EM 13c, EM 12c: How to Prevent Alerts from the Server Adaptive Threshold Metric (Server_Adaptive_Threshold_Metric) (Doc ID 1929003.1)

In this Document

APPLIES TO:

Enterprise Manager for Oracle Database - Version 12.1.0.6.0 and later
Information in this document applies to any platform.

GOAL

To prevent alerts from the 'Server Adaptive Threshold' metric in Enterprise Manager (EM) Cloud Control.
An example of an email alert from the server_adaptive_threshold_metric:

```
Host=<host name>
Target type=Database Instance
Target name=<instance name>
Categories=Load
Message=Metrics "Physical Writes Per Sec" is at 2.449
Severity=Critical
Event reported time=Sep 23, 2014 2:23:01 PM BST
Operating System=Linux
Platform=x86_64
Associated Incident Id=7262
Associated Incident Status=New
Associated Incident Owner=
Associated Incident Acknowledged By Owner=No
Associated Incident Priority=None
Associated Incident Escalation Level=0
Event Type=Metric Alert
Event name=Server_Adaptive_Threshold_Metric:instance_throughput__physwrites_ps
Metric Group=Server_Adaptive_Threshold_Metric
Metric=Physical Writes Per Sec
Metric value=2.44918360546485
Key Value=SYSTEM
Rule Name=DEFAULT_RULESET_FOR_ALL_TARGETS,METRIC_ALERT_INCIDENT_CREATION
Rule Owner=
Update Details:
Metrics "Physical Writes Per Sec" is at 2.449
Incident created by rule (Name = Incident management rule set for all targets, Create incident for critical metric alerts [System generated rule]).
```

An example of the alert in incident manager:

Background

There have been some changes to the way that metrics are evaluated in Cloud Control 12.1.0.4 (when used in conjunction with the 12.1.0.6 or higher database plug-in). The biggest change is that most metrics which were previously evaluated by the server generated alert system (in older versions of Enterprise Manager) will now be evaluated purely by the agent. In 12.1.0.4 and higher the new behaviour is that all metrics where it is possible to set a threshold (via the 'metric and collection settings' screen) will be evaluated by the agent and the server generated alert system is no longer relied on. See Note 1680788.1 "EM 12c, EM 13c: Changes to Oracle Database Monitoring in Enterprise Manager Cloud Control 12.1.0.4 and later with DB-Plugin 12.1.0.6 and later" which explains this further.

The 'server adaptive threshold' metric is *not* one of the metrics which uses the 'new behavior'. It is not possible to set thresholds for the 'server adaptive threshold metric' in Enterprise Manager's 'metric and collection settings' screen nor is it possible to disable this metric via the EM interface. This means that the 'server adaptive threshold' metric will still be evaluated via the server generated alert system and will generate an alert according to the thresholds which are set in sys.dba_thresholds on the database.

Please note that the server adaptive threshold metric has been provided for backwards compatibility only for customers who need to continue to utilize the server generated alert system for database adaptive threshold metrics. However, EM 12.1.0.4 and higher also provides a new "Adaptive Threshold" mechanism, which is controlled by the Enterprise Manager metric alert subsystem and which does not rely on the server generated alert system or the settings in sys.dba_thresholds. This is the preferred method to be used when working with adaptive thresholds (thresholds which are not set statically but which self modify over time according to user specifications). These new adaptive thresholds features can be found under Advanced Threshold Management and can be accessed as follows:

- Database Home Page Oracle Database pull-down menu
- Monitoring /Metric and Collection Settings
- scroll down to Related Links/Advanced Threshold Management

It is recommended that these new 'adaptive threshold management/adaptive metrics ' be used instead of relying on the 'server adaptive threshold metric'. For example, it is possible to create a monitoring template which sets the 'advanced threshold management/adaptive metrics' . It is not possible to create a monitoring template based on the 'server adaptive threshold metric' because the thresholds are not controlled by 'metric and collection settings' page, but are set individually for each database.

The 'server adaptive threshold' metric will fire when the thresholds are crossed which are set on the database itself (in sys.dba_thresholds). This means that if any of the following metrics, have thresholds set in sys.dba_thresholds, the server adaptive threshold metric could fire if the thresholds are crossed:-

Average Active Sessions, Average Synchronous Single Block Read Latency (ms), CPU Usage Per Txn, Cumulative Logons Per Sec, Cumulative Logons Per Txn, Current Logons Count, Current Open Cursors Count, DBWR Checkpoints Per Sec, Database Block Changes Per Txn, Enqueue Requests Per Txn, Executes Per Sec, Global Cache Average CR Block Request Time (centi-seconds), Global Cache Average, current Block Request Time (centi-seconds), Hard Parses Per Sec, I/O Megabytes Per Sec, I/O Requests Per Sec, Network Bytes Per Sec, Number of Transactions Per Sec, Parallel Execution Downgraded Per Sec, Parallel Execution Downgraded to Serial Per Sec, Physical reads Per Sec, Physical Reads per TXN, Physical Writes per sec, physical writes per txn, physical writes per txn, recursive calls per sec, recursive calls per txn, redo generated per sec, redo generated per txn, redo writes per txn, response time (centi-seconds per call), response time per txn, rows processed (per sort), scans on long tables per sec , scans on long tables per txn, session logical reads per sec, session logical reads per txn, sorts to disk per txn, total parses per sec, total parses per txn, totaltable scans per sec, total table scans per txn, user calls per sec, user calls per txn, user commits per sec, user commits per txn, user rollbacks per sec, user rollbacks per txn.

For a database which is newly added to EM Cloud Control it is likely that the thresholds will not be set on the database in sys.dba_thresholds for these metrics. This is because EM provides a different set of metrics (mostly under 'Throughput') For example:

Throughput/Average Active Sessions
Throughput/Physical Reads Per Sec

To see the full list, go to

- Database home page
- Oracle Database (drop down menu)

- Monitoring/metric and collection settings/Choose "All metrics"

Throughput metrics which can be set like this from the "metric and collection settings" page, will not update the thresholds in sys.dba_thresholds because they will be evaluated purely by the Agent.
If it is required to utilize the 'server adaptive threshold' metric (for backwards compatibility purposes), then the thresholds should be set as follows:

- Go to database home page/performance/adaptive thresholds
- locate the metric, click on Edit Thresholds
- Setting the thresholds in this way, will update sys.dba_thresholds.

How this metric can be influenced by older Enterprise Manager products

In previous versions of Enterprise Manager, if the throughput metric was set (for example), this relied on the server generated alert system. This means that in previous versions of the product, setting a 'throughput' metric would have updated the sys.dba_thresholds on the database. It is possible therefore, that if the database is also being monitored by an alternative agent (eg. dbconsole agent component, earlier version of Cloud Control agent, or Grid control agent), that the thresholds in sys.dba_thresholds could be set (by the other 'old' EM agent).   It is also possible that if EM has been updated from 12.1.0.x to 12.1.0.4 (or higher), that some of the databases monitored may have had metrics set which will have modified sys.dba_thresholds on the database.

SOLUTION

If alerts from the 'Server Adaptive Threshold Metric' are unwanted, then it is necessary to do the following:

a) Check whether any adaptive thresholds are configured on the database

Run the following SQL (on the database on which the alert is generated)

```
select distinct d.metric_id, m.metric_name,d.warning_param, d.critical_param, d.adaptive,
d.in_effect
from dbsnmp.bsln_threshold_params d, v$metricname m
where d.metric_id=m.metric_id ;
```

Any metric returned from this query will be using an adaptive threshold.

b) Remove the adaptive and static thresholds

Both adaptive and static thresholds can be removed from:

- database home page/performance/adaptive thresholds
- locate the metric, click on Edit Thresholds
- click on the button Clear Thresholds (or remove any critical and warning values so that the Critical and Warning threshold setting boxes are blank)
- and ok this

c) Confirm that the adaptive threshold has been removed in dbsnmp.bsln_threshold_params (adaptive thresholds only)

If an adaptive threshold was identified in step a) confirm that the adaptive threshold no longer shows in dbsnmp.bsln_threshold_params by running the query from step a) again.

d) Confirm that that the thresholds are removed from dba_thresholds

For all thresholds (adaptive and static), query:

```
select metrics_name, warning_value,critical_value, instance_name, object_type,object_name from
dba_thresholds;
```

If the thresholds do not get unset, for example on a cluster database:

select * from dba_thresholds where metrics_name like '%Cursor%';

```
METRICS_NAME WARNING_OPER WARNING_VA CRITICAL_OPE CRITICAL_VALUE OBSERVATION_PERIOD
CONSECUTIVE_OCCURRENCES INSTANCE_NAME OBJECT_TYP OBJECT_NAME STATUS
------------------------------ ------------ ---------- ------------ -------------------- ------------------ ----------------------- ----------------
---------- -------------------- -------
Current Open Cursors Count GT 1200 NONE 1 3 orcl SYSTEM VALID
Current Open Cursors Count GT 1200 NONE 1 3 orcl_db02 SYSTEM VALID
```

then use the instance name and db name as follows:

```
SQL> BEGIN
DBMS_SERVER_ALERT.SET_THRESHOLD(
metrics_id => DBMS_SERVER_ALERT.OPEN_CURSORS_CURRENT,
warning_operator => NULL,
warning_value => NULL,
critical_operator => NULL,
critical_value => NULL,
observation_period => NULL,
consecutive_occurrences => NULL,
instance_name => 'orcl_db02',
object_type => DBMS_SERVER_ALERT.OBJECT_TYPE_SYSTEM ,
object_name => NULL
);
END;
/

SQL> BEGIN
DBMS_SERVER_ALERT.SET_THRESHOLD(
metrics_id => DBMS_SERVER_ALERT.OPEN_CURSORS_CURRENT,
warning_operator => NULL,
warning_value => NULL,
critical_operator => NULL,
critical_value => NULL,
observation_period => NULL,
consecutive_occurrences => NULL,
instance_name => 'orcl',
object_type => DBMS_SERVER_ALERT.OBJECT_TYPE_SYSTEM ,
object_name => NULL
);
END;
/

commit;
```

And then check dba_thresholds again.

e) Alternatively, the thresholds can be removed from dbms_server_alert package.

An example of changing the thresholds for the 'Throughput/Physical Reads Per Sec Metric' is given below.

If the metric Throughput/Physical Reads Per Sec is set in Monitoring/Metric and Collection Settings, this will be evaluated by the agent. However if thresholds for this metric also exist on the database in dba_thresholds, then it's possible that an alert could also be received from the Server Adaptive Threshold Metric. If the intention is to prevent the Server Adaptive Threshold Metric alerts, then the threshold can be set to a higher value (or set to null) on the database itself, eg.:

Example of setting the thresholds to a higher value:

```
BEGIN
DBMS_SERVER_ALERT.SET_THRESHOLD(
metrics_id => DBMS_SERVER_ALERT.PHYSICAL_READS_SEC,
warning_operator => DBMS_SERVER_ALERT.OPERATOR_GE,
warning_value => '300',
critical_operator => DBMS_SERVER_ALERT.OPERATOR_GE,
critical_value => '500',
observation_period => 1,
consecutive_occurrences => 2,
instance_name => 'orcl',
object_type => DBMS_SERVER_ALERT.OBJECT_TYPE_SYSTEM ,
object_name => ''
);
END;
/
```

Or an example of setting the thresholds to a Null value

```
BEGIN
DBMS_SERVER_ALERT.SET_THRESHOLD(
metrics_id => DBMS_SERVER_ALERT.PHYSICAL_READS_SEC,
warning_operator => NULL,
warning_value => NULL,
critical_operator => NULL,
critical_value => NULL,
observation_period => NULL,
consecutive_occurrences => NULL,
instance_name => 'orcl',
object_type => DBMS_SERVER_ALERT.OBJECT_TYPE_SYSTEM ,
object_name => ''
);
END;
/
```

then confirm that the thresholds have been changed either by:

```
select metrics_name, warning_value, critical_value, consecutive_occurrences from dba_thresholds;
```

or with the dbms_server_alert procedure:

```
DECLARE
warning_operator BINARY_INTEGER;
warning_value VARCHAR2(60);
critical_operator BINARY_INTEGER;
critical_value VARCHAR2(60);
observation_period BINARY_INTEGER;
consecutive_occurrences BINARY_INTEGER;
BEGIN
DBMS_SERVER_ALERT.GET_THRESHOLD(
DBMS_SERVER_ALERT.PHYSICAL_READS_SEC, warning_operator, warning_value,
critical_operator, critical_value, observation_period, consecutive_occurrences, 'orcl',
DBMS_SERVER_ALERT.OBJECT_TYPE_SYSTEM, '');
DBMS_OUTPUT.PUT_LINE('Warning operator: ' || warning_operator);
DBMS_OUTPUT.PUT_LINE('Warning value: ' || warning_value);
DBMS_OUTPUT.PUT_LINE('Critical operator: ' || critical_operator);
DBMS_OUTPUT.PUT_LINE('Critical value: ' || critical_value);
DBMS_OUTPUT.PUT_LINE('Observation_period: ' || observation_period);
DBMS_OUTPUT.PUT_LINE('Consecutive occurrences:' || consecutive_occurrences);
END;
/
```

For example, sample output will be:

```
Warning operator: 4
Warning value: 121
Critical operator: 4
Critical value: 220
Observation_period: 1
Consecutive occurrences:2
```

REFERENCES

Didn't find what you are looking for?