OMNeT++ Community Summit 2018, University of Pisa, Sept 5-7.

Python-based Result Analysis in the OMNeT++ IDE

Attila Török

Motivation

- Fine for quick browsing of the results, but...
- UI and the concept of “Datasets” are unintuitive
- Limited data transformation and charting options
  - ~~e.g. computing a histogram from a vector~~
  - Only a few type of charts
- Difficult to add analytical “ideal values”
  - Example: wireless/throughput INET showcase

Goals

- Extensive statistical analysis
- Publication quality charts
- All integrated into the IDE

Solution

- Incorporate existing tools
  - Python, numpy, Pandas, matplotlib
- Each chart is a Python script
  - Using APIs provided by the IDE
- A custom matplotlib backend is implemented
  - Various interactive charts inside the IDE

DEMO

Improvements

- Easy to add theoretical values
  - The equation can be put directly in the script
- The data can be transformed as needed
- Any statistical package can be used
  - numpy, scipy, pandas
- Many different kinds of charts are possible
  - With extensive customization options

API

- Built-in Python objects:
- Result querying (`results`)
  - getScalars(), getVectors(), …
  - DataFrame transformation utilities
- Using built-in charts (`chart`)
  - plotting and styling
- The entire `matplotlib` API
  - Including extensions like seaborn or ggplot

Technicalities

- Charts scripts run in a separate process
  - Every execution (update) is in a fresh one
  - Python process can be killed, restricted
  - Performance suffers
- Communication with the IDE using Py4J
  - Uses a network socket - to be portable

Known Issues

- Usability and convenience
  - Debugging is basically print()
- Scalability and Performance
  - All data is held in memory
- Security
  - Any package can be imported
  - Unconstrained system access

Further Plans

- Wizards for easy chart creation
- Conversion of old .anf files
- Performance improvements (result querying)
- Sandboxing of the Python process
  - Needs to work on all major platforms
- Option to export as stand-alone script
  - Custom result loading directly from files
  - Emulating built-in charts using matplotlib