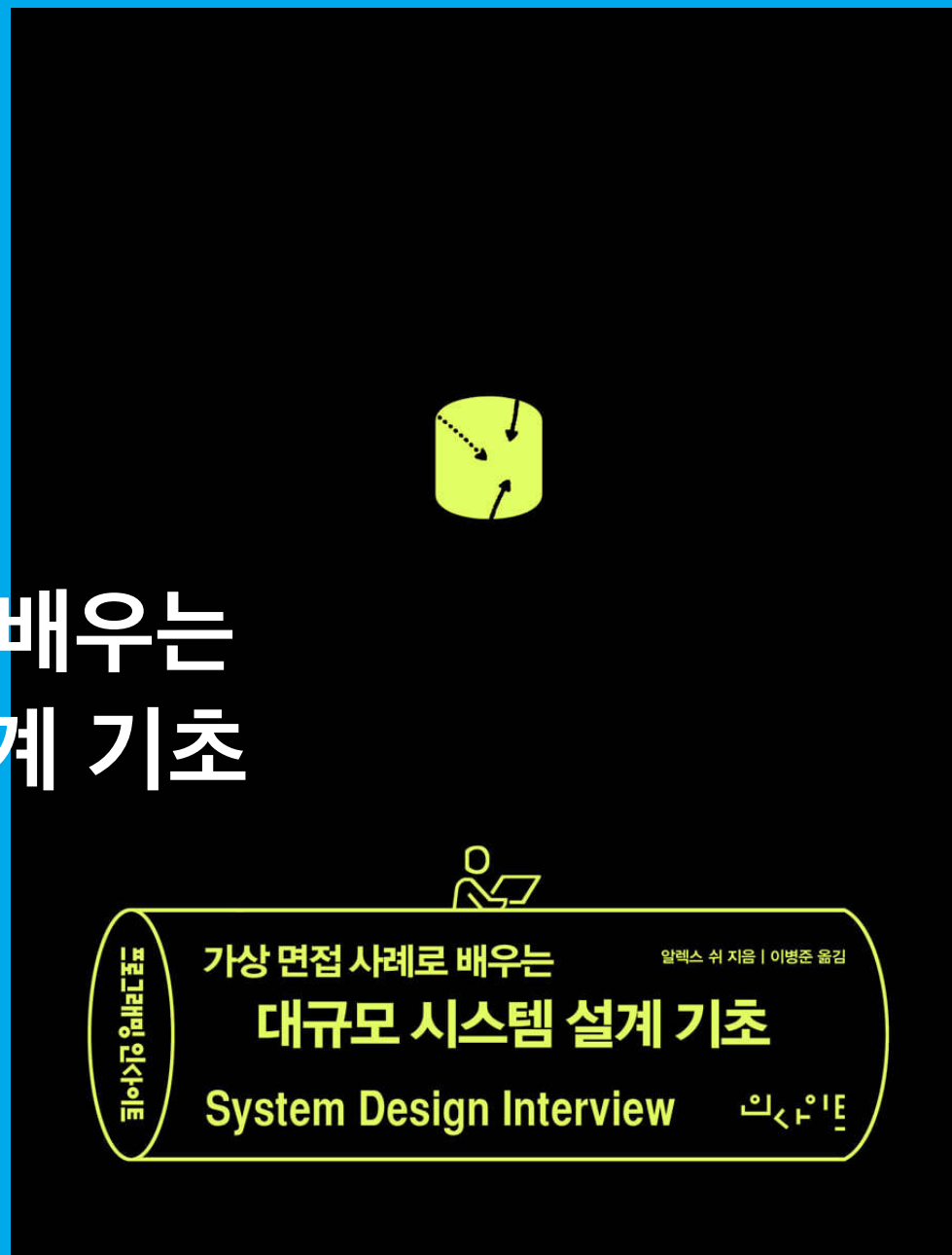


가상면접 사례로 배우는 대규모 시스템 설계 기초

PPT by 김주혁



목차

- 01 문제 이해 및 설계 범위 확정
- 02 개략적 설계안 제시 및 동의 구하기
- 03 상세 설계
- 04 마무리


9장
웹 크롤러 설계

크롤러 ... ?



크롤러

- **검색 엔진 인덱싱** : 크롤러의 가장 보편적인 예시로서, 웹 페이지를 모아 검색 엔진을 위한 로컬 인덱스를 만든다. 구글봇은 구글 검색엔진이 사용하는 웹 크롤러다.
- **웹 아카이빙** : 나중에 사용할 목적으로 장기보관하기 위해 웹에서 정보를 모으는 절차를 말한다.
- **웹 마이닝** : 크롤러를 사용해 정보를 모으고 해당 정보를 통해 데이터 마이닝을 하는 것을 의미한다.
- **웹 모니터링** : 웹 크롤러를 통해 저작권이나 상표권이 침해된 사례를 찾아낸다.




1단계 문제 이해 및 설계 범위 확정

웹 크롤러의 기본 알고리즘

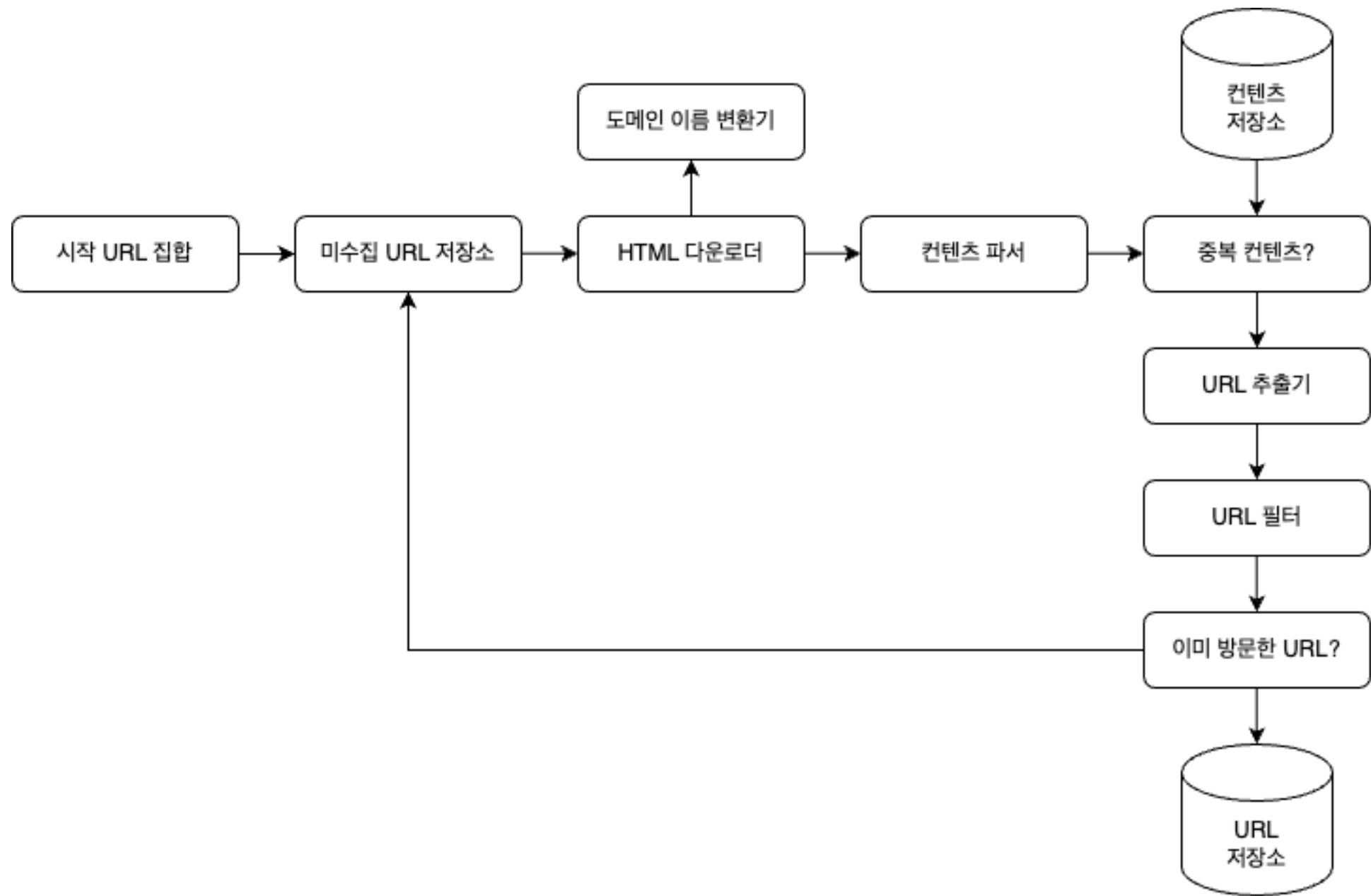
1. URL 집합이 주어지면, 해당 URL들이 가리키는 웹 페이지를 다운로드 한다.
2. 다운로드 웹 페이지에서 URL들을 추출한다.
3. 추출된 URL들을 다운로드 할 URL 목록에 추가하고 위의 과정을 처음부터 다시 반복한다.

웹 크롤러의 중요 속성

- **규모 확장성** : 웹은 거대하다.
병렬로 돌아갈 수 있다면 보다 더 효과적으로 크롤링 할 수 있을 것이다.
- **안정성** : 웹은 함정으로 가득하다.
잘못 작성된 HTML, 응답 없는 서버, 장애, 악성 코드 등 다양한 요소가 있다. 비정상적이거나 악의적인 요소에 대응할 수 있어야 한다.
- **예절** : 웹 사이트에 너무 잦은 요청을 보내서는 안된다.
- **확장성** : 새로운 형태의 콘텐츠를 지원하기가 쉬워야 한다.
이미지 비디오, pdf 등 다양한 콘텐츠에 대응 가능한 크롤러가 설계되어야 한다.



2단계 개략적 설계안 제시 및 동의 구하기



1. 시작 URL들을 미수집 URL 저장소에 저장
2. HTML 다운로더는 미수집 URL 저장소에서 URL 목록을 가져온다.
3. HTML 다운로더는 도메인 이름 변환기를 사용하여 URL의 IP 주소를 알아내고, 해당 IP 주소로 접속하여 웹페이지를 다운받는다.
4. 콘텐츠 파서는 다운된 HTML 페이지를 파싱하여 올바른 형식을 갖춘 페이지인지 검증한다.
5. 콘텐츠 파싱과 검증이 끝나면 중복 콘텐츠인지 확인하는 절차를 개시한다.
6. 중복 콘텐츠인지 확인하기 위해서, 해당 페이지가 이미 저장소에 있는지 본다.
 - 이미 저장소에 있는 콘텐츠인 경우에는 처리하지 않고 버린다
 - 저장소에 없는 콘텐츠인 경우에는 저장소에 저장한 뒤 URL 추출기로 전달한다.
7. URL 추출기는 해당 HTML 페이지에서 링크를 골라낸다.
8. 골라낸 링크를 URL 필터로 전달한다.
9. 필터링이 끝나고 남은 URL만 중복 URL 판별 단계로 전달한다.
10. 이미 처리한 URL인지 확인하기 위하여, URL 저장소에 보관된 URL인지 살핀다. 이미 있으면 버림
저장소에 없는 URL은 URL 저장소에 저장할 뿐 아니라 미수집 URL 저장소에도 전달한다.



3단계 상세 설계



DFS를 쓸 것인가,
BFS를 쓸 것인가

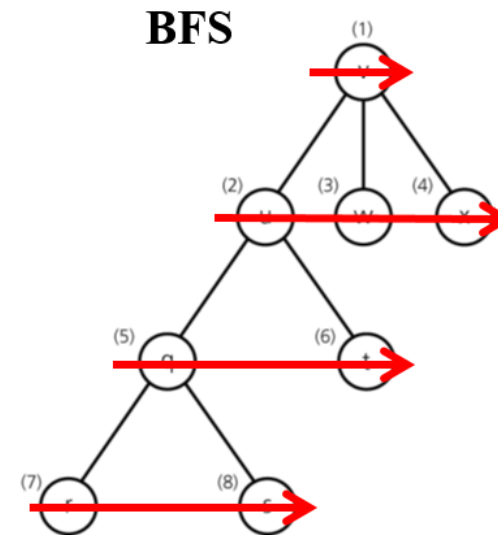
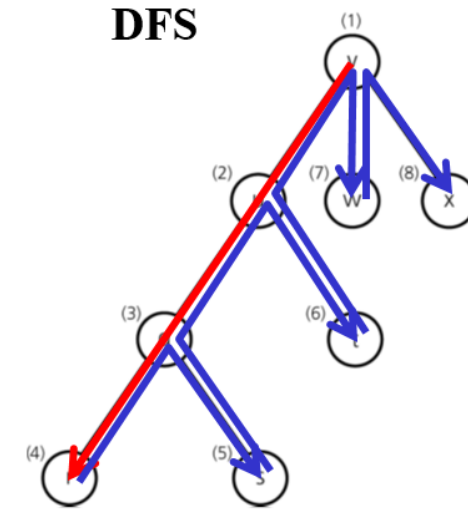
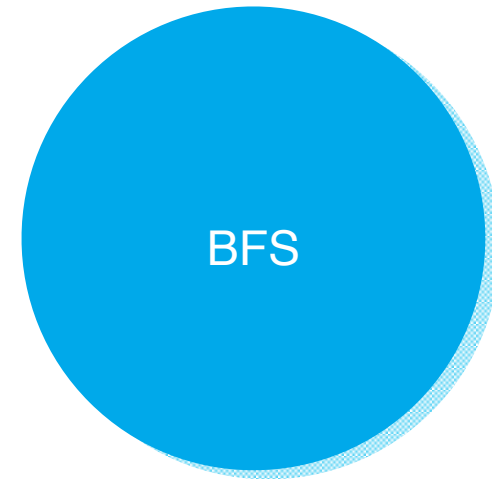
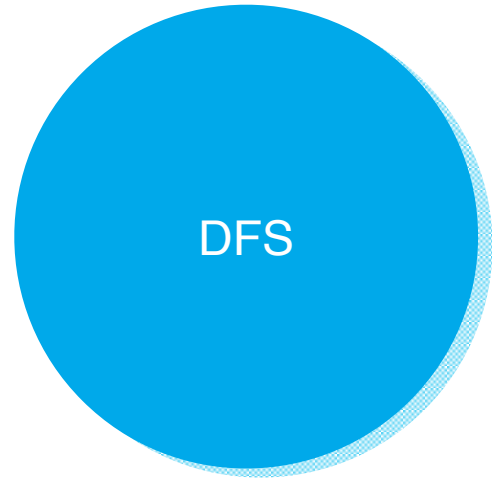
미수집 URL 저장소

HTML 다운로더

문제 있는 콘텐츠
감지 및 회피



DFS를 쓸 것인가,
BFS를 쓸 것인가



FIFO(First In First Out)





미수집 URL 저장소

예의

- Queue Router
- Mapping Table
- FIFO Queue
- Queue Selector
- Worker Thread

• 호스트	• 큐
• wikipedia	• a
• apple	• b
• ...	• ...
• naver	• z

우선순위

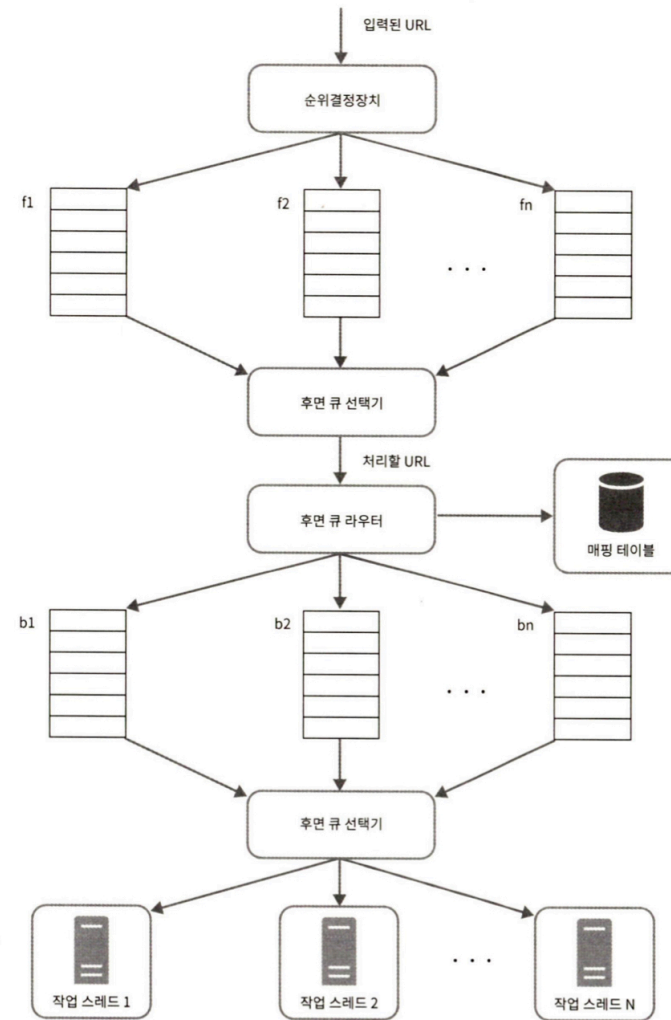


그림 9-8



HTML 다운로더

Robots.txt



shutterstock.com · 1994846876

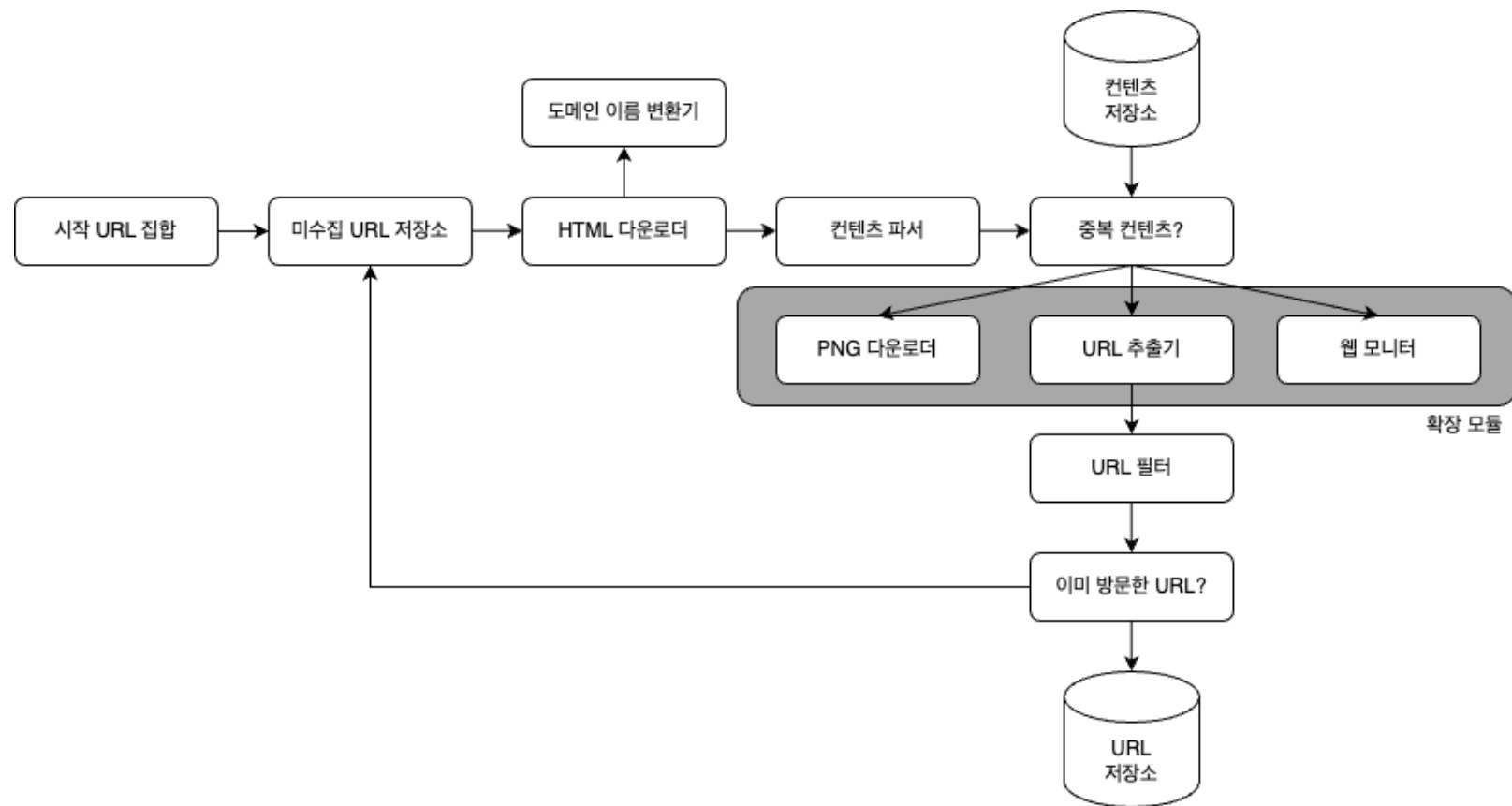
성능 최적화

- 분산 크롤링
- 도메인 이름 변환 결과 캐시
- 지역성
- 짧은 타임아웃

안정성

- 안정 해시 : 다운로드 서버들에 부하 분산을 적용할 때 사용할 수 있다.
- 크롤링 상태 및 수집 데이터 저장 : 장애가 발생한 경우도 쉽게 복구할 수 있도록 상태와 데이터를 수시로 지속적 저장장치에 기록해두는 것이 바람직하다.
- 예외 처리
- 데이터 검증

확장성



문제 있는 콘텐츠 감지 및 회피

- 중복 콘텐츠 : 해시나 체크섬을 통해 탐지할 수 있다.
- 거미 덩어리 : 크롤러가 무한 루프에 빠지도록 설계된 페이지의 경우 URL의 최대 길이를 제한하면 회피할 수 있다. 이런 사이트가 발견된다면 필터에 제한조건을 걸어두자.
- 데이터 노이즈 : 가치 없는 콘텐츠는 배제

4단계 마무리

- 서버 측 렌더링 : JS의 동작에 의해 링크가 생성되는 경우도 있기 때문에 페이지를 다운받아 파싱하기 전 다이내믹 렌더링을 거친다면 문제를 해결할 수 있을 것이다.
- 원치 않는 페이지 필터링 : 스팸 방지 컴포넌트를 두어 품질지 조악하거나 스팸성 페이지를 걸러내면 좋다.
- 데이터베이스 다중화 및 샤딩
- 수평적 규모 확장성 : Scale out될 수 있도록 준비해놓는 것이 좋고, 서버의 무상태성(stateless)을 유지하도록 만드는 것이 중요하다.
- 가용성 / 일관성 / 안정성
- 데이터 분석 솔루션