INTERNATIONAL CONFERENCE ON COMPUTER SIMULATION IN PHYSICS AND BEYOND SEPTEMBER 24-27, 2018, MOSCOW

THE ANALYSIS OF OPTIMISTIC PARALLEL DISCRETE EVENT SIMULATION ALGORITHM ON SMALL-WORLD NETWORKS

Ziganurova Liliia[1,2], Lev Shchur[1,2,3]

1. National Research University Higher School of Economics, Moscow
2. Scientific Center in Chernogolovka, Chernogolovka, Moscow area
3. Landau Institute for Theoretical Physics, Chernogolovka, Moscow area

OVERVIEW

OVERVIEW

- What is the Parallel Discrete Event Simulation?

OVERVIEW

- What is the Parallel Discrete Event Simulation?
- Optimistic synchronisation algorithm

OVERVIEW

- What is the Parallel Discrete Event Simulation?
- Optimistic synchronisation algorithm
- The simulation of the algorithms on Small-World networks

OVERVIEW

- What is the Parallel Discrete Event Simulation?
- Optimistic synchronisation algorithm
- The simulation of the algorithms on Small-World networks
- Results: How the underlying topology affects the synchronisation in PDES?

MOTIVATION

MOTIVATION

- Modern computer systems: **10^6 of cores**

MOTIVATION

- Modern computer systems: **10^6 of cores**
- Nodes have a number of CPUs, cores, and numerical accelerator

MOTIVATION

- Modern computer systems: **10^6 of cores**
- Nodes have a number of CPUs, cores, and numerical accelerator
- CPU's (cores, threads, …) must be **synchronised** to efficiently execute one parallel program

MOTIVATION

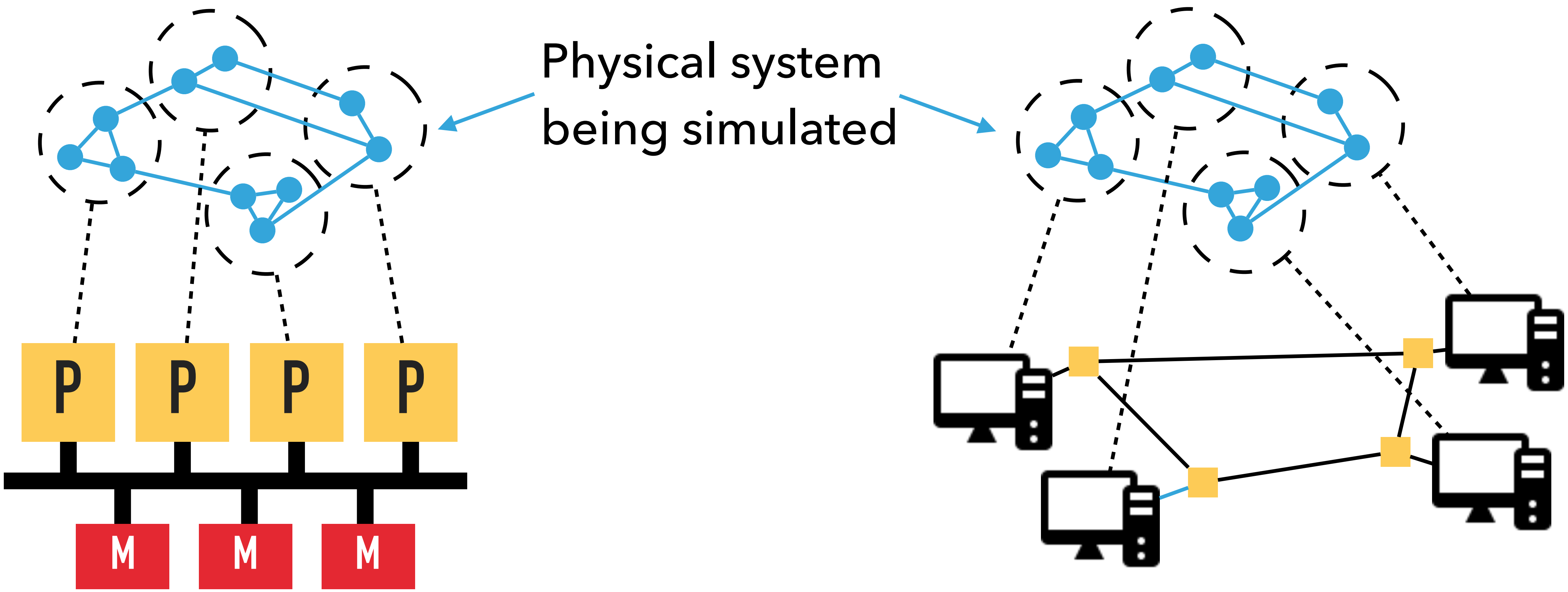
- Modern computer systems: **10^6 of cores**
- Nodes have a number of CPUs, cores, and numerical accelerator
- CPU's (cores, threads, …) must be **synchronised** to efficiently execute one parallel program
- High performance computing requires new approaches to programming models

MOTIVATION

- Modern computer systems: **10^6 of cores**
- Nodes have a number of CPUs, cores, and numerical accelerator
- CPU's (cores, threads, …) must be **synchronised** to efficiently execute one parallel program
- High performance computing requires new approaches to programming models

Parallel Discrete Event Simulation is a method of large-scale simulation which allows to execute a single program on a parallel computer.

PARALLEL AND DISTRIBUTED SIMULATION*



Parallel simulation involves the execution of a *single* simulation on a collection of **tightly** coupled processors (e.g. a shared memory multiprocessor)

Distributed simulation involves the execution of a *single* simulation on a collection of **loosely** coupled processors (e.g. PCs interconnected by a LAN or WAN)

**from R.Fujimoto[1]*

ESSENTIAL PROPERTIES OF PDES:

ESSENTIAL PROPERTIES OF PDES:

- Changes in subsystems occur at some instant of time and are called **discrete events.**

ESSENTIAL PROPERTIES OF PDES:

- Changes in subsystems occur at some instant of time and are called **discrete events.**
- To preserve causality between dependent objects some **synchronisation protocol** is used.

ESSENTIAL PROPERTIES OF PDES:

- Changes in subsystems occur at some instant of time and are called **discrete events.**
- To preserve causality between dependent objects some **synchronisation protocol** is used.
- Using the **virtual time concept**.

ESSENTIAL PROPERTIES OF PDES:

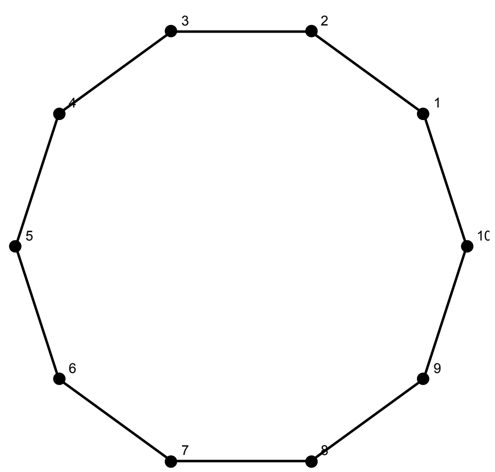
- Changes in subsystems occur at some instant of time and are called **discrete events.**
- To preserve causality between dependent objects some **synchronisation protocol** is used.
- Using the **virtual time concept**.
- Communication between parallel processes goes via timestamped messages.

ESSENTIAL PROPERTIES OF PDES:

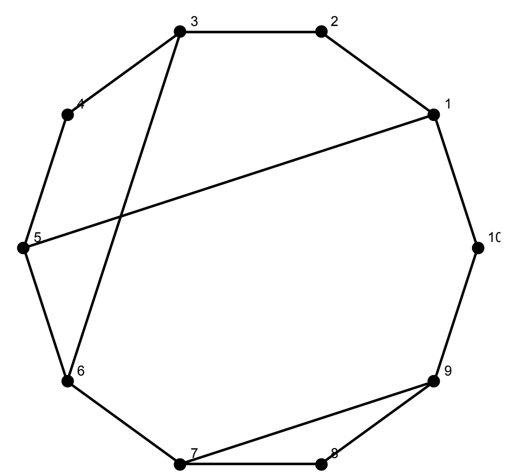
- Changes in subsystems occur at some instant of time and are called **discrete events.**
- To preserve causality between dependent objects some **synchronisation protocol** is used.
- Using the **virtual time concept**.
- Communication between parallel processes goes via timestamped messages.
- No shared memory between subsystems.

UNDERLYING TOPOLOGIES

Regular



Small-World-like [4]



p=0

0 < p << 1

Vertices - PEs

Edges - dependencies

1. Average shortest path ~ log(N)
2. Clustering coefficient ≈ 0

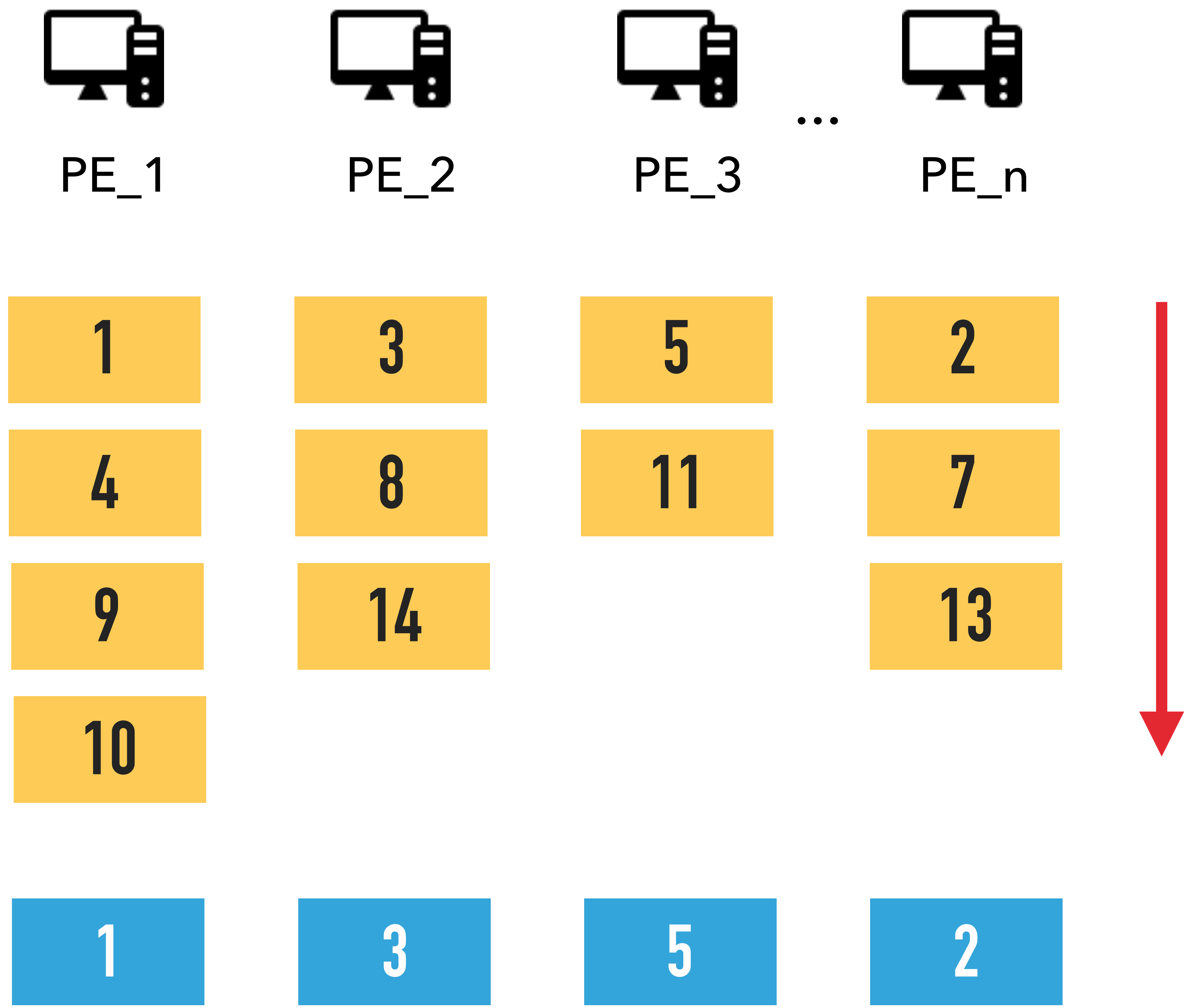
p - concentration of added links

VIRTUAL TIME CONCEPT (AN EXAMPLE)

PE = node/CPU/core/etc.

Lists of events with timestamps

Local virtual time of PEs

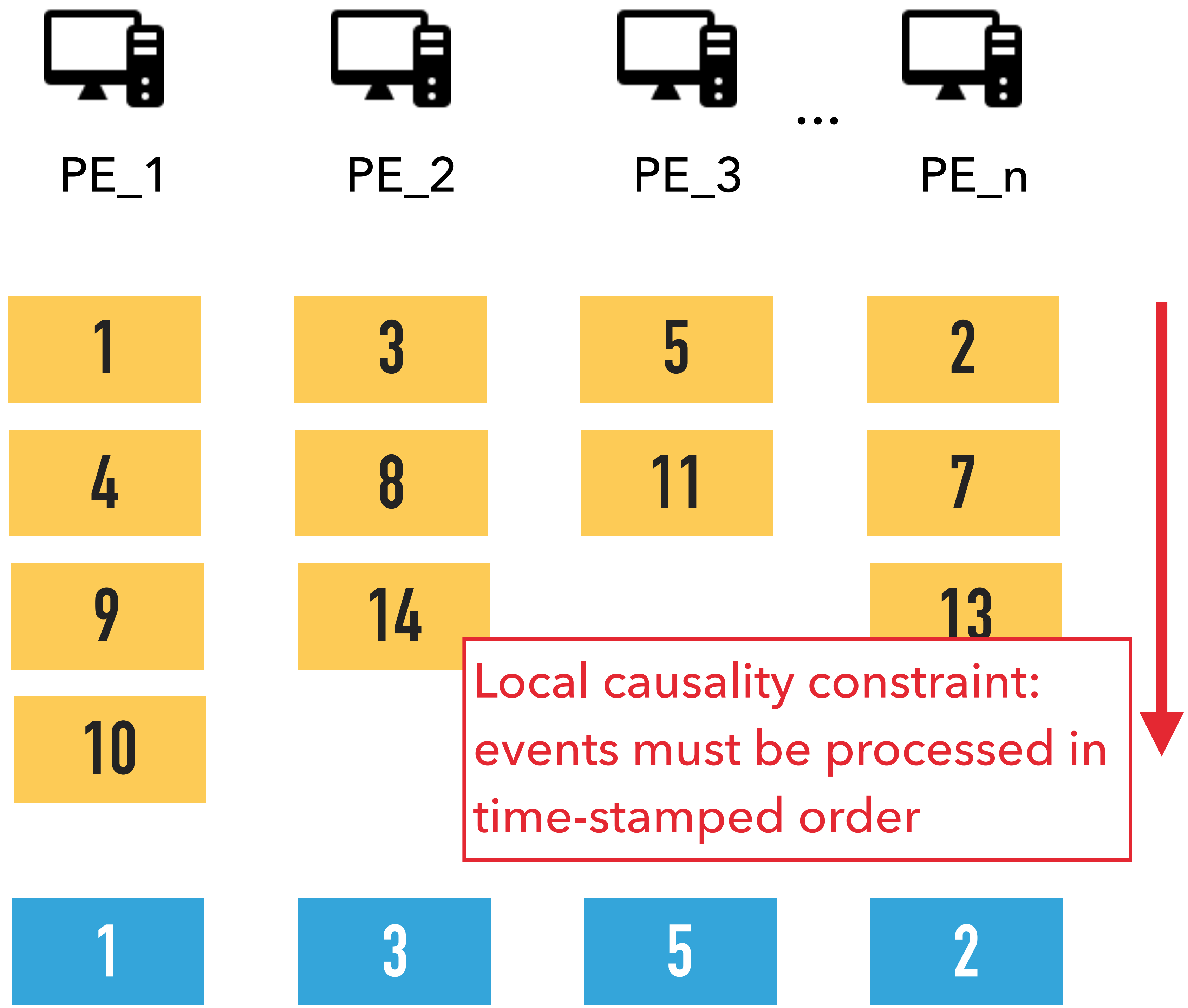


VIRTUAL TIME CONCEPT (AN EXAMPLE)

PE = node/CPU/core/etc.

Lists of events with timestamps

Local virtual time of PEs



Optimistic algorithm

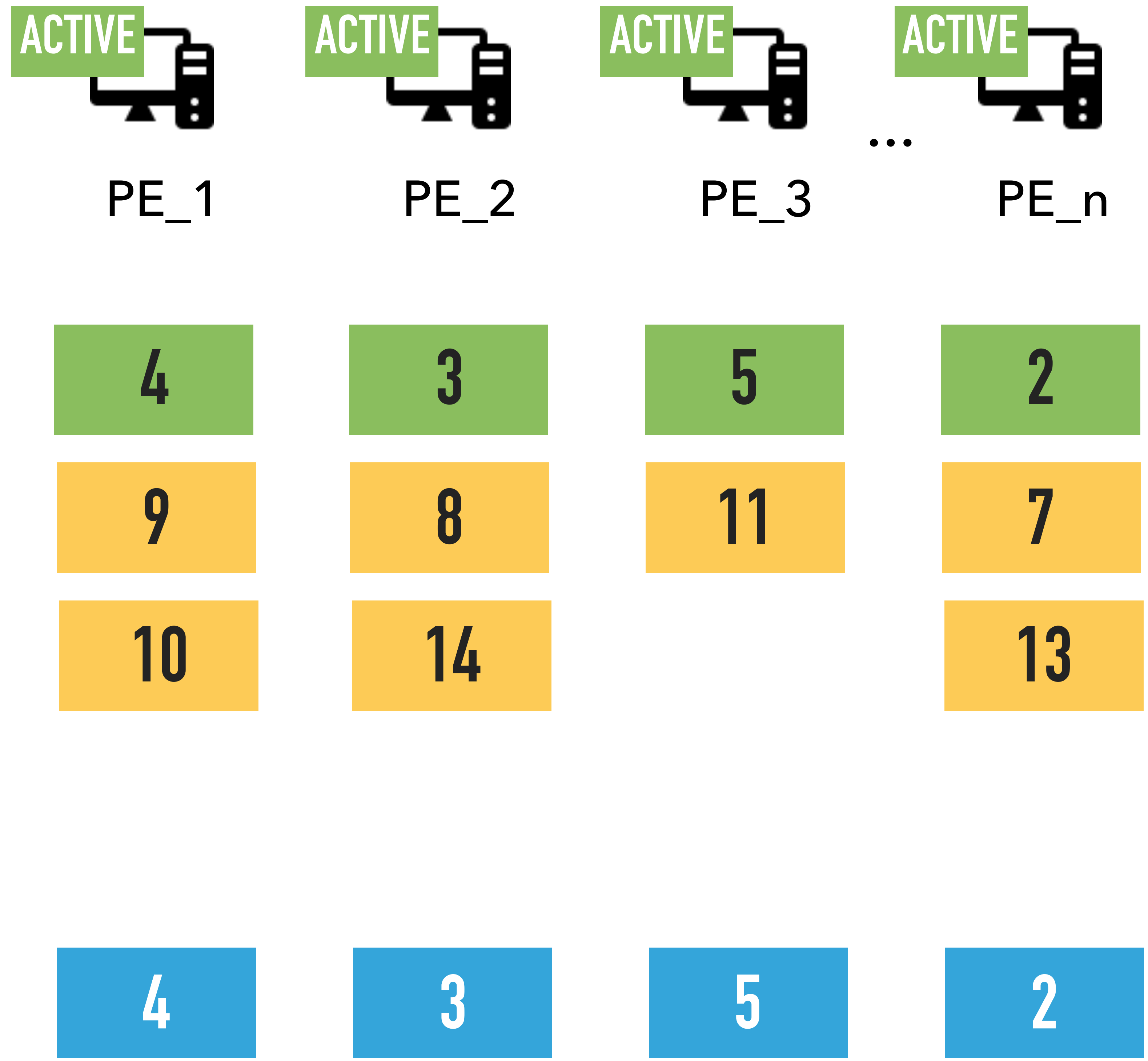
Allows emergence of causality errors but has a roll-back mechanism. The process, received a message with timestamp lower than its LVT (‘‘from the past’’), rolls-back to the state with lower time. It also sends anti messages to other processes to cancel previously sent messages.

OPTIMISTIC SYNCHRONISATION

PE = node/CPU/core/etc.

Lists of events with timestamps

Local virtual time of PEs



OPTIMISTIC SYNCHRONISATION

PE = node/CPU/core/etc.

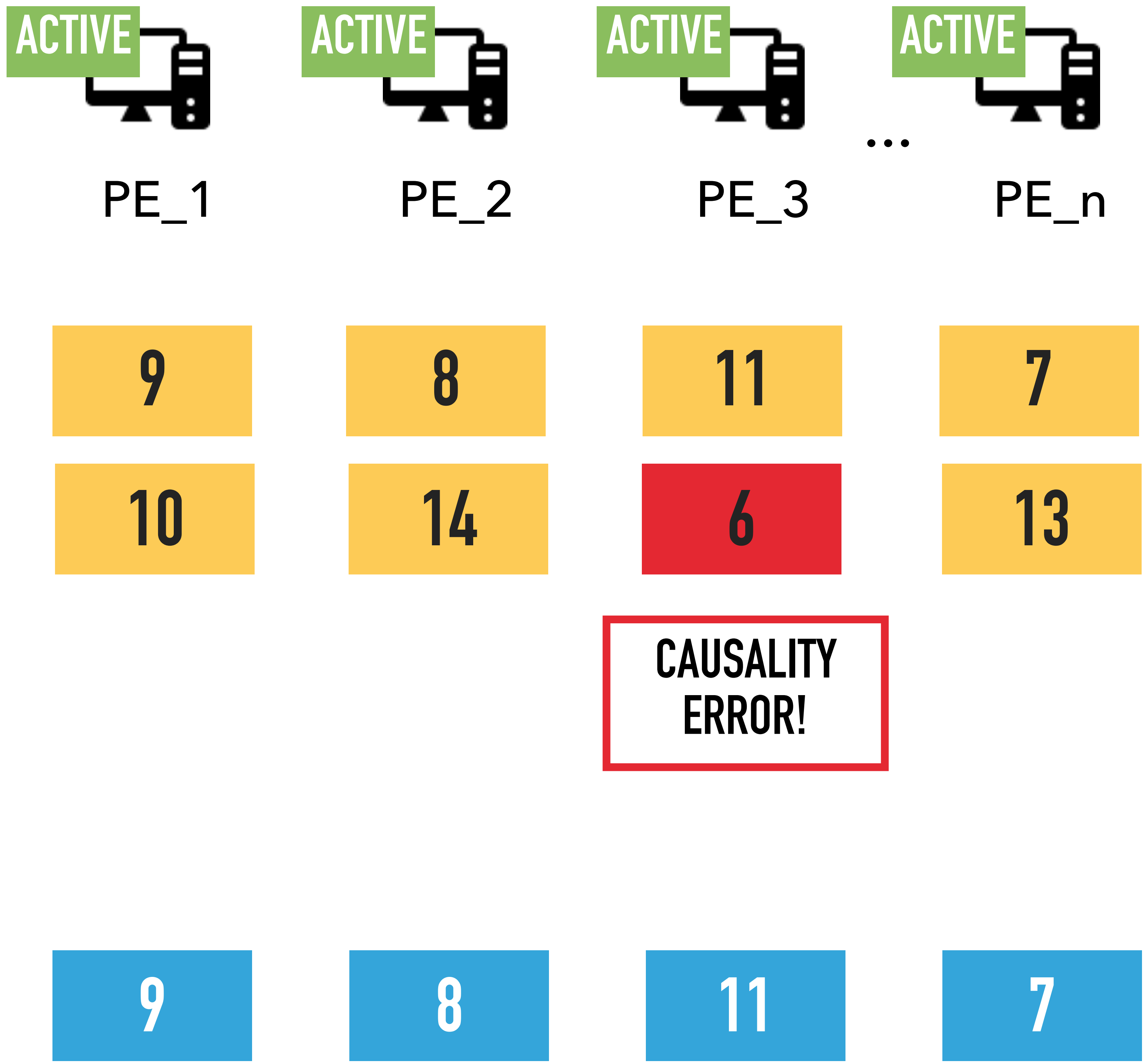
	PE_1	PE_2	PE_3	...	PE_n
Status	ACTIVE	ACTIVE	ACTIVE		ACTIVE
Lists of events with timestamps	9	8	11		7
	10	14			13
Local virtual time of PEs	9	8	11		7

OPTIMISTIC SYNCHRONISATION

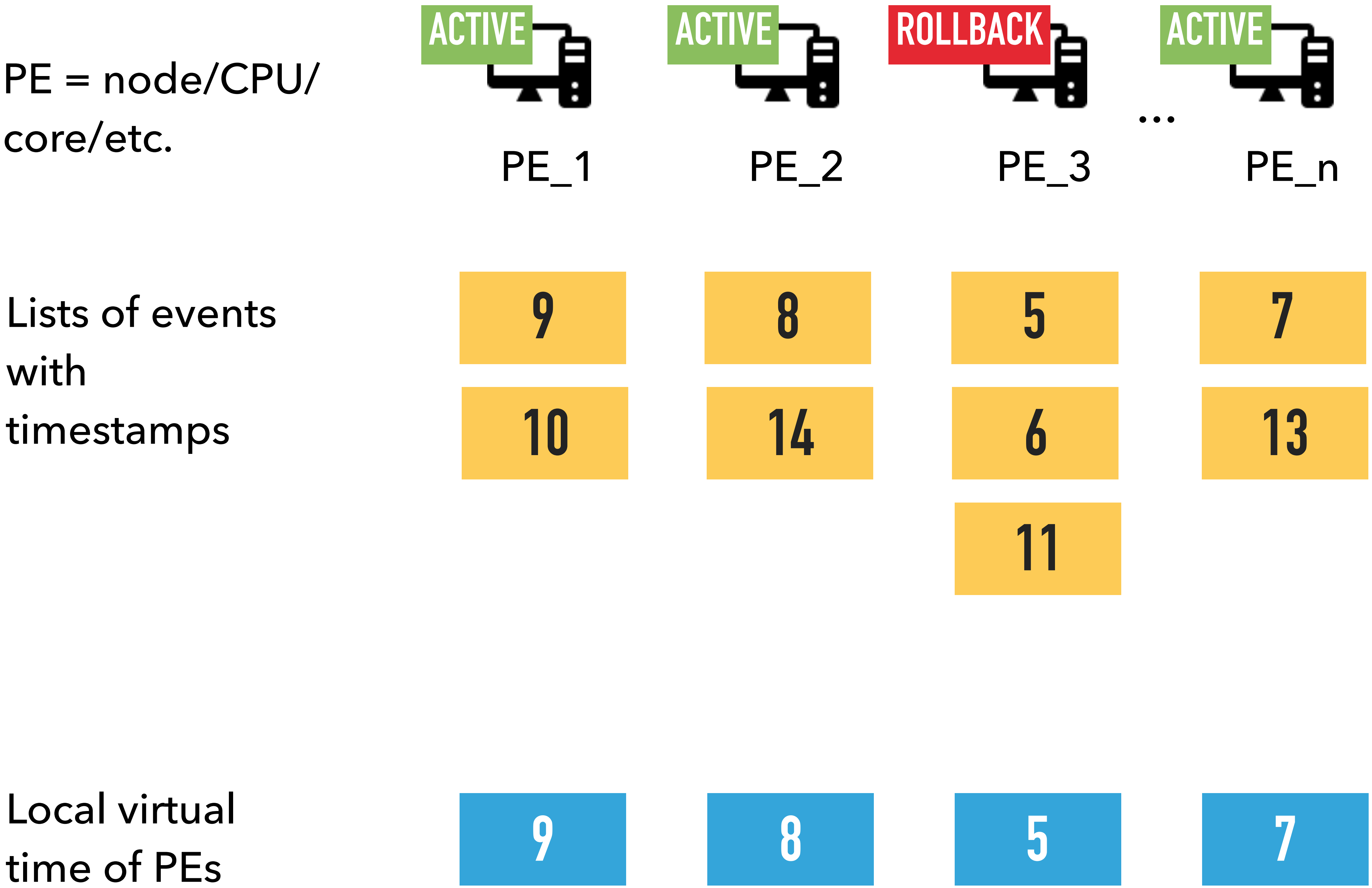
PE = node/CPU/core/etc.

Lists of events with timestamps

Local virtual time of PEs



OPTIMISTIC SYNCHRONISATION

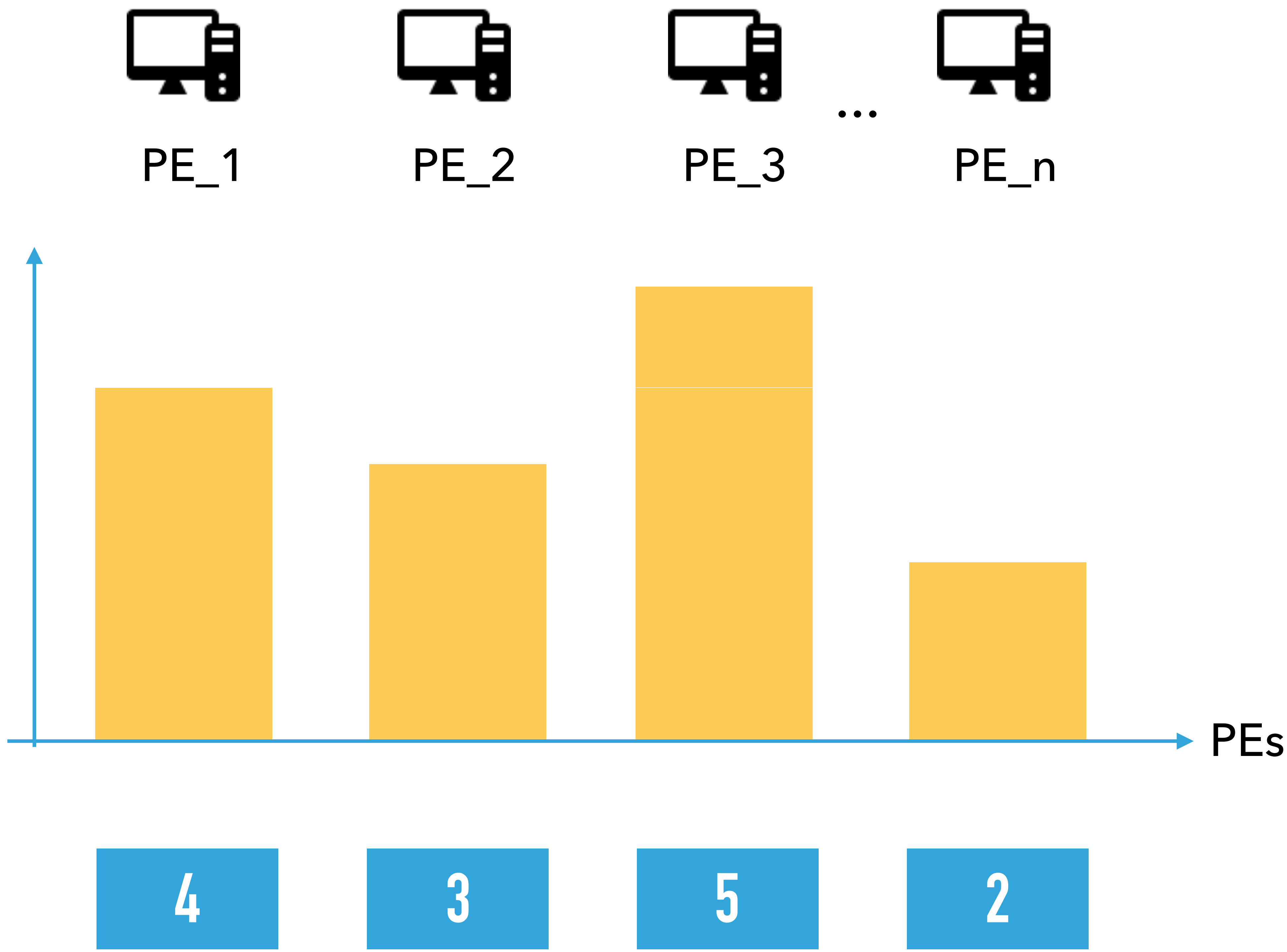


THE CONCEPT OF VIRTUAL TIMES (EXAMPLE)

PE = node/CPU/core/etc.

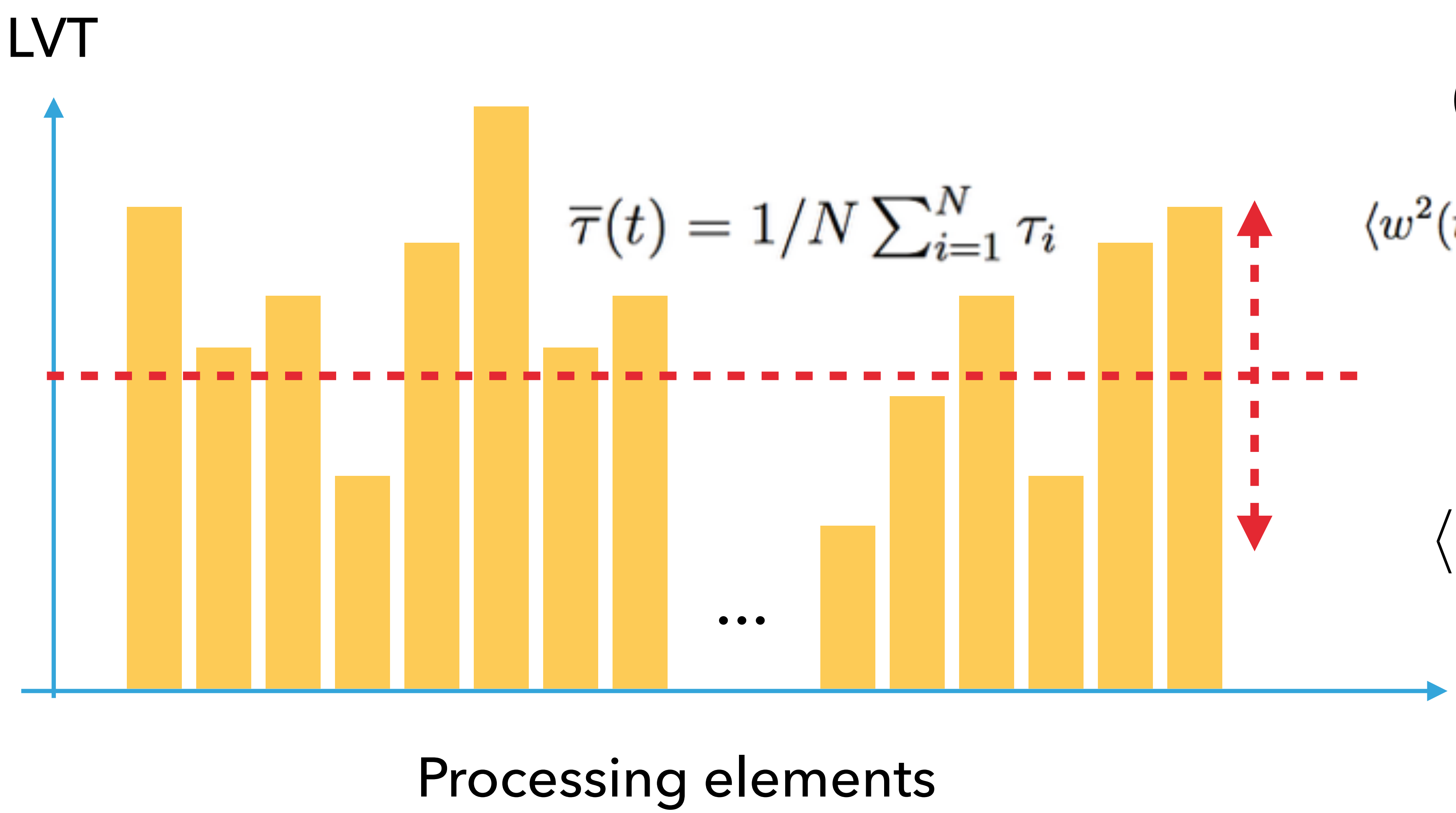
Local virtual time profile

Local virtual time of PEs



THE OBJECT OF THE RESEARCH

We study the scalability properties of **Optimistic** synchronisation algorithm on small-world communicational network.



1. width of the profile (=desynchronisation)

$$\langle w^2(t) \rangle = \langle \frac{1}{N} \sum_{i=1}^{N} [\tau_i(t) - \overline{\tau}(t)]^2 \rangle$$

2. speed of the profile (=utilization)

$$\langle u(t) \rangle = \langle \tau_i(t) - \tau_i(t-1) \rangle_i$$

THE MODEL OF EVOLUTION OF THE LOCAL VIRTUAL TIME PROFILE

Set parameters N, M, p, b
Create small-world graph with pN random long-range links.
for $t = 0..M$ **do**
    for $i = 0..N$ **do** $\tau_i(t) += \eta_i$
    k = Poisson(b)
    for $j = 1..kN$ **do**
        Choose random PE_m
        Choose random neighbour of PE_m PE_r
        If $\tau_m(t) > \tau_r(t)$ then $\tau_m(t) = \tau_r(t)$
    Calculate observables.

$$\langle w^2(t)\rangle = \langle \frac{1}{N}\sum_{i=1}^{N}[\tau_i(t) - \overline{\tau}(t)]^2\rangle$$

$$\langle u(t)\rangle = \langle \tau_i(t) - \tau_i(t-1)\rangle_i$$

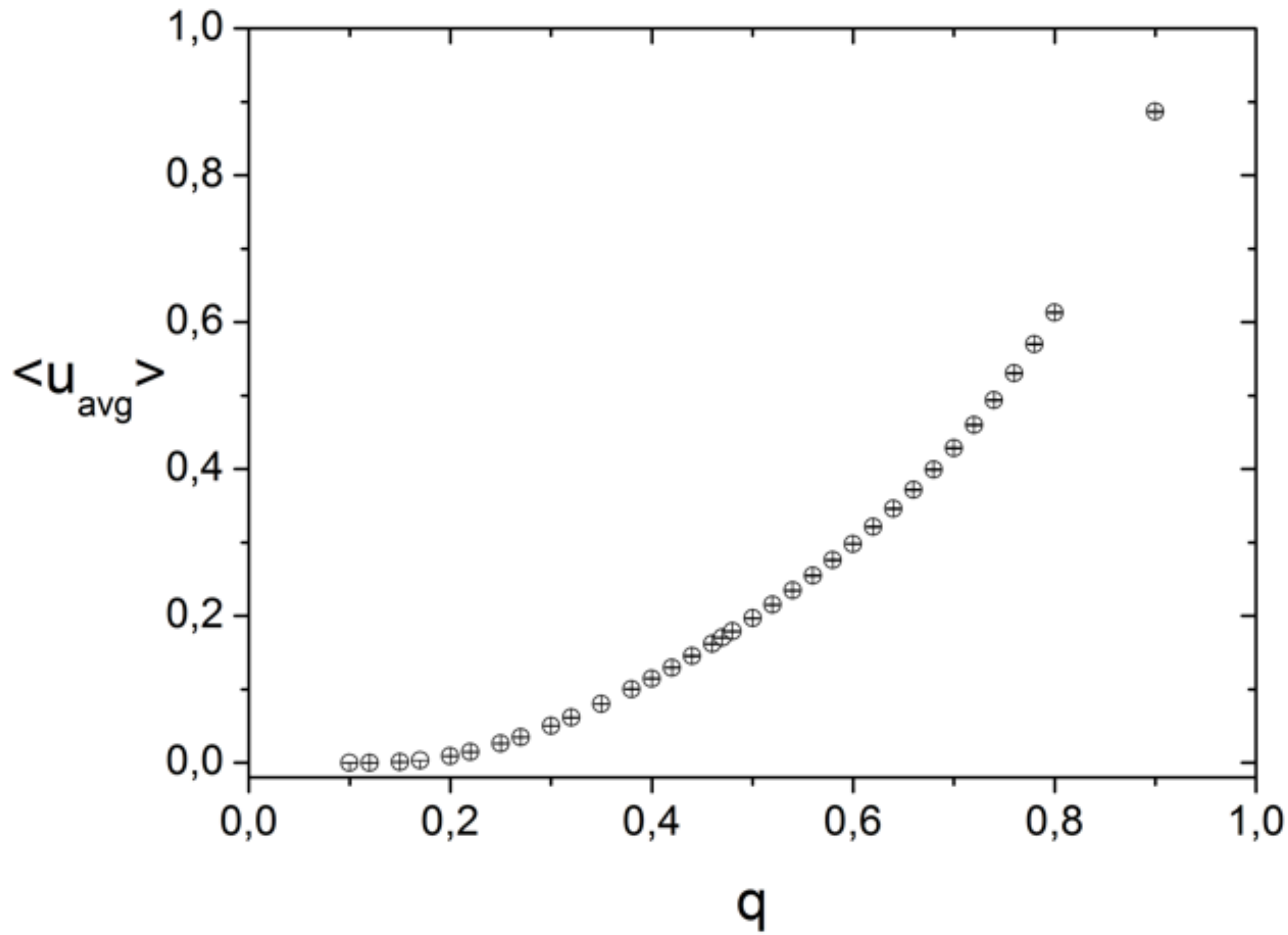
RESULT#1: AVERAGE SPEED

The parameter (progress rate):

$$q = \frac{1}{1+b}$$

where b is a mean avalanche length (the number of PEs, which rolled back during one simulation step)

$$u = u_0(q - q_c)^\nu$$



RESULT#1: AVERAGE SPEED FITTING

DIRECTED PERCOLATION UNIVERSALITY CLASS

$$\nu = 1.733847(6)$$

The phase transition takes place at

$$q_c = 0.189(2)$$

for the RSOS model

and

$$q_c = 0.233(1)$$

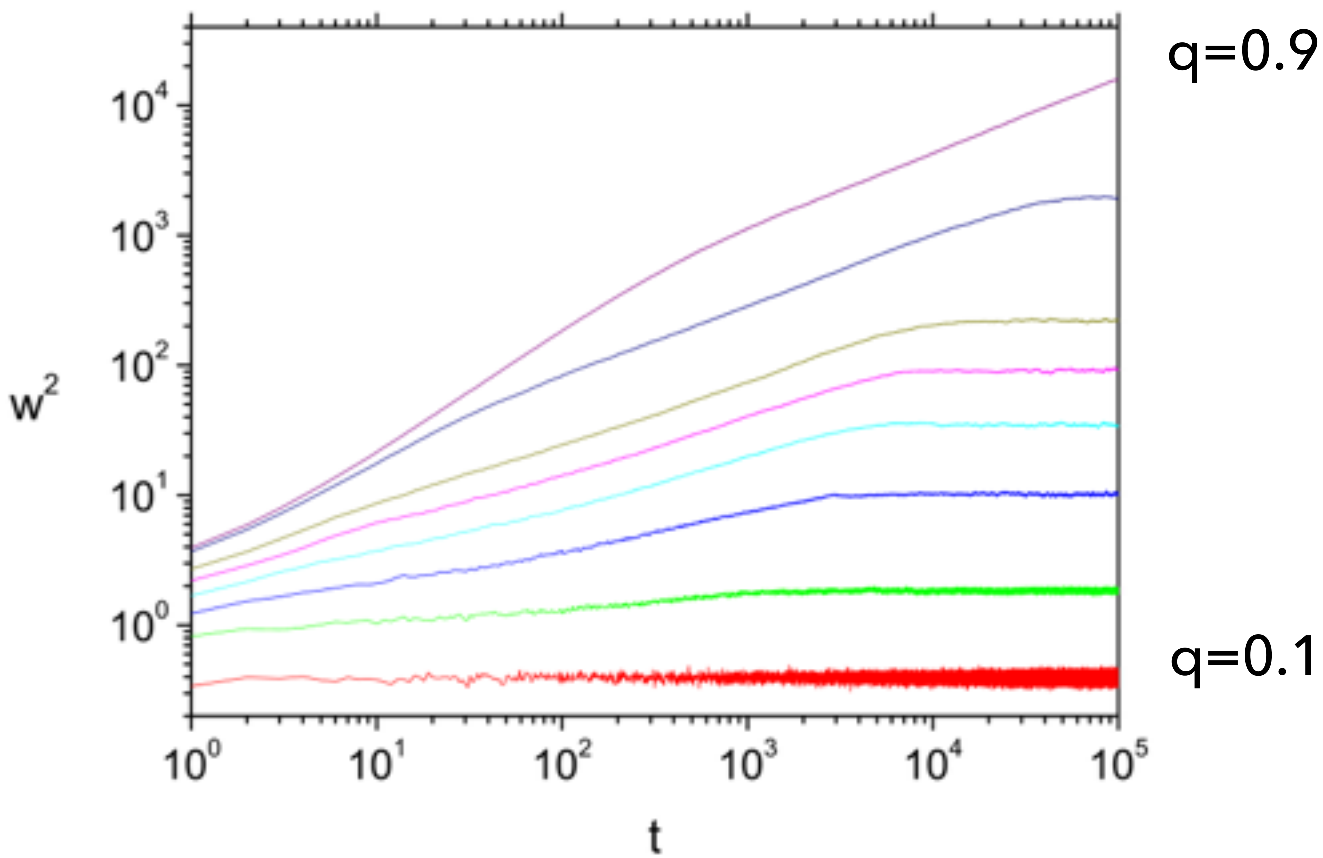
for the unrestricted model

$$u = u_0(q - q_c)^\nu$$

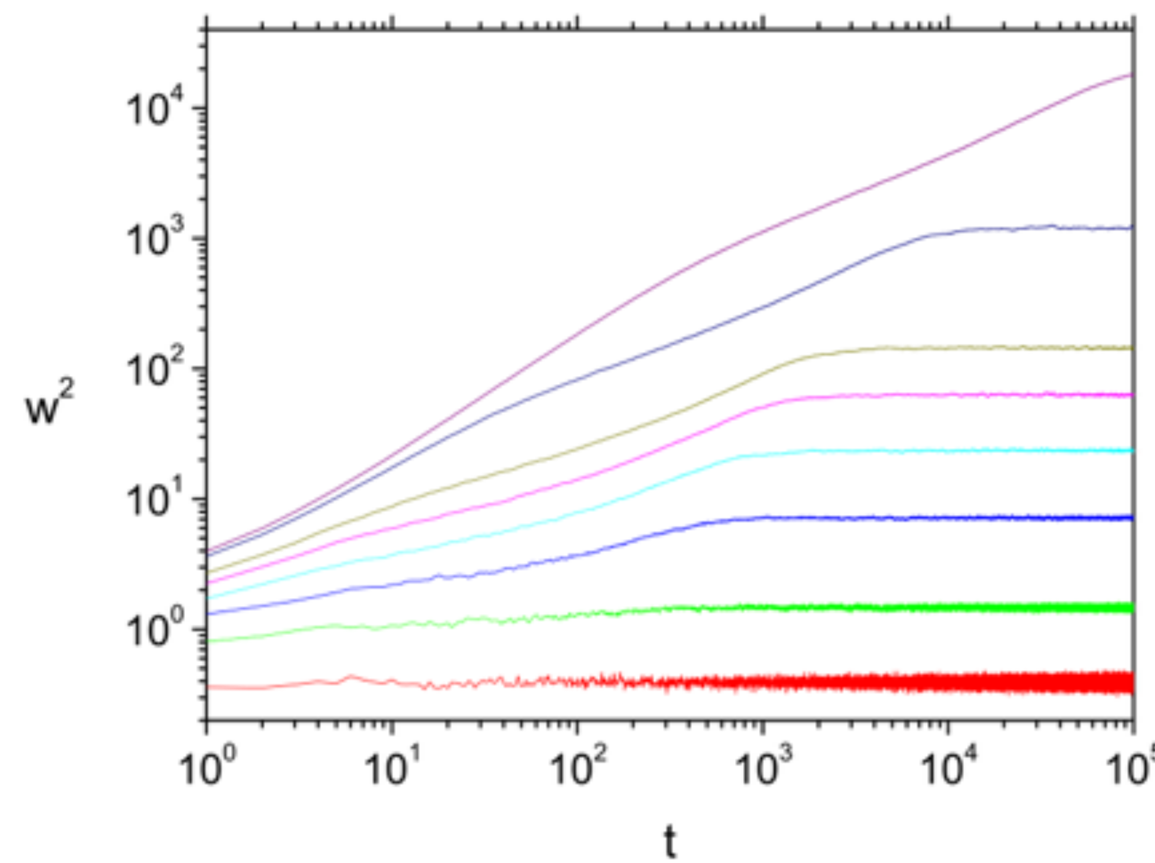
p	u_0	q_c	v
0	1,093(5)	0,143(1)	1,66(1)
0,001	1,126(9)	0,149(3)	1,69(2)
0,01	1,165(8)	0,166(2)	1,70(1)
0,05	1,27(1)	0,202(2)	1,74(2)
0,1	1,38(1)	0,224(2)	1,79(1)
0,2	1,54(2)	0,249(3)	1,88(2)

RESULT#2: AVERAGE WIDTH

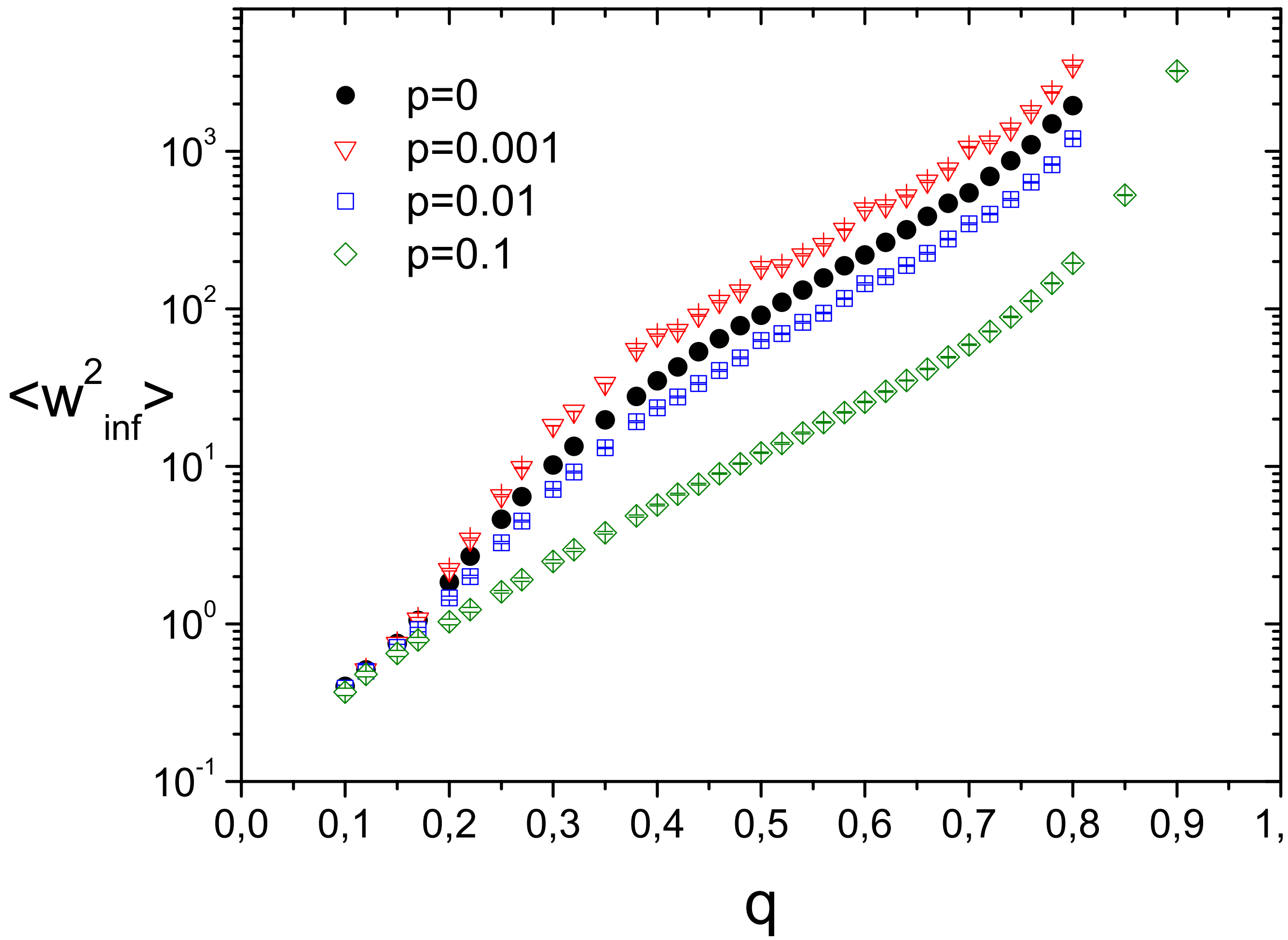
Regular lattice

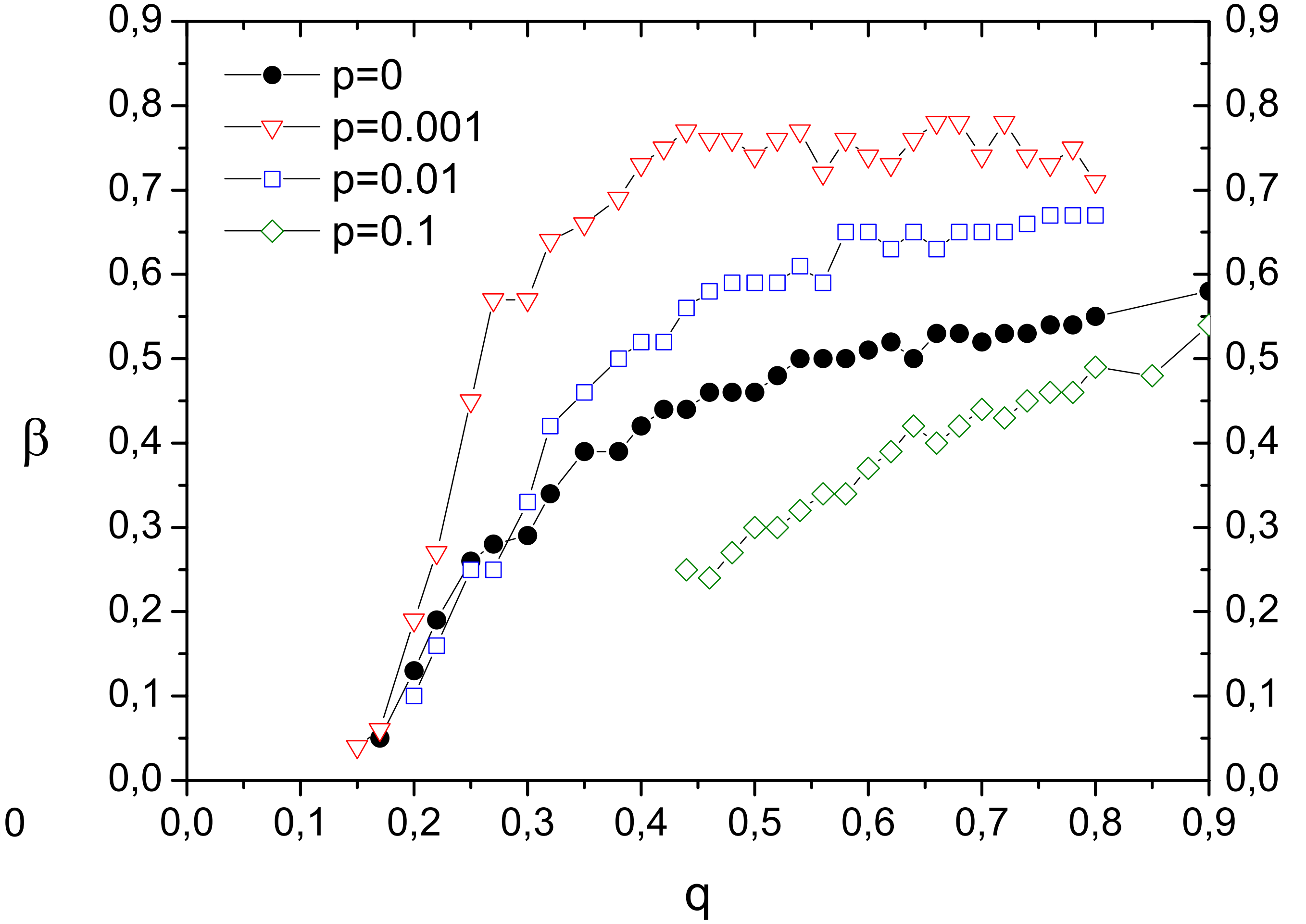


Small-world network, p=0.01



RESULT#2: AVERAGE WIDTH





CONCLUSION:

- **We built the model of local virtual time profile growth in optimistic PDES**
- **The average speed of the profile behaves as:** $u = u_0(q - q_c)^v$
- **The model belongs to DP universality class**
- **The behaviour of the average width does not change with the topology, i.e. it saturates after some time.**

REFERENCIES

1. *Fujimoto R.M*. Parallel Discrete Event Simulation // Commun. ACM. 1990. V. 33. P. 31–53.

2. *Jefferson D.R*. Virtual Time // ACM Trans. on Programming Languages and Systems (TOPLAS). 1985. V. 7. P. 404–425.

3. Shchur, Lev N., and M. A. Novotny. "Evolution of time horizons in parallel and grid simulations." Physical Review E 70.2 (2004): 026703.

4. Watts, Duncan J., and Steven H. Strogatz. "Collective dynamics of'small-world'networks." nature 393.6684 (1998): 440.

5. G. Korniss, Z. Toroczkai, M. Novotny, and P. A. Rikvold, Physical review letters 84, 1351 (2000).

6. Korniss G. et al. Suppressing roughness of virtual times in parallel discrete-event simulations //Science. – 2003. – T. 299. – №. 5607. – C. 677-679.

7. Wilsey, P.A.: Some Properties of Events Executed in Discrete-Event Simulation Models. In: Proceedings of the 2016 annual ACM Conference on SIGSIM Principles of Advanced Discrete Simulation, pp. 165-176 (2016). ACM, New York (2016).

8. *Ziganurova L.F., Shchur L.N., Novotny M.A*. Model for the evolution of the time profile in optimistic parallel discrete event simulations // Journal of Physics: Conference Series **681** (2016) 012047

9. Ziganurova, Liliia & Shchur, Lev. (2017). Properties of the Conservative Parallel Discrete Event Simulation Algorithm. in Lecture Notes in Computer Science series, Vol. 10421, p. 246-253. doi: 10.1007/978-3-319-62932-2_23.