Make IT

SIOUG 2023

Merjenje performanc pred in po Oracle migraciji

Speaker:

- **Urh Srečnik** <urh.srecnik@abakus.si>

AS Abakus As na disku.

ORACLE Gold Partner

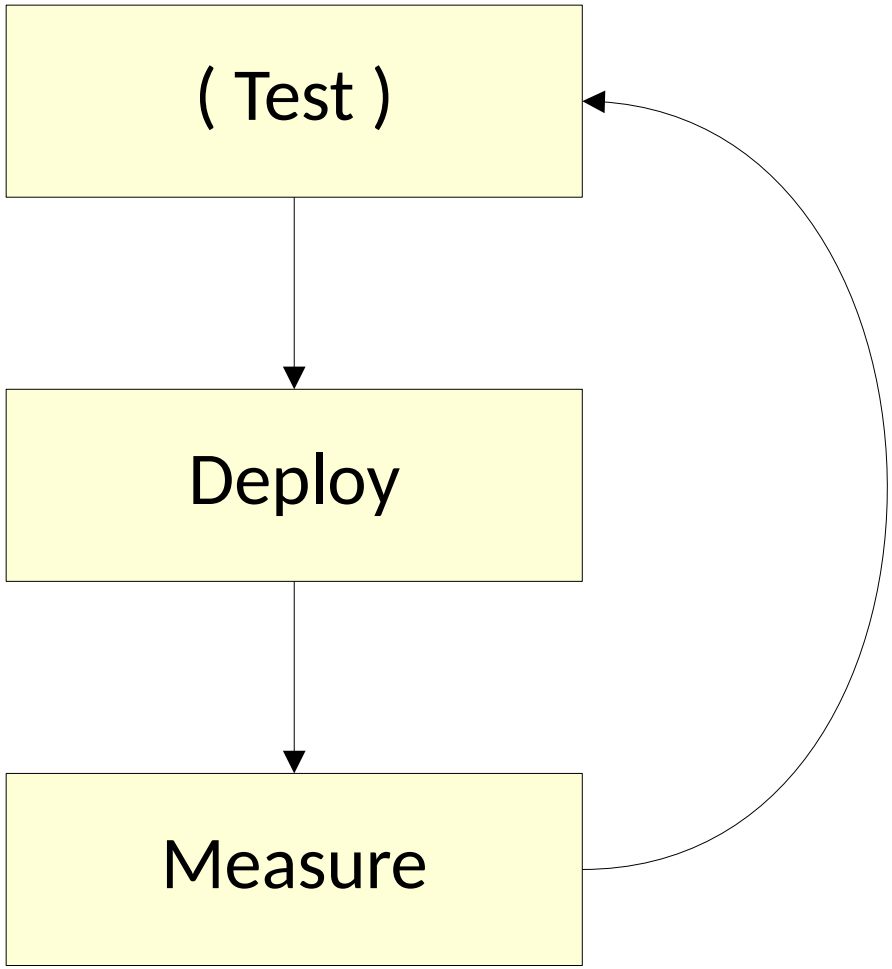
Abakus Plus d.o.o.

- Infrastructure Team
  - Services
    - OS & NET admin
    - DBA, Programming
  - Applications
    - **APPM**
    - **Backup Server & Deja Vu**
    - Arbiter
- Development Team
  - Enterprise Applications
  - Document Management
  - Newspaper Distribution
  - Flight Information System

Customers

Process

Some changes might not be planned, thus not tested before deployment.



Test Prerequisite

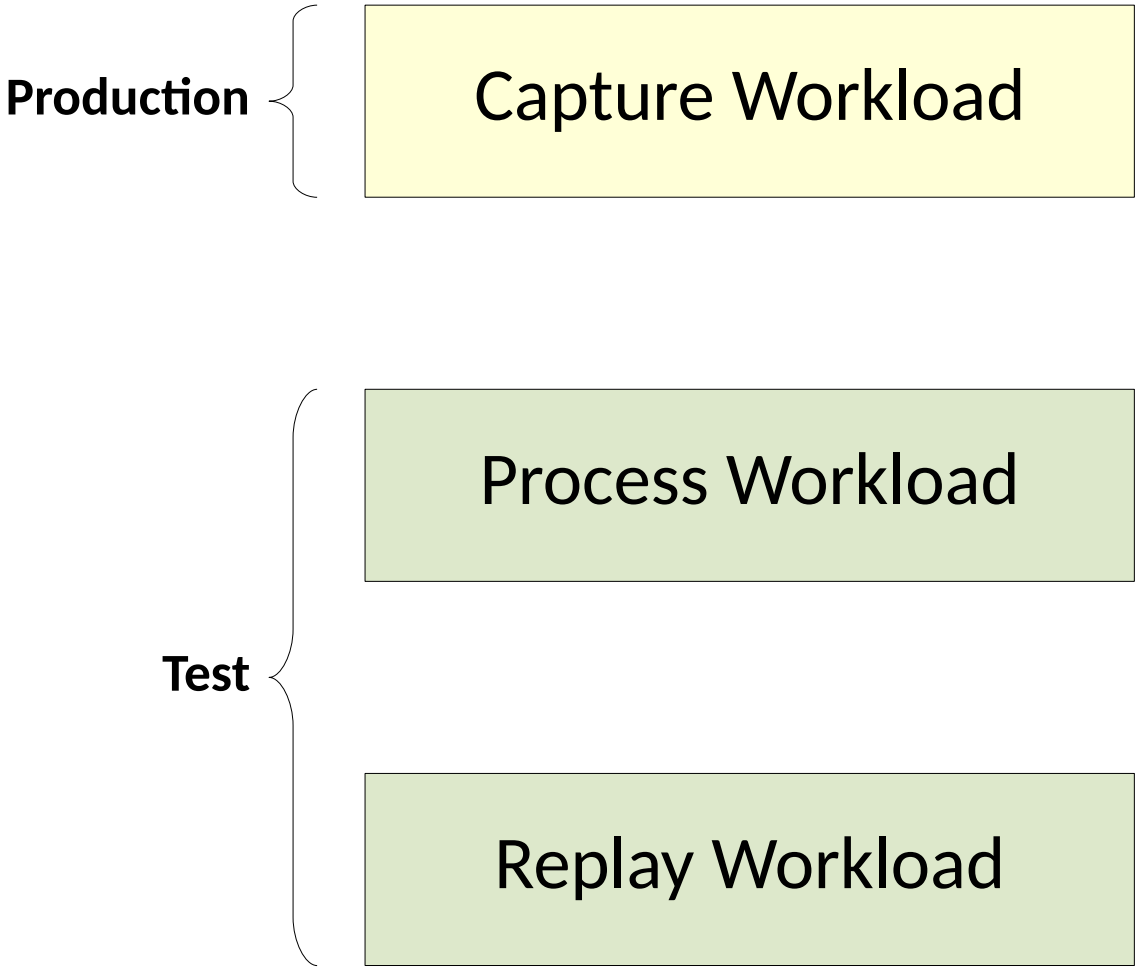
- Production Database Clone
  - rman
  - snapshots
  - standby
  - …
  - or:

http://www.abakus.si/sl/produkti/programska_oprema/dejavu

Test

- Let users do the testing
- Let developers do the testing
- Let’s test automatic workload generators
- Capture and Replay producton workload

Real Application Testing



»Manual« Testing

- Custom Scripts (SQL, Bash, ...)
- Live Testers (developers?)

- Open-Source Workload generators
  - Swing Bench
    http://www.dominicgiles.com/swingbench.html
  - RWLoadSim (maintained by Oracle)
    https://github.com/oracle/rwloadsim
  - SLOB
    https://kevinclosson.net/slob/

Measure What?

- Wall Time
- LIO
- ASH
- Undo, Temp Usage
- IO Throughput
- AWR reports
- **Response Time**
- …

… and Compare To

- Historical measurements (aka »baselines«)

Oracle's Official Solution

- Oracle Real Application Testing
  - Uses ASH/AWR to measure prev/curr performance and display the differences
  - But… this is only available on Enterprise Edition as Extra Cost Option.

Abakus's Preferred Solution

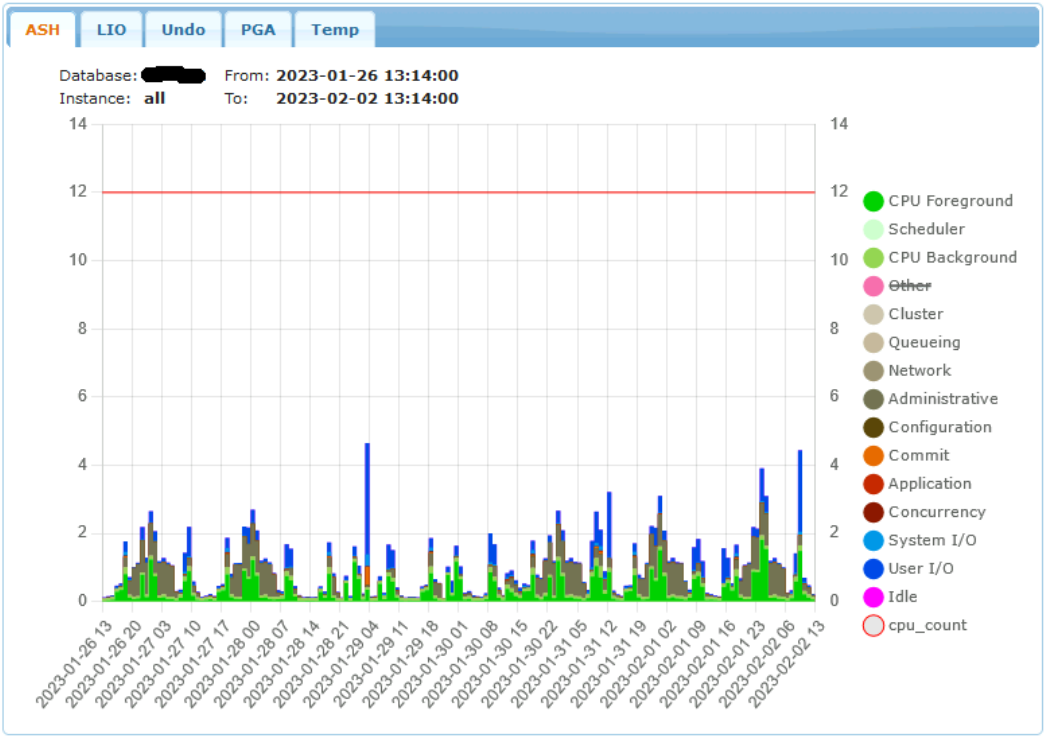
- APPM – Built using Open Source tech (Java, Postgres)

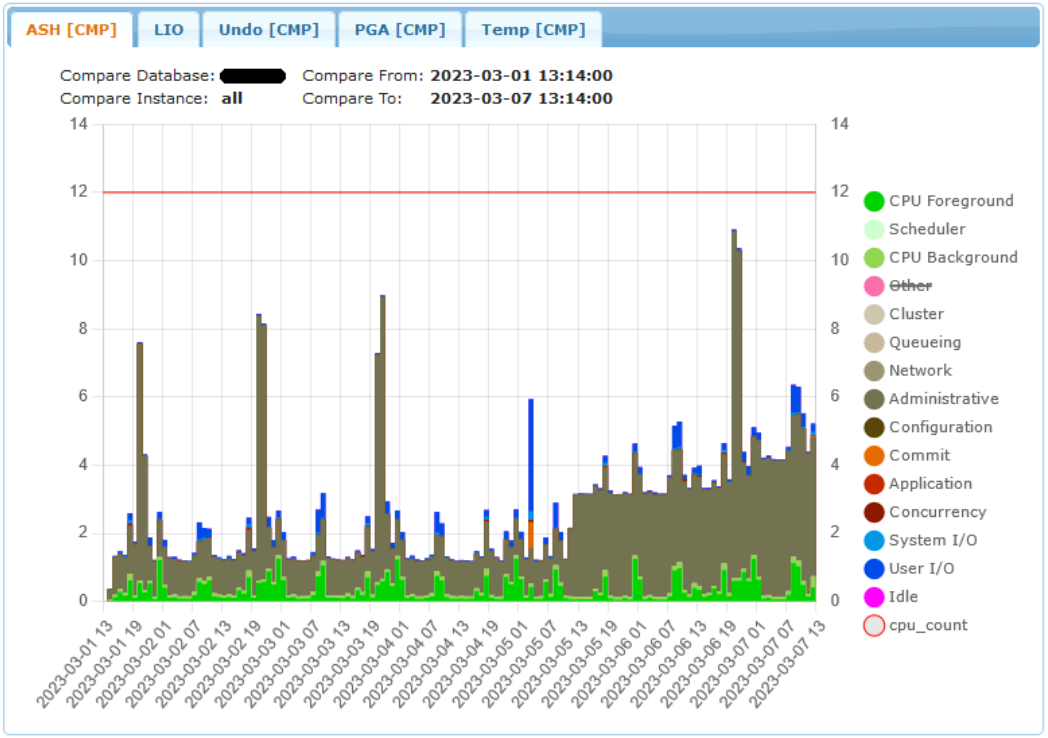
http://www.abakus.si/sl/produkti/programska_oprema/appm

Migration 1: SE → EE

- Database edition changes from Standard to Enterprise Edition
- Everything else (hardware, operating system, storage, etc) stays **exactly the same.**

Migration: SE → EE





Top Table (compare mode)

Activity	Activity [CMP]	SQL_ID (per exec)	SQL Text	Duration	Duration [CMP]	Absolute Diff	Ratio
		6wx9w5c6b21wx	SELECT /*+ PARALLEL(2) *[redacted]	00d 01:29:50.57	00d 01:11:39.43	00d 00:18:11	+20.24%
		420huc7uqrk54	/* SQL Analyze(1) */ select /*+ full(t) parall	00d 00:46:14.67	00d 00:32:29.67	00d 00:13:45	+29.73%
		as2dr3ag24gay	select not_stale.obj# from (select s.obj# obj#, co	00d 00:21:08.12	00d 00:10:59.91	00d 00:10:08	+47.96%
		fzvx3vkv5mf8z	select /*+ opt_param('_optimizer_use_auto_indexes'	00d 00:16:59.00	00d 00:12:31.33	00d 00:04:27	+26.27%
		an1hm4x5unrw7	select /*+ opt_param('_optimizer_use_auto_indexes'	00d 00:13:07.33	00d 00:10:24.50	00d 00:02:42	+20.68%
		57j6panxbfb66	SELECT [redacted]	00d 00:06:15.12	00d 00:05:31.57	00d 00:00:43	+11.61%
		25z6cubb844qr	SELECT [redacted]	00d 00:06:06.29	00d 00:05:28.83	00d 00:00:37	+10.22%
		dnrrqz7yr1xyj	SELECT [redacted]	00d 00:04:43.38	00d 00:04:16.29	00d 00:00:27	+09.56%
		0qwf3g87pzudn	select /*+ opt_param('_optimizer_use_auto_indexes'	00d 00:04:08.00	00d 00:03:04.00	00d 00:01:04	+25.81%
		f1vvh35vdxpn5	[redacted]	00d 00:03:46.88	00d 00:02:49.57	00d 00:00:57	+25.26%

Abakus
As na disku.

Top Events

Backup took much longer on EE...

Top Table (compare mode)

Activity	Activity [CMP]	Event	Duration	Duration [CMP]	Absolute Diff	Ratio
		Backup: MML create a backup piece	00d 00:05:03	10d 15:03:53	10d 14:58:50	-99.97%
		CSS initialization	04d 11:38:00	00d 07:16:23	04d 04:21:37	+93.24%
		ON CPU	02d 22:06:47	02d 14:39:57	00d 07:26:50	+10.62%
		Backup: MML write backup piece	02d 10:05:54	02d 09:15:24	00d 00:50:30	+01.45%
		db file sequential read	00d 22:19:25	00d 13:16:31	00d 09:02:54	+40.53%
		db file scattered read	00d 03:23:20	00d 01:33:37	00d 01:49:43	+53.96%
		log file parallel write	00d 03:16:52	00d 02:20:32	00d 00:56:20	+28.61%
		Disk file I/O Calibration	00d 03:05:07	00d 03:05:56	00d 00:00:49	-00.44%
		log file sync	00d 01:32:27	00d 01:22:29	00d 00:09:58	+10.78%
		db file parallel read	00d 01:17:17	00d 00:36:43	00d 00:40:34	+52.49%

… but general database performance is better on EE.

Average Duration of SQL

Standard Edition

***top queries of SE** compared to EE are now performing better.*

	Duration	Duration [CMP]	Absolute Diff	Ratio
[redacted]FROM ((S	00d 01:29:50.57	00d 01:11:39.43	00d 00:18:11	+20.24%
rall	00d 00:46:14.67	00d 00:32:29.67	00d 00:13:45	+29.73%
obj#, co	00d 00:21:08.12	00d 00:10:59.91	00d 00:10:08	+47.96%
uto_indexes'	00d 00:16:59.00	00d 00:12:31.33	00d 00:04:27	+26.27%
uto_indexes'	00d 00:13:07.33	00d 00:10:24.50	00d 00:02:42	+20.68%
[redacted]	00d 00:06:15.12	00d 00:05:31.57	00d 00:00:43	+11.61%

Enterprise Edition

Average Duration of SQL

Standard Edition

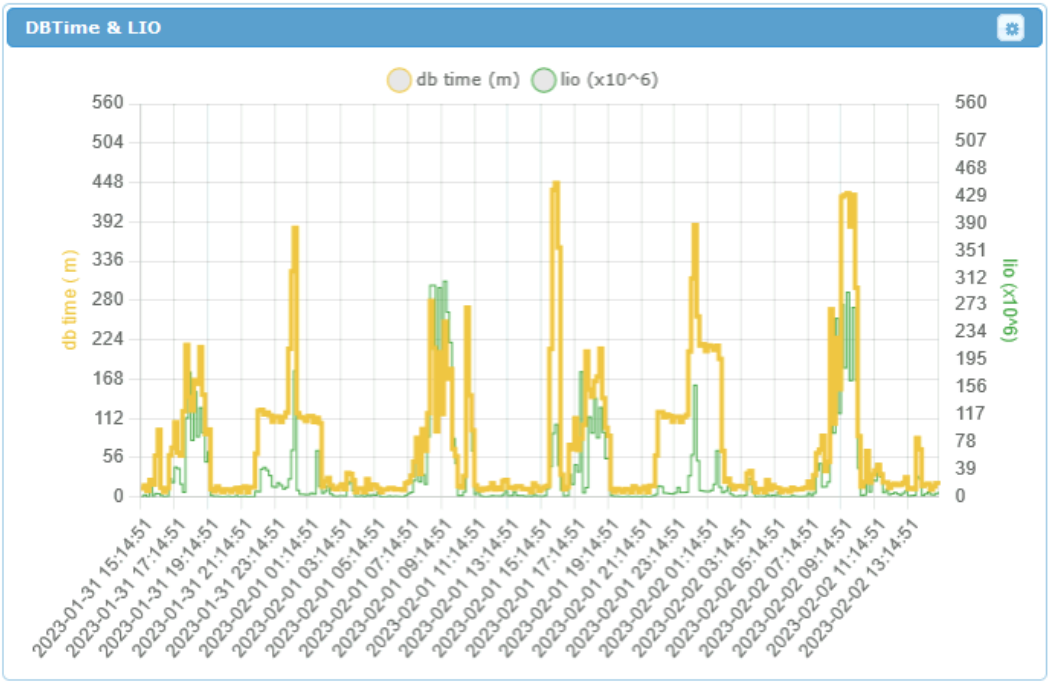
***But not all of them**, some were faster on SE:*

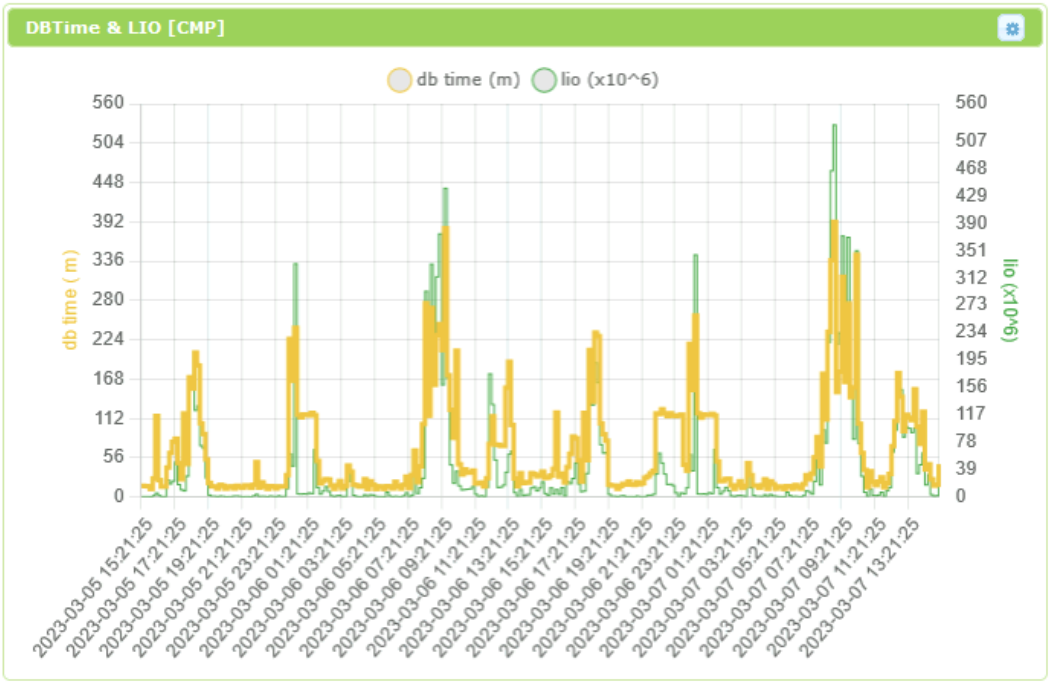
	00d 00:00:42.00	00d 00:01:55.00	00d 00:01:13	-63.48%
	00d 00:01:26.00	00d 00:01:49.00	00d 00:00:23	-21.10%
	00d 00:00:04.50	00d 00:01:43.00	00d 00:01:38	-95.63%

Enterprise Edition

10fmp0zmppqw7	/* SQL Analyze(0) */ select /*+ full(t) no_par
4rxchsm8s4f73	/* SQL Analyze(0) */ select /*+ full(t) no_par
5fcf5pmuaj2t7	SELECT [redacted]

DBTime & LIO



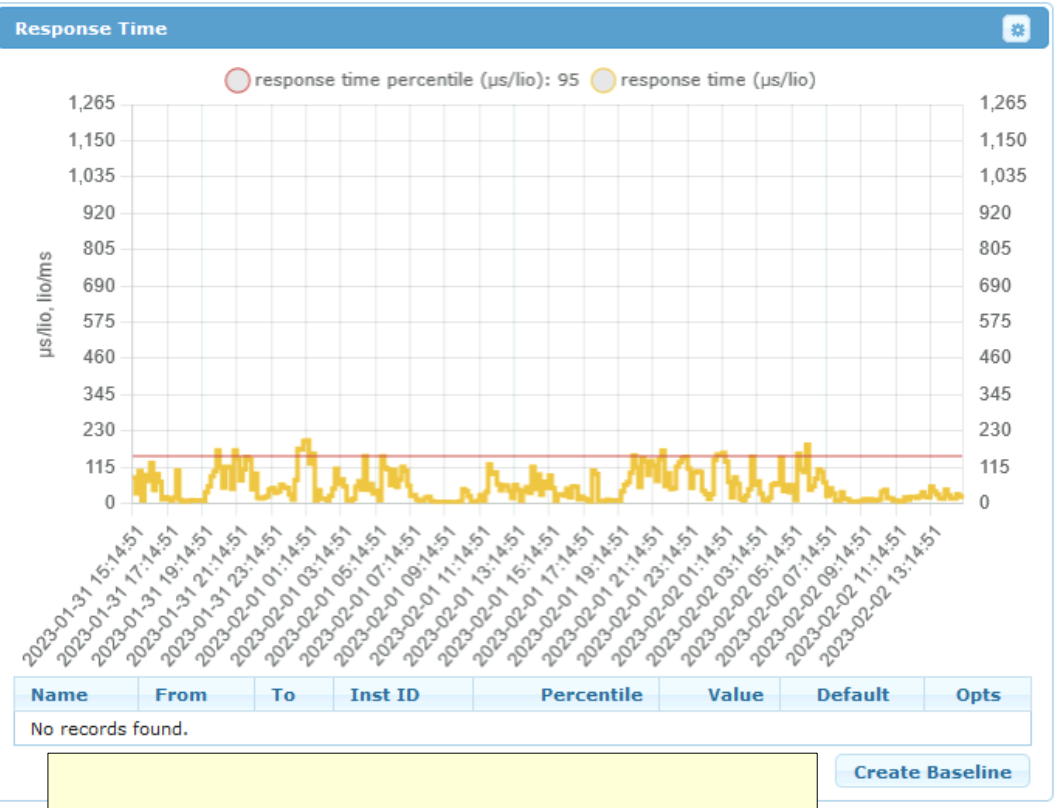


Amount of work (LIO) *done by the database is roughly the same in both cases. As is* ***amount of time (DBTime)*** *it took to do this work.*

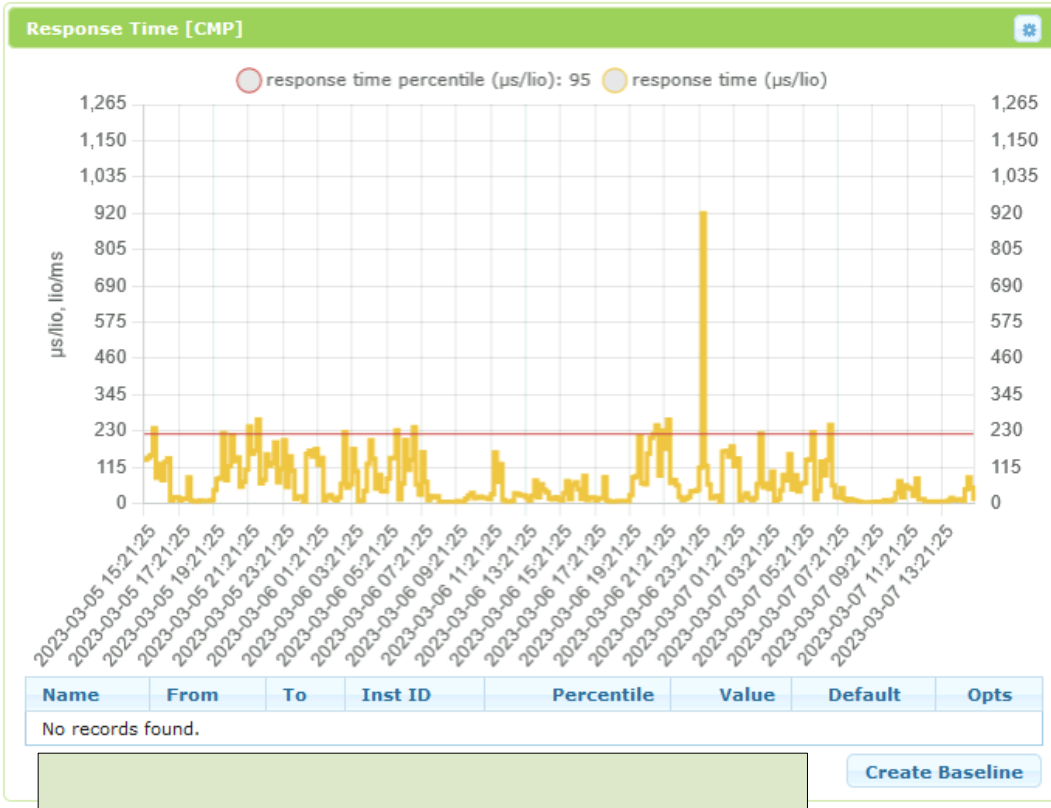
Response Time

Response time is thus also in the same ballpark in both cases because:

response_time = dbtime / lio



Standard Edition



Enterprise Edition

Wait, what?

If both (SE and EE) made the same amount of work (LIO) in the same time (dbtime)…

How can most »expensive« queries still be performing better on EE?!

Work and Time

LIO

1 worker 12 wheelbarrows in **60 min**

dbtime

Work and Time

LIO

2 workers 24 wheelbarrows in **60 min**

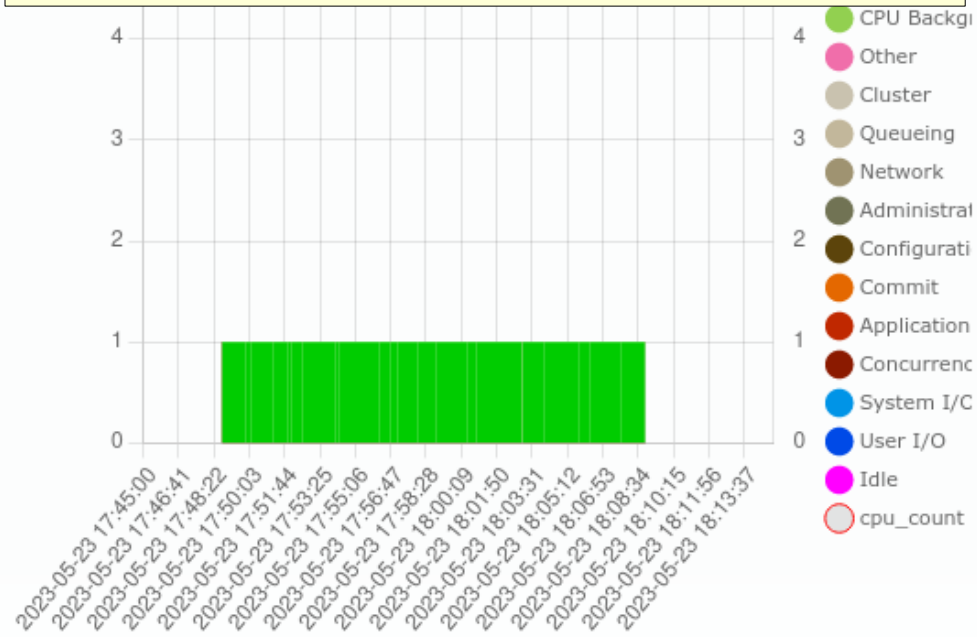
dbtime=120 min

Parallel Example 1)

```
create or replace function f_calc(p_input number)
return number PARALLEL_ENABLE as
    l_i number := 10000;
begin
    while l_i > 0
    loop
        l_i := l_i - 1;
    end loop;
    return p_input;
end;
/

select f_calc(col_a) from sample_tab;
-- sample_tab is a table with 1 mio rows
```

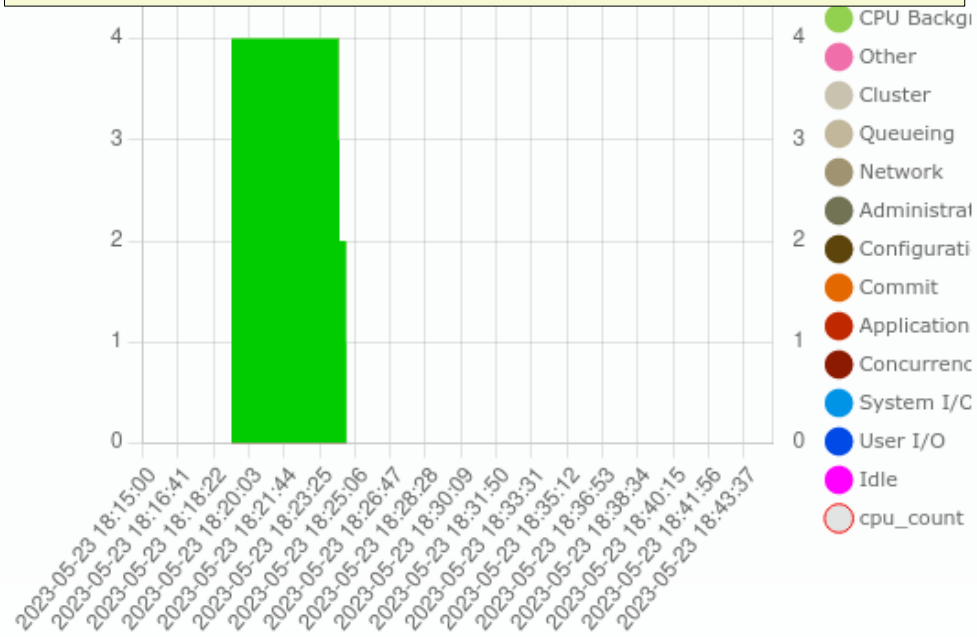
ASH | LIO | Undo | PGA | Temp

```
/* noparallel(t) full(t) */

~ 20 minutes
```



ASH [CMP] | LIO | Undo [CMP] | PGA [CMP] | Temp [CMP]

```
/* parallel(t,4) full(t) */

~ 5 minutes
```



Top Table (compare mode)

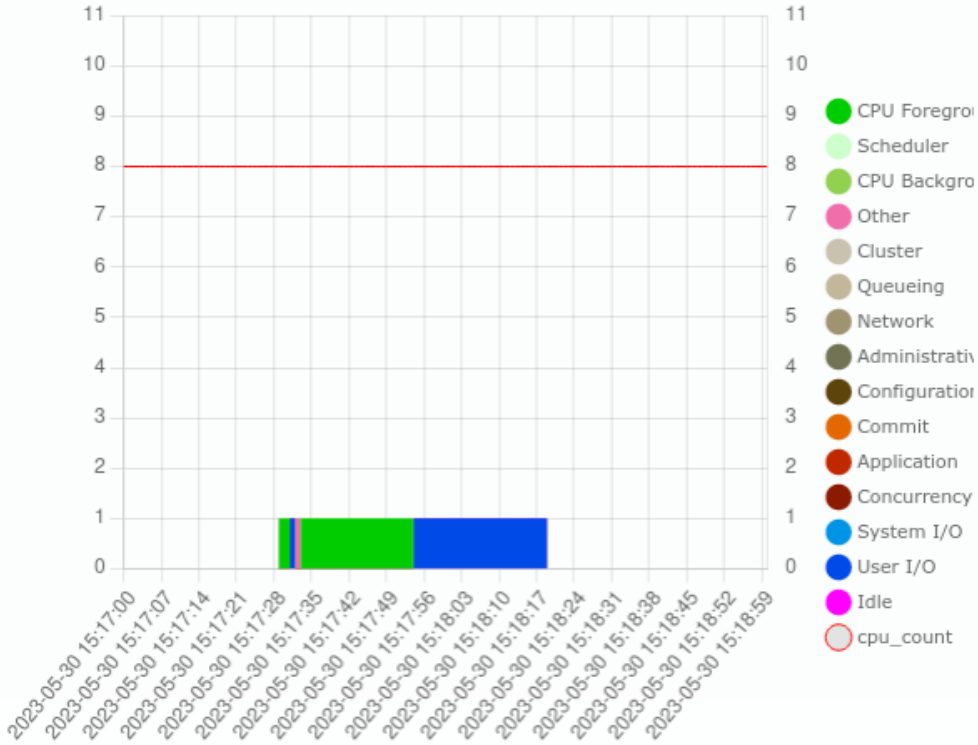
Activity	Activity [CMP]	SQL ID	SQL Text	Duration	Duration [CMP]	Absolute Diff	Ratio
		d01k06wt1t22k	[redacted]		00d 00:21:09		00.00%
		5u8kxjs9r591h	[redacted]	00d 00:19:43			00.00%

Parallel Example 2)

```
insert into test_dst
    select * from test_src;


-- test_src is a table with 500 mio rows
-- test_dst is an empty table without
   constraints or indexes
```

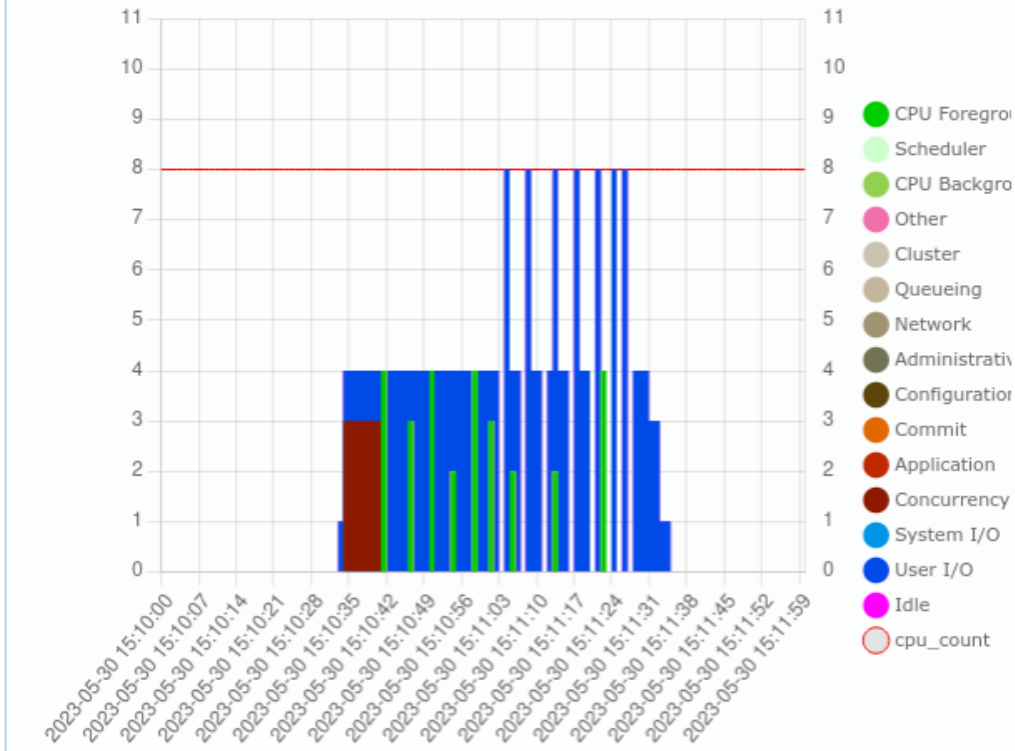
/*+ noparallel */

~ 0 min 50 sec



/*+ parallel(4) */

~ 0 min 58 sec



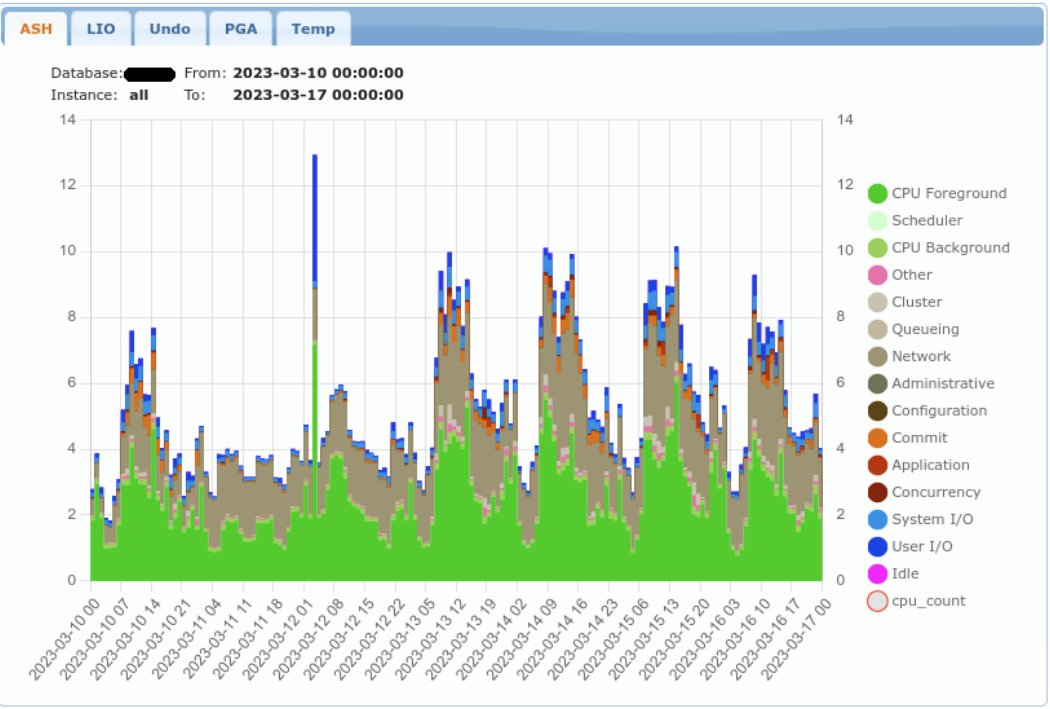
Top Table (compare mode)

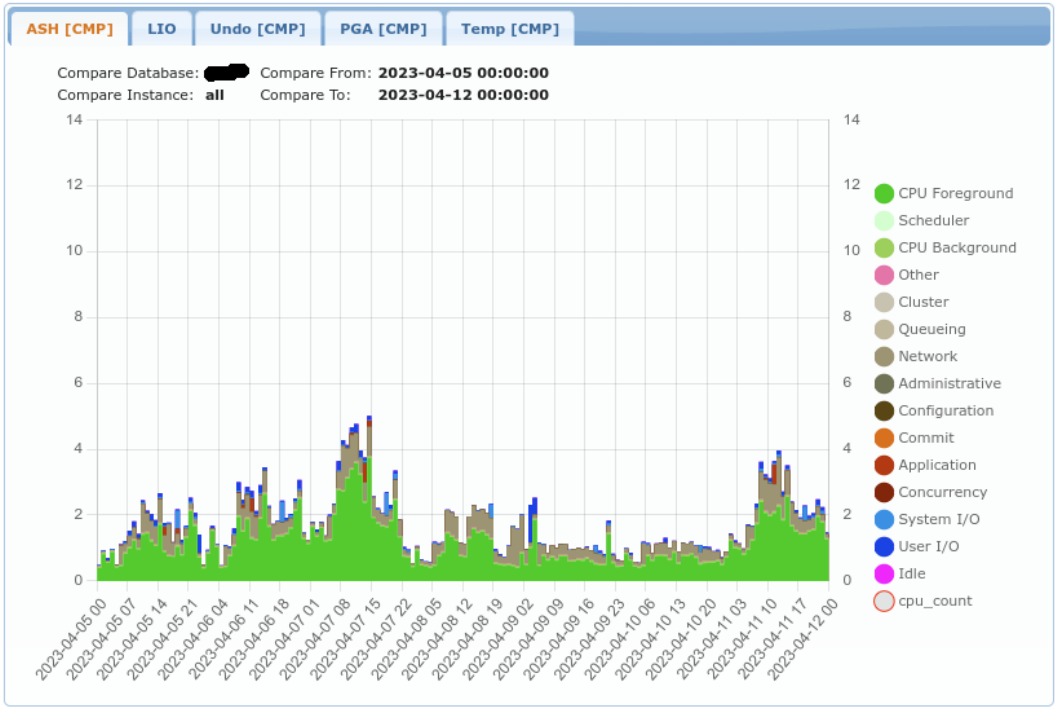
Activity	Activity [CMP]	SQL_ID (per exec)	SQL Text	Duration	Duration [CMP]	Absolute Diff	Ratio
		53ajjvrtwhyu9	insert /*+ enable_parallel_dml parallel(4) append		00d 00:01:56.50		00.00%
		920xjavz9xw67	insert /*+ noparallel append */ into test_tab2 d	00d 00:00:50.00			00.00%

Migration 2: Hardware Change

- Database Software stays the same (edition, version, parameters)
- New physical machine for VM hypervisor (faster CPU)
- New NVMEoF (all flash) storage (faster I/O)
- New Hypervisor (OVM -> OLVM)

→ NVMEoF + OLVM (db1)

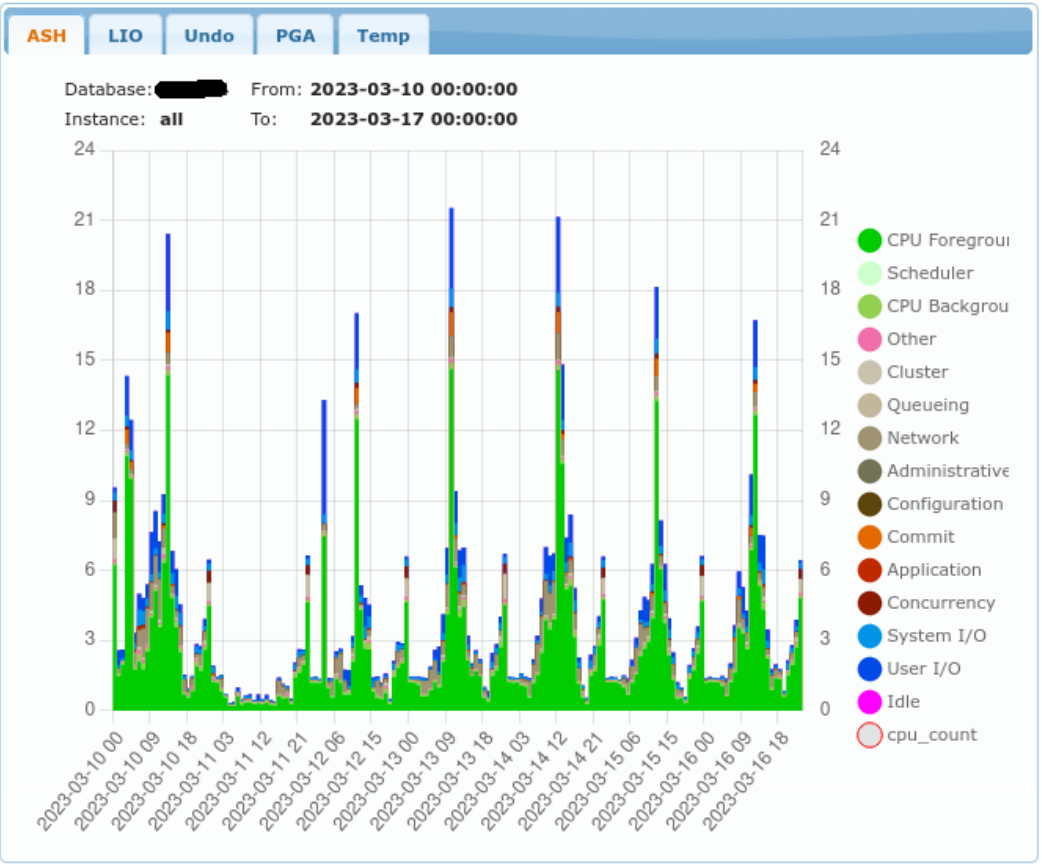


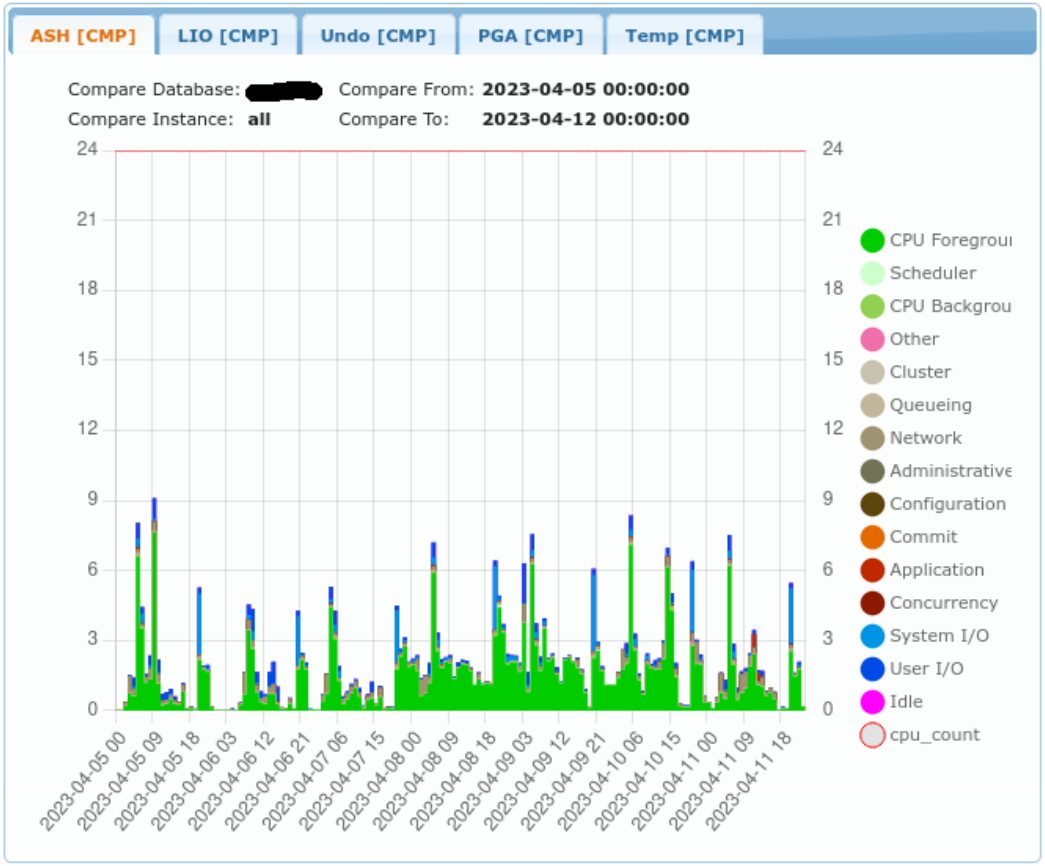


Top Table (compare mode)

Activity	Activity [CMP]	SQL ID	SQL Text	Duration	Duration [CMP]	Absolute Diff	Ratio
		65fdh8g6qb7zr	SELECT /*+ FULL(P) +*/ * FROM [redacted]	05d 23:58:36	00d 00:05:35	05d 23:53:01	+99.94%
		b4mk89z00j23g	SELECT * FROM [redacted]	01d 22:49:58	00d 23:48:41	00d 23:01:17	+49.16%
		12ru9wmf84jmw	BEGIN [redacted]	01d 14:39:48	00d 22:00:18	00d 16:39:30	+43.09%
		4hy9z3ta817fv	SELECT MAX([redacted]) FROM [redacted]	01d 11:48:30	00d 16:56:20	00d 18:52:10	+52.70%
		9bdhy2q1fujmu	INSERT INTO GTT$[redacted]	00d 17:42:54	00d 00:27:13	00d 17:15:41	+97.44%
		8qcbx3km55mkx	SELECT [redacted]	00d 17:05:17	00d 06:36:28	00d 10:28:49	+61.33%
		q185aqp8ssd0b	SELECT * FROM [redacted]	00d 15:44:48	00d 09:26:32	00d 06:18:16	+40.04%
		149zrk14ycwpk	UPDATE GTT$[redacted]	00d 15:40:28	00d 06:15:11	00d 09:25:17	+60.11%
		4xw97c4604u9s	SELECT COUNT(*) FROM [redacted]	00d 14:28:34	00d 07:07:50	00d 07:20:44	+50.74%
		1t9us5ng9w0jd	SELECT COUNT(*) FROM [redacted]	00d 12:20:50	00d 00:33:11	00d 11:47:39	+95.52%

→ NVMEoF + OLVM (db2)

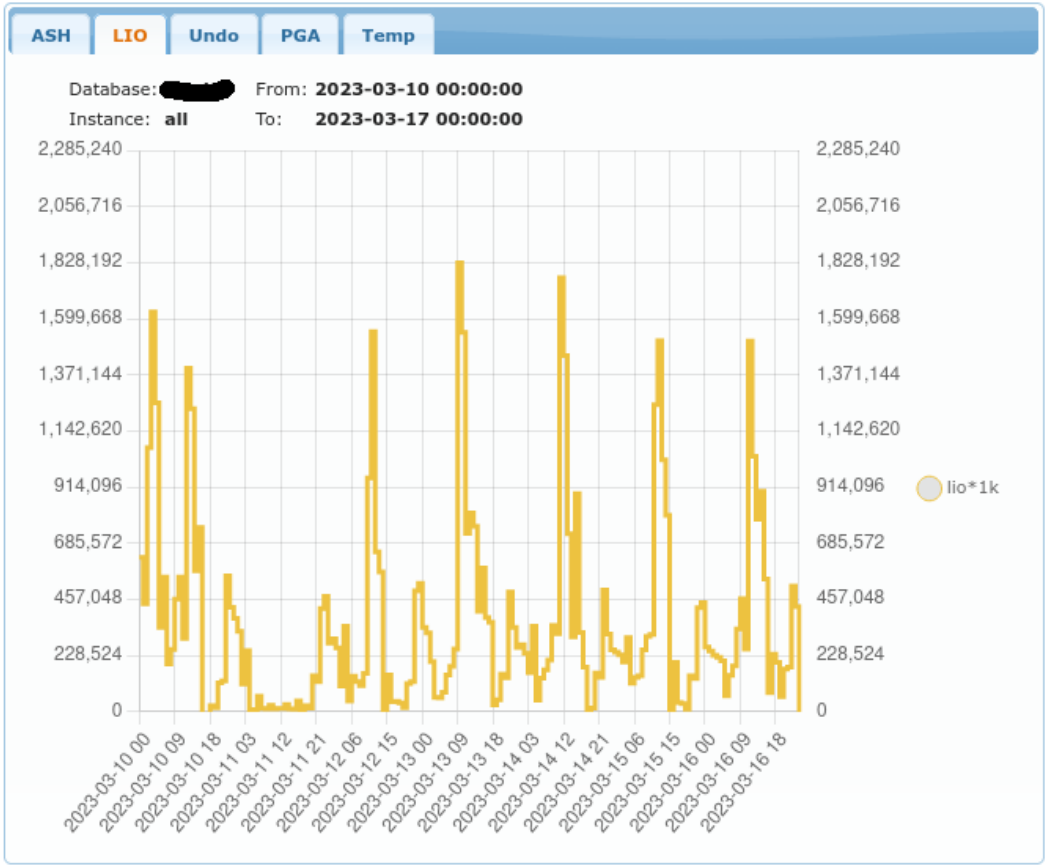


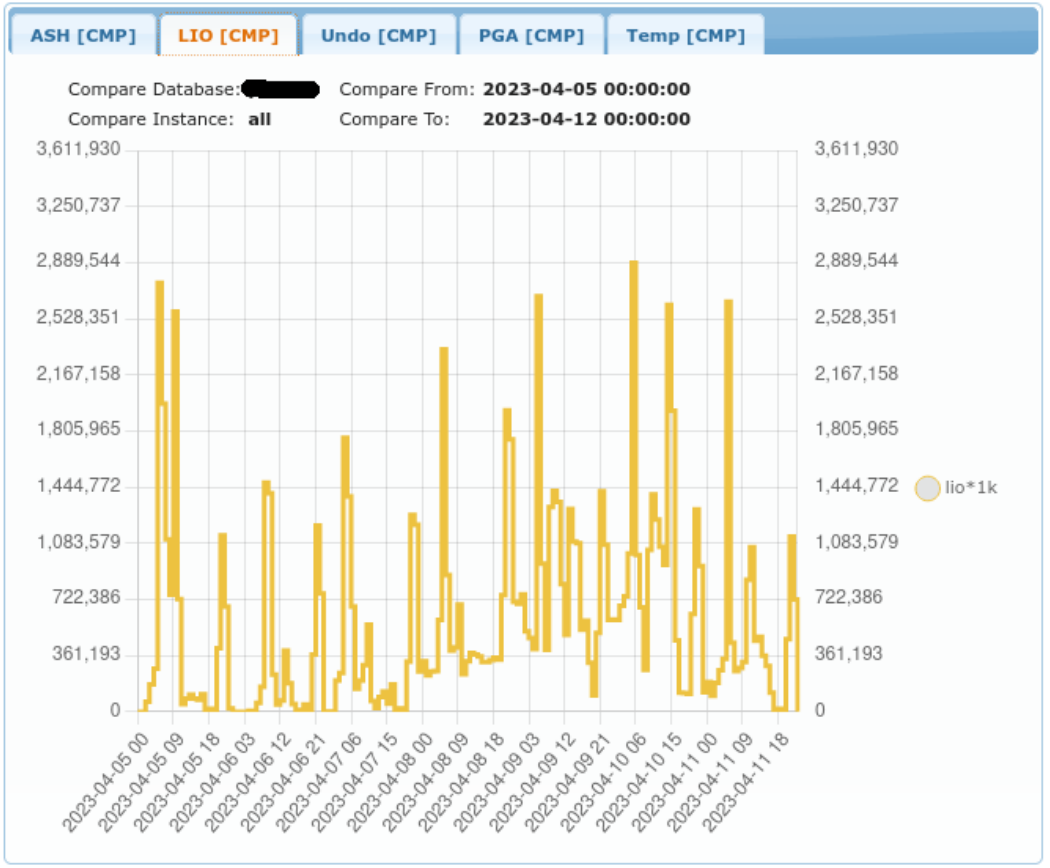


Top Table (compare mode)

Activity	Activity [CMP]	SQL_ID (per exec)	SQL Text	Duration	Duration [CMP]	Absolute Diff	Ratio
		1btcwsy51t6jm	DELETE FROM [redacted]	00d 01:03:52.33	00d 00:22:29.71	00d 00:41:22	+64.78%
		6m0kznaux1j3j	INSERT INTO [redacted]	00d 00:57:56.71	00d 00:40:21.57	00d 00:17:35	+30.35%
		3tbndmsxkum50	SELECT DISTINCT [redacted]	00d 00:41:52.00	00d 00:09:10.29	00d 00:32:41	+78.09%
		94smfxbduu7js	INSERT INTO [redacted]	00d 00:39:42.75	00d 00:21:33.11	00d 00:18:09	+45.73%
		artj44qzb2n8y	INSERT INTO [redacted]	00d 00:26:10.29	00d 00:10:01.12	00d 00:16:09	+61.72%
		6986mcjqsnugj	INSERT INTO [redacted]	00d 00:25:39.88	00d 00:08:27.38	00d 00:17:12	+67.05%
		1xrjx70vvpub6	with [redacted]	00d 00:25:10.29	00d 00:10:08.12	00d 00:15:02	+59.73%
		5xv0chtkjpu00	INSERT INTO [redacted]	00d 00:24:56.67	00d 00:08:46.57	00d 00:16:10	+64.82%
		fp3vs6bdgugfa	INSERT INTO [redacted]	00d 00:24:37.43	00d 00:11:39.00	00d 00:12:58	+52.69%
		b14sr2p26m586	WITH [redacted]	00d 00:20:46.00	00d 00:03:29.00	00d 00:17:17	+83.23%

→ NVMEoF + OLVM (db2)



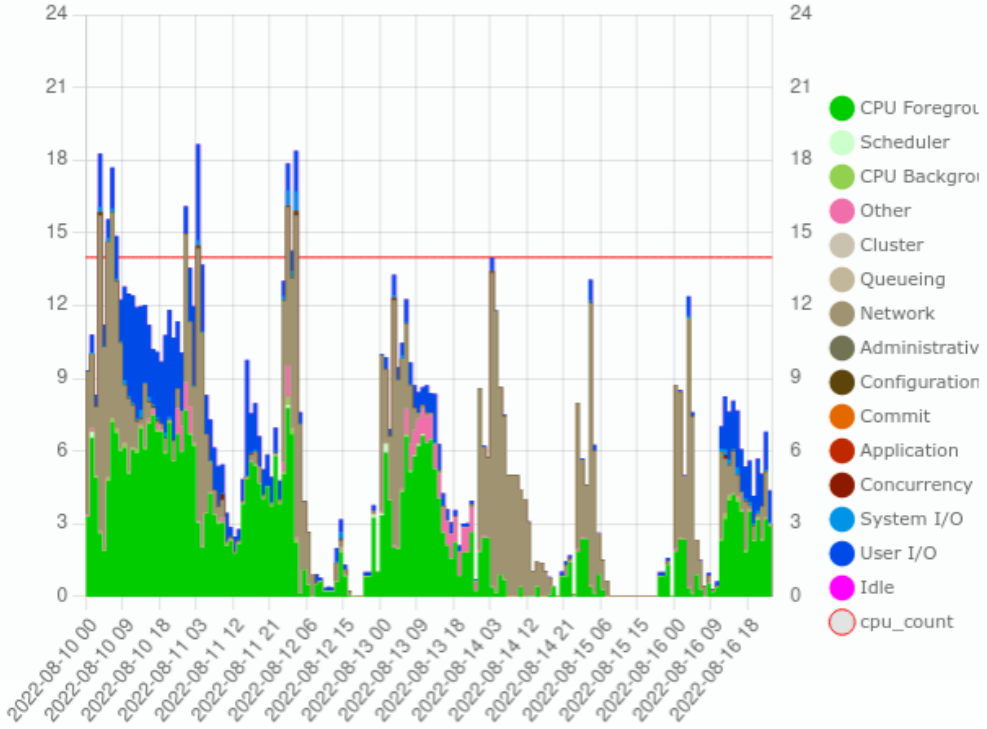


Migration 3: Exadata → Server

- Move from Exadata to Abakus Oracle server.

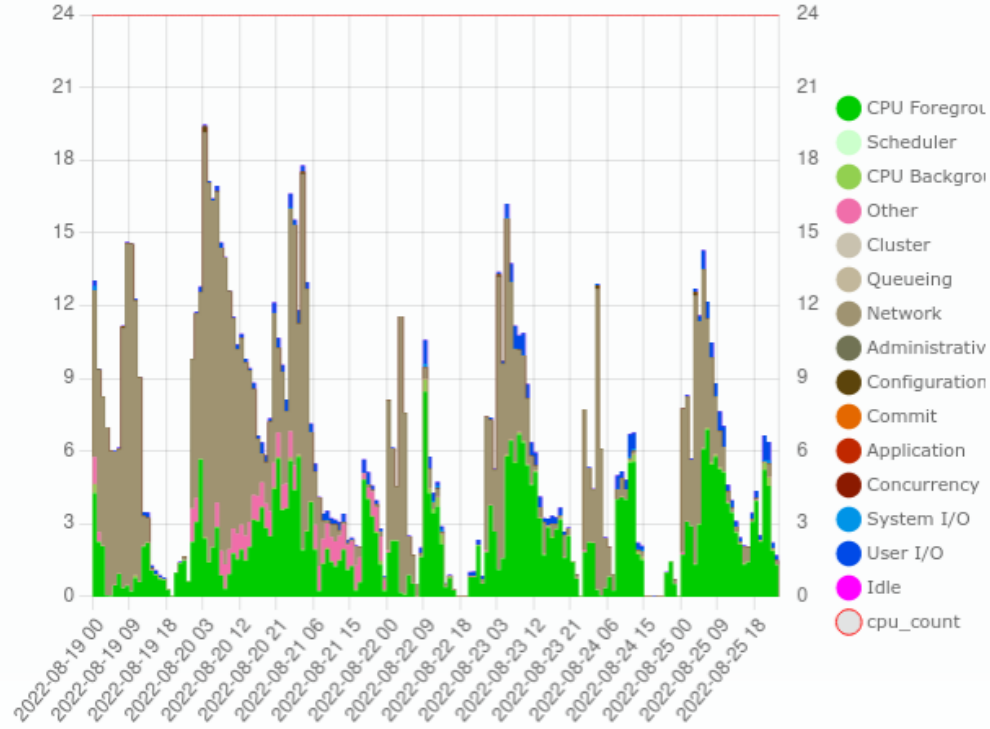
ASH | LIO | Undo | PGA | Temp

Database: [redacted] From: **2022-08-10 00:00:00**
Instance: **all** To: **2022-08-17 00:00:00**



ASH [CMP] | LIO [CMP] | Undo [CMP] | PGA [CMP] | Temp [CMP]

Compare Database: [redacted] Compare From: **2022-08-19 00:00:00**
Compare Instance: **all** Compare To: **2022-08-26 00:00:00**

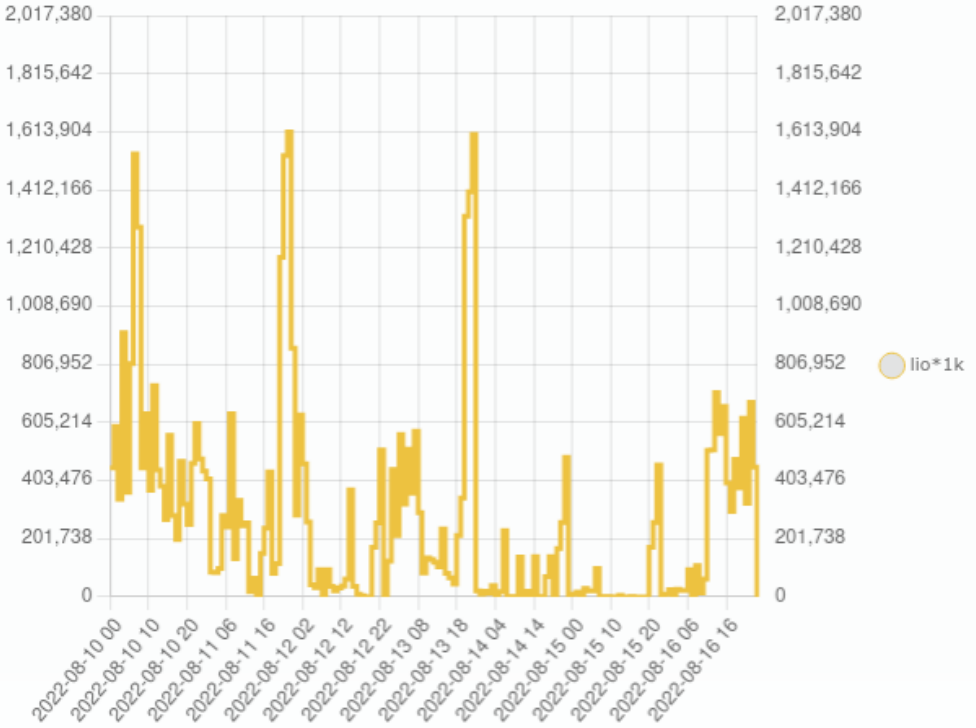


Top Table (compare mode)

Activity	Activity [CMP]	SQL_ID (per exec)	SQL Text	Duration	Duration [CMP]	Absolute Diff	Ratio
		2cfhrhgf5b1bs	call [redacted]	00d 18:57:34.00	00d 12:13:20.17	00d 06:44:13	+35.53%
		dyfzdswzgtgmq	BEGIN [redacted]remote[redacted]	00d 12:19:51.00	00d 11:30:54.00	00d 00:48:57	+06.62%
		7vh98x578a288	BEGIN [redacted]_remote[redacted]	00d 06:35:35.00	00d 05:39:29.00	00d 00:56:06	+14.18%
		1z26cvgm4pvdk	BEGIN [redacted]remote[redacted]	00d 05:02:19.00	00d 05:48:09.57	00d 00:45:50	-13.17%
		59h530haayxa0	BEGIN [redacted]remote[redacted]	00d 04:25:43.83	00d 03:37:54.60	00d 00:47:49	+18.00%
		5hh0djakv8g6t	BEGIN [redacted]remote[redacted]	00d 04:18:18.00	00d 06:08:05.00	00d 01:49:47	-29.83%
		dzg8bvzt48mrc	BEGIN [redacted]_remote[redacted]	00d 03:52:09.00	00d 10:18:31.00	00d 06:26:22	-62.47%
		4vz3dh8087pkz	BEGIN [redacted]remote[redacted]	00d 03:29:08.00	00d 04:58:23.00	00d 01:29:15	-29.91%
		d12mhqfrhn8as	BEGIN[redacted]_remote[redacted]	00d 03:10:46.00	00d 07:13:26.00	00d 04:02:40	-55.99%
		gxtxsfbg6h37b	BEGIN [redacted]_remote[redacted]	00d 03:02:04.00	00d 02:55:20.00	00d 00:06:44	+03.70%

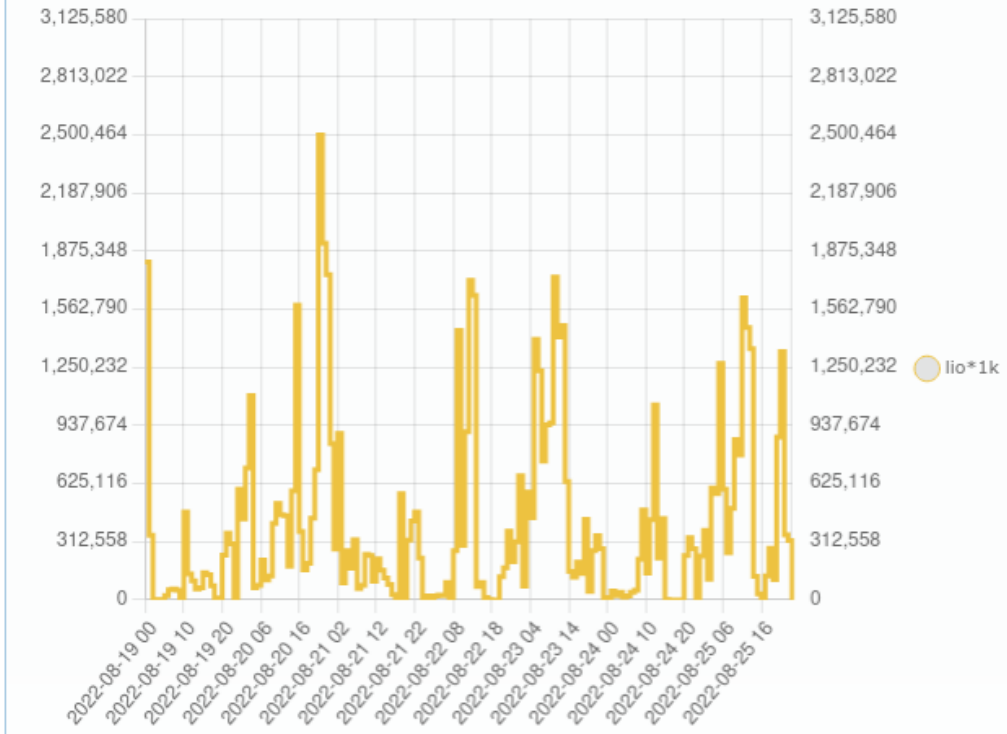
ASH | LIO | Undo | PGA | Temp

Database: [redacted] From: **2022-08-10 00:00:00**
Instance: **all** To: **2022-08-17 00:00:00**



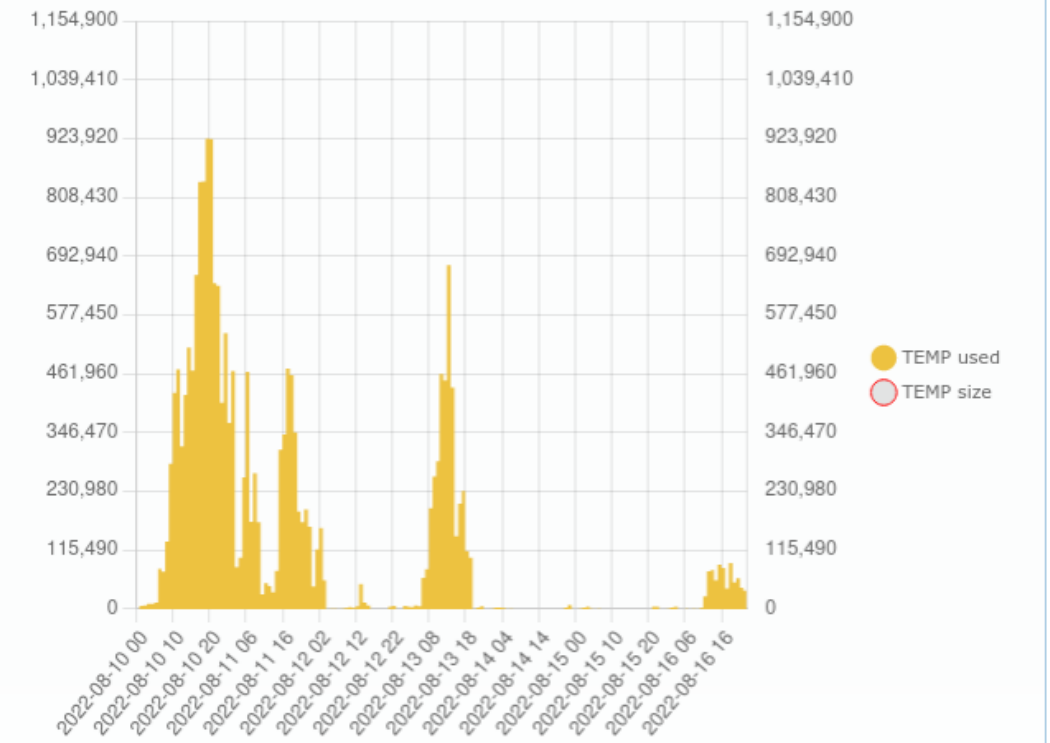
ASH [CMP] | LIO [CMP] | Undo [CMP] | PGA [CMP] | Temp [CMP]

Compare Database: [redacted] Compare From: **2022-08-19 00:00:00**
Compare Instance: **all** Compare To: **2022-08-26 00:00:00**



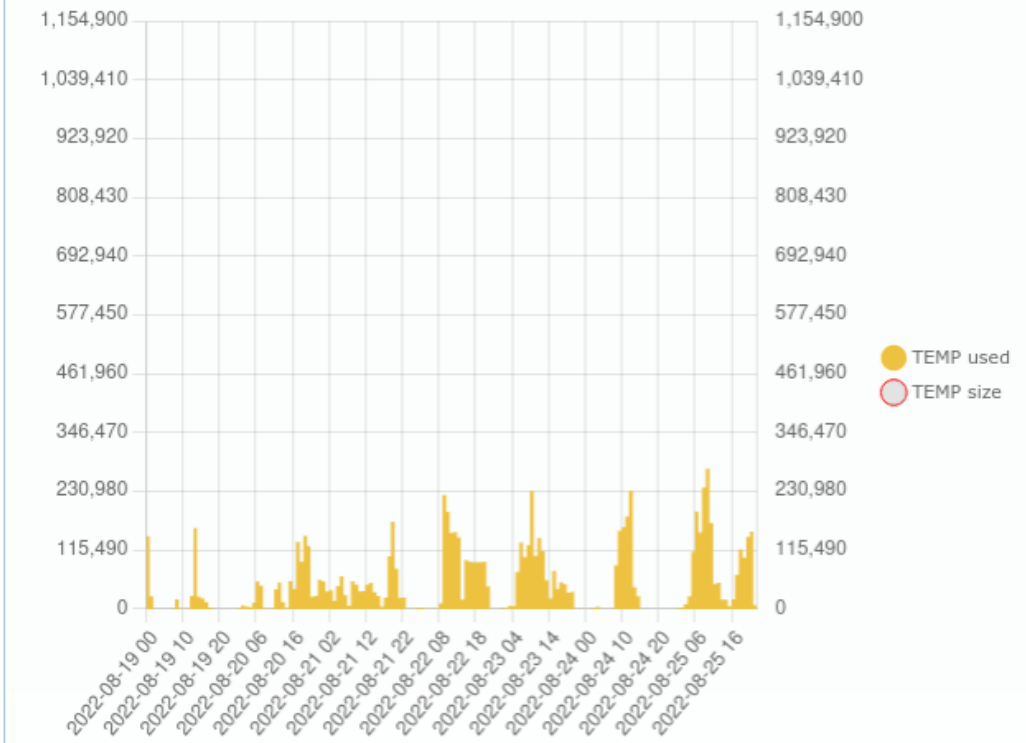
ASH | LIO | Undo | PGA | **Temp**

Database: [redacted] From: **2022-08-10 00:00:00**
Instance: **all** To: **2022-08-17 00:00:00**

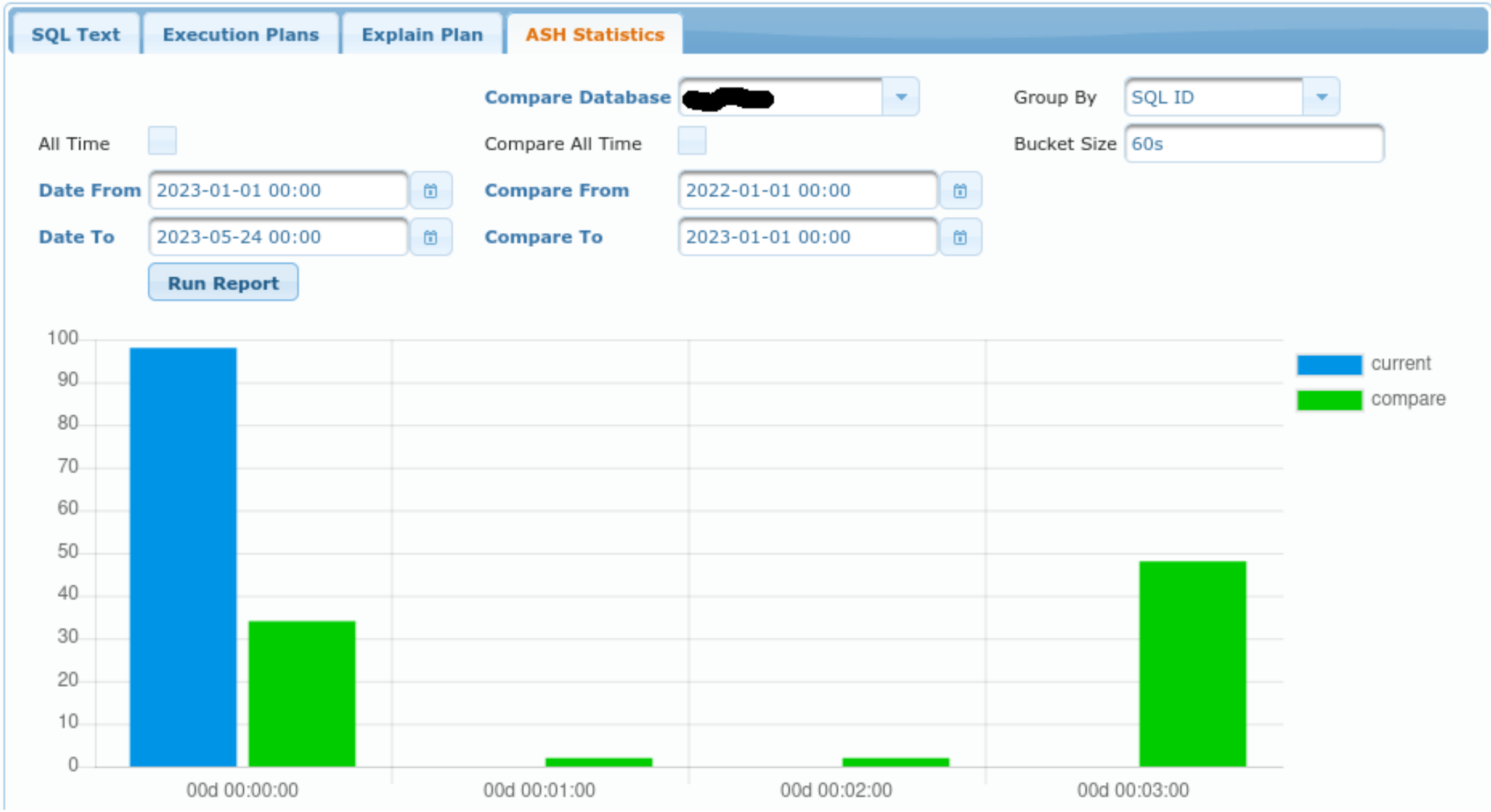


ASH [CMP] | LIO [CMP] | Undo [CMP] | PGA [CMP] | **Temp [CMP]**

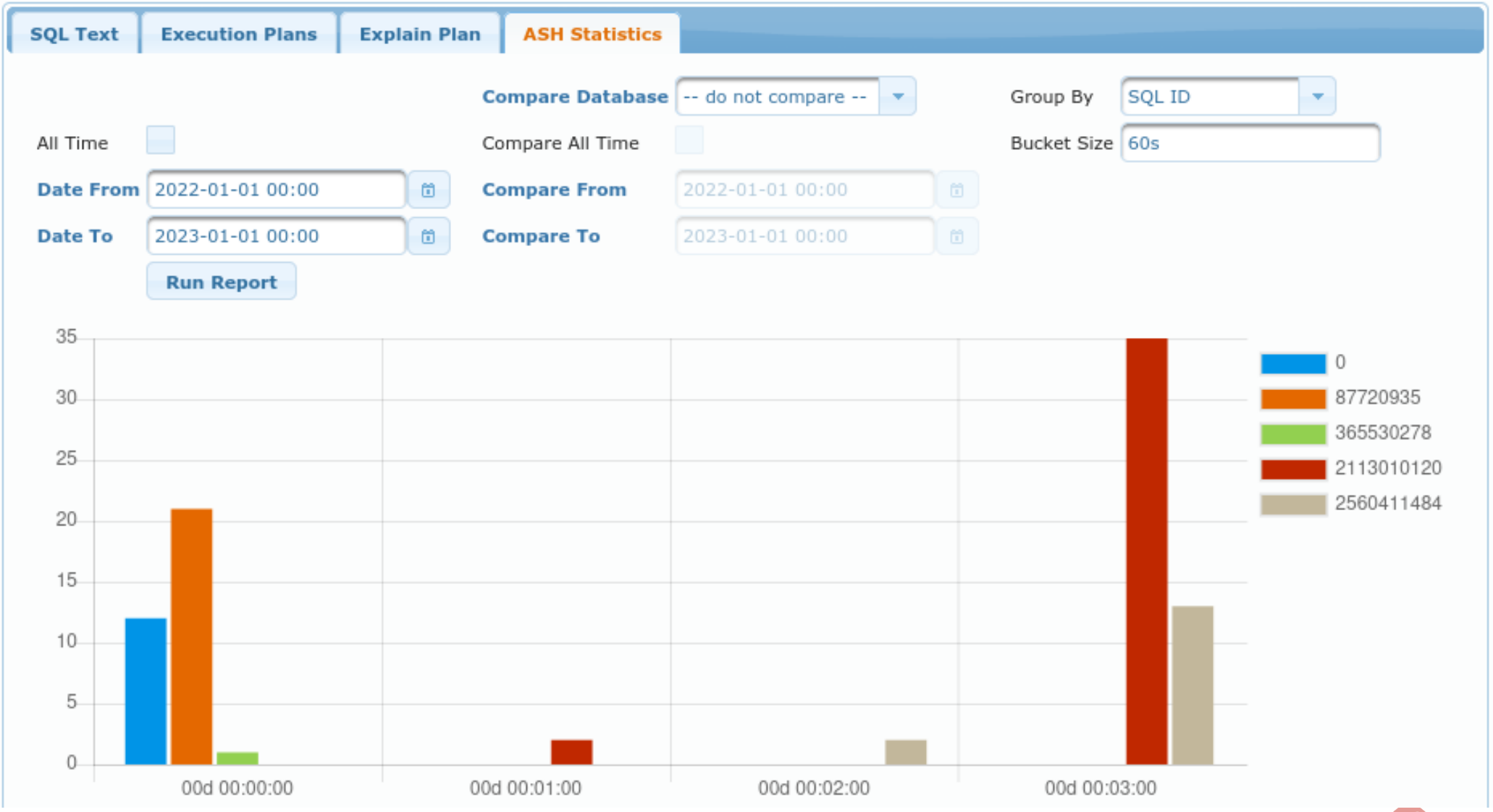
Compare Database: [redacted] Compare From: **2022-08-19 00:00:00**
Compare Instance: **all** Compare To: **2022-08-26 00:00:00**



Execution of SQL over time



The Change of PLAN!



… to be continued

Kako ukrotiti pobegle SQL stavke po nadgradnji baze, aplikacije ali hardvera?

11:00 - 11:45

Boris Oblak Abakus plus d.o.o.

DBA

Dvorana: **A**

http://www.abakus.si/

AS
Abakus
As na disku.