Universität Hamburg
Department Informatik
Scientific Computing

NVMM - Der Weg in die Zukunft?

Seminararbeit

Effiziente Programmierung

Kira Duwe

Simon Oelgeschläger

Matr.Nr. 6930356

6oelgesc@informatik.uni-hamburg.de

28.02.2020

Inhaltsverzeichnis

1 Motivation **2**

2 Dateisysteme **3**
2.1 POSIX 3
2.2 NTFS 4
2.3 Aktuelle Probleme 4

3 Speichertechnologien **4**
3.1 Grundsätzliche Eigenschaften 4
3.2 Physikalischer Aufbau 5
3.3 NVMM 6
3.3.1 Intel 3D XPoint 6
3.3.1.1 Aufbau & Funktionsweise 7

4 Warum NVMM? **8**

5 Problemanalyse **8**
5.1 Herkömmliche Systeme 8
5.2 Systeme mit NVMM 8

6 Abschlussworte **9**

Literaturverzeichnis **10**

Einleitung

In dieser Seminararbeit werden neuartige, nichtflüchtige Hauptspeicher, englisch ‚non-volatile main memory' anhand von Einsatzzwecken und weiteren Faktoren diskutiert, sowie ein Ausblick auf Probleme und Lösungen, die diese Technologie mit sich bringt, geworfen. Im Folgenden wird für diese Speicher die Abkürzung NVMM verwendet.

1 Motivation

In modernen Computersystemen ist eine Vielzahl unterschiedlichster Komponenten enthalten. Sie alle wurden nach immer komplexeren Anforderungen entwickelt und arbeiten eng zusammen. Dabei lassen sich die grundlegenden Bausteine grob nach ihren Einsatzgebieten aufgliedern. So steigt die Gesamtleistung eines Systems im Normalfall an, wenn seine Rechenleistung steigt – wenn also beispielsweise ein Prozessor mit höherer Taktrate und zusätzlichen Kernen verwendet wird. Aktuell erhältliche Prozessoren im Privatbereich sowie auch in professionellen Umgebungen sind meist mit etwa 3-5 GHz getaktet. In den vergangenen Jahren wurde diese Taktrate stets kontinuierlich erhöht und stabilisiert, jedoch sind die Änderungen meist marginal und entsprechen nicht den sprunghaften Entwicklungen anderer Bauteile. Um hier mitzuhalten, werden Prozessoren unter anderem mit mehr Kernen ausgestattet; aber auch hier beschränkt der aktuelle Stand der wissenschaftlichen-technischen Forschung die profitable Serienfertigung. Da der Weltmarkt jedoch stets Neuerungen mit gestiegener Leistung erwartet, liegt der Fokus zuletzt vermehrt auf der Optimierung der Low-Level Software dieser Prozessoren. Diese heuristisch arbeitenden Algorithmen organisieren unter anderem das vorausschauende Anfordern von Daten aus dem Speicher zu dem Zeitpunkt, zu dem sie benötigt werden. Da jedoch Magnetfestplatten und selbst SSDs eine um mehrere Magnituden größere Zugriffszeit haben, als die Taktfrequenz des Prozessors die Daten verarbeiten könnte, wird in fast allen Systemen eine Speicherhierarchie verwendet. Die Festplatten bilden dabei das untere Ende dieser Hierarchie, sie sind vergleichsweise langsam und haben eine hohe Kapazität. In der Ebene dafür befindet sich der sogenannte Arbeits- oder Hauptspeicher. Dieser hat deutlich geringere Zugriffszeiten als alle Medien der Ebene darunter, jedoch auch eine geringere Kapazität. Über dem Hauptspeicher befindet sich die Ebene des Cache-Speichers. Hier kann meist noch zwischen L1-, L2- und L3-Cache unterschieden werden – auch diese haben jedoch vereinfacht gesehen eine analoge Rolle in der Hierarchie: Der L1-Cache steht an der Spitze der Hierarchie und arbeitet am schnellsten. Somit ist er ideal für den direkten Zugriff des Prozessors. Damit dieses System optimal arbeitet, ist es erforderlich, dass die Daten, die in der Gegenwart und der nahen Zukunft vom Prozessor benötigt werden, so weit oben in der Hierarchie auffindbar sind, wie möglich. Realisiert wird dies durch die bereits angesprochenen heuristischen Algorithmen, welche dafür sorgen, dass benötigte Informationen vor der Verarbeitung durch den Prozessor bereits aus der Festplatte in den Hauptspeicher geladen wurden, sowie die aktuell verarbeiteten Teilsegmente dieser Informationen auch im Cache verfügbar sind.

2 Dateisysteme

Die Erfindung von Rechnern geht auf den Wunsch zurück, Informationen automatisiert verarbeiten zu können. Dabei standen anfangs zunächst mathematische Operationen im Vordergrund. Dies änderte sich jedoch relativ schnell, als die Prozessoren immer leistungsfähiger wurden und auch erste graphische Ausgaben möglich wurden. Trotzdem gilt auch heute noch, dass ein Rechner Operationen auf einer Menge von Daten durchführt, unabhängig davon, ob der Nutzer nun ein Programm starten, einen Video-Livestream dekodieren und anzeigen oder einen Text verfassen will. Damit ein Computer all diese Aufgaben mühelos erledigen kann, ist es notwendig, eine konsistente Datenbasis zu haben. In der weltweit verbreiteten Von-Neumann-Architektur [1] sind Programme und Daten konzeptionell identisch, dies erfordert ein System, nach dem Informationen gespeichert und verwaltet werden. Ermöglicht wird das durch ein Dateisystem – es bildet die Schnittstelle zwischen Betriebssystem und den physischen Laufwerken, indem es die Dateien auf den Datenträgern organisiert und den ordnungsgemäßen Zugriff auf diese sicherstellt. Durch die Entwicklung hin zu verteilten Computersystemen und Supercomputern gerät die gängige Konzeption von Dateisystemen jedoch zunehmend unter Druck. Werden beispielsweise Informationen mehrfach parallel angefordert, muss sichergestellt werden, dass verschiedene Operationen auf den Daten immer die aktuellste Version dieser nutzen. Sollten die Konsistenzbedingungen nicht erfüllt sein, können unvorhersehbare Zustände eintreten. Um dieses Problem anzugehen werden meist Locks verwendet, die Datei wird also für andere Zugriffe gesperrt – lediglich ausschließlich lesende Zugriffe können parallel existieren. Zusätzlich dazu werden Metadaten zu jeder Datei generiert und in einem Journal, siehe [2], aufgeführt, etwa welcher Prozess wann zuletzt darauf zugegriffen hat. Dies führt zu einem enorm erhöhten Aufwand im Dateisystem und einem beachtlichen Flaschenhals bei Systemen mit einer Vielzahl von Prozessoren. Als Nebeneffekt wird dazu auch jeder lesende Zugriff zu einem Schreibzugriff, da in den Metadaten der Datei der Zeitstempel des letzten Zugriffs aktualisiert werden muss. Wie bereits angesprochen takten moderne CPUs um viele Größenordnungen höher als die Festplatten, welche die Basis für das Dateisystem stellen. Dies erhöht nur den Druck, für besonders diese verteilten Systeme eine Lösung zu finden, welche die Speichermedien entlastet und eine bessere Auslastung der Prozessoren ermöglicht.

2.1 POSIX

Als Standardschnittstelle entwickelt, sind POSIX-kompatible Dateisysteme bereits seit etwa 30 Jahren im Einsatz [3]. Ziel war damals die Portabilität von Anwendungs-Sourcecode, also die Möglichkeit ein Programm auf verschiedenen Systemen laufen zu lassen, wenn beide ein mit POSIX kompatibles Dateisystem verwenden. Da POSIX jedoch aufgrund einiger zuvor erwähnter Probleme nicht sonderlich gut für verteilte Systeme geeignet ist, divergieren viele Neuerungen vom Standard. Es bilden sich auf Einzelfälle spezialisierte Weiterentwicklungen, welche untereinander meist nicht kompatibel sind. Zugleich werden immer häufiger High-Level Frameworks verwendet, die eigentlich Nutzung von POSIX Libraries ist oft nicht ersichtlich. Aufgrund des Alters des Standards werden heute benötigte Operationen teils über nicht dafür ausgelegte Operationen realisiert, sodass eine ‚(R)evolution' von POSIX zwar notwendig erscheint, jedoch durch die Vielzahl an aktuell genutzten Implementierungen schwierig sein dürfte.

2.2 NTFS

Das proprietäre Dateisystem von Microsoft (zur Info [4]) ist bereits seit einer ähnlichen Zeit wie POSIX-Implementierungen im Betrieb und hat auch mit ähnlichen Problemen zu kämpfen. NTFS ist im Gegensatz zu POSIX nicht Inode-basiert, die Struktur des Dateisystems wird also nicht dadurch gekennzeichnet, dass theoretisch alles eine Datei ist und seine eigenen Metadaten besitzt. NTFS verwendet stattdessen eine Master File Table, also eine Datei, in der alle Metadaten des Systems gesammelt gelagert werden.

2.3 Aktuelle Probleme

In größeren Systemen mit einer Vielzahl an Festplatten und Prozessoren treten deutlich andere Engpässe und Probleme auf als bei Endanwender-Computern unter dem Schreibtisch. So ist es vergleichsweise simpel, die Metadaten von Dateien auf einer Festplatte zu organisieren und Zugriffe darauf zu verwalten. Wenn jedoch mit mehreren Prozessoren gleichzeitig auf multiple Festplatten mit redundanten Datensätzen zugegriffen wird, ist ein fehlerfreies und exakt definiertes Verhalten notwendig. So muss beispielsweise eine redundante Kopie einer Datei auf Festplatte 1 aktualisiert werden, wenn Prozess 0 das Original auf Festplatte 0 bearbeitet. Dies muss jedoch zusätzlich sofort erfolgen, bevor Prozess 1 die nicht mehr aktuelle Kopie auf Festplatte 1 lesen will. Kommt es zwischen diesen beiden Lese- oder Schreiboperationen zu einem Stromausfall, muss das Dateisystem auch nach einem Neustart die Daten konsistent halten, sodass besonders hier ein striktes und detailliertes Journaling unabdingbar ist, welches natürlich auch die Festplatten fordert. Es müssen permanent immense Datenmengen durch das System fließen, das Bus-System kommt dabei oft an seine Grenzen. Zusätzlich erhöht dies die Auslastung der ohnehin schon langsamen und oft fehleranfälligen Festplatten, mit der Folge, dass deren Lebenszeit weiter reduziert wird.

3 Speichertechnologien

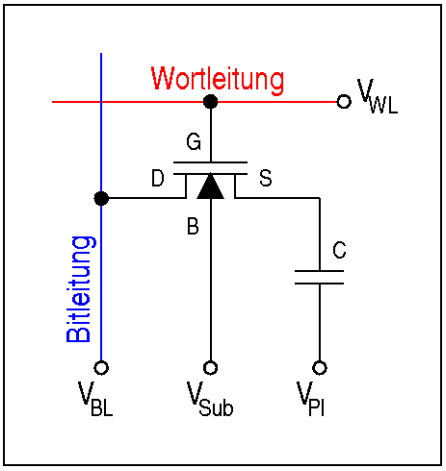
3.1 Grundsätzliche Eigenschaften

Ein Speichermedium ist zuerst ein Gegenstand, welcher Informationen physisch speichern kann. Dabei bilden etwa Magnetbänder und herkömmliche Festplatten den Anfang der technischen Entwicklungen. Als besonders wichtig lässt sich die Erfindung von Halbleiterspeichern nennen. Darauf basierende Technologien speichern die Daten in integrierten Schaltkreisen, meist zu Matrizen angeordnet.
Die verschiedenen Technologien lassen sich anhand von einigen Eigenschaften gut gruppieren. So können Magnetbänder nur einen sequenziellen Zugriff auf die Daten gewähren – ist eine Information an einer anderen Stelle des Bandes gespeichert, muss erst dorthin gespult werden. Dies ist offensichtlich nicht für Anwendungsfälle geeignet, in denen Daten in Bruchteilen von Sekunden verfügbar sein müssen. Das Gegenstück, der wahlfreie Zugriff, lässt hingegen einen Zugriff auf beliebige Stellen im Speicher mit konstanter Zeit zu. Zu dieser Kategorie gehören neben HDD Festplatten (zumindest annähernd konstante Zeit) auch SSDs sowie DRAM und SRAM Module, welche als Hauptspeicher, beziehungsweise Cache verwendet werden. Da in dieser Arbeit der Einsatz von ‚Non volatile memory' kurz NVM als Hauptspeicher untersucht wird, ist der Vergleich zu

ähnlichen Technologien interessant. So bieten SSD, DRAM sowie SRAM zwar wahlfreien Zugriff, jedoch muss beachtet werden, dass es sich nur bei der SSD um einen nichtflüchtigen Speicher handelt. Bei DRAM und SRAM muss stets eine Stromversorgung der Module gewährleistet sein, um die gespeicherten Informationen zu erhalten. Dies macht beide Speicherarten ungeeignet für den Einsatz als dauerhaften Massenspeicher. Die nachfolgend in Kapitel 3.3 eingeführte Technologie NVMM grenzt sich von ihrem direkten Konkurrenten, dem DRAM, dadurch ab, dass sie ihre Daten nichtflüchtig speichern kann, also Informationen auch nach einem ordnungsgemäßen Herunterfahren des Rechners und sogar nach einem Stromausfall noch unverändert vorzufinden sind. Dies ermöglicht vollkommen neue Anwendungsszenarien, birgt jedoch auch bisher unbekannte Gefahren.

3.2 Physikalischer Aufbau

Abbildung 1: Eine DRAM Speicherzelle mit Kondensator und Transistor

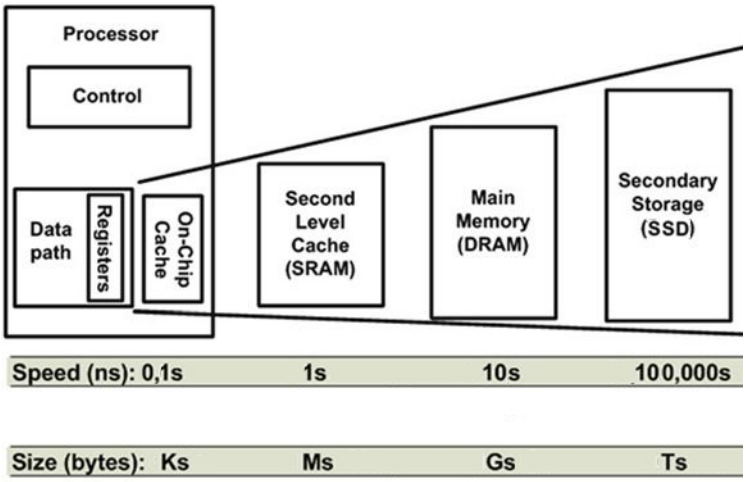


Quelle: https://de.wikipedia.org/wiki/Dynamic_Random_Access_Memory

Halbleiterspeicher basieren auf dem physikalischen Prinzip des Halbleiters. So lassen sich Transistoren, Dioden und Kondensatoren auf kleinstem Raum realisieren. Ein klassischer DRAM Speicher besteht dabei aus einigen Milliarden in einer Matrix angeordneten Zellen, welche jeweils aus einem Transistor und einem Kondensator aufgebaut sind, siehe Abb: 1. Durch die Anordnung in einer Matrix lässt sich mit Angabe von Zeile und Spalte theoretisch jede Zelle einzeln ansprechen. Da im praktischen Einsatz fast immer mehr als nur 1 Bit ausgelesen werden muss, wird bei einem DRAM Modul bauartbedingt immer eine gesamte Zeile ausgelesen und zurückgeschrieben. Der zusätzliche Aufwand ist dabei deutlich geringer als der Vorteil, mehrere Bits mit einer Operation auslesen oder beschreiben zu können. Durch den Einsatz eines Kondensators in jeder Zelle kommt es durch Kriechströme unweigerlich zum Verlust der Ladung. Dies passiert im Normalfall nach weniger als einer Sekunde. Somit muss eine Zelle regelmäßig aktualisiert werden – der Grund für den Namen ‚Dynamic-RAM'. Die Zugriffszeit eines DRAM-Moduls liegt bei einigen Nanosekunden, dies ist für den direkten Zugriff durch einen Prozessor zwar noch zu langsam, der Unterschied liegt jedoch nur bei 1-2 Größenordnungen, gegenüber den 4-5 Größenordnungen, die

eine SSD langsamer als DRAM ist, siehe Abb: 2. Somit erkauft man sich die deutlich geringere Zugriffszeit mit der Flüchtigkeit der Daten. Da diese Technologie sich seit ihrer Erfindung nicht mehr grundlegend geändert hat und weltweit standardmäßig eingesetzt wird, haben sich andere Bereiche eines Computersystems angepasst, beispielsweise bei der Entwicklung des teuren und ebenfalls flüchtigen Cache-Speichers um Performance-Engpässe zu beheben.

Abbildung 2: Klassische Hierarchie verschiedener Speicher



Quelle: https://link.springer.com/book/10.1007/978-1-4614-0818-5, bearbeitet.

3.3 NVMM

NVMM steht für ‚Non-volatile main memory', wobei anhand des Namens bereits erkennbar ist, dass es sich um eine nichtflüchtige Speichertechnologie handelt, welche als Hauptspeicher eingesetzt wird. Dies stellt einen Durchbruch in der Entwicklung von Speichern dar, da sich bisher Nichtflüchtigkeit und als Hauptspeicher einsatzbare Technologien insbesondere durch stark unterschiedliche Zugriffszeiten in der Praxis ausgeschlossen haben. Lediglich für wichtige Systembestandteile wie Informationen für den Bootprozess wurden bisher teure und kleine Module angefertigt, bei denen ein DRAM-Chip mit einer Batterie versehen war. NVMM vereint die Möglichkeiten eines schnellen und vergleichsweise günstigen Speichers mit genügend großer Kapazität und den Eigenschaften der persistenten Datenhaltung durch Nichtflüchtigkeit.

3.3.1 Intel 3D XPoint

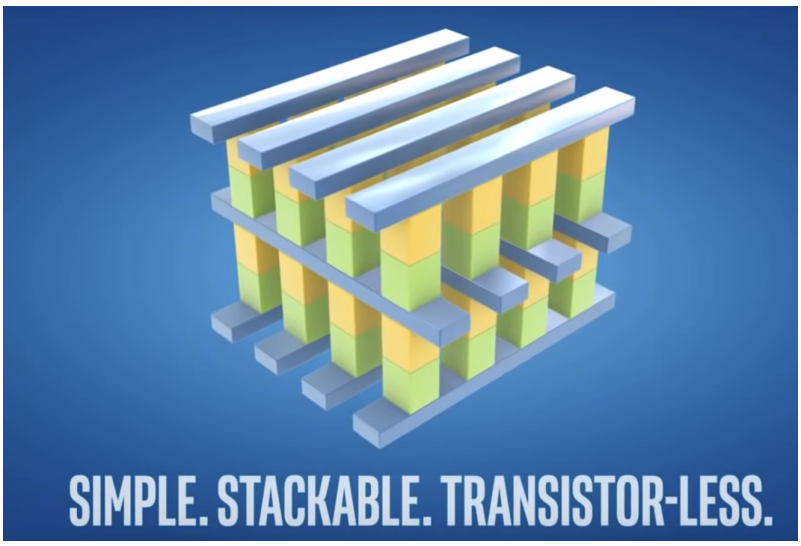
Als erste auf dem freien Markt erhältliche Implementierung der NVM-Technologie, verkauft Intel seine Produkte unter dem Namen ‚Intel Optane', erhältlich unter anderem hier: [5]. Diese aktuell noch als Evolution von SSDs verkauften Speichermodule basieren auf Intels 3D XPoint Technologie. Diese verwendet mutmaßlich ReRam (Resistive random-access memory, siehe [6]), oder den ähnlichen PCM (Phase change memory, siehe [7]). Die Informationen werden dort mittels Änderungen im Widerstand des Materials speichert. S. Ghosh schreibt, dass die Optane Produkte zwar deutlich schneller und leistungsfähiger als herkömmliche SSDs sind [8], aber auch hoch-

preisiger und damit für den normalen Endanwender zu diesem Zeitpunkt noch nicht realistisch einzusetzen. Allerdings spielt NVM in SSDs noch nicht seine volle Stärke aus, da SSDs bereits nichtflüchtig sind und somit einer der großen Vorteile in diesem Einsatzgebiet nicht zum Tragen kommt. In Kombination mit spezieller ab Mitte 2019 erhältlicher Hardware können Intel Optane Module auch als Hauptspeicher genutzt werden. Somit handelt es sich in diesem Fall um NVMM. Die Zugriffszeiten sinken in diesem Fall von wenigen Mikrosekunden als Massenspeicher auf einige Nanosekunden beim Einsatz als Hauptspeicher. Diese Werte liegen im Bereich von DRAM, sodass sich hier zwar keine Performancegewinne ergeben, die Technologie allerdings erst in ihrer ersten / zweiter Generation ist und somit noch einiges an Potential birgt. Weiterhin dürfte der Preis pro GB mittelfristig deutlich unter dem von DRAM liegen, sodass bei gleichbleibender oder sogar besserer Performance hardwareseitig NVMM die Wahl der Zukunft sein dürfte. Durch die identischen Technologien ist davon auszugehen, dass sich Optane SSDs und Optane Arbeitsspeicher immer mehr ähneln und schließlich miteinander verschmelzen.

3.3.1.1 Aufbau & Funktionsweise

Die Intel Optane Speicher sind ähnliche wie DRAM-Module aus gitterförmigen angeordneten einzelnen Zellen ausgebaut. Zusätzlich sind diese aber auch übereinander gestapelt, daher der Name der Technologie: 3D XPoint. Die Speicherdichte ist dabei gegenüber DRAM deutlich höher. Weiterhin kann 3D XPoint byte-adressierbar Daten lesen und schreiben, dies auch mit einer deutlich höheren Geschwindigkeit als konventioneller DRAM. Dennoch gibt es weiterhin Bereiche, in denen die neue Technologie noch Aufholbedarf hat, da im Praxistest bei Endanwendern die potentiell deutlich kürzeren Zugriffszeiten nicht vollends zum Tragen kommen. Aus einem von ComputerBase durchgeführten Benchmark [9] ergibt sich, dass im Realfall die Lesegeschwindigkeit der im Handel erhältlichen Optane Speicher bereits höher als die einer vergleichbaren SSD sei, die Schreibgeschwindigkeiten jedoch sehr ähnlich seien.

Abbildung 3: Intel 3D XPoint



Quelle: https://www.intel.de/content/www/de/de/architecture-and-technology/optane-technology-animation.html

4 Warum NVMM?

Das seit vielen Jahrzehnten existierende Konzept der Trennung in Haupt- und Massenspeicher ist wirtschaftlich und technologisch bedingt, stellt aber keinesfalls eine optimale Lösung dar. Aus diesem Grund ist die Forschung auf diesem Gebiet äußerst interessant, da hier massives Verbesserungspotential liegt. Besonders bei hoch performanten und verteilten Systemen ist IO als Flaschenhals für Performance zu sehen, sodass mit NVMM ein erster Ansatz zur Verschmelzung beider Speichertypen durchaus interessant ist. Wenn größerer und nichtflüchtiger Arbeitsspeicher verfügbar ist, kann die Anzahl an Operationen zum Kopieren von Daten aus dem Massenspeicher minimiert werden. Da beispielsweise die Module von Intel sowohl als Massen- als auch als Hauptspeicher einsetzbar sind, ist hier auf eine stetige Angleichung zu hoffen, da somit ein Stück Hardware beide Anwendungszwecke zufriedenstellend erfüllen kann. Die Möglichkeiten sind gleichermaßen mächtig als auch offensichtlich:

- Der Vorgang des Bootens entfällt komplett, das gesamte Betriebssystem befindet sich zu jeder Zeit im Arbeitsspeicher
- Paging (zur Info, [10]) wird langfristig überflüssig, da ein deutlich größerer Arbeitsspeicher zur Verfügung steht und irgendwann auf Hardwarebasis kein Unterschied zum Massenspeicher besteht.
- Eine Reduzierung der Anzahl an Komponenten erlaubt einheitlichere und damit potentiell höhere Taktraten von Hardware und Bussystemen

5 Problemanalyse

5.1 Herkömmliche Systeme

Sollte es bei einem beliebigen Computer zu einem schwerwiegenden Fehler in einem Anwendungsprogramm oder dem Betriebssystem kommen, ist für fast alle Anwender der Neustart des Computers eine schnelle und unkomplizierte Problemlösung. Dies funktioniert, da nach einem Neustart alle Daten erneut von der Festplatte geladen werden und somit wieder ein Grundzustand hergestellt werden kann, in dem das System läuft.

5.2 Systeme mit NVMM

Betreibt man einen Rechner mit einer identischen Konfiguration und Software wie im vorherigen Abschnitt genannt, verwendet aber NVMM als Hauptspeicher, läuft man zwangsläufig in folgendes Problem: Kommt man in die gleiche Situation wie oben, dass Betriebssystem oder Programme nicht mehr fehlerfrei laufen, muss Abhilfe geschaffen werden. Hier liegt jedoch eines der schwerwiegendsten Probleme von NVMM, da die Informationen hier nichtflüchtig und somit nach einem Neustart noch identisch vorhanden sind. Ein Betriebssystem kann sich in diesem Fall möglicherweise nicht selbst aus seinem Zustand befreien. Es kann nur eine Low-Level Routine unterhalb des Betriebssystems helfen, die einen Fehler aufspürt und einen kompletten Reset des Arbeitsspeichers veranlasst. Kann der Fehler aber nicht eindeutig lokalisiert werden, würde dies bedeuten,

dass alle Daten aus dem Speicher gelöscht werden müssten. Genau dies sollte aber vermieden werden, da bei einer Größe von möglicherweise hunderten von GB ein enormer Aufwand für solch eine Operation entsteht. Blickt man weiter in die Zukunft und stellt sich ein System vor, in dem Haupt- und Massenspeicher miteinander verschmolzen sind, wird das Problem noch drastischer, da hier alle Informationen nur an einem Ort liegen und somit nicht gelöscht werden können. Dies erfordert für die Zukunft ein grundlegendes Umdenken in der Konzeption von Datei- und Betriebssystemen, um solche Szenarien auszuschließen oder wenigstens effizient damit umgehen zu können. Die heutigen Formen von Journaling und Logging müssen stark abgeändert und an die neuen Systeme angepasst werden. Ob die redundante Speicherung innerhalb eines NVMM Moduls Abhilfe schafft, wird sich zeigen, zumindest könnte in diesem Fall eine aktive und eine passive Version jeder Information existieren. Sollte es zu einem Fehler kommen, können die Flags vertauscht werden und die hoffentlich stabile Version übernimmt den Reparationsprozess.

Nicht zu vernachlässigen sind auch weitere Auswirkung auf den Einsatz von Software. Zwar werden Algorithmen zum intelligenten Laden von Informationen aus dem Massen- in den Hauptspeicher überflüssig, trotzdem besteht auch hier Verbesserungsbedarf. In den vergangenen Jahrzehnten der Computergeschichte waren Prozessoren stets deutlich höher getaktet als die Speichereinheiten, sodass sich das Konzept der ‚Waitstates' durchgesetzt hat, die Prozessoren also oftmals eine Pause einlegen müssen, um auf ihre Daten zu warten. Zwar fällt dies durch cleveres Threading wie etwa nach [11] im Alltag nicht auf, aber viele Programme sind genau auf diese Eigenschaften ausgelegt. Wenn nun aber ein deutlich performanterer NVMM für Haupt- und Massenspeicher eingesetzt wird, sinken die Latenzen für den Zugriff auf Informationen massiv. Programme, die darauf ausgelegt sind während des Ladens von Daten andere Operationen durchzuführen, sind nun möglicherweise höchst ineffizient. Die revolutionäre Hardware erfordert in diesem Fall auch revolutionäre und effizientere Programmierung – auf der Ebene des Dateisystems sowie auch bei Anwenderprogrammen.

6 Abschlussworte

Der technische Fortschritt führt unweigerlich zu neuartigen Konzepten und Möglichkeiten. Diese können den Weg zu immer komplexeren und leistungsfähigeren Maschinen ebnen. Getrieben von dem Wunsch nach schnellen Ergebnissen können dabei auch falsche Wege eingeschlagen werden - oder Kompromisse eingegangen werden, die sich in der Zukunft als falsch heraus stellen. Der Einsatz von NVMM eröffnet Heimanwendern sowie professionellen Betreibern großer Systeme fantastische Chancen. So werden die Produkte der kommenden Generationen nicht nur schneller und energieeffizienter arbeiten, sondern auch immer robuster gegen Störungen und Ausfälle. Der Weg ist jedoch noch weit und erfordert nicht nur gänzlich neue Hardware und physikalische Erfindungen, sondern auch ein Umdenken in der Entwicklung von Software. Einige der grundlegenden Konzepte wie etwa die NUMA-Architektur und solche auf der Ebene von Dateisystemen wurden bereits in der Präsentation *Non-volatile main memory (NVMM) Speichersysteme* von Simon Oelgeschläger im Kurs Effiziente Programmierung vorgestellt. Sollten Hauptspeicher auf Basis von NVM erfolgreich sein, können wir uns alle in Zukunft über Rechner freuen, die ohne Zeitverzögerung hoch- und runterfahren, sowie viele Operationen, die eine größere Datenmenge bewegen, drastisch schneller ausführen.

Abbildungsverzeichnis

1 Eine DRAM Speicherzelle mit Kondensator und Transistor 5
2 Klassische Hierarchie verschiedener Speicher 6
3 Intel 3D XPoint 7

Literatur

[1] elektronik kompendium, "Computer - architektur." https://www.elektronik-kompendium.de/sites/com/1309261.htm Zugriff am 25.02.2020.

[2] J. Jordan, *Journaling Dateisysteme*. 2004. https://www4.cs.fau.de/Lehre/WS03/PS_KVBK/talks/Handout_Journaling.pdf.

[3] V. Atlidakis, J. Andrus, and R. Gaembasu, *POSIX Has Become Outdated*. 2016. http://www.cs.columbia.edu/~vatlidak/resources/POSIXmagazine.pdf.

[4] M. Myers, "Comptia a+ all in one: Prüfungsvorbereitung und hardware-buch." https://books.google.de/books?id=GxdO8Bpm4CYC&lpg=PA448&dq=ntfs&hl=de&pg=PA448#v=onepage&q=ntfs&f=false Zugriff am 23.02.2020.

[5] Intel, "Intel optane ssds." https://www.intel.de/content/www/de/de/products/memory-storage/solid-state-drives/consumer-ssds/optane-ssd-9-series.html Zugriff am 26.02.2020.

[6] J. Boukhobza and P. Olivier, *Flash Memory Integration: Performance and Energy Issues*. 2017. https://books.google.de/books?id=zEoADQAAQBAJ&lpg=PA221&dq=reram&hl=de&pg=PA221#v=onepage&q=reram&f=false.

[7] J. Handy, "Emerging memories today: The technologies: Mram, reram, pcm/xpoint, fram, etc.." https://thememoryguy.com/emerging-memories-today-the-technologies-mram-reram-pcm-xpoint-fram-etc/ Zugriff am 28.02.2020.

[8] S. Ghosh, *Sensing of Non-Volate Memory Demystified*. 2019. https://books.google.de/books?id=qqJoDwAAQBAJ&lpg=PA81&dq=optane%20speed&hl=de&pg=PA81#v=onepage&q=optane%20speed&f=false.

[9] M. Günsch, "Intel optane benchmarks." https://www.computerbase.de/2017-06/intel-optane-memory-test/2/#abschnitt_benchmarks Zugriff am 25.02.2020.

[10] V. Fachhochschule, "Swapping und paging." https://vfhcab.oncampus.de/loop/Swapping_und_Paging Zugriff am 28.02.2020.

[11] J. Anderson, "Intro to threading in python." https://realpython.com/intro-to-python-threading/ Zugriff am 24.02.2020.