

# SystemTap

William Cohen  
Performance Tools Engineer  
Red Hat Software, Inc.

# Abstract

- SystemTap, a dynamic instrumentation tool, is being developed by Red Hat, IBM, and Intel.

# SystemTap Purpose

- To provide insight into system operation
- To make it easier to identify root cause of performance problems
- Tool set to build instrumentation

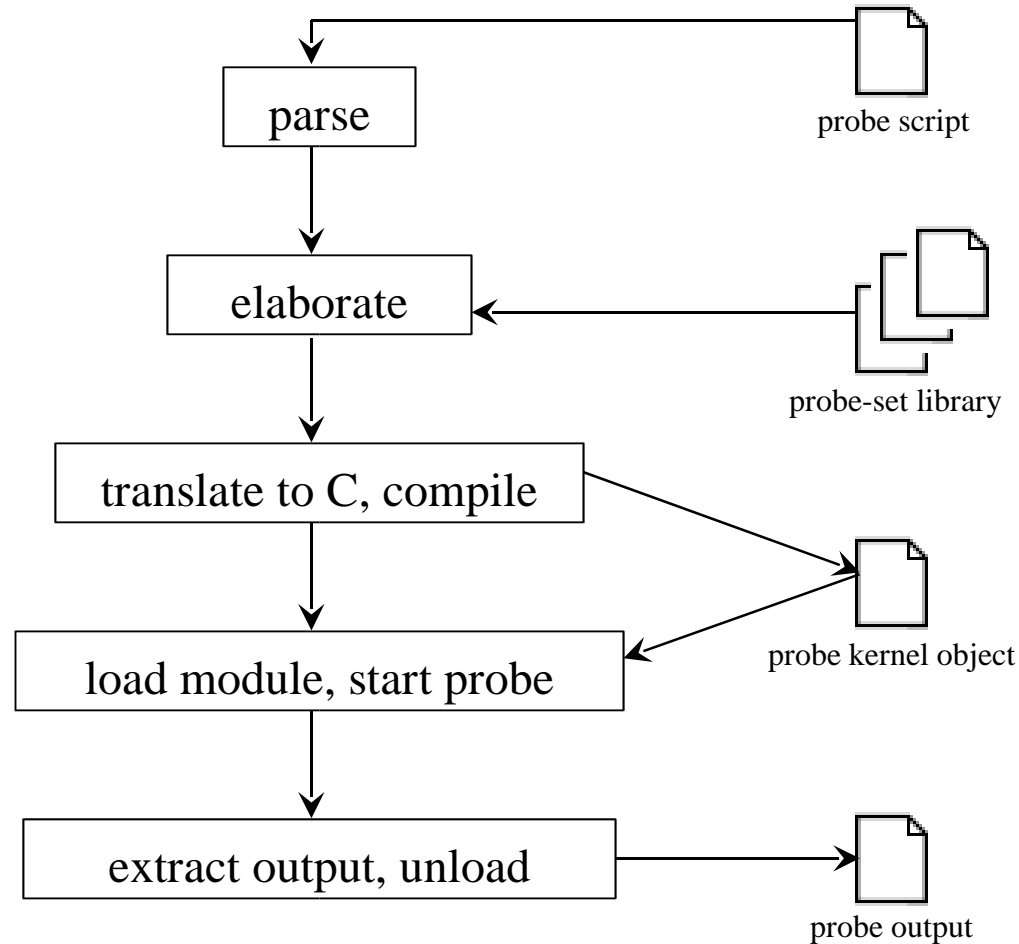
# Examples Data Collection

- Which processes generating network traffic
- Which parts of kernel allocating memory

# Goals

- Ease of use
- Extensibility
- Performance
- Transparency
- Simplicity
- Flexibility
- Safety

# Process Generating Instrumentation



# Key Technologies

- Kprobes
- Runtime libraries
- System tap instrumentation script compiler
- Turnkey instrumentation scripts
- libdw
- debuginfo files

# Kprobes

- In 2.6 kernel, enabled in fc4 kernels
- Backport available for RHEL 3 kernels
- Implemented with software interrupts
- Trap routine searches for kprobe for location
  - Do associated prehandler
  - Single-step through instruction
  - Do associated post handler
  - Resume execution



# Kprobes Enhancements

- Jprobes, access arguments to function
- Multiple kprobes at single address
- Return probes
- Improved concurrency, avoid serializing kprobe handling

# Runtime Libraries

- Used by handwritten and translated instrumentation
- Provide:
  - Access to state information:
    - PID
    - Return address
  - associative arrays
  - mechanism to transfer data from kernel to user-space

# Instrumentation Translator

- Provide safety
- awk-like language
- Translate instrumentation scripts into C code and library calls

# Turnkey Instrumentation

- Provide instrumentation to handle common cases
  - Scheduler operations
  - Systemcalls being invoked across the system
  - VM – alloc /deallocs
  - I/O – VM interactions in device driver

# Libdw

- Needed factor out code to for debug information
- map from user source code to addresses
- map data address back to source code

# Debuginfo Files

- By default built when RPMs created in Beehive
- Provide debugging information for binaries
- Need to RHN to provide these files externally

# Future Work

- Get volunteers contributing instrumentation
- Lots of testing to verify everything works
- Integration of components to provide a “solution”