# 阅读报告4

1. 论文题目：

The Scalable Commutativity Rule:Designing Scalable Software for Multicore Processors

1. 阅读报告

接口设计对可伸缩性的实现有很大影响，作者定义可伸缩性交换规则，并编写了一个称为COMMUTER的工具, 以及一个原型操作系统SV6. 来共同指导接口的设计，通过这些技术，重新设计了文件系统调用相关的POSIX调用，并大大提高可伸缩性。

论文核心是可伸缩性的交换规则，如果他们的实现具有无冲突内存访问，则声明一组操作规则，并且在通信可以帮助避免无序操作之间的内存访问冲突。在共享内存系统中，如果一个内核没有写入由另一个内核读取/写入的高速缓存行，则操作系统会通信。这种无冲突情况下，接口操作1以可扩展的方式进行通信。

COMMUTER工具采用界面模型，并确定操作组通行的确切条件。该工具可以集成到开发过程中，以推动初始设计和实现，逐步改进现有实现，或帮助开发人员理解接口的交换性。用户使用Python表示接口的符号模型，并且COMMUTER的ANALYZER组件根据参数和状态生成表达式，这些表达式准确地指示操作集的通信时间。这些表达式可以直接检查，也可以传递给TESTGEN，TESTGEN可以将这些条件转换为真正的测试用例。

实验表明对COMMUTER产生的13664个test而言, linux能对其中的68%良好扩展, 而SV6能良好扩展其中的99%。

论文最后，作者进行了详细的评估，在COMMUTER中对几个POSIX文件系统和虚拟内训调用进行建模，并评估Linux的可伸缩性，COMMUTER确定sv6在13664次测试中的13528次是无冲突的，而Linux在9389次测试中没有冲突。