Http 완벽가이드 #3(~p122)
커넥션관리

2023.08.26

1. 다음 설명 중 틀린 것은?

1. 일단 커넥션이 맺어지면 클라이언트와 서버 간 주고받는 메세지들은 손실, 손상되거나 순서가 바뀌지 않고 안전하게 전달된다.
2. TCP는 IP패킷이라는 작은 조각을 통해 데이터를 전송한다.
3. IP패킷헤더는 발신지,목적지의 IP주소, 크기, TCP와 기타 제거 플러그를 가진다.
4. 서로 다른 두개의 TCP커넥션은 네가지 주소 구성요소값이 모두 같을수 없다.
5. 대부분의 HTTP지연은 TCP 네트워크 지연때문에 발생한다.

정답 : 3
TCP 제어 플러그는 TCP 세그먼트 헤더가 가진다.

2. 다음 설명 중 틀린 것은?

1. 커넥션 설정시간은 새로운 TCP 커넥션에서 항상 발생한다.
2. 커넥션 생성요청의 플래그는 SYN이다.
3. 크기가 작은 HTTP트랜잭션은 50% 이상의 시간을 TCP를 구성하는데 쓴다.
4. 인터넷 라우터는 과부하가 걸렸을때, 패킷을 마음대로 파기할 수 있다.
5. TCP커넥션은 시간이 지나면 자체적으로 튜닝되어, 커넥션 최대 속도를 만들어 내고, 차차 속도를 늦춘다.

정답 : 5
처음에는 최대속도를 제한하고 차차 속도제한을 높여나간다.
(TCP 느린시작)

3. 다음 설명 중 틀린 것은?

1. 일단 커넥션이 맺어지면 클라이언트와 서버 간 주고받는 메세지들은 손실, 손상되거나 순서가 바뀌지 않고 안전하게 전달된다.
2. TCP는 IP패킷이라는 작은 조각을 통해 데이터를 전송한다.
3. IP패킷헤더는 발신지,목적지의 IP주소, 크기, TCP와 기타 제거 플러그를 가진다.
4. 서로 다른 두개의 TCP커넥션은 네가지 주소 구성요소값이 모두 같을수 없다.
5. 대부분의 HTTP지연은 TCP 네트워크 지연때문에 발생한다.

정답 : 3
TCP 제어 플러그는 TCP 세그먼트 헤더가 가진다.

4. 다음은 무엇에 대한 설명인가? (주관식)

XX은 네트워크 효율을 위해서, 패킹을 전송하기 전에 많은 양의 TCP 데이터를 한개의 덩어리로 합친다. XX는 세그먼트가 최대 크기가 되지 않으면 전송을 하지 않는다. 다만 다른 모든 패킷이 확인 응답을 받았을 경우에는 최대 크기보다 작은 패킷 전송을 허락한다.

XX는 HTTP 성능과 관련해 여러 문제를 발생시킨다. 먼저 크기가 작은 HTTP 메세지는 패킷을 채우지 못하기 때문에, 앞으로 생길지 생기지 않을지 모르는 추가적인 데이터를 기다리며 지연된다.

정답 :
네이글 알고리즘

5. 다음 설명 중 맞는 것은?

1. TIME_WAIT 포트 고갈은 실제상황에서 심각한 문제를 발생시킨다.
2. TCP 종단에서 TCP커넥션을 끊으면, 커넥션의 IP주소와 포트번호를 메모리의 작은 제어영역에 기록해 둔다.
3. 2의 정보는 3MLS의 시간동안만 유지된다.
4. 3의 시간을 더 짧은 시간으로 수정해도 문제없다.
5. HTTP Connection 헤더 필드는 커넥션 토큰을 쉼표로 구분하여 가지고 있으며, 그 값은 다른 커넥션으로 전달된다.

정답 : 2
1. 성능측정시 심각하게 나오지만, 보통실제는 문제없음
3. 2MLS
4. 조심해야 한다. TCP데이터 충돌이 나타날수 있다.
5. 다른 커넥션으로 전달되지 않는다.

6. 다음 설명 중 틀린 것은?

1. hop by hop은 특정 두 서버 간에만 영향을 미치고 다른 서버간에는 영향을 미치지 않음을 의미한다.
2. Proxy-Athenticate, Proxy-Connection 등은 Connection 헤더에 기술되어 있다.
3. Connection 헤더는 전송자가 특정 커넥션에만 해당되는 옵션을 지정하게 해준다.
4. 파이프라인 커넥션, 다중 커넥션은 HTTP 커넥션 성능을 향상시킬수 있다.
5. 병렬 커넥션이 항상 더 빠르지는 않다.

정답 : 2
2. 위와 같은 것들은 Connection헤더에는 기술되어 있지않다.

7. 다음 설명 중 맞는 것은?

1. 브라우저는 실제로 병렬 커넥션을 사용하지만 대부분 적은수(대부분 6개)의 병렬 커넥션만 사용한다.
2. 서버에 HTTP 요청을 시작한 애플리케이션이 또 다른 리소스 요청을 하기 위해 재차 요청하는 것을 site locality라고 부른다.
3. 병렬 커넥션은 지속 커넥션에 비해 커넥션 수를 줄인다는 장점이 있다.
4. 지속커넥션은 파이프라인 커넥션과 함께 사용될때 가장 효과적이다.
5. HTTP/1.0+ 에는 지속커넥션이 있고, HTTP/1.1에는 Keep-alive 커넥션이 있다.

정답 : 2
1. 4개
3. 지속커넥션이 병렬 커넥션에 비해
4. 파이프라인이 아니라 병렬 커넥션
5. 둘이 바뀜

8. 다음 설명 중 틀린 것은?

1. keep-alive 요청을 받았다고 해서 무조건 따를 필요는 없다.
2. keep-alive는 HTTP/1.0 에서 기본으로 사용되지는 않는다.
3. 기술적으로 HTTP/1.0을 따르는 기기로 부터 받은 모든 Connection 헤더 필드는 무시해야 한다.
4. 넷스케이프는 dumb proxy문제 해결을 위해 connection헤더 대신 Proxy-connection을 표준으로 채택했다.
5. 프락시가 많은 구조에서는 Proxy-connection을 사용해도 여전히 문제가 생길수 있다.

정답 : 4
3. (오래된 프락시 서버에서 실수로 전달될수도 있어서)
4. 비표준이다.

9. 다음 설명 중 틀린 것은?

1. HTTP 클라이언트는 커넥션이 지속커넥션인지 확인 전까지는 파이프라인을 이어서는 안된다.
2. HTTP 클라이언트는 POST와 같은 멱등요청을 파이프라인으로 재차 보내면 문제가 생길수 있으므로, 그런 요청을 보내서는 안된다.
3. 어떠한 HTTP클라이언트, 서버, 프락시는 언제든지 TCP전송 커넥션을 끊을수 있다.
4. TCP 커넥션을 개별적으로 끊으려면 shutdown()을 호출하면 된다.
5. 연결이 끊긴 커넥션에 데이터가 도착하게 되면 ‘connection reset by peer’에러가 발생한다.

정답 : 2
2. 비멱등