2015

DBSeer

USER GUIDE

DONG YOUNG YOON

Article I. Table of Contents

DBSeer User Guide

1 About DBSeer

DBSeer is a workload intelligence framework that exploits advanced machine learning and causality techniques to aid the database administrators (DBAs) in his/her various responsibilities.

DBSeer uses machine learning and statistical techniques to identify the bottleneck resources and predict performance for a given set of resources. These features help DBAs decide how to best allocate their budget to various types of resources.

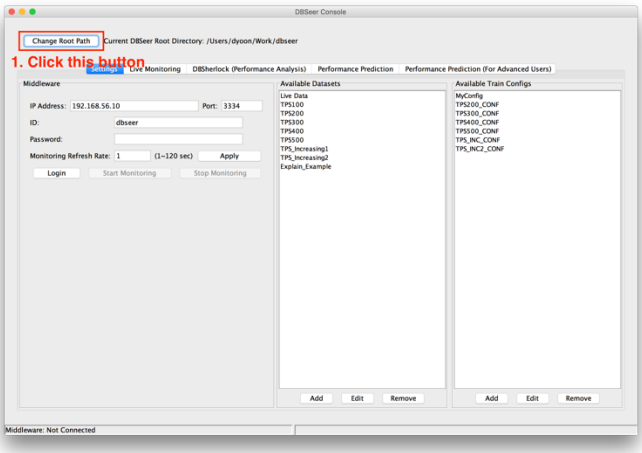
DBSeer also comes with a performance explanation module, called **DBSherlock.** DBSherlock utilizes the statistics collected from the database and the operating system. By combining techniques from outlier detection and causality analysis, DBSherlock assists DBAs in diagnosing performance problems more easily, more accurately, and in a principled manner.

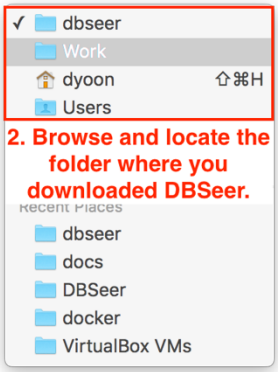
In this guide, we will help you get started with DBSeer after installation to start gathering workload data from your database and analyze them for accurate performance prediction and explanation. Please refer to our **README** document included in the DBSeer package for installation steps. The screenshots in this guide were generated with Mac OS X, but Linux users should be able to follow them without any problems.

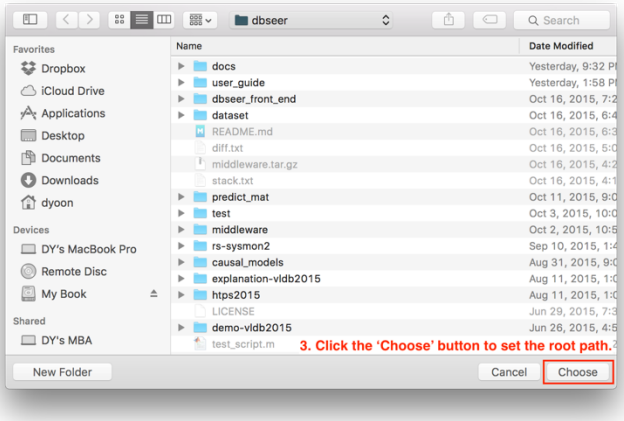
2 Getting Started with DBSeer

2.1 Setting the Root Path

You need to set up the root path for DBSeer when starting it for the first time. This step is necessary for DBSeer to locate its libraries.



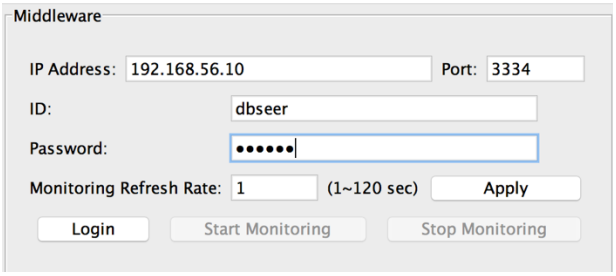




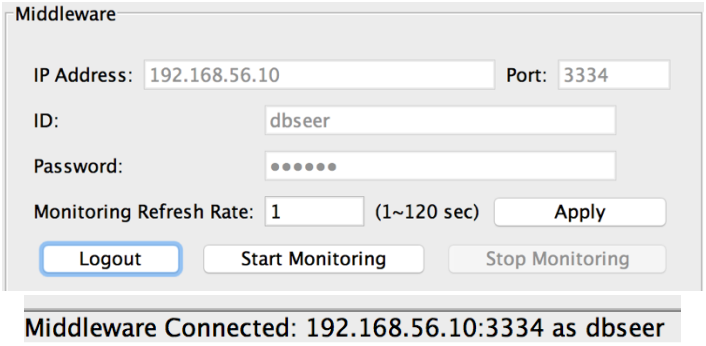
3 Middleware

3.1 Connecting to the Middleware

DBSeer requires the middleware for its collection of monitoring data from OS and DBMS. Our middleware currently only supports a single user and may cause errors if multiple users try to use it concurrently. You can connect to the middleware using *Middleware* panel under the *Settings* tab.

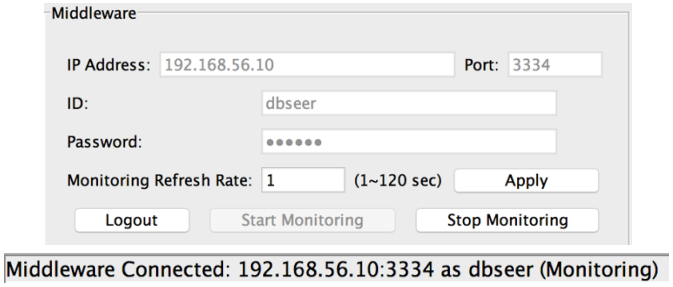


Set IP address and port of the middleware. Then type your ID and password into the corresponding fields and click the *Login* button to connect to the middleware. After you successfully connect to with the middleware, the *Start Monitoring* button will be activated and the status bar at the bottom will display the connection information of the middleware.



3.2 Starting Live Monitoring

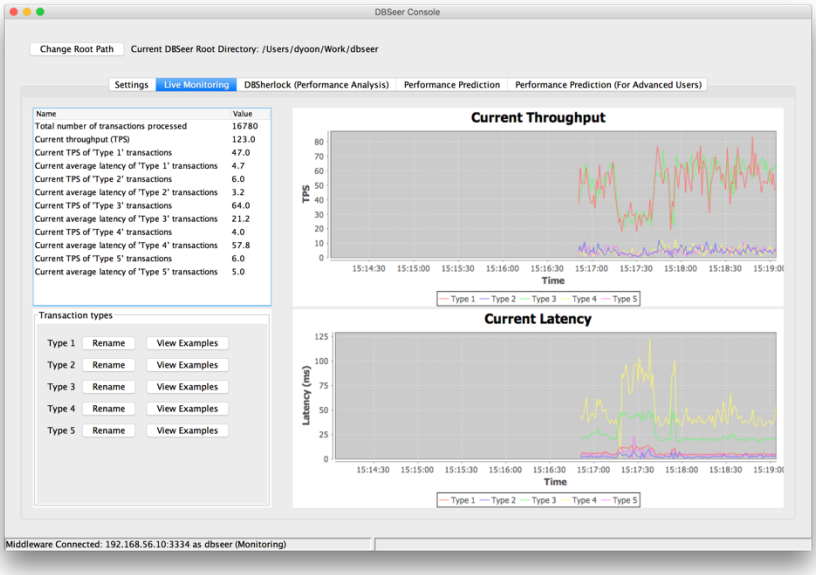
Click the *Start Monitoring* button to start live monitoring. The middleware will start gathering the OS and DBMS statistics.



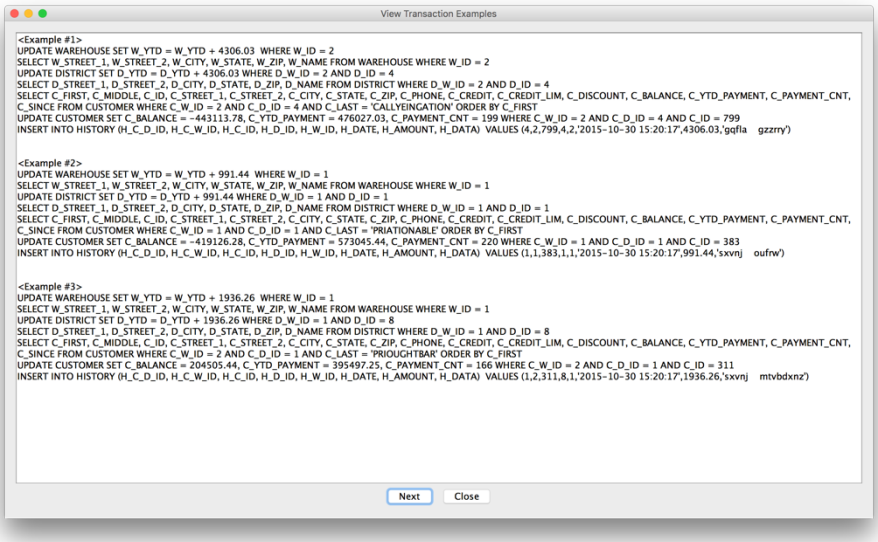
Monitoring Refresh Rate will set the rate of which the transaction statistics under *Live Monitoring* tab is updated. As the middleware gathers the transaction data and transmits it to DBSeer GUI, *Live Monitoring* tab will display the current transaction statistics once enough transaction data is gathered for automatic transaction classification.

3.3 Using the Live Monitoring Tab

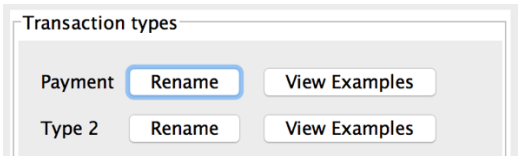
DBSeer will automatically classify transactions by analyzing the table usage and access pattern of each transaction.



You can view actual SQL statements of each classified transaction by clicking the *View Examples* button.

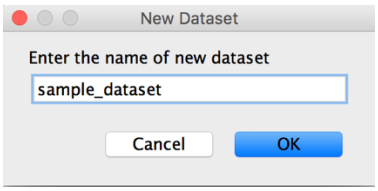


DBAs can look at these transaction examples and identify different types of workload in their system. A classified transaction can be renamed by using the *Rename* button.



3.4 Stopping the Live Monitoring

Click the *Stop Monitoring* button in the *Middleware* panel under the *Setting* tab to stop the monitoring. DBSeer will ask you the name of the dataset, which will be used as the name of the directory where the monitoring data will be stored.



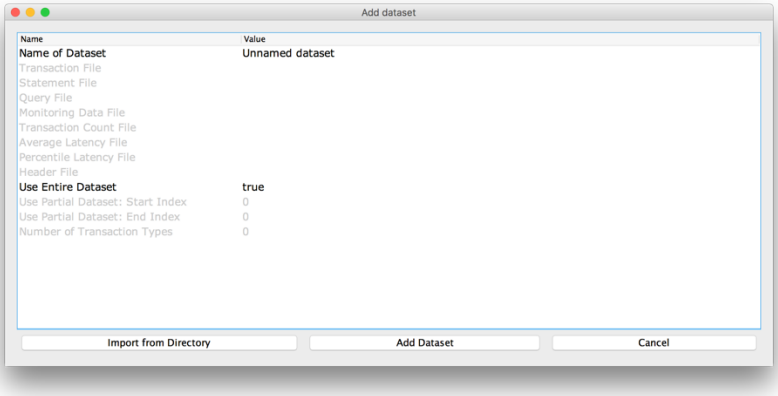
The gathered monitoring data will be stored under *<DBSeer_root_path>/dataset/<name_of_dataset>*. (e.g., */Users/dyoon/Work/dbseer/dataset/sample_dataset*)

4 Managing Dataset and Config

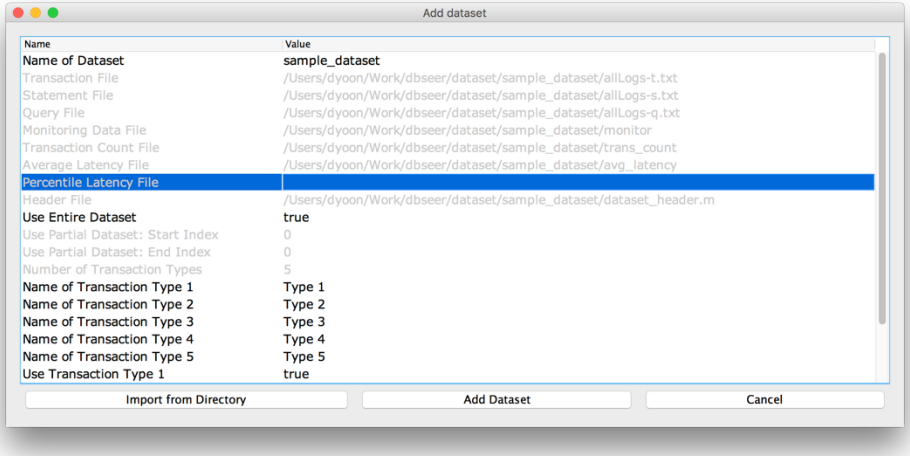
DBSeer manages each monitoring data as a *dataset*. A *config* contains a single dataset with a number of parameters for different performance predictions. You need to manually add a dataset by specifying the directory of the dataset you have previously created from the middleware.

4.1 Adding a Dataset

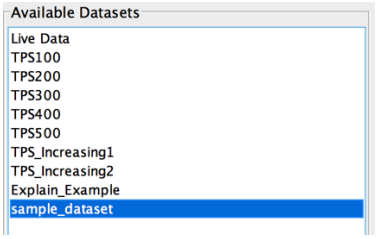
Go to the *Setting* tab. There are *Add*, *Edit* and *Remove* buttons in the *Available Datasets* panel. Click on the *Add* button to add the dataset. The following window will pop up.



Click on the *Import from Directory* and locate the directory of the dataset (e.g., */Users/dyoon/Work/dbseer/dataset/sample_dataset)*

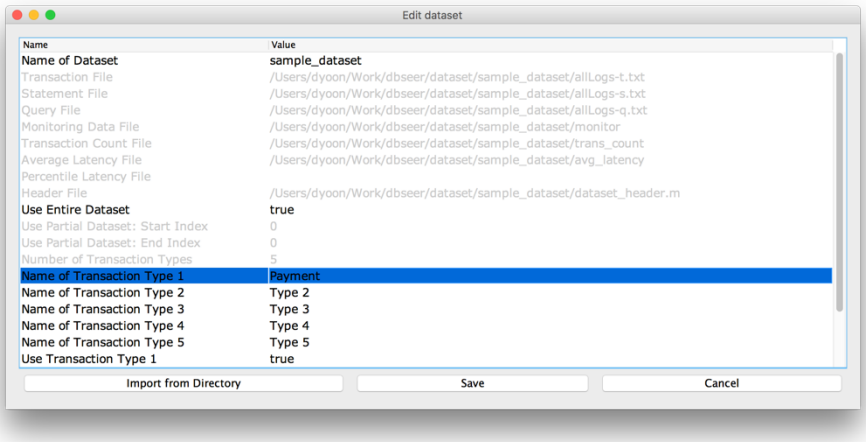


Fields will be automatically filled up. It is okay that *Percentile Latency File* has no values. DBSeer will automatically find the necessary files that were generated during the live monitoring in the directory. Click the *Add Dataset* button to add the dataset. The dataset will be displayed under the Available Datasets panel.



4.2 Editing a Dataset

Select a dataset in the list of *Available Datasets* and click on the *Edit* button to edit the selected dataset.



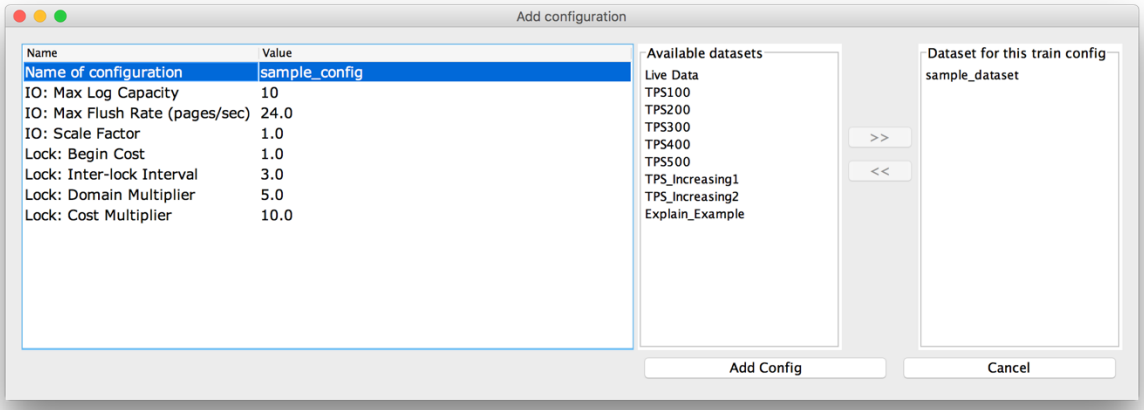
Click on the *Save* button to save the changes to the dataset.

4.3 Deleting a Dataset

Select a dataset in the list of *Available Datasets* and click on the *Remove* button to delete the dataset.

4.4 Adding a Config

Adding a config is similar to adding a dataset. Click on the *Add* button in the *Available Configs* panel.



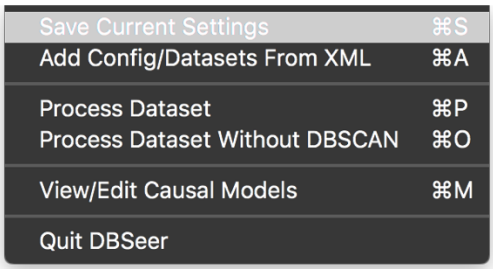
Enter the I/O and lock parameters and select your dataset from the list of *Available Datasets*. Click the *Add Config* button to add the config.

4.5 Editing, Deleting a Config

Editing and Deleting a config are again similar to editing and deleting a dataset.

4.6 Saving Current Settings

Added datasets and configs will be lost when you quit DBSeer unless you save the settings. Go to *File* → *Save Current Settings* to make sure that the details of added datasets and configs are saved to the disk. DBSeer will load the setting file upon its launch and previously added datasets and configs will be reloaded.

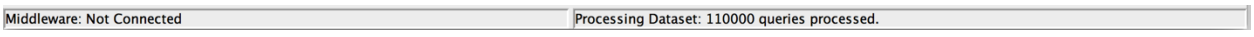


4.7 Processing a Dataset

The dataset collected using the live monitoring feature is being processed on-the-fly as DBSeer receives the data from the middleware. Therefore, you do not require to process the dataset, but you may want to process the dataset separately (e.g., using the data obtained from the middleware manually).

By processing, DBSeer performs a number of tasks to prepare the raw collected data into the processed data that can be used for different performance prediction and explanation features of DBSeer.

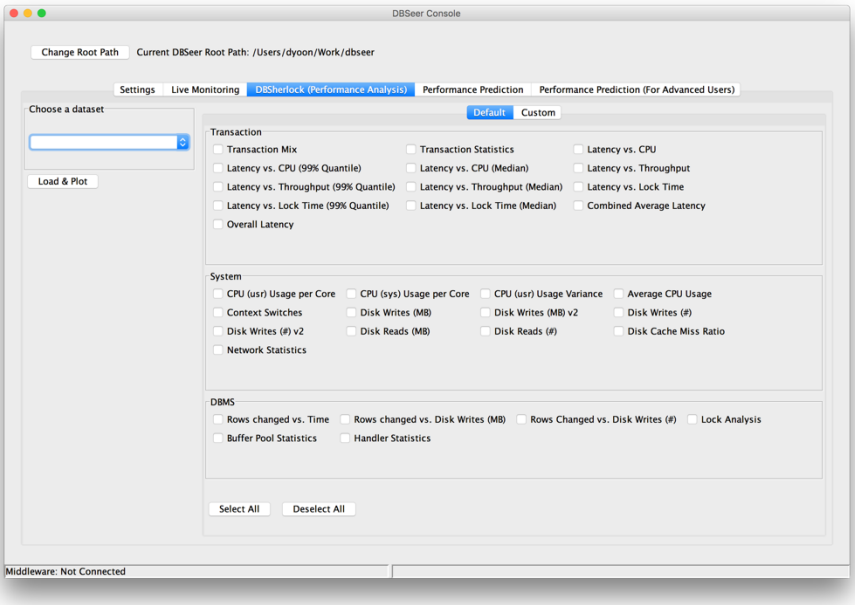
Go to *File* → *Process Dataset* and locate the directory of the dataset. DBSeer will process the dataset. The right status bar will display the progress. Upon its completion, DBSeer will notify you with the pop-up window.



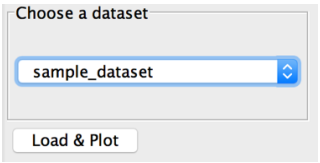
Please note that processing a dataset can take a significant amount of time. The DBSCAN algorithm used for automatic clustering/classification of transactions is likely to cause the large portion of the processing time. For users who want to reduce the processing time and do not require the classification of transactions, we provide an option of processing a dataset without running the DBSCAN algorithm. Go to *File* → *Process Dataset Without DBSCAN* to process a dataset without the DBSCAN. This will finish a lot faster than the normal processing option, but the processed data will not have transaction classifications.

5 Visualizing a Dataset

You can use DBSeer to visualize the dataset you have collected. DBSeer provides a number of default graphs, or you can generate a custom graph by specifying different X and Y axes. Go to the DBSherlock (Performance Analysis) tab for visualization.

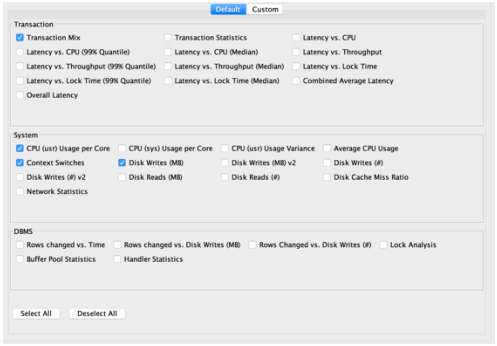


Choose a dataset before generating any plot/graphs. You can choose *Live Data* to generate plot/graphs for live monitoring data (Note: this will only work during the live monitoring).

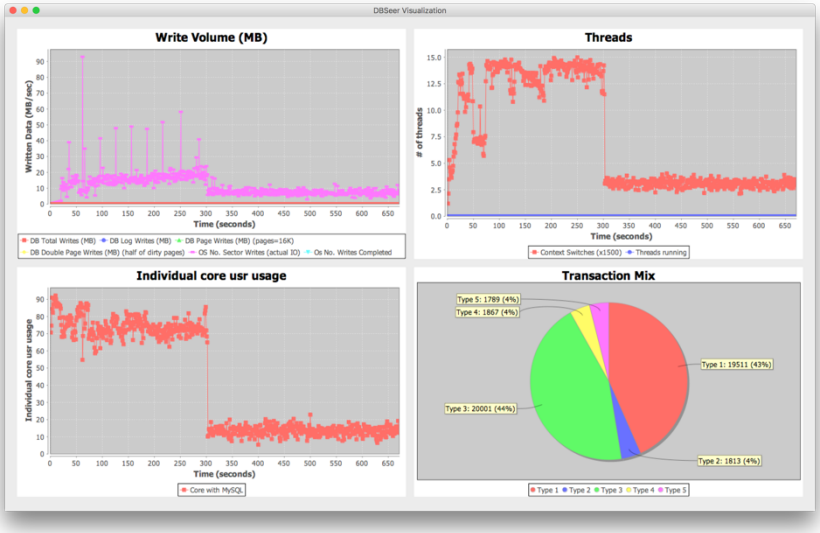


5.1 Plotting Default Plot/Graphs

DBSeer provides a number of default plot/graphs for visualization. Under the *Default* tab, select plot/graphs that you want to visualize. Then click on the *Load & Plot* button.

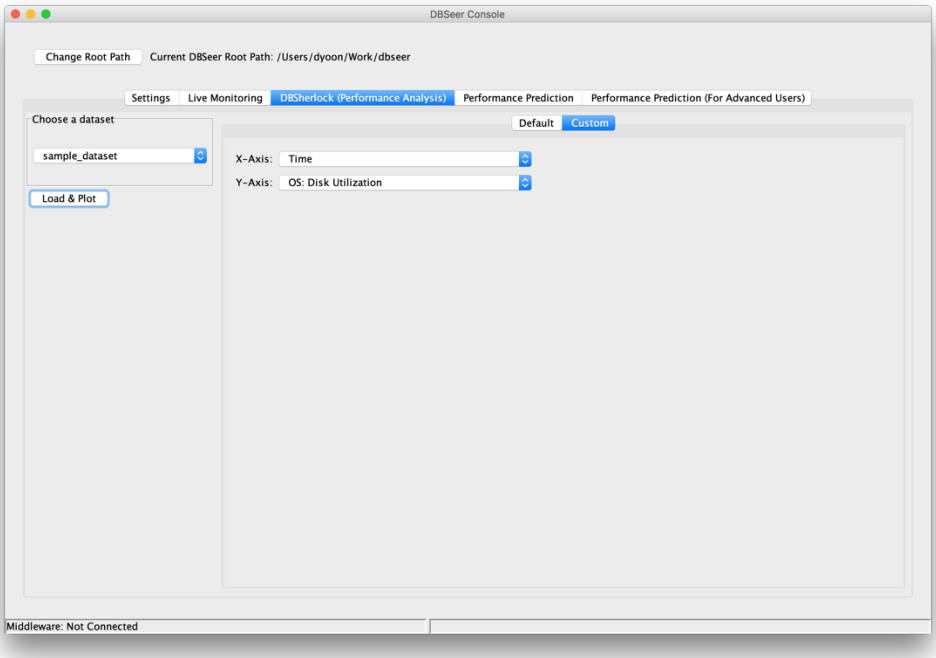


DBSeer will display a visualization in the separate window.

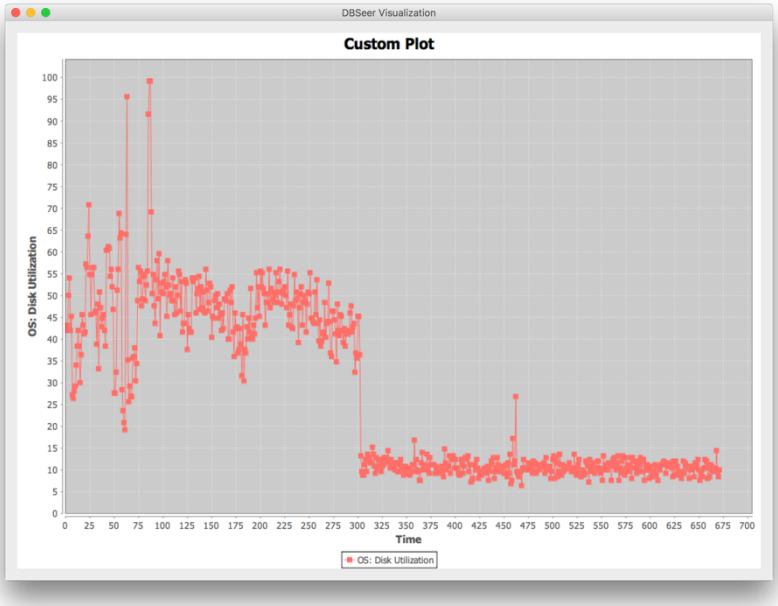


5.2 Plotting Custom Plot/Graphs

Under the *Custom* tab, you can select a custom X and Y axis that suits your needs.



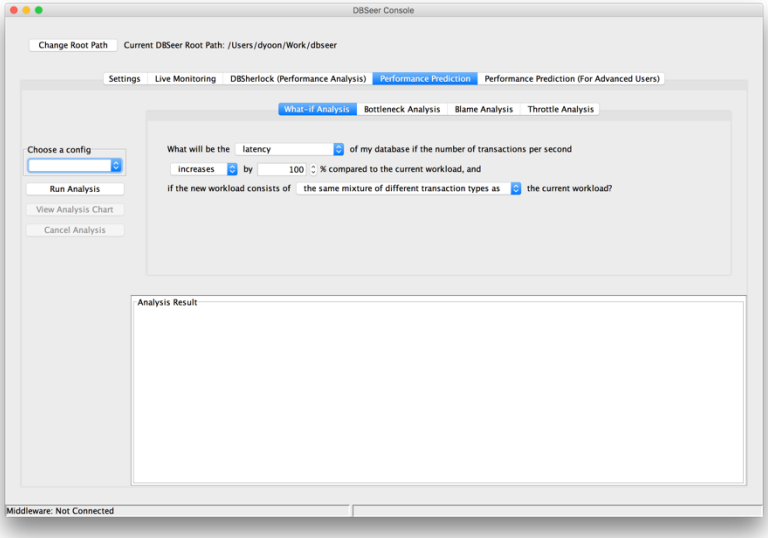
Again, click on the Load & Plot to generate the plot.



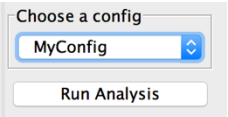
6 Performance Prediction

DBSeer provides a number of performance predictions that can aid DBAs analyzing the workload of their system:

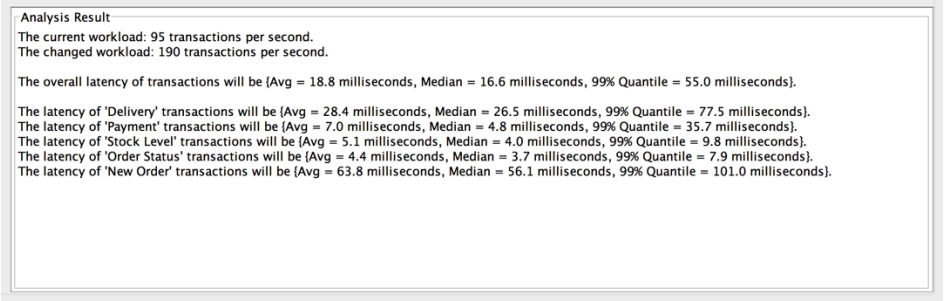
- What-if Analysis: *"What happens if I scale up my workload by 3x?"*
- Bottleneck Analysis: *"Which resource will bottleneck first if my workload increases?"*
- Blame Analysis: *"Which transaction type is the most responsible for the high resource usage?"*
- Throttle Analysis: *"How much should I throttle each transaction type in order to guarantee the latency of X milliseconds 99% of the time?"*



Choose a config before you run any of above analyses.



Running each analysis is straightforward. You select the sub-tab of the analysis that you wish to run and complete the analysis question by selecting appropriate values in provided combo boxes and fields. Click the *Run Analysis* button to let DBSeer perform the anlaysis. The answer will be provided in the *Analysis Result* panel.



6.1 What-If Analysis

You can ask a hypothetical question in order to predict {latency, disk I/O, CPU usage}. You can specify whether your workload increase/decreases and how the mixture of different transaction changes.

6.2 Bottleneck Analysis

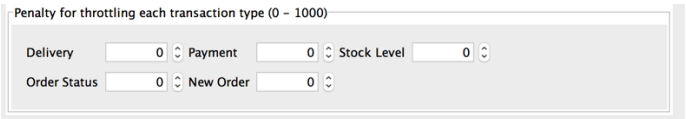
You can ask for a maximum sustainable throughput or the bottleneck resource of your system given the collected statistics.

6.3 Blame Analysis

DBSeer will identify the type of transaction that is most responsible for the high resource usage (e.g., CPU, disk I/O, lock contention).

6.4 Throttle Analysis

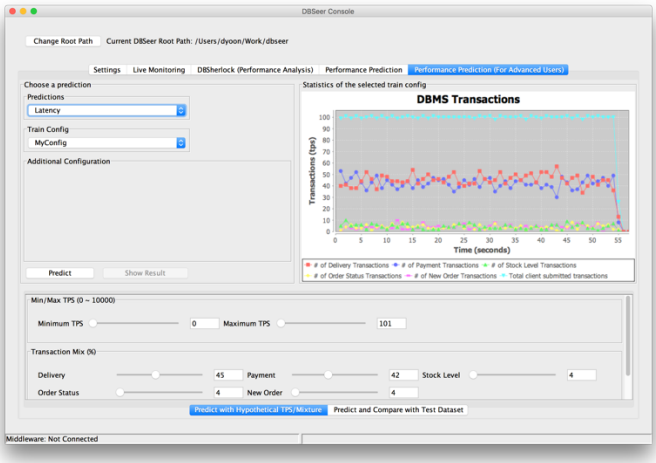
You can ask for how much throttling is required on each transaction type in order to ensure a good performance on a certain type of transaction.



By selecting *individual transactions* in the first combo box, you can specify penalties for throttling each transaction type. DBSeer will use an integer programming model to compute the optimal throttling of transactions given the penalty.

7 Performance Prediction (Advanced)

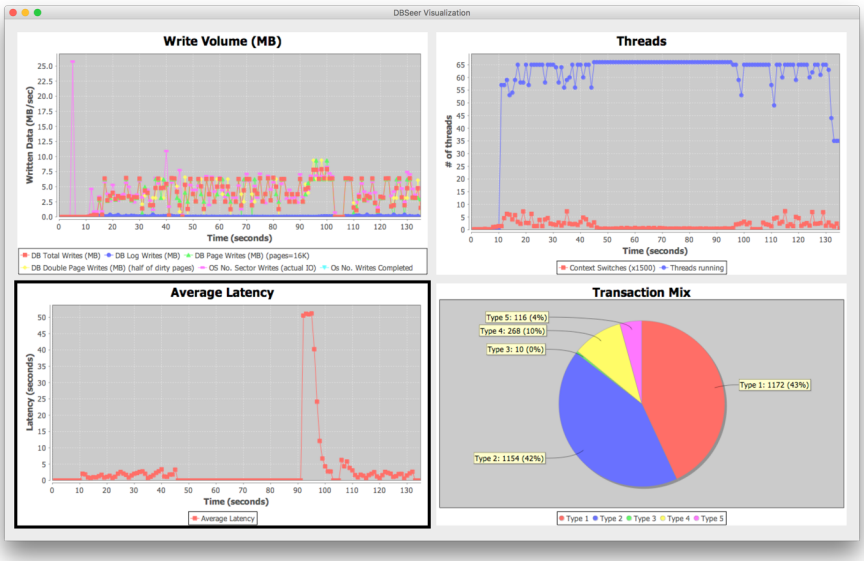
DBSeer provides an option for advanced users to manually choose from every performance prediction model available and run as desired. Go to Performance Prediction (For Advanced Users) for this feature. We will not go into details on using this feature in this usage guide. Please feel free to contact us if you have any questions on using this feature.



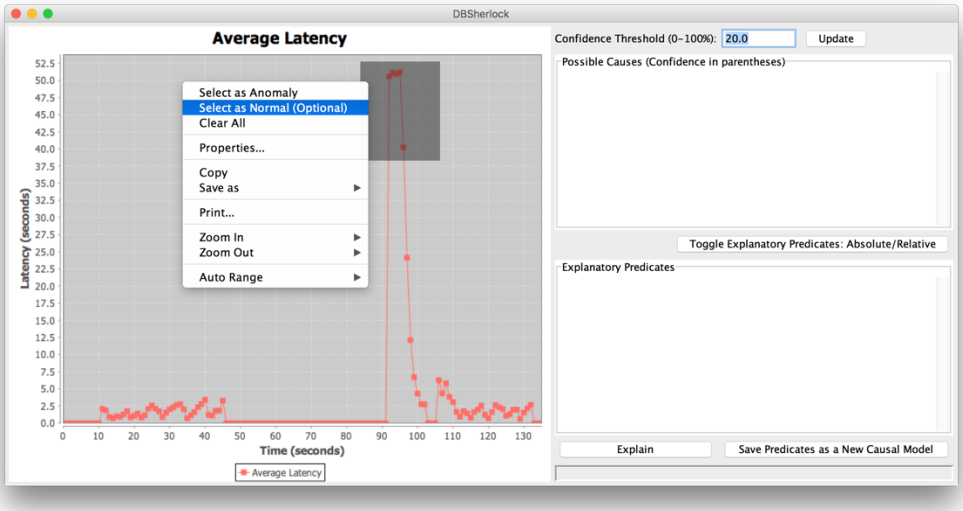
8 Performance Explanation (DBSherlock)

DBSeer includes a performance explanation module, called DBSherlock. It allows users to ask performance anomalies from plot/graphs. DBSherlock will answer the query by providing possible causes and predicates.

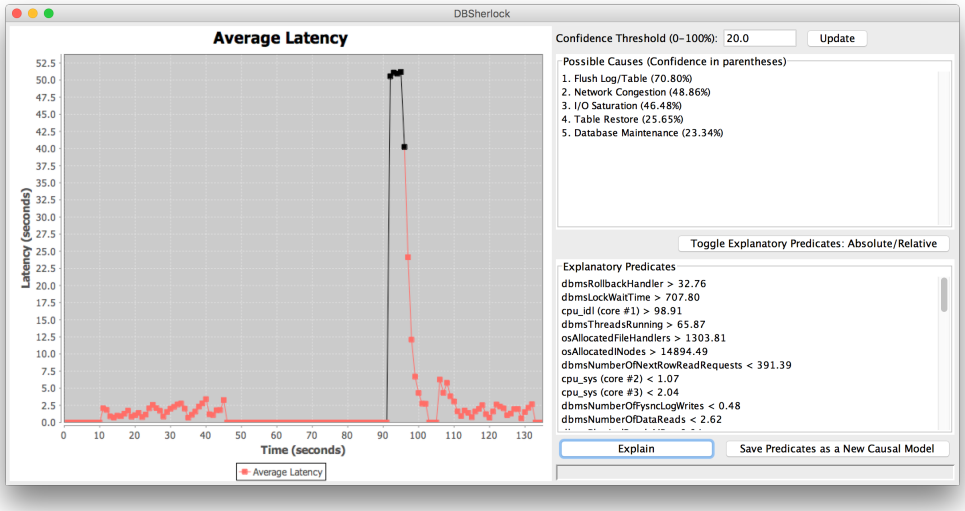
Go to *DBSherlock (Performance Analysis)* tab and generate plot/graphs of the dataset that you think demonstrates a performance anomaly.



In this example, you can see that there is a latency spike in the *Average Latency* graph. You may want to ask DBSherlock about its possible cause. Double-click on the graph to open the DBSherlock window.

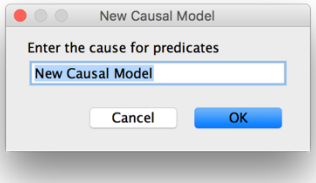


You can select a region that exhibits a weird or suspicious behavior and label them as an anomaly region. Optionally, you can also select other regions of a graph as a normal region. Click on the *Explain* button.



DBSherlock will show possible causes and explanatory predicates for a given anomaly. For possible causes, DBSherlock uses a causality analysis using causal models gathered from previous performance explanation results. DBSherlock contains a set of sample causal models and in the screenshot above, it correctly identifies the correct cause of “Flush Log/Table” with the highest confidence. Even if causal models are not available, DBSherlock will generate explanatory predicates that you can look at and use to diagnose the problem and identify the root cause of the problem. Once you correctly identify the root

cause, you can provide a feedback to DBSherlock by saving predicates as a causal model. Click on the *Save Predicates as a New Causal Model* and name it with the root cause. DBSherlock will save it as a causal model and use the model in future explanations.



9 Contact Us

If you have any questions regarding DBSeer, please feel free to contact us at **mozafari at umich dot edu**.