Virtual Memory Partitioning for Enhancing Application Performance in Mobile Platforms

Chaitanya Vooradi

vooradichaitanya@gmail.com

Abstract: Now-a-days software users on mobile platforms are increasing rapidly due to the introduction of application stores. Mobile phones have limited memory storage compared to laptops and server's due to their mobility and they don't have the facility to expand the memory through memory slots. Memory management techniques like Low memory killer (LMK) and out of memory killer (OOMK) were used widely. Fragmentation and thrashing are the other problems because of forced termination of applications.

This paper gives solution to the problems caused due to deterioration of application because of LMK and OOMK. It provides an isolated virtual node at operating system.

Introduction: Most of the devices support both built-in and downloaded applications from application store. Memory management of applications downloaded are tested at manufacturing time. Because of it most of the devices faces memory shortage. LMK is the most widely used memory management technique which terminates the less important applications until sufficient space available for the running application. LMK and OOMK seriously deteriorate user perceived performance in two ways: 1) Victim information was unloaded and during the next usage of this application all the info need to be fetched.

2)Built in application like Phone, SMS and Contacts are forcibly terminated.

At the point when page issues caused by the memory deficiency happen much of the time, the cost of page substitution rules CPU usage, making applications more inclined to miss the required due date. Therefore, rather than really getting free memory, the thrashing often happens. Even the built in applications suffers because of this drawbacks. This paper propose a new technique were memory partitioning at OS level, which limits page reclamation within the memory partitioned area. VNODES are created for built-in application, trusted applications and untrusted applications.

Memory Management Techniques Existing:

Conventional Memory Management, it blindly handles all processes without the platform level semantics, which are important system applications in a mobile platform. LMK is other popular technique where it terminates the application based on the LRU table. OOMK is other technique where it depends memory score of processes heuristically. Thrashing and fragmentation are the main problems all the three techniques are not handled properly.

Proposed Virtual Node Creation technique to improve System Performance: Proposed technique consists of three main components:1) vnode_setup_memblock: manages mapping between physical memory address and virtual node. 2) vnode_generation: partitions the physical memory to specified virtual nodes and determined the table size to hold the address range of physical memory.

3)vnode_set_cpumask: CPU masks are allocated in order to map virtual memory node and specified CPU to recognize CPU-HotPlug and CPU-DVFS. Advantages of using it are VNODE controls unnecessary memory consumption by untrusted applications and reduces the LMK/OOMK operations.

VNODE include dynamic memory partitioning interface at boot time to support various characteristics of mobile phones. Case 1: Generates virtual Nodes for external applications downloaded from the application store and the built-in applications. Case 2: VNODE generates three virtual memory nodes for built-in applications, the applications from the trusted application store, and the untrusted applications. Case 3: Generates Virtual Nodes for responsive-aware applications and external applications in low end mobile devices which have low memory capacity. Case 4: Generated virtual nodes for the responsive-aware built-in applications area provided by the manufacturer, the built-in applications area provided by the telecommunication company, and the large external applications area available due to the sufficient physical memory capacity.

Performance Enhancement Evaluation: Free memory space for application operation increased in the proposed system. Number of executions of LMK and OOMK reduced drastically. Memory fragmentation decreased in a convincing manner. Execution time of the trusted application reduced and for untrusted application it increased.

Conclusion: The proposed memory management technique handles the problems like thrashing, page fault, and page replacement to secure free memory. It minimizes the page reclamation for trusted applications and limits the memory range of untrusted applications. Dynamic memory-controlling interface was used to create different memory layouts for all types of devices. In addition, it supports complete memory isolation of trusted and untrusted applications. Reduces LMK/OOMK operations and increases the performance of trusted applications.

References: http://ieeexplore.ieee.org/document/6689690/