THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

Rack Scale Apps

Neel Shah (nshah95@gwu.edu)

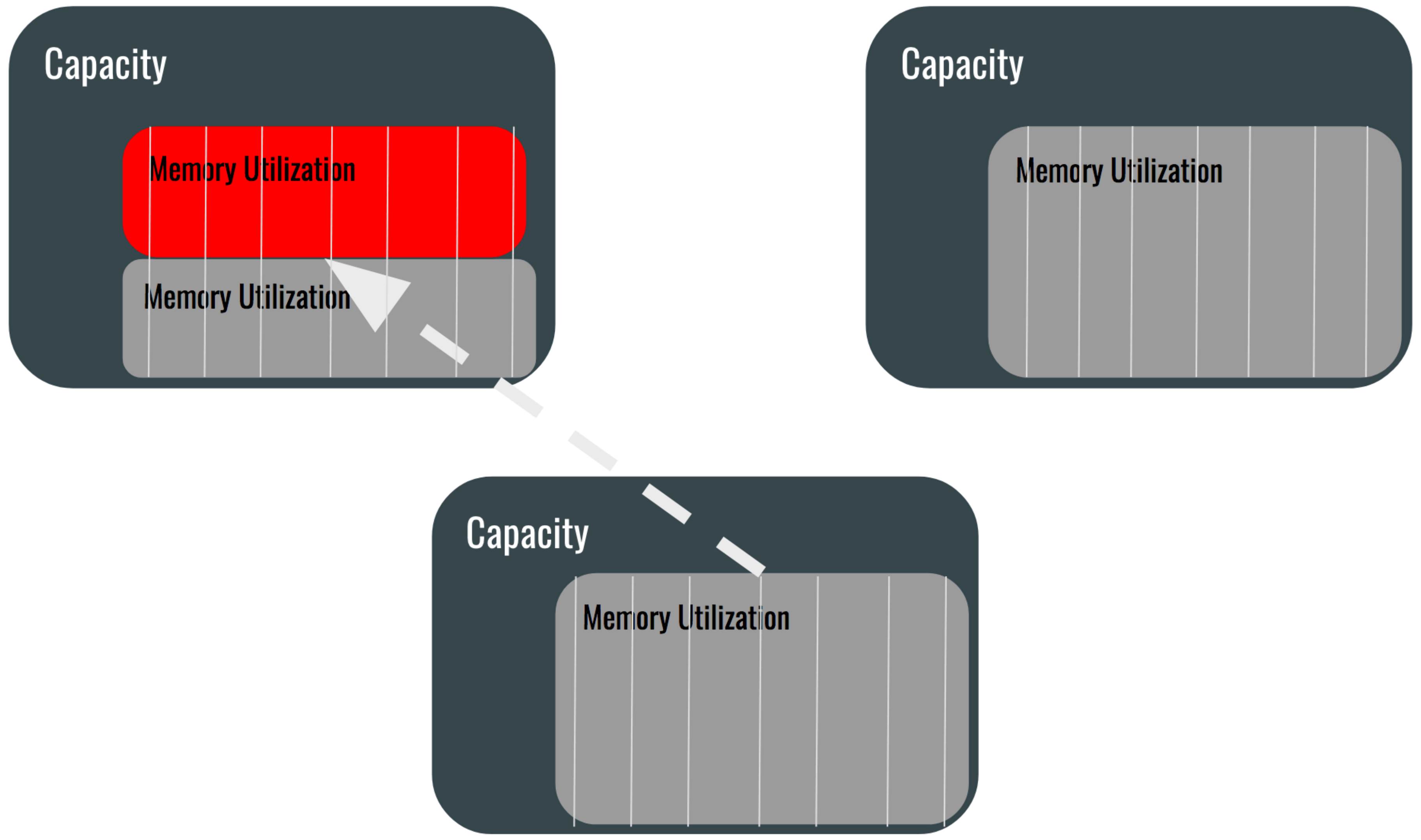
THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

Background

It has been said that “the cloud” is one of the biggest trends that computing has seen. The cloud is a model where servers in datacenters host programs or compute data for end users to consume. Datacenters are not fit to reliably scale to compensate the cloud’s growth rate. Many servers waste resources by sitting idle, costing the world upwards of $30 billion per year. We need to better manage our computing resources.

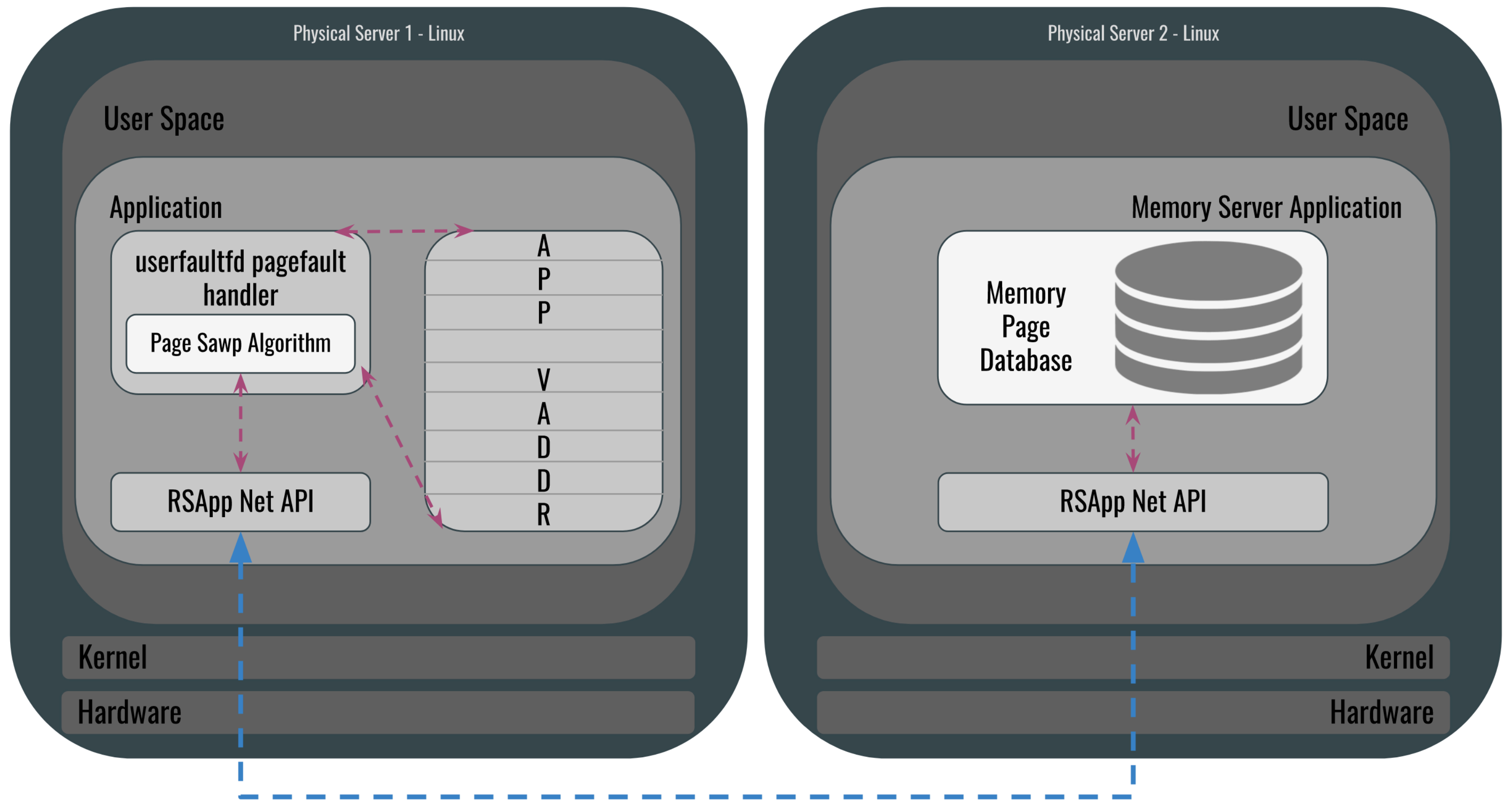
Key Problems



- Three memory utilization scenarios: **underutilization**, **high-utilization**, **exceeding capacity**
- **Goal:** maintain equal memory utilization across all of the servers in a rack
- **Solution:** move memory pages between servers to distribute load evenly
- **Challenges:**
  - Memory is organized into **pages**
  - **When** do we “swap” these pages back and forth between servers?

Rack Scale Apps

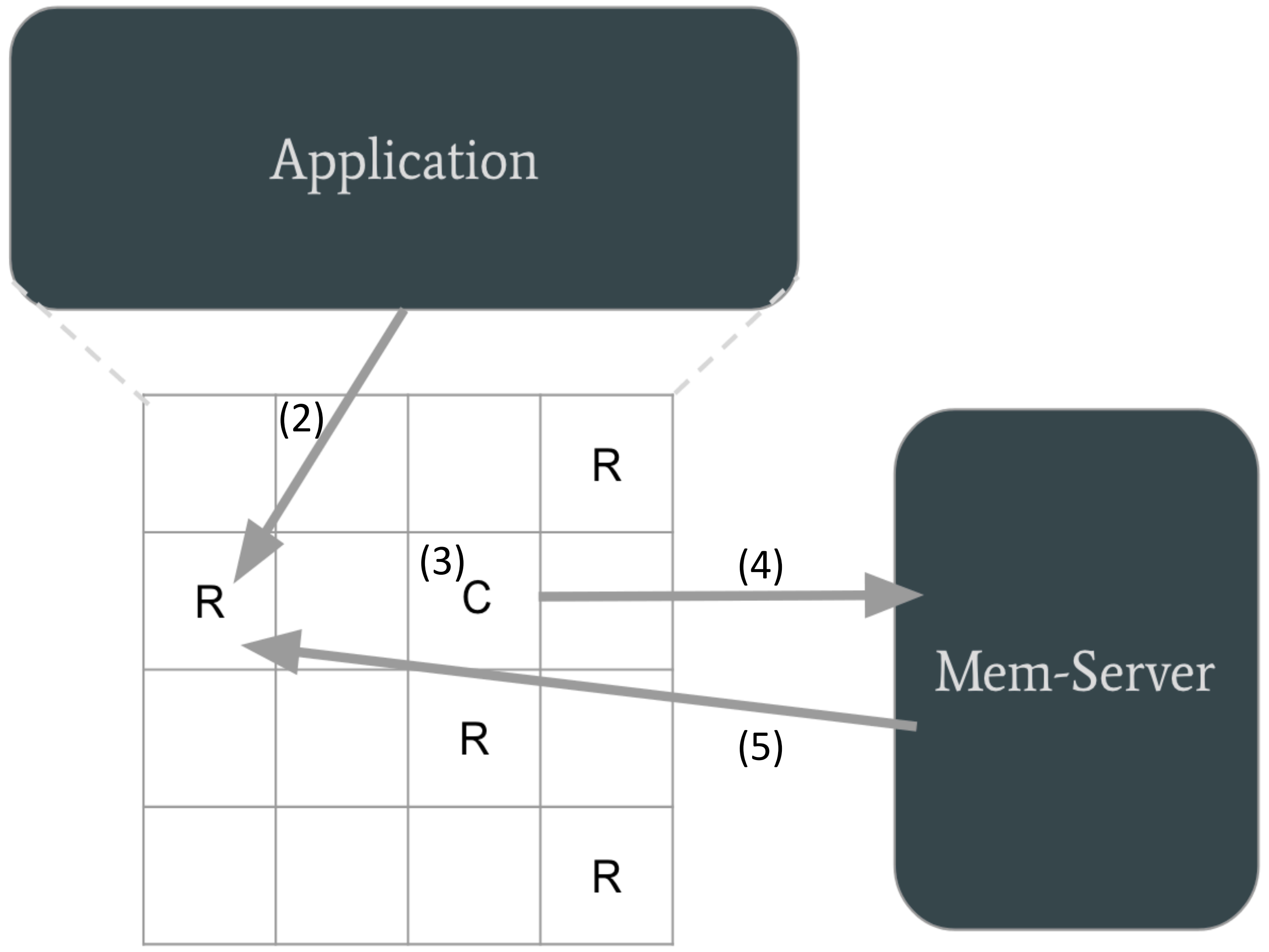
- **userfaultfd**-based page fault handler monitors when an application access certain pages of memory
- **RSApi Net Library** transfers data to and from a remote memory server
- **Memory Server** receives, stores, maintains and sends back all memory data from a remote application
- Pools server hardware (memory) together in a rack by allowing an application to store and retrieve memory pages on another server
- Page swapping algorithm determines when to swap pages of memory between the application and memory server.



Page Swapping Algorithm

The page swapping algorithm intelligently decides when to swap pages back and forth between an application and the memory server.

1) Mark arbitrary amount of pages as “remote” pages
2) Wait for the application to access a “remote” page
3) Find the “coldest” page in the application’s address space
4) Swap the contents of the “coldest” page to the memory server
5) Check the memory server to see if the ”remote” page that was accessed has data stored there. If so, swap it back


Application
(2)
R
R
(3)
C
(4)
R
(5)
R
Mem-Server