TEXAS ADVANCED COMPUTING CENTER
WWW.TACC.UTEXAS.EDU

TEXAS
The University of Texas at Austin

Compiler Detection

XALT : Compiler Detection

PRESENTED BY:
Amit Ruhela
Texas Advanced Computing Center,
Austin Texas
User Group Meeting, Sept 22, 2022

Agenda

- Compiler Detection
- Code Walkthrough

Compiler Detection

Goal : To learn compilers usage on a HPC cluster

Compiler Information used to :

1. Generate a watermark {xalt_generate_watermark} in ELF binary
2. & Save Link record {xalt_generate_linkdata} for further analysis

File : sh_src/ld.in

```
EXTRACT_LINKER=$XALT_LIBEXEC_DIR/xalt_extract_linker

…

COMP_T=$(XALT_EXECUTABLE_TRACKING=no LD_PRELOAD= PATH=@xalt_system_path@
LD_LIBRARY_PATH=$XALT_LD_LIBRARY_PATH $EXTRACT_LINKER)

...

EPOCH=$(XALT_EXECUTABLE_TRACKING=no LD_PRELOAD= PATH=@xalt_system_path@
LD_LIBRARY_PATH=$XALT_LD_LIBRARY_PATH $GEN_WATERMARK "$UUID" "$SYSHOST" "$WATERMARK_S"
"$COMP_T")

...

XALT_EXECUTABLE_TRACKING=no LD_PRELOAD= PATH=@xalt_system_path@
LD_LIBRARY_PATH=$XALT_LD_LIBRARY_PATH $GEN_LINKDATA "$UUID" "`pwd`" "$SYSHOST" "$EXEC" "$SHA1"
"$WATERMARK_O" "$EPOCH" "$FUNCRAW" "$LINKLINE_OUT" "$COMP_T“

...
```

Implementation

SRC Code : [xalt/src/linker/xalt_extract_linker.C]

Function : void extract_linker(std::string& compiler, std::string& compilerPath,
UT_array** linklineA, UT_array** p_parentProcA);

Executable: <install-dir>/libexec/xalt_extract_linker

Example:

$ XALT_TRACING=link mpicc hello.c -o hello

COMP_T:{"compiler":"mpiicc(icc)",
"compilerPath":"icc:/opt/intel/compilers_and_libraries_2020.1.217/linux/bin/intel64/icc",
"link_line":["icc","hello.c","-o","hello","-I/opt/intel/compilers_and_libraries_2020.4.304/linux/mpi/intel64/include",
"-L/opt/intel/compilers_and_libraries_2020.4.304/linux/mpi/intel64/lib/release", "-
L/opt/intel/compilers_and_libraries_2020.4.304/linux/mpi/intel64/lib", "-Xlinker","--enable-new-dtags","-Xlinker",
"-rpath","-Xlinker","/opt/intel/compilers_and_libraries_2020.4.304/linux/mpi/intel64/lib/release","-Xlinker","-rpath","-
Xlinker","/opt/intel/compilers_and_libraries_2020.4.304/linux/mpi/intel64/lib","-lmpifort","-lmpi","-ldl","-lrt",
"-lpthread"],"
parentProcs":["mpiicc:/usr/bin/bash","mpicc:/usr/bin/bash","bash:/usr/bin/bash","sshd:","sshd:","sshd:"]}

A few Unix Utilities

Get Name of the Executable and its Parent Process id

1. Getppid() : Get the parent process ID of the calling process.

2. /proc/[pid]/stat

Status information about the process. This is used by ps(1). The fields, in order, with their proper scanf(3) format specifiers, are:

pid %d (1) The process ID.

comm %s (2) The filename of the executable, in parentheses. This is visible whether or not the executable is swapped out.

state %c (3) One character from the string "RSDZTW" where R is running, S is sleeping in an interruptible wait, D is waiting in uninterruptible disk sleep, Z is zombie, T is traced or stopped (on a signal), and W is paging.

ppid %d (4) The PID of the parent.

```
staff.frontera(1017)$ ps -fjH -u aruhela
UID          PID    PPID    PGID     SID  C STIME TTY          TIME CMD
aruhela   190086  190030  190030  190030  0 14:46 ?        00:00:00 sshd: aruhela@pts/133
aruhela   190087  190086  190087  190087  0 14:46 pts/133  00:00:02   -bash
aruhela   189010  188979  188979  188979  0 14:44 ?        00:00:00 sshd: aruhela@pts/130
aruhela   189011  189010  189011  189011  0 14:44 pts/130  00:00:00   -bash
aruhela   280156  189011  280156  189011  0 21:21 pts/130  00:00:00     ps -fjH -u aruhela
aruhela   115050       1  115050  115050  0 Aug24 ?        00:00:19 ssh: /home1/05231/aruhela/.ssh/cm_socket/aruhela@ls6.tacc.utexas.edu:22 [mux]
```

```
staff.frontera(1021)$ cat /proc/190086/stat
190086 (sshd) S 190030 190030 190030 0 -1 1077944640 576 42219 0 0 24 13 23 13 20 0 1 0 1389832887 123703296 633 184
46744073709551615 1 1 0 0 0 0 0 4096 65536 18446744073709551615 0 0 17 22 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Get pathname of the Executable

/proc/[pid]/exe

This file is a symbolic link containing the actual pathname of the executed command.

readlink() places the contents of the symbolic link pathname in the buffer buf, which has size bufsiz.
readlink does not append a terminating null byte to buf. It will (silently) truncate the contents (to a length of bufsiz characters), in case the buffer is too small to hold all of the contents.

Get command line of the executable

/proc/[pid]/cmdline

This holds the complete command line for the process, unless the process is a zombie. In the latter case, there is nothing in this file: that is, a read on this file will return 0 characters. The command-line arguments appear in this file as a set of strings separated by null bytes ('\0'), with a further null byte after the last string.

Code Flow

Loop until Init process (pid=1) or found the compiler

- Get Parent Process ID & Executable Full PATH(/proc/%d/stat, /proc/%d/exe)
- Continue Loop if the process name matches ignore list
- Record Compiler Name, Path and Link Arguments

Loop until init process (pid=1)

- Verify if compiler is "rustc", "chpl", "nim", "ghc"
- Verify if Compiler is MPI
- Record Compiler Name and Path in parent procs

Code Walkthrough

https://github.com/xalt/xalt/blob/master/src/linker/xalt_extract_linker.C

Next Meeting

. Oct 27th at 10:00 AM CST (15:00 UTC)

Thanks for Listening

Links

1. XALT Documentation : https://xalt.readthedocs.io/en/latest/index.html#
2. XALT LD: https://github.com/xalt/xalt/blob/master/sh_src/ld.in
3. Bash https://pubs.opengroup.org/onlinepubs/9699919799/utilities/V3_chap02.html#tag_18_06_02
4. Shared vs Static Libraries : https://www.linuxtopia.org/online_books/an_introduction_to_gcc/gccintro_25.html