Contribution ID: **273**   Type: **not specified**

Debugging RISC-V Kernel Crashes: KDUMP, vmcore, and Crash Utility Enhancements

Background

When we bring up a RISC-V board from a chipset vendor, the kernel log cannot give us enough details about what happens inside the kernel. Kernel logs do not contain sufficient debugging information. Because of this, a vmcore is necessary to understand what is really happening inside the kernel.

For binary analysis, many Linux developers use the vmcore file. They usually enable KDUMP, then extract the vmcore from the device, and finally analyze it. The vmcore is a kernel dump that keeps a memory snapshot at the exact moment when the crash happened.

It is more important to show the process of enabling KDUMP and extracting vmcore on real RISC-V hardware boards, such as VisionFive2, than only demonstrating it on QEMU. QEMU is useful as a reference board, but it is not the same as the real boards that engineers use in production.

Challenge

1. In this conference, we will first introduce the method and patches for extracting vmcore from RISC-V device(e.g: VisionFive2). We will also provide a demo of analyzing vmcore with the crash utility.

2. When we enabled KDUMP, the vmcore was not extracted immediately. We had to apply specific patches to solve this problem, that will be discussed.

3. Another issue appeared when loading the vmcore into the crash utility. Some important information was not printed correctly We developed and applied a patch set to make the debugging information display properly.

Future Work

We plan to extend this work for wider usage. Many silicon vendors face unexpected issues when they develop RISC-V devices. In such cases, the vmcore is critical because it provides meaningful signatures such as CSRs and kernel data structures.

As our future tasks:

1. We will enable KDUMP and extract vmcore on Beagle boards and other popular RISC-V boards. If problems appear, we will solve them with additional patches.

2. We will continue to improve the crash utility so that it can show correct and complete debugging information when analyzing RISC-V based vmcore files.

Primary author: KIM, Austin (LG Electronics)

Presenter: KIM, Austin (LG Electronics)

Session Classification: RISC-V MC

Track Classification: RISC-V MC