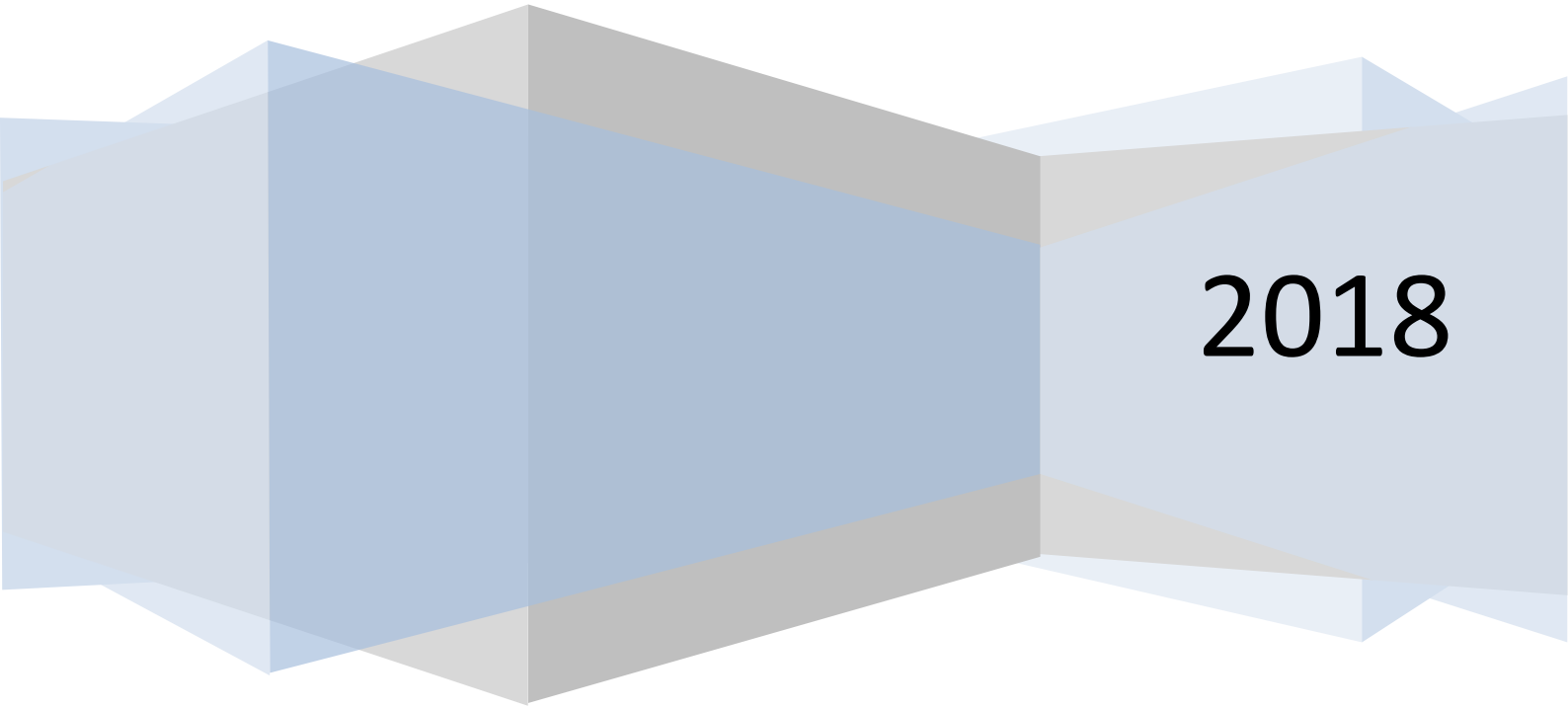
Performance monitor

User guide

Sergey Nebosenko



2018

Annotation

Performance monitor – it is a program for performance measurement of GridEx server.

Performance monitor is a high frequency trading client, which connecting to server and creating Sell- and Buy-orders with random-based prices and volumes. Range for prices is [0.1 … 0.10020001] (Double-type value). Range for volumes is [0.01 … 0.100001] (Double-type value).

After starting Performance monitor run several clients (default=8) in asynchronous mode. Each of them starting send orders of Sell and Buy types in alternately (Sell-Buy-Sell-Buy-…) and receive data from server at another asynchronous process. But client has another one random-based characteristic – Batch size: when count of Sell-orders **+** count of Buy-orders **=** Batch size -> client send request for cancelling all uncompleted orders. Range for batch size is [10 … 20] (Integer-type value). Batch size calculating once and not changes for client lifetime.

Each measured value («metric») represents on its 2D chart plot (value and time). Some of metrics displayed on one chart plot. Each second program updates condition of all metrics in users interface. Program stores values of each metric for 2 minute interval

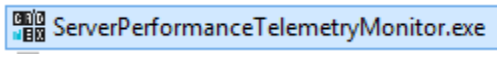
Program uses GridEx API.

Requirements: Windows 7+, the CPU better than Core i7, 16Gb RAM, Internet connection is not less than 100 Mbit, .NET Framework 4.7.2: https://www.microsoft.com/net/download/dotnet-framework-runtime

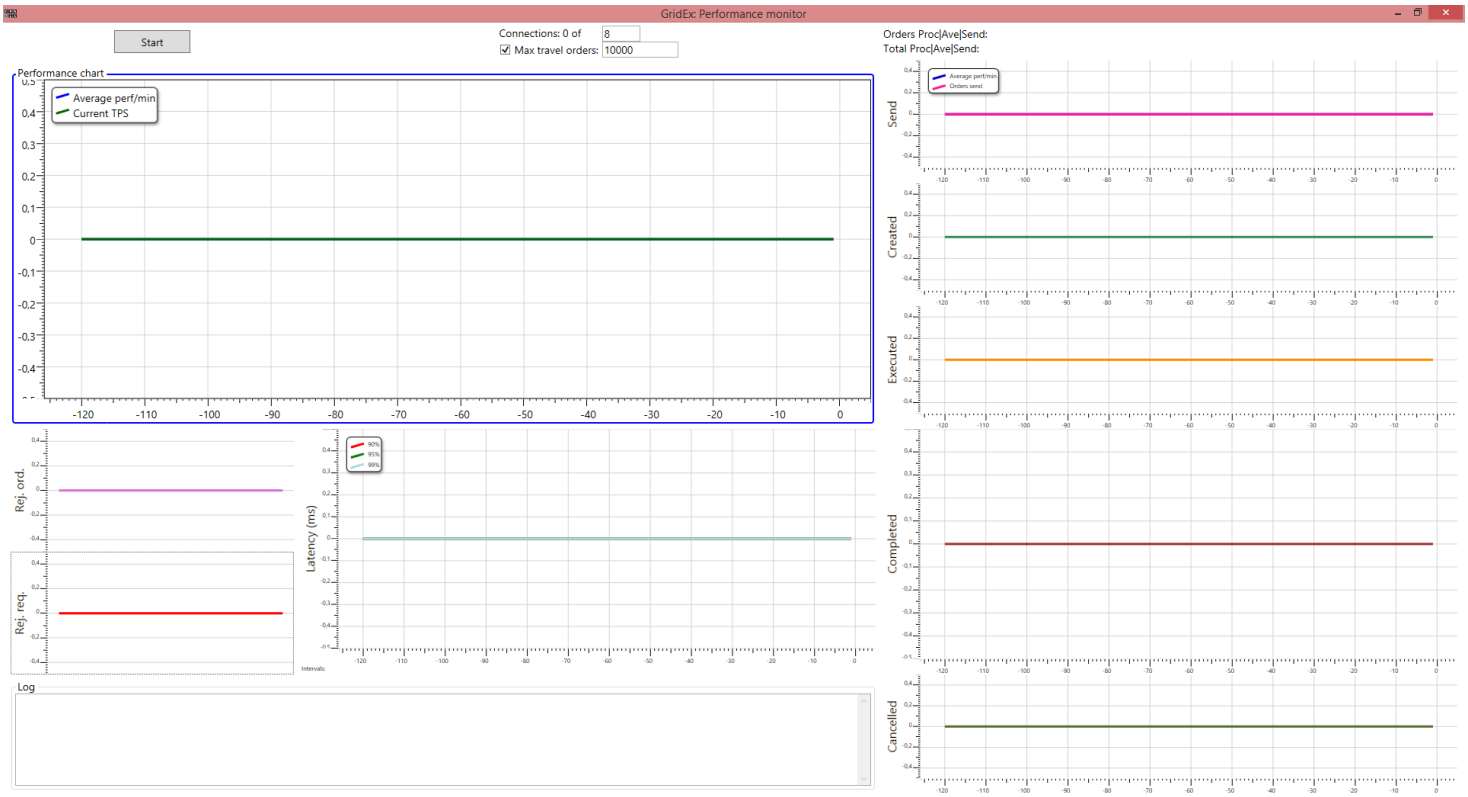
The total latency and performance is **highly dependent** on the quality and width of the internet channel, as well as ping to the HFT Server GridEx.

Running and stopping the GridEx Performance monitor

1) Run the «ServerPerformanceTelemetryMonitor.exe» file

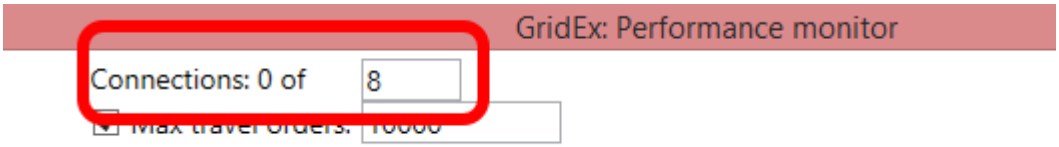


2) Wait while opening main window:

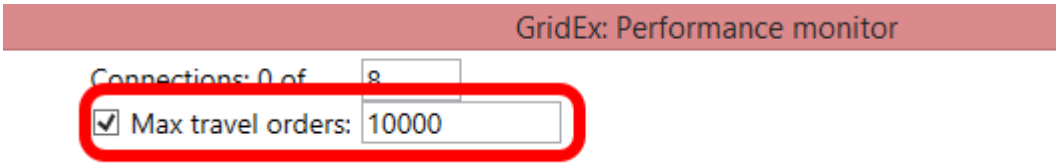


3) Change session options if you need (we don't recommend change them):

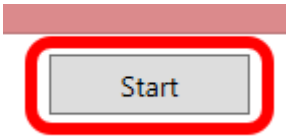
- Number of clients (they connecting to server)



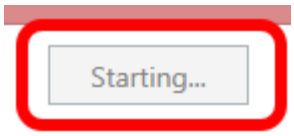
- Travels orders (this orders send from all clients to server, but Performance monitor don't received an answer from server yet)



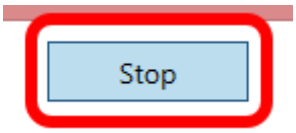
4) Click on the «Start» button to start process of performance measurement:



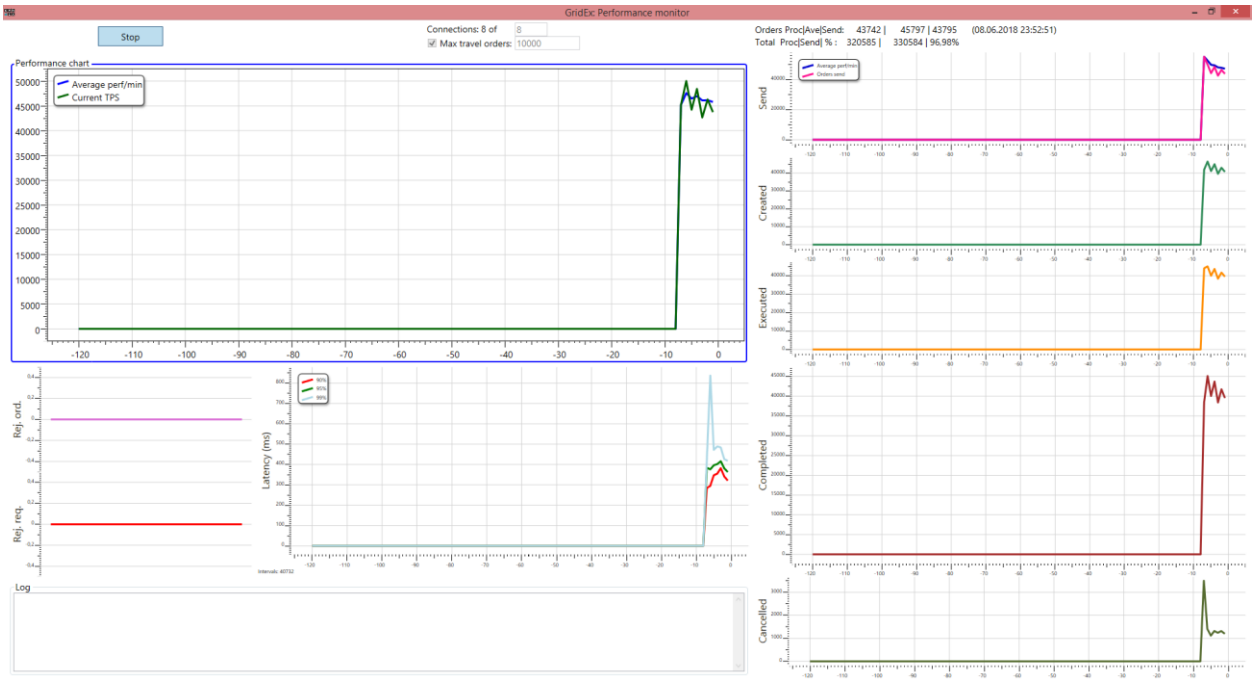
5) Wait while program connecting to server and running backgroung processes. At this time period Start button disabled and display «Strarting…»:



6) When program finished connection Start button is Enabled and display «Stop»:

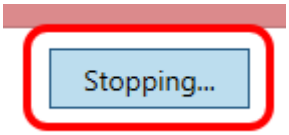


7) Is all is Ok application starting calculates and show the metrics:



8) For stop press click on the Stop button:

9) Stop button changes to «Stopping…»:



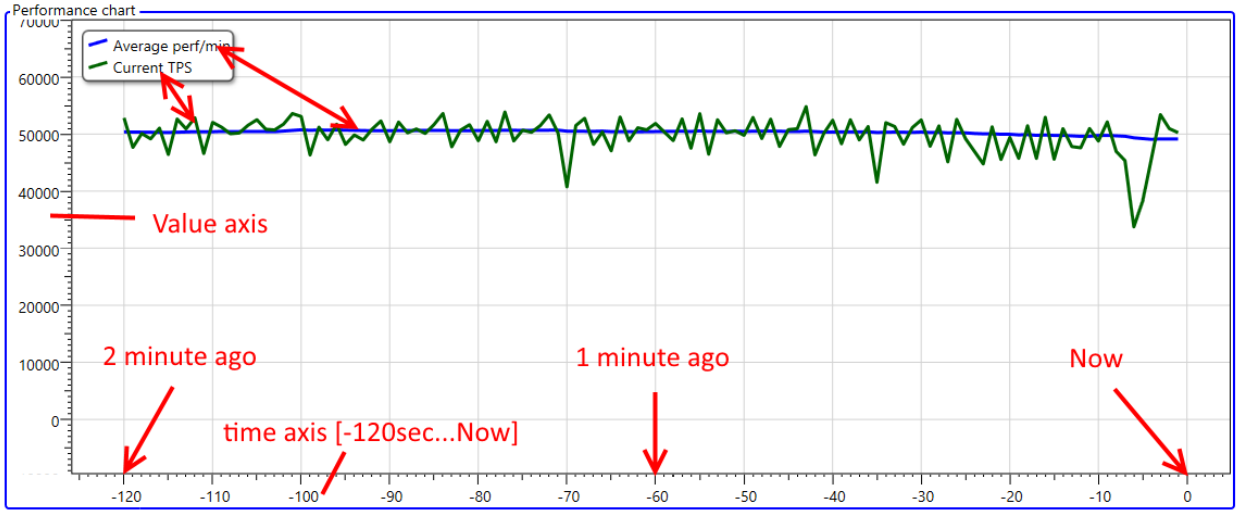
10) When background processes finished «Stopping…» changes to «Start» and metrics will clear.

11) You can exit from application or close is by standard window button  (program will disconnect from server correct).

Metrics

Each metric updates each 1 second and stores 2-minute interval. Metrics are displaying on a chart plot.

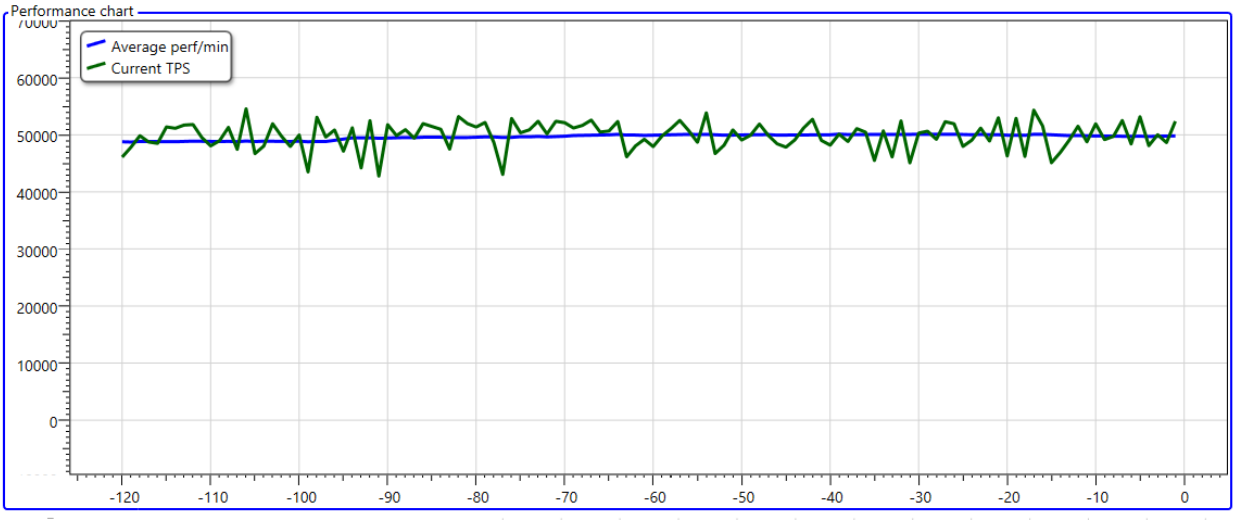
Symbols and objects on the diagram:



Each metric in main window shows as sum of metric of working client at application.

Available metric plots:

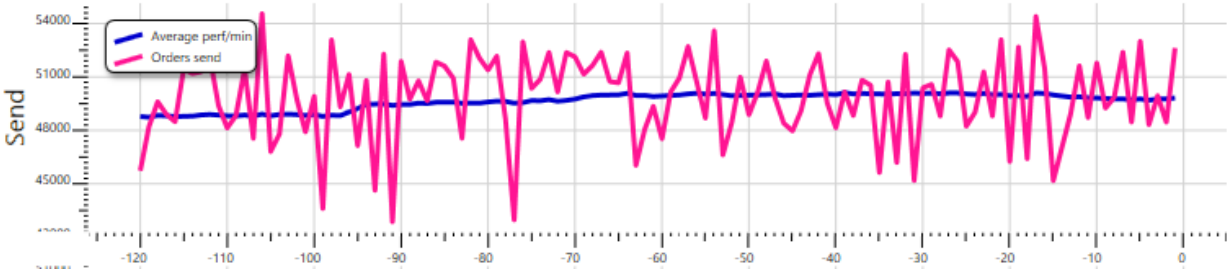
1) «Performance chart». Main chart at the window.



This chat display 2 metric:

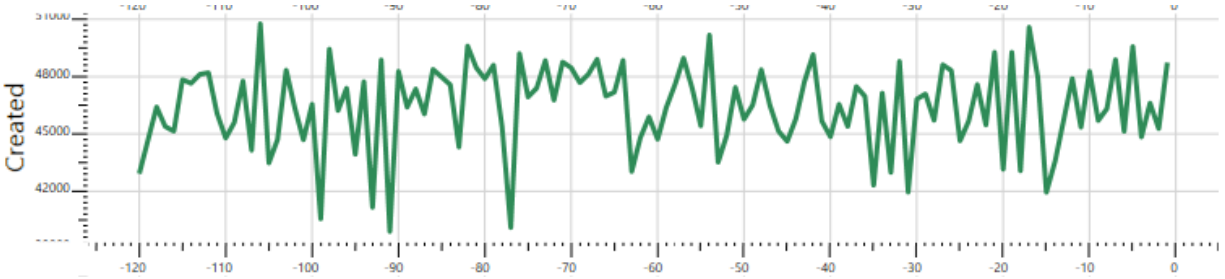
- «Average perf/min» - average performance for last 1 minute: Sum of all performances for last 1minute derived by number of time checkpoints;
- «Current TPS» - current performance – count of completed rejected or cancelled orders for last second.

2) «Send» - sent orders from Performance monitor:

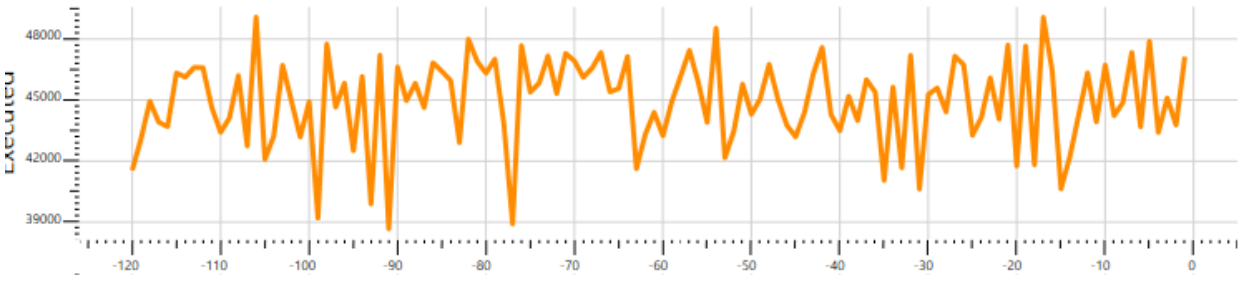


- «Average perf/min» - average performance of sent order to server for last 1 minute: Sum of all sent orderders for last 1minute derived by number of time checkpoints;
- «Orders send» - count of sent orders for last second.

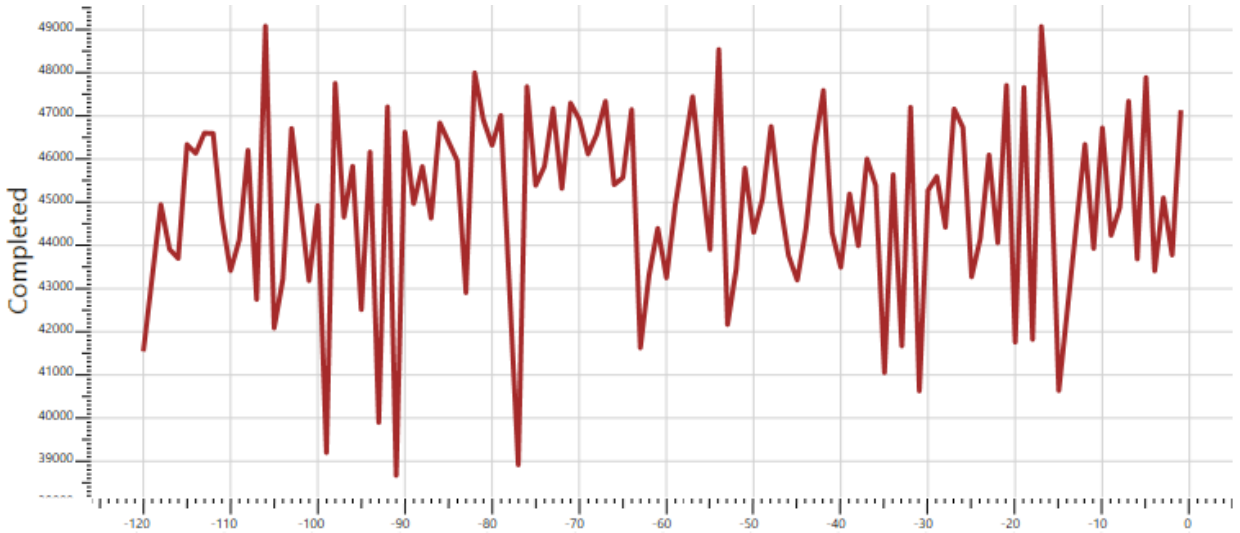
3) «Created» - count of created orders by server for last second:



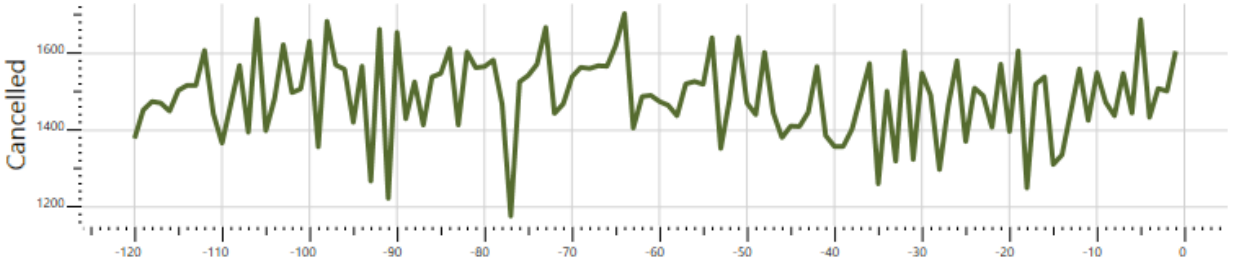
4) «Executed» - count of executed orders by server for last second:



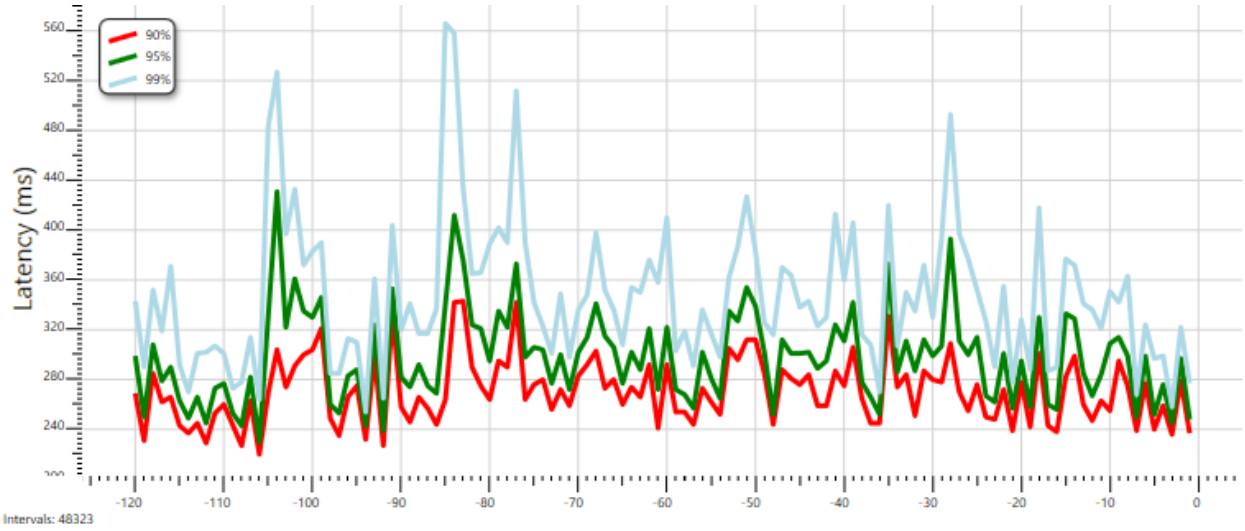
5) «Comleted» - count of comleted orders by server for last second:



6) «Cancelled» - count of cancelled orders by server for last second:



7) «Latency» - this metric display how long server processing each order. Metrics shows times of processing of 90%, 95% and 99% of processed orders in millisecond (important: this times includes ping times to server and from server).

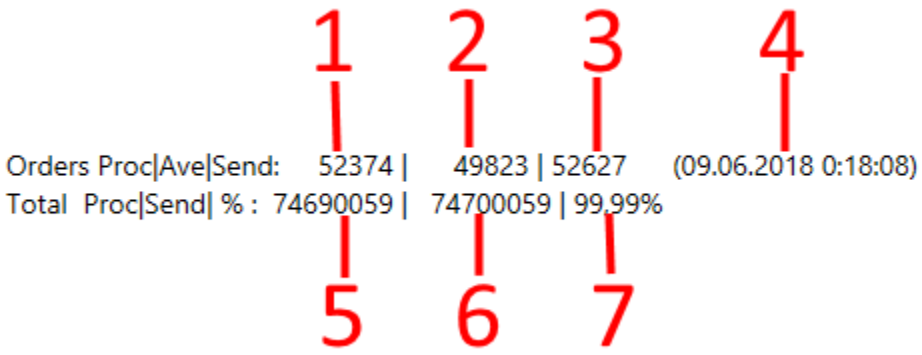


8) «Rej. ord» - count of rejected orders by server for last second:

9) «Rej. req» - count of rejected requests by server for last second:

Overall control of clients

Performance monitor has additional panel for clients control:



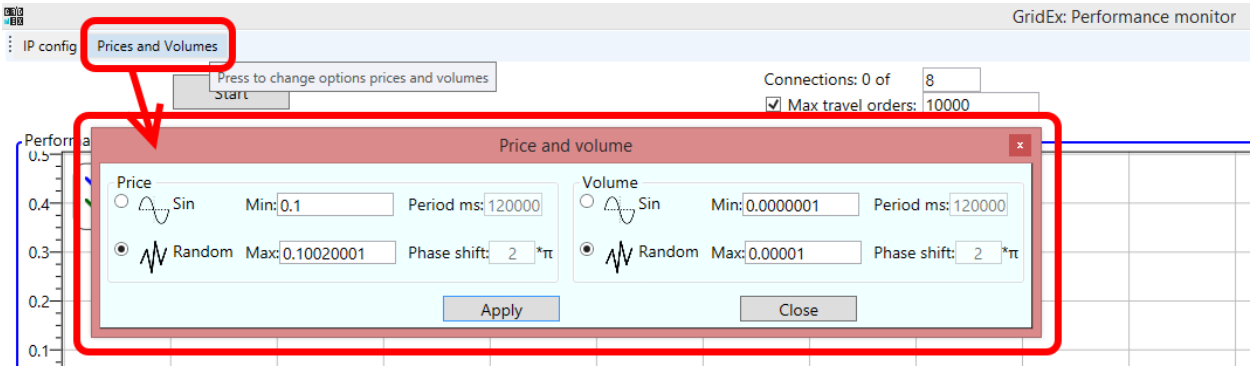
1) Processed orders for last second.
2) Average processed orders for last second.
3) Sent orders to server for last second
4) Last time of update all metrics.
5) Total processed orders for current session.
6) Total sent orders for current session.
7) Percent of processed orders for current session (total sent derived by total processed).

Prices and volumes generation strategy

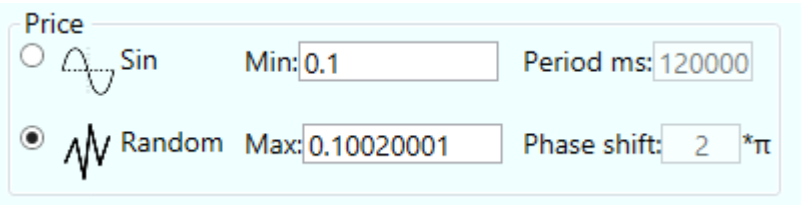
GridEx Performance monitor could generate prices and volumes by 2 ways:

- use pseudorandom distribution to create a small chaos in a set of values;
- use trigonometric function (we select sinus) for smoother formation of values.

You must tap on the “Prices and Volumes” to call settings window:

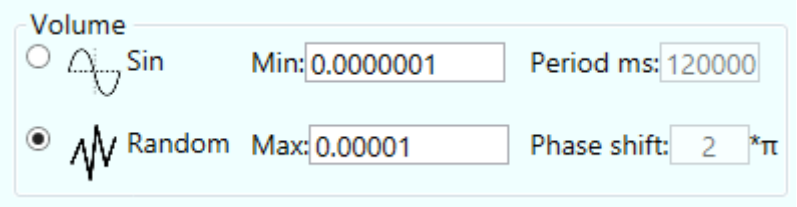


Prices settings of orders changing at “Price” area (example with default values):



- *“Min”* – minimum value for prices (type of Double, possible interval is [`0.00000001` … “Max”]);
- *“Max”* – maximum value for prices (type of Double, possible interval is [“Min” … `0.0002`]);
- *“Period ms”* – period in ms (available only with Sin-strategy) of sinusoid for prices (type of UInt, possible interval is [`1000` … 120000]);
- *“Phase shift”* – phase shift of start price for next client (available only with Sin-strategy) (type of Double, possible interval is [`0` … 2]);
- *“Sin and Random”* – buttons to choosing current algorithm of forming prices of orders: Random will forming values by pseudo-random function, Sinus will form values by Sinus function. You can choose only one of them.

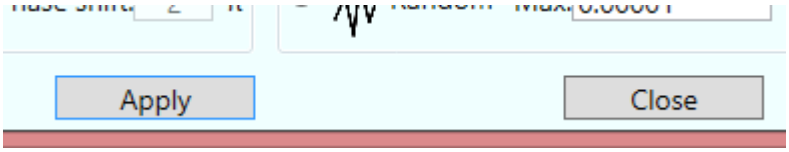
Volumes settings of orders changing at “Volume” area (example with default values):



- *“Min” – minimum value for volumes (type of Double, possible interval is [0.0000001 … “Max”]);*

- *"Max" – maximum value for volumes (type of Double, possible interval is ["Min" … 1]);*
- *"Period ms" – period in ms (available only with Sin-strategy) of sinusoid for volumes (type of UInt, possible interval is [1000 … 120000]);*
- *"Phase shift" – phase shift of start volume for next client (available only with Sin-strategy) (type of Double, possible interval is [0 … 2]);*
- *"Sin and Random" – buttons to choosing current algorithm of forming prices of orders: Random will forming values by pseudo-random function, Sinus will form values by Sinus function. You can choose only one of them.*

Buttons "Apply" and "Close".

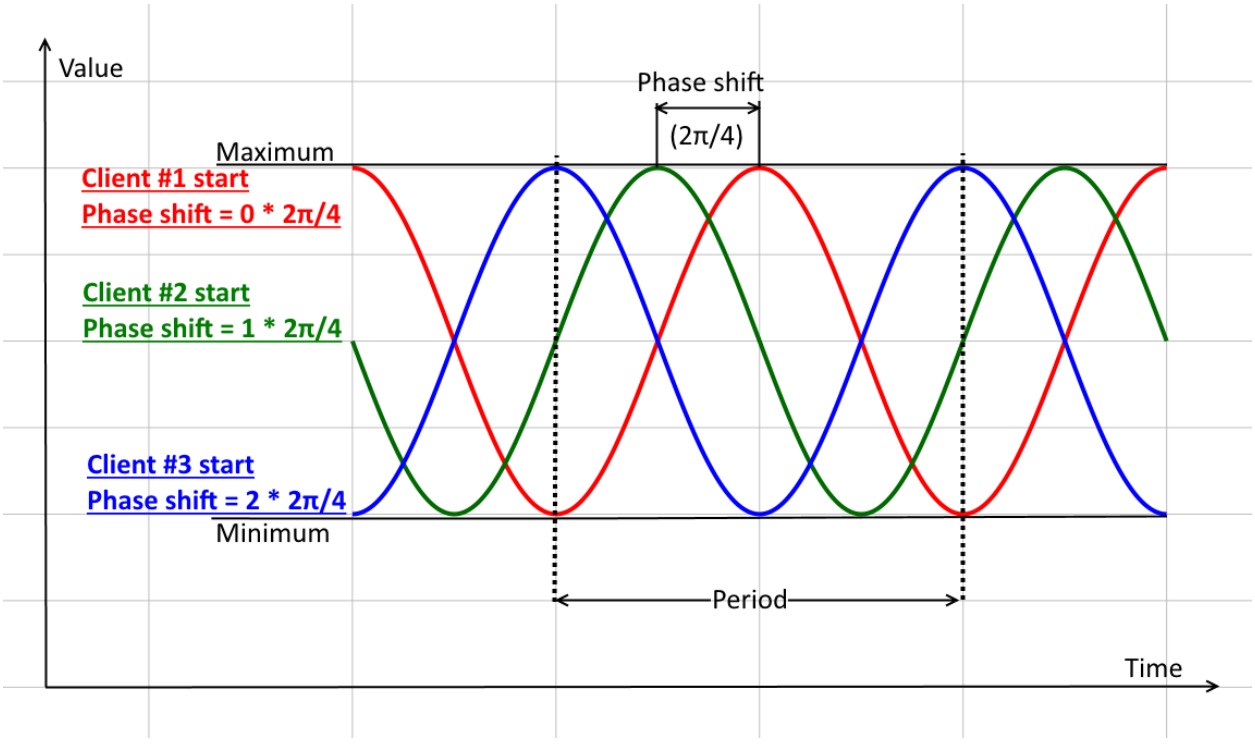


Click on the "Apply" button and program will save you settings and start immediately use new settings to create new Sell- and Buy-orders. "Close" button will discard any of changes. When you click any of this buttons, settings-window would be closed.

Example for Sinus strategy.

If you choose Sinus strategy for prices or volumes, program will be form next value by mathematical function Sinus:

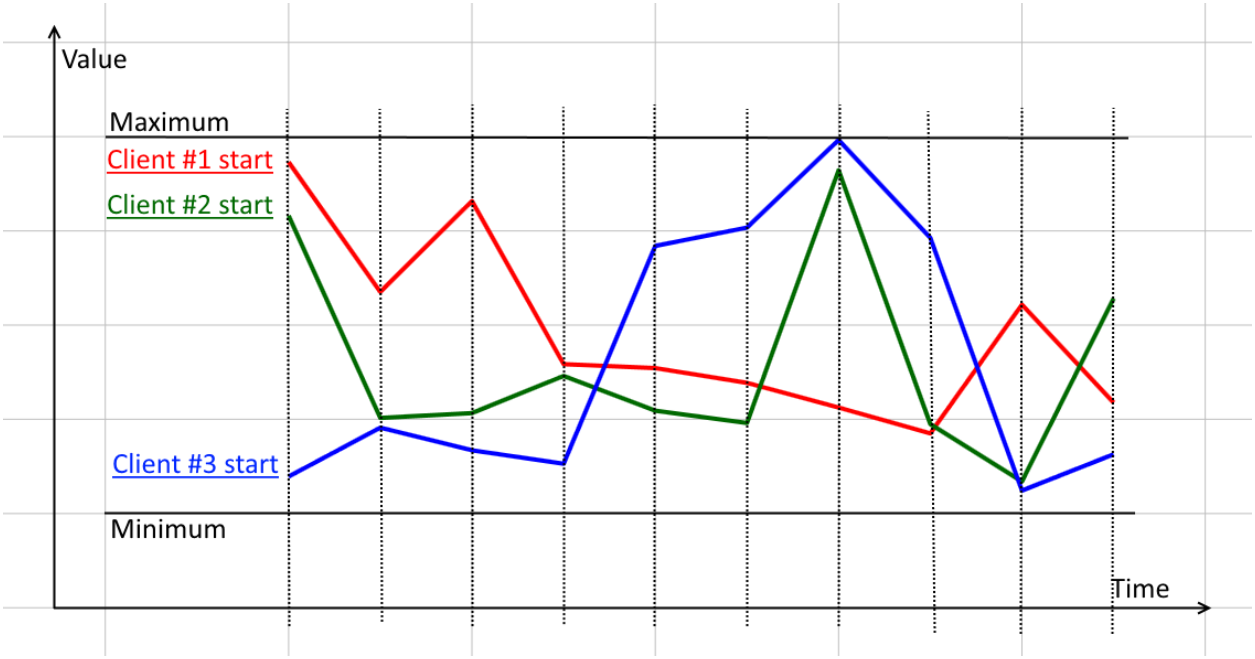
*Value(Time) = (1 + Sin(Time % Period + Phase_shift)) * (Maximum - Minimum) + Minimum*



Example for Random strategy.

If you choose Random strategy for prices or volumes, program will be form next value by pseudo-random function:

Value = Random(Minimum; Maximum)



Parameters of each strategy will be save saved to file “Options.xml” and they will be restored from this file when program you start Performance monitor.