Project Springfield

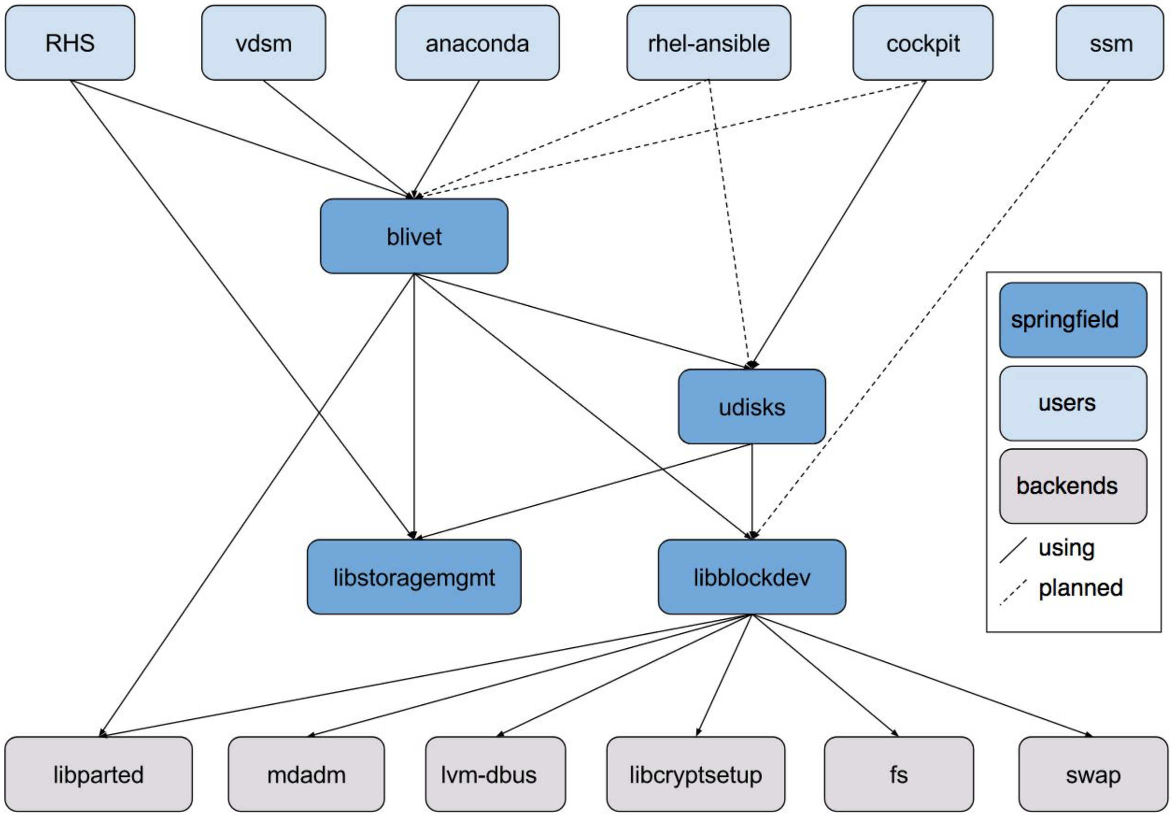
A smart and automated storage

Jan Ťulák - jtulak@redhat.com

What is Springfield?

- A collaborative effort to solve certain problems across projects.
  - No “springfield.so” or any binary/executable, but a joint kernel/userland project.
  - “Officially” right now: udisks, libblockdev, blivet, libstoragemgmt
  - Other projects cooperating more loosely or in early development/planning stages.
- Some issues we are trying to solve:
  - Good for human ≠ good for automation (the UNIX way unfit for all?)
  - Informations are fragmented over multiple sources:
    - Redundant queries.
    - Higher CPU usage.
  - No good notification system for storage.
  - Complex APIs, not easy to use by application developers.

Project interactions



What the specific projects do?

- Blivet
  - Python module for storage configuration.
  - Can model arbitrary number of changes in memory before committing them.
- udisks
  - Daemon providing DBus API for storage configuration and monitoring.
- libblockdev
  - A library wrapping around various fs-utils and storage tools and providing an unified API.
  - Low-level.
- libstoragemgmt
  - Manages a controller-based storage (e.g. NFS).

Discussion time

Three main topics:

- Filesystem API
- Mount API
- Storage Information Provider

Filesystem API - libblockdev

- I asked about the FS API few weeks ago at linux-fsdevel list.
  - Some interest from potential users.
  - Maintainers: Chicken/egg issue.
  - So we reached for something already there that is “good enough”.
- Works right now (C library with Python bindings).
- Does the screenscraping for you.
- All-in-one, maybe too much dependencies (RAID, …).
- Is this enough, is it useful for anyone else?
  - E.g. not everyone wants to depend on a 3rd party library, but we probably can’t do anything else until there is enough of users of this...

Mount API 1/2

- A single source, data right from kernel and supporting large mount tables.
    - A kernel module + userspace library
- Why not just use libmount as it is?
    - A terrible performance with thousands/ten of thousands volumes (containers, autofs, etc.)
    - For example, starting autofs with a direct mount map of 15,000 entries several key processes consume all available CPU (to name a few of these, systemd, udisksd, gvfs-udisks2-volume-monitor and gvfs-trash). And any change does it again.
    - API is too low-level (ideally we want ~5 lines of code, excluding option setup.)

Mount API 2/2

- Mount namespaces complicates everything.
  - /proc is currently the only way to get namespace-specific information.
  - Maintaining a database of namespaces and mounts is too costly even for moderate load.
- So we can’t have a daemon, only a library, right?
- Getting incremental changes of mount table straight from the kernel is important for performance, but it seems like a really difficult thing to do.
  - How to split it? Timestamp and co. is probably too costly/complicated.
  - So a notification to all subscribed listeners on every change?

Storage Information Provider 1/2

- Motivation: no reliable way to detect things like “FS is full”
  - Stacked layers, thin provisioning, deduplication… → user gets write failures for no apparent reason.
  - Notifications, better structured errors able to explain what and where happened.
- Really early preview: https://github.com/raven-au/usip
  - Really, really early

Storage Information Provider 2/2

- How to calculate a relative path to a provided root struct? (Namespaces again…)
  - Some functions in kernel can do it, but they are not exported.
  - It probably needs to be a standalone module - it doesn’t fit into any fs code or VFS.
  - Workqueue tasks are unconditionally created with a shared current->fs so set_fs_root() can't really be done in a worker, so we need to use kernel_thread(), but is not exported too. Would exporting any of these lead to abuse of mount name spaces? (Probably yes… or other issues.)
  - What are the implications of cloning a workqueue worker thread without setting CLONE_FS?

Thank you

Do you know about a project that would benefit from the cooperation? Tell us!

Contact:

- https://springfield-project.github.io/
- springfield@sourceware.org

These slides:

- https://springfield-project.github.io/lsf2018/

Me:

- Jan Ťulák - jtulak@redhat.com