www.PragimTech.com

Pragim@PragimTech.com

SQL Server Tutorial

Part 79 - SQL Server deadlock victim selection

Venkat
PRAGIM Technologies
kudvenkat@gmail.com
http://csharp-video-tutorials.blogspot.com

In this session we will learn

- How SQL Server detects deadlocks
- What happens when a deadlock is detected
- What is DEADLOCK_PRIORITY
- What is the criteria that SQL Server uses to choose a deadlock victim when there is a deadlock

Link to Dot Net Basics, ASP.NET, C#, ADO.NET and SQL Server video series
http://www.youtube.com/user/kudvenkat/playlists

Suggested Videos
Part 76 - Read committed snapshot isolation level in sql server
Part 77 - Difference between snapshot isolation and read committed snapshot
Part 78 - SQL Server deadlock example

SQL Server deadlock victim selection

If you are in need of the DVD with all the videos and PPT's, please visit http://pragimtech.com/order.aspx

How SQL Server detects deadlocks

Lock monitor thread in SQL Server, runs every 5 seconds by default to detect if there are any deadlocks. If the lock monitor thread finds deadlocks, the deadlock detection interval will drop from 5 seconds to as low as 100 milliseconds depending on the frequency of deadlocks. If the lock monitor thread stops finding deadlocks, the Database Engine increases the intervals between searches to 5 seconds

What happens when a deadlock is detected

When a deadlock is detected, the Database Engine ends the deadlock by choosing one of the threads as the deadlock victim. The deadlock victim's transaction is then rolled back and returns a 1205 error to the application. Rolling back the transaction of the deadlock victim releases all locks held by that transaction. This allows the other transactions to become unblocked and move forward.

What is DEADLOCK_PRIORITY

What is DEADLOCK_PRIORITY

By default, SQL Server chooses a transaction as the deadlock victim that is least expensive to roll back. However, a user can specify the priority of sessions in a deadlock situation using the SET DEADLOCK_PRIORITY statement. The session with the lowest deadlock priority is chosen as the deadlock victim

Example :

```
SET DEADLOCK_PRIORITY NORMAL
```

DEADLOCK_PRIORITY

1. The default is Normal
2. Can be set to LOW, NORMAL, or HIGH
3. Can also be set to a integer value in the range of -10 to 10
   - LOW : -5
   - NORMAL : 0
   - HIGH : 5

Deadlock Victim Selection Criteria

What is the deadlock victim selection criteria

1. If the DEADLOCK_PRIORITY is different, the session with the lowest priority is selected as the victim

2. If both the sessions have the same priority, the transaction that is least expensive to rollback is selected as the victim

3. If both the sessions have the same deadlock priority and the same cost, a victim is chosen randomly

Deadlock Victim Selection - Example 1

```
-- Transaction 1
Begin Tran
Update TableA Set Name = Name + ' Transaction 1' where Id IN (1, 2, 3, 4, 5)

-- From Transaction 2 window execute the first update statement

Update TableB Set Name = Name + ' Transaction 1' where Id = 1

-- From Transaction 2 window execute the second update statement
Commit Transaction
```

```
-- Transaction 2
Begin Tran
Update TableB Set Name = Name + ' Transaction 2' where Id = 1

-- From Transaction 1 window execute the second update statement

Update TableA Set Name = Name + ' Transaction 2' where Id IN (1, 2, 3, 4, 5)

-- After a few seconds notice that this transaction will be chosen as the deadlock
-- victim as it is less expensive to rollback this transaction than Transaction 1
Commit Transaction
```

Deadlock Victim Selection - Example 2

```
-- Transaction 1
Begin Tran
Update TableA Set Name = Name + ' Transaction 1' where Id IN (1, 2, 3, 4, 5)

-- From Transaction 2 window execute the first update statement

Update TableB Set Name = Name + ' Transaction 1' where Id = 1

-- From Transaction 2 window execute the second update statement
Commit Transaction
```

```
-- Transaction 2
SET DEADLOCK_PRIORITY HIGH
GO
Begin Tran
Update TableB Set Name = Name + ' Transaction 2' where Id = 1

-- From Transaction 1 window execute the second update statement

Update TableA Set Name = Name + ' Transaction 2' where Id IN (1, 2, 3, 4, 5)

-- After a few seconds notice that Transaction 2 will be chosen as the deadlock
-- victim as it's DEADLOCK_PRIORITY (Normal) is lower than the DEADLOCK_PRIORITY
-- this transaction (HIGH)
Commit Transaction
```

Additional Resources

PRAGIM Home Page:

www.PragimTech.com

Resources:

C#, ADO.NET, ASP.NET, SQL Server & MVC youtube Playlists
http://www.youtube.com/user/kudvenkat/playlists

Code samples and text version of all the videos on my blog
http://www.csharp-video-tutorials.blogspot.com

To receive email alerts when new videos are uploaded, please subscribe to my YOUTUBE channel
www.YouTube.com/kudvenkat