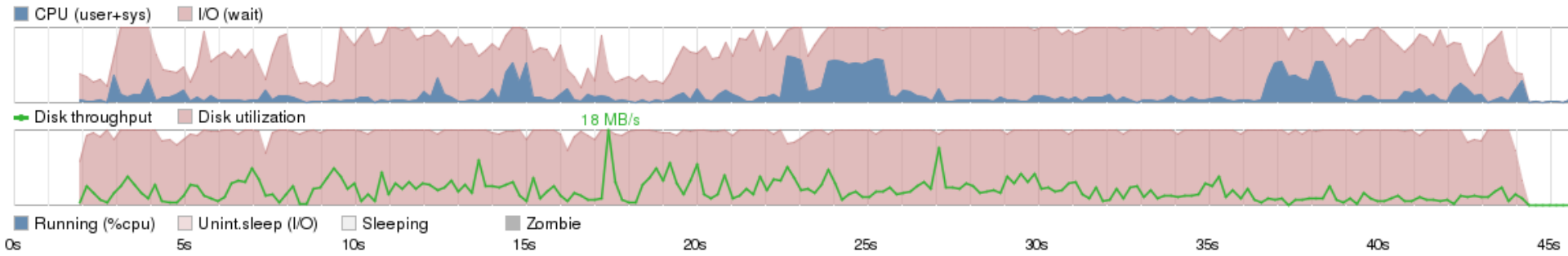
Booten in 15 statt 44 Sekunden

Optimieren durch physikalische Umsortierung

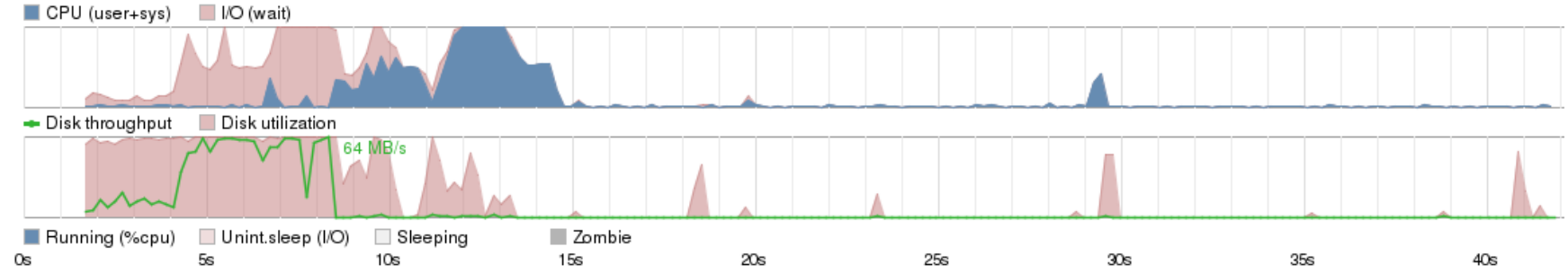
Andreas Rid
Gundolf Kiefer

Bootchart

Vorher:

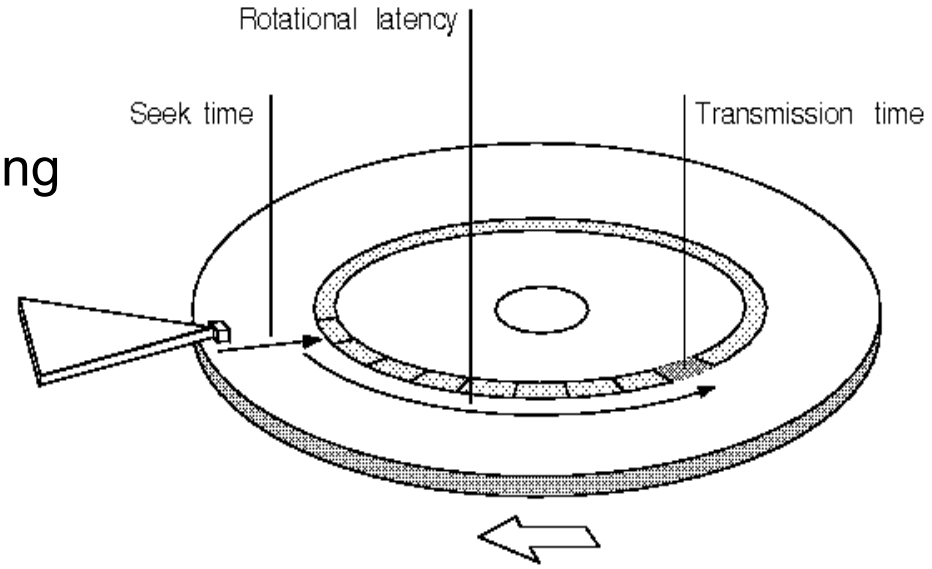


Nachher:



Zugriffszeiten der Festplatte

- Zugriffszeit
  - (8.9 ms) Spurwechselzeit
  - (4.2 ms) Rotationsverzögerung
  - (0.1 ms) Kommando-Latenz



- Problem
  - Fragmentierte Dateien
  - Streuung von Dateien
  - Insbesondere im Bootvorgang sehr kleine Dateien

Übersicht

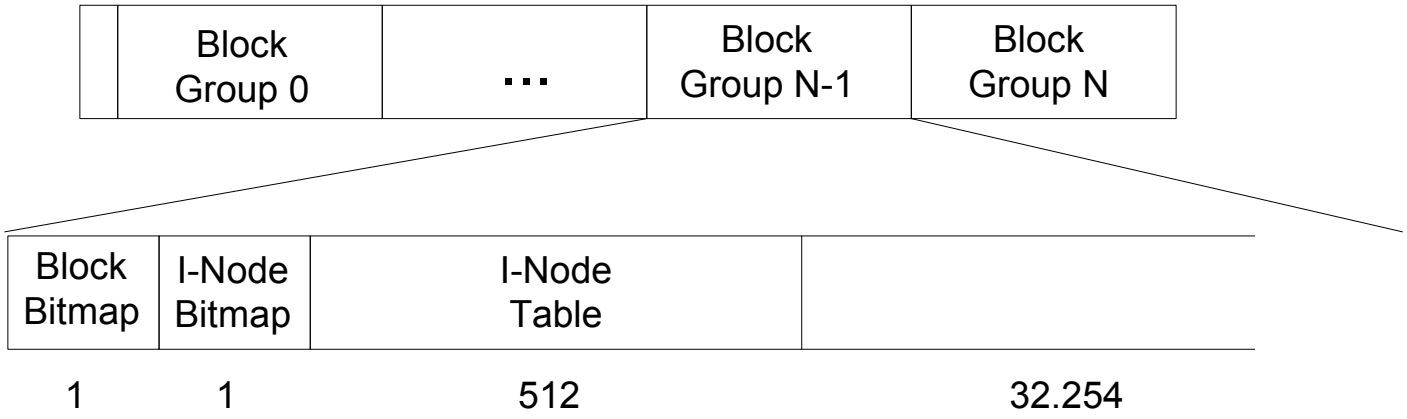
1. Was gibt es schon?
2. Grundlagen: Das Ext4 Dateisystem
3. Ansatz
4. Sammeln von Dateizugriffen
5. Dateien umsortieren
6. Vorführung

Was gibt es schon?

- ureadahead / sreadahead
- Readahead von Fedora
- Upstart, ...

Grundlagen: Das Ext4 Dateisystem

- Layout



- I-Node (Index Node)
  - Datei, Verzeichnis, Soft-Link, …
  - Zugriffsrechte
  - Block-Mapping

Grundlagen: Das Ext4 Dateisystem

- Neue Features in Ext4
  - Flexible Blockgruppen
  - Extents

```
struct ext4_extent {
        __le32  ee_block;
        __le16  ee_len;
        __le16  ee_start_hi;
        __le32  ee_start_lo; }
```

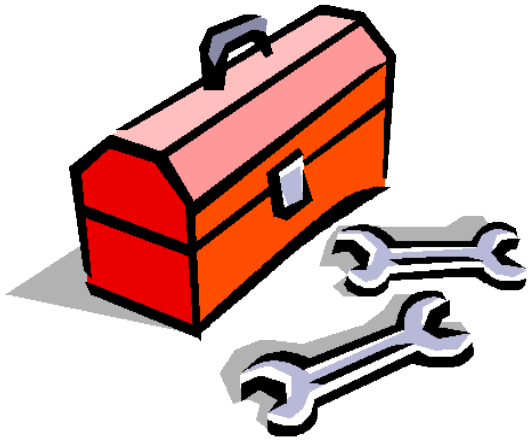
- Neuerungen im Dateisystemtreiber
  - Multi-Block-Allokator
  - Pre-Allokation
    - I-Node Pre-Allokation Space
    - Locality Group
  - Online Defragmentierung (e4defrag)

Ansatz

- System überwachen

  Welche Dateien stehen im Zugriff?
- Dateien physikalisch auf der Festplatte anordnen
  - Suche nach freien Speicher
  - Datei verschieben

Werkzeuge



1) e4rat-collect

   → Sammeln von Dateizugriffen

2) e4rat-realloc

   → Dateien auf der Festplatte umsortieren

3) e4rat-preload

   → Dateien schnell in den Page Cache übertragen

e4rat-collect – relevanter Dateien

- Systemaufrufe überwachen (Linux Audit)

  execve, open, ...
- Art des Zugriffes

  Lese- oder Schreibzugriff?
- Befindet sich die Datei bereits im Cache?

  libc.so

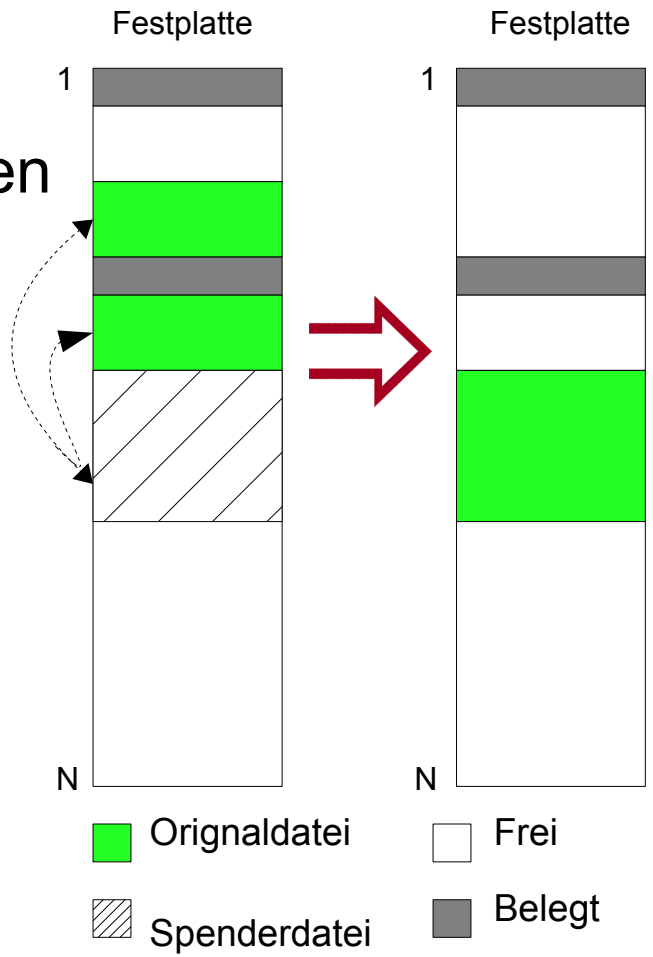
Online Defragmentierung

- EXT4_IOC_MOVE_EXT

  Vertauschen von Blockbelegungen

  zwischen zwei Dateien

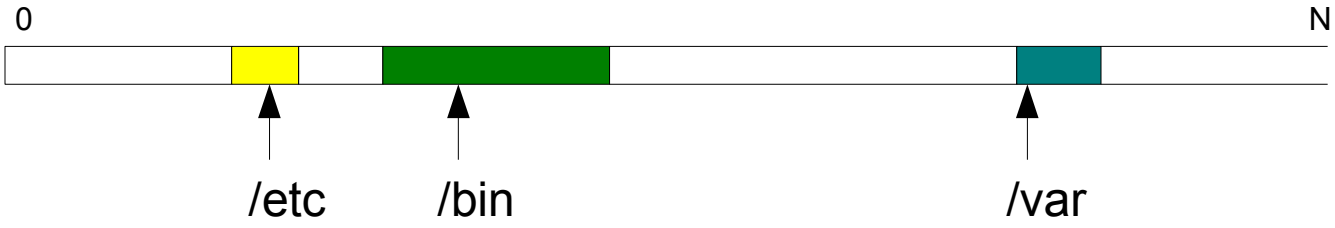
- E4defrag
  - Spenderdatei erstellen
  - Ioctl EXT4_IOC_MOVE_EXT
  - Spenderdatei löschen



e4rat-realloc – Modi

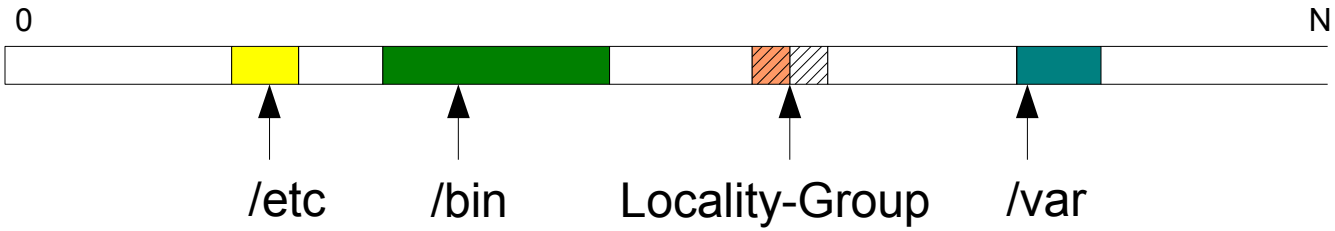
- Pre-Allokation
  - Kernel Patch einspielen 
  - Keine Grenzen durch Multi-Block-Allokator
    - Blöcke gezielt allozieren
    - Suche nach freien Speicherbereichen über die Grenze einer Blockgruppe hinaus

e4rat-realloc – Modi



- Top-Level Directory
  - Orlov-Algorithmus gruppiert Dateien im gleichen Anfangsverzeichnis
  - Problem:
    - Block-Allokator alloziert die nächst höhere Potenz von 2
    - Grenze von kleine Dateien erhöhen

e4rat-realloc – Modi



- Locality Group
  - Ursprünglich vorgesehen für kleine Dateien
  - Grenze kleiner Dateien erhöhen
  - Die größe der Locality Group kann gesetzt werden
  - Problem
    - Füllgrad der Locality Group unbekannt
    - Fremde Blockanfragen

Zusammenfassung

- Startzeit verringern durch
  - Blockumsortierung
  - IO-Wartezeiten ausnutzen (e4rat-preload)
- Notwendige Schritte
  - Generieren einer Dateiliste
  - Dateiinhalte verschieben
- Projektseite: http://e4rat.sf.net

Vorführung

Auf die Plätze …

Fertig ….

Los!

Vielen Dank für Ihre Aufmerksamkeit!