가상 면접 사례로 배우는

대규모 시스템 설계 기초

이정민

9. 웹 크롤러 설계

개요

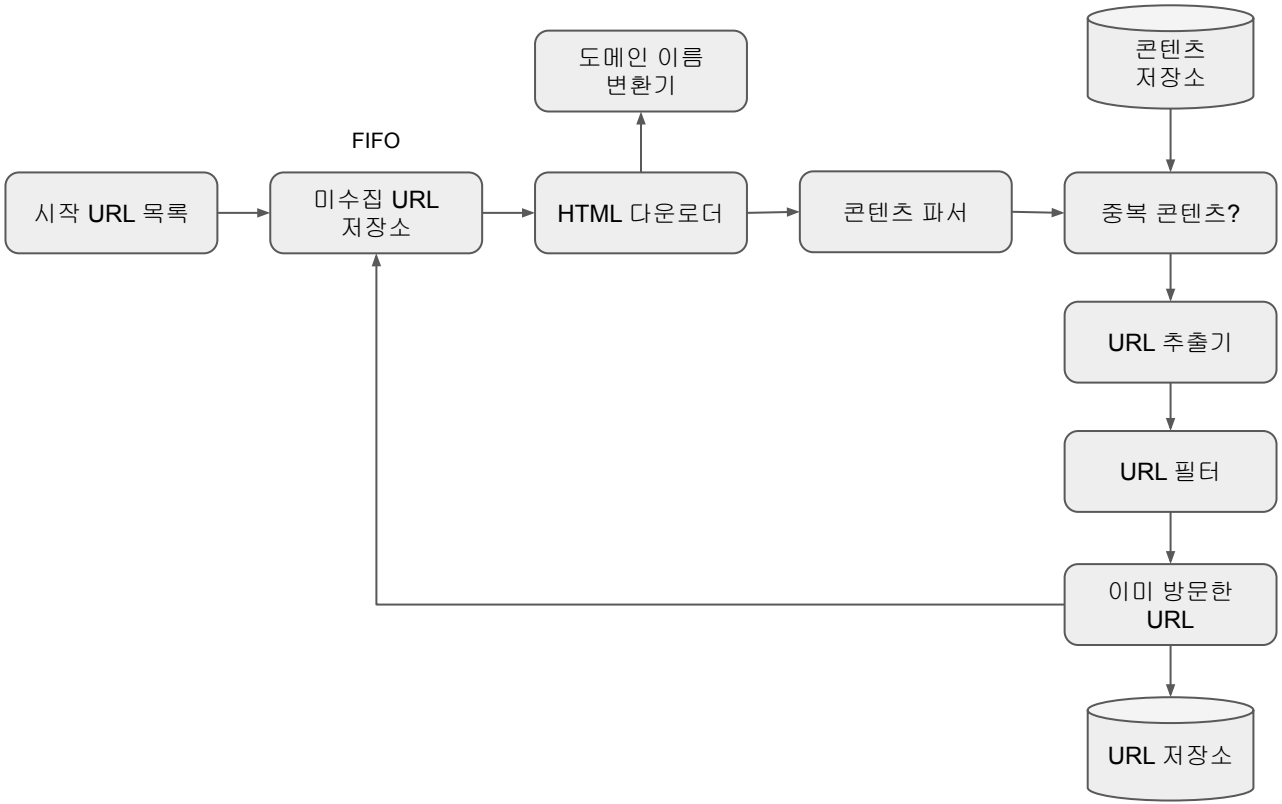
■ 웹 크롤러란?

- 로봇 또는 스파이더. 웹에 새로 올라오거나 갱신 된 콘텐츠를 찾아내는 목적(수집, 분석)으로 사용
- 검색 엔진 인덱싱 : 웹 페이지를 모아 검색엔진을 위한 로컬 인덱스 생성
- 웹 아카이빙 : 장기 보관을 위해 웹에서 정보를 수집 (미국 국회 도서관, EU 웹 아카이브)
- 웹 마이닝 : 웹 페이지 내용에서 유용한 정보를 추출
- 웹 모니터링 : 인터넷에서 저작권이나 상표권이 침해되는 사례 모니터링 등에 활용

1단계) 문제 이해 및 설계 범위 확정

- 기본 알고리즘
  - URL 목록을 입력받아 모든 웹 페이지를 다운로드
  - 다운로드한 웹 페이지에서 URL 목록 추출
  - 추출한 URL 목록을 입력으로 위의 과정 반복
- 요구사항
  - 주된 용도 : 검색 인덱싱
  - 매달 웹페이지 수집량 : 10억개
  - 대상 : 신규 및 수정된 웹 페이지, 중복 콘텐츠 무시
  - 5년간 저장 필요
- 추가 고려 요소
  - 규모 확장성 : 거대한 웹 페이지. 병렬성 활용 필요
  - 안정성 : 무반응, 장애, 악성 코드 등 대응 필요
  - 예절 : 짧은 시간 대량의 요청 방지
  - 확장성 : 새로운 형태의 콘텐츠(이미지 등) 지원
- 개략적 규모 추정
  - QPS = 400페이지/초, 최대 = 800페이지/초
    - 1,000,000,000 /30일/24시간/3600초 = 400페이지
  - 5년간 저장 용량 = 30PB 500TB/월
    - 웹 페이지 평균 크기 = 500K (가정)
    - 10억 페이지 * 500K = 500TB/월,
    - 500TB*12개월*5년 = 30PB

2단계) 개략적 설계안 제시 및 동의 구하기



3단계) 상세 설계

■ 주요 구성요소 및 기술

- DFS(Depth-First Search) vs BFS(Breath-First Search)
- 미수집 URL 저장소
- HTML 다운로더
- 안정성 확보 전략
- 확정성 확보 전략
- 문제 있는 콘텐츠 감지 및 회피 전략

3단계) 상세 설계

- Directed Graph 탐색 알고리즘 : DFS(깊이 우선) vs BFS(너비 우선)
  - 보통 BFS 사용
  - BFS 사용 시 문제점
    - 한 페이지에서 추출된 링크의 대다수는 동일 서버의 내부 링크, 병렬 처리 시 동일 서버에 과부하
    - 표준 BFS 알고리즘은 탐색 우선순위가 없음, 페이지 순위, 트래픽 양, 업데이트 빈도 고려 필요

3단계) 상세 설계

■ 미수집 URL 저장소

- 예의
  - 동일 웹 사이트에 대해서 한번에 한 페이지만 요청
  - 매핑 테이블 : 호스트 이름과 큐를 매핑
  - 큐 라우터 : 동일 호스트의 URL은 하나의 큐로 가도록 보장
  - FIFO 큐(n개) : 동일 호스트의 URL은 하나의 큐에 보관
  - 큐 선택기 : 큐들을 순회하면서 URL을 꺼내어 다운로드 작업 스레드에 전달
  - 작업 스레드 : 전달 받은 URL을 다운로드 (다운로드 간 delay 설정)

3단계) 상세 설계

■ 미수집 URL 저장소

- 우선 순위(순위 결정 장치)
  - 순위결정장치 : URL을 입력받아 우선순위를 계산
  - 큐(n개) : 우선 순위별로 큐를 하나씩 할당
  - 큐 선택기 : 우선 순위가 높은 큐에서 더 자주 URL을 꺼냄
- 통합 설계 (우선순위 + 예의)
  - 전면 큐 : 우선순위 결정 과정 처리
  - 후면 큐 : 예의 바르게 동작하도록 보장

3단계) 상세 설계

■ 미수집 URL 저장소

- 신선도 : 이미 다운로드한 콘텐츠라도 재수집 할 필요가 있음
  - 웹 페이지의 변경 이력 활용
  - 우선순위를 활용하여 중요한 페이지는 좀 더 자주 재수집
- 미수집 URL 저장소를 위한 지속성 저장장치
  - 메모리 보관 시 안정성이나 규모 확장성 측면에서 바람직 하지 않음
  - 전부 디스크에 저장하면 느려서 성능 병목이 생김
  - 절충안 : 디스크 + 메모리 버퍼

3단계) 상세 설계

■ HTML 다운로더

- Robots.txt (로봇 제외 프로토콜)
  - 웹 사이트가 크롤러와 소통하는 표준적 방법. 크롤러가 수집해도 되는 페이지 목록 보관
  - 웹 사이트 크롤링 전 해당 파일에 나열된 규칙을 먼저 확인 필요
  - Robots.txt의 반복적 다운로드를 피하기 위해 캐시에 보관

3단계) 상세 설계

■ HTML 다운로더

- 성능 최적화
  - 분산 크롤링 : 크롤링 작업을 여러 서버에 분산. 각 서버는 멀티 스레드로 다운로드 작업 처리
  - 도메인 이름 변환 결과 캐시 : 도메인 이름 변화기는 병목 지점이 될 수 있으므로 캐시에 보관하고 주기적으로 갱신
  - 지역성 : 크롤링 작업 서버를 지역 별로 분산. 다운로드 시간을 줄임 (크롤 서버, 캐시, 큐, 저장소 등 적용 가능)
  - 짧은 타임아웃 : 서버 응답이 없거나 느릴 경우 대기 시간이 길어지므로 짧게 설정 한 타임아웃 초과 시 스킵

3단계) 상세 설계

■ HTML 다운로더

- 안정성
  - 안정 해시 : 부하 분산 시 적용. 다운로더 서버를 쉽게 추가/제거 가능
  - 크롤링 상태 및 수집 데이터 저장 : 장애 발생 시 쉽게 복구 할 수 있도록 크롤링 상태, 수집 정보를 지속적으로 저장
  - 예외 처리 : 예외가 발생하더라도 전체 시스템이 중단되는 일 방지
  - 데이터 검증 : 시스템 오류를 방지하기 위한 중요 수단 중 하나

3단계) 상세 설계

■ HTML 다운로더

- 확장성
  - 새로운 형태의 콘텐츠를 쉽게 지원 가능해야 함
  - 새로운 모듈을 추가하여 새로운 형태의 콘텐츠를 지원
  - URL 추출기 외에 PNG 다운로드 또는 웹 모니터 플러그인 모듈 추가

3단계) 상세 설계

■ HTML 다운로더

- 문제 있는 콘텐츠 감지 및 회피
  - 중복 콘텐츠 : 웹 콘텐츠의 30%가량 중복. 해시나 체크섬 사용
  - 거미 덫 : 크롤러를 무한 루프에 빠뜨리도록 설계한 웹 페이지 (무한히 깊은 디렉토리 구조를 포함하는 링크)

    URL 최대 길이 제한으로 회피. 수작업으로 확인 후 URL 필터 목록 추가
  - 데이터 노이즈 : 광고, 스크립트 코드, 스팸 URL 등

4단계) 마무리

■ 추가 논의 사항

- 서버 측 렌더링
- 원치 않는 페이지 필터링
- 데이터베이스 다중화 및 샤딩
- 수평적 규모 확장성
- 가용성, 일관성, 안정성
- 데이터 분석 솔루션