Collecting Whole-System Reference Traces of Multiprogrammed and Multithreaded Workloads

Scott F. Kaplan
Dept. Mathematics and Computer Science
Amherst College

Presented by Christine Hung

Motivation

- ***Completeness***
  - Thoroughness, applicability, & concurrency
- Distortion
  - Discontinuities, dilations, & self-tracing
- Detail
- ***Portability & Maintainability***
- Efficiency

Trace Collectors

- Existing techniques
  - Modify micro-code, simulate instruction set, annotate code
- Existing collectors
  - ATUM, Shade, ATOM & QPT & Etch
- Not comprehensive
  - Memory references, kernel level threads and events, processes, system calls, application concurrency, shared memory and cache

Laplace

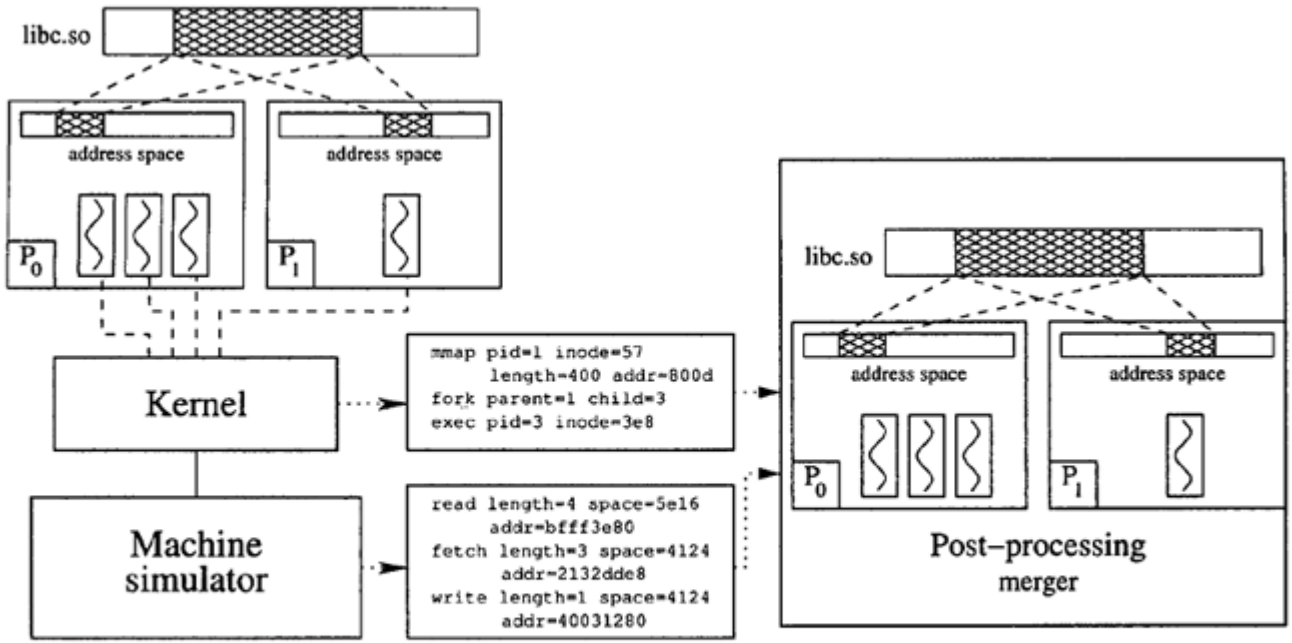

Figure 1: The modified machine simulator and modified kernel each emit logs that are consumed by the post-processor, which interleaves the records and reconstructs the relationships between processes, threads, and address spaces, as well as mappings of shared spaces in each virtual address space.

Laplace

- Modified Processor Simulator
  - Instruction type, virtual address, kernel/user, & timestamp
- Modified Kernel
  - Process/threads, memory mapping, file system cache, & system calls
- Kernel to simulator interface
  - Kernel stores values and signals simulator
  - Simulator stores kernel trace records and ignores these storage references

Laplace

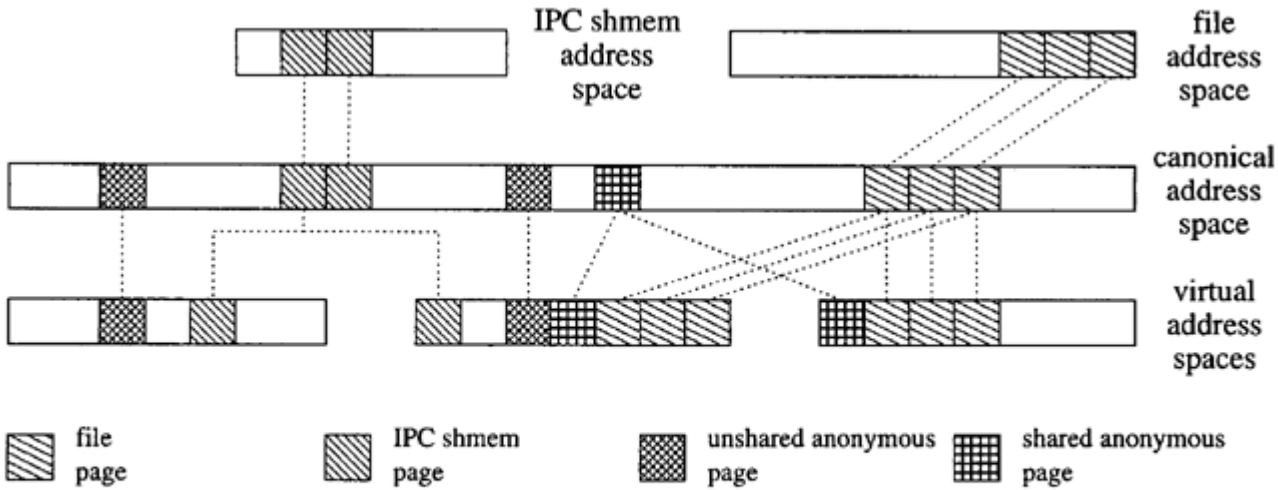

Figure 2: Each canonical page is associated either with a file page, an IPC shared memory page, or an anonymous page (which can be shared or unshared). Every virtual page must be mapped to some canonical page, and the post-processor uses that canonical page to determine the type of the page and its shared status.

- Post Processor
  - Canonical pages
  - Reconstructs system states (processes, threads, tasks, pages, buffer cache, & kernel space)
  - Output in different/desired trace formats (to be extended by user)

Implementation

- Performance
  - Execution overhead
  - ***Reference handling***
    - Production rate
    - Streaming to disk
- Compression and reduction
- Online simulation

Application	Bochs slowdown	Laplace/Bochs no output slowdown	Laplace/Bochs into /dev/null slowdown
equake	195x	275x	377x
gcc	85x	136x	157x
gzip	402x	444x	491x
Mean	227x	285x	342x

- 1GHz Pentium III
- Modified Linux 2.2.21
  - 317 lines of code, 20 files
- Modified Bochs (processor simulation)
  - 873 lines of code
- SPEC2000 benchmarks
- Trace workload at 4.4 MIPS
- Laplace runs at 5.5 MRPS
- Streaming to disk at 93.5 MB/s

Conclusion & Future Work

- Completeness of trace collected
- Easy to modify and port
- Biggest problems
  - Reference handling
  - Storage space
- Possible solutions?
  - Compression, “Lossless reduction”
  - Filtering