

Андрей Криушин

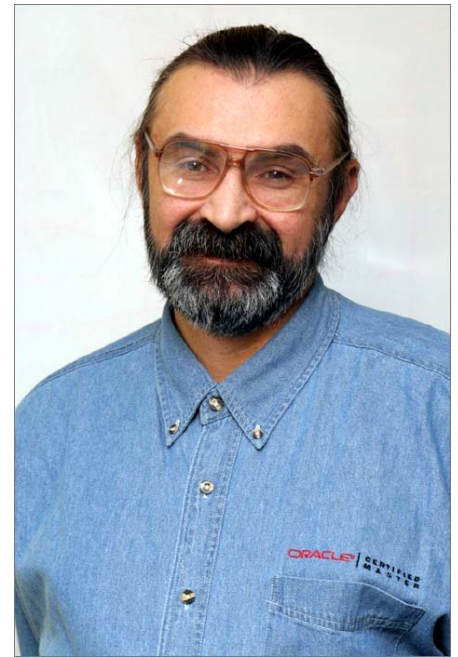
Эксперт по программным продуктам Oracle
компании ЗАО «РДТЕХ»

Директор Центра Компетенции Oracle
по направлению "Grid & Consolidation"

Andrey.Kriushin@rdtex.ru

+7 903 593 2408

<http://www.rdtex.ru>



ORACLE®

Certified Master

Latches, enqueues & Oracle RAC Защёлки, блокировки и Oracle RAC

Oracle RAC seminar DD4D, 2008,
October 22-23, Moscow

Темы презентации

- О многопользовательских системах
- О **транзакционных** многопользовательских системах
- Модели служебных структур сервера Oracle RDBMS (средства обеспечения непротиворечивости результатов в многопользовательских система)
- Конкуренция за ресурсы в Oracle RDBMS – single instance case
- Конкуренция за ресурсы в Oracle RDBMS – multiple instances (+ RAC option)

О многопользовательских системах

- В действительно «однопользовательских» системах не приходится думать о взаимодействии/конкуренции с другими пользователями / процессами.
- В многопользовательских системах ОС (или **VOS**) создаёт **иллюзию**, что каждый потребитель ресурсов системы (CPU, IO, network, swap) – единственный в этой системе.
- В тех редких (?) случаях, когда разным пользователям/процессам требуется одновременный доступ к общему ресурсу, ОС/VOS использует **механизмы сериализации доступа** к ресурсу.
- Механизм сериализации задействуется всегда (т.е. заранее), даже если в какой-то момент в системе присутствует только один «потребитель ресурса». Иначе (неожиданно появится второй), может возникнуть ситуация с недетерминированным результатом.

Pessimistic locking at low level !!!

О многопользовательских системах

- Механизмы сериализации
 - == **ВСЕГДА** накладные расходы
 - (непродуктивное избыточное потребление ресурсов)
 - == плата за «виртуальную» *многопользовательскость*
- **VOS** (не путать с Veritas-Oracle-Sun initiative)
 - **Virtual Operating System (Oracle RDBMS kernel layer)**
 - Кеширование (buffer cache, row cache, library cache)
 - Служебные структуры для обслуживания кешей (поиск, замена, ...)
 - Двойственная роль кешей в Oracle RDBMS
 - собственно кеш и средство установки большинства блокировок – как транзакционных (TX, TM), так и «более других»**

О транзакционных многopользовательских системах

- **Дополнительные** механизмы сериализации для соблюдения принципов транзакционности

A

Atomicity

Атомарность

C

Consistency

Согласованность

I

Isolation

Изолированность

D

Durability

Продолжительность

A C I D

Модель служебных структур сервера Oracle RDBMS

- Задачи
 - Обслуживание кешей (быстрый поиск, сериализация доступа, управление списками, etc)
 - Обслуживание транзакций в соответствии с **ACID**
 - Обслуживание других структур базы данных (доступ к управляющему файлу, восстановление файлов данных ...)

Модель служебных структур сервера Oracle RDBMS

- Три основных механизма

- Защёлки(+ с 10g mutex)

- Pin'ы (а как же library cache lock/pin ?)

ПРИМЕЧАНИЕ: это краткосрочные нетранзакционные блокировки.

Режимы блокирования, как правило,

X, S, N == eXclusive, Shared, Null (None)

- Enqueue/Locks (продолжительные блокировки с более «богатым» выбором режимов, матрицей совместимости и строгим FIFO)

Модель служебных структур сервера Oracle RDBMS

- Реализация

- Защёлки (Latches) – ячейка памяти (+ небольшая служебная структура для восстановления в результате сбоя серверного процесса) и изменение «атомарными инструкциями CPU» - Test & Set, Swap & Compare ... Со сбросом процессорных кешей всех CPU системы
 - Exclusive only (до 9i)
 - Shared (9i+)
 - Mutex (10g+)
 - Racing – кто кого обгонит. Очередность не соблюдается. Если первая попытка неудачна, регистрируется промах (MISS), далее выполняется N повторных циклов (_SPIN_COUNT), после чего регистрируется SLEEP. Повторные попытки после первого SLEEP уже не увеличивают MISS, а также производятся через интервалы времени по exponential backoff механизму
- Pin – после преодоления барьера «latch», можно кратковременно заблокировать одну из структур, защищаемых «latch». Сама защелка после установки Pin освобождается. Если другой процесс, преодолев барьер latch, добирается до той же структуры, а Pin ещё не снят – получается простая очередь FIFO (пример – buffer busy как ожидание «неснятого Pin'a»)

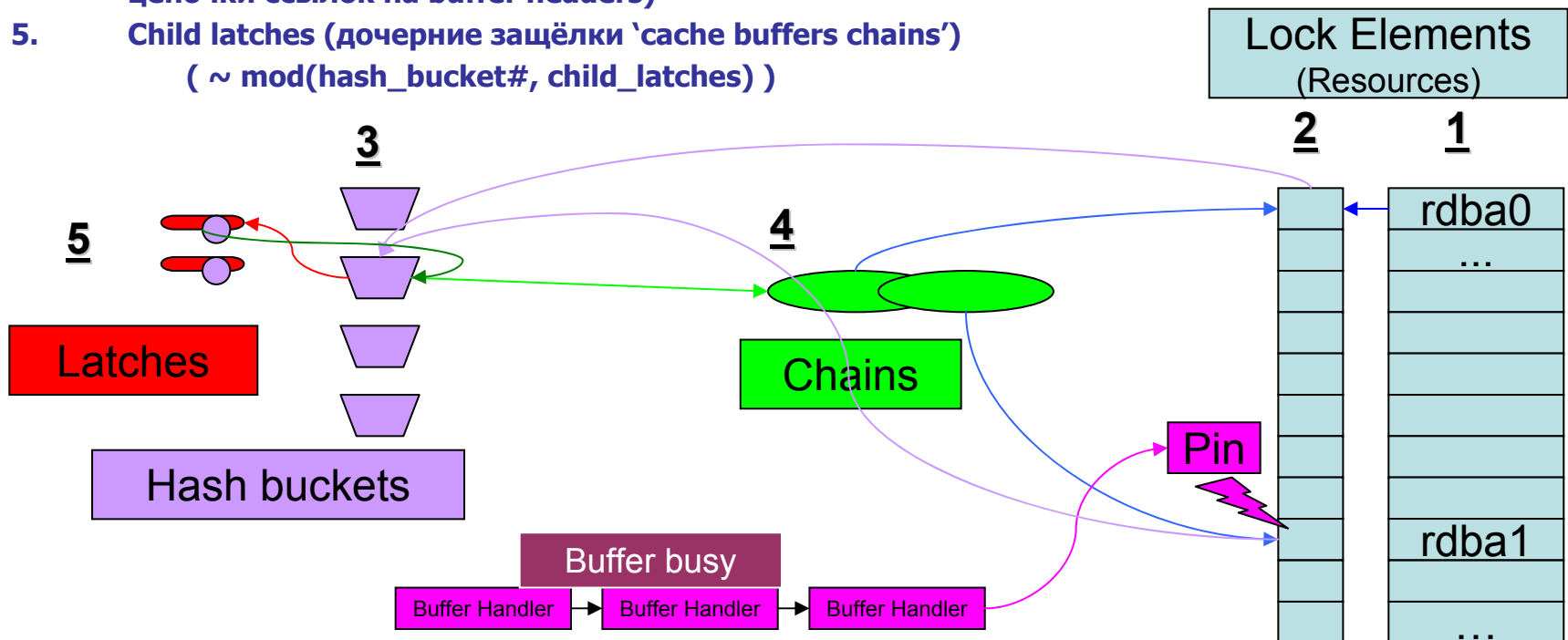
Always pessimistic locking at low level !!!

- Блокировки (Enqueue/Lock) – механизм очередей, множество состояний, очереди на получение/преобразование. Список режимов разнообразнее (X, S, SX, SSX, N).

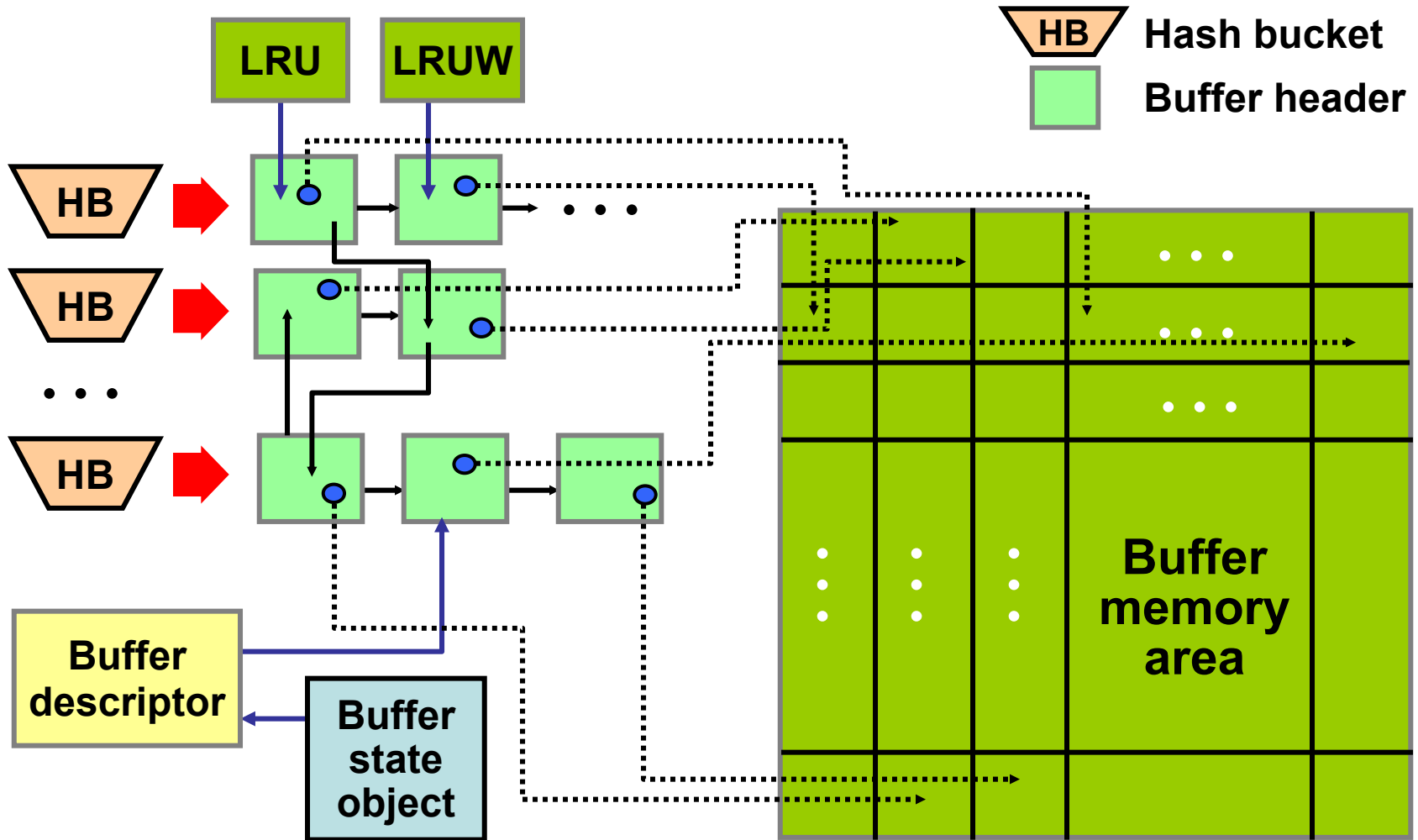
См. далее

Модель служебных структур сервера Oracle RDBMS

1. RDBA (Relative DataBlock Address – из словаря данных), блок считывается в буфер
2. Buffer header (заголовок буфера – RDBA, TS#, статус, элементы списков, ...). **Каждому буферу – по хедеру ☺**
3. Hash bucket ($\sim \text{mod}(\text{RDBA}, \text{hash_buckets})$)
4. Cache buffers chains (каждый элемент массива hash buckets является началом списка цепочки ссылок на buffer headers)
5. Child latches (дочерние защёлки 'cache buffers chains')
($\sim \text{mod}(\text{hash_bucket\#}, \text{child_latches})$)

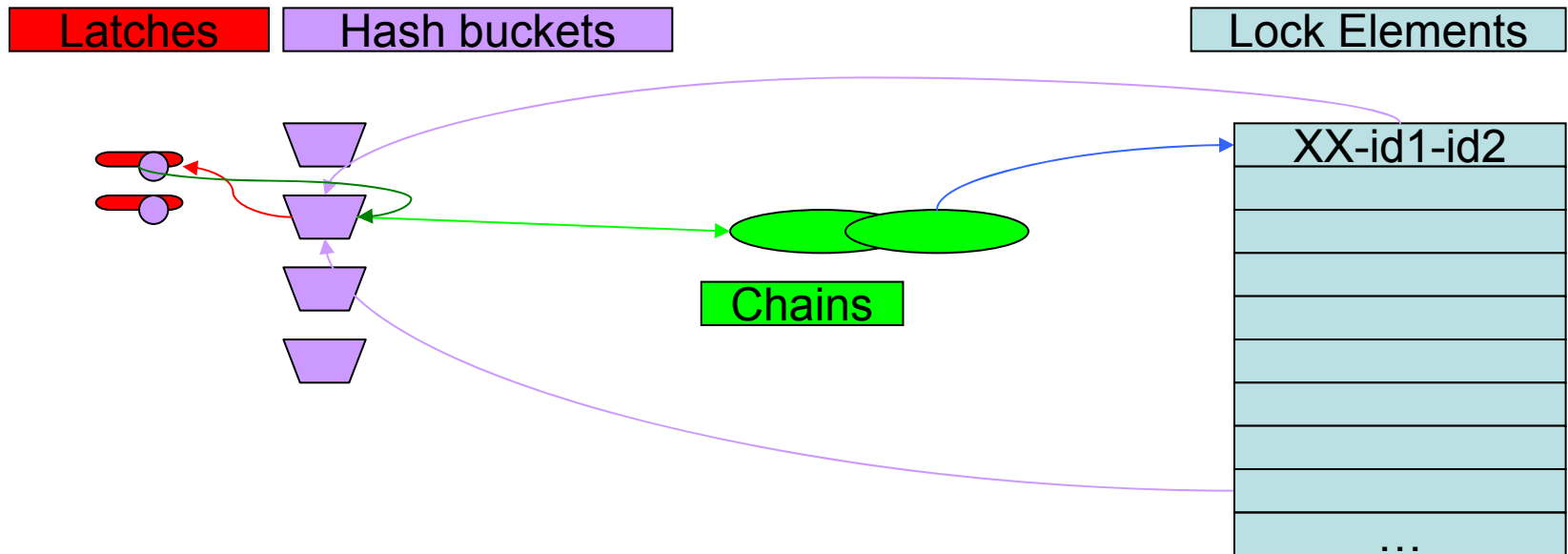


Overview of Buffer Cache



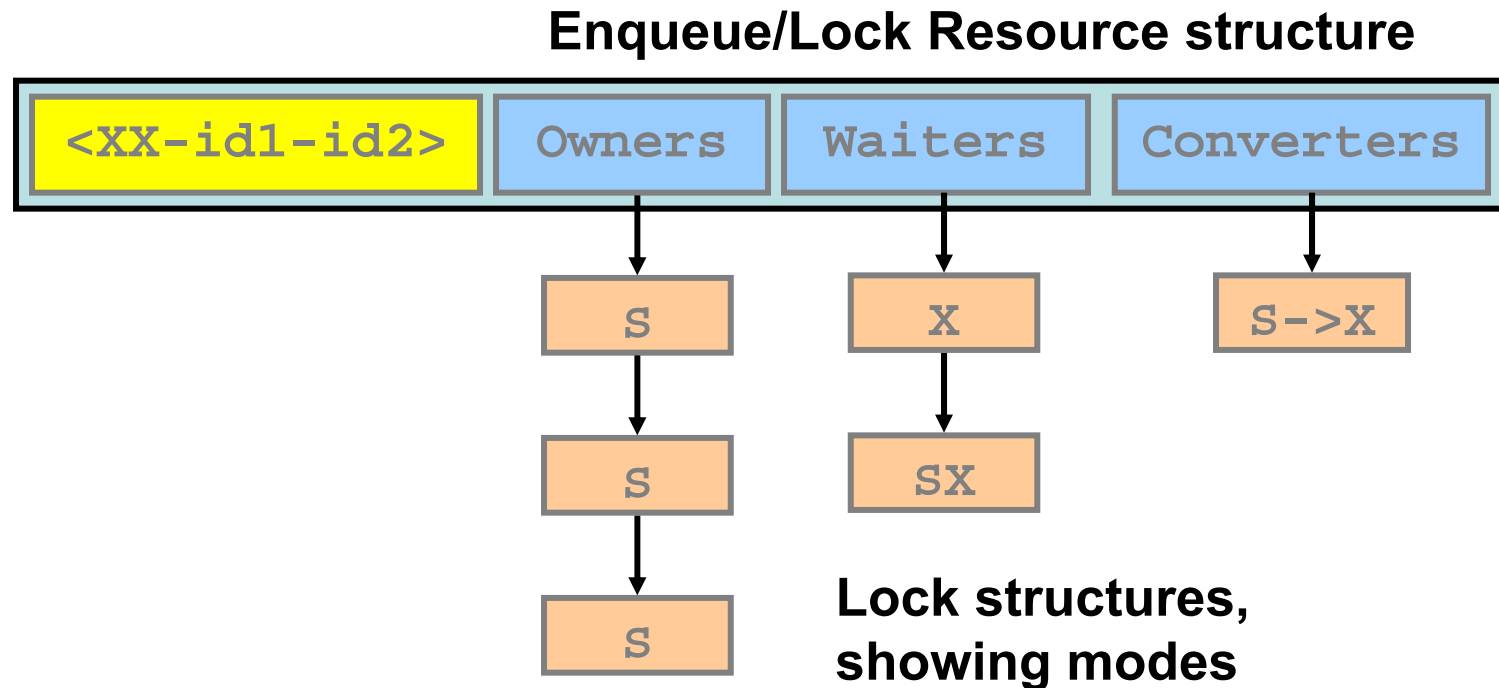
Модель служебных структур сервера Oracle RDBMS

V\$GES_RESOURCE

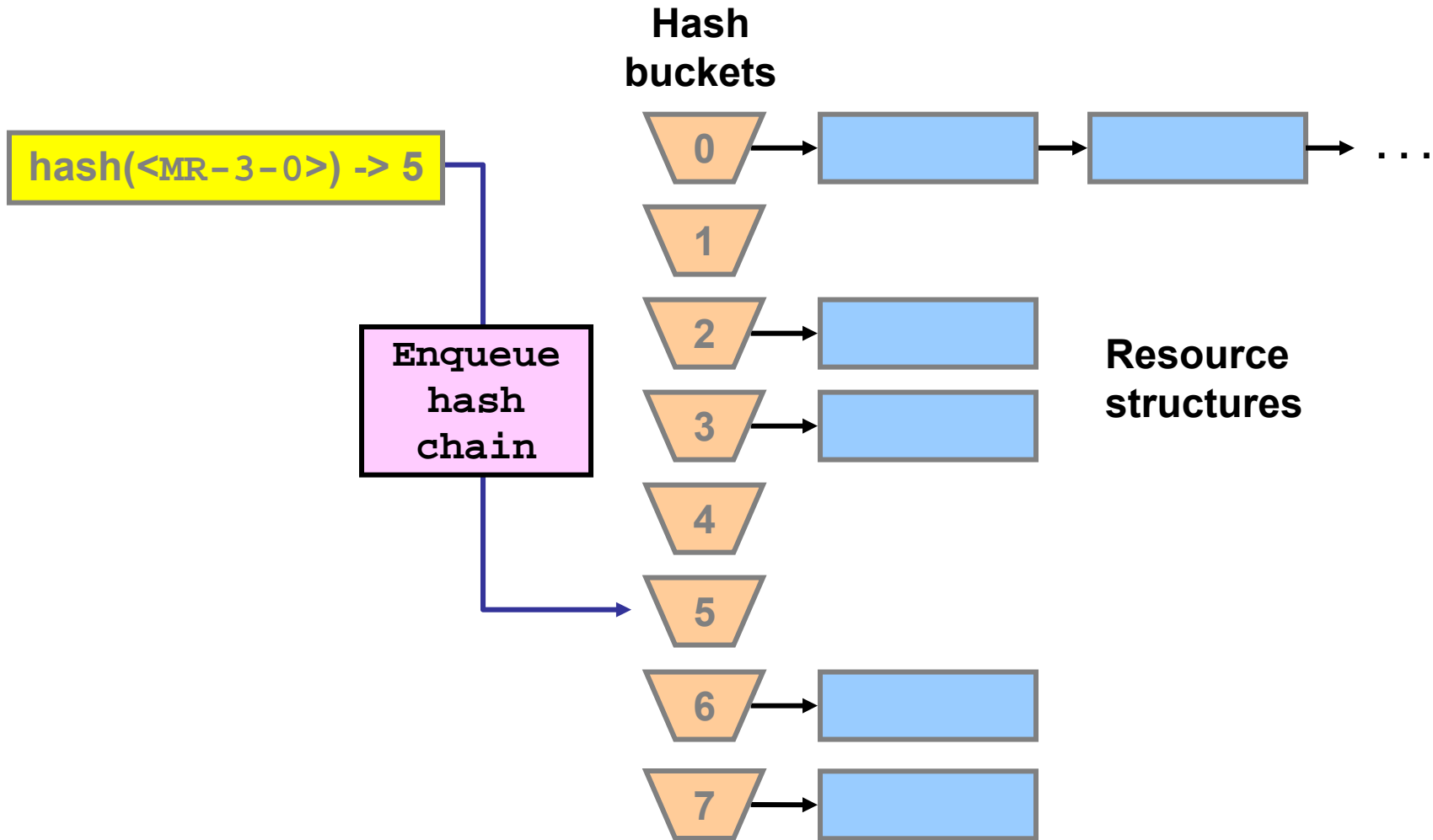


Resource and Lock Structures

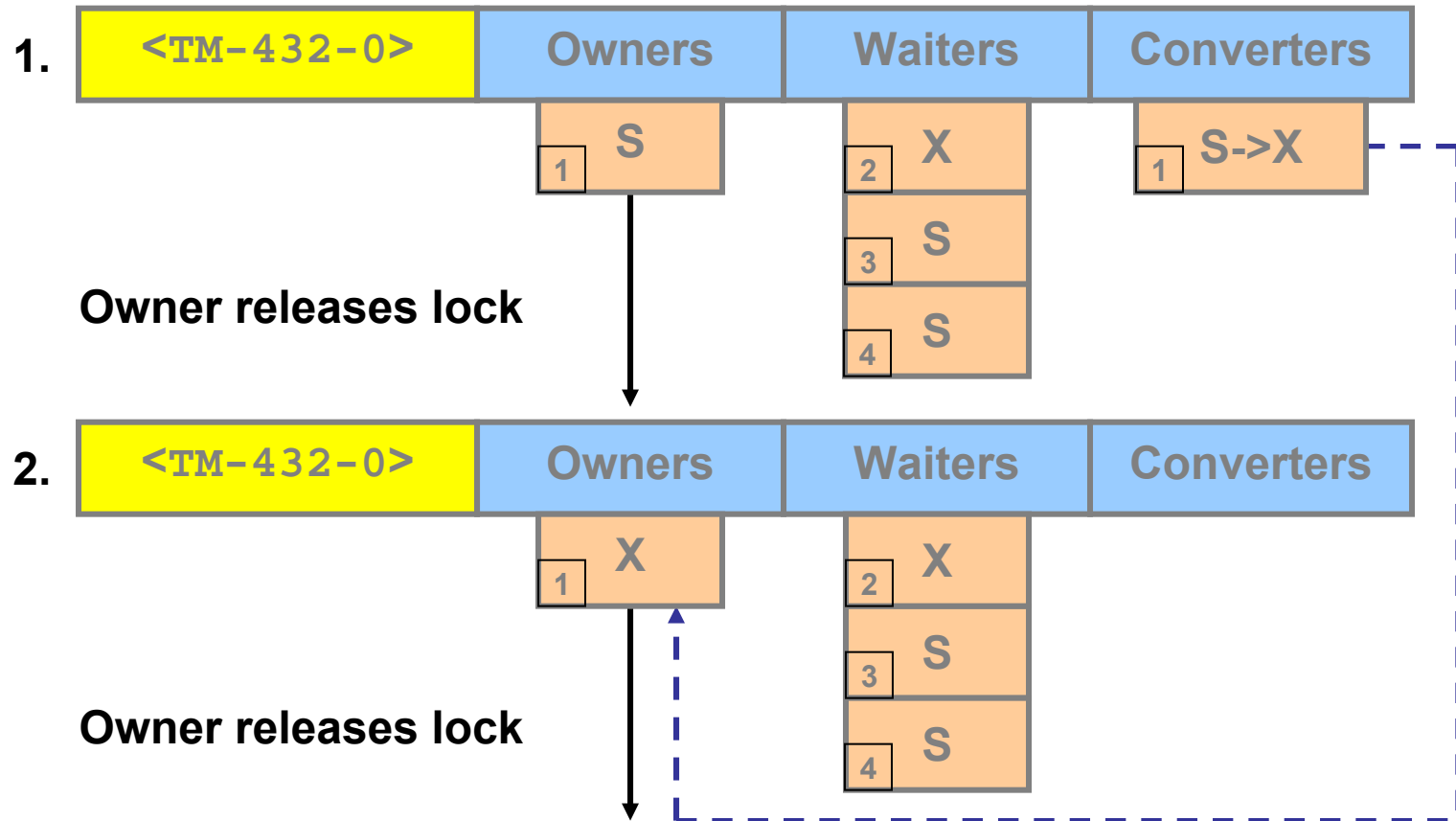
(O9iAIT – Oracle University Course code)



Hashing and Latching



Enqueue Operations Example



Wait Event: Enqueue

```
SQL> select ksqssttyp TYPE
       2 ,      ksqsstsgt GETS
       3 ,      ksqsstwat WAITS
       4 from    x$ksqsst
       5 where   ksqsstsgt > 0
       6 order  by WAITS desc;
```

TYPE	GETS	WAITS
TX	24208	3251
US	24173	2885
TM	981	115
CF	55	27
ST	32	26
...		

Single Instance Summary

- В Single Instance задействовано множество механизмов и структур, обеспечивающих прозрачность в многопользовательской среде

- В большинстве случаев при разработке приложений они игнорируются, поскольку (за исключением аномалий) не влияют на время отклика, т.е. на «производительность»

Cluster resources

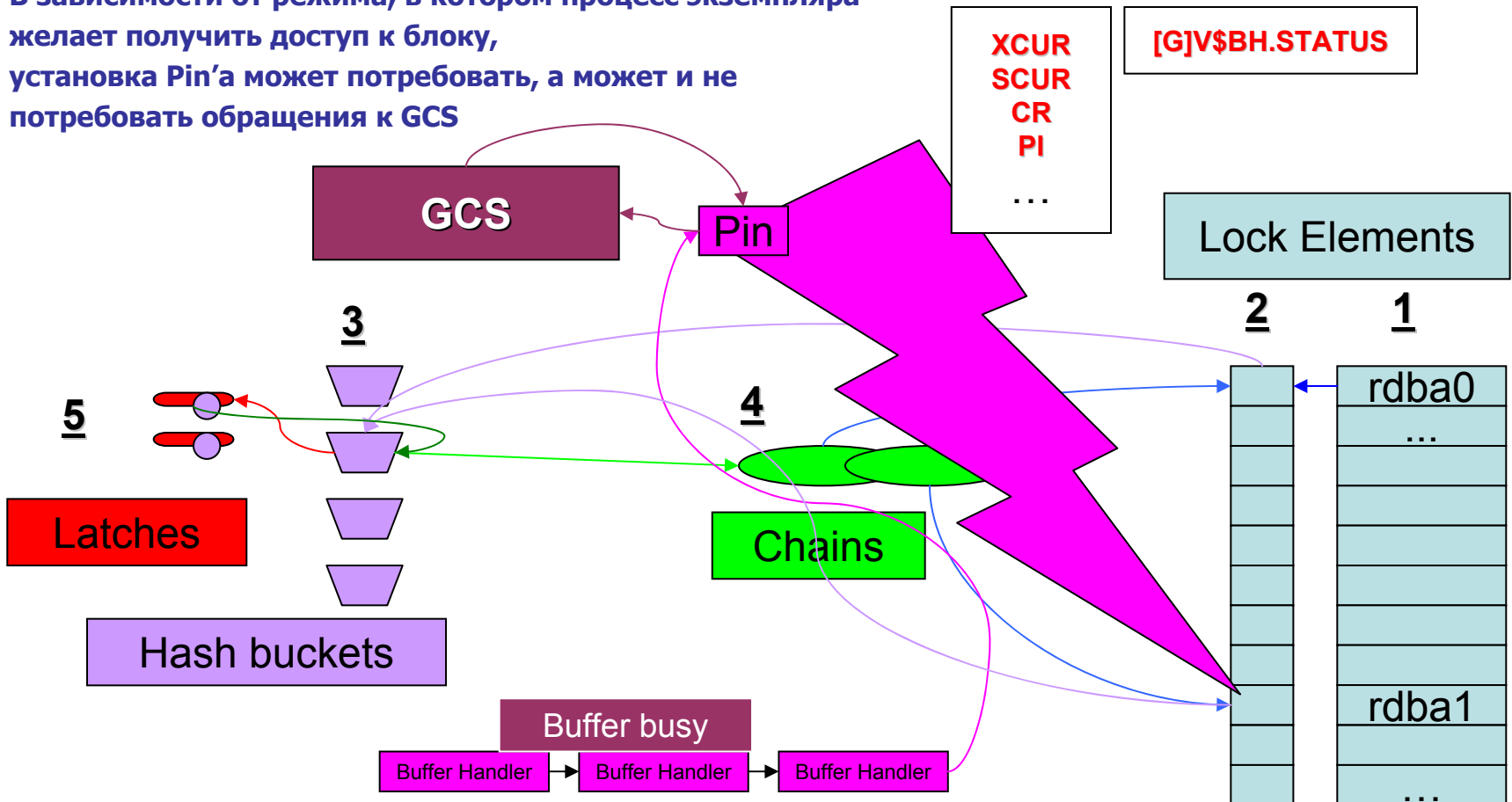
- Многие из **ресурсов** одиночного экземпляра (single instance) в кластере становятся «глобальными»
 - Блоки (Pin остаётся в локальном кеше, но прежде чем его установить, может потребоваться согласование с мастером «ресурса»)
 - Все enqueue/lock (см.выше. Чтобы стать в очередь, требуется изменение глобального ресурса, которое надо согласовать с мастером «ресурса»)
 - L[x] (Lock), N[x] (piN) для курсоров (в RAC эти ресурсы «неожиданно» становятся глобальными – DROP/ALTER/TRUNCATE/... должны же как-то влиять на курсоры... которые в разных экземплярах)
 - ...
- Глобальные ресурсы
 - Управляются схожим образом, требуют дополнительных структур в SGA каждого экземпляра, а также механизмов глобальной (inter-instance) синхронизации доступа и обеспечивающих их фоновых процессов
 - Для BL (GCS, Cache Fusion) некоторые механизмы упрощены из-за более простой схемы блокирования --- X (eXclusive), S (Shared), N (Null)
и поэтому выделены в собственную подструктуру, которой управляет GCS
 - Для остальных ресурсов работает GES (Global Enqueue Services)

Cluster resources

- Однако это **НЕ** означает, что при доступе к глобальному ресурсу **ВСЕГДА** требуется обращение к мастеру ресурса – т.е. к глобальному, иногда расположенному на другом экземпляре «координатору доступа к ресурсу».
- Статус ресурса, однажды востребованного каким-либо экземпляром, «кешируется» в локальном экземпляре (в т.ч. не мастере) – shadow lock,
- Если повторный запрос с этого экземпляра к такому «кешированному» ресурсу не требует изменения его статуса/режима, то действие происходит абсолютно также, как в single instance.

Cluster resources

В зависимости от режима, в котором процесс экземпляра желает получить доступ к блоку, установка Pin'а может потребовать, а может и не потребовать обращения к GCS



Андрей Криушин

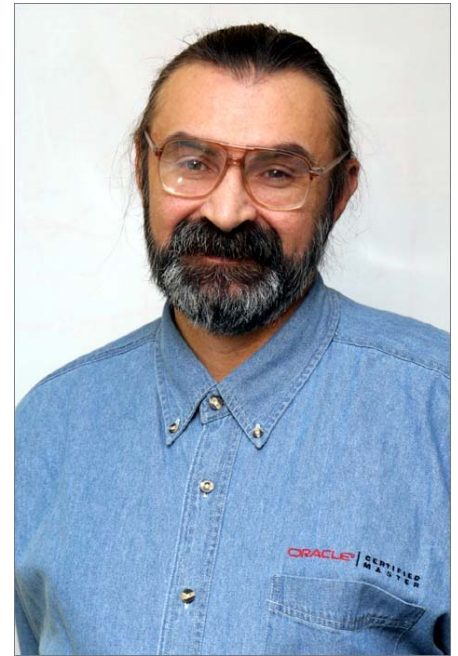
Эксперт по программным продуктам Oracle
компании ЗАО «РДТЕХ»

Директор Центра Компетенции Oracle
по направлению "Grid & Consolidation"

Andrey.Kriushin@rdtex.ru

+7 903 593 2408

<http://www.rdtex.ru>



ORACLE®
Certified Master

Questions & Answers