



Customer Experiences Implementing Databases on IBM Z/LinuxONE

David Simpson, IBM WSC
simpson.dave@us.ibm.com



Large Bank Fraud Detection Application

Wait Classes by Total Wait Time

Wait Class	Waits	Total Wait Time (sec)	Avg Wait Time	% DB time	Avg Active Sessions
DB CPU		10,679		72.2	87.7
Cluster	3,735,935	2,207	590.81us	14.9	18.1
Concurrency	479,734	452	941.25us	3.1	3.7
System I/O	3,855,601	258	66.94us	1.7	2.1
Commit	244,801	224	913.77us	1.5	1.8
Other	350,381	70	199.84us	.5	0.6
Configuration	4,464	5	1.05ms	.0	0.0
User I/O	1,113,397	4	3.69us	.0	0.0
Network	283,545	0	1.08us	.0	0.0
Application	13	0	1.31ms	.0	0.0

Host CPU

CPUs	Cores	Sockets	Load Average Begin	Load Average End	%User	%System	%WIO	%Idle
100	7	1	113.89	107.12	87.3	3.1	0.0	8.3

- Hipersockets vs OSA vs new NETH (Network Express) ports – a mixed configuration worked best for this workload
- Linux Kernel tuning parameters
- Cross-drawer performance (49, 53, 53, 53), crossing CPU
- Oracle In memory – slower for this workload

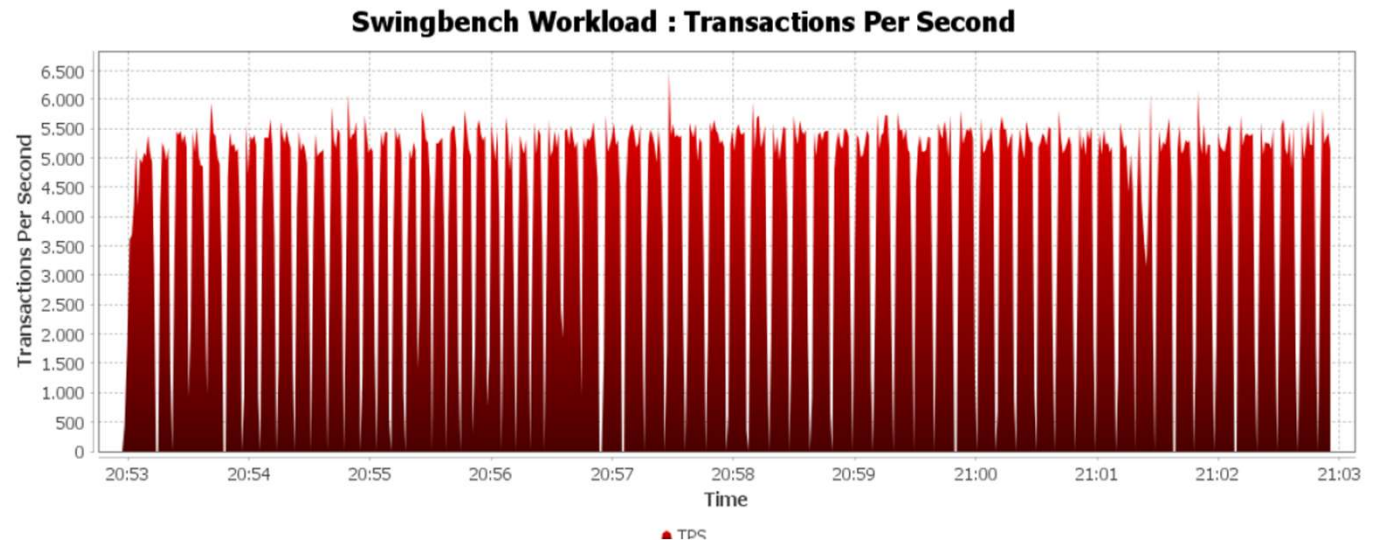


Benchmarking is NOT Easy!

Customers come to us with on-prem tests , sometimes drawing wrong conclusions!

Parameter	Value
Total completed transactions	2.479.564
Average transactions per second	4132,61
Maximum transaction rate	264.596
Total Failed Transactions	0

Overview Charts





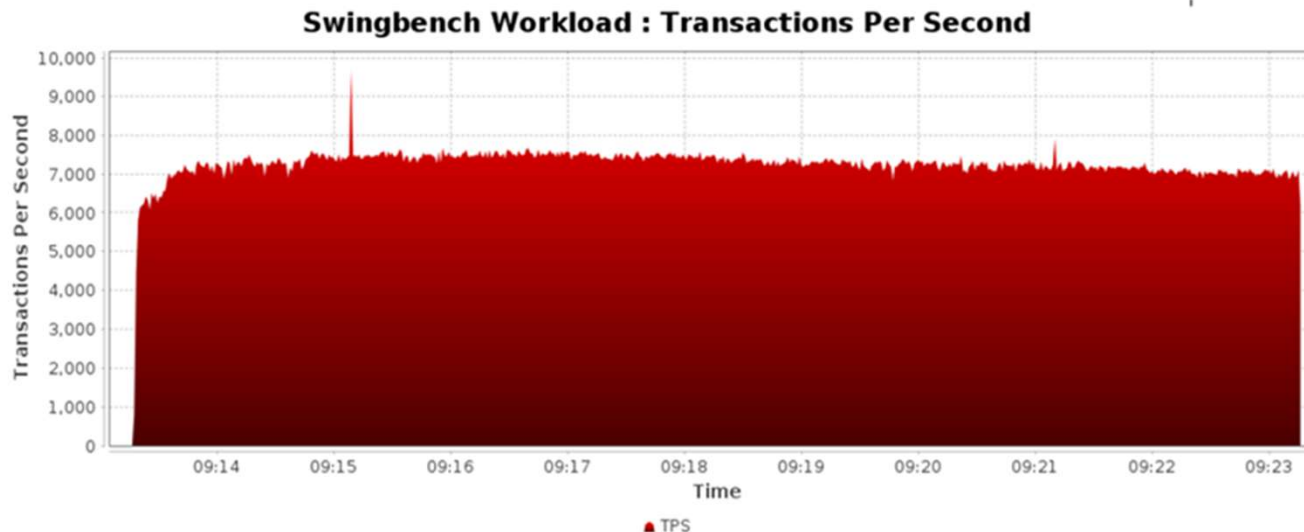
Customer Result – After Tuning

4132.6 TPS -> 7245.6

Parameter	Value
Total completed transactions	4.347.336
Average transactions per second	7245.56
Maximum transaction rate	451,217
Total Failed Transactions	0

- Oracle REDO Logs – larger & more
- Oracle Automated Storage Management (ASM)
(File system database files have cache effect)
- Benchmark test harnesses have a level of randomization to them

Overview Charts





Large European Bank - Sizing

Wait Classes by Total Wait Time

Wait Class	Waits	Total Wait Time (sec)	Avg Wait Time	% DB time	Avg Active Sessions
Application	12,590	26,908	2137.23ms	65.0	7.5
User I/O	67,549,319	8,407	124.45us	20.3	2.3
DB CPU		7,479		18.1	2.1
Other	1,841,351	694	376.75us	1.7	0.2
Cluster	6,726,823	570	84.72us	1.4	0.2
Concurrency	17,672,424	462	26.16us	1.1	0.1
System I/O	912,375	291	318.60us	.7	0.1
Network	11,493,636	137	11.91us	.3	0.0
Commit	286,749	102	356.17us	.2	0.0
Configuration	519	4	7.07ms	.0	0.0

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Avg Wait	% DB time	Wait Class
enq: TX - row lock contention	1,349	26.9K	19.94 s	65.0	Application
DB CPU		7479.4		18.1	
read by other session	39,516,794	4613	116.73us	11.1	User I/O
db file sequential read	7,370,601	931.9	126.44us	2.3	User I/O
latch: cache buffers chains	17,428,002	450.4	25.84us	1.1	Concurrency
direct path read	1,995,705	292.9	146.77us	.7	User I/O
db file parallel read	837,394	130.7	156.08us	.3	User I/O
reliable message	7,408	129.6	17.49ms	.3	Other
gc buffer busy acquire	1,086,436	127.4	117.27us	.3	Cluster
log file sync	286,730	102.1	356.13us	.2	Commit

SLA – for customer 0.3 seconds



Large European Bank - Sizing

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Avg Wait	% DB time	Wait Class
DB CPU		37.2K		59.1	
db file sequential read	22,662,108	4734.2	208.90us	7.5	User I/O
log file sync	5,707,779	4318.7	756.63us	6.9	Commit
gc current block 2-way	14,242,978	1950.2	136.92us	3.1	Cluster
direct path read	6,497,180	1517.3	233.54us	2.4	User I/O
gc current grant read-mostly invalidation	926,230	1170	1.26ms	1.9	Cluster
latch: shared pool	38,047	926.5	24.35ms	1.5	Concurrency
gc buffer busy acquire	651,601	886.5	1.36ms	1.4	Cluster
gc cr block 2-way	6,339,908	876.5	138.25us	1.4	Cluster
gc current block busy	595,276	762.1	1.28ms	1.2	Cluster

Wait Classes by Total Wait Time

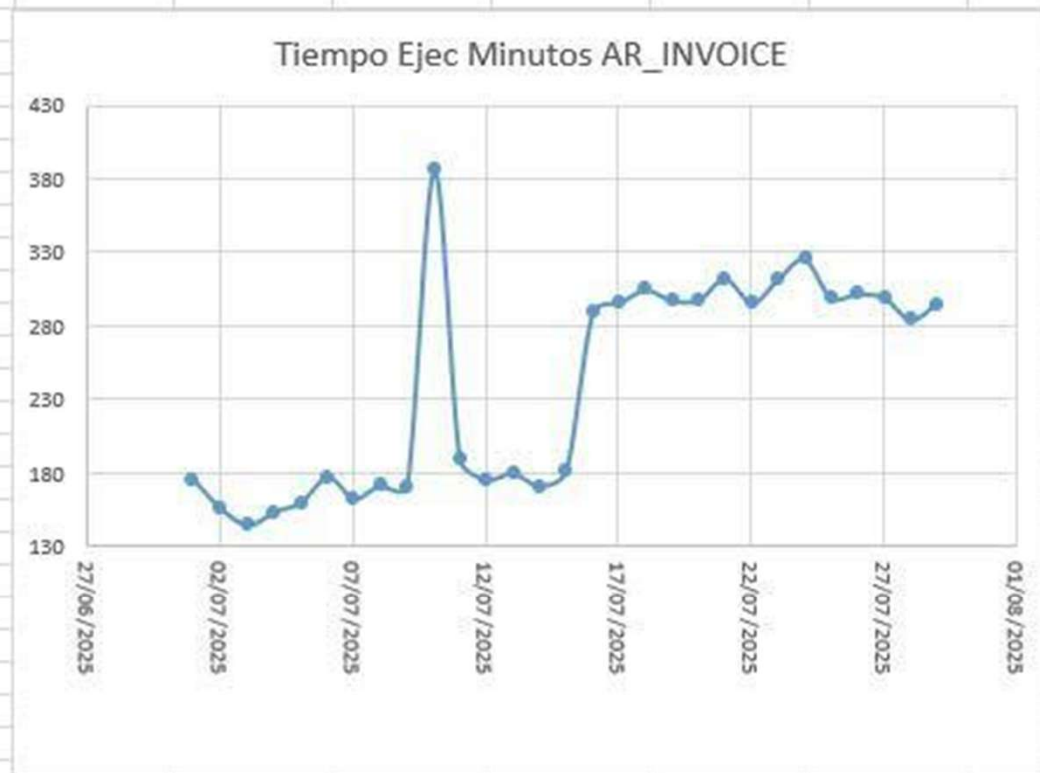
Wait Class	Waits	Total Wait Time (sec)	Avg Wait Time	% DB time	Avg Active Sessions
DB CPU		37,231		59.1	15.6
Cluster	34,906,887	10,029	287.32us	15.9	4.2
User I/O	52,761,637	6,690	126.79us	10.6	2.8
Commit	5,707,779	4,319	756.63us	6.9	1.8
Concurrency	1,418,129	2,239	1.58ms	3.6	0.9
Other	2,033,016	1,462	719.13us	2.3	0.6
Network	185,999,897	1,277	6.87us	2.0	0.5
Application	155,222	108	694.06us	.2	0.0
Configuration	1,773	7	4.14ms	.0	0.0

Application tuning and extensive NETH network tests with firmware tests – everything ran great (at least Oracle wise)



Large Latin American Telecom

Fecha Inicio	Tiempo Eiec
01/07/2025	175.4333333
02/07/2025	155.1833333
03/07/2025	144.05
04/07/2025	153.1666667
05/07/2025	159.6666667
06/07/2025	176.85
07/07/2025	162.1166667
08/07/2025	171.4
09/07/2025	170.7
10/07/2025	386.15
11/07/2025	189.95
12/07/2025	175.2166667
13/07/2025	179.4166667
14/07/2025	170.7833333
15/07/2025	181.3166667
16/07/2025	288.8166667
17/07/2025	295.4166667
18/07/2025	304.55
19/07/2025	297.3833333
20/07/2025	297.65
21/07/2025	312.2166667
22/07/2025	295.8
23/07/2025	311.4333333
24/07/2025	325.5
25/07/2025	299.1333333
26/07/2025	301.7666667
27/07/2025	299.0666667
28/07/2025	284.0666667
29/07/2025	294.95



IBM Customer Care was Engaged – Jul 16th 00:30 problem started



Jul 14th – Everything OK – Network waits ~ 1,722 seconds during 6 hour Window

Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
ithbmbd102.telcel.com	IBM zSeries Based Linux	36			566.4

Snap Id	Snap Time	Sessions	Cursors/Session	Instances	Pluggable Databases Open
Begin Snap: 7667	14-Jul-25 00:38:21	143	1.3	2	
End Snap: 7679	14-Jul-25 06:38:07	169	1.3	2	
Elapsed:	359.78 (mins)				
DB Time:	1,608.29 (mins)				

Report Summary

Top ADDM Findings by Average Active Sessions

Finding Name	Avg active sessions of the task	Percent active sessions of finding	Task Name	Begin Snap Time	End Snap Time
"User I/O" wait Class	7.28	37.53	ADDM:1204327370_2_7675	14-Jul-25 04:08	14-Jul-25 04:37
"User I/O" wait Class	6.61	34.20	ADDM:1204327370_2_7672	14-Jul-25 02:37	14-Jul-25 03:08
"User I/O" wait Class	6.33	33.33	ADDM:1204327370_2_7679	14-Jul-25 06:07	14-Jul-25 06:38
Commits and Rollbacks	7.28	12.00	ADDM:1204327370_2_7675	14-Jul-25 04:08	14-Jul-25 04:37
Commits and Rollbacks	6.61	10.82	ADDM:1204327370_2_7672	14-Jul-25 02:37	14-Jul-25 03:08

Top Process Types by Wait Class

- This table displays top process types ordered by wait class

Process Type	Description	Wait Class Type	Wait Time (sec)
	Others (foregrounds + unclassified backgrounds)	User I/O	22,776.24
	Others (foregrounds + unclassified backgrounds)	Commit	9,471.21
LG	Log Writer Slave	System I/O	8,494.99
P	Parallel query slave	User I/O	7,348.43
	Others (foregrounds + unclassified backgrounds)	Cluster	6,778.14
	Others (foregrounds + unclassified backgrounds)	Network	1,722.75
P	Parallel query slave	Cluster	1,077.26
LM	global cache service process	Other	958.15
	Others (foregrounds + unclassified backgrounds)	Other	812.29
LGWR	Redo etc.	Other	610.36
O	ASM Connection Pool	Other	331.22
	Others (foregrounds + unclassified backgrounds)	Concurrency	331.08
P	Parallel query slave	Other	326.35



Jul 15th (Later) – Database Re-org was done

```
SQL> SELECT (SELECT name FROM v$containers) db,TO_CHAR(sysdate,'DD-MM-YYYY HH24:MI:SS') day FROM dual;
```

DB	DAY
-----	-----
MFBETL	15-07-2025 17:27:43

```
Elapsed: 00:00:00.044
```

```
SQL> show user
```

```
USER is "SYS"
```

```
SQL> prompt DROP TABLES BACK - BIBES
```

```
DROP TABLES BACK - BIBES
```

```
SQL> --Bloque 1
```

```
SQL> DROP TABLE BIBETL01.BC_PAYMENT_RELATION_HIS_BACK;
```

```
Table BIBETL01.BC_PAYMENT_RELATION_HIS_BACK dropped.
```

```
Elapsed: 00:00:00.303
```

```
SQL> DROP TABLE BIBETL01.BC_ACCT_HIS_BACK;
```

```
Table BIBETL01.BC_ACCT_HIS_BACK dropped.
```

```
Elapsed: 00:00:00.167
```

```
SQL> DROP TABLE BIBETL01.BES_SHOPPINGCART_ITEM_HIS_BACK;
```

```
Table BIBETL01.BES_SHOPPINGCART_ITEM_HIS_BACK dropped.
```

```
Elapsed: 00:00:00.255
```

```
SQL> DROP TABLE BIBETL01.BC_SUB_DFT_ACCT_HIS_BACK;
```

```
Table BIBETL01.BC_SUB_DFT_ACCT_HIS_BACK dropped.
```

```
Elapsed: 00:00:00.440
```

```
SQL> DROP TABLE BIBETL01.INF_OFFERING_INSTALLMENT_BHIS_BACK;
```



Jul 16th – Everything **NOT** OK – Network waits ~ **112,665** seconds during 6 hour Window

Host Name	Platform	CPU	Cores	Sockets	Memory (GB)
lthbmbd102.telcel.com	IBM zSeries Based Linux	36			566.41

Snap Id	Snap Time	Sessions	Cursors/Session	Instances	Pluggable Databases Open
Begin Snap: 7763	16-Jul-25 00:37:54	239	1.2	2	4
End Snap: 7775	16-Jul-25 06:37:58	163	1.3	2	4
Elapsed:	360.06 (mins)				
DB Time:	4,153.00 (mins)				

Report Summary

Top ADDM Findings by Average Active Sessions

Finding Name	Avg active sessions of the task	Percent active sessions of finding	Task Name	Begin Snap Time	End Snap Time
Unusual "Network" Wait Event	18.37	27.66	ADDM:1204327370_2_7766	16-Jul-25 01:37	16-Jul-25 02:08
Unusual "Network" Wait Event	22.12	22.74	ADDM:1204327370_2_7767	16-Jul-25 02:08	16-Jul-25 02:38
Unusual "Network" Wait Event	15.66	26.97	ADDM:1204327370_2_7768	16-Jul-25 02:38	16-Jul-25 03:07
Unusual "Network" Wait Event	22.12	16.39	ADDM:1204327370_2_7767	16-Jul-25 02:08	16-Jul-25 02:38
"User I/O" wait Class	22.12	15.65	ADDM:1204327370_2_7767	16-Jul-25 02:08	16-Jul-25 02:38

Top Process Types by Wait Class

- This table displays top process types ordered by wait class

Process Type	Description	Wait Class Type	Wait Time (sec)
	Others (foregrounds + unclassified backgrounds)	Network	112,665.29
	Others (foregrounds + unclassified backgrounds)	User I/O	16,725.79
P	Parallel query slave	User I/O	11,292.32
	Others (foregrounds + unclassified backgrounds)	Cluster	7,011.52
	Others (foregrounds + unclassified backgrounds)	Commit	6,221.97
LGWR	Redo etc.	System I/O	3,558.56
P	Parallel query slave	Cluster	2,369.57
	Others (foregrounds + unclassified backgrounds)	Other	2,118.65
LG	Log Writer Slave	System I/O	1,474.12
LM	global cache service process	Other	937.49
	Others (foregrounds + unclassified backgrounds)	Concurrency	725.07
P	Parallel query slave	Other	617.58



Aug 1st – Everything **NOT** OK – Network waits ~ **224,306** seconds during 6 hour Window

Host Name	Platform	CPU	Cores	Sockets	Memory (GB)
ithbmbd102.telcel.com	IBM zSeries Based Linux	36			566.41

	Snap Id	Snap Time	Sessions	Cursors/Session	Instances	Pluggable Databases Open
Begin Snap:	8531	01-Aug-25 00:38:15	491	1.2	2	4
End Snap:	8543	01-Aug-25 06:38:37	482	1.5	2	4
Elapsed:		360.37 (mins)				
DB Time:		25,998.98 (mins)				

Report Summary

Top ADDM Findings by Average Active Sessions

Finding Name	Avg active sessions of the task	Percent active sessions of finding	Task Name
Top SQL Statements	108.90	88.18	ADDM:1204327370_2_8
Top SQL Statements	94.49	85.82	ADDM:1204327370_2_8
Top SQL Statements	85.58	66.48	ADDM:1204327370_2_8
Unusual "Network" Wait Event	108.90	33.72	ADDM:1204327370_2_8
Unusual "Network" Wait Event	94.49	22.09	ADDM:1204327370_2_8

Top Process Types by Wait Class

- This table displays top process types ordered by wait class

Process Type	Description	Wait Class Type	Wait Time (sec)
P	Parallel query slave	User I/O	366,467.12
	Others (foregrounds + unclassified backgrounds)	Network	224,306.44
	Others (foregrounds + unclassified backgrounds)	User I/O	110,264.23
	Others (foregrounds + unclassified backgrounds)	Cluster	54,573.00
	Others (foregrounds + unclassified backgrounds)	Commit	42,811.97
	Others (foregrounds + unclassified backgrounds)	Concurrency	22,353.02
	Others (foregrounds + unclassified backgrounds)	Other	18,647.68
P	Parallel query slave	Cluster	13,000.92
LG	Log Writer Slave	System I/O	10,424.00
P	Parallel query slave	Other	3,614.71
LG	Log Writer Slave	Other	3,309.41
	Others (foregrounds + unclassified backgrounds)	Configuration	3,190.46
LM	global cache service process	Other	3,142.85



Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module
18,600.75	35,499	0.52	5.94	90.28	0.00	<u>48xq7nkc99hs8</u>	osh@zsvmfdstgrpap01.telcel.com (TNS V1-V3)
16,532.09	73,338	0.23	5.28	97.97	0.00	<u>bxbmqm5qkhz16</u>	osh@zsvmfdstgrpap01.telcel.com (TNS V1-V3)
8,939.94	7,873	1.14	2.85	77.08	0.19	<u>1ktwbsvmk9hy0</u>	osh@zsvmfdstgrpap01.telcel.com (TNS V1-V3)
7,866.18	1	7,866.18	2.51	75.52	17.31	<u>2q6pfc1mfz155</u>	osh@zsvmfdstgrpap01.telcel.com (TNS V1-V3)
7,531.12	4	1,882.78	2.40	90.61	7.85	<u>du27gz9ggvjwy</u>	osh@ithhpetlipap002 (TNS V1-V3)

Recommendation 1: Application Analysis

Estimated benefit is 14.57 active sessions, 48.84% of total activity.

Action

Investigate the cause for high "SQL*Net more data to client" waits.
Refer to Oracle's "Database Reference" for the description of this wait event.

Action

Look at the "Top SQL Statements" finding for SQL statements consuming significant time on the "SQL*Net more data to client" wait event. For example, the SELECT statement with SQL_ID "du27gz9ggvjwy" is responsible for 21% of these waits.

Recommendation 2: Application Analysis

Estimated benefit is 14.37 active sessions, 48.18% of total activity.

Action

Investigate the cause for high "SQL*Net more data to client" waits in Module "osh@ithhpetlipap002 (TNS V1-V3)".

Recommendation 3: Application Analysis

Estimated benefit is 14.37 active sessions, 48.18% of total activity.

Action

Investigate the cause for high "SQL*Net more data to client" waits in Service "mfbbet1".

Even though we switched to Hipersockets
(which helped a little), connections
From legacy Data Stage were slow
Network wise





Problem Resolved – Aug 26th

Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
ithbmbd102.telcel.com	IBM zSeries Based Linux	54			763.29

	Snap Id	Snap Time	Sessions	Cursors/Session	Instances	Pluggable Databases Open
Begin Snap:	9923	30-Aug-25 00:38:15	339	1.3	2	4
End Snap:	9935	30-Aug-25 06:38:01	166	1.3	2	4
Elapsed:		359.76 (mins)				
DB Time:		1,977.32 (mins)				

Report Summary

Top ADDM Findings by Average Active Sessions

Finding Name	Avg active sessions of the task	Percent active sessions of finding	Task Name	Begin Snap Time	End Time
"User I/O" wait Class	10.67	32.72	ADDM:1204327370_2_9930	30-Aug-25 03:37	30-Aug-25 04:08
"User I/O" wait Class	6.84	39.45	ADDM:1204327370_2_9931	30-Aug-25 04:08	30-Aug-25 04:38
"User I/O" wait Class	6.91	31.17	ADDM:1204327370_2_9935	30-Aug-25 06:07	30-Aug-25 06:38
Global Cache Messaging	10.67	5.94	ADDM:1204327370_2_9930	30-Aug-25 03:37	30-Aug-25 04:08
Commits and Rollbacks	10.67	4.71	ADDM:1204327370_2_9930	30-Aug-25 03:37	30-Aug-25 04:08

Top Process Types by Wait Class

- This table displays top process types ordered by wait class

Process Type	Description	Wait Class Type	Wait Time (sec)
	Others (foregrounds + unclassified backgrounds)	User I/O	22,562.74
	Others (foregrounds + unclassified backgrounds)	Cluster	8,719.04
P	Parallel query slave	User I/O	8,588.55
	Others (foregrounds + unclassified backgrounds)	Commit	8,177.48
LG	Log Writer Slave	System I/O	3,611.08
LGWR	Redo etc.	System I/O	3,250.83
P	Parallel query slave	Cluster	2,009.23
	Others (foregrounds + unclassified backgrounds)	Other	1,908.68
	Others (foregrounds + unclassified backgrounds)	Network	1,706.98
LM	global cache service process	Other	1,252.38
	Others (foregrounds + unclassified backgrounds)	Concurrency	926.76



AR_INVOICE Timings - Final

Fecha Inicio	Cantidad Registros Verify	Tiempo Ejec Minutos	Fecha Inicio	Fecha Fin
8/4/2025	257,888	258.888	05/08/2025 01:13:46 AM	05/08/2025 05:53:43 AM
8/5/2025	1,095,886	279.950	06/08/2025 01:07:53 AM	06/08/2025 05:45:37 AM
8/6/2025	1,237,493	277.750	07/08/2025 01:17:51 AM	07/08/2025 05:57:46 AM
8/7/2025	666,877	279.933	08/08/2025 01:23:44 AM	08/08/2025 05:55:09 AM
8/8/2025	899,265	271.417	09/08/2025 01:24:49 AM	09/08/2025 05:55:48 AM
8/9/2025	1,229,387	270.967	10/08/2025 01:39:20 AM	10/08/2025 06:11:15 AM
8/10/2025	262,834	271.933	11/08/2025 01:10:29 AM	11/08/2025 05:54:34 AM
8/11/2025	982,874	284.083	12/08/2025 01:07:11 AM	12/08/2025 05:50:15 AM
8/12/2025	1,338,492	283.050	13/08/2025 01:20:10 AM	13/08/2025 06:04:42 AM
8/13/2025	1,099,772	284.550	22/08/2025 01:04:28 AM	22/08/2025 04:39:14 AM
8/22/2025	599,109	214.767	23/08/2025 01:23:33 AM	23/08/2025 04:46:15 AM
8/23/2025	787,783	202.717	24/08/2025 01:46:42 AM	24/08/2025 04:52:50 AM
8/24/2025	931,324	186.150	25/08/2025 01:27:08 AM	25/08/2025 04:47:06 AM
8/25/2025	985,753	199.983	26/08/2025 12:58:58 AM	26/08/2025 02:05:54 AM
8/26/2025	879,575	66.933	27/08/2025 01:11:32 AM	27/08/2025 02:19:05 AM
8/27/2025	529,562	67.550	28/08/2025 12:41:40 PM	28/08/2025 01:23:41 PM
8/28/2025	870,456	42.033	29/08/2025 01:15:43 AM	29/08/2025 02:34:56 AM
8/29/2025	1,309,153	79.217	30/08/2025 01:36:55 AM	30/08/2025 02:44:30 AM
8/30/2025	2,125,485	67.583	31/08/2025 01:20:59 AM	31/08/2025 02:24:21 AM
8/31/2025	823,297	63.383	01/09/2025 12:55:14 AM	01/09/2025 02:06:29 AM
9/1/2025	291,693	71.233	02/09/2025 01:24:04 AM	02/09/2025 02:23:00 AM
9/2/2025	2,228,869	58.933	03/09/2025 01:32:31 AM	03/09/2025 02:37:28 AM
9/3/2025	802,017	64.933	04/09/2025 01:02:10 AM	04/09/2025 02:16:55 AM
9/4/2025	516,555	74.733	05/09/2025 01:01:01 AM	05/09/2025 02:11:26 AM
9/5/2025	941,444	70.417	06/09/2025 01:16:33 AM	06/09/2025 02:38:23 AM
9/6/2025	1,181,452	81.817		





MongoDB is a NoSQL document database designed for ease of application development and scaling

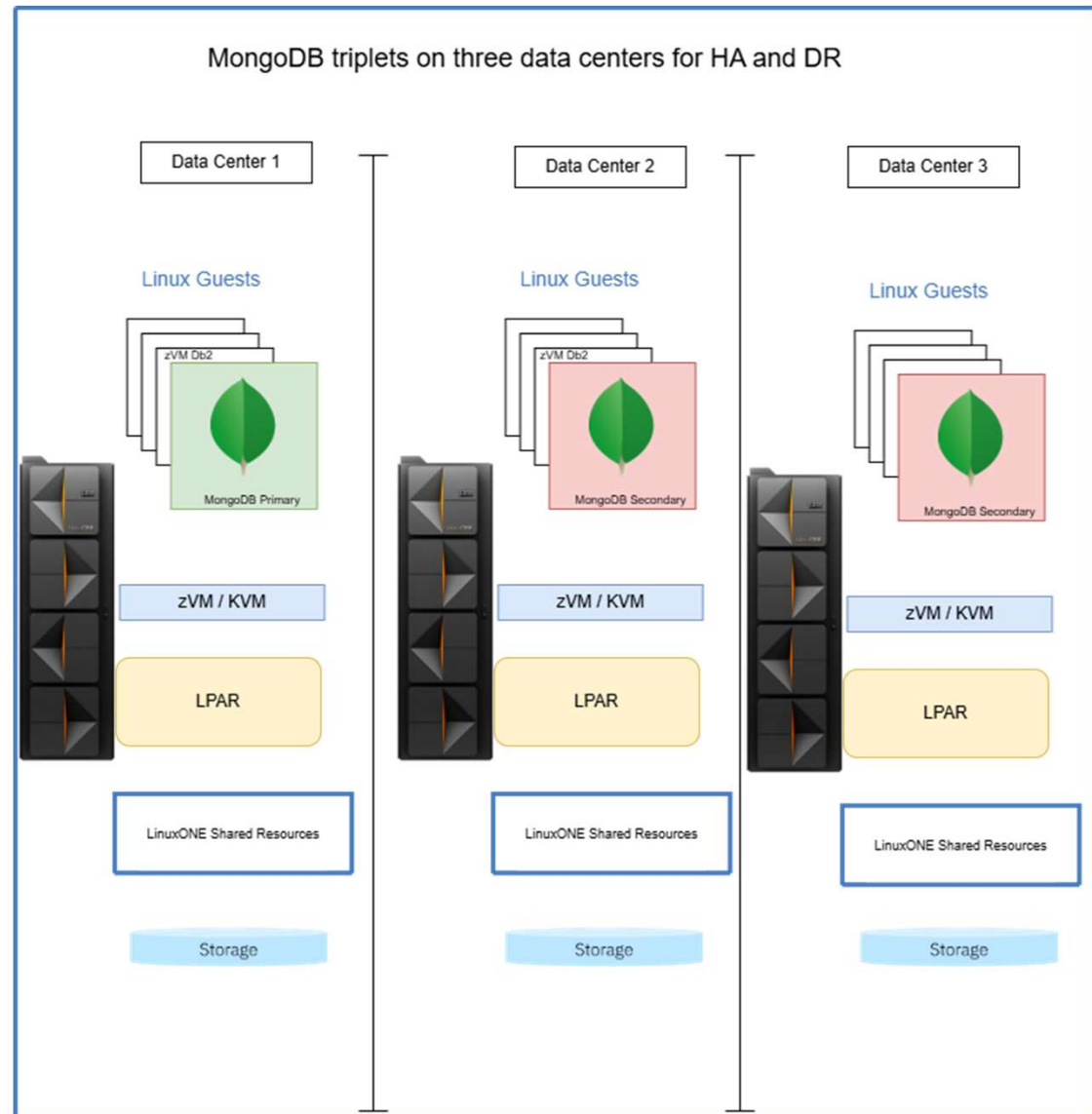
It stores data in BSON format, like JSON document structure so that it can operate with dynamic and unstructured data

```
{  
  "name": "Alice",  
  "age": 30,  
  "email": alice@example.com  
}
```

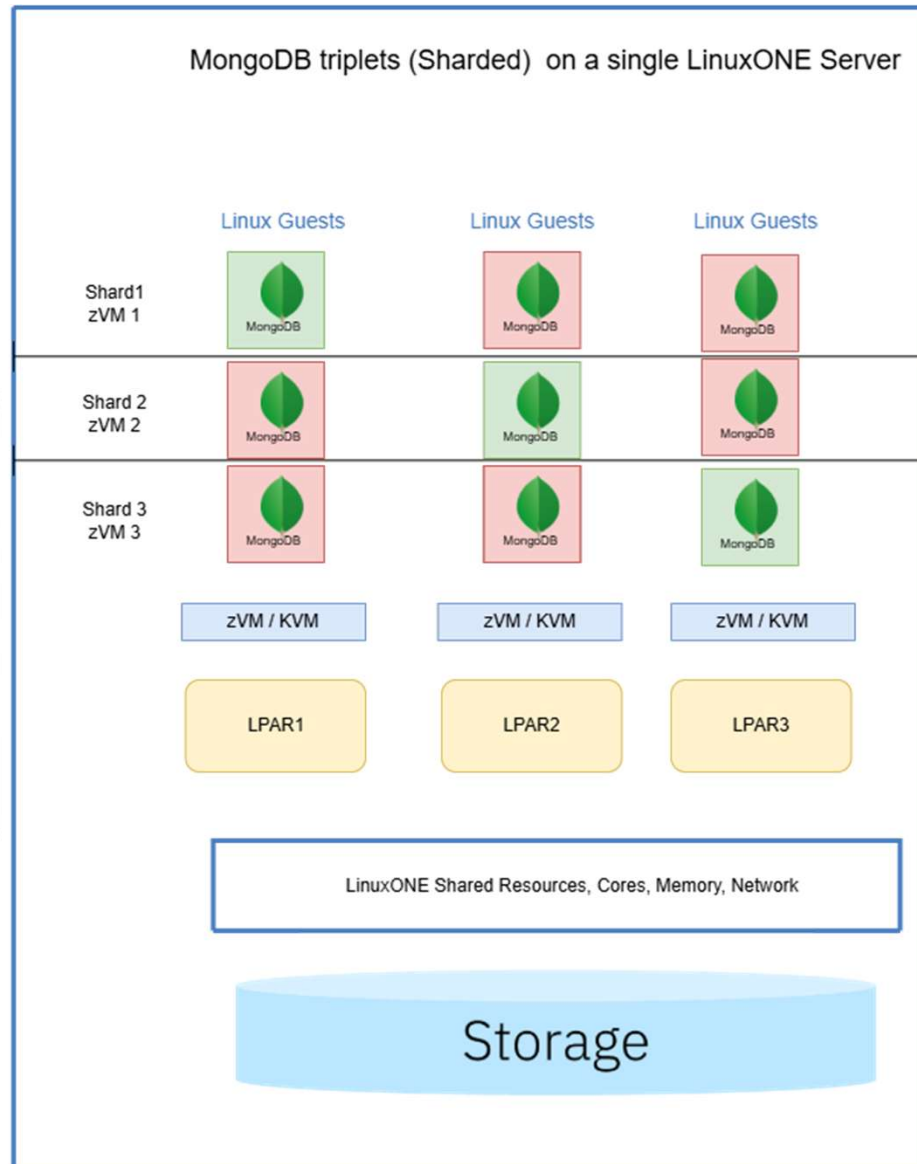
MongoDB excels at handling large volumes of data. The word Mongo is derived from the word humongous

Flexible data model, high performance and rich query Language

Primary and
secondary
replicas in
different
LinuxONE located
in distinct sites



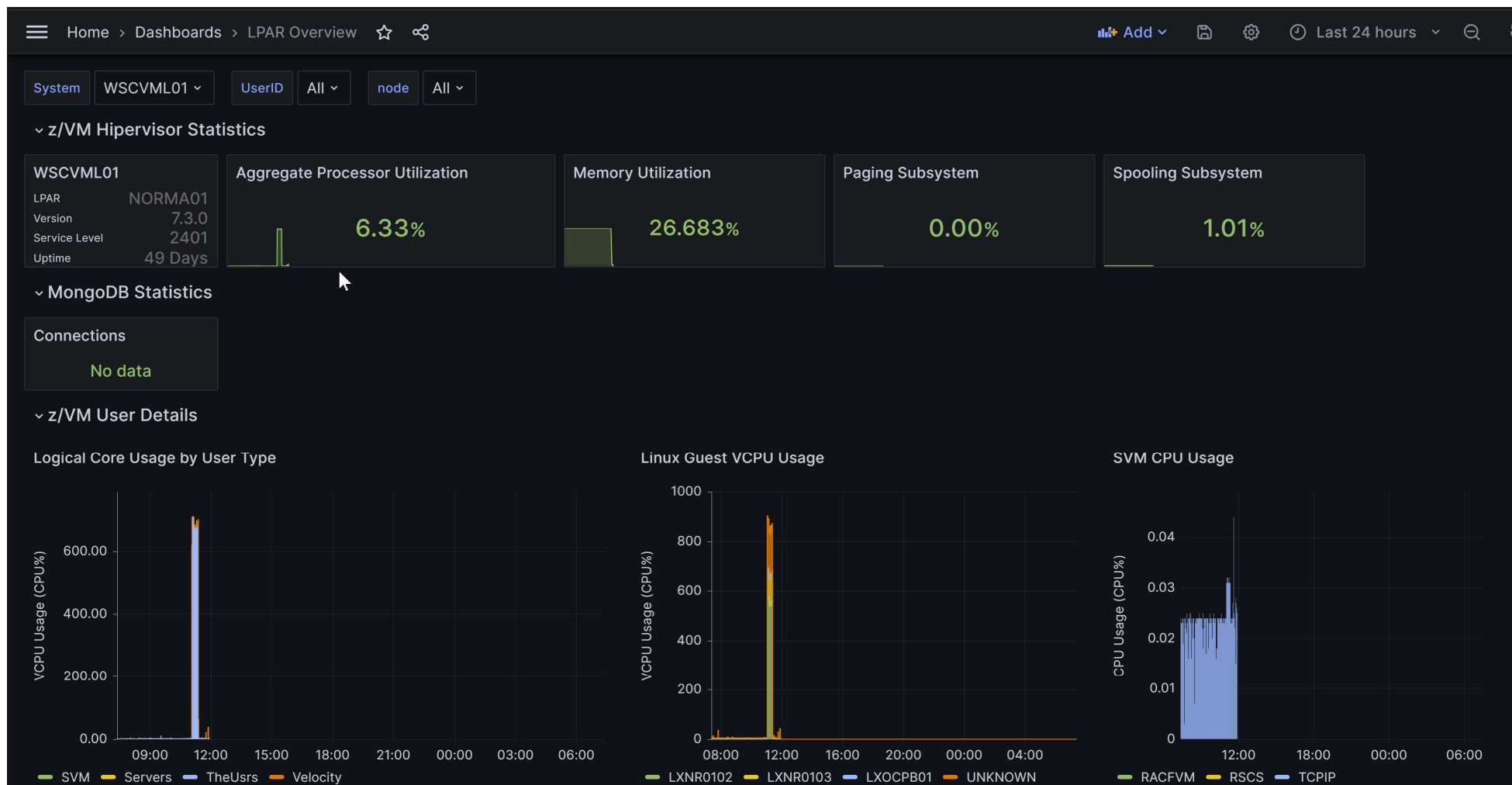
Sharding inside the same LinuxONE



MongoDB Metrics on the Same Grafana Dashboard



z/VM & LPAR – Memory and CPU Data from Grafana Dashboard





MongoDB Backups on LinuxONE



- Large MongoDB customer planning to go to production with IBM LinuxONE, Flash/Pure Storage and MongoDB as the major components of their solution (9 PB database size)
- Backup and restore is done through MongoDB Ops Manager and has the following challenges:
 - Backup and restore processes take more than a week to complete with around 2 TB of data size per shard
 - RTO and RPO requirements (RTO 4 hours, RPO 8 hours – 3 full snaps / day, 10% change/day, retention 14 days)
 - Backup and restore lengthy time imposes restriction on shard sizes and increases the number of nodes
 - Non disruptive I/O to MongoDB cluster during backup operation

1 Initiate the backup through MongoDB backup api calls

http://sb1.cpolab.ibm.com:8080/api/public/v1.0/backup/third_party/group/67eed68e0ba1970a07e16b77/clusters

```
{
  "clusterId": "695154083e4bf100f30af512",
  "clusterName": "sgcCluster",
  "groupId": "67eed68e0ba1970a07e16b77",
  "jobId": "695154083e4bf100f30af512",
  "state": "ACTIVE"
}
```

1.1 Start the Snapshot

Post the following api to start OpsManager Snapshot

[http://sb1.cpolab.ibm.com:8080/api/public/v1.0/backup/third_party/group/67eed68e0ba1970a07e16b77/clusters/695154083e4bf100f30af512/snapshot/\\$sid/start](http://sb1.cpolab.ibm.com:8080/api/public/v1.0/backup/third_party/group/67eed68e0ba1970a07e16b77/clusters/695154083e4bf100f30af512/snapshot/$sid/start)

1.2 Poll until snapshot state is "Ready"

Use the following api to poll until the status change to "Ready" from "Pending"

http://sb1.cpolab.ibm.com:8080/api/public/v1.0/backup/third_party/group/67eed68e0ba1970a07e16b77/clusters/695154083e4bf100f30af512/snapshot/6953ee773e4bf100f3721f5b

when the response was Ready:
Initiate the snapshots in the Flash Systems

2 Finish the snapshots using Mongoddb backup APIs

Post the following api to finish OpsManager Snapshot and save the metadata, we need that during restore

[http://sb1.cpolab.ibm.com:8080/api/public/v1.0/backup/third_party/group/67eed68e0ba1970a07e16b77/clusters/695154083e4bf100f30af512/snapshot/\\$sid/finish](http://sb1.cpolab.ibm.com:8080/api/public/v1.0/backup/third_party/group/67eed68e0ba1970a07e16b77/clusters/695154083e4bf100f30af512/snapshot/$sid/finish)

3 initiate the restore through MongoDB restore api calls

3.1 Create OpsManager Restore

Post the following api to create OpsManager Restore and note the restoreId

http://sb1.cpolab.ibm.com:8080/api/public/v1.0/backup/third_party/group/67eed68e0ba1970a07e16b77/clusters/695154083e4bf100f30af512/restore

And pass with following parameters snapshot metadata and nodeids in the body

3.2 Start the OpsManager Restore

Post the following api to start the OpsManager Restore status by passing the restoreId

[http://sb1.cpolab.ibm.com:8080/api/public/v1.0/backup/third_party/group/67eed68e0ba1970a07e16b77/restore/\\$RID/start](http://sb1.cpolab.ibm.com:8080/api/public/v1.0/backup/third_party/group/67eed68e0ba1970a07e16b77/restore/$RID/start)

3.3 Get the OpsManager Restore Status

Use the following api to get the OpsManager Restore status by passing the restoreId

[http://sb1.cpolab.ibm.com:8080/api/public/v1.0/backup/third_party/group/67eed68e0ba1970a07e16b77/restore/\\$RID/](http://sb1.cpolab.ibm.com:8080/api/public/v1.0/backup/third_party/group/67eed68e0ba1970a07e16b77/restore/$RID/)

The restore state will be "PENDING" to "SHUTDOWN_IN_PROGRESS"

Poll until restore state changes from "SHUTDOWN_IN_PROGRESS" to "COPY_FILES"

SAN restore files

Resync files to other replicas (via SAN replication)

`rsync -avzr /fimongo1/ root@mdb4:/fimongo1`
`rsync -avzr /fimongo1/ root@mdb2:/fimongo1`

Tell MongoDB OpsManager Files have been copied via API:

http://sb1.cpolab.ibm.com:8080/api/public/v1.0/backup/third_party/group/67eed68e0ba1970a07e16b77/restore/6942ac5b4ad5445e0250ae93/filesCopied

IBM Bob in the Benchmark Center



MongoDB Environment Health Check Report

Date: May 7, 2026
Time: 12:07 PM EDT
Performed From: mdb-rh1 (129.40.36.97)

Executive Summary

✔ Overall Status: HEALTHY

All operational nodes are functioning correctly. The environment consists of 14 active nodes with proper automation agent integration and stable MongoDB processes.

Infrastructure Overview

Node Inventory

Node Type	Total	Active	Offline (Intentional)
Routers	3	3	0
Config Servers	3	1	2
Data Nodes	9	9	0

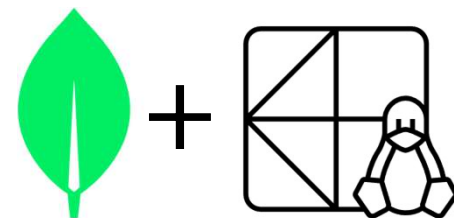
MongoDB & IBM Confluent Kafka Run Great on LinuxONE!

Multi-Agentive Ticket-Based Complaint Resolution System

September 4, 2025

In the AI-first era of customer experiences, financial institutions face increasing pressure to deliver faster, smarter, and more cost-effective service, all while maintaining transparency and trust.

This blog introduces a multi-agentive architecture, co-developed with MongoDB and **Confluent Data Streaming Platform**, designed to automate ticket-based complaint resolution. It leverages the AI-native capabilities of both platforms within a real-time event streaming environment. With continuously active agents orchestrated across the entire system, financial institutions can automatically resolve common customer issues in seconds, significantly improving resolution times and boosting customer satisfaction.



| SHARE THIS



| AUTHOR



Shashank Pandey
Solutions Architect



Scott Sappenfield
(Confluent)
Partner Solutions Engineer

Source: <https://www.mongodb.com/company/blog/technical/multi-agentive-ticket-based-complaint-resolution-system>

LinuxONE Data Integration Summary – DataGate with LinuxONE



IBM Z

IBM Z Data

- Db2z
- IMS
- VSAM

Data Gate

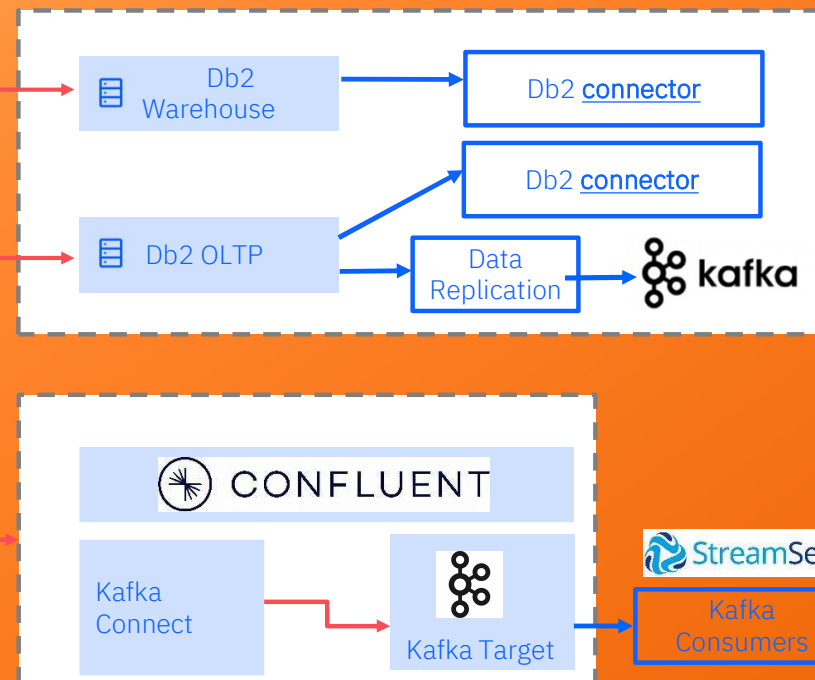
Db2z data only

Data Gate for Confluent*
Announcement: May 5th, 2026
GA: June 12th, 2026

- Direct access and virtualization
- Data synchronization / copy

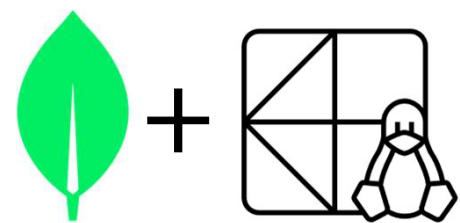
DVM : Data Virtualization Manager for z/OS

LinuxONE

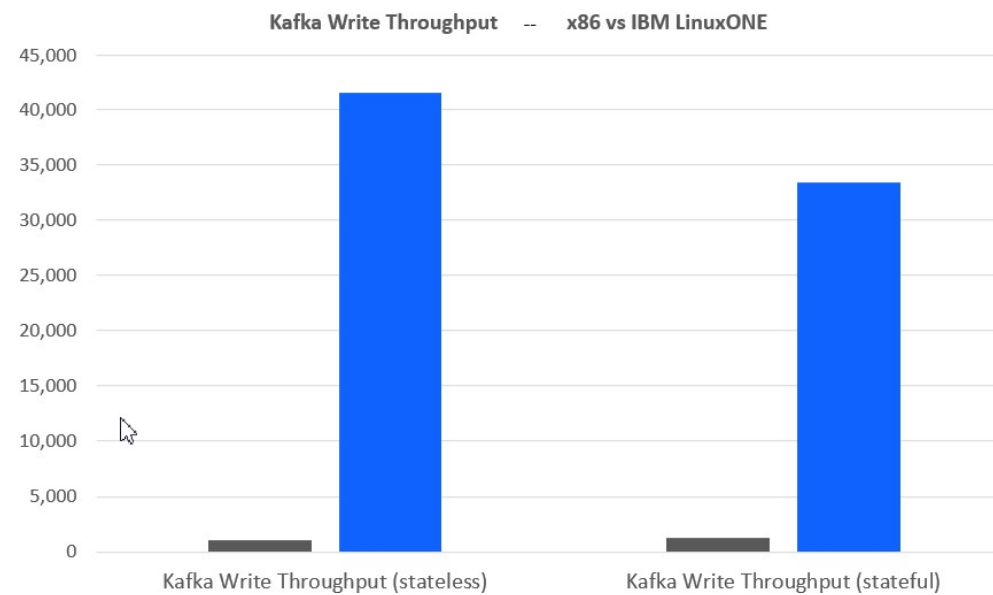


30x improved write throughput after migrating Confluent Platform from x86 to LinuxONE

Confluent Kafka Performance Benefits on IBM LinuxONE



Overall Performance	x86	LinuxONE
OpenShift Stateless Applications		
Startup/Recovery Time	~5 min	~10 seconds
Kafka Topic Write Throughput	652-1077 msg/s	41.5k msg/s
Data Processing Time Per Million Records	1000 seconds	26 seconds
OpenShift Stateful Applications		
Startup/Recovery Time	~30 min	~10 min
Kafka Topic Write Throughput	771-1266 msg/s	33.4k msg/s
Data Processing Time Per Million Records	1297 seconds	98 seconds

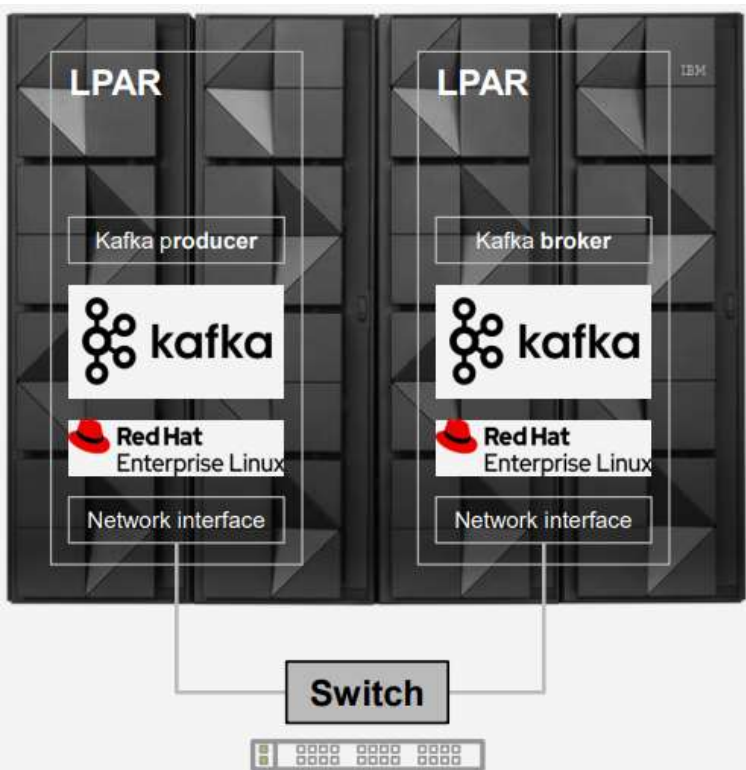




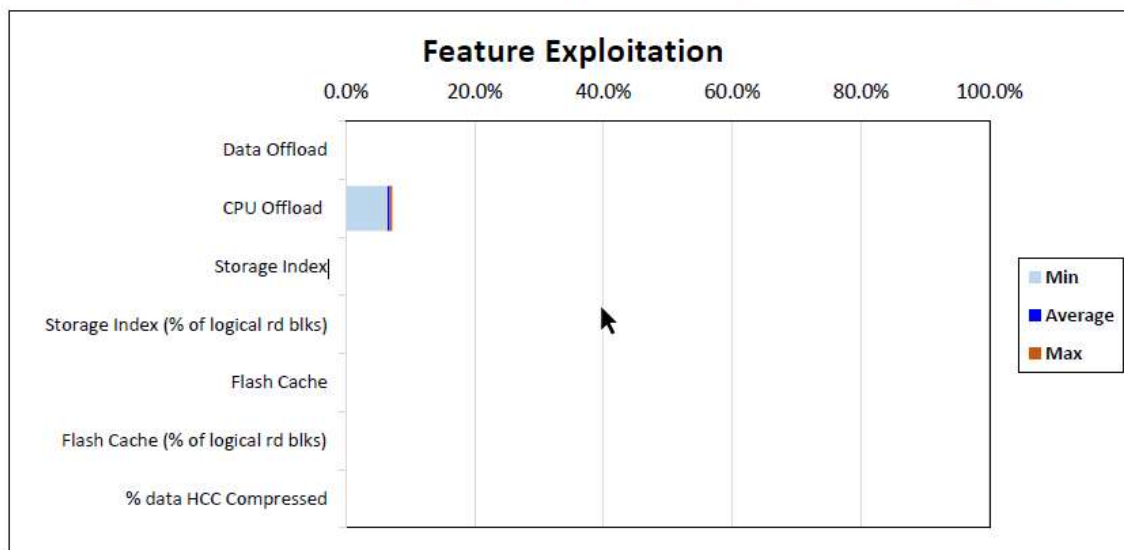
Streaming Data Opportunities

Client Goal – Increase Data Streaming workload by 2X
-> Move workload from Exadata Appliance beside Kafka

```
SELECT /*+ INDEX(TIMESERIES_TRANS) NO_RESULT_CACHE */
```



Mirrored Writes	Minimal Negative Net Impact	Between 0.4% and 0.4% (average 0.4%) of the I/O requests were writes. Because of ASM HIGH redundancy, Exadata must perform 3 physical writes for every DB block that is updated, which can add significant I/O overhead.
-----------------	-----------------------------	--





Challenge: Improve the customer experience through data while staying compliant with industry regulations.

Solution: Implemented Confluent Platform to stream data in real-time, migrate to an event-based architecture and embark on a microservices transformation.

Results:

- Greatly reduced costs
- Increased efficiency in application building
- Lowered anomaly detection time from weeks to seconds
- Implemented data reuse across teams for relevant business insights
- Higher quality, faster and scalable customer services

“There isn't another product that competes with Confluent Platform. Streaming data as events enables completely new ways for solving problems at scale.” — Head of Engineering, Enterprise Cloud

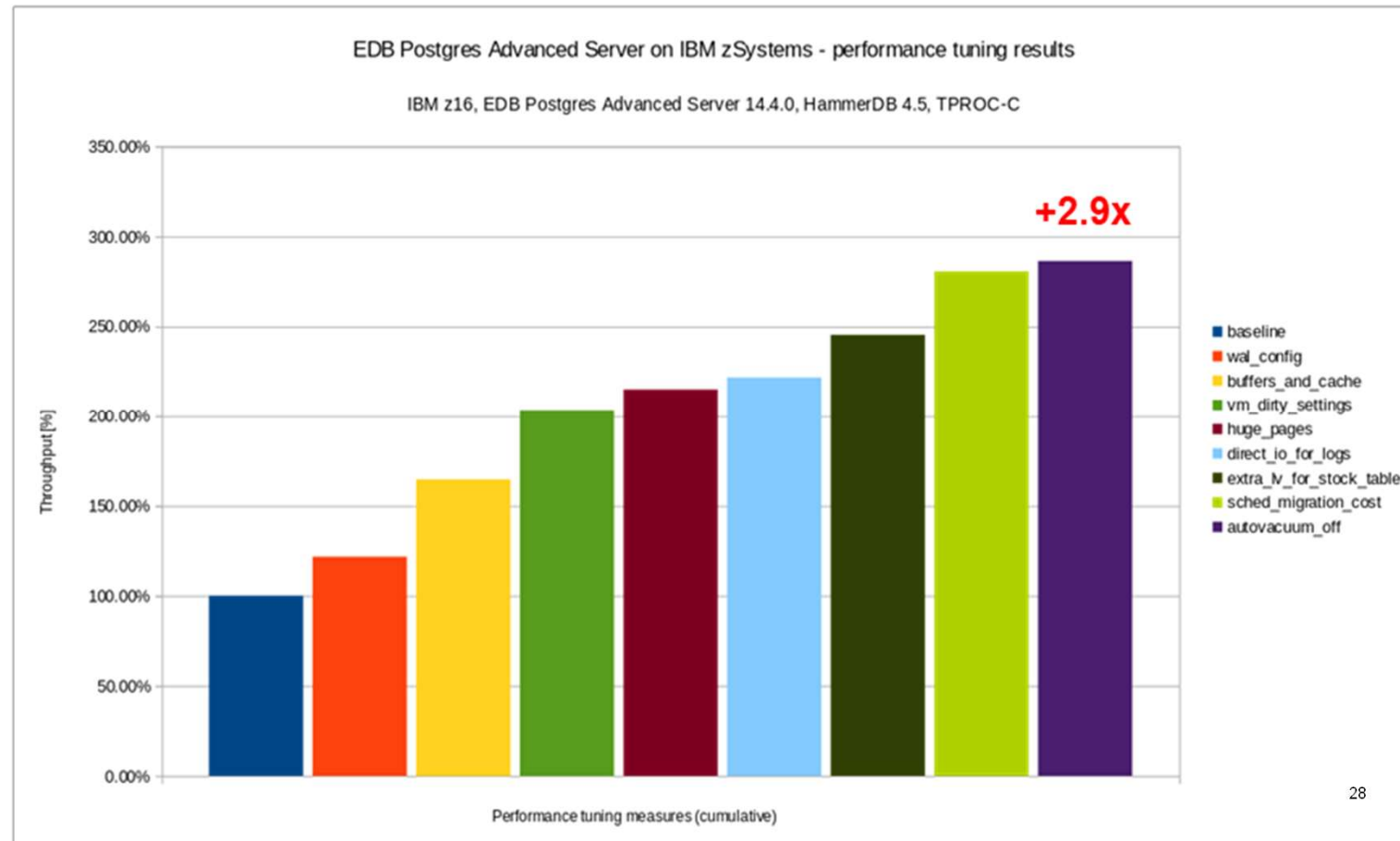
Industry: Financial Services | **Geo:** AMER | **Product:** Confluent Platform



EnterpriseDB on IBM LinuxONE Tuning Guide

The *Systems Performance – Linux® on IBM Z®* team has performed **extensive research** on the performance of EDB Postgres Advanced Server on the IBM Z platform (on which IBM® LinuxONE is based).

The outcome of this research is now available as a [tuning guide](#), which summarizes the key findings and recommendations of this study. The recommendations given in this report can **significantly boost** the performance of EDB Postgres environments on IBM Z and IBM® LinuxONE systems.





Summary