CS 정리

Date	@2023년 1월 25일
Tags	취준스터디
출처	
참고자료	

개요

지금까지 공부했던 내용들을 대표적인 면접 질문들을 중심으로 정리해보고자한다.

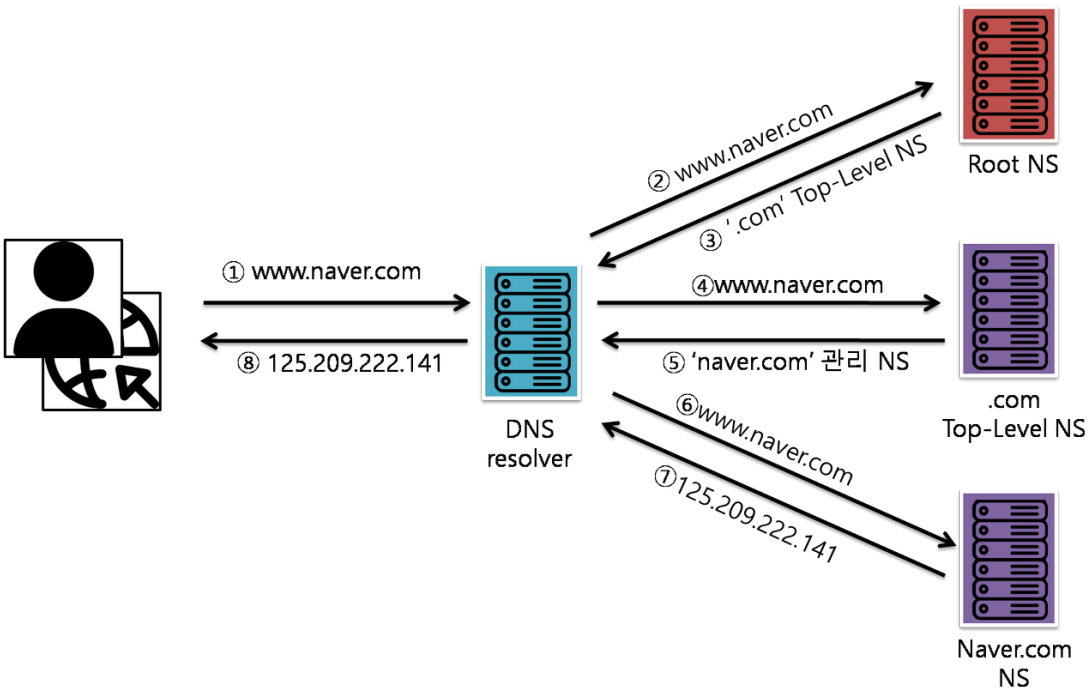
네트워크

검색창에 www.google.com을 치면 생기는 일

1. **브라우저는 캐싱된 DNS 기록에서 `www.google.com`에 대응하는 IP 주소가 있는지 검색한다.**

   여기서 **DNS(Domain Name System)**란 IP주소와 Host 이름을 서로 연결시켜주는, **분산 구조화된 트리 구조**를 말한다.

   > 호스트에 대한 이름 주소 변환을 체계적이고,안정적이고,효율적으로 하기 위해, 계층적이고,분권화된,클라이언트/서버 구조의 데이터베이스 시스템을 말한다.



호스트에게 받은 도메인 주소를 ISP의 DNS recursor는 다음과 같은 과정을 통해 IP 주소를 제공한다.

1. Root Name Server에서 도메인 주소를 찾는다
2. 1번에서 못찾으면 최상위 도메인(.com) Top Level Name Server 에서 도메인 주소를 찾는다
3. www.google.com 에 해당하는 Name Server를 알게되면, 해당 Name Server에서 도메인 주소를 찾는다.
4. Name Server에서 www.google.com의 IP 주소를 받는다.

여기서 **DNS 캐싱**이란, DNS 서버가 한번 요청된 DNS 질의를 TTL 만큼 메모리에 저장하여 두었다가 똑같은 질의에 대해 신속히 처리 할 수 있도록 하는 기능이다.

- Positive Caching - 일반적으로 이해하고 있는 캐싱을 말함
- Negative Caching - 잘못된 DNS 질의가 반복적으로 발생되는 것을 막기위해 일정기간 잘못된 질의를 캐싱하는 것을 말한다.

2. **캐시에 해당 기록이 없으면, ISP의 DNS 서버에서 다른 DNS 서버를 DNS Query를 통해 검색하여 IP 주소를 찾는다.**

DNS Query

재귀적 질의

질의된 도메인에 대해 즉각 응답하거나, 다른 서버에게 질의한 결과로 응답하거나, 찾고 있는 정보가 없다는 에러 메세지를 보내 준다. 가장 단순한 DNS 쿼리

유형이라고 볼 수 있다

반복적 질의

질의된 도메인에 대해 응답하거나,아니면 **이 작업을 할 수 있는 다른 DNS 서버에 클라이언트를 연결** 시켜 주는 작업을 한다.

자신이 관리하지 않는 알수 없는 질의에 대한 응답 가능한 네임 서버 목록을 전달함.

이를 통해 클라이언트가 다수의 DNS 서버에 같은 질의를 반복할 수 있게 된다.

3. **받은 IP 주소를 사용하여 TCP 통신을 통해 해당 IP 서버와 브라우저가 연결한다.**

TCP (Transmission Control Protocol)

OSI 계층 모델의 관점에서 전송계층(4계층)에 해당한다. 양 종단 호스트 내 프로세스 상호간에 **신뢰적인 연결지향성 서비스**를 제공하는 프로토콜이다.

특징

신뢰성 있음

패킷 손실, 중복 , 순서바뀜 등이 없도록 보장한다.

TCP의 하위 계층인 IP 계층의 신뢰성 없는 서비스에 대해 다방면으로 신뢰성을 제공한다

연결지향적

같은 전송계층의 UDP가 비연결성인 것과는 달리, TCP는 연결지향적이다.

느슨한 연결 : 중간 연결층에서는 신경쓰지 않고 종단에서 전송순서와 에러등을 처리하는 것을 말함

연결 관리를 위한 연결설정 및 연결해제가 필요 : 양단간 어플리케이션/프로세스는 TCP가 제공하는 연결성 회선을 통하여 서로 통신

흐름제어

송신 및 수신 속도를 일치시키는 것

주로 순서번호,확인응답번호,수신윈도우 크기 라는 3개의 변수로 흐름제어를 한다

관련 키워드 : 슬라이딩 윈도우

혼잡제어

네트워크가 혼잡하다고 판단될 때 송신률을 감속함

이에따른 여러 장치들을 다양한 TCP 버전에서 마련하고 있다 → 느린시작, 혼잡회피,수신 윈도우 및 혼잡 윈도우 크기 결정

혼잡제어를 위해 수신 윈도우 및 혼잡 윈도우 2개의 변수를 관리

관련 키워드 : TCP 혼잡 제어

4. **브라우저와 서버가 연결되었다면 서버는 데이터를 전송한다. 브라우저는 HTTP 프로토콜로 GET 요청을 통해 `www.google.com` 의 웹페이지 데이터를 요구한다.**

   GET

   리소스를 조회하는 메서드

   Path에 있는 자원을 요청한다

5. **요청을 받은 서버는 응답을 생성해 보낸다. 이때 응답은 특정한 포맷(Json,Xml,HTML)등으로 작성한다. 웹 서버 혼자서 모든 요청을 수행하면 서버에 과부하가 일어날 수 있기 때문에 이를 돕는 WAS 를 사용한다.**

   WAS ? WA ?

   Web Application

   소프트웨어와 하드웨어로 구분되며, 하드웨어는 웹 서버가 설치되어있는 컴퓨터를 말하며 소프트웨어는 웹 브라우저 클라이언트로부터 HTTP 요청을 받아 **정적인 컨텐츠**를 제공하는 컴퓨터 프로그램

   HTTP 프로토콜을 기반으로 하여 클라이언트의 요청을 서비스하는 기능을 담당한다

   정적인 컨텐츠를 제공한다 → WAS 를 거치지 않고 바로 자원을 제공

   동적인 컨텐츠 제공을 위한 요청 전달 → Request를 WAS에 보내고 WAS가 처리한 결과를 클라이언트에게 전달한다.

   정적 컨텐츠만 처리하도록 기능을 분배하여 서버의 부담을 줄이기 위해 필요하다.

   웹 서버를 통해 어플리케이션 서버까지 가지 않고 앞단에서 빠르게 보내줄 수 있다.

   예) 아파치 서버,엔진엑스 등등..

   Web Application Server

   DB 조회나 다양한 로직 처리를 요구하는 **동적인 컨텐츠**를 제공하기 위해 만들어진 Application Server

   HTTP를 통해 컴퓨너타 장치에 애플리케이션을 수행해주는 미들웨어이다.

   Web Server + Web Container

Web Server의 기능들을 구조적으로 분리하여 처리하고자 하는 목적으로 제시되었다.

웹서버만 이용하면 사용자가 원하는 요청에 대한 결과값을 모두 미리 만들어놓고 서비스를 해야한다 → 하지만 자원이 부족하기 때문에 불가능함

WAS를 통해 요청에 맞는 데이터를 데이터베이스에서 가져와 비즈니스 로직에 맞게 그때 그때 결과를 만들어서 제공함으로써 자원을 효율적으로 사용할 수 있다.

예) 톰캣 , Jeus

6. **브라우저가 렌더링되어 사용자에게 `www.google.com` 의 화면이 보여지게된다. 이때 정적인 파일들은 브라우저에 의해 캐싱되어 다시 방문할 경우 다시 데이터를 요청하지 않도록 한다.**

HTTP 1.1 vs 2.0 vs 3.0

1.x (표준 프로토콜)

Header + Body 구성

Header에는 URI, Request method,여러 헤더 정보가 포함되어있음.

HTTP Body가 문자열로 구성됨

(1.1) HOL blocking 문제가 발생함 → 패킷의 순서를 보장하기 위해 패킷이 도착할 때까지 다른 패킷은 전송되지 못하는 것

TCP 커넥션을 이용 (3-way-handshake)

요청에 대한 응답이 끝나기 전 다음 데이터를 미리 요청 → 파이프라이닝 추가

2 (성능 향상 버전)

바이너리가 아닌 텍스트로 데이터를 전송

TCP를 사용

요청 데이터가 동기적으로 진행 (1개를 요청하면 올때까지 대기하는 등..)

헤더에 중복된 데이터

HTTP Body가 이진 데이터, binary framing layer이라는 공간에 이진 데이터로 전송됨

스트림

여러 요청/응답 병렬 처리 (하나의 TCP 연결에 여러 스트림)

TCP 연결이 1개이므로 3-way-handshake 오버헤드 X

네트워크 가용성 증가로 속도 상승,이미지 스프라이트 등이 필요없어짐

3

UDP기반의 QUIC 프로토콜을 사용하여 통신

Zero RTT

패킷 손실에 대한 빠른 대응

사용자 IP가 바뀌어도 연결이 유지되는 것

동영상 서비스에서 큰 차이를 보임

TCP와 UDP의 차이

TCP는 연결형 서비스로 3-way-handshaking 과정을 통해 연결을 설정한다. 그렇기 때문에 높은 신뢰성을 보장하지만 속도가 느리다는 단점이있다. 이와 달리 UDP는 TCP에서 했던 신뢰성 보장이 없다고 보면 된다. 신뢰성이 떨어지는 단점이 있지만 수신 여부를 확인하지 않기 때문에 속도가 매우 빠르다.

TCP의 경우 신뢰성이 중요한 이메일전송,파일 교환과 같은 경우에 사용되며 UDP의 경우 실시간성이 중요한 스트리밍 서비스에 사용이 된다.

GET vs POST

Get은 데이터를 조회하기 위해 사용되는 방식으로 데이터를 헤더에 추가하여 전송하는 방식이다. URL에 데이터가 노출되기 때문에 보안적으로 중요한 데이터를 포함하면 안된다.

POST는 데이터를 추가 또는 수정하기 위해 사용하는 방식으로 데이터를 바디에 추가하여 전송하는 방식이다. 데이터가 URL로 노출되지 않기 때문에 GET에 비해 비교적 안전한 방식이다.

알고리즘

정렬 알고리즘에 대한 설명을 해주세요

버블정렬

버블정렬의 인접한 두원소를 비교하여 정렬하는 알고리즘이다.마치 위로 거품이 터져 올라오듯이 정렬을 하기 때문에 버블정렬이라고 한다. 버블 정렬의 경우 오름차순으로 정렬한다고 가정하면, 1회차에는 n-1 번 순회하고, 가장 큰 값은 맨 마지막에 저장되어있기 때문에 하나의 값을 뺀 n-2만큼 2회차에 순회한다.. 이 식을 계산하면 다음과 같다

`(n-1) + (n-2) + (n-3) + ... + 2 + 1 => n(n-1)/2` 으로 O(n^2) 의 시간복잡도를 가진다.

주어진 배열 안에서 교환을 통해 정렬을 수행하기 때문에 공간복잡도의 경우 O(n)이다

선택정렬

원소가 삽입될 위치가 이미 정해져있어 어떤 원소를 그 위치에 넣을지 선택하는 알고리즘이다.삽입정렬과 달리 선택정렬은 해당 자리를 선택하고 그 자리에 올 값을 찾는 것이라고 생각하면 된다. 버블정렬과 비슷하게 주어진 범위 내에서 하나의 자리를 찾는 것이기 때문에 시간복잡도가 O(n^2)이다. 주어진 배열 안에서 연산이 이루어지기 때문에 공간복잡도는 O(n)이다.

삽입정렬

2번째 원소부터 시작하여 그 앞의 원소들과 비교하여 삽입할 위치를 지정한 뒤, 원소를 뒤로 옮겨 지정된 자리에 자료를 삽입하여 정렬하는 알고리즘이다. 최선의 경우 O(n)이라는 효율성을 가지고 있다.

역으로 정렬되었을 경우 O(n^2) 시간복잡도를 가진다. 하지만 모두 정렬이 되어있는 경우 한번씩밖에 비교를 안하기 때문에 O(n)의 시간복잡도를 가지고, 이미 정렬 되어있는 배열에 하나씩 삽입 제거를 하는 경우 최고의 정렬 알고리즘이 된다.

이것 또한 주어진 배열안에서 교환을 하기 때문에 O(n)의 공간복잡도를 가진다.

퀵정렬

분할정복 방법을 통해 주어진 배열을 정렬하는 방법이다.

배열 가운데에서 하나의 원소를 고른 뒤 pivot으로 설정한다. 피봇을 기준으로 작은 원소는 앞으로 큰 원소는 뒤로 보내는 분할의 과정을 거친다. 이러한 분할 작업을 재귀적으로 진행하여 계속 작은단위로 피봇을 설정해 배열을 둘로 나누는 작업을 계속한다.

최선의 경우 O(nlogn) 최악의 경우(배열이 이미 정렬되어있는 경우) O(n^2)의 시간복잡도를 가진다.

퀵정렬또한 주어진 배열 안에서 교환을 통해 이루어지기 때문에 공간복잡도는 O(n)이다.

병합정렬

큰 문제를 작은 문제 단위로 쪼개면서 해결해나가는 방식이다. 퀵정렬과 비슷하면서도 매우 다르다. 퀵정렬의 경우 피봇을 이용해 점점 잘개 쪼개나가지만 병합정렬의 경우 영역을 쪼갤 수 있는 만큼 쪼갠 뒤, 머지하는 과정에서 정렬을 한다.

시간복잡도는 O(nlogn)이다.