

Virtuelle Speicherverwaltung: Freund oder Feind großer Hauptspeicher?

20. Januar 2020

Julian Lehrhuber

Friedrich-Alexander-Universität Erlangen-Nürnberg

Motivation

Hauptspeicher



- steigende Kapazität
- flüchtig → persistent

Hauptspeicher



- steigende Kapazität
- flüchtig → persistent

Virtuelle Speicherverwaltung



- seitenbasiert
- *demand paging*
- ...

Hauptspeicher



- steigende Kapazität
- flüchtig → persistent

Virtuelle Speicherverwaltung



- seitenbasiert
- *demand paging*
- ...

Virtuelle Speicherverwaltung mit großen Hauptspeichern?

Neue Speichertechnologien

Eigenschaften virtueller Speicherverwaltung

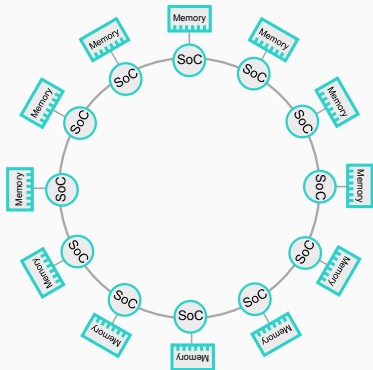
Anwendbarkeit

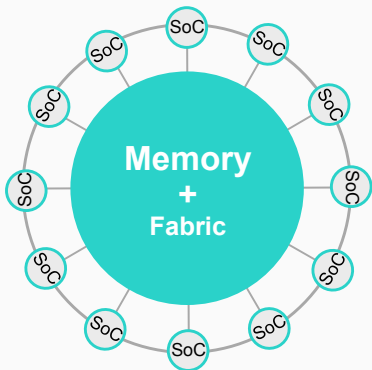
Neue Verwaltungsmethoden

Fazit

Neue Speichertechnologien

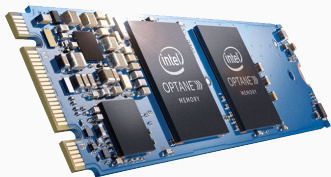
Fabric Attached Memory (FAM) [2]





„The Machine“ (HPE)

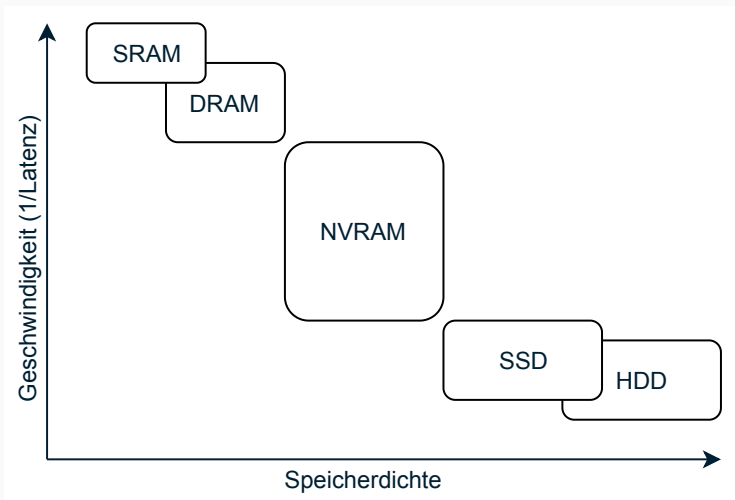
- speicherorientierte Datenverarbeitung
- großer, geteilter Adressraum
- persistenter Speicher



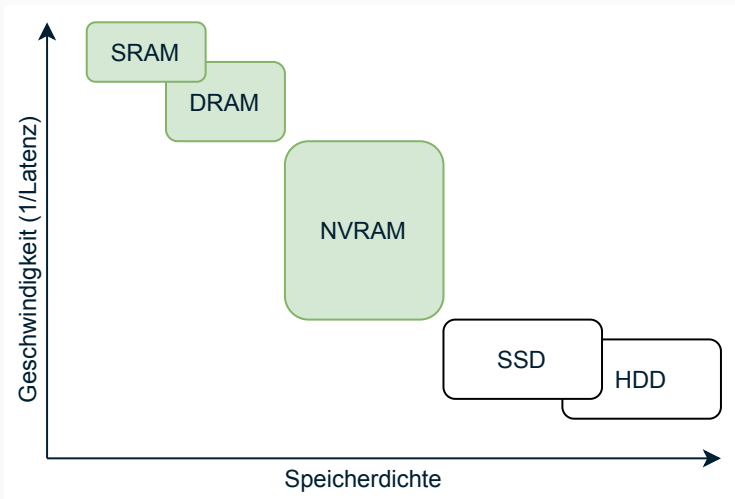
Eigenschaften

- persistent
- energieeffizient
- hohe Speicherdichte
- asymmetrische
Zugriffslatenzen
- kurze Lebensdauer

Einordnung von NVRAM [3]

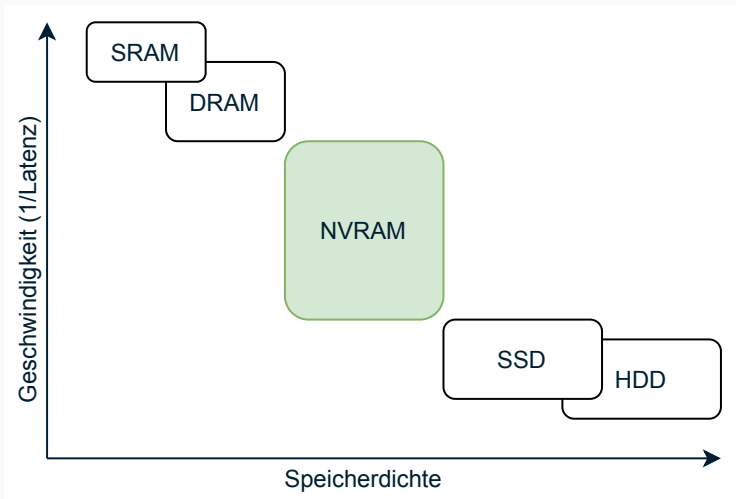


Einordnung von NVRAM [3]



DRAM-Hauptspeicher, NVRAM-Sekundärspeicher

Einordnung von NVRAM [3]



großer NVRAM Hauptspeicher, kein Sekundärspeicher

Eigenschaften virtueller Speicherverwaltung

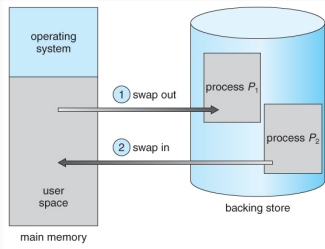


Seitenbasiert

Eigenschaften virtueller Speicherverwaltung



Seitenbasiert

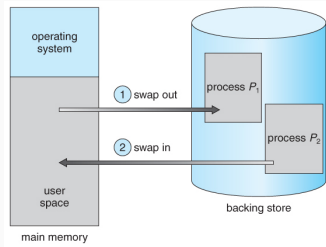


Ein-/Auslagerung

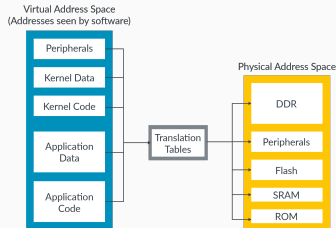
Eigenschaften virtueller Speicherverwaltung



Seitenbasiert



Ein-/Auslagerung

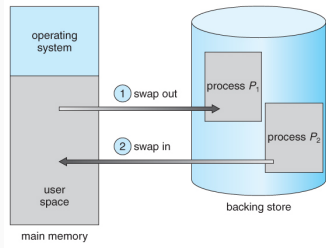


Adressumsetzung

Eigenschaften virtueller Speicherverwaltung

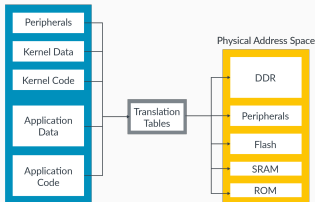


Seitenbasiert

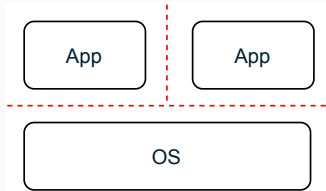


Ein-/Auslagerung

Virtual Address Space
(Addresses seen by software)



Adressumsetzung



Speicherschutz

Anwendbarkeit

Fokus: Latenzen, Speicherkonfiguration

DRAM + NVRAM	NVRAM-only
<i>Demand paging</i> anwendbar, aber sinnvoll?	<i>Demand paging</i> nicht anwendbar

Fokus: Latenzen, Speicherkonfiguration

DRAM + NVRAM	NVRAM-only
<i>Demand paging</i> anwendbar, aber sinnvoll?	<i>Demand paging</i> nicht anwendbar

⇒ NVRAM-Latenzen ähnlich niedrig wie DRAM

⇔ HDD/SSD-Latenzen

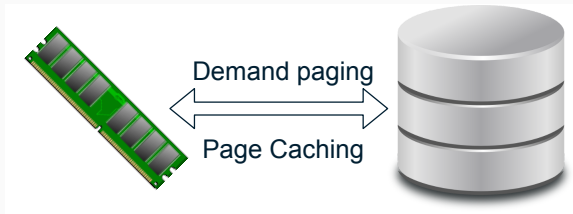
Fokus: Latenzen, Speicherkonfiguration

DRAM + NVRAM	NVRAM-only
<i>Demand paging</i> anwendbar, aber sinnvoll?	<i>Demand paging</i> nicht anwendbar

⇒ NVRAM-Latenzen ähnlich niedrig wie DRAM

⇔ HDD/SSD-Latenzen

⇒ Vergleich mit *page caching* [4]:



⇒ 50-80% weniger Latenzen ohne *page caching*

Fokus: Größe des Hauptspeichers

- 48 Bit großer virtueller Adressraum \Rightarrow 256 TB adressierbar

Fokus: Größe des Hauptspeichers

- 48 Bit großer virtueller Adressraum \Rightarrow 256 TB adressierbar
- Größe der Seitentabelle (bei 2 MB Seitengröße):

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

\Rightarrow 1 GB bei 8 Byte/Eintrag!

\Rightarrow 2 MB Seitengröße \Rightarrow keine grobe Zugriffskontrolle!

Fokus: Größe des Hauptspeichers

- 48 Bit großer virtueller Adressraum \Rightarrow 256 TB adressierbar
- Größe der Seitentabelle (bei 2 MB Seitengröße):

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

\Rightarrow 1 GB bei 8 Byte/Eintrag!

\Rightarrow 2 MB Seitengröße \Rightarrow keine granulare Zugriffskontrolle!

- Größe der Seitentabelle (bei 4 KB Seitengröße):

$$2^{48} \text{ Bytes} \div 2^{12} \text{ Bytes} = 2^{36} \text{ Einträge}$$

\Rightarrow 512 GB bei 8 Byte/Eintrag!

Wie können wir mehr Speicher adressieren?

Mehr Adressbits?

Größere Seiten?

Mehr Adressbits? **Nein, da...**

- Löst das Problem der großen Seitentabelle nicht
- Höhere Kosten der Komponenten [1]
- Erhöhte *TLB-miss* Latenz [5]

Größere Seiten?

Mehr Adressbits? Nein, da...

- Löst das Problem der großen Seitentabelle nicht
- Höhere Kosten der Komponenten [1]
- Erhöhte *TLB-miss* Latenz [5]

Größere Seiten? Nein, da...

- Kein granularer Speicherschutz mehr
- Sicherheitsrisiken für Speicherverwaltung [6]

Fokus: Speicherkonfiguration & -charakteristiken

xRAM	SSDs/HDDs
Byteweise	Blockweise

Fokus: Speicherkonfiguration & -charakteristiken

xRAM	SSDs/HDDs
Byteweise	Blockweise

Wo kommt die Seite ins Spiel?

Fokus: Speicherkonfiguration & -charakteristiken

xRAM	SSDs/HDDs
Byteweise	Blockweise

Wo kommt die Seite ins Spiel?

- Granularität des Speicherschutzes [7]
- 1 Block = x Seiten \Rightarrow Kommunikation mit Blockgeräten/Sekundärspeichern

Fokus: Speicherkonfiguration & -charakteristiken

xRAM	SSDs/HDDs
Byteweise	Blockweise

Wo kommt die Seite ins Spiel?

- Granularität des Speicherschutzes [7]
- 1 Block = x Seiten \Rightarrow Kommunikation mit Blockgeräten/Sekundärspeichern

Und ohne Sekundärspeichern? Wie groß sollte eine Seite sein?
 \Rightarrow unklar! [7]

Allerdings: NVRAM benötigt zur Verlängerung der Lebensdauer derzeit einen *wear-leveling* Mechanismus! [8, 9]

Allerdings: NVRAM benötigt zur Verlängerung der Lebensdauer derzeit einen *wear-leveling* Mechanismus! [8, 9]

In Hardware?

In Software?

Allerdings: NVRAM benötigt zur Verlängerung der Lebensdauer derzeit einen *wear-leveling* Mechanismus! [8, 9]

In Hardware? Mögl. schwierig, da Anforderungen an Hardware zum Erhalt der Metadaten über Abnutzungsgrad der Speicherzellen wachsen [10]

In Software?

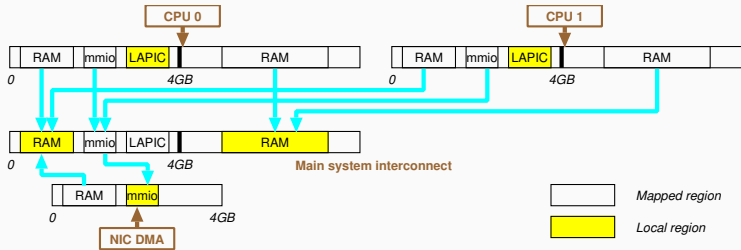
Allerdings: NVRAM benötigt zur Verlängerung der Lebensdauer derzeit einen *wear-leveling* Mechanismus! [8, 9]

In Hardware? Mögl. schwierig, da Anforderungen an Hardware zum Erhalt der Metadaten über Abnutzungsgrad der Speicherzellen wachsen [10]

In Software? Chance für virtuelle Speicherverwaltung, *wear-leveling* über geeignete Adressumsetzung zu realisieren.

Neue Verwaltungsmethoden

Neustrukturierung physischen Speichers (Gerber et. al [5])



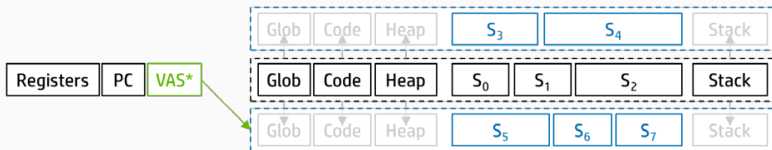
Adressräume nach Gerber et. al¹

- Nutzt *Hardwarefähigkeiten* (Speicherzugriffskontrolle via Berechtigungsschlüssel)
- **Baumartige Struktur von Adressräumen**
- Einfacher Speicherschutz
- Verhindert Adress-Loops

¹...in einem vereinfachten Zweikern-Rechensystem mit einer 32-Bit PCI Netzwerkkarte mit DMA [5]

SpaceJMP (El Hajj et. al [1])

New Process Abstraction: {PC, registers, **VAS***, {VAS}}



Virtuelle Adressräume nach El Hajj et. al [1]

- Definiert einen virt. Adressraum als Menge von Segmenten
- Jeder virt. Adressraum hat eine Nummer
- Virt. Adressräume sind von Anwendung kontrollier-/austauschbar
⇒ Mehr Speicher adressierbar
- geeignet für FAM-Systeme [2]

Fazit

Die Aufgaben von virtueller Speicherverwaltung sind stark Abhängig von der gewählten Speichertechnologie und -architektur.

Veränderung	Fragestellung
Kein Sekundärspeicher	Seitengröße? <i>demand paging</i> ?
NVRAM Sekundärspeicher	<i>demand paging</i> ? <i>wear-leveling</i> ?
Großer Hauptspeicher	Adressumsetzung? Speicherschutz?

Ist virtuelle Speicherverwaltung hinsichtlich großer Hauptspeicher und neuer Speichertechnologien weiterhin relevant?

Ist virtuelle Speicherverwaltung hinsichtlich großer Hauptspeicher und neuer Speichertechnologien weiterhin relevant?

⇒ Ja, denn Segment sowie Hardwarefähigkeiten-basierte Speicherverwaltungsansätze versprechen eine einfachere Verwaltung von großen Hauptspeichern und bieten mit neuen Speichertechnologien wie NVRAM neue Chancen für virtuelle Speicherverwaltung.

Danke für eure Aufmerksamkeit!



Izzat El Hajj, Alexander Merritt, Gerd Zellweger, Dejan Milojicic, Reto Achermann, Paolo Faraboschi, Wen-mei Hwu, Timothy Roscoe, and Karsten Schwan.

Spacejump: programming with multiple virtual address spaces.

ACM SIGARCH Computer Architecture News, 44(2):353–368, 2016.



Kimberly Keeton.

Memory-driven computing.

In *FAST*, 2017.



Katsuhiko Hoya, Kosuke Hatsuda, Kenji Tsuchida, Yohji Watanabe, Yusuke Shirota, and Tatsunori Kanai.

A perspective on nvram technology for future computing system.

In 2019 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA), pages 1–2. IEEE, 2019.



Shuo-Han Chen, Tseng-Yi Chen, Yuan-Hao Chang, Hsin-Wen Wei, and Wei-Kuan Shih.

A partial page cache strategy for nvram-based storage devices.

IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018.



Simon Gerber, Gerd Zellweger, Reto Achermann, Kornilios Kourtis, Timothy Roscoe, and Dejan Milojevic.

Not your parents' physical address space.

In *15th Workshop on Hot Topics in Operating Systems (HotOS {XV})*, 2015.



Kirk M Bresniker, Paolo Faraboschi, Avi Mendelson, Dejan Milojevic, Timothy Roscoe, and Robert NM Watson.

Rack-scale capabilities: Fine-grained protection for large-scale memories.

Computer, 52(2):52–62, 2019.



Katelin Bailey, Luis Ceze, Steven D Gribble, and Henry M Levy.

Operating system implications of fast, cheap, non-volatile memory.

In *HotOS*, volume 13, pages 2–2, 2011.



Ruicheng Liu, Peiquan Jin, Zhangling Wu, Xiaoliang Wang, Shouhong Wan, and Bei Hua.

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), pages 1979–1986. IEEE, 2019.



Lingyu Zhu, Zhiguang Chen, Fang Liu, and Nong Xiao.
Wear leveling for non-volatile memory: a runtime system approach.

IEEE Access, 6:60622–60634, 2018.



Moinuddin K Qureshi, John Karidis, Michele Franceschini, Vijayalakshmi Srinivasan, Luis Lastras, and Bulent Abali.
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, pages 14–23. ACM, 2009.