Enhancement of the Dynamic Memory Management for Multimedia Applications

David Atienza Alonso

Abstract

In current day Multimedia applications – such as interactive systems, video conferencing or games – a high demand of memory and performance is required. It results in highly-cost and highly-power consumption systems whereas a superior portability demands cheap and low-power consumption ones. Moreover, the dynamic memory subsystem is currently one of the main sources of power consumption and an inattentive management of this dynamic memory can lessen severely the performance of the whole system. Thus, an optimal management of the dynamic memory has become a fundamental part in the development of the previously mentioned kind of applications. Nonetheless, the enhancement of the usage of the aforementioned dynamic memory is a main issue that, oftentimes, it is not considered as important as it deserves.

In point of fact, on the one hand, programmers use frequently current general-purpose memory allocators to reduce the amount of memory required – a large number of different allocation policies exist presently in the literature and a vast space of mechanisms that can support them likewise –. Nevertheless, they do not take into account either the specific behaviour pattern of the final application or the final architecture of the system to choose the most appropriate one. As a result, they do not attain the expected outcome and the most common solution is the removal of the dynamic memory in the system, and all data become static, with an obvious wastage of power and memory footprint.

On the other hand, other designers and programmers use custom memory managers – a difficult and error-prone process – hoping to accomplish important performance and cost improvements based upon their own experience. However, they do not take into consideration the power consumption of the system and the point is that the possible trade-off between these two important factors, namely performance and memory consumption, is overlooked completely. Moreover, although the programmers write their own ad hoc dynamic memory allocators, the result is only as optimal as it should be if some years of intense research are devoted to tuning it carefully.

As a consequence of the aforementioned deficiencies, the work presented here explores the design of an optimal and fast tuneable dynamic memory manager for multimedia applications, from a systematic approach. Therefore, the definition of a clear and methodical exploration of the possible-solutions-space is required. Moreover, eventually, it will imply a reasonably fast process design and the selection of the most suitable platform – either pure hardware, software or an intermediate solution – for the resulting system.