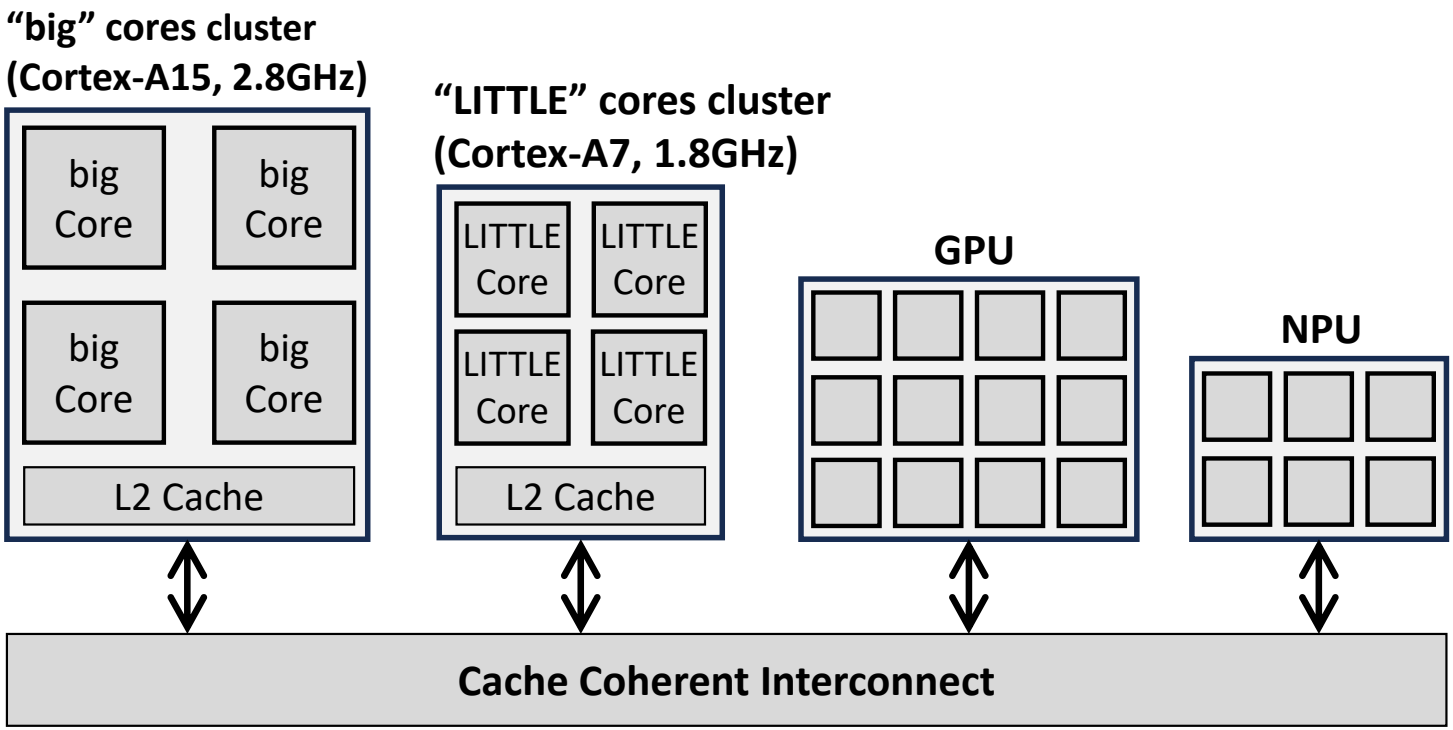
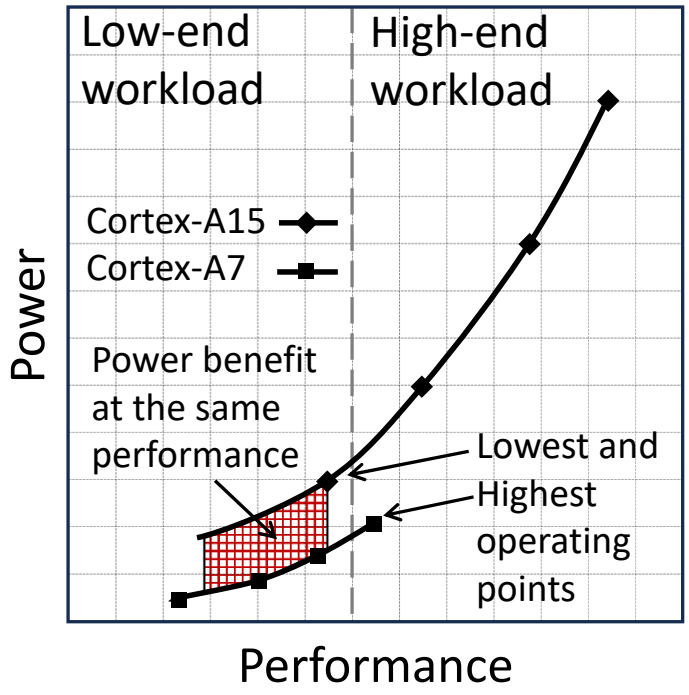
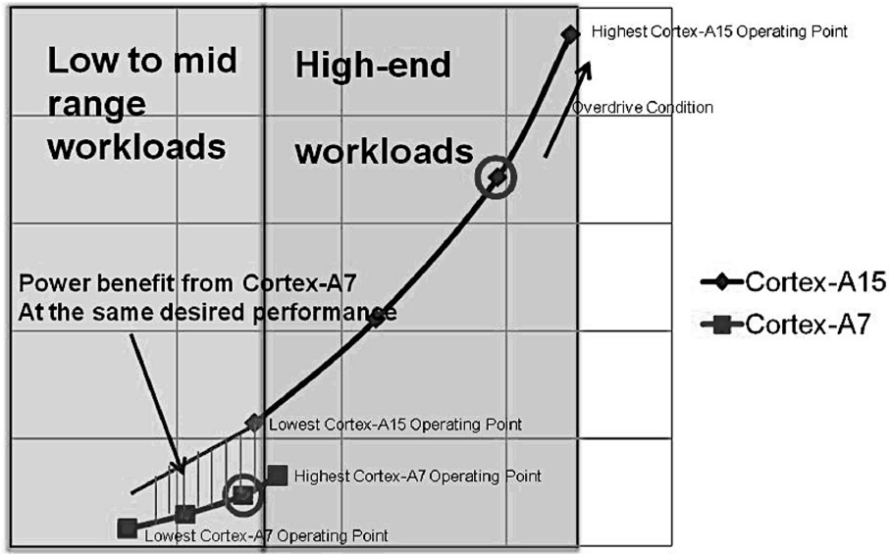
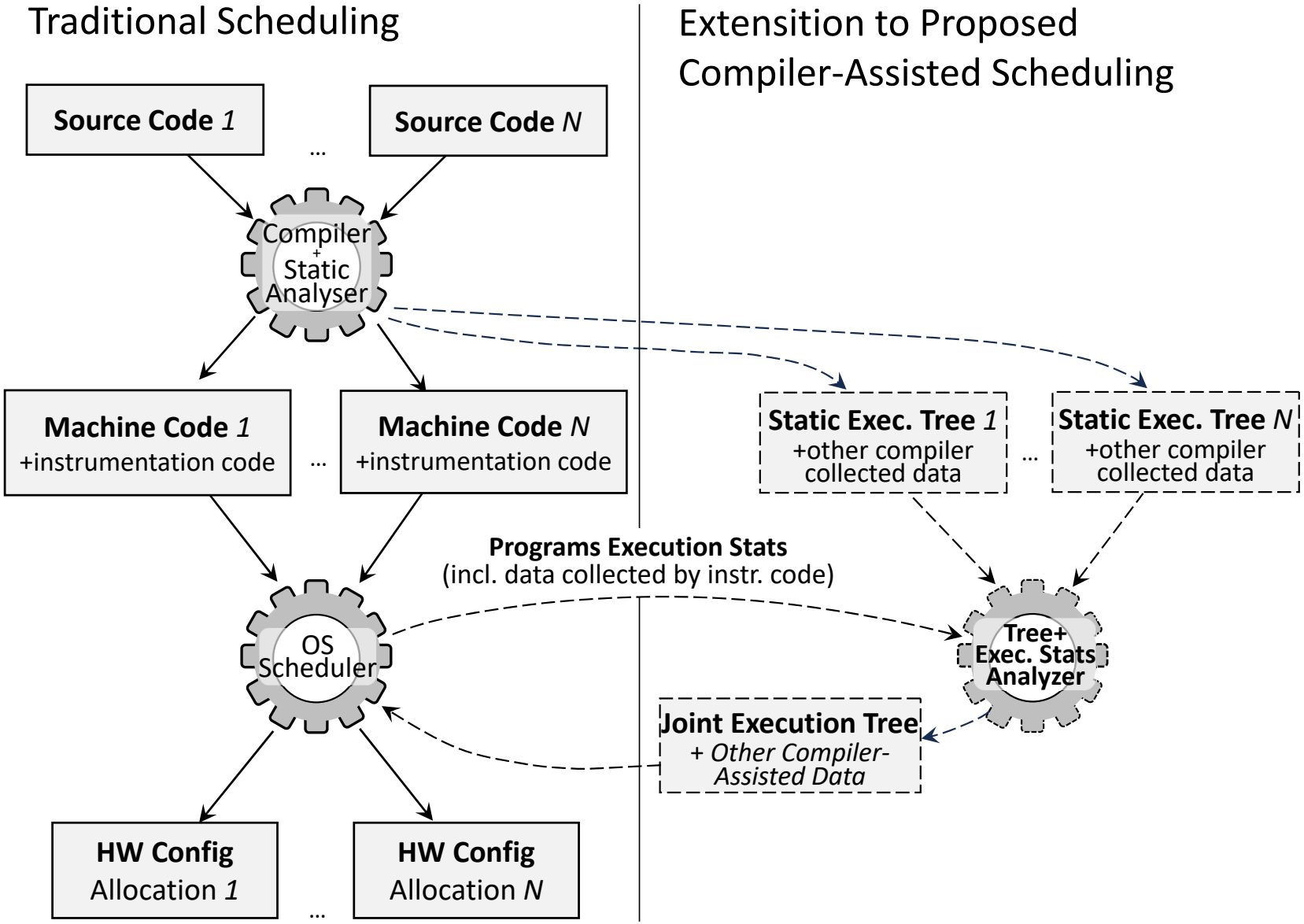
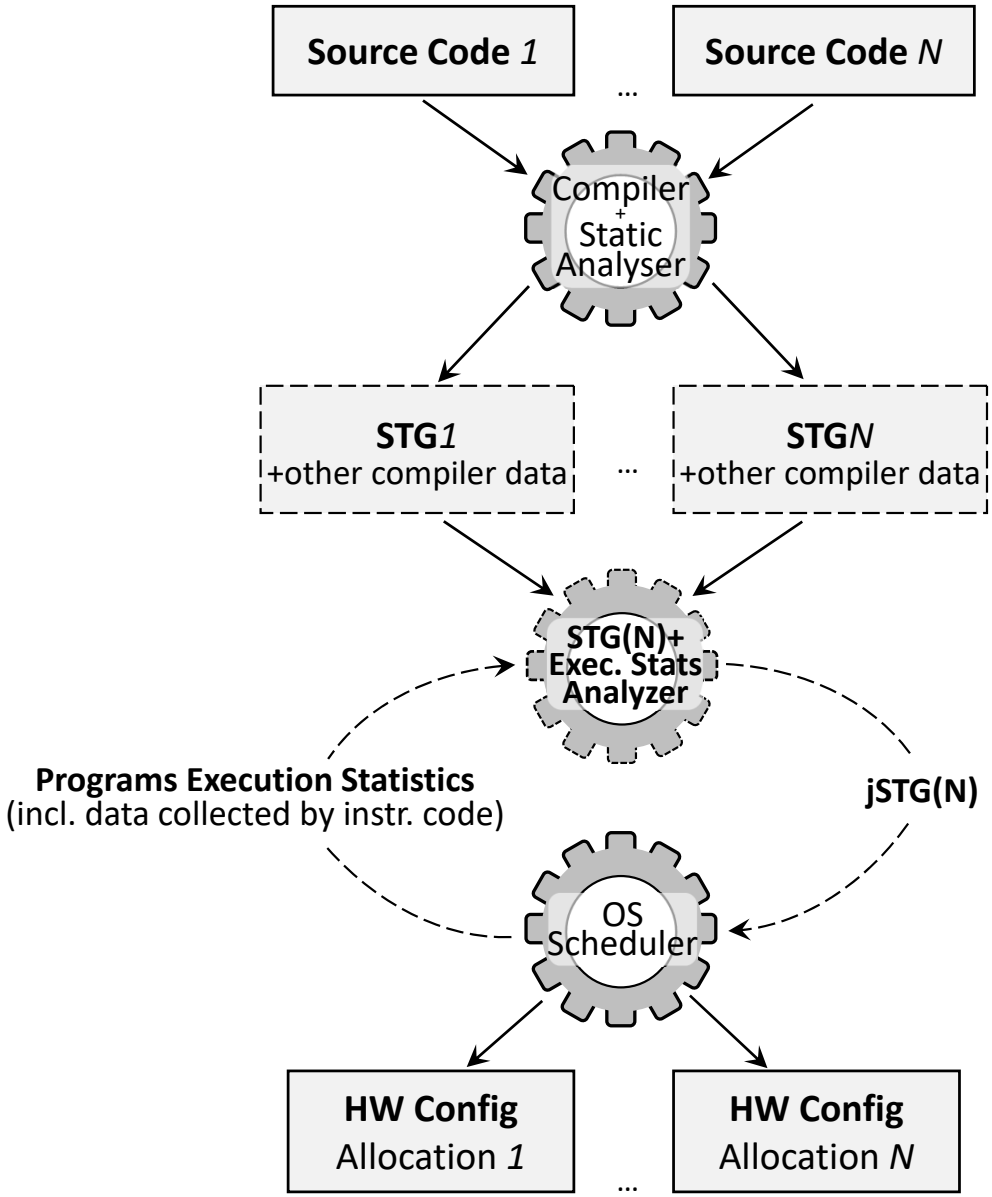
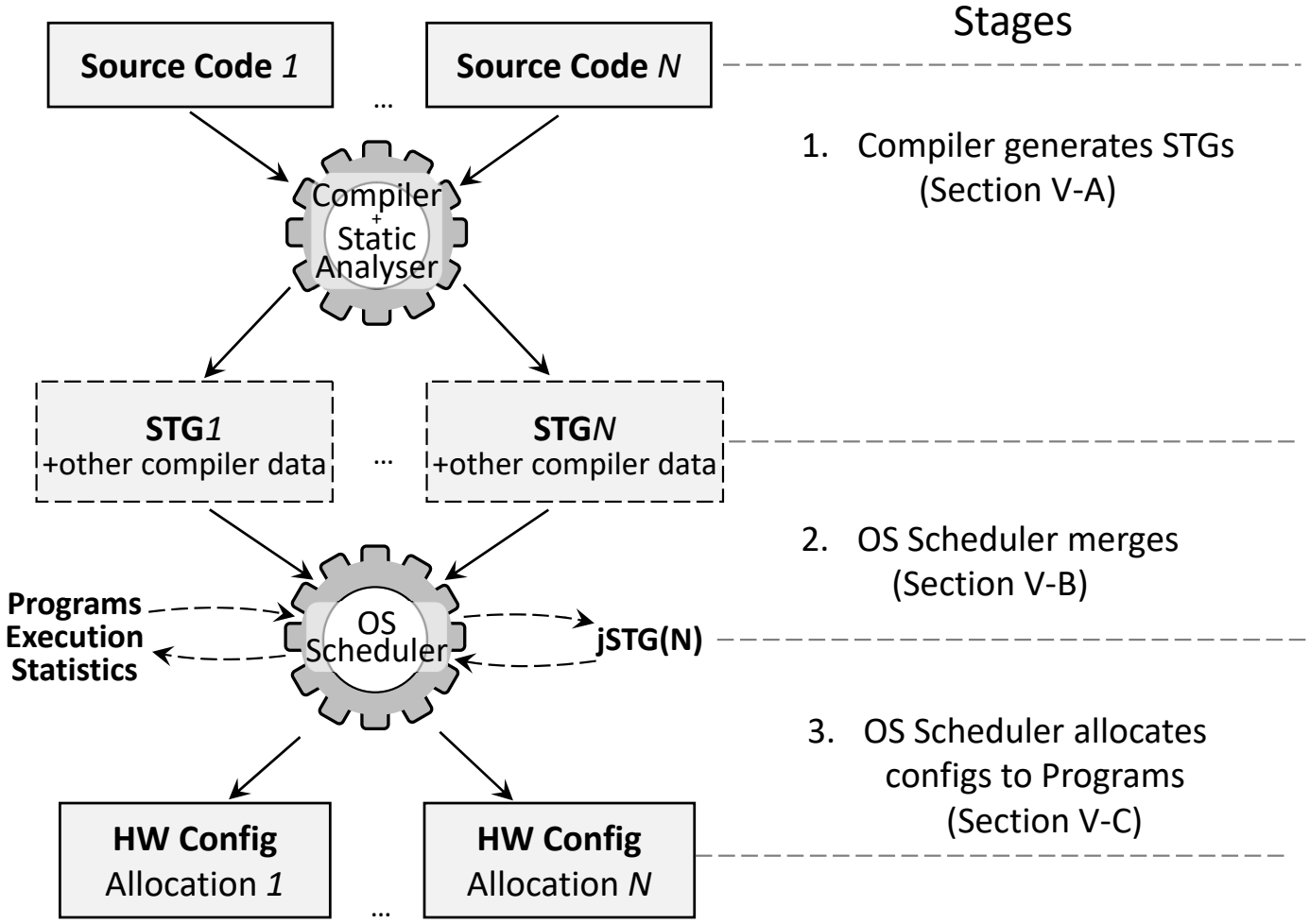

“big” cores cluster
(Cortex-A15, 2.8GHz)
big Core
big Core
big Core
big Core
L2 Cache
“LITTLE” cores cluster
(Cortex-A7, 1.8GHz)
LITTLE Core
LITTLE Core
LITTLE Core
LITTLE Core
L2 Cache
GPU
NPU
Cache Coherent Interconnect

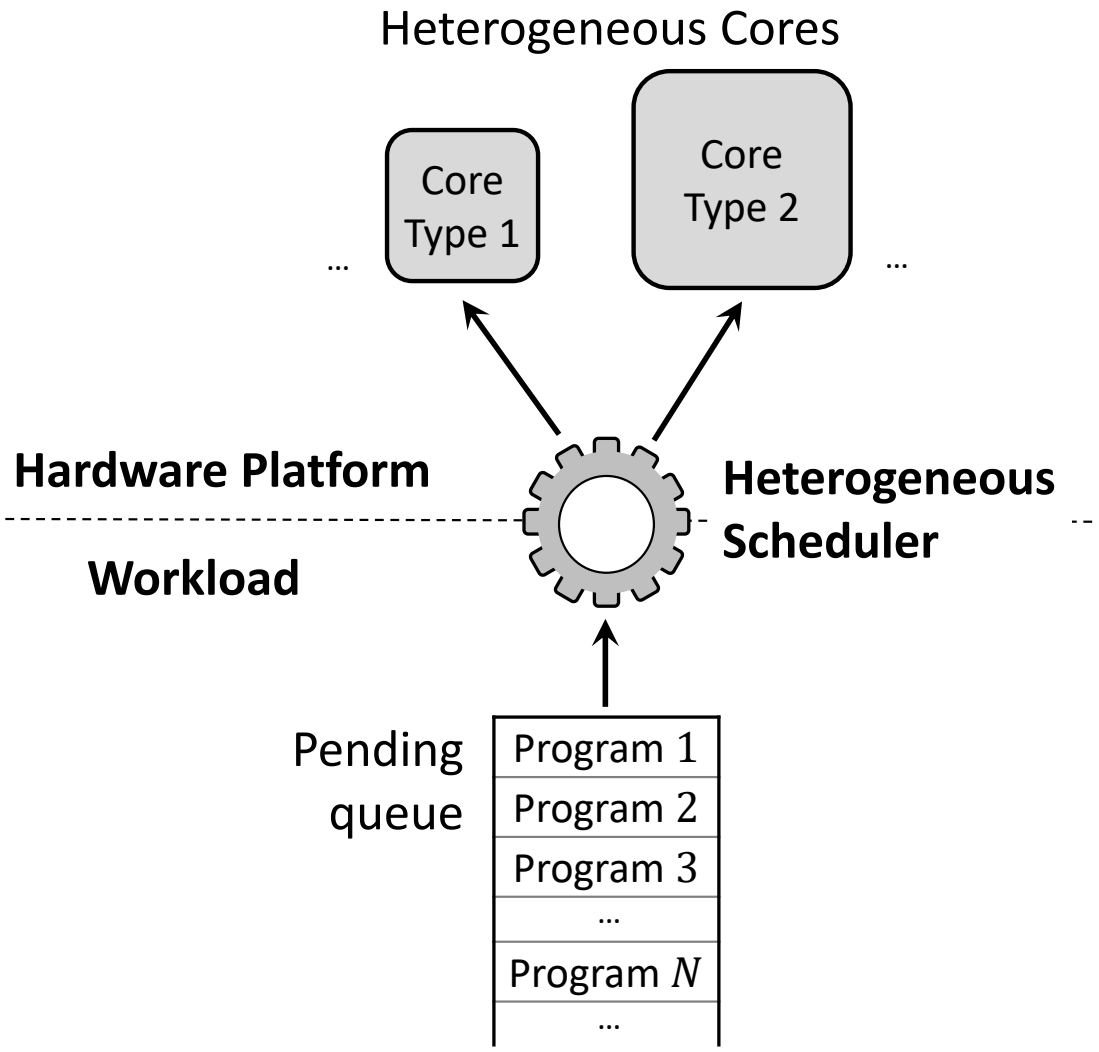

Low-end workload
High-end workload
Cortex-A15
Cortex-A7
Power benefit at the same performance
Lowest and Highest operating points
Power
Performance

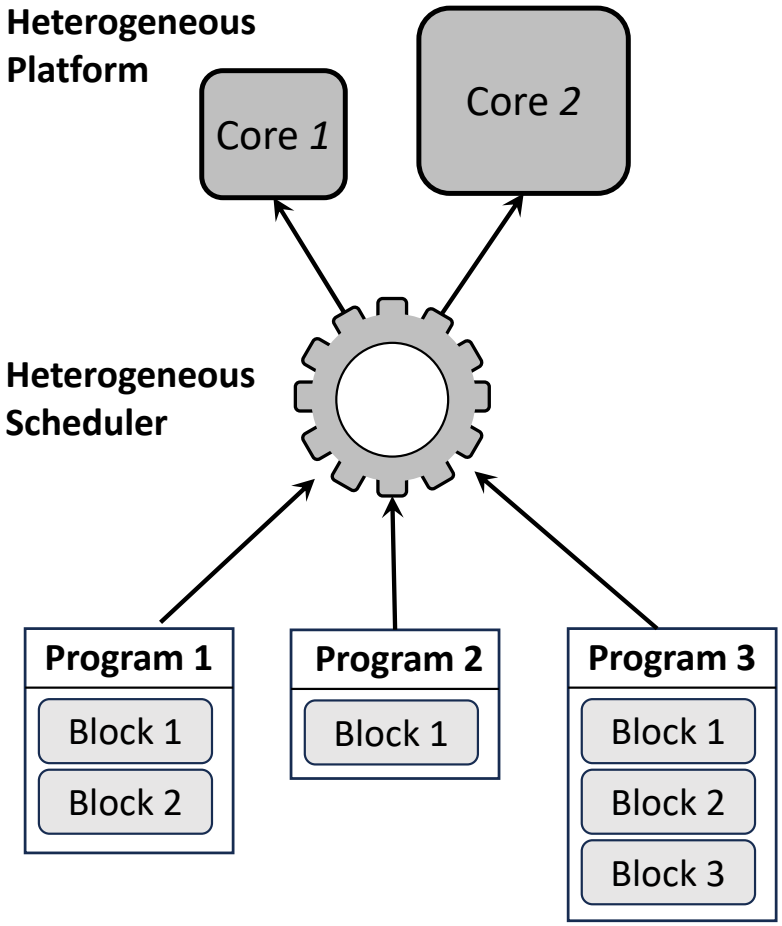

Low to mid range workloads
High-end workloads
Highest Cortex-A15 Operating Point
Overdrive Condition
Power benefit from Cortex-A7
At the same desired performance
Lowest Cortex-A15 Operating Point
Highest Cortex-A7 Operating Point
Lowest Cortex-A7 Operating Point
Cortex-A15
Cortex-A7


Traditional Scheduling
Extension to Proposed Compiler-Assisted Scheduling
Source Code 1
...
Source Code N
Compiler + Static Analyser
Machine Code 1 +instrumentation code
...
Machine Code N +instrumentation code
Static Exec. Tree 1 +other compiler collected data
...
Static Exec. Tree N +other compiler collected data
Programs Execution Stats (incl. data collected by instr. code)
OS Scheduler
Tree+ Exec. Stats Analyzer
Joint Execution Tree + Other Compiler-Assisted Data
HW Config Allocation 1
...
HW Config Allocation N


Source Code 1
...
Source Code N
Compiler
+
Static
Analyser
STG1
+other compiler data
...
STGN
+other compiler data
STG(N)+
Exec. Stats
Analyzer
Programs Execution Statistics
(incl. data collected by instr. code)
jSTG(N)
OS
Scheduler
HW Config
Allocation 1
...
HW Config
Allocation N


Stages
Source Code 1
Source Code N
...
Compiler + Static Analyser
1. Compiler generates STGs (Section V-A)
STG1 +other compiler data
...
STGN +other compiler data
2. OS Scheduler merges (Section V-B)
Programs Execution Statistics
OS Scheduler
jSTG(N)
3. OS Scheduler allocates configs to Programs (Section V-C)
HW Config Allocation 1
...
HW Config Allocation N


Heterogeneous Cores
Core Type 1
Core Type 2
...
...
Hardware Platform
Workload
Heterogeneous Scheduler
Pending queue
Program 1
Program 2
Program 3
...
Program N
...

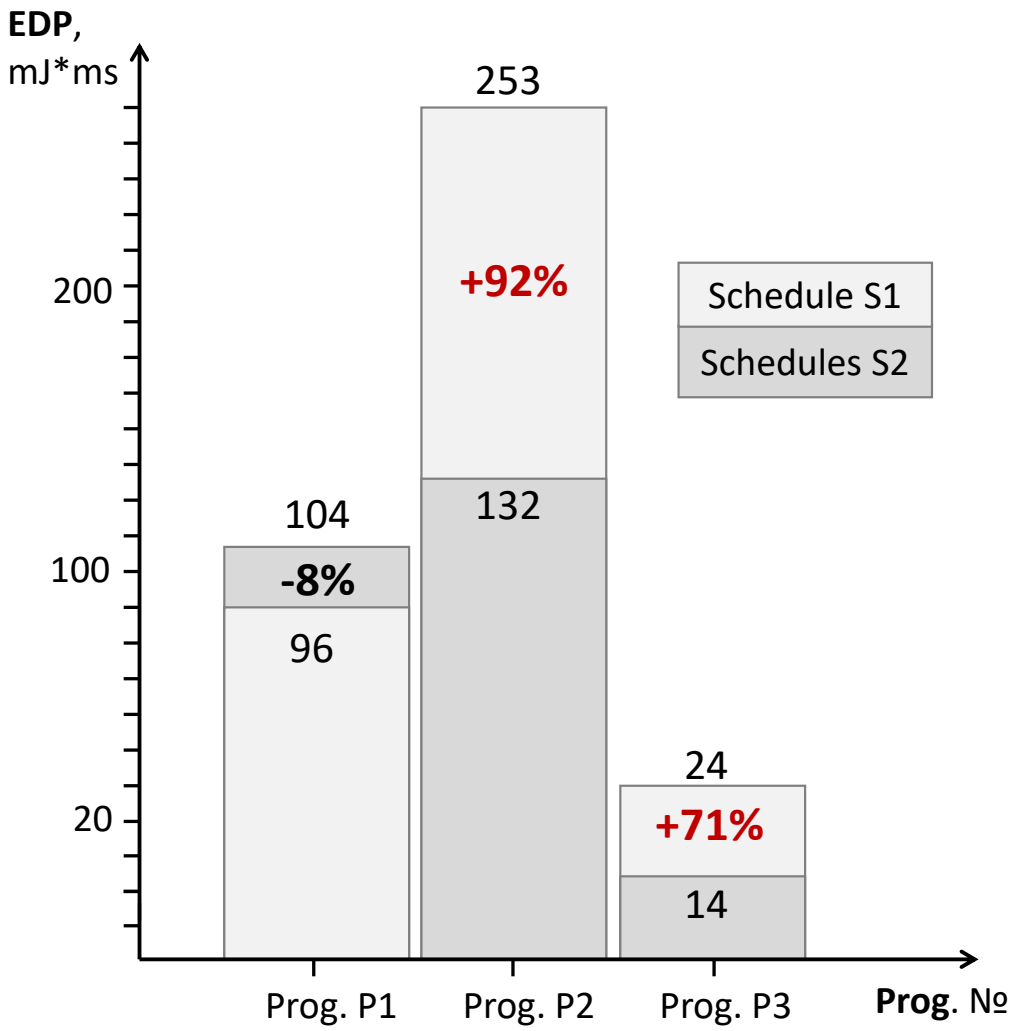


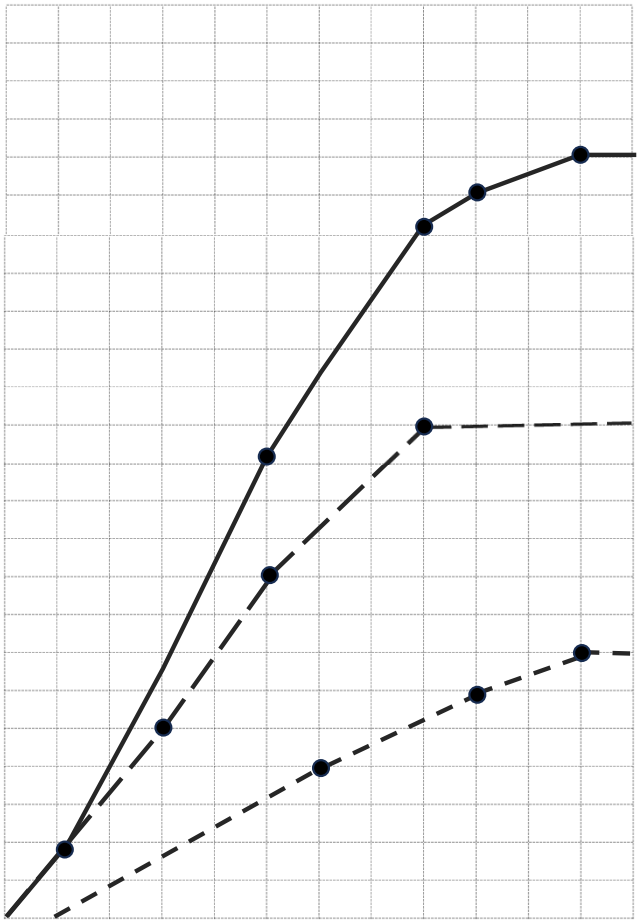
High-level scheduling of 3 programs of several blocks to be executed over a heterogeneous platform of 2 cores.

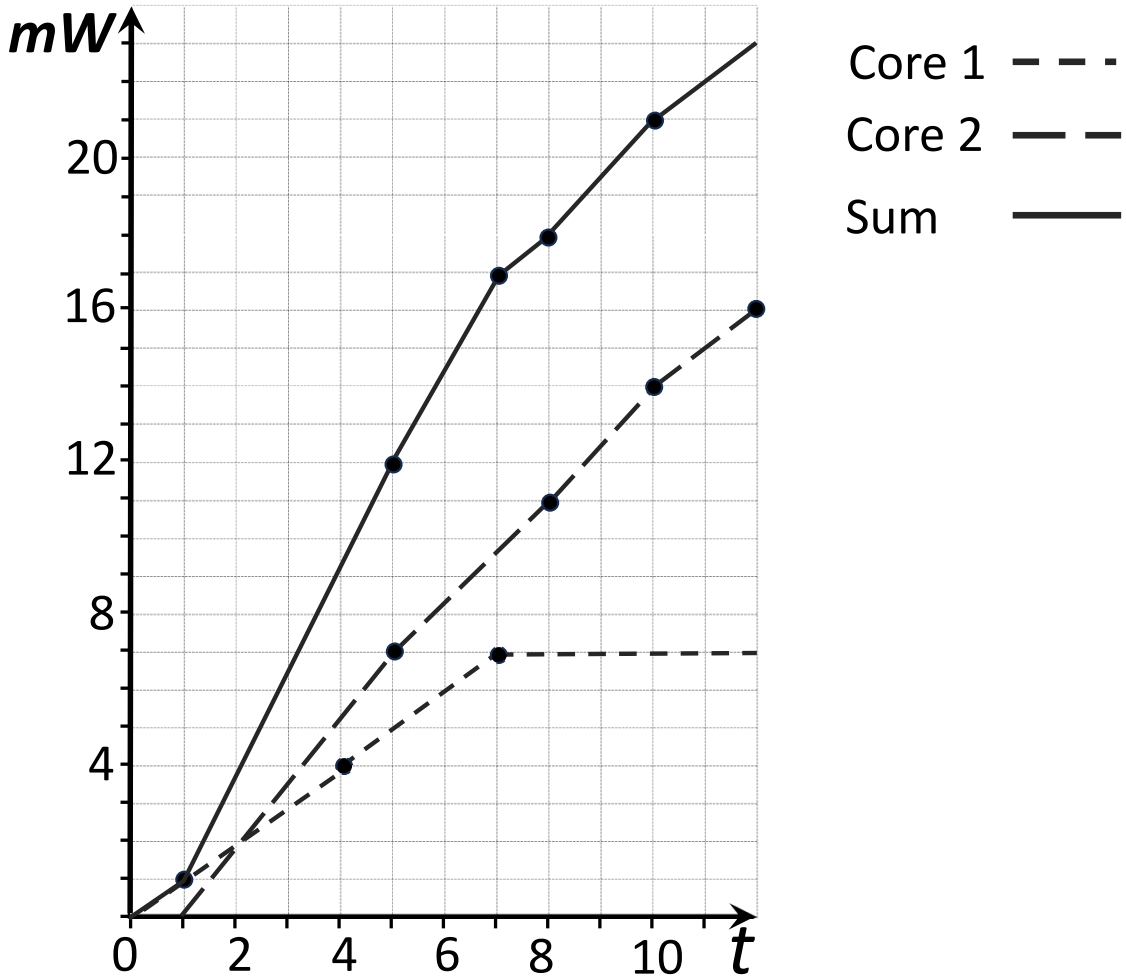
Sample execution requirements of program blocks of 3 programs for 2 heterogeneous cores.

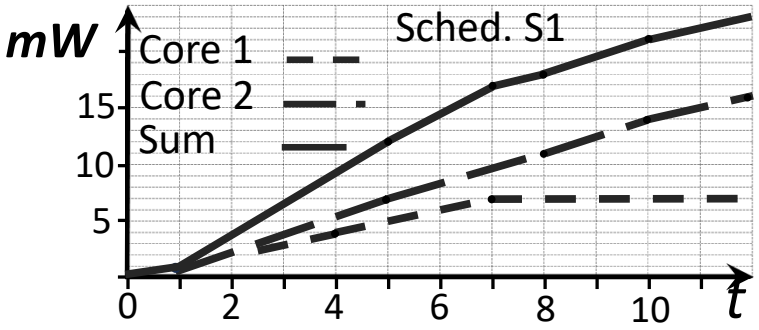
Programs /Blocks		Execution time, ms		Energy use, mJ	
		Core 1 (slow)	Core 2 (fast)	Core 1	Core 2
P1	B1	4	3	8	9
	B2	5	3	3	4
P2	B1	5	4	6	10
	B2	3	2	3	7
	B3	3	2	3	6
P3	B1	3	2	6	7

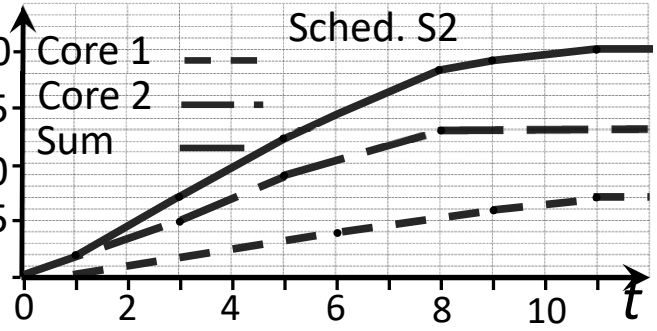
Programs	Execution time, ms		Energy use, mJ	
	Core (no DVFS)	Core (slower)	Core (no DVFS)	Core (slower)
P1	5	7	9	8
P2	4	5	10	6
P3	2	3	7	6

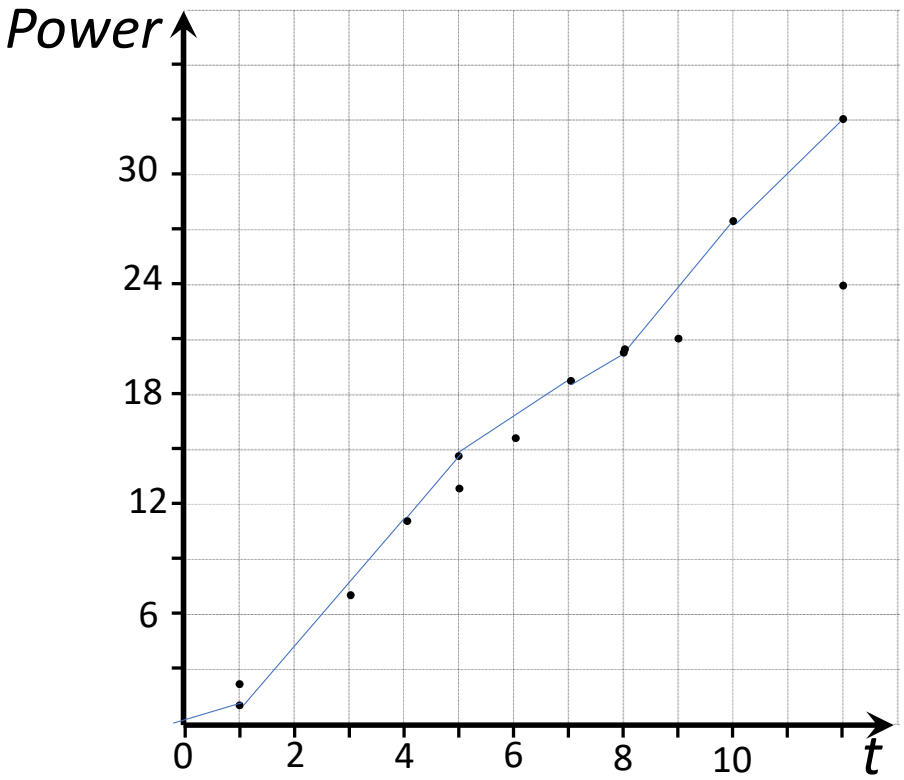


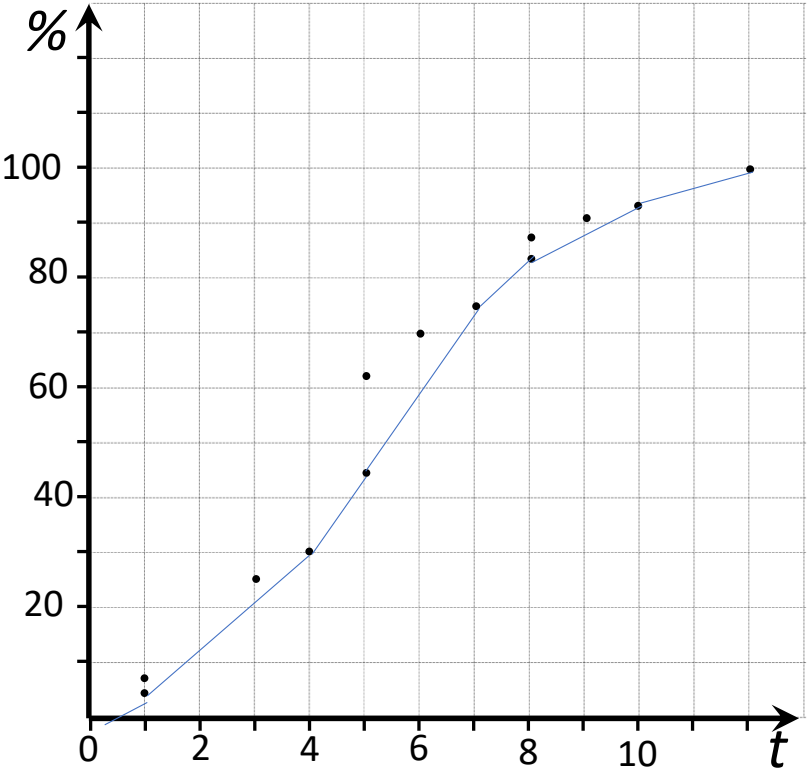


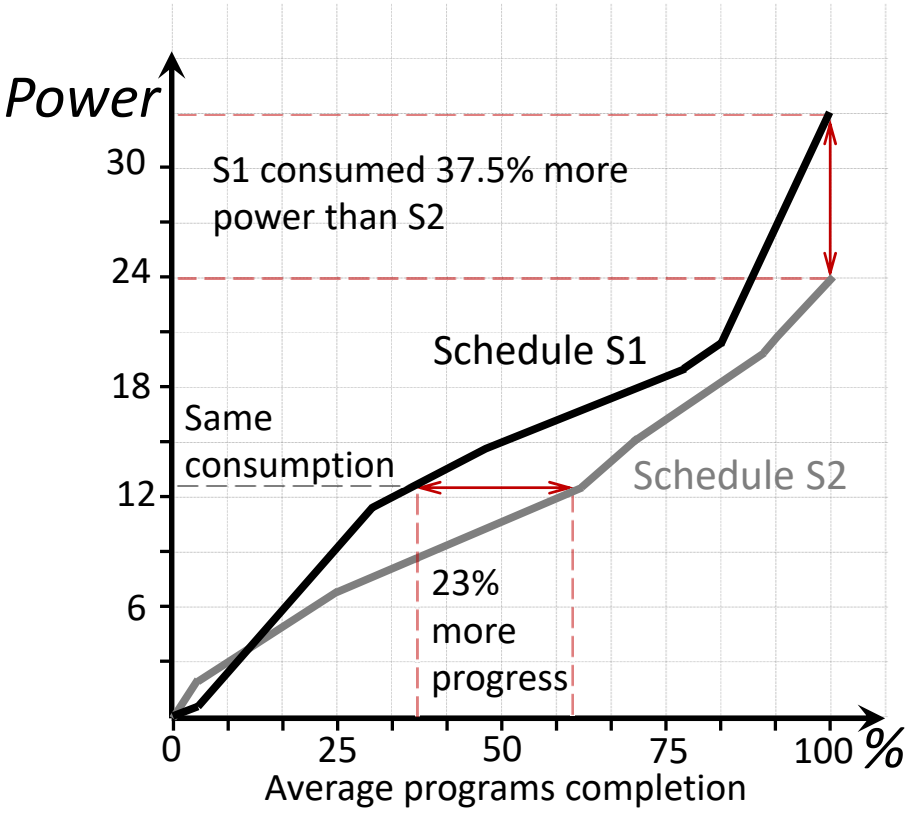

mW
20
16
12
8
4
0
2
4
6
8
10
t
Core 1
Core 2
Sum

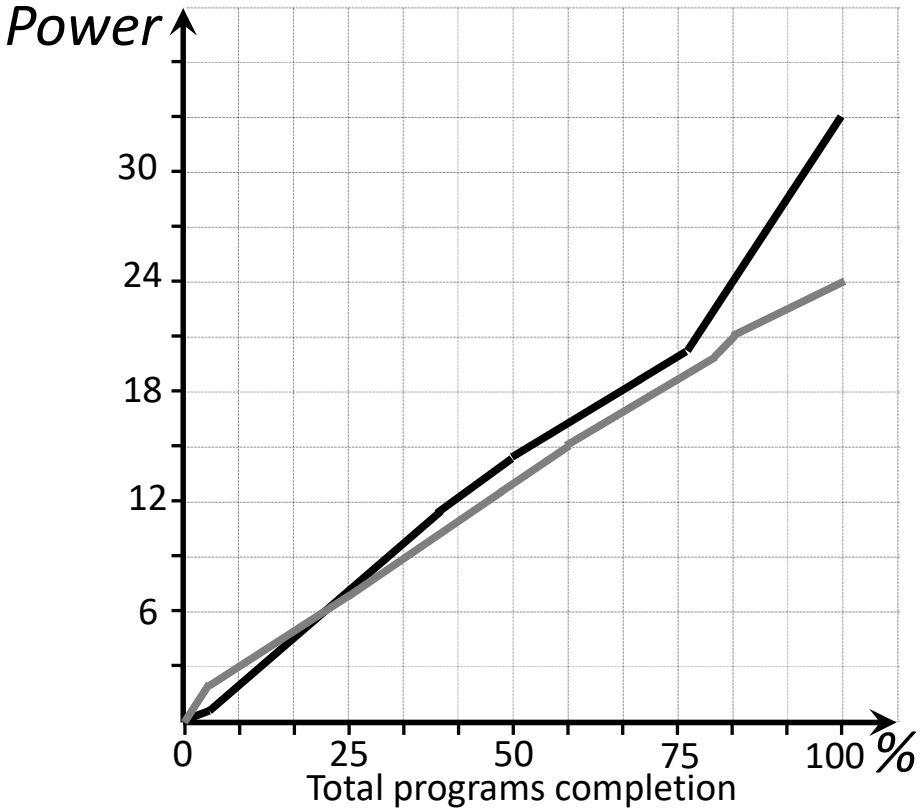

mW
Sched. S1
Core 1
Core 2
Sum
15
10
5
0
2
4
6
8
10
t


Sched. S2
Core 1
Core 2
Sum
20
15
10
5
0
2
4
6
8
10
t

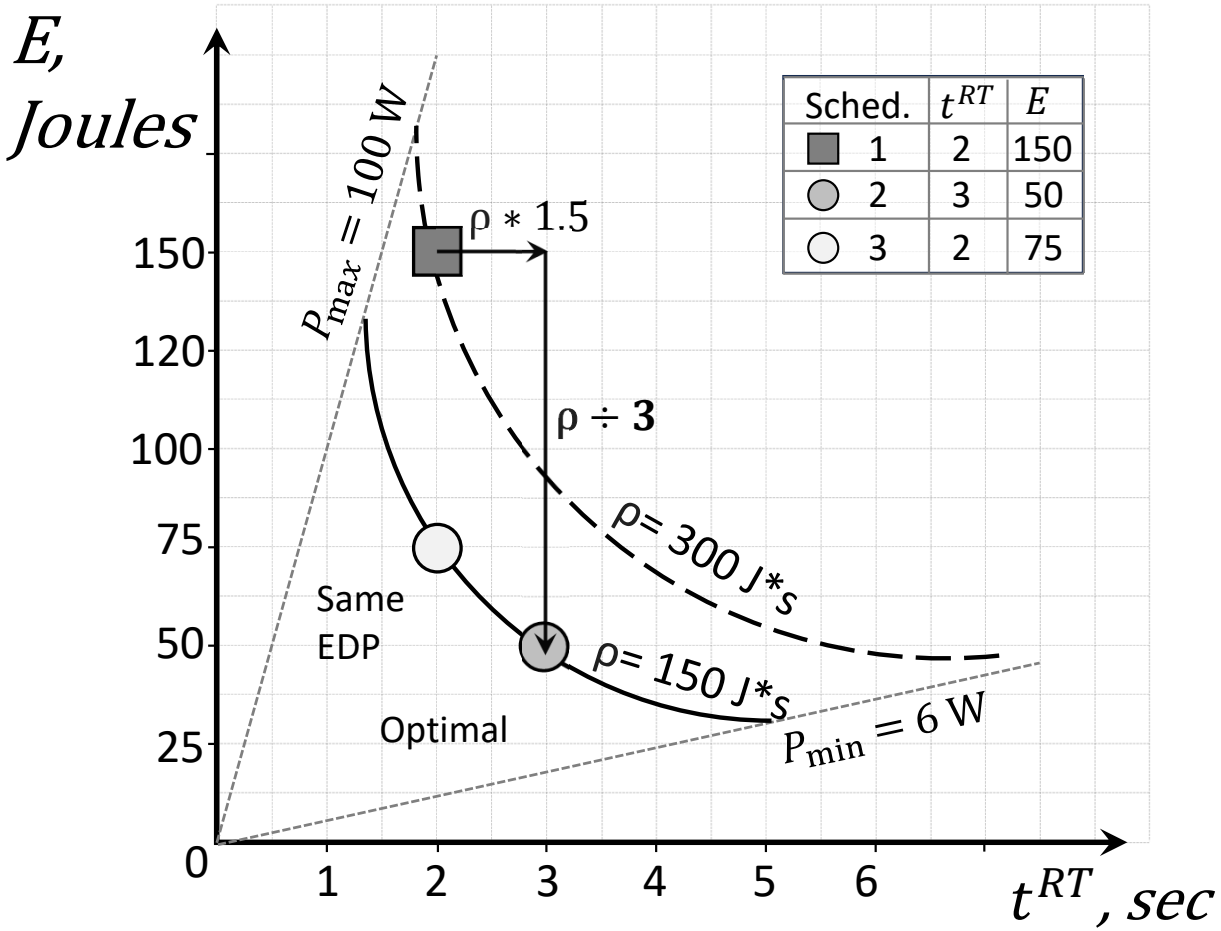

Power
30
24
18
12
6
0
2
4
6
8
10
t


%
100
80
60
40
20
0
2
4
6
8
10
t

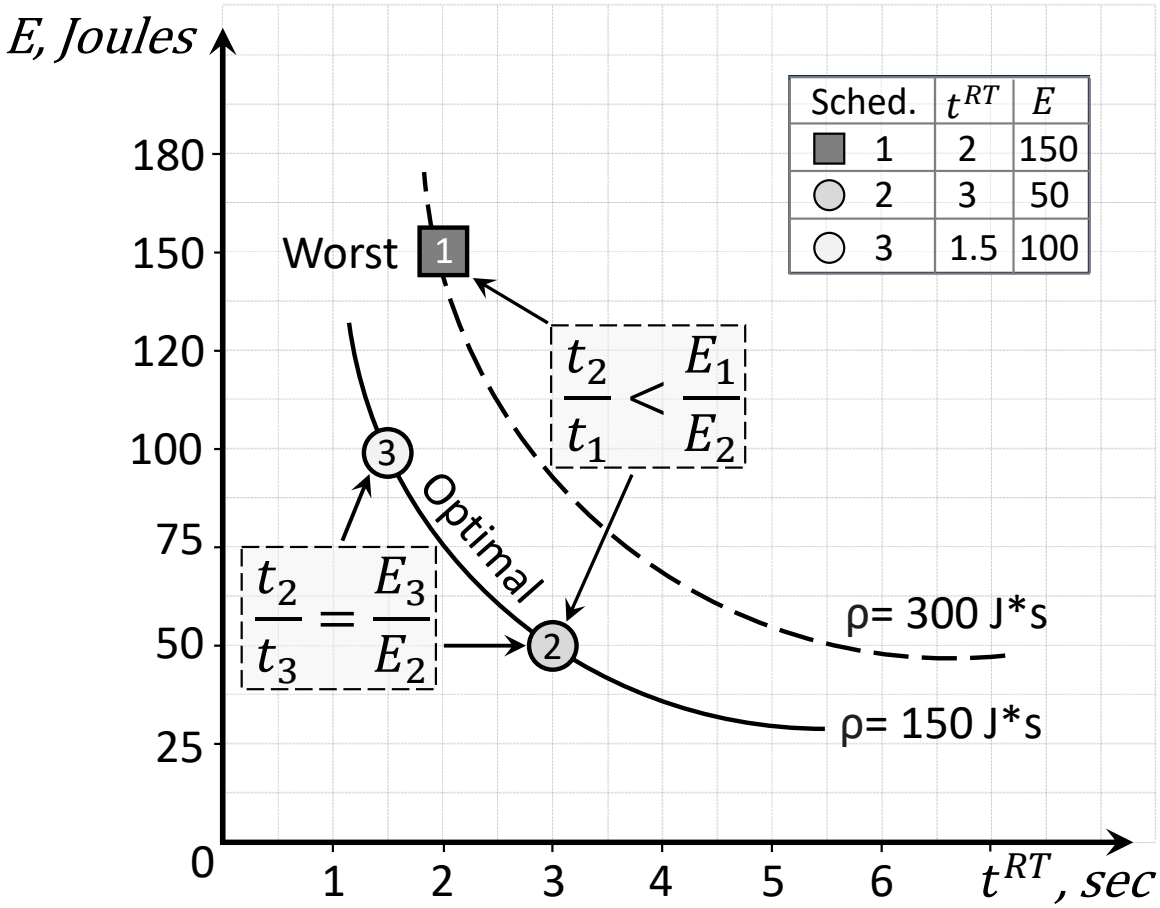

Power
30
24
18
12
6
S1 consumed 37.5% more
power than S2
Schedule S1
Same
consumption
Schedule S2
23%
more
progress
0
25
50
75
100
%
Average programs completion


Power
30
24
18
12
6
0
25
50
75
100
%
Total programs completion

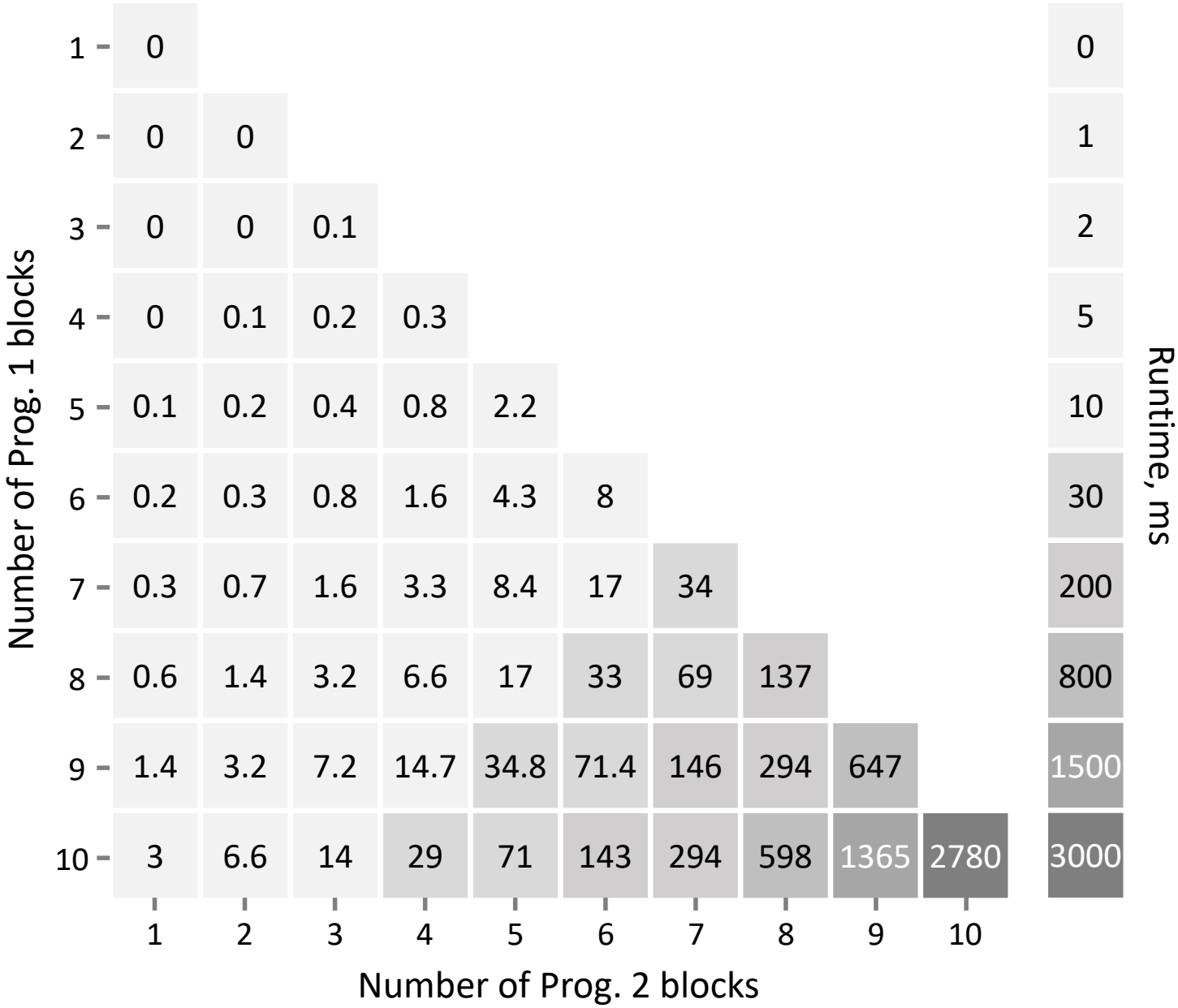
Energy efficiency

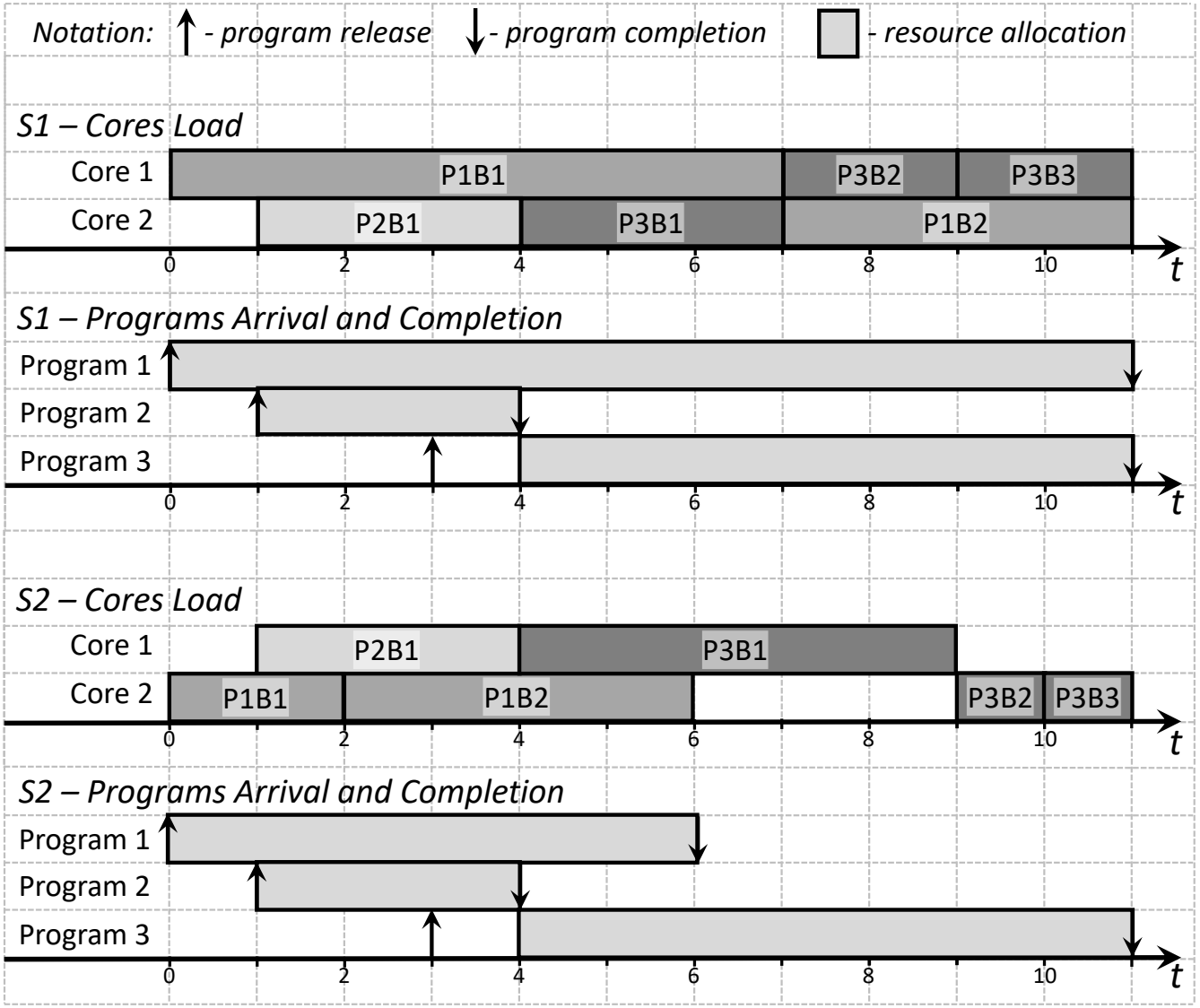


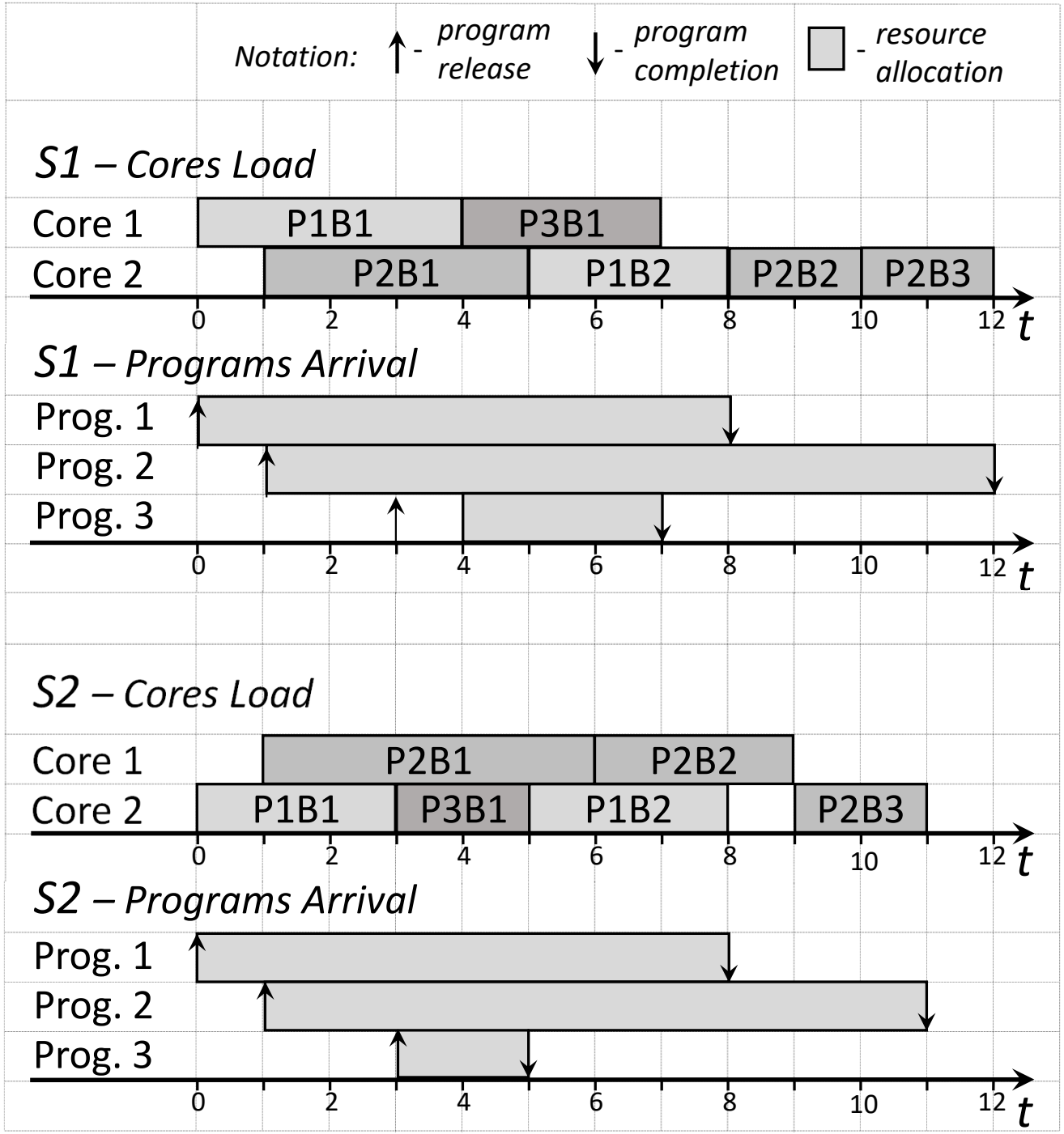
Energy efficiency

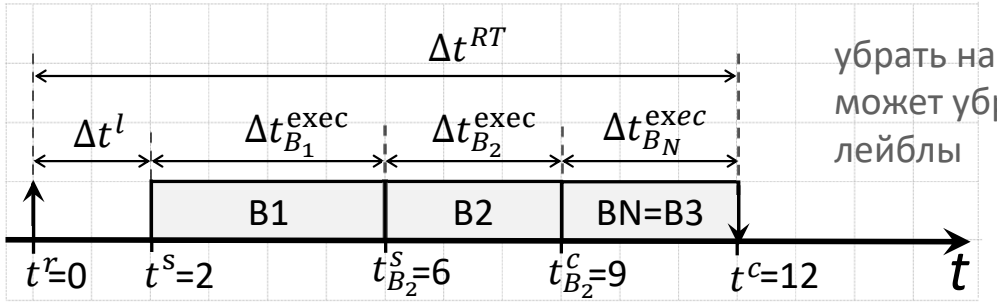


Experiments

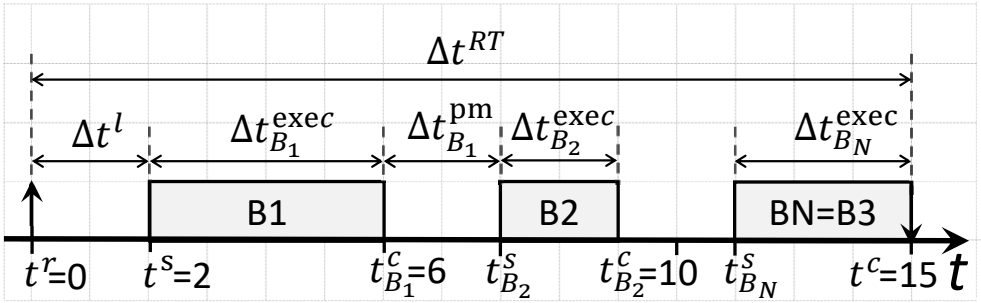


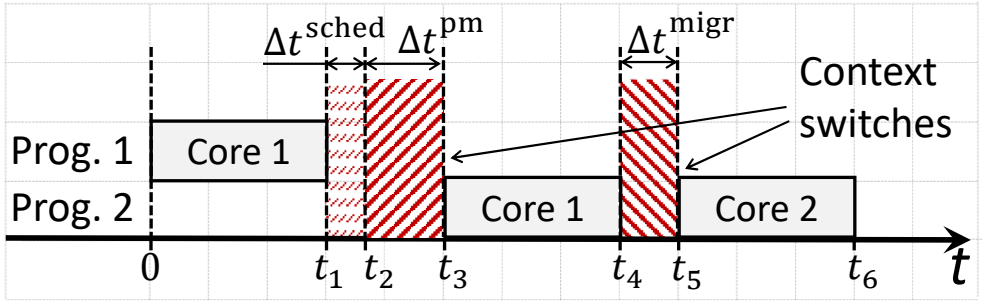

Notation:
- program release
- program completion
- resource allocation
S1 – Cores Load
Core 1
Core 2
P1B1
P3B2
P3B3
P2B1
P3B1
P1B2
0
2
4
6
8
10
t
S1 – Programs Arrival and Completion
Program 1
Program 2
Program 3
0
2
4
6
8
10
t
S2 – Cores Load
Core 1
Core 2
P2B1
P3B1
P1B1
P1B2
P3B2
P3B3
0
2
4
6
8
10
t
S2 – Programs Arrival and Completion
Program 1
Program 2
Program 3
0
2
4
6
8
10
t

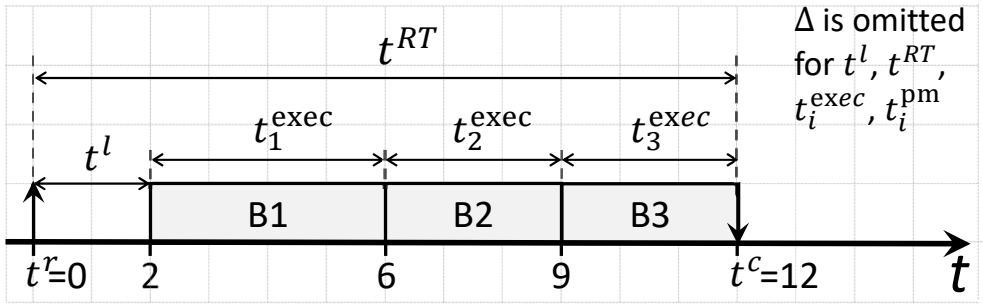

Notation:
- program release
- program completion
- resource allocation
S1 – Cores Load
Core 1
Core 2
P1B1
P3B1
P2B1
P1B2
P2B2
P2B3
0
2
4
6
8
10
12
t
S1 – Programs Arrival
Prog. 1
Prog. 2
Prog. 3
0
2
4
6
8
10
12
t
S2 – Cores Load
Core 1
Core 2
P2B1
P2B2
P1B1
P3B1
P1B2
P2B3
0
2
4
6
8
10
12
t
S2 – Programs Arrival
Prog. 1
Prog. 2
Prog. 3
0
2
4
6
8
10
12
t

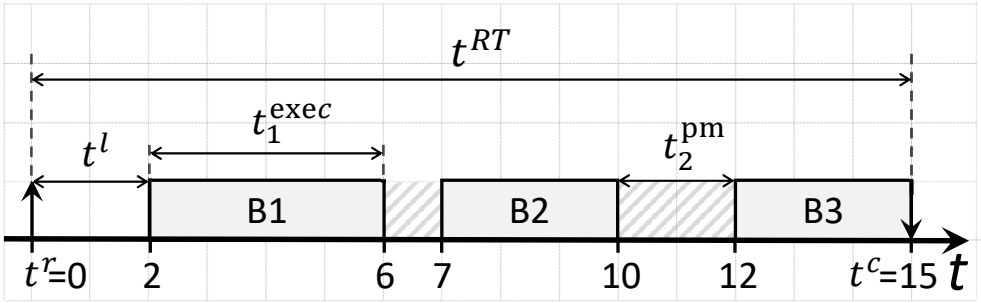


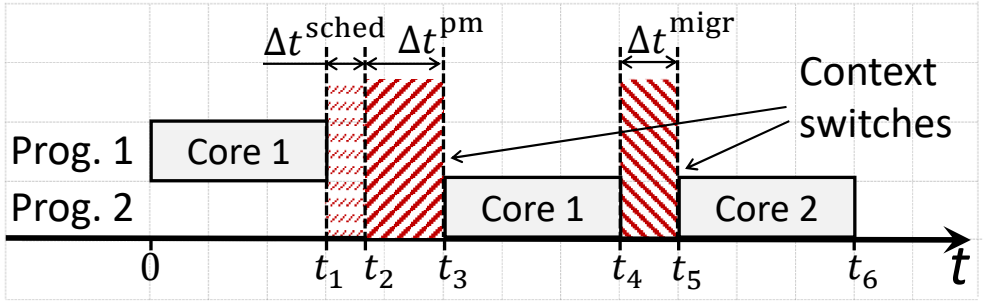
убрать нагромождение, может убрать некоторые лейблы






t^{RT}
t^{l}
t_1^{exec}
t_2^{exec}
t_3^{exec}
B1
B2
B3
t^{r}=0
2
6
9
t^{c}=12
t
Δ is omitted for t^{l}, t^{RT}, t_i^{exec}, t_i^{pm}

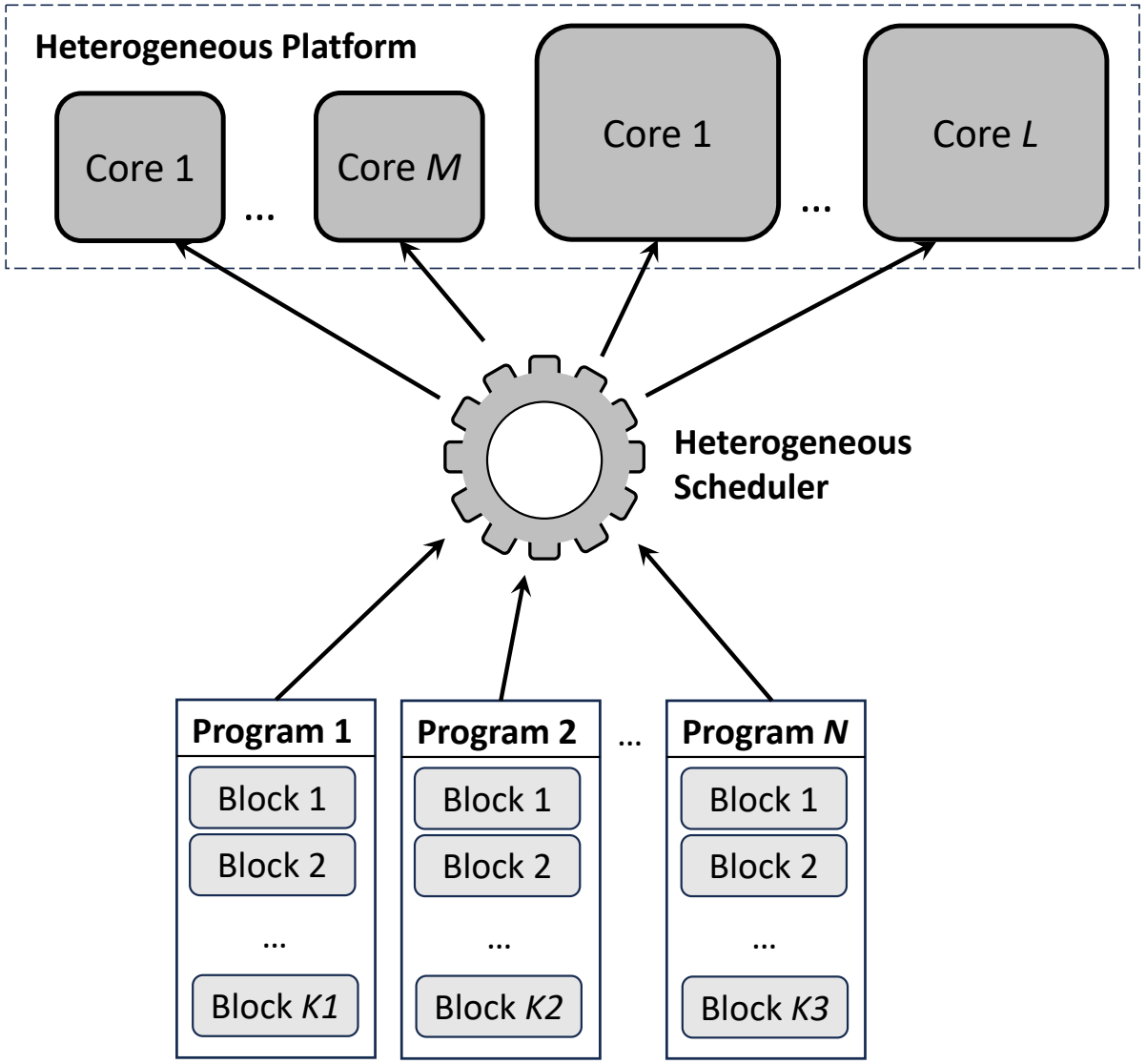

t^{RT}
t^{l}
t_1^{exec}
t_2^{pm}
B1
B2
B3
t^{r}=0
2
6
7
10
12
t^{c}=15
t


Δt^{sched}
Δt^{pm}
Δt^{migr}
Context switches
Prog. 1
Prog. 2
Core 1
Core 1
Core 2
0
t_1
t_2
t_3
t_4
t_5
t_6
t

Program	Schedule	Scheduling Objective to Minimize		
		Latency	Execution Time	Response Time
Prog. 1	S1	0	7	8
	S2	0	6	8
Prog. 2	S1	0	8	11
	S2	0	10	10
Prog. 3	S1	1	3	4
	S2	0	2	2
Average	S1	**0.33**	**6**	**7.67**
	S2	**0**	**6**	**6.67**

Schedule	Program	Latency	Execution Time	Response Time	Average Execution Time	Average Response Time
S1	P1	0	7	8	**6**	**7.67**
	P2	0	8	11		
	P3	1	3	4		
S2	P1	0	6	8	**6.33**	**7**
	P2	0	11	11		
	P3	0	2	2		

Schedule	Program	Latency	Execution Time	Response Time	Average Execution Time	Average Response Time
S2	P1	0	6	8	**6.33**	**7**
	P2	0	11	11		
	P3	0	2	2		


Heterogeneous Platform
Core 1
...
Core M
Core 1
...
Core L
Heterogeneous
Scheduler
Program 1
Block 1
Block 2
...
Block K1
Program 2
Block 1
Block 2
...
Block K2
...
Program N
Block 1
Block 2
...
Block K3

Algorithm	Block №	Config 1	Config 2
Winograd	1	71	75
	2	5	5
	3	1813	2240
	4	1717	1985
	5	1773	1736
	6	1773	1802
	7	1781	917
	8	1616	755
	9	1775	757
	10	7	3
LZ4	1	40	35
	2	26	20
	3	50	46
	4	74	52
	5	31	26
	6	19736	9808
	7	965	311
	8	1	1
	9	15	5

Comparison point	Winograd	LZ4
Purpose	Matrix multiplication algorithm (for large matrices)	High-speed lossless data compression
Number of blocks	10	9
Improvement from switching to Config 2	x1.2	X2
Memory usage	High (depends on matrix size)	Low
Cache utilization	Inefficient (matrices are larger than cache)	Efficiently utilizes caches (64KB window)
Main block type	Recursive calls	Loops
Practical usage	Rarely used due to high constant factors	Extensively used in production
Main performance issues	Disk swapping, page faults	Slow memory access