1 How To Deploy Hydra

1.1 Dependencies

All projects in Hydra target **dot-net** framework version 4.5. You will also need **Z3** version 4.6.0 from here :
`https://github.com/Z3Prover/z3/releases/tag/z3-4.6.0`.

1.1.1 For Windows Platform

1. Install **dot-net** framework from here :

   `https://dotnet.microsoft.com/download`.

2. **Optional:** Get **Visual Studio** if you are interested in making changes.

1.1.2 For Linux Platform

1. Install **dot-net** framework from here :

   `https://docs.microsoft.com/en-us/dotnet/core/install/linux-ubuntu`.

2. Get **Mono** from here :

   `https://www.monodevelop.com/download/#fndtn-download-lin`.

1.2 Building Hydra

1. Pull the *hydra* branch from this repository :

   `https://github.com/boogie-org/corral`.

2. Navigate to `<path-where-you-pulled-hydra>` and update the Boogie submodule by doing:

   git submodule init

   git submodule update

3. Build `<path-where-you-pulled-hydra>/AddOns/DistributedService/LocalServerInCsharp/LocalServerInCsharp.sln`.

   You can use **Visual Studio** or, **MonoDevelop** or,

   msbuild LocalServerInCsharp.sln from terminal or,

   xbuild LocalServerInCsharp.sln from terminal to build Hydra.

   This will build all the required binaries in

   `<path-where-you-pulled-hydra>/AddOns/DistributedService/LocalServerInCsharp/LocalServerInCsharp/bin/Debug/`.

4. Copy the **Z3** binary from where you downloaded **Z3** version 4.6.0 to

   `<path-where-you-pulled-hydra>/AddOns/DistributedService/LocalServerInCsharp/LocalServerInCsharp/bin/Debug/`

1.3 Running Hydra

1.3.1 For Windows Platform

Run `<path-where-you-pulled-hydra>/AddOns/DistributedService/LocalServerInCsharp/LocalServerInCsharp/bin/Debug/LocalServerInCsharp.exe` with the following arguments:

1. Path of the program to verify
2. Path to the **Configuration** file which defines the settings for Hydra. A sample **Configuration** file os provided in:

   `<path-where-you-pulled-hydra>/AddOns/DistributedService/LocalServerInCsharp/LocalServerInCsharp/config.txt`

1.3.2 For Linux Platform

Run **mono** from the terminal with the following arguments:

1. `<path-where-you-pulled-hydra>/AddOns/DistributedService/LocalServerInCsharp/LocalServerInCsharp/bin/Debug/LocalServerInCsharp.exe`
2. Path of the program to verify
3. Path to the **Configuration** file which defines the settings for Hydra. A sample **Configuration** file os provided in:

   `<path-where-you-pulled-hydra>/AddOns/DistributedService/LocalServerInCsharp/LocalServerInCsharp/config.txt`

1.4 Explanation Of the Hydra Configuration

The configuration file accepts the following settings:

1. **numListeners:** Set the value to the number of machines on which you want to run Hydra clients.
2. **numMaxClients:** Set the value to the number of clients you want to run per machine.
3. **timeout:** How long should verification run before timing out (in seconds).
4. **hydraBin:** Path where you have built the Hydra binaries. It should be

   `<Path-where-you-pulled-the-hydra-repository>/AddOns/DistributedService/LocalServerInCsharp/LocalServerInCsharp/bin/Debug`
5. **smackBin:** Path where you have built the Smack binaries. If the input program is not a **boogie** program, Hydra will use Smack to compile the given **C** program to **boogie** and dump it in the location specified by **boogieDumpDirectory**. This compiled **boogie** will then be verified by Hydra. If Smack is already added to path, then this flag is not required.

6. **inputFilesDirectoryPath:** If you are invoking Hydra with a single program (the second argument while running **LocalServerInCsharp.exe**), this setting is irrelevant. If you invoke Hydra without specifying an input program, Hydra will verify each program in this directory. For each program ⟨filename.bpl⟩, Hydra will write the result (outcome, total time etc.) in ⟨filename.bpl.txt⟩ in the same directory.

7. **dumpSIBoogieFiles:** Set this to false in case you are feeding a SI boogie file to Hydra. Note that, if your SI boogie file was not generated using Corral 1.0.12, Hydra may crash as Hydra uses Corral 1.0.12 for verification. Otherwise, if your input is an original boogie program, set this to *true*. Hydra will then dump the SI boogie file and verify that.

8. **boogieDumpDirectory:** Path to the directory where Hydra will dump SI boogie files if you set **dumpSIBoogieFiles** to *true*. If the directory does not exist, it will be created, but this location should be writable.

9. **serverAddress:** ip address and port of the Hydra server. You may need to set custom in/out tcp and udp rules (in Linux, use *ufw allow* ⟨*port-number*⟩) in order to enable the server to listen the specified port. If you are running server and clients on the same machine, you can set it to `http://localhost:<port-number>`

10. **corralArguments:** set of arguments which Hydra will use to dump intermediate SI boogie files if you set **dumpSIBoogieFiles** to *true*.

11. **hydraArguments:** set of arguments which Hydra will use to verify the intermediate SI boogie files.

12. **startLocalListener:** setting this to *true* will let Hydra run clients on the Server machine as well.

13. **ListenerAddress:** $user - name$ and $ip - address$ of a remote machine where you want to run Hydra clients. If you do not want to use remote machines, you can remove this. If you want to use remote machines, make sure that the server can *ssh* to the remote machines without password and they have **dot-net** libraries and **mono** installed. If you have multiple remote machines, specify each one in a separate line with this flag. (**This is only for Linux**)

14. **ListenerExecutablesPath:** You need to specify a path for each remote machine where the server will automatically set up Hydra binaries. This location should be writable in the remote machine. If you do not want to use remote machines, you can remove this. If you have multiple remote machines, specify the path for each one in a separate line with this flag. (**This is only for Linux**)

Note that, if you want to distribute verification over multiple **Linux** machines, you can do so by specifying **ListenerAddress** and **ListenerExecutablesPath** for each one. HYDRA will automatically setup the remote machines and distribute verification tasks. However, if you are using multiple **Windows** machines, then you will need to setup HYDRA and start the `<path-where-you-pulled-hydra>/AddOns/DistributedService/LocalServerInCsharp/LocalServerInCsharp/bin/Debug/Client.exe` on each one manually.

2 How to distribute Hydra on Windows

The following instructions assume that you are going to run hydra on 2 machines. Machine $m1$ will run the server along with 32 clients and $m2$ will run another 32 clients. So, overall you will be verifying each program with 64 clients.

1. Pull the "hydra" branch from the corral repository in $m1$. I will assume that you have pulled it to `C:\hydra`.

   (a) Copy the benchmarks to be verified in some directory. I will assume this is `C:\benchmarks`.

   (b) Setup dotnet framework following the links in Section 1.1.1.

   (c) Build `C:\hydra\AddOns\DistributedService\LocalServerInCsharp\LocalServerInCsharp.sln`.

   (d) Copy $Z3$ version 4.6.0 in `C:\hydra\AddOns\DistributedService\LocalServerInCsharp\LocalServerInCsharp\bin\Debug`.

   (e) You will need to provide a config file to HYDRA. Explanations for each flag in the config file is provided in Section 1.4.

   (f) In the config file, set
   numListeners = 2
   maxClients = 64
   timeout = 3600
   hydraBin = `C:\hydra\AddOns\DistributedService\LocalServerInCsharp\LocalServerInCsharp\bin\Debug\`
   inputFilesDirectoryPath = `C:\benchmarks\`
   startLocalListener=true

   (g) Now, you will need to enter the ip address of the server and port so that clients from $m2$ can send messages to the server in $m1$. You can look up this link to figure out the ip address of $m1$. Look up this link to figure out how to open a port, e.g., 5000 to allow incoming and outgoing messages. Once you have done so, you can set
   serverAddress = `http://<ip-address>:5000/`

   (h) Now, from inside `C:\hydra\AddOns\DistributedService\LocalServerInCsharp\LocalServerInCsharp\bin\Debug`, you can execute hydra with:
   LocalServerInCsharp.exe config.txt
   Doing this will spawn the server on $m1$. However, verification will not start as the server is still waiting for the clients to start on $m2$.

2. In $m2$, follow the similar steps as $m1$ except you do not have to create/modify any config file. You can simply copy the config file from $m1$ to $m2$. Once you have done so, from inside `C:\hydra\AddOns\DistributedService\LocalServerInCsharp\LocalServerInCsharp\bin\Debug` in $m2$, execute

   Client.exe config.txt `C:\hydra\AddOns\DistributedService\LocalServerInCsharp\LocalServerInCsharp\bin\Debug\`