

# **XALT Monthly Meeting: Using \$XALT\_TRACING to Debug XALT**

Robert McLay

Nov. 11, 2021

# XALT: Outline



# XALT

- ▶ XALT provides a way to debug its behavior
- ▶ `xalt_extract_record ./user_program`: XALT watermark
- ▶ `XALT_TRACING=link icc -o try try.c`
- ▶ `XALT_TRACING=run ./user_program ...`
- ▶ It shows what decisions XALT is making.

# XALT Watermark: xalt\_extract\_record mpi\_hello

```
$ xalt_extract_record ./mpi_hello
```

```
*****
```

```
XALT Watermark: ./mpi_hello
```

```
*****
```

Build_CWD	/home/user/t
Build_EPOCH	1636147194.4143
Build_LMFILES	...
Build_LOADEDMODULES	intel/19.1.1:impi/19.0.9:...:xalt/2.10.33
Build_OS	Linux 3.10.0-1160.45.1.el7.x86_64
Build_Syshost	frontera
Build_UUID	70611e21-bea5-465f-bf12-9537dc565240
Build_User	joe_user
Build_Year	2021
Build_compiler	mpiicc(icc)
Build_compilerPath	icc:/opt/intel/compilers_and_libraries_2020.1.217/...
Build_date	Fri Nov 05 16:19:54 2021
Build_host	staff.frontera.tacc.utexas.edu
XALT_Version	2.10.34

# XALT Watermark checks

- ▶ Set \$COMPILER\_PATH to /path/to/xalt/xalt/bin
- ▶ Use priority path in XALT Module

```
local base = "/path/to/xalt/xalt" --> Change to match  
local bin  = pathJoin(base,"bin")  
prepend_path{"PATH", bin, priority="100"}
```

# Linking w/ XALT

```
$ XALT_TRACING=link gcc -o hello hello.c
```

```
14:57:32 XALT Tracing Activated for version: xalt-2.10.31
```

```
14:57:32 Starting ld
```

```
14:57:32 find_real_command: Searching for the real: ld
```

```
14:57:32 find_real_command: found /opt/apps/gcc/8.3.0/bin/ld
```

```
14:57:32 COMP_T: "compiler":"gcc","compilerPath":"/opt/apps/gcc/8.3.0/bin/gcc","link_line":  
o","hello","hello.c"],"parentProcs":["zsh:/usr/bin/zsh","sshd:","sshd:","sshd:"]
```

```
14:57:32 SYSHOST: frontera
```

```
14:57:32 WRKDIR: /tmp/user_dd798a17-2dda-411f-8029-077bf67ad3a4_FChre5
```

```
14:57:32 EPOCH 1636142252.7051
```

```
14:57:32 Using ld to find functions: /opt/apps/gcc/8.3.0/bin/ld ...
```

```
14:57:32 Not adding XALT initialize routines to user code
```

```
14:57:32 /opt/apps/gcc/8.3.0/bin/ld ...
```

```
14:57:32 XALT_TRANSMISSION_STYLE file  
Using USE_HOME method to store json file  
Wrote json link file :
```

# Source Code

- ▶ The first set of tests come from `src/libxalt/xalt_initialize.c`
- ▶ The second test of test come from `src/libxalt/run_submission.c`

# Tracing on login node

```
staff.fta(3013)% XALT_TRACING=run ./hello
```

```
-----  
Date:          Fri Nov  5 13:16:26 2021  
XALT Version:  xalt-2.10.31  
Nodename:      staff.frontera.tacc.utexas.edu  
System:       Linux  
Release:      3.10.0-1160.45.1.el7.x86_64  
O.S. Version:  #1 SMP Wed Oct 13 17:20:51 UTC 2021  
Machine:      x86_64  
-----
```

```
myinit(0/1,LD_PRELOAD,/home/user/t/xalt/hello){  
  Test for __XALT_INITIAL_STATE__: "(NULL)", STATE: "LD_PRELOAD"  
  Test for XALT_EXECUTABLE_TRACKING: yes  
  Test for rank == 0, rank: 0  
    hostname: "staff.frontera.tacc.utexas.edu" is rejected -> exiting  
}
```

Hello World!

```
myfini(0/1,LD_PRELOAD,/home/user/t/xalt/hello){  
  -> exiting because reject is set to: XALT has found the host does  
    not match hostname pattern. If this is unexpected check your  
    config.py file: Config/TACC_config.py for program: /home/user/t/xalt/hello  
}
```

# Tracing on compute node

```
c202-001.fta[clx](3017)% XALT_TRACING=run ./hello

myinit(0/1,LD_PRELOAD,/home/user/t/xalt/hello){
  Test for __XALT_INITIAL_STATE__: "LD_PRELOAD", STATE: "LD_PRELOAD"
  Test for XALT_EXECUTABLE_TRACKING: yes
  Test for rank == 0, rank: 0
  GPU tracing is turned off. This directory "/sys/module/nvidia" does not exist!
  No GPU tracking
    -> Found watermark via vendor note: true
    -> MPI_SIZE: 1 < MPI_ALWAYS_RECORD: 128, XALT is build to track
        all programs, Current program is a scalar
        program -> Not producing a start record
    -> Signals capturing disabled
    -> Leaving myinit
}

Hello World!

myfini(0/1,LD_PRELOAD,/home/user/t/xalt/hello){
  -> exiting because sampling. run_time: 0.000298977, (my_rand:
  0.943536 > probab: 0.0001)
  for program: /home/user/t/xalt/hello
}
```



# Tracing on compute node, sampling off

```
c202-001.fta[clx](3017)% XALT_TRACING=run XALT_SAMPLING=no ./hello

myinit(0/1,LD_PRELOAD,/home/user/t/xalt/hello){
    ...
}

Hello World!

myfini(0/1,LD_PRELOAD,/home/user/t/xalt/hello){
    -> XALT_SAMPLING = "no" All programs tracked!
    Recording State at end of scalar user program

    run_submission(.zzz) {
        Built processTree table
        Built envT
        Extracted recordT from watermark. Compiler: icc, XALT_Version: 2.10.31
        Built userT, userDT, scheduler: SLURM
        Compute sha1 (...) of exec: /home/user/t/xalt/hello
        Parsed ProcMaps
        Using XALT_TRANSMISSION_STYLE: file
        Built json string
        Transmitting jsonStr
        Using USE_HOME method to store json file
        Wrote json run file : ...
        Done transmitting jsonStr
    }

    -> leaving myfini
}
```

# Tracing MPI program (I)

```
$ cat run_me.sh  
XALT_MPI_ALWAYS_RECORD=2 XALT_TRACING=run XALT_SAMPLING=no ./mpi_hello
```

# Tracing MPI program (II)

```
$ ibrun -np 2 ./run_me.sh
myinit(0/2,LD_PRELOAD,/home/user/t/xalt/mpi_hello){
...
-> Found watermark via vendor note: true
-> MPI_SIZE: 2 >= MPI_ALWAYS_RECORD: 2 => recording start record!
Recording state at beginning of MPI user program:
/home/user/t/xalt/mpi_hello

run_submission(.aaa) {
  Built processTree table
  Built envT
  Extracted recordT from watermark. Compiler: mpicc(icc), XALT_Version: 2.10.31
  Built userT, userDT, scheduler: SLURM
  Compute sha1 (...) of exec: /home/user/t/xalt/mpi_hello
  Parsed ProcMaps
  Using XALT_TRANSMISSION_STYLE: file
  Built json string
  Transmitting jsonStr
  Using USE_HOME method to store json file
  Wrote json run file : ...
  Done transmitting jsonStr
}
-> uuid: b2f87517-8878-4449-bc93-5d899456b3c2
-> Signals capturing disabled
-> Leaving myinit
}
Hello World from proc: 0 out of 2 !
...
```

# Tracing MPI program (III)

```
$ ibrun -np 2 ./run_me.sh
...
Hello World from proc: 0 out of 2 !

myfini(0/2,LD_PRELOAD,/home/user/t/xalt/mpi_hello)
Recording State at end of MPI user program

run_submission(.zzz)
  Built processTree table
  Built envT
  Extracted recordT from watermark. Compiler: mpicc(icc), XALT_Version: 2.10.31
  Built userT, userDT, scheduler: SLURM
  Reuse sha1 (...) of exec: /home/user/t/xalt/mpi_hello
  Parsed ProcMaps
  Using XALT_TRANSMISSION_STYLE: file
  Built json string
  Transmitting jsonStr
  Using USE_HOME method to store json file
  Wrote json run file : ...
  Done transmitting jsonStr

-> leaving myfini

TACC: Shutdown complete. Exiting.
```

# XALT Env. Vars for Tracing

- ▶ `xalt_extract_record user_program` to see the XALT Watermark
- ▶ `XALT_TRACING=run` for execution
- ▶ `XALT_TRACING=link` for linking
- ▶ `XALT_SAMPLING=no` to turn off sampling
- ▶ `XALT_TESTING_RUNTIME=3600` to override real runtime
- ▶ `XALT_MPI_ALWAYS_RECORD=2` to force start and end record generation.
- ▶ `XALT_COMM_WORLD_SIZE=256` to override number of real mpi tasks (XALT-2.10.34+)

# Future Topics?

- ▶ Python package tracking
- ▶ R package tracking
- ▶ MATLAB package tracking