بسم الله الرحمن الرحیم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

ترجمه ی رویه های ذخیره شده و جداول به dll

مدرس : مهندس افشین رفوآ

دوره آموزش SQL Server Administration

ترجمه ی رویه های ذخیره شده (stored procedure) و جداول به dll (native compilation)

معماری In-Memory OLTP مفهوم native compilation را ارائه می دهد، بدین معنا که SQL Server می تواند stored procedure هایی که به جداول memory-optimized دسترسی دارند را به dll کامپایل کند. SQL Server همچنین می تواند جداول بهینه سازی شده بر اساس حافظه را به dll ترجمه کند. قابلیت native compilation امکان دسترسی سریع تر به داده ها و query execution کارامدتری را نسبت به دستورات Transact-SQL تفسیر شده، فراهم می آورد. خروجی native compilation جداول و رویه های ذخیره شده، DLL می باشد.

Native compilation نوع جداول بهینه سازی شده بر اساس حافظه (memory optimized table types) نیز پشتیبانی می شود.

Native compilation به فرایندی اشاره دارد که طی آن سازه های برنامه نویسی (ساختارهای کنترلی) به کدهای native یا ذاتی تبدیل می شوند که شامل دستورات و فرامین پردازنده بدون نیاز به ترجمه و تفسیر بیشتر آن ها می شود.

In-Memory OLTP جداول بهینه سازی شده بر اساس حافظه را به مجرد ایجاد شدن آن ها و stored procedure های کامپایل شده به کدهای native را به هنگام بارگذاری یا لود آن ها به native dll، ترجمه می کند. علاوه بر آن، dll ها پس از بازآغازی (restart) پایگاه داده یا سرویس دهنده، مجددا کامپایل می شوند.

اطلاعات لازم برای بازسازی و ایجاد مجدد DLL ها در metadata های پایگاه داده مربوطه ذخیره می شوند. اگرچه DLL ها به پایگاه داده متصل و مربوط هستند ولی بخشی از آن محسوب نمی شوند. به عنوان مثال، DLL ها در فایل های پشتیبان پایگاه داده گنجانده نمی شوند.

نکته: دقت داشته باشید که جداول بهینه سازی شده بر اساس حافظه در طی بازآغازی سرویس دهنده (server restart)، مجددا کامپایل می شوند. به منظور افزایش سرعت پروسه ی بازیابی پایگاه داده (database recovery)، stored procedure های کامپایل شده به DLL حین بازآغازی سرور مجدد کامپایل نمی شوند، بلکه درست در زمان اولین اجرا ترجمه می شوند. در نتیجه ی این به تاخیر انداختن پروسه ی کامپایل، تنها زمان فراخوانی (Transact-SQL) sys.dm_os_loaded_modules و پس از اولین execution، stored procedure های ترجمه شده به dll پدیدار می شوند.

نگهداشت in-memory OLTP DLLS

Query زیر تمامی DLL های stored procedure و جدول را که در حال حاضر درون حافظه بر روی سرور بار گذاری شده اند را نمایش می دهد:

```
Transact-SQL
SELECT name, description FROM sys.dm_os_loaded_modules
where description = 'XTP Native DLL'
```

مدیران پایگاه داده ملزوم به نگهداری از فایل های تولید شده توسط native compilation نیستند. SQL Server به صورت خودکار فایل های تولید شده ای که دیگر استفاده ندارند را حذف می کند. به عنوان مثال، فایل های تولید شده هنگامی که یک جدول یا رویه ی ذخیره شده حذف می شوند و یا خود پایگاه داده به طور کلی پاک (drop) می شود، حذف می گردند.

نکته: چنانچه فرایند ترجمه با شکست مواجه و یا در آن تداخل ایجاد شود، برخی از فایل های تولید شده حذف نمی گردند. این فایل ها به طور عمد و به منظور دستیابی به قابلیت پشتیبانی (supportability) پاک نمی شوند و زمانی که پایگاه داده حذف می گردد، فایل های ذکر شده نیز همراه با آن پاک می شوند.

نکته: SQL Server، فایل های DLL را برای تمامی جداول مورد نیاز فرایند بازیابی پایگاه داده، کامپایل می کند. اگر یک جدول درست پیش از بازآغازی پایگاه داده حذف گردد، ممکن است باقی مانده های جدول مربوطه در فایل های checkpoint یا گزارش تراکنش همچنان ماندگار باشد، تا DLL ویژه ی جدول مورد نظر در طی راه

اندازی (startup) پایگاه داده مجدد کامپایل شود. پس از بازآغازی، DLL بارگذاری (upload) شده و فایل ها توسط پروسه ی متعارف پاک سازی (cleanup) حذف می گردند.

کامپایل جداول به DLL (native compilation)

ساخت و ایجاد یک جدول memoey-optimized به وسیله ی دستور CREATE TABLE باعث می شود که اطلاعات جدول نام برده در metadata پایگاه داده ضبط (write) شود و ساختارهای اندیس و جدول نیز در حافظه ایجاد گردند. جدول نیز به DLL ترجمه می شود.

پردازه ی نمونه (sample script) زیر را در نظر بگیرید. این پردازه یک پایگاه داده و همچنین یک جدول بهینه سازی شده بر اساس حافظه ایجاد می کند:

```
Transact-SQL
use master
go
create database db1
go
alter database db1 add filegroup db1_mod contains memory_optimized_data
go
-- adapt filename as needed
alter database db1 add file (name='db1_mod', filename='c:\data\db1_mod') to filegroup db1_mod
go
use db1
go
create table dbo.t1
  (c1 int not null primary key nonclustered,
   c2 INT)
with (memory_optimized=on)
go
-- retrieve the path of the DLL for table t1
select name, description FROM sys.dm_os_loaded_modules
where name like '%xtp_t_' + cast(db_id() as varchar(10)) + '_' + cast(object_id('dbo.t1') as varchar(10)) + '.dll'
go
```

با ایجاد جدول (به واسطه ی دستور CREATE TABLE)، DLL آن جدول نیز ایجاد گردیده، سپس DLL یاد شده در حافظه بار گذاری (load) می شود. DMV query که بلافاصله پس از دستور CREATE TABLE بکار گرفته شده، مسیر قرار گیری DLL را بازیابی می نماید.

DLL جدول، ساختارهای اندیس (index structure) و فرمت سطر های جدول را می شناسد. SQL Server با استفاده از DLL اندیس ها را پیموده (traverse) و سطرها را بازیابی می نماید، سپس محتویات سطرها را ذخیره می کند.

ترجمه ی رویه های ذخیره شده به DLL (native compilation of stored procedures)

رویه های ذخیره شده ای که با NATIVE_COMPILATION علامت گذاری شده اند، به DLL ترجمه می شوند. این کامپایلر فایل هایی تولید می کند که بر روی دیسک نوشته شده، سپس در حافظه بار گذاری می شود. این بدین معنا که دستورات Transact-SQL موجود در رویه (procedure)، به منظور دستیابی به اجرای بهینه و کارامد منطق business که کارایی برای آن از اهمیت بسیار بالایی برخوردار است، تماما به کدهای native ترجمه می شوند.

Stored procedure نمونه ی زیر را درنظر بگیرید، این رویه ی ذخیره شده سطرهایی را از مثال قبل در جدول t1 درج می کند.

```
Transact-SQL
create procedure dbo.native_sp
with native_compilation, schemabinding, execute as owner
as
begin atomic
with (transaction isolation level=snapshot, language=N'us_english')
 declare @i int = 1000000
 while @i > 0
 begin
  insert dbo.t1 values (@i, @i+1)
  set @i -= 1
 end
end
go
exec dbo.native_sp
go
-- reset
delete from dbo.t1
go
```

DLL ویژه ی native_sp می تواند علاوه بر موتور پایگاه داده (storage engine) In-Memory OLTP ، به طور مستقیم با DLL جدول t1 به منظور درج هر چه سریع تر سطرهای مورد نظر، تعامل برقرار می کند. کامپایلر In-Memory OLTP با بهره گیری از query optimizer (تابع بهینه ساز query) یک برنامه ی اجرای کارامد (execution plan) به ازای هر query در رویه ی ذخیره شده ایجاد می کند. توجه داشته باشید که در صورت تغییر داده های درون جداول، stored procedure های ترجمه شده به dll به صورت خودکار مجدد کامپایل نمی شود.

ملاحظات امنیتی در برخورد با native compilation

ترجمه ی جداول و رویه های ذخیره شده به dll (native compilation) از کامپایلر In-Memory OLTP بهره می گیرد. کامپایلر مزبور فایل هایی تولید می کند که بر روی دیسک نوشته شده و در حافظه بار گذاری می شود. SQL Server با استفاده از مکانیزم زیر دسترسی به این فایل ها را محدود می سازد.

Native compiler

فایل های اجرایی (compiler executable) و نیز فایل های دودویی و header مورد نیاز برای native compilation به عنوان جزئی از نمونه ی SQL Server تحت پوشه ی MSSQL\Binn\Xtp نصب می شوند. بنابراین اگرکه نمونه ی پیش فرض تحت پوشه ی C:\Program Files نصب گردد، فایل های کامپایلر در C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\Binn\Xtp نصب و جایگذاری می شوند.

به منظور محدود سازی دسترسی به کامپایلر، SQL Server از ACL ها (لیست مدیریت دسترسی) بهره می گیرد و بدین وسیله دسترسی به فایل های دودویی (باینری) را کنترل (محدود) می کند. کلیه ی فایل های دودویی SQL Server در برابر دستکاری و اصلاح از طریق ACL ها، حفاظت شده هستند. ACL های native compiler نیز استفاده از کامپایلر را محدود می سازند؛ فقط service account و مدیران سیستم SQL Server مجوزهای لازم برای خواندن و اجرا کردن فایل های native compiler را دارند.

فایل های ایجاد شده توسط native compilation

فایل هایی که به هنگام کامپایل یک جدول یا رویه ی ذخیره شده تولید می شوند، شامل DLL و فایل های میانجی (intermediate file) که خود دربردارنده ی فایل هایی با پسوند: .c، .obj، .xml، .pdb هستند، می شوند. فایل های ایجاد شده در یکی از زیرپوشه های data folder پیش فرض ذخیره می گردند. زیرپوشه ی مورد نظر Xtp نامیده می شود. هنگامی که نمونه ی پیش فرض را به همراه data folder پیش فرض نصب می کنید، فایل های ایجاد شده در C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\DATA\Xtp جایگذاری می شوند.

SQL Server به سه روش زیر از دستکاری DLL های ایجاد شده جلوگیری می کند:

1. زمانی که یک جدول یا رویه ی ذخیره شده به DLL ترجمه می شود، فایل DLL بلافاصله در حافظه بار گذاری شده، سپس به فرایند sqlserver.exe متصل (link) می شود. لازم است توجه داشته باشید که نمی توان یک DLL را هنگامی که به یک فرایند متصل است، اصلاح کرد.

2. هنگامی که یک پایگاه داده مجدد راه اندازی (restart) می شود، تمامی جداول و رویه های ذخیره شده، بر مبنای metadata پایگاه داده recompile می شوند(حذف شده و دوباره ایجاد می گردند). این کار باعث می شود که تمامی تغییراتی که توسط یک عامل مخرب (malicious agent) به فایل تولید شده اعمال گردیده، کاملا پاک شوند.
3. فایل های ایجاد شده بخشی از اطلاعات کاربر محسوب می شوند و دارای همان محدودیت های امنیتی (اعمال شده از طریق ACL) که فایل های پایگاه داده نیز دارند، هستند: فقط و فقط service account و مدیران سیستم اجازه ی دسترسی به این فایل ها را دارند.

برای مدیریت این فایل ها هیچ نیازی به تعامل کاربر نیست. SQL Server خود، فایل ها را در صورت لزوم ایجاد یا حذف می کند.