

Chapter 6

Intuitionistic Fuzzy Set Theory with Fair Share CPU Scheduler: A Dynamic Approach

Supriya Raheja
NorthCap University, India

ABSTRACT

CPU Scheduler is one of the crucial modules of any operating system. Operating system uses CPU schedulers based on the requirements. Fair Share CPU scheduler is preferable scheduler by many operating systems. It distributes fixed processor time amongst different parts of a system like groups, users etc. Fixed distribution of processor time can distress the performance of system. Moreover, traditional CPU schedulers ignore the impreciseness and uncertainty of the tasks. This work presents the Intuitionistic Fuzzy based Fair Share scheduler (IFFS). IFFS scheduler has extended the research in the field of the impreciseness of tasks. Intuitionistic Fuzzy Inference Systems, IFIS-DP and IFIS-DCS have been designed to share the CPU time dynamically and for assigning dynamic priority to each task. Experimental results prove the better performance of IFFS scheduler over FS scheduler.

CPU SCHEDULER: MAIN MODULE OF AN OPERATING SYSTEM

Operating System is a significant software component of any computer system. These software components are mainly divided into two categories,

- System software and
- Application software.

Operating system belongs to the system software category. The main task of system software is to run the application software. System software supports the running of programs which uses computer system to run. In modern era, computer system contains a number of different hardware components like:

DOI: 10.4018/978-1-5225-1848-8.ch006

- Processor (CPU),
- Hard disk drives,
- Monitors,
- Different memory,
- Keyboards and
- Other input and output devices.

The other main task of operating system is to manage these resources. The operating system has various modules that work together to carry out the resource management functions and there are interactions between them. For example Task Manager, which creates and destroys the tasks, Memory Manager, which manages the usage of main memory (allocation and release of memory blocks), Device Drivers, which deal with hardware devices, File Manager that manages the file system etc (Galvin & Gagne, 2009). Another main module of an operating system is the CPU Scheduler which handles the scheduling of tasks.

In a multitasking environment, multiple tasks are simultaneously stored in main memory. Operating system requires CPU schedulers to provide the concurrent execution of tasks. When multiple tasks are stored in main memory, the CPU scheduler makes an illusion that each task has its own CPU. It is also known as Short Term scheduler. However, the CPU will be executing only one task at a time, but it switches from one task to another task to support concurrent execution (Dhamdhare, 2006; Stallings, 2014). The CPU scheduler at particular time selects one ready task from the main memory (Ready Queue) and allocates CPU to that task. The CPU scheduler decides whom to execute next and for how long. The performance of the system varies in accordance to the chosen criteria i.e. on scheduling technique since it changes the ready state of task to running state. When the task is in ready queue then it is in ready state and when CPU is assigned to it, it is in running or active state. An effective scheduling can enhance the performance; on the other hand an ineffective scheduling can lead to reduce the performance of the entire system. Like, with non-preemptive scheduling algorithm, a system can suffer with the convoy effect. When many tasks are in waiting state while one big task is executing it is called convoy effect. It degrades the performance as the one big task halts the execution of all other small tasks.

Different operating systems use different schedulers based on their requirements like the operating systems which handle the tasks based on their criticality prefer to use priority based schedulers. The CPU scheduler usually executes the highest-priority tasks first. For example, whenever a currently running task is interrupted by an event then the CPU scheduler must make some optimal adjustments and schedule the next highest-priority task to processor. Windows & Solaris based operating systems prefer to use multilevel feedback queue scheduler whereas UNIX based operating systems usually use fair share scheduler. This chapter discusses Fair Share (FS) CPU scheduler.

FAIR SHARE SCHEDULER AND ITS STATIC APPROACH

FS CPU scheduler equally/statically distributes the processor time between groups, users and tasks, as opposed to the fixed distribution among tasks (Round Robin Schedulers). Fairness should be the major consideration for any scheduler. Traditional schedulers considered that every task belongs to a different user, therefore schedulers like Round Robin logically distribute equal CPU services/processor time to all tasks (Sabin et al., 1996; Nie et al., 2011). When several tasks need a processor, operating system often allots processor time slices on a round robin fashion i.e. scheduler distributes equal time to all tasks

26 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/intuitionistic-fuzzy-set-theory-with-fair-share-cpu-scheduler/174733

Related Content

Bio-Inspired Computing through Artificial Neural Network

Nilamadhab Dash, Rojalina Priyadarshini, Brojo Kishore Mishra and Rachita Misra (2017). *Fuzzy Systems: Concepts, Methodologies, Tools, and Applications* (pp. 1285-1313).

www.irma-international.org/chapter/bio-inspired-computing-through-artificial-neural-network/178441

Embedded System Verification Using Formal Model an Approach Based on the Combined Use of UML and Maude Language

Meliouh Ameland Chaoui Allaoua (2018). *International Journal of Conceptual Structures and Smart Applications* (pp. 42-58).

www.irma-international.org/article/embedded-system-verification-using-formal-model-an-approach-based-on-the-combined-use-of-uml-and-maude-language/233534

Intelligent Information Integration: Reclaiming the Intelligence

Naveen Ashish and David A. Maluf (2009). *International Journal of Intelligent Information Technologies* (pp. 28-54).

www.irma-international.org/article/intelligent-information-integration/4038

The Undervaluation of Family Businesses: The Explanatory Power of Governance Mechanisms

Narjess Skhiri Ep Hellara (2024). *Social and Ethical Implications of AI in Finance for Sustainability* (pp. 247-267).

www.irma-international.org/chapter/the-undervaluation-of-family-businesses/345122

Review on Healthcare Quality Using Machine Learning Methods

Pokkuluri Kiran Sree and N. Usha Devi (2023). *AI and IoT-Based Technologies for Precision Medicine* (pp. 413-430).

www.irma-international.org/chapter/review-on-healthcare-quality-using-machine-learning-methods/332847