
CASSYOPIA

Compiler Assisted Systems Optimization

Mohan Rajagopalan Saumya Debray
Matti Hiltunen Rick Schlichting



Background

- Orthogonal concerns in optimization
- What's missing
- Symbiotic optimizations
 - Optimize whole system: OS + Application
 - Use compiler optimization techniques along with OS design to minimize overheads
 - Share information between OS and executable to allow fine grained control and tuning
- Design concerns



Example: System Call Optimization

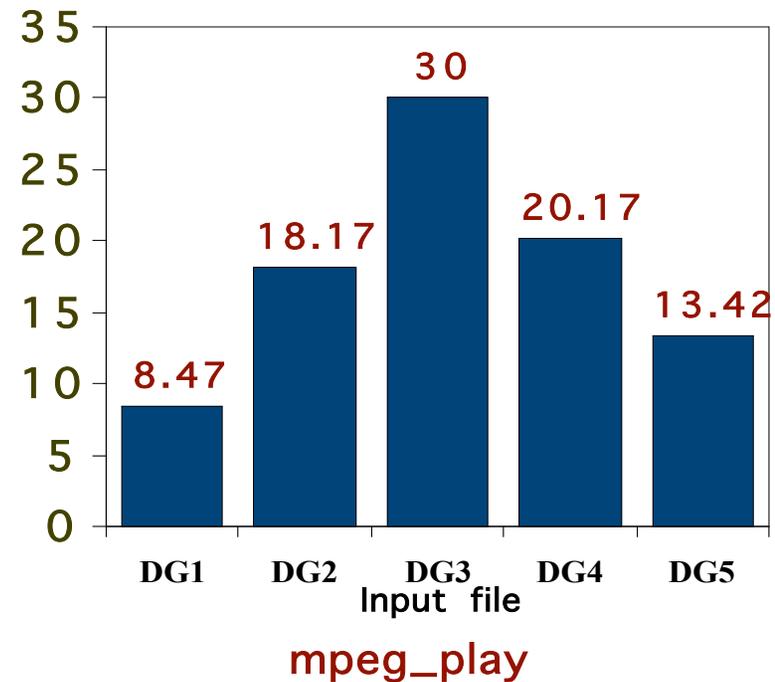
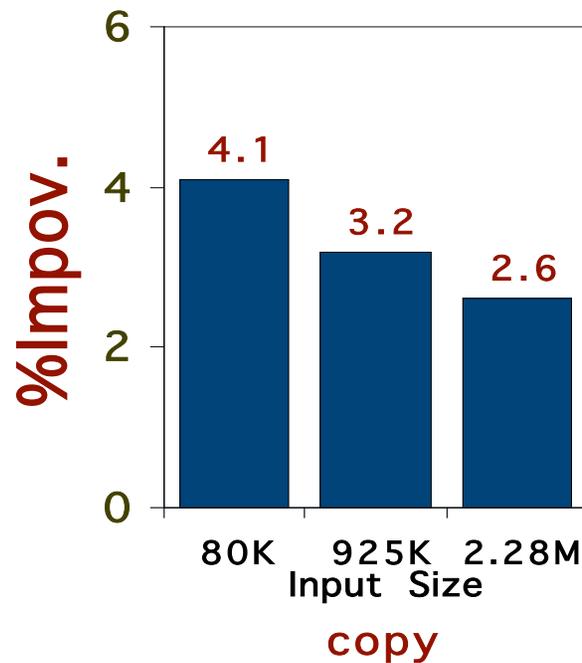
Optimize a program's system call behavior

- Profile system call sequences
- Cluster system calls
 - Compiler optimization techniques, e.g., code motion, loop unrolling
- Kernel support: *multi-call*
 - Execute multiple calls per boundary crossing
- Implementation
 - Optimization pass using PLTO binary rewriting toolkit



Preliminary Results

System Call Optimization



- Power consumption
- Cache misses



Looking Ahead

- Boundary crossing costs
 - Distributed systems
- Fine-grain adaptation control
 - Power management
 - Context-based adaptive scheduling
- Generic optimizations
 - Permission checking
 - Redundant code elimination



CASSYOPIA

HotOS IX

