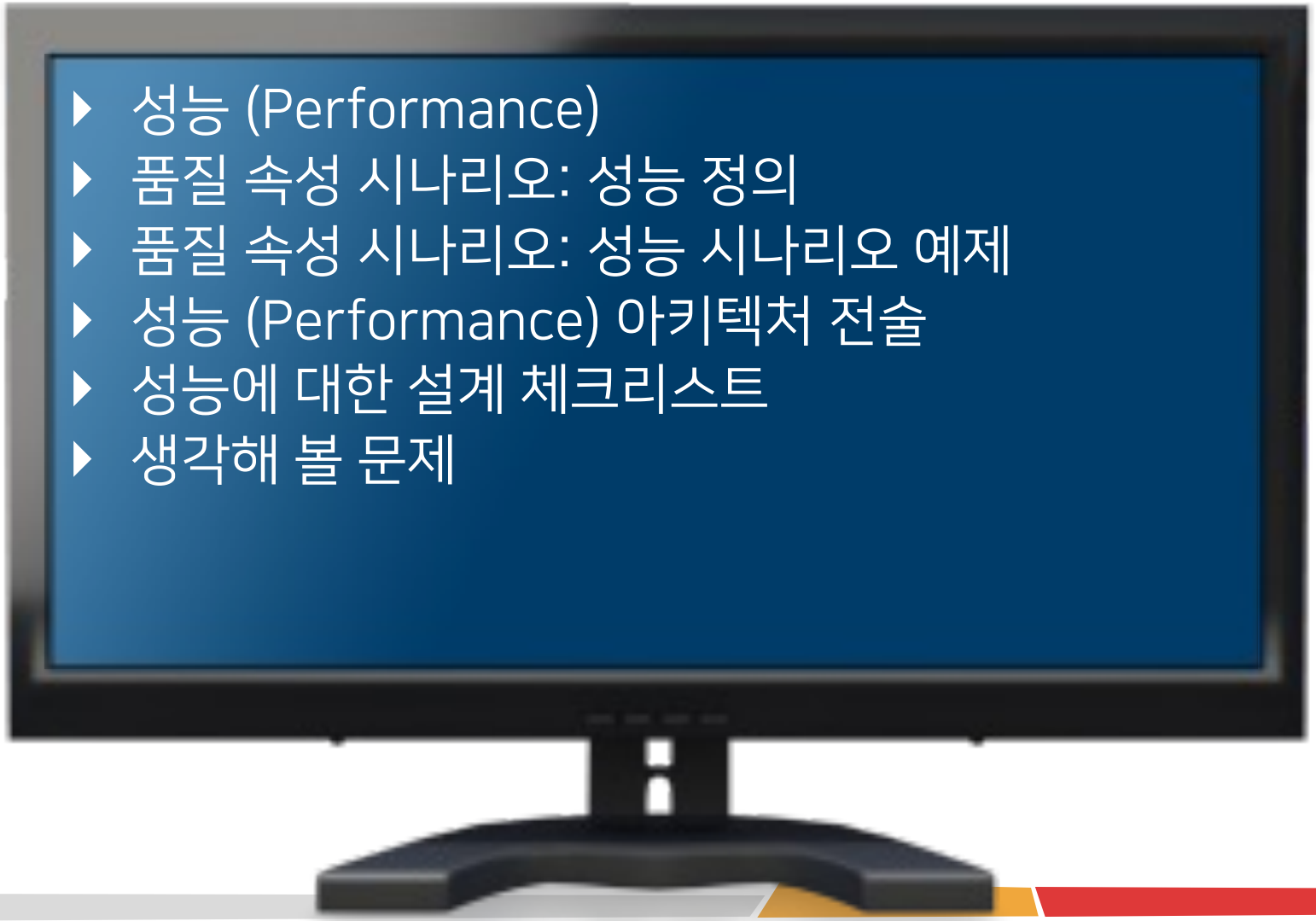
Spring 2023

Performance Tactics

Seonah Lee

Gyeongsang National University

성능 (Performance) 아키텍처 전술

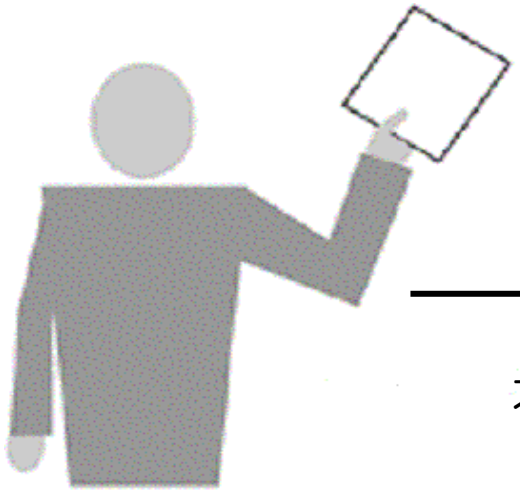


Performance

- 성능 (**Performance**)
  - 시간적 요구사항을 만족하기 위한 시간과 소프트웨어 시스템의 능력
  - 이벤트가 발생했을 때 시스템과 시스템의 요소들이 적절한 시점에 반응할 수 있어야 함
    - 이벤트 예: 인터럽트, 메시지, 사용자 흐름, 시간 흐름을 마킹하는 타이머 이벤트 등
  - 이를 위해 발생하는 이벤트의 특성을 파악하고, 이벤트에 반응하는 시스템과 시스템 요소의 시간에 기반한 반응을 파악해야 함

Quality Attribute Scenario for Performance

- 성능: **Event**가 발생했을 때, 이를 처리하는 시간과 동시에 이를 처리하는 능력



자극의 근원:
사용자

자극:
거래 요청

대상(Artifact):
시스템

환경:
정상 오퍼레이션

응답:
거래 처리

응답 측정:
평균 2초 대기시간

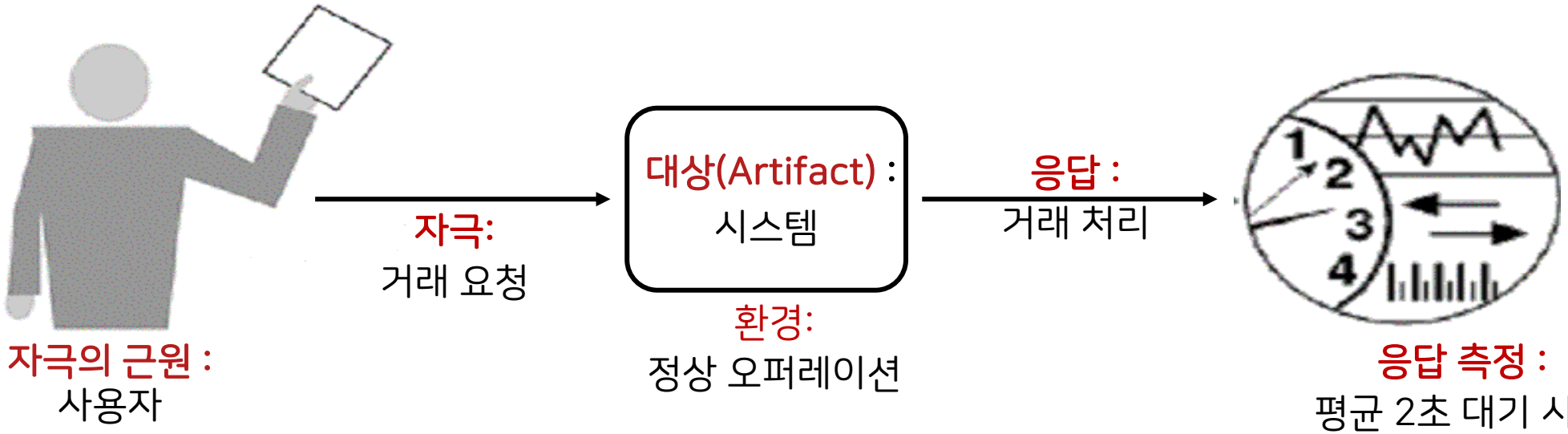
Quality Attribute Scenario for Performance

Component	Description
자극의 근원 (Source)	자극의 외부나 내부 소스로부터 발생 - 사용자, 시스템 내부
자극 (Stimulus)	이벤트의 도착 - 주기적(Periodic): 일정 시간 간격으로 예측가능하게 이벤트 도착 - 통계적(Stochastic): 확률적 분포에 따라 이벤트가 도착 - 산발성(Sporadic): 주기적이지도 않고 통계적이지도 않게 이벤트 도착 예: 사용자 입력은 0.2초의 간격을 기대할 수 있는 등의 특성화 할 수도 있음
환경 (Environment)	운용 모드: 정상, 비상 상황, 과부하(overload), 최대 부하(peak load) 등
대상 (Artifact)	시스템 혹은 시스템의 하나 이상의 컴포넌트들
응답 (Response)	도착한 이벤트의 처리, 서비스의 변경 수준
응답 측정 (Response Measure)	이벤트를 처리하는데 소요된 시간 - Latency: 이벤트가 도착하고 이에 시스템이 반응하는데 걸리는 시간 - Deadline: 예-엔진 제어에서 실린더가 특정 위치에 있을 때 연료가 점화됨 - Throughput: 일정한 단위 시간 내에 처리할 수 있는 트랜잭션의 수 - Jitter: 지연에 있어 허용 가능한 변동 정도 - Miss Rate: 시스템이 너무 바빠서 처리하지 못한 이벤트 수

Quality Attribute Scenario Example for Performance

- 성능 (Performance)의 시나리오 예
  - 정산 모드에서 사용자는 확률적으로 분당 1000개의 거래
    - 사용자: 자극의 근원
    - 분당 1000개의 거래: 자극

    를 요청하며, 거래는 평균 2초의 대기시간 내에 처리된다.
    - 요청: 응답
    - 평균 2초의 대기시간: 응답 측정



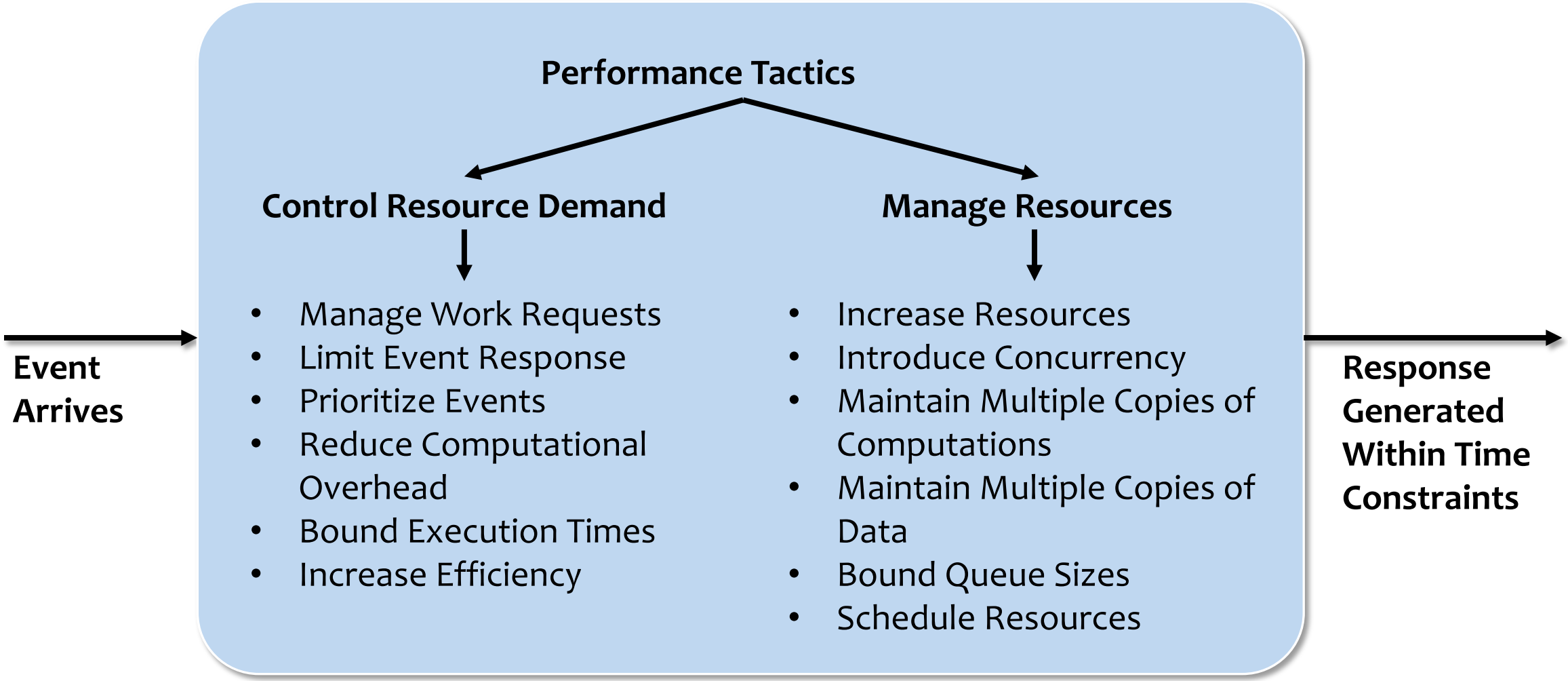
Performance Tactics

- 성능의 목적은 시스템에서 받은 이벤트에 대해서 일정 시간 제약 내에 반응을 발생하는 것이다.
  - 사용자 요청, 메시지, 시간 초과, 상태 변화 등
  - 리소스 사용과 리소스 대기 시간과 관련됨

Performance Tactics

- 응답 시간에 기여하는 두 가지 요소
  - **처리 시간**: 시스템이 반응하기 위해 작업하는 시간
    - 이벤트는 하나 이상의 컴포넌트에서 처리함
    - 하드웨어 자원: **CPU**, 데이터 저장장치, 네트워크 커뮤니케이션 대역폭, 메모리
    - 소프트웨어 자원: 설계에서 정의된 요소 (예: 버퍼, 크리티컬 섹션 접근 등)
  - **블록 시간**: 시스템이 반응할 수 없는 시간
    - 자원 차지: 시스템에 여러 이벤트 스트림이 도착할 때, 자원 사용 대기로 지연 발생
    - 자원 가용성: 자원이 오프라인, 컴포넌트의 실패 등으로 자원을 사용할 수 없음
    - 계산 의존성: 다른 계산의 결과와 동기화, 혹은 시작 전 대기 등

Performance Tactics



Performance Tactics

- 자원 요구의 이유
  - 성능 향상을 위한 하나의 방법은 리소스 요구를 주의 깊게 관리하는 것임
- 자원 요구 **(Control Resource Demand)**
  - **이벤트 도착 관리 (Manage event arrival)**
    - **SLA(Service Level Agreement):** 지원가능한 최대 이벤트 도착을 명시
      - 해당 최대치보다 더 많으면 요청을 처리할 요소의 중복 인스턴스 활용 가능
  - **샘플링 비율 관리 (Manage sampling rate)**
    - 데이터 스트림에서의 샘플링 빈도를 줄이는 것이 가능하다면, 자원 요구도 줄일 수 있으나, 다른 손실이 있을 수 있음
      - 예: 신호 처리에서 서로 다른 코덱은 서로 다른 샘플링 빈도와 포맷을 가짐.
    - 본 설계 결정은 지연을 예측할 수 있는 유지해야 함

Performance Tactics

- 자원 요구 (Control Resource Demand)
  - 이벤트 반응 제한 (Limit event response)
    - 이벤트가 시스템에 너무 빨리 도착할 때 이벤트는 큐에서 대기하며 처리될 때를 기다림.
      - 혹은 단순하게 이벤트를 무시해 버림
    - 이벤트를 처리하는 최대 비율을 선택
      - 실제 처리하는 이벤트에 대해 예측을 확인할 수 있음
    - 큐의 크기 충분히 크게 함. 혹은 이벤트를 무시할 경우 이벤트를 기록하거나 통지할지 결정
  - 이벤트 우선순위 설정 (Prioritize events)
    - 이벤트가 동등하게 중요하지 않을 때 우선순위화를 하여 중요도가 낮은 이벤트는 최소한의 자원을 사용 혹은 무시
      - 예: 빌딩 관리 시스템에서 다양한 알람이 울릴 수 있음,
      - 이 때 화재 경보는 냉방에 대한 경보보다 우선순위가 높아야 함

Performance Tactics

- 자원 요구 (Control Resource Demand)
  - 연산 부하 감소 (Reduce computational overhead)
    - Reduce indirection
      - 중재자 사용은 이벤트 스트림에 소모되는 자원을 증가
      - 중재자를 제거하여 지연 시간을 줄임
    - Co-locate communicating resources
      - 네트워크 커뮤니케이션 비용 등을 줄이기 위해서는 자원을 동등한 위치에 둠
      - 즉, 같은 프로세서에 컴포넌트를 배치하여 커뮤니케이션 비용을 줄임
    - Periodic cleaning
      - 오버헤드를 줄이기 위해 비효율적이 되는 리소스를 청소함,
      - 예를 들어 해시 테이블과 버츄얼 메모리 맵은 재초기화 작업이 필요함
      - 많은 시스템이 이러한 이유로 주기적으로 재부팅함

Performance Tactics

- 자원 요구 (**Control Resource Demand**)
  - 실행 횟수 제한 (**Bound execution times**)
    - 이벤트에 반응하는 실행 시간의 제한을 둠
    - 반복적인 데이터 기반 알고리즘들
      - 반복하는 횟수를 제한하여 실행 시간을 제한, 댓가로 계산의 정확도가 낮아짐
      - 해당 실행 횟수 제한은 샘플링 비율 전술을 관리하는 것과 연계될 수 있음
      - 정확도의 영향을 평가하고 그 결과가 충분히 받아들일 만한지 확인해야 함
  - 연산 효율성 증가 (**Increase efficiency of resource usage**)
    - 크리티컬 영역에 사용되는 알고리즘을 개선하여 지연 시간을 줄일 수 있음

Performance Tactics

- 자원 관리의 이유
  - 자원에 대한 요구를 제어할 수 없을 때라도, 자원 관리는 가능
- 자원 관리 **(Manage Resources)**
  - **가용한 자원 증가 (Increase resources)**
    - 빠른 프로세서, 추가 메모리, 빠른 네트워크 모두 지연을 줄일 수 있는 잠재력 있음
    - 가장 즉각적인 효과를 가져옴
  - **동시성 도입 (Introduce concurrency)**
    - 자원 요청을 병행으로 처리할 수 있다면, 블록되는 시간을 줄일 수 있음
    - 서로 다른 **Threads**의 이벤트 흐름 처리; 서로 다른 집합의 활동을 하는 추가 **Threads** 도입
    - 쓰레드를 사용한 동시성 도입 시, 스케줄링 정책을 선택하여 스케줄된 리소스 적절한 사용 필요

Performance Tactics

- 자원 관리 **(Manage Resources)**
  - 계산의 다중 복사본 관리 **(Maintain multiple copies of computations)**
    - 클라이언트 서버 패턴에서 여러 서버를 두어 계산을 중복적으로 하게 만들 수 있음
    - 로드 밸런싱으로 가용한 중복된 서버로 일을 옮기는 작업 고려 필요
      - 일 할당 전략은 단순한 **round-robin** 방식, 가장 바쁘지 않은 서버로의 할당 등 다양함
  - 데이터의 다중 복사본 관리 **(Maintain multiple copies of data)**
    - 데이터 복제 **(Data replication)**
      - 다수의 동시적인 접근의 경쟁을 줄이기 위하여 데이타를 복제 하는 것
    - 캐싱 **(Cashing)**
      - 데이터 사본을 저장하되, 접근 속도가 다른 저장소에 복사본을 둠
        - 최근 요청한 것을 복제로 유지, 사용자의 향후 요청을 예측한 복제 유지,
        - 사용자가 필요로 하는 데이타를 미리 예측하여 선제적으로 가져다 놓는 전술 등이 있을 수 있음
    - 복사본의 일관성 유지, 동기화, 복제할 데이타 선택 등을 시스템이 책임져야 함

Performance Tactics

- 자원 관리 (**Manage Resources**)
  - 큐 크기 제한 (**Bound queue sizes**)
    - 이는 큐에 도착하는 이벤트의 최대 갯수를 제어하고 도착한 이벤트를 처리하는 리소스를 제어
      - 큐가 오버플로우되었을 때의 전략과 분실하는 이벤트에 대한 미반응이 수용가능한지를 확인 필요
  - 자원 스케쥴링 (**Schedule resources**)
    - 자원에 대한 점유 경쟁이 있다면, 자원에 대한 스케쥴링이 필요
      - 프로세서, 버퍼, 네트워크에 대한 스케쥴링이 필요,
    - 각 자원의 특성을 이해하고 적절한 스케쥴링 전략을 선택해야 함
      - **First-in/First-out. FIFO**는 모든 자원을 동등하게 취급하며 순서대로 처리
      - **Fixed-priority scheduling**: 각 자원 요청에 고정된 우선순위를 할당하여 처리
      - **Semantic importance, Deadline monotonic** (짧은 데드라인 우선), **Rate monotonic** (짧은 주기 우선)
      - **Dynamic priority scheduling**: 각 자원 요청에 동적인 우선순위를 할당하여 처리
      - **Round-robin, Earliest-deadline-first, Least-slack-first**

Questions

Q1. “모든 시스템은 실시간 성능 제약사항을 갖는다”를 논의하자.

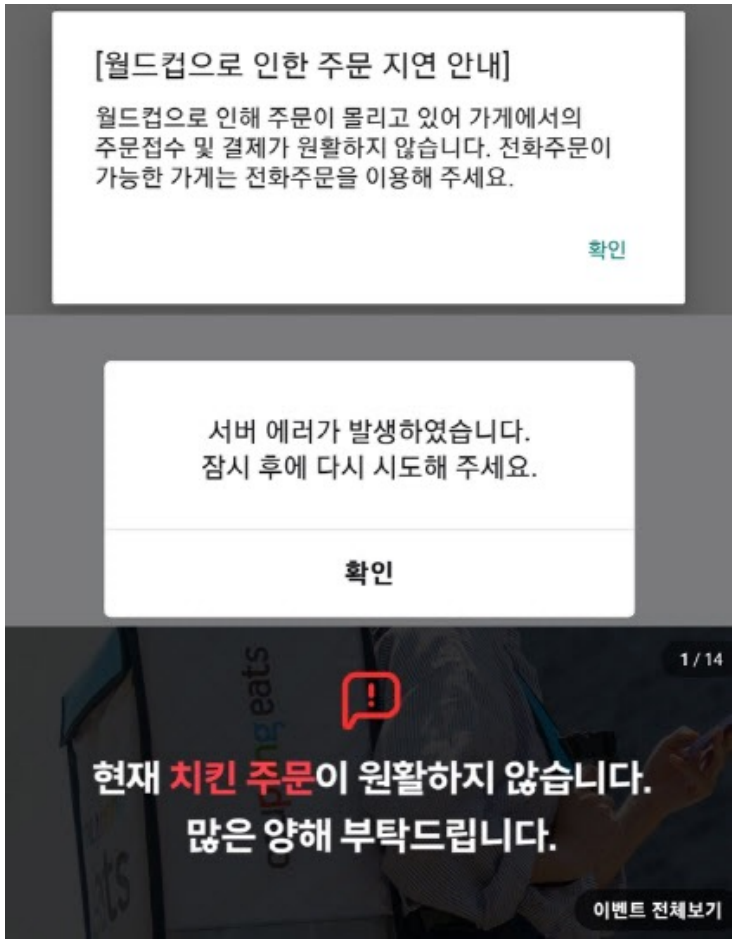
- 또는 반대 예를 제공하자.

Q2. 리소스 관리 전술을 적용하는 실세계(즉, 비소프트웨어) 예를 찾아보자.

- 예를 들어 재래식 큰 상자 소매점을 관리한다고 하자. 이들 전술을 사용하여 어떻게 사람들이 더 빨리 계산대를 통과하게 할 것인가?

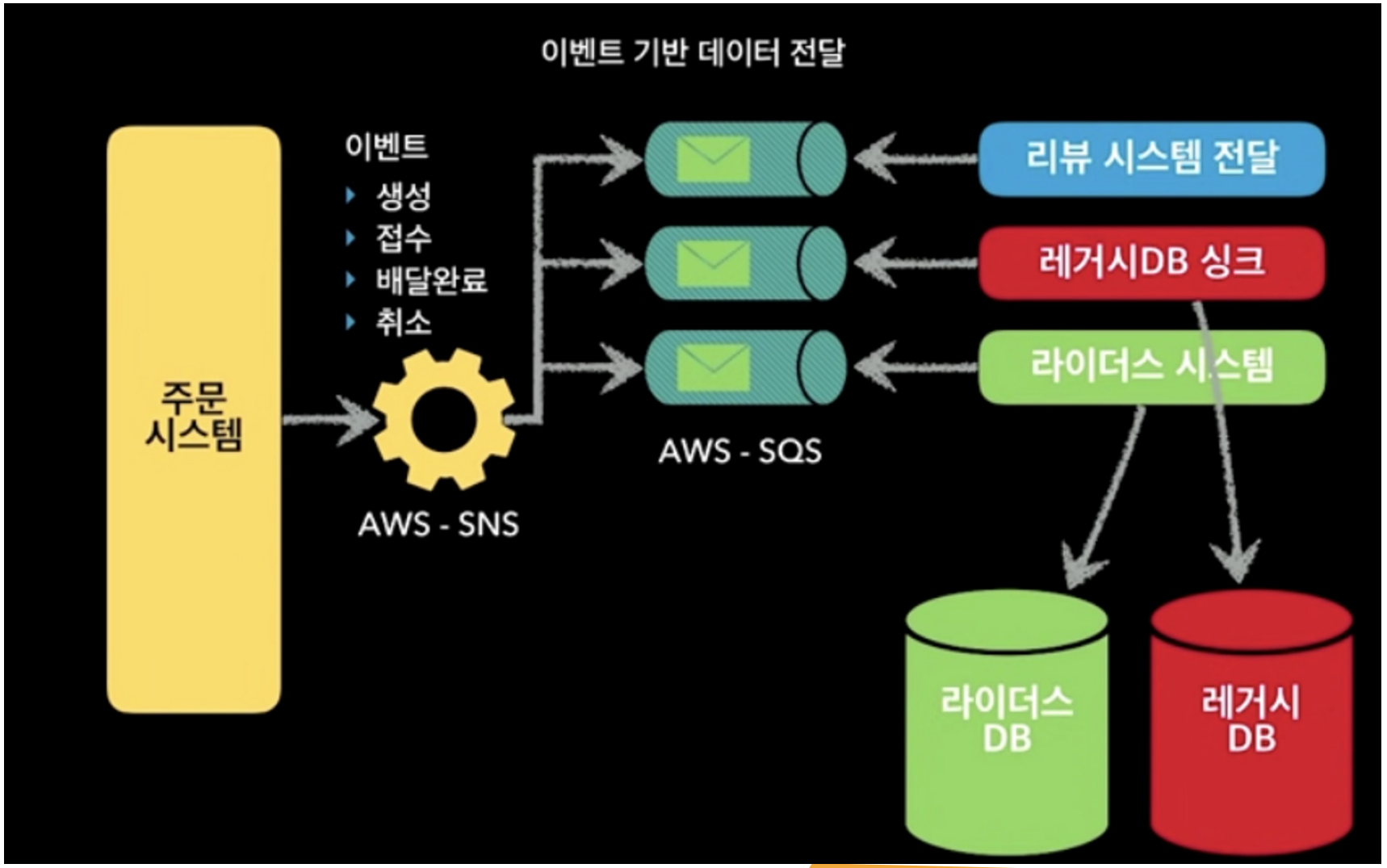
배달의 민족 앱 결제 오류

- 월드컵에 치킨 주문...배민 서버 오류 대혼란
  - **2022년 11월 24일** 고객의 결제 느려짐
    - 주문이 실패하는 경우가 생김
    - 짧은 시간에 주문이 많이 몰리면서 주문 실패
- 돈은 빠져나갔는데 음식 주문은 안되...
  - **2022년 5월 19일** 주문 후 결제는 이루어짐
  - 가게에 결제 내용이 전달되지 않았음



https://www.edaily.co.kr/news/read?newsId=04018006632529000&mediaCodeNo=257
https://www.seoul.co.kr/news/newsView.php?id=20220519500134

배달의 민족: 마이크로서비스



Question?

Seonah Lee
saleese@gmail.com