

# Fast Wide Area Live Migration with a Low Overhead Through Page Cache Teleportation

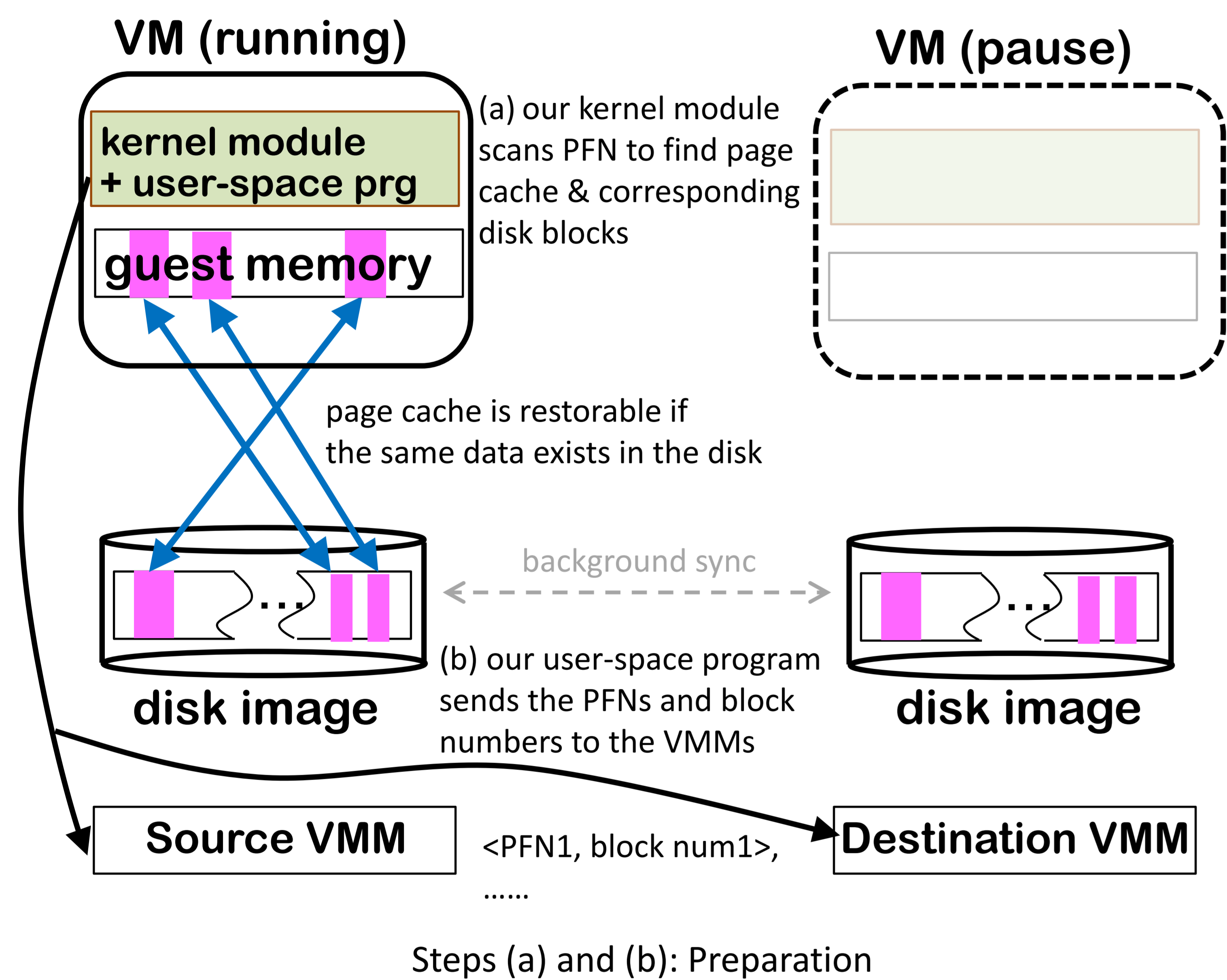
Soramichi Akiyama\* Takahiro Hirofuchi† Ryousei Takano† Shinichi Honiden\*‡

Email: akiyama@nii.ac.jp \* University of Tokyo † National Institute of Advanced Industrial Science and Technology ‡ National Institute of Informatics

## - Abstract -

Wide area live migration is essential for VM placement optimization between datacenters. However, it takes long time due to **slow WAN and large page cache in the VM memory**. **Restorable page cache is redundant** on the VM memory & disk image thus must be deduplicated. We propose page cache teleportation, which **copies restorable page cache from the disk image instead of transferring it via WAN**. It greatly **reduces the total migration time of wide area migration** with lower performance overhead.

## - Page Cache Teleportation -



Steps (a) and (b): Preparation

Kernel Module v.s. Alternative Approaches

	Kernel Module (our proposal)	Introspection-based	Disk IO-Monitoring*
Implementation	Easy (<200 loc)	Hard	Middle
Runtime overhead	None	None	Disk write hooked
Migration overhead	Small (<1 sec)	Big (binary scan)	Small
Guest modification	Yes	No	No

\*Jo et al., VEE'13

## - Challenges of WAN Migration -

### 1. Bandwidth is narrow between datacenters

- links are shared by many users

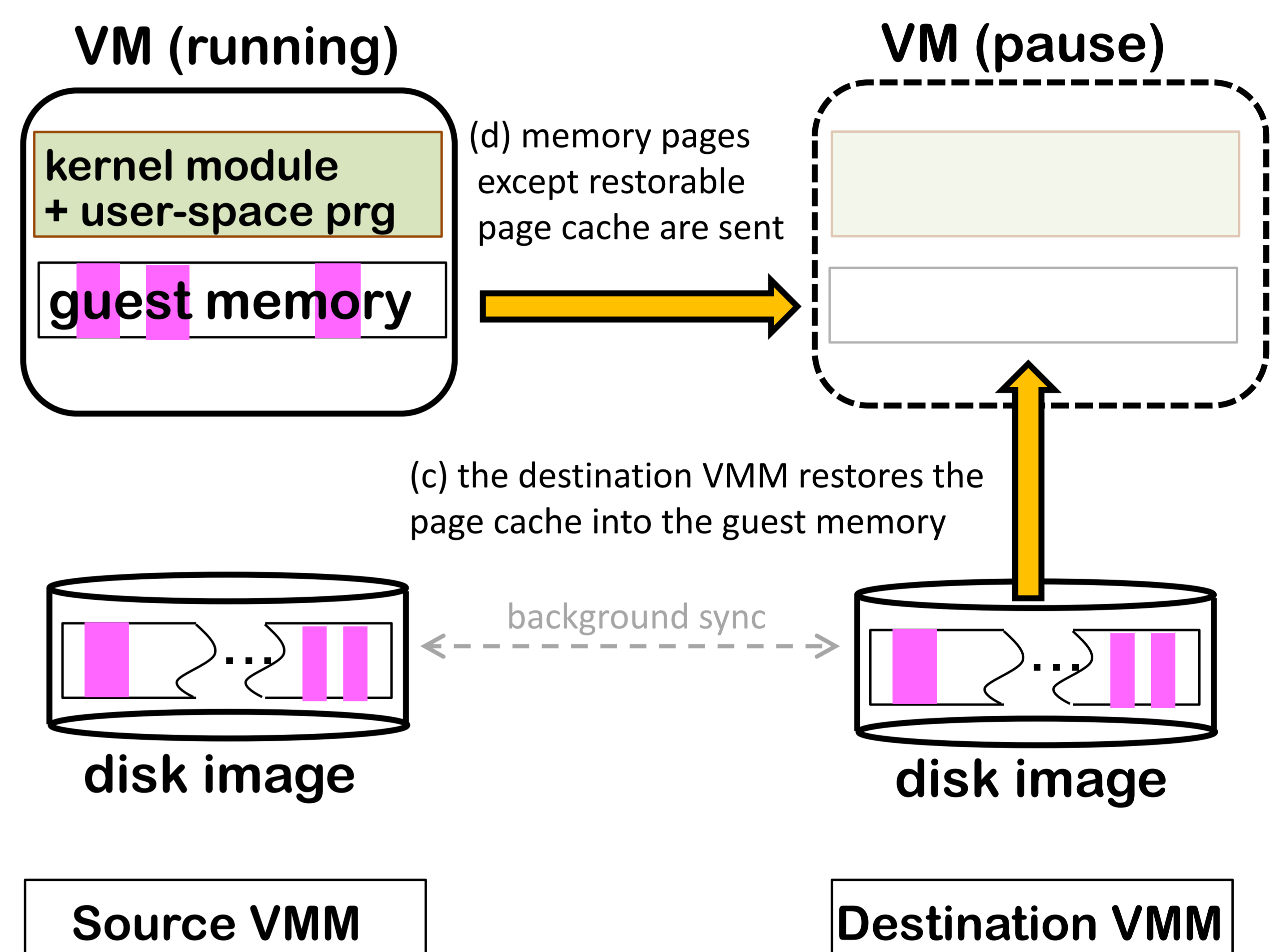
→ Reducing transferred data is important

### 2. Large page cache is duplicated on VM memory & disk

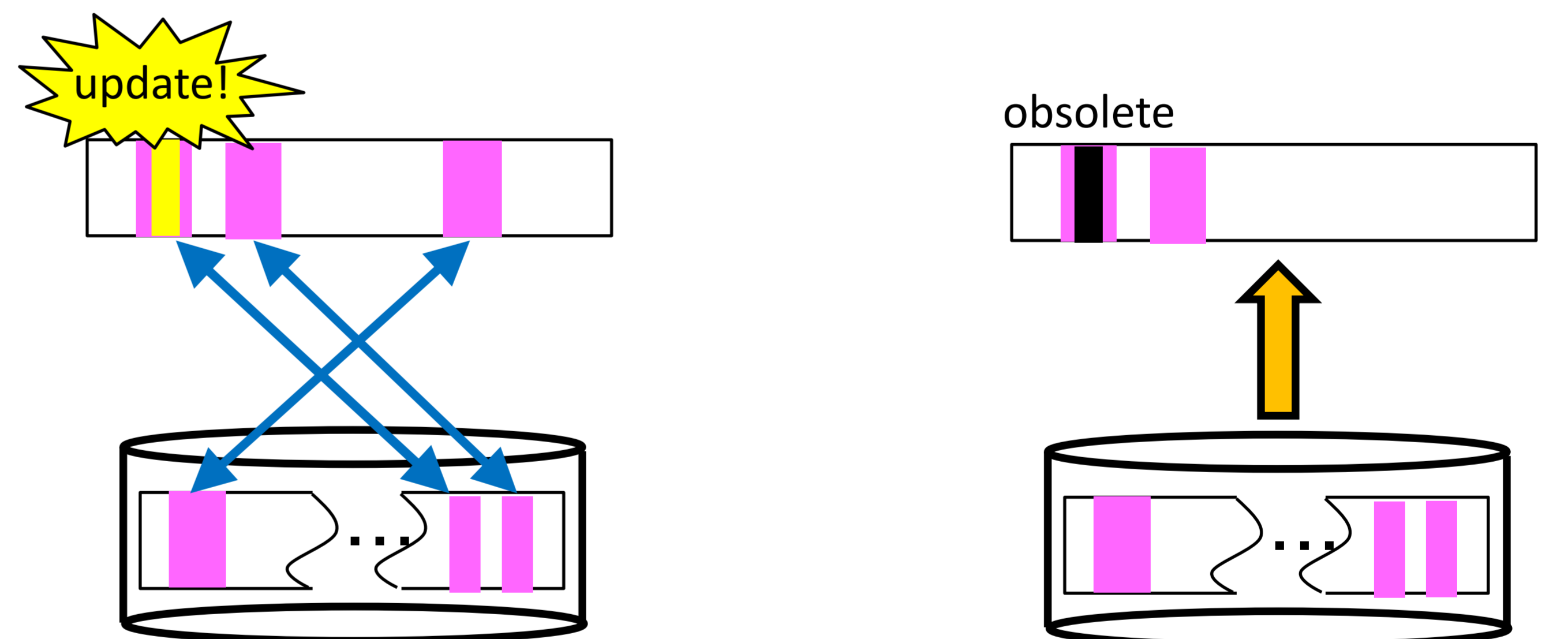
- workloads with large data (e.g. web server, DB)
- OSs assign all free memory for page cache

→ page cache must be deduplicated to reduce the cost

➡ **Fast wide area live migration focusing on page cache is required!**



Steps (c) and (d): Migration Execution

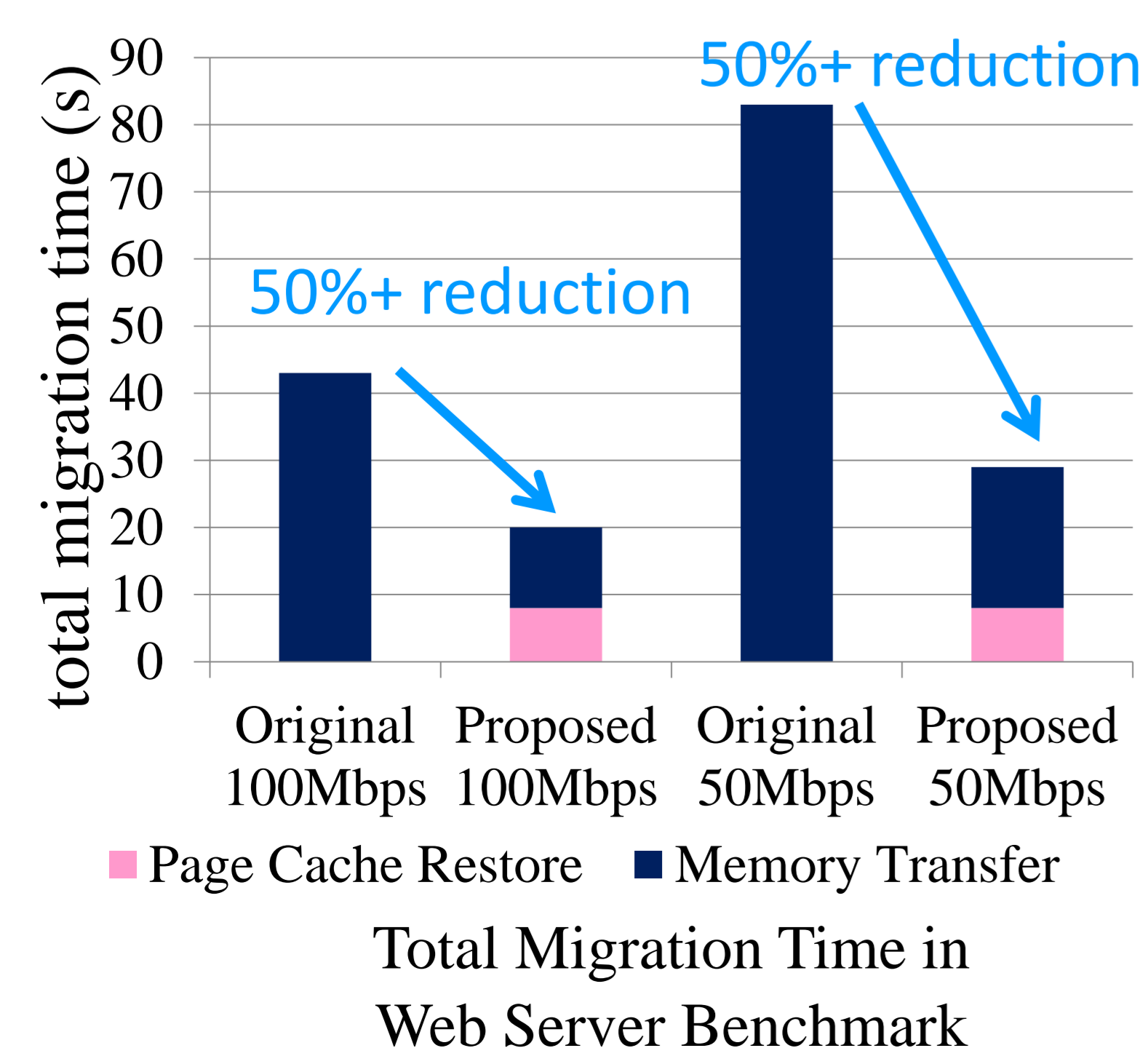


## - Experimental Results -

Total migration time in the original/proposal with 1G mem/1 vCPU VM

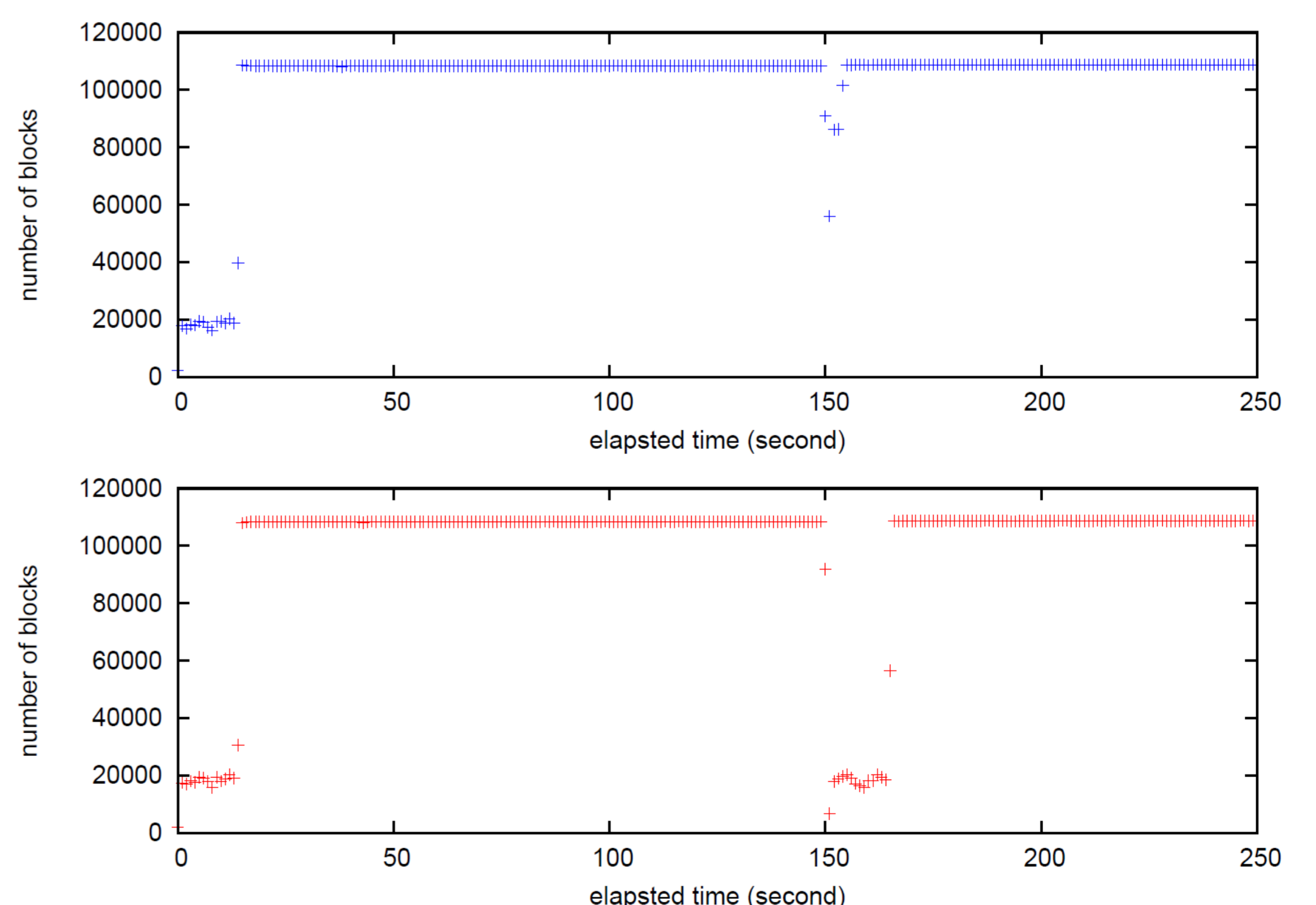
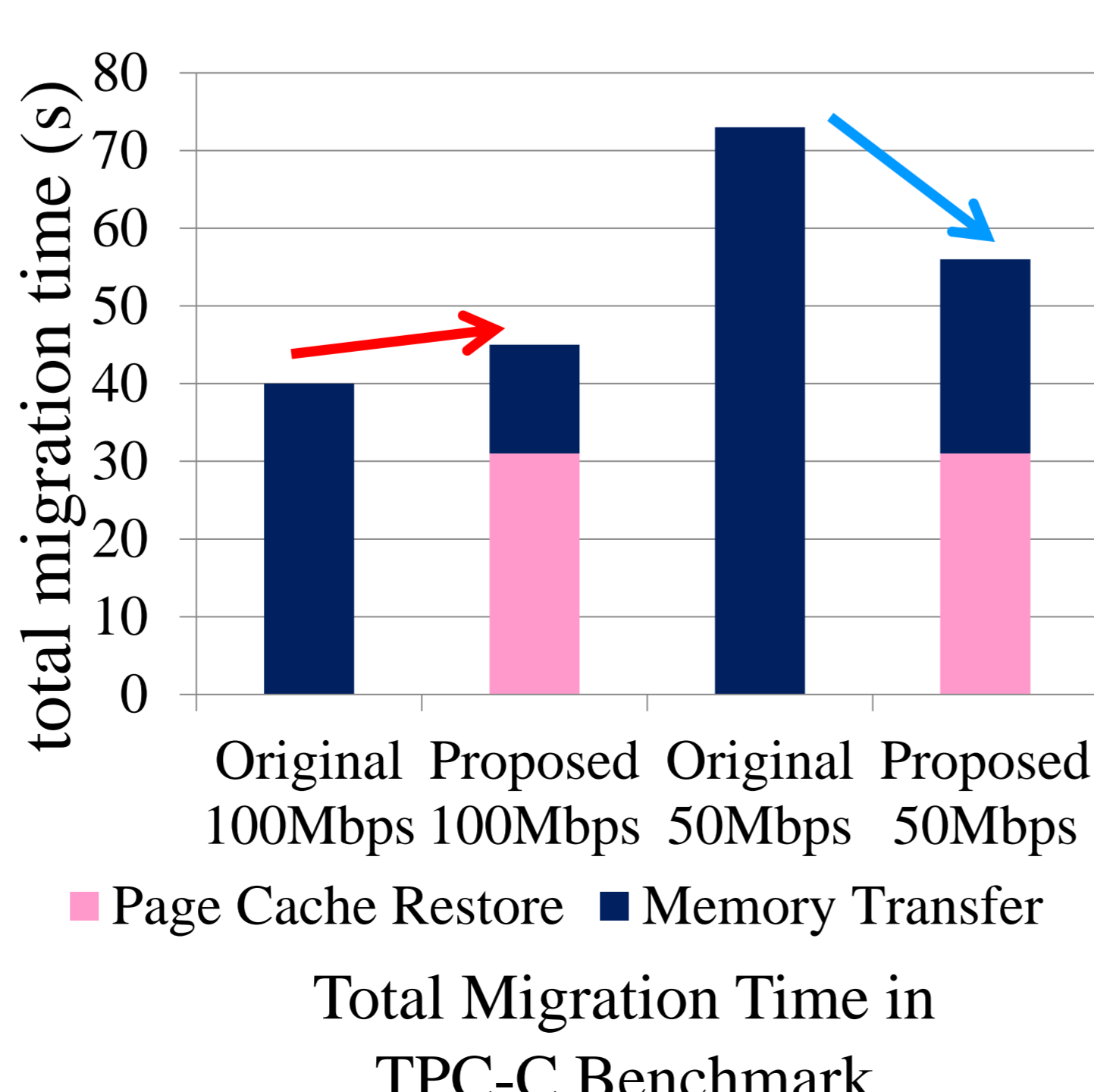
Web Server:

- Simulated static web site
- HTMLs are 300 MB in total



TPC-C:

- Simulated net-shopping site
- DB data is 1.9GB in total



File read throughput (blocks/s) with a migration executed at t = 150. With our proposal it recovers in 3 seconds (top). If however, page cache is deleted to reduce transferred data, it degrades for 15 seconds (bottom).