November 2019

PRE-OS REMOTE ANALYSIS ON BROKEN SYSTEM

HP INC

Pre-OS Remote Analysis on Broken System

Problem:

Current BIOS designs can log the latest post code in the ECMOS for a developer to track the latest system fail and can also log through WMI to track the system fail in the OS environment. However, this still requires that the failed unit can still complete a successful boot up after the fail. Sometimes, if the machine cannot recover after the fail, meaning it cannot complete the boot sequence (boot to OS, Storage, Shell, DOS…etc.), it will become a dead board. Therefore, the services team will never know what the latest fail post code is and might not able to further investigate the fail point.

Objectives:

- Record latest BIOS post code or error log
- Analyze the fail symptom without needing a successful boot
- Provide remote issue triage

Solution:

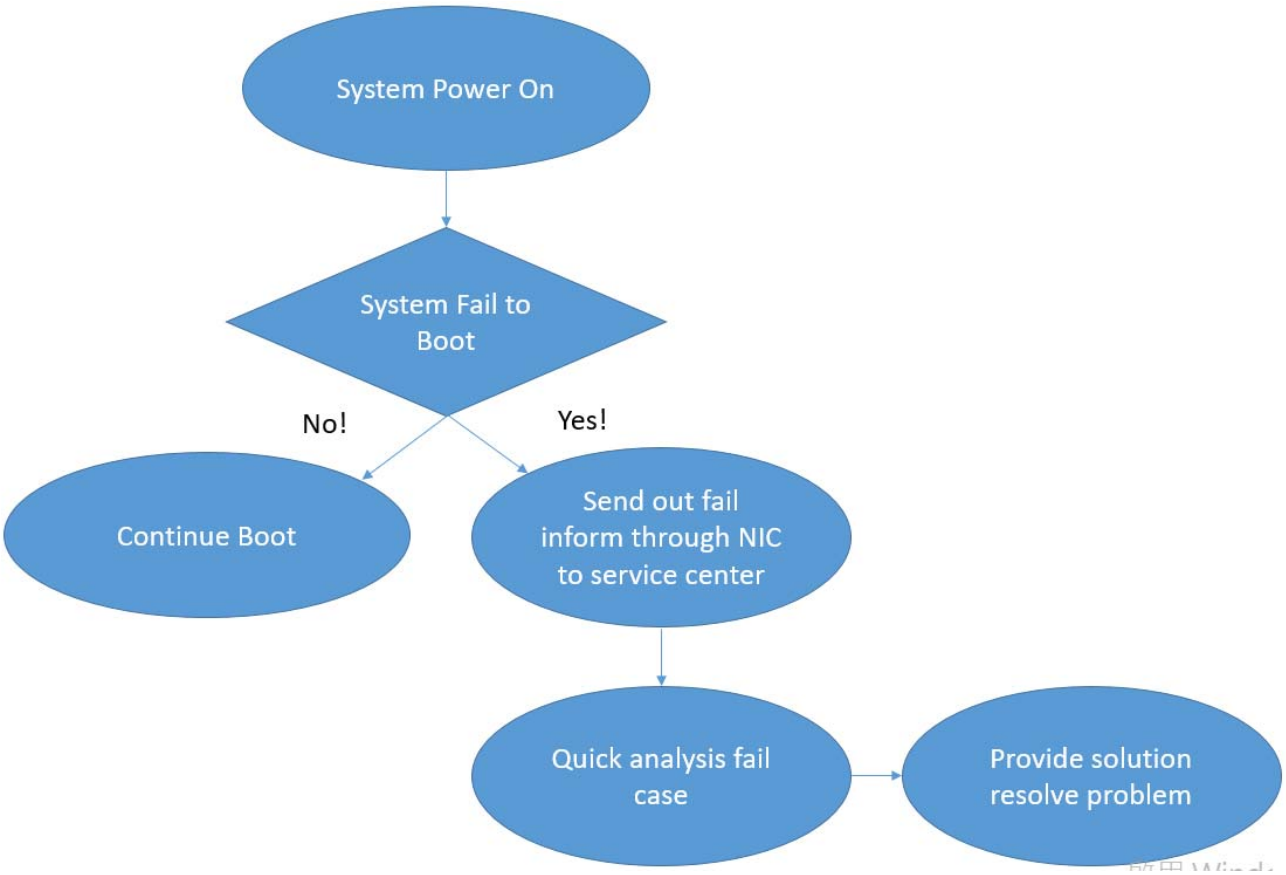
Most notebook (NB) and desktop (DT) systems have an onboard NIC on the unit, and we know that once the system has power after a user presses the power button, the NIC has the capability to receive/send messages through the IPV4 protocol, even if the system cannot boot successfully.

So far, there are some algorithms to collect user/customer machine status or latest failed-boot BIOS post code through a network so an OEM can collect big data to analyze and identify the most common failures customers have experienced. The OEM can use this information to provide improvements for the next generation of systems. However, such techniques require that the affected systems can be rebooted into the OS. After such a reboot, the vendors AP (OEM) will connect to the server and send such logs back for fail symptom analysis once the network can be used. But, if the system cannot boot into the OS anymore, then such algorithms will fail.

To prevent such a situation and allow the service team to still be able to analyze the fail point, the NIC can be used to send a failure message once a system has power. We can work with NIC vendor to define the format or message that we want to send through the NIC. The NIC can connect to the internet even if the system fails to boot to the OS. Currently, a NIC can connect to Ethernet without a full system boot up, therefore we can use such fundamental behavior to create/define our algorithm and the NIC vendor can build up specific firmware for us to achieve that.

For example, BIOS post code is the first thing we need to check once we encounter a boot fail. A major check point can let the service team quickly resolve customer problems without fail system delivery, e.g., the system may be stuck on memory or graphics or boot device…etc. So, we can define what we want to send out from the failed unit to a service server for a quick triage.

Flow Chart – Algorithm1



Disclosed by Chia-Cheng Lin, Harry Chang, Matt Lin and Sharon Wei, HP, Inc.