Programming Agent-Based Demographic Models with Cross-State and Message-Exchange Dependencies: A Study with Speculative PDES and Automatic Load-Sharing

Alessandro Pellegrini
Cristina Montañola-Sales
Francesco Quaglia
Josep Casanovas-García

Sapienza, University of Rome
Barcelona Supercomputing Center

WSC 2016

Context

- We study the interactions of ABM run on top of PDES Systems
- We concentrate on shared-memory multicore systems

- The average degree of interaction in ABM can be significantly higher than in other scenarios
- We want to be able to detect these interactions
- We want to exploit them to increase the overall performance

- Interactions can be explicit or implicit (thanks to ECS)

Event Cross-State Synchronization [PADS 2014]

- Event-Cross State Synchronization (ECS) allows multiple LPs to exchange information via in-place memory accesses

- LPs do not need to be disjoint entities anymore
- Exchange of large amount of data is faster
- Allows for a simpler programming model

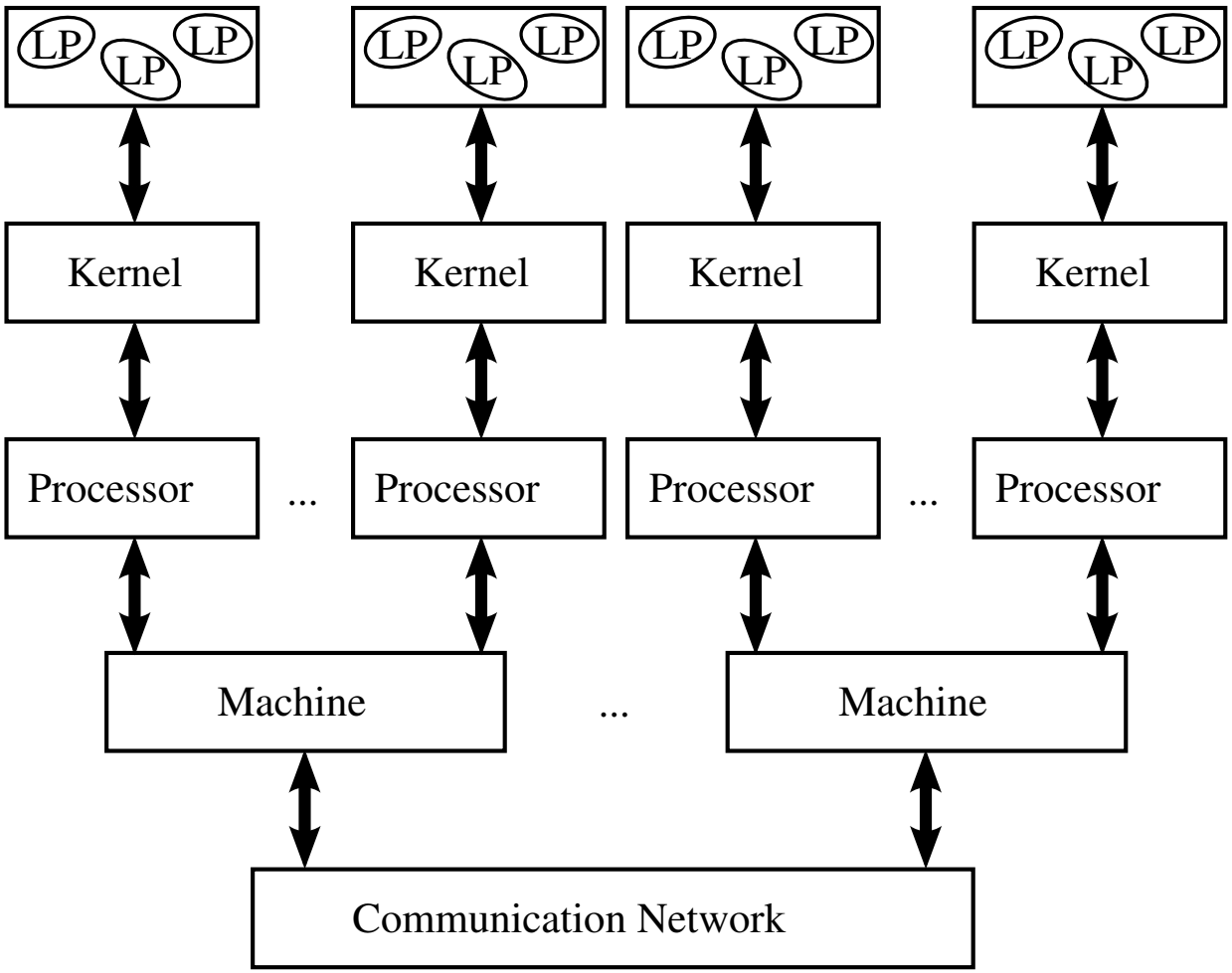
- Syncrhonization with ECS can be costly
- This is even more the case when LPs cross-synchronize a lot
- Load-Sharing can significantly reduce the cost

Contributions

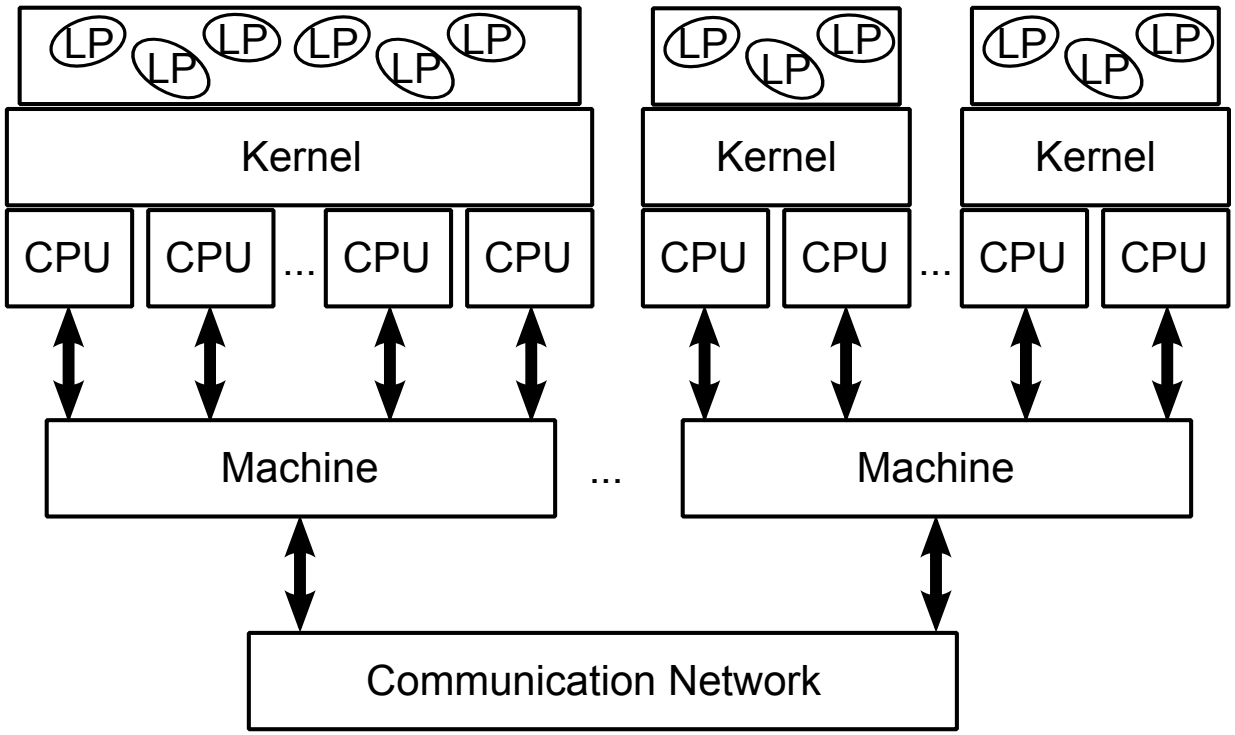
1. We clearly specify agent-based programming guidelines for demographic simulations to be run on top of PDES
   - This is done by proposing a general skeleton model

2. We tackle the complex interaction pattern often exhibited by ABM, to enforce load sharing
   - Performance can be increased
   - All types of interactions should be captured

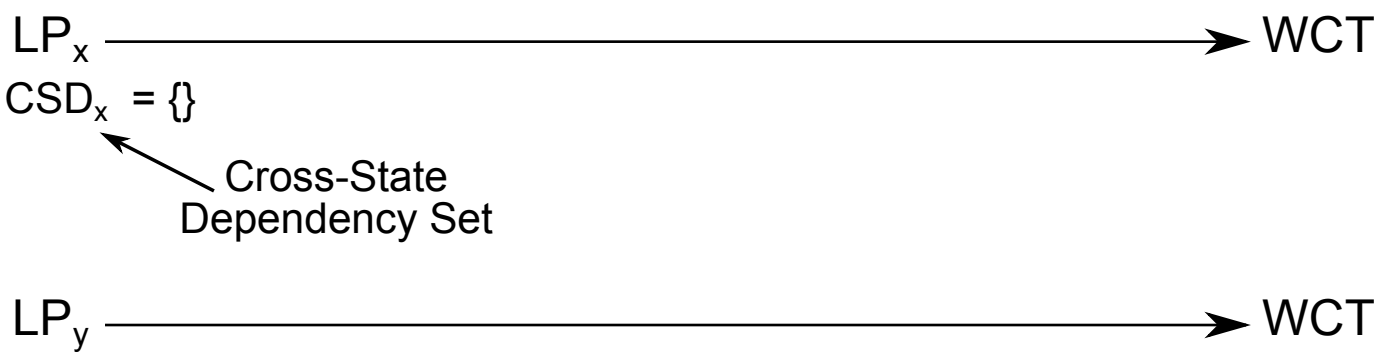
Reference Target Architecture



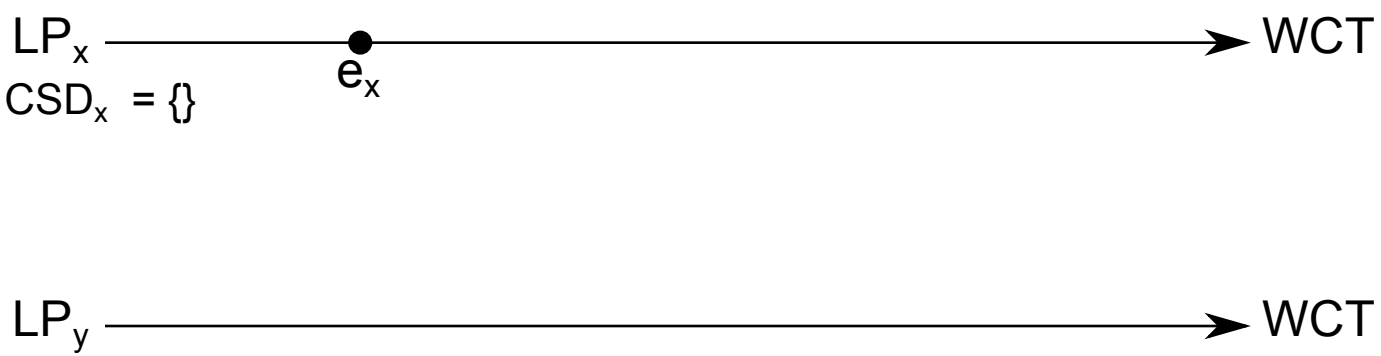
Reference Target Architecture



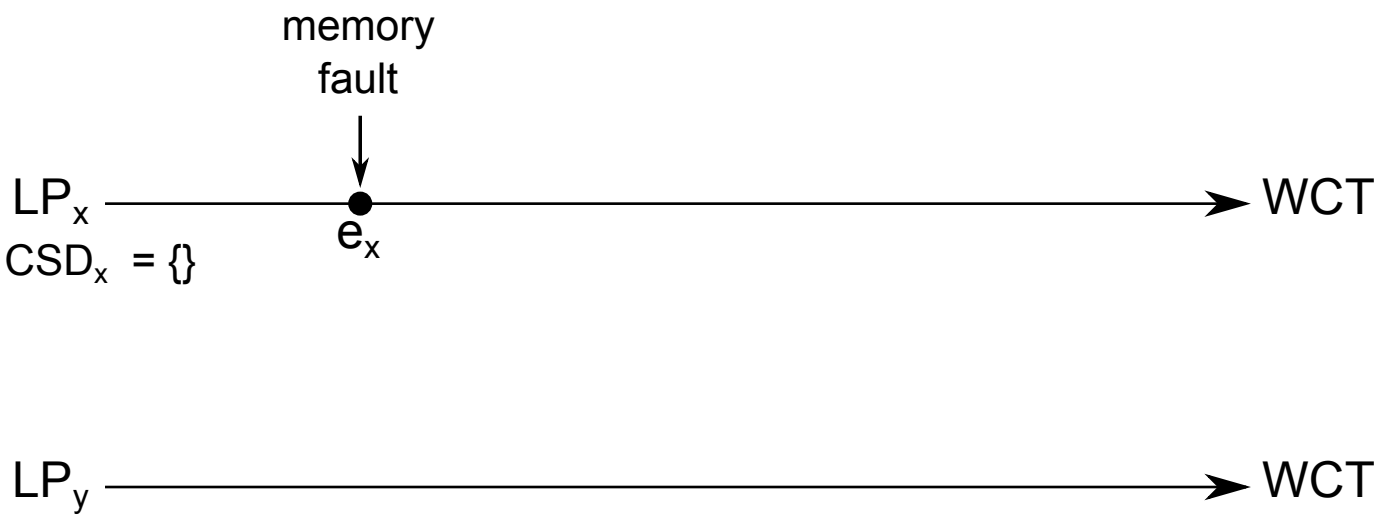
Event Cross-State Synchronization [PADS 2014]



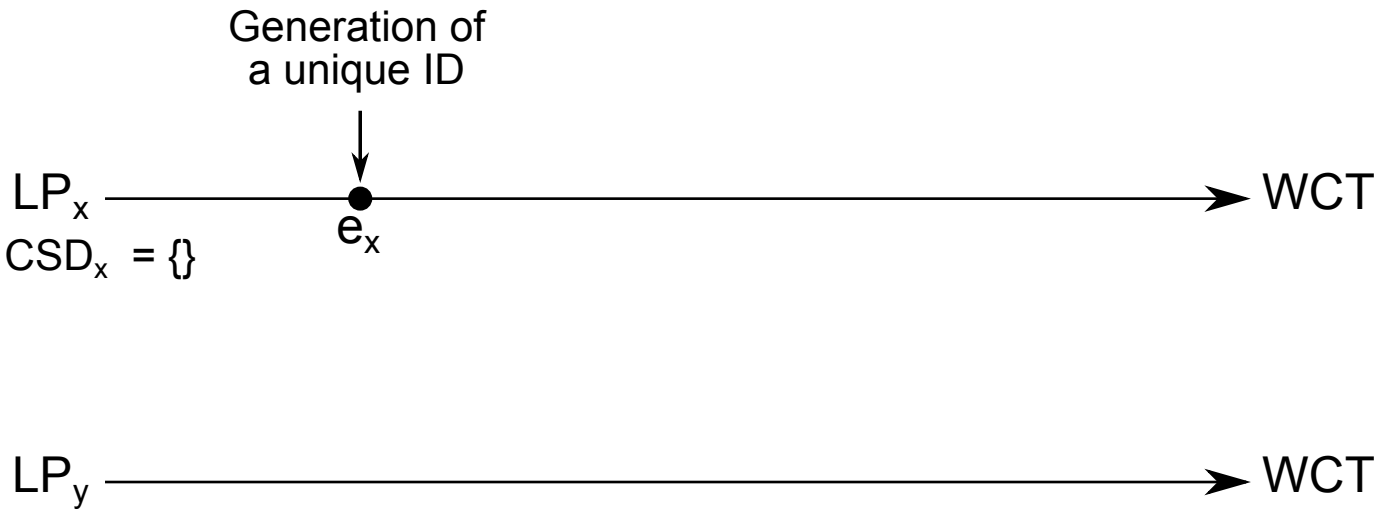
Event Cross-State Synchronization [PADS 2014]



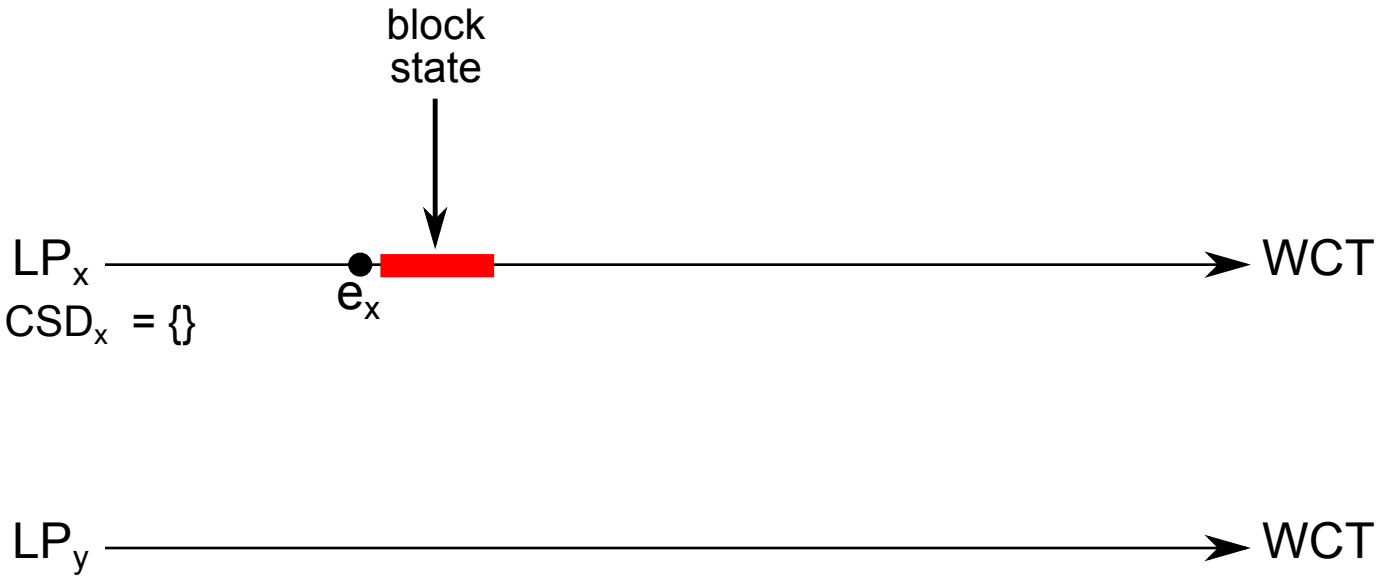
Event Cross-State Synchronization [PADS 2014]



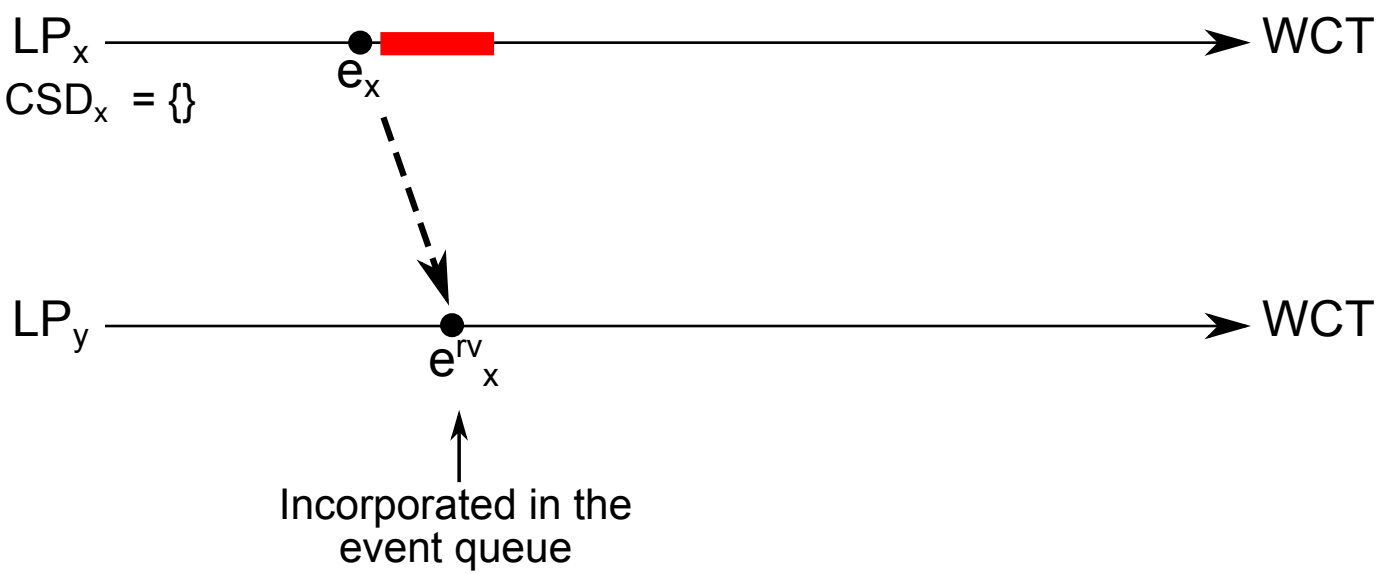
Event Cross-State Synchronization [PADS 2014]



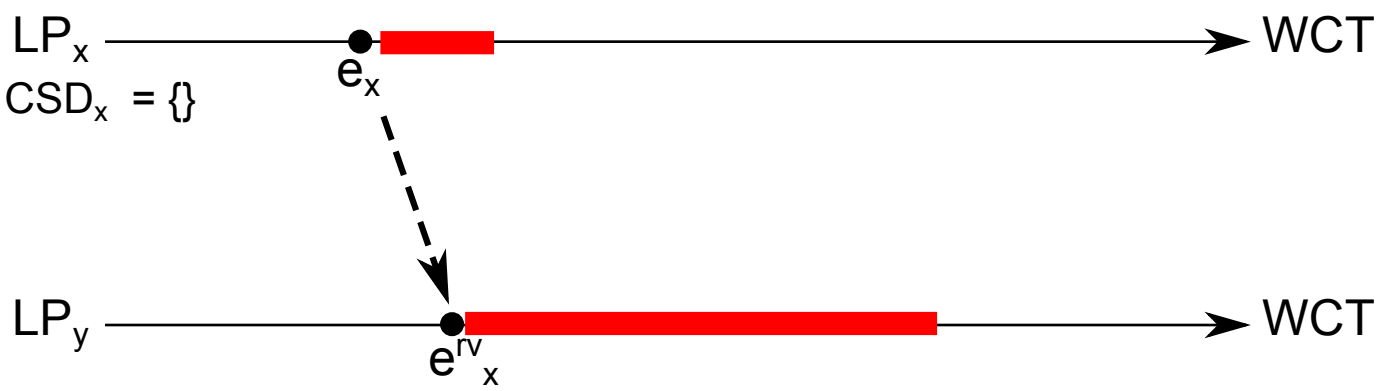
Event Cross-State Synchronization [PADS 2014]



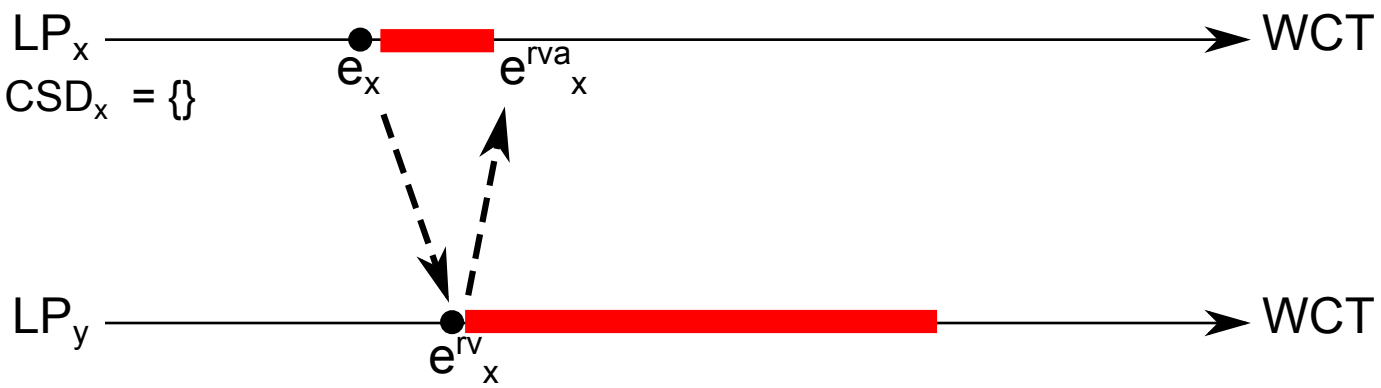
Event Cross-State Synchronization [PADS 2014]



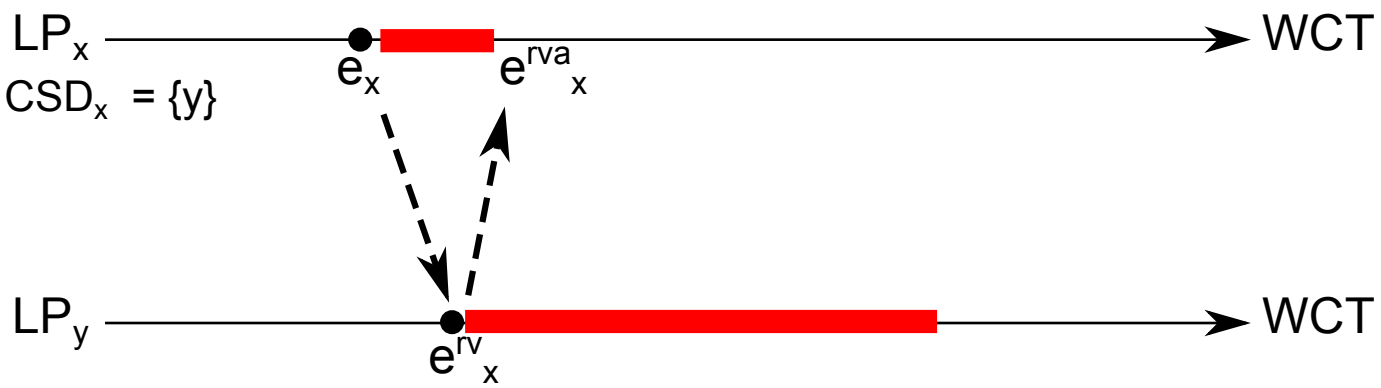
Event Cross-State Synchronization [PADS 2014]



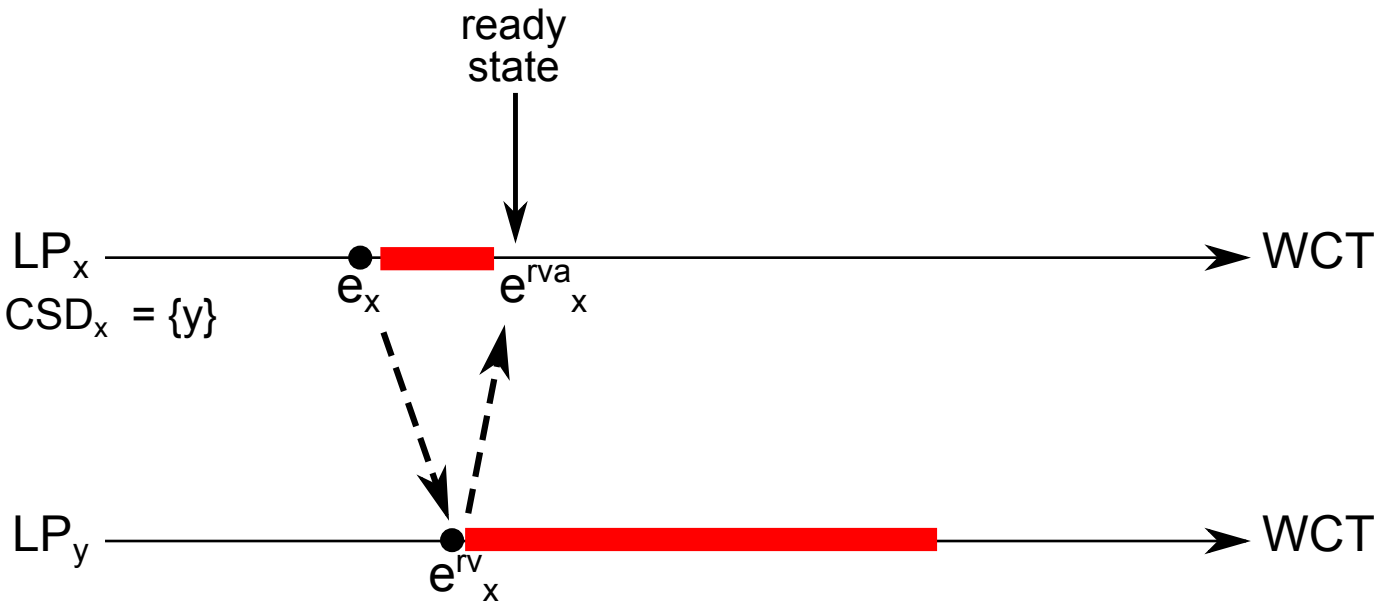
Event Cross-State Synchronization [PADS 2014]



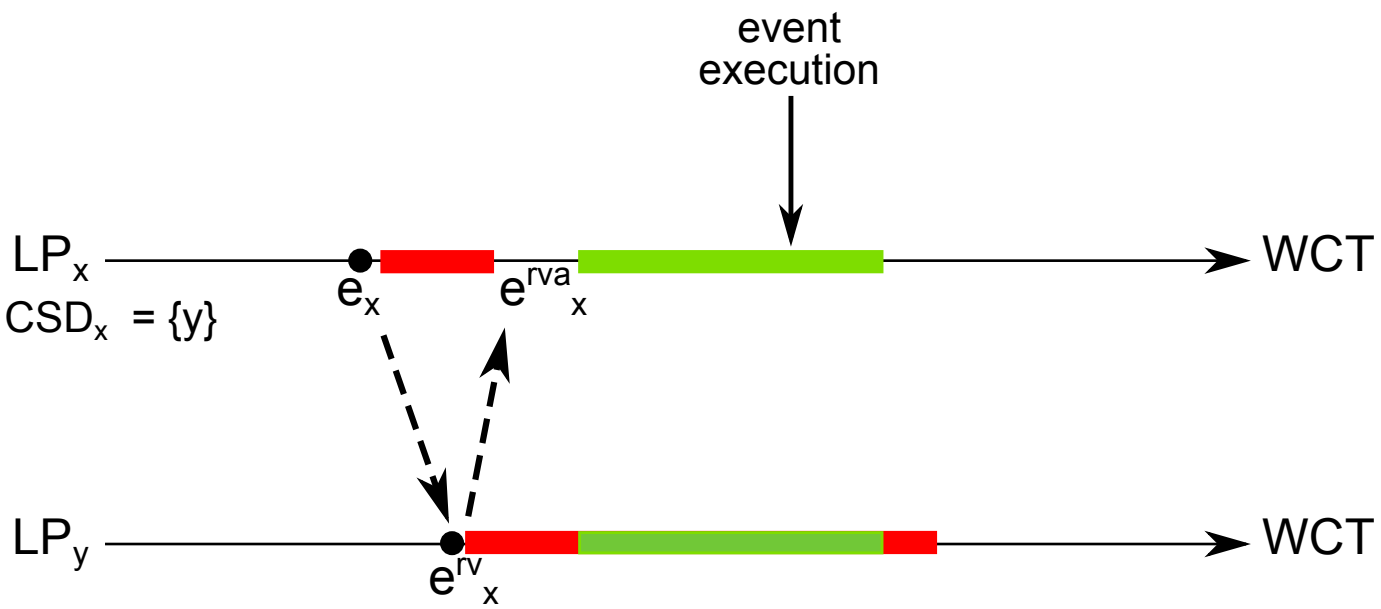
Event Cross-State Synchronization [PADS 2014]



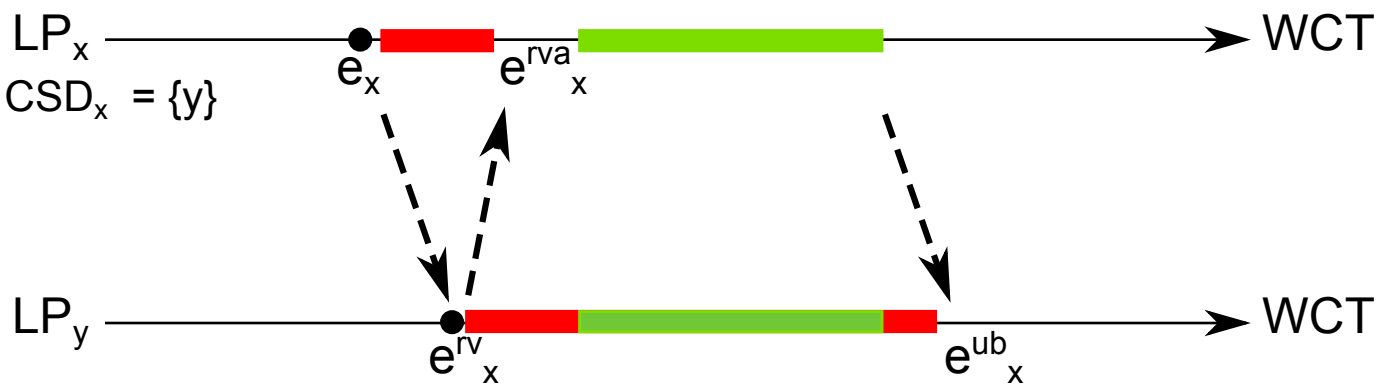
Event Cross-State Synchronization [PADS 2014]



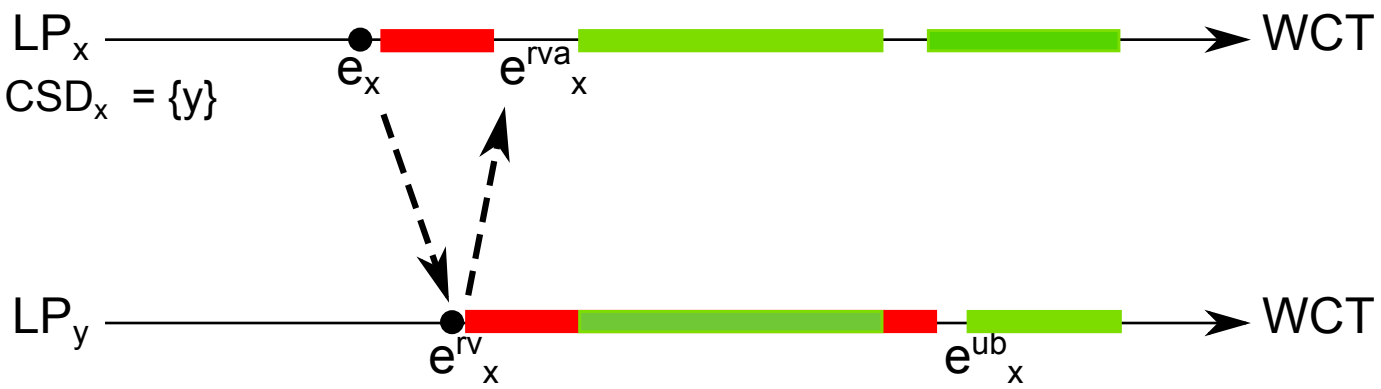
Event Cross-State Synchronization [PADS 2014]



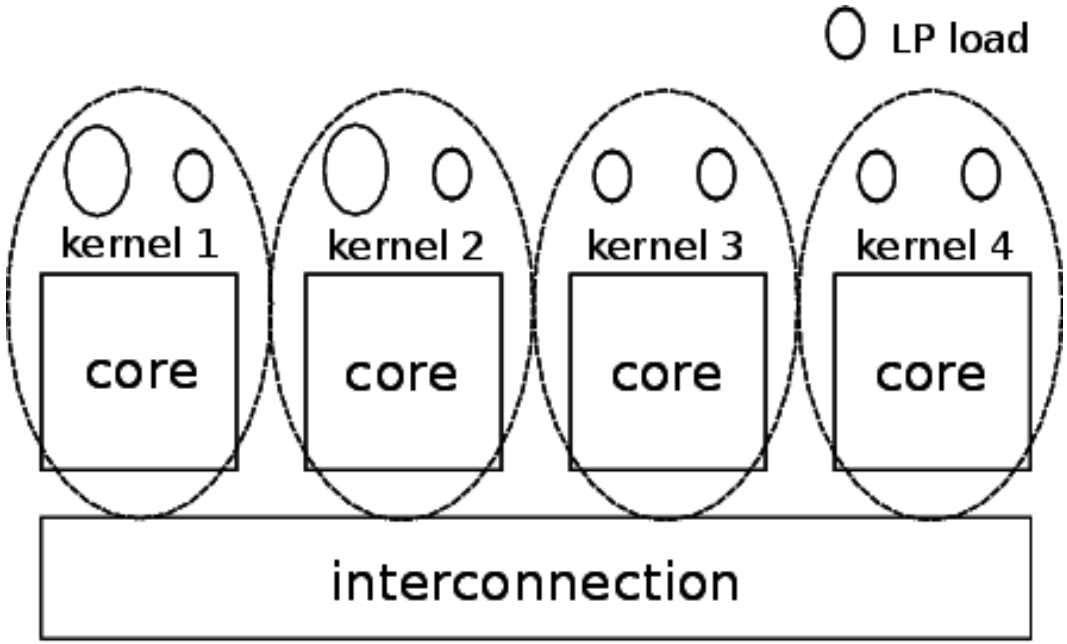
Event Cross-State Synchronization [PADS 2014]



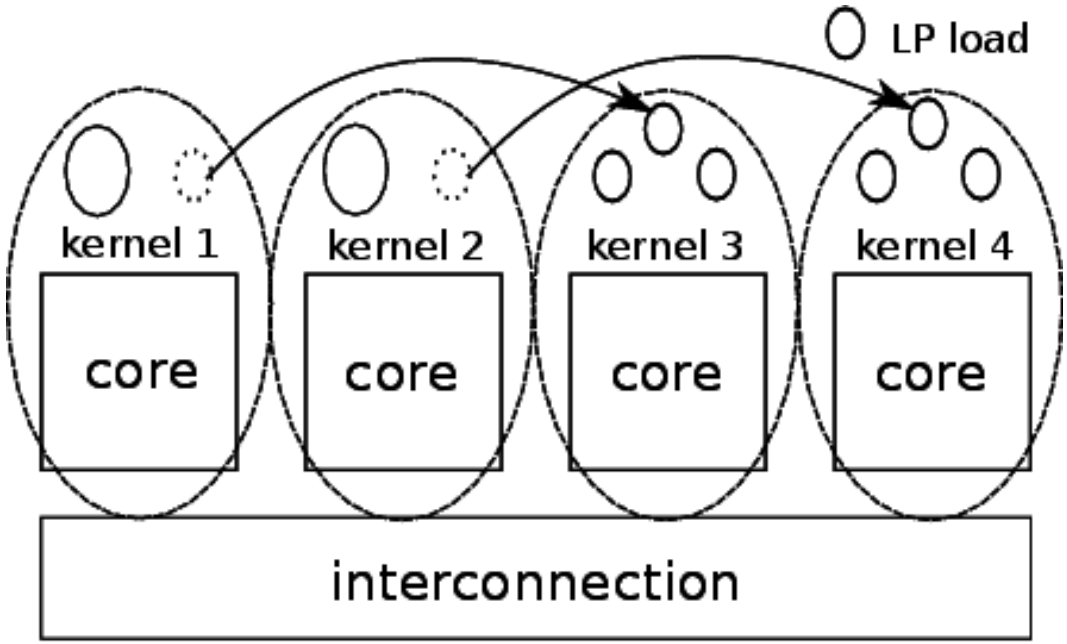
Event Cross-State Synchronization [PADS 2014]



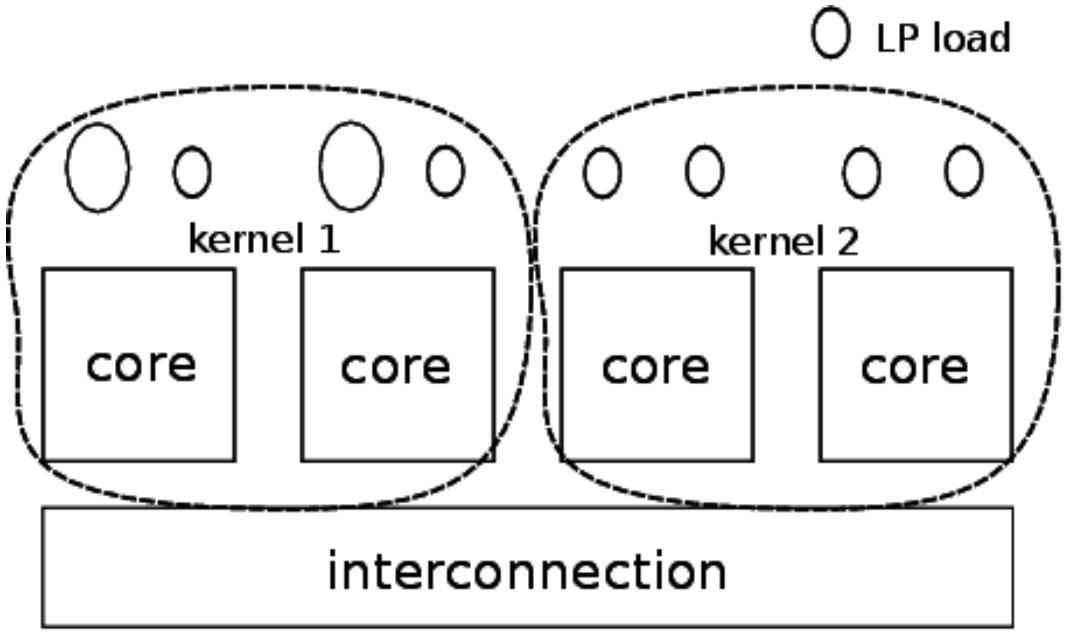
Load Sharing vs. Load Balancing



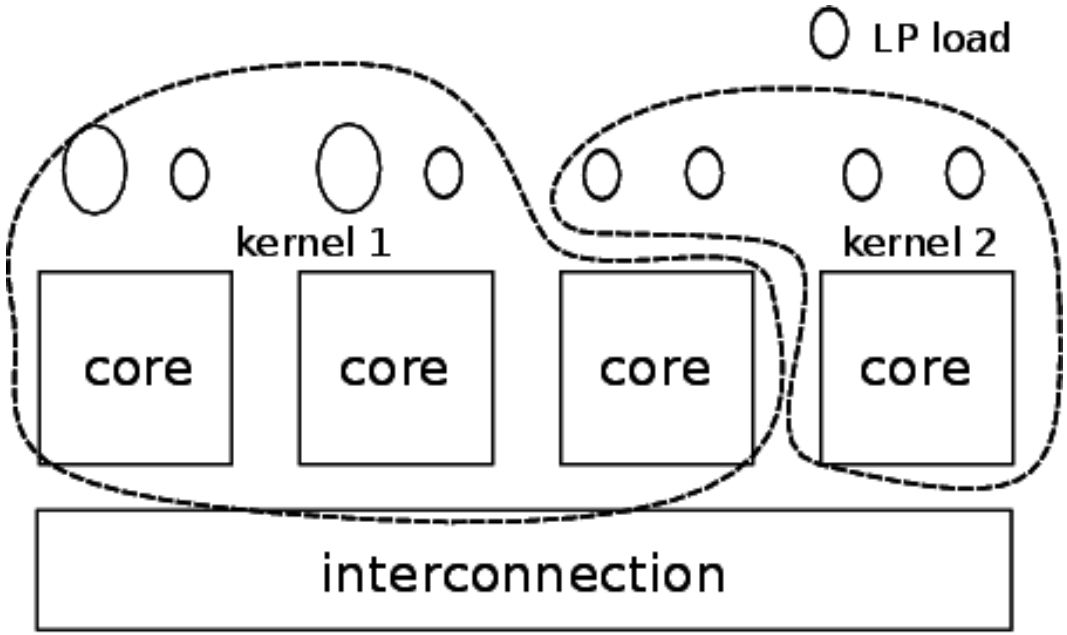
Load Sharing vs. Load Balancing



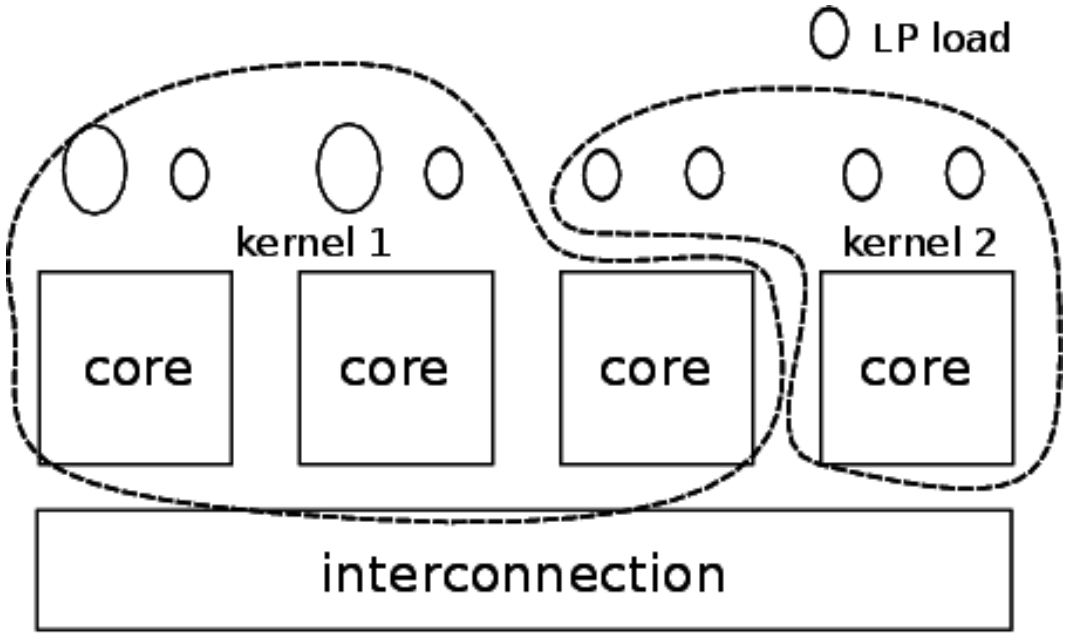
Load Sharing vs. Load Balancing



Load Sharing vs. Load Balancing



Load Sharing vs. Load Balancing



How to decide upon the binding?

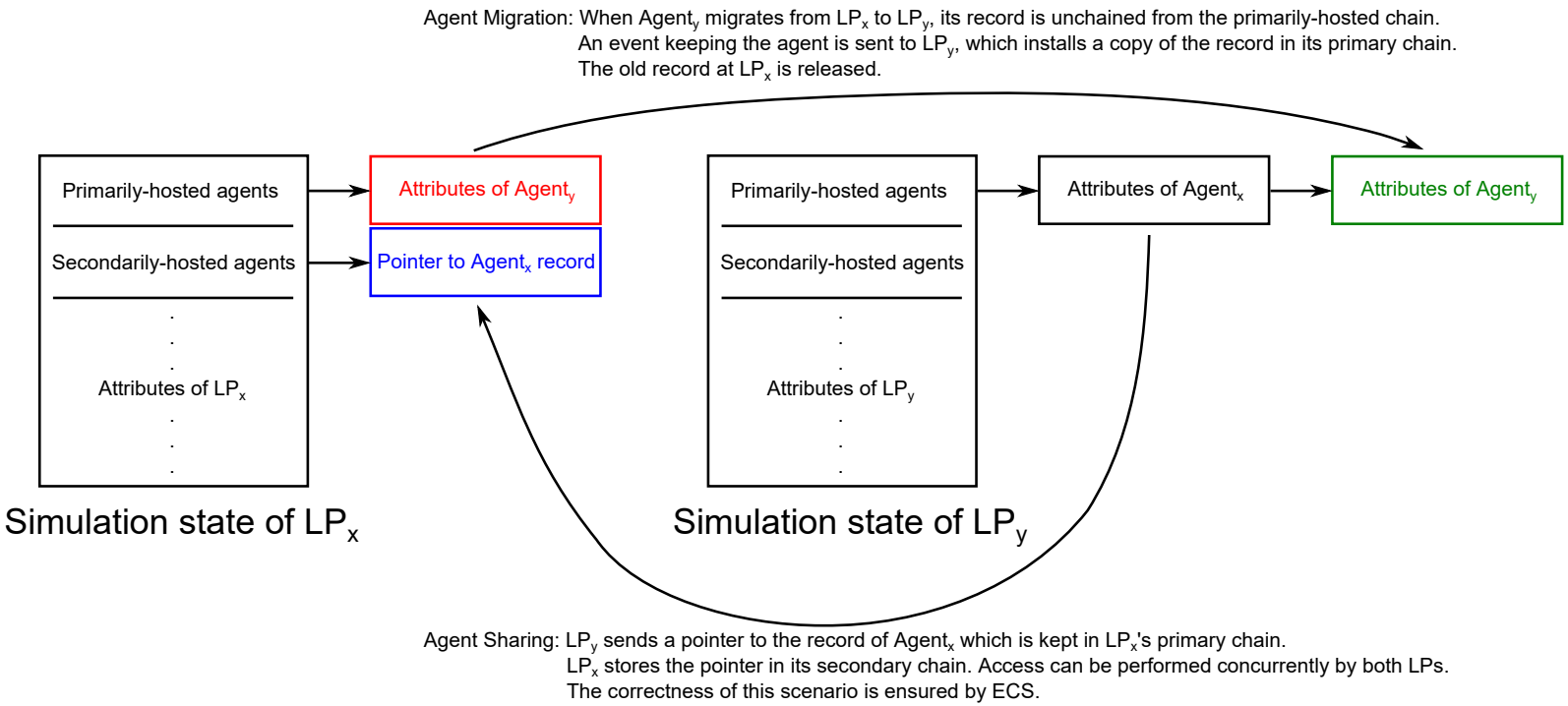
Programming guidelines for Demography on PDES

- The core element is the life course of *individuals*
- Behaviour and decisions depend on the *environment*

- Individuals (*agents*) are mapped to `structs`
- Portions of the environment (*regions*) are mapped to LPs

- Event handlers at LPs implement the logic to manipulate individuals
  - An agent is described in terms of individual-specific explanatory variables
  - State transitions define agent *histories*
- Message passing is used to migrate agents (as the payload)

ABM programming guidelines on PDES



- Easy local interaction
- Easy remote interaction, via direct memory access

Load-Sharing Policy

- The goal is to cluster together LPs that interact most
- For each LP_i we rely on counters:
  - The number of implicit interactions I_j with LP_j
  - The number of explicit interactions E_j with LP_j
- Each LP is thus associated with a tuple:

$$\langle I_0, I_1, \ldots, I_{maxLP-1}, E_0, E_1, \ldots, E_{maxLP-1} \rangle$$

- For the case $i = j$ we set the value to the number of executed events

Load-Sharing Policy

- Each tuple is a point in the n-dimensional space of LPs interactions
- The strategy is to identify *clusters of LPs* which show a high inter-dependence

- We rely on the *k-medoids* algorithm to find evenly-sized Voronoi regions in an Euclidean space
- The LPs are partitioned into K clusters, K being the number of available cores
- The distance between the coordinates $\mathbf{i}$ and $\mathbf{j}$ of LP_i and LP_j is their Manhattan distance:

$$d(\mathbf{i}, \mathbf{j}) = \|\mathbf{i} - \mathbf{j}\| = \sum_{i=1}^{n} |i_i - j_i|$$

Load-Sharing Policy

- This is used in the algorithm's objective function:

$$D = \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j \in C_k} d_{i,j}$$

- C_k is the set of all LPs in cluster k
- The recomputation can be periodical or triggered by a certain event of the system
- An initial LP is selected having the shortest distance to any other LP in the n-dimensional space, which is *approximately* in the center
- Other $k - 1$ LPs are selected so that they decrease the value of D as much as possible

Experimental Evaluation: Test-bed Platform

- Hardware configuration:
  - HP ProLiant server equipped with 64GB of RAM
  - 4 8-cores CPU (32 total cores)
- Software configuration:
  - ROOT-Sim Optimistic Simulation Kernel, using 32 symmetric worker threads
  - Debian 6
  - 2.6.32-5-amd64 Linux kernel

Synthetic Demographic Model

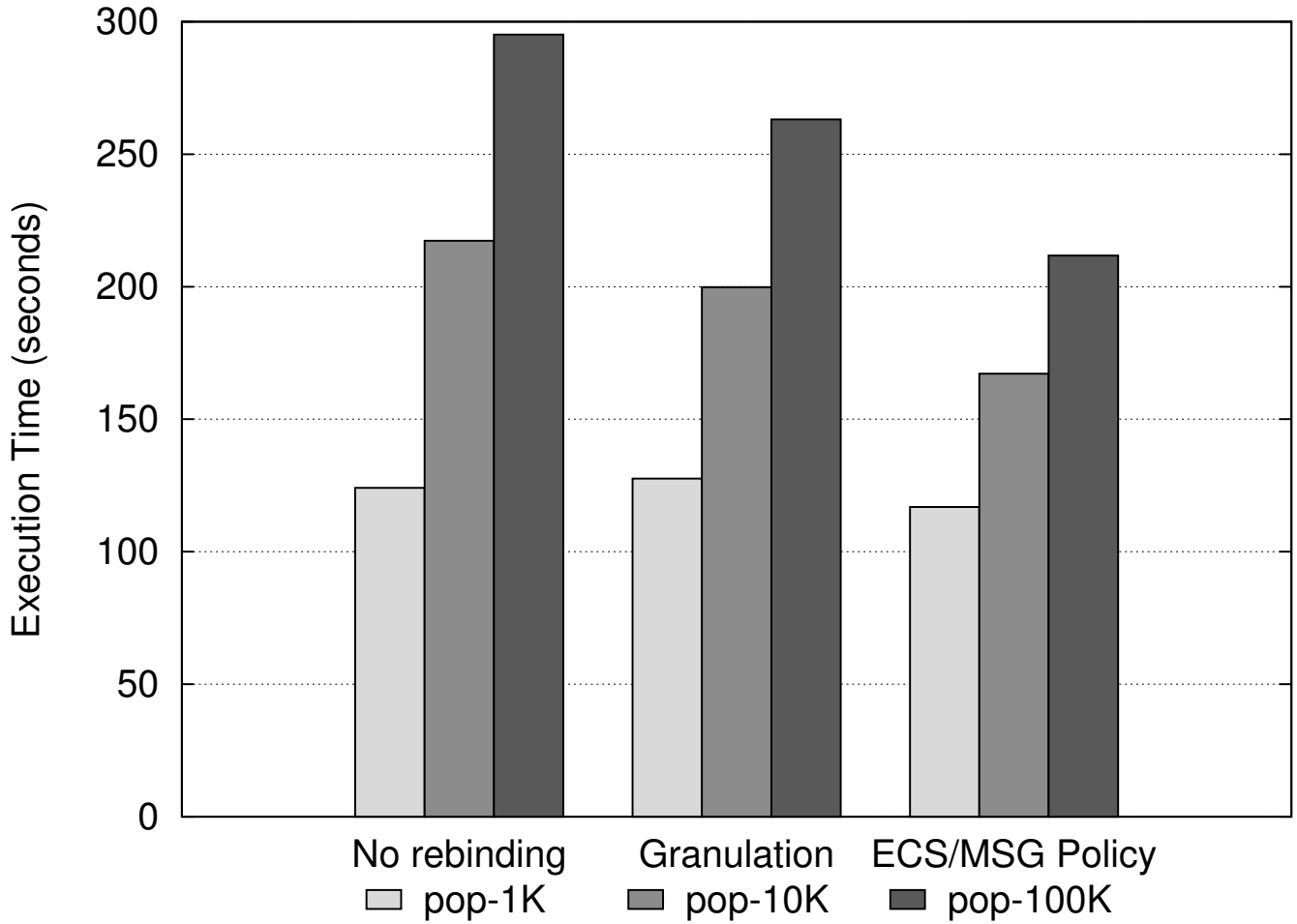
- Built according to our general guidelines
- Explanatory variables are described using a bitmap (concise description)
- Each agent carries as well a non-concise payload

- Additional operative events:
  - *State-machine update*: with probability p_{smu}, a bit in the bitmask is negated (state transition)
  - *Memory update*: with p_{mu}, a portion of the payload of the agent's structure is written with random data;
  - *Remote agent interaction*: with probability p_{rai}:
    - a random LP is selected to send random data
    - Upon receipt, a random hosted agent is picked
    - The payload of the event is copied into the agent's buffer
  - This latter event mimics kinship interactions

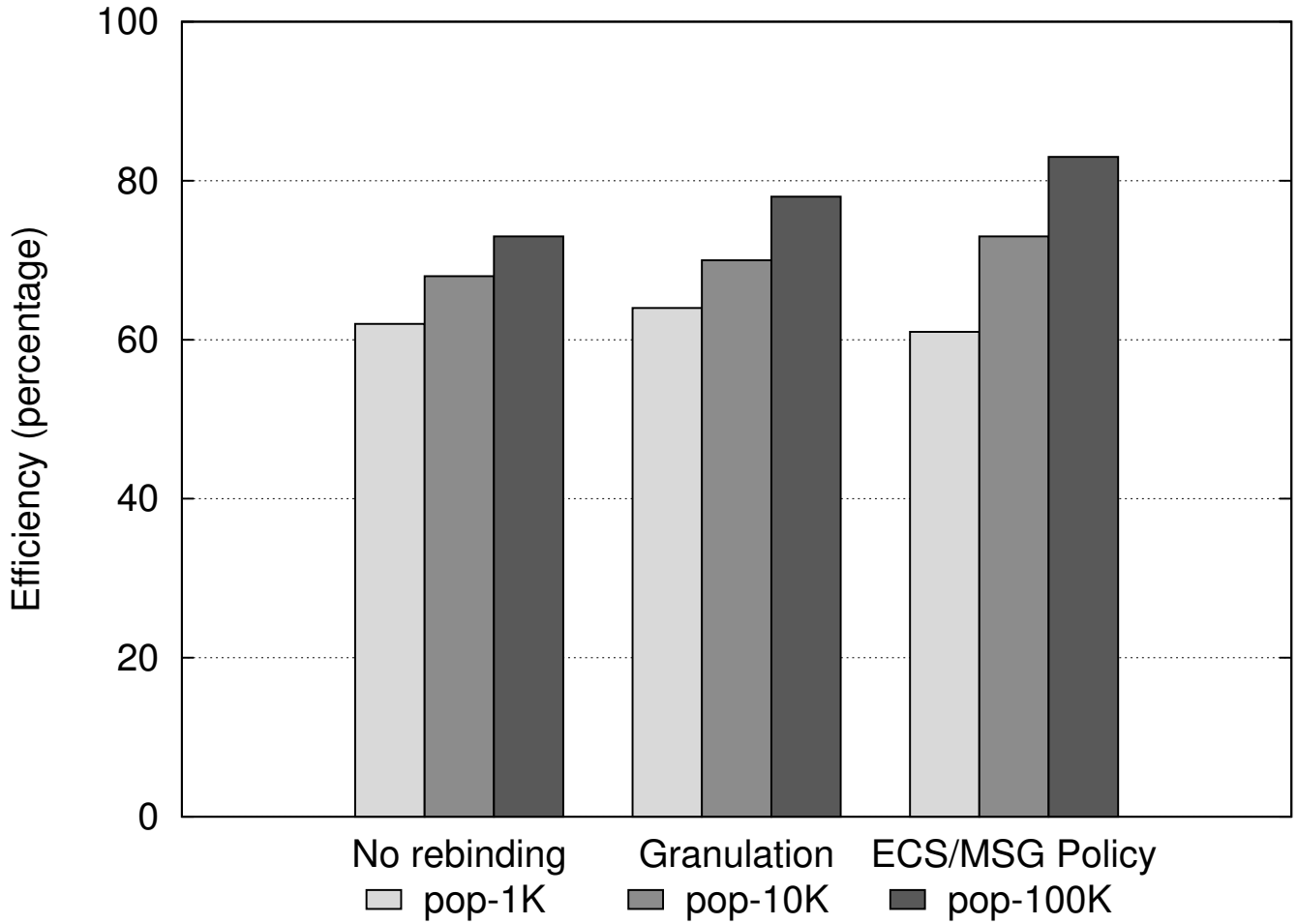
Baseline

- No Rebinding
- LP Granulation [PADS 2016]
  - ECS-based interactions are still captured
  - They are used to form *groups of LPs*
  - A group is a *sequential entity*
  - A group is thus bound on a single worker thread
  - Groups are ephemeral

Synthetic Demographic Model: Results



Synthetic Demographic Model: Results

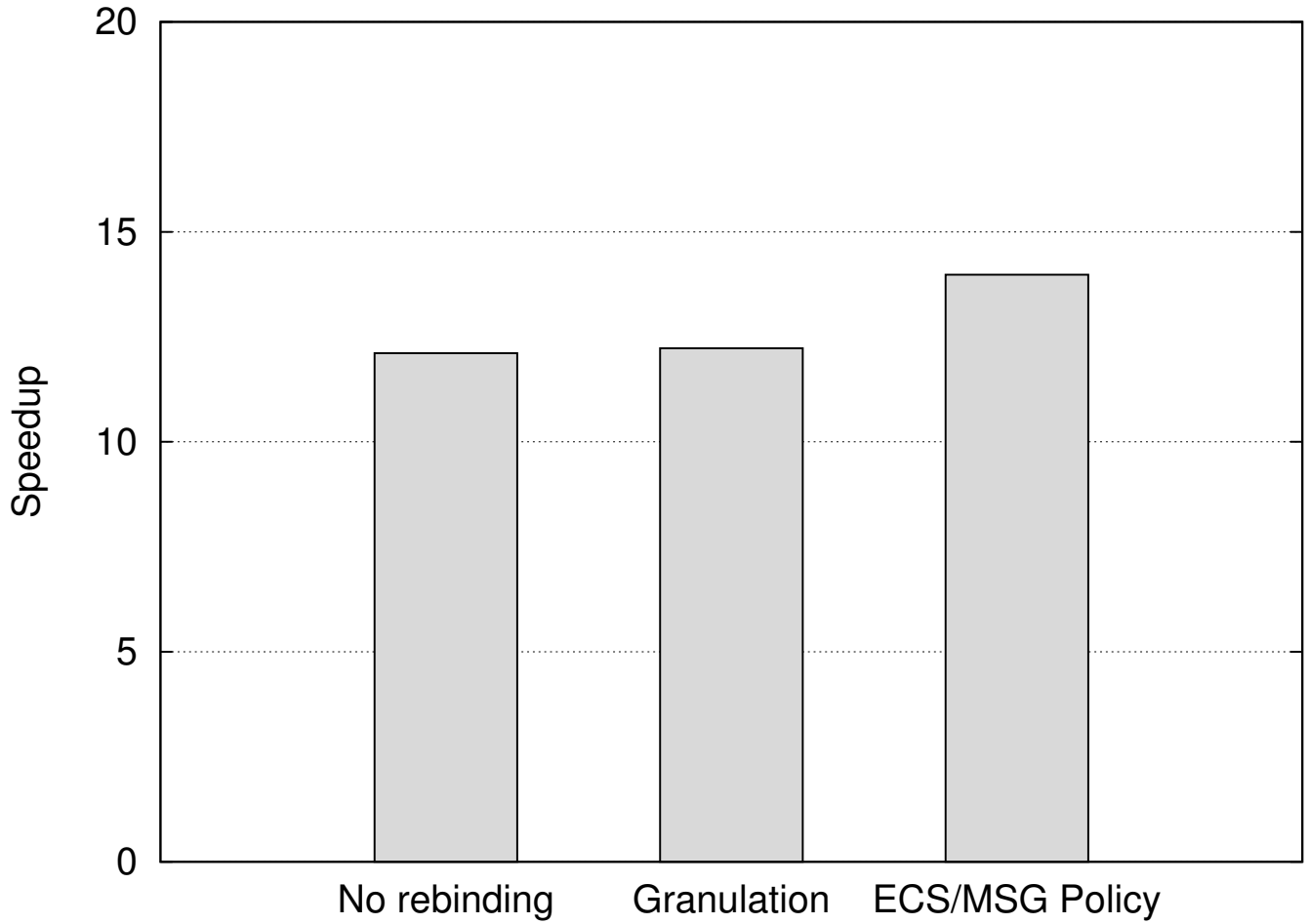


Yades

- Parallel demographic agent-based simulation tool
- Life course of individuals is modeled via agents
  - fertility
  - mortality
  - economic status
  - family composition
  - family migration

- Gambian immigration in Spain during 10 years
- 40,000 families

Yades: Results



Thanks for your attention

pellegrini@dis.uniroma1.it
http://www.dis.uniroma1.it/∼pellegrini
http://github.com/HPDCS/ROOT-Sim