



SQL Server Monitoring

Data Collector

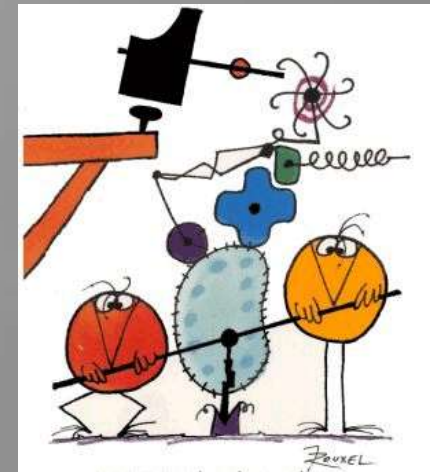
Eléments de monitoring

- Compteurs de performances
- DMV's et DBCC
- Evènements (jobs, services, ...)
- Stratégies (policies)
- Traces
- Plans d'exécution



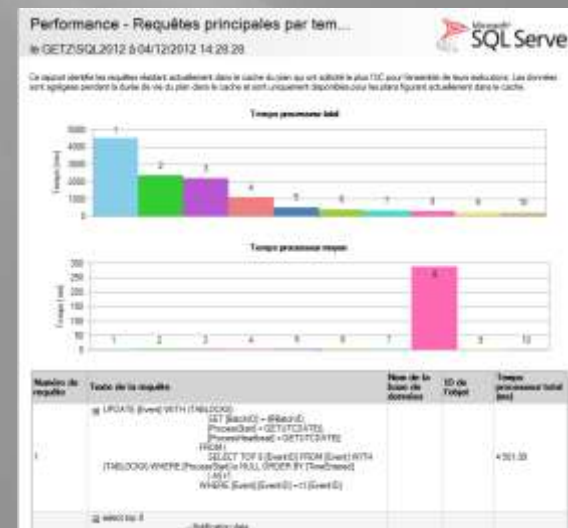
Outils de mesure

- Multiples, nombreux et dépendent du besoin
 - Rapports de base de données
 - Data Collector
 - Moniteur de performances, PAL
 - Moniteur d'activité
 - Profiler, ReadTrace, RML, Nexus
 - Plans d'exécutions, statistics io & time
- Outils tiers
 - Quest, Red Gate, Idera



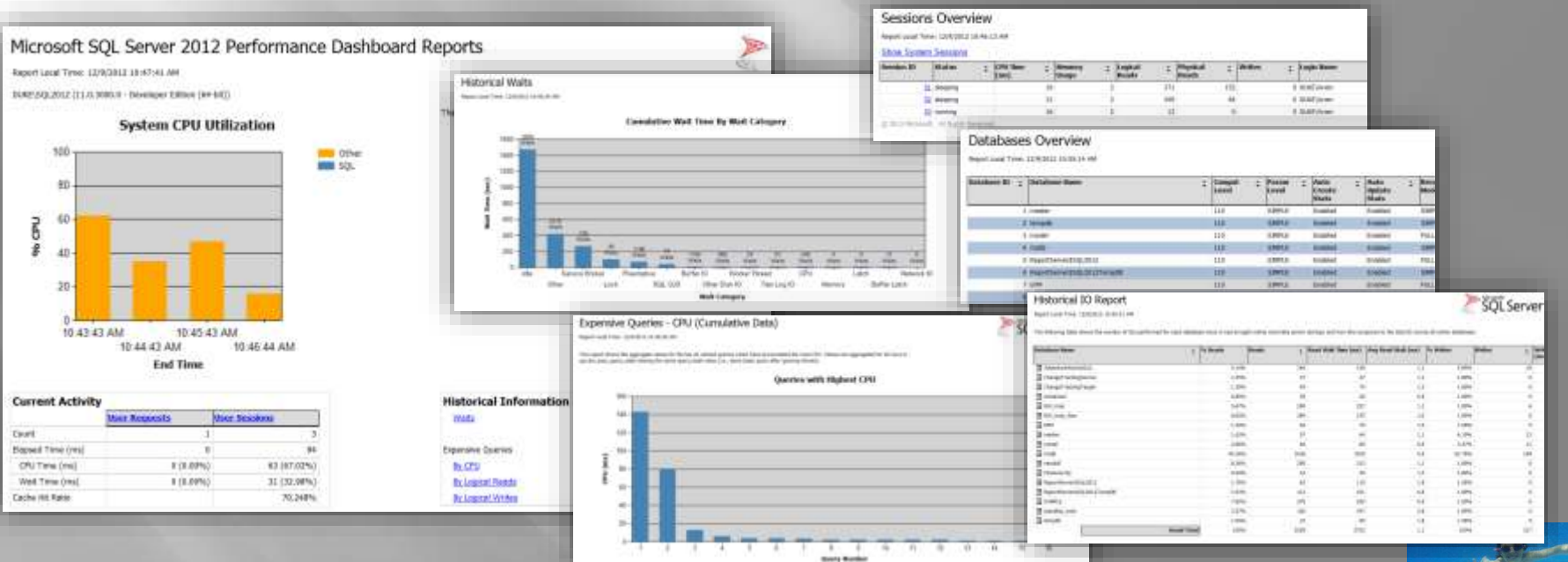
Les rapports SSMS

- Exploitent les DMV's
- Rapports au niveau serveur ou base
 - Utilisation du stockage
 - Fichiers, tables
 - Requêtes coûteuses
 - Lots ou requêtes en cache agrégées par coût (UC, IO) ou par objet
 - Verrouillage
 - Attentes de verrous
 - Index
 - Utilisation des index
 - Fragmentation



Encore plus de rapports ?

- SQL Server 2012 Performance Dashboard Reports
 - Téléchargement gratuit : <http://www.microsoft.com/en-us/download/details.aspx?id=29063>
 - Rapports « Custom » orientés performance
 - Utilisables aussi avec SQL 2008 & R2



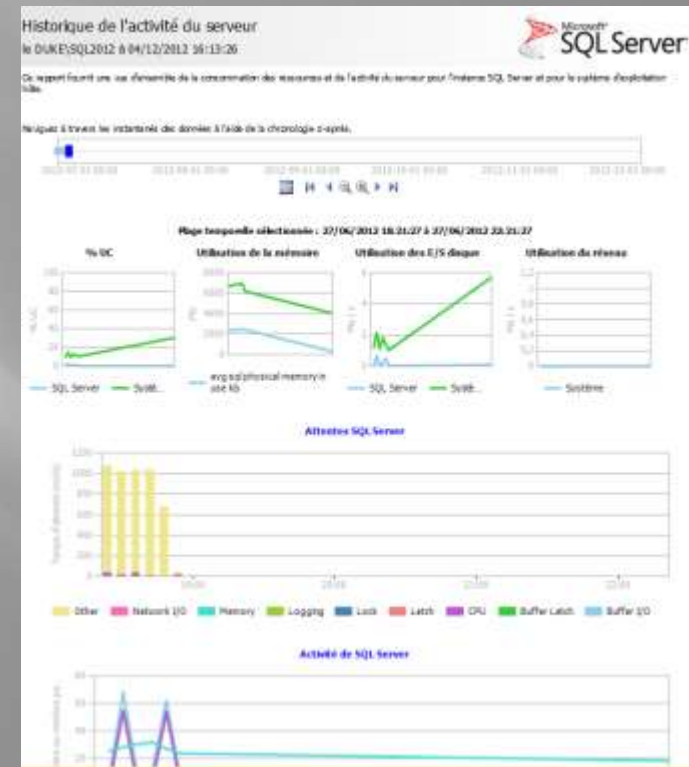
Vie des DMV's

- Accumulation depuis le démarrage du serveur
- Réinitialisation :
 - Redémarrer le serveur
 - DBCC SQLPERF("dmvname",CLEAR)
 - Pour sys.dm_os_wait_stats et sys.dm_os_latch_stats
- Pas de données historiques entre redémarrages !

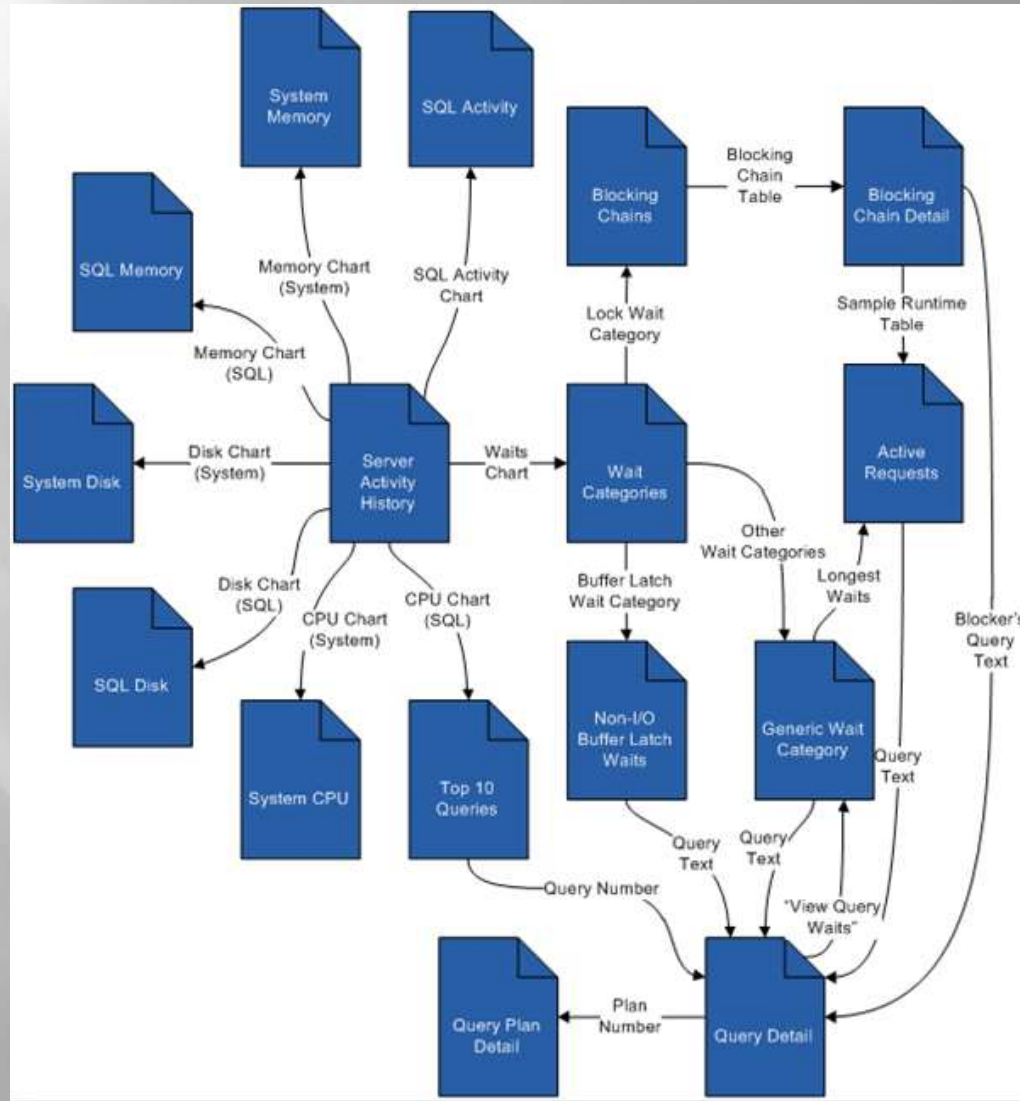


Data collector

- Mécanisme de collecte de données automatisée
 - Compteurs de performance et DMV's
 - Consolide dans une base de données
 - Historise, rétention de plusieurs jours ou semaines
- Rapports historiques
 - Sélection de la plage consultée

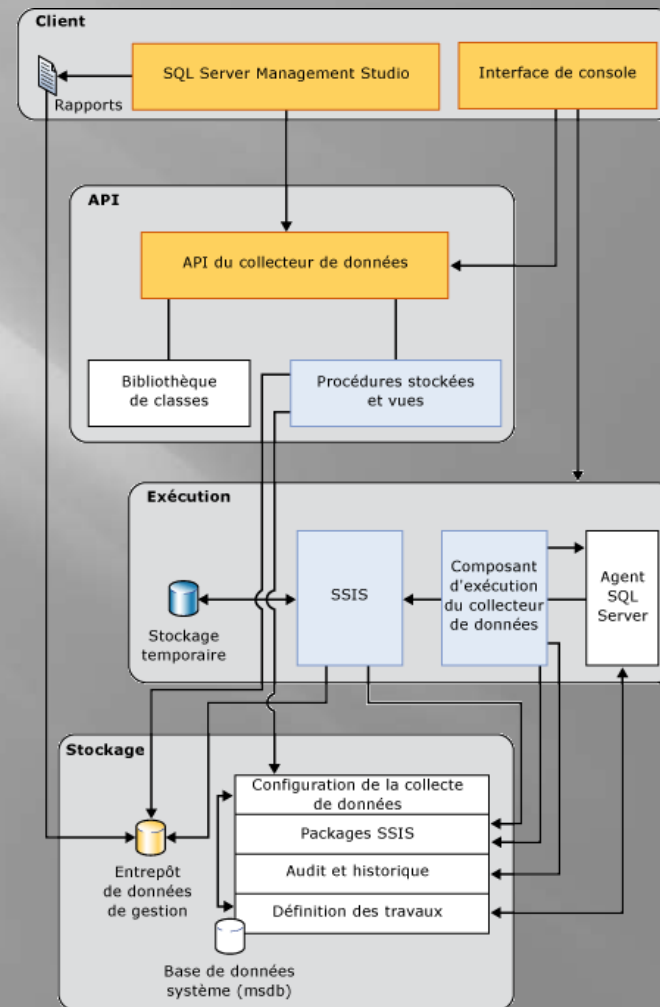


Rapports fournis



Infrastructure

- Infrastructure de collecte **extensible**
 - S'appuie sur SSIS et l'agent SQL
 - Possibilité de créer sa collection



Collecteurs

- Types de collecteurs :
[msdb].[dbo].[syscollector_collector_types]
 - Generic T-SQL Query Collector Type
 - Generic SQL Trace Collector Type
 - Query Activity Collector Type
 - Performance Counters Collector Type




Collecteurs fournis

- Activité du serveur
 - Perf Counters et Generic T-SQL
 - Mise en cache
- Statistiques sur les requêtes
 - Query activity
 - Non mise en cache
- Utilisation du disque
 - Generic T-SQL



Management Datawarehouse

- Possibilité de base centralisée pour plusieurs serveurs à monitorer
- Possibilité d'utiliser la même base pour l'UCP
 - Nom de la base nécessaire pour la configuration de l'UCP : sysutility_mdw
- Volumétrie « pessimiste » : 600 Mo/jour/instance
- Rétention configurable par collecteur
 - 14 jours query / 730 jours disk usage
 - Job de purge  mdw_purge_data_[sysutility_mdw]
 - Commande exec core.sp_purge_data
 - @retention_days=nn



Tables

- sysutility_mdw
 - Schémas de base de données
 - Tables
 - Tables système
 - FileTables
 - core.performance_counter_report_group_iter
 - core.purge_info_internal
 - core.snapshot_timetable_internal
 - core.snapshots_internal
 - core.source_info_internal
 - core.supported_collector_types_internal
 - core.wait_categories
 - core.wait_types
 - custom_snapshots.IndexUsage
 - custom_snapshots.query_hash_stats
 - custom_snapshots.query_hash_text_and_plan
 - custom_snapshots.VolumeUsage
 - dbo.custom_mdw_overview_reports
 - snapshots.active_sessions_and_requests
 - snapshots.disk_usage
 - snapshots.distinct_queries
 - snapshots.distinct_query_to_handle
 - snapshots.io_virtual_file_stats
 - snapshots.log_usage
 - snapshots.notable_query_plan
 - snapshots.notable_query_text
 - snapshots.os_latch_stats
 - snapshots.os_memory_clerks
 - snapshots.os_memory_nodes
 - snapshots.os_process_memory
 - snapshots.os_schedulers
 - snapshots.os_wait_stats
 - snapshots.performance_counter_instances

- snapshots.performance_counter_values
- snapshots.query_stats
- snapshots.sql_process_and_system_memory
- snapshots.sysutility_ucp_batch_manifests_inte
- snapshots.sysutility_ucp_cpu_memory_config
- snapshots.sysutility_ucp_dac_collected_execu
- snapshots.sysutility_ucp_smo_properties_inte
- snapshots.sysutility_ucp_volumes_internal
- snapshots.trace_data
- snapshots.trace_info
- sysutility_ucp_core.computers_internal
- sysutility_ucp_core.cpu_utilization_internal
- sysutility_ucp_core.dacs_internal
- sysutility_ucp_core.databases_internal
- sysutility_ucp_core.datafiles_internal
- sysutility_ucp_core.filegroups_internal
- sysutility_ucp_core.logfiles_internal
- sysutility_ucp_core.smo_servers_internal
- sysutility_ucp_core.space_utilization_internal
- sysutility_ucp_core.volumes_internal
- sysutility_ucp_staging.consistent_batch_mani



Jobs de collecte

- Décaler les planifications pour éviter les jobs simultanés



Créer un collecteur Generic T-SQL

- Obtenir le GUID du type de collecteur

```
DECLARE @collector_type uniqueidentifier
SELECT @collector_type = collector_type_uid
FROM [msdb].[dbo].[syscollector_collector_types]
WHERE name = N'Generic T-SQL Query Collector Type'
```

- Définir/obtenir la planification

```
DECLARE @schedule uniqueidentifier
SELECT @schedule =schedule_uid
FROM [msdb].[dbo].[sysschedules]
WHERE name = N'CollectorSchedule_Every_30min'
```



Créer un collecteur Generic T-SQL

- Créer le collection set
 - [msdb].[dbo].[sp_syscollector_create_collection_set]

```
DECLARE @collectionsetid int
DECLARE @collectionsetuid uniqueidentifier
EXEC [msdb].[dbo].[sp_syscollector_create_collection_set]
@name=N'Index Fragmentation Usage Report',
@collection_mode=1, -- Non-cached mode
@description=N'Records fragmentation of indexes over 100
pages',
@days_until_expiration=400, -- A year and a month
@schedule_uid=@schedule,
@collection_set_id=@collectionsetid OUTPUT,
@collection_set_uid=@collectionsetuid OUTPUT
```



Créer un collecteur Generic T-SQL

- Définir ce qui va être monitoré
 - [msdb].[dbo].[sp_syscollector_create_collection_item]
 - `sp_syscollector_create_collection_item` [@collection_set_id =] collection_set_id , [@collector_type_uid =] 'collector_type_uid' , [@name =] 'name' , [[@frequency =] frequency] , [@parameters =] 'parameters' , [@collection_item_id =] collection_item_id OUTPUT



Créer un collecteur Generic T-SQL

- Définir le paramètre @parameters en XML conforme au schéma

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="DataCollectorType">
  <xs:element name="TSQLQueryCollector">
    <xs:complexType>
      <xs:sequence>
        <xs:sequence>
          <xs:element name="Query" minOccurs="1" maxOccurs="unbounded">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Value" type="xs:string" />
                <xs:element name="OutputTable" type="xs:string" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="Databases" minOccurs="0" maxOccurs="1">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Database" minOccurs="0"
maxOccurs="unbounded" type="xs:string" />
                <xs:attribute name="UseSystemDatabases" type="xs:boolean"
use="optional" />
                <xs:attribute name="UseUserDatabases" type="xs:boolean"
use="optional" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
```

Créer un collecteur Generic T-SQL

```
DECLARE @parameters XML
Set @parameters=CONVERT(XML,N'<ns:TSQLQueryCollector
xmlns:ns="DataCollectorType"><Query><Value>
Select
....
FROM .....
</Value><OutputTable>IndexFragmentation</OutputTable></Query><D
atabases UseUserDatabases="true"/></ns:TSQLQueryCollector>')
SELECT @parameters
DECLARE @collectionitem INT
EXEC [msdb].[dbo].[sp_syscollector_create_collection_item]
@name=N'Index Fragmentation',
@parameters=@parameters,
@collection_item_id=@collectionitem OUTPUT,
@collection_set_id=@collectionsetid,
@collector_type_uid=@collector_type
```



Table de snapshot

- Créée automatiquement à la collecte
 - Mais non détruite si le collecteur est supprimé
- Schéma custom_snapshots

```
[-] [-] custom_snapshots.IndexFragmentation
      [-] [-] Colonnes
            [-] [-] IndexName (nvarchar(515), NULL)
            [-] [-] FragmentationPercentage (float, NULL)
            [-] [-] database_name (nvarchar(128), NULL)
            [-] [-] collection_time (datetimeoffset(7), NULL)
            [-] [-] snapshot_id (FK, int, NULL)
```



Démo

- Créer un collecteur Generic TSQL



Collecteur SQLTrace

- Configurer une trace dans le profiler
- Exporter le script
 - Pour le jeu d'éléments de collecte TraceSQL
- Modifier (nom du set et de l'item) puis exécuter le script

- Attention aux problèmes de permissions du service sur le dossier C:\Program Files\Microsoft SQL Server\MSSQL11.SQL2012\MSSQL\JOBS\UtilityDC

- Tables : snapshots.trace_info et snapshots.trace_data



Collecteur Perf Counters

- Voir le script de création provenant de l'activité serveur
- Schéma :

```
<ns:PerformanceCountersCollector xmlns:ns="DataCollectorType">
  <PerformanceCounters
    Objects="SQLServer:Locks"
    Counters="Lock *"
    Instances="_Total"
  />
  <PerformanceCounters
    Objects="SQLServer:SQL Statistics"
    Counters="*/sec"
  />
  <PerformanceCounters
    Objects="Processor"
    Counters="% Processor Time"
    Instances="*"
  />
</ns:PerformanceCountersCollector>
```



Documentation

- **BOL**
- **White Paper** : Using Management Data Warehouse for Performance Monitoring
- **Exemple sur CodePlex** : Query Hash Statistics



