Virtual Machine Time Travel Using Continuous Data Protection and Checkpointing
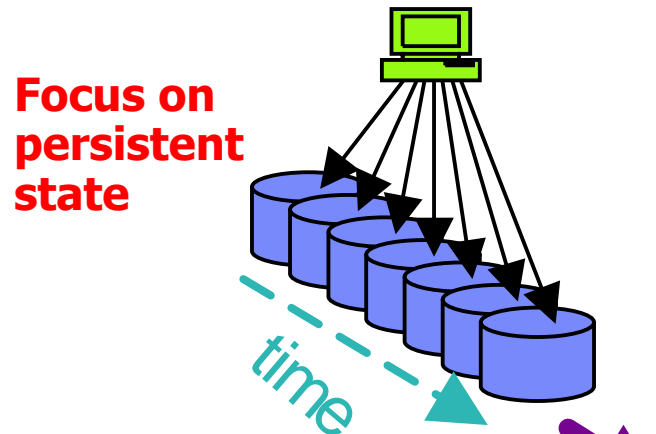
Paula Ta-Shma

Joint work with Guy Laden, Muli Ben-Yehuda and Michael Factor

CDP with Virtual Machine Checkpointing

- **CDP at storage level**

  - Enables restoration of persistent state to past points in time

  - Transient state is not captured

  - Typically requires application recovery (if available)

- **VM checkpointing**

  - Allows creation of a snapshot of a transient state at a given point in time

 - Can later be restored to continue running from that point

Focus on persistent state

time

Focus on transient state

time
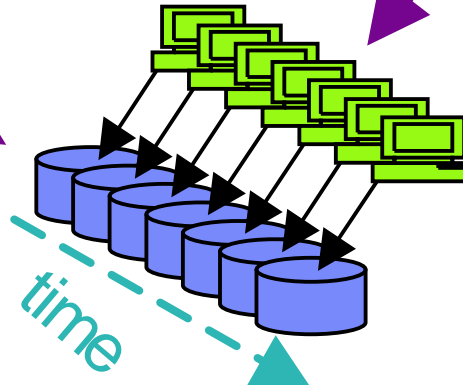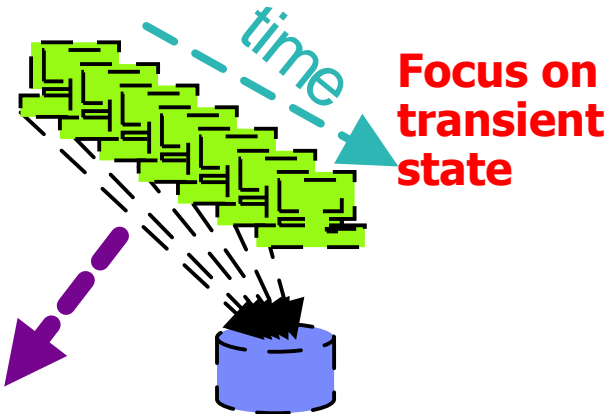
Combination

- Store frequent coordinated checkpoints of entire system image

time

Entire system state

Applications

- ## Improving Availability
  - Quickly recover from serious corruptions of a VM e.g. administration errors or viruses

- ## Forensics
  - Root cause analysis without affecting the production system

- ## System Administration
  - Fast testing of upgrades, patches, changes in configuration

- ## Boot Speedup
  - Checkpoint taken immediately after boot could be reverted to

Forensics

Analyze why a system failed while repairing the system
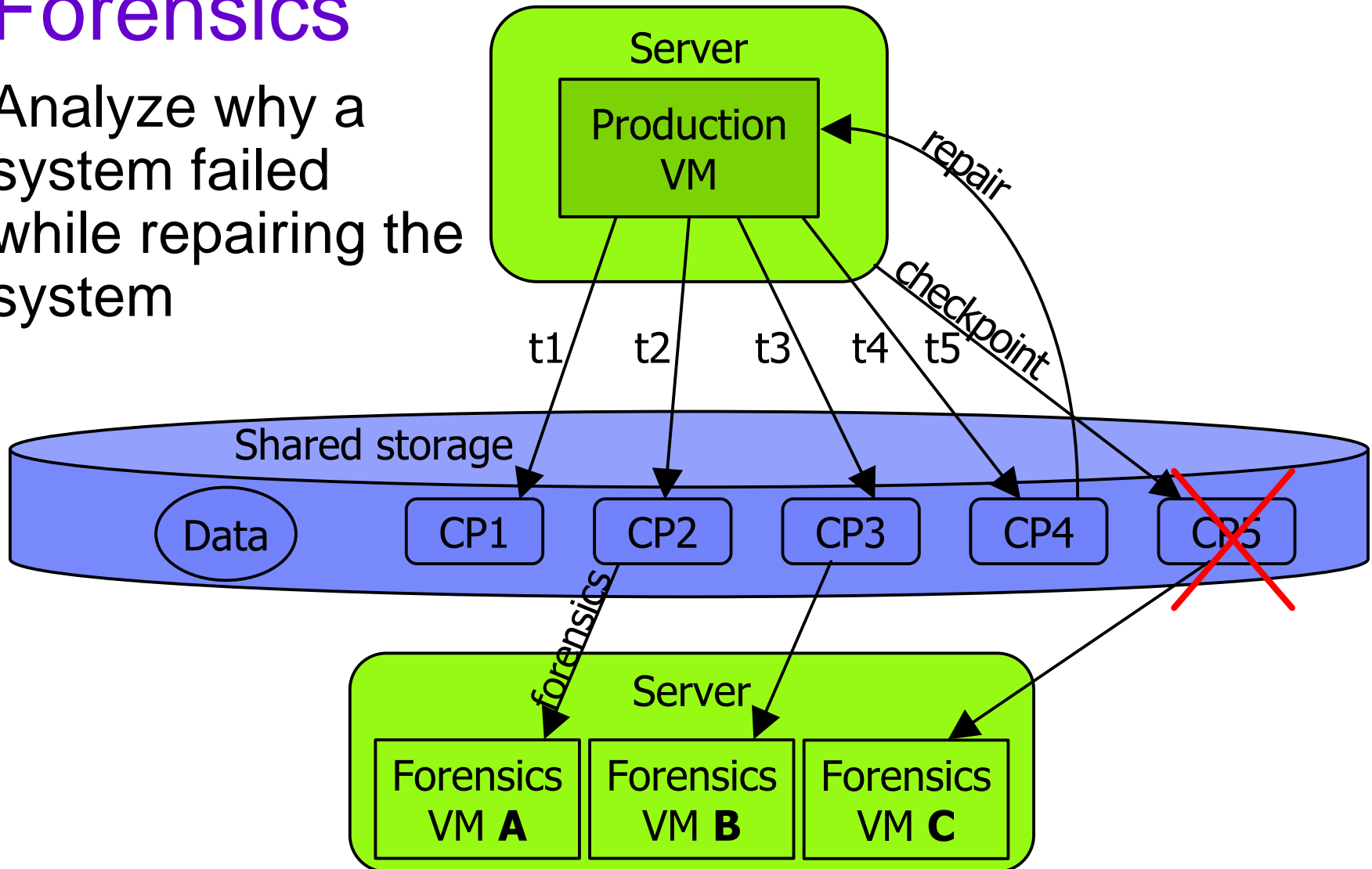
Talk Overview

- Storage Subsystem

- Checkpointer

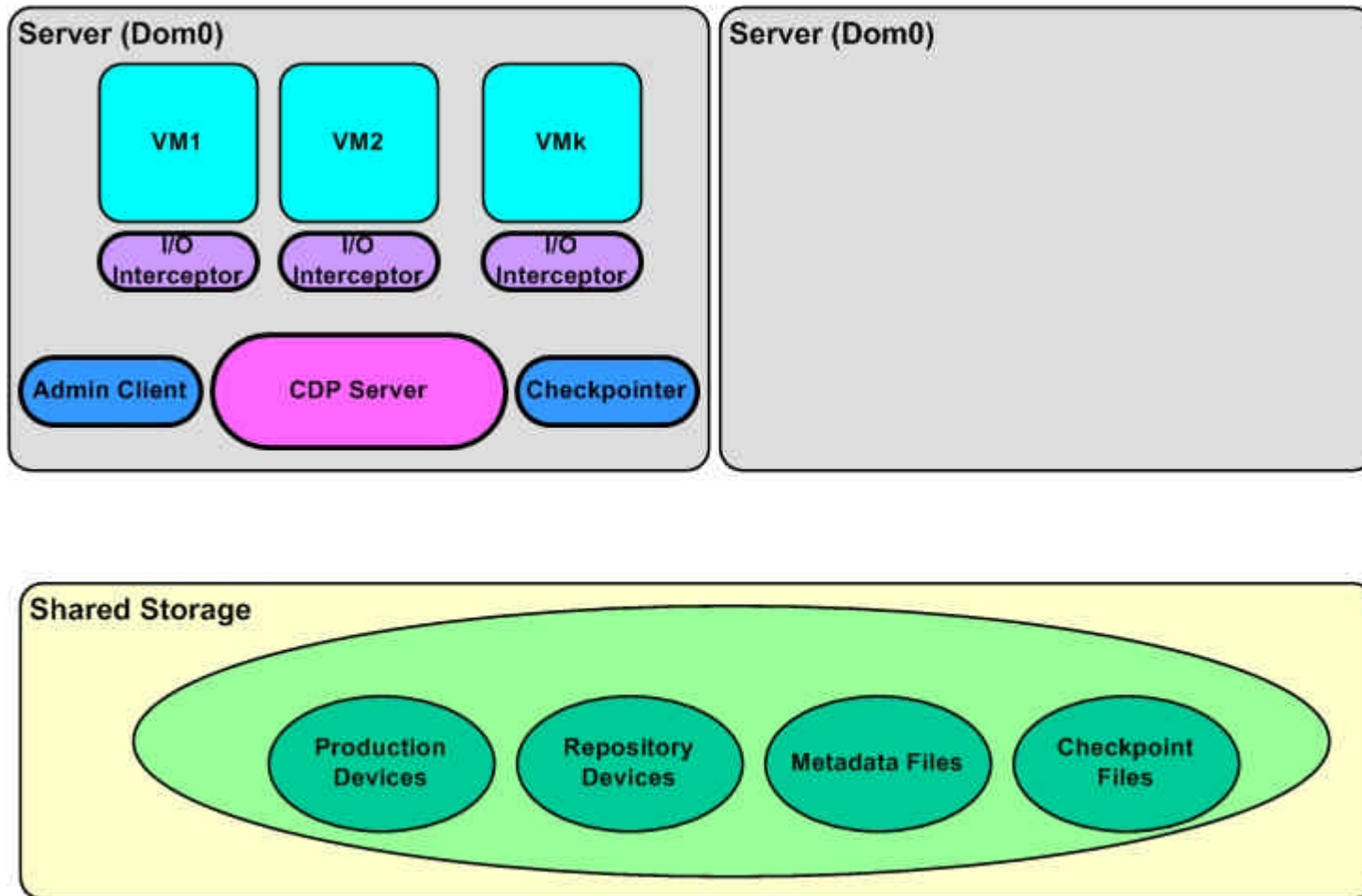- Integration

Storage Subsystem : Approach

- CDP as a system level service for VMs

- Hypervisor-based CDP
  - Generic, portable (storage hardware agnostic)
  - Cost effective
  - Inherits advantages of hypervisor approach
    - Applicable to all OS'es supported by hypervisor
    - Install once per physical (not virtual) machine
  - Integrates well with platform

Storage Subsystem: Requirements

- Fine granularity protection

- Fast and reversible revert

- Clones

- Space Management
  - CDP window and space reclamation
  - Write sharing vs. copying

- Performance
  - Minimal impact in normal (non revert) case

Storage System Architecture

Paula Ta-Shma | **Virtual Machine Time Travel Using CDP and Checkpointing** | SYSTOR 2007 © 2007 IBM Corporation
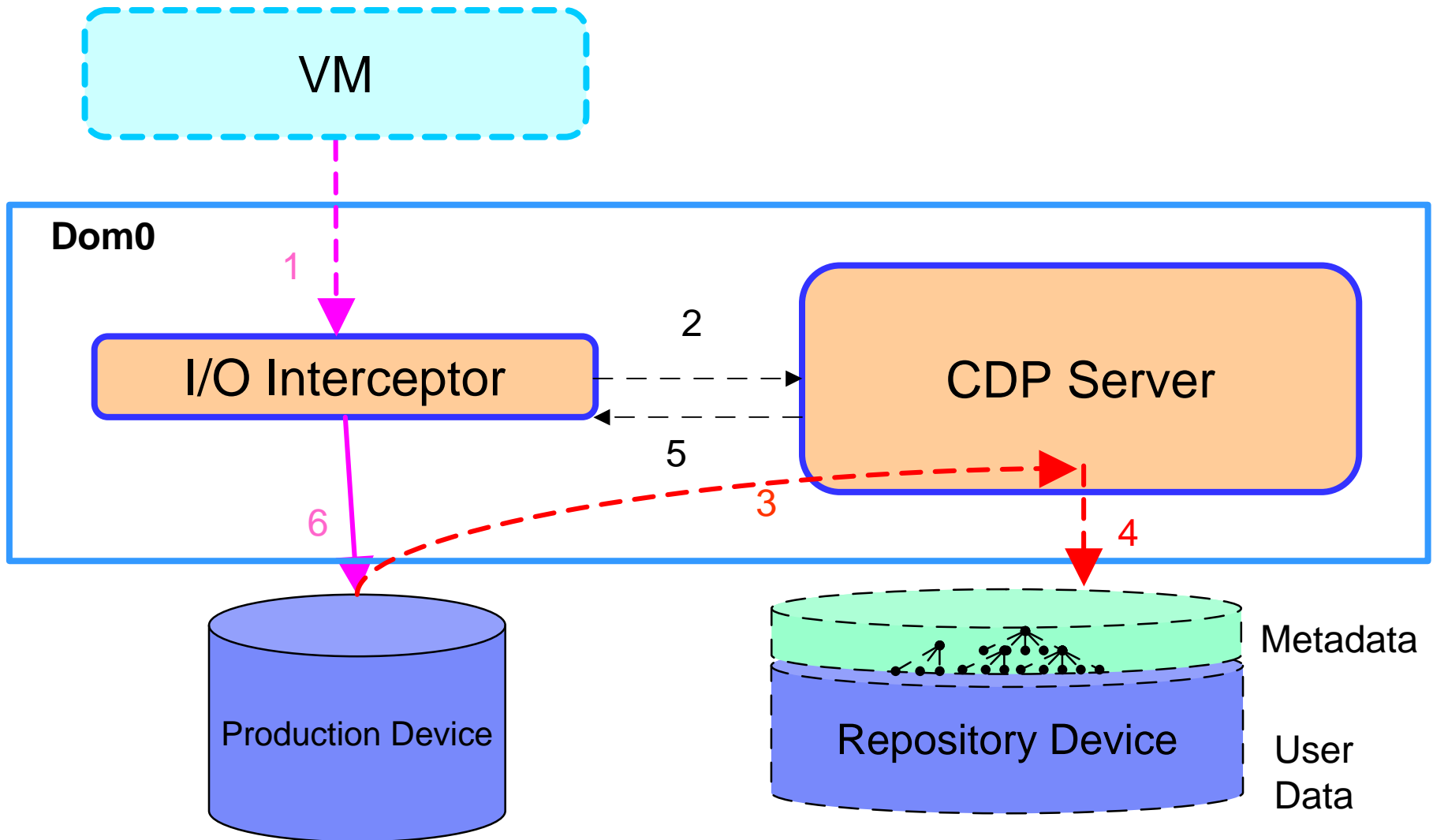
Storage System Components

- ## I/O Interceptor
  - User space driver using Xen blktap framework
  - Communicates with server via IPC

- ## Cdp admin client
  - Communicates with server via HTTP
  - Create/Destroy, Mark Event, Clone, Revert

- ## CDP Server
  - Uses database library to implement metadata indexes

CDP Architecture :
When to Copy a Write ?

- ## Never

  - ### Logging

- ## On receipt by controller

  - ### SplitStream

- ## Before overwrite (COW)
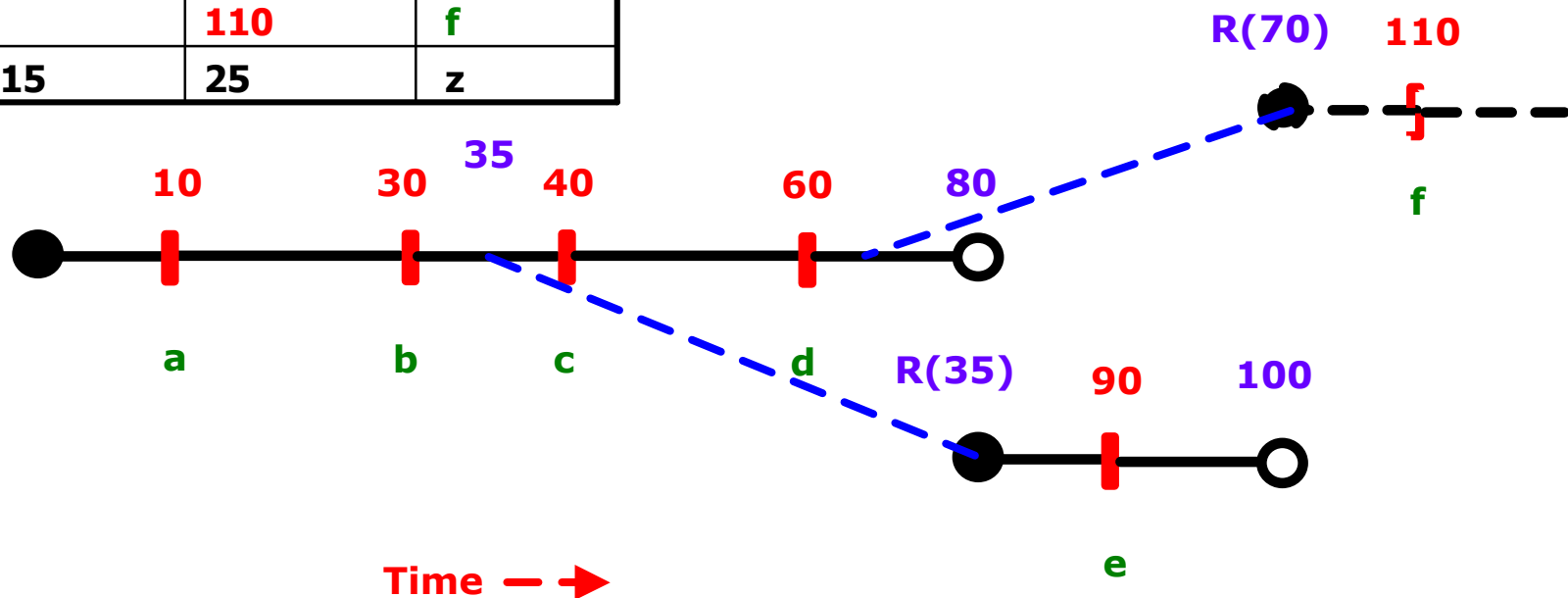
  - ### Checkpointing

Checkpointing – Write Flow

VM

Dom0

1

I/O Interceptor

2

CDP Server

5

6

3

4

Production Device

Repository Device

Metadata

User
Data

Metadata Indexes

LPMap

lba (key)	time (key)	physical address
12	10	a
	30	b
	40	c
	60	d
	90	e
	110	f
15	25	z

Branch Table

id	start time	end time	revert-to time
1	0	80	-
2	80	100	35
3	100	-	70

Checkpointer

- Xen's save/restore are disruptive

- No checkpointing support was available at the time of our work

- Xen supports live migration of VMs between physical machines

  - VM continues running while most of state is transferred

- Our approach is based on localhost live migration

  - Implemented a migration proxy

  - Proxy intercepts migration bitstream and saves it to disk as a checkpoint file

Integration

- Towards end of live migration, VM is paused momentarily
- At this point in time, checkpointer invokes a user-defined callback
- Our callback
  - Marks a CDP event using CDP admin interface
  - Associates this event with the appropriate checkpoint file

Conclusions

- **Implemented VM time travel as a system level service**
  - Xen checkpointer
  - CDP Server
  - Coordination

- **Useful for applications such as**
  - Availability, forensics, sys admin etc.

- **Time 4 lunch** ☺