[9장] 웹 크롤러 설계

가상 면접 사례로 배우는 대규모 시스템 설계 기초

이민석 / unchaptered

https://inblog.ai/monthly-cs
https://github.com/monthly-cs/2024-03-system-design-interview-1

웹 크롤러(Web Crawler)란 무엇인가?

웹 크롤러는 웹사이트를 탐색하며 데이터를 수집하는 도구이다.

웹사이트에 포함된 링크(URL)을 따라가면서 정보를 수집하고 분류하며, 이 특징은 다음과 같이 활용된다.

1. 검색 엔진 인덱싱(Search Engine Indexing)
2. 웹 아카이빙(Web Archiving)
3. 웹 마이닝(Web Mining)
4. 웹 모니터링(Web Monitoring)

검색 엔진 인덱싱(Search Engine Indexing)

- [예시]

  Google, Naver, Bing 등

- [정의]

  검색 엔진이 검색 전에 정보를 구성하여 쿼리에 대한 초고속 응답을 가능하게 하는 프로세스

- [프로세스]

  역색인은 텍스트를 저장하는 데이터베이스를 해당 텍스트 문서와 이를 가리키는 포인터와 함께 컴파일합니다.

  이후, 검색엔진은 토큰화를 통해 단어를 핵심 의미로 줄여 데이터를 저장하고 검색하는데 필요

  줄입니다.

https://www.lumar.io/learn/seo/indexability/search-engine-indexing/#:~:text=Indexing%20is%20the%20process%20by,engines%20to%20identify%20relevant%20information.

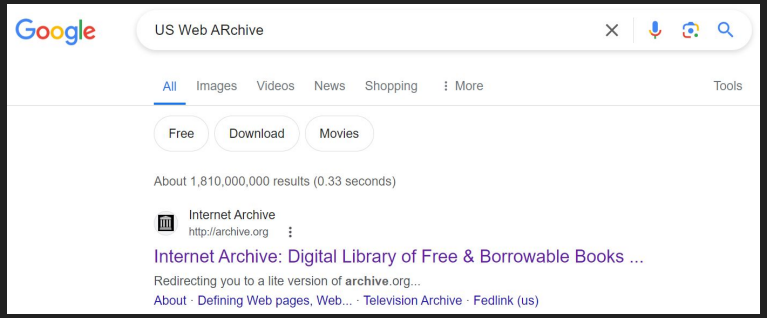
웹 아카이빙(Web Archiving)이란?

- [예시]

  US Library of Congress, EU Web Archive, Internet Archive

- [정의]

  나중에 사용할 목적으로 **장기보관**하기 위해서 웹에서 정보를 모으는 절차



https://www.google.com/search?q=US+Web+ARchive&oq=US+Web+ARchive&gs_lcrp=EgZjaHJvbWUyBggAEEUYOTIHCAEQABiABDIHCAIQABiABDIHCAMQABiABDIHCAQQABiABDIHCAUQABiABDIHCAYQABiABDIHCAcQLhiABDIHCAgQABiABDIHCAkQABiABNIBCDQwMDNqMGo5qAIAsAIB&sourceid=chrome&ie=UTF-8

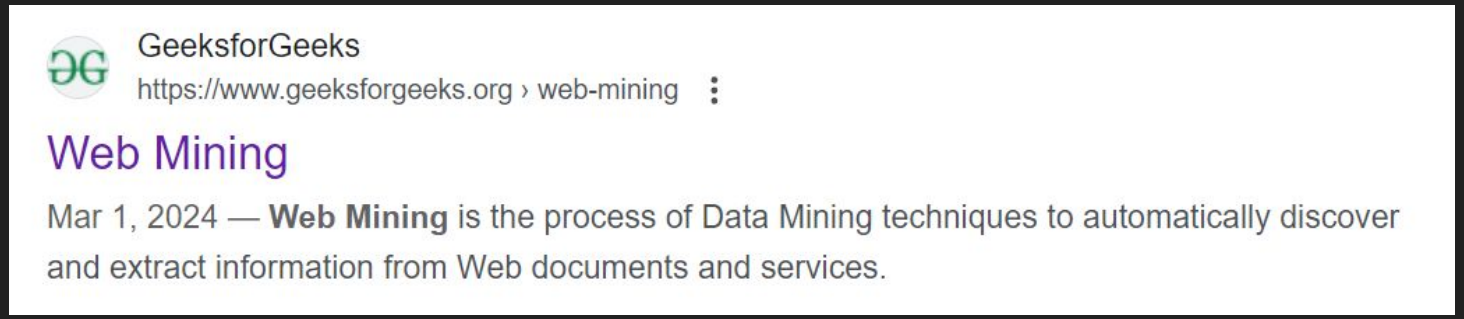
https://www.lumar.io/learn/seo/indexability/search-engine-indexing/#:~:text=Indexing%20is%20the%20process%20by,engines%20to%20identify%20relevant%20information.

웹 마이닝(Web Mining)이란?

- [정의]

  문서 및 서비스에서 정보를 **자동으로 검색하고 추출**하는 데이터 마이닝 기술 프로세스

  요구사항에 따라 관련 정보를 찾고 추출하기 위해 WWW(World Wide Web)에서 사용할 수 있는 시스템의 방대한 양의 데이터를 선별하는 가장 좋은 방법



https://www.geeksforgeeks.org/web-mining/

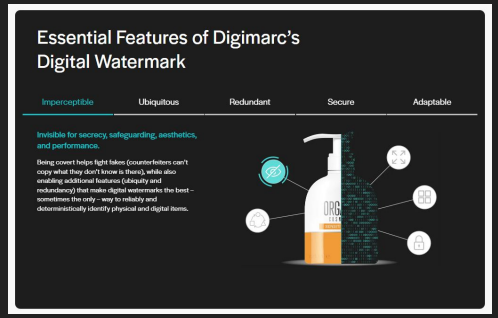
웹 모니터링(Web Monitoring)

- [정의]

  인터넷에서 저작권이나 상표권이 침해되는 사례 탐지

- [사례]

  Digimarc에서 다양한 웹 모니터링을 위한 서비스 및 기능을 지원



https://www.digimarc.com/product-digitization/data-carriers/digital-watermarks?utm_source=google+ads&utm_medium=search&utm_campaign=google+ads_search_Digital+Watermarks

결론

- 웹 크롤러(Web Cralwer)는 다음의 **기능이 반드시 포함**
  - **웹 사이트 탐색**
  - **웹 사이트 정보 저장**

요구사항 질의

- [목적]
  - 검색 엔진 인덱싱(Search Engine Indexing)
- [수량]
  - 1,000,000,000개 (10억개)
- [제한사항]
  - 새로 만들어진 웹 페이지, 수정용 웹 페이지 고려
  - 중복 콘텐츠를 가진 페이지는 무시
  - 수집한 웹 사이트는 5년 간 저장
- [추가 고려사항]
  - 규모 확장성 : 병행성(parallelism)을 활용
  - 안정성(robustness) : 잘못 작성된 HTML, 반응 없는 서버, 장애, 악성 코드가 붙은 링크
  - 예절(politeness) : 수집 대상 웹사이트에 적적량의 요청 전송
  - 확장성(extensibility) : 새로운 형태의 콘텐츠 지원이 쉬워야 함

요구사항 질의

[개요]

- 매달 10억 개의 페이지 다운로드
- 페이지당 평균 크기 = 500 KiB

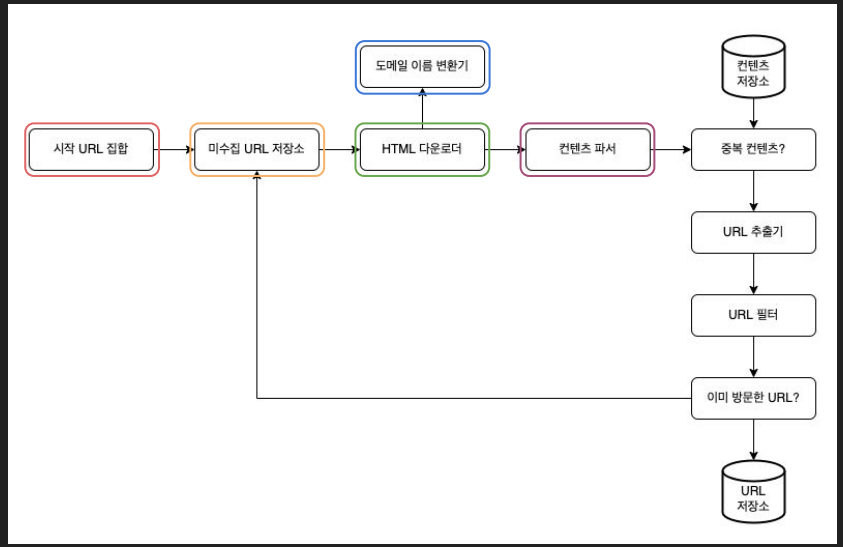
[QPS]

- 기본 QPS = 1,000,000,000 / 30 / 24 / 3600 ≒ **385**
- 피크 QPS = 기본 QPS * 2 ≒ 385 * 2 ≒ 770

[용량]

- 월간 필요 용량 = 1,000,000,000 * 500 KiB = 500,000,000,000 KiB ≒ 500 TiB
- 연간 필요 용량 ≒ 500TiB * 12month * 5year ≒ 30,000TiB ≒ 30 PiB

웹 크롤러 선행 연구 1



https://azderica.github.io/til/docs/dev/system-design-interview/ch9

시작 URL 집합

웹 크롤러가 크롤링을 시작하는 시작 지점
웹 크롤러가 가장 많은 페이지를 탐색할 수 있도록 별도의 전략이 필요

미수집 URL 저장소

다운로드 할 URL을 FIFO Queue 형태로 저장 (다운로드 할 URL 상태)
→ 상태 : 다운로드할 URL

HTML 다운로더

미수정 URL 저장소에서 URL을 꺼내고 HTML을 다운로드
→ 상태 : 다운로드 된 URL

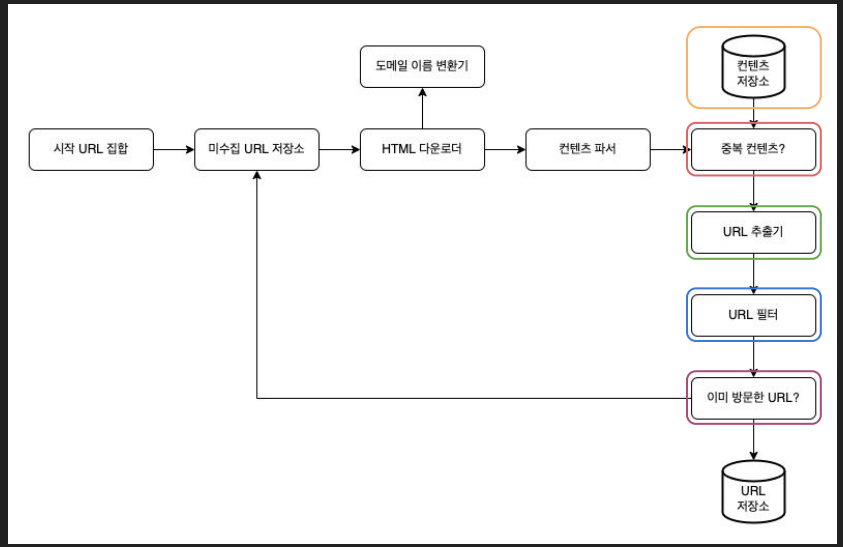
도메인 이름 변환기

URL을 IP 주소로 변환

컨텐츠 파서

웹 사이트에 대한 파싱 및 유효성 검증 (성능 및 보안 관점)

웹 크롤러 선행 연구 2



https://azderica.github.io/til/docs/dev/system-design-interview/ch9

중복 컨텐츠?

A와 B 페이지의 중복을 검사
전문을 비교하는 것은 너무 느리기에, 두 페이지의 해쉬값을 비교

컨텐츠 저장소

인기있는 컨텐츠 → 메모리 기반 저장소
일반 컨텐츠 → 디스크 기반 저장소

URL 추출기

상대 경로를 전부 절대 경로로 치환

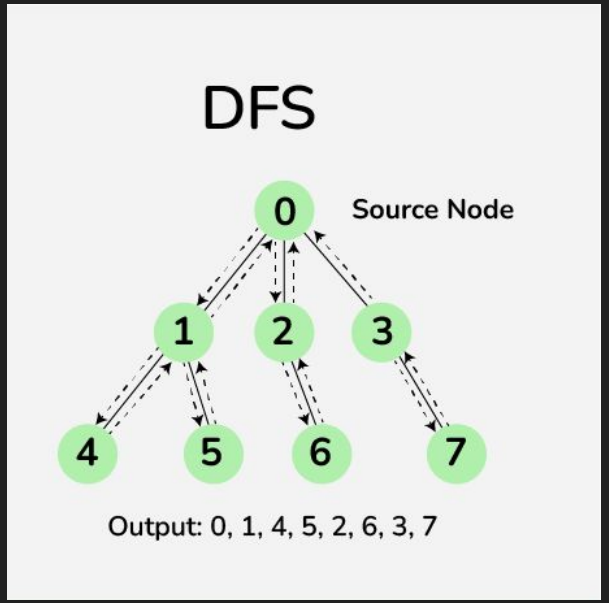
URL 필터

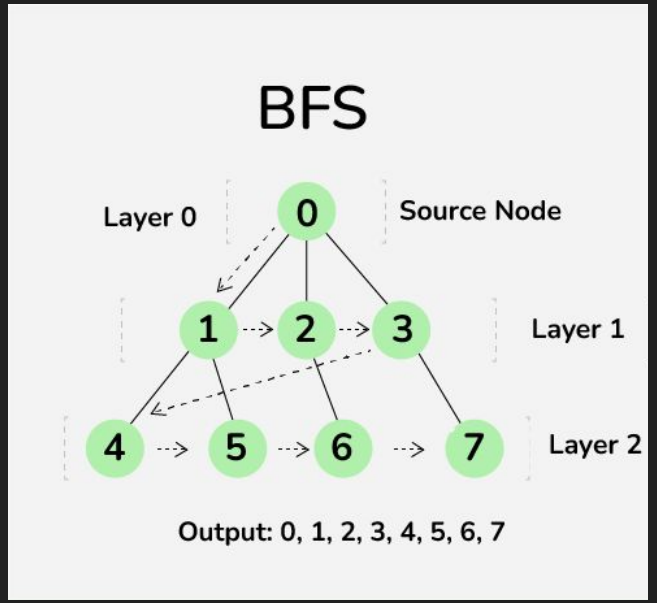
특정한 콘텐츠 타입, 파일 확장자를 갖는 URL은 제거

이미 방문한 URL?

블룸 필터나 해시 테이블을 이용하여 URL의 중복 방문을 방지
이미 방문한 URL은 URL 저장소에 기록이 되어 있음

DFS vs BFS





https://www.geeksforgeeks.org/difference-between-bfs-and-dfs/

예의

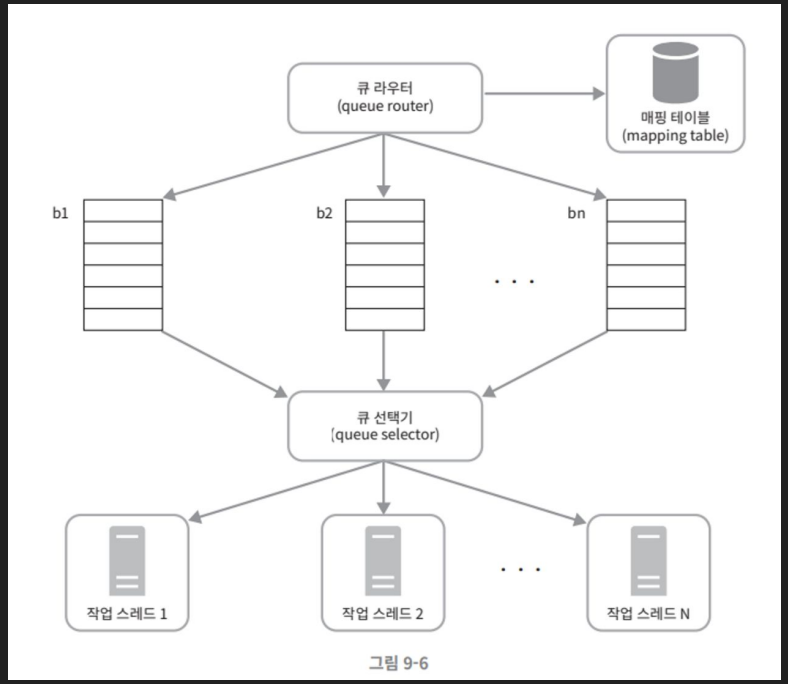


그림 9-6

https://velog.io/@vixloaze/9%EC%9E%A5-%EC%9B%B9-%ED%81%AC%EB%A1%A4%EB%9F%AC-%EC%84%A4%EA%B3%84

큐 라우터

같은 호스트 소속의 URL을 같은 큐(b1, b2, …, bn)으로 가도록 제어

매핑 테이블

호스트 이름 - 큐 사이를 보관

wikipedia.com - b1
apple.com - b2

선입선출 큐(FIFO Queue)

같은 호스트에 속한 URL은 언제나 같은 큐(b1, b2, …, bn)에 보관

큐 선택기

여러 큐(b1, b2, …, bn)들을 순회하면서 한 지정된 큐를 작업 스레드에 전달

작업 스레드

전달된 URL을 다운로드

우선순위

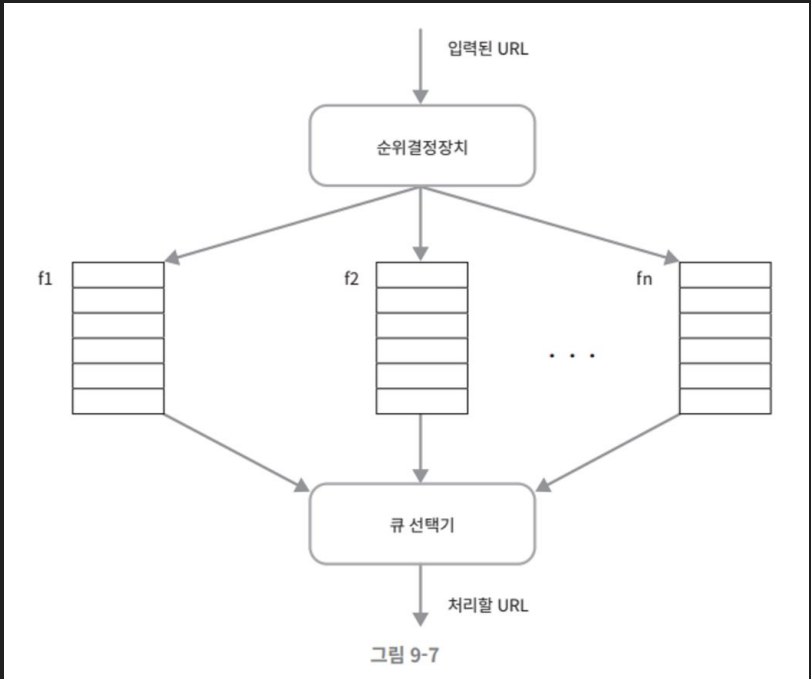


그림 9-7

https://velog.io/@vixloaze/9%EC%9E%A5-%EC%9B%B9-%ED%81%AC%EB%A1%A4%EB%9F%AC-%EC%84%A4%EA%B3%84

순위 결정 장치

URL을 입력 받아 우선순위 계산

큐(f1, …, fn)

우선 순위 별로 큐가 하나씩 할당
우선 순위에 따라서 선택 확률이 높아짐

큐 선택기

임의 큐에서 처리할 URL을 선택
순위 결정 장치에서 정한 순위에 따라서 확률 적으로 큐를 선택

신선도 + 지속적 저장장치

[신선도]

- 정보의 신선도를 유지하기 위해서, 이미 수집한 데이터도 주기적으로 재수집해야 함
- 이 때, 재수집의 빈도는 웹 페이지의 변경 이력 혹은 우선 순위를 참조

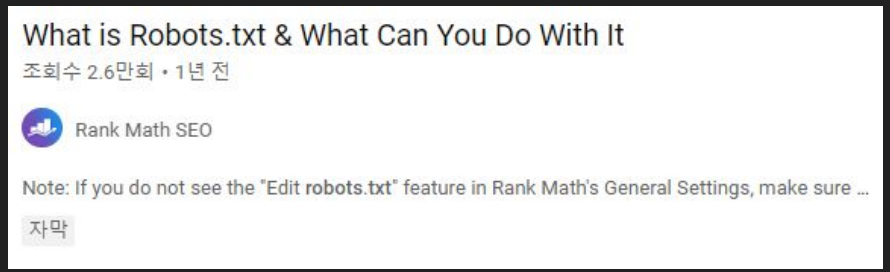
[지속적 저장장치]

- 인기 있는 URL은 메모리 기반 저장소에 저장
- 일반 URL은 디스크 기반 저장소에 저장

Robots.txt

웹 사이트의 개발자는 다양한 목적에 따라서 robots.txt라는 파일을 만들고 있음

- 웹 크롤링 방지의 목적 (도서에서 소개)
- 웹 크롤링이 잘 되게 하기 위한 목적 (그 외에도… 웹툰 불법다운로드 사이트를 보면, 웹툰 제목들이 따로 리스트업 되어 있음)



https://www.youtube.com/watch?v=qRlQ965pGCA

성능 최적화

기본적으로 분산 크롤링을 통해서 성능을 극대화한다.
1. 하지만 동기적으로 작동하는 DNS 쿼리로 인한 성능 저하가 발생할 수 있음
2. 분산 서버가 크롤링 대상에 가까운 곳에 있도록 지역 별로 분산할 수 있음
3. 타임아웃 설정 필요

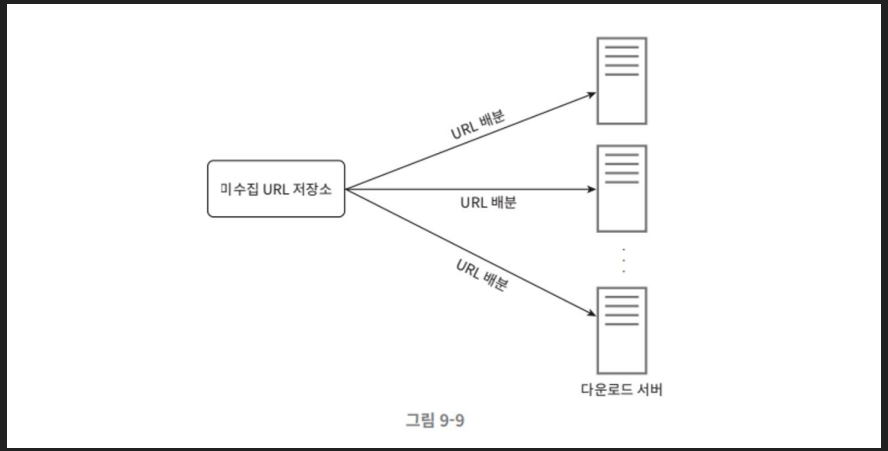


그림 9-9

https://azderica.github.io/til/docs/dev/system-design-interview/ch9

안정성

안정 해시를 통해서 부하 분산

지속적 저장 장치를 이용해서 크롤링 상태 및 수집 데이터를 보관

예외 처리

데이터 검증

https://velog.io/@vixloaze/9%EC%9E%A5-%EC%9B%B9-%ED%81%AC%EB%A1%A4%EB%9F%AC-%EC%84%A4%EA%B3%84

확장성

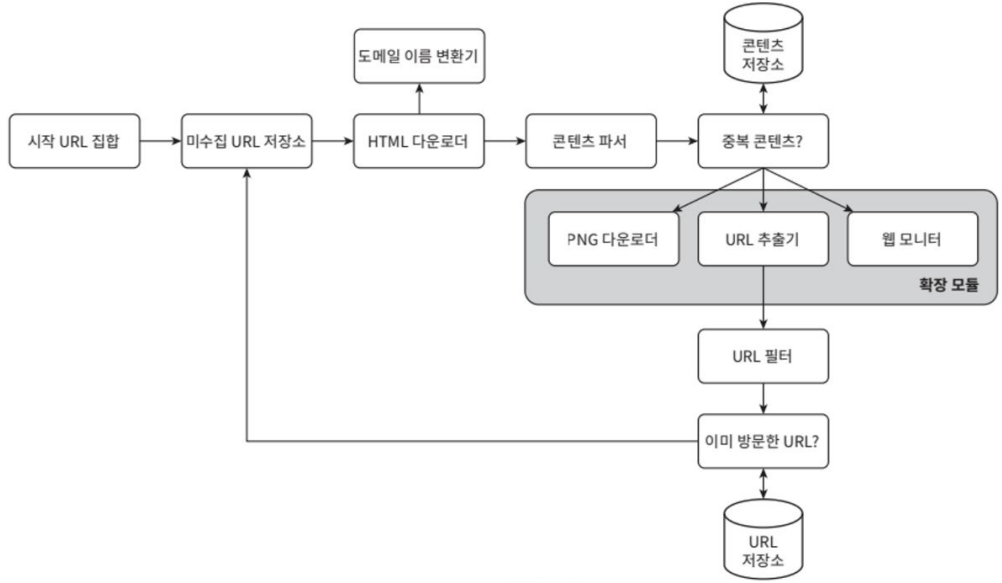


그림 9-10

https://azderica.github.io/til/docs/dev/system-design-interview/ch9

문제 있는 콘텐츠 감지 및 회피

중복 컨텐츠는 해시/체크섬을 이용해서 필터링

거미 덫 등은 URL의 최대길이를 제한하는 것으로 막음

→ 그 외…? https://www.marketingtracer.com/seo/crawler-traps

무의미 한 데이터(데이터 노이즈)들을 필터링해서 제거

→ <script> 태그 등

https://velog.io/@vixloaze/9%EC%9E%A5-%EC%9B%B9-%ED%81%AC%EB%A1%A4%EB%9F%AC-%EC%84%A4%EA%B3%84

추가로 논의할 것들에 대하여

- 서버 측 랜더링(Server-side Rendering)
- 원치 않는 페이지 필터링
- 데이터베이스 다중화 및 샤딩
- 수평적 규모 확장성

감사합니다.