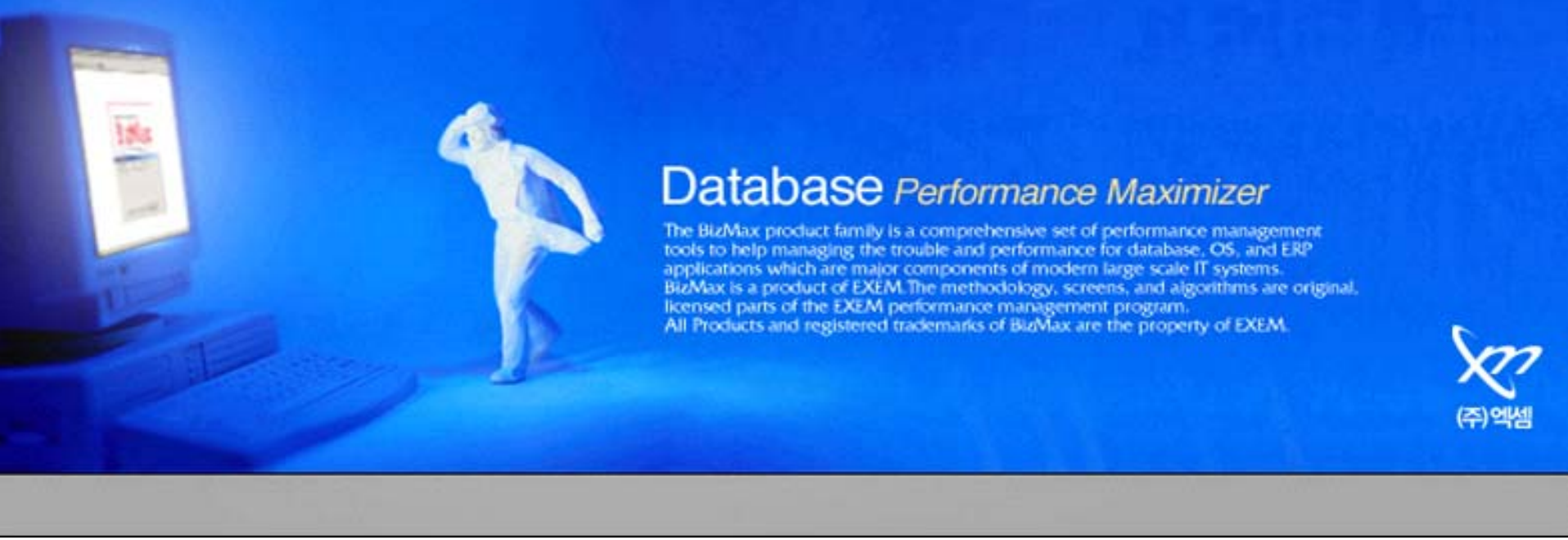


오라클 **STAT/EVENT** 를 활용한 분석 방법 사례**- Part I (I/O)**

㈜ 엑셈

2004. 12

목 차

I. 개요
II. STAT/EVENT 분석
III. 분석 사례
IV. 질의 응답

목 차

I. 개요
  I. STAT/EVENT 란?
  II. STAT/EVENT 분석의 필요성

II. STAT/EVENT 분석
III. 분석 사례
IV. 질의 응답

I. STAT/EVENT 란?

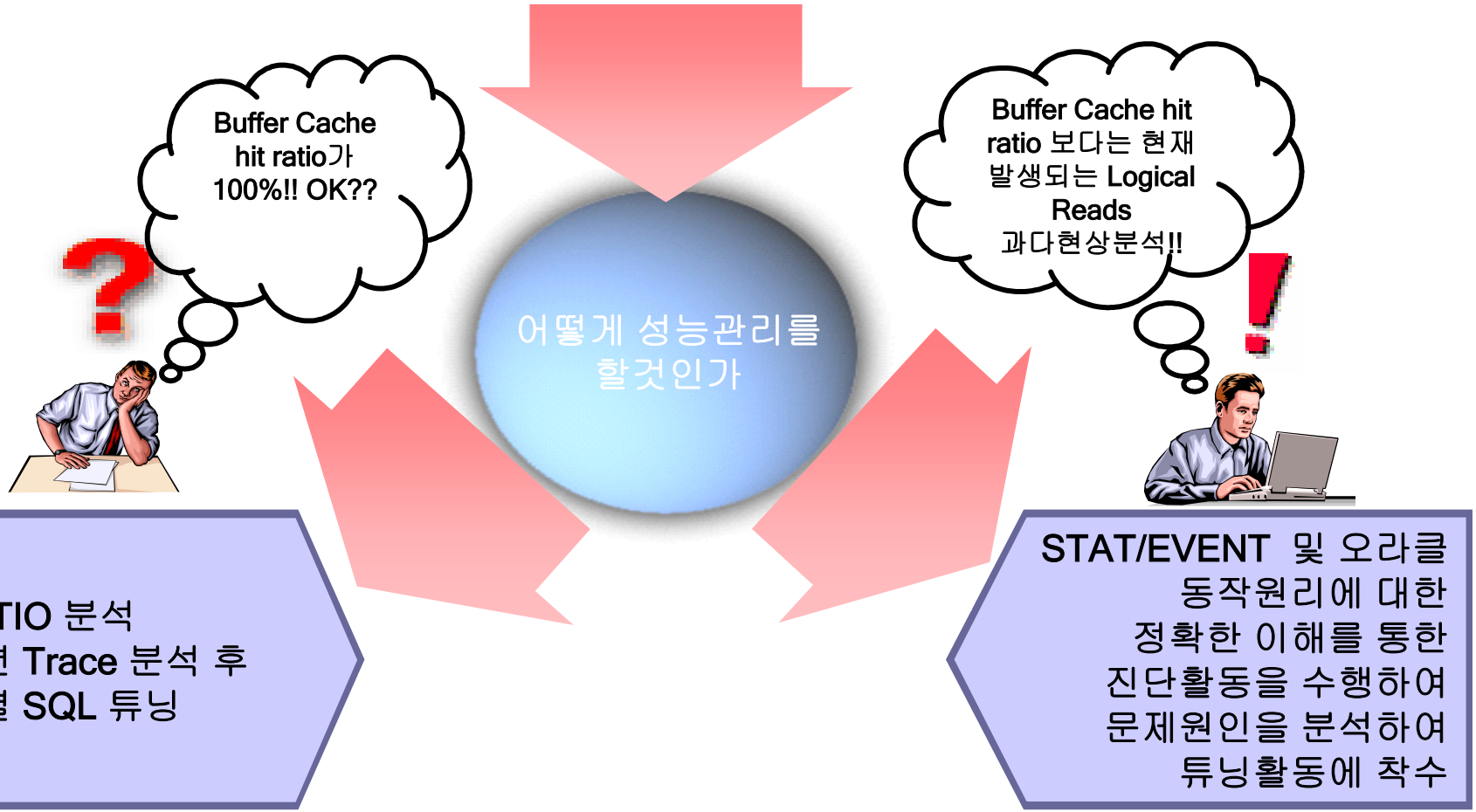
- STAT (Statistics)
  - 오라클 세션들이 ACTIVE 하게 일을 수행하면서 발생시키는 일량
  - 오라클 9.2.0.4 (264 개)
  - STAT CLASS

1	2	4	8	16	32/40	64	128
User	Redo	Enqueue	Cache	OS	PQ,RAC	SQL	DEBUG

- EVENT
  - 오라클 세션들이 일을 수행하면서 겪는 대기현상
  - 오라클 9.2.0.4 (401개)
  - 오라클 10.1.0 (808개)
  - 10g 부터 EVENT 를 상세하게 구분 (latch free, enqueue etc)
  - 10g 부터 WAIT CLASS 적용 (User I/O, System I/O, Commit etc)

II. STAT/EVENT 분석의 필요성

- VLDB 화에 따른 대량의 I/O 발생, DSS 시스템에서의 대량의 PQ 수행, OLPT 시스템에서 수천명의 사용자에 의한 동시 Transaction 과다에 따른 경합 현상 발생



목 차

I. STAT/EVENT 분석

I. Session Logical Reads

1. Logical Read 의 개요
2. Data Buffer Cache 개요
3. Logical Read 세부처리절차
4. LRU 알고리즘의 변천사
5. table fetch by rowid 와 상관관계 분석

II. Physical Reads

1. Physical Read 의 개요
2. Physical Read 시 발생되는 Event 분석
    1. db file sequential reads 분석
    2. db file scattered reads 분석
    3. buffer busy wait 분석
3. table scan blocks/rows gotten 분석
4. SORT_AREA_RETAINED_SIZE 의 의미

1. Logical Reads의 개요

- Logical Reads 의 사전적 의미
  - SGA 내의 Data Buffer Cache 메모리에서의 Block Read 작업
  - session logical reads = 'consistent gets' 와 'db block gets' 를 합한 수

- Logical Read 는 Physical Reads 보다 얼마나 빠른가?
  - Disk access latency = 0.01 초 (10 milliseconds)
  - Memoy access latency = 0.000 001 초 (1 microsecond)
  - Logical Reads 가 Physical Reads 보다 10,000 배 빠른가?

- 과도한 Logical Reads 발생시 대표적인 문제점 2가지는 ?

2. Data Buffer Cache 개요- 용어

- Hash buckets
  - dba (data block address) 와 class number (DB,sort,undo,etc) 의 hash 값에 의해 그룹핑된 data buffer header들의 리스트를 관리하는structure

- Hash chains
  - 하나의 hash bucket 에 안에 존재하는 buffer header들의 List

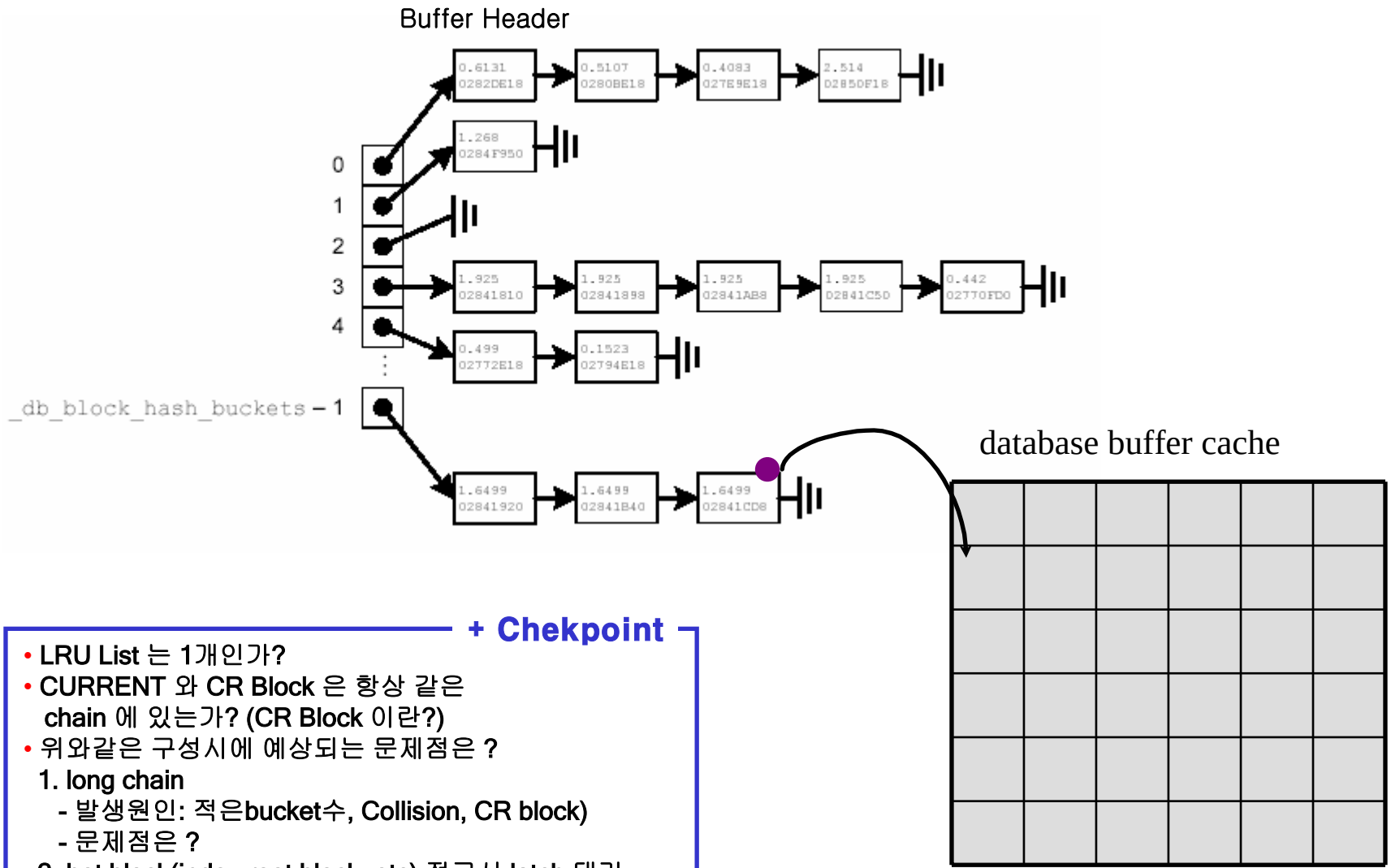
- LRUW (LRU write list 또는 dirty list)
  - dirty buffer 들을 관리하는 List

- LRU (least recently used list)
  - LRUW 에서 관리하는 buffer 를 제외한 나머지 buffer들을 관리하는 List
  - cache 된 buffer 는 LRUW 또는 LRU 중 1개에만 존재해야됨

- Data block reader = Foreground ORACLE process

- Data block writer = Background ORACLE process (DBWR)

2. Data Buffer Cache 개요- 구성



+ Chekpoint

- LRU List 는 1개인가?
- CURRENT 와 CR Block 은 항상 같은 chain 에 있는가? (CR Block 이란?)
- 위와같은 구성시에 예상되는 문제점은 ?

1. long chain
   - 발생원인: 적은bucket수, Collision, CR block)
   - 문제점은 ?
2. hot block(index root block..etc) 접근시 latch 대기
   - 9i 부터 shared latch 디자인 적용

2. Data Buffer Cache 개요

- Buffer Cache Block vs Bucket vs Cache buffers chains latch

	7.3~8.0.X	8i~10g
Hash bucket 수	Buffer수 / 4 (prime ?)	Buffer 수 * 2 (10g ≈ Buffer수*2)
Hash bucket 관련 Parameter	_db_block_hash_bucket	_db_block_hash_bucket
Cache buffers chains latch 수 (V$LATCH_CHILDREN)	≈ hash bucket 수	Buffer size 16M~1G 인 경우 1024개

+ Chekpoint

- **8i 부터 buffer 수보다도 많게 Bucket을 설정한 이유는?**
- **8i 부터 cache buffers chains latch 의 수가 일정한 이유는 ?**

3. Logical Read 세부처리 절차

- function LIO (*dba,class,mode,…)*

  # *dba* is the data block address (*file#, block#)*

  # *mode* is either 'consistent' or 'current'

  address = BUFFER_ADDRESS(dba,class,.); # (Hasing, Latch get..)

  if no *address* was found

  address = PIO(dba,class,…);

  update the LRU chain if necessary # (언제 필요한가?)

  if (*mode* is 'consistent')

  construct cr image if necessary, by clone and calling LIO undo

  increment 'cr' statistic in trace and 'consistent gets' stat in v$

  else (*mode* is 'current')

  increment 'cu' statistic in trace and 'db block gets' stat in v$

  parse the content of the block;

  return the relevant row source;

- Latch Get 알고리즘 및 과도한 cache buffers caches lath
- 대기현상에 대한 자세한 사항은 Part-2 에서 다룸

출처: *Why You Shoud Focus on LIOs Instead of PIOs* hotsos (http://www.hotsos.com)

4. LRU 알고리즘의 변천사

- 64bit Oracle 사용이 보편화 되면서 Very Big Buffer Cache 를 효율적으로 관리하기 위한 LRU 알고리즘의 도입이 필요하게 됨

Standard LRU 알고리즘

위험 요소및 해결방안은?

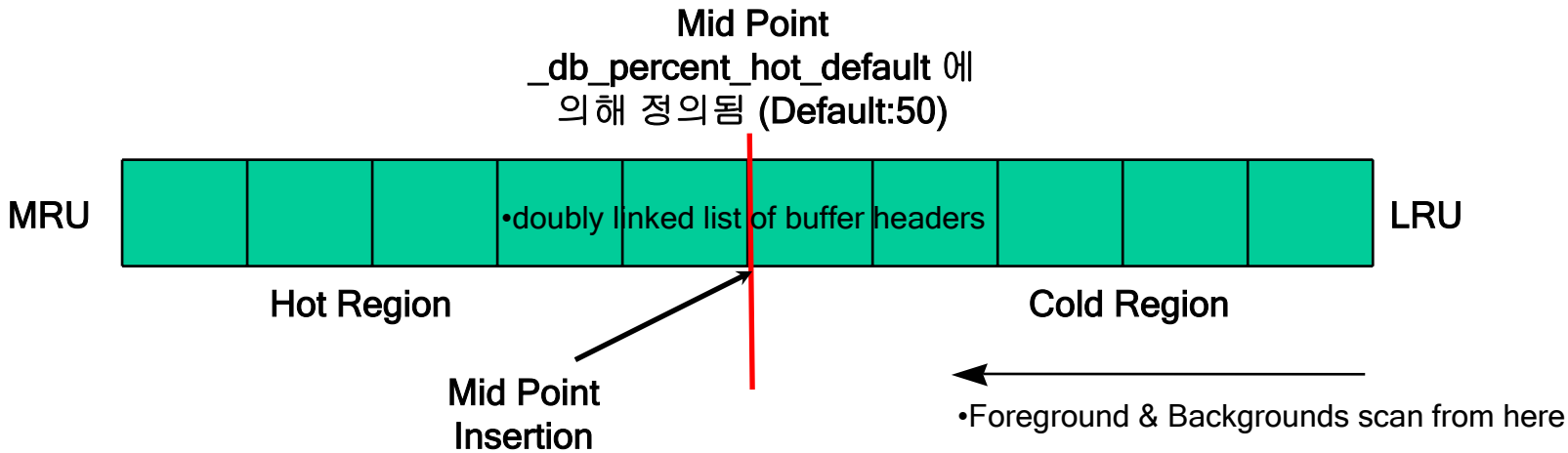
Modified LRU 알고리즘

위험 요소및 해결방안은?

Touch Count Based LRU 알고리즘

GOOD!!

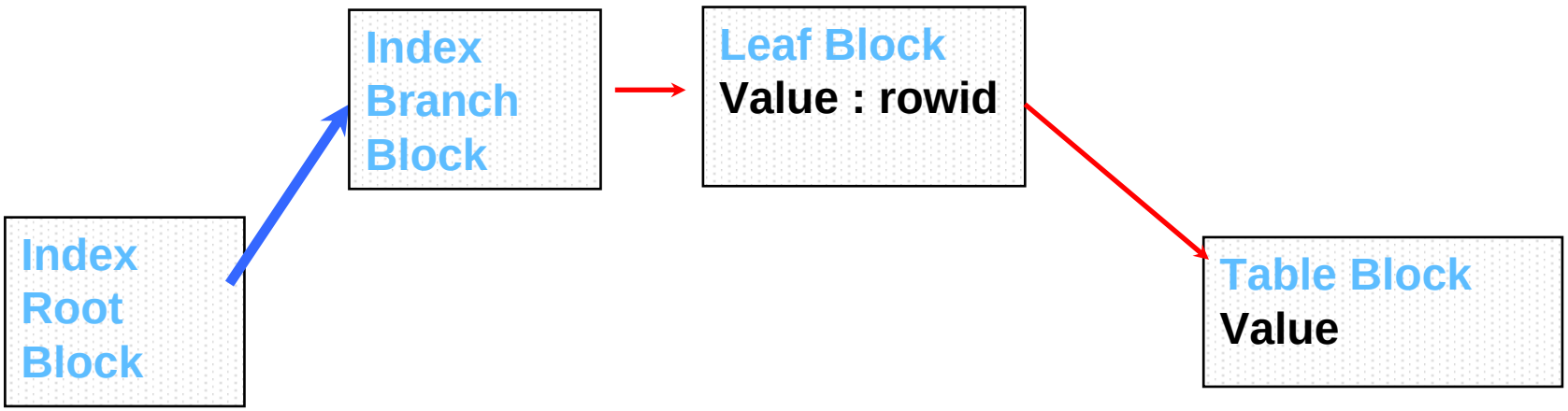
4. LRU 알고리즘의 변천사-Touch Count



- Buffer 마다 Touch Count 관리 - SYS.X$BH.TCH
- Buffer Touch 시 마다 Touch Count를 증가시키지 않음 (Why?)
  - _db_aging_touch_time (default : 3 초)
- 따라서 Latch 획득없이 Buffer의 Touch Count를 증가시키는것이 가능함
- Free buffer Search 시 Touch Count 가 2이상인 Buffer를 MRU End 로 이동한후 Mid Point 위치 조정
  - MRU End 이동대상 Buffer 선정기준(_db_aging_hot_criteria : default 2)
  - MRU End 이동 후 Touch Count 0으로 초기화
  - Mid Point 위치 조정에 따라 Cold region 으로 이동되는 Buffer의 Touch count를 1로 Reset (_db_aging_cool_count : default 1)

5. table fetch by rowid 와 상관관계 분석

- table fetch by rowid : rowid 를 이용하여 fetch 된 row 수



- Index Scan 시에 발생되는 Logical Read는 Index Scan에 대한 Logical Read와 테이블에 대한 Random access로 구분됨
- 위와같은 경우 Logical Reads 와 Table fetch by rowid 수치는 ?
- Table fetch by rowid >> Logical Reads 인 경우는 인덱스 분포도가 않좋은 경우
- Table fetch by rowid << Logical Reads 인 경우는 결합인덱스 구성시에 Index 필터로 처리되는 경우에 많이 발생됨 (결합인덱스 순서조정또는 다른 Index 사용)

1. Physical Reads의 개요

- Physical Reads 의 사전적 의미
  - Oracle Datafile 의 Block 을 Buffer Cache 로 읽어들이는 활동
  - single block read 와 multi block read 로 구분됨

- Physical Reads/Writes Direct 란?
  - Buffer Cache를 거치지 않고 PGA 와 Datafile 블록사이의 발생되는 I/O 작업
  - SORT , HASH Operation 을 수행할 때 발생됨
  - Physical Reads = physical reads - physical reads direct/(lob)

2. Physical Reads시 발생되는 Event 분석

1) db file sequential reads 분석
- disk i/o 를 위해 O/S 에 single block i/o 를 요청한후에 끝나기를 기다리는 event
- p1 : file#
- p2 : block#
- p3 : blocks (오라클 8이상부터는 모두 1)

- Index Scan 시에 db file seqential reads 가 많이 높다면 문제가 되나?

- 만일 문제가 된다면 db file sequential reads 의 주 발생원인은 ?
  A) Index 분포도 불량
  B) Where절 조건에 결합인덱스의 선행컬럼만 존재
  C) Clustering Factor 불량 (Worst Case: rows 만큼 block read)
  D) 주기적인 Data purge 작업에 의해 index blevel 증가 (90만건에 blevel 3)

+ Chekpoint

- Index Scan 시에 항상 db file sequential reads 가 발생할까?

2. Physical Reads시 발생되는 Event 분석

2) db file scattered reads 분석
- disk i/o 를 위해 O/S 에 multi block i/o 를 요청한후에 끝나기를 기다리는 event
- p1 : file#
- p2 : block#
- p3 : blocks (항상 db_file_multiblock_read_count 는 아님. Why?)

- db file scattered reads 는 맞지 않는 표현이며, scattered 의 의미는 읽어들인 multi block 을 buffer cache 의 임의의 공간에 흩뿌린다는 의미임
(동시에 여러 개의 latch 를 획득한후 loading 하는것으로 추정됨)

Quiz> 64Kb Extent 들로 구성된 1G 테이블과, 동일한 데이터를 가지고 1Mb Extent 들로 구성된 1G 테이블에 대한 Full Scan 시에 어느것이 효율이 좋을까?
만일 후자가 좋게 나왔다면 원인은 무었일까? 크게 3가지 원인이 있을수 있음
(해당 테이블들은 인덱스를 통한 Read 작업도 빈번하게 발생한다고 가정함)
(db_block_size=4Kb, db_file_multiblock_read_count=8)

+ Chekpoint

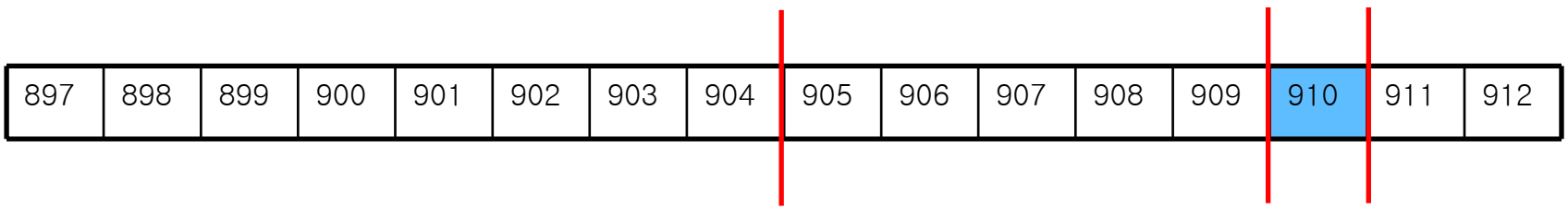
- Table Full Scan 시에 항상 db file scattered reads 가 발생할까?

2. Physical Reads시 발생되는 Event 분석

TEST Sample >>
현재 BLOCK ID 910 은 메모리에 있음 (몇번의 Multi Block I/O Call 이 발생할까?)

```
WAIT #1: nam='db file scattered read' ela= 605 p1=8 p2=897 p3=8
WAIT #1: nam='db file scattered read' ela= 342 p1=8 p2=905 p3=5
WAIT #1: nam='db file scattered read' ela= 150 p1=8 p2=911 p3=2
WAIT #1: nam='db file scattered read' ela= 473 p1=8 p2=913 p3=8
```

BLOCK_ID	BLOCKS
897	16
913	16
929	16



2. Physical Reads시 발생되는 Event 분석

3) buffer busy waits
- 말그대로 buffer가 busy 하기때문에 대기할때 발생되는 event
- p1 : file#
- p2 : block#
- p3 : reason code

- 대표적인 reason code (130) 설명
  : 읽고자하는 block 이 다른 세션에 의해 disk 로 부터 buffer cache로 읽어들이기 때문에 buffer가 busy 한 경우 .
  Tuning Point >> physical reads를 제거

- 대표적인 reason code (210,220) 설명
  : DML을 수행하기 위해 해당 buffer에 buffer lock 을 걸려고할때,다른 세션이 buffer lock 을 점유하고 있으므로 buffer mode가 incompatible 상태여서 buffer가 busy 한 경우
  Tuning Point >> Transaction을 여러block 으로 최대한 분산 시킴
  A. Partitioning
  B. Freelists 증가
  C. small block size(2Kb) 로 테이블 rebuild (9i ~)

3. table scan blocks/rows gotten 분석

- table scan blocks gotten : scan operation 시에 읽어들인 block 수
- table scan rows gotten : scan operation 시에 읽어들이 row 수

• table scan blocks gotten ≈ table scan rows gotten 인 경우는 1개의 block 안에 1~2개의 rows만 존재하는 경우임 (Table 재생성 필요)

4. SORT_AREA_RETAINED_SIZE 의 의미

- SORT_AREA_RETAINED_SIZE는 Sort 에 사용된 UGA공간을 얼마만큼만 남겨두고 PGA로 반납할지를 결정하는 파라메터임.

- SORT_AREA_SIZE=10Mb
- SORT_AREA_RETAINED_SIZE=1Mb
- 위와 같이 파라메터를 설정한 후에 10Mb 메모리 Sort 를 수행할 경우 어떠한 일이 발생하게 될까?

- Dedicated 방식으로 오라클을 운영할 경우 위와 같은 세팅이 O/S 메모리 절약에 도움이 될까?

목 차

III. 분석 사례

DEMO 를 통해 진행

- Case#1~Case#6

목 차

IV. 질의 응답

http://www.ex-em.com
support@ex-em.com