ZRAM - RAM Kompression

Seminar

Benjamin Warnke

Arbeitsbereich Wissenschaftliches Rechnen
Fachbereich Informatik
Fakultät für Mathematik, Informatik und Naturwissenschaften
Universität Hamburg

2018-01-18

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

informatik die zukunft

Gliederung (Agenda)

1. Grundlagen
2. Funktionsweise
3. Anwendungsgebiete
4. LIVE-Beispiel
5. Benchmarks

Gliederung (Agenda)

1 Grundlagen

2 Funktionsweise

3 Anwendungsgebiete

4 LIVE-Beispiel

5 Benchmarks

SWAP

- Seit 1960 RAM Virtualisierungsverfahren (swap sth. out)
- Nicht benötigte Daten werden ausgelagert
- Ursprünglich Partition auf HDD
- Verfügbarer Arbeitsspeicher = physikalischer Arbeitsspeicher + virtueller Arbeitsspeicher
- Enthalten in jedem Kernel
- Nicht beeinflussbar, was ausgelagert wird
- Unter Linux konfigurierbar mit der boot-flag *vm.swappiness*

Linux-Kernel

- Geschrieben in C
- Open-source
- Speicher ist aufgeteilt in Seiten
- Vorteile gegenüber user-space-C
  - Direkter Speicherzugriff (struct page*)
  - Mehr Rechenzeit
- Nachteile gegenüber user-space-C
  - Direkter Speicherzugriff (struct page*)
  - Keine Bibliotheken (stdlib, stdio, boost, mpi, openmp, $\cdots$)
  - Segfault $\rightarrow$ kernel panic = crash
  - Keine Fließkommazahlen, keine Vektorregister
  - Eigener Code muss threadsafe sein

ZRAM

- Bestandteil des offiziellen Linux-Kernel-Source-Codes
- Linux Kernel *module* oder *build-in*
- Erzeugt virtuelles blockdevice (Festplatte) $\rightarrow$ 4096 Byte Blöcke
- Benutzbar als RAM-disk und/oder SWAP-device(s)
- Aktuell bis zu 5 Kompressionsalgorithmen zur Auswahl
- Kompressionsrate pro Seite mindestens 1.2 oder gar nicht

Gliederung (Agenda)

1 Grundlagen

2 Funktionsweise

3 Anwendungsgebiete

4 LIVE-Beispiel

5 Benchmarks

ZRAM

- memory-management Modul entscheidet, was ausgelagert wird
- Komprimiert einzelne Seiten (4096 Bytes)
- Kompressionsalgorithmen bereitgestellt durch die crypto-api
- Komprimierte Daten landen in einem Buffer
- Komprimierte Daten werden in *size-classes* gruppiert
- Buffer wird umkopiert in den Speicher der *size-class*

/sys/kernel/debug/zsmalloc/zram0/classes

class	size	obj_used	pages_used	pages_per_zspage
0	32	0	0	1
1	48	0	0	3
2	64	0	0	1
...	...	...	...	...
107	1744	1662	714	3
111	1808	1030	460	4
126	2048	3783	1892	1
144	2336	4398	2516	4
151	2448	2298	1380	3
168	2720	7775	5184	2
190	3072	16425	12321	3
202	3264	15197	12160	4
254	4096	143155	143155	1
Total		201485	181846	

Gliederung (Agenda)

1 Grundlagen

2 Funktionsweise

3 Anwendungsgebiete

4 LIVE-Beispiel

5 Benchmarks

Android

- Flash-Speicher hat begrenzte Lebenszeit
  $\rightarrow$ Swap auf Flash-Speicher nicht sinnvoll
- Benutzt Linux Kernel
- Enthält ZRAM seit Version 4.4
- Pro-Contra
  - ZRAM 'vergrößert' verfügbaren RAM
  - Kompression + Dekompression kosten CPU-Zeit + Energie
- Kompressionsrate nicht vorhersagbar
- Nicht jeder Android-Kernel unterstützt ZRAM

Linux-Server

- ZRAM ist enthalten im Linux Kernel
- Im mainline Kernel seit Version 3.14 (30.März 2014)
- Über 4 Jahre *staging*
- Schneller als HDDs
- Kompressionsraten zwischen 1 und 19 (je nach Datensatz)
- Kompressionsgeschwindigkeiten bis zu 2.9GiB/s (best-case)

Gliederung (Agenda)

1 Grundlagen

2 Funktionsweise

3 Anwendungsgebiete

4 LIVE-Beispiel

5 Benchmarks

LIVE-Beispiel

LIVE-Beispiel

Gliederung (Agenda)

1 Grundlagen

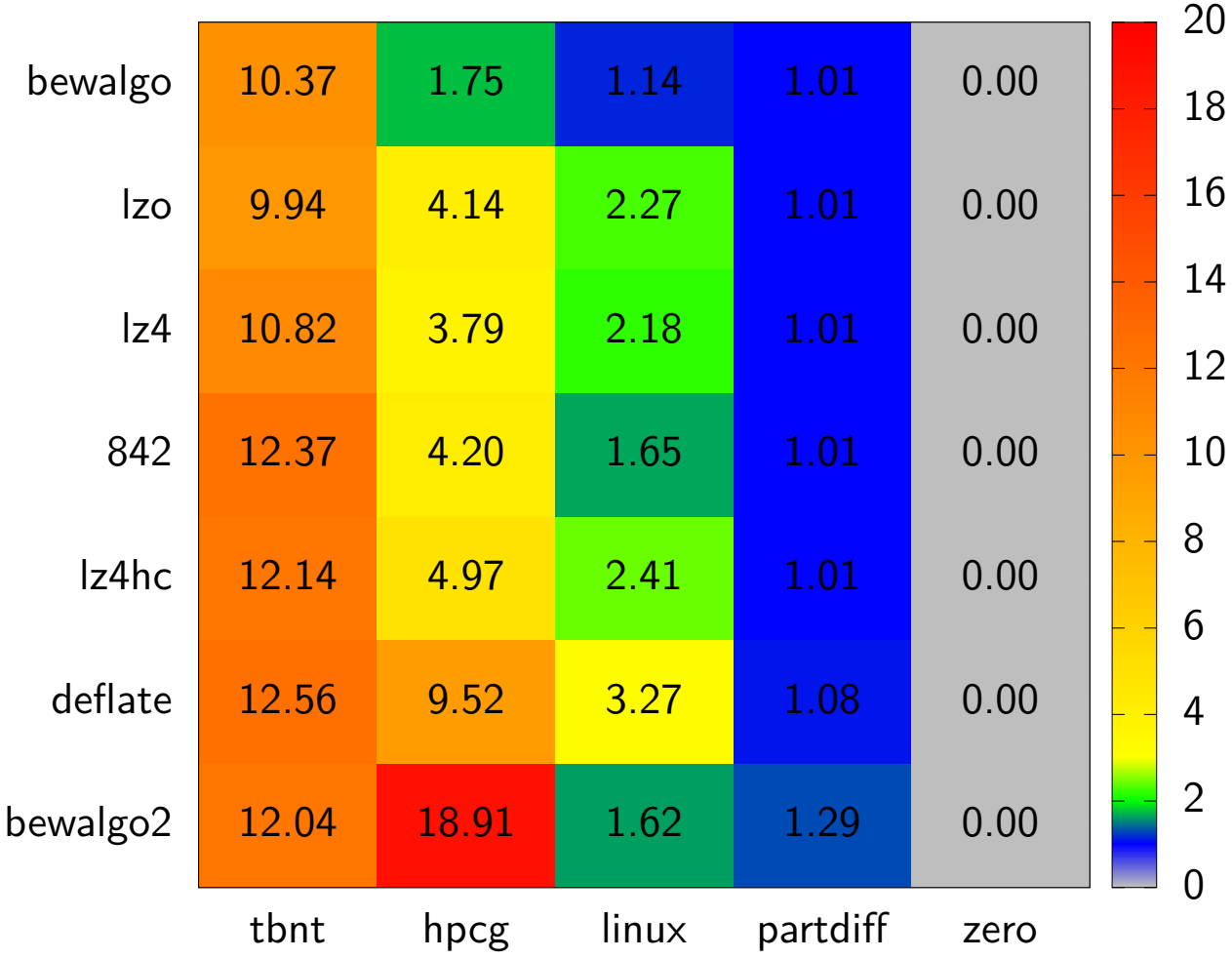
2 Funktionsweise

3 Anwendungsgebiete

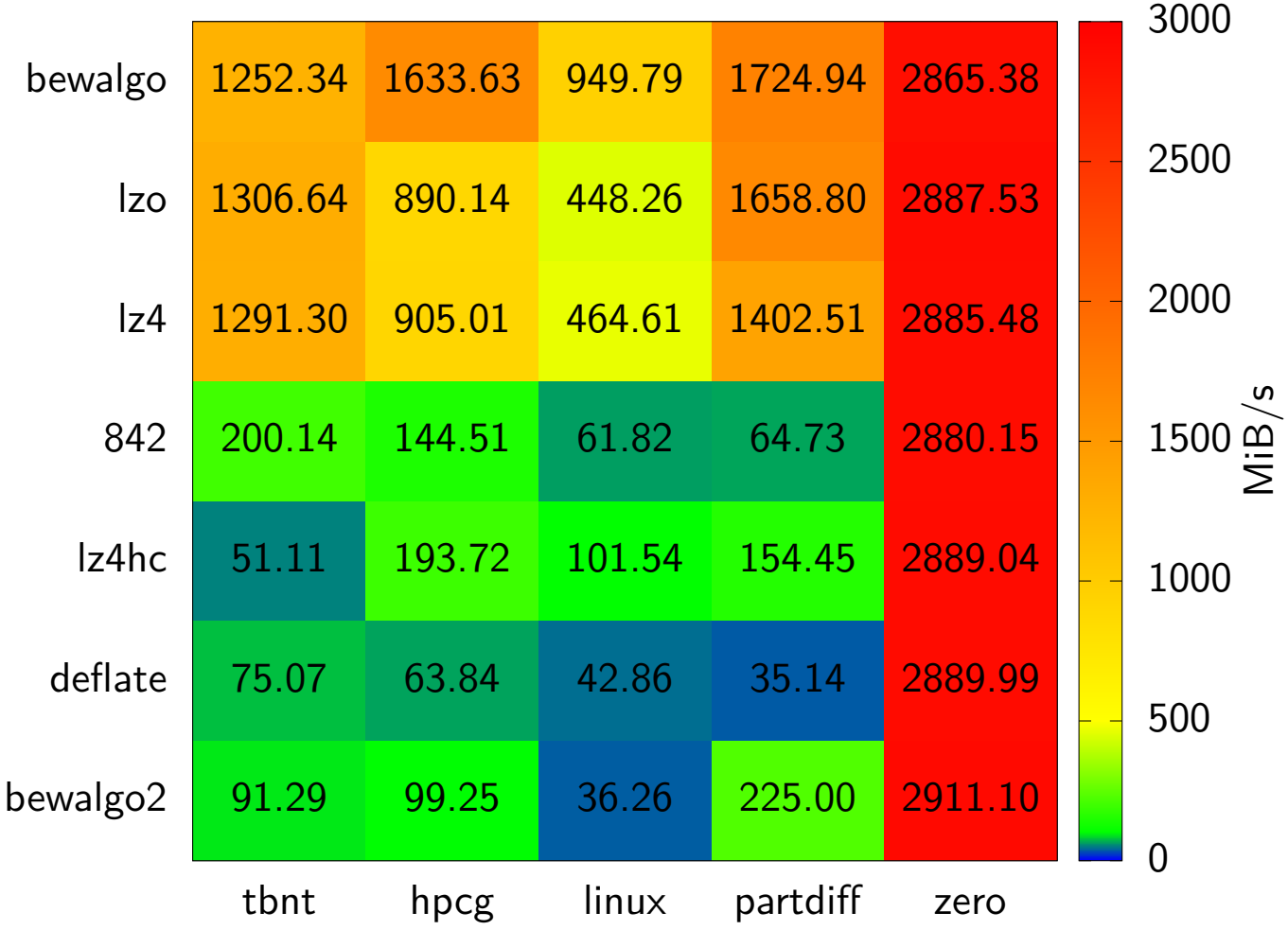
4 LIVE-Beispiel

5 Benchmarks

Benchmarks - Kompressionsrate



Benchmarks - Kompressionsgeschwindigkeit



Was bringt ZRAM für mich?

- Vorteile
  - mehr Arbeitsspeicher insbesondere bei low-memory Systemen
  - mehr Programme gleichzeitig ausführen
  - Programme mit hohen Arbeitsspeicher Anforderungen starten
  - relativ schneller swap
  - reduzierte Festplattenzugriffe
- Nachteile
  - CPU-Zeit
  - Energie-verbrauch
  - keine garantierte Komprimierbarkeit der Daten