Review of

Exokernel

An Operating System Architecture for Application-Level Resource Management

Dawson R. Engler, M. Frans Kaashoek, and James O'Toole Jr.

What is the problem?

Abstraction of machine resources behind a veil of system processes inherently affects the performance of the applications that run on the system. This limits innovation of new abstractions and restricts hardware optimizations and in general hampers the flexibility of the developer.

Summary

The exokernel uses the end-to-end argument and extends it to low-level operating system software. It aims to implement operating system abstractions efficiently in the form of libraries. By doing so, they observe that they can significantly improve performance and reduce the overheads that are associated with traditional systems.

Key Insights

- Traditional systems do pose a performance overhead simply because they do not support specialized implementations. Using the end-to-end argument, the exokernel operating system architecture addresses this problem by providing application-level management of physical resources.
- They identified the key issue of security for mutually distrusting applications. They came up with a good framework to provide secure resource multiplexing called Secure Bindings.

Strengths

- As intended, the exokernel does give more freedom to the developer of an application to decide how to best manage their resources. This also decreases the number of kernel crossings as more functions are carried out in the application space.
- The concept of "Secure Binding" solves the most fundamental problem with the exokernel approach. The issue of safeguarding against malicious applications that aim to hijack the hardware resources for some purposes. They propose three good different techniques to ensure this secure binding works for most malicious software.

Weaknesses

- Though the paper states that Aegis and ExOS don't offer the same level of functionality as Ultrix, it still compares their performances head to head. The paper does state that they do not expect the additions to cause large increases in their timing measurements. I feel the need for more justifications to that end.
- I am not sure if the end-to-end argument and implementation translate to a general, personal use PC. Do we really need to put a lot of emphasis on developers of simple tools to also have the technical ability to handle all the low-level interactions.

Summary of Key Results

- The exokernel exports hardware resources rather than emulating them in virtual machines thereby allowing for a simple and efficient implementation.
- The abstraction led to the most primitive kernel operations being 100 times faster than an implementation of the UNIX operating system. In general, the minimalistic exokernel architecture systems performed better than traditional ones in almost all benchmarks.

Open Questions

- Are secure binding and ASHs secure enough?
- What are possible attacks? How to defend against them? (Perhaps a security application that always runs on the exokernel.)