트리밍된 영상의 채팅을 가운받는데 1시간당 약 1분 소요된다. 채팅을 크롤링하기 위해 사용되는 BeautifulSoup, Requests은 병렬처리 로직을 별도로 작성하지 않으면 느린편이다.

2. 처리 방안

Celery를 이용한 분산처리



C E L E R Y

Celery란?

비동기 작업 큐(Asynchronous task queue) 기반으로 동작하는 파이썬 프레임워크로 요청된 작업을 큐에 담아 비동기적으로 처리한다.

언제 Celery(비동기 작업 큐)를 사용하는가?

웹 서버에 부담을 주는 작업 요청 받았을 때 **비동기 작업 큐**를 이용하면 작업이 실행되는 시점과 생성되는 시점, 또는 생성 순서에 상관없이 웹 서버에 부담을 주는 작업을 순차적으로 처리하여 웹 서버를 가볍게 유지할 수 있다.

웹 서버에 부담을 주는 작업

웹 서버에 부담을 주는 작업이란 특정 코드에서 병목 현상이 나타나거나 페이지 로드에서 부담을 주는 상황이 발생하는 작업을 말한다.

- API에서 대량의 데이터를 받아오는 작업
- 테이블에 대량의 데이터를 추가/수정하는 작업
- 긴 시간이 필요한 연산처리를 수행하는 작업

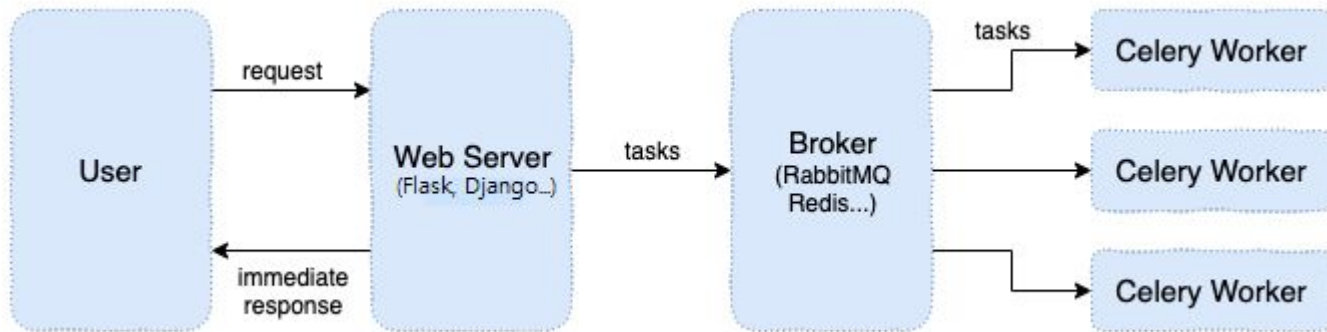


대량의 채팅과 오디오를
다운로드하는 작업

이러한 작업은 비동기 작업 큐를 사용하면 비동기적으로 요청에 대한 부하를 외부에 전가시킬 수 있다.

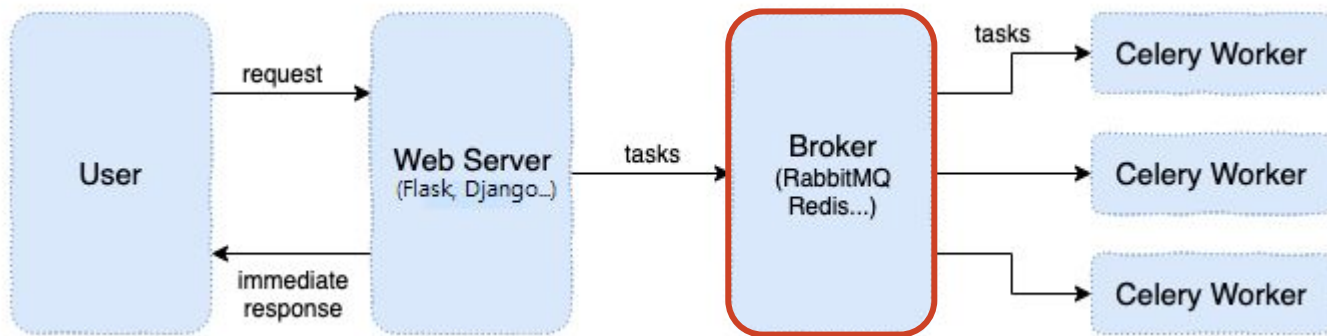
Celery의 아키텍처

웹 서버가 요청받은 작업을 **브로커**에게 전달하면 하나 이상의 **워커**가 작업을 받아서 처리한다.



Celery의 아키텍처

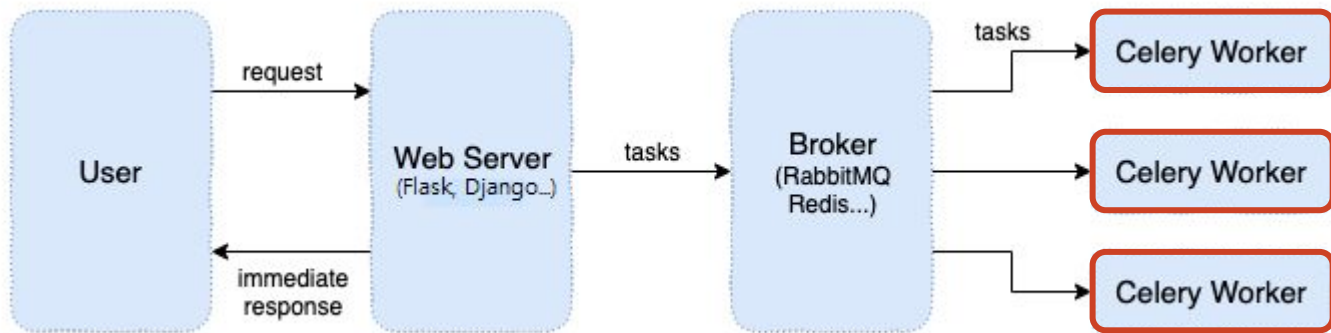
브로커 : 요청받은 작업들이 보관되어 있는 큐이다. 대기중인 작업을 관리하고, 워커에서 전달하기 위해서 중간 단계에서 관장하는 시스템이다.



요청받은 작업을 큐에 담아 보관했다가 적절한 워커가 가져가서 작업을 수행하게 된다. 일반적으로 redis, rabbitmq를 브로커로 사용한다.

Celery의 아키텍처

워커 : 작업을 브로커에서 가져와 실행한다. 일반적으로는 하나 이상의 워커가 있다.

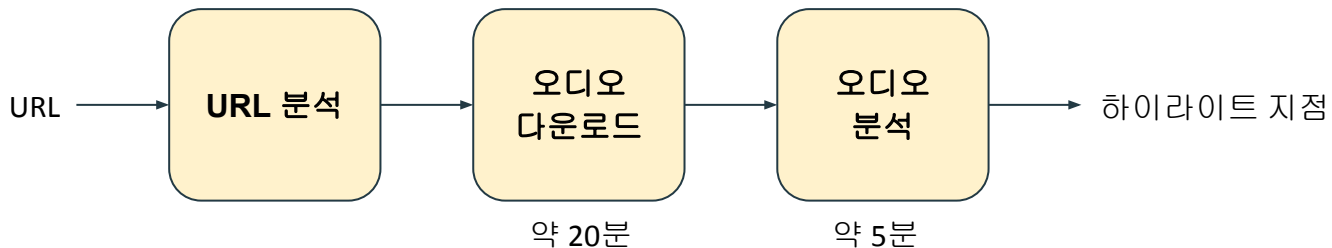


워커는 미리 처리하려는 작업 특정 사이즈만큼 미리 가져와서 큐에 대기 시켜놓고, 작업이 진행 중인 작업이 완료되면 브로커로 요청을 보내지 않고 미리 대기시켜두었던 작업을 꺼내와서 처리하는 **Prefetch 방식**으로 작동한다.

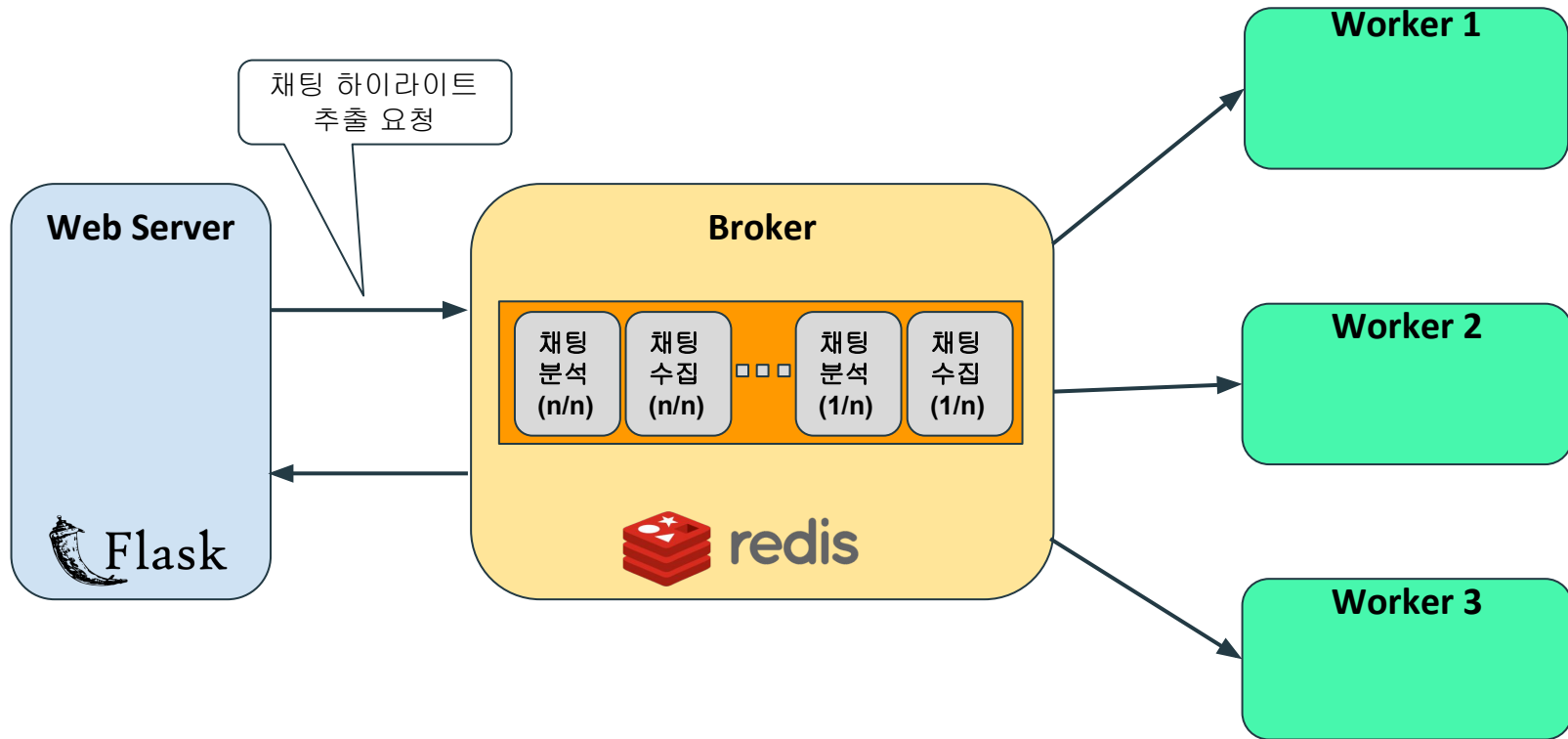
현 프로젝트에 적용 방안

기존의 프로젝트는 무거운 연산과 오래 걸리는 작업의 경우, 처리할 때까지 기다리고 다음 작업으로 넘어가는 방식이었다.

[예시] 4시간 길이의 아프리카TV 영상의 오디오 하이라이트 지점 추출 요청



현 프로젝트에 적용 방안



현 프로젝트에 적용 방안

