Module 3: Locking and Concurrency

Student Lab Manual

SQL Server 2012: Performance Tuning – Design, Internals, and Architecture

Version 1.0

Locking and Concurrency

Introduction

In this Lab, we will simulate a production environment with blocking issue and deadlock issue. You can practice how to troubleshoot these problems and try several methods to resolve them.

Objectives

After completing this lab, you will be able to:

- Find blocking and analyze blocking issue.
- Use extended events to trouble shoot blocking and deadlock issue.
- Understand read committed snapshot behavior and optimize heavy blocking environment.

Prerequisites

- Basic knowledge about lock concept and transaction isolation level concept
- Basic skills of profiler usage

Estimated time to complete this lab

30 minutes

Scenario

There are several applications based on AdventureWorks database. These applications include OLTP type sales order processing and OLAP type statistical query. Recently, customer has found some queries running very slowly and the development team also reported that they see database 1205 errors in the application log.

Please note that some of problems may be related to store procedures and triggers.

Exercise 1: Identify blocking

Objectives

In this exercise, you will:

- Identify blocking using DMVs
- Identify blocking using Extended Events

Prerequisites

- Connect to the SQL2012PT Virtual Machine
- Log in using the following credentials

  User: Administrator

  Password: P@ssw0rd

Note: The Virtual Machine for this workshop is time bombed for security purposes. You may need to rearm the virtual machine if the activation has expired. If the VM issues a message that it needs to be reactivated, you can use slmgr.vbs with the rearm option as follows:

1. Open an elevated command prompt (right click on "Command Prompt" in the Start menu and click "Run As Administrator)

2. Execute the following command
   slmgr.vbs –rearm

Scenario

Customer reported that some queries running very slow and suspected that our databases have blocking issue. Now you need to identify the reason of blocking and related queries.

Generate User Activities and troubleshoot blocking

1. Navigate to the directory \Labs\Module3\Exercise1\, and double-click the batch file GenerateData.cmd. This file should run for roughly 5 minutes and close automatically when completed. Do not wait for this batch file to finish. Continue to the next step while it is running.

2. Double-click the GenerateActivity.cmd batch file. This action will call several T-SQL scripts to generate user activities on the server, and as a result, you will see several SQLCMD windows open up. Do not close any of these command windows at this time. Continue to the next step while it is running.

3. Open ViewBlockingDMVs.sql in SSMS and run the following select statement to determine if any blocking is occurring. You may need to run this a few times before you catch any blocking.

```
SELECT
CASE WHEN er.session_id IS NULL
               THEN es.session_id
               ELSE er.session_id
       END AS session_id,
       er.blocking_session_id,
       er.wait_time,
       er.wait_resource,
       er.wait_type,
       er.last_wait_type,
       er.status,
       CASE WHEN er.session_id IS NULL
               THEN (SELECT text
FROM sys.dm_exec_sql_text(ec.most_recent_sql_handle))
               ELSE (SELECT text
FROM sys.dm_exec_sql_text(er.sql_handle))
       END AS QueryText
FROM sys.dm_exec_connections ec
       JOIN sys.dm_exec_sessions es ON ec.session_id = es.session_id
       LEFT JOIN sys.dm_exec_requests er ON es.session_id = er.session_id
WHERE er.blocking_session_id > 0
OR es.session_id IN (SELECT blocking_session_id
FROM sys.dm_exec_requests
WHERE blocking_session_id > 0)
```

Question A: Which session_id has blocking_session_id column value > 0?

Question B: What are the values for status, last_wait_type and wait_resource of blocked session?

4. Looking at the query text of the blocking and blocked session:

Question C: Can you find the cause of the blocking by examining the text of the SQL statement?

5. Make sure your configuration parameter 'blocked process threshold (s)' is set to 5, otherwise use the statement below to set it :

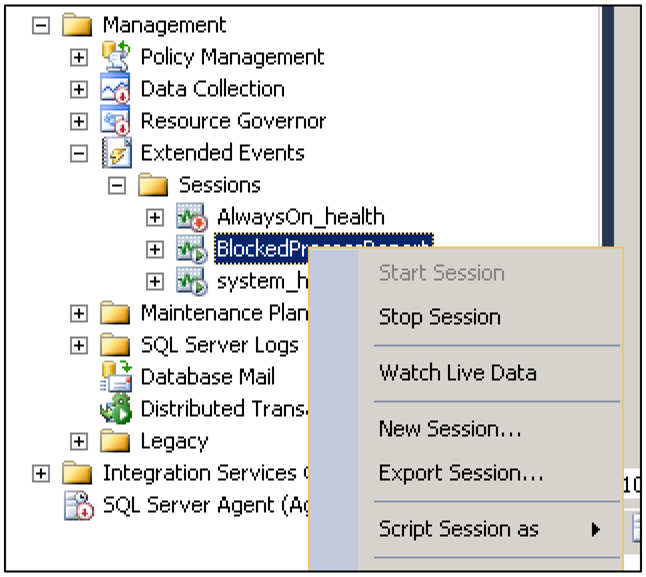
```
sp_configure 'blocked process threshold (s)',5
go
RECONFIGURE WITH OVERRIDE
Go
```

6. Open and execute the file BlockedProcessReport.sql to create and start an Extended Events session that will collect the Blocked Process Report event in an in-memory Ring Buffer.

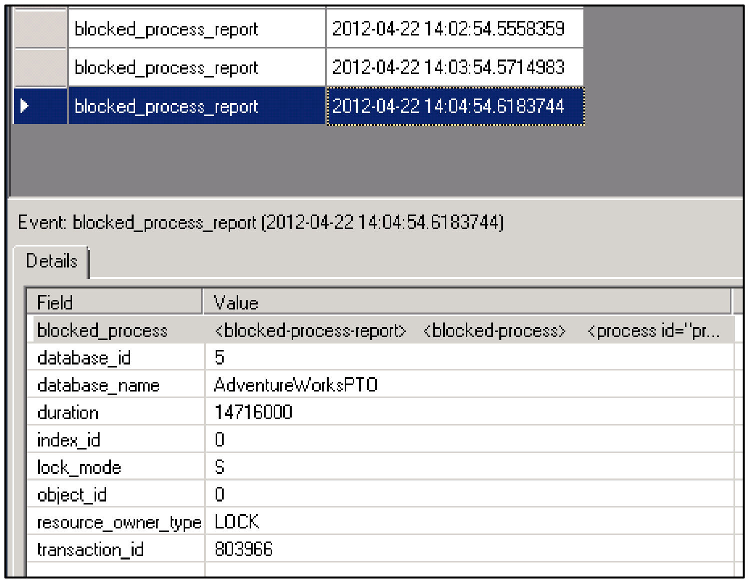
```
CREATE EVENT SESSION [BlockedProcessReport] ON SERVER
ADD EVENT sqlserver.blocked_process_report
ADD TARGET package0.ring_buffer(SET max_events_limit=(0),max_memory=(2048))
WITH (MAX_MEMORY=4096
KB,EVENT_RETENTION_MODE=ALLOW_SINGLE_EVENT_LOSS,MAX_DISPATCH_LATENCY=30
SECONDS,MAX_EVENT_SIZE=0
KB,MEMORY_PARTITION_MODE=NONE,TRACK_CAUSALITY=OFF,STARTUP_STATE=OFF)
GO

ALTER EVENT SESSION [BlockedProcessReport]
ON SERVER
STATE=START
GO
```

7. In the Management Studio Object Explorer, browse to Management -> Extended Events -> Sessions. If you do not see the BlockedProcessReport session, right-click Sessions and click “Refresh.” Once you see the BlockedProcessReport session, right-click it and click “Watch Live Data.”



8. Wait some time to see if there is blocking event captured in the Ring Buffer. Once you see an event, double-click the blocked_process row in the bottom pane to view the XML Blocked Process Report



	blocked_process_report	2012-04-22 14:02:54.5558359
	blocked_process_report	2012-04-22 14:03:54.5714983
▶	blocked_process_report	2012-04-22 14:04:54.6183744

Event: blocked_process_report (2012-04-22 14:04:54.6183744)

Details

Field	Value
blocked_process	<blocked-process-report> <blocked-process> <process id="pr...
database_id	5
database_name	AdventureWorksPTO
duration	14716000
index_id	0
lock_mode	S
object_id	0
resource_owner_type	LOCK
transaction_id	803966

Question D: What is wait resource of blocked process? Can you identify the object from it (think back to what you learned about table structure in Module 2)

Question E: What are the statements of the blocked process and blocking process? Think about how to resolve the blocking scenario:

---

9. Close the Extended Event Live Data window and any other windows you have open in Management Studio. Right-click the BlockedProcessReport Extended Events session in the Object Explorer and click “Stop Session.” You can also Delete the session at this point, or keep it around for future use. Close all the open Command Windows to stop all the database activity.

Exercise 2: Identifying deadlocks

Objectives

After this exercise, you will be able to:

- Observe deadlock in SQL error log
- Analyze deadlock from deadlock graph extended event collected by the system_health session
- Solve common deadlock issue by adjust code logic

Scenario

A developer found database 1205 error logged in application log. We need to find the deadlock issue and suggest how to improve it.

Troubleshoot deadlock

1. Run below statement to force 1205 error be logged in SQL errorlog and Windows application log:

```
sp_altermessage 1205, 'WITH_LOG', 'true';
```

2. Use traceflag 1222 to find deadlock scenario. The sample code as below:

```
dbcc traceon(1222,-1)
go
```

3. Double-click the GenerateDeadlock.cmd batch file from the C:\Labs\Module3\Exercise2 folder. This action will call 2 vbs scripts to generate a deadlock. Wait until you receive a popup window with the deadlock error message before you proceed to the next step.

4. View the error log by running sp_readerrorlog from a query window or by browsing to the log from the Object Explorer (Management -> SQL Server Logs -> Current). Search for the traceflag 1222 result, it should be at the end of the log.

   Question A: What lock resource was requested and owned by deadlock participating sessions?

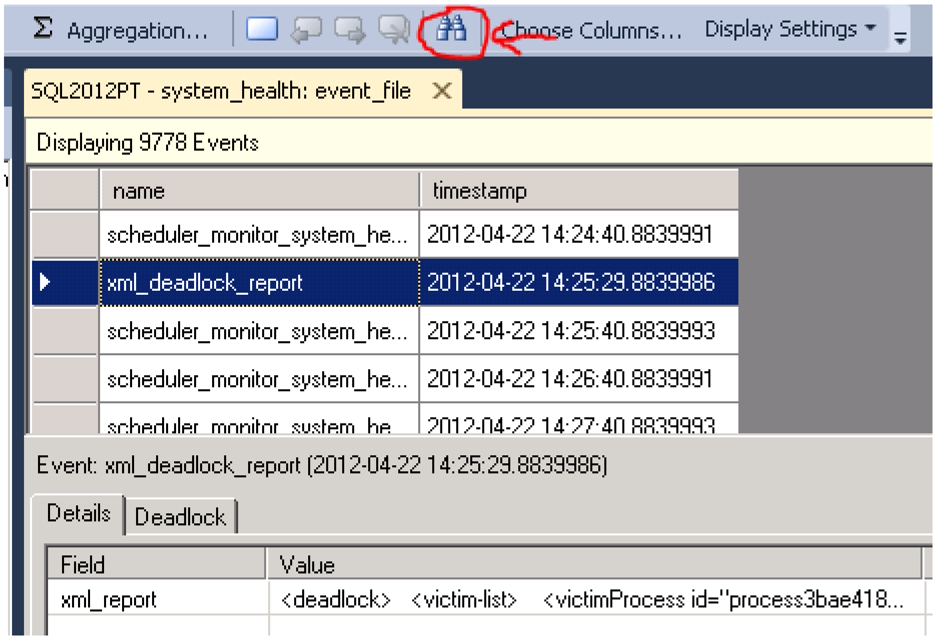
   ____

   ____

5. From Management Studio, browse to the system_health Extended Event session in the Object Explorer, Management -> Extended Events -> Sessions -> system_health. Expand the session, right-click on package0.event_file and choose “View Target

Data…" This will open a window that displays the events collected so far by the session, which should include the same deadlock event you just viewed in the error log.

6. Search for the deadlock events by clicking the "Find in extended events" button on the toolbar (it looks like binoculars), then searching for "deadlock" in the Table Columns until you find a row with the name "xml_deadlock_report." You should see something similar to the picture below:



name	timestamp
scheduler_monitor_system_he…	2012-04-22 14:24:40.8839991
xml_deadlock_report	2012-04-22 14:25:29.8839986
scheduler_monitor_system_he…	2012-04-22 14:25:40.8839993
scheduler_monitor_system_he…	2012-04-22 14:26:40.8839991
scheduler_monitor_system_he…	2012-04-22 14:27:40.8839993

Event: xml_deadlock_report (2012-04-22 14:25:29.8839986)

Field	Value
xml_report	<deadlock> <victim-list> <victimProcess id="process3bae418…

7. Click the "Deadlock" tab in the bottom pane to view the deadlock graph. You can hover over the bubbles to view the statements involved in the deadlock.

Question B: Which session has been chosen as victim?

Question C: Can you find the reason of deadlock from graphical interface? Think about why.

8. Go back to the “Details” tab on the bottom pane of the Extended Event window and double-click the xml_report row. This will display the full XML of the report. This XML can be saved as an .XDL file which you can open later in Management Studio.

Question D: Looking at the full detail of the report, can you find the real statement causing the deadlock? Where is the statement coming from?

9. In the customer’s scenario, we have to process the sales order data according to these queries, think about how to avoid this deadlock by modifying the code logic.
10. Close the Extended Event target data window and any other windows you have open in Management Studio.
11. Close all command windows opened in last 2 exercises.
12. Open Task Manager, end the process whose image name is “wscript”. That’s the vbs script that is running in the background generating deadlocks.

Exercise 3: Read committed snapshot

Objectives

After this exercise, you will be able to:

- Illustrate how snapshot isolation level effect blocking and locking

Scenario

Customer wants to resolve blocking issue as soon as possible while their developer can't modify code in short period.

Apply read committed snapshot

1. Be sure you have cleaned up from all previous exercises and that there are no more open connections to the server.
2. Enable the READ_COMMITTED_SNAPSHOT database option on the AdventureWorksPTO database by opening a new query window and executing the following statement :

```
USE master
GO
ALTER DATABASE AdventureWorksPTO
SET READ_COMMITTED_SNAPSHOT ON;
```

   Make sure all other query windows have been closed before running this statement.
3. Repeat steps 2 and 3 in exercise 1 (run GenerateActivity.cmd, then ViewBlockingDMVs.sql in SSMS).

Question A: Do you see any blocking? Try to explain how read committed snapshot affects locking and blocking.

____

____

4. Repeat step 3 in exercise 2 (run GenerateDeadlock.cmd) and wait a few minutes.

Question B: Do you still see deadlocks? Why or why not?

____

5. Clean up as in previous exercises by closing all command windows, stopping all “wscript.exe” sessions via Task Manager and closing all Management Studio windows.
6. Disable READ_COMMITTED_SNAPSHOT database option by below statement.

```
USE master
GO
ALTER DATABASE AdventureWorksPTO
SET READ_COMMITTED_SNAPSHOT OFF;
```