Building Resilience Against Hardware and Software Errors in Cloud Data Centers

H. Howie Huang
The George Washington University
howie@gwu.edu

Delivering reliable services in cloud data centers is of paramount importance, as errors are no longer a rare occurrence at scale. With thousands of CPU cores running even more virtual machines, any large-scale virtualization infrastructure needs a robust mechanism for achieving high reliability, which monitors the health of computer systems in runtime, detects hardware and software errors quickly, recovers from erroneous states, and preferably doing so in an automated manner.

Towards achieving this goal, this position paper proposes a new experiment of collecting and analyzing hardware and software errors within the Chameleon and CloudLab projects (NSFCloud). We propose to collect a comprehensive set of logs on NSFCloud over an extended period of time (e.g., one year), perform in-depth analysis of the errors, and share the analysis and traces with the CISE community. Currently public traces were collected with a different focus [1, 3, 4, 5].

In this project, we are interested in understanding reliability implications of hardware and software errors on the large number of computer nodes when running various types of workloads. To do this, we will leverage existing measurement capabilities that are already built in the hardware and software. For example, modern CPUs have a number of hardware performance counters to report performance events related to the processor and memory controller. Furthermore, the software stack including the virtualization layer (e.g., Xen and KVM), operating system (e.g., Linux), and applications also provide various levels of event logging capability. On the system level, the industry standards, e.g., IPMI (Intelligent Platform Management Interface)[2], support sensing and monitoring on a wide range of system states, e.g., temperatures, fans, and voltages.

A wide range of hardware errors, both permanent (hard) and transient (soft) errors will be investigated, including CPU, memory and I/O errors. Similarly, we are interested in tracking software errors that happen in high-level applications, operating systems, and hypervisors. The errors include application crashes, kernel panic, system hangs, crashes, etc. Many of these software events can be captured by using the tools in cloud management system, hypervisor, and operating system. One important aspect of this study is to correlate hardware measurement with high-level software activities, e.g., VM creation and migration.

In all, we plan to not only gather high-level statistics (e.g., how many errors per hour and per day, and error types), but also obtain the context of an error (e.g., when and where it happens). Once the traces are collected, we will apply data cleaning to remove possible inconsistency, and the obfuscation techniques to remove potentially sensitive information. All the traces will be stored and later released online for public dissemination.

We believe that this long-term, comprehensive trace data will fill a critical need of the community. Built upon our early work [7, 6], we will also leverage the data from this study in several of our on-going efforts in characterizing the resource usage of a virtualized data center, understanding the impacts of hardware and software failures, analyzing the behaviors of users, applications, and operating systems, as well as developing new techniques to achieve high reliability in virtualized data centers.

Acknowledgment

This work is supported in part by National Science Foundation grants 1350766, 1320226, and 1124813.

References

[1] The computer failure data repository (cfdr). http://http://cfdr.usenix.org/.

[2] Intelligent platform management interface (ipmi). http://http://www.intel.com/design/servers/ipmi/.

[3] Google. Goolge cluster data version 2. Online, 2011. http://code.google.com/p/googleclusterdata.

[4] SNIA. Storage Networking Industry Association. IOTTA repository. http://iotta.snia.org/traces.

[5] UMass. Umass trace repository. http://traces.cs.umass.edu/index.php/Storage/Storage, 2007.

[6] Xin Xu, Ron Chiang and H. Howie Huang. Xentry: Hypervisor-level soft error detection. In *The 43rd International Conference on Parallel Processing (ICPP14)*, Minneapolis, MN, September 2014.

[7] Xin Xu and H. Howie Huang. Understanding reliability implication of hardware error in virtualization infrastructure. In *10th Workshop on Hot Topics in System Dependability (HotDep 14)*, Broomfield, CO, October 2014. USENIX Association.