Simon Alexander Oelgeschläger

Non-volatile main memory (NVMM) Speichersysteme

Eine Präsentation im Seminar:
Effiziente Programmierung

17.12.2019

Aufbau:

- Einleitung
- Überblick Flüchtige / Nichtflüchtige Speichertechnologien
- Änderung in Architektur und Software bei Nutzung von NVM als Hauptspeicher
- Zusammenfassung und Aussicht in die Zukunft

Dateisysteme

- Entwicklung ursprünglich für lokale Nutzung auf einzelnen Computern
- Trend zu Hochleistungsrechnern aus vielen Computern in einem Netzwerk mit parallelem, verteiltem Dateisystem
- Definitionen:

Google Wörterbuch

```
1 Computerprogramm, das als Bestandteil des Betriebssystems
2 das Speichern, Lesen und Löschen von Dateien
3 auf einem Datenträger organisiert.
```

https://www.itwissen.info/Dateisystem-file-system-FS.html

```
1 Das Dateisystem ist Bestandteil des Betriebssystems
2 und bildet die Schnittstelle zwischen diesem und den Laufwerken.
3 Es legt fest, wie der Computer Dateien auf den Datenträgern benennt,
4 speichert, organisiert und verwaltet.
5 Ein Dateisystem besteht aus Dateien, Verzeichnissen und Adressen,
6 über die die Dateien lokalisiert werden.
```

Dezentrale Systeme – Probleme ?

- Wie sind Konsistenzbedingungen definiert?
- Wie werden Locks sichergestellt?
- Jedes Lesen auch ein Schreiben durch Aktualisierung der Zeitstempel?

Zwischenwurf

- POSIX
- Journaling
- Key-Value-Stores

POSIX – *(Portable Operating System Interface)*

- Unix Standard für APIs – Schnittstelle zwischen Anwendungssoftware und Betriebssystem
- Erste Veröffentlichung 1988
    - Regelmäßige Revisionen – Nur kleine Änderungen
- Hauptziel: Portabilität von Anwendungs-Sourcecode
- Entkopplung von Anwendungssoftware und Betriebssystem bei Datenmanagement
- POSIX nicht sonderlich gut für verteilte Systeme geeignet

POSIX – Heutiger Stand

- Anwendungen werden nicht mehr nach standardisierten POSIX Schnittstellen gebaut
- Nutzung von plattformabhängigen Libraries und Frameworks
- Aufruf von High-Level Frameworks
- Low-Level Frameworks am Ende der Kette nutzen möglicherweise POSIX Libraries
- Neue Entwicklungen konvergieren nicht zu einem Standard: gehen auseinander

POSIX – Probleme

- Fragmentierung von POSIX Implementierungen bei verschiedenen UNIX Betriebssystemen
- Kein Standard Interface für GPUs
- Erfolg von OpenGL
- ioctl (i/o-control) wird genutzt, aber eigentlich nicht dafür gedacht (eher z.B. für CD Laufwerk)
- Inter-Process Communication (IPC) auf allen Plattformen unterschiedlich entwickelt – immer weiter als POSIX Standard

POSIX – Implementierungen

- Vergleich von IPC (Inter-process communication) im POSIX-Stil und modernen Implementierungen

- Beispiel Binder (eingesetzt in Android):
  - File Descriptor kann an mehrere Prozesse übergeben werden
  - Multithread Model: Prozess kann mehrere simultane Anfragen bearbeiten
  - Schnelle Single- und Zero-Copy Mechanismen
  - Senden von Nachrichten in nahezu konstanter Zeit

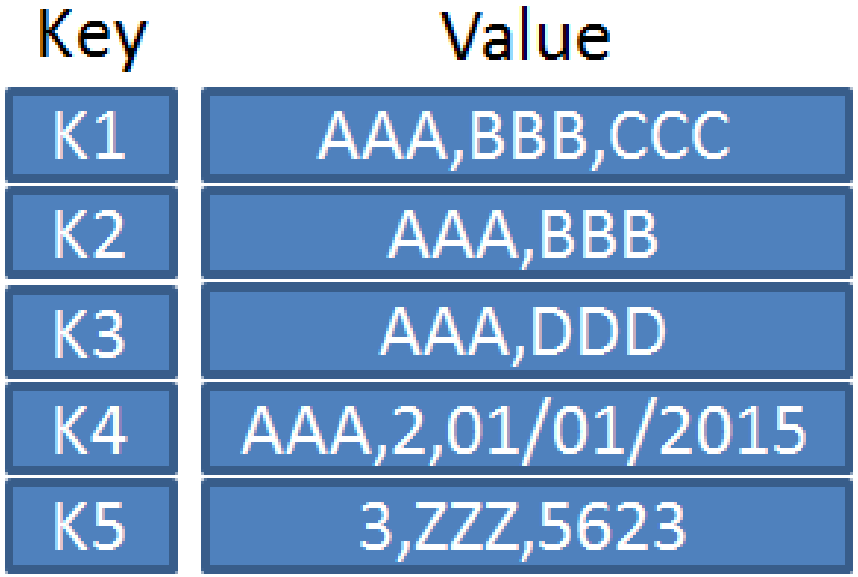
Tx/Rx–Android	UNIX avg (µs)	Binder avg (µs) |
32 bytes	54	115
128 bytes	56	114
1 page	73	93
10 pages	276	93
100 pages	1898	94

Journaling

- Informationen über Operationen werden in ein Journal geschrieben – Idee wie das strikte Führen eines Tagebuches
- Qualität und Quantität der Informationen kann variieren
- Manchmal nur Metadaten, allerdings im Extremfall auch alle Daten ins Journal schreiben
- Ziel: Konsistenzsicherung und Schadensminimierung bei Systemausfällen

Key-value-store

- Auch bekannt als ‚*dictionary*' oder ‚*hash table*'
- Alternative zu SQL: NoSQL, also keine relationale Datenbank
- Ein Paar aus Schlüssel und Wert
- Wert kann theoretisch beliebige Form annehmen
- Sehr flexible und effiziente Art der Datenspeicherung
- Lange Zeit schlechte Performance und wenig Standardisierung
  Durchsuchen der Values nur durch einzelnes Auslesen!



https://en.wikipedia.org/wiki/Key-value_database#/media/File:KeyValue.PNG

Flüchtige & Nichtflüchtige Speichertechnologien

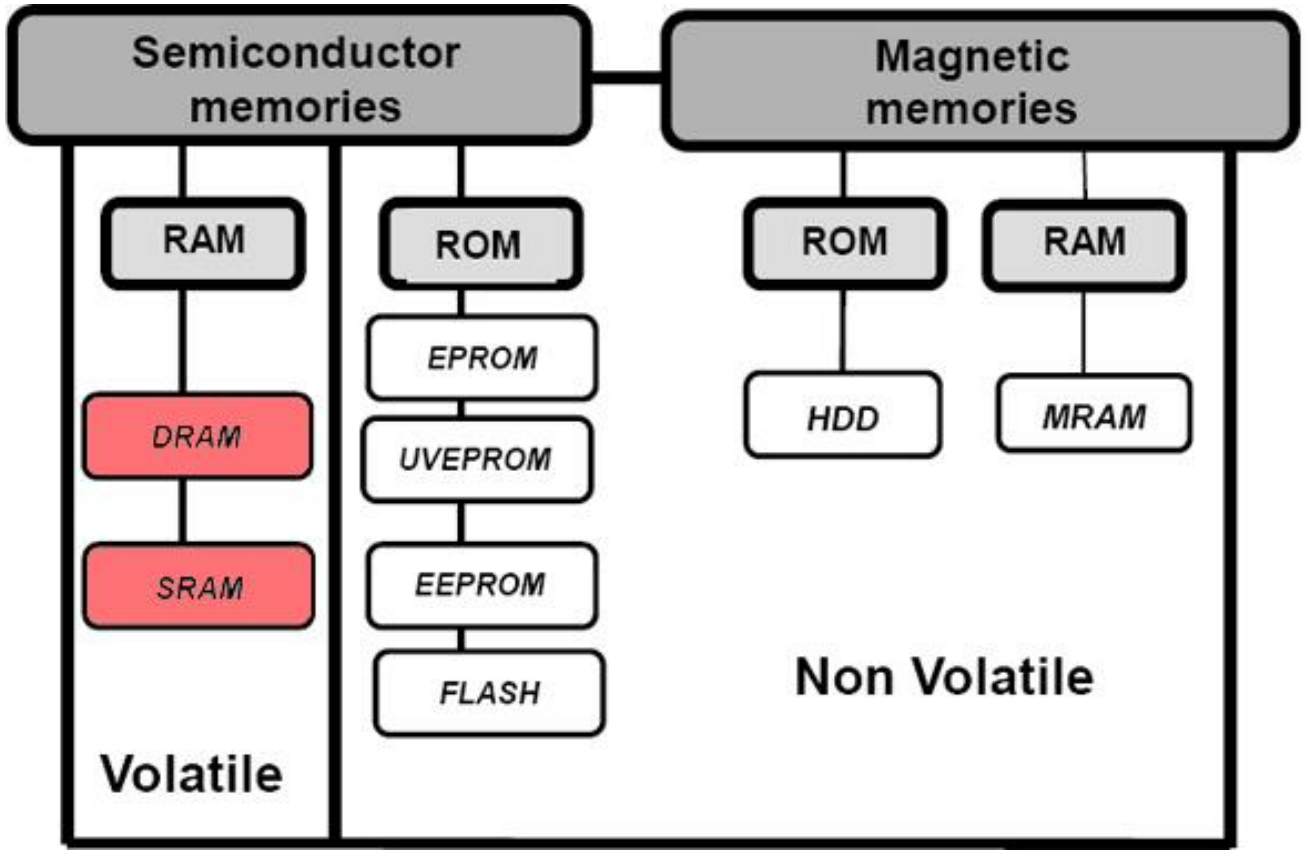
- Theorie und Hintergrund
- DRAM gegen NVMM
- Stand der Forschung
  - Fe-RAM, M-RAM, …

Speichertechnologien

Jegliche Art von Medien, die jegliche Art von Daten speichert (Computer)

Datenspeicher für elektronische Geräte:

- Elektronische Speicherung
  Halbleiterspeicher
- andere Datenträger
  Festplatten – magnetisch
  DVDs – optisch
  …



https://www.book-ebooks.com/products/reading-epub/product-id/2998977/title/Non-volatile%2BMemories.html?autr=%22Jean-Claude+Lacroix%22

Vergleich von ‚volatile' & ‚non-volatile' memory

- volatile memory (flüchtig)
  - Daten bleiben nur bei konstanter Stromversorgung erhalten
  - Kurze Zugriffszeit
  - Höherer Preis pro Speichereinheit
  - *zB. SRAM / DRAM*
- non-volatile memory (nichtflüchtig)
  - Speicherung von Daten auch ohne anliegende Spannung
  - Längere Zugriffszeit
  - Niedriger Preis pro Speichereinheit
  - *zB. HDD Festplatten, SSDs*

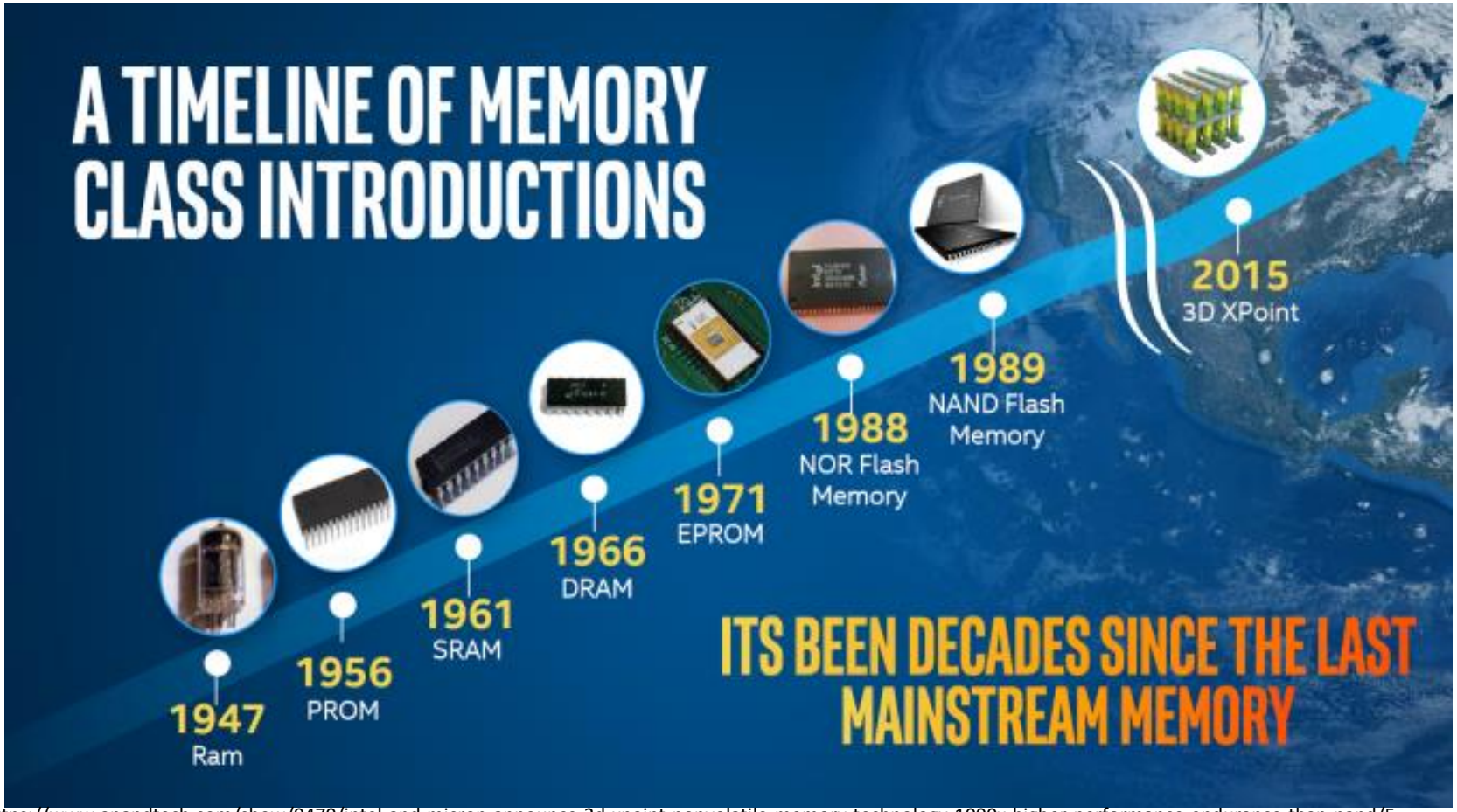
Übersicht über einige Speichertypen

Typ	Kategorie	Löschen	byte-adressierbar	flüchtig	Typische Anwendung
SRAM	Lesen/Schreiben	elektrisch	ja	ja	Level-2 Cache
DRAM	Lesen/Schreiben	elektrisch	ja	ja	Hauptspeicher (alt)
SDRAM	Lesen/Schreiben	elektrisch	ja	ja	Hauptspeicher
ROM	nur Lesen	—	nein	nein	Geräte in großen Stückzahlen
PROM	nur Lesen	—	nein	nein	Geräte in kleinen Stückzahlen
EPROM	vorw. Lesen	UV-Licht	nein	nein	Prototypen
EEPROM	vorw. Lesen	elektrisch	ja	nein	Prototypen
Flash	Lesen/Schreiben	elektrisch	nein	nein	Speicherkarten, Mobile Geräte, SSDs

aus: Rechnerstrukturen und Betriebssysteme von Dr. Andreas Mäder (S. 748)

Rückblick – Entwicklung von Speichertechnologien

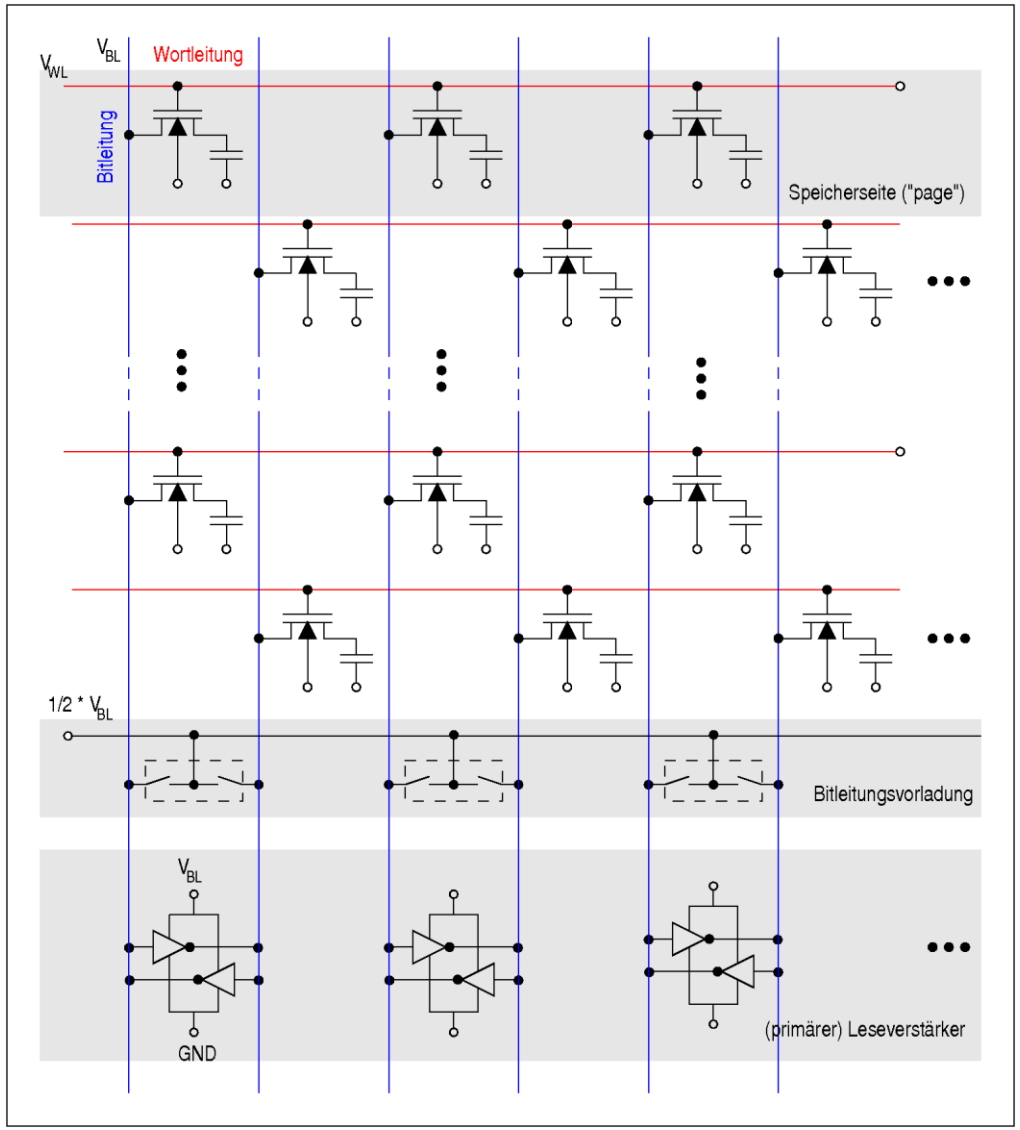
- Heutiger Hauptspeicher ist nahezu ausnahmslos DRAM
- Stetige Weiterentwicklung, aber selbe physikalische Bauweise
- NVMM – Umbruch in eine neue Zeit?



https://www.anandtech.com/show/9470/intel-and-micron-announce-3d-xpoint-nonvolatile-memory-technology-1000x-higher-performance-endurance-than-nand/5

DRAM: Funktionsweise

- Einzelne Speicherzelle kann 1 Bit speichern
- Zelle besteht aus 1 Transistor und 1 Kondensator
- Gitterförmige Anordnung der Zellen
- Zugriff:
  1. Aktivierung der Wortleitung
  2. Auslesen der „Zeile“ in alle Verstärker
  3. Weiterleiten der angefragten Information
  4. Zurückschreiben der Zeile



https://commons.wikimedia.org/wiki/File:DRAM_Zellenfeld.png

Latenzen und Zugriffszeiten verschiedener Technologien

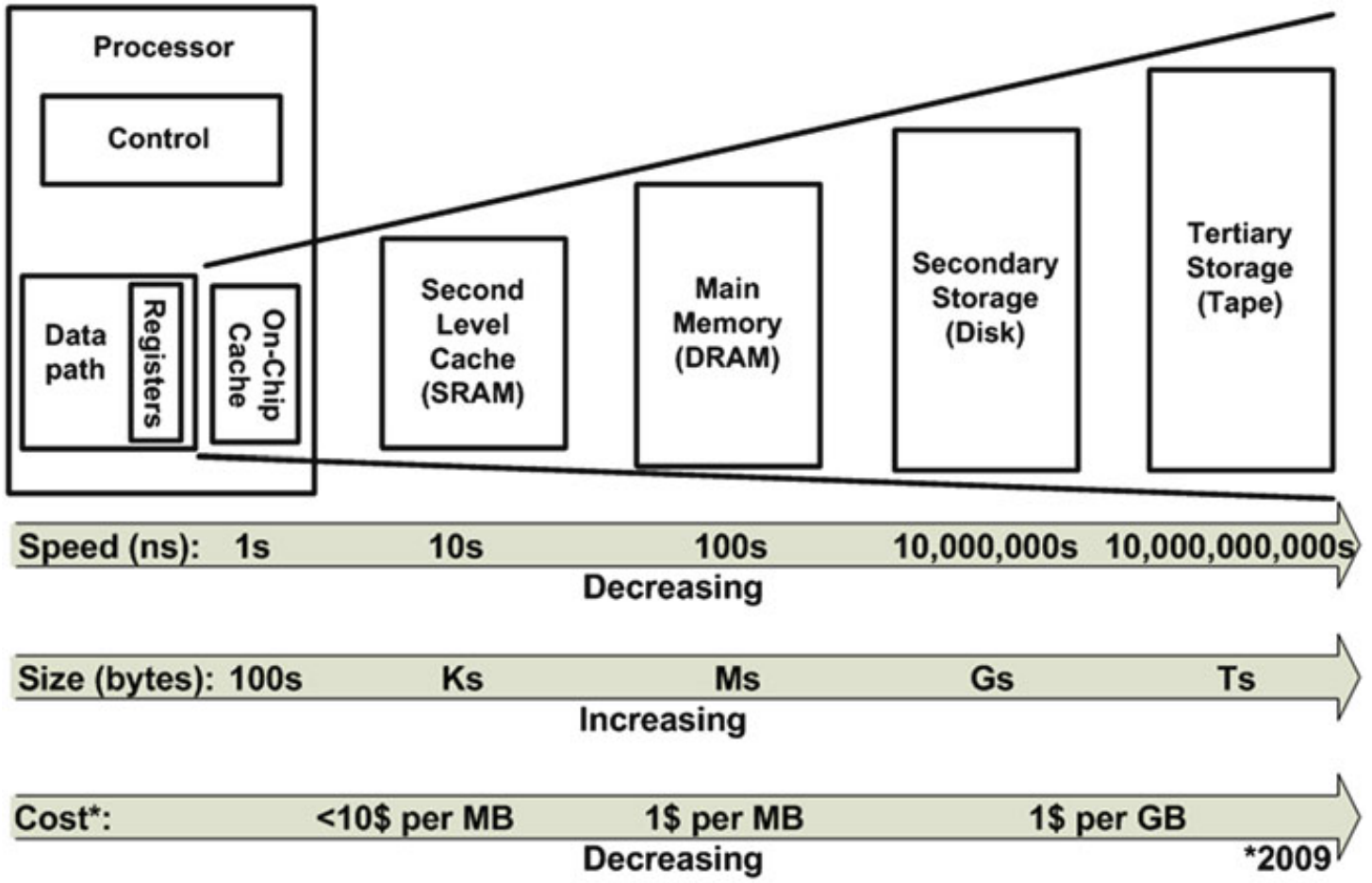
- Verschiedene Einsatzgebiete der Speichertypen bedingt durch stark variierende Eigenschaften
- Bei einem Arbeitsspeicher sollte die Zeit für einen Lese- oder Schreibvorgang < 10ns sein
- SRAM zwar deutlich schneller, aber sehr teuer und viel geringere Speicherdichte
- Flash-Speicher um Größenordnungen zu langsam, sonst aber sehr kompakt und energiesparend

	SRAM	DRAM	Flash (NAND)	HDD
Reciprocal density (F^2)	140	6-8	1-4	2/3
Energy per bit (pJ)	0.0005	0.005	0.00002	$5x10^3$- 10^4
Read time (ns)	0.1-0.3	6	100 000	$5\text{-}8x10^6$
Write time (ns)	0.1-0.3	6	100 000	$5\text{-}8x10^6$
Retention	as long as V applied	<<second	years	years
Endurance (cycles)	$> 10^{16}$	$> 10^{16}$	10^5	10^4

https://www.book-ebooks.com/products/reading-epub/product-id/2998977/title/Non-volatile%2BMemories.html?autr=%22Jean-Claude+Lacroix%22

Speicherhierarchie

- Aktuell wird eine Vielzahl verschiedener Speicher verwendet
- In einer utopischen Welt, gäbe es einen einzigen *persistenten*, beliebig *schnellen*, *günstigen* Speicher



Jawar Singh, Saraju P. Mohanty, Dhiraj K. Pradhan; Robust SRAM Designs and Analysis

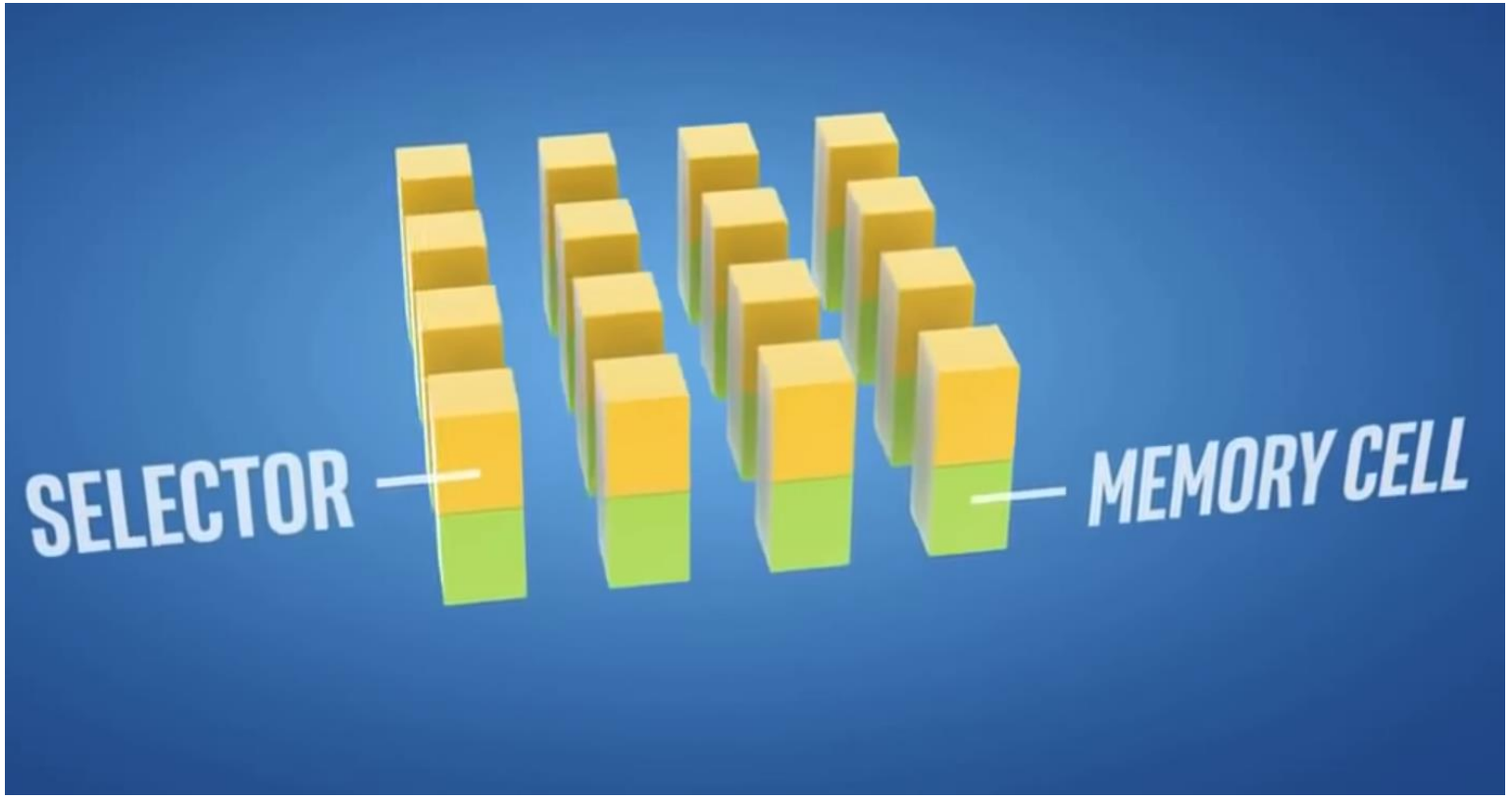
Intel 3D XPoint

https://www.mindfactory.de/product_info.php/960GB-Intel-Optane-905P-Add-In-PCIe-3-0-x4-3D-XPoint--SSDPED1D960GAX1-_1255326.html

- NVM-Technologie von Intel und Micron
  Verkauft u.a. als Intel Optane
- Mit passenden Chipsets wie Intels Cascade Lake auch als Caching Disk nutzbar – theoretisch auch als NVRAM
- Technische Details nicht veröffentlicht, aber vermutlich Unterart von ReRAM (Resistive random-access memory)
- Basiert auf Änderungen des Widerstands im Material
- Als SSD Zugriffszeiten im Mikrosekundenbereich
- Als Hauptspeicher mit Zugriffszeiten im Nanosekundenbereich, erfordert spezielle Hardware (erste ab 2019)

Intel 3D XPoint

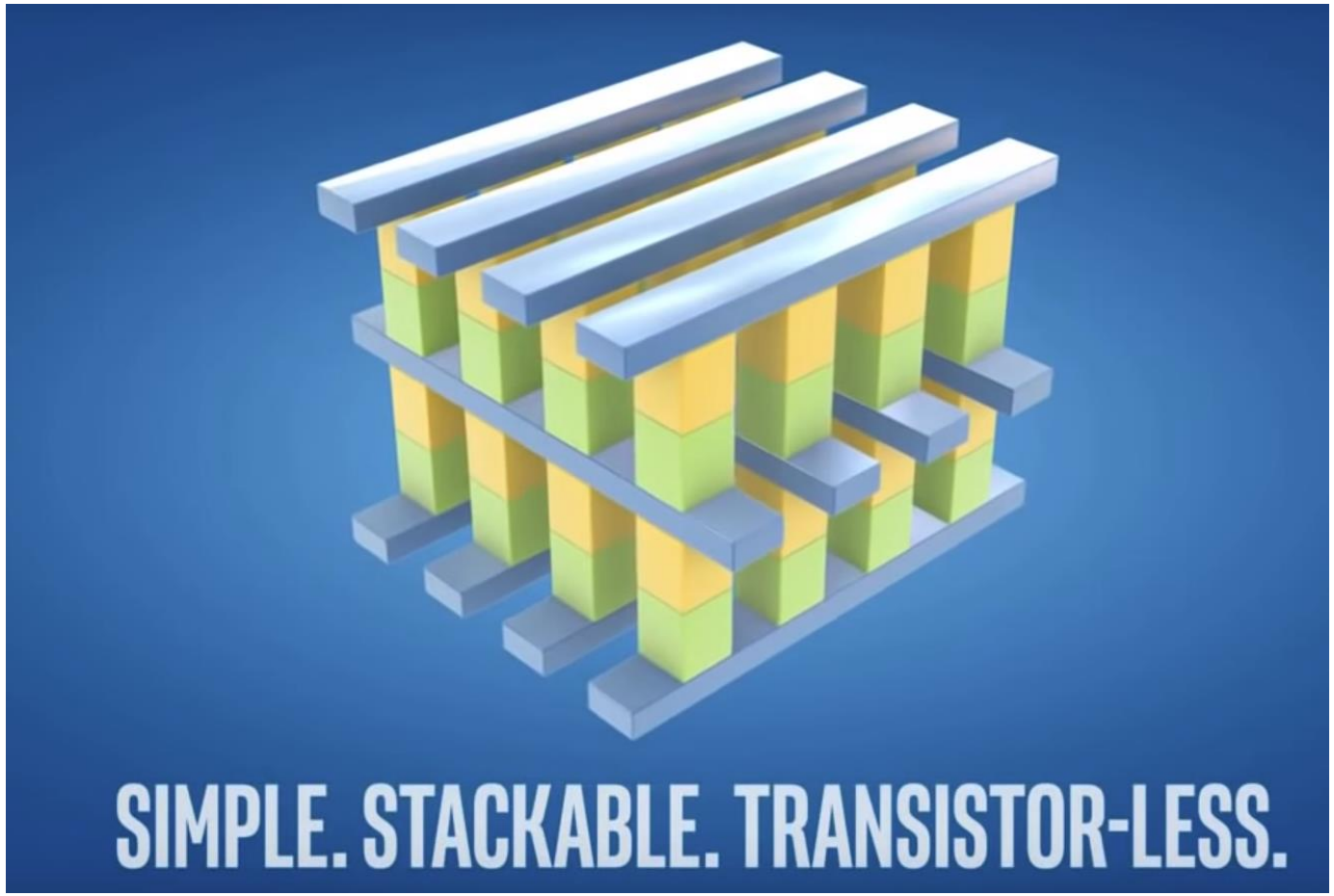
- Eine Speicherzelle mit jeweils einem Selektor
- Detaillierter Aufbau nicht exakt bekannt



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

Intel 3D XPoint

- Struktur erinnert an DRAM
- 3-Dimensionale Gitterstruktur
- Namensgebung analog dazu
- Höhere Speicherdichte als DRAM

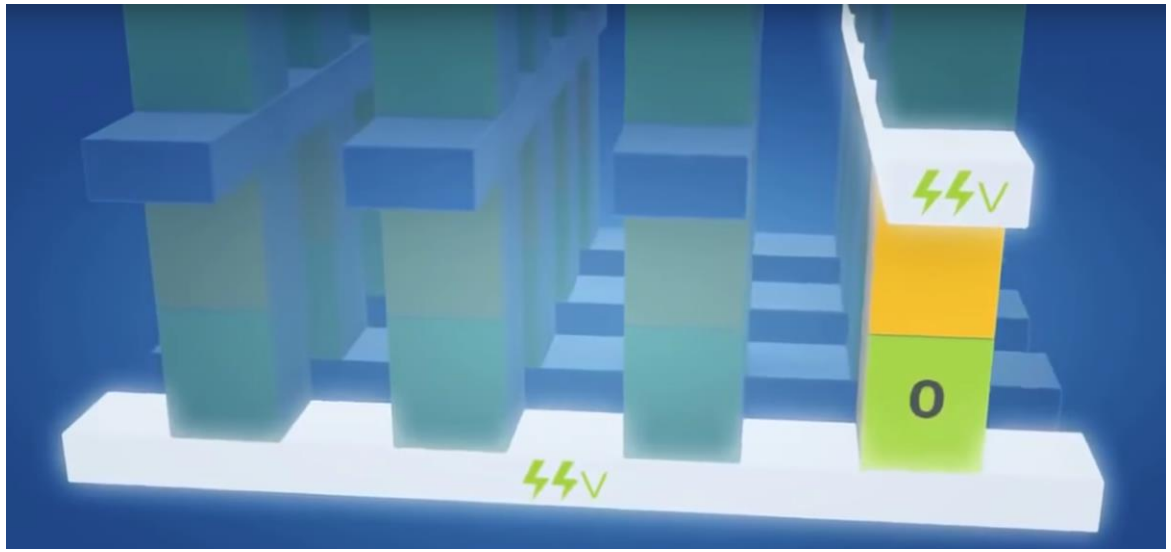


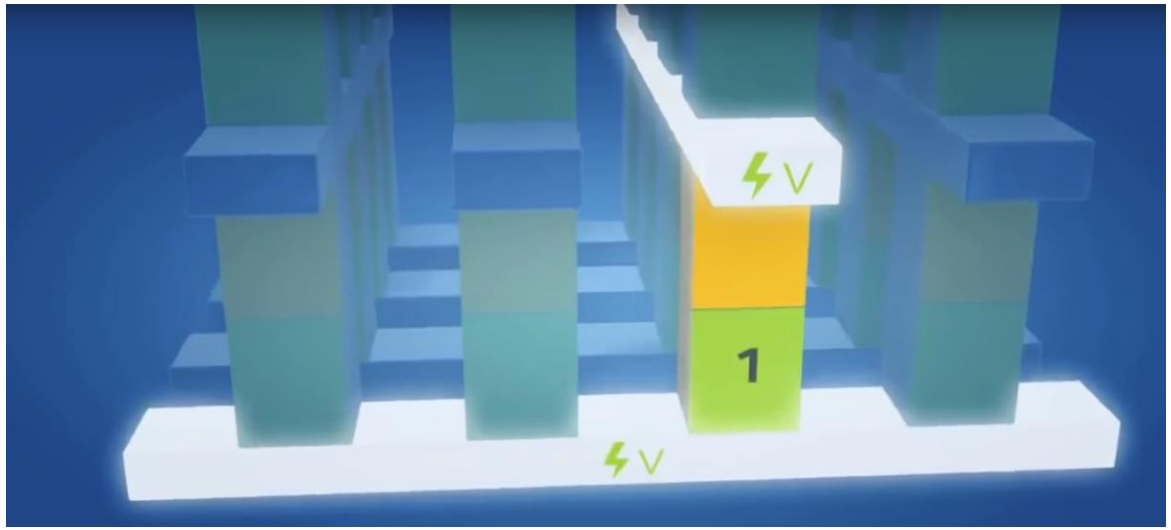
https://www.intel.de/content/www/de/de/architecture-and-technology/optane-technology-animation.html

Intel 3D XPoint

- Schreib- und Lesevorgang durch Anlegen einer definierten Spannung am jeweiligen Selektor

- Kein Transistor notwendig

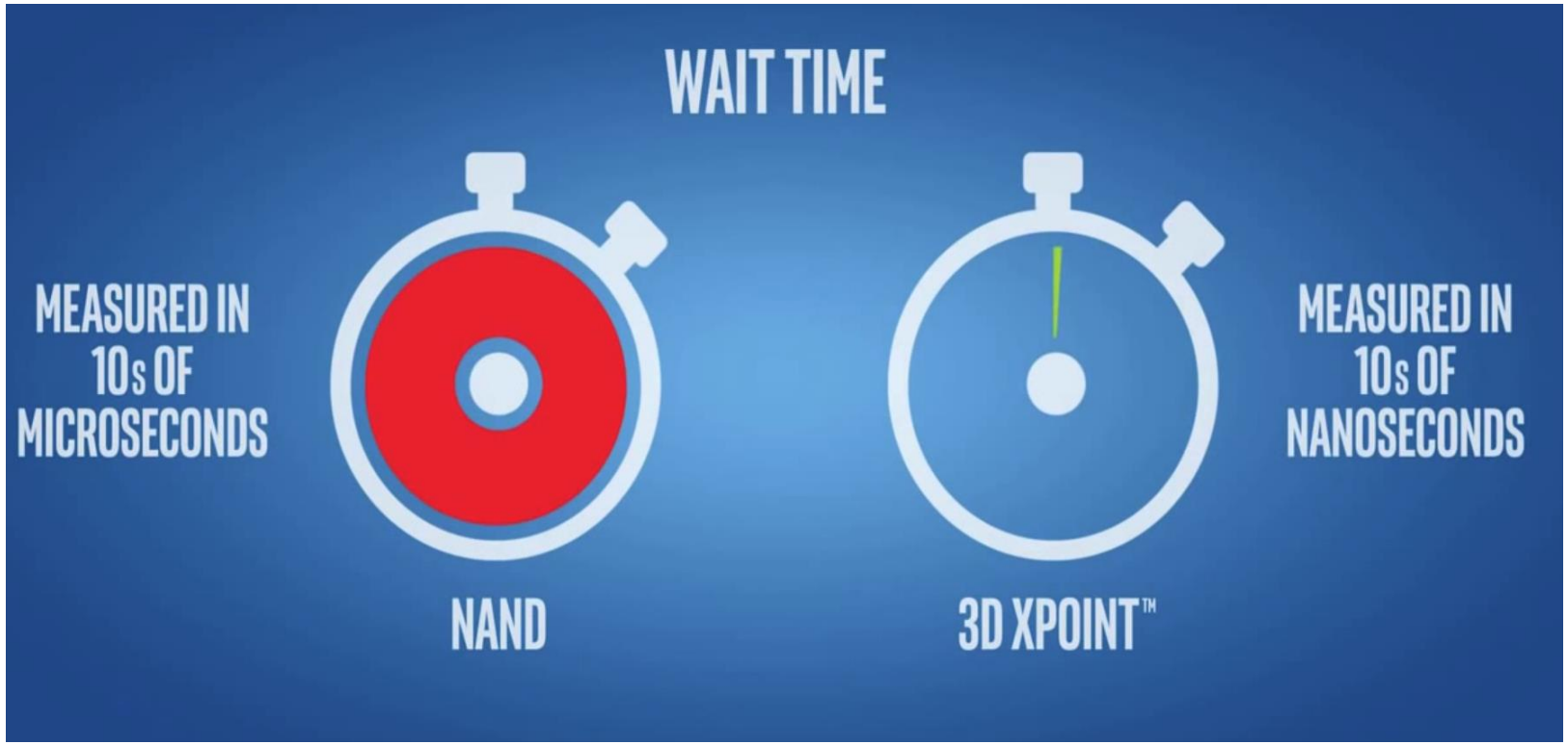




https://www.intel.de/content/www/de/de/architecture-and-technology/optane-technology-animation.html

Intel 3D XPoint

- Werbeaussagen, reale Zeiten abhängig von restlicher Hardware
- Zeiten < 10ns wahrscheinlich bald erreichbar



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

Alternativen in Entwicklung

- **Fe-RAM** *(Ferroelectric Random Access Memory)*
  - nutzt ferromagnetische Eigenschaften
  - Destructive Read – Daten müssen nach dem Lesen zurückgeschrieben werden
  - theoretisch schneller als DRAM, noch in früher Entwicklung

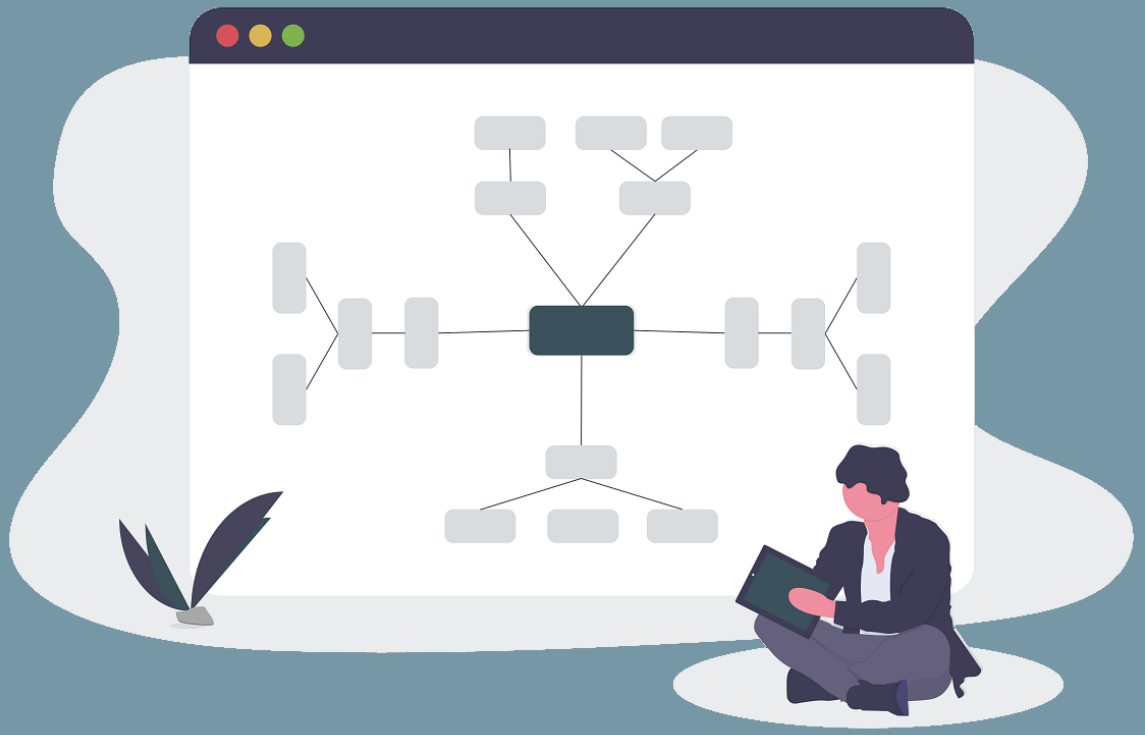
- **PCM** *(Phase-change Random Access Memory)*
  - Höhere Speicherdichte als DRAM
  - Nutzung von amorphen / kristallinen Strukturen
  - Miniaturisierung macht noch größere Probleme

- **STT-RAM** *(Spin-transfer torque Random Access Memory)*
  - Geringere Latenz als DRAM, aber größere Zellfläche
  - Möglicher Einsatz als on-chip Cache in der Zukunft

- **M-RAM** *(Magnetoresistive Random Access Memory)*
  - bisher teuer und geringe Speicherdichte
  - Nischenprodukt

- **Re-RAM** *(Resistive random-access memory)*
  - Höhere Speicherdichte als DRAM
  - Möglicher Einsatz als Cache

Änderungen in Systemarchitektur und Software bei Nutzung von NVM als Hauptspeicher



Motivation für NVMM

- Trennung in Haupt- und Massenspeicher erfordert sehr zeitaufwendiges Kopieren von Daten im Programmablauf
- Unterschiedliche Speichersysteme müssen einzeln über Bussysteme angebunden werden und laufen mit verschiedenen Taktraten

- Daten im NVMM müssen nicht mehr kopiert werden
- NVMM ist direkt am CPU Memory Bus angebunden
- Konsistente Daten benötigen viel weniger, aber andere Formen von Logging / Journaling
- NVMM hat deutlich geringere Latenzen als z.B. Flash und unterstützt viele parallele Zugriffe

Nachteile der bisherigen Architektur

- Daten werden per Paging angefragt und zugewiesen
- Prozess ist abhängig von sehr langsamen externen Speichermedien
- Geringe Latenz von NVMM lässt effiziente Programmierung sehr relevant werden, bisher oft vernachlässigt

Non-volatile *main* memory – NVMM

- Erlaubt deutlich schnellere Dateizugriffe ohne jegliche Anpassung von Code
- Erlaubt Direct Access (DAX) mmap – Mapping der Pages in den User Space, kein Kopieren mehr
- Probleme bei Konsistenzbedingungen zwischen Daten im RAM und Daten auf der Festplatte verschwinden
- Bei NVMM sind Memory Leaks und Data Corruption persistent!
  Rebooting hilft nicht

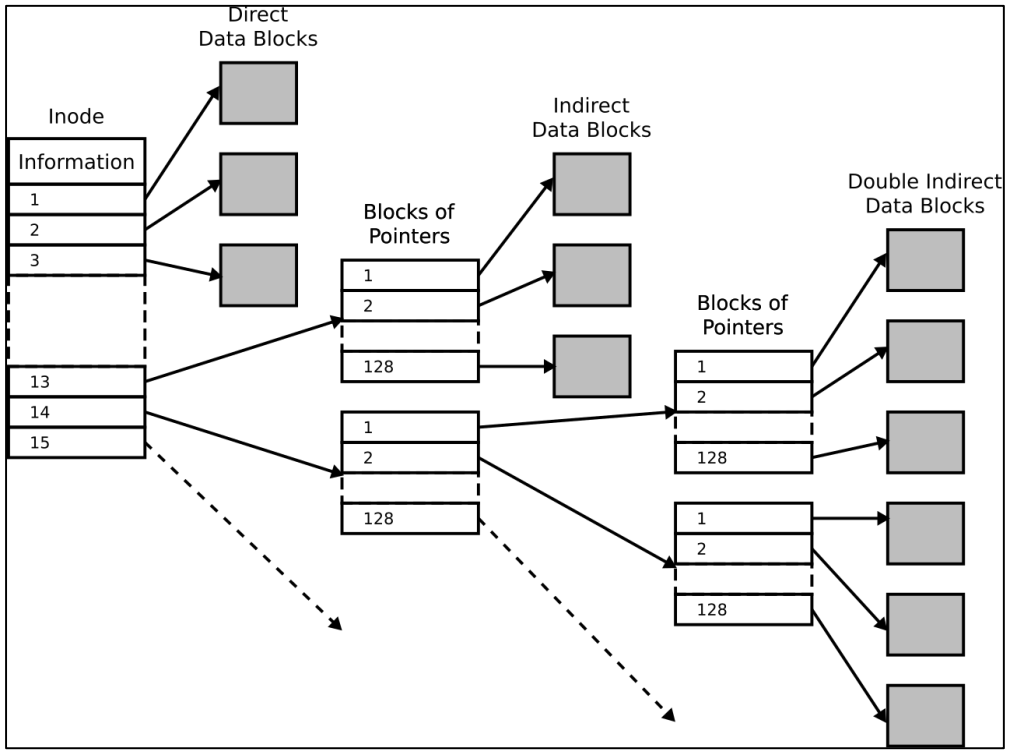
Optimierungen des VFS

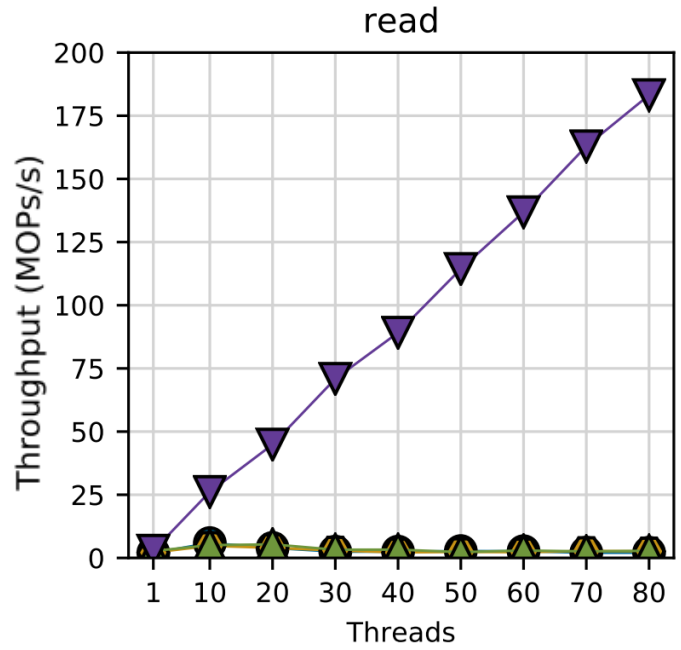
- Inode-Locking
- NUMA-awareness

Skalierbarkeit existierender Virtual File Systems (VFS)

Inode-locks

- Globale Inode-Struktur & per-Inode-locking vernichten die Vorteile von Mehrprozessor-Systemen
- Per-CPU Inode-Struktur erlaubt feinere Locks pro log-Zeile



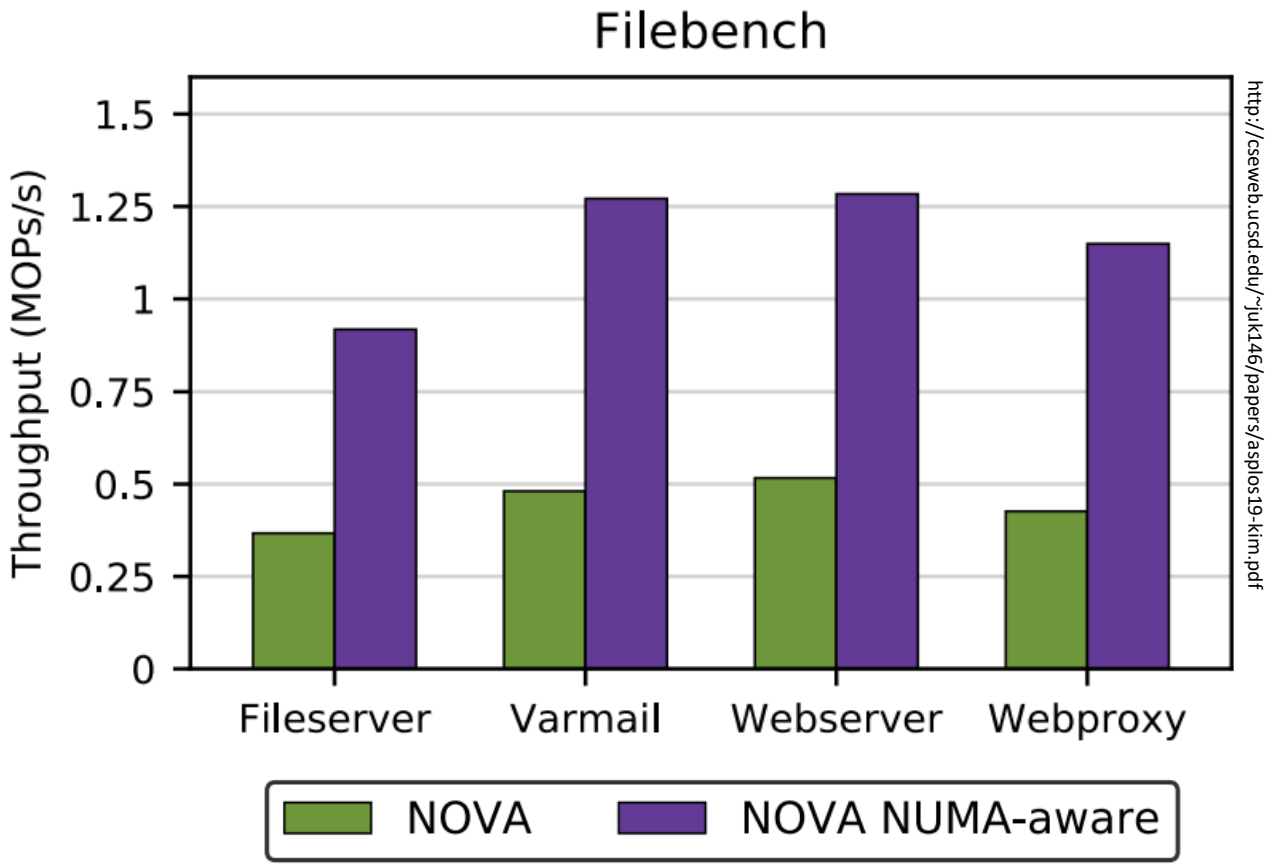


http://cseweb.ucsd.edu/~juk146/papers/asplos19-kim.pdf

NUMA – Non-Uniform Memory Architecture

- Symmetrisches Multiprozessor-System hat Bussystem als limitierenden Faktor
  - Geteilte Adressräume für Prozessoren
  - CPU Hopping kann auftreten

- Entgegenwirken durch NUMA
  - “Eigener” lokaler Arbeitsspeicher pro Prozessor, aber geteilter Adressraum für direkten Zugriff anderer Prozessoren

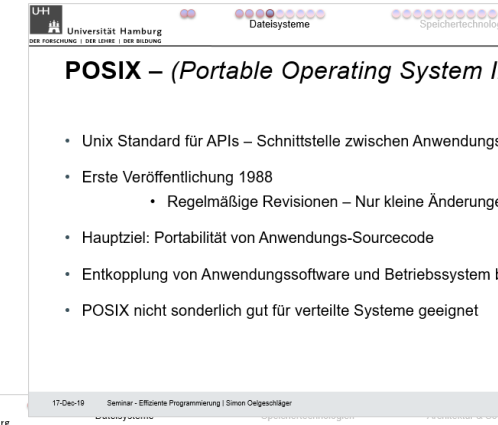


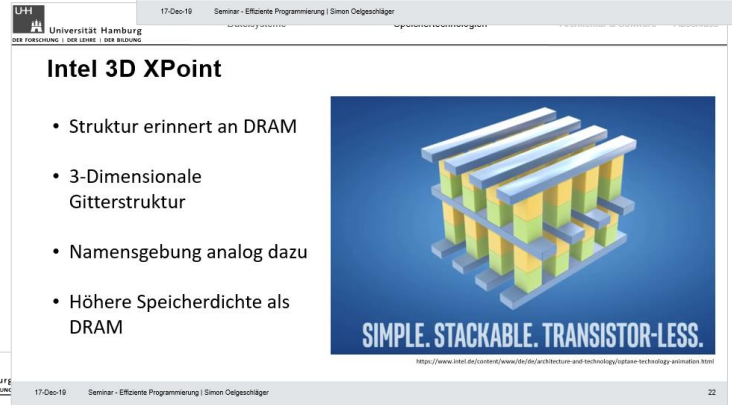
Ausblick & Zusammenfassung

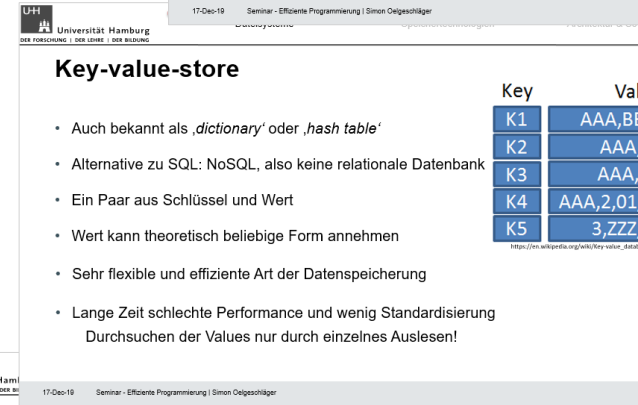
<< What to remember >>

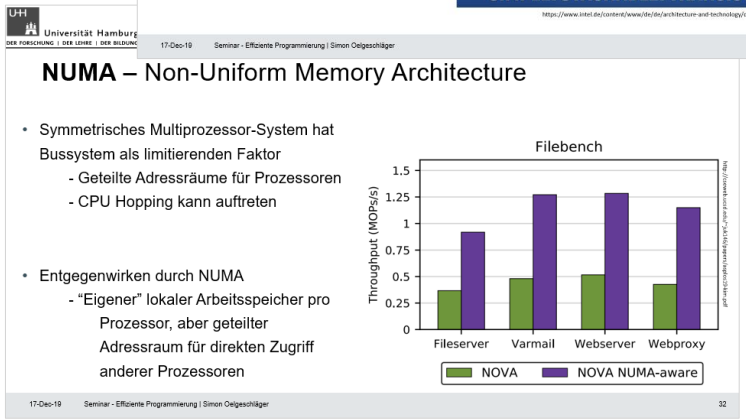
- Einsatz von NVMM rückt immer näher
- Neue Hardwarestruktur ermöglicht deutliche Performancegewinne
- Simple Softwareanpassungen führen bereits zu spürbar besserer Leistung
- Native NVMM Filesystems holen das Maximum aus der neuen Technologie

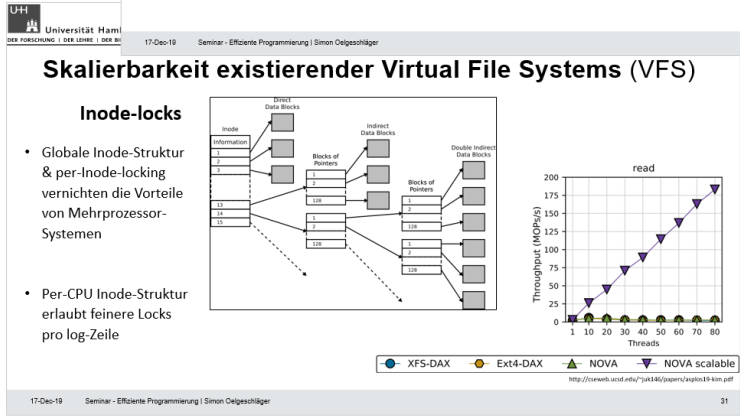
Frohe Weihnachten!!











Quellenverzeichnis

- Grafiken von: https://undraw.co/license

- http://cseweb.ucsd.edu/~juk146/papers/asplos19-kim.pdf
- https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/storage_administration_guide/configuring-persistent-memory-for-file-system-direct-access-dax
- https://en.cppreference.com/w/cpp/string/byte/memcpy
- https://www.quora.com/What-is-the-difference-between-a-journaling-vs-a-log-structured-file-system
- https://www.all-electronics.de/fefet-non-volatile-memories/
- https://www.sigops.org/s/conferences/sosp/2017/slides/fortis-sosp17-slides.pdf
- https://en.wikichip.org/wiki/snia/direct_access
- https://www.computer.org/csdl/proceedings-article/ipccc/2017/08280456/12OmNvjgWDI
- https://www.kernel.org/doc/Documentation/filesystems/dax.txt
- https://www.intel.de/content/www/de/de/architecture-and-technology/optane-technology-animation.html
- https://www.anandtech.com/show/9470/intel-and-micron-announce-3d-xpoint-nonvolatile-memory-technology-1000x-higher-performance-endurance-than-nand/5
- https://ieeexplore.ieee.org/document/6307756
- https://dl.acm.org/citation.cfm?id=3154757
- https://p4.org/assets/P4WE_2018/Robert_Soule.pdf
- https://www.usenix.org/conference/inflow16/workshop-program/presentation/keynote-swanson
- https://www.usenix.org/conference/atc17/technical-sessions/presentation/hu
- https://www.quora.com/When-will-we-have-non-volatile-main-memory
- http://man7.org/linux/man-pages/man2/mmap.2.html
- http://www.cs.columbia.edu/~vatlidak/resources/POSIXmagazine.pdf
- https://itp.tugraz.at/~ahi/VO/2012-04_GLT_mooseFS.pdf