Überblick Ceph / CephFS

Einsatz in einem HTC Cluster

Oliver Freyermuth, Thomas Schneider

Universität Bonn
freyermuth@physik.uni-bonn.de

Ruhr-Universität Bochum
Thomas.Schneider-q2p@ruhr-uni-bochum.de

3. Juli 2019

RUB

Physikalisches Institut
UNIVERSITÄT BONN

Ceph-Design: Verteilung von Objekten

Placement Groups: Gruppe von Objekten

- Reduziert Verwaltungsaufwand für das Mapping zum Ablageort (Millionen Objekte!)

CRUSH

- **C**ontrolled **R**eplication **U**nder **S**calable **H**ashing
- schneller Algorithmus, um Objekte auf PGs zu verteilen
- Pseudo-Random — Gleichverteilung über Statistik
- Ausformulierte Regeln von Failure-Domains

Beispiel für Failure Domain

data-center: Bei 3 Data Centers und 3 Replikas ist eine Kopie in jedem Data Center (auf irgendeinem Server)

Ceph-Design: Verteilung von Objekten

Balancing

- statistische Verteilung ist nicht uniform (da nicht ∞ PGs)
- Balancing über „Gewichte“ möglich
- direkte Zuordnung der PGs über „Upmap“ möglich
- Automatischer Balancer ist Teil von Ceph
- Damit kann ein Cluster beliebig wachsen / schrumpfen

„Selbstheilung“

Fällt eine Disk / Server / Data Center aus, werden die PGs remapped und versucht, die Failure-Domain Vorgabe zu erfüllen.

Neu ab Nautilus: Disk-Fehlererkennung per SMART, präventives „Leerräumen“.

Ceph-Dienste

MON (ungerade Anzahl, z.B. 3)

- Hält Konfiguration und Cluster-Map (OSD / MON Maps, ...), Quorum mit ungerader Anzahl nötig
- Erster Ansprechpartner für alle Dienste und Clients

OSD (pro Disk)

- Hält Placement-Groups mit Objekten (üblicherweise 100 PGs)
- Kümmert sich um den Storage / Scrubbing / int. Metadaten
- Primärer OSD (peert mit anderen OSDs der gleichen PG) ist Ansprechpartner der Clients, z.B. bei Erasure Coding: sammelt Daten ein, verteilt Shards

MGR (min. 1)

Monitoring und Management (alles, was nicht für den Normalbetrieb essentiell ist)

Ceph: Was landet auf den Disks / Hardwaredesign?

- Backend-Storage (Bluestore) ist Copy-on-Write (COW) direkt „roh“ auf den Block Devices (eigenes FS)
- Durch COW auch Features wie Snapshotting, RBD Cloning etc. implementiert
- Interne Metadaten in RocksDB (idealerweise SSD / NVMe)
- WAL für RocksDB kann separat abgelegt werden
- RocksDB / Daten on-the-fly komprimierbar (Snappy, zlib, zstd)
- Checksummen (crc32c, xxhash) erlauben Scrubbing, Fehlererkennung
- Viel RAM für Caching / Heilung (ca. 4 GB per OSD)

Hinweis für Ceph „early adopter“

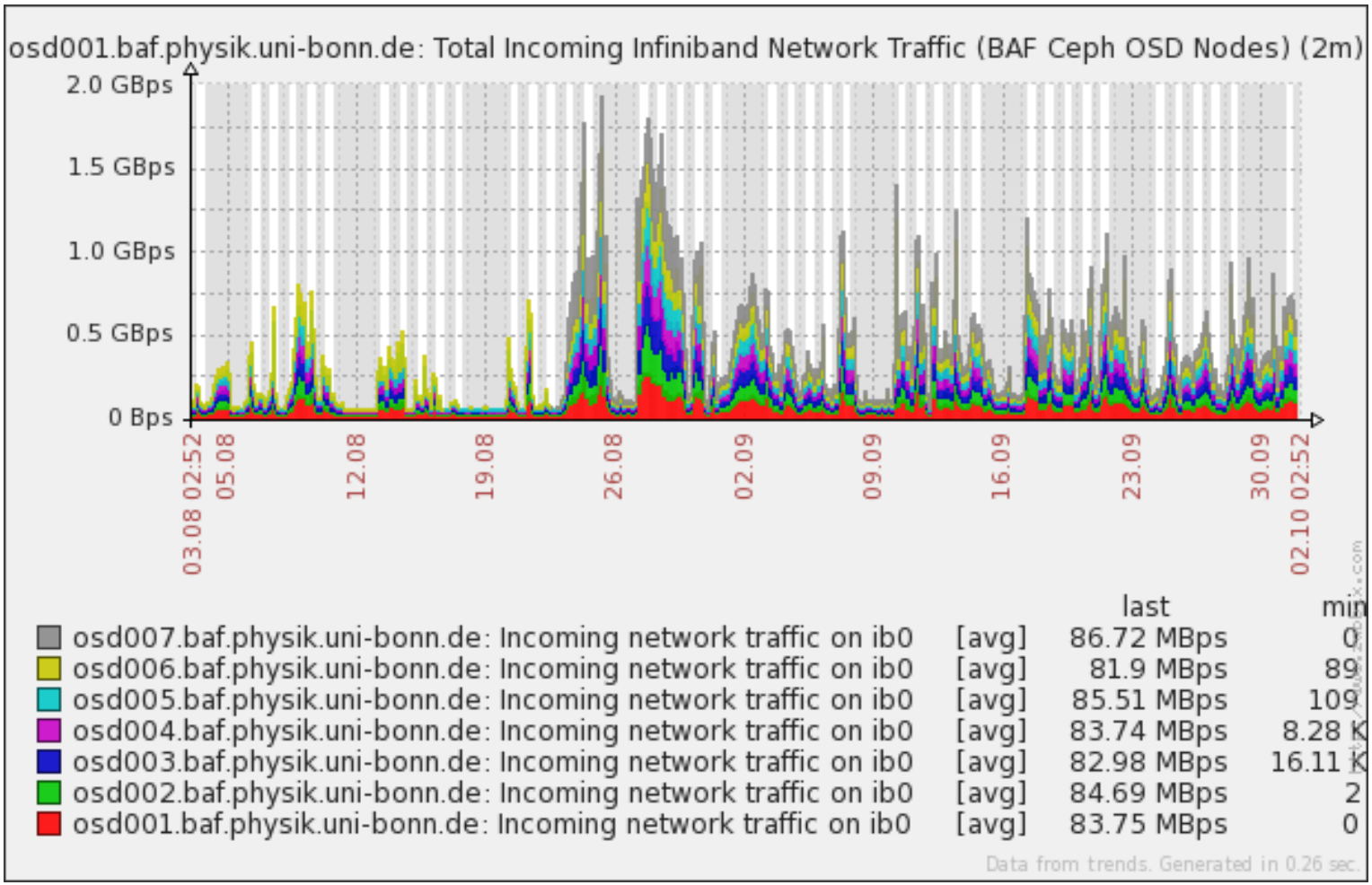
ehemals Filestore im Einsatz (Ablage etwa auf XFS), zusätzlich fast sämtliche Daten vorher auf SSDs geschrieben (write duplication!)

CephFS

- POSIX Dateisystem (inkl. Support für xattr)
- Alle Features, die Ceph bietet, nutzbar (*Snapshotting exp.*)
- Zusätzlich zu Ceph-Diensten: Metadaten-Service (MDS)
  - benutzt RADOS für Storage ($\Rightarrow$ komplett „mobil“)
  - MDS kein Single-Point-of-Failure (Standby-Replay möglich)
  - MDS kann auch Last verteilen (Multi-MDS)
- Wichtiger Unterschied zu Netzwerk-FS wie NFS/SMB:
  Sehr granulares Locking: Clients halten *Capabilities* an Inodes
- Clients: FUSE, Kernel-Client (Upstream)
- ACLs (derzeit nur FUSE) und Quotas (FUSE / ab Kernel 4.17)
- Export per NFS Ganesha möglich
- Prinzipiell auch RDMA-Support
- Ab Nautilus: LazyIO (experimentell) möglich:
  Applikationen müssen Cache-Kohärenz sicherstellen
- Ab Nautilus: Scrubbing der Metadaten möglich

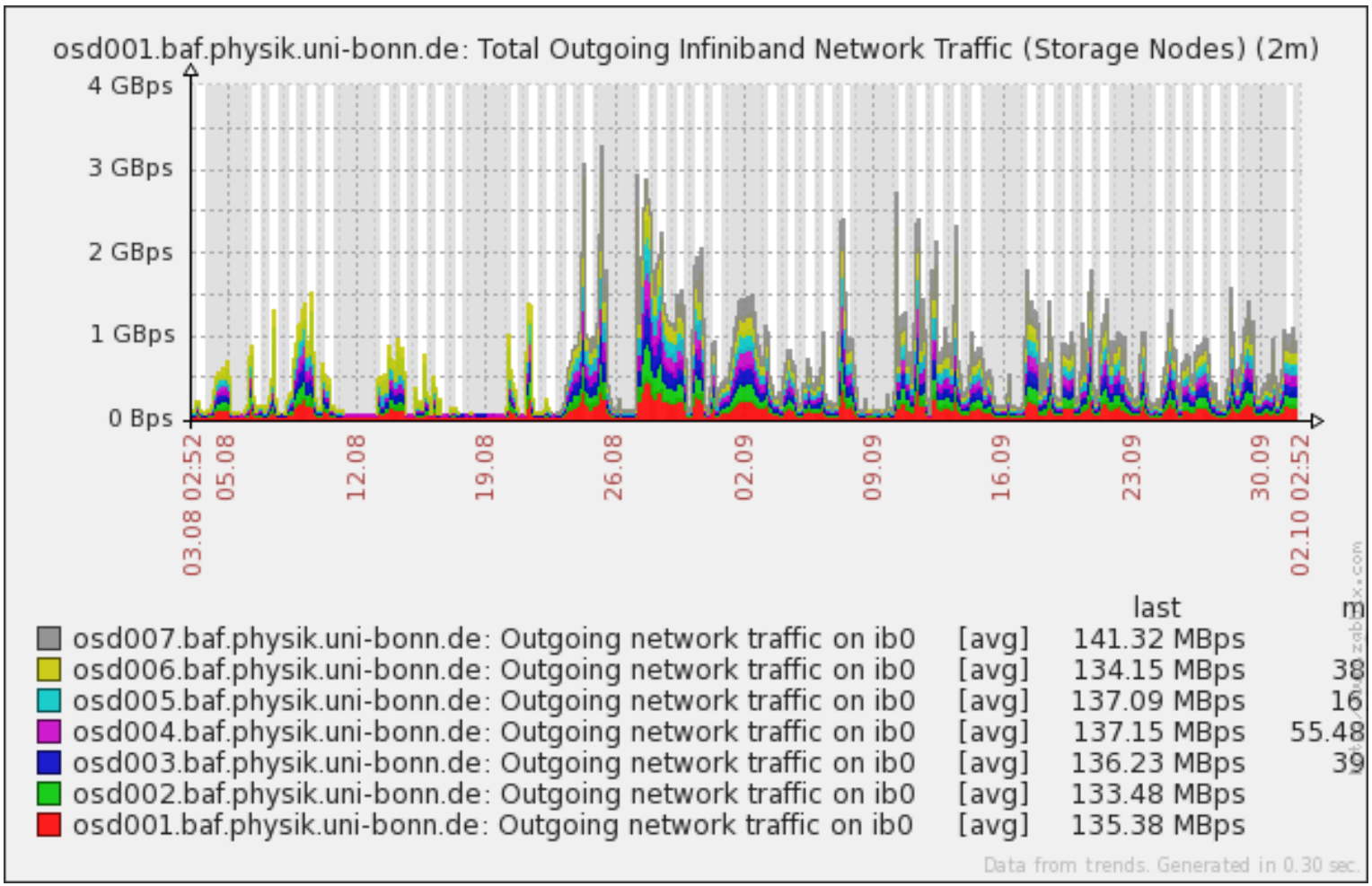
CephFS: Performance

Durchsatz InfiniBand FDR mit IPoIB



CephFS: Performance

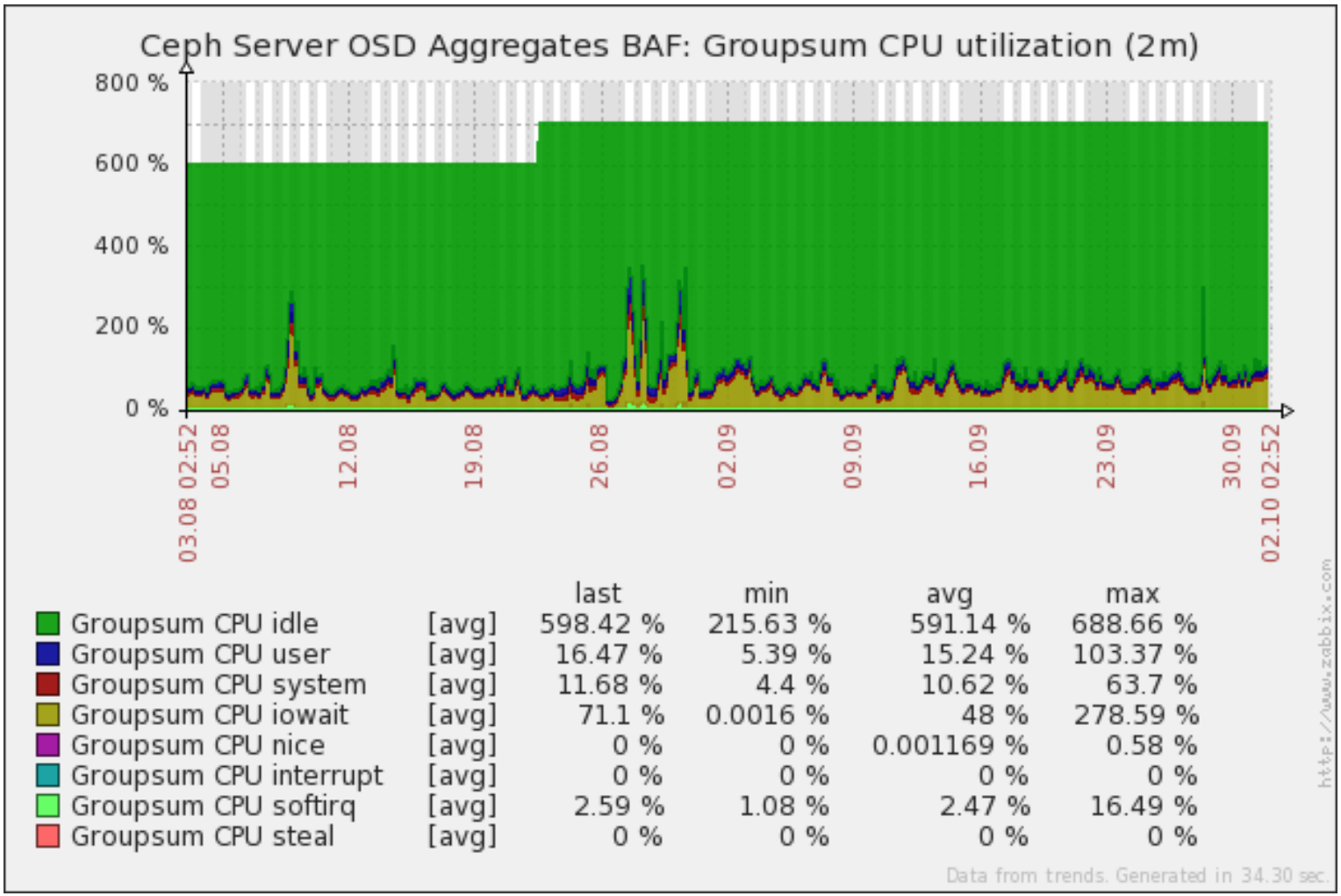
„Lesen“ erzeugt auch eingehenden OSD-Verkehr (EC)



UNIVERSITÄT BONN

CephFS: Performance

Disks gut ausgenutzt (`iowait`), inzwischen fast $\times 2$ (RocksDB)



Weitere Infos

- CephFS:
  `http://docs.ceph.com/docs/master/cephfs/`

Danke

für die Aufmerksamkeit!