

Disco: Running Commodity Operating Systems on Scalable Multiprocessors
Bugnion, Devine, and Rosenblum
SOSP 1997

1. What was the goal of Disco?
2. What are the pros and cons of a CC-NUMA architecture (vs SMP UMA)? What are some issues with building or modifying an OS for CC-NUMA?
3. What are the advantages of Virtual Machines?
4. At a high level, what are some of the challenges of Virtual Machine?
5. How does Disco virtualize the MIPS R10000 CPU? What happens on a system call w/o and w/ Disco? Why are 3 modes for user, supervisor, and kernel key?
6. How does Disco virtualize memory? What will be held in the TLB? What happens on a TLB miss with w/o and w/ Disco? What data structure does Disco add?
7. What complexity was caused by IRIX living in kseg0? What was Disco's solution?
8. Why are TLB misses more significant with Disco? What is Disco's solution?
9. What are Disco's goals that are specific to NUMA? When should a page be replicated? migrated? How does Disco perform replication and migration? What should happen if a page is heavily write-shared?
10. Why are large memory footprints a concern for Disco? Why does sharing occur across VMs? Why is copy-on-write useful?
11. Running a completely unmodified commodity OS on Disco is tricky. What changes did Disco make to IRIX to improve performance?
12. As shown in Figure 5, how much time overhead does DISCO impose for a uniprocessor workload? Why does some of the original kernel time decrease?
13. What does Figure 6 show? Does Disco do a decent job sharing buffer cache space across VMs? Of sharing IRIX text? IRIX data?
14. What does Figure 7 show? Where can you find an evaluation of Disco's replication and migration policies?
15. Conclusions?