**Motivation**

Emulab is a widely-used network testbed
- Experiment = network of physical and virtual nodes
- Primitive FCFS scheduling

Demand for nodes far exceeds capacity

Inactive experiments are destructively "swapped-out"
- Accumulated experiment state lost

**Goal**

Time-sharing in Emulab through preemptive scheduling

Key enabler: Stateful swapout of experiments

**Key Techniques**

VM encapsulation (Xen VMM)
- Suspend/resume node execution
- Snapshot node-local state

Consistent group checkpoint
- Snapshot global (experiment-wide) state

Time virtualization
- Inactivity between swapout and swapin transparent to experiment

**Major Challenge**

Make experiment context switch fast enough to be practical
- Per-node memory and disk state
- 100s of nodes

**Addressing the Challenge**

Reducing the context switch time

Pipelining
- Proactive swapin of incoming experiment
- Lazy swapout of outgoing experiment

**Scalable storage server**

Separate metadata and data paths
- Metadata server stores content hashes
- Eliminates intra- and inter-experiment redundancy
- CAS bricks store the data blocks
- Enables parallel data transfer

**Minimizing swapped-out state**

Exploit data redundancy
- Copy-on-write branching storage system for node-local redundancy
- Store only changes since swapin
- Content addressing for intra- and inter-experiment redundancy
- Redundant data never sent on the wire or stored

**Local redundancy elimination**

VM disks are CoW branches of the base image. During a swapout, the content hashes of the branches are sent to the metadata server.

**The Vision**

Emulab as an OS-like entity that takes scheduling decisions, "pages out" idle nodes and manages physical resource utilization through VM migration and ballooning.

**Status**

We have implemented the key techniques for stateful swapout and CoW branching storage. The scalable storage server is currently a work-in-progress.