SAMSUNG SDS

Foresee

Techtonic 2021

Partner

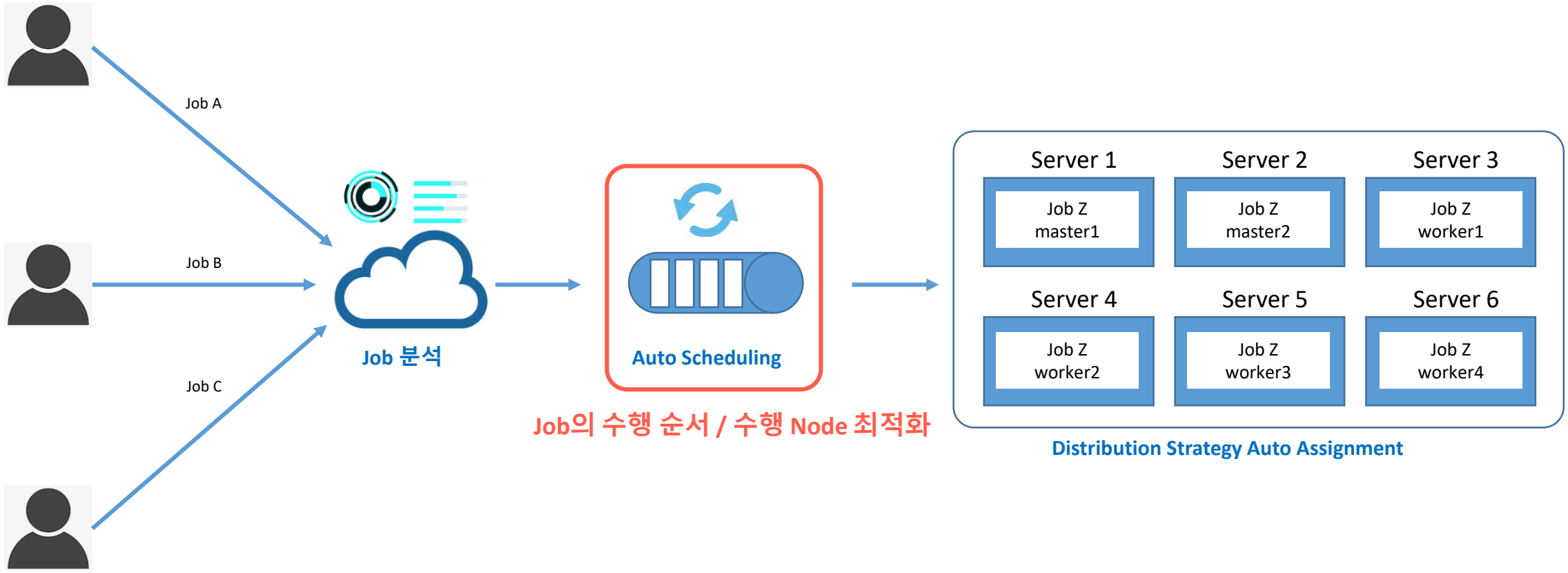
Disrupt

GPU 놀지마!
작업시간 예측을 통한 스케줄링 방법

이창주 프로

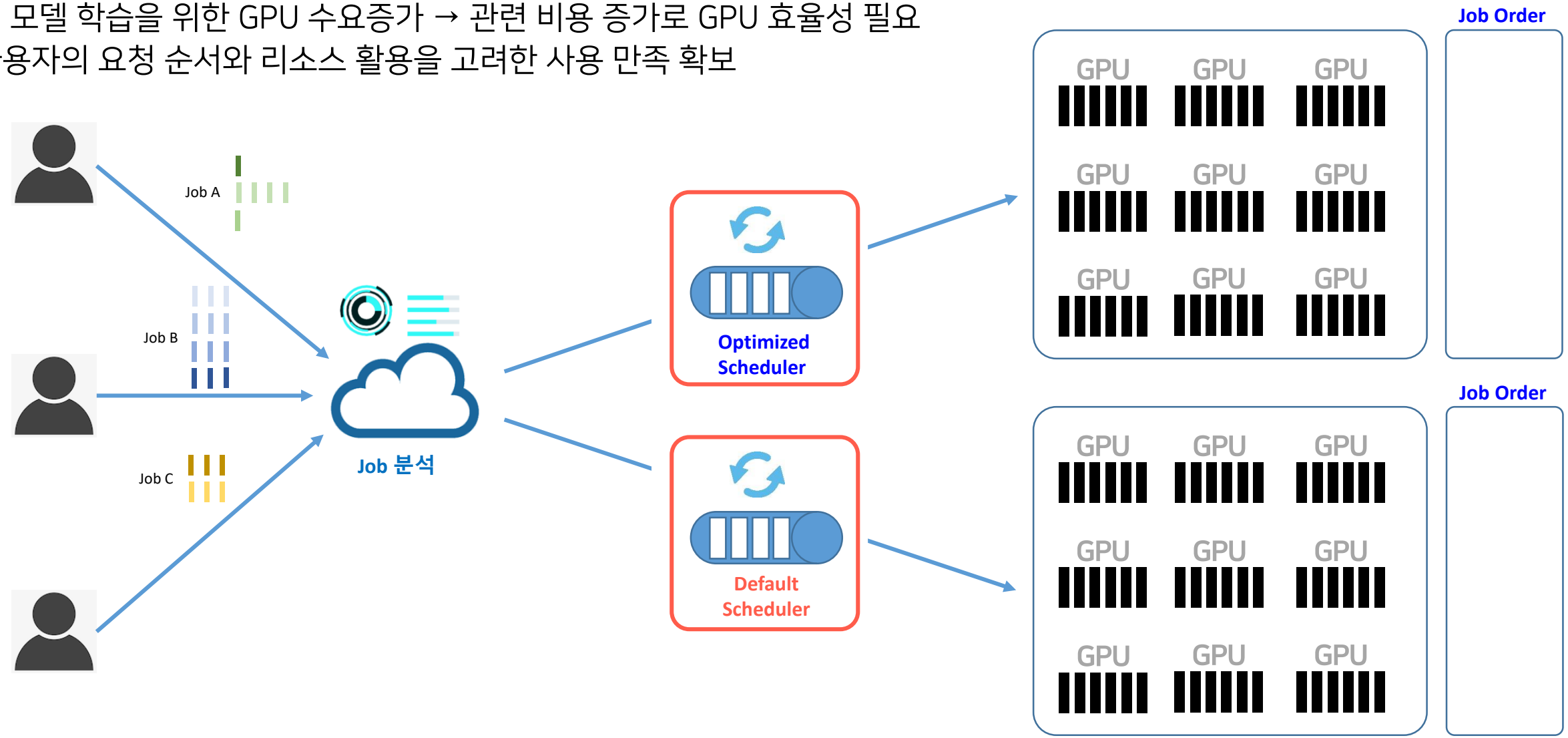
클라우드에서 스케줄러는 무엇인가?

ML/DL Job의 특성을 파악하여 최적의 자원을 분배하는 기술



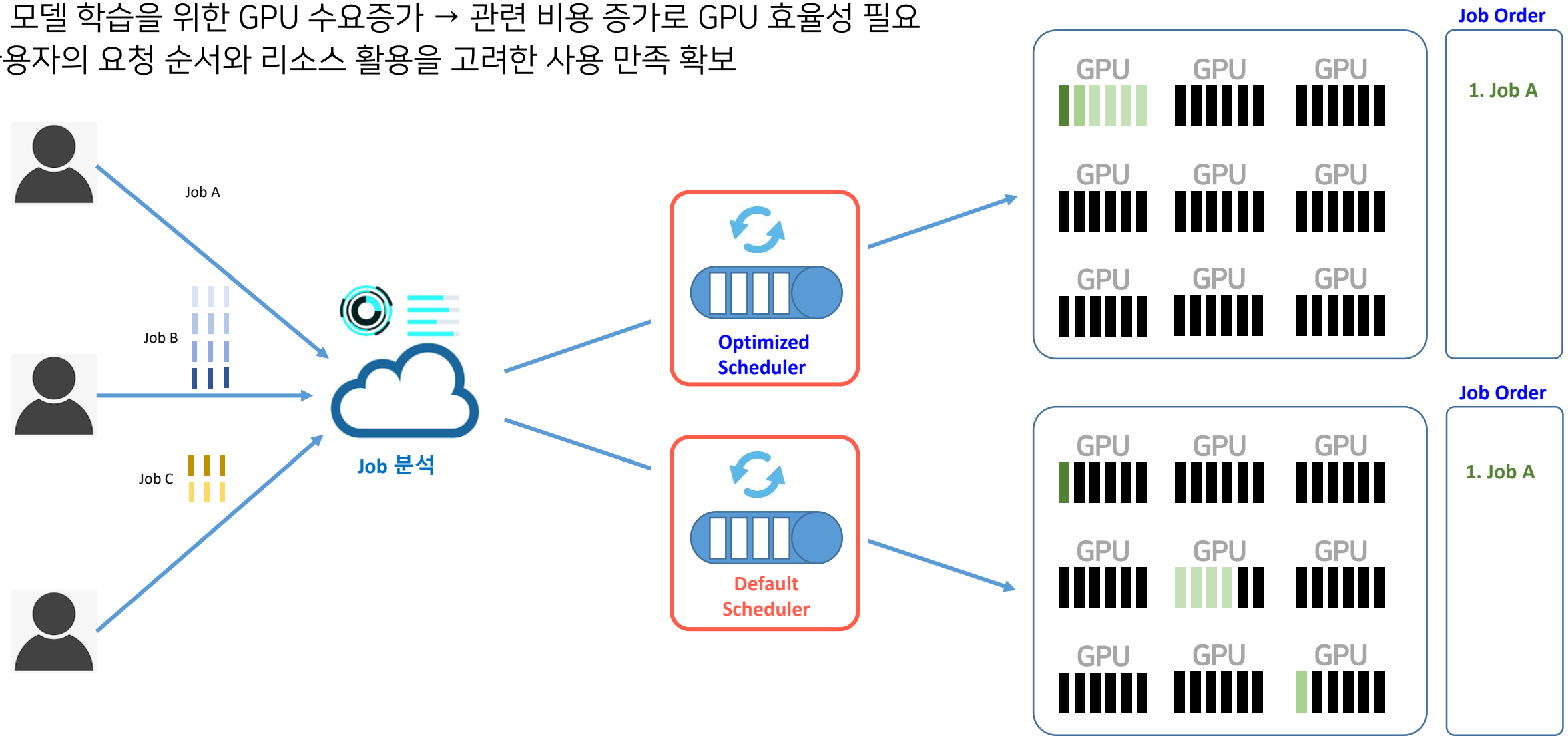
ML/DL에 적합한 스케줄러는 어떤 모습일까?

AI 모델 학습을 위한 GPU 수요증가 → 관련 비용 증가로 GPU 효율성 필요
사용자의 요청 순서와 리소스 활용을 고려한 사용 만족 확보



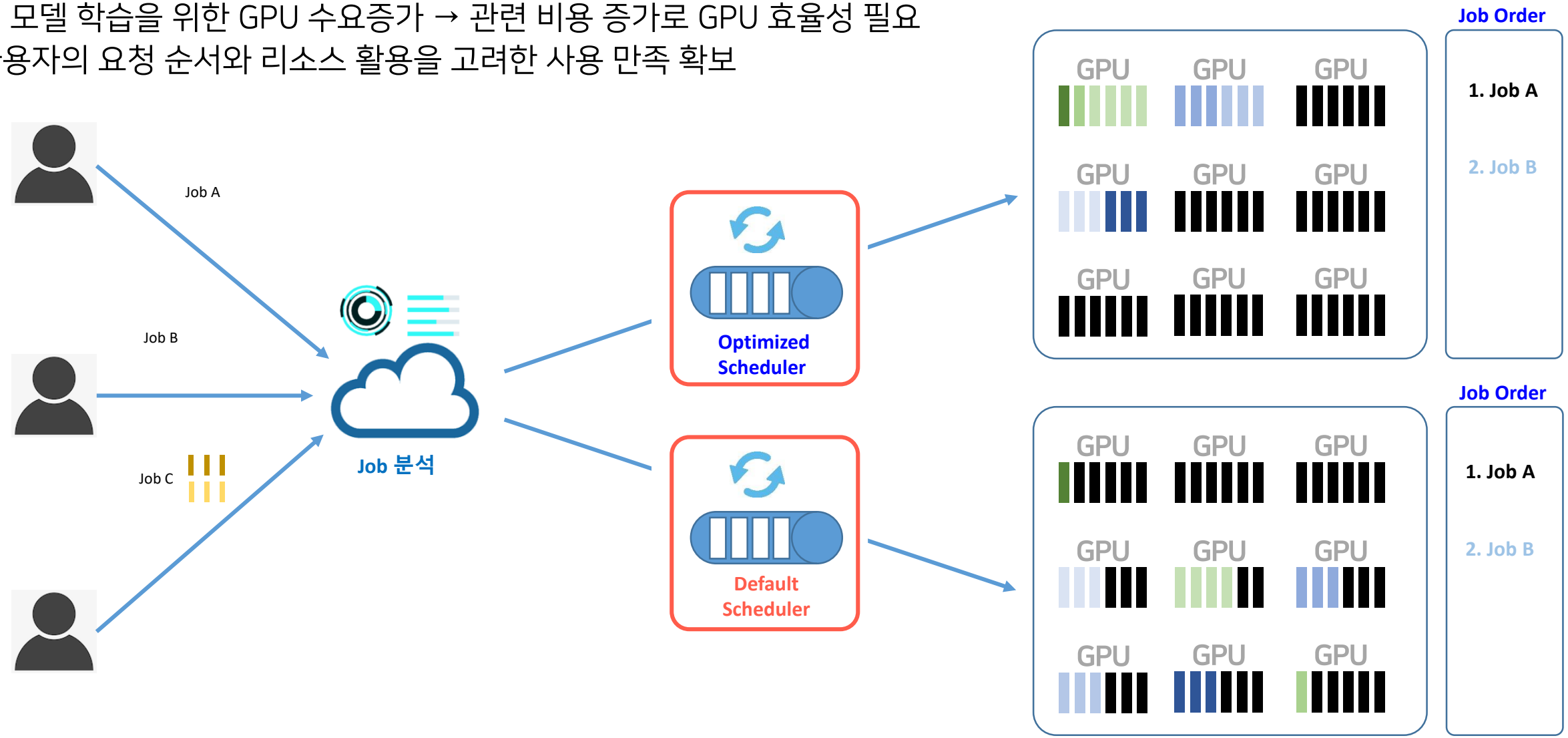
ML/DL에 적합한 스케줄러는 어떤 모습일까?

AI 모델 학습을 위한 GPU 수요증가 → 관련 비용 증가로 GPU 효율성 필요
사용자의 요청 순서와 리소스 활용을 고려한 사용 만족 확보



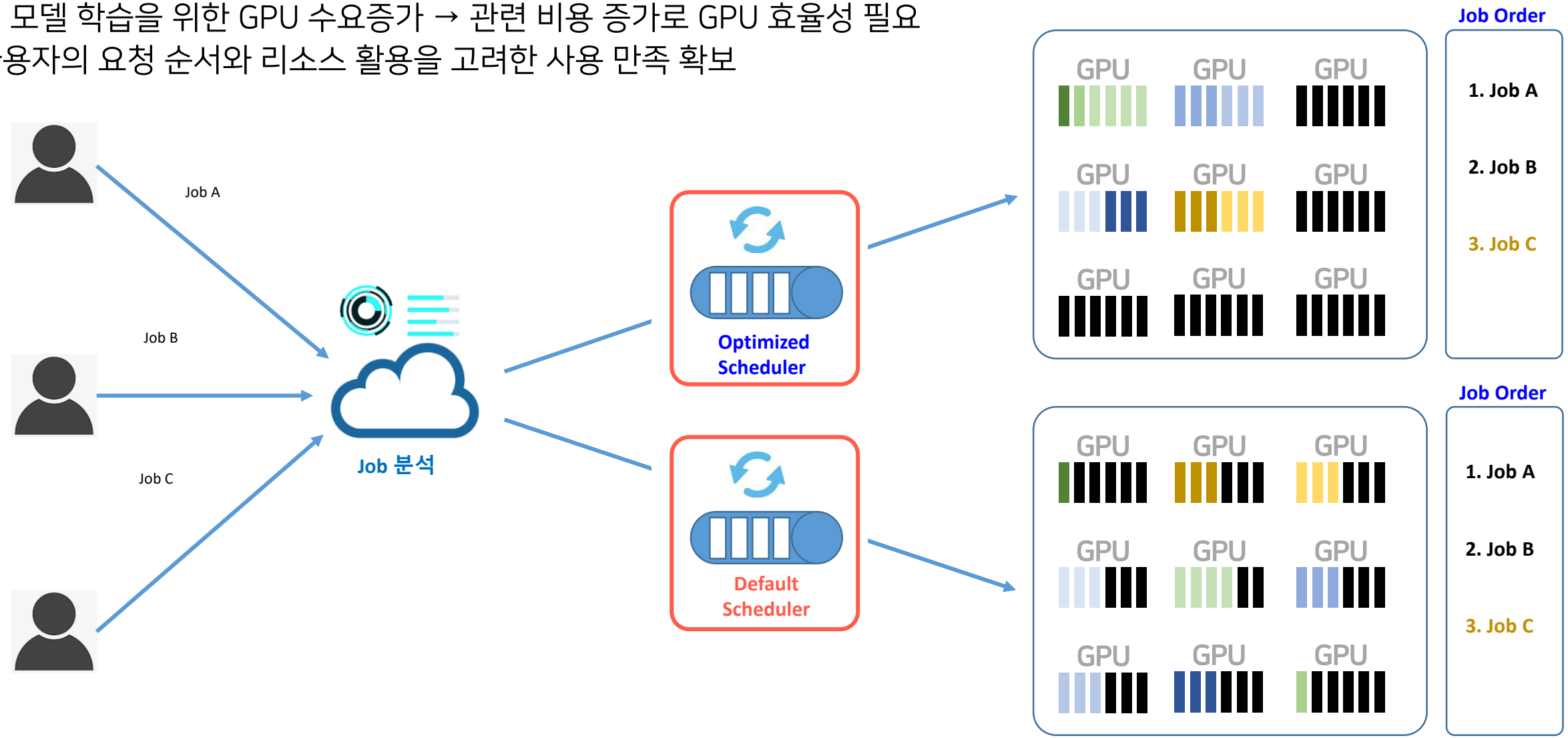
ML/DL에 적합한 스케줄러는 어떤 모습일까?

AI 모델 학습을 위한 GPU 수요증가 → 관련 비용 증가로 GPU 효율성 필요
사용자의 요청 순서와 리소스 활용을 고려한 사용 만족 확보



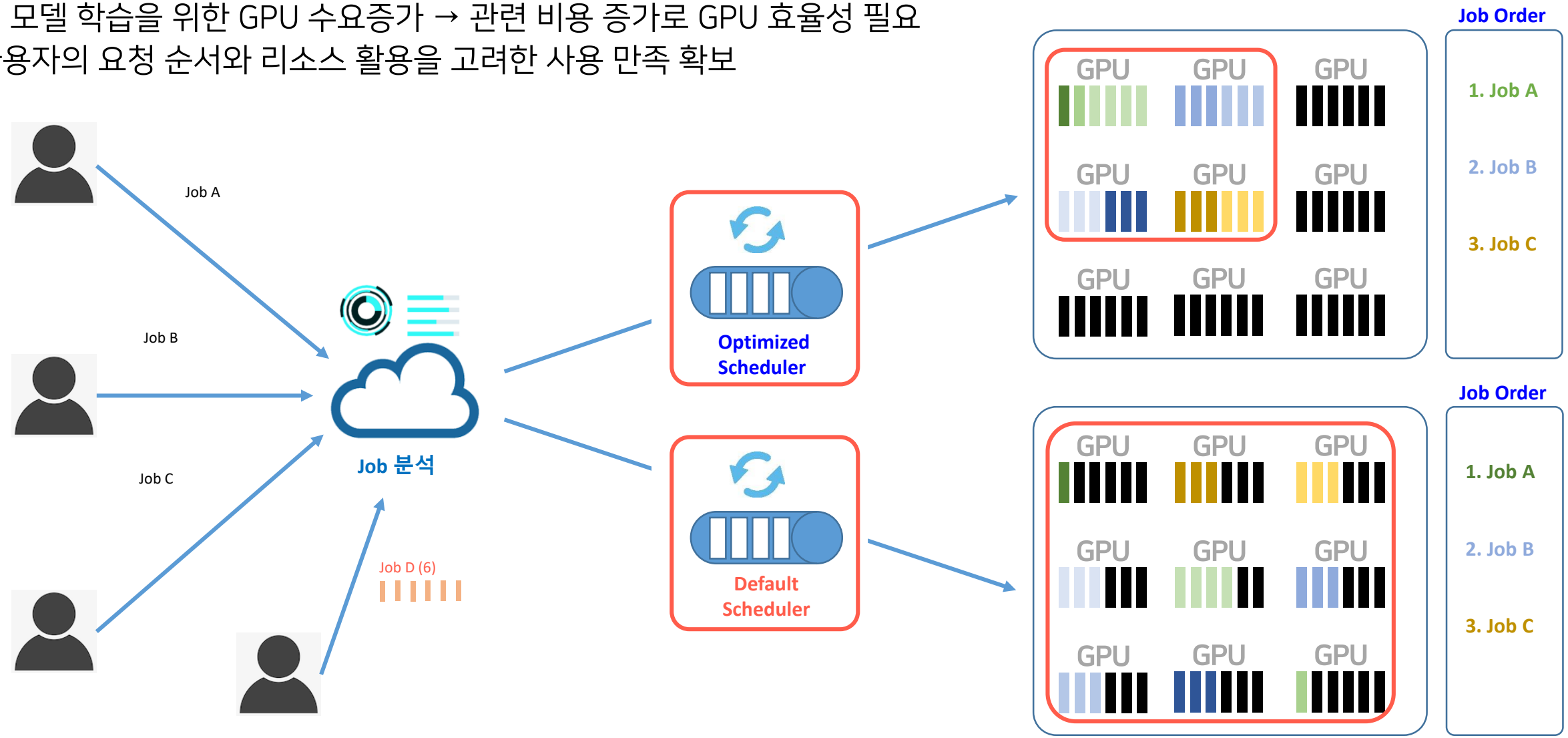
ML/DL에 적합한 스케줄러는 어떤 모습일까?

AI 모델 학습을 위한 GPU 수요증가 → 관련 비용 증가로 GPU 효율성 필요
사용자의 요청 순서와 리소스 활용을 고려한 사용 만족 확보



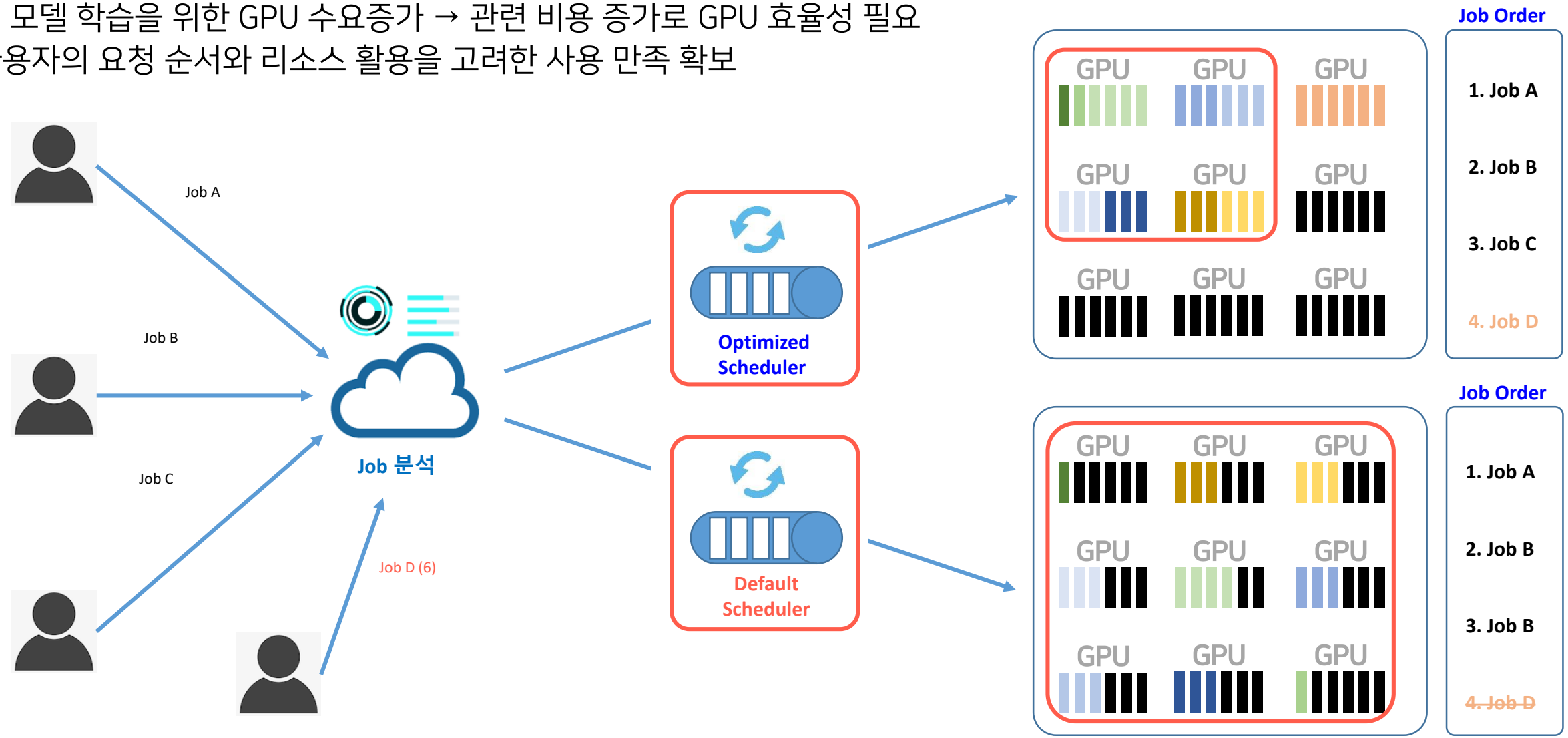
ML/DL에 적합한 스케줄러는 어떤 모습일까?

AI 모델 학습을 위한 GPU 수요증가 → 관련 비용 증가로 GPU 효율성 필요
사용자의 요청 순서와 리소스 활용을 고려한 사용 만족 확보



ML/DL에 적합한 스케줄러는 어떤 모습일까?

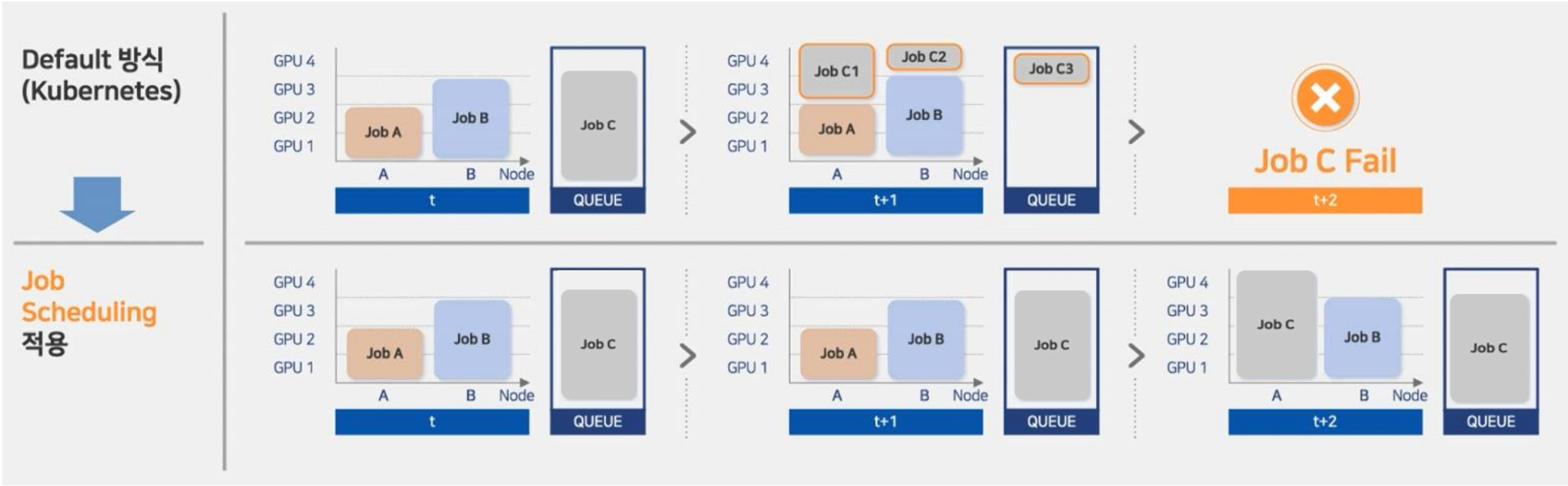
AI 모델 학습을 위한 GPU 수요증가 → 관련 비용 증가로 GPU 효율성 필요
사용자의 요청 순서와 리소스 활용을 고려한 사용 만족 확보



연구소 적용 스케줄러
(Gang, Binpacking, FIFO, Multi-Queue)

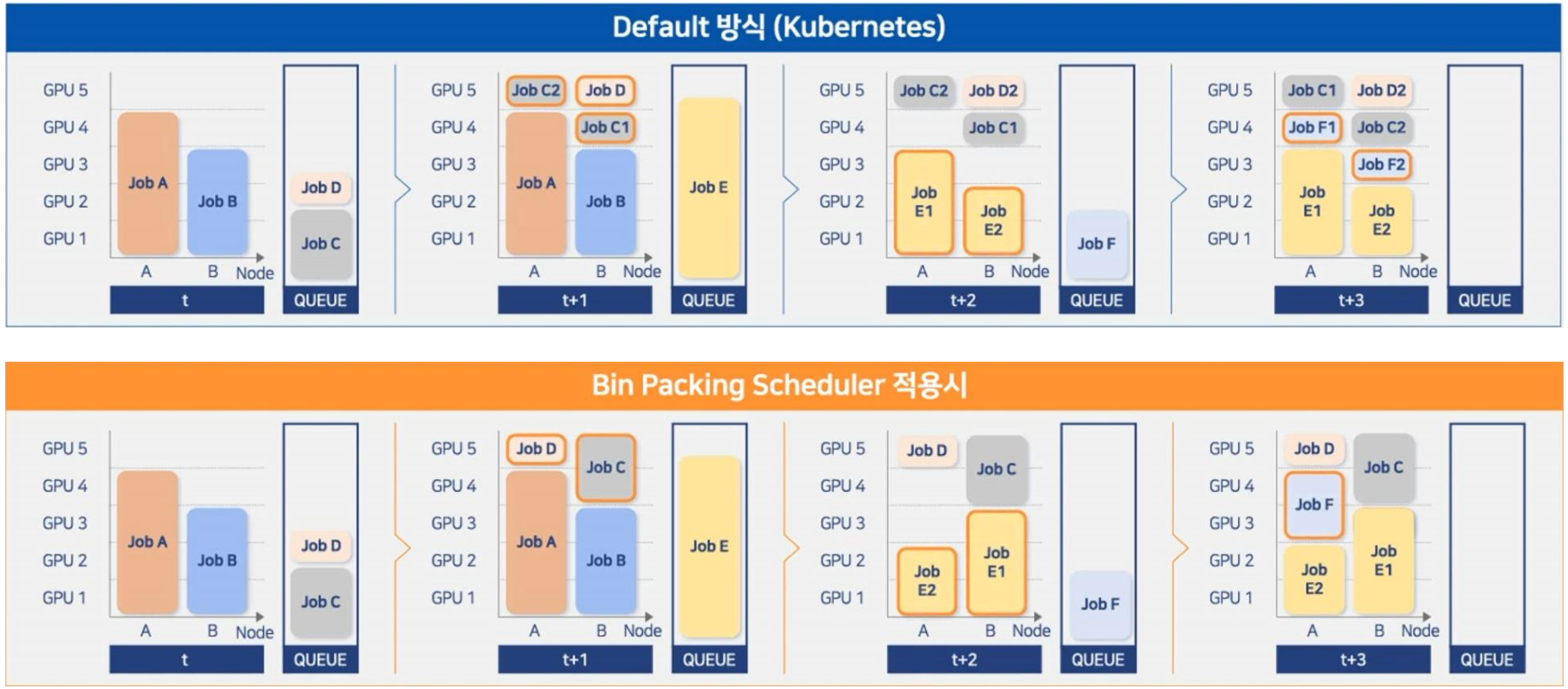
Gang

단일 ML/DL Job 수행에 필요한 resource(GPU, CPU, Memory등)가 모두 확보되었을 때 scheduling을 수행
분산 학습 Job 을 구성하는 sub Job 들의 동시 실행 시작을 보장



Bin-Packing

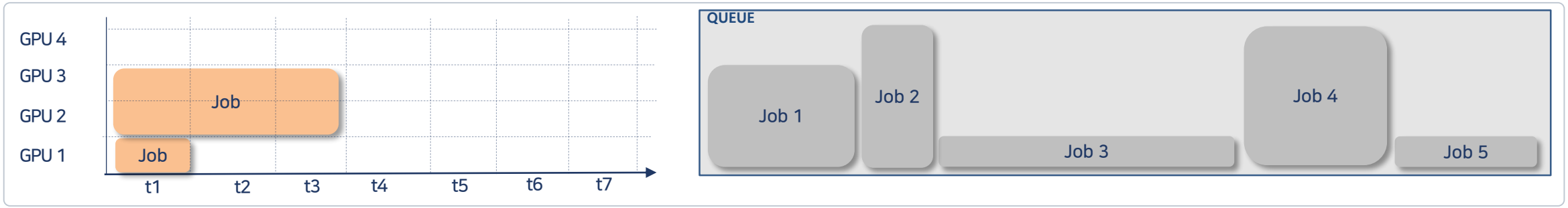
Job 에서 요청된 GPU 개수와 가장 근접한 가용 GPU를 확보하고 있는 node를 우선 배정하는 방식
유휴 자원 최소화, node인접성 증가로 인한 분산학습 성능 개선



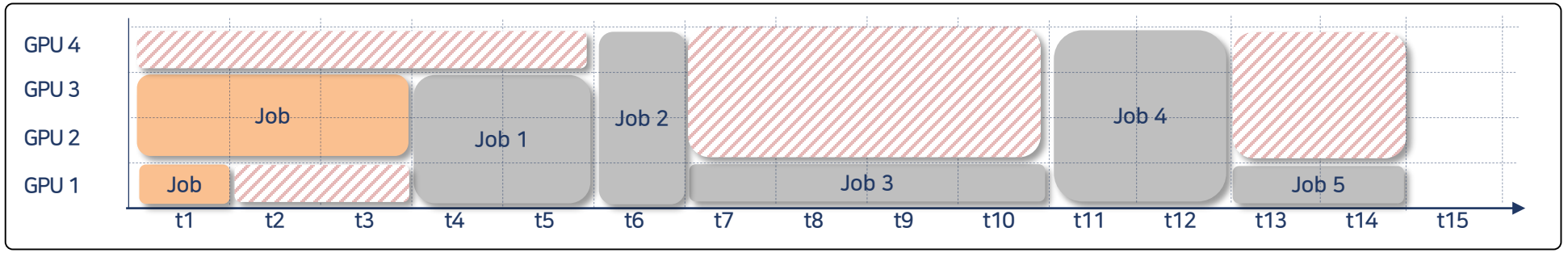
FIFO (First In First Out)

Job이 요청된 순서로 실행되는 방법

< Initial >



< FIFO >



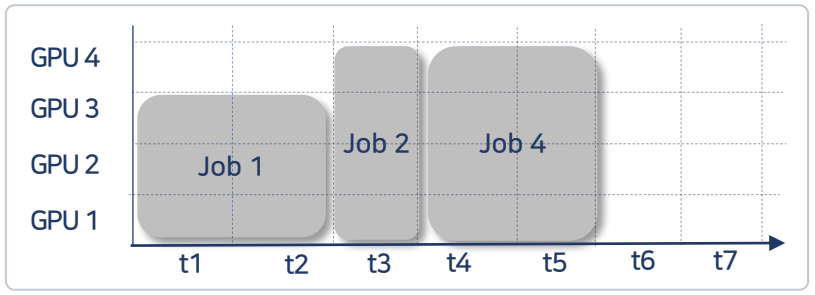
$$\text{GPU 사용율} = \frac{\text{사용된 } GPU * \text{시간}}{\text{가용 } GPU * \text{시간}} = 48.4\%$$

Multi Queue 스케줄러

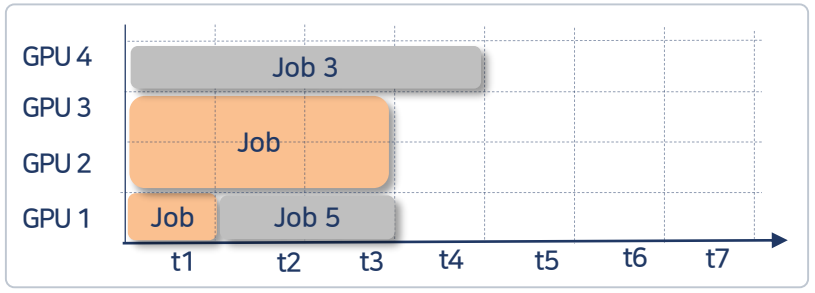
Queue를 추가로 만들어 요청 GPU가 많은 Job과 적은 Job을 따로 스케줄링 하는 방법

요청 GPU가 많은 Job에 의한 병목 해소

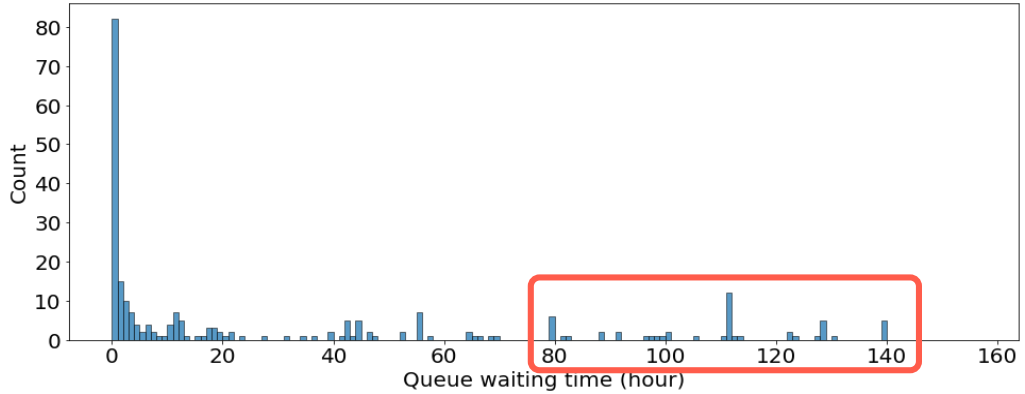
<High> * requested GPU 3개 이상



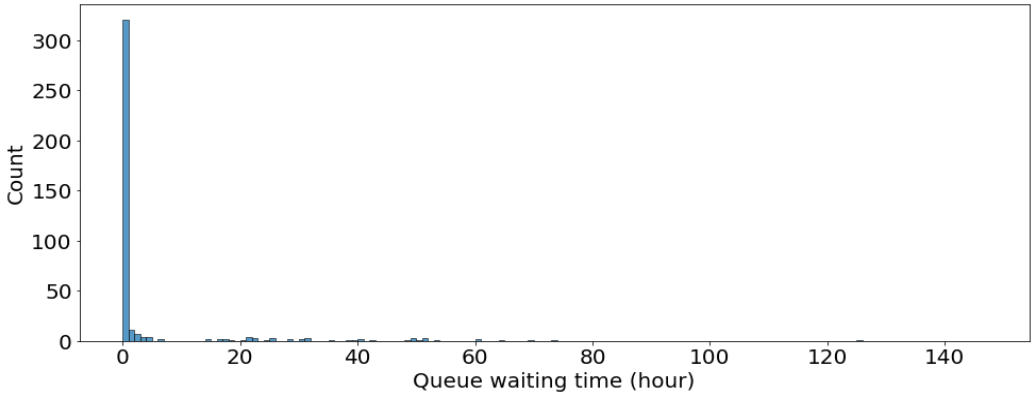
<Low> * requested GPU 2개 이하



Job의 Queue 대기시간 분포 (멀티 Queue 적용전) 평균: 32.3시간



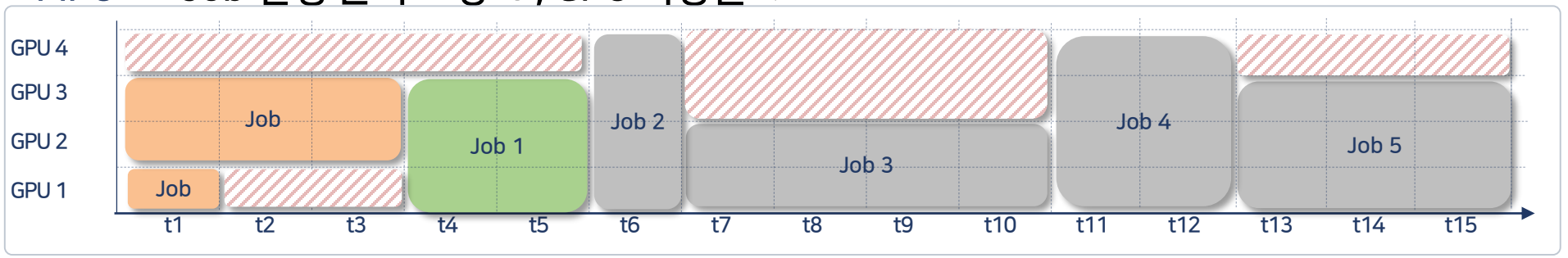
Job의 Queue 대기시간 분포 (멀티 Queue 적용후) 평균: 5.77시간



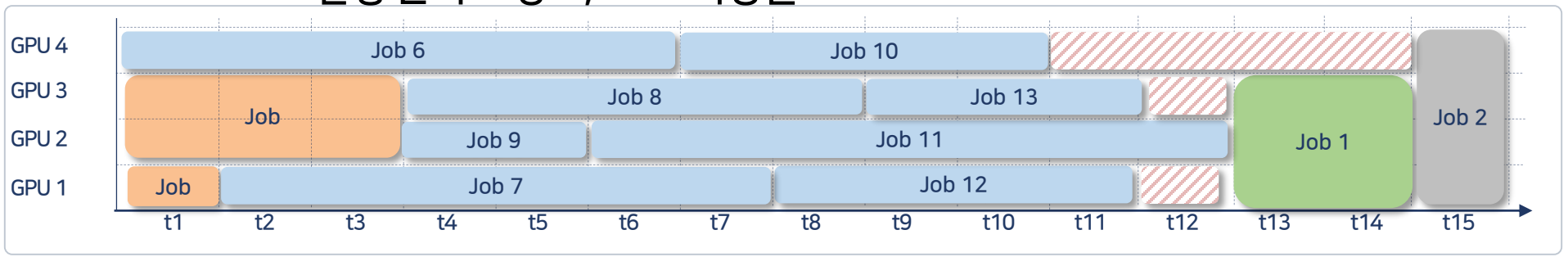
수행시간 예측 기반 스케줄러

수행시간 예측 기반 스케줄러 개발 배경

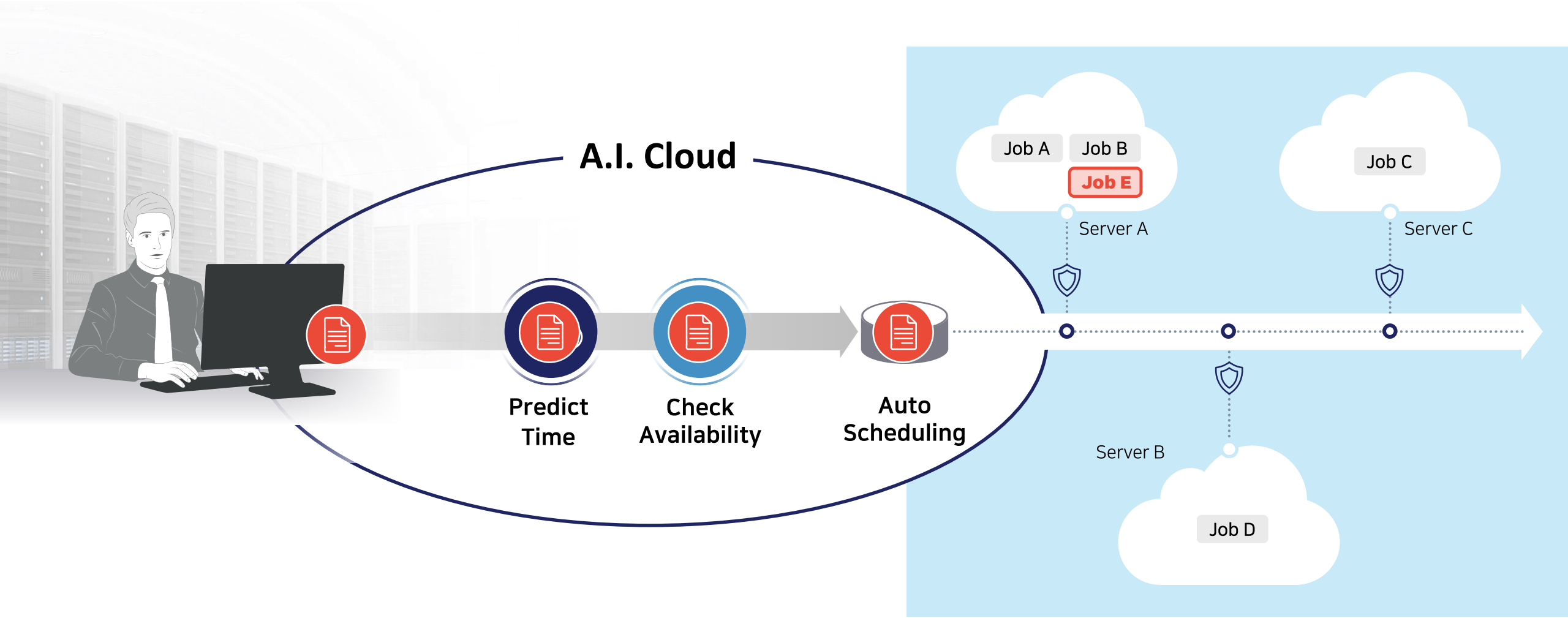
< FIFO > : Job 실행 순서 보장 ○, GPU 사용률 ↓



< Backfill > : Job 실행 순서 보장 X, GPU 사용률 ↑



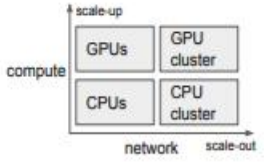
Job수행 예측 기반 스케줄러



ML/DL Job 수행 시간 예측 모델

연산 모델

Software framework



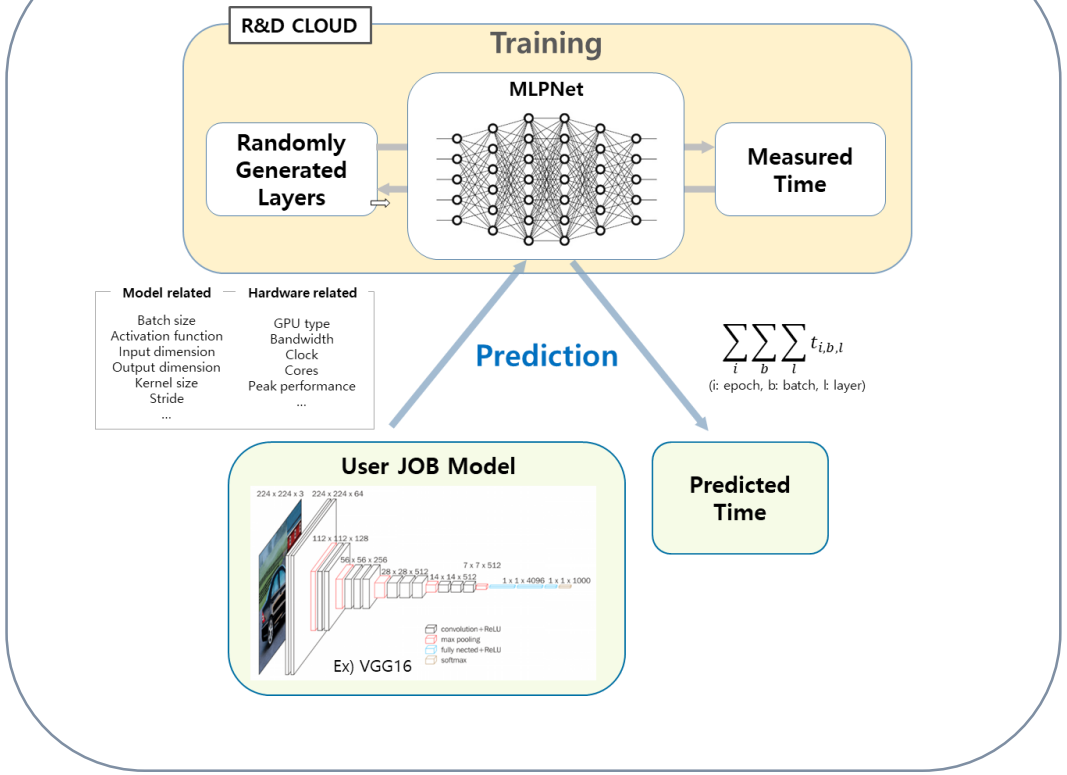
- Memory (Data, weights, gradients, activations)
- Dependencies (Network architecture)
- Complexity (e.g. FLOP counts)
- Operation selection (e.g. GEMM, FFT, Tiled FFT)
- Parallelization strategies (Model parallel, data parallel)
- Communication scheme (OneToAll, Tree AllReduce, Butterfly AllReduce)
- Communication bandwidth (GB/s)
- Computation speed (TFLOPS)

Computation

Communication

Execution Time

데이터 모델



ML/DL Job 수행시간 예측

플랫폼 내에서의 Job 수행시간 예측을 위한 입력 Feature 최적화

Feature

Model Related

Dense	Conv	Recurrent
Activation function		
Optimizer		
Batch size		
Num. of inputs	Matrix size	Recurrent type
Num. of neurons	Kernel size	Bidirectional
	Input depth	
	Output depth	
	Stride size	
	Input padding	
	Kernel size	

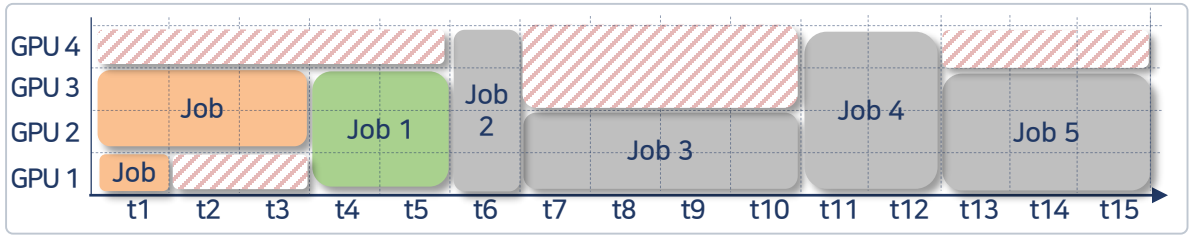
Hardware Related

GPU
Type
Memory
Clock speed
Bandwidth
Core count
Peak performance
Count
Connectivity

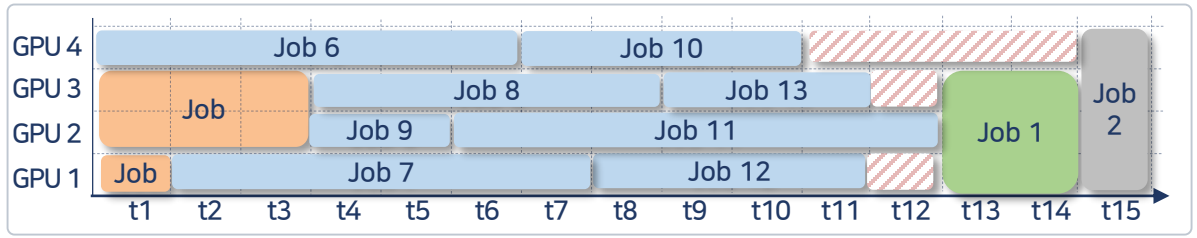
Multi-GPUs

수행시간 예측 기반 Backfill

< FIFO >



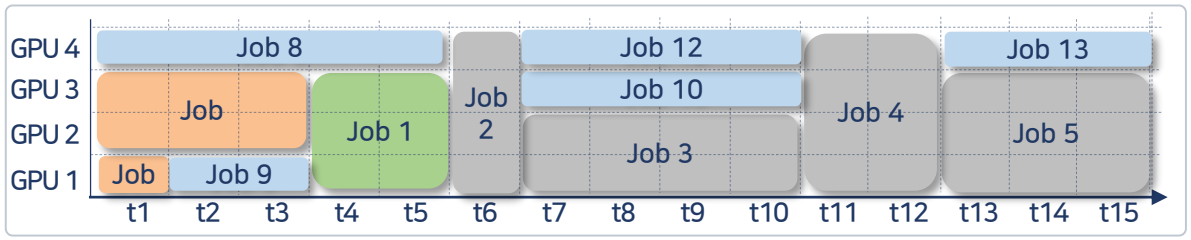
< Default Backfill >



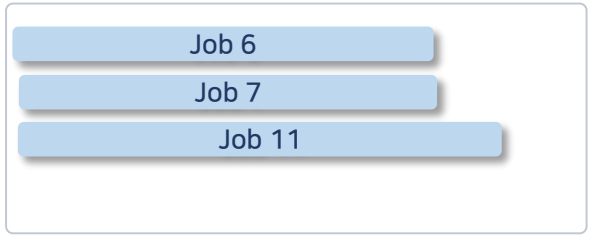
+ 순서보장

+ 사용율

< Predicted Time Based Backfill >

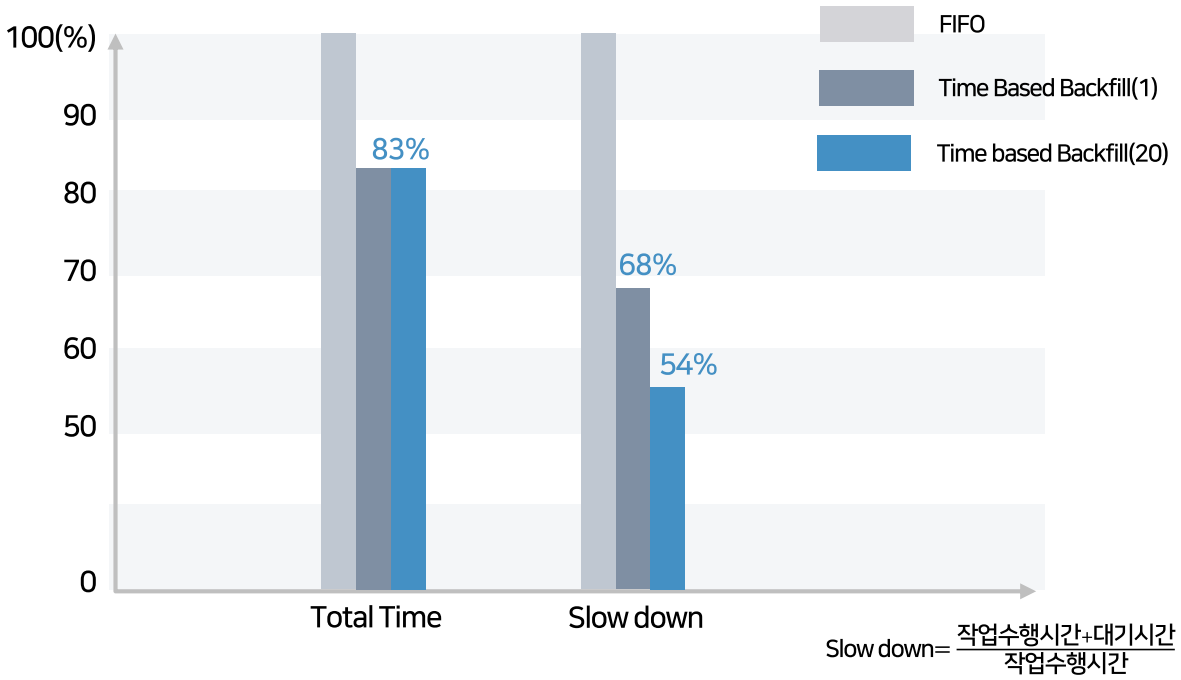


QUEUE



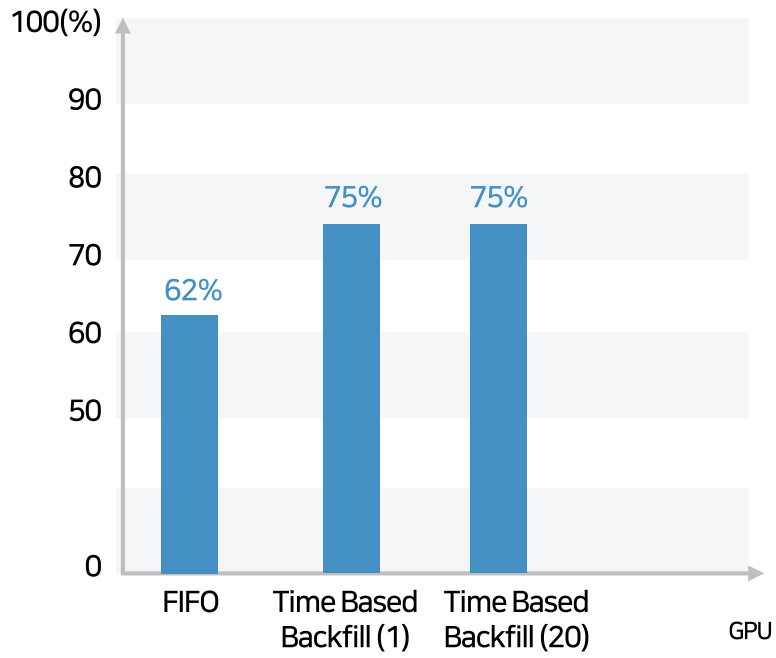
수행 예측 기반 Backfill 스케줄러 성능 측정 결과

Time Related (Normalized)



Slow down= $\frac{\text{작업수행시간+대기시간}}{\text{작업수행시간}}$

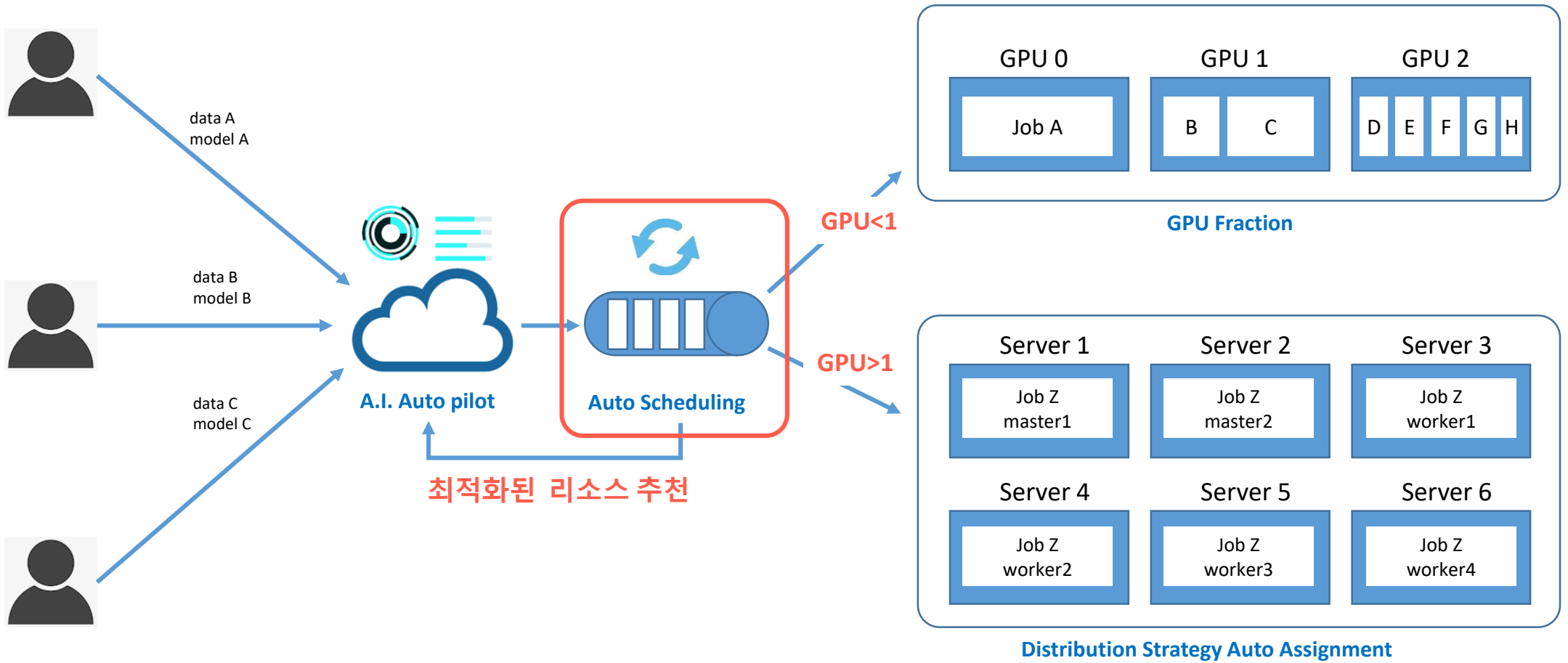
GPU Utilization



GPU Utilization = $\frac{\text{사용된 } GPU* \text{시간}}{\text{가용 } GPU* \text{시간}}$

Job수행 시간 예측 활용 방법

최적화 된 리소스 추천
추천된 리소스 기반 스케줄링



Summary

- Job 시간 예측을 통한 효율적인 스케줄링 방법 확보
- GPU 사용율 극대화를 통한 가격 경쟁력 확보

- ML/DL Job을 분석한 최적의 GPU 개수 배정
- 네트워크 상황을 고려한 최적의 Job 배정

- 대기 시간 최소화의 따른 사용자 만족도 상승
- Job 순서 배정에 따른 공정함 확보

Thank you

SAMSUNG SDS