Artigo
David Loveluck · Dez. 9, 2020 26min de leitura

# APM - Monitorando o Desempenho da Consulta SQL

Desde o Caché 2017, o mecanismo SQL inclui um novo conjunto de estatísticas. Ele registra o número de vezes que uma consulta é executada e o tempo que leva para executá-la.

Esta é uma mina de ouro para qualquer pessoa que está monitorando e tentando otimizar o desempenho de uma aplicação que inclui muitas instruções SQL, mas que não é tão fácil de acessar os dados como algumas pessoas desejam.

Este artigo e o código de exemplo associado explicam como usar essas informações e como extrair rotineiramente um resumo de estatísticas diárias e manter um registro histórico de desempenho do SQL de sua aplicação.

## O que está registrado?

Toda vez que uma instrução SQL é executada, o tempo gasto é registrado. Esse registro é muito leve e você não pode desativá-lo. Para minimizar o custo, as estatísticas são mantidas na memória e gravadas no disco periodicamente. Os dados incluem o número de vezes que uma consulta foi executada no dia e a média e o tempo total gasto.

Os dados não são gravados no disco imediatamente, e depois de gravados, as estatísticas são atualizadas pela tarefa "Atualizar estatísticas de consulta SQL" que geralmente é programada para se executada a cada hora. Essa tarefa pode ser executada manualmente, mas se você quiser ver as estatísticas em tempo real enquanto testa uma consulta, todo o processo requer um pouco de paciência.

Aviso: No InterSystems IRIS 2019 e anteriores, essas estatísticas não são coletadas para SQL embutido em **classes ou rotinas** que foram **implantadas** usando o mecanismo **%Studio.Project:Deploy**. Nada quebrará com o código de exemplo, mas pode enganar você (me enganou) fazendo-o pensar que tudo estava OK porque nada parecia tão custoso.

## Como você vê as informações?

Você pode ver a lista de consultas no portal de gerenciamento. Vá para a página SQL e clique na aba ‘ Instruções SQL ’ . É ótimo para uma nova consulta que você está executando e examinando, mas se houver milhares de consultas em execução, pode-se tornar incontrolável.

A alternativa é usar o SQL para pesquisar as consultas. As informações são armazenadas em tabelas no esquema INFORMATIONSCHEMA. Este esquema tem várias tabelas e incluí alguns exemplos de consultas SQL no final deste artigo.

## Quando as estatísticas são removidas?

Os dados de uma consulta são removidos sempre que ela é recompilada. Portanto, para consultas dinâmicas, isso pode significar quando as consultas em cache são limpas. Para SQL embutido, isso significa quando a classe ou rotina em que o SQL está embutido é recompilado.

Em um site ativo, é razoável esperar que as estatísticas sejam mantidas por mais de um dia, mas as tabelas que contêm as estatísticas não podem ser usadas como uma fonte de referência para relatórios ou análises de longo prazo.

## Como você pode resumir as informações?

Recomendo extrair os dados todas as noites em tabelas permanentes que são mais fáceis de trabalhar ao gerar os relatórios de desempenho. Há uma chance de que algumas informações sejam perdidas se as classes forem compiladas durante o dia, mas é improvável que isso faça alguma diferença real em sua análise de consultas lentas.

O código a seguir é um exemplo de como você pode extrair as estatísticas em um resumo diário para cada consulta. Inclui três classes curtas:

* Uma tarefa que deve ser executada todas as noites.
* DRL.MonitorSQL é a classe principal que extrai e armazena os dados das tabelas INFORMATIONSCHEMA.

- A terceira classe DRL.MonitorSQLText é uma otimização que armazena o texto da consulta (potencialmente longo) uma vez e armazena apenas o hash da consulta nas estatísticas de cada dia.

Notas sobre o exemplo

A tarefa é extraída do dia anterior e, portanto, deve ser agendada logo após a meia-noite.

Você pode exportar mais dados históricos, se houver. Para extrair os últimos 120 dias

Faça ##class(DRL.MonitorSQL).Capture($h-120,$h-1)

O código de exemplo lê o ^rIndex global diretamente porque as versões mais antigas das estatísticas não expunham a Data ao SQL.

A variação, incluí loops em todos os namespaces na instância, mas isso nem sempre é apropriado.

## Como consultar os dados extraídos

Uma vez que os dados são extraídos, você pode encontrar as consultas mais pesadas executando

```
SELECT top 20
S.RunDate,S.RoutineName,S.TotalHits,S.SumpTIme,S.Hash,t.QueryText
from DRL.MonitorSQL S
left join DRL.MonitorSQLText T on S.Hash=T.Hash
where RunDate='08/25/2019'
order by SumpTime desc
```

Além disso, se você escolher o hash para uma consulta custosa, poderá ver o histórico dessa consulta com

```
SELECT S.RunDate,S.RoutineName,S.TotalHits,S.SumpTIme,S.Hash,t.QueryText
from DRL.MonitorSQL S
left join DRL.MonitorSQLText T on S.Hash=T.Hash
where S.Hash='CgOlfRw7pGL4tYbiijYznQ84kmQ='
order by RunDate
```

No início deste ano, obtive dados de um site ativo e examinei as consultas mais custosas. Uma consulta demorava em média menos de 6 segundos, mas estava sendo chamada 14.000 vezes por dia, totalizando quase 24 horas decorridas todos os dias. Efetivamente, um núcleo estava totalmente ocupado com esta única consulta. Pior ainda, a segunda consulta que leva uma hora foi uma variação da primeira.

| Data de Execução | Nome da Rotina | Visitas Totais | Tempo Total | Hash | Texto da Consulta (abreviado) |
|---|---|---|---|---|---|
| 16/03/2019 | | 14.576 | 85.094 | 5xDSguu4PvK04se2pPiOexeh6aE= | DECLARE C CURSOR FOR SELECT * INTO :%col(1) , :%col(2) , :%col(3) , :%col(4) … |
| 16/03/2019 | | 15.552 | 3.326 | rCQX+CKPwFR9zOplmtMhxVnQxyw= | DECLARE C CURSOR FOR SELECT * INTO :%col(1) , :%col(2) , :%col(3) , :%col(4) , … |
| 16/03/2019 | | 16.892 | 597 | yW3catzQzC0KE9euvlJ+o4mDwKc= | DECLARE C CURSOR FOR SELECT * INTO :%col(1) , :%col(2) , :%col(3) , :%col(4) , :%col(5) , :%col(6) , :%col(7) , |
| 16/03/2019 | | 16.664 | 436 | giShyiqNR3K6pZEt7RWAcen55rs= | DECLARE C CURSOR FOR SELECT * , TKGROUP INTO :%col(1) , :%col(2) , :%col(3) , .. |
| 16/03/2019 | | 74.550 | 342 | 4ZClMPqMfyje4m9Wed0NJzxz9qw= | DECLARE C CURSOR FOR SELECT … |

***Tabela 1: Resultados reais do site do cliente***

## Tabelas no esquema INFORMATIONSCHEMA

Assim como as estatísticas, as tabelas neste esquema controlam onde as consultas, colunas, índices etc. são usados. Normalmente, a instrução SQL é a tabela inicial e é juntada em algo como “ Statements.Hash=OtherTable.Statement ” .

A consulta equivalente para acessar essas tabelas diretamente para encontrar as consultas mais custosas para um dia seria…

```
SELECT DS.Day,Loc.Location,DS.StatCount,DS.StatTotal,S.Statement,S.Hash
FROM INFORMATION_SCHEMA.STATEMENT_DAILY_STATS DS
left join INFORMATION_SCHEMA.STATEMENTS  S
on S.Hash=DS.Statement
left join INFORMATION_SCHEMA.STATEMENT_LOCATIONS  Loc
on S.Hash=Loc.Statement
where Day='08/26/2019'
order by DS.stattotal desc
```

Esteja você pensando em configurar um processo mais sistemático ou não, recomendo que todos com uma grande aplicação que usa SQL executem essa consulta hoje.

Se uma consulta específica parecer custosa, você pode obter o histórico  executando

```
SELECT DS.Day, Loc.Location, DS.StatCount, DS.StatTotal, S.Statement, S.Hash
FROM INFORMATION_SCHEMA.STATEMENT_DAILY_STATS DS
left join INFORMATION_SCHEMA.STATEMENTS S
on S.Hash=DS.Statement
left join INFORMATION_SCHEMA.STATEMENT_LOCATION Loc
on S.Hash=Loc.Statement
where S.Hash='jDqCKaksff/4up7Ob0UXlkT2xKY='
order by DS.Day
```

Exemplo de código para extrair estatísticas diariamente

#Desempenho #Monitoramento #SQL #Caché #InterSystems IRIS

**URL de origem:**https://pt.community.intersystems.com/post/apm-monitorando-o-desempenho-da-consulta-sql