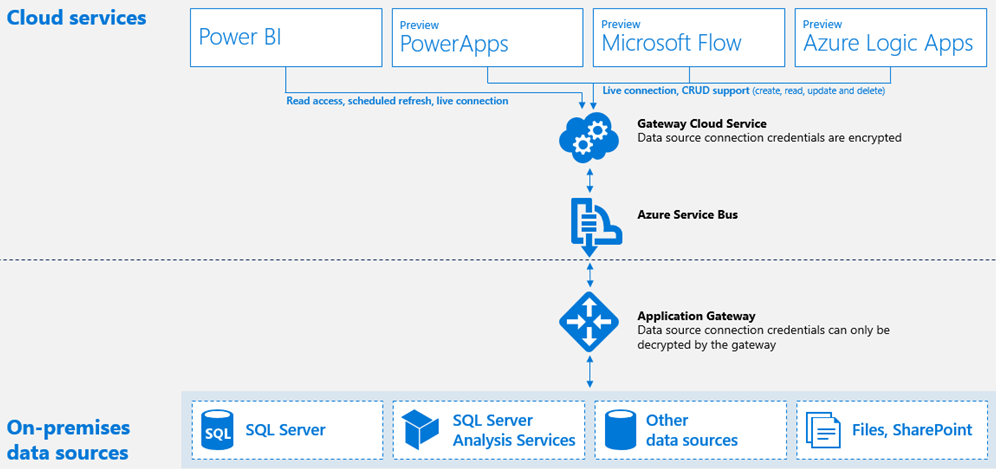
Monitoring On-Premises Data Gateways

# **Summary**

The purpose of this paper is to describe an approach for collecting and analyzing performance counters associated with the On-Premises Data Gateway. Detailed monitoring of the gateway-specific counters as well as resource usage of the gateway server is recommended to guide gateway infrastructure decisions such as scaling up or down an existing gateway server or installing multiple gateways on separate servers to meet current and projected workloads between MS Azure Services and On-Premises data sources.

## **Background**

The On-Premises Data Gateway ([On Prem Gateway](https://powerbi.microsoft.com/en-us/gateway/)), formerly referred to as the Power BI enterprise gateway, is a service that runs in on-premises environments that supports data transfer between On-Premises data sources and MS Azure services. Per the [July Announcement](https://powerbi.microsoft.com/en-us/blog/gateways-july-update-going-beyond-power-bi/), the On-premises gateway can now be leveraged by MS Business Application Platform services and Azure Logic Apps.

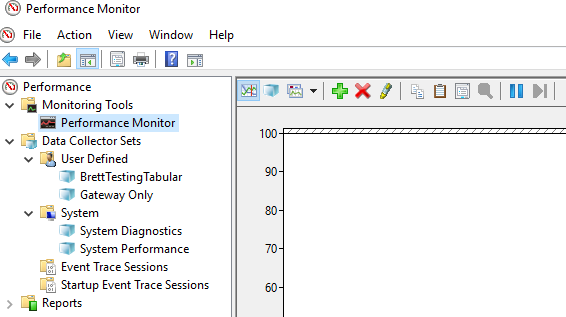


A single On-Premises gateway can be configured with multiple data sources, multiple users (with rights to the given source via the gateway), and multiple gateway administrators. The recommended starting point for an on-premises gateway server ([Gateway Requirements](https://powerbi.microsoft.com/en-us/documentation/powerbi-gateway-onprem/)) is currently 8 CPU Cores and 8 GB of memory but actual resource requirements can vary dramatically based on the type of data sources configured (Scheduled Refresh/Import, Direct Query, and Live Connections to SSAS instances) as well as usage of the given source by consumers (e.g. Power BI dashboards) and scheduled refresh activity.

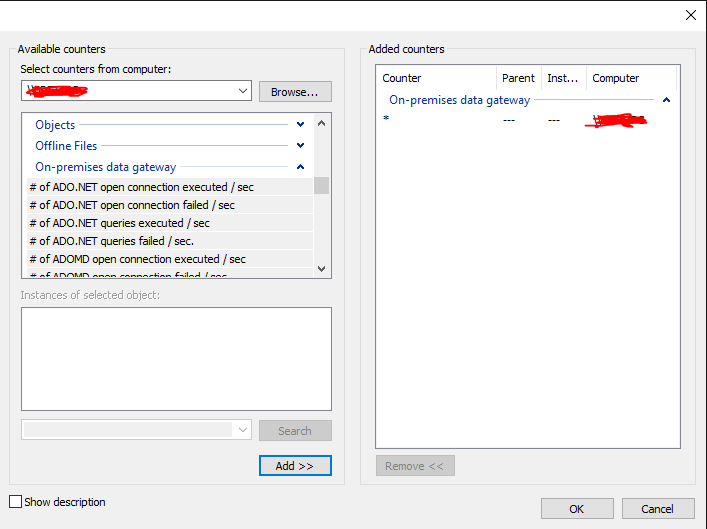
For example, the transmission of a DAX query from a report in the Power BI Service to an on-premises SSAS instance via Live Connection requires far less resources than the scheduled refresh of a large dataset. BI/IT organizations with hybrid architectures must therefore plan to align on-premises data source refresh and query workloads with the necessary gateway servers and, as High Availability is not currently available ([Gateway FAQ](https://powerbi.microsoft.com/en-us/documentation/powerbi-gateway-onprem-faq/)), should plan to use the Recovery Key to quickly restore a gateway to backup servers should a primary gateway server become unavailable.

## **Gateway Performance Counters**

[Performance Counters](https://msdn.microsoft.com/en-us/library/windows/desktop/aa371643(v=vs.85).aspx) provide information on how well a system, application, or service is performing. Individual counters are included in Data Collector Sets which can be scheduled and log files can be exported for further data storage and analysis in other tools. [PowerShell](https://technet.microsoft.com/en-us/library/hh849685.aspx) scripts can be used to obtain and export counter data at specified intervals or the Windows Performance Monitor utility provides an interface for defining, managing, and viewing counters as well.



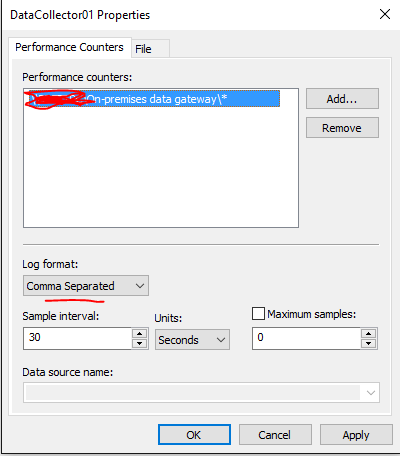
In creating a new collector set for the gateway and optionally its host server via Performance Monitor you can find the ‘On-premises data gateway’ category of counters per the image below.



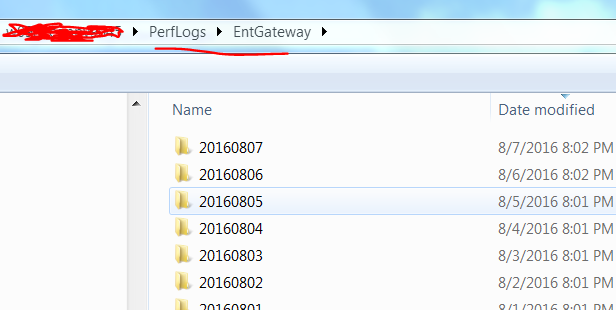
As of 8/7/2016 the following On-premises data gateway counters are available:

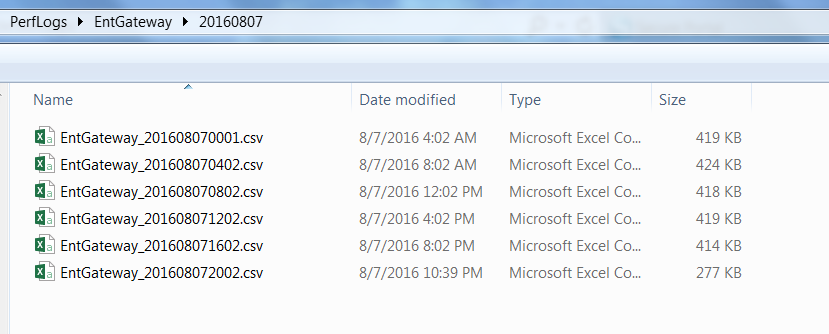
* # of ADO.NET open connection executed / sec
* # of ADO.NET open connection failed /sec
* # of ADO.NET queries executed / sec
* # of ADO.NET queries failed / sec
* # of ADOMD open connection executed / sec
* # of ADOMD open connection failed / sec
* # of ADOMD queries executed / sec
* # of ADOMD queries failed / sec
* # of all open connection executed /sec
* # of all open connection failed / sec
* # of all queries executed / sec
* # of items in the ADO.NET connection pool
* # of items in the OLEDB connection pool
* # of items in the Service Bus pool
* # of Mashup open connection executed / sec
* # of Mashup open connection failed / sec
* # of Mashup queries executed / sec
* # of Mashup queries failed / sec
* # of multiple result set OLEDB queries failed / sec
* # of OLEDB multiple result set queries executed / sec
* # of OLEDB open connection executed / sec
* # of OLEDB open connection failed / sec
* # of OLEDB queries executed / sec
* # of OLEDB queries failed / sec
* # of OLEDB single resultset queries executed / sec
* # of queries failed / sec
* # of single result set OLEDB queries failed / sec

With the counters selected, The Data Collector Set properties can be defined including the interval between samples, log format (CSV, Tab Separated, SQL, Binary), file names and export directory.



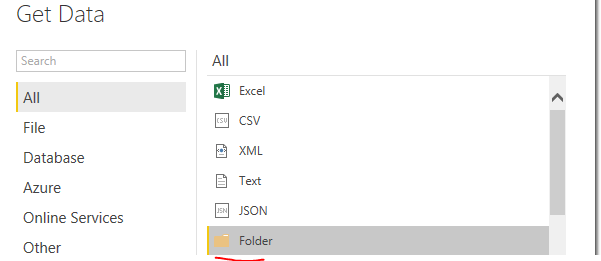
With the collector set scheduled a collection of CSVs can accumulate at a network directory in folders by date such as the following images:



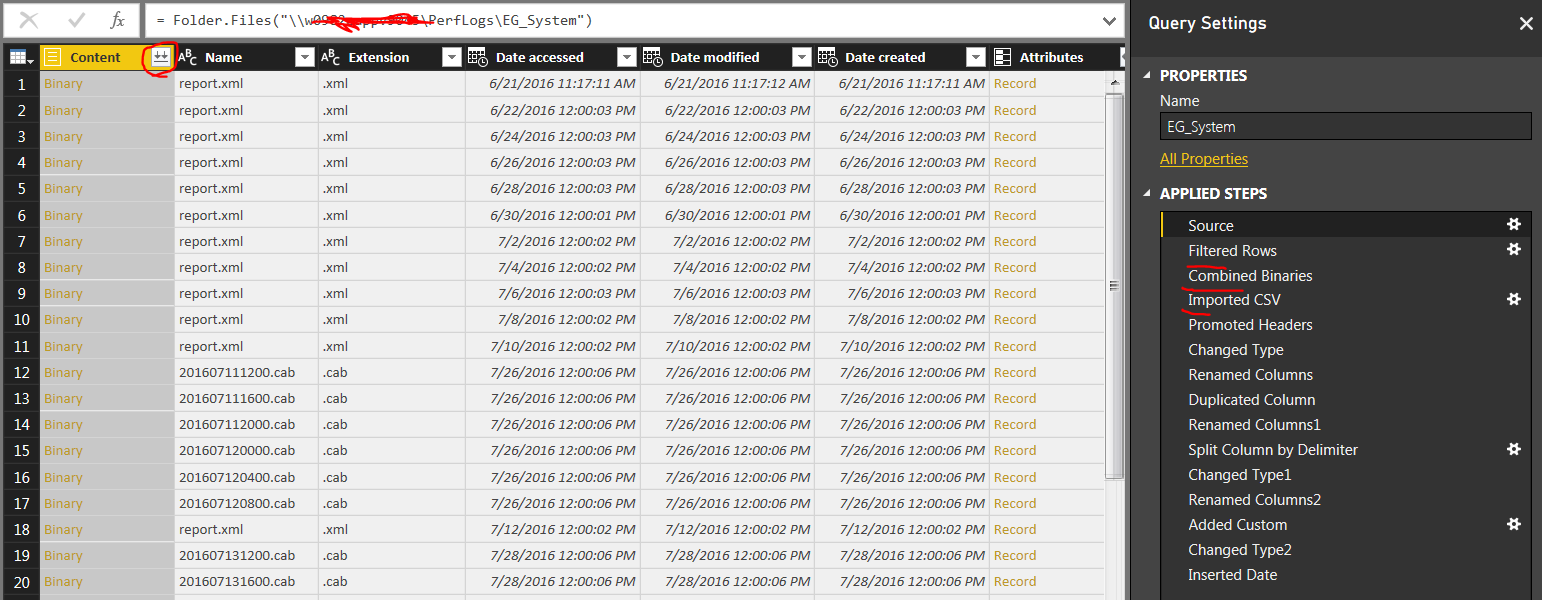


## **Power BI Desktop Retrieval**

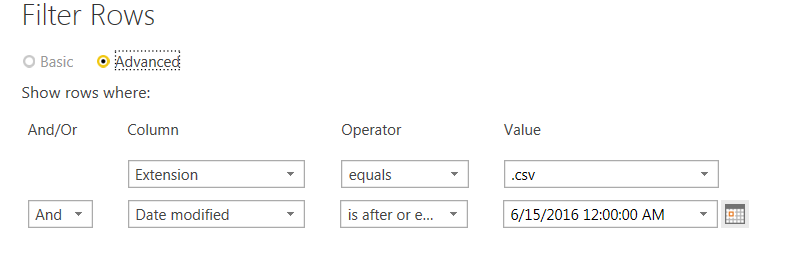
One of the powerful features of Query Editor for Power BI Desktop (and Power Query for Excel) is the ability to consolidate multiple files from a file directory with a common structure into a single query.



After choosing Get Data – Folder – Connect you browse to the parent folder location (or paste in the path):

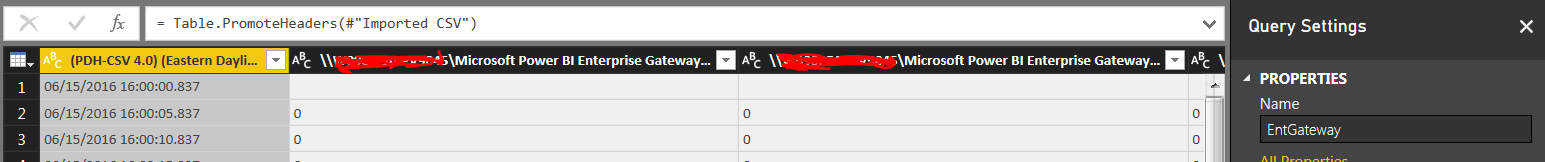


The Folder connection exposes all files in the path (Source step). A filter is then applied to only retrieve CSV files modified after a certain date.

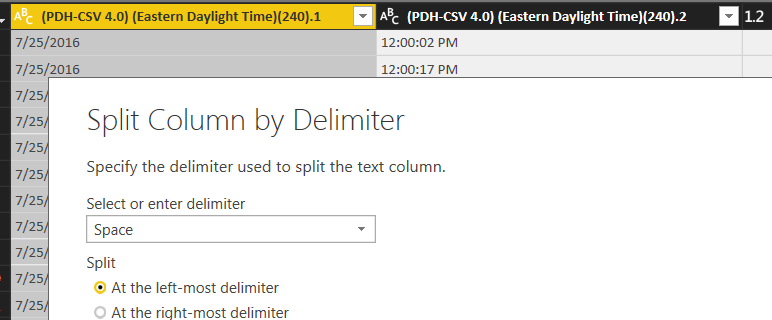


The double arrows circled in the first image indicate the Combine Binaries transform is available to integrate the source files.

Following the combine binaries and after the CSVs have been imported and the headers are promoted the gateway counters are now available for load into the Power BI data model.

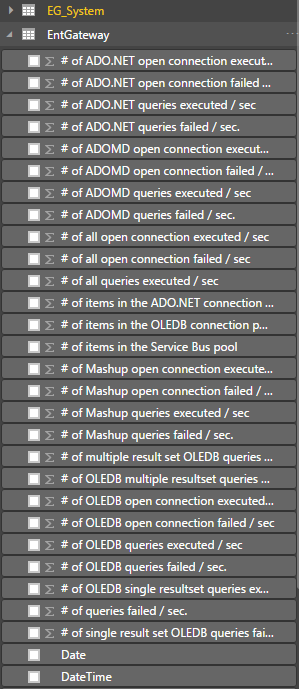


To improve the report design experience further transforms are applied to adjust the data types, rename the columns to remove the server name that’s included in all counters, create a new time column.



\*In this example, a 15 second interval is used and the date/time column is split using the Split Column transform providing access to Time as a distinct column.

Note: For a more robust reporting solution, dedicated date and time dimensions could be added with relationships to the counter data and other modeling metadata could be set.



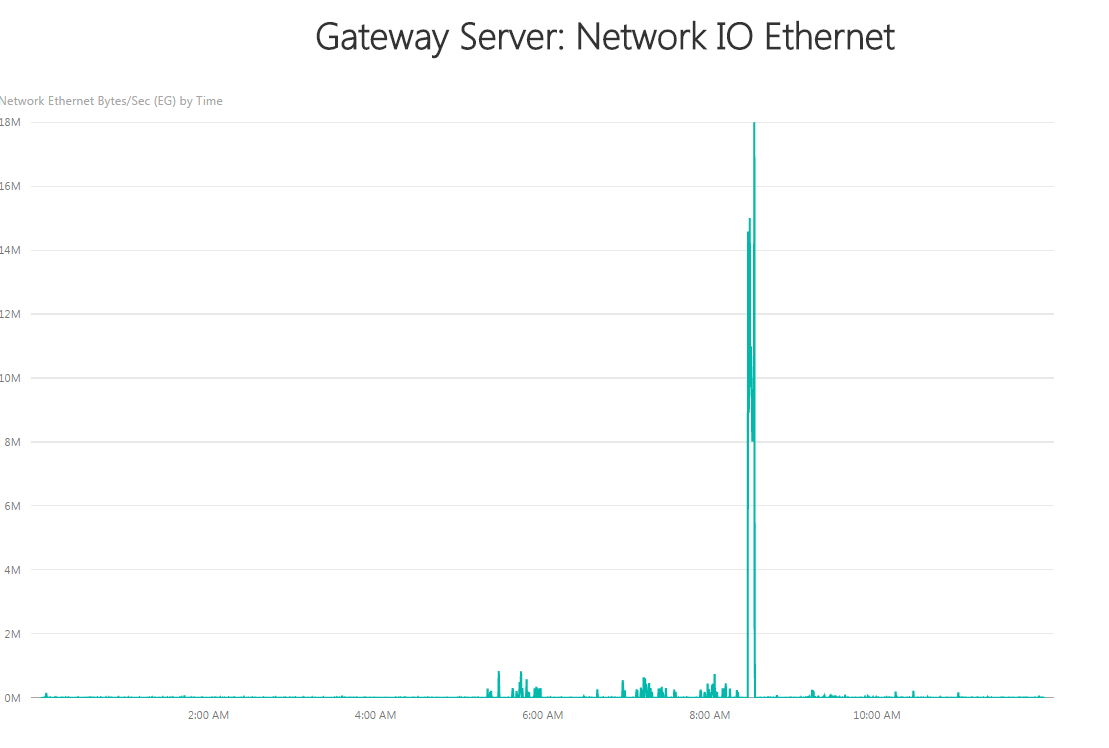
In this example, an additional and separate collector set (EG\_System) contains the following performance counters:

* % Processor Time
* Memory % Committed Bytes in Use
* Network IO Ethernet Bytes Total /Sec.

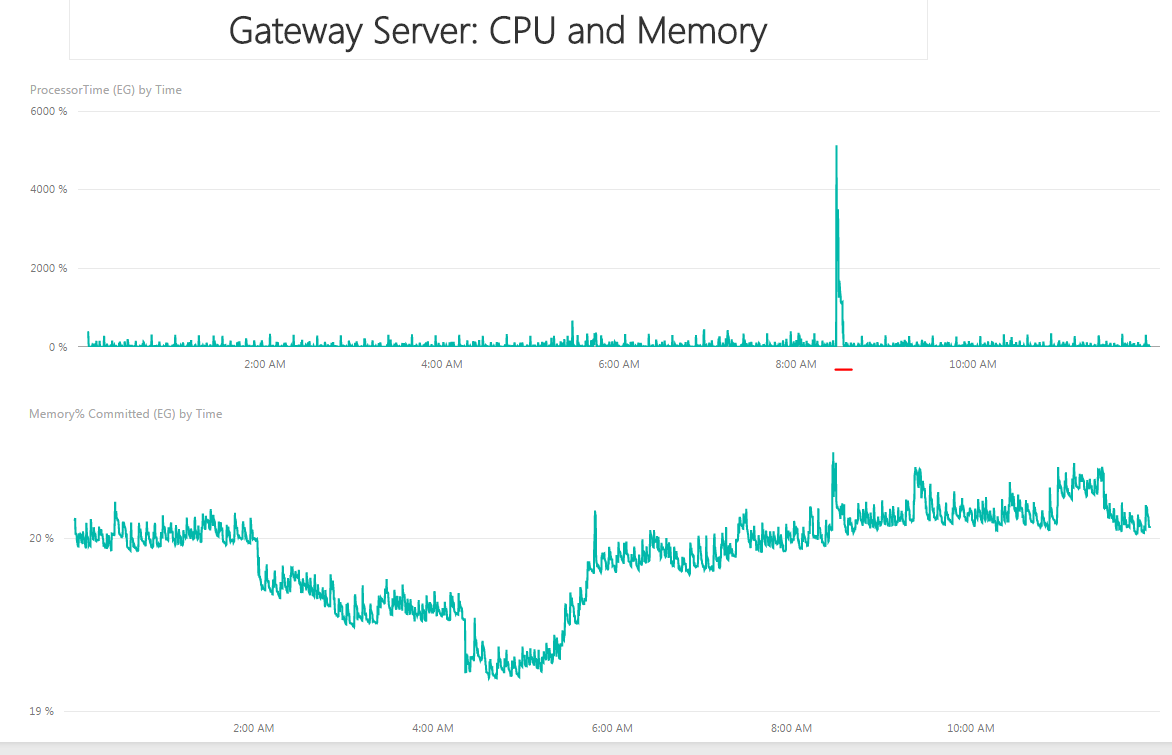
To avoid double-counting values simple DAX measures are applied to these columns using the AVERAGE function.

## **Power BI Desktop Reports**

In the following PBI line chart visual a large spike in Network IO reflected the scheduled refresh of a 400+MB Power BI Desktop file using the gateway at 8:30 AM.

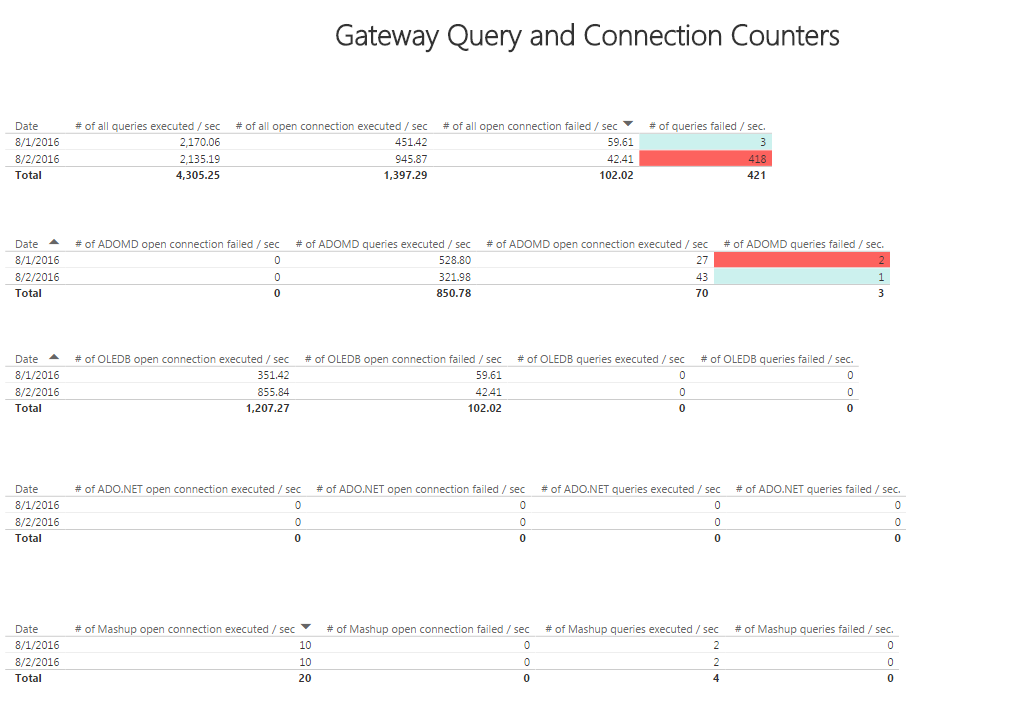


Similar to the Network IO spike, Processor Time on the gateway server also spiked along with a significant increase in memory usage at 8:30 AM.



For the gateway-specific counters, table visualizations with conditional formatting applied to values can help isolate the failures.

Significant declines in the volume of queries executed per second may suggest performance regression for Power BI and other services.

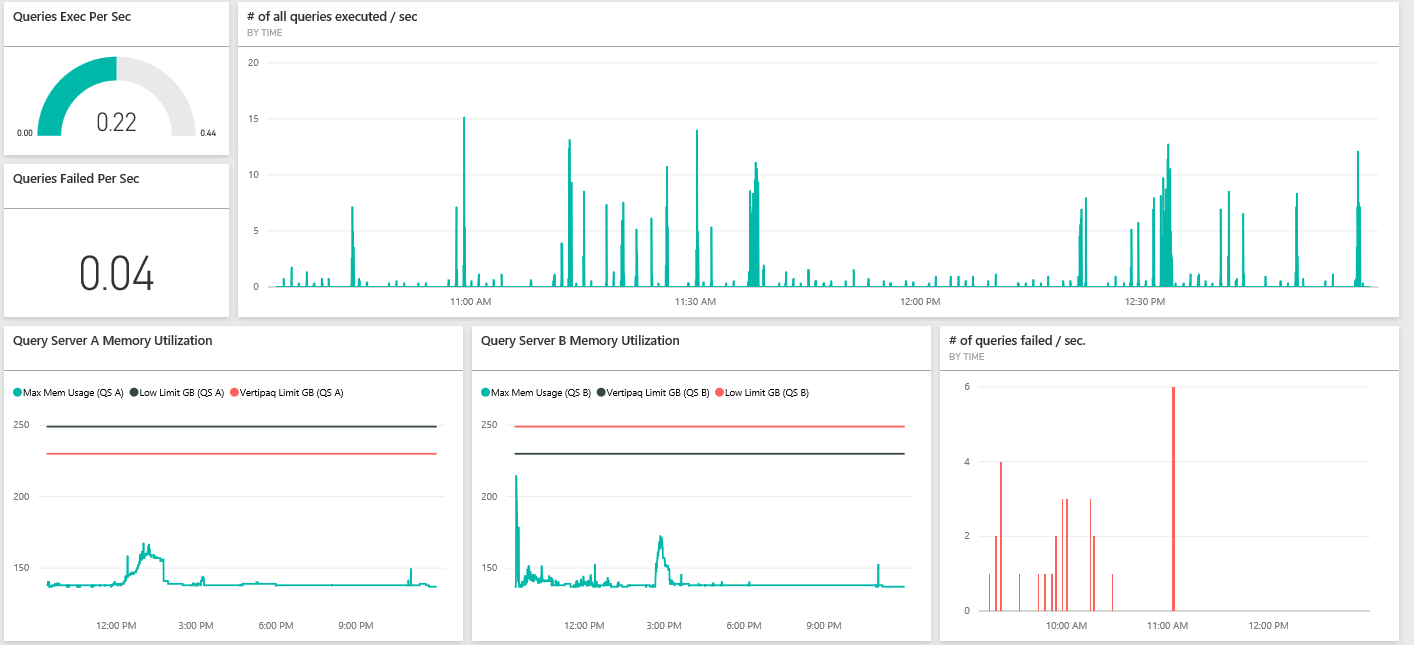


## **Monitoring Enhancements and Collaboration**

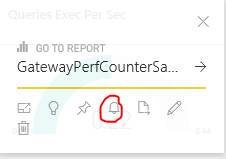
The example in this paper is focused on only one gateway service on one gateway server. Monitoring of additional gateway servers as well as primary BI servers (e.g. SSAS production instances) and the integration of these datasets into a Power BI dashboard is likely necessary to provide a comprehensive view supporting larger scale deployments of Power BI in the Hybrid BI architecture.

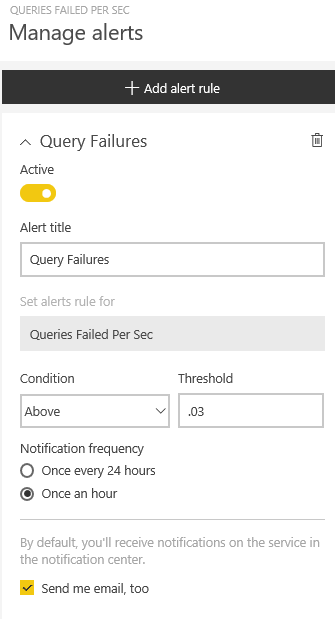
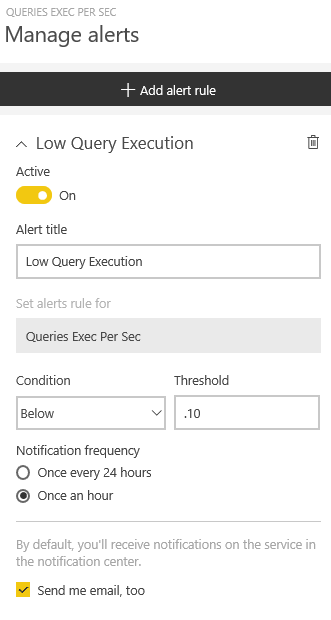
A dedicated Power BI Group Workspace might be helpful for BI/IT team members to access the monitoring content. Additionally, [data driven alerts](https://powerbi.microsoft.com/en-us/blog/always-be-in-the-know-new-and-improved-data-driven-alerts/) with customized rules of gateway counter activity could be configured (e.g. # of queries failed over X) with notifications sent via email. Per the Gateway homepage additional monitoring and auditing for gateway and data sources is coming. It was also suggested at MS Data Insights Summit 2016 that details on the gateway counters and how to best analyze them would also be added to Power BI Documentation.

The following dashboard example provides visibility to the memory usage of two SSAS Tabular query servers relative to current memory limits and the volume of queries executed and failed via gateway counters. Each tile could be linked to a dedicated report or report page with additional detail such as gateway server resource and gateway service detail reports referenced previously.



Alerts can be configured for gauge and card dashboard tiles from mobile or browser interfaces such as the following two examples for Low Query Execution and Query Failures:





The actual metrics or counters to set alerts on and the conditions and thresholds to use depend on many factors (service level, query workload, hardware, etc). Once essential gateway and BI server monitoring is in place, BI/IT teams can collaborate on enhancing the solution with additional data cleansing and integration, analytics, and visualization. Examples of this could include replacing the PBI Desktop Query functions with a SQL Server Integration Services package, adding trend and variance DAX metrics, and leveraging Power BI custom visuals and formatting options.