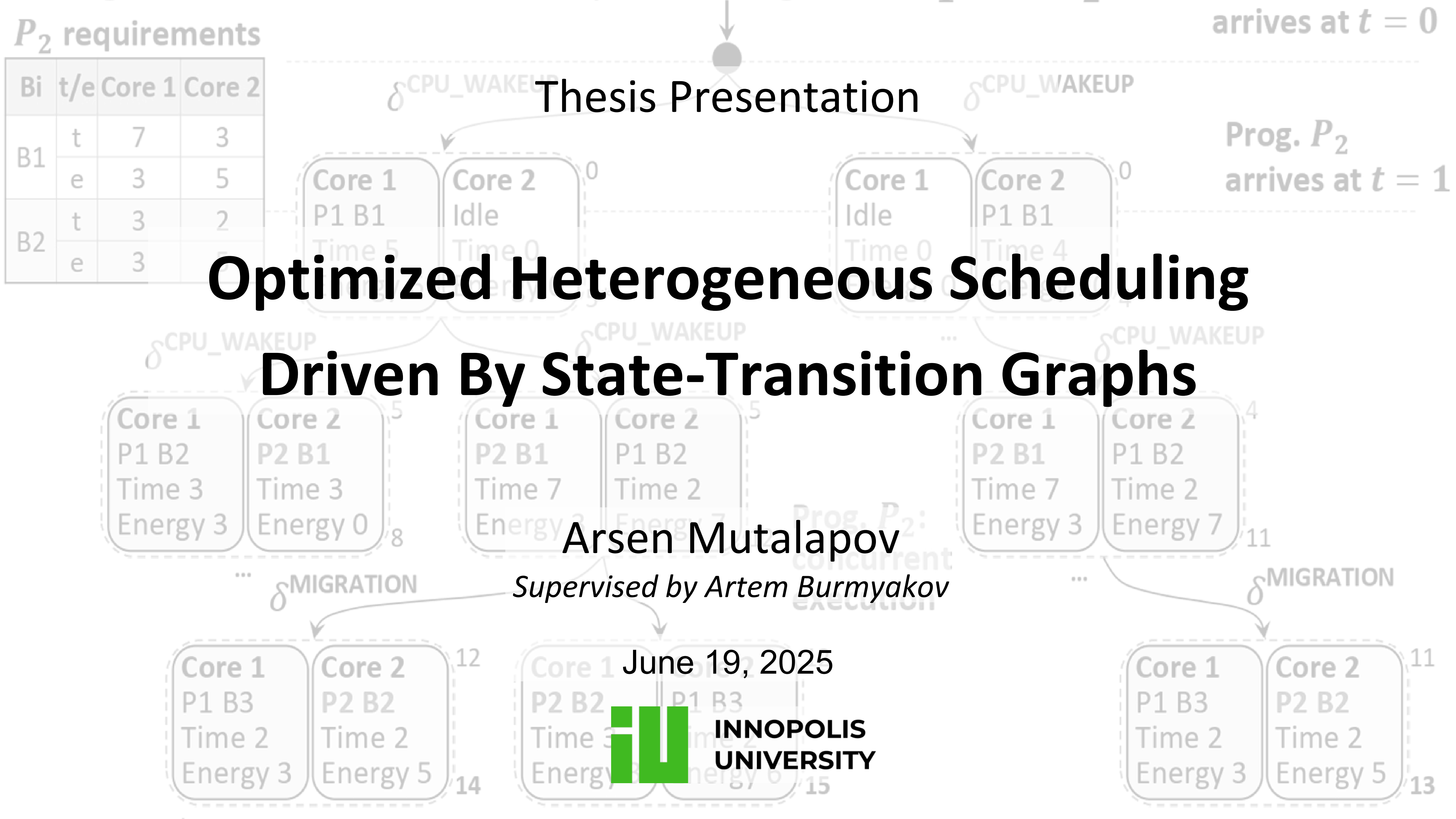


Thesis Presentation

Optimized Heterogeneous Scheduling Driven By State-Transition Graphs

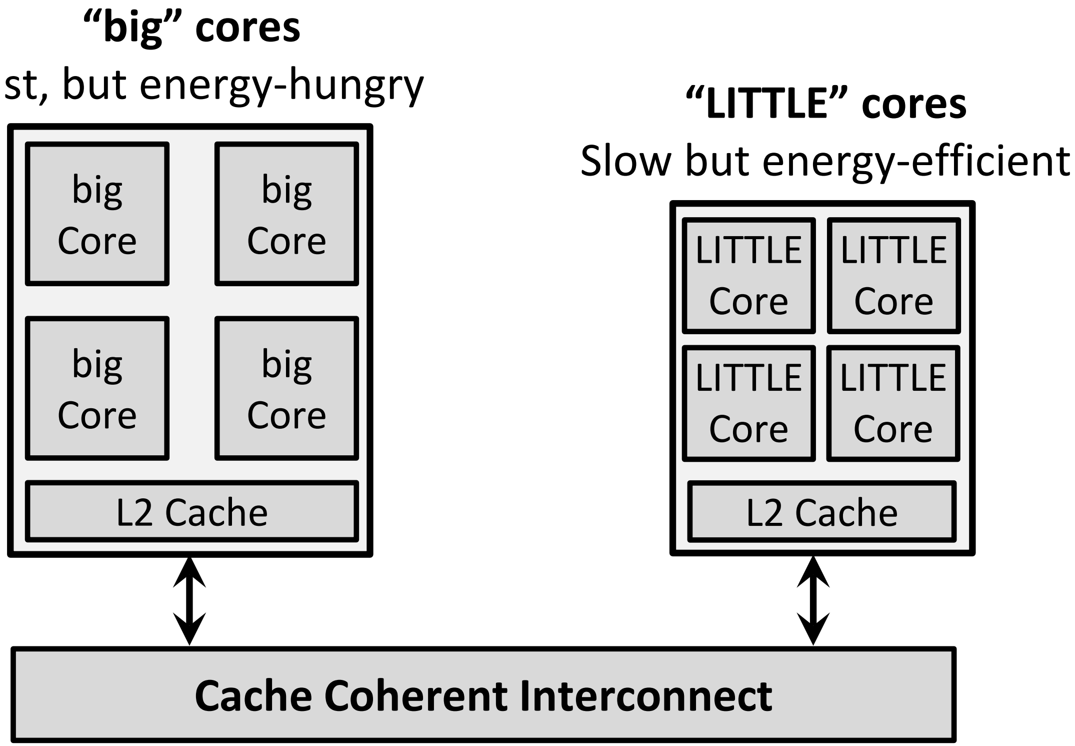
Arsen Mutalapov

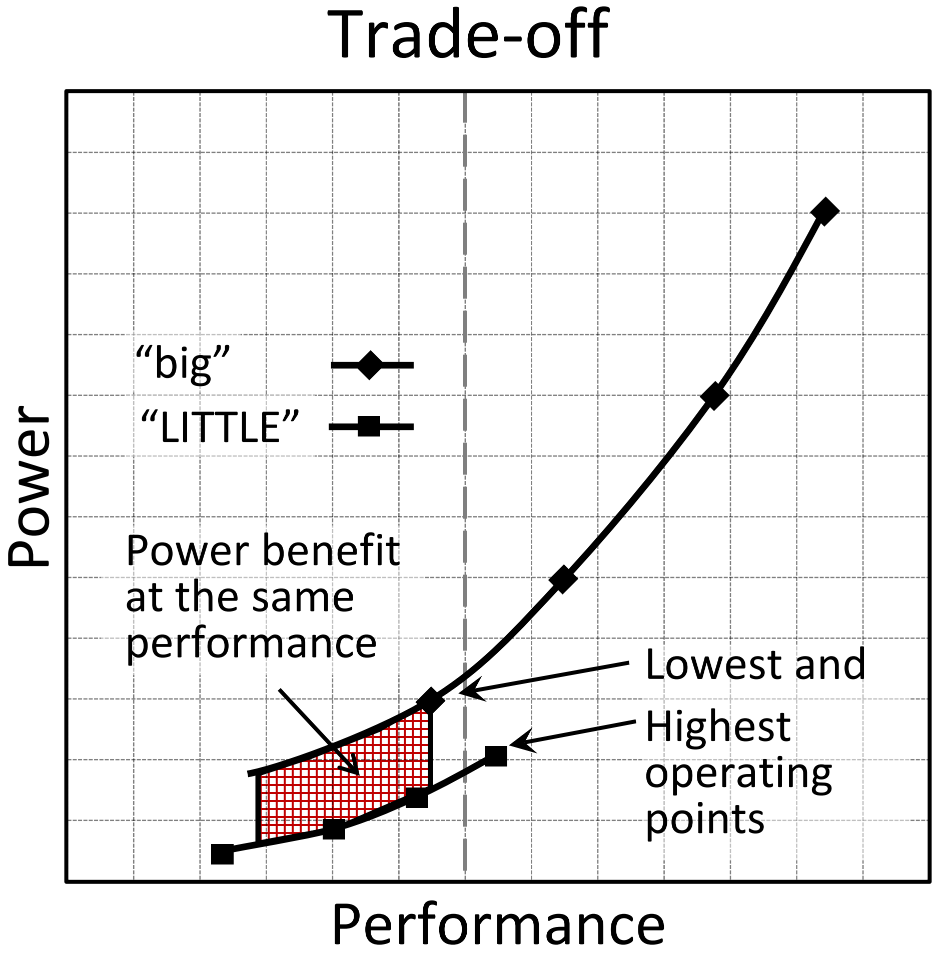
Supervised by Artem Burmyakov

June 19, 2025

INNOPOLIS
UNIVERSITY

Time vs Energy Trade-off: ARM big.LITTLE Case





Heterogeneous Schedules: Time/Energy Trade-off

Prog.	“big” Core		“LITTLE” Core	
	Time	Energy	Time	Energy
A	5	2	8	1
B	4	4	10	3

Heterogeneous Schedules: Time/Energy Trade-off

Prog.	"big" Core		"LITTLE" Core	
	Time	Energy	**Time**	Energy
A	**5**	2	**8**	1
B	**4**	4	**10**	3

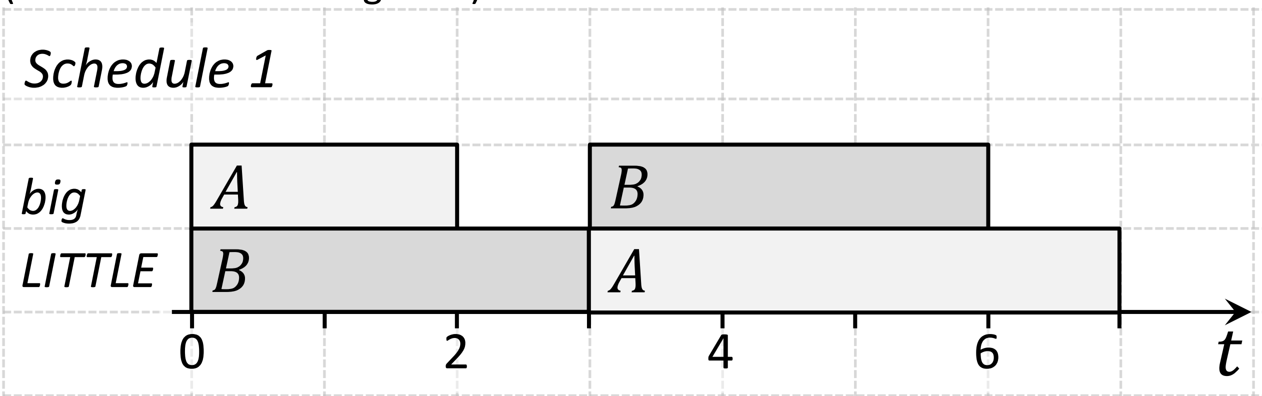
Heterogeneous Schedules: Time/Energy Trade-off

Prog.	“big” Core		“LITTLE” Core	
	Time	**Energy**	Time	**Energy**
A	5	**2**	8	**1**
B	4	**4**	10	**3**

Heterogeneous Schedules: Time/Energy Trade-off

Prog.	"big" Core		"LITTLE" Core	
	Time	Energy	Time	Energy
A	5	2	8	1
B	4	4	10	3

(non-work-conserving case)



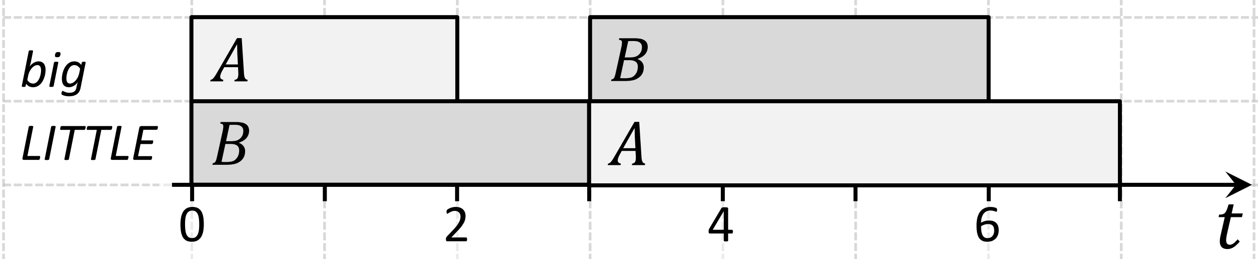
Prog.	**RT**	**E**
A	**7**	**4**
B	**6**	**5**

Heterogeneous Schedules: Time/Energy Trade-off

Prog.	"big" Core		"LITTLE" Core	
	Time	Energy	Time	Energy
A	5	2	8	1
B	4	4	10	3

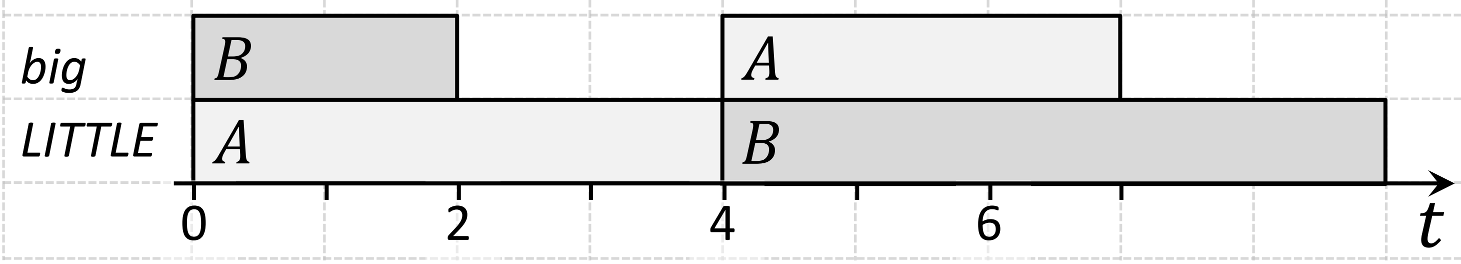
(non-work-conserving case)

Schedule 1



Prog.	RT	E
A	7	4
B	6	5

Schedule 2

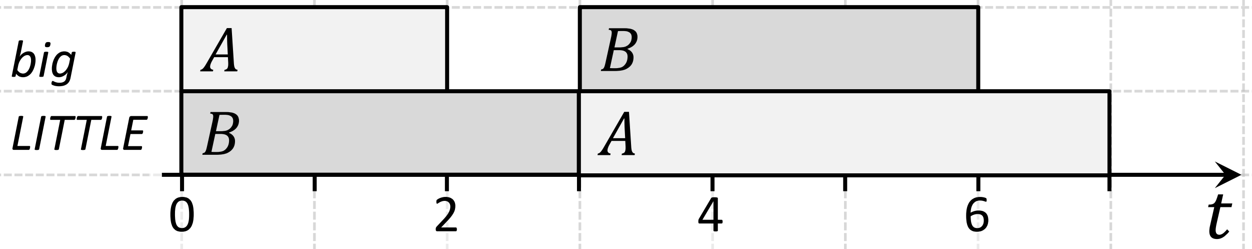


Prog.	**RT**	**E**
A	**7**	**4**
B	**9**	**4**

Heterogeneous Schedules: Time/Energy Trade-off

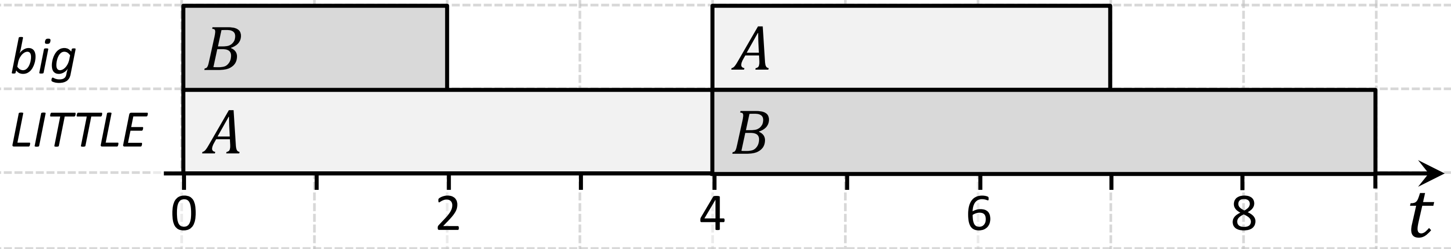
Prog.	"big" Core		"LITTLE" Core	
	Time	Energy	Time	Energy
A	5	2	8	1
B	4	4	10	3

*Schedule 1: Optimized **Avg. RT***



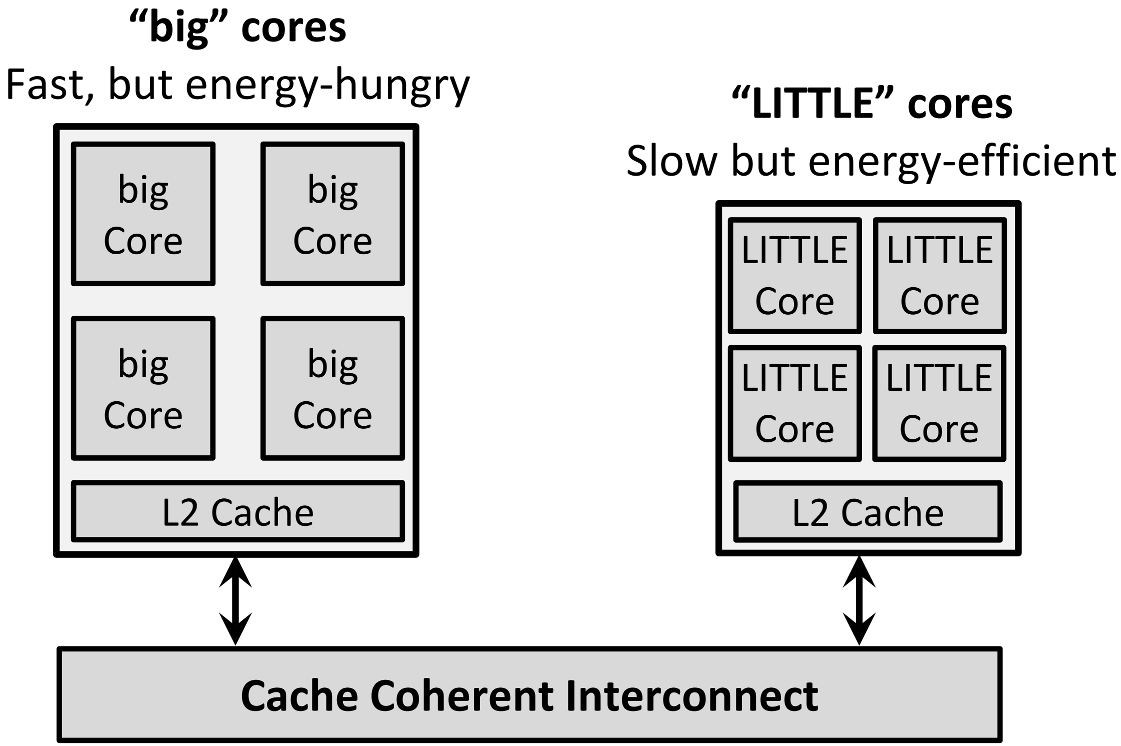
Prog.	**RT**	E
A	**7**	4
B	**6**	5

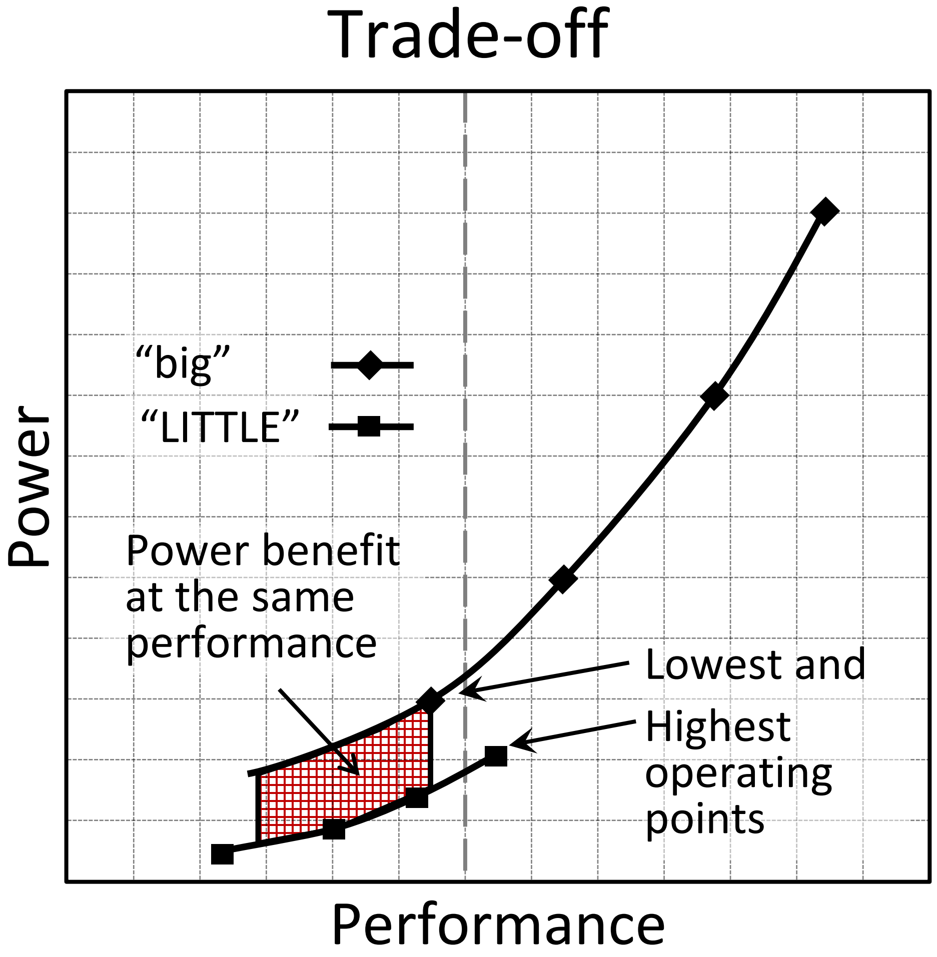
*Schedule 2: Optimized **Avg. Energy***



Prog.	RT	**E**
A	7	**4**
B	9	**4**

Time vs Energy Trade-off: ARM big.LITTLE Case





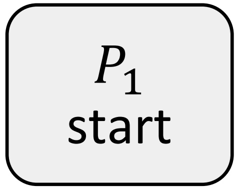
- Scheduler affects time and energy consumption;
- **Efficient scheduler is demanded (e.g. for mobile devices)**

Existing Heterogeneous Schedulers

- HEFT: Heterogeneous Earliest Finish Time;
- HASS: Heterogeneity-Aware Signature-Supported

Properties of HEFT and HASS Schedulers	
Objective	Response times minimization
Principle	Greedy-based heuristics
Limitations	Suboptimal; Energy consumption not considered

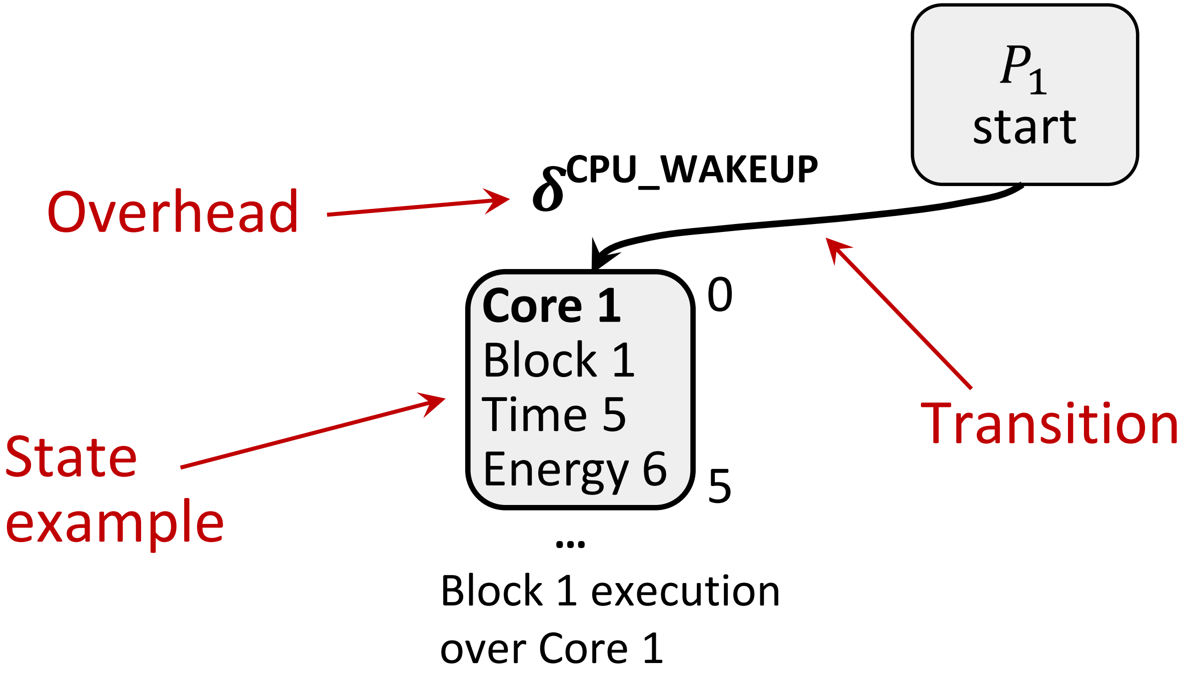
State-Transition Graph for a Solo Program P_1



P_1 requirements

Blocks	Param	Core 1	Core 2
Block 1	Time	5	4
	Energy	6	10
Block 2	Time	3	2
	Energy	3	7
Block 3	Time	2	2
	Energy	3	6

State-Transition Graph for a Solo Program P_1



P_1 start

Overhead

$\delta^{\textbf{CPU\_WAKEUP}}$

Transition

Core 1
Block 1
Time 5
Energy 6

0

5

State example

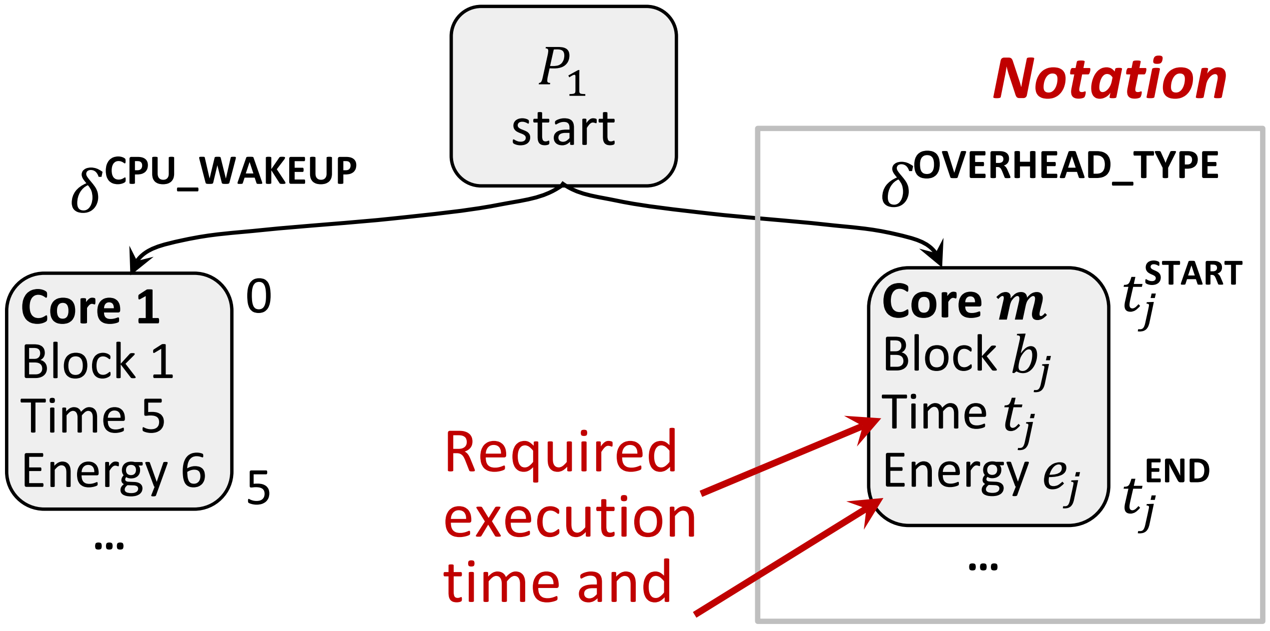
…

Block 1 execution over Core 1

P_1 requirements

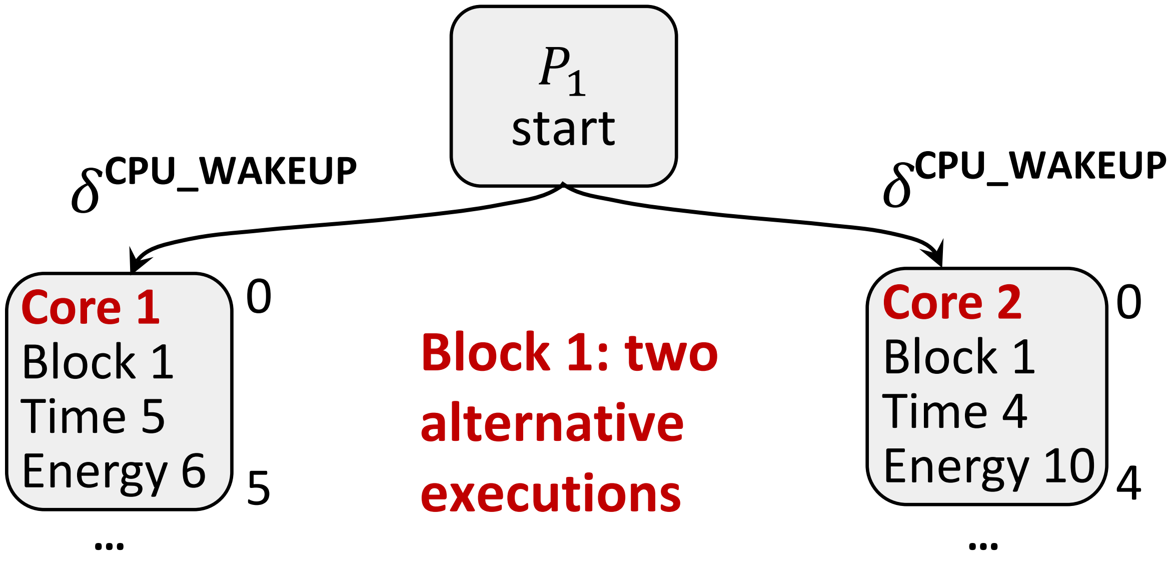
Blocks	Param	Core 1	Core 2
Block 1	**Time**	**5**	4
	Energy	**6**	10
Block 2	Time	3	2
	Energy	3	7
Block 3	Time	2	2
	Energy	3	6

State-Transition Graph for a Solo Program P_1



Required execution time and energy use

State-Transition Graph for a Solo Program P_1



Block 1: two alternative executions

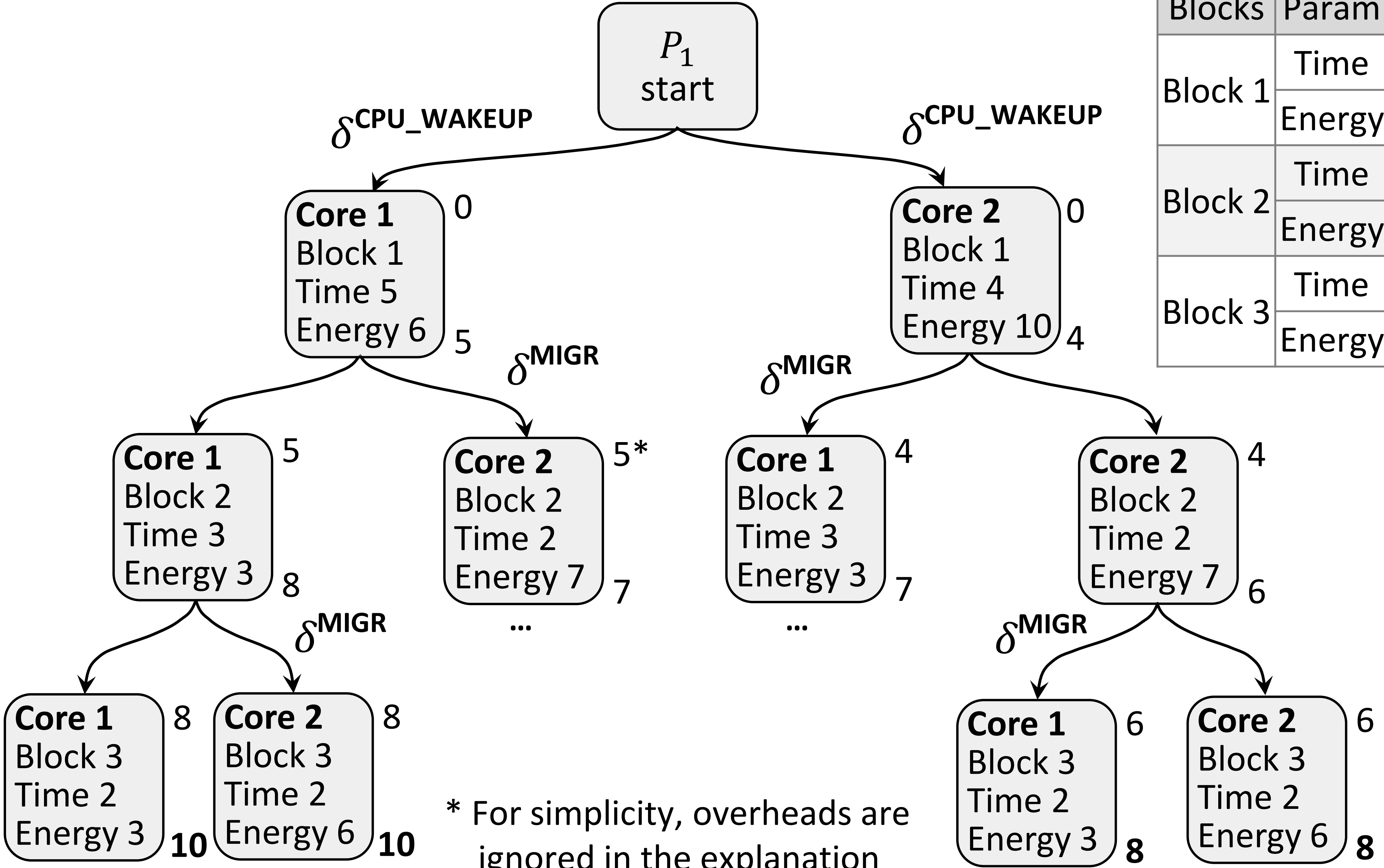
P_1 requirements

Blocks	Param	Core 1	Core 2
Block 1	**Time**	**5**	**4**
	Energy	**6**	**10**
Block 2	Time	3	2
	Energy	3	7
Block 3	Time	2	2
	Energy	3	6

State-Transition Graph for a Solo Program P_1

P_1 requirements

Blocks	Param	Core 1	Core 2
Block 1	Time	5	4
	Energy	6	10
Block 2	Time	3	2
	Energy	3	7
Block 3	Time	2	2
	Energy	3	6



P_1 start

$\delta^{\textbf{CPU\_WAKEUP}}$

Core 1 Block 1 Time 5 Energy 6 (0, 5)

$\delta^{\textbf{MIGR}}$

Core 1 Block 2 Time 3 Energy 3 (5, 8)

Core 1 Block 3 Time 2 Energy 3 (8, **10**)

Core 2 Block 3 Time 2 Energy 6 (8, **10**)

Core 2 Block 2 Time 2 Energy 7 (5*, 7) …

Core 2 Block 1 Time 4 Energy 10 (0, 4)

Core 1 Block 2 Time 3 Energy 3 (4, 7) …

Core 2 Block 2 Time 2 Energy 7 (4, 6)

Core 1 Block 3 Time 2 Energy 3 (6, **8**)

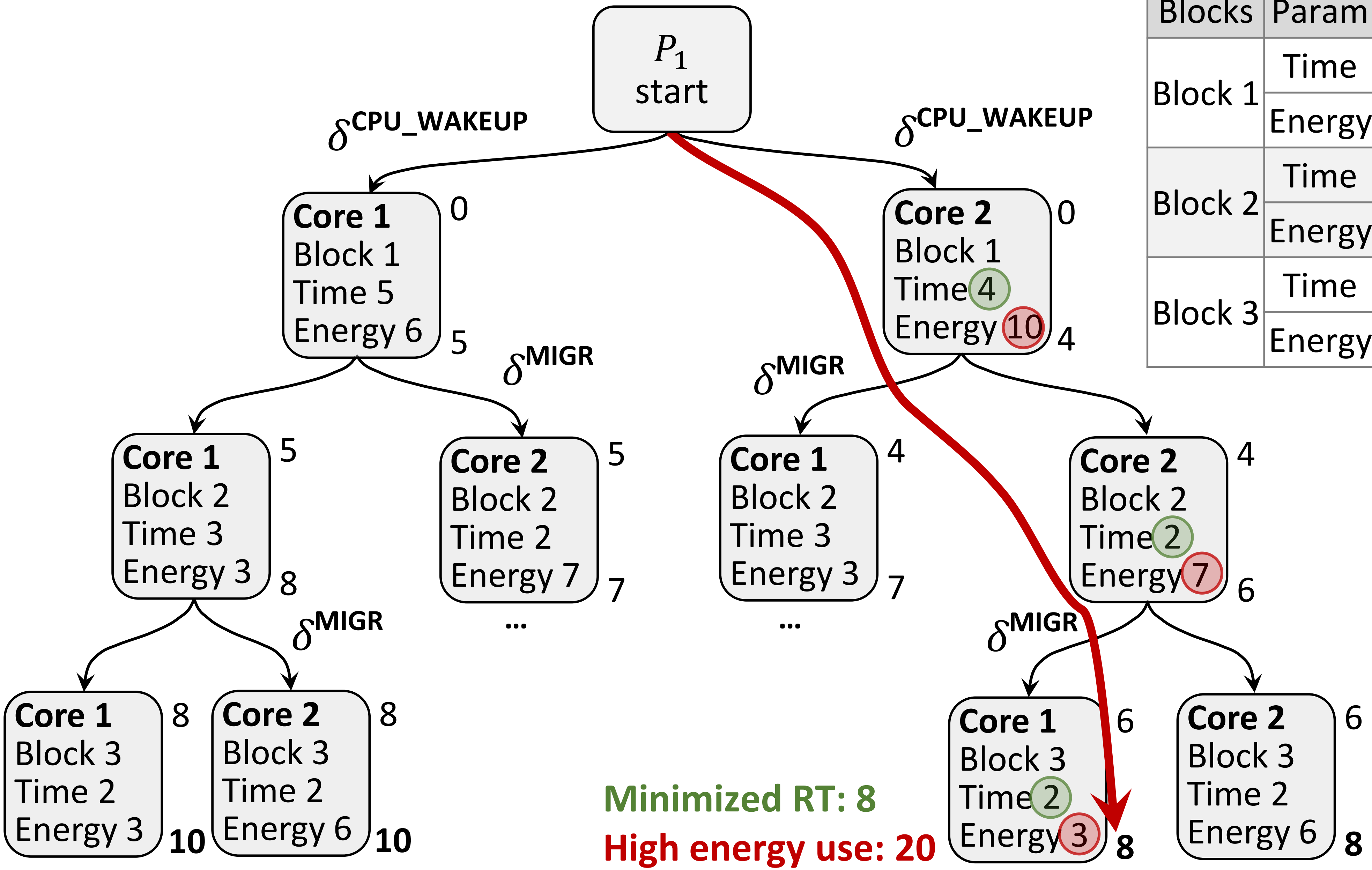
Core 2 Block 3 Time 2 Energy 6 (6, **8**)

* For simplicity, overheads are ignored in the explanation

State-Transition Graph for a Solo Program P_1

P_1 requirements

Blocks	Param	Core 1	Core 2
Block 1	Time	5	4
	Energy	6	10
Block 2	Time	3	2
	Energy	3	7
Block 3	Time	2	2
	Energy	3	6



P_1 start

$\delta^{\textbf{CPU\_WAKEUP}}$

$\delta^{\textbf{CPU\_WAKEUP}}$

Core 1
Block 1
Time 5
Energy 6
0
5

Core 2
Block 1
Time 4
Energy 10
0
4

$\delta^{\textbf{MIGR}}$

$\delta^{\textbf{MIGR}}$

Core 1
Block 2
Time 3
Energy 3
5
8

Core 2
Block 2
Time 2
Energy 7
5
7
…

Core 1
Block 2
Time 3
Energy 3
4
7
…

Core 2
Block 2
Time 2
Energy 7
4
6

$\delta^{\textbf{MIGR}}$

$\delta^{\textbf{MIGR}}$

Core 1
Block 3
Time 2
Energy 3
8
10

Core 2
Block 3
Time 2
Energy 6
8
10

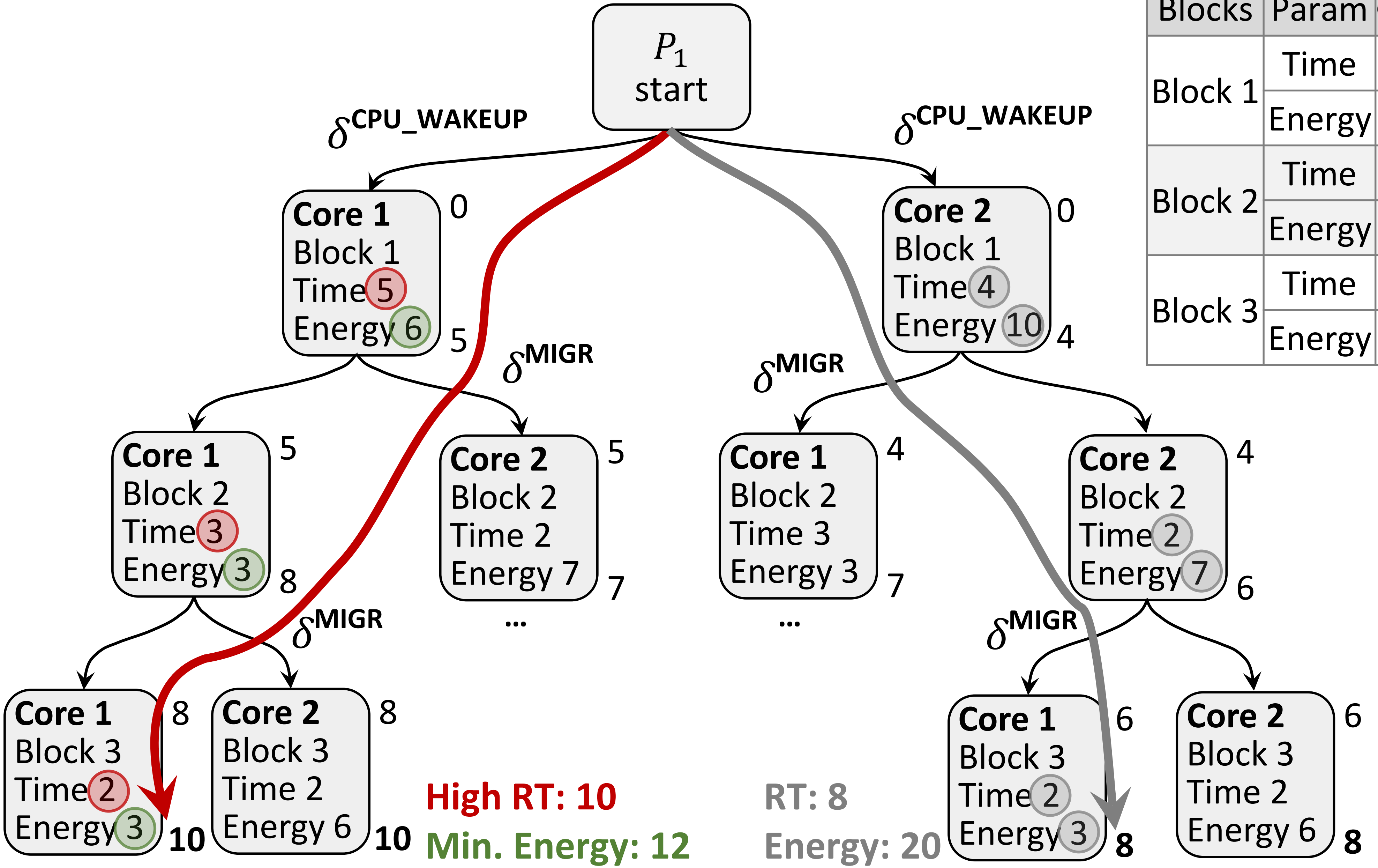
Core 1
Block 3
Time 2
Energy 3
6
8

Core 2
Block 3
Time 2
Energy 6
6
8

Minimized RT: 8

High energy use: 20

State-Transition Graph for a Solo Program P_1



P_1 start

$\delta^{\text{CPU\_WAKEUP}}$

$\delta^{\text{CPU\_WAKEUP}}$

Core 1 Block 1 Time 5 Energy 6 — 0, 5

Core 2 Block 1 Time 4 Energy 10 — 0, 4

δ^{MIGR}

δ^{MIGR}

Core 1 Block 2 Time 3 Energy 3 — 5, 8

Core 2 Block 2 Time 2 Energy 7 — 5, 7

...

Core 1 Block 2 Time 3 Energy 3 — 4, 7

...

Core 2 Block 2 Time 2 Energy 7 — 4, 6

δ^{MIGR}

δ^{MIGR}

Core 1 Block 3 Time 2 Energy 3 — 8, **10**

Core 2 Block 3 Time 2 Energy 6 — 8, **10**

Core 1 Block 3 Time 2 Energy 3 — 6, **8**

Core 2 Block 3 Time 2 Energy 6 — 6, **8**

High RT: 10

Min. Energy: 12

RT: 8

Energy: 20

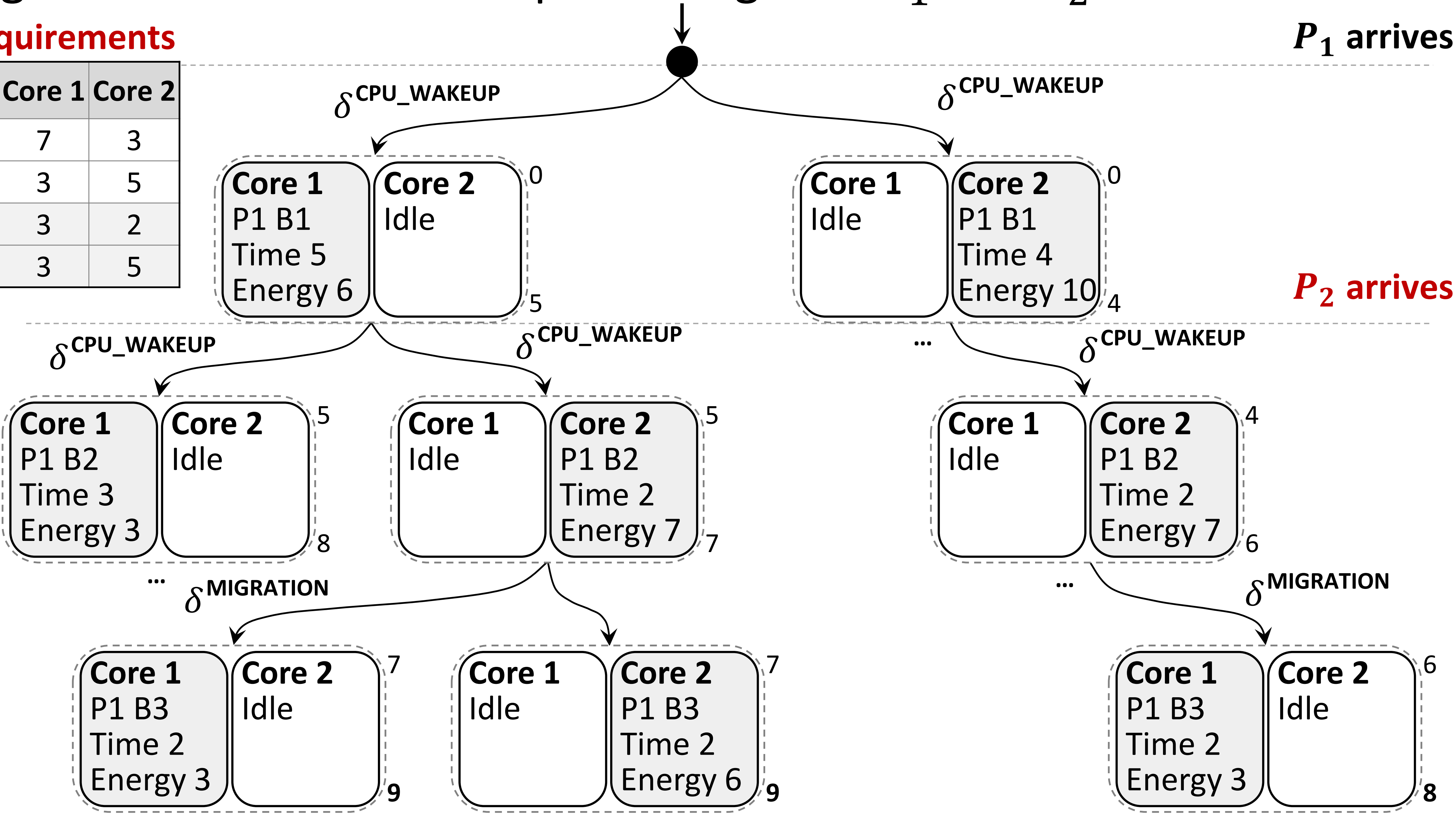
P_1 requirements

Blocks	Param	Core 1	Core 2
Block 1	Time	5	4
	Energy	6	10
Block 2	Time	3	2
	Energy	3	7
Block 3	Time	2	2
	Energy	3	6

Merged State-Transition Graph of Programs P_1 and P_2

P_2 requirements

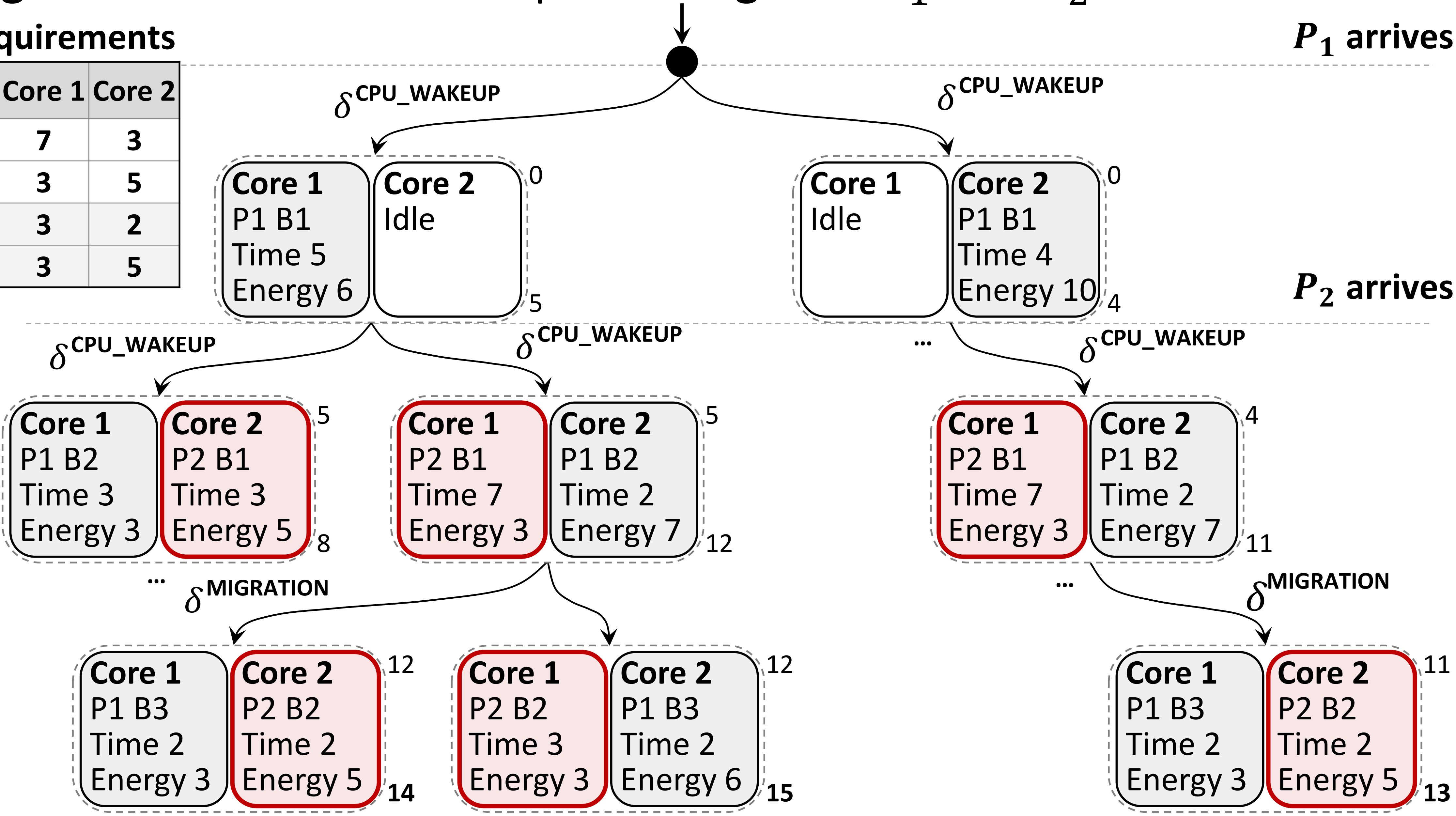
Bi	t/e	Core 1	Core 2
B1	t	7	3
	e	3	5
B2	t	3	2
	e	3	5



Merged State-Transition Graph of Programs P_1 and P_2

P_2 requirements

Bi	t/e	Core 1	Core 2
B1	t	**7**	**3**
	e	**3**	**5**
B2	t	**3**	**2**
	e	**3**	**5**



P_1 arrives

P_2 arrives

$\delta^{\text{CPU\_WAKEUP}}$

Core 1 P1 B1 Time 5 Energy 6 | **Core 2** Idle — 0, 5

$\delta^{\text{CPU\_WAKEUP}}$

Core 1 Idle | **Core 2** P1 B1 Time 4 Energy 10 — 0, 4

$\delta^{\text{CPU\_WAKEUP}}$

Core 1 P1 B2 Time 3 Energy 3 | **Core 2** P2 B1 Time 3 Energy 5 — 5, 8

$\delta^{\text{CPU\_WAKEUP}}$

Core 1 P2 B1 Time 7 Energy 3 | **Core 2** P1 B2 Time 2 Energy 7 — 5, 12

...

$\delta^{\text{CPU\_WAKEUP}}$

Core 1 P2 B1 Time 7 Energy 3 | **Core 2** P1 B2 Time 2 Energy 7 — 4, 11

...

$\delta^{\text{MIGRATION}}$

Core 1 P1 B3 Time 2 Energy 3 | **Core 2** P2 B2 Time 2 Energy 5 — 12, **14**

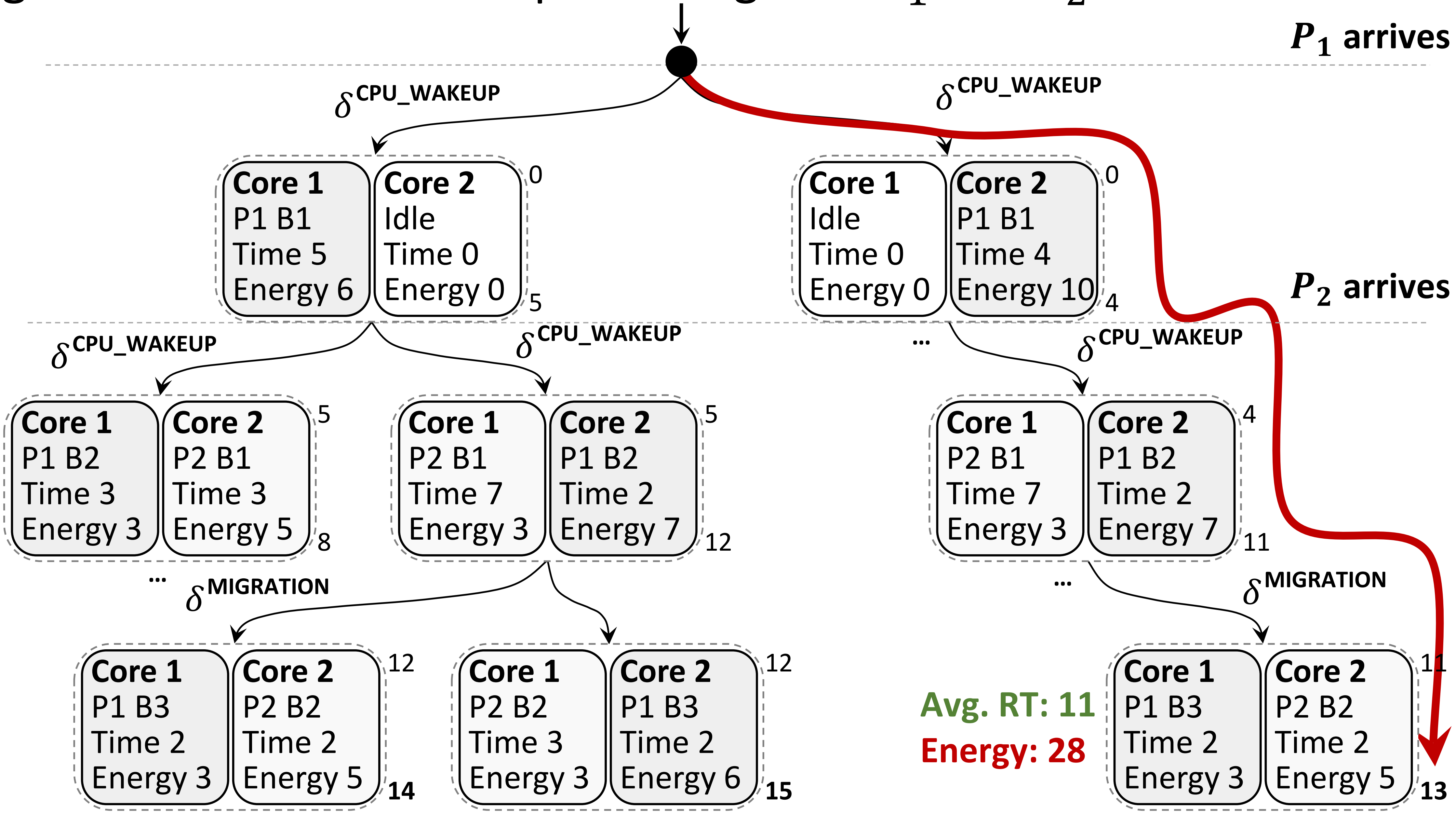
Core 1 P2 B2 Time 3 Energy 3 | **Core 2** P1 B3 Time 2 Energy 6 — 12, **15**

...

$\delta^{\text{MIGRATION}}$

Core 1 P1 B3 Time 2 Energy 3 | **Core 2** P2 B2 Time 2 Energy 5 — 11, **13**

Merged State-Transition Graph of Programs P_1 and P_2



P_1 **arrives**

P_2 **arrives**

$\delta^{\text{CPU\_WAKEUP}}$

Core 1 P1 B1 Time 5 Energy 6 | Core 2 Idle Time 0 Energy 0 — 0, 5

$\delta^{\text{CPU\_WAKEUP}}$

Core 1 Idle Time 0 Energy 0 | Core 2 P1 B1 Time 4 Energy 10 — 0, 4

$\delta^{\text{CPU\_WAKEUP}}$

Core 1 P1 B2 Time 3 Energy 3 | Core 2 P2 B1 Time 3 Energy 5 — 5, 8

...

$\delta^{\text{CPU\_WAKEUP}}$

Core 1 P2 B1 Time 7 Energy 3 | Core 2 P1 B2 Time 2 Energy 7 — 5, 12

...

$\delta^{\text{CPU\_WAKEUP}}$

Core 1 P2 B1 Time 7 Energy 3 | Core 2 P1 B2 Time 2 Energy 7 — 4, 11

$\delta^{\text{MIGRATION}}$

Core 1 P1 B3 Time 2 Energy 3 | Core 2 P2 B2 Time 2 Energy 5 — 12, **14**

Core 1 P2 B2 Time 3 Energy 3 | Core 2 P1 B3 Time 2 Energy 6 — 12, **15**

...

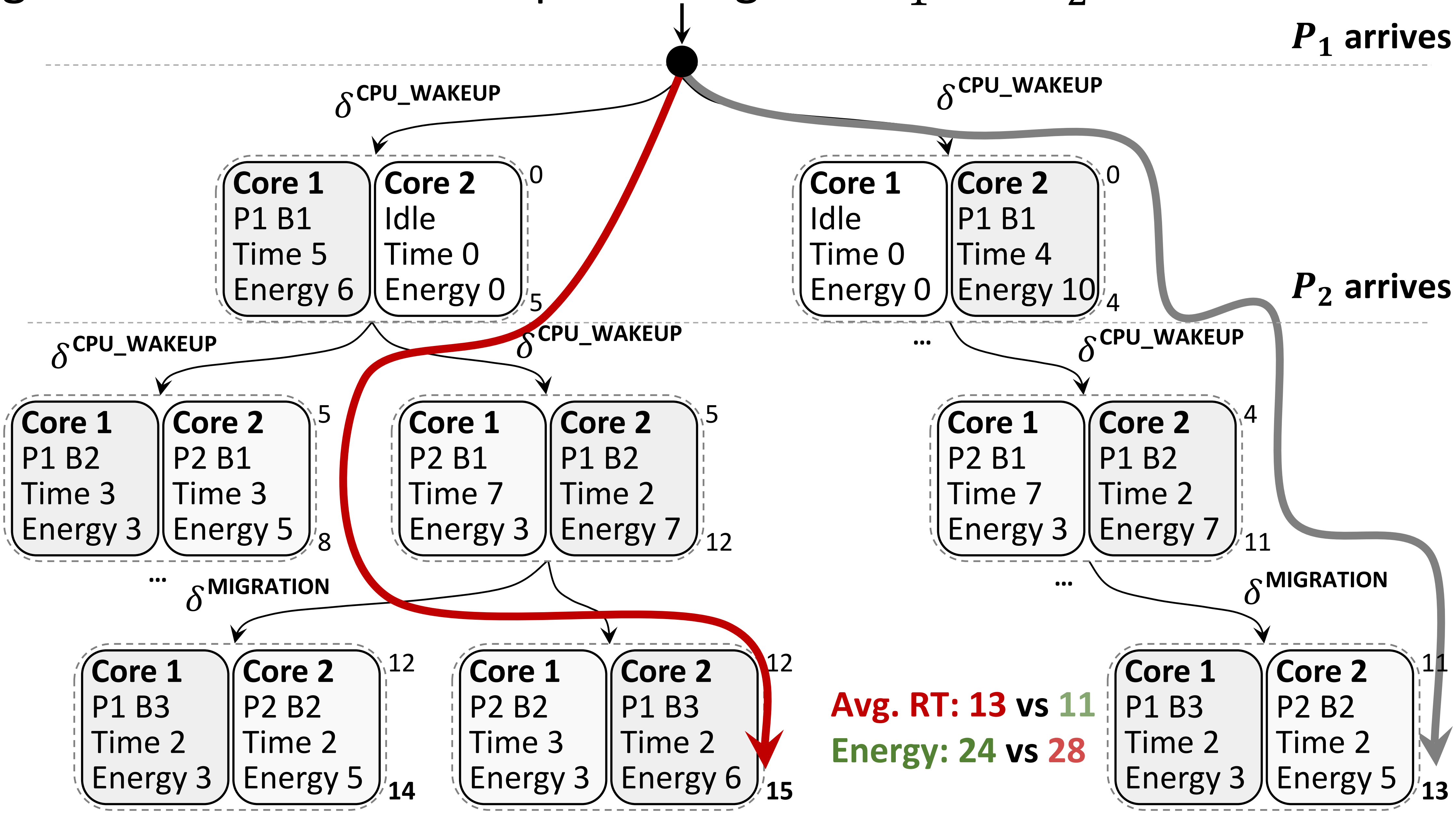
$\delta^{\text{MIGRATION}}$

Core 1 P1 B3 Time 2 Energy 3 | Core 2 P2 B2 Time 2 Energy 5 — 11, **13**

Avg. RT: 11

Energy: 28

Merged State-Transition Graph of Programs P_1 and P_2



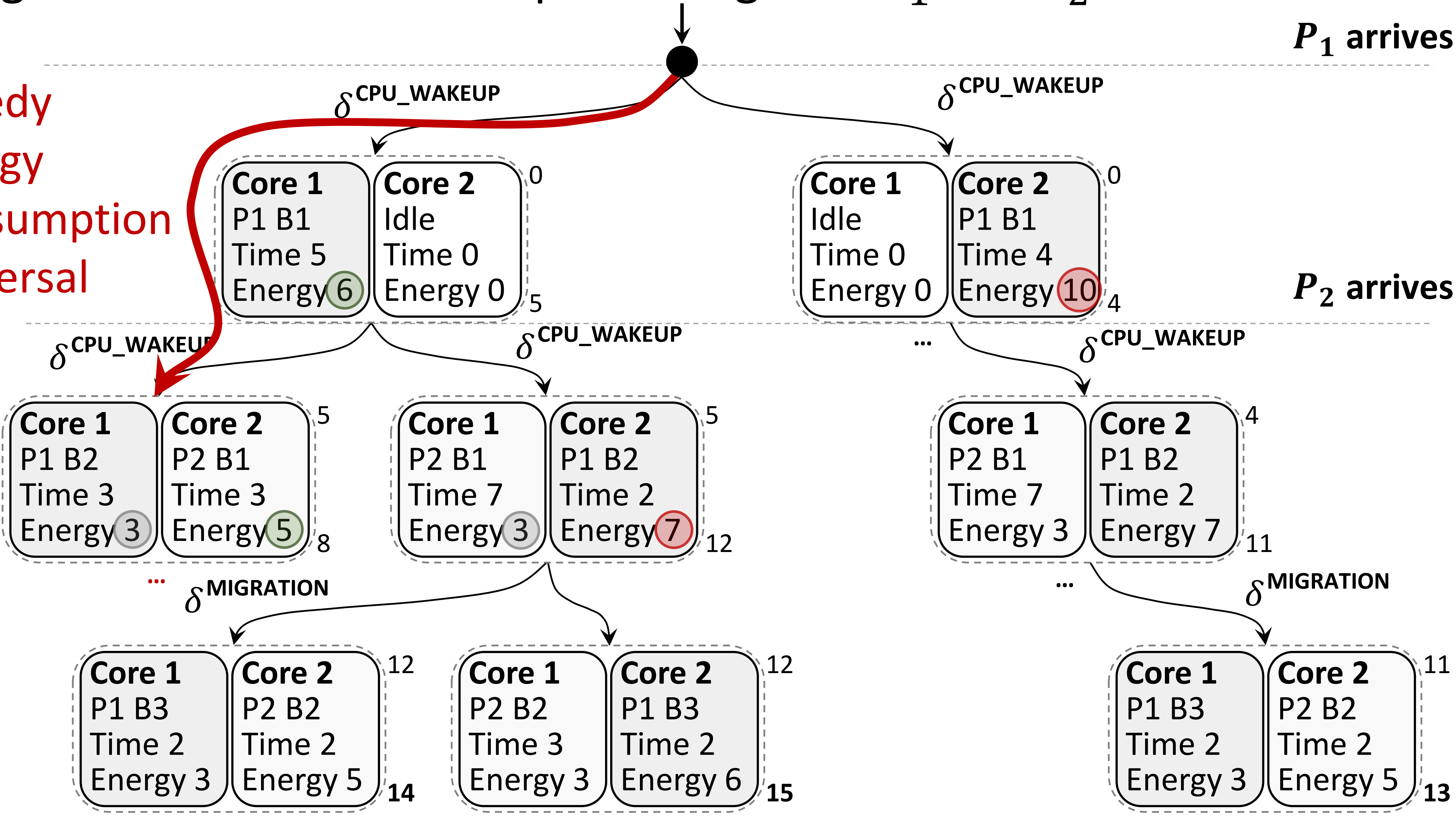
P_1 arrives
P_2 arrives
$\delta^{\text{CPU\_WAKEUP}}$
$\delta^{\text{CPU\_WAKEUP}}$
Core 1 P1 B1 Time 5 Energy 6
Core 2 Idle Time 0 Energy 0
0
5
Core 1 Idle Time 0 Energy 0
Core 2 P1 B1 Time 4 Energy 10
0
4
$\delta^{\text{CPU\_WAKEUP}}$
$\delta^{\text{CPU\_WAKEUP}}$
...
$\delta^{\text{CPU\_WAKEUP}}$
Core 1 P1 B2 Time 3 Energy 3
Core 2 P2 B1 Time 3 Energy 5
5
8
Core 1 P2 B1 Time 7 Energy 3
Core 2 P1 B2 Time 2 Energy 7
5
12
Core 1 P2 B1 Time 7 Energy 3
Core 2 P1 B2 Time 2 Energy 7
4
11
...
$\delta^{\text{MIGRATION}}$
...
$\delta^{\text{MIGRATION}}$
Core 1 P1 B3 Time 2 Energy 3
Core 2 P2 B2 Time 2 Energy 5
12
14
Core 1 P2 B2 Time 3 Energy 3
Core 2 P1 B3 Time 2 Energy 6
12
15
Core 1 P1 B3 Time 2 Energy 3
Core 2 P2 B2 Time 2 Energy 5
11
13

Avg. RT: 13 vs 11

Energy: 24 vs 28

Merged State-Transition Graph of Programs P_1 and P_2

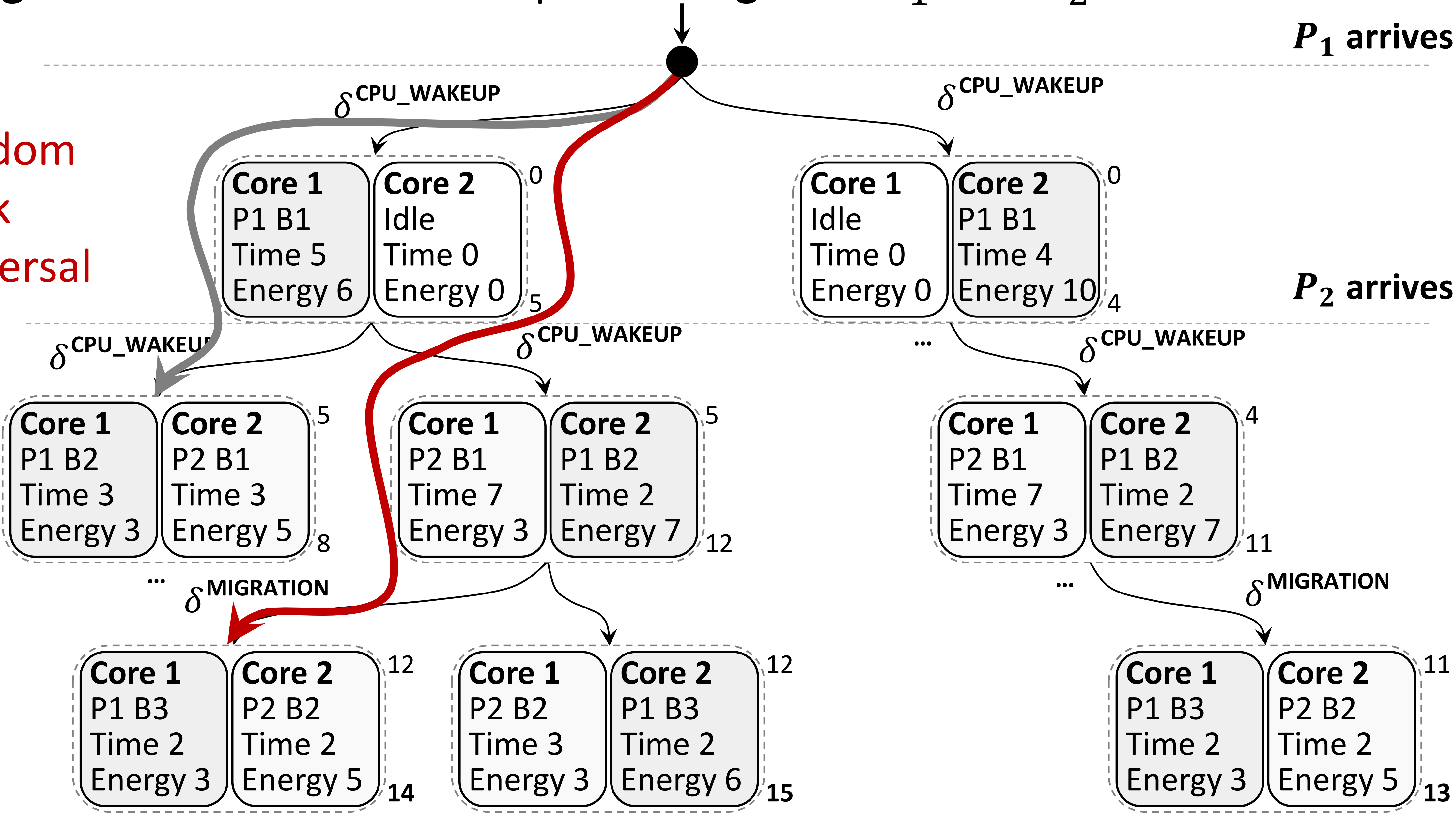
Greedy Energy Consumption Traversal



P_1 arrives
P_2 arrives
$\delta^{\text{CPU\_WAKEUP}}$
$\delta^{\text{CPU\_WAKEUP}}$
Core 1 P1 B1 Time 5 Energy 6
Core 2 Idle Time 0 Energy 0
0
5
Core 1 Idle Time 0 Energy 0
Core 2 P1 B1 Time 4 Energy 10
0
4
$\delta^{\text{CPU\_WAKEUP}}$
$\delta^{\text{CPU\_WAKEUP}}$
...
$\delta^{\text{CPU\_WAKEUP}}$
Core 1 P1 B2 Time 3 Energy 3
Core 2 P2 B1 Time 3 Energy 5
5
8
Core 1 P2 B1 Time 7 Energy 3
Core 2 P1 B2 Time 2 Energy 7
5
12
Core 1 P2 B1 Time 7 Energy 3
Core 2 P1 B2 Time 2 Energy 7
4
11
...
$\delta^{\text{MIGRATION}}$
...
$\delta^{\text{MIGRATION}}$
Core 1 P1 B3 Time 2 Energy 3
Core 2 P2 B2 Time 2 Energy 5
12
14
Core 1 P2 B2 Time 3 Energy 3
Core 2 P1 B3 Time 2 Energy 6
12
15
Core 1 P1 B3 Time 2 Energy 3
Core 2 P2 B2 Time 2 Energy 5
11
13

Merged State-Transition Graph of Programs P_1 and P_2

Random Walk Traversal

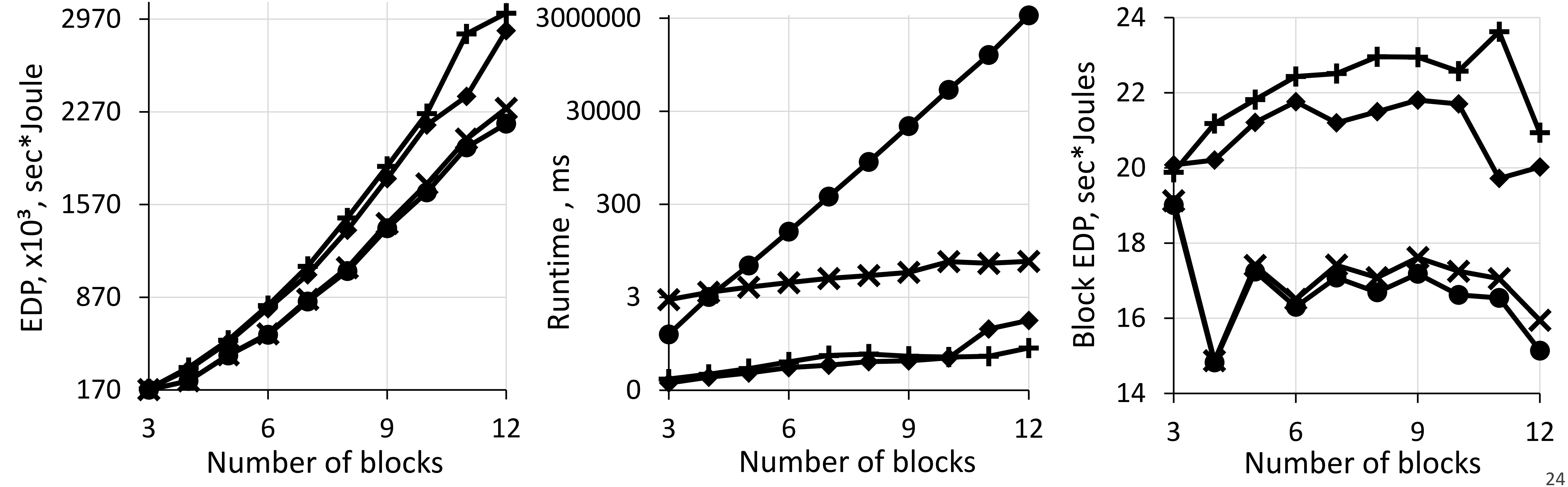

P_1 arrives
P_2 arrives
$\delta^{\text{CPU\_WAKEUP}}$
$\delta^{\text{CPU\_WAKEUP}}$
Core 1 P1 B1 Time 5 Energy 6
Core 2 Idle Time 0 Energy 0
0
5
Core 1 Idle Time 0 Energy 0
Core 2 P1 B1 Time 4 Energy 10
0
4
$\delta^{\text{CPU\_WAKEUP}}$
$\delta^{\text{CPU\_WAKEUP}}$
...
$\delta^{\text{CPU\_WAKEUP}}$
Core 1 P1 B2 Time 3 Energy 3
Core 2 P2 B1 Time 3 Energy 5
5
8
Core 1 P2 B1 Time 7 Energy 3
Core 2 P1 B2 Time 2 Energy 7
5
12
Core 1 P2 B1 Time 7 Energy 3
Core 2 P1 B2 Time 2 Energy 7
4
11
...
$\delta^{\text{MIGRATION}}$
...
$\delta^{\text{MIGRATION}}$
Core 1 P1 B3 Time 2 Energy 3
Core 2 P2 B2 Time 2 Energy 5
12
14
Core 1 P2 B2 Time 3 Energy 3
Core 2 P1 B3 Time 2 Energy 6
12
15
Core 1 P1 B3 Time 2 Energy 3
Core 2 P2 B2 Time 2 Energy 5
11
13

Schedulers Evaluation

Settings:	Blocks Requirements per Core 1,2:	
Programs number – 3;	Runtime –	1: [220; 250], 2: [210; 230];
Cores number – 2;	Energy use –	1: [120; 130], 2: [125; 150].

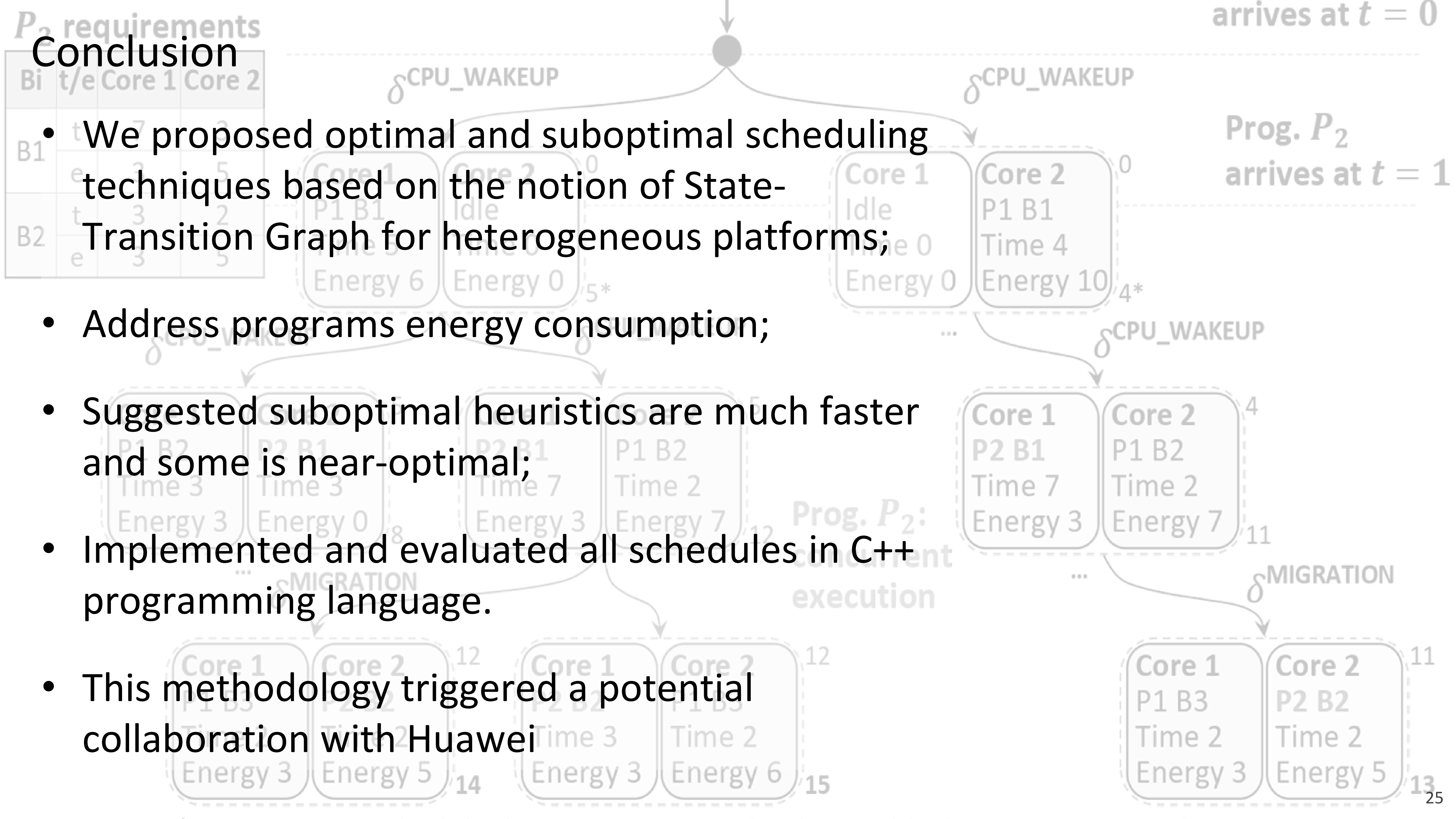
Input generation: numbers of programs and cores are fixed, blocks requirements are picked from uniform distributions

- ● Optimal
- ◆ 1-Random
- ✕ 100-Random
- ✚ Greedy



Conclusion

- We proposed optimal and suboptimal scheduling techniques based on the notion of State-Transition Graph for heterogeneous platforms;
- Address programs energy consumption;
- Suggested suboptimal heuristics are much faster and some is near-optimal;
- Implemented and evaluated all schedules in C++ programming language.
- This methodology triggered a potential collaboration with Huawei



References

1. E. L. Padoin, L. L. Pilla, M. Castro, F. Z. Boito, P. O. Alexandre Navaux, and J.-F. M´ehaut, “Performance/energy trade-off in scientific computing: the case of arm big.little and intel sandy bridge,” IET Computers & Digital Techniques, vol. 9, no. 1, pp. 27–35, 2015. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/ietcdt. 2014.0074

2. H. Topcuoglu, S. Hariri, and M.-Y. Wu, “Task scheduling algorithms for heterogeneous processors,” in Proceedings. Eighth Heterogeneous Computing Workshop (HCW’99), 1999, pp. 3–14;

3. D. Shelepov, J. C. Saez Alcaide, S. Jeffery, A. Fedorova, N. Perez, Z. F. Huang, S. Blagodurov, and V. Kumar, “Hass: a scheduler for heterogeneous multicore systems,” SIGOPS Oper. Syst. Rev., vol. 43, no. 2, p. 66–75, Apr. 2009. [Online]. Available: https://doi.org/10.1145/1531793.1531804