

Configuring Rdb Hot Standby for Maximum Performance

Keith W. Hare
Thomas H. Musson
JCC Consulting, Inc.

Introduction

- What is Hot Standby
 - How does it work?
- Hot Standby Configuration Choices
 - Understanding implications
- Network Configuration Choices
 - How they affect Hot Standby
- Balancing Competing Goals

What Is Hot Standby?

- All information from Master Database replicated to Standby database
- Continuous recovery of standby database
- Useful for disaster recovery
- Read only access to Standby is possible but:
 - No locks are held on standby database
 - Possible to view incomplete transactions

Hot Standby: How It Works

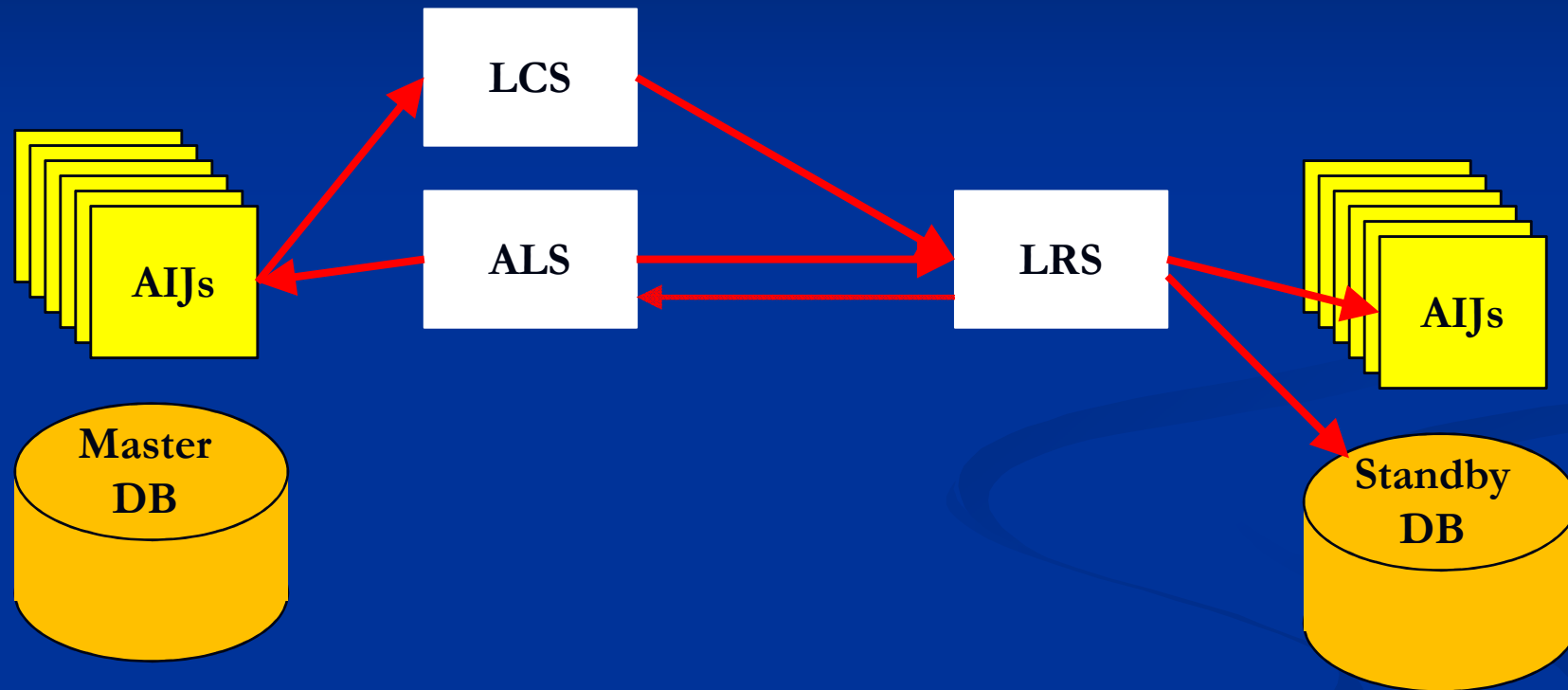
■ Master Database

- After Image Journaling is enabled
- Fast Commit is enabled
- AIJ Log Server is automatic

■ Standby Database

- Restore of a backup of the master database
- Same number of AIJs as the master database
- AIJs must be the same size as master database AIJs
- No update transactions
- Row Caches disabled

Hot Standby



Hot Standby Choices

- Network Transport
- Synchronization Level
- Checkpoint Frequency
 - Transaction Count
 - Time
- LCS (Log Catch-up Server) Checkpoint Frequency

Hot Standby: Use Of Network Layer

- DECnet
- TCP/IP
- Possible to define both a primary and secondary network path
- This talk focuses on TCP/IP

Hot Standby Synchronization Levels

- Cold (Default)
 - Standby Servers never return a message
 - On failure, standby might lose in-flight transaction
- Warm
 - Send message to master
 - Apply transaction to AIJ and database
- Hot
 - Write to AIJ on standby
 - Send message to master
 - Apply to database
- Commit
 - Write to Standby AIJ and database
 - Acknowledge successful commit to Master

Hot Standby Governor

- Governor Enabled/Disabled on Standby
- Enabled – If standby DB gets behind, throttle Master DB
- Disabled – Mostly don't throttle Master DB
 - If recovery buffers on standby must be written to disk, throttle the master DB
 - Can happen with very large transactions

Hot Standby Checkpoint Transaction Count

- Commits
 - /checkpoint=
 - RDM\$BIND_HOT_CHECKPOINT
- ALS on master waits for standby acknowledgement
- Example Values
 - 5 (Minimum) – We did not test this
 - 100 (Default)
 - 1024 (RMU72 stated maximum in help)
 - 10000
 - 50000 (Actual Maximum)

Hot Standby Checkpoint Time Interval

- RDM\$BIND_HOT_CHECKPOINT_INTERVAL
- Specifies a checkpoint interval, in minutes, to be used in addition to the /CHECKPOINT qualifier specified at Hot Standby startup.
- Defaults to 1 minute
- First threshold to be exceeded (message count or elapsed time) will cause the LRS checkpoint.
- Default is usually sufficient

Hot Standby LCS Sync Commit

- LCS message checkpoint
 - RDM\$BIND_LCS_SYNC_COMMIT_MAX
 - Translated at Master database open
- LCS on master stalls waiting for coordination with standby
- Example Values
 - 32 (Minimum)
 - 128 (Default)
 - 500
 - 1000
 - 10000

TCP/IP Parameters

- Drop_Count

“Number of idle probes that can go unsatisfied before the software declares a TCP connection dead and closes it.”

- Probe_Timer

“Number of seconds between probes for idle TCP connections (when the SO_KEEPALIVE option is set). If the remote system fails to respond, the connection is removed.”

- RDMAIJ72 service sets the SO_KEEPALIVE option

Testing Systems Setup

■ Source System

- BL870c – 4 x 1.59GHz/9.0MB, 16GB
- VMS 8.4
- Rdb 7.2.5.0

■ Target System

- RX4640 – 1.10GHz/4.0MB, 48GB
- VMS 8.4
- Rdb 7.2.5.0

Testing Network Setup

- Source System
 - 1Gbps (full duplex)
- Target System
 - 1Gbps (full duplex)
- Network Simulation
 - WANem (Sourceforge)
 - 1Gbps (full duplex)
 - Effectively 500Mbps (full duplex)

Testing Database Setup

- 5 identical tables
 - 11 columns
 - 1 identity (bigint) column
 - 10 char(20) with random data – e.g.)Y\$9k6NBk(R{;yDe'Pt2
 - 100k rows
- Global buffers
 - Large memory
 - 50,000
 - User limit 5,000

Testing Update Load Setup

- Use JCC LogMiner Loader Data Pump
 - 5 in parallel; 1 for each table
 - All rows
 - 10 rows per transaction
 - Total of 50,000 transactions
 - All transactions are same size

Hot Standby Network Latency

- <1ms
- 5ms
- 20ms
- 300ms
- 600ms

Hot Standby Network Bandwidth

- [1Gb]
- 100Mb
- 10Mb
- 1.544Mb

What to Monitor?

- Wall Time – affected by other work on systems
- Network Bandwidth used – indicative but imprecise
- Hot Standby Statistics
 - Messages Sent and Received – indicator of amount of traffic
 - Stall Time – best indicator of performance

Stall Time

- Cold Stall Time – Cold Synchronization
- Warm Stall Time – Warm Synchronization
- Hot Stall Time – Hot Synchronization
- Commit Stall Time
 - Commit Synchronization
 - Checkpoint Messages

Hot Standby Statistics Example

Node: PANDOR (1/1/1) Oracle Rdb V7.2-500 Perf. Monitor 1-SEP-2011 14:12:27.71
 Rate: 3.00 Seconds Synchronization Mode Statistics Elapsed: 00:18:38.93
 Page: 1 of 1\$1\$DGA104:[JCC_ROOT.TOM.SQL_CLASS.MF_V72.MASTER]HS.RDB;1Mode: Online

```

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

transactions                864          0      89.4      100017        1.0

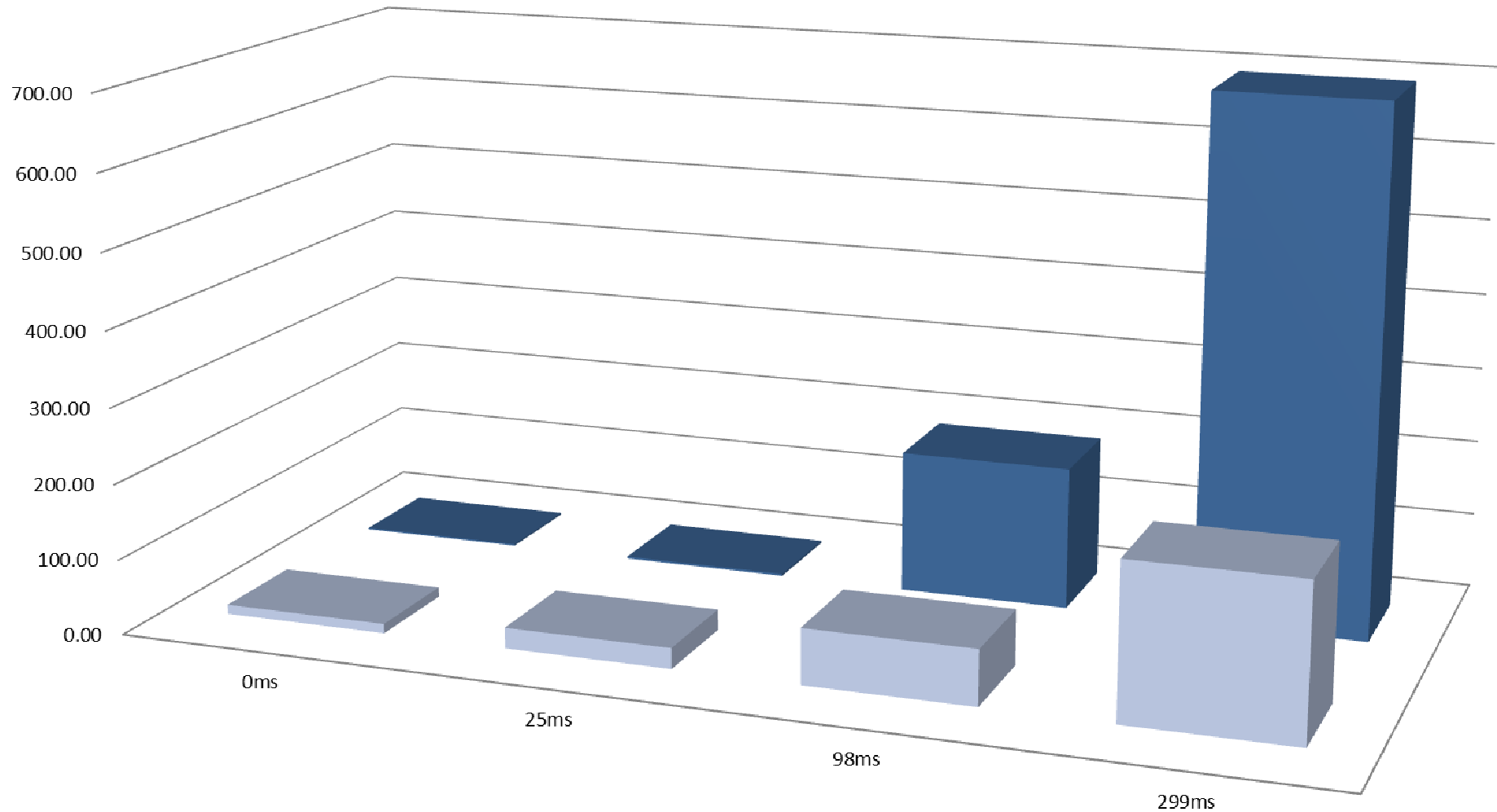
Cold sync send              432          0      44.7       50008        0.4
Warm sync send               0          0       0.0         0        0.0
Hot sync send                0          0       0.0         0        0.0
Commit sync send             4          0       0.4         532        0.0

Cold stall x1000             26          0       2.2       2454        0.0
Warm stall x1000              0          0       0.0         0        0.0
Hot stall x1000               0          0       0.0         0        0.0
Commit stall x1000           181          0       4.2       4716        0.0

Startup/Shutdown             0          0       0.0         1        0.0
Unexpected Failure            0          0       0.0         0        0.0
-----
  
```

Network Delay versus Stall Time

Synchronization Cold, Checkpoint 100, Govenor Disabled



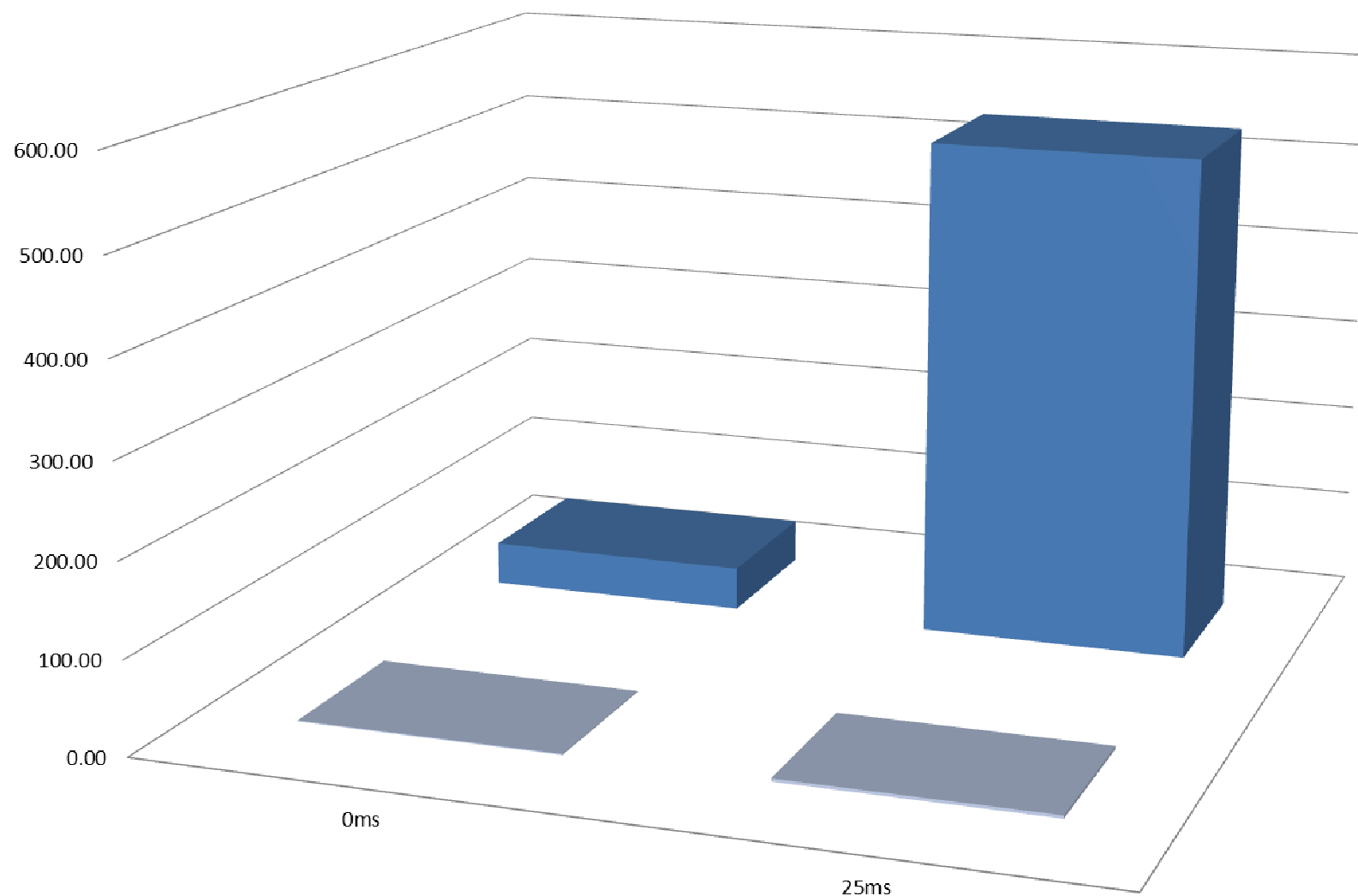
	0ms	25ms	98ms	299ms
Commit Stall	12.49	27.49	73.90	205.93
Cold Stall	1.88	2.68	189.16	695.30

Notes on Graph

- Cold Stall Time
 - TCP Acknowledgement packet
 - Becomes more significant on slower network
- Commit Stall Time
 - Function of checkpoint frequency
- Did not record statistics for 600Ms network delay

Network Delay versus Stall Time

Synchronization Warm, Checkpoint 100, Govenor Disabled



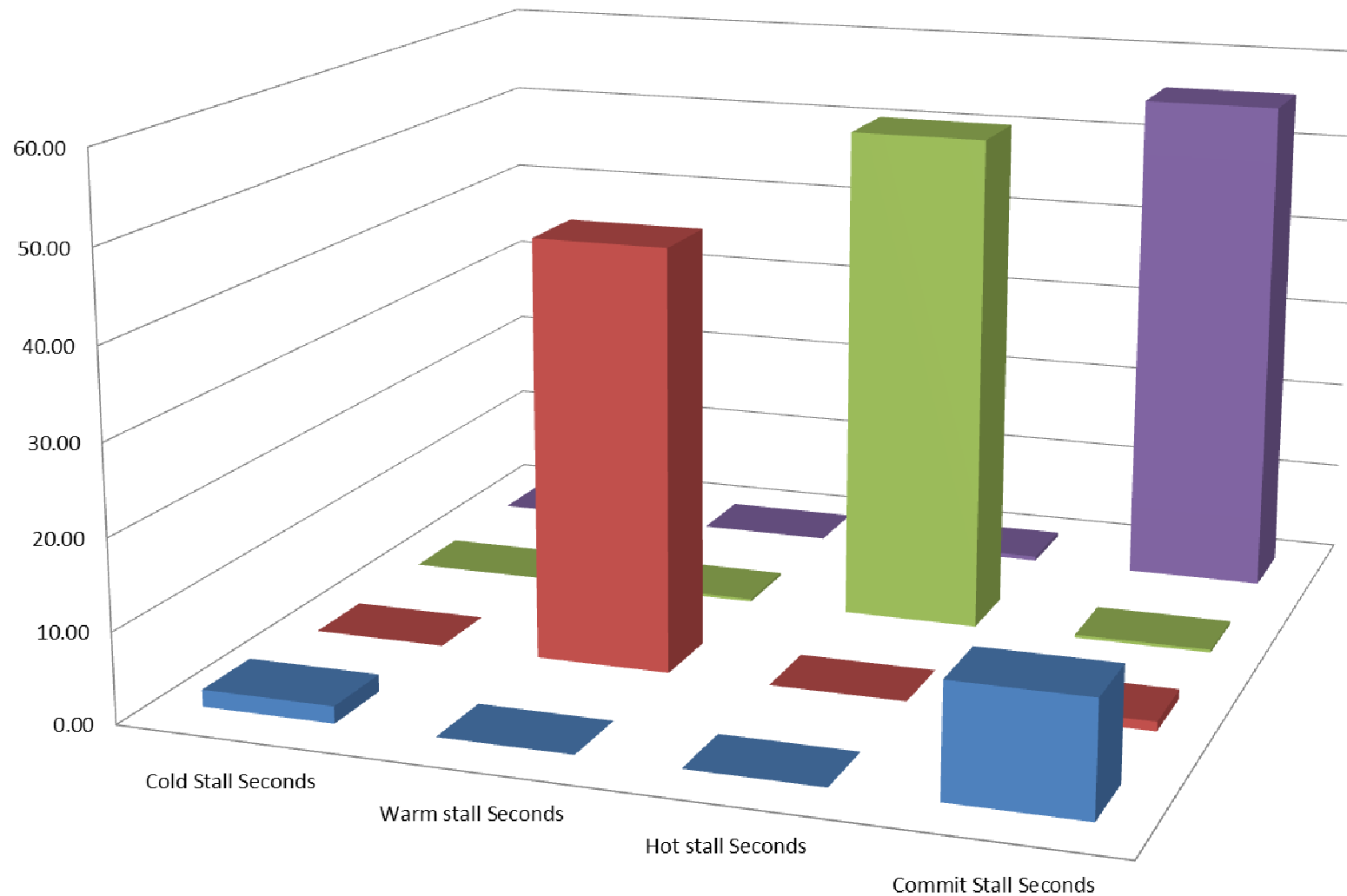
	0ms	25ms
Commit Stall	1.04	3.25
Warm Stall	46.52	536.00

Notes on Graph

- Warm Synchronization spends more time waiting for acknowledgement that transaction has been applied
- Network delays are magnified

Synchronization Level versus Stall Time

0ms Delay, Checkpoint 100, Govenor Disabled



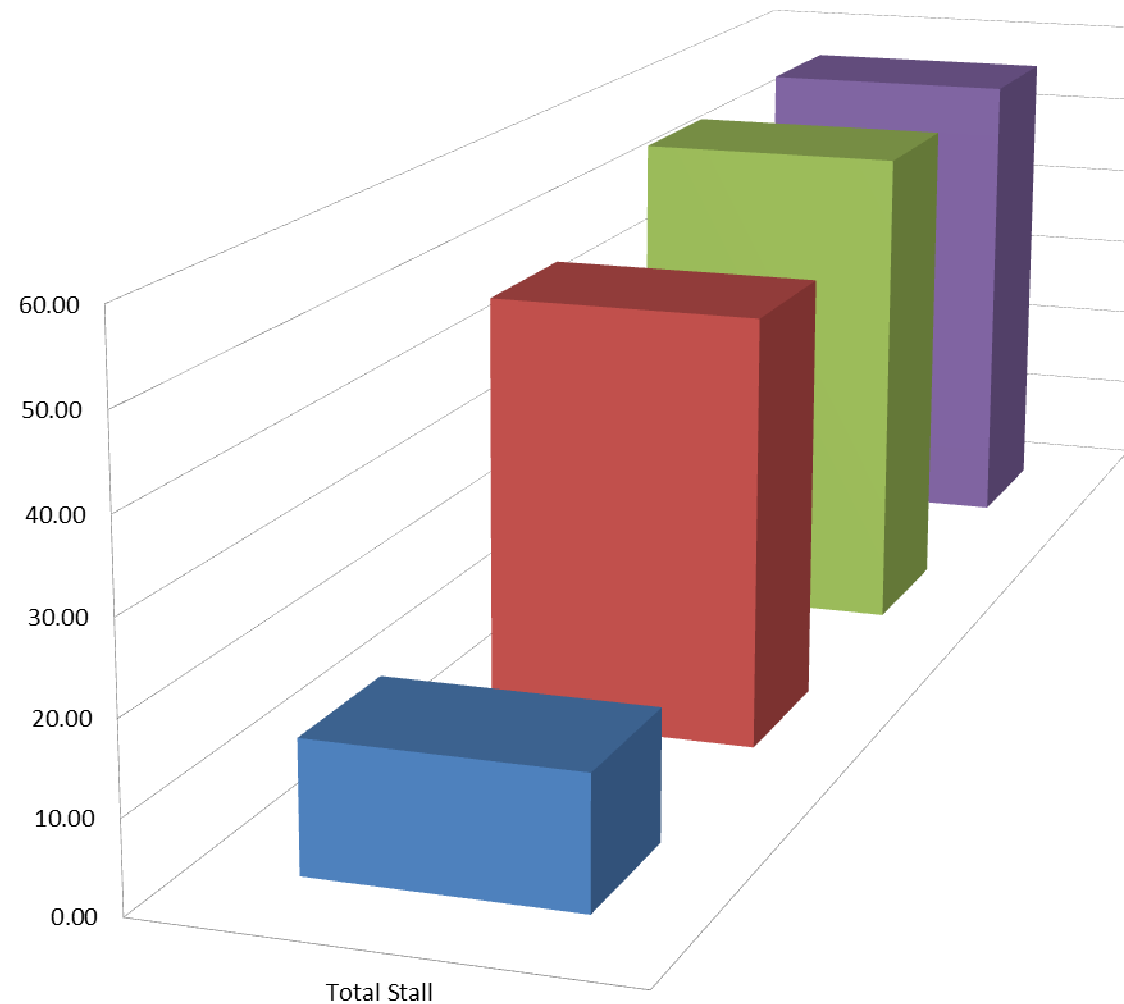
	Cold Stall Seconds	Warm stall Seconds	Hot stall Seconds	Commit Stall Seconds
Cold Synchronization	1.88	0.00	0.00	12.49
Warm Synchronization	0.01	46.52	0.00	1.04
Hot Synchronization	0.00	0.29	55.04	0.32
Commit Synchronization	0.00	0.00	0.48	55.89

Comments on Graph

- Warm, Hot, and Commit synchronization levels
 - Significant stall time for sending transactions
 - Small stall time for checkpoints
- Cold
 - Small stall time for sending transactions
 - Larger stall time for checkpoints – function of checkpoint interval

Synchronization Level versus Stall Time

0ms Delay, Checkpoint 100, Govenor Disabled



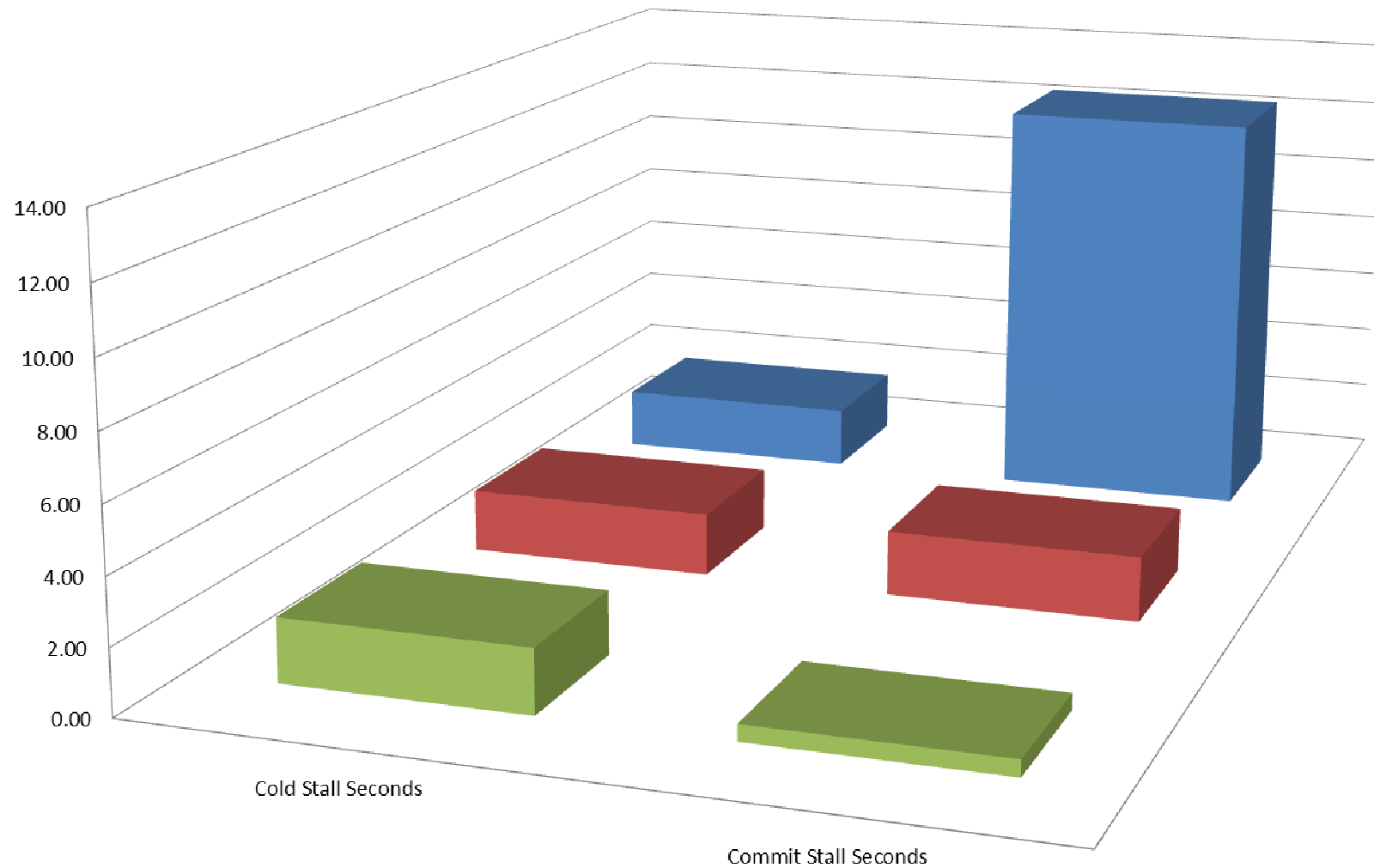
	Total Stall
Cold Synchronization	14.37
Warm Synchronization	47.58
Hot Synchronization	55.65
Commit Synchronization	56.38

Comments on Graph

- Total stall time for each synchronization level
- Stall time for cold synchronization much shorter

Checkpoint Frequency versus Stall Time

Cold Synchronization, 0ms Delay, Govenor Disabled

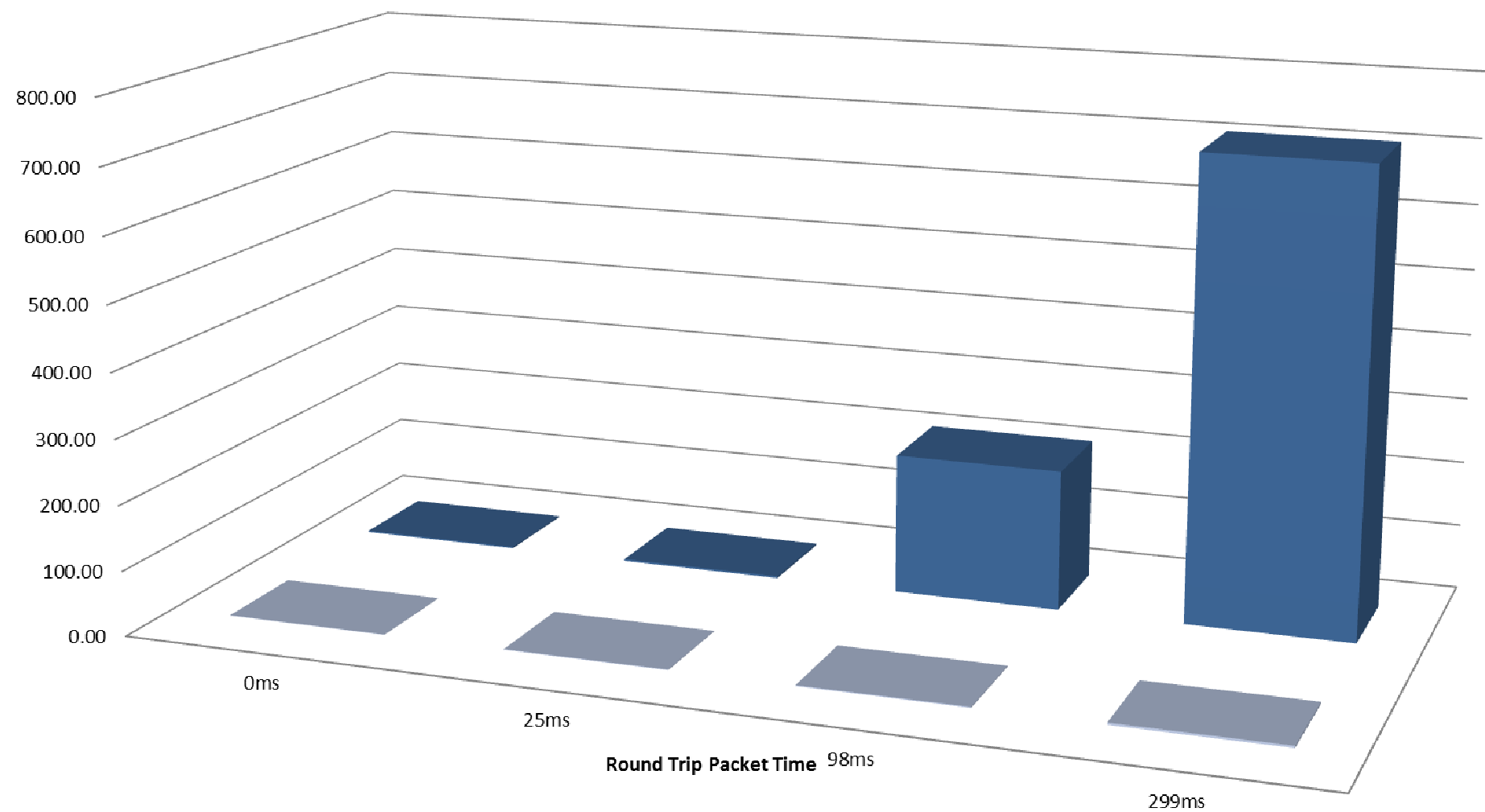


	Cold Stall Seconds	Commit Stall Seconds
heckpoint 10000	1.94	0.50
heckpoint 1024	1.92	1.98
heckpoint 100	1.88	12.49

Comments on Graph

- With smaller checkpoint interval, checkpoint stalls are more significant

Synchronization Cold, Checkpoint 10,000, Govenor Disabled



	0ms	25ms	98ms	299ms
Commit Stall	0.50	0.87	1.36	3.35
Cold Stall	1.94	2.74	216.60	711.74

Comments on Graph

- Stalls for checkpoints are insignificant
- Stalls for network packets very significant

What's Really Happening?

- AIJ Messages sent from ALS to LRS
 - Messages may exceed TCP/IP Maximum Transmission Unit (MTU – often 1514 bytes)
 - Potentially multiple packets per AIJ Message
 - Larger packet size would allow fewer TCP packets – Jumbo Frames
- Cold
 - Send multiple transactions → Standby
 - TCP ACK → Master (60 bytes)
- Warm, Hot, and Commit
 - Send multiple transactions → Standby
 - TCP ACK → Master (60 bytes)
 - Standby ACK → Master – Master stalls waiting for this ACK (566 bytes)
 - Sometimes TCP ACK → Standby (Huh?) (60 bytes)

Hot Standby Catch-up

- Log Catch-up Server (LCS)
 - Reads from AIJ on Master
 - Sends to standby
- ALS takes over when caught up

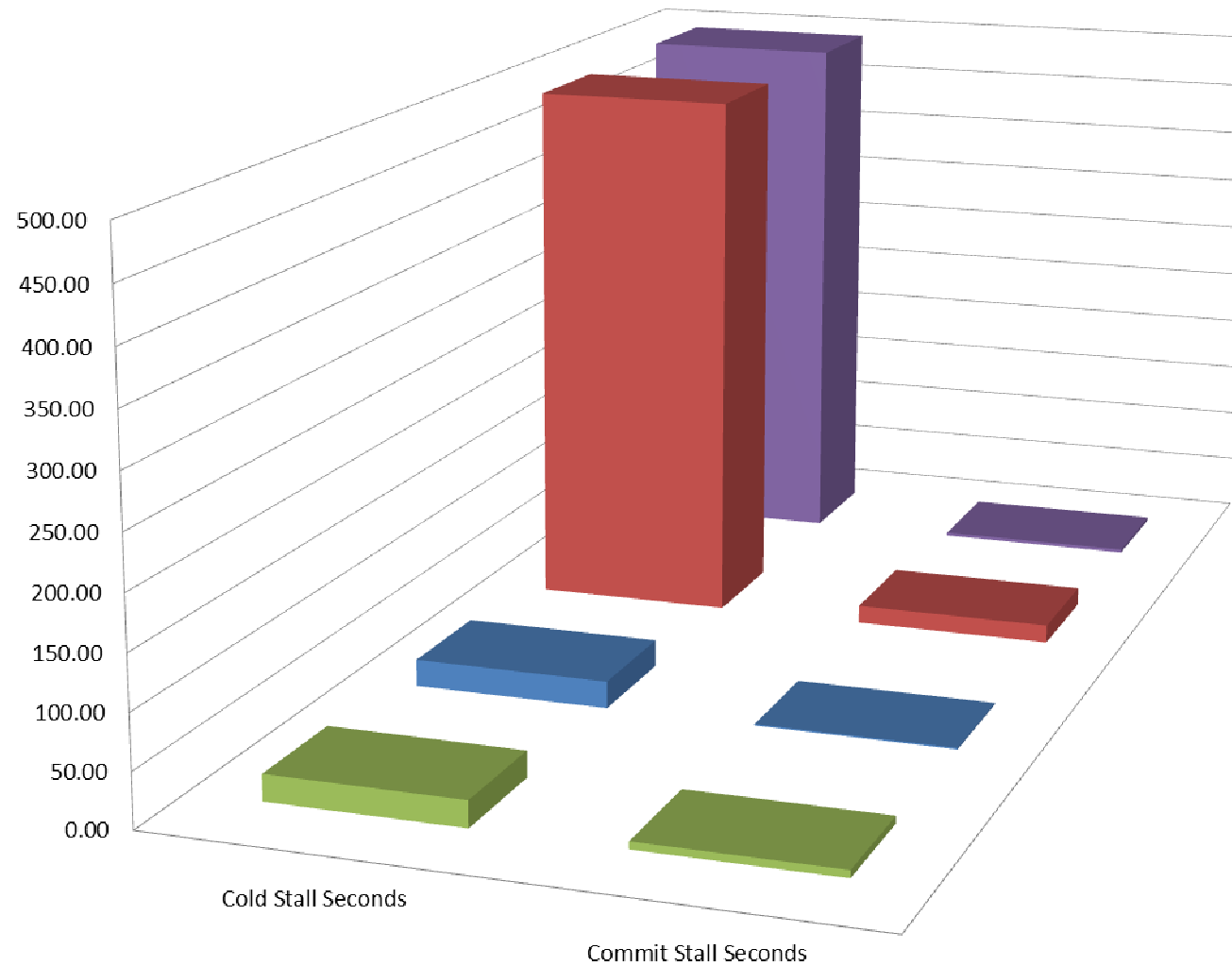
Catch-up Test

- Shut down hot standby
- 100,000 update transactions
 - 2x workload described earlier
- Start up hot standby
- Record statistics when hot standby catches up

Catch-up Test Variables

- Network Delays
 - 0ms
 - 98ms
- `rdm$bind_lcs_sync_commit_max`
 - 128
 - 10,000

Catch-up Test versus Stall Time and LCS Synchronization Cold Synchronization, Govenor Disabled



	Cold Stall Seconds	Commit Stall Seconds
ms, 128	24.12	6.93
ms, 10000	24.75	1.25
ms, 128	491.94	16.60
ms, 10000	495.66	3.01

Comments on Catch-up

- Data transfer most significant
- Checkpoints significant at 0ms

What am I forgetting?

- Ahh...yes...

- Memory

```
$ RMU/REPLICATE AFTER_JOURNAL CONFIGURE hs_standby -  
  /MASTER_ROOT=172.16.1.13::JCC_ROOT:[TOM.SQL_CLASS.MF_V72.MASTER]hs.rdb -  
  /buffers=10000 -  
  /governor=disabled
```

- More buffers reduce Read I/O

- Our tests did not stress the standby database

Hot Standby: Throughput Summary

- Use checkpoint of 1000 or higher
- Use LCS checkpoint of 10,000
- Faster network is better
- Consider using jumbo-frames
 - More data in single TCP packet
- Use more memory on standby side
- ...but what happens when the network fails

Hot Standby Network Failure

- Hot Standby Timeouts
- TCP Parameters that affect network failure
- DNS Lookups

Hot Standby Timeouts – Master

■ Example

```
rmu/replicate after_journal configure -  
  jcc_root:[keith.sql_class.mf_v72]mf_personnel -  
  /standby_root=192.168.27.14::jcc_root:[keith.sql_class.mf_v72_standby]mf_standby-  
  /Checkpoint=25000 -  
  /Connect_Timeout=5 -  
  /Log -  
  /Quiet_Point -  
  /Synchronization=cold -  
  /Transport="TCPIP"
```

■ Connect_Timeout – applies to startup

Hot Standby Timeouts – Standby

■ Example

```
$ rmu/replicate after_journal configure -  
    jcc_root:[keith.sql_class.mf_v72_standby]mf_standby -  
    /master_root=192.168.4.13::jcc_root:[keith.sql_class.mf_v72]mf_personnel -  
    /Checkpoint=25000 -  
    /Gap_Timeout=5 -  
    /Governor=Disabled -  
    /Log -  
    /Online
```

- Gap_Timeout – time Standby waits before shutting down hot standby

TCP Parameters

```
TCPIP> set protocol tcp/probe_timer=5/drop_count=5
```

```
TCPIP> disable service rdmaij72
```

```
TCPIP> enable service rdmaij72
```

```
TCPIP> show protocol tcp/parameter
```

TCP

Delay ACK:	enabled
------------	---------

Window scale:	enabled
---------------	---------

Drop count:	5
-------------	---

Probe timer:	5
--------------	---

Receive

Send

Push:	disabled
-------	----------

disabled

Quota:	61440
--------	-------

61440

```
TCPIP>
```

Hot Standby Network Failure

- TCP Protocol Examples
 - Probe: 5, Drop: 5
 - Probe: 3, Drop: 16
 - Probe: 10, Drop: 3
 - Probe: 7200, Drop: 8 (Default)
 - Probe: 8, Drop: 7
- Simulate network failure by unplugging cable

Stalls on Network Failure

Rate: 3.00 Seconds Active User Stall Messages Elapsed: 00:12:38.02

Page: 1 of 1 JCC_ROOT:[KEITH.SQL_CLASS.MF_V72]MF_PERSONNEL.RDB;1 Mode: Online

Process.ID	Elapsed....	T	Stall.reason.....	Lock.ID.
------------	-------------	---	-------------------	----------

000000AB:1s			RCS idle - waiting for work request	
-------------	--	--	-------------------------------------	--

000000AC:1s	00:00:16.07	-	connecting to remote database (0)	
-------------	-------------	---	-----------------------------------	--

000000AF:1s	00:00:16.17	-	connecting to remote database (0)	
-------------	-------------	---	-----------------------------------	--

000000AA:2	00:00:48.04	W	hibernating on AIJ submission	
------------	-------------	---	-------------------------------	--

- 000000AC – ALS
- 000000AF – LCS
- 000000AA – User process
 - User visible stall
 - Synchronization is Cold, so not sure why this stall occurs

TCP Parameters and Stalls

TCPIP			
PROBE _TIMER	DROP _COUNT	Ave HS Stall Seconds	Ave User Stall Seconds
5	3	25	25
5	5	35	35
5	8	48	48
3	8	30	30
3	16	54	54

Domain Name Services

- Integral to the functioning of the internet
 - Access to unknown resources
 - Access to dynamic resources
 - Transient connections
- Useful for large corporate networks
 - Changing resources
 - Expanding network
- Inappropriate for Hot Standby

Hot Standby: Domain Name Services

- Network connection is fixed
 - Primary and Secondary network settings
- Constant connection required
 - Must be shutdown for network changes
- Loss of network and DNS
 - DNS lookup timeout adds to Hot Standby shutdown stall
 - Adds to stall time for interactive users
 - Can still use name by adding name to hosts file

Summary

- Hot Standby Performance is very sensitive to
 - Network bandwidth
 - Network Latency
- In (almost) all cases, use:
 - Cold Synchronization
 - Disable Governor
 - Checkpoint ≥ 1000
 - LRS Checkpoint = 10,000

Summary (Cont.)

- Reduce Impact of network Failures
 - TCP Probe Timer and Drop Count parameters
 - Configure alternate network path
 - Understand implications of DNS lookups

Questions?



<http://www.jcc.com/>

Join the worldwide Rdb community.
Send mail to

OracleRdb-request@JCC.com

with “SUBSCRIBE” in the body of
the message.

For more information send mail to info@jcc.com