

HPC Software Image Test

with BATS/ TAP

Gerard Kennedy¹, Ahmed Shamsul Arefin², Steve McMahon³

¹ Research School of Engineering, ANU, Canberra, Australia, ² Scientific Computing, IM&T, CSIRO Canberra, Australia, ³ IM&T, DST Group Canberra, Australia

CSIRO INFORMATION MANAGEMENT & TECHNOLOGY
www.csiro.au



We present a Software Image Test (SIT) method that can test the software image of a node or nodes in a HPC cluster system. It comprises of a collection of BATS tests that run in an automated SLURM job where the outcomes are sent to the executing user via email. The results help to decide if the software image is ready for rolling on the production cluster.

Tools utilised

- BATS
- nVIDIA Validation Suit
- Intel Cluster Checker
- Intel MPI Benchmark

What is BATS?

BATS (Bash Automated Testing System) is a TAP (Test Anything Protocol) compliant testing framework for Bash. It provides a simple way to verify the functionality of the executing programs. The test uses BATS files, which are essentially Bash scripts with special syntax for defining the test cases. If every command in a test case exits with a 0 status code (success) the test is considered as passed.

addition.bats:

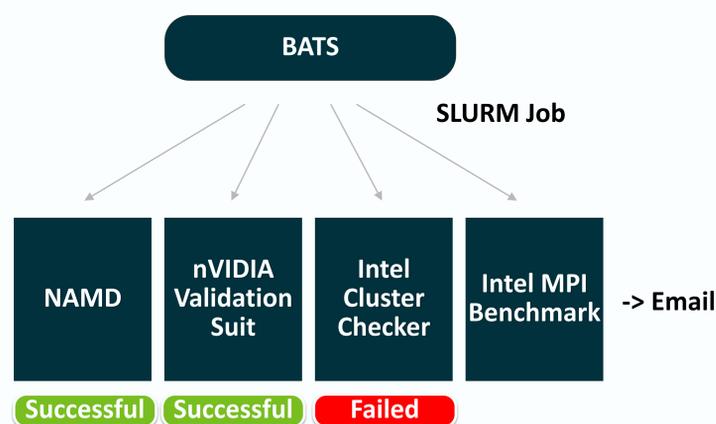
```
#!/usr/bin/env bats
@test "addition using bc" {
  result="$(echo 2+2 | bc)"
  [ "$result" -eq 4 ]
}

$ bats addition.bats ✓ addition using bc1
test, 0 failures
```

In order to execute the SIT script, user must provide a valid set of input arguments. The possible input arguments are:

- **Partition:** The SIT script runs as a batch job, therefore the user needs define the partition in which the node or the set of nodes are located. If the nodes that we wish to test are spread across multiple partitions, need to enter the partitions as a comma separated list.
- **Node(s):** The user can input as many nodes as they wish.

```
# To use with multiple nodes, on a
partition named: "gpu"
./softwareimagetest.sh gpu g124,g125
```



The SIT sends an email to the user and hence users do not need to wait on the console. Based on the results, the user will further tune the software image as required.

Dear Tester,

The test suite FAILED for batch job 1177418.

The test suite ran on node(s) g124,g125 on the galaxy partition.

The test(s) that were scheduled to run were (in order):
apoal.bats nvidia.bats intel.bats benchmark.bats.

The test that failed was: intel.bats.

Time taken for each test in order:

test	time
apoal.bats	0:0:5 (h:m:s)
nvidia.bats	0:0:9 (h:m:s)
intel.bats	0:0:1 (h:m:s)

The script as demonstrated above is simple, but robust enough to accommodate as many factors we wish to test. Our future plan includes to create a GUI, possibly web version where user add/remove tests and get outcomes visually, also use the nVIDIA's DCGM tool which has recently replaced the validation suit.

REFERENCES

1. Stephenson, S., "BATS", <https://github.com/sstephenson/bats>
2. Test Anything Protocol <http://testanything.org/>
3. Nvidia Validation Suit <http://docs.nvidia.com/deploy/nvns-user-guide/index.html>
4. Intel cluster checker <https://software.intel.com/en-us/intel-cluster-checker>
5. Intel MPI Benchmark <https://software.intel.com/en-us/articles/intel-mpi-benchmarks>
6. NAMD <https://www.ks.uiuc.edu/Research/namd/>