Silas Boyd-Wickizer, Austin T. Clements, Yandong Mao, Aleksey Pesterev, M. Frans Kaashoek, Robert Morris, and Nickolai Zeldovich

An Analysis of Linux Scalability to Many Cores In Proceedings of the 9th Symposium on Operating Systems Design and Implementation (OSDI), Vancouver, Canada, October 2010

1. What is the motivation for this work? Why look at scalability of OS kernel?
2. What is their approach for finding scalability problems?
3. Why might scalability be limited?
4. On how many cores is Exim no longer scalable? What is the bottleneck in Exim?
5. Why can even short critical sections lead to scalability problems? What is the problem with ticket locks? How do they fix the problem of the expensive ticket lock protecting hash\_get()?
6. How does exsim perform after this first problem is fixed? What is the next problem?
7. What are slopping counters and why do they help scalability?
8. Which applications needed no improvements from OS for improved scalability?
9. Did they adequately remove bottlenecks from OS?

Austin T. Clements, M. Frans Kaashoek, Nickolai Zeldovich, Robert T. Morris, and Eddie Kohler

The Scalable Commutativity Rule: Designing Scalable Software for Multicore Processors.

In Proceedings of the 24th ACM Symposium on Operating Systems Principles (SOSP), Farmington, PA, November 2013.

1. How do previous approaches determine if a system is scalable? What is the problem with that general approach?
2. Why does it matter if an interface is scalable or not? What does it mean for an interface to be commutative? Does an interface need to commutative for an implementation to be scalable?
3. What operations scale on today’s multi-core systems?
4. Commutativity of operations is sensitive to what aspects? When is creat() commutative and when is it not?
5. What are some guidelines for designing commutative interfaces?
6. What does their tool Commuter do? What does it find? What test cases does it generate? How does it determine that interface will not be scalable for some situations?
7. How scalable is Linux 3.8?
8. What principles do they use to make a more scalable OS?
9. Does their approach increase scalability for multithreaded applications?