

DINH DUY KHA

2019712308

The Scalable Commutativity Rule: Designing Scalable Software for Multicore Processors

Summary

This paper proposes a rule of interface designing to design scalable software by considering commutativity: **whenever interface operations commute, they can be implemented in a way that scales**. This rule is proven true by using a formalism based on abstract actions, histories and implementations. A new tool called COMMUTER is also proposed to help applying the rule by analyzing high-level interface models and generates tests of operations. Base on the proposed principles, the authors implemented an operating system named sv6 that scale better than Linux according to the tests.

Strengths

The authors take a look at the fundamental factors which makes an implementation scale by analyzing actions, histories and implementations and come up with a general rule. This approach helps developers design scalable software interfaces even before implementation.

An automated process to apply the proposed rule is also introduced, which is then used to analyze system calls of Linux to discover many problems that limit scalability.

This paper shows that commutativity is important for achieving scalability and performance.

Weaknesses

The presentation of the rule is very general and developers has to find scalable implementation for themselves.