The Impact of Operating System Structure on Memory System Performance

J. Bradley Chen, Brian N. Bershad

2017710637 류 정 민
Contents

- Abstract
- Introduction
- Trace overview
- Comparative system behavior
- Seven assertions
- Conclusion
- Q&A
Abstract

- Monolithic System VS Microkernel-based System
- Monolithic: DEC’s Ultrix
- Microkernel-based: Mach 3.0 with CMU’s UNIX server

<table>
<thead>
<tr>
<th></th>
<th>Ultrix</th>
<th>Mach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle system instruction</td>
<td>high</td>
<td>Low</td>
</tr>
<tr>
<td>MCPI</td>
<td>low</td>
<td>high</td>
</tr>
</tbody>
</table>
Introduction

- Monolithic System VS Microkernel-based System

- Monolithic: DEC’s Ultrix
- Microkernel-based: Mach 3.0 with CMU’s UNIX server

- Common Feature
  1) Derived from 4.2 BSD UNIX
  2) Share identical application programming interface
  3) Large amounts of code
# Introduction

- Seven Popular Assertions

<table>
<thead>
<tr>
<th>Assertion</th>
<th>Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The operating system has less instruction and data locality than user programs [14, 15].</td>
<td>The operating system isn’t getting faster as fast as user programs.</td>
</tr>
<tr>
<td>2. System execution is more dependent on instruction cache behavior than is user execution [35].</td>
<td>A balanced cache system for user programs may not be balanced for the system.</td>
</tr>
<tr>
<td>3. Collisions between user and system references lead to significant performance degradation in the memory system (cache and TLB) [30, 35, 36].</td>
<td>A split user/system cache could improve performance.</td>
</tr>
<tr>
<td>4. Self-interference is a problem in system instruction reference streams [28, 35].</td>
<td>Increased cache associativity and/or the use of text placement tools could improve performance.</td>
</tr>
<tr>
<td>5. System block memory operations are responsible for a large percentage of memory system reference costs [31, 35].</td>
<td>Programs that incur many block memory operations will run more slowly than expected.</td>
</tr>
<tr>
<td>6. Write buffers are less effective for system (as opposed to user) reference streams [5, 18].</td>
<td>A write buffer adequate for user code may not be adequate for system code.</td>
</tr>
<tr>
<td>7. Virtual page mapping strategies can have significant impact on cache performance [25, 29].</td>
<td>Systems should support a flexible page mapping interface, and should avoid default strategies that are prone to pathological behavior.</td>
</tr>
</tbody>
</table>
Introduction

Section 2. Trace methodology & broad summary

Section 3. Compare system behavior & performance between Mach and Ultrix

Section 4. Evaluate implementation of monolithic and microkernel according to the assertions

Section 5. Summarize
Trace overview - simulation

- DS5000/200 memory system
  - Fairly conventional
  - Mach3.0 and Ultrix both run on this system

- Memory system parameters

  - Instruction cache: 64 KB, direct-mapped, physical, 8 byte line, 15 cycle miss penalty.
  - Data cache: 512 KB, direct-mapped, physical, 8 byte line, write allocate, 15 cycle read miss penalty, read miss fetches 16 aligned bytes.
  - Write buffer: 64 entries, page-mode
    - Write complete in 1 cycle, non page-mode writes complete in 5 cycles; CPU reads have priority for memory access, but wait for writes that have already started. 4 KB page size for page-mode writes.
  - Translation buffer: 54 entries, 56 random/8 wired entries, trap to software on TLB miss. Each TLB entry maps a 4 KB page.
  - Page mapping: Deterministic. The physical page used to back a given virtual page is determined by the virtual page number and its address space identifier. The deterministic strategy prevents conflicts within any 64 KB (cache size) range of virtual addresses.
  - Kernel memory: All kernel text and most kernel data is in unmapped physical memory.
Trace overview – virtual memory

- Ulitrix
  - Machine dependent
  - Derived from the original BSD abstractions

- Mach
  - Flexible, and aggressive
  - Machine dependent layer and machine independent layer are separated
Trace overview – distortion source

- Memory dilation
  - Paging & TLB miss rate increase
- Time dilation
  - Activity dependent on external events are caused
- Virtual memory system’s page mapping strategy
  - Due to simulator’s model of the virtual memory system’s page mapping strategy
Trace overview – MCPI

- MCPI: Memory Cycles Per Instructions
- # of CPU stall cycles
- Function of the ratio of memory speed to processor speed
- Affected by overall CPI more than processor architecture
- High system MCPI: sed, egrep, yacc, gcc, compress
- Small system MCPI: fppp, liv, doduc, tomcatv
Comparative system behavior

- Mach vs. Ultrix
  - The number and cost of non-idle instruction to run application is different

- Eleven system overheads
  - Trap: syscall and exception handling
  - UTLB: user TLB miss
  - KTLB: kernel TLB miss
  - VM-md: machine-dependent virtual memory
  - VM-mi: Mach’s machine independent virtual memory
  - Block Ops: block memory moves and zeroes
  - UNIX service: the remaining routines in the Ultrix kernel and Mach UNIX server
  - Microkernel: Mach’s microkernel, including device management & scheduling
  - IPC: the Mach kernel's message system
  - Emulator: Mach’s transparent emulation library
  - S-MCPI: system memory cycles per system instruction
Comparative system behavior

Figure 3-1: Relative system overheads for programs running on Ultrix and Mach.
Comparative system behavior

- Memory penalty for system instructions
  - Mach > Ultrix

- The number of instructions executed in virtual memory system
  - Mach > Ultrix

- Relative instruction cost
  - Mach < Ultrix

- Overhead of Mach’s IPC is responsible for a small portion of overall system overhead
Seven assertions
system & user locality

• **Assertion:** *The operating system has less instruction and data locality than user programs*

• **Conclusion:** *The percentage of misses due to the system is larger*
  • Mach occurs more cache misses than Ultrix
  • Most of additional cache misses are due to the system
Seven assertions
system instruction locality

• **Assertion:** System execution is more dependent on instruction cache behavior than is user execution

• **Conclusion:** MCPI of both instruction cache miss and data cache miss are larger in system than user except gcc
  • System MCPI: Ultrix < Mach
  • User MCPI: Ultrix ≈ Mach
Seven assertions

competition between the user and system

• **Assertion:** Collisions between user and system references lead to significant performance degradation in the memory system

• **Conclusion:** System performance is not affected by user and system competition
  • It can be measured by cache and TLB miss rate

<table>
<thead>
<tr>
<th>workload</th>
<th>inst</th>
<th>data</th>
<th>total</th>
<th>inst</th>
<th>data</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>sed</td>
<td>0.010</td>
<td>0.006</td>
<td>0.004</td>
<td>0.009</td>
<td>0.004</td>
<td>0.013</td>
</tr>
<tr>
<td>egrep</td>
<td>0.003</td>
<td>0.000</td>
<td>0.003</td>
<td>0.002</td>
<td>0.002</td>
<td>0.004</td>
</tr>
<tr>
<td>yacc</td>
<td>0.005</td>
<td>0.002</td>
<td>0.007</td>
<td>0.004</td>
<td>0.005</td>
<td>0.009</td>
</tr>
<tr>
<td>gcc</td>
<td>0.050</td>
<td>0.077</td>
<td>0.057</td>
<td>0.047</td>
<td>0.018</td>
<td>0.065</td>
</tr>
<tr>
<td>compress</td>
<td>0.004</td>
<td>0.018</td>
<td>0.022</td>
<td>0.010</td>
<td>0.034</td>
<td>0.044</td>
</tr>
<tr>
<td>ab</td>
<td>0.038</td>
<td>0.006</td>
<td>0.044</td>
<td>0.029</td>
<td>0.000</td>
<td>0.029</td>
</tr>
<tr>
<td>espresso</td>
<td>0.005</td>
<td>0.002</td>
<td>0.007</td>
<td>0.004</td>
<td>0.004</td>
<td>0.008</td>
</tr>
<tr>
<td>lisp</td>
<td>0.002</td>
<td>0.006</td>
<td>0.008</td>
<td>0.022</td>
<td>0.004</td>
<td>0.026</td>
</tr>
<tr>
<td>equ</td>
<td>0.000</td>
<td>0.004</td>
<td>0.005</td>
<td>0.000</td>
<td>0.005</td>
<td>0.005</td>
</tr>
<tr>
<td>fppp</td>
<td>0.072</td>
<td>0.002</td>
<td>0.074</td>
<td>0.047</td>
<td>0.002</td>
<td>0.049</td>
</tr>
<tr>
<td>deduce</td>
<td>0.023</td>
<td>0.002</td>
<td>0.025</td>
<td>0.016</td>
<td>0.002</td>
<td>0.018</td>
</tr>
<tr>
<td>liv</td>
<td>0.001</td>
<td>0.004</td>
<td>0.005</td>
<td>0.000</td>
<td>0.001</td>
<td>0.002</td>
</tr>
<tr>
<td>tomcatv</td>
<td>0.000</td>
<td>0.005</td>
<td>0.006</td>
<td>0.000</td>
<td>0.005</td>
<td>0.006</td>
</tr>
</tbody>
</table>

**Table 4-2:** MCPI contributions from cache competition.
Seven assertions
system self-interference

- **Assertion:** Self-interference is a problem in system instruction reference streams

- **Conclusion:** As the associativity increases, the fraction of misses decreased, which is effective to instruction than data reference.
  - High MCPI -> cache is full
    - Difficult to be improved by the increment of associativity
  - The associativity is more helpful to Ultrix than Mach
    - Mach’s MCPI is higher than Ultrix

![Figure 4-2: System self-interference.](image)
Seven assertions
block operation

• **Assertion:** System block memory operations are responsible for a large percentage of memory system reference costs

• **Conclusion:** Block memory operations can be responsible for a substantial fraction of total MCPI

  • Block operations incur a larger overhead for programs running on Mach than on Ultrix

<table>
<thead>
<tr>
<th>Block operation type</th>
<th>In kernel</th>
<th>In UNIX Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mach</td>
<td>Part of VM and IPC system</td>
<td>Part of file system</td>
</tr>
<tr>
<td>Ultrix</td>
<td>VM and file system</td>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>workload</th>
<th>Ultrix</th>
<th>Mach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MCPI</td>
<td>%total</td>
</tr>
<tr>
<td>sed</td>
<td>0.066</td>
<td>29.2</td>
</tr>
<tr>
<td>egrep</td>
<td>0.014</td>
<td>39.3</td>
</tr>
<tr>
<td>yacc</td>
<td>0.017</td>
<td>25.6</td>
</tr>
<tr>
<td>gcc</td>
<td>0.116</td>
<td>26.8</td>
</tr>
<tr>
<td>compress</td>
<td>0.055</td>
<td>22.1</td>
</tr>
<tr>
<td>ab</td>
<td>0.100</td>
<td>23.4</td>
</tr>
<tr>
<td>espresso</td>
<td>0.009</td>
<td>21.3</td>
</tr>
<tr>
<td>lisp</td>
<td>0.000</td>
<td>0.3</td>
</tr>
<tr>
<td>eqntott</td>
<td>0.000</td>
<td>0.4</td>
</tr>
<tr>
<td>fppp</td>
<td>0.003</td>
<td>1.2</td>
</tr>
<tr>
<td>doduc</td>
<td>0.003</td>
<td>0.9</td>
</tr>
<tr>
<td>liv</td>
<td>0.008</td>
<td>7.1</td>
</tr>
<tr>
<td>tomcatv</td>
<td>0.000</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Table 4-4:** MCPI from block memory operations.
Seven assertions
streaming writes

• **Assertion:** Write buffers are less effective for system references

• **Conclusion:** System behavior is worse than user behavior in case of streaming write operations
  • System write buffer stalls per instruction are generally higher for Mach than for Ultrix

<table>
<thead>
<tr>
<th>workload</th>
<th>Ultrix system</th>
<th>Ultrix user</th>
<th>Mach system</th>
<th>Mach user</th>
</tr>
</thead>
<tbody>
<tr>
<td>sed</td>
<td>0.061</td>
<td>0.000</td>
<td>0.076</td>
<td>0.000</td>
</tr>
<tr>
<td>egrep</td>
<td>0.050</td>
<td>0.002</td>
<td>0.065</td>
<td>0.002</td>
</tr>
<tr>
<td>yacc</td>
<td>0.006</td>
<td>0.000</td>
<td>0.076</td>
<td>0.000</td>
</tr>
<tr>
<td>gcc</td>
<td>0.010</td>
<td>0.012</td>
<td>0.129</td>
<td>0.012</td>
</tr>
<tr>
<td>compress</td>
<td>0.043</td>
<td>0.011</td>
<td>0.063</td>
<td>0.013</td>
</tr>
<tr>
<td>ab</td>
<td>0.040</td>
<td>0.009</td>
<td>0.043</td>
<td>0.010</td>
</tr>
<tr>
<td>espresso</td>
<td>0.093</td>
<td>0.001</td>
<td>0.111</td>
<td>0.001</td>
</tr>
<tr>
<td>lisp</td>
<td>0.007</td>
<td>0.004</td>
<td>0.064</td>
<td>0.005</td>
</tr>
<tr>
<td>eqn</td>
<td>0.014</td>
<td>0.000</td>
<td>0.024</td>
<td>0.000</td>
</tr>
<tr>
<td>fppp</td>
<td>0.036</td>
<td>0.017</td>
<td>0.037</td>
<td>0.015</td>
</tr>
<tr>
<td>doduc</td>
<td>0.101</td>
<td>0.018</td>
<td>0.095</td>
<td>0.018</td>
</tr>
<tr>
<td>liv</td>
<td>0.052</td>
<td>0.090</td>
<td>0.075</td>
<td>0.090</td>
</tr>
<tr>
<td>tomcatv</td>
<td>0.023</td>
<td>0.033</td>
<td>0.044</td>
<td>0.033</td>
</tr>
</tbody>
</table>

**Table 4-6:** Write buffer stall cycles per instruction.
Seven assertions
page mapping strategy

- **Assertion:** Larger page sizes have significantly different performance.

- **Conclusion:** Larger page sizes are not as ineffective as they seem.
  - `egrep`, `ls`, `gzip`: performance is worse with smaller page sizes.
  - `espresso`, `lisp`, `egrep`: performance improves with larger page sizes.
  - `fftp`, `ddscale`, `libv`: performance is generally better with larger page sizes.
  - `tar`: performance is similar across different page sizes.

### Performance Breakdown

- **System l-cache hits**: Performance is better with larger page sizes.
- **System l-cache misses**: Performance is worse with larger page sizes.
- **System u-cache hits**: Performance improves with larger page sizes.
- **System u-cache misses**: Performance is worse with larger page sizes.
- **System u-cache stalls**: Performance is similar across different page sizes.
- **MCPT**: Performance is generally better with larger page sizes.
Conclusion

- True: 6
  - System locality is worse than user locality
  - System text locality is worse than system data locality
  - In case of system text, system misses occurs self-interference
  - Block operations can be responsible for a large component of overall MCPI
  - System code presents a higher load to the write buffer than user code
  - Page mapping strategies can have a large effect on cache performance

- False: 1
  - System performance is affected by user and system competition (False)

*Memory system latency is more significant factor to the performance of the operating system (monolithic, microkernel-based) than applications*
Q&A