

Ein Konzept zum Aufbau eines föderierten, dezentralen Speichersystems im Hochschulumfeld

Konrad Meier, Dennis Wehrle
Lehrstuhl für Kommunikationssysteme
Universität Freiburg
Vorname.Nachname@rz.uni-freiburg.de

Nico Schlitter
Steinbuch Centre for Computing
Karlsruher Institut für Technologie (KIT)
nico.schlitter@kit.edu

Abstract: Traditionell werden an den Hochschulen lokale Speichersysteme betrieben. Eine kosteneffizientere Alternative stellen zentralisierte Speichersysteme dar, die allerdings über WAN-Verbindungen häufig nicht performant genutzt werden können. In dieser Arbeit werden deshalb Anforderungen für zukünftige Speichersysteme im Hochschulumfeld definiert und die Schwächen existierender Systeme aufgezeigt. Basierend auf dieser Analyse wird der Entwurf eines föderierten Speichersystems vorgestellt. Das konzipierte System vereinigt dabei die Vorzüge klassischer lokaler Datenspeichersysteme mit den Vorteilen von Cloud-basierten Ansätzen und Object-Storage-Systemen.

1 Einleitung

Traditionell betreiben die Rechenzentren der Hochschulen lokale Speichersysteme, um die von ihren Nutzern nachgefragten Dienste effizient anzubieten. Dabei führt die unmittelbare Nähe zwischen Daten speichernden und Daten verarbeitenden Systemen in Bezug auf Latenz und Datendurchsatz zu einer guten Systemperformance. Diese in der aktuellen Hochschullandschaft allgegenwärtigen Systeme sind jedoch häufig nicht kostenoptimal. Durch die Zentralisierung von Speichersystemen, die damit verbundene Beschaffung großer Stückzahlen sowie dem verringerten Personalbedarf für Management, Wartung und Betrieb kann eine deutliche Kostenreduktion erzielt werden. Dem auf diese Weise zu realisierenden finanziellen Vorteil stehen jedoch technologiebedingte Nachteile entgegen. Die skizzierten Ansätze bilden somit zwei Extreme, die ein interessantes Forschungsfeld aufspannen.

Um den implizierten Widerspruch zwischen Kosten- und Performance-Effizienz aufzulösen, werden im Rahmen des Landesprojekts bwLSDF [bwL] neue Speicherkonzepte für die Hochschulen des Landes Baden-Württemberg erarbeitet und evaluiert. Der schnelle Austausch von Daten zwischen den verschiedenen Hochschulstandorten sowie die rasche und flexible Bereitstellung projektbezogener Speichercontainer nehmen dabei einen hohen Stellenwert ein. Zukünftig könnten die vorhandenen klassischen Ansätze wie beispielsweise NFS¹ oder SMB/CIFS² von Entwicklungen abgelöst werden, welche sich die

¹Sun Microsystems Protokoll für Dateizugriffe über Netzwerk. NFS: Network File System

²Microsoft Protokoll für Dateizugriffe. SMB: Server Message Block; CIFS: Common Internet File System

neuen Cloud-Paradigmen zunutze machen, um so ein Speicher-Agglomerat zu etablieren und über die Hochschulen aufzuspannen. Somit wird der im Land vorhandene Speicher integriert, dezentral verwaltet und standortübergreifend genutzt. Die dafür notwendigen Technologien wie Cloud Data Management Interface (CDMI) [SNI12], Representational State Transfer (REST) [Fie00] und Object-Storage werden aktuell im bwLSDF-Projekt näher betrachtet und in Testumgebungen evaluiert.

Die hier vorgestellte Arbeit ist ein erster Entwurf, der die Anforderungen an ein föderiertes Speichersystem für die Hochschulen in Baden-Württemberg skizziert. Dabei erheben die Autoren bezüglich der Anforderungsanalyse keinen Anspruch auf Vollständigkeit. Die Präsentation des föderierten Speichersystems beschränkt sich auf die Darstellung einer generalisierten Systemarchitektur und abstrahiert dabei von der konkreten Ausprägung der technischen Realisierung.

Die Arbeit ist wie folgt strukturiert. In Abschnitt 2 erfolgt eine Analyse der Anforderungen an ein zukünftiges föderiertes Speichersystem für die baden-württembergische Hochschullandschaft. Anschließend werden in Abschnitt 3 die verschiedenen Ansätze und Technologien hinsichtlich der Erfüllung der Anforderungen analysiert und offene Problemstellungen aufgezeigt. In Abschnitt 4 wird ein mögliches Konzept für die zukünftige Struktur und die daraus resultierende erweiterte Funktionalität eines föderierten Speichersystems vorgestellt. Eine mögliche Umsetzung dieser Struktur wird im darauf folgenden Abschnitt 5 anhand einer prototypischen Implementierung aufgegriffen. Abschnitt 6 fasst die Arbeit zusammen und gibt einen Ausblick auf die zukünftig zu bearbeitenden Problemfelder.

2 Anforderungen an ein zukünftiges Speichersystem

Nachfolgend werden basierend auf einer Analyse von typischen Anwendungsfällen Anforderungen abgeleitet, die zukünftige Speichersysteme im Kontext der Hochschulen erfüllen sollten.

Flexibles Speichermanagement und Skalierbarkeit Rechenzentren sollen ihren ungenutzten, lokalen Speicher in ein föderiertes Speichersystem einfügen können. Der auf diese Weise integrierte Speicher kann dann - aufgeteilt in Speichercontainer - anderen Rechenzentren zur Verfügung gestellt werden, um so deren Speicherbedarf zu stillen. Mittels Metadaten soll es möglich sein den Containern bestimmte Eigenschaften zuzuordnen, die sich auf Datenschutz-, Datensicherheits- oder Performanceklassen beziehen. Das System soll dabei eine hohe Flexibilität und Skalierbarkeit hinsichtlich Containergröße, Nutzungsdauer und physikalischen Eigenschaften des Speichers ermöglichen. Die vorhandenen lokalen Speicherressourcen sollen durch Hinzufügen von Hardware zu jedem Zeitpunkt erweiterbar und somit skalierbar sein.

Robustheit Zur Erbringung der von den Rechenzentren erwarteten Dienste ist ein funktionierendes Speichersystem existenziell. Im lokalen Umfeld bieten üblicherweise Speichervirtualisierungslösungen und vorhandene Hardwareredundanzen die geforderte Aus-

fallsicherheit. Im Kontext eines föderierten Speicherkonzeptes führt dies unmittelbar zur Bedingung, dass die Grundfunktionalität des Gesamtsystems weiterhin gewährleistet sein muss, selbst wenn Teile dieses Systems ausfallen. Für den Fall, dass ein oder mehrere Rechenzentren ihren Dienst einstellen müssen - beispielsweise durch Ausfall der Energieversorgung oder des Kühlsystems - sind die dort vorgehaltenen Daten für die übrigen Rechenzentren zwar nicht mehr zugreifbar, die Verfügbarkeit der übrigen Rechenzentren sowie die Funktionalität des verbleibenden Speicherverbundes darf jedoch von diesem Ausfall nicht beeinträchtigt werden.

Mandantenfähigkeit und Accounting Nachdem ein Speichercontainer einem bestimmten Rechenzentrum zur Verfügung gestellt wurde, ist dieses völlig frei in der Nutzung der überlassenen Speicherressourcen. Das Speichersystem garantiert, dass eine Interaktion zwischen verschiedenen Speichercontainern lediglich über wohldefinierte und sichere Schnittstellen stattfinden kann. Darüber hinaus stellt das System Funktionen bereit, um eine mandantenorientierte Abrechnung der genutzten Ressourcen zu ermöglichen.

Zugriffsmanagement und Datenschutz Die Speicherung von Daten innerhalb der Speicherföderation muss den geltenden Datenschutzerfordernungen genügen. Folglich soll das Speichersystem die dafür notwendigen Funktionalitäten bieten. Der Zugriff auf Speichercontainer muss durch die Bereitstellung geeigneter Authentifizierungs- und Autorisierungsmechanismen beschränkbar sein. Darüber hinaus besteht bei der Ablage personenbezogener Daten die Notwendigkeit geeigneter Verschlüsselungsverfahren und eine sichere Schlüsselverwaltung zu realisieren, sodass eine kryptographische Sicherung der gespeicherten Daten möglich wird.

Flexibles Datenmanagement Um einen hohen Flexibilisierungsgrad im Datenmanagement zu ermöglichen, unterstützt das System nutzerdefinierte Policies. Diese ermöglichen unter Berücksichtigung von Zugriffsrechten die Durchführung automatisierter Datenmanipulationen und containerübergreifende Datentransfers.

Datensicherheit Die Sicherheit der abgelegten Daten wird zunächst durch die Charakteristika der lokalen Speicherhardware und der darauf aufbauenden Speichervirtualisierung bestimmt. Unabhängig von diesen lokalen Faktoren soll das System Funktionalitäten bereitstellen, die auf föderierter Ebene eine redundante Datenhaltung an mehreren Standorten ermöglichen.

Performance Der Zugriff auf die im Speichersystem abgelegten Daten soll effizient sein. Der zu erreichende Datendurchsatz ist dabei prinzipiell durch die zugrundeliegende Netzwerkinfrastruktur beschränkt. Die innerhalb des Speichersystems verwendeten Übertragungsprotokolle sollten jedoch so skalieren, dass es prinzipiell möglich ist, die zur Verfügung stehende Bandbreite maximal zu nutzen.

Kosteneffizienz Durch ein flexibles Speicher- und Datenmanagement sowie durch Skalierbarkeit des Systems sollen die Kosten für Management, Wartung und Betrieb minimiert werden.

3 Stand der Technik und resultierende Probleme

Nachfolgend werden aktuelle Ansätze zur Speicherverwaltung anhand der in Abschnitt 2 definierten Anforderungen an ein zukünftiges Speichersystem analysiert. Dabei wird grundsätzlich zwischen lokalen, zentralisierten und verschiedenen Ausprägungen dezentraler Lösungen unterschieden.

Lokaler Ansatz Aus Sicht der Performance (Datendurchsatz und Latenz) ist es optimal, Daten an dem Standort abzuspeichern, an dem sie auch verarbeitet werden. Diese Performance ist jedoch für Anwendung wie Archivierung und Backup nicht zwingend notwendig. Eine Zentralisierung der Speicherbedarfe für diese Anwendungsfälle würde in Bezug auf die anfallenden Management-, Wartungs- und Betriebskosten zu einer effizienteren Lösung führen. Darüber hinaus können zusätzliche Bedarfe für die temporäre Speicherung von Daten nur durch Beschaffung neuer Speicherhardware befriedigt werden. Folglich fehlt dem lokalen Ansatz die geforderte Kosteneffizienz und die Flexibilität bezüglich der Nutzungsdauer von Speicher.

Zentralisierter Ansatz Das Zentralisieren und Konsolidieren von Speichersystemen an einem Standort scheint zunächst Kosteneffizient. Es entstehen jedoch unweigerlich Performanceprobleme, die der vorhandenen Netzwerkinfrastruktur geschuldet sind. In Baden-Württemberg sind die Hochschulen über das gemeinsam genutzte Netzwerk BelWü [bela] miteinander verbunden. Folglich müssen Daten zwischen einem zentralen Speicher und den Daten verarbeitenden Rechenzentren über dieses Netzwerk übertragen werden. Obwohl die aktuelle Netztopologie zum großen Teil aus 10 GBit/s Verbindungen besteht, [belb] kann es hierbei zu Engpässen kommen, die sich in hohen Latenzen und geringem Datendurchsatz widerspiegeln:

- **Latenz** Bei Signalübertragungen zwischen geografisch verteilten Rechenzentren treten Latenzen auf. Ein Teil dieser Latenzen ist aufgrund der Limitierung auf Lichtgeschwindigkeit unabänderlich. Die restliche Signalverzögerung kann jedoch durch aktive Komponenten (z. B. Firewalls, Switches oder Router) der Netzwerkinfrastruktur begründet werden.
- **Datendurchsatz** Der Datendurchsatz ist durch die gemeinsam genutzte Netzwerkinfrastruktur und die verwendeten Übertragungsprotokolle beschränkt. Firewalls und Switches können - abhängig von deren Leistungsfähigkeit - einen erheblichen Einfluss auf den zu erzielenden Datendurchsatz aufweisen. Darüber hinaus limitiert die verwendete TCP-Fenstergröße und die auftretende Latenz zwischen zwei Standorten die Übertragungsrate.

Messungen anhand des iSCSI Protokolls zeigen im lokalen Kontext einen Durchsatz von 220 MB/s. Im zentralisierten Kontext fällt die Durchsatzrate bei einer Datenübertragung zwischen Karlsruhe und Freiburg auf 18 MB/s. Diese vorläufigen Messungen deuten an, dass ein zentraler Speicher für viele Anwendungsfälle ungeeignet ist. Aus diesem Grund wird ein alternatives Konzept notwendig.

Dezentrale Ansätze Bei dezentralen Ansätzen wird das Speichernetz über mehrere Standorte hinweg aufgespannt. Dabei kommen sowohl (verteilte) Dateisysteme, als auch neuartige Object-Storage-Systeme in Betracht. Dabei werden bei einem Object-Storage-System die Daten als Objekte in Datencontainern abgelegt und mit Metadaten angereichert.

Dezentrale Systeme lassen sich sowohl durch Dateisysteme als auch durch Object-Storage-Systeme realisieren. Beispiele für dezentral verteilte Dateisysteme sind GFS [GGL03], GlusterFS [Glu], MooseFS [Moo], Lustre [Sch03], Ceph [WBM⁺06] und GPFS [SH02]. Bei einem über mehrere Standorte aufgespannten Speichernetz sollte jedoch kein einzelnes großes Dateisystem verwendet werden. Eine mögliche Inkonsistenz des Dateisystems würde eine zeitaufwendige Reparatur nach sich ziehen, während dieser die gesamten Daten nicht zugänglich sind.

Dieses Problem lässt sich durch geeignete Object-Storage-Systeme lösen, die nicht auf großen Dateisystemen aufbauen. Aktuelle Implementierungen wie GlusterFS [Glu], Open-Stack Storage [Ope] und Scality's RING Organic Storage [Sca] bieten diese Funktionalität - jedoch lässt sich der Speicherort nicht frei bestimmen. Folglich besteht analog zum zentralisierten Ansatz ein Performanceproblem.

Aus diesem Grund ist die Bestimmbarkeit des Speicherortes für die Erfüllung der oben genannten Anforderungen von zentraler Bedeutung. Daher wird ein Object-Storage-System favorisiert, bei dem der Speicherort durch den Nutzer frei wählbar ist. Exemplarisch sei hier das Speichersystem iRODS [MR10] genannt, welches durch die integrierte Rule-Engine und vom Nutzer definierten Policies die Möglichkeit bietet den Speicherort von Daten festzulegen.

4 Föderiertes Speichersystem

Das im Folgenden vorgestellte, föderierte Speichersystem verbindet unterschiedliche Standorte und deren jeweilige Systeme zu einem Speicherverbund. Das System basiert auf dem Ansatz eines Object-Storage-Systems mit frei wählbarem Datenspeicherort und wird durch eine Abstraktions- und Verwaltungsschicht über der lokalen Speicherverwaltung ermöglicht. Die Abstraktionsschichten und die Kommunikation zwischen ihnen ist in Abbildung 1 dargestellt. Die Abbildung zeigt für die Rechenzentren A und B das vorgeschlagene Schichtenmodell. Die Kommunikation zwischen den Rechenzentren erfolgt über die Schicht der föderierten Speicherverwaltung. Die Rechenzentren C und D sind äquivalent aufgebaut.

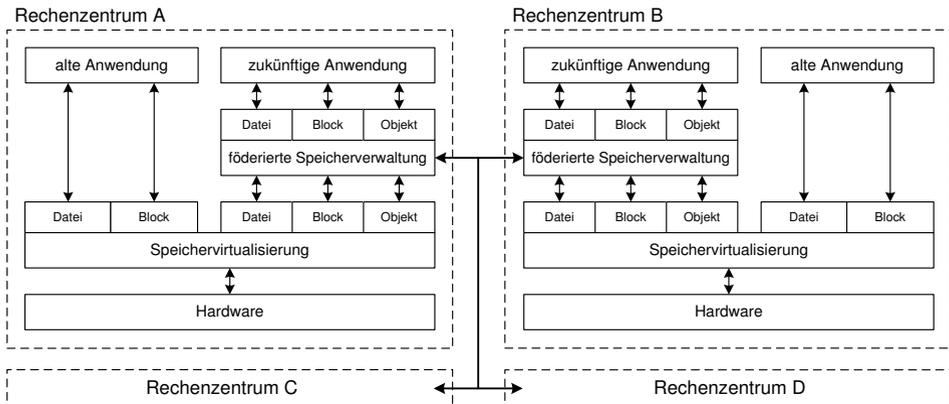


Abbildung 1: Schichtenmodell des föderierten Speichersystems mit Kommunikationsbeziehungen

Hardware Auf der untersten Schicht befindet sich die lokale Speicher-Hardware. Die Aufgabe dieser Schicht besteht darin, Speicher für die darüberliegende Schicht bereitzustellen. Hierbei ist es unerheblich, aus welchen Systemen der Speicher besteht. Es können klassische Festplatten (HDD) oder Solid-State-Drives (SSD), aber auch Bandlaufwerke eingesetzt werden. Die Hardwareausstattung kann somit sowohl innerhalb eines Standortes, also auch zwischen den Standorten unterschiedlich sein.

Speichervirtualisierung Die Speichervirtualisierung verwendet den von der Hardware-Schicht bereitgestellten Speicher und stellt ihn in Speicherpools über Datei- oder Blockzugriff bereit. Wird als Speichervirtualisierung ein Object-Storage wie beispielsweise Open-Stack Storage oder Scality's RING Organic Storage verwendet, stellt diese Schicht einen Objektzugriff bereit. Darüber hinaus kann die Speichervirtualisierungsschicht für eine lokale Redundanz der Daten verantwortlich sein.

Föderierte Speicherverwaltung Die von der lokalen Speichervirtualisierung bereitgestellten Speicherpools werden in der föderierten Speicherverwaltung als Object-Storage angesprochen. Falls bei der lokalen Speichervirtualisierung kein Object-Storage, sondern ein Datei- oder Blockspeicherpool verwendet wird, übernimmt die föderierte Speicherverwaltung die Übersetzung in einen Object-Storage. Über die föderierte Speicherverwaltung sind die Standorte untereinander verbunden. Diese Schicht ermöglicht es, Objekte zwischen den Standorten zu verschieben und zu kopieren. Diese Funktionen können durch den Nutzer oder mittels Policies gesteuert werden. Dabei sind Policies Regeln, die den Speicherort von Objekten beeinflussen. Sie ermöglichen es unter anderem die folgenden Funktionen zu realisieren:

- **Auslagern von Daten** Der Speicherort von Daten kann innerhalb der Föderation verändert werden. So wird es möglich, Daten, die längere Zeit nicht verwendet wurden, automatisch an einen anderen Standort auszulagern. Dies kann im Bedarfsfall

kosteneffizienter sein als das Bereitstellen lokaler Ressourcen.

- **Überlauffunktion** Die Überlauffunktion verlagert Daten an andere Standorte, um lokale Kapazitätsengpässe zu kompensieren. Abbildung 2(a) zeigt einen Anwendungsfall, bei dem Daten des Rechenzentrums C anhand einer Policy automatisch an einen geeigneten Standort (Rechenzentrum A) ausgelagert werden, sobald die verfügbare Speicherkapazität unter 20% fällt.
- **Redundanz** Um die Datensicherheit zu erhöhen, können Daten automatisch repliziert und an einen anderen Standort ausgelagert werden. Abbildung 2(b) illustriert ein Szenario, in dem jeder Standort anhand von Policies eigenständig bestimmt, welche Daten wie häufig repliziert werden.

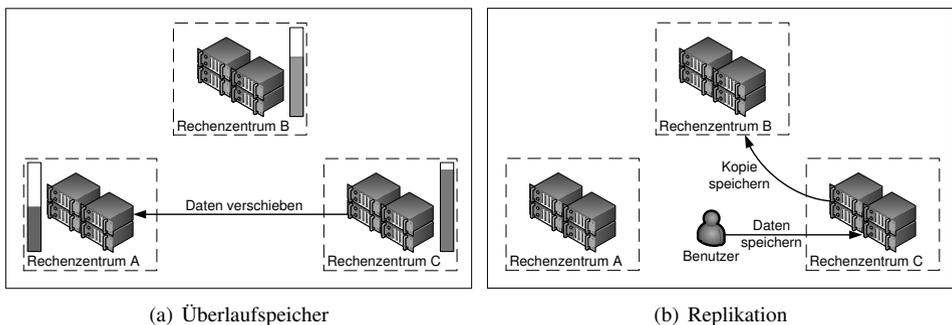


Abbildung 2: Anwendungsfälle für policy-basiertes Datenmanagement

Die föderierte Speicherverwaltung präsentiert den Anwendungen die Speichercontainer mittels Konnektoren. Für den Datei- und Blockzugriff muss der entsprechende Konnektor den internen Object-Storage als Dateisystem oder als Blockdevice darstellen. Eine Anwendung kann somit über Block- oder Dateisystembefehle auf den Speicher zugreifen.

Anwendung Die vorgestellte Struktur ermöglicht es existierende Anwendungen weiterhin über die vorhandene Speichervirtualisierung zu betreiben. Bei diesen Systemen erfolgt der Zugriff auf den Speicher üblicherweise über Block- oder Dateiebene. So kann zum Beispiel ein Fileserver angebunden werden, der Benutzerverzeichnisse über NFS oder SMB/CIFS bereitstellt.

Zukünftige Anwendungen können auf das föderierte Speichersystem über Datei-, Block- und Objekt-Konnektoren zugreifen. Im Idealfall verwenden die Anwendungen dabei direkt den Konnektor für Objekte, um Daten zusammen mit Metadaten im System zu speichern. Ist dies nicht der Fall und die Anwendung benötigt Zugriff auf ein Datei- oder Block-System, werden diese über entsprechende Konnektoren bereitgestellt. Ein Zugriff von alten Anwendungen über das föderierte Speichersystem erfordert eine Migration der zugehörigen Daten in das föderierte System.

5 Prototyp

Für das vorgestellte System wird aktuell ein Prototyp auf Basis von OpenStack Storage (Swift) entwickelt, um eine Evaluierung des Systems zu ermöglichen und dessen Performance zu beurteilen. Die Entwicklung beschränkt sich dabei aktuell auf die Funktionen zum flexiblen Datenmanagement sowie der Datensicherheit.

Der in Abbildung 3 gezeigte Prototyp simuliert drei Rechenzentren mit drei Servern pro Standort. Diese drei Server bilden somit die lokale Speichervirtualisierung (vergleiche Abbildung 1) und realisieren einen Object-Store. Zwei der Server werden zur Speicherung der Objekte verwendet (Swift-Storage). Der dritte Server ist ein Proxy (Swift-Proxy) und ermöglicht den Zugriff auf den Object-Store. Um Datenredundanz für den Fall eines

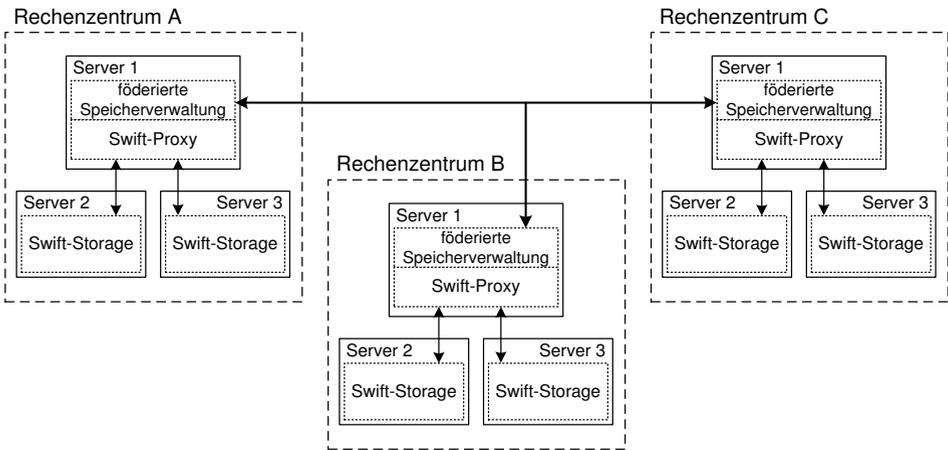


Abbildung 3: Aufbau des Prototyps

Ausfalls zu schaffen, werden Objekte in Swift auf mehrere Server repliziert. Die Verteilung der Daten auf die unterschiedlichen Server lässt sich bei Swift nicht beeinflussen, was ein rechenzentrumsübergreifendes Swift-System verhindert. Aus diesem Grund wird der Swift-Proxy um die Software der föderierten Speicherverwaltung erweitert. Diese Schicht ermöglicht die Kommunikation zwischen den Rechenzentren. Der Zugriff auf das OpenStack Storage System erfolgt dabei transparent über den Proxy der föderierten Speicherverwaltung. Die aktuelle Implementierung ermöglicht es, manuell Objekte zwischen den Standorten zu verschieben (Auslagern von Daten) sowie frei wählbare Standorte für die Replizierung von Objekten (Redundanz) zu verwenden. Dabei ist es möglich, die Daten mit einem Standort-Schlüssel zu verschlüsseln, wenn diese den ursprünglichen Standort verlassen.

6 Fazit und Ausblick

Die Anforderungen an Speichersysteme im Hochschul Umfeld können durch lokale Speichersysteme nicht erfüllt werden. Zentrale Speichersysteme weisen vor allem hinsichtlich ihrer Performance Schwächen auf. Das vorgestellte föderierte Speichersystem kombiniert die Vorzüge beider Systeme und berücksichtigt dabei die vorhandene Infrastruktur an den Hochschulen. Es wird somit ermöglicht, die Speicherkapazitäten an unterschiedlichen Standorten flexibel zu nutzen.

Der entwickelte Prototyp enthält erste Möglichkeiten zum flexiblen Datenmanagement, indem der Standort von Daten beeinflusst werden kann. Zukünftig muss insbesondere die Definition von Policies und deren Umsetzung betrachtet werden, um beispielsweise eine automatisierte Überlauffunktionalität zu ermöglichen. Aktuell implementiert die föderierte Speicherverwaltung die REST-Schnittstelle von OpenStack Storage sowohl für die Kommunikation zur Speichervirtualisierung als auch für die Kommunikation zwischen den Standorten. Für die Kommunikation zwischen den Standorten ist zukünftig eine standardisierte Kommunikation über CDMI wünschenswert um eine herstellerunabhängige Schnittstelle zu schaffen. Anhand des Prototyps soll auch ein Datenschutzkonzept entwickelt werden, das Daten vor unberechtigtem Zugriff durch Dritte schützt. Ein erster Ansatz findet sich in der Arbeit von Bessani et al. [BCQ⁺11]. Zur Gewährleistung der Datensicherheit sind jedoch unterschiedliche Ansätze und somit auch unterschiedliche Verantwortlichkeiten denkbar. Diese Überlegungen sind Teil weiterer Arbeiten. Darüber hinaus wird ein Konzept zur mandantenorientierten Abrechnung der genutzten Ressourcen und ein entsprechendes Accounting benötigt.

Literatur

- [BCQ⁺11] Alysson Bessani, Miguel Correia, Bruno Quaresma, Fernando André und Paulo Sousa. DepSky: dependable and secure storage in a cloud-of-clouds. In *Proceedings of the sixth conference on Computer systems*, EuroSys '11, Seiten 31–46, New York, NY, USA, 2011. ACM.
- [bela] BelWü - das Landeshochschulnetz, <http://www.belwue.de> [letzter Zugriff: 14.01.2013].
- [belb] BelWü Netz-Topologie [Stand 01.2013], <http://www.belwue.de/topology> [letzter Zugriff: 14.01.2013].
- [bwL] bwLSDF, <http://bwlsdf.scc.kit.edu> [letzter Zugriff: 15.01.2013].
- [Fie00] Roy Thomas Fielding. *Architectural styles and the design of network-based software architectures*. Dissertation, 2000. AAI9980887.
- [GGL03] Sanjay Ghemawat, Howard Gobioff und Shun-Tak Leung. The Google file system. In *Proceedings of the nineteenth ACM symposium on Operating systems principles*, SOSP '03, Seiten 29–43, New York, NY, USA, 2003. ACM.
- [Glu] GlusterFS, <http://www.gluster.org> [letzter Zugriff: 13.01.2013].
- [Moo] MooseFS, <http://www.moosefs.org> [letzter Zugriff: 13.01.2013].
- [MR10] Reagan W. Moore und Arcot Rajasekar. Irods: Data sharing technology integrating communities of practice. In *IGARSS*, Seiten 1984–1987. IEEE, 2010.
- [Ope] OpenStack Storage, <http://www.openstack.org/software/openstack-storage> [letzter Zugriff: 18.01.2013].
- [Sca] Scality RING Organic Storage, <http://www.scality.com/ring-organic-storage> [letzter Zugriff: 18.01.2013].
- [Sch03] Philip Schwan. Lustre: Building a File System for 1,000-node Clusters. In *Proceedings of the Linux Symposium*, Seite 9, 2003.
- [SH02] Frank Schmuck und Roger Haskin. GPFS: A Shared-Disk File System for Large Computing Clusters. In *Proceedings of the 1st USENIX Conference on File and Storage Technologies*, FAST '02, Berkeley, CA, USA, 2002. USENIX Association.
- [SNI12] SNIA. Information Technology - Cloud Data Management Interface (CDMI), Version 1.0.22, <http://snia.org/sites/default/files/CDMIv1.0.2.pdf>. Juni 2012.
- [WBM⁺06] Sage A. Weil, Scott A. Brandt, Ethan L. Miller, Darrell D. E. Long und Carlos Maltzahn. Ceph: a scalable, high-performance distributed file system. In *Proceedings of the 7th symposium on Operating systems design and implementation*, OSDI '06, Seiten 307–320, Berkeley, CA, USA, 2006. USENIX Association.