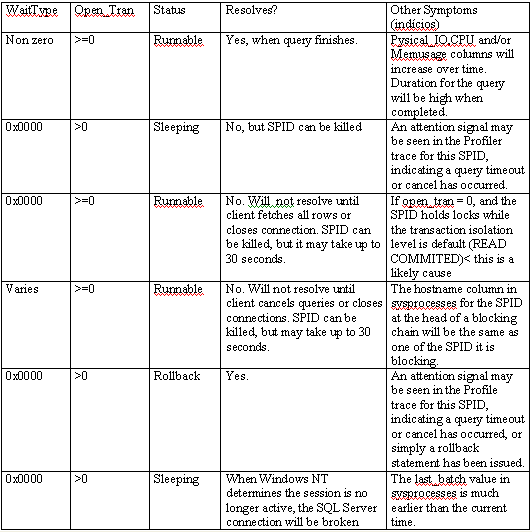
|  |  |  |
| --- | --- | --- |
| ***CheckList para identificar problemas de lentidão*** | | |
| **Passo** | **Comando** | **Descrição** |
| 1 | **sp\_who** | Serve para ter uma visão geral sobre as conexões existentes e verificar se existe alguma conexão bloqueada (Block). Coluna blk <>0. |

|  |  |  |
| --- | --- | --- |
| ***Sem Lock*** | | |
| **Passo** | **Comando** | **Descrição** |
| 1 | **sp\_who active** | Ter uma visão geral sobre as conexões ativas e verificar se existe alguma conexão bloqueada |
| 2 | **xp\_readerrorlog** | Verique o Error Log do SQL na busca de erros. Se for encontrado algum erro, pesquisar no Books On Line, Premier, TechNet, etc |
| 3 | **Verifique a situação do servidor** | Analizar os contadores do Performance Monitor (Contadores Padrao) |
| Obs: As stored procedures **sp\_usrheadblocker e sp\_usrinputbuffer** podem ser encontradas na seção de scripts do website: | | |

|  |  |  |
| --- | --- | --- |
| ***Com Lock*** | | |
| **Passo** | **Comando** | **Descrição** |
| 1 | **sp\_who active** | Serve para ter uma visão geral sobre as conexões ativas e verificar se existe alguma conexão bloqueada |
| 2 | **sp\_usrheadblocker** | Serve para identificar a conexão que está bloqueando e as conexões que estão bloqueadas. A conexão causadora do bloqueio vem sempre na primeira linha e a coluna RunAs, mostra a quanto tempo (minuto) a conexão está em execução. |
| 4 | **Select \* from sysprocesses/** SELECT \* FROM master.sys.sysprocesses | Serve para verificar o status da conexão (waittype) de acordo com a tabela padrão da Microsoft |
| 5 | **dbcc inputbuffer (<spid>)** | Serve para verificar o comando que está sendo executado |
| 5.1 | ***sp\_usrinputbuffer <spid>*** | Se for um processo que ainda esteja ativo, identifica o ponto exato onde o bloqueio está sendo realizado |
| 6 | **Kill <spid>** | Após permissão do analista responsável, derrubar a conexão que está bloqueando as demais. Se ficar em Killed/Rollback por muito tempo, seguir os passo 6.1 e 6.2 |
| 6.1 | ***alter database <banco de dados> set restricted\_user with rollback immediate*** | Se ficar em status de killed/rollback por muito tempo |
| 6.2 | ***alter database <banco de dados> set multi\_user*** |
| 7 | **xp\_readerrorlog** | Verique o Error Log do SQL na busca de erros. Se for encontrado algum erro, pesquisar no Books On Line, Premier, TechNet, etc |
| 3 | **Verifique a situação do servidor** | Analizar os contadores do Performance Monitor.  (contadores padrão) |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Descrição dos Contadores*** | | | | | | | | |
| ***Memória:*** | | | | | | | | |
| ***Contador*** | ***Descrição*** | | | | | | | |
| ***Memory: Available Bytes*** | Indica a quantidade de memória disponível | | | | | | | |
| ***Memory: Pages/Sec*** | Indica se está havendo paginação. Em perfeitas condições deve-se ver pouca atividade neste contador. | | | | | | | |
| ***SQL Server:*** | | | | | | | | |
| ***Contador*** | | | | ***Descrição*** | | | | |
| ***SQL Server: Buffer Manager: Buffer Cache Hit Ratio*** | | | | Indica o percentual de requisições a dados que são obtidas no próprio ***cache do sql*** e portanto, sem precisar acessar o disco. O ideal é estar maior que 90% | | | | |
| ***SQL Server:Access Methods:Page Splits/sec*** | | | | Mostra quantos page splits estão ocorrendo no servidor. Este valor deve ser o mais baixo possível. Se o valor estiver alto, configurar os índices com um fillfactor apropriado pode ajudar a reduzir este valor. | | | | |
| ***SQL Server: Memory Manager: Total Server Memory (KB)*** | | | | Quando maior que a quantidade de memória física pode indicar falta de memória. | | | | |
| ***CPU:*** | | | | | | | | |
| ***Contador*** | | | ***Descrição*** | | | | | |
| ***Processor:% Processor Time:\_Total*** | | | Indica o consumo de CPU no servidor (ideal abaixo de 80%). | | | | | |
| ***Process: % Processor time: sqlservr*** | | | Indica o consumo do processador pelo processo do SQL Server | | | | | |
| ***System: Processor Queue Length*** | | | Indica o número de threads aguardando para execução no processador e **nunca** deve exceder 1 ou 2 (por processador) por um período superior a 10 minutos. | | | | | |
| ***DISK I/O:*** | | | | | | | | |
| ***Contador*** | ***Descrição*** | | | | | | | |
| ***PhysicalDisk: % idle Time*** | Indica o percentual de tempo que o disco esta ocioso. Supostamente este contador apresenta uma estimativa mais precisa sobre a utilização dos discos. Subtraindo o valor encontrado de 100, temos uma ideia do quanto o disco esta trabalhando. | | | | | | | |
| ***PhysicalDisk: % Disk Time*** | Mostra o quanto o disco esta ocupado.Se valor > 60% durante 10 minutos, verificar os contadores ***PhysicalDisk: % Disk Read Time*** e ***PhysicalDisk: % Disk Write Time*** para verificar se é leitura ou escrita quem está causando maior utilização**.** | | | | | | | |
| ***PhysicalDisk: Avg. Disk  Queue Length*** | Mostra o número médio de requisições de I/O aguardando para acesso ao disco. Este contador nunca deve exceder 2 (por disco) por um extenso período de tempo. | | | | | | | |
| ***PhysicalDisk: Avg. Disk  Sec/Transfer*** | Taxa de transferência em bytes de ou para um disco durante operações de leitura ou escrita:  ·        Cache < 5Ms  ·        Excelente < 15Ms  ·        Bom < 30Ms  ·        Regular < 60Ms  ·        Ruim < 90Ms | | | | | | | |
| ***Banco de Dados:*** | | | | |  |  |  |  |
| ***Contador*** | | ***Descrição*** | | | | | | |
| ***SQL Server: Access Methods:  Full Scans/sec*** | | Mostra a quantidade de Table Scan sendo executada no SQL Server. ***Este valor deve ser o mais baixo possível*** e se for constantemente alto, use o Profile para identificar quais queries estão causando os scans. | | | | | | |
| ***SQL Server: Locks:  Average Wait Time (ms)*** | | Este contador mostra quanto tempo em milessegundos um processo esta aguardando para obter lock no SQL Server. ***O valor ideal para este contador é zero***. Se este valor estiver muito alto, utilize o Profile para identificar quais processos estão mantendo locks por um período longo de tempo. | | | | | | |

Tabela:



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* sp\_usrheadblocker \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if exists (select \* from dbo.sysobjects where id = object\_id(N'[dbo].[sp\_usrheadblocker]') and OBJECTPROPERTY(id, N'IsProcedure') = 1)

drop procedure [dbo].[sp\_usrheadblocker]

GO

Create Proc dbo.sp\_usrheadblocker

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Procedure: sp\_usrheadblocker

Descrição:Possui a característica de reunir em apenas um comando as instruções: sp\_who Active

e DBCC InputBuffer(n). Retorna apenas as linhas que estão com Status de Blocked <> 0 e no

final de cada linha traz o InputBuffer.

A primeira linha sempre será a ponta (head blocker-a conexão que está causando o bloqueio de

todas as outras) e as demais linhas, as que estão aguardando a liberação do recurso.

Pode-se ver a quanto tempo cada uma das conexões está aguardando a liberação.

Autor Original: Marcelo Andretto

Adaptada por : Nilton Pinheiro

WebSite: http://www.mcdbabrasil.com.br

-- Melhor visualizada em modo ..GRADE..

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

as

Set NoCount on

-- Tabelas de Apoio.

CREATE TABLE #tbheadBlocked (

[Host\_Id] [int] NULL ,

[SPID] [int] NULL ,

[a] [varchar] (14) NULL ,

[b] [int] NULL ,

[TextBuffer] [varchar] (255) NULL

)

Create Table #tbInputBuffer(

a VarChar(14),

b int,

TextBuffer VarChar(255))

-- Passo1 -- Insere na tabela temporária "#tbheadblocked" todos os SPID que estão bloqueados

Insert Into #tbheadBlocked Select Host\_id(),SPID,null,null,null

from master..sysprocesses (NoLock)

Where SPID in (Select Blocked from master..sysprocesses (NoLock)

Where Blocked <>0)or Blocked <> 0

-- Passo 2 -- Abre um cursor para obter o DBCC InputBuffer de todas os SPID que foram inseridas

-- na tabela do passo 1 e armazena em uma nova tabela temporária "#tbInputBuffer"

Declare @SPID Int

Declare C\_Buffer CURSOR For Select SPID from #tbheadBlocked

Open C\_Buffer

Fetch C\_Buffer Into @SPID

While @@Fetch\_Status = 0

Begin

Insert Into #tbInputBuffer exec ('Dbcc InputBuffer(' + @SPID + ') with NO\_INFOMSGS ')

Update #tbheadBlocked Set TextBuffer = #tbInputBuffer.TextBuffer from #tbInputBuffer Where Spid = @spid

Fetch C\_Buffer Into @SPID

End

Close C\_Buffer

Deallocate C\_Buffer

-- Passo 3 -- Faz o Join das tabelas temporárias apresentando o resultado final.

-- Uma concatenação da SP\_Who Active + DBCC InputBuffer()

select distinct a.SPID,

a.Blocked,

a.ECID,

a.WaitTime as WaitTimeMS,

datediff (mi,a.last\_batch,getDate() ) as RunAs, --Tempo de execução em minutos

SubString(a.Status,1,10) as Status,

a.CPU,

SubString(Cast(a.Physical\_IO as Varchar(10)),1,10) as Physical\_IO,

SubString(a.HostName,1,15) as HostName,

SubString(a.LogiName,1,15) as LoginName,

SubString(DB\_Name(a.dbid),1,13) as DBName,

SubString(convert(VarChar(24),a.last\_batch ,113),1,24) as Last\_Batch,

a.open\_tran,

a.MemUsage,

b.TextBuffer

from master..sysprocesses a (NoLock) Right Outer Join #tbheadBlocked b (NoLock)

On a.Spid = b.spid

Where a.SPID in (Select c.Blocked from master..sysprocesses c (NoLock)Where Blocked <>0)

or a.Blocked <> 0

Order By a.Blocked

Set NoCount off

Drop Table #tbInputBuffer

Drop table #tbheadBlocked

GO

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* sp\_usrinputbuffer \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

-- Procedure para identificar o ponto exato de execução de um statement

USE MASTER

GO

if exists (select \* from dbo.sysobjects where id = object\_id(N'[dbo].[sp\_usrinputbuffer]') and OBJECTPROPERTY(id, N'IsProcedure') = 1)

drop procedure [dbo].[sp\_usrinputbuffer]

GO

CREATE PROCEDURE dbo.sp\_usrinputbuffer

(

@SPID smallint, -- O SPID a ser analizado

@WAIT tinyint = 1, -- Intervalo de repetição entre as verificações.Valor deve estar entre 0 e 60 segundos

-- Default é de 1 em 1 segundo

@NoLoop bit = 1 -- Se =1, o SPID é analizado apenas uma vez. Se =0 fica em loop até o final do processo.

)

AS

BEGIN

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

sp\_usrinputbuffer: Esta procedure permite identificar a instrução exata que está sendo

executado por uma dada conexão. Semelhante ao DBCC INPUTBUFFER mas diferente desta, em caso

de procedure, a proc mostra toda a instrução inteira e não apenas o nome da procedure. Muito boa

para ser usada em conjunto com a procedure sp\_usrheadblocker.

Exemplo: sp\_usrinputbuffer 54 (Onde 54 é o ID da conexão SPID)

Obs: Se preferir, esta pode ser criada em qualquer banco, porém, ao executá-la deverá passar o nome da

base onde a mesma se encontra. Exemplo: Base..sp\_usrinputbuffer 54

Author Original: Narayana Vyas Kondreddi

Source: http://vyaskn.tripod.com

Date Created: 18/12/2003

Alterada por : Nilton Pinheiro

WebSite: http://www.mcdbabrasil.com.br

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

SET NOCOUNT ON

DECLARE @sql\_handle binary(20), @handle\_found bit

DECLARE @stmt\_start int, @stmt\_end int

DECLARE @line varchar(8000), @wait\_str varchar(8)

SET @handle\_found = 0

IF @WAIT NOT BETWEEN 0 AND 60

BEGIN

RAISERROR('Valores válidos para @WAIT estão entre 0 e 60 segundos', 16, 1)

RETURN -1

END

ELSE

BEGIN

SET @wait\_str = '00:00:' + RIGHT('00' + CAST(@WAIT AS varchar(2)), 2)

END

WHILE 1 = 1

BEGIN

SELECT @sql\_handle = sql\_handle,

@stmt\_start = stmt\_start/2,

@stmt\_end = CASE WHEN stmt\_end = -1 THEN -1 ELSE stmt\_end/2 END

FROM master.dbo.sysprocesses

WHERE spid = @SPID

AND ecid = 0

IF @sql\_handle = 0x0

BEGIN

IF @handle\_found = 0

BEGIN

RAISERROR('Não pode encontrar o handle ou o SPID é inválido', 16, 1)

RETURN -1

END

ELSE

BEGIN

RAISERROR('Query/Stored procedure completada', 0, 1)

RETURN 0

END

END

ELSE

BEGIN

SET @handle\_found = 1

END

Print '\*\*\*\*\*\*\*\* STATEMENT SENDO EXECUTADO NO MOMENTO \*\*\*\*\*\*\*\*\*\*\*\*'

Print ''

SET @line =

(

SELECT

SUBSTRING( text,

COALESCE(NULLIF(@stmt\_start, 0), 1),

CASE @stmt\_end

WHEN -1

THEN DATALENGTH(text)

ELSE

(@stmt\_end - @stmt\_start)

END

)

FROM ::fn\_get\_sql(@sql\_handle)

)

Print @line

IF @NoLoop = 1

BEGIN

RETURN 0

END

WAITFOR DELAY @wait\_str

END

END

GO