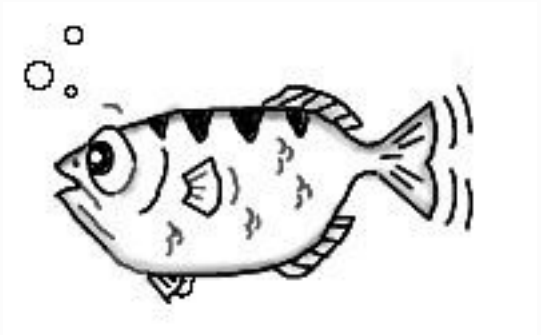
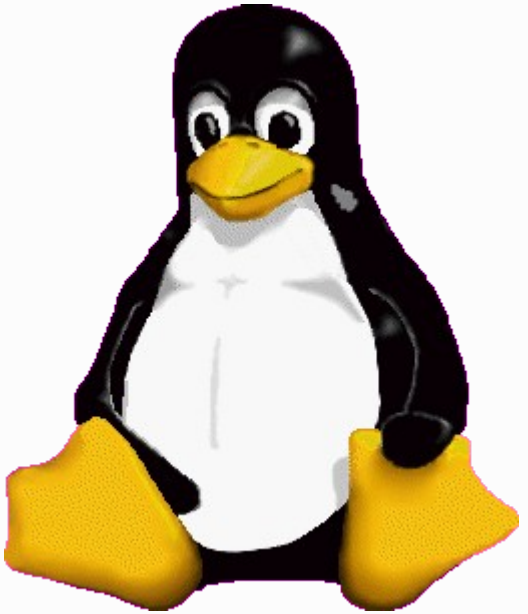
KGTP, GDB 和 Linux





https://code.google.com/p/kgtp/

朱辉 (teawater@gmail.com)

什么是 KGTP ？

- KGTP 是一个 灵活 轻量级 实时 Linux 调试器 和 跟踪器 。
- 主要针对线上不方便停止的软件中的问题或者不易在线下环境复现的问题。
- 能处理一些嵌入式系统中的问题。
- 被一些公司使用，最主要的是在今年 1 月合入了 taobao 的内核 tree 。

轻量级 代码轻量级

- 主要是开发者利用业余时间维护，所以是一个轻量级的项目。 :)

- 因为分析数据主要使用 GDB 所以不需要很多数据分析代码。

  另： http://code.google.com/p/gdbt/ 提供 GDB for KGTP 支持。 x86_64 和 i386 的 static GDB 下载，还提供 Ubuntu PPA，还有提供自己编译的介绍。 Opensuse 的源也在计划中，支持后 fedora debian 等软件都可支持直接安装。

- GTP 20120920:

  gtp.c 227023 字节 10654 行

  gtp.h 2680 字节 119 行

  gtp_rb.c 11464 字节 510 行

轻量级 实现轻量级

- Tracepoint 使用 Kprobe 实现， trace 点动态插入。不使用的时候不浪费系统资源。
- Kprobes-optimization 还可提高 kprobe 的速度。

实时

- 一般 GDB 在调试程序的时候需要对这个程序的执行进行中断，但是和 KGTP 一起工作的时候不会。
- KGTP 提供了一个远程 GDB 调试接口。
- 在本地或者远程的主机上的 GDB 可以在不需要停止内核的情况下用 GDB tracepoint 和其他一些功能调试和跟踪 Linux 内核和应用程序。
- 我经常在自己的机器上一边听评书一边使用 KGTP 。

演示助手

实时 演示

- 演示 1： 直接访问 Linux 内核中的变量
- 演示 2： trace 内核
- 演示 3： 直接访问用户程序的内存
- 演示 4： trace 用户程序的系统调用

灵活 对内核的支持灵活

- 灵活算是轻量级引入的带来的优点。

- 使用 KGTP 不需要 在 Linux 内核上打 PATCH 或者重新编译，只要编译 KGTP 模块并 insmod 就可以。
- 当然 KGTP 也提供 PATCH，从 2.6.18 到 upstream 都能使用，方便 KGTP 的用户集成 KGTP 到其内核 tree。
- 大部分功能在 Linux 内核 2.6.18 到 upstream 上都被可用。
- KGTP 支持 X86-32 ， X86-64 ， MIPS 和 ARM 。
  http://code.google.com/p/kgtp/wiki/HOWTOCN# 配置 KGTP
- 而且还可以用在 Android 上。
  http://code.google.com/p/kgtp/wiki/HowToUseKGTPinAndroid

灵活 连接方式灵活

- 从本机连接 KGTP，刚才已经演示过。

  http://code.google.com/p/kgtp/wiki/HOWTOCN#GDB 在本地主机上

- 从远程主机连接 KGTP。

  http://code.google.com/p/kgtp/wiki/HOWTOCN# 如果 GDB 在远程主机上

  演示 5： 通过网络连接 KGTP

- 即使板子上没有 GDB 而且其没有可用的远程接口， KGTP 也可以用离线调试的功能调试内核。

  http://code.google.com/p/kgtp/wiki/HOWTOCN#/sys/kernel/debug/gtpframe 和离线调试

  演示 6： 使用离线调试

灵活 灵活的设置 tracepoint

- 灵活的在不同的地址设置 tracepoint，只要支持 kprobe 的地址都可以设置 tracepoint。
- Tracepoint action 灵活的设置要收集的数据
- Tracepoint condition 灵活的设置 tracepoint 的条件。
- 和有些 tracer 的区别就是：

  他们关心能 trace 哪里， KGTP 要关心不能 trace 哪里。

  他们关心哪些数据能 trace， KGTP 要关心哪些数据不能 trace。

灵活 通过 TSV 灵活的读取内核数据

- TSV 全名 trace state variables 是 GDB 自带的一种对 tracepoint 进行支持的功能。
- TSV 举例： $bt ， $current_task_pid
- TSV 可以由 target 端也就是 KGTP 定义，也可以由 GDB 端也就是用户来定义。
- 在 tracepoint 任何状态下， GDB 都可以直接读 TSV 。
- tracepoint action 可以读写 TSV 的值。

灵活

- 演示 7

  这个例子记录了每个 CPU 上关闭 IRQ 时间最长的函数的 stack dump 。

灵活 灵活的 KGTP ring buffer

- KGTP 曾经使用过简单 buffer 和 ftrace ring buffer，但是都有限制。
- 使用 KGTP ring buffer 不依赖很多其他内核代码，可以让 KGTP 运行在更多内核版本中。同时 KGTP ring buffer 针对 KGTP 应用场景设计，速度更理想。

灵活 灵活的 KGTP ring buffer

- 可以在 buffer 满的时候 tracepoint 自动停止。
- 可以用 GDB 命令“ set circular-trace-buffer on” 当 buffer 自动删除最早的数据继续 trace 。

  http://code.google.com/p/kgtp/wiki/HOWTOCN# 设置 trace 缓存为循环缓存
- 在 tracepoint 执行的时候，可以读取 /sys/kernel/debug/gtpframe_pipe 直接从 ring buffer 中读取数据。

  http://code.google.com/p/kgtp/wiki/HOWTOCN# 如何使用 _/sys/kernel/debug/gtpframe_pipe

灵活 灵活的数据处理方式 python

- 新版本的 GDB 可以直接用 python 处理 trace frame 中的数据。
- 演示 8： 在演示 7 取得的 trace frame 中用 python 脚本找出每个 CPU 最慢的 frame id
- http://code.google.com/p/kgtp/wiki/hotcode

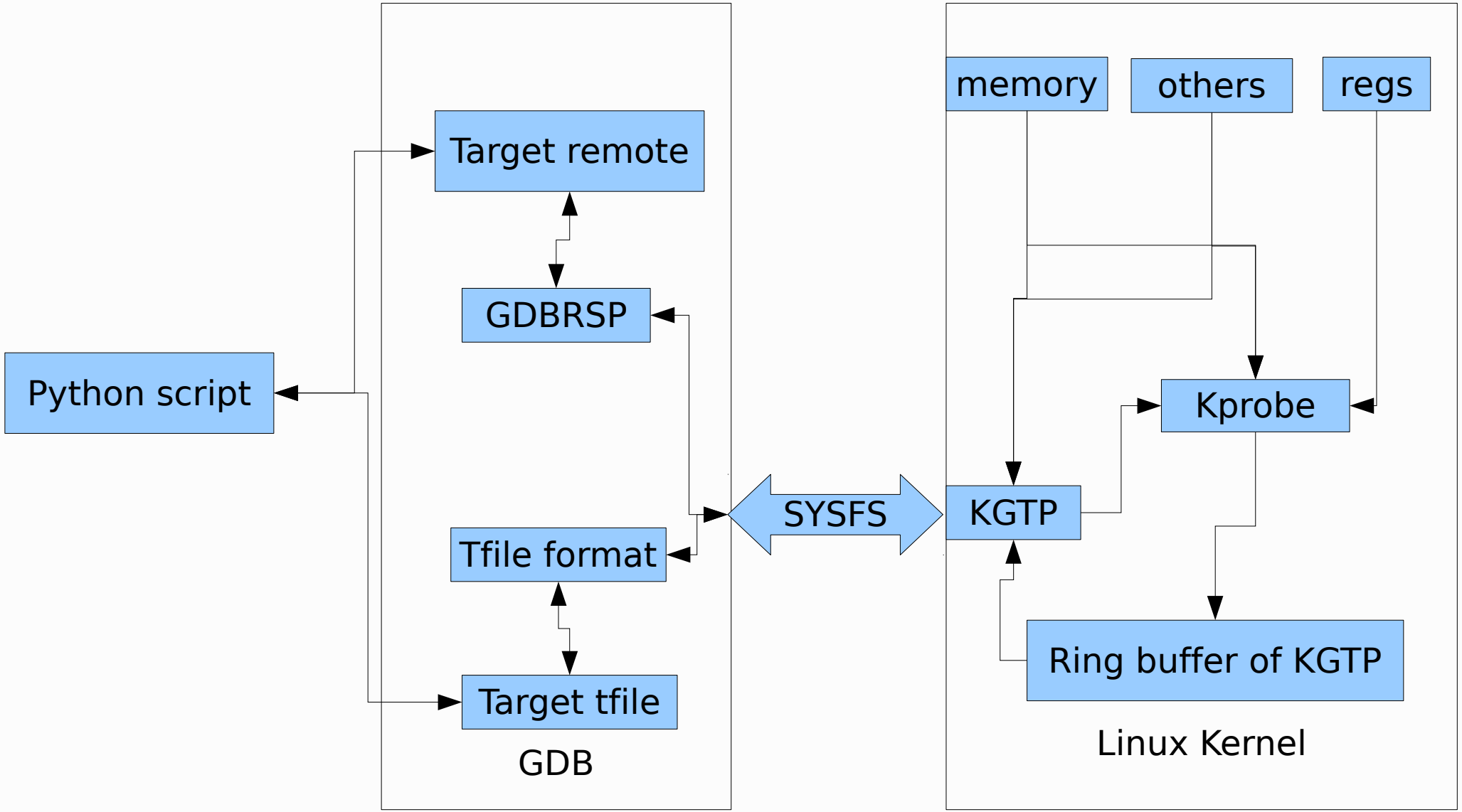
灵活 直接将数据输出到系统日志

- KGTP 可以通过特殊格式的 action 命令输出数据到系统日志。
- 其结合离线调试功能让调试更加方便。
- http://code.google.com/p/kgtp/wiki/HOWTOCN# 如何让 tracepoint 直接输出信息
- 演示 9: 将演示 7 的 action 改为直接输出，并用离线调试。

灵活 通过其他模块对 KGTP 功能进行扩展 plugin

- plugin 可以调用 KGTP 函数 gtp_plugin_var_add 向 KGTP 添加 TSV
- 这样 tracepoint action 或者 GDB 可以访问这些 TSV 从而访问到插件中的内容。
- http://code.google.com/p/kgtp/wiki/HOWTOCN# 如何增加用 C 写的插件
- 演示 10： 演示 KGTP 自带 plugin 例子
- 可支持多个 plugin

KGTP 的结构


Target remote
GDBRSP
Python script
Tfile format
Target tfile
GDB
SYSFS
memory
others
regs
Kprobe
KGTP
Ring buffer of KGTP
Linux Kernel

URL

- 主页 http://code.google.com/p/kgtp/
- 中文 Howto http://code.google.com/p/kgtp/wiki/HOWTOCN
- 报 BUG 提功能 http://code.google.com/p/kgtp/issues/list
- 邮件列表 http://www.freelists.org/list/kgtp
- 我的信箱 teawater@gmail.com
- Irc.freenode.net #hellogcc teawater

谢谢！

问题？