Hitachi Data Systems

HITACHI
Inspire the Next

Installing ivy

Allart Ian Vogelesang ian.vogelesang@hds.com

2016-05-19

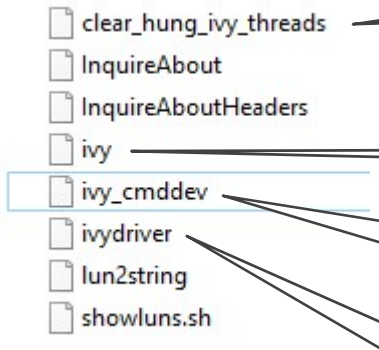
Prepare executables

HITACHI
Inspire the Next

- Source for LUN_discovery is at https://github.com/Hitachi-Data-Systems/LUN_discovery
- Source for ivy is at https://github.com/Hitachi-Data-Systems/ivy
- Build using the "codeblocks" IDE http://codeblocks.org/
- Have not figured out how/where to post pre-built binaries.
  - ian.vogelesang@hds.com

ivy family executables

HITACHI
Inspire the Next



Some failure modes may leave "orphan" threads hanging either on the ivy master host or on test (I/O driver) hosts. Use this on the affected host to remove them. "clear_hung_ivy_threads.sh" is built each time ivy runs, and will clear hung threads on all test hosts.

The "main" ivy command line executable

Hitachi-proprietary command device connector.

Remotely invoked using SSH on the test host having a command device to a subsystem under test.

Only available to authorized Hitachi internal users, and employs a license key mechanism.

SCSI Inquiry-based "`LUN_discovery`" toolset.

Remotely invoked using SSH on each test to discover the attributes of all storage LUNs.

Separately packaged from ivy.

Invoked by "`ivy`" using SSH on each ivy I/O driver test host

1) Put `ivy` & `LUN_discovery` executables in a folder

HITACHI
Inspire the Next

- For example, put in `/usr/local/bin`
- For HDS performance team users
  - Check that the test hosts have `/scripts` mapped to "the filer".
  - The ivy executables are in `/scripts/ivy/bin`
  - ivy was link-edited with dynamic links to the normal C library routines, that is, ivy uses each test host's own C library routines.
  - ivy was statically linked with the C++ libraries on Ian's Linux development host, meaning that ivy uses the C++ libraries from Ian's Linux development host, which have been copied into the ivy executables, making them quite big, but ensuring that ivy will work on hosts that don’t have up-to-date C++ libraries.

2) Put executable folder in *background* PATH

HITACHI
Inspire the Next

- The folder containing the ivy binary executables must be put in the environment PATH variable for background tasks on all hosts running ivy.
  - ivy uses SSH to remotely invoke executables on test hosts (I/O driver hosts), and invoked this way, the remote ivy executables run as a "background" process.
  - For background processes, the normal BASH or `/etc/profile` login profile files associated with foreground command windows are not executed.
- Edit `ivy_etc_profile_d.sh` to reflect the ivy binary folder path, and place the edited file in `/etc/profile.d` to put ivy in the PATH for background processes.
  - Even for background processes, all the scripts in `/etc/profile.d` are executed when the process starts up.

Sample `/etc/profile.d/` script

HITACHI
Inspire the Next

- You only need this if the folder where you put the executables is not already in the PATH environment variable for background processes.

This example is for when the executables were put in `/scripts/ivy/bin`.

```
#!/bin/bash

if ! echo ${PATH} | /bin/grep -q /scripts/ivy/bin ; then

        PATH=${PATH}:/scripts/ivy/bin
        export PATH
fi
```

3) Set up certificate-based SSH logins

HITACHI
Inspire the Next

- Certificate-based SSH logins must be set up so the central test control host (running the "ivy" executable) can SSH into the I/O driver hosts without SSH asking for a password.
- Search for "certificate based SSH logins" to find instructions on how to do this.

ivy has only been tested running as `root`

HITACHI
Inspire the Next

- The `ivydriver` executable may or may not need to run as `root`.
  - This may possibly be required for the ivydriver executable to perform I/O to "raw" LUNs without a file system – not tested yet.
- SCSI Inquiry commands definitely can only be run as root, and therefore the `InquireAbout` executable is "`setuid`" to `root`.
  - `InquireAbout`, `InquireAboutHeaders`, and `showluns.sh` are executables forming part of Ian's `LUN_discovery` tool package.
  - `lun2string` is also part of Ian's LUN lister tool, but is not used by ivy.
    - lun2string lets you build a text string, plugging in decoded Hitachi proprietary LUN attributes.

Ivy output folder root

HITACHI
Inspire the Next

- When ivy runs a program like "`xxxx.ivyscript`", an output folder named `xxxx` is created in a root folder specified by the `[OutputFolderRoot]` statement in the `.ivyscript` program.
  - The default is `"."`, the current directory.
- To put the output somewhere else, put the following statement in your `.ivyscript` program:
  - `[OutputFolderRoot] "/your/output/file/root/folder";`
  - Note that for this one ivyscript statement, the operand may not be a string expression; it must be a string literal (a string constant).
  - This is because ivy creates the output folder for a test run after compiling the `.ivyscript` program, but before the `.ivyscript` program starts to run. (`[OutputFolderRoot]` is evaluated at compile time.)

HITACHI

Inspire the Next