Implementing a High-Performance Trading Database With Oracle Rdb

Jeffrey S. Jalbert,
Thomas H. Musson,
Keith W. Hare

JCC Consulting, Inc.

Abstract

Trading systems are generally characterized by very high transaction rates, real-time process models and TP monitors. In general these systems are carefully hand crafted to provide a tight design with carefully controlled queries with minimum indexing, especially sorted indexes. Seldom is a trading application data store constructed in a relational database

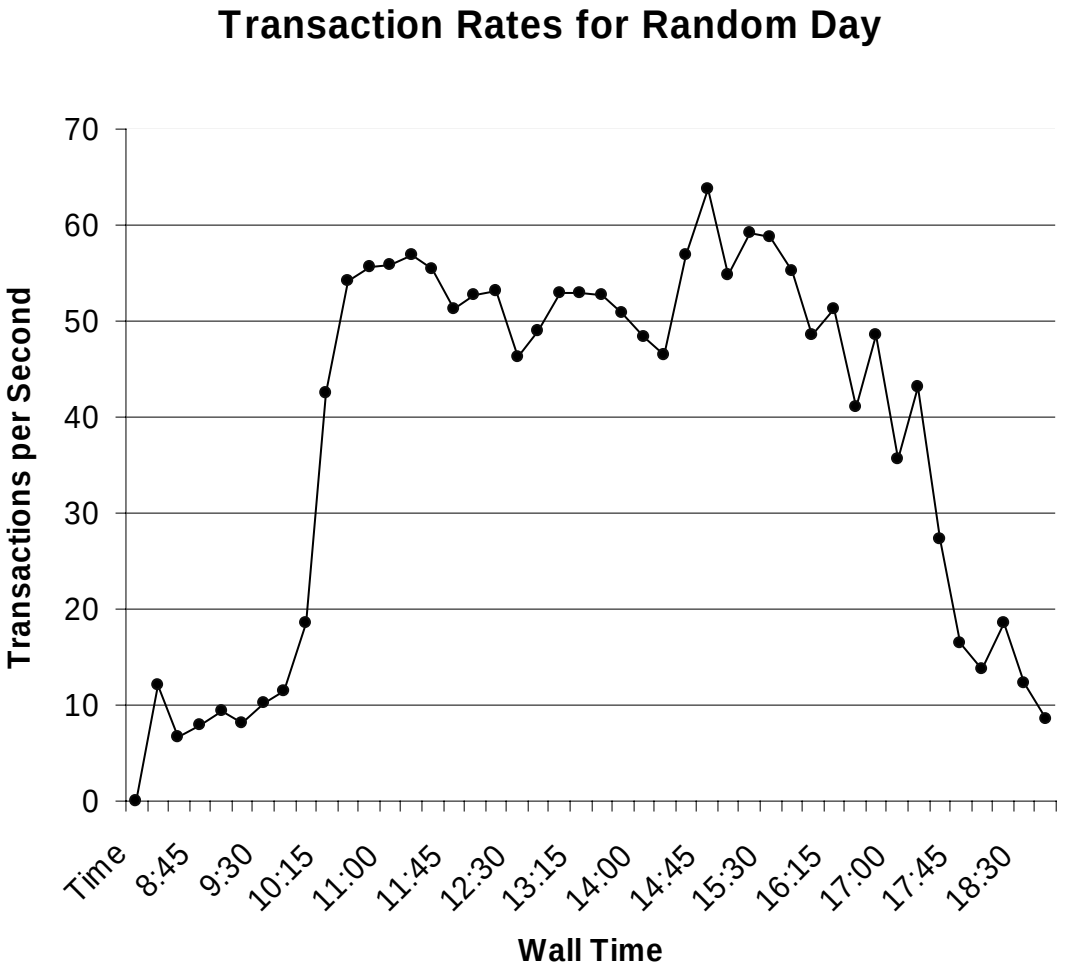
This presentation will describe the design of a trading/clearing database in Rdb. The remarkable thing about this database is that it also supports a large range of relatively unconstrained queries and therefore has no hashed indexes and a voluminous number of sorted indexes. For Rdb, this database is unusual in that read-only transactions are *disabled.* Currently in production with one-half of its projected workload, this database generates over 4 Gb of AIJ per day with over 1,250,000 transactions.

Problem Description

- Trading systems generally are partitioned into two major components
- Initial trade processing. This is generally characterized by
  - Very high transaction rates
  - Real time designs
    - Initial trade entry
    - Subsequent steps in processing
  - Often done with “memory resident” data structures and ad-hoc programming
- Back office systems to perform clearing functions

Problem Description

- Initial estimate for transaction rate is for the first ½ of the application is 1,000,000 database transactions per trading day
  - Resulting from 40,000+ trades
  - Determined by analyzing the process model
- Approximately ½ of these trades will occur in two 1-hour bursts, one in the morning and one in the afternoon
  - 250,000 database transactions per hour
  - About 69 transactions per second
- Many of these transactions are very complicated, nothing remotely similar to TPC-A



Data obtained via RMU and averaged over 15 minute samples

Problem Description

- The second component of a trading system is the back-office functions totaling up who ended up with what and who owes what to whom. Also includes risk analysis if one holds options or futures
  - Characterized by computationally intense functions
  - Generally access “lots” of data
- Positions analysis is especially important if one wants to make mid-day analysis of risk

Problem Description

- Trading systems must be extremely reliable
  - Huge volumes of money involved
- Must have the ability to restart on backup system should primary fail with minimal latency.
  - Target goal is 5 to 10 minutes, *maximum*
- Backup system in remote computer room

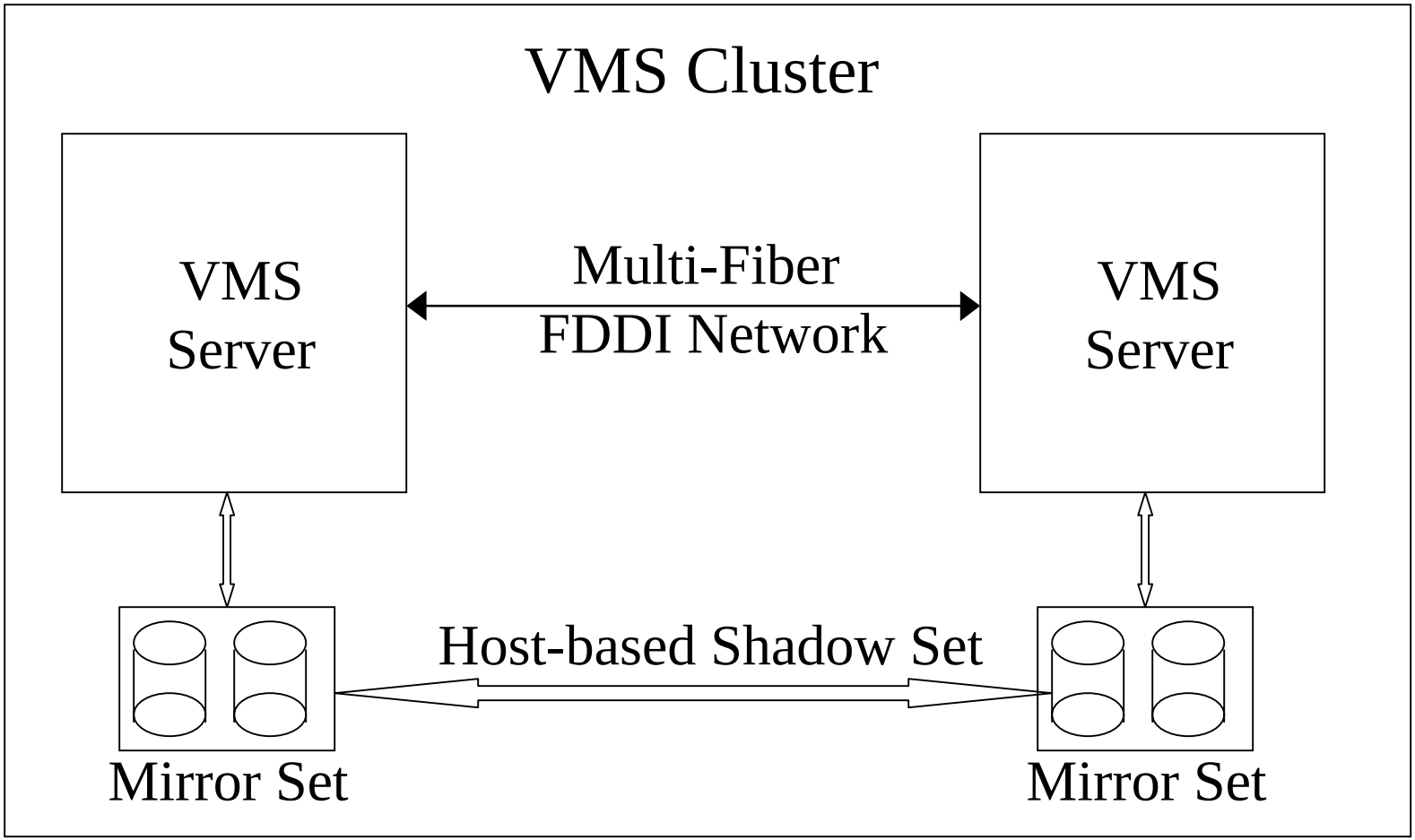
Problem Description

- Large numbers of concurrent users (600-1,000)
  - Traders and their assistants
  - Back-office personnel
- Required sub-second response time to interactive queries
- Generally implemented with transaction processing monitor
- The brave new world of PCs introduces the GUI concept of querying on anything!
  - Nobody could predict how users would respond to new opportunities
  - Requires large number of indexes on very active tables
  - Some indexes dropped nightly during database purge

Problem Description

- Application was being built by two partners
- One was a traditional MVS/DB2 shop
- The other was targeting the VMS/Rdb environment

Initial Production Hardware Solution



Additional VMS Servers

- QA system which duplicates a production server
- AXP 2100 development machine
- Numerous PCs etc.

Initial Production VMS Servers

- Dual processor TurboLasers
- “Slow” chips (relatively)
- 2 Gb memory
- Very high performance I/O subsystem based on HSJ Controllers
- Additional “small” 2100-class quorum/mount machine at each site
  - Keeps the host based shadowing going if one of the production class servers has to be shut down

Initial Storage Configuration

- HSJ disk controllers
  - Lots of writeback cache
- All disks were mirrored and shadowed for availability, 4-Gb drives
- 20 virtual database volumes to handle anticipated I/O demands
- 3 AIJ volumes
- Work & user volumes
- About 30 total virtual disks [all mirrored & shadowed for a total of about 120 spindles]

Network Configuration

- Basic network transport is TCP/IP
  - At fail-over DNS switches IP address of service for primary node
- DECnet is available for incidental convenience
- LAN network for communications to client PCs

Client Configuration

- Client machines run Windows NT
- Pipe to VMS server is proprietary based on the tool developer
- Clients also use TCP/IP broadcast to maintain certain local data
  - Server can supply it on demand

Software Context

- Main development tool is IEF from Texas Instruments
  - Later sold to Sterling Software & renamed CoolGen
  - Creates all code in both client & server
    - Can generate C or COBOL. Application uses C
  - Creates ACMS task definitions
- Can generate code to multiple database/OS platforms
  - Rdb/VMS/ACMS
  - DB2/MVS/CICS

Data Model

- Data model is highly normalized
- And also very complex
  - Large numbers of joins and sub queries
  - Longest example is a 4-foot query [printed using normal fonts!]
- At some points designers became confused between normalization and synthetic keys

Data Model

- Many tables have two sets of indexes
  - First set maintains referential integrity
    - IEF confuses unique indexes with primary keys so some indexes are untouchable as to columns
      - But order of columns can be tuned
  - Second set are natural data attributes on which users query
  - Except for primary keys for referential integrity, almost all application queries involve some sort of range retrieval.

Data Model

- Is very hierarchically oriented
- In some cases, even with synthetic keys in places keys involve 15-16 columns
  - No indexes exceed byte limit for B-trees
- Since application code is machine generated and data is often obtained via joins, can often find zig-zag join strategies

Data Model

- Data model designed according to “theory”
- Incorporates concept of subject areas
- Applications also developed according to subject areas
- Subject areas become almost sub-databases
- Limited numbers of queries across subject areas

Data Model

- One source for concurrency is to isolate applications within their individual subject areas
- DENORM tables created and populated daily provide interfaces between subject areas
- Suits the high transaction rate environment well

Use External Transport

- In one case, an external transport, MQ, is used to beam messages between trading system and positions system
  - Trading & positioning need to execute concurrently
  - Reduces potential for locking problems
- MQ does not participate in distributed transactions
- MQ is significantly resource (I/O) intensive
  - Tune via commit intervals, usually an Rdb thing but with database physical design is an MQ tuning tool

Software Context

- Many developers initially had no idea about databases, SQL, network, client-server, ACMS…
- Focus on development in their own machines, forms etc. PC development database is Oracle 7
- Actual SQL is generated by IEF and is not under developers' control
- Only edits allowed are in IEF. No hand tailoring of code
  - Special exits called EABs are allowed to access system routines etc. We code these in C

Software Context

- The use of the 4-GL tool restricts developers in what they can do
  - Much of the application flow is outside developers' hands
    - Start/end transaction points are not obvious or accessible
    - The SQL generated is unusual and not editable
    - Many joins also require sorts across columns in all participating tables, no natural sort orders are possible
  - Deadlock retry is built in
  - Complete dependence on logical data model keys
    - No DB-keys ever!

Software Context

- Application developed by a large number of developers across multiple organizations
- Development style, even within IEF, highly variable
- Given complex data model, results in many creative ways to access the same data
- Results in a larger number of indexes

Program Modules Generated

- Each database conversation is generated as a single program module. For interactive programs these are like ACMS tasks
- ACMS task definitions are very crude
  - Details of ACMS tasks are really buried within IEF constructs & not visible to ACMS
- No way to separate functions by read and update activities

ACMS Server Design

- By default, almost all servers are read-write
  - One update anywhere in the conversation set forces this
- Can affect the packaging of servers so they support multiple conversations
  - Total number of servers and database attaches can be kept in some sort of bounds
- Deployment model is to have all instantiated servers be reasonably active

Transaction Model

- At start of project a decision was made to use the default IEF transaction model
  - No DBA involvement in this
  - Too many EABs otherwise
  - Could not be projected into the DB2 environment
- Default model could be manipulated to be any version of transaction
  - Read-only or read-write
  - Isolation level
  - That's it! By the structure of the application all transactions must be read write!

DB2 Coexistence

- By definition, the application would coexist on both Rdb and DB2 database platforms
- Rdb would not be required to provide greater data integrity than DB2
  - Any transactional consistency problems were to be handled by the application
- Close examination of DB2 transactional consistency revealed that it was close to “read committed.”

Rdb Transaction Model

Set transaction

read write

isolation level

read committed;

No reserving clauses anywhere

Rdb Transaction Model

- Read committed transactions:
  - Dismiss row locks after rows returned to application
  - Dismiss index locks after index structure processing is finished
    - Locks held on B-tree node for longer periods while an index scan is performed
- Data integrity side effects are possible
  - Requires code to defeat them

Application Style

- The end-user visible portion of the application is a traditional PC GUI interface
- Users point and click
- May fill out one or several fields to specify retrieval goals
- Queries use wildcard matching and *like* predicates freely.
  - All end-user queries result in range scans
- Requires many B-tree indexes to support
- Live and die by the capabilities of the dynamic optimizer

Rdb Considerations

- Restriction of the database to execution on a single VMS node offers significant opportunities for performance tuning
  - Commit to journal transactions
  - Elimination of the distributed lock manager
  - Row caches

Physical Database Design

- Design is simple, each table to its own storage area, each index to its own area
- Horizontally partitioned tables and indexes are exception
- Results in 1,500+ storage areas
- Fortunately Rdb now caches the File Ids so file opens are faster

Physical Database Design

- Storage areas for tables are distributed sequentially across disks, storage area 1 on disk 1, etc.
  - Some tables are partitioned
- Index storage areas are distributed on disks different from table, one index per storage area except for partitioned indexes
- Resulting round-robin placement yields disks which are busier than others
- Haven’t gotten around to manually balancing the I/O; Haven’t needed to yet

Fast Commit

- This is an extremely update intensive database
- Many servers do only one thing, repeatedly
- Good setup for fast commit protocol
  - Writes to live database accomplished in bunches only periodically
  - Objects updated across several transactions are written only once
  - Many B-tree indexes on important tables
    - 11 on Trade table
    - 6 on Allocation table
    - Some DENORM tables have > 20 indexes

Need for Commit to Journal

- Commit to journal transactions do not post transaction start/end to the database root file
- They also grab TSN's in bunches
- Result in substantially reduced I/O to root file
- CTJ becomes mandatory when the following are considered:
  - Projected transaction rates
  - 4 memory-speed disks would be required for the root file
  - FDDI link to the remote site

RMU Summary I/O Statistics

```
Node: TMSA (1/1/1)      Oracle Rdb V7.0-2 Perf. Monitor   2-NOV-1999 17:56:10.50
Rate: 1.00 Second            Summary IO Statistics          Elapsed: 08:52:08.45
Page: 1 of 1      PD_DATABASE_ROOT:[DATABASE]DATABASE.RDB;2         Mode: Online
```

statistic.....	rate.per.second.....			total.....	average.....
name.....	max.....	cur.....	avg.....	count.....	per.trans....
transactions	187	10	33.2	1062941	1.0
verb successes	7900	379	1456.9	46557410	43.8
verb failures	30	0	0.0	2666	0.0
synch data reads	315	0	22.0	703761	0.6
synch data writes	613	0	40.4	1290877	1.2
asynch data reads	93	0	0.0	1680	0.0
asynch data writes	128	0	3.1	99307	0.0
RUJ file reads	157	0	0.1	6071	0.0
RUJ file writes	61	0	0.9	31875	0.0
AIJ file reads	12	0	0.0	814	0.0
AIJ file writes	98	0	24.9	798203	0.7
ACE file reads	0	0	0.0	0	0.0
ACE file writes	0	0	0.0	0	0.0
root file reads	0	0	0.0	1	0.0
root file writes	57	0	7.4	239085	0.2

```
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write X_plot
```

Transaction Rates & Duration

```
Node: TMSA (1/1/1)      Oracle Rdb V7.0-2 Perf. Monitor   2-NOV-1999 17:56:10.50
Rate: 1.00 Second         Transaction Duration (Total)      Elapsed: 08:52:08.45
Page: 1 of 1         PD_DATABASE_ROOT:[DATABASE]DATABASE.RDB;2      Mode: Online
------------------------------------------------------------------------------
Total transaction count:    1062866
Seconds    Tx.Count:   %  #Complete:   % #Incomplete:   %
 0-< 1:     1040950  97%    1040950  97%       21916   3% <- avg=0.39 95th=0.60
 1-< 2:        5712   0%    1046662  98%       16204   2%
 2-< 3:        2234   0%    1048896  98%       13970   2%
 3-< 4:        1071   0%    1049967  98%       12899   2%
 4-< 5:         664   0%    1050631  98%       12235   2%
 5-< 6:         563   0%    1051194  98%       11672   2%
 6-< 7:         480   0%    1051674  98%       11192   2%
 7-< 8:         612   0%    1052286  99%       10580   1%
 8-< 9:         512   0%    1052798  99%       10068   1%
 9-<10:         406   0%    1053204  99%        9662   1%
10-<11:         236   0%    1053440  99%        9426   1%
11-<12:         142   0%    1053582  99%        9284   1%
12-<13:          75   0%    1053657  99%        9209   1%
13-<14:          80   0%    1053737  99%        9129   1%
14-<15:          49   0%    1053786  99%        9080   1%
 15+++:        9080   0%    1062866 100%           0   0%
------------------------------------------------------------------------------
Config Exit Graph Help Menu Options Reset Set_rate Unreset Write !
```

Memory Management

- Initially we used global buffers
  - P0 global buffer pool
  - Limited to just under 1 Gb of global buffer space
  - Created enormous difficulties for managing the application because of VMS tuning issues
  - With servers binding to the database once per day, is not so bad, though.

Global Buffers

- Each global buffer & global page is a resource that must be managed
- Rdb's management technique involves use of hash tables which are sized by a 2^N algorithm
- Given our buffer size of 12 blocks, resulted in 250 Mb of management overhead, ¼ of our total!

Global Buffers

- The management overhead was deemed to be too large
- On the basis of “if you don’t read it, you won’t pay to read it” we decided to increase the buffer size to 36 blocks
  - Page sizes remained 6 & 12 blocks
- Result was an additional 200 Mb for data!

System Space Global Buffers

- System space global buffers allowed us to eliminate the VMS memory management nightmare of the 1GB S0 global buffer pool
- Collided with VMS for access to the 2Gb of virtual system memory
  - Enormous VA were reserved for VMS pool growth
  - Placing a max value on pool growth allowed us to reserve VA for Rdb

System Space Global Buffers

- Result was that we exhausted VMS VA capability
- Can only have one of these databases operational in a machine at a time
- Meant that important DBA tuning functions had to be done only in the QA machine or in production

Locking Issues and Deadlock

- The application had a large number of deadlock problems
  - > 3,500 deadlocks per day
  - Mostly page-level deadlocks
  - Most of these were due to:
    - Multiple processes updating the same page
    - SPAM page contention
- Deadlock rate generated unacceptable performance even with deadlock wait set to 1 second

Page Contention

- Page contention was due to:
  - Multiple inserters
    - We separated processes via horizontal partitioning whenever possible
    - Large numbers of B-tree indexes forbid that in many circumstances
  - The flow-through nature of the application
    - Process A inserts and notifies process B [used BLASTS]
    - Process B queries db and does its work and notifies process C
    - C does its work etc.
    - Different processes all contend for access to the same page & row

Page Contention

- This page contention is an interaction between
  - Fast commit, pages are retained in memory
  - Flow through nature of the application
- Could not be defeated by hashed placement of data rows
  - We tried that and it didn't work
- Page transfer via memory [OPT] was tried and yielded a 20% I/O benefit but we had to disable because of reliability concerns

Flow Through Issues

- Work was flowed through by primary key
- Would lead to immense problems with a chronological index
- Application calculated its own hashing function & we used sorted index.
  - Works surprisingly well
  - DBAs had added a hashed index at same time and interaction between two algorithms was awful!
- Defeated some of the page collisions by passing primary keys in shared memory queues

Row Caches

- The page deadlock rate was still much too high to be able to perform at required speed
- This was defeated, finally, by using row caches
  - Only process that inserts data needs to see the page
  - Plus the RCS process when it hits a database checkpoint or is required to free a row cache slot
  - Page contention dropped, essentially, to zero
- Used logical area row caches, cache name is identical to table name/index name

File I/O Rates

```
Node: TMSA (1/1/1)      Oracle Rdb V7.0-2 Perf. Monitor   5-NOV-199914:26:20.61
Rate: 1.00 Second        File IO Overview (Total I/Os)      Elapsed:01:03:56.99
Page:  1 of 54       PD_DATABASE_ROOT:[DATABASE]DATABASE.RDB;2       Mode:Online
```

File/Storage.Area.Name.....	Sync.Reads	SyncWrites	AsyncReads	AsyncWrits	PgDis
All data/snap files	38033	177525	261	393	10226
AIJ (After-Image Journal)	105	111977	0	0	0
Database Root	0	34026	0	312	0
data POSITION_INDX_47	442	15380	0	0	203
data POSITION_INDX_49	265	11668	0	0	475
data POSITION_INDX_45	0	11481	0	0	0
data TMS_DATA_86	366	10218	155	0	0
data DATABASE_RDB$SYSTEM	9852	322	0	0	0
data POSITION_INDX_43	32	10142	0	0	0
data TMS_INDX_006	3528	5767	0	0	2610
data TMS_INDX_005	0	8911	0	0	0
data POSITION_INDX_51	10	8532	0	0	0
data TMS_INDX_052	1922	5391	0	4	0
data TMS_DATA_01	2416	3998	0	13	62
data TMS_INDX_003	767	5451	0	2	9
data POSITION_DATA_15	0	5021	0	0	0
data POSITION_INDX_32	0	4577	0	0	0
data TMS_INDX_007	991	3443	0	4	10
data POSITION_DATA_11	0	4193	0	0	0
RUJ (Recovery-Unit Journal)	1105	45	0	2724	0

Row Caches and Buffers

- With some thought, we replaced the global buffer pool entirely with row caches
  - Large caches are kept in VLM
    - Blows away the 1+ Gb VA limit for database structures
  - Smaller caches are kept in system space
  - Split determined by whether the address space required for VLM in system space was larger than address space just to have in system space!
  - Increased system memory and CPU (6 EV5.6) at this time
- Eliminated the global buffer pool
  - Large global buffer pool would just result in having data in memory twice
- Processes now use local buffers

Row Caches & Concurrency

- For the real-time front-end applications
  - All row caches for tables are sized to handle somewhat more slots than expected rows for a day's work
  - All index caches are sized to handle the entire B-tree
- Becomes a virtually memory-resident database

Deadlocks Across A Day

```
Node: TMSA (1/1/1)      Oracle Rdb V7.0-2 Perf. Monitor   2-NOV-1999 17:56:10.50
Rate: 1.00 Second          Summary Locking Statistics       Elapsed: 08:52:08.45
Page: 1 of 1       PD_DATABASE_ROOT:[DATABASE]DATABASE.RDB;2        Mode: Online
-------------------------------------------------------------------------------
statistic.........      rate.per.second.............. total....... average......
name..............      max..... cur..... avg....... count....... per.trans....
locks requested            28314      446     3712.0    118623065         111.5
 rqsts not queued           3054        0      297.7      9515746           8.9
 rqsts stalled                95        0       16.8       539233           0.5
 rqst timeouts                 0        0        0.0            0           0.0
 rqst deadlocks                2        0        0.0           45           0.0
locks promoted               907       10      215.0      6873136           6.4
 proms not queued             26        0        1.4        46790           0.0
 proms stalled                68        0        4.7       152138           0.1
 prom timeouts                 0        0        0.0            0           0.0
 prom deadlocks                1        0        0.0          144           0.0
locks demoted                6588       10      184.2      5887274           5.5
locks released              27170      445     3415.4    109144142         102.6
blocking ASTs                 819        0       21.6       692664           0.6
stall time x100              7138        0       47.7      1524898           1.4
invalid lock block              0        0        0.0            0           0.0
ignored lock mode               0        0        0.0            0           0.0
-------------------------------------------------------------------------------
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write X_plot
```

Request Deadlocks

```
Node: TMSA (1/1/1)      Oracle Rdb V7.0-2 Perf. Monitor   2-NOV-1999 17:56:10.50
Rate: 1.00 Second          Locking ( rqst deadlocks)        Elapsed: 08:52:08.45
Page: 1 of 1         PD_DATABASE_ROOT:[DATABASE]DATABASE.RDB;2      Mode: Online
```

statistic.....	rate.per.second.....			total.....	average.....
name.....	max.....	cur.....	avg.....	count.....	per.trans....
total locks	2	0	0.0	45	0.0
area locks	0	0	0.0	0	0.0
buffer/page locks	0	0	0.0	1	0.0
record locks	2	0	0.0	44	0.0
SEQBLK lock	0	0	0.0	0	0.0
FILID locks	0	0	0.0	0	0.0
TSNBLK locks	0	0	0.0	0	0.0
RTUPB lock	0	0	0.0	0	0.0
ACTIVE lock	0	0	0.0	0	0.0
MEMBIT lock	0	0	0.0	0	0.0
AIJ locks	0	0	0.0	0	0.0
snapshot locks	0	0	0.0	0	0.0
freeze lock	0	0	0.0	0	0.0
quiet point lock	0	0	0.0	0	0.0
logical area locks	0	0	0.0	0	0.0
nowait transaction	0	0	0.0	0	0.0

```
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write X_plot
```

Promotion Deadlocks

```
Node: TMSA (1/1/1)      Oracle Rdb V7.0-2 Perf. Monitor   2-NOV-1999 17:56:10.50
Rate: 1.00 Second          Locking ( prom deadlocks)        Elapsed: 08:52:08.45
Page: 1 of 1        PD_DATABASE_ROOT:[DATABASE]DATABASE.RDB;2       Mode: Online
```

statistic.....	rate.per.second.....			total.....	average.....
name.....	max.....	cur.....	avg.....	count.....	per.trans....
total locks	1	0	0.0	144	0.0
area locks	0	0	0.0	0	0.0
buffer/page locks	0	0	0.0	9	0.0
record locks	1	0	0.0	135	0.0
SEQBLK lock	0	0	0.0	0	0.0
FILID locks	0	0	0.0	0	0.0
TSNBLK locks	0	0	0.0	0	0.0
RTUPB lock	0	0	0.0	0	0.0
ACTIVE lock	0	0	0.0	0	0.0
MEMBIT lock	0	0	0.0	0	0.0
AIJ locks	0	0	0.0	0	0.0
snapshot locks	0	0	0.0	0	0.0
freeze lock	0	0	0.0	0	0.0
quiet point lock	0	0	0.0	0	0.0
logical area locks	0	0	0.0	0	0.0
nowait transaction	0	0	0.0	0	0.0

```
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write X_plot
```

Row Caches and I/O

- The RCS becomes the focus for writing to disk for objects modified in the cache
- Transfers responsibility from application processes
- Commits occur even faster because I/O now done mostly by the RCS process
- Checkpoints going to backing store and not to the live storage areas
  - I/O is done sequentially in the backing store
  - Database shutdown requires a bit more time
  - Starting read only transactions is slower
    - RMU Load/Unload
    - On-line backup

I/O By Processes

```
Rate: 1.00 Second              Process Accounting          Elapsed:00:52:23.42
Page: 1 of 2       PD_DATABASE_ROOT:[DATABASE]DATABASE.RDB;2      Mode:Online
-------------------------------------------------------------------------------
Process.ID Process.name... CPUtime.... EnqCnt.. PGflts. NumDio. WSsize. VMsize.
20A11FCF:1sRDM_RCS70_1     03:33:34.60  1046720  180123 2610347  100000  728064
20A11AD1:1sRDM_ALS70_1     01:21:27.94  1048504     348 4740258    7552  184240
20A102DB:1sRDM_LCS70_1     00:04:28.37  1048557     335  111862    7520  183952
20A10881:1 PD_SYNCFUTTRD   00:01:34.37   149265    1625   28646   30080  231120
20A10189:1 PD_BRDCASTSRV   00:40:26.18   149509    1393    1394   26528  290800
20A1248A:1 PD_MAILSRV      00:00:50.17   149712    4759     350   28416  280528
20A10790:1 PD_SYNCFUTXFER  00:01:40.70   149275    1707   28961   28832  231120
20A0EC8F:1 PD_TM395        00:00:08.96   148945    2808    2129   52496  379840
20A10196:1 PD_SYNCOPTTRD   00:00:12.36   149593    1522    4908   25232  229776
20A0EA94:1 PD_TM152        00:28:42.89   148213    2057  197629   51360  337040
20A0FA99:1 PD_952_O        00:00:24.17   149710    1732    1582   30112  287792
20A0E495:1 PD_TM153        00:05:40.91   148945    2079   45005   41008  330800
20A1209C:1 PD_SYNCOPTXFER  00:00:09.79   149617    1381    3961   25488  229712
20A0EE9F:1 PD_952_F        00:07:48.39   148352    2573   39545   53968  371760
20A11197:1 PD_TM991        00:11:15.20   148636    3367  130691   67792  363424
20A14CBB:1 ACMS15CSP015020 00:14:38.16   148870   59092   28732   30096  376784
20A0EBDC:1 ACMS152SP001000 00:00:00.65   149730    1501     213   34032  276816
20A0F098:1 PD_TM992        00:00:11.02   148686    2001    2694   40336  271968
20A103A0:1 PD_951_O        00:00:29.10   149044    2608    2269   54048  412928
20A120A1:1 PD_951_F        00:06:14.44   143224    5296   23800   92128  439696
20A139A2:1 PD_TM792        00:00:11.10   149279    1970    1324   42272  338656
```

Row Cache Slot Sizing

- Row cache slots were sized to handle maximum row size
  - Determined from AIP
- Size was rounded up such that an AXP page would hold an integer number of slots, exactly
  - Indexes only, table rows differ in size too radically
  - Eliminates thrashing of addressing at boundary points

Row Cache Issues

- The hashing algorithm for finding a row in the cache is determined in part by the line number on the page
- Results in more collisions than expected because we have large pages (6 & 12 blocks)
  - Algorithm assumes line number < 23
  - We have up to 40 & 50 per page
  - See the hash misses on next slide
    - Is something like 13%!
- We are ignoring the computational cost of this
  - Probably not much anyway
  - Maybe agitation can get the algorithm changed at some point

General Cache Statistics

```
Rate: 1.00 Second          Summary Cache Statistics        Elapsed: 08:52:08.45
Page: 1 of 1       PD_DATABASE_ROOT:[DATABASE]DATABASE.RDB;2        Mode: Online
--------------------------------------------------------------------------------
statistic.........      rate.per.second.............. total....... average......
name..............      max..... cur..... avg....... count....... per.trans....
latch requests               878        0        1.2        38438           0.0
  retried                      1        0        0.0           21           0.0
cache searches             48617      609     4146.3    132500475         124.6
  found in workset         18899      233     2075.1     66313710          62.3
  found in cache           47998      375     2043.5     65303980          61.4
  found too big               64        0        0.1         5624           0.0
insert cache                1897        0       33.8      1079570           1.0
  row too big                  0        0        0.0            0           0.0
  cache full                  42        0        0.0          371           0.0
  collision                    0        0        0.0            4           0.0
VLM requests               47829      375     2509.2     80187788          75.4
  window turns              8309        0      406.8     13001167          12.2
skipped dirty slot        421903        0      425.4     13594398          12.7
skipped inuse slot          4518        0        2.9        93513           0.0
hash misses                53261      127      553.0     17671797          16.6
cache unmark                4335        0       55.9      1786296           1.6
--------------------------------------------------------------------------------
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write X_plo
```

Row Cache Backing Store

- All row cache backing stores are on same disk
  - Is second most busy disk on system, AIJ is first
- RCS writes these serially anyway, so no I/O contention
- Sized initially so that RCS backing store files do not grow during the day
  - Eliminated a large source of file fragmentation
- RCS checkpoints keep total size of files down to reasonable number
  - Fits comfortably on a 4 Gb drive

Row Caches

- The introduction of row caches eliminated most of the page deadlocks in the application
  - There are still a few per day
- Application throughput is now satisfactory
- We believe we have considerable growth potential
  - Had better be because we are only 50% of the way there
  - We also know where the greatest risks are

Reading The Database

```
Node: TMSA (1/1/1)      Oracle Rdb V7.0-2 Perf. Monitor   2-NOV-1999 17:56:10.50
Rate: 1.00 Second         PIO Statistics--Data Fetches      Elapsed: 08:52:08.45
Page: 1 of 1        PD_DATABASE_ROOT:[DATABASE]DATABASE.RDB;2       Mode: Online
-------------------------------------------------------------------------------
statistic.........      rate.per.second.............. total....... average......
name..............      max..... cur..... avg....... count....... per.trans....

fetch for read             34655        0     1039.4     33217705          31.2
fetch for write            67952        0      683.7     21851249          20.5

in LB: all ok              81805        0     1624.8     51923059          48.8
   LB: need lock             462        0       77.7      2483438           2.3
   LB: old version           225        0        4.1       133828           0.1

not found: read              168        0       16.9       540444           0.5
         : synth              87        0        1.4        46650           0.0

DAPF: success                 0        0        0.0            0           0.0
DAPF: failure                 0        0        0.0            0           0.0
DAPF: utilized                0        0        0.0            0           0.0
DAPF: discarded               0        0        0.0            0           0.0

-------------------------------------------------------------------------------
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write X_plot
```

Writing Buffers

```
Node: TMSA (1/1/1)      Oracle Rdb V7.0-2 Perf. Monitor   2-NOV-1999 17:56:10.50
Rate: 1.00 Second         PIO Statistics--Data Writes       Elapsed: 08:52:08.45
Page: 1 of 1        PD_DATABASE_ROOT:[DATABASE]DATABASE.RDB;2       Mode: Online
------------------------------------------------------------------------------
statistic.........      rate.per.second.............. total....... average......
name..............      max..... cur..... avg....... count....... per.trans....
unmark buffer                528        0       41.9      1340035           1.2
  transaction                  5        0        0.0           20           0.0
  pool overflow              117        0        2.9        95677           0.0
  blocking AST                 0        0        0.0            0           0.0
  lock quota                   0        0        0.0            0           0.0
  lock conflict              137        0        4.4       140497           0.1
  user unbind                100        0        0.0          925           0.0
  batch rollback               0        0        0.0            0           0.0
  new area mode                0        0        0.0            0           0.0
  larea change                16        0        0.0          600           0.0
  incr backup                  0        0        0.0            0           0.0
  no AIJ access                0        0        0.0            0           0.0
  truncate snaps               0        0        0.0            0           0.0
  checkpoint                 528        0       34.5      1102506           1.0
  AIJ backup                   0        0        0.0            0           0.0

------------------------------------------------------------------------------
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write X_plot
```

Sorted Indexes

- All indexes are sorted, no hash indexes at all
- Many indexes permit duplicate values
  - Helps keep track of exactly what is needed in terms of tuning
  - Provides more accurate information to optimizer
- We tried using ranked indexes
  - Permits duplicate values more densely
  - Avoids problems with discarded pages
  - Failed because of bugs
- Node sizes are standard
  - 480/960 uncached
  - 488/1000 cached
    - Must be [(even divisor of 8192) – 24], e.g. 1000, 2024, …

Discarded Pages Due to Duplicates Indexes

```
Node: TMSA (1/1/1)      Oracle Rdb V7.0-2 Perf. Monitor   2-NOV-1999 17:56:10.50
Rate: 1.00 Second             Record Statistics           Elapsed: 08:52:08.45
Page: 1 of 1        PD_DATABASE_ROOT:[DATABASE]DATABASE.RDB;2       Mode: Online
-------------------------------------------------------------------------------
statistic.........      rate.per.second.............. total....... average......
name..............      max..... cur..... avg....... count....... per.trans....
record marked              11214        0      141.4      4518735           4.2
record fetched             48783      609     5176.2    165415279         155.6
    fragmented               141        0        1.3        41598           0.0
  record stored             1513        0       35.4      1132816           1.0
     fragmented                0        0        0.0            0           0.0
  pages checked             1513        0       36.8      1177686           1.1
       saved IO             1513        0       24.7       789988           0.7
      discarded              875        0        1.4        44871           0.0
record erased                2803        0        1.5        50255           0.0
   fragmented                   0        0        0.0            0           0.0

temp record marked              0        0        0.0            0           0.0
temp record fetchd              0        0        0.0            0           0.0
temp record stored              0        0        0.0            0           0.0
temp record erased              0        0        0.0            0           0.0

-------------------------------------------------------------------------------
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write X_plot
```

The Effects of Shadowing

- Shadowing the large number of virtual disks across a wide-area cluster proved to be infeasible
  - If node goes down it took 2-3 days to synchronize the disks
  - Exposed application to significant outage if two consecutive failures occurred

Hot Standby

- Wide-area volume shadowing weakness addressed by deploying Hot Standby
- AIJs fill at the rate of 6*630,000 block AIJs per day
  - Again, mostly due to the on-line transactions
  - Very “peaky” activity
- FDDI link is able to handle the data rate
  - Synch to warm.
  - Seldom do we see HS throttling activity by synchronizing to hot or commit (see next slide)
    - AIJ switches
    - Checkpoints?

Hot Standby

```
Node: TMSA (1/1/1)      Oracle Rdb V7.0-2 Perf. Monitor   2-NOV-1999 17:56:10.50
Rate: 1.00 Second       Synchronization Mode Statistics     Elapsed: 08:52:08.45
Page: 1 of 1        PD_DATABASE_ROOT:[DATABASE]DATABASE.RDB;2       Mode: Online
-------------------------------------------------------------------------------
statistic.........      rate.per.second.............. total....... average......
name..............      max..... cur..... avg....... count....... per.trans....

transactions                 187       10       33.2      1062941           1.0

Cold sync send                99        0       21.2       677637           0.6
Warm sync send                 9        0        3.7       120566           0.1
Hot sync send                  0        0        0.0            0           0.0
Commit sync send               1        0        0.0         1254           0.0

Cold stall x100                4        0        0.0           51           0.0
Warm stall x100              105        0       43.3      1383664           1.3
Hot stall x100                 0        0        0.0            0           0.0
Commit stall x100            250        0        0.5        16591           0.0

Startup/Shutdown               0        0        0.0            0           0.0
Unexpected Failure             0        0        0.0            0           0.0

-------------------------------------------------------------------------------
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write X_plot
```

Hot Standby Performance

- The LCS process on the standby node was configured with the maximum possible buffer count
- Prevents excessive database reads
- Tuned the TCP I/O subsystem by manipulating buffers and sizes

Stall Times in the Database

```
Node: TMSA (1/1/1)      Oracle Rdb V7.0-2 Perf. Monitor   2-NOV-1999 17:56:10.50
Rate: 1.00 Second         IO Stall Time (seconds x100)      Elapsed: 08:52:08.45
Page: 1 of 1        PD_DATABASE_ROOT:[DATABASE]DATABASE.RDB;2       Mode: Online
------------------------------------------------------------------------------
statistic.........      rate.per.second.............. total....... average......
name..............      max..... cur..... avg....... count....... per.trans....

root read time                0        0        0.0            0           0.0
root write time             256        0       10.3       332705           0.3

data read time              778        0       27.6       882814           0.8
data write time             512        0       19.3       618851           0.5
data extend time             60        0        0.0          266           0.0

RUJ read time                50        0        0.0          738           0.0
RUJ write time                5        0        0.0          471           0.0
RUJ extend time             201        0        0.3        11934           0.0
AIJ read time                 1        0        0.0           10           0.0
AIJ write time                4        0        0.0           24           0.0
AIJ hiber time             1490        0      228.9      7315049           6.8
AIJ extend time               0        0        0.0            0           0.0
Database bind time           15        0        0.0         1791           0.0
------------------------------------------------------------------------------
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write X_plot
```

Sample Database Stalls

```
Node: TMSA (1/1/1)      Oracle Rdb V7.0-2 Perf. Monitor   5-NOV-199914:12:30.47
Rate: 1.00 Second               Stall Messages             Elapsed:00:50:06.86
Page: 1 of 1        PD_DATABASE_ROOT:[DATABASE]DATABASE.RDB;2      Mode:
Online
------------------------------------------------------------------------------
-
Process.ID Since..... T Stall.reason............................. Lock.ID.
20A0EE9F:1 14:12:30.56 W waiting for record 837:8:1 (PR)           3B07D612
20A10790:1 14:12:30.62 - writing 2 pages back to database
20A125BE:1 14:12:30.62 - writing 23 pages back to database
20A0EA94:1 14:12:30.64 W hibernating on AIJ submission
20A12DAB:1 14:12:30.64 W waiting for record 846:8:1 (PR)           060E230C
20A11AD1:1s14:12:30.65 - waiting for "Data_Req" reply 4716801 from standby database
20A14CBB:2 14:12:30.66 W hibernating on AIJ submission
```

Hot Standby Availability

- Define hot standby node on primary system to use one FDDI link
- Define second link to be available should first link become unavailable.
- Logistics of testing is awkward because the only place to test fail-over is on production systems
  - Implies lots of weekend work for more than just the DBA team

Snapshots

- Snapshots are enabled deferred
- Supports on-line backup
- When read-only transaction started, creates “event” in database
  - Loose high-performance characteristics for a while
  - RMU Load/Unload to read metadata
  - Backup
  - These activities are restricted to non-trading hours (mostly)

Snapshot Statistics

```
Rate: 1.00 Second            Snapshot Statistics          Elapsed: 08:52:08.45
Page: 1 of 1       PD_DATABASE_ROOT:[DATABASE]DATABASE.RDB;2      Mode: Online
-------------------------------------------------------------------------------

statistic.........     rate.per.second.............. total....... average......
name..............     max..... cur..... avg....... count....... per.trans....

Total transactions          187       10       33.2      1062941           1.0
R/O transactions              0        0        0.0            0           0.0

retrieved record              0        0        0.0            0           0.0
  fetched line                0        0        0.0            0           0.0
    read snap page            0        0        0.0            0           0.0

stored snap record            0        0        0.0            0           0.0
    page in use               0        0        0.0            0           0.0
    page too full             0        0        0.0            0           0.0
    page conflict             0        0        0.0            0           0.0
    extended file             0        0        0.0            0           0.0


-------------------------------------------------------------------------------
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write X_plot
```

Rdb Features Not Used

- Global buffers were abandoned when we went to row caching
- While we were using global buffers we were forced to abandon OPT, page transfer via memory, because of bugs. This caused us to take at least a 20% hit in increased I/O
- We abandoned our use of ranked indexes because of bugs
  - Will re-enable when reliability is back
    - Dynamic optimizer support
    - Discarded page problems solved

Checkpoint Management

- Initially some servers became idle
  - Cannot therefore checkpoint
  - Provide a window of unavailability should process fail
    - DBR would have to redo since the last checkpoint
- We forced manual checkpoints
  - Initially every 5 minutes
  - Currently at 30 minutes
  - Soon to be abandoned as developers get control of server rundown etc.
- Identified checkpoint anomalies & got engineering to fix

Checkpoint Statistics

```
e: TMSA (1/1/1)      Oracle Rdb V7.0-2 Perf. Monitor   2-NOV-1999 17:56:10.50
Rate: 1.00 Second            Checkpoint Statistics          Elapsed: 08:52:08.45
Page: 1 of 1        PD_DATABASE_ROOT:[DATABASE]DATABASE.RDB;2       Mode: Online
```

statistic.....	rate.per.second.....			total.....	average.....
name.....	max.....	cur.....	avg.....	count.....	per.trans....
transactions	187	10	33.2	1062941	1.0
checkpoints	52	0	7.3	234560	0.2
AIJ growth	5	0	0.0	2178	0.0
txn limit	16	0	0.7	25081	0.0
time limit	0	0	0.0	0	0.0
rollback	3	0	0.0	718	0.0
AIJ backup	11	0	0.0	162	0.0
global	48	0	0.1	4727	0.0
interval: AIJ blks	134963	0	3494.2	111662485	105.0
interval: tx count	112	0	15.9	507771	0.4
interval: seconds	10018	0	43.8	1401093	1.3
checkpoint stall	4467	0	69.7	2227757	2.0
flushed buffers	528	0	34.5	1102506	1.0

```
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write X_plot
```

Summary

- Rdb has been able to manage a very intensive and difficult trading database
  - No hash indexes, > 1,000 sorted indexes
  - Very high transaction rates
  - High concurrency
  - No read-only transactions
- Very high availability

Thanks Rdb Engineering!

The authors want to thank Rdb engineering for their close cooperation and support while we were trying the new Row Caching code and Hot Standby and …