

mgr inż. Paweł Potasiński
Warszawska Wyższa Szkoła Informatyki
pawel.potasinski@gmail.com

Wybrane zastosowania widoków i funkcji dynamicznych w systemie Microsoft SQL Server 2012.

Selected uses of dynamic management views and functions in Microsoft SQL Server 2012

Streszczenie

W artykule omówiono przykłady zastosowania widoków i funkcji dynamicznych w systemie Microsoft SQL Server 2012. Podstawowa wiedza z tego zakresu jest niezbędna do sprawnego administrowania zasobami serwera baz danych.

Słowa kluczowe: dynamiczny widok, plan wykonania, procedura składowana

Abstract

Dynamic management views and functions (DMV) are one of the most useful source of knowledge on how Microsoft SQL Server 2012 database system is performing. They can be extremely useful for database administrators and people specialized in performance tuning and optimization. In this article three uses of DMVs have been described.

Keywords: dynamic view, execution plan, stored procedure

1. WSTĘP

Jednym ze źródeł informacji na temat funkcjonowania i zdrowia systemu Microsoft SQL Server 2012 są tak widoki jak i funkcje dynamiczne (ang. Dynamic Management Views – DMV). Oferują one administratorowi baz danych dostęp do informacji o samym systemie, jak również o aktualnym stanie wielu liczników, na podstawie których można wyciągnąć wnioski na temat funkcjonowania baz danych i całej instancji SQL Server. Część widoków dynamicznych gromadzi dane dotyczące wybranego aspektu działania systemu od ostatniego uruchomienia usługi SQL Server, dzięki czemu można często stwierdzić, czy przeprowadzone przez administratora strojenia i optymalizacje przyniosły korzyści. Celem artykułu jest przedstawienie wybranych zastosowań widoków i funkcji dynamicznych w procesach strojenia i optymalizacji systemu SQL Server.

W SQL Server 2012 w wersji RTM (numer wersji 11.0.2100) mamy do dyspozycji aż 177 widoków i funkcji dynamicznych. Pełną ich listę można wyświetlić wykonując poniższe zapytanie:

```
SELECT
    'sys.' + name AS [Nazwa DMV],
    type_desc AS [Rodzaj obiektu]
FROM sys.system_objects
WHERE schema_id = 4 AND name LIKE 'dm[_]%'
ORDER BY dmv_name;
```

Rys. 1 Zapytanie do wyświetlenia listy DMV.

Powyższe zapytanie wykorzystuje fakt, że wszystkie widoki i funkcje dynamiczne są obiektami systemowymi (stąd zapytanie do widoku sys.system_objects zwracającego listę obiektów systemowych), znajdują się w schemacie sys oraz mają nazwy rozpoczynające się od prefiksu „dm_”.

DMV są podzielone na kategorie oznaczone dodatkowymi prefiksami. Listę tych kategorii wraz z ilością DMV należących do każdej z nich można uzyskać wykonując następujące zapytanie:

```
SELECT
    ISNULL(LEFT(name,CHARINDEX('_',name,4)), '<<All>>') + '*' AS [Grupa DMV],
    COUNT(*) AS [Ilosc]
FROM sys.system_objects
WHERE schema_id = 4 AND name LIKE 'dm[_]%'
GROUP BY LEFT(name,CHARINDEX('_',name,4))
WITH ROLLUP
ORDER BY LEFT(name,CHARINDEX('_',name,4));
```

Rys. 2 Zapytanie do wyświetlenia grup DMV

Wynik powyższego zapytania:

Grupa DMV	Ilosc
<<All>>*	177
dm_audit_*	2
dm_broker_*	4
dm_cdc_*	2
dm_clr_*	4
dm_cryptographic_*	4
dm_database_*	1
dm_db_*	21
dm_exec_*	22
dm_filestream_*	3
dm_fts_*	12
dm_hadr_*	12
dm_io_*	4
dm_logconsumer_*	2
dm_logpool_*	4
dm_logpoolmgr_*	3
dm_os_*	41
dm_qn_*	1
dm_repl_*	4
dm_resource_*	4
dm_server_*	4
dm_sql_*	2
dm_tcp_*	1
dm_tran_*	11
dm_xe_*	9

Rys. 3 Lista grup DMV

A zatem w SQL Server 2012 istnieją 24 grupy DMV, z których najliczniejsza (41 DMV) jest grupa dm_os_* zawierająca widoki i funkcje poświęcone wewnętrznemu systemowi operacyjnemu SQLOS zaszytemu w systemie SQL Server.

2. INFORMACJE O SYSTEMIE

Często zdarza się, że podczas pracy z systemem SQL Server administrator lub audytor nie ma bezpośredniego dostępu do serwera, na którym działa system zarządzania bazami danych. Dzieje się tak dlatego, że nie zawsze na serwerze baz danych administratorzy udostępniają zdalne konsole, dzięki którym można podłączać się bezpośrednio do systemu operacyjnego serwera. Jak wówczas sprawdzić parametry serwera i dowiedzieć się, jakie konta w systemie operacyjnym zostały wykorzystane do uruchomienia usług związanych z systemem SQL Server? Odpowiedzią są DMV.

Poniższe zapytanie do DMV o nazwie `sys.dm_os_sys_info` zwraca podstawowe informacje o zasobach serwera, takich jak procesory i pamięć:

```
SELECT
  cpu_count AS [Logiczne procesory],
  cpu_count / hyperthread_ratio AS [Fizyczne procesory],
  CONVERT(numeric(19,2), physical_memory_kb / (1024 * 1024.)) AS [Pamięć w GB],
  sqlserver_start_time AS [Data startu systemu]
FROM sys.dm_os_sys_info;
```

Rys. 4 Zapytanie do wyświetlania informacji o serwerze

Wynik powyższego zapytania może prezentować się na przykład tak:

Logiczne procesory	Fizyczne procesory	Pamięć w GB	Data startu systemu
4	1	7.88	2012-05-11 10:53:17.333

Rys. 5 Wynik zapytania o zasoby serwera

Na podstawie powyższego wyniku jesteśmy w stanie stwierdzić, że dany serwer dysponuje jednym procesorem o czterech rdzeniach oraz ośmioma gigabajtami pamięci operacyjnej, a także, że ostatnie uruchomienie usługi SQL Server nastąpiło 11 maja 2012 roku o godzinie 10:53.

Z kolei, jeśli chcemy sprawdzić, na jakich kontach działają usługi systemu SQL Server, możemy wykonać zapytanie do DMV o nazwie `sys.dm_server_services`:

```
SELECT
  servicename AS [Usługa],
  service_account AS [Konto],
  status_desc AS [Stan]
FROM sys.dm_server_services;
```

Rys. 6 Zapytanie do sprawdzania stanu usług systemu SQL Server

Przykładowy wynik:

Usługa	Konto	Stan
SQL Server (MSSQLSERVER)	NT Service\MSSQLSERVER	Running
SQL Server Agent (MSSQLSERVER)	NT Service\SQLSERVERAGENT	Stopped
SQL Full-text Filter Daemon Launcher (MSSQLSERVER)	NT Service\MSSQLFDLauncher	Running

Rys. 7 Wynik zapytania o stan usług systemu SQL Server

Z powyższego wyniku można odczytać, że o ile dla każdej usługi zostało stworzone osobne konto lokalne w systemie operacyjnym, o tyle usługa SQL Server Agent odpowiedzialna za automatyzację procesów w systemie SQL Server jest zatrzymana (Stan = Stopped).

3. NA CO CZEKA SYSTEM?

Jednym z często stosowanych podejść w rozwiązywaniu problemów z wydajnością systemu SQL Server jest badanie stanów oczekiwania systemu. Każda milisekunda oczekiwania każdej sesji / zapytania w systemie SQL Server jest kategoryzowana i zapisywana w swoistym „pamiętniku”. Dostęp do zagregowanych danych z tego „pamiętnika” daje administratorowi DMV o nazwie sys.dm_os_wait_stats. Konstruując stosowne zapytanie do tego widoku można uzyskać informacje o tym, na co najczęściej czeka system baz danych podczas realizacji zadań. Poniższe zapytanie wyświetli te kategorie stanów oczekiwania, które generują ponad 90% czasu oczekiwania systemu:

```
WITH Waits AS (
  SELECT
    wait_type,
    waiting_tasks_count,
    wait_time_ms / 1000 AS wait_time_s,
    100. * wait_time_ms / SUM(wait_time_ms) OVER() AS pct,
    ROW_NUMBER() OVER(ORDER BY wait_time_ms DESC) AS rn
  FROM sys.dm_os_wait_stats
  WHERE wait_type NOT IN (
    'CLR_SEMAPHORE', 'LAZYWRITER_SLEEP', 'RESOURCE_QUEUE',
    'SLEEP_TASK', 'SLEEP_SYSTEMTASK', 'SQLTRACE_BUFFER_FLUSH',
    'WAITFOR', 'LOGMGR_QUEUE', 'CHECKPOINT_QUEUE', 'REQUEST_FOR_DEADLOCK_
SEARCH',
    'XE_TIMER_EVENT', 'BROKER_TO_FLUSH', 'BROKER_TASK_STOP', 'CLR_MANUAL_
EVENT',
    'CLR_AUTO_EVENT', 'DISPATCHER_QUEUE_SEMAPHORE', 'FT_IFTS_SCHEDULER_
IDLE_WAIT',
    'XE_DISPATCHER_WAIT', 'XE_DISPATCHER_JOIN'
  )
)
SELECT
  W1.wait_type AS [Kategoria],
  W1.waiting_tasks_count AS [Ilość oczekiwań],
  CAST(W1.wait_time_s AS decimal(12, 2)) AS [Czas oczekiwania w sek.],
  CAST(W1.pct AS decimal(12, 2)) AS [Procent],
  CAST(SUM(W2.pct) AS decimal(12, 2)) AS [Procent w sumie]
FROM Waits AS W1
INNER JOIN Waits AS W2
ON W2.rn <= W1.rn
GROUP BY W1.rn, W1.wait_type, W1.waiting_tasks_count, W1.wait_time_s, W1.pct
HAVING SUM(W2.pct) - W1.pct < 90;
```

Rys 8. Zapytanie do wyświetlania stanów oczekiwania systemu

Powyższe zapytanie zostało skonstruowane tak, by nie pokazywać kategorii oczekiwania, które zazwyczaj mogą zostać zignorowane podczas monitorowania systemów SQL Server 2008 R2 i starszych wersji. Przykładowy wynik zapytania:

Kategoria	Ilość oczekiwania	Czas oczekiwania w sek.	Procent	Procent w sumie
WRITELOG	252162	2394.41	75.64	75.64
PAGEIOLATCH_SH	11876	219.60	6.94	82.57
SLEEP_BPOOL_FLUSH	23465	149.25	4.71	87.29
ASYNC_NETWORK_IO	44104	78.87	2.49	89.78
PREEMPTIVE_OS_WAITFORSINGLEOBJECT	36537	77.56	2.45	92.23

Rys. 9 Przykładowe stany oczekiwania systemu

Na podstawie powyższego wyniku można stwierdzić, że na badanej instancji SQL Server najdłużej (75% czasu oczekiwania) system oczekuje na zapis do dzienników transakcji baz danych (kategoria WRITELOG). Może to wskazywać na problemy związane z nieprawidłowym rozmieszczeniem plików baz danych lub zbyt wolnymi dyskami przeznaczonymi na pliki dzienników transakcji. Dzięki takiej analizie administrator może przystąpić do dalszego monitorowania ukierunkowanego na konkretne problemy, przez co jest w stanie szybciej zdiagnozować przyczynę, dla której operacje w systemie trwają dłużej, niż powinny.

Kategorie stanów oczekiwania systemu SQL Server opisane są w dokumentacji do widoku sys.dm_os_wait_stats (<http://technet.microsoft.com/en-us/library/ms179984.aspx>) oraz w bardzo dobrym, nadal aktualnym artykule SQL Server 2005 Performance Tuning using the Waits and Queues (<http://technet.microsoft.com/en-us/library/cc966413.aspx>).

4. PLANY WYKONANIA PROCEDUR SKŁADOWANYCH

Procedury składowane są często wykorzystywane przez programistów baz danych pracujących z systemem SQL Server. Dzięki zastosowaniu procedur programista może zaszyć dowolny kod implementujący logikę biznesową w bazie danych, co upraszcza proces utrzymania aplikacji. Procedury dają także możliwość ograniczenia dostępu użytkowników do tabel. Wystarczy, że programista stworzy stosowne procedury składowane realizujące operacje na wybranej tabeli i nada użytkownikowi uprawnienia do wykonywania tych procedur, a użytkownik będzie mógł realizować operacje modyfikacji danych w tabeli bez potrzeby posiadania uprawnień na samej tabeli. Jednocześnie za pomocą kodu zaszytego w procedurach programista może wprowadzić własne mechanizmy kontroli jakości danych oraz przebiegu operacji.

Rozwiązywanie problemów wydajnościowych zaistniałych w procedurach składowanych nie należy do prostych zadań. Jeśli procedura realizuje złożoną logikę i dokonują się za pomocą jej kodu modyfikacje danych, uruchamianie procedury w celach testowych nie wchodzi często w grę. W takich momentach dobrym podejściem jest odnalezienie zapisanego w pamięci planu wykonania procedury i wykonanie analizy tego planu. Dostęp do planów wykonania oraz statystyk uruchamiania procedur składowanych można zrealizować za pomocą następującego zapytania:

```
SELECT
  DB_NAME(p.database_id) AS [Baza danych],
  QUOTENAME(OBJECT_SCHEMA_NAME(p.[object_id], p.database_id)) + '.' +
  QUOTENAME(OBJECT_NAME(p.[object_id], p.database_id)) AS [Procedura],
```

```
p.cached_time AS [Data zapisania planu],
p.last_execution_time AS [Data ostatniego wykonania],
p.execution_count AS [Ilość wykonan],
CONVERT(
    numeric(38,2),
    CASE
        WHEN p.execution_count = 0
        THEN 0
        ELSE p.total_elapsed_time * 1.0 / p.execution_count
        END) AS [Średni czas wykonania],
q.query_plan AS [Plan]
FROM sys.dm_exec_procedure_stats AS p
CROSS APPLY sys.dm_exec_query_plan(p.plan_handle) AS q
WHERE p.database_id = DB_ID()
ORDER BY p.execution_count DESC;
```

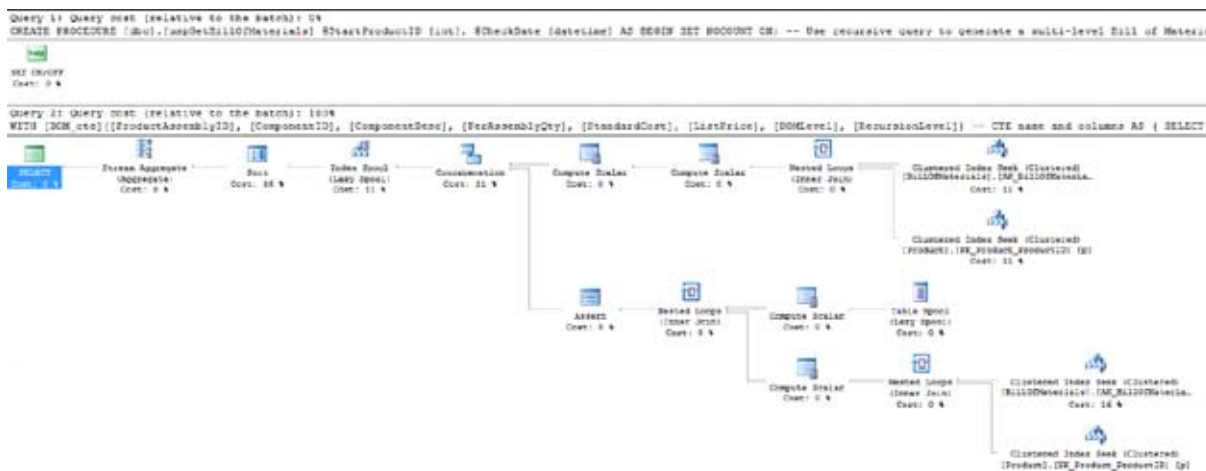
Rys. 10 Zapytanie do wyświetlania statystyk oraz planów wykonania procedur składowanych

Zapytanie korzysta z DMV o nazwie sys.dm_exec_procedure_stats zwracającego statystyki wykonania procedur składowanych oraz z funkcji sys.dm_exec_query_plan, która zwraca między innymi plan wykonania w formacie XML. Przykładowy wynik powyższego zapytania dla bazy danych AdventureWorks2012:

Baza danych	Procedura	Data zapisania planu	Data ostatniego wykonania	Ilość wykonan	Średni czas wykonania	Plan
AdventureWorks2012	[dbo].[uspGetManagerEmployees]	2012-05-15 08:58:33.803	2012-05-15 08:58:33.897	1	121903.00	ShowPlanXML.xml?url=http://schemas.microsoft.com...
AdventureWorks2012	[dbo].[uspGetEmployeeManagers]	2012-05-15 08:58:24.587	2012-05-15 08:58:24.697	1	28599.00	ShowPlanXML.xml?url=http://schemas.microsoft.com...
AdventureWorks2012	[dbo].[uspGetBOMMaterials]	2012-05-15 08:58:13.773	2012-05-15 08:58:13.803	1	350.00	ShowPlanXML.xml?url=http://schemas.microsoft.com...

Rys. 11 Statystyki i plany wykonania procedur składowanych

Narzędzie SQL Server Management Studio potrafi interpretować plany wykonania zapisane w formacie XML i wyświetlać je w postaci graficznej. Przykład takiego graficznego planu wykonania ilustruje rysunek 6.



Rys. 12 Przykładowy plan wykonania procedury składowanej

Mając plan wykonania procedury w postaci graficznej i w formacie XML, administrator może przeanalizować plan i dowiedzieć się, które polecenie w procedurze wykonywało się najdłużej i powodowało wykorzystanie największej ilości zasobów serwera.

5. PODSUMOWANIE

Widoki i funkcje dynamiczne to potężna broń w rękach osób pracujących z systemem Microsoft SQL Server. Dzięki ich znajomości administrator baz danych może uzyskać wiele cennych informacji, których użycie może prowadzić do rozwiązania nawet najbardziej skomplikowanych problemów zarówno z pojedynczymi poleceniami Transact-SQL, jak i z całą bazą danych, a nawet instancją systemu SQL Server.

W artykule przedstawiłem zaledwie trzy możliwe zastosowania DMV, ale warto wymienić inne często spotykane zastosowania tych obiektów:

- monitorowanie sesji i połączeń – w tym wystąpień problemu blokowania,
- poszukiwanie brakujących indeksów,
- monitorowanie użycia i fragmentacji indeksów,
- monitorowanie wykorzystania zasobów w bazie tempdb (obiekty tymczasowe),
- wyszukiwanie zależności między obiektami w bazach danych,
- przeszukiwanie bufora planów wykonania (np. celem znalezienia planów wykonania wykorzystujących wiele procesorów).

Podsumowując, widoki i procedury dynamiczne powinien poznać każdy, kto sprawnie posługuje się językiem Transact-SQL, a chciałby mieć lepszą kontrolę nad procesami, które dzieją się w systemie SQL Server.

