

Db2 Query Monitor Anomaly Detection Feature

by Tom Hubbard

With the application of PTFs UI67426, UI63751, and UI67428, Db2 Query Monitor (CQM) will have the ability to set exception and alert thresholds based on deviations from “normal” performance for an SQL statement. CQM will automatically adapt and adjust the thresholds as the system performance changes throughout the day.

When anomaly detection is enabled, Db2 Query Monitor will calculate the standard deviation for all monitored SQL. The StdDev for each SQL statement is calculated for the current interval and stored in the VSAM back-store datasets. In addition, CQM will also keep rolling mean and standard deviation values. These values are calculated constantly for the life of the CQM subsystem. These “rolling” values are used for triggering anomaly-based exceptions and alerts.

Db2 Query Monitor administrators can configure the anomaly detection for distinct SQL workloads. This enables Db2 Query Monitor to inform users with alerts or exceptions whenever the performance of SQL deviates from recent historical averages. When configuring anomaly detection, administrators can specify how sensitive the anomaly detection should be to deviations. The thresholds used to trigger exceptions/alerts by anomaly detection change as a function of how a system operates over time. That is CQM uses a rolling calculation so that the triggering threshold changes as the SQL performance changes. The speed with which CQM changes is controlled by the ADETECT_ALPHA_SMOOTHING startup parameter described below. The larger the value provided, the faster the code will “learn” about normal performance changes.

The new detection and sensitivity thresholds are controlled by new startup parameters and monitoring profile options. The new startup parameters are included in the Db2 Query Monitor User’s Guide, and are also described here:

ADETECT_ENABLED(Y) - Instructs Db2 Query Monitor whether or not to collect anomaly detection data.

SUMMARY_VARIATION(Y) - Instructs Db2 Query Monitor whether or not to collect standard deviation data in its summary metrics collection for workloads.

ADETECT_MIN_COUNT_FOR_ANOMALY(100) - Specifies the number of SQL statements to be seen for a given workload prior to Db2 Query Monitor detecting anomaly exceptions or alerts for the workload.

ADETECT_MIN_GETPAGES_DELTA_FOR_ANOMALY(100) - Specifies the minimum deviation from rolling mean that can be potentially categorized as an anomaly.

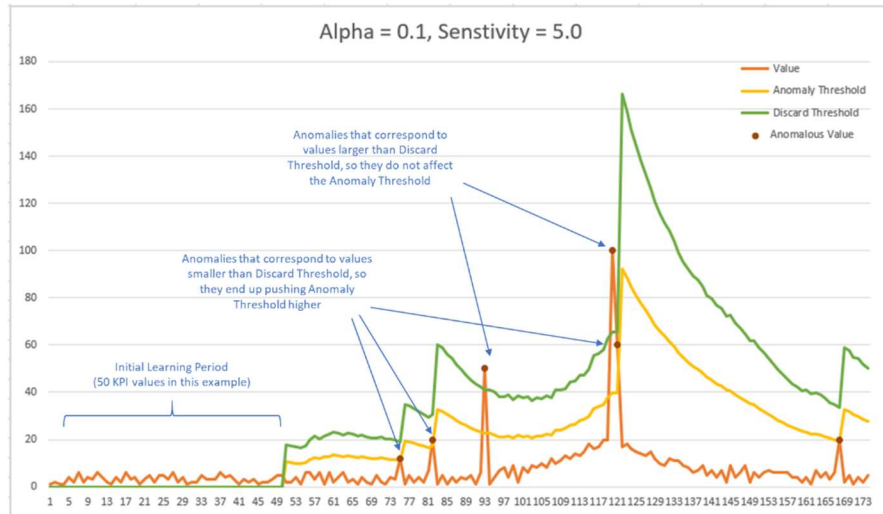
ADETECT_MEMORY_LIMIT(1024) - Specifies the number of megabytes available for Db2 Query Monitor's anomaly detection process.

ADETECT_ALPHA_SMOOTHING(0.05) – This parameter is used in calculating the new rolling mean each time an SQL statement is captured. The value is used as a degree of weighting decrease, a

constant smoothing factor between 0 and 1. A higher α value discounts older observations faster. So, if the smoothing value is 0.1, then the formula looks something like this:

$$\text{New rolling Mean} = (0.1 * \text{SQL_Observed_Value}) + (0.9 * \text{Current_Rolling_Mean})$$

The chart below shows the effect of using a 0.1 smoothing value on the anomaly threshold.



The panel for updating monitoring profile lines in the ISPF UI has been updated to include the new exception threshold specifications is shown below:

Anomaly-based Exceptions

CPU Anomalies	Y	(Y/N)	Toleration Level	15	_____
Elapsed Anomalies	N	(Y/N)	Toleration Level	6	_____
Getpage Anomalies	Y	(Y/N)	Toleration Level	15	_____

The “Toleration Level” or threshold specified for each type of anomaly based alert represents the number of standard deviations above the rolling mean which will cause the specified type of exception or alert to be generated. The calculation would look something like this:

$$\text{CPU_Time} > \text{rolling_mean} + (\text{Toleration_Level} * \text{StdDev}) = \text{trigger alert.}$$

CPU Anomalies - Indicates whether or not to generate exceptions for CPU anomaly and what Toleration Level (threshold) is set for it.

Elapsed Anomalies - Indicates whether or not to generate exception for elapsed time anomaly and what Toleration Level (threshold) is set for it.

Getpage Anomalies - Indicates whether or not to generate exception for getpages anomaly and what Toleration Level (threshold) is set for it.

In addition to the new exception specification, there is also an area for specifying the corresponding alert tolerations levels. That sections of the panel is shown below:

Anomaly-based Alerts

CPU Anomalies	N	(Y/N)	Toleration Level	5	_____
Elapsed Anomalies	N	(Y/N)	Toleration Level	5	_____
Getpage Anomalies	N	(Y/N)	Toleration Level	5	_____

As with threshold based alerts, all alerts are only sent to the CAE Server. No alert is stored in either the VSAM back-store datasets or offloaded to the Db2 table archives. So, be sure that the toleration levels for alert anomalies is greater than the corresponding exception toleration level.

In addition to the threshold specification, the administrator will need to specify a level of deviation which will tell Db2 Query Monitor to not include the instance of a particular SQL statement in calculating the new rolling mean. This discard value is necessary in order to not allow extreme variances to abnormally skew the rolling mean calculation.

Discard Above (sigmas)	
CPU Discard Level	30
Elapsed Discard Level	30
Getpage Discard Level	30

The screen capture below shows a sample of the information captured by anomaly based CPU exceptions. Note the new columns which have been added to the display. The new columns are described after the screen capture.

2020/02/12 11:18:03 ----- Summary Exceptions ----- Row 54 of 80

Option ==>

DB2 QM Subsystem: QM/R1

Filters Enabled : N

CMD - Enter "/" to select action

Interval Start Date: 02/12/2020 Time: 10:45:58

Interval End Date: CURRENT Time: CURRENT

Scroll ==> CSR

CMD	SSID	Exceptions	Workload	Plan	Program	Section	DB2 CPU Time	RollAv CPU	StdDev CPU	CPU Factor	ECPU RollM	ACPU RollM	
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.000972	0.000046	0.000055	16.704288	0.000878	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.000970	0.000048	0.000054	17.056957	0.000859	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.000877	0.000037	0.000054	15.518324	0.000849	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.000946	0.000031	0.000057	16.023491	0.000888	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.001157	0.000042	0.000044	25.267846	0.000704	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.001169	0.000032	0.000063	18.073172	0.000976	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.001006	0.000030	0.000059	16.675996	0.000908	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.001037	0.000052	0.000056	17.446394	0.000899	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.000879	0.000041	0.000055	15.259770	0.000865	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.001082	0.000052	0.000065	15.906788	0.001023	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.001016	0.000046	0.000053	18.339646	0.000839	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.001027	0.000031	0.000049	20.483906	0.000760	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.001107	0.000038	0.000069	15.519872	0.001071	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.001027	0.000049	0.000057	17.279553	0.000898	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.000950	0.000047	0.000058	15.602988	0.000915	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.000990	0.000029	0.000047	20.435153	0.000734	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.001179	0.000032	0.000063	18.112145	0.000982	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.001128	0.000048	0.000053	20.415750	0.000842	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.001100	0.000033	0.000057	18.702304	0.000889	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.000951	0.000048	0.000051	17.685811	0.000814	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.001139	0.000036	0.000073	15.066063	0.001134	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.000966	0.000036	0.000050	18.492806	0.000790	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.001156	0.000034	0.000060	18.745946	0.000932	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.001099	0.000032	0.000061	17.562468	0.000943	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.001063	0.000045	0.000057	17.860370	0.000900	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.001073	0.000043	0.000055	18.753385	0.000867	0.000000
---	IAIA	P	Anomaly	worklo	DSNREXX	PRJADD	18	0.001131	0.000040	0.000063	17.293029	0.000986	0.000000
***** Bottom of Data *****													

RollAv CPU – This column represents the value of the CPU rolling average at the time the exception was generated.

StdDev CPU – This column represents the value of the CPU rolling standard deviation at the time the exception was generated.

CPU Factor - This column represents the amount of standard deviations from the mean the CPU was at the time the exception was generated.

ECPU RollLm - This column represents the tolerance level that would have to be crossed in order for a SQL statement to generate an anomaly exception for CPU.

ACPU RollLm - This column represents the tolerance level that would have to be crossed in order for a SQL statement to generate an anomaly alert for CPU.

CPU Dsc – This column indicates whether or not the value of the CPU for a given SQL statement was factored into the rolling mean and standard deviation.

CPU DscT - This column represents the discard level that would have to be crossed in order for a SQL statement's CPU time not to update the rolling mean and standard deviation.

Corresponding new columns have been added to support elapsed time and getpage anomaly based exceptions. All of the new columns are documented in the on-line help and Db2 Query Monitor User's Guide.

[Recommendations for initial settings](#)

Initial settings for the "Tolerance Level" should be limited to turning on exceptions only. Once a level is identified which does not result in an excessive number of exceptions, then a level to start generating alerts can be established.

CPU time and GetPage exceptions tend to be more consistent when setting exceptions in a non-production system. Elapsed time exceptions and alerts probable should not be used in a non-production system. However, in a production system, elapsed time may be a key indicator of total transaction response time. Therefore, elapsed time exceptions and alerts should be carefully considered in the production environment.