

2장 — 개략적인 규모 추정

요구사항에 부합하는 설계를 위해서는 성능 요구사항을 개략적으로 추정할 수 있어야 한다.

2의 제곱수

2의 x 제곱	근사치	이름
10	1천	1KB
20	1백만	1MB
30	10억	1GB
40	1조	1TB
50	1000조	1PB

데이터 볼륨의 단위를 2의 제곱수로 표현할 수 있어야 기본적인 계산이 가능하다.

프로그래머가 알아야 하는 응답지연 값



https://colin-scott.github.io/personal_website/research/interactive_latency.html

연산명	시간
L1 캐시 참조	0.5ns
분기 예측 오류(branch mispredict)	5ns
L2 캐시 참조	7ns
뮤텍스(mutex) 락/언락	100ns
주 메모리 참조	100ns
Zippy로 1 KB 압축	10,000ns = 10μs
1 Gbps 네트워크로 2 KB 전송	20,000ns = 20μs
메모리에서 1 MB 순차적으로 read	250,000ns = 250μs
같은 데이터 센터 내에서의 메시지 왕복 지연시간	500,000ns = 500μs
디스크 탐색(seek)	10,000,000ns = 10ms
네트워크에서 1 MB 순차적으로 read	10,000,000ns = 10ms
디스크에서 1 MB 순차적으로 read	30,000,000ns = 30ms
한 패킷의 CA(캘리포니아)로부터 네덜란드까지의 왕복 지연시간	150,000,000ns = 150ms

ns = nanosecond(나노초), μs = microsecond(마이크로초), ms = millisecond(밀리초)

1나노초 = 10^{-9} 초

1마이크로초 = 10^{-6} 초 = 1,000나노초

1밀리초 = 10^{-3} 초 = 1,000μs = 1,000,000ns

- 메모리는 빠르지만 디스크는 느리다
- 디스크 탐색은 가능한 피하라
- 단순한 압축 알고리즘은 빠르다
- 데이터를 인터넷에 보내기 전에, 최대한 압축하라
- 데이터 센터는 여러 지역에 분산되어 있기 때문에, 데이터를 주고 받는 비용은 비싸다

가용성

- 고가용성
 - 시스템이 오랜 시간 지속적으로 중단 없이 운영될 수 있는 능력
- 가용성 수치

가용률	하루당 장애시간	주당 장애시간	개월당 장애시간	연간 장애시간
99%	14.40분	1.68시간	7.31시간	3.65일
99.9%	1.44분	10.08분	43.83분	8.77시간
99.99%	8.64초	1.01분	4.38분	52.60분
99.999%	864.00밀리초	6.05초	26.30초	5.26분
99.9999%	86.40밀리초	604.80밀리초	2.63초	31.56초

예제 : 트위터 QPS와 저장소 요구량 추정

가정

- MAU : 3억
- 50% 사용자는 매일 트위터 사용한다
- 사용자는 매일 평균 2건의 트윗을 올림
- 미디어 포함 트윗은 10%
- 데이터는 5년 보관

추정

- DAU : 3억 * 50% → 1.5억
- QPS : 1.5억 * 2트윗 / 24시간 / 3600초 = 약 3,500
- 최대 QPS : 2 * QPS = 약 7,000
- 미디어 저장을 위한 저장소 요구량
 - 하루 저장소 요구량 * 365일 * 5년
 - 평균 트윗 크기
 - tweet_id : 64Byte
 - text : 140Byte
 - media : 1MB
 - 1.5억 * 2 (하루 평균 트윗 수) * 10% * 1MB = 30TB/일

- $30TB * 365 * 5 = \text{약 } 55PB$

팁

- 결과보단 올바른 절차가 중요하다.
- 근사치를 활용해 계산하자
- 가정을 적어두자, 나중에 살펴볼 수 있도록
- 단위를 붙이자
- 많이 출제되는 문제를 미리 연습해보자 : QPS, 최대 QPS, 저장소 요구량, 캐시 요구량, 서버 수

3장 — 시스템 설계 면접 공략법

설계 기술을 시연하고, 설계 과정에서 내린 결정들에 대한 방어 능력을 보이는 면접이다.

효과적 면접을 위한 4단계 접근법

1. 문제 이해 및 설계 범위 확정
2. 개략적인 설계안 제시 및 동의 구하기
3. 상세 설계
4. 마무리

문제 이해 및 설계 범위 확정

- 답을 바로 내지 마라
 - 퀴즈쇼가 아니고, 정답 또한 없다.
- 깊이 생각하고 질문하고, 적절한 가정을 하고, 시스템 구축에 필요한 정보를 모으자
- 예시 질문 리스트
 - 구체적으로 어떤 기능을 만들어야하는가?

- 제품 사용자 수는 얼마나 되는가?
- 회사의 규모는 얼마나 빠르게 커지리라 예상하는가?
- 회사의 주 기술 스택은 무엇인가?

개략적인 설계안 제시 및 동의 구하기

- 설계안의 청사진을 제시하고 면접관들의 의견을 구하라
- 화이트보드, 종이에 핵심 컴포넌트를 포함하는 다이어그램을 그려라
- 설계안이 시스템 규모에 관계된 제약사항들을 만족하는지 계산해보아라

상세 설계

- 면접관과 함께 설계 대상 컴포넌트 사이의 우선순위를 정하라
- 시간 관리를 잘하자
 - 지나치게 사소한 세부사항을 긴 시간 설명하는 것은 좋지 않다
 - 확장 가능한 시스템을 설계할 능력이 있는 것을 입증하라

마무리

- 개선할 점은 언제나 있기 마련이다. 개선점을 찾고 비판적 사고 능력을 보여라
- 설계를 요약해주는 것도 도움이 될 수 있다
- 오류가 발생한다면 무슨 일이 생기는지 따져보면 흥미로울 것이다
- 운영 이슈도 논의할 가치가 충분하다
- 미래에 닥칠 규모 확장에 어떻게 대처할 것인지 고려해보자
- 시간이 남는다면 세부적인 사항들도 제안

해야 할 것

- 질문을 통해 확인하라, 절대 스스로 가정하고 진행하지 말자
- 문제의 요구사항을 정확히 이해하라

- 설계의 정답, 최선의 답안은 없다는 것을 명심하자
 - 요구사항을 규모와 함께 고려하자
- 사고의 흐름을 이해할 수 있게 보여라, 소통하라
- 가능한 여러 해법을 제시하라
- 개략적 설계에 면접관이 동의하면, 각 컴포넌트의 세부사항을 설명하기 시작하라
 - 가장 중요한 컴포넌트 부터 시작하자
- 면접관의 아이디어, 힌트를 이끌어내자 → 팀원처럼 협력하게 만들어라
- **포기하지 말라**

하지 말아야 할 것

- 전형적인 문제에 대비하지 않고 면접장에 가지 말자
- 요구사항, 가정을 분명히 하지 않은 상황에서 설계를 제시하지 말자
- 처음부터 특정 컴포넌트를 너무 깊이 파고들지 말자
- 진행 중 막혔을 때 힌트를 청하기를 주저하지 말라
- 소통을 주저하지 말라
- 설계안을 완성한다고 면접이 끝난 것이 아니다. 의견을 일찍, 자주 구해라