What is a tapset?

Vara Prasad
prasadav@us.ibm.com

Topics

- Purpose of a tapset
- Tapset authors and their needs
- Tapset users and their needs
- Proposals

Tapset description

- Tapset is a set of functions in a given kernel subsystem to export data about that subsystem.
- Two types of tapsets
  - Synchronous to the execution
  - Asynchronous to the execution
- Asynchronous tapset functions can only refer to global data as there is no guarantee of the context
- Each synchronous tapset function is associated with a probe point PC location.
- Tapset function API's are published for system tap script writers.

Tapset Authors and their needs

- Authors are usually experts in a given subsystem
- They would prefer to write in familiar "C" language, not learn a new one.
- Tapset functions are called as handlers to the associated kernel functions.
- Authors would like same environment as that of probed function.
  - Access to function arguments and local variables
- Authors want flexibility to implement tapset functions as long as the API needs are met.

Tapset users and their needs

- Script writers are the end users of tapset functions via systemtap infrastructure.
- End users may not be kernel experts.
- It is not mandatory to use the associated tapset function if the probe point is activated.
- Output of the tapset functions is well documented to use in the scripts.
- Prefers a easy high level language to access tapset functions.

API Proposal

- There is an API for each tapset function from systemtap runtime
- API has a void * for exporting data
- Tapset writer has to provide unpack functions to extract exported data
- Advantages
  - Tapset code can be part of the kernel hence easy to maintain
  - Tapset writers have the freedom to implement their code in their familiar environment

API Proposal Continued

- Advantages
  - Locking issues are hidden from the rest of the systemtap infrastructure.
- Disadvantages
  - Script writers are limited by the data exported by the tapsets

DPCC Proposal

- As part of the script user specifies what data they would like to see
- Based on the users data needs code will be generated to access kernel data structures.
- Advantages:
  - Advanced users can extend the tapset library
- Disadvantages
  - Tapsets are written in their own language hence not likely in the kernel, maintenance issue
  - Locking could be problematic

Frank’s hybrid proposal

- In this proposal there are extensions in various layers that facilitate users to extend the existing tapset library
- Advantages
  - Extensible
- Disadvantages
  - Language can get more complicated from end users perspective
  - Not clear if this poses a code maintenance problem for not having kernel code in the kernel tree

What do we need to decide?

- Should we make it a requirement that tapset should be written using or not using systemtap language and library.
- What features that we need to allow in GURU mode that are not allowed in the default mode
- Can we implement most of the GURU mode features as library or language constructs
- What features are basic that we should first implement to get some feedback from users.
- How necessary is to provide extensibility in systemtap to add new tapsets or tapsets are outside systemtap language.