Hypervisor-based Persistence for Virtual Machines

Vasily A. Sartakov
TU Braunschweig
sartakov@tu-bs.de

Rüdiger Kapitza
TU Braunschweig
rkapitza@tu-bs.de

ABSTRACT

Power outages and subsequent recovery are a major cause for service downtimes. This issue is amplified by the ongoing trend of steadily growing in-memory state of Internet-based services which increases the risk of data loss and extends recovery time. Protective measures against power outages such as an uninterruptible power supply are expensive, maintenance-intensive and often fragile. With the advent of non-volatile random-access memory provided by commodity servers, there is a scalable, less costly and robust alternative to recover from power outages and other failures without major data loss. However, as of today off-the-shelf software is not ready for benefiting from persistent main memory.

In this poster, we present NV-Hypervisor a lightweight hypervisor extension that transparently provides persistence for virtual machines. NV-Hypervisor paves the way for utilising persistent memory in virtualised environments (i.e., infrastructure-as-a-service clouds) and protects memory-heavy services such as key-value stores and databases from data loss and time-consuming recovery.

Introduction

In today's rapidly progressing information society, we rely more and more on the availability of Internet-based services of all kinds. To cope with the demand to provide services that are available 24/7, service implementations as well as hosting infrastructure should be resilient to faults. One cause for service disruptions are power outages that can be addressed by fault-tolerance features of service implementations (i.e., a crash-tolerant design) and additional infrastructure, such as an uninterruptible power supply. While the former results in a time-consuming recovery operation, the latter is expensive, maintenance-intensive and nevertheless fragile [2].

With the advent of non-volatile random-access memory (NV-RAM) provided by commodity servers, in-memory data can be retained without an external source of power. This not only enables to tolerate power outages without data loss, but also provides additional benefits such as preserving data in case of system crashes. While there is a number of evolving technologies such as phase-change random-access memory and spin-transfer torque random-access memory, there are already market available solutions that utilise commodity memory technology [1]. For example by providing DRAM memory modules that are backed by flash memory of the same size and a capacitor. In case of a power drop the module uses the capacitor energy to mirror DRAM state to flash memory and vice versa at recovery time.

The availability of persistent main memory needs to be reflected throughout the whole software stack. So far a number of different approaches have been proposed: at the user-space level as a new memory allocator [3], at the kernel level where persistence becomes a new feature of processes[5] and system-wide[4]. All of these approaches, however, require software modifications to utilise the support of NV-RAM.

We propose *hypervisor-based persistence* as a means to enable NV-RAM-usage for legacy software. Thereby, neither system nor application software of a virtual machine has to be adapted as NV-RAM support is transparently provided by the virtualisation layer. We have implemented hypervisor-based persistence as part of NV-Hypervisor, which builds a lightweight extension to the QEMU virtualisation platform. To create a persistent virtual machine, NV-Hypervisor utilises a special allocator to place the virtual machine in non-volatile memory. If power drops, this is noticed by a power outage detector that emits an interrupt to initiate a fast power-down procedure. Next, volatile CPU and management state related to persistent virtual machines is saved in non-volatile memory using the remaining residual energy of the power supply. Once the system is rebooted, NV-Hypervisor recovers the persistent virtual machines.

In our evaluation we have used a market available, capacitor-backed NV-RAM solution [1] and measured the time to recover a database after a power outage. Results are promising (see bottom half of our poster): while the commodity solution recovers slightly faster, our NV-Hypervisor-based solution enables services to continue request processing at full speed immediately after recovery without any data loss.

We consider hypervisor-based persistence a promising means to take immediate advantage of market available persistent memory, especially for proprietary and legacy software. In a next step, we will optimise our current implementation by placing parts of the virtualisation layer and the host operating system in non-volatile memory.

[1] Viking Technology. ArxCis-NV (TM) Non-Volatile Memory Technology. `http://www.vikingtechnology.com`, 2012.

[2] 2013 Cost of Data Center Outages, Ponemon Inst., 2013.

[3] Coburn, J., Caulfield, A. M., Akel, A., Grupp, L. M., Gupta, R. K., Jhala, R., and Swanson, S. Nv-heaps: making persistent objects fast and safe with next-generation, non-volatile memories. In *ACM SIGARCH Comp. Arch. News* (2011), vol. 39, pp. 105–118.

[4] Narayanan, D., and Hodson, O. Whole-system persistence. In *ACM SIGARCH Com. Arch. News* (2012), vol. 40, pp. 401–410.

[5] Wang, X. L. K. L. X., and Zhou, X. NV-process: A Fault-Tolerance Process Model Based on Non-Volatile Memory. In *Asia-Pacific Work. on Systems* (2012).

Technische Universität Braunschweig

Institut für Betriebssysteme und Rechnerverbund

Hypervisor-based Persistence

Vasily A. Sartakov, Rüdiger Kapitza
Technische Universität Braunschweig | Institut für Betriebssysteme und Rechnerverbund

Motivation

Problem

- Power outages lead to data corruption and service downtime
- Use of Uninterruptable Power Supply is expensive, maintenance-intensive and fragile

Non-volatile memory (NV-RAM) a promising solution

- Maintains **data persistence despite power outages**
- Allows to decrease the required recovery time

NV-RAM Technologies

Multiple evolving technologies:

- MRAM, PC-RAM, STT-RAM, BB-RAM
- Initial market available systems

Challenge: Integration of software with NV-RAM

No solution allows direct usage of NV-RAM for legacy and proprietary software!

Approach

NV-Hypervisor

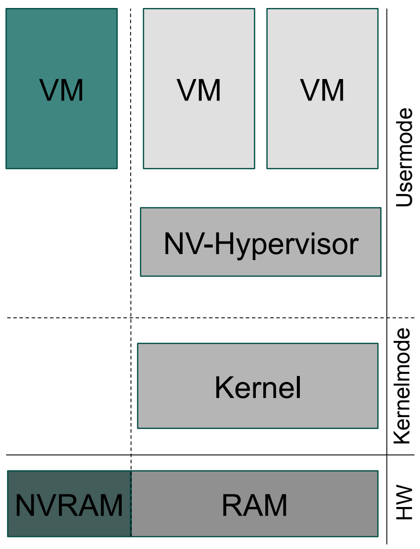
Transparent persistence for VMs based on a NV-RAM-aware hypervisor

→Provides support for legacy software

Hardware assumptions

- Volatile CPU and device states
- Hybrid memory: NV-RAM and DRAM co-exist
- Power-fault detector
- Sufficient residual energy for saving CPU state

System Overview



Managing persistent VMs

VMs are placed in RAM or NV-RAM according to user demand

- Special allocator manages NV-RAM memory

Power fault handler initiates saving sequence when power outage occurs

- Volatile devices and CPU states of VMs are saved

Persistent VMs are restored on power recovery

- Restore device and CPU states
- Resurrect VMs stored in NV-RAM

Evaluation

Comparisons

- System recovery time
- Performance recovery time

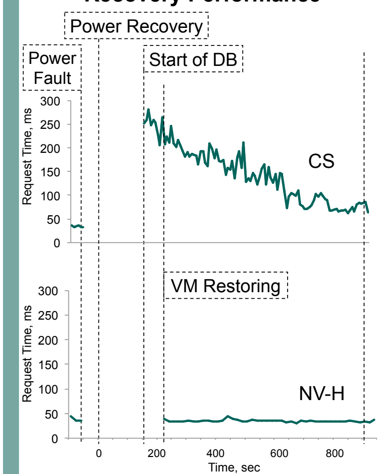
Hardware

- Capacitor-backed NV-RAM provided by Viking Technology
- 2x Xeon, 4GB DRAM & NV-RAM

Software

- Host: Linux (3.4.12) & QEMU
- Guest VM: Linux (3.12.7)
- Evaluation service: MySQL DB
- Workload: Sysbench OLTP

Recovery Performance



Booting Step	Commodity System (sec)	NV-Hypervisor (sec)
DB warm up	566	n/a
DB recovery	54	n/a
Guest OS boot	31	n/a
QEMU starts	0.2	8
Host boot	108	108
BIOS	15	15
NV-DIMMs recovery	n/a	109
Server boot	36	36

Recovery time