Review of

Microreboot

George Candea et al.

What is the problem?

Rebooting is a trivial and useful way to fix software failures in large-scale systems but this entails service disruptions and downtime caused by this may cause significant damage.

Summary

Using rebooting as a way to to reclaim resources and to fix a failure could cause a lot of collateral damage. When softwares lack clean separation between data and process recovery, data loss and unexpected recovery problems could arise. Microreboot introduces an individual rebooting of fine-grain application components. Microreboot is meant to be a front-line first attempt to recover the system.

Key Insights

- The paper identifies the main issue of decoupling. The components must be loosely coupled in order to tolerate a microreboot.
- They also identify that micro reboots are not only better in terms of performance and overhead, but also the fact that using micro reboots makes the system recovery faster.

Strengths

- The detailed explanations about designing a microrebootable software acutely identifies all the different design considerations needed.
- Microreboot protects the system for data loss and enables an uninterrupted experience for the user.

Weaknesses

- There is an unknown impact to shared state, as the paper identifies. A shared state between two components may be left inconsistent when a micro reboot takes place whereas a full reboot will make all systems reboot simultaneously so this shared state inconsistency is never discovered in that case.
- They have not discussed enough in the case of making existing systems micro reboot friendly, the guidelines to creating systems were clear but what of current systems that aren't J2EE.

Summary of Key Results

- EJB-level or WAR-level micro rebooting in j2EE prototype that the paper uses is effective in recovering a majority of the failure modes seen in today's production J2EE systems. But

microcreboot is ineffective against other types of failures, where only coarser grained reboots or complete manual repair is required.
- Using micro reboots instead of JVM restarts reduced the number of failed requests by 98%.
- Using rebooting is only a means of recovering the system for the time being; the underlying issue must be resolved after recovery.

Open Questions

- Can it offer the same experience even in systems that don't work with the Internet?
- Is microreboot valuable for small systems in which there are only few components and all of them are interlinked?
- Can we use microreboot system to also identify and simultaneously try to fix the failure cause?