Capture Time: **2024-09-26T08:51:36 to 2024-09-26T16:51:36** UTC

RDS instance: enterworks-engineering11-production-dbpartitioning (non-partition table)

A screenshot of a computer

Description automatically generated

CPUUsage and Read IO/Memory (PAGEIOLATCH\_SH) are the major bottleneck.

**Top query 1 ( Performance Insights PI)**

Execution: 100

Total elapsed time(ms): 27475

Physical reads: 19566.00

CPU time(ms): 453.61

Logical Read/sec: 7.74

SELECT REPOSITORY\_ITEM\_HISTORY\_ID, MASTER\_REPOSITORY\_ID, ITEM\_ID, ATTR\_DATA, MODIFICATION\_DATETIME, MODIFIED\_BY, MODIFY\_ACTION, USER\_LOGIN, RECLOCK, RECORD\_STATE, PRODUCTION\_STATE, WORKFLOW\_STATE, MESSAGE\_ID, TRANSACTION\_ID, STATE\_UPDATE\_TIME, STATE\_UPDATE\_MSG, EXTERNAL\_SESSION\_INFO, ENTRY\_DATETIME, STAGING\_ITEM\_ID, OPERATION\_USER, OPERATION\_JOB\_ID, VALIDATION\_LEVEL\_IND, SECURITY\_CONTEXT\_VALUE, MERGED\_INTO\_ITEM\_ID

from B\_REPOSITORY\_ITEM\_HISTORY brih WHERE ITEM\_ID = 12937091 and MODIFY\_ACTION = 'create'

-- **Missing index on MODIFY\_ACTION**

**Top query 2 (PI)**

Execution: 22639119

Total elapsed time(ms): 4500561

Physical reads: 2281387

CPU time(ms): 920110

Logical Read/sec: 1578.08

SELECT [bmi].[ITEM\_ID],[REPOSITORY\_ID],[HAS\_ERROR\_IND],[SYNC\_ACTION],[SYNC\_ACTION\_DELETE],[IS\_DUPLICATE],[RECORD\_STATE],[PRODUCTION\_STATE],[WORKFLOW\_STATE],[MESSAGE\_ID],[TRANSACTION\_ID],[STATE\_UPDATE\_TIME],[STATE\_UPDATE\_MSG],[PK\_COL\_1],[PK\_COL\_2],[PK\_COL\_3],[PK\_COL\_4],[PK\_COL\_5],

[ATTR\_DATA],[bmi].[CREATION\_DATETIME],[CREATED\_BY],[bmi].[LAST\_UPDATE\_DATETIME],[bmi].[LAST\_UPDATE\_BY],[MODIFY\_ACTION],[ATTR\_LAST\_UPDATE\_DATETIME],[ATTR\_LAST\_UPDATE\_BY],[EXTERNAL\_SESSION\_INFO],[PLT\_ITEM\_ID],[STAGING\_ITEM\_ID],[GLOBAL\_IND],[VALIDATION\_LEVEL\_IND],[OVERALL\_ERROR\_IND],[HAS\_ERROR\_IND\_1],[HAS\_ERROR\_IND\_2],[HAS\_ERROR\_IND\_3],[HAS\_ERROR\_IND\_4],[HAS\_ERROR\_IND\_5],[SECURITY\_CONTEXT\_VALUE],[MERGED\_INTO\_ITEM\_ID]

FROM [B\_MASTER\_REPOSITORY\_ITEM] [bmi] WHERE [ITEM\_ID]=@1 AND [REPOSITORY\_ID]=@2

**Top query 3 (PI)**

Execution: 255792

Total elapsed time(ms): 1636137

Physical reads: 3458

CPU time(ms): 1551272

Logical Read/sec: 27154

SELECT s.ITEM\_ID from B\_SNAPSHOT\_10142 s WHERE s.F\_1004814 = ? AND s.F\_1004810 = ? AND s.F\_1004811 = ? AND s.F\_1004812 = ? AND s.F\_1004813 = ?

**Missing Index: CREATE INDEX [IDX\_B\_SNAPSHOT\_10142\_F\_4814\_4810\_4811\_4812\_4813] ON [dbo].[B\_SNAPSHOT\_10142] ([F\_1004810],[F\_1004811],[F\_1004814])INCLUDE ([F\_1004813], F\_1004812);**

**Top query 4 (PI)**

Execution: 24445

Total elapsed time(ms): 621345

Physical reads: 0

CPU time(ms): 588016

Logical Read/sec: 14432

select bjob0\_.JOB\_ID as job\_id1\_58\_, bjob0\_.MASTER\_REPOSITORY\_ID as master\_r2\_58\_, bjob0\_.VIEW\_ID as view\_id3\_58\_, bjob0\_.JOB\_ABORT\_ID as job\_abor4\_58\_, bjob0\_.AUTO\_INTERVAL as auto\_int5\_58\_, bjob0\_.AUTO\_SEED\_TIME as auto\_see6\_58\_, bjob0\_.JOB\_TYPE\_CODE as job\_type7\_58\_, bjob0\_.RECLOCK as reclock8\_58\_, bjob0\_.NEXT\_EXECUTION\_TIME as next\_exe9\_58\_, bjob0\_.STATUS as status10\_58\_, bjob0\_.AUTO\_EXECUTE\_IND as auto\_ex11\_58\_, bjob0\_.TRANSMIT\_FILE\_IND as transmi12\_58\_, bjob0\_.CUST\_REPOSITORY\_LIST as cust\_re13\_58\_, bjob0\_.ERROR\_RETRY\_COUNTER as error\_r14\_58\_, bjob0\_.USER\_ID as user\_id15\_58\_, bjob0\_.ADDITIONAL\_XML\_DATA as additio16\_58\_, bjob0\_.START\_TIME as start\_t17\_58\_, bjob0\_.APPSERVER\_JNDI as appserv18\_58\_, bjob0\_.BATCH\_IND as batch\_i19\_58\_, bjob0\_.PRIORITY as priorit20\_58\_, bjob0\_.DEPENDENT\_JOBS as depende21\_58\_

from B\_JOB bjob0\_ where bjob0\_.NEXT\_EXECUTION\_TIME<= @P0 and upper(bjob0\_.APPSERVER\_JNDI)= @P1

order by bjob0\_.PRIORITY, bjob0\_.NEXT\_EXECUTION\_TIME

**No index on PRIORITY,** UPPER() very costy, it is case insensitive DB/column why use UPPER? Requires Application Code change to get rid of UPPER()? Hopefully SQL optimizer will take care of it.

**Top query 5 (PI)**

Execution: 11125856

Total elapsed time(ms): 1801355

Physical reads: 51178

CPU time(ms): 1620001

Logical Read/sec: 12663

UPDATE B\_MASTER\_REPOSITORY\_ITEM SET REPOSITORY\_ID = @P0 , HAS\_ERROR\_IND = @P1 , SYNC\_ACTION = @P2 , SYNC\_ACTION\_DELETE = @P3 , IS\_DUPLICATE = @P4 , RECORD\_STATE = @P5 , PRODUCTION\_STATE = @P6 , WORKFLOW\_STATE = @P7 , MESSAGE\_ID = @P8 , TRANSACTION\_ID = @P9 , STATE\_UPDATE\_TIME = @P10 , STATE\_UPDATE\_MSG = @P11 , PK\_COL\_1 = @P12 , PK\_COL\_2 = @P13 , PK\_COL\_3 = @P14 , PK\_COL\_4 = @P15 , PK\_COL\_5 = @P16 , ATTR\_DATA = @P17 , LAST\_UPDATE\_DATETIME = @P18 , LAST\_UPDATE\_BY = @P19 , MODIFY\_ACTION = @P20 , ATTR\_LAST\_UPDATE\_DATETIME = @P21 , ATTR\_LAST\_UPDATE\_BY = @P22 , EXTERNAL\_SESSION\_INFO = @P23 , PLT\_ITEM\_ID = @P24 , STAGING\_ITEM\_ID= @P25 , GLOBAL\_IND= @P26 , VALIDATION\_LEVEL\_IND= @P27 , OVERALL\_ERROR\_IND= @P28 , HAS\_ERROR\_IND\_1= @P29 , HAS\_ERROR\_IND\_2= @P30 , HAS\_ERROR\_IND\_3= @P31 , HAS\_ERROR\_IND\_4= @P32 , HAS\_ERROR\_IND\_5= @P33 , SECURITY\_CONTEXT\_VALUE= @P34 , MERGED\_INTO\_ITEM\_ID= @P35 , CREATED\_BY= @P36

WHERE ITEM\_ID = @P37

**Disable Full-Text search**

**Top query 6 (PI)**

Execution: 1990502

Total elapsed time(ms): 1725113

Physical reads: 378994

CPU time(ms): 106085

Logical Read/sec: 307

MERGE B\_SNAPSHOT\_10143 as target using (SELECT @P0 as ITEM\_ID, getDate() as DLUD, @P1 as F\_1005642, @P2 as F\_1004837, @P3 as F\_1004839, @P4 as F\_1005965, @P5 as F\_1005964, @P6 as F\_1005966, @P7 as F\_1005967, @P8 as F\_1005955, @P9 as F\_1005963, @P10 as F\_1004840, @P11 as F\_1004835, @P12 as F\_1005643, @P13 as F\_1004833, @P14 as F\_1004842, @P15 as F\_1004832, @P16 as F\_1004834)

as source on source.ITEM\_ID = target.ITEM\_ID when matched then update set DATA\_LAST\_UPDATE\_DATETIME = source.DLUD,F\_1005642 = source.F\_1005642,F\_1004837 = source.F\_1004837,F\_1004839 = source.F\_1004839,F\_1005965 = source.F\_1005965,F\_1005964 = source.F\_1005964,F\_1005966 = source.F\_1005966,F\_1005967 = source.F\_1005967,F\_1005955 = source.F\_1005955,F\_1005963 = source.F\_1005963,F\_1004840 = source.F\_1004840,F\_1004835 = source.F\_1004835,F\_1005643 = source.F\_1005643,F\_1004833 = source.F\_1004833,

F\_1004842 = source.F\_1004842,F\_1004832 = source.F\_1004832,F\_1004834 = source.F\_1004834

when not matched then insert (ITEM\_ID,F\_1005642,F\_1004837,F\_1004839,F\_1005965,F\_1005964,F\_1005966,F\_1005967,F\_1005955,F\_1005963,F\_1004840,F\_1004835,F\_1005643,F\_1004833,F\_1004842,F\_1004832,F\_1004834,DATA\_LAST\_UPDATE\_DATETIME) values (source.ITEM\_ID,source.F\_1005642,source.F\_1004837,source.F\_1004839,source.F\_1005965,source.F\_1005964,source.F\_1005966,source.F\_1005967,source.F\_1005955,source.F\_1005963,source.F\_1004840,source.F\_1004835,source.F\_1005643,source.F\_1004833,source.F\_1004842,source.F\_1004832,source.F\_1004834,getDate());

**Merge operation**, Will it help by using option ROW LOCK?

**Top query 7 (PI)**

Execution: 75130

Total elapsed time(ms): 136295

Physical reads: 0

CPU time(ms): 129413

Logical Read/sec: 251

SELECT s.ITEM\_ID from B\_SNAPSHOT\_10149 s WHERE s.F\_1004904 = @P0 AND s.F\_1004905 = @P1 AND s.F\_1004906 = @P2

**Modify Index**: CREATE NONCLUSTERED INDEX [SNAP\_10149\_F\_1004904] ON [dbo].[B\_SNAPSHOT\_10149]

([F\_1004904]) INCLUDE (F\_1004906, F\_1004905)

OR CREATE

CREATE NONCLUSTERED INDEX [SNAP\_10149\_F\_1004904] ON [dbo].[B\_SNAPSHOT\_10149]

([F\_1004904], F\_1004906) INCLUDE (F\_1004905)

**Top query 8 (PI)**

Execution: 7514505

Total elapsed time(ms): 878582

Physical reads: 0

CPU time(ms): 876577

Logical Read/sec: 2609

select brulemappi0\_.RULE\_MAPPING\_ID as rule\_map1\_106\_, brulemappi0\_.RULE\_ID as rule\_id2\_106\_, brulemappi0\_.OBJECT\_TYPE as object\_t3\_106\_, brulemappi0\_.OBJECT\_ID as object\_i4\_106\_, brulemappi0\_.SEQUENCE\_NUM as sequence5\_106\_, brulemappi0\_.APPLIED\_IND as applied\_6\_106\_, brulemappi0\_.LANGUAGE\_EXT as language7\_106\_

from B\_RULE\_MAPPING brulemappi0\_

where brulemappi0\_.RULE\_ID= @P0 and brulemappi0\_.OBJECT\_TYPE= @P1 order by brulemappi0\_.SEQUENCE\_NUM

**-- Missing index** on OBJECT\_TYPE and SEQUENCE\_NUM

-- Not sure if composite index will be better

**Top query 9 (PI)**

Execution: 8470163

Total elapsed time(ms): 813391

Physical reads: 19

CPU time(ms): 489527

Logical Read/sec: 5700

INSERT INTO B\_SYNC\_IN\_RESULT (ENTRY\_TIMESTAMP, LINE\_NUM, JOB\_ID, ACTION, STATUS, MESSAGE, SOURCE\_DATA) VALUES (getdate(), @P0 ,6542071,'Validate Input','OK', @P1 , @P2 )

**-- Use proper JDBC driver**

**Top query 10 (PI)**

Execution: 11308425

Total elapsed time(ms): 639485

Physical reads: 41

CPU time(ms): 636723

Logical Read/sec: 1575

select ATTR\_DATA from B\_MASTER\_REPOSITORY\_ITEM bmi where ITEM\_ID = ?

**-- "Top query 2 (PI)" already get this XML column ATTR\_DATA**

**Top query 11 (PI)**

Execution: 3

Total elapsed time(ms): 498442

Physical reads: 1379491

CPU time(ms): 14270

Logical Read/sec: 159

Select MIN(sync\_in\_result\_id) FROM B\_SYNC\_IN\_RESULT WITH (NOLOCK) WHERE JOB\_ID = ?

**Create composite index** ON (JOB\_ID, sync\_in\_result\_id) or create an index on JOB\_ID include (sync\_in\_result\_id) ?

See **Top Query 2 (DataDog sort by Avg Duration)**

**Top query 12 (PI)**

Execution: 11318252

Total elapsed time(ms): 281526

Physical reads: 0

CPU time(ms): 278899

Logical Read/sec: 786

select bimportcon0\_.IMPORT\_CONFIG\_DETAIL\_ID as import\_c1\_52\_, bimportcon0\_.MASTER\_REPOSITORY\_ID as master\_r2\_52\_, bimportcon0\_.CHANGE\_NOTIFY\_ATTR\_ID as change\_n3\_52\_, bimportcon0\_.IMPORT\_CONFIG\_ID as import\_c4\_52\_, bimportcon0\_.START\_REPOSITORY\_IND as start\_re5\_52\_, bimportcon0\_.IMPORT\_ORDER\_NUM as import\_o6\_52\_, bimportcon0\_.EXTENDED\_IMPORT\_TYPE\_CODE as extended7\_52\_, bimportcon0\_.EXTENDED\_ATTR\_START\_COL\_NAME as extended8\_52\_, bimportcon0\_.EXTENDED\_ATTR\_END\_COL\_NAME as extended9\_52\_, bimportcon0\_.OLD\_CONTENTS\_VIEW\_NAME as old\_con10\_52\_, bimportcon0\_.NEW\_CONTENTS\_VIEW\_NAME as new\_con11\_52\_, bimportcon0\_.CHANGE\_NOTIFY\_VALUE as change\_12\_52\_, bimportcon0\_.INACTIVE\_STATUS\_VALUE as inactiv13\_52\_, bimportcon0\_.SHORT\_NAME as short\_n14\_52\_, bimportcon0\_.CLONE\_BEHAVIOR\_IND as clone\_b15\_52\_, bimportcon0\_.UUID as uuid16\_52\_, bimportcon0\_.PREV\_UUID as prev\_uu17\_52\_, bimportcon0\_.CLONE\_CALLOUT\_CLASSPATH as clone\_c18\_52\_, bimportcon0\_.ALL\_DYNAMIC\_ATTRS\_IND as all\_dyn19\_52\_

from B\_IMPORT\_CONFIG\_DETAIL bimportcon0\_ where bimportcon0\_.IMPORT\_CONFIG\_DETAIL\_ID= @P0

**This table B\_IMPORT\_CONFIG\_DETAIL only has 197 records**, maybe just keep in application cache?

**Top query 13 (PI)**

-

Execution: 1222115

Total elapsed time(ms): 416634.

Physical reads: 49649

CPU time(ms): 104453

Logical Read/sec: 255.

MERGE B\_SNAPSHOT\_10151 as target using (SELECT @P0 as ITEM\_ID, getDate() as DLUD, @P1 as F\_1005477, @P2 as F\_1005670, @P3 as F\_1005671, @P4 as F\_1005106, @P5 as F\_1005107, @P6 as F\_1005085, @P7 as F\_1005125, @P8 as F\_1005056, @P9 as F\_1005060, @P10 as F\_1005109, @P11 as F\_1005058, @P12 as F\_1005081, @P13 as F\_1005668, @P14 as F\_1005057, @P15 as F\_1005669, @P16 as F\_1005119, @P17 as F\_1005131, @P18 as F\_1006042, @P19 as F\_1006043, @P20 as F\_1006044, @P21 as F\_1006045, @P22 as F\_1006046, @P23 as F\_1006047, @P24 as F\_1006048, @P25 as F\_1006049, @P26 as F\_1006050, @P27 as F\_1006051, @P28 as F\_1006052, @P29 as F\_1006053, @P30 as F\_1006054, @P31 as F\_1006055, @P32 as F\_1006056, @P33 as F\_1006057, @P34 as F\_1006058, @P35 as F\_1006032, @P36 as F\_1006059, @P37 as F\_1006060, @P38 as F\_1006061, @P39 as F\_1006033, @P40 as F\_1006064, @P41 as F\_1006034, @P42 as F\_1006041, @P43 as F\_1006035, @P44 as F\_1006063, @P45 as F\_1006036, @P46 as F\_1006037, @P47 as F\_1006038, @P48 as F\_1006039, @P49 as F\_1006040, @P50 as F\_1006062, @P51 as F\_1005103, @P52 as F\_1005063, @P53 as F\_1005931, @P54 as F\_1005129, @P55 as F\_1005067, @P56 as F\_1005062, @P57 as F\_1005126, @P58 as F\_1005453, @P59 as F\_1005061, @P60 as F\_1005114, @P61 as F\_1005121, @P62 as F\_1005673, @P63 as F\_1005145, @P64 as F\_1005135, @P65 as F\_1005136, @P66 as F\_1005133, @P67 as F\_1005134, @P68 as F\_1005665, @P69 as F\_1005672, @P70 as F\_1005083, @P71 as F\_1005084, @P72 as F\_1005082, @P73 as F\_1005077, @P74 as F\_1005079, @P75 as F\_1005466, @P76 as F\_1005467, @P77 as F\_1005113, @P78 as F\_1005078, @P79 as F\_1005080, @P80 as F\_1005150, @P81 as F\_1005068, @P82 as F\_1005070, @P83 as F\_1005112, @P84 as F\_1005139, @P85 as F\_1005441, @P86 as F\_1005054, @P87 as F\_1005123, @P88 as F\_1005583, @P89 as F\_1005053, @P90 as F\_1005055, @P91 as F\_1005132, @P92 as F\_1005144, @P93 as F\_1005130) as source on source.ITEM\_ID = target.ITEM\_ID when matched then update set DATA\_LAST\_UPDATE\_DATETIME = source.DLUD,F\_1005477 = source.F\_1005477,F\_1005670 = source.F\_1005670,F\_1005671 = source.F\_1005671,F\_1005106 = source.F\_1005106,F\_1005107 = source.F\_1005107,F\_1005085 = source.F\_1005085,F\_1005125 = source.F\_1005125,F\_1005056 = source.F\_1005056,F\_1005060 = source.F\_1005060,F\_1005109 = source.F\_1005109,F\_1005058 = source.F\_1005058,F\_1005081 = source.F\_1005081,F\_1005668 = source.F\_1005668,F\_1005057 = source.F\_1005057,F\_1005669 = source.F\_1005669,F\_1005119 = source.F\_1005119,F\_1005131 = source.F\_1005131,F\_1006042 = source.F\_1006042,F\_1006043 = source.F\_1006043,F\_1006044 = source.F\_1006044,F\_1006045 = source.F\_1006045,F\_1006046 = source.F\_1006046,F\_1006047 = source.F\_1006047,F\_1006048 = source.F\_1006048,F\_1006049 = source.F\_1006049,F\_1006050 = source.F\_1006050,F\_1006051 = source.F\_1006051,F\_1006052 = source.F\_1006052,F\_1006053 = source.F\_1006053,F\_1006054 = source.F\_1006054,F\_1006055 = source.F\_1006055,F\_1006056 = source.F\_1006056,F\_1006057 = source.F\_1006057,F\_1006058 = source.F\_1006058,F\_1006032 = source.F\_1006032,F\_1006059 = source.F\_1006059,F\_1006060 = source.F\_1006060,F\_1006061 = source.F\_1006061,F\_1006033 = source.F\_1006033,F\_1006064 = source.F\_1006064,F\_1006034 = source.F\_1006034,F\_1006041 = source.F\_1006041,F\_1006035 = source.F\_1006035,F\_1006063 = source.F\_1006063,F\_1006036 = source.F\_1006036,F\_1006037 = source.F\_1006037,F\_1006038 = source.F\_1006038,F\_1006039 = source.F\_1006039,F\_1006040 = source.F\_1006040,F\_1006062 = source.F\_1006062,F\_1005103 = source.F\_1005103,F\_1005063 = source.F\_1005063,F\_1005931 = source.F\_1005931,F\_1005129 = source.F\_1005129,F\_1005067 = source.F\_1005067,F\_1005062 = source.F\_1005062,F\_1005126 = source.F\_1005126,F\_1005453 = source.F\_1005453,F\_1005061 = source.F\_1005061,F\_1005114 = source.F\_1005114,F\_1005121 = source.F\_1005121,F\_1005673 = source.F\_1005673,F\_1005145 = source.F\_1005145,F\_1005135 = source.F\_1005135,F\_1005136 = source.F\_1005136,F\_1005133 = source.F\_1005133,F\_1005134 = source.F\_1005134,F\_1005665

**Top query 14 (PI)**

Execution: 1063305

Total elapsed time(ms): 182563

Physical reads: 60397

CPU time(ms): 105821

Logical Read/sec: 2863

SELECT s.ITEM\_ID from B\_SNAPSHOT\_10143 s WHERE s.F\_1004832 = @P0 AND s.F\_1004833 = @P1 AND s.F\_1004834 = @P2 AND s.F\_1004835 = @P3

**Missing index**:

CREATE INDEX IDX\_xxxxx ON dbo.B\_SNAPSHOT\_10143(F\_1004832, F\_1004833, F\_1004835) INCLUDE(F\_1004834)

OR

CREATE INDEX IDX\_xxxxx ON dbo.B\_SNAPSHOT\_10143(F\_1004832, F\_1004833) INCLUDE(F\_1004834, F\_1004835)

OR

CREATE INDEX IDX\_xxxxx ON dbo.B\_SNAPSHOT\_10143(F\_1004835, F\_1004832, F\_1004833) INCLUDE(F\_1004834)

**Top query 15 (PI)**

Execution: 1392799

Total elapsed time(ms): 150175

Physical reads: 57

CPU time(ms): 141912

Logical Read/sec: 967

SELECT \* FROM P\_WORK\_ITEM\_PROPERTY WHERE WORK\_VERSION\_ID= @P0 ORDER BY WORK\_PROPERTY\_ID

**Top query 16 (PI)**

Execution: 338

Total elapsed time(ms): 672403

Physical reads: 1197

CPU time(ms): 349174

Logical Read/sec: 2830

select s10148.F\_1005655 as r10225s10148F\_1005655,s10151.item\_id as s10151item\_id

FROM B\_SNAPSHOT\_10151 s10151

left outer join B\_SNAPSHOT\_10148 s10148

on ((s10148.F\_1004936 = s10151.F\_1005054) or (s10148.F\_1004936 is null and s10151.F\_1005054 is null)) and ((s10148.F\_1004935 = s10151.F\_1005053) or (s10148.F\_1004935 is null and s10151.F\_1005053 is null)) and ((s10148.F\_1004937 = s10151.F\_1005055) or (s10148.F\_1004937 is null and s10151.F\_1005055 is null))

where s10151.ITEM\_ID IN ( @P0 , @P1 , @P2 , @P3 , @P4 , @P5 , @P6 , @P7 , @P8 , @P9 , @P10 , @P11 , @P12 , @P13 , @P14 , @P15 , @P16 , @P17 , @P18 , @P19 , @P20 , @P21 , @P22 , @P23 , @P24 , @P25 , @P26 , @P27 , @P28 , @P29 , @P30 , @P31 , @P32 , @P33 , @P34 , @P35 , @P36 , @P37 , @P38 , @P39 , @P40 , @P41 , @P42 , @P43 , @P44 , @P45 , @P46 , @P47 , @P48 , @P49 )

**Top queries from DataDog**

**Top Query 1 (DataDog sort by Total Duration)**

Total Duration: 11 min 41 s

Count: 29.4K

Avg Duration: 23.9 ms

**The same query as "Top query 4 (PI)"**

**Top Query 2 (DataDog sort by Total Duration)**

Total Duration: 10 min 44 s

Count: 1.92K

Avg Duration: 336 ms

SELECT count ( ITEM\_ID ) as count from epim. dbo.B\_MASTER\_REPOSITORY\_ITEM

**Can we use this query instead**? this will be much faster, **requires Application code change**:

SELECT SUM (row\_count) AS count FROM sys.dm\_db\_partition\_stats

WHERE (index\_id=0 or index\_id=1) AND OBJECT\_NAME(object\_id) = 'B\_MASTER\_REPOSITORY\_ITEM';

Please note that the above query requires “VIEW DATABASE STATE” and “VIEW DEFINITION” permissions, db\_owner should be sufficient

**Top Query 3 (DataDog sort by Total Duration)**

Total Duration: 3 min 27 s

Count: 480

Avg Duration: 431 ms

Please note that it seems this query is **not from our application**. Maybe from monitor tool

with qstats as (

select qs.query\_hash, qs.query\_plan\_hash, qs.last\_execution\_time, qs.last\_elapsed\_time, CONCAT ( CONVERT ( VARCHAR ( ? ), CONVERT ( binary ( ? ), qs.plan\_handle ), ? ), CONVERT ( VARCHAR ( ? ), CONVERT ( varbinary ( ? ), qs.statement\_start\_offset ), ? ), CONVERT ( VARCHAR ( ? ), CONVERT ( varbinary ( ? ), qs.statement\_end\_offset ), ? ) ) as plan\_handle\_and\_offsets, (

select value

from sys.dm\_exec\_plan\_attributes ( qs.plan\_handle )

where attribute = ?

) as dbid, eps.object\_id as sproc\_object\_id, qs.execution\_count as execution\_count, qs.total\_worker\_time as total\_worker\_time, qs.total\_physical\_reads as total\_physical\_reads, qs.total\_logical\_writes as total\_logical\_writes, qs.total\_logical\_reads as total\_logical\_reads, qs.total\_clr\_time as total\_clr\_time, qs.total\_elapsed\_time as total\_elapsed\_time, qs.total\_rows as total\_rows, qs.total\_dop as total\_dop, qs.total\_grant\_kb as total\_grant\_kb, qs.total\_used\_grant\_kb as total\_used\_grant\_kb, qs.total\_ideal\_grant\_kb as total\_ideal\_grant\_kb, qs.total\_reserved\_threads as total\_reserved\_threads, qs.total\_used\_threads as total\_used\_threads, qs.total\_columnstore\_segment\_reads as total\_columnstore\_segment\_reads, qs.total\_columnstore\_segment\_skips as total\_columnstore\_segment\_skips, qs.total\_spills as total\_spills

from sys.dm\_exec\_query\_stats qs

left join sys.dm\_exec\_procedure\_stats eps ON eps.plan\_handle = qs.plan\_handle

), qstats\_aggr as (

select query\_hash, query\_plan\_hash, CAST ( qs.dbid as int ) as dbid, D.name as database\_name, max ( plan\_handle\_and\_offsets ) as plan\_handle\_and\_offsets, max ( last\_execution\_time ) as last\_execution\_time, max ( last\_elapsed\_time ) as last\_elapsed\_time, sproc\_object\_id, sum ( qs.execution\_count ) as execution\_count, sum ( qs.total\_worker\_time ) as total\_worker\_time, sum ( qs.total\_physical\_reads ) as total\_physical\_reads, sum ( qs.total\_logical\_writes ) as total\_logical\_writes, sum ( qs.total\_logical\_reads ) as total\_logical\_reads, sum ( qs.total\_clr\_time ) as total\_clr\_time, sum ( qs.total\_elapsed\_time ) as total\_elapsed\_time, sum ( qs.total\_rows ) as total\_rows, sum ( qs.total\_dop ) as total\_dop, sum ( qs.total\_grant\_kb ) as total\_grant\_kb, sum ( qs.total\_used\_grant\_kb ) as total\_used\_grant\_kb, sum ( qs.total\_ideal\_grant\_kb ) as total\_ideal\_grant\_kb, sum ( qs.total\_reserved\_threads ) as total\_reserved\_threads, sum ( qs.total\_used\_threads ) as total\_used\_threads, sum ( qs.total\_columnstore\_segment\_reads ) as total\_columnstore\_segment\_reads, sum ( qs.total\_columnstore\_segment\_skips ) as total\_columnstore\_segment\_skips, sum ( qs.total\_spills ) as total\_spills

from qstats qs

left join sys.databases D on qs.dbid = D.database\_id

group by query\_hash, query\_plan\_hash, qs.dbid, D.name, sproc\_object\_id

), qstats\_aggr\_split as (

select TOP ? convert ( varbinary ( ? ), convert ( binary ( ? ), substring ( plan\_handle\_and\_offsets, ?, ? ), ? ) ) as plan\_handle, convert ( int, convert ( varbinary ( ? ), substring ( plan\_handle\_and\_offsets, ? ?, ? ), ? ) ) as statement\_start\_offset, convert ( int, convert ( varbinary ( ? ), substring ( plan\_handle\_and\_offsets, ? ?, ? ), ? ) ) as statement\_end\_offset, \*

from qstats\_aggr

where DATEADD ( ms, last\_elapsed\_time / ?, last\_execution\_time ) > dateadd ( second, -@ P1, getdate ( ) )

)

select SUBSTRING ( text, ( statement\_start\_offset / ? ) + ?, ( ( CASE statement\_end\_offset WHEN ? THEN DATALENGTH ( text ) ELSE statement\_end\_offset END - statement\_start\_offset ) / ? ) + ? ) AS statement\_text, SUBSTRING ( qt.text, ?, ? ) as text, encrypted as is\_encrypted, s. \*

from qstats\_aggr\_split s cross apply sys.dm\_exec\_sql\_text ( s.plan\_handle ) qt

**Top Query 4 (DataDog sort by Total Duration)**

Total Duration: 2 min 36 s

Count: 1.29M

Avg Duration: 122 μs

**Same as "Top query 5 (PI)"**

UPDATE B\_MASTER\_REPOSITORY\_ITEM

……\_

WHERE ITEM\_ID = @P37

**Top Query 5 (DataDog sort by Total Duration)**

Total Duration: 3 min 27 s

Count: 480

Avg Duration: 431 ms

SELECT \*

FROM S\_USER

WHERE DELETED\_IND = @P0 AND USER\_ID <> @P1 AND APPLICATION\_USER\_IND = @P2

ORDER BY FIRST\_NAME, LAST\_NAME

**The Above table doesn’t exists in EPIM.**

**Top Query 1 (DataDog sort by Avg Duration)**

Total Duration: 1.26 S

Count: 1

Avg Duration: 1.25 s

with PagedJob as (

SELECT j.JOB\_ID, j.JOB\_TYPE\_CODE, j.STATUS AS jobStatus, jh.STATUS AS historyStatus, ja.ABORT\_IND, j.APPSERVER\_JNDI, COALESCE ( COALESCE ( jh.JOB\_DATE, j.START\_TIME ), j.NEXT\_EXECUTION\_TIME ) AS jobdate, j.NEXT\_EXECUTION\_TIME, j.START\_TIME, jh.JOB\_DATE, j.USER\_ID, jh.USER\_LOGIN, j.MASTER\_REPOSITORY\_ID, j.ADDITIONAL\_XML\_DATA, jh.LOGFILE, jh.FILENAME, jh.PROCESSING\_COUNT, jh.TOTAL\_ITEMS, jh.TOTAL\_TRANSACTIONS\_REQUIRED, jh.TOTAL\_TRANSACTIONS\_GENERATED, jh.NUM\_ERRORS, jh.NUM\_ITEM\_ERRORS, jh.NUM\_ATTR\_ERRORS, jh.NUM\_INSERTS, jh.NUM\_ITEM\_UPDATES, jh.NUM\_ATTR\_UPDATES, jh.NUM\_UPDATES, jh.NUM\_DELETES, jh.MESSAGE\_ID, row\_number ( ) over (

order by j.JOB\_ID DESC

) as RowNumber

FROM B\_JOB AS j

LEFT OUTER JOIN B\_JOB\_ABORT AS ja ON j.JOB\_ABORT\_ID = ja.JOB\_ABORT\_ID

LEFT OUTER JOIN B\_JOB\_HISTORY AS jh ON j.JOB\_ID = jh.JOB\_ID

WHERE ( j.STATUS IN ( @P0, @P1, @P2, @P3, @P4, @P5, @P6, @P7 ) ) AND ( COALESCE ( COALESCE ( jh.JOB\_DATE, j.START\_TIME ), j.NEXT\_EXECUTION\_TIME ) >= @P8 ) AND ( ja.ABORT\_IND IS ? OR ja.ABORT\_IND <> @P9 )

)

select \*

from PagedJob

where RowNumber between @P10 and @P11

order by JOB\_ID DESC

**Top Query 2 (DataDog sort by Avg Duration)**

Total Duration: 17.5 s

Count: 14

Avg Duration: 1.25 s

DELETE

from B\_SYNC\_IN\_RESULT

WHERE JOB\_ID = ? AND sync\_in\_result\_id >= @P0 AND sync\_in\_result\_id < @P1

CREATE INDEX IX\_B\_SYNC\_IN\_RESULT\_TEST ON B\_SYNC\_IN\_RESULT(JOB\_ID, sync\_in\_result\_id);

See **Top query 11 (PI)**

**Top Query 3 (DataDog sort by Avg Duration)**

**Same as "Top Query 3 (DataDog sort by Total Duration)"**

**Top Query 4 (DataDog sort by Avg Duration)**

**Same as "Top Query 2 (DataDog sort by Total Duration)"**

**Application Code change needed for:**

* **Top Query 2 (DataDog sort by Total Duration)**
* **Top query 4 (PI) , the same as (Top Query 1 (DataDog sort by Total Duration)**