## Suli Yang, Tyler Harter, Nishant Agrawal, Salini Selvaraj Kowsalya, Anand Krishnamurthy, Samer Al-Kiswany, Andrea C. Arpaci-Dusseau, Remzi H. Arpaci-Dusseau

## Split-Level I/O Scheduling

## Proceedings of the 25th ACM Symposium on Operating Systems Principles (SOSP '15)

1. What is the motivation for this work?
2. What types of I/O scheduling policies could be needed?
3. What are the three main limitations of existing I/O scheduling frameworks?
4. Cause Mapping:
   1. Why is this difficult to achieve in current file systems?
   2. What motivating experiment shows this problem?
   3. How does the Split-IO framework fix this problem?
   4. How do the resulting experimental results change?
5. Reordering:
   1. Why is this difficult to achieve in current file systems?
   2. What motivating experiment (or two or three) shows this problem?
   3. How does the Split-IO framework fix this problem?
   4. How does the resulting experiment change?
6. Cost Estimation
   1. Why is this difficult to achieve in current file systems?
   2. What motivating experiment shows this problem?
   3. How does the Split-IO framework fix this problem?
   4. How do the resulting experimental results change?
7. Which of these problems does a traditional block-level I/O scheduler have?   
   Which problems does a system-call level I/O scheduler have?
8. Which issues could change with SSDs instead of HDDs?
9. In what ways are I/O scheduling and CPU scheduling the same vs. different?   
   Does CPU scheduling have these 3 issues?