



Creating a Centralized Performance Collection Solution for SQL Server



Free MSSQLTips Webinar: [Azure SQL Database Performance Monitoring and Optimization](#)

Many organizations have already deployed or plan to deploy databases in the cloud, both in Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) implementations. People who move to the cloud might think everything's completely hands-off, but monitoring and optimization is even more critical because you share resources and have no control over the infrastructure. So, how do you address your SQL Server performance challenges?

Problem

The SQL Server query cache contains valuable performance statistics and execution plans of recently executed query workloads that can be useful to tuning your server. Many DBA's use this information on a daily basis for performance analysis and optimization. While this information is easily obtained using DMV's, a thing to note is that it gets wiped out on each SQL Server restart or partially lost when the server is under memory pressure.

Another problem is that execution plans are in XML format. That means some diagnostic queries which require content analysis of execution plans may need shredding of multiple XML files which could be quite resource intensive and sometimes not desirable to be run on production servers. So, it would be very useful to collect and analyze this information on a centralized non-production server where we could create XML indexes to facilitate more efficient XML parsing.

Once collected on a central server, we could create various reports to show top 'N' underperforming queries, average executions stats, historical trends, etc. What's more important is that this would allow us to analyze performance data at the enterprise

Creating a Centralized Performance Collection Solution for SQL Server



Free MSSQLTips Webinar: [Azure SQL Database Performance Monitoring and Optimization](#)

Many organizations have already deployed or plan to deploy databases in the cloud, both in Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) implementations. People who move to the cloud might think everything's completely hands-off, but monitoring and optimization is even more critical because you share resources and have no control over the infrastructure. So, how do you address your SQL Server performance challenges?

Problem

The SQL Server query cache contains valuable performance statistics and execution plans of recently executed query workloads that can be useful to tuning your server. Many DBA's use this information on a daily basis for performance analysis and optimization. While this information is easily obtained using DMV's, a thing to note is that it gets wiped out on each SQL Server restart or partially lost when the server is under memory pressure.

Another problem is that execution plans are in XML format. That means some diagnostic queries which require content analysis of execution plans may need shredding of multiple XML files which could be quite resource intensive and sometimes not desirable to be run on production servers. So, it would be very useful to collect and analyze this information on a centralized non-production server where we could create XML indexes to facilitate more efficient XML parsing.

Once collected on a central server, we could create various reports to show top 'N' underperforming queries, average executions stats, historical trends, etc. What's more important is that this would allow us to analyze performance data at the enterprise

Creating a Centralized Performance Collection Solution for SQL Server



Free MSSQLTips Webinar: [Azure SQL Database Performance Monitoring and Optimization](#)

Many organizations have already deployed or plan to deploy databases in the cloud, both in Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) implementations. People who move to the cloud might think everything's completely hands-off, but monitoring and optimization is even more critical because you share resources and have no control over the infrastructure. So, how do you address your SQL Server performance challenges?

Problem

The SQL Server query cache contains valuable performance statistics and execution plans of recently executed query workloads that can be useful to tuning your server. Many DBA's use this information on a daily basis for performance analysis and optimization. While this information is easily obtained using DMV's, a thing to note is that it gets wiped out on each SQL Server restart or partially lost when the server is under memory pressure.

Another problem is that execution plans are in XML format. That means some diagnostic queries which require content analysis of execution plans may need shredding of multiple XML files which could be quite resource intensive and sometimes not desirable to be run on production servers. So, it would be very useful to collect and analyze this information on a centralized non-production server where we could create XML indexes to facilitate more efficient XML parsing.

Once collected on a central server, we could create various reports to show top 'N' underperforming queries, average executions stats, historical trends, etc. What's more important is that this would allow us to analyze performance data at the enterprise

Creating a Centralized Performance Collection Solution for SQL Server



Free MSSQLTips Webinar: [Azure SQL Database Performance Monitoring and Optimization](#)

Many organizations have already deployed or plan to deploy databases in the cloud, both in Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) implementations. People who move to the cloud might think everything's completely hands-off, but monitoring and optimization is even more critical because you share resources and have no control over the infrastructure. So, how do you address your SQL Server performance challenges?

Problem

The SQL Server query cache contains valuable performance statistics and execution plans of recently executed query workloads that can be useful to tuning your server. Many DBA's use this information on a daily basis for performance analysis and optimization. While this information is easily obtained using DMV's, a thing to note is that it gets wiped out on each SQL Server restart or partially lost when the server is under memory pressure.

Another problem is that execution plans are in XML format. That means some diagnostic queries which require content analysis of execution plans may need shredding of multiple XML files which could be quite resource intensive and sometimes not desirable to be run on production servers. So, it would be very useful to collect and analyze this information on a centralized non-production server where we could create XML indexes to facilitate more efficient XML parsing.

Once collected on a central server, we could create various reports to show top 'N' underperforming queries, average executions stats, historical trends, etc. What's more important is that this would allow us to analyze performance data at the enterprise

Creating a Centralized Performance Collection Solution for SQL Server



Free MSSQLTips Webinar: [Azure SQL Database Performance Monitoring and Optimization](#)

Many organizations have already deployed or plan to deploy databases in the cloud, both in Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) implementations. People who move to the cloud might think everything's completely hands-off, but monitoring and optimization is even more critical because you share resources and have no control over the infrastructure. So, how do you address your SQL Server performance challenges?

Problem

The SQL Server query cache contains valuable performance statistics and execution plans of recently executed query workloads that can be useful to tuning your server. Many DBA's use this information on a daily basis for performance analysis and optimization. While this information is easily obtained using DMV's, a thing to note is that it gets wiped out on each SQL Server restart or partially lost when the server is under memory pressure.

Another problem is that execution plans are in XML format. That means some diagnostic queries which require content analysis of execution plans may need shredding of multiple XML files which could be quite resource intensive and sometimes not desirable to be run on production servers. So, it would be very useful to collect and analyze this information on a centralized non-production server where we could create XML indexes to facilitate more efficient XML parsing.

Once collected on a central server, we could create various reports to show top 'N' underperforming queries, average executions stats, historical trends, etc. What's more important is that this would allow us to analyze performance data at the enterprise

- [Download all of the scripts for this solution \(https://www.mssqltips.com/tipimages2/4264_CreatePerfStatsCollectionJobs.zip\)](https://www.mssqltips.com/tipimages2/4264_CreatePerfStatsCollectionJobs.zip)

Creating a Centralized Performance Collection Solution for SQL Server



Free MSSQLTips Webinar: [Azure SQL Database Performance Monitoring and Optimization](#)

Many organizations have already deployed or plan to deploy databases in the cloud, both in Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) implementations. People who move to the cloud might think everything's completely hands-off, but monitoring and optimization is even more critical because you share resources and have no control over the infrastructure. So, how do you address your SQL Server performance challenges?

Problem

The SQL Server query cache contains valuable performance statistics and execution plans of recently executed query workloads that can be useful to tuning your server. Many DBA's use this information on a daily basis for performance analysis and optimization. While this information is easily obtained using DMV's, a thing to note is that it gets wiped out on each SQL Server restart or partially lost when the server is under memory pressure.

Another problem is that execution plans are in XML format. That means some diagnostic queries which require content analysis of execution plans may need shredding of multiple XML files which could be quite resource intensive and sometimes not desirable to be run on production servers. So, it would be very useful to collect and analyze this information on a centralized non-production server where we could create XML indexes to facilitate more efficient XML parsing.

Once collected on a central server, we could create various reports to show top 'N' underperforming queries, average executions stats, historical trends, etc. What's more important is that this would allow us to analyze performance data at the enterprise

Creating a Centralized Performance Collection Solution for SQL Server



Free MSSQLTips Webinar: [Azure SQL Database Performance Monitoring and Optimization](#)

Many organizations have already deployed or plan to deploy databases in the cloud, both in Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) implementations. People who move to the cloud might think everything's completely hands-off, but monitoring and optimization is even more critical because you share resources and have no control over the infrastructure. So, how do you address your SQL Server performance challenges?

Problem

The SQL Server query cache contains valuable performance statistics and execution plans of recently executed query workloads that can be useful to tuning your server. Many DBA's use this information on a daily basis for performance analysis and optimization. While this information is easily obtained using DMV's, a thing to note is that it gets wiped out on each SQL Server restart or partially lost when the server is under memory pressure.

Another problem is that execution plans are in XML format. That means some diagnostic queries which require content analysis of execution plans may need shredding of multiple XML files which could be quite resource intensive and sometimes not desirable to be run on production servers. So, it would be very useful to collect and analyze this information on a centralized non-production server where we could create XML indexes to facilitate more efficient XML parsing.

Once collected on a central server, we could create various reports to show top 'N' underperforming queries, average executions stats, historical trends, etc. What's more important is that this would allow us to analyze performance data at the enterprise