Artigo
Danusa Calixto · jan 11, 2023 5min de leitura

Open Exchange

Armazenamento Colunar na 2022.3

Como você deve se lembrar do Global Summit 2022 ou do 2022.2 launch webinar, estamos lançando um novo e empolgante recurso para incluir em suas soluções analíticas no InterSystems IRIS. O Armazenamento Colunar apresenta uma maneira alternativa de armazenar os dados da tabela SQL que oferece uma aceleração de ordem de grandeza para consultas analíticas. Lançado pela primeira vez como um recurso experimental em 2022.2, o mais recente Developer Preview 2022.3 inclui várias atualizações que achamos que valeriam uma postagem rápida aqui.

Uma rápida recapitulação

Se você não está familiarizado com o Armazenamento Colunar, por favor dê uma olhada neste vídeo de breve introdução ou na sessão do GS2022 sobre o assunto. Resumindo, estamos codificando os dados da tabela em blocos de 64 mil valores por coluna usando um novo tipo de dado $vector. $vector é um tipo de dados interno (por enquanto) que utiliza esquemas de codificação adaptáveis para permitir o armazenamento eficiente de dados esparsos e densos. A codificação também é otimizada para um monte de operações $vector dedicadas, como para calcular agregados, agrupamentos e filtros de partes inteiras de 64k valores por vez, aproveitando instruções SIMD onde possível.

No momento da consulta SQL, aproveitamos essas operações criando um plano de consulta que também opera nesses blocos, o que, como você pode imaginar, gera uma redução massiva na quantidade de IO e no número de instruções ObjectScript para executar sua consulta, em comparação com o clássico processamento linha a linha. É claro que os IOs individuais são maiores e as operações $vector são um pouco mais pesadas do que as contrapartes de valor único do mundo orientado a linha, mas os ganhos são enormes. Usamos o termo planos de consulta vetorizados para estratégias de execução que lidam com dados $vector, empurrando blocos inteiros por meio de uma cadeia de operações individuais rápidas.

Apenas mais rápido

Mais importante ainda, as coisas ficaram mais rápidas. O novo kit inclui várias alterações na pilha que melhoram o desempenho, desde otimizações até aquelas operações $vector de baixo nível sobre alguns aprimoramentos de processamento de consulta e um conjunto mais amplo de planos de consulta vetorizados que podem ser paralelizados. Certas formas de carregar dados, como por meio de instruções INSERT .. SELECT, agora também empregarão um modelo de buffer que já usamos para criar índices e agora permitem a construção de tabelas inteiras com alto desempenho.

JOINs vetorizados

O recurso mais empolgante que adicionamos nesta versão é o suporte para unir dados colunares de maneira vetorizada. Em 2022.2, quando você deseja combinar dados de duas tabelas em uma consulta, ainda recorremos a uma estratégia robusta de JOIN linha por linha que funciona tanto em dados organizados em colunas quanto em linhas. Agora, quando ambas as extremidades do JOIN são armazenadas em formato colunar, usamos uma nova API do kernel para JOIN na memória, mantendo seu formato $vector. Este é outro passo importante para planos de consulta totalmente vetorizados, mesmo para as consultas mais complexas.

Aqui está um exemplo de uma consulta que aproveita a nova função, fazendo um self-JOIN do conjunto de dados New York Taxi que usamos em demonstrações anteriores:

```
SELECT
   COUNT(*),
   MAX(r1.total_amount - r2.total_amount)
FROM
   NYTaxi.Rides r1,
   NYTaxi.Rides r2
WHERE
   r1.DOLocationID = r2.PULocationID
   AND r1.tpep_dropoff_datetime = r2.tpep_pickup_datetime
   AND r2.DOLocationID = r1.PULocationID
   AND r1.passenger_count > 2
   AND r2.passenger_count > 2
```

Esta consulta procura pares de viagens com mais de 2 passageiros, onde a segunda viagem começou onde a primeira terminou, exatamente ao mesmo tempo e onde a segunda viagem levou um de volta para onde a primeira começou. Não é uma análise superútil, mas eu só tinha uma tabela real neste esquema e a chave JOIN composta tornou isso um pouco menos trivial. No plano de consulta dessa instrução, você verá trechos como Apply vector operation %VHASH (para construir a chave JOIN composta) e Read vector-join temp-file A, que indicam nosso novo marceneiro vetorizado em ação! Isso pode soar como uma pepita pequena e trivial em um plano de consulta longo, mas envolve muita engenharia inteligente internamente, e há alguns fornecedores de bancos de dados colunares de alto nível por aí que simplesmente não permitem nenhum dos isso e coloque severas restrições em seu layout de esquema, então, por favor, JUNTE-SE a nós para aproveitar isso! :-)

Quando o plano de consulta lê esse arquivo temporário, você pode perceber que ainda há algum processamento linha por linha no trabalho pós-junção, o que nos leva a...

O que vem a seguir?

O Armazenamento Colunar ainda está marcado como "experimental" em 2022.3, mas estamos nos aproximando da prontidão de produção e da vetorização completa de ponta a ponta para consultas de várias tabelas. Isso inclui o trabalho pós-junção mencionado acima, suporte mais amplo no otimizador de consulta, carregamento ainda mais rápido de tabelas colunares e aprimoramentos adicionais no joiner, como suporte a memória compartilhada. Resumindo: agora é um ótimo momento para dar uma primeira chance a tudo isso com o conjunto de dados de taxi de Nova York (agora em IPM ou com script docker) usando o 2022.3 Community Edition, então você só precisa pressionar "Executar" quando lançarmos o 2023.1!

Se você estiver interessado em obter conselhos mais personalizados sobre como aproveitar o armazenamento colunar com seus próprios dados e consultas, entre em contato comigo ou com sua equipe de contas diretamente e talvez nos encontremos no Global Summit 2023 ;-).

#SQL #InterSystems IRIS
Confira o aplicativo relacionado no InterSystems Open Exchange