# ivy developers’ to-do items

2018-08-23

Not in any particular order

* Run source code static analysis tool, and fix things it shows.
* Ability to run ivy to extend the results of a previous ivy run, adding steps to the end of what’s already there.
  + To recover from an error without re-running everything.
* Review all error messages that can be generated in ivy, and improve meaning to end users, and to suggest corrective actions.
  + Maybe add a way that a test could be marked “invalid” if the interlock latencies are excessive.
* Switch from CodeBlocks build process to using “cmake” or “autotools”.
* Remove iosequencer templates.
  + Now not needed, remove to reduce maintenance burden.
  + “edit rollup” can now be used instead to set a workload parameter value across a group of workloads.
* With a command device, check after each test step for a failed component and if one is found, mark the test step as invalid and terminate.
  + Currently only testing for failed components upon initial fireup.
* Redesign the “measure” feature to do the math on each individual I/O across a series of subintervals, instead of doing the math on the average over each subinterval.
  + Would this be more useful? Would this recognize a valid measurement faster?
* Integrate the SVP configuration csv file set collection into ivy.
  + Was previously implemented in PERL for catscan + TMEA.
  + Would give us additional configuration items to deal with even with a command device.
  + Enable functions requiring subsystem configuration data (over and above the info from SCSI Inquiry) even without a command device, e.g. selecting on drive type.
* Possibly convert ivy C++ engine to use JSON for a) workload parameters, b) go parameters, c) edit rollup parameters.
  + The REST API layer does some of this, receiving JSON from the user, and then translating to the “parameter1 = value1, parameter2 = value2, …” format used in the ivy C++ engine.
  + This would reduce code size, eliminate a couple of old manual parsers.
  + This would enable more complex workload parameter structures.
* Option to write .xlsx instead of .csv.
  + Currently ivy uses a hack, showing values like =”1-1” instead of just 1-1, because Excel would interpret 1-1 as a date, not a character string name of a PG.
* Test LUN\_discovery and ivy in HDLM or other multipath environments.
* Test LUN\_discovery in software stack LVM environments.
* Test LUN\_discovery and ivy in VM environments.
  + Halo lab runs in a VM.
  + VMs are discouraged for actual performance measurement due to possible overhead / latencies impacting accuracy of measurement.
* Test with “top” to see if ivy is accurately reporting test host CPU % busy.
* Explore if users might need to use a different “focus metric” for each of “measure” and “DFC=PID” when both are used together.
* Known ivyscript interpreter issues:
  + -t tracing was broken at some point, fix.
  + Recreate and fix bug that appeared once when declaring a string variable inside a loop body. (Has not reoccurred, but this might just be that users haven’t written programs that encounter the issue.)
* New iosequencer and/or DFC that will drive data patterns to:
  + Find subsystem cache size automatically.
  + Find cache segment size, slot size, and size of cache directory in slots.
  + Measure drive cache buffer size.
* Improve ivy\_cmddev by tagging individual RMLIB API call timestamps instead of timestamping a group of ivy\_cmddev calls.
* In LDEV config data, path type for a (default) host group shows as iSCSI instead of Fibre Channel. Investigate.
* Add Export Tool post-processing step for use in environments without a command device.