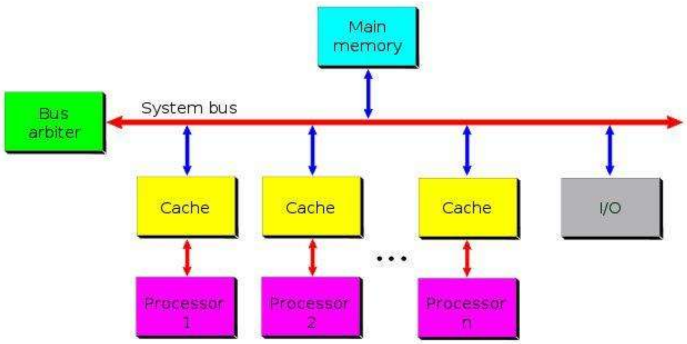
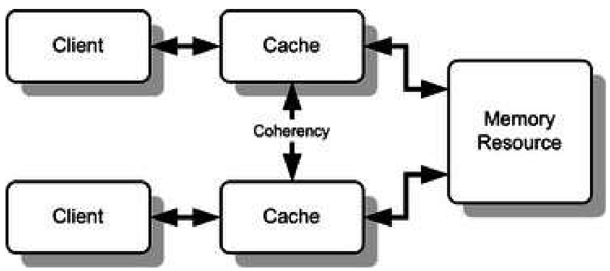
Disco: Running Commodity Operating Systems on Scalable Multiprocessors

Amogh Lonkar

Shared Memory Multiprocessing



- Improve performance
- Centralized main memory shared among many cores
  - Localized cache for quick access
  - Coherency required for correctness
- NUMA-based commercial models released in mid-1980s
  - SGI Origin2000, Honeywell XPS-100



Challenges of SMP

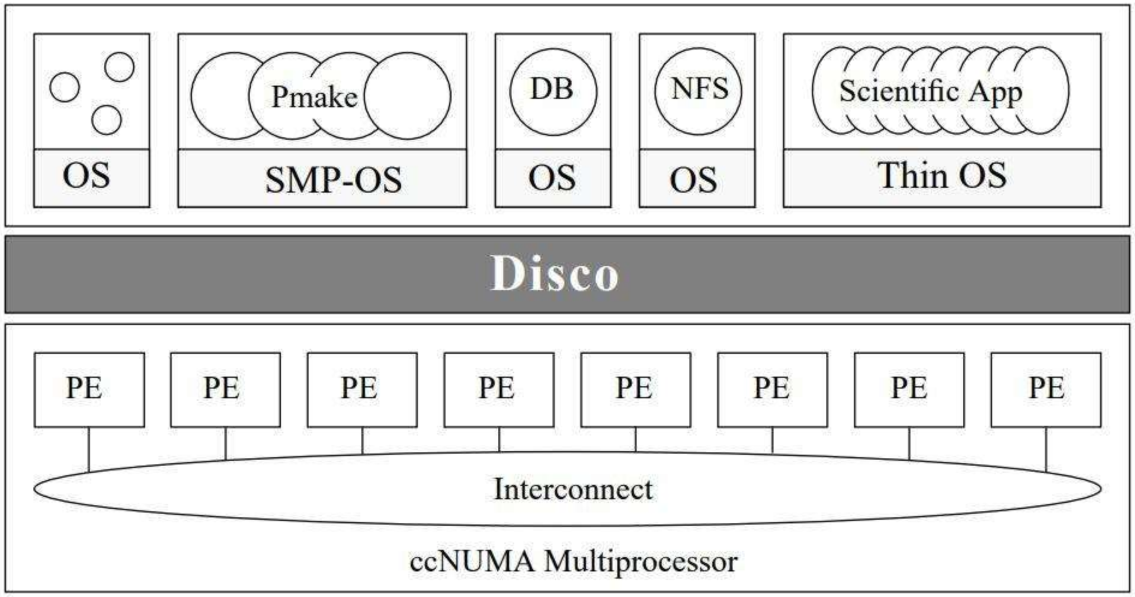
- Requires significant support from OS
  - Resource, work allocation
  - System partitioning
  - ccNUMA management
- Long effort to develop system software
  - Usually late, incompatible and buggy
- Reliability vs Performance
- HW vendors need to convince SW companies to port over

How can we make use of innovations in hardware without getting bogged down in developing system software?

Solution

- Insert ‘Virtual Machine Monitor’ between HW and OS
- Interface between HW and OS
  - Manages resources of virtual machines
- “Small” implementation effort

Virtual Machines



Virtual Machines (cont.)

Benefits:

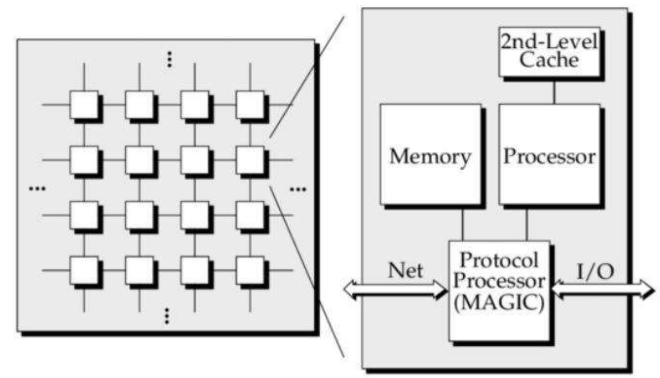
- Flexible
  - Can run many OS at once
- Scalable
  - Add more monitors
- Fault containment
  - Isolate SW failures in the VM
- Backwards compatible
  - Can run older OS version for legacy code

Virtual Machines (cont.)

Challenges:

- Additional Overhead
  - Emulating privileged instructions, exception processing, intercept+remap I/O request, memory usage
- Resource Management
  - Make policy decisions without best information
- Sharing and Communication
  - Non-trivial file sharing
- Security (?)

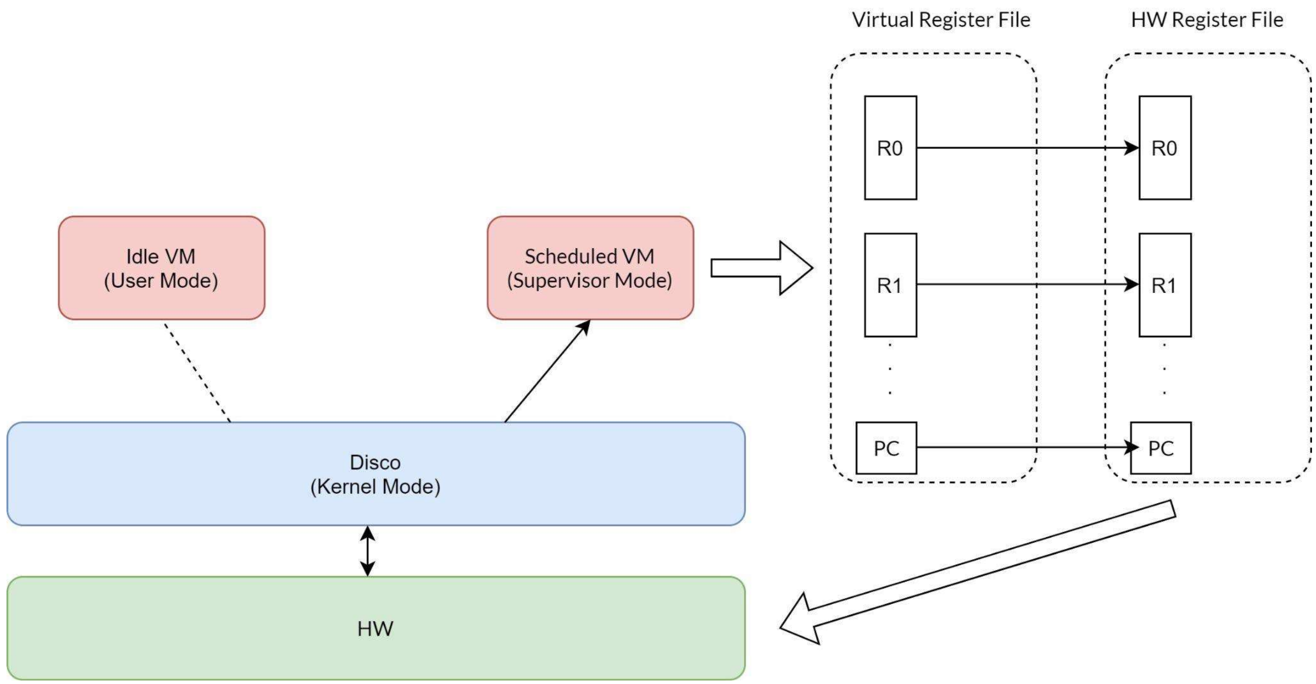
Disco



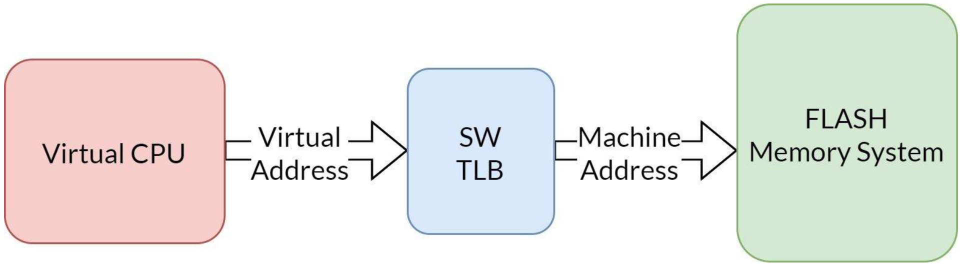
- Designed for FLASH multiprocessor
- Runs multiple individual virtual machines simultaneously
- MIPS R10000 abstraction for virtual CPU
- Main memory as contiguous physical address space
- Provides illusion of UMA to OS
    - Dynamic Page Migration/Replication
- Virtualizes I/O devices

Virtualizing CPUs

- Process table entry to save state
- Emulates operations not allowed in supervisor mode
- vCPUs time-shared over physical CPUs



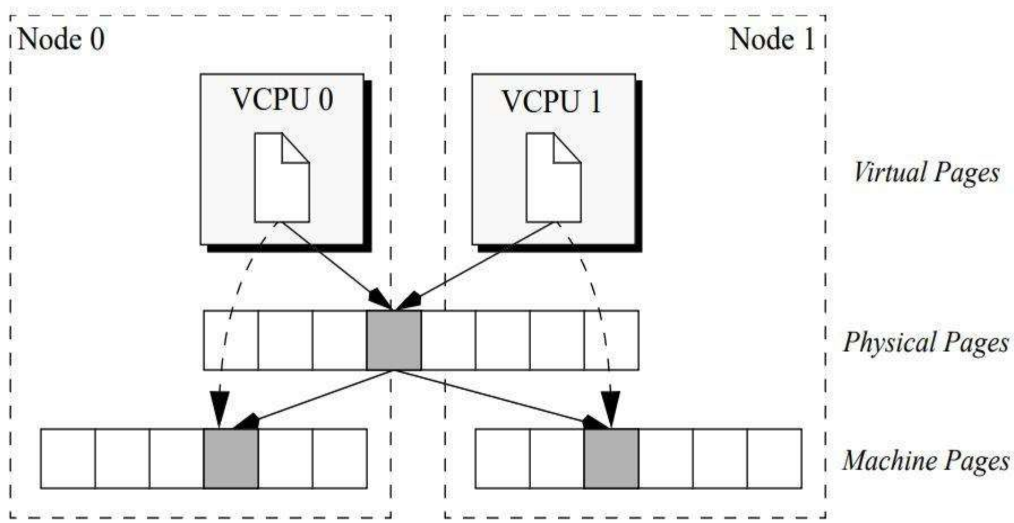
Virtualizing Memory

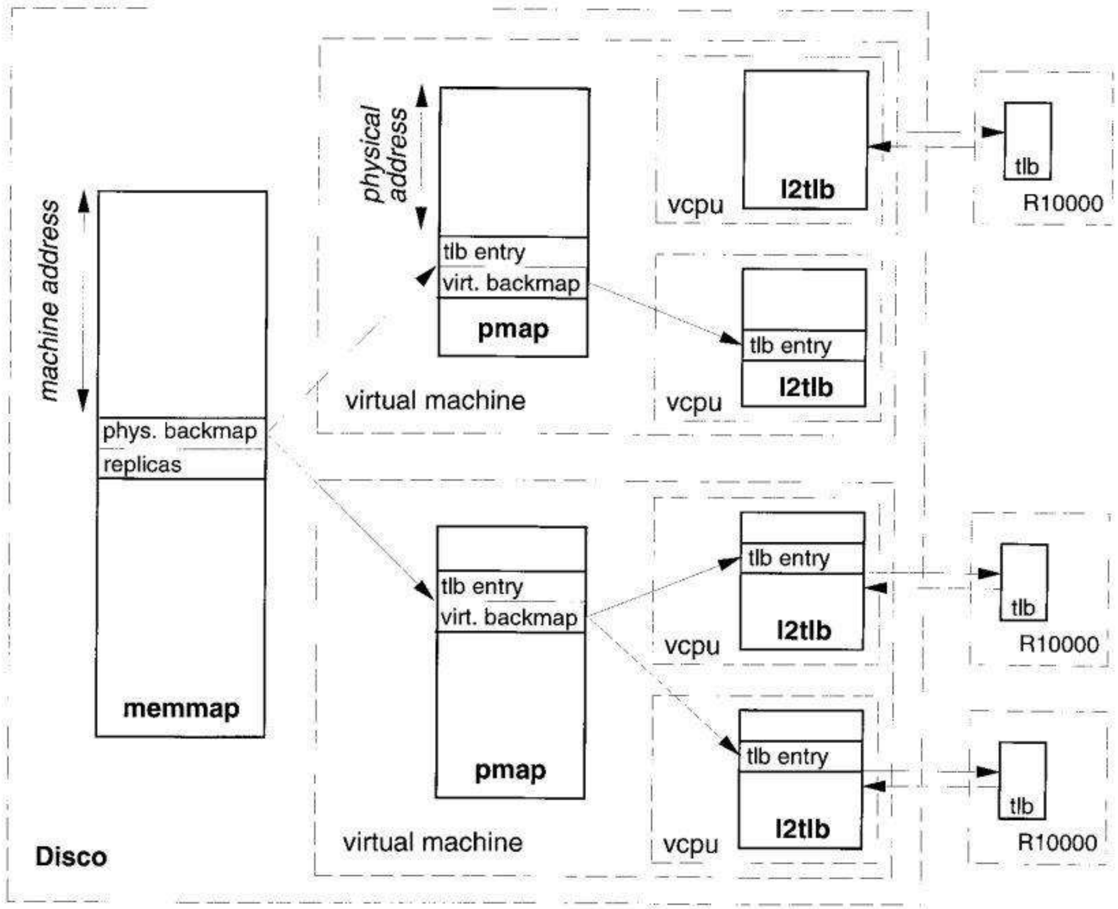


- Address translation requires extra remapping
- Optimization: pmap data structure
  - Entry per PPN
    - Pointer to physical page location
  - Merge entry with original protection bits
- Flush TLB on every vCPU switch to avoid virtualizing ASID
  - Problem: More TLB misses + higher miss overhead
  - Solution: Multilevel TLB heirarchy

NUMA Management

- Dynamic Page Migration/Replication
    - Improve locality
- Disco moves “hot” pages
- Heuristics to decide between migrating or replicating
    - Based on cache-miss counter in FLASH
- Algorithm:
    - Invalidate/Downgrade TLB entry
    - Copy to local machine page
    - Update TLB entry (only for replication)

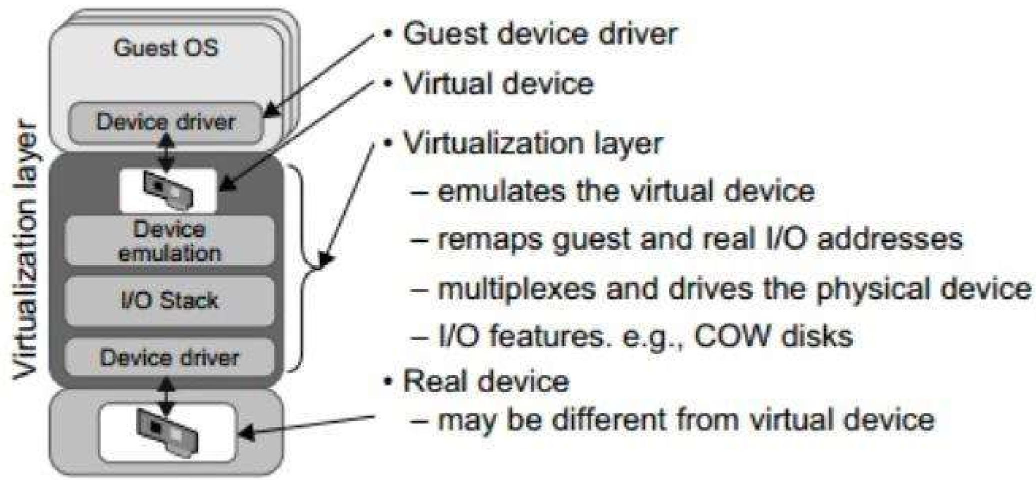




machine address
physical address
tlb entry
virt. backmap
pmap
vcpu
l2tlb
tlb
R10000
tlb entry
l2tlb
vcpu
virtual machine
phys. backmap
replicas
memmap
tlb entry
virt. backmap
pmap
virtual machine
tlb entry
l2tlb
vcpu
tlb
R10000
tlb entry
l2tlb
vcpu
tlb
R10000
Disco

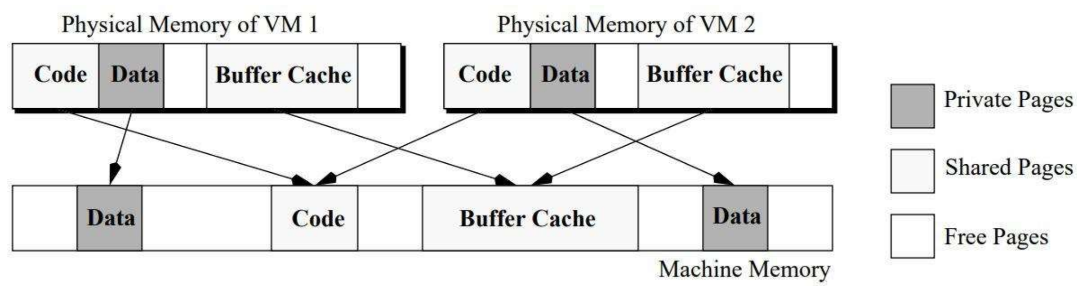
Virtualizing I/O

- Intercepts device accesses and forwards them to physical device



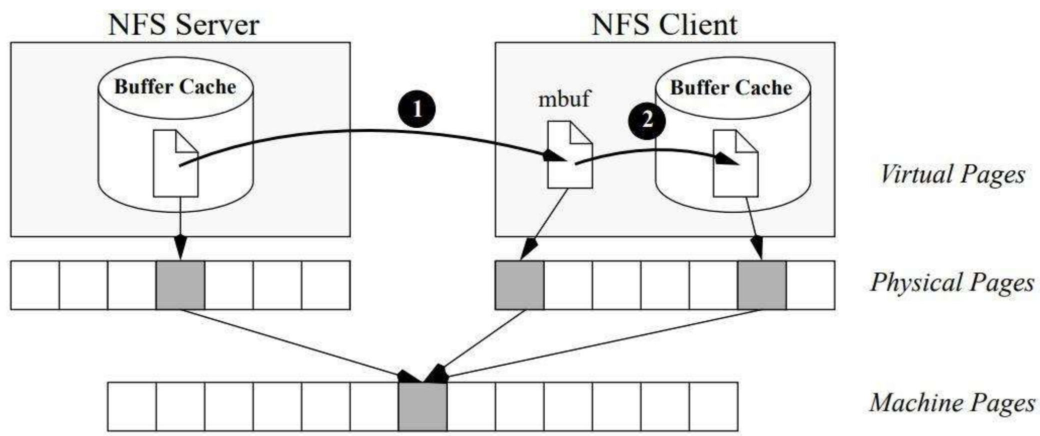
Copy-On-Write Disks

- Intercepts every disk request
- Copy-On-Write fault while modifying shared pages
- Multiple VMs share page by remapping
- 2 B-Trees for sharing
  - Index: Range of disk sectors requested, Entries: Machine memory address in global disk cache
  - Modifications to block



Virtual Network Interface

- VMs communicate via standard distributed protocols
- Virtual subnet for communication
  - Avoids data replication
- Share read-only pages
  - Poor locality (?)
- Replicate “hot” pages



Experimental Setup

- SimOS for modeling the FLASH multiprocessor
- Model statically scheduled, 1-wide processors at 2x clock rate

Workload	Environment	Description	Characteristics
Pmake	Software Development	Parallel compilation (-J2) of the GNU chess application	Multiprogrammed, short-lived, system and I/O intensive processes
Engineering	Hardware Development	Verilog simulation (Chronologics VCS) + machine simulation	Multiprogrammed, long running processes
Splash	Scientific Computing	Raytrace from SPLASH-2	Parallel applications
Database	Commercial Database	Sybase Relational Database Server decision support workload	Single memory intensive process

Execution Overhead

- All workloads run on uniprocessor
- 3%-16% overhead
- Reduction in kernel time

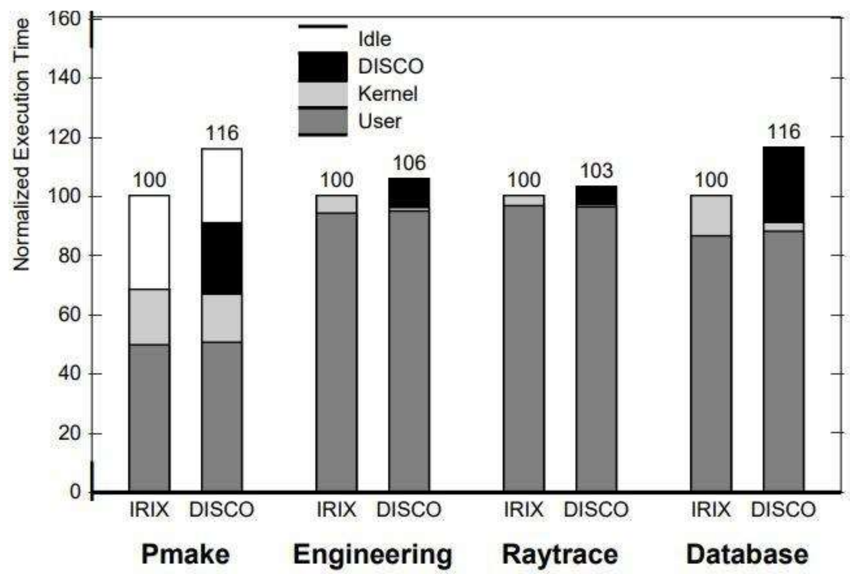
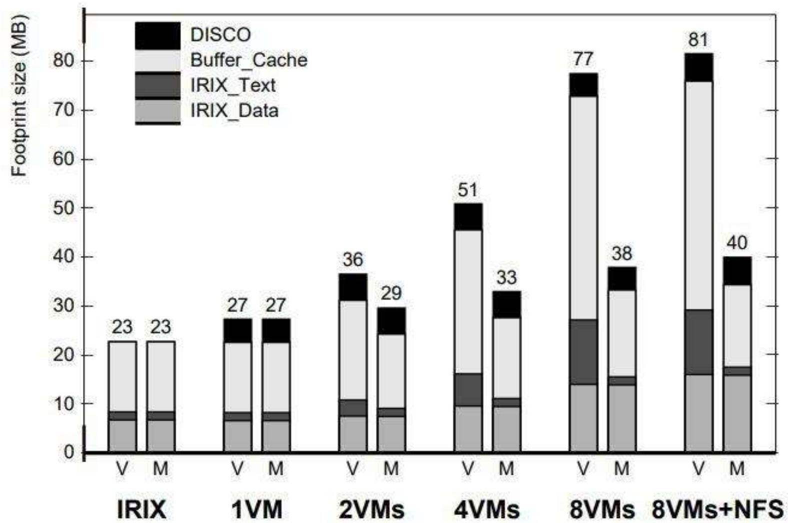


Table II. Overhead Breakdown for the Top Kernel Services of the Pmake Workload

OS Service	Execution on IRIX			Slowdown	Relative Execution Time on Disco				
	Time	Count	Avg time		Kernel Exec	TLB Writes	Other Privs	Monitor Services	Kernel TLB faults
	IRIX 5.3 – 32 bit – 4KB pages								
DEMAND-ZERO	30%	4328	21 μs	1.42	0.43	0.21	0.16	0.47	0.16
QUICK-FAULT	10%	5745	5 μs	3.17	1.27	0.80	0.56	0.00	0.53
open	9%	667	42 μs	1.63	1.16	0.08	0.06	0.02	0.30
UTLB-MISS	7%	630 K	0.035 μs	1.35	0.07	1.22	0.05	0.00	0.02
write	6%	1610	12 μs	2.14	1.01	0.24	0.21	0.31	0.17
read	6%	733	23 μs	1.53	1.10	0.13	0.09	0.01	0.20
execve	6%	42	437 μs	1.60	0.97	0.03	0.05	0.17	0.40
	IRIX 6.2 – 64 bit – 16KB pages								
DEMAND-ZERO	27%	1134	57 μs	1.01	0.17	0.05	0.10	0.68	0.01
execve	16%	42	608μs	1.30	0.81	0.01	0.03	0.33	0.11
open	10%	667	37 μs	1.37	1.17	0.00	0.08	0.02	0.11
read	6%	733	21 μs	1.32	1.13	0.00	0.12	0.00	0.07
write	6%	1640	9 μs	1.71	0.98	0.00	0.31	0.35	0.06
COPY-ON-WRITE	5%	120	94 μs	1.16	0.59	0.03	0.06	0.41	0.06
QUICK-FAULT	5%	2196	5 μs	2.83	1.28	0.49	0.89	0.00	0.16

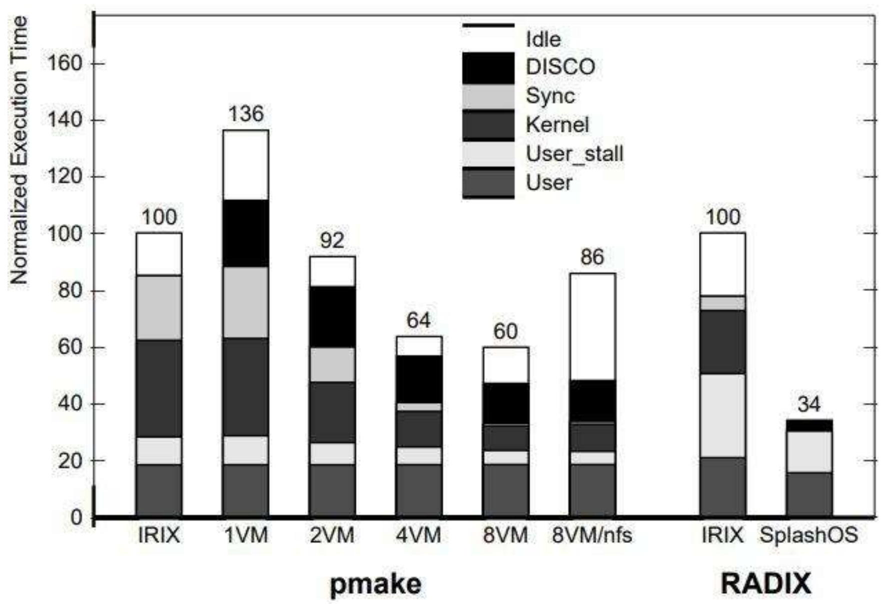
Memory Overhead

- Various system configurations
- Pmake is the best candidate
- Kernel text and buffer size footprint constant



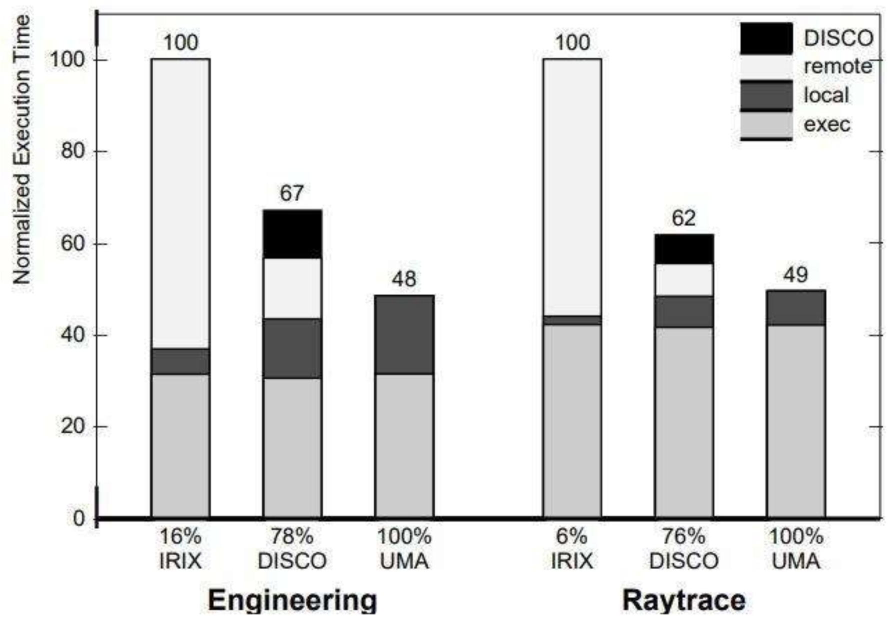
System Scalability

- Pmake
  - High synchronization overhead for IRIX (spinlock)
  - Slow critical region for Disco with single VM
  - Reduced kernel stall and synchronization
- Radix Sorting
  - Lazy page allocation in IRIX leads to kernel stalls
  - NUMA-aware policy reduces hot spots and user stall



Page Migration and Replication

- Workloads with poor memory behavior
- Engineering
  - 6 verilog simulations + 6 memory system simulations on 8 cores
- Raytrace
  - Spans 16 cores
- Achieves close to UMA performance



Impact

- Authors went on to co-found VMware
- Cellular Disco: Virtual clustering service
- FLUSH+RELOAD: Side-channel attack technique for page-sharing systems
- VMM-bypass for time-critical I/O
- Dune
- Multikernel, IX

Discussion Questions

- What are the security implications of sharing memory and emulating privileged instructions between multiple OSes on the same machine?
- The system seems to assume access to high performing TLBs and proper memory alignment? How realistic is this assumption? Should we expect to see similar results on longer running workloads?
- What benefits do VMMs have over systems like the Multikernel? How is the Multikernel better? (Inspiration from Lakshmi’s question in the review)