DHBW Lörrach

Project „virt-rbd-backup“

Lecture: Programming Languages – Introduction To Python

Lecturer: Dr. Stephan Laage-Witt

Course: TIF20A

Student: Fabian Zaremba

Overview

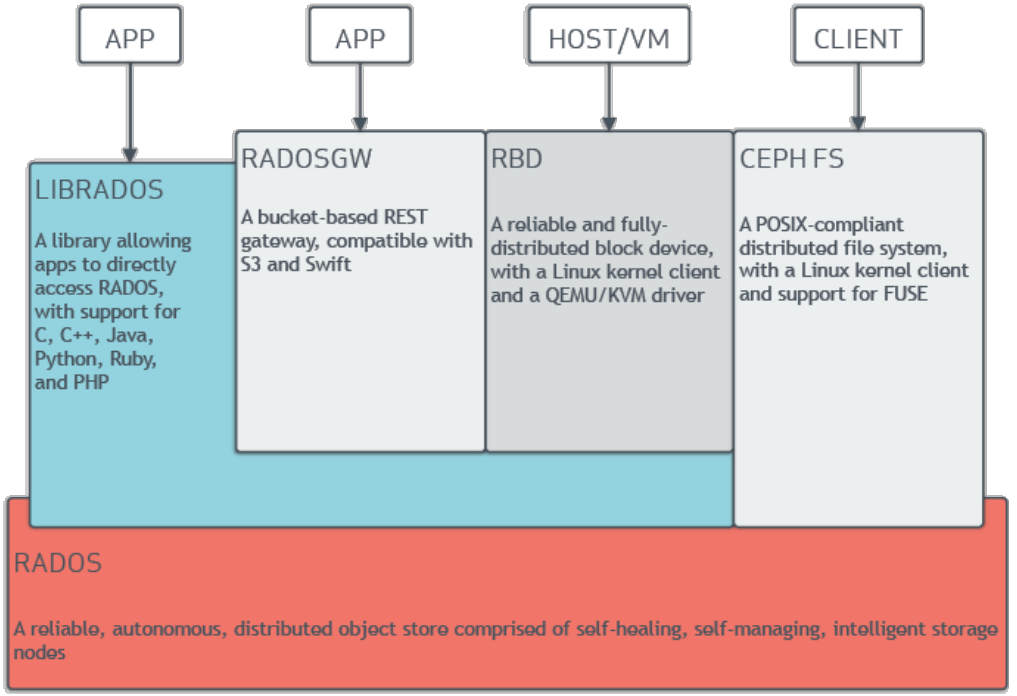
- Project context / Introduction
- Software architecture
- Results
- Excursion: Parallel processing with worker pools
- Conclusions, Evaluation

Project Context

- Ceph

  - Software-defined storage solution
  - Designed for performance, reliability and scalability
  - Clustered architecture (redundant servers)
  - Cluster-aware clients (kernel or libraries)

Project Context



ceph

https://docs.ceph.com/en/latest/architecture/

Project Context

- Ceph

  - RBD: Virtual RADOS block devices providing logical block devices stored in Ceph storage
  - RBD images efficiently support snapshots, clones, resizing, etc.

Project Context

- QEMU
  - Open Source machine emulator and hypervisor
  - Runs isolated guest VM on host OS, provides emulated hardware
  - Supports paravirtualization (e.g. virtio) with abstracted devices instead of emulating real hardware
  - Hardware acceleration with KVM (Kernel Virtual Machine) on Linux
  - Integrated Ceph RBD client for VM storage

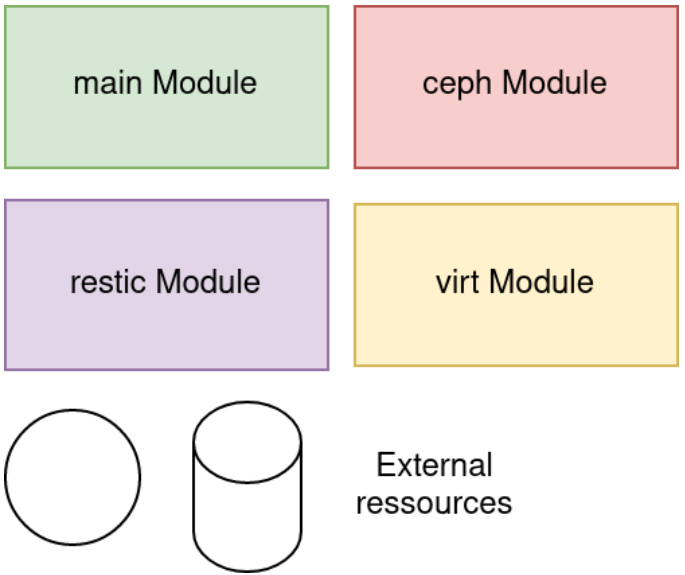
Project Context

- libvirt
  - Open Source VM management solution
  - Provides support for different hypervisors
  - GUI and CLI management tools & API

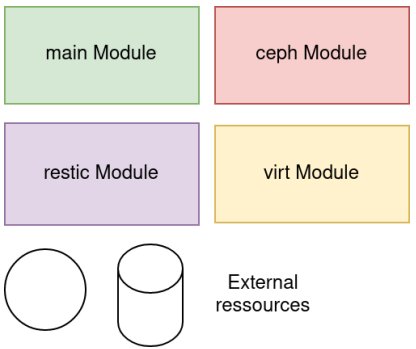
Introduction

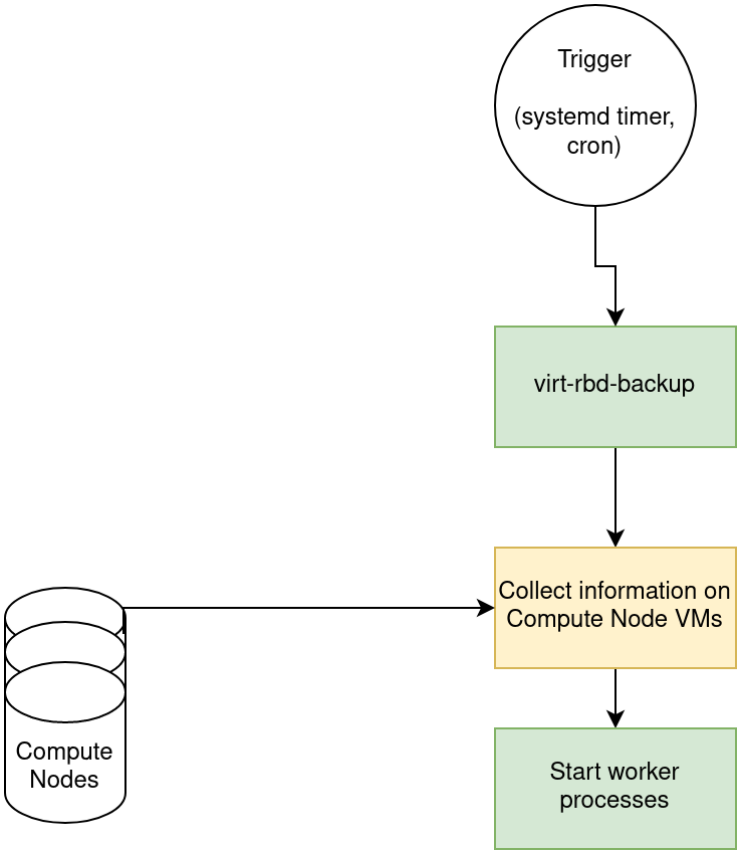
- virt-rbd-backup as part of a backup solution
- Designed for VMs managed by libvirt
- QEMU provides filesystem freeze capability
- Ceph RBD images provide snapshots to minimize service interruptions

Architecture

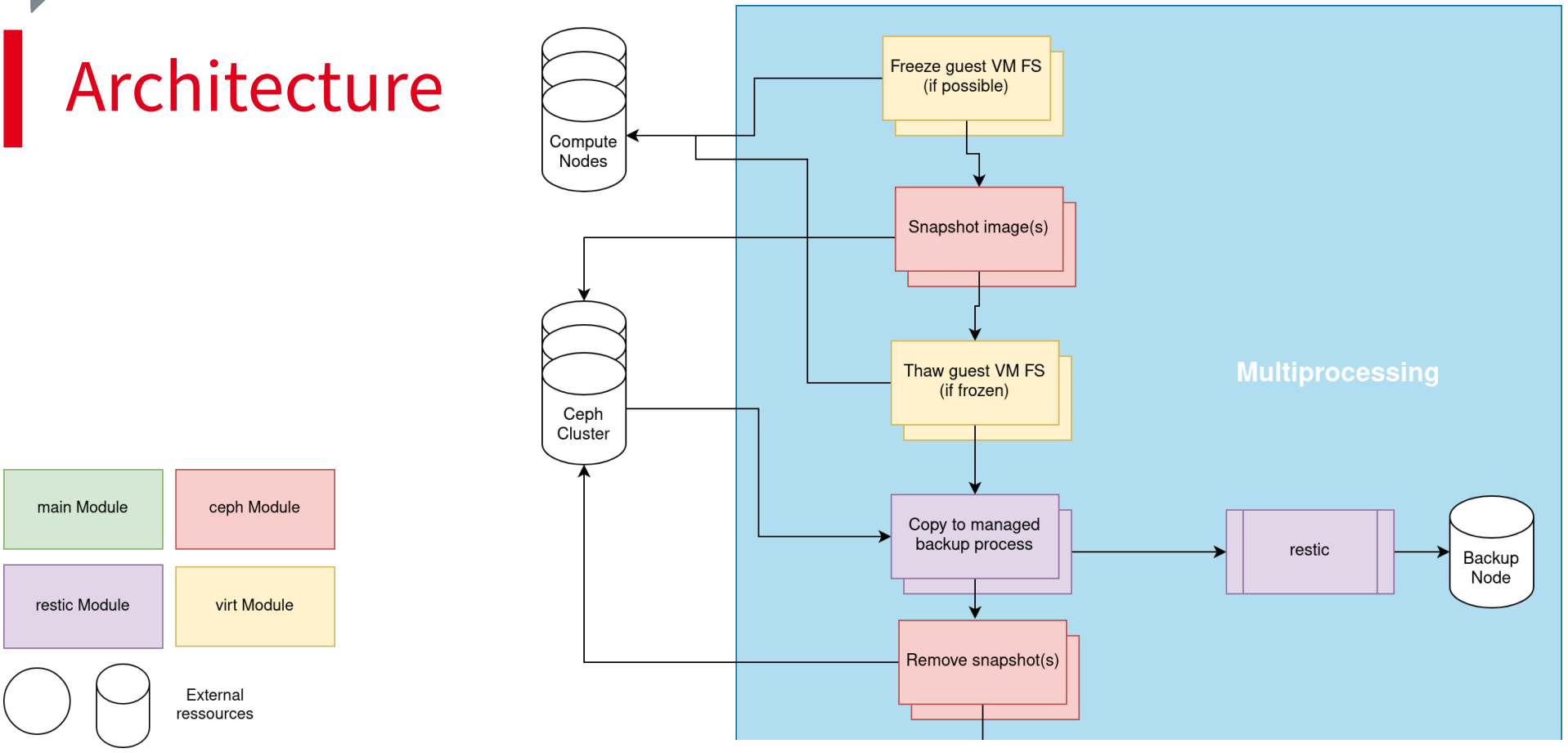


Architecture





Architecture



Results

```
root@node1:~/virt-rbd-backup# python3 main.py
Ignoring non-network disk for domain: TERMSRV
Processing 1 images for domain 7c104238-a949-4c1b-a244-9bcd43de9a21
Processing 1 images for domain e5de2bad-34d8-4171-b676-f6ba03330497
Processing 1 images for domain 7e7ebf0c-f302-44cb-a56f-491ae9967328
Processing 1 images for domain 76e895c8-52ea-4b8f-94dc-0cab169ca4bc
Backup successful: No error occurred for domain e5de2bad-34d8-4171-b676-f6ba03330497
Backup successful: No error occurred for domain 76e895c8-52ea-4b8f-94dc-0cab169ca4bc
Processing 1 images for domain c2813f16-0250-4be1-8fbb-11a26f9ab227
Backup successful: No error occurred for domain 7e7ebf0c-f302-44cb-a56f-491ae9967328
Processing 1 images for domain 41abe680-68b0-45d0-ab6d-7003a729fd8d
Backup successful: No error occurred for domain 41abe680-68b0-45d0-ab6d-7003a729fd8d
Processing 1 images for domain 639a55e4-d0b1-47d1-b5a9-62983a13f0db
Backup successful: No error occurred for domain c2813f16-0250-4be1-8fbb-11a26f9ab227
Backup successful: No error occurred for domain 639a55e4-d0b1-47d1-b5a9-62983a13f0db
Backup successful: No error occurred for domain 7c104238-a949-4c1b-a244-9bcd43de9a21
```

Results

- Verified working backups, stable execution
- Used in production with automated daily backups
  - ~10 mission critical workloads on two compute nodes
  - Two compute nodes using three Ceph cluster nodes (~ 150 TiB total capacity)
- Performance in expected range
  - 3 Workers with 10GbE with speed up to 550 MiB/s
  - Backup of ~600 GiB images in < 20 minutes to backup target server (HDD)

Excursion: Worker Pools

- Python module multiprocessing
- Provides process-based parallelism
- Workers are created as new processes and added to a pool
  - Receives tasks or termination signal over an input queue
  - Returns result (success indication, error message) over output queue to main process

Excursion: Worker Pools

- Configurable amount of spawned workers allow control over resource usage and performance
- Use of queues as synchronization primitive eliminates the need for locking
- Showcase: https://docs.python.org/3/library/multiprocessing.html#examples

Conclusions / Evaluation

- Difficulties with Python
  - Very expressive error handling, requiring nested exception handling to guarantee expected control flow
  - Lack of static typing / checking of type annotations can lead to type confusion errors

Conclusions / Evaluation

- Python 3: Best fit for this project
  - Ceph and libvirt provide Python APIs out of the box
  - Comprehensive standard library avoids dependencies on third party libraries
  - Performance concerns eliminated by efficient upstream APIs and benchmarking