Google

Google Hotword Stress Test OpenHST

March Update | March 20th, 2020

jschung@, wonil@

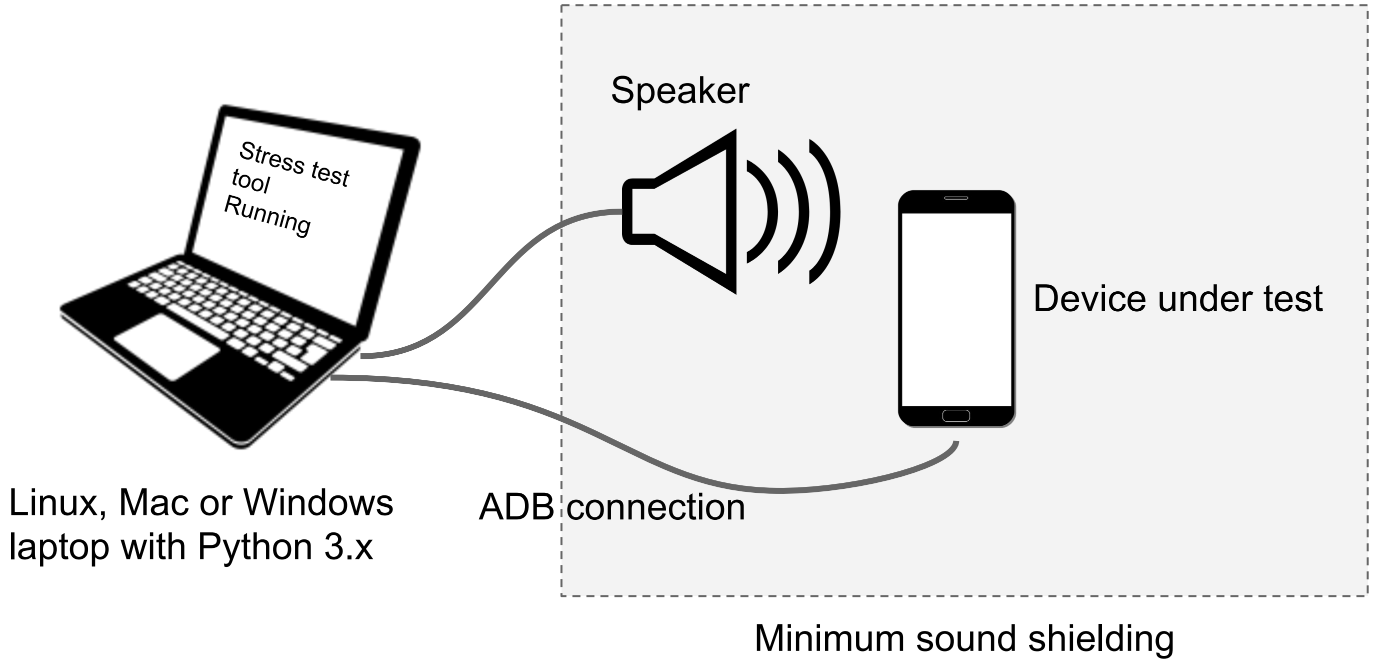
Contents

Objectives

- Check stability & quality of DSP hotword integration
    - By checking 3,000 screen off test case with 95%+ success ratio
- Reduce manual testing
    - Manual testing is painful!
    - Wrong test method e.g. non English native speaker for en-US locale
- Better debugging
    - Stress tool enables detailed logging of DSP hotword. Logging result in output folder can be helpful for debugging

Setup

Environment Setup Example



Environment Setup

- Linux, Mac or Windows machine with speakers
- Test device (phone / tablet)
  - USB debugging environment
  - DSP hotword capable devices
  - Network connection
  - There should be “NONE” lock screen setting
- Noise shielded test room / box
  - Expensive shield machine is not required
  - Test in a quiet room and just make sure whether the handset can recognize the hotword

Installation

- GNU make (3.8.1)
- Python 3.x
  - Verified with Python 3.7.5 on Linux and Windows machines
  - Verified with Python 3.6.4 on Macbook
- Dependency modules

Linux, Mac	Windows
• make start • make proto-compile • source env/bin/activate • ./start_venv.sh	• make start • make proto-compile • .\env\Scripts\activate • .\start_venv.bat

DSP Hotword Capable device

- APK / DSP lib Integration
  - Refer to Hotword integration guide
  - Get the latest APKs / Lib from GMS release folder
  - Integrate “OK Google” and “Hey Google” enrollment APKs
  - Integrate appropriate DSP library per each chipset
- Whitelisting
  - Refer to Hotword Whitelisting & Testing guide
  - In order to test hotword, one of followings should be prepared
    - Model name (in browser UA string) whitelisting, or
    - Whitelisted test account

Run stress test

Test Cases

Test Name	Description
enroll	Plays “OK Google” repeatedly to enroll
dsp_trigger_and_screen_off	Plays “OK Google” on screen off status to launch Google assistant via DSP hotword
dsp_trigger_on_homescreen	Plays “OK Google” on home screen to launch Google Assistant via DSP hotword.
dsp_trigger_sw_rejection	Plays “OK Google” with enrolled voice and different speaker’s voice.

Run Stress Test

- Options

--test_name	mandatory	Specify test name. Predict test name from stress_test.**test_name**.ascii_proto file
--num_iterations	optional	By default, test runs infinitely if there is no issue on testing
--output_root	optional	By default, output folder will be created on your work directory

- Example
  - **python stress_test.py --test_name enroll --num_iterations 10 --output_root test_out**
    - enrollment test with 10 iterations, create output folder under “test_out”
    - Output folder name will be automatically created based on timestamp
    - Outcome files in output folder: stress_test.log / xxxx_logcat.txt / xxxx_kmsg.txt

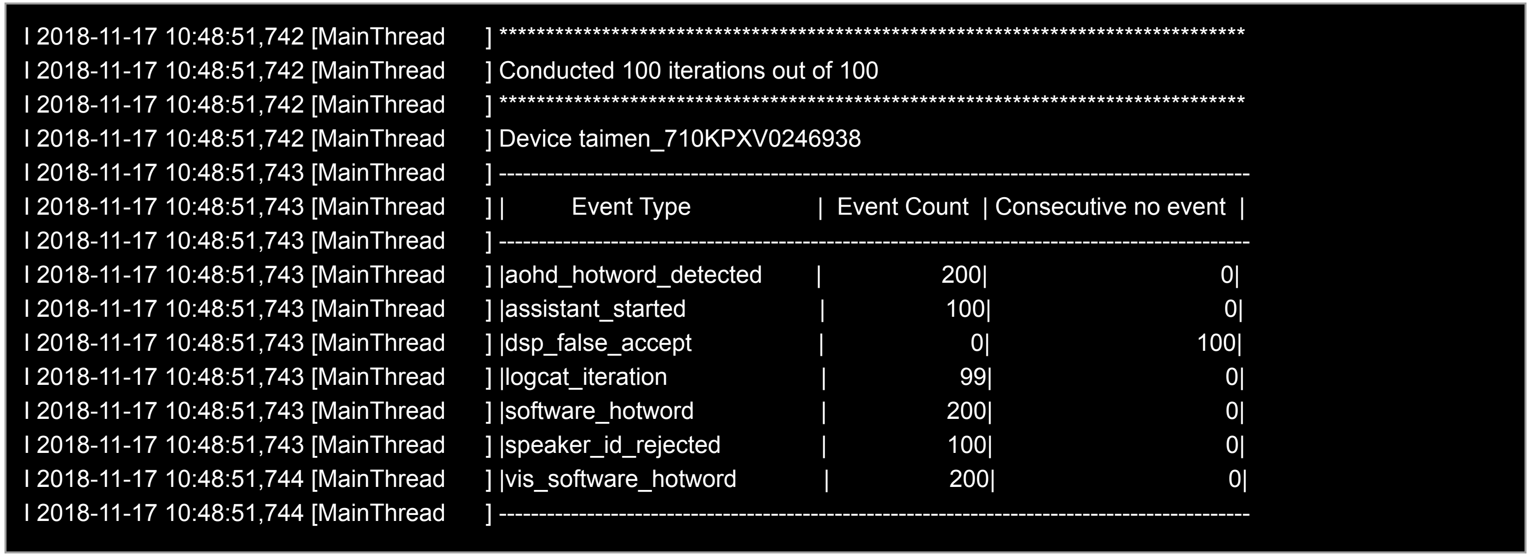
Test Result

Major Events

Event Name	Description
aohd_hotword_detected	Hotword Detection on DSP (Always On Hotword Detection)
software_hotword	Software Hotword detection
vis_software_hotword	AGSA Voice Interaction Service
assistant_started	Launch Google Assistant
speaker_id_rejected	Rejected by the speaker ID mismatch

Test Result - stress_test.log

- main log example for “dsp_trigger_sw_rejection” test case



```
I 2018-11-17 10:48:51,742 [MainThread     ] ********************************************************************************
I 2018-11-17 10:48:51,742 [MainThread     ] Conducted 100 iterations out of 100
I 2018-11-17 10:48:51,742 [MainThread     ] ********************************************************************************
I 2018-11-17 10:48:51,742 [MainThread     ] Device taimen_710KPXV0246938
I 2018-11-17 10:48:51,743 [MainThread     ] ------------------------------------------------------------------------------------------
I 2018-11-17 10:48:51,743 [MainThread     ] |         Event Type                  |  Event Count  | Consecutive no event  |
I 2018-11-17 10:48:51,743 [MainThread     ] ------------------------------------------------------------------------------------------
I 2018-11-17 10:48:51,743 [MainThread     ] |aohd_hotword_detected       |                200|                             0|
I 2018-11-17 10:48:51,743 [MainThread     ] |assistant_started                 |                100|                             0|
I 2018-11-17 10:48:51,743 [MainThread     ] |dsp_false_accept                |                    0|                         100|
I 2018-11-17 10:48:51,743 [MainThread     ] |logcat_iteration                    |                  99|                             0|
I 2018-11-17 10:48:51,743 [MainThread     ] |software_hotword                 |                200|                             0|
I 2018-11-17 10:48:51,743 [MainThread     ] |speaker_id_rejected             |                100|                             0|
I 2018-11-17 10:48:51,744 [MainThread     ] |vis_software_hotword            |                200|                             0|
I 2018-11-17 10:48:51,744 [MainThread     ] ------------------------------------------------------------------------------------------
```

Develop test cases

Customize Stress Test Cases

- Write or modify stress_test.XYZ.ascii_proto

Description	Test starts with the description of the test description: "Test description here."
Pre Steps	Run these steps before running stress test. Pre-steps are annotated with setup_command. setup_command : "shell input keyevent 26" # Pressing the lock button

Test Steps	Actual steps for the stress test. We can specify delay before and after the step.
	step {
	  delay_before : 5
	  audio_file : "speech/micro/api/testdata/okgoogle-16k.wav"
	}
	step {
	  command: "shell input keyevent 3"
	  delay_after: 2
	}

Customize Stress Test Cases

Termination Condition	Last step of the test is to write termination conditions. When these conditions are met, stress test will fail. We can also specify whether to capture bug report or not.
	
	`event {`
	`  name: "test failed"`
	`  condition: "logcat_iteration * 5 != assistant_started"`
	`            " or logcat_iteration * 6 != hal_dsp_hotword"`
	`            " or logcat_iteration * 6 != vis_software_hotword"`
	`            " or logcat_iteration != speaker_id_rejected"`
	`  action: "BUGREPORT"`
	`  action: "NOTIFY"`
	`  action: "REMOVE_DEVICE"`
	`}`

Define new event

- Specify common event in device_config.common.ascii_proto

software_hotword	event {
	  source: "LOGCAT"
	  name: "software_hotword"
	  regex: "MicroDetectionWorker: #onHotwordDetected"
	}

aissistant_started	event {
	  source: "LOGCAT"
	  name: "assistant_started"
	  regex: "ActivityManager: START.*opa.OpaActivity"
	}

Define expected result

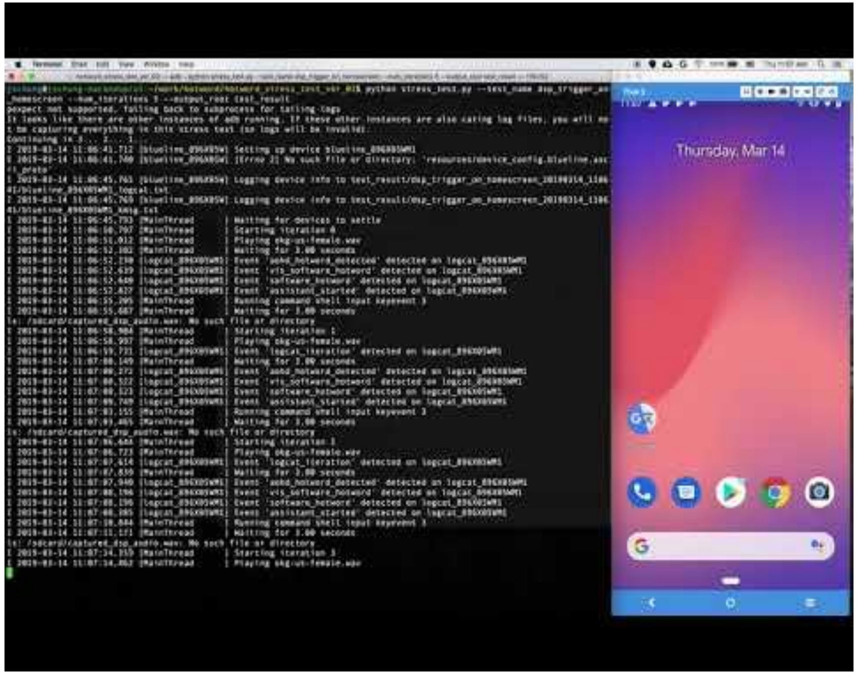
- Specify expected event counts for 1 iteration

software_hotword	Stress Test Tool will check whether all expected events were invoked per each iteration. If any event was missed, will print error message with iteration count.
	
	`expected_result {`
	`  aohd_hotword_detected : 2`
	`  assistant_started  : 1`
	`  dsp_false_accept  : 0`
	`  logcat_iteration  : 1`
	`  software_hotword  : 2`
	`  speaker_id_rejected  : 1`
	`  vis_software_hotword  : 2`
	`}`

Demo

Define expected result

- Run “dsp_trigger_on_homescreen” test
    - Python stress_test.py --test_name dsp_trigger_on_homescreen --num_iterations 5



```
[illegible]:~/work/hotword/hotword_stress_test_ver_03$ python stress_test.py --test_name dsp_trigger_on
_homescreen --num_iterations 5 --output_root test_result
pexpect not supported, falling back to subprocess for tailing logs
It looks like there are other instances of adb running. If these other instances are also cating log files, you will no
t be capturing everything in this stress test (so logs will be invalid).
Continuing in 3... 2... 1...
I 2019-03-14 11:06:41.712 [blueline_896X05W] Setting up device blueline_896X05WM1
E 2019-03-14 11:06:41.780 [blueline_896X05W] [Errno 2] No such file or directory: 'resources/device_config.blueline.asc
ii_proto'
I 2019-03-14 11:06:45.761 [blueline_896X05W] Logging device info to test_result/dsp_trigger_on_homescreen_20190314_1106
41/blueline_896X05WM1_logcat.txt
I 2019-03-14 11:06:45.769 [blueline_896X05W] Logging device info to test_result/dsp_trigger_on_homescreen_20190314_1106
41/blueline_896X05WM1_kmsg.txt
I 2019-03-14 11:06:45.793 [MainThread      ] Waiting for devices to settle
I 2019-03-14 11:06:50.797 [MainThread      ] Starting iteration 0
I 2019-03-14 11:06:51.012 [MainThread      ] Playing okg-us-female.wav
I 2019-03-14 11:06:52.102 [MainThread      ] Waiting for 3.00 seconds
I 2019-03-14 11:06:52.230 [logcat_896X05WM1] Event 'aohd_hotword_detected' detected on logcat_896X05WM1
I 2019-03-14 11:06:52.639 [logcat_896X05WM1] Event 'vis_software_hotword' detected on logcat_896X05WM1
I 2019-03-14 11:06:52.640 [logcat_896X05WM1] Event 'software_hotword' detected on logcat_896X05WM1
I 2019-03-14 11:06:52.827 [logcat_896X05WM1] Event 'assistant_started' detected on logcat_896X05WM1
I 2019-03-14 11:06:55.205 [MainThread      ] Running command shell input keyevent 3
I 2019-03-14 11:06:55.407 [MainThread      ] Waiting for 3.00 seconds
ls: /sdcard/captured_dsp_audio.wav: No such file or directory
I 2019-03-14 11:06:58.904 [MainThread      ] Starting iteration 1
I 2019-03-14 11:06:58.907 [MainThread      ] Playing okg-us-female.wav
I 2019-03-14 11:06:59.731 [logcat_896X05WM1] Event 'logcat_iteration' detected on logcat_896X05WM1
I 2019-03-14 11:07:00.149 [MainThread      ] Waiting for 3.00 seconds
I 2019-03-14 11:07:00.272 [logcat_896X05WM1] Event 'aohd_hotword_detected' detected on logcat_896X05WM1
I 2019-03-14 11:07:00.322 [logcat_896X05WM1] Event 'vis_software_hotword' detected on logcat_896X05WM1
I 2019-03-14 11:07:00.323 [logcat_896X05WM1] Event 'software_hotword' detected on logcat_896X05WM1
I 2019-03-14 11:07:00.749 [logcat_896X05WM1] Event 'assistant_started' detected on logcat_896X05WM1
I 2019-03-14 11:07:03.155 [MainThread      ] Running command shell input keyevent 3
I 2019-03-14 11:07:03.465 [MainThread      ] Waiting for 3.00 seconds
ls: /sdcard/captured_dsp_audio.wav: No such file or directory
I 2019-03-14 11:07:06.644 [MainThread      ] Starting iteration 2
I 2019-03-14 11:07:06.723 [MainThread      ] Playing okg-us-female.wav
I 2019-03-14 11:07:07.614 [logcat_896X05WM1] Event 'logcat_iteration' detected on logcat_896X05WM1
I 2019-03-14 11:07:07.839 [MainThread      ] Waiting for 3.00 seconds
I 2019-03-14 11:07:07.940 [logcat_896X05WM1] Event 'aohd_hotword_detected' detected on logcat_896X05WM1
I 2019-03-14 11:07:08.196 [logcat_896X05WM1] Event 'vis_software_hotword' detected on logcat_896X05WM1
I 2019-03-14 11:07:08.196 [logcat_896X05WM1] Event 'software_hotword' detected on logcat_896X05WM1
I 2019-03-14 11:07:08.385 [logcat_896X05WM1] Event 'assistant_started' detected on logcat_896X05WM1
I 2019-03-14 11:07:10.844 [MainThread      ] Running command shell input keyevent 3
I 2019-03-14 11:07:11.171 [MainThread      ] Waiting for 3.00 seconds
ls: /sdcard/captured_dsp_audio.wav: No such file or directory
I 2019-03-14 11:07:14.359 [MainThread      ] Starting iteration 3
I 2019-03-14 11:07:14.862 [MainThread      ] Playing okg-us-female.wav
```