

## Rechnerarchitektur Projekt: Cache Simulator

Der Simulator soll Caches mit verschiedenen Einstellungen simulieren können und dabei verschiedene Trace-Dateien verwenden. Dies funktioniert, und auch das Ändern der Simulationskonfiguration beeinflusst in der Regel die Ergebnisse der Simulation wie erwartet. Zum Beispiel führt ein größerer Cache zu weniger Misses und ähnlichem. Im Folgenden werden die einzelnen Trace-Dateien sowie die Auswirkungen der verschiedenen Parameter auf die Ergebnisse genauer untersucht.

Alle Trace-Dateien wurden zuerst mit einer Standardkonfiguration simuliert, und anschließend wurden die einzelnen Parameter jeweils verändert, um zu sehen, welchen Effekt dies auf das Ergebnis hatte.

Die Standardkonfiguration war:

Anzahl der Cache-Blöcke: 1024, Größe der Blöcke: 32, Assoziativität: 4

Ersetzungsrichtlinie: LRU, Schreibrichtlinie: Write Allocate

Bei der Auswertung der Ergebnisse lag der Fokus hauptsächlich auf dem Effekt auf die Miss-Rate.

### **art.trace**

Die Miss-Rate mit der Standardkonfiguration betrug 21,46 %. Durch die Verwendung von mehr Cache-Blöcken konnte die Miss-Rate leicht gesenkt werden, aber nicht signifikant. Mit 4096 Blöcken hatte der Cache eine Miss-Rate von 21,15 %, was nur eine Verbesserung um 1,44 % bei einem viermal so großen Cache bedeutete.

Die Größe der Blöcke hatte einen deutlicheren Effekt auf die Reduzierung der Miss-Rate. Mit einer Blockgröße von 64 sank die Miss-Rate auf 18,84 %. Wenn der Cache erneut um das Vierfache vergrößert wurde, diesmal durch die Blockgröße, sank die Miss-Rate um 29,4 % auf 15,14 %.

Die Assoziativität hatte den stärksten Einfluss auf die Miss-Rate bei dieser Trace-Datei. Eine Verdopplung der Assoziativität führte fast zu einer Halbierung der Miss-Rate auf 11,193 %. Weitere Verdopplungen der Assoziativität zeigten keine abnehmenden Erträge. Eine Assoziativität von 16 führte zu einer Miss-Rate von 5,58 % und eine Assoziativität von 32 zu einer Miss-Rate von 2,82 %.

Der Unterschied bei den Ersetzungsrichtlinien war bei LRU und FIFO nicht sehr groß. FIFO hatte eine Miss-Rate von 22,7 %, was nur 5,79 % schlechter war als LRU. Im Vergleich dazu war Random mit einer Miss-Rate von 28,16 % deutlich schlechter, also 31,22 % schlechter als LRU.

Die Schreibrichtlinie "No Allocate" hatte eine Miss-Rate von 24,52 %, welche um 14,28 % schlechter war als "Allocate".

Es scheint, dass das Programm, von dem die Trace-Datei stammt, in der Regel Cache-Einträge verwendet, die nahe beieinander liegen, aber nicht unbedingt die gleichen Einträge oft hintereinander. Daher bestand kein großer Unterschied zwischen LRU und FIFO.

## **swim.trace**

Die Miss-Rate mit der Standardkonfiguration betrug 21,61 %. Durch die Verwendung von mehr Cache-Blöcken konnte die Miss-Rate leicht gesenkt werden, aber nicht signifikant. Mit 4096 Blöcken hatte der Cache eine Miss-Rate von 21,15 %, ähnlich wie bei der "art.trace" Trace-Datei.

Bei der Größe der Blöcke gab es große Unterschiede im Vergleich zur "art.trace". Die Miss-Rate sank mit einer Blockgröße von 64 auf 8,98 %. Mit einer Blockgröße von 128 sank die Miss-Rate sogar um beeindruckende 80,81 % auf 4,14 % im Vergleich zur Standardkonfiguration.

Die Assoziativität hatte im Vergleich zur "art.trace" einen geringeren Einfluss auf die Miss-Rate. Eine höhere Assoziativität führte nur zu minimalen Verbesserungen der Miss-Rate, wobei eine Assoziativität von 16 die beste Verbesserung von 0,05 % zeigte. Höhere Assoziativitäten hatten keinen signifikanten Einfluss auf die Miss-Rate.

Die Ersetzungsrichtlinien zeigten einen etwas größeren Unterschied zwischen LRU und FIFO. Die Miss-Rate stieg bei der Verwendung von FIFO um 19,19 % auf 25,76 %. Bei Random war der Anstieg mit 5,33 % auf 22,76 % geringer.

Bei den Schreibrichtlinien gab es deutlich größere Unterschiede. Die Miss-Rate bei "No Allocate" war um 108,56 % höher, also 45,07 %.

Bei der Trace-Datei scheinen die Cache-Einträge, auf die zugegriffen wird, noch näher beieinander zu liegen als bei der "art.trace". Es wird auch häufig auf Einträge zugegriffen, die kürzlich gespeichert wurden, ohne vorher in den Cache geladen worden zu sein. Dies führt zu einer drastischen Verschlechterung der Miss-Rate bei Verwendung der "No Allocate" Schreibrichtlinie.

## **trace.out**

Die Miss-Rate dieser Trace-Datei war mit der Standardkonfiguration mit Abstand am schlechtesten, bei 45,67 %. Durch die Verwendung von mehr Cache-Blöcken konnte die Miss-Rate leicht gesenkt werden, mit 2048 Blöcken um 6,51 % auf 42,70 % und mit 4096 Blöcken um 11,58 % auf 40,39 %.

Die Blockgröße spielte erneut eine wichtige Rolle und ermöglichte eine Verbesserung der Miss-Rate auf 35,29 % mit einer Blockgröße von 64 und sogar auf 19,13 % mit 128 Byte großen Blöcken, was eine Verbesserung von 58,07 % darstellt.

Auch die Assoziativität trug dazu bei, die Miss-Rate zu senken. Mit einer 8-fachen Assoziativität sank sie auf 38,53 % und mit einer 16-fachen Assoziativität auf 30,90 %, was einer Verbesserung von 32,34 % entspricht. Eine weitere Erhöhung der Assoziativität führte zu abnehmenden Erträgen.

Bei den Ersetzungsrichtlinien gab es einen großen Unterschied zwischen LRU und FIFO. FIFO wies eine Miss-Rate von 57,84 % auf, was um 26,60 % schlechter war. Random hingegen zeigte keine so drastische Veränderung und hatte eine Miss-Rate von 46,44 %.

Die Schreibrichtlinien wiesen ebenfalls deutliche Unterschiede auf. "No Allocate" war um 26,81 % schlechter als "Allocate" und hatte eine Miss-Rate von 57,92 %.

Die Trace-Datei scheint nicht oft auf die gleichen Adressen hintereinander zuzugreifen, wodurch es zu vielen Cache-Misses kommt, da die Einträge häufig überschrieben werden. Um eine niedrige Miss-Rate zu erreichen, muss ein sehr großer Cache verwendet werden, damit die Einträge nicht so häufig überschrieben werden.

## **crafty\_mem.trace**

Dies war im Kontrast zur "trace.out" die beste Datei mit der Standardkonfiguration, mit einer Miss-Rate von nur 1,80 %. Allerdings handelte es sich auch um die größte Trace-Datei mit etwa 10-20 Mal so vielen Cache-Zugriffen insgesamt im Vergleich zu den anderen Dateien. Durch die Verwendung von mehr Cache-Blöcken konnte auch hier die Miss-Rate deutlich reduziert werden, auf 1,05 % mit 2048 Blöcken und auf 0,4 % mit 4096 Blöcken, was einer Verbesserung von 77,78 % entspricht.

Die Blockgröße hatte einen etwas geringeren Einfluss auf die Miss-Rate im Vergleich zur Anzahl der Blöcke. Mit 64 Byte großen Blöcken erreichte man eine Miss-Rate von 1,38 %, und mit 128 Byte großen Blöcken sank die Miss-Rate auf 0,64 %. Das entspricht einer Verbesserung um 64,44 %.

Die Assoziativität hatte hingegen nur einen geringen Einfluss auf die Miss-Rate bei dieser Trace-Datei. Mit einer 8-fachen Assoziativität gab es nur eine Verbesserung von 6,67 % auf 1,68 % und mit einer 16-fachen Assoziativität eine Verbesserung von 17,78 % auf 1,48 %.

Die Ersetzungsrichtlinien hatten einen sehr großen Einfluss auf das Ergebnis. FIFO war um 148,89 % schlechter als LRU mit einer Miss-Rate von 4,48 %. Interessanterweise schnitt Random bei dieser Trace-Datei besser ab als die anderen beiden Ersetzungsrichtlinien, mit einer Miss-Rate von 1,60 %, also um 11,11 % besser als LRU.

Bei den Schreibrichtlinien gab es erneut einen etwas kleineren Unterschied, wobei "No Allocate" mit 1,47 % um 18,33 % besser abschnitt als "Write Allocate".

Diese Trace-Datei steht im Kontrast zur "trace.out"-Datei, bei der nicht oft die gleichen Cache-Einträge verwendet und häufig überschrieben wurden. Hier wurden die gleichen Cache-Einträge sehr häufig wiederverwendet und mussten somit nicht oft ersetzt werden. Dies lässt sich insbesondere daraus ableiten, dass die Ersetzungsrichtlinie LRU deutlich besser abschnitt als FIFO.

Abschließend kann festgestellt werden, dass die Trace-Dateien von verschiedenen Programmen stammen und jede Datei ihre eigenen Anforderungen an den Cache stellt und bestimmte Parameter der Konfiguration auf die Probe stellt. Daher ist es in der Praxis nicht einfach, einen optimalen Cache zu erstellen, da er universell einsetzbar sein muss und verschiedene Anwendungen unterschiedliche Anforderungen an einen Cache haben können.