Technische Hochschule
Brandenburg
University of
Applied Sciences

Anwendungsmöglichkeiten von Speichervirtualisierung für kleinere Unternehmen am Beispiel von OpenZFS und Microsoft Storage Spaces

Bachelorarbeit

zur Erlangung des Grades Bachelor of Science
des Fachbereichs Informatik und Medien der
Technischen Hochschule Brandenburg

vorgelegt von:
Otto-Leander Werse (Matrikelnummer: 20206046)

Betreuer: Prof. Dr.-Ing. Thomas Preuß
Zweitgutachter: Dipl.-Ing. Holger Matho (AGILIScom AG)

Brandenburg an der Havel, 11. Oktober 2021

Kurzfassung

Mit zunehmender Digitalisierung müssen immer mehr Daten gelagert werden, insbesondere im kommerziellen Bereich. Auch kleinere Unternehmen, welche nicht im IT-Bereich tätig sind, sind davon betroffen. Klassische, professionelle Speicherlösungen sind oftmals aufwändig zu administrieren und zu inflexibel für solche Unternehmen.

Microsoft Storage Spaces und OpenZFS sind zwei RAID-ähnliche Systeme zur Virtualisierung von Speicherplatz. Die Systeme entspringen sehr unterschiedlichen Philosophien. Storage Spaces wird sowohl für den Konsumentenmarkt als auch den professionellen Bereich vermarktet. Und von Microsoft als anwenderfreundliche Alternative zu RAID angesehen. Es handelt sich um ein proprietäres System, welches großflächig auf die Features des unterliegenden Betriebssystems (Windows oder Windows Server) zurückgreift. Die Funktionalität ist hierbei auf Desktop Betriebssystemen gegenüber Server Betriebssystemen stark eingeschränkt.

OpenZFS dagegen stammt ursprünglich aus der Datacenter-Anwendung von Sun Microsystems und später der Oracle Corporation. Es wird mittlerweile als Free Open Source Software durch das OpenZFS Project vertrieben und findet viele Anhänger im SOHO-Bereich. Es ist für eine Vielzahl von Betriebssystemen verfügbar, wird allerdings primär unter Linux entwickelt.

Beide Systeme bieten eine Vielzahl an zusätzlichen Features an, welche traditionelle RAID-Systeme lediglich über zusätzliche Software bieten können. OpenZFS bietet hierbei, aufgrund seines Aufbaus, tiefere Integration.

Initiale Kostenfaktoren sind im Falle von Storages Spaces deutlich höher, da für die Nutzung aller Features eine aktuelle Windows Server Lizenz benötigt wird. OpenZFS ist selbst kostenfrei, und bietet auch Support für eine Vielzahl an kostenfreien Betriebssystemen an.

Schlüsselwörter

Microsoft Storage Spaces – OpenZFS – RAID – Speichervirtualisierung

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

Die Suche nach immer performanteren, effizienteren und zuverlässigeren Methoden der elektronischen Datenhaltung ist so alt, wie der Computer selbst. Bis in die 1970er Jahre lag der Fokus der Forschung vorwiegend auf der Entwicklung neuer physischer Medien, mit immer höherer Zuverlässigkeit, schnelleren Transfergeschwindigkeiten und höheren Speicherkapazitäten.

In den Jahren zwischen 1970 und 1990 rückte nun ein neues Forschungsgebiet im Bereich der Datenhaltung in den Vordergrund, die Speichervirtualisierung. Man spricht hierbei vom Zusammenschluss mehrerer physischer Speichermedien, wie z.B. Festplatten, zu einem einzelnen virtuellen Speichermedium. (DataCore Software, kein Datum)

Das wahrscheinlich bekannteste System zur Virtualisierung von Speicher ist das erstmals im Jahre 1988 benannte „RAID“-System. Dieses wurde von David Patterson, Garth A. Gibson und Randy Katz an der University of California, Berkeley entwickelt und in der Publikation „A Case for Redundant Arrays of Inexpensive Disks (RAID)“ veröffentlicht. Bereits damals war der Kerngedanke der Forschung, die Möglichkeit der Kosteneinsparung durch das Verwenden von gewöhnlicher Hardware aus dem Konsumentenmarkt, anstelle von spezialisierter Hardware für professionelle Anwender. (Patterson, Gibson, & Katz, 1988) Die Bedeutung des Akronyms „RAID“ wurde von der Industrie im Laufe der Jahre zu „Redundant Arrays of Independent Disks“ geändert. (RAID, kein Datum) Dennoch lebt der ursprüngliche Ansatz des RAID-Systems auch heute weiter.

In einer zunehmend digitalisierten Welt besteht großer Bedarf nach Speicherplatz. Schließlich generiert im Jahre 2020 im Durchschnitt jeder Mensch rund 1,7 Megabytes an Daten pro Sekunde. (Bulao, 2021) Ganz zu schweigen von den Datenmengen, welche im gewerblichen Bereich entstehen. So entstanden aus der ursprünglich hardwarebasierten RAID-Technologie seitdem eine Vielzahl Software-basierter Systeme. Neben einer Vielzahl an Software-Implementierungen traditioneller RAID-technologie (auch Software-RAID genannt), entstanden auch einige Systeme, welche sich die Vorteile einer rein in Software implementierten Lösung zu Nutze machen.

Diese Systeme benötigen meist nur ein sehr kleines initiales Investment, da anstatt spezialisierten Hardware-Steuereinheiten lediglich eine Software-Lizenz erworben werden muss, welche in einigen Fällen sogar kostenfrei ausfallen kann. (RAID Incorporated, 2020) Auch die Flexibilität solcher Systeme ist oftmals deutlich größer. Dies kommt daher, dass die Funktionalität des Systems nicht durch spezialisierte Schaltkreise (ASIC) oder Code, welcher an deren Aufbau angelehnt ist, geboten wird. Somit ist diese theoretisch vollständig anpassbar. Durch den oftmals flexibleren Aufbau und die niedrigen initialen Kosten sind solche Systeme besonders interessant für kleinere Unternehmen und Privatanwender. So kann ein Speichersystem aufgebaut werden, welches problemlos mit den Ansprüchen eines wachsenden Unternehmens skaliert werden kann und das mit möglichst kleinem Aufwand. (Foskett, 2017)

1.2 Aufgabenstellung

Diese Arbeit wird sich mit zwei Software-basierten RAID-ähnlichen Lösungen auseinandersetzen, welche jeweils einer sehr unterschiedlichen Philosophie entsprangen. Einerseits wird sich mit dem proprietären „Storage Spaces“-Ökosystem von Microsoft auseinandergesetzt. Andererseits soll sich mit „OpenZFS“, einem Open-Source-Projekt des OpenZFS Project, befasst werden. (OpenZFS Project, History, 2014)

Hierbei soll auf mögliche Anwendungsmöglichkeiten eingegangen werden, welche sich durch die Skalierbarkeit und die Kostenfaktoren dieser Systeme anbieten. Hierzu wird primär auf technische Analyse gesetzt, welche je nach Bedarf durch praktische Tests visuell unterstützt werden soll. So soll eine Argumentation entstehen, welche auf den Anwendungsbereich kleinerer Unternehmen eingeht. Diese soll möglichst unabhängig von deren Tätigkeitsbereich sein. Es soll also nicht spezifisch auf Unternehmen aus dem IT-Bereich eingegangen werden.

Ziel soll es ebenfalls nicht sein, einen Vergleich zu bestehenden Cloud-basierten Lösungen, wie Amazon AWS oder Microsoft Azure, zu ziehen. Es soll sich ausschließlich mit On-Permisse-Lösungen befasst werden.

1.3 Gliederung

Zunächst sollen die beiden Systeme und deren Features, Aufbau, sowie die Bedienung durch den Nutzer vorgestellt und erläutert werden. Hierbei wird zu Beginn auf die Alleinstellungsmerkmale gegenüber traditionellen Dateisystemen, sowie herkömmlichen RAID-Implementierungen in Form von Features eingegangen. Anschließend wird der interne Aufbau des Systems so weit erklärt, wie es für den Umfang dieser Arbeit anschaulich und notwendig ist. Abschließend wird kurz angeschnitten, wie der Administrator bzw. die Endnutzer mit dem System interagieren können.

Nachdem die Charakteristiken der zwei Systeme beschrieben wurden, werden die Anwendungsmöglichkeiten der Systeme für kleinere Unternehmen anhand einiger sinnvoller Beispiele evaluiert.

Abschließend wird innerhalb des Kontextes von Kleinst- bzw. Kleinunternehmens verglichen, wie groß der Aufwand bzw. der Nutzen der beiden Systeme innerhalb eines solchen Unternehmens ausfallen würde. Hierbei soll insbesondere auf initiale Kostenfaktoren, wie Lizenzkosten, sowie langzeitliche Kostenfaktoren, wie der Wartungsaufwand im Sinne der benötigten Zeit und Qualifikation, eingegangen werden.

2 OpenZFS

2.1 Entwicklung

Ursprünglich wurde das „ZFS“-System unter dem Namen „Zettabyte File System“ von Sun Microsystems als Teil des Betriebssystems „Solaris“ entwickelt. Die Arbeit an ZFS begann bereits im Jahre 2001, es wurde aber erstmals 2004 veröffentlicht. Das ursprünglich proprietäre Closed-Source Solaris wurde 2005 erstmals als „OpenSolaris“ unter Open-Source-Lizenzen veröffentlicht. Die darin enthaltene Implementation von ZFS wurde um 2006 auf eine Vielzahl von Betriebssystemen portiert. Im Jahre 2013 wurde die Organisation „OpenZFS Project“ gegründet, welche seitdem die Code Base von OpenZFS verwaltet. (OpenZFS Project, History, 2014)

Analog zur Open-Source-Entwicklung hat auch die Oracle Corporation, welche Sun Microsystems im Jahr 2009 aufgekauft hat (Oracle Corporation, Oracle Buys Sun, 2009), weiter an ihrer Version von ZFS gearbeitet. Diese Version ist, gemessen am eigentlichen Quellcode, mittlerweile sehr unterschiedlich zu OpenZFS. Allerdings wird selbst in der offiziellen Dokumentation des OpenZFS Project vielmals auf die Dokumentation von Oracle ZFS verwiesen, da die fundamentale Funktionalität größtenteils dennoch sehr ähnlich ist. (OpenZFS Project, System Administration, 2016) Daher wird diese Dokumentation in den folgenden Abschnitten des Öfteren als Quelle mitverwendet. Es sei allerdings vermerkt, dass nicht alle Inhalte dieser Quellen, insbesondere die darin enthaltenen Kommandos, auch zwangsläufig auf OpenZFS zutreffen.

2.2 Features

2.2.1 Transparente Komprimierung

OpenZFS unterstützt eine Vielzahl an Komprimierungsalgorithmen (Salter, ZFS 101—Understanding ZFS storage and performance, 2020), welche je nach Host-Betriebssystem und Version unterschiedlich ausfallen können. Auf FreeBSD beispielsweise stehen LZ4, LZJB, GZIP und ZLE zur Verfügung. (The FreeBSD Project, 2019) Diese können für jedes Dataset vom Nutzer frei gewählt werden und zu einem

späteren Zeitpunkt wieder geändert werden. In diesem Falle bleiben bereits existierende Daten standardmäßig unverändert. Lediglich neue Daten werden von der Änderung betroffen. OpenZFS hat kein Problem damit, mit Daten zu arbeiten, welche mit unterschiedlichen Formen von Komprimierung versehen sind, auch wenn diese innerhalb eines Datasets liegen. Bei Bedarf ist es allerdings möglich, auch alte Daten neu zu komprimieren. (Klara Inc, kein Datum)

Auf die Funktionsweise der einzelnen Komprimierungsalgorithmen soll an dieser Stelle nicht genauer eingegangen werden, da diese nicht Inhalt der Arbeit sind.

2.2.2 Deduplication

OpenZFS unterstützt Deduplication auf der Blockebene. Hierbei werden, grob betrachtet, alle Blöcke von Informationen, welche innerhalb des Systems gespeichert werden, mittels Prüfsummen analysiert. Auf die hierfür verwendeten Prüfsummen wird später weiter eingegangen.

Identische Blöcke müssen so lediglich einmal gespeichert werden. Dieser einzelne gespeicherte Block wird dann an allen Stellen, an welchen ein identischer Block liegen sollte, lediglich als Referenz angegeben. So kann, abhängig von den gespeicherten Daten, Speicherplatz eingespart werden. Besonders regelmäßige Backups von geringfügig veränderten Dateien und Images von virtuellen Maschinen können so besonders klein gehalten werden. Im Falle von OpenZFS werden diese Blöcke bereits vor dem Schreiben auf die Disks gebildet, es handelt sich also um synchrone Deduplication. So werden unnötige Schreibvorgänge auf die Disks vermieden. Allerdings werden so die Schreibanweisungen abhängig von der Geschwindigkeit, in der das Host-System diese deduplizieren kann. (Bonwick, 2009)

Da Seitens des OpenZFS Projektes keine genauen Angaben zur hierdurch erlangten Ersparnis von Speicherplatz gemacht werden, folgt eine kurze Beispielrechnung:

Angenommen Unternehmen A und Unternehmen B haben je 10 Mitarbeiter, welche eine virtuelle Maschine benötigen. Unternehmen B verwendet Deduplication, Unternehmen A nicht. Auf den Hypervisoren beider Unternehmen werden nun je 10 identische Instanzen von Windows 10 eingerichtet, welche je 20GB Speicherplatz benötigen. (Microsoft Corporation, Windows 10 system requirements, kein Datum) Unternehmen A benötigt hierfür 200GB an Speicherplatz, Unternehmen B dagegen benötigt unter Idealbedingungen lediglich 20GB.

Angenommen alle benötigen nun 5GB an zusätzlicher Software. Unternehmen A muss hierfür 50GB an Speicherplatz einplanen, Unternehmen B dagegen benötigt unter Idealbedingungen lediglich circa 5GB.

Daraus ergibt sich in beiden Beispielen eine Ersparnis von circa 90% des benötigten Speicherplatzes.

Auf die genaue Implementierung der Deduplication und deren Einschränkungen soll an dieser Stelle nicht eingegangen werden.

2.2.3 Transaktionelles Speichermodell

OpenZFS verwendet ein transaktionelles Speichermodell. Änderungen am Dateisystem und den sich darauf befindlichen Daten werden also zunächst in Transaktionsgruppen gesammelt, bevor sie auf die Disks geschrieben werden. (Delphix Engineering, 2012) Da Transaktionen immer entweder als vollständig abgeschlossen oder vollständig fehlgeschlagen angesehen werden können, wird so die Korruption des Dateisystem im Falle eines unerwarteten Herunterfahrens verhindert. (Isaac Computer Science, kein Datum)

Auf die interne Funktionsweise der Transaktionsgruppen und deren Lebenszyklus wird im Sinne der Lesbarkeit nicht weiter eingegangen.

2.2.4 Copy-on-Write-Semantik

OpenZFS operiert nach dem Redirect-on-Write-Prinzip, (RAID Incorporated, 2020), auch wenn dies oftmals als Copy-on-Write-Prinzip (Oracle Corporation, What Is Oracle Solaris ZFS?, kein Datum) angegeben wird. In beiden Fällen werden veränderte Daten niemals direkt überschrieben. Der Unterschied besteht darin, dass nach dem Copy-on-Write-Prinzip Daten vor einer Veränderung zunächst in leere Sektoren kopiert werden, bevor die ursprünglichen Daten überschrieben werden. Im Falle von OpenZFS werden die veränderten Daten zunächst an eine zusätzliche, als ungenutzt markierte, Speicherstelle geschrieben. Anschließend werden die Referenzen auf die Daten verändert, um auf die neuen Daten zu zeigen. (RAID Incorporated, 2020)

Zu beachten ist, dass die alten Daten hierbei völlig unberührt bleiben. So ist es im Falle eines unvollständigen Schreibauftrags möglich, den Zustand der bearbeiteten Daten, auf den Ausgangszustand zurückzusetzen. Dies bietet enorme Vorteile für andere Funktionalitäten von OpenZFS, wie Snapshots und Clones, welche in den folgenden Abschnitten erklärt werden.

2.2.5 ZFS Intent Log

Das „ZFS Intent Log“, oder kurz „ZIL“ dient als Puffer von Transaktionen, welche synchron stattfinden müssen. Dort werden diese zunächst gesammelt und anschließend auf die eigentlichen physischen Disks geschrieben. (Oracle Corporation, Differences Between Synchronous and Asynchronous I/O, kein Datum)

Das Übertragen von ZIL auf die physischen Disks kann anschließend asynchron stattfinden, da die Daten bereits auf dem ZIL als vollständig persistent gespeichert angesehen werden. (iX Systems, 2015)

Auf die Lagerung und Geschwindigkeit des ZIL wird im Aufbau von OpenZFS weiter eingegangen.

2.2.6 Selbstheilende Daten mittels Ende-zu-Ende Prüfsummen

Ein weiterer Fokus von OpenZFS liegt darin dem sogenannten „Bit Rot“, also dem Datenverlust durch den physischen Verfall von Materialien, entgegenzuwirken. (Tunstall, 2016)

Hierfür, verwendet OpenZFS Prüfsummen, welche mit den Metadaten gelagert werden. Diese ermöglichen es, bei jedem Lesen eines Blockes zu verifizieren, ob dieser noch fehlerfrei vorliegt, oder durch Hardware- bzw. Softwarefehler oder äußere Einwirkungen verändert wurde. Solche Checks finden sowohl während der normalen Nutzung der Daten, also beim Zugriff durch den Nutzer, als auch periodisch als Teil sogenannter „Scrubs“ statt. Wenn ein Block als fehlerhaft identifiziert worden ist, kann dieser, bei gegebener Resilienz, automatisch wiederhergestellt werden. Sollte keinerlei Resilienz vorliegen, ist das System lediglich in der Lage auf den fehlerhaften Block hinzuweisen, dies sollte bei der Einrichtung in Betracht gezogen werden. (Oracle Corporation, Creating and Destroying ZFS Storage Pools, kein Datum)

Es kann aus einer Vielzahl unterschiedlicher Prüfsummenalgorithmen ausgewählt werden. Standardmäßig verwendet OpenZFS aktuell entweder „sha256“ oder „fletcher4“, wobei ersteres nur automatisch für Datasets mit aktivierter Deduplication angewandt wird. (OpenZFS Project, Checksums and Their Use in ZFS, kein Datum)
Anders als traditionelle Dateisysteme lagert OpenZFS die Prüfsummen nicht mit den Daten, sondern mit den Metadaten. Dadurch soll verhindert werden, dass die Prüfsummen zusammen mit den Daten beschädigt werden. (OpenZFS Project, Checksums and Their Use in ZFS, kein Datum)

2.2.7 Dynamische Allokation

OpenZFS allokiert den für seine Nutzer- und Metadaten benötigten Speicherplatz dynamisch mit der Erzeugung neuer Daten. So gibt es keine im Voraus determinierten Limits für die Anzahl von Dateien und Directories oder deren Größe, und das Erstellen eines Dateisystems benötigt keinen initialen Speicherplatz. (Oracle Corporation, Replication Features of a ZFS Storage Pool, kein Datum)
Auch das Hinzufügen von zusätzlichem Speicherplatz wird so vereinfacht, da Blöcke erst während des Schreibens von Daten allokiert werden müssen. So kann das Dateisystem problemlos dynamisch auf zusätzliche Disks gestreckt werden. (Oracle Corporation, Replication Features of a ZFS Storage Pool, kein Datum)

2.3 Aufbau

OpenZFS kombiniert ein Dateisystem mit einem Volumenmanager, dadurch hat das System sowohl Einsicht auf die physischen Speichermedien und deren Layout als auch auf die sich darauf befindlichen Daten. (Oracle Corporation, Traditional Volume Management, kein Datum)
Dies ist vor allem daher interessant, da so jeder Bestandteil der Datenhaltung direkt von einem zentralen Punkt auf Fehler überprüft werden kann. Somit wird es ermöglicht Schreibfehler sowohl auf Hardware-, als auch auf Softwareebene zu erkennen und zu korrigieren. (Salter, ZFS 101—Understanding ZFS storage and performance, 2020)

Durch diese weite Einsicht von OpenZFS wird es ermöglicht, im Falle eines Ausfalls, nur tatsächlich mit Dateien beschriebene Blöcke wiederherstellen zu müssen. Leere Sektoren der Disks werden übersprungen. So wird das Wiederherstellen der

Resilienz eines Systems mit viel freiem Speicherplatz erheblich beschleunigt. Zu beachten ist hierbei, dass dieser Vorteil gegenüber traditionellem RAID mit zunehmender Speicherauslastung verschwindet. In diesem Falle ist OpenZFS oft langsamer als traditionelles RAID, da die Blöcke der Disks nicht sequenziell gelesen bzw. geschrieben werden, sondern in der Reihenfolge, wie sie von den Daten beschrieben worden sind. (Gubin, 2019)

Auf die spezifische Funktionsweise des Festplattenzugriffs soll, aufgrund der Lesbarkeit dieser Arbeit, nicht weiter eingegangen werden.

2.3.1 Storage VDEV

Architektonisch besteht ein OpenZFS-System zunächst aus den sogenannten „Virtual Devices“, oder kurz „VDEV“. Jedes VDEV kann aus einer beliebigen Anzahl an blockbasierten Speichermedien, im Folgenden als „Disks“ bezeichnet, bestehen. Diese Disks können von beliebiger Größe über 128 Megabyte sein, der nutzbare Speicherplatz ist allerdings je nach Konfiguration abhängig von der kleinsten Disk. Es wird daher davon abgeraten, Kapazitäten innerhalb eines VDEV zu mischen. (Oracle Corporation, Storage Configuration Rules and Guidelines, kein Datum)
Unterschieden wird hierbei zwischen Storage VDEV und Support VDEV. Erstere beinhalten die eigentlichen Daten, welche auf dem Speichersystem abgelegt werden, während letztere zusätzliche Daten wie Cache und Metadaten beherbergen. (Salter, ZFS 101—Understanding ZFS storage and performance, 2020)

Jegliche Resilienz von OpenZFS besteht auf dem VDEV-Level. VDEV können hierbei in verschiedenen Topologien konfiguriert werden. (Salter, ZFS 101—Understanding ZFS storage and performance, 2020)

2.3.1.1 Basic Topologie

Die Basic Topologie (Oracle Corporation, Creating and Destroying ZFS Storage Pools, kein Datum) oder auch Single-Device Topologie, unterstützt lediglich ein einzelnes physisches Gerät. Sie bietet demzufolge keinerlei Resilienz. (Salter, ZFS 101—Understanding ZFS storage and performance, 2020)

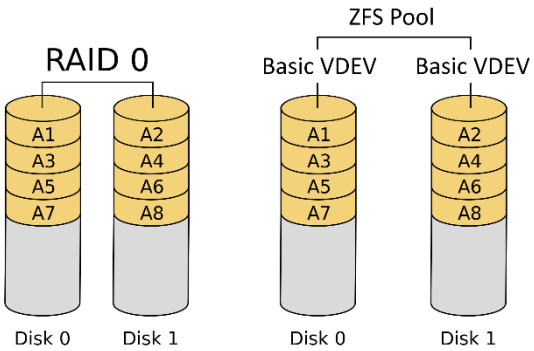

Abbildung 1: RAID0 im Vergleich zu ZFS Basic Topologie, verändert nach (Burnett, 2006) sowie (Kiljan, 2018).

2.3.1.2 Mirror Topologie

Die Mirror Topologie ist an ein traditionelles RAID1 System angelehnt. Es werden alle Daten auf allen Disks innerhalb eines Mirror VDEV repliziert. So kann innerhalb des VDEV vor Datenverlust geschützt werden, solange noch mindestens eine einzelne fehlerfreie Disk vorhanden ist. Die Kapazität des VDEV entspricht hierbei, wie auch in einem RAID1, der einer einzelnen Disk. Die Lesegeschwindigkeit kann allerdings ebenfalls um einen Faktor, welcher theoretisch äquivalent mit der Anzahl der Disks ist, erhöht werden. (Oracle Corporation, Creating and Destroying ZFS Storage Pools, kein Datum)

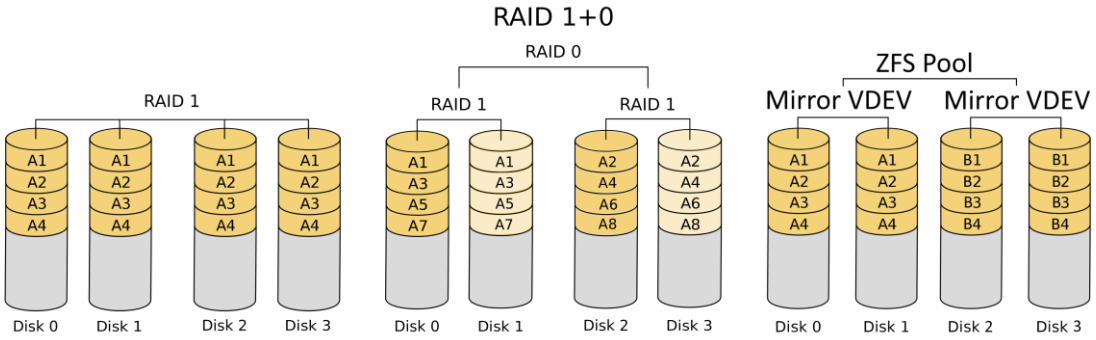

Abbildung 2: RAID1 und RAID10 im Vergleich zu ZFS Mirror Topologie, verändert nach (Burnett, 2006), (Kiljan, 2018) sowie (NudelSuppe, 2012).

Ein VDEV mit Mirror Topologie kann auch im Nachhinein, um weitere Disks erweitert werden, um dessen Resilienz zu erhöhen. (Oracle Corporation, Managing Devices in ZFS Storage Pools, kein Datum)

2.3.1.3 RAID-Z Topologie

Ein RAID-Z VDEV gewährleistet Ausfallsicherheit mittels Paritätsdaten und operiert nach dem Prinzip der diagonalen Parität. (Salter, ZFS 101—Understanding ZFS storage and performance, 2020) Hierbei kann zwischen einfacher (RAIDZ-1, ähnlich zu RAID5), doppelter (RAIDZ-2, ähnlich zu RAID6) oder dreifacher (RAIDZ-3) Parität gewählt werden.

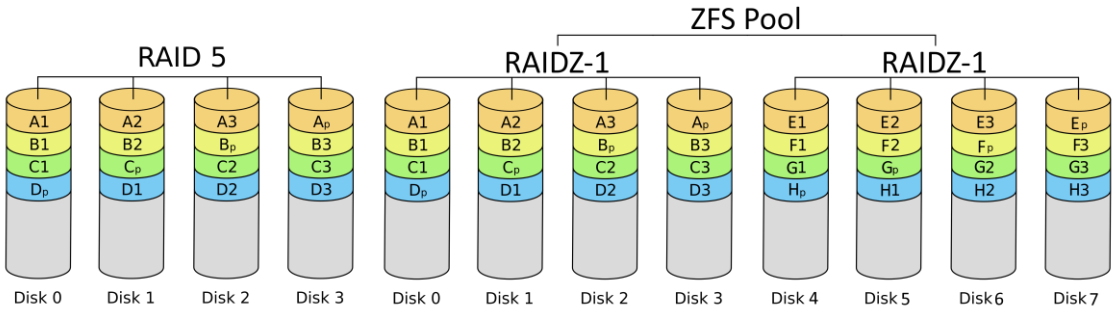

Abbildung 3: RAID5 im Vergleich zu RAIDZ-1, verändert nach (Burnett, 2006) sowie (Kiljan, 2018).

Ein RAID-Z VDEV ist resilient gegenüber dem Ausfall von einer Anzahl von Disks, welche äquivalent zur Anzahl der Paritäten ist. Sollte während des Prozesses der Neuberechnung der Daten aus der Parität eine weitere Disk ausfallen, kommt es zu Datenverlust innerhalb des VDEV. Aus diesem Grund wird oftmals von RAIDZ-1-Layouts abgeraten, vor allem bei größeren einzelnen Speichermedien. Dies kommt daher, dass selbst moderne Festplatten mit 7200 RPM Rotationsgeschwindigkeit gerade einmal Schreibgeschwindigkeiten von 80-160 Megabyte die Sekunde aufweisen, (SSD vs HDD Speed, Lifespan, and Reliability, 2019) bei den heutzutage üblichen Kapazitäten von mehreren Terabyte (Alsop, 2021), können also mehrere Stunden oder sogar Tage benötigt werden, um eine neue Disk nach einem Ausfall wieder vollständig mit den nötigen Daten zu beschreiben.

Oracle empfiehlt für RAID-Z Konfigurationen, die Anzahl der Disks je VDEV in einem einstelligen Bereich zu halten. Statt mehr Disks pro VDEV wird ab neun Disks dazu geraten, mehr VDEV zu verwenden. (Oracle Corporation, Replication Features of a ZFS Storage Pool, kein Datum)

Ursprünglich war es nicht möglich ein RAID-Z VDEV, um zusätzliche Disks zu erweitern. Mitte des Jahres 2021 wurde von Matthew Ahrens, einem der Gründer des OpenZFS Project, die Erweiterung von RAID-Z VDEV, als neues Feature für zukünftige Versionen von OpenZFS, angekündigt. Aufgrund der Tatsache, dass dieser Code aktuell noch nicht Teil der regulär verfügbaren Versionen von OpenZFS ist, und sich dessen genaue Implementation demzufolge jederzeit ändern könnte, wird an dieser Stelle nicht weiter darauf eingegangen. (Salter, ZFS fans, rejoice—RAIDz expansion will be a thing very soon, 2021)

2.3.1.4 Distributed RAID (dRAID) Topologie

Distributed RAID oder kurz dRAID ist die neueste Topologie, welche von OpenZFS angeboten wird. Fundamental ist sie ebenfalls ähnlich zu RAID5 und damit auch RAID-Z, mit dem signifikanten Unterschied, dass dRAID auch Spare-Disks in Form

von diagonal angeordneten Blöcken anbietet. Es gibt also keine physische Reserve, welche im Falle eines Ausfalls mit den aus der Parität errechneten Daten beschrieben wird und somit direkt mit einer enormen Schreiblast belegt wird. Stattdessen existieren auf allen Disks freie Blöcke, welche im Falle eines Ausfalls simultan beschrieben werden können. So lässt sich der Prozess des Wiederherstellens der Resilienz nach einem Ausfall erheblich beschleunigen. dRAID Topologien werden mit drei Zahlen gekennzeichnet, die Erste steht hierbei für die Anzahl an Paritäts-Disks, die Mittlere für die Anzahl der Daten-Disks und die Letzte für die Anzahl der Spare-Disks. (Salter, OpenZFS 2.1 is out—let's talk about its brand-new dRAID vdevs, 2021)

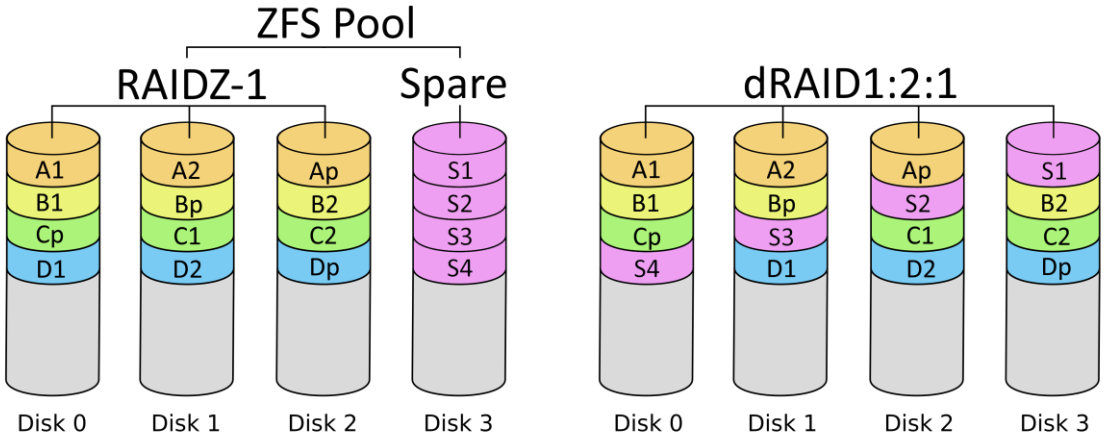

Abbildung 4: RAIDZ-1 im Vergleich zu dRAID1:2:1, verändert nach (NudelSuppe, 2012) sowie (Salter, OpenZFS 2.1 is out—let's talk about its brand-new dRAID vdevs, 2021)

Es ist zu beachten, dass der Prozess des Neuverteilens der Daten nach dem Austausch der ausgefallenen Disks keinen Geschwindigkeitsvorteil gegenüber RAID-Z hat, da hierbei, wie bei traditionellem RAID, die gesamte Disk sektorenweise übertragen wird. In diesem Zustand ist die Resilienz allerdings bereits wiederhergestellt, die Operation kann also, als wesentlich weniger zeitkritisch betrachtet werden. (Salter, OpenZFS 2.1 is out—let's talk about its brand-new dRAID vdevs, 2021)

Da die dRAID-Topologie primär für Datacenter-Anwendungen mit um die 90 Disks interessant ist, (Salter, OpenZFS 2.1 is out—let's talk about its brand-new dRAID vdevs, 2021) wird in dieser Arbeit jedoch nicht weiter auf dRAID eingegangen.

2.3.2 Support VDEV

Anders als Storage VDEV dienen Support VDEV nicht der Speicherung der eigentlichen Nutzerdaten, sondern zusätzlicher, vom System benötigter Daten. (Salter, ZFS 101—Understanding ZFS storage and performance, 2020)

Abgesehen von den abgelegten Daten unterscheiden sich Support VDEV prinzipiell nicht von Storage VDEV. Auch sie können mit verschiedenen Topologien eingerichtet werden und dementsprechend verschiedene Formen von Resilienz bieten. Hierbei gibt es allerdings je nach Typ einige Einschränkungen, auf welche im Folgenden eingegangen wird. (Oracle Corporation, Creating and Destroying ZFS Storage Pools, kein Datum)

Im Folgenden ist zu beachten, dass in OpenZFS verwendete SSDs unbedingt in der Lage seien müssen, Daten immer persistent zu speichern. (Kennedy, 2017) Die genauen technischen Voraussetzungen, welche hierbei gelten, sind allerdings nicht Inhalt dieser Arbeit.

2.3.2.1 Cache VDEV

Standardmäßig verwendet OpenZFS einen Teil des Arbeitsspeichers seines Host-Systems als Cache für Daten, auf welche es häufige Lesezugriffe gibt. Dieser kann jedoch durch Cache VDEV erweitert werden. Es können beliebig viele Cache VDEV erstellt werden, allerdings unterstützen diese lediglich die Basic-Topologie. (Oracle Corporation, Creating a ZFS Storage Pool With Cache Devices, kein Datum)
Dies stellt allerdings kein Risiko für die Datensicherheit dar, da der Cache von OpenZFS, als instabil angesehen wird. Es handelt sich bei den Daten im Cache, also ausschließlich um Kopien bereits vorhandener Daten. (OpenZFS Project, System Administration, kein Datum)

2.3.2.2 Log VDEV

Das Log VDEV, eigentlich „Separate Intent LOG SSD“, oder kurz SLOG genannt, bietet die Möglichkeit das ZIL auf einem separaten Speichermedium zu lagern. (Kennedy, 2017) Wenn kein Log VDEV existiert, werden Blöcke aus den Storage VDEV dazu verwendet. Dies kann je nach Konfiguration Einbußen in Sachen Performanz mit sich bringen. (iX Systems, 2015)

Das Verwenden eines dedizierten Log VDEV, vorzugsweise bestehend aus besonders performanten Disks wie z.B. SSDs, kann die Performanz beim synchronen Schreiben erheblich erhöhen. (Oracle Corporation, Creating a ZFS Storage Pool With Log Devices, kein Datum) Dies beschleunigt, bei korrekter Konfiguration, die von Endnutzer und Programmen wahrgenommene Interaktion mit dem Speichersystem, ohne die Integrität der gespeicherten Daten zu gefährden. (OpenZFS Project, System Administration, kein Datum)

Für das Log VDEV werden lediglich Basic- oder Mirror-Topologien unterstützt. (Oracle Corporation, Creating a ZFS Storage Pool With Log Devices, kein Datum)
Als maximale Größe hierfür wird in der Dokumentation für Oracle ZFS unter Solaris circa die Hälfte des physischen Arbeitsspeichers empfohlen. Dies ist allerdings lediglich eine Empfehlung, keine feste Vorgabe. (Oracle Corporation, Creating a ZFS Storage Pool With Log Devices, kein Datum)

Ein Log VDEV kann einem Pool sowohl bei seiner Erstellung als auch im Nachhinein hinzugefügt werden. (Oracle Corporation, Creating and Destroying ZFS Storage Pools, kein Datum) Sollte das Log VDEV ausfallen, so wird das ZIL automatisch auf den Storage VDEV des Pools neu angelegt. Hierbei gehen allerdings alle Transaktionen verloren, welche noch nicht vollständig aus dem ZIL auf die Storage VDEV übertragen wurden. (OpenZFS Project, System Administration, kein Datum)

2.3.2.3 Special VDEV

Das Special VDEV beinhaltet die Metadaten der eigentlichen Nutzerdaten. Sollte kein Special VDEV vorhanden sein, so werden Metadaten innerhalb der Storage VDEV des Pools gelagert. Durch das Verwenden von SSDs, innerhalb des Special VDEV kann das Durchlaufen des Speicherbaumes erheblich beschleunigt werden.
Hierbei ist unbedingt zu beachten, dass der Ausfall eines Special VDEV nicht von ZFS toleriert werden kann, es ist also essenziell an dieser Stelle eine Topologie mit besonders hoher Resilienz zu verwenden. (Salter, ZFS 101—Understanding ZFS storage and performance, 2020)

2.3.2.4 Spare VDEV

Das Spare VDEV beinhaltet alle Disks, welche einem Pool zur Verfügung stehen, um im Falle eines Ausfalles die Rolle der beschädigten Disks zu übernehmen. Im Falle eines Ausfalls, werden diese dem betroffenen VDEV automatisch hinzugefügt. Anschließend beginnt der Prozess der Wiederherstellung der Daten. Dieses Verhalten erfordert typischerweise keinerlei Interaktion vom Nutzer und geschieht automatisch. Sobald die ausgefallene Disk im betroffenen VDEV ausgetauscht worden ist, werden die Daten von der Disk aus dem Spare VDEV auf die neue Disk übertragen. Anschließend wird sie wieder Teil des Spare VDEV und steht wieder allen VDEV des Pools zur Verfügung. (Salter, ZFS 101—Understanding ZFS storage and performance, 2020)

2.3.3 ZFS Pool (zPool)

Ein OpenZFS Pool oder kurz „zPool“ wird aus einer beliebigen Anzahl von VDEV gebildet.
Ein zPool kann, durch das Hinzufügen, Entfernen oder Modifizieren von enthaltenen VDEV beliebig vergrößert bzw. verkleinert werden. Es können beliebig viele zPools auf einem Host-System existieren, diese sind allerdings völlig unabhängig voneinander. Es ist nicht möglich Storage VDEV oder Support VDEV zwischen zPools zu teilen. (Oracle Corporation, Creating ZFS Storage Pools, kein Datum)

Der zPool setzt das Konzept von Pool-basiertem Speicher um, welcher dynamisch zwischen den sich darauf befindlichen Datasets aufgeteilt werden kann. Die auf dem Pool abgelegten Daten werden dabei algorithmisch so auf den dem zPool zugeordneten VDEV verteilt, dass deren Speicherplatz möglichst gleichmäßig ausgelastet wird. Des Weiteren wird in moderneren Versionen von OpenZFS oftmals auch die momentane Auslastung der Schreib- und Lesebandbreite der jeweiligen VDEV in Betracht gezogen, um die Performanz des zPools als Ganzes möglichst stabil zu halten. (Salter, ZFS 101—Understanding ZFS storage and performance, 2020)

Unbedingt zu beachten ist, dass der zPool selbst keinerlei Ausfallsicherheit bietet. Die VDEV werden innerhalb eines zPools, ähnlich einem RAID0, zusammengefasst. Die Resilienz eines zPools aus Mirror VDEV ist also beispielsweise äquivalent zu einem RAID10, und die eines zPool aus RAIDZ-1 äquivalent zu einem RAID50. Dies hat zur Folge, dass beim Ausfall eines einzelnen Storage VDEV innerhalb eines zPools, auch der gesamte zPool einen Datenverlust erleidet. (Salter, ZFS 101—Understanding ZFS storage and performance, 2020)

Daher ist es von größter Wichtigkeit, innerhalb eines zPools nur VDEV mit angemessener Ausfallsicherheit für die darauf zu speichernden Daten zu verwenden. In der Dokumentation für Oracle ZFS unter Solaris wird empfohlen, innerhalb eines Pools nur VDEV mit identischer Ausfallsicherheit zu verwenden, dies ist allerdings keine feste Vorgabe. (Oracle Corporation, Replication Features of a ZFS Storage Pool, kein Datum)

2.3.4 Datasets

Innerhalb eines zPools können beliebig viele Datasets existieren. Datasets skalieren dynamisch, der von ihnen benötigte Speicher ändert sich also abhängig von den Daten. (Salter, ZFS 101—Understanding ZFS storage and performance, 2020)
Hierbei können allerdings Einschränkungen in Form von Quoten getroffen werden, um ein bestimmtes Dataset auf eine Maximalgröße zu beschränken. Ebenfalls ist es möglich, einem Dataset ein Minimum an Speicherplatz zu garantieren, welches von anderen respektiert wird. (Oracle Corporation, Setting ZFS Quotas, kein Datum)

2.3.4.1 Dateisysteme

Dateisysteme sind der zentrale Punkt der Verwaltung von OpenZFS. Sie sind dafür entwickelt, vom Nutzer verwendet zu werden, wie herkömmliche Ordner. Sie werden in hierarchischen Strukturen angeordnet, wie Ordner in herkömmlichen Dateisystemen. Oracle rät dazu, viele kleinere Dateisysteme zu verwenden, anstelle eines großen, monolithischen Dateisystems, auf welchem alle Daten abgelegt werden. Beispielsweise kann für jeden Nutzer oder jedes Projekt ein Dateisystem angelegt werden. (Oracle Corporation, Creating a Basic ZFS File System, kein Datum)

2.3.4.2 Volumen

Ein Volumen, auch „ZVOL“ genannt, simuliert ein blockbasiertes Speichermedium und kann mit anderen Dateisystemen verwendet werden. Dies ist interessant, wenn Speicherplatz für Systeme bereitgestellt werden soll, welche OpenZFS selbst nicht unterstützen. (Salter, ZFS 101—Understanding ZFS storage and performance, 2020)

2.3.4.3 Snapshots

Zusätzlich zu Kopien bietet OpenZFS die Möglichkeit Snapshots zu erstellen. Hierbei handelt es sich um schreibgeschützte Kopien eines vorhandenen Datensets. Anders als Kopien nutzen diese die Copy-on-Write-Semantik von OpenZFS aus. Sie benötigen daher keinerlei zusätzlichen initialen Speicherplatz. Erst wenn das zugehörige Dataset modifiziert wird, werden neue Blöcke mit den veränderten Daten beschrieben und es erhöht sich der benötigte Speicherplatz für den Snapshot. (Salter, ZFS 101—Understanding ZFS storage and performance, 2020)

2.3.4.4 Clones

Ein Clone ist eine veränderbare Kopie eines bereits vorhandenen Datasets, welche aus einem Snapshot dieses Datasets erstellt wird. Wie auch der Snapshot selbst, benötigt ein Clone keinerlei initialen Speicherplatz, lediglich vorgenommene Änderungen müssen gesondert gespeichert werden. (Oracle Corporation, Overview of ZFS Clones, kein Datum)

2.4 Bedienung

2.4.1 Benutzeroberfläche

Aufgrund der vielen verschiedenen Implementationen von OpenZFS, gibt es keine einheitliche Benutzeroberfläche. Viele Implementationen setzen hierbei primär auf die Kommandozeile, wobei auch hierbei nicht immer gegeben ist, dass alle Kommandos gleich vorhanden sind bzw. die gleiche Funktionalität aufweisen. Einige Implementationen von OpenZFS, beispielsweise Oracle ZFS, bieten auch grafische Benutzeroberflächen an (Tunstall, 2016) und es existieren diverse Programme von Drittanbietern, welche das Verwalten von OpenZFS vereinfachen sollen, beispielsweise „Zyggy“. (Desouza, 2021) Diese sind allerdings auch nicht immer unabhängig vom Betriebssystem bzw. der OpenZFS Implementierung.

Auf die Benutzeroberflächen und deren Benutzerfreundlichkeit im Sinne der Human-Computer-Interaction wird an dieser Stelle nicht weiter eingegangen, da es nicht der Fokus dieser Arbeit ist.

2.4.2 Sandboxing

Viele der Kommandos für OpenZFS bieten die Möglichkeit eines Testdurchlaufes, einem sogenannten „Dry Run“. Dieser nimmt keine Änderungen am System vor. Es werden lediglich die notwendigen Berechnungen durchgeführt und anschließend mitgeteilt, wie die gewünschte Veränderung sich auf das System auswirken würde. (Oracle Corporation, Doing a Dry Run of Storage Pool Creation, kein Datum)

2.4.3 Erweiterung des Speichers

Zu beachten ist bei der Verwendung von OpenZFS, dass die Erweiterung des Speicherplatzes eines VDEV aktuell nur durch den Austausch aller enthaltenen Disks durch neue mit höherer Kapazität ist. (Mad About Brighton, 2016)
Es ist dank der Verwendung von dynamischer Allokation allerdings problemlos möglich, einem zPool zusätzliche VDEVs hinzuzufügen. So kann dessen maximaler Speicherplatz erhöht werden. Allerdings ist hierbei zu beachten, dass diese VDEV identische Topologien verwenden sollten, wie bereits vorhandene VDEV. (Oracle Corporation, Adding Devices to a Storage Pool, kein Datum)

3 Microsoft Storage Spaces

3.1 Entwicklung

Microsofts Storage Spaces wurde erstmals mit dem Betriebssystem Windows Server 2012 vorgestellt und war als Ersatz für das in Windows Home Server implementierte Drive Extender Feature angedacht. (Cunningham, 2012)
Dieses ermöglichte es, mehrere physische Festplatten zu einem einzelnen Mount-Punkt zusammenzufassen. Hierbei wurden allerdings keinerlei Form von Resilienz zur Absicherung gegen Disk Ausfälle angeboten. Die Daten wurden lediglich über die einzelnen Speichermedien verteilt und für den Nutzer zusammengefasst. So sollte das Verwalten der zunehmend großen Datenmengen für Heimanwender vereinfacht werden. (Hartsock, 2011)

Storage Spaces erweiterte dieses Feature erheblich und bietet eine Vielzahl zusätzlicher, RAID-ähnlicher, Features an. (Cunningham, 2012) Storage Spaces ist heute Teil von allen gängigen Versionen von Windows 10 (Microsoft Corporation, Storage Spaces in Windows 10, kein Datum), sowie allen Versionen von Windows Server seit Windows Server 2012. (Microsoft Corporation, Deploy Storage Spaces on a stand-alone server, 2021) Implementierungen auf anderen Betriebssystemen, wie Linux oder MacOS, konnten zum Zeitpunkt dieser Arbeit nicht gefunden werden.

3.2 Features

3.2.1 ReFS Dateisystem

Das „Resilient File System“ oder kurz „ReFS“ ist kein eigentlicher Bestandteil von Storage Spaces. Es stellt allerdings einen wichtigen Bestandteil des, von Microsoft entwickelten, Ökosystems für Speichervirtualisierung dar. Es wurde von Microsoft im Tandem mit Storage Spaces veröffentlicht, und bietet tiefe Integration mit dem System. (Microsoft Corporation, Resilient File System (ReFS) overview, 2021)

ReFS bietet selbstständig einige ähnliche Features wie OpenZFS an, welche dazu dienen sollen Bit-Rot zu erkennen und entgegenzuwirken. Es ist dabei funktional ähnlich zum Dateisystemteil von OpenZFS. Auch ReFS arbeitet mit separat

gelagerten Prüfsummen. Ein wichtiger Unterschied ist hierbei jedoch, dass ReFS aktuell lediglich Prüfsummen für Metadaten bildet, nicht wie OpenZFS für alle gespeicherten Daten. Auch ReFS durchsucht periodisch mittels Scrubs, das gesamte Dateisystem nach Unstimmigkeiten zwischen Daten und Prüfsummen. ReFS ist ebenfalls in der Lage, die erkannten Fehler selbstständig und ohne Benutzerintervention anhand vorhandener Kopien oder Paritätsdaten zu korrigieren. Sollte dies nicht möglich sein, so ist ReFS in der Lage, die beschädigten Daten aus dem Namensraum des Dateisystems zu entfernen, um potenziell daraus resultierende Fehler zu vermeiden. Dies geschieht in der Regel ohne, dass das betroffene Volumen offline gehen muss. (Microsoft Corporation, Resilient File System (ReFS) overview, 2021)

Zusätzlich dazu bietet ReFS laut Microsoft noch einige andere Vorteile gegenüber älteren Dateisystemen. Auf diese wird allerdings nicht genauer eingegangen.

3.2.2 Deduplication

Deduplication ist kein Bestandteil von Storage Spaces, sondern Teil des unterliegenden Betriebssystems. Dieses Feature ist ausschließlich auf Server-Versionen von Windows verfügbar. (Microsoft Corporation, Data Deduplication Overview, 2017)

Microsoft verwendet hierbei asynchrone Deduplication. Daten werden also zunächst auf die Disks geschrieben, und anschließend periodisch auf Duplikate überprüft. So sind Schreibanweisungen nicht abhängig von der Geschwindigkeit, in welcher die Duplikate identifizieret und verarbeitet werden können. Jedoch wird initial mehr Speicherplatz benötigt, bis die Daten dedupliziert wurden.

Ein weiterer Unterschied zu OpenZFS ist, dass Microsofts Deduplication auf Volumenebene stattfindet, sich also die zu deduplizierenden Daten alle innerhalb eines Volumens befinden müssen.

Laut Microsoft sollte, wenn Deduplication verwendet wird pro 1TB Speicherplatz 1GB an Arbeitsspeicher zur Verfügung gestellt werden. Zu beachten ist außerdem, dass Volumen hierbei maximal 64TB an Speicherplatz beinhalten können. (Microsoft Corporation, Install and enable Data Deduplication, 2017)

3.2.3 Thin Provisioning

Storage Spaces unterstützt sowohl traditionelles „Thick“ oder „Fixed“ Provisioning als auch „Thin“ Provisioning. Dies ist der Funktionsweise der dynamischen Allokation von OpenZFS sehr ähnlich. Auch hierbei werden die eigentlichen Blöcke auf den Disks erst dann allokiert, wenn der Speicherplatz tatsächlich von Daten erfordert wird. (Microsoft Corporation, Thin Provisioning, 2019)

So ist es möglich Storage Spaces mit größerer Kapazität anzulegen, als tatsächlich physisch vorhanden ist. Physischer Speicherplatz kann bei Bedarf durch zusätzliche Disks hinzugefügt werden. Auf diese Weise können beispielsweise 10 Mitarbeiter je einen Storage Space von 500GB zugeteilt bekommen, obwohl physisch lediglich 2TB zur Verfügung stehen. (Microsoft Corporation, Thin Provisioning, 2019)

3.3 Aufbau

Anders als OpenZFS beinhaltet Storage Spaces kein eigenes Dateisystem. Es besteht allerdings eine tiefere Integration mit den NTFS- und ReFS-Dateisystemen. Storage Spaces ist ein blockbasiertes System. Alle Daten, welche innerhalb eines Storage Spaces gespeichert werden, werden in sogenannte „Slabs“, oft auch „Interleaves“ oder „Stripes“ genannt, zerteilt. Slabs stellen innerhalb des Storage Space-Systems die kleinste mögliche Einheit von Daten dar. Ein Slab muss also immer vollständig auf einer physischen Disk abgelegt werden. (Dell Technologies Inc., Interleave size, kein Datum) Slabs sind standardmäßig 256KB groß, können aber entsprechend der erwarteten Workload angepasst werden. (Finn, 2015)

Auf solches Tuning wird jedoch innerhalb dieser Arbeit nicht weiter eingegangen, da dies den Rahmen einer Bachelorarbeit sprengen würde.

3.3.1 Physische Disks

Storage Spaces ist in der Lage mit einer Vielzahl verschiedener Typen von physischen Disks zu interagieren, diese müssen jedoch jeweils mindestens 4GB an Speichergröße aufweisen. Für die Interfaces USB, SAS und SATA besteht voller Support aller Features. Geräte mit iSCSI oder Glasfaser (FC) Anbindung werden zwar prinzipiell unterstützt, allerdings lediglich für Storage Spaces ohne Resilienz. (Microsoft Corporation, Deploy Storage Spaces on a stand-alone server, 2021)

3.3.2 Storage Pool

Das Grundelement des Storage Spaces-Systems sind Storage Pools. In diesen werden eine beliebige Anzahl Disks zusammengefasst. Speichergrößen können hierbei frei kombiniert werden. Eine wichtige Einschränkung besteht jedoch darin, dass innerhalb eines Storage Pools nur Disks mit identischer logischer Sektorengröße existieren können. (Dell Technologies Inc., Logical sector size, kein Datum)

Disks können bei der Erstellung eines Storage Pools so konfiguriert werden, dass Storage Spaces diese automatisch verwalten kann. Alternativ können Disks aber auch so eingerichtet werden, dass diese manuell durch den Administrator verwaltet werden müssen. Außerdem gibt es an dieser Stelle noch die Möglichkeit Disks als „Hot-Spare“ zuzuweisen. Hot-Spare-Disks geben dem Storage Pool die Möglichkeit, im Falle des Ausfalls von Disks, diese automatisch zu ersetzen. Dies funktioniert jedoch nur insofern die Kapazität der Hot-Spare-Disk mindestens äquivalent zur Kapazität der ausgefallenen Disk ist. (IT Free Training, 2014)

3.3.3 Storage Space

Virtuelle Festplatten, oder auch oft „Storage Spaces“ genannt, definieren das eigentliche Layout des Storage Spaces und die damit verbundene Resilienz. (Microsoft Corporation, Deploy Storage Spaces on a stand-alone server, 2021)
Auf einem Storage Pool können beliebig viele Storage Spaces angelegt werden, welche je unterschiedliche Formen von Resilienz bieten können. (Cunningham, 2012) Es sind zum Zeitpunkt dieser Arbeit folgende Layouts verfügbar:

3.3.3.1 Simple-Layout

Ein Storage Space mit Simple-Layout verteilt die Slabs auf alle vorhandenen Speichermeiden ohne jegliche Resilienz. Dies ähnelt einem traditionellen RAID0. Ähnlich wie RAID0 soll hierbei lediglich die Kapazität als auch die Performanz des Speichers optimiert werden. Hierbei wird nur eine Disk benötigt. (Microsoft Corporation, Deploy Storage Spaces on a stand-alone server, 2021)

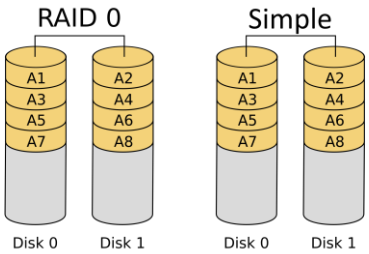

Abbildung 5: Aufbau von RAID0 und Storage Spaces Simple verglichen, modifiziert nach (Burnett, 2006).

3.3.3.2 Mirror-Layout

Ein Storage Space mit Mirror-Layout speichert für jeden Slab eine zusätzliche Kopie. Diese zwei identischen Slabs werden auf zwei unterschiedlichen Disks gespeichert. Hierbei werden mindestens zwei Disks benötigt. Zu beachten ist, dass die identischen Slabs nicht alle auf zwei jeweils eindeutig zueinander zuordenbaren Disks abgelegt werden. Stattdessen werden die Slabs frei auf allen Disks verteilt, mit der einzigen Einschränkung, dass ein Slab und seine zugehörige Kopie niemals auf derselben Disk gelagert werden können. So soll die Auslastung gleichmäßig verteilt werden und die Performanz erhöht werden. (Microsoft Corporation, Deploy Storage Spaces on a stand-alone server, 2021)

Durch diesen Aufbau gewährleistet der Storage Space Resilienz gegenüber einem einzelnen Disk Ausfall ähnlich zu der von traditionellem RAID1. Da die Anzahl der Kopien nicht, wie bei einem RAID1, mit der Anzahl der Disks skaliert, hat ein solcher Storage Space eine äquivalente Speichereffizienz zu einem RAID10 mit je zwei Disks in jeder RAID1 Gruppe. (Microsoft Corporation, Deploy Storage Spaces on a stand-alone server, 2021)

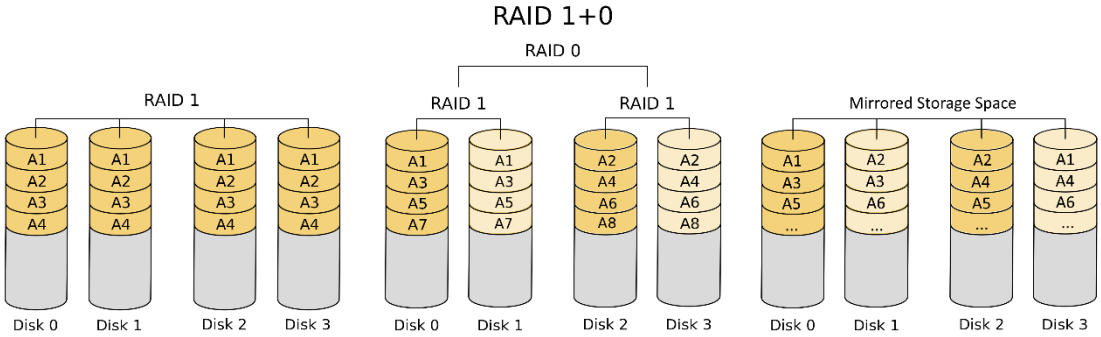

Abbildung 6: Aufbau von RAID0, RAID10 und Storage Spaces Mirror verglichen, modifiziert nach (Burnett, 2006) sowie (NudelSuppe, 2012).

Microsoft empfiehlt dieses Layout für die meisten Anwendungen, da es den größten Schutz vor Datenverlust bietet. (Microsoft Corporation, Deploy Storage Spaces on a stand-alone server, 2021)

3.3.3.3 Three-Way Mirror-Layout

Ein Storage Space mit einem Three-Way Mirror Layout funktioniert nahezu identisch zu einem Storage Space mit einem Mirror-Layout. Der Unterschied besteht darin, dass in diesem Falle drei, anstatt zwei Versionen jedes Slabs verteilt werden. So kann die Resilienz weiter erhöht werden. Es ist aktuell nicht möglich, die Anzahl der zu speichernden Kopien über drei hinaus zu erhöhen. Für dieses Layout werden mindestens fünf Disks benötigt. (Microsoft Corporation, Deploy Storage Spaces on a stand-alone server, 2021)

3.3.3.4 Parity Layout

Ein Storage Space mit Parity-Layout arbeitet nach dem Paritätsprinzip, ähnlich wie RAID5 und bietet damit Resilienz gegenüber dem Ausfall einer einzelnen Disk. Hierbei werden mindestens 3 Disks benötigt.

Microsofts Dokumentation rät dazu, Paritätslayouts nur auf Storage Spaces zu verwenden, welche vorwiegend für stark sequenzielle Daten genutzt werden. (Microsoft Corporation, Deploy Storage Spaces on a stand-alone server, 2021) Dies kommt daher, dass die Performanz von zufälligen Schreibanweisungen mitunter weit unterhalb der eines Mirror-Layouts liegt. (Microsoft Storage Spaces Is Hot Garbage For Parity Storage, 2018)

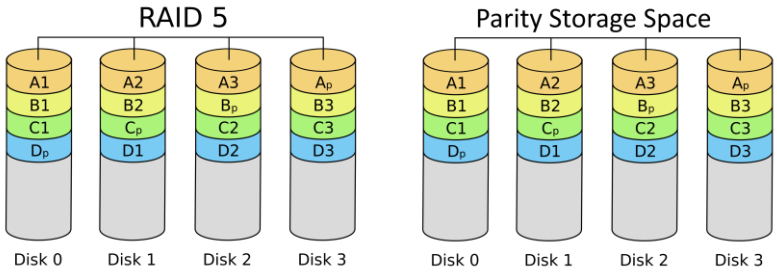

Abbildung 7: RAID5 im Vergleich zu Storage Spaces Parity, verändert nach (Burnett, 2006).

3.3.3.5 Dual Parity-Layout

Ein Storage Space mit Dual Parity-Layout ist funktionell nahezu identisch zu einem Storage Space mit einem Parity Layout. Der Unterschied liegt hier dabei, dass ein weiteres Paritätsbit pro Stripe gespeichert wird. Ein solcher Storage Space ist damit, ähnlich einem RAID6, resilient gegenüber dem Ausfall von bis zu zwei Disks. (Microsoft Corporation, Deploy Storage Spaces on a stand-alone server, 2021) Für dieses Layout werden mindestens 7 Disks benötigt. (Microsoft Corporation, Storage Spaces in Windows 10, kein Datum)

3.3.4 Volumen

Da Storage Spaces vom Betriebssystem angesehen werden, wie physische Disks, kann auf einem Storage Space eine beliebige Anzahl an Volumen erstellt werden. Innerhalb der Benutzeroberflächen von Storage Spaces selbst, werden nur NTFS oder ReFS als Dateisystem angeboten. (Microsoft Corporation, Deploy Storage Spaces on a stand-alone server, 2021) Da Volumen auf Storage Spaces vom Betriebssystem gleichbehandelt werden, wie Volumen auf herkömmlichen Disks, ist es allerdings nicht unmöglich diese mit anderen Dateisystemen zu formatieren.

3.4 Bedienung

3.4.1 Benutzeroberfläche

Storage Spaces bietet sowohl die Bedienung per Konsole, (Microsoft Corporation, Deploy Storage Spaces on a stand-alone server, 2021) als auch eine grafische Benutzeroberfläche für alle essenziellen Features an. Bei letzterer besteht allerdings großer Unterschied zwischen Desktop Betriebssystemen wie Windows 10 (Otachi, 2020) und Server Betriebssystemen wie Windows Server 2019. (Posey, 2018)
Auf die Benutzeroberflächen und deren Benutzerfreundlichkeit im Sinne der Human-Computer-Interaction wird an dieser Stelle nicht weiter eingegangen, da es nicht der Fokus dieser Arbeit ist.

3.4.2 Sandboxing

Storage Spaces bietet nativ keine Möglichkeit an, Konfigurationen im Voraus zu testen, es ist lediglich möglich mittels virtueller Disks, beispielsweise innerhalb einer virtuellen Maschine, Testszenarien aufzubauen.

3.4.3 Erweiterung des Speichers

Bezüglich der Erweiterung des Speichers ist Storage Spaces recht flexibel. Einem Storage Pool können prinzipiell beliebig Disks hinzugefügt werden. Wenn Thin Provisioning verwendet wird, können auch ungenutzte Disks problemlos wieder entnommen werden. (Huc, 2021)

Zu beachten ist hierbei jedoch, dass die Speichereffizienz eines Storage Pools nach dessen Erstellung nicht modifiziert werden kann. Beispielsweise hat ein Pool, welcher mit Parity Layout und 3 Disks erstellt wurde, auch mit 6 Disks eine Speichereffizienz von circa 66 Prozent. Dies kommt daher, dass die Anzahl der Daten-Slabs pro Paritäts-Slabs nicht verändert werden kann. Für Storage Spaces mit Mirror-Layout gilt das gleiche, da die Hälfte des Speicherplatzes für die Kopien der Labs benötigt wird. (Dell Technologies Inc., Column count, kein Datum)
Auch die Resilienz eines Storage Pools kann nicht verändert werden. Beispielsweise ist ein Storage Pool mit Mirror Layout auch mit 10 Disks nur gegen den Ausfall einer Disk resilient. (Rathnam, WINDOWS 10 STORAGE SPACES — A COMPREHENSIVE GUIDE, 2018)

4 Anwendungsmöglichkeiten

Unabhängig des eigentlichen Verwendungsortes besteht, sowohl bei OpenZFS als auch bei Storage Spaces mit ReFS-Dateisystem, der Vorteil des Schutzes vor Bit-Rot. So kann der Verlust von, eventuell kritischen, Informationen vermieden werden. Dies ist sicherlich in jedem professionellen Anwendungskontext wünschenswert und wird daher in den folgenden Abschnitten nicht wiederholt gesondert erwähnt.

4.1 Zentraler Netzwerkspeicher für alle Mitarbeiter

Durch Features, wie Deduplication und das dynamische Skalieren von Dateisystemen bzw. Storage Spaces innerhalb eines Pools, eignet sich virtualisierter Speicher besonders gut als zentraler Speicherort für digitale Dokumente.

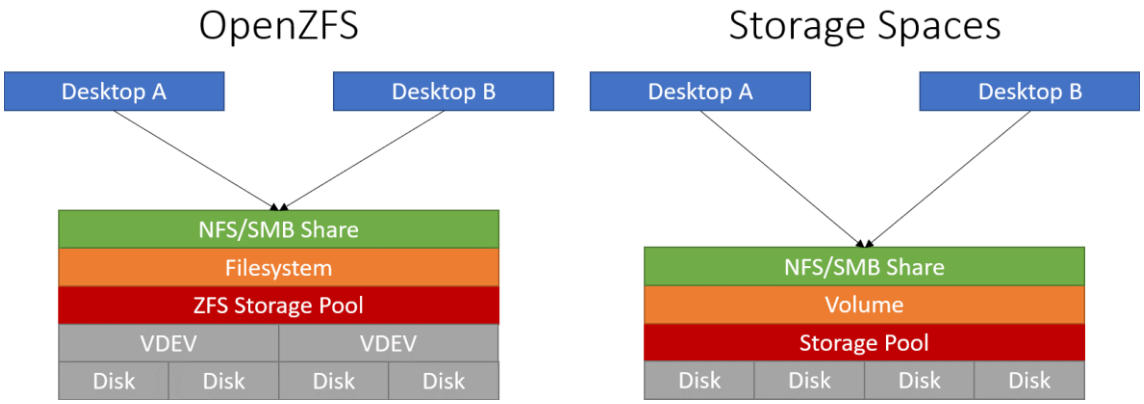

Abbildung 8: Schematische Darstellung der Verwendung von virtualisiertem Speicher als monolithischer zentraler Netzwerkspeicher, eigene Darstellung

So können Dateien, welche oft identische Passagen enthalten, dedupliziert werden, um Speicherplatz zu sparen. Ein gutes Beispiel hierfür sind Rechnungen, welche oftmals ähnliche Kopfzeilen mit Firmenlogos und Kontaktdaten beinhalten. Auch Verträge sollten hierbei gute Ergebnisse liefern, da diese oft lange Passagen an Informationen beinhalten, welche für viele Kunden identisch sind.

Zusätzlich dazu, kann auf ein plötzliches Wachstum an zu lagernden Daten, beispielsweise durch eine hohe Neukundenzahl, besonders schnell reagiert werden. Es muss nicht erst ein neues Speichersystem aufgebaut werden und alle Daten übertragen werden. Stattdessen können dem existierenden System zusätzliche Disks hinzugefügt werden.

4.2 Individuelle Workspaces für Mitarbeiter

Die Möglichkeit einen zentralen OpenZFS Pool für das Lagern aller Workspace-Daten von Mitarbeitern eines Büros zu verwenden, bietet sich an. Jedem Mitarbeiter könnte so ein eigenes Dateisystemen zugewiesen werden. Dieses würde aus der Sicht des Mitarbeiters eine fixe Größe haben, in welcher er frei arbeiten kann. Intern könnten die Dateisysteme allerdings dynamisch wachsen, und untereinander dedupliziert werden. (Toponce, 2012) So würden beispielsweise die Installationsdateien essenzieller Programme lediglich ein einziges Mal gespeichert.

Auch mit Storage Spaces ließe sich ein ähnliches System realisieren. Allerdings müssten hierbei für das Nutzen von Deduplication alle Mitarbeiter ein Volumen teilen. Auf diesem könnten dann Ordner als Workspaces angelegt werden. Da so nicht mehrere Storage Spaces existieren, muss die Verwaltung von maximalen Speichergrößen der einzelnen Ordner extern von Storage Spaces geregelt werden.

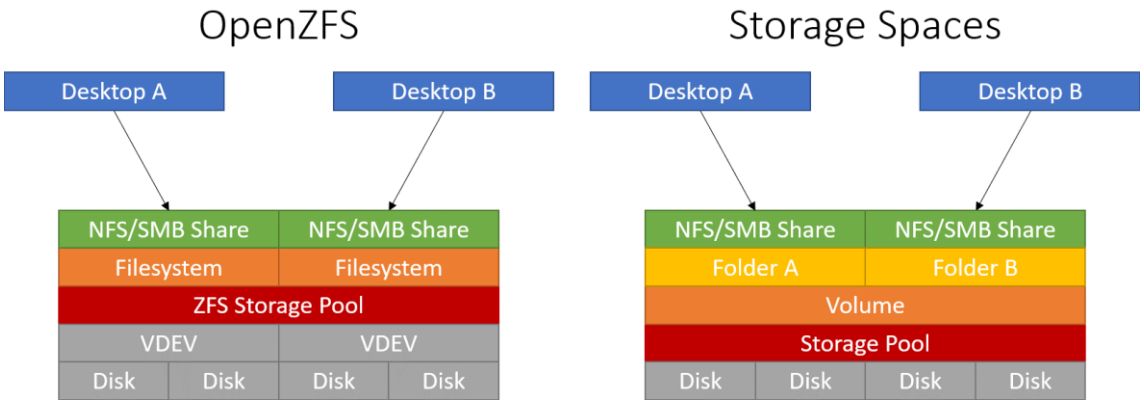

Abbildung 9: Schematische Darstellung der Verwendung von virtualisiertem Speicher als zentraler Netzwerkspeicher mit individuellen Workspaces, eigene Darstellung

Der unterliegende Pool könnte in beiden Fällen problemlos auf ein Mitarbeiterwachstum reagieren. Speicherplatz der zusätzlichen Dateisysteme wird erst dann benötigt, wenn diese tatsächlich mit Daten beschrieben werden. In Kombination mit Deduplication könnte auch der initial benötigte Speicherplatz eines neuen Mitarbeiters gesenkt werden. Zusätzliche Installationen von bereits verwendeter Software würden im Idealfall quasi keinen eigenen Speicherplatz benötigen. So kann die Anzahl an möglichen Mitarbeiter-Workspaces pro Speichereinheit drastisch erhöht werden.

4.3 Virtualisierung und Remote-Zugriff für Mitarbeiter

Auch im Tandem mit der Nutzung von virtuellen Maschinen kann Speichervirtualisierung mittels OpenZFS und Storage Spaces enorme Vorteile gegenüber traditionellen Dateisystemen und RAID-Lösungen bieten.

Die Bereitstellung von virtuellen Maschinen mit Windows 10 Installation benötigt gewöhnlicherweise je Nutzer mindestens 20 Gigabyte Speicherplatz. (Microsoft Corporation, Windows 10 system requirements, kein Datum) Unter Verwendung von Deduplication könnte dies jedoch drastisch reduziert werden, da lediglich die Unterschiede zwischen den einzelnen Maschinen gespeichert werden müssen.
Von Microsoft wird hierfür mit einer Ersparnis von 80-95% des Speicherplatzes geworben, (Microsoft Corporation, Data Deduplication Overview, 2017) das OpenZFS Project macht diesbezüglich keine genauen Angaben, allerdings ist, rein rechnerisch, mit ähnlichen Ergebnissen zu rechen.

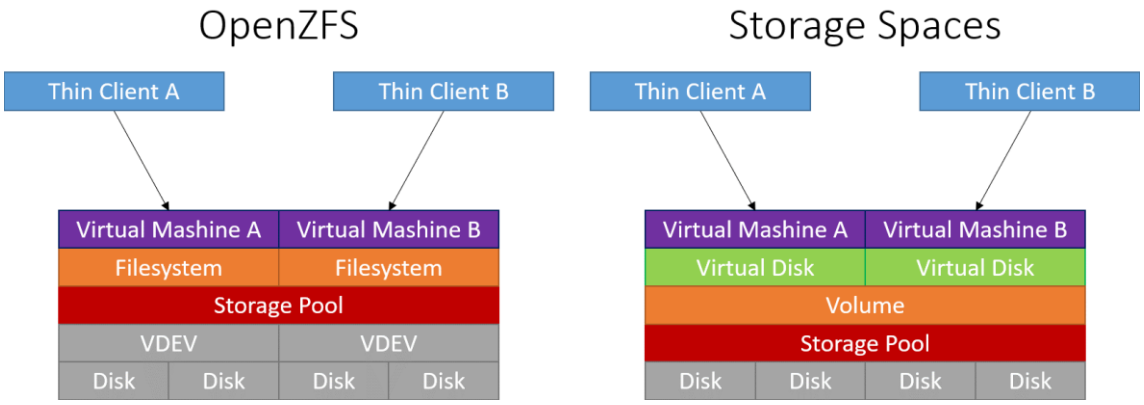

Abbildung 10: Schematische Darstellung der Verwendung von virtualisiertem Speicher in Kombination mit einem Hypervisor, eigene Darstellung

Auch installierte Software, welche von mehreren Mitarbeitern genutzt wird, würde so sehr viel weniger Speicherplatz pro Mitarbeiter beanspruchen. Hierfür wird von Microsoft mit einer Speicherplatz Ersparnis von 70-80% geworben. (Microsoft Corporation, Data Deduplication Overview, 2017)

Auf die Umsetzbarkeit von Virtualisierung je nach der Natur des physischen Aufbaus eines Kleinunternehmens soll an dieser Stelle nicht weiter eingegangen werden.

4.4 Lokaler Speicher für Mitarbeiter

Für Speicherplatz innerhalb eines Desktopcomputers einzelner Mitarbeiter, bietet die Verwendung von virtualisiertem Speicher nur geringfügige Vorteile, im Vergleich mit herkömmlichen Systemen.

Im Falle von OpenZFS wären primär das Copy-on-Write- bzw. Redirect-on-Write Prinzip und die damit verbundenen Snapshots interessant. Diese könnten beispielsweise im Falle von Bedienungsfehlern seitens des Benutzers helfen, versehentlich veränderte Daten wiederherzustellen. Auch Komprimierung und Deduplication sind theoretisch nicht gänzlich uninteressant. Allerdings würde die Ersparnis an benötigtem Speicherplatz im Falle von individuellen Desktops weniger drastisch ausfallen, da ein einzelner Desktop in der Regel nicht mehrere identische Versionen derselben Software oder desselben Betriebssystems beinhaltet.

Storage Spaces bietet hierbei ebenfalls verhältnismäßig wenige Vorteile, da diese sicherlich mit einem Desktop-Betriebssystem, wie Windows 10, ausgestattet wären. Somit wäre Deduplication unabhängig von deren Effektivität in diesem Falle nicht nutzbar.

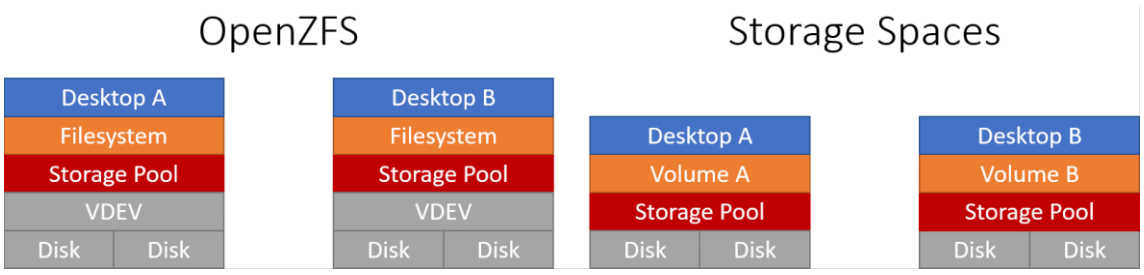

Abbildung 11: Schematische Darstellung der Verwendung von virtualisiertem Speicher als lokalen Speicher auf mehreren Desktops, eigene Darstellung

Unabhängig des verwendeten Systems wäre Resilienz gegenüber Ausfällen von Disks umsetzbar. Allerdings ist der Kostenfaktor, jeden Mitarbeiter-Desktop mit mehreren Disks auszustatten, in den meisten Fällen deutlich höher, als dies auf einem zentralen Server zu tun. Insbesondere daher, da Disks mit weniger Speicherplatz meist einen sehr viel höheren Preis pro Gigabyte aufweisen als solche in den aktuell typischen Speichergrößen. Der geringste Preis pro TB für HDDs liegt auf Amazon.com aktuell bei circa 18 US-Dollar. Hierbei handelt es sich um eine 4TB Disk aus der „Blue“-Produktlinie des Herstellers Western Digital. Die günstigste 1TB HDD, ebenfalls ein Western Digital „Blue“-Modell, kostet dagegen circa 40 US-Dollar. (Legitimate Data Company LLC., kein Datum)

5 Vergleich

5.1 Initiale Kostenfaktoren

5.1.1 Host-System

Weder OpenZFS noch Storage Spaces benötigen selbst spezialisierte Hardware, da sie ausschließlich Softwarelösungen sind. Sie können also prinzipiell auf jeder Hardware installiert werden, welche ausreichend ist, um eines der jeweils unterstützten Host-Betriebssysteme ausführen zu können. An dieser Stelle ist erwähnenswert, dass dies auch für viele Software-RAID-Lösungen der Fall ist.

OpenZFS hat bezüglich des Host-Systems sehr wenige Einschränkungen. Es sind Versionen für viele bekannte Linux Distributionen verfügbar, welche oftmals selbst auf Einplatinencomputern, wie dem Raspberry Pi, installiert werden können. Da Versionen für eine Vielzahl von Architekturen verfügbar sind, besteht kein Zwang zu typischer x86 Hardware. (OpenZFS Project, Supported Architectures, kein Datum)

Da die Systemanforderungen der unterstützten Betriebssysteme unterschiedlich sind, wird im Folgenden Ubuntu Server als Beispiel verwendet. Diese Lauten:

- Ein Prozessor mit 1 Gigahertz Taktfrequenz und 64bit Support
- 1 Gigabyte Arbeitsspeicher
- 2,5 Gigabyte Festplattenspeicher

(Canonical Ltd., kein Datum)

Das OpenZFS Project selbst empfiehlt ein Minimum von 2 Gigabyte Arbeitsspeicher beziehungsweise 8 Gigabyte, wenn Deduplication verwendet werden soll. (OpenZFS Project, Hardware Requirements, kein Datum) Für Deduplication werden an einigen Stellen zusätzlich Angaben für Arbeitsspeicher pro Terabyte Speicherplatz gegeben, beispielsweise 1-3 Gigabyte. (iXsystems Inc., 2021)
ECC-fähiger Arbeitsspeicher wird ebenfalls empfohlen, um die Beschädigung von Daten durch Fehler im Arbeitsspeicher zu vermeiden. Alternativ ist es konfigurierbar, dass auch Daten, welche aus dem Arbeitsspeicher auf die Disks geschrieben werden sollen, zunächst mit Prüfsummen kontrolliert werden sollen. (OpenZFS Project, Do I have to use ECC memory for ZFS?, kein Datum)

Storage Spaces hat im Vergleich erhebliche Einschränkungen bezüglich der unterliegenden Hardware, da es lediglich im Windows-Ökosystem verfügbar ist.
Für eine Installation unter Windows 10 lauten diese:

- Ein Prozessor mit 1 Gigahertz Taktfrequenz
- 2 Gigabyte Arbeitsspeicher
- eine DirectX fähige Grafikeinheit, mit mindestens SVGA-Auflösung
- 20 Gigabyte Festplattenspeicher

(Microsoft Corporation, Windows 10 system requirements, kein Datum)
Für eine Installation unter Windows Server 2019 lauten diese:

- Ein Prozessor mit 1,4 Gigahertz Taktfrequenz
- 512 Megabyte ECC Arbeitsspeicher für Versionen ohne GUI
- 2 Gigabyte ECC Arbeitsspeicher für Versionen mit GUI
- 32 Gigabyte Festplattenspeicher

(Microsoft Corporation, Hardware requirements for Windows Server, 2021)

Die Anforderungen beider Systeme sind in den meisten Punkten recht ähnlich, fallen allerdings beim Arbeitsspeicher sehr unterschiedlich aus. Insbesondere wenn Deduplication genutzt werden soll. Für ein Beispielsystem, welches 4TB an nutzbarem Speicher deduplizieren soll, würde OpenZFS circa 5-14GB Arbeitsspeicher benötigen. Storage Spaces benötigt, nach eigenen Angaben, lediglich 4,5-6GB Arbeitsspeicher. Auch die tatsächliche Auswirkung der Prozessorleistung sollte, rein logisch betrachtet, zwischen den Systemen unterschiedlich ausfallen, da es sich bei Storage Spaces um asynchrone und bei OpenZFS um eine asynchrone Deduplication handelt, allerdings lässt sich dieser nur schwer quantifizieren.

Im Falle von Storage Spaces ist zudem noch zu beachten, dass einige Features ausschließlich auf Server Betriebssystemen unterstützt werden. Dies bietet einige Einschränkungen für die verwendete Hardware des Host-Systems. Insbesondere daher, dass es hierbei Einschränkungen seitens der Hersteller der Hardware gibt. Für viele konsumentenorientierte Hardware existieren keine offiziellen Treiber für Windows Server Betriebssysteme. (Windows Server Catalog, kein Datum)

5.1.2 Lizenzkosten

OpenZFS ist eine Free-Open-Source-Software, oder kurz „FOSS“. Es ist also kein Kauf einer Lizenz für die Software selbst notwendig. Auch unter den offiziell unterstützten Betriebssystemen findet sich eine Vielzahl an Optionen, welche auch

für den gewerblichen Nutzen lizenzkostenfrei sind, beispielsweise Ubuntu Linux oder FreeBSD. (OpenZFS Project, Distributions, kein Datum)

Microsoft Storage Spaces dagegen ist eine proprietäre Lösung. Zwar werden für Storage Spaces selbst keinerlei Lizenzkosten erhoben, allerdings ist das System ausschließlich innerhalb von Microsofts Windows Betriebssystemen verfügbar. Diese wiederum bringen Lizenzkosten von 259€ für eine Windows 10 Pro Desktop Lizenz (Microsoft Corporation, Windows 10 Pro (Download), kein Datum) bzw. mindestens 501 US-Dollar für eine Windows Server 2019 Lizenz. (Microsoft Corporation, Preise und Lizenzierungsoptionen für Windows Server 2019, kein Datum) Hierbei handelt es sich um Windows Server 2019 Essentials, welches speziell für „Kleine Unternehmen mit bis zu 25 Benutzern und 50 Geräten“ (Microsoft Corporation, Preise und Lizenzierungsoptionen für Windows Server 2019, kein Datum) geeignet ist.

Hierbei ist zu beachten, dass der Kauf einer Windows Server Lizenz unabdingbar ist, wenn Storage Spaces mit Deduplication verwendet werden soll. Für OpenZFS gibt es diesbezüglich keine Einschränkungen, Deduplication ist auf allen unterstützten Betriebssystemen vorhanden.
In diesem Falle wird der Vorteil einer FOSS-Software sehr deutlich.

5.1.3 Speichereffizienz

Die Speichereffizienz beider Systeme ist prinzipiell abhängig von den gewählten Topologien bzw. Layouts. In beiden Fällen sind die theoretischen Effizienzwerte allerdings sehr nah bzw. identisch zu denen eines RAID-Layouts mit äquivalenter Resilienz und somit auch zueinander.
An dieser Stelle ist es jedoch erwähnenswert, dass durch die Anwendung von Deduplication die offenbare Speichereffizienz erhöht wird. Dies beeinflusst zwar nicht die eigentliche Effizienz des Skalierens des Dateisystems auf mehrere Disks, erhöht allerdings dennoch in vielen Fällen die vom Endnutzer wahrgenommene Kapazität des Speicherplatzes. Dadurch können die Kosten, pro wahrgenommener Speichereinheit, erheblich gesenkt werden.

OpenZFS hat hierbei den klaren Vorteil, dass für die Nutzung von Deduplication keine gesonderte Lizenz benötigt wird. Storage Spaces dagegen benötigt zwangsläufig eine Windows Server Lizenz. Diese ist sowohl mit höheren

Lizenzkosten als auch mit zusätzlichen Einschränkungen bezüglich des Host-Systems verbunden.

Ebenfalls zu beachten ist, dass die synchrone Deduplication von OpenZFS weniger Overhead-Speicherplatz benötigt, da die Daten nicht zunächst zwischengespeichert werden müssen. Dies ist im Sinne der Speichereffizienz ein weiterer Vorteil für OpenZFS.

5.1.4 Einrichtungsaufwand

Nach der Analyse der ersten beiden Kapitel dieser Arbeit lässt sich mit Sicherheit sagen, dass unabhängig des gewählten Systems ein nicht unbeträchtlicher Rechercheaufwand vor dem Deployment benötigt wird.

Storage Spaces bietet zwar eine GUI an, allerdings werden für viele Funktionalitäten nach wie vor Kommandozeilenbefehle benötigt. Auf Windows Server Versionen des Systems beispielsweise kann die Disk Nutzung eines Storage Pools nicht über die GUI optimiert werden. (Brink, 2021)

OpenZFS selbst bietet ausschließlich Bedienung über die Kommandozeile an, grafische Benutzeroberflächen existieren lediglich durch Drittanbieter.

Beide Systeme müssten also durch Personal bedient werden, welches im Umgang mit der Kommandozeile geübt ist. Der Einrichtungsaufwand im Sinne von Kosten für das Anstellen von Fachpersonal bzw. das Beauftragen von externen Firmen ist in diesem Sinne also als gleichwertig anzusehen.

5.2 Langzeit Kostenfaktoren

5.2.1 Wartungsaufwand

Aufgrund der Tatsache, dass es sich bei beiden Systemen um reine Software-Lösungen handelt, beläuft sich der Wartungsaufwand auf ein Minimum. Da das System unabhängig von der unterliegenden Hardware ist, müssen auch im Falle von Ausfällen keine spezifischen Hardware RAID-Karten ausgetauscht werden, welche zu späteren Zeitpunkten eventuell nicht mehr erhältlich sind. (Rathnam, HARDWARE RAID VS. SOFTWARE RAID: PROS AND CONS FOR EACH, 2020)

5.2.2 Erweiterbarkeit

Im Falle von OpenZFS gibt es aktuell viele Artikel zum Thema RAIDZ-Erweiterung. Diese ist zwar bereits als Teil einer zukünftigen Version angekündigt und der dazugehörige Quelltext schon in vorläufiger Form publiziert worden, allerdings ist dieser noch nicht Teil der regulär verfügbaren Versionen. (Salter, ZFS fans, rejoice—RAIDz expansion will be a thing very soon, 2021)
Da im Umfang dieser Arbeit allerdings nur der aktuelle Stand der Dinge betrachtet werden kann, wird im Folgenden lediglich von der Möglichkeit, einen zPool durch das Hinzufügen von zusätzlichen VDEV zu erweitern, ausgegangen.

In beiden Fällen kann das System mit verhältnismäßig geringem Aufwand erweitert werden. Wenn zusätzliche Disks angeschlossen werden sollen, werden keine zusätzliche RAID-Karten benötigt, sondern lediglich handelsübliche SATA bzw. SAS Speichercontroller.

Ein weiterer Vorteil besteht darin, dass Erweiterungen des Speichers live durchgeführt werden können. Insofern Hot-Plug-fähige Hardware verwendet wird, müssen die betroffenen Systeme lediglich für das Hinzufügen zusätzlicher Speichercontroller heruntergefahren werden. Bei herkömmlichen RAID-Systemen ist solche Erweiterbarkeit nicht allgemein gegeben, sondern abhängig von der genutzten Hardware bzw. Software.

Untereinander verglichen, bestehen hierbei nur geringe Unterschiede zwischen den Systemen. Beide Systeme unterstützen das Erweitern von Speicherplatz eines Pools, ohne diesen Offline nehmen zu müssen. In beiden Fällen kann jedoch weder die Resilienz noch die Speichereffizienz durch eine Erweiterung verändert werden. Sollte dies gewünscht sein, müssen die Systeme ebenfalls komplett neu aufgebaut werden und alle Daten zwischengelagert werden, so wie es auch bei herkömmlichem RAID der Fall ist. Hierbei muss sich also zwischen der Einfachheit der Umsetzung und der Speichereffizienz entschieden werden.

5.3 Tabellarische Zusammenfassung

Tabelle 1: Zusammenfassung der Kostenfaktoren, eigene Darstellung

	OpenZFS (Ubuntu Server)	**Storage Spaces (Windows Desktop)**	**Storage Spaces (Windows Server)**
Plattform	Konsumenten-Hardware	Konsumenten-Hardware	Limitiert durch verfügbare Treiber
CPU	Min. 1x 1GHz	Min. 1x 1GHz	Min. 1x 1.4GHz
Arbeitsspeicher (mit Deduplication)	Min. 1GB ECC (1-3GB pro TB)	Min. 2GB (1GB pro TB)	Min. 512MB ECC (1GB pro TB)
Dedizierte Hardware	Keine	Keine	Keine
Lizenzkosten + OS	Keine Keine (Ubuntu Server)	Keine 259 EUR (Windows 10 Pro)	Keine 501 USD (Windows Server 20XX)
Deduplication	Synchron auf Pool Ebene	Nein	Asynchron auf Storage Space Ebene
Bedienung	CLI & GUI (inoffiziell)	CLI & GUI	CLI & GUI
Live-Erweiterung	Ja	Ja	Ja

6 Fazit

Abschließend lässt sich sagen, dass Speichervirtualisierung als solches für kleinere Unternehmen eine Vielzahl von Vorteilen gegenüber herkömmlichen Systemen, wie RAID, bieten kann, insofern diese korrekt angewandt wird.

Die initialen Kosten für Hardware sind zwar in beiden Fällen geringer als für eine Hardware-RAID Lösung, allerdings fallen im Falle von Microsoft Storage Spaces erhebliche Lizenzkosten an. Im Falle von OpenZFS stellt der Einstieg in ein solches System eine geringe Hürde dar. Dies ist allerdings auch für Software-RAID der Fall.

Des Weiteren bieten die Systeme einen höheren Grad an Flexibilität. Dies gilt sowohl gegenüber traditionellem Hardware-RAID als auch gegenüber Software-RAID-Lösungen, welche sich in ihrer Funktionalität von Hardware-RAID kaum unterscheiden. Diese Flexibilität, in Verbindung mit geringen initialen Kosten, erlaubt es kleineren Unternehmen ein System aufzubauen, welches mit dem Unternehmen wachsen kann. So können Kosten in Form von regelmäßiger Neuanschaffung kompletter Speichersysteme vermieden werden.

Zusätzlich dazu bieten die Systeme eine Vielzahl von Features, welche die offenbare Speichereffizienz erhöhen können. Im Falle von Storage Spaces sind diese allerdings abhängig vom Betriebssystem. OpenZFS dagegen integriert diese in ein zentrales System. So können alle Aspekte des Speichersystems zentral verwaltet werden, was erhebliche Vorteile in Sachen Verwaltungsaufwand bietet. Zwar ist OpenZFS selbst dadurch recht komplex, das Speichersystem als Ganzes wird jedoch verhältnismäßig simpel gehalten. Soll das System ausschließlich als Speichersystem verwendet werden, so muss lediglich mit einer Software interagiert werden.

Microsoft bewirbt Storage Spaces zwar als ein Speichersystem für jedermann, und liefert es mit einer anwenderfreundlichen GUI aus. Allerdings ist für dessen korrekte Bedienung ein ähnlicher Wissensgrad nötig, wie für OpenZFS. Was sich aus dem Inhalt dieser Arbeit ergeben sollte. Beide Systeme können nur von entsprechend ausgebildetem Fachpersonal in einer Art und Weise bedient und konfiguriert werden, in welcher die Vorteile der Systeme voll ausgenutzt werden können. Demzufolge hat OpenZFS durch seine tiefere Integration aller relevanten Features, sowie die FOSS-Natur des Systems, einen klaren Vorteil für die Anwendung innerhalb von kleineren Unternehmen.

Quellenverzeichnis

Alsop, T. (12. August 2021). *Seagate's average capacity of hard disk drives (HDDs) worldwide from FY2015 to FY2021, by quarter*. Abgerufen am 22. September 2021 von Statista: https://www.statista.com/statistics/795748/worldwide-seagate-average-hard-disk-drive-capacity/

Bonwick, J. (2. November 2009). *ZFS Deduplication*. Abgerufen am 22. September 2021 von Jeff Bonwick's Blog: https://blogs.oracle.com/bonwick/zfs-deduplication-v2

Brink, S. (25. Januar 2021). *Optimize Drive Usage in Storage Pool for Storage Spaces in Windows 10*. Abgerufen am 6. Oktober 2021 von TenForums: https://www.tenforums.com/tutorials/83769-optimize-drive-usage-storage-pool-storage-spaces-windows-10-a.html

Bulao, J. (9. September 2021). *How Much Data Is Created Every Day in 2021?* Abgerufen am 22. September 2021 von TechJury: https://techjury.net/blog/how-much-data-is-created-every-day/#gref

Burnett, C. M. (31. Dezember 2006). *Standard RAID Levels*. Abgerufen am 21. September 2021 von Wikipedia: https://commons.wikimedia.org/wiki/File:RAID_0.svg

Canonical Ltd. (kein Datum). *Basic installation*. Abgerufen am 26. September 2021 von Ubuntu Server: https://ubuntu.com/server/docs/installation

Cunningham, A. (28. Oktober 2012). *Storage Spaces explained: a great feature, when it works*. Abgerufen am 19. September 2021 von ars Technica: https://arstechnica.com/information-technology/2012/10/storage-spaces-explained-a-great-feature-when-it-works/

DataCore Software. (kein Datum). *Storage Virtualization*. Abgerufen am 22. September 2021 von DataCore.com: https://www.datacore.com/storage-virtualization/

Dell Technologies Inc. (kein Datum). *Column count*. Von Dell Storage with Microsoft Storage Spaces Best Practices Guide: https://www.dell.com/support/manuals/en-ca/storage-md1400-dsms/dsms_bpg_pub-v2/column-count?guid=guid-68f002f4-bc8b-4992-a1cc-b99767fbc86a abgerufen

Dell Technologies Inc. (kein Datum). *Interleave size*. Abgerufen am 23. September 2021 von Dell Storage with Microsoft Storage Spaces Best Practices Guide: https://www.dell.com/support/manuals/en-ca/storage-md1420-

dsms/dsms_bpg_pub-v2/interleave-size?guid=guid-3b4b3770-242b-401d-b9f5-9502565cb994&lang=en-us

Dell Technologies Inc. (kein Datum). *Logical sector size*. Abgerufen am 21. September 2021 von Dell Storage with Microsoft Storage Spaces Best Practices Guide: https://www.dell.com/support/manuals/en-ca/storage-md1420-dsms/dsms_bpg_pub-v2/logical-sector-size?guid=guid-3aceacbf-59ff-4c24-a1bb-f61f56b10724&lang=en-us

Delphix Engineering. (11. Dezember 2012). *ZFS fundamentals: transaction groups*. Abgerufen am 21. September 2021 von Delphix.com: https://www.delphix.com/blog/delphix-engineering/zfs-fundamentals-transaction-groups

Desouza, M. (21. Januar 2021). *Zyggy*. Abgerufen am 21. September 2021 von Github: https://github.com/manoeldesouza/zyggy

Finn, A. (23. Februar 2015). *Storage Spaces Performance Tuning*. Abgerufen am 23. September 2021 von Petri: https://petri.com/storage-spaces-performance-tuning

Foskett, S. (10. Juli 2017). *ZFS Is the Best Filesystem (For Now…)*. Abgerufen am 22. September 2021 von Stephen Foskett, Rack Rat: https://blog.fosketts.net/2017/07/10/zfs-best-filesystem-now/

Gubin, A. V. (4. Juli 2019). *ZFS RAIDZ vs. traditional RAID*. Abgerufen am 10. September 2021 von Klennet Storage Software: https://www.klennet.com/notes/2019-07-04-raid5-vs-raidz.aspx

Hartsock, D. (14. August 2011). *Windows Home Server V1 Drive Extender*. Abgerufen am 20. September 2021 von Daves Computer Tips: https://davescomputertips.com/windows-home-server-v1-drive-extender/

Huc, M. (12. Januar 2021). *How to create parity volume with Storage Spaces in Settings on Windows 10*. Abgerufen am 23. September 2021 von Pureinfotech: https://pureinfotech.com/create-parity-volume-storage-spaces-settings-windows-10/

Isaac Computer Science. (kein Datum). *Transaction processing*. Abgerufen am 21. September 2021 von Isaac Computer Science: https://isaaccomputerscience.org/concepts/data_dbs_transaction?examBoard=all&stage=all

IT Free Training. (25. Mai 2014). *Windows Storage Spaces Demonstration*. Abgerufen am 23. September 2021 von IT Free Training: http://itfreetraining.com/server/storage-spaces-demonstration/

iX Systems. (12. November 2015). *The ZFS ZIL and SLOG Demystified*. Abgerufen am 21. September 2021 von iX Systems Blog: https://www.ixsystems.com/blog/zfs-zil-and-slog-demystified/

iXsystems Inc. (14. September 2021). *ZFS Deduplication*. Abgerufen am 24. September 2021 von TrueNAS: https://www.truenas.com/docs/references/zfsdeduplication/

Kennedy, P. (12. November 2017). *What is the ZFS ZIL SLOG and what makes a good one*. Abgerufen am 21. September 2021 von Serve the Home: https://www.servethehome.com/what-is-the-zfs-zil-slog-and-what-makes-a-good-one/

Kiljan, S. (23. September 2018). *A reference guide to ZFS on Arch Linux*. Abgerufen am 26. September 2021 von Sven and the Art of Computer Maintenance: https://kiljan.org/2018/09/23/a-reference-guide-to-zfs-on-arch-linux/

Klara Inc. (kein Datum). *OpenZFS: Understanding Transparent Compression*. Abgerufen am 10. September 2021 von Klara Systems: https://klarasystems.com/articles/openzfs1-understanding-transparent-compression/

Legitimate Data Company LLC. (kein Datum). Abgerufen am 30. September 2021 von diskprices.com: https://diskprices.com/?locale=us&condition=new&disk_types=internal_hdd

Mad About Brighton. (14. April 2016). *Increase ZFS pool by adding larger disks*. Abgerufen am 22. September 2021 von Madaboutbrighton.net: https://madaboutbrighton.net/articles/increase-zfs-pool-by-adding-larger-disks

Microsoft Corporation. (9. Mai 2017). *Data Deduplication Overview*. Abgerufen am 26. September 2021 von Microsoft Documentation: https://docs.microsoft.com/en-us/windows-server/storage/data-deduplication/overview

Microsoft Corporation (Hrsg.). (9. Mai 2017). *Install and enable Data Deduplication*. Abgerufen am 26. September 2021 von Microsoft Documentation: https://docs.microsoft.com/en-us/windows-server/storage/data-deduplication/install-enable

Microsoft Corporation. (10. Oktober 2019). *Thin Provisioning*. Abgerufen am 20. September 2021 von Microsoft Documentation: https://docs.microsoft.com/en-us/windows-hardware/drivers/storage/thin-provisioning

Microsoft Corporation. (19. September 2021). *Deploy Storage Spaces on a stand-alone server*. Von Microsoft Documentation: https://docs.microsoft.com/en-us/windows-server/storage/storage-spaces/deploy-standalone-storage-spaces abgerufen

Microsoft Corporation. (16. August 2021). *Hardware requirements for Windows Server*. Abgerufen am 22. September 2021 von Microsoft Documentation: https://docs.microsoft.com/en-us/windows-server/get-started/hardware-requirements

Microsoft Corporation. (31. August 2021). *Resilient File System (ReFS) overview*. Abgerufen am 19. September 2021 von Microsoft Documentation: https://docs.microsoft.com/en-us/windows-server/storage/refs/refs-overview

Microsoft Corporation. (kein Datum). *Preise und Lizenzierungsoptionen für Windows Server 2019*. Abgerufen am 24. September 2021 von Windows Server: https://www.microsoft.com/de-de/windows-server/pricing

Microsoft Corporation. (kein Datum). *Storage Spaces in Windows 10*. Abgerufen am 30. August 2021 von Microsoft Support: https://support.microsoft.com/en-gb/windows/storage-spaces-in-windows-b6c8b540-b8d8-fb8a-e7ab-4a75ba11f9f2

Microsoft Corporation. (kein Datum). *Windows 10 Pro (Download)*. Abgerufen am 24. September 2021 von Windows-Geräte für Unternehmen: https://www.microsoft.com/de-de/d/windows-10-pro/df77x4d43rkt?rtc=1&activetab=pivot:overviewtab

Microsoft Corporation. (kein Datum). *Windows 10 system requirements*. Abgerufen am 22. September 2021 von Microsoft Support: https://support.microsoft.com/en-us/windows/windows-10-system-requirements-6d4e9a79-66bf-7950-467c-795cf0386715

Microsoft Storage Spaces Is Hot Garbage For Parity Storage. (17. Dezember 2018). Abgerufen am 23. September 2021 von The Data Center Overlords: https://datacenteroverlords.com/2018/12/17/microsoft-storage-spaces-is-hot-garbage-for-parity-storage/

NudelSuppe. (8. November 2012). *Nested RAID levels*. Abgerufen am 21. September 2021 von Wikipedia: https://en.wikipedia.org/wiki/Nested_RAID_levels

OpenZFS Project. (25. Mai 2014). *History*. Abgerufen am 24. August 2021 von OpenZFS: https://openzfs.org/wiki/History

OpenZFS Project. (16. Januar 2016). *System Administration*. Abgerufen am 21. September 2021 von OpenZFS: https://openzfs.org/wiki/System_Administration

OpenZFS Project. (kein Datum). *Checksums and Their Use in ZFS*. Abgerufen am 22. September 2021 von OpenZFS: https://openzfs.github.io/openzfs-docs/Basic%20Concepts/Checksums.html

OpenZFS Project. (kein Datum). *Distributions*. Abgerufen am 24. September 2021 von OpenZFS: https://openzfs.org/wiki/Distributions

OpenZFS Project. (kein Datum). *Do I have to use ECC memory for ZFS?* Abgerufen am 21. September 2021 von OpenZFS: https://openzfs.github.io/openzfs-docs/Project%20and%20Community/FAQ.html#do-i-have-to-use-ecc-memory-for-zfs

OpenZFS Project. (kein Datum). *Hardware Requirements*. Abgerufen am 22. September 2021 von OpenZFS: https://openzfs.github.io/openzfs-docs/Project%20and%20Community/FAQ.html#do-i-have-to-use-ecc-memory-for-zfs

OpenZFS Project. (kein Datum). *Supported Architectures*. Abgerufen am 21. September 2021 von OpenZFS: https://openzfs.github.io/openzfs-docs/Project%20and%20Community/FAQ.html#supported-architectures

OpenZFS Project. (kein Datum). *System Administration*. Abgerufen am 21. September 2021 von OpenZFS: https://openzfs.org/w/index.php?title=System_Administration&mobileaction=toggle_view_mobile

Oracle Corporation. (20. April 2009). *Oracle Buys Sun*. Abgerufen am 21. September 2021 von Oracle.com: https://www.oracle.com/corporate/pressrelease/oracle-buys-sun-042009.html

Oracle Corporation. (kein Datum). *Adding Devices to a Storage Pool*. Abgerufen am 22. September 2021 von Managing ZFS File Systems in Oracle® Solaris 11.4: https://docs.oracle.com/cd/E37838_01/html/E61017/gayrd.html

Oracle Corporation. (kein Datum). *Creating a Basic ZFS File System*. Abgerufen am 4. September 2021 von Oracle Solaris ZFS Administration Guide: https://docs.oracle.com/cd/E26505_01/html/E37384/gaypm.html#scrolltoc

Oracle Corporation. (kein Datum). *Creating a ZFS Storage Pool With Cache Devices*. Abgerufen am 22. September 2021 von Managing ZFS File Systems in Oracle® Solaris 11.4: https://docs.oracle.com/cd/E37838_01/html/E61017/gfxtd.html

Oracle Corporation. (kein Datum). *Creating a ZFS Storage Pool With Log Devices*. Abgerufen am 22. September 2021 von Managing ZFS File Systems in

Oracle® Solaris 11.4: https://docs.oracle.com/cd/E37838_01/html/E61017/gffyt.html#scrolltoc

Oracle Corporation. (kein Datum). *Creating and Destroying ZFS Storage Pools*. Abgerufen am 6. September 2021 von Oracle Solaris ZFS Administration Guide: https://docs.oracle.com/cd/E26505_01/html/E37384/gaypw.html#scrolltoc

Oracle Corporation. (kein Datum). *Creating ZFS Storage Pools*. Abgerufen am 22. September 2021 von Managing ZFS File Systems in Oracle® Solaris 11.4: https://docs.oracle.com/cd/E37838_01/html/E61017/gaypw.html#scrolltoc

Oracle Corporation. (kein Datum). *Differences Between Synchronous and Asynchronous I/O*. Abgerufen am 22. September 2021 von Oracle Documentation: https://docs.oracle.com/cd/E19253-01/816-4854/character-11/index.html

Oracle Corporation. (kein Datum). *Doing a Dry Run of Storage Pool Creation*. Abgerufen am 22. September 2021 von Managing ZFS File Systems in Oracle® Solaris 11.4: https://docs.oracle.com/cd/E37838_01/html/E61017/gazhd.html

Oracle Corporation. (kein Datum). *Managing Devices in ZFS Storage Pools*. Abgerufen am 5. September 2021 von Oracle Solaris ZFS Administration Guide: https://docs.oracle.com/cd/E26505_01/html/E37384/gayrd.html#scrolltoc

Oracle Corporation. (kein Datum). *Overview of ZFS Clones*. Abgerufen am 22. September 2021 von Managing ZFS File Systems in Oracle® Solaris 11.4: https://docs.oracle.com/cd/E37838_01/html/E61017/gbcxz.html

Oracle Corporation. (kein Datum). *Replication Features of a ZFS Storage Pool*. Abgerufen am 5. September 2021 von Oracle Solaris ZFS Administration Guide: https://docs.oracle.com/cd/E26505_01/html/E37384/gcfof.html#scrolltoc

Oracle Corporation. (kein Datum). *Setting ZFS Quotas*. Abgerufen am 22. September 2021 von Managing ZFS File Systems in Oracle® Solaris 11.4: https://docs.oracle.com/cd/E37838_01/html/E61017/gazvb.html

Oracle Corporation. (kein Datum). *Storage Configuration Rules and Guidelines*. Abgerufen am 22. September 2021 von Oracle® ZFS Storage Appliance Administration Guide: https://docs.oracle.com/cd/E51475_01/html/E52872/goddw.html

Oracle Corporation. (kein Datum). *Traditional Volume Management*. Abgerufen am 21. September 2021 von Oracle Solaris ZFS Administration Guide: https://docs.oracle.com/cd/E18752_01/html/819-5461/gazcc.html

Oracle Corporation. (kein Datum). *What Is Oracle Solaris ZFS?* Abgerufen am 2. September 2021 von Oracle Solaris ZFS Administration Guide: https://docs.oracle.com/cd/E26505_01/html/E37384/zfsover-2.html#scrolltoc

Otachi, E. (29. November 2020). *How to Use Storage Spaces on Windows 10 for Data Backups*. Abgerufen am 21. September 2021 von Help Desk Geek: https://helpdeskgeek.com/windows-10/how-to-use-storage-spaces-on-windows-10-for-data-backups/

Patterson, D. A., Gibson, G., & Katz, R. H. (1. Juni 1988). *A Case for Redundant Arrays of Inexpensive Disks (RAID)*. Abgerufen am 26. August 2021 von Carnegie Mellon University: https://www.cs.cmu.edu/~garth/RAIDpaper/Patterson88.pdf

Posey, B. (31. Juli 2018). *How To Use Storage Spaces in Windows Server 2016, Part 1*. Abgerufen am 21. September 2021 von Redmond Magazine: https://redmondmag.com/articles/2018/07/31/storage-spaces-windows-server-2016-1.aspx

RAID. (kein Datum). Abgerufen am 21. September 2021 von Free On-Line Dictionary Of Computing: http://foldoc.org/RAID

RAID Incorporated. (11. Februar 2020). *A SHORT GUIDE ABOUT ZFS: THE LAST WORD IN FILE SYSTEMS*. Abgerufen am 21. September 2021 von RAID Incorporated: https://www.raidinc.com/2020/02/a-short-guide-about-zfs-the-last-word-in-file-systems/

Rathnam, L. (12. November 2018). *WINDOWS 10 STORAGE SPACES — A COMPREHENSIVE GUIDE*. Abgerufen am 28. September 2021 von TechGenix: https://techgenix.com/windows-10-storage-spaces/#:~:text=You%20cannot%20change%20the%20resiliency,a%20part%20of%20storage%20spaces.

Rathnam, L. (2. Juli 2020). *HARDWARE RAID VS. SOFTWARE RAID: PROS AND CONS FOR EACH*. Abgerufen am 6. Oktober 2021 von TechGenix: https://techgenix.com/hardware-raid-vs-software-raid/

Salter, J. (5. August 2020). *ZFS 101—Understanding ZFS storage and performance*. Abgerufen am 3. September 2021 von ars Technica: https://arstechnica.com/information-technology/2020/05/zfs-101-understanding-zfs-storage-and-performance/

Salter, J. (7. März 2021). *OpenZFS 2.1 is out—let's talk about its brand-new dRAID vdevs*. Abgerufen am 27. August 2021 von ars Technica: https://arstechnica.com/gadgets/2021/07/a-deep-dive-into-openzfs-2-1s-new-distributed-raid-topology/

Salter, J. (15. Juli 2021). *ZFS fans, rejoice—RAIDz expansion will be a thing very soon*. Abgerufen am 30. August 2021 von ars Technica.

SSD vs HDD Speed, Lifespan, and Reliability. (8. April 2019). Abgerufen am 22. September 2021 von Tekie.com: https://tekie.com/blog/hardware/ssd-vs-hdd-speed-lifespan-and-reliability/#:~:text=A%20typical%207200%20RPM%20HDD,s%20to%20550%20MB%2Fs.

The FreeBSD Project. (30. Juni 2019). *Maintenance Commands*. Abgerufen am 21. September 2021 von FreeBSD Manual Pages: https://www.freebsd.org/cgi/man.cgi?zfs(8)

Toponce, A. (18. Dezember 2012). *ZFS Administration, Part XI- Compression and Deduplication*. Abgerufen am 24. September 2021 von https://pthree.org/2012/12/18/zfs-administration-part-xi-compression-and-deduplication/

Tunstall, S. (11. Januar 2016). *Disk Scrub - Why and When?* Abgerufen am 21. September 2021 von The Wonders of ZFS Storage: https://blogs.oracle.com/wonders-of-zfs-storage/disk-scrub-why-and-when-v2#:~:text=ZFS%20will%20automatically%20protect%20your,checksum%2C%20and%20automatically%20fixes%20it.&text=If%20you%20don't%20normally,disk%20scrub%20about%20every%20month.

Windows Server Catalog. (kein Datum). Abgerufen am 6. Oktober 2021 von Microsoft: https://www.windowsservercatalog.com/

Abbildungsverzeichnis

Tabellenverzeichnis

Ehrenwörtliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt wurde.

Brandenburg an der Havel, 11.10.2021

Ort, Datum

Unterschrift

(Otto-Leander Werse)