IDUG

Customer deployed Db2 Resiliency architectures

Regional DB2 Users Group

Dale McInnis, IBM Canada Ltd.

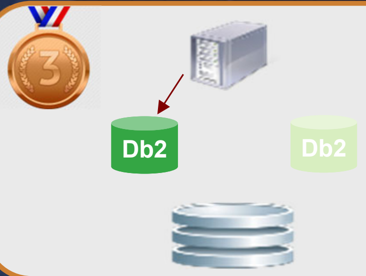
Who am I

- Certified Technical Specialist (level 3) on the America's Data and AI Technical Sales team
- 25 years at the Db2 LUW High Availability Architect
- Vice Chair for Canadian Information Processing Society – Ontario Canada
  - Information Systems Professional (I.S.P) Certified
  - Information Technology Certified Professional (ITCP)
- IBM WorldWide Performance and Availability Community co-leader
- IBM Academy of Technology Member
- IBM Academy PREVAIL Conference Lead
- 11 Patents in database resiliency
- Published author
- IDUG Speaker Hall of Fame member

Agenda

- Feature Positioning
- pureScale
- Replication
- HADR
- Backup and Recovery

Db2 HA Options : 24x7x365 Continuous Availability for OLTP

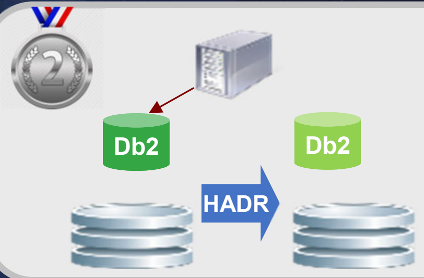


Integrated Clustering

- Active/passive
- Hot/cold, with failover typically in minutes
- Easy to setup

99% Availability

- Db2 ships with integrated TSA failover software
- No additional licensing required

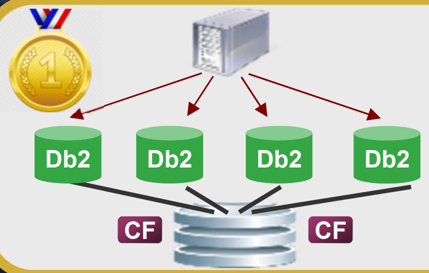


HADR

- Active/passive or active/active (with Reads on Standby)
- Hot/warm or hot/hot (with RoS), with failover typically less than one minute
- Easy to setup

99.99% Availability

- Db2 ships with integrated TSA
- Minimal licensing (full licensing required if standby is active)
- Perform system and database updates with miminal interruption



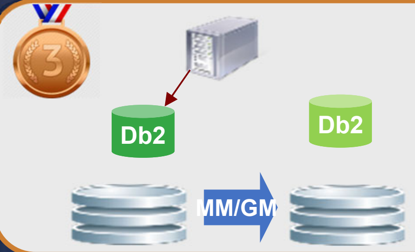
pureScale

- Active/active
- Hot/hot, with automatic and online failover
- Integrated solution includes CFs, clustering, and shared data access

99.99+% Availability

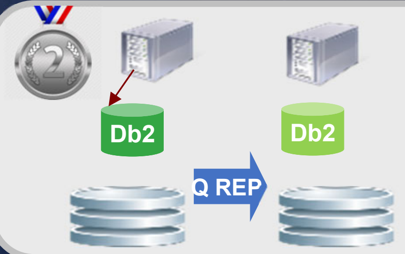
- Perform system and database updates in rolling online fashion
- Also works with HADR (single target)
- Geographically Dispersed Cluster for multi-site deployment

Db2 Disaster Recovery Options



Log Shipping / Storage Based Replication

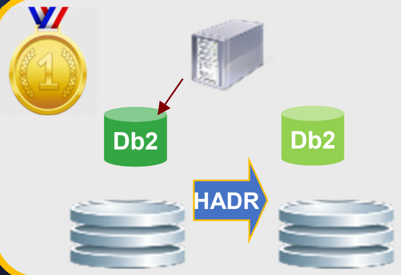
- Active/passive
- Hot/cold, with failover typically in minutes
- Asynchronous
- Complete DB replication only



Logical Replication

- Active/active (updates require conflict resolution / avoidance
- Hot/Hot (Instant failover)
- Asynchronous

- Added flexibility
  - Subsetting
  - Different versions
  - Different topology
  - Multiple standby
  - Time delay
- DDL considerations



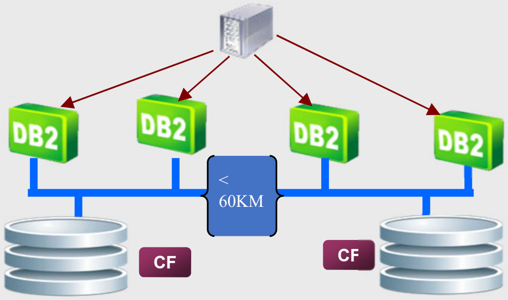
HADR

- Active/passive or active/active (with Reads on Standby)
- Hot/warm or hot/hot (with RoS), with failover typically less than one minute
- Easy to setup

- Complete DB Replication
- Minimal licensing (full licensing required if standby is active)
- Time Delay
- Perform system and database updates minimal interruption

Situational Platinum

GDPC



- Active / active (fully coherent)
- Hot / hot (**online** failover)
- Synchronous
- Complete DB replication
- Continuous testing of DR site
- Distance limitations
- Only available through lab services

Agenda

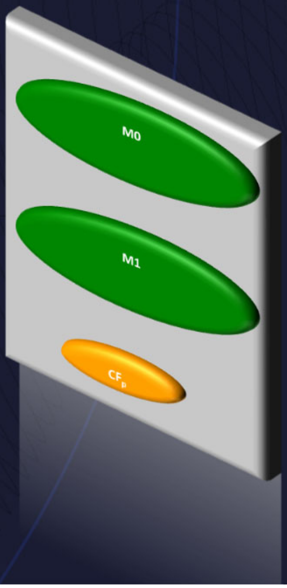
- Positioning
- pureScale
- Replication
- HADR
- Backup and Recovery

How to deploy pureScale with minimal amount of H/W but still be resilience

- Customer interest in pureScale is growing at a very rapid rate
- Challenge is how to do deploy with minimal cost but still retain near continuous availability
- Production Deployment Recommendations:
  - Hosts:
    - Minimum # of physical hosts is 2
  - Network:
    - Do not deploy on less than 10GB
    - Use a private network (preferably on a separate switch) for the interconnect between Members and CF
    - Eliminate SPOF and use 2 switches
  - Storage
    - Eliminate SPOF be deploying GPFS Sync Replication

Proposed H/W and LPAR Layout on each frame

9117-MMB x 2
Available Cores: 30
Available Memory: 1TB
Available slots: 16



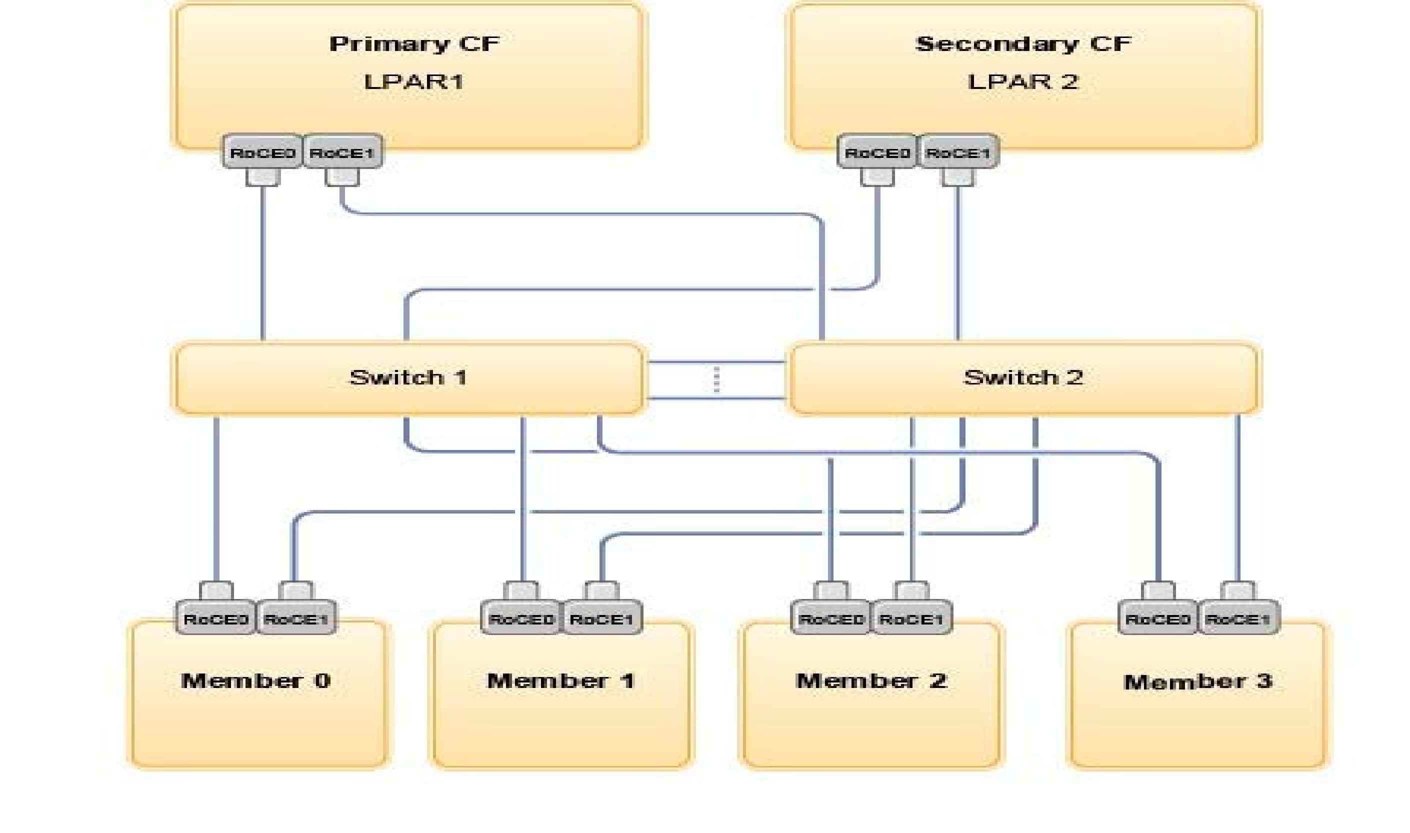
1 CF and 2 pureScale members per frame
Each CF = 6 Cores + 128 GB RAM
Each Member = 12 Cores + 64 GB RAM

CF LPAR (CF0)
6 Cores
128 GB RAM

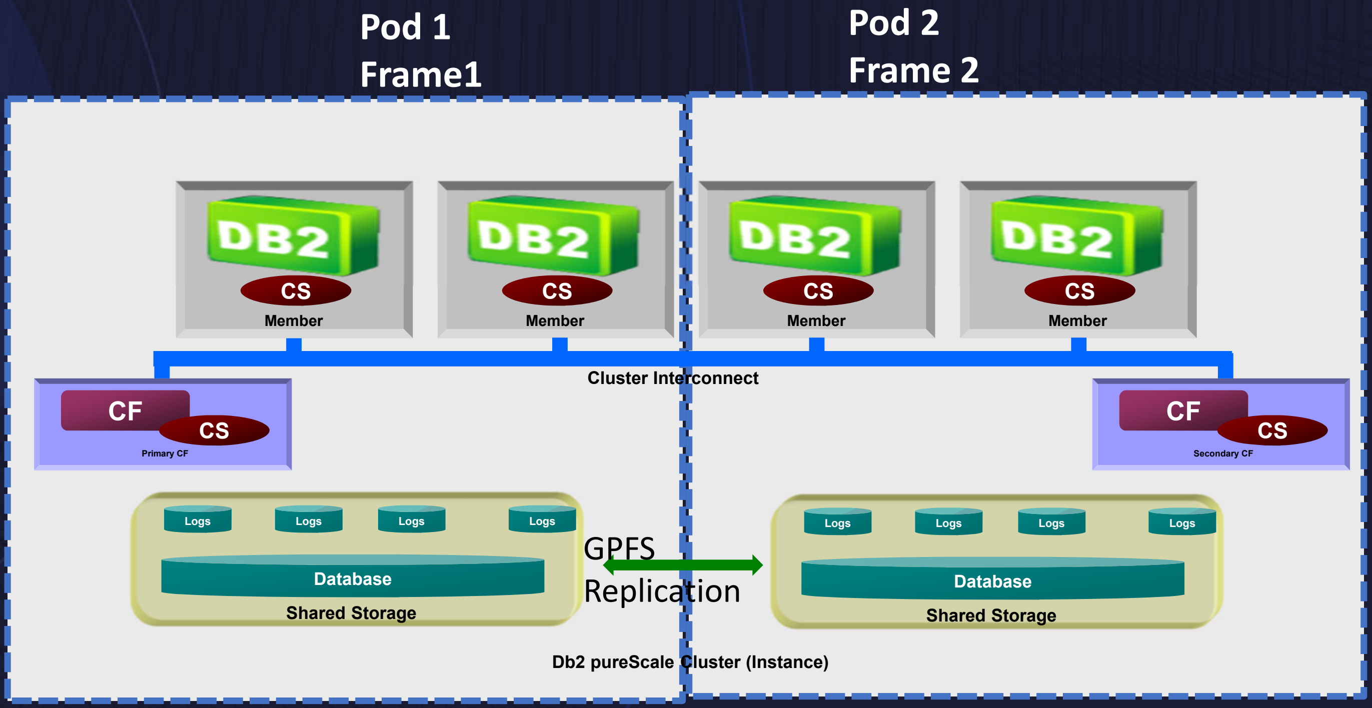
Db2 Member LPAR (M0)
12 Cores
64 GB RAM

Db2 Member LPAR (M1)
12 Cores
64 GB RAM

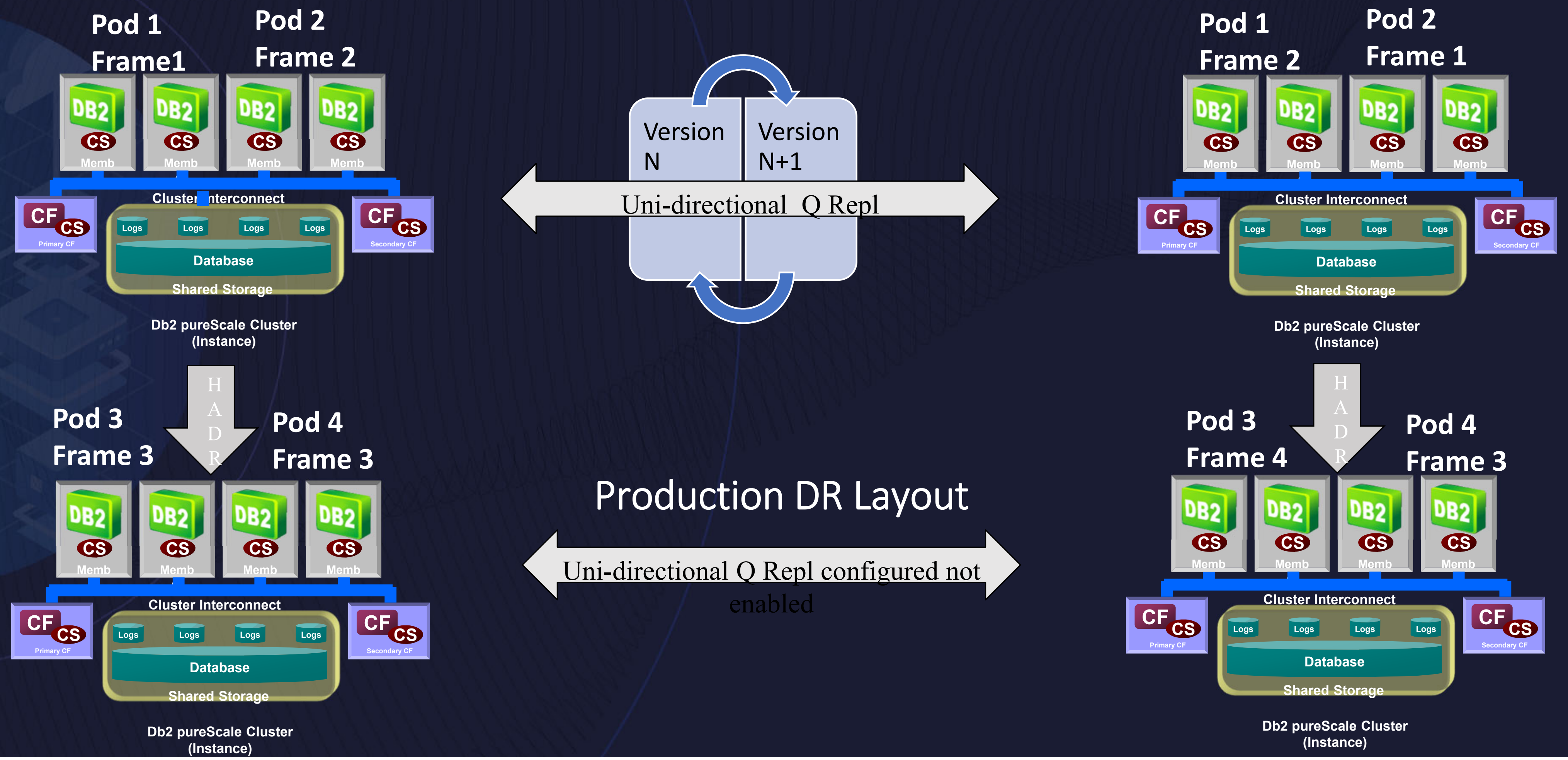
PRODUCTION: pureScale cluster spans two 9117-MMB



Layout of Production environment across Pod1 and Pod2



Production Layout



Pod 1 Frame1

Pod 2 Frame 2

Db2 pureScale Cluster (Instance)

Version N

Version N+1

Uni-directional Q Repl

Pod 1 Frame 2

Pod 2 Frame 1

Db2 pureScale Cluster (Instance)

HADR

Pod 3 Frame 3

Pod 4 Frame 3

Production DR Layout

Uni-directional Q Repl configured not enabled

HADR

Pod 3 Frame 4

Pod 4 Frame 3

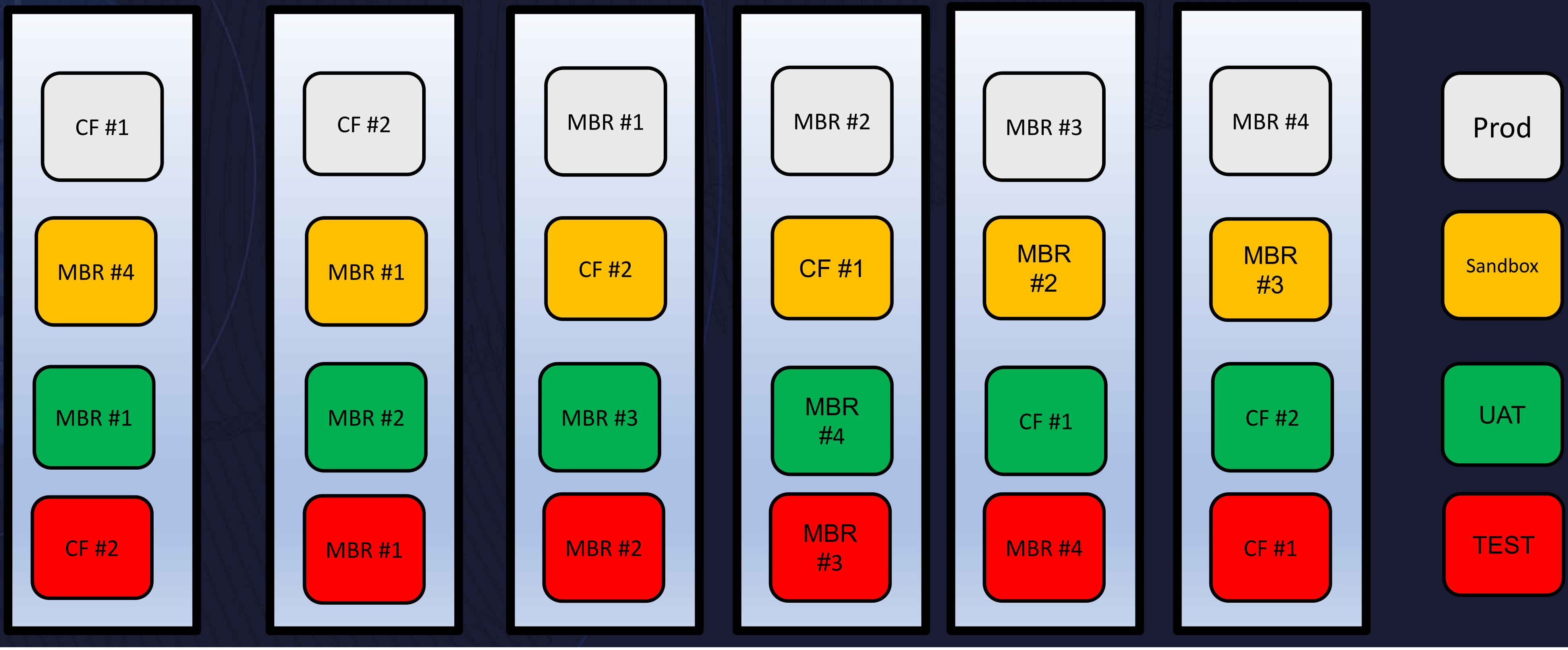
Db2 pureScale Cluster (Instance)

Db2 pureScale Cluster (Instance)

Deploying pureScale databases on shared hardware

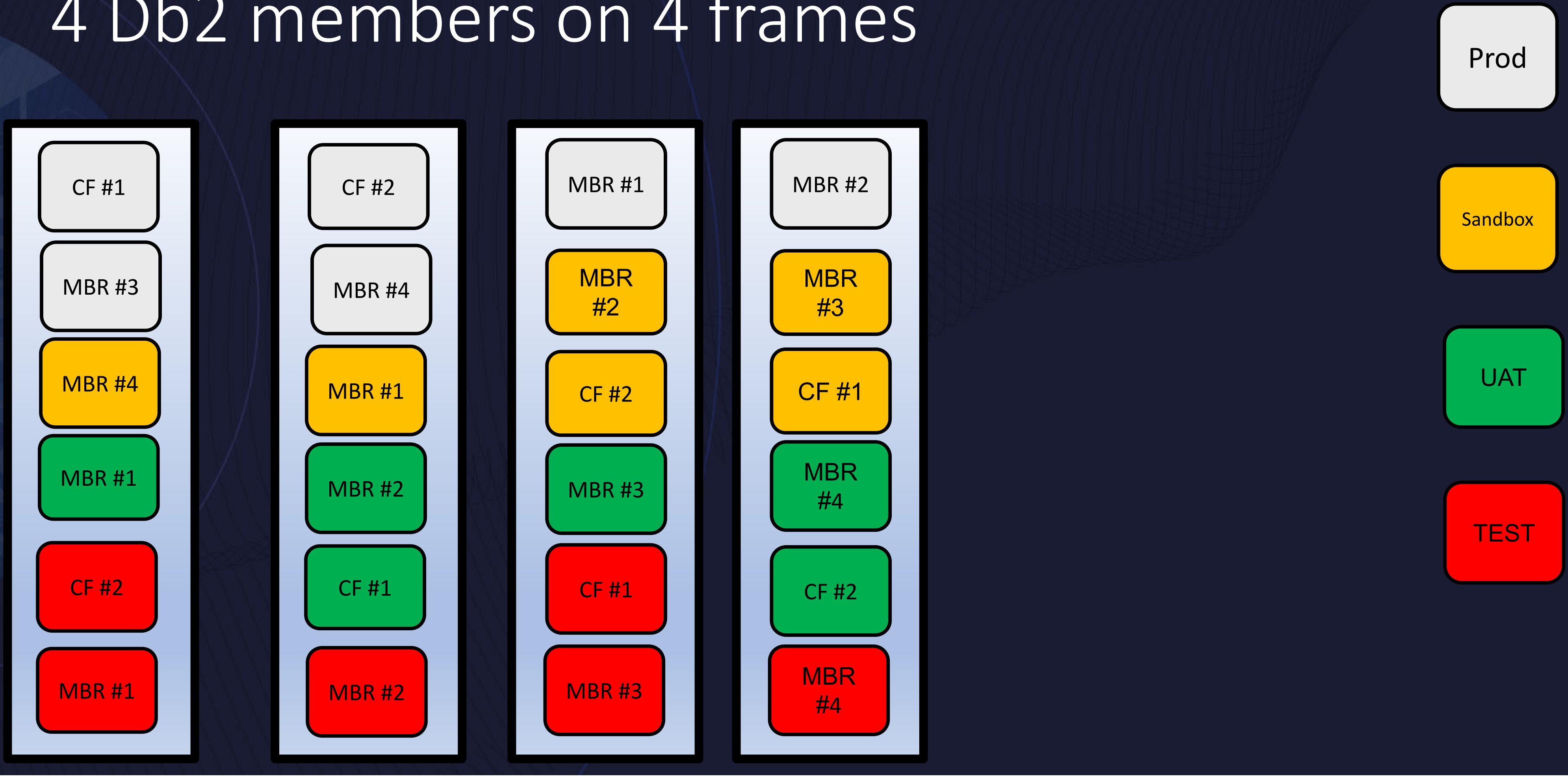
- Try to ensure that the CFs are on separate frames
- Spread the load around if deploying multiple databases
- Use the fastest possible network

H/W to LPAR mapping – 4 Environments with 4 Db2 members on 6 frames



CF #1	CF #2	MBR #1	MBR #2	MBR #3	MBR #4	Prod
MBR #4	MBR #1	CF #2	CF #1	MBR #2	MBR #3	Sandbox
MBR #1	MBR #2	MBR #3	MBR #4	CF #1	CF #2	UAT
CF #2	MBR #1	MBR #2	MBR #3	MBR #4	CF #1	TEST

H/W to LPAR mapping – 4 Environments with 4 Db2 members on 4 frames



⬜ CF #1	⬜ CF #2	⬜ MBR #1	⬜ MBR #2
⬜ MBR #3	⬜ MBR #4	🟨 MBR #2	🟨 MBR #3
🟨 MBR #4	🟨 MBR #1	🟨 CF #2	🟨 CF #1
🟩 MBR #1	🟩 MBR #2	🟩 MBR #3	🟩 MBR #4
🟥 CF #2	🟩 CF #1	🟥 CF #1	🟩 CF #2
🟥 MBR #1	🟥 MBR #2	🟥 MBR #3	🟥 MBR #4

⬜ Prod

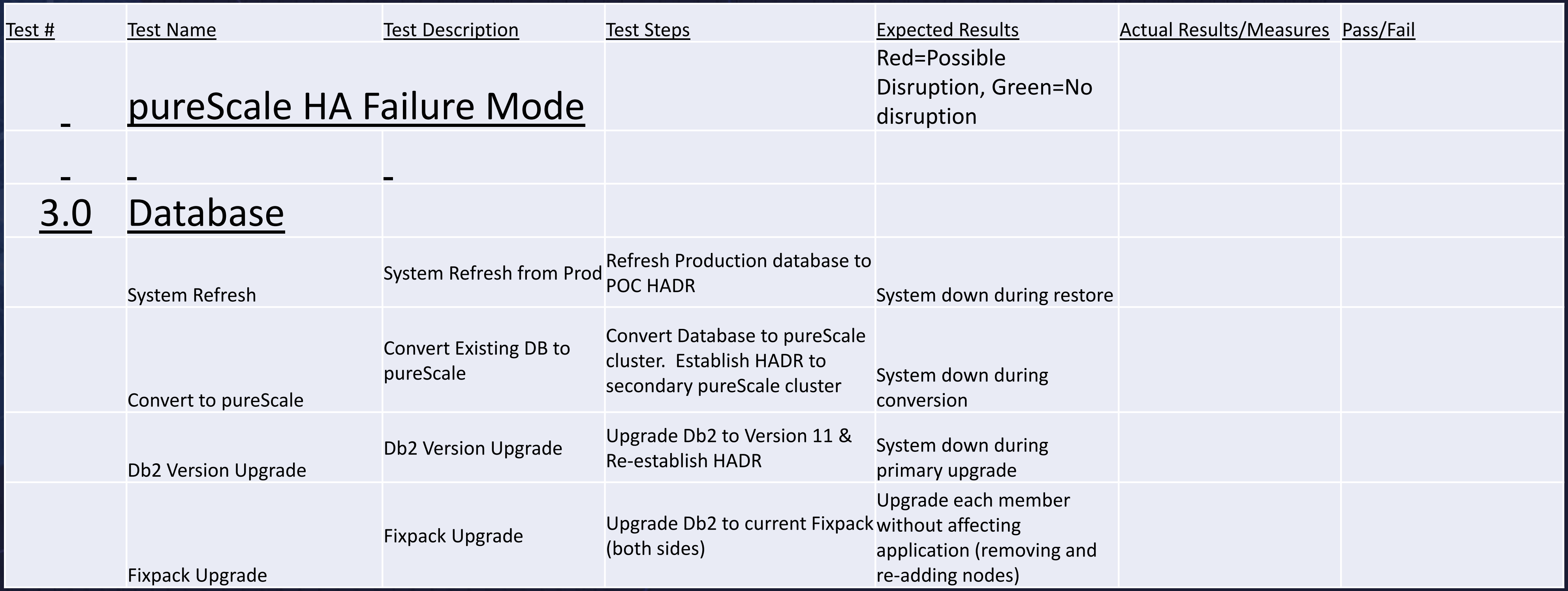
🟨 Sandbox

🟩 UAT

🟥 TEST

pureScale test scenarios

- I have an excel spreadsheet with most of the scenarios that should be tested.



Test #	Test Name	Test Description	Test Steps	Expected Results	Actual Results/Measures	Pass/Fail
_	pureScale HA Failure Mode			Red=Possible Disruption, Green=No disruption		
-	-	-				
3.0	Database					
	System Refresh	System Refresh from Prod	Refresh Production database to POC HADR	System down during restore		
	Convert to pureScale	Convert Existing DB to pureScale	Convert Database to pureScale cluster. Establish HADR to secondary pureScale cluster	System down during conversion		
	Db2 Version Upgrade	Db2 Version Upgrade	Upgrade Db2 to Version 11 & Re-establish HADR	System down during primary upgrade		
	Fixpack Upgrade	Fixpack Upgrade	Upgrade Db2 to current Fixpack (both sides)	Upgrade each member without affecting application (removing and re-adding nodes)		

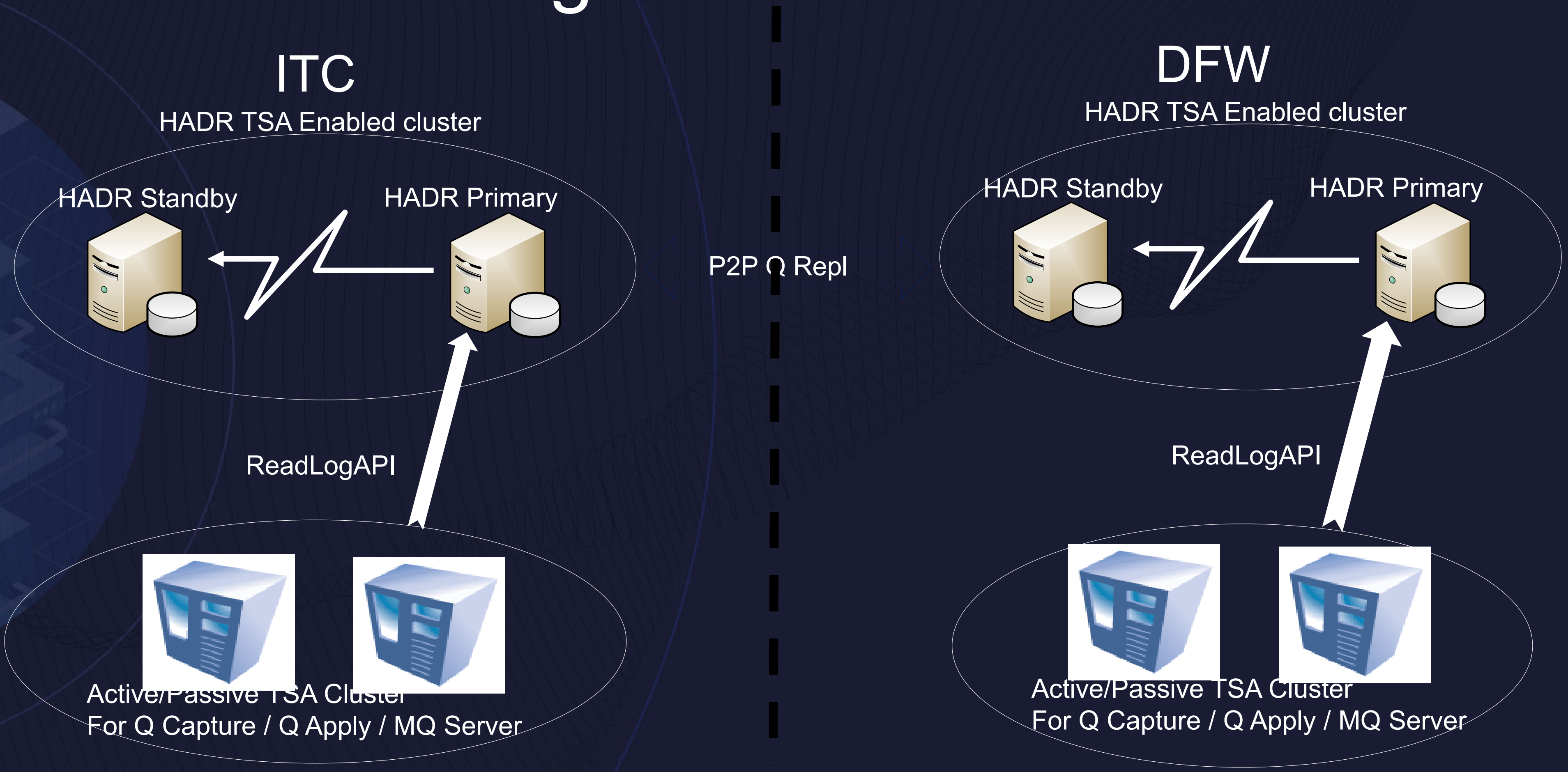
Agenda

- Positioning
- pureScale
- Replication
- HADR
- Backup and Recovery

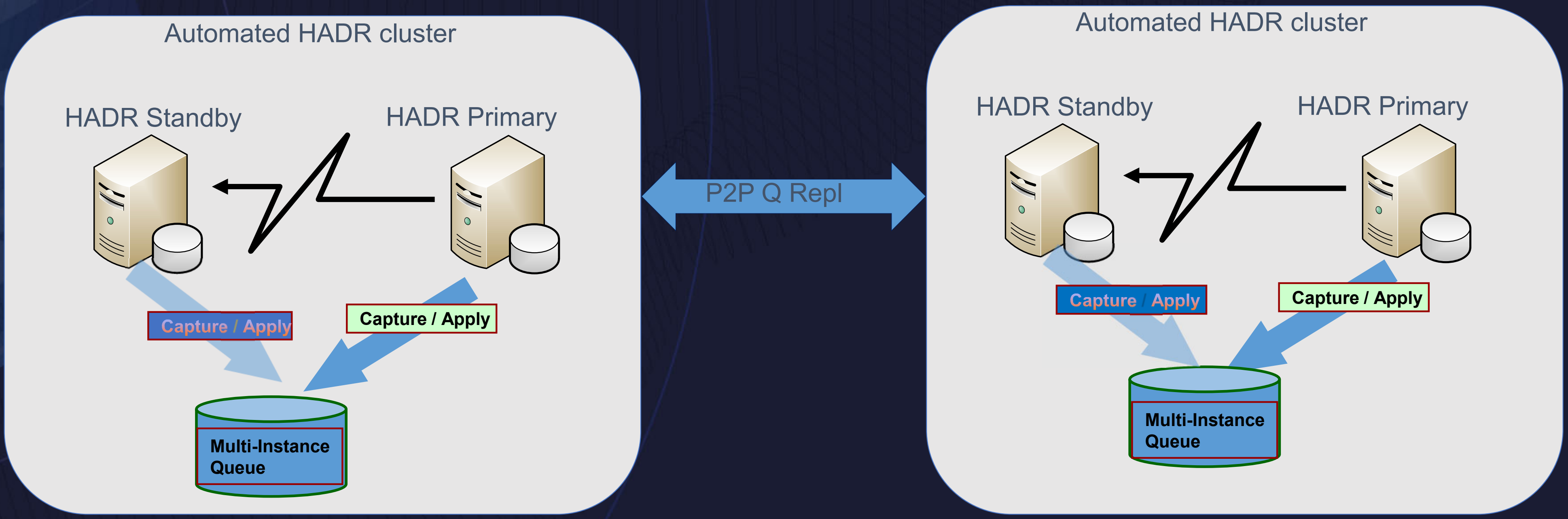
Making Q Repl resilient

- Customer deployed Q Repl separated from the database server
  - Thought was that if the DB failed there would be no need to restart Q Repl
  - Capture and Apply ran on a remote node
  - The Q Manager was setup on a Shared Disk TSA cluster
- Poor performance with using the remote read log API
- Several problems with restarting capture and apply

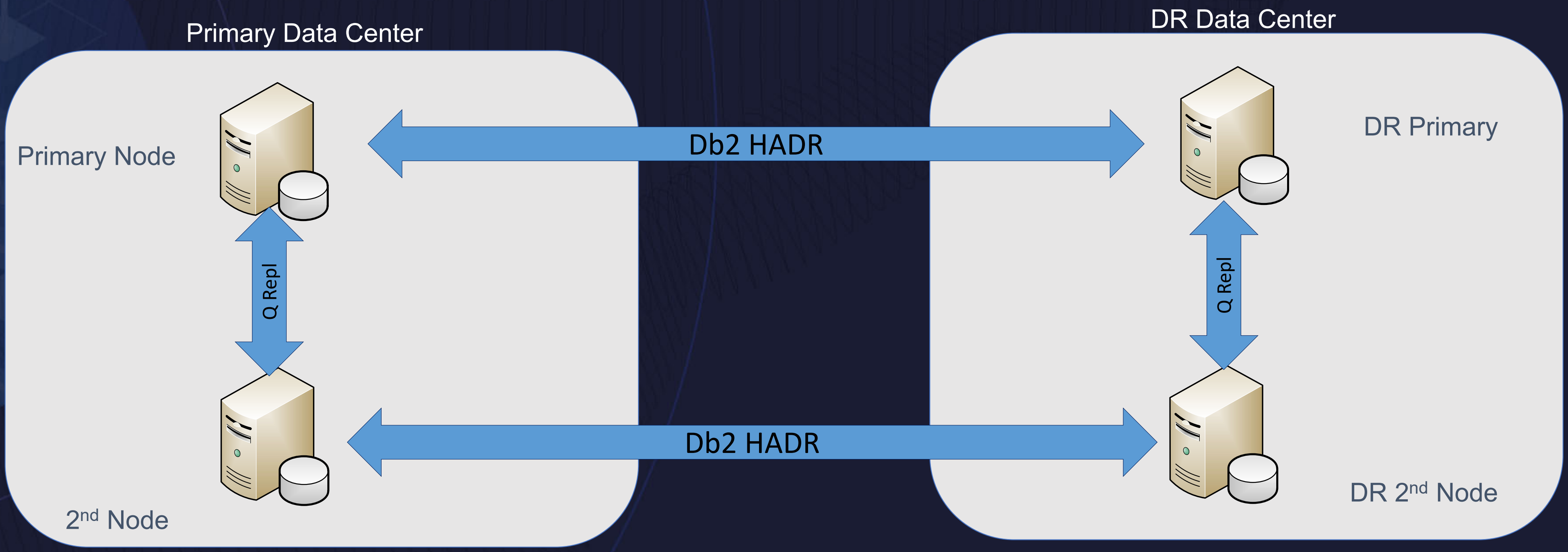
Active-Active Design



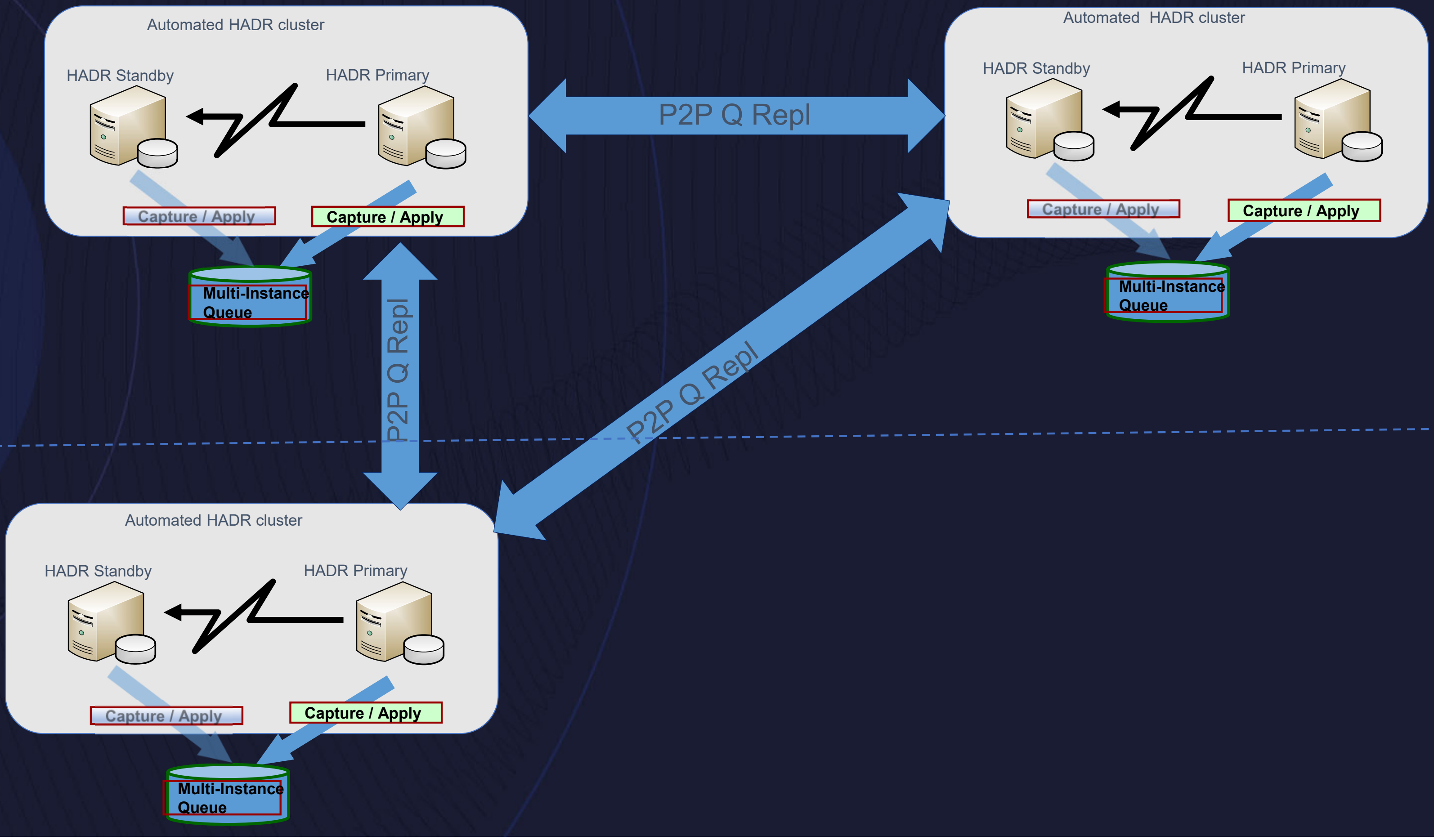
Local Active-Active Design Modified



Local Active-Active Design Modified



3 way(Local + DR) Active-Active Design



Agenda

- Positioning
- pureScale
- Replication
- HADR
- Backup and Recovery

HADR Performance

- **What do I do when I have a HADR performance issue?**
- **Check the Hardware**
  - Both sides should be identical: Number of active cores and amount of memory
  - Remember to check the storage arrays, preferably SD/NVMe for transactions logs be identical on all nodes
- **Check the interconnect**
  - Use hadr simulator to find out how much bandwidth you really have, versus what you are told you have
- **Run db2pd on both the primary and standby**

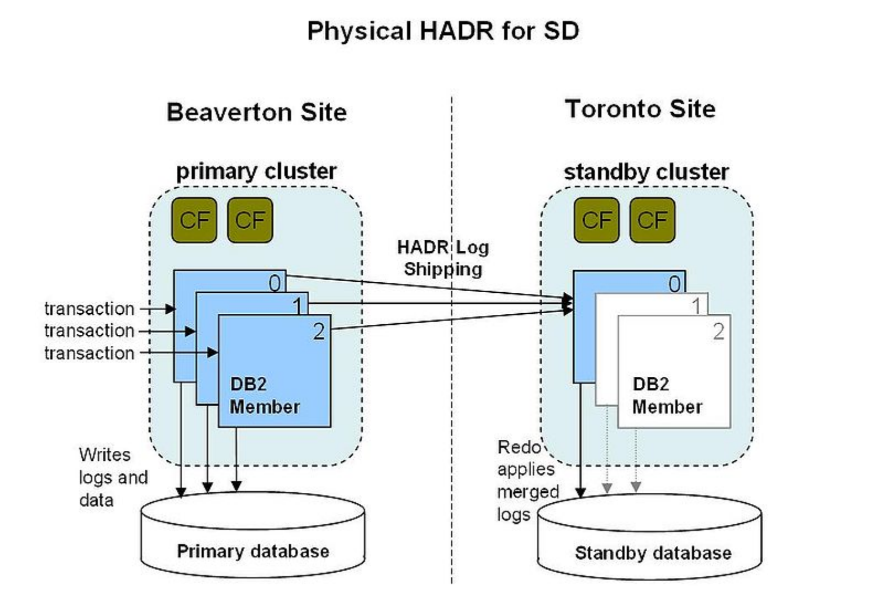
Identify bottle necks

For every problem there is an opportunity! anonymous

- HADR data flow is as follows
  - Primary generates log pages
  - Primary **sends** log pages to the standby
  - Standby **receives** log pages
  - Standby **writes** received log pages to disk and sends acknowledgment
  - Standby **replays** written log pages

- The operations on the critical path are send - receive - write - replay

- There are two common bottlenecks along the data flow:
  1. Slow network
  2. Slow standby



Collect monitoring information

- To gather information for diagnostics, monitor HADR at regular intervals
- Example shell script:

```
while :
do
   issue "db2pd -hadr" command on primary
   record output

   issue "db2pd -hadr" command on standby
   record output

   sleep 60
done
```

- db2pd is preferred over MON_GET_HADR because
  - it is light weight
  - can run on a standby without reads on standby enabled

Differentiate slow network vs slow standby

- Slow standby
  - Standby log processing is slow or blocked
  - Usually the problem is with the log replay being blocked or not able to keep up
  - Can be a storage issue
  - Standby spool and HADR gap grows
- Slow network
  - Log data is not shipped to standby host fast enough
  - Standby database often waits for more data
  - Standby spool remains very low

How to determine if the standby is the bottleneck

- Monitor the HADR_FLAGS
  - STANDBY_RECV_BLOCKED: indicates a slow standby

- standby receive buffer is full?
  - STANDBY_RECV_BUF_PERCENT is 100%
  - Turn on spooling to resolve

- Spooling has reached its spool limit?
  - STANDBY_SPOOL_PERCENT is 100%
  - Increase the amount of space available to spool

- When the standby logging device is full
  - STANDBY_LOG_DEVICE_FULL flag is set in the HADR_FLAGS
  - Increase the log disk on the standby

- In rare scenarios log writes on the standby is to blame
  - When replay is slower than receiving, more and more log data queues in the buffer and spool
  - Eventually, buffer or spool gets full and cannot receive more data
  - Measure the disk speed and log write size

Tuning a slow standby

- Hardware Utilization
  - Check hardware bottleneck on standby using tools like vmstat and iostats
  - It is recommended that primary and standby have the same hardware

- Number of Replay Threads
  - Recovery is done in parallel using multiple worker threads, which defaults to the number of physical CPUs
  - When there are a large number of CPUs, the default may be too high
  - To check the number of threads used, look for lines like this in db2diag.log:
    "Using parallel recovery with 6 agents 4 QSets 20 queues and 0 chunks"
  - To tune the number of threads, use Db2 registry variable Db2BPVARS:
    db2set Db2BPVARS=<path to buffer pool config file>
    In the config file, put this line:
    PREC_NUM_AGENTS=<number of threads>
  - You may need to experiment with a few numbers to find out the best one.

- Reads on Standby
  - When reads on standby is enabled, read queries will compete against replay thread for resources
  - Experiment with disabling reads on standby and gauge the impact

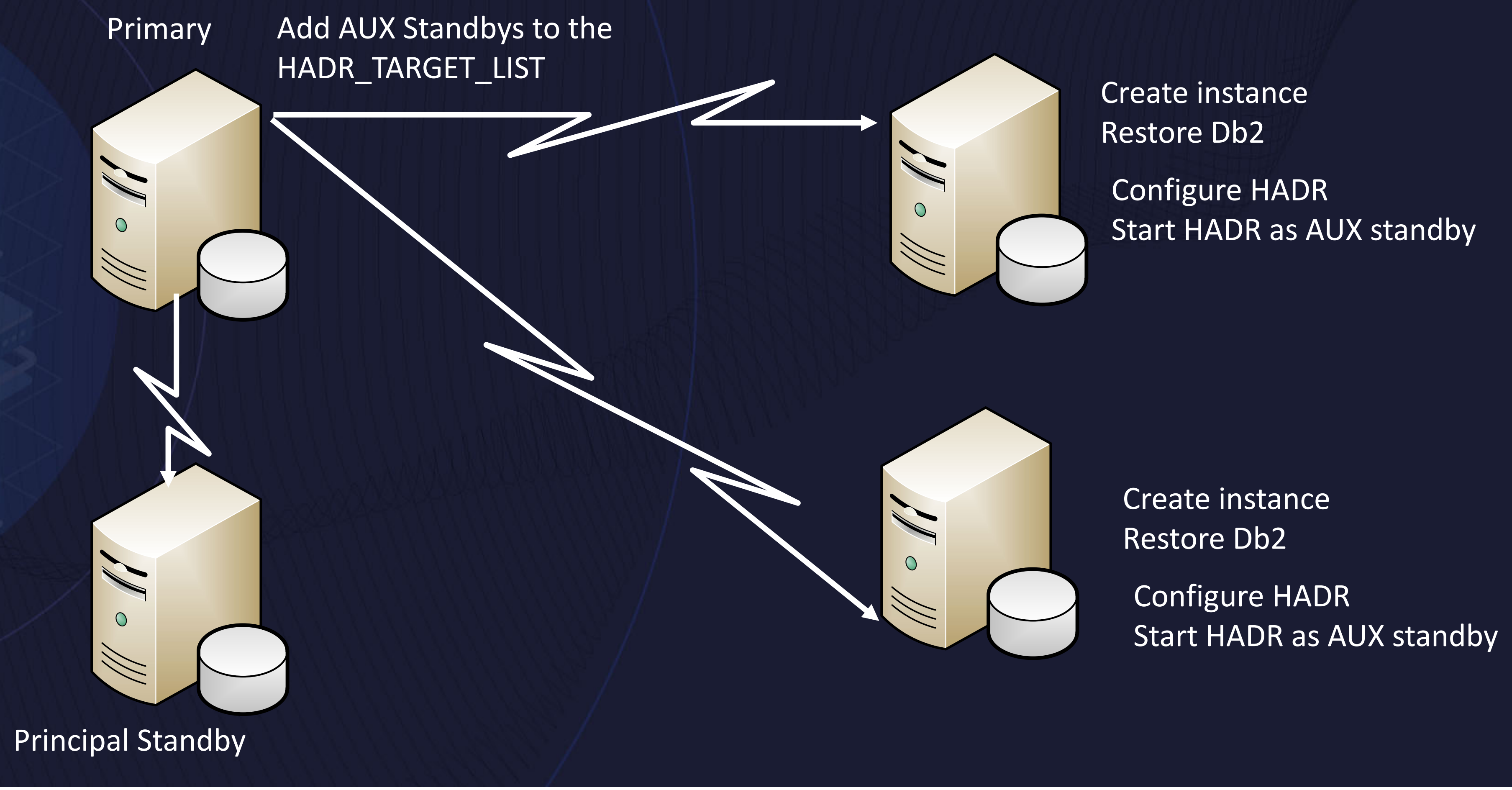
How to determine of the network is the bottleneck

- **Benchmark network speed**
  - Use HADR Simulator, or ping, or various other tools
- **Tuning a slow network**:
  - Use a less demanding HADR sync mode, avoid ASYNC Mode
  - Use the peer wait limit (Db2_HADR_PEER_WAIT_LIMIT) to cap the length of time the primary is blocked waiting for standby
  - Tune or upgrade network if possible
- **Time-out valves:**
  - HADR_PEER_WINDOW: defines primary database behavior upon connection loss
  - HADR_TIMEOUT: puts a limit on network failure detection time.
  - Db2_HADR_PEER_WAIT_LIMIT: limit puts a limit on log write wait time

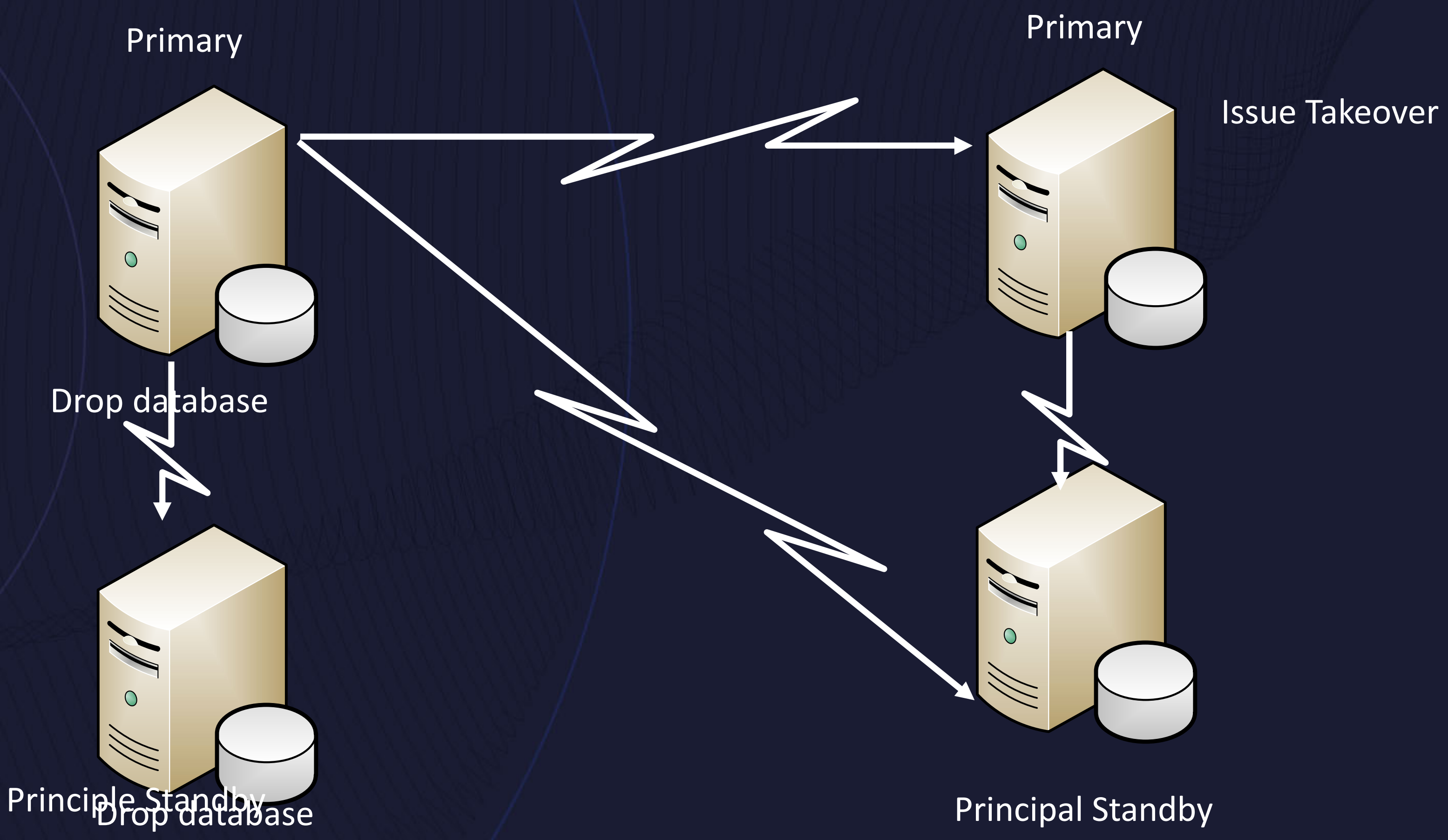
Rehosting – upgrading hardware, moving to a new data center, …

- If you are using an HADR ESE system then
  - Create an instance on the new H/W (HostC+HostD)
  - Take an online backup of the existing DB (HostA)
  - Restore on the new H/W (HostC+HostD)
  - Configure the new DBs(HostC+HostD) as an auxiliary standby for the original DB on HostA
    - Set the HADR_TARGET_LIST to reflect the new topology (Only HostC+HostD)
  - Configure the original DB add HostC+HostD as Auxiliary servers
  - Once the HADR systems are in peer state, issue a “normal” takeover
    - This is a zero-data loss roll reversal
    - Since HostA is not in the target list it will be orphaned
  - The original (HostA+HostB) database can be dropped

Rehosting – using HADR on an existing HADR environment



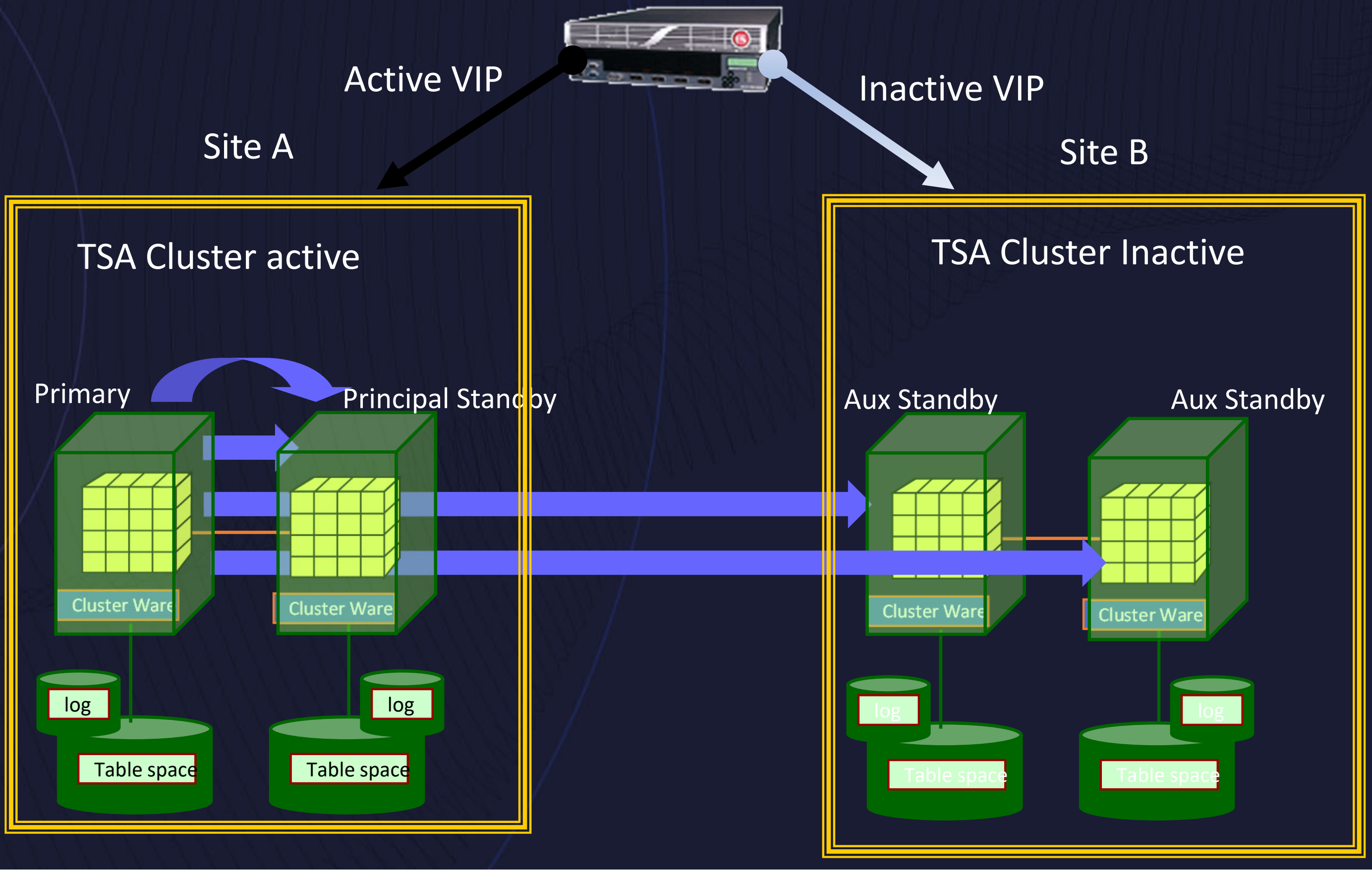
Rehosting – using HADR on an existing HADR environment



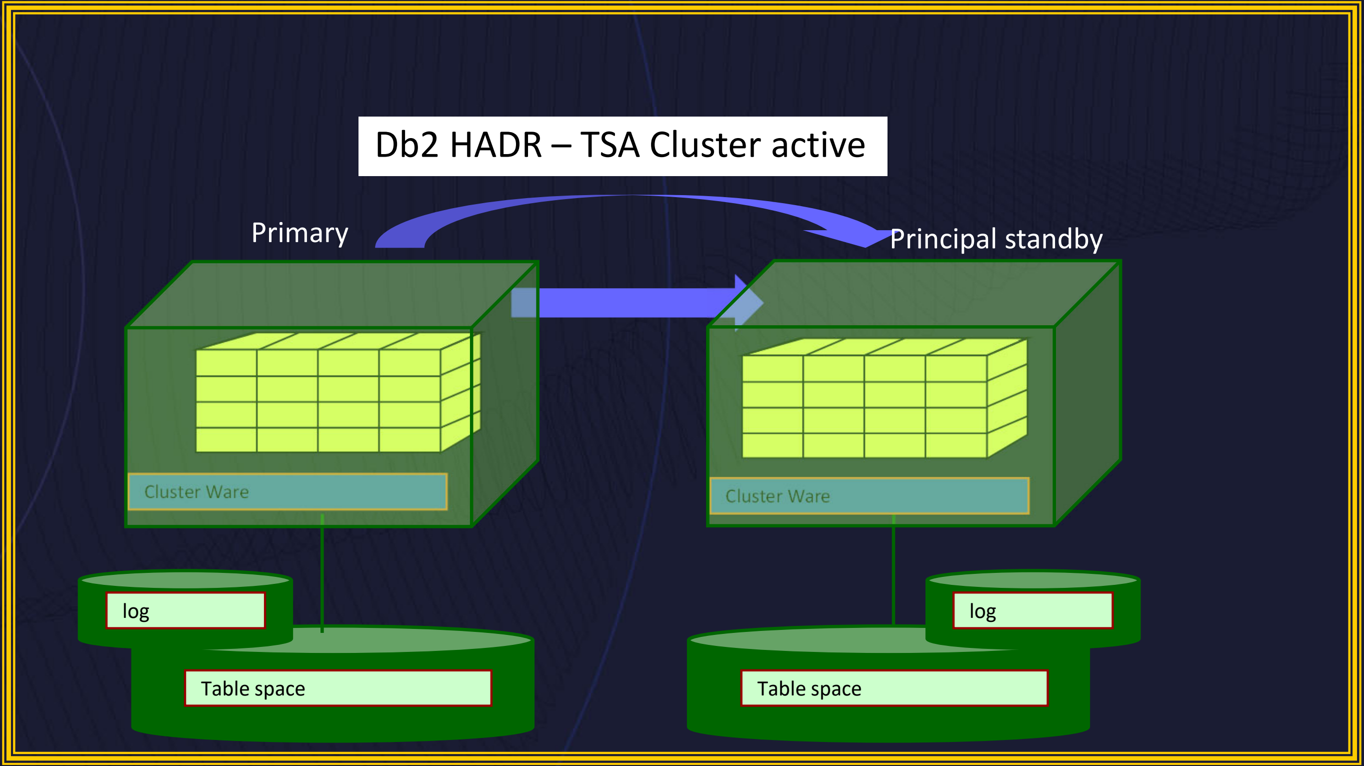
Workload Swap Requirement

- There is a desire to “workload swap” periodically, moving all workload to site B and running there for an extended period of time
- When the workload is swapped (scheduled event) there is a need for automation (TSA/MP) to be enabled.
- Is it possible to “preconfigure” TSA on site B but have it disabled.
- As part of the workload swap OR in the case of a real disaster TSA would be enabled manually.

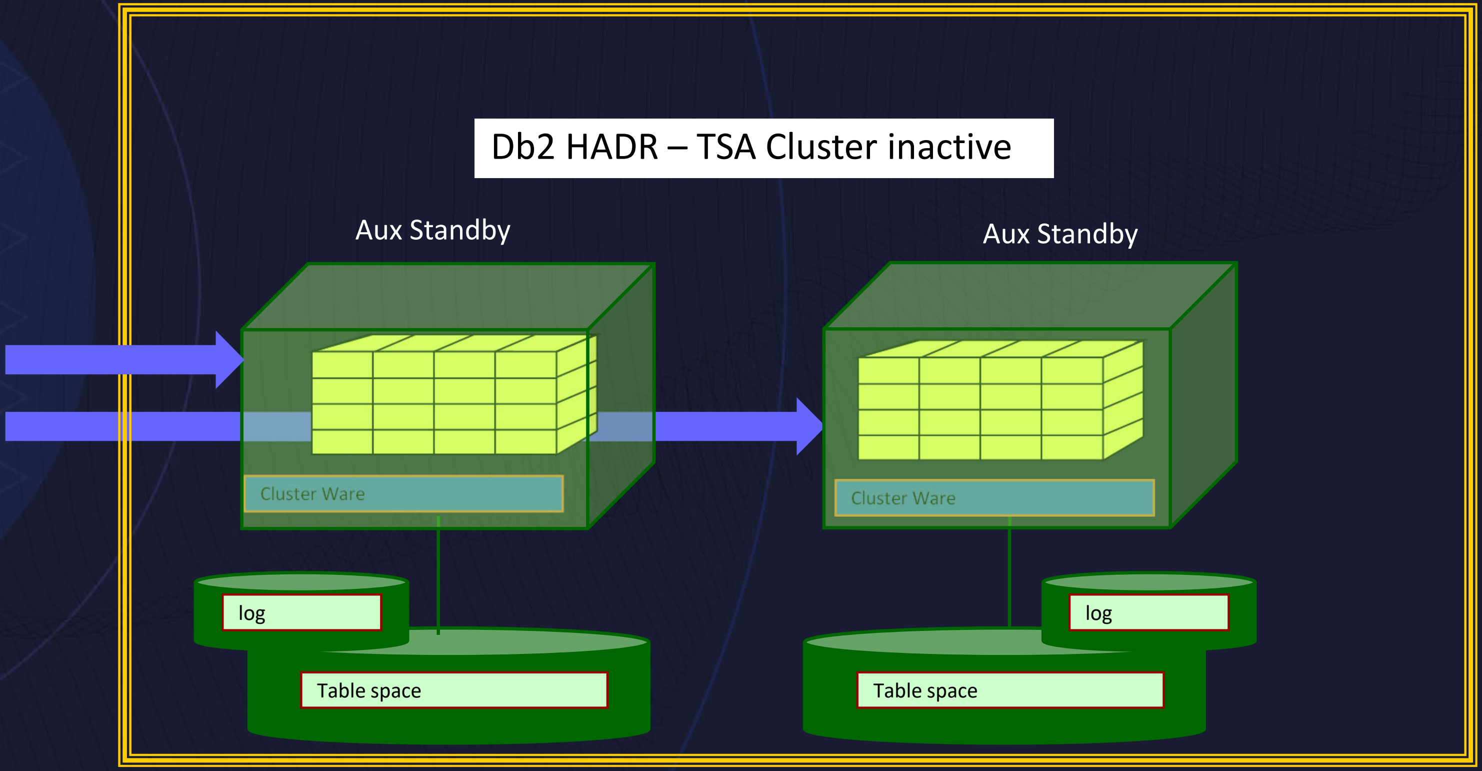
4 node HADR environment



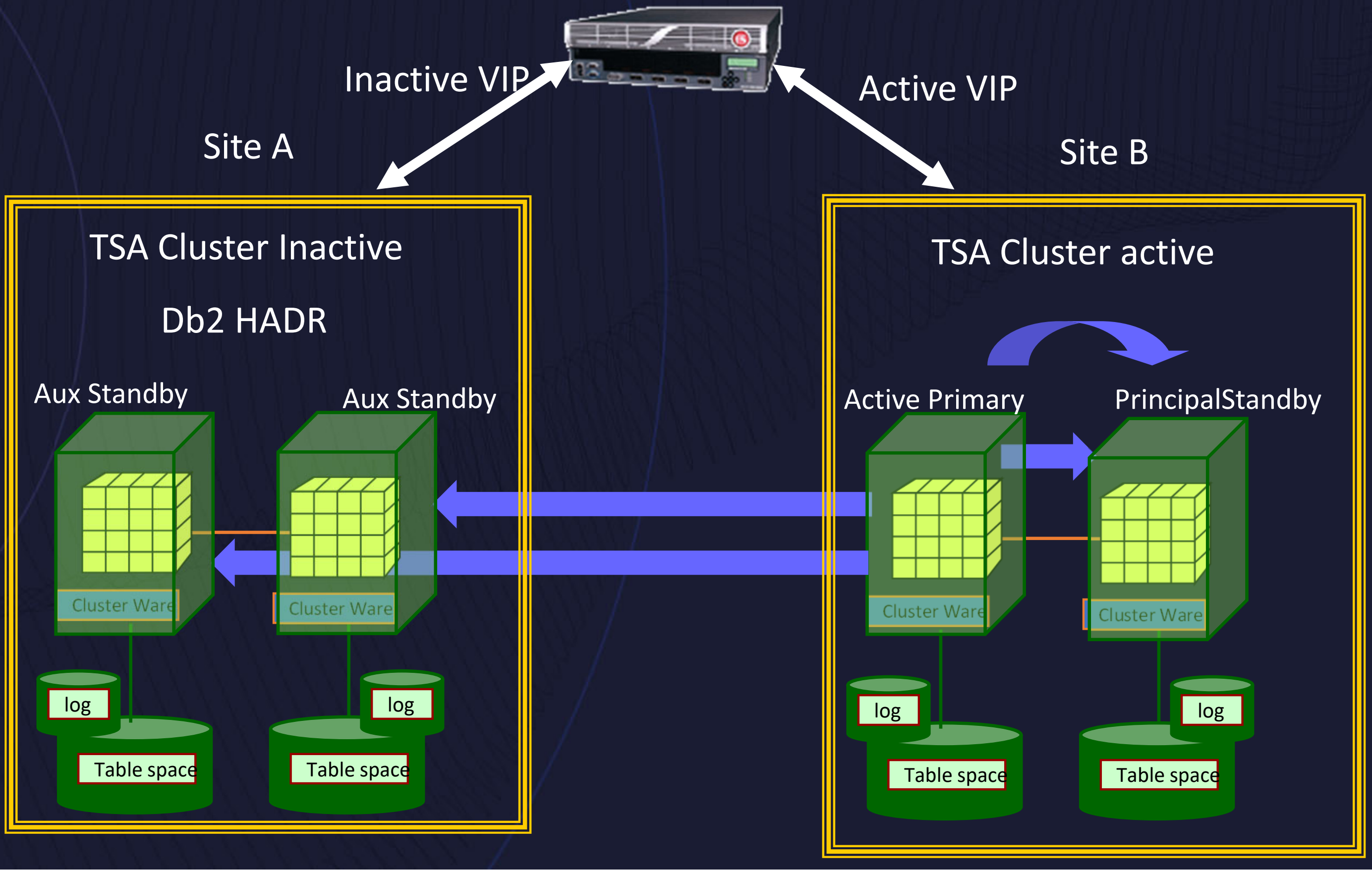
Active Site - Primary and Principal



DR site – Auxiliary #1 and Auxiliary #2 Standbys



Workload Swap – single command issued on Site B



Initial Setup – to configure both automation clusters

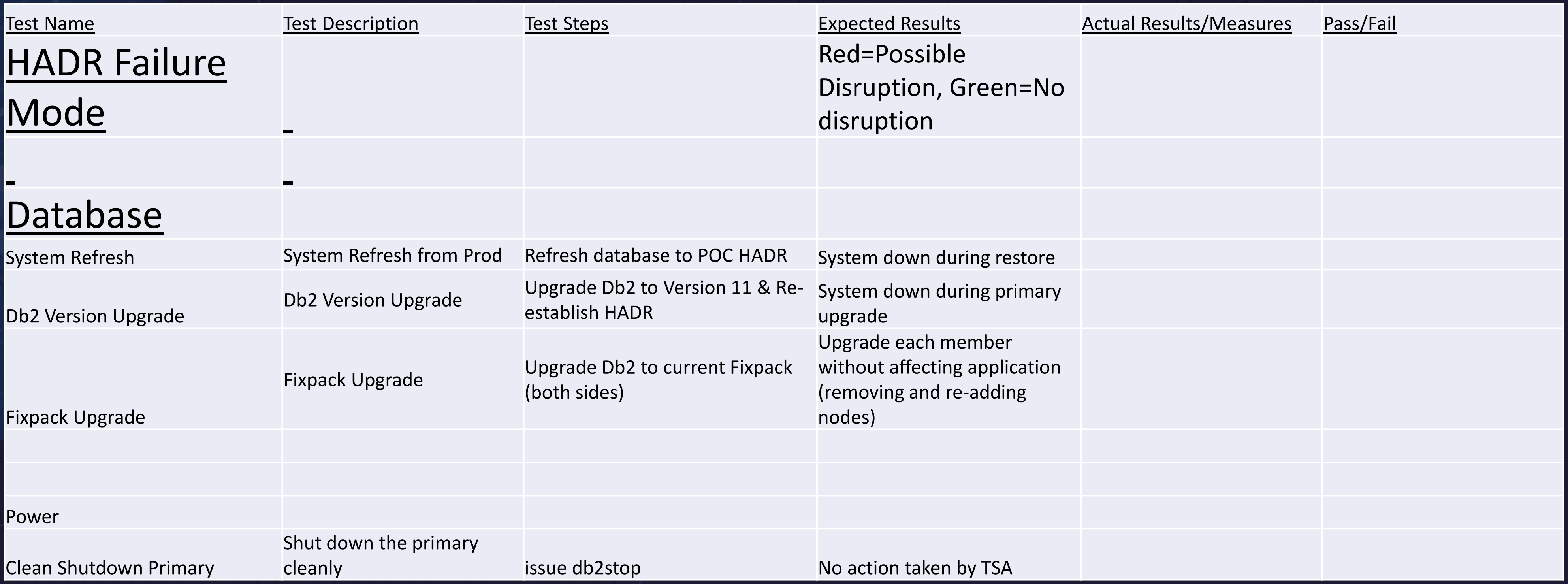
- Setup 4 node HADR cluster without automation
- Run db2haicu on the site with the primary and principal standby (Site A)
- Issue takeover on an auxiliary standby (Site B)
- Disable TSA on site A
- Run db2haicu on the site with the primary and principal standby (Site B)
- Issue takeover on an auxiliary standby (Site A)
- Disable TSA on Site B
- Enable TSA on Site A

Workload Swap – Site A to B

- Issue takeover on an auxiliary standby (Site B)
- Disable TSA on site A
- Enable TSA on Site B

HADR Test Scenarios

I have an excel spreadsheet with most of the scenarios that should be tested.

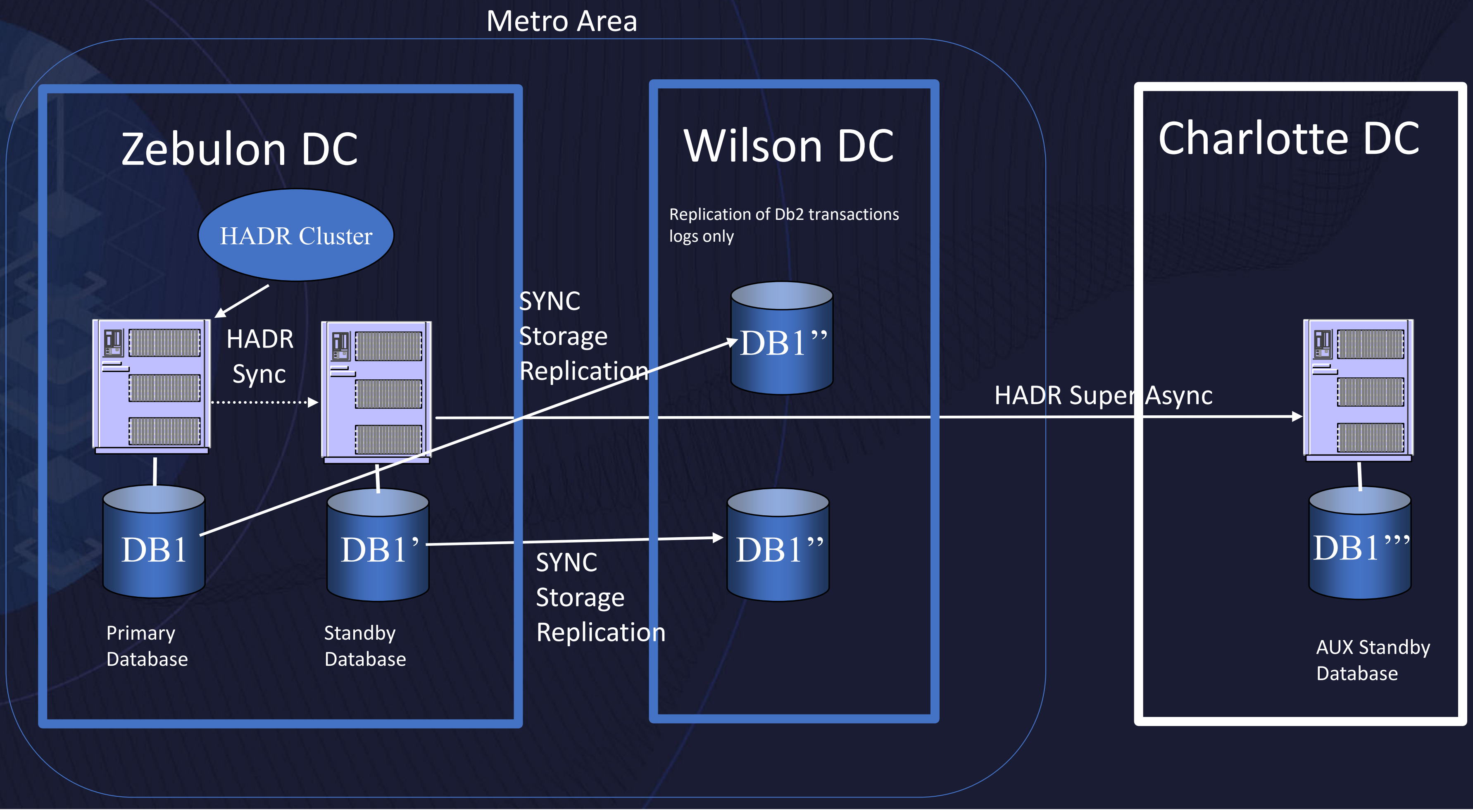


Test Name	Test Description	Test Steps	Expected Results	Actual Results/Measures	Pass/Fail
HADR Failure Mode	_		Red=Possible Disruption, Green=No disruption		
_	_				
Database					
System Refresh	System Refresh from Prod	Refresh database to POC HADR	System down during restore		
Db2 Version Upgrade	Db2 Version Upgrade	Upgrade Db2 to Version 11 & Re-establish HADR	System down during primary upgrade		
Fixpack Upgrade	Fixpack Upgrade	Upgrade Db2 to current Fixpack (both sides)	Upgrade each member without affecting application (removing and re-adding nodes)		
Power					
Clean Shutdown Primary	Shut down the primary cleanly	issue db2stop	No action taken by TSA		

Combining HADR and Storage Replication

- Requirement
  - Local HA and DR
  - In the case of DR want to minimize data loss
  - Have a “bunker site” near the primary data center that can be utilized

Recommended Configuration



Sample Configuration

- Deploy Db2 HADR between floors in ZDC in SYNC mode
  - Provides zero data loss failover with RTO < 1 minute
  - All data maintained in both systems at all time

- Deploy Storage Based replication between ZDC and WDC
  - Provides full insync copy of all data
  - Bunker site to maintain full copy of data and transaction logs

- Deploy Db2 HADR Standby between ZDC and CDC
  - Charlotte will be an Auxiliary Db2 HADR Standby
  - Can be used with zero data loss for scheduled DC move without accessing data in WDC
  - In the case of a true disaster, e.g. ZDC is completely down, the transaction file(s) can be copied from WDC to CDC to eliminate data loss

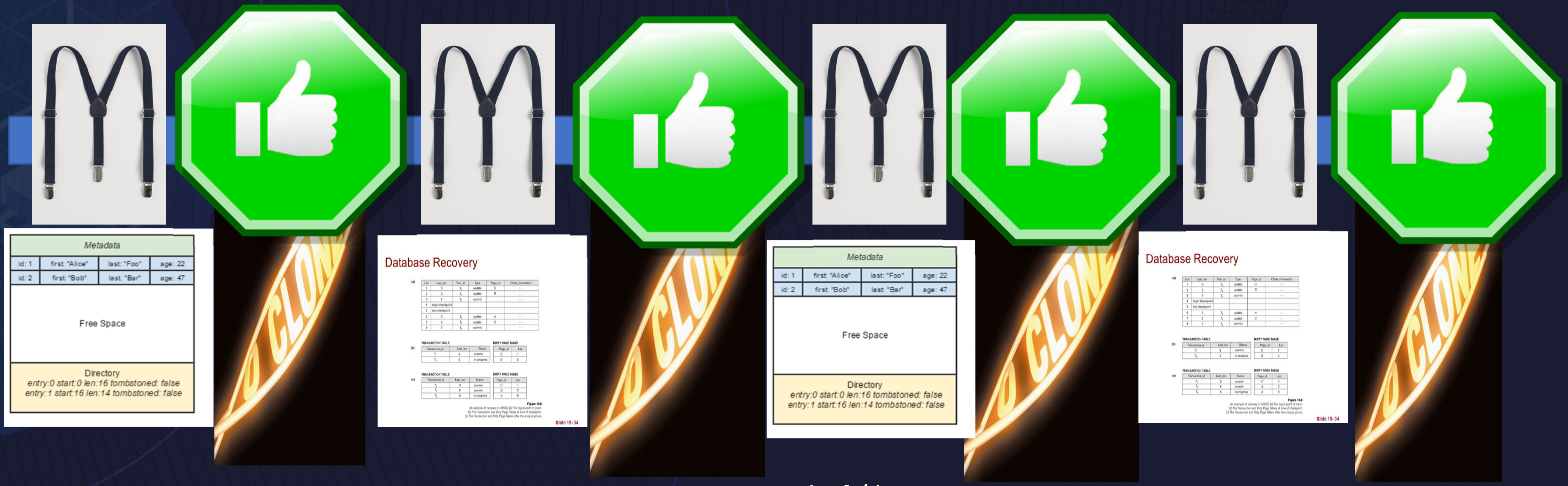
Agenda

- Positioning
- pureScale
- Replication
- HADR
- Backup and Recovery

Backup and Recovery Best Practices

- Adoption of Snapshot / clone backups increasing
  - Be careful with certain vendors claiming that you no longer need to suspend the database when making a clone/snapshot/flashcopy
- Free support for IBM storage arrays
  - Non-IBM array support available through Rocket Software
  - FCM Follow on product dropped GPFS support, thus you must use the ACS Scripted Interface for pureScale
- For tradition backup performance issues
  - Examine the barstats output in the db2diag.log file

Consistent Snapshots (SET WRITE SUSPEND)



Metadata

id: 1	first: "Alice"	last: "Foo"	age: 22
id: 2	first: "Bob"	last: "Bar"	age: 47

Free Space

Directory
entry:0 start:0 len:16 tombstoned: false
entry:1 start:16 len:14 tombstoned: false

Database Recovery

Slide 19-34

Metadata

id: 1	first: "Alice"	last: "Foo"	age: 22
id: 2	first: "Bob"	last: "Bar"	age: 47

Free Space

Directory
entry:0 start:0 len:16 tombstoned: false
entry:1 start:16 len:14 tombstoned: false

Database Recovery

Slide 19-34

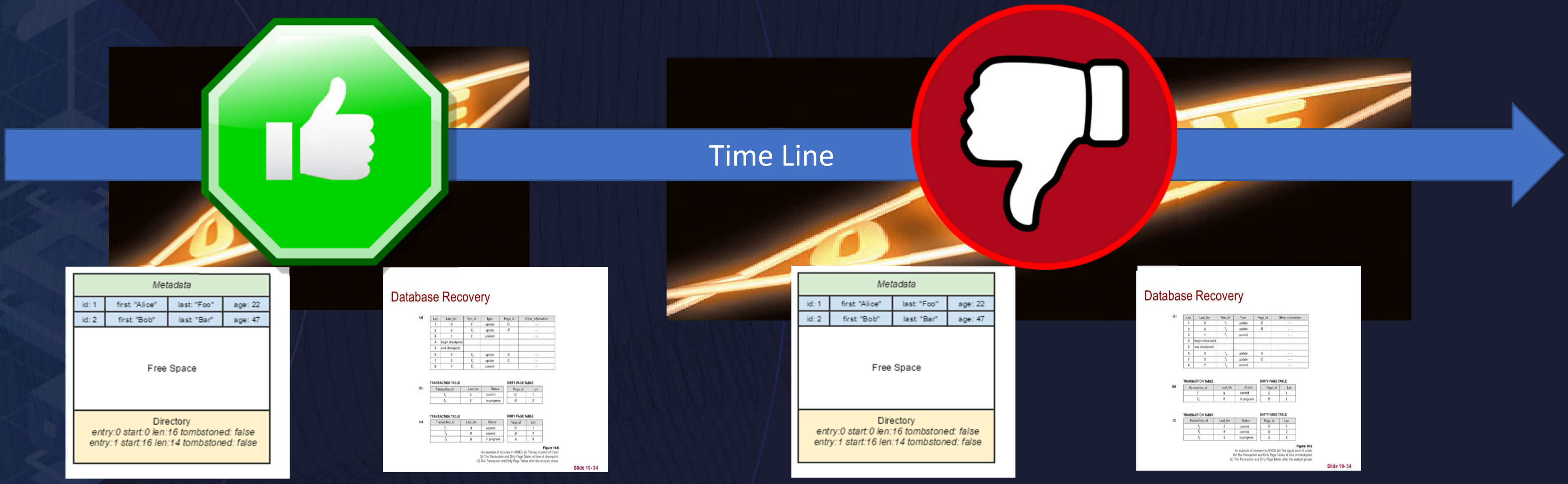
Minbuf / lowtran
Page LSN 100

LFH Recovery LSN 100

Minbuf / lowtran
Page LSN 150

LFH Recovery LSN 100

Dirty Snapshots (No SET WRITE SUSPEND)



Metadata

id: 1	first: "Alice"	last: "Foo"	age: 22
id: 2	first: "Bob"	last: "Bar"	age: 47

Free Space

Directory
entry:0 start:0 len:16 tombstoned: false
entry:1 start:16 len:14 tombstoned: false

Minbuf / lowtran
Page LSN 100

LFH Recovery LSN 100

Metadata

id: 1	first: "Alice"	last: "Foo"	age: 22
id: 2	first: "Bob"	last: "Bar"	age: 47

Free Space

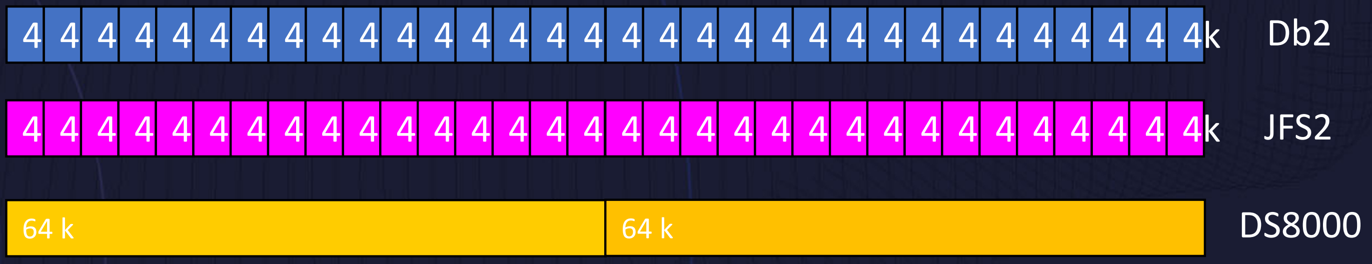
Directory
entry:0 start:0 len:16 tombstoned: false
entry:1 start:16 len:14 tombstoned: false

Minbuf / lowtran
Page LSN 150

LFH Recovery LSN 175

Examples for page alignment issues!

perfect alignment between different layers

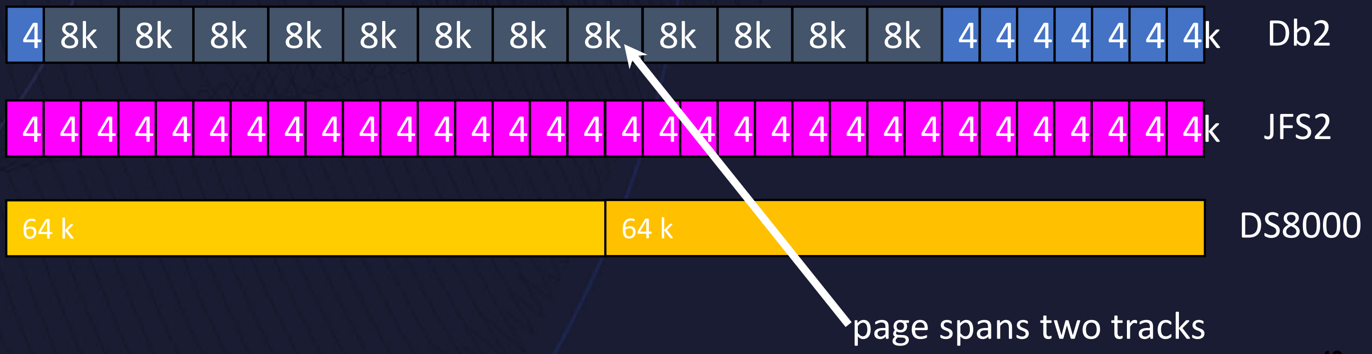


4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4k Db2

4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4k JFS2

64 k 64 k DS8000

missing alignment between different layers
(here: different Db2 page sizes in one filesystem)



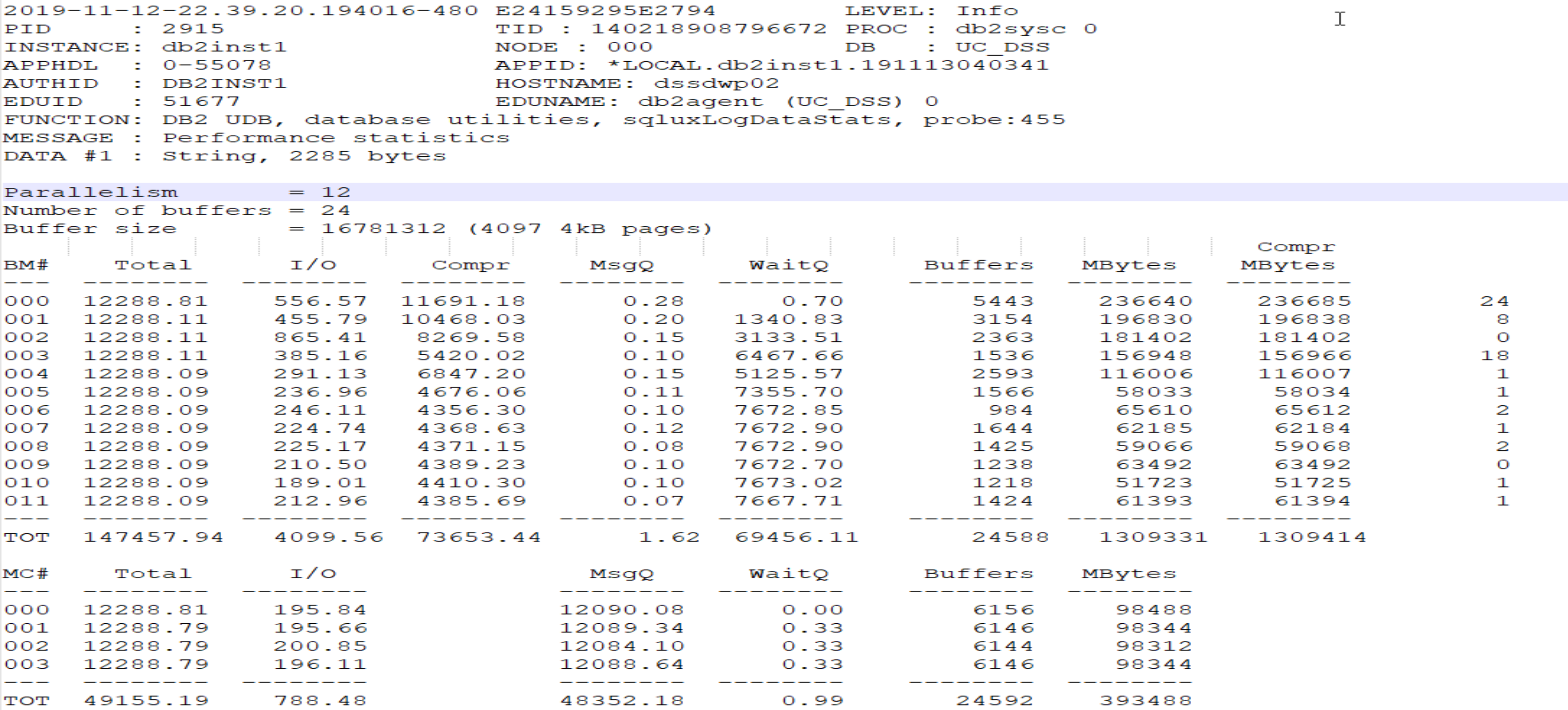
4 8k 8k 8k 8k 8k 8k 8k 8k 8k 8k 8k 8k 4 4 4 4 4 4 4k Db2

4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4k JFS2

64 k 64 k DS8000

page spans two tracks

BARSTATS – how to interpret what backup and restore is doing



```
2019-11-12-22.39.20.194016-480 E24159295E2794       LEVEL: Info
PID     : 2915                 TID : 140218908796672 PROC : db2sysc 0
INSTANCE: db2inst1             NODE : 000            DB   : UC_DSS
APPHDL  : 0-55078              APPID: *LOCAL.db2inst1.191113040341
AUTHID  : DB2INST1             HOSTNAME: dssdwp02
EDUID   : 51677                EDUNAME: db2agent (UC_DSS) 0
FUNCTION: DB2 UDB, database utilities, sqluxLogDataStats, probe:455
MESSAGE : Performance statistics
DATA #1 : String, 2285 bytes

Parallelism       = 12
Number of buffers = 24
Buffer size       = 16781312 (4097 4kB pages)
                                                                               Compr
BM#    Total      I/O      Compr     MsgQ      WaitQ      Buffers   MBytes    MBytes
---  --------  --------  --------  --------  --------    --------  --------  --------
000  12288.81    556.57  11691.18      0.28      0.70        5443    236640    236685        24
001  12288.11    455.79  10468.03      0.20   1340.83        3154    196830    196838         8
002  12288.11    865.41   8269.58      0.15   3133.51        2363    181402    181402         0
003  12288.11    385.16   5420.02      0.10   6467.66        1536    156948    156966        18
004  12288.09    291.13   6847.20      0.15   5125.57        2593    116006    116007         1
005  12288.09    236.96   4676.06      0.11   7355.70        1566     58033     58034         1
006  12288.09    246.11   4356.30      0.10   7672.85         984     65610     65612         2
007  12288.09    224.74   4368.63      0.12   7672.90        1644     62185     62184         1
008  12288.09    225.17   4371.15      0.08   7672.90        1425     59066     59068         2
009  12288.09    210.50   4389.23      0.10   7672.70        1238     63492     63492         0
010  12288.09    189.01   4410.30      0.10   7673.02        1218     51723     51725         1
011  12288.09    212.96   4385.69      0.07   7667.71        1424     61393     61394         1
---  --------  --------  --------  --------  --------    --------  --------  --------
TOT  147457.94   4099.56  73653.44      1.62  69456.11       24588   1309331   1309414

MC#    Total      I/O                MsgQ      WaitQ      Buffers   MBytes
---  --------  --------            --------  --------    --------  --------
000  12288.81    195.84            12090.08      0.00        6156     98488
001  12288.79    195.66            12089.34      0.33        6146     98344
002  12288.79    200.85            12084.10      0.33        6144     98312
003  12288.79    196.11            12088.64      0.33        6146     98344
---  --------  --------            --------  --------    --------  --------
TOT  49155.19    788.48            48352.18      0.99       24592    393488
```

Explanation

BM# - the number we assigned to an individual Buffer Manipulator. BM's READ data from the databases tablespace during a backup and place them into buffers.

MC# - the number assigned to an individual Media Controller. MC's WRITE buffers out to the target location.

Total - The total amount of time spent by the process in seconds.

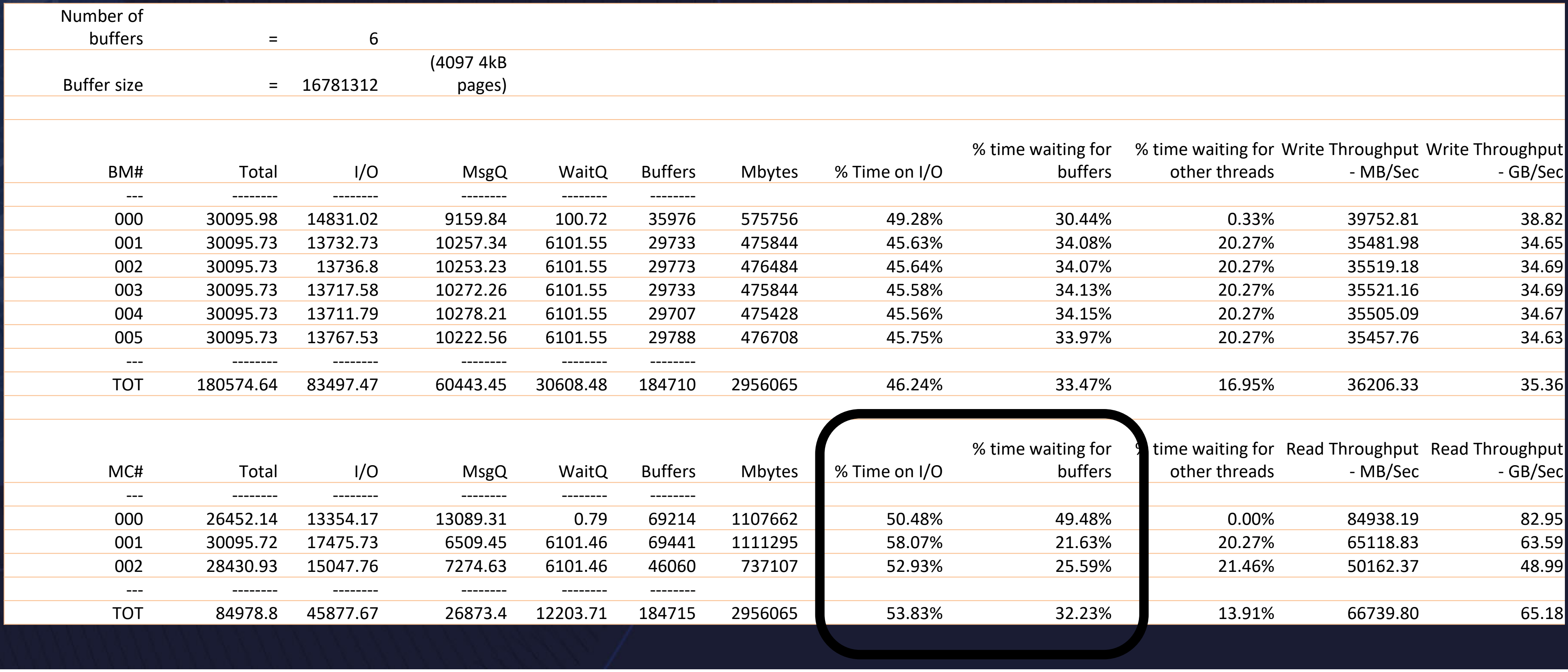
I/O - The amount of time spent either reading or writing data. For the BM's this represents time reading data from tablespace, and filling the buffer. For MC it's time spent reading from buffer and sending it to the target destination.
MsgQ - This is the amount of time we spend waiting to get a buffer. For BM's it's how long is spent waiting to get an empty buffer for filling. For MC's it's time spent waiting to get a full buffer in order to write out.
Wait Q - Amount of time spent waiting on directives from the agent overseeing the whole backup.

Buffers - The number of Buffers Processed by a particular BM or MC. A BM filled X number of buffers. An MC wrote out X number of buffers.
MBytes - The amount of data handled by a particular BM or MC in Mbytes.

Compr Mbytes - the amount of data that was processed by the compression/encryption library. This value will always be >= the I/O column because we will sometimes process the same data more than once in order to make it fit in the buffer.

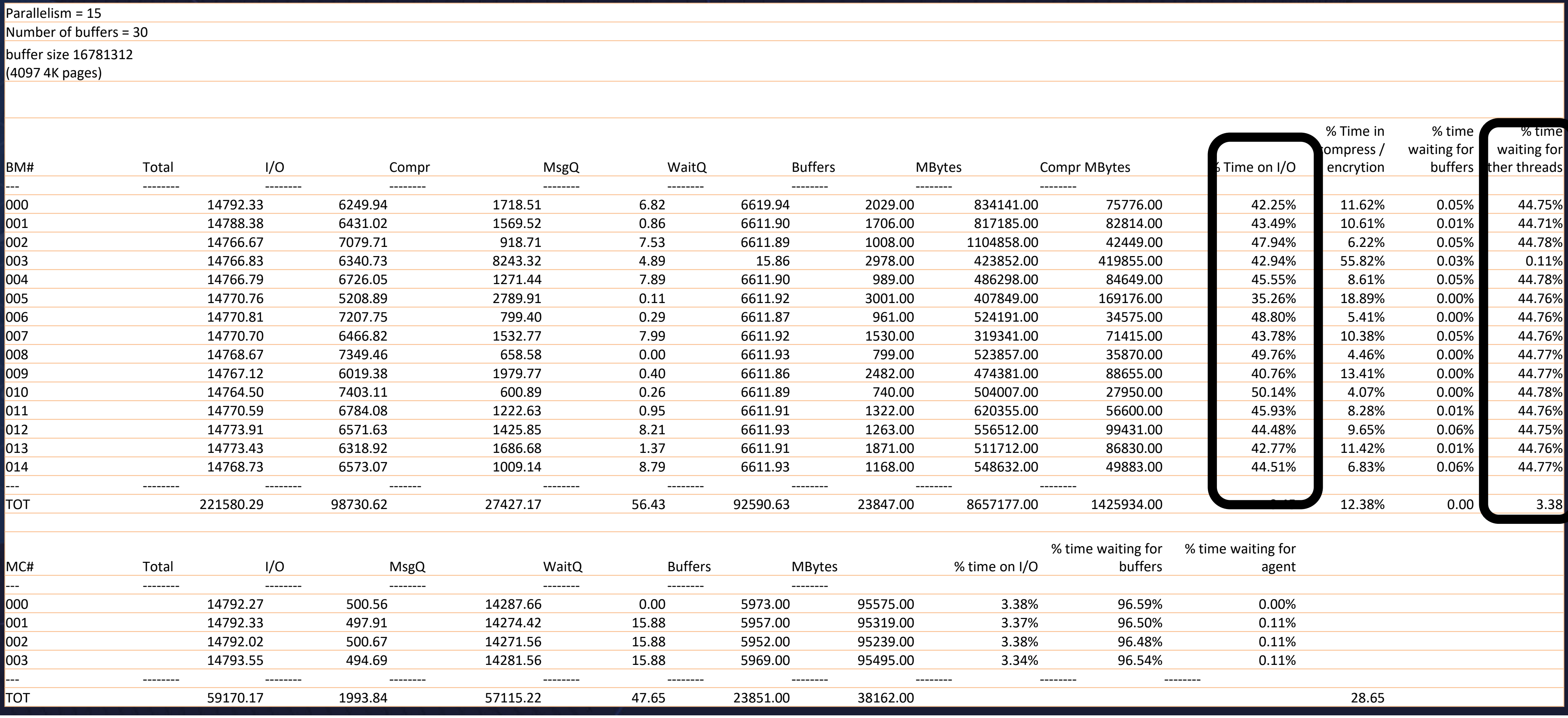
The mystery column will be fully revealed if you run with Db2_BAR_STATS=on. It represents wasted effort on the part of the compr/encr library, where it processed some data and then had to throw away the work because the data didn't fit in the target buffer

How can this backup be improved?



Number of buffers	=	6								
Buffer size	=	16781312	(4097 4kB pages)							
BM#	Total	I/O	MsgQ	WaitQ	Buffers	Mbytes	% Time on I/O	% time waiting for buffers	% time waiting for other threads	Write Throughput - MB/Sec	Write Throughput - GB/Sec
---	--------	--------	--------	--------	--------						
000	30095.98	14831.02	9159.84	100.72	35976	575756	49.28%	30.44%	0.33%	39752.81	38.82
001	30095.73	13732.73	10257.34	6101.55	29733	475844	45.63%	34.08%	20.27%	35481.98	34.65
002	30095.73	13736.8	10253.23	6101.55	29773	476484	45.64%	34.07%	20.27%	35519.18	34.69
003	30095.73	13717.58	10272.26	6101.55	29733	475844	45.58%	34.13%	20.27%	35521.16	34.69
004	30095.73	13711.79	10278.21	6101.55	29707	475428	45.56%	34.15%	20.27%	35505.09	34.67
005	30095.73	13767.53	10222.56	6101.55	29788	476708	45.75%	33.97%	20.27%	35457.76	34.63
---	--------	--------	--------	--------	--------						
TOT	180574.64	83497.47	60443.45	30608.48	184710	2956065	46.24%	33.47%	16.95%	36206.33	35.36
MC#	Total	I/O	MsgQ	WaitQ	Buffers	Mbytes	% Time on I/O	% time waiting for buffers	% time waiting for other threads	Read Throughput - MB/Sec	Read Throughput - GB/Sec
---	--------	--------	--------	--------	--------						
000	26452.14	13354.17	13089.31	0.79	69214	1107662	50.48%	49.48%	0.00%	84938.19	82.95
001	30095.72	17475.73	6509.45	6101.46	69441	1111295	58.07%	21.63%	20.27%	65118.83	63.59
002	28430.93	15047.76	7274.63	6101.46	46060	737107	52.93%	25.59%	21.46%	50162.37	48.99
---	--------	--------	--------	--------	--------						
TOT	84978.8	45877.67	26873.4	12203.71	184715	2956065	53.83%	32.23%	13.91%	66739.80	65.18

How can this backup be improved?



Parallelism = 15

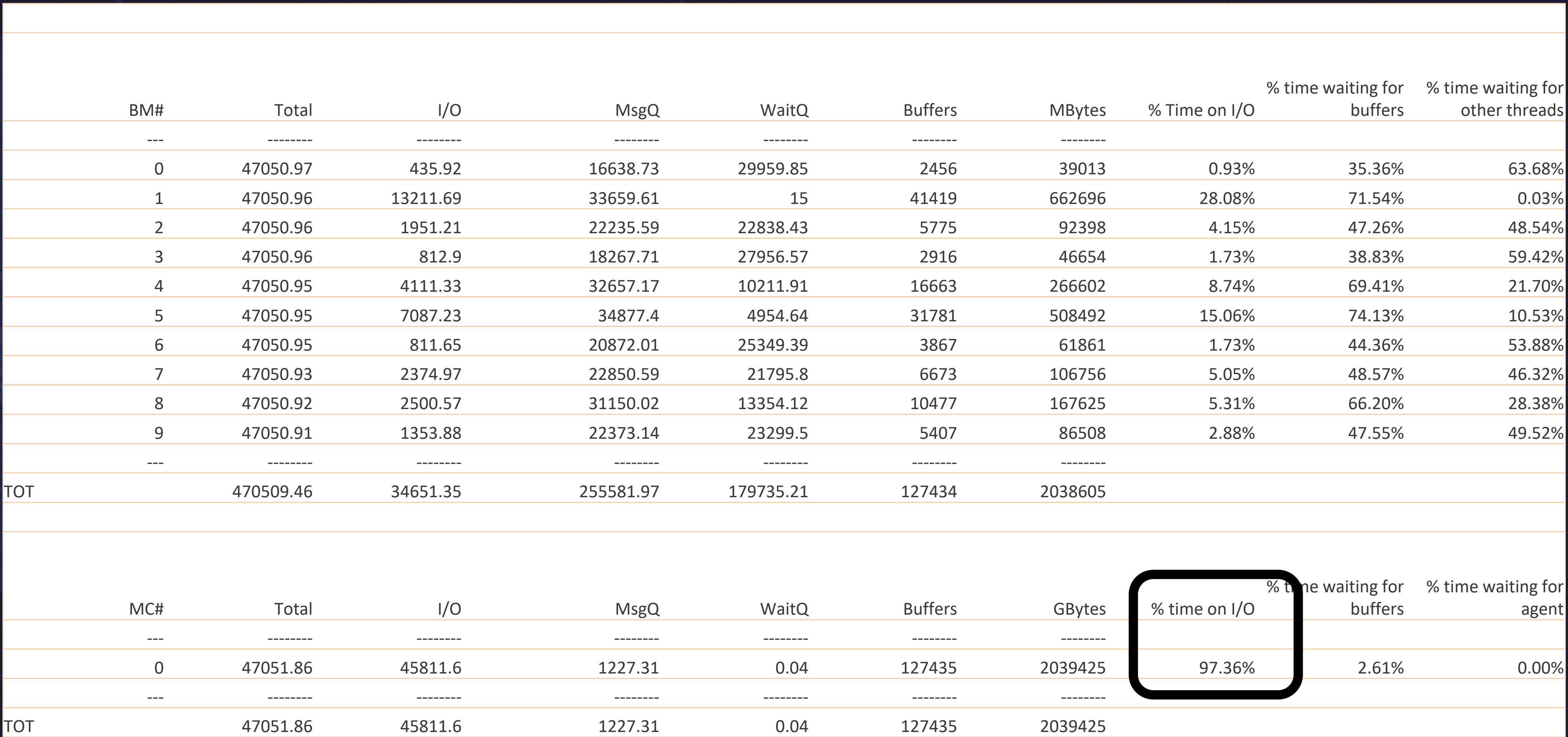
Number of buffers = 30

buffer size 16781312
(4097 4K pages)

BM#	Total	I/O	Compr	MsgQ	WaitQ	Buffers	MBytes	Compr MBytes	Time on I/O	% Time in ompress / encryption	% time waiting for buffers	% time waiting for ther threads
---	--------	--------	--------	--------	--------	--------	--------	--------				
000	14792.33	6249.94	1718.51	6.82	6619.94	2029.00	834141.00	75776.00	42.25%	11.62%	0.05%	44.75%
001	14788.38	6431.02	1569.52	0.86	6611.90	1706.00	817185.00	82814.00	43.49%	10.61%	0.01%	44.71%
002	14766.67	7079.71	918.71	7.53	6611.89	1008.00	1104858.00	42449.00	47.94%	6.22%	0.05%	44.78%
003	14766.83	6340.73	8243.32	4.89	15.86	2978.00	423852.00	419855.00	42.94%	55.82%	0.03%	0.11%
004	14766.79	6726.05	1271.44	7.89	6611.90	989.00	486298.00	84649.00	45.55%	8.61%	0.05%	44.78%
005	14770.76	5208.89	2789.91	0.11	6611.92	3001.00	407849.00	169176.00	35.26%	18.89%	0.00%	44.76%
006	14770.81	7207.75	799.40	0.29	6611.87	961.00	524191.00	34575.00	48.80%	5.41%	0.00%	44.76%
007	14770.70	6466.82	1532.77	7.99	6611.92	1530.00	319341.00	71415.00	43.78%	10.38%	0.05%	44.76%
008	14768.67	7349.46	658.58	0.00	6611.93	799.00	523857.00	35870.00	49.76%	4.46%	0.00%	44.77%
009	14767.12	6019.38	1979.77	0.40	6611.86	2482.00	474381.00	88655.00	40.76%	13.41%	0.00%	44.77%
010	14764.50	7403.11	600.89	0.26	6611.89	740.00	504007.00	27950.00	50.14%	4.07%	0.00%	44.78%
011	14770.59	6784.08	1222.63	0.95	6611.91	1322.00	620355.00	56600.00	45.93%	8.28%	0.01%	44.76%
012	14773.91	6571.63	1425.85	8.21	6611.93	1263.00	556512.00	99431.00	44.48%	9.65%	0.06%	44.75%
013	14773.43	6318.92	1686.68	1.37	6611.91	1871.00	511712.00	86830.00	42.77%	11.42%	0.01%	44.76%
014	14768.73	6573.07	1009.14	8.79	6611.93	1168.00	548632.00	49883.00	44.51%	6.83%	0.06%	44.77%
---	--------	--------	-------	--------	--------	--------	--------	--------				
TOT	221580.29	98730.62	27427.17	56.43	92590.63	23847.00	8657177.00	1425934.00	[illegible]	12.38%	0.00	3.38

MC#	Total	I/O	MsgQ	WaitQ	Buffers	MBytes	% time on I/O	% time waiting for buffers	% time waiting for agent	
---	--------	--------	--------	--------	--------	--------				
000	14792.27	500.56	14287.66	0.00	5973.00	95575.00	3.38%	96.59%	0.00%	
001	14792.33	497.91	14274.42	15.88	5957.00	95319.00	3.37%	96.50%	0.11%	
002	14792.02	500.67	14271.56	15.88	5952.00	95239.00	3.38%	96.48%	0.11%	
003	14793.55	494.69	14281.56	15.88	5969.00	95495.00	3.34%	96.54%	0.11%	
---	--------	--------	--------	--------	--------	--------	--------	--------	--------	
TOT	59170.17	1993.84	57115.22	47.65	23851.00	38162.00				28.65

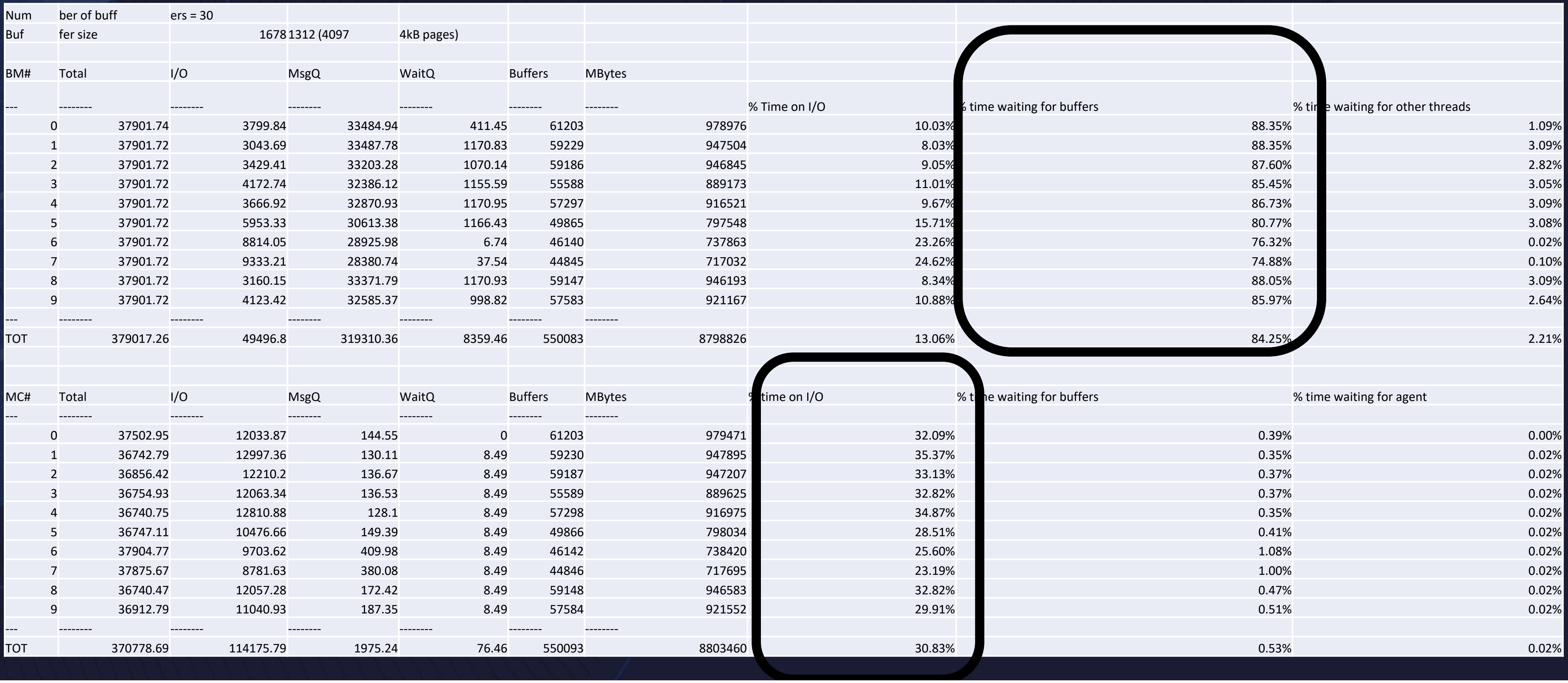
How can this backup be improved?



BM#	Total	I/O	MsgQ	WaitQ	Buffers	MBytes	% Time on I/O	% time waiting for buffers	% time waiting for other threads
---	--------	--------	--------	--------	--------	--------			
0	47050.97	435.92	16638.73	29959.85	2456	39013	0.93%	35.36%	63.68%
1	47050.96	13211.69	33659.61	15	41419	662696	28.08%	71.54%	0.03%
2	47050.96	1951.21	22235.59	22838.43	5775	92398	4.15%	47.26%	48.54%
3	47050.96	812.9	18267.71	27956.57	2916	46654	1.73%	38.83%	59.42%
4	47050.95	4111.33	32657.17	10211.91	16663	266602	8.74%	69.41%	21.70%
5	47050.95	7087.23	34877.4	4954.64	31781	508492	15.06%	74.13%	10.53%
6	47050.95	811.65	20872.01	25349.39	3867	61861	1.73%	44.36%	53.88%
7	47050.93	2374.97	22850.59	21795.8	6673	106756	5.05%	48.57%	46.32%
8	47050.92	2500.57	31150.02	13354.12	10477	167625	5.31%	66.20%	28.38%
9	47050.91	1353.88	22373.14	23299.5	5407	86508	2.88%	47.55%	49.52%
---	--------	--------	--------	--------	--------	--------			
TOT	470509.46	34651.35	255581.97	179735.21	127434	2038605			

MC#	Total	I/O	MsgQ	WaitQ	Buffers	GBytes	% time on I/O	% time waiting for buffers	% time waiting for agent
---	--------	--------	--------	--------	--------	--------			
0	47051.86	45811.6	1227.31	0.04	127435	2039425	97.36%	2.61%	0.00%
---	--------	--------	--------	--------	--------	--------			
TOT	47051.86	45811.6	1227.31	0.04	127435	2039425			

How can this backup be improved?



Num	ber of buff	ers = 30							
Buf	fer size	1678	1312 (4097	4kB pages)					
BM#	Total	I/O	MsgQ	WaitQ	Buffers	MBytes			
---	--------	--------	--------	--------	--------	--------	% Time on I/O	% time waiting for buffers	% time waiting for other threads
0	37901.74	3799.84	33484.94	411.45	61203	978976	10.03%	88.35%	1.09%
1	37901.72	3043.69	33487.78	1170.83	59229	947504	8.03%	88.35%	3.09%
2	37901.72	3429.41	33203.28	1070.14	59186	946845	9.05%	87.60%	2.82%
3	37901.72	4172.74	32386.12	1155.59	55588	889173	11.01%	85.45%	3.05%
4	37901.72	3666.92	32870.93	1170.95	57297	916521	9.67%	86.73%	3.09%
5	37901.72	5953.33	30613.38	1166.43	49865	797548	15.71%	80.77%	3.08%
6	37901.72	8814.05	28925.98	6.74	46140	737863	23.26%	76.32%	0.02%
7	37901.72	9333.21	28380.74	37.54	44845	717032	24.62%	74.88%	0.10%
8	37901.72	3160.15	33371.79	1170.93	59147	946193	8.34%	88.05%	3.09%
9	37901.72	4123.42	32585.37	998.82	57583	921167	10.88%	85.97%	2.64%
---	--------	--------	--------	--------	--------	--------			
TOT	379017.26	49496.8	319310.36	8359.46	550083	8798826	13.06%	84.25%	2.21%
MC#	Total	I/O	MsgQ	WaitQ	Buffers	MBytes	% time on I/O	% time waiting for buffers	% time waiting for agent
---	--------	--------	--------	--------	--------	--------			
0	37502.95	12033.87	144.55	0	61203	979471	32.09%	0.39%	0.00%
1	36742.79	12997.36	130.11	8.49	59230	947895	35.37%	0.35%	0.02%
2	36856.42	12210.2	136.67	8.49	59187	947207	33.13%	0.37%	0.02%
3	36754.93	12063.34	136.53	8.49	55589	889625	32.82%	0.37%	0.02%
4	36740.75	12810.88	128.1	8.49	57298	916975	34.87%	0.35%	0.02%
5	36747.11	10476.66	149.39	8.49	49866	798034	28.51%	0.41%	0.02%
6	37904.77	9703.62	409.98	8.49	46142	738420	25.60%	1.08%	0.02%
7	37875.67	8781.63	380.08	8.49	44846	717695	23.19%	1.00%	0.02%
8	36740.47	12057.28	172.42	8.49	59148	946583	32.82%	0.47%	0.02%
9	36912.79	11040.93	187.35	8.49	57584	921552	29.91%	0.51%	0.02%
---	--------	--------	--------	--------	--------	--------			
TOT	370778.69	114175.79	1975.24	76.46	550093	8803460	30.83%	0.53%	0.02%

Thank You

Speaker: Dale McInnis

Company: IBM Canada Ltd.

Email Address: dmcinnis@ca.ibm.com

Please fill out your session evaluation before leaving!

Thank
You