

SQL Server 성능관리

(주)엑셈 컨설팅본부/SQL Server팀 김 범규

1. SQL Server 성능 모니터링

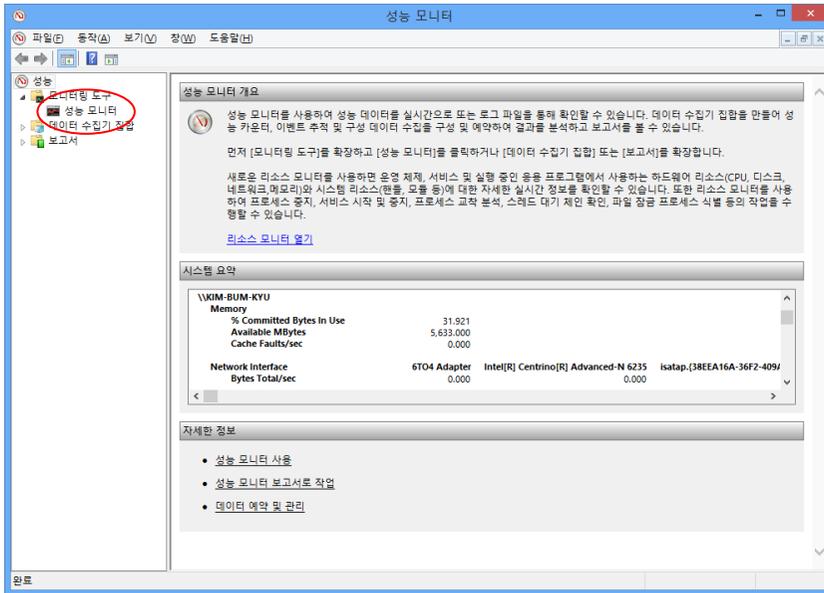
관리자가 SQL Server의 현재 상태가 정상인지 확인하고자 한다. 어떻게 SQL Server의 상태를 확인할 수 있을까?

가장 쉽게 DB 서버의 상황을 확인하는 방법은 작업관리자를 실행시켜 현재 CPU와 Memory 사용률을 확인하는 것이다. 하지만 SQL Server의 Memory 설정이 자동(기본값)으로 되어 있다면 Memory 사용률을 확인하는 것은 의미가 없다. (SQL Server는 한번 사용한 Memory를 여간해서는 OS에 반환하지 않으므로 여유 메모리가 부족하다는 것만으로 문제임을 판단할 수 없다.) 결국 작업관리자를 통해 얻을 수 있는 정보는 SQL Server의 CPU 사용률 정도이고, 보다 정확한 정보를 확인하기 위해서는 Windows에서 제공하는 Perfmon (Performance Monitor)과 SQL Server의 DMV(Dynamic Management View) 등을 이용하여야 한다.

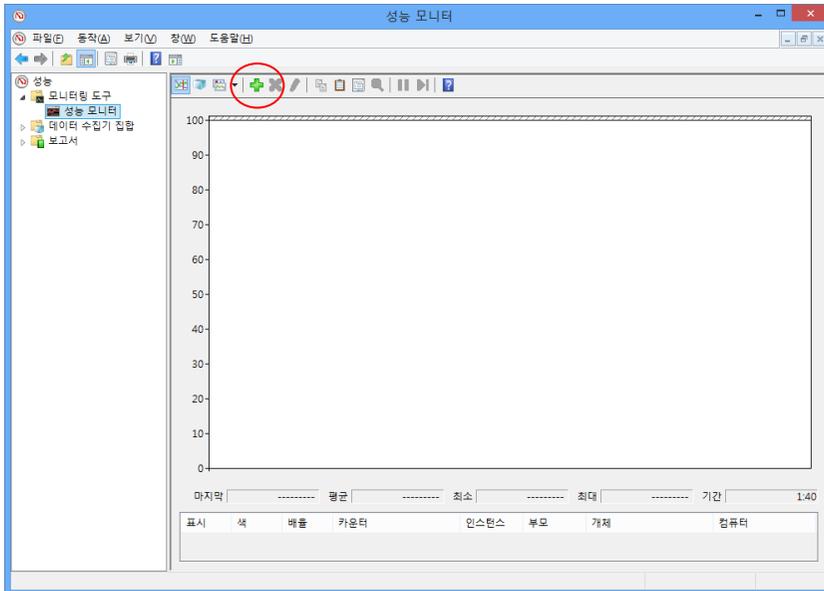
모니터링 ■

Performance Monitor

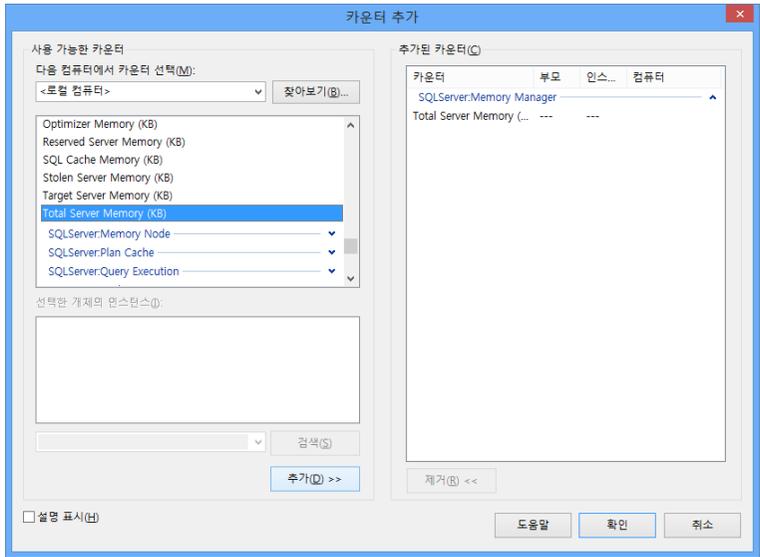
Windows에서 기본적으로 제공되는 모니터링 툴로써, SQL Server 설치 시에 SQL Server 관련 모니터링 항목들이 추가된다. [시작]에 Perfmon을 입력하여 실행하고, 다음과 같은 화면이 나타나면 모니터링에 필요한 항목을 선택한다. 예를들어 현재 SQL Server가 사용하고 있는 메모리 사용량을 확인하기 위해 하고자 한다면, [인스턴스명]:Memory Manager\Total Server Memory(KB)을 선택한다.



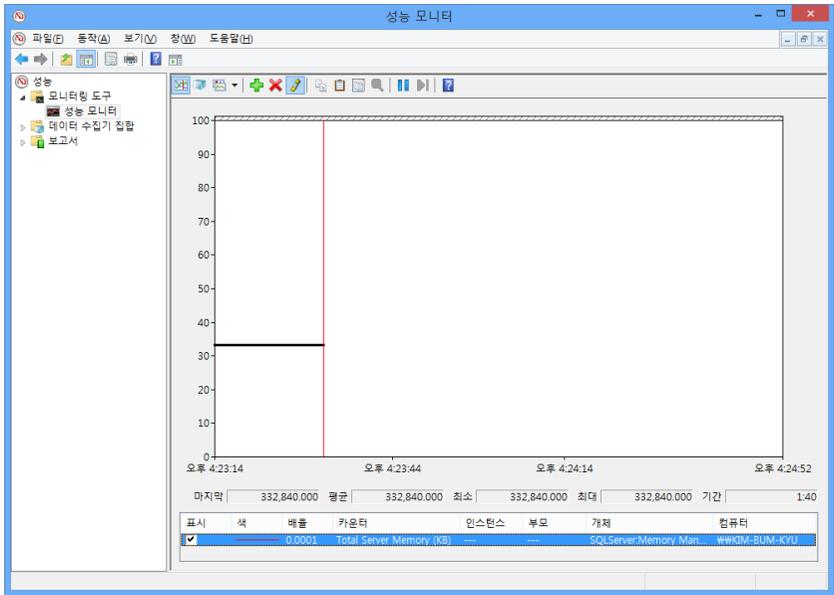
[그림1] Perfmon 실행



[그림2] 모니터링 카운터 추가



[그림3] Total Server Memory (KB) 선택



[그림4] Total Server Memory (KB) 선택

이와 같은 방법으로, 다음과 같은 지표들을 참고하면 시스템의 상태를 보다 정확하게 파악할 수 있다.

- CPU 관련

[인스턴스명]:Processor\Processor Time

[인스턴스명]:Processor\User Time

- Memory 관련

[인스턴스명]:Buffer Manager\Page lookups/sec

[인스턴스명]:Buffer Manager\Page life expectancy

- Disk 관련

[인스턴스명]:Buffer Manager\Page reads/sec

[인스턴스명]:Buffer Manager\Page writes/sec

[인스턴스명]:Buffer Manager\Readahead pages/sec

[인스턴스명]:PhysicalDisk\Avg. Disk Queue Length

[인스턴스명]:PhysicalDisk\Current Disk Queue Length

- Transaction 관련

[인스턴스명]:Databases\Active Transactions

[인스턴스명]:Databases\Transactions/sec

[인스턴스명]:Databases\Active Transactions

- Workload 관련

[인스턴스명]:SQL Statistics\Batch Requests/sec

[인스턴스명]:SQL Statistics\SQL Compilations/sec

Dynamic Management View

SQL Server 2005 부터 Database Engine 이 직접 성능관련 정보를 관리하고 있으며, 성능관리 뷰(Dynamic Management View)를 통해 관련 정보를 확인하는 것도 가능하다. 이를 통해 시스템&오브젝트 정보 뿐만 아니라, 현재 수행되고 있는 세션관련 정보(SQL, Plan)와 대기정보(lock, latch, spinlock 등)를 실시간으로 확인 가능하다.

- 시스템&오브젝트관련

```
sys.databases  
sys.objects  
sys.objectcaches  
sys.indexes
```

- 세션관련

```
sys.sysprocesses  
sys.dm_exec_requests  
sys.dm_exec_sessions
```

- 대기관련

```
sys.dm_tran_locks  
sys.dm_waiting_tasks
```

- SQL&PROCEDURE 관련

```
sys.dm_exec_sql_text  
sys.dm_exec_query_plan  
sys.dm_exec_query_stats
```

CAWL 모니터링 방법론

앞서 데이터베이스의 상태를 파악하기 위한 필요한 정보를 확인하는 방법에 대해서 알아 보았다. 그렇다면 이 정보들을 활용하여 상태를 파악하는 방법에 대해 알아보자.

필자는 기본적으로 데이터베이스의 상태파악을 위해 CPU, Active Sessions, Wait Time, 그리고 Logical Reads 를 확인한다. 그래서 이 4 가지 지표의 앞글자를 모아 CAWL(카울)이라고 부른다. 이렇게 4 가지의 기본 정보만으로도 시스템의 이상유무를 판단하는데 부족함이 없다.

Active Sessions 은 세션의 상태가 running(SQL 실행중)이거나, suspended(대기중)인 세션으로 정의하며, sys.sysprocesses 에서 spid>50 이고 status<>'sleeping' 조건을 만족하는 세션들이다.

Wait Time 은 각 세션들이 1 초동안 대기한 시간들의 합으로 정의한다. 하나의 세션이 1 초동안 대기시간할 수 있는 최대 시간은 1 초이다. 따라서 만약 10 개의 세션이 1 시간동안 lock 을 대기하고 있다면 이 시스템의 현재 대기시간은 10 초가 된다. 이 정보를 확인하기 위해서는 해당 정보를 약간 가공해서 보여 줄 수 있는 스크립트 혹은 툴을 활용하면 보다 효과적으로 실시간 정보를 확인하기가 용이하다.

Logical Reads 는 데이터 검색을 위해 발생된 Memory IO 를 의미하며, SQL Server 에서는 Page lookups/sec 지표가 해당값을 의미한다. SQL 이 수행되기 위해서는 Memory IO 가 반드시 사용되므로 Logical Reads 를 확인 함으로써 가장 직관적으로 DB 의 일량을 확인할 수 있다.

DBMS 의 정상여부를 확인하는 방법에 대해 살펴보자.

먼저, 가장 손쉽게 확인가능한 정보인 CPU 를 확인하고, Logical Reads 과 Active Sessions 을 확인하여 DBMS 의 이상유무를 확인할 수 있다.

CPU ↑ Logical Reads ↓ : DB 문제 가능성 少. SQL Server 이외의 프로그램에 의한 CPU 사용여부 확인
CPU ↑ Logical Reads ↑ : DB 문제 가능성 多
CPU ↑ Logical Reads ↑ Active Sessions ↑ : 비상
CPU ↑ Logical Reads ↑ Active Sessions ↓ : 특정 악성쿼리 수행여부 확인
CPU ↓ Logical Reads ↓ Active Sessions ↑ : Lock 과 같은 Blocking 대기 확인
Wait Time ↑ : DB 내 경합으로 처리시간 지연 발생 가능

Wait Time ↓ CPU ↑ : DB 가 문제라기 보다 업무 처리량이 증가한 상태로 추정 가능. DB 접속이 원활한지 확인 필요.

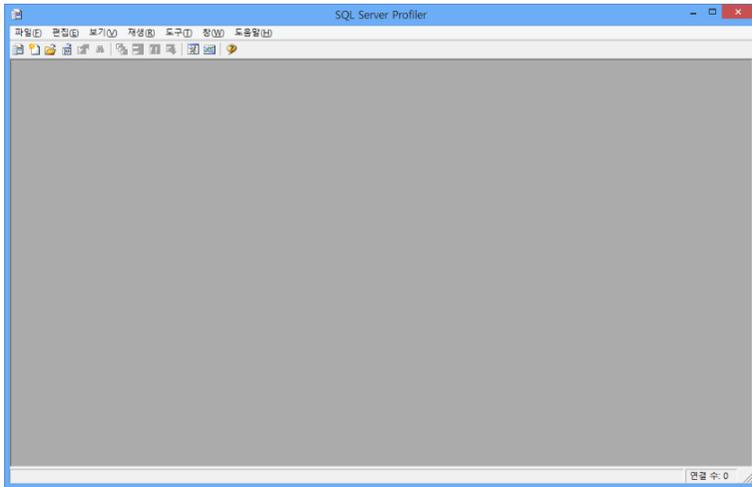
2. SQL튜닝

SQL튜닝대상 수집

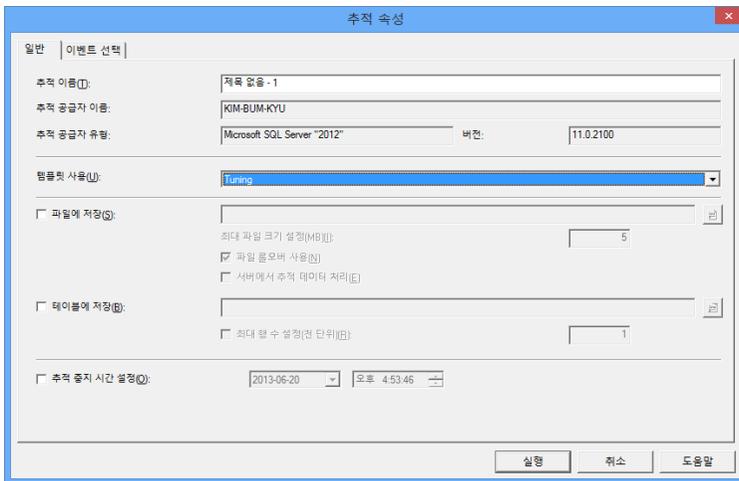
SQL 튜닝 대상 선정을 위해 일반적으로 사용되는 방법으로 Trace 정보를 활용하는 방법과 DMV(Dynamic Management View)를 활용하는 방법이 있다. Trace 정보를 수집하여 SQL 튜닝 대상을 추출하는 방법은 전통적으로 튜닝에 사용되어 온 매우 효과적인 방법으로, 다양한 filter 조건을 이용하여 원하는 정보만을 수집가능하다. 예를 들어, 특정 사용자, DB 를 사용하는 SQL 만을 찾을 수 있으며, 1 회 수행 시 수행시간이 3 초 이상이면서 5,000 Pages 이상을 Access 하는 SQL 만을 수집하는 것도 가능하다. 하지만 조건을 잘 못 기술할 경우 시스템에 매우 큰 부하를 유발시킬 수 있으므로 주의가 필요하다.

가. Profiler 를 이용한 Trace 정보 수집

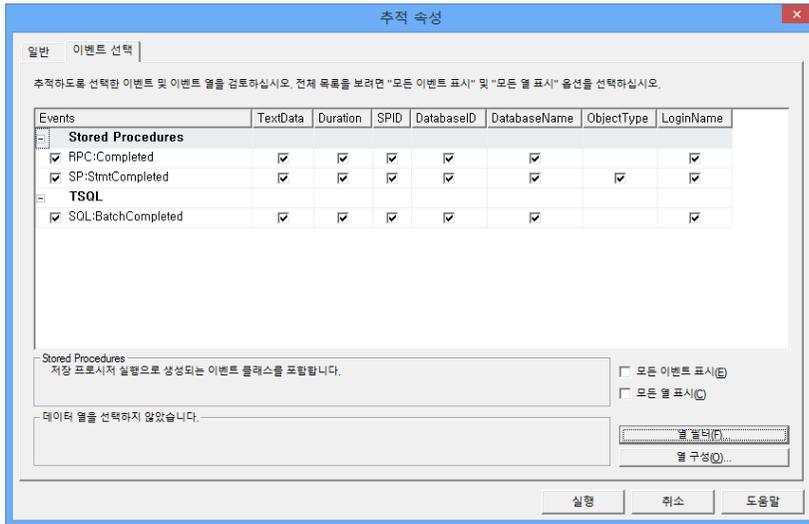
SQL Server 는 수행되는 SQL 에 대한 Trace 정보를 손쉽게 수집할 수 있도록 UI 를 제공한다. SSMS 의 도구메뉴에서 Profiler 를 실행하면 다음과 같은 프로그램이 실행된다.



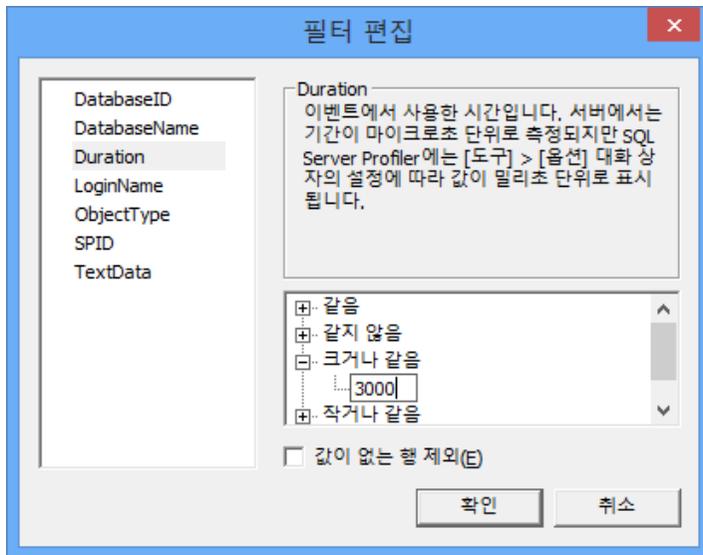
[그림5] Profiler 실행



[그림6] Profiler 접속



[그림7] 이벤트 선택1



[그림8] 이벤트 선택2

EventClass	TextData	Duration	SPID	Databas...	DatabaseN...	ObjectTy...	LoginN...
Trace Start							
SQL: BatchCompleted	select + fro...	4210	53	8	escalet ion		exem

[그림9] Trace 시작

나. Script 를 이용한 Trace 정보 수집

앞에서 살펴본 바와 같이 UI 를 통해 설정한 Profiler 설정을 Script 로 저장도 가능하다.

파일→내보내기→추적 정의 스크립팅→SQL Server 2005 - SQL11

이렇게 기존에 설정한 정보를 Script 저장하고, 해당 Script 를 열어 다음의 내용을 적당한 값으로 수정한다.

```
set @maxfilesize = 5 - 단위파일의 최대 사이즈(MB), 일반적으로 100 정도로 설정
exec @rc = sp_trace_create @TraceID output, 0, N'C:\trace_test', @maxfilesize, NULL -
생성파일 위치와 이름 지정
```

수정이 완료되면 해당 Script 를 실행하고, 일정시간 이후 중지하고자 할 때는 다음의 명령을 수행한다.

```
exec sp_trace_setstatus 2, 0 -- 지정한 추적을 중지합니다.
exec sp_trace_setstatus 2, 2 -- 지정한 추적을 닫고 서버에서 해당 정의를 삭제합니다.
```

C:\에 생성된 trace_test.trc 파일을 profiler 를 이용하여 열어 보면, 해당 시점에 수행된 다양한 SQL 을 확인 할 수 있다. 이렇게 Script 를 이용하여 Trace file 을 수집하는 방법이 부하가 적어, 운영 담당자들은 이 방법을 선호한다.

다. PSSDIAG 를 이용한 긴급 로그 수집하기

장애가 발생하거나 하여, 긴급히 데이터를 수집해야 할 경우가 발생한다. 다음과 같은 방법을 참고하여 향후 장애시점의 상황을 추정할 수 있는 데이터를 남겨놓을 수 있으며, 기술지원을 요청하게 될 경우 큰 도움이 될 수 있으므로 아래의 프로세스는 반드시 숙지하고 있어야 한다.

1. SQL Server 머신에 공간이 충분한 임의의 폴더를 생성(예를들어, D:\WPSSDIAG)하고, Microsoft홈페이지에서 최신의 파일을 다운받는다. 파일을 실행하면 파일의 압축이 자동으로 해제된다.

2. 명령프롬프트(Command Prompt)를 실행하고 앞에서 생성한 폴더로 이동한 다음, 압축 해제 후 생성된 pssdiag.cmd를 실행한다. pssdiag.cmd를 실행하면 몇가지 실행관련 메시지 발생후 다음과 같은 내용의 메시지가 출력되며 이때부터 필요한 자료수집이 시작된다.

2011/03/16 17:44:59.13 PSSDIAG Collection started. Press Ctrl+C to stop.

3. 명령 프롬프트에서 'Ctrl+C' 를 눌러 자료수집이 중지된다.

'Ctrl+C' 누른후 수집된 자료를 저장하는 과정에 약간의 시간이 소요되므로 작업이 완전히 끝날때까지 기다려야하며, 모든 작업이 완료되면 다음과 같은 내용의 메시지가 출력된다.

2011/03/16 17:47:00.99 PSSDIAG Collection complete. Collector exiting

4. pssdiag.exe가 존재하는 폴더의 아래에 output이라는 폴더가 생성되는데 수집된 자료는 모두 output 폴더에 저장된다.

튜닝정보 분석

지금까지 시스템 부하분석을 위한 정보를 수집하는 방법에 대해 살펴보았다. 지금 부터는 이정보를 통해 우리가 튜닝에 활용할 수 있는 정보를 추출하는 방법에 대해서 살펴보도록 한다.

가. SQL Server 에 Trace file Loading 하여 분석하기

우리가 지금까지 수집한 Trace file 을 데이터베이스 넣어 활용하는 방법이다. 이 모든 데이터가 데이터베이스에 넣을 수만 있다면 굳이 설명하지 않더라도 자연스럽게 데이터를 활용하고 분석할 수 있을 것이다. 그럼 Trace file 을 데이터베이스에 넣는 방법에 대해 알아보자.

```
USE tempdb;
GO
SELECT * INTO temp_trc
FROM fn_trace_gettable('C:\trace_test.trc', default);
GO
```

이제는 기다리기 하면 Trace file 이 temp_trc 테이블에 자동으로 로딩된다.

참고로 이것은 우리가 수집한 Event Class 번호이다. 이것을 참고하여 검색하여 튜닝에 활용하면 되며, 해당 정보 중에 reads, duration 등을 내림차순으로 정렬하여 조회하면 가장 시스템에 큰 부하를 주는 SQL 을 쉽게 확인 할 수 있다.

```
--RPC:Completed          10
--SQL:BatchCompleted     12
--SQL:StmtCompleted      41
--SP:StmtCompleted       45

SELECT endtime, reads, duration FROM dbo.temp_trc
WHERE eventclass IN (10,12)
AND reads>1000
AND duration>1000
ORDER BY reads
```

나. RML 을 이용하여 Trace file 분석하기

Microsoft 에서 제공하는 RML(Replay Markup Language) utilities 을 이용하면 분석된 정보를 그래프를 통해 보다 쉽게 확인할 수 있다. 먼저 최신의 RML utilities 를 Microsoft Download Center 에서 다운받는다. (<http://support.microsoft.com/kb/944837/en-us>) 설치가 완료되면 Command 창에서 해당 폴더로 이동하여 ReadTrace 실행하면 수집한 Trace file 을 자동으로 분석하여 Graphical Report 를 작성하여 제공해준다. ReadTrace 는 해당정보를 분석하기 위해

데이터베이스에 해당 Trace file 을 load 한다. 따라서 먼저 사용할 수 있는 사용할 수 있는 데이터베이스에 대한 접속정보가 필요하다.

```
cd C:\Program Files\Microsoft Corporation\RMLUtils
-- DISK C:를 소문자로 하면 에러남
ReadTrace -I"C:\trace_test.trc" -S"KIM-BUM-KYU" -E
```

다. DMV 를 이용하여 분석하기

SQL Server 2005 이상의 버전에서는 Profiler 를 통한 별도의 Trace file 수집 없이도 자체적으로 관리되는 DMV 를 이용하면 수행되었던 SQL Text 와 Procedure 에 대한 통계정보를 확인하는 것도 가능하다. 경우에 따라 Profiler 를 이용하는 방법에 비해 정보가 조금 부족할 경우도 있으나, 서버에 부하없이 원하는 정보를 확인할 수 있다는 특별한 장점 때문에 잘 활용만 한다면 매우 유용한 방법이다. 다음의 DMV 를 활용하면 성능모니터링 및 관리에 많은 정보를 얻을 수 있으므로 참고하기 바란다.

- sys.sysprocesses
- sys.dm_exec_requests
- sys.dm_tran_locks
- sys.dm_waiting_tasks
- sys.dm_exec_query_stats
- sys.dm_exec_procedure_stats
- sys.dm_exec_query_plan