

Today

Distributed Objects (Lesson 6)

✓ Spring Kernel

✓ Java RMI

✓ ETB

Distributed Subsystems (Lesson 7)

⇒ \* Global memory system

Friday

Yousef Khalidi Guest Lecture

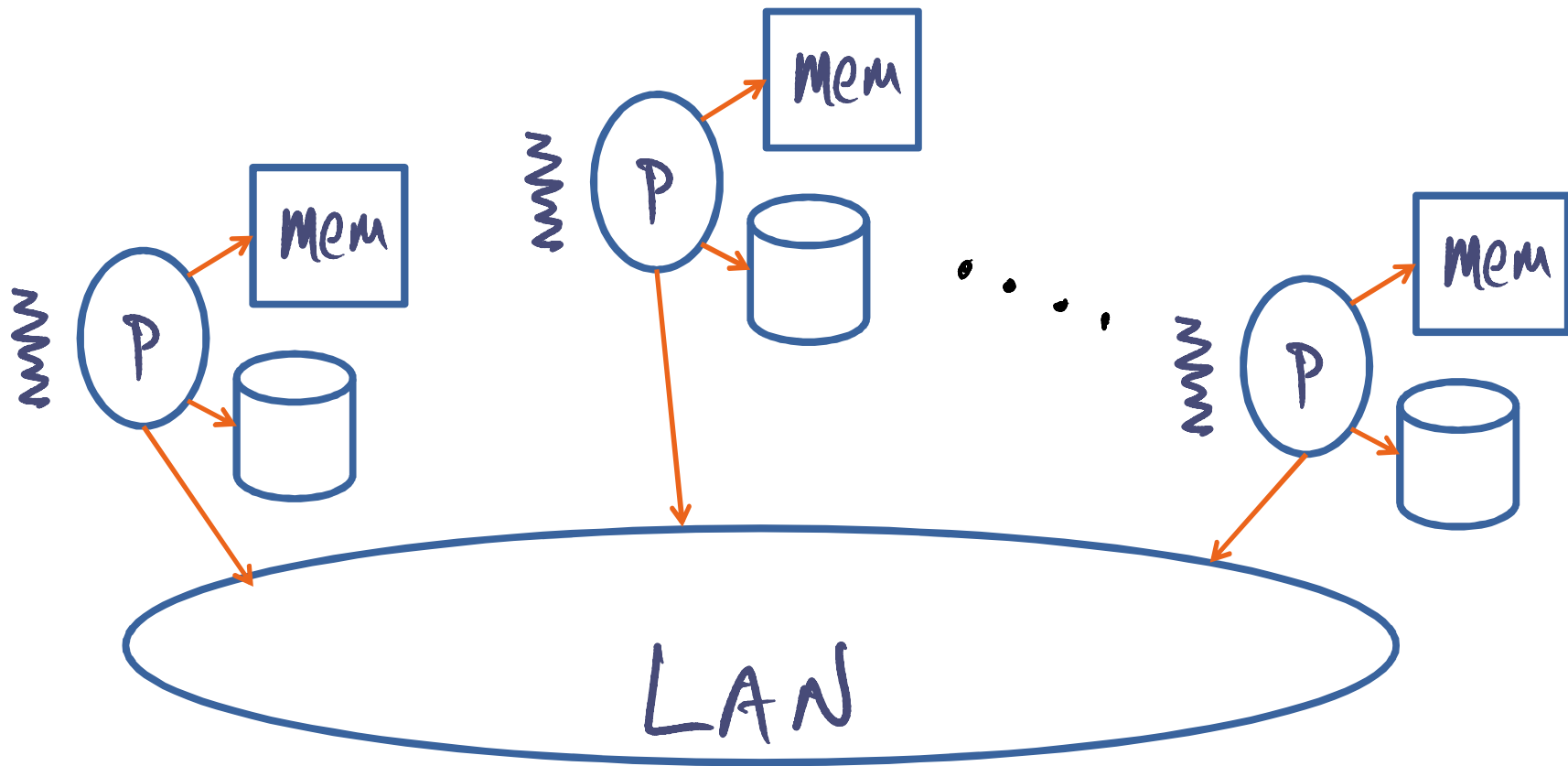
+  
Pizza !!

# Thought Experiment

GMS

- How can we use peer memory  
for paging across LAN?

# Context for global memory system



memory pressure

- \* different for each node

- \* How to use idle cluster mem?

- \* remote mem access faster than disk

memory manager:

VA  $\rightarrow$  PA or disk

GMS:

VA  $\rightarrow$  PA or cluster mem  
or disk

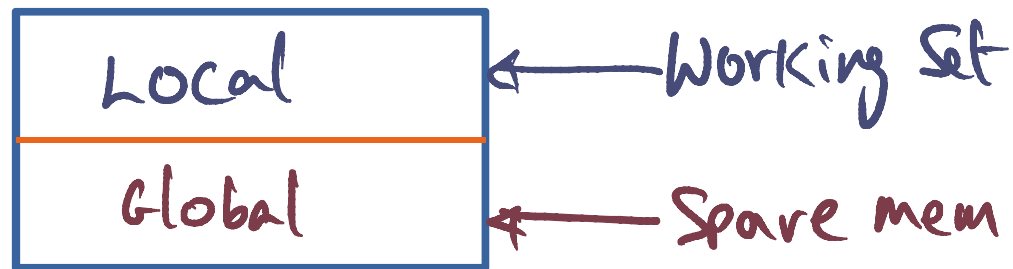
## GSM Basics

"Cache" refers to physical memory (i.e., DRAM)

not processor cache

Sense of "community" to handle page faults at a node

Physical memory at a node



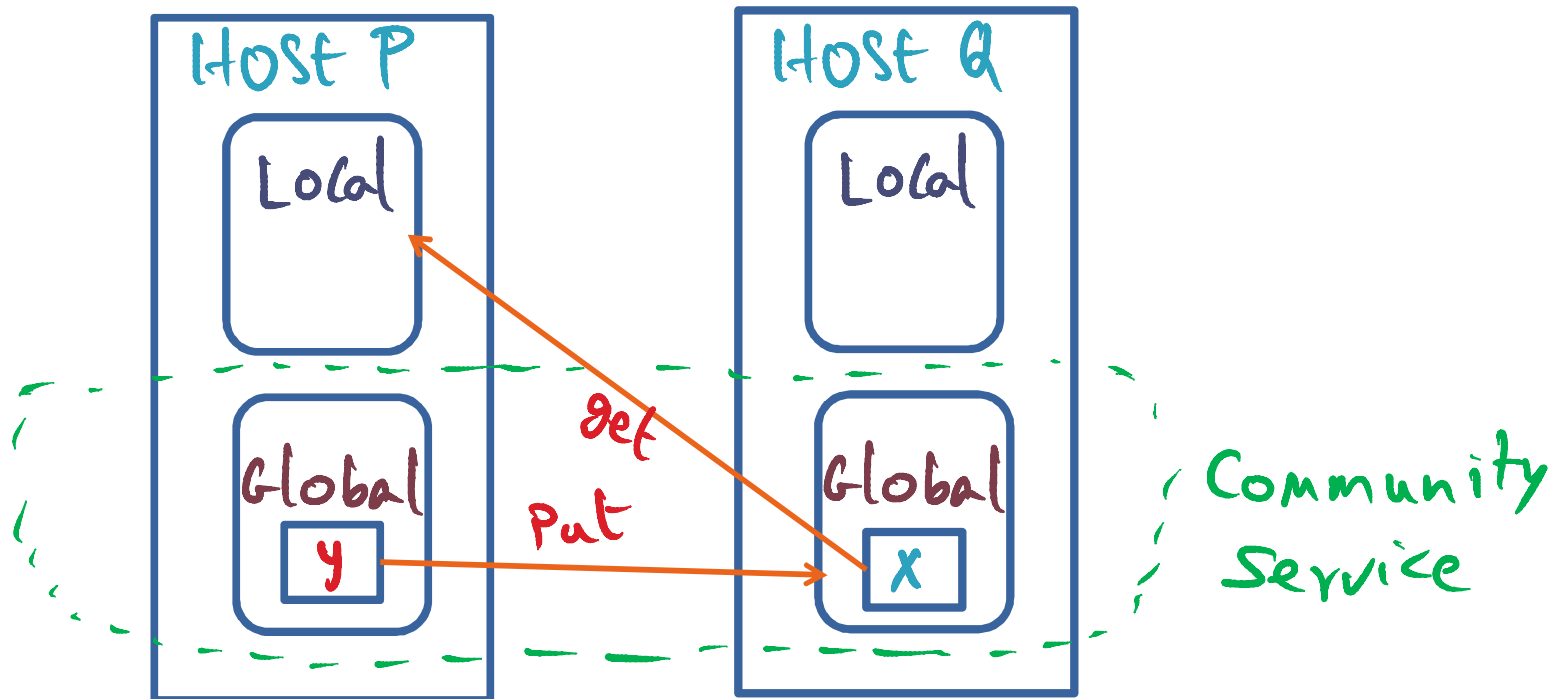
— "private" Vs. "shared" pages

— Coherence for shared

# Handling Page faults - Case 1

## Common Case

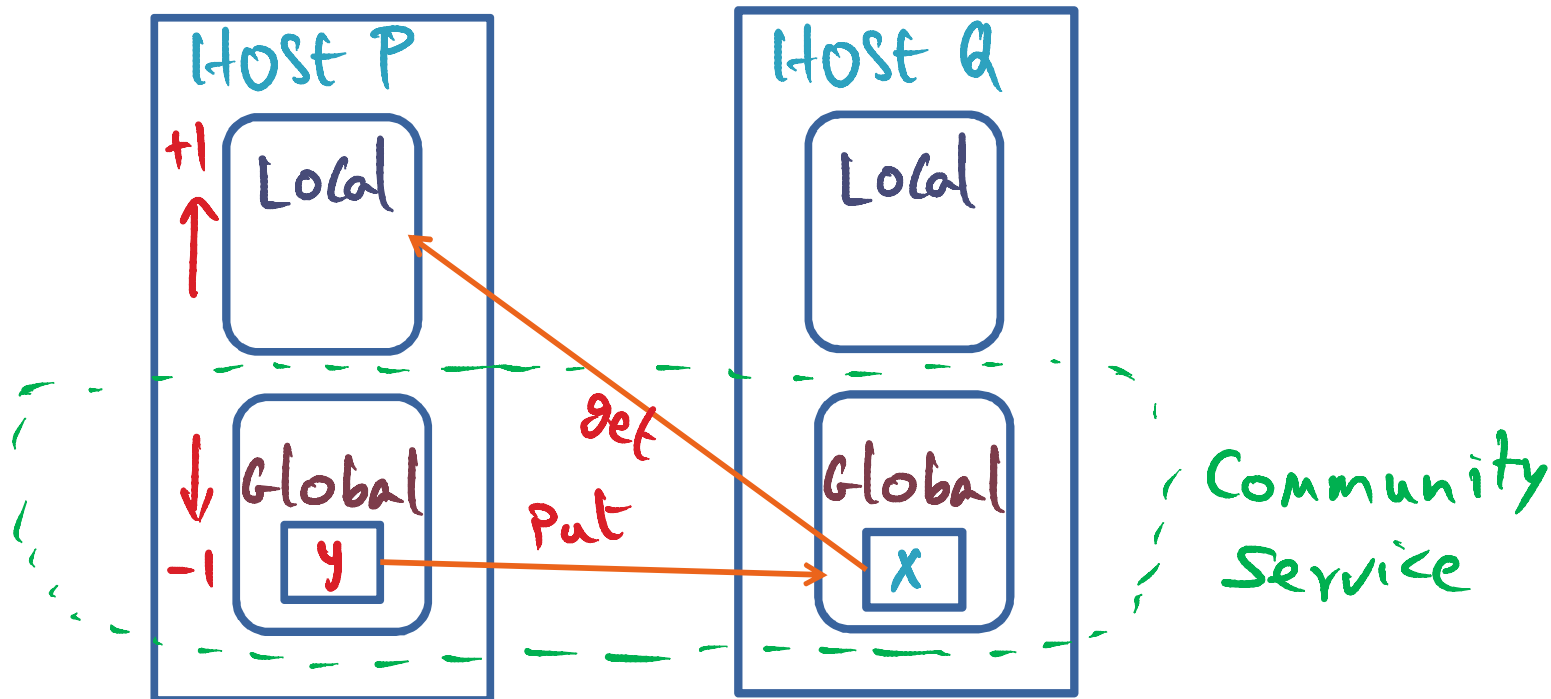
- Page fault for  $x$  on node  $P$
- hit in global cache of some node  $Q$



# Handling Page faults - Case 1

## Common Case

- Page fault for  $x$  on node  $P$
- hit in global cache of some node  $Q$

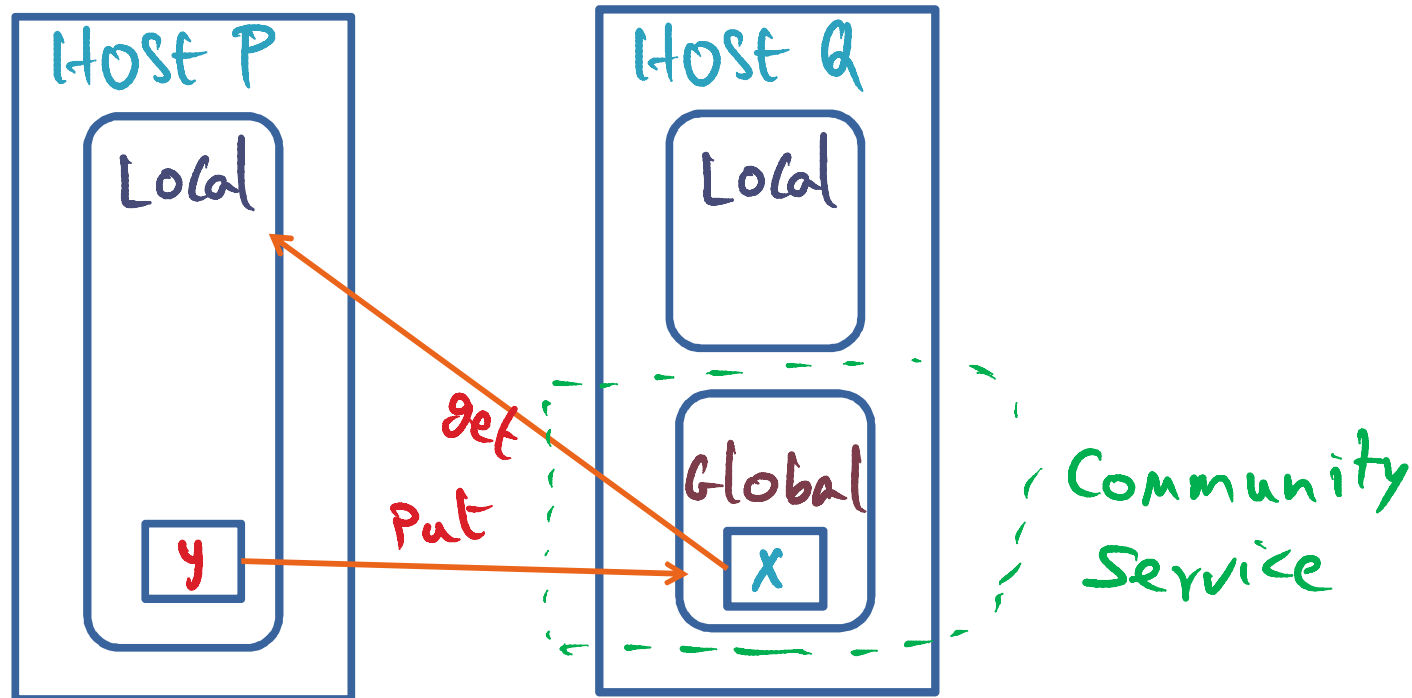


## Handling Page faults - Case 2

Common Case with memory pressure at P

- Page fault for  $x$  on node P

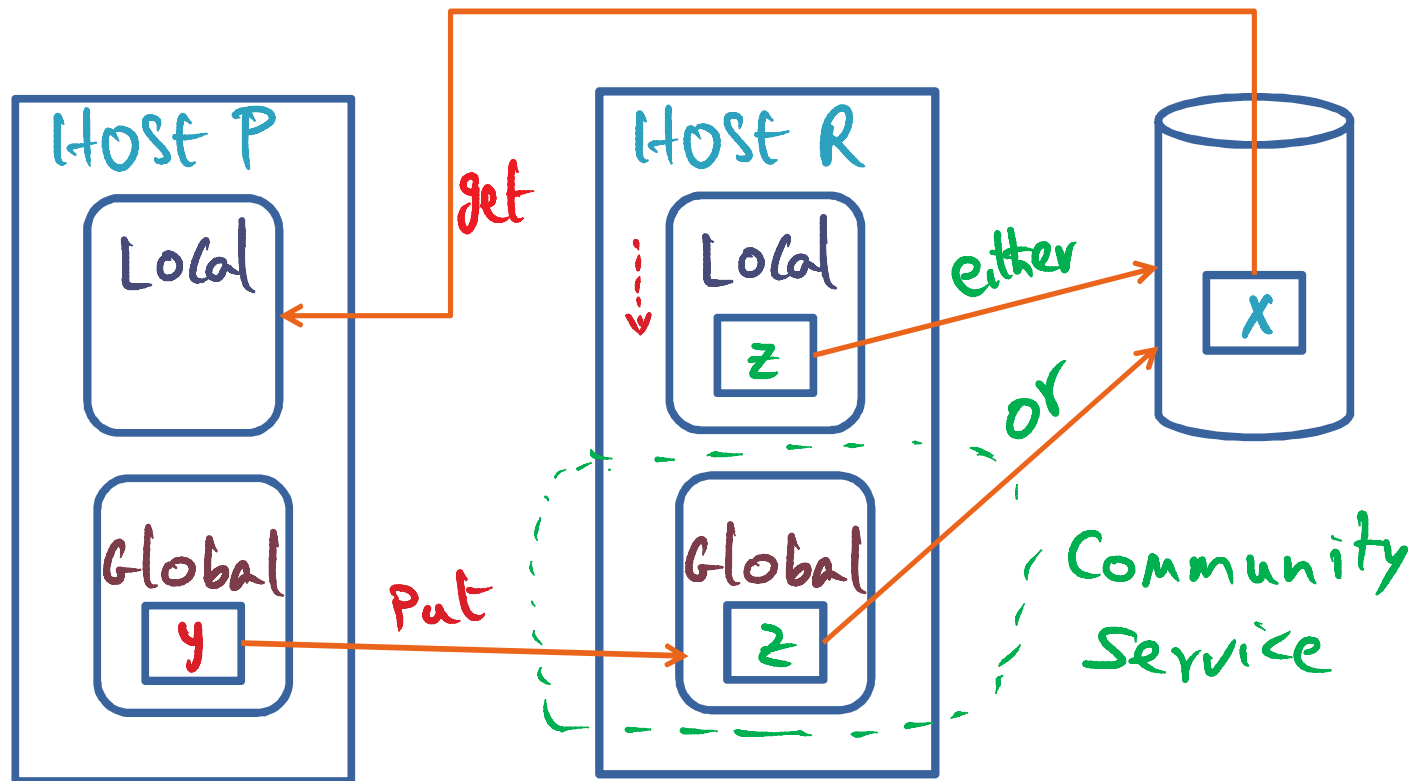
- swap **LRU** page  $y$  for  $x$



## Handling Page faults - Case 3

Faulting Page on disk

- Page fault for  $x$  on node  $P$
- page not in cluster

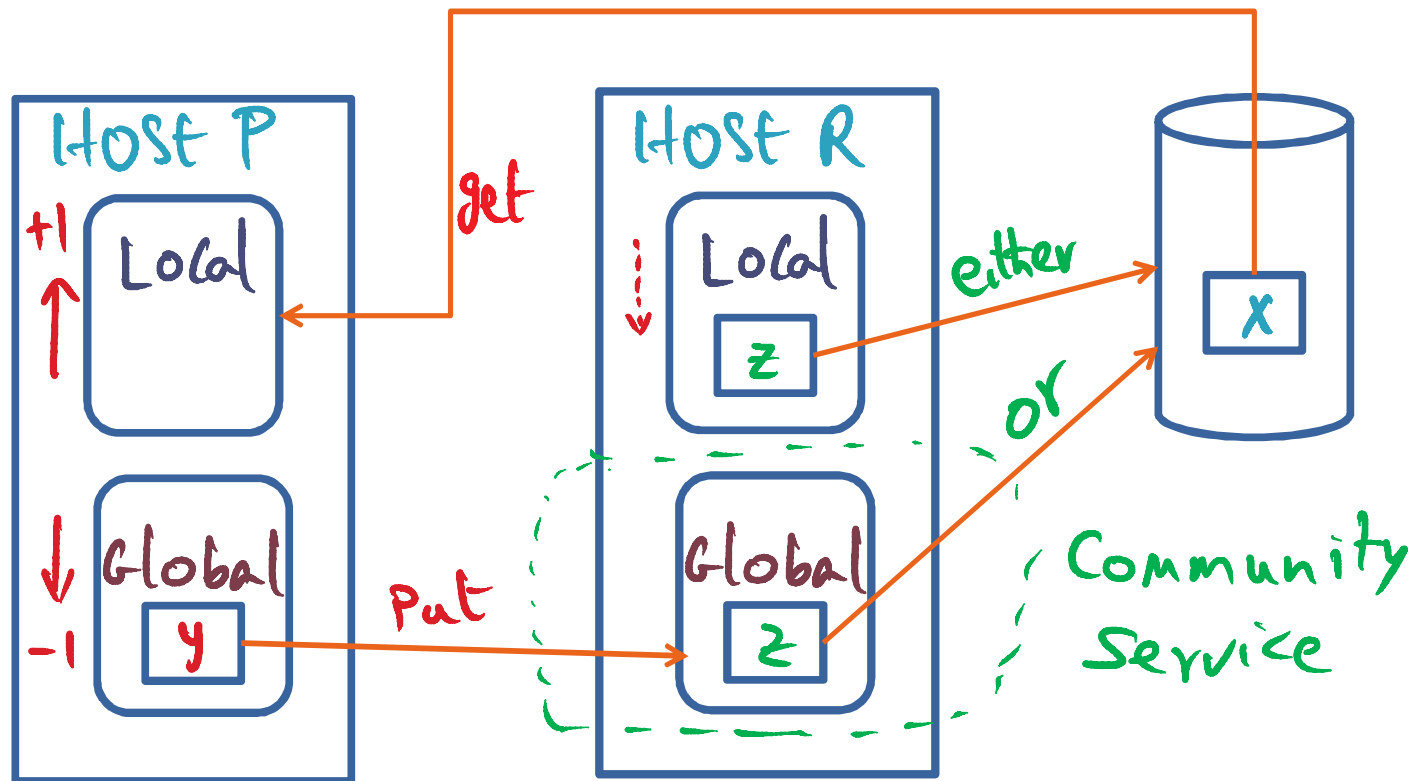




## Handling Page faults - Case 3

Faulting Page on disk

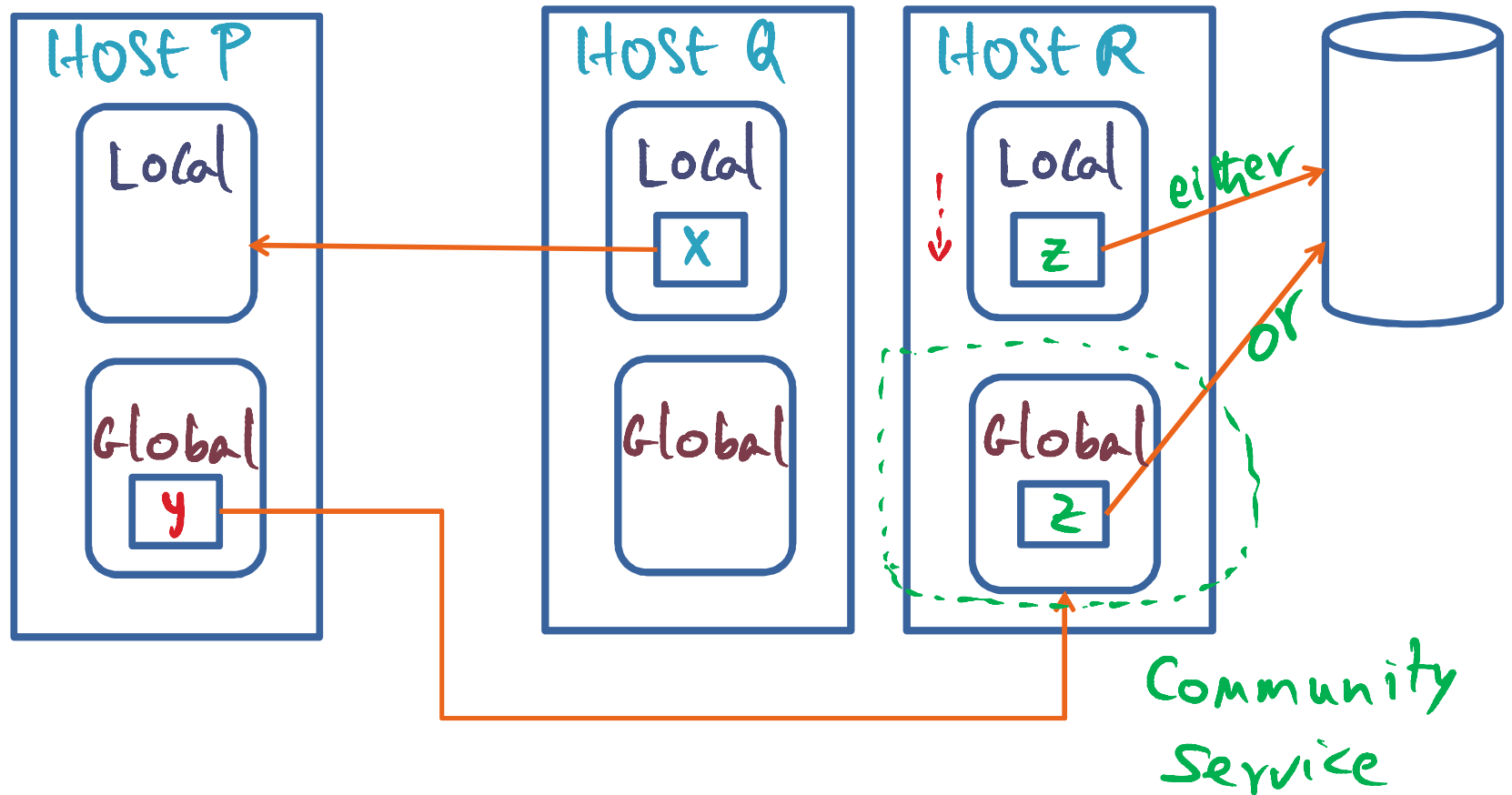
- Page fault for  $x$  on node  $P$
- page not in cluster



## Handling Page faults - Case 4

Faulting Page actively shared

- Page fault for  $X$  on node  $P$
- page in some peer node  $Q$ 's local cache



## Handling Page faults - Case 4

Faulting Page actively shared

- Page fault for  $x$  on node  $P$
- page in some peer node  $Q$ 's local cache

