

PRECISE: A PERFORMANCE MONITORING APPROACH THAT RESOLVES ISSUES QUICKER

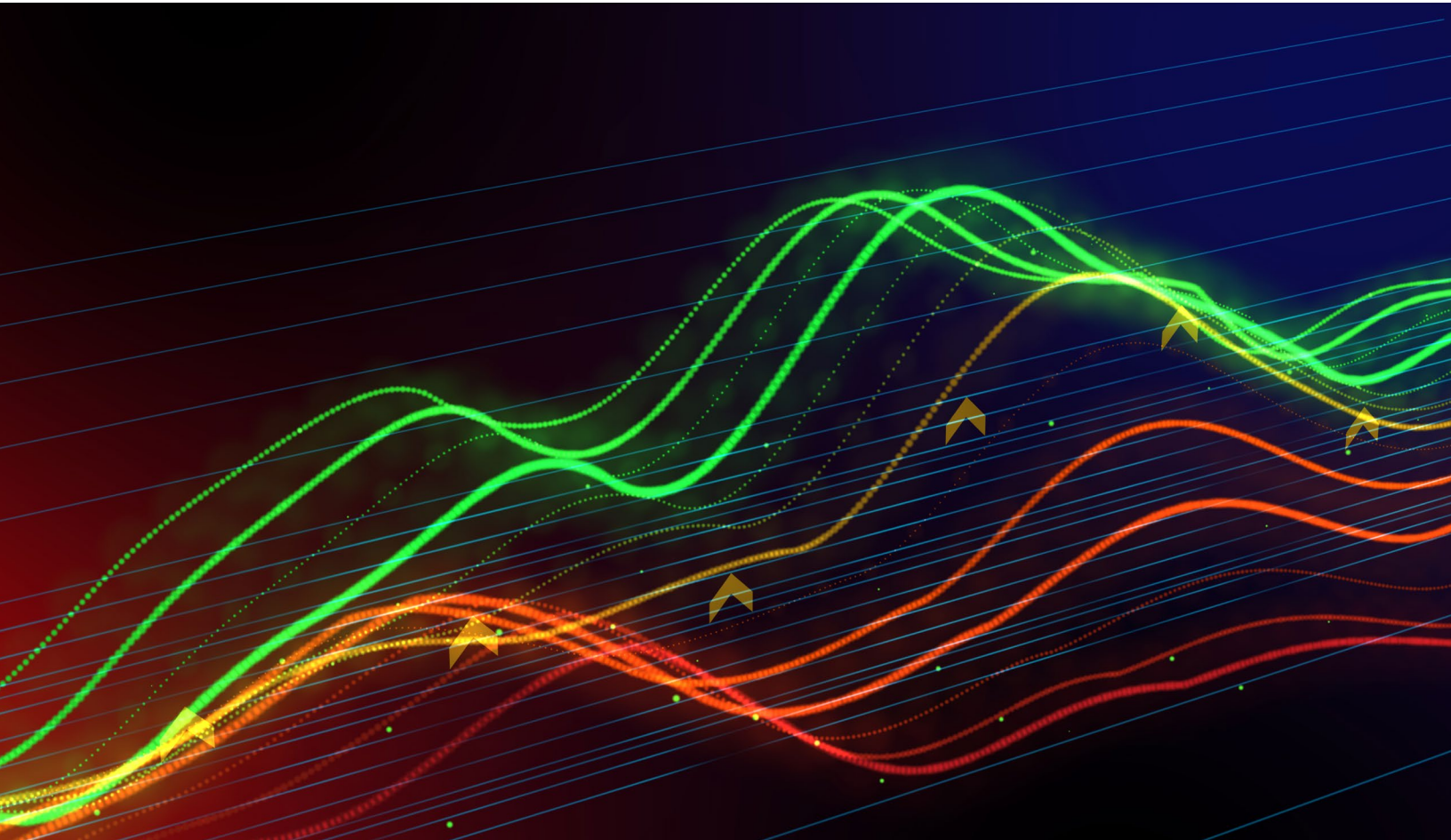
WHEN MONITORING HOW YOUR DATABASE PERFORMS, WHERE DO YOU START?

There are many methodologies to help you. You can look at the slowest transactions on average and tackle those. You could look at those wait events that are the heaviest in time.

The problem with these approaches is that they are hit or miss. A slow transaction (that is, a SQL statement) that runs for 30 seconds may seem like an obvious choice. However, that statement can only run once. Whereas a transaction that executes 3 seconds on average and executes over 1000 times is a better choice for investigating performance.

It would be a better approach to understand what is consuming the most time and resource in the database or instance and then focus on that. This approach shows you the events that are consuming the most time and resource and therefore having the biggest impact on the instance.

Knowing what is consuming the most resources and time allows you to achieve the biggest benefit by tackling these events first. If an event is consuming 25% or more of the processing time in an instance, then by tuning that event first might allow us to save up to 25% of that instance. If we tackle an event that is consuming only 5% of the time and resource in the instance, then the biggest improvement we can make is only up to 5%.



WHAT DO I MEAN BY AN EVENT?

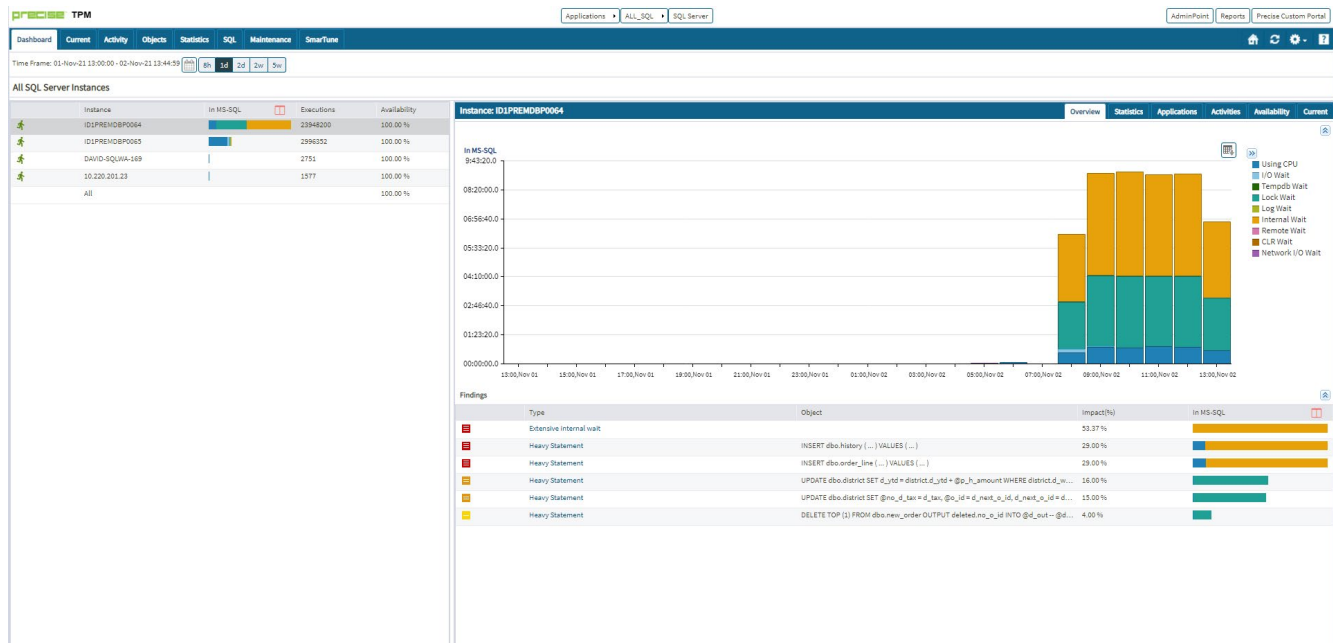
- An event is anything that is consuming time and resources in the instance.
- An event could be a SQL statement that is running.
- An event can be a wait event that is consuming a lot of time, such as a locking issue.
- An event could also be a database object (that is, a table or index) that is hot and is being accessed often and inefficient.

A lot of performance monitoring tools do not use this approach and therefore do not help you achieve the benefit of reducing performance issues in your instance.

With Precise, we have a tool that can monitor performance and that uses this approach of identifying those events that are consuming the most resources and time.

Precise categorizes the events and shows you those that are consuming the most in your instance.

Precise identifies performance issues and labels them as 'Findings'. It color codes them based on the impact that they are having on your instance.



RED FINDING

A red finding is a performance issue that is consuming over 20% of the total time in your instance. This is something that one needs to investigate with priority.

AMBER FINDING

An amber finding is a performance issue that is consuming over 10% of the total time in your instance. This is having a heavy impact on your instance.

YELLOW FINDING

A yellow finding is a performance issue that is consuming over 5% of the total time in your instance.

This is the starting point for looking at how your instance performs. By investigating each of these performance issues, you will go a long way to reducing the performance impact on your instance.

Precise allows you to investigate each of these performance issues fast and identify recommendations for them.



Type	Object	Impact(%)	In MS-SQL
Extensive internal wait		53.37 %	
Heavy Statement	INSERT dbo.history (...) VALUES (...)	29.00 %	
Heavy Statement	INSERT dbo.order_line (...) VALUES (...)	29.00 %	
Heavy Statement	UPDATE dbo.district SET d_ytd = district.d_ytd + @p_h_amount WHERE district.d_w...	16.00 %	

The statement being run by user **dbo**, on the TPCC database, consumes **16.00 %** of SQL Server resources.

Learn more or proceed with the following:

- Tune the statement.
- Examine the statement activity over time and related programs.
- Examine indexes, statistics and partitions recommendations for the statement.

Heavy Statement	UPDATE dbo.district SET @no_d_tax = d_tax, @o_id = d_next_o_id, d_next_o_id = d...	15.00 %	
Heavy Statement	DELETE TOP (1) FROM dbo.new_order OUTPUT deleted.no_o_id INTO @d_out -- @d...	4.00 %	

“

Precise allows you to investigate each of these performance issues fast and identify recommendations for them.

”

Precise shows performance issues within SQL statements and provides the ability to diagnose the performance issue. This includes poor access paths, missing indexes or locking, and provides recommendations for fixing them.

The screenshot shows the Precise TPM interface with the following details:

- Statement:** 13283.51006.16186.43857
- Execution Plan:** Estimated Execution Plan as of Jun 24, 2021 13:03. It shows a nested loop inner join with two child nodes:
 - 1) Index Seek w/ Stalls (49%)
 - 2) Clustered Index Seek customer_j1 (49%)
- SQL Text:**

```
UPDATE dbo.district
SET @no_d_tax = d_tax,
    @o_id = d_next_o_id,
    @next_o_id = district_d_next_o_id + 1
WHERE district_d_id = @no_d_id
AND district_d_o_id = @no_o_id
```
- Findings:** A table with columns: Locate, Type, Object, Potential Gain (%). It lists one finding: "Locked Statement" with a potential gain of 98.65%.

Precise shows performance issues related to wait events and allows you to investigate these further.

The screenshot shows the Precise TPM interface with the following details:

- Instance:** ID1PREMDBP0064
- Top 10 Wait Events:** A pie chart showing the distribution of wait events. The legend includes:
 - REQUEST_FOR_DEADLOCK_S...
 - LOGMGR_QUEUE
 - LOGMGR_QUEUE
 - LATCH_EX
 - LATCH_EX
 - LATCH_EX
 - LATCH_EX
 - LATCH_EX
 - LATCH_EX
 - LATCH_EX
- Wait Events Table:**

Wait Name	Description	Wait Group	Wait for Resource	Times Waited Instance Level (%)	Times Waited Group Level (%)	Waits
LOGMGR_QUEUE	Occurs while the log writer task waits for work requests.	Log wait		21.05%	99.95%	694152
LATCH_EX	Occurs when a task is waiting for a generic event to occur.	IO		21.00%	66.56%	64046
CHECKPOINT_QUEUE	Occurs while the checkpoint task is waiting for the next checkpoint request.	Internal/Miscellaneous		18.80%	100.00%	155
LATCH_EX	Occurs when a task is waiting to acquire an Exclusive lock. For a lock compatibility matrix, see sys.dm_...	Lock wait		11.84%	85.18%	66172
SLEEP_TASK	Occurs when a task sleeps while waiting for a generic event to occur.	IO		10.55%	32.44%	27874-K
LATCH_SH	Occurs when waiting for an Sx (share) latch. This does not include buffer latches or transaction mark...	Internal/Latch		7.88%	63.42%	2900-K
LATCH_EX	Occurs when waiting for an Ex (exclusive) latch. This does not include buffer latches or transaction m...	Internal/Latch		4.55%	36.58%	1862-K
SLEEP_BPOOL_FLUSH	Occurs when a checkpoint is throttling the issuance of new I/Os in order to avoid flooding the disk s...	IO		1.50%	79.52%	466802
LCK_M_U	Occurs when a task is waiting to acquire an Update lock. For a lock compatibility matrix, see sys.dm_...	Lock wait		1.46%	10.53%	153131
LCK_M_S	Occurs when a task is waiting to acquire a Shared lock. For a lock compatibility matrix, see sys.dm_br...	Lock wait		0.80%	4.29%	34424
ASYNC_IO_COMPLETION	Occurs when a task is waiting for I/Os to finish.	IO		0.39%	10.06%	9
ASYNC_NETWORK_IO	Occurs on network writes when the task is blocked behind the network. Verify that the client is proce...	Network IO		0.14%	100.00%	9072
PAGELOCK_EX	Occurs when a task is waiting on a latch for a buffer that is in an I/O request. The latch request is in E...	IO		0.13%	6.82%	12017
PAGELATCH_SH	Occurs when a task is waiting on a latch for a buffer that is not in an I/O request. The latch request is L...	Internal/Buffer Pool Wait		0.12%	81.19%	15590
PAGELATCH_SH	Occurs when a task is waiting on a latch for a buffer that is in an I/O request. The latch request is in I...	IO		0.06%	3.37%	4656
REQUEST_FOR_DEADLOCK_S...	Occurs when a task is waiting on a latch for a buffer that is not in an I/O request. The latch request is I...	Internal/Buffer Pool Wait		0.05%	44.76%	9178

Follow the correct method for investigating performance issues in your instance and you will soon have your instance and applications performing better.

PRECISE

Precise helps database and IT administrators to find and fix database and application performance problems in physical, virtual, and cloud environments. Unlike its competition, it provides deep database optimization, end-to-end transaction view, isolation of problems and causes, scalable deployment, what-if analysis for changes, and history, trending, and planning.

Start for Free

