oracle_br

- Por tôpico
- Por data
- Pesquisar
- 0

[oracle_br] Re: Library Cache Pin - Ajustando Shared_Pool

JosÃ© Laurindo
Wed, 08 Jun 2011 16:38:16 -0700

ammorrimm , vou aproveitar o seu e-mail e dar umas dicas, que aí já serve não só pra vc como pra outros que estejam com dúvidas similares. Primeiro, quando vc pede por um "procedimento", por um "aprendizado" de como proceder em relação à latches, eu vou repetir algo que já disse muitas vezes mas continua Verdadeiro a mais não poder, que é : SEJA o que for no rdbms Oracle que vc quer tunar, o setp ZERO, o pré-requisito, é conhecer os Conceitos do tópico em questão, okdoc ? Com isso em mãos E com o conhecimento da Aplicação que vc quer melhorar (tudo que vc puder saber dela : como ela se conecta no banco, se é OLTP ou BATCH, quantos usuários simultãneos, se há objetos mais usados que outros, como os objetos são usados tipicamente, como se faz limpeza de dados, como os dados entram, retenção, enfim, TUDO o que puder) aí sim vc pode começar a avaliar as opções...
Vamos então dar um vapt-vupt sobre LATCHES em primeiro lugar : a documentação (e no metalink as notas What are Latches and What Causes Latch Contention (Doc ID 22908.1) - esta uma Crucial no assunto imho - além da Solutions for possible AWR Library Cache Latch Contention Issues in Oracle 10g (Doc ID 296765.1) e a Understanding and Tuning the Shared Pool and Tuning Library Cache Latch Contention (Doc ID 62143.1) aprofundam as refs e dão excelentes sugestões e definições mas vamos dar um geralzinha. No caso vamos falar de bloco, que é o recurso mais comum protegido por latch, E ainda há pontos como o tipo de request do latch (se willing-to-waiting ou não), há bem mais aí nesse angú, mas vamos falar mais basicamente...

Seguinte, como o rdbms Oracle é multi-usuário (foi PENSADO para permitir múltiplas sessões simultâneas E com integridade - nada de leitura suja, de dados que podem ter sido alterados !!) , é absolutamente crítico que nos milisegundos enquanto um bloco estiver sendo lido ninguém o altere (quem quiser alterar TEM que esperar a leitura acabar pra não alterar o que está sendo lido, E que nos milisegundos enquanto o bloco estiver sendo gravado quem quiser ler o bloco tem que esperar a gravação acabar : o mecanismo que o rdbms Oracle implementa pra isso é o LATCH, que nada mais é do que um FLAG, um Indicador em memória (uma variável se vc quiser pensar assim) que quando vazia indica que o recurso está livre, quando não vazia indica que está em uso - e isso (com as execções óbvias, tipo GTT) vale ******* PARA PRATICAMENTE QUALQUER I/O ********, seja para I/O de bloco vindo do disco, seja pra I/O de bloco já em cache... Assim, TODAS as sub-rotinas no rdbms Oracle foram programadas de modo que antes que um I/O seja feito num bloco qquer, pesquisa-se se o LATCH que aponta para/protege o recurso tá disponível, se estiver vazio, disponível, a sessão obtém o latch (marca-o como Em uso) e aí sim faz o I/O , e enquanto isso todas as sessões que quiserem usar esse bloco nesses milisegundos em que o bloco está em uso pot outrem vão tentar obter o latch (marcá-lo com em uso),

não vão conseguir , aí vão entrar no que se chama SPIN : vão "dormir" por uns milisegundos, tentam de novo, se ainda não tá livre o latch vão dormir de novo, assim por diante, por um número pré-definido de vezes (normalmente entre 10 a 16 vezes , isso varia de acordo com a versão do banco, e pode ser alterado via params ocultos) : e se após passar por todos os SPINS possíveis ainda assim a sessão não obteve o latch, aí ela passa pra espera Passiva mesmo, ie : fica pendurada esperando até o latch ser liberado .

Muito bem : com esses pontos em vista algumas conclusões devem começar a cair nos seus lugares. Por exemplo, sobre aumento de cache - a idéia do cache é que o I/O seja feito da RAM ao invés do disco, o que via de regra é muito mais rápido, aí a sessão que tem o latch mais rapidamente o libera pros outros usar .... A teoria é boa, MAS sabemos nós que a teoria de caching é implacável : caches não são mágicos, eles funcionam muito bem SE vc está lendo a mesma informação repetidas vezes : SE, como é comum por exemplo em aplicações DW, vc faz scans frequentes, a cada vez vc normalmente está acessando dados diferentes , Com Certeza nesse ambiente o cache não vai te comprar lá grande coisa.... Sacou porque realmente DEPENDE MUITO do seu ambiente, do SEU caso particular se vai ser efetivo ou não vc aumentar caches ?/ Então bote Sempre um grão de sal nessas recomendações genéricas, okdoc ?? E TESTE, TESE e TESTE, sempre...

Segundo : como eu falei e REPITO, praticamente TODOS os I/Os (e acessos a outros recursos, mas I/O principalmente é o caso + comum) Necessariamente Tem que ser protegido por latch, Então temos que :

a. óbvio ululante, para diminuir latches, dimunua os I/Os, simples assim .... Isso se faz com Tuning de SQL (alterando-se o SQL pra ser o mais eficiente possível, retornando os dados com o menor número de blocos acessados), E/OU com Tuning de estruturas físicas (assegurando-se que onde possível/viável vc usa Particionamento, clustering, índices do tipo adequado - até índice seletivo se for o caso - , assegurando-se que vc não tem nenhum dos problemas de I/O comuns no bd Oracle como white-space, HWM alto, bloco com espaço reservado/não-usado à toa, etc, etc) , E/OU (outra Obviedade) NÂO TENDO HOT BLOCKs : se a tua aplicação é daquelas 'doentinhas' aonde há uma tabela de controle/auditoria/whatever que todo mundo quer ler/acessar/gravar a toda hora, simplesmente vão ser Inevitáveis hot blocks...

b. já que algum latch é Inevitável, vc NECESSARIAMENTE TEM QUE analisar também o quanto de latch wait vc está tendo EM RELAÇÃO AO TOTAL de waits do programa que vc está tunando, okdoc ? De repente vc pode ter tido, sei lá, dez mil latches MAS o tempo de espera por liberação de latches no total pra essa sessão/programa é comparativamente pequeno, num caso desse vc Com Certeza vai ter peixes maiores pra fritar.... Pra vc obter Tempos de Waits dum programa/sessão, é aquilo de sempre : trace + tkprof , OU consulta às V$ de eventos/waits (é o que o AWR e outros recursos do banco 10g em diante fazem automaticamente pra vc)... O ponto é, NÂO faça trabalho pela metade, além de ver se há latches, DESCUBRA EXATAMENTE O QUANTO os latches estão interferindo no programa que vc está tunando, okdoc ? E eu disse NO PROGRAMA, pois análise geral e genérica, a nível de banco de dados como um todo, é Extremamente difícil, pois as estats de performance do banco são GERAIS, valem pra todas as sessões, e como eu disse qquer acesso a nível de banco usa latches, então vc acaba tenbdo resultados de sessões onde o latch não foi significativo misturado com outras, é o tipo de análise que Dificilmente dá certo... faça POR PARTES, analise os PROGRAMAS MAIS IMPORTANTES primeiro, antes de sair querendo fazer tuning de banco em geral...

Finalmente, falando especificamente sobre o latch de Library Cache : cfrme nós sabemos, o Oracle guarda os SQLs e os planos de execução em tabelas internas (que vão pra memória em caches específicos) e em estruturas de memória - é a Exata mesma história de sempre, pra ter consistência de leitura sempre que alguém quer gravar nesses caras ele Tem que se assegurar que ninguém está lendo, E quando se for ler Tem que se assegurar que ninbguém está gravando, então esse LATCH citado (entre outros) é usado.... Sim, é verdade que quando o TEXTO de um SQL recém-enviado para o banco de dados NÂO é (byte-por-byte!!)

idêntico aos textos dos outros SQLs já presentes vai ser feito um PARSE, e o resultado desse parse será inserido nas estruturas, então sim, uma aplicação que faça PARSE a toda hora estará a toda hora gravando coisas, portanto (já que a gravação via de regra demora mais que a leitura) ocuparão o latch mais frequentemente e por mais tempo... Solução é a mesma que para os outros latches, ie , DIMINUA OS I/Os - no caso, re-aproveitando um cursor SQL várias vezes antes de fechá-lo (principalmente via params de cache de cursor E tendo uma aplicação decente que não força PARSE, como acontece em ODBC por exemplo) , evite o REPARSE por diferenças de SQL (asseguranbdo que os textos estão em maiúsculas/minúsculas de mesmo modo, usando BIND VARIABLES, enfim, as técnicas-padrão de re-aproveitamento de SQL...
Aí a sua última pergunta : aumento de cache (SHARED POOL no caso) resolve automagicamente latch contention ??? resposta, a mesma de cima, ie : PODE AJUDAR um pouco, mas apenas SE e apenas SE vc acessa os mesmos dados repetidamente - não há mágica, cache é isso : no caso de SQLs, se vc re-usa os SQLs...

[]s

Chiappa

OBS : NÂO deixe de checar no metalink, pois houve diversos bugs de latch relacionados a SQLs internos (sim, os trabalhos internos do rdbms Oracle ** são ** sim feitos em SQLs, e os programadores da Oracle não são super-humanos, podem SIM errar, não usar bind, não reaproveitar, E mesmo os programadores C do kernel do banco na hora de criar as "variáveis" que são os latches.... Alguns exemplos foram : Bug 7122093 - 'latch: library cache' contention caused by queries on V$ views. (Doc ID 7122093.8) e Bug 4339128 - Heavy latch contention from queries against library cache views (Doc ID 4339128.8) . mas outros Existiram/Existem, não deixe de os verificar...

--- Em oracle_br@yahoogrupos.com.br, "ammorrimm" <ammorrimm@...> escreveu
>
> Amigos,
>
> Estou iniciando algumas analise na minha Shared_Poll, mais precisamente na
> Library Cache pois tenho verificado alguns pequenos latchs e gostaria de uma
> ajuda...
>
> Pelo que entendo e andei lendo, este lacth esta relacionado a execução ou
> compilação de PL / SQL e a interpretação destes, ficam armazenados nesta área
> de memória.
>
> Qual seria a melhor maneira de analisar esta questão e saber quais são os
> objetos que precisariam ser revistos ?
>
> Este latch esta sempre relacionado a performance do objetos a nível de parse?
> Ou seja, quanto mais reparsing pior será o cenário ?
>

- [oracle_br] Library Cache Pin - Ajustando Shared_Poolammorrimm
  - Re: [oracle_br] Library Cache Pin - Ajustando Shared_PoolRafael Trevisan
    - Re: [oracle_br] Re: Library Cache Pin - Ajustando Shared_PoolHevandro Veiga
  - [oracle_br] Re: Library Cache Pin - Ajustando Shared_PoolJosÃ© Laurindo
  - Re: [oracle_br] Library Cache Pin - Ajustando Shared_PoolPaulo H. P. da Silva

- Responder a
  JosÃ© Laurindo