

EventFlow Scalability

Juan Morales

Work in progress: August 3, 2012

1 EventFlow Scalability

The data you can load into EventFlow should be within specific margins so you can use the tool at full capacity. There are a number of known limits regarding memory use, drawing time, screen resolution and human cognitive factors.

We review the 4 main factors affecting scalability and describe symptoms, explanations, and solutions.

1.1 Memory Use

Symptom: The entire application freezes when you load the dataset, or more likely after pressing the “explore” button).

Explanation: The memory use limit depends on the configuration of the Java virtual machine. The Java virtual machine limits the Heap memory it can use. Currently the default threshold is set around 1 GB for a typical computer with 4GB of RAM.

The memory use is correlated with the total number of events in the dataset and rises linearly. Some conditions in the dataset affects the memory use more than others. We have observed that the number of events per record increase the memory use more than the number of records.

With a standard configuration we have been able to load a dataset with 100000 events.

What can you do today:

1. The memory limit can be configured in the source code or by the user with the next command:

```
java -Xmx3000m -jar EventFlow.jar
```

where the maximum of heap memory is configured to 3000MB.

You can also add this command in the .bat file.

2. Load only the dataset you want to analyze. When loading multiple datasets in EventFlow, all of them will use memory even though only one dataset can be explored at a time. You need to restart EventFlow since it is not possible to delete the loaded dataset.
3. If this is not enough, and you still cannot analyze the whole dataset, consider trimming the dataset by removing event types you are less interested in. Or start without including event or records attributes at first.

Solutions we might explore in the future to address this limitation:

- A database backend
- A control panel to allow users to preselect event types of interest before the first exploration
- Freeing the memory used by loaded-but-not-explored datasets
- Pattern aggregation i.e. allowing users to replace small event patterns by new higher level events.

1.2 Drawing Time

Symptom: The tool is unresponsive whenever the overview display is updated

Explanation: This limitation mainly depends on the total number of graphic elements (squares) rendered in the EventFlow overview, which depends on the total number of patterns in the data. We have been able to load a substantial amount of events (around 50000) without a sensible loss of interactivity.

When many records have the same sequence of events, (i.e. they share the same pattern) those records will be grouped, so only one visual pattern will appear in the screen, reducing the drawing time. By contrast if the records have very different patterns, then the aggregation is not as helpful and EventFlow has to draw a lot of patterns. In the worst case there might be a 1:1 relation between number of events and number of graphic elements.

Currently the tool maintains smooth interaction with 20000 graphic elements to be painted. With 100000 elements it is slow but still usable with some patience.

What can you do today?

1. Unselect some event types. This will reduce the number of patterns
2. Specify a time window to reduce the total numbers of patterns. Look in the “Window” tab in the control panel

If this is not enough, then you will have to trim the dataset outside the tool and do the analysis gradually.

Solutions we might explore in the future to address this limitation:

- Allow users to reduce the number of patterns before the first exploration
- Allowing users to replace small event patterns by new higher level events.
- Improve the rendering time when the user interact with the overview (e.g. hovering or selection of patterns). We can play with pre-rendered backgrounds and transparency layers.

1.3 Screen Resolution

Symptom: When you display the data, the scroll bar at the top of the overview is small and can be slided to the right showing more events.

Explanation: The volume of data that the user can analyze efficiently is limited by the number of pixels in the canvas.

Every event in EventFlow is represented as a box, each box has a minimum width (7 pixels for point events and 4 for intervals) and are distributed sequentially along the horizontal axis of the canvas. When there are more pixels needed than are available users will not be able to see the entire dataset without scrolling back and forth.

For a screen resolution of 1600 pixels, users will have to scroll if one sequence includes more than 150 events.

With a very large number of different patterns, the same limitation could be found in the vertical axis. Typically other limits – such as memory consumption – are reached before this becomes a problem.

What can you do today?

1. You can hide the TimeLine panel and the Control panel in the View menu.
2. The width of the point events can be modified (as other visual parameters) in the Control panel, in the Overview > Options tab.

If you really need to see everything at once consider using a screen with more resolution

What we hope to do in the future?

- Fix the horizontal slider: currently users are allowed to widen the length of the slider even if the zoom cannot be wider, so the user expects to see a zoom take place but nothing happens, which is confusing.

- Allow the width of the interval events to be configured. Currently only the point events width can be changed in the options tab of the control panel.

1.4 Human perception

EventFlow assigns one color to each different type of event in the dataset. If the number of event types is over ten – or if they are poorly chosen – they will not be distinguishable.

What can you do today?

1. Limit the number of event types and chose colors carefully.
2. Assign similar colors to similar groups of related types

What we hope to do in the future?

- A better control panel to help users select good default color sets
- Assigning the less distinguishable colors to the categories with few events.