Tool Design: gloss

Grep for Logs on Open Source Systems. Allows you to gloss over the data more efficiently than the plain textual form.

Idea is to grab information and make sense of it. Uses multiple log files,

- log files can be mentioned explicitly after the options, to read instead of default or all; double dash notation may be used to explicitly mark the end of options when the separation would otherwise run into ambiguity
- **-l** for log facility; used to find references in syslog.conf
- **-b** and **-a** for before and after; when out of order, drop a range instead of requiring it; may be an integer timestamp, a date/time, or a time (ranging back 24h)
- **-p** to set a PID; may be a path to a file holding a PID; may be tcp:port or udp:port or sctp:port; may be a program name to match
- **-h** for the logging host name, as represented in the log files; defaults to match all hosts; multiple hosts can be presented as explicit alternatives
- **-d** for a driver; may be used to recognise a program's specific log file formatting; a directory holds a file with these drivers, and programs can install their data in here as a modular extension; these modules define similar parameters to the above to select whether they might apply, and then still they may fail; drivers may share variables and/or specify aliases in other drivers; there is a special driver named "pass" that will match any free form, which by default would not have passed; drivers are applied in the order of occurrence in these options
- **-m** to run in another mode than the usual client; specify an http URI for an HTTP server; the address may be a localhost port, an address:port, a UNIX domain socket, or ssh: for an SSH style service
- **-c** for coding the output in a particular manner; html is an option, and so are csv and count; default is text for plain text display of selected lines from the log files
- **-s** selects one or more variables to display as they occur in the various lines of text; by default, all variables are shown; when multiple variables are used, they are separated by equals signs
- **-w** for where-clause selection, requiring a pattern for a line that binds the given variable to the following value (after an = sign for a match or != for a non-match); multiple criteria may be entered to further constrain the selection
- **-o** for an or-separation between where-clauses, which are normally conjugated; not that the combination and the lowest-level negation through != or = allows the expression of any logical combination
- **-r** require free-form regexp in the line's freeform text
- **-v** increased verbosity: suggest files that match the criteria; report when

drivers miss lines that they would have liked to match; show log entries with variables explicitly marked inline; warns about regexps without obliged fixed text

Useful examples of Driver Plugins

- **Postfix** dumps a lot of structured information into mail.log, with identities that can be searched for. Use it to create cross-referenced fields and easily step between aspects of a problem.
- **Apache** dumps a lot of information about clients, URLs, reply codes and so on. It may be useful to be able to group on either of these.
- **OpenDNSSEC** creates a lot of output while processing a domain name. It is very useful to be able to look at just one domain, and see it progress.