ASPLOS

Enabling Efficient Mobile Tracing with BTrace

Jiawei Wang[1], Nian Liu[1], Arnau Casadevall-Saiz[1,2], Yutao Liu[1],
Diogo Behrens[1], Ming Fu[1], Ning Jia[1], Hermann Härtig[3], and Haibo Chen[1,4]

[1] HUAWEI

[2] VUB

[3]

[4] SHANGHAI JIAO TONG UNIVERSITY 1896

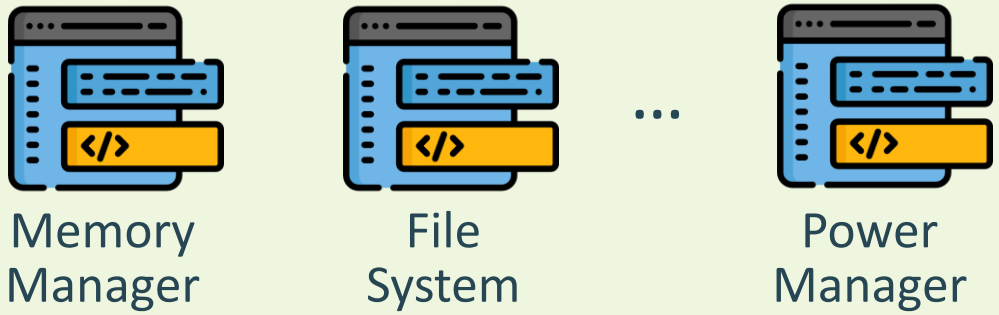
Tracing

Tracing

OS Components

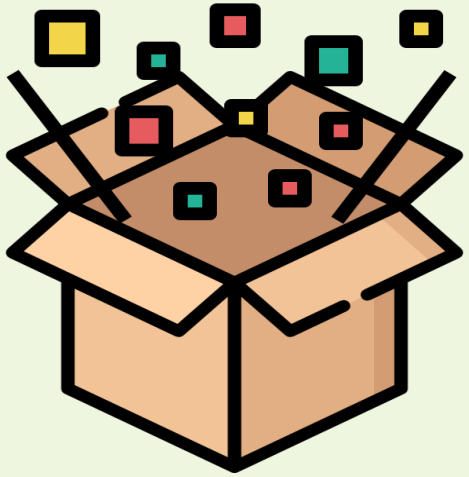
Tracing

OS Components



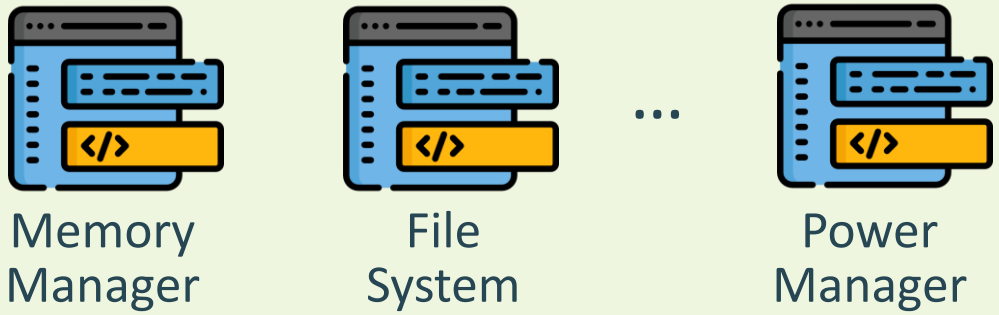
trace_write

Tracers

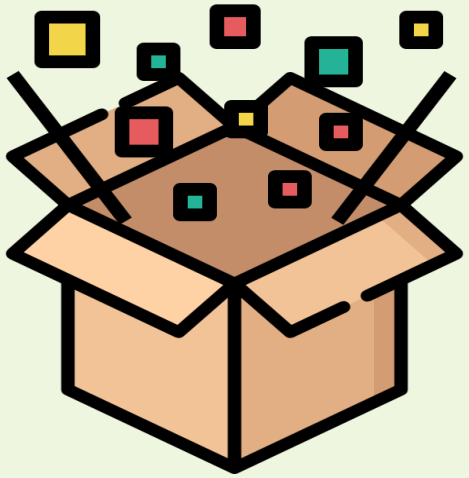


Tracing

OS Components



trace_write

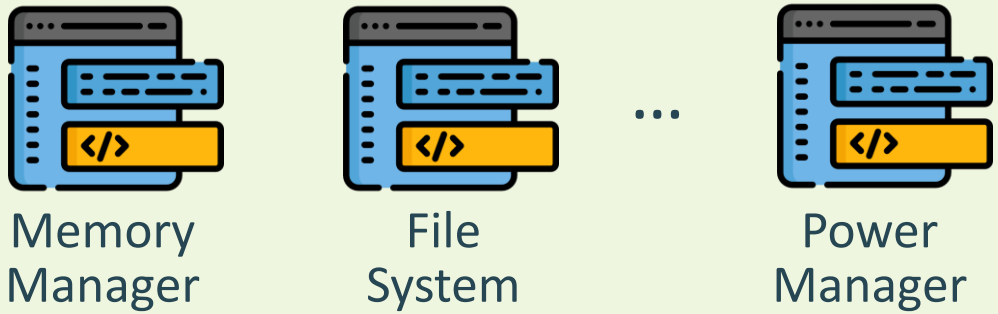


Tracers

Overwrite
Old traces

Tracing

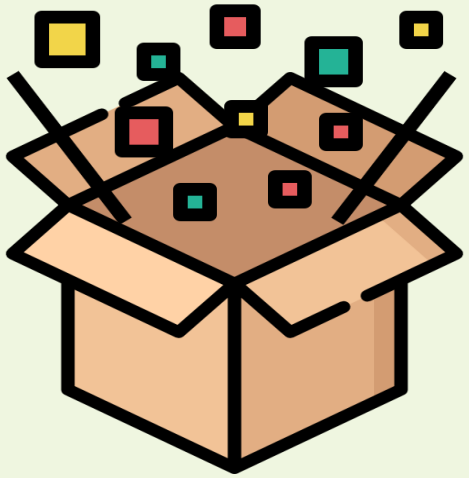
OS Components



trace_write

Tracers

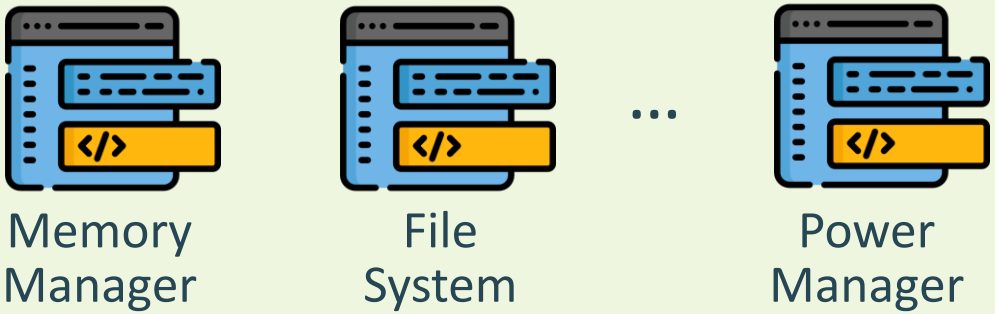
Overwrite
Old traces



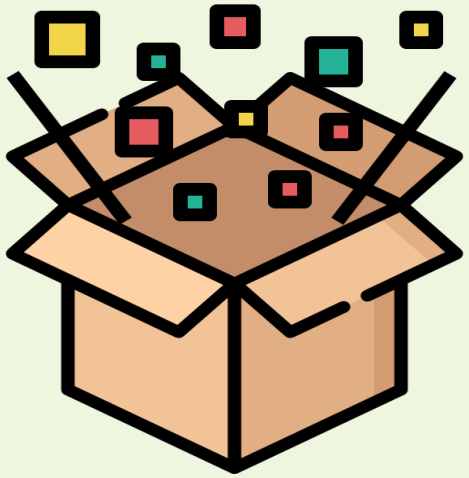
trace_read

Tracing

OS Components



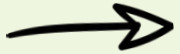
trace_write



Tracers

Overwrite
Old traces

trace_read

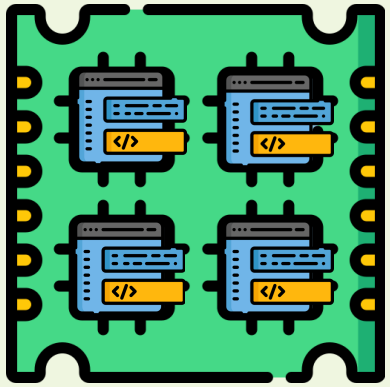


Analyzing issues

Performance Bottlenecks
Resource Utilization
System Stability
...

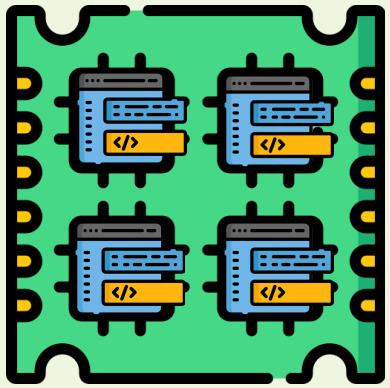
Existing Tracers

Multi-core Device

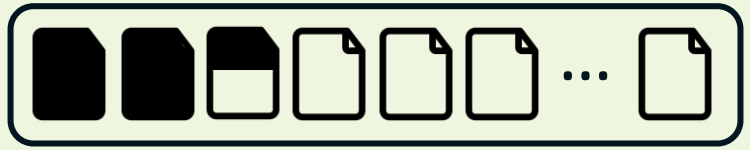


Existing Tracers

Multi-core Device

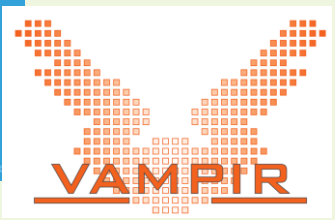


High Contention



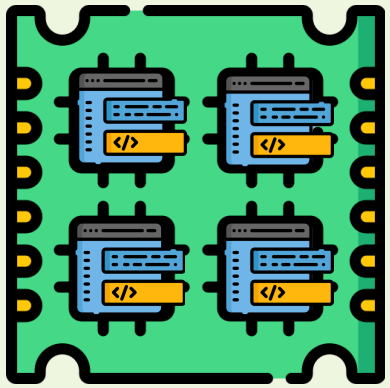
Global Buffer

Existing Tracers

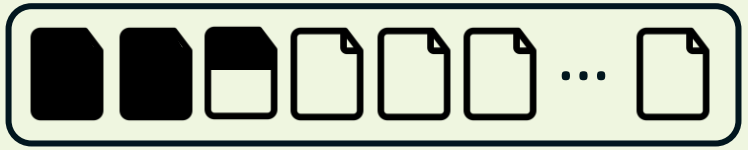


...

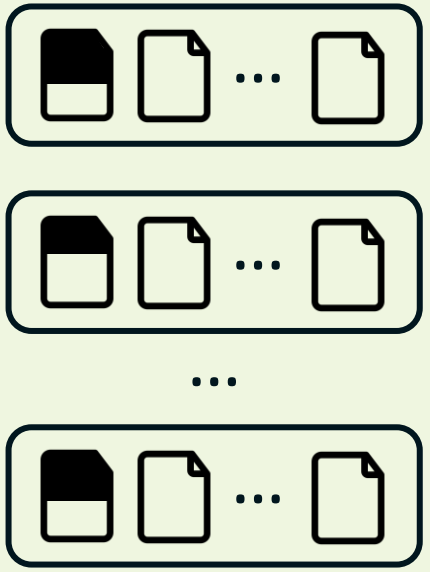
Multi-core Device



High Contention



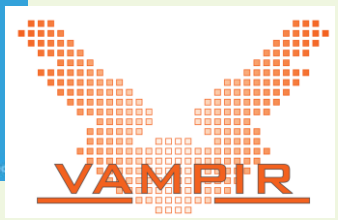
Global Buffer



Per-core Buffer

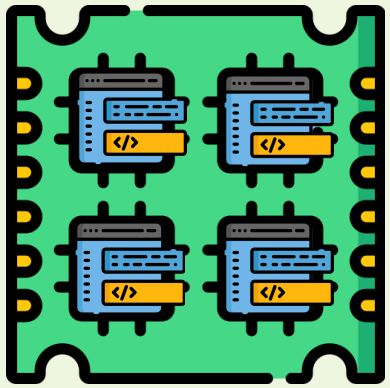
Per-thread Buffer

Existing Tracers

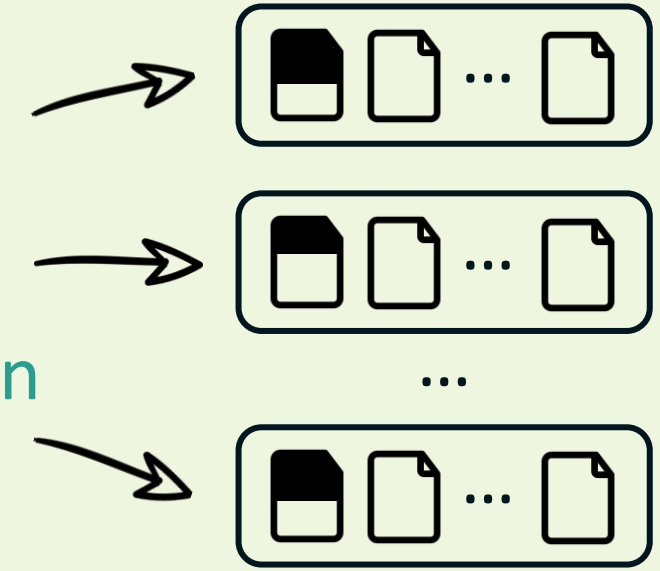


...

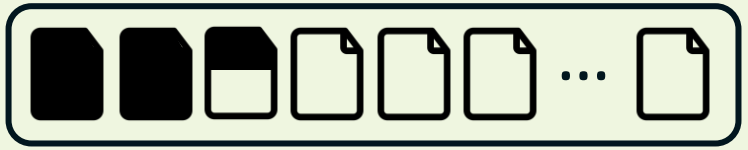
Multi-core Device



Lower Contention



High Contention

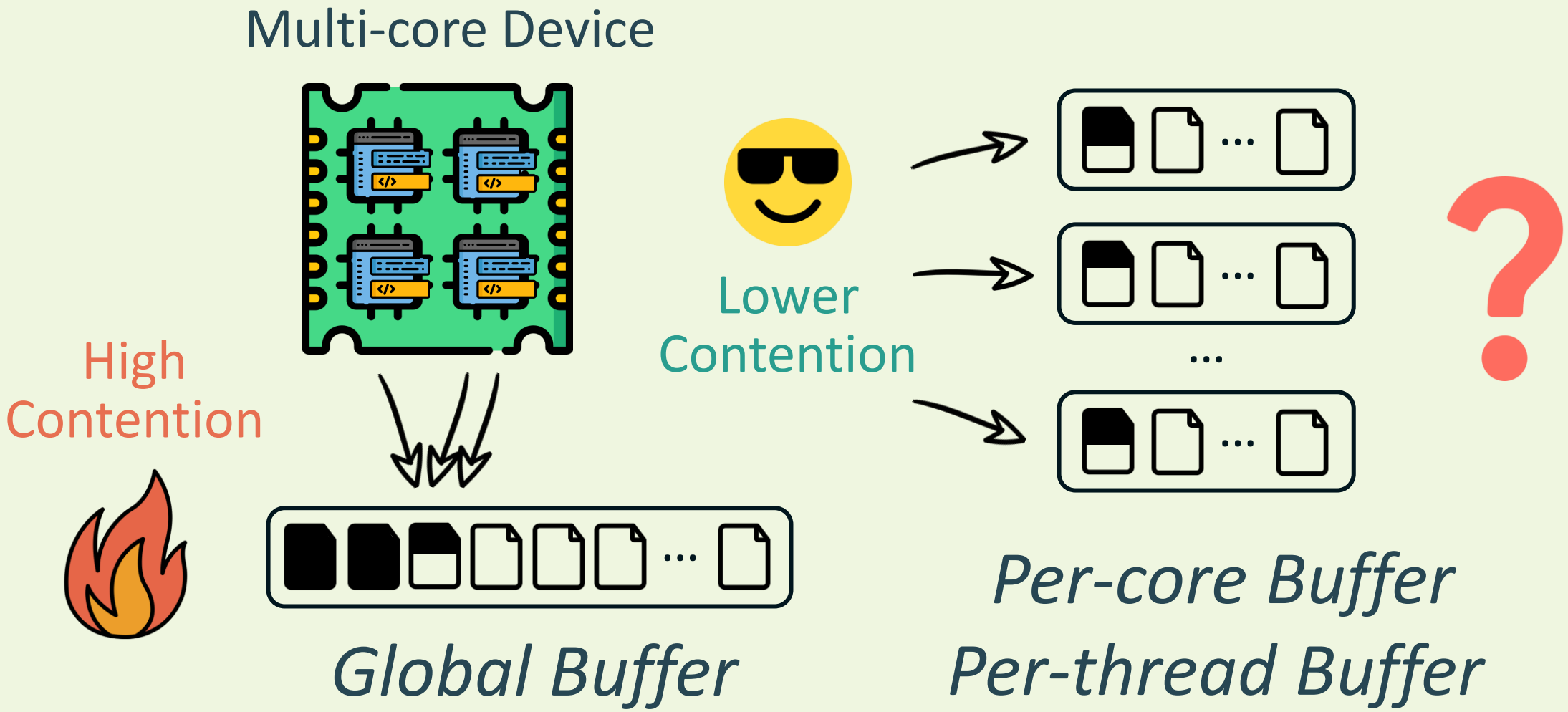


Global Buffer

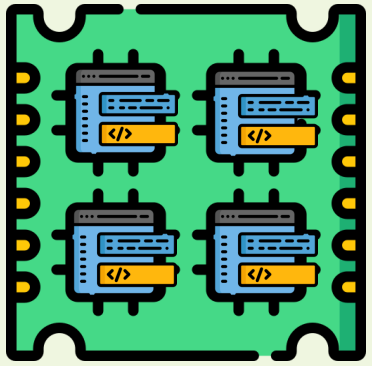
Per-core Buffer

Per-thread Buffer

Existing Tracers on Mobiles



Existing Tracers on Mobiles



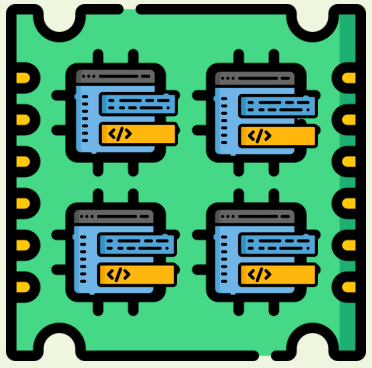
Shopping App.

Existing Tracers on Mobiles

Memory Manager

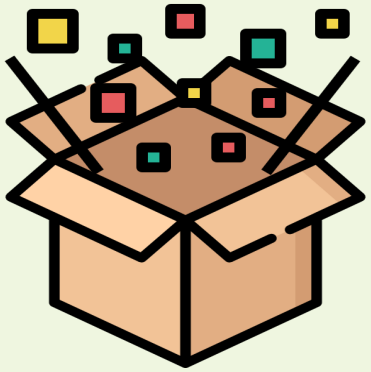
File System

Power Manager



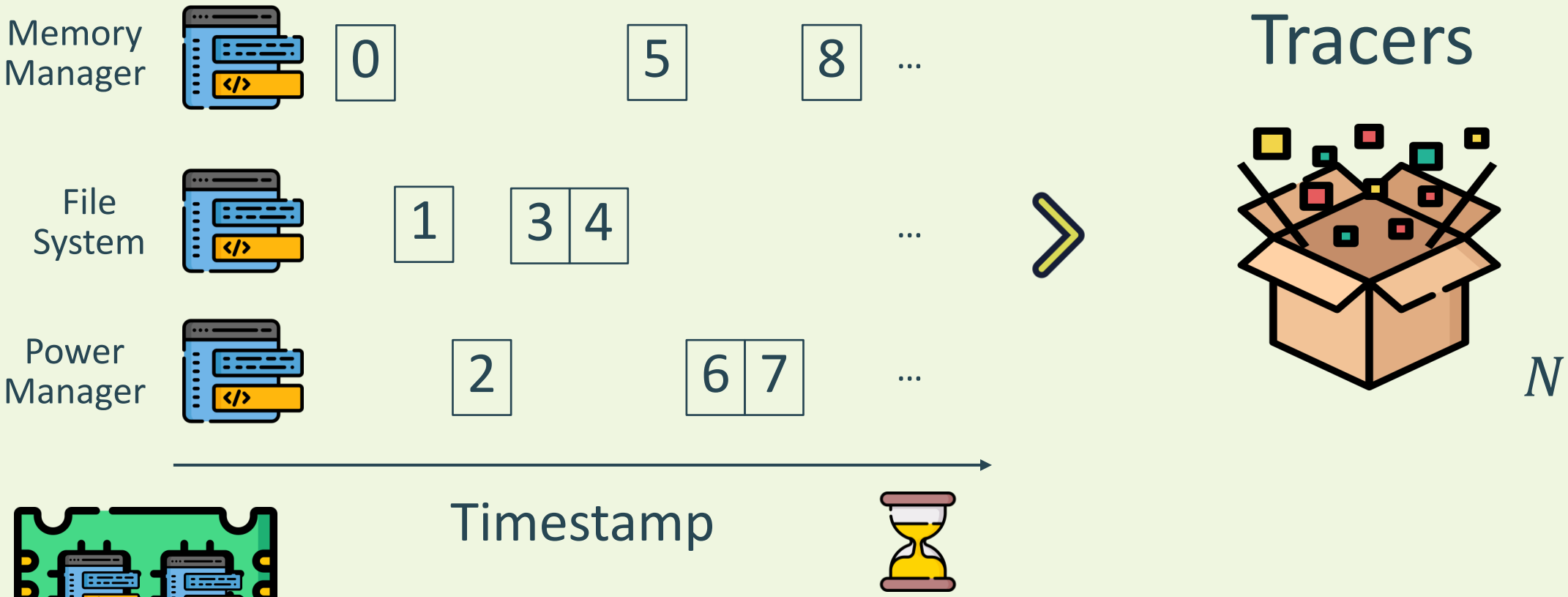
Shopping App.

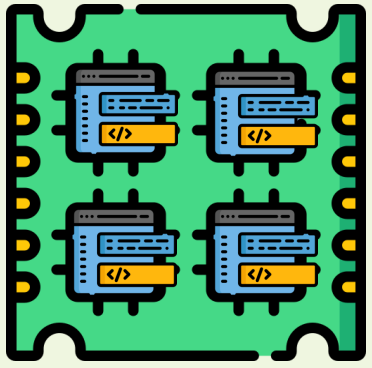
Tracers



N

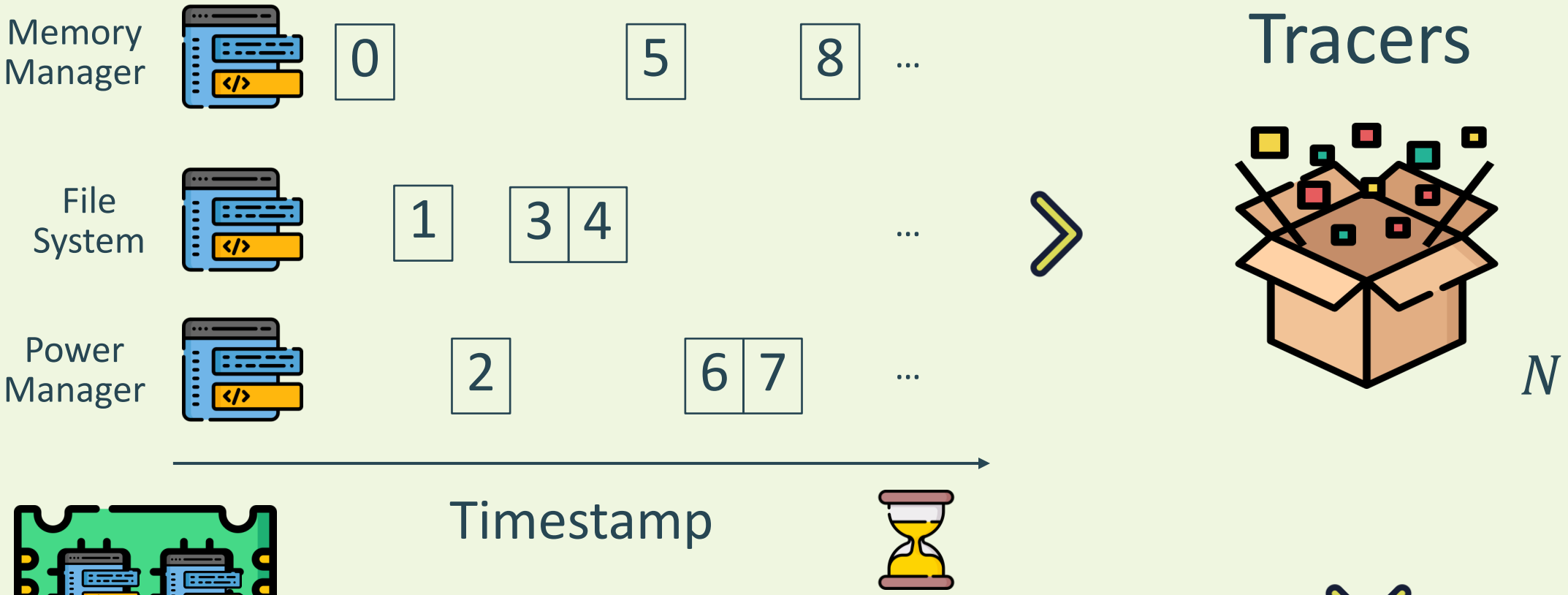
Existing Tracers on Mobiles



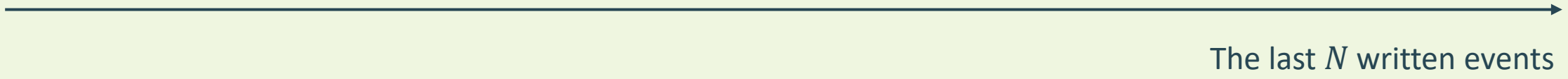


Shopping App.

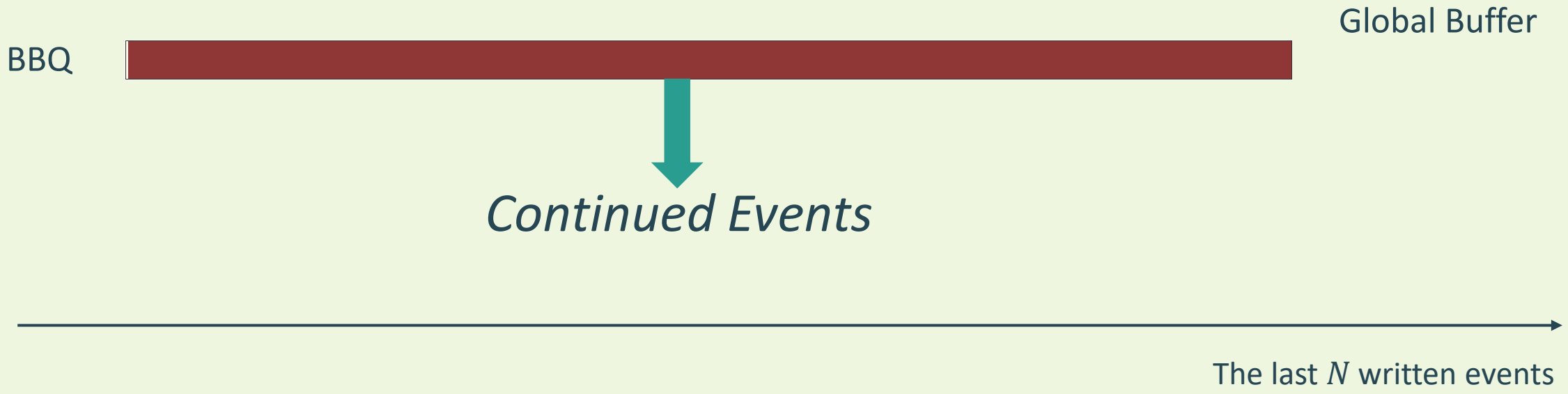
Existing Tracers on Mobiles



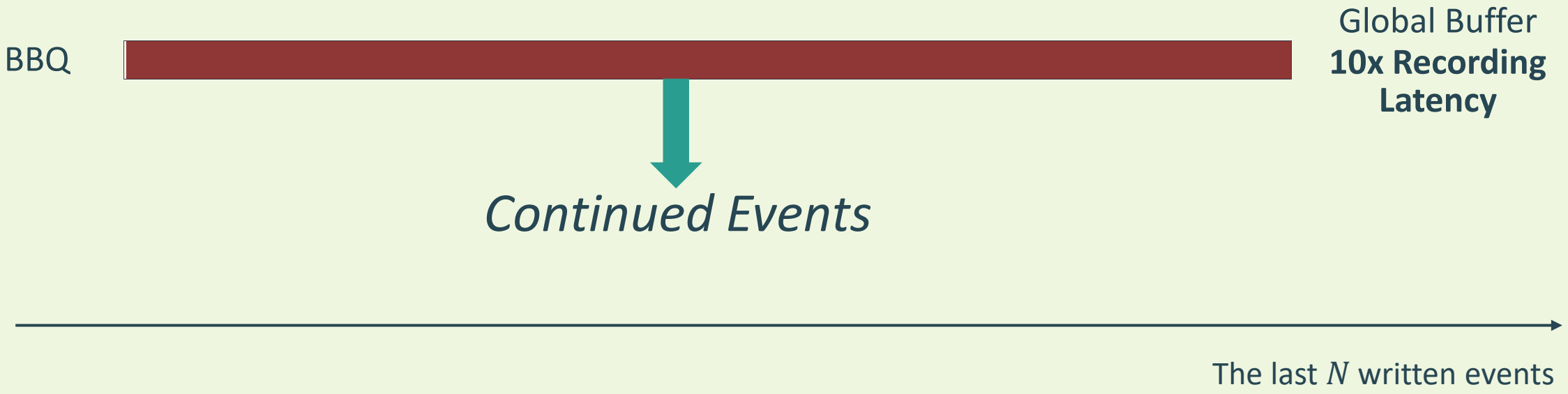
Existing Tracers on Mobiles



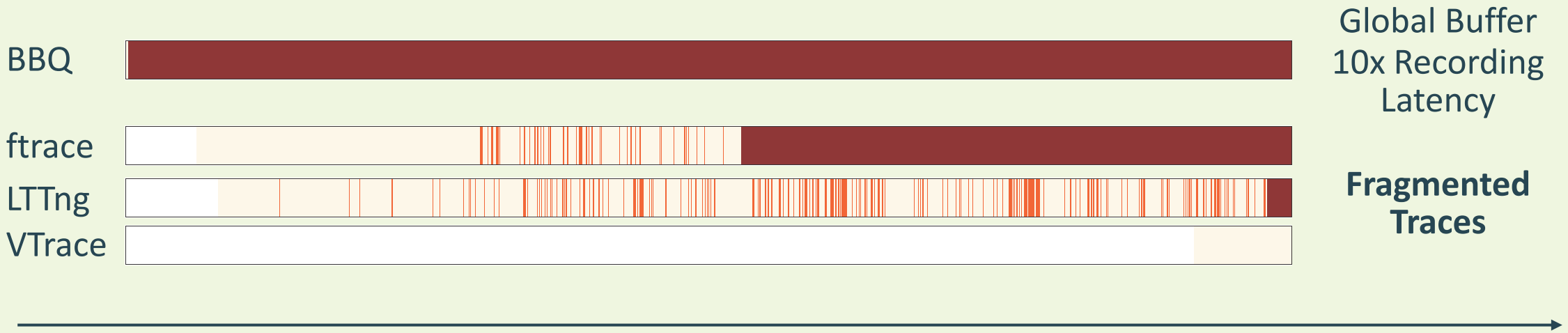
Existing Tracers on Mobiles



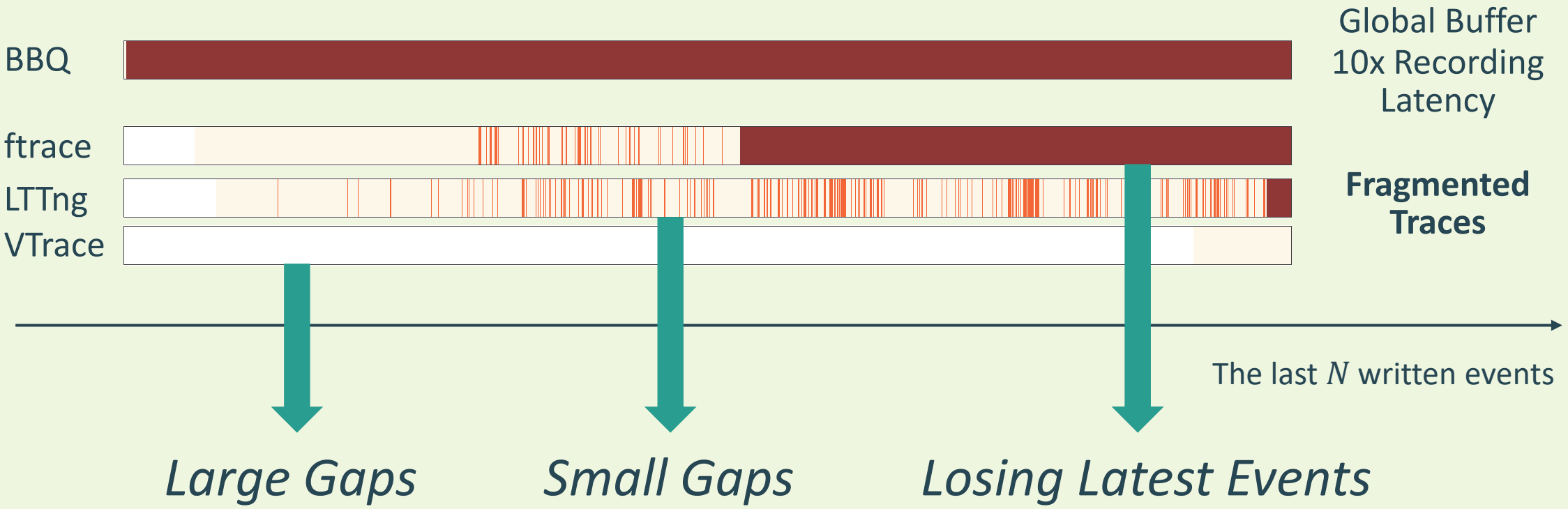
Existing Tracers on Mobiles



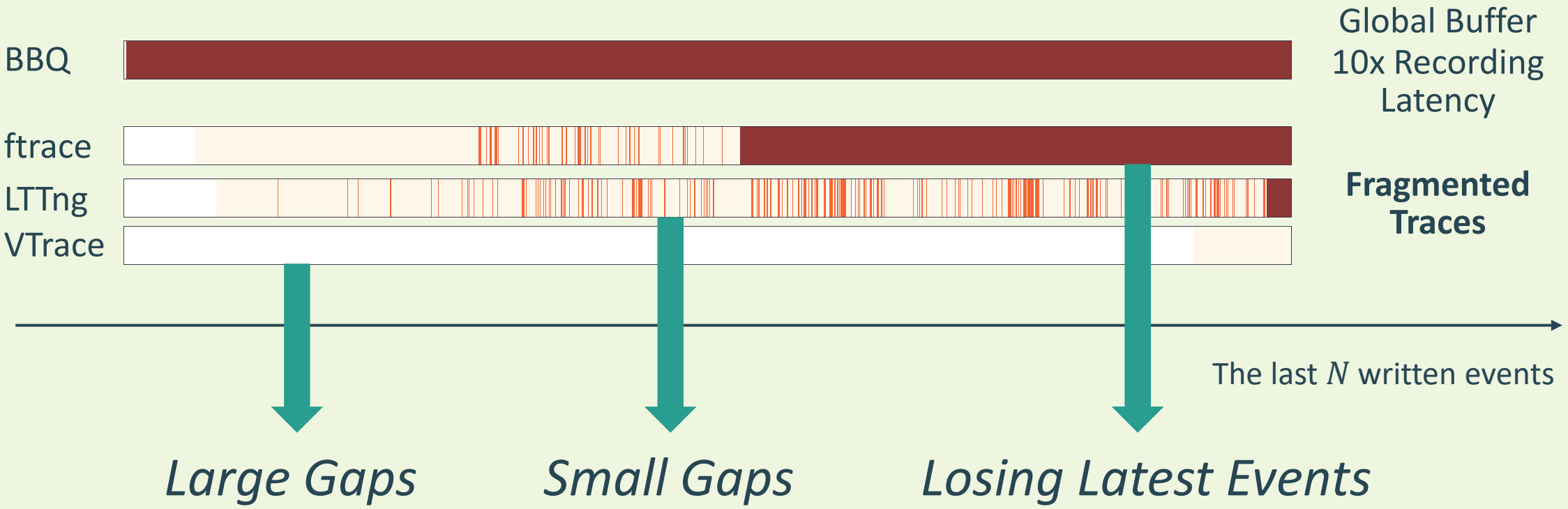
Existing Tracers on Mobiles



Existing Tracers on Mobiles



Existing Tracers on Mobiles



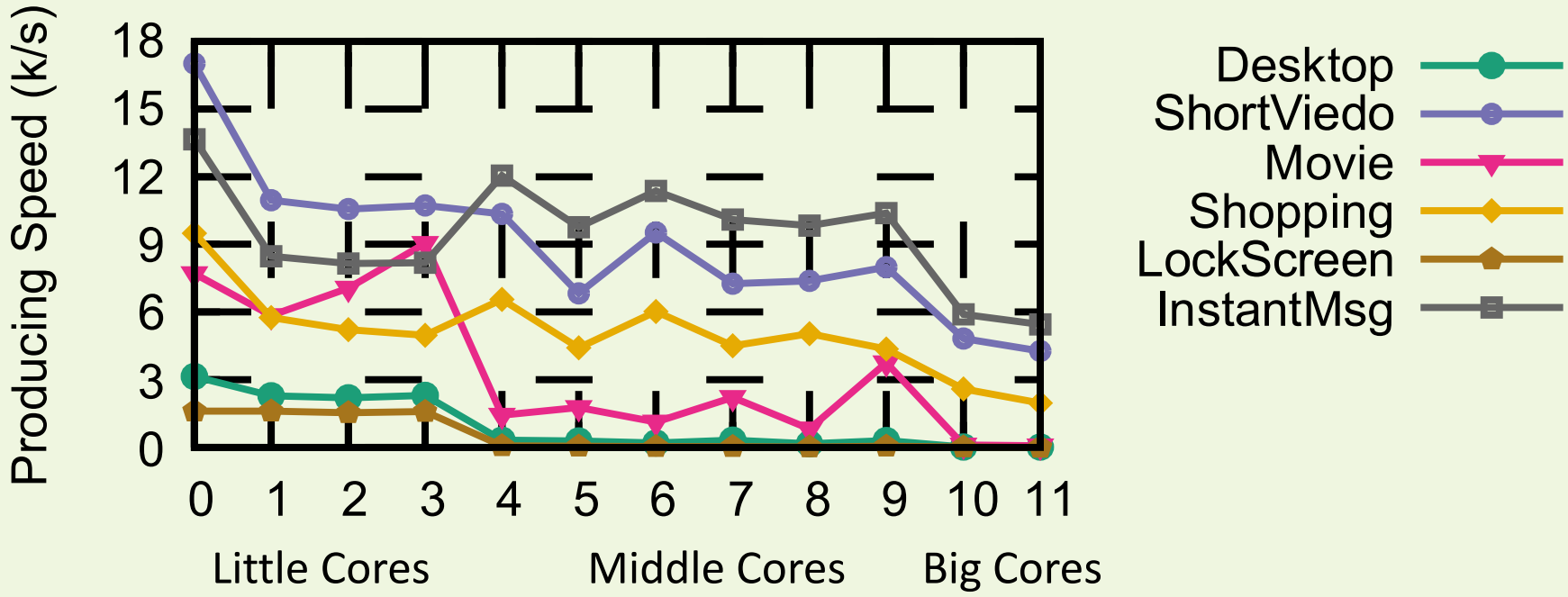
Why It's Happening?

Why This Matters?

Highly Imbalanced Trace Speed

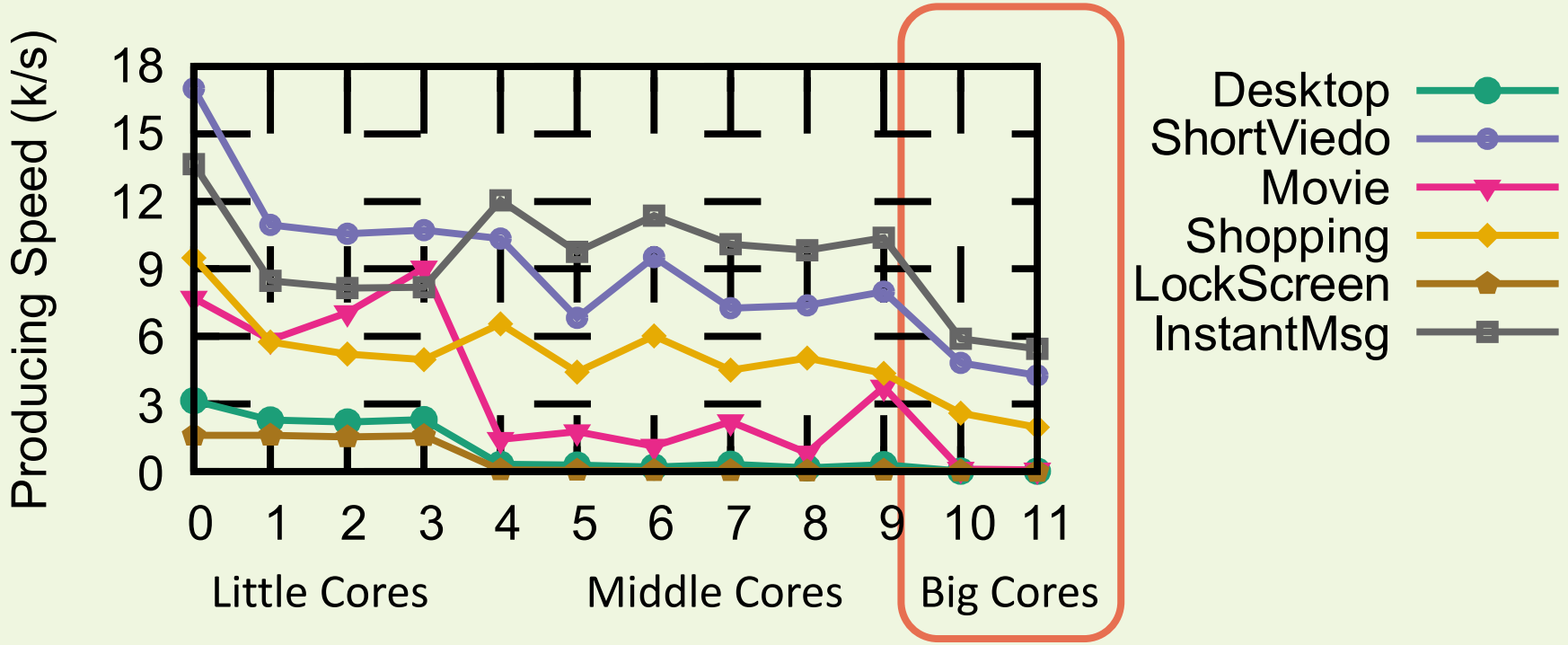
Highly Imbalanced Trace Speed

Trace producing speeds of different cores



Highly Imbalanced Trace Speed

Trace producing speeds of different cores

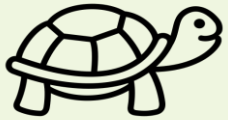


Idle big cores to save energy

Highly Imbalanced Trace Speed

Little Core

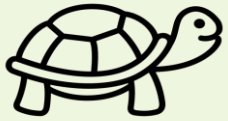
Big Core

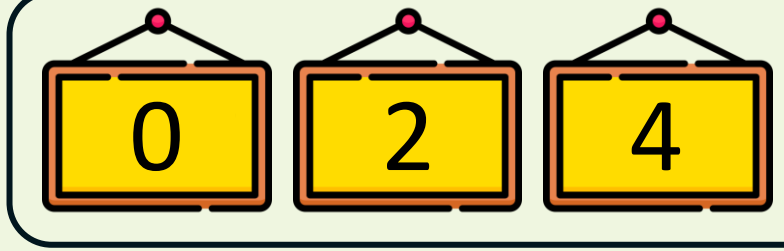


Highly Imbalanced Trace Speed

Little Core

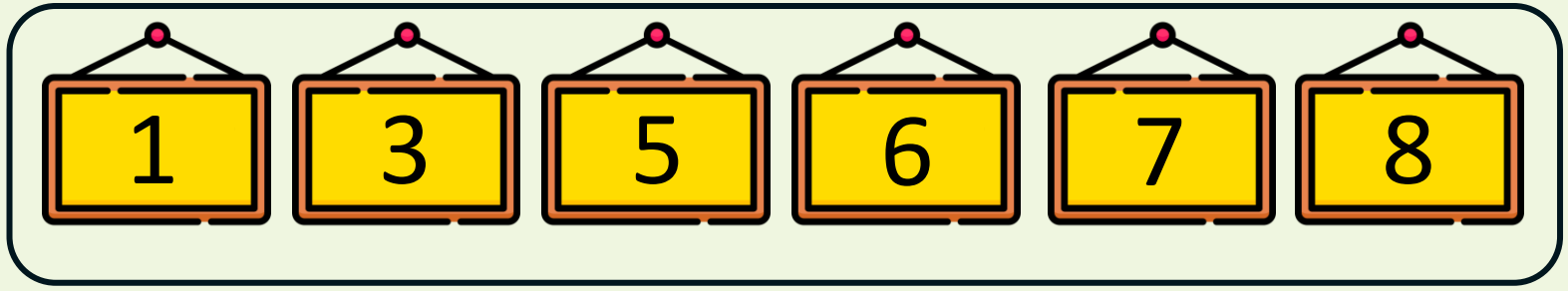
Big Core



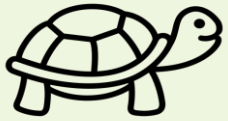


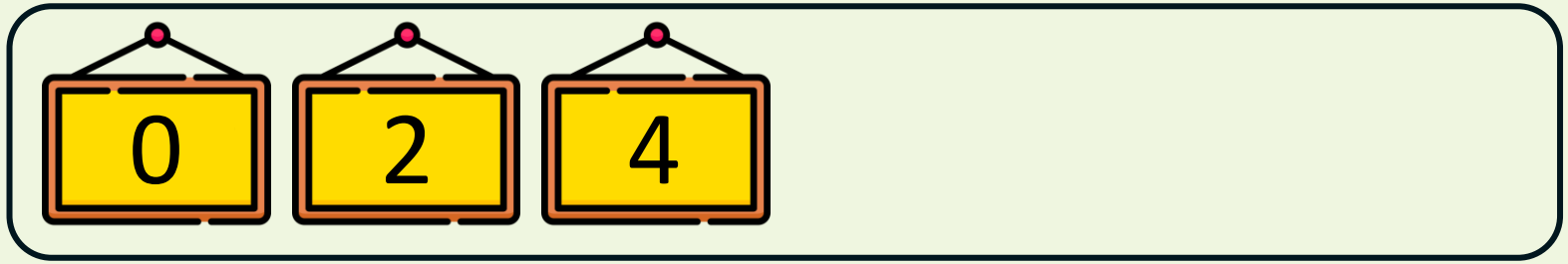
Highly Imbalanced Trace Speed

Little Core



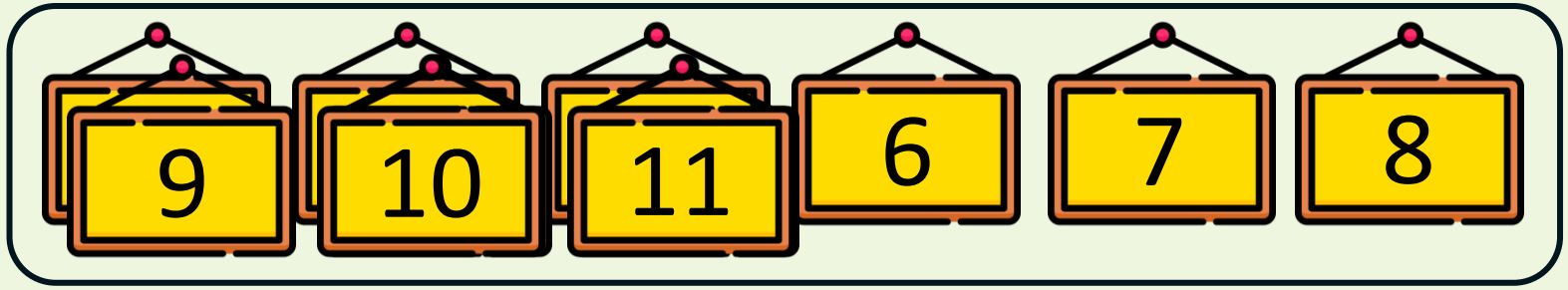
Big Core



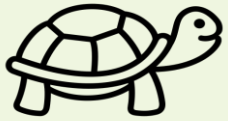


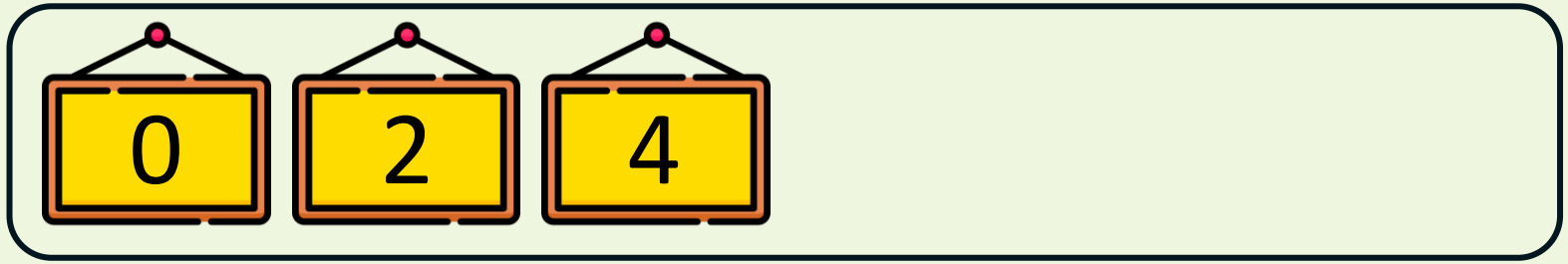
Highly Imbalanced Trace Speed

Little Core

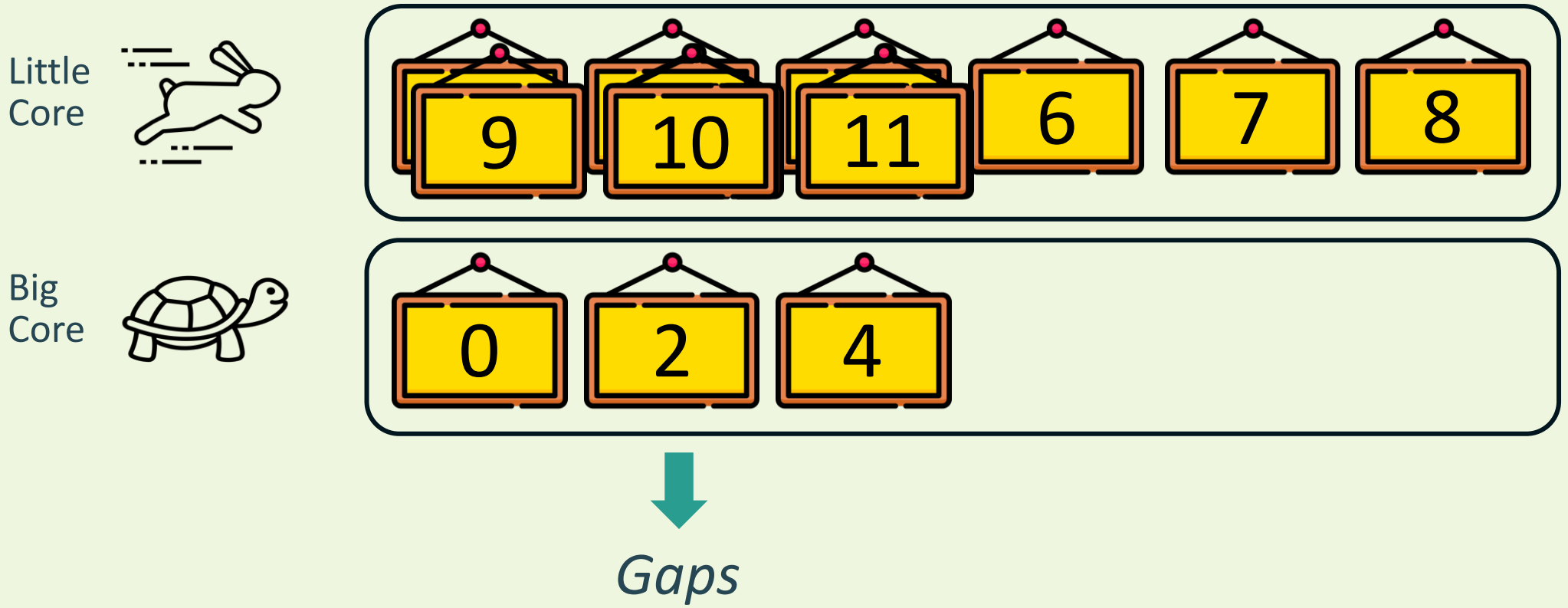


Big Core

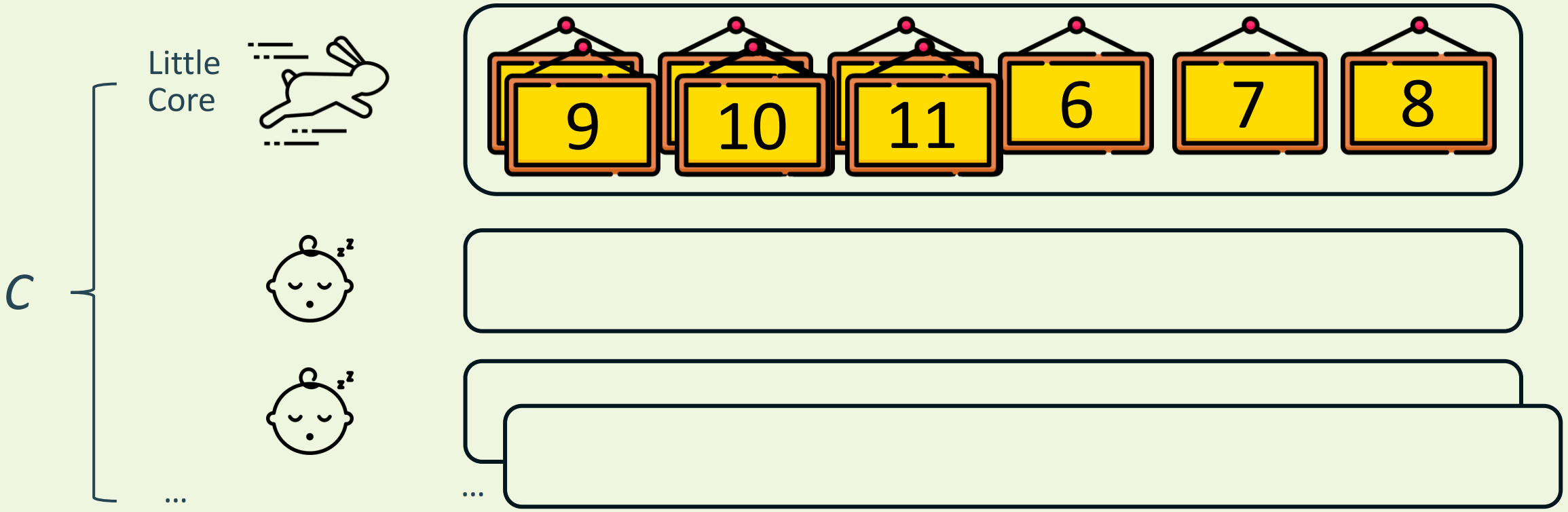




Highly Imbalanced Trace Speed



Highly Imbalanced Trace Speed

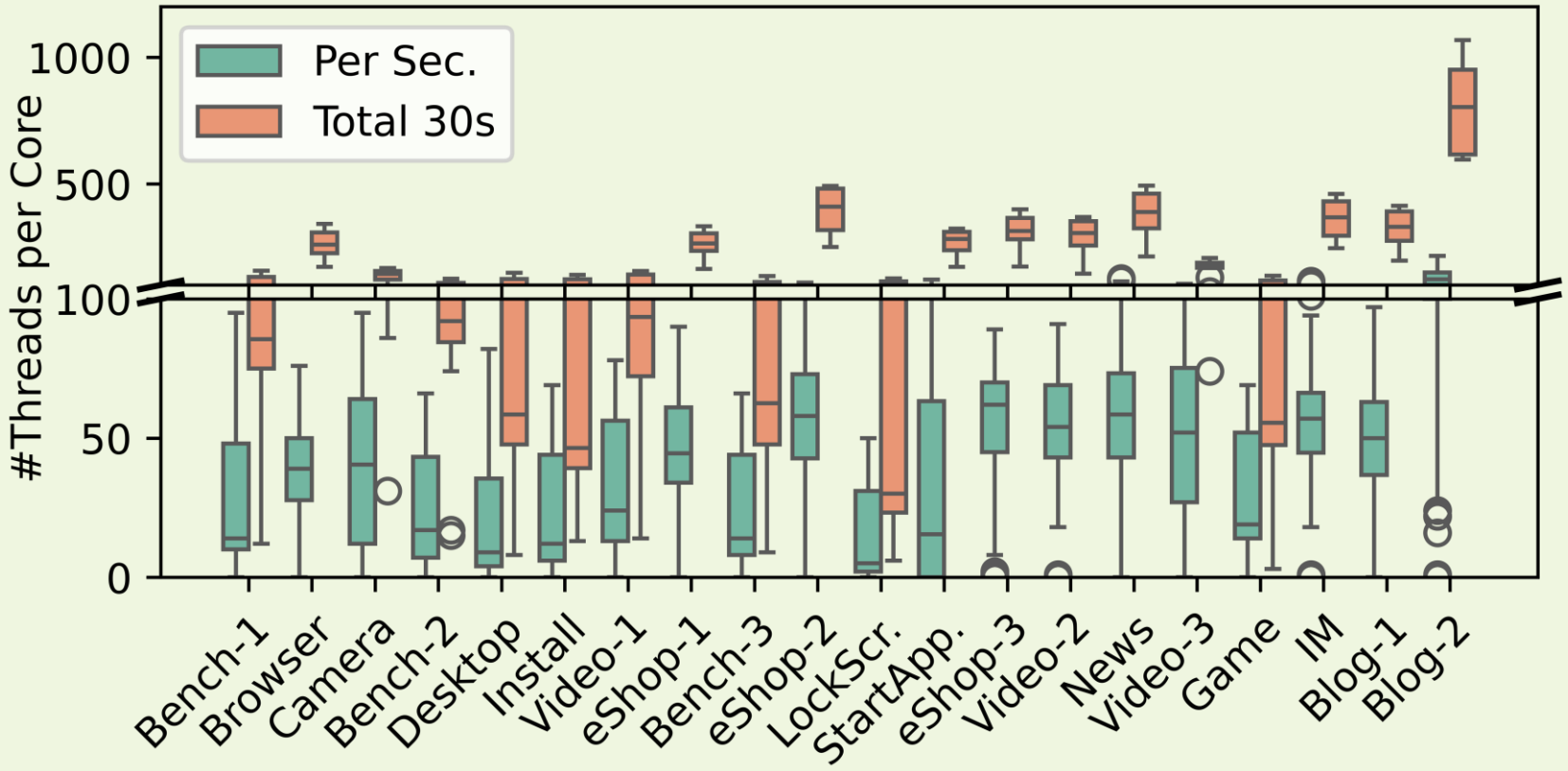


Worst Case: utilization = 1/C, effectivity ratio = 1/C

Core Oversubscription

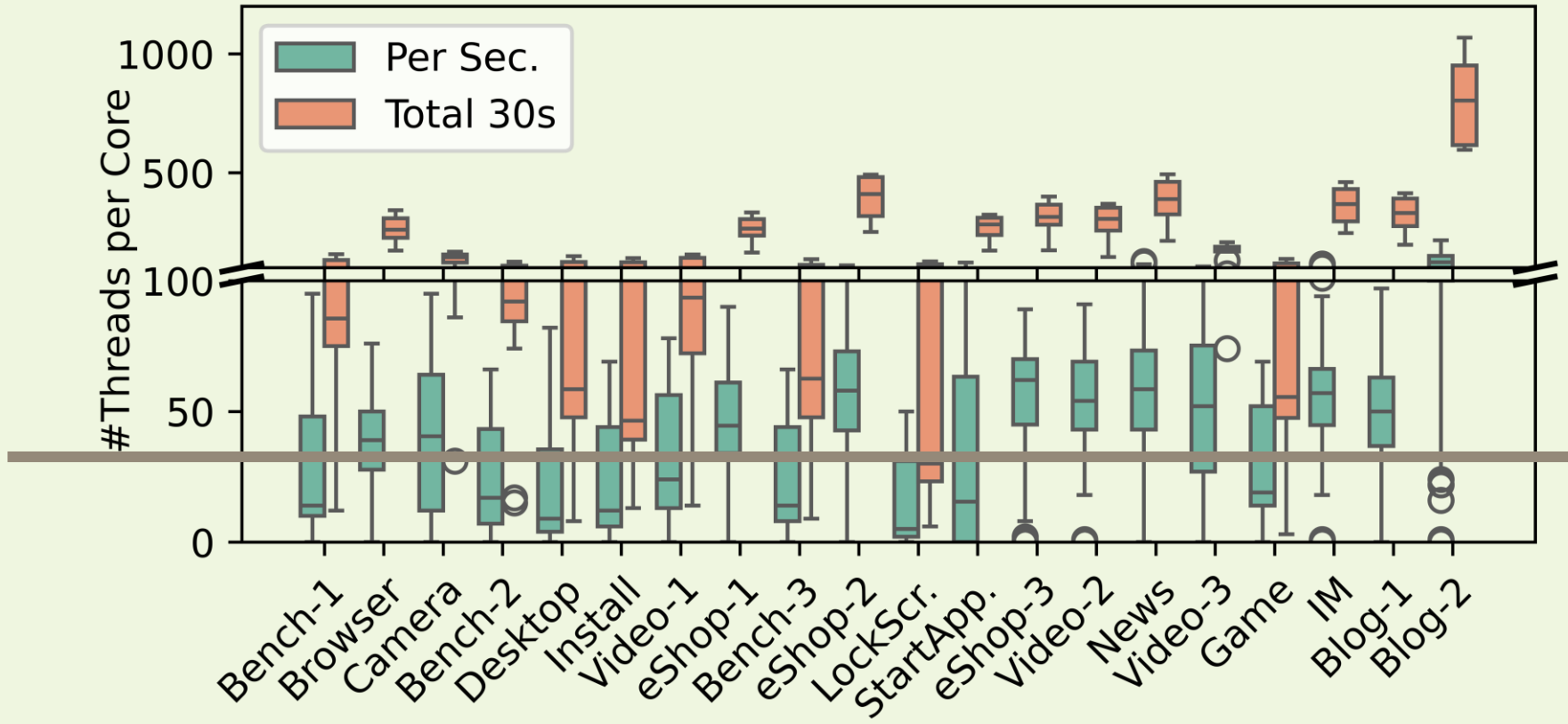
Core Oversubscription

Thread number per core producing traces



Core Oversubscription

Thread number per core producing traces

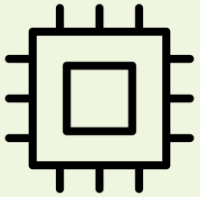


30

An average of 30 threads per second

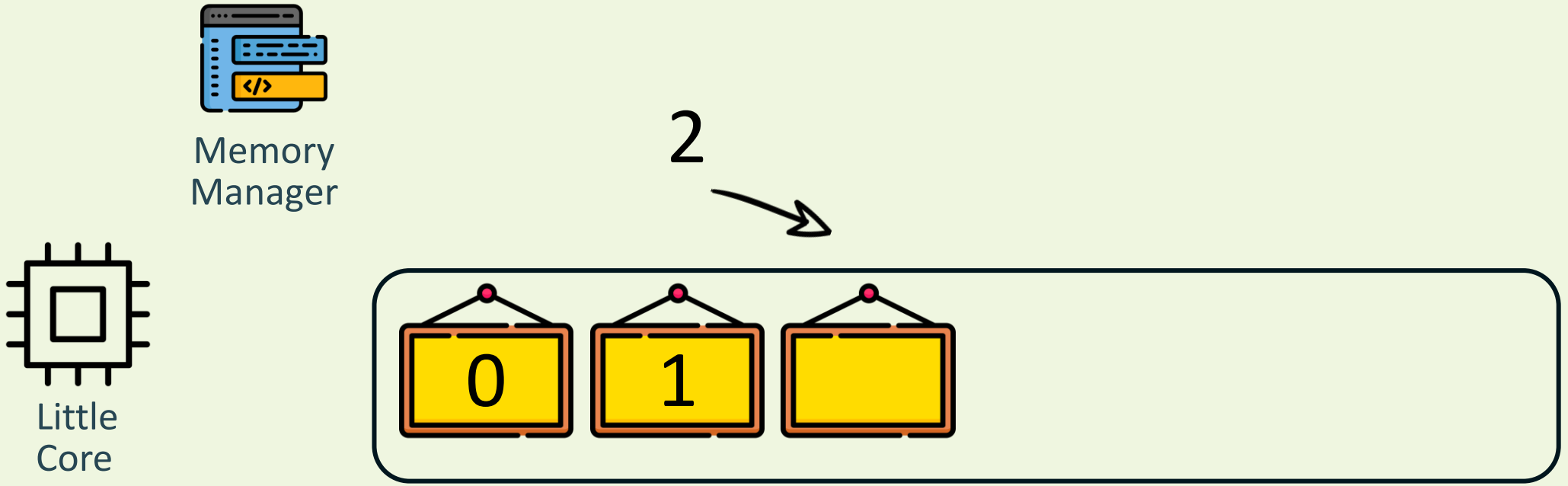
Core Oversubscription

Memory Manager

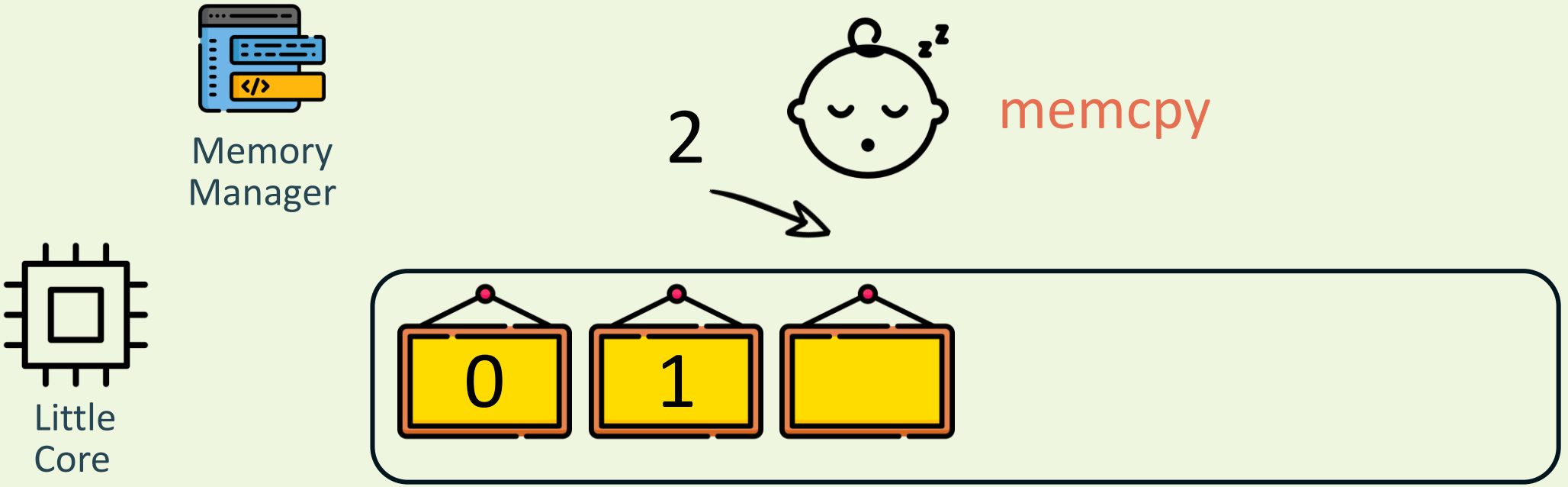


Little Core

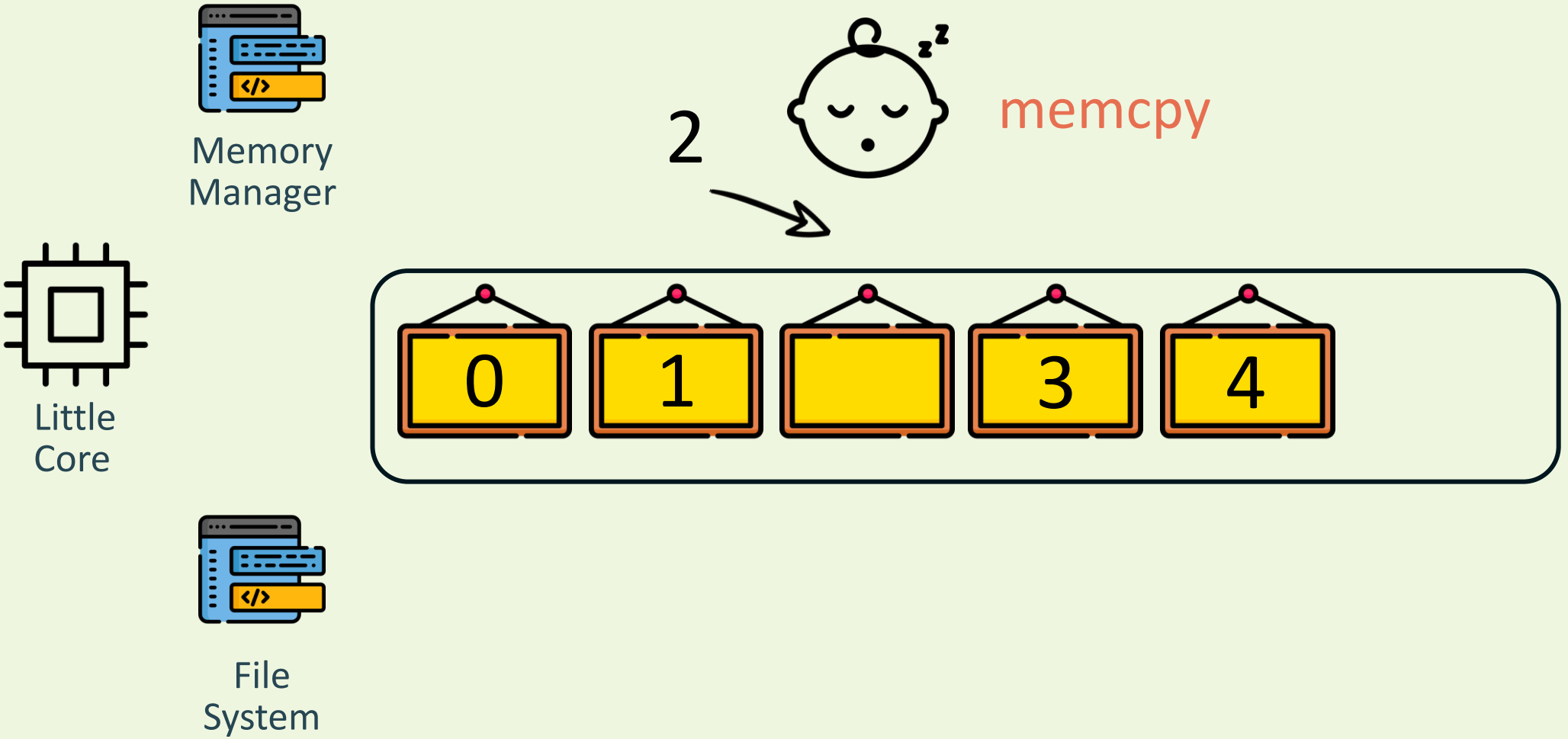
Core Oversubscription



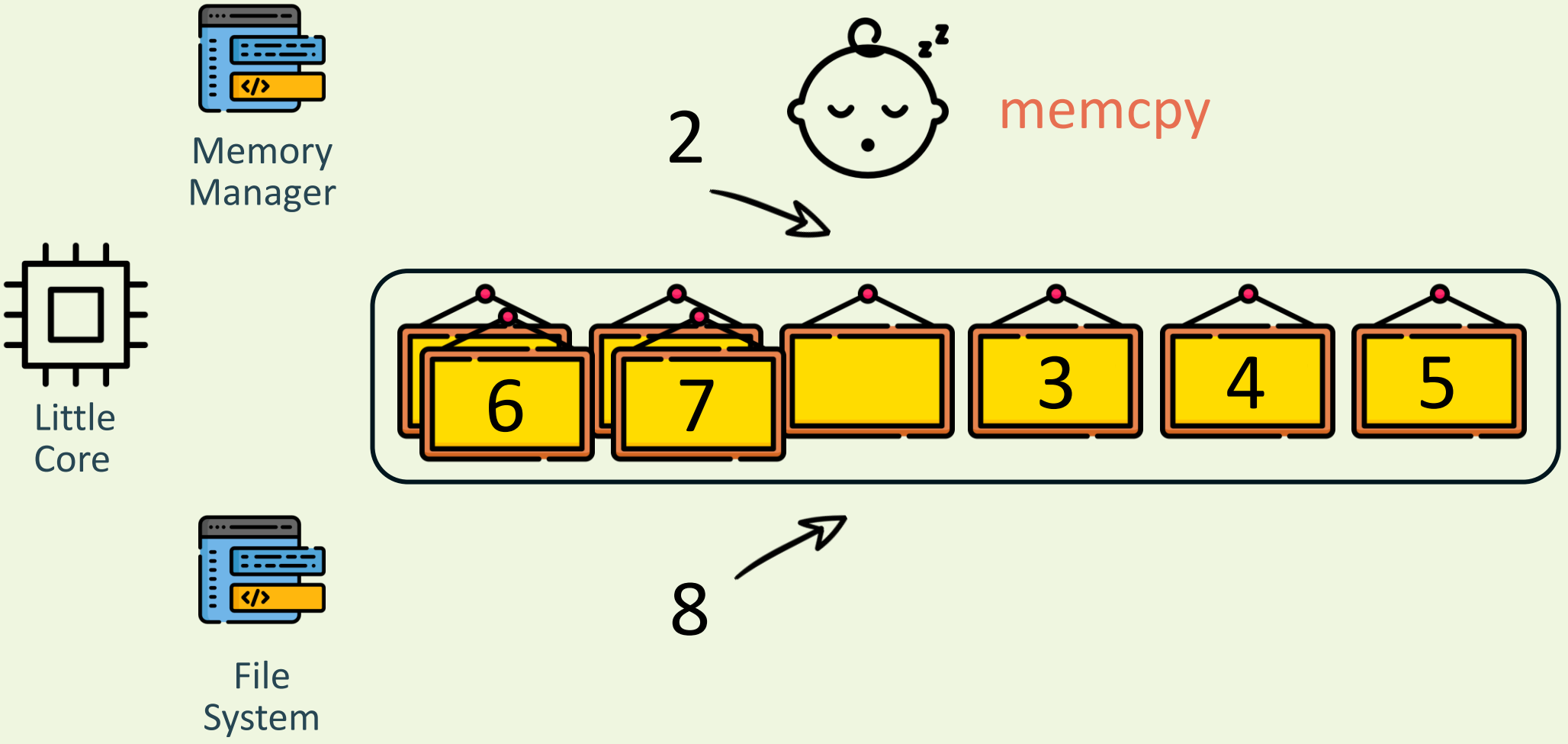
Core Oversubscription



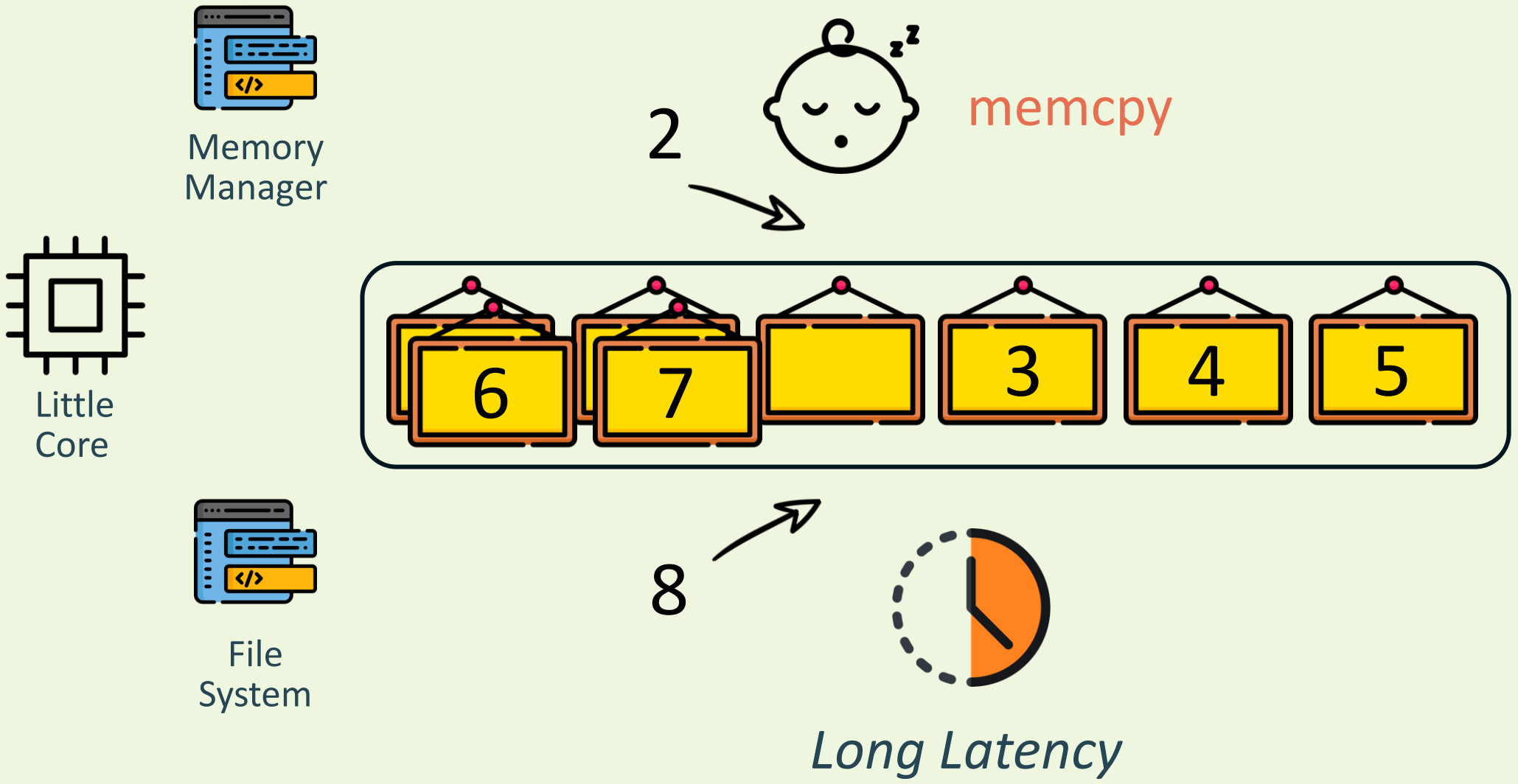
Core Oversubscription



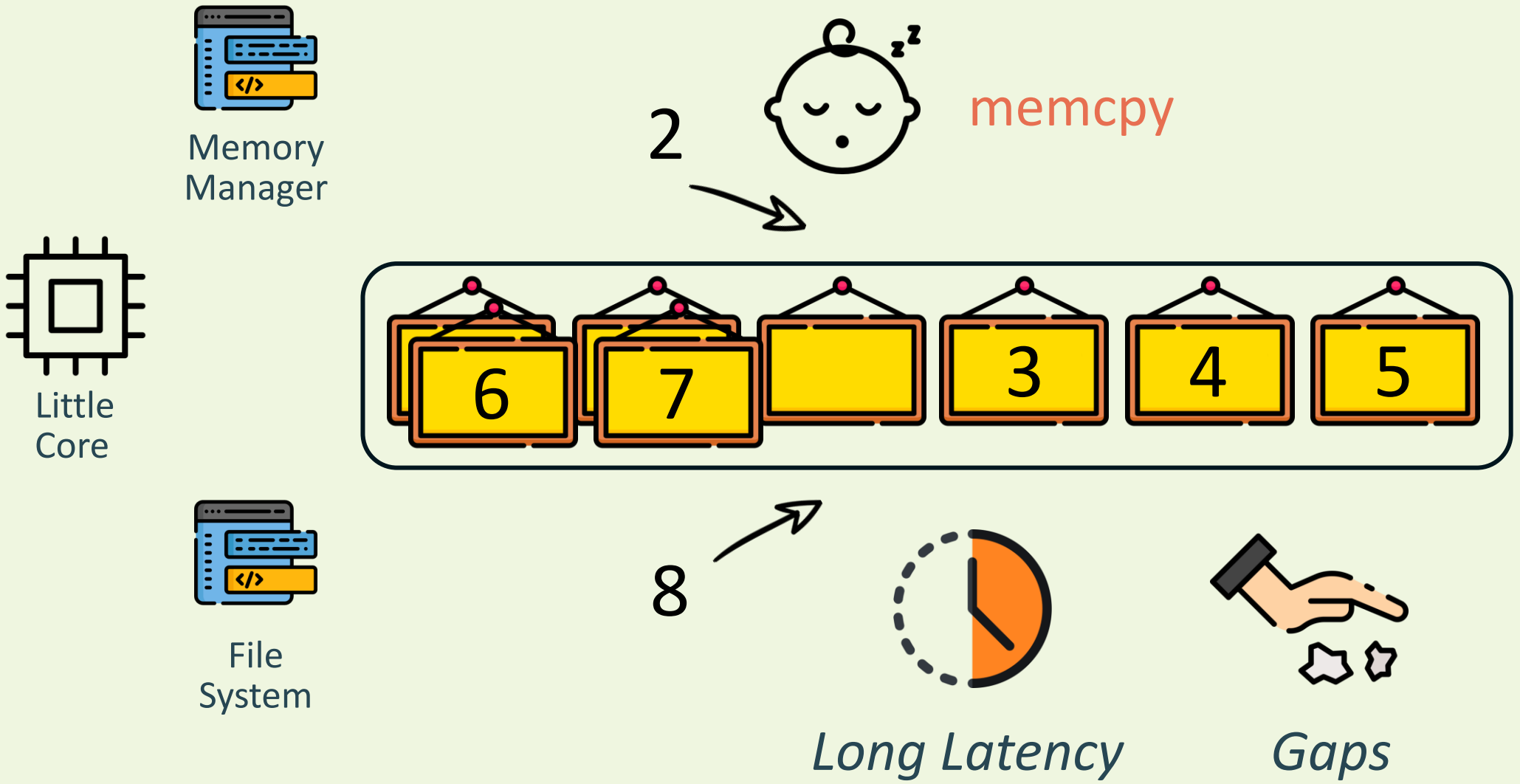
Core Oversubscription



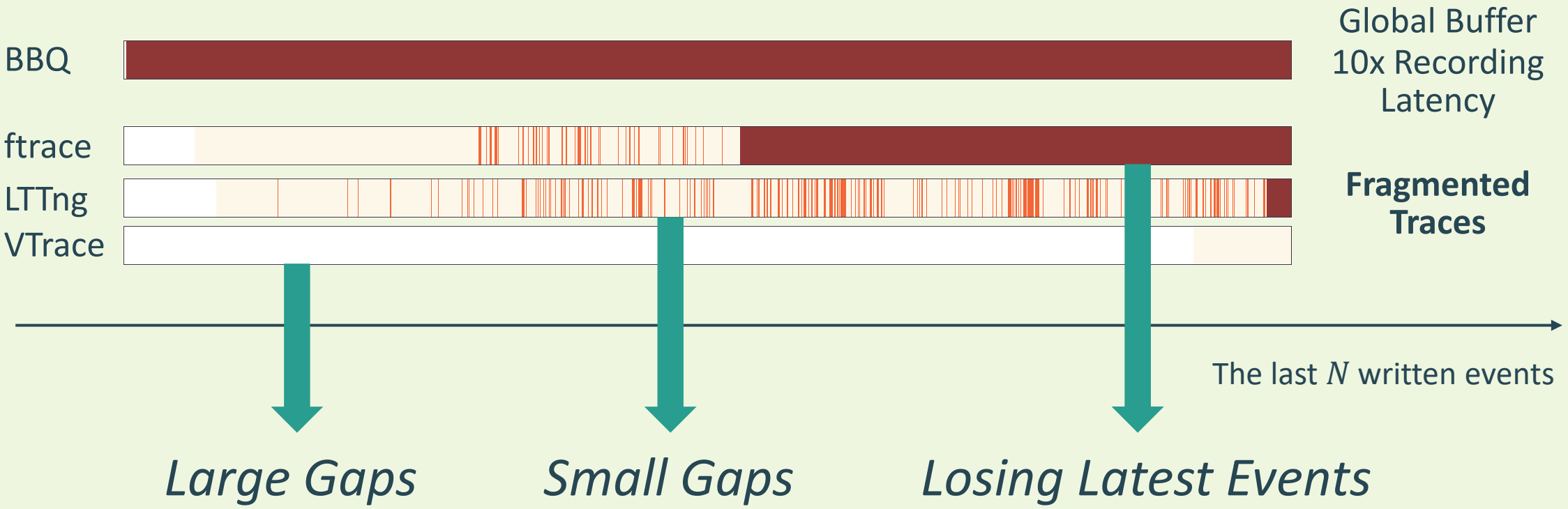
Core Oversubscription



Core Oversubscription



Existing Tracers on Mobiles



Why It's Happening?

Why This Matters?

Trace Requirements for Mobiles

Tracers have to store detailed traces over a long time duration.

Trace Requirements for Mobiles

Tracers have to store detailed traces over a long time duration.

Energy Defects	Detailed scheduling and frequency adjustment decisions (100MB/min)
Silent Defects (freezing)	Tracing longer than the timeout (e.g., 20s)

Trace Requirements for Mobiles

Tracers have to store detailed traces over a long time duration.

Energy Defects	Detailed scheduling and frequency adjustment decisions (100MB/min)
Silent Defects (freezing)	Tracing longer than the timeout (e.g., 20s)

BTrace can store all related trace in 30s with a 450MB buffer.

Trace Requirements for Mobiles

Tracers have to store detailed traces over a long time duration.

Energy Defects	Detailed scheduling and frequency adjustment decisions (100MB/min)
Silent Defects (freezing)	Tracing longer than the timeout (e.g., 20s)

BTrace can store all related trace in 30s with a 450MB buffer.

Increasing the buffer size?

Trace Requirements for Mobiles

Tracers have to store detailed traces over a long time duration.

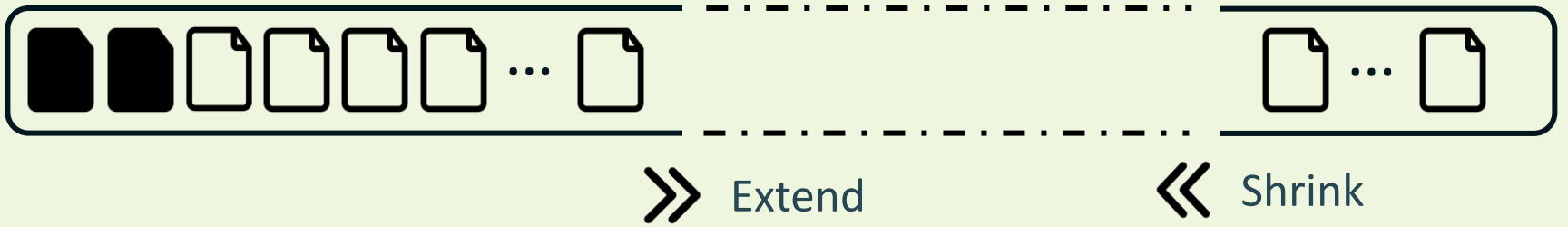
Energy Defects	Detailed scheduling and frequency adjustment decisions (100MB/min)
Silent Defects (freezing)	Tracing longer than the timeout (e.g., 20s)

BTrace can store all related trace in 30s with a 450MB buffer.

Increasing the buffer size?  Limited capacity of smartphones: 4GB-8GB memory

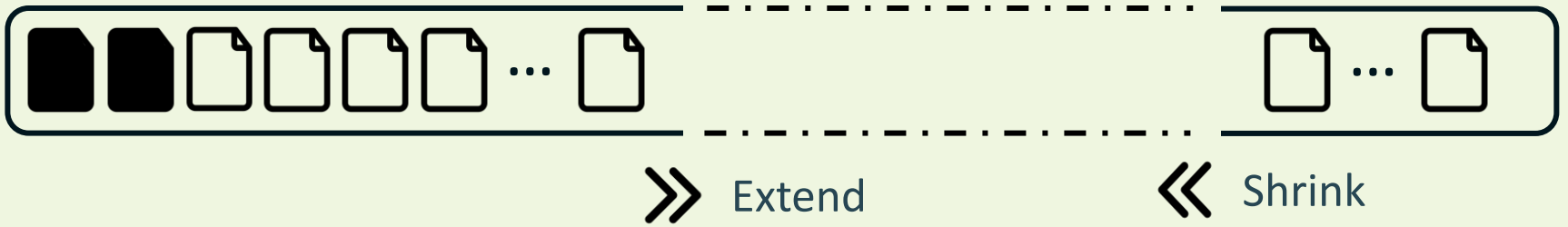
Trace Requirements for Mobiles

Buffer Resizing



Trace Requirements for Mobiles

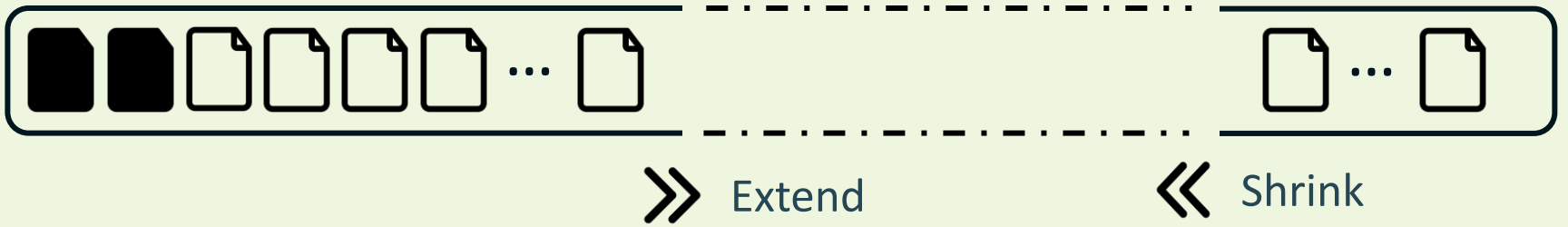
Buffer Resizing



Shrinking typically requires a safe memory reclamation mechanism.

Trace Requirements for Mobiles

Buffer Resizing

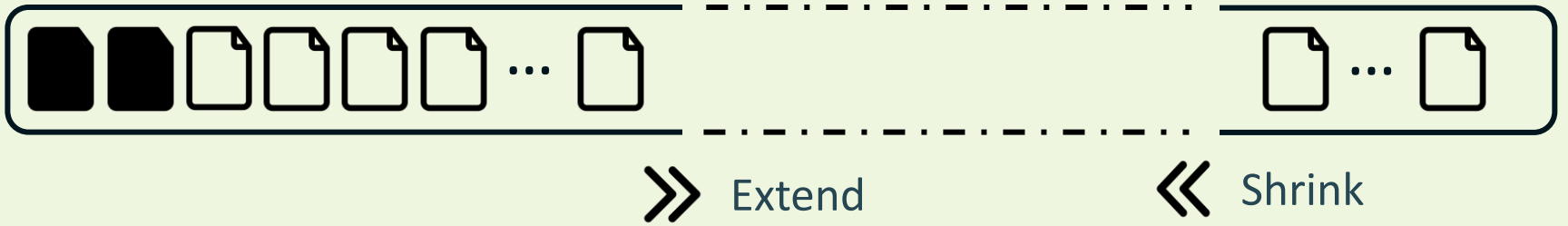


Shrinking typically requires a safe memory reclamation mechanism.

- Kernel Space: Disable Preemption 

Trace Requirements for Mobiles

Buffer Resizing

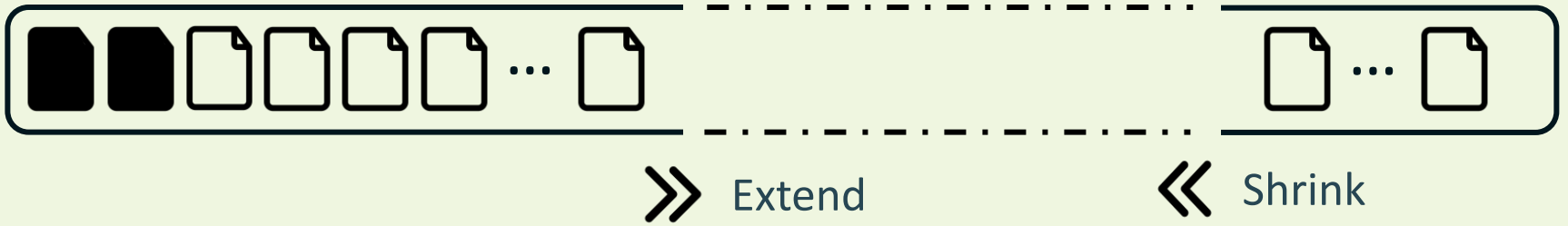


Shrinking typically requires a safe memory reclamation mechanism.

- Kernel Space: Disable Preemption 
- User Space: Disable Preemption 
  RC, RCU, … 

Trace Requirements for Mobiles

Buffer Resizing

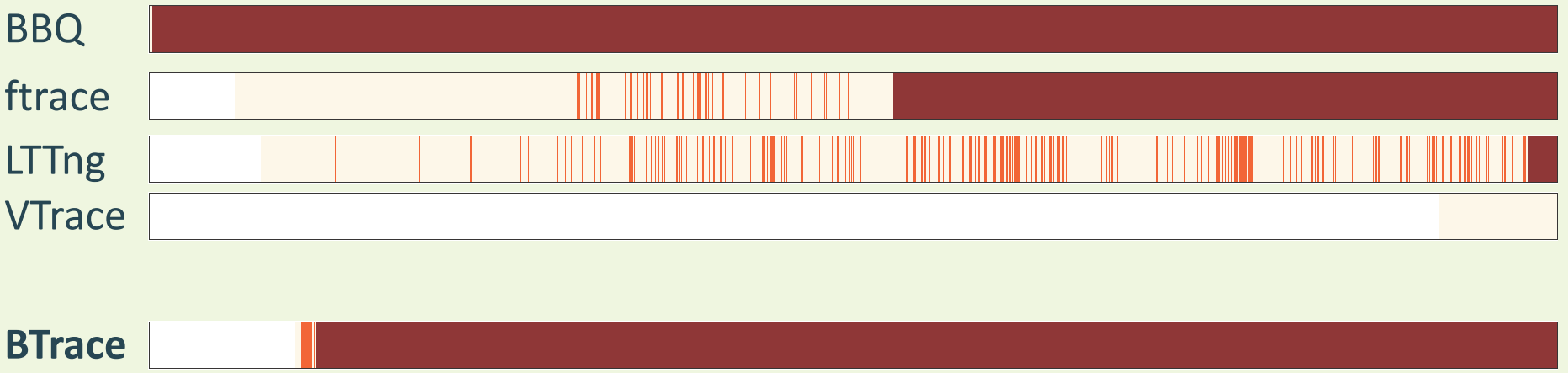


Shrinking typically requires a safe memory reclamation mechanism.

- Kernel Space: Disable Preemption 
- User Space: Disable Preemption 
  RC, RCU, … 

A Better Solution?

BTrace

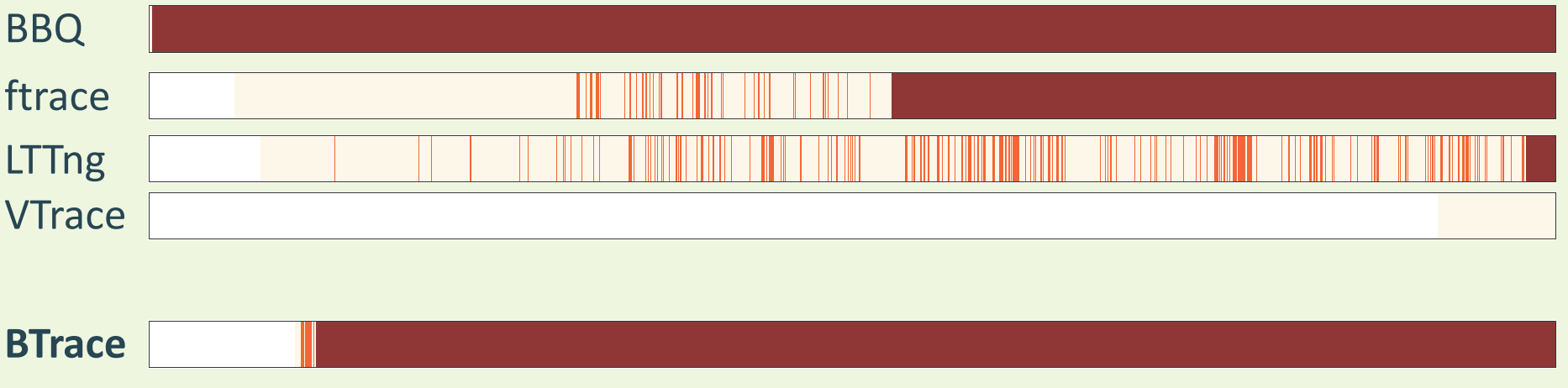


10x Recording Latency

Fragmented Traces

20% Lower Latency
Resizing without SMR

BTrace



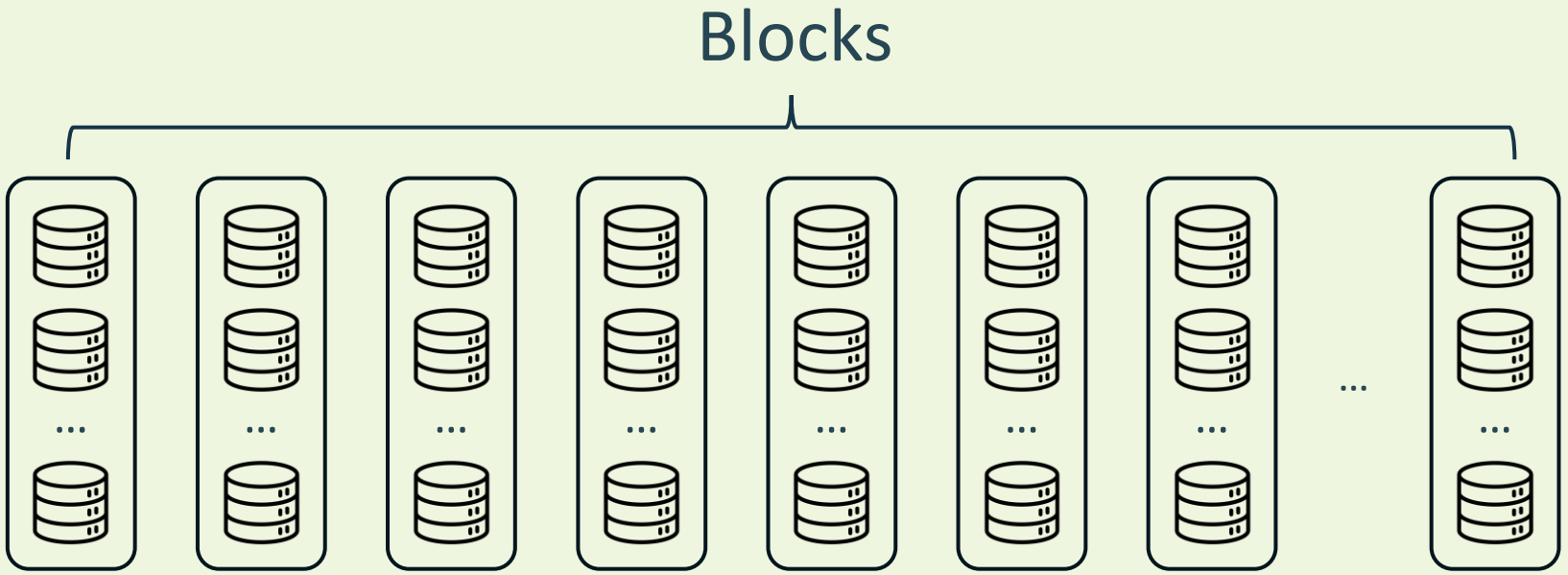
10x Recording Latency

Fragmented Traces

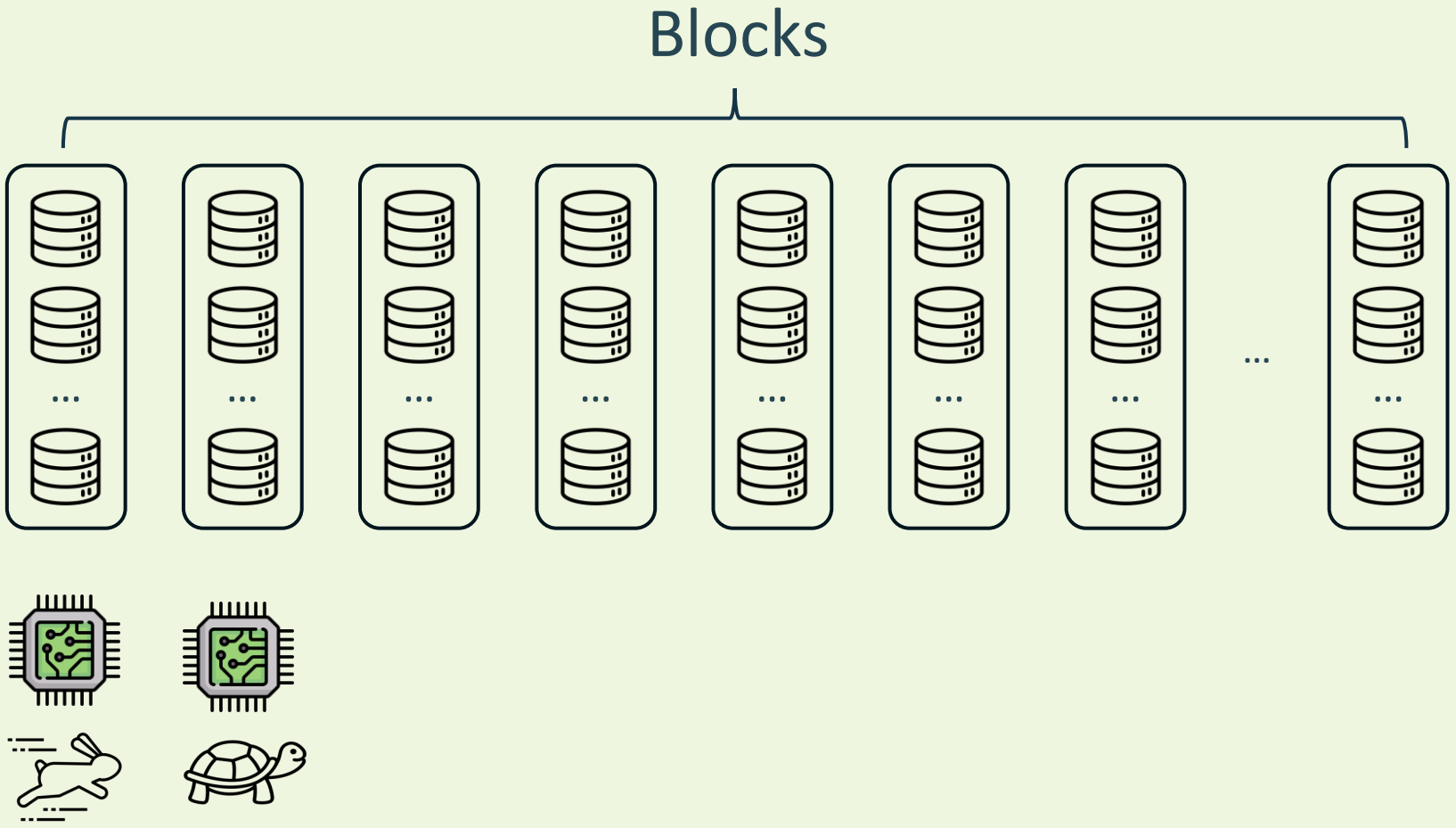
20% Lower Latency
Resizing without SMR

- Improving Utilization via Block Partitioning (more data, less fragments)
- Enhance Effectivity via Block Closing (more latest data)
- Ensure Availability via Block Skipping (less fragments)
- Enable Resizing via Implicit Reclaiming (user & kernel space resizing without SMR)

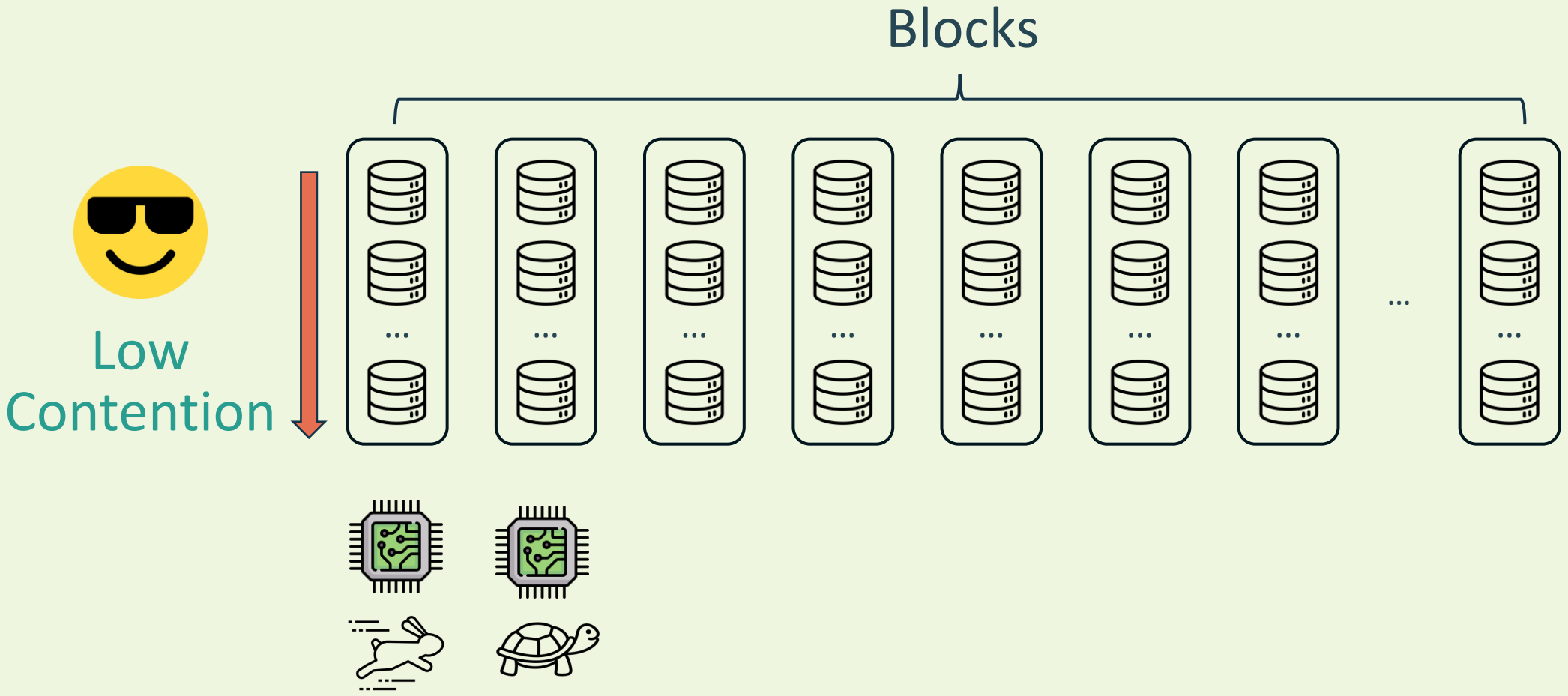
Improving Utilization via Block Partitioning



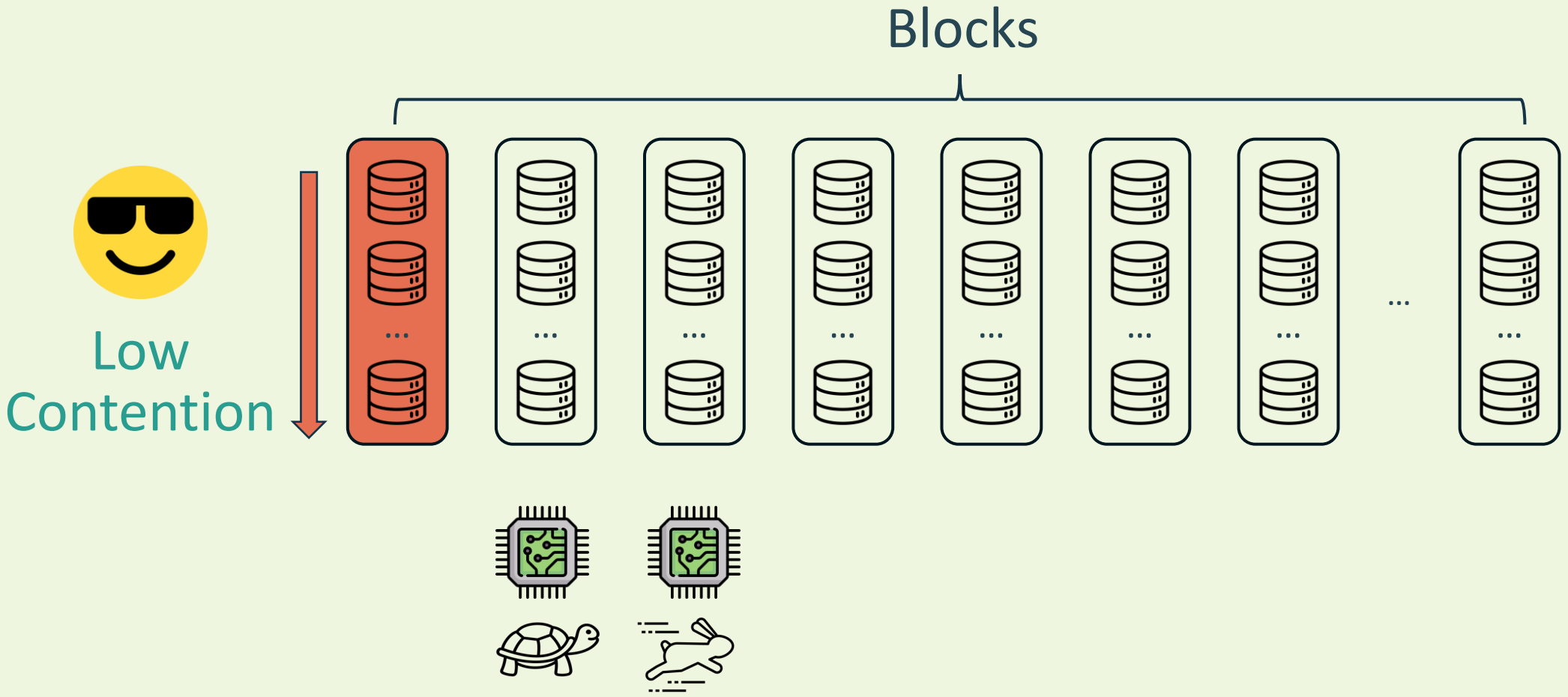
Improving Utilization via Block Partitioning



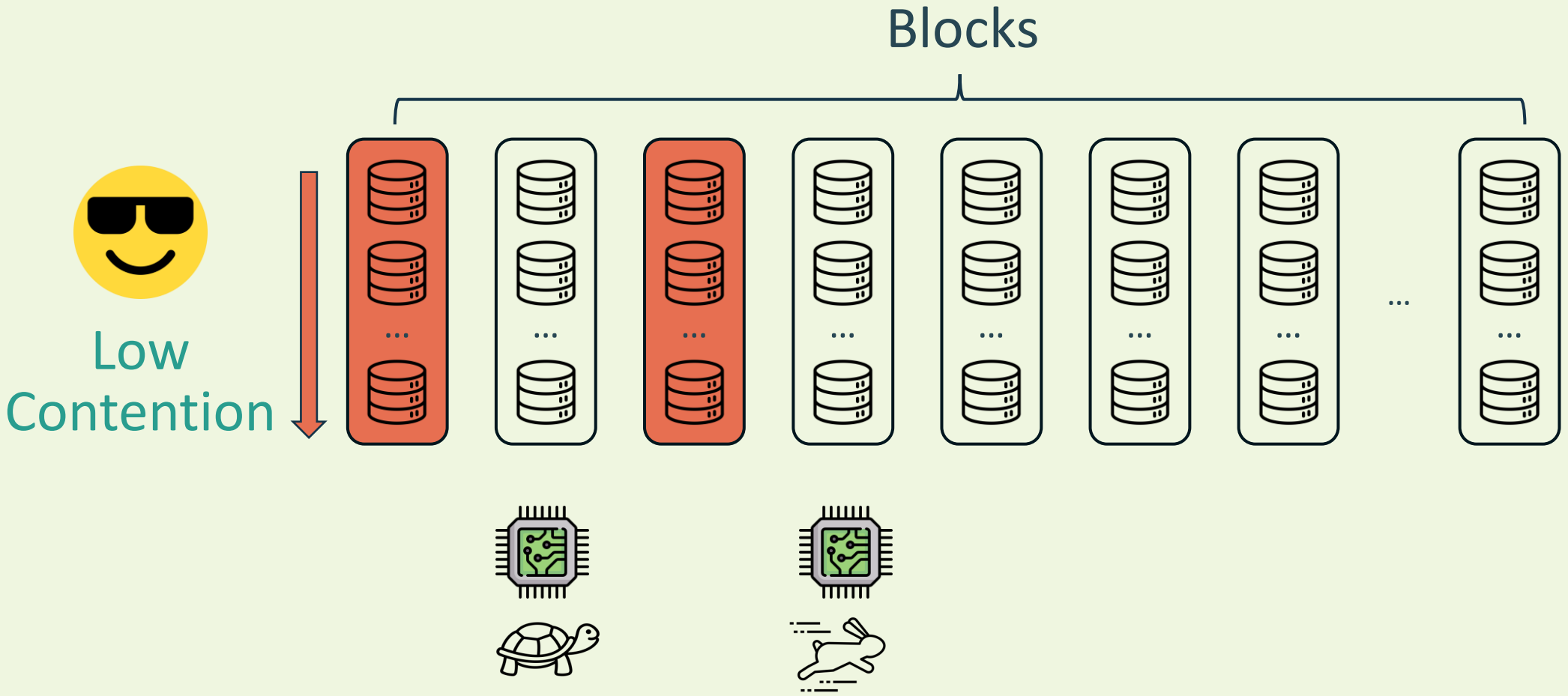
Improving Utilization via Block Partitioning



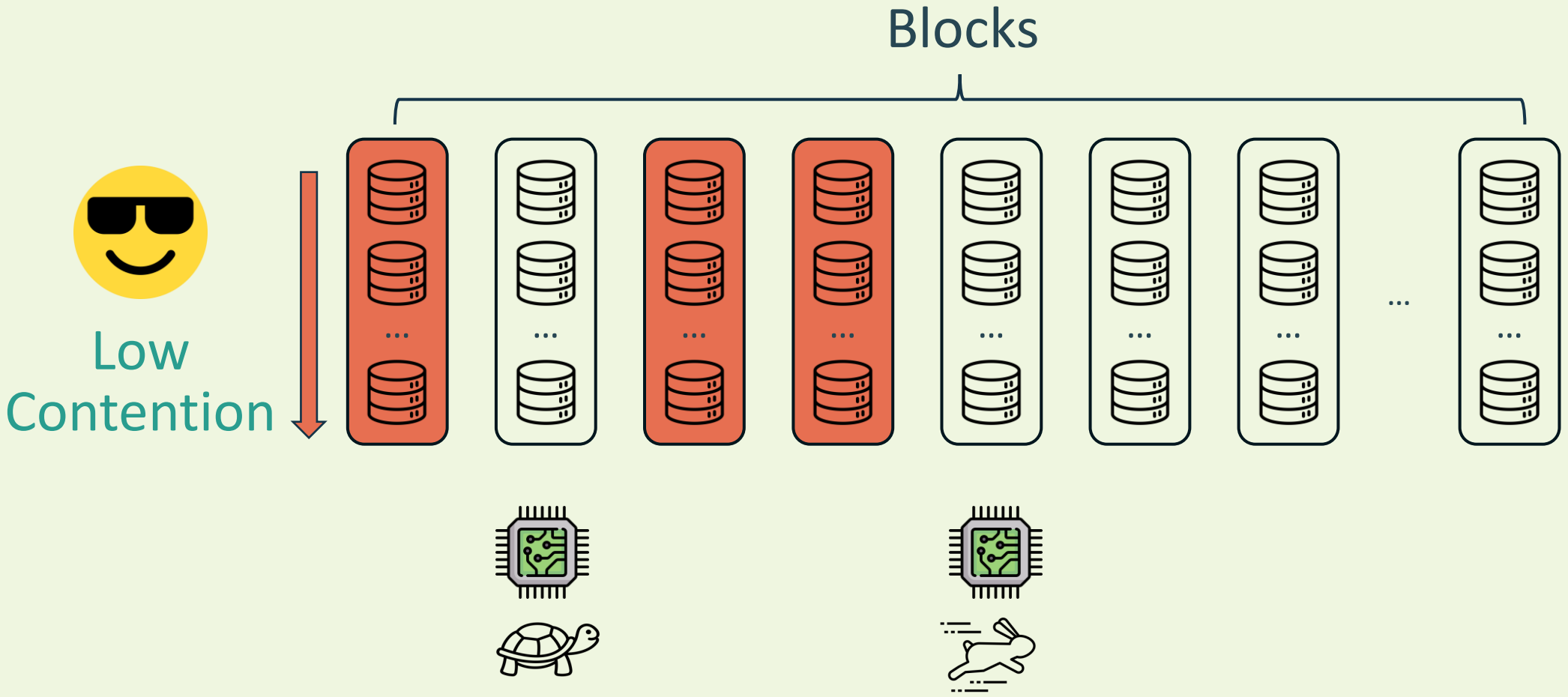
Improving Utilization via Block Partitioning



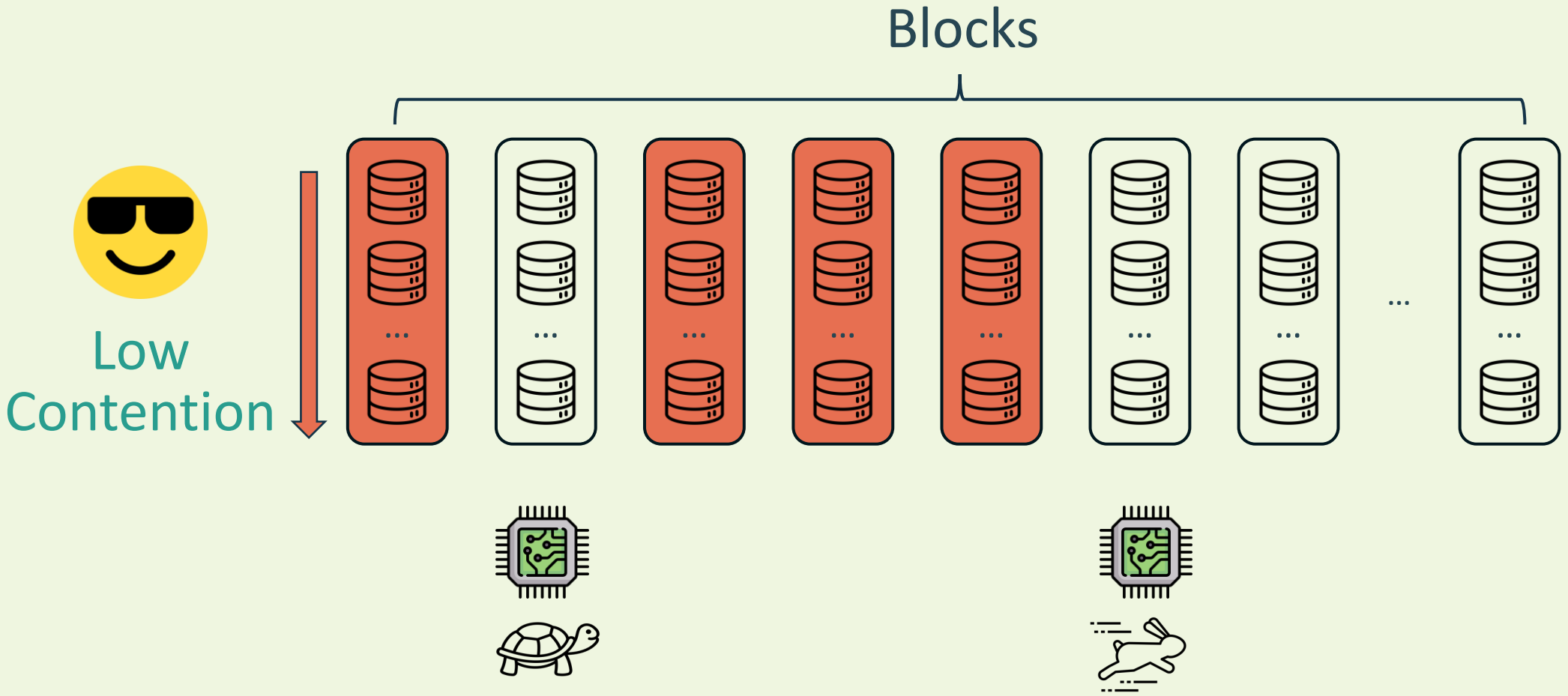
Improving Utilization via Block Partitioning



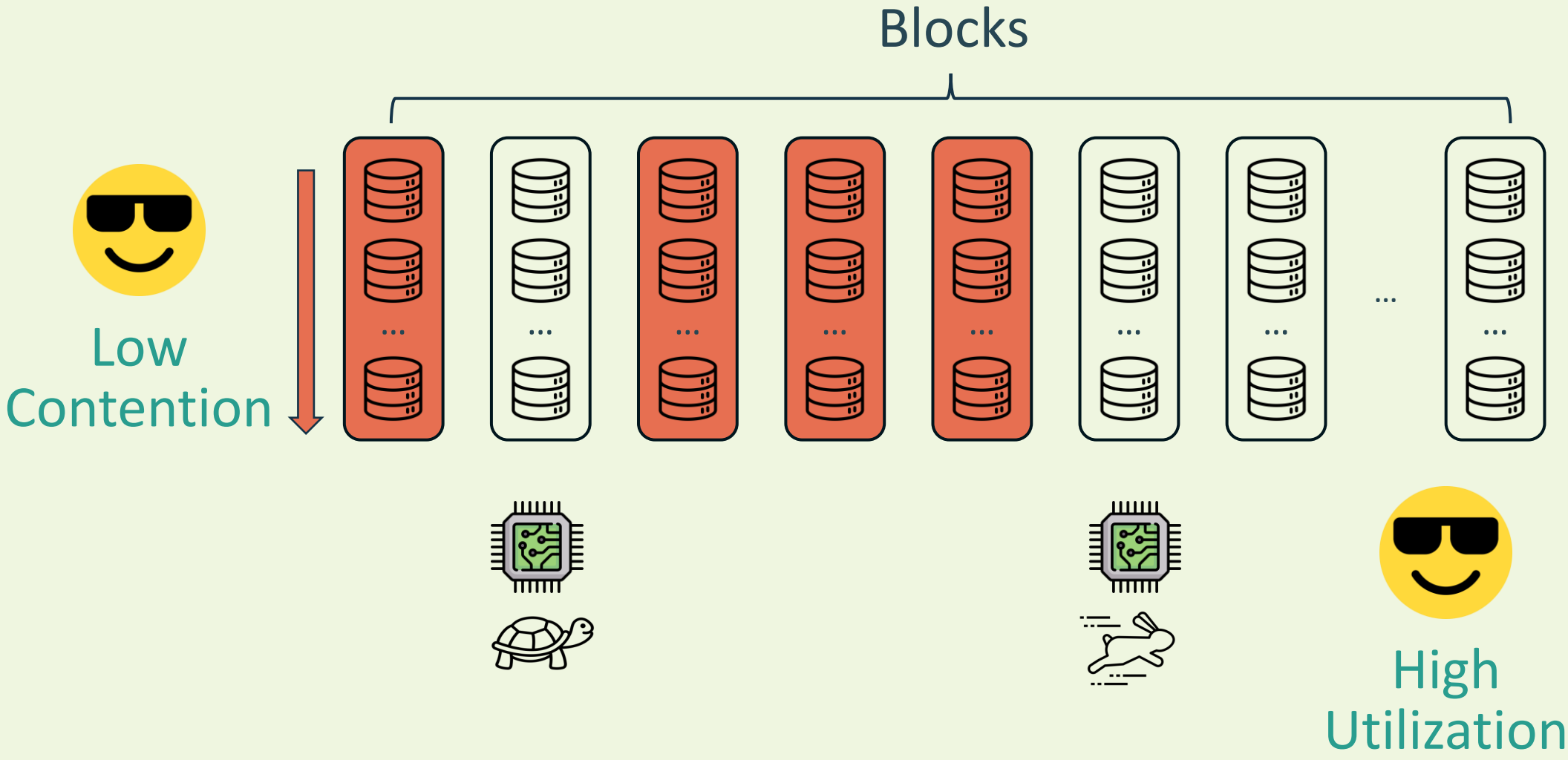
Improving Utilization via Block Partitioning



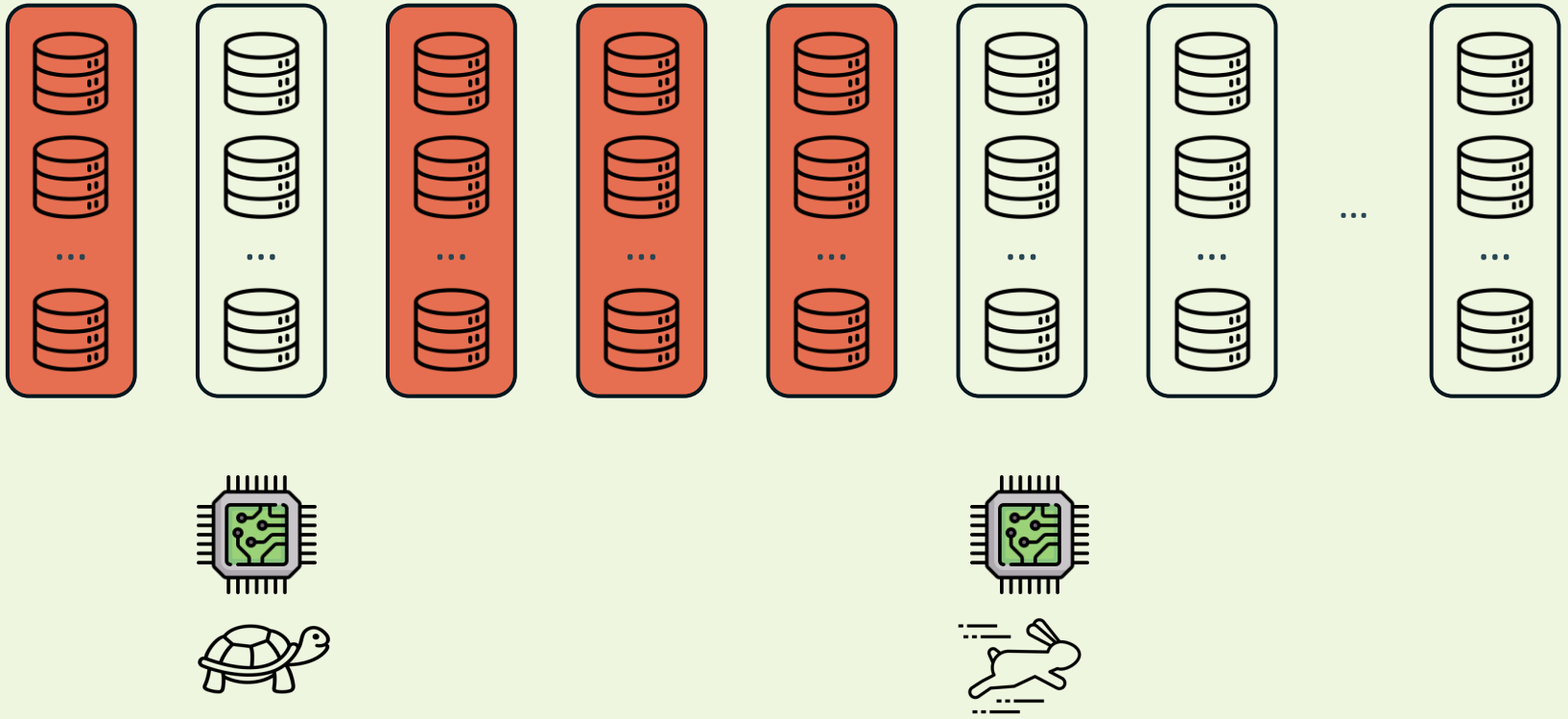
Improving Utilization via Block Partitioning



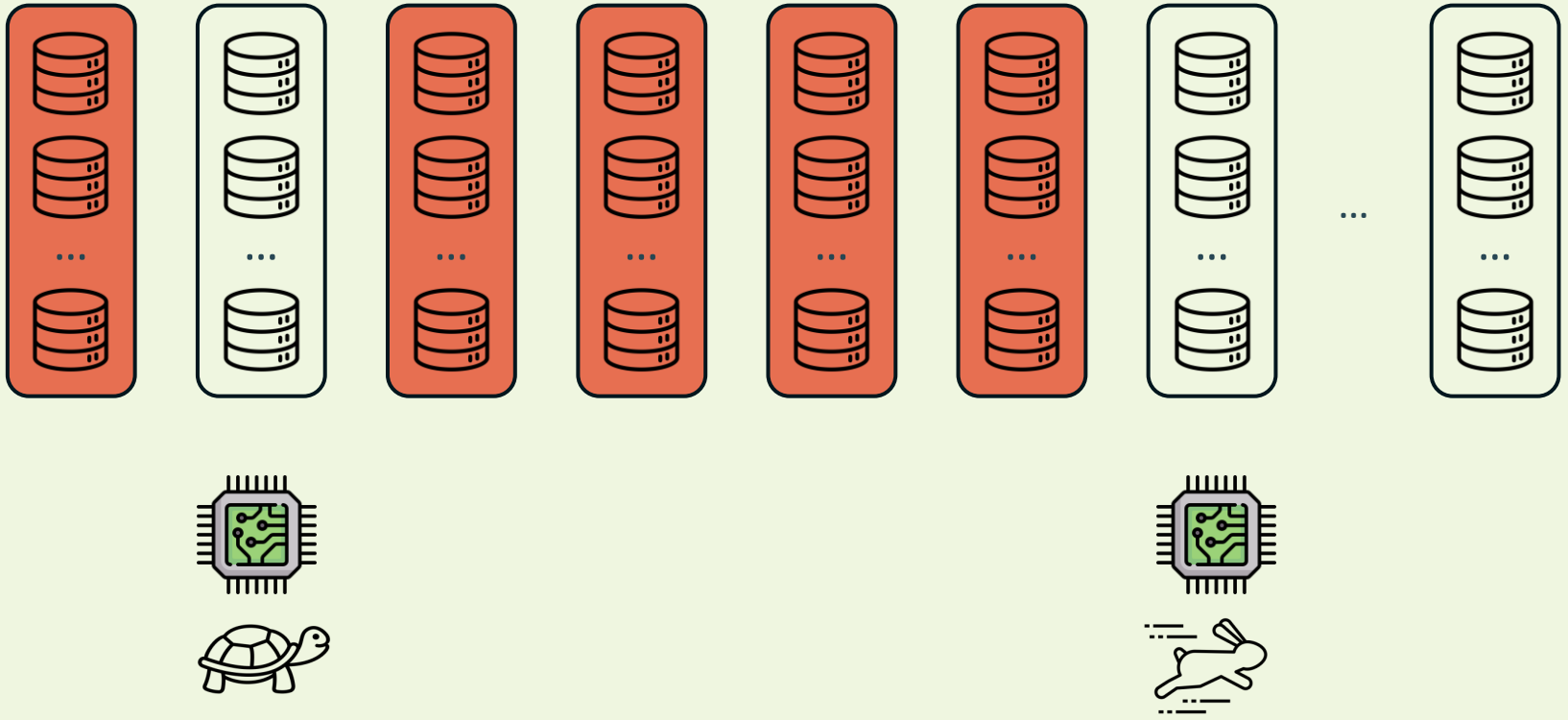
Improving Utilization via Block Partitioning



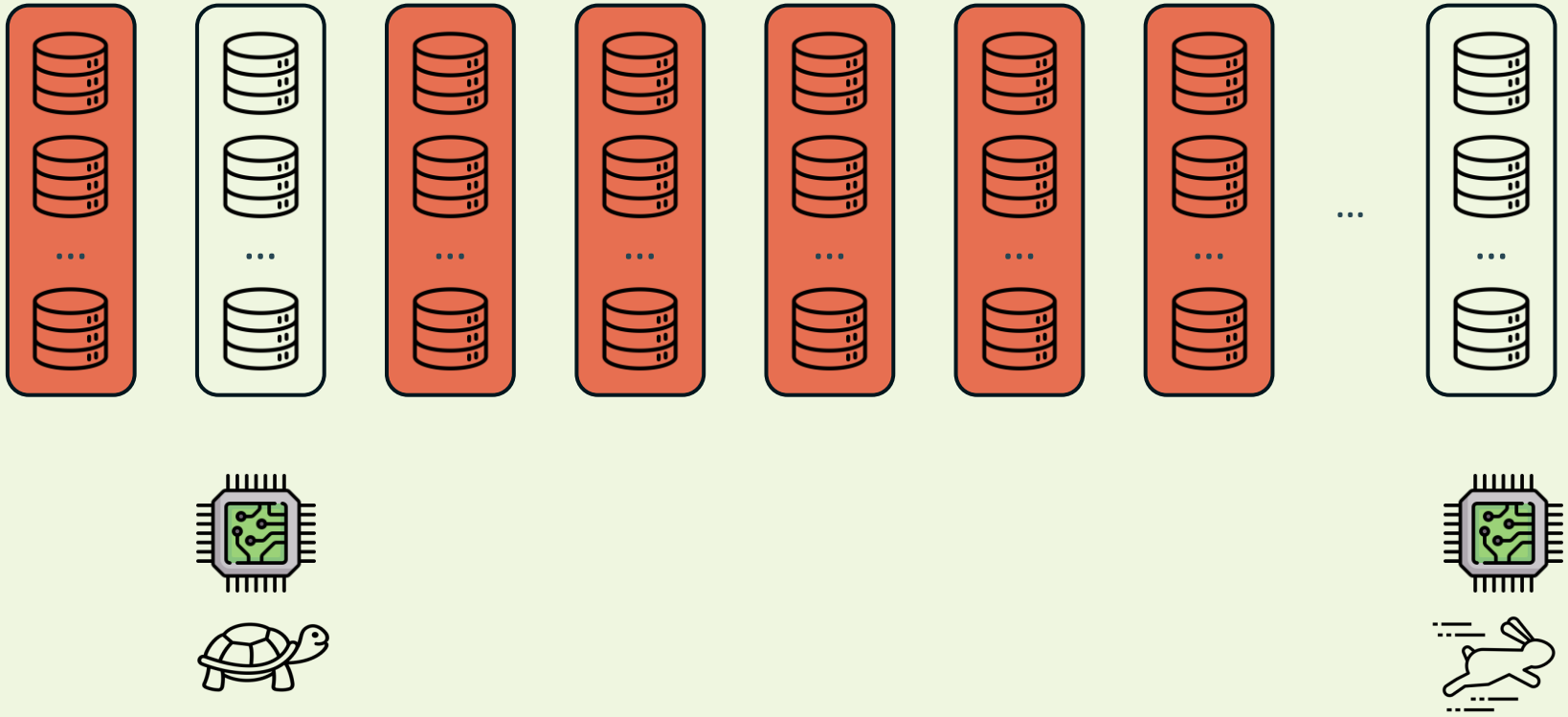
Enhance Effectivity via Block Closing



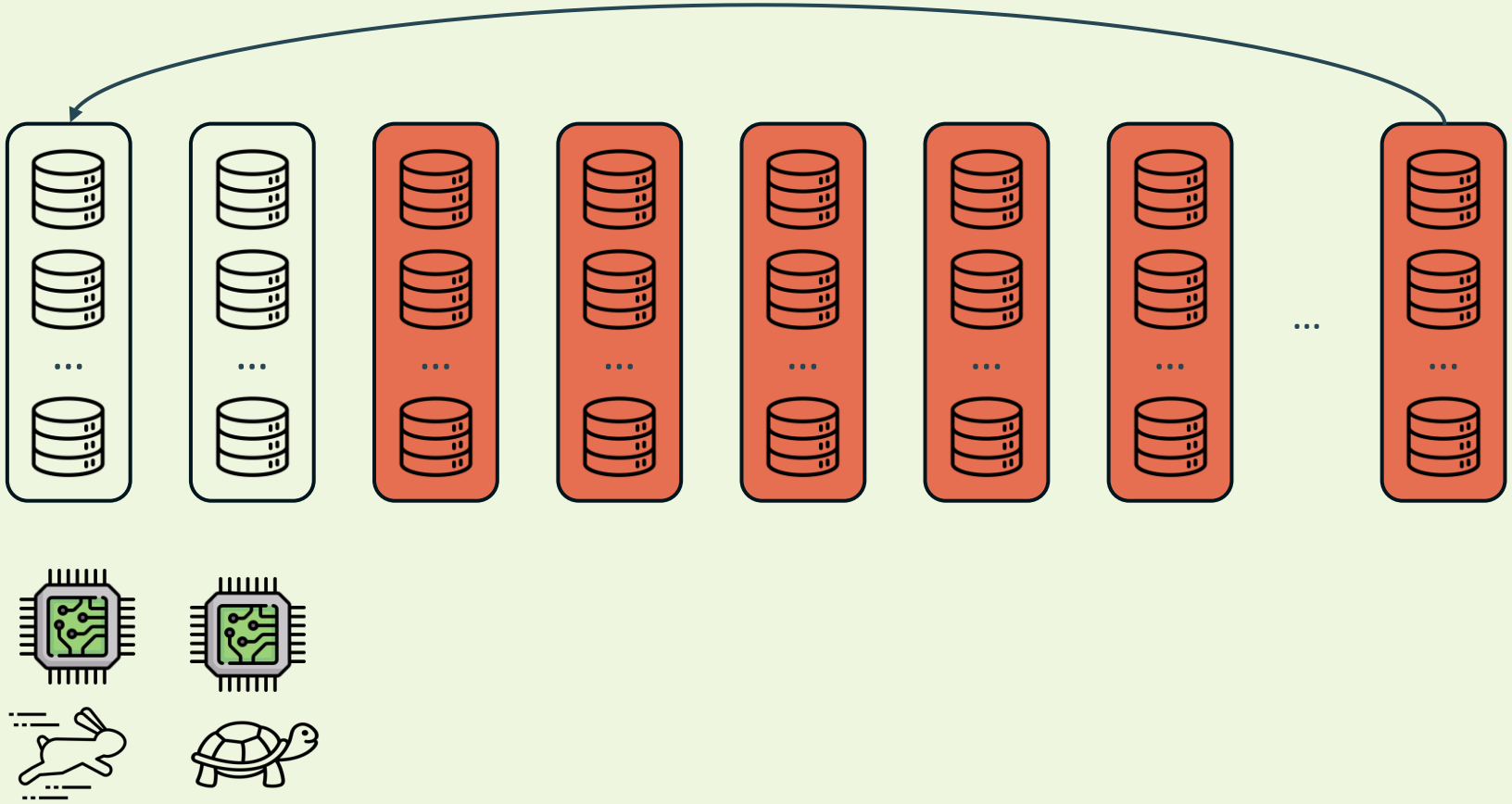
Enhance Effectivity via Block Closing



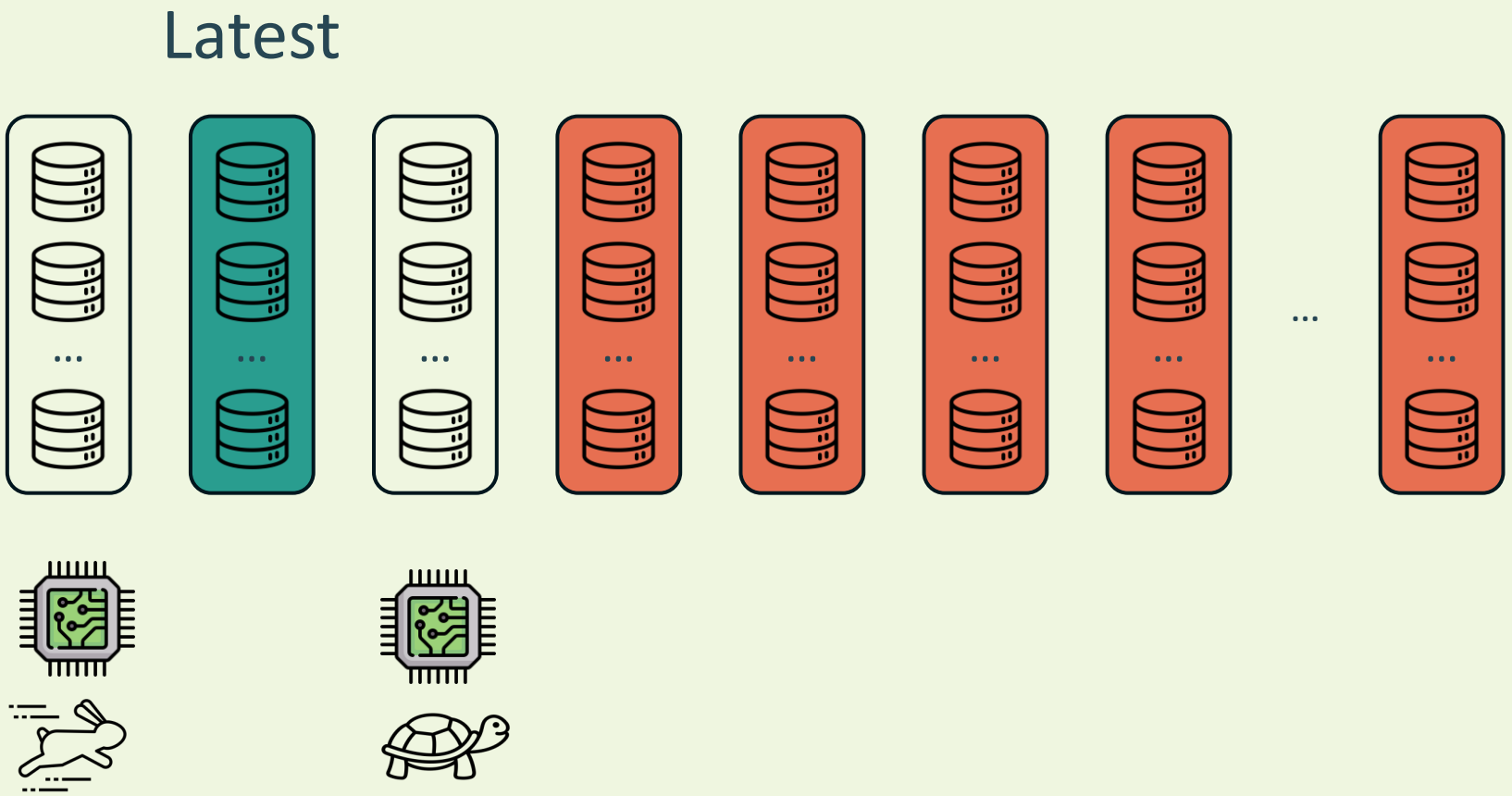
Enhance Effectivity via Block Closing



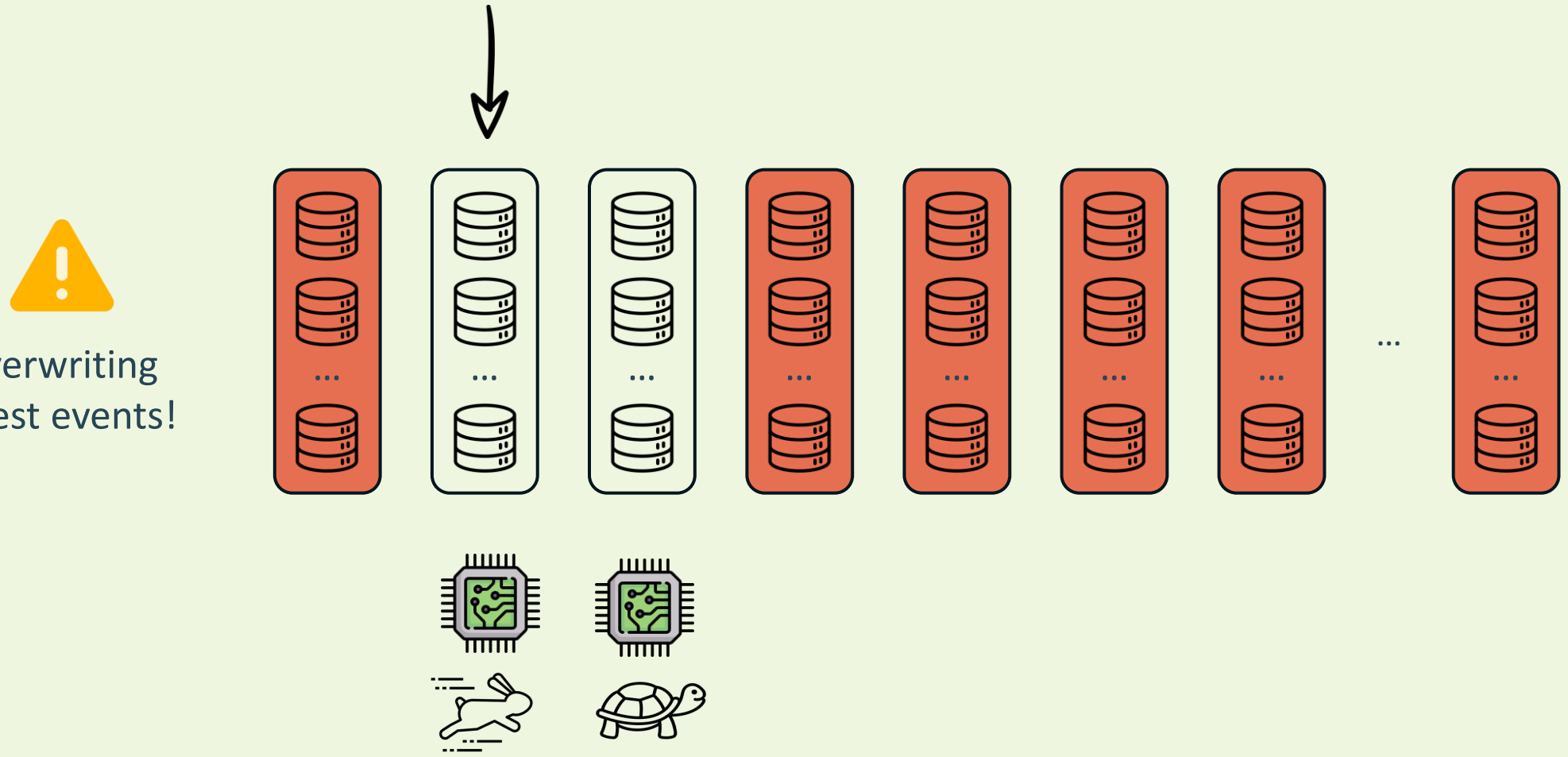
Enhance Effectivity via Block Closing



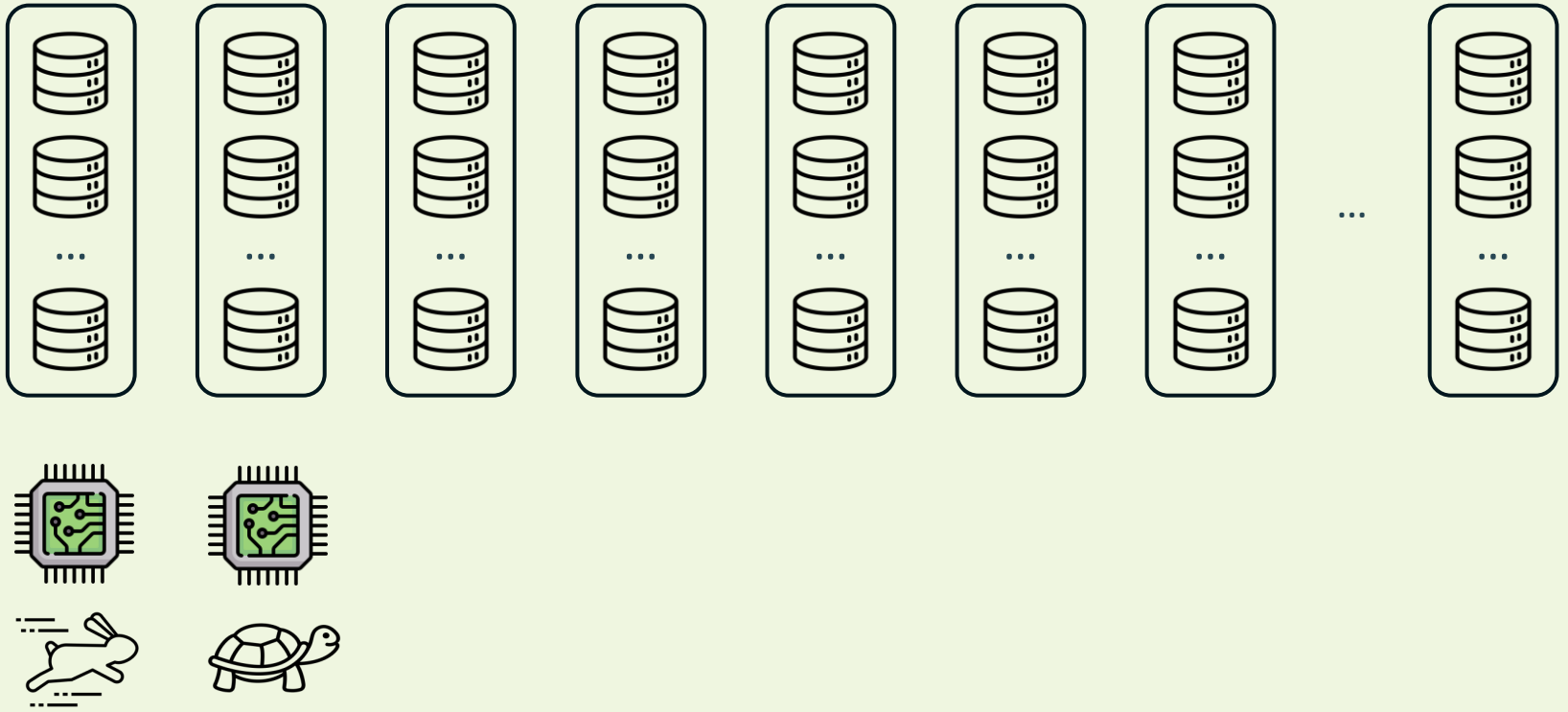
Enhance Effectivity via Block Closing



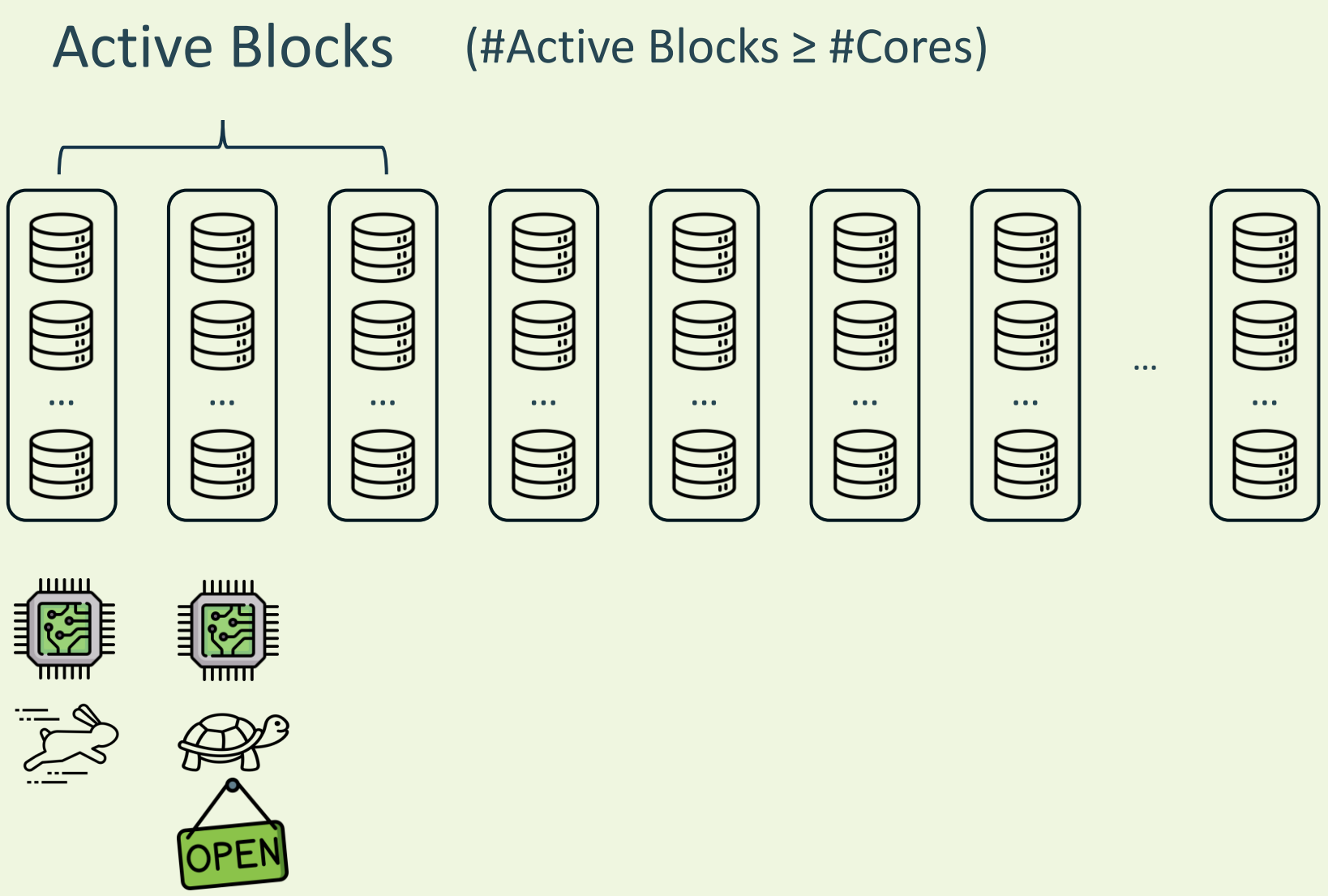
Enhance Effectivity via Block Closing



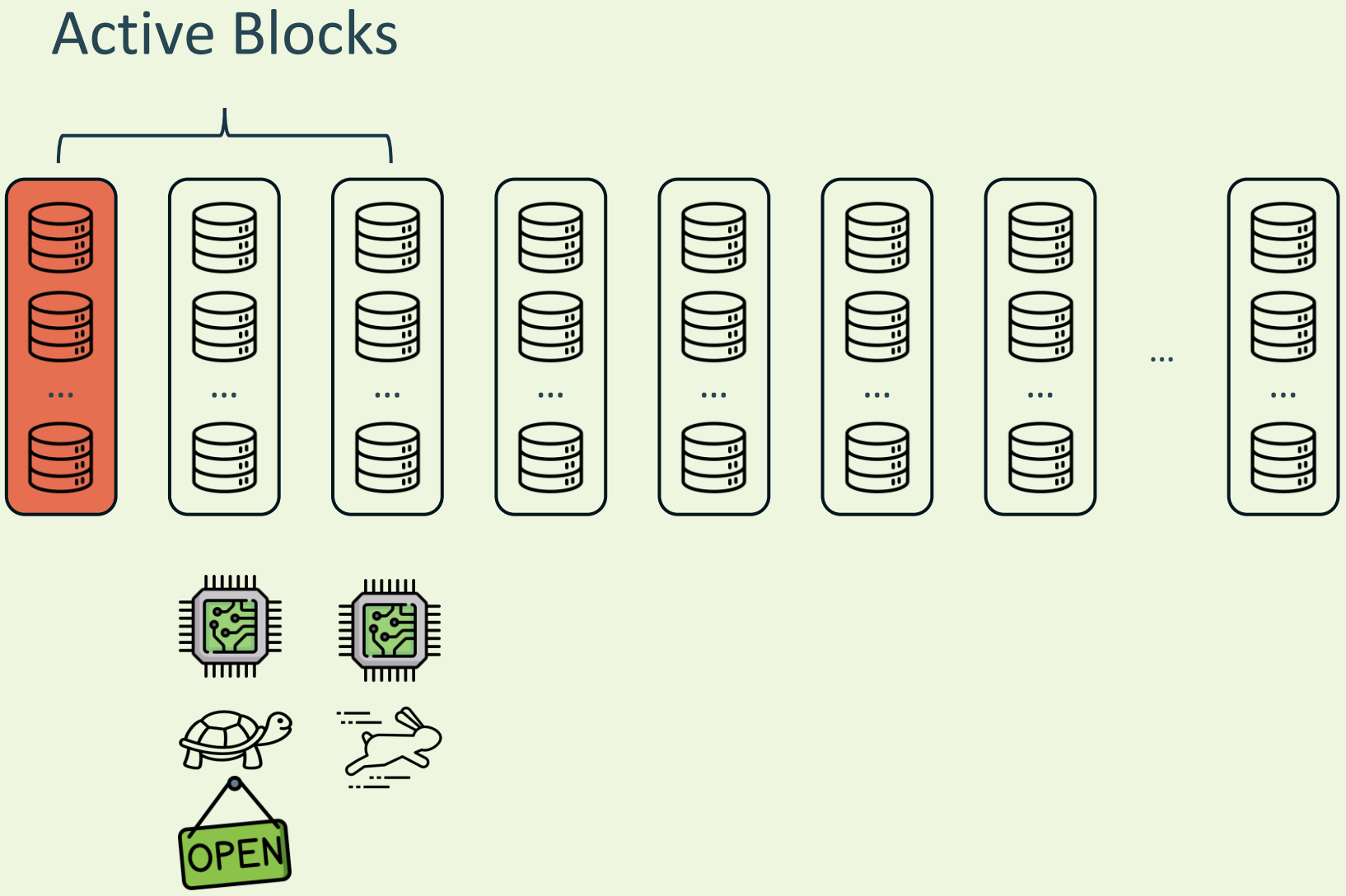
Enhance Effectivity via Block Closing



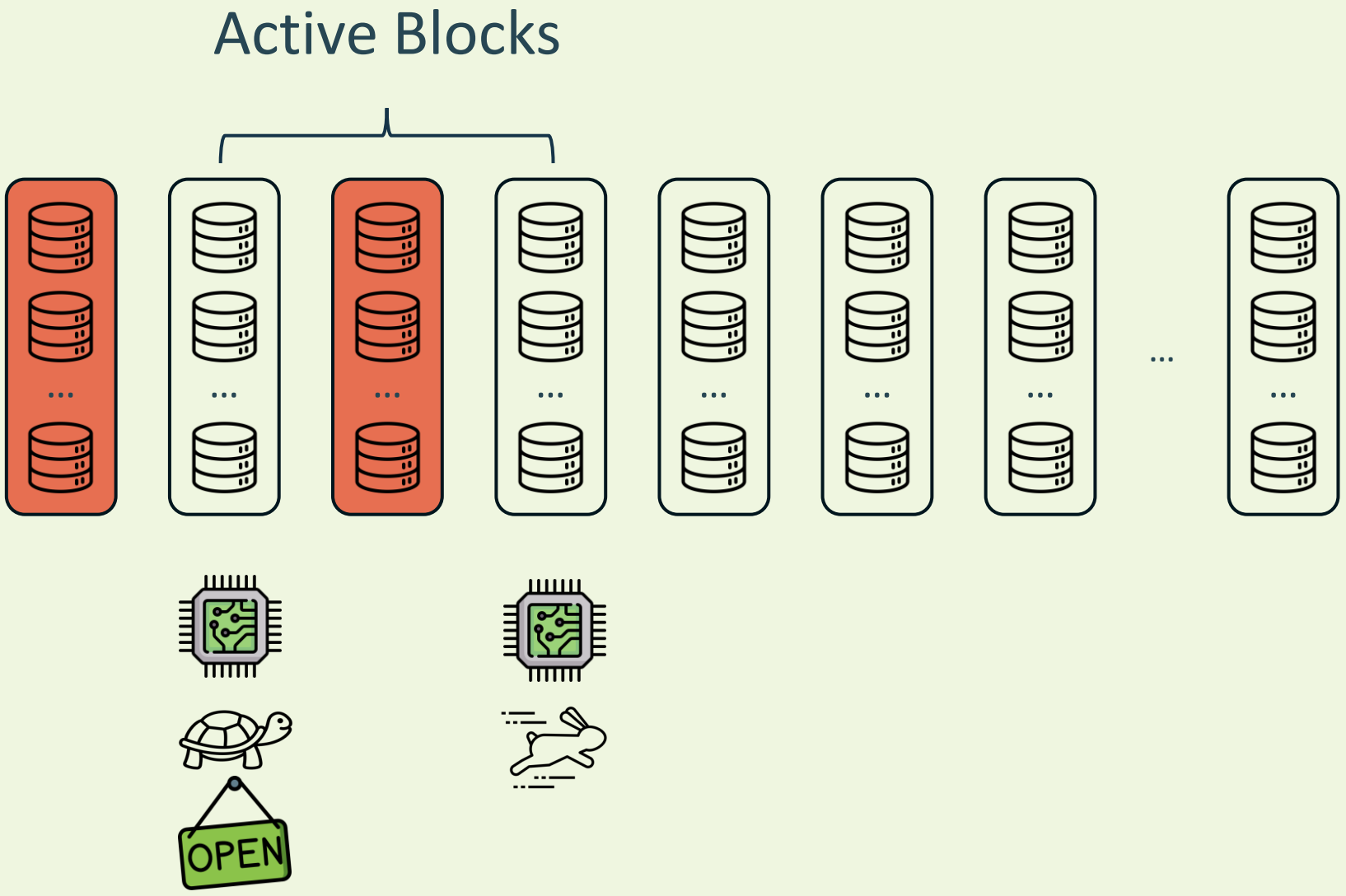
Enhance Effectivity via Block Closing



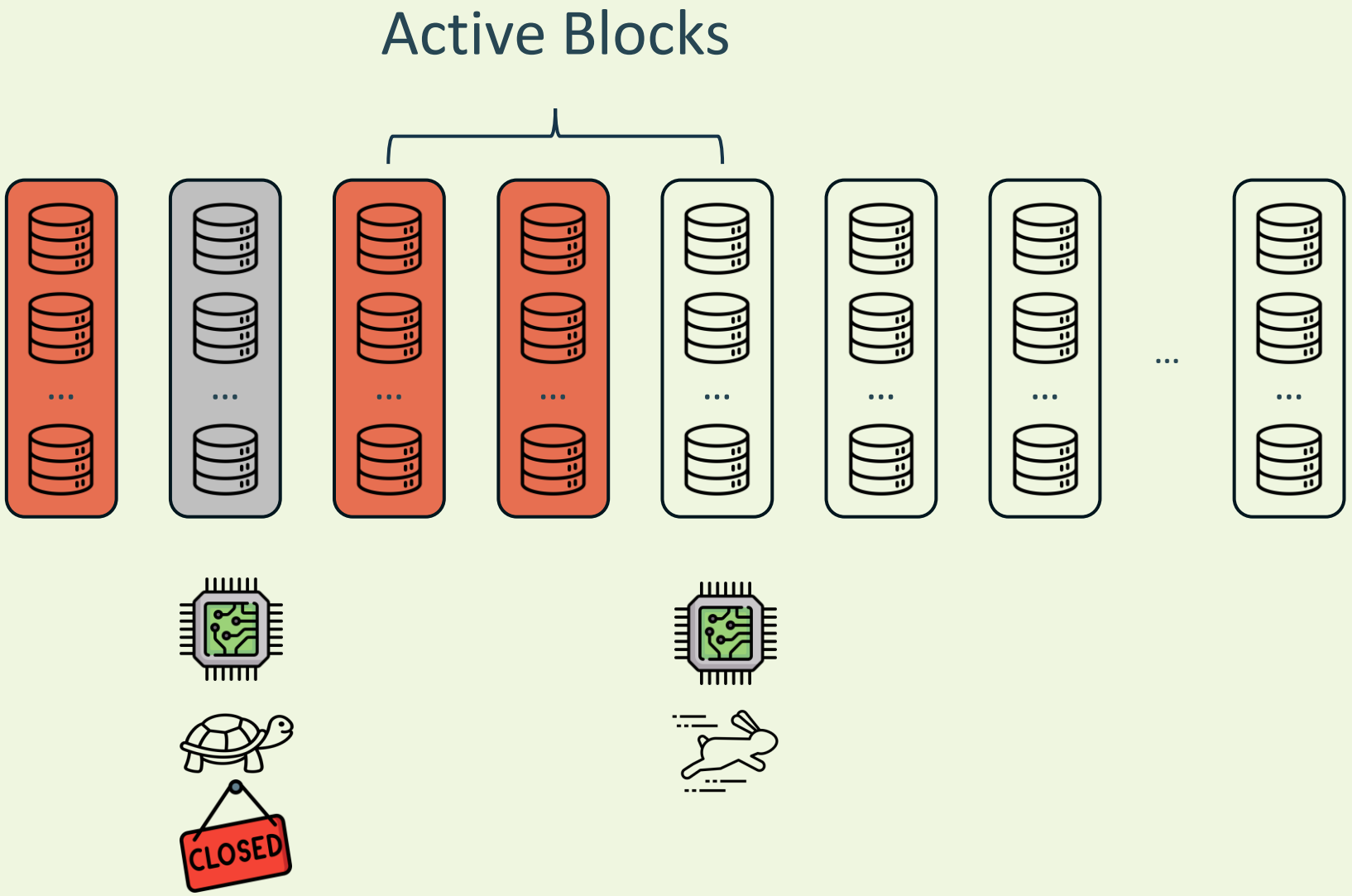
Enhance Effectivity via Block Closing



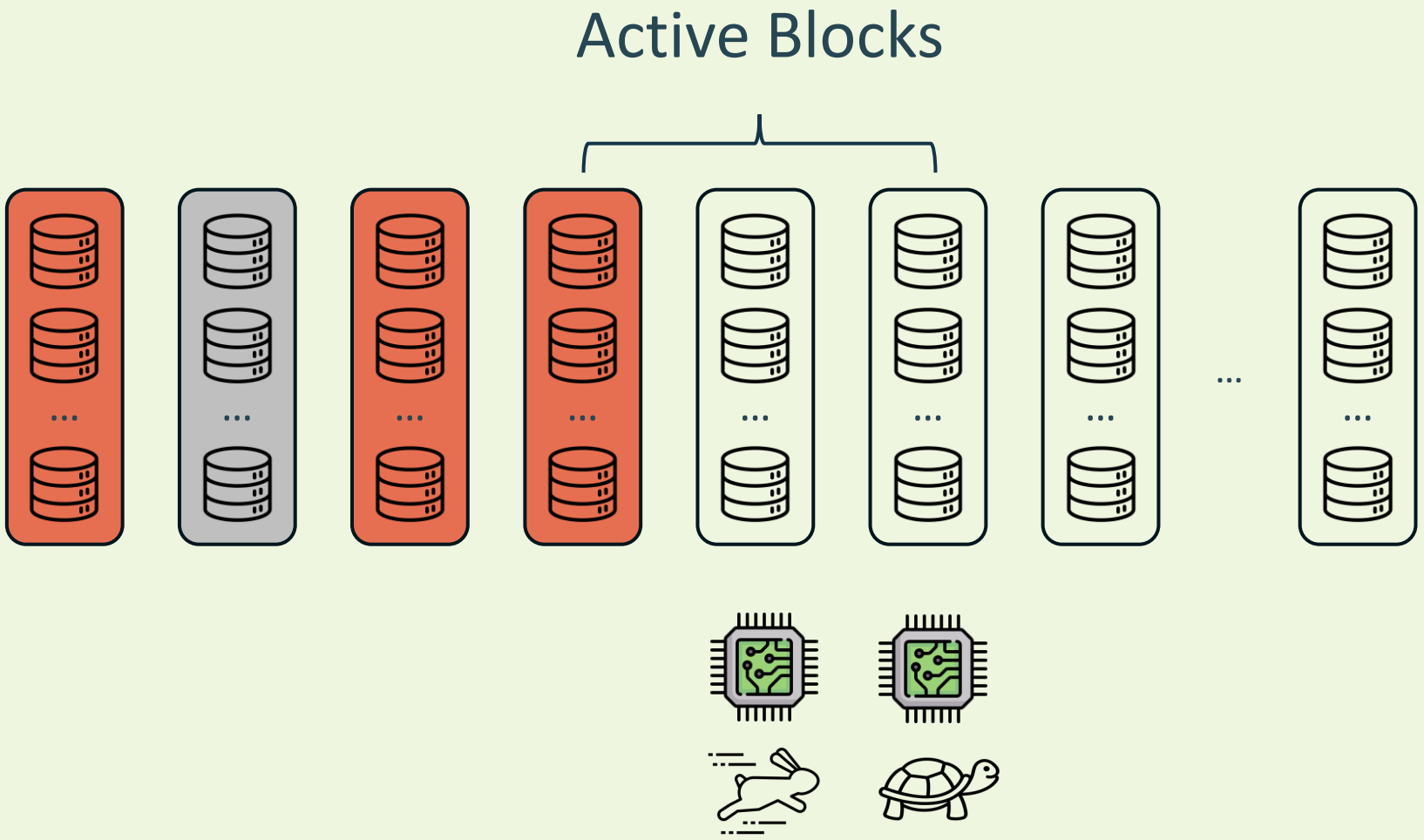
Enhance Effectivity via Block Closing



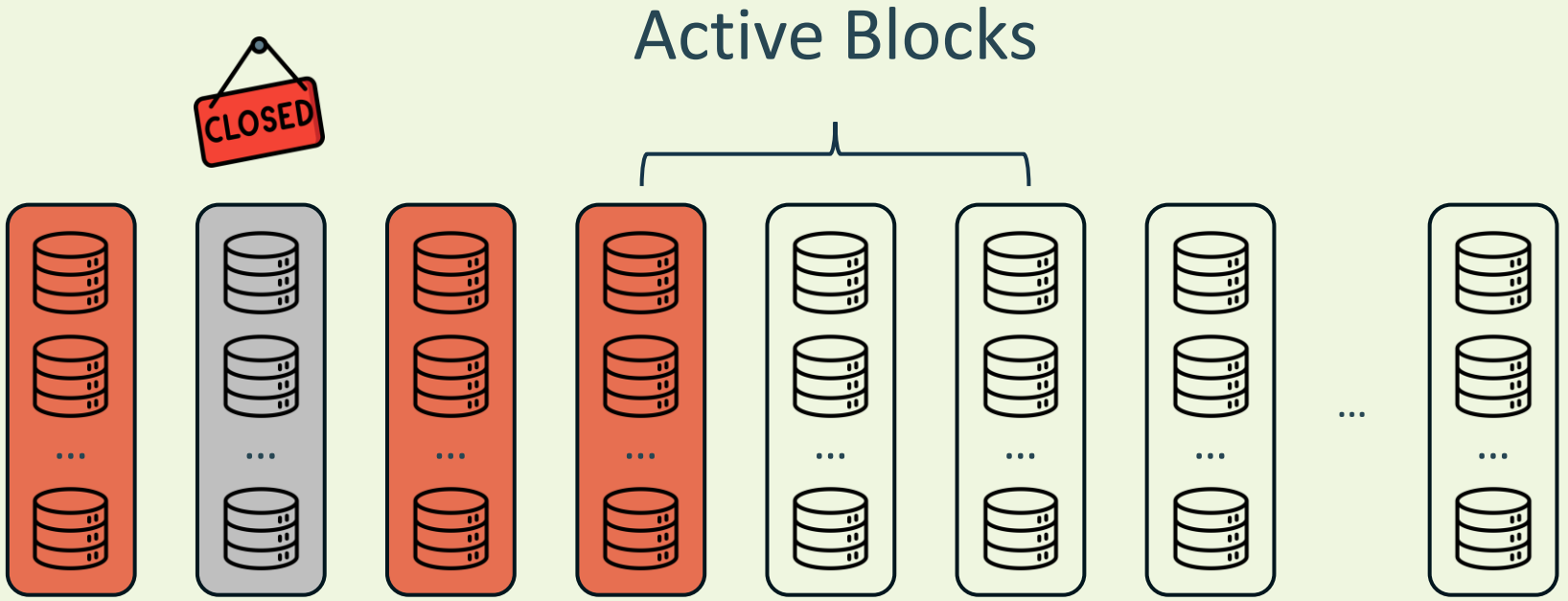
Enhance Effectivity via Block Closing



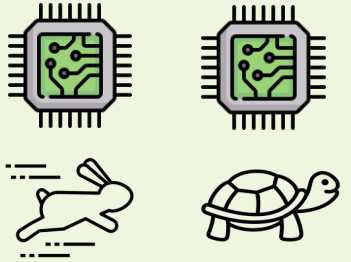
Enhance Effectivity via Block Closing



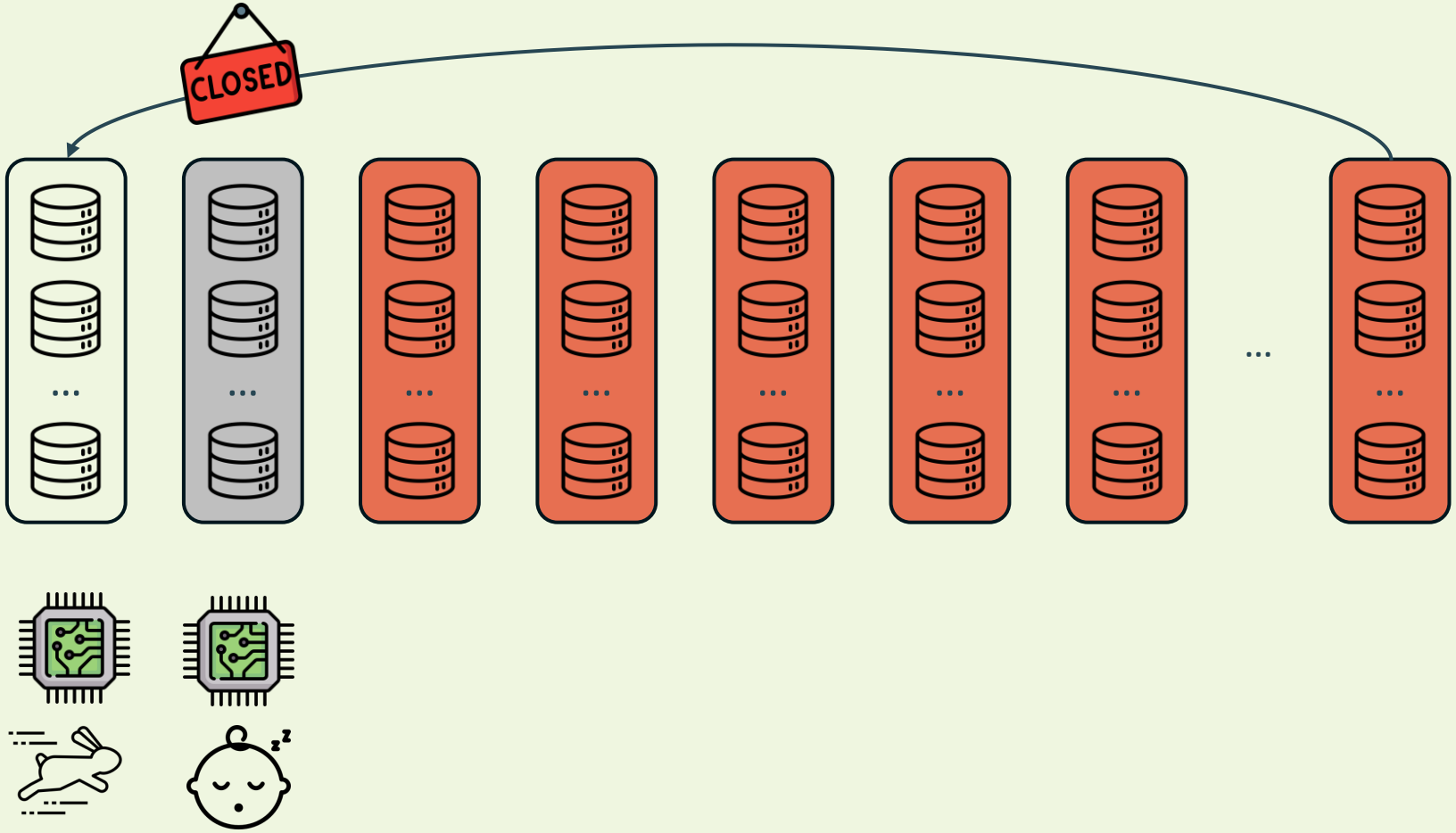
Enhance Effectivity via Block Closing



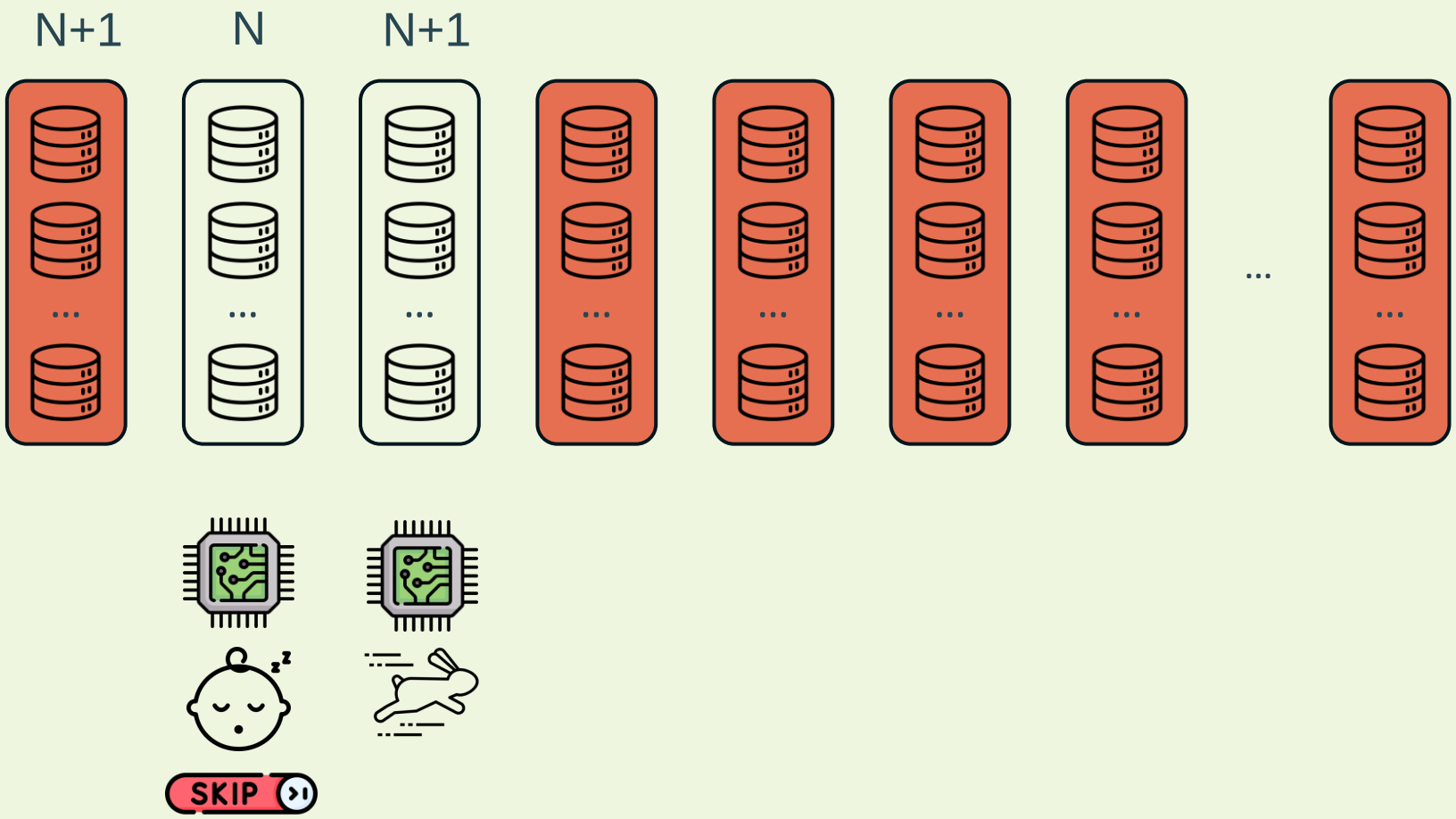
High
Effectivity Ratio



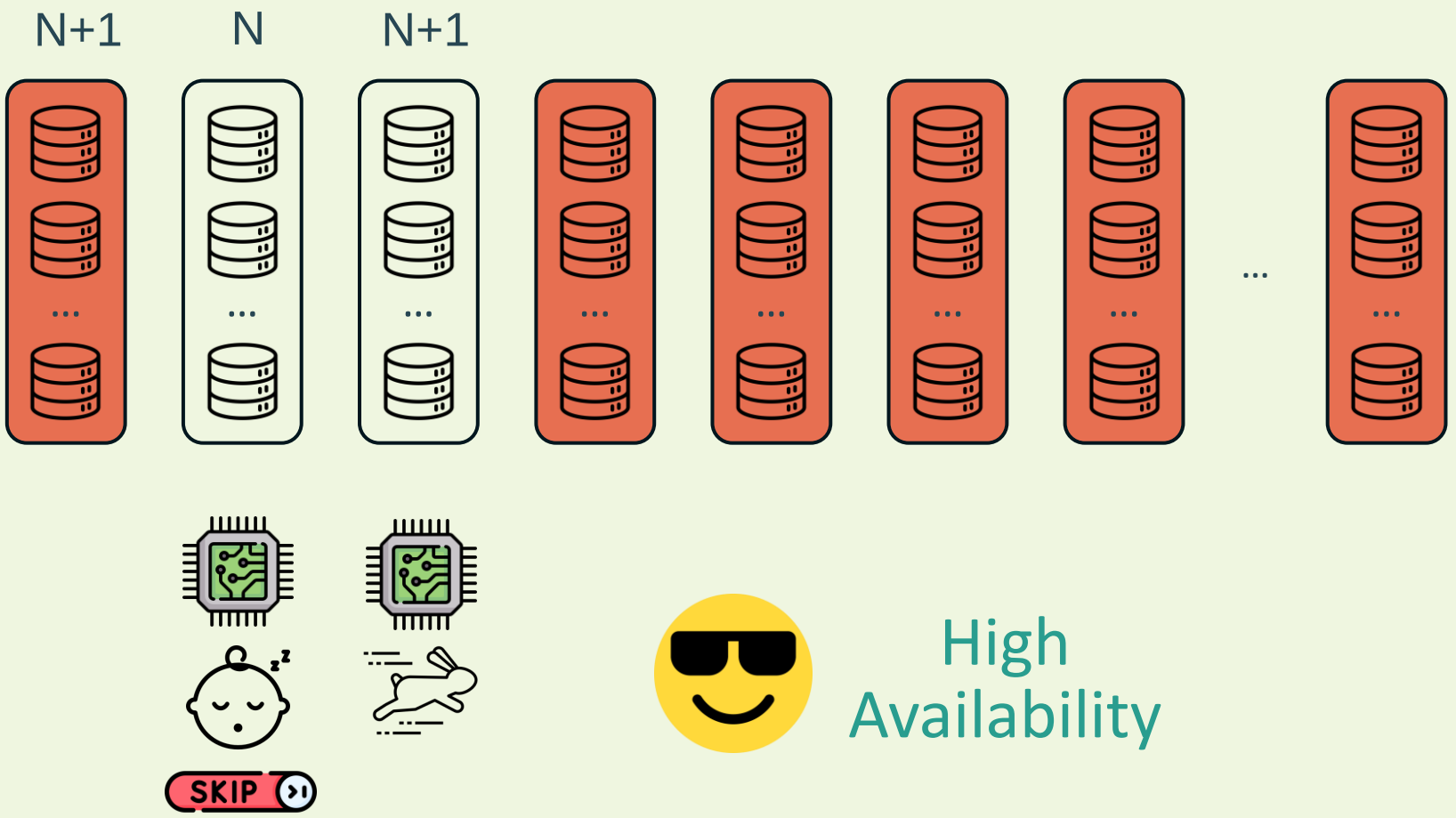
Ensure Availability via Block Skipping



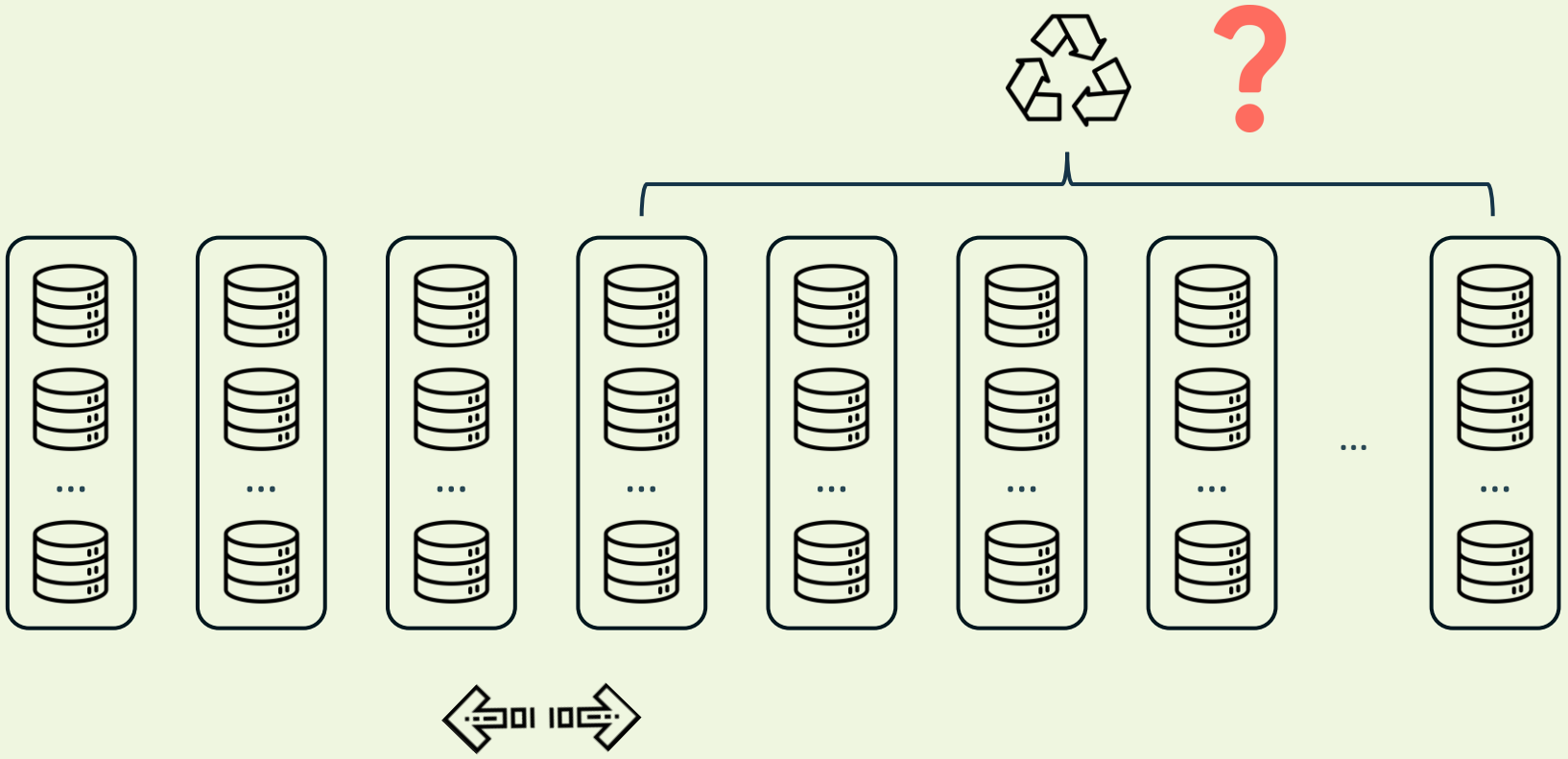
Ensure Availability via Block Skipping



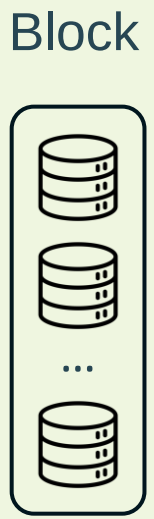
Ensure Availability via Block Skipping



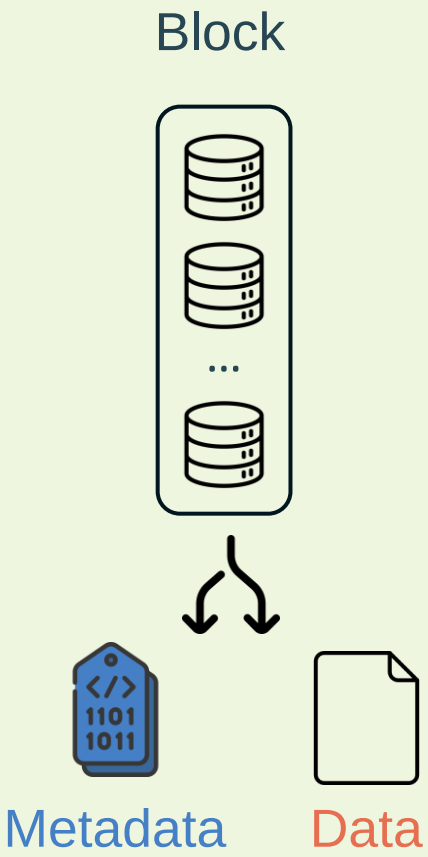
Enable Resizing via Implicit Reclaiming



Enable Resizing via Implicit Reclaiming



Enable Resizing via Implicit Reclaiming



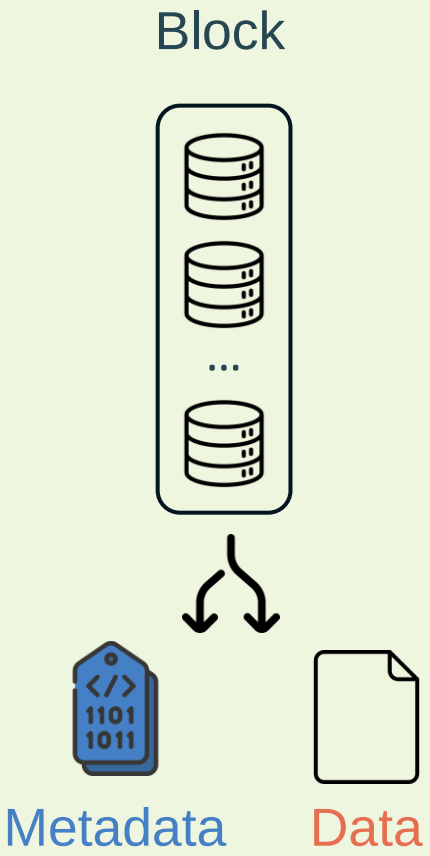
Enable Resizing via Implicit Reclaiming

trace_write:

①

②

③



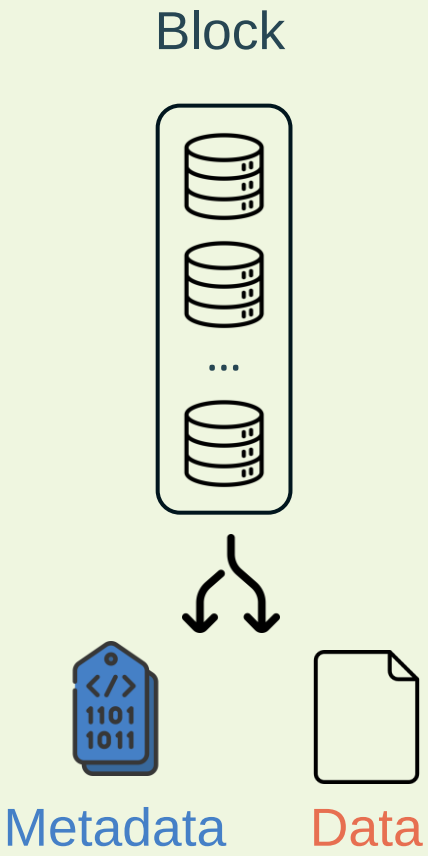
Enable Resizing via Implicit Reclaiming

trace_write:

① Allocating the space; Metadata

②

③



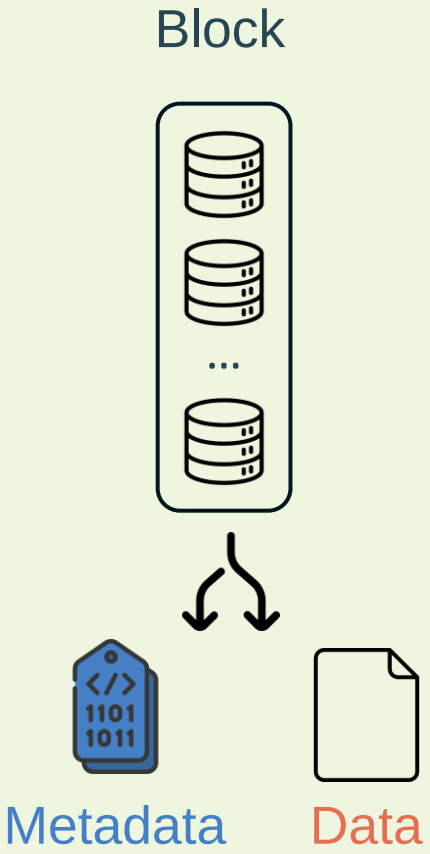
Enable Resizing via Implicit Reclaiming

trace_write:

① Allocating the space; Metadata

② Writing the event; Data

③



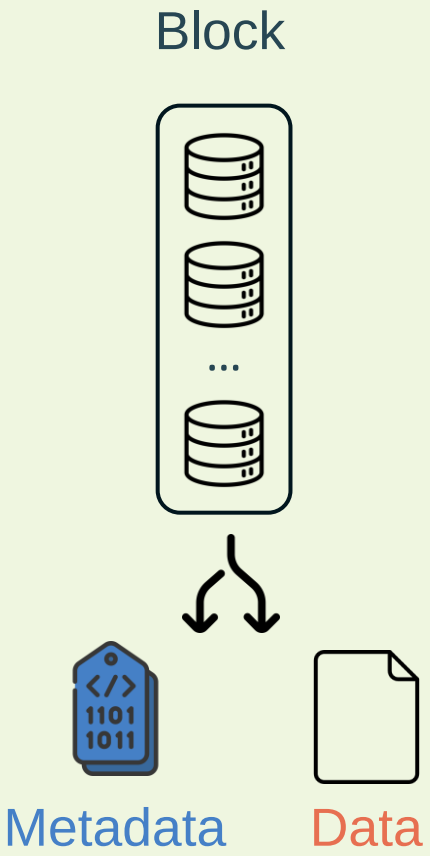
Enable Resizing via Implicit Reclaiming

trace_write:

① Allocating the space; Metadata

② Writing the event; Data

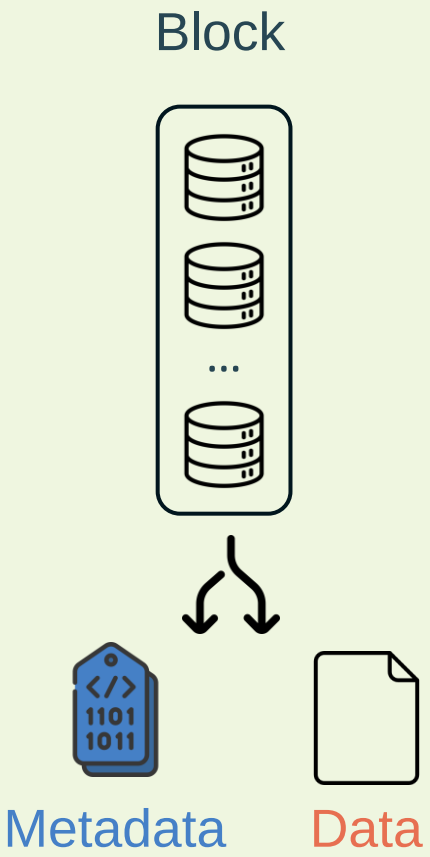
③ Confirming the completion; Metadata



Enable Resizing via Implicit Reclaiming

trace_write:

		Reference Counting
① Allocating the space;	Metadata	① rc++;
② Writing the event;	Data	② R/W;
③ Confirming the completion;	Metadata	③ rc--;



Enable Resizing via Implicit Reclaiming

trace_write:

Reference Counting

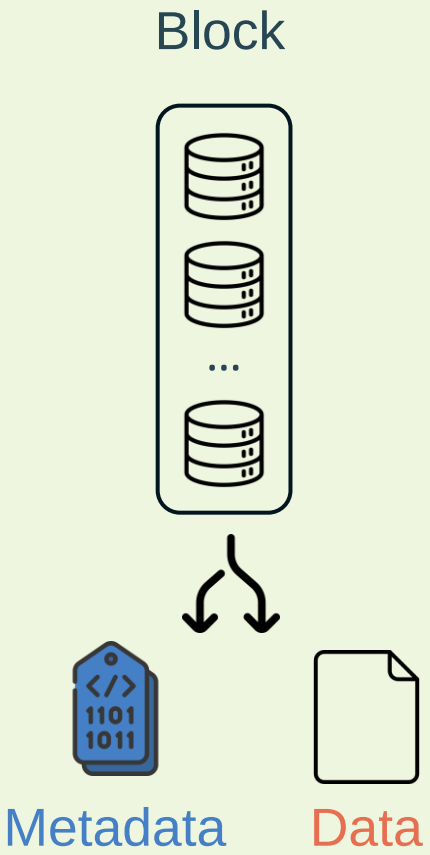
① Allocating the space; Metadata ① rc++;

② Writing the event; Data ② R/W;

③ Confirming the completion; Metadata ③ rc--;

Metadata as reference counter:

allocated_space == confirmed_space ⟷ rc == 0



Enable Resizing via Implicit Reclaiming

trace_write:

① Allocating the space; Metadata

② Writing the event; Data

③ Confirming the completion; Metadata

Reference Counting

① rc++;

② R/W;

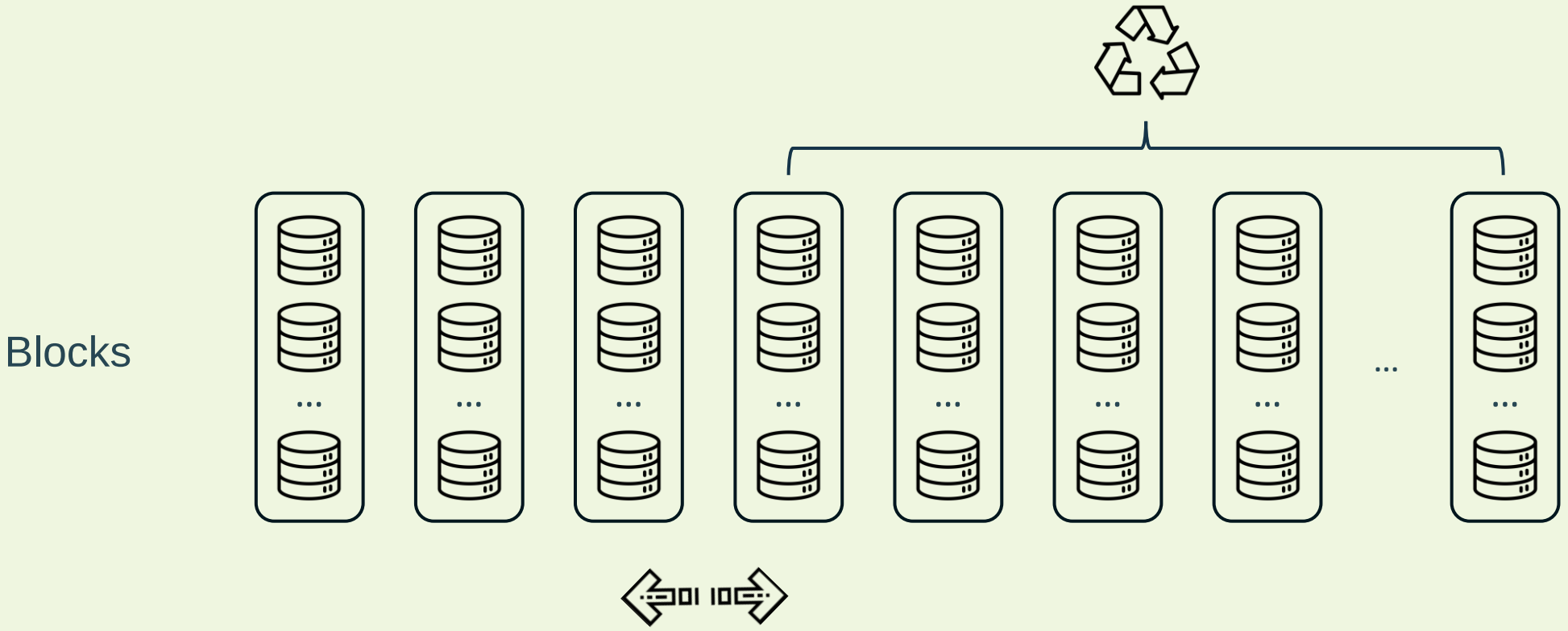
③ rc--;

<u>Metadata</u> as reference counter:

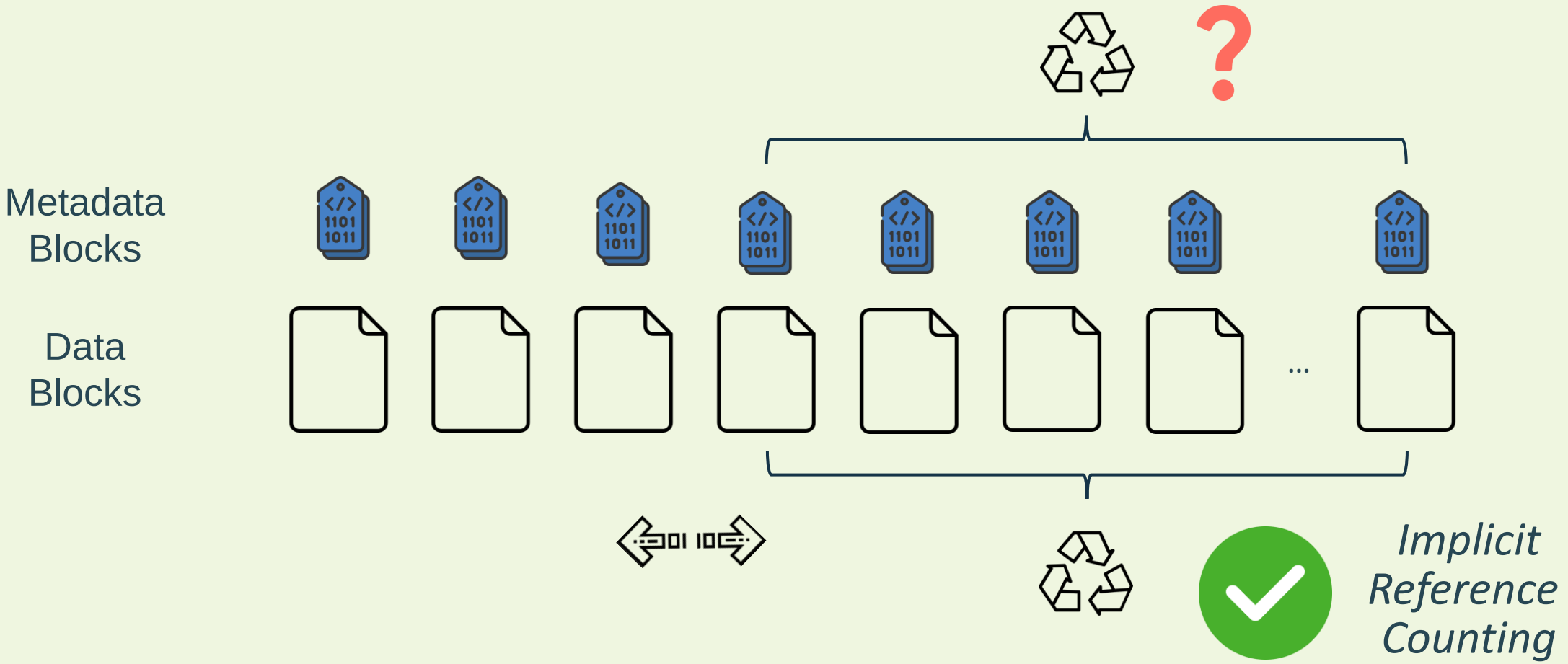
allocated_space == confirmed_space ⟷ rc == 0

Safely reclaim the <u>data block</u>

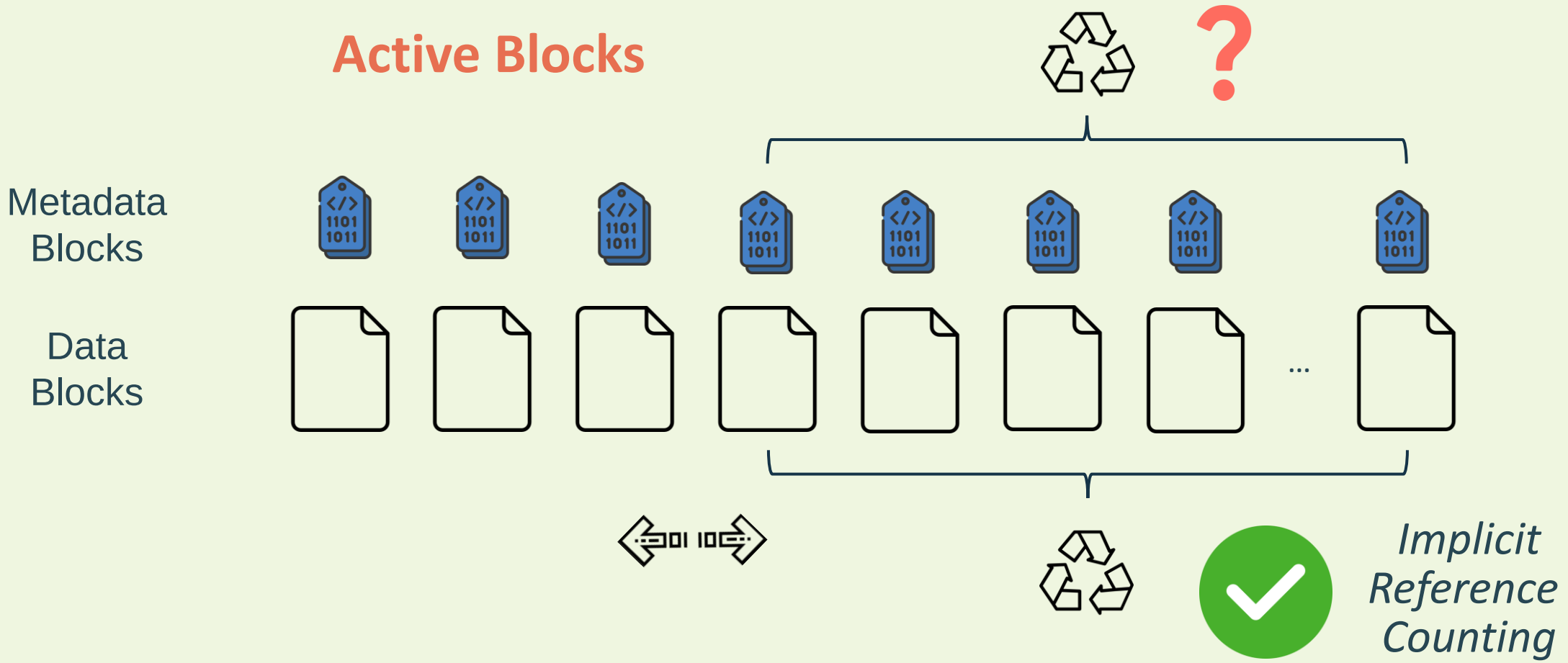
Enable Resizing via Implicit Reclaiming



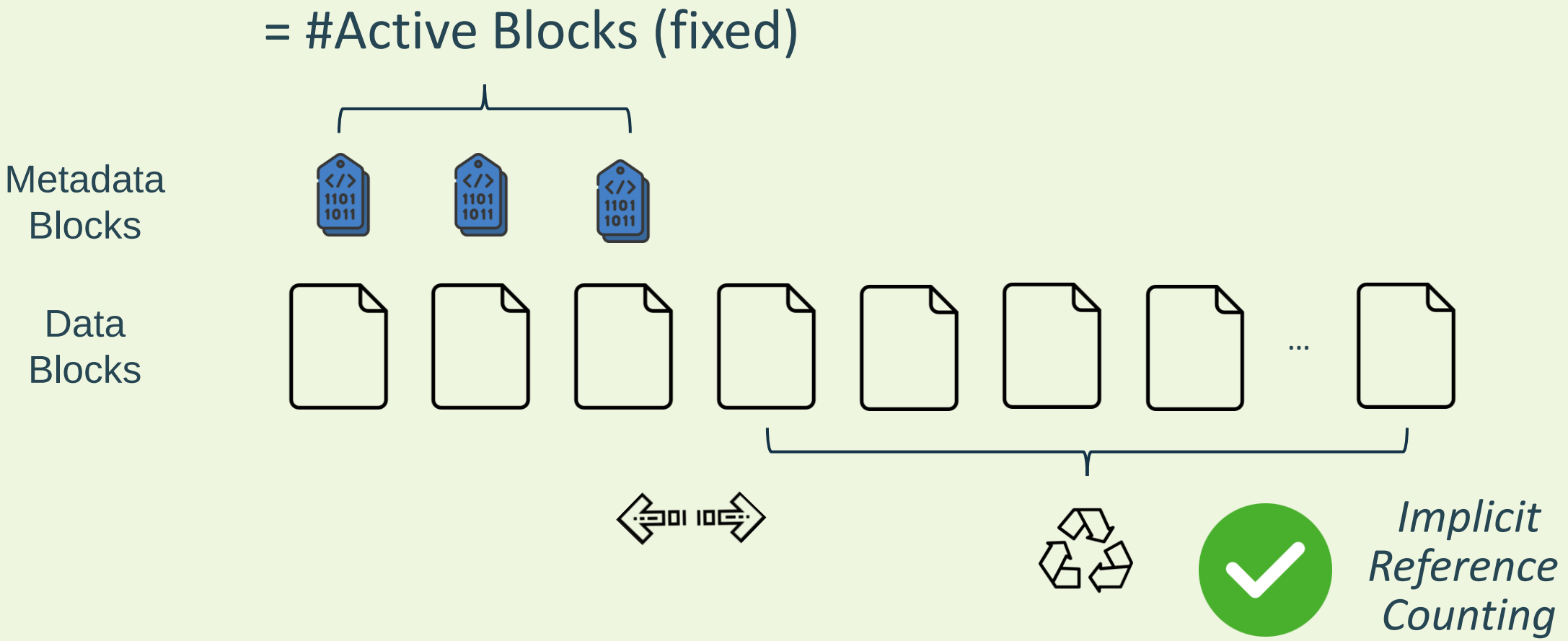
Enable Resizing via Implicit Reclaiming



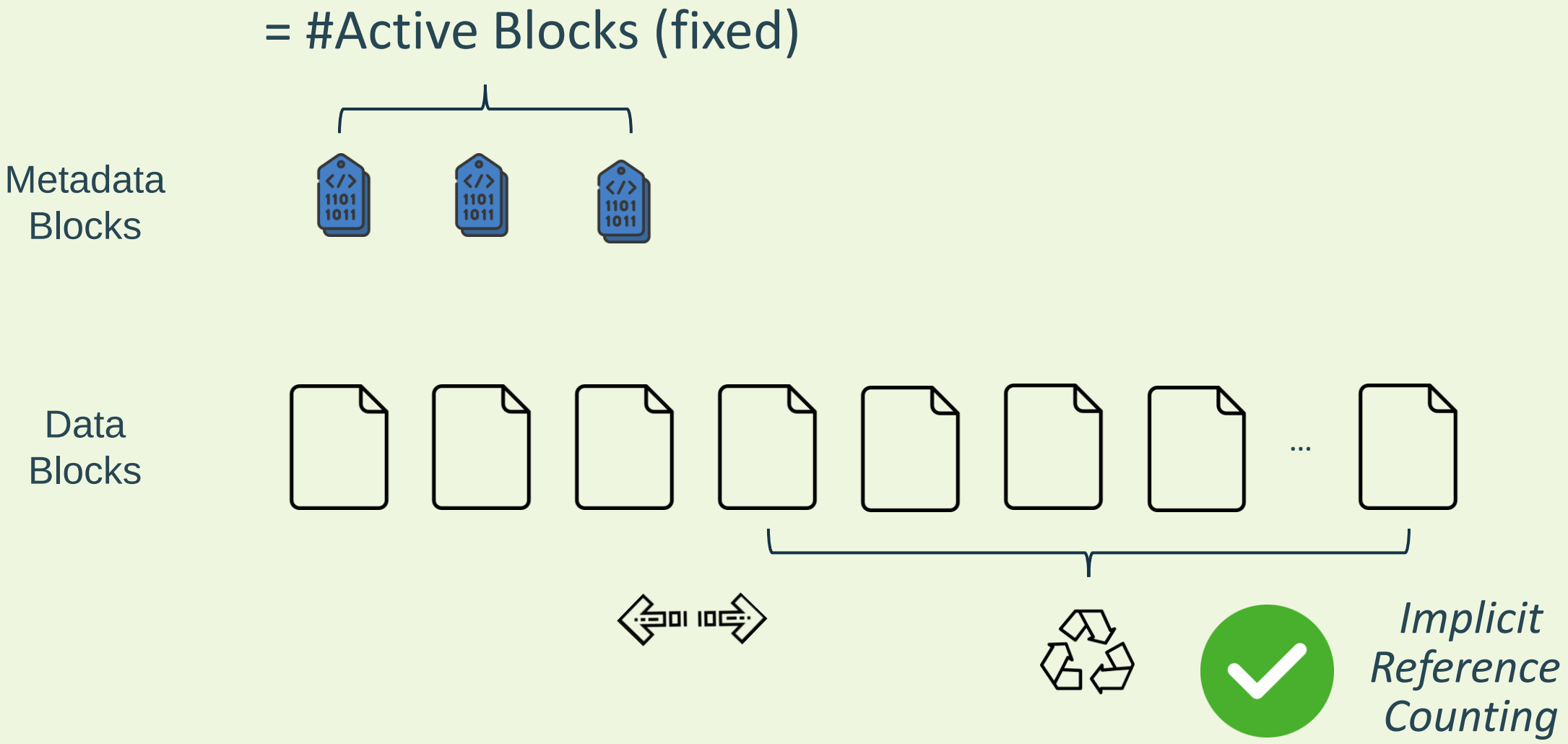
Enable Resizing via Implicit Reclaiming



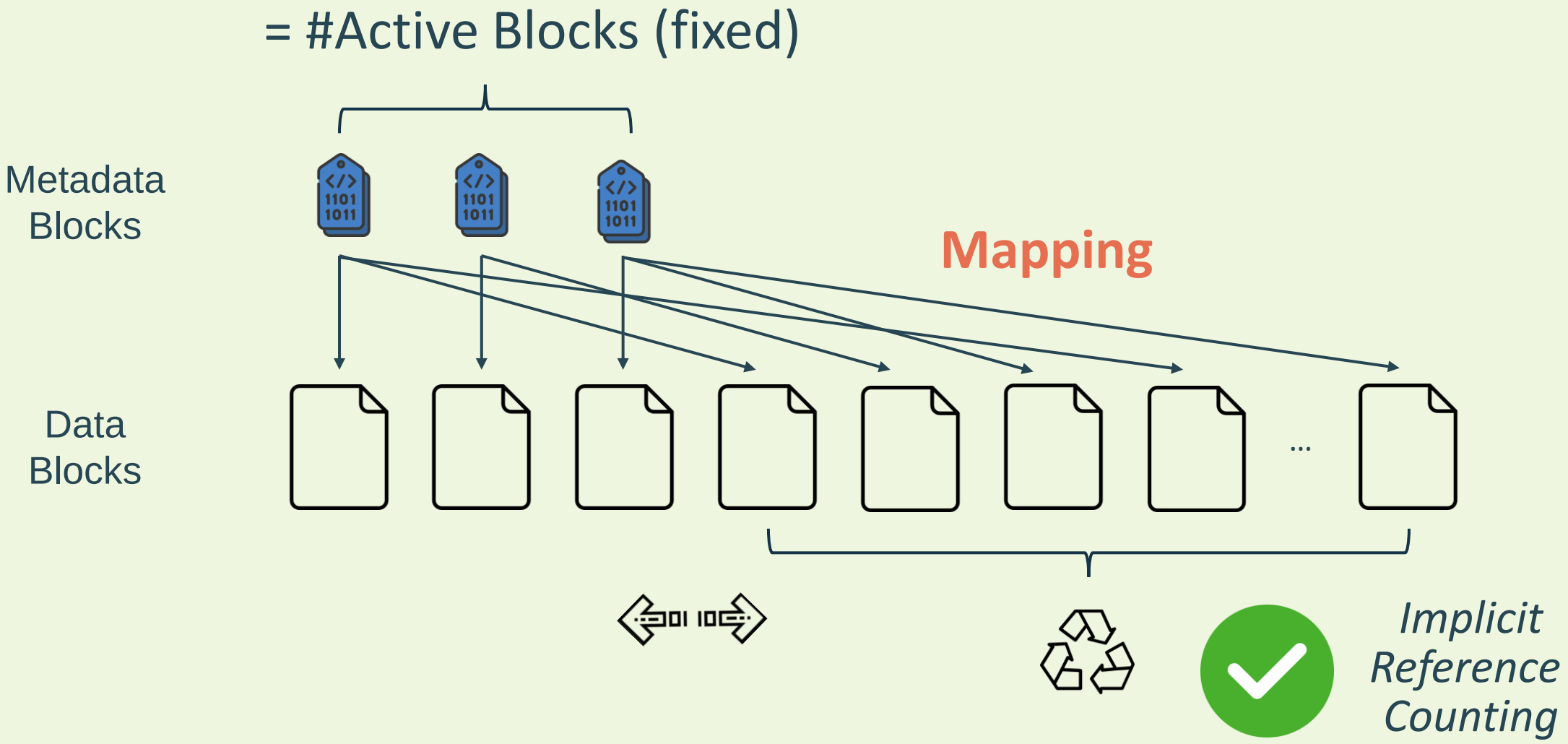
Enable Resizing via Implicit Reclaiming



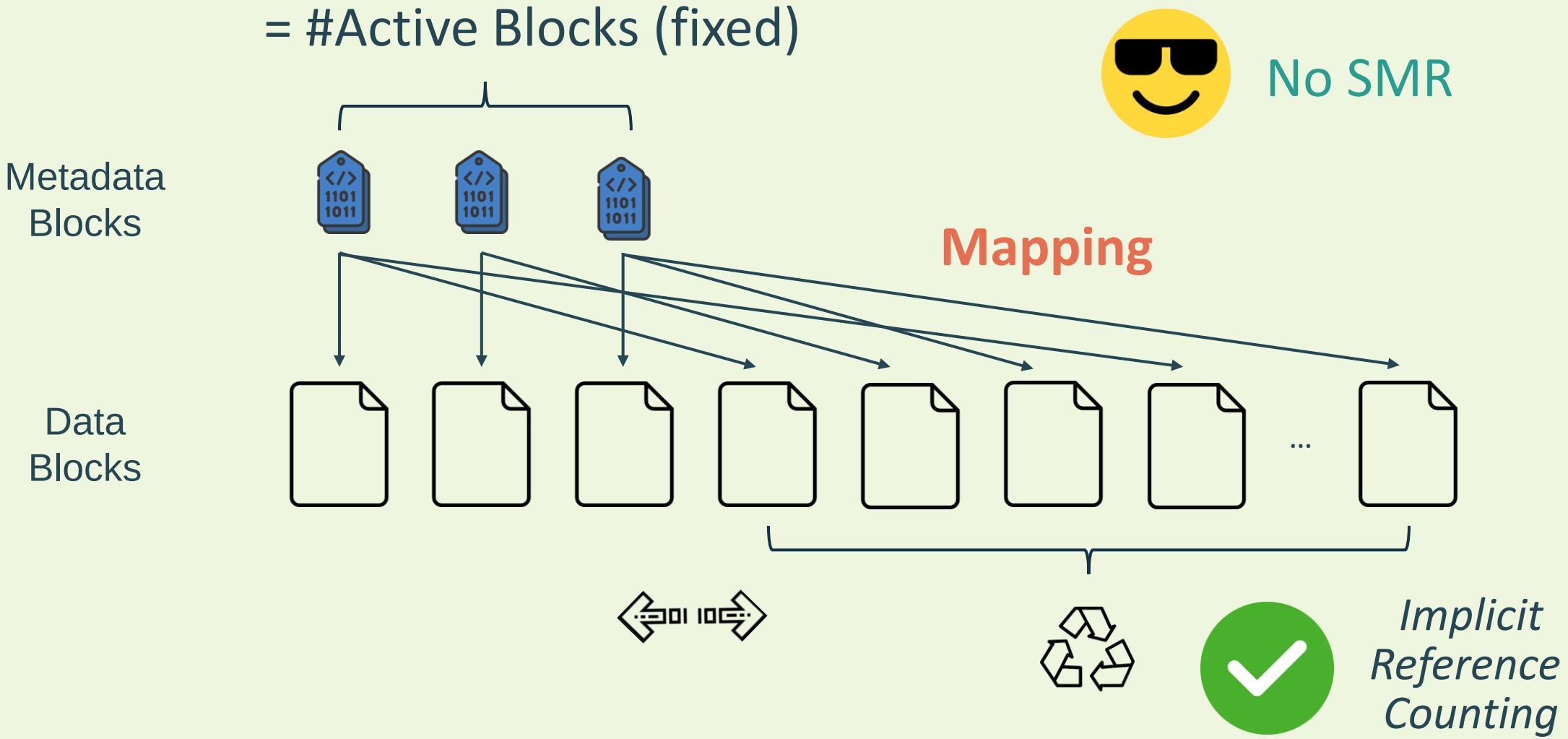
Enable Resizing via Implicit Reclaiming



Enable Resizing via Implicit Reclaiming



Enable Resizing via Implicit Reclaiming



Evaluation

Workloads (20 in total)

- The top 10 applications and games by number of downloads in the app store
- Performance testing software
- Typical usage scenarios (e.g., lock screen)

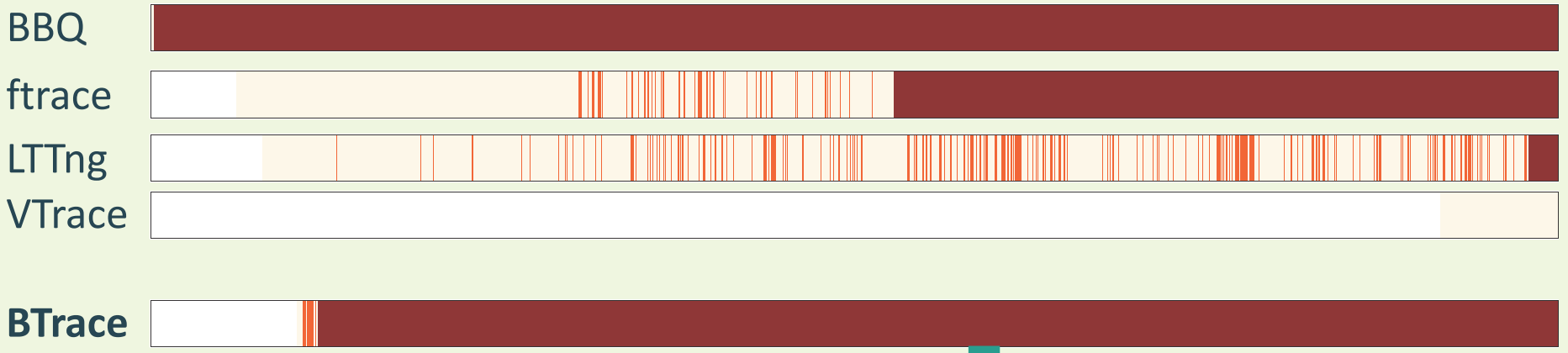
Evaluation

		Bench-1	Browser	Camera	Bench-2	Desktop	Install	Video-1	eShop-1	Bench-3	eShop-2	LockScr.	StartApp.	eShop-3	Video-2	News	Video-3	Game	IM	Blog-1	Blog-2	G.M.
Latest	BTrace	10.8	10.6	10.6	11.2	11.0	10.7	11.0	10.8	10.6	10.6	10.8	10.9	10.8	10.6	10.4	11.0	11.1	10.6	10.2	10.9	10.8
	BBQ	11.6	11.6	11.7	11.7	11.6	11.6	11.6	11.5	11.6	11.5	11.6	11.6	11.5	11.5	11.5	11.6	11.6	11.6	11.6	11.5	11.6
	ftrace	3.4	7.1	5.2	5.6	3.3	4.3	3.3	6.8	4.1	5.8	3.8	3.2	7.6	7.2	6.5	4.9	7.5	7.5	7.6	8.0	5.4
	LTTng	2.8	7.3	6.5	5.4	3.2	3.7	0.1	6.7	4.2	0.3	3.7	3.1	7.9	1.8	0.7	5.2	6.8	7.4	0.4	1.2	2.5
	VTrace	0.1	0.8	0.3	0.3	0.5	0.1	0.2	0.8	0.3	0.2	0.7	0.0	0.4	0.2	0.2	0.2	0.6	0.4	0.1	0.4	0.3
Loss Rate	BTrace	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	BBQ	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ftrace	0.81	0.41	0.83	0.92	0.76	0.84	0.84	0.58	0.90	0.51	0.78	0.83	0.43	0.38	0.56	0.91	0.15	0.57	0.49	0.49	0.60
	LTTng	0.81	0.38	0.82	0.92	0.76	0.84	0.84	0.57	0.90	0.46	0.78	0.83	0.31	0.40	0.53	0.91	0.19	0.53	0.56	0.62	0.60
	VTrace	0.89	0.85	0.95	0.93	0.79	0.85	0.91	0.90	0.93	0.92	0.80	0.86	0.95	0.96	0.96	0.95	0.75	0.95	0.93	0.94	0.90
#Fragments	BTrace	47	86	87	53	34	49	60	101	49	73	35	83	69	111	87	68	73	70	86	54	65
	BBQ	2	21	106	3	3	4	7	243	12	169	1	105	222	66	45	236	122	152	76	385	34
	ftrace	2e4	1e4	2e4	1e4	1e4	1e4	3e4	1e4	2e4	1e4	2e3	3e4	8e3	1e4	1e4	2e4	6e3	1e4	9e3	1e4	1e4
	LTTng	2e4	9e3	1e4	1e4	1e4	1e4	3e4	9e3	2e4	1e4	3e3	3e4	6e3	1e4	1e4	2e4	7e3	1e4	2e4	2e4	1e4
	VTrace	7e4	8e4	7e4	7e4	4e4	4e4	7e4	8e4	7e4	9e4	1e4	8e4	8e4	9e4	7e4	8e4	6e4	9e4	8e4	9e4	6e4
G.M. Lat. (ns)	BTrace	54	52	52	53	55	56	56	51	52	52	56	53	52	52	53	54	53	52	51	52	53
	BBQ	111	138	495	119	95	102	148	510	115	894	88	485	828	738	794	447	520	822	763	776	324
	ftrace	62	63	64	62	62	62	63	63	61	64	61	63	64	64	65	64	66	65	63	64	63
	LTTng	263	251	246	260	266	266	260	250	251	227	271	251	236	238	239	250	250	232	243	228	249
	VTrace	278	296	283	264	238	229	235	360	264	359	207	290	355	308	332	321	300	341	297	359	292

Evaluation

Workloads (20 in total)

- The top 10 applications and games by number of downloads in the app store
- Performance testing software
- Typical usage scenarios (e.g., lock screen)



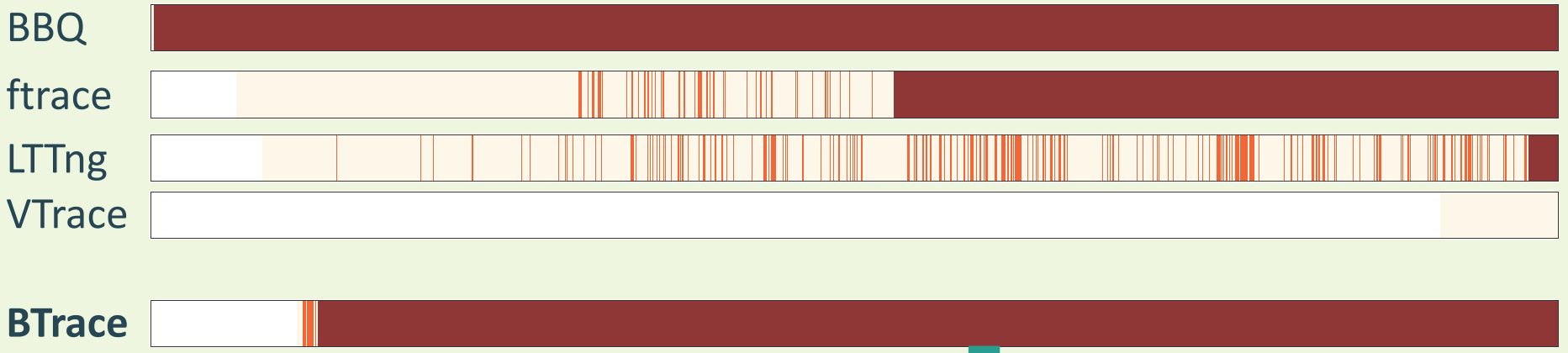
2x continuous traces

20% lower latency

Evaluation

Workloads (20 in total)

- The top 10 applications and games by number of downloads in the app store
- Performance testing software
- Typical usage scenarios (e.g., lock screen)



2x continuous traces

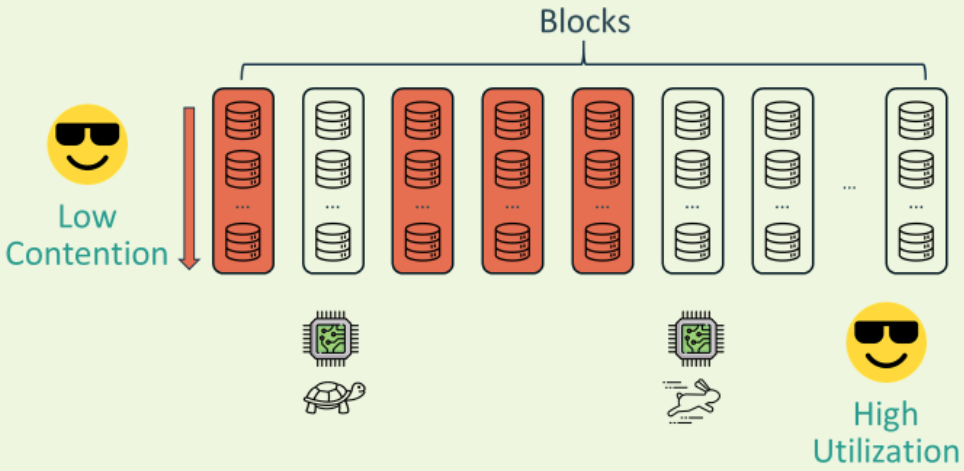
20% lower latency

Deployed on Beta release smartphones (>3M users), found 200 long-duration cause-effect bugs.

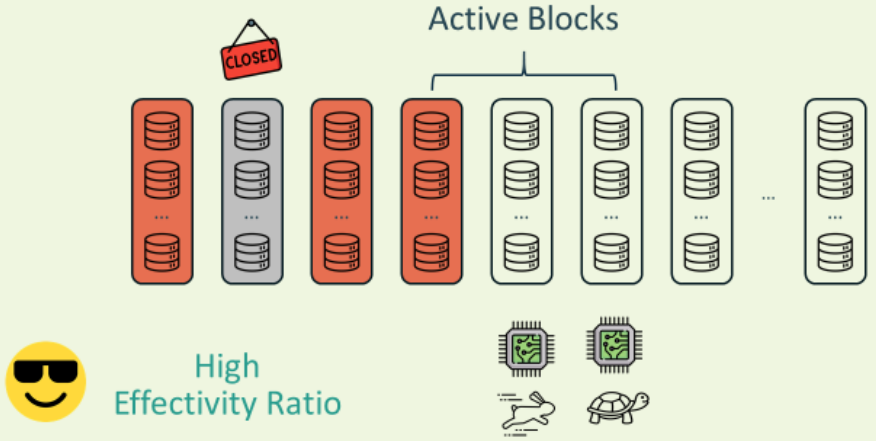
BTrace

2x continuous traces 20% lower latency Resizing without SMR

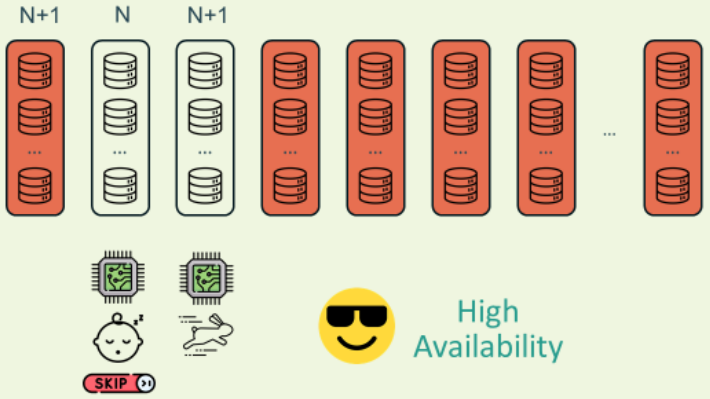
Improving Utilization via Block Partitioning



Enhance Effectivity via Block Closing



Ensure Availability via Block Skipping



Enable Resizing via Implicit Reclaiming