Profiling Tools

rostopic bw, rostopic hz, rosnode info

$ rostopic bw /chatter
subscribed to [/chatter]
average: 295.38B/s
  mean: 29.00B min: 29.00B max: 29.00B window: 9
average: 292.35B/s
  mean: 29.00B min: 29.00B max: 29.00B window: 19
average: 291.43B/s
  mean: 29.00B min: 29.00B max: 29.00B window: 29
average: 298.44B/s
  mean: 29.00B min: 29.00B max: 29.00B window: 40

rxgraph, rqt_graph
New Profiling Tools in ROS Indigo

- Topic Statistics  
  - publishes on /statistics
- rosprofiler package  
  - Host Statistics
  - Node Statistics
  - Graph Connections
- rqt_graphprofiler package  
  - Publish/Subscribe visualization tool
rqt_graphprofiler

http://github.com/osrf/rqt_graphprofiler
Features

- Nodes automatically ordered from most producing (inputs) to most consuming (outputs)
- Nodes and Topics can be hidden or rearranged
- Information flows from left to right on top, right to left on bottom
- Nodes' relative CPU usage indicated by distance between inputs and outputs
- Topics with same name grouped together
- Topics ordered by publication frequency
- Topics' line with indicates relative bandwidth
- Topic bandwidth and frequency displayed as numbers
Example System: Remote Teleoperation Interface

- Interface with video, position on map, distance display (laser view)
- Joystick Control of motors
- Map based localization
More Features
(what is not being shown)

- Mousing over a topic line highlights all lines of that topic, as well as their connection blocks inside nodes.
- Mousing over a node name highlights the node, all topic lines directly connected to the node, and the connection blocks they connect to inside other nodes.
- Drag nodes around to re-order them.
- Pan and Zoom (rxgraph actually has this feature).
Nodes Sorted By Machine Location

- Nodes grouped by physical location, ordering still occurs internally
- Data flows from left to right on top as before
- Data flows from right to left on bottom (feedback)
Example Optimization: Heavy Wireless Network Traffic
Example Optimization: Possible Solution
Why this is awesome

- Quickly understand system in terms of inputs, outputs, and processing components
- Quickly understand network utilization
- Better understanding CPU utilization with respect to data flow through the system
- Lets you find places to optimize!
- System state can be logged and observed later
Related Links
wiki.ros.org/rosprofiler
github.com/osrf/rqt_graphprofiler

Questions or Comments
dbrooks@cs.uml.edu