Time-Synchronized Visualization of Arbitrary Data Streams for Real-Time Monitoring and Historical Analysis*

Paul Z. Kolano
NASA Advanced Supercomputing Division
NASA Ames Research Center, M/S 258-6
Moffett Field, CA 94035 U.S.A.
paul.kolano@nasa.gov

Abstract

Large installations involve huge numbers of interacting components that are subject to a multitude of hardware failures, transient errors, software bugs, and misconfiguration. Monitoring the health, utilization, security, and/or configuration of such installations is a challenging task. While various frameworks are available to assist with these tasks at a high level, administrators must more often than not revert to using command line tools on individual systems to get a low-level understanding of system behavior. The output from such tools can be difficult to grasp on even a single system, so when taken across a large number of hosts, can become completely overwhelming.

A variety visualization tools and techniques have been proposed to increase the amount of information that can be processed by humans at once. Existing tools, however, do not provide the flexibility, scalability, or usability needed to assist with all the varied information streams possible in large installations. In particular, these tools often require data in a specific format and/or in a specific location with interfaces that have little relation to the underlying commands from which the data originates.

Savors is a new visualization framework for the Synchronization And Visualization Of aRbitrary Streams. The goal of Savors is to supercharge the command-line tools already used by administrators with powerful visualizations that help them understand the output much more rapidly and with far greater scalability across systems. Savors not only supports the output of existing commands, but does so in a manner consistent with those commands by combining the line-editing capabilities of vi, the rapid window manipulation of GNU screen, the power and compactness of perl expressions, and the elegance of Unix pipelines. Savors was designed to support *impromptu visualization*, where the user can simply feed in the commands they were already using to create alternate views with optional on-the-fly aggregation of information across many systems. In this way, visualization becomes part of the administrator's standard repertoire of monitoring and analysis techniques with no need for a priori aggregation of data at a centralized resource or conversion of the data into a predefined format.

Savors can show any number of data streams either consolidated in the same view or spread out across multiple views. In multi-data scenarios, data streams can be synchronized by time allowing even distributed data streams to be viewed in the same temporal context. In single-data multi-view scenarios, views are updated in lockstep fashion so they show the same data at the same time. Together with its integrated parallelization capabilities, this allows Savors to easily show meaningful results from across even very large installations.

Savors consists of three components: a console, some number of data servers, and some number of views. The console is responsible for user interaction, spawning data servers and views according to the given command pipelines, and controlling synchronization between data streams. The data servers are responsible for spawning and interacting with the commands that generate data, manipulating the data as specified, and sending the data to the console and views. Finally, the views are responsible for visualizing the data as specified on one or more displays.

Savors is open source and available for download at http://savors.sf.net.

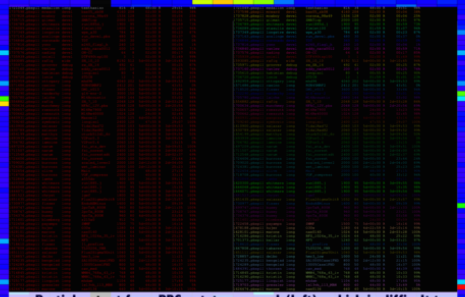
*This work is supported by the NASA Advanced Supercomputing Division under Task Number ARC-013 (Contract NNA07CA29C) with Computer Sciences Corporation

Time-Synchronized Visualization of Arbitrary Data Streams

Paul Z. Kolano
paul.kolano@nasa.gov
http://savors.sf.net

1. Take Control Of Data Overload

- Large installations have many interacting components subject to many types of failures
  - Often must inspect/analyze by command line tools
  - Too much output to easily understand at once
- Savors is a new visualization framework
  - Designed for "impromptu visualization" of output generated from command line tools already used
  - Powerful parallelization and aggregation of data streams with time synchronization
  - Flexible window layouts with rapid manipulation
  - Consistent with standard tools: vi editing, screen-like controls, perl expressions, and unix pipelines
  - Quickly gain insights about system status/issues



Partial output from PBS qstat command (left), which is difficult to understand in full, along with a limited increase in understanding after colorization by user in the Savors text rainfall view (right)

3. Edit/Manage/Inspect Views Via Interactive Console

- The Savors console is responsible for
  - Providing user interaction
    - Vi-like command line editing
    - Screen-like manipulation of layouts/regions/windows
    - Pause and step of data streams
    - Data inspection and view color mappings
    - Save and restore of view commands
    - Snapshots of any view window at full resolution
    - Context-sensitive help system
    - Control of multi-host displays from single point
  - Spawning data servers and views as specified
  - Controlling synchronization between data streams

Savors console with top left showing 8 views arranged in 3-2-3 layout and details of current paused data line, top right showing graph view command and help, and bottom showing color legend

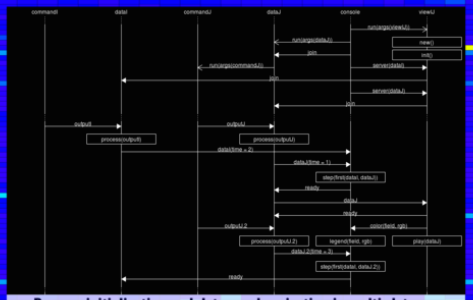
5. Aggregate Multiple Streams Into A Single View

- Savors data servers are responsible for
  - Spawning and interacting with commands that generate data
  - Manipulating data as specified in directives
  - Sending data to console and "subscribed" views
- The "data" directive allows multiple data servers to be spawned and fed into one view
- A data server is spawned for each "data" value
  - Before spawning, the special variable "fD" is replaced by each value in pipelines/view options
  - e.g. env data=1-6 ltop /nobackupfD ... would create a data stream of stats from 6 Lustre file systems and allow them to be unified in one view

Word cloud of users and the commands they are running via ps across 16 cluster front-end systems colored by command and sized by cpu time

7. Synchronize Streams By Time

- All data streams are associated with a synchronization group
  - Default or specified via "sync" directive
- Each data line is associated with a time
  - Output with time information
    - Parsed out using "time" and "time_grep" directives
  - Output without time information
    - Each line assigned time it was received
    - Lines in same iteration of "repeat" directive assigned the same time
- Data servers send current line to console
- Only earliest data stream in each sync group allowed to send data to "subscribed" views



Process initialization and data synchronization in multi-data single-view model where new view joins existing and new data servers before data stream with earliest time allowed to proceed

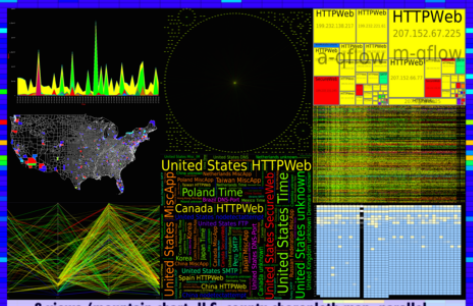
2. Visualize Any Command Pipeline

- Savors can visualize any pipeline by adding data directives and a view to the original args
  - env opt=val... (args... | ...) | view ...
- Data directives specify how to process output
  - Synchronization: time, time_grep, sync
  - Line manipulation: cut, grep, sed, split
  - Common colorization: color, ctype
  - Parallelization: data, layout, view
  - Other: host, label, label_grep, repeat, replay, tee
- Views specify how to display data
  - Type: axis,chart,cloud,graph,grid,map,rain,tree
  - Fields to show: e.g. --fields=f1,f4-f7,f9
  - Other: colors, period, view-specific options, etc.

Treemap of PBS jobs from earlier output grouped by queue, user, and job id, colored by user, and sized by # of cpus (env repeat=15 grep_v='^\D' qstat -r | tree --color=f2 --fields=f3,f2,f1,f5)

4. Create Arbitrary Layouts Across One Or More Displays

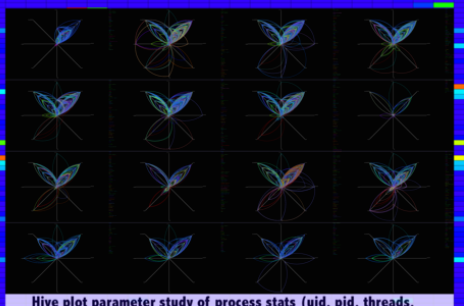
- Savors views are responsible for visualizing data as specified on one or more displays
- Savors currently supports axis plots, charts, word clouds, graphs, grid plots, maps, text/binary rainfalls, and treemaps
- Views are arranged in layouts like GNU screen
  - A layout consists of one or more regions
  - A region consists of one or more windows
  - Each window corresponds to a view
  - Multiple layouts/windows may exist with shuffling
- Layouts may be arbitrarily mapped to a grid of one or more physical displays
  - Displays may be attached to different hosts



8 views (mountain chart, U.S. county choropleth map, parallel coordinate plot, graph, word cloud, treemap, binary rainfall, and heatmap) of a single data stream from a QRadar security appliance

6. Generate Parameter Studies Using A Single Command

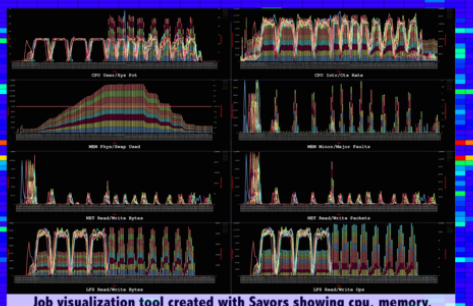
- The "view" directive specifies the parameters to use in each view
- The "layout" directive specifies the arrangement of windows in the current region
  - Any combination of grids (wxh), vertical splits (l|r), and horizontal splits (t-b)
  - e.g. 1x3 | 1x2 | 1x3 is earlier 3-2-3 console layout
- A new view is spawned for each "view" value
  - Before spawning, the special variable "fV" is replaced by each value in pipelines/view options
  - e.g. env layout=4x4 view=1-16 ssh hostfV ... would create view for each of 16 hosts in 4x4 grid
- "view"/"data" directives can be used together



Hive plot parameter study of process stats (uid, pid, threads, resident size, swap size, virtual size, cpu time, and elapsed time) via ps across 16 cluster front-end systems colored by command

8. Encapsulate Knowledge Into New Tools w/ Simpler Interfaces

- Any new/saved view can be executed without the console interface via the --command option
  - Easily create visual versions of standard/useful commands that can be deployed for all users
  - Normal users do not need knowledge of Savors
- Visualization becomes just another standard tool in workflow of system monitoring/analysis
  - No need to centralize or reformat data streams ahead of time
  - No need to install extra software beyond one host
  - No long-running components so no cycles wasted
  - Do what you were already doing but see it visually
- Download at http://savors.sf.net



Job visualization tool created with Savors showing cpu, memory, network, and lustre stats over time for nodes of given PBS job id colored by node (shown: integrity-verified parallel file transfer)