Radio Communications Research Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Evaluation of Computing Resource Management Methods for SDR Clouds

Vuk Marojevic

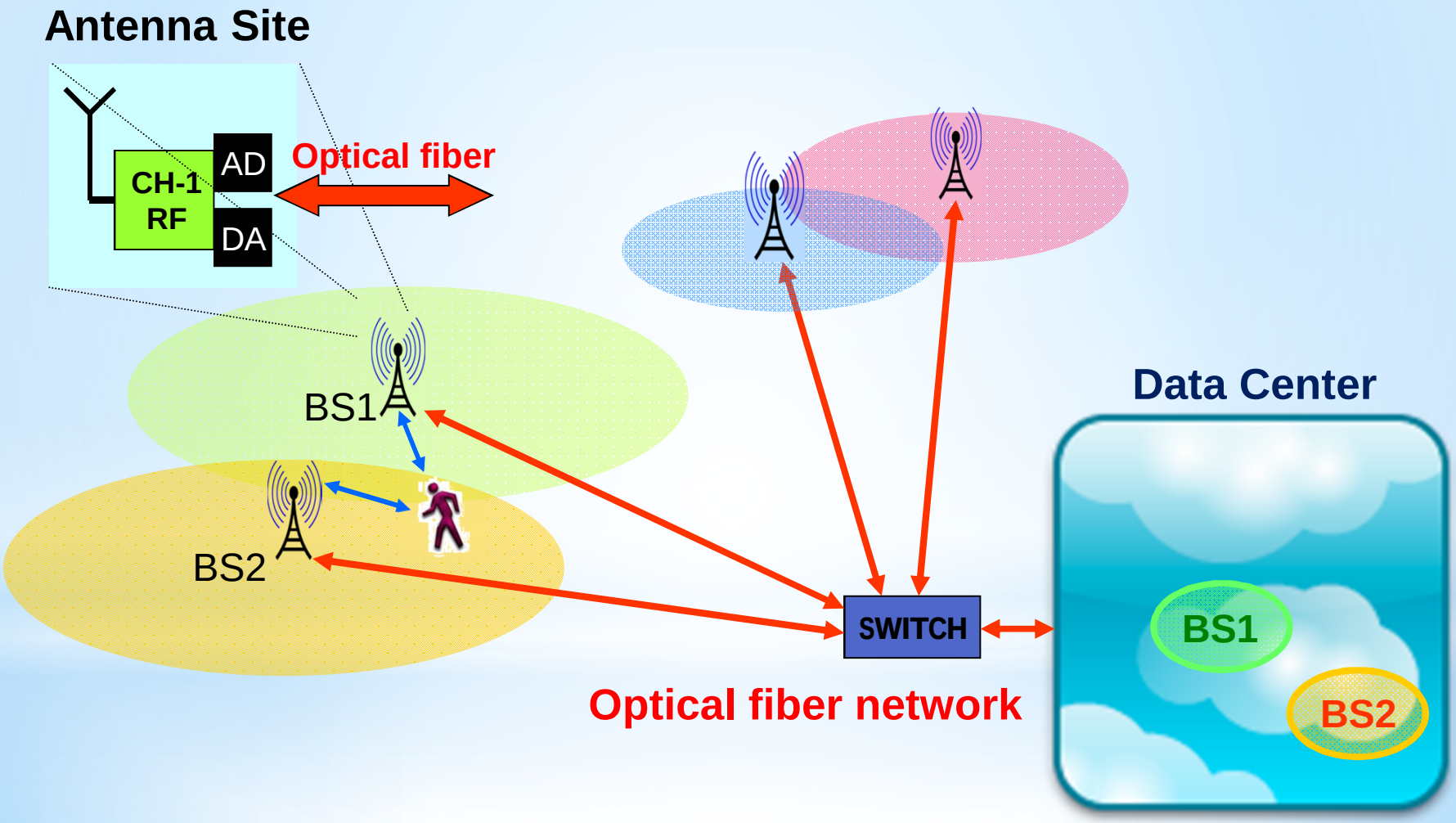
Ismael Gomez

Antoni Gelonch

Index

* Context and Problem Formulation
* Scheduling Techniques
* Analysis
* Conclusions

SDR Cloud


Antenna Site
CH-1
RF
AD
DA
Optical fiber
BS1
BS2
SWITCH
Optical fiber network
Data Center
BS1
BS2

Real-Time Scheduling

* "In a real-time computing system *correctness* depends not only on the logical result of the computation but also on the time at which the results are produced"

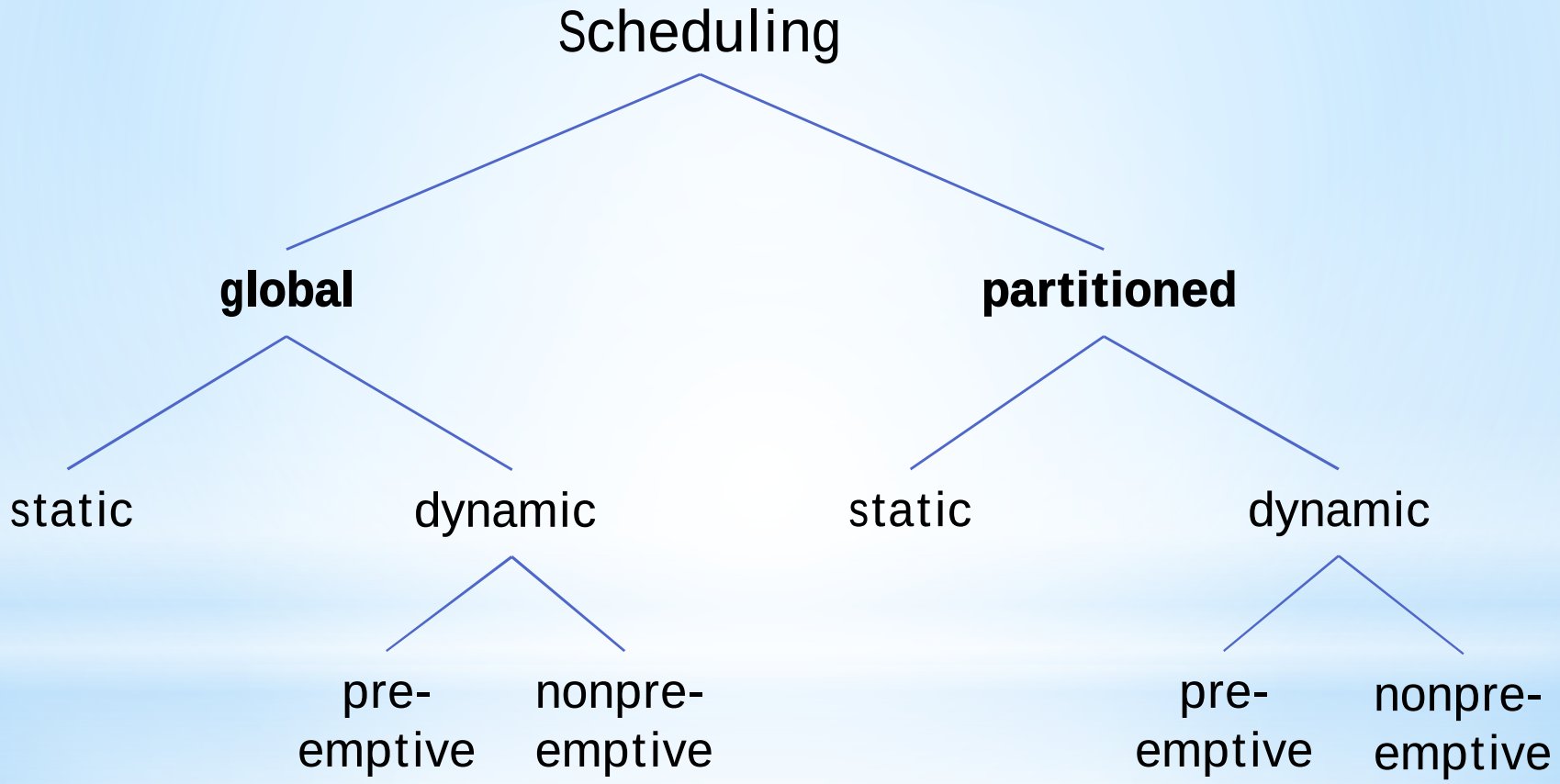
  J.A. Stankovic, A serious problem for next-generation systems. IEEE Computer, vol. 21, iss. 10, 1988.

* SDR Cloud:
  * Allocate computing resources for real-time waveform execution (throughput and latency)
  * Runtime resource allocation (real-time)

Problem Formulation

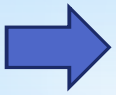
* Analyze different scheduling techniques for SDR clouds
  * in terms of complexity/resource overhead
  * *not* in terms of performance

Scheduling Classification



Global vs. Partitioned

Global	Partitioned
Global scheduling queue	Distributed schedulers
Implicit resource assignment and scheduling	Multiprocessor mapping, uniprocessor scheduling
Objective function can optimize for execution time (speedup)	Different objective for mapping and scheduling
Non-scalable	Scalable



Clustered or semi-partitioned scheduling

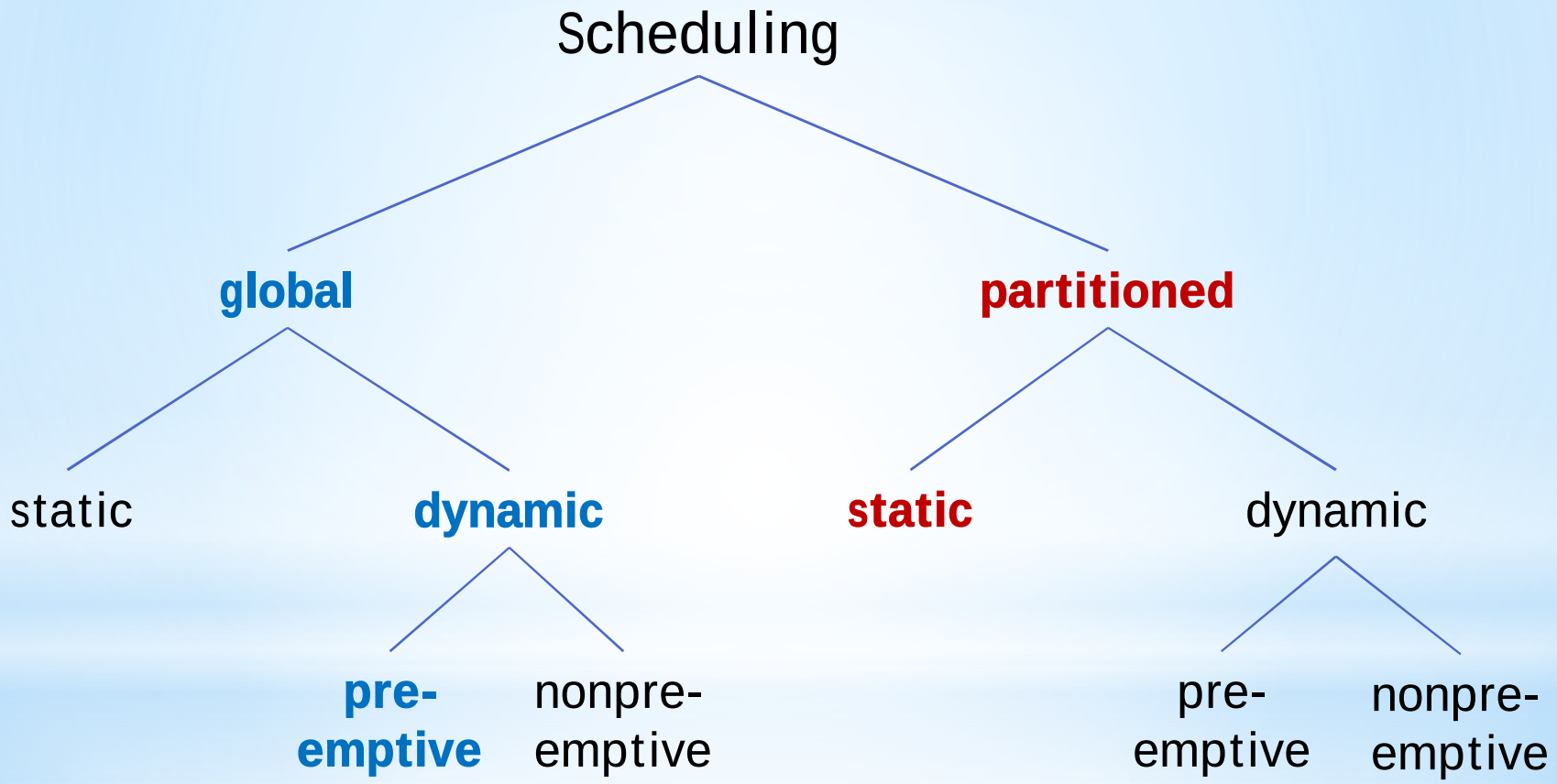
Static vs. Dynamic

Static	Dynamic
Offline (compile time), before execution	Online (runtime), at execution
Deterministic performance	Nondeterministic performance
Avoid migrations → less overhead & fewer cache misses	Migrations → overhead & cache misses
Regular, periodic tasks with a priori information	Irregular, aperiodic tasks with unknown characteristics a priori
Runtime rescheduling costly	Easy to add new task at runtime

Preemptive vs. Nonpremptive

Preemptive	Nonpreemptive
Can stop an executing process and resume it later	Cannot stop a process in execution
Many context switches → overhead	Fewer context switches → less overhead
Dynamic task arrivals with different priorities	Periodic task arrivals or equal priorities
Simple schedulers	More complex schedulers

Analysis


Scheduling
global
partitioned
static
dynamic
static
dynamic
pre-
emptive
nonpre-
emptive
pre-
emptive
nonpre-
emptive

Schedulers

- **Partitioned Static Scheduler (PSS)**
  - Platform and waveform models (computing resources and requirements)
  - Mapping of processing demands and data flows to processing and interprocessor bandwidth resources
  - Local scheduling
- **Global Dynamic Preemptive Scheduler (GDPS)**
  - Recalculates schedule of all active waveforms each time a user initiates or terminates a session

Complexity Models

- Platform models
  - Cluster of *n* processing elements (part of SDR cloud data center)
- Waveform models
  - *Single-node waveform* vs. *processing chain* of *m* tasks
- Same scheduling complexity per waveform
- Different user arrival rates λ
- Different processing loads ρ *(# users u)*

Complexity Models

* Scheduling complexity per waveform
  * Two models:
    - *Single-node waveform:* $t_1 = A \cdot \boldsymbol{n}$
    - *Distributed waveform processing:* $t_2 = B \cdot \boldsymbol{m} \cdot \boldsymbol{n^2}$

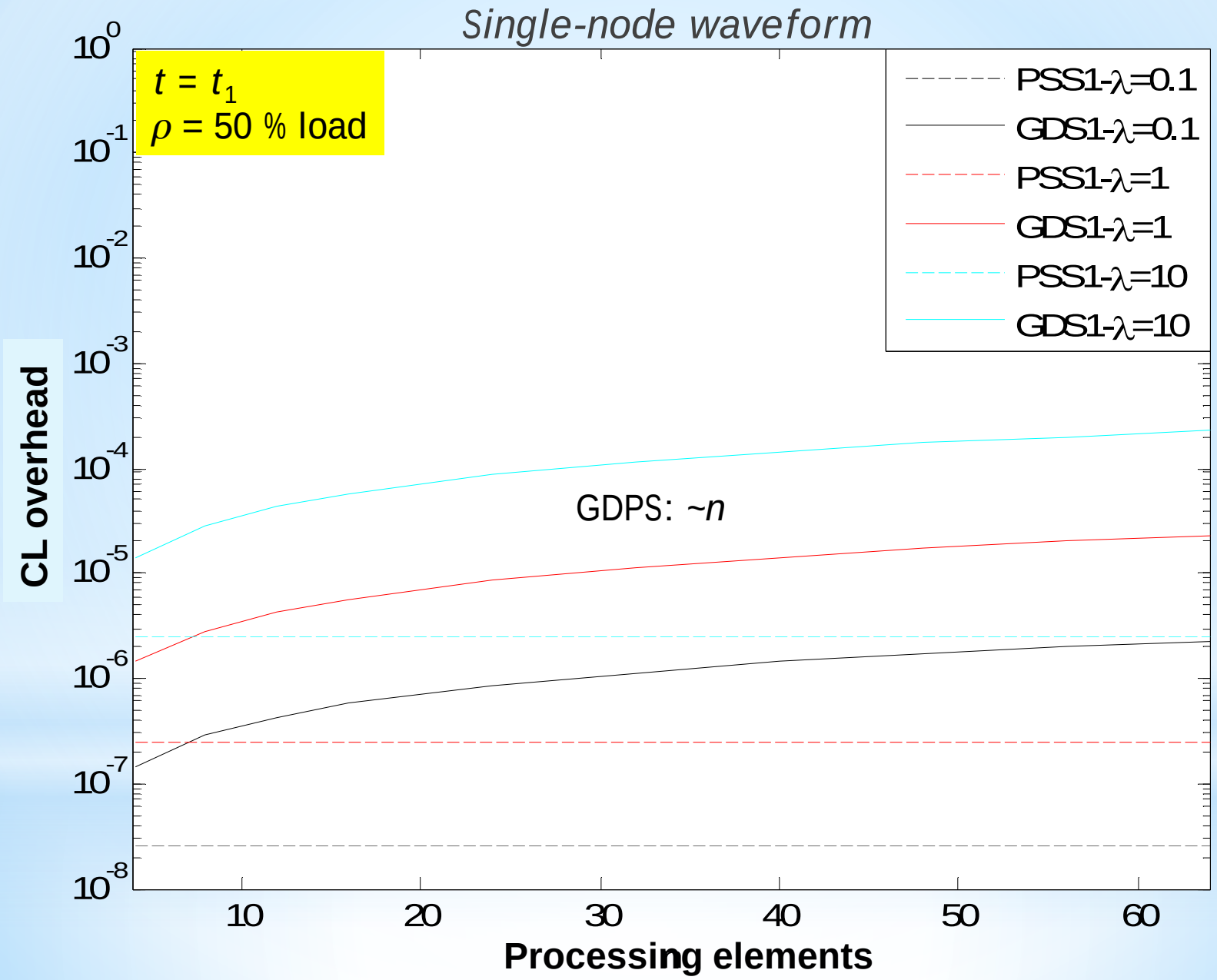
A, B: Scaling factors [s]

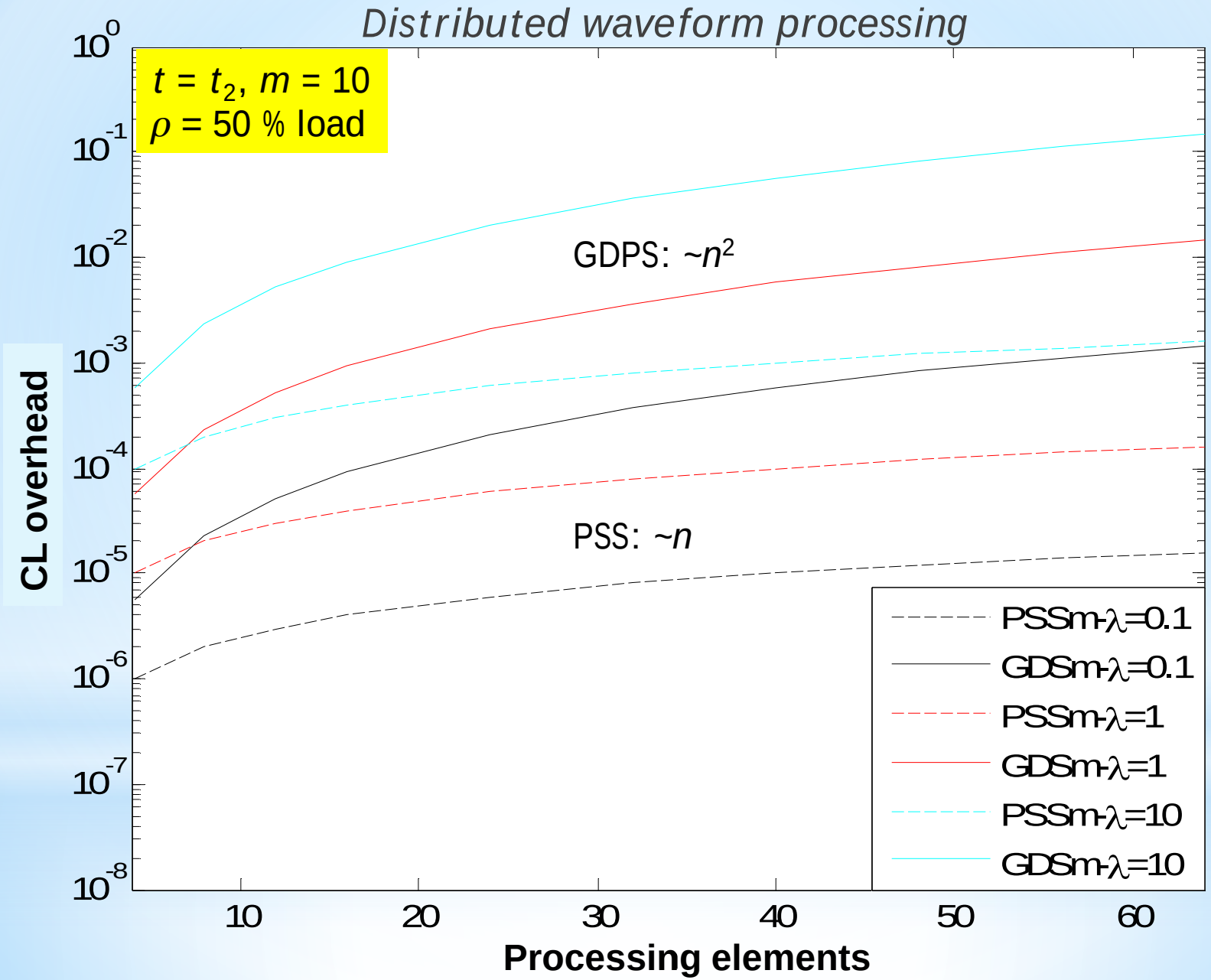
* Cluster processing overhead*:

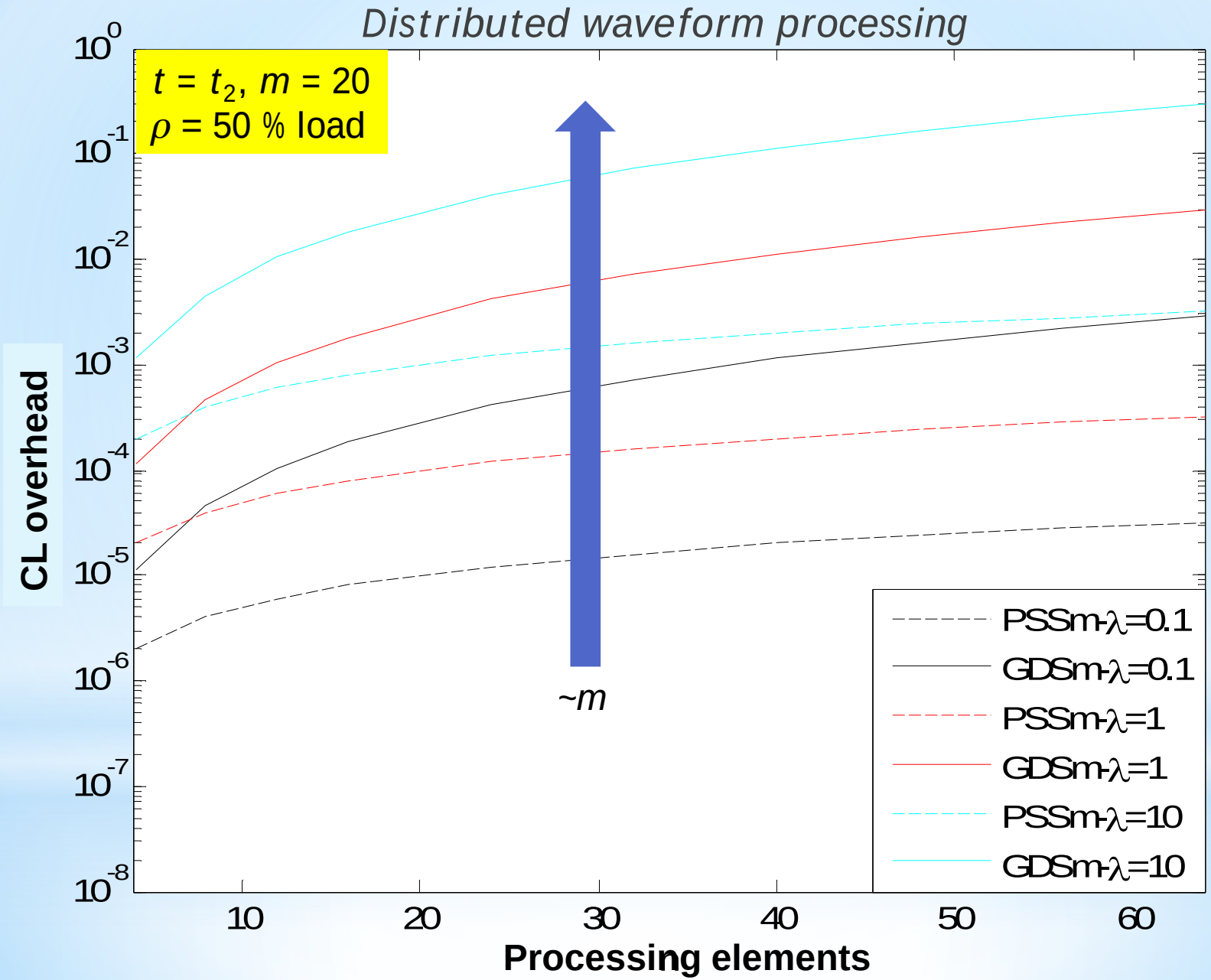
$$CL\text{-}overhead_{PSS} = \boldsymbol{\lambda} \cdot \boldsymbol{t} / n$$

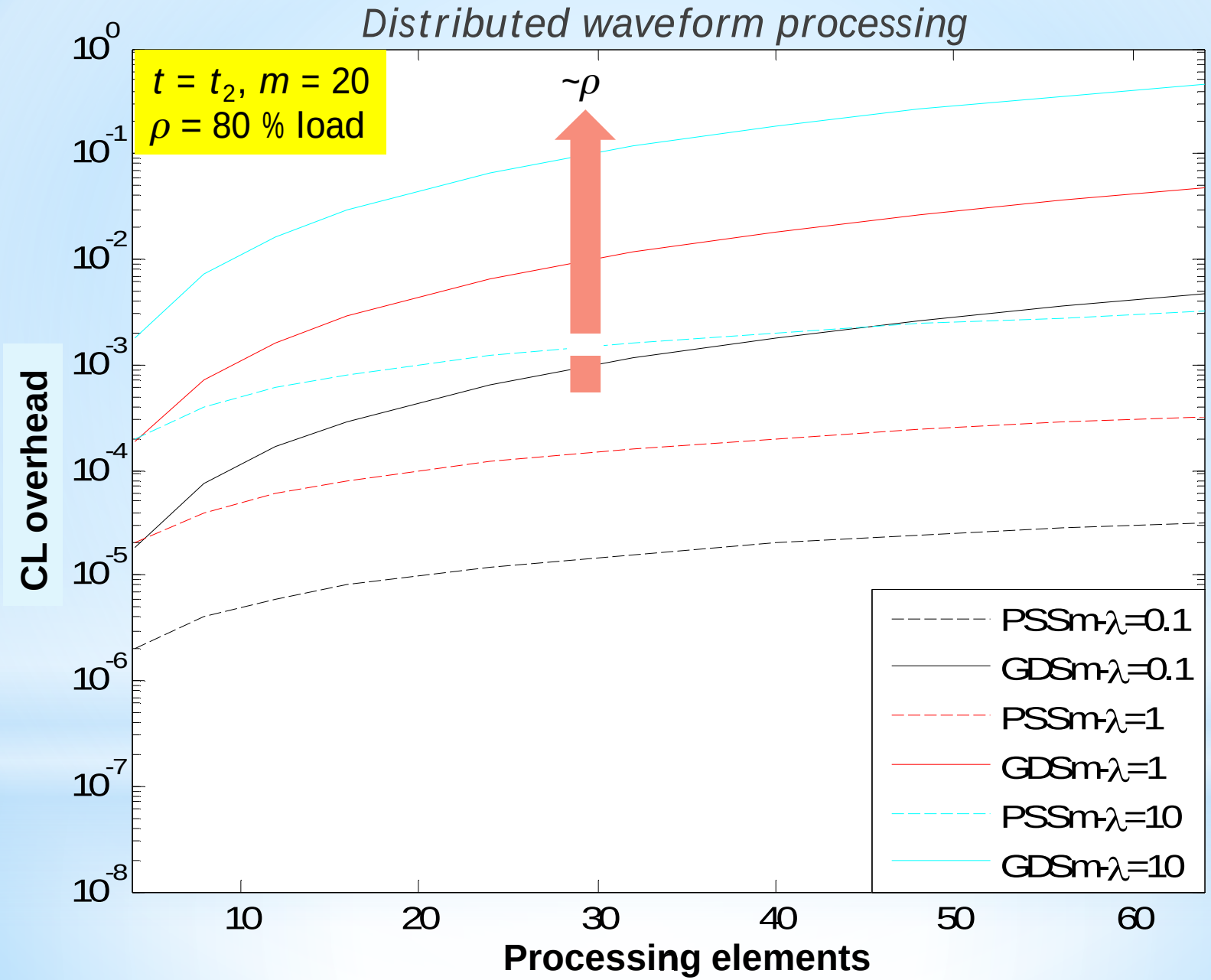
$$CL\text{-}overhead_{GDPS} = 2\boldsymbol{\lambda} \cdot \boldsymbol{u} \cdot \boldsymbol{t} / n \qquad u \sim \boldsymbol{\rho} \cdot \boldsymbol{n}$$

*e.g. 0.1: 10 % cluster processing resources dedicated to scheduling


Single-node waveform
$t = t_1$
ρ = 50 % load
PSS1-λ=0.1
GDS1-λ=0.1
PSS1-λ=1
GDS1-λ=1
PSS1-λ=10
GDS1-λ=10
GDPS: ~n
CL overhead
Processing elements
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10
20
30
40
50
60


Distributed waveform processing
$t = t_2$, $m = 10$
ρ = 50 % load
GDPS: ~n^2
PSS: ~n
PSSm-λ=0.1
GDSm-λ=0.1
PSSm-λ=1
GDSm-λ=1
PSSm-λ=10
GDSm-λ=10
CL overhead
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10
20
30
40
50
60
Processing elements


Distributed waveform processing
$t = t_2$, $m = 20$
ρ = 50 % load
CL overhead
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
~m
PSSm-λ=0.1
GDSm-λ=0.1
PSSm-λ=1
GDSm-λ=1
PSSm-λ=10
GDSm-λ=10
10
20
30
40
50
60
Processing elements


Distributed waveform processing
$t = t_2$, m = 20
ρ = 80 % load
~ρ
CL overhead
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10
20
30
40
50
60
Processing elements
PSSm-λ=0.1
GDSm-λ=0.1
PSSm-λ=1
GDSm-λ=1
PSSm-λ=10
GDSm-λ=10

Conclusions

* Scheduling overhead increases
  * with the waveform granularity (*m*)
  * with the number of processing elements(*n*)
  * inversely to the task execution time
* *Global-dynamic-preemptive* scheduling
  + flexible
  - may incur significant resource overhead
* *Partitioned-static* scheduling scales better