

Kernel Samepage Merging (KSM)

Funktionsweise und Schwachstellen

Martin Heckel

13. Dezember 2021

Hochschule für angewandte Wissenschaften Hof

Grundlagen

Arbeitsspeicher, DRAM

Kernel Samepage Merging (KSM)

Rowhammer

Angriffe auf KSM

Information leaks

Covert Channel

Flip Feng Shui (FFS)

Fazit

Grundlagen

- Herausforderung: Rechner verwenden verschiedene Arten von Speicher:
 - DRAM
 - „Auslagerungsspeicher“ (Swap) in Datei oder Partition (Sekundärspeicher)

- **Herausforderung:** Rechner verwenden verschiedene Arten von Speicher:
 - DRAM
 - „Auslagerungsspeicher“ (Swap) in Datei oder Partition (Sekundärspeicher)
- **Herausforderung:** Mehrere Prozesse gleichzeitig auf einem System, voneinander getrennt

- **Herausforderung:** Rechner verwenden verschiedene Arten von Speicher:
 - DRAM
 - „Auslagerungsspeicher“ (Swap) in Datei oder Partition (Sekundärspeicher)
- **Herausforderung:** Mehrere Prozesse gleichzeitig auf einem System, voneinander getrennt
- **Idee:** Jeder Prozess hat eigenen Adressraum, physischer Speicher wird in diesen Adressraum gemappt

- **Herausforderung:** Rechner verwenden verschiedene Arten von Speicher:
 - DRAM
 - „Auslagerungsspeicher“ (Swap) in Datei oder Partition (Sekundärspeicher)
- **Herausforderung:** Mehrere Prozesse gleichzeitig auf einem System, voneinander getrennt
- **Idee:** Jeder Prozess hat eigenen Adressraum, physischer Speicher wird in diesen Adressraum gemappt
- Um Zugriffe zu beschleunigen, werden Bereiche von 4 KiB Gemappt („Pages“)

Page Tables

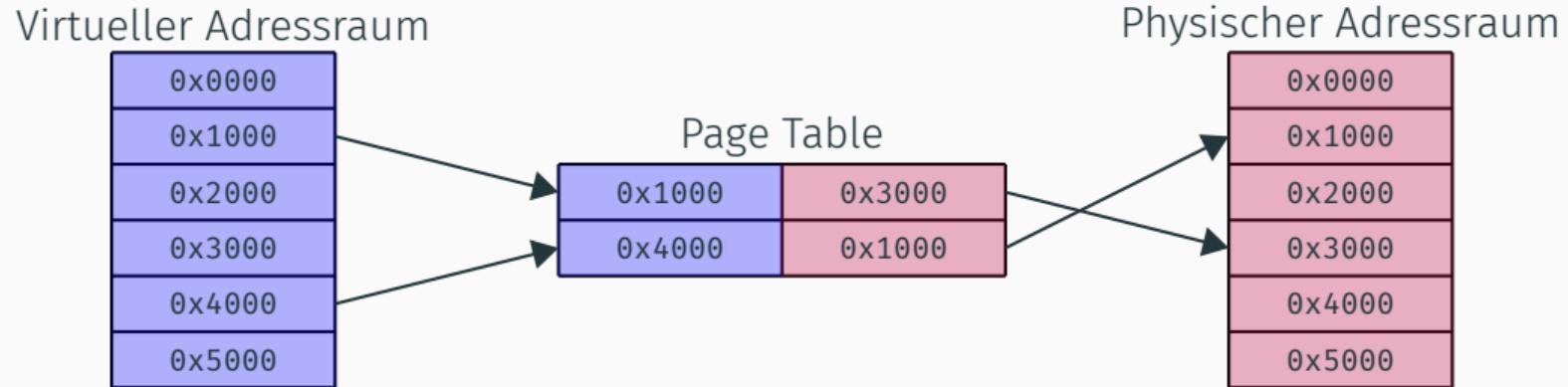


Abbildung basiert auf dem Foliensatz „0x0D Low-Level Fundamentals“ von Prof. Dr. Florian Adamsky

Physischer Aufbau von DRAM

DRAM Zelle

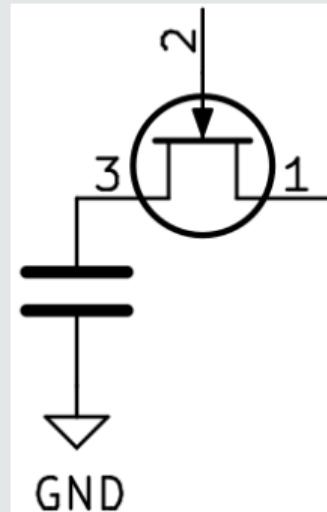


Abbildung aus den Slides zu [3]

Physischer Aufbau von DRAM

DRAM Array

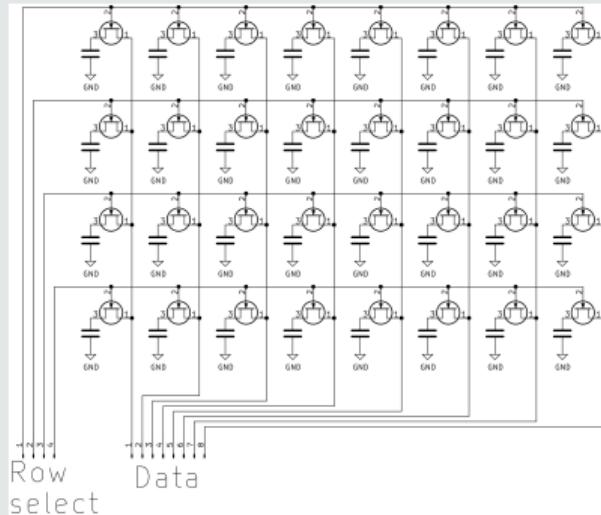


Abbildung aus den Slides zu [3]

Physischer Aufbau von DRAM

Innerhalb einer Bank

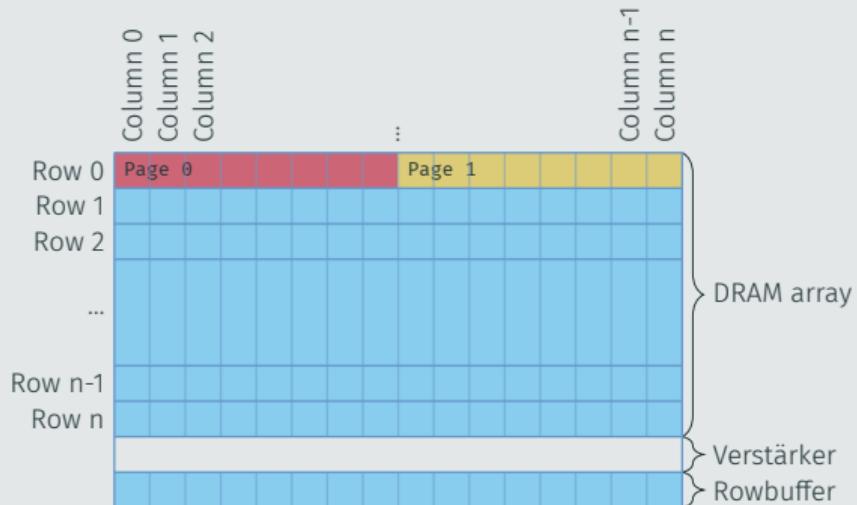


Abbildung aus den Slides zu [3]

Physischer Aufbau von DRAM

Bank

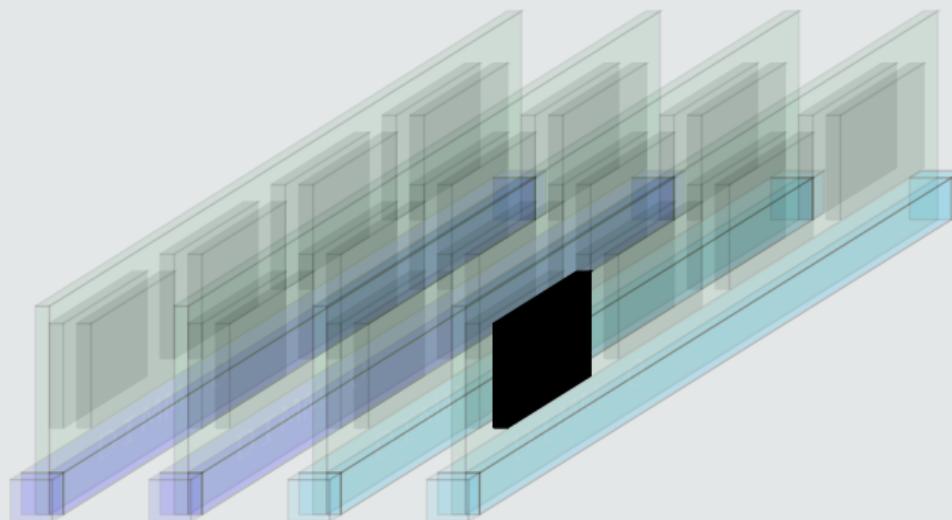


Abbildung aus den Slides zu [3]

Physischer Aufbau von DRAM

Rank

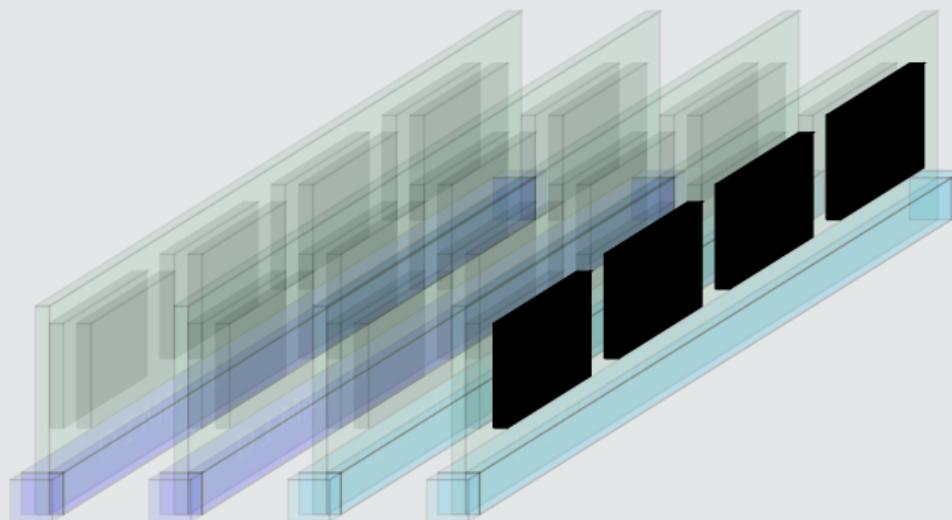


Abbildung aus den Slides zu [3]

Physischer Aufbau von DRAM

DIMM

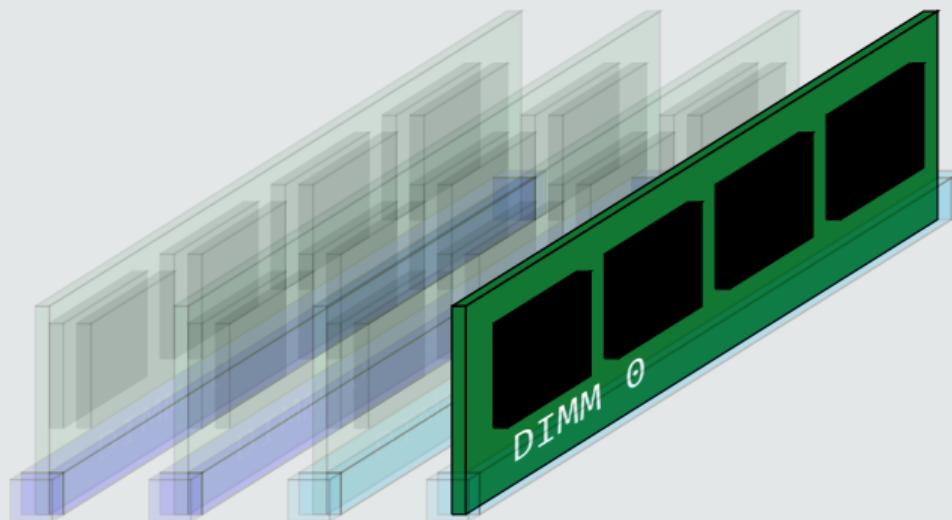


Abbildung aus den Slides zu [3]

Channel

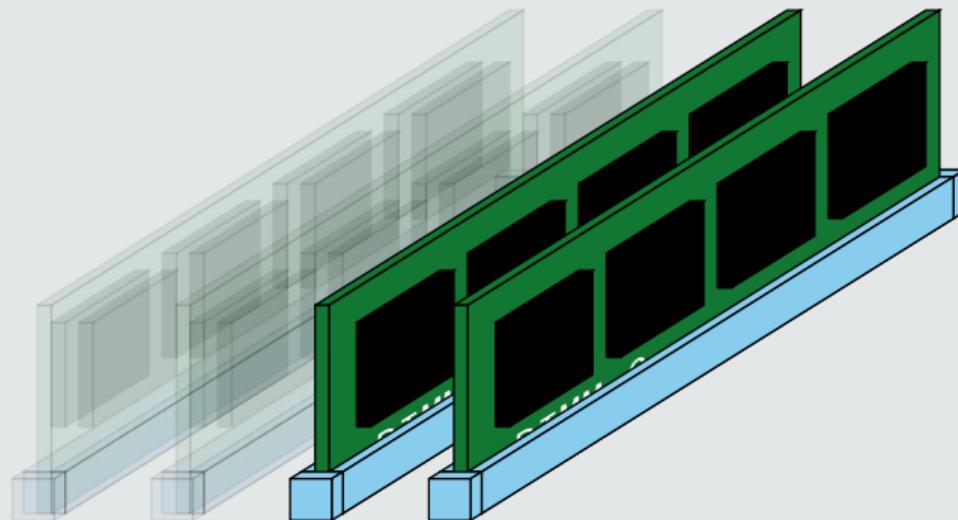


Abbildung aus den Slides zu [3]

DRAM des Systems

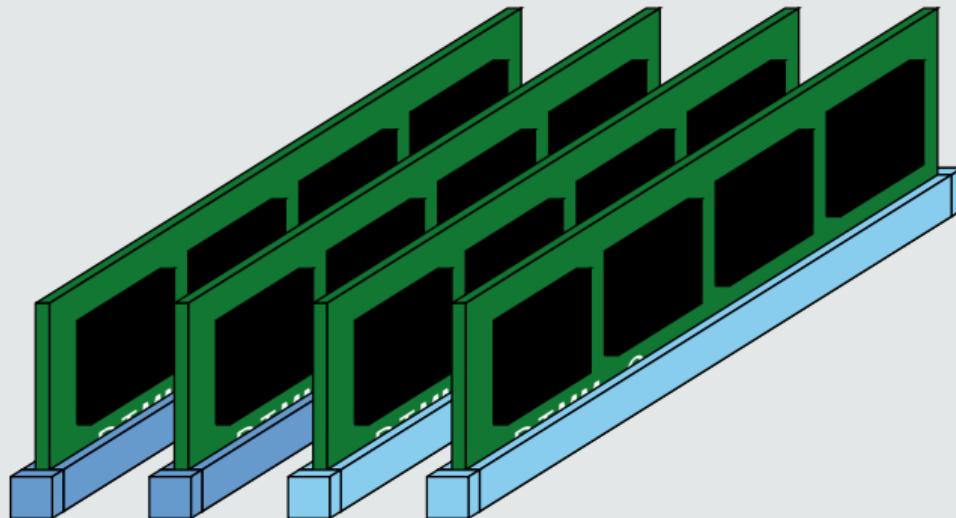


Abbildung aus den Slides zu [3]

Physischer Aufbau von DRAM

- Der Ladungszustand eines Kondensators wird einem logischen Zustand („1“ oder „0“) zugeordnet

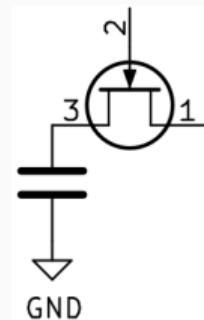


Abbildung aus den Slides zu [3]

Physischer Aufbau von DRAM

- Der Ladungszustand eines Kondensators wird einem logischen Zustand („1“ oder „0“) zugeordnet
- Kondensatoren entladen sich über die Zeit („Leckstrom“)

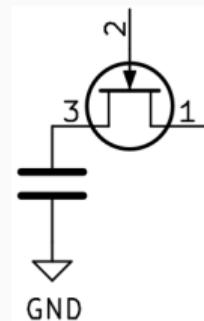


Abbildung aus den Slides zu [3]

Physischer Aufbau von DRAM

- Der Ladungszustand eines Kondensators wird einem logischen Zustand („1“ oder „0“) zugeordnet
- Kondensatoren entladen sich über die Zeit („Leckstrom“)
- Herausforderung:** Aufladen eines vollen Kondensators bevor dieser als „leer“ interpretiert wird

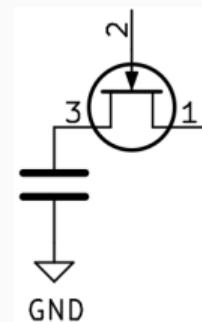


Abbildung aus den Slides zu [3]

Physischer Aufbau von DRAM

- Der Ladungszustand eines Kondensators wird einem logischen Zustand („1“ oder „0“) zugeordnet
- Kondensatoren entladen sich über die Zeit („Leckstrom“)
- Herausforderung:** Aufladen eines vollen Kondensators bevor dieser als „leer“ interpretiert wird
- Lösung:** Zyklisches Lesen und zurück Schreiben von Rows (typ. alle 64 ms), **Refresh**

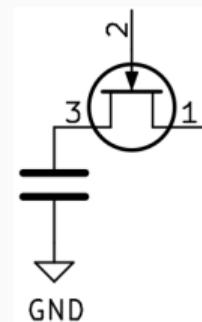
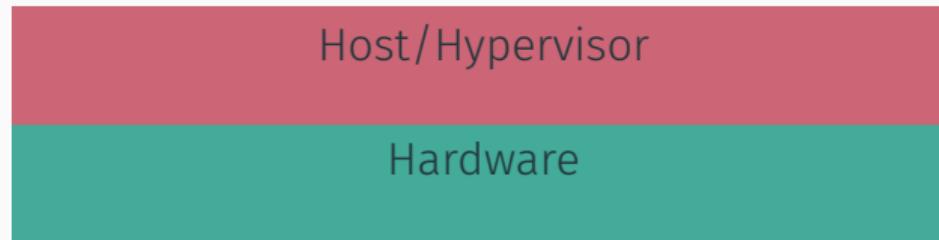


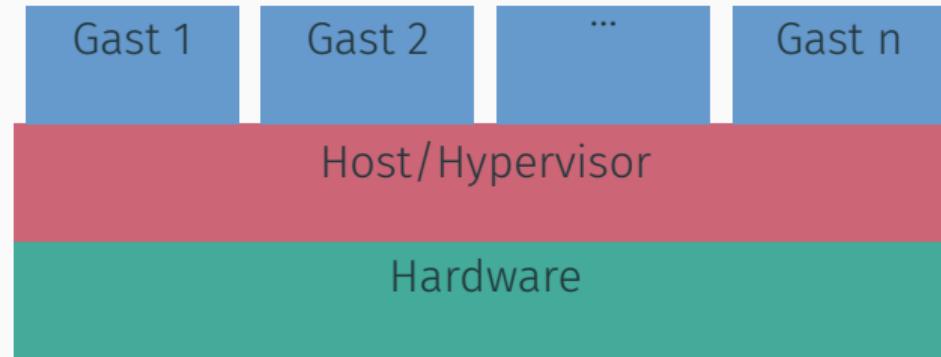
Abbildung aus den Slides zu [3]

Hardware

Virtualisierung



Virtualisierung



Speicherverwaltung in virtualisierten Umgebungen

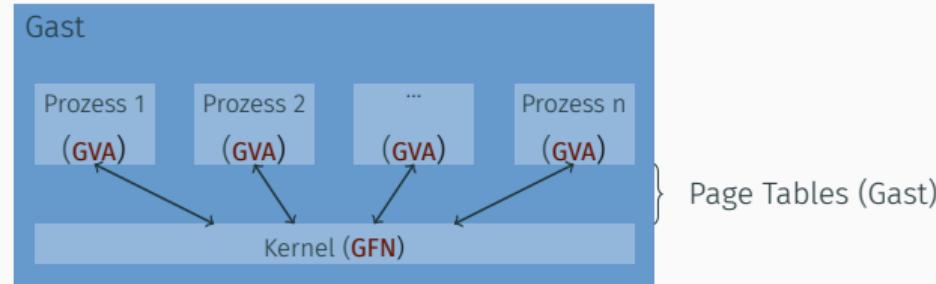


Abbildung aus den Slides zu [3]

Speicherverwaltung in virtualisierten Umgebungen

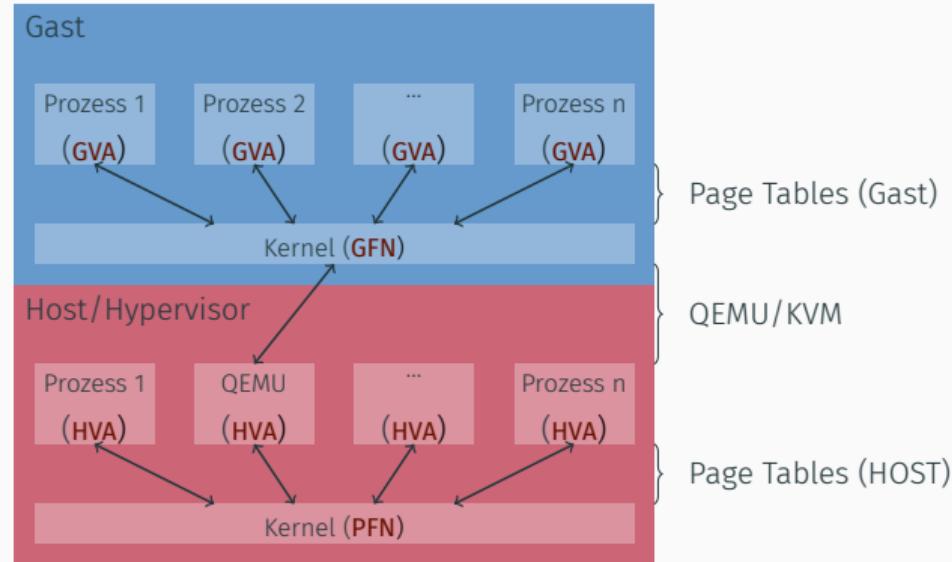


Abbildung aus den Slides zu [3]

Speicherverwaltung in virtualisierten Umgebungen

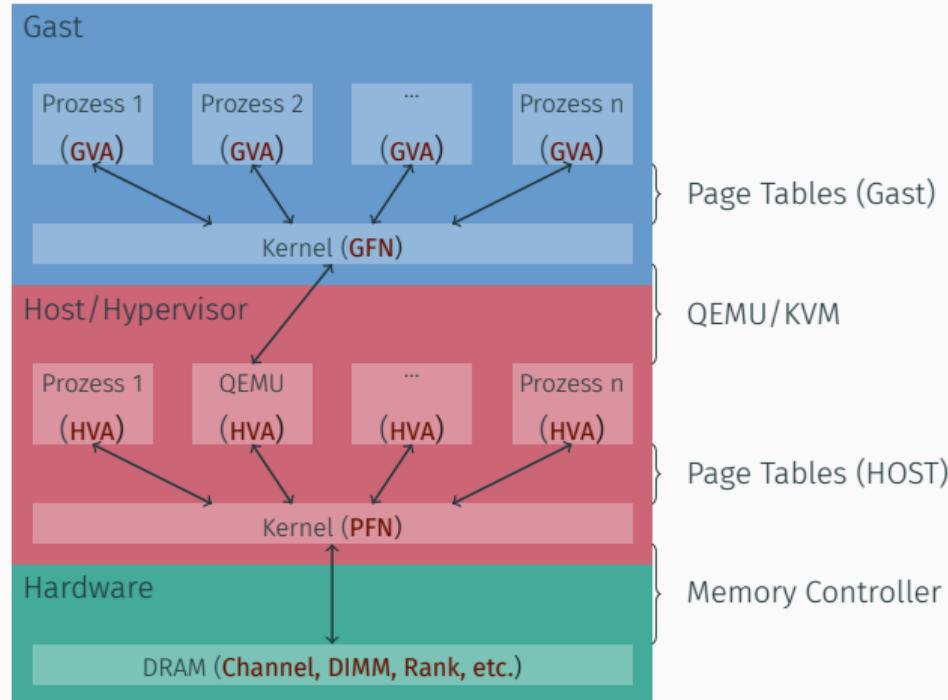


Abbildung aus den Slides zu [3]

Demo 1

Analyse virtueller Adressierung

Kernel Samepage Merging (KSM)

- Einige Speicherbereiche werden normalerweise nur lesend verwendet
- Einige Speicherbereiche haben identische Inhalte

Kernel Samepage Merging (KSM)

- Einige Speicherbereiche werden normalerweise nur lesend verwendet
- Einige Speicherbereiche haben identische Inhalte
- Anpassen der Page Tables, um mehrfach vorkommende Bereiche nur einmal im physischen Speicher zu halten (mergen von Speicherseiten mit identischen Inhalten)

Kernel Samepage Merging (KSM)

- Einige Speicherbereiche werden normalerweise nur lesend verwendet
- Einige Speicherbereiche haben identische Inhalte
- Anpassen der Page Tables, um mehrfach vorkommende Bereiche nur einmal im physischen Speicher zu halten (mergen von Speicherseiten mit identischen Inhalten)
- Copy-on-Write Policy, um Schreibzugriff auf deduplizierte Seiten zu verhindern

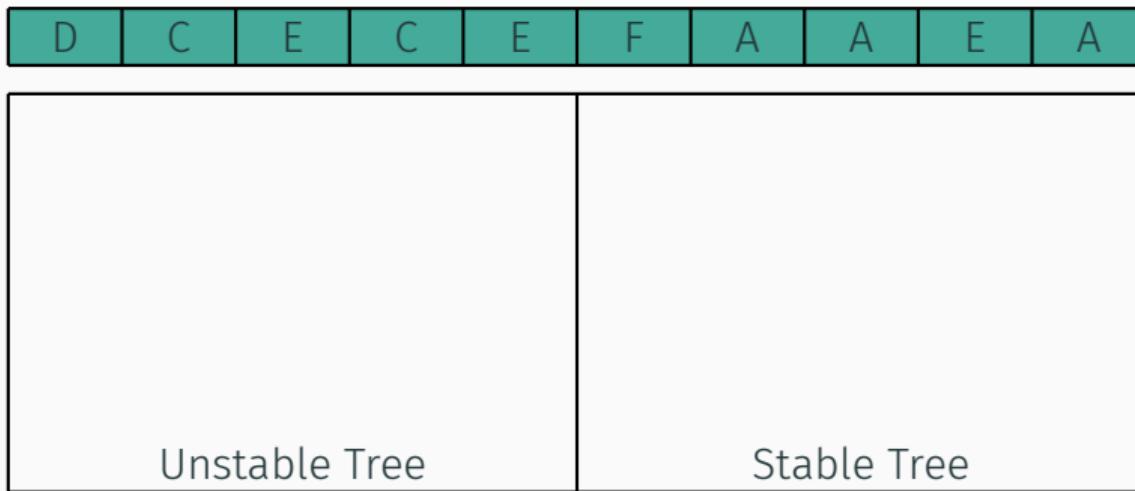
Kernel Samepage Merging (KSM)

- KSM berücksichtigt nur Seiten, die als *merge candidat* deklariert sind (s. MADVISE(2))

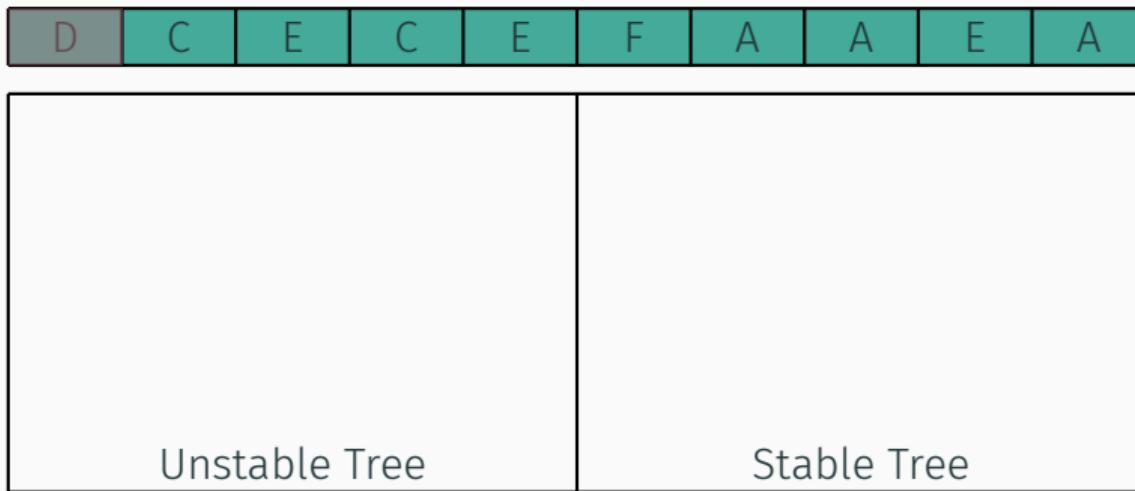
Kernel Samepage Merging (KSM)

- KSM berücksichtigt nur Seiten, die als *merge candidat* deklariert sind (s. MADVISE(2))
- Verwendung von zwei rot-schwarz Bäumen, um Seiten zu verwalten:
 - *Stable tree*: enthält Seiten, die bereits dedupliziert wurden
 - *Unstable tree*: enthält Seiten, die dedupliziert werden können (wenn eine zweite Seite mit identischen Inhalt gefunden wird)

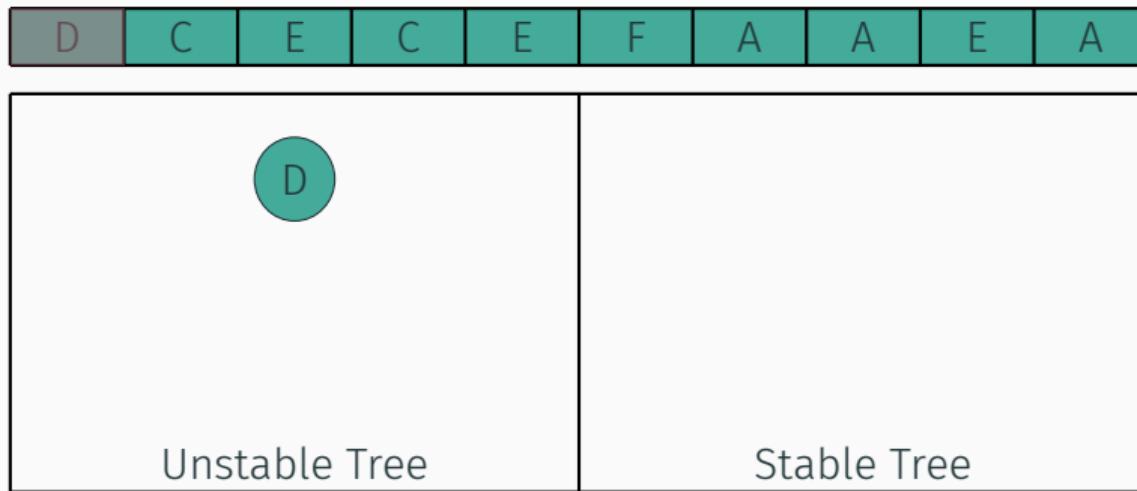
Kernel Samepage Merging (KSM)



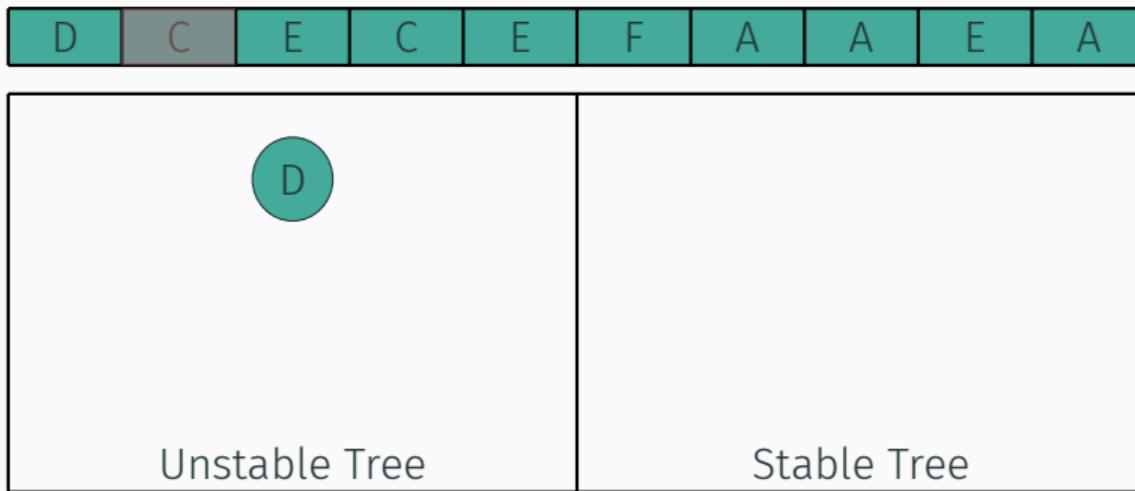
Kernel Samepage Merging (KSM)



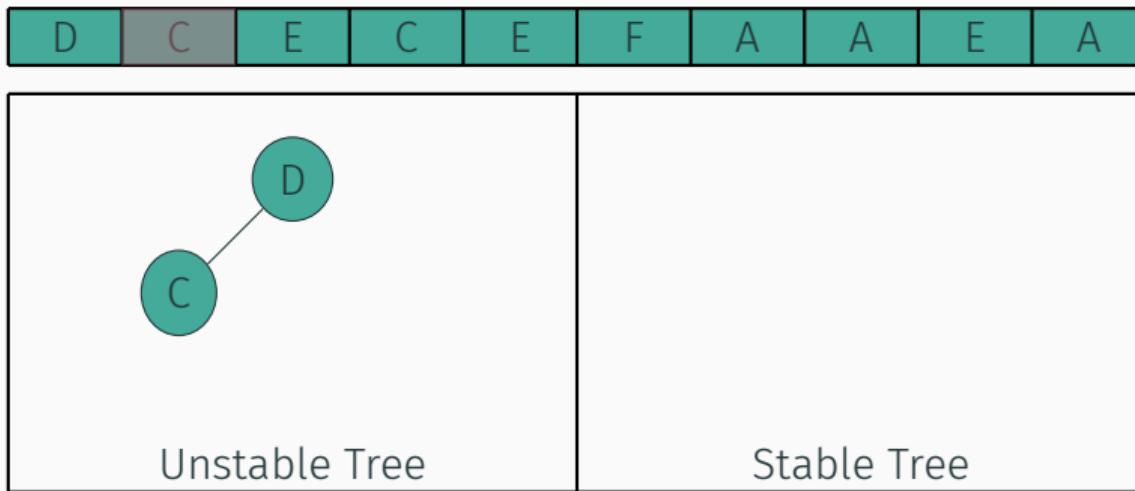
Kernel Samepage Merging (KSM)



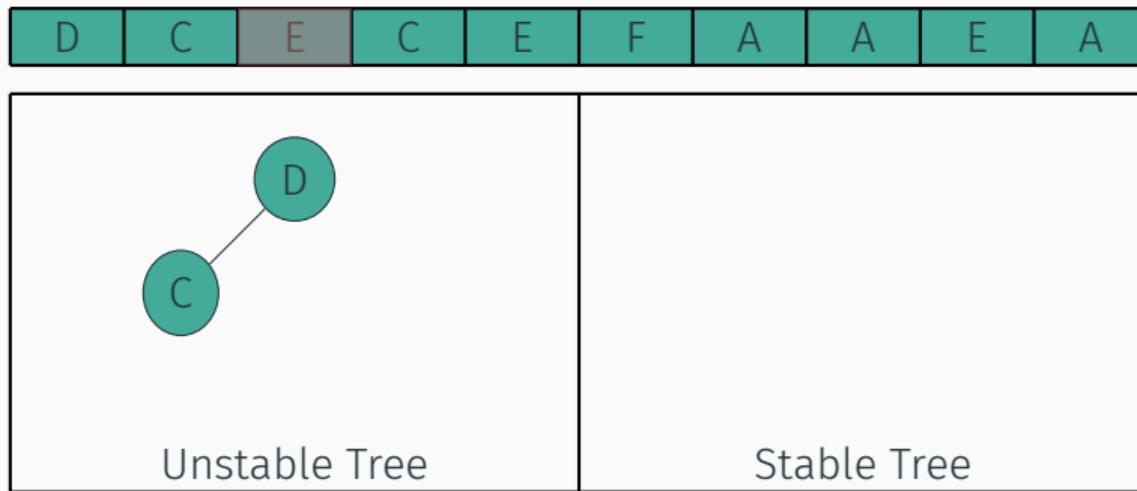
Kernel Samepage Merging (KSM)



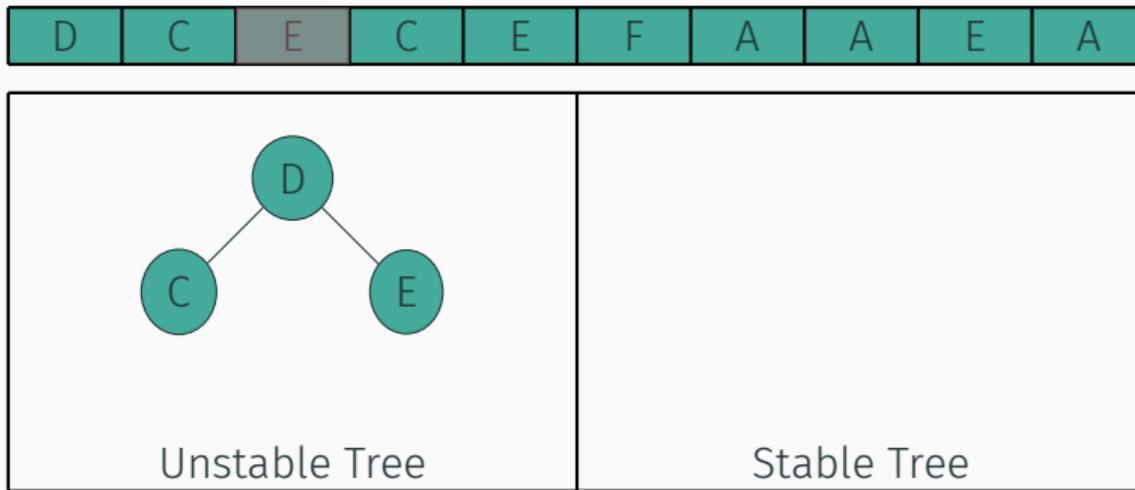
Kernel Samepage Merging (KSM)



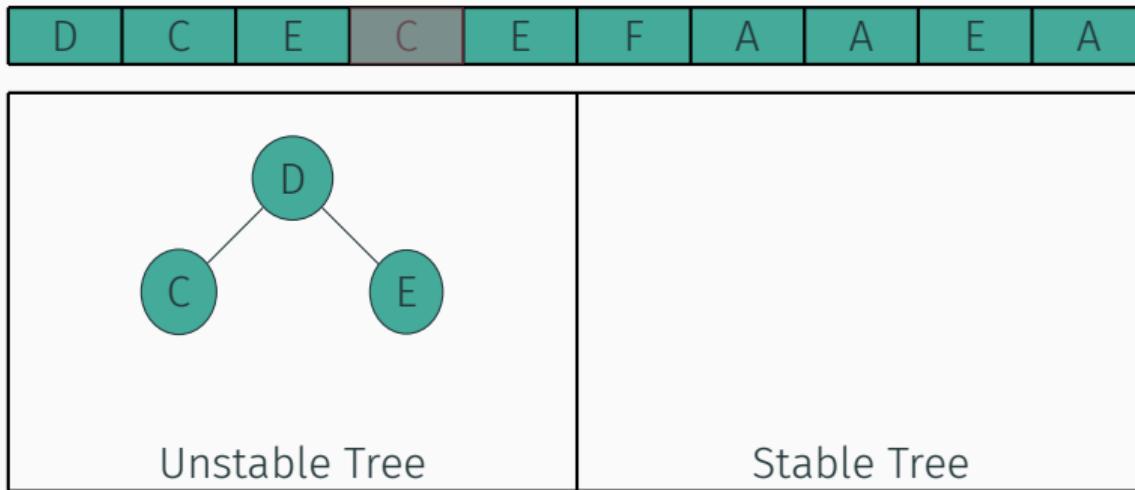
Kernel Samepage Merging (KSM)



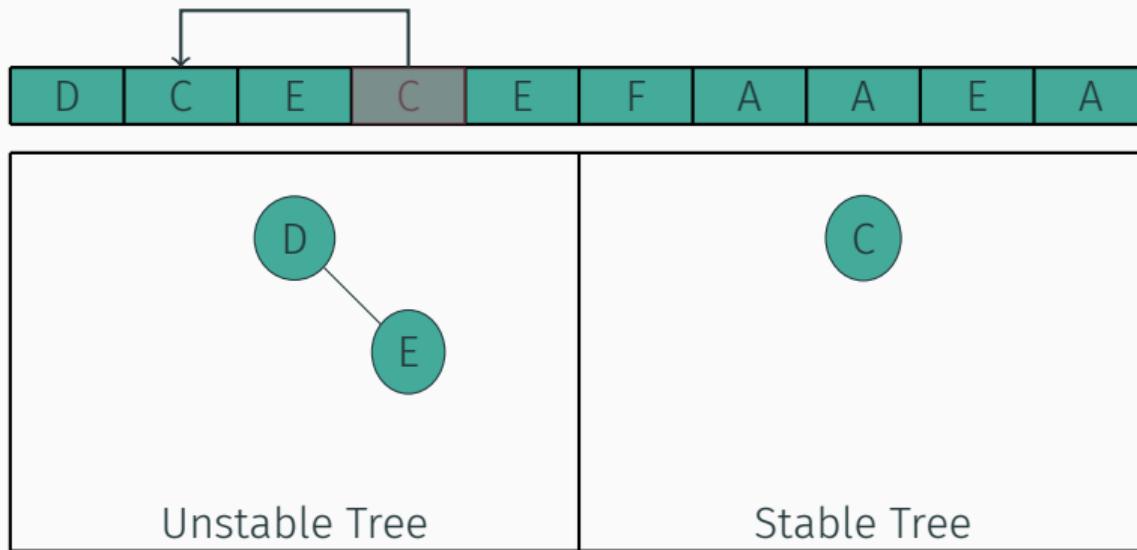
Kernel Samepage Merging (KSM)



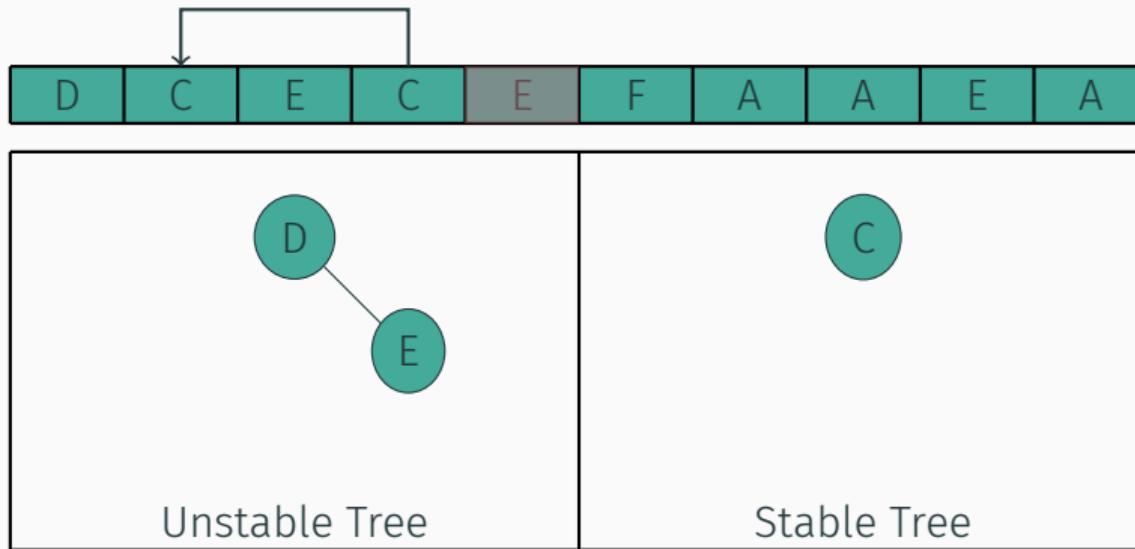
Kernel Samepage Merging (KSM)



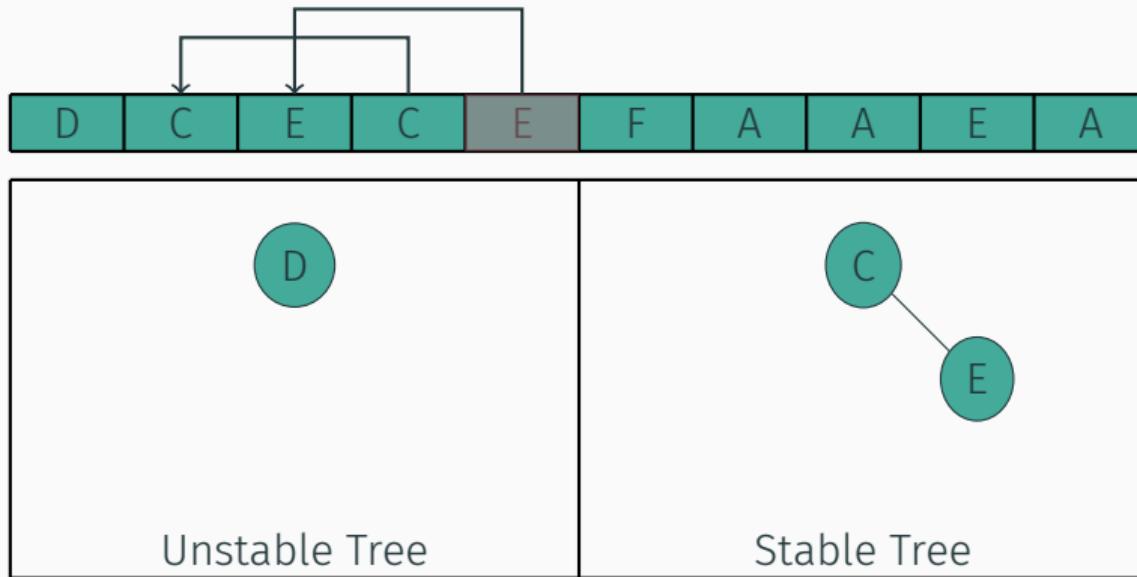
Kernel Samepage Merging (KSM)



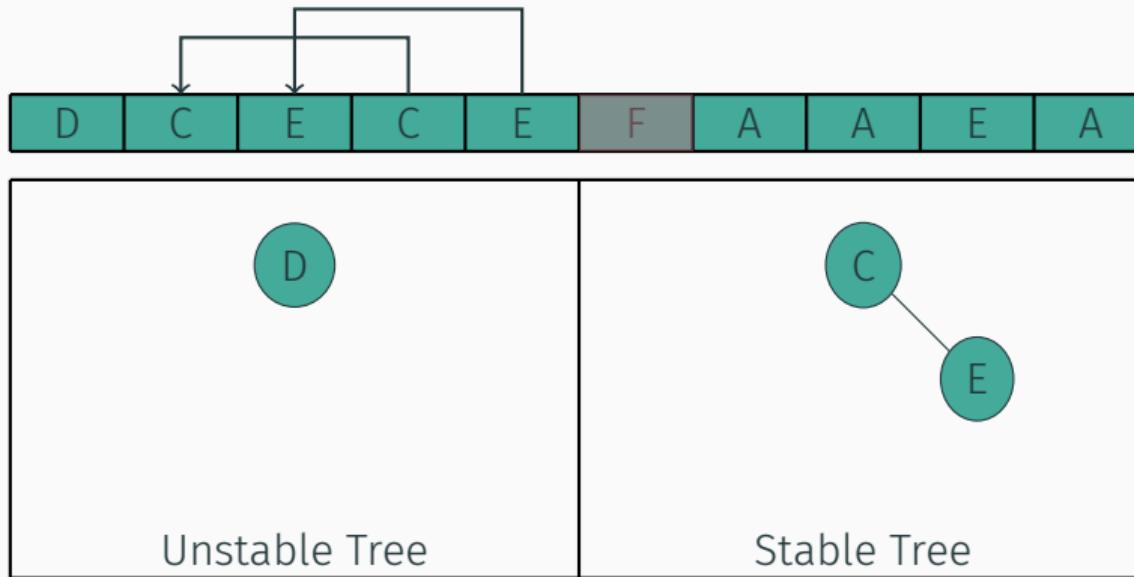
Kernel Samepage Merging (KSM)



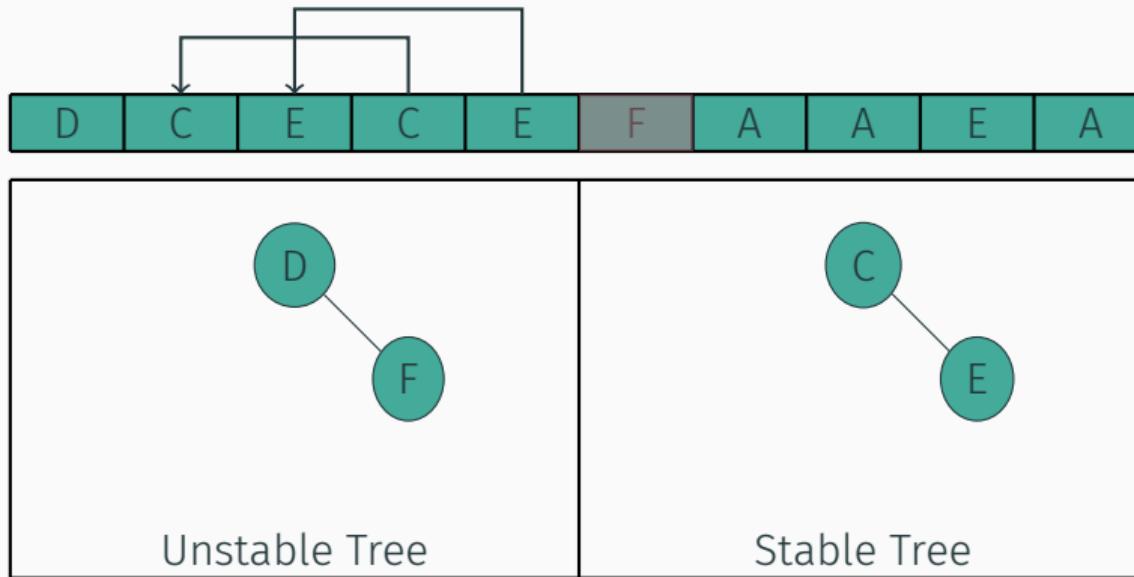
Kernel Samepage Merging (KSM)



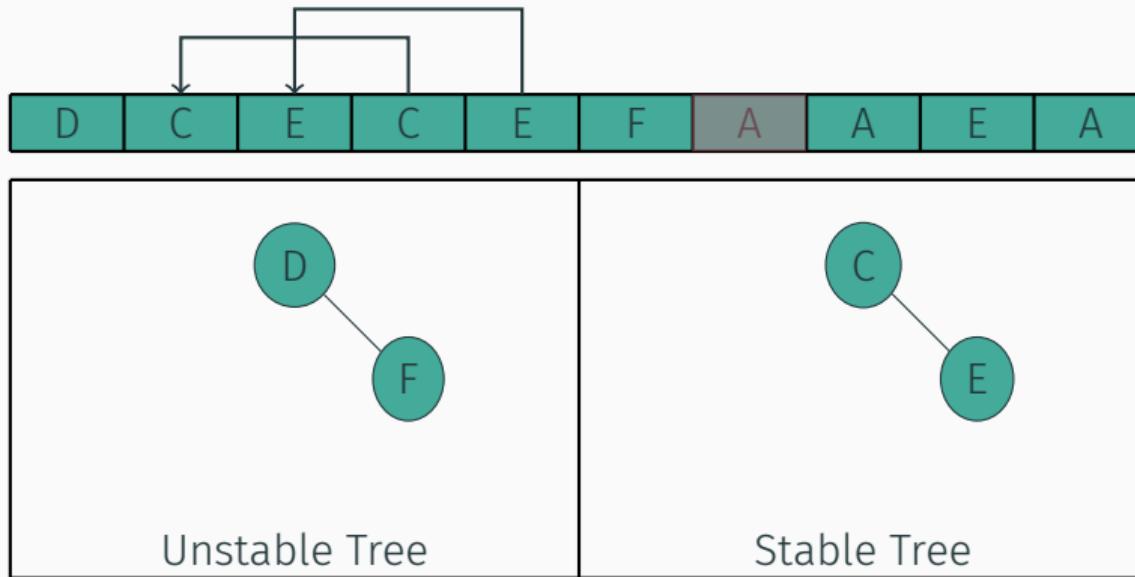
Kernel Samepage Merging (KSM)



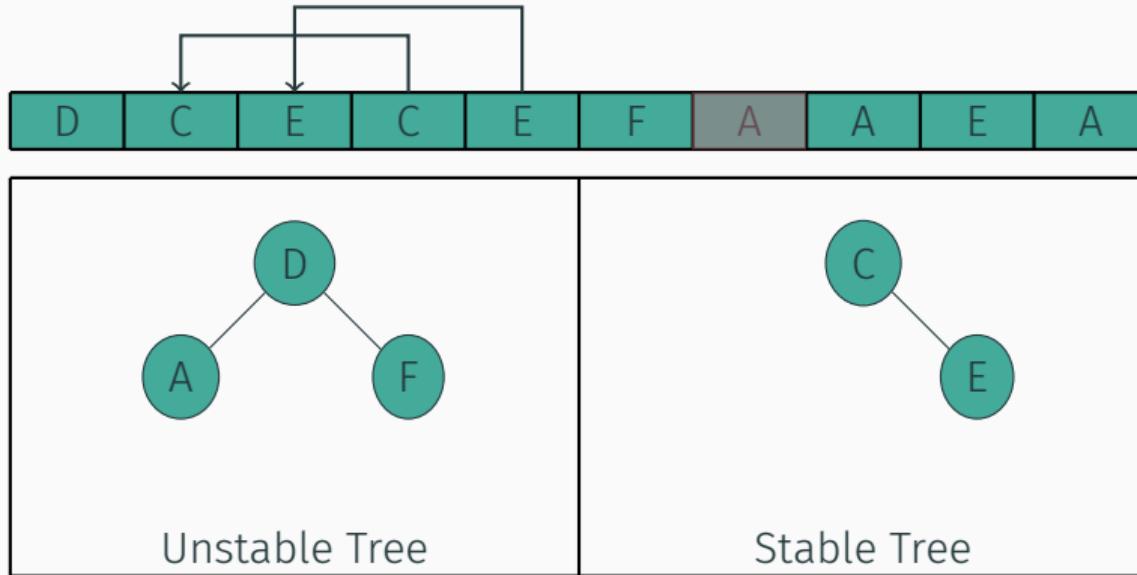
Kernel Samepage Merging (KSM)



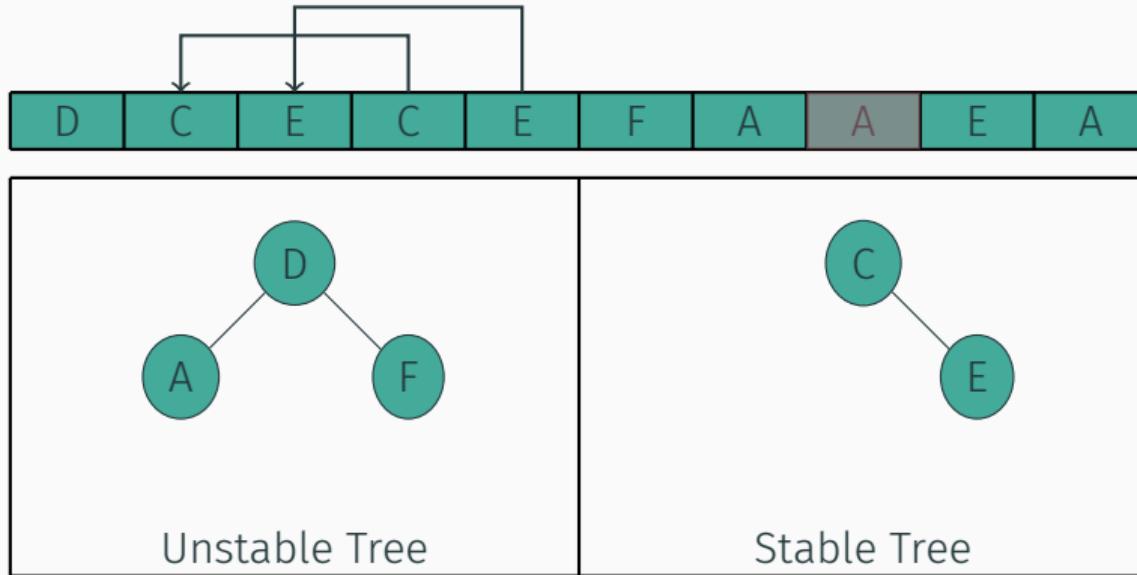
Kernel Samepage Merging (KSM)



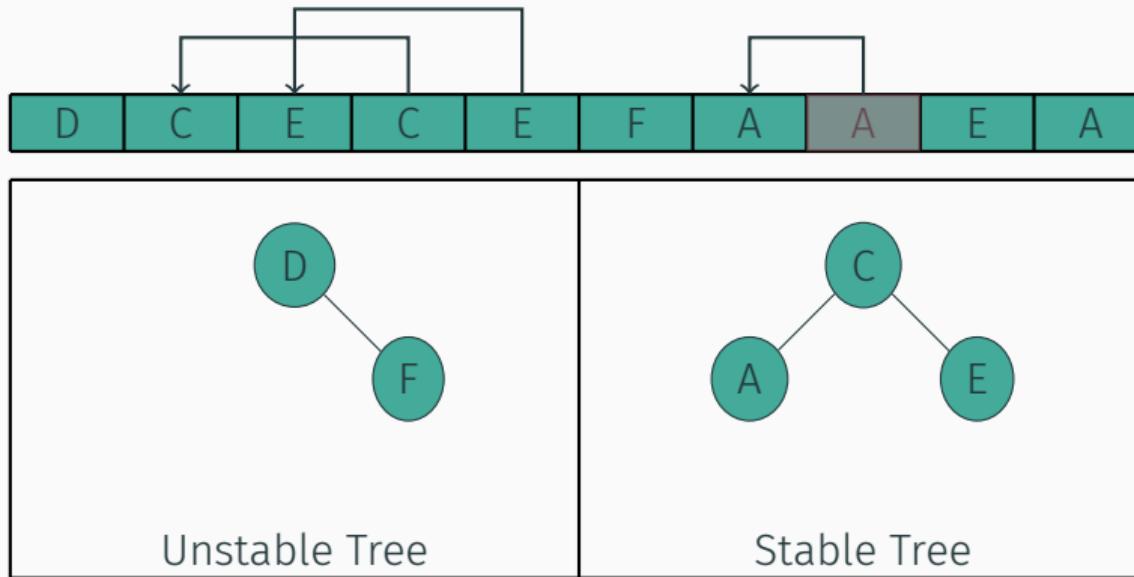
Kernel Samepage Merging (KSM)



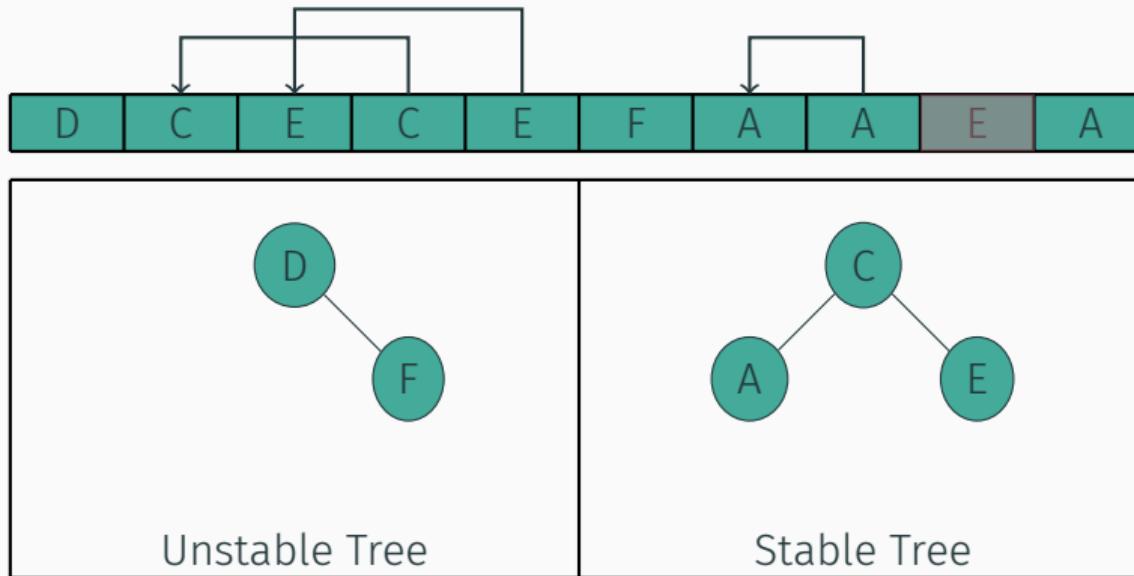
Kernel Samepage Merging (KSM)



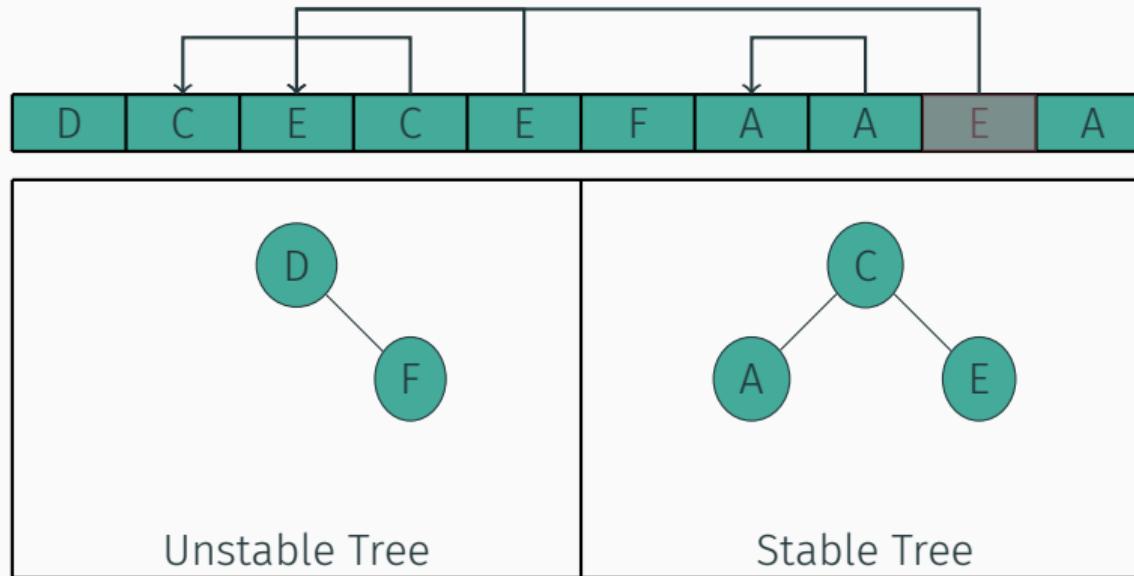
Kernel Samepage Merging (KSM)



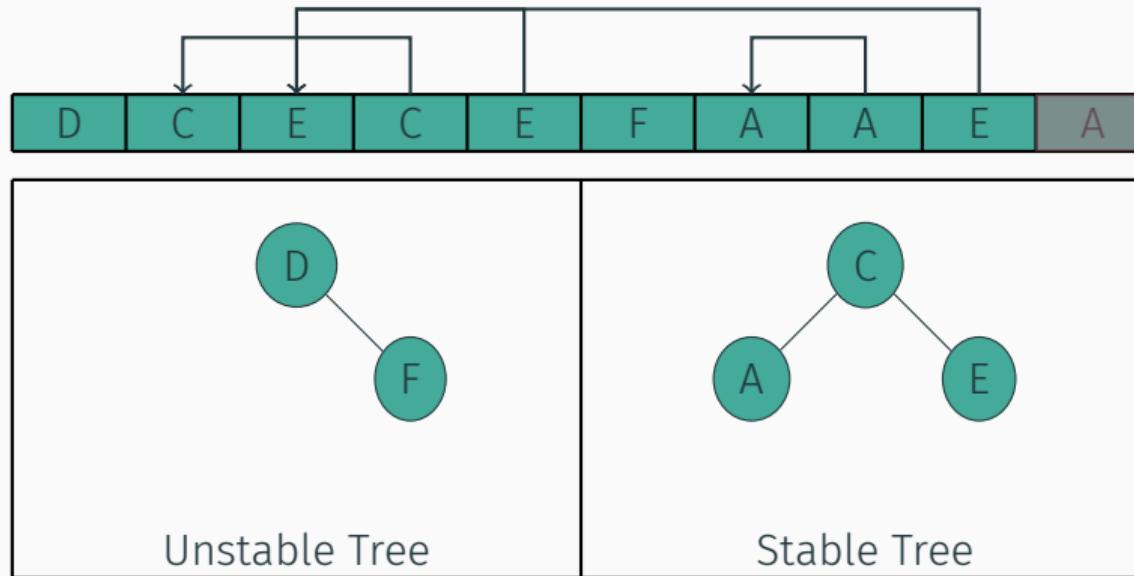
Kernel Samepage Merging (KSM)



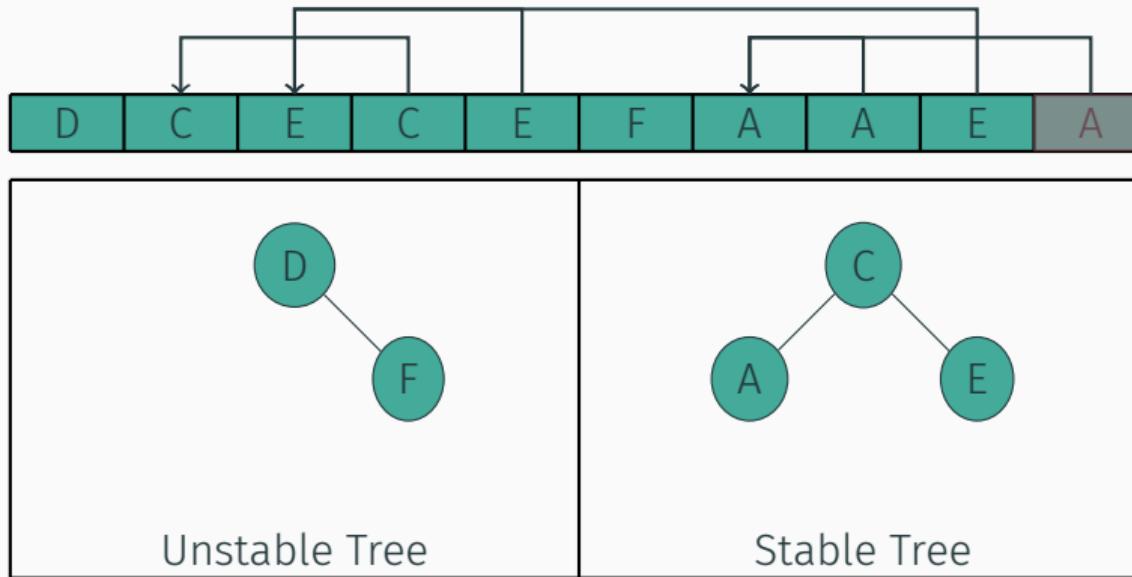
Kernel Samepage Merging (KSM)



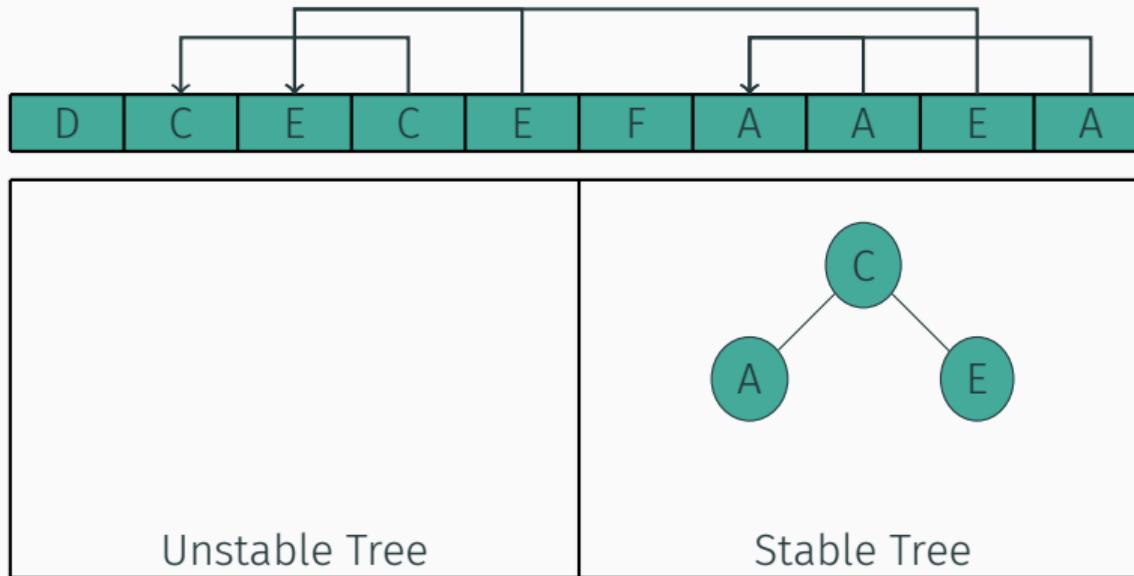
Kernel Samepage Merging (KSM)



Kernel Samepage Merging (KSM)



Kernel Samepage Merging (KSM)



Demo 2

Analyse virtueller Adressierung mit KSM

- DRAM Zellen sind in einem Array aus Zeilen (*Rows*) und Spalten (*Columns*) angeordnet
- Zugriff erfolgt über den Rowbuffer gepuffert auf ganze Rows

- DRAM Zellen sind in einem Array aus Zeilen (*Rows*) und Spalten (*Columns*) angeordnet
- Zugriff erfolgt über den Rowbuffer gepuffert auf ganze Rows
- Höhere Speicherkapazität erfordert höhere Integrationsdichte der DRAM Zellen

- DRAM Zellen sind in einem Array aus Zeilen (*Rows*) und Spalten (*Columns*) angeordnet
- Zugriff erfolgt über den Rowbuffer gepuffert auf ganze Rows
- Höhere Speicherkapazität erfordert höhere Integrationsdichte der DRAM Zellen
- Zellen sind so nah zusammen dass häufige Zugriffe auf Rows zu Speicherfehlern in benachbarten Rows führen

Rowhammer

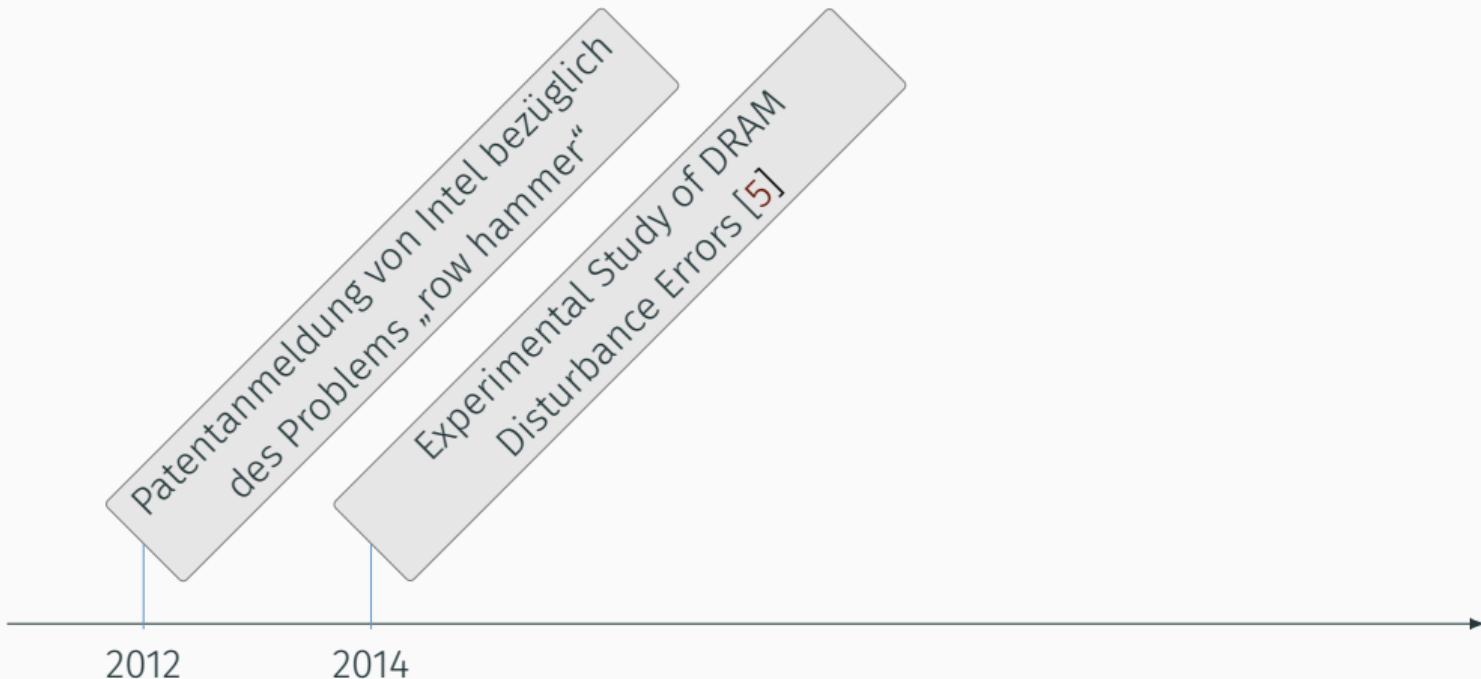
Rowhammer

Patentanmeldung von Intel bezüglich
des Problems „row hammer“

The diagram features a horizontal timeline arrow pointing to the right. A vertical blue line extends from the year 2012 on the timeline up to a grey rectangular callout box. The callout box contains the text "Patentanmeldung von Intel bezüglich des Problems „row hammer“" in black font. The entire diagram is set against a white background.

2012

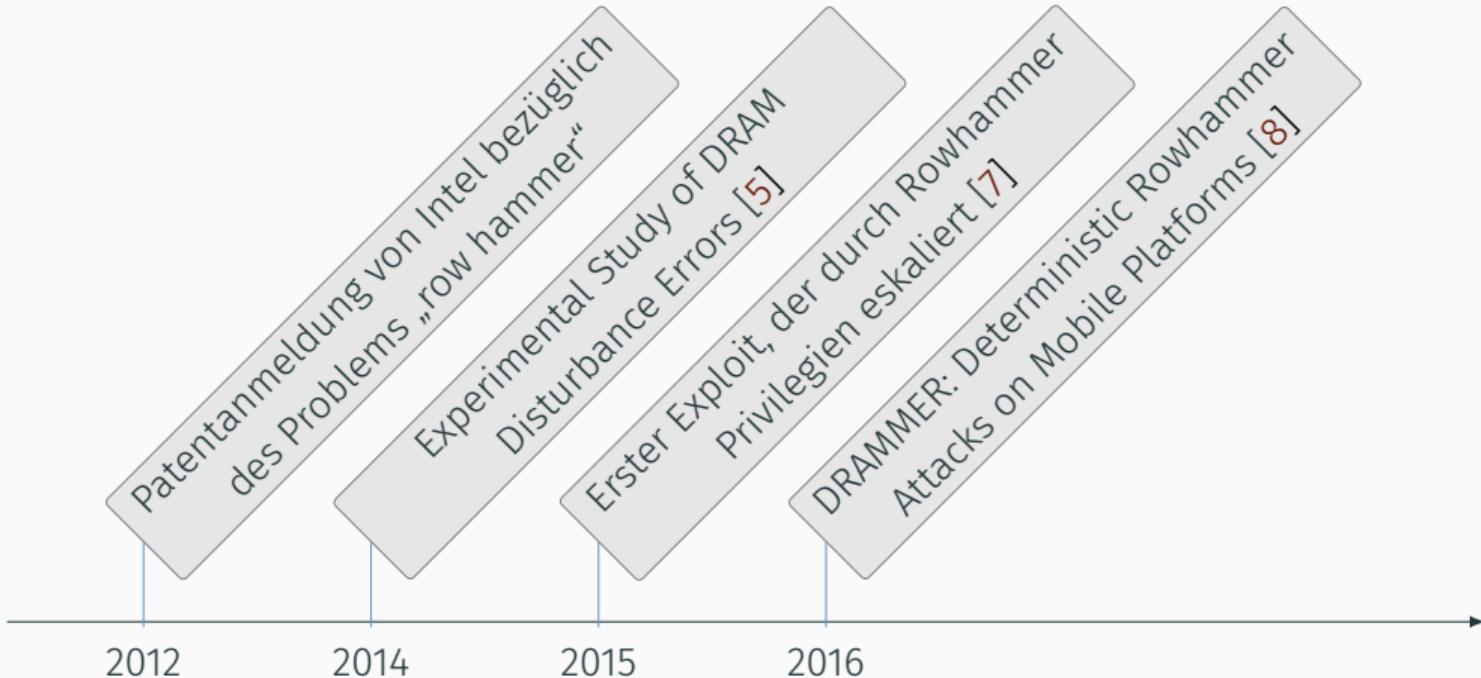
Rowhammer



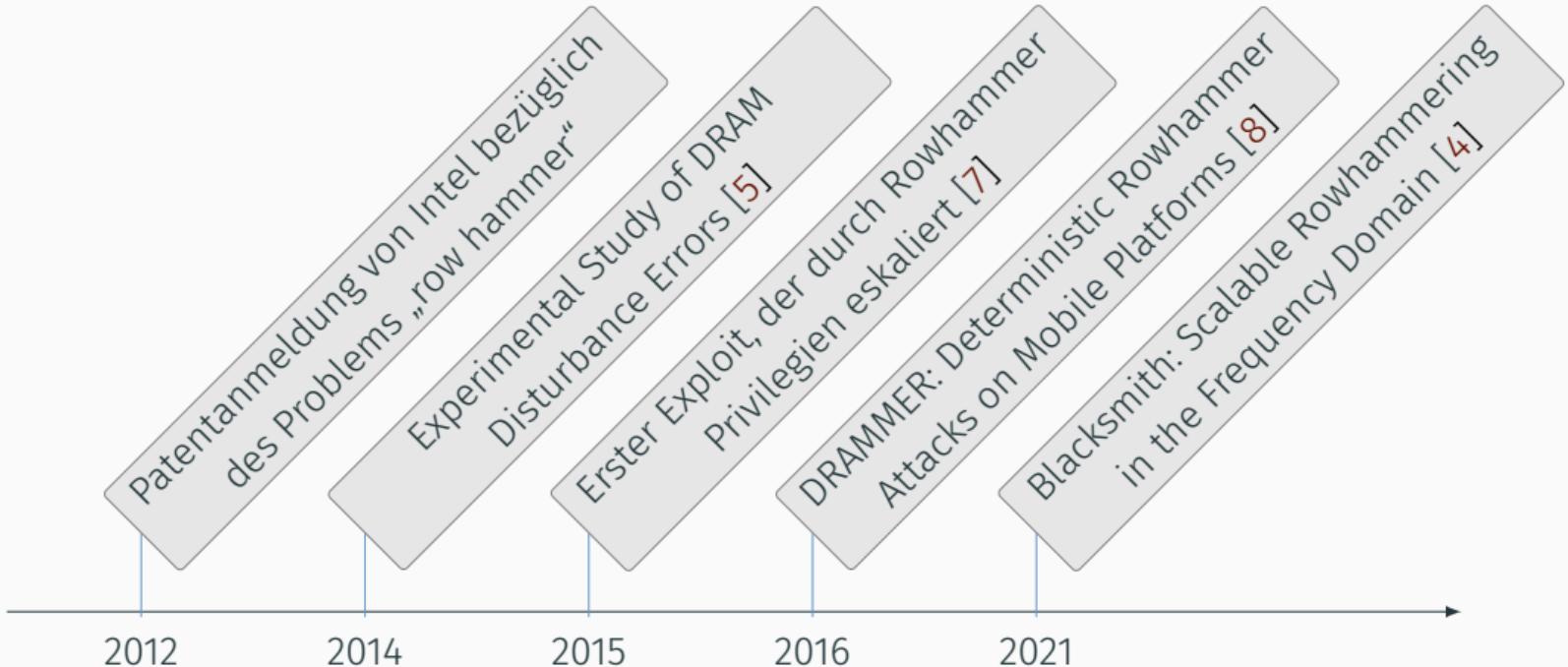
Rowhammer



Rowhammer

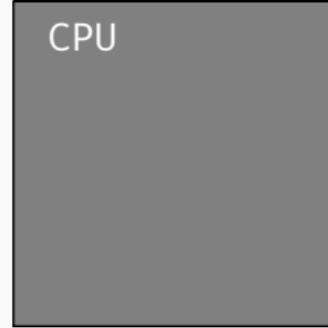


Rowhammer



Rowhammer

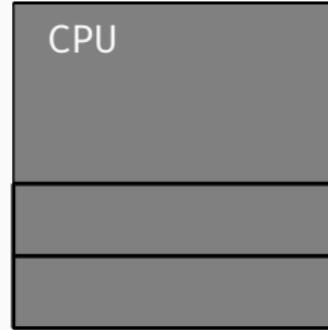
```
1  hammer:  
2      mov  eax, X  
3      mov  ebx, Y  
4      clflush X  
5      clflush Y  
6      jmp  hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

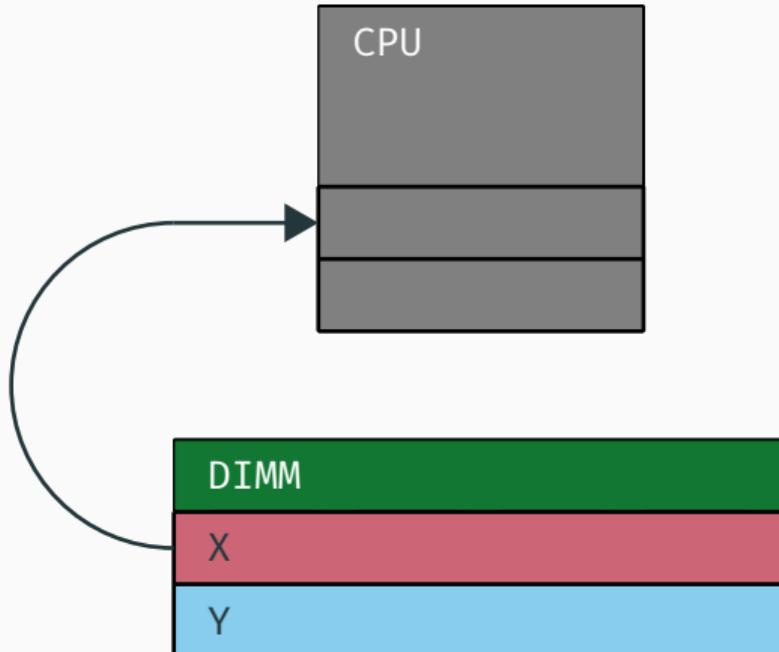
```
1  hammer:
2    mov eax, X
3    mov ebx, Y
4    clflush X
5    clflush Y
6    jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

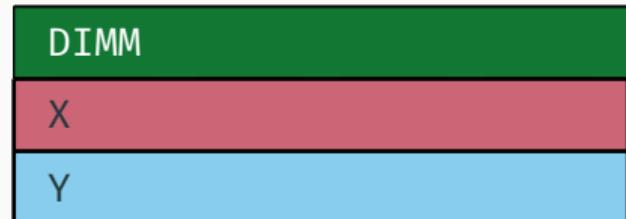
```
1  hammer:  
2      mov  eax, X  
3      mov  ebx, Y  
4      clflush X  
5      clflush Y  
6      jmp  hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

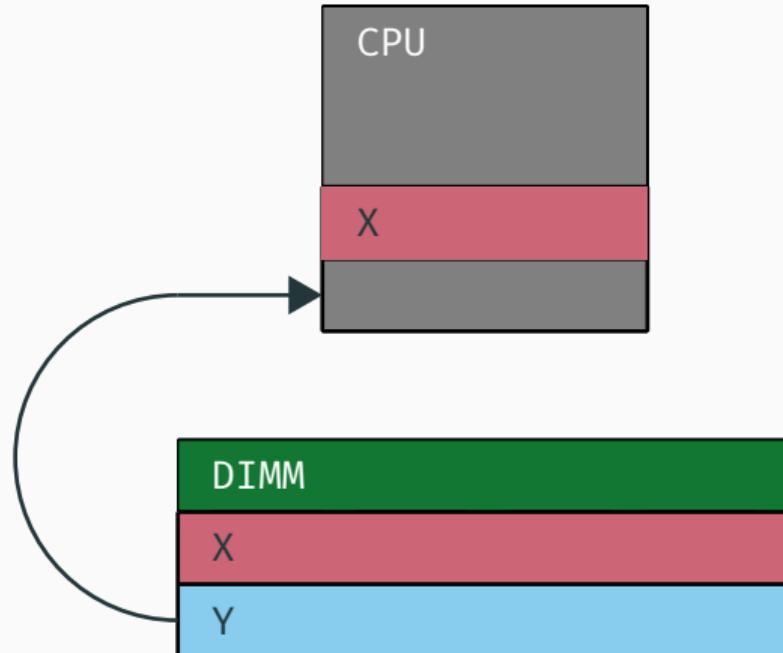
```
1 hammer:  
2     mov eax, X  
3     mov ebx, Y  
4     clflush X  
5     clflush Y  
6     jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

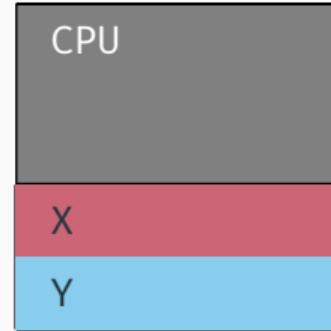
```
1  hammer:
2      mov eax, X
3      mov ebx, Y
4      clflush X
5      clflush Y
6      jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

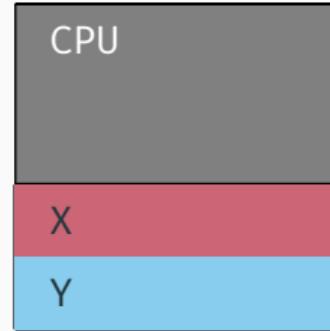
```
1  hammer:
2      mov eax, X
3      mov ebx, Y
4      clflush X
5      clflush Y
6      jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

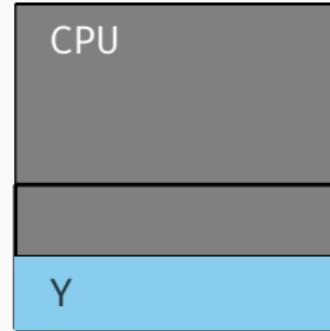
```
1  hammer:
2      mov eax, X
3      mov ebx, Y
4      clflush X
5      clflush Y
6      jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

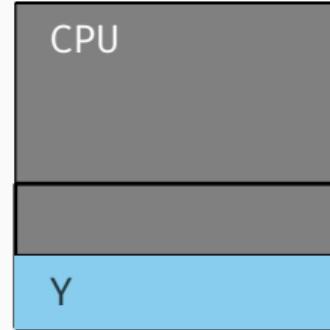
```
1  hammer:
2      mov eax, X
3      mov ebx, Y
4      clflush X
5      clflush Y
6      jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

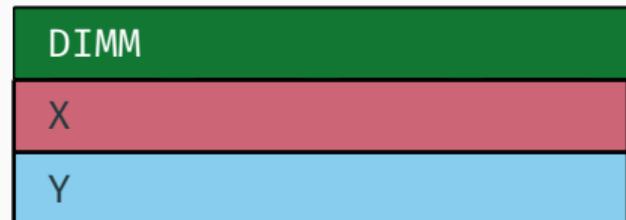
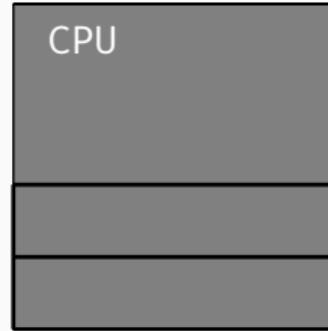
```
1  hammer:
2      mov eax, X
3      mov ebx, Y
4      clflush X
5      clflush Y
6      jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

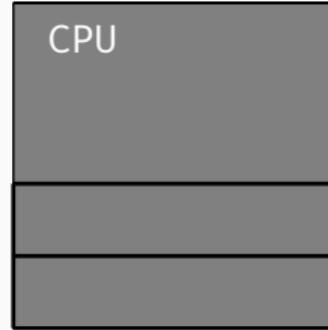
```
1  hammer:
2      mov eax, X
3      mov ebx, Y
4      clflush X
5      clflush Y
6      jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

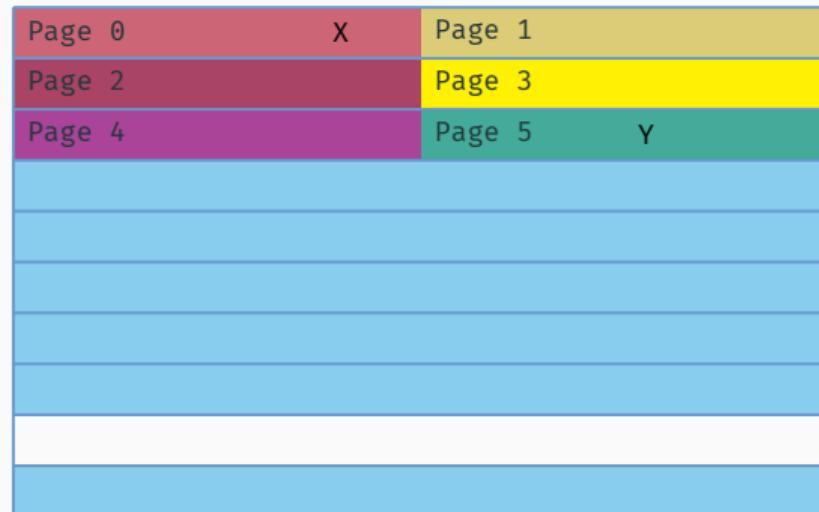
```
1  hammer:
2      mov eax, X
3      mov ebx, Y
4      clflush X
5      clflush Y
6      jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

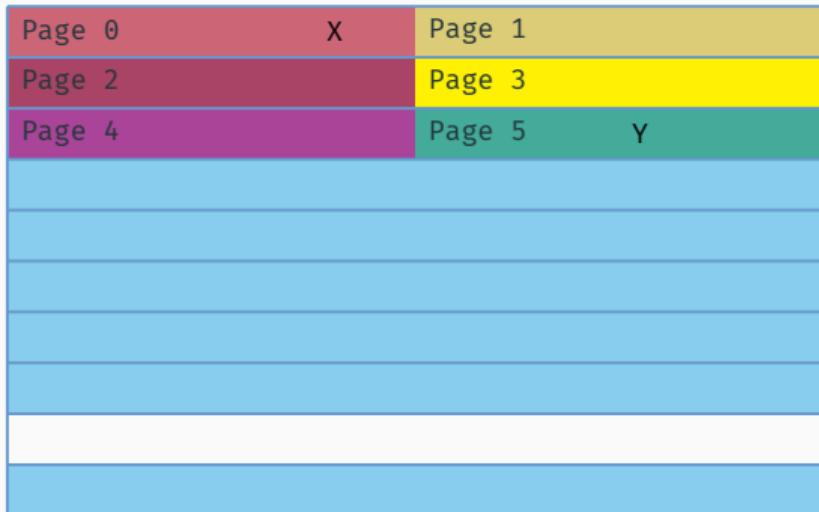
```
1    hammer:  
2        mov eax, X  
3        mov ebx, Y  
4        clflush X  
5        clflush Y  
6        jmp hammer
```



Quellcode von Wikipedia, Abbildung aus den Slides zu [3]

Rowhammer

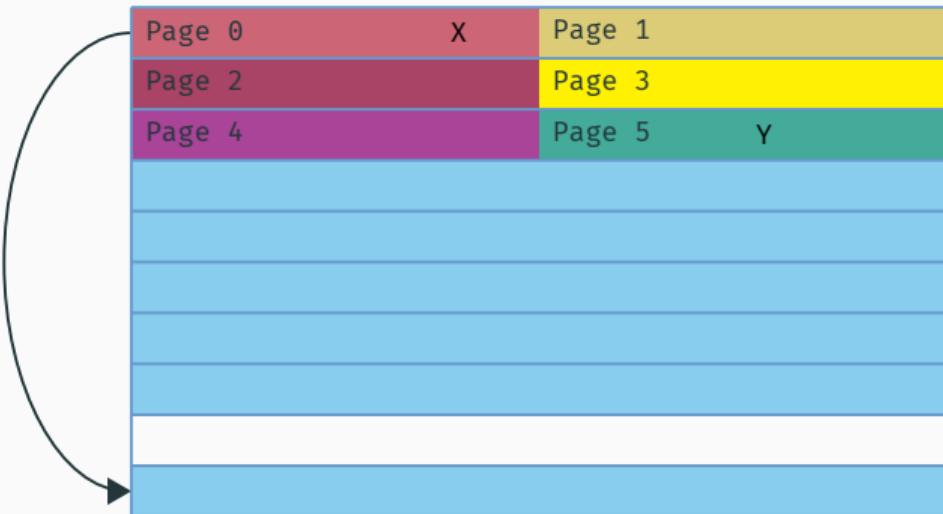
```
1 hammer:  
2     mov eax, X  
3     mov ebx, Y  
4     clflush X  
5     clflush Y  
6     jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

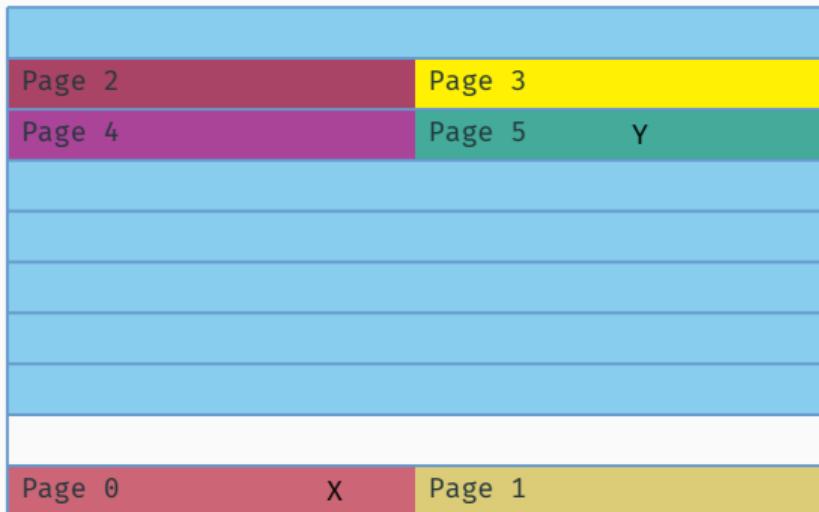
```
1 hammer:  
2     mov eax, X  
3     mov ebx, Y  
4     clflush X  
5     clflush Y  
6     jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

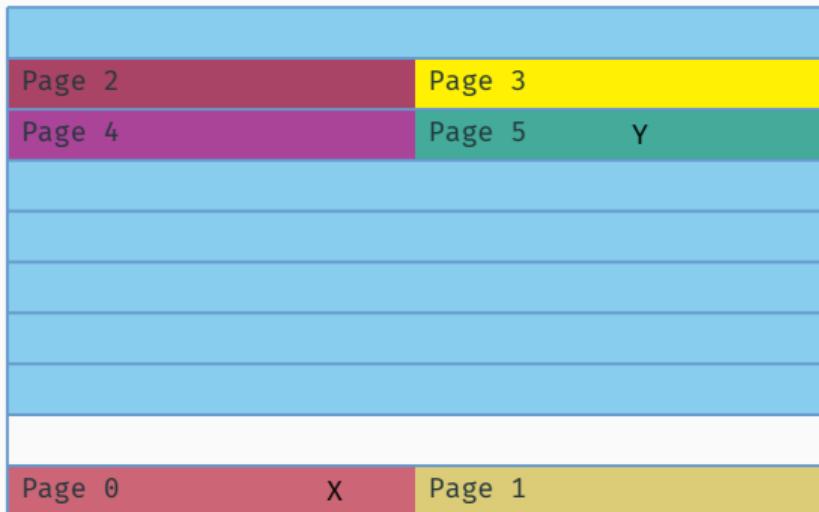
```
1 hammer:  
2     mov eax, X  
3     mov ebx, Y  
4     clflush X  
5     clflush Y  
6     jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

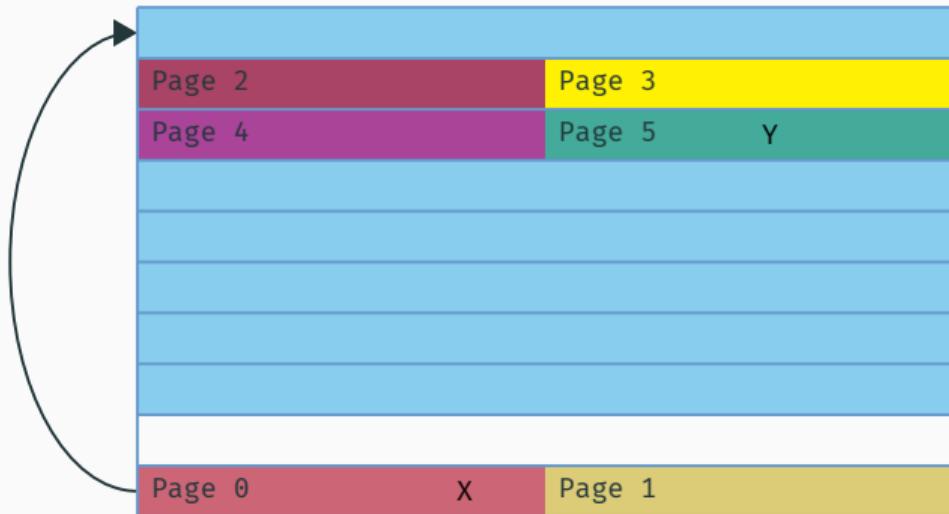
```
1 hammer:  
2     mov eax, X  
3     mov ebx, Y  
4     clflush X  
5     clflush Y  
6     jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

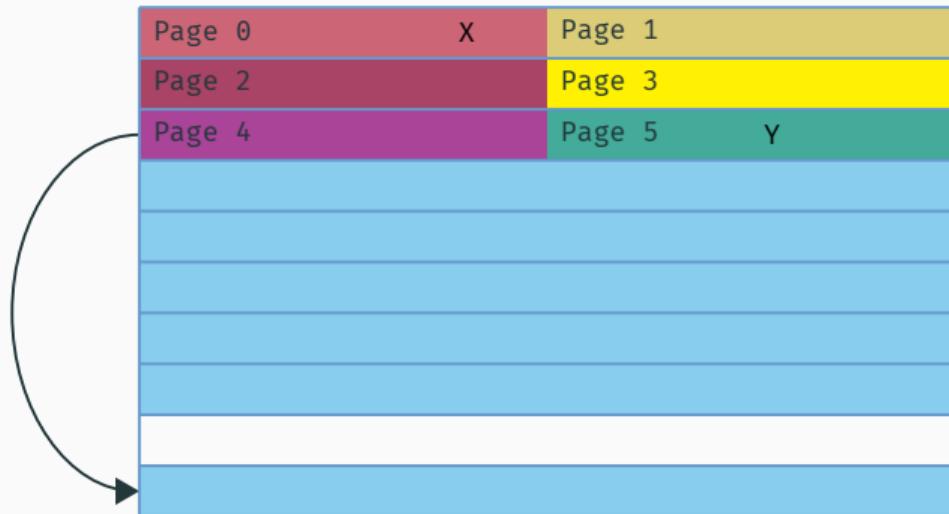
```
1  hammer:  
2      mov eax, X  
3      mov ebx, Y  
4      clflush X  
5      clflush Y  
6      jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

```
1    hammer:  
2        mov eax, X  
3        mov ebx, Y  
4        clflush X  
5        clflush Y  
6        jmp hammer
```



Quellcode von Wikipedia, Abbildung aus den Slides zu [3]

Rowhammer

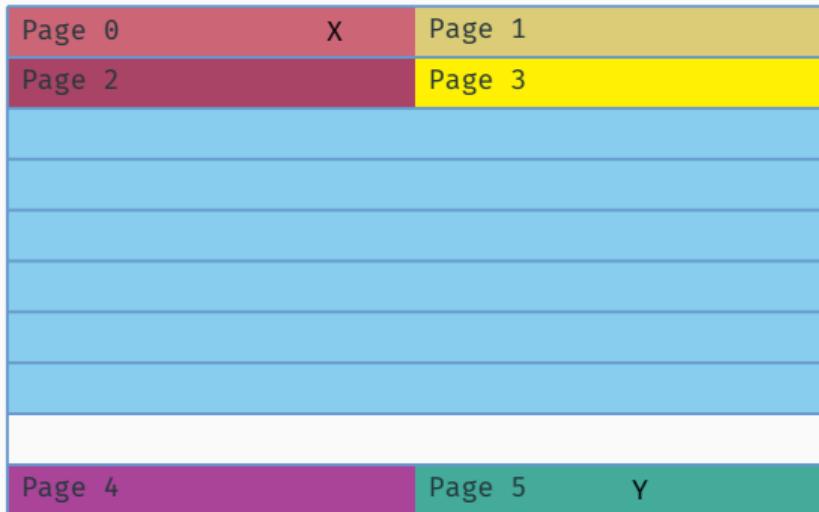
```
1 hammer:  
2     mov eax, X  
3     mov ebx, Y  
4     clflush X  
5     clflush Y  
6     jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

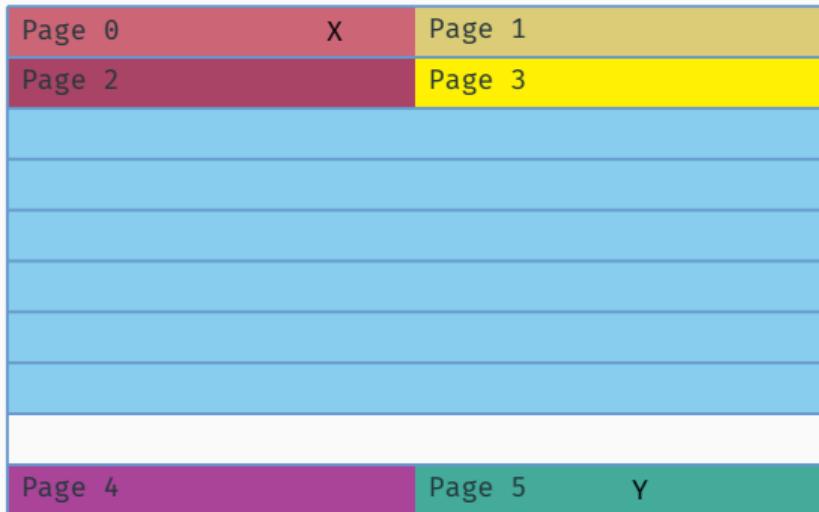
```
1  hammer:
2      mov eax, X
3      mov ebx, Y
4      clflush X
5      clflush Y
6      jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

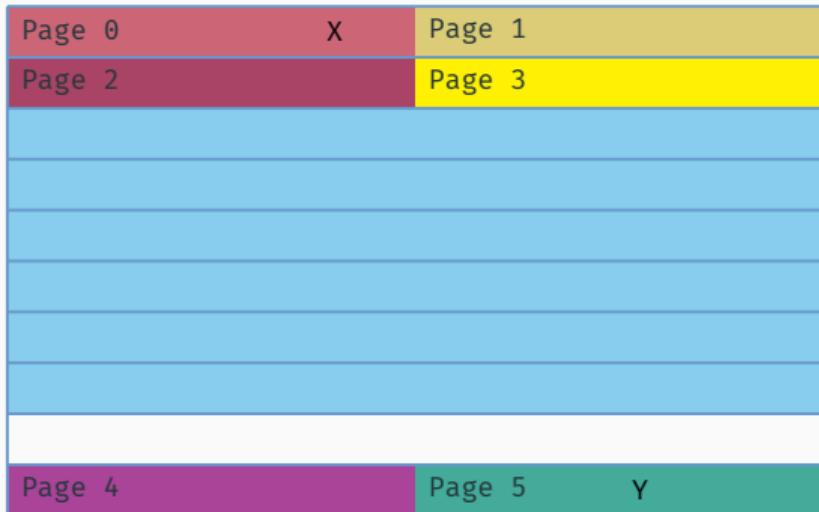
```
1  hammer:
2      mov eax, X
3      mov ebx, Y
4      clflush X
5      clflush Y
6      jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

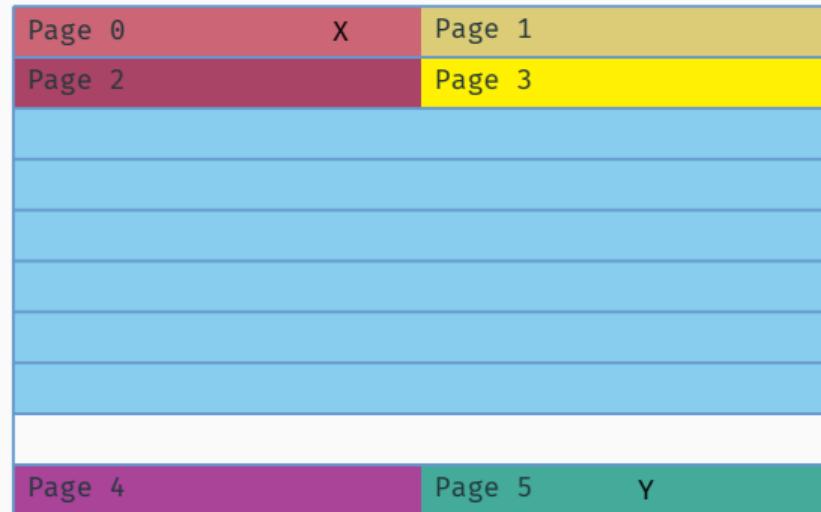
```
1  hammer:  
2      mov eax, X  
3      mov ebx, Y  
4      clflush X  
5      clflush Y  
6      jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

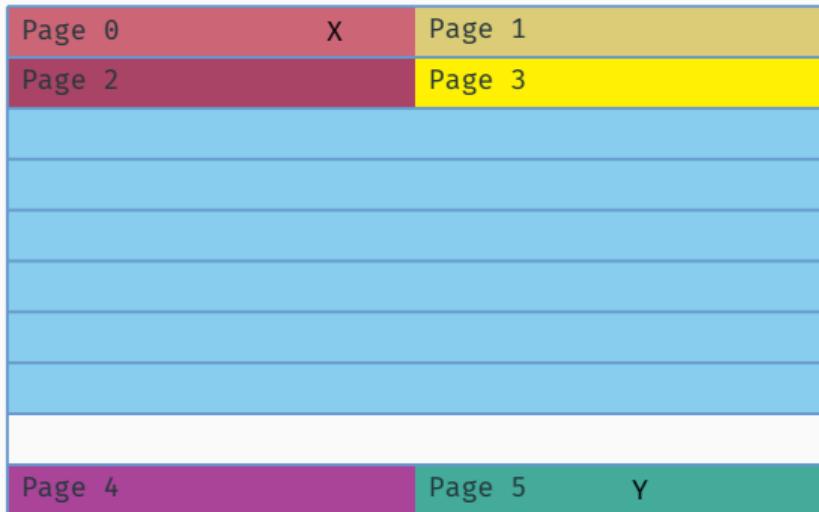
```
1     hammer:  
2     mov eax, X  
3     mov ebx, Y  
4     clflush X  
5     clflush Y  
6     jmp hammer
```



Quellcode von Wikipedia, Abbildung aus den Slides zu [3]

Rowhammer

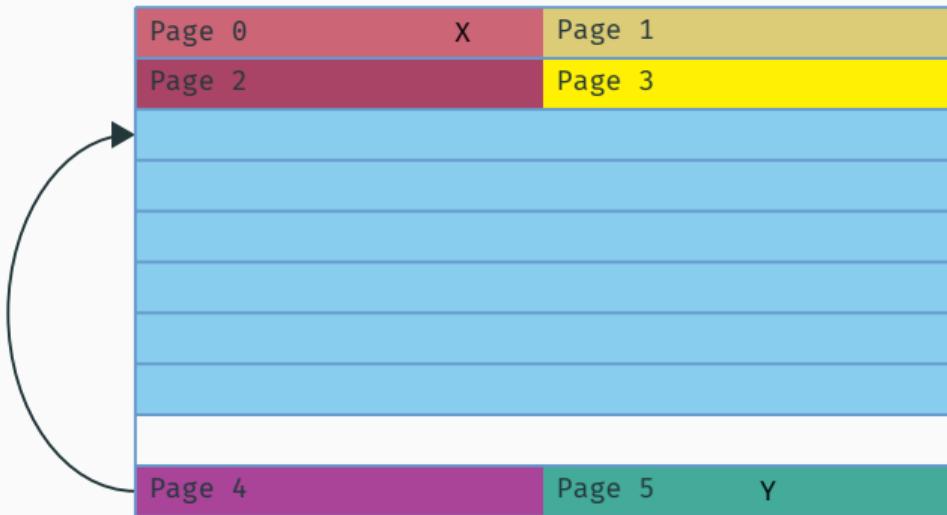
```
1 hammer:  
2     mov eax, X  
3     mov ebx, Y  
4     clflush X  
5     clflush Y  
6     jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

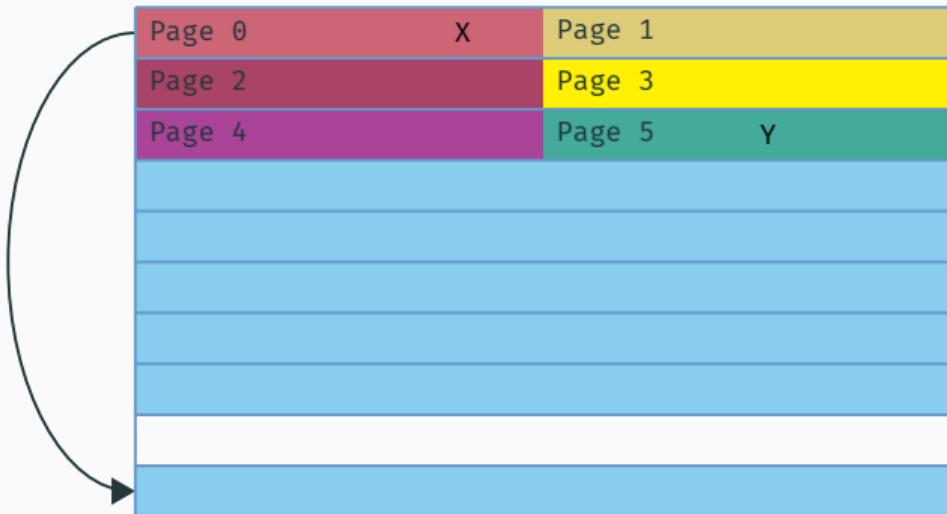
```
1 hammer:  
2     mov eax, X  
3     mov ebx, Y  
4     clflush X  
5     clflush Y  
6     jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

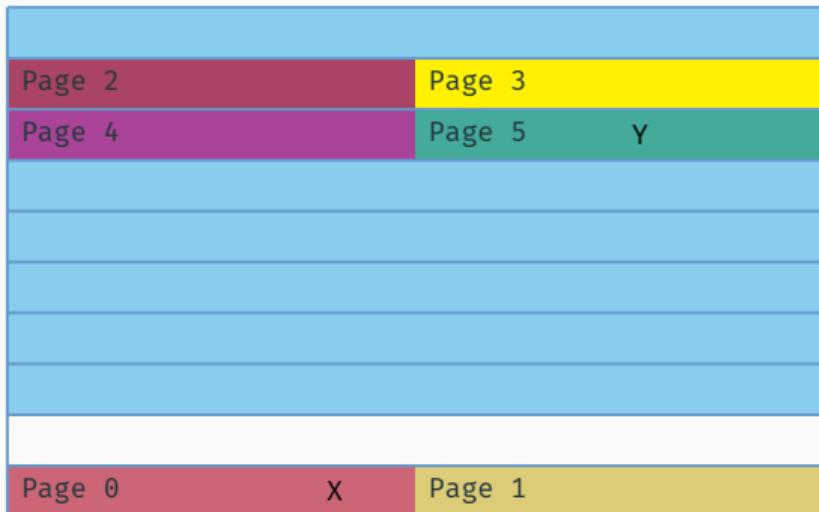
```
1 hammer:  
2     mov eax, X  
3     mov ebx, Y  
4     clflush X  
5     clflush Y  
6     jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

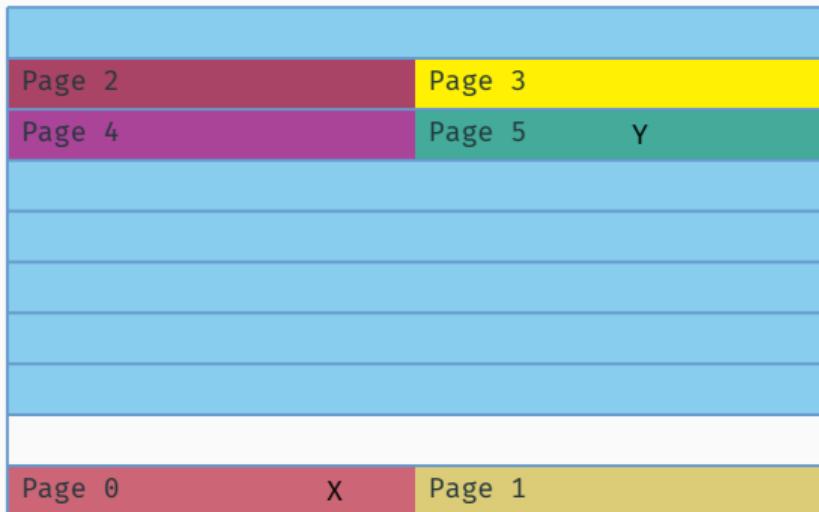
```
1 hammer:  
2     mov eax, X  
3     mov ebx, Y  
4     clflush X  
5     clflush Y  
6     jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

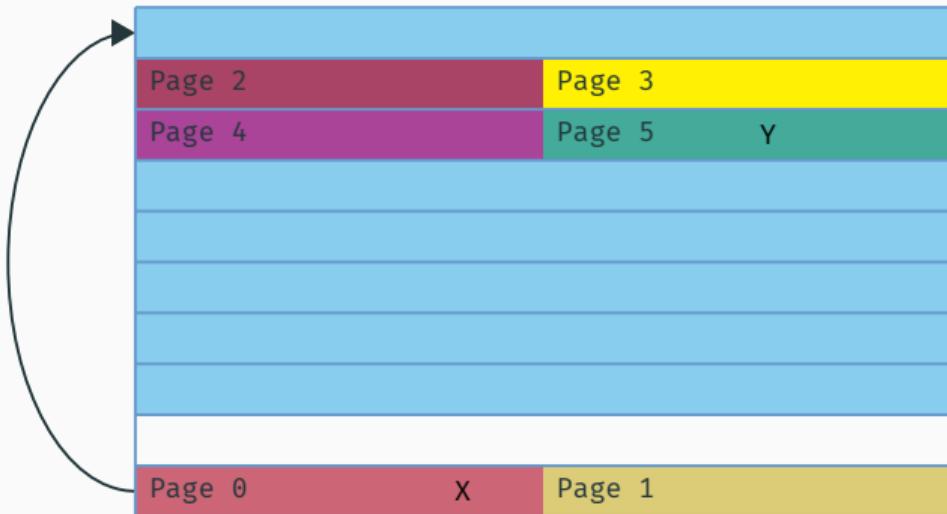
```
1 hammer:  
2     mov eax, X  
3     mov ebx, Y  
4     clflush X  
5     clflush Y  
6     jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

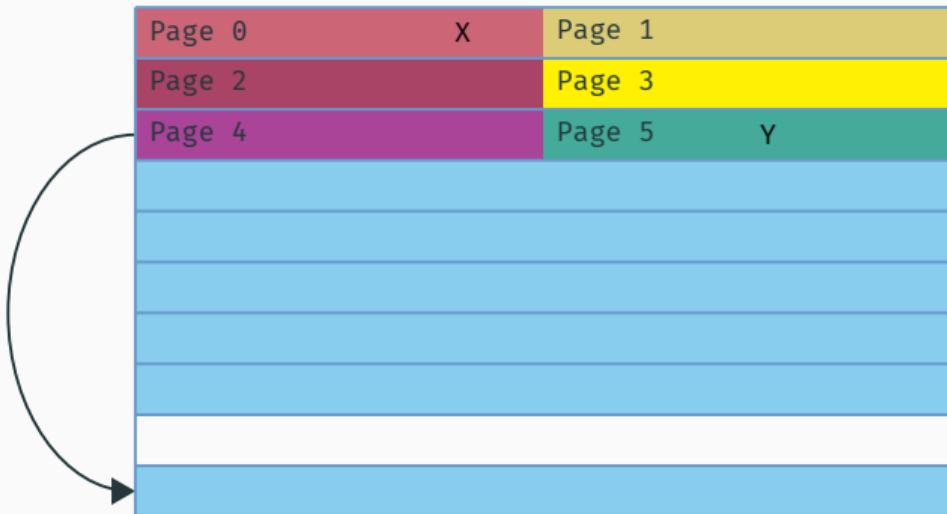
```
1  hammer:  
2      mov eax, X  
3      mov ebx, Y  
4      clflush X  
5      clflush Y  
6      jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

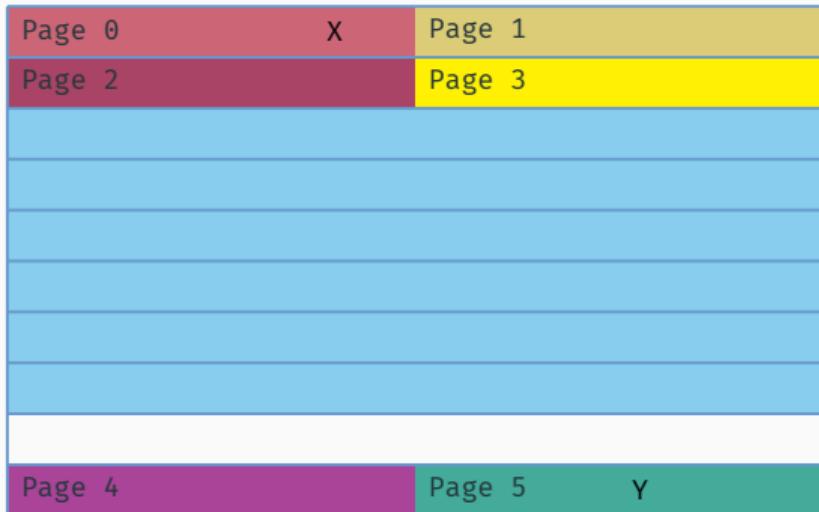
```
1      hammer:  
2      mov eax, X  
3      mov ebx, Y  
4      clflush X  
5      clflush Y  
6      jmp hammer
```



Quellcode von Wikipedia, Abbildung aus den Slides zu [3]

Rowhammer

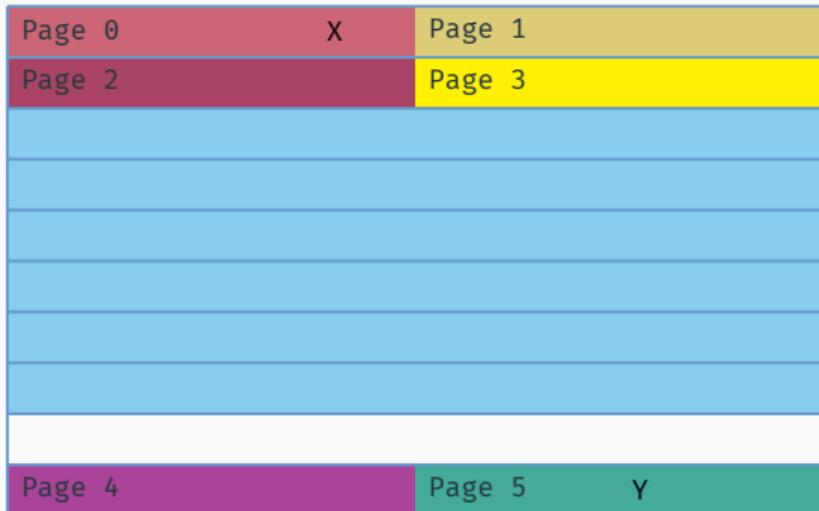
```
1 hammer:  
2     mov eax, X  
3     mov ebx, Y  
4     clflush X  
5     clflush Y  
6     jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

```
1  hammer:
2      mov eax, X
3      mov ebx, Y
4      clflush X
5      clflush Y
6      jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

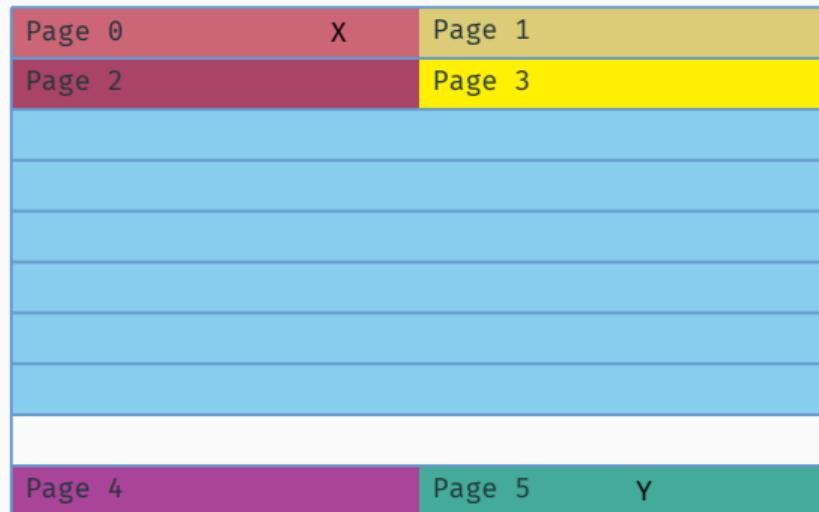
```
1  hammer:
2      mov eax, X
3      mov ebx, Y
4      clflush X
5      clflush Y
6      jmp hammer
```



Quellcode von [Wikipedia](#), Abbildung aus den Slides zu [3]

Rowhammer

```
1    hammer:  
2        mov eax, X  
3        mov ebx, Y  
4        clflush X  
5        clflush Y  
6    jmp hammer
```



Quellcode von Wikipedia, Abbildung aus den Slides zu [3]

Rowhammer

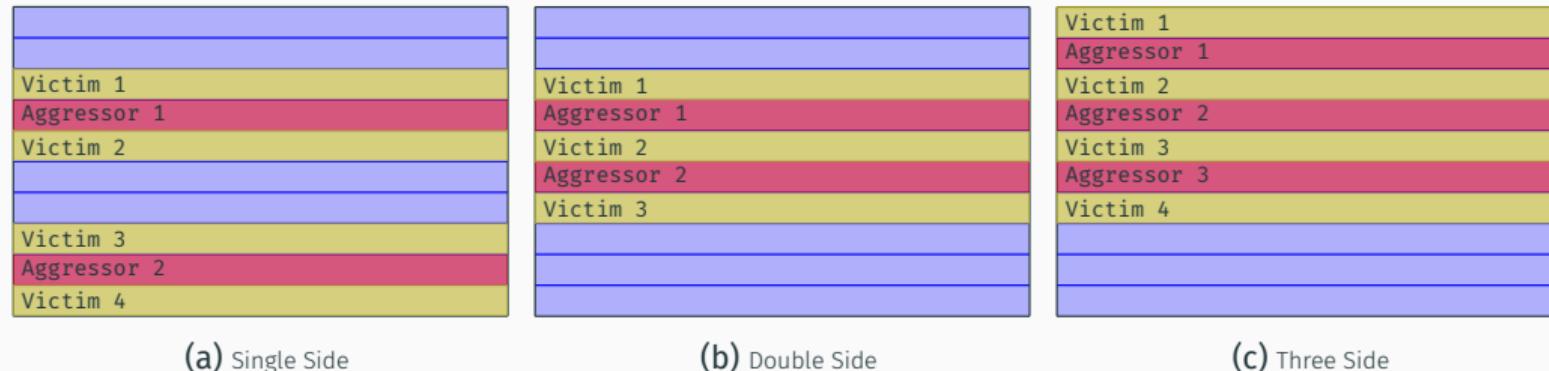


Abbildung 1: Beispiele für Rowhammer-Muster

Abbildung aus den Slides zu [3]

- **Hypothese:** Je höher die Speicherkapazität eines IC bei gleicher Bauart und gleicher Größe ist, desto höher ist die Integrationsdichte
- **Hypothese:** Je höher die Integrationsdichte eines DRAM IC, desto anfälliger ist dieser für Rowhammer [2]

- **Hypothese:** Je höher die Speicherkapazität eines IC bei gleicher Bauart und gleicher Größe ist, desto höher ist die Integrationsdichte
- **Hypothese:** Je höher die Integrationsdichte eines DRAM IC, desto anfälliger ist dieser für Rowhammer [2]
- **Problem:** Integrationsdichte wird weiter steigen, Rowhammer wird schlimmer

- **Hypothese:** Je höher die Speicherkapazität eines IC bei gleicher Bauart und gleicher Größe ist, desto höher ist die Integrationsdichte
- **Hypothese:** Je höher die Integrationsdichte eines DRAM IC, desto anfälliger ist dieser für Rowhammer [2]
- **Problem:** Integrationsdichte wird weiter steigen, Rowhammer wird schlimmer
- **Mitigationen:** Da das Problem nicht gelöst werden kann, kommen verschiedene Mitigationen zum Einsatz:

- **Hypothese:** Je höher die Speicherkapazität eines IC bei gleicher Bauart und gleicher Größe ist, desto höher ist die Integrationsdichte
- **Hypothese:** Je höher die Integrationsdichte eines DRAM IC, desto anfälliger ist dieser für Rowhammer [2]
- **Problem:** Integrationsdichte wird weiter steigen, Rowhammer wird schlimmer
- **Mitigationen:** Da das Problem nicht gelöst werden kann, kommen verschiedene Mitigationen zum Einsatz:
 - **Allgemein:** Verdopplung der Refresh-Rate, Error Correction Code (ECC), ...

- **Hypothese:** Je höher die Speicherkapazität eines IC bei gleicher Bauart und gleicher Größe ist, desto höher ist die Integrationsdichte
- **Hypothese:** Je höher die Integrationsdichte eines DRAM IC, desto anfälliger ist dieser für Rowhammer [2]
- **Problem:** Integrationsdichte wird weiter steigen, Rowhammer wird schlimmer
- **Mitigationen:** Da das Problem nicht gelöst werden kann, kommen verschiedene Mitigationen zum Einsatz:
 - **Allgemein:** Verdopplung der Refresh-Rate, Error Correction Code (ECC), ...
 - **Mustererkennung:** Target Row Refresh (TRR) und pseudo-TRR (pTRR), ...

- **Problem bei allgemeinen Mitigationen:** Hoher Overhead für effektive Mitigationen erforderlich (Overhead deutlich spürbar)

- **Problem bei allgemeinen Mitigationen:** Hoher Overhead für effektive Mitigationen erforderlich (Overhead deutlich spürbar)
- **Problem bei Mustererkennung:** Muster müssen von „normalen“ Zugriffen unterschieden werden, müssen also bekannt sein

- **Problem bei allgemeinen Mitigationen:** Hoher Overhead für effektive Mitigationen erforderlich (Overhead deutlich spürbar)
- **Problem bei Mustererkennung:** Muster müssen von „normalen“ Zugriffen unterschieden werden, müssen also bekannt sein
- **Offene Forschungsfrage:** Finden von allgemeinen Mitigationen mit geringen/tolerierbaren Overhead

Angriffe auf KSM

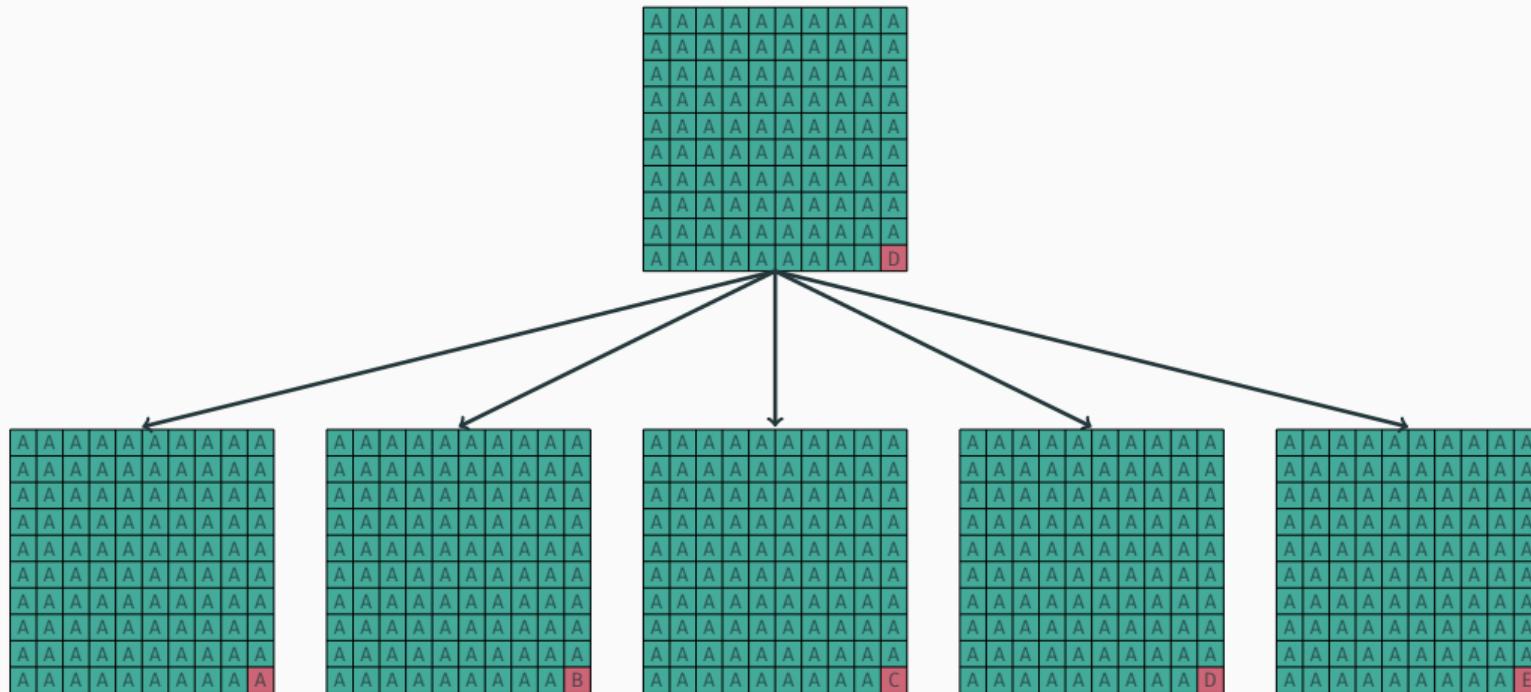
- Copy-On-Write Policy für gemergete Seiten
 - Schreiben auf einer deduplizierten Seite ist deutlich langsamer als auf einer nicht deduplizierten Seite

- Copy-On-Write Policy für gemergete Seiten
 - Schreiben auf einer deduplizierten Seite ist deutlich langsamer als auf einer nicht deduplizierten Seite
- **Idee:** Ausnutzen von KSM zum Raten von unbekannten Inhalten [1]

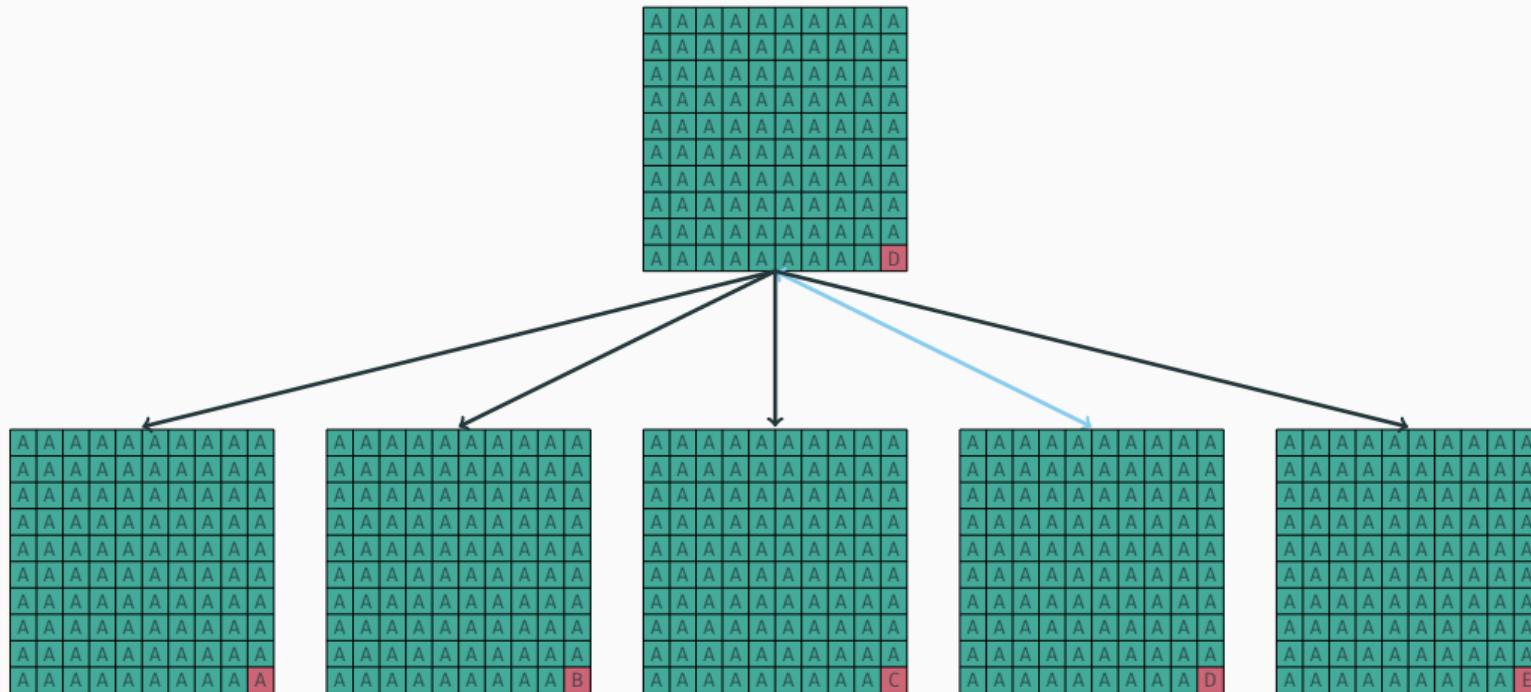
Information leaks

A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	D

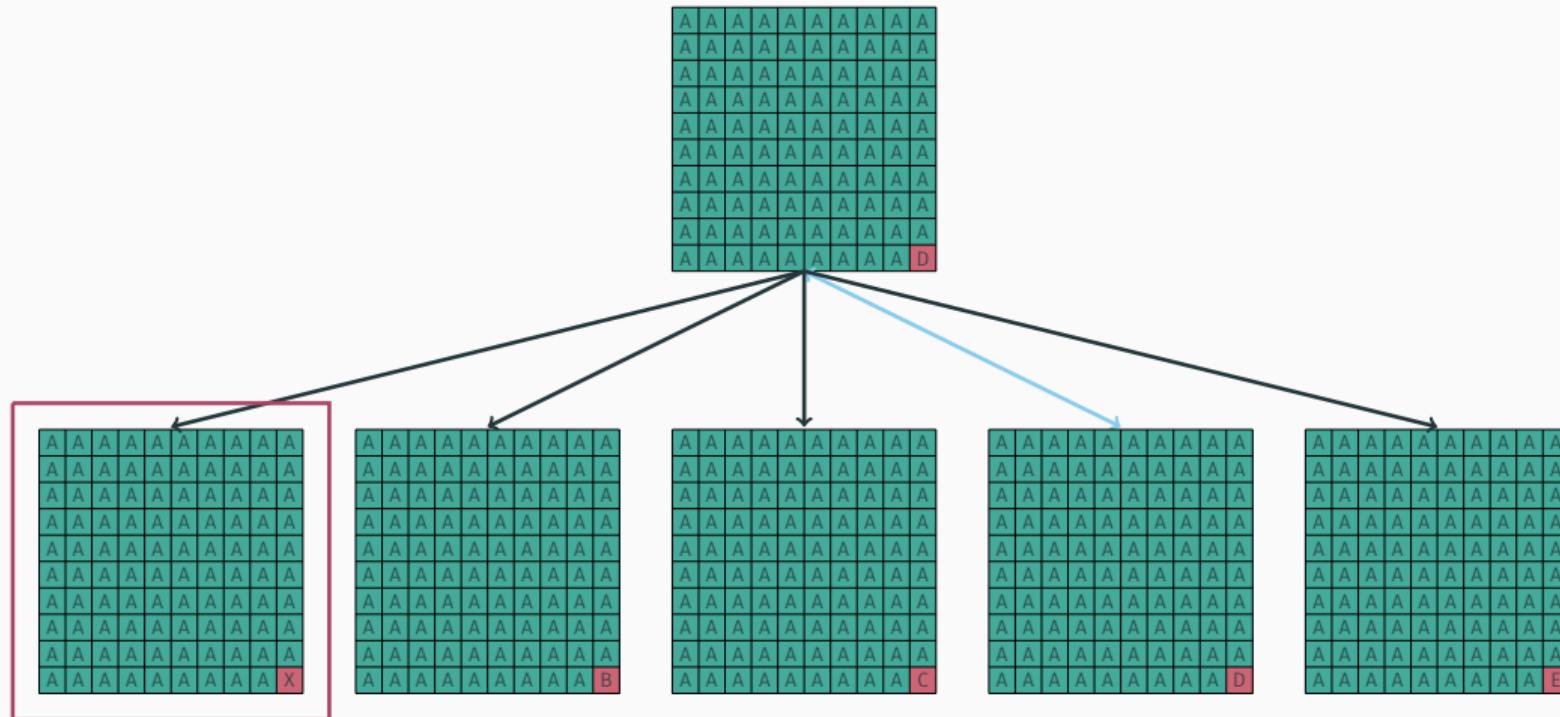
Information leaks



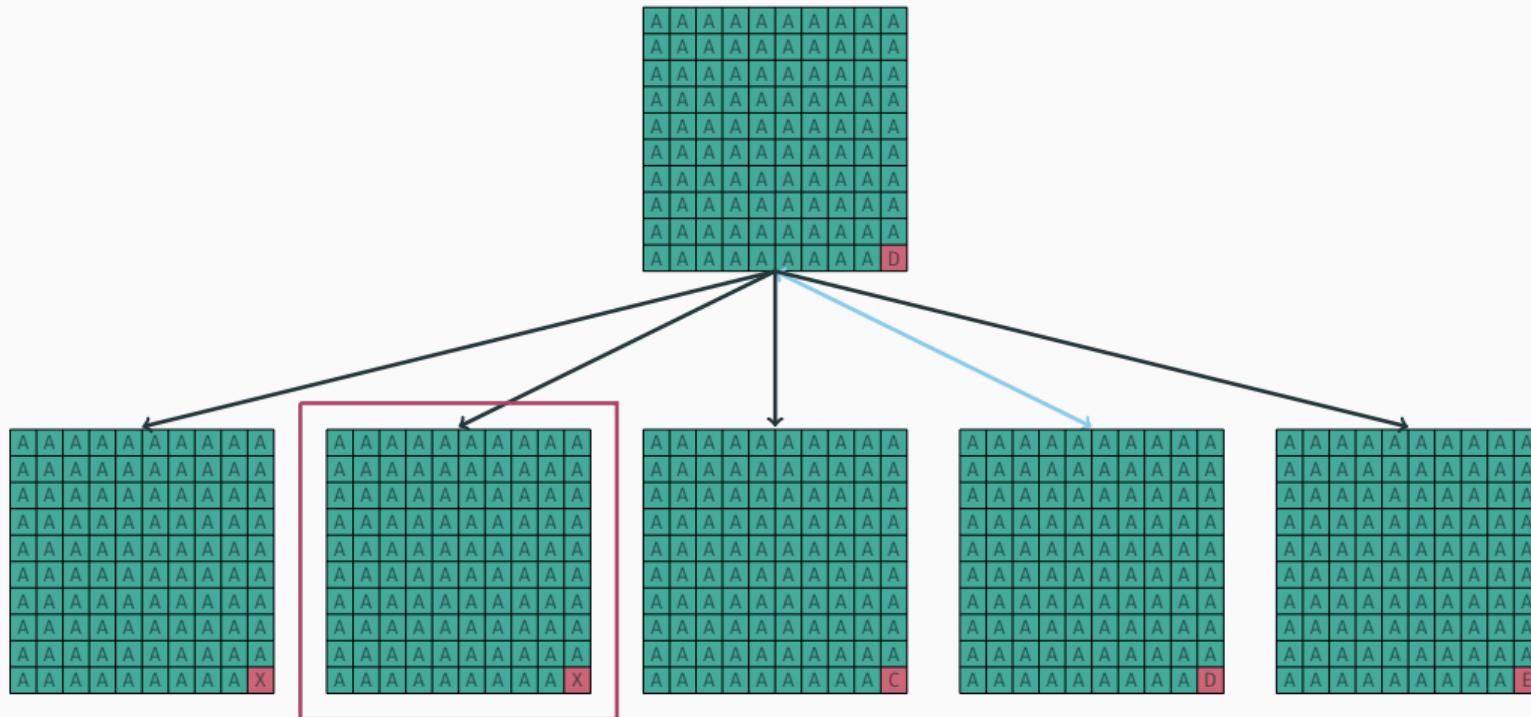
Information leaks



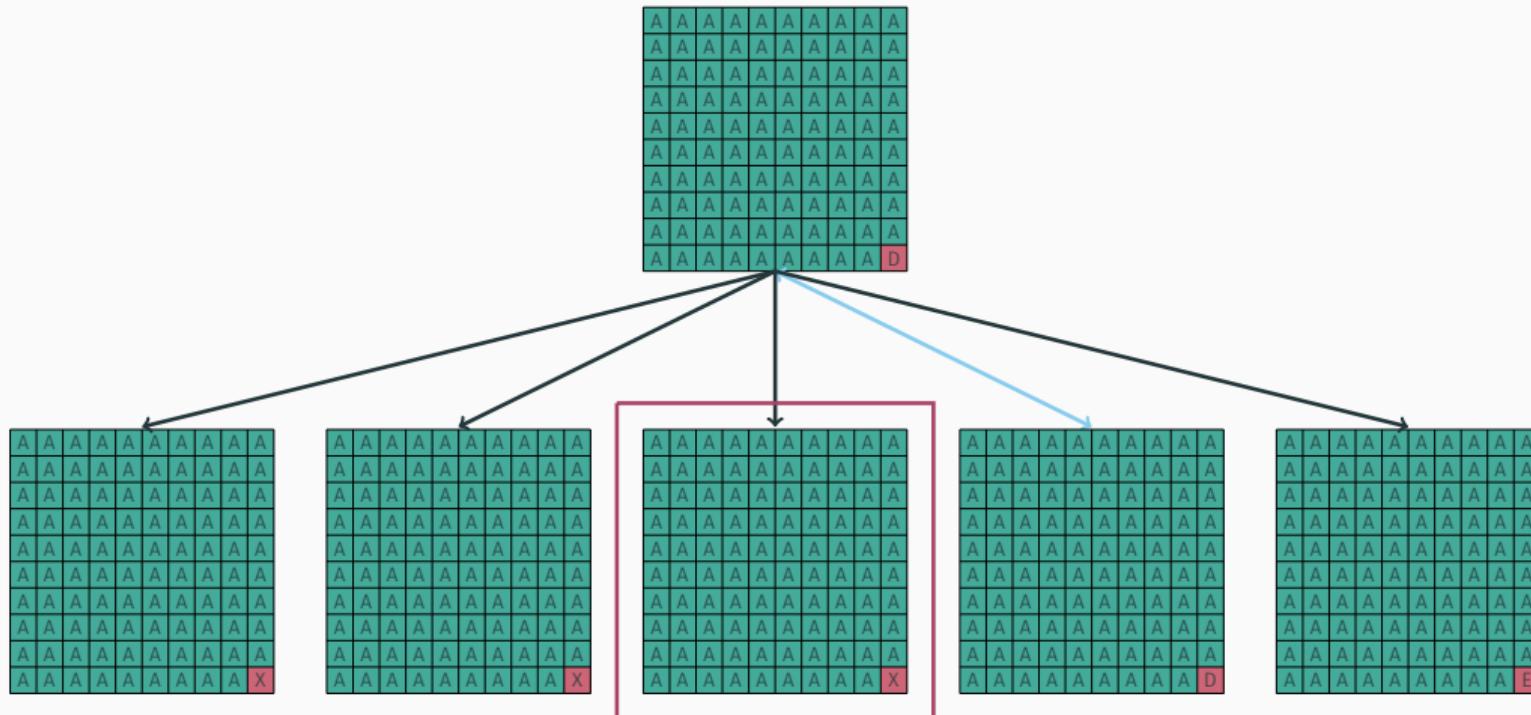
Information leaks



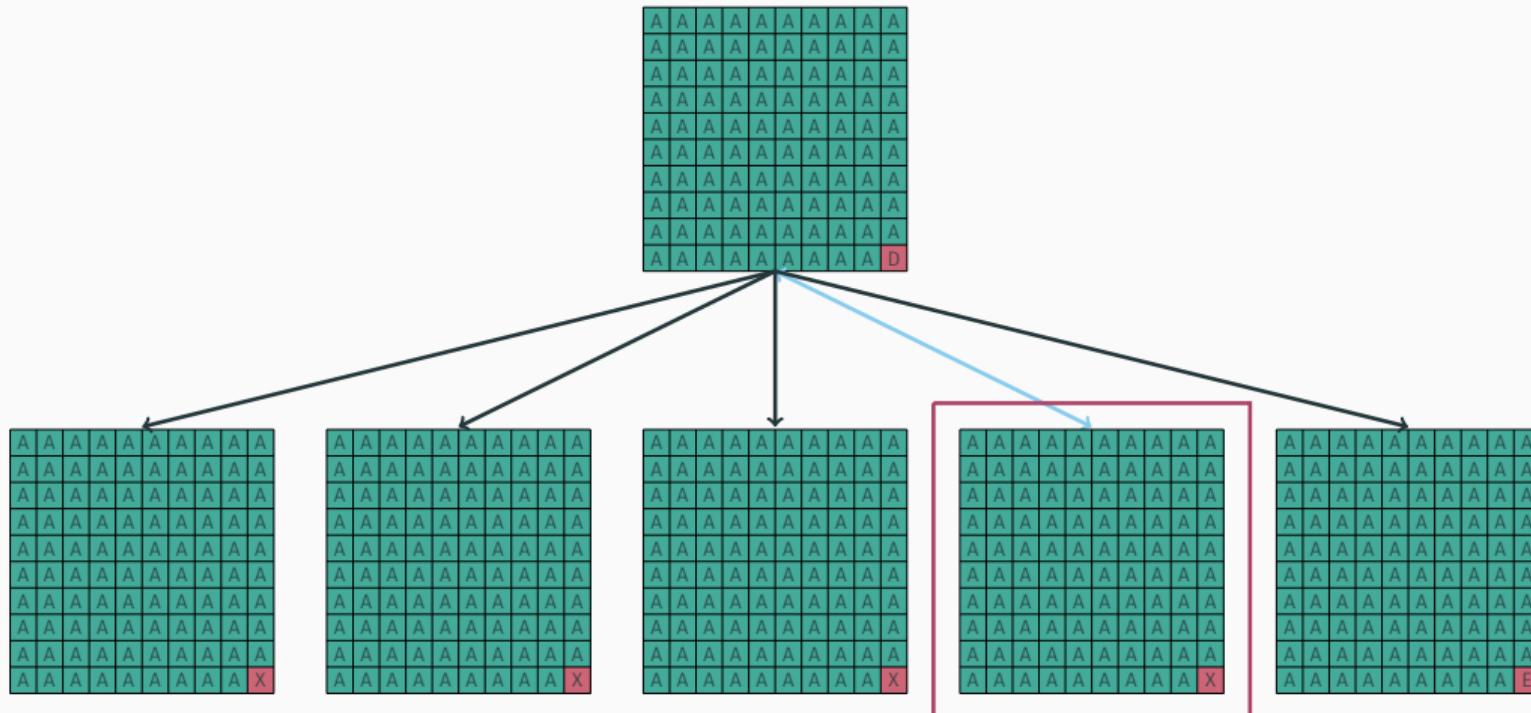
Information leaks



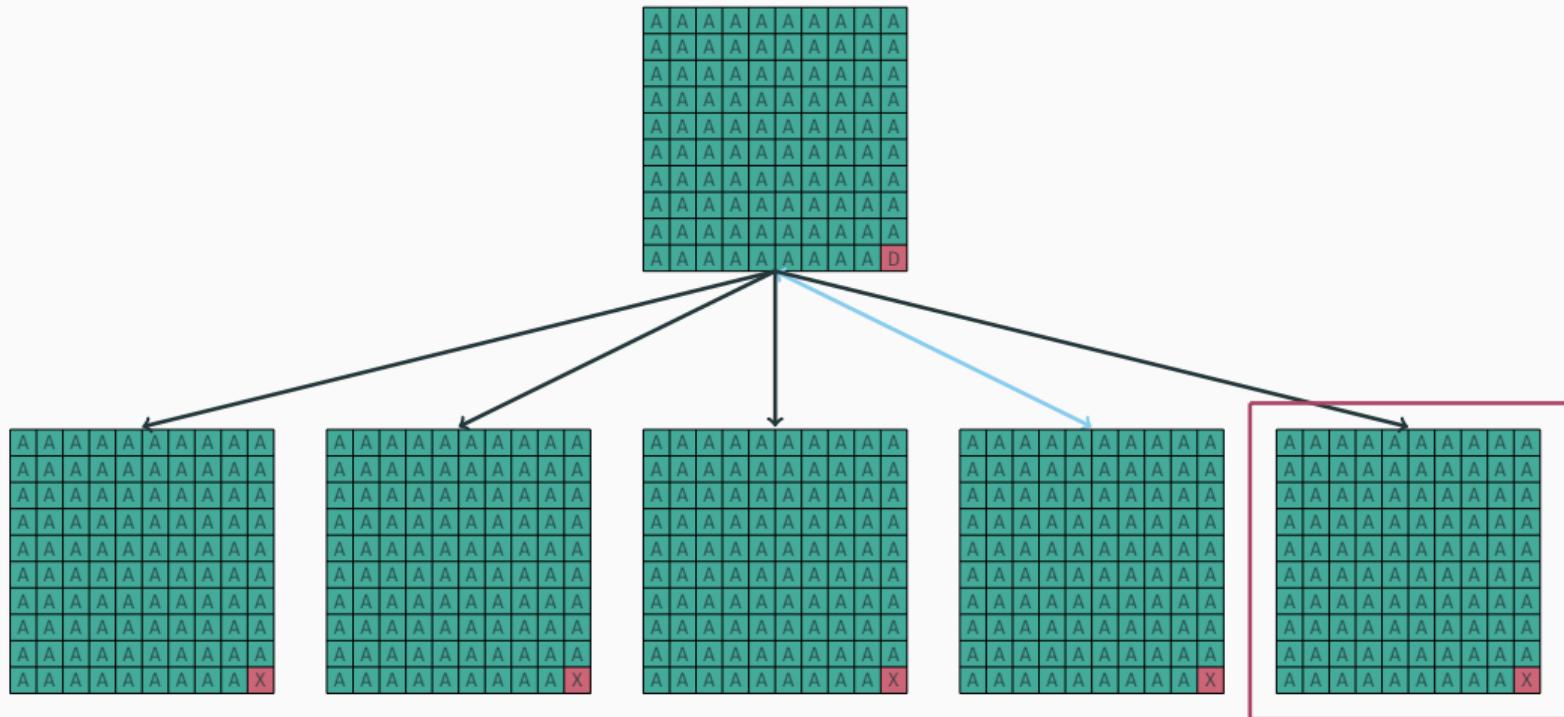
Information leaks



Information leaks



Information leaks



- Kommunikation zwischen Prozessen ist zentraler Bestandteil der Informationstechnik

- Kommunikation zwischen Prozessen ist zentraler Bestandteil der Informationstechnik
- In manchen Fällen ist die Kommunikation unerwünscht und wird unterbunden (z.B. Firewall)

- Kommunikation zwischen Prozessen ist zentraler Bestandteil der Informationstechnik
- In manchen Fällen ist die Kommunikation unerwünscht und wird unterbunden (z.B. Firewall)
- Covert Channels ermöglichen Kommunikation ohne Nutzung „offizieller“ Kommunikationswege, umgehen damit entsprechende Einschränkungen

- Copy-On-Write Policy für gemergete Seiten
 - Schreiben auf einer deduplizierten Seite ist deutlich langsamer als auf einer nicht deduplizierten Seite

- Copy-On-Write Policy für gemergete Seiten
 - Schreiben auf einer deduplizierten Seite ist deutlich langsamer als auf einer nicht deduplizierten Seite
- **Idee:** Ausnutzen von KSM zur verdeckten Kommunikation [1]

Covert Channel

A 10x10 grid of 100 squares, all colored green except for the bottom-right square which is red.

Covert Channel

A 10x10 grid of green squares. The bottom-right square is red, while all other squares are green.

A 10x10 grid of 100 green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The letters 'D' are placed in the following positions: Row 1: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Row 2: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Row 3: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Row 4: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Row 5: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Row 6: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Row 7: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Row 8: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Row 9: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Row 10: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. The letter 'D' is also present in the bottom right corner of the grid.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

Covert Channel

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 7x7 grid of green squares. The bottom right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The bottom right square is highlighted with a red border. The grid is composed of 100 individual squares, arranged in 10 rows and 10 columns.

A 10x10 grid of blue squares. The squares are arranged in 10 rows and 10 columns. The bottom-right square is colored red, while all other squares are blue. This represents a 10x10 matrix with a single element highlighted.

A 10x10 grid of 100 green 'C' characters. The bottom right cell is colored red.

A 10x10 grid of green squares. The bottom-right square is red and contains the letter 'A'.

A 10x10 grid of green squares. The bottom-right square is colored red and contains the letter 'A'.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of 100 F-shaped blocks, with the bottom-right block colored red.

Covert Channel

Covert Channel

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The bottom-right square is colored red and contains the letter 'B'.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of 100 blue squares. The bottom right square is red, and the bottom row and rightmost column are outlined in red.

A 10x10 grid of squares. Most of the squares are a light blue color. In the bottom right corner, there is a single red square. The grid is composed of 100 individual squares.

A 10x10 grid of green squares. The bottom-right square is colored red and contains the letter 'A'.

A 10x10 grid of teal squares. The squares are arranged in 10 rows and 10 columns. The bottom-right square is colored red and contains the letter 'B'.

A 10x10 grid of green squares. The squares are arranged in 10 rows and 10 columns. The bottom-right square is colored red and contains the letter 'B'.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of 100 small red squares. The bottom-right square is colored blue, while all other squares are red.

Covert Channel

A 10x10 grid of 100 green squares. The last square in the bottom right corner is colored red.

A 10x10 grid of 100 green squares. The last square in the bottom right corner is colored red.

A 10x10 grid of green squares. The bottom right square is red, while all other squares are green.

A 10x10 grid of green squares. The bottom right square is red and contains the letter 'A'.

A 10x10 grid of green squares. The bottom-right square is red. The rest of the grid is filled with green squares.

A 10x10 grid of 100 squares, each filled with a solid blue color. The squares are arranged in a perfect square pattern. In the bottom right corner, there is a single square that is colored red, making it stand out from the rest of the grid.

A 10x10 grid of 100 green squares. The bottom right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

Covert Channel

A 10x10 grid of green squares. The bottom-right square is colored red and contains the letter 'A'.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares, each containing a black 'A'. The grid is enclosed in a black border.

A 10x10 grid of 100 blue squares. The bottom-right square is red, while all other squares are blue.

A 10x10 grid of green squares. The bottom right square is colored red and contains the letter 'A'.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The bottom-right square is red, while all other squares are green.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

Covert Channel

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The bottom-right square is colored red and contains the letter 'B'.

A 10x10 grid of squares. The squares are colored in a repeating pattern of red and black. The pattern starts with a red square in the top-left corner, followed by a black square, then a red square, and so on. This creates a checkerboard-like effect across the entire grid.

A 10x10 grid of teal squares. The bottom-right square is red, while all other squares are teal.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

Covert Channel

A 10x10 grid of green squares. Each square contains a white letter 'D', except for the bottom-right square which is red and contains a white 'D'.

A 10x10 grid of green squares. The bottom right square is colored red, while all other squares are green.

A 10x10 grid of green squares. Each square contains a white letter 'D', except for the bottom-right square which is red and contains a white 'D'.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The bottom-right square is red, while all other squares are green.

A 10x10 grid of letters. All letters are 'E' except for the bottom-right corner, which is a red 'B'.

A 10x10 grid of 100 green squares. The last square in the bottom right corner is colored red.

Covert Channel

A 10x10 grid of green squares. Each square contains a white letter 'D', except for the bottom-right square which is red and contains a white 'D'.

A 10x10 grid of 100 green squares. The last square in the bottom right corner is colored red.

A 10x10 grid of blue squares. The squares are arranged in a 10x10 pattern, with each square being a small blue square. The grid is positioned in the center of the page. In the bottom right corner of the grid, there is a red 'X' mark.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of 100 green squares. The last square in the bottom right corner is colored red.

A 10x10 grid of green squares. The bottom-right square is red, while all other squares are green.

Covert Channel

A 10x10 grid of green squares. Each square contains a white letter 'D', except for the bottom-right square which is red and contains a white 'D'.

A 10x10 grid of 100 green squares. The last square in the bottom right corner is colored red.

A 10x10 grid of green squares. The bottom-right square is red and contains a black 'X'.

A 10x10 grid of squares. All squares are green except for the bottom-right square, which is red. The grid is composed of 100 individual squares arranged in 10 rows and 10 columns.

A 10x10 grid of teal squares. The squares are arranged in 10 rows and 10 columns. A red square with the letter 'A' is located in the bottom right corner of the grid.

A 10x10 grid of green squares. The squares are arranged in 10 rows and 10 columns. A red 'X' is located in the bottom right corner of the grid.

A 10x10 grid of green squares. The bottom-right square is red, while all other squares are green.

A 10x10 grid of letters. All letters are 'E' except for the bottom-right corner, which is a 'B'.

A 10x10 grid of 100 green squares. The last square in the bottom right corner is colored red.

Covert Channel

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The bottom-right square is colored red and contains the letter 'B'.

A 10x10 grid of green squares. The bottom-right square is highlighted with a red 'X'.

A 10x10 grid of 100 green squares. The bottom right square is red, and the bottom row and rightmost column are outlined in red.

A 10x10 grid of green squares. The bottom right square is red and contains a black 'X'.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of 100 green squares. The bottom right square is red, representing the element at index 99 in a 1D array.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The bottom-right square is colored orange, while all other squares are green.

A 10x10 grid of green squares. The bottom right square is colored red and contains the letter 'B'.

Covert Channel

A 10x10 grid of green squares. Each square contains a white letter 'D', except for the bottom-right square which is red and contains a white 'D'.

A 10x10 grid of letters. The letters are mostly 'F', arranged in a pattern where they are grouped in sets of four. The bottom-right cell contains a 'B'.

A 10x10 grid of green squares. The bottom right square is colored red, while all other squares are green.

A 10x10 grid of squares. All squares are green except for the bottom-right square, which is red. The grid is composed of 100 individual squares arranged in 10 rows and 10 columns.

A 10x10 grid of green squares. The bottom-right square is colored red and contains a black 'X'.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 6x6 grid of green squares. Each square contains the letter 'F', except for the bottom-right square which is red and also contains the letter 'F'.

A 10x10 grid of green squares. The squares are arranged in 10 rows and 10 columns. A red 'X' is located in the bottom right corner of the grid.

A 10x10 grid of green squares. The bottom-right square is colored red and contains a black 'X' symbol.

A 10x10 grid of letters. All letters are 'E' except for the bottom-right corner, which is a red 'B'.

A 10x10 grid of 100 small green squares. The bottom-right square is colored red, while all other squares are green.

Covert Channel

A 10x10 grid of green squares. The bottom right square is colored red, while all other squares are green.

A 10x10 grid of colored squares. The colors are arranged in a repeating pattern of red, green, blue, and yellow. The pattern repeats every 2 columns and every 3 rows. The bottom-right square is colored pink.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of 100 cells, each containing a black letter 'F'. The bottom-right cell is highlighted in red.

A 10x10 grid of green squares. The bottom-right square is colored red and features a black 'X' symbol.

A 10x10 grid of 100 squares. All squares are green except for the bottom-right square, which is red. The bottom row and the rightmost column of squares are also outlined in red, while the rest of the grid is solid green.

A 10x10 grid of green squares, each containing a white letter 'E', representing a 10x10 matrix.

Covert Channel

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares, each containing a black 'A'. The grid is enclosed in a black border.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The bottom-right square is colored red and contains a black 'X'.

A 10x10 grid of green squares. The squares are arranged in 10 rows and 10 columns. A red 'X' is located in the bottom right corner of the grid.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

Covert Channel

A 10x10 grid of green squares. The bottom right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The bottom-right square is colored red and features a black 'X' symbol.

A 10x10 grid of 100 squares, all colored green except for the bottom-right square which is red.

A 10x10 grid of green squares. The squares are arranged in 10 rows and 10 columns. A red 'X' is located in the bottom right corner of the grid.

Covert Channel

A 10x10 grid of green squares. The bottom right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of 100 cells, each containing the letter 'F'. The bottom-right cell is colored red.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of teal squares. The bottom-right square is red and contains the letter 'A'.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The squares are arranged in 10 rows and 10 columns. A red 'X' is located in the bottom right corner of the grid.

A 10x10 grid of green squares. The bottom-right square is red, while all other squares are green.

Covert Channel

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares, each containing a black 'A'. The grid is enclosed in a black border.

A 10x10 grid of green squares. The bottom right square is red and contains a black 'X'.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The bottom-right square is red and contains a black 'X'.

A 10x10 grid of 100 green squares. The bottom right square is red, and the bottom row and rightmost column are outlined in red.

A 10x10 grid of green squares. The squares are arranged in 10 rows and 10 columns. A red 'X' is located in the bottom right corner of the grid.

Covert Channel

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The bottom right square is colored red, while all other squares are green.

A 10x10 grid of 100 squares. All squares are green except for the bottom-right square, which is red. The bottom row and the rightmost column of squares are also outlined in red, while the rest of the grid is solid green.

A 10x10 grid of green squares. The bottom-right square is colored pink, while all other squares are green.

A 10x10 grid of green squares. The squares are arranged in 10 rows and 10 columns. A red 'X' is located in the bottom right corner of the grid.

A 10x10 grid of green squares. The bottom-right square is highlighted with a red 'X'.

Covert Channel

A 10x10 grid of green squares. The bottom right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The bottom-right square is colored red, while all other squares are green.

A 10x10 grid of green squares. The squares are arranged in 10 rows and 10 columns. A red 'X' is located in the bottom right corner of the grid.

- Dedupliizierte Pages liegen physisch an der gleichen Stelle
- Auftretende Speicherfehler betreffen alle Seiten, die physisch an der gleichen Stelle liegen

- Dedupliizierte Pages liegen physisch an der gleichen Stelle
- Auftretende Speicherfehler betreffen alle Seiten, die physisch an der gleichen Stelle liegen
- **Idee:** Provokieren von Speicherfehlern an bestimmten Stellen (offsets) in deduplizierten Seiten

- Dedupliizierte Pages liegen physisch an der gleichen Stelle
- Auftretende Speicherfehler betreffen alle Seiten, die physisch an der gleichen Stelle liegen
- **Idee:** Provokieren von Speicherfehlern an bestimmten Stellen (offsets) in deduplizierten Seiten
- **Konsequenz:** Umgehen der CoW Policy, effektives Schreiben beliebiger Pages (mit einigen Einschränkungen) [6]

- **Annahme:** Bit Flips sind stabil, d.h. treten bei erneuten Zugriffen wieder an den gleichen Stellen auf
- **Annahme:** Der Inhalt der Page, sowie der Offset, an dem ein Bit Flip auftreten soll und dessen Richtung, sind bekannt

Flip Feng Shui (FFS)

- Suchen nach Speicherstellen, an denen Bits an der richtigen Stelle (Offset) flippen

- Suchen nach Speicherstellen, an denen Bits an der richtigen Stelle (Offset) flippen
- Schreiben des Inhalts der (bekannten) Page an die Stelle, bei der der gewünschte Flip auftritt
- Schreiben des Inhalts der (bekannten) Page an eine andere Stelle, sodass beide Seiten an der anfälligen Speicherstelle gemerged werden
- Warten, bis KSM die Seiten gemerged hat
- Erstellen der Zielseite triggern (z.B. Aufrufen eines Programms, etc.)
- Warten bis die Seiten gemerged wurden

Flip Feng Shui (FFS)

- Suchen nach Speicherstellen, an denen Bits an der richtigen Stelle (Offset) flippen
- Schreiben des Inhalts der (bekannten) Page an die Stelle, bei der der gewünschte Flip auftritt
- Schreiben des Inhalts der (bekannten) Page an eine andere Stelle, sodass beide Seiten an der anfälligen Speicherstelle gemerged werden
- Warten, bis KSM die Seiten gemerged hat
- Erstellen der Zielseite triggern (z.B. Aufrufen eines Programms, etc.)
- Warten bis die Seiten gemerged wurden
- Erneutes Triggern des Bit Flips

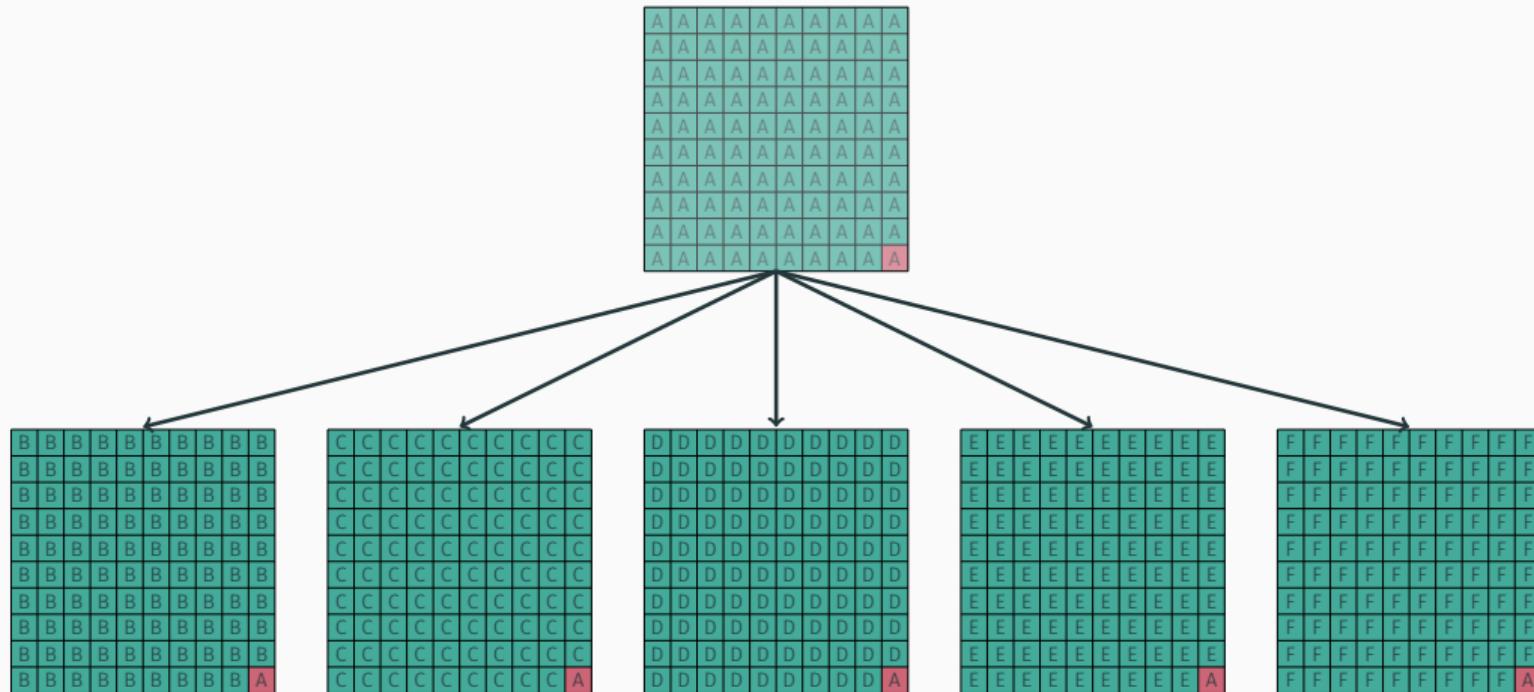
Flip Feng Shui (FFS) – Scannen des Speichers

A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A

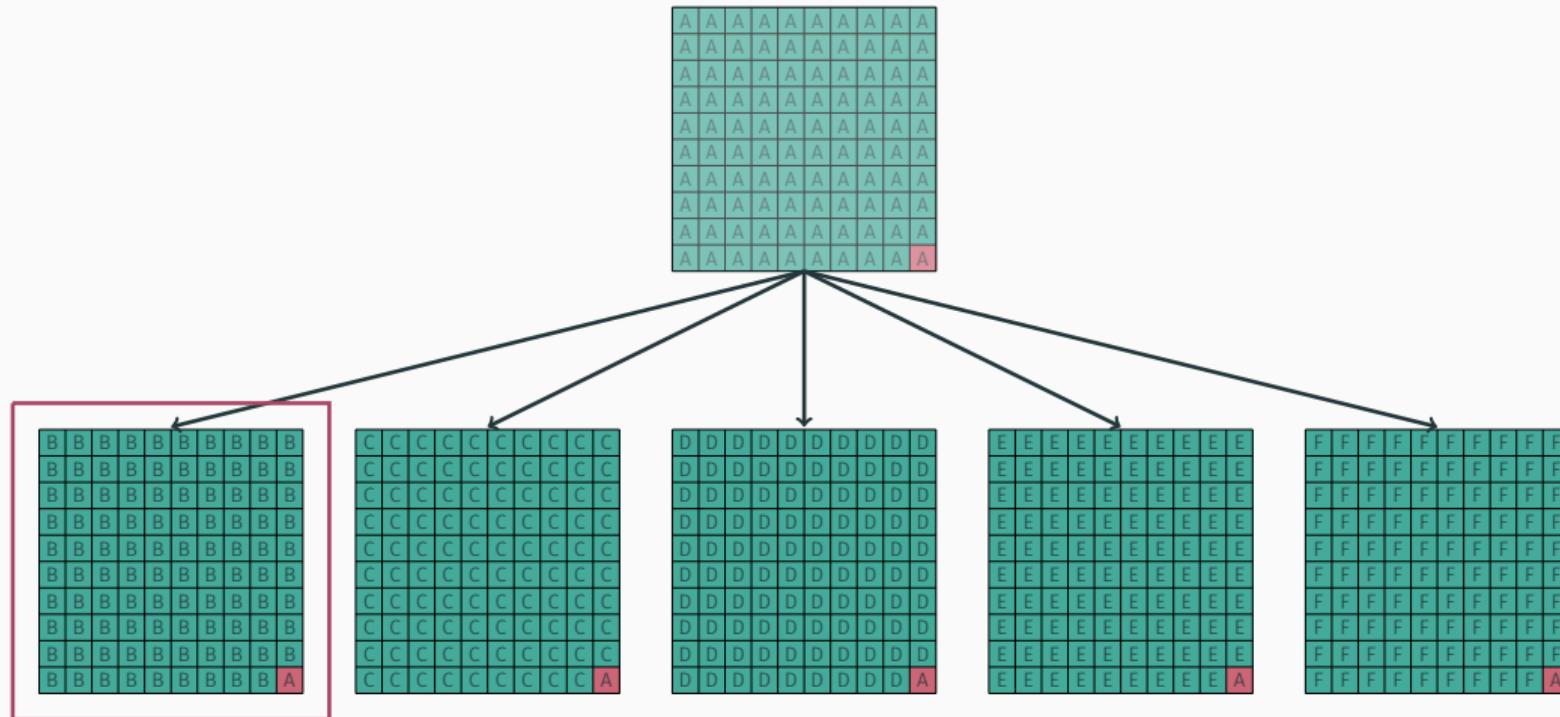
Binäre Darstellung von Zeichen

Zeichen	Hexadezimal	Binär
@	0x40	0010 0000
A	0x41	0010 0001
B	0x42	0010 0010
C	0x43	0010 0011
D	0x44	0010 0100
E	0x45	0010 0101
F	0x46	0010 0110
G	0x47	0010 0111

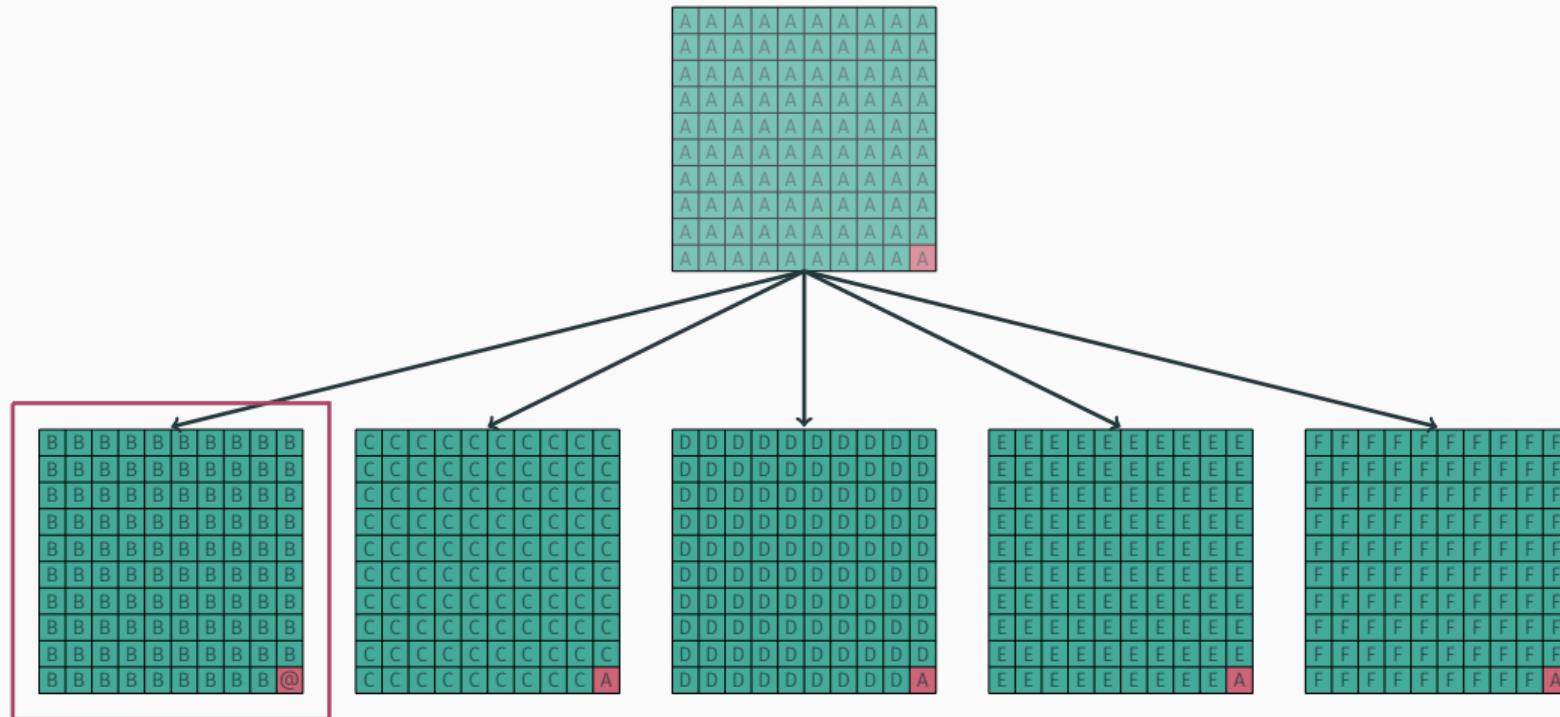
Flip Feng Shui (FFS) – Scannen des Speichers



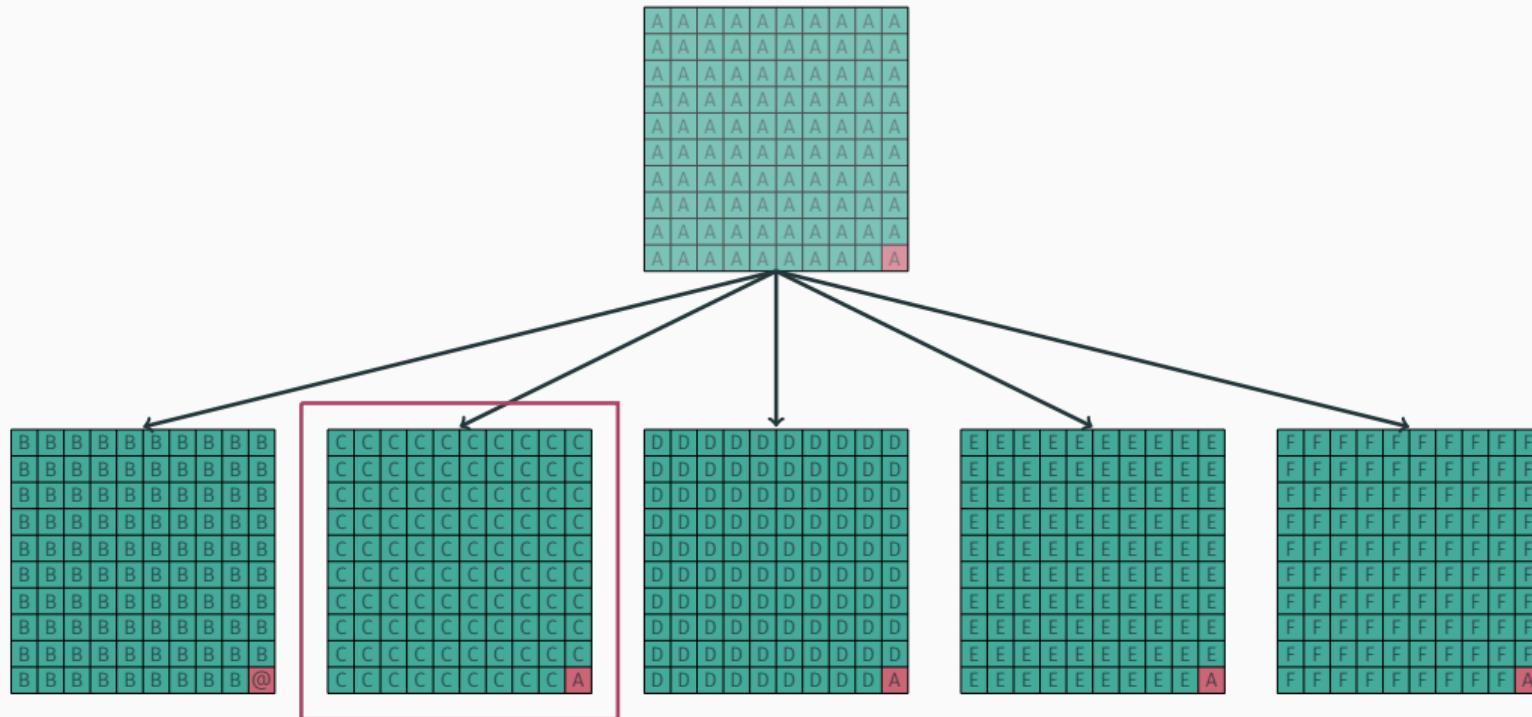
Flip Feng Shui (FFS) – Scannen des Speichers



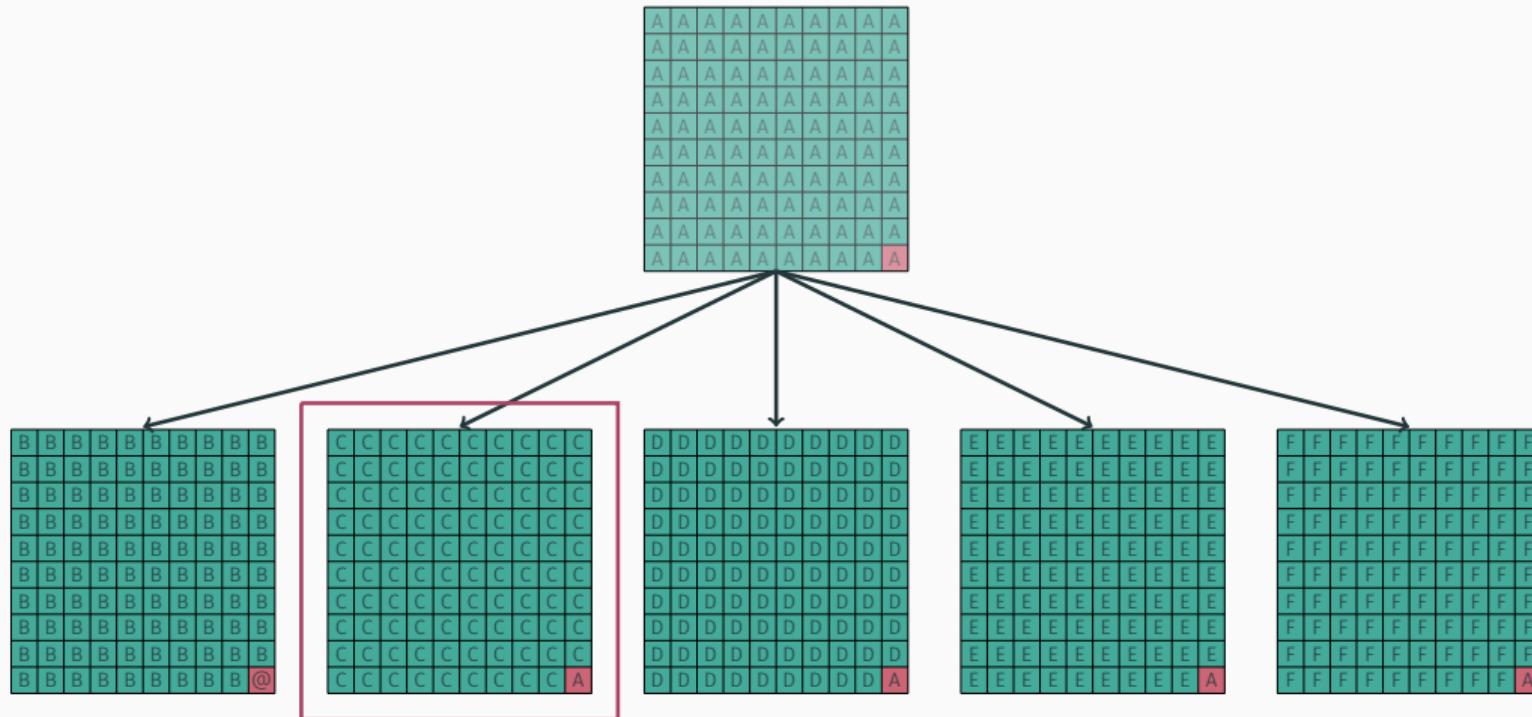
Flip Feng Shui (FFS) – Scannen des Speichers



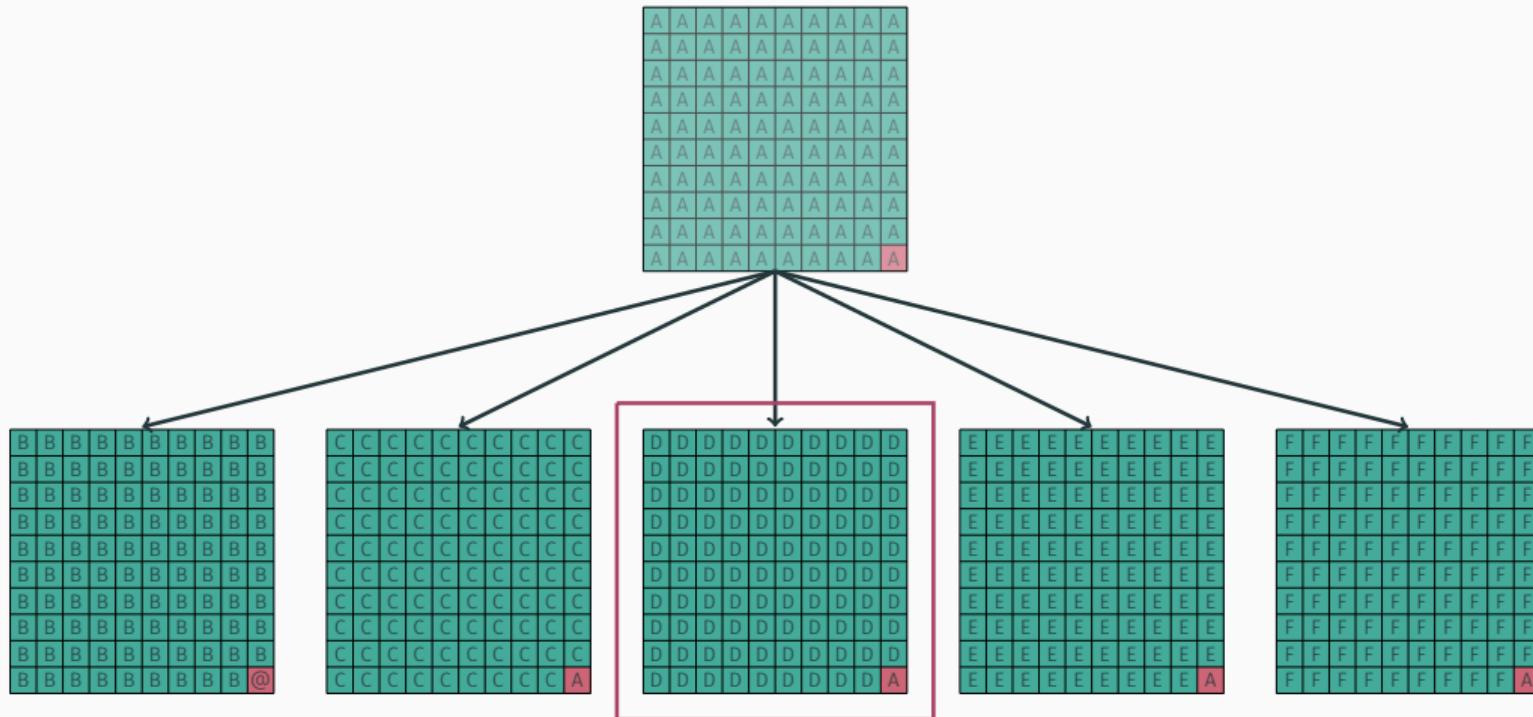
Flip Feng Shui (FFS) – Scannen des Speichers



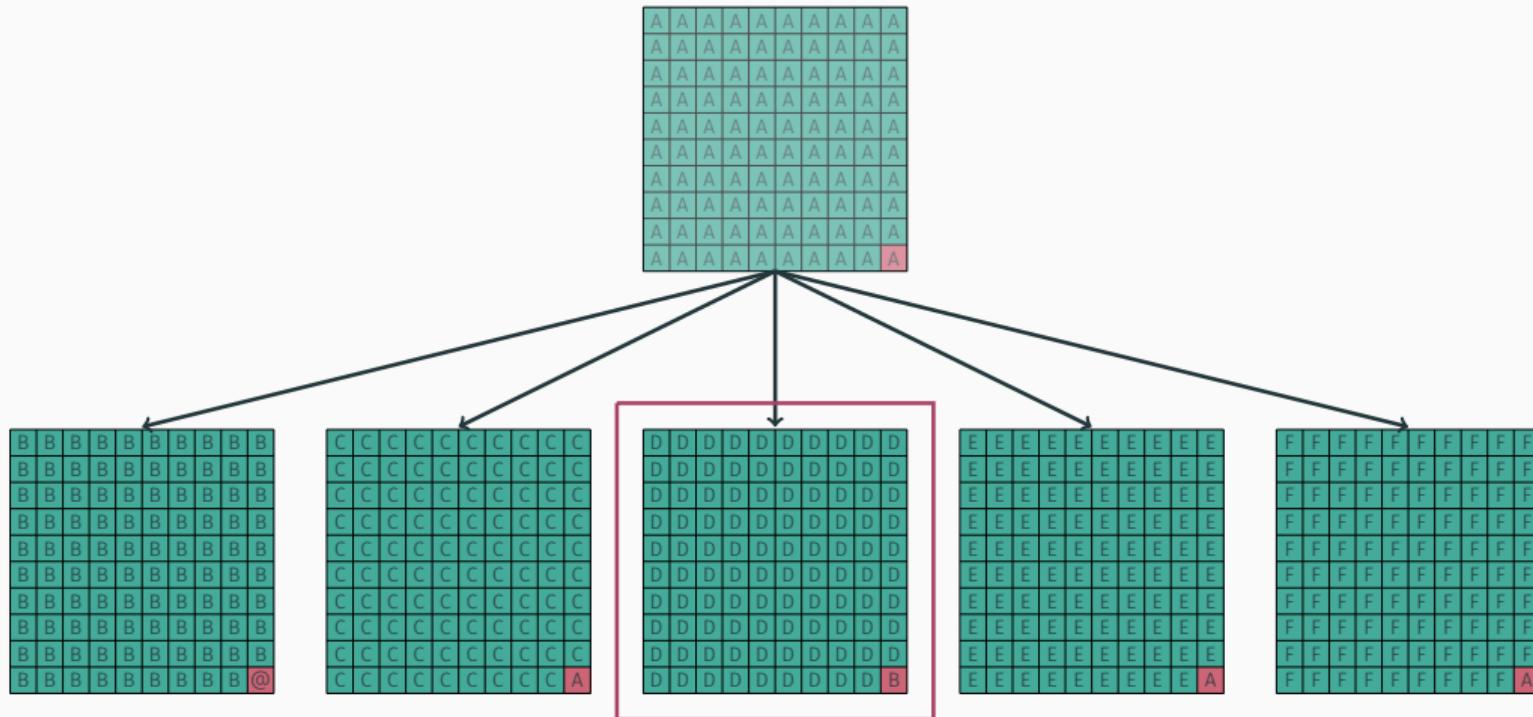
Flip Feng Shui (FFS) – Scannen des Speichers



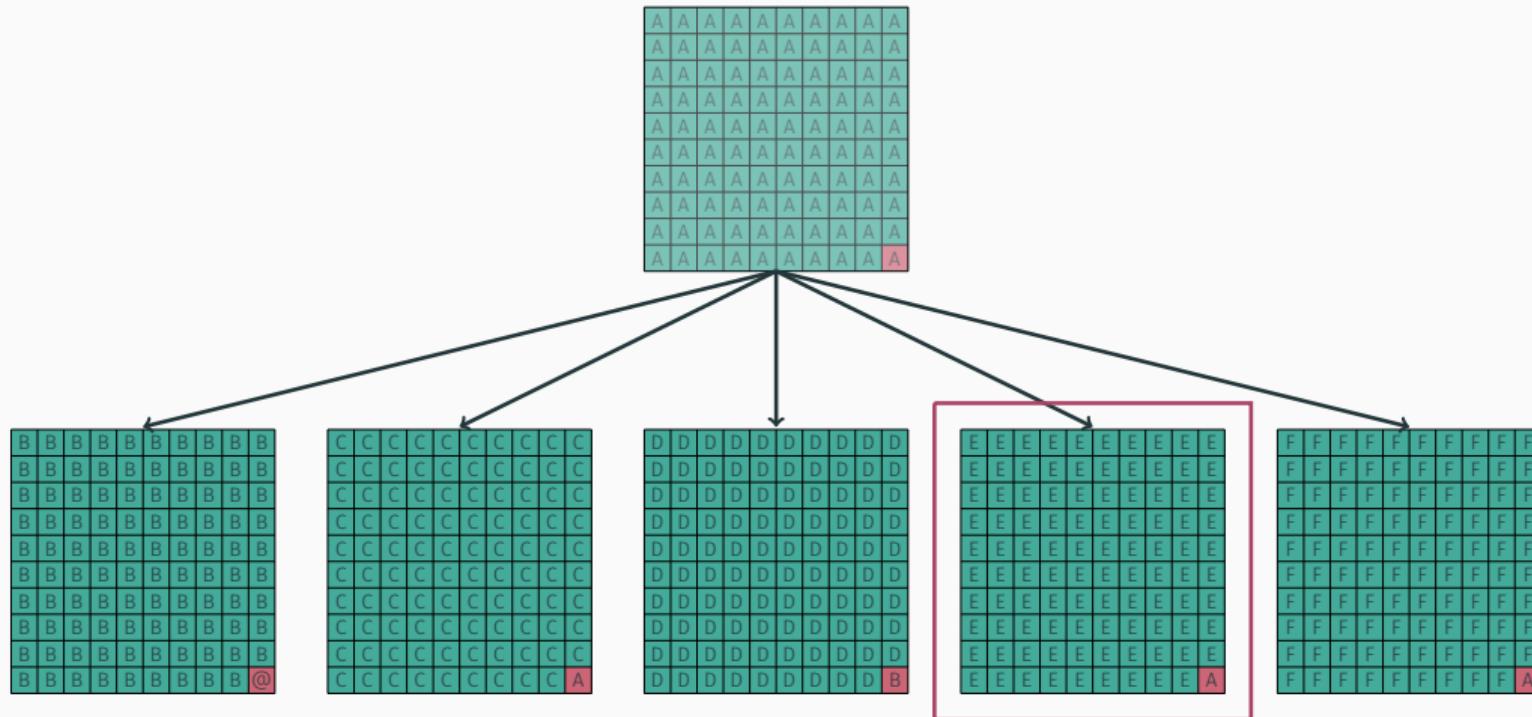
Flip Feng Shui (FFS) – Scannen des Speichers



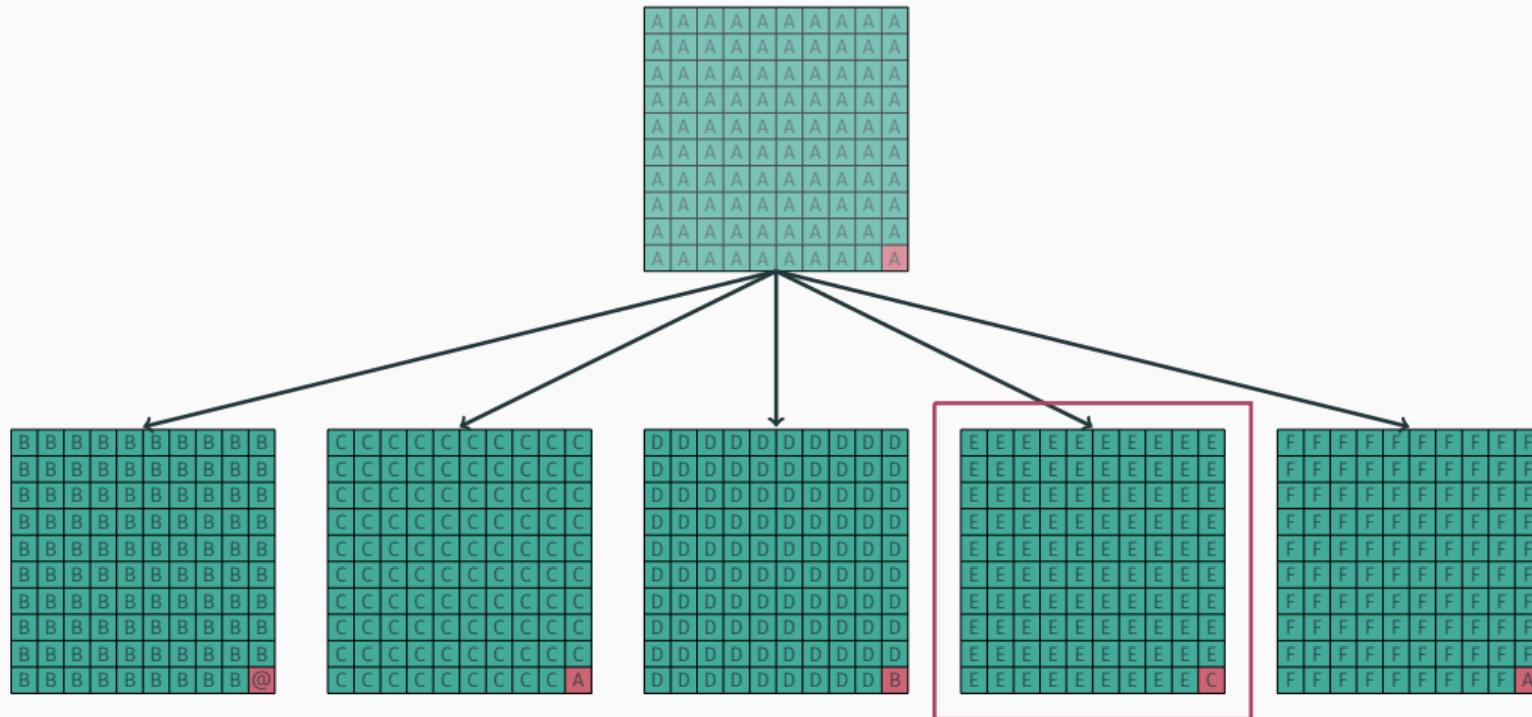
Flip Feng Shui (FFS) – Scannen des Speichers



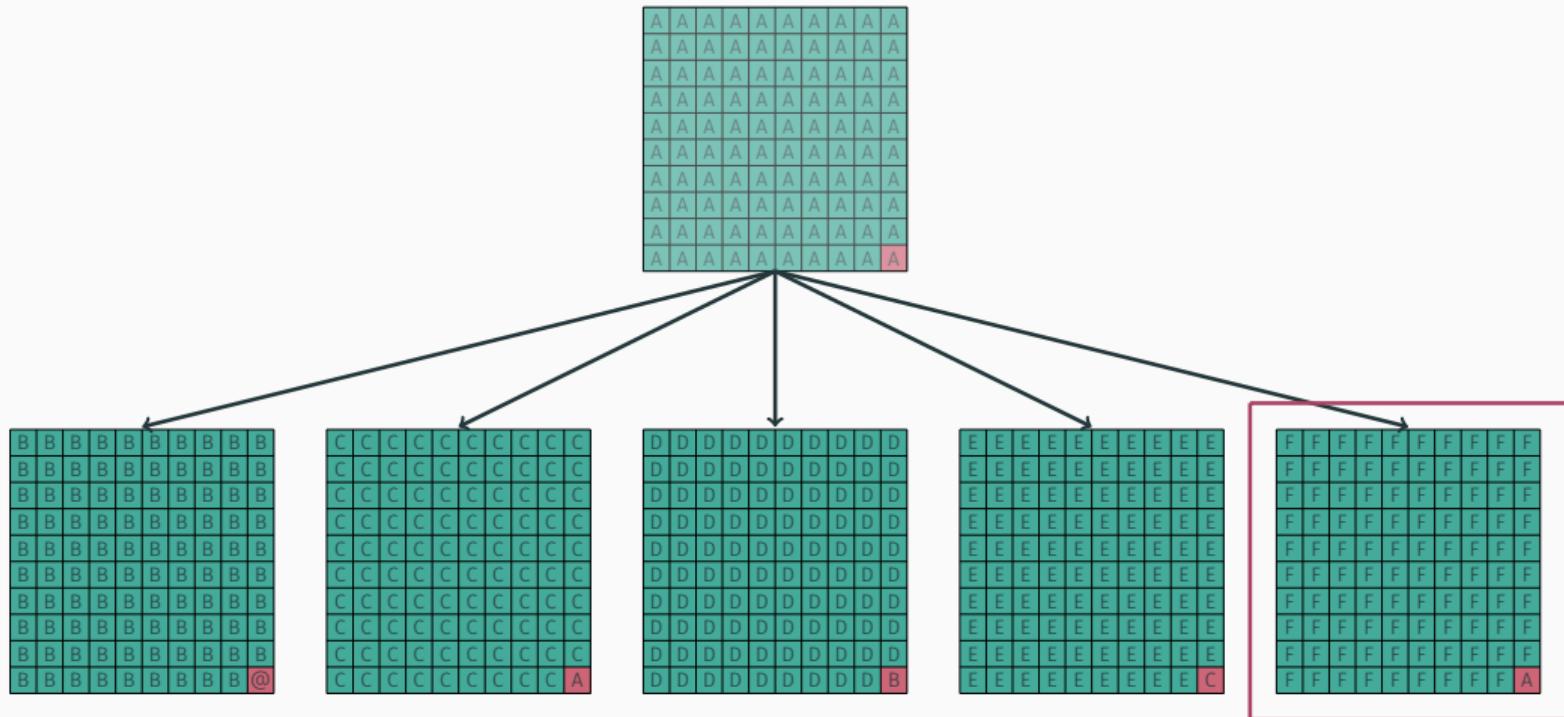
Flip Feng Shui (FFS) – Scannen des Speichers



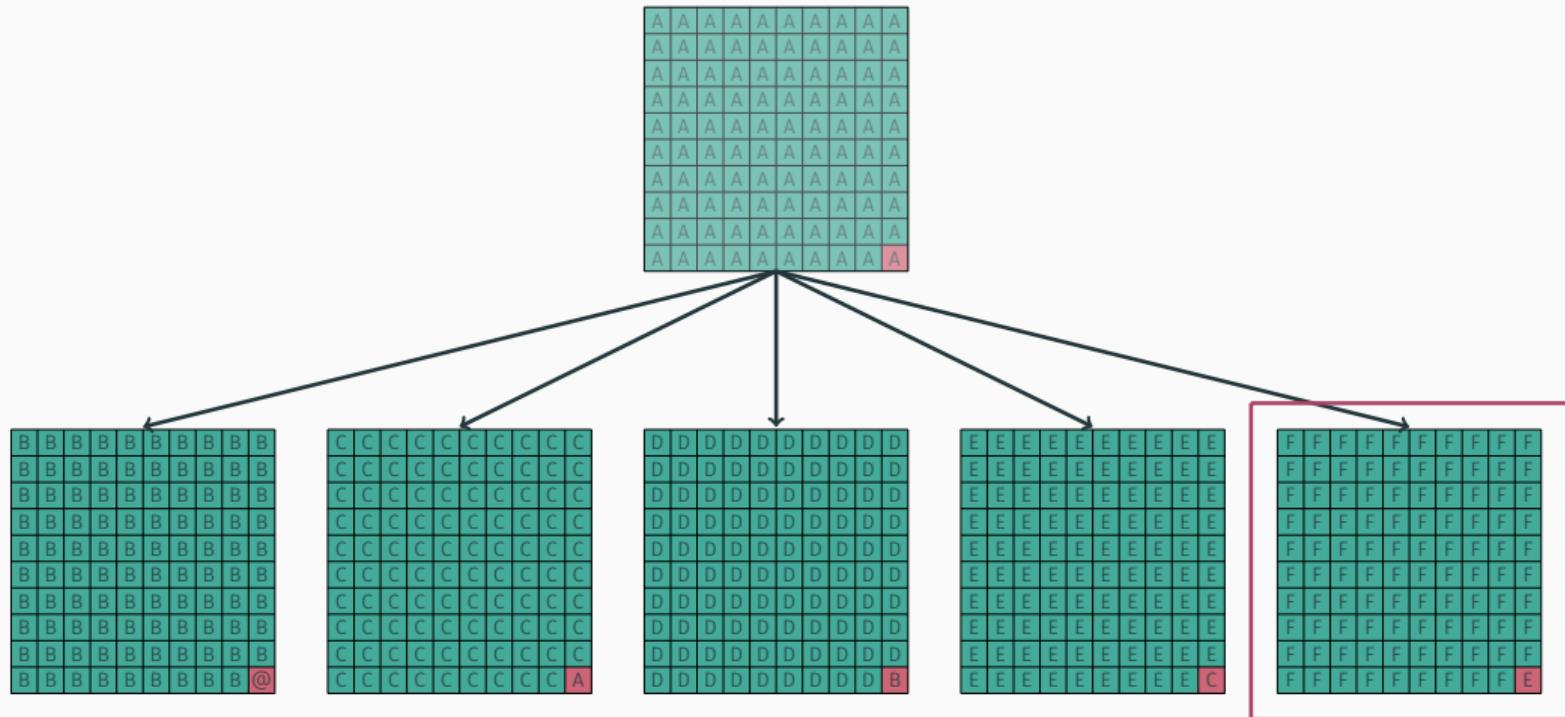
Flip Feng Shui (FFS) – Scannen des Speichers



Flip Feng Shui (FFS) – Scannen des Speichers



Flip Feng Shui (FFS) – Scannen des Speichers



Flip Feng Shui (FFS) – Mergen der Seiten und Exploit

A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A

A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A

Flip Feng Shui (FFS) – Mergen der Seiten und Exploit

A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A

A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A

A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A

Flip Feng Shui (FFS) – Mergen der Seiten und Exploit

A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A

A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A



A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A

Flip Feng Shui (FFS) – Mergen der Seiten und Exploit

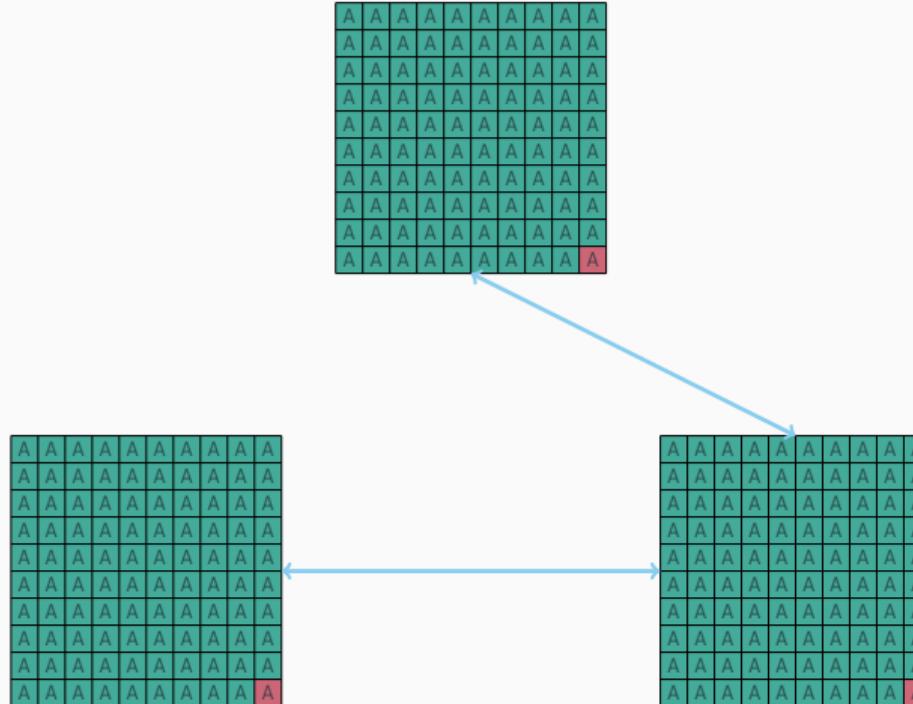
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A

A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A

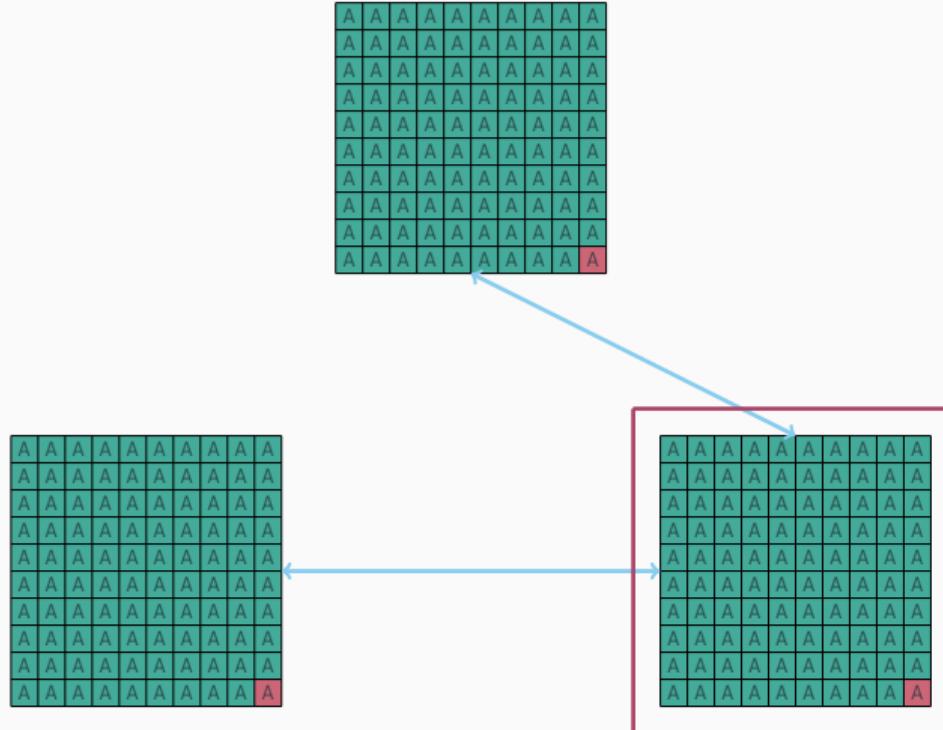


A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A

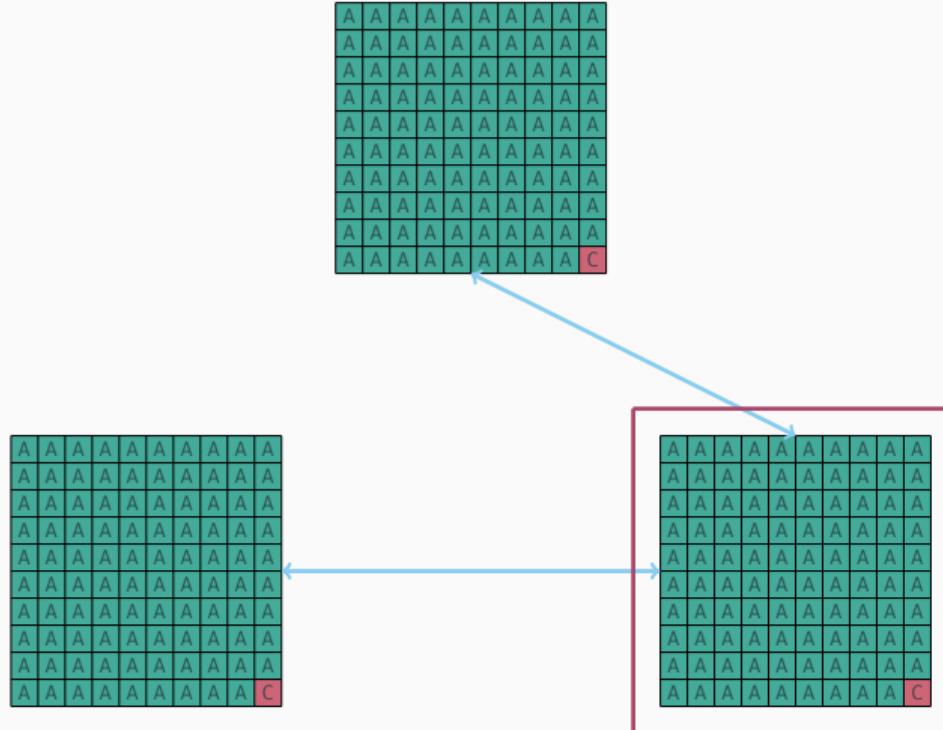
Flip Feng Shui (FFS) – Mergen der Seiten und Exploit



Flip Feng Shui (FFS) – Mergen der Seiten und Exploit



Flip Feng Shui (FFS) – Mergen der Seiten und Exploit



Demo 3

Flip Feng Shui (FFS) mit FFSTOOL

Fazit

Zusammenfassung

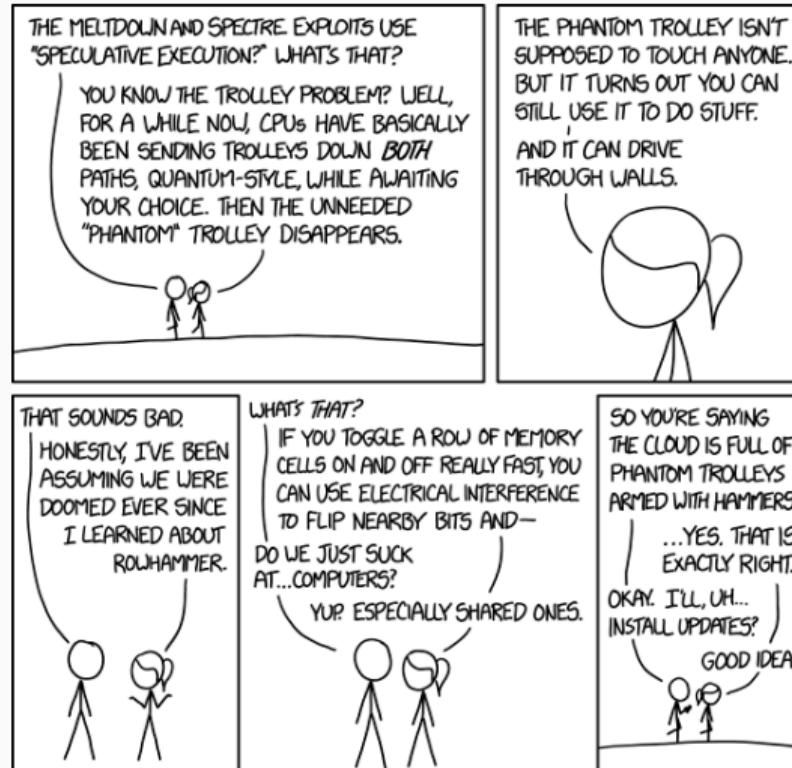


Abbildung von **XKCD**

Fragen?

- [1] Erik Bosman u.a. „Dedup Est Machina: Memory Deduplication as an Advanced Exploitation Vector“. In: S&P. Pwnie Award for Most Innovative Research. Mai 2016. URL: [Paper](https://download.vusec.net/papers/dedup-est-machina_sp16.pdf) [Web](https://www.vusec.net/projects/dedup-est-machina) [Press](https://goo.gl/ogBXTm).
- [2] Pietro Frigo u.a. „TRRespass: Exploiting the Many Sides of Target Row Refresh“. In: S&P. Best Paper Award. Mai 2020. URL: https://download.vusec.net/papers/trrespass_sp20.pdf.
- [3] Martin Heckel. „RAEAX – Rowhammer Amplification by Execution of Additional X86 instructions“. Hof University of Applied Sciences, 2021, S. 1–40.

- [4] Patrick Jattke u.a. „BLACKSMITH: Rowhammering in the Frequency Domain“. In: *IEEE S&P '22*.
https://comsec.ethz.ch/wp-content/files/blacksmith_sp22.pdf. Nov. 2021.
- [5] Yoongu Kim u.a. „Flipping Bits in Memory without Accessing Them: An Experimental Study of DRAM Disturbance Errors“. In: *SIGARCH Comput. Archit. News* 42.3 (Juni 2014), S. 361–372. ISSN: 0163-5964. DOI: [10.1145/2678373.2665726](https://doi.org/10.1145/2678373.2665726). URL:
<https://doi.org/10.1145/2678373.2665726>.

Referenzen iii

- [6] Kaveh Razavi u.a. „Flip Feng Shui: Hammering a Needle in the Software Stack“. In: *USENIX Security*. Juni 2016. URL:
https://download.vusec.net/papers/flip-feng-shui_sec16.pdf.
- [7] Mark Seaborn und Thomas Dullien. *Exploiting the DRAM rowhammer bug to gain kernel privileges*. 2015. URL:
<https://www.cs.umd.edu/class/fall2019/cmsc8180/papers/rowhammer-kernel.pdf> (besucht am 16.11.2020).
- [8] Victor van der Veen u.a. „Drammer: Deterministic Rowhammer Attacks on Mobile Platforms“. In: CCS. Pwnie Award for Best Privilege Escalation Bug, Android Security Reward, CSAW Best Paper Award, DCSR Paper Award. Okt. 2016. URL: <https://vvdveen.com/publications/drammer.pdf>.