Btrfs Dateisystem

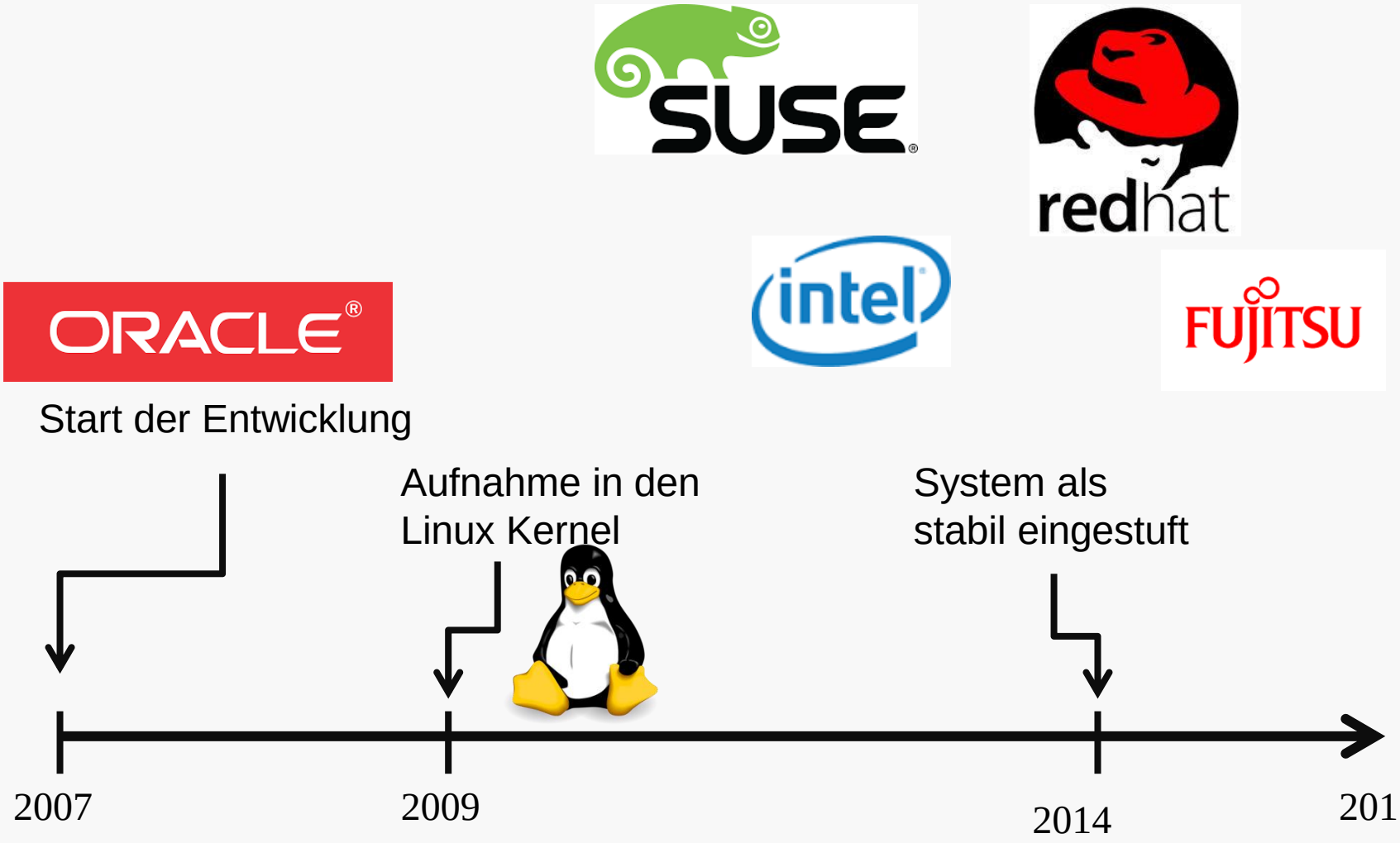
Proseminar Speicher- und Dateisysteme

Ein Vortrag von Tom Maier

Agenda

- Allgemeine Informationen
- Konzept Copy-On-Write und CoW freundlichen B+-Bäumen
- Funktionen
- Zusammenfassung
- Quellen

Einführung in btrfs



Definition

Daten Konsistenz

„Unter der Konsistenz von Daten versteht man deren Stimmigkeit. Sie liegt dann vor, wenn bestimmte Kriterien an die Datenintegrität und Plausibilität erfüllt sind“

Daten Integrität

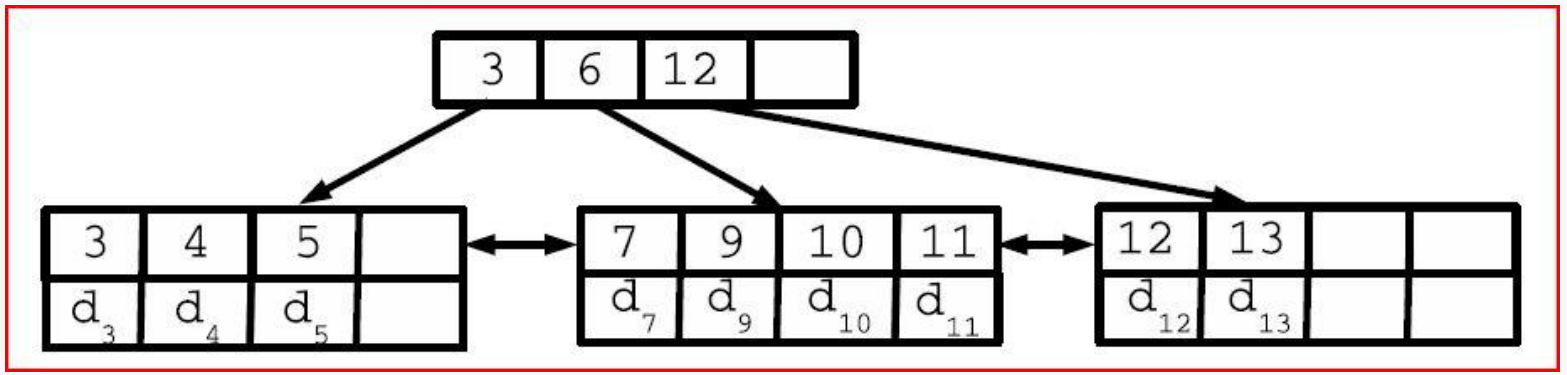
- *„[…]umfasst die Maßnahmen, die dafür sorgen, dass die geschützten Daten während der Verarbeitung nicht beschädigt oder verändert werden können.“*

Copy-On-Write

- „Überschreibe niemals irgendwelche Daten/Metadaten“
- Veränderte Daten werden auf einen neuen Platz geschrieben
- Alte Daten sind erst frei, wenn die Änderung erfolgreich abgeschlossen ist

B+-Bäume

- Datenelemente nur in den Blättern
- Innere Knoten enthalten lediglich Schlüssel
- Blätter doppelt verkettet
- Vorteil: geringere Zugriffszeiten auf die Daten
- Inkompatibel mit Copy-On-Write

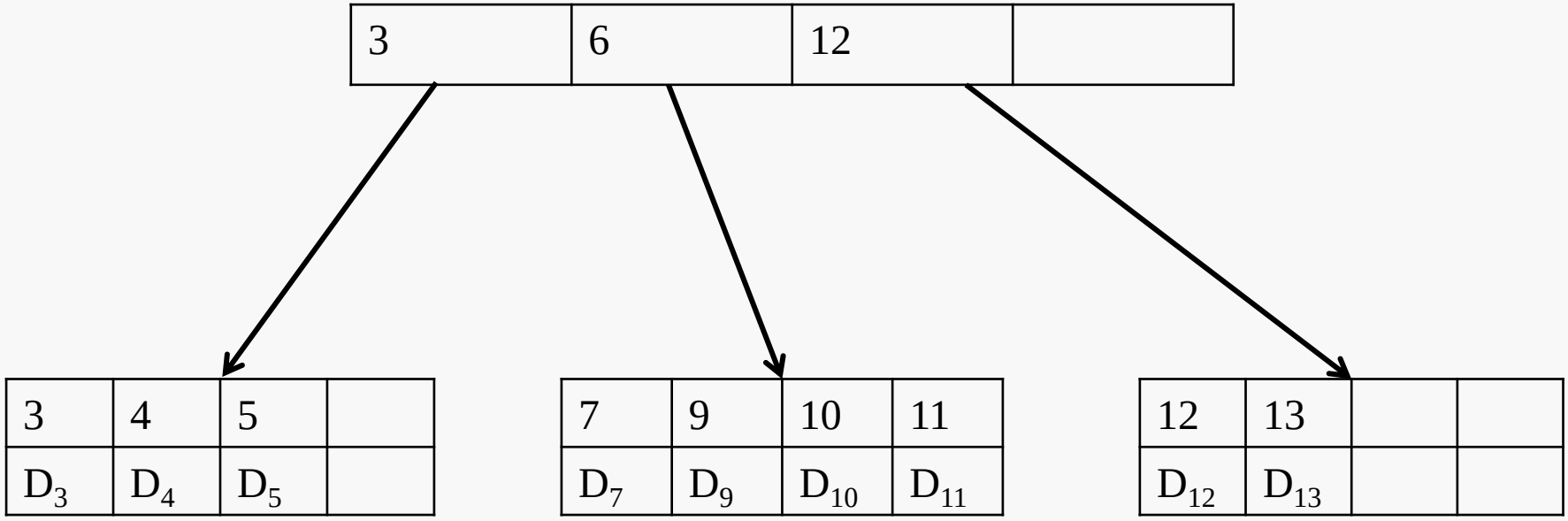


Ein klassischer B+-Baum

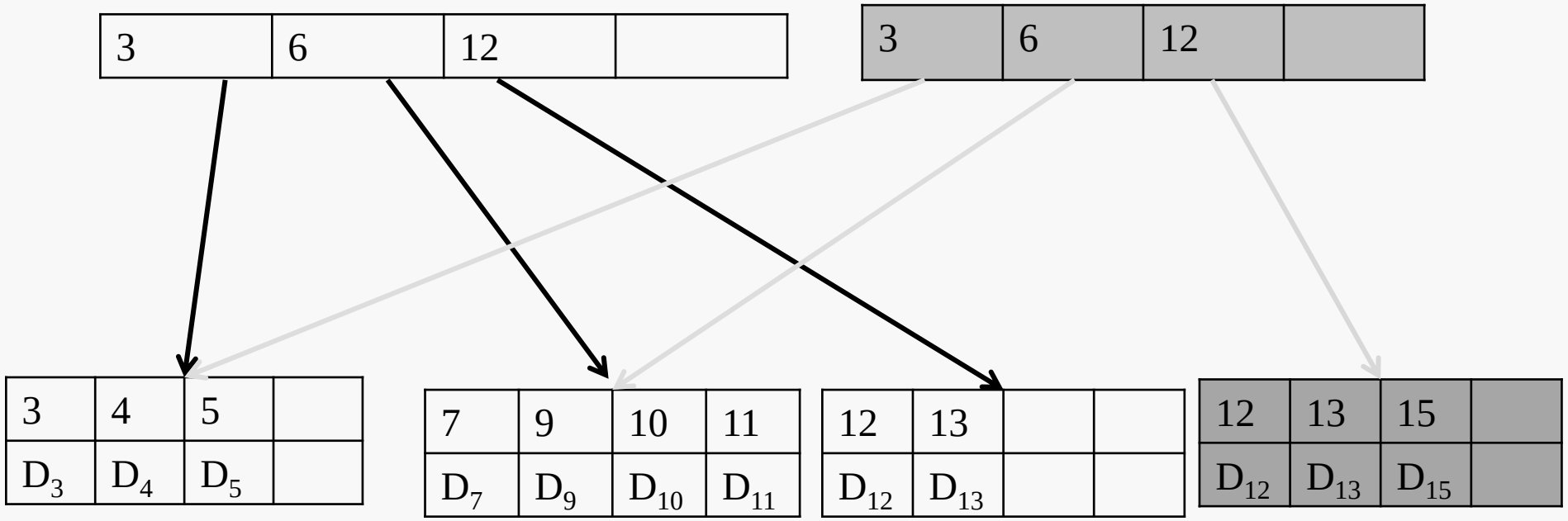
COW freundliche b+-Bäume

Unterschiede zum eigentlichen b+ Baum

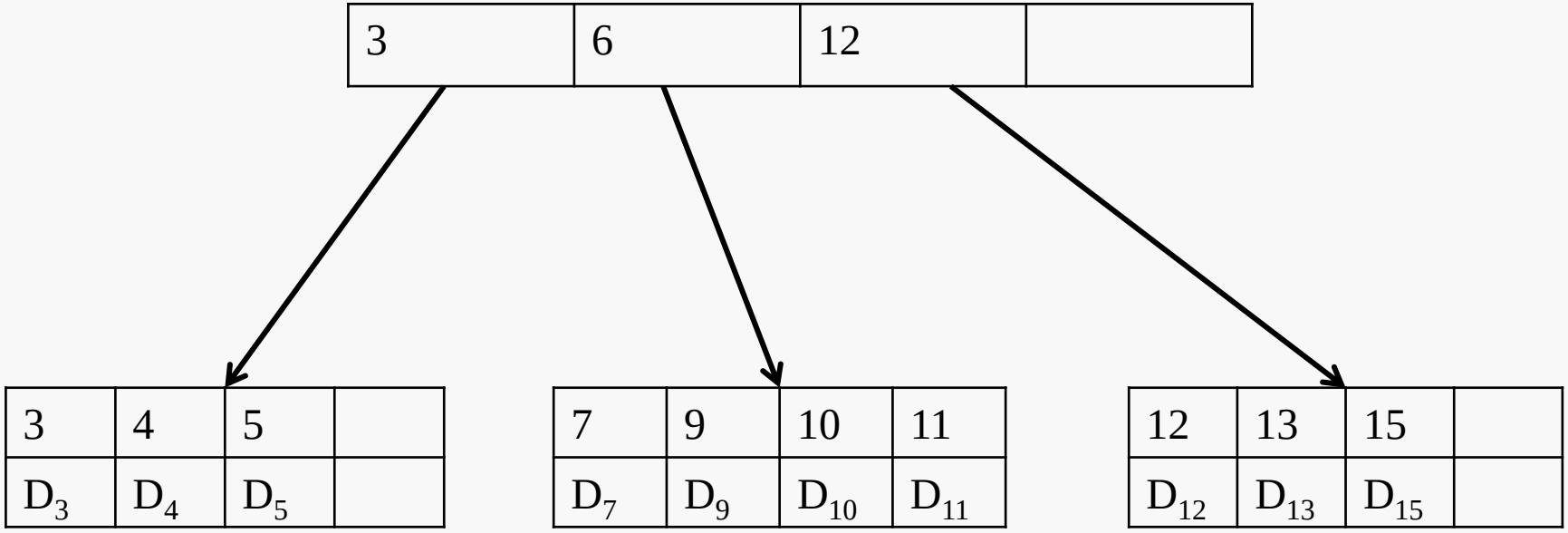
1. Blätter sind nicht mehr verbunden
2. Aktualisierungen von oben nach unten
3. Entspanntes Referenzen zählen



COW Verfahren



COW Verfahren



Datenstruktur

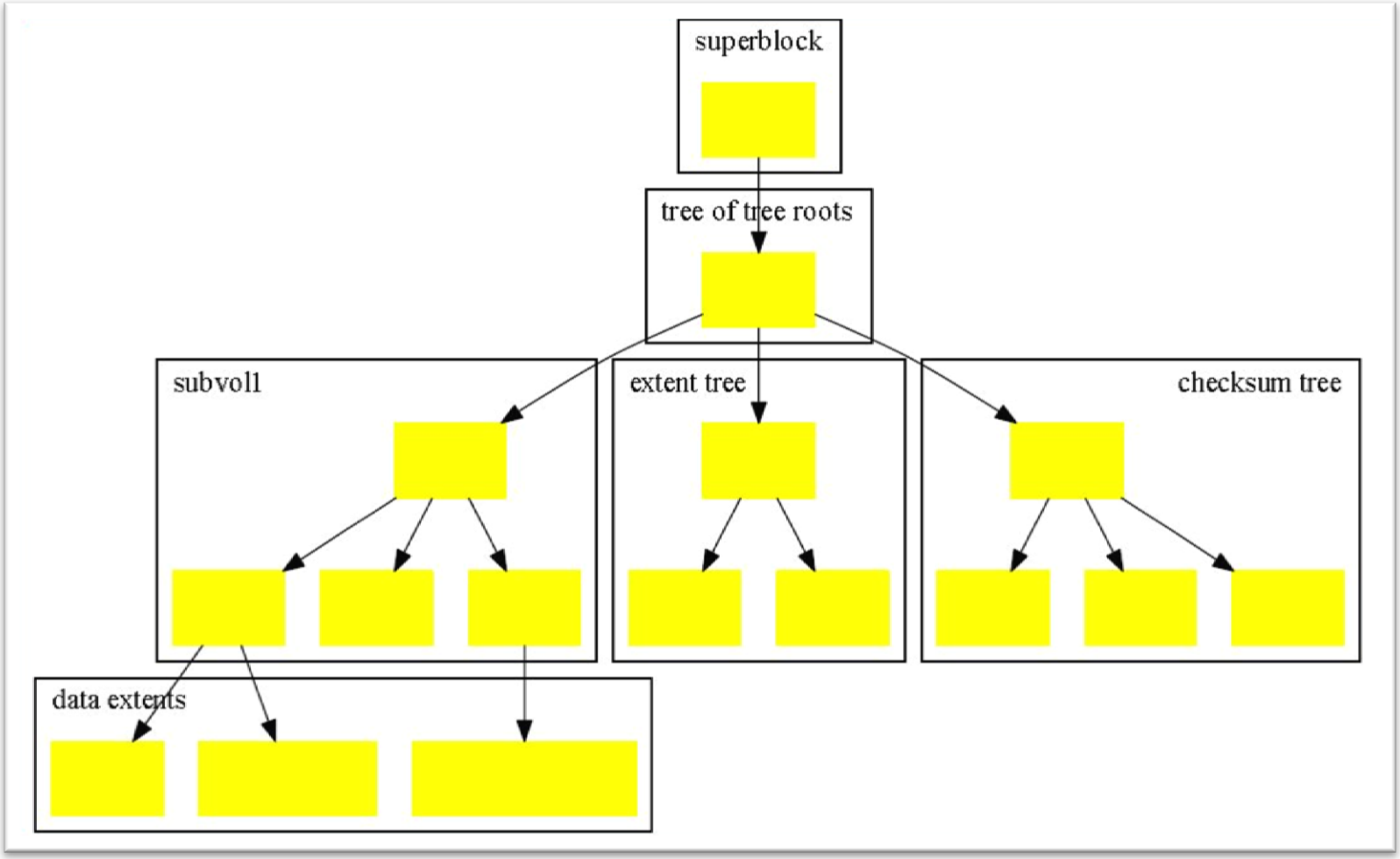
- Header
  - Prüfsummen
  - Id‘s
- Key
  - Objekt Adresse
- Items
  - Ist ein Key mit zusätzlichen Feldern

Datenstruktur

- Zwischen Knoten:
  - [key, Pointer]
- Blatt-Knoten:
  - Header
  - Paare von Items und Daten

- Extents
  - Zusammenhänge Blöcke ohne zusätzlichen Header oder Formatierungen

Ein Wald von Bäumen

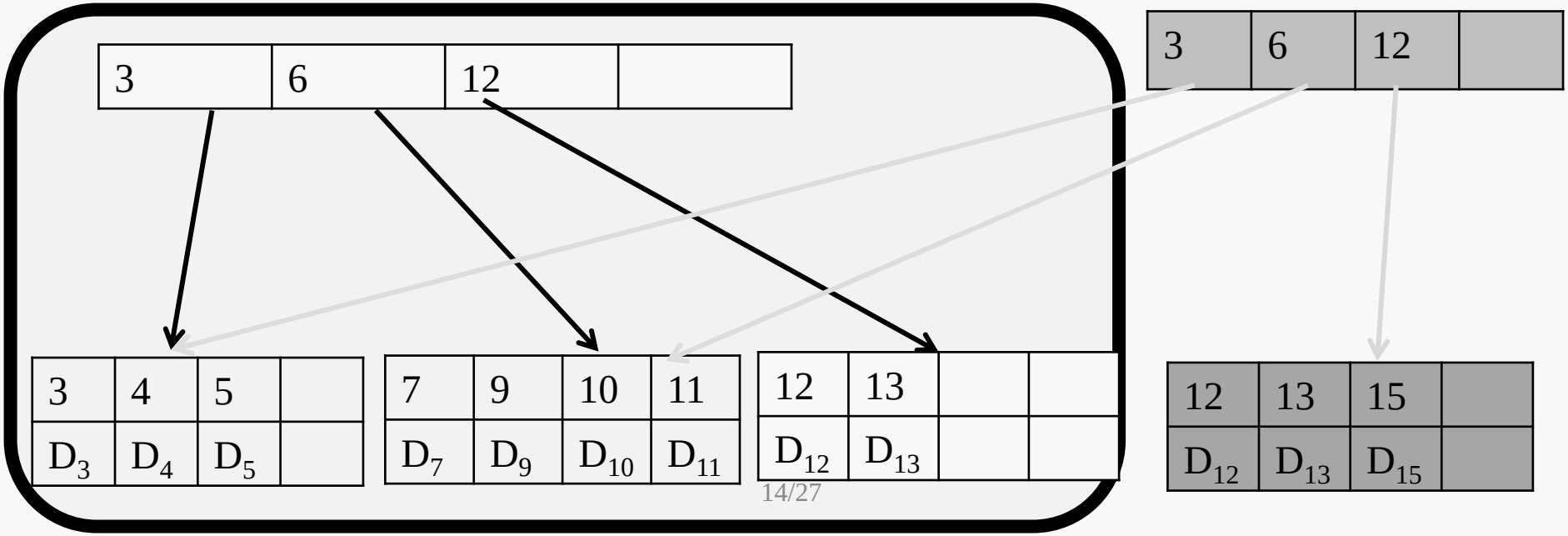


Funktionen

- Erweiterter Speicherbereich (2^{64} Byte)
- Snapshots
- Prüfsummen
- Dateisystem scrub
- Volumenverwaltung
- Datenkomprimierung
- Defragmentierung

Snapshots

- Konsistente Bilder von einem bestimmten Ort zu einem bestimmten Zeitpunkt
- Read only oder read/write Zugänge
- Durch Copy-On-Write begünstigt



+ Unterstützung bei backups
+ Sicherungspunkte vor Updates
+ Können unabhängig arbeiten

- Keine angaben wie viel Speicherplatz ein snapshot benötigt
- Einschränkung: Ein Snapshot kann eigentlich nicht das ursprüngliches Basissystem ersetzen

Prüfsummen berechnen mithilfe des crc-Algorithmus

Der Algorithmus

- Einen Generator der Länge (g+1), Eine Datei der Länge n
- An die Datei werden g Nullen angehängt
- Polynomdivison durch den Generator, wobei der Quotient nicht relevant ist

Beispiel

- Geneartor = 110101,
  Datei = 11011
- Datei + g= 1101100000
- 1101100000 ÷ 110101

  -110101

  0000110000

  -110101

  000101
- Prüfsumme = 101

Dateisystem scrub

- Fehler werden gesucht
  - Durch prüfsummen
- und falls möglich behoben
  - Beispielsweise durch Benutzung von RAID 1
- Läuft im Hintergrund

Volumenverwaltung

- BTRFS teilt alle angeschlossenen Speichermedien in Chunks auf
- Chunk-Tree verwaltet die Zuordnung von logischen zu physischen Chunks
- Device-Tree kümmert sich um die umgekehrte zuordnung
- Dateisystem sieht nur logische Chunks
- Unterstützung von Raid 0, 1, 5, 6 und 10
- Vorteile:
  - Leichte Änderung von RAID-Leveln
  - Verschiedene Subvolumes können verschieden RAID-Level haben
  - Geräte zur Laufzeit hinzufügen und entfernen

Balance-Algorithmus

(a) 2 disks	logical chunks	disk 1	disk 2	
	L_1	C_{11}	C_{21}	
	L_2	C_{12}	C_{22}	
	L_3	C_{13}	C_{23}	
(b) disk added				disk 3
	L_1	C_{11}	C_{21}	
	L_2	C_{12}	C_{22}	
	L_3	C_{13}	C_{23}	
(c) rebalance				
	L_1	C_{11}	C_{21}	
	L_2		C_{22}	C_{12}
	L_3	C_{13}		C_{23}

- Sehr langsamer Algorithmus -> lohnt nur selten

Datenkomprimierung

ZLIB

+ hohes Komprimierungslevel

- Sehr langsam

LZO

+ schnelles komprimieren

+ schnelles extrahieren

- Niedriges Komprimierungslevel

- „alltäglichen Gebrauch“ geeignet

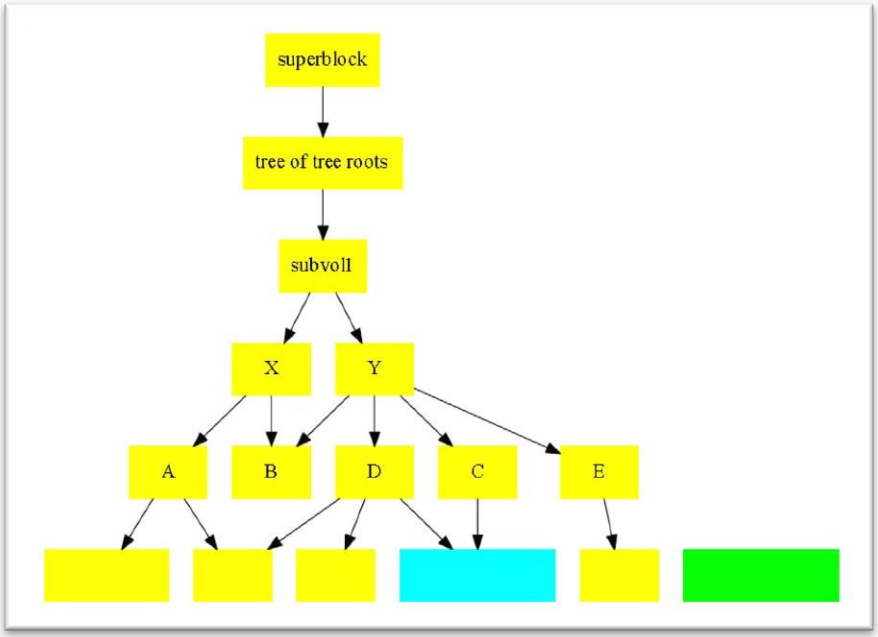
→ Für jedes Verzeichnis kann der Algorithmus neu gewählt werden

Defragentierung

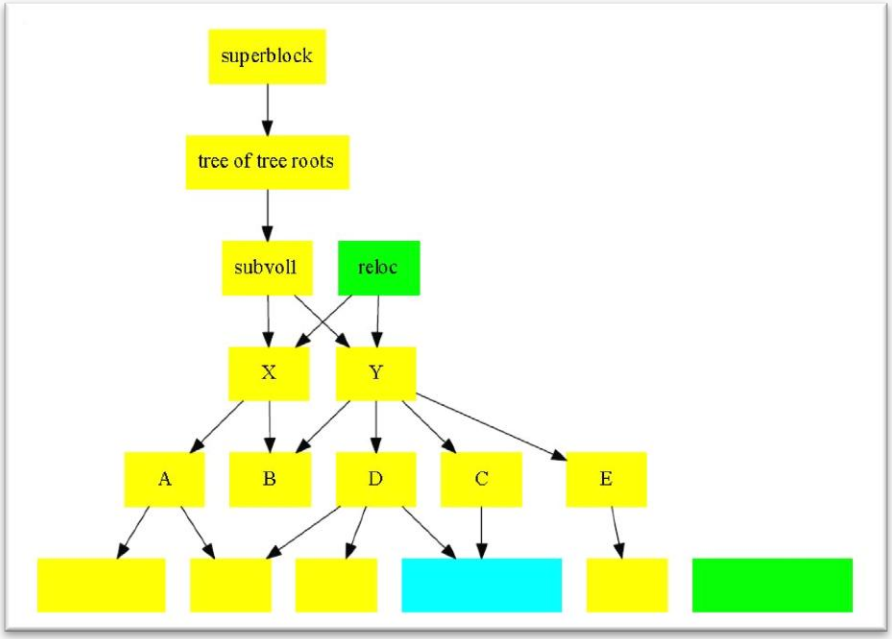
- Schwierigkeiten
  - Extents können erst verschoben werden, nachdem alle verweise geupdatet wurden
  - nahe extents begünstigen snapshots
- Lösung:
  - Reloc – Algorithmus

Reloc-Algorithmus

1. Extents an den neuen Platz kopieren

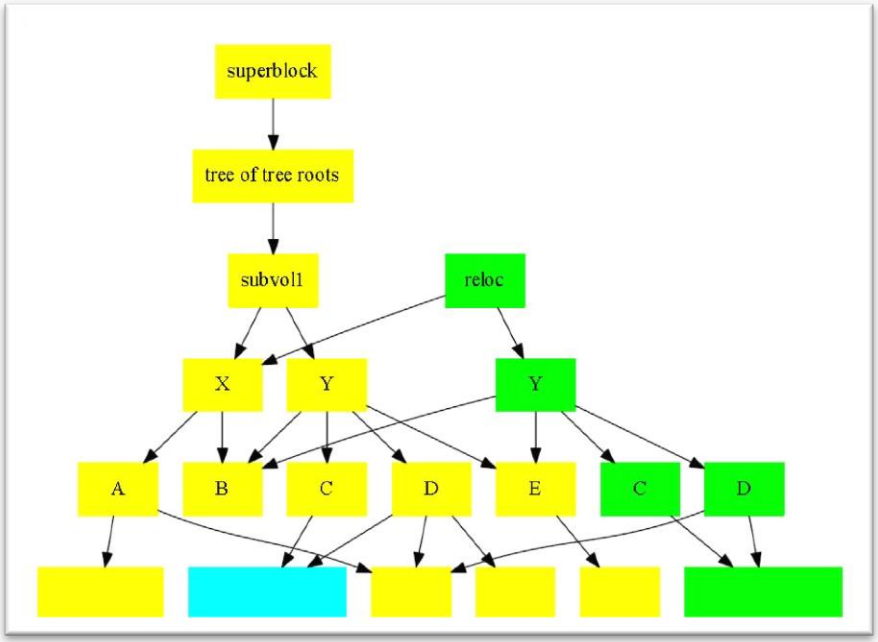


2. Subvolume1 kopieren und die Referenzen speichern

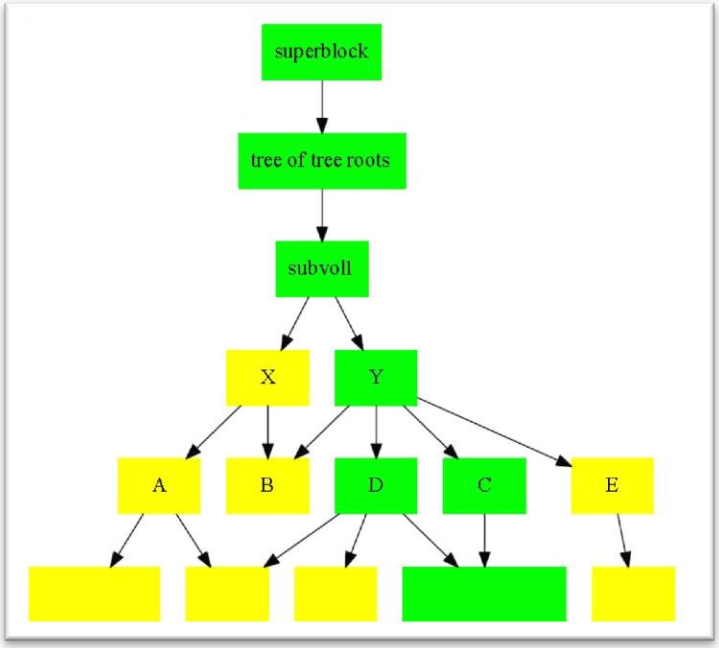


Reloc-Algorithmus

3. Referenzen updaten



4. Reloc-Tree einfügen und löschen



Zusammenfassung

- Schwierige Entwicklungsphase
- Immer noch leichteProbleme
- Unwesentlich schneller als die Konkurenz (ZFS, Ext4)
- Erhöhte Datensicherheit
- Viele neue und hilfreiche Features
  - Entwichlung noch nicht abgeschlossen

→Dateisystem der Zukunft

Quellen

- „Main Page“ https://btrfs.wiki.kernel.org/index.php/Main_Page,
- Ohad Rodeh, Josef Bacik, Chris Mason, „BTRFS: The Linux B-Tree Filesystem“ http://domino.research.ibm.com/library/cyberdig.nsf/papers/6E1C5B6A1B6EDD9885257A38006B6130/$File/rj10501.pdf
- „Schnappschuss (Informationstechnik) https://de.m.wikipedia.org/wiki/Schnappschuss_(Informationstechnik)
- „Btrfs“ https://en.m.wikipedia.org/wiki/Btrfs
- Jan Kára, “Ext4, btrfs, and the others” http://atrey.karlin.mff.cuni.cz/~jack/papers/lk2009-ext4-btrfs.pdf
- „Fehlererkennung mit CRC“http://banane-krumm.de/crc/index.html
- „Zyklische Redundanzprüfung“https://de.m.wikipedia.org/wiki/Zyklische_Redundanzprüfung
- http://www.itwissen.info/definition/lexikon/Datenintegritaet-data-integrity.html

Quellen

- http://www.itwissen.info/definition/lexikon/Datenintegritaet-data-integrity.html
- **„Compression“** https://btrfs.wiki.kernel.org/index.php/Compression
- Kofler, Michael: Linux das umfassende Handbuch, Bonn 2014, S.840-852
- http://www.mapsysinc.com/wp-content/uploads/2013/08/oracle-logo.gif
- http://www.fusionio.com/load/-media-/1uliym/imagesPartner/suse.png
- http://ee0ac3a21a162862e68e-92846f6a8249f4291472a4f3d00d0a60.r16.cf5.rackcdn.com/wp-content/uploads/2015/03/red-hat-logo.jpg
- https://upload.wikimedia.org/wikipedia/commons/thumb/c/c9/Intel-logo.svg/150px-Intel-logo.svg.png
- https://upload.wikimedia.org/wikipedia/commons/thumb/5/53/Fujitsu-Logo.svg/200px-Fujitsu-Logo.svg.png

Viele für ihre Aufmerksamkeit