ADAPTIVE LIVE VM MIGRATION OVER A WAN

MODELING AND IMPLEMENTATION

Weida Zhang, King Tin Lam, Cho-Li Wang
Department of Computer Science, The University of Hong Kong

Live Migration of VMs

- Live migration: the VM is lively on the move
  - Dynamic resource provisioning within a data center
  - An enablement of cloud technology
  - Enhancing IT's efficiency and cost-effectiveness.

Wide-area Live Migration (LM) !?

- WAN App Scenarios:
  - Facilitate business operations:
    - Recent report: Instagram migrated user photos from Amazon EC2 to Facebook VPC*

    * How Facebook Moved 20 Billion Instagram Photos Without You Noticing
  - Mobile working env.: a virtual workplace migrating from your home desktop PC to your smartphone, and then to your office workstation, and vice versa (OT!).
  - Cloud federation: move VMs from vendor to vendor
  - Global job scheduling: move the VM around the world

Overview

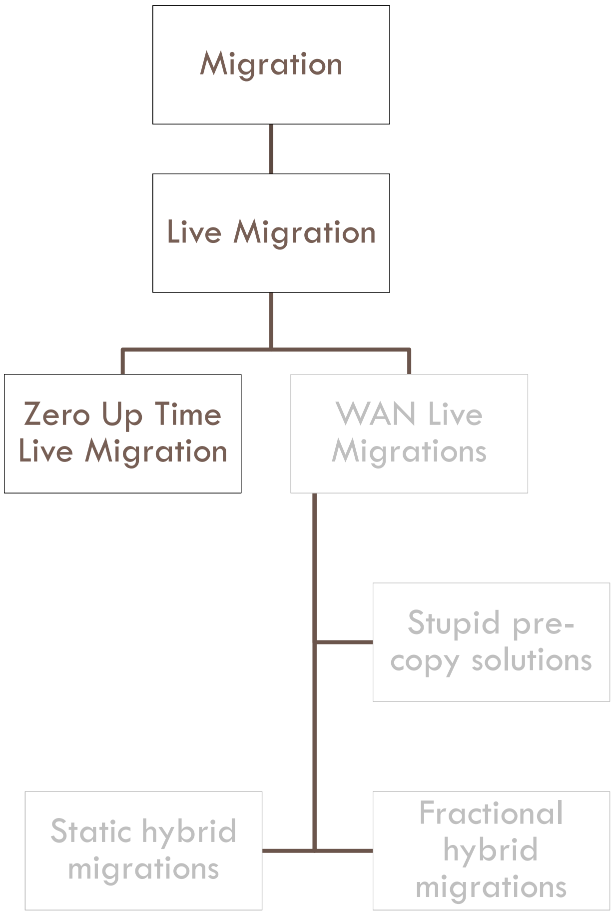
- Introduction
  - Related Work
  - Problem Description
- Methodology
  - Our Invention: A Fractional Hybrid-copy LM Framework
  - Methodology Overview
  - Profiling, Modeling & Simulation
  - Recursion
  - Implementation
- Experiments & Results

Introduction

Related Work

Problem Definition

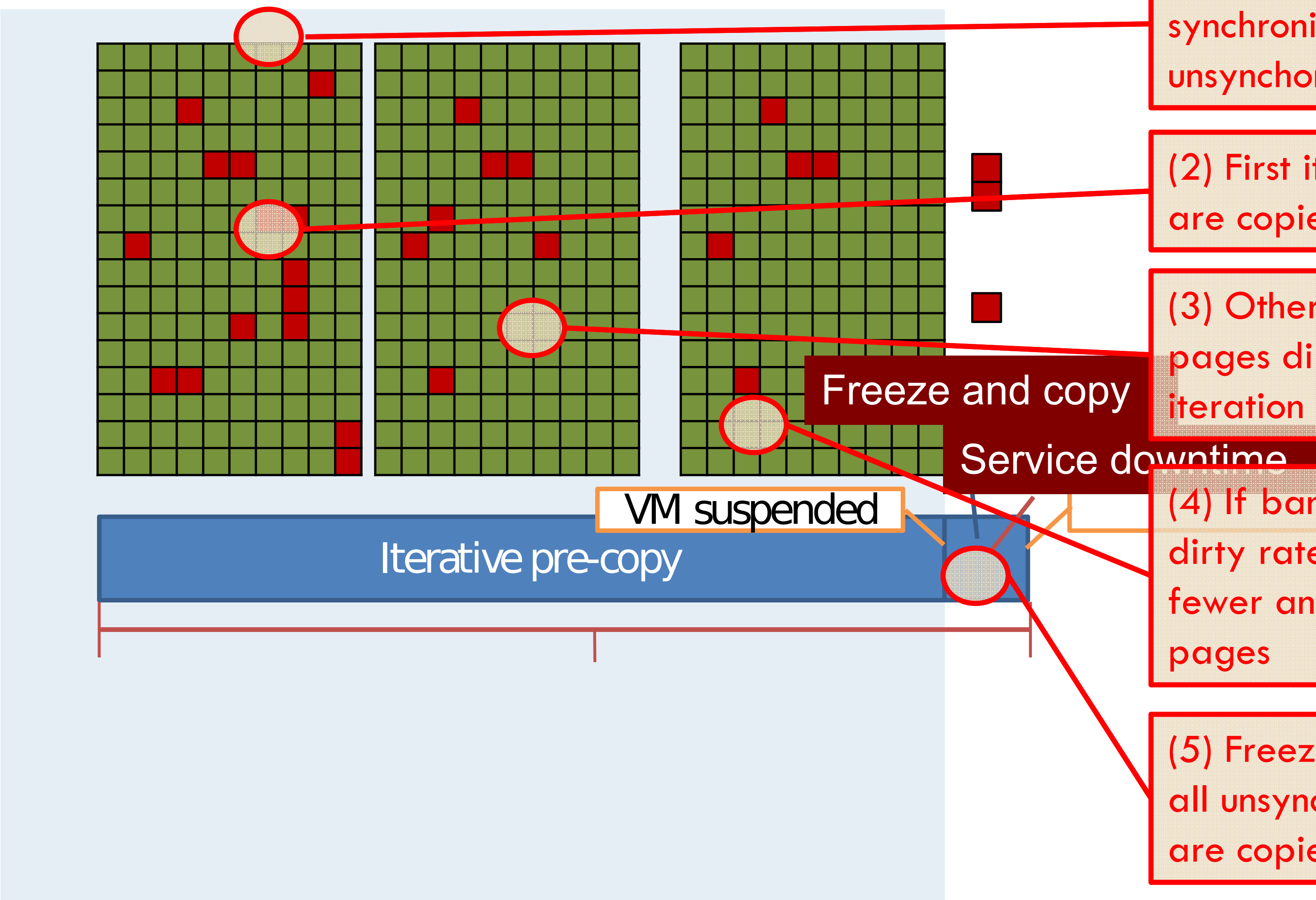
Existing Work on Live Migration



- Pre-copy [2,3]
  - small downtime
- Post-copy [4]
  - zero downtime
  - performance penalty

[2] Nelson, USENIX’ 05
[3] Clark, NSDI’ 05
[4] Hines, SIGOPS’ 09

Pre-copy Algorithms



(1) Green means synchronized; red means unsynchonrized

(2) First iteration, all pages are copied

(3) Other iterations, only the pages dirtied in the previous iteration are copied

(4) If bandwidth > page dirty rate, there will be fewer and fewer dirty pages

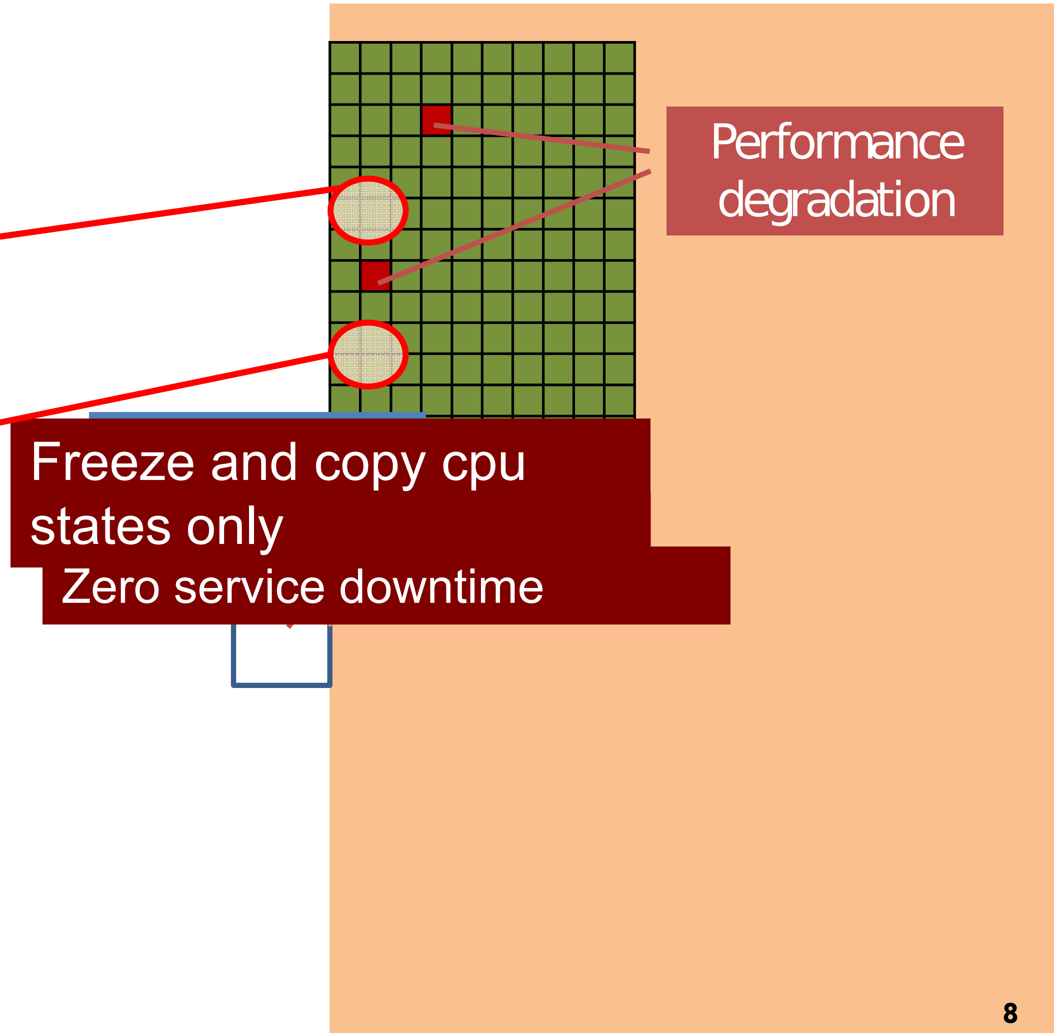
(5) Freeze-and-copy phase, all unsynchronized pages are copied

Post-copy Algorithms

(1) Resumes the VM in the destination immediately

(2) Background transferred pages turn green

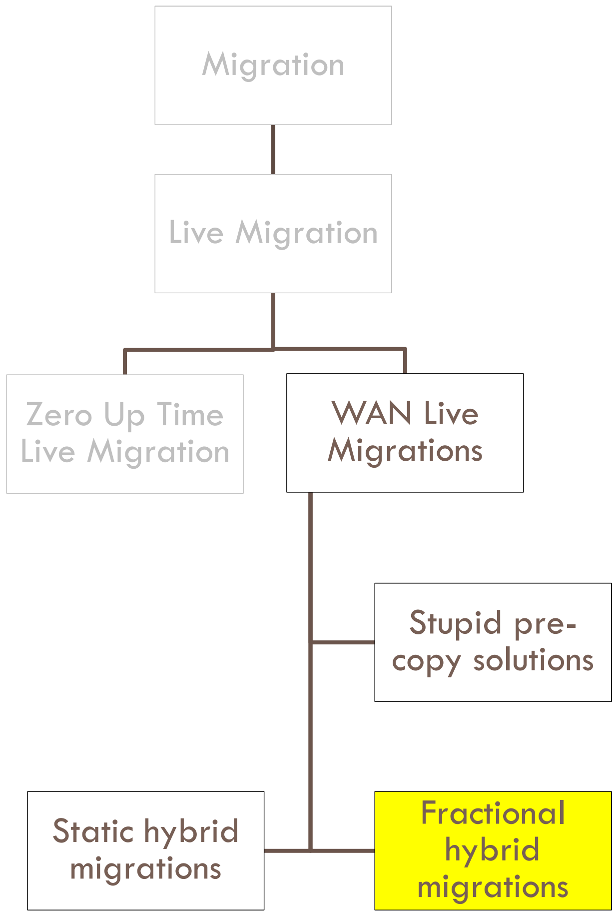
(3) On-demand requested pages introduce performance penalties



Problem

- Problem: pure pre-copy and post-copy are not doing well on a WAN.
- Hybrid: tradeoff between downtime and performance penalty

Existing Work on Wide-Area LM



- Pre-copy memory & pre-copy storage [7,9]
  - [7] Akoush, MASCOTS' 11
  - [9] Bradford, VEE' 07
- Pre-copy memory & post-copy storage [11,13]
  - [11] Hirofuchi, CCGrid' 10
  - [13] Luo, CLUSTER' 08
- Pre-copy memory & hybrid-copy storage [14] = Pre-copy memory & pre-copy S% of storage
  - [14] Zheng, VEE' 11

Our contribution of a new approach:

- A fractional hybrid-copy = Pre-copy M% memory & pre-copy S% storage
- Adaptive = Fractional + Model to find (M, S)

Methodology

A Fractional Hybrid-copy LM Framework

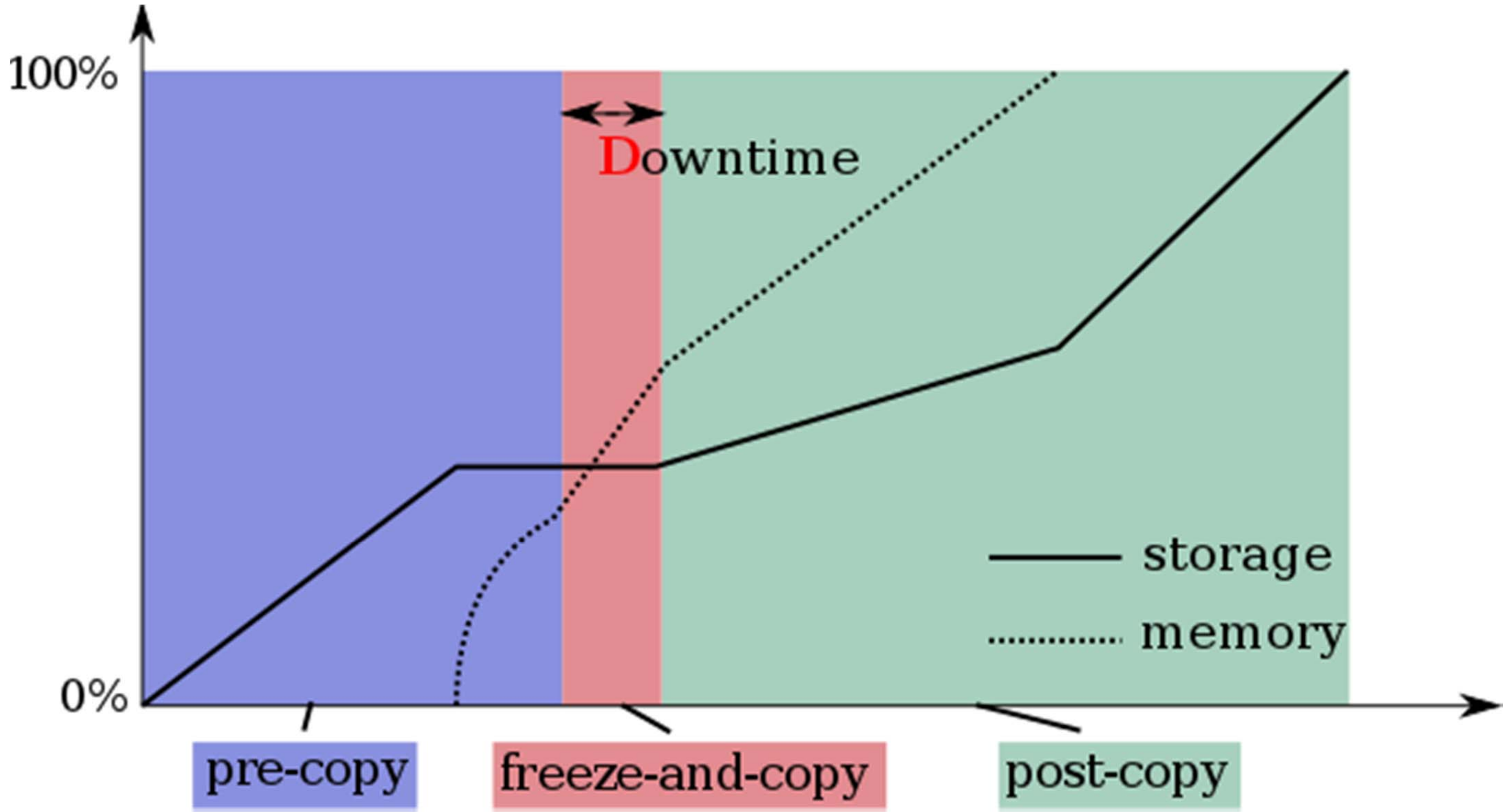
Methodology Overview: An Adaptive Process

Profiling, Modeling and Simulation

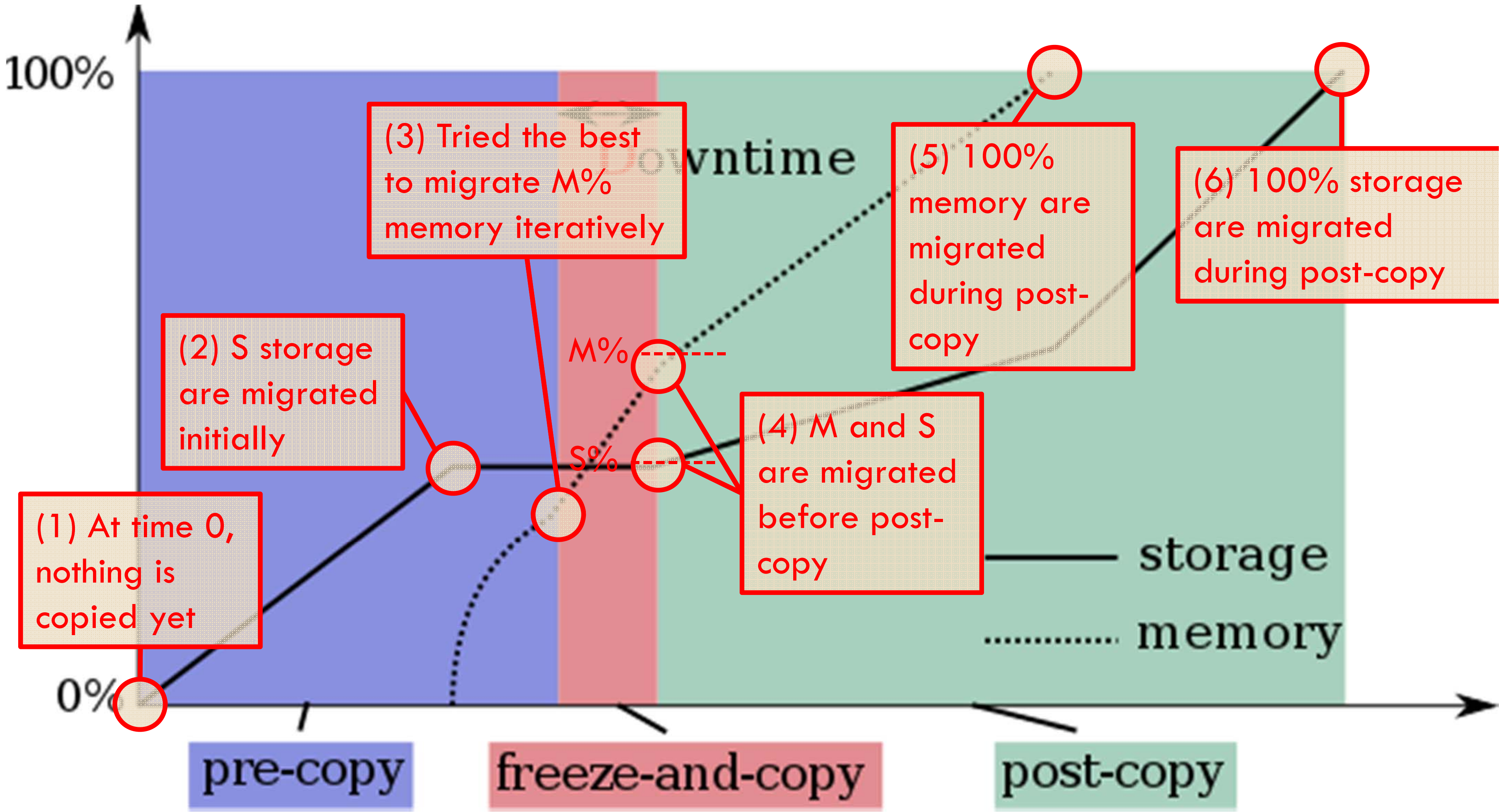
Recursive Searching of (M, S)

Implementation

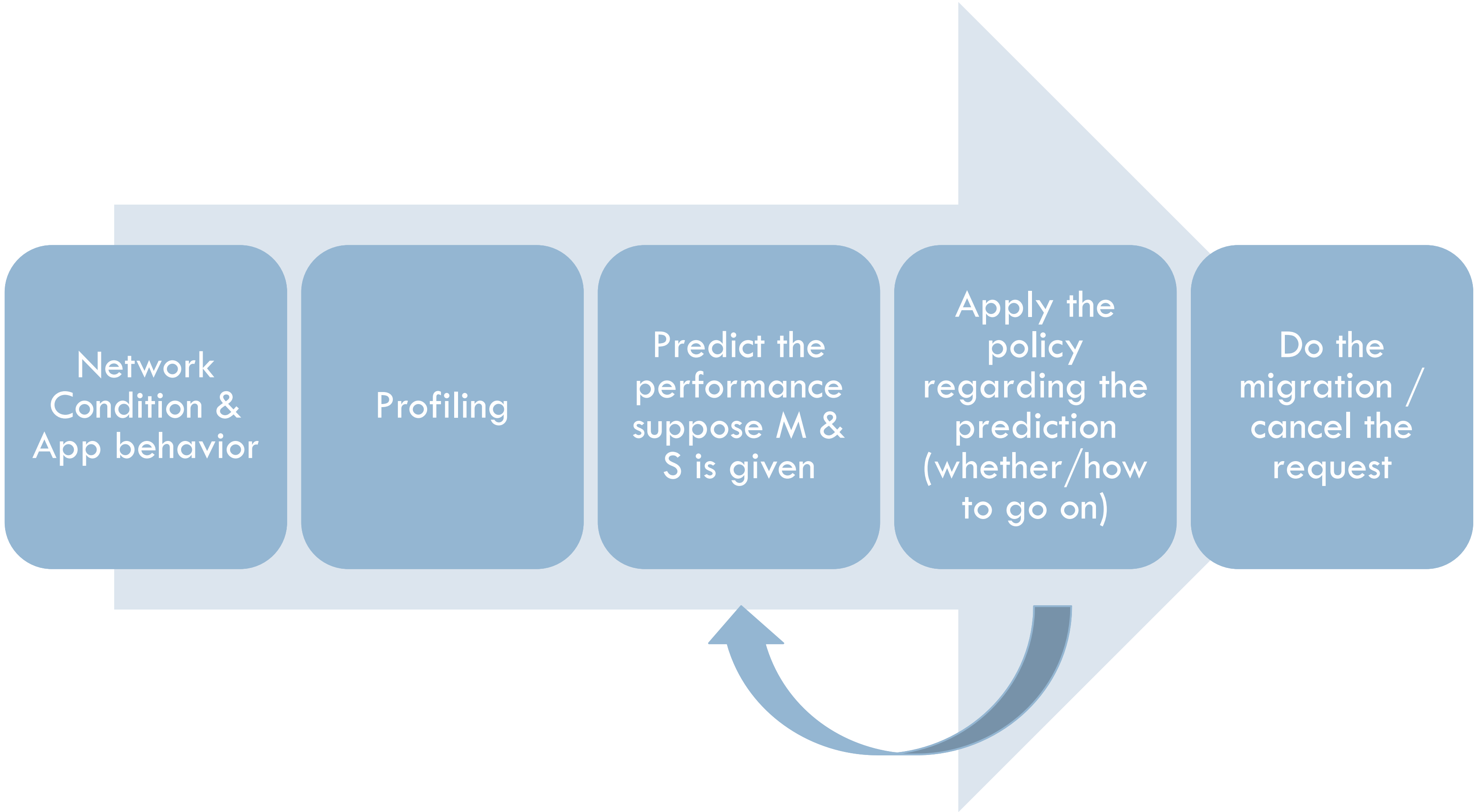
Fractional Hybrid-copy



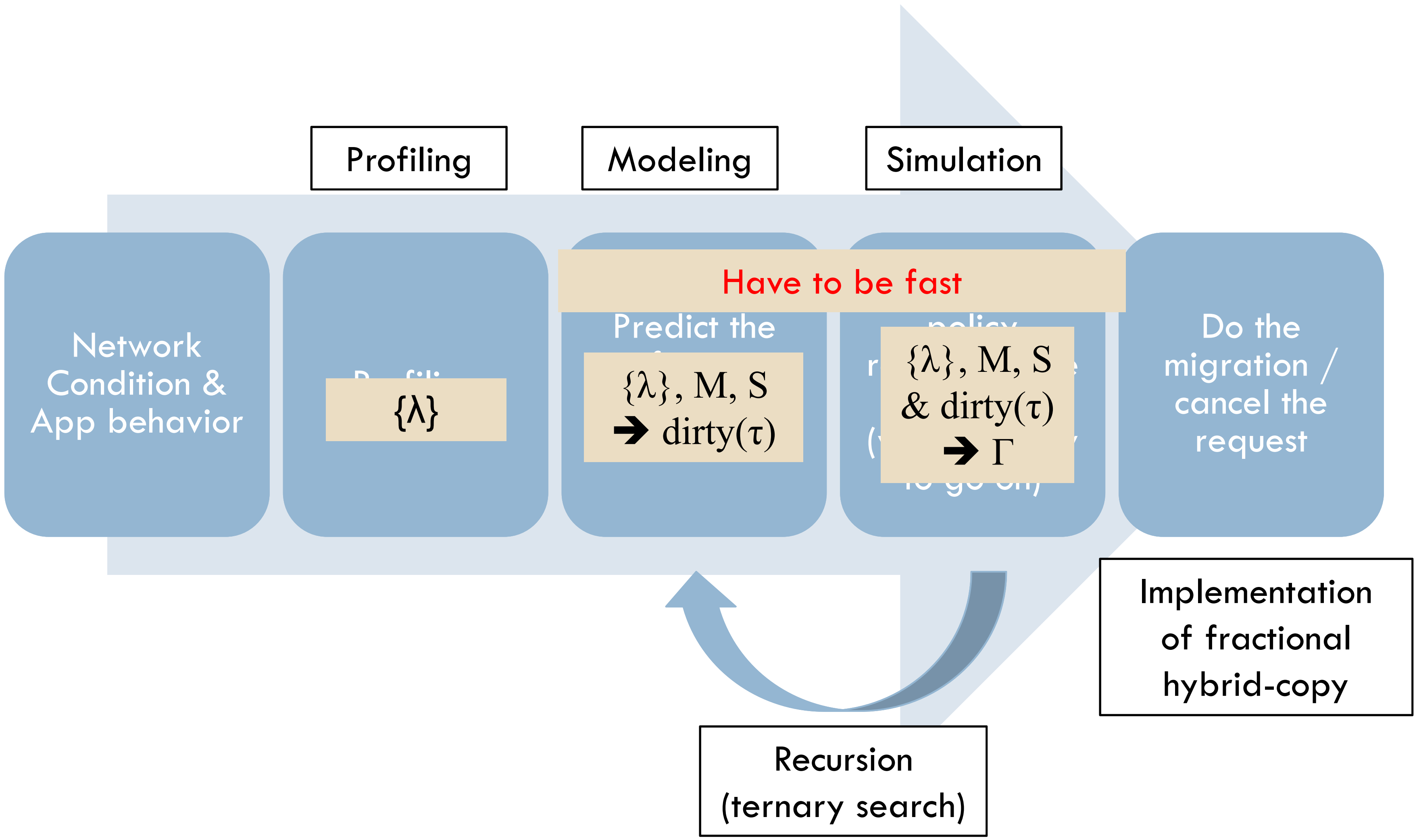
Fractional Hybrid-copy



Methodology Overview



Methodology Overview



Profiling, Modeling & Simulation

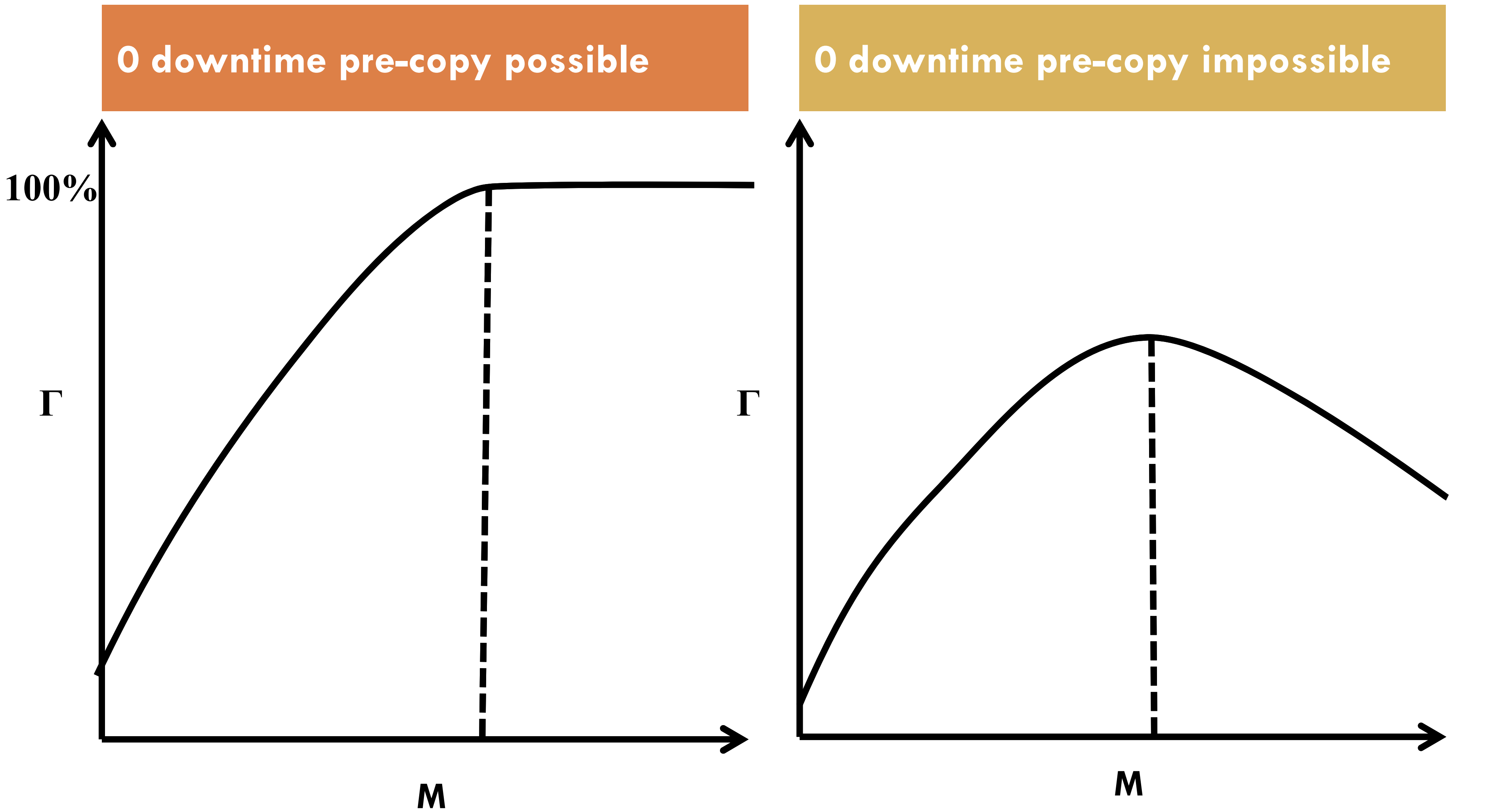
- Key components of simulation: dirtying rate
  - Constant dirtying rate [10]
    - Simple profiling: count how many pages are updated
    - O(1) simulation
  - Full-history profile + replay-based dirtying rate [10]
    - Heavy profiling overhead: record every update of a page
    - O(N), N is the size of memory or storage
  - Assuming Poisson distribution
    - Reduced overhead: how many times a page is updated
    - n samples, one λ for each page/trunk
    - O(n)

[10] Akoush, MASCOTS' 10

Profiling, Modeling & Simulation

- ***Performance restoration agility, Γ*** ⟵ **Our proposed new metric**
  - Γ is the variable to be optimized
  - Γ is a function of profile $\{\lambda\}$, M, S, D
  - $\Gamma = \delta T / (D + \Delta T)$
    - δT: a configurable time, we use 20 seconds
    - ΔT: time needed for the VM at restore to execute the workload of δT during normal execution
  - $\Gamma = 1 / (D * weight_1 + \text{Penalty} * weight_2)$
    - you can use different policies to balance downtime and penalty, i.e. balance between pre-copy and post-copy

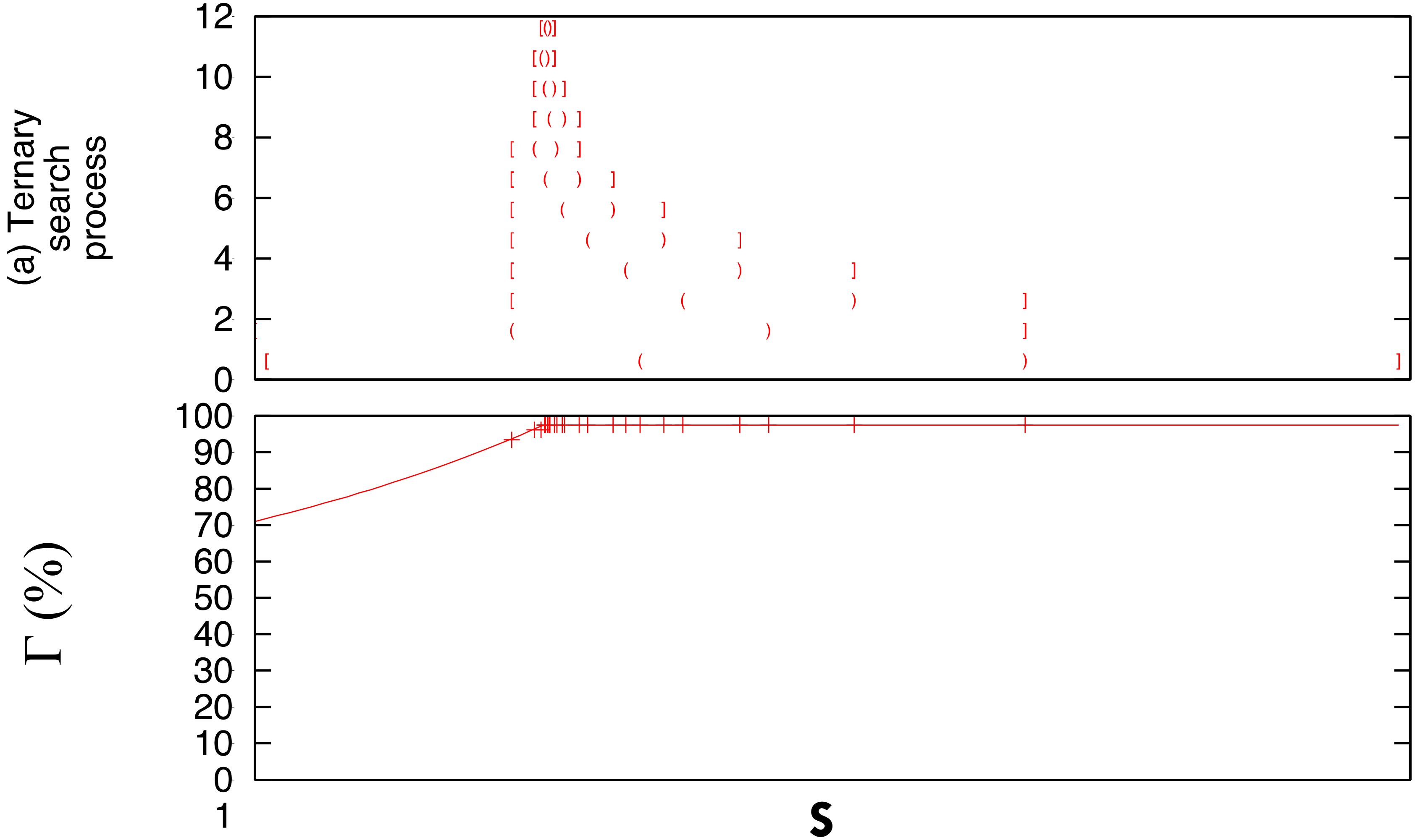
1-dimension View of Γ



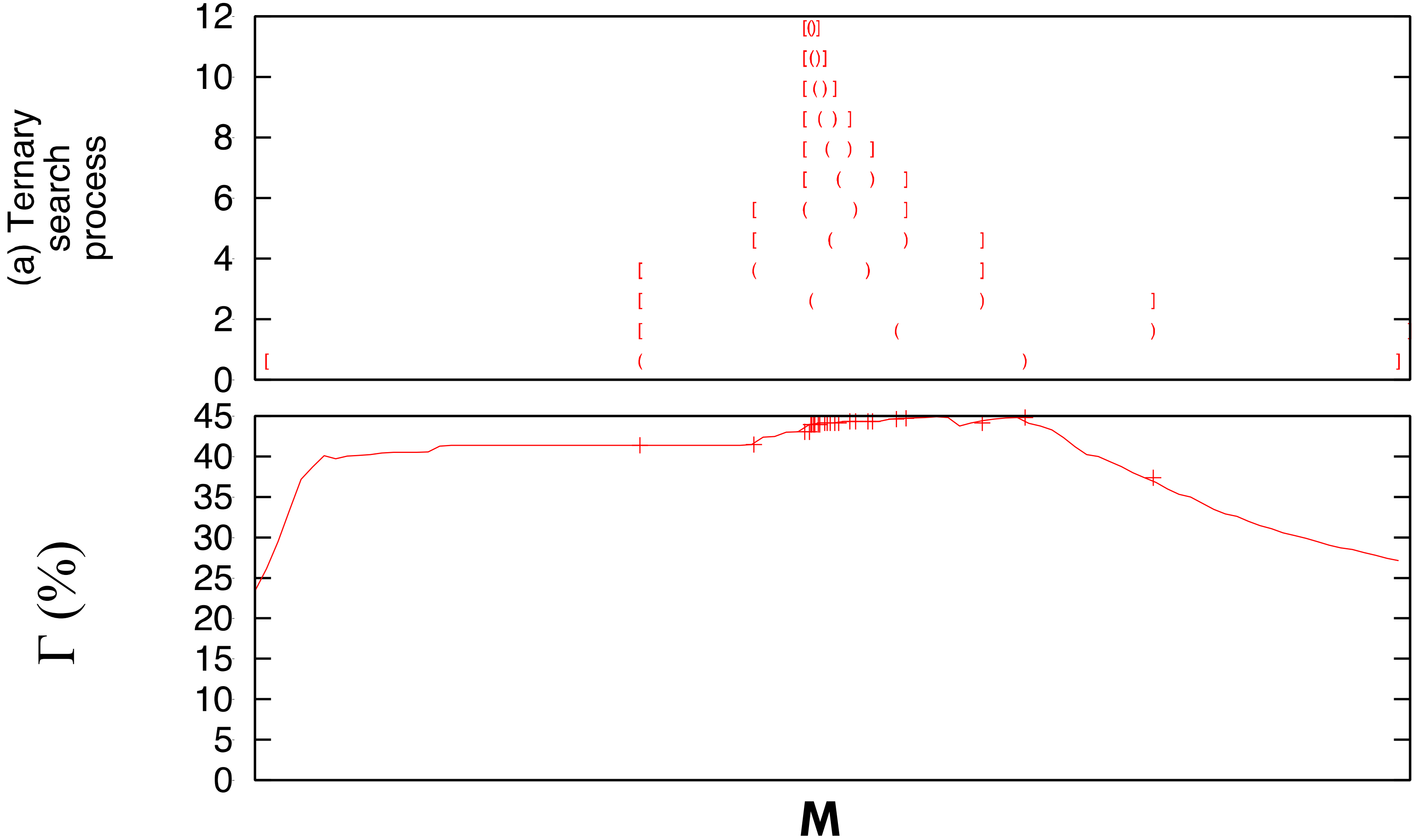
Recursion: Searching for M & S

1. Assume the M is magically instant-copied (greedy)
   - Find S using Ternary Search
   - Assume the S could be live pre-copied, i.e. 0-down time pre-copy possible
   - If the migration of storage-only cannot be live, there is no way to do the live migration
2. Fix the found S
   - Find M using Ternary Search

Ternary Search (Magical M, sysbench)

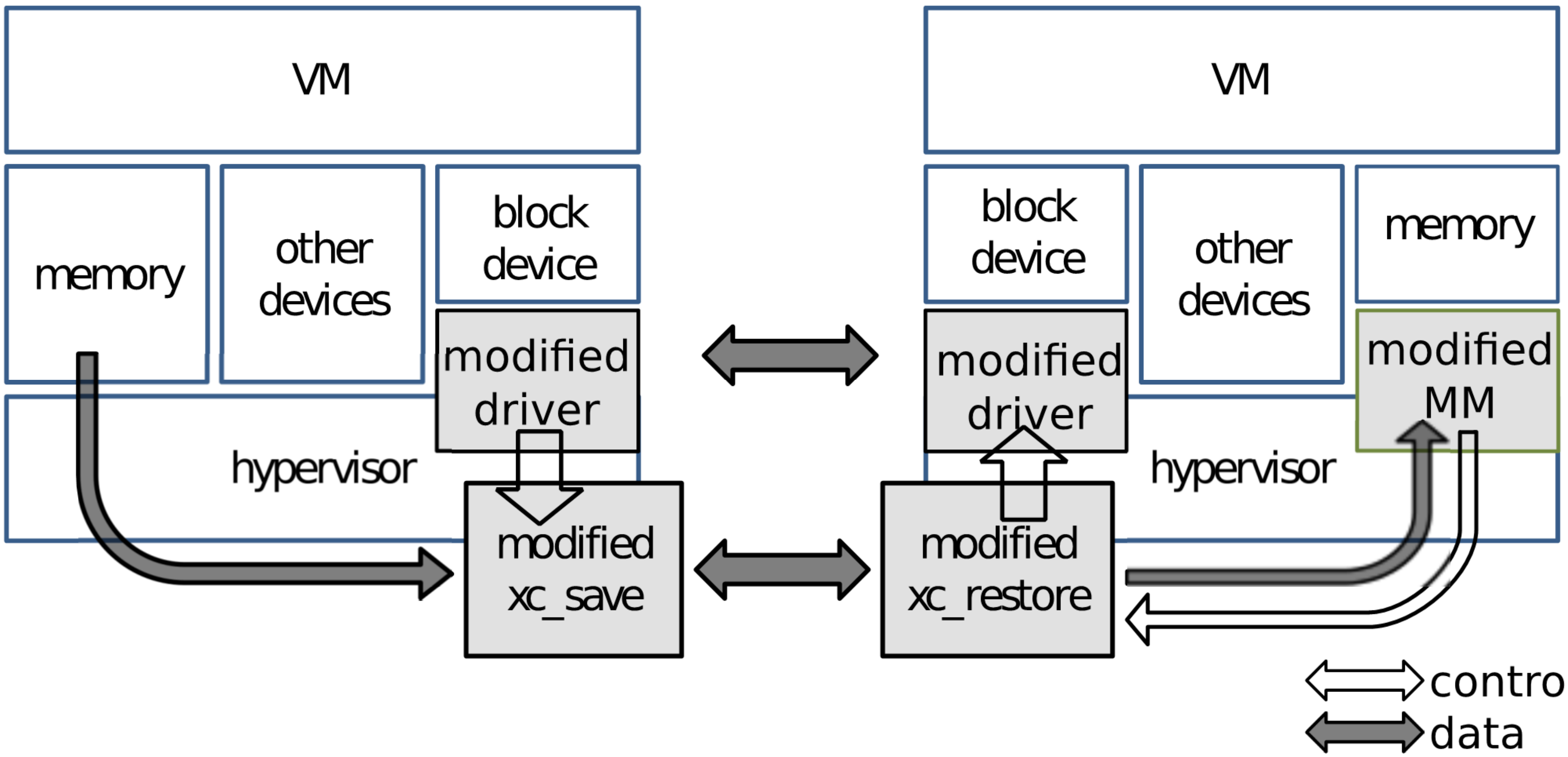


Ternary Search (Fixed S, v8)



Implementation

- Implemented on Xen



Experiments & Results

Experimental Settings

- v8 benchmark (JavaScripts on Google v8 engine)
- Sysbench (intensive read/write operations)
- Move VM from A to B, migration channel separated from application's network channel
- Migration channel: 5ms RTT, 40 Mbps (two ends within a city)

Result 1: Predictabilities (Memory, v8)

TABLE I. OVERALL EVALUATION OF THE MEMORY PREDICTION

Read			Write		
j^* \ j_{actual}	0	1	j^* \ j_{actual}	0	1
0	59.8%	10.1%	0	54.1%	3.5%
1	3.8%	26.3%	1	0.4%	42.0%
accuracy_R	86.1%		accuracy_W	96.1%	

Result 1: Predictabilities (Storage, sysbench)

TABLE II. OVERALL EVALUATION OF THE STORAGE PREDICTION

Read			Write		
j^* \ j_{actual}	0	1	j^* \ j_{actual}	0	1
0	75.1%	0.9%	0	96.6%	3.4%
1	2.2%	21.9%	1	0.0%	0.0%
accuracy_R	96.9%		accuracy_W	96.6%	

Result 1: Predictabilities (Simulation, v8)

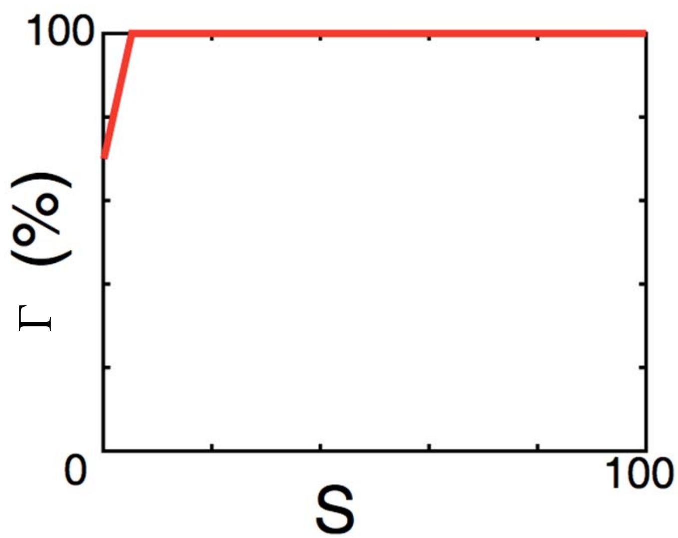
- When (M,S) = (60%,50%)
- $\Gamma = 20\%$

TABLE III. PREDICTION OF T, U AND D

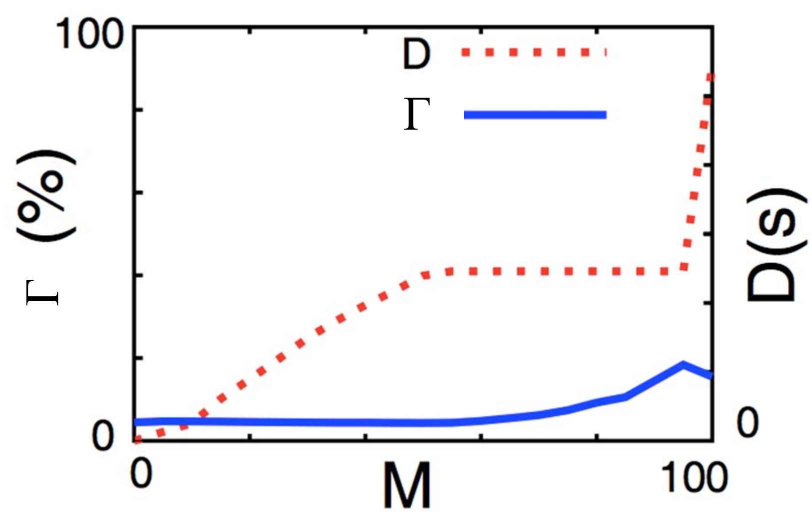
	Predicted (s)	Actual (s)
Total migration time (T)	1063.3	988
Remote uptime (U)	554.3	493
Downtime (D)	49.2	53.7

Result 2: Search of (M, S) (v8)

Searching of S when M is magically copied



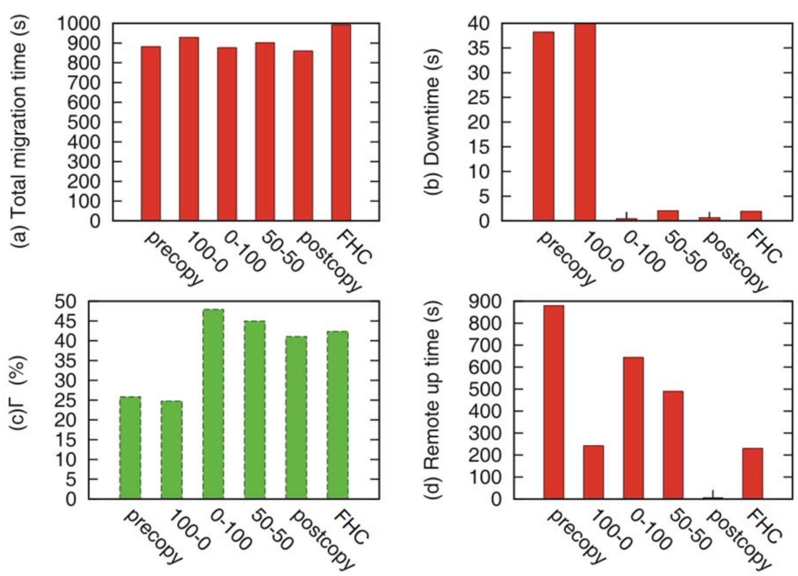
Searching of M when S = 3$
Found M = 95% is the best



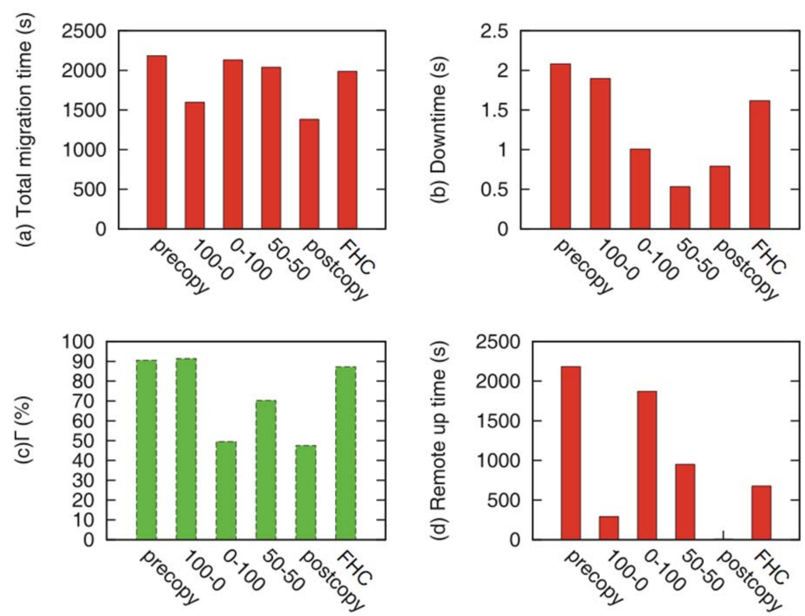
Whole-page overwriting technique:
We found if a whole page-writing (4K) causes a fault during post-copying, it is good to just overwrite the page, without remote fetching the page.

Result 3: Overall Performance

v8 (M, S) = (48%, 0%)



Sysbench = (98%, 25%)



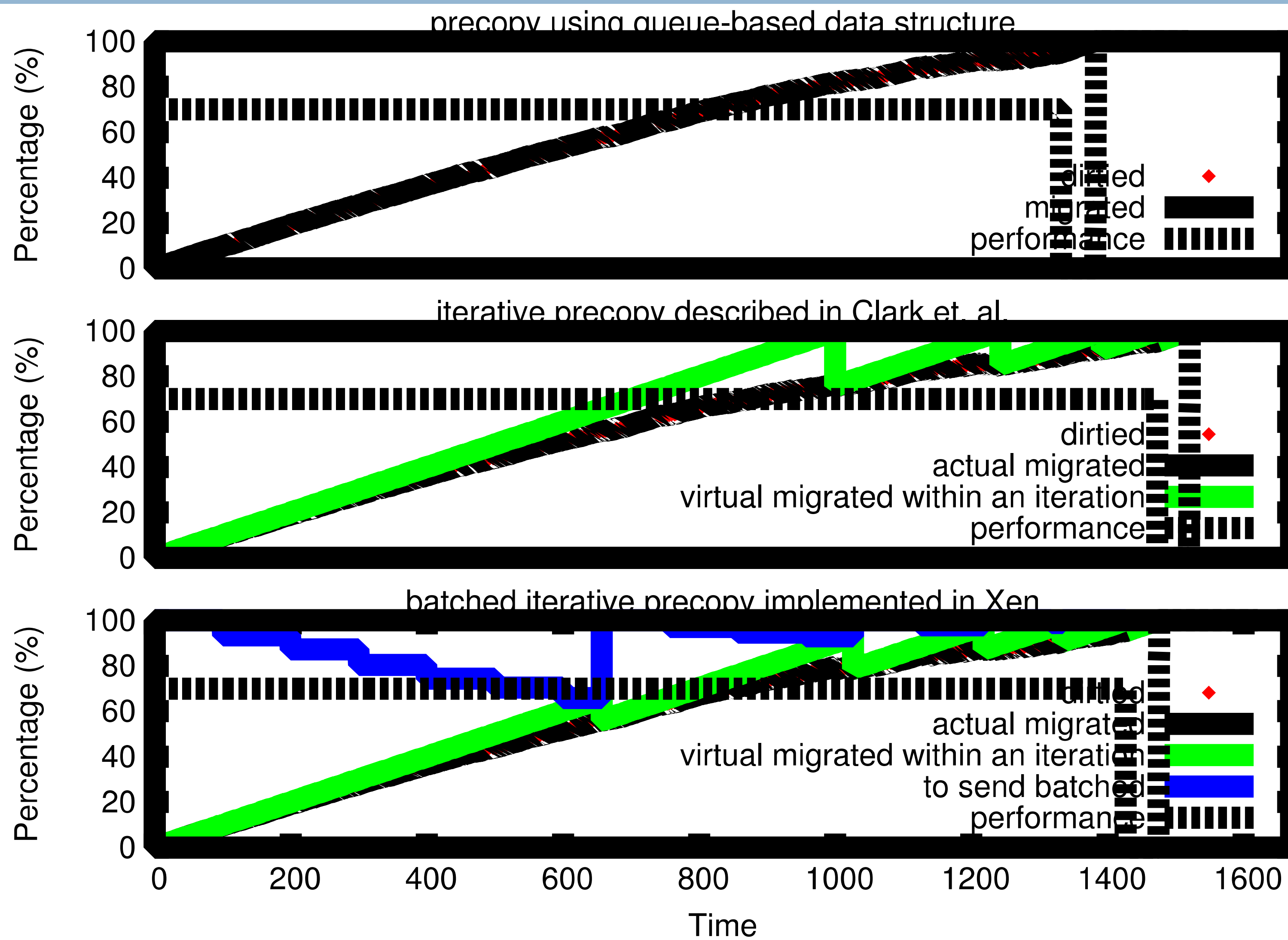
Conclusions

- Generalized the hybrid combination of memory and storage migration by (M, S)
- Defined the restoration agility, Gamma, to describe the liveliness/performance of a (M, S) migration
- Proposed a method to find the best (M, S) pair to achieve good restoration agility
  - Improved prediction with profiling and dirtying rate function
  - Ternary search of (M, S)
- Unique implementation of fractional hybrid copy

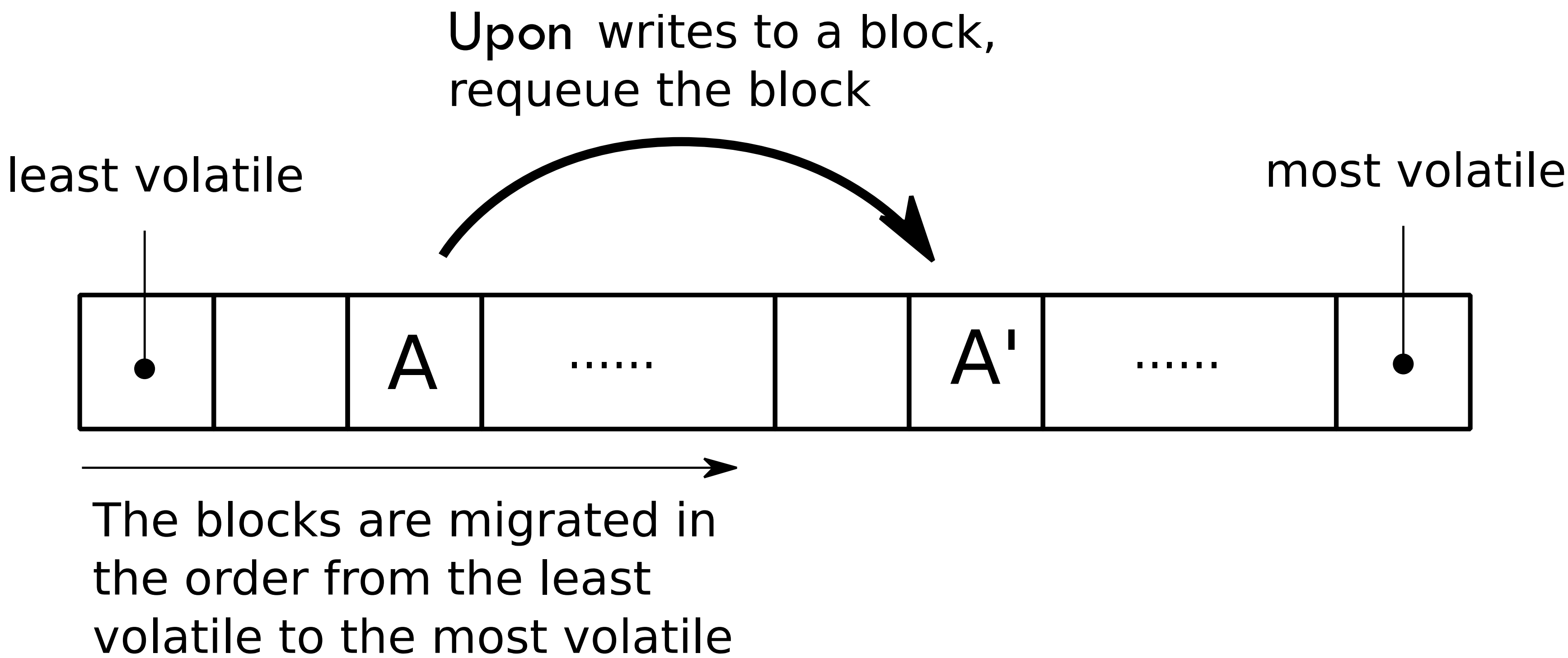
Thanks and Q&A!

Backup Slides

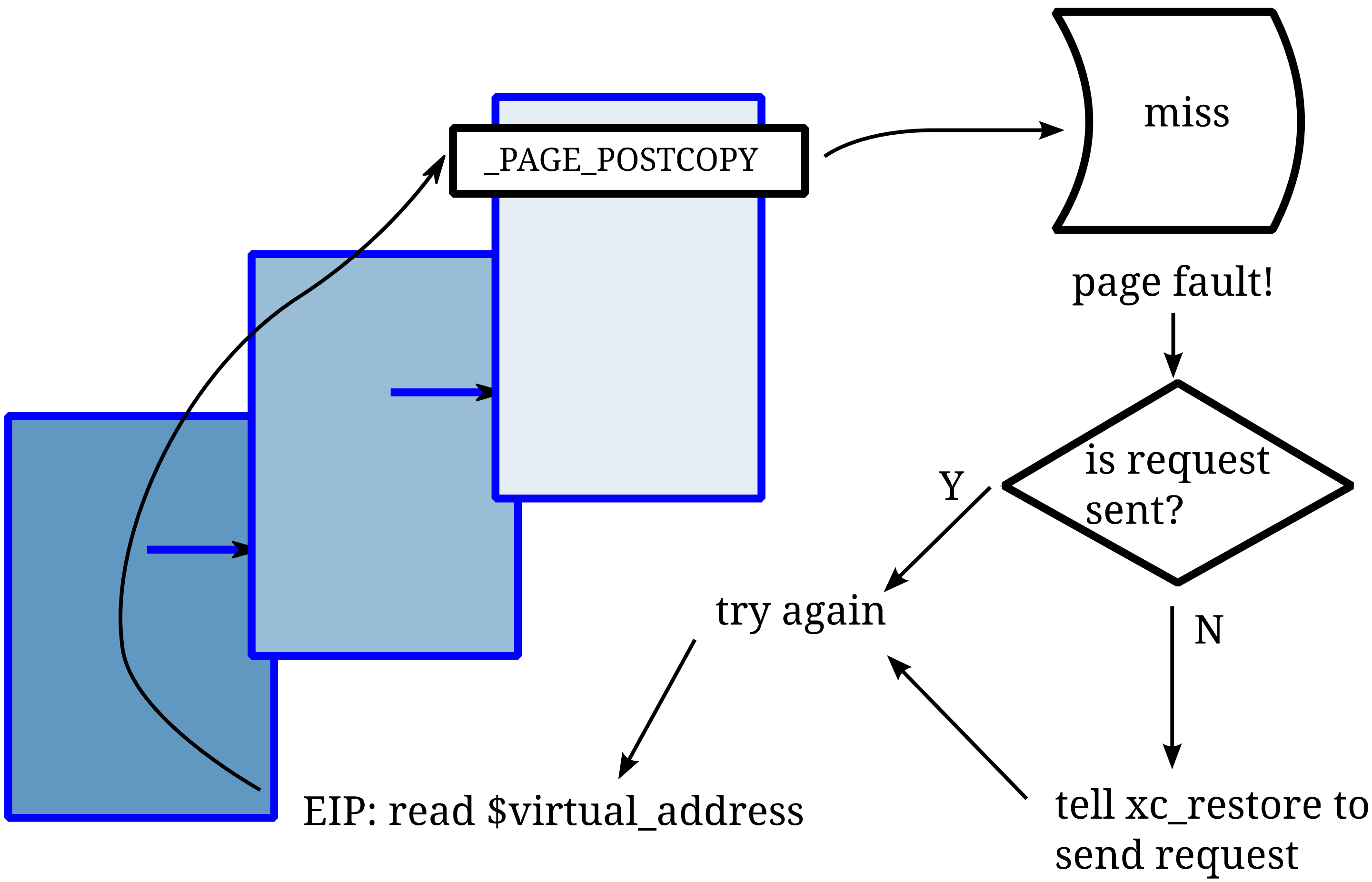
Pre-copying of Storage



Post-copying of Storage



Post-copy of Memory (Miss)



Post-copying of Memory (Hit)