Iteration-2 Report

Symbolic Execution Of Smart Contracts Using Manticore

Advanced Topics in Software Engineering

CSE 6324

GitHub link: https://github.com/PavanKumarChaparla/ADV-TOPICS-IN-SE-CSE-6324

Team 8:

Pavan Kumar Chaparla-1001865675

Keerthi Chittineni-1001967726

Anupama Rani Neerukonda-1002007833

Lokesh Bogam-1001946861

Jai Venkata Sai Sriram Batchu-1001949158

1. Project Plan

Introduction

There are a lot of open issues with the Manticore. The main goal of the project is to improvise the Manticore performance on the smart contracts by working on the internal Manticore API's.

In the iteration 2:

With the same motive, we have looked into the open issues that the Manticore tools currently have, in which we were able to successfully resolve one such issue #2427.

In the same way, we are looking into the issues related to time complexity i.e #2427. That can help in improving the performance of the manticore when tested with updated API, it returned results in quicker time.

Major goals:

Picked up three open issues in manticore and for Iterations.

Iteration 1: Compatibility issues - #2577

Iteration 2: Time complexity issues- #2427

Iteration 3: Code coverage

Project plan Features:

1. Compile and run Solidity Code
   The IDE will allow the user to write the solidity code that they would like to analyze.
2. Analyzing the smart contract with Manticore
   The user can compile the smart contract code using the Manticore API.
3. Generating the results
   The tool, once finished with the analysis, will produce the results into reports and save them in the specified directory.

2. Design and Specifications

Pictorial Representation:

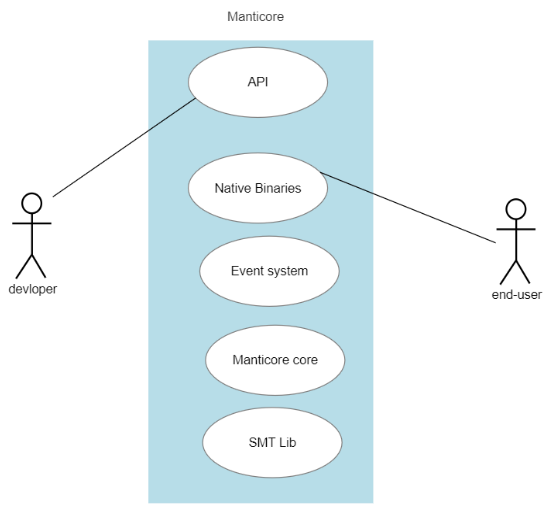


Fig 1: High level architectural view of manticore usage

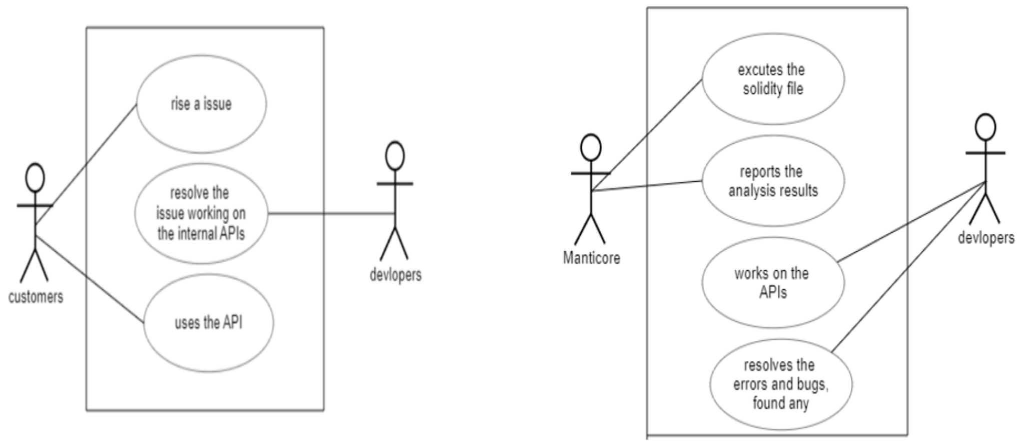

rise a issue
resolve the issue working on the internal APIs
uses the API
customers
devlopers
excutes the solidity file
reports the analysis results
works on the APIs
resolves the errors and bugs, found any
Manticore
devlopers

Fig 2**:** User and developer interacting with the API

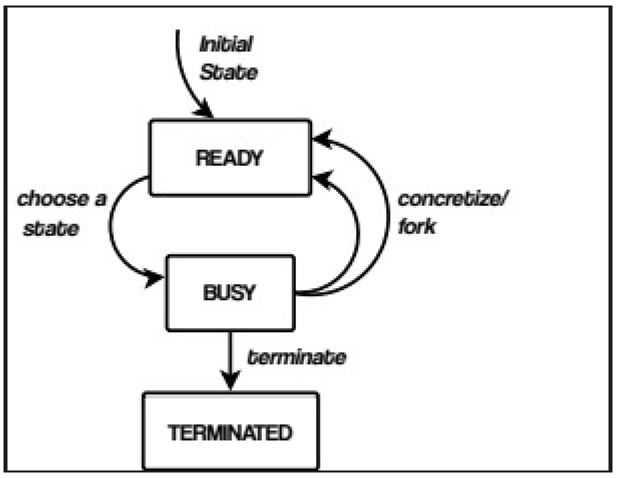


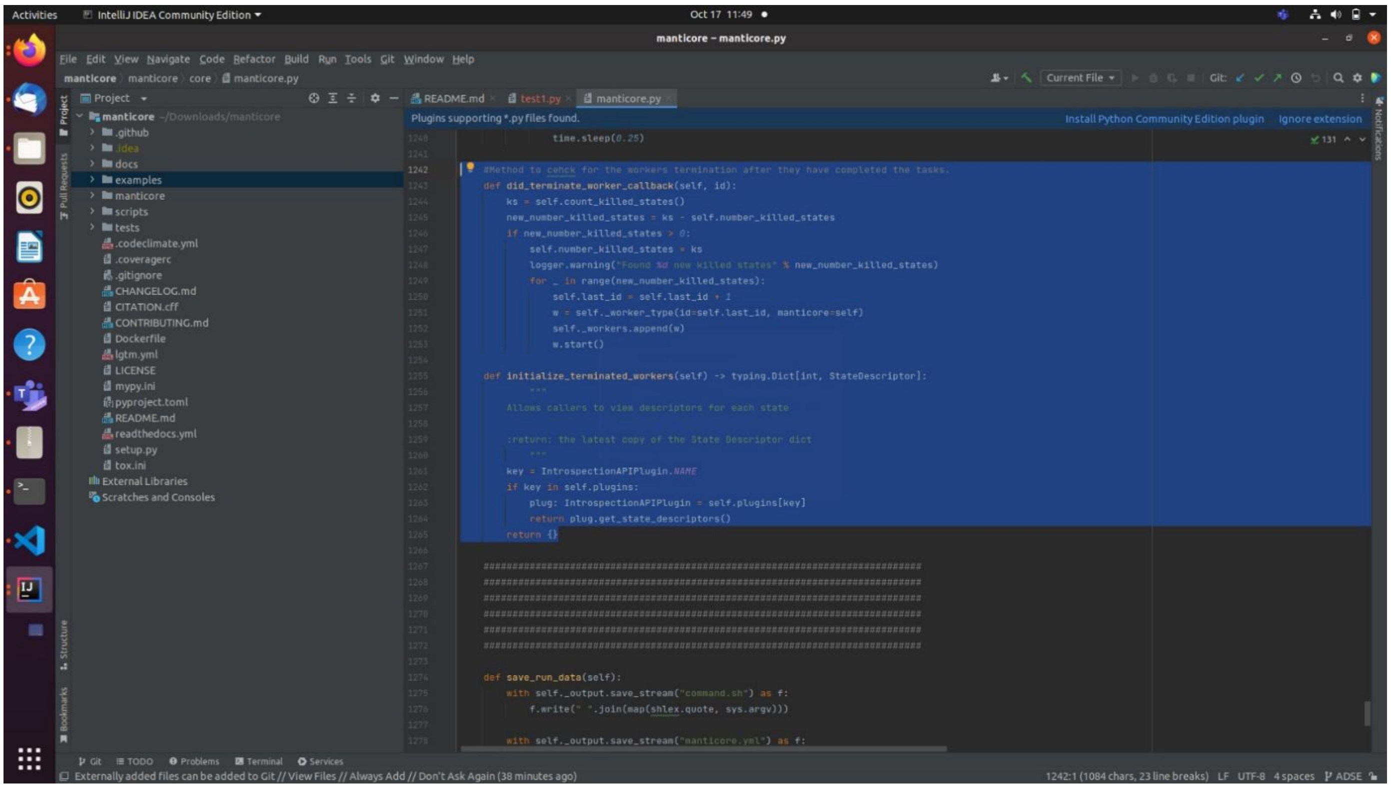
Fig 3: High level architecture of manticore

Issue #2427

We have identified an open issue in the manticore development repository, which is related to time complexity. Because of a state goes from busy to killed, the worker handling that state won't take another. In larger targets, it is likely that some workers will stop running, degrading the overall performance of our tool. Also, in the extreme case of a single worker configuration, Manticore will hang forever (until the user presses ctrl+c or the process is terminated). When the workers are done working with their targets, It was designed in such away that workers will be terminated after the process.

3. Code and Test:

We have made the following changes to the manticore API i.e. the changes are highlighted in the below screen shots.

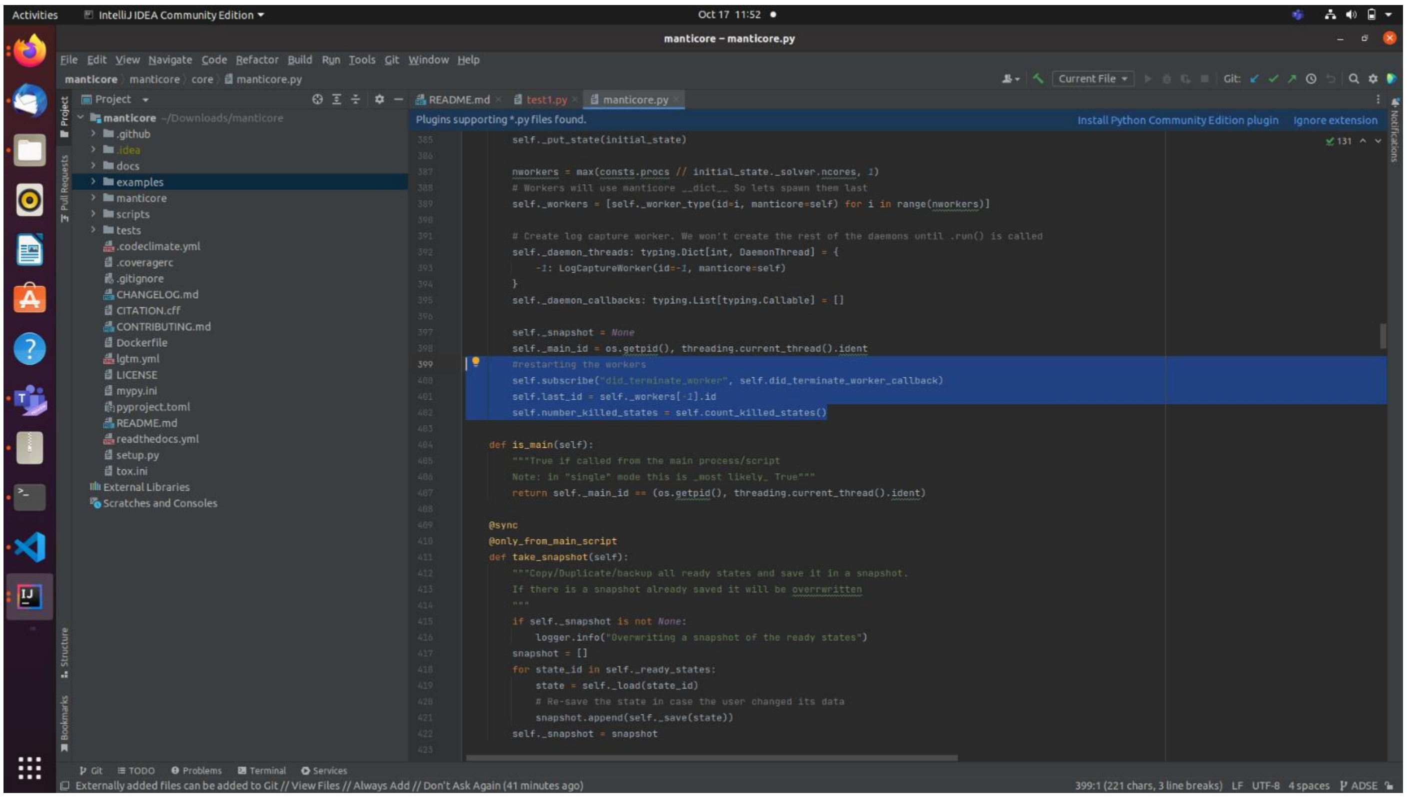


Plugins supporting *.py files found.

```
                time.sleep(0.25)

    #Method to cehck for the workers termination after they have completed the tasks.
    def did_terminate_worker_callback(self, id):
        ks = self.count_killed_states()
        new_number_killed_states = ks - self.number_killed_states
        if new_number_killed_states > 0:
            self.number_killed_states = ks
            logger.warning("Found %d new killed states" % new_number_killed_states)
            for _ in range(new_number_killed_states):
                self.last_id = self.last_id + 1
                w = self._worker_type(id=self.last_id, manticore=self)
                self._workers.append(w)
                w.start()

    def initialize_terminated_workers(self) -> typing.Dict[int, StateDescriptor]:
            """
        Allows callers to view descriptors for each state

        :return: the latest copy of the State Descriptor dict
            """
        key = IntrospectionAPIPlugin.NAME
        if key in self.plugins:
            plug: IntrospectionAPIPlugin = self.plugins[key]
            return plug.get_state_descriptors()
        return {}

    ############################################################################
    ############################################################################
    ############################################################################
    ############################################################################
    ############################################################################
    ############################################################################

    def save_run_data(self):
        with self._output.save_stream("command.sh") as f:
            f.write(" ".join(map(shlex.quote, sys.argv)))

        with self._output.save_stream("manticore.yml") as f:
```

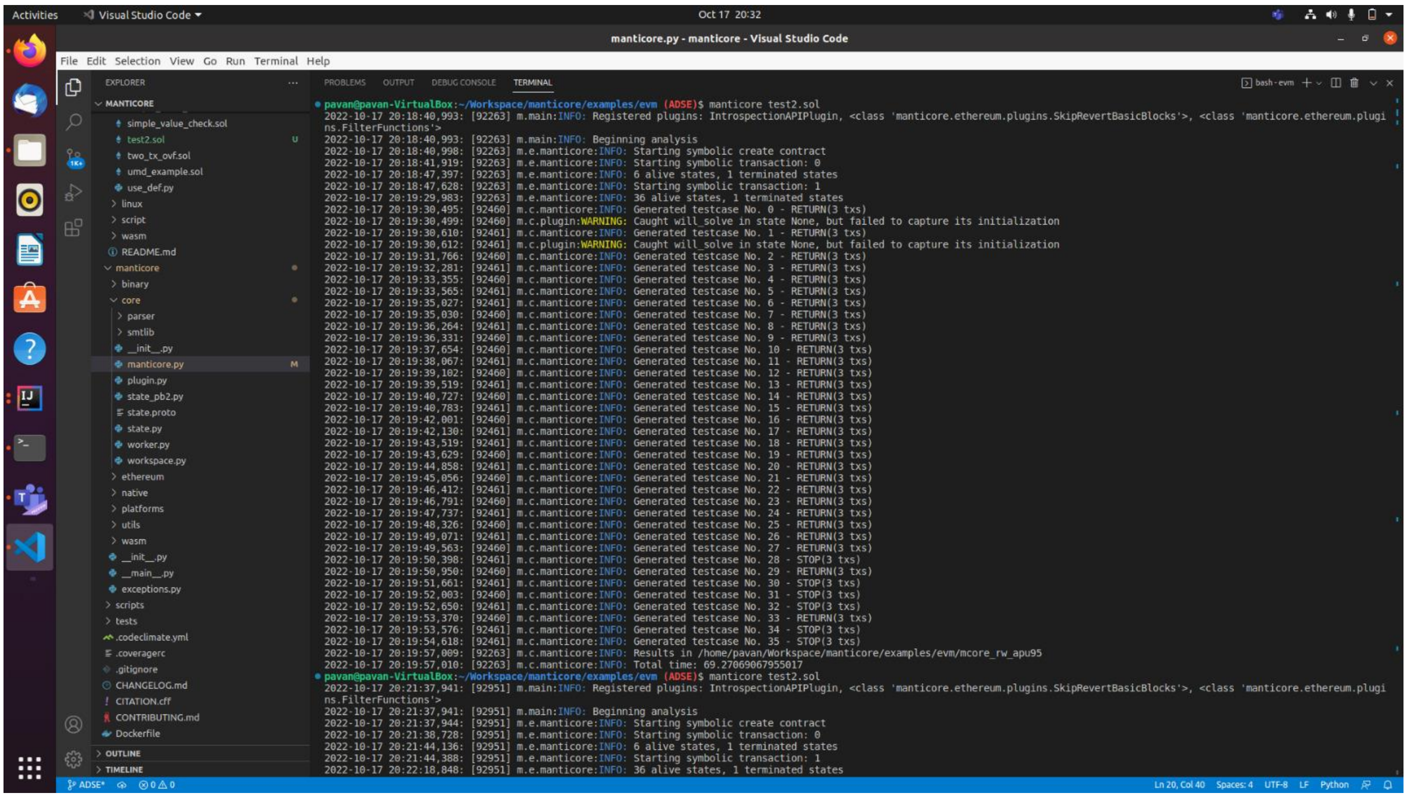
Externally added files can be added to Git // View Files // Always Add // Don't Ask Again (38 minutes ago)

Fig 4: API code



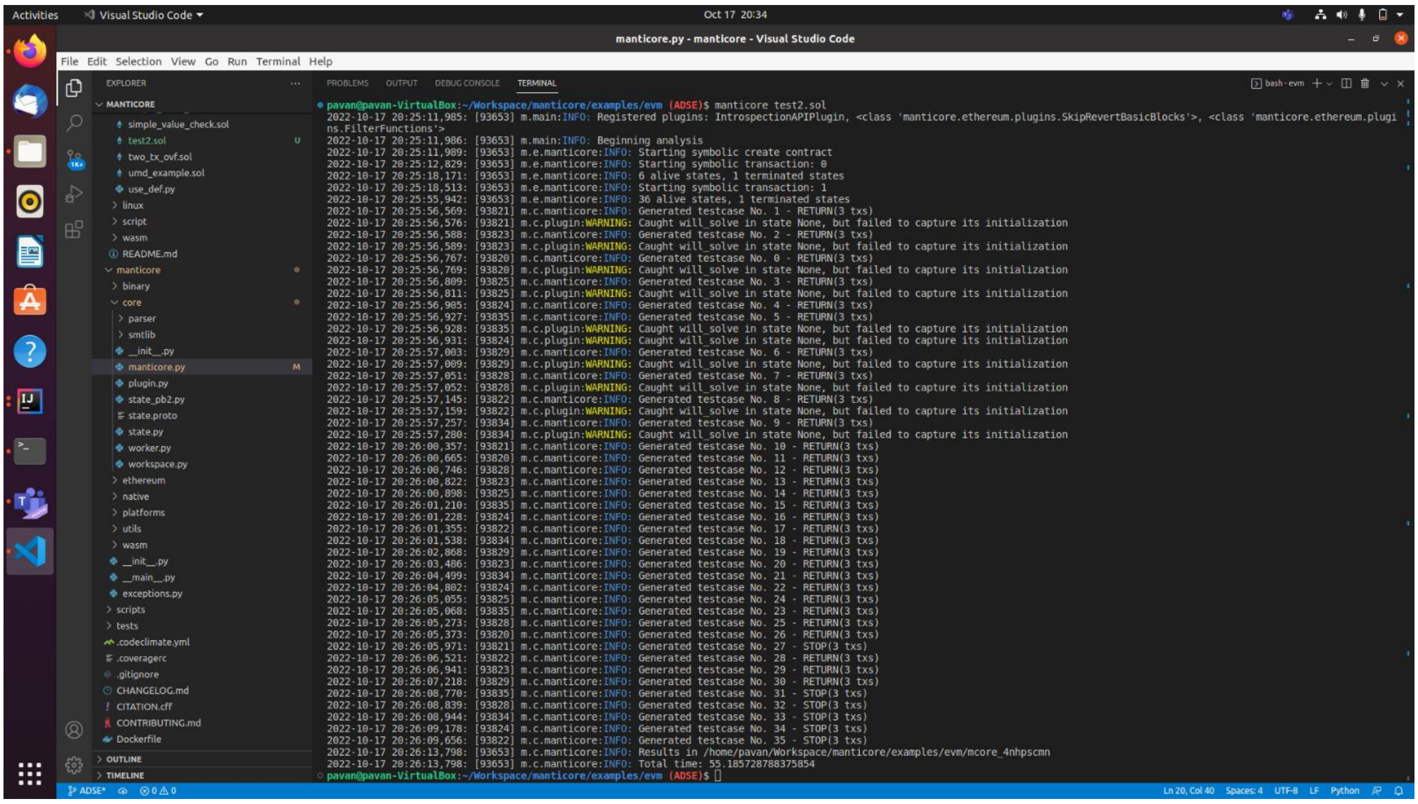
```python
        self._put_state(initial_state)

        nworkers = max(consts.procs // initial_state._solver.ncores, 1)
        # Workers will use manticore __dict__ So lets spawn them last
        self._workers = [self._worker_type(id=i, manticore=self) for i in range(nworkers)]

        # Create log capture worker. We won't create the rest of the daemons until .run() is called
        self._daemon_threads: typing.Dict[int, DaemonThread] = {
            -1: LogCaptureWorker(id=-1, manticore=self)
        }
        self._daemon_callbacks: typing.List[typing.Callable] = []

        self._snapshot = None
        self._main_id = os.getpid(), threading.current_thread().ident
        #restarting the workers
        self.subscribe("did_terminate_worker", self.did_terminate_worker_callback)
        self.last_id = self._workers[-1].id
        self.number_killed_states = self.count_killed_states()

    def is_main(self):
        """True if called from the main process/script
        Note: in "single" mode this is _most likely_ True"""
        return self._main_id == (os.getpid(), threading.current_thread().ident)

    @sync
    @only_from_main_script
    def take_snapshot(self):
        """Copy/Duplicate/backup all ready states and save it in a snapshot.
        If there is a snapshot already saved it will be overrwritten
        """
        if self._snapshot is not None:
            logger.info("Overwriting a snapshot of the ready states")
        snapshot = []
        for state_id in self._ready_states:
            state = self._load(state_id)
            # Re-save the state in case the user changed its data
            snapshot.append(self._save(state))
        self._snapshot = snapshot
```

Fig 5: API Code 2



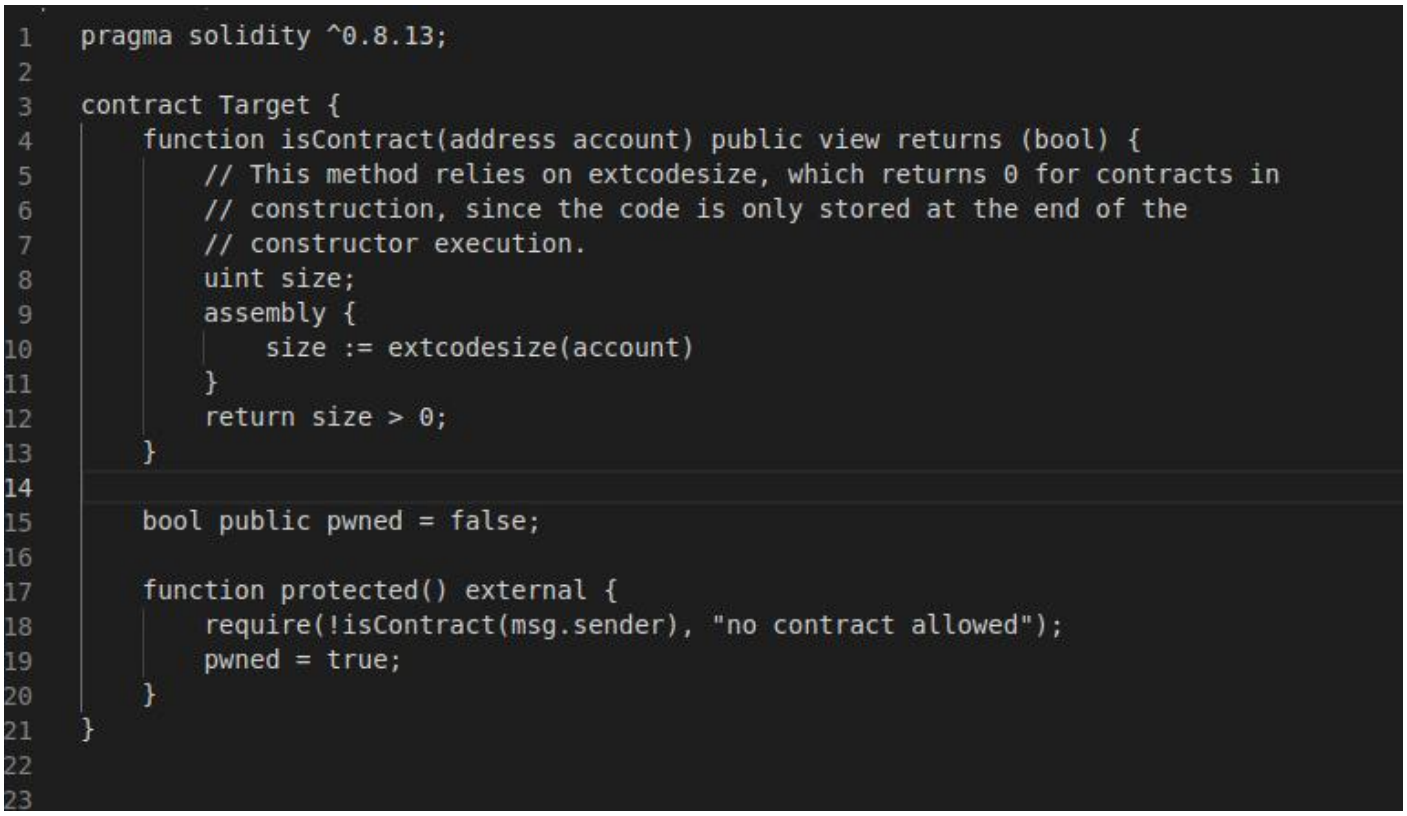
```
pavan@pavan-VirtualBox:~/Workspace/manticore/examples/evm (ADSE)$ manticore test2.sol
2022-10-17 20:18:40,993: [92263] m.main:INFO: Registered plugins: IntrospectionAPIPlugin, <class 'manticore.ethereum.plugins.SkipRevertBasicBlocks'>, <class 'manticore.ethereum.plugi
ns.FilterFunctions'>
2022-10-17 20:18:40,993: [92263] m.main:INFO: Beginning analysis
2022-10-17 20:18:40,998: [92263] m.e.manticore:INFO: Starting symbolic create contract
2022-10-17 20:18:41,919: [92263] m.e.manticore:INFO: Starting symbolic transaction: 0
2022-10-17 20:18:47,397: [92263] m.e.manticore:INFO: 6 alive states, 1 terminated states
2022-10-17 20:18:47,628: [92263] m.e.manticore:INFO: Starting symbolic transaction: 1
2022-10-17 20:19:29,983: [92263] m.e.manticore:INFO: 36 alive states, 1 terminated states
2022-10-17 20:19:30,495: [92460] m.c.manticore:INFO: Generated testcase No. 0 - RETURN(3 txs)
2022-10-17 20:19:30,499: [92460] m.c.plugin:WARNING: Caught will_solve in state None, but failed to capture its initialization
2022-10-17 20:19:30,610: [92461] m.c.manticore:INFO: Generated testcase No. 1 - RETURN(3 txs)
2022-10-17 20:19:30,612: [92461] m.c.plugin:WARNING: Caught will_solve in state None, but failed to capture its initialization
2022-10-17 20:19:31,766: [92460] m.c.manticore:INFO: Generated testcase No. 2 - RETURN(3 txs)
2022-10-17 20:19:32,281: [92461] m.c.manticore:INFO: Generated testcase No. 3 - RETURN(3 txs)
2022-10-17 20:19:33,355: [92460] m.c.manticore:INFO: Generated testcase No. 4 - RETURN(3 txs)
2022-10-17 20:19:33,565: [92461] m.c.manticore:INFO: Generated testcase No. 5 - RETURN(3 txs)
2022-10-17 20:19:35,027: [92461] m.c.manticore:INFO: Generated testcase No. 6 - RETURN(3 txs)
2022-10-17 20:19:35,030: [92460] m.c.manticore:INFO: Generated testcase No. 7 - RETURN(3 txs)
2022-10-17 20:19:36,264: [92461] m.c.manticore:INFO: Generated testcase No. 8 - RETURN(3 txs)
2022-10-17 20:19:36,331: [92460] m.c.manticore:INFO: Generated testcase No. 9 - RETURN(3 txs)
2022-10-17 20:19:37,654: [92460] m.c.manticore:INFO: Generated testcase No. 10 - RETURN(3 txs)
2022-10-17 20:19:38,067: [92461] m.c.manticore:INFO: Generated testcase No. 11 - RETURN(3 txs)
2022-10-17 20:19:39,102: [92460] m.c.manticore:INFO: Generated testcase No. 12 - RETURN(3 txs)
2022-10-17 20:19:39,519: [92461] m.c.manticore:INFO: Generated testcase No. 13 - RETURN(3 txs)
2022-10-17 20:19:40,727: [92460] m.c.manticore:INFO: Generated testcase No. 14 - RETURN(3 txs)
2022-10-17 20:19:40,783: [92461] m.c.manticore:INFO: Generated testcase No. 15 - RETURN(3 txs)
2022-10-17 20:19:42,001: [92460] m.c.manticore:INFO: Generated testcase No. 16 - RETURN(3 txs)
2022-10-17 20:19:42,130: [92461] m.c.manticore:INFO: Generated testcase No. 17 - RETURN(3 txs)
2022-10-17 20:19:43,519: [92461] m.c.manticore:INFO: Generated testcase No. 18 - RETURN(3 txs)
2022-10-17 20:19:43,629: [92460] m.c.manticore:INFO: Generated testcase No. 19 - RETURN(3 txs)
2022-10-17 20:19:44,858: [92461] m.c.manticore:INFO: Generated testcase No. 20 - RETURN(3 txs)
2022-10-17 20:19:45,056: [92460] m.c.manticore:INFO: Generated testcase No. 21 - RETURN(3 txs)
2022-10-17 20:19:46,412: [92461] m.c.manticore:INFO: Generated testcase No. 22 - RETURN(3 txs)
2022-10-17 20:19:46,791: [92460] m.c.manticore:INFO: Generated testcase No. 23 - RETURN(3 txs)
2022-10-17 20:19:47,737: [92461] m.c.manticore:INFO: Generated testcase No. 24 - RETURN(3 txs)
2022-10-17 20:19:48,326: [92460] m.c.manticore:INFO: Generated testcase No. 25 - RETURN(3 txs)
2022-10-17 20:19:49,071: [92461] m.c.manticore:INFO: Generated testcase No. 26 - RETURN(3 txs)
2022-10-17 20:19:49,563: [92460] m.c.manticore:INFO: Generated testcase No. 27 - RETURN(3 txs)
2022-10-17 20:19:50,398: [92461] m.c.manticore:INFO: Generated testcase No. 28 - STOP(3 txs)
2022-10-17 20:19:50,950: [92460] m.c.manticore:INFO: Generated testcase No. 29 - RETURN(3 txs)
2022-10-17 20:19:51,661: [92461] m.c.manticore:INFO: Generated testcase No. 30 - STOP(3 txs)
2022-10-17 20:19:52,003: [92460] m.c.manticore:INFO: Generated testcase No. 31 - STOP(3 txs)
2022-10-17 20:19:52,650: [92461] m.c.manticore:INFO: Generated testcase No. 32 - STOP(3 txs)
2022-10-17 20:19:53,370: [92460] m.c.manticore:INFO: Generated testcase No. 33 - RETURN(3 txs)
2022-10-17 20:19:53,576: [92461] m.c.manticore:INFO: Generated testcase No. 34 - STOP(3 txs)
2022-10-17 20:19:54,618: [92461] m.c.manticore:INFO: Generated testcase No. 35 - STOP(3 txs)
2022-10-17 20:19:57,009: [92263] m.c.manticore:INFO: Results in /home/pavan/Workspace/manticore/examples/evm/mcore_rw_apu95
2022-10-17 20:19:57,010: [92263] m.c.manticore:INFO: Total time: 69.27069067955017
pavan@pavan-VirtualBox:~/Workspace/manticore/examples/evm (ADSE)$ manticore test2.sol
2022-10-17 20:21:37,941: [92951] m.main:INFO: Registered plugins: IntrospectionAPIPlugin, <class 'manticore.ethereum.plugins.SkipRevertBasicBlocks'>, <class 'manticore.ethereum.plugi
ns.FilterFunctions'>
2022-10-17 20:21:37,941: [92951] m.main:INFO: Beginning analysis
2022-10-17 20:21:37,944: [92951] m.e.manticore:INFO: Starting symbolic create contract
2022-10-17 20:21:38,728: [92951] m.e.manticore:INFO: Starting symbolic transaction: 0
2022-10-17 20:21:44,136: [92951] m.e.manticore:INFO: 6 alive states, 1 terminated states
2022-10-17 20:21:44,388: [92951] m.e.manticore:INFO: Starting symbolic transaction: 1
2022-10-17 20:22:18,848: [92951] m.e.manticore:INFO: 36 alive states, 1 terminated states
```

Output 1: Before updating the manticore API where the workers have been terminated



```
pavan@pavan-VirtualBox:~/Workspace/manticore/examples/evm (ADSE)$ manticore test2.sol
2022-10-17 20:25:11,985: [93653] m.main:INFO: Registered plugins: IntrospectionAPIPlugin, <class 'manticore.ethereum.plugins.SkipRevertBasicBlocks'>, <class 'manticore.ethereum.plugi
ns.FilterFunctions'>
2022-10-17 20:25:11,986: [93653] m.main:INFO: Beginning analysis
2022-10-17 20:25:11,989: [93653] m.e.manticore:INFO: Starting symbolic create contract
2022-10-17 20:25:12,829: [93653] m.e.manticore:INFO: Starting symbolic transaction: 0
2022-10-17 20:25:18,171: [93653] m.e.manticore:INFO: 6 alive states, 1 terminated states
2022-10-17 20:25:18,513: [93653] m.e.manticore:INFO: Starting symbolic transaction: 1
2022-10-17 20:25:55,942: [93653] m.e.manticore:INFO: 36 alive states, 1 terminated states
2022-10-17 20:25:56,569: [93821] m.c.manticore:INFO: Generated testcase No. 1 - RETURN(3 txs)
2022-10-17 20:25:56,576: [93821] m.c.plugin:WARNING: Caught will_solve in state None, but failed to capture its initialization
2022-10-17 20:25:56,588: [93823] m.c.manticore:INFO: Generated testcase No. 2 - RETURN(3 txs)
2022-10-17 20:25:56,589: [93823] m.c.plugin:WARNING: Caught will_solve in state None, but failed to capture its initialization
2022-10-17 20:25:56,767: [93820] m.c.manticore:INFO: Generated testcase No. 0 - RETURN(3 txs)
2022-10-17 20:25:56,769: [93820] m.c.plugin:WARNING: Caught will_solve in state None, but failed to capture its initialization
2022-10-17 20:25:56,809: [93825] m.c.manticore:INFO: Generated testcase No. 3 - RETURN(3 txs)
2022-10-17 20:25:56,811: [93825] m.c.plugin:WARNING: Caught will_solve in state None, but failed to capture its initialization
2022-10-17 20:25:56,905: [93824] m.c.manticore:INFO: Generated testcase No. 4 - RETURN(3 txs)
2022-10-17 20:25:56,927: [93835] m.c.manticore:INFO: Generated testcase No. 5 - RETURN(3 txs)
2022-10-17 20:25:56,928: [93835] m.c.plugin:WARNING: Caught will_solve in state None, but failed to capture its initialization
2022-10-17 20:25:56,931: [93824] m.c.plugin:WARNING: Caught will_solve in state None, but failed to capture its initialization
2022-10-17 20:25:57,003: [93829] m.c.manticore:INFO: Generated testcase No. 6 - RETURN(3 txs)
2022-10-17 20:25:57,009: [93829] m.c.plugin:WARNING: Caught will_solve in state None, but failed to capture its initialization
2022-10-17 20:25:57,051: [93828] m.c.manticore:INFO: Generated testcase No. 7 - RETURN(3 txs)
2022-10-17 20:25:57,052: [93828] m.c.plugin:WARNING: Caught will_solve in state None, but failed to capture its initialization
2022-10-17 20:25:57,145: [93822] m.c.manticore:INFO: Generated testcase No. 8 - RETURN(3 txs)
2022-10-17 20:25:57,159: [93822] m.c.plugin:WARNING: Caught will_solve in state None, but failed to capture its initialization
2022-10-17 20:25:57,257: [93834] m.c.manticore:INFO: Generated testcase No. 9 - RETURN(3 txs)
2022-10-17 20:25:57,280: [93834] m.c.plugin:WARNING: Caught will_solve in state None, but failed to capture its initialization
2022-10-17 20:26:00,357: [93821] m.c.manticore:INFO: Generated testcase No. 10 - RETURN(3 txs)
2022-10-17 20:26:00,665: [93820] m.c.manticore:INFO: Generated testcase No. 11 - RETURN(3 txs)
2022-10-17 20:26:00,746: [93828] m.c.manticore:INFO: Generated testcase No. 12 - RETURN(3 txs)
2022-10-17 20:26:00,822: [93823] m.c.manticore:INFO: Generated testcase No. 13 - RETURN(3 txs)
2022-10-17 20:26:00,898: [93825] m.c.manticore:INFO: Generated testcase No. 14 - RETURN(3 txs)
2022-10-17 20:26:01,210: [93835] m.c.manticore:INFO: Generated testcase No. 15 - RETURN(3 txs)
2022-10-17 20:26:01,228: [93824] m.c.manticore:INFO: Generated testcase No. 16 - RETURN(3 txs)
2022-10-17 20:26:01,355: [93822] m.c.manticore:INFO: Generated testcase No. 17 - RETURN(3 txs)
2022-10-17 20:26:01,538: [93834] m.c.manticore:INFO: Generated testcase No. 18 - RETURN(3 txs)
2022-10-17 20:26:02,868: [93829] m.c.manticore:INFO: Generated testcase No. 19 - RETURN(3 txs)
2022-10-17 20:26:03,486: [93823] m.c.manticore:INFO: Generated testcase No. 20 - RETURN(3 txs)
2022-10-17 20:26:04,499: [93834] m.c.manticore:INFO: Generated testcase No. 21 - RETURN(3 txs)
2022-10-17 20:26:04,802: [93824] m.c.manticore:INFO: Generated testcase No. 22 - RETURN(3 txs)
2022-10-17 20:26:05,055: [93825] m.c.manticore:INFO: Generated testcase No. 24 - RETURN(3 txs)
2022-10-17 20:26:05,068: [93835] m.c.manticore:INFO: Generated testcase No. 23 - RETURN(3 txs)
2022-10-17 20:26:05,273: [93828] m.c.manticore:INFO: Generated testcase No. 25 - RETURN(3 txs)
2022-10-17 20:26:05,373: [93820] m.c.manticore:INFO: Generated testcase No. 26 - RETURN(3 txs)
2022-10-17 20:26:05,971: [93821] m.c.manticore:INFO: Generated testcase No. 27 - STOP(3 txs)
2022-10-17 20:26:06,521: [93822] m.c.manticore:INFO: Generated testcase No. 28 - RETURN(3 txs)
2022-10-17 20:26:06,941: [93823] m.c.manticore:INFO: Generated testcase No. 29 - RETURN(3 txs)
2022-10-17 20:26:07,218: [93829] m.c.manticore:INFO: Generated testcase No. 30 - RETURN(3 txs)
2022-10-17 20:26:08,770: [93835] m.c.manticore:INFO: Generated testcase No. 31 - STOP(3 txs)
2022-10-17 20:26:08,839: [93828] m.c.manticore:INFO: Generated testcase No. 32 - STOP(3 txs)
2022-10-17 20:26:08,944: [93834] m.c.manticore:INFO: Generated testcase No. 33 - STOP(3 txs)
2022-10-17 20:26:09,178: [93824] m.c.manticore:INFO: Generated testcase No. 34 - STOP(3 txs)
2022-10-17 20:26:09,656: [93822] m.c.manticore:INFO: Generated testcase No. 35 - STOP(3 txs)
2022-10-17 20:26:13,798: [93653] m.c.manticore:INFO: Results in /home/pavan/Workspace/manticore/examples/evm/mcore_4nhpscmn
2022-10-17 20:26:13,798: [93653] m.c.manticore:INFO: Total time: 55.185728788375854
pavan@pavan-VirtualBox:~/Workspace/manticore/examples/evm (ADSE)$
```

Output2: where the works are reinitiated and the results are obtained in lesser time when compared with the previous output.



```
pragma solidity ^0.8.13;

contract Target {
    function isContract(address account) public view returns (bool) {
        // This method relies on extcodesize, which returns 0 for contracts in
        // construction, since the code is only stored at the end of the
        // constructor execution.
        uint size;
        assembly {
            size := extcodesize(account)
        }
        return size > 0;
    }

    bool public pwned = false;

    function protected() external {
        require(!isContract(msg.sender), "no contract allowed");
        pwned = true;
    }
}
```

Fig 6: Sample contract execution result.

4. Risks:

Modification to the manticore API file in local turned out to be very challenging.

Picking up the proper solidity program for testing the against the time complexity was challenging.

The latest libraries, which are added to the latest versions are not supported by the Z3.

Manticore analysis reports, which are generated by manticore were not user friendly and clear.

OS centric and testing on various machines needed to be done on centric bugs raised by the developers.

For the faster analysis, we required a powerful local machine setup.

As a tool, the manticore CLI has additional dependencies on the manticore API's which will have long term effects.

5. Customers and Users:

Usually, Symbolic Execution for Smart Contracts are used by the Smart Contract Engineers and Smart Contract Developers. However general customers involve those who would like to perform symbolic testing for scripts written in various languages like python, solidity, etc as the Manticore tool supports those languages mentioned. This is an open-source dynamic framework and the end user, or the customer could be the general audience who need the convenience of having the automated analysis for binaries and Ethereum smart contracts.

The Symbolic Execution for Analysis of Smart Contracts using Manticore can be helpful in the following ways: [1]

- Create inputs automatically for a variety of special code paths.
- Trace the inputs that caused the program to crash.
- Keep track of the execution of each instruction.
- Make its analytical engine accessible via the Python API.

The end users can use the feedback returned by the manticore symbolic analysis results in finding the unique code paths and inputs that crashes the programs. The Manticore which after the API's being modified performs better with regards to the results and also has the bugs resolved which are identified as the open issues by the developers.

References:

- **Git hub issue Link:** https://github.com/trailofbits/manticore/issues/2427
- [1] https:/medium.com/haloblock/introduction-to-manticore-a-symbolic-analysis-tool-for-smart-contract- 9de08dae4e1e
- Fig 6: sample code: https://solidity-by-example.org/
- Fig 3: https://arxiv.org/pdf/1907.03890.pdf
- Git hub pull request: https://github.com/trailofbits/manticore/pull/2597