Zukünftige Entwicklungen

Hochleistungs-Ein-/Ausgabe

Michael Kuhn

Wissenschaftliches Rechnen
Fachbereich Informatik
Universität Hamburg

2016-07-01

Exascale

- Supercomputer werden immer leistungsfähiger
  - Speichersysteme werden immer größer
  - Höhere Anforderungen durch hohe Prozesszahlen
- Häufig eine Datei pro Prozess
  - TaihuLight besitzt über 40.000 Knoten
  - Mehr als 10.000.000 Kerne
- POSIX immer noch weit verbreitet
  - Strikte Kohärenzanforderungen
  - Globale Sichtbarkeit aller Änderungen
- Immer weiter wachsende Datenmengen
  - Supercomputer haben mehr als 1 PB RAM
  - Bei 1 TB/s mehr als 15 Minuten für einen Checkpoint

Speicherhierarchie

- Aktueller Stand
  - L1-, L2-, L3-Cache, RAM, SSD, HDD, Band
- Große Latenz-Lücke zwischen RAM und SSD
  - Hohe Leistungsverluste sobald Daten nicht im RAM liegen
  - Overhead durch das Netzwerk ist noch nicht eingerechnet
- Lücke ist auf Supercomputern besonders offensichtlich
  - Knotenlokal im RAM oder im parallelen verteilten Dateisystem
- Neue Technologien sollen die Lücke schließen
  - NVRAM, 3D XPoint etc.
  - Dadurch eventuell auch zusätzliche Stufen im Supercomputer

Speicherhierarchie…

- Bisher ein Faktor von 1.000 zwischen RAM und SSD

Stufe	Latenz
L1-Cache	≈ 1 ns
L2-Cache	≈ 5 ns
L3-Cache	≈ 10 ns
RAM	≈ 100 ns
NVRAM	≈ 1.000 ns
3D XPoint	≈ 10.000 ns
SSD	≈ 100.000 ns
HDD	≈ 10.000.000 ns
Band	≈ 50.000.000.000 ns

Tabelle: Latenzen [5, 4]

Speicherhierarchie…

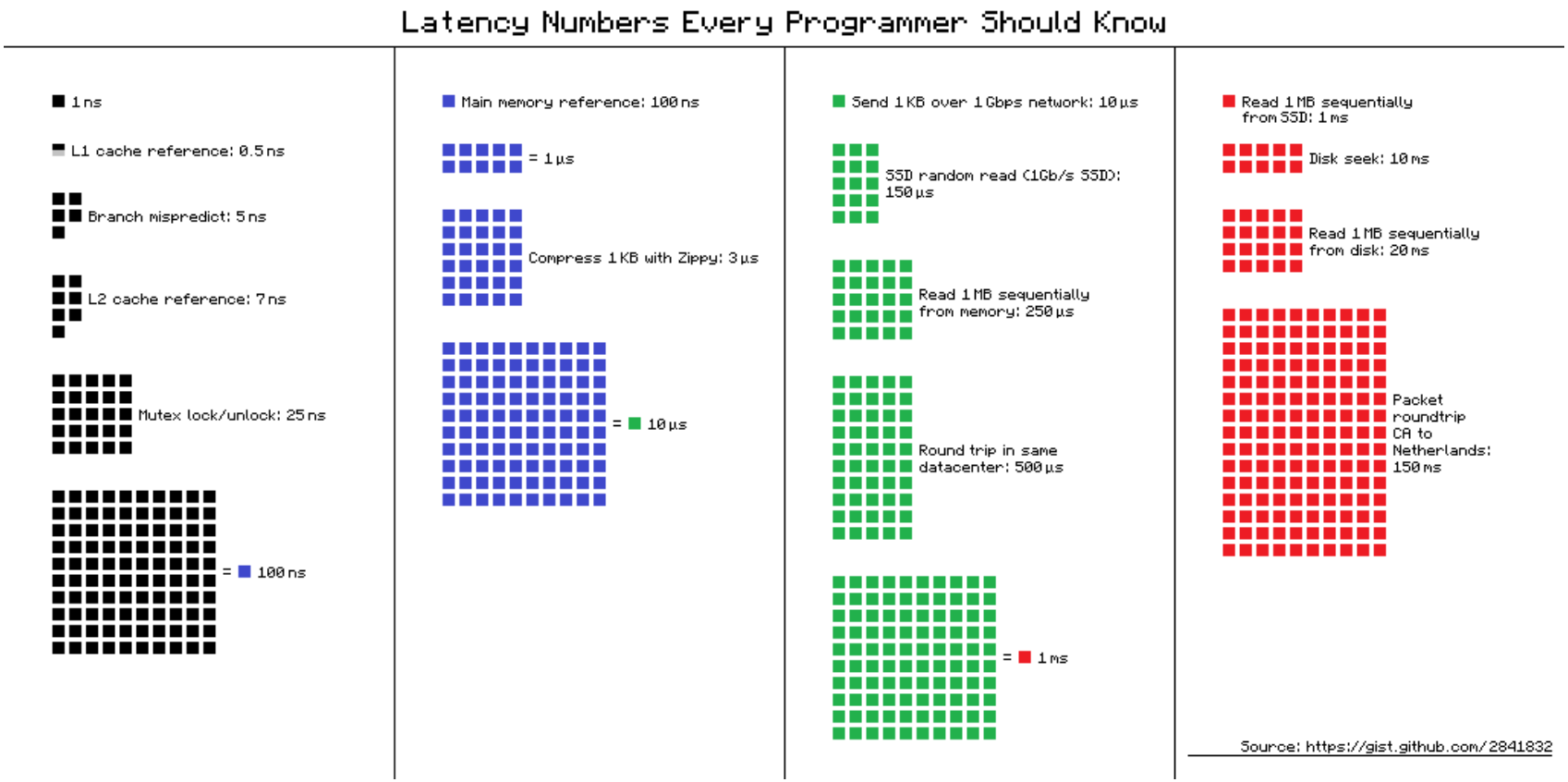


Abbildung: Latenzen [5, 8]

Speicherhierarchie…

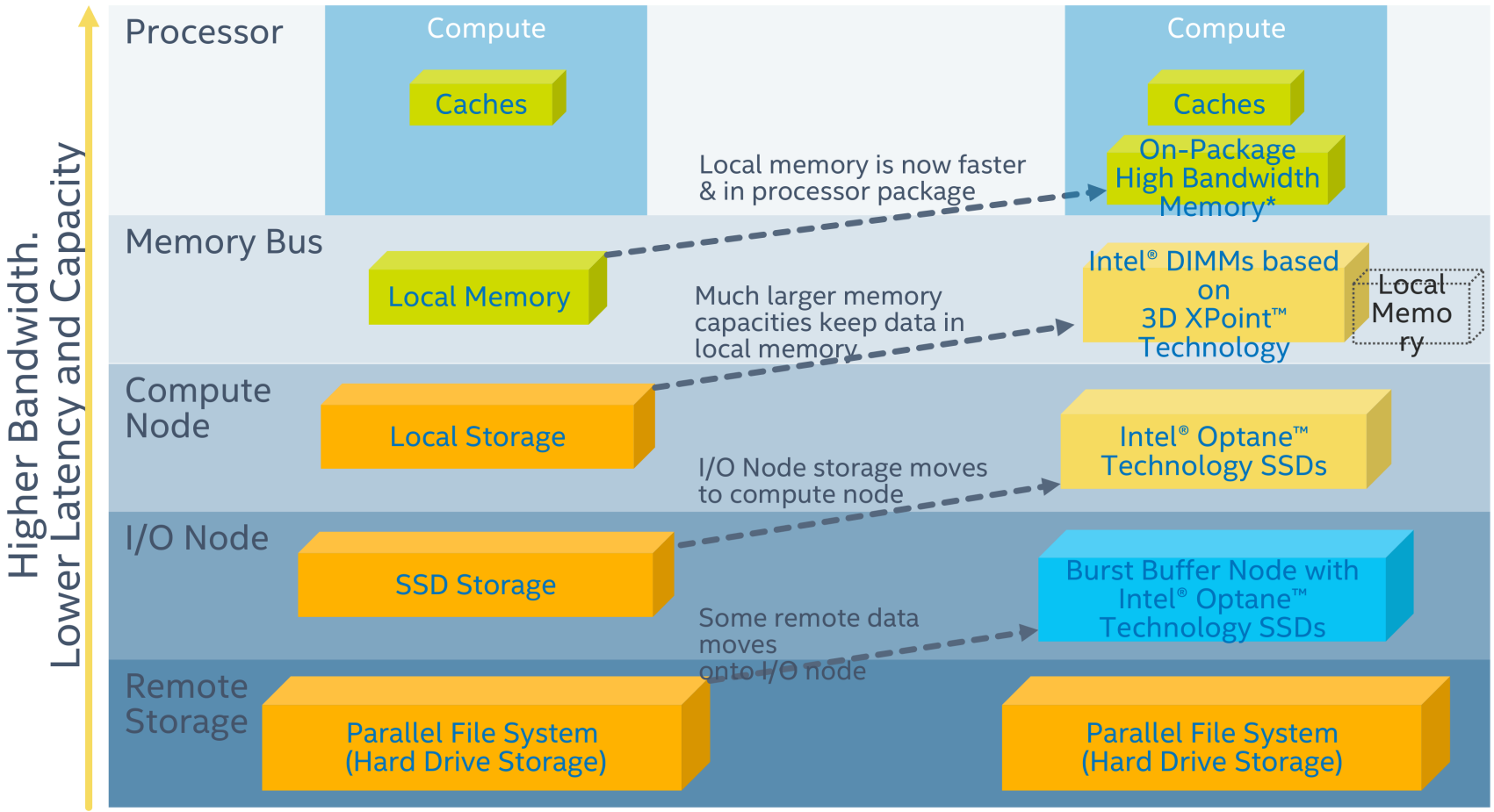
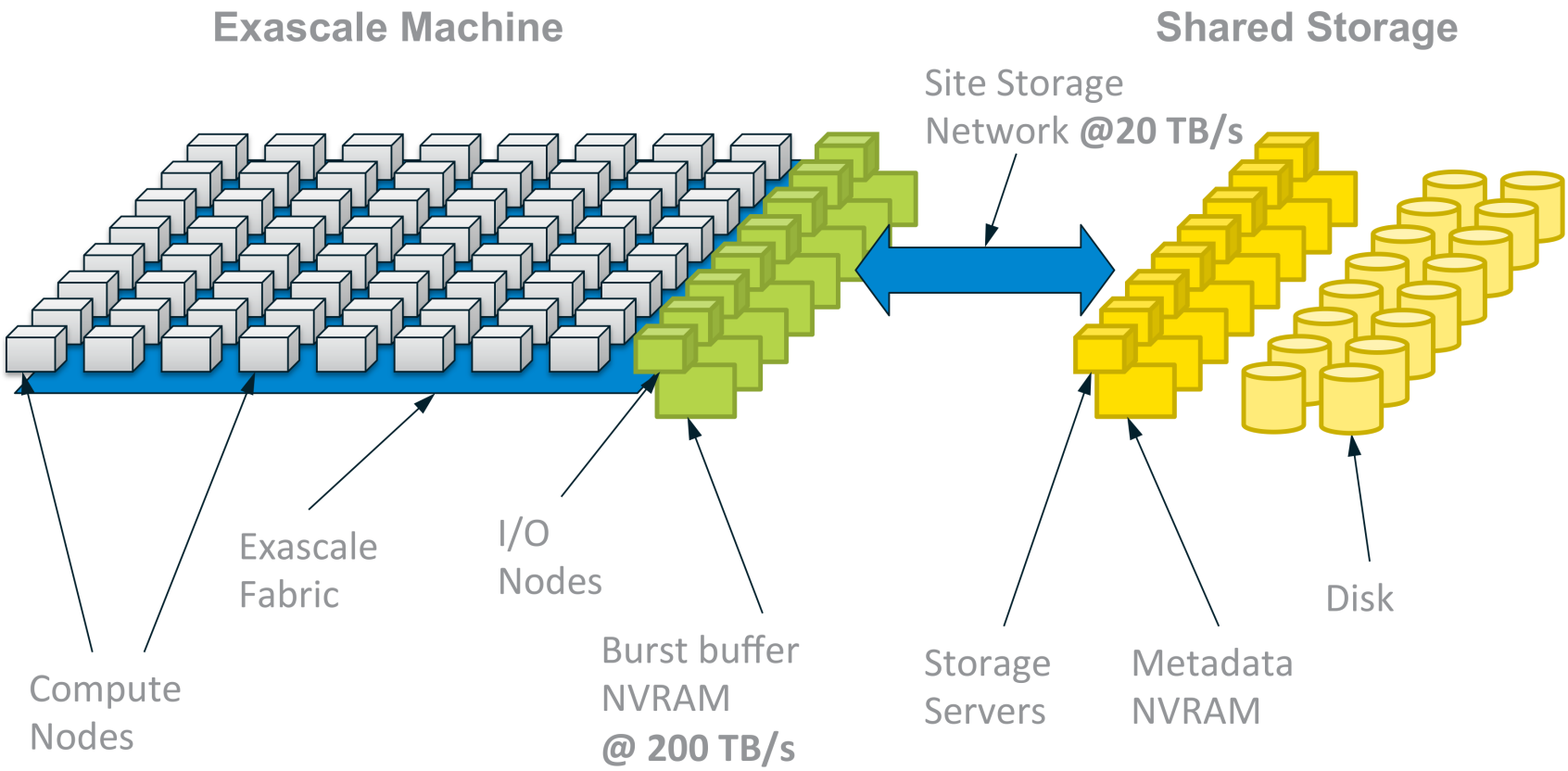


Abbildung: Aktuelle und zukünftige Speicherhierarchie [1]

Exascale [2]

- E/A-Knoten mit Burst Buffern nahe an Rechenknoten
- Langsameres Speichernetzwerk zu Dateisystem-Knoten



Exascale…

- E/A-Knoten können zusätzliche Aufgaben übernehmen
  - Auslagerung von Berechnungen und Transformationen
  - Umsortieren und Zusammenfassen der E/A-Anfragen
- Umorganisation der Daten
  - Z. B. explizites Speichern von Unterdatenmengen
  - Dafür aber Wissen über Datenformat notwendig
- Je näher an den Rechenknoten, desto mehr Rechenleistung
  - Andererseits müssen Daten zuerst übertragen werden

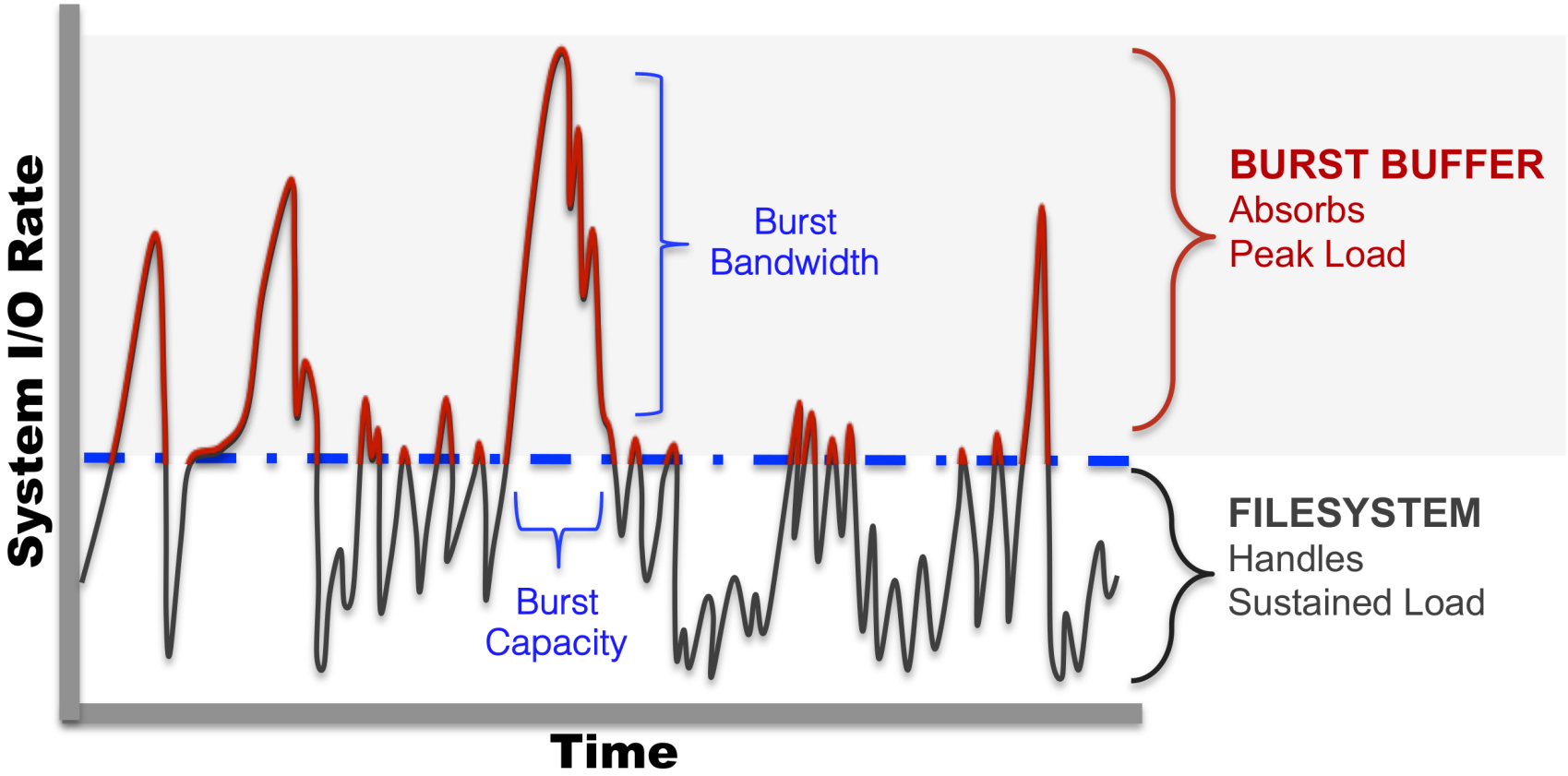
Burst Buffer

- E/A-Verhalten ist meist nicht gleichmäßig
  - Anwendungen rechnen, schreiben dann gemeinsam Checkpoint
  - Hohe E/A-Last während Checkpoint, danach deutlich weniger
- Lastspitzen bremsen Anwendungen aus
  - Mehrere Anwendungen schreiben gleichzeitig
  - Speichersysteme oft unterdimensioniert
- Koordination zwischen Anwendungen nicht gegeben
  - Könnte genutzt werden, um Last auszugleichen
  - Ist allerdings auch schwierig zu realisieren
- Hohe Kosten, um Durchsatz bereitzustellen
  - HDDs für Kapazität, SSDs für Durchsatz
  - Außerdem beschränkt durch das Netzwerk

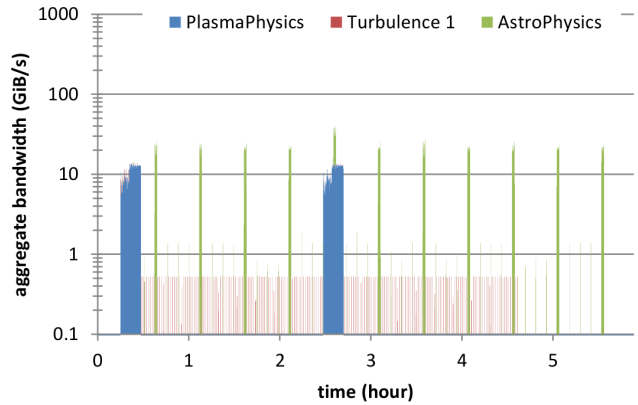
Burst Buffer... [7]

Analysis of a major HPC production storage system

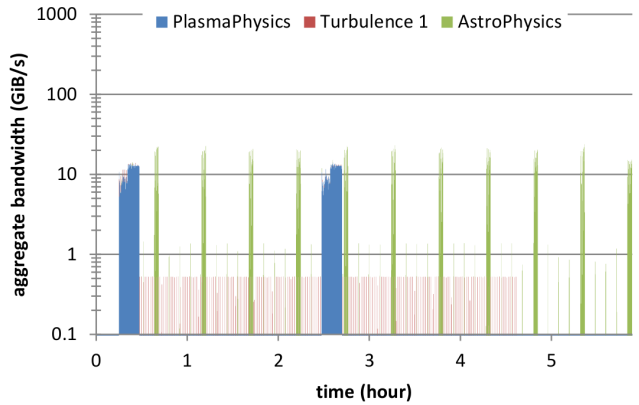
- 99% of the time, storage BW utilization < 33% of max
- 70% of the time, storage BW utilization < 5% of max



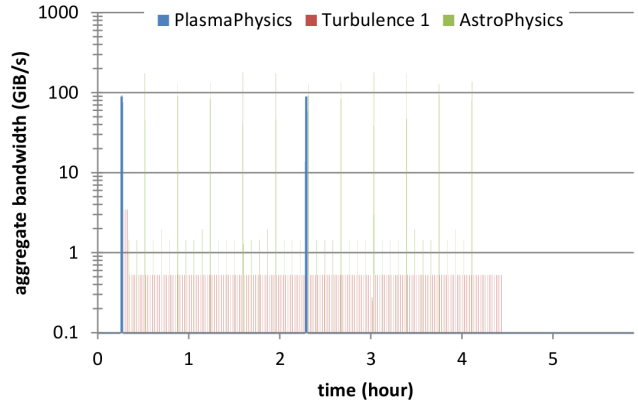
Burst Buffer... [6]



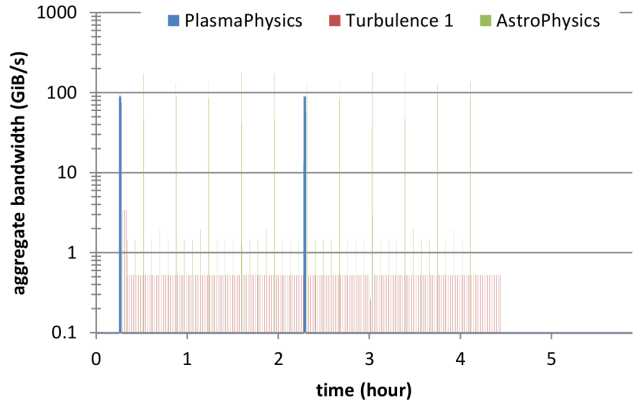
(a) burst buffer turned off, full storage system in use, 128 file servers



(b) burst buffer turned off, half storage system in use, 64 file servers



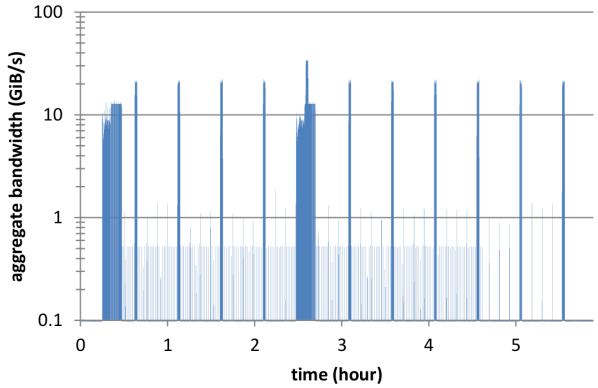
(c) burst buffer turned on, full storage system in use, 128 file servers



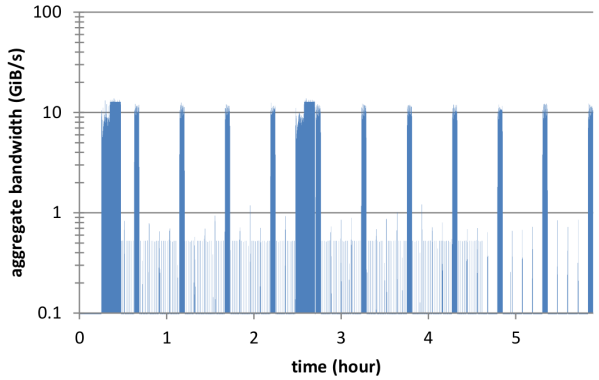
(d) burst buffer turned on, half storage system in use, 64 file servers

Fig. 5: Ten second average data transfer rate for the compute nodes observed during the multiapplication simulations.

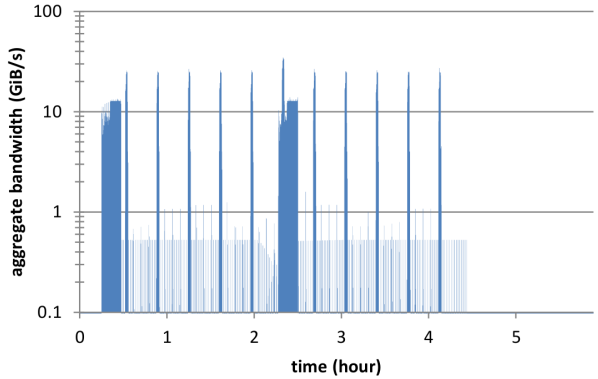
Burst Buffer... [6]



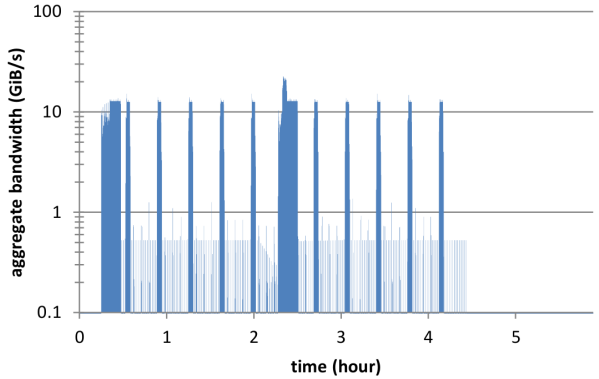
(a) burst buffer turned off, full storage system in use, 128 file servers



(b) burst buffer turned off, half storage system in use, 64 file servers



(c) burst buffer turned on, full storage system in use, 128 file servers



(d) burst buffer turned on, half storage system in use, 64 file servers

Fig. 6: Ten second average data transfer rate for the external storage system observed during the multiapplication simulations.

Burst Buffer…

- Hohe Kosteneinsparungen möglich
  - E/A-System muss nicht mehr für Lastspitzen ausgelegt werden
  - Dadurch ist u. U. ein kleineres Speichersystem ausreichend
- Eventuell können langsamere Technologien genutzt werden
  - Ethernet statt InfiniBand
  - 5.400 RPM statt 7.200 RPM
- Insgesamt erhöht sich die Ausnutzung der Hardware
  - Burst Buffer absorbieren auch ungünstige E/A problemlos
  - E/A wird „aufbereitet“ an das Speichersystem weitergegeben

Beschleuniger

- Zusätzliche Berechnungen teilweise aufwendig
  - Deduplikation, Kompression etc.
- GPUs ungeeignet, da PCIe-Bus die E/A beschränkt
  - Maximal $\approx$ 16 GB/s
- In Zukunft eventuell interessant
  - Seit Haswell QuickAssisst mit DEFLATE und LZS
  - Xeon Phi in Zukunft gesockelt, direkt am RAM angebunden

Lustre [1]

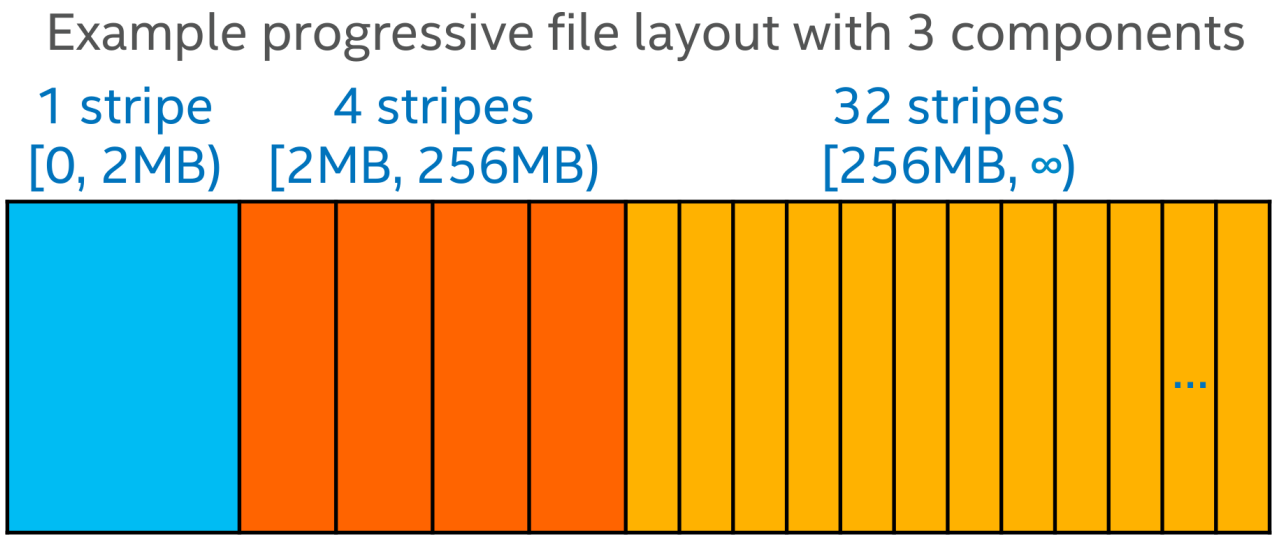
- Das klassische parallele verteilte Dateisystem gilt als überholt
  - Teile davon sollen als Basis für zukünftige Systeme dienen
  - Außerdem momentan einzige verfügbare Lösung
- Für Lustre ist eine Vielzahl neuer Funktionen geplant
  - Verschlüsselung
  - Kompression
  - Komplexe Dateilayouts
  - Weitere E/A-Optimierungen

Lustre… [1]

- Verschlüsselung zunehmend wichtig
  - Regierungen, Militär etc.
- Mehrere Zugriffsstufen
  - Frei zugänglich, vertraulich, geheim, streng geheim
  - Kein Datentransfer zwischen unterschiedlichen Stufen
- Authentifizierung und Autorisierung
  - Z. B. via Kerberos
- Verschlüsselung der Kommunikation
  - Und in Zukunft auch der persistenten Daten

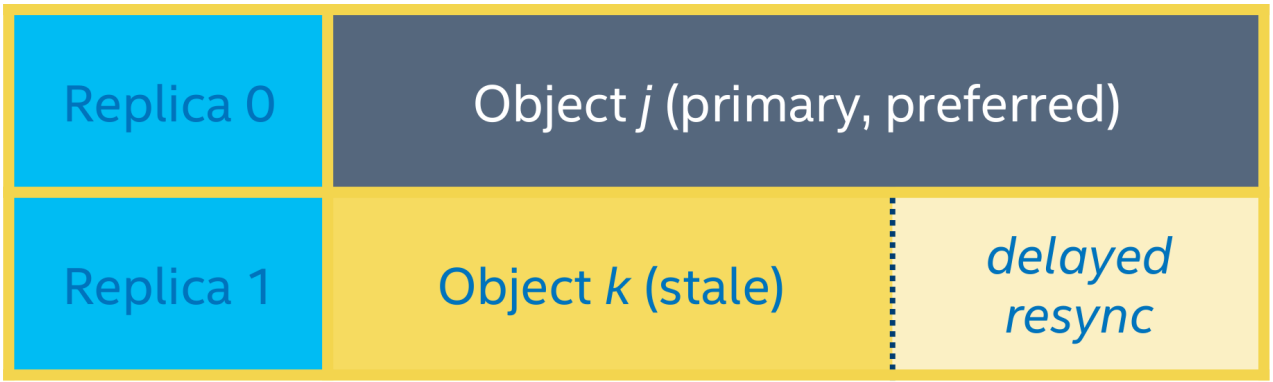
Lustre… [1]

- Unterschiedliche Anforderungen an Dateiverteilung
  - Kleine Dateien nur auf wenige OSTs
  - Große Dateien auf möglichst viele
- Automatische Anpassung der Streifenparameter



Lustre… [1]

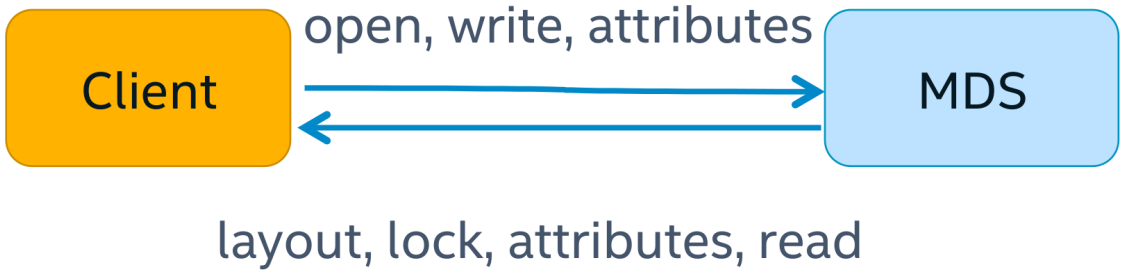
- Replikation von Dateien
  - Kann pro Datei festgelegt werden
  - High Availability, Robustheit, höhere Lesegeschwindigkeit, Migration zwischen Speicherklassen etc.



Overlapping (mirror) layout

Lustre… [1]

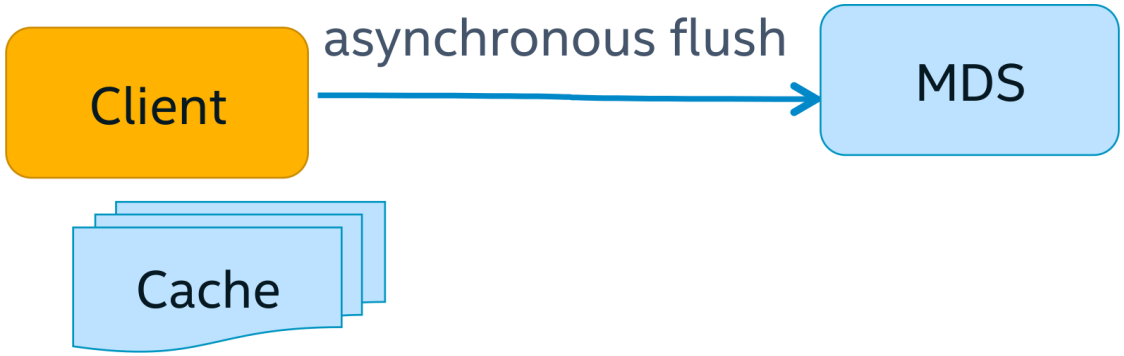
- Daten kleiner Dateien werden direkt auf dem MDT gespeichert
  - Spart Kommunikation mit OSTs
  - Erlaubt diverse Optimierungen wie z. B. Readahead
- MDTs sind üblicherweise für kleine Anfragen optimiert
  - Große Anfragen werden weiterhin über OSTs abgewickelt
- Daten müssen u. U. migriert werden, wenn Dateien wachsen



Small file IO directly to MDS

Lustre… [1]

- Metadatenoperationen sind typischerweise klein
  - Verursachen daher viel Netzwerk-Overhead
- Ein Cache erlaubt das Zusammenfassen mehrerer Operationen
  - Nur mit Sperren möglich, da sonst Konflikte auftreten können
  - Z. B. kann ein Verzeichnis gesperrt werden und darin viele Dateien angelegt werden



Client Metadata Operations Cache

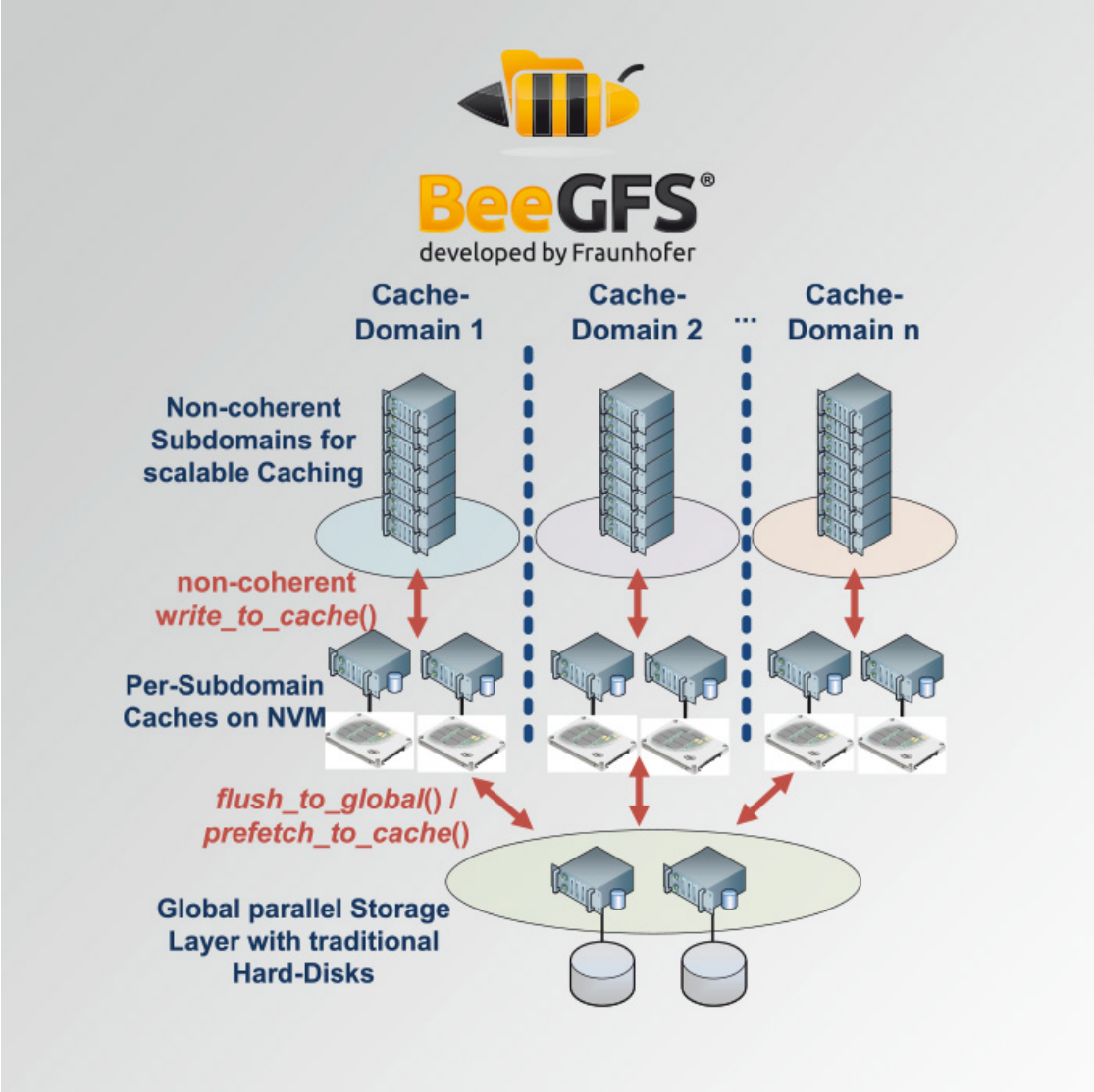
POSIX

- POSIX stellt aber immer noch eine Limitierung dar
  - POSIX ist portabel, schränkt aber Leistung ein
- Einige Alternativen
  - MPI-IO meist aber doch auf POSIX
  - Häufig POSIX-Dateisysteme auf unterster Ebene
- Abkehr von POSIX
  - Object Stores oft ausreichend
  - Häufig Einschränkungen einiger Garantien

POSIX…

- Keine globale Kohärenz mehr
  - Stattdessen nicht-kohärente Zonen
  - Zum Beispiel durch Burst Buffer oder Forwarder
- Beispiel: Cache-Domains in BeeGFS
  - Anwendungen laufen in unterschiedlichen Domänen
  - Daten werden zuerst in nicht-kohärenten Cache geschrieben
  - Caches befinden sich beispielsweise auf lokalem NVRAM
  - Daten werden aus Cache in Dateisystem migriert

POSIX... [3]


BeeGFS®
developed by Fraunhofer
Cache-Domain 1
Cache-Domain 2
...
Cache-Domain n
Non-coherent Subdomains for scalable Caching
non-coherent write_to_cache()
Per-Subdomain Caches on NVM
flush_to_global() / prefetch_to_cache()
Global parallel Storage Layer with traditional Hard-Disks

DAOS

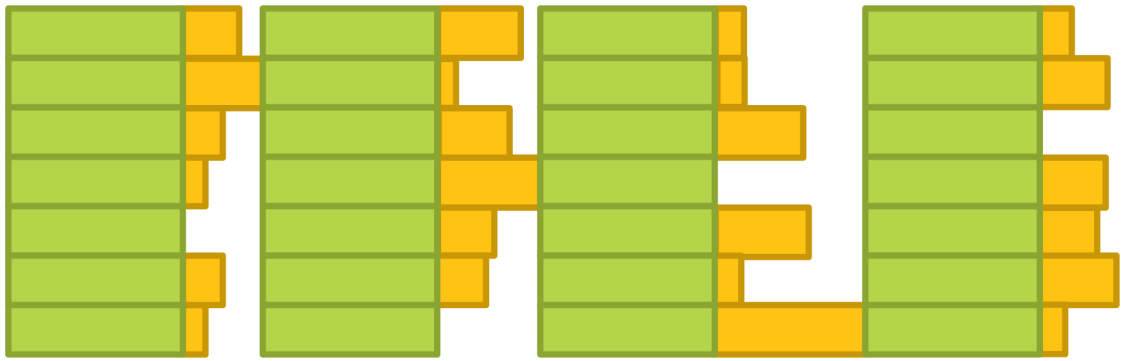
- Ganzheitlichere Ansätze in Arbeit
  - Distributed Application Object Storage (DAOS)
- DAOS unterstützt mehrere Speichermodelle
  - Container: eine Menge von Shards
  - Shard: eine Menge von Objekten
  - Objekt: Key-Value-Speicher
- Unterstützung für Versionierung
  - Operationen werden in Transaktionen durchgeführt
  - Transaktionen werden als Epochen persistiert
- Ausnutzung moderner Speichertechnologien

DAOS…

- E/A-Granularität problematisch
  - Momentan 1 MiB, bald 16 MiB
  - Dadurch mehr Konflikte und Sperren notwendig
- Bedingt durch darunterliegende Speichermedien
  - 3D XPoint bietet bytebasierten Zugriff

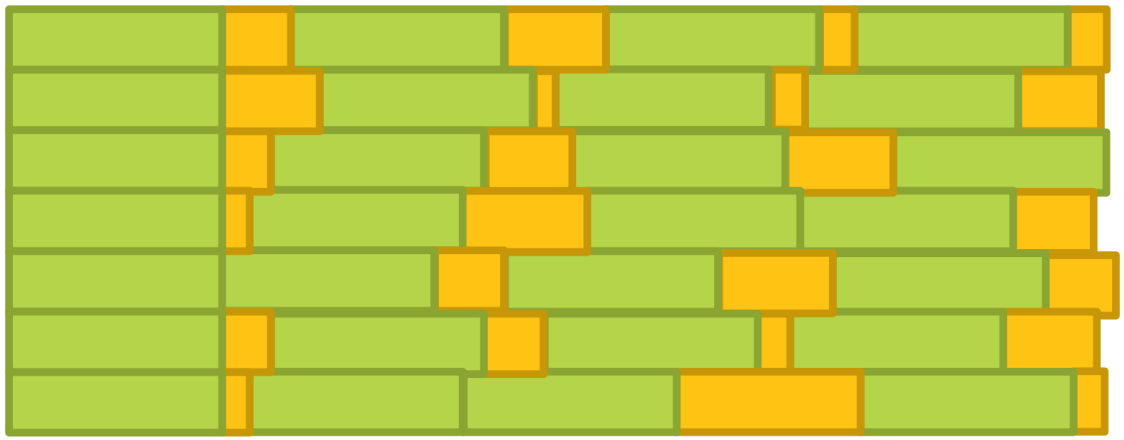
DAOS… [2]

- Üblicherweise synchrone E/A
  - Anwendungen werden durch den langsamsten ausgebremst
- Ähnliches Problem bei kollektiver Kommunikation
- E/A-Schwankungen sind der Normalfall
  - Speichersystem ist eine geteilte Ressource
  - Selten Quality of Service oder andere Garantien



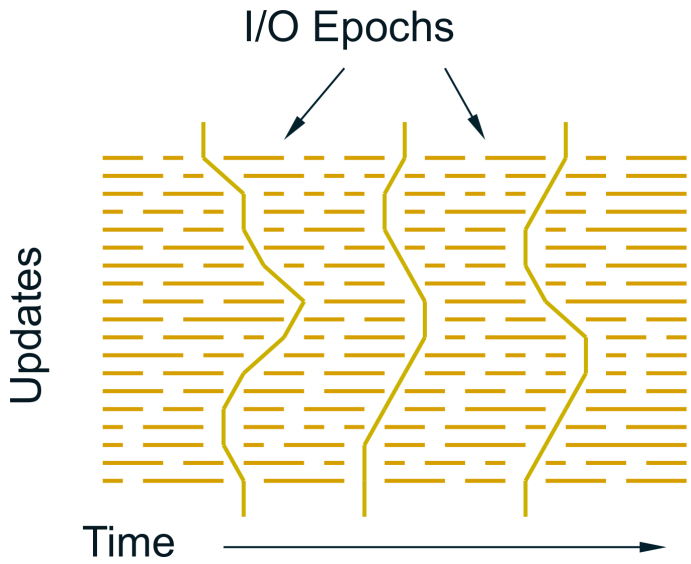
DAOS… [2]

- E/A sollte komplett asynchron stattfinden
  - Dadurch keine Wartezeiten mehr
- Frage: Wann ist die Datei konsistent?
  - Prozess könnte nächste Iteration schreiben während andere Prozesse noch mit der vorherigen beschäftigt sind



DAOS… [2]

- Lösung: Transaktionen bzw. Epochen
  - Operationen finden in Transaktionen statt
  - Mehrere Transaktionen können zu einer Epoche zusammengefasst werden
- Epochen sind global konsistente Schnappschüsse
  - Epochen werden pro Objekt vergeben
  - Dadurch keine Kohärenzprobleme beim Lesen



DAOS… [2]

- Native Unterstützung für HDF5
  - HDF-Datenstrukturen werden direkt in DAOS-Objekte überführt
  - Dadurch z. B. Umorganisation der Daten für effizienten Zugriff
- Unterstützung für unterschiedliche Anwendungen
  - POSIX für Legacy-Anwendungen
  - Big-Data-Formate für MapReduce etc.

Big Data

- Big Data wird immer wichtiger
  - Häufig leistungstechnisch suboptimal
- Hadoop nutzt normalerweise HDFS
  - Daten werden auf lokale Speichergeräte kopiert
  - Kommunikation über HTTP
- Zunehmend Unterstützung für Big-Data-Anwendungen
  - Lustre, OrangeFS etc.

Big Data…

- Andererseits interessante Ansätze aus dem Big-Data-/Cloud-Umfeld
  - Elastizität zur dynamischen Anpassung des Dateisystems
  - Zuschalten von zusätzlichen Dateisystem-Servern bei Bedarf
- Nutzung von Object Stores
  - Viele Anwendungen benötigen keine POSIX-Dateisysteme
  - MPI-IO-Funktionalität kann auf Object Stores abgebildet werden

Zusammenfassung

- Neue Hardware-Technologien werden den E/A-Stack verändern
  - Neue Hierarchieschichten durch NVRAM und 3D XPoint
  - Systeme werden dadurch komplexer und leistungsfähiger
  - Gleichzeitig sind Kosteneinsparungen möglich
  - Datentransformationen immer weiter verbreitet (Kompression)
- Die aktuelle E/A-Software wird grundlegend überarbeitet
  - High-Level-Schnittstellen für Anwendungen
  - POSIX schränkt Leistung ein und wird nicht mehr benötigt
  - Neue Ansätze aus Datenbanken und Big Data

Quellen I

[1] Brent Gorda. HPC Storage Futures – A 5-Year Outlook. http://lustre.ornl.gov/ecosystem-2016/documents/keynotes/Gorda-Intel-keynote.pdf.

[2] Brent Gorda. HPC Technologies for Big Data. http://www.hpcadvisorycouncil.com/events/2013/Switzerland-Workshop/Presentations/Day_2/3_Intel.pdf.

[3] DEEP-ER. DEEP-ER. http://www.deep-er.eu/.

[4] Jian Huang, Karsten Schwan, and Moinuddin K. Qureshi. Nvram-aware logging in transaction systems. *Proc. VLDB Endow.*, 8(4):389–400, December 2014.

Quellen II

[5] Jonas Bonér. Latency Numbers Every Programmer Should Know. https://gist.github.com/jboner/2841832.

[6] N. Liu, J. Cope, P. Carns, C. Carothers, R. Ross, G. Grider, A. Crume, and C. Maltzahn. On the role of burst buffers in leadership-class storage systems. In *012 IEEE 28th Symposium on Mass Storage Systems and Technologies (MSST)*, pages 1–11, April 2012.

[7] Mike Vildibill. Advanced IO Architectures. http://storageconference.us/2015/Presentations/Vildibill.pdf.

[8] N/A. Latency Numbers Every Programmer Should Know. http://i.imgur.com/k0t1e.png.