

Debugging Impala Core Dumps on Another System

Background

For debugging problems on remote systems, it can be a pain to use core dumps if they are generated by a different Linux flavour (e.g. CentOS vs Ubuntu). Different OS distributions and version have different system shared libraries from the Centos machines, so the symbols in those libraries are useless and make it impossible to get stack traces in many circumstances.

Core files can, however, be "remotely" debugged if we grab the required libraries and debug symbols from the remote system.

Procedure

On the system where impalad crashed:

- Find out the shared libraries referenced by impalad

```
gdb -c core /path/to/impalad  
(gdb) info shared
```

- Take this list of .so's and tar them up, being sure to follow symlinks:

```
tar chvzf so.tar.gz <list of so files>
```

- Copy the core file, impalad, impalad.debug, and so.tar.gz files in the jira directory on impala-desktop.

On the system where you'll debug:

- Extract the so files to a directory

```
tar xvf so.tar.gz
```

- Launch GDB

```
gdb impalad
```

- That should find and load symbols from impalad.debug automatically.
 - If symbols look messed up after this, you can try combining the impalad and impalad.debug files into one using the eu-unstrip tool (usually found in the elfutils package).

```
eu-unstrip -o impalad.full impalad impalad.debug  
gdb impalad.full
```

- Tell gdb to look for the copied so's rather than the system so's

```
(gdb) set sysroot /path/to/extracted/so/tar
```

- Now load the core file

```
(gdb) core core
```

- You should now be able to move around and debug the core file as normal!