I/O process bound
CPU process bound

화생방 스터디
박희원

목차

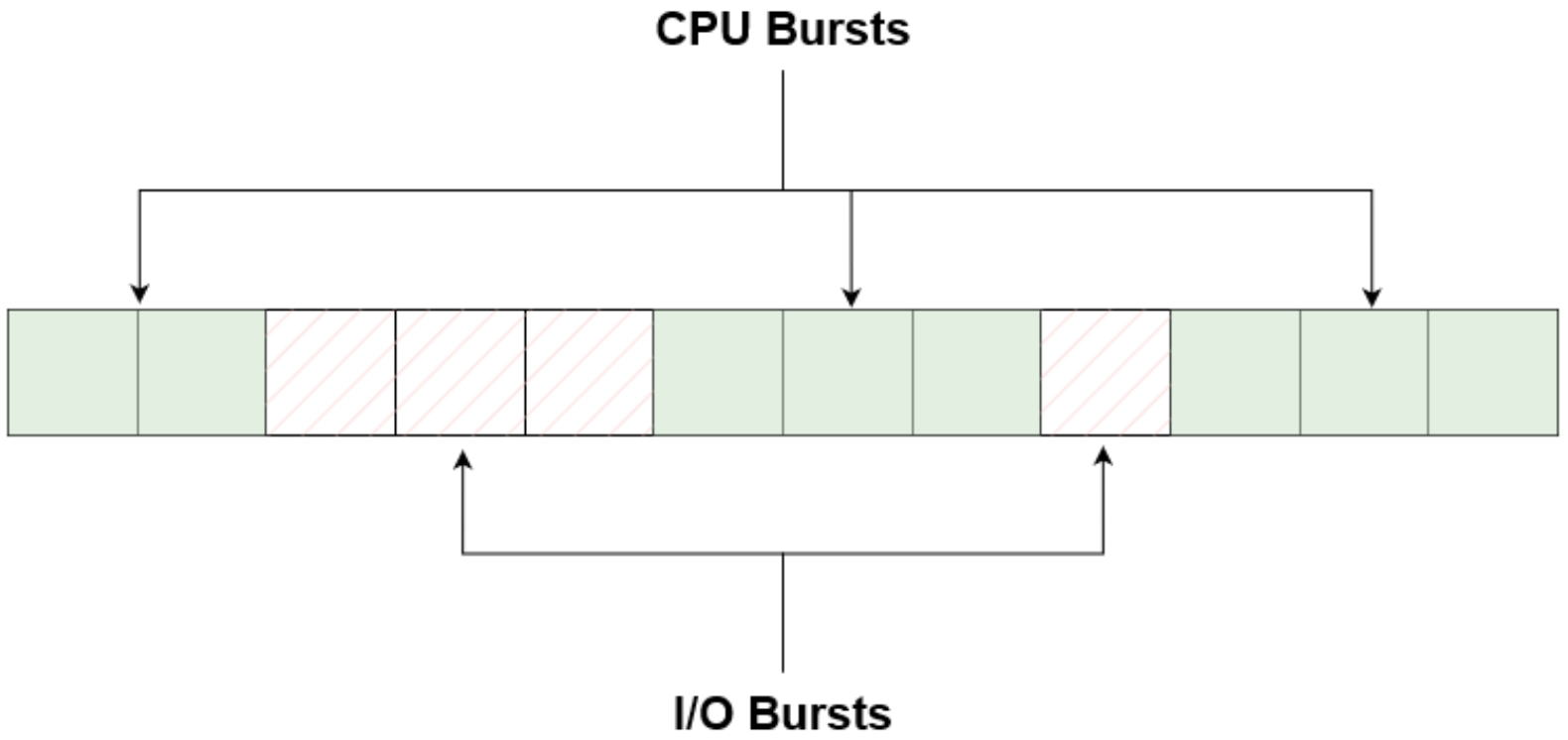
CPU Burst / IO Burst

✔ Burst
- 어떤 현상이 짧은 시간 안에 집중적으로 일어나는 일
- 계속되는 작업

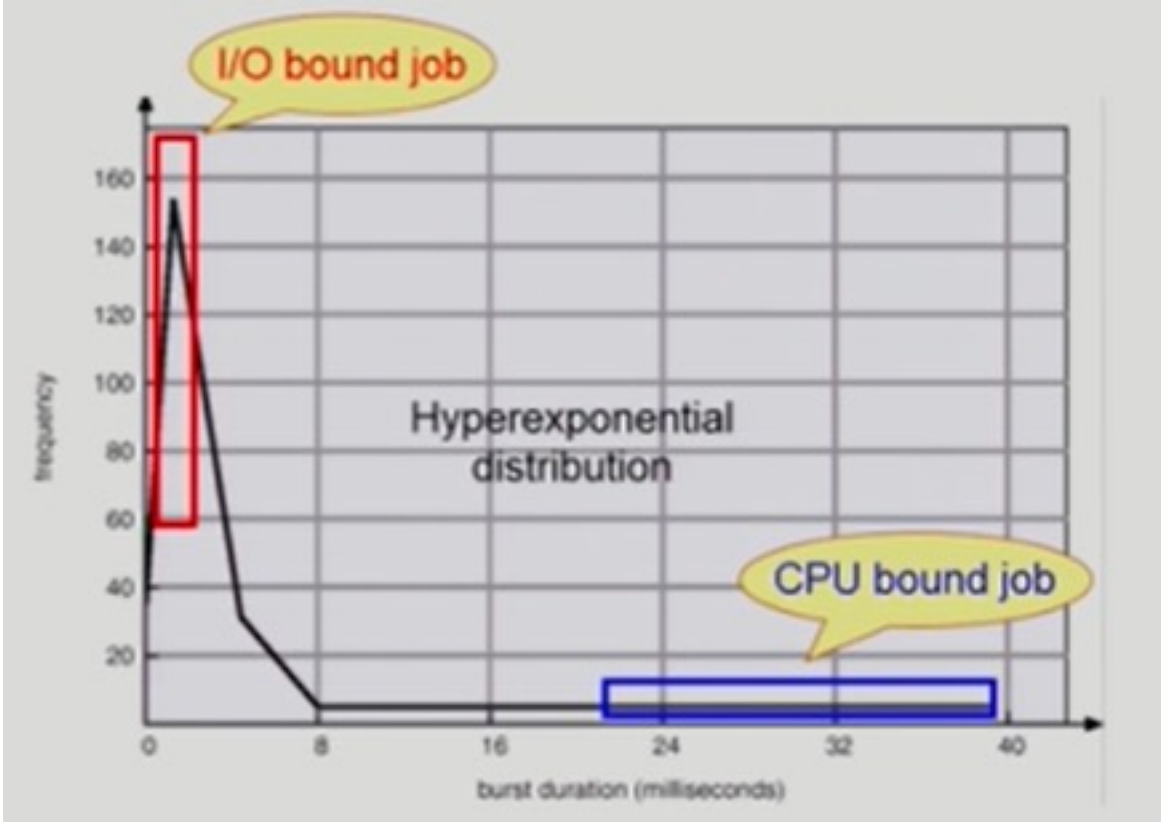
✔ CPU Burst
- 프로세스가 **CPU**에서 **한 번에 연속적으로 실행**되는 시간
- 메모리에 올라가 있는 프로세스가, 자신의 차례가 되어 CPU에서 실행되었을 때, 자신의 명령어가 CPU에서 연속적으로 실행되는 시간

✔ IO Burst
- 프로세스가 **IO 작업**을 **요청하고 결과를 기다리는** 시간

CPU Burst / IO Burst



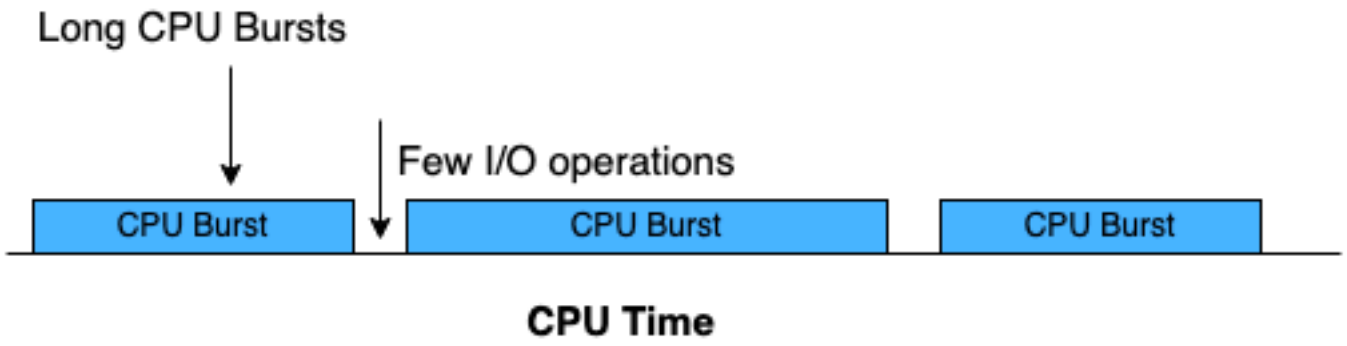
CPU Bound / IO Bound



CPU Bound / IO Bound

✔ CPU Bound 프로세스

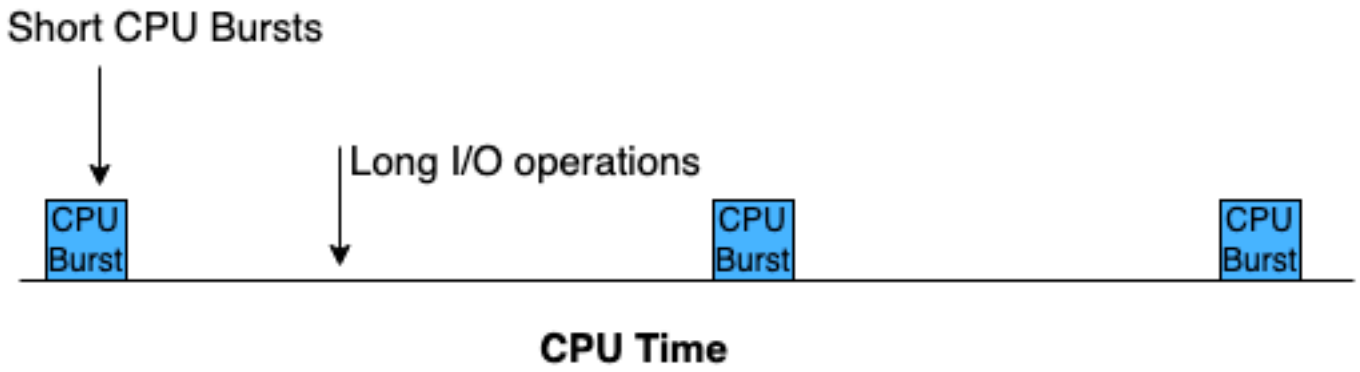
- CPU Burst가 많은 프로세스 = CPU를 많이 사용하는 프로세스
- 예) 동영상 편집 프로그램, 머신 러닝 프로그램(Applications that usually require tons of calculations)



CPU Bound / IO Bound

✔ IO Bound 프로세스

- IO Burst가 많은 프로세스 = IO를 많이 사용하는 프로세스
- 예) 백엔드 API 서버 => DB 서버, 캐시 서버에 요청을 보내는 경우가 많음



적절한 스레드의 개수는?

✔ 듀얼 코어 CPU에서 동작할 CPU Bound 프로그램을 구현한다면 몇 개의 스레드를 쓰는게 좋을까?

- **number of CPUs + 1**

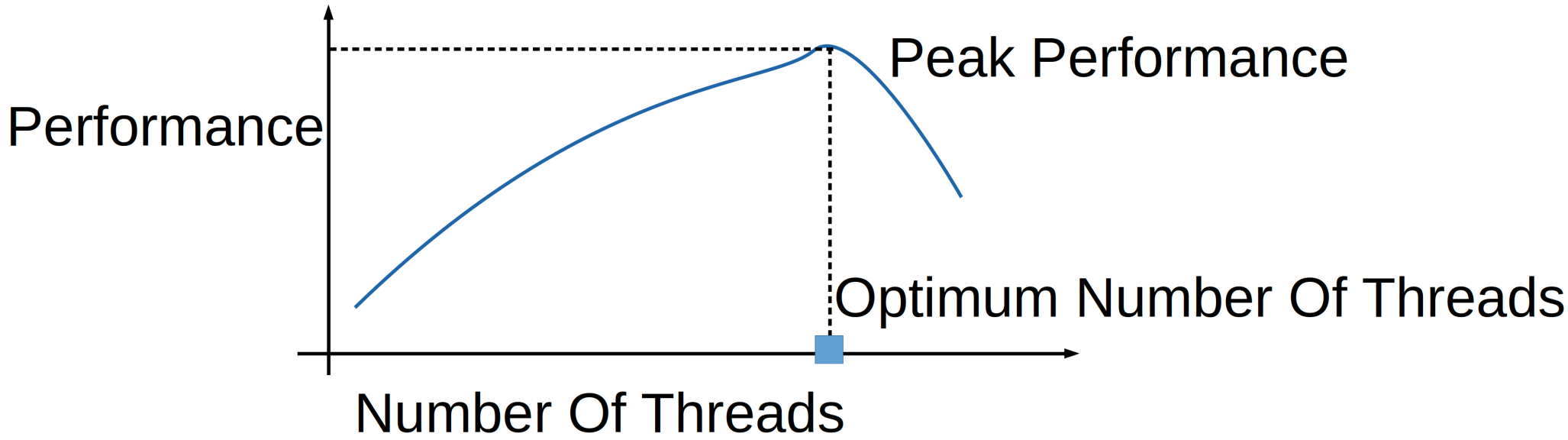
- 스레드가 너무 많으면, 스레드끼리 코어를 사용하기 위해 경합을 하면서, 짧은 시간 단위로 컨텍스트 스위칭 발생 ⇒ 오버헤드가 발생
- 코어의 개수만큼, 아니면 코어 개수에서 크게 벗어나지 않는 선에서(1개로 제안함) 스레드 개수를 설정하는 것이 효율적이다.

✔ 듀얼 코어 CPU에서 동작할 IO Bound 프로그램을 구현한다면 몇 개의 스레드를 쓰는게 좋을까?

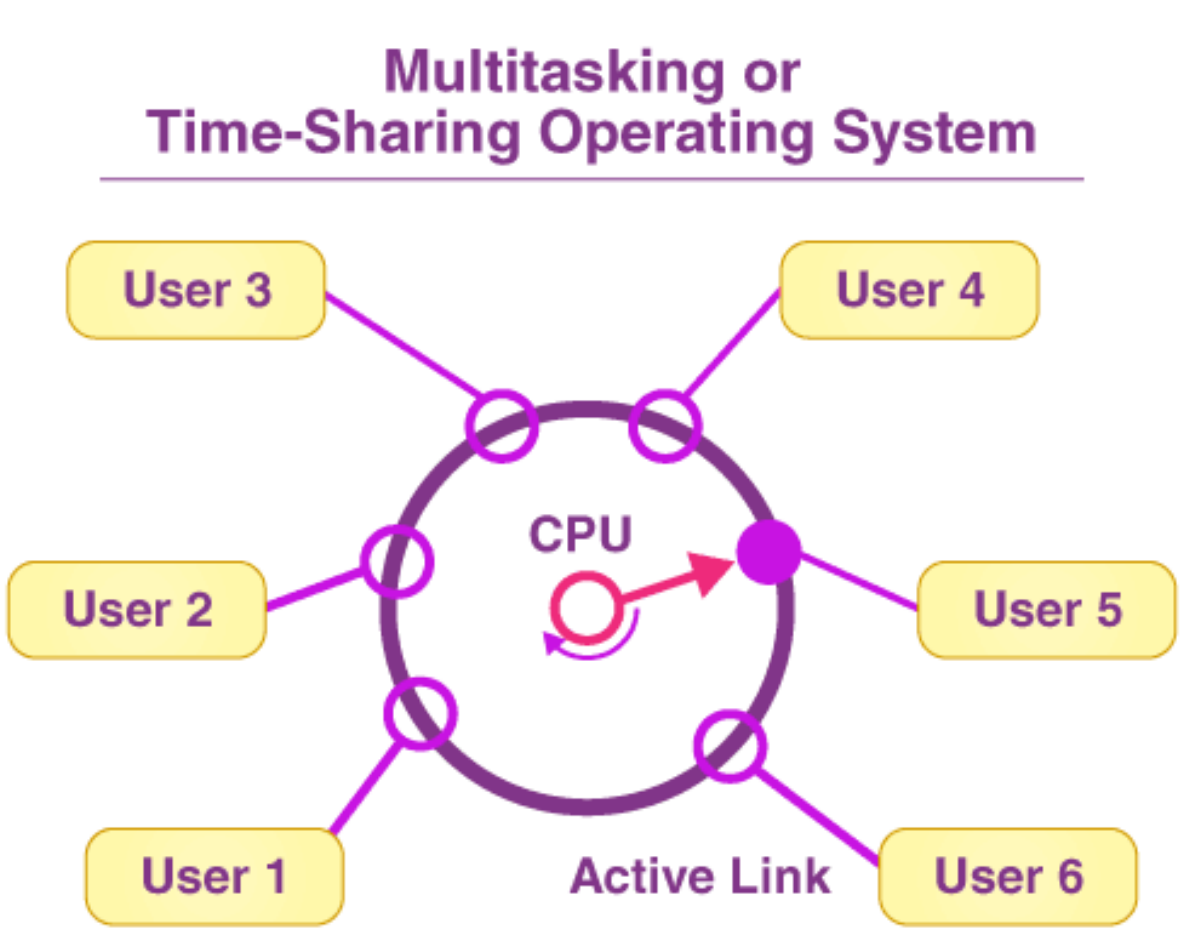
- 정해진 가이드라인은 없으므로, 여러 상황에 맞춰 적절한 스레드 개수를 찾아야 한다.
- 먼저 애플리케이션을 실행하면서, 경험적으로 최적의 스레드 개수를 찾아가는 것도 방법이다.

적절한 스레드의 개수는?

- 무조건 스레드가 많다고 해서 (물론 동시에 실행되는 것처럼 보이겠지만) 성능이 좋다는 건 아님
- 오히려 컨텍스트 스위칭이라는 오버헤드가 발생해서, 성능은 저하될 가능성이 크다.



Time Sharing System



Time Sharing System

✔ 시분할 시스템(Time Sharing System)이란?

- 운영체제가 제공하는 주요기능
- CPU가 더 효율적으로 사용될 수 있도록 여러 작업을 번갈아가며 실행하는 방식
- 여러 프로세스가 사용하는 시스템에서 컴퓨터가 자원을 시간적으로 분할해주어 사용자들의 프로그램을 번갈아가며 처리해줌으로써 각 프로세스에게 독립된 컴퓨터를 사용하는 느낌을 주는 것

Time Sharing System

✔ 시분할 시스템(Time Sharing System)의 특징

- 여러 사용자가 각자의 단말장치를 통하여 동시에 운영체제와 대화하면서 각자의 프로그램을 실행
- 하나의 CPU는 같은 시점에서 여러 개의 작업을 동시에 실행할 수 없기 때문에 CPU의 전체 사용시간을 작은 작업 시간량으로 쪼개어 그 시간량 동안만 번갈아가며 CPU사용이 할당되어 각 작업을 처리
- 다중 프로그래밍 방식과 결합하여 모든 작업이 동시에 진행되는 것처럼 대화식 처리가 가능
- 시스템의 전체 효율은 좋아지나 개인별 사용자 입장에서는 반응 속도가 느려질 수 있다
- 각 작업에 대한 응답 시간을 최소한으로 줄이는 것을 목표, 하드웨어를 보다 능률적으로 사용 가능

Q & A