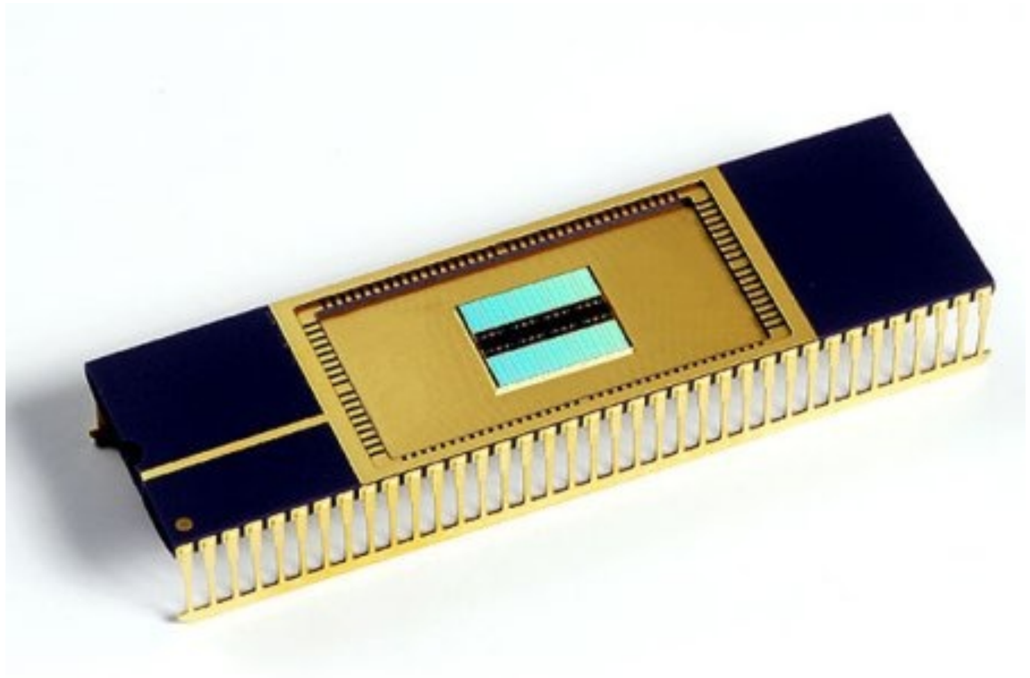
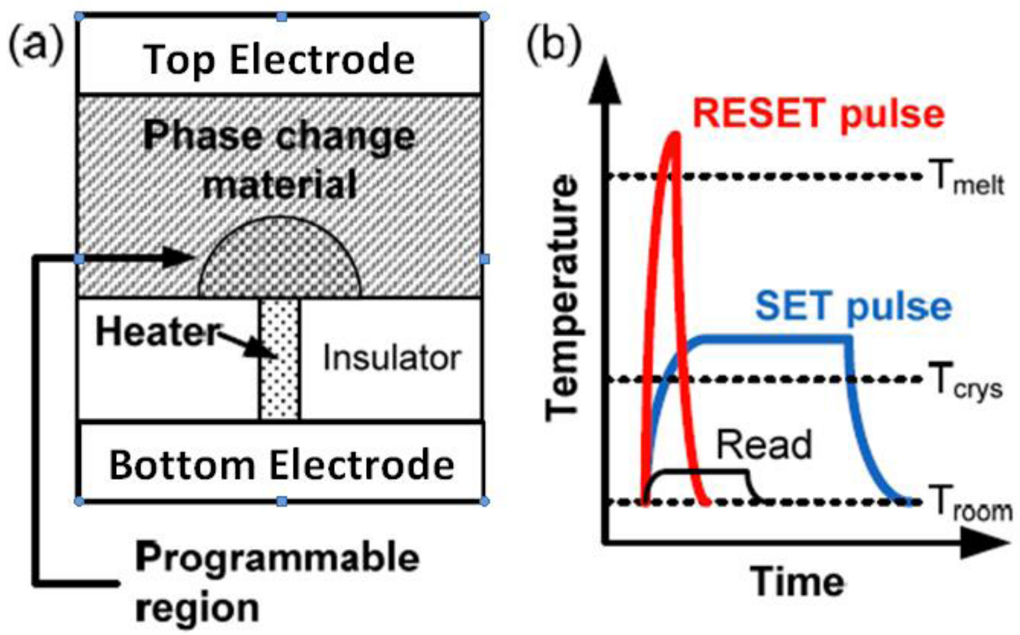
HotOS - XIII Napa, California May 9-11, 2011

Operating System Implications of Fast, Cheap, Non-Volatile Memory

Katelin Bailey, Luis Ceze, Steven D. Gribble, Henry M. Levy
Department of Computer Science
University of Washington

NVRAM: a disruptive technology





- Viable NVRAM is just around the corner:
  - phase change memory
  - memristors
- Persistent memory could be disruptive to OS
  - main memory
  - persistent storage
- This is a great opportunity to re-imagine OS design

*photos from www.gizmodo.com and Assessment of the Potential & Maturity of Selected Emerging Research Memory Technologies Workshop & ERD/ERM Working Group Meeting

A reminder about the world we're living in...

Fast
Dense, cheap
Byte addressable
Volatile
High refresh power

Slow
Denser per bit
Sector-oriented
Persistent
Lower power

Wouldn't you like to have the best of both worlds?

DRAM	Disk
Fast	Slow
Dense, cheap	Denser per bit
Byte addressable	Sector-oriented
Volatile	Persistent
High refresh power	Lower power

Memory technologies that bridge the gap

Phase-Change Memory
Memristors
FeRAM
Spin-transfer Torque MRAM

volatility ← DRAM —— Disk → persistence

DRAM	Disk
Fast	Slow
Dense, cheap	Denser per bit
Byte addressable	Sector-oriented
Volatile	**Persistent**
High refresh power	**Lower power**

Memory technologies that bridge the gap

NVM or NVRAM

Non-volatile memory

(a general term for these technologies)

Fast

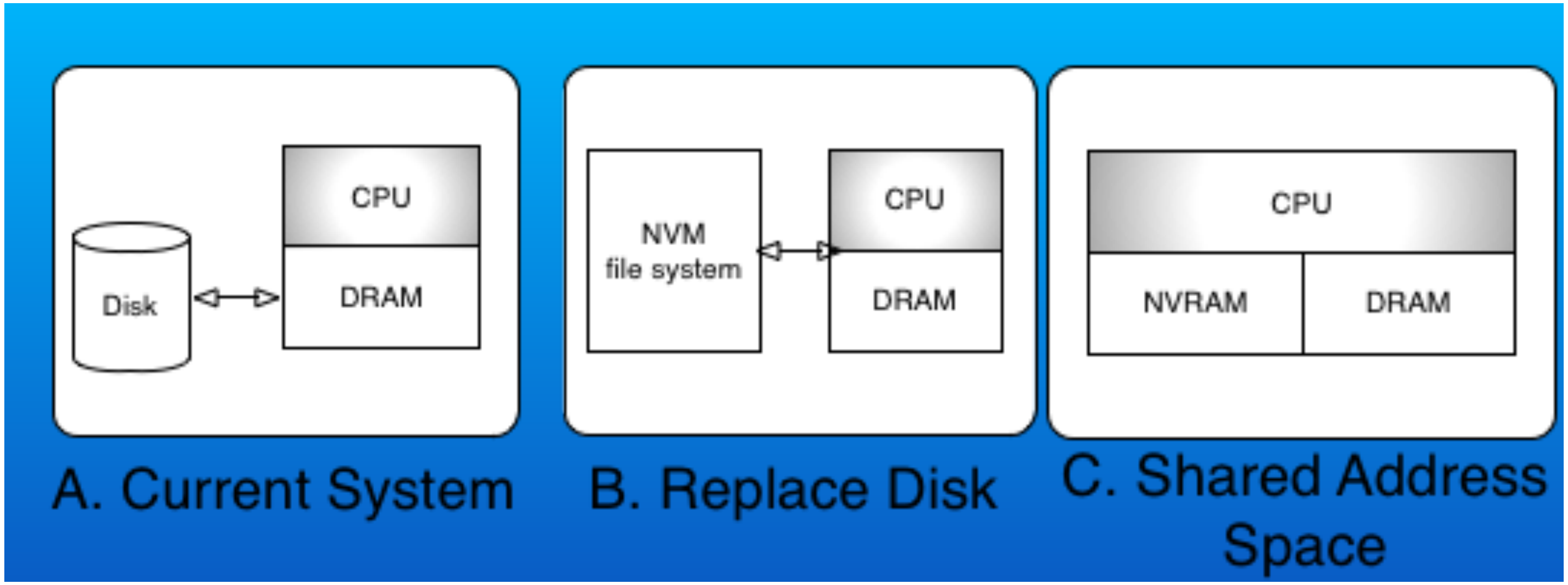
Dense, cheap

Byte addressable

Persistent

Lower power

Alternatives for hardware technology



Related work

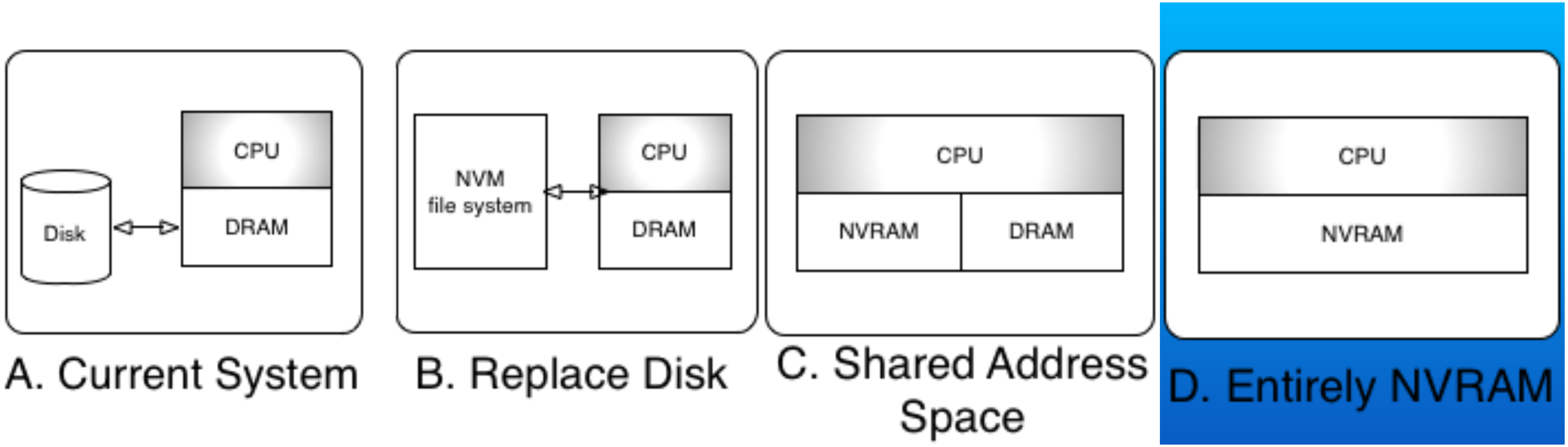
- **Filesystems**: maintains current semantics on NVM with smoother NVM implementation [Condit 2009]

- **Closing the gap**: architecture solution to write endurance and latency: NVM for main memory [Qureshi et al.]

- **Persistent data structures**: with limited semantics for application access [NVheaps, Mnemosyne, and Moneta]

While these approaches are essentially evolutionary, we'd like to look for something a bit more revolutionary.

Alternatives for hardware technology



- **We're discussing only this last option: entirely NVRAM**
- The most extreme option offers the widest array of research and design opportunities

Long-term opportunities in OS design

New technology can serve as a jumping point for new OS research opportunities

- Processes could be eternally long lived
    - ...but how do we reinitialize structures or state?
- Systems could keep very long logs of execution
    - ....but what to forget? when?
- Reduce the cost of startup, reboot, and hibernation.

- This talk shows you some (of a great many) examples in filesystems, virtual memory, processes. More in the paper.

Reliability built on persistence

Every piece of data is now persistent, increasing reliability.

Every piece of data in the stack and heap is persistent whether consistent or not.

- But what if there are bugs?
- Where is it safe to roll back to?
- How does corruption show up?
- What are the consistency guarantees?

Security benefits and concerns

Any data in the stack, heap, or elsewhere is durable.

Sensitive data is therefore also durable and must be explicitly removed.

- How is data scrubbed or verified?
- How many places do we rely on volatility to clean up for us?
- How do we deal with things like the cold boot attack?

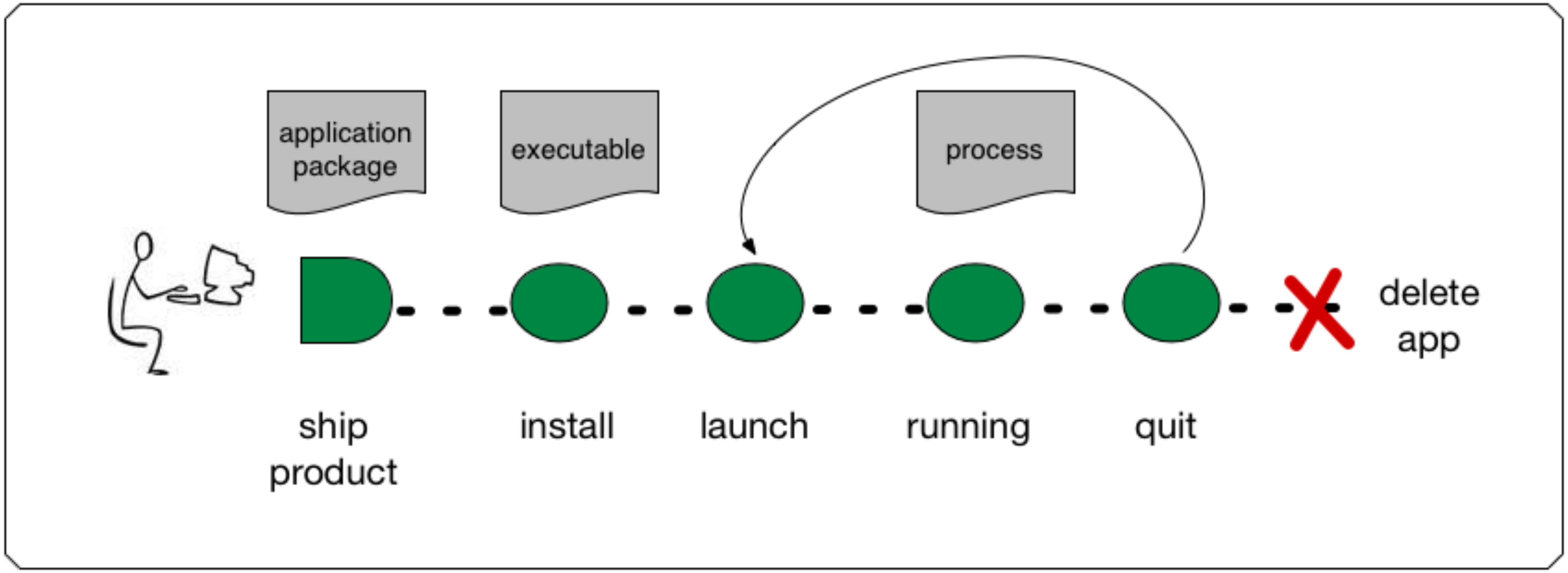
Virtual memory, naming, and addressing

Virtual memory is a bridge between main memory and storage. With a single level, this bridge is no longer needed.

Swapping is clearly not needed, but we still need naming and protection.

- Does page granularity still make sense?
- In a single-level system what protection systems make sense? What naming system?
- What does the address space look like?

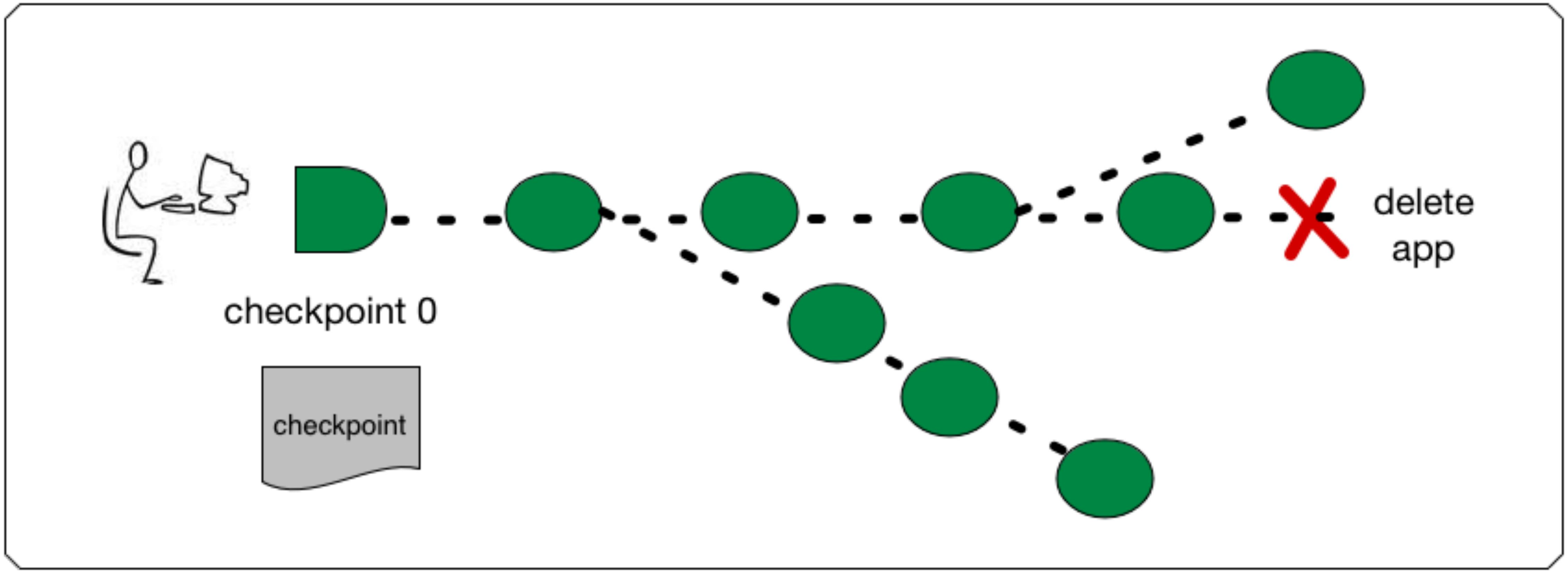
Current process models...



Applications today have a number of distinct, well-defined states.

- Is there really a need for so many different states?
- Why do we have 3 different process formats?

... New process models



At every moment in time a process is persistent: a sequence of checkpoints.

- How do we handle faults? updates?
- What does a "reboot" imply?

Conclusions

- Non-volatile main memory affects nearly every area of OS design:

  virtual memory, filesystems, processes, reliability, security, etc.

- Many questions to be answered
- Many opportunities for new designs
- We don’t know what the “right” answer is

Questions?

Contact at katelin@cs.washington.edu
http://www.cs.washington.edu/homes/katelin