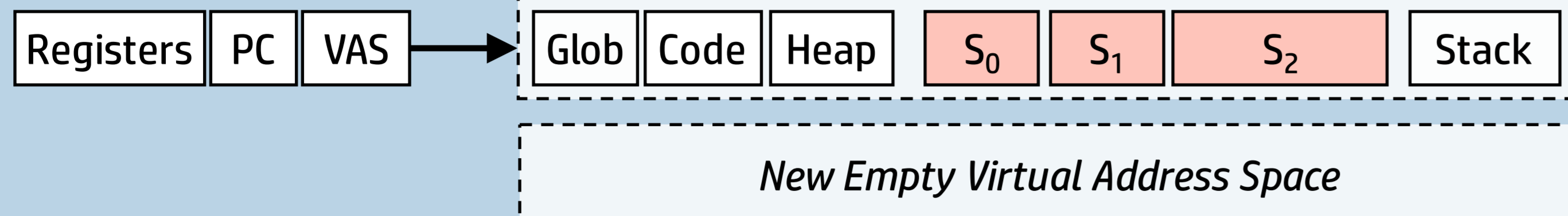


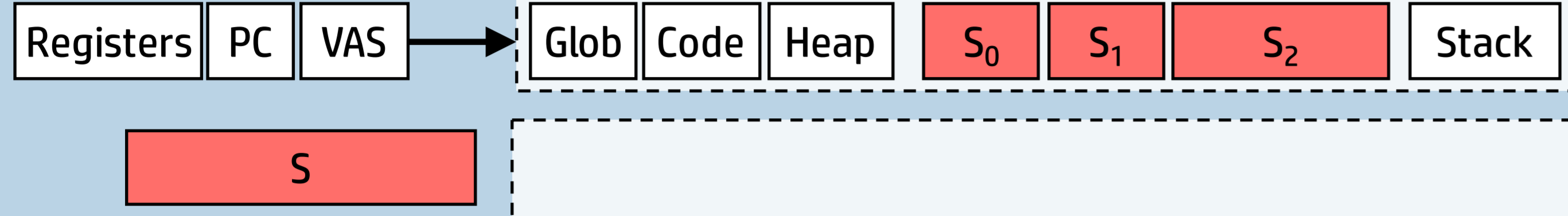
SpaceJMP: Programming with Multiple Virtual Address Spaces

Izzat El Hajj¹ (UIUC), Alexander Merritt² (GaTech), Gerd Zellweger³ (ETH), Dejan Milojicic (HP Labs),
 Reto Achermann (ETH), Paolo Faraboschi (HP Labs), Wen-mei Hwu (UIUC), Timothy Roscoe (ETH), Karsten Schwan (GaTech)
¹elhajj2@illinois.edu, ²merritt.alex@gatech.edu, ³gerd.zellweger@inf.ethz.ch

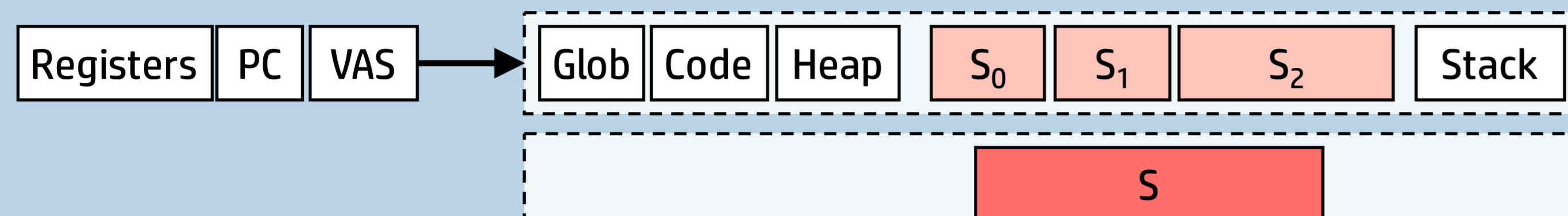
Create Address Space



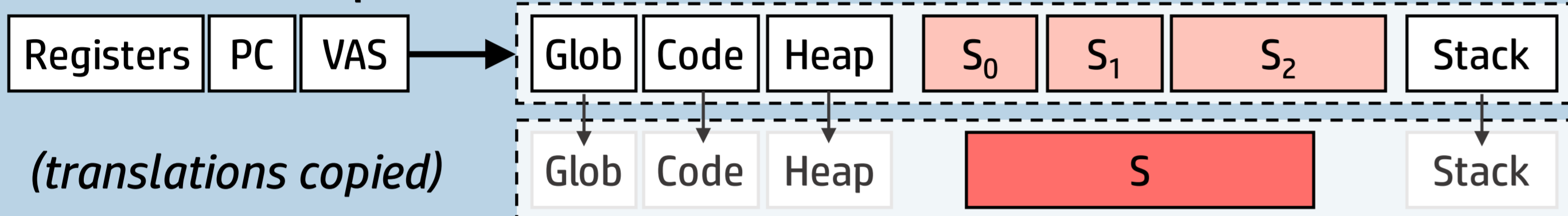
Create Segment



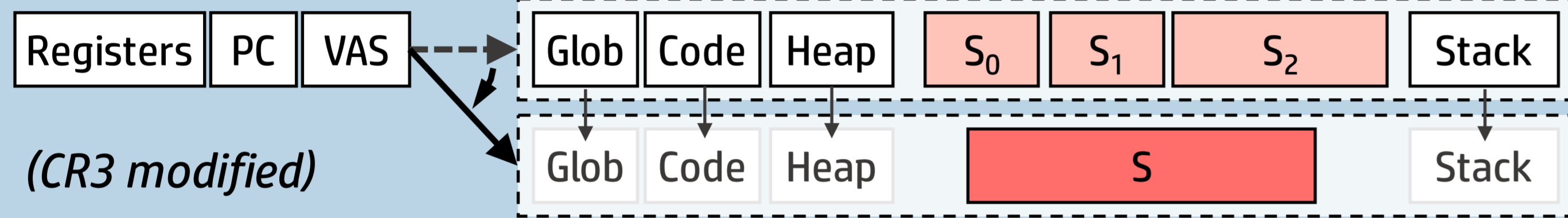
Attach Segment to Address Space



Attach Address Space to Process



Switch Address Space

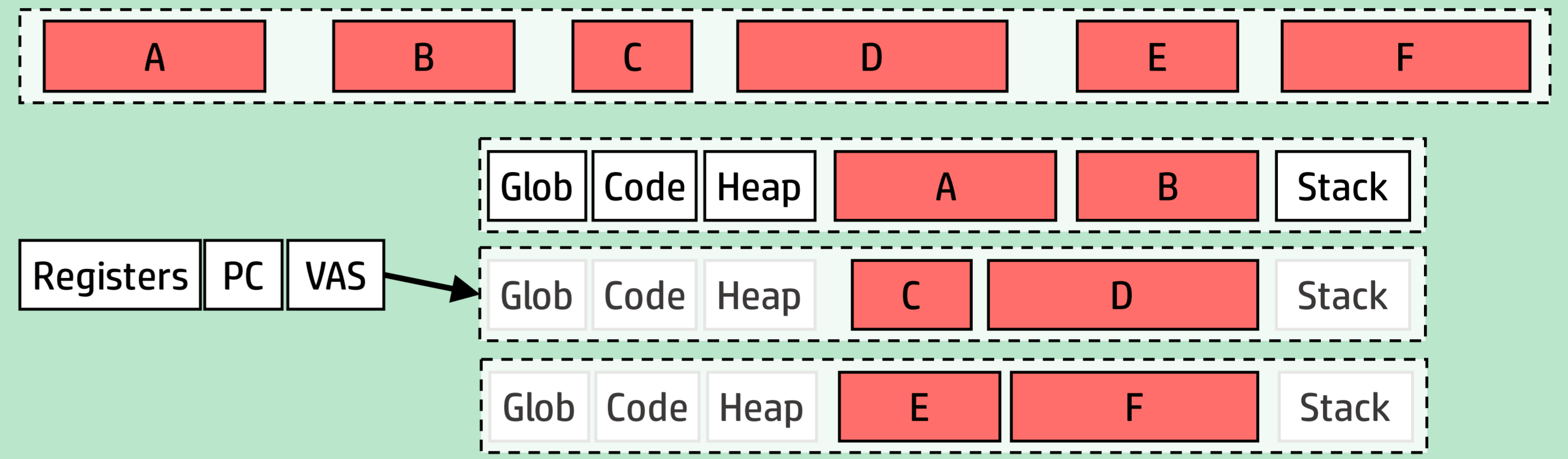


Promote address spaces to *first-class citizens*.

Processes create, delete, attach, and *switch* address spaces.

Overview

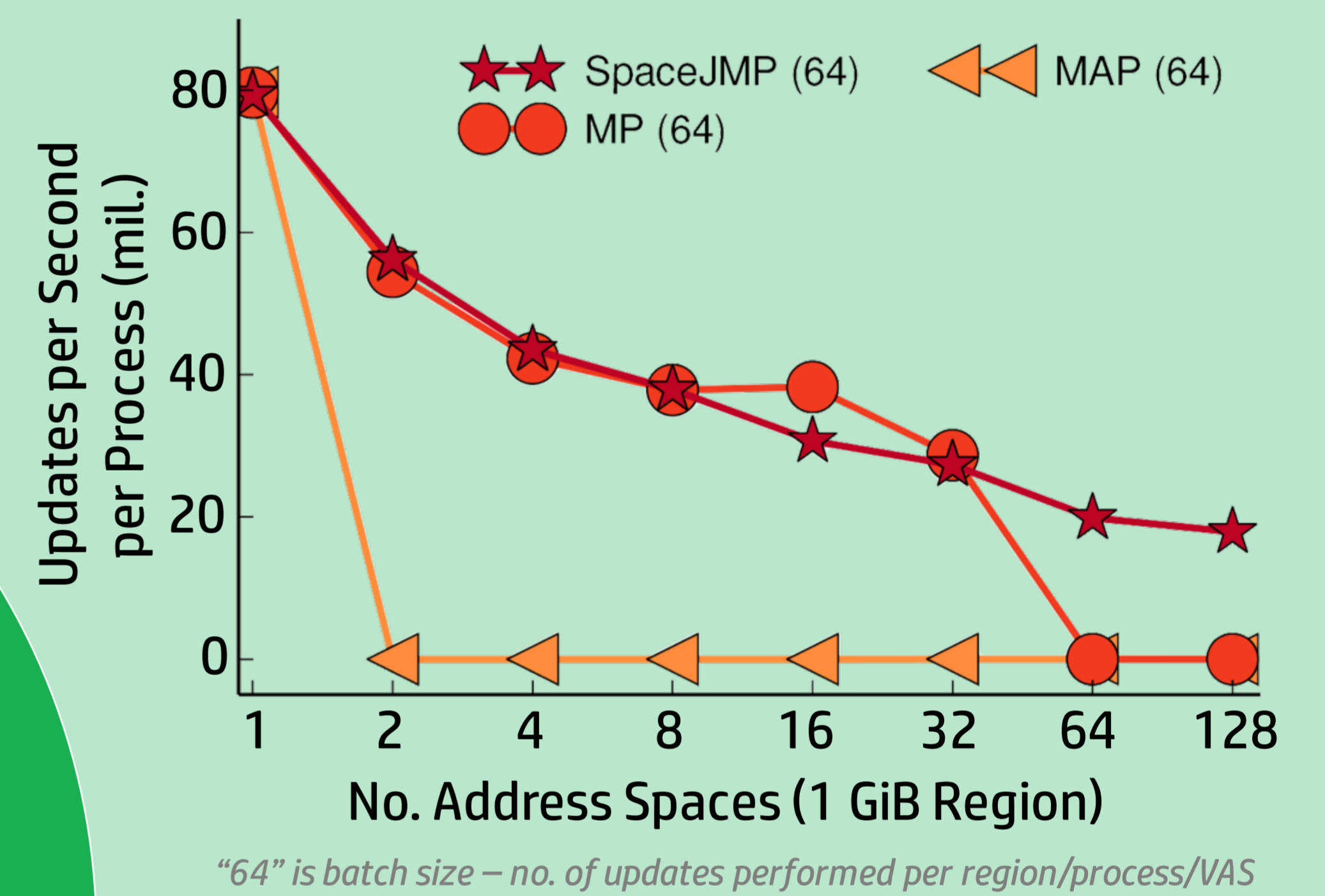
Addressing More Physical Memory than Available Virtual Memory



How to address table larger than virtual address space?

GUPS Benchmark – random updating of large linear table

- MP** – many processes, each with part of table + messaging
- SpaceJMP** – one process switches across VASes
- MAP** – map+unmap regions of table in/out of one VAS

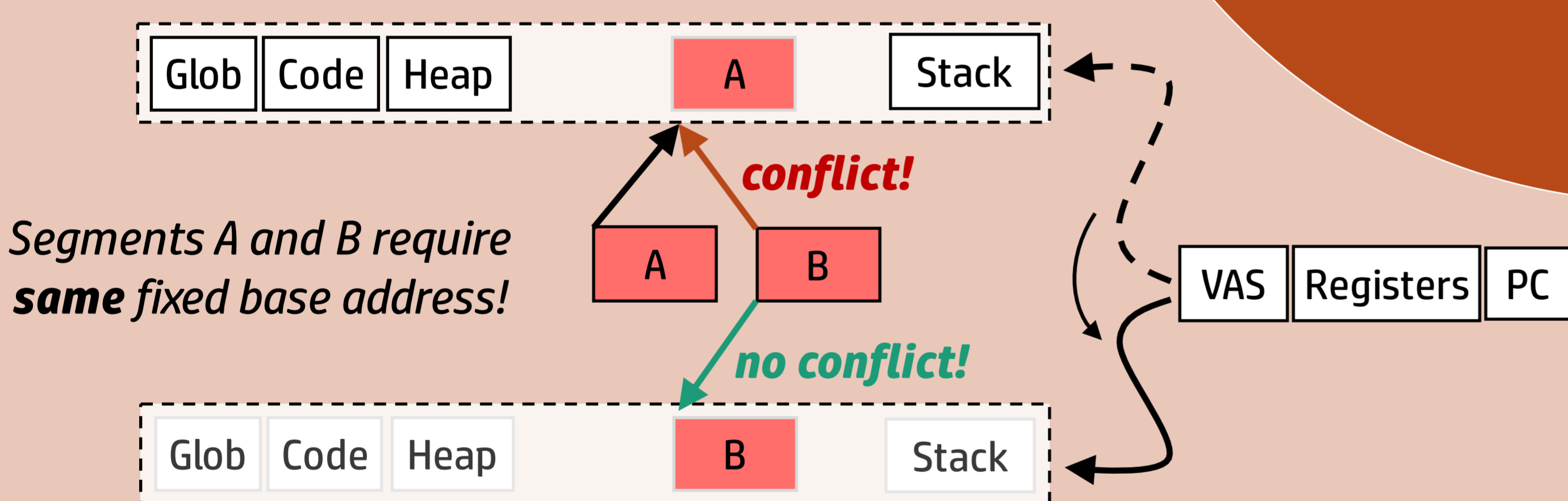


"64" is batch size – no. of updates performed per region/process/VAS

Large Physical Memory

Resolve Address Conflicts without Unmapping or Data Marshaling

Virtual Address Conflicts

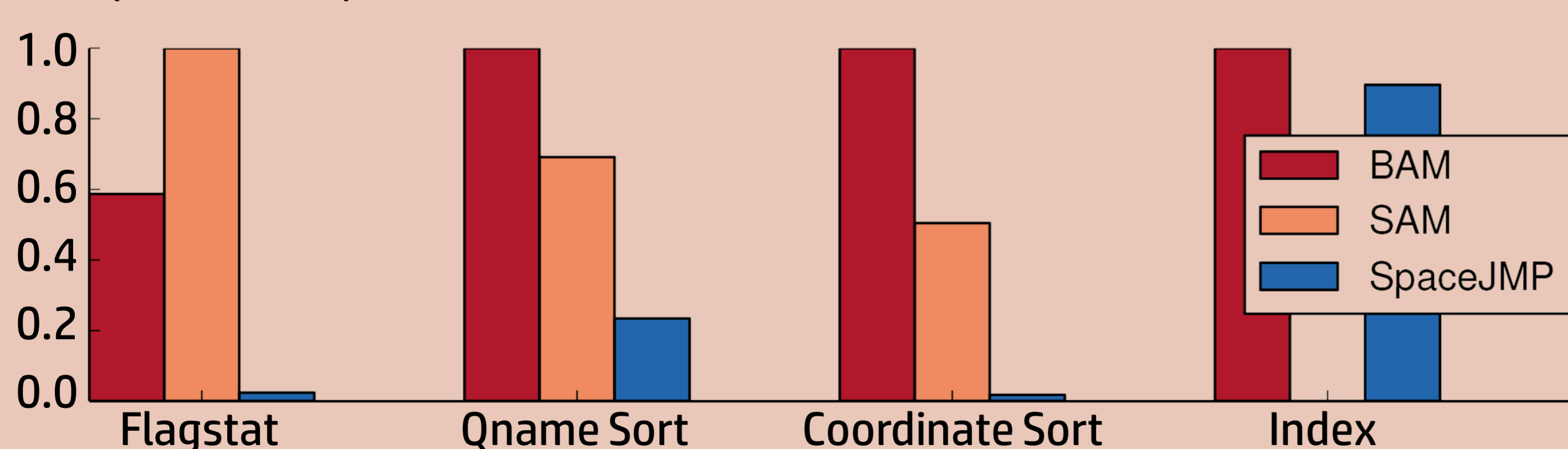


How to share pointer-rich data structures between compute stages?

SAMTools – Variant Calling (Genomics) Workflow

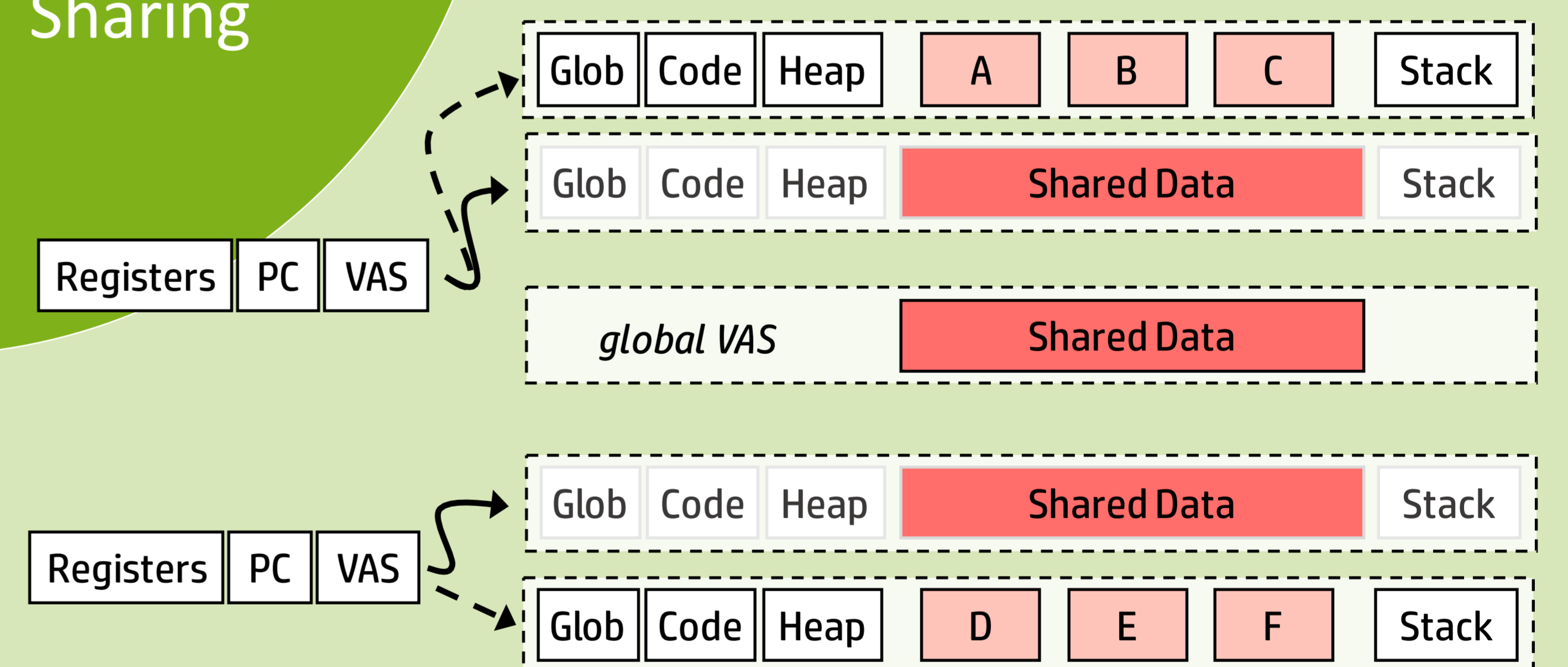
- SAM/BAM** – marshal data between stages to/from storage
- SpaceJMP** – data remains as-is and stages switch into VAS

Time (normalized)



Redesign Client-Server Model for Shared-Memory Systems

Data Sharing



How can we support shared use of common segments?

Redis – High-performance in-memory key-value store

- Standard** configuration – unix domain sockets
- SpaceJMP** – for RPC communication

Requests per second

