Linux Computation Scrapbook System

Richard Potter

Flexible Computation Scrapbook For Linux

- Save multiple copies
- Save persistent data (e.g. hard disks)
- Also save runtime transient state
  - specific runtime configuration
  - window positions
  - invisible state
  - etc.

Demo Many Eyes Snapshot

Related Techniques

- Process Migration, Fault tolerance, Fast Initialization
- Computation Scrapbooks Distinctions
  - Multiple Persistent Copies
  - Thread persistence
  - Use for programming tools
  - Generality of Snapshot Programming idea

Related Work

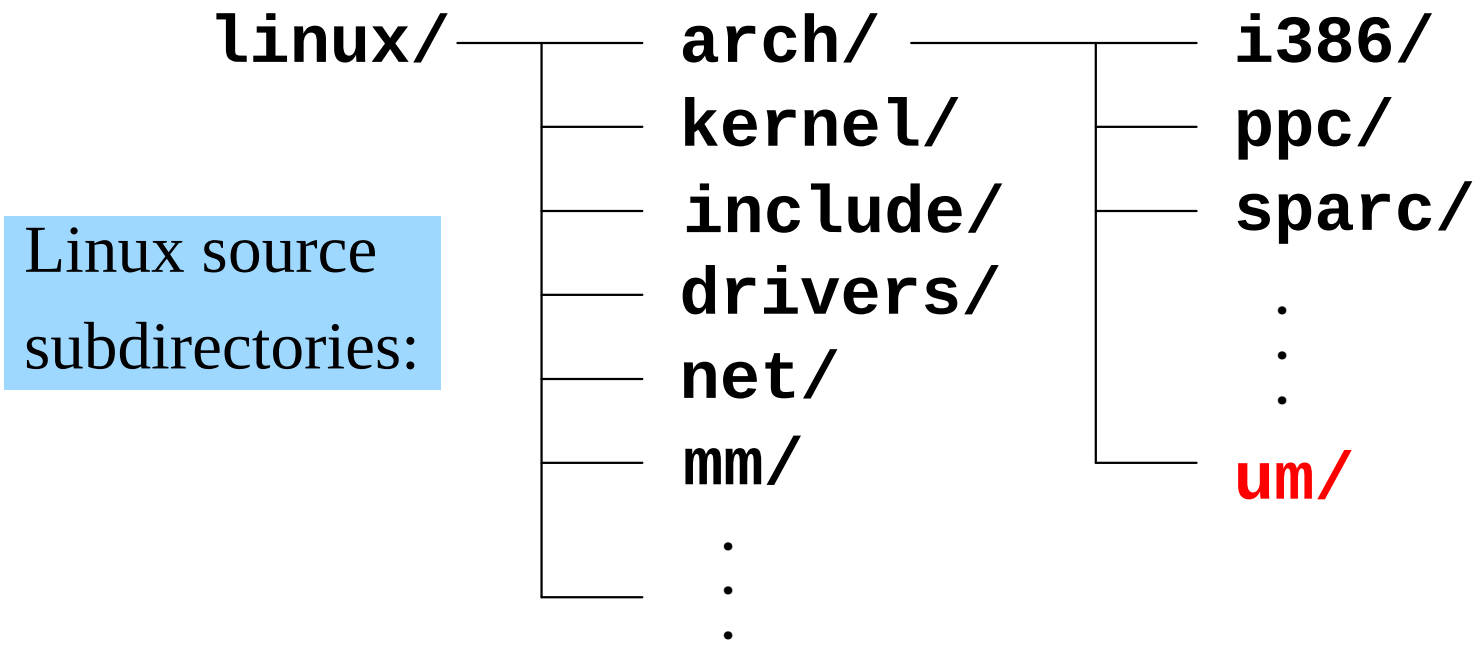
- SoftwarePot [Kato, Oyama ’02]
  - allows sandboxing of arbitrary code with an encapsulated, transferable file system
  - also distributes configuration information
  - flexible access policies based on Secure Software Circulation (SSC) model
  - does not checkpoint runtime state

Contents of Many Eyes Snapshot

- 660MB of Hard Disk
- 32MB of RAM
- 38 Processes
- 800x600 pixel VNC GUI

User-Mode Linux Implementation

http://user-mode-linux.sourceforge.net



- Linux ported to itself
- Almost all changes in the architecture-specific **um/** directory

UML Implementation

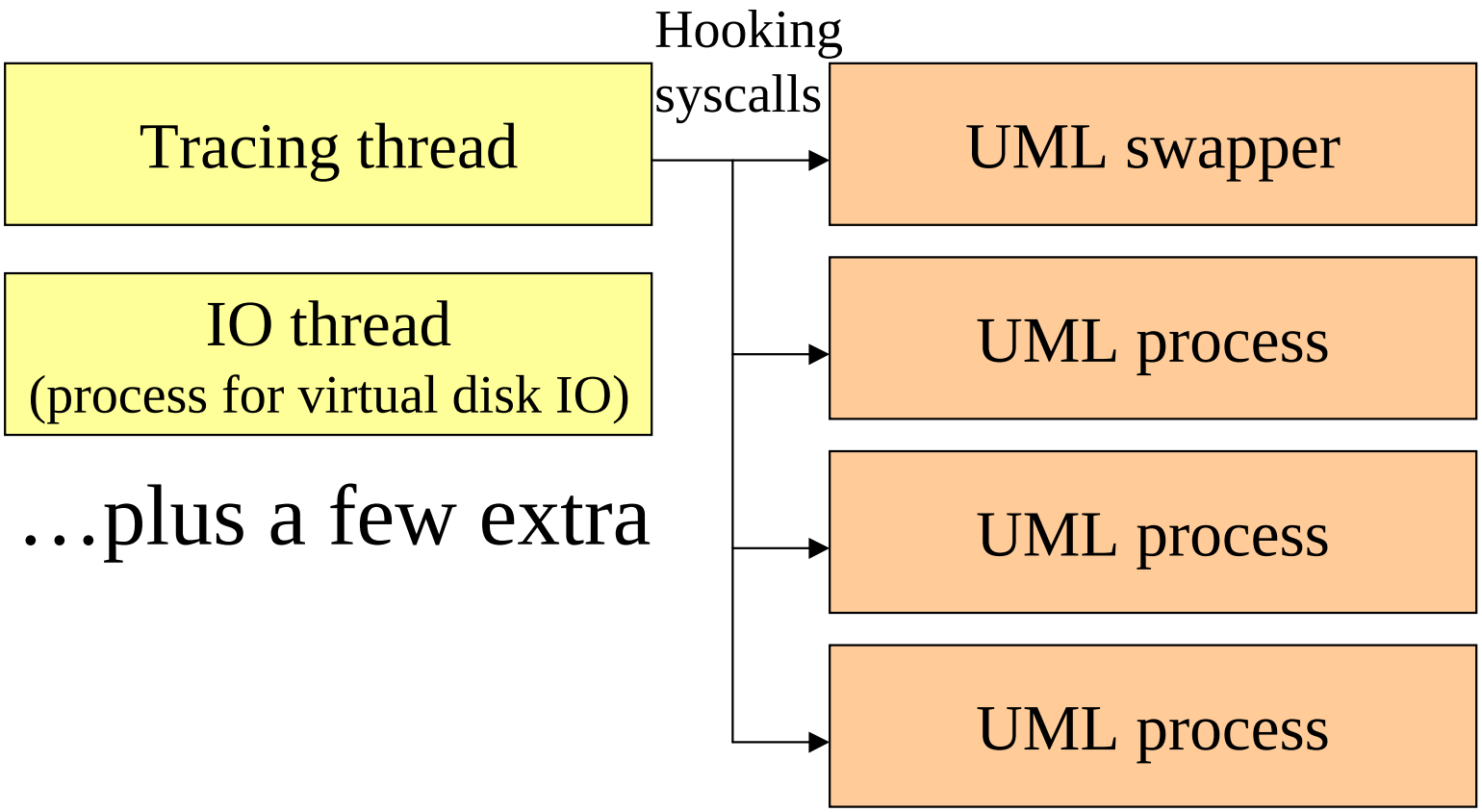
Real Linux HW	UML Linux Host
RAM	Files
MMU	mmaps() to files
Hard disks	Files/COW files
Devices	Sockets, ttys, xterms
Networking	TUN/TAP
System Traps	interception by ptrace()

UML Implementation

```
[rlp@localhost demodir110303]$ ls -l machines/m1
total 34700
-rwxrwxrwx    1 rlp      rlp           233 Nov  4 00:44 ancestors
-rwxrwxrwx    1 rlp      rlp           113 Nov  4 00:40 basecmdline
-rwxrwxrwx    1 rlp      rlp           142 Nov  4 00:44 console
-rwxrwxrwx    1 rlp      rlp             6 Nov  4 00:44 ctpid
-rwxrwxrwx    1 rlp      rlp         19456 Nov  4 00:40 home_fs.delta
-rwxrwxrwx    1 rlp      rlp         55576 Nov  4 00:40 java_fs.delta
-rwxrwxrwx    1 rlp      rlp     940389888 Nov  4 02:15 root_fs.delta
-rwxrwxrwx    1 rlp      rlp        954369 Nov  4 00:44 status
-rwxrwxrwx    1 rlp      rlp         24105 Nov  4 00:44 stdout
-rwxrwxrwx    1 rlp      rlp         36888 Nov  4 00:40 swapfs.delta
-rwxrwxrwx    1 rlp      rlp        388376 Nov  4 00:40 usr_lib_fs.delta
-rwxrwxrwx    1 rlp      rlp        413976 Nov  4 00:40 usr_share_fs.delta
-rwxrwxrwx    1 rlp      rlp       1560577 Nov  4 00:40 vm-1
-rwxrwxrwx    1 rlp      rlp        159745 Nov  4 00:40 vm-2
-rwxrwxrwx    1 rlp      rlp        401409 Nov  4 00:40 vm-3
-rwxrwxrwx    1 rlp      rlp      33554433 Nov  4 00:40 vm-4
```

UML Implementation

- One host process for each guest process:



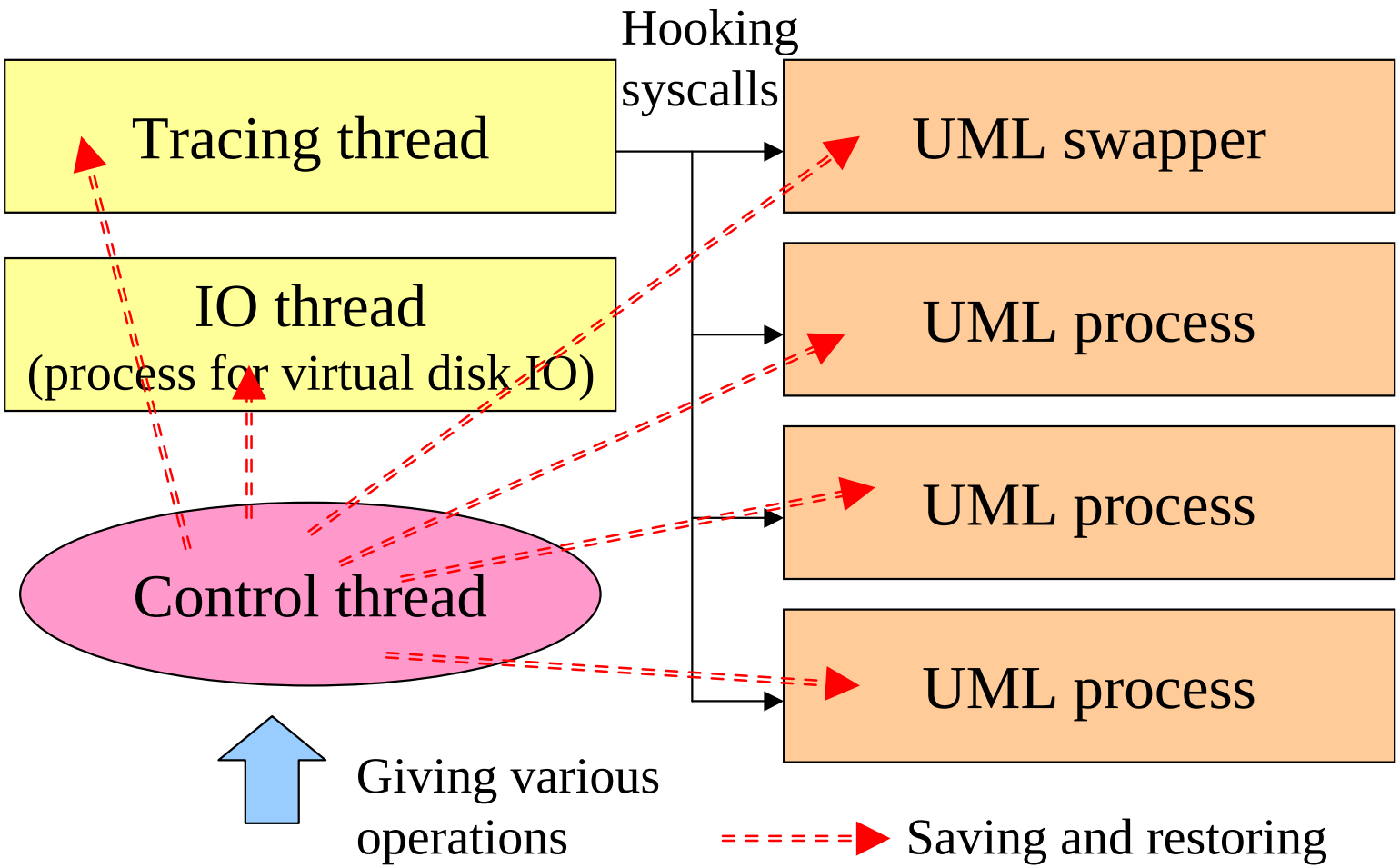
- …plus a few extra

SBUML Implementation

- Save Snapshot:
  1- freeze all processes into safe state
  2- save all process contexts
  3- save information about open file descriptors
  4- copy VM and COW files to snapshot directory
- Restore Snapshot:
  1- copy VM and COW files to machine directory
  2- reopen file descriptors
  3- recreate processes (mmaps and contexts)
  4- un-freeze all processes

SBUML Implementation

- New control thread to coordinate save/restore:



Scrapbook for User Mode Linux (SBUML)

- Adds snapshot save and restore to UML
- Core shell commands:
  - sbumlboot -boots a fresh machine
  - sbumlsave -saves snapshot image
  - sbumlrestore -restores snapshot image
  - sbumlhalt -halts and deletes machine

Snapshots many Linux-based applications possible

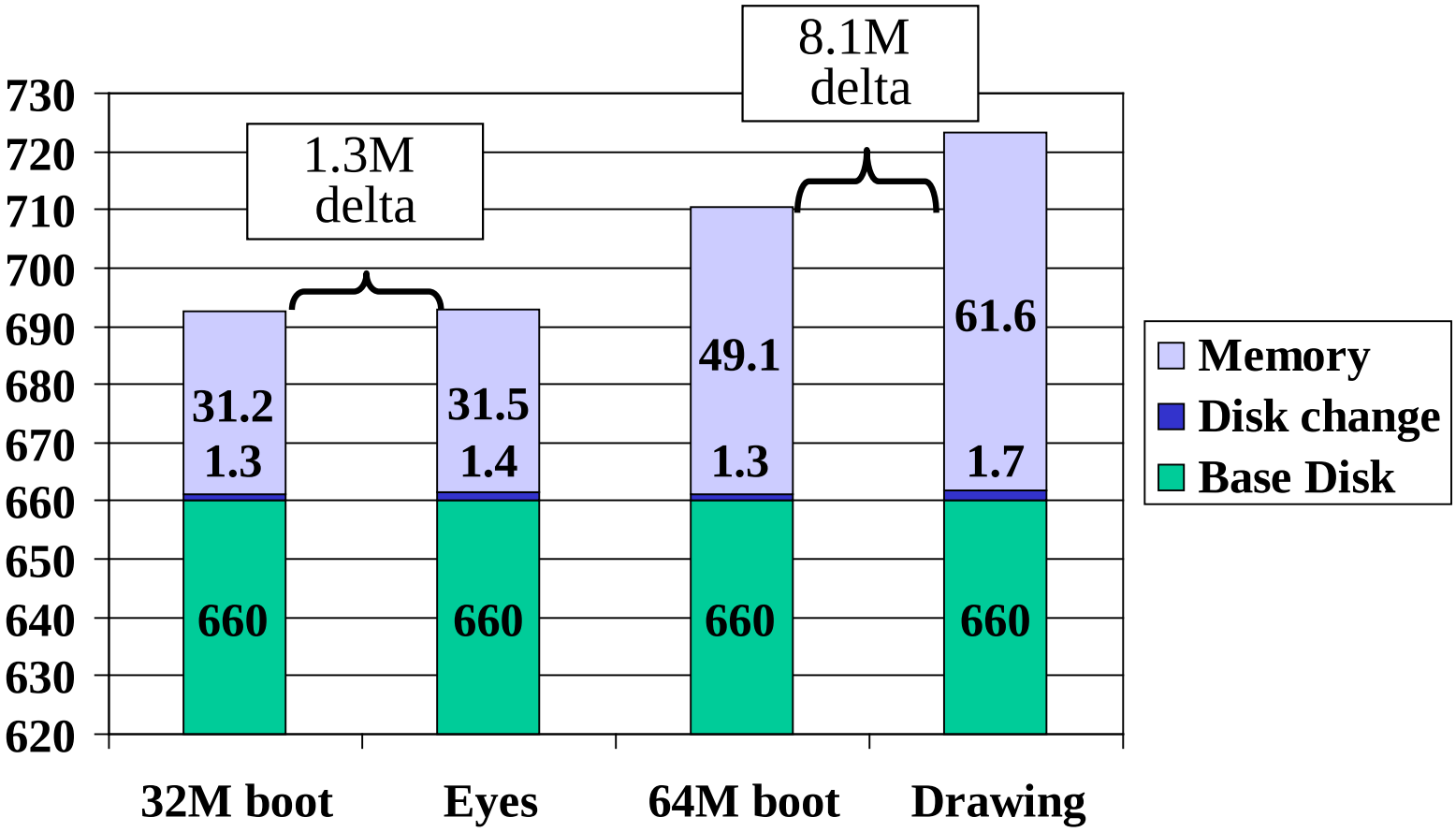
- Web servers
- Games
- Programming languages
- Office applications
- Image processing
- Emulators

….all can be saved in detail.

Drawing Program Demo

- restore
- edit, reconfigure
- save

Snapshot Sizes



Demo Delta Compression

With Small Snapshot Sizes...

- Can distribute 100’s on a CD
- Can archive 1000’s on a server
- Can download from the Internet
- Opens up many practical uses

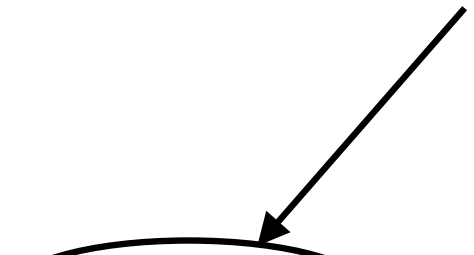
Many Uses for Distributing Pre-configured State

- Software demonstrations
- Pre-configured software distribution
- Language documentation
- Application feature documentation
- Collaborative debugging
- Classroom demonstrations

Snapshot Programming

- Simple example:

Like an object variable



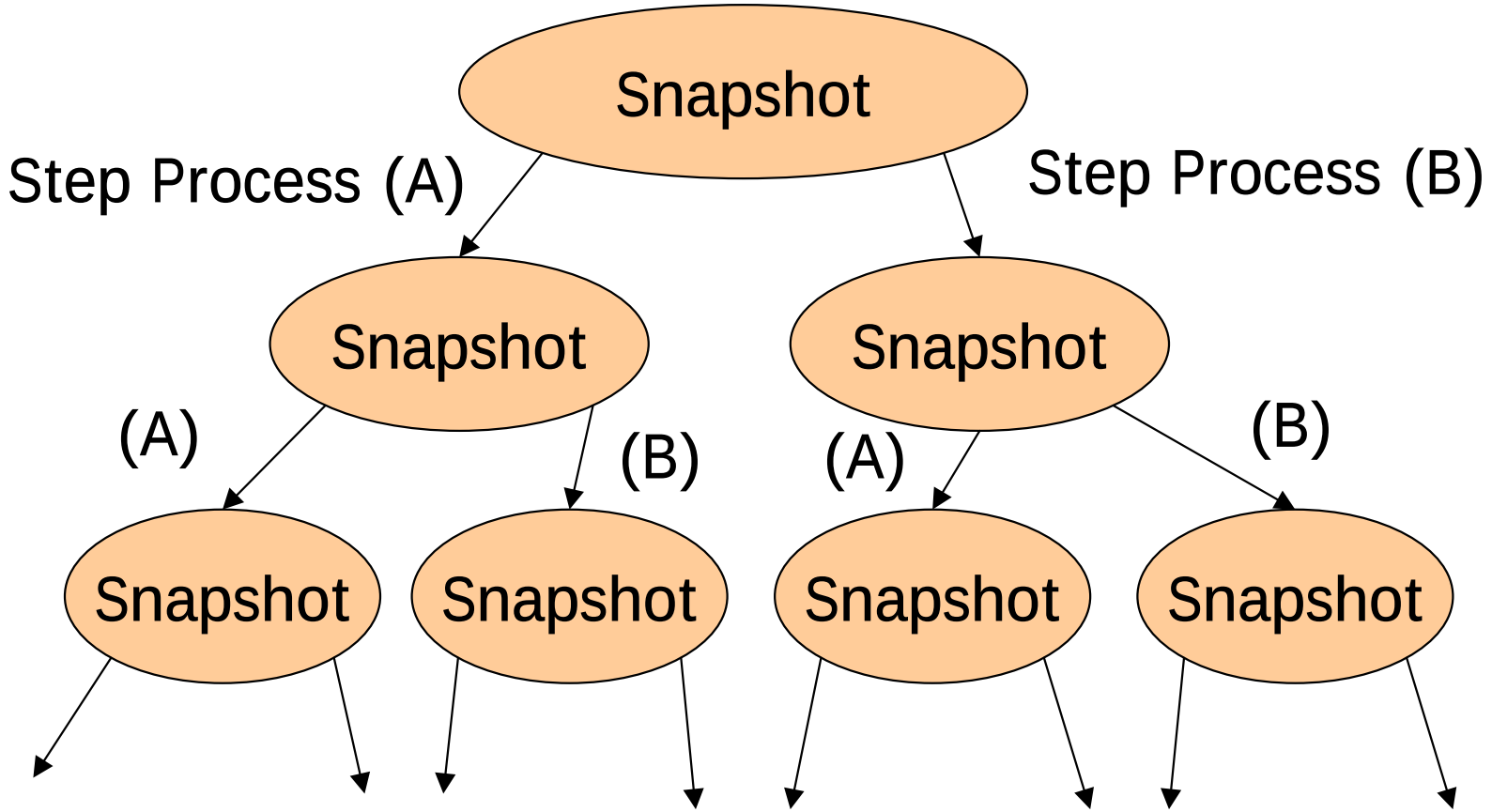
sbumlclone =

```
#!/bin/bash
sbumlsave $1 tmpclone -f -c
sbumlrestore m tmpclone -c -s
rm snapshots/tmpclone* -fr
```

- Therefore, potential for many creative applications….

Model Checking

successfully applied to dining philosophers



Ensure all states meet global specifications and avoid deadlock

Model Checking Script

```
1-  enumerate_state()
2-      BEGIN
3-          save_snapshot(temporary_snapshot);
4-          process_list = get_runnable_processes();
5-          IF process_list is empty THEN report "deadlock";
6-          FOR each p in process_list
7-              run_process_until_hash_value_changes(p);
8-              hash = compute_hash_value();
9-              IF hash is an element of hash_list THEN return;
10-             append hash to hash_list;
11-             verify_global_properties();
12-             enumerate_state();  // recursive call
13-             restore_snapshot(temporary_snapshot);
14-     END
```

Why is it Interesting?

- Previous model checking checked a separate model that might not match the actual implementation.
- Model checking with SBUML can be performed on actual code running on a real operating system.
- Systems that can be proved correct are important to society.

Conclusions

- Linux Computation Scrapbook
- Flexible saving of Linux runtime state
- Snapshots compression to a few MB
- Automatic download and restore from Web pages
- Snapshot programming applications
- http://sbuml.sourceforge.net/

Future Work

- Faster copying of sparse files
- Shared memory mappings like used with fork()
- Lazy restore of processes and mappings
  - current: 3 seconds, .03 seconds possible?
- Integration with remote files systems (Suzaki)