CernVM-FS

Ein skalierbares, zuverlässiges, wartungsarmes Software-Verteil-System

Oliver Freyermuth

Universität Bonn, Physikalisches Institut
freyermuth@physik.uni-bonn.de

3. Juli 2019

UNIVERSITÄT BONN

Verteilen von Software / Containern

- Üblicherweise viele Millionen (kleine) Dateien
- Viele Versionen parallel vorhalten
- Hoher Duplikationsfaktor
- Read-only

Probleme für Dateisysteme

- Viele Metadatenzugriffe in „Bursts“
- Deduplikation
- Cluster-Dateisysteme sind hier besonders problematisch...

Ansatz CernVM-FS

- Read-only Dateisystem
- Content-addressable Storage (komprimiert, dedupliziert)
- HTTP-basierte Verteilung von Content:
  - Stratum-Modell
  - aggressives Caching
  - Einsatz klassischer Squid-Proxies etc. möglich
  - Load-balancing und HA „einfach“
- FUSE-Clients mit lokalen On-Access-Caches
- Versionierung / Tagging
- Repositories signiert (X.509)

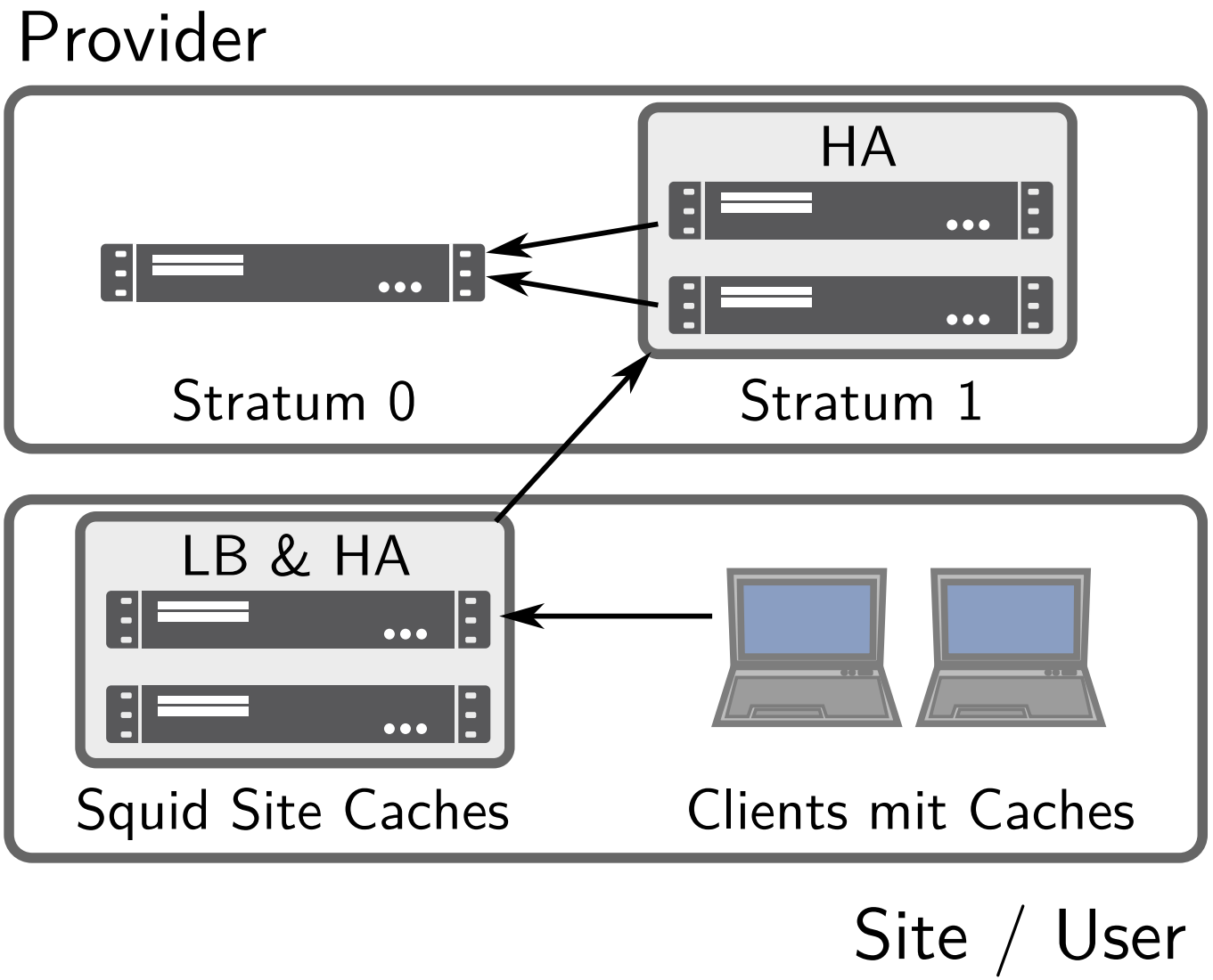
CVMFS „unter der Haube“

CAS: Content-addressable storage

```
data/00/...
data/01/00036a7e9adb53bfcfd48e6370ee00885cbae5-shake128
        0037c833bf4994c6b935ac7824014074e37055-shake128
        00525794543314af488161b3f3b60fa6f0bfde-shake128
        ...
data/02/...
...
data/ff/...
```

- Chunks, benannt nach Hash des komprimierten Inhalts
  Datei-Chunks, „volle“ Dateien, Verzeichnisse, Links etc.
- Client lädt Repository-Manifest mit Key-Value-Pairs, enthält etwa Hashes der Kataloge, Revision, TTL,. . .
- Client lädt pro Ordner-Baum eine SQLite-Datenbank (**Katalog**) mit allen Metadaten, Hashes etc.

Schematischer Aufbau CVMFS (klassisch)



Änderungs-Workflow

1. Transaction wird gestartet (RW-OverlayFS auf RO-CVMFS)
2. Publizieren der Transaction committed neue Chunks und Kataloge
3. Stratum 1 ziehen Änderungen komplett
4. Clients synchronisieren Kataloge, Änderungen bei Bedarf

Varianten auf Client-Seite

FUSE

- **Klassisch:** FUSE Client, lokaler Cache
- FUSE Client mit gesharetem oder preloadetem Cache (für offline-Einsatz)

Ohne privilegiertes `mount`

- Moderne Kernel (z.B. CentOS 8):
  Unprivilegiertes FUSE / unprivilegierte User Namespaces
- Shrinkwrap:
  Per Hardlinks abgebildetes CVMFS, kein Client nötig

Varianten auf Server-Seite

- CVMFS Ingest: Direktes „ingest“ z.B. von Container-Tarballs (ohne OverlayFS)
- CVMFS Gateway: Mehrere Committer pushen Änderungen über eine Gateway-Maschine,
  statt Stratum 0 direkt zu nutzen
- Storage auch auf S3 möglich
- Mit S3 und „CVMFS-Server-Container“ Betrieb in künftigem Release ohne weitere Server-Infrastruktur möglich

Wofür wir CVMFS nutzen...

- Versorgung mit Software vom CERN
- Eigene Software mit `lmod` environment modules:
  2 905 403 Einträge in CVMFS-Katalogen
  Genutzt auf Desktops, Cluster-Workernodes
- Eigene Container von 3 Linux-Distributionen, für 30 Tage aufbewahrt:
  15 088 054 Einträge in CVMFS-Katalogen
- Gesamt ca. 55 GB

Weitere Infos

- Dokumentation:
  `https://cvmfs.readthedocs.io/en/stable/`
- CernVM-Workshop 2019:
  `https://indico.cern.ch/event/757415/`

Danke

für die Aufmerksamkeit!