

**Summary**

The paper presents a new OS architecture called “Multikernel”, which tries to provide better scalability and adaptability across multi-core heterogeneous hardware systems unlike monolithic OS.

**Main Points**

The paper claims that hardware is growing at a faster pace than software due to which many optimizations are done at hardware and software level to increase the performance of systems, is difficult to scale with increasing number of cores and portability becomes an issue as well. As the new diverse and improved hardware is created we have to repeat these optimizations all over again for different kind of hardware. To decouple this dependency between hardware and software this paper presents a distributed system like approach for OS.

*Multikernel* is designed to treat system as a network of interconnected cores that communicate through message passing and does not share any memory across different cores.

The multikernel model is built on three design principles:

- Make inter-core communication explicit : Instead of sharing memory among cores all communications are done via message passing between multiple cores. It will provide better isolation, resource management on heterogeneous cores and provides modularity by using well defined interfaces.
- Make OS structure hardware-neutral: This will make OS scalable and future proof. It can be achieved by making hardware interfaces and messaging transport mechanism, hardware independent. This will remove the need to do optimizations for specific hardware.
- View State as replicated: No memory is shared among cores and each core maintains their own replica of state. Consistency is maintained between these states. Since replicas are closer to the cores results in better performance and improves scalability by reducing interconnect loads, memory contention and synchronization overheads.

Authors present a prototype called “Barrelfish” based on *Multikernel* OS design and its evaluation against conventional OS designs. *Barrelfish* although performing bad for lower core counts, it scales well and performs better as number of cores increases in variety of tests.

**Limitations/Critique**

The Drawbacks of Multikernel can be degrading performance of OS in homogeneous system with low core counts. Also not using Cache-coherence among different core can result in bad performance as well since current programs are optimized to take advantage of that.

**Proposed Extension**

Different message queuing optimizations can be done to improve the message sharing between cores. Cache coherence can be implemented if heterogeneous cores are involved. Different File system can be designed to take full advantage of distributed core system.