CSE 506 Operating Systems
Paper 5

Your Name: Sarthak Parakh
Paper Number: 5
Paper Title: The Battle of the Schedulers: FreeBSD ULE vs. Linux CFS
Paper Authors: Justinien Bouron, Sebastien Chevalley, Baptiste Lepers, Willy Zwaenepoel, Redha Gouicem, Julia Lawall, Gilles Muller, and Julien Sopena

1. **What problem does the paper address? How does it relate to and improve upon previous work in its domain? (one paragraph, <= 7 sentences)**

The paper analyzes two widely used schedulers - **FreeBSD's ULE and Linux's CFS** and performs an **apple-to-apple comparison** between them. The problem addressed revolves around understanding the **impact of scheduler design choices and implementation on system performance**, mainly in the context of **multicore machines** under various **workloads**. As a solution, by porting one scheduler into the other's kernel, they create a controlled environment to isolate the scheduler's influence, mitigating biases associated with different operating systems. The paper also discusses how design discrepancies in handling **interactive tasks, preemption, load balancing, and thread placement** affect overall system performance. It presents a unique perspective on the ongoing discourse on scheduler design and its ramifications for real-world workloads.

2. **What are the key contributions of the paper? (one paragraph <= 7 sentences)**

The paper performs detailed and apples-to-apples evaluation of two widely used schedulers, FreeBSD ULE, and Linux CFS. By transplanting one scheduler into the kernel of the other, the study shows the impact of these design disparities on system performance through a variety of benchmarks and workloads, providing a nuanced understanding of how scheduling decisions influence the execution of tasks on multicore machines. The paper discusses informed choice and reasons to **transplant which scheduler** (ULE into Linux). Further, it discusses challenges during implementation such as **interface mismatch** (Linux provides APIs whereas FreeBSD doesn't), but they can easily map their functions, and **different low-level assumptions** which include locking mechanism, thread transfer, runqueue management and priority. Overall, they slightly tweak ULE's code to bind with Linux's assumptions. Lastly, the study evaluates both schedulers on single-core and multicore scenarios, considering various workloads, including single and multi-application scenarios.

3. **Briefly describe how the paper's experimental methodology supports the paper's conclusions. (one paragraph <= 7 sentences)**

The experiments include a mix of **synthetic benchmarks and real-world applications**, executed on a 32-core machine, ensuring a comprehensive evaluation across different hardware configurations. The paper evaluates performance under various workloads, shedding light on the impact of design choices on scheduling behavior. Workloads include synthetic benchmarks, realistic applications, and specific use cases like database benchmarks, HPC applications, and parallel applications. The methodology provides concrete evidence to substantiate the observed differences in **interactive task handling, preemption policies, load balancing efficiency, and thread placement**. It focuses on **per-core scheduling** to analyze the impact of design decisions on **fairness and starvation.** Through figures illustrating cumulative runtimes, interactivity penalties, and performance comparisons, the paper effectively demonstrates the differences between CFS and ULE in scenarios involving both multi-application workloads and single-application workloads. Overall, the well-designed and diverse experimental methodology contributes robust evidence to support the paper's findings on the performance disparities between CFS and ULE.