

# *SystemTap Runtime*

Martin Hunt

Red Hat

20 Apr 2005



## *What is It?*

- C code used by translator
  - Compiled into kernel module
  - Could be used to write providers
- 
-

# *Runtime Goals*

1.Safety

2.Low impact on system. Low kernel stack usage

3.High Performance

4.Flexible

5.Ease of Use



# *What it Does*

- Associative arrays
  - Kernel to user transport
  - Output formatting
  - Copy from userspace funcs
  - Backtrace and register dump
  - Safe Strings
  - Other things as needed by translator
- 
-

# *Associative Arrays (Maps)*

- Keys can be 1 or 2 strings or longs
- Values can be int64, string, or statistics
- By default, Maps have a maximum number of members, which are preallocated.
- Different memory allocation schemes can be plugged in.
- ~~Lists are implemented as Maps with an~~

# *Kernel to User Transport*

- RELAYFS
  - Chunks of data
  - Per-CPU files
  - Very fast
  - NETLINK
  - Stream of data
  - Slower?
- 
-

# *Output Formatting*

- Timestamp and CPU info
  - Print buffer is per-cpu static buffer used so accumulated data can be printed with newlines but still only one timestamp. For example -  
backtraces.
  - `stp_printf()` sends to print buffer
  - `stp_print_flush()` timestamps and sends to `stpd` via `relayfs` or `netlink`
- 
-

# Backtraces

- Currently we can backtrace (imprecisely) in kernel and userspace.
  - Symbolic lookups (with kallsyms) in kernel space only.
  - Symbolic lookups are slow and should probably be deferred until postprocessing.
  - Will need rethought when we have DWARF lib.
  - <stack 0xffffffff 0xffffffff ...>
- 
-

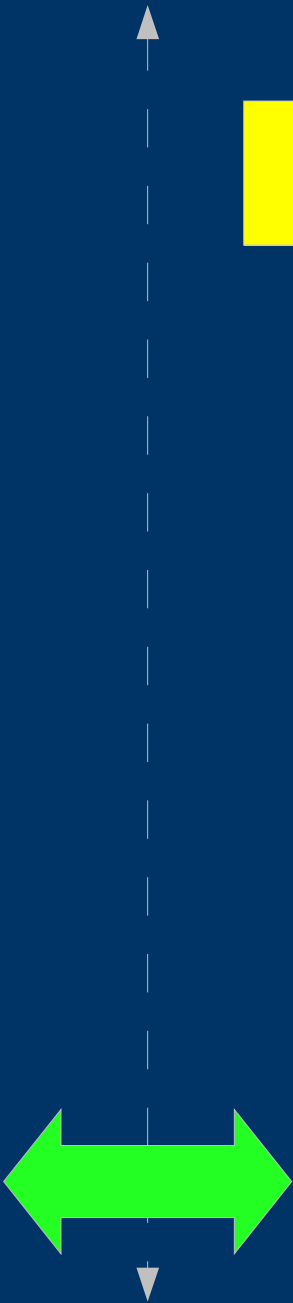
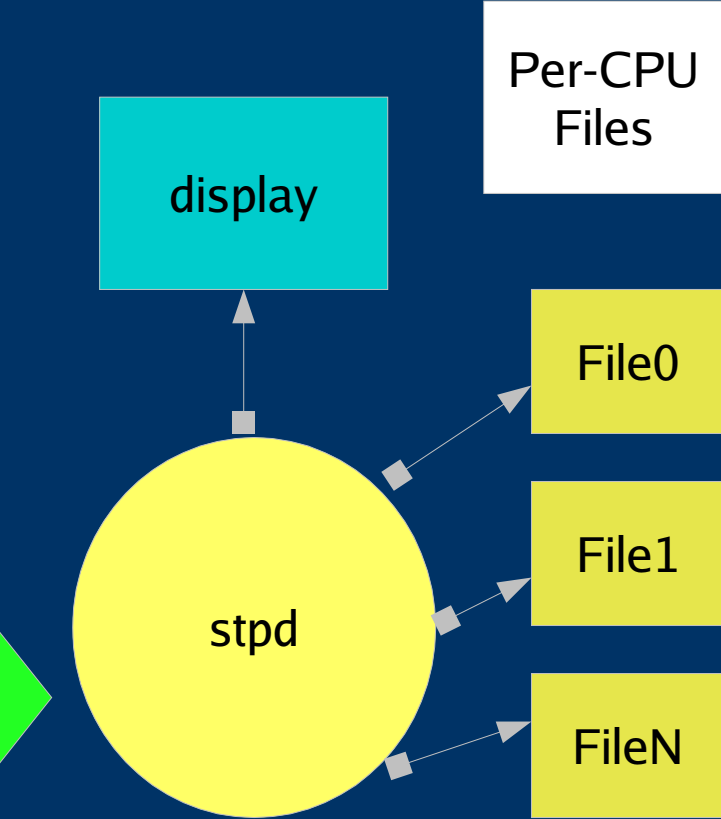
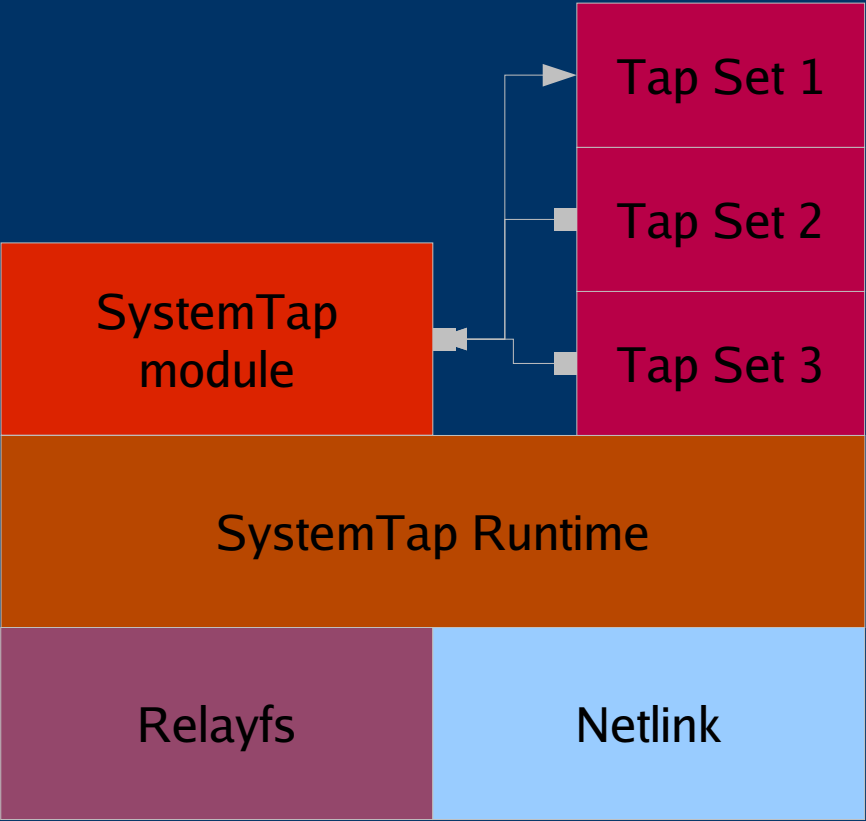


# *Safe Strings*

- Strings should not be allocated dynamically.
  - We cannot use the stack either.
  - Solution: a pool of statically allocated per-cpu buffers.
  - Implementation does not allow buffer overflows.
- 
-

Kernel Space

User Space



# *STPD*

- There is one stpd per loadable systemtap module.
  - Multiple stpd's can run simultaneously.
  - Stpd loads and unloads module.
  - It saves data to per-cpu data files.
  - It displays log messages
  - Responds to user requests (^C)
- 
-

# *Postprocessing*

- Done by stpd?
  - Deferred symbolic lookups. (stack traces, etc)
  - Per-cpu data files need integrated and saved.
  - Data displayed as systemtap script output.
- 
-

# *Current Status*

- Code base is rapidly changing. Should be more complete in the next week or two.
  - Latest working version, with working example probes and detailed documentation checked into CVS.
- 
-