ivy output csv files

and the ivy output csv file loader Excel tool

2016-03-30

Allart Ian Vogelesang ian.vogelesang@hds.com +1 408 396 6511

About this presentation

HITACHI
Inspire the Next

- Prerequisite
  - "introduction to ivy" PowerPoint material.
- Please go over this presentation to learn how to explore the ivy demo output.
- The "programming ivy reference" presentation has full details and can actually be read standalone, but it's recommended to go through this presentation and review the demo video series first.

ivy engine control statements

HITACHI
Inspire the Next

- The `[hosts]` statement specifies the test hosts (I/O driver hosts) to use, and the `[select]` clause in that statement is used to filter "all discovered LUNs" to arrive at "available test LUNs".
- The `[CreateWorkload]` sets up I/O driver threads on test LUNs.
- The `[CreateRollup]` statement sets up two-way structures used to centrally roll up detail data from test host workloads, and to send out I/O driver parameter updates to those workloads.
- The `[Go]` statement runs a subinterval sequence.

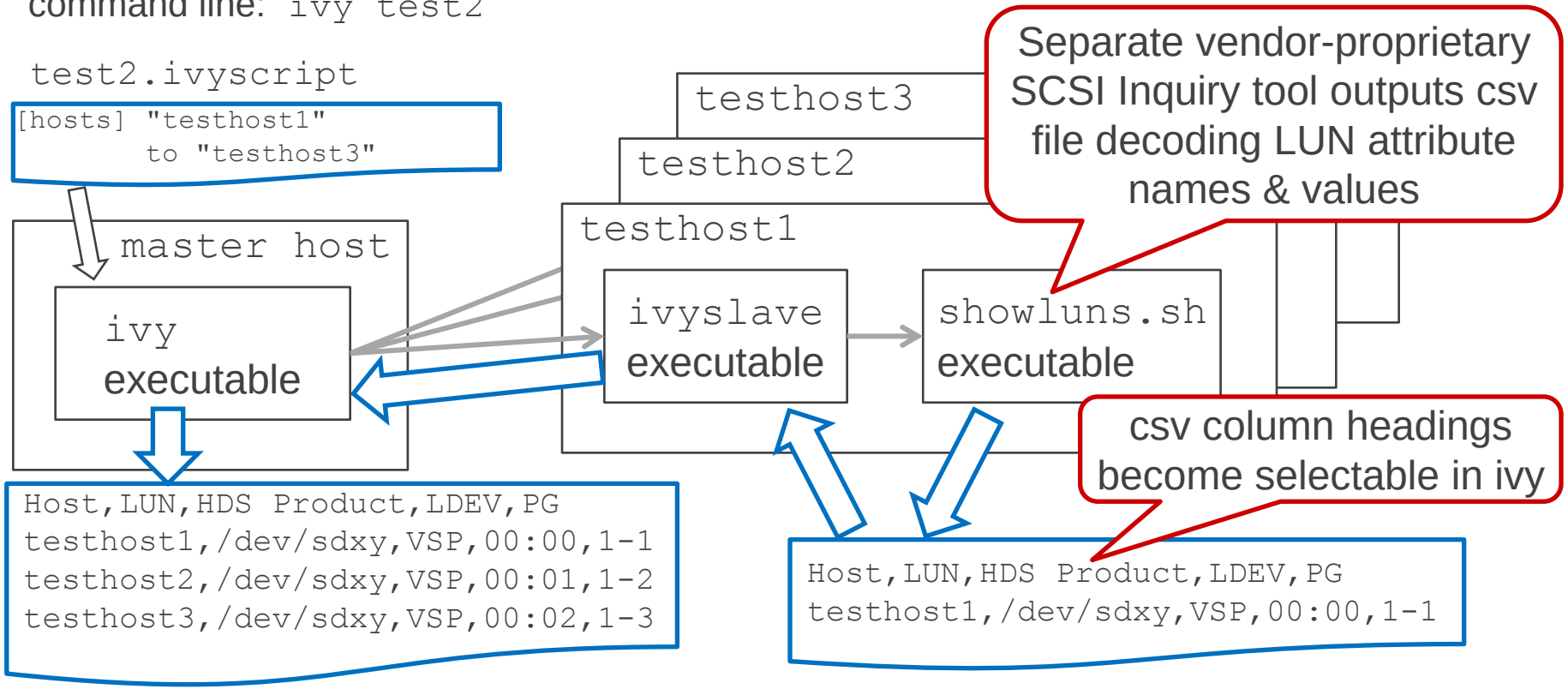
[hosts] statement – configuration discovery

HITACHI
Inspire the Next

command line: `ivy test2`

test2.ivyscript

```
[hosts] "testhost1"
        to "testhost3"
```



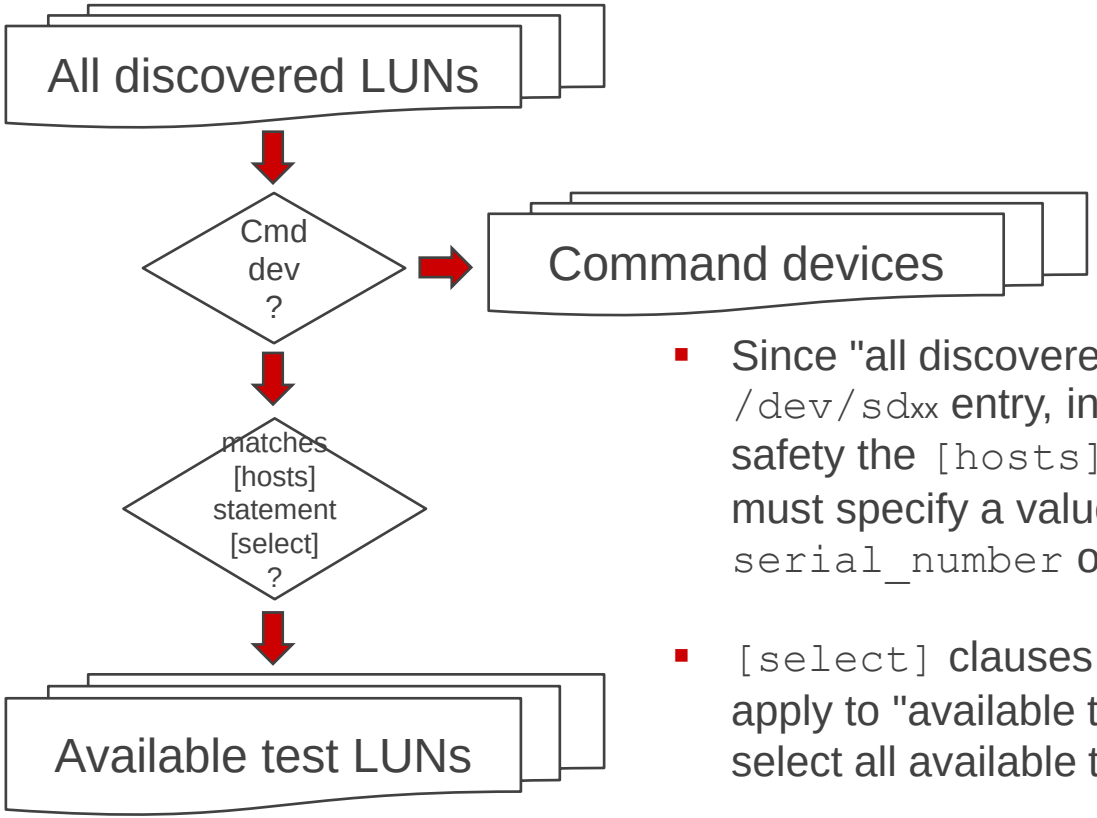
"All discovered LUNS"

HITACHI
Inspire the Next

```
Host,LUN,HDS Product,LDEV,PG
testhost1,/dev/sdxy,VSP,00:00,1-1
testhost2,/dev/sdxy,VSP,00:01,1-2
testhost3,/dev/sdxy,VSP,00:02,1-3
```

- The "showluns.sh" csv files from all the test hosts are combined
- Each data line is loaded into a "LUN" object where for each column in the csv file, we file the data value into the LUN under the attribute name taken from the corresponding header line column.
- Later on, if we find out more information about this LUN, using a command device, we may fill in more attribute values.
- **KEY ivy CONCEPT: ivyscript `[select]` clauses operate on LUN attributes**.

Available test LUNs



- Since "all discovered LUNs" contains every `/dev/sdxx` entry, including test host boot drives, for safety the `[hosts]` statement `[select]` clause must specify a value for at least one of `serial_number` or `vendor`.
- `[select]` clauses for other ivyscript statements apply to "available test LUNs" and the default is to select all available test LUNs.

Augmenting SCSI Inquiry data using cmd. dev.

HITACHI
Inspire the Next

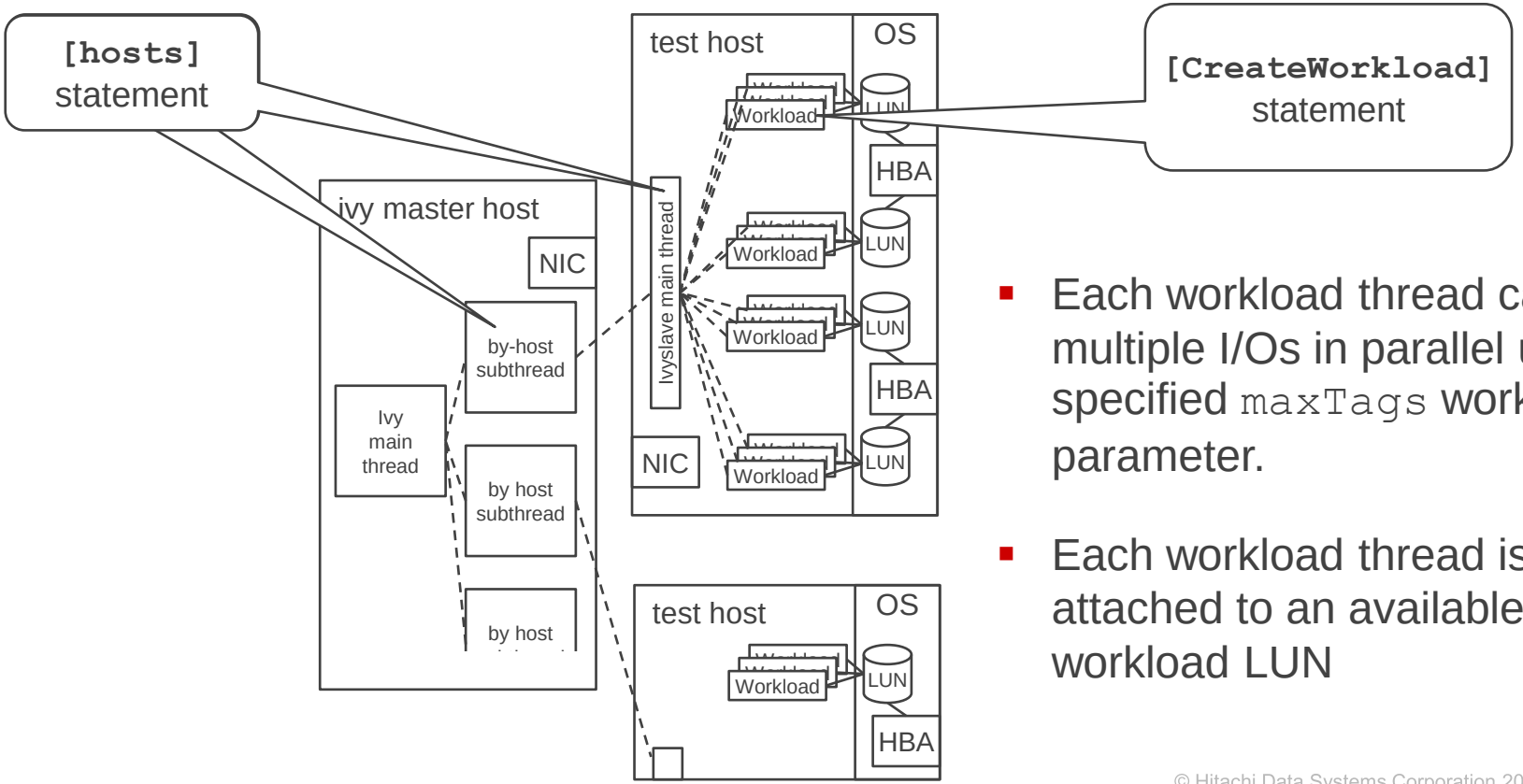
- For every Available Test LUN that is mapped to a Hitachi RAID subsystem LDEV for which we have a command device:

1. We augment the Available Test LUN SCSI Inquiry attributes with the RMLIB API attributes of the underlying LDEV
   - Enables things like `[select] "drive_type = \"NFH1B-P3R2SS\""`

2. Using the RMLIB API configuration data and using static tables of relationships for the specific subsystem model, we further augment the LUN with "indirect" attributes.
   - Tagging the LUN with the names of the associated subsystem configuration elements of different types enables us to filter real-time RMLIB API performance data by the workload LUNs which comprise a rollup instance.

Setting up test host workload threads

HITACHI
Inspire the Next



- Each workload thread can issue multiple I/Os in parallel up to the specified `maxTags` workload parameter.
- Each workload thread is attached to an available workload LUN

The [go] statement launches a "test step"

HITACHI
Inspire the Next

- [go] starts the workload threads running a series of "subintervals", typically each 5 seconds long.
- At the end of each subinterval, data are rolled up to the ivy master host.
  - Test host workload data
  - Command device data, if a command device was discovered
- The master host examines the data, optionally sends out real-time I/O generator updates to the workload threads, and then decides whether to continue for another subinterval or not.

For each subinterval within a test step …

HITACHI
Inspire the Next

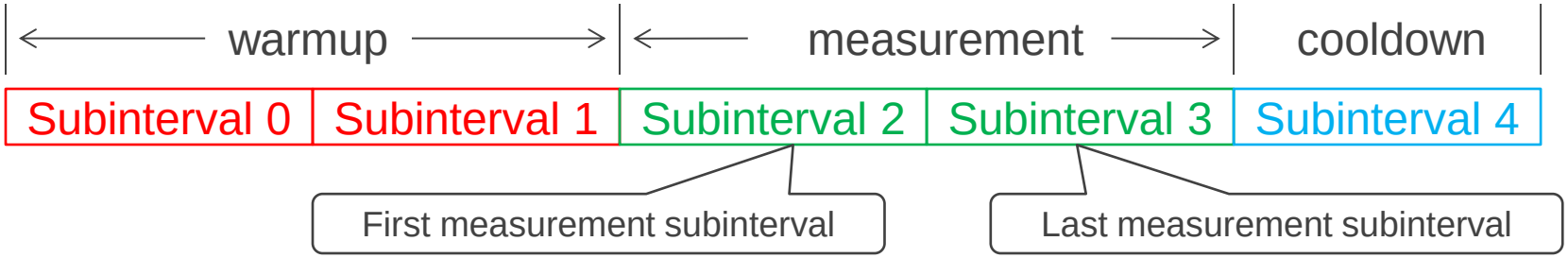
- You get a csv line in each csv file within the test step subfolder
- Within the test step subfolder
  - There is a subfolder for each rollup with a csv file for each rollup instance.
    - For example, for a rollup by `host`, you might have csv files for `testhost1`, `testhost2`, and `testhost3`, in the `host` subfolder.
  - There is a subfolder for each subsystem we have a command device for, and within that there is a subfolder for each type of command device resource type for which we collect data.
    - E.g., a csv file for `00:00` in the `LDEV` folder in the `410034` subfolder.

Test step "success" vs. "failure"

HITACHI
Inspire the Next

- A test step may run for a fixed number of "`warmup_seconds`" and "`measure_seconds`", in which case the test step is always a "success".
- However, if "`measure=on`" is specified on the `[Go]` statement, `warmup_seconds` and `measure_seconds` specify minimum times. After that, ivy keeps running more subintervals until it has "seen enough" to make a statistically valid measurement to the required accuracy (success), or until `timeout_seconds` have been reached (failure).
- If the test step is a "success", a summary csv result line is written describing the rollup over the measurement subintervals, otherwise an error message line is written to the summary csv files.
  - Note: Even if a statistically valid measurement was observed, the test step may still be marked a failure if other test configuration validity criteria are not met.

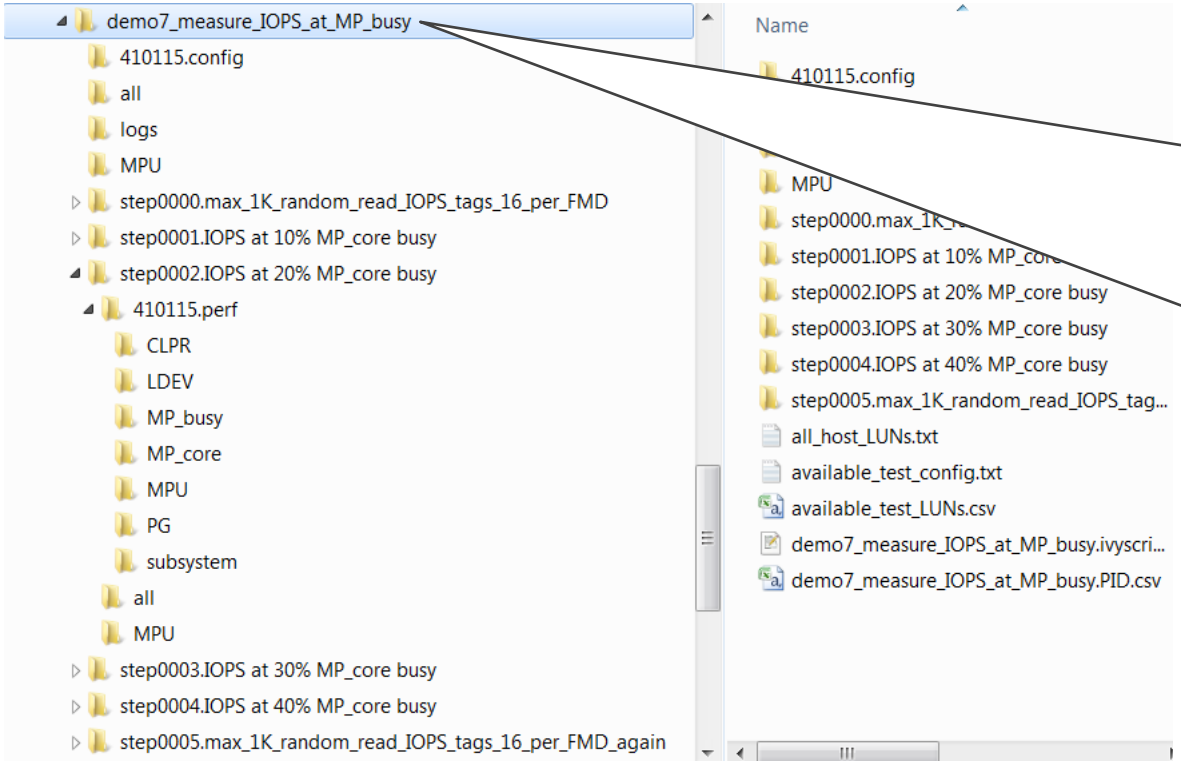
Success = warmup, measure, cooldown



- For each test step, there is one "measurement" csv line in the "summary" csv files.
- For each rollup, there is a "summary" subfolder in the root test output folder containing a csv file for each rollup instance.
  - The equivalent of the summary.html file in vdbench is the "`all=all`" csv file in the "`all`" rollup subfolder.
    - There is always an "`all`" rollup that has one instance "`all`" covering all workload threads.
  - With a command device, there are no measurement summary csv files for RMLIB API data like there are for by-subinterval RMLIB API data within a test step subfolder, but some RMLIB API data is filtered by rollup instance, and there are filtered RMLIB API columns within workload csv files.

Now we are ready to look at ivy output

ivy output folder structure

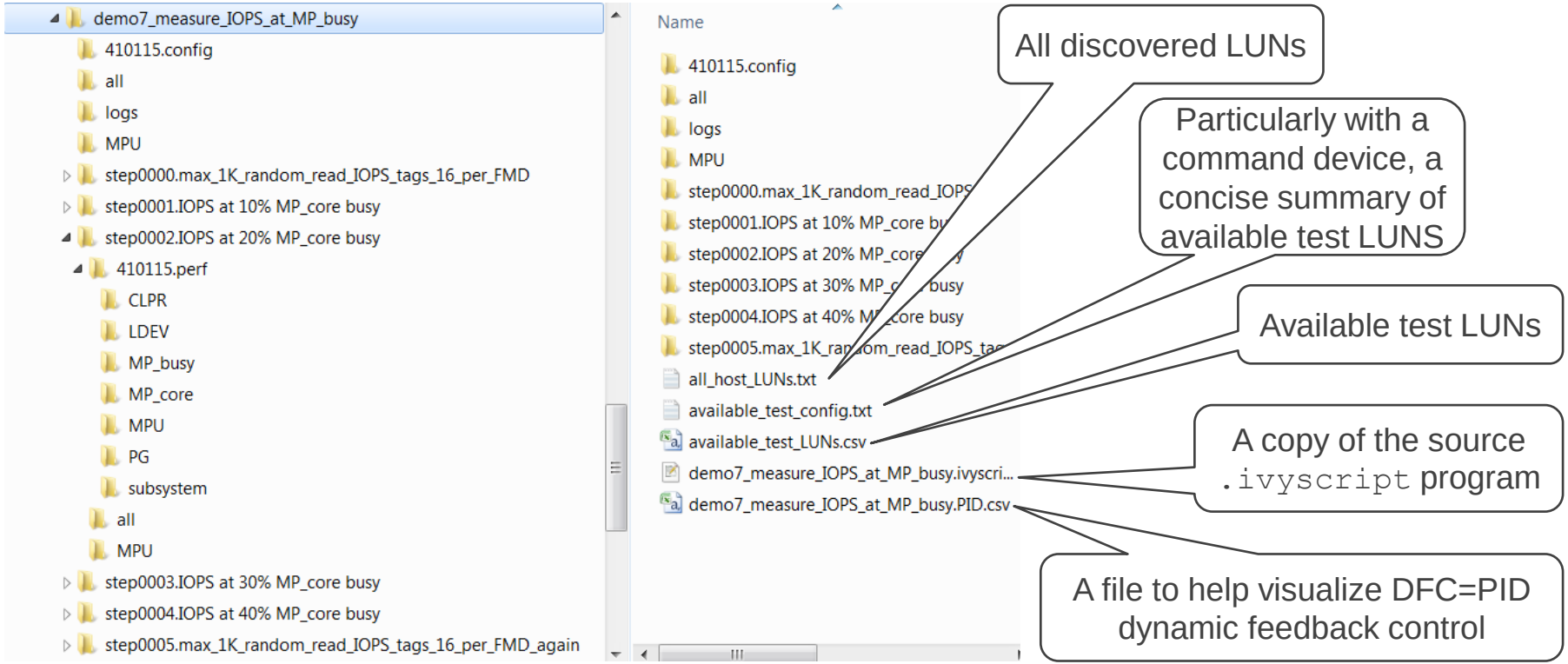


The test output folder name is the ivyscript filename with "`.ivyscript`" removed.

This is also called the "test name", and can be retrieved using the `testName()` ivyscript library function.

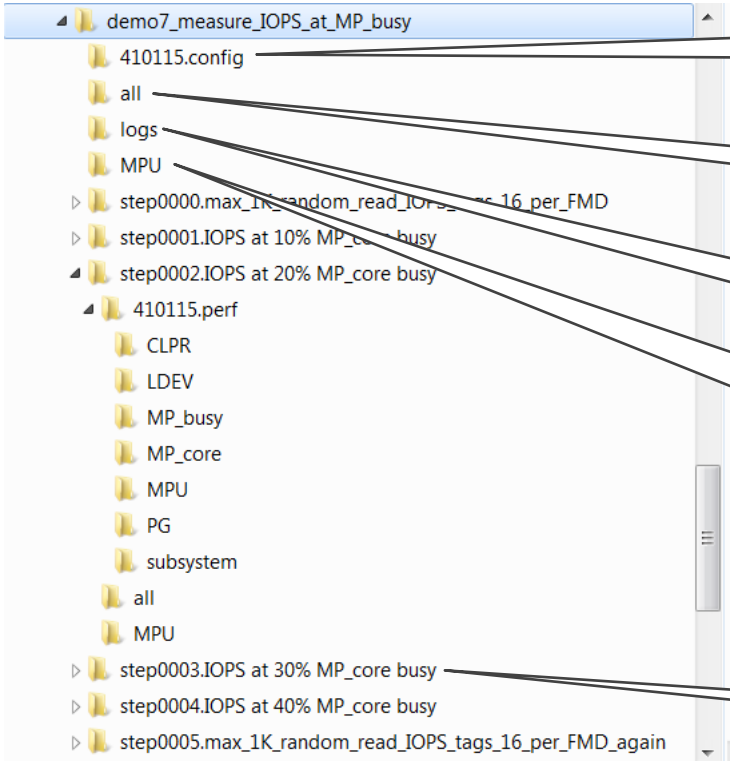
This test folder is a subfolder of the `[OutputFolderRoot]` folder, which defaults to `/scripts/ivy/ivyoutput`.

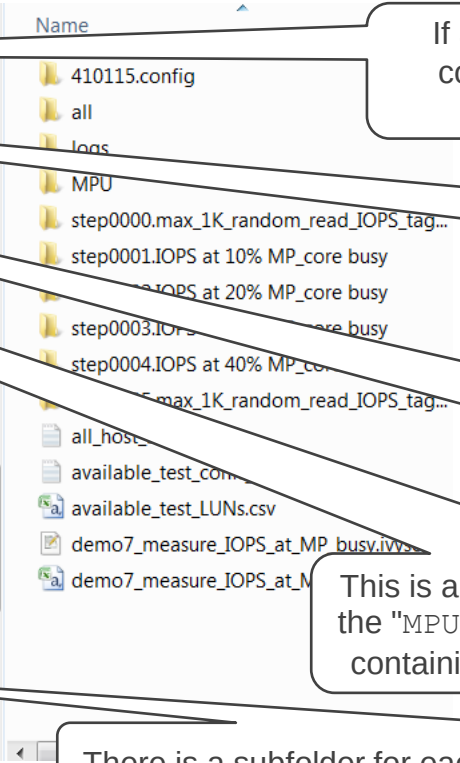
Files in the test output folder itself



Subfolders of the ivy test output folder

HITACHI
Inspire the Next





If a command device is discovered, a folder containing detailed set of command device configuration csv files is put here

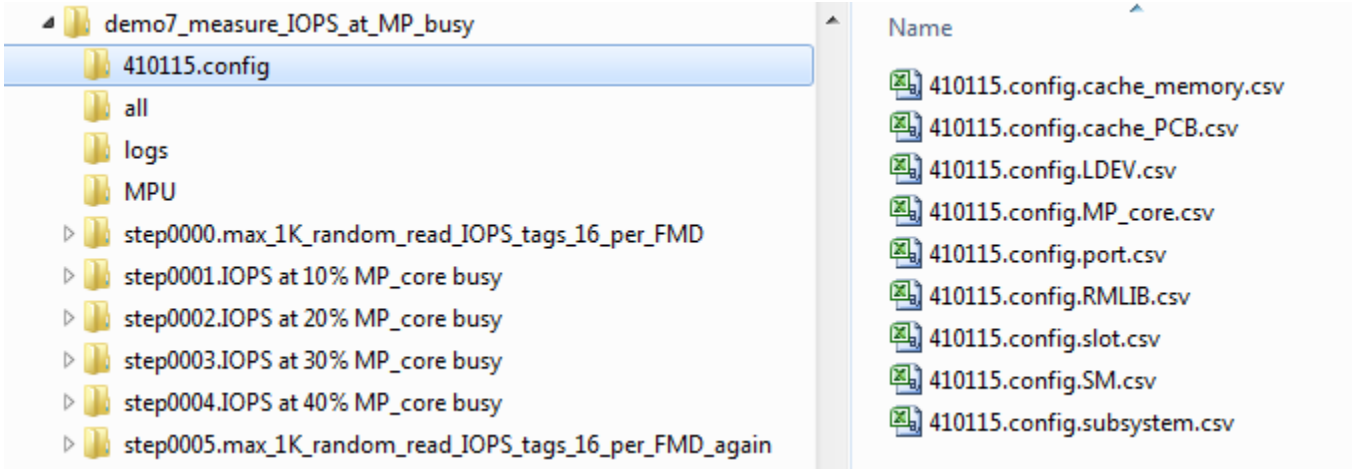
This is a measurement summary csv folder for the "`all`" rollup with one summary line for each test step.

Each ivy process thread on both the master host and the test hosts has its own log file. Use the `-log` command line option to turn on logging of routine events.

This is a measurement summary csv folder for the "`MPU`" rollup with one csv file for each MPU containing a summary line for each test step.

There is a subfolder for each test step containing by-subinterval csv files

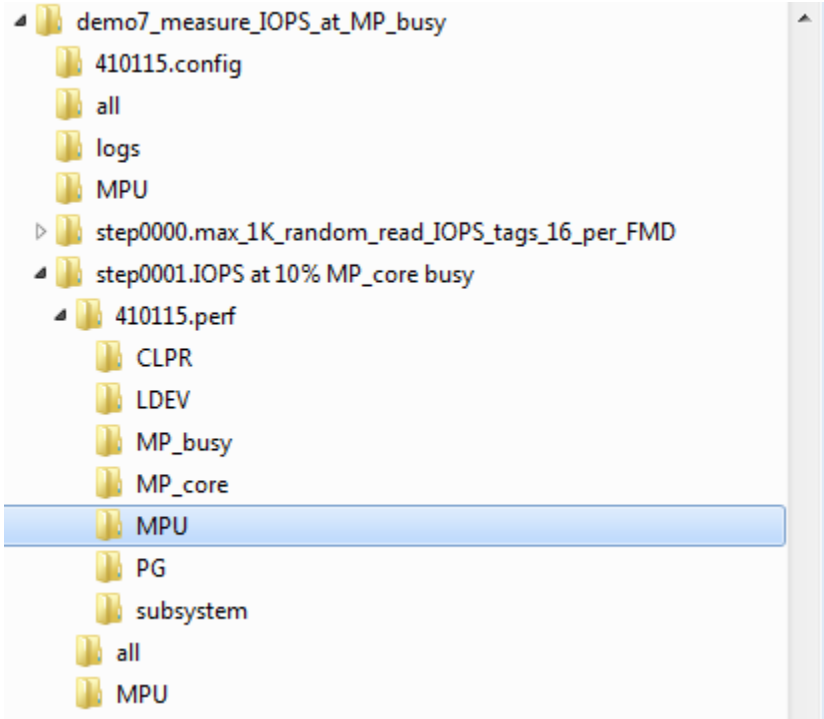
Ivy records the subsystem configuration

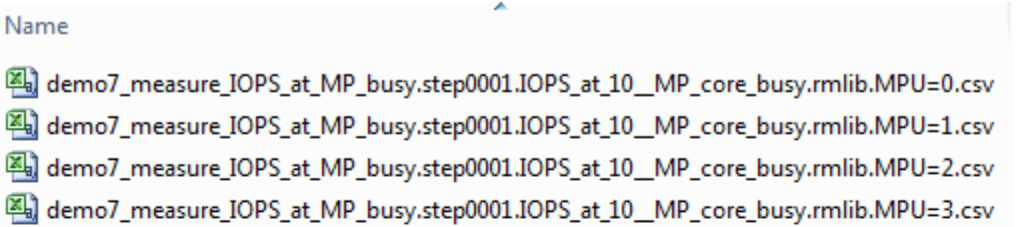


- … if there is a command device present

Ivy records subsystem performance data

HITACHI
Inspire the Next

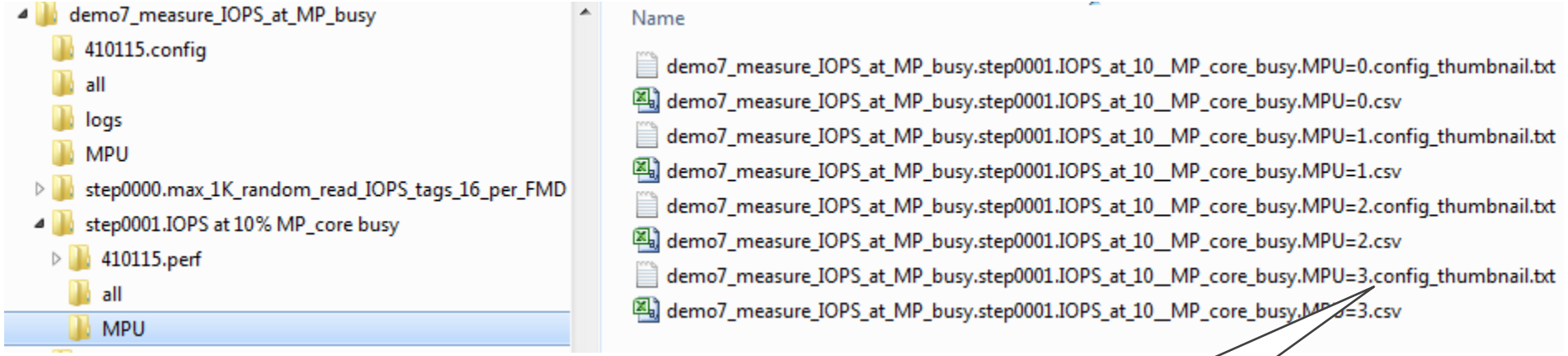




- With a command device, data for the entire subsystem is recorded synchronized with test step subintervals.

Config thumbnails for each rollup instance

HITACHI
Inspire the Next



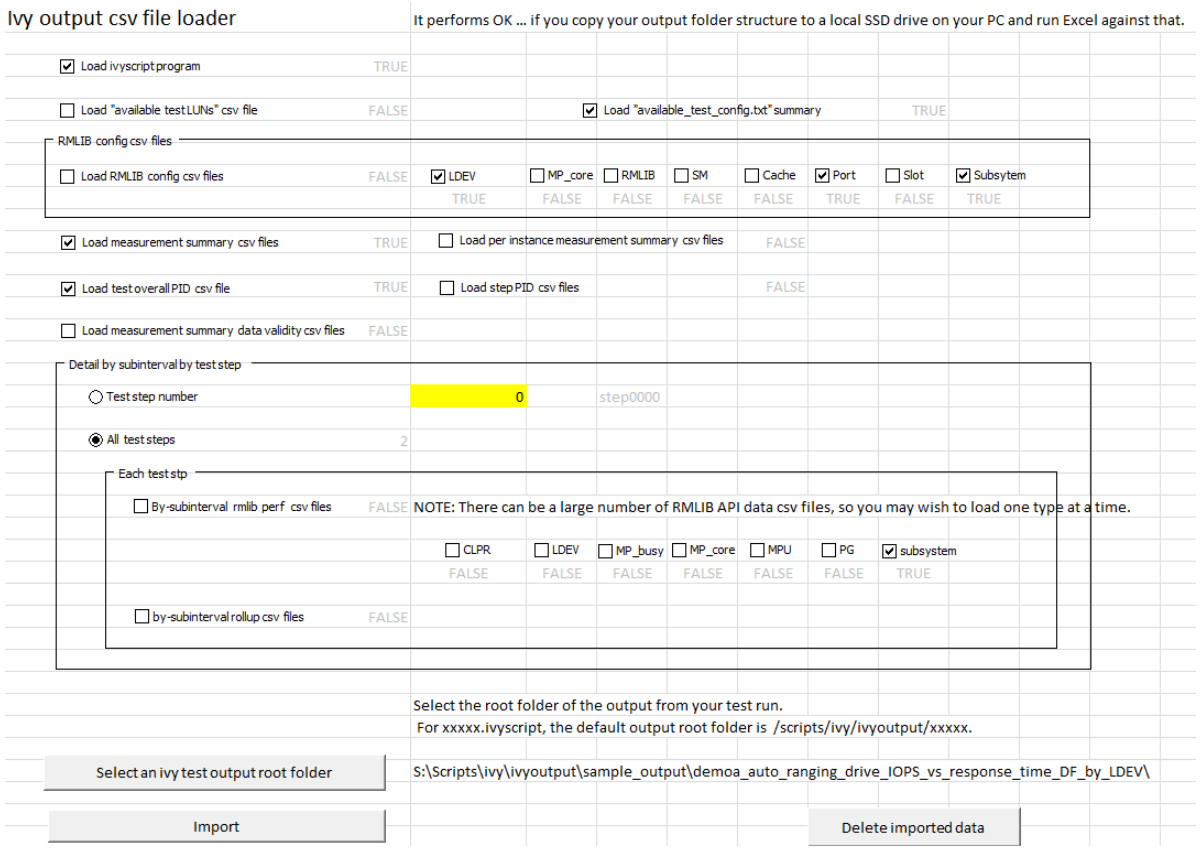

```
Host-mapped LDEVs by subsystem and drive type or pool ID:
HM800 = 410115, NFH1B-P3R2SS, LDEVs = 00:01 (1)

LDEVs by test host:
172.17.62.15, HM800 = 410115, LDEVs = 00:01 (1)

LDEVs by subsystem port:
HM800 = 410115, port = 3C, LDEVs = 00:01 (1)

Drive counts:
HM800 = 410115, NFH1B-P3R2SS, RAID-5 - 7+1, 8 drives in 1 PG   1-1
HM800 = 410115, NFH1B-P3R2SS total 8 drives.
HM800 = 410115 total 8 drives.
```

(Drive counts only available with a command device)

There's a tool to automatically load ivy output csv files into an Excel spreadsheet

load_ivy_output

load_ivy_output.xlsm



Makes it easy to look at the data you want to see

Loads both .txt as well as .csv files.

Use this to explore the ivy demo output.

Questions and Discussion

Thank You

HITACHI

Inspire the Next