heinlein

Einführung in Ceph
[Chemnitzer Linux-Tage 2021]
Robert Sander <r.sander@heinlein-support.de>

# Einführung in Ceph

- **Heinlein Support**
  - IT-Consulting und 24/7 Linux-Support mit ~40 Mitarbeitern
  - Eigener Betrieb eines ISPs seit 1992
  - Täglich tiefe Einblicke in die Herzen der IT aller Unternehmensgrößen

- 24/7-Notfall-Hotline: 030 / 40 50 5 - 110
  - 28 Spezialisten mit LPIC-2 und LPIC-3
  - Für alles rund um Linux & Server & DMZ
  - Akutes: Downtimes, Performanceprobleme, Hackereinbrüche, Datenverlust
  - Strategisches: Revision, Planung, Beratung, Konfigurationshilfe

# Software defined Storage

# Abstraktion von Hardware

- Hardware ist „egal“
- Fing eigentlich schon mit LVM an
- Beschränkt sich aber nicht nur auf eine Maschine
- Redundanz nicht über RAID-Controller
- Jede Hardware kann ausfallen
  - Software natürlich auch

# Skalierbarkeit

- Beliebig in die Breite skalieren
- Keine „teure“ vertikale Skalierung notwendig
- günstigere Commodity Hardware einsetzbar
- Trotzdem: Blick auf Performance wichtig

Robert Sander <r.sander@heinlein-support.de>

# Ceph

- Es war einmal eine Open Source Speicherlösung namens Ceph

# Ceph ist …

- gesetzt
  - gibt es seit 2006
  - Doktorarbeit von Sage Weil
- interessant
  - verteilter Objektspeicher
  - Redundanz
  - Datensicherheit
  - effiziente Skalierung
  - lauffähig auf (fast) jeder Hardware

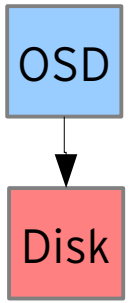
# Ceph bietet …

- einen Storage-Cluster
  - der sich selbst verwaltet
  - der sich selbst heilt
  - ohne Engpässe
- drei Schnittstellen
  - Objektspeicher (kompatibel zu S3)
  - Blockspeicher (für VMs, etc)
  - verteiltes Dateisystem

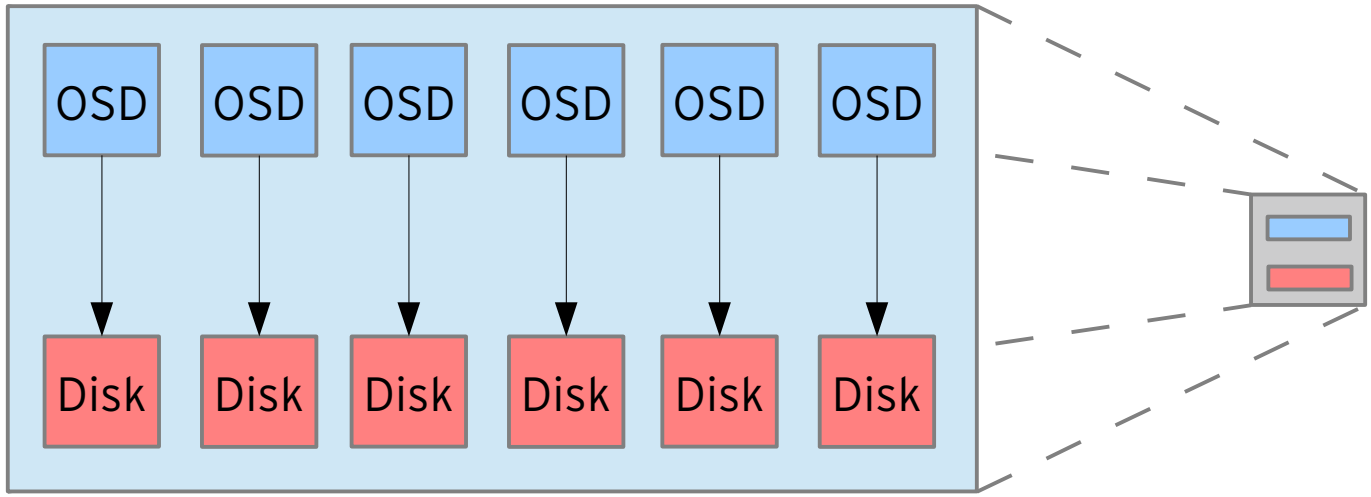
- Es war einmal eine Open Source Speicherlösung namens Ceph
- Mit verschiedenen Komponenten, die zu kennen sich lohnt

# Object Storage Daemon

- OSD speichert Daten auf
  - HDD
  - SSD
  - NVMe
  - oder was es noch geben wird
- Ein Prozess pro Blockdevice
- OSDs liefern Daten direkt an Clients
- OSDs sprechen mit anderen OSDs
  - Replikation
  - Datenwiederherstellung



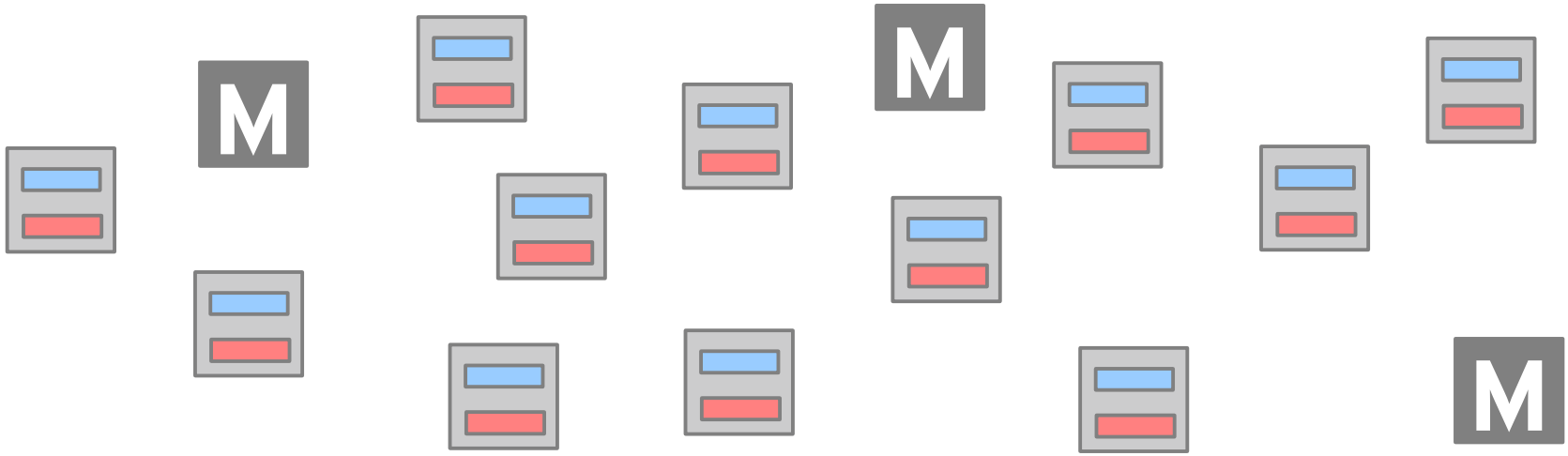
# Mehrere OSDs in einem Ceph-Knoten



# Dazu kommen Monitore

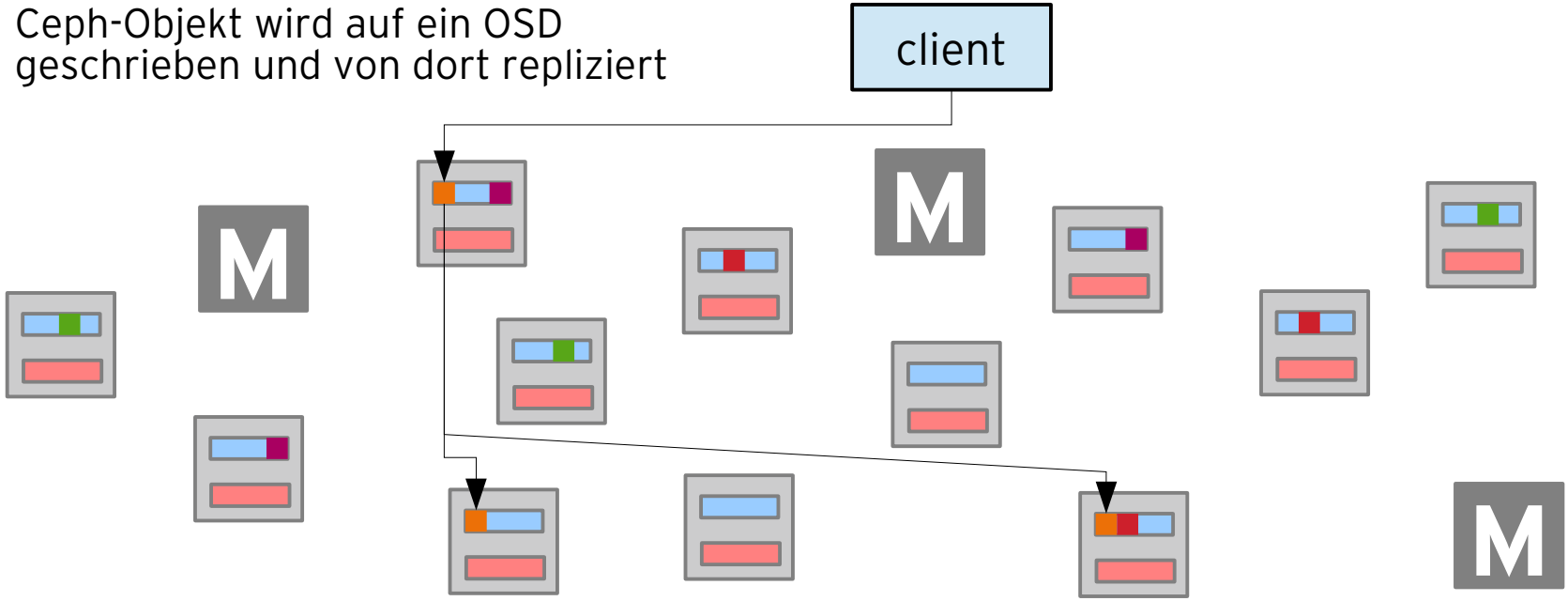
- MONs bilden das Gehirn des Clusters
- Quorum für Entscheidungen
- Nicht im Datenpfad

# Und schon ist der Ceph-Cluster fertig



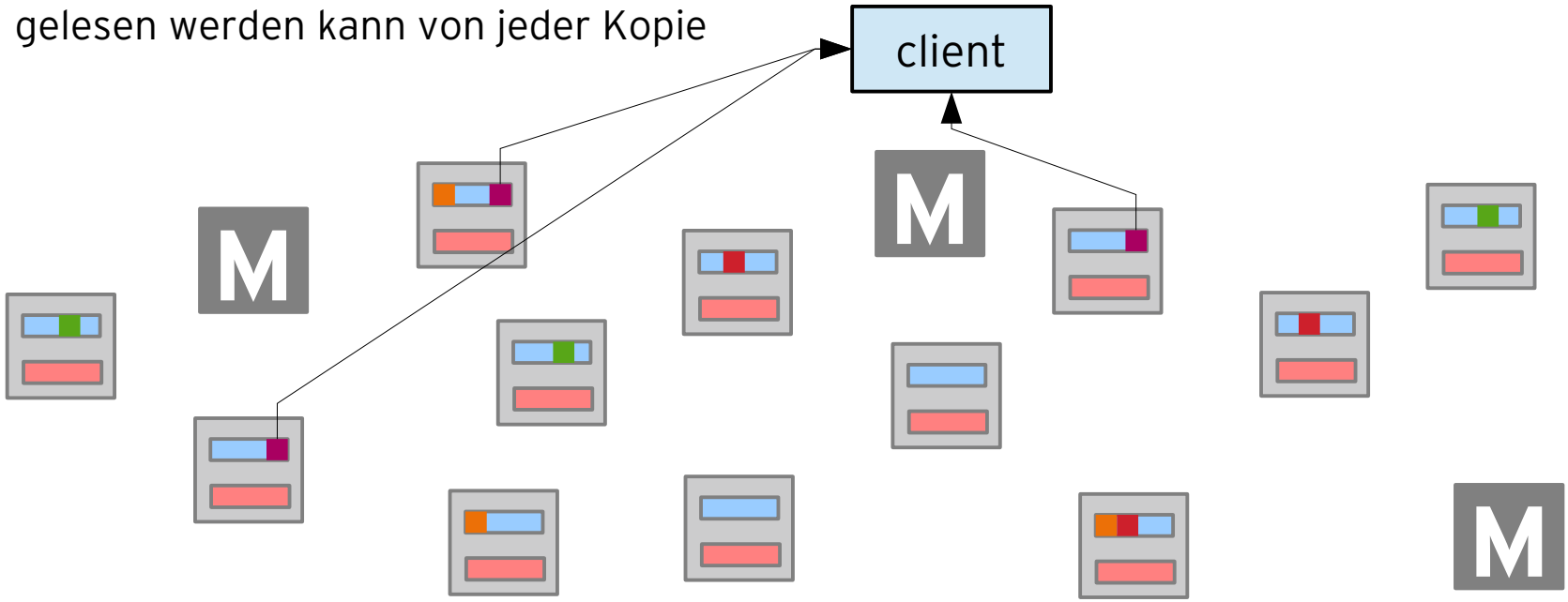
# Schreiben ...

- Ceph-Objekt wird auf ein OSD geschrieben und von dort repliziert



# … und Lesen

- gelesen werden kann von jeder Kopie



# Konzeptionelle Komponenten

## Pool

- logischer Container für Ceph-Objekte
- Eigenschaften
  - Name + ID
  - Anzahl der Objektkopien
  - Erasure Coding Einstellungen
  - CRUSH Regel
  - Besitzer
  - Quota
  - Snapshots

## CRUSH

- Controlled Replication Under Scalable Hashing
  - Algorithmus
- MONs verwalten CRUSH Map
  - Topologie des Clusters
  - Ausfallzonen
- Clients kalkurieren selber
  - Platzierung der Daten
  - keine Engpässe im Datenpfad

- Es war einmal eine Open Source Speicherlösung namens Ceph
- Mit verschiedenen Komponenten, die zu kennen sich lohnt
- Alle wollen Software Defined Storage anstelle von „Legacy“ Storage-Arrays

# Klassische (proprietäre) Speichersysteme

**Vorteile**

- Einfach zu verstehen
- Vorhandene Erfahrung
- „Bauchgefühl“ besser
- Vorhersagbar in Verhalten und Kosten

**Nachteile**

- Streng kontrollierte Umgebung
- Begrenztes Wachstum
- Weniger Optionen
  - bestimmte HDDs / SSDs
  - Anzahl der Blockdevices
  - Netzwerk-Varianten
  - Controller
  - CPU

# Software defined Storage

**Vorteile**

- Selbermachen
- Unbegrenztes Wachstum
- Anpassungsfähigkeit
- Auswahlmöglichkeiten

**Nachteile**

- Selbermachen
- Komplexität
- Performance (Software CPU-bound)

# Software defined Storage

- Durchsatz
- Latenz
- IOPS
- Verfügbarkeit
- Zuverlässigkeit
- Kapazität
- Packungsdichte
- Kosten

# Konzipierung von SDS

**Zielkonflikte**

- → Verfügbarkeit gegen Packungsdichte
- → IOPS gegen Packungsdichte
- → Alles gegen Kosten

- → Große Auswahl an Hardware → Unübersichtlich
- → Softwareauswahl (gibt ja nicht nur Ceph)

- → Es gibt kein Standardrezept, das allen passt

- Es war einmal eine Open Source Speicherlösung namens Ceph
- Mit verschiedenen Komponenten, die zu kennen sich lohnt
- Alle wollen Software Defined Storage anstelle von „Legacy“ Storage-Arrays
- und hatten viele Fragen zu Konfigurationsmöglichkeiten

# Netzwerk

- die schnellste Netzwerktechnologie, die das Budget hergibt
- Client-Zugriffs- und Cluster-Replikations-Netzwerk trennen
- Replikations-Netzwerk mit mindestens Faktor 2 in der Bandbreite
- Ethernet 10G < 40G < 25G < 100G
  - wegen der Latenz

# Storage-Knoten

- CPU, CPU, CPU
- RAM, RAM, RAM
  - 4GB pro OSD
- guter Storage Controller
- SSDs, SSDs, SSDs
- HDDs
  - günstiger
  - eher für Archivdaten

# Redundanz

## Replikation

- *n* genaue Kopien
- hohe Leseraten
- gute Schreibraten
  - schnelles Cluster-Netzwerk
- Wiederherstellung von mehrere Quellen
- Netto-Kapazität nur 1/*n*
  - *n* = 3 → 33%

## Erasure Coding

- Daten aufgeteilt in *k* Teile und *m* Parity
- Platzeffizient
- hoher CPU-Verbrauch beim Schreiben
  - Parity-Berechnung
- Wiederherstellung braucht CPU
- Netto-Kapazität: *k* / (*k* + *m*)
  - *k* = 8, *m* = 2 → 80%
  - *k* = 2, *m* = 2 → 50%
- *k* + *m* + 2 unabhängige Knoten notwendig

# Cluster ausbauen – Mehr Storage-Knoten

- → Gesamtkapazität steigt
- → Gesamtdurchsatz steigt
- → Gesamt-IOPS steigen
- → Verfügbarkeit erhöht sich
- → Latenz unverändert

- → Limit: Netzwerktopologie
- → Neuverteilung der Daten erzeugt temporär höhere Last

- Es war einmal eine Open Source Speicherlösung namens Ceph
- Mit verschiedenen Komponenten, die zu kennen sich lohnt
- Alle wollen Software Defined Storage anstelle von „Legacy“ Storage-Arrays
- und hatten viele Fragen zu Konfigurationsmöglichkeiten
- und lernten, mit den richtigen Fragen Proof-of-Concepts zu bauen und auszubauen

# Wie setze ich einen Ceph-Cluster zusammen?

- Was soll der Cluster tun?
- Budget abschätzen
- Pilotprojekt bauen in ~10% der Zielgröße
- Stellschrauben verändern, bis die Performance passt
- Skalieren durch Hinzufügen weiterer Komponenten
- Nicht von Anfang an perfekt, kann aber über die Zeit wachsen

# Ceph-Clients

# librados

- native Ceph-Objekte
- wird praktisch nicht verwendet
- `rados` als CLI-Tool praktisch für Debugging etc

# RADOS Block Device, RBD

- virtuelle Festplatte
- direkt per Kernel eingebunden
  - /dev/rbd0
- als (Boot-) Image für KVM-VMs mit qemu+rbd
  - libvirt
  - Proxmox
  - OpenStack
  - u.a.
- teilt Gigabyte-großes Image in viele kleine 4MB Ceph-Objekt auf
- Gateway für iSCSI möglich

# Rados-Gateway, S3, Swift

- Object-Store mit HTTP-API
- weitgehend kompatibel zu Amazon S3
- Support für Object Locks
  - WORM-ähnliches S3-Feature
  - interessant für Archive / Backup
- Nutzt mehrere Pools
  - schneller SSD-Pool für Index
  - platzsparender erasure coded HDD-Pool für Daten

# CephFS

- POSIX-kompatibles verteiltes Dateisystem
- Client-Implementierung nur für Linux
  - FUSE
  - Kernel
- Daten auf mehrere Pools aufteilbar
  - schnelle oder langsame Pools für verschiedene Teile des Dateisystems
- Quota, Snapshots
- Dateisystem-Zugriffsrechte bestimmt der Client (!)
  - ähnlich wie NFSv3 oder lokales Dateisystem
- Gateways für SMB (Samba VFS) und NFS (Ganesha FSAL) möglich

# FAQ

- Kann ich mit Ceph meinen 300GB Fileserver ausfallsicher machen?
- Nein.
- Aber wenn Du Ceph brauchst, gibt es nichts anders.

- Es war einmal eine Open Source Speicherlösung namens Ceph
- Mit verschiedenen Komponenten, die zu kennen sich lohnt
- Alle wollen Software Defined Storage anstelle von „Legacy“ Storage-Arrays
- und hatten viele Fragen zu Konfigurationsmöglichkeiten
- und lernten, mit den richtigen Fragen Proof-of-Concepts zu bauen und auszubauen
- und lebten glücklich bis ans Ende ihrer Tage

- Dank an Tim Serong <tserong@suse.com> und Lars Marowsky-Brée <lmb@suse.com>

Robert Sander <r.sander@heinlein-support.de>

# Soweit, so gut.

# Gleich sind Sie am Zug:
# Fragen und Diskussionen!

**Wir suchen:**
Admins, Consultants, Trainer!

**Wir bieten:**
Spannende Projekte, Kundenlob, eigenständige Arbeit, ein tolles Team, Work-Life-Balance

...und natürlich: Linux, Linux, Linux...

**http://www.heinlein-support.de/jobs**

# Heinlein Support hilft bei allen Fragen rund um Linux-Server

## HEINLEIN AKADEMIE

Von Profis für Profis: Wir vermitteln die oberen 10% Wissen: geballtes Wissen und umfangreiche Praxiserfahrung.

## HEINLEIN HOSTING

Individuelles Business-Hosting mit perfekter Maintenance durch unsere Profis. Sicherheit und Verfügbarkeit stehen an erster Stelle.

## HEINLEIN CONSULTING

Das Backup für Ihre Linux-Administration: LPIC-2-Profis lösen im CompetenceCall Notfälle, auch in SLAs mit 24/7-Verfügbarkeit.