

Paper Review: Insight: In-situ Online Service Failure Path Inference in Production Computing Infrastructures

Sohail Ahmed Shaikh

February 12, 2019

1 Summary:

This paper describes Insight, which is a system that reproduces the execution path of a failed service when a failure is detected. Insight achieves this by creating a shadow copy of the production server dynamically. The binary is executed on the shadow server to explore and gain information about how the failure occurred, by utilizing environment data and runtime outputs as guides. The authors also demonstrate the system in a production cloud management system.

2 Description:

When an online service experiences failures, the debugging information available is often scarce. This is certainly true for non crashing failures. Since these failures go unnoticed it becomes difficult to track them. Also the error messages that are available might lack detail and can be misleading. Insight attempts to address this problem.

Insight can infer the execution path of a failed service request inside the production environment. Insight detects failures employing system anomaly detection tools. When a failure is detected, Insight dynamically creates a shadow component of the faulty server. The state of the shadow component is assumed to be similar to the state of the production server since the cloning process is done immediately. The system also requires brief stopping of the faulty production server so that a copy of all the pages to the shadow server is performed. The shadow server is then reconfigured to prepare it for failure reproduction for example, the network stack will have to be reconfigured as it would be the same as the faulty server. The shadow server is allowed to interact with the other servers in the production environment for failure reproduction. In this scheme reads are allowed, however, writes are filtered out to avoid undesired disturbance to the production environment.

Insight then finds the failure path in the shadow component by performing a binary execution exploration, which is guided. This is done by intercepting conditional jump statements in the binary code. The last input, in the input log is replayed to start the execution exploration. During replay the output of the shadow component is compared with the output of the server. An unmatched output is a clue that the path being explored is wrong and the execution is rolled back. Concurrent multi-path search is achieved in this system by using a set of probing processes/threads. This is done by forking. When a conditional jump is encountered, the parent probe, forks a child probe for exploring both the true and false branches concurrently. Overloading the system is avoided by having a concurrency quota. This process is continued until the system finds outputs that match the faulty server to isolate the calls that caused the problem.

The runtime outputs that are matched in the system are from: console log messages and system calls. Console log messages are valuable since they are inserted by developer to understand error conditions. If the number of console log messages are less, system call traces could be used as they can be easily collected using kernel level tracing tools with negligible overhead and less context switches.

The authors finally, also evaluate the performance of the system on some open source production environments and VCL. From the results, the author infer that Insight achieves high accuracy. However, environment data plays a crucial role in timely failure path finding. Also, the system overhead is quite low and imposes $< 1\%$ performance impact and $< 1.5\%$ CPU load.

3 Strong points:

- 1) It does not require source code of the application.
- 2) It is extremely lightweight and has less overhead.

4 Weak points:

- 1) It works in only a VM environent, since shadow cloning depends on live VM cloning procedures.
- 2) Since, a shadow clone is created on a separate physical host, additional resources are required.
- 3) It requires the faulty server to be stopped briefly for the creation of the shadow server.
- 4) Currently works on Perl and C/C++ only.