Cache And Memory in Operating Systems

Introduction

In our journey to virtualization, we have seen step by step how the operating systems use the hardware for every running process while it creates a virtual representation of every one of the parts involved. This time we cover an important aspect of virtualization such as job scheduling and the different techniques that can be implemented in the seek of performance, scalability, and efficiency as well as a basic introduction to virtualization of memory.

Cache coherency.

This principle focuses on keeping data consistency in a system with multiprocessor, consistency can be tricked when making changes in one specific processor such as writing new data and later on another processor need to retrieve the original information, when poorly implemented solutions some data access by later processor may be outdated. There are two strategies that can be implemented when solving this problem, the first approach where all the data written in the cache is written as well in memory is known as write-through and is implemented because it easily addresses the issue, the second approach, on the other hand, focuses on optimization and it only copies to memory in the moment of deletion the block that has been changed.

Benefits of using Space Addresses in virtual memory.

Space Address is the virtualization of the memory created by the OS to ensure every application and high-level program have enough space for execution. It focuses mainly on creating
1. Transparency: Implementing virtual memory in a way that the individual program will not notice that the memory is just a virtual representation of the actual memory.
2. Efficiency: Time efficiently and space efficiently, programs should not run slowly and not using unnecessary structures.
3. Protection: The OS need to make sure that the integrity of data is kept, a program can not make changes to data that is being used by another application or program, this way a program has access to the memory allocated to such program.

States of a process.

A process in any given time can be in either of the following three states, running, ready or blocked and to give a clear and brief explanation of every one of these states I will reference the ones listed on the study book.
Running: this state is assigned to a program when it is executing the list of instructions and such task is being performed by a processor, it will change to *"Ready"* whenever a process is ready to be executed but for some reason the OS has decided not to run execute it at the moment, it can be due to the scheduling algorithm or any OS implementation for running processes and the last state that a job may be in is *"Blocked"* that happens when the process has run an operation that prevents the process of being ready, a common example: when a process initiates an I/O request to a disk, it becomes blocked and thus some other process can use the processor. (Arpaci-Dusseau, R. & Arpaci-Dusseau, A., 2012)

Load Balancing.

In modern computing such as cloud computing we find that the algorithm applied for load balancing may vary greatly depending on the service however there are some techniques that are implemented somehow in some of these algorithms.

Round Robin: As reviewed in earlier assignments this technique is a time slicing where a job runs for a given amount of time "slice" which has to be a multiple of the timer-interrupt, and then switches to another job until the jobs are done.
Shortest Time-to-Completion First (STCF): Also known as Preemptive Shortest Job First (PSJF) scheduler which evaluates which job has the least time to finish even after a new job has entered the system and this way it schedules the next job to run. (Arpaci-Dusseau, R. & Arpaci-Dusseau, A., 2012)

Differences between physical, virtual, and logical memory.
Physical memory: It is the RAM of the system that usually takes the form of a card and is part of the hardware that makes up a computer or device.
Virtual Memory: Is the abstraction layer of the physical memory. This virtualization of the memory gives additional control over the space being allocated to a program and the creation of space addresses for the efficiency of jobs and programs.
Logical memory: The simplest explanation I could give is that this is just a logical partition or an address space allocated by the OS as its main storage. For those who have installed an OS in a virtual machine or a Linux distribution will be able to relate with this when assigning the logical partitions in the hard drive for storage and in the RAM for the performance of the OS.

Reference list:
Arpaci-Dusseau, R. & Arpaci-Dusseau, A. (2012). Operating Systems: Three Easy Pieces. Madison, WI: University of Wisconsin-Madison. Available at http://pages.cs.wisc.edu/~remzi/OSTEP/
- Cache Coherence. (n.d.). Retrieved from https://www.d.umn.edu/~gshute/arch/cache-coherence.xhtml