Contribution ID: **297** Type: **not specified**

Cooperation between CPU and system level cache by using MPAM

MPAM enables fine-grained control over shared resources such as CPU caches, memory bandwidth, and interconnect bandwidth. In a typical memory hierarchy, the data path looks like this:
CPU(L2/L3)/GPU/NPU/... → NoC → SLC → DDR
This structure introduces several components including System Level Cache between clients and DDR memory. This raises an important question: how can CPU caches and system-level caches be coordinated to optimize overall data path efficiency?
Modern hardware supports tagging each transaction with a client-specific identifier. These tags allow the monitoring unit to track transaction behavior across the system. A userspace daemon can read this monitoring data and, based on current workload scenarios, dynamically adjust cache policies using MPAM. For example, in gaming scenarios where GPU performance is critical, we can create a dedicated MPAM partition for game-related threads, increase the cache allocation for these threads in the CPU cache to prevent interference from other processes and adjust system-level cache usage to prioritize GPU-related data flows.
This can help to achieve better performance and lower power cost. In general, monitoring data plays a crucial role in informing cache policy decisions, enabling adaptive and scenario-specific optimizations across the memory hierarchy.

Primary author: HUANG, Yiwei

Presenter: HUANG, Yiwei

Session Classification: Android MC

Track Classification: Android MC