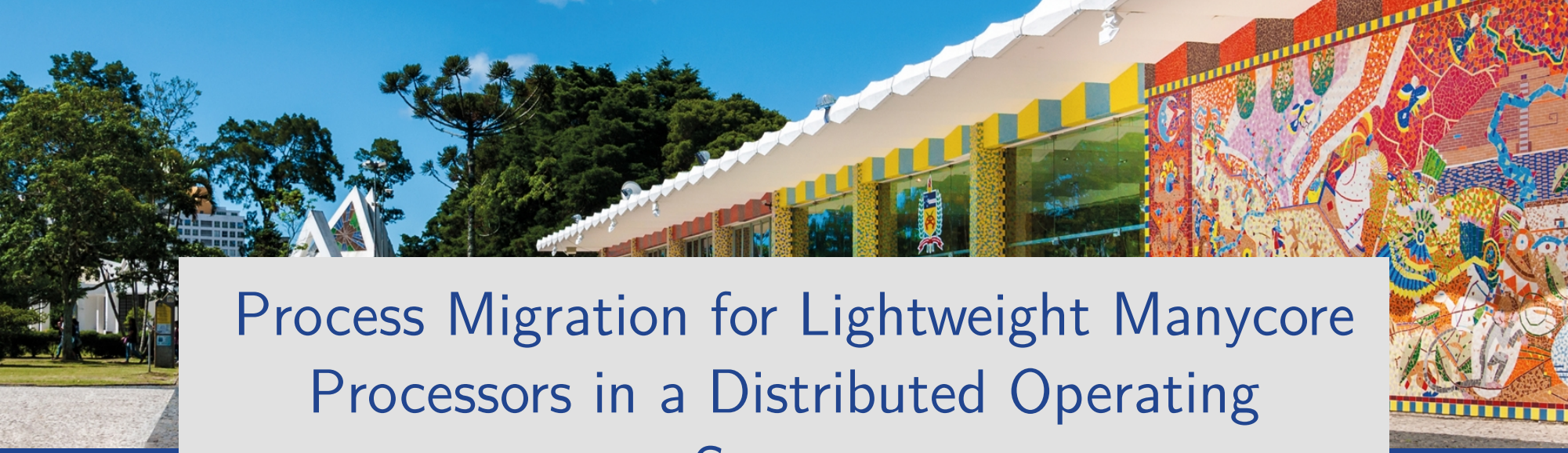


Process Migration for Lightweight Manycore Processors in a Distributed Operating System

João Vicente Souto

joao.vicente.souto@posgrad.ufsc.br

Parallel Computing - Computer Science
INE/UFSC, Florianópolis

September 29, 2020

LAPESD
INE UFSC

UFSC

Presentation Outline

1 Introduction

2 Virtualization

3 Migration
- Pre-Copy Migration
- Post-Copy Migration

4 Metrics

5 LW Processors
- Kalray MPPA-256

6 Motivation and Justification

LAPESD

Introduction

Process Migration

- **Transferring a process between machines**
- Proposed for:
  - Load balancing
  - Fault tolerance
  - System administration
  - Data access locality
- Arising of distributed systems
- Solutions established:
  - MOSIX (1985)
  - V (1988)
  - OSF/1 AD TNC (1995)

LAPESD

Process Migration

- Transferring a process between machines
- Proposed for:
  - Load balancing
  - Fault tolerance
  - Improved system administration
  - Data access locality
- Arising of **distributed systems**
- Solutions established:
  - MOSIX (1985)
  - V (1988)
  - OSF/1 AD TNC (1995)

LAPESD

Process Migration

- Despite the research efforts, **migration has not achieved widespread use**
- Treating residual dependency is a difficult task:
  - Open files stored in the source node
  - Open communicators with other processes
  - Shared resources
  - Internal kernel state.
- Solution: Virtualization

LAPESD

Process Migration

- Despite the research efforts, migration has not achieved widespread use
- **Treating residual dependency is a difficult task**:
  - Open files stored in the source node
  - Open communicators with other processes
  - Shared resources
  - Internal kernel state.
- Solution: Virtualization

LAPESD

Process Migration

- Despite the research efforts, migration has not achieved widespread use
- Treating residual dependency is a difficult task:
  - Open files stored in the source node
  - Open communicators with other processes
  - Shared resources
  - Internal kernel state.
- Solution: **Virtualization**

LAPESD

Virtualization

Virtualization

- Create a **Virtual Version** of Computing Resources
  - CPU
  - Memory
  - Storage device
  - Network device
- Features
  - Improved previous features
  - Power management
  - Security
- Two types
  - Virtual Machines (VM)
  - Containers

LAPESD

Virtualization

- Create a Virtual Version of Computing Resources
  - CPU
  - Memory
  - Storage device
  - Network device
- **Features**
  - Improved previous features
  - Power management
  - Security
- Two types
  - Virtual Machines (VM)
  - Containers

LAPESD

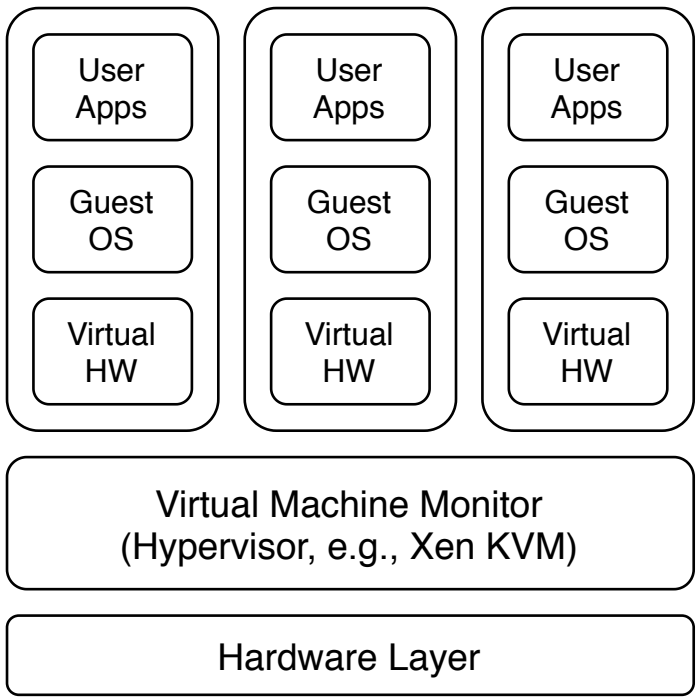
Virtualization

- Create a Virtual Version of Computing Resources
  - CPU
  - Memory
  - Storage device
  - Network device
- Features
  - Improved previous features
  - Power management
  - Security
- **Two types**
  - Virtual Machines (VM)
  - Containers

LAPESD

Virtual Machines (VM)

- Encapsulates a whole operating system
- Depends on a Hypervisor to provide the hardware resources
  - Full virtualization
  - Paravirtualization



LAPESD

Migration of VMs

- **Memory content**
  - Hypervisor allocated memory
  - Guest OS allocated memory
  - Application requested memory
- Disk content
  - Hypervisor allocated blocks
  - Guest OS used blocks
  - Application used blocks

LAPESD

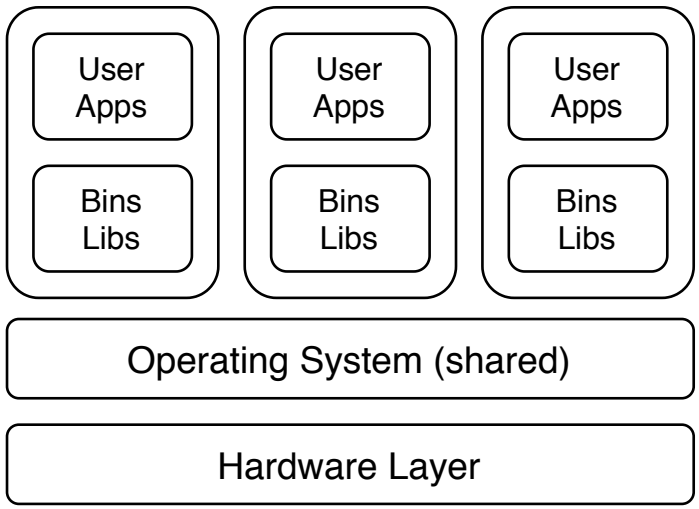
Migration of VMs

- Memory content
  - Hypervisor allocated memory
  - Guest OS allocated memory
  - Application requested memory
- **Disk content**
  - Hypervisor allocated blocks
  - Guest OS used blocks
  - Application used blocks

LAPESD

Containers

- OS-level virtualization
- Multiple isolated user space instances (Shared OS)
- Lightweight compared to VMs
  - Linux Containers
  - Docker



LAPESD

Migration of Containers

- Transfer the file system of the container
- Save the state of the container into a file (all processes and their resources)
- Transfer the container file
- Restart the container

LAPESD

Migration

Migration Techniques

- Generally, there are **three stages of memory transfer**:
  - Push Copy stage (iteration copy)
  - Stop-and-Copy stage
  - Pull Copy stage (on-demand copy)
- Migration techniques
  - Stop-and-Copy
  - *Pre-Copy*
  - *Post-Copy*
  - Hybrid

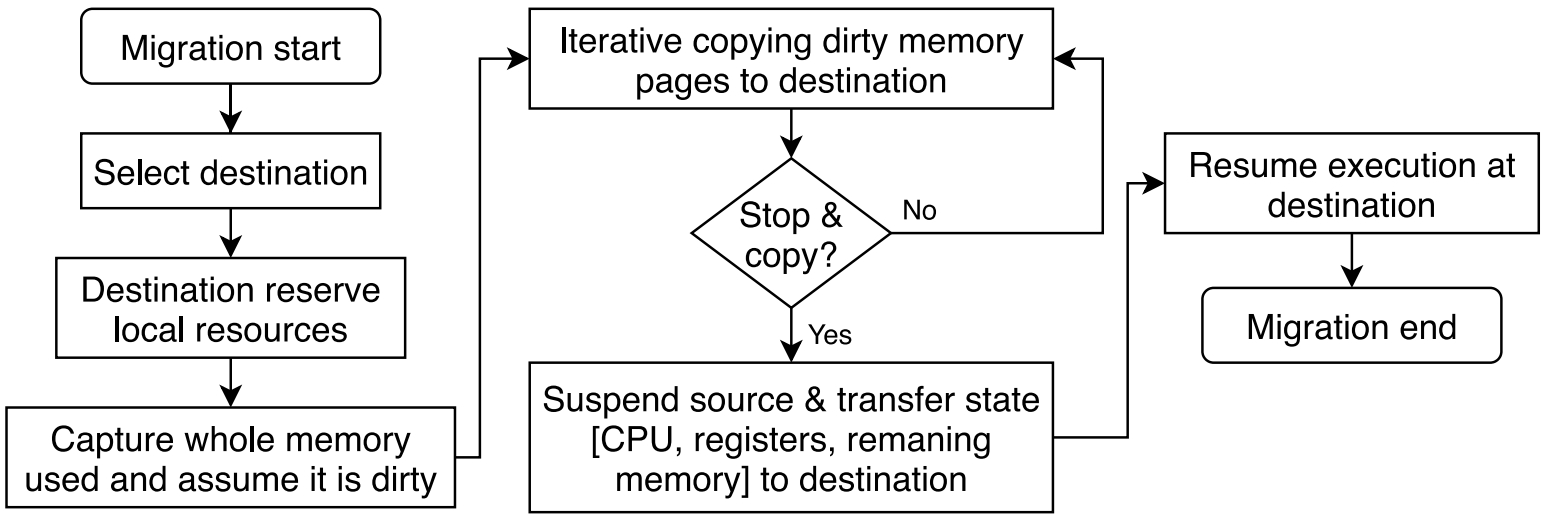
LAPESD

Migration Techniques

- Generally, there are three stages of memory transfer:
  - Push Copy stage (iteration copy)
  - Stop-and-Copy stage
  - Pull Copy stage (on-demand copy)
- **Migration techniques**:
  - Stop-and-Copy
  - ***Pre-Copy***
  - ***Post-Copy***
  - Hybrid

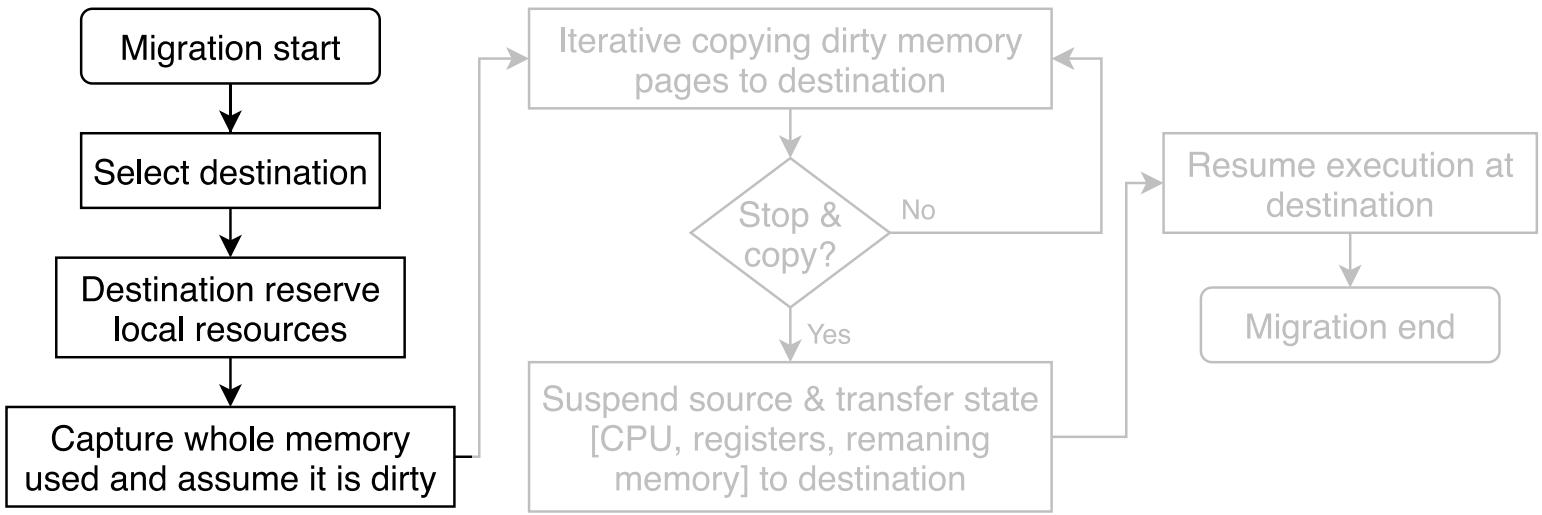
LAPESD

Pre-Copy Migration



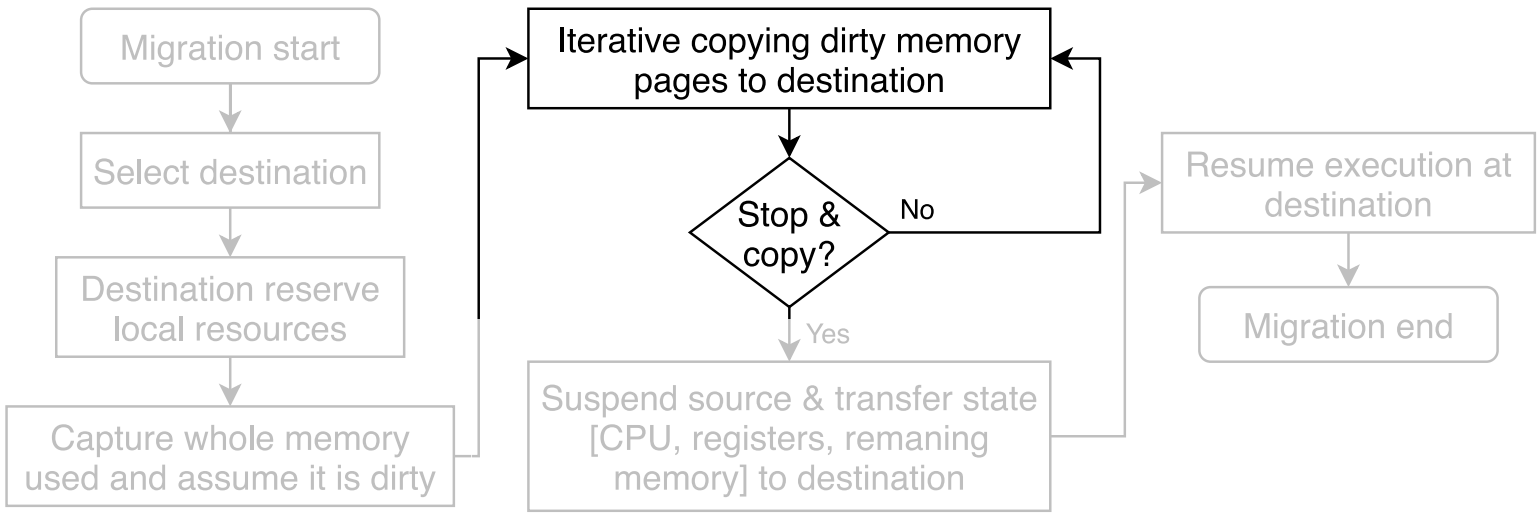
Pre-Copy Scenario

Pre-Copy Migration



Resource reservation and **transfer preparation**

Pre-Copy Migration

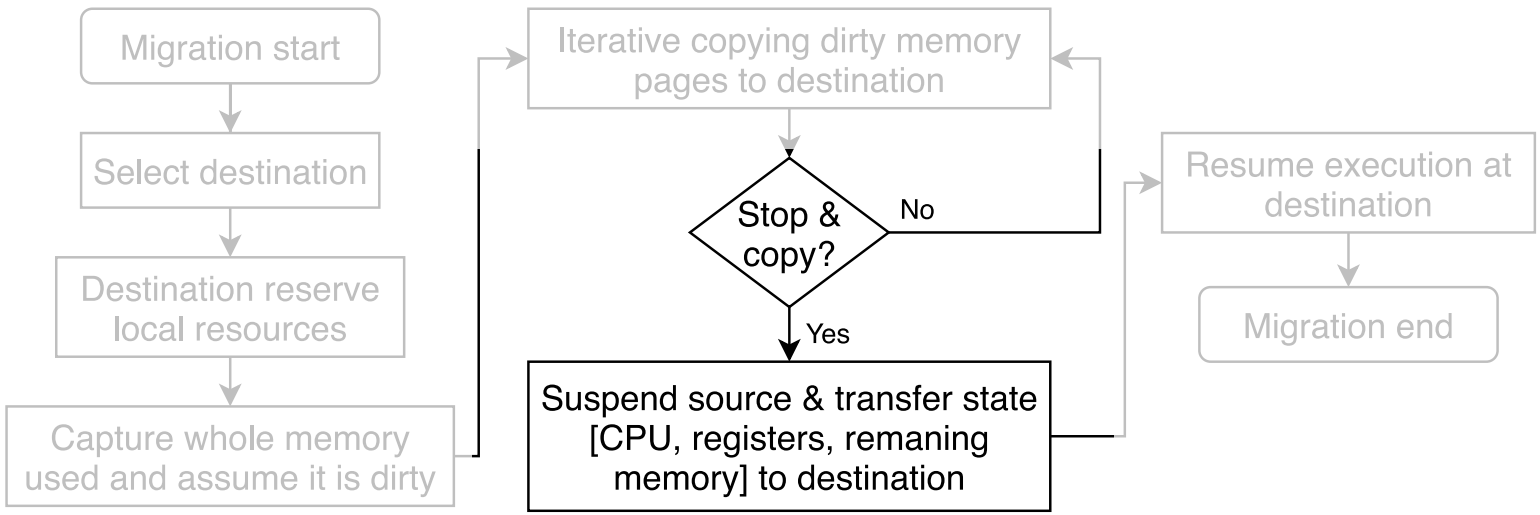


Sending the memory pages before the execution context

Push Copy Stage

LAPESD

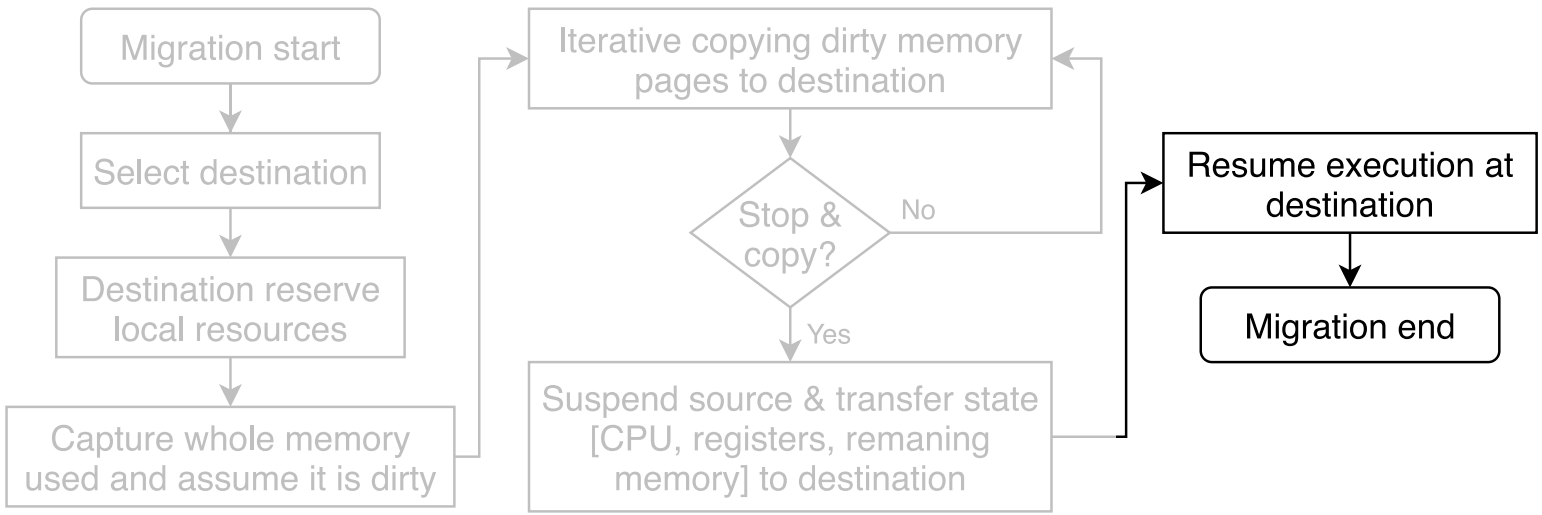
Pre-Copy Migration



When enough memory has been transferred, send the execution context

Stop-and-Copy Stage

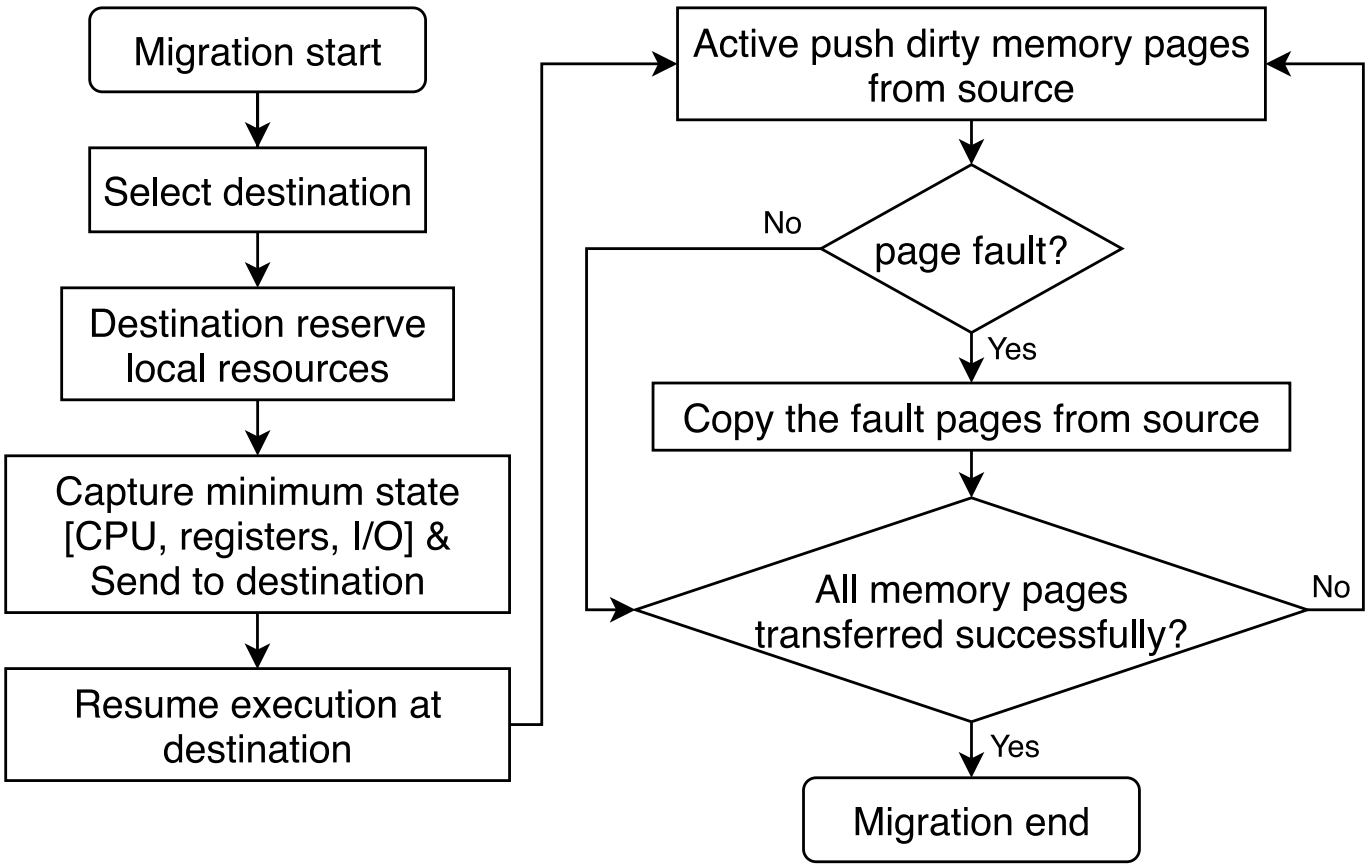
Pre-Copy Migration



Resume stopped execution context in the destination

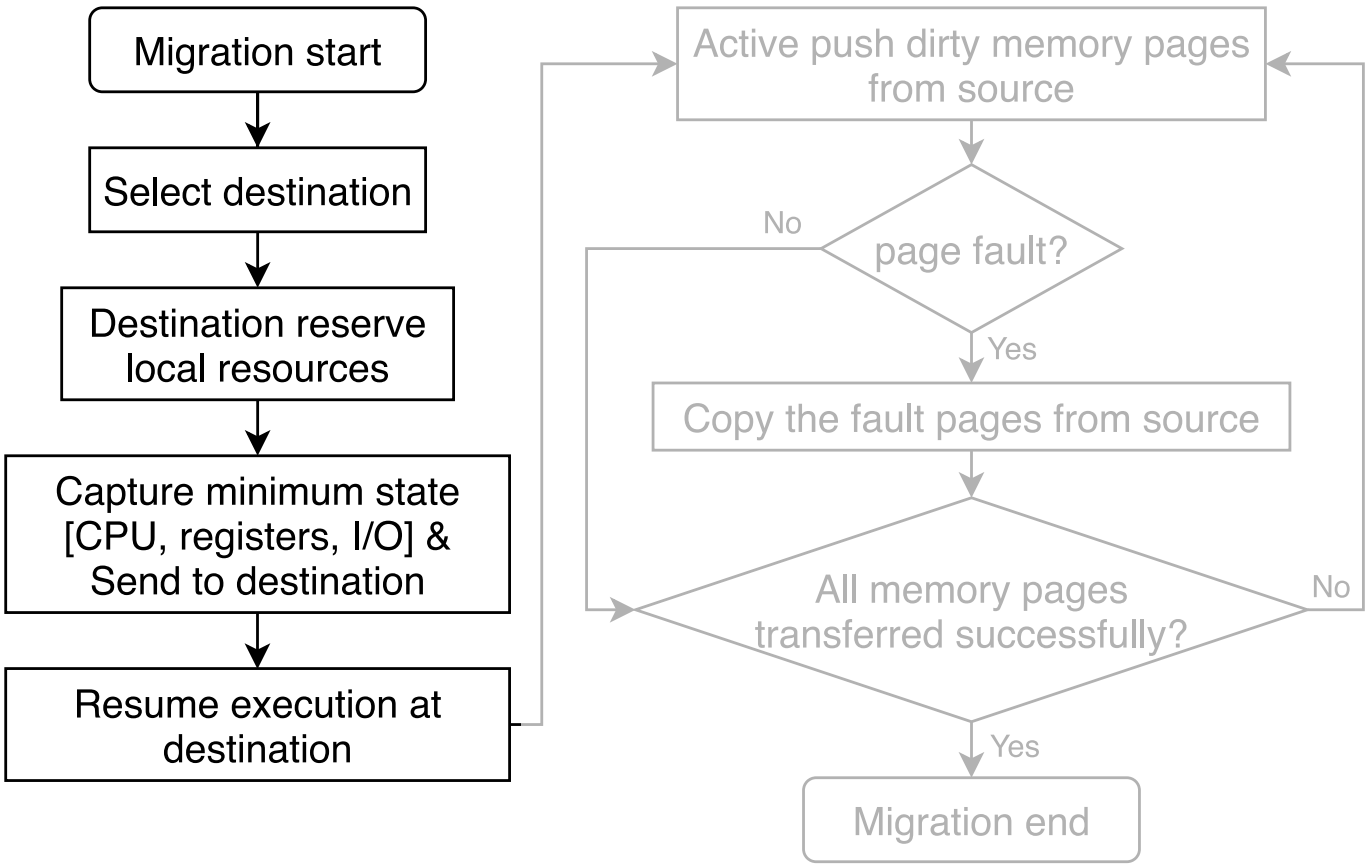
LAPESD

Post-Copy Migration



Post-Copy Scenario

Post-Copy Migration

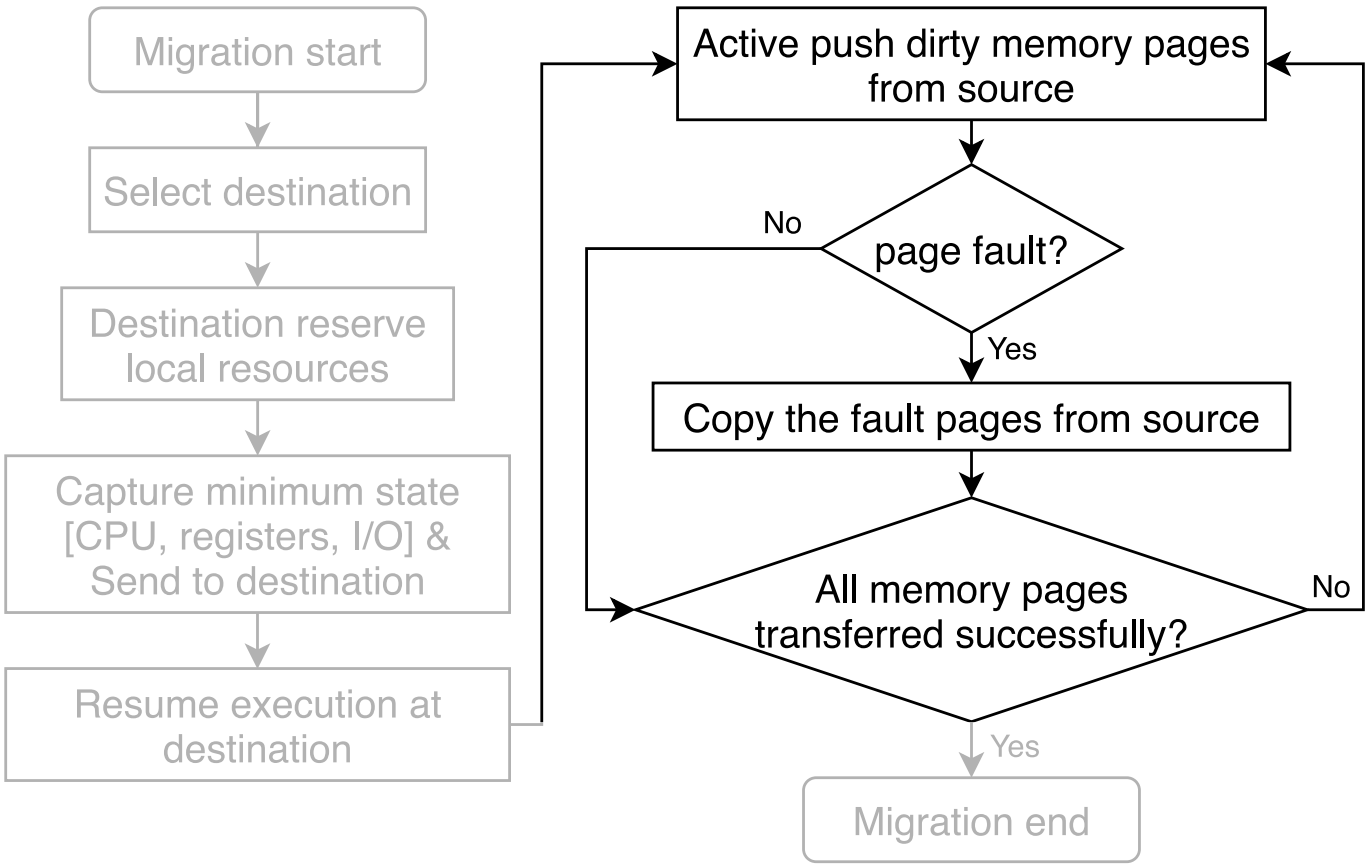


Sends execution context before memory pages

Stop-and-Copy Stage

LAPESD

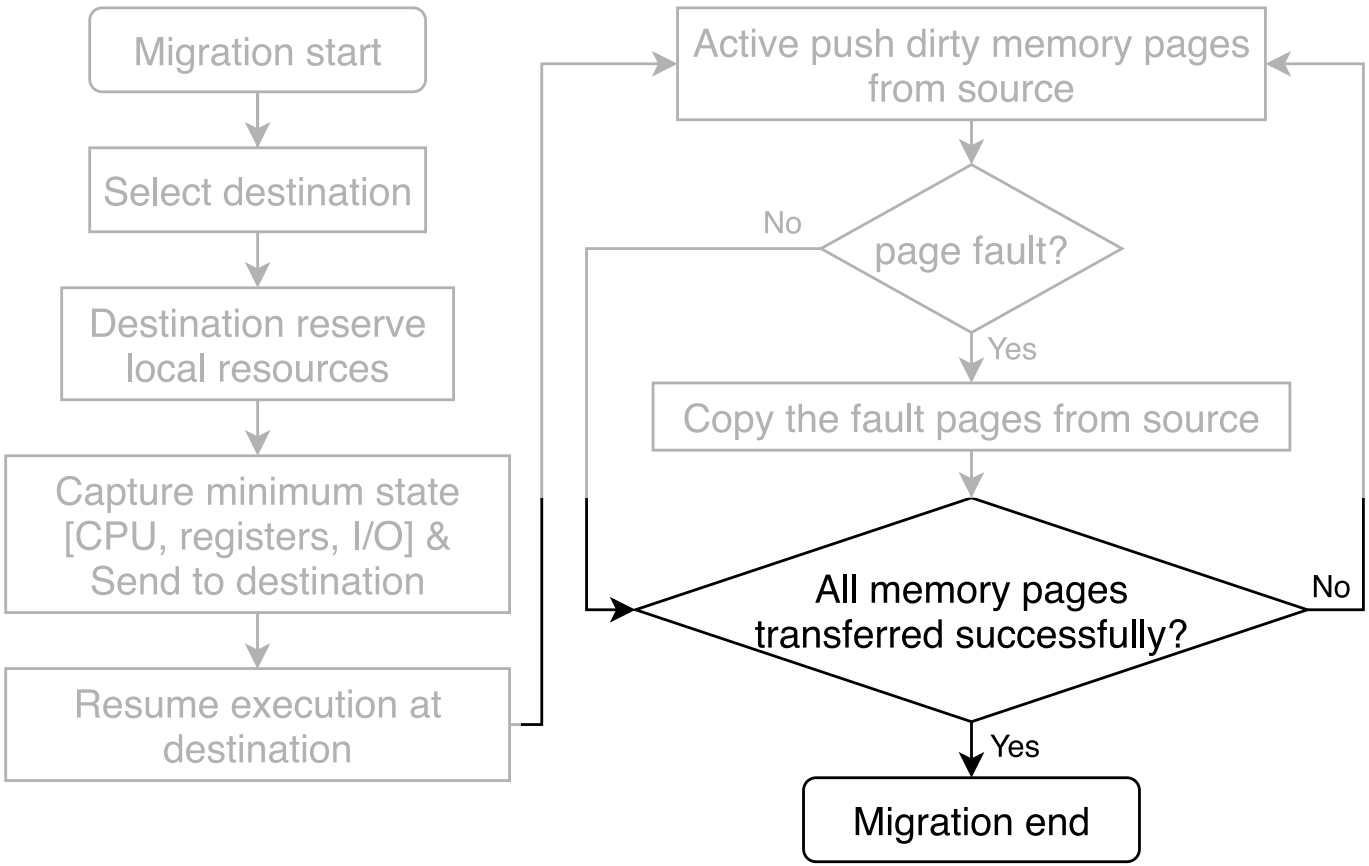
Post-Copy Migration



Merges **pull rest of memory pages** and any page that generate **page faults**

Pull Copy Stage

Post-Copy Migration



Continues execution normally

LAPESD

Metrics

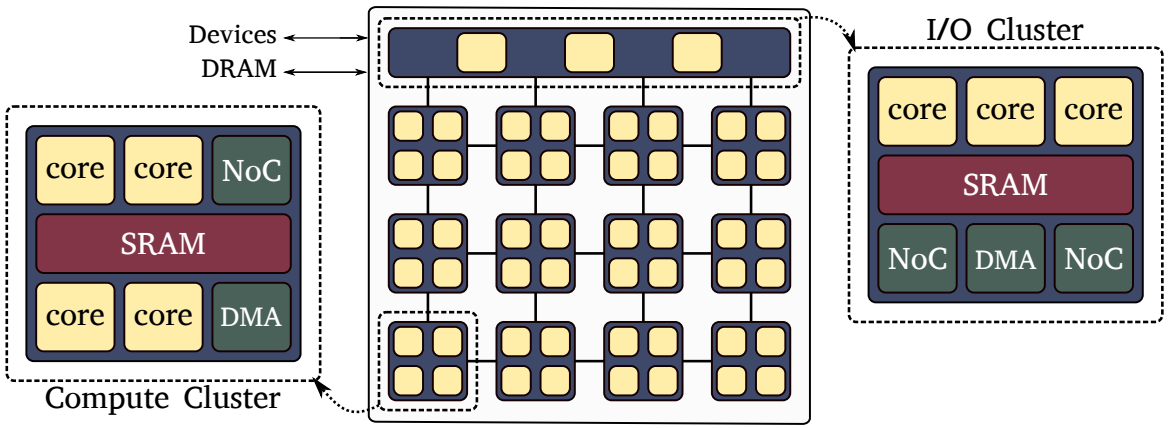
Performance Metrics

- Total Migration Time
- Downtime
- Pages Transfered
- Preparation Time
- Resume Time
- Application Degradation
- Page Dirty Rate
- Link Degradation

LAPESD

LW Processors

Lightweight Manycores Processors

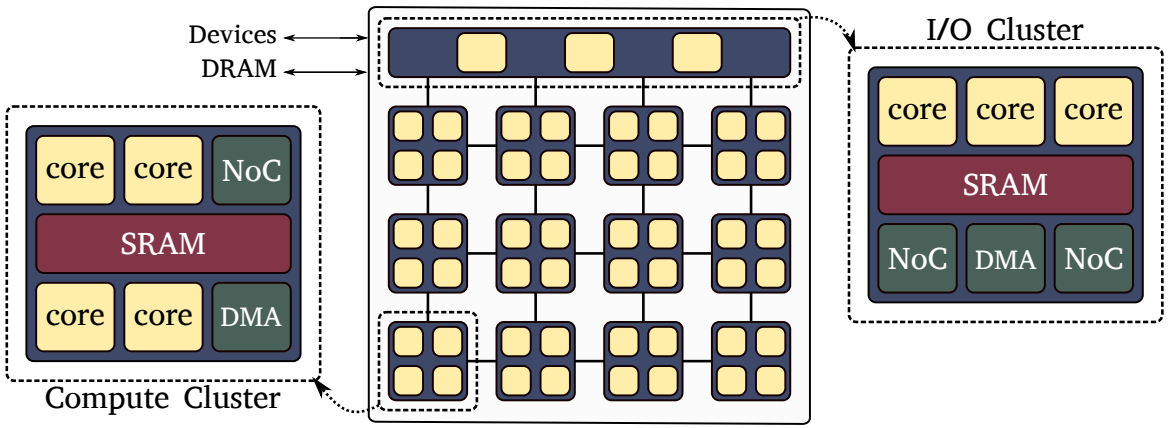


Overview of a Manycore

- **Hundreds of Lightweight Cores**
  - Expose Massive thread-level parallelism
  - Feature low-power consumption
  - Target MIMD workloads
- Distributed Memory Architecture
- On-Chip Heterogeneity

LAPESD

Lightweight Manycores Processors

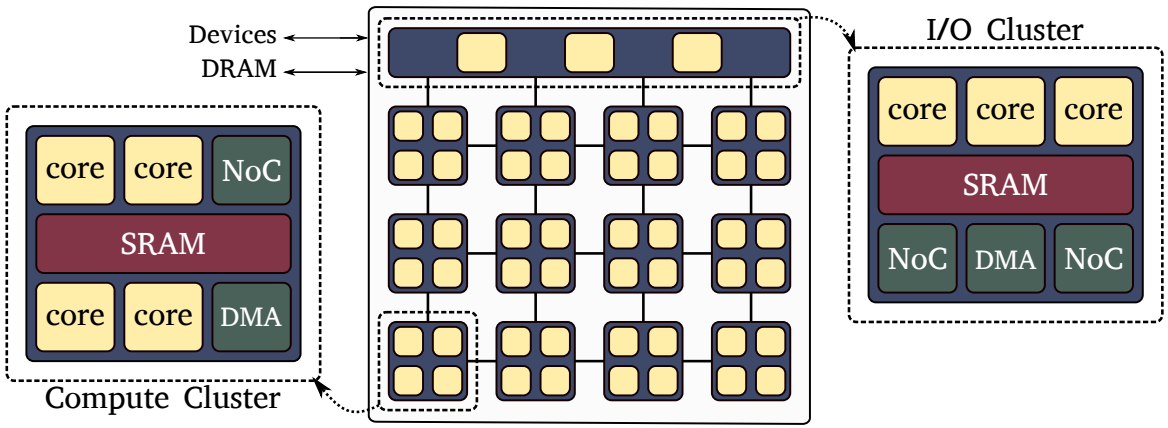


Overview of a Manycore

- Hundreds of Lightweight Cores
- **Distributed Memory Architecture**
  - Grants scalability
  - Relies on a Network-on-Chip (NoC)
  - Has constrained memory systems
- On-Chip Heterogeneity

LAPESD

Lightweight Manycores Processors



Overview of a Manycore

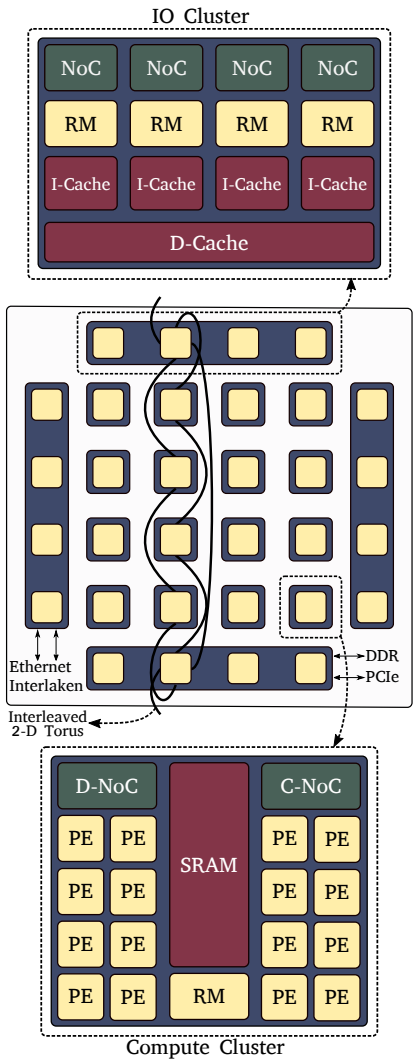
- Hundreds of Lightweight Cores
- Distributed Memory Architecture
- **On-Chip Heterogeneity**
  - Features different components

LAPESD

Kalray MPPA-256

A Lightweight Manycore Processor

- **288 processing cores**
  - 16 Compute Cluster (CC)
  - 4 I/O Cluster (IO)
- **Data NoC (D-NoC)**
  - 256 RX slots
  - 8 TX channels
  - 8 μthreads for async TX
- **Control NoC (C-NoC)**
  - 128 RX slots
  - 4 TX channels



Motivation and Justification

Motivation

- Bring the **benefits of migration and process virtualization to lightweight manycores**
- Load balancing
- Fault tolerance
- System administration
- Data locality
- Power management
- Security

LAPESD

Motivation

- Bring the benefits of migration and process virtualization to lightweight manycores
- **Load balancing**
  - Better work distribution
- Fault tolerance
- System administration
- Data locality
- Power management
- Security

LAPESD

Motivation

- Bring the benefits of migration and process virtualization to lightweight manycores
- Load balancing
- **Fault tolerance**
  - Save/restore processes state
- System administration
- Data locality
- Power management
- Security

LAPESD

Motivation

- Bring the benefits of migration and process virtualization to lightweight manycores
- Load balancing
- Fault tolerance
- **System administration**
  - Allow multiuser execution and management
- Data locality
- Power management
- Security

LAPESD

Motivation

- Bring the benefits of migration and process virtualization to lightweight manycores
- Load balancing
- Fault tolerance
- System administration
- **Data locality**
  - Move processes to closer to the data source
- Power management
- Security

LAPESD

Motivation

- Bring the benefits of migration and process virtualization to lightweight manycores
- Load balancing
- Fault tolerance
- System administration
- Data locality
- **Power management**
  - Separate CPU-bound processes to improve heat dissipation
- Security

LAPESD

Motivation

- Bring the benefits of migration and process virtualization to lightweight manycores
- Load balancing
- Fault tolerance
- System administration
- Data locality
- Power management
- **Security**
  - Better isolation and kill suspicious processes

LAPESD

Justification

- **Parallel and distributed nature** of lightweight manycores
- Memory restrictions do not support a complete virtual environment like VMs
- Explore lighter forms of virtualization as at the OS level
- Integrate virtualization into a distributed OS designed for lightweight manycores

LAPESD

Justification

- Parallel and distributed nature of lightweight manycores
- **Memory restrictions** do not support a complete virtual environment like VMs
- Explore lighter forms of virtualization as at the OS level
- Integrate virtualization into a distributed OS designed for lightweight manycores

LAPESD

Justification

- Parallel and distributed nature of lightweight manycores
- Memory restrictions do not support a complete virtual environment like VMs
- Explore **lighter forms of virtualization**, e.g., OS-level
- Integrate virtualization into a distributed OS designed for lightweight manycores

LAPESD

Justification

- Parallel and distributed nature of lightweight manycores
- Memory restrictions do not support a complete virtual environment like VMs
- Explore lighter forms of virtualization, e.g., OS-level
- Integrate virtualization into a **distributed OS designed for lightweight manycores**

LAPESD

Thank you!
Questions?

João Vicente Souto

joao.vicente.souto@posgrad.ufsc.br

Parallel Computing - Computer Science
INE/UFSC, Florianópolis

September 29, 2020

LAPESD
INE UFSC

UFSC

References I

 CHOUDHARY, A. et al. A critical survey of live virtual machine migration techniques. *Journal of Cloud Computing*, v. 6, n. 1, p. 23, 2017. ISSN 2192-113X. Disponível em: <https://doi.org/10.1186/s13677-017-0092-1>.

 CLARK, C. et al. Live migration of virtual machines. In: *Proceedings of the 2nd Conference on Symposium on Networked Systems Design amp; Implementation - Volume 2*. USA: USENIX Association, 2005. (NSDI'05), p. 273–286.

 Hosseini Shirvani, M.; RAHMANI, A. M.; SAHAFI, A. A survey study on virtual machine migration and server consolidation techniques in dvfs-enabled cloud datacenter: Taxonomy and challenges. *Journal of King Saud University - Computer and Information Sciences*, v. 32, n. 3, p. 267 – 286, 2020. ISSN 1319-1578. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1319157818302842>.

 MILOJIcIć, D. S. et al. Process migration. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 32, n. 3, p. 241–299, set. 2000. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/367701.367728>.

LAPESD

References II

 Tao, Z. et al. A survey of virtual machine management in edge computing. *Proceedings of the IEEE*, v. 107, n. 8, p. 1482–1499, 2019.

 WANG, Z. et al. Ada-things: An adaptive virtual machine monitoring and migration strategy for internet of things applications. *Journal of Parallel and Distributed Computing*, v. 132, p. 164 – 176, 2019. ISSN 0743-7315. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0743731518304404>.

LAPESD

Citations

- (MILOJIclć et al., 2000)
- (CLARK et al., 2005)
- (CHOUDHARY et al., 2017)
- (Tao et al., 2019)
- (Hosseini Shirvani; RAHMANI; SAHAFI, 2020)
- (WANG et al., 2019)

LAPESD