

Cache-conscious Garbage Collection

Gernot Inthaler

Cache Architektur

Blockgröße

- 4 bis 128 Bytes

Assoziativität

- Direct mapped
- Voll assoziativ
- Mengenassoziativ

Schreibstrategien

Block ist im Cache

- Write through
- Write back (copy back)

Block ist nicht im Cache

- Write allocate (fetch-on-write)
- Write no allocate (write-around)

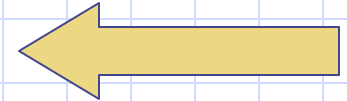
Typische Kombinationen

Write through – Write no allocate

Write back – Write allocate

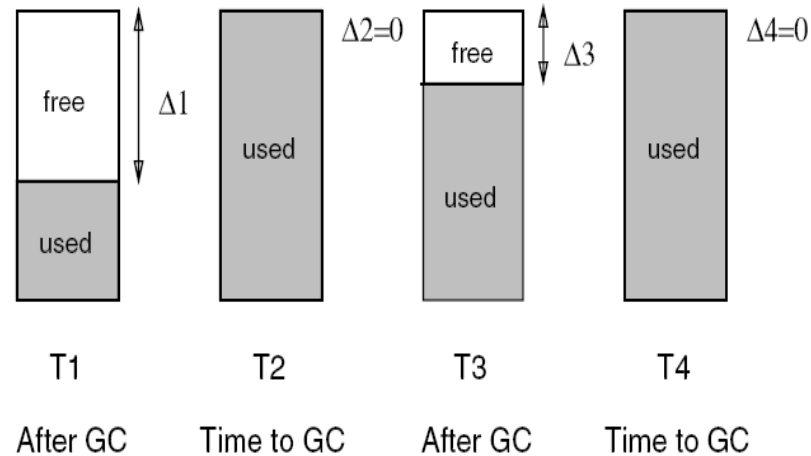
Verbesserung der Cache Performance

- Reduzierung der Fehlzugriffsrate
- Reduzierung der Strafe (miss penalty)
- Reduzierung der Zugriffszeit



Collection Policies

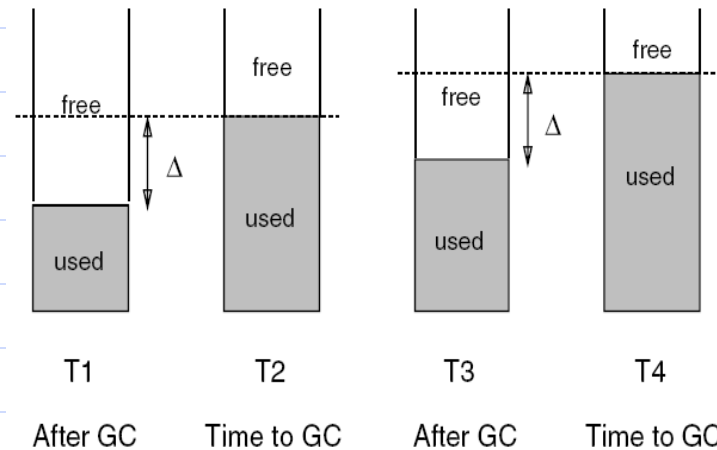
Fixed space collection



Garbage Collector wird erst gestartet wenn der Speicher voll ist

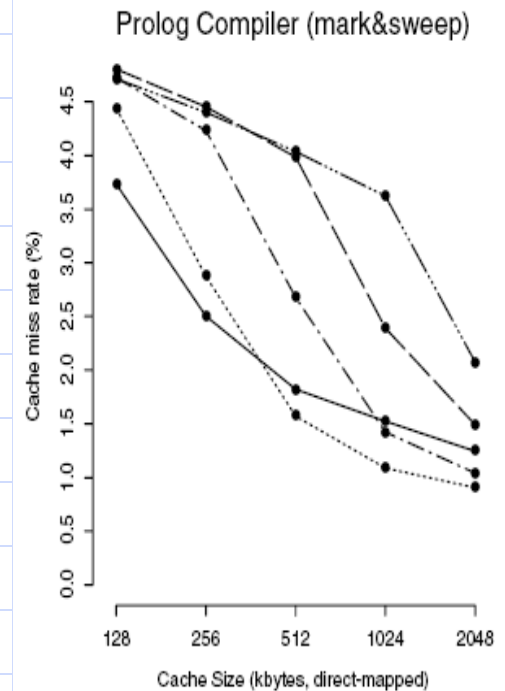
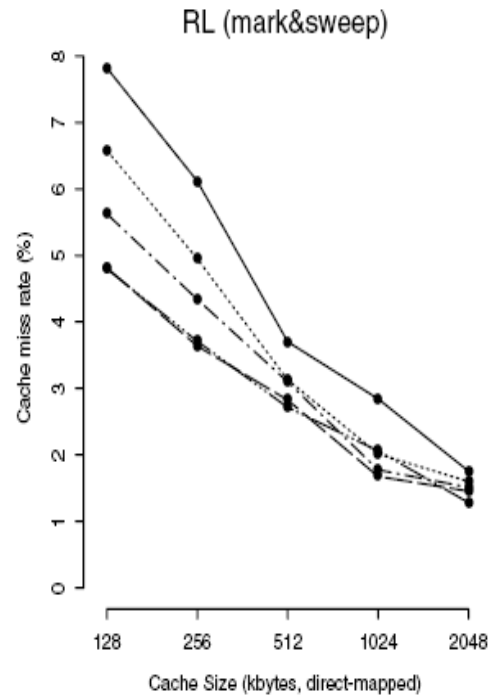
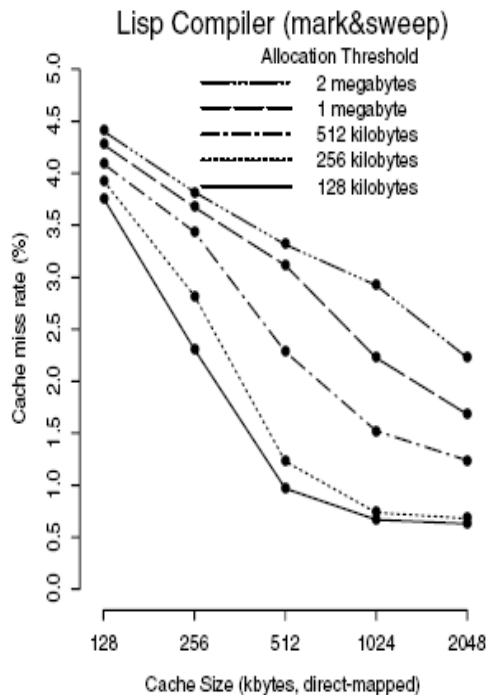
Collection Policies

Allocation threshold collection



Garbage Collector wird nach einer bestimmten Menge allokiertem Speichers gestartet

Mark & Sweep

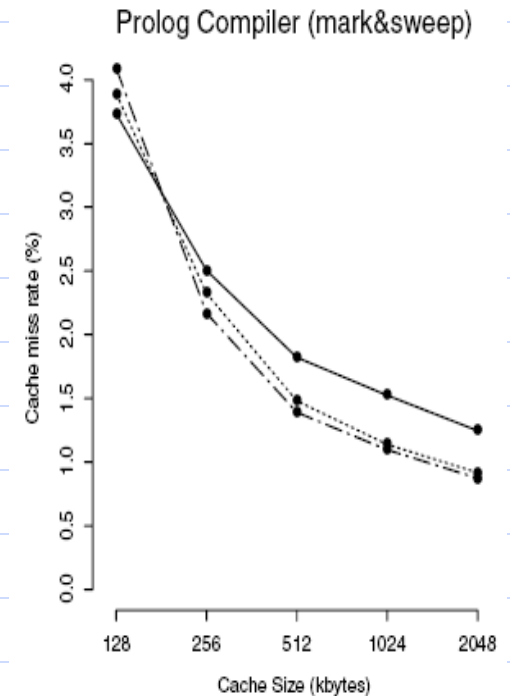
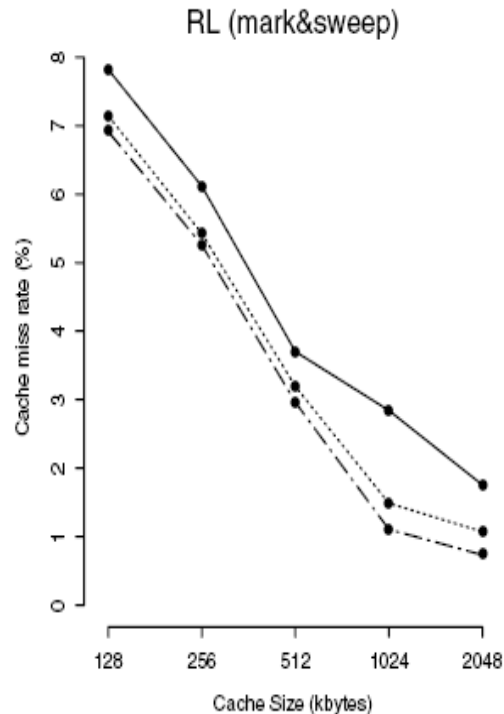
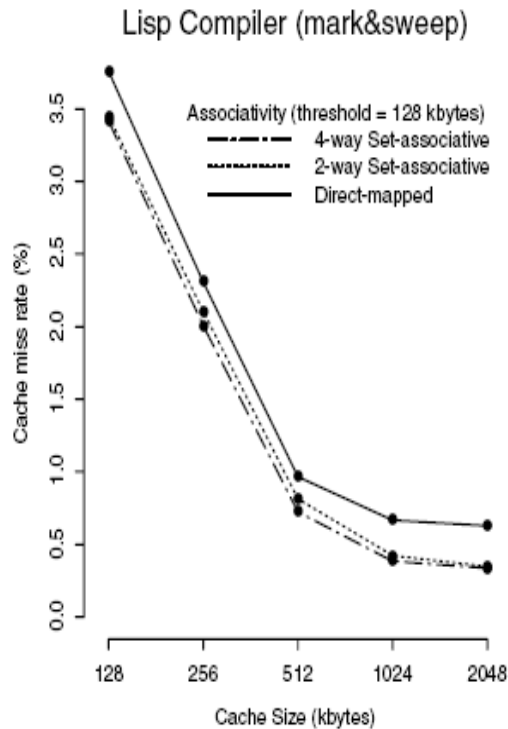


Fehlerzugriffsrate für verschiedene Cacherößen und unterschiedlicher Allokationsschwelle, direct mapped Cache

Mark & Sweep

- Größerer Cache → bessere Fehlerrate
- Verbesserung wenn Programm in den Cache „passt“
- Richtige Wahl der Allokations-Schwelle

Mark & Sweep

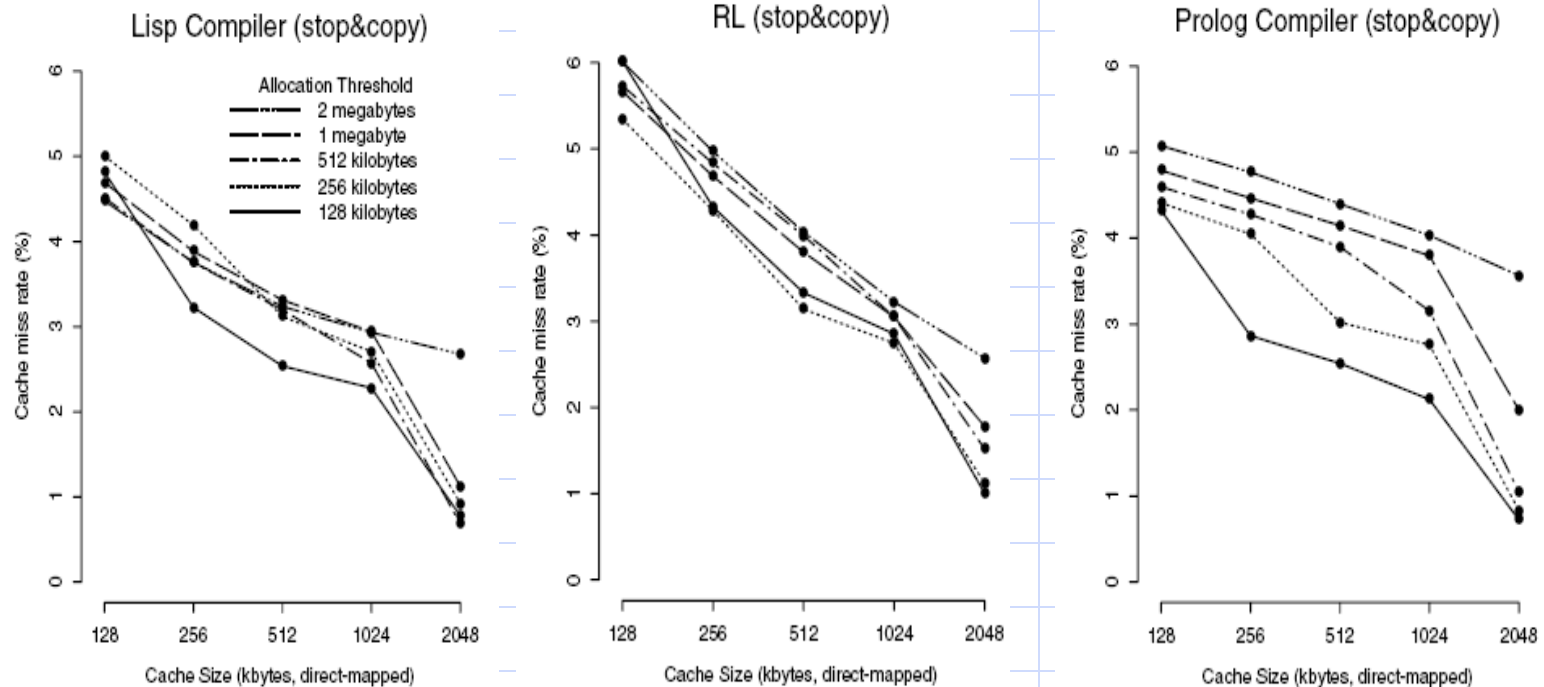


Fehlerzugriffsrate für verschiedene Cacherößen und unterschiedlicher Allokationsschwelle, mengenassoziativer Cache

Mark & Sweep

- Größerer Cache → bessere Fehlerrate
- Verbesserung durch mengenassoziativen Cache gering
- Unterschied zwischen 2-Weg mengenassoziativ und 4-Weg mengenassoziativ vernachlässigbar

Copying collection

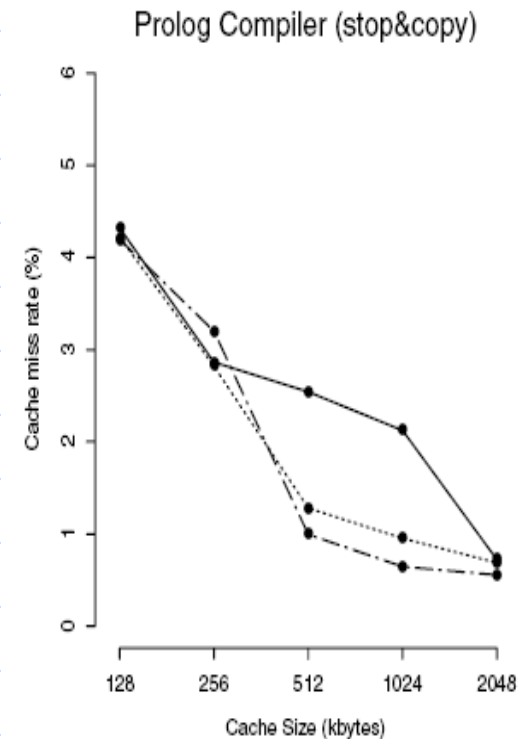
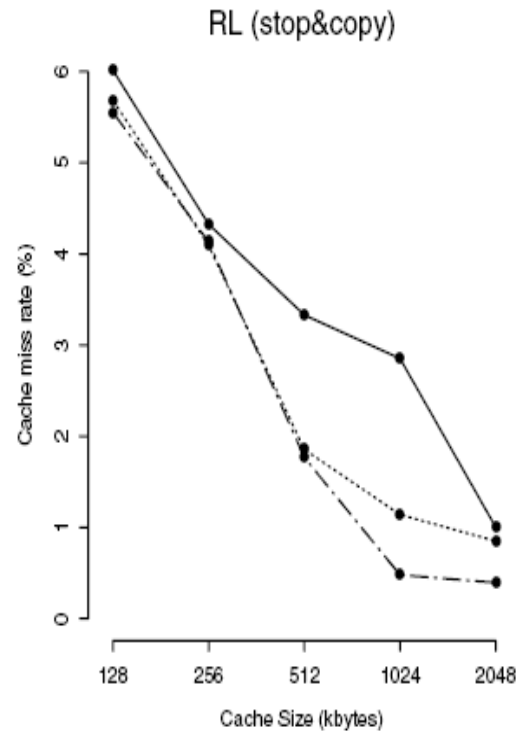
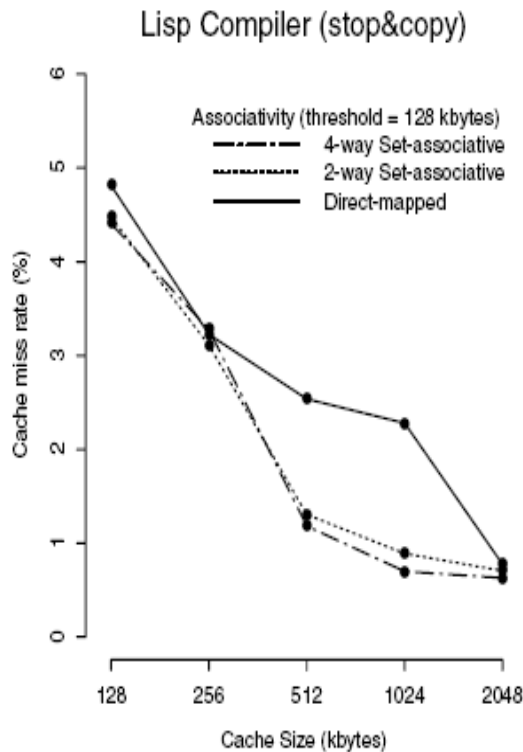


Fehlerzugriffsrate für verschiedene Cachegrößen und unterschiedlicher Allokationsschwelle, direct mapped Cache

Copying Collection

- Größerer Cache → bessere Fehlerrate
- Einfluß der Allokationsschwelle geringer als bei Mark & Sweep
- Fehlerzugriffsrate höher als bei Mark & Sweep

Copying Collection



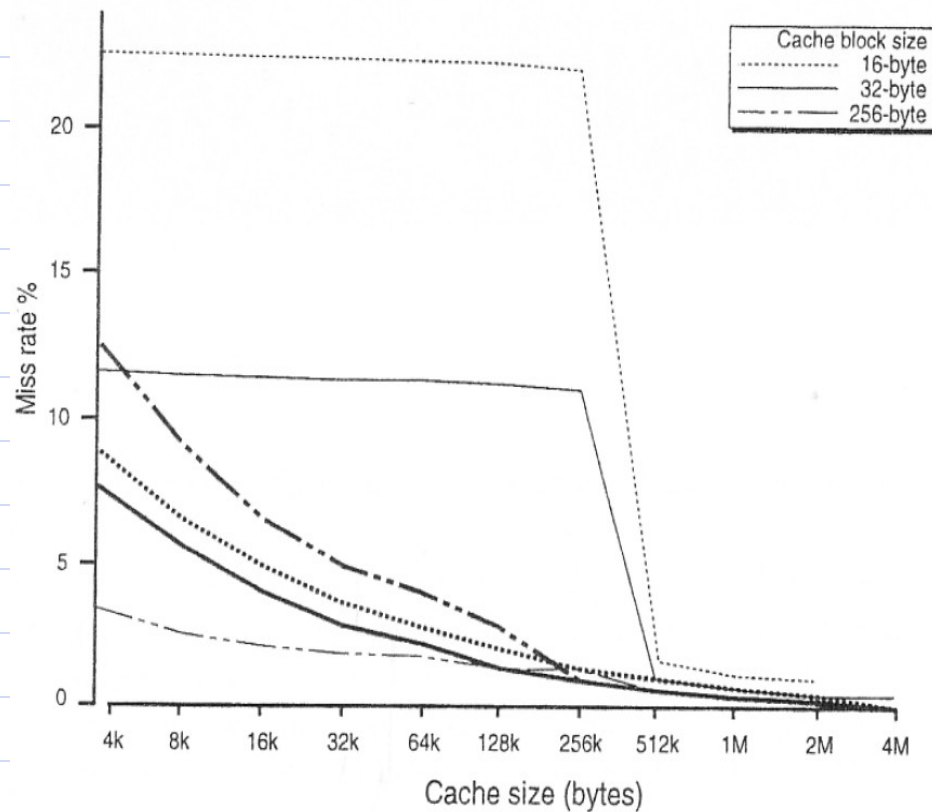
Fehlerzugriffsrate für verschiedene Cacherößen und unterschiedlicher Allokationsschwelle, mengenassoziativer Cache

Copying collection

- Größerer Cache → bessere Fehlerrate
- Verbesserung durch mengenassoziativen Cache groß (bei mittlerer Cache Größe)
- Unterschied zwischen 2-Weg mengenassoziativ und 4-Weg mengenassoziativ vernachlässigbar

Blockgröße

Einfluß der Cache Blockgröße am Beispiel eines Copying collectors

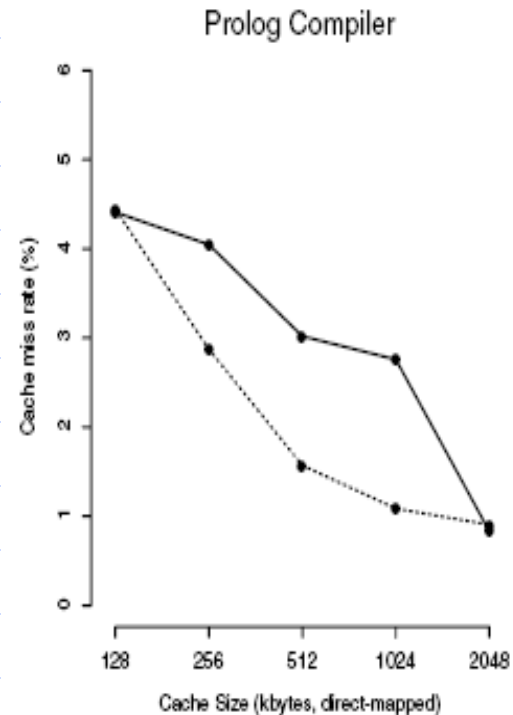
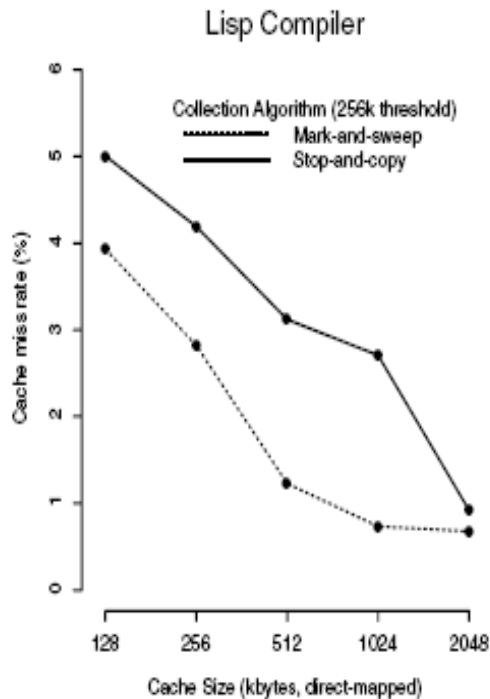


Größerer Block



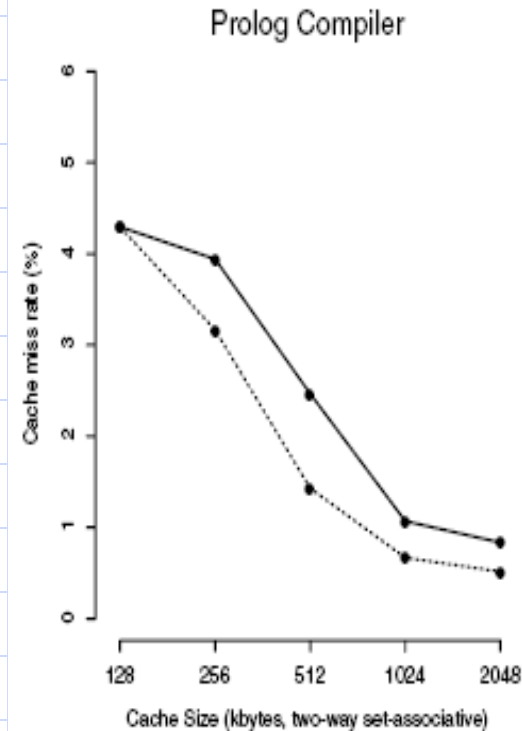
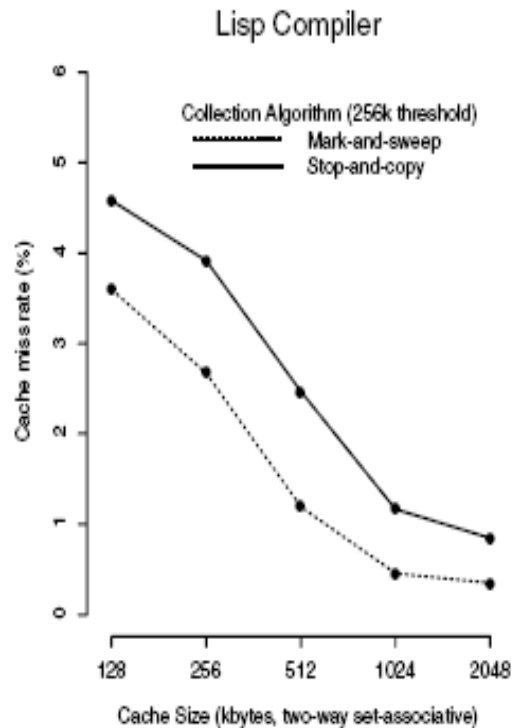
bessere Fehlerrate

Mark & Sweep – Copying Collection



Allokationsschwelle 256 kB
Direct mapped cache

Mark & Sweep – Copying Collection



Allokationsschwelle 256 kB
2-Weg mengenassoziativer Cache

Zusammenfassung

- Vergrößerung des Cache reduziert die Fehlerzugriffsrate
- Verwendung eines mengenassoziativen Speichers bei Copying Collection sinnvoll
- Vergrößerung der Cache Blockgröße reduziert die Fehlerzugriffsrate
- Mark & Sweep ergeben bessere Fehlerzugriffsraten als Copying collection



Vielen Dank für Ihre Aufmerksamkeit !