Department of Computer Science

St. Francis Xavier University

Presents

An Adaptive QoE-Driven DVFS: A Comprehensive Approach to Enhancing User Experience and Optimizing Power Across Concurrent Workloads

by

Aravind Kashyap Kaveripakkam Raghuraman

St. Francis Xavier University

M.Sc. Thesis Proposal Presentation

December 16, 2024 @ 10:00am

Annex 113

Dynamic Voltage and Frequency Scaling (DVFS) is one of the most common techniques used for manipulating the frequency of a processor according to the needs of the application workload that is being run on the said system by the user. Common techniques often try to address this challenge by matching the frequency of the processor to the demands of the system, often in terms of utilization seen when the applications are run. The main goal here is to save the power and reduce the heat generation while optimizing the frequency with which the processor runs in order to accommodate the workload. These techniques often make sure that the performance of the application workload is not being compromised in the seek for saving the power, reducing temperature or the additional frequency of the processor. One familiar term that is used while addressing the performance of the applications is QoE (Quality of Experience).

Recent practices, including the most advanced frameworks like GearDVFS [6] and zTT [5], often focus only on Frames Per Second (FPS) as QoE quantifying metrics in their experiments. While FPS is a significant indicator of user experience for certain applications like video playback or gaming, it fails to capture or generalize in a broader spectrum of defining the user experience requirements across various varieties of application workloads. For example, to a user telecommunicating using Skype or Zoom, maintaining the lowest achievable latency and constant jitter is more important to ensure his Quality of Experience than FPS. Or, for a person wanting to book the flight tickets on his mobile phone, we need to prioritize quick page loading times and the least time-to-first-byte (ttfb) rather than the visual quality of the page with FPS. In the above two of many real-time scenarios, we see that FPS does not play any critical role in defining rich QoE, and for a user who is running multiple applications in the foreground with some of them still inactive in the background, when he wants to switch to a background application again, often like google tabs, it should be back with a quick response, where loading time and latency are quite very essential QoE quantifiers than others.

As discussed, in scenarios where multiple applications are simultaneously run, and all of them have a completely different set of QoE requirements and metrics, this complexity requires a more QoE-driven DVFS that should dynamically prioritize and strike a perfect balance between the different QoE requirements of all the applications that are being run on the system. Critical Metrics such as page load times, jitter, buffering time, caching performance, time to first byte and resolution must be considered alongside FPS and latency to provide a holistic measure of QoE. One more very important scenario where both GearDVFS and zTT fall short is when dealing with multiple concurrently running applications that each has distinct and complex QoE needs, and both of them do not keep the user's interests in mind while coming up with a governor that meets this criterion. For instance, zTT, relies on a lightweight Q-network structure and specified user-defined QoE targets, just in terms of frames per second to optimize the QoE of a **single**, active foreground application. However, it is not at all adaptable enough to handle the demands of even more than one application at once.

To address this challenge, our approach proposes an advanced DVFS mechanism that incorporates an extended set of QoE metrics tailored to different application types. By dynamically profiling and weighting these metrics—such as FPS, latency, jitter, buffering time, cache hit and miss latencies, page load times, TTFB, and video resolution—our DVFS model aims to maintain optimal performance across diverse and concurrent workloads. This comprehensive strategy ensures that user satisfaction is upheld, even when multiple applications with varying QoE types and demands are running simultaneously, thereby elevating the overall user experience while still achieving the goals of power efficiency and thermal management. Furthermore, our model uses techniques to dynamically take into account the user-specific requirements in terms of priority of each application and each of the QoE metrics in an application while trying to maximize and enhance his personal experience instead of the one-size-fits-all approach.

Power efficiency also being a very critical aspect of our DVFS design, which is often undermined by agnostic approaches like GearDVFS and Schedutil, which pushes the CPUs to maintain arbitrary utilization levels (e.g., 80%) or by aggressive scaling strategies in OnDemand governors. These methods waste energy by either over-allocating resources or reacting excessively to transient demands. Our model overcomes this by dynamically adjusting CPU and GPU frequencies based on application-specific QoE metrics and real-time priorities. This ensures a fine-grained balance between power consumption and performance, avoiding unnecessary energy expenditure while enhancing user satisfaction across diverse and concurrent workloads.