IX. Appendix

To illustrate our approach for computing LCBs, consider the following example. Assume we have two tasks, τ_1 and τ_2 with UCBs and ECBs for each listed in Figure **??**.

Task ID	Evicting Cache Blocks $ECB(\delta_i^j)$				
	δ_i^1	δ_i^2	δ_i^3	δ_i^4	δ_i^5
τ_1	{1,2}	{3,4,8}	{4,5,6,8}	{1,2,7,8}	{1,2,7,8}
τ_2	{1,9}	{3,10}	{11,12}	{5,7,13}	{1,3,7,8}

Task ID	Useful Cache Blocks $UCB_{out}(\delta_i^j)$				
	δ_i^1	δ_i^2	δ_i^3	δ_i^4	δ_i^5
τ_1	{1,2}	{1,2,4,8}	{1,2,8}	{1,2,7,8}	{1,2,7,8}
τ_2	{1}	{1,3}	{1,3}	{1,3,7}	{1,3,7,8}

Fig. 9: Taskset ECBs and UCBs.

The computation for LCBs uses the accessed useful cache blocks (AUCBs) since the cache blocks that are re-loaded during execution of the non-preemptive region between preemption points is a function of the memory that is explicitly accessed by the preempted task. The computed AUCBs for each task is shown in Figure **??**. In accordance with Equation 7 one can readily see that the AUCBs are simply the intersection of the UCBs and ECBs for each basic block.

Task ID	Accessed Useful Cache Blocks $AUCB_{out}(\delta_i^j)$				
	δ_i^1	δ_i^2	δ_i^3	δ_i^4	δ_i^5
τ_1	{1,2}	{4,8}	{8}	{1,2,7,8}	{1,2,7,8}
τ_2	{1}	{3}	{1,3}	{7}	{1,3,7,8}

Fig. 10: Taskset AUCBs.

In our example, assume preemptions are taken at basic blocks δ_1^2 and δ_1^4 for task τ_1. For simplicity, we calculate the LCBs associated with these two preemption points. For $LCB(\delta_1^2, \delta_1^4)$, we have $UCB(\delta_1^2) = \{1, 2, 4, 8\}$. The second expression is the set of memory that is accessed in basic blocks δ_1^3 and δ_1^4, namely $\{8\} \cup \{1, 2, 7, 8\} = \{1, 2, 7, 8\}$ comprising the set of AUCBs. The third expression is the set of ECBs for task τ_2 where $ECB(\tau_2) = \{1, 3, 5, 7, 8, 9, 10, 11, 12, 13\}$. Thus, $LCB(\delta_1^2, \delta_1^4)$ is given by the intersection of the three sets:

$$\begin{aligned} LCB(\delta_1^2, \delta_1^4) =& \{1, 2, 4, 8\} \cap \{1, 2, 7, 8\} \cap \\ & \{1, 3, 5, 7, 8, 9, 10, 11, 12, 13\} = \{1, 8\} \end{aligned}$$

The preemption cost $\gamma(\delta_1^2, \delta_1^4)$ for a BRT = 390μs is given by:

$$\gamma(\delta_1^2, \delta_1^4) = |\{1, 8\}| \cdot 390 = 780\mu s$$

Using the same method, $LCB(\delta_1^4, \delta_1^5)$ is given by:

$$\begin{aligned} LCB(\delta_1^4, \delta_1^5) =& \{1, 2, 7, 8\} \cap \{1, 2, 7, 8\} \cap \\ & \{1, 3, 5, 7, 8, 9, 10, 11, 12, 13\} = \{1, 7, 8\} \end{aligned}$$

The preemption cost $\gamma(\delta_1^4, \delta_1^5)$ for a BRT = 390μs is given by:

$$\gamma(\delta_1^4, \delta_1^5) = |\{1, 7, 8\}| \cdot 390 = 1170\mu s$$

Using the method illustrated here, the preemption cost matrix entries for each pair of basic blocks are computed in a similar fashion and used as input to our preemption point placement algorithm.

To further exemplify the interdependence of preemption points, consider the example shown below. In order to account for all re-loaded cache blocks (LCBs), preemptions are always included at the first basic block δ_i^0 and the last basic block $\delta_i^{N_i}$ as shown in Figure **??**. This is commensurate with the preemptions that occur before and after the task executes. Assume we have two tasks where τ_2 contains four basic blocks which may be preempted by task τ_1. For simplification, assume that the ECBs of task τ_1 evicts all UCBs of task τ_2. Let us further assume that we have $\rho_2 = \{\delta_2^0, \delta_2^1, \delta_2^3\}$. Using our LCB computation approach, only the re-loaded lines as captured in the terms $LCB(\delta_2^0, \delta_2^1)$ and $LCB(\delta_2^1, \delta_2^3)$ are included in the C_2 computation.

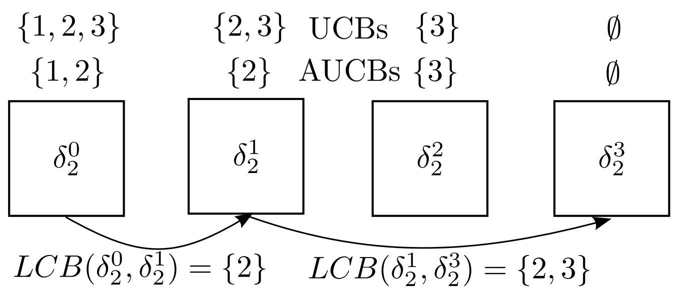

Fig. 11: LCB Interdependence

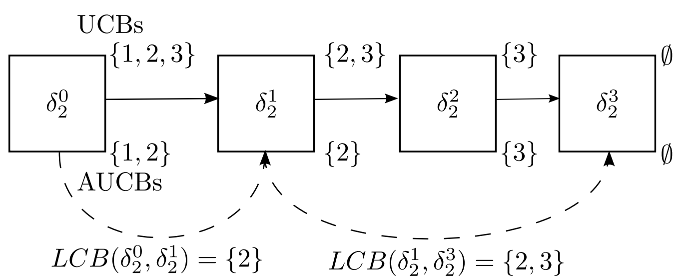

Fig. 12: LCB Interdependence