



Parallele Dateisysteme

Proseminar: Speicher- und Dateisysteme
Phillipp Jäger

- Einleitung
 - verteilte & parallele Dateisysteme
- parallele Ein- und Ausgabe
 - allgemeine Architektur
 - Parallelisierung
- moderne parallele Dateisysteme
- Technologie der Speichersysteme
- Zusammenfassung

Einleitung

■ Gemeinsamkeiten

- persistente Speicherung
- Verteilung der Daten auf mehrere Speicherknoten (*storage nodes*)
- client- und serverseitige Implementation
- meist räumliche Trennung von *Client* und *Server*
- Verteilung ist komplex und kostet Zeit

■ verteilte Dateisysteme

- kein globaler *namespace*
- Metadaten dezentral gespeichert

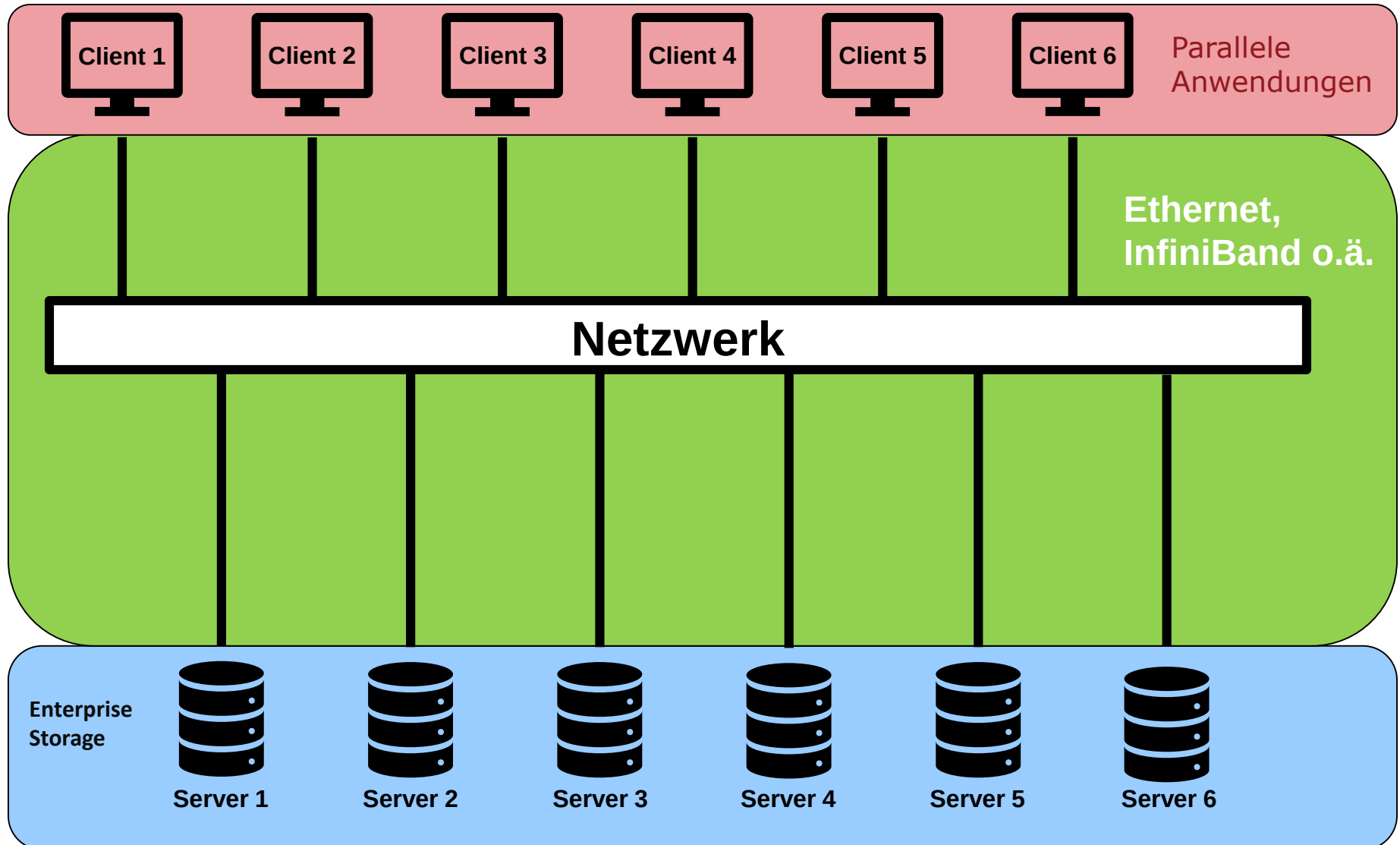
■ parallele Dateisysteme

- global geteilter *namespace* für Dateien und Verzeichnisse
- paralleler Zugriff mehrerer *compute nodes* auf mehrere *storage nodes*
- Einsatz im *High Performance Computing* (HPC)
- parallele Anwendungen auf *Clients*



Parallele Ein- und Ausgabe

allgemeine Architektur



parallele Anwendungen

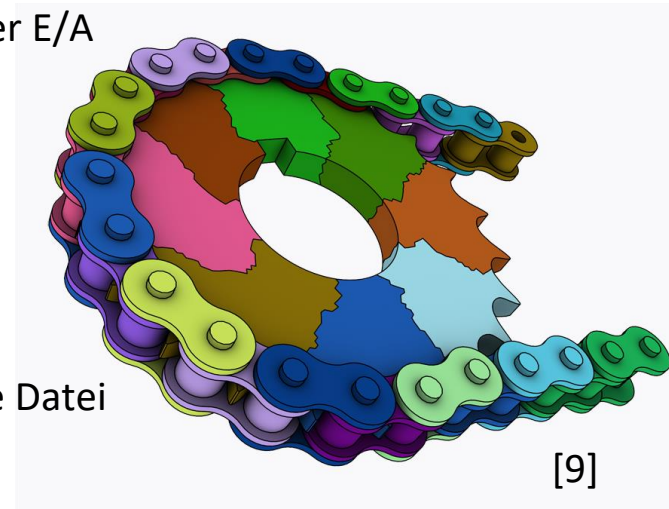
■ domain decomposition methods (DDM)

- Unterteilung mathematischer Probleme in *subdomains*
- *subdomains* sind (größtenteils) unabhängig voneinander
 - ermöglicht parallele Prozesse

■ Ein- und Ausgabe Parallelisierung

- Dateisystem übernimmt zum Teil die Parallelisierung der E/A
- Workload-Manager verteilt Prozesse auf *nodes*
- jeder Prozess hat seinen eigenen E/A-Kanal
- Daten können geschrieben werden in:

- *shared file*
 - $n:1$ Beziehung
 - alle Prozesse schreiben auf eine gemeinsame Datei
 - kann zu *lock contention* führen
- *file per process / independent file*
 - $n:n$ Beziehung
 - jeder Prozess schreibt auf seine eigene Datei
 - Daten werden über viele Dateien verteilt



parallele Anwendungen

■ Ein- und Ausgabe Parallelisierung

– striping

- Funktion:

- Round-Robin-Verfahren (z.B. GPFS, Lustre)

- Parameter:

- » Größe (*stripe size*)

- » Anzahl der *targets* auf die die Daten verteilt werden sollen (*stripe count*)

- Vorteile:

- Datei kann größer sein als Speicherplatz eines *storage nodes*

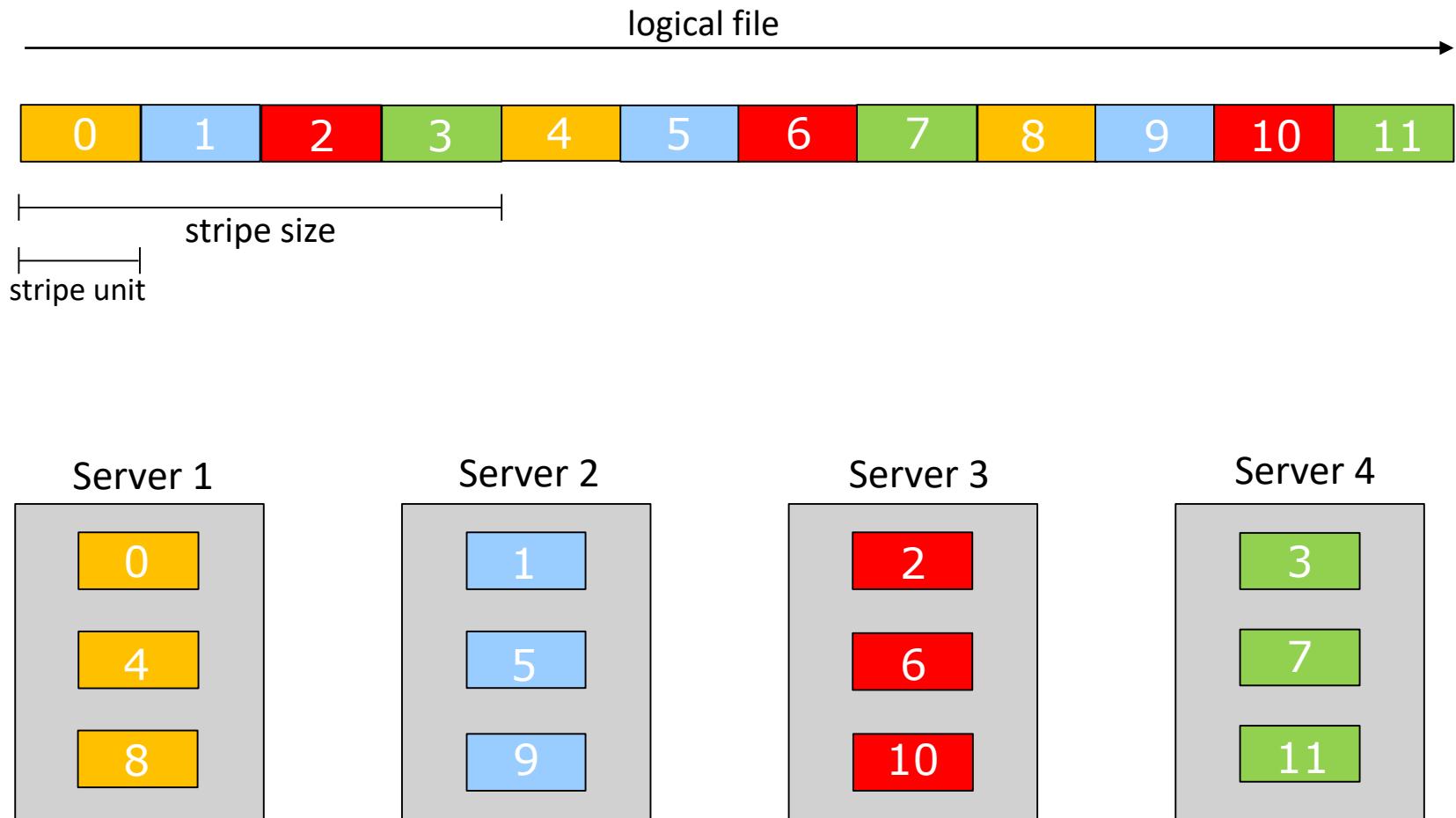
- verhindert, dass eine Datei den gesamten Server einnimmt

- Nachteil:

- höheres Dateiverlustrisiko

- Stripes können *unaligned* (unbündig) sein

Beispiel: Striping [1]



Beispiel: Striping [2]

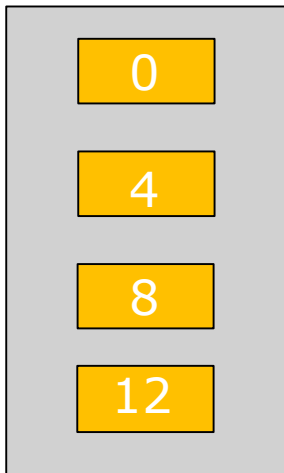
Client 1:



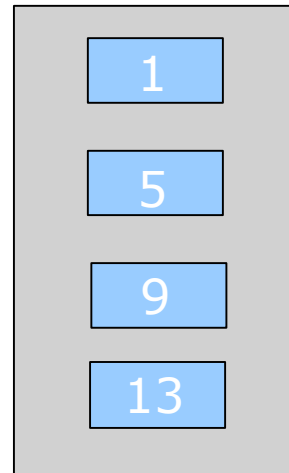
Client 2:



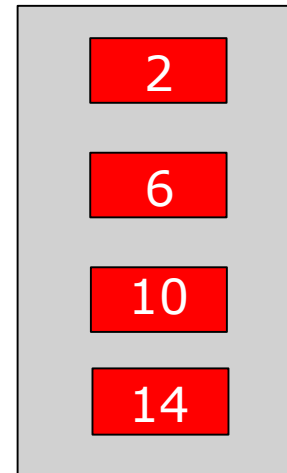
Server 1



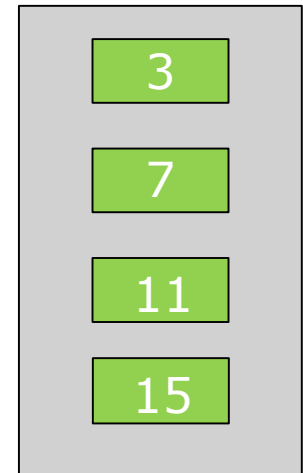
Server 2



Server 3

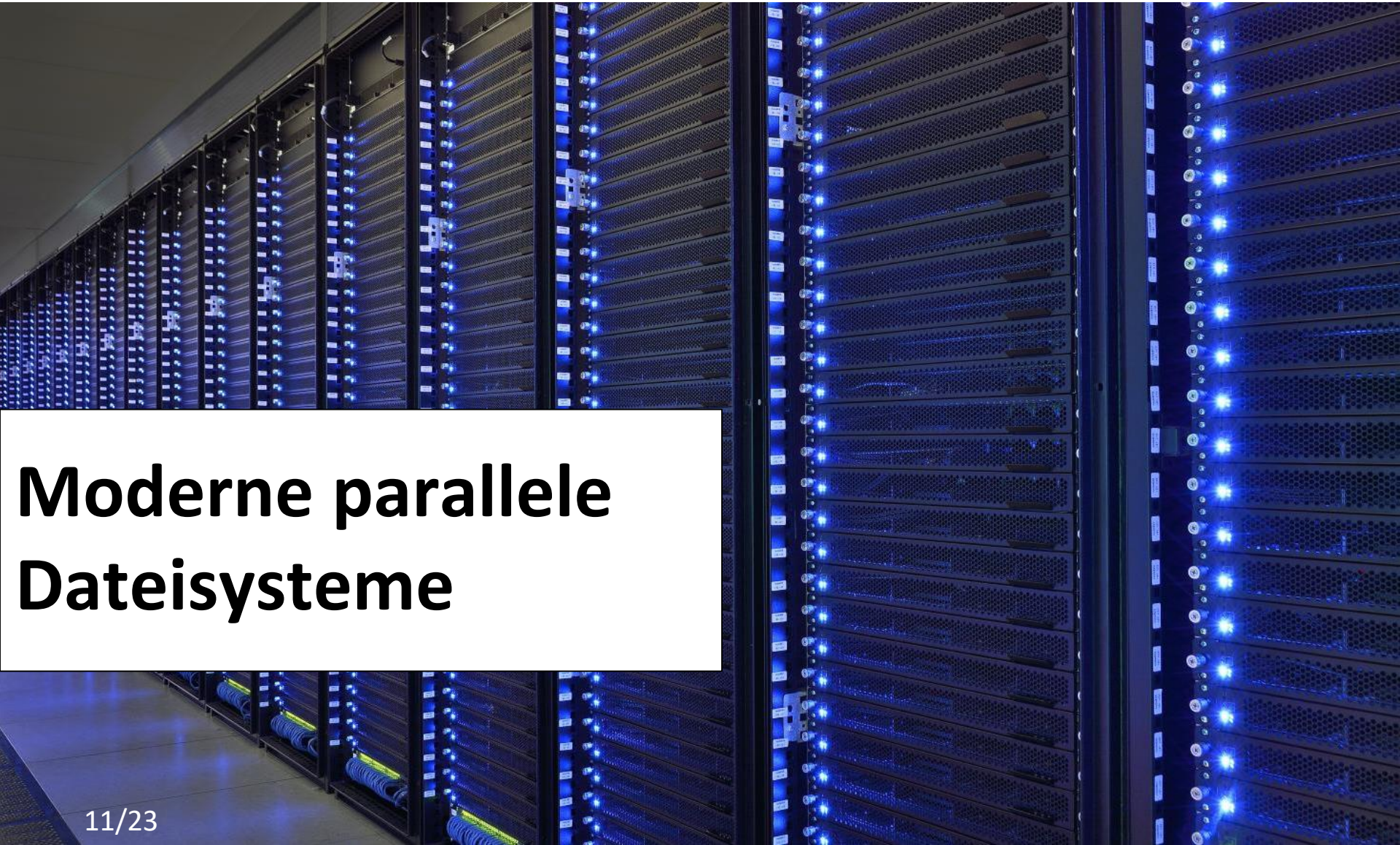


Server 4



Metadatenverwaltung

- **strukturelle Metadaten**
 - Informationen über den *Container* bzw. das Verzeichnis
- **deskriptive Metadaten**
 - beschreibt den eigentlichen Dateiinhalt
 - z.B. Erstellungsdatum und Dateirechte
- **zentrale Speicherung**
 - Daten auf wenigen zentralen Servern gespeichert
 - Problem:
 - hohe Auslastung einzelner Server; möglicher Flaschenhals
 - *single point of failure*
 - linear ansteigender *overhead* bei größerer Anzahl von *nodes*
- **dezentral Speicherung**
 - Daten und Metadaten werden gemeinsam gespeichert (z.B. GPFS)



Moderne parallele Dateisysteme

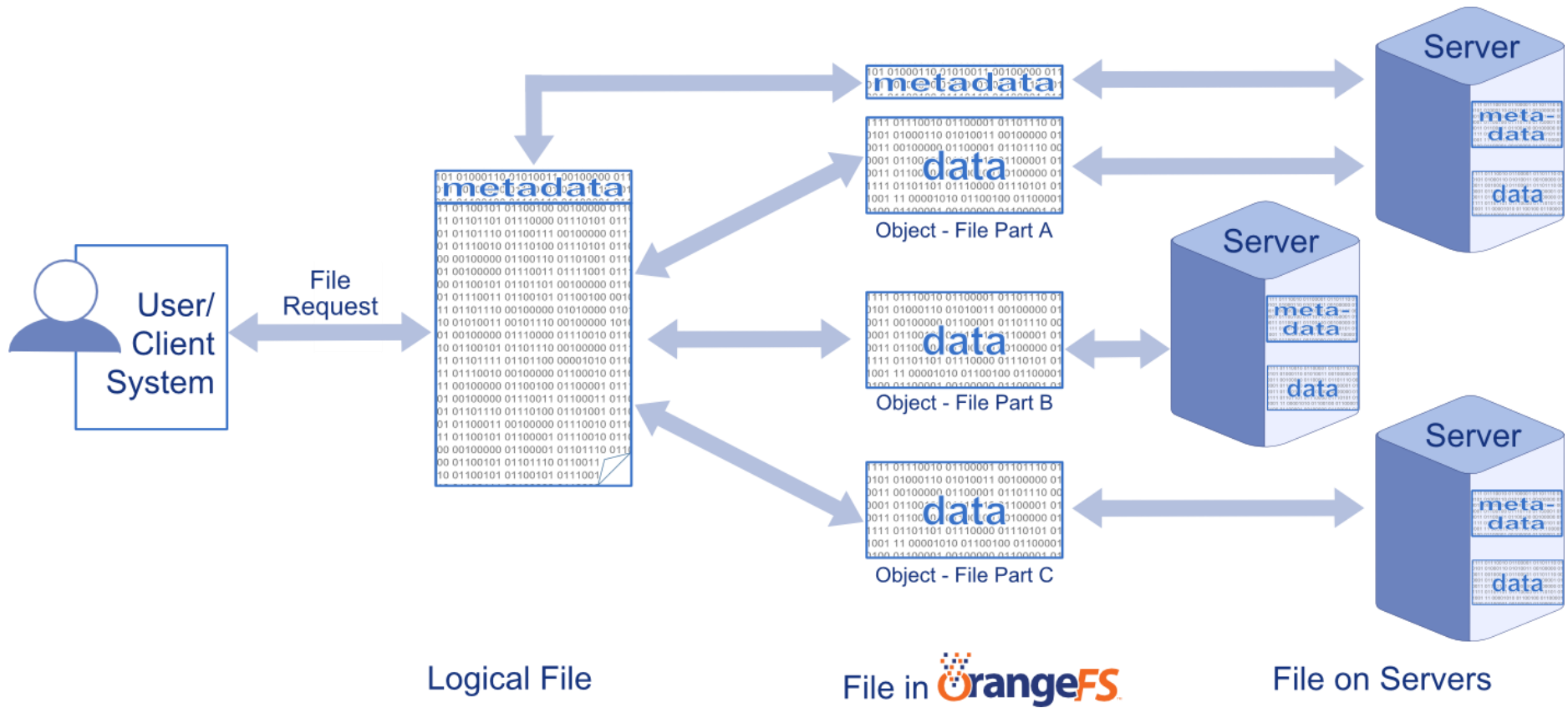
Aktuelle Dateisysteme

	BeeGFS	CephFS	Spectrum Scale /GPFS	Lustre	OrangeFS
Striping	Ja	Ja	Ja	Ja	Ja
RDMA	Ja	Ja	Ja	Ja	Ja
Metadataserver	Metadata + Manager	Ja	Metadata pool optional	Ja	Ja
Open Source	Client: Ja Server: EULA	Ja	Nein	Ja	Ja
POSIX	Ja	Ja	Ja	Ja	Ja

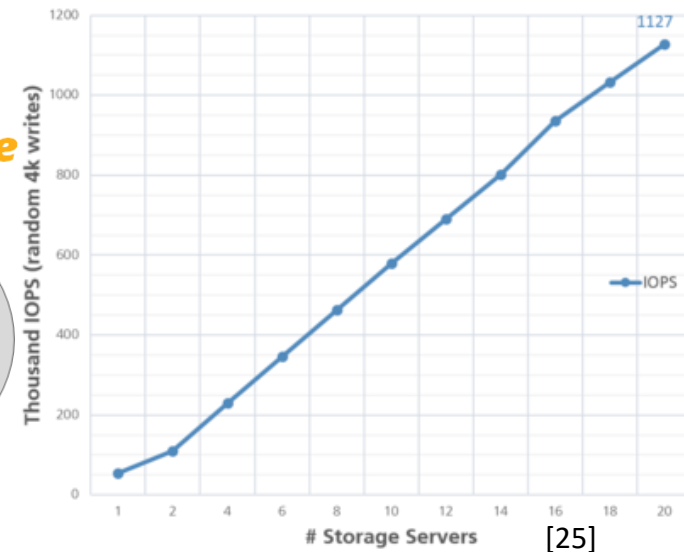
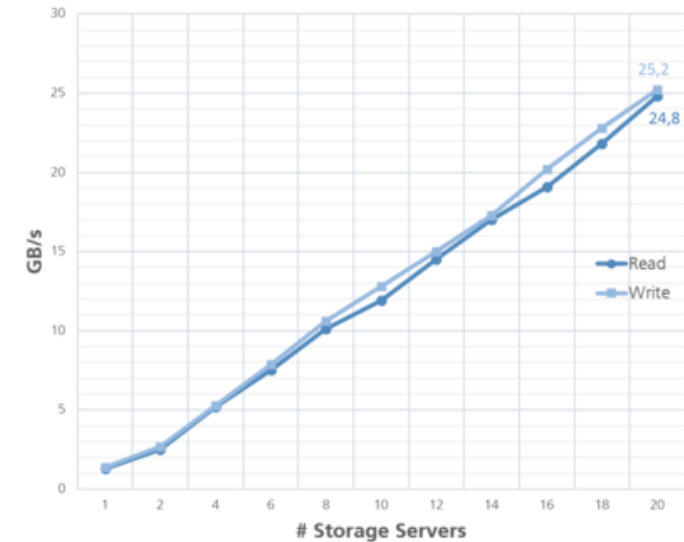
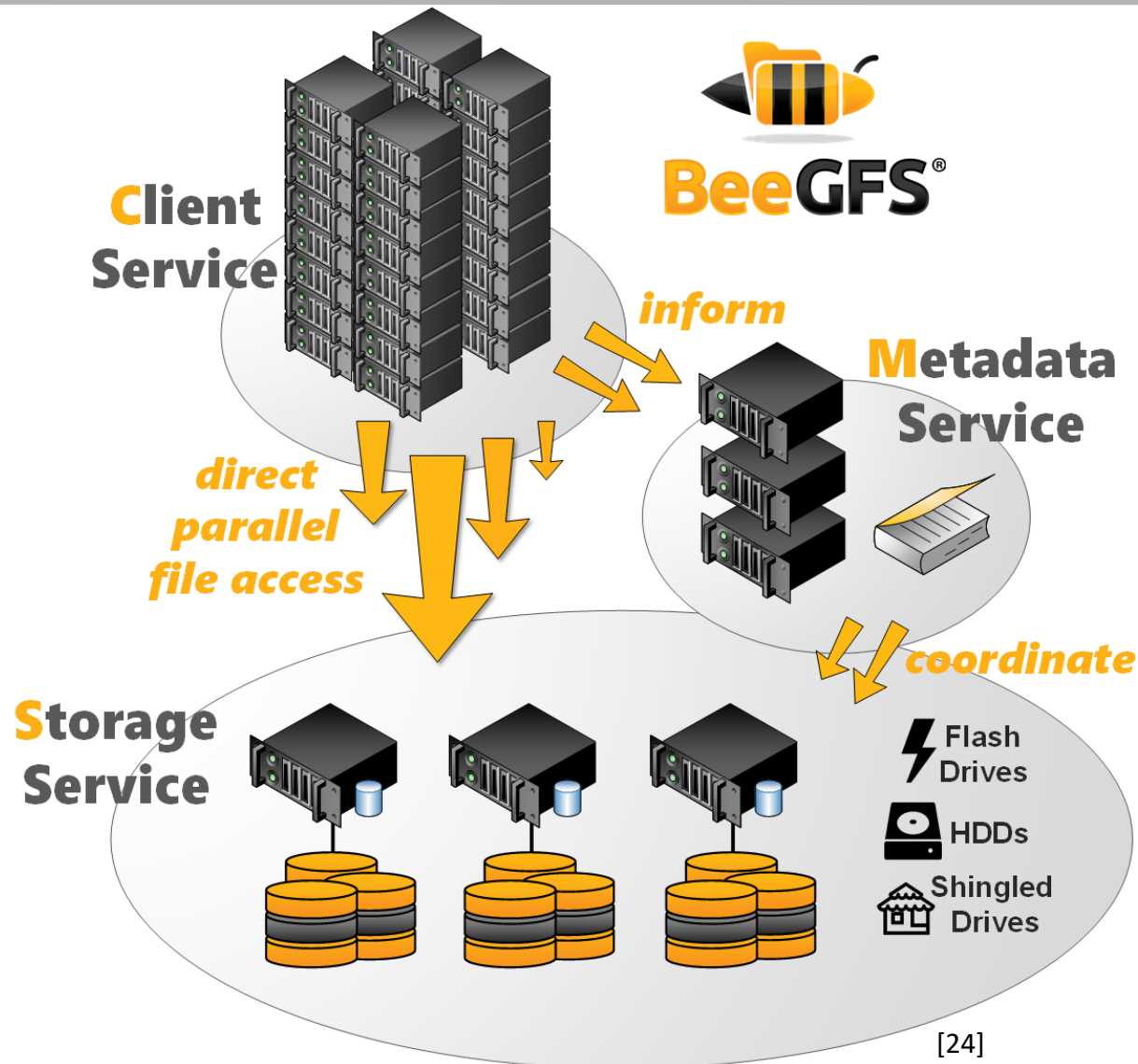
■ POSIX-Konformität

- wird umgesetzt mit Distributed Lock Manager (DLM)
- suboptimal für parallele Anwendungen
- viele wissenschaftliche Programme benötigen Rückwärtskompatibilität

Übersicht: OrangeFS



Übersicht: BeeGFS





Technologie der Speichersysteme

Virtual Institute for I/O Top 5

#	information							io500		
	institution	system	storage vendor	filesystem type	client nodes	client total procs	data	score	bw	md
									GiB/s	kIOP/s
1	JCAHPC	Oakforest-PACS	DDN	IME	2048	16384	zip	137.78	560.10	33.89
2	Korea Institute of Science and Technology Information (KISTI)	NURION	DDN	IME	2048	4096	zip	160.67	554.23	46.58
3	KAUST	ShaheenII	Cray	DataWarp	1024	8192	zip	77.37	496.81	12.05
4	Oak Ridge National Laboratory	Summit	IBM	Spectrum Scale	504	1008	zip	366.47	88.20	1522.69
5	University of Cambridge	Data Accelerator	Dell EMC	Lustre	528	4224	zip	158.71	71.40	352.75

#	information							io500		
	institution	system	storage vendor	filesystem type	client nodes	client total procs	data	score	bw	md
									GiB/s	kIOP/s
1	Oak Ridge National Laboratory	Summit	IBM	Spectrum Scale	504	1008	zip	366.47	88.20	1522.69
2	Korea Institute of Science and Technology Information (KISTI)	NURION	DDN	IME	2048	4096	zip	160.67	554.23	46.58
3	University of Cambridge	Data Accelerator	Dell EMC	Lustre	528	4224	zip	158.71	71.40	352.75
4	JCAHPC	Oakforest-PACS	DDN	IME	2048	16384	zip	137.78	560.10	33.89
5	WekaIO	WekaIO	WekaIO		17	935	zip	92.95	37.39	231.05

burst buffer

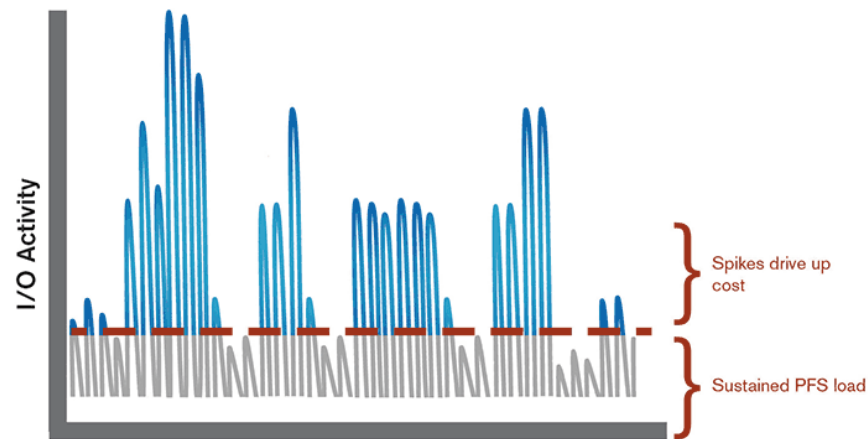
■ Idee:

- E/A-Spitzen von den *compute nodes* abfangen
- Daten im Hintergrund auf den Speicher schreiben

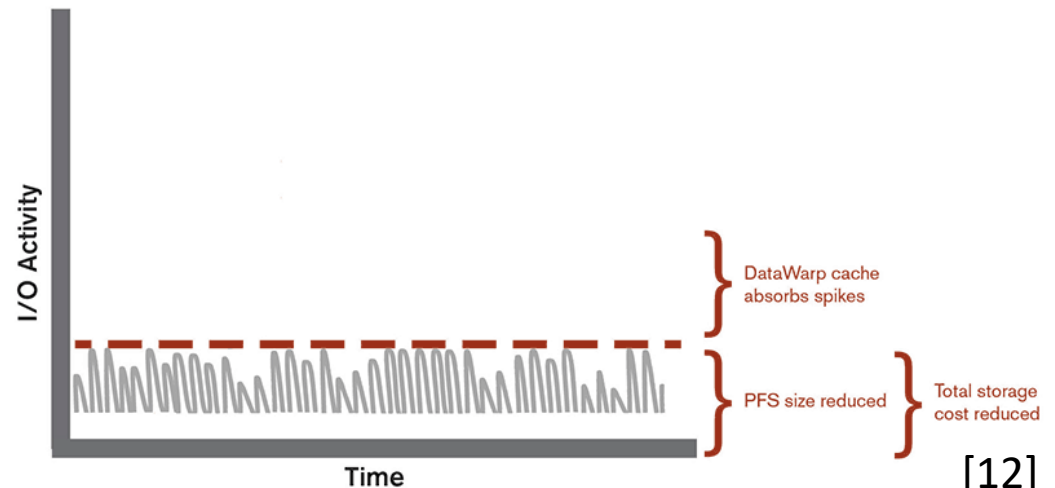
■ Ziel:

- kurze E/A-Phasen
- kürzere Anwendungs-
laufzeit

Before I/O Accelerator



After DataWarp I/O Accelerator



[12]

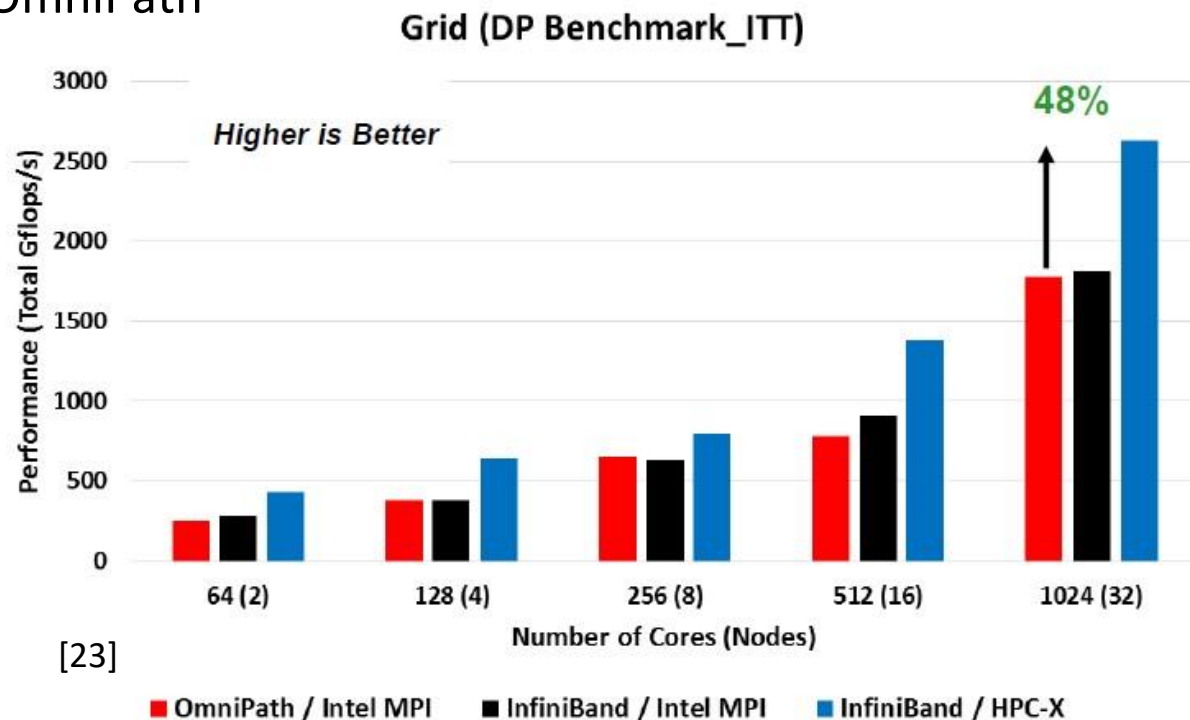
Speicher- und Netzwerktechnologie

■ Speicher

- Tapes zum Archivieren von Daten
- traditionelle Festplatten wegen hoher Ausdauer & Kapazität
- *burst buffer* zur Beschleunigung

■ Netzwerk

- InfiniBand und Intel OmniPath



Beispiel: Oakforest-PACS

supercomputer Oakforest-PACS >		storage system Lustre >	
integrator	Fujitsu	vendor	DDN
installation date	2016	model	SFA14KE
nodes	8208	type	file system
cores	558144	software	Lustre
compute peak	24.91 PFLOPS	installation date	2016
memory capacity	836.09 TiB	net capacity	23.27 PiB
		peak write	465.66 GiB/s
		peak read	465.66 GiB/s
		servers	22
energy >		nodes Lustre >	
max power supplied	2718.70 kW	count	22
benchmarks >		storage system Burst Buffer >	
top500	13.55 PFLOPS	vendor	DDN
green500	4985.69 MFLOPS/W	model	IME14K IME flash
		type	burst buffer
		software	Lustre
		installation date	2016
nodes Oakforest-PACS >		net capacity	0.83 PiB
count	8208	peak write	1452.86 GiB/s
operating system	Linux	peak read	1452.86 GiB/s
processor >		servers	25
vendor	Intel	nodes Burst Buffer >	
model	Xeon Phi 7250 68C	count	25
cores	68		
frequency	1.4 GHz		

[5]

Zusammenfassung

- Parallelisierung auf vielen Ebenen notwendig
 - jeder Prozess verfügt über eigenen E/A-Kanal
 - gleichzeitige Berechnung
 - besser verteilte Rechen- und Speicherauslastung
 - wichtig für zeitkritische Berechnungen (z.B. beim Deutschen Wetter Dienst)
 - POSIX-Konformität stört
- Architektur der Dateisysteme hat einen direkten Einfluss auf die E/A-Leistung
 - zentrale Metadatenserver und E/A oft Flaschenhals des Systems
 - können auf kontinuierliche oder zufällige Zugriffe optimiert werden
 - *burst buffer* zur Beschleunigung

Literatur

1. Titelbild: <https://phys.org/news/2017-01-cluster-gravitational-faster-previous-supercomputer.html> (25.11.2018)
2. Benedikt Johannes Riehm: „Das parallele Dateisystem Lustre“, www.scc.kit.edu/scc/docs/Lustre/Riehm-Lustre.pdf (28.11.2018)
3. https://www.nas.nasa.gov/hecc/support/kb/Lustre-Basics_224.html (29.11.2018)
4. „The Cost of Locks“, <https://web.archive.org/web/20140619205834/http://www.informit.com/guides/content.aspx?g=dotnet&seqNum=600> (4.12.2018)
5. <https://www.vi4io.org/io500/start> (4.12.2018)
6. Joni Laitala: „Metadata Management in Distributed File Systems“, <http://jultika.oulu.fi/files/nbnfioulu-201709092881.pdf> (4.12.2018)
7. Alfio Quarterino: „Introduction to Domain Decomposition Methods“, <http://www1.mat.uniroma1.it/people/garroni/Quarteroni-Lecture-1-DDM.pdf> (5.12.2018)

Literatur

8. Samuel Lang, „Parallel File Systems, IIT Course: Data-Intensive Computing“, September 20, 2010
9. <http://industry.it4i.cz/en/research/basic-research/feti-based-domain-decomposition-methods/> (6.12.2018)
10. Vilobh Meshram et al.: „Can a Decentralized Metadata Service Layer Benefit Parallel Filesystems?“
11. <https://www.ddn.com/products/ime-flash-native-data-cache/> (7.12.2018)
12. <https://www.cray.com/products/storage/datawarp> (7.12.2018)
13. <http://www.cs.fsu.edu/~baker/cop5611.S03/graphics/F6-1.jpg> (7.12.2018)
14. „OrangeFS Concepts Guide“, http://docs.orangefs.com/v_2_9/!SSL!/Printed_Documentation/OrangeFS_2.9_Concepts_Guide.pdf (10.12.2018)
15. <https://indico.fnal.gov/event/12347/contribution/3/material/slides/0.pdf> (10.12.2018)
16. <https://insidehpc.com/2018/04/accelerating-ceph-rdma-nvme/> (10.12.2018)

Literatur

17. <http://docs.ceph.com/docs/mimic/dev/file-striping/> (10.12.2018)
18. <https://ceph.com/community/new-luminous-cephfs-metadata-server-memory-limits/> (10.12.2018)
19. [https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/General+Parallel+File+System+\(GPFS\)/page/Data+and+Metadata](https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/General+Parallel+File+System+(GPFS)/page/Data+and+Metadata) (10.12.2018)
20. <https://www.rup.com/wp-content/uploads/dcd12374dede1.pdf> (10.12.2018)
21. <https://www.thomas-krenn.com/de/wiki/Ceph> (10.12.2018)
22. <https://de.slideshare.net/LarryCover/ceph-open-source-storage-software-optimizations-on-intel-architecture-for-cloud-workloads> (10.12.2018)
23. <https://www.nextplatform.com/2017/11/29/the-battle-of-the-infinibands/> (10.12.2018)
24. <https://www.beegfs.io/content/> (10.12.2018)
25. https://en.wikipedia.org/wiki/BeeGFS#BeeGFS_and_exascale (10.12.2018)