Cameron Calv and Nicholas Sica
ECEC 621 High-Performance Computer Architecture
Project Three – Implementing Cache Replacement Policy and evaluating its performance using AI Workloads.

Evaluating the LRU and LFU Replacement Policies

The Least Frequently Used (LFU) replacement policy updates the cache by removing the least frequently used entry and replacing it with the next. A frequency counter is implemented to keep track of how often a cache entry is a hit and the entry with the lowest counter value is booted out. The Least Recently Used (LRU) replacement policy updates the cache by removing the least recently used entry and replacing it with the next. A counter increments every clock cycle and the entry with the largest counter value is booted out.

Hit rate is used to evaluate how often hits of the cache occur as oppose to misses. Various values of the number of associative caches were tested as well as the cache size to on the three AI workloads to determine what produced the greatest hit rate.

Table 1: Various hit rate values for the LRU replacement policy. Worst hit rates highlighted in red and best hit rates highlighted in green.

N-Associatives	Cache Size (kBytes)	Sample	Deepsjeng	Leela	Exchange2
4	128	42.580000%	76.939192%	42.554451%	99.926982%
4	256	56.600000%	88.972509%	70.743799%	99.984771%
4	512	65.070000%	93.843631%	92.363097%	99.985094%
4	1024	65.580000%	95.515018%	98.220996%	99.985094%
4	2048	65.600000%	95.906338%	99.455119%	99.985094%
8	128	42.780000%	79.376751%	42.586477%	99.958232%
8	256	60.870000%	90.789126%	75.736557%	99.985094%
8	512	65.410000%	94.859542%	94.443418%	99.985094%
8	1024	65.600000%	95.682981%	98.867637%	99.985094%
8	2048	65.600000%	95.946182%	99.596940%	99.985094%
16	128	43.740000%	81.195524%	43.731201%	99.985094%
16	256	62.250000%	91.505108%	78.815214%	99.985094%
16	512	65.600000%	95.036707%	96.020614%	99.985094%
16	1024	65.600000%	95.751690%	99.011727%	99.985094%
16	2048	65.600000%	95.958330%	99.627833%	99.985094%

Table 2: Various hit rate values for the LFU replacement policy. Worst hit rates highlighted in red and best hit rates highlighted in green.

N-Associatives	Cache Size (kBytes)	Sample	Deepsjeng	Leela	Exchange2
4	128	47.520000%	75.302526%	45.554564%	86.318316%
4	256	59.450000%	88.242389%	69.851638%	99.984948%
4	512	65.300000%	93.506566%	89.236583%	99.985094%
4	1024	65.600000%	95.333935%	96.634206%	99.985094%
4	2048	65.600000%	95.813472%	99.132972%	99.985094%
8	128	50.740000%	75.948467%	47.165396%	72.666252%
8	256	63.420000%	88.800169%	71.792928%	99.985094%
8	512	65.520000%	93.951883%	87.7.36296%	99.985094%
8	1024	65.600000%	95.036979%	96.410854%	99.985094%
8	2048	65.600000%	95.603750%	99.282395%	99.985094%
16	128	51.690000%	74.911263%	46.780800%	99.985094%
16	256	64.840000%	88.167742%	71.405116%	99.985094%
16	512	65.600000%	93.180060%	85.241210%	99.985094%
16	1024	65.600000%	94.467310%	94.451335%	99.985094%
16	2048	65.600000%	95.212336%	99.058609%	99.985094%

It seems that as a general rule, increasing the number of associated caches increases the hit rate up until a certain saturation point. The same general rule applies for cache size as well. Some workloads saturated much earlier than others and it appears that the hit rate curves defined by the variables of N and cache size peak at a particular value on other workloads such as Deepsjeng and Leela.