Virtuelle Speicherverwaltung: Freund oder Feind großer Hauptspeicher?

Julian Lehrhuber
Friedrich-Alexander-Universität Erlangen-Nürnberg
Erlangen
julian.lehrhuber@fau.de

ABSTRACT

Virtuelle Speicherverwaltung hat in einem modernen Rechensystem eine große Bedeutung. Sie übernimmt viele Funktionen wie die Einteilung des Speichers in Seiten, die Abschottung von Anwendungen untereinander sowie von Anwendungen und Betriebssystem und den bedarfsabhängigen Seitenwechsel. Dieser ist vor allem dann relevant, wenn wenig Hauptspeicher zur Verfügung steht und deshalb Inhalte des Hauptspeichers kurzzeitig auf den Sekundärspeicher ausgelagert werden müssen. Durch neue Speichertechnologien wie NVRAM und Methoden wie FAM wird es nun allerdings möglich, nicht nur persistente sondern auch sehr große Hauptspeicher zu realisieren und damit den traditionellen Aufbau der Speicherhierarchie zu beeinflussen. In dieser Arbeit werden daher die aktuellen Einsatzgebiete von virtueller Speicherverwaltung definiert, sowie die Möglichkeiten und Auswirkungen von großen Hauptspeichern auf virtuelle Speicherverwaltung betrachtet. Es stellt sich heraus, dass nicht alle Aspekte virtueller Speicherverwaltung nahtlos auf die Anforderungen von großen Hauptspeichern und neuen Speichertechnologien übertragen werden können, sowie dass vor allem die Verwendung von NVRAM neue Anforderungen an die Speicherverwaltung stellt. Am Beispiel aktueller Veröffentlichungen werden zudem neue Methoden der Speicherverwaltung vorgestellt.

1 EINFÜHRUNG

Virtuelle Speicherverwaltung erfüllt in einem modernen Rechensystem vielerlei Zwecke. Zunächst ist sie dafür verantwortlich, dass jeder Prozess einen eigenen virtuellen Adressraum sieht, der ihm als linear und lückenlos adressierbar erscheint. Diese virtuellen Adressräume werden durch eine Übersetzungsschicht realisiert, welche die von einer Recheneinheit angeforderten virtuellen Adressen in physische Adressen umwandelt und die konfigurierten Berechtigungen von Zugriffen überprüft. Der Schutzmechanismus basiert in modernen Rechensystemen auf dem Konzept der Speicherseite und dient der Abschottung zwischen Anwendungen, sowie zwischen Anwendungen und dem Betriebssystem. Die Zuordnungen von virtuellen zu physischen Adressen werden dabei in der sog. *Seitentabelle*, welche i.d.R. im Hauptspeicher abgelegt wird, gespeichert. Darüber hinaus dient die virtuelle Speicherverwaltung aber auch dem sog. bedarfsabhängigen Seitenwechsel. Dieser erlaubt es dem Rechensystem, Teile des Hauptspeichers auf Sekundärspeicher aus- und bei Bedarf wieder einzulagern. Durch diesen Seitenwechsel wird also der real existierende Hauptspeicher durch „virtuellen“, d.h. nicht real existierendem Hauptspeicher, erweitert.

Betrachtet man die in der virtuellen Speicherverwaltung involvierten Speichertechnologien, finden sich dort vor allem als Hauptspeicher eingesetzter DRAM sowie HDDs und SSDs als Sekundärspeicher. In den vergangenen Jahren ließ sich jedoch auch eine zunehmende Verbreitung von unterschiedlichen nichtflüchtigen RAM-Technologien (NVRAM) erkennen, die HDDs und SSDs als Sekundärspeicher aber auch DRAM als Hauptspeicher ablösen könnten [13]. Besonders hinsichtlich der byteweisen Adressierbarkeit sowie der Zugriffszeiten auf zufällige Speicherzellen lassen sich bestimmte NVRAM-Technologien eher mit DRAM vergleichen und haben damit einen großen Vorteil gegenüber HDDs. Im Falle von Phase-Change Memory (PCM) lassen sich aufgrund der höheren Speicherdichte sogar zwei bis viermal größere Speichervolumen realisieren als mit DRAM [11].

Eine weitere aufstrebende Hauptspeicherarchitektur ist der sog. *Fabric-Attached Memory* (FAM). Besonders im Kontext der speicherorientierten Datenverarbeitung, bei welcher mehrere Rechensysteme auf gemeinsamen, über ein Netzwerk verbundenen Hauptspeicher zugreifen, hat FAM eine wichtige Bedeutung [9]. Durch FAM und NVRAM lassen sich riesige Hauptspeichervolumen realisieren, die den aktuell auf 256 TB limitierten virtuellen Adressraum einer 64-Bit Recheneinheit weit übersteigen können.

Vor diesem Hintergrund beschäftigt sich diese Arbeit mit den Herausforderungen aber auch Chancen für die Betriebssystementwicklung, die durch große und persistente Hauptspeicher entstehen. Dazu wird im Folgenden zunächst auf die charakteristischen Eigenschaften von NVRAM und die Auswirkungen auf virtuelle Speicherverwaltung eingegangen, bevor der Effekt von großen Hauptspeichermengen auf die Seitentabelle und die daraus resultierenden Konsequenzen beleuchtet werden. In welcher Weise muss sich virtuelle Speicherverwaltung verändern, um große und persistente Hauptspeicher sinnvoll in ein Rechensystem integrieren zu können? Müssen Aspekte der virtuellen Speicherverwaltung wie der Einteilung des Speichers in Seiten, der bedarfsabhängige Seitenwechsel, die Adressumsetzung und der Speicherschutz möglicherweise sogar neu gedacht werden? Darauf folgend werden exemplarisch am Beispiel zweier Veröffentlichungen die Möglichkeiten alternativer Ansätze zur virtuellen Speicherverwaltung vorgestellt. Abschließend wird die Gesamtheit der Themen diskutiert und ein Fazit gebildet.

Typ	DRAM	PCM	SST-RAM	RRAM	NAND
nichtflüchtig	Nein	Ja	Ja	Ja	Ja
Byteweise adressierbar	Ja	Ja	Ja	Ja	Nein
Zugriffseinheit (Bytes)	64	64	64	64	4096
Lebensdauer (Schreibzyk.)	$> 10^{16}$	10^{10}	10^{15}	10^{8}	10^{5}
Leseenergie (J/GB)	0,8	1	1	0,25	1,5
Schreibenergie (J/GB)	1,2	6	21,74	14,02	17,5
Leselatenz (ns)	20-50	50	3,06	1,9	$25 \cdot 10^3$
Schreiblatenz (ns)	20-50	150	25,45	100	$500 \cdot 10^3$

Tabelle 1: Vergleich von unterschiedlichen Speichertechnologien nach Chen et al. [4]

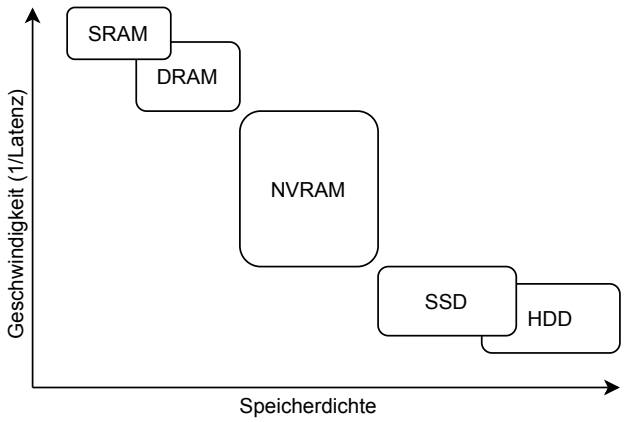

Abbildung 1: Speicherhierarchie nach Hoya et al. [8]

2 WELCHEN UNTERSCHIED MACHT NVRAM?

NVRAM ist die Oberbezeichnung für eine Gruppe nichtflüchtiger, byteweise adressierbarer Speichertechnologien wie PCM, SST-RAM und RRAM. In Tabelle 1 werden diese Technologien sowie DRAM und NAND hinsichtlich ihrer Eigenschaften gegenübergestellt. Besonders hervorzuheben ist der große Vorteil von NVRAM-Medien hinsichtlich ihrer Lese- & Schreiblatenz gegenüber einem NAND-Medium, sowie die Tatsache dass NVRAM-Medien im Vergleich zu DRAM diese Latenzen zusätzlich zur Erhaltung der Daten bei Stromverlust garantieren. Außerdem können NVRAM-Medien besonders in der benötigten Lese-Energie mit der DRAM-Technologie mithalten oder diese sogar unterbieten, was NVRAM-Medien besonders für batteriebetriebene, eingebettete Systeme auf welchen überwiegend Leseoperationen ausgeführt werden interessant macht [4].

Wie in Abbildung 1 erkennbar bietet NVRAM nicht nur einen großen Vorteil hinsichtlich der vergleichsweise schlechten Latenz von NAND-Medien, sondern bietet zudem eine höhere Speicherdichte als DRAM. Daher würde sich NVRAM nicht nur als schnelle Alternative zu NAND-Medien anbieten, sondern auch um heutige DRAM-Hauptspeicher durch NVRAM-Speicher zu ersetzen und damit noch größere Hauptspeicherkapazitäten günstig zu realisieren. Aktuell bietet NVRAM also gewissermaßen das Beste aus beiden Welten: Geringe Latenzen sowie geringer Energieverbrauch, byteweise Adressierbarkeit und große, persistente Speichervolumen. Dadurch lassen sich nun folgende, von der herkömmlichen Speicherarchitektur abweichende Konfigurationen abbilden:

(1) Ausschließliche Verwendung von NVRAM als Hauptspeicher. Sekundärspeicher könnte aufgrund des persistenten Hauptspeicher möglicherweise vollständig entfernt werden.
(2) Verwendung von NVRAM als Sekundärspeicher. DRAM-Hauptspeicher könnte zudem verkleinert werden, da Sekundärspeicher byteweise adressierbar ist.

Beide Konfigurationen teilen sich ein Merkmal: Sie suggerieren den Verzicht auf bedarfsabhängigen Seitenwechsel. Während der Konfiguration aus Punkt 1 ohne Sekundärspeicher kein Medium für das Auslagern von Hauptspeicherinhalten zur Verfügung steht, kann bei der Konfiguration aus Punkt 2 der bedarfsabhängige Seitenwechsel aufgrund der ähnlichen Latenzen von DRAM und NVRAM mindestens infrage gestellt werden.

Im Nachfolgenden steht die hohe Dichte dieser Speichertechnologien im Fokus. Er wird eine Verwendung des Speichers wie klassischer DRAM angenommen, d.h. er wird wie flüchtiger Speicher behandelt - ungeachtet der Möglichkeiten zur persistenten Speicherung von Daten.

2.1 Auswirkungen auf bedarfsabhängigen Seitenwechsel

Die Sinnhaftigkeit von bedarfsabhängigem Seitenwechsel in Systemen mit großem Hauptspeicher lässt sich am Beispiel der Arbeit von Chen et al. [4] näher diskutieren. In ihrer Veröffentlichung untersuchten Chen et al. die Auswirkung von NVRAM auf das Zwischenspeichern von Seiteninhalten, wenn dieser sowohl als Haupt- als auch Sekundärspeicher eingesetzt wird. Das Zwischenspeichern von Seiteninhalten ist ein traditioneller Mechanismus, um erhöhte Latenzen beim Zugriff auf Sekundärspeicher zu verbergen. Wie in Tabelle 1 erkennbar, fallen beim Zugriff auf herkömmliche SSDs (und auch HDDs) mindestens 1000-mal höhere Latenzen als beim Zugriff auf DRAM oder NVRAM an. Werden häufig verwendete Seiten im Hauptspeicher zwischengespeichert, entfallen diese erhöhten Latenzen, da die Inhalte direkt vom Hauptspeicher abgerufen werden können. Da beim Einsatz von NVRAM als Sekundärspeicher diese erhöhten Latenzen grundsätzlich entfallen und mit dem Zwischenspeichern von Sekundärspeicherinhalten weitere, nicht notwendige Datenbewegungen einhergehen, ist also die Frage nach der Sinnhaftigkeit der Zwischenspeicherung von Seiteninhalten berechtigt.

Wie in Abbildung 2 dargestellt bewegen sich beim bedarfsabhängigen Seitenwechsel zunächst Daten aus dem Hauptspeicher zum Sekundärspeicher, bevor sie von dort bei Notwendigkeit wieder in den Hauptspeicher geladen werden. Diese Datenbewegung ist äquivalent zum Zwischenspeichern von Seiteninhalten des Sekundärspeichers: Werden Daten des

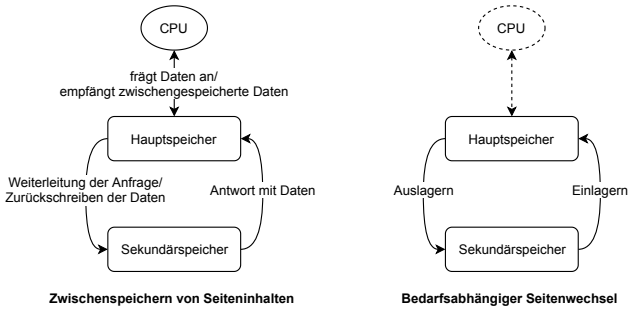

Abbildung 2: Vereinfachte Darstellung der Datenbewegungen beim Zwischenspeichern von Seiteninhalten und bedarfsabhängigem Seitenwechsel

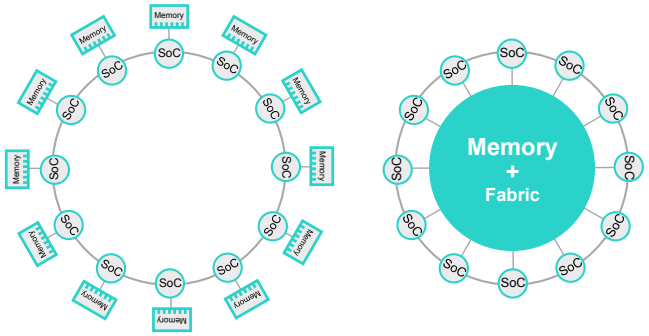

Abbildung 3: Unterschied von prozessororientierter zu speicherorientierter Datenverarbeitung nach Keeton [9]

Sekundärspeichers angefordert, werden diese im Hauptspeicher zwischengespeichert und wieder in den Sekundärspeicher zurückgeschrieben, falls notwendig.

Chen et al. untersuchten die Auswirkung einer von ihnen vorgestellten Technik, die es ihnen unter anderem erlaubte, Daten direkt vom NVRAM-Sekundärspeicher in die Zwischenspeicherzeile der Recheneinheit einzulesen - ohne diese im Hauptspeicher zwischenzuspeichern. Abhängig von der verwendeten NVRAM-Technologie und der Software zum Leistungstest konnten sie Latenzverringerungen im Bereich von 60%-85% für Leseoperationen und 56%-86% für Schreiboperationen messen. Basierend auf diesen Erkenntnissen ist also davon auszugehen, dass ebenso ein bedarfsabhängiger Seitenwechsel bei einer Konfiguration von NVRAM als Haupt- und/oder Sekundärspeicher nicht sinnvoll ist.

2.2 Das Problem

NVRAM hat zum aktuellen Zeitpunkt aber nicht nur Vorteile. Wie in Tabelle 1 erkennbar erschwert nicht nur die Asymmetrie der Lese- und Schreiboperationen den Einsatz von nichtflüchtigen RAM-Technologien als Ersatz für DRAM, sondern auch die Lebensdauer. Während herkömmlicher DRAM heute mehr als 10^{16} Schreibzyklen erlaubt, erreicht NVRAM im Falle von SST-RAM bestenfalls 10^{15} Schreibzyklen. PCRAM, der Hoffnungsträger als Ersatz für DRAM [11, 12] hingegen bietet lediglich 10^{8} Schreibzyklen und unterliegt DRAM in dieser Hinsicht damit deutlich. Da die Lebensdauer für jede Speicherzelle individuell gilt, kann die Gesamt-Lebensdauer des NVRAM-Mediums durch geschickte Verteilung der Schreiboperationen (sog. *wear leveling* [Abnutzungsnivellierung]) verlängert werden [12, 15].

Algorithmen zur Abnutzungsnivellierung können sowohl in Hardware als auch in Software implementiert werden. Bei einer Hardwareimplementierung wäre die Adressierung der NVRAM-Speicherzellen natürlich transparent für die Anwendung/das Betriebssystem. Allerdings steigen nach Qureshi [14] mit einer Hardwareimplementierung auch die Anforderungen an die Hardware. Da viele Algorithmen mit Tabellen und Zählern für oft genutzte Speicherzellen arbeiten und diese Strukturen in der Lage sein müssen Metadaten für den kompletten Speicherbereich zu halten, werden sie entsprechend groß. Qureshi et al. vertreten daher die Ansicht, dass hardwaregestützte Abnutzungsnivellierung auf einer algebraischen Lösung basieren muss. Bei dieser werden keine statistischen Daten über die Speicherzugriffe o.Ä. erhoben, sondern Schreibzugriffe anhand einer Formel über die Speicherzellen hinweg verteilt.

Allerdings sind auch Softwareimplementierungen der Abnutzungsnivellierung denkbar; beispielsweise könnte diese Funktion auch ins Betriebssystem integriert [12] und Teil der virtuellen Speicherverwaltung werden. Betrachtet man nun die angesprochenen Auswirkungen von NVRAM auf virtuelle Speicherverwaltung wird klar, dass für diese neuartige Speichertechnologie ein Umdenken in der virtuellen Speicherverwaltung erfolgen muss. Nicht nur ist der Einsatz von bedarfsabhängigem Seitenwechsel stark von der Konfiguration des Rechensystems abhängig, sondern auch die begrenzte Lebensdauer von NVRAM beeinflussen den Umgang mit NVRAM-Hauptspeicher stark.

3 EINFLUSS GROßER HAUPTSPEICHER AUF VIRTUELLE SPEICHERVERWALTUNG

Nicht nur NVRAM begünstigt bereits heute schon den Einsatz von großen Hauptspeichermengen, sondern auch Architekturen wie FAM. FAM findet seinen Einsatz vor allem im Rahmen der speicherorientierten Datenverarbeitung. Dort werden wie in Abbildung 3 erkennbar mehrere Rechensysteme über ein Netzwerk an einen sehr großen, gemeinsamen Hauptspeicher angeschlossen. Beim Projekt „The Machine“ [10] besteht dieser Rechenverbund aus 40 Systemen, die jeweils mit bis zu 4 TB geteiltem, nichtflüchtigem Hauptspeicher ausgestattet sind. Daraus ergeben sich 160 TB geteilter Hauptspeicher. Wir erinnern uns: eine aktuelle 64-Bit Recheneinheit besitzt heute einen 48-Bit virtuellen Adressraum, kann also bis zu

$$2^{48} \text{ Bytes} = 256 \text{ TB}$$

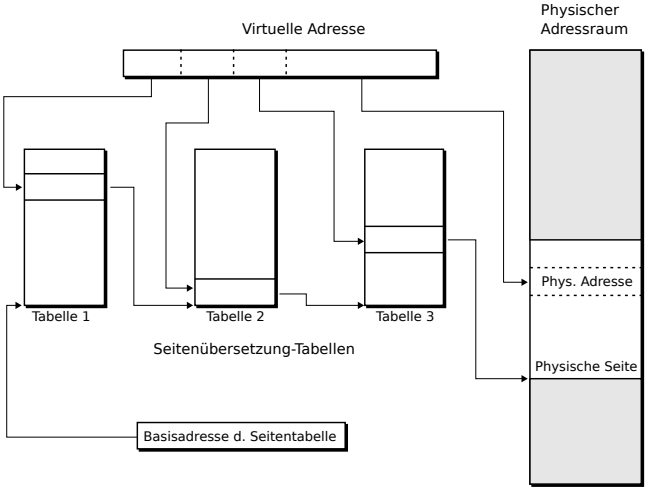

Abbildung 4: Auf drei Stufen vereinfachte Darstellung einer mehrstufigen Seitentabelle nach [5]

Speicher adressieren. Mit „The Machine“ sind also bereits 62,5% des insgesamt zur Verfügung stehenden virtuellen Adressraums genutzt. Ein Erweitern des gemeinsamen Hauptspeichers um weniger als 2,5 TB pro Rechenknoten oder die Zuschaltung von nur mehr 24 gleichartigen Rechenknoten würde also bereits den gesamten virtuellen Adressraum ausreizen. In den folgenden Abschnitten soll daher auf die Auswirkungen von solch großen oder noch größeren Hauptspeichermengen auf die virtuelle Speicherverwaltung eingegangen werden.

3.1 Auswirkungen auf die Seitentabelle

Virtuelle Adressumsetzung erfolgt durch Seitentabellen, welche die Zuordnung einer virtuellen zu einer physischen Adresse enthält. Sie wird i.d.R. im Hauptspeicher des Systems abgelegt und ist in modernen Systemen mehrstufig aufgebaut [5]. Wie in Abbildung 4 dargestellt strukturiert sich eine virtuelle Adresse bei Verwendung einer mehrstufigen Seitentabelle in mehrere Regionen. In den höherwertigen Bits befinden sich jeweils eine Region für eine Stufe der Seitentabelle, die den Versatz in dieser Stufe angibt. Die verbleibenden Bits der virtuellen Adresse geben dann den Versatz der Adresse in der physischen Seite an.

Der mehrstufige Aufbau bringt den Vorteil, dass nicht die komplette Seitentabelle zu jeder Zeit im Hauptspeicher gehalten werden muss. Betrachtet man die zu erwartende Größe einer einstufigen Seitentabelle in der IA-32-Architektur wird schnell klar, dass die mehrstufige Hierarchie benötigt wird: Bei einer für die IA-32-Architektur typischen Seitengröße von 4 KB kann die Seitentabelle bis zu

$$2^{32}\ \text{Bytes} \div 2^{12}\ \text{Bytes} = 2^{20}$$

Einträge fassen, was bei einer Eintragsgröße von 4 Bytes bereits 2 MB entspricht. Für die AMD64-Architektur wächst die Tabellengröße (2 MB Seitengröße) bereits auf

$$2^{48}\ \text{Bytes} \div 2^{21}\ \text{Bytes} = 2^{27}$$

Einträge, also 1 GB bei einer Eintragsgröße von 8 Bytes. Um die Größe der Seitentabelle zu reduzieren wird gerne die Seitengröße erhöht - man spricht dann von sog. *huge pages* (großen Seiten).

Diese großen Seiten bedeuten laut Bresniker et al. [3] allerdings Sicherheitsrisiken für die Speicherverwaltung. Nicht nur sollen solche Seiten anfällig gegen Fehler oder bösartige Angriffe sein, weil auf jede Adresse in einer solch großen Seite ohne granulare Kontrolle zugegriffen werden kann. Bresniker et al. vertreten auch die Ansicht, dass große Übersetzungeinheiten, also große Seiten, fundamental inkompatibel zur Idee feiner Zugriffskontrolle seien. Zusätzlich erfahren nur manche Arbeitslasten einen Vorteil von großen Seiten, während andere stark davon beeinträchtigt werden.

3.2 Einfluss von Adressbits

Um mehr als 256 TB Hauptspeicher adressierbar zu machen, könnte man zunächst natürlich die Anzahl der Adressbits erhöhen. Dies ist in der Vergangenheit bereits bei 32-Bit Recheneinheiten mittels PAE (*Physical Address Extension* [physische Adresserweiterung]) passiert, damit diese mehr als 4 GB Hauptspeicher adressieren können. Während für 32-Bit Recheneinheiten auf Techniken wie PAE zur Erweiterung des virtuellen Adressraums zurückgegriffen werden musste, könnte dieser bei einer 64-Bit CPU durch Aufstocken der Adressbits erweitert werden. Mit einem 64-Bit virtuellen Adressraum würden folglich maximal

$$2^{64}\ \text{Bytes} = 16384\ \text{PB} = 16\ \text{EB}$$

Speicher adressiert werden können. Die Seitentabelle - erneut bei einer Seitengröße von 2 MB und einer Eintragsgröße von 8 Byte - würde in diesem Fall bis zu

$$2^{64}\ \text{Bytes} \div 2^{21}\ \text{Bytes} = 2^{43}$$

Einträge, also 64 TB groß werden können.

Solch große Datenstrukturen bedeuten nicht nur einen sehr hohen organisatorischen Aufwand. Wissenschaftler sind sich einig [6, 7], dass sich die Erweiterung der Adressbits negativ auf das Rechensystem auswirken wird. Neben dem erhöhten Energieverbrauch sowie Herstellungskosten [7] hat die Erweiterung der Adressbits auch Auswirkungen auf die sog. *TLB-miss* Latenz [6]. Der TLB (*translation look-aside buffer* [Übersetzungpuffer]) ist ein Zwischenspeicher für kleine Teile der Seitentabelle.

Vor dem Hintergrund von persistenten Hauptspeichertechnologien ist zudem die Frage der idealen Größe einer Speicherseite ungeklärt [2]. Traditionellerweise ist die Seitengröße in der virtuellen Speicherverwaltung ein Maß für die Größe der jeweiligen Speicherallokation und der Granularität des Speicherschutzes. Die Seitengröße wird i.d.R. so gewählt, dass Speicherfragmentierung bestmöglich vermieden sowie die Betriebslast der Seitentabellenstruktur und beim Transfer der Daten zwischen Haupt- und Sekundärspeicher minimiert wird. Mit den in Abschnitt 2 beschriebenen und durch NVRAM möglichen Speicherkonfigurationen ist allerdings aufgrund von beispielsweise fehlendem Sekundärspeicher nicht mehr klar, welche Seitengröße ideal ist. Ohne Sekundärspeicher ist

möglicherweise sogar das Konzept der Speicherseite obsolet, da nur mehr ein einziger Speicherlevel besteht. Wie angedeutet basiert auf Speicherseiten auch der Speicherschutz. Beim Wegfall von Speicherseiten müsste also zudem auf eine Alternative zu seitenbasiertem Speicherschutz - besonders zur Verwendung mit persistenten Hauptspeichern - zurückgegriffen werden.

3.3 Auswirkungen auf Speicherschutz

Den Zugriff auf große, persistente Hauptspeicher zu schützen wird wie eben dargestellt aufgrund der veränderten Speicherhierarchie alternative Schutzmechanismen benötigen. Achermann et al. [1] vertreten die Ansicht, dass MMU-gestützte Schutzmechanismen vor allem aufgrund eines derzeit noch unspezifizierten Mechanismus zum persistenten Speichern von Metadaten wie den Zugriffsrechten für nichtflüchtige Hauptspeicher ungeeignet ist. Stattdessen stellen sie zwei mögliche Schutzmechanismen vor: „Hardwarefähigkeiten“ (*hardware capabilities*) und Speicherschlüssel. Eine alternative Speicherverwaltungsmethode die auf Hardwarefähigkeiten im Betriebssystem „Barrelfish“ setzt, ist das später vorgestellte „SpaceJMP“ [6] sowie die Arbeit von Gerber et al. [7].

Beim Konzept der Speicherschlüssel wird der Speicher in gleich große Blöcke aufgeteilt, während jeder dieser Blöcke mit einem Schlüssel versehen wird. Dabei wird der Zugriff auf einen Speicherblock dann gewährt, wenn der Schlüssel im Statusregister mit dem des Speicherblocks übereinstimmt. Ein System, das Speicherschlüssel unterstützt, ist beispielsweise IBM's „System/360“. Laut Achermann wägt aber auch Intel die Einführung von Speicherschlüsseln in künftige Prozessoren ab.

4 ALTERNATIVE SPEICHERVERWALTUNGSSTRATEGIEN

Bevor im Folgenden alternative Speicherverwaltungsstrategien vorgestellt werden, fassen wir noch einmal die möglicherweise notwendigen Änderungen an der heute üblichen Umsetzung von virtueller Speicherverwaltung bei Verwendung großer, nichtflüchtiger Speichertechnologien zusammen.

Bedarfsabhängiger Seitenwechsel. Aufgrund der geringen Zugriffslatenzen von NVRAM kann je nach Systemkonfiguration auf bedarfsabhängigen Seitenwechsel verzichtet werden. Hinsichtlich der Größe des Speichers entfällt gleichzeitig der Bedarf eines Sekundärspeichers, womit keine Möglichkeit mehr zur Auslagerung von Hauptspeicherinhalten besteht.

Adresszuordnung. Virtuelle Speicherverwaltung umfasst bereits heute eine Übersetzungfunktion, die virtuelle in physische Adressen übersetzt. Dies dient aktuell lediglich der Bereitstellung von linearen und lückenlos adressierbaren Adressräumen für Anwendungen. Da die Lebensdauer von NVRAM noch der von DRAM unterlegen sind, wird zum Einsatz als Hauptspeicher eine Art von Abnutzungsnivellierung benötigt. Diese Funktion ließe sich als Teil des Betriebssystems, insbesondere als Teil der Adresszuordnung der virtuellen Speicherverwaltung, realisieren.

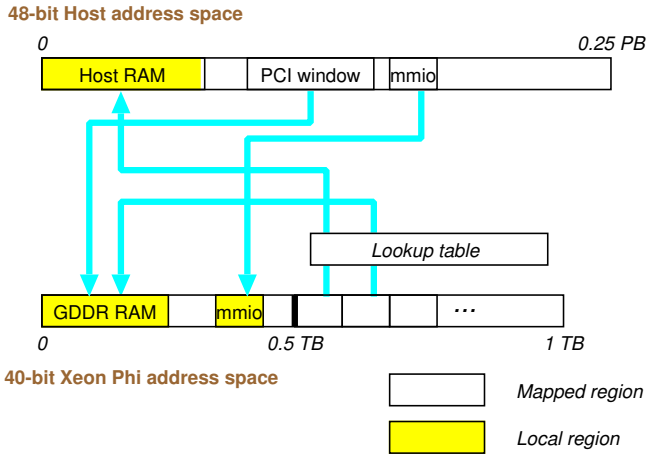

Abbildung 5: Xeon Phi Speicherlayout nach Gerber et al. [7]

Speicherseiten. Virtuelle Speicherverwaltung bedient sich der Einteilung des Speichers in Seiten. Speicher wird in Granularität dieser Seiten alloziert und geschützt. Aufgrund der mit NVRAM neuen Möglichkeiten hinsichtlich der Speicherkonfiguration ist allerdings noch ungeklärt, ob das Konzept der Speicherseiten weiter bestand hat.

Adressbits. Das Erweitern des virtuellen Adressraums durch Aufstockung der Adressbits ist zwar grundsätzlich möglich, aber hinsichtlich der Kosten in sowohl Anschaffung als auch Betrieb nicht wünschenswert.

Speicherschutz. Besonders abhängig von der Einteilung des Speichers ist der Speicherschutz. Sollte Aufgrund der neuartigen Speicherhierarchien das Konzept der Speicherseite verändert werden, müssen die Auswirkungen auf den Speicherschutz bedacht werden. Bereits heute existieren alternative Schutzmechanismen zum traditionellen Speicherschutz mittels MMU, darunter Hardwarefähigkeiten und Speicherschlüssel.

4.1 Neustrukturierung physischen Speichers

Gerber et al. vertreten die Ansicht, dass in modernen Rechensystemen viele Annahmen über den physischen Adressraum nicht mehr gelten [7]. Unter diesen Punkten befinden sich unter anderem die Annahme, dass jede Recheneinheit zu jeder Zeit den gesamten physischen Adressraum adressieren kann, sowie dass jede Recheneinheit die gleiche physische Adresse für eine bestimmte Speicherzelle verwendet. Sie widerlegen diese Annahmen exemplarisch am Beispiel der Xeon Phi Recheneinheiten, die als Koprozessoren in ein Rechensystem eingesetzt werden können. Wie in Abbildung 5 erkennbar wird der Hauptspeicher des Xeon Phi Koprozessors an einer gewissen Stelle im physischen Adressraum eingeblendet. In einem Rechensystem können mehrere Koprozessoren verbaut werden. Diese können wiederum den Hauptspeicher eines anderen Xeon Phi Koprozessors oder Teile des Hauptspeichers des Gastgebersystems in ihren eigenen Adressraum einblenden. Somit sind die Adressen für den Zugriff auf bestimmte

Speicherzellen weder für alle Xeon Phi Koprozessoren, noch für das Gastgebersystems und einen Koprozessor identisch.

Um diesen Adressüberschneidungen entgegenzuwirken stellen Gerber et al. eine neue Methode zur Adressierung physischen Speichers vor. In Abbildung 6 ist zu erkennen, wie alle Speicherbereiche der in einem Rechensystem integrierten Komponenten zunächst in den physischen Adressraum des Gastgebersystems eingeblendet werden. Von diesem physischen Adressraum wird für jede Systemkomponente ein virtueller Adressraum erstellt, der die notwendigen Abbildungen auf spezielle Speicherbereiche enthält. Eine wichtige Eigenschaft dieser Struktur ist, dass es auch „Zwischen-Adressräume“ geben kann, die die Adressen von einem unterliegenden Adressraum weiter abstrahieren. Diese Funktion dient dazu, dass jede real existierende Speicherzelle lediglich einmal in maximal einem virtuellen Adressraum abgebildet ist. Fordern mehrere Komponenten Zugriff auf diese Adresse, wird in den virtuellen Adressräumen dieser Komponenten lediglich auf die Adresse des „Zwischen-Adressraums“ abgebildet. Die entstehende Struktur gleicht nun eher einer Baumstruktur, die im Gegensatz zu der für Adresszyklen anfälligen traditionellen Adressierungsmethode ein klareres und konsistenteres Schema darstellt.

Die von Gerber et al. vorgestellte Art der Adressierung physischen Speichers ist wohl vor allem für kleine, speicherorientierte Rechensysteme mit FAM interessant. Durch den strukturierten Aufbau der Adressräume könnte jede Recheneinheit mit der gleichen Adresse auf eine bestimmte Speicherzelle zugreifen. Da jede Komponente und sogar jede Anwendung Kontrolle über ihren eigenen virtuellen Adressraum hat, kann sogar mehr Speicher adressiert werden als mit dem traditionellen Adressierungsmodell. Selbst der Speicherschutz scheint sich mit diesem Modell zu vereinfachen, da jede reale Speicherzelle nur in genau einem virtuellen Adressraum abgebildet ist und damit zum Ändern der Zugriffsrechte nicht alle virtuellen Adressräume durchgesucht werden müssen.

4.2 SpaceJMP: Alternative Verwaltung virtueller Adressräume

Eine weitere, alternative Adressierungsmethode ist „SpaceJMP“, die Arbeit von Hajj et al. [6]. Ihr Ansatz ist, virtuelle Adressräume als sog. *first-class citizens* im Betriebssystem zu behandeln, an welche sich Fäden an- und abdocken können. Zudem wird ermöglicht, dass Fäden zwischen mehreren virtuellen Adressräumen wechseln, um noch mehr Speicher adressieren zu können. Dazu werden die Adressräume nummeriert und eine Schnittstelle bereitgestellt, womit Anwendungen ihren virtuellen Adressraum verwalten können. Durch die An- und Abkopplungsmöglichkeit ist es zudem möglich, dass ein Adressraum über die Laufzeit eines Fadens hinweg existiert, sowie mehrere Fäden an denselben Adressraum andocken und somit Speicherfragmente zu teilen.

Um Speicherschutz zu gewährleisten greifen Hajj et al. auf das im jeweiligen Betriebssystem (die Referenzimplementierung erfolgte in „Barrelfish“) gängige Rechtesystem zurück. Vor allem zum Teilen von Speicherfragmenten ist diese Entscheidung der traditionellen Implementierung überlegen, da die Zugriffskontrolle in dieser über eine Datei geregelt wird, die die Übersetzung zwischen unterschiedlichen Rechtemodellen erfordert. Zudem sind SpaceJMP-Segmente ähnlich des Monitor-Konzeptes sperrbar. Dies bedeutet, dass bei einer aktiven Lese-Sperrung mehrere Lese-Anfragen aber keine Schreib-Anfrage bewilligt werden, während bei einer Schreib-Sperrung die Schreibanfrage alleinigen Zugriff auf das Segment zugesichert bekommt. SpaceJMP hat vor allem in der speicherorientierten Datenverarbeitung große Bekanntschaft erreicht, sodass sogar Keeton [9] im Kontext von „The Machine“ über das Adressierungsmodell berichtet hat.

Es lässt sich erkennen, dass SpaceJMP vor allem zum einfachen Teilen von Speicherinhalten zwischen Fäden entworfen wurde. Dies macht das Modell im Feld der speicherorientierten Datenverarbeitung, wo mehrere Recheneinheiten auf einen großen, gemeinsamen Speicherbereich zugreifen können, nützlich. Durch die Verwendung von Speichersegmenten würde sich das Adressierungsmodell womöglich auch für den Einsatz in ausschließlich auf NVRAM basierenden Rechensystemen eignen.

5 FAZIT

In dieser Arbeit wurden die Möglichkeiten und Auswirkungen von großen Hauptspeichern auf virtuelle Speicherverwaltung diskutiert. Vor allem durch neue Speichertechnologien wie NVRAM aber auch Methoden wie FAM wird es in Zukunft möglich, sehr große Hauptspeicher zu realisieren. Am Beispiel von NVRAM und dessen Charakteristiken hinsichtlich Zugriffslatenzen, Speicherdichte, Speicherpersistenz und Lebensdauer wurden Aspekte wie die Einteilung des Speichers in Seiten, der bedarfsabhängige Seitenwechsel und der Speicherschutzmechanismus von virtueller Speicherverwaltung betrachtet. Durch die mit DRAM vergleichbaren Zugriffslatenzen und große Speicherdichte von NVRAM kann auf die Verwendung von langsamen Sekundärspeichern wie SSDs und HDDs verzichtet werden. Dies wiederum hat großen Einfluss auf den bedarfsabhängigen Seitenwechsel, dessen Sinnhaftigkeit daher infrage gestellt wurde. Allerdings kann NVRAM nicht die gleiche Lebensdauer wie DRAM aufzeigen, weshalb Techniken zur Abnutzungsnivellierung benötigt werden. Ein solcher Mechanismus kann als Funktion der virtuellen Speicherverwaltung in das Betriebssystem integriert werden.

Zudem wurden die aktuellen Grenzen der virtuellen Speicherverwaltung aufgezeigt. Es wurde die Adressknappheit hinsichtlich großer Hauptspeicher sowie mögliche Lösungen und Auswirkungen großer Hauptspeicher auf die Seitentabelle und den Speicherschutz diskutiert. Dies zeigte, dass aufgrund der mit großen und persistenten Hauptspeichern mögliche Speicherkonfigurationen das Konzept der Speicherseite womöglich obsolet ist und eine Einteilung in Segmente wieder praxisrelevanter wird. Abschließend wurden zwei neuartige Methoden zur Realisierung virtueller Speicherverwaltung mittels Hardwarefähigkeiten vorgestellt.

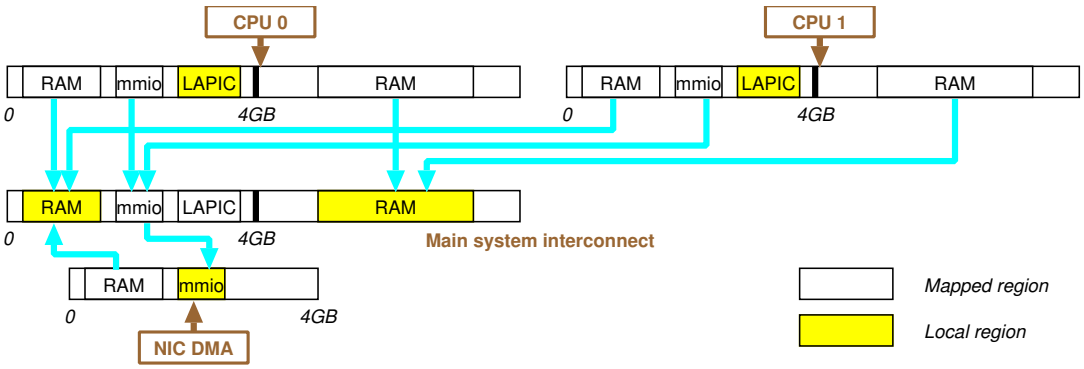

Abbildung 6: Adressräume in einem vereinfachten Zweikern-Rechensystem mit einer 32-Bit PCI Netzwerkkarte mit DMA nach Gerber et al. [7]

Die Umsetzung von virtueller Speicherverwaltung wird sich besonders bei Einsatz neuer Speichertechnologien verändern müssen. Da die gewählte Speichertechnologie sowie die daraus resultierende Speicherkonfiguration starken Einfluss auf die Anforderungen der virtuellen Speicherverwaltung haben, ist sich eine Universallösung schwierig vorzustellen. Besonders für große Hauptspeicher zeichnet sich allerdings der Trend ab, den Speicher nicht weiterhin in Seiten sondern in Segmente einzuteilen.

LITERATUR

[1] Reto Achermann, Chris Dalton, Paolo Faraboschi, Moritz Hoffmann, Dejan Milojicic, Geoffrey Ndu, Alexander Richardson, Timothy Roscoe, Adrian L Shaw, and Robert NM Watson. 2017. Separating translation from protection in address spaces with dynamic remapping. In *Proceedings of the 16th Workshop on Hot Topics in Operating Systems*. ACM, 118–124.

[2] Katelin Bailey, Luis Ceze, Steven D Gribble, and Henry M Levy. 2011. Operating System Implications of Fast, Cheap, Non-Volatile Memory.. In *HotOS*, Vol. 13. 2–2.

[3] Kirk M Bresniker, Paolo Faraboschi, Avi Mendelson, Dejan Milojicic, Timothy Roscoe, and Robert NM Watson. 2019. Rack-Scale Capabilities: Fine-Grained Protection for Large-Scale Memories. *Computer* 52, 2 (2019), 52–62.

[4] Shuo-Han Chen, Tseng-Yi Chen, Yuan-Hao Chang, Hsin-Wen Wei, and Wei-Kuan Shih. 2018. A Partial Page Cache Strategy for NVRAM-Based Storage Devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2018).

[5] A Micro Devices. 2006. AMD64 architecture programmer's manual volume 2: System programming. (2006).

[6] Izzat El Hajj, Alexander Merritt, Gerd Zellweger, Dejan Milojicic, Reto Achermann, Paolo Faraboschi, Wen-mei Hwu, Timothy Roscoe, and Karsten Schwan. 2016. SpaceJMP: programming with multiple virtual address spaces. *ACM SIGARCH Computer Architecture News* 44, 2 (2016), 353–368.

[7] Simon Gerber, Gerd Zellweger, Reto Achermann, Kornilios Kourtis, Timothy Roscoe, and Dejan Milojicic. 2015. Not your parents' physical address space. In *15th Workshop on Hot Topics in Operating Systems (HotOS {XV})*.

[8] Katsuhiko Hoya, Kosuke Hatsuda, Kenji Tsuchida, Yohji Watanabe, Yusuke Shirota, and Tatsunori Kanai. 2019. A perspective on NVRAM technology for future computing system. In *2019 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA)*. IEEE, 1–2.

[9] Kimberly Keeton. 2017. Memory-Driven Computing.. In *FAST*.

[10] HP Labs. [n. d.]. The Machine. https://www.hpl.hp.com/research/systems-research/themachine/. ([n. d.]). Abgerufen am: 22.12.2019.

[11] Eunji Lee, Hyokyung Bahn, Seunghoon Yoo, and Sam H Noh. 2014. Empirical study of NVM storage: an operating system's perspective and implications. In *2014 IEEE 22nd International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems*. IEEE, 405–410.

[12] Ruicheng Liu, Peiquan Jin, Zhangling Wu, Xiaoliang Wang, Shouhong Wan, and Bei Hua. 2019. Efficient Wear Leveling for PCM/DRAM-Based Hybrid Memory. In *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 1979–1986.

[13] Steven Pelley, Thomas F Wenisch, Brian T Gold, and Bill Bridge. 2013. Storage management in the NVRAM era. *Proceedings of the VLDB Endowment* 7, 2 (2013), 121–132.

[14] Moinuddin K Qureshi, John Karidis, Michele Franceschini, Vijayalakshmi Srinivasan, Luis Lastras, and Bulent Abali. 2009. Enhancing lifetime and security of PCM-based main memory with start-gap wear leveling. In *Proceedings of the 42nd annual IEEE/ACM international symposium on microarchitecture*. ACM, 14–23.

[15] Lingyu Zhu, Zhiguang Chen, Fang Liu, and Nong Xiao. 2018. Wear Leveling for Non-Volatile Memory: a Runtime System Approach. *IEEE Access* 6 (2018), 60622–60634.