

# 비즈니스 트랜잭션 기반의 성능이슈

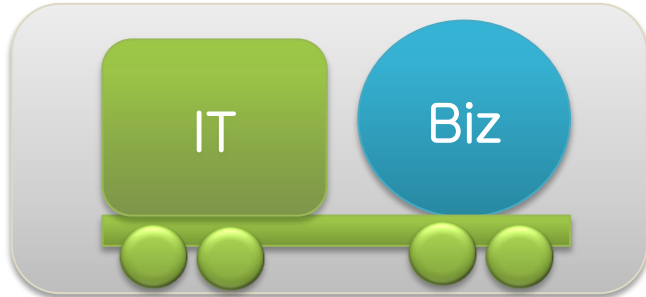
## 목 차

---

- 1. 비즈니스와 IT의 상호 연관성**
- 2. IT 성능관리의 중요성**
- 3. DB 성능 문제**
- 4. Infra 기반의 성능관리 방법 및 한계점**
- 5. 사용자 트랜잭션 관점의 성능관리란?**
- 6. TOP-T 서비스 정의**
- 7. Oracle Database Internal**

# 1. 비즈니스와 IT의 상호 연관성

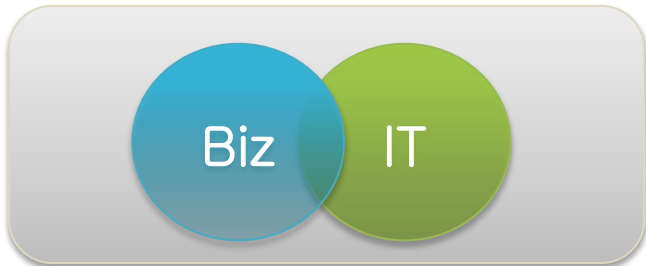
현재



- IT가 비즈니스를 창출하고, 매출을 이끄는 단계
- 비즈니스를 발전시키기 위해서, IT 성능관리는 필수적인 요소임



- 비즈니스의 중심부에 IT가 위치하는 단계
- 비즈니스를 유지하기 위해서, IT 성능관리는 필수적인 요소임



- IT가 비즈니스를 지원하는 단계
- 비즈니스와 IT가 상호 의존성을 갖게 됨
- IT 성능관리의 중요성이 부각됨

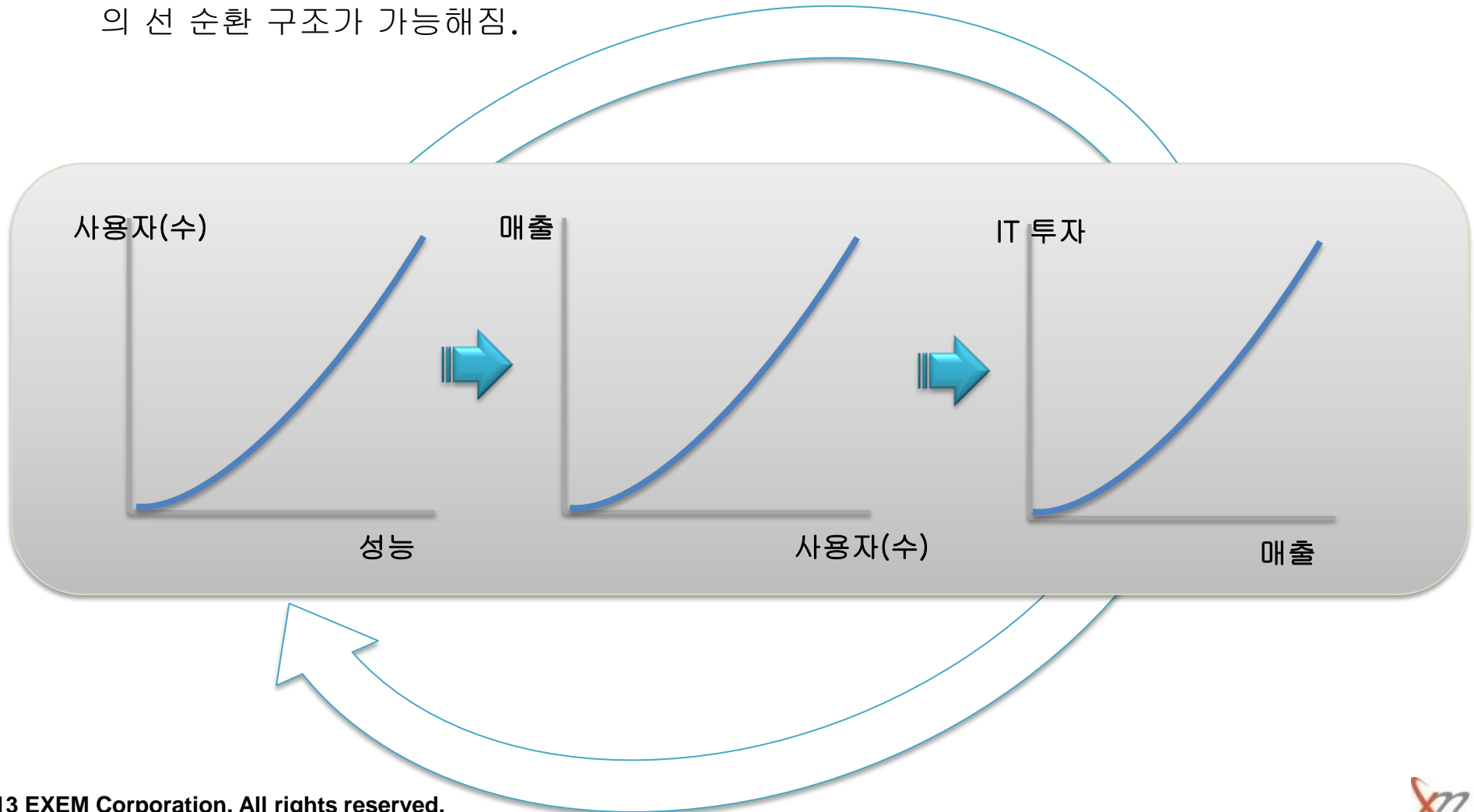


- 비즈니스와 IT를 별개로 취급
- IT 성능관리는 관심 대상이 아님

과거

## 2. IT 성능관리의 중요성

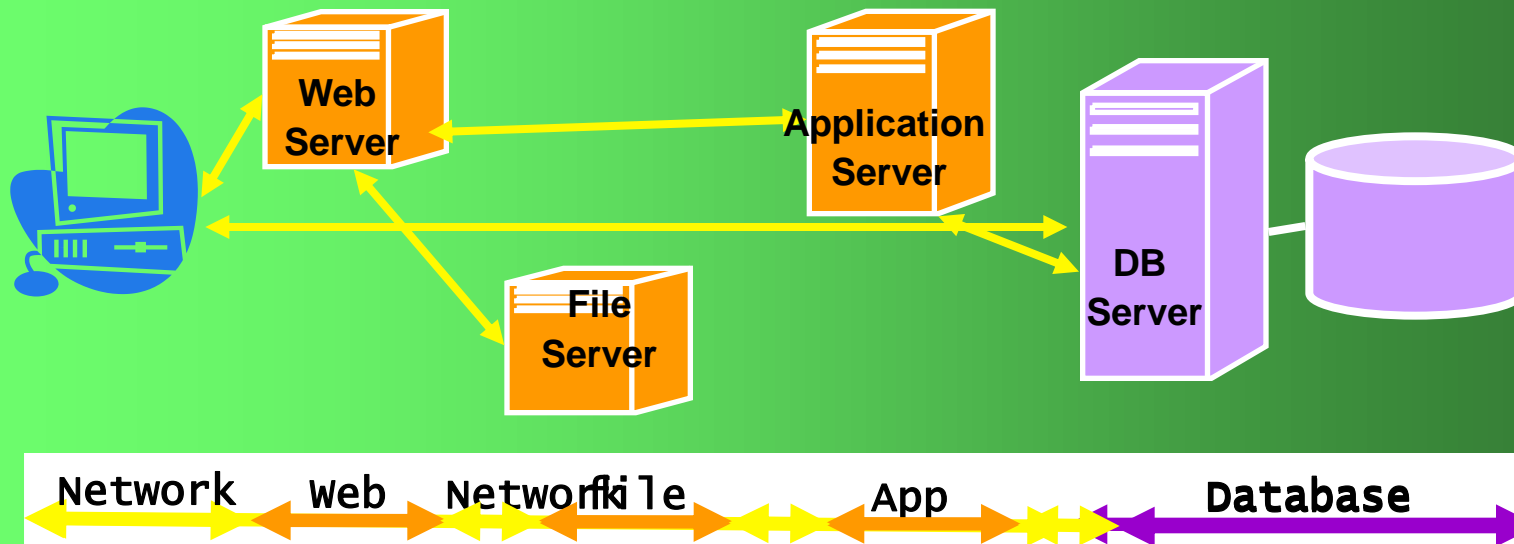
- IT 성능이 우수할수록, 사용자(또는 고객) 수가 증가하며, 이로 인한 매출 상승을 기대할 수 있음. 또한 매출 상승으로 인해, IT에 대한 재 투자가 가능하며, 이로 인한 IT 투자와 매출간의 선 순환 구조가 가능해짐.



## 2. IT 성능관리의 중요성

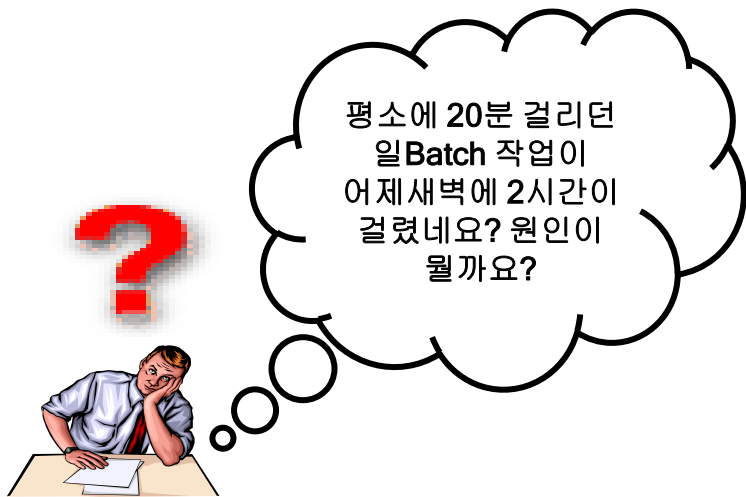
### 임계경로 (Critical Path)와 성능

#### Response Time

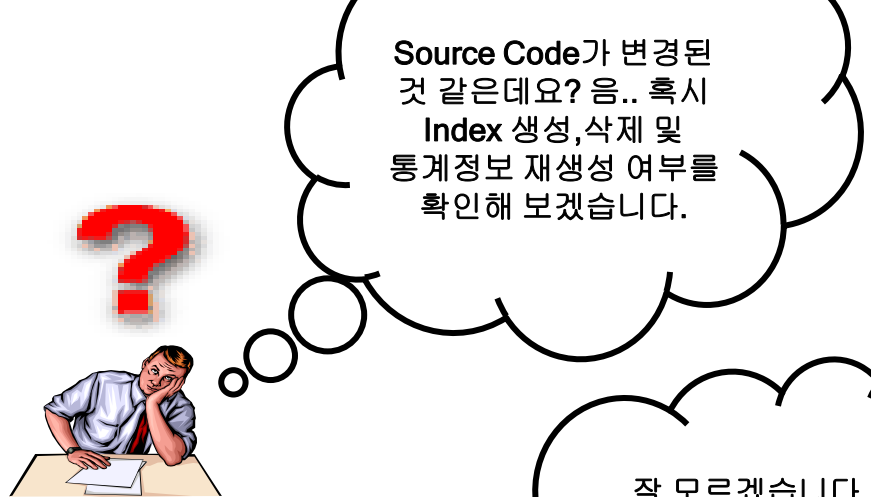


### 3. DB 성능 문제

#### 성능지연 사례1)



Developer



DBA



Developer

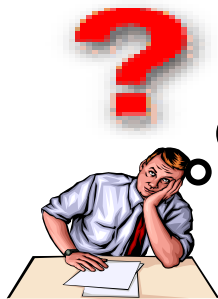


DBA



### 3. DB 성능 문제

#### 성능지연 분석사례1)



Developer

평소에 20분 걸리던  
일Batch 작업이  
어제새벽에 2시간이  
걸렸네요? 원인이  
뭘까요?



DBA

어제 수행되었던 일 Batch 작업 분석 결과,  
Data Fetch 작업 시에 수행된 SQL은 Latch  
free (cache buffers chains) 를 1시간  
대기하였습니다. 원인은 테스트 프로그램에서  
수행된 SQL이 동일 Block을 Access하면서  
문제를 유발한 것입니다.  
또한 Data 처리 작업 시에는 Enqueue Lock 을  
40분 대기하였습니다.  
Lock Holder는 개발자가 수행한 DML입니다.  
Job Scheduling 및 공지를 통해 이러한  
문제의 재발을 방지하도록 하겠습니다.



### 3. DB 성능 문제

#### 성능지연 사례2)



평소에 1초 이내 수행되던 화면이 특정 시간대에서 10초 이상 소요 되어 온라인 처리가 지연 되는 현상이 있습니다. 원인이 뭘까요?



Developer



확인해 보겠습니다.  
아니! CPU 사용율이 99%  
어디에 이상이 있는 걸까 ???  
왜? 종종 이런 일이 생기는 걸까  
혹시 바이러스..?  
아니면 해킹..?



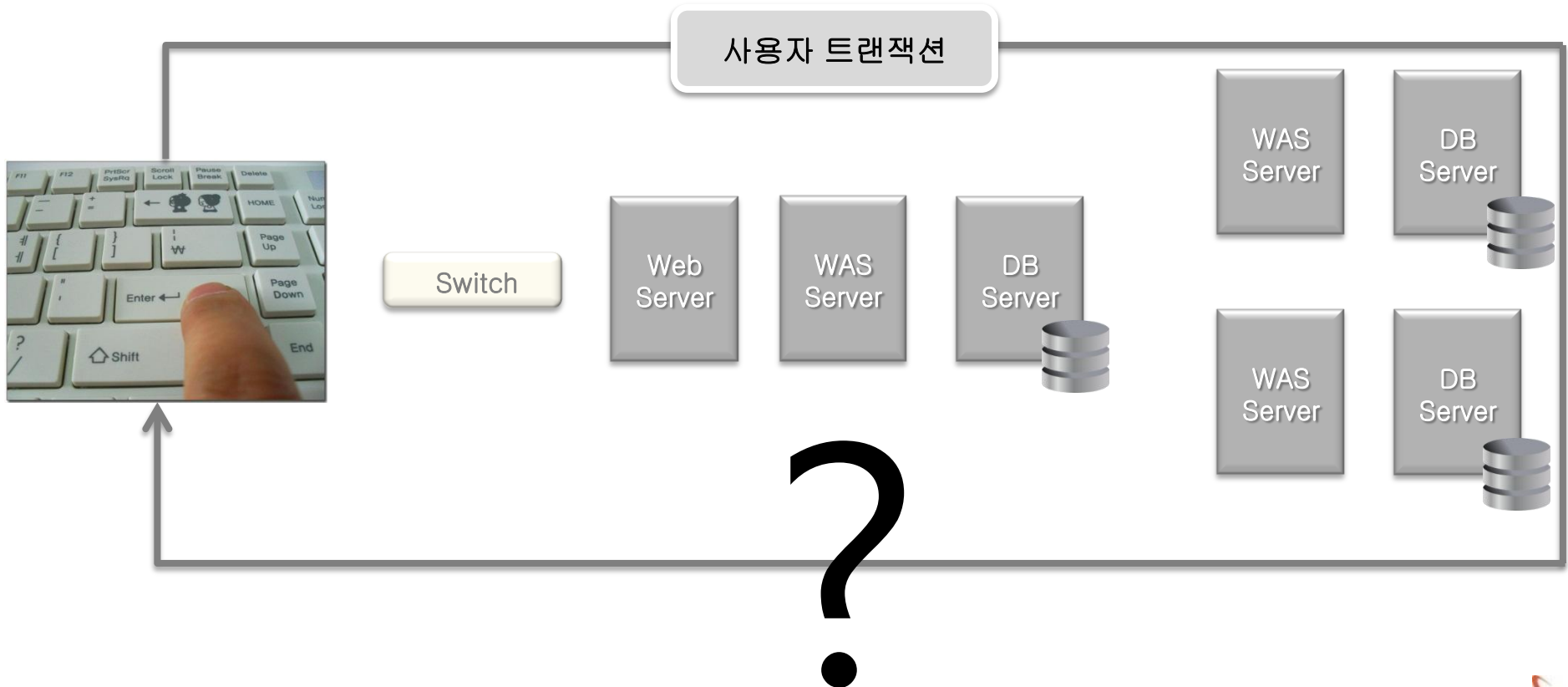
DBA





## 4. Infra 기반의 성능관리 방법 및 한계점

- IT 환경의 복잡도로 인해, 부분 최적화를 통한 전체 최적화가 힘들
- 다시 말해, Infra 기반의 개별 시스템 성능관리 기법은 효율성과 효과 측면에서 한계가 있음
  - 예) 구간 분석 불가능 등
- 이로 인해, 사용자 트랜잭션 관점의 성능관리 필요성이 대두됨



# DB 성능 문제

## 성능지연 사례)



DB 튜닝을 통해 CPU 사용율이  
감소하였으나, 제 프로그램은 여전히  
성능변화가 없습니다.  
확인 부탁드립니다.



Developer



DB모니터링 결과 3초 이상 수행  
되는 쿼리는 없습니다.  
다른 부분을 확인해 보셔야  
될 것 같습니다.



DBA



# WAS 성능 문제

## 성능지연 사례)



평소에 3초에 수행되는 프로그램이  
간혹 2분 이상 수행됩니다.  
확인 부탁드립니다.



Developer



모니터링 결과 DB 수행시간이  
불규칙적입니다.  
DB 파트에 확인해 보셔야  
될 것 같습니다.

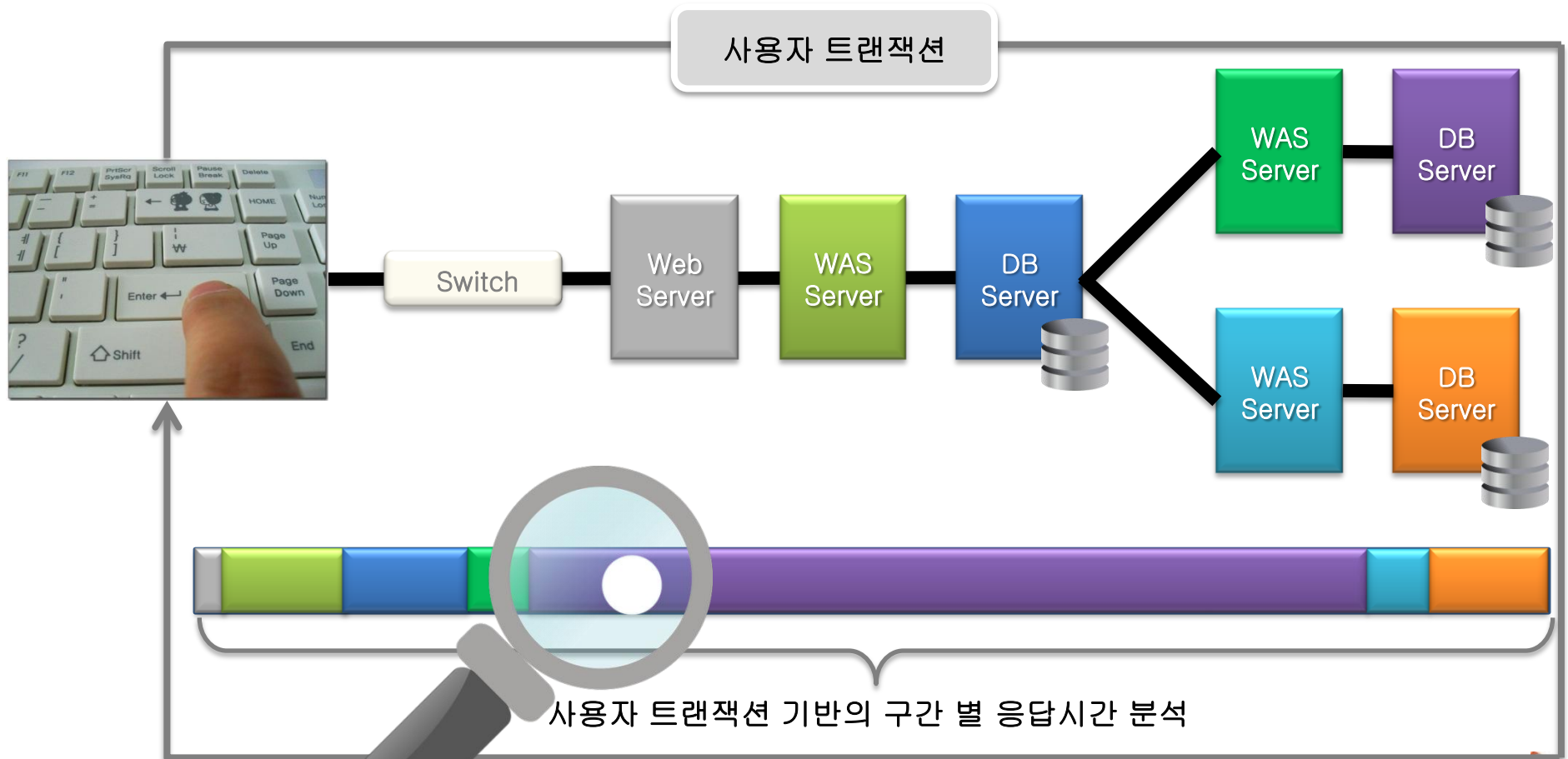


WAS ADMIN



## 5. 사용자 트랜잭션 기반의 성능관리란?

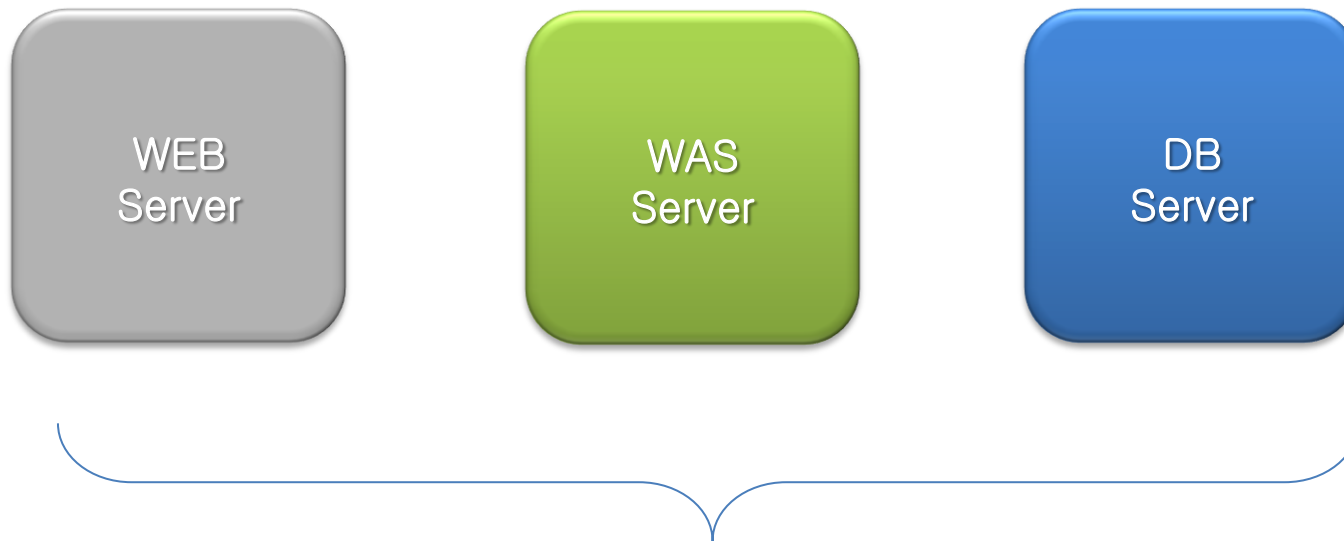
- 사용자가 체감하는 응답 시간을 구간별로 분석함으로써, 사용자 트랜잭션의 가시성을 확보하고, 성능 저하 구간에 대한 상세 분석을 통해, 문제 원인을 효과적으로 분석하고 조치하는 것을 의미함



## 6. TOP-T 서비스의 정의

- (주)엑셈은 사용자 트랜잭션 관점의 성능관리를 위해 TOP-T(Transaction Oriented Performance - Tuning) 서비스를 제공함
- TOP-T 서비스는 WEB 서비스 전 구간에 대해, 사용자 트랜잭션의 응답시간을 분석하고 개선하는 컨설팅 서비스임

### 서비스 범위

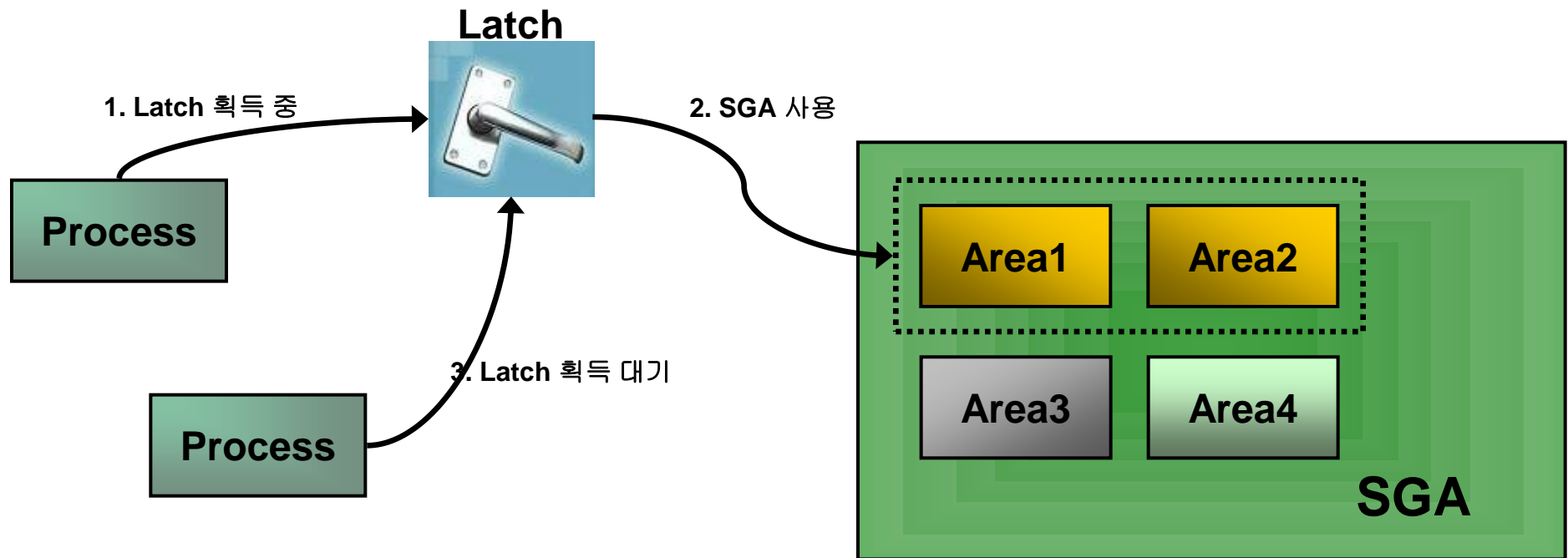


구간 분석을 통한 성능 저하 구간에 대한 상세 분석 및 개선

# Latch 개요

## Latch

- ✓ 가벼운 Lock(Light-weight Lock)
- ✓ 공유 Memory 영역(SGA)을 보호하기 위한 동기화 객체



## Latch 개요

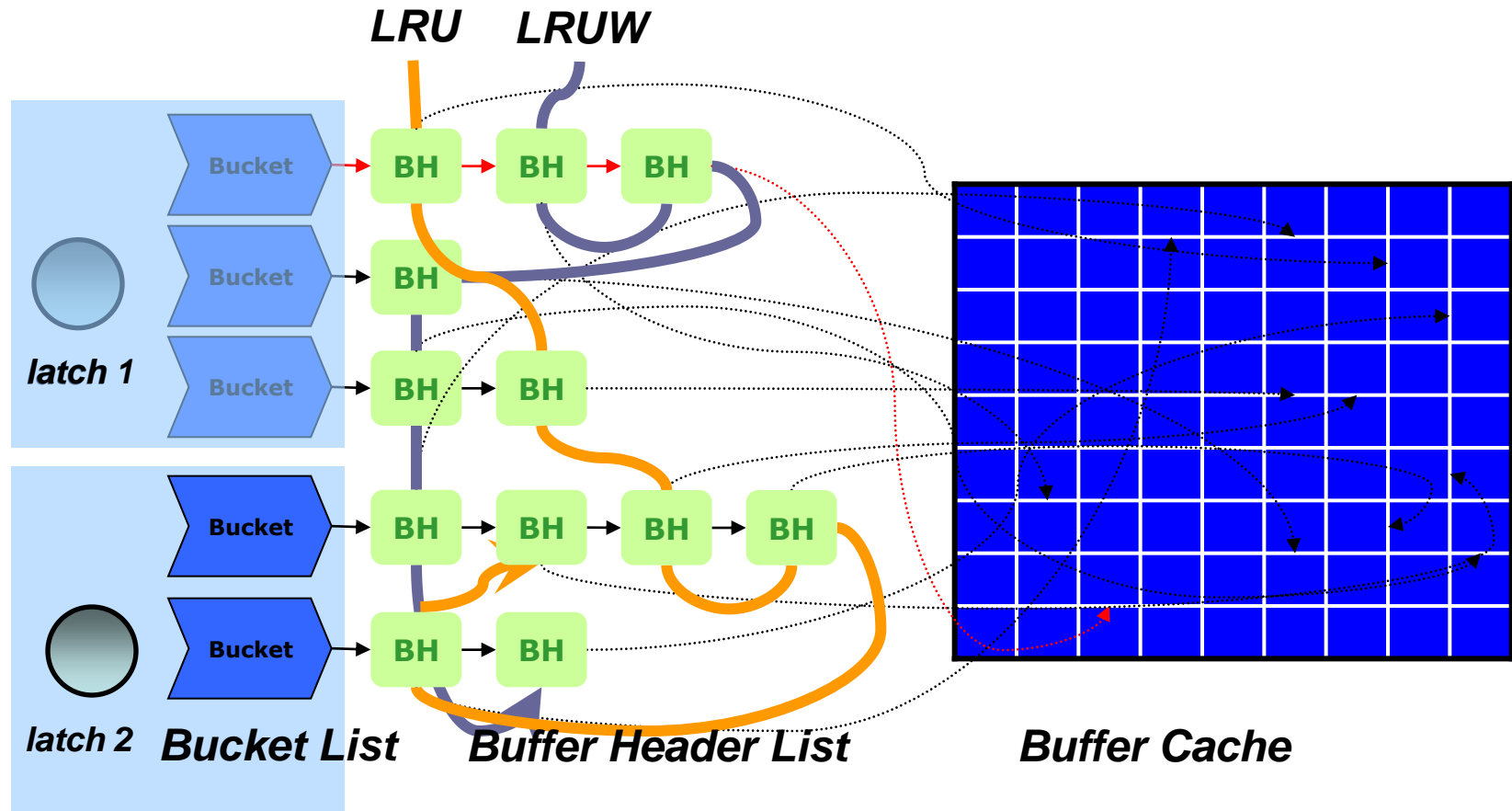
---

### Latch가 보호하는 Resource

- ✓ Latch는 **SGA**의 모든 하위 **Memory** 영역을 보호
  - Cache Buffer
  - Shared Pool
  - Library Cache
  - Redo Buffer
  - ...
- ✓ **Memory** 영역의 속성에 따라 하나 혹은 여러 개의 **Latch** 사용
  - 여러 개의 **Latch**를 사용하는 경우 **Parent/Child**의 관계
  - **Cache Buffer** : 수천 개 ~ 수 만개의 **Latch**
  - **Shared Pool** : 하나의 **Latch**
  - **Library Cache** : **CPU** 개수에 비례
  - **Redo Buffer** : 하나 ~ 여러 개의 **Latch**
  - ...

# Cache Buffer

## Data Buffer Cache 구조도

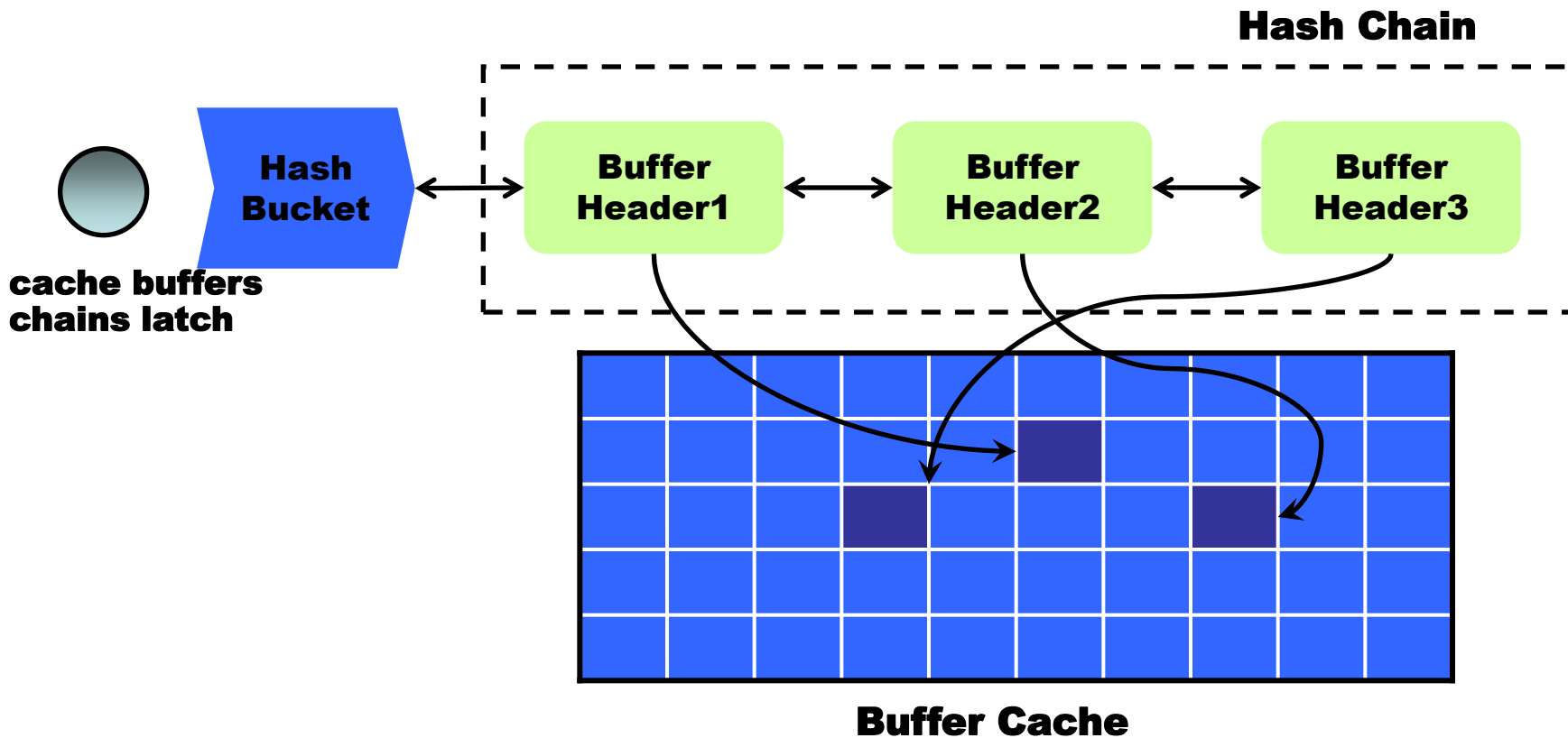




# Cache Buffer

## Hash Chain

- ✓ 하나의 Hash Bucket 에 연결된 Buffer Header 집합



## Cache Buffer

---

### cache buffers chains latch

- ✓ Hash Chain을 보호하는 목적의 Latch
- ✓ Shared / Exclusive Mode 로 획득 가능
  - 대부분의 다른 Latch들은 Exclusive Mode만 사용
- ✓ 하나의 Cache Buffers Chains Latch가 여러 개의 Hash Chain을 보호
- ✓ 부모/자식 Latch
  - V\$LATCH\_CHILDREN : 자식 Latch 통계 정보

## ***latch: cache buffers chains***

---

### Event 정의

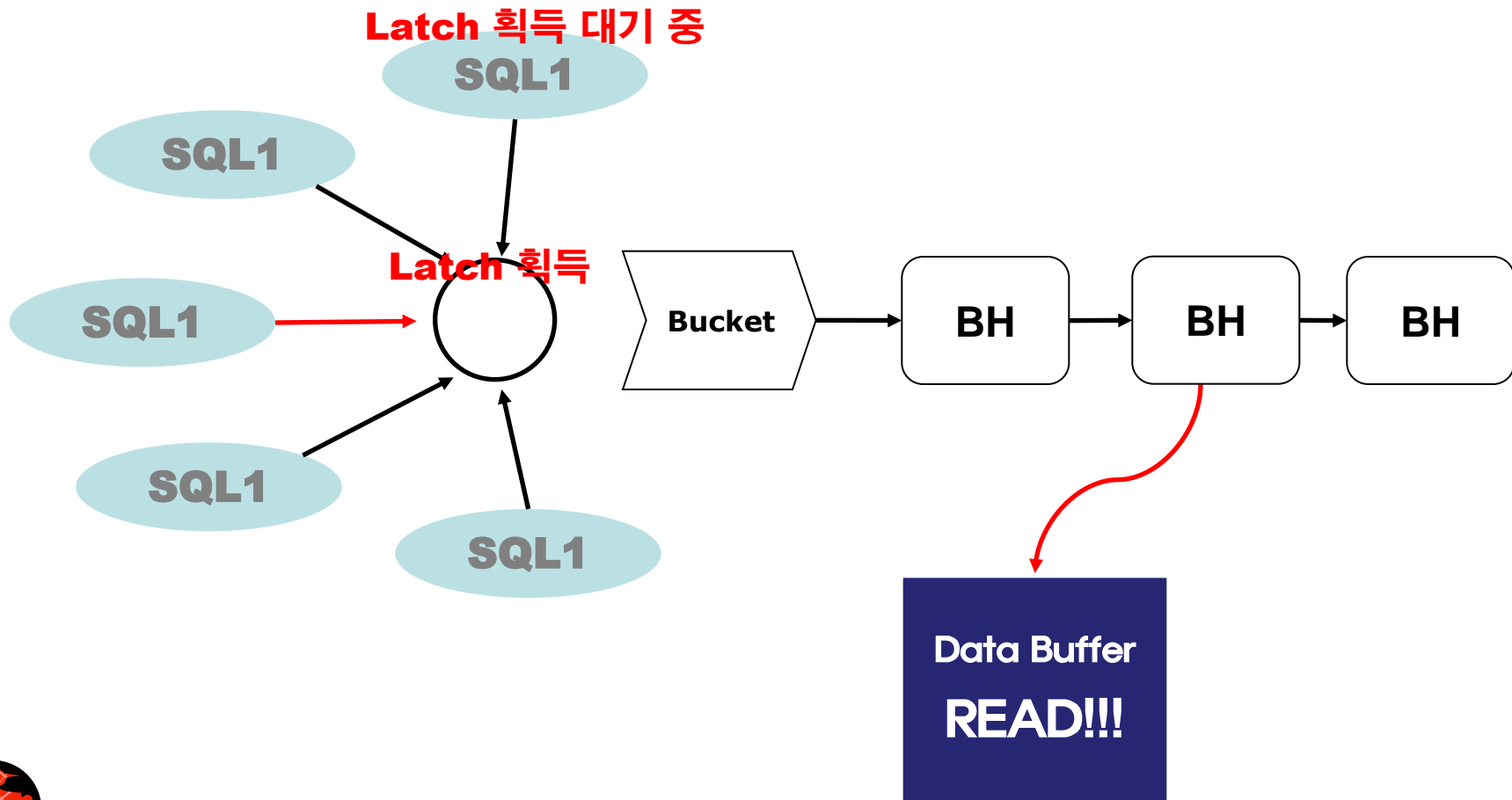
- ✓ **Cache Buffer Chain**을 탐색하기 위해 **cache buffers chains latch**를 획득하고자 대기하는 **Event**
  - **P1 = address**
  - **P2 = number**
  - **P3 = tries**

### 대기발생 사유

- ✓ 여러 세션이 동시에 동일 **cache buffers chains latch**를 **Exclusive Mode**로 요청
  - 동일 세그먼트에 대한 동시 다발적인 **Logical Reads**에 의해 유발
  - 핫 블록 또는 불량 **SQL**에 의해 발생

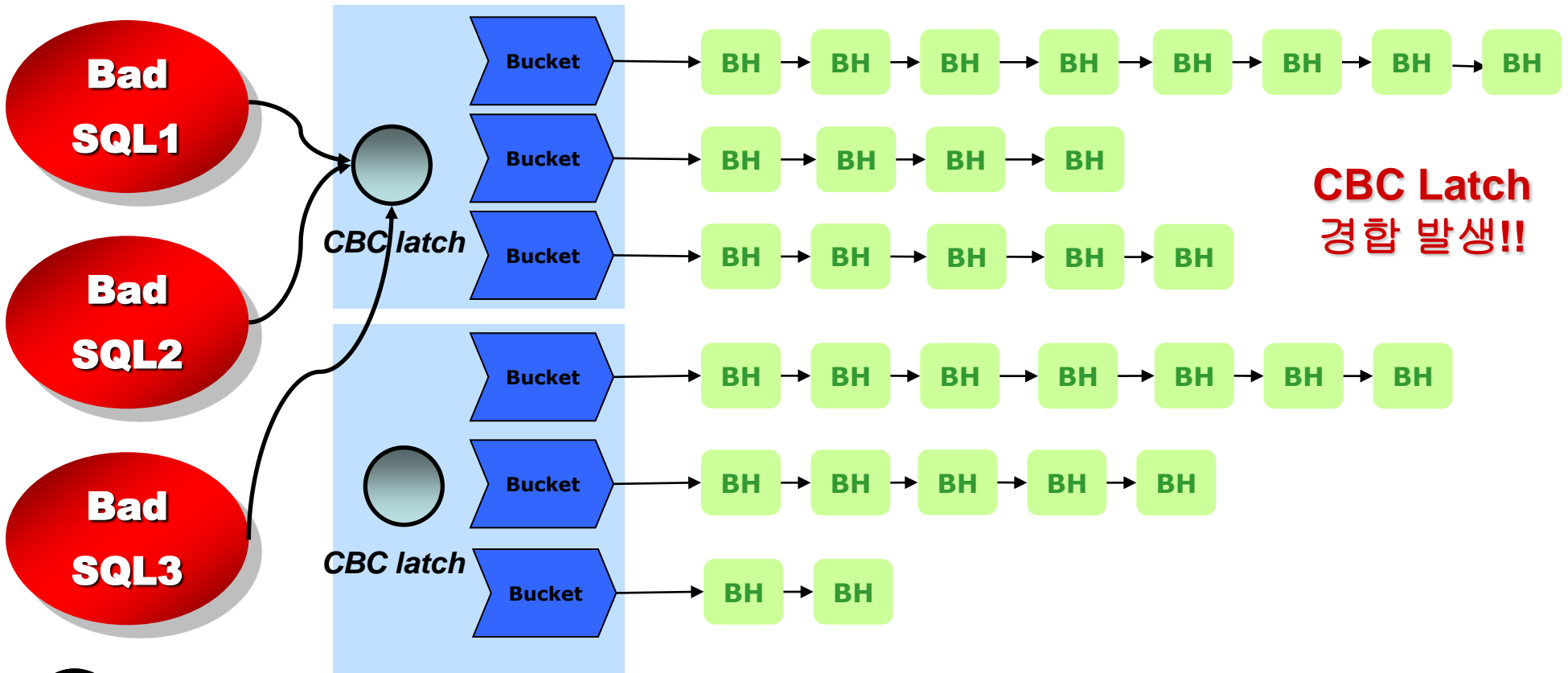
## ***latch: cache buffers chains***

### Hot Block에 의한 latch 경합



# ***latch: cache buffers chains***

## **Bad SQL에 의한 latch 경합**



## ***latch: cache buffers chains***

---

### Event 해소 방안

#### ✓ Hot Block 인 경우

- 블록 분산
- 인덱스 블록인 경우 인덱스 키 변경이나 다른 인덱스 경유 유도
- SQL Tuning
- 블록에 대한 요청 회수를 줄일 수 있는 방안 모색 - 어플리케이션 변경

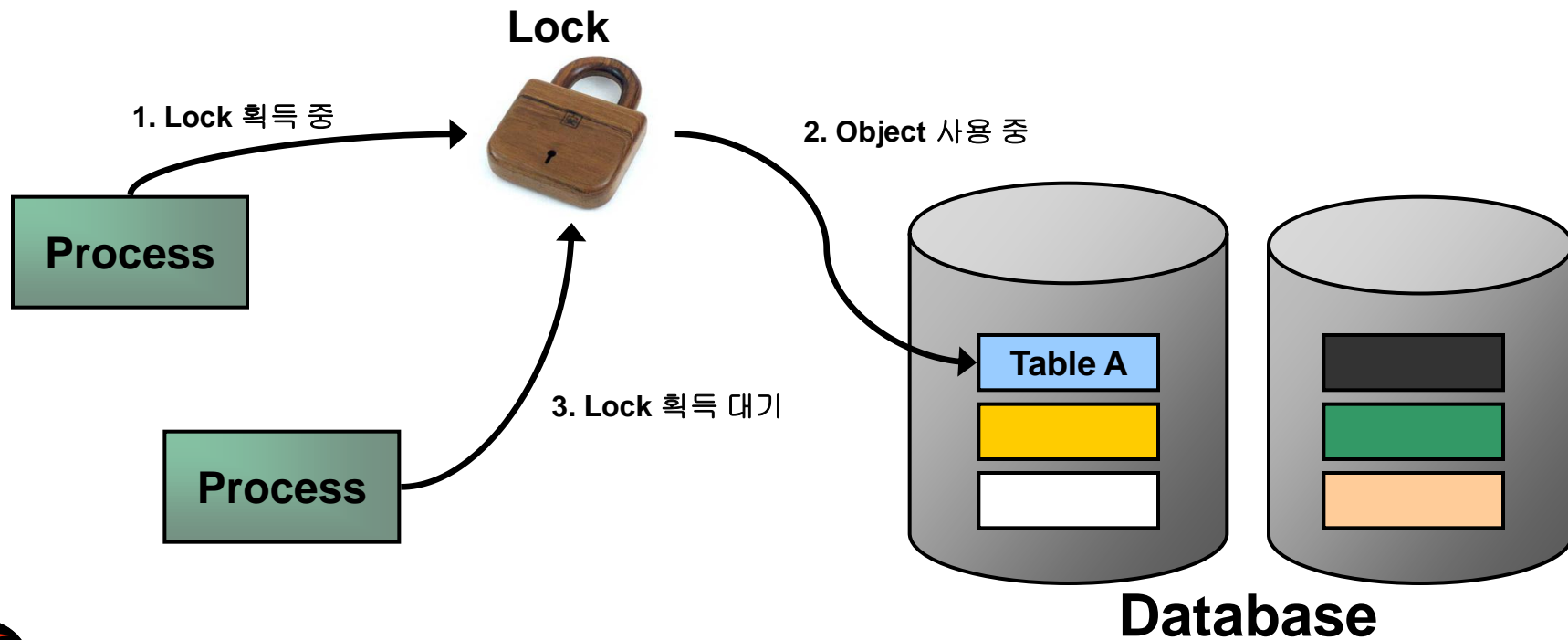
#### ✓ Bad SQL인 경우

- SQL Tuning

# Lock 개요

## Lock

- ✓ Latch보다 무거운 동기화 객체
- ✓ Database와 관련된 객체(Object)를 보호하는 동기화 객체



## Lock 개요

---

### Lock이 보호하는 Resource

- ✓ Latch가 보호하는 Resource 외의 모든 Database Object
  - Tablespace
  - Table
  - Row
  - Transaction
  - LCO(Library Cache Object)
  - RCO(Row Cache Object)
  - ...



# Lock 개요

---

## Lock 분류

### ✓ Enqueue Lock

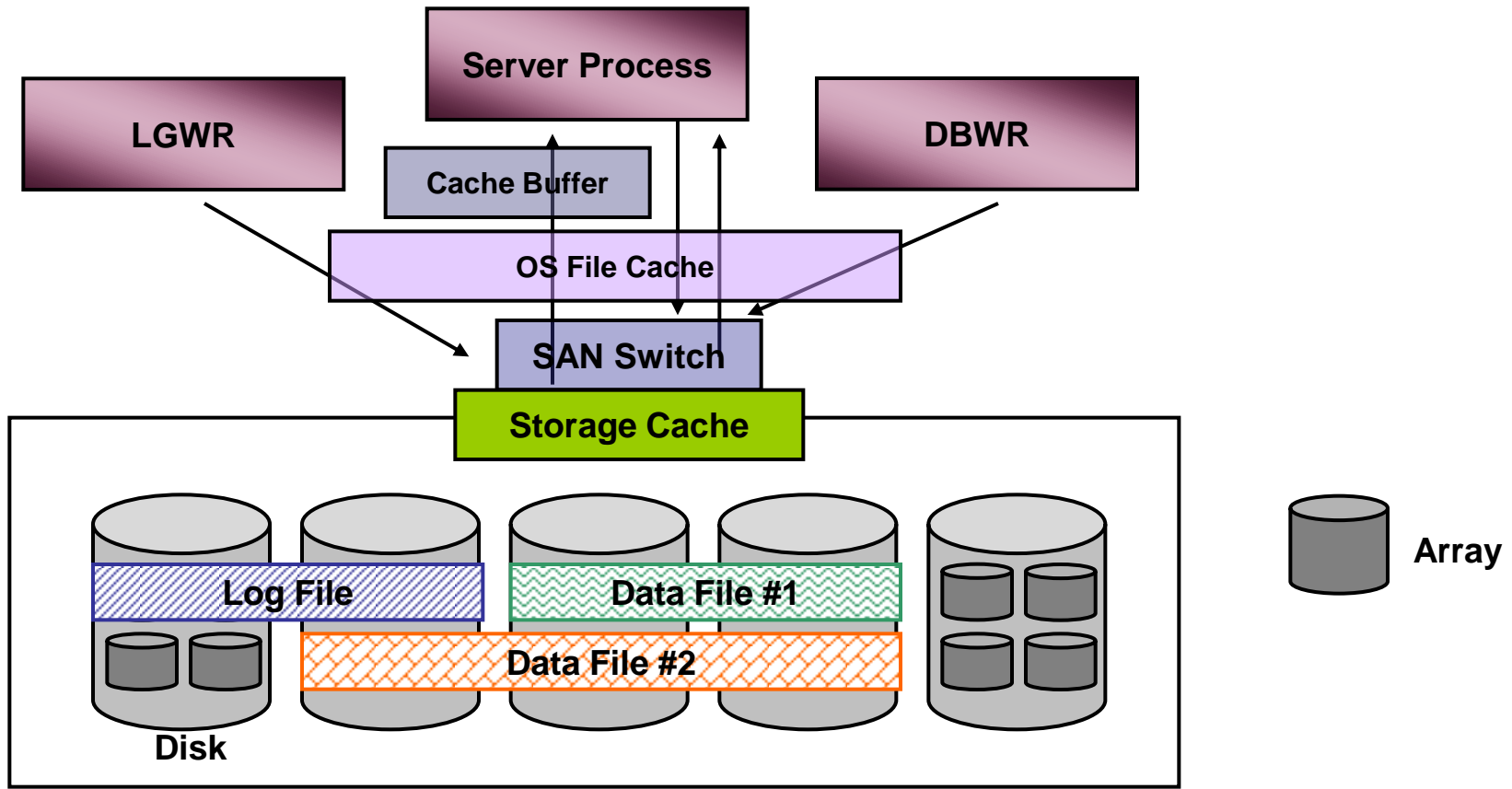
- User Type: TX, TM, UL
- System Type: US, HW, SQ, SV, TT, WF, ...
- V\$LOCK\_TYPE 뷰 참조(10g)

### ✓ 기타 Lock

- library cache lock, library cache pin
- row cache lock
- buffer lock

# Oracle I/O

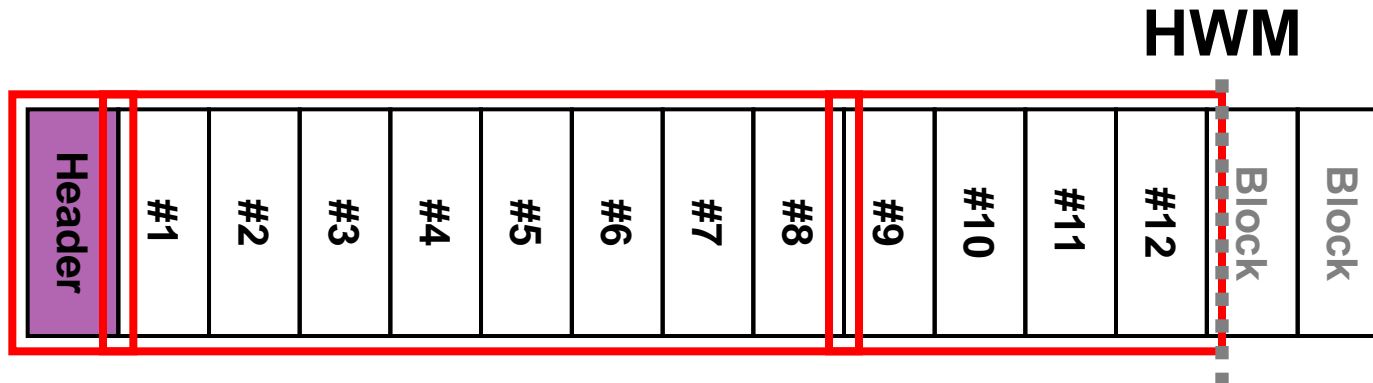
## Modern Enterprise Oracle I/O Architecture



# Oracle I/O

## Multi Block I/O

- ✓ 한번에 최대 MBRC(DB\_FILE\_MULTIBLOCK\_READ\_COUNT) 만큼 Block 읽음
- ✓ Full Table Scan, Index Fast Full Scan에서 사용
- ✓ Full Scan = Segment의 처음부터 끝(HWM)까지 Multi Block I/O로 읽어 들이는 작업



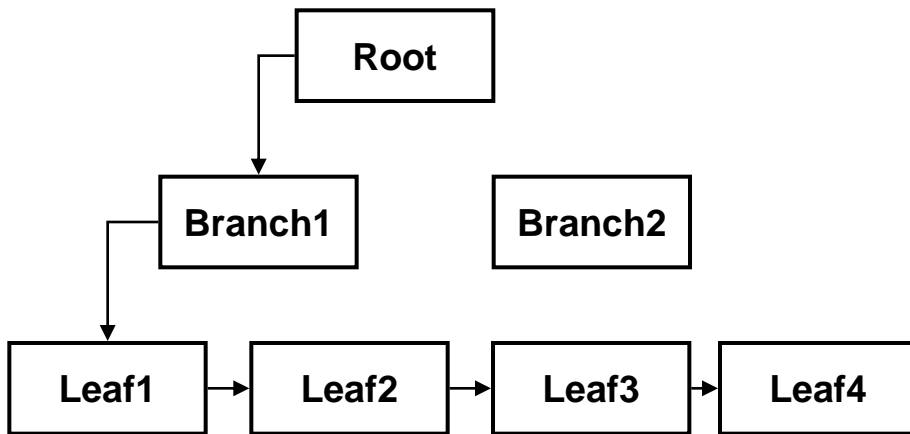
Physical Reads=1+8+4 = 13



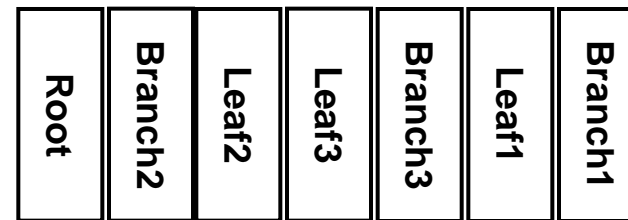
# Oracle I/O

## Index Full Scan vs Index Fast Full Scan

- ✓ Index Full Scan = Single Block I/O
- ✓ Index Fast Full Scan = Multi Block I/O



Index Full Scan



Index Fast Full Scan



## ***db file scattered read***

---

### Event 정의

- ✓ I/O System에 Multi Block I/O를 요청하고 응답이 오기를 기다리는 Event
  - P1 = file#
  - P2 = block#
  - P3 = blocks(실제로 읽은 블록 수)
- ✓ FTS, IFFS(Index Fast Full Scan) 등에서 발생

```
SQL> SELECT * FROM v$dbh WHERE file#=<P1> AND block#=<P2>
```

Or

```
SQL> ALTER SYSTEM DUMP DATAFILE <P1> BLOCK MIN <P2> BLOCK MAX <P2+P3-1>
```

## ***db file scattered read***

---

### ***db file scattered read*** 대기 과다 원인

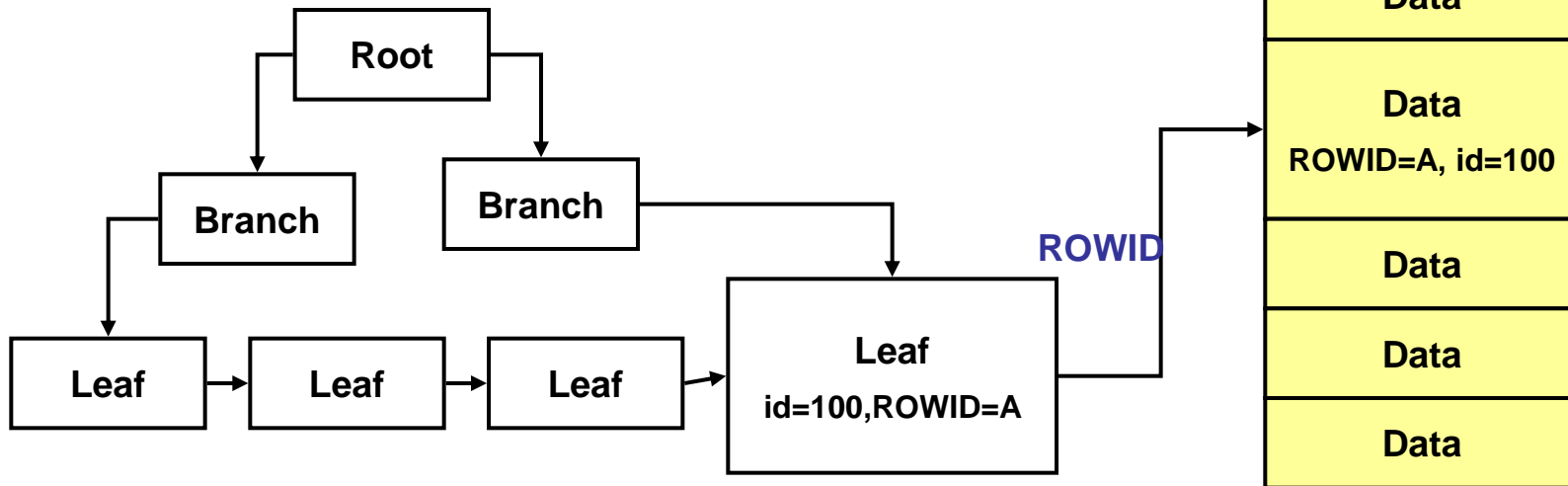
- ✓ 불필요한 Full Scan
  - SQL Tuning, Index Tuning
  
- ✓ I/O System 성능 저하
  - Disk I/O 성능 개선
  - Hot Spot 해소 : Logical Volume Level Striping, ASM 등

# Oracle I/O

## Single Block I/O

- ✓ 한번에 하나의 **Block**만을 읽어 들임
- ✓ **Index Scan(IFFS 제외), ROWID에 의한 Table Scan**

SELECT \* FROM TAB WHERE ID = 100



Logical Reads = 1 + 1 + 1 + 1 = 4



## ***db file sequential read***

---

### Event 정의

- ✓ I/O System에 **Single Block I/O**를 요청하고 응답이 오기를 기다리는 Event
  - P1 = file#
  - P2 = block#
  - P3 = 1

```
SQL> SELECT * FROM v$db WHERE file#=<P1> AND block#=<P2>
```

Or

```
SQL> ALTER SYSTEM DUMP DATAFILE <P1> BLOCK <P2>
```



## ***db file sequential read***

---

### ***db file sequential read* 발생 사유**

#### ✓ **Single Block I/O**

- **Index를 경유한 Scan**
  - **Index Range Scan**
  - **Index Unique Scan**
  - **Index Full Scan**
- **Chained Row / Migrated Row**
- **Multi Block I/O 도중에도 발생 가능**

## ***db file sequential read***

---

### ***db file sequential read*** 대기 과다 원인

- ✓ 비효율적인 **Index Scan**
  - SQL Tuning
  - Index 변경
- ✓ 지나치게 높은 **Clustering Factor**
  - Table Reorg를 통해 해결
- ✓ 지나치게 깊은 **Index 깊이**
  - Index Rebuild를 통해 해결
- ✓ **I/O System** 성능 저하
  - Disk I/O 성능 개선
  - Hot Spot 해소 : Logical Volume Level Striping, ASM 등

---

본서는 대한민국 저작권법에 의해 보호를 받는 저작물이므로 무단전재와 무단 복제 및 변형을 금합니다.