4 Multicore Cache Partitioning [55 points]

Suppose we have a system with 32 cores that share a physical second-level cache. Assume each core is running a single single-threaded application, and all 32 cores are concurrently running applications. Assume that the page size of the architecture is 8KB, the block size of the cache is 128 bytes, and the cache uses LRU replacement. We would like to ensure each application gets a *dedicated* space in this shared cache without any interference from other cores. We would like to enforce this using the OS-based page coloring mechanism to partition the cache, as we discussed in lecture. Recall that with page coloring, the operating system ensures, using virtual memory mechanisms, that the applications do not contend for the same space in the cache.

(a) [10 points] What is the minimum size the L2 cache needs to be such that each application is allocated its dedicated space in the cache via page coloring? Show your work.

256KB.

Explanation:
For OS based page coloring to work in this case, we need at least 32 colors. This means we need at least 5 bits of the cache index to intersect with the physical page number.

<table>
<tr><td>Cache line Tag</td><td colspan="2">Cache Index</td><td>Bytes in Block</td></tr>
<tr><td></td><td>5 bits</td><td colspan="2"></td></tr>
<tr><td colspan="2">Physical Page Number</td><td colspan="2">Page Offset</td></tr>
</table>

So, with associativity A, page size P, the minimum cache size is given by,

$$C \geq A \times 2^5 \times P = A \times 32 \times P$$

$$C \geq A \times 32 \times 8KB = A \times 256KB$$

Minimum cache size (associativity = 1) is 256KB

(b) [10 points] Assume the cache is 4MB, 32-way associative. Can the operating system ensure that the cache is partitioned such that no two applications interfere for cache space? Show your work.

No.

Explanation:
For a given associativity, minimum cache size = A × 256KB (from part a). Therefore, for a 32-way associative cache, minimum cache size required for the OS to ensure partitioning without interference is 32 × 256KB = 8MB. Since the cache size is only 4MB, the OS, in this case, cannot ensure partitioning without interference.

(c) Assume you would like to design a 32MB shared cache such that the operating system has the ability to ensure that the cache is partitioned such that no two applications interfere for cache space.

(i) [5 points] What is the minimum associativity of the cache such that this is possible? Show your work.

Minimum associativity = 1.

Explanation:
From part a),
$C \geq A \times 256KB$
$32000KB \geq A \times 256KB$
Therefore, minimum associativity = 1

(ii) [10 points] What is the maximum associativity of the 32MB cache such that this is possible? Show your work.

Maximum associativity = 128.

Explanation:
From part a),
$C \geq A \times 256KB$; $A \leq C / 256KB$; $A \leq 32MB / 256KB$
$A \leq 128$
Therefore, maximum associativity is 128.

(d) [5 points] Suppose we decide to change the cache design and use utility based cache partitioning (UCP) to partition the cache, instead of OS-based page coloring. Assume we would like to design a 4MB cache with a 128-byte block size. What is the minimum associativity of the cache such that each application is guaranteed a minimum amount of space without interference? Recall that UCP aims to minimize the cache miss rate by allocating more cache ways to applications that obtain the most benefit from more ways, as we discussed in lecture.

Minimum associativity = 32.

Explanation:
Utility based cache partitioning needs to give at least one way for each application. Otherwise, the application will receive no cache space. Hence, the minimum associativity is 32.

(e) [5 points] Is it desirable to implement UCP on a cache with this minimum associativity? Why, why not? Explain.

No, it is not desirable to implement UCP.

Explanation:
There will be no benefit gained from UCP since UCP guarantees at least one way per application. This means all applications will be allocated exactly one way of the cache, i.e. the cache is equally and statically partitioned regardless of applications' utility for caching.

(f) [5 points] What is the maximum associativity of a 4MB cache that uses UCP such that each application is guaranteed a minimum amount of space without interference?

32k ways.

Explanation:
The maximum associativity corresponds to a fully associative design. For the given configuration, it is 4 MB / 128 bytes = $2^{22} / 2^{7} = 2^{15}$ = 32k ways.

(g) [5 points] Is it desirable to implement UCP on a cache with this maximum associativity? Why, why not? Explain.

No.

Explanation:
It is not desirable to implement UCP with this maximum associativity because the overhead of UCP for 32 applications on this cache will likely outweigh its benefits. UCP will only work with LRU replacement policy. But implementing LRU on top of a 32k-way cache is impractical. Also the number of counters needed by UCP and the partitioning solution space for UCP are very large for such a cache.