Software Prefetching and Caching for TLBs

Kavita Bala
M. Frans Kaashoek
William E. Weihl

MIT Laboratory for Computer Science

November 17, 1994

Overview

- Problem
- Prefetching TLB entries
- Caching TLB entries
- Prefetch+Cache for TLBs
- Conclusions

Translation Lookaside Buffer

- Hardware Cache for Virtual $\rightarrow$ Physical Address Translations
- Typically Small, High Associativity

Problem: TLB Penalties

- **Software Management of TLBs**
  - e.g., DEC Alpha, MIPS R2000/3000, HP PA-RISC
  - Simplifies chip design, greater flexibility to OS

    Makes TLB misses expensive

- **Micro-kernel Operating Systems**
  - e.g., Mach 3.0, Chorus, Amoeba, V
  - Greater modularity

    Increases number of TLB misses

$\Rightarrow$ **Higher TLB Penalties [Nagle93, Chen93]**

TLB Penalties Increasing

- Higher Cache Penalties on Faster Architectures
- Multiple Clients and Servers
- Commercial Applications: OLE, OpenDoc

TLB Penalties

For MIPS R3000-based 40MHz DECStation

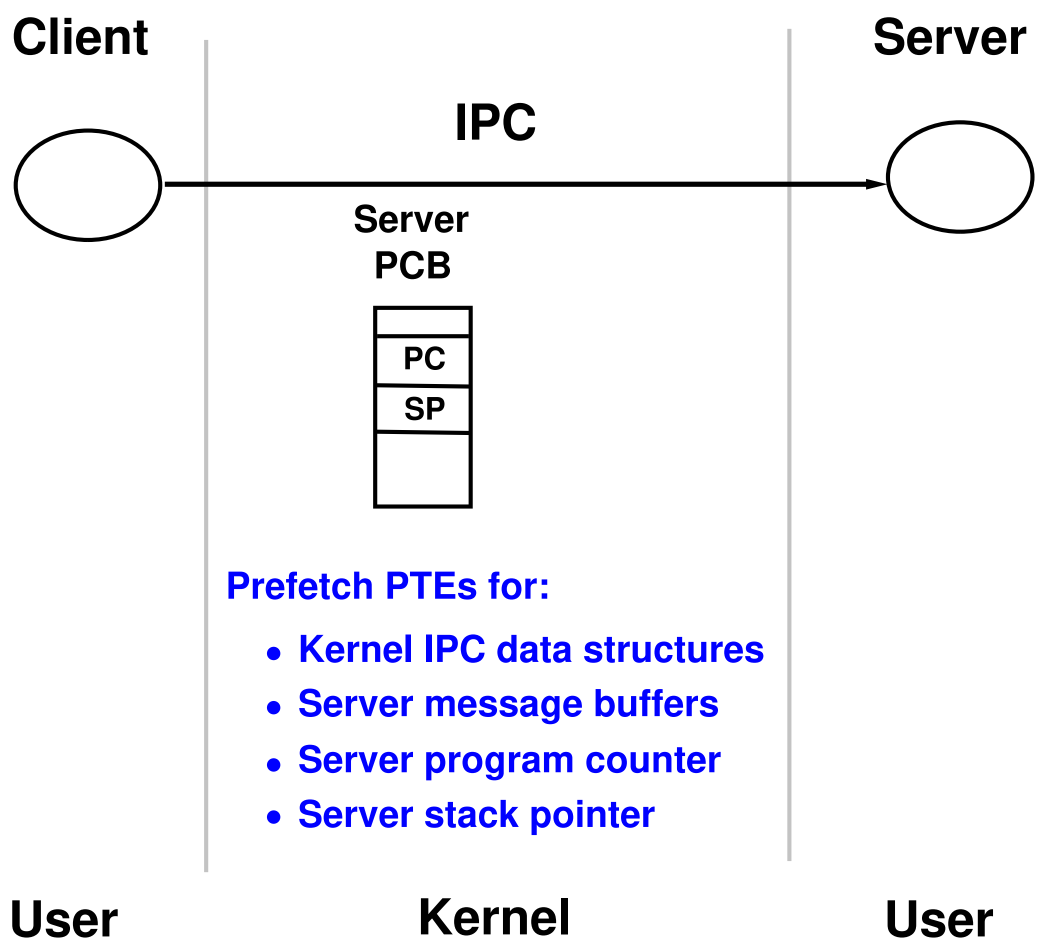
Miss Type	Penalty (cycles)	Handler Type
$User_{PTE}$	10 to 40	specialized
$Kernel_{PTE}$	512	generic
$User\text{-}PT_{PTE}$	555	generic
$Kernel\text{-}PT_{PTE}$	407	generic

Kernel TLB Misses

Solution #1: Prefetch PTEs

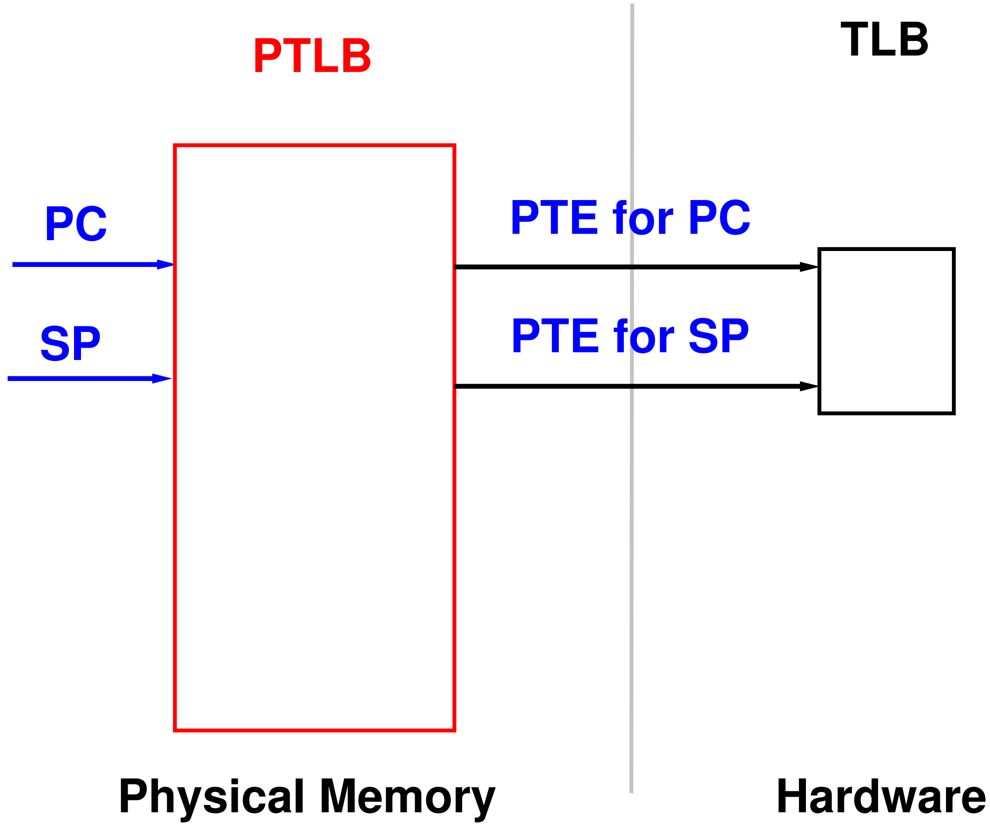
- **Many TLB Misses are Predictable**
- **Prefetch PTEs Directly into Hardware TLB**
- **Prefetch on IPC Path**

Prefetch Example



- Eliminates 50% of Kernel TLB Misses

Prefetch Implementation



- Direct-Mapped
- First IPC:
  Probe TLB,
  Load into PTLB
- Later IPCs:
  Look up in PTLB,
  Load into TLB

Experimental Methodology

- **Platform**
  - MIPS R3000, 64-entry fully-associative TLB
  - Mach 3.0 (MK 82)
- **High Resolution Timer**
- **Reduced Variability**
  - Multiple runs
  - No network traffic

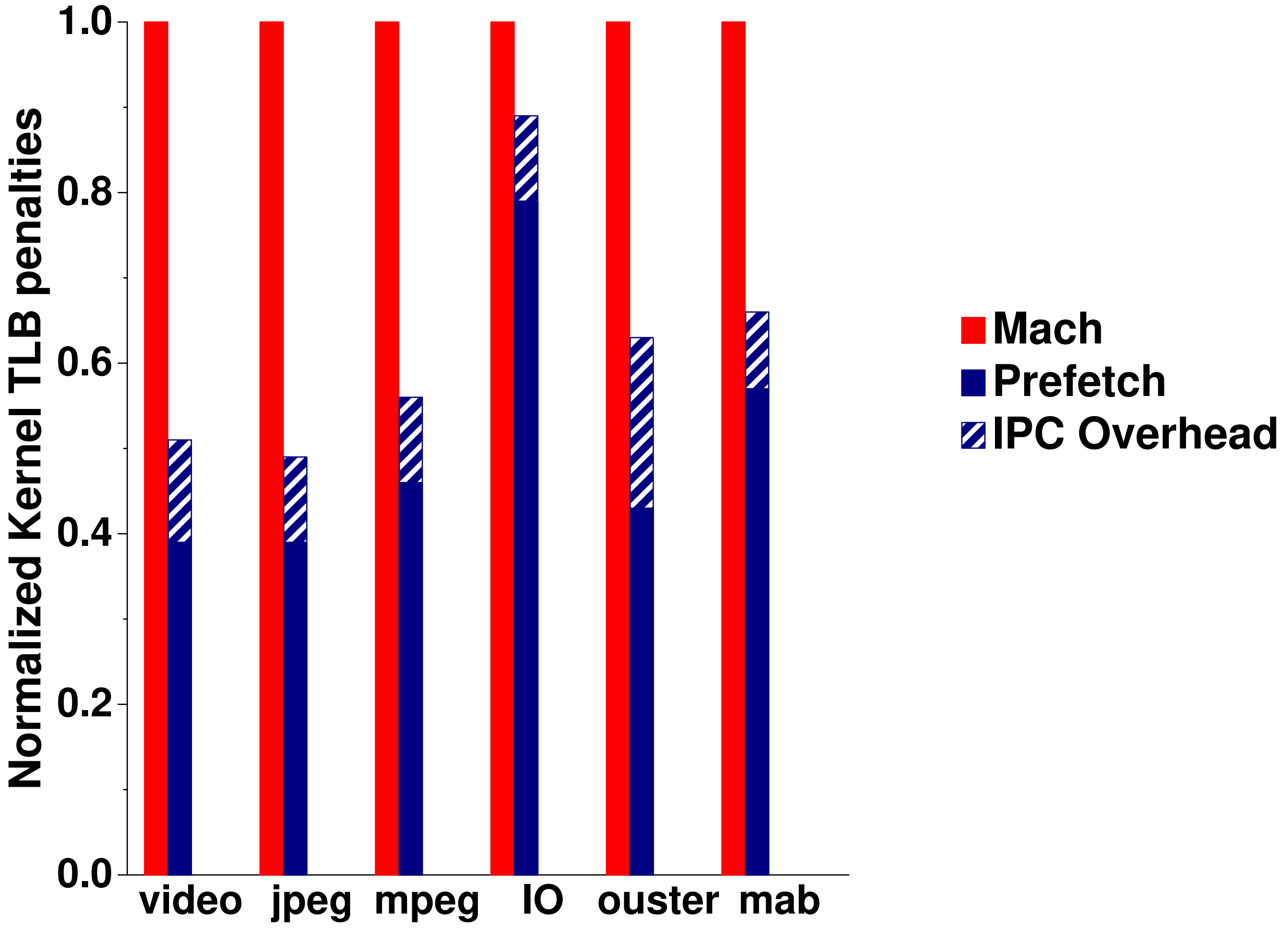
Benchmarks

	# TLB misses	Runtime in secs	TLB Misses per msec
video	211000	37	5.7
mpeg	245500	78	3.1
jpeg	25600	58	0.4
IO	5900	114	0.1
ouster	43000	52	0.8
mab	89600	214	0.4

Multiple Servers

Single Server

Prefetch Results



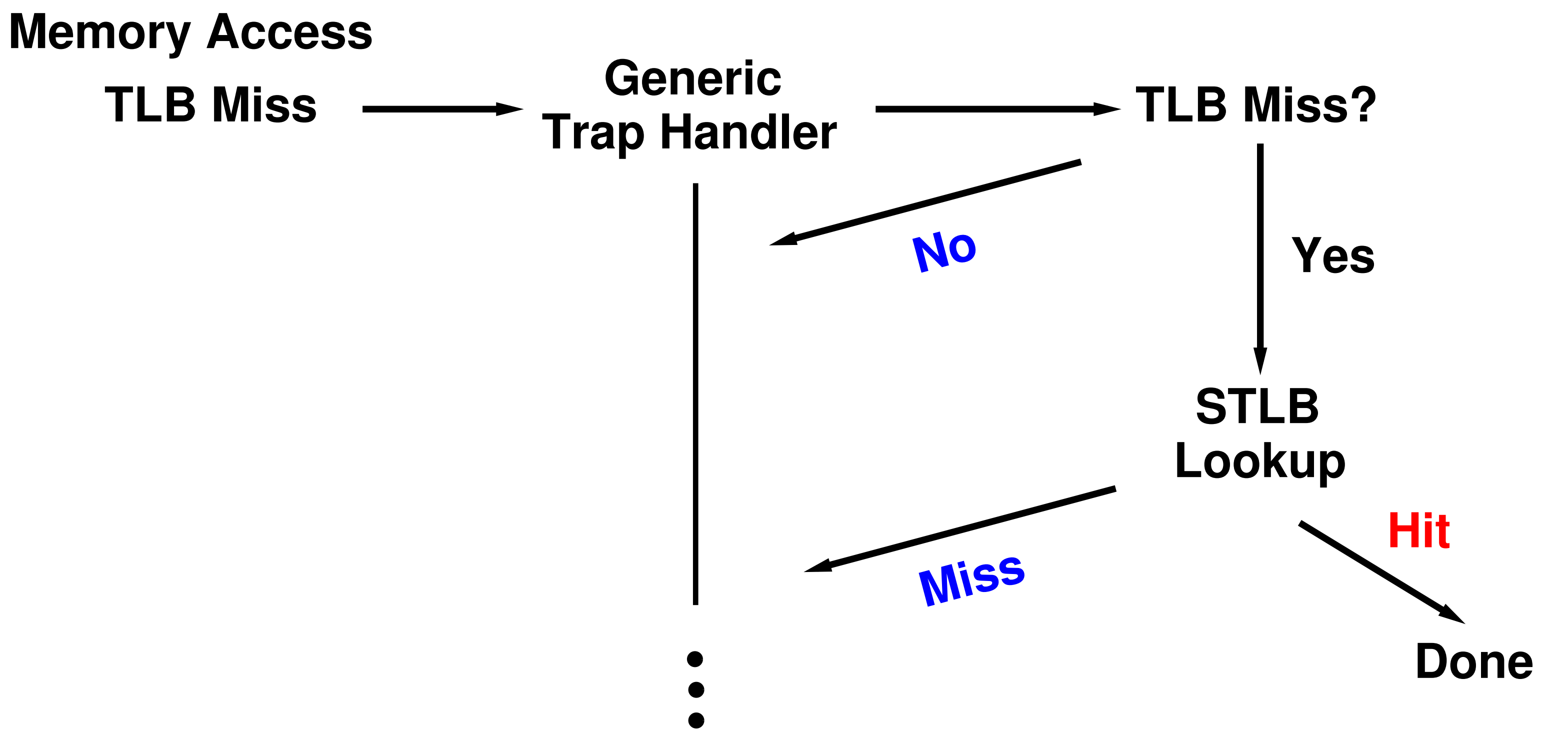
Prefetch Summary

- **Prefetch PTEs on IPC Path**
  - Server message buffers
  - Client and server PC and SP
  - Kernel IPC data structures
- **Eliminates up to 50% of Kernel TLB Misses**
- **More Effective with More Communicating Processes**

Solution #2: Software TLB

- **Software, Second-level Cache for PTEs**
- **Advantages**
  - Reduces cascaded misses
  - Lower cost for each Kernel TLB Miss
- **Related: Huck and Hays ’93**

STLB Implementation



- Fast Path to PTEs
- Direct-Mapped

Hit Rates and Penalties

Direct-Mapped STLB

Hit Rates

	STLB Size	
	1K	4K
video	71%	100%
mpeg	83%	98%
jpeg	85%	100%
IOzone	99%	99%
ouster	86%	99%
mab	71%	93%

- **STLB Hit $\approx$ 100 cycles**
- **Mach Trap Handler for TLB Misses $\approx$ 400-500 cycles**

Eliminates Cascaded Misses

Number of Cascaded Misses

	Mach	4K STLB
video	61400	0
mpeg	72300	3300
jpeg	7900	0
IOzone	1700	0
ouster	13100	0
mab	26400	200

- **Cascaded Misses are 30% of Kernel TLB Misses**

STLB Results

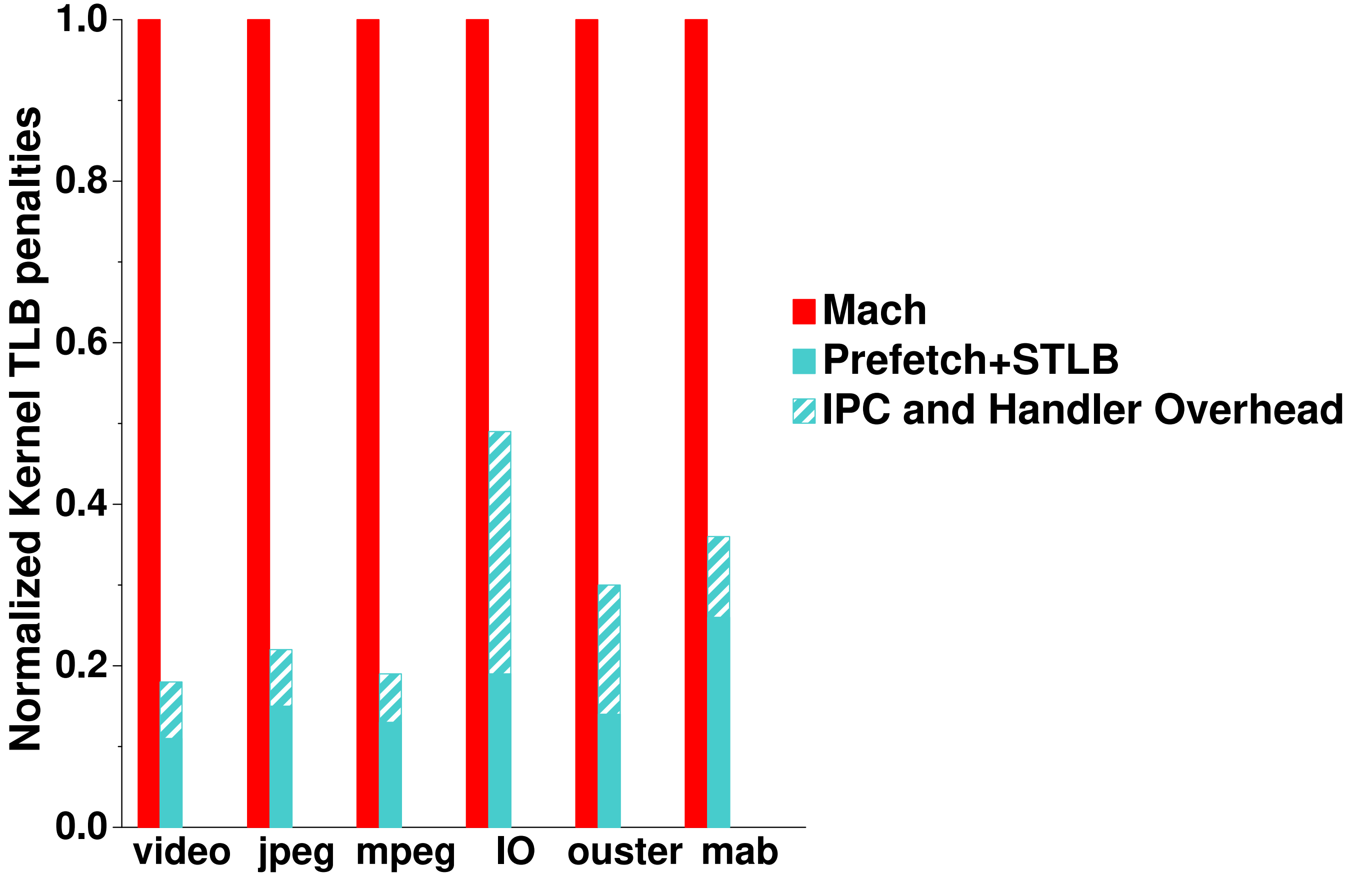
STLB Summary

- **Advantages**
  - High hit rates
  - Fast path access for all Kernel TLB misses
  - Eliminates cascaded misses
- **Limitations**
  - Adds overhead to trap handler

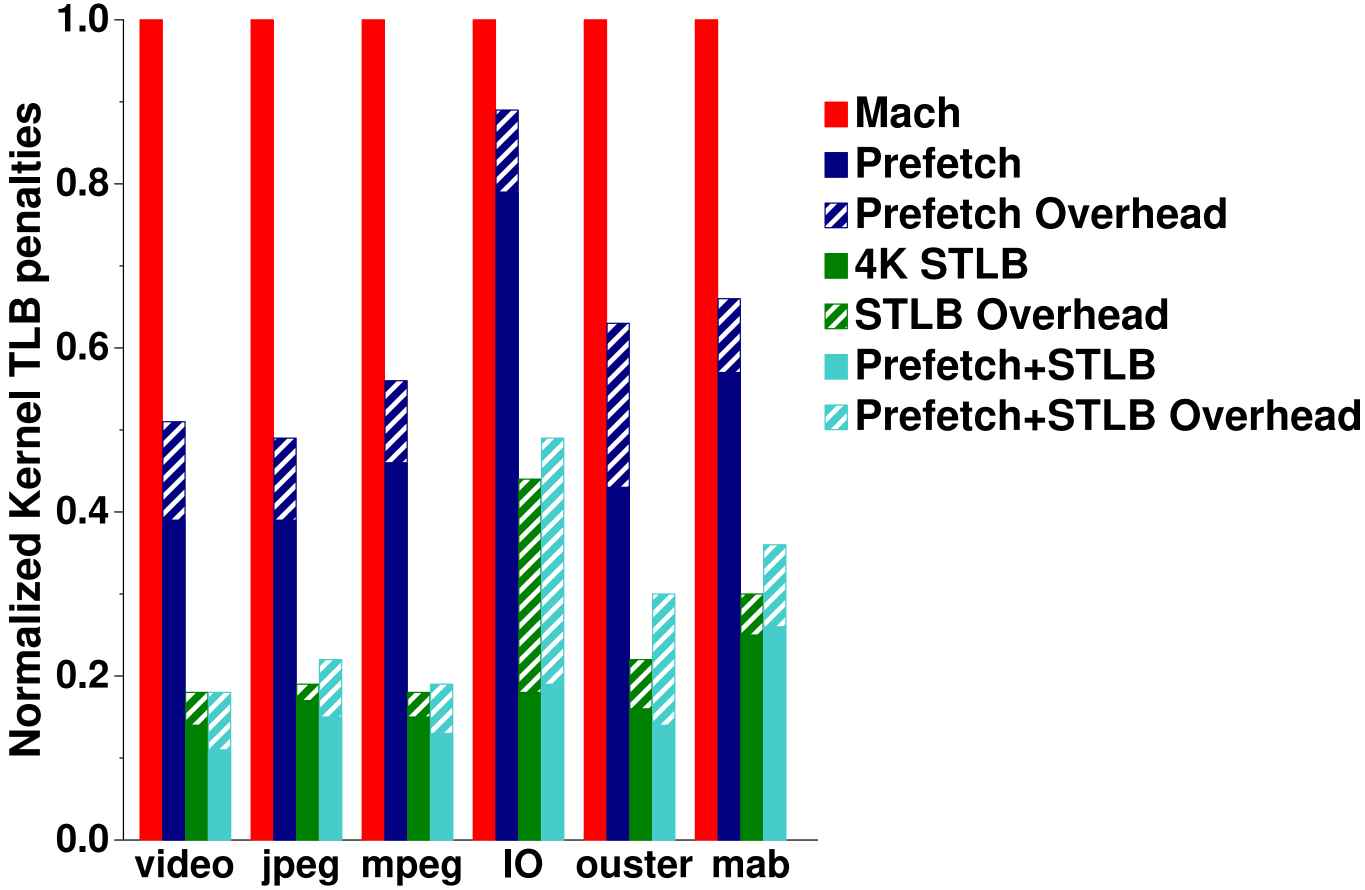
Integrated Scheme

- **Integrated STLB and PTLB**
  - Single data structure
  - Access from IPC and trap handler
- **Avoid Redundancy**

Prefetch+STLB Results



Comparion between Schemes



Application Speedup

Percent Improvement in Application Runtime

	PTLB	4K STLB	Prefetch+ STLB	Approximate Ideal
video	3.5	3.5	3.0	7.0
mpeg	1.7	1.7	1.1	2.6
jpeg	0.8	0.2	0.3	0.6

Multiple Clients

- Synthetic Database-like Workload
- IPC-intensive
- 4 User Processes: 3 Clients and 1 Server
- Percent Improvement in Application Runtime

Prefetch	6%
STLB	10%
Prefetch+STLB	12%

Architectural Trends

- **Cache Miss Times**

  24 cycles on 40 MHz $\rightarrow$ 90 cycles on 200 MHz

- **Effect on TLB Miss Times**

Mach 3.0	**500 cycles $\rightarrow$ 1500 cycles**
STLB	**100 cycles $\rightarrow$ 300 cycles**
Prefetch	**Eliminates TLB Misses**

Conclusions

Two Software Solutions

- **Prefetch**
  - Prefetch PTEs into hardware TLB on IPC path
  - Eliminates 50% of Kernel TLB misses
- **Cache**
  - Fast path for PTEs
  - Eliminates cascaded Kernel TLB Misses

Decreases the Number and the Cost of TLB Misses

Q&A

Kavita Bala

MIT

kaybee@lcs.mit.edu

http://www.psg.lcs.mit.edu/˜kaybee