2

Scheduling

StarPU Scheduling Policies

- No *one size fits all* policy

StarPU Scheduling Policies

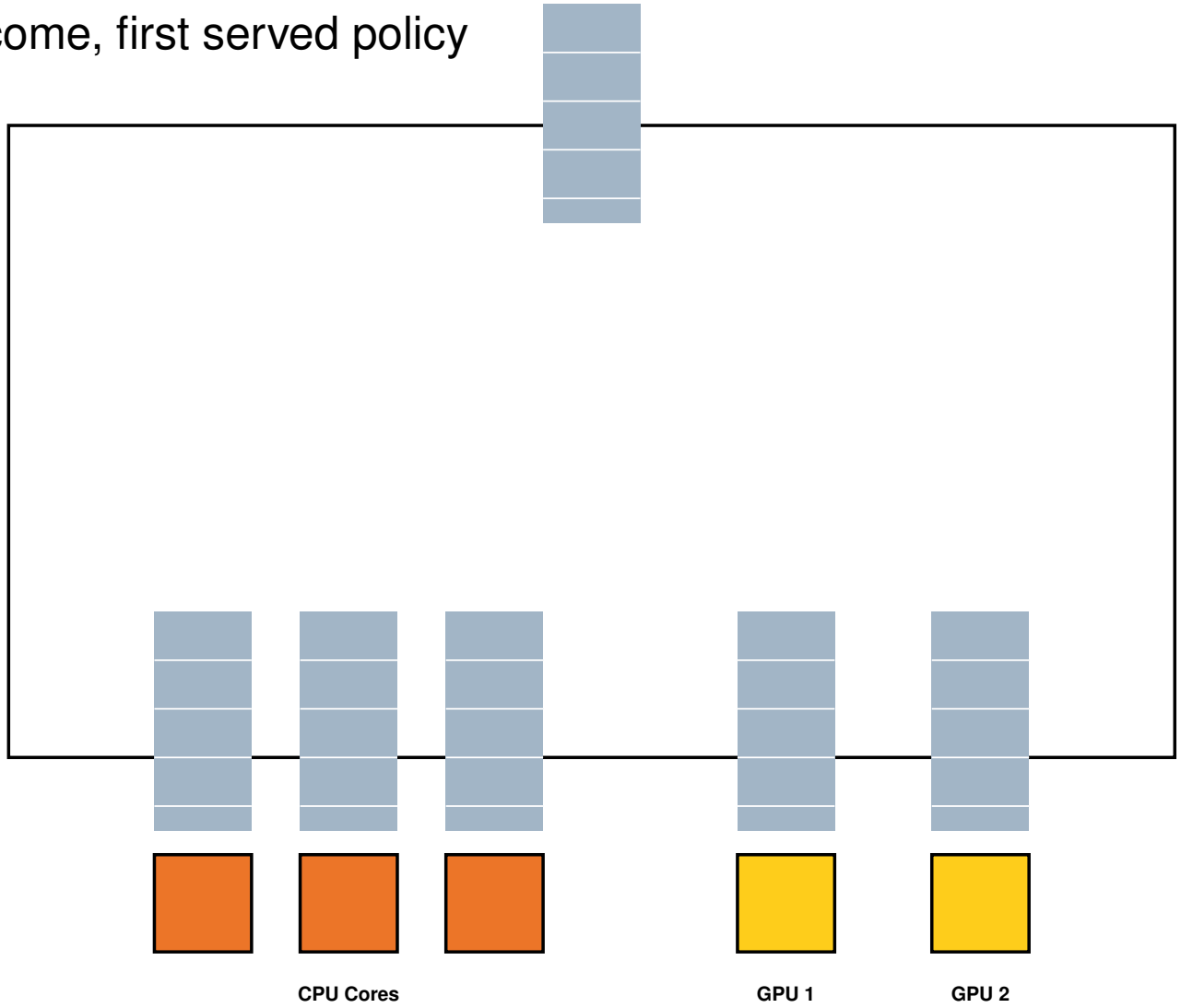
- No *one size fits all* policy
- Selectable scheduling policy
  - Predefined set of popular policies

The Eager Scheduler

- First come, first served policy

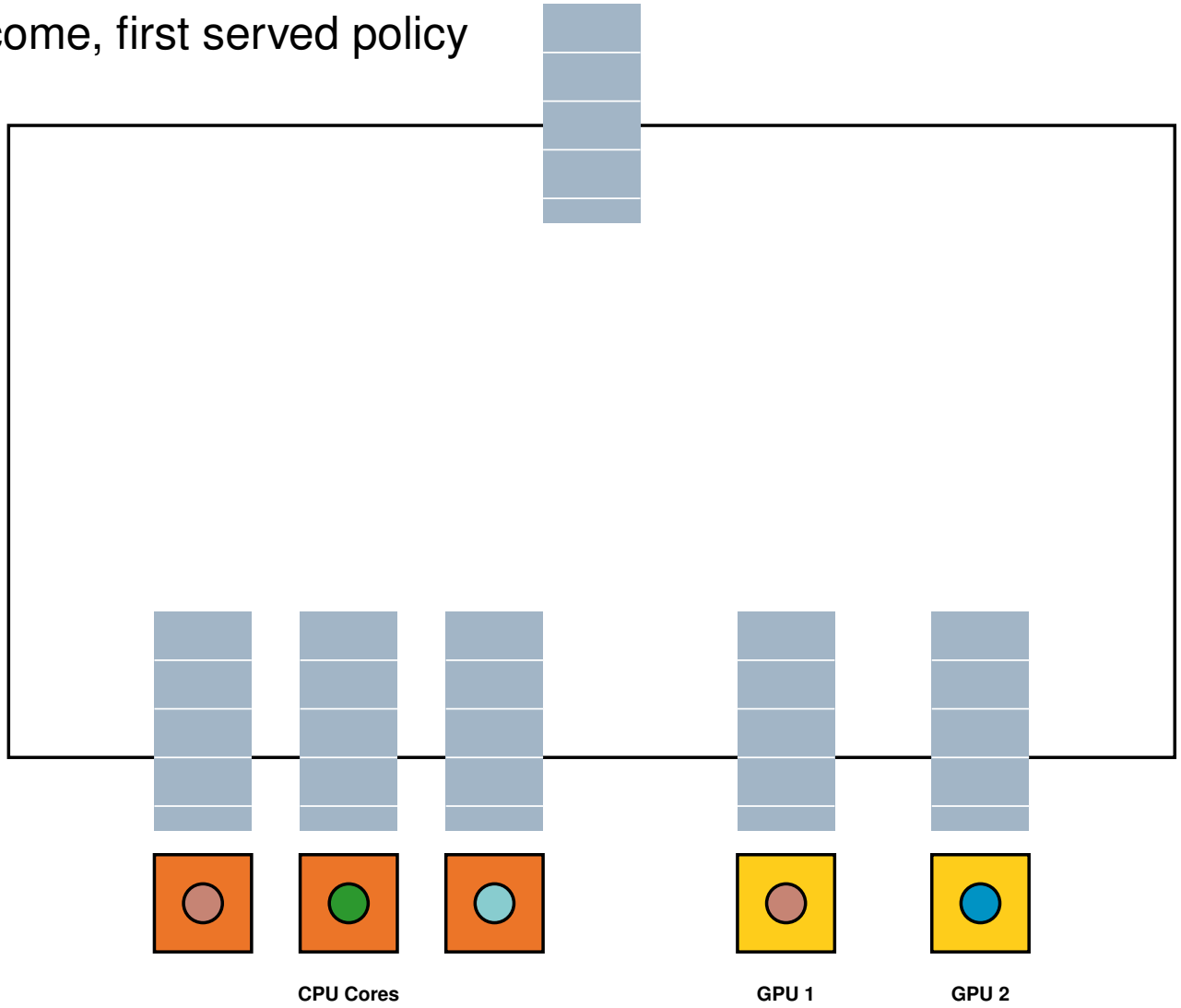
The Eager Scheduler

- First come, first served policy



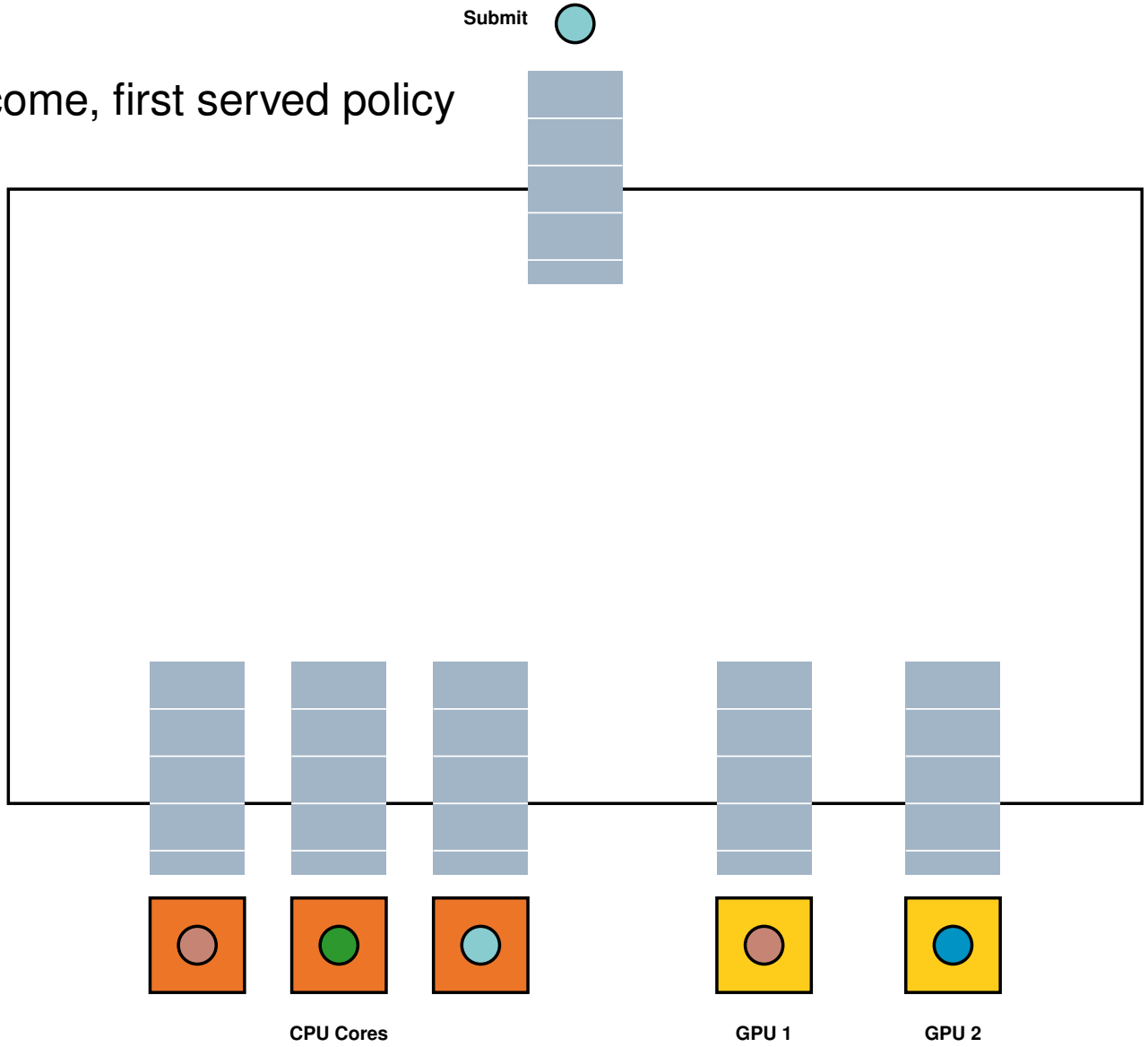
The Eager Scheduler

- First come, first served policy



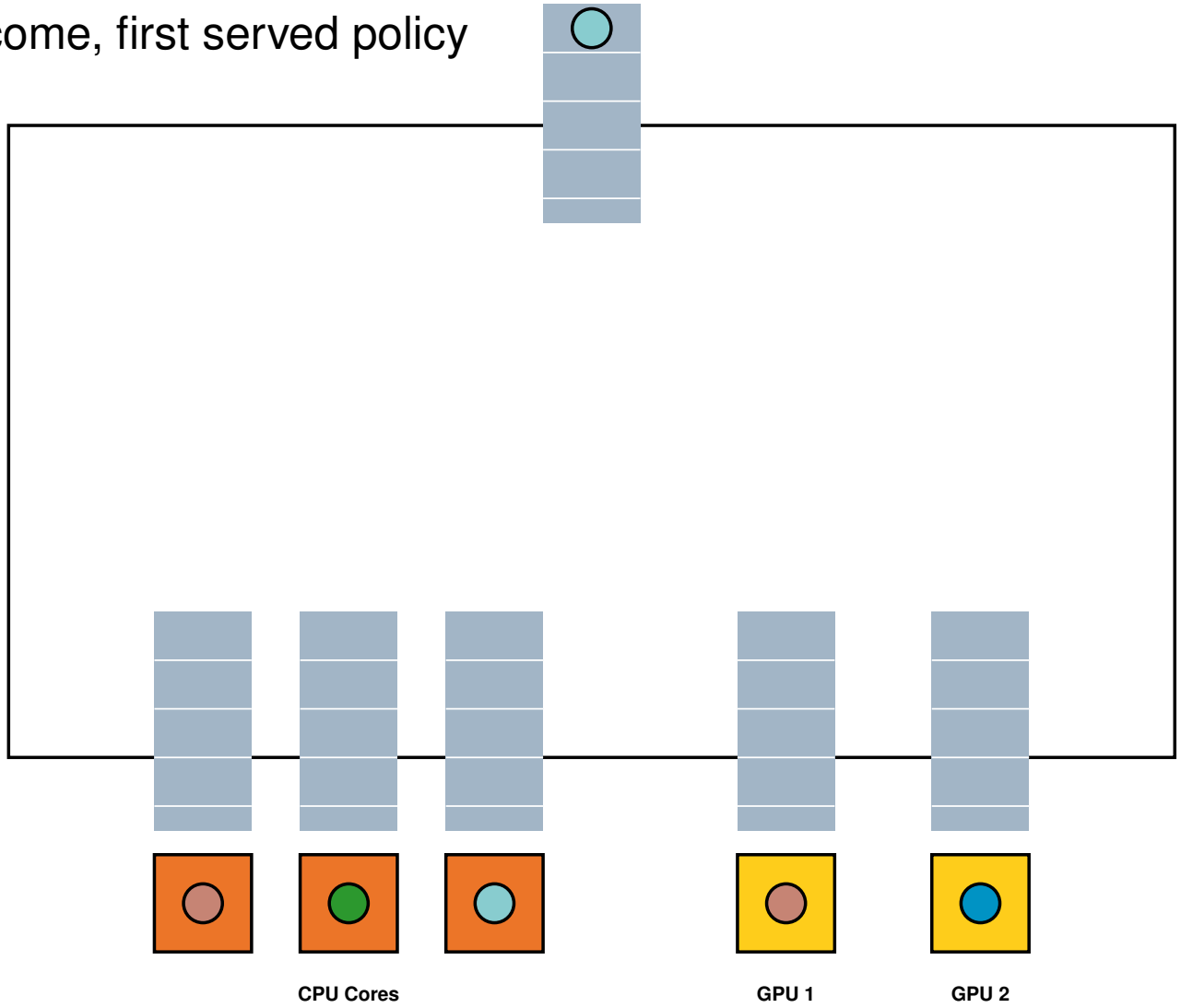
The Eager Scheduler

- First come, first served policy



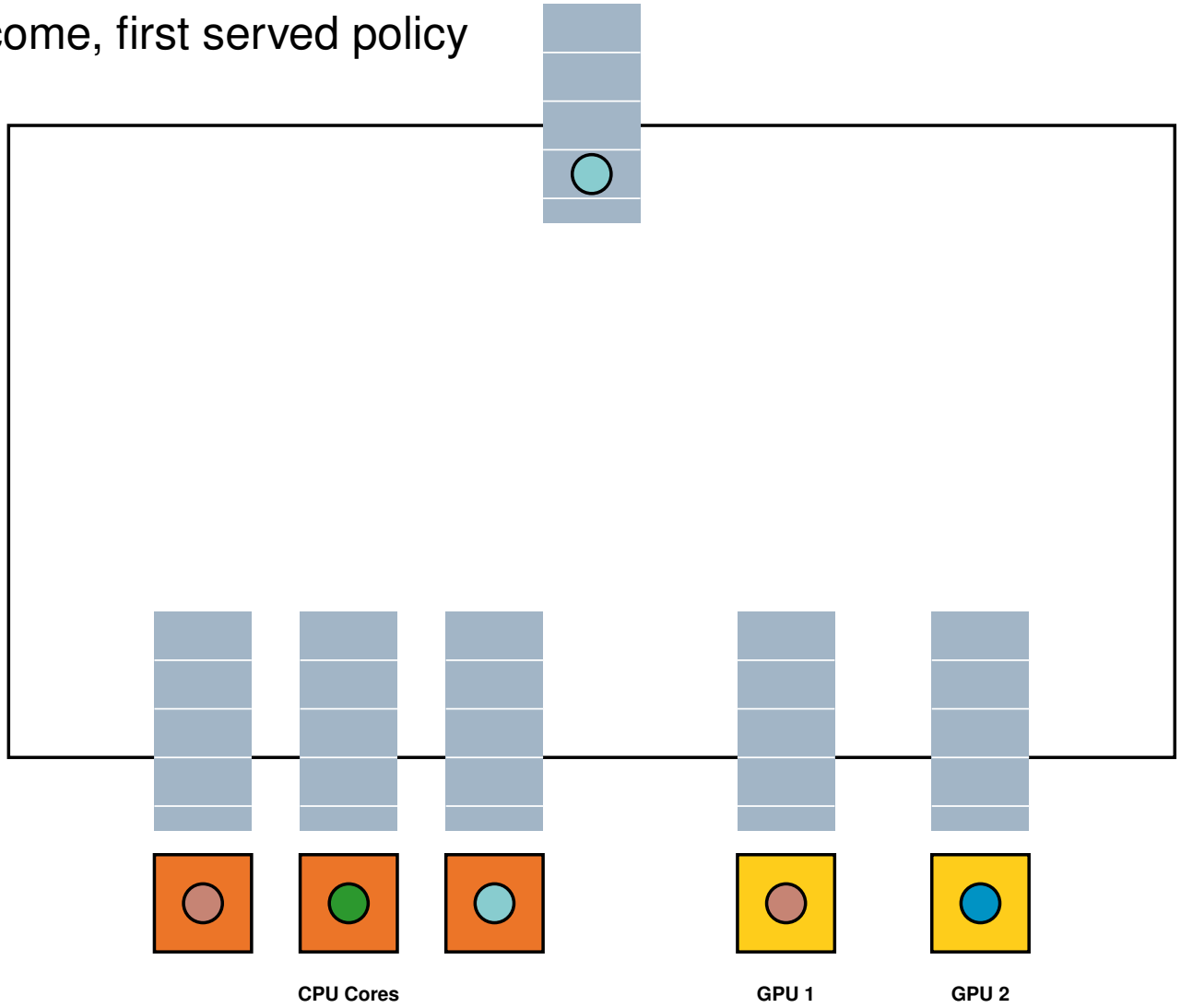
The Eager Scheduler

- First come, first served policy



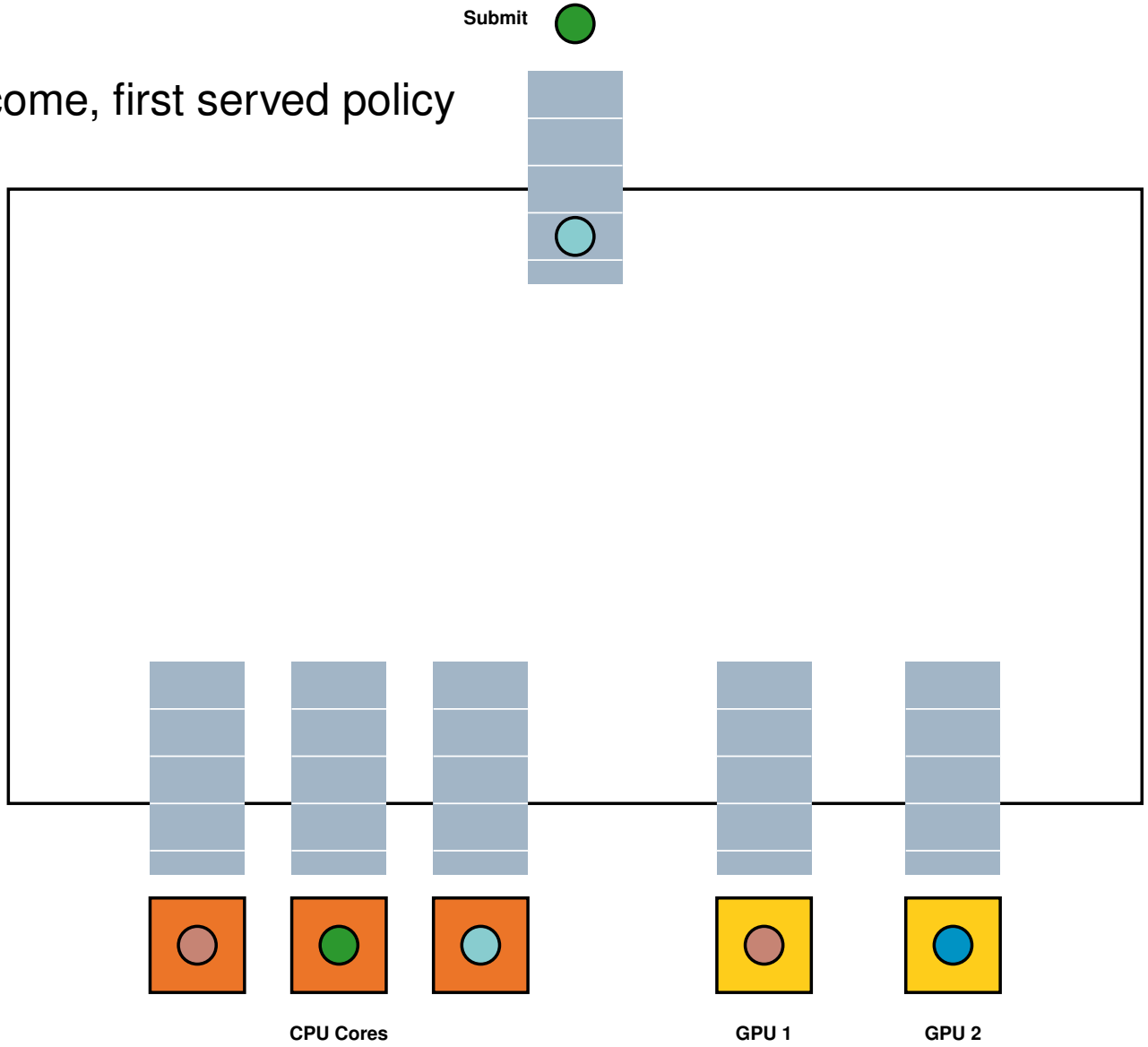
The Eager Scheduler

- First come, first served policy



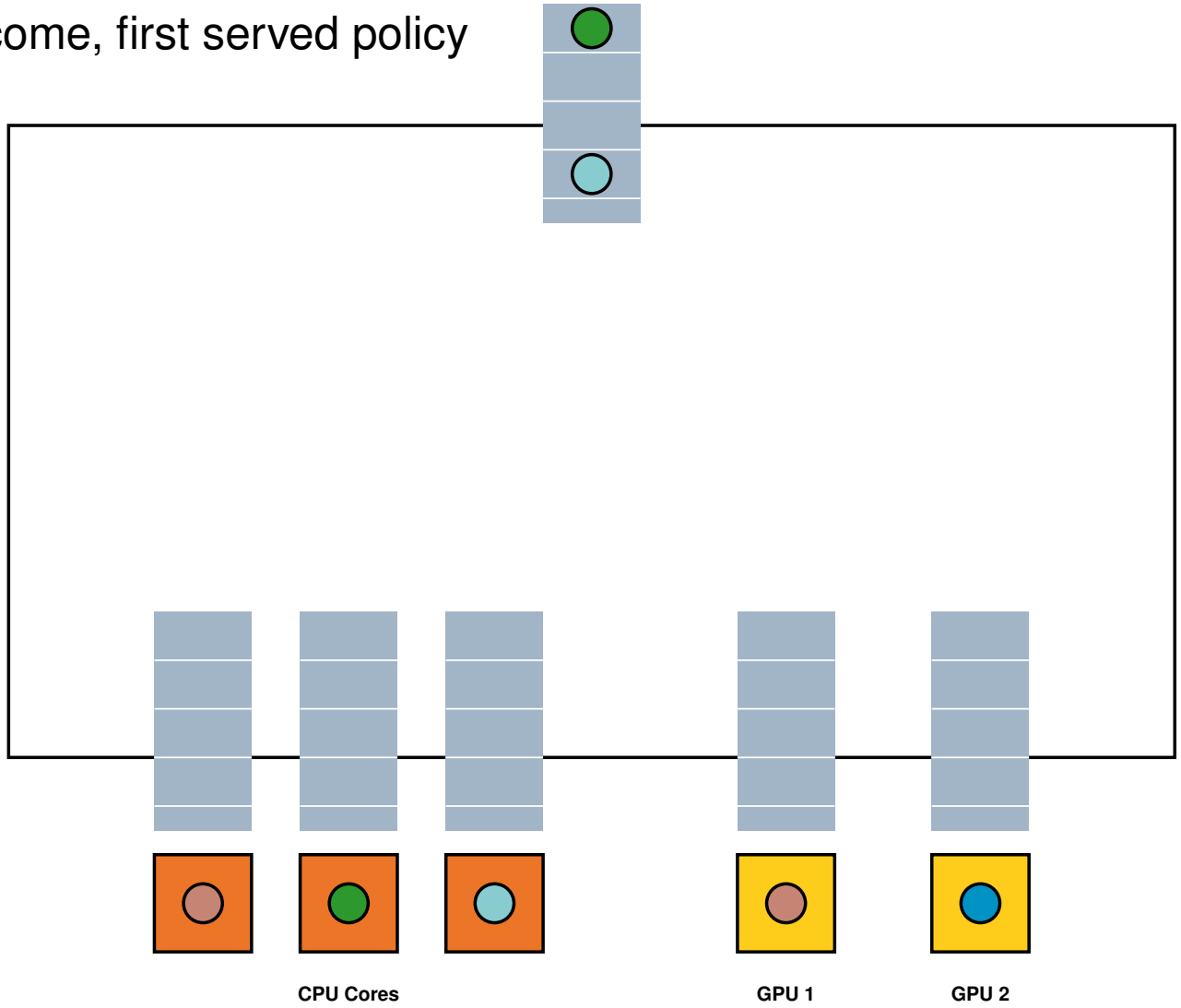
The Eager Scheduler

- First come, first served policy



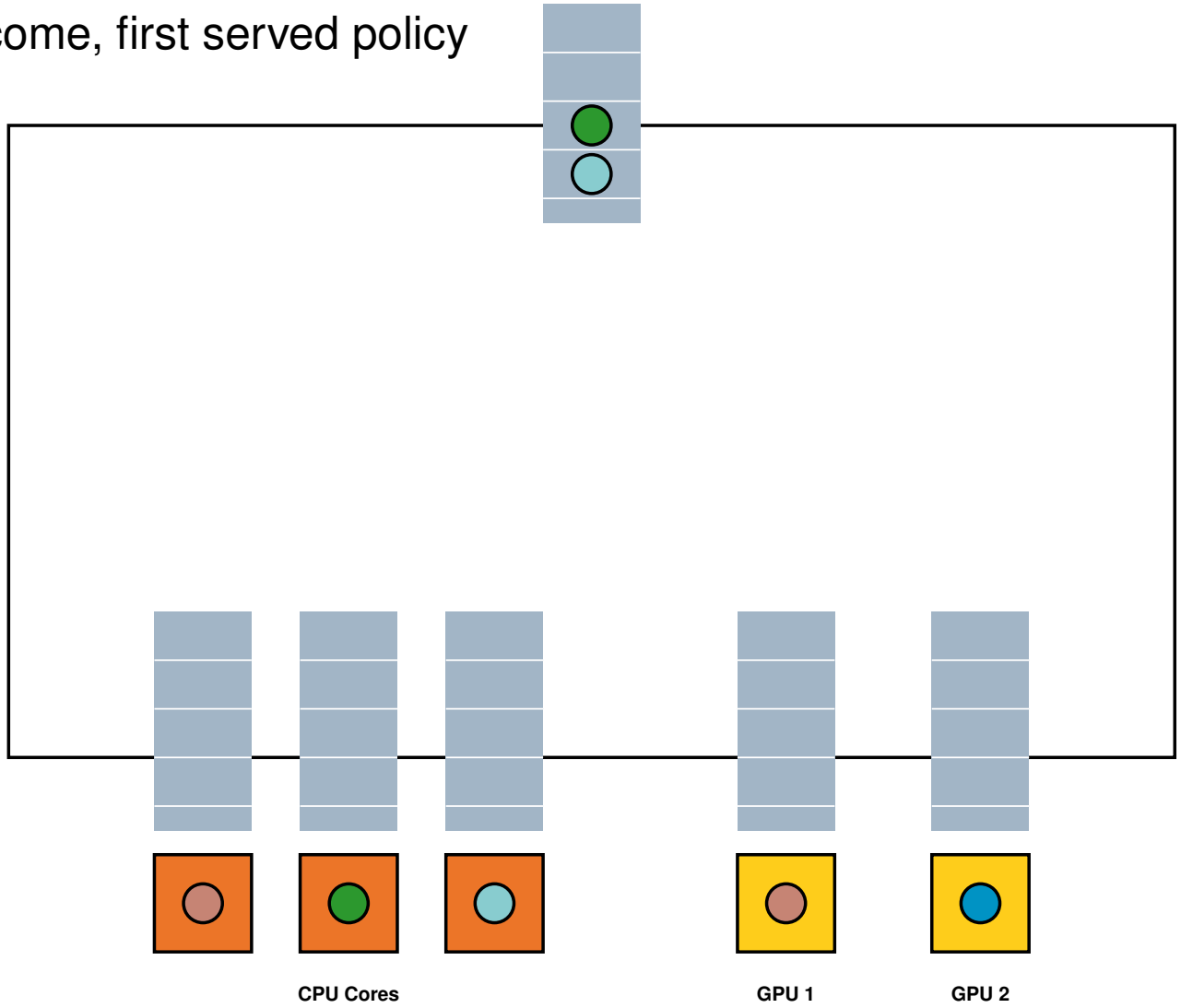
The Eager Scheduler

- First come, first served policy



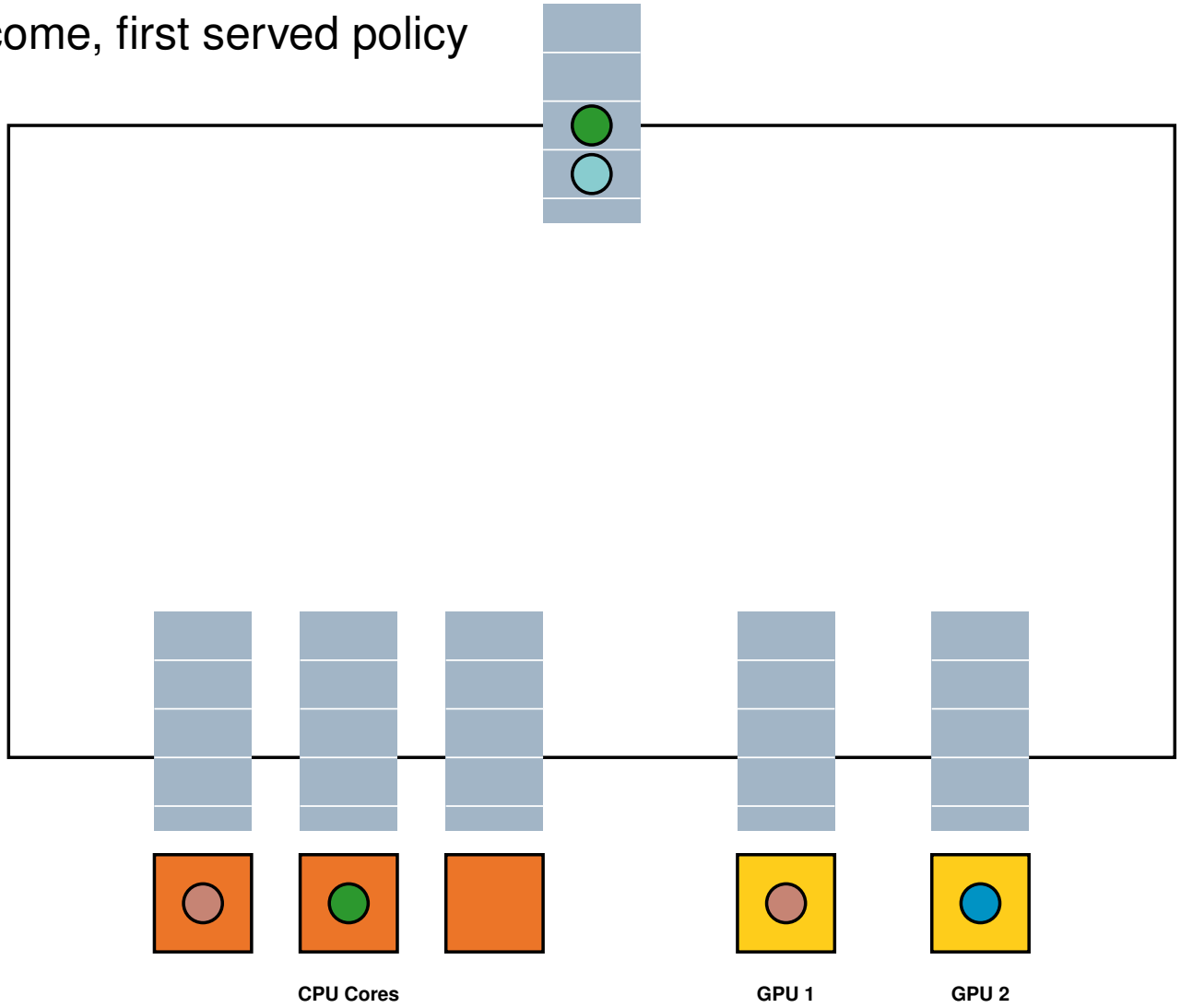
The Eager Scheduler

- First come, first served policy



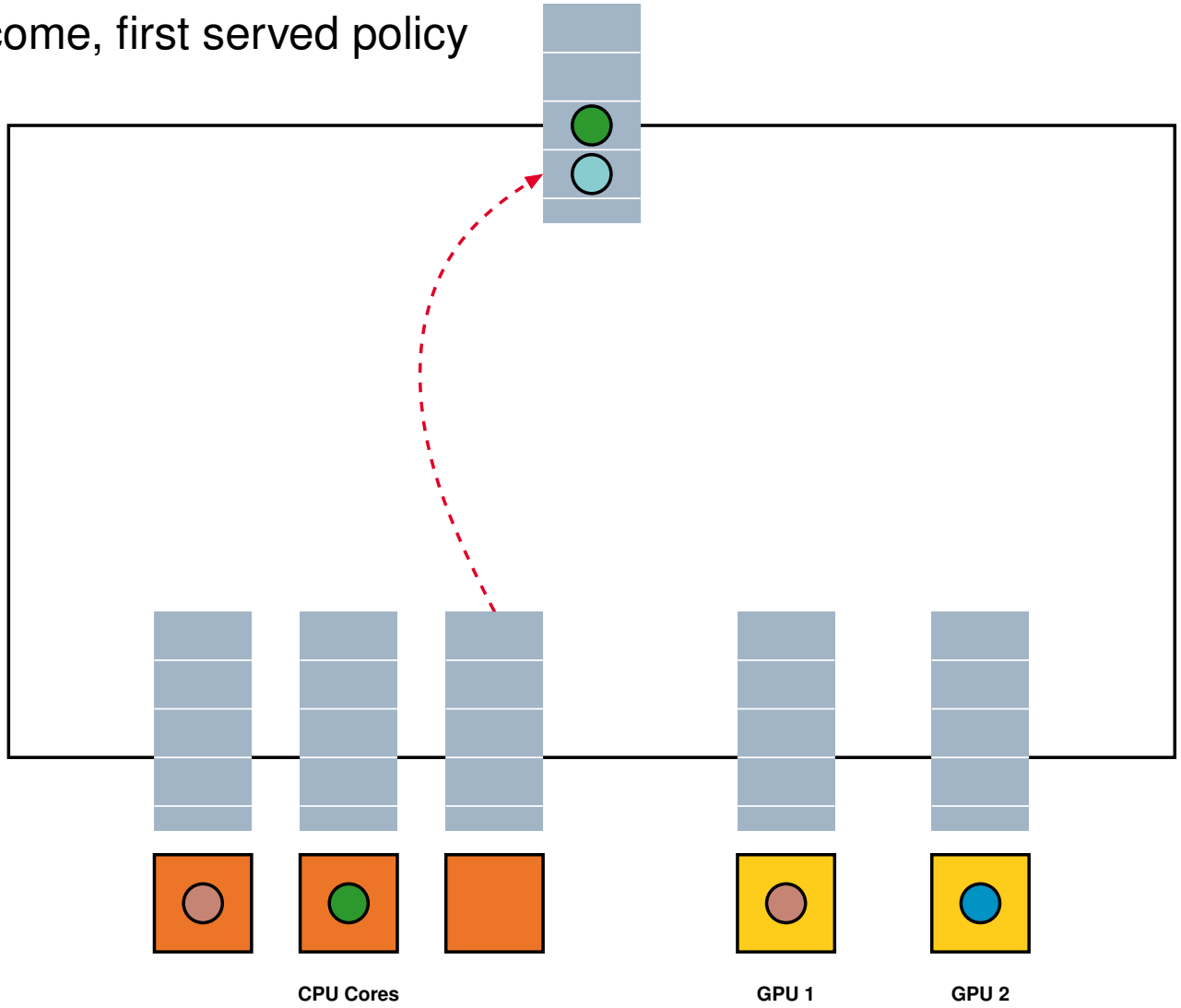
The Eager Scheduler

- First come, first served policy



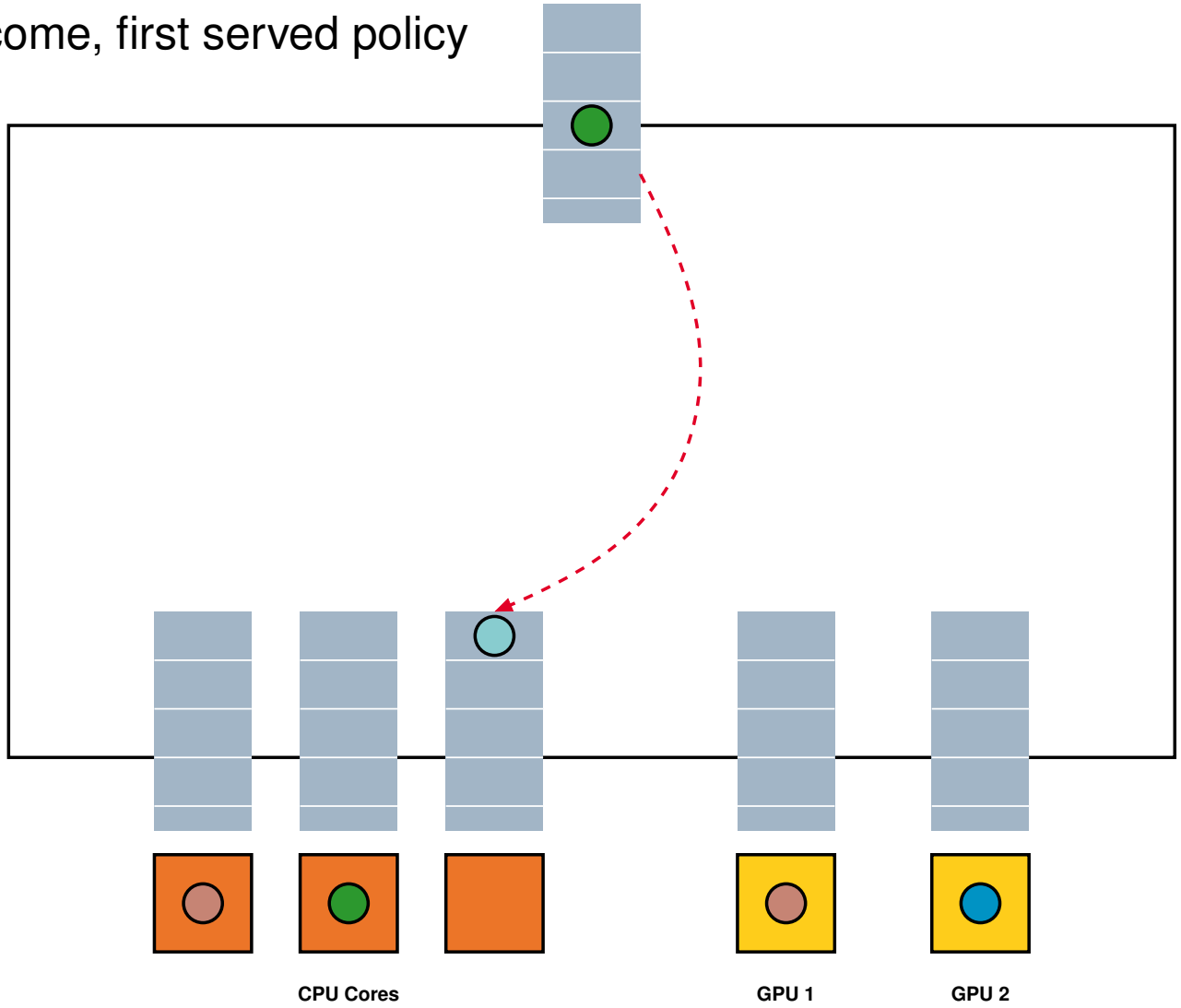
The Eager Scheduler

- First come, first served policy



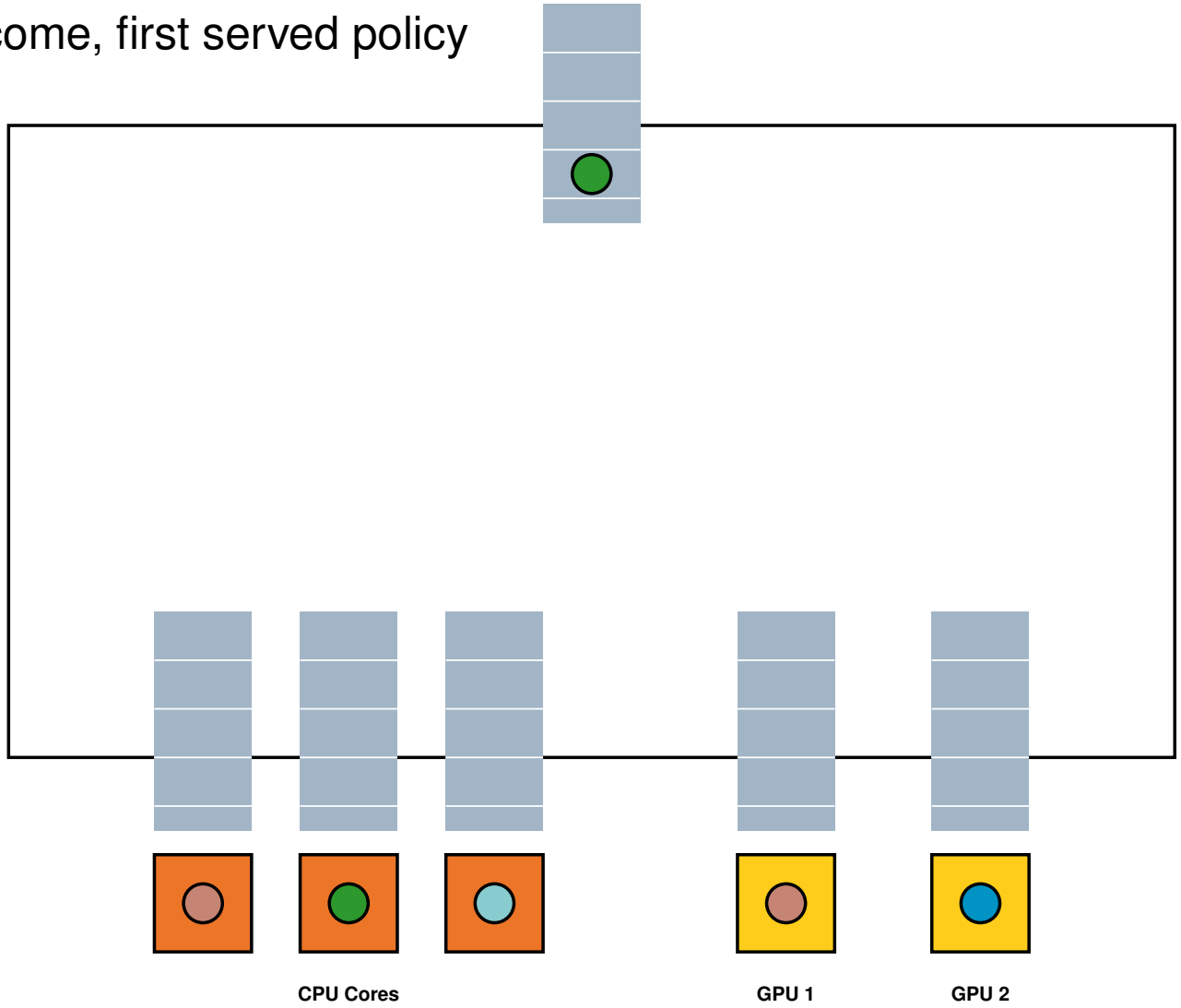
The Eager Scheduler

- First come, first served policy



The Eager Scheduler

- First come, first served policy

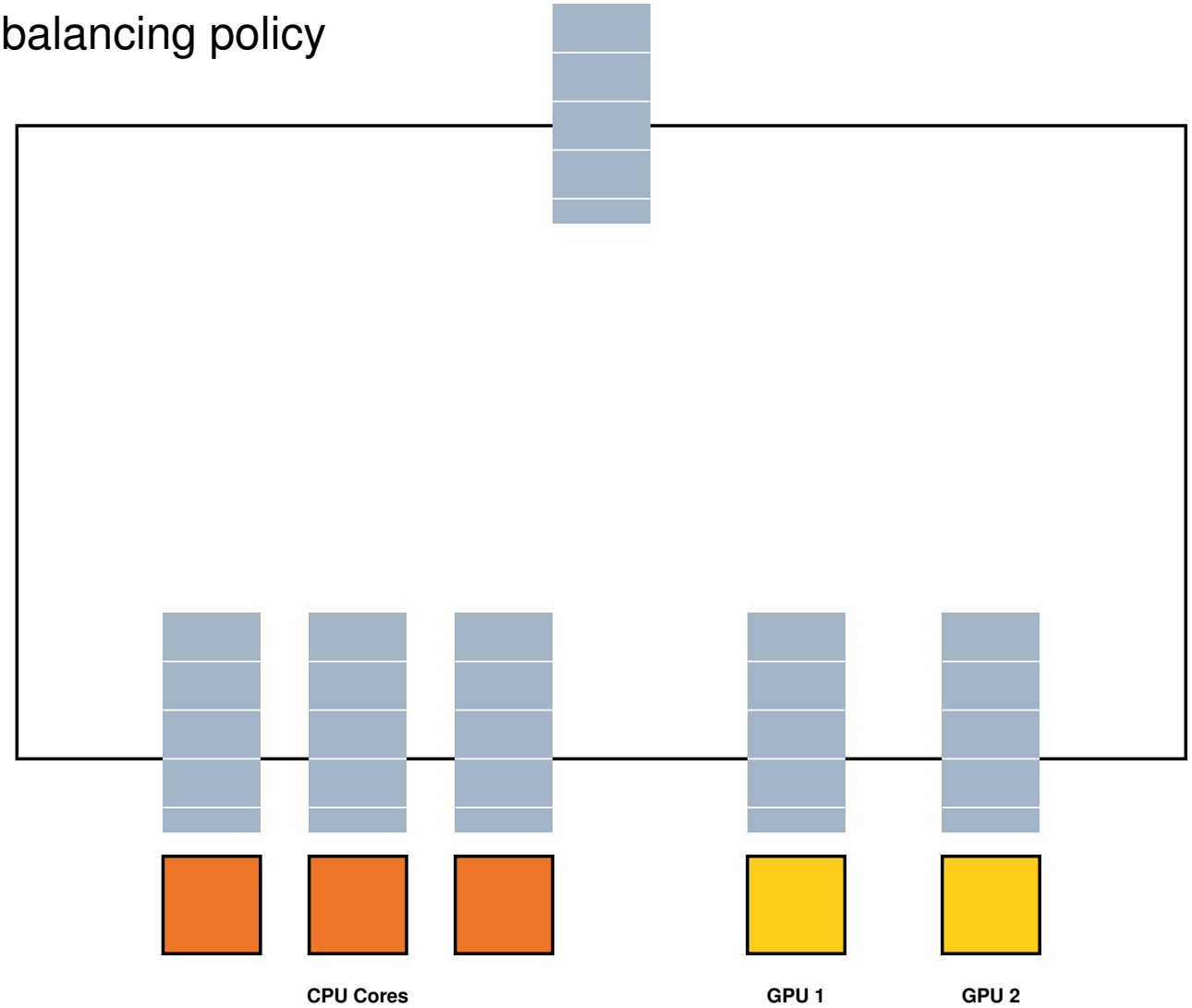


The Work Stealing Scheduler

- Load balancing policy

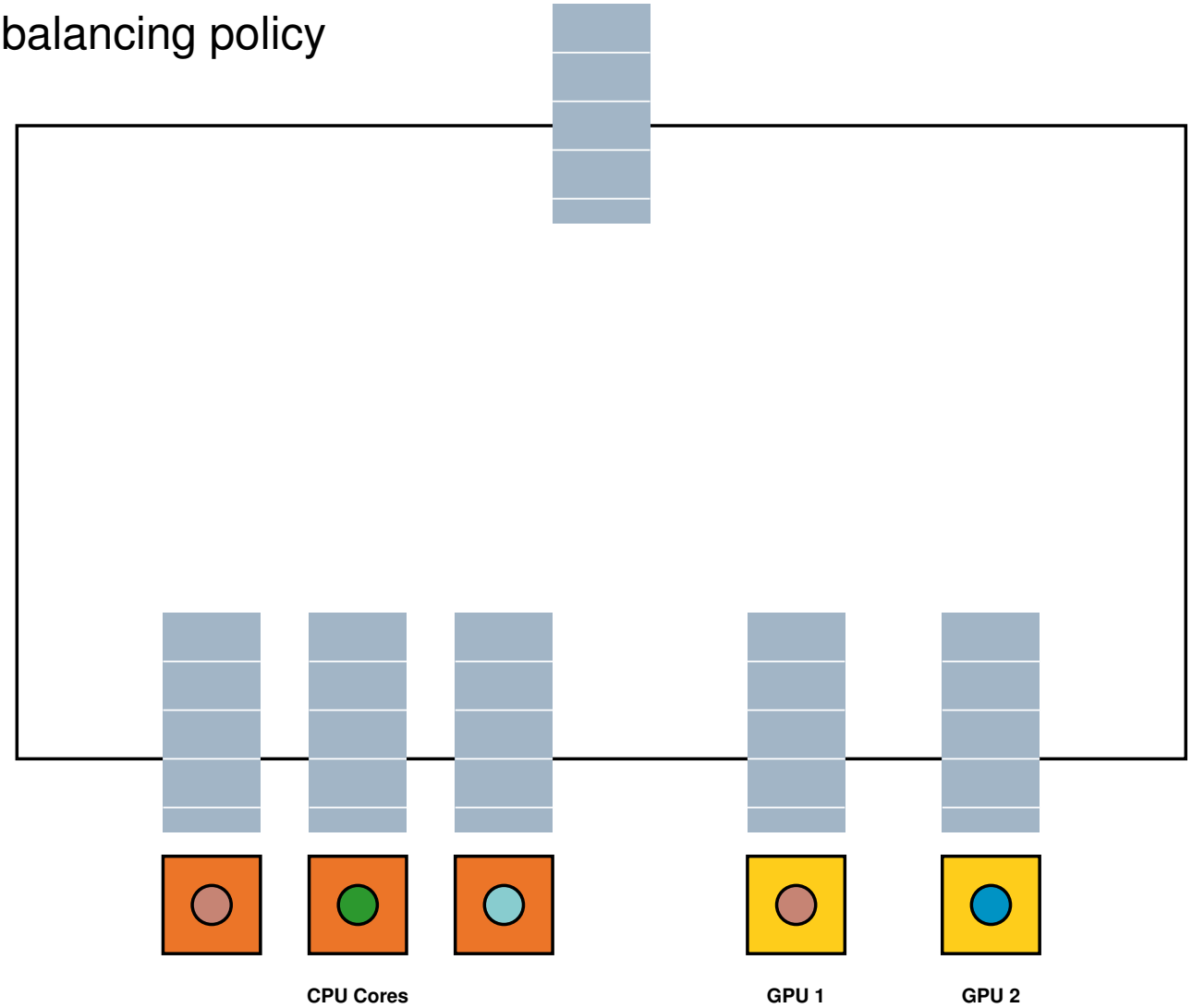
The Work Stealing Scheduler

- Load balancing policy



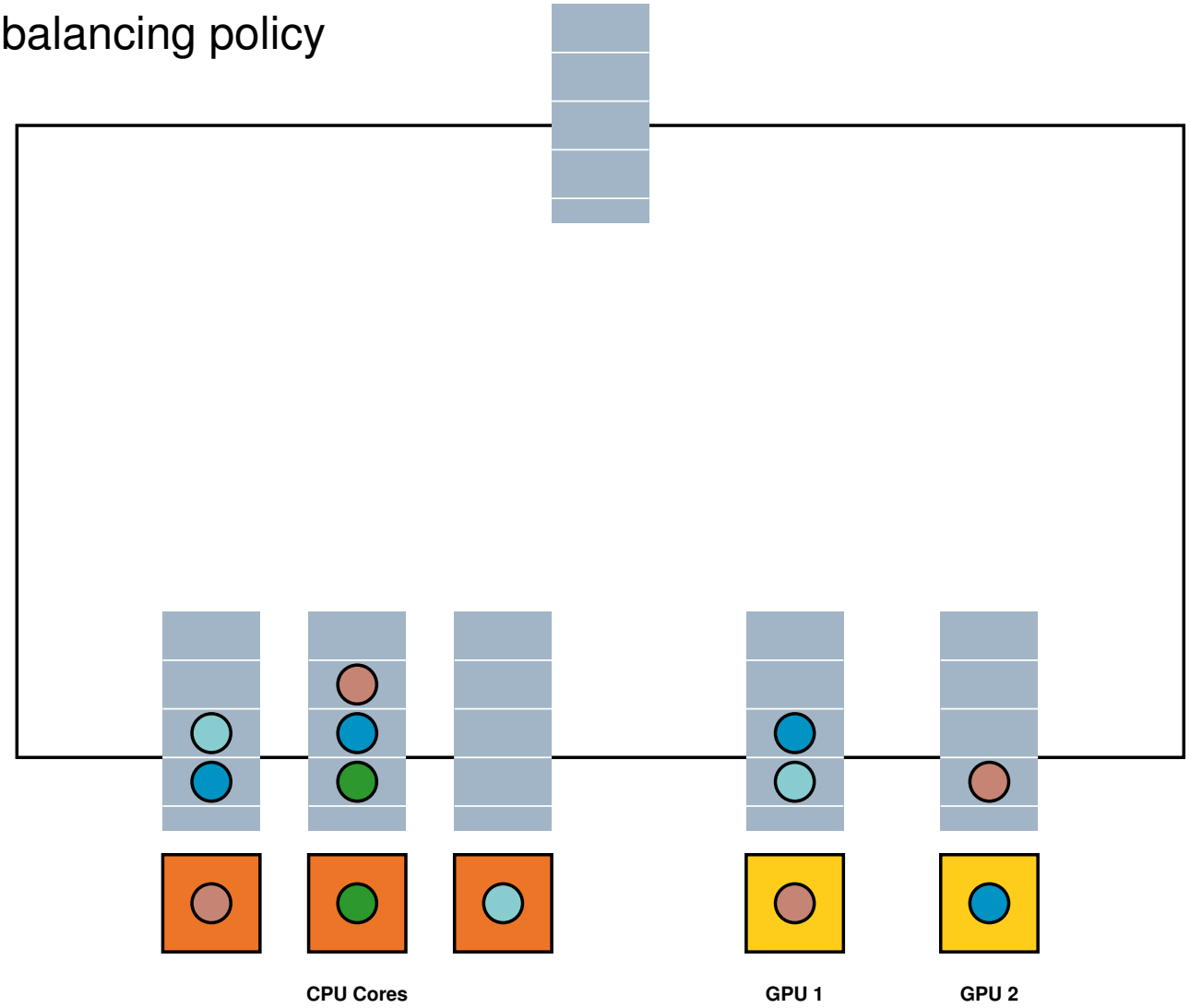
The Work Stealing Scheduler

- Load balancing policy



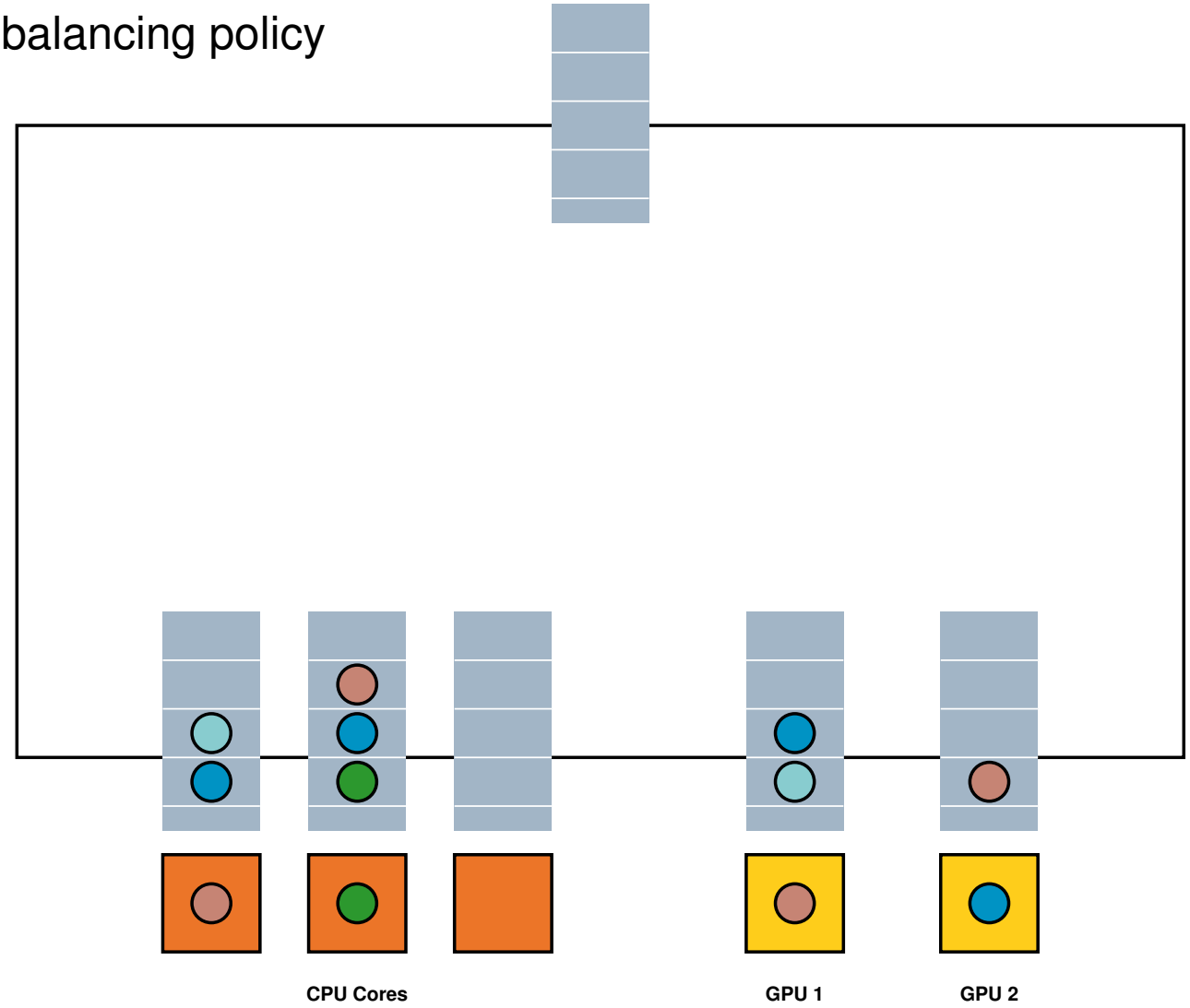
The Work Stealing Scheduler

- Load balancing policy



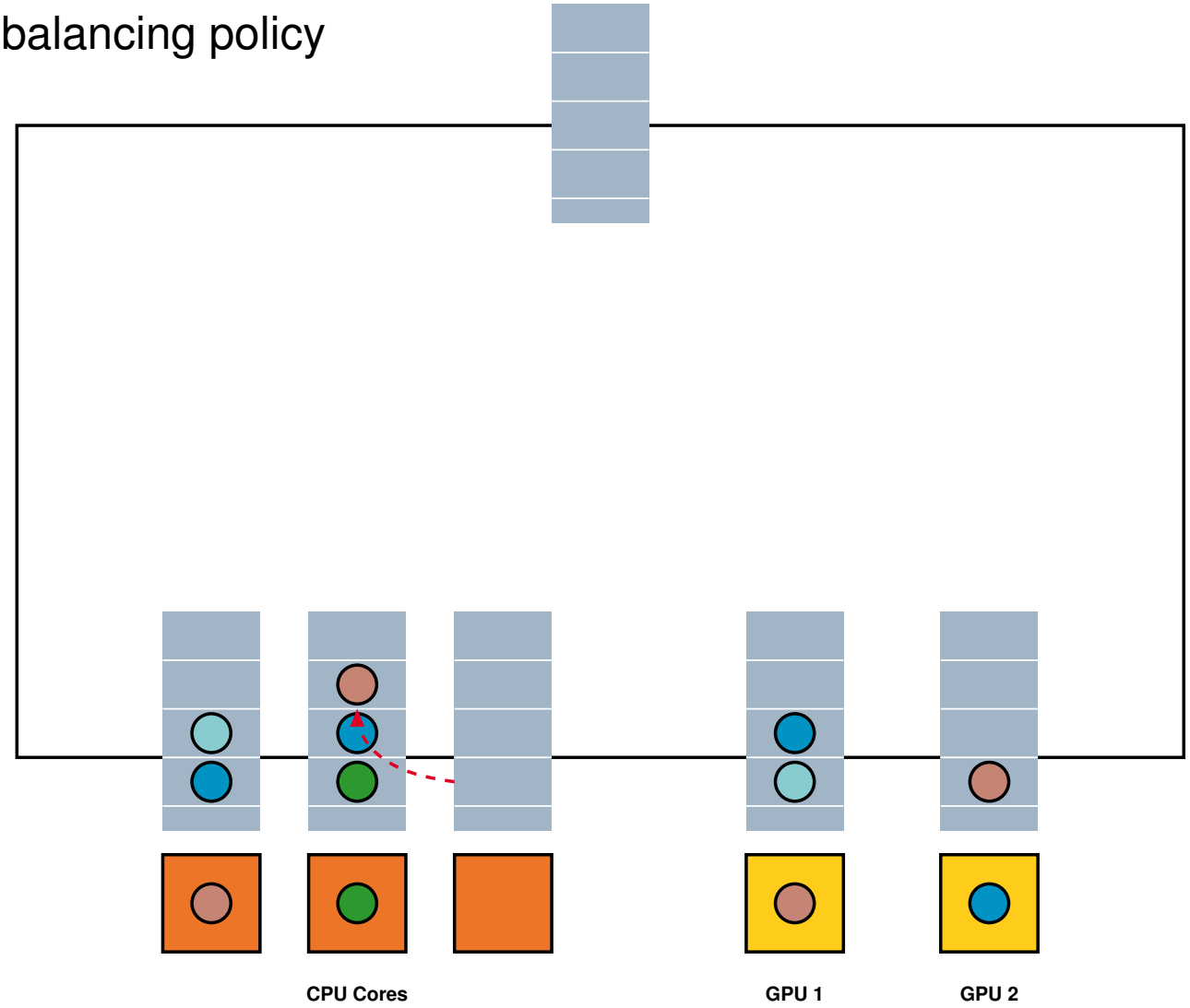
The Work Stealing Scheduler

- Load balancing policy



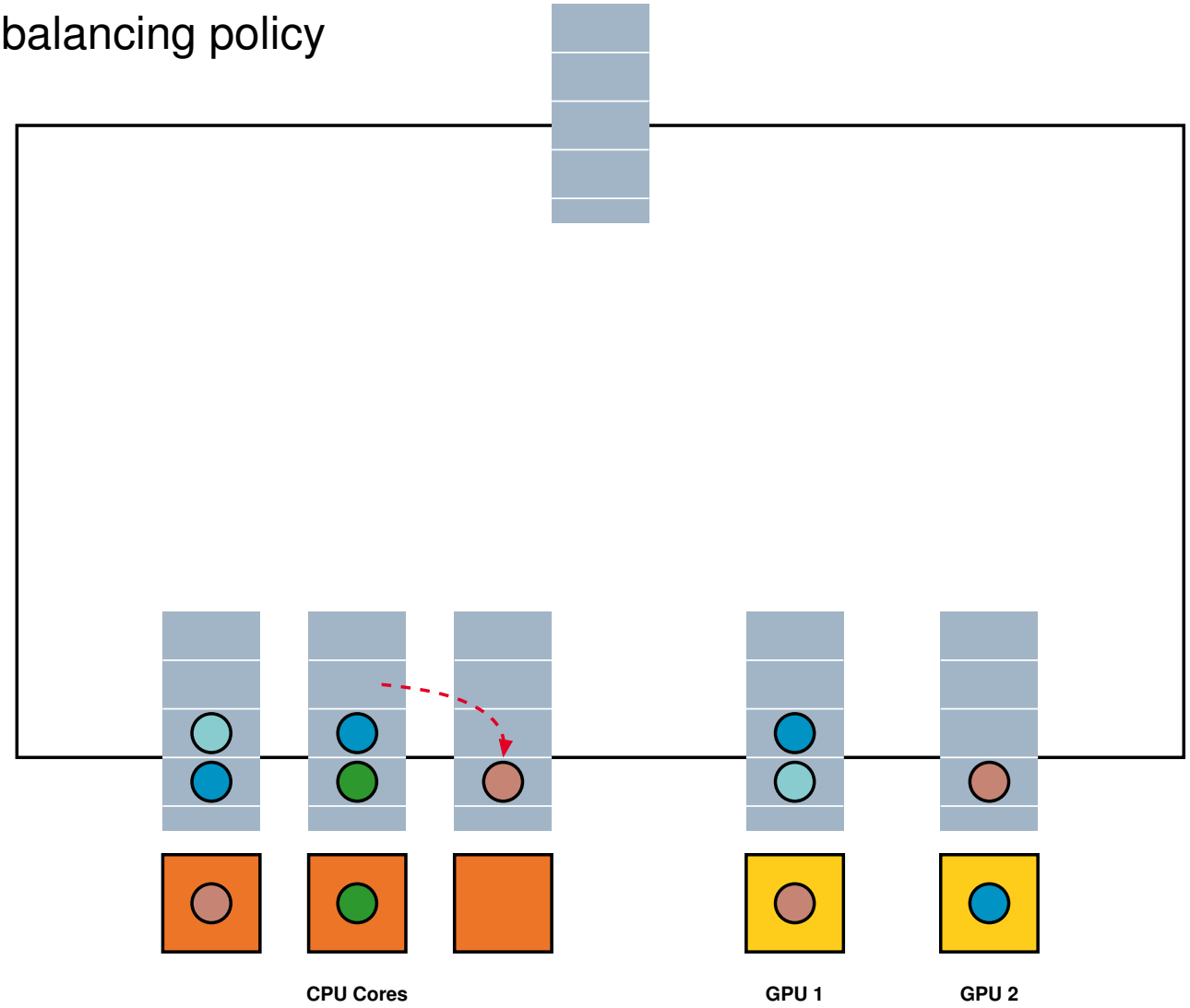
The Work Stealing Scheduler

- Load balancing policy



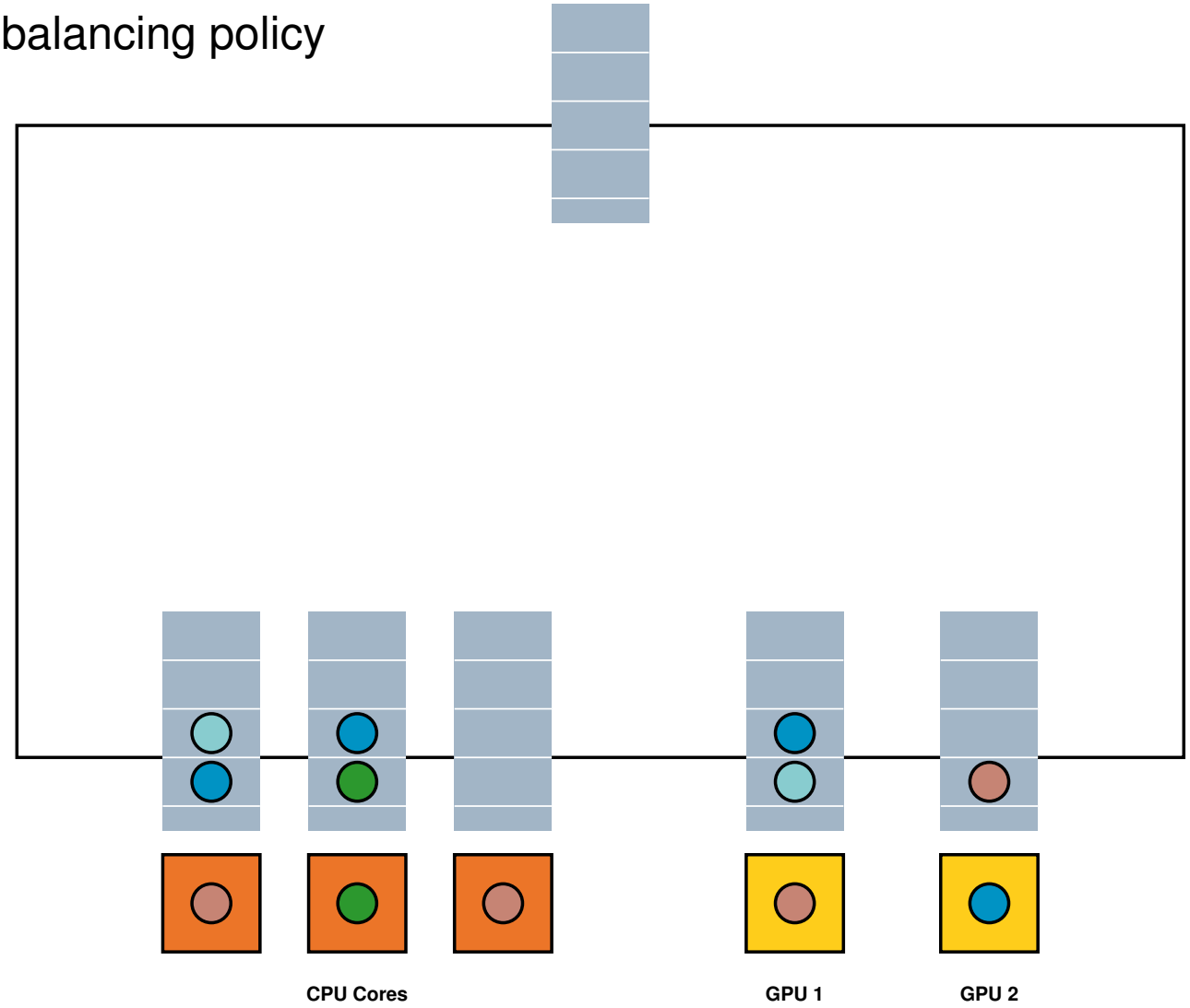
The Work Stealing Scheduler

- Load balancing policy



The Work Stealing Scheduler

- Load balancing policy



Going Beyond

Scheduling is a decision process

Going Beyond

Scheduling is a decision process

- Providing more input to the scheduler. . .

Going Beyond

Scheduling is a decision process

- Providing more input to the scheduler. . .
- . . . can lead to better scheduling decisions

Going Beyond

Scheduling is a decision process

- Providing more input to the scheduler. . .
- . . . can lead to better scheduling decisions

What kind of information?

Going Beyond

Scheduling is a decision process

- Providing more input to the scheduler. . .
- . . . can lead to better scheduling decisions

What kind of information?

- Relative importance of tasks
  - Priorities

Going Beyond

Scheduling is a decision process

- Providing more input to the scheduler. . .
- . . . can lead to better scheduling decisions

What kind of information?

- Relative importance of tasks
  - Priorities
- Cost of tasks
  - Codelet models

Going Beyond

Scheduling is a decision process

- Providing more input to the scheduler. . .
- . . . can lead to better scheduling decisions

What kind of information?

- Relative importance of tasks
  - Priorities
- Cost of tasks
  - Codelet models
- Cost of transferring data
  - Bus calibration

The Prio Scheduler

- Describe the relative importance of tasks

The Prio Scheduler

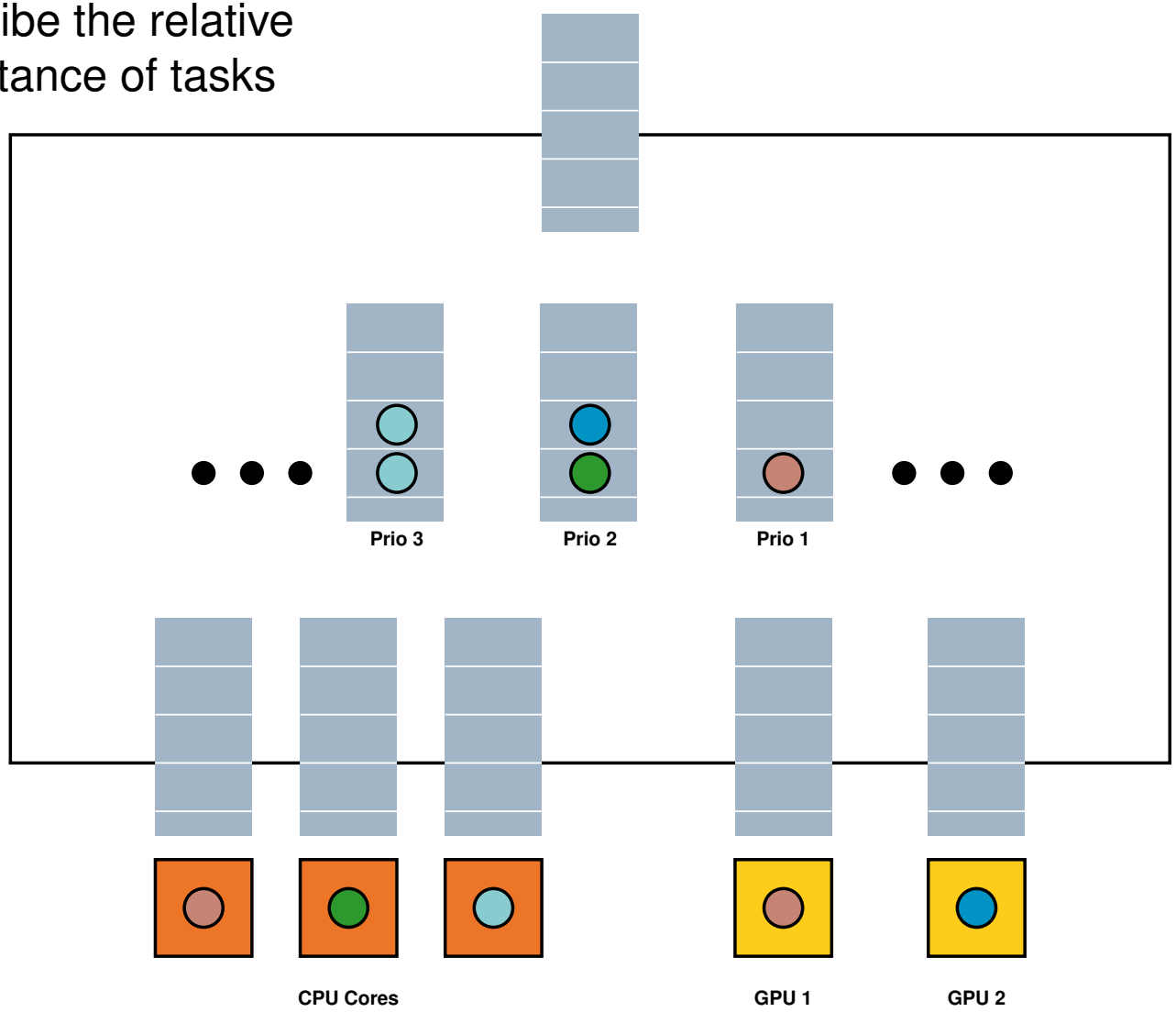
- Describe the relative importance of tasks
- Assign priorities to tasks
  - Values: −5 .. 0 .. +5

The Prio Scheduler

- Describe the relative importance of tasks
- Assign priorities to tasks
  - Values: -5 .. 0 .. $+5$
- Tell which task matter
  - Tasks that unlock key data pieces
  - Tasks that generate a lot of parallelism

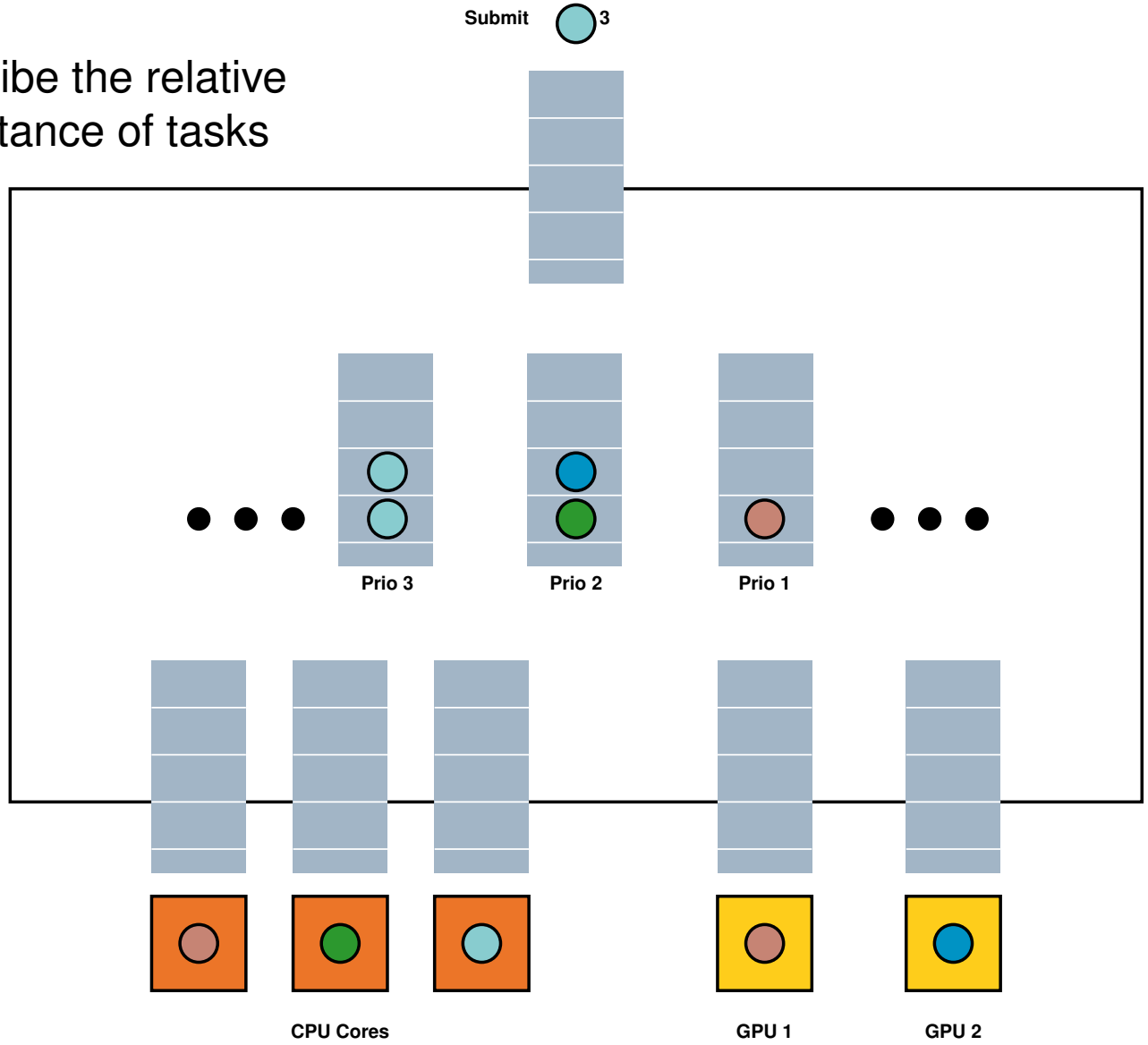
The Prio Scheduler

- Describe the relative importance of tasks



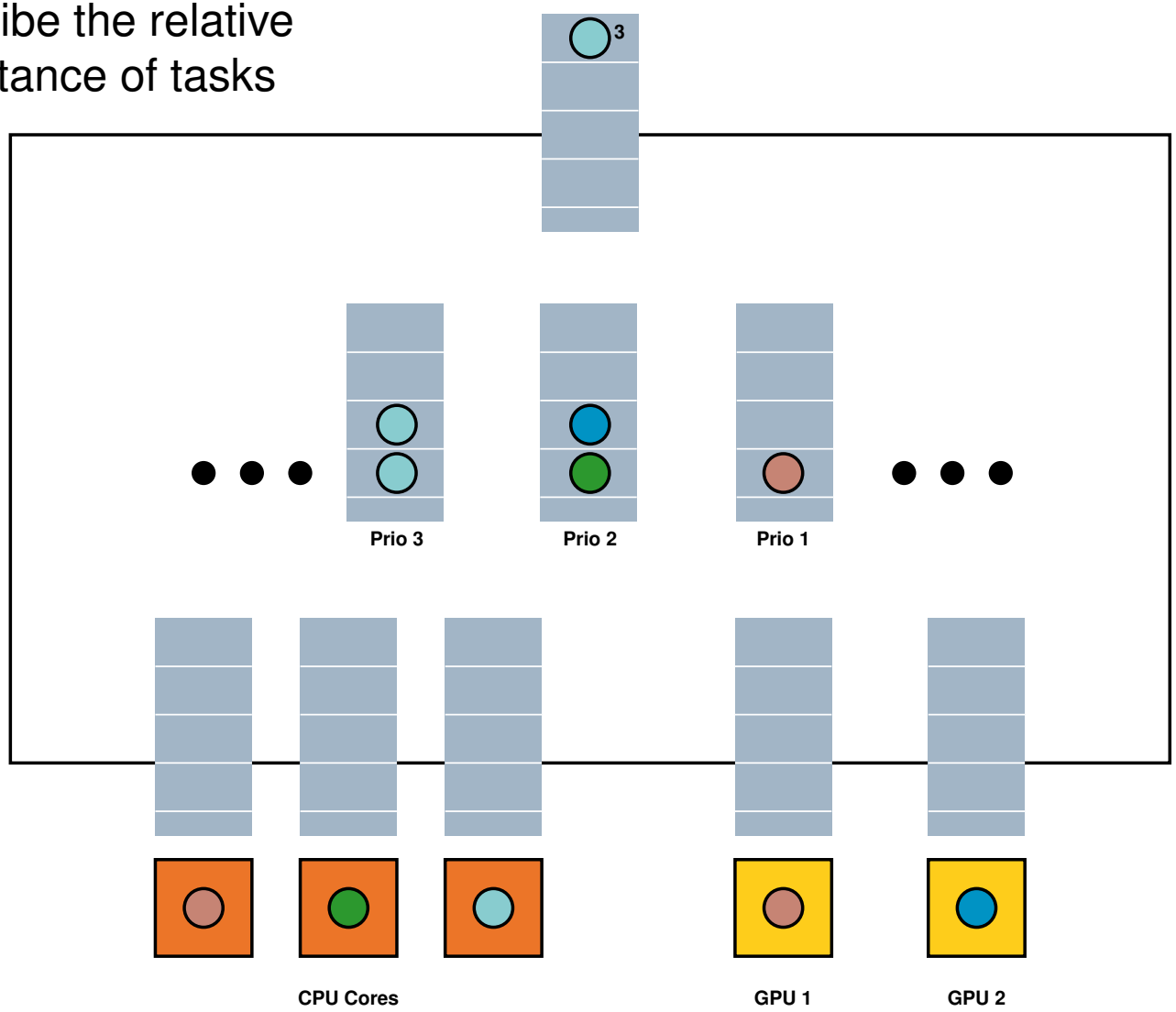
The Prio Scheduler

- Describe the relative importance of tasks



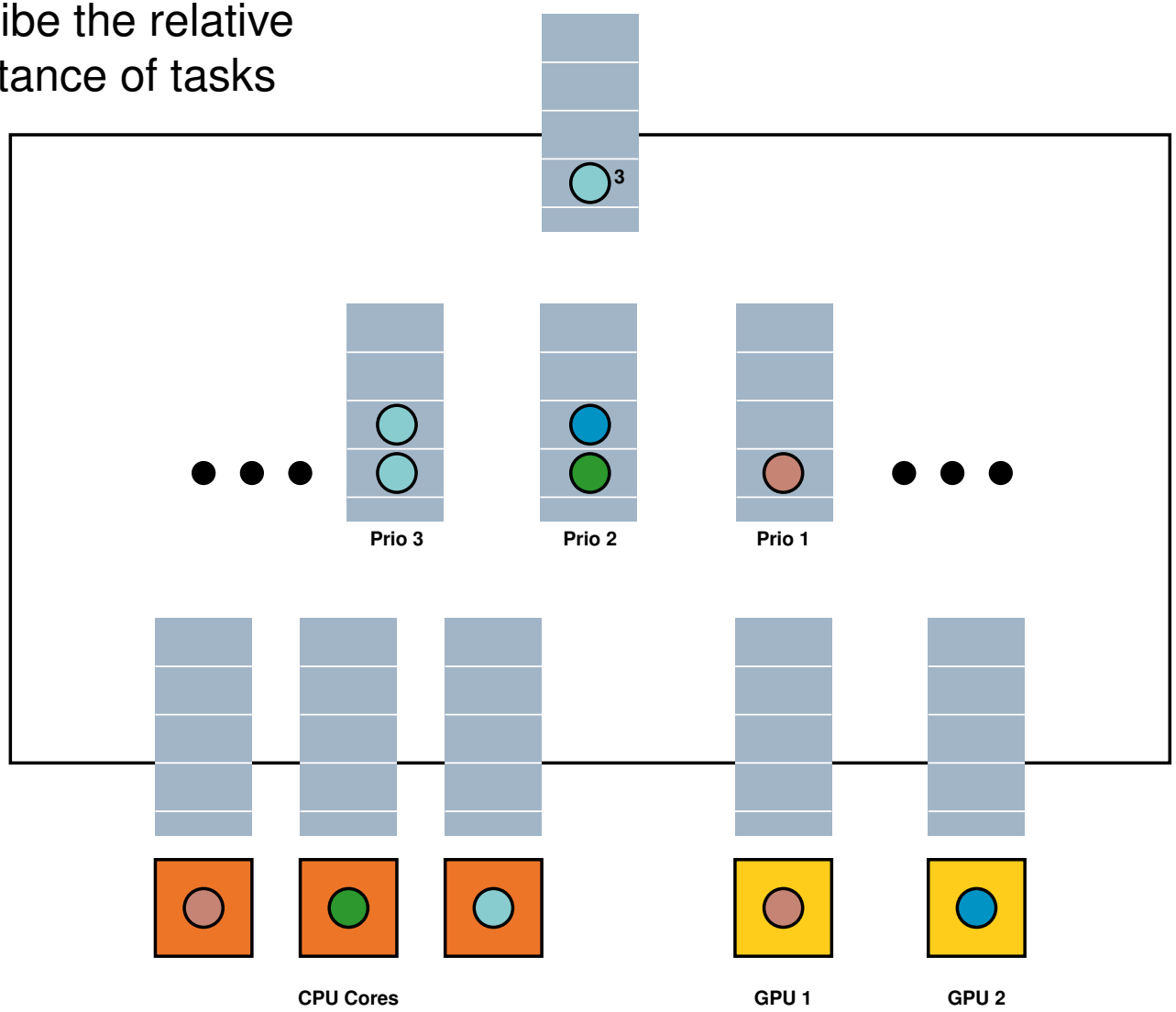
The Prio Scheduler

- Describe the relative importance of tasks



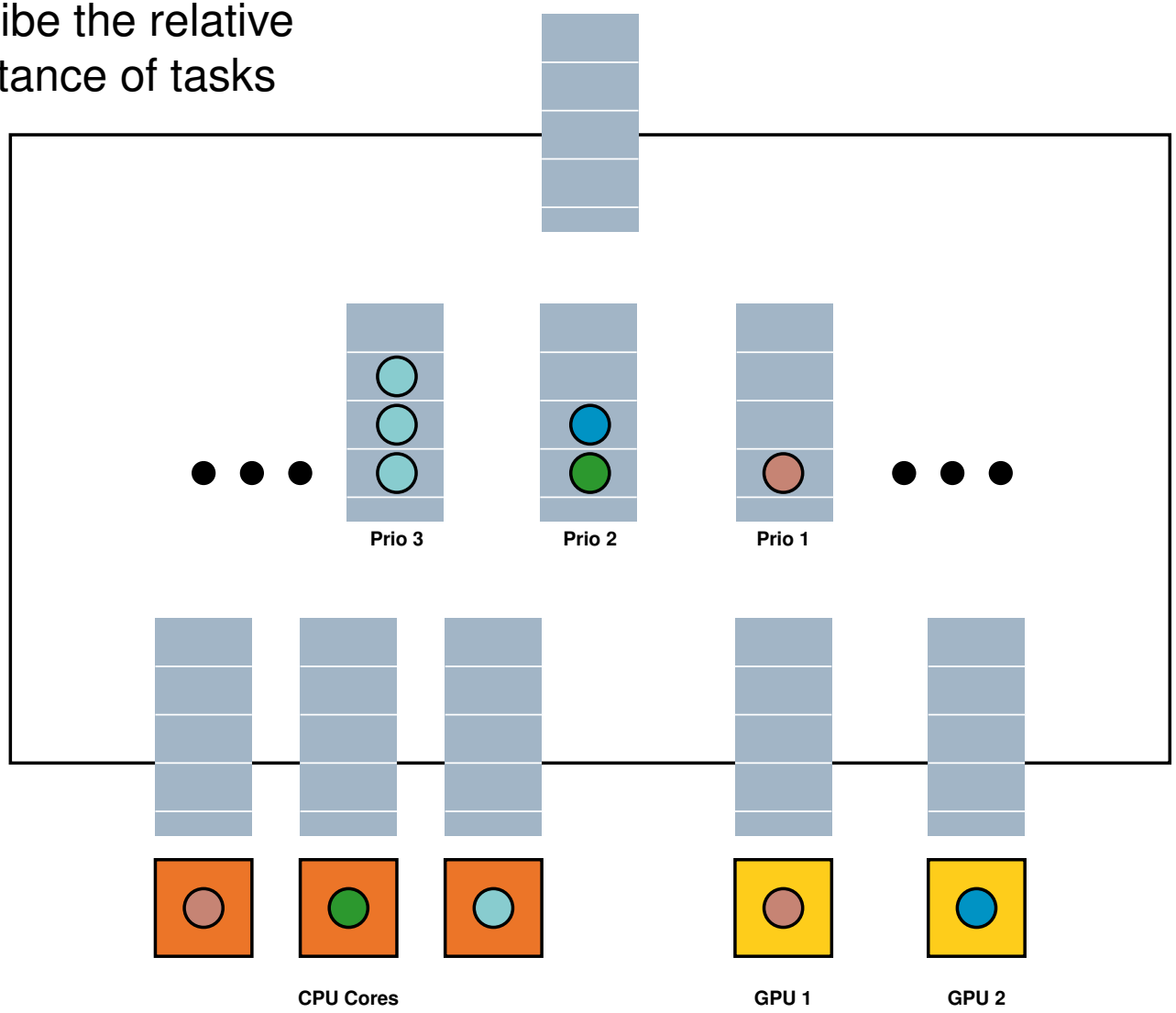
The Prio Scheduler

- Describe the relative importance of tasks



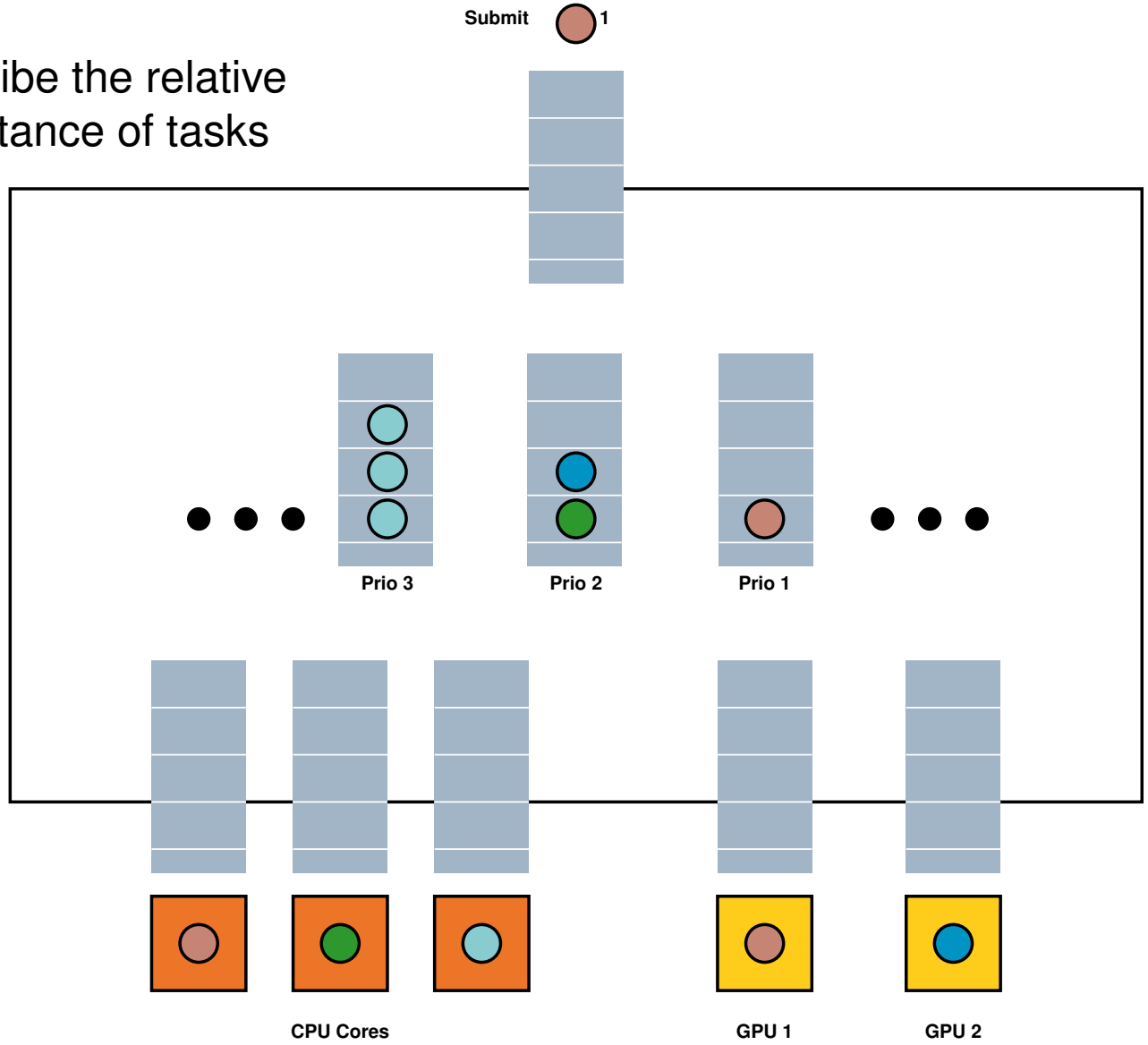
The Prio Scheduler

- Describe the relative importance of tasks



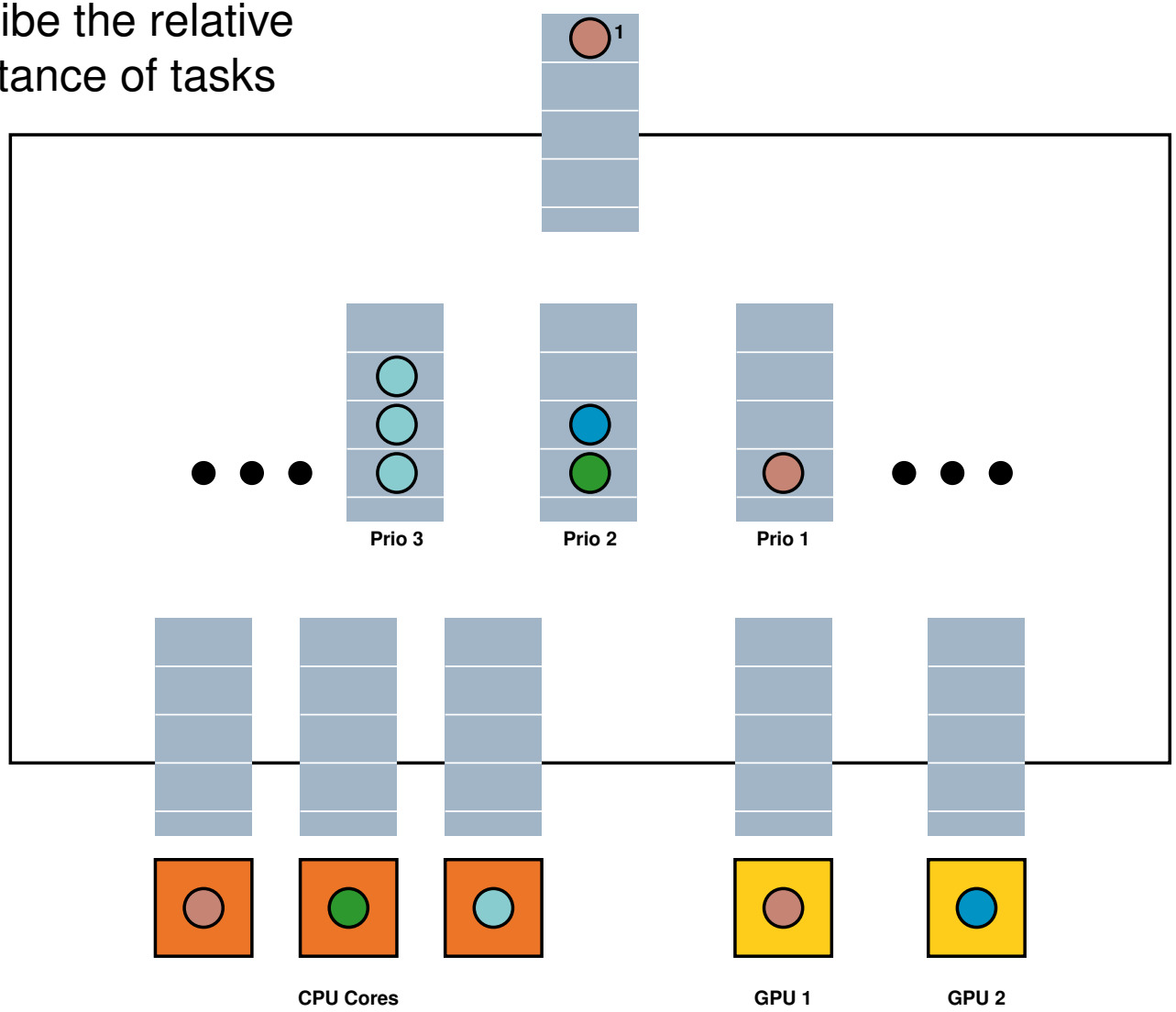
The Prio Scheduler

- Describe the relative importance of tasks



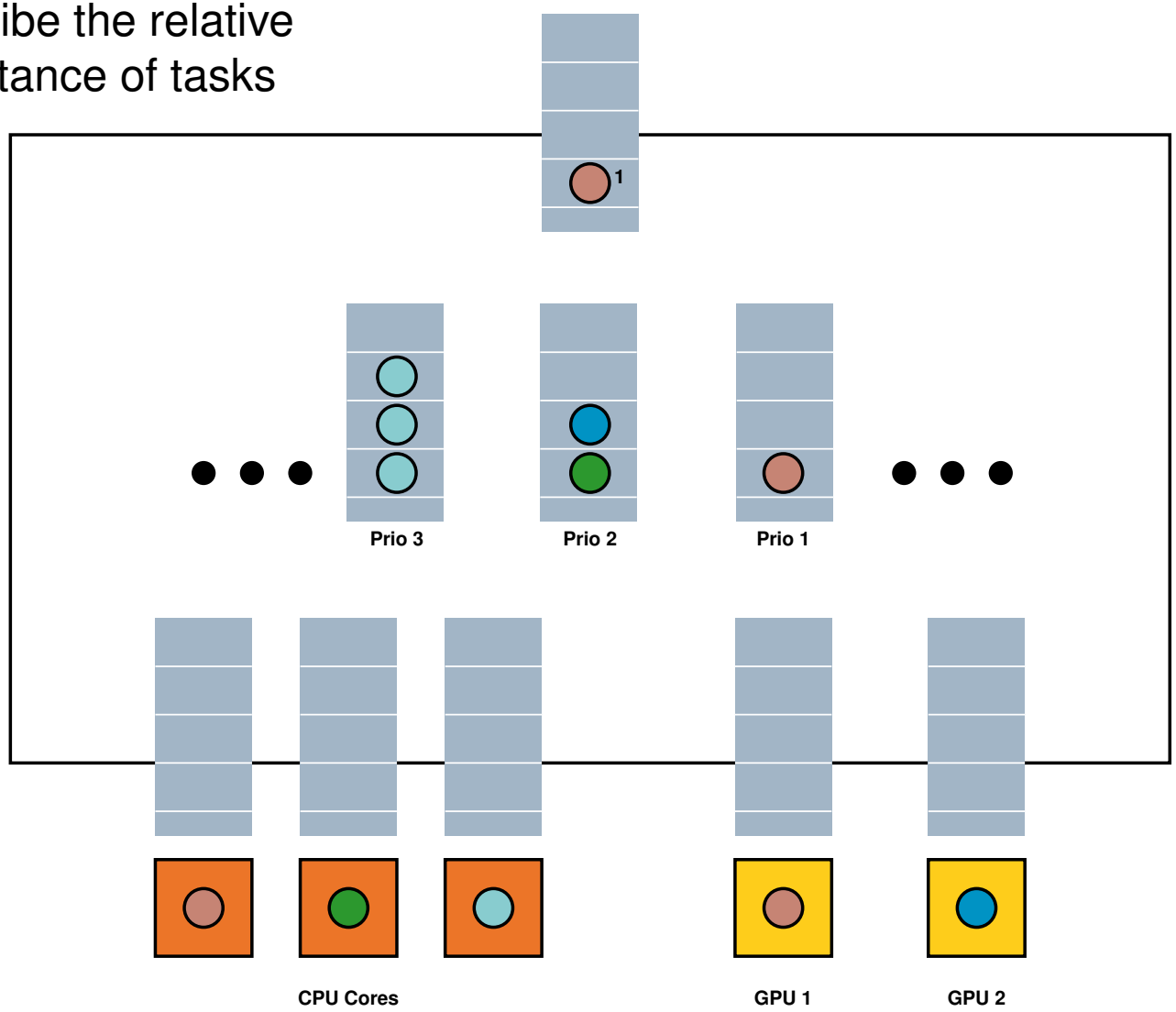
The Prio Scheduler

- Describe the relative importance of tasks



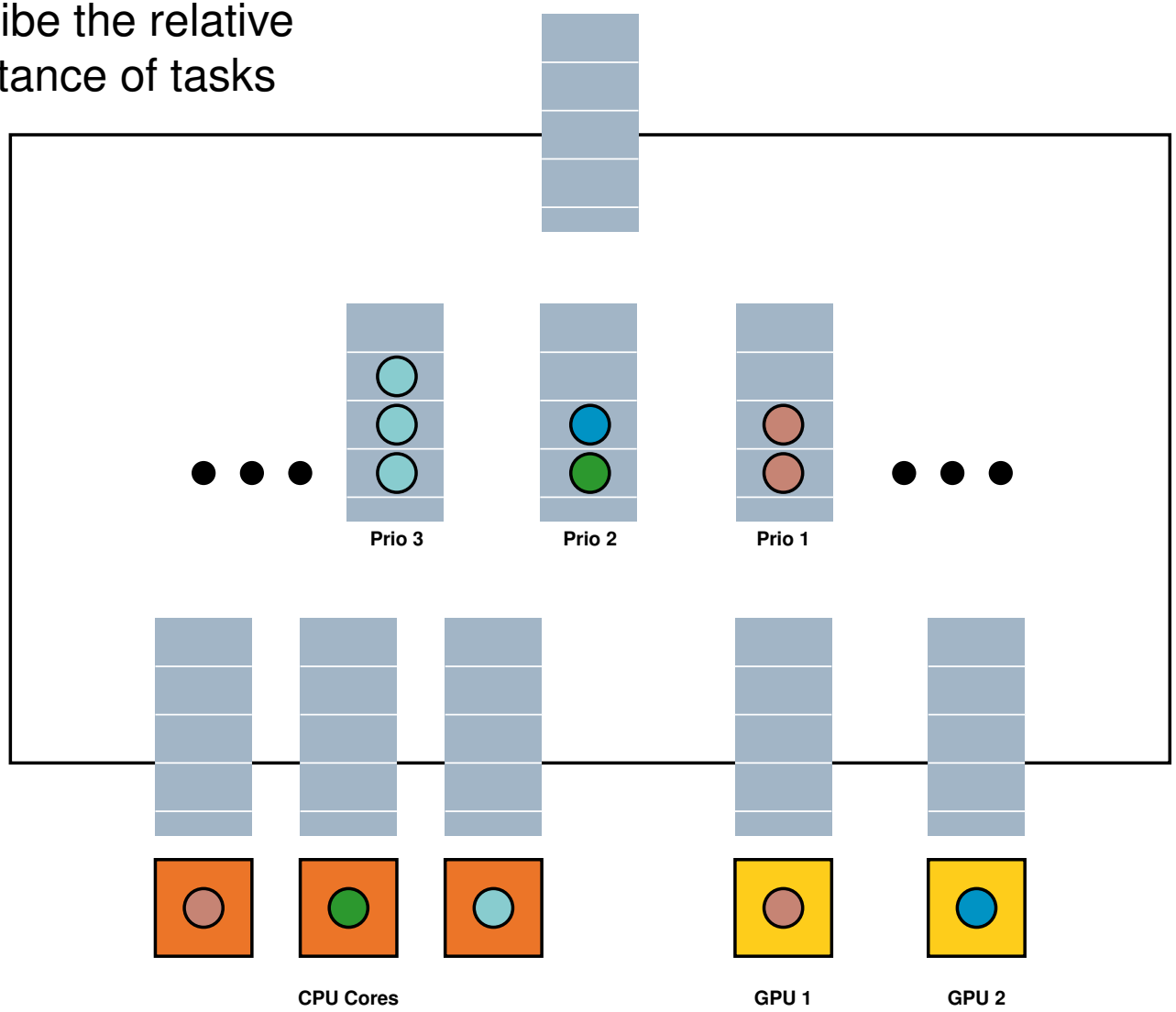
The Prio Scheduler

- Describe the relative importance of tasks



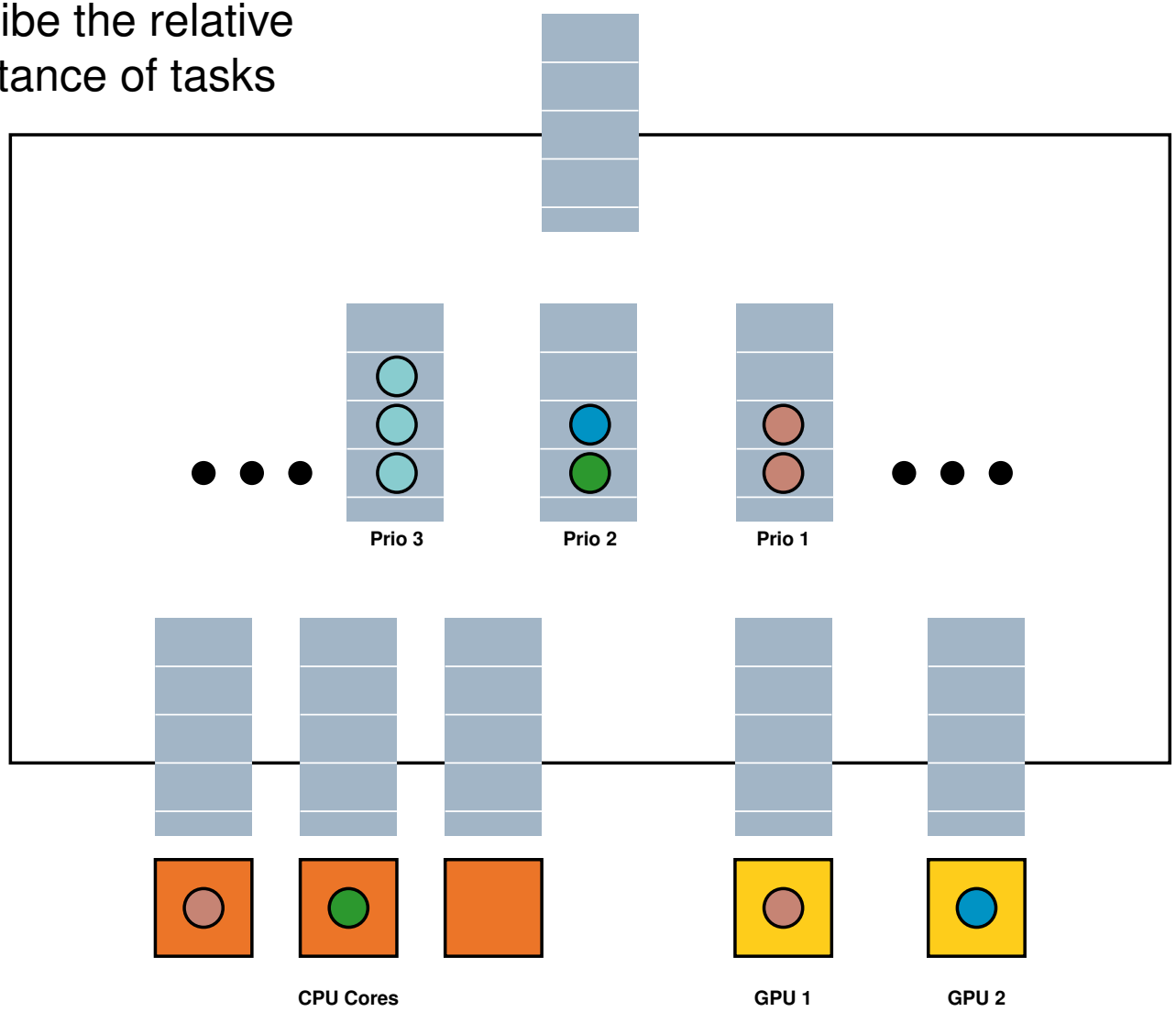
The Prio Scheduler

- Describe the relative importance of tasks



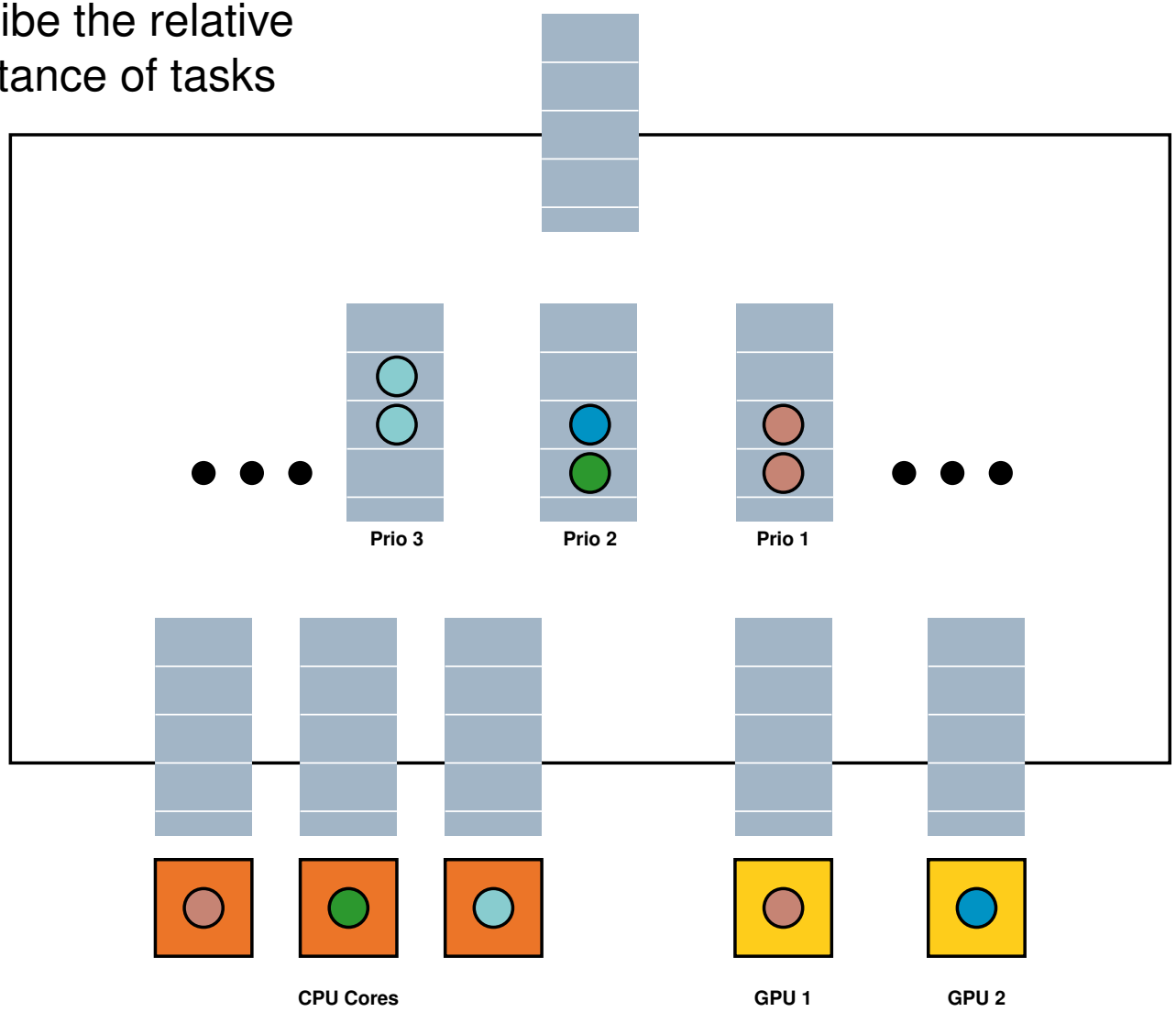
The Prio Scheduler

- Describe the relative importance of tasks



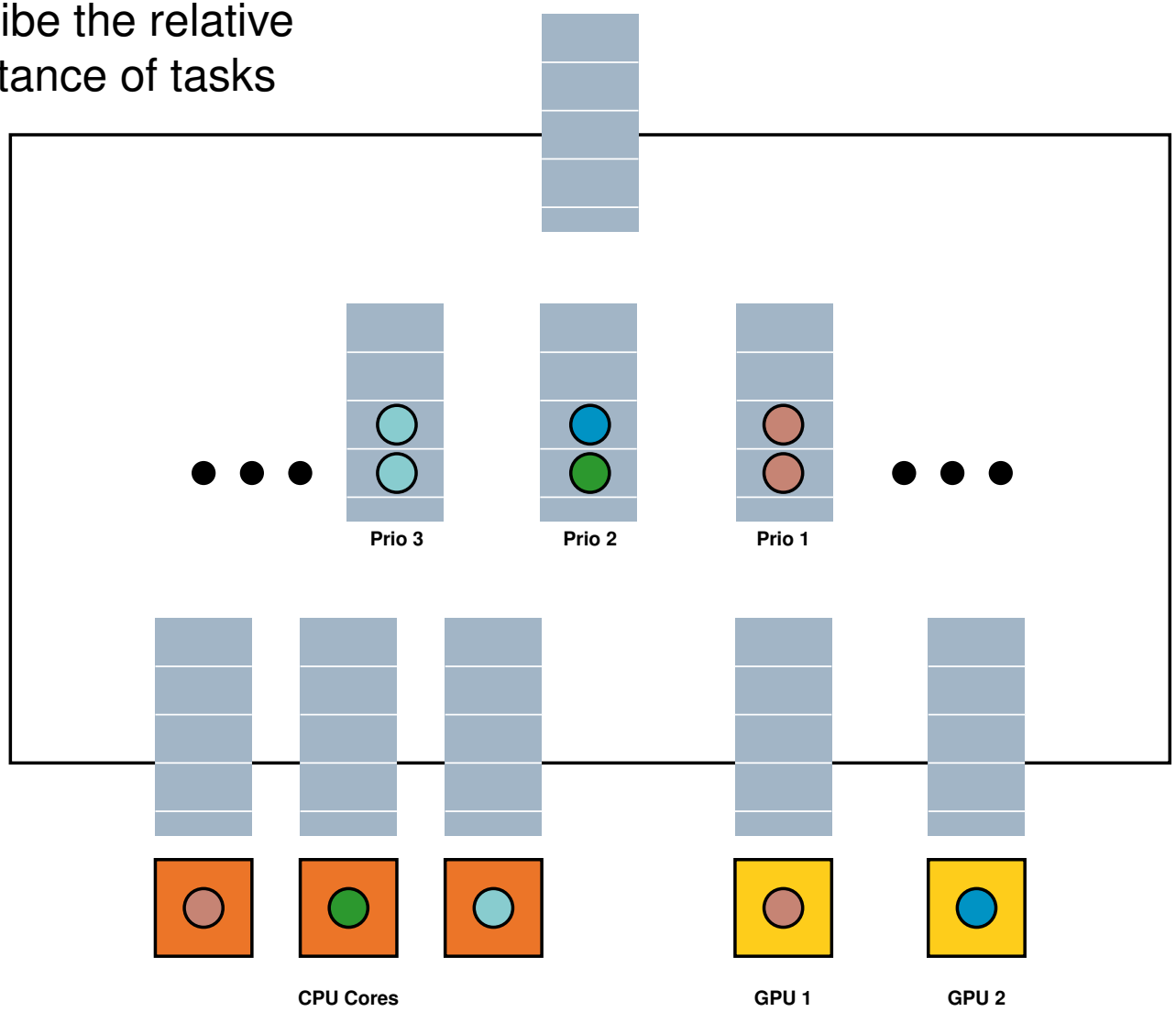
The Prio Scheduler

- Describe the relative importance of tasks



The Prio Scheduler

- Describe the relative importance of tasks



The Deque Model (dm) Scheduler

- Inspired by HEFT popular scheduling algorithm
  - Heterogeneous Earliest Finish Time
- Try to get the best from accelerators **and** CPUs

The Deque Model (dm) Scheduler

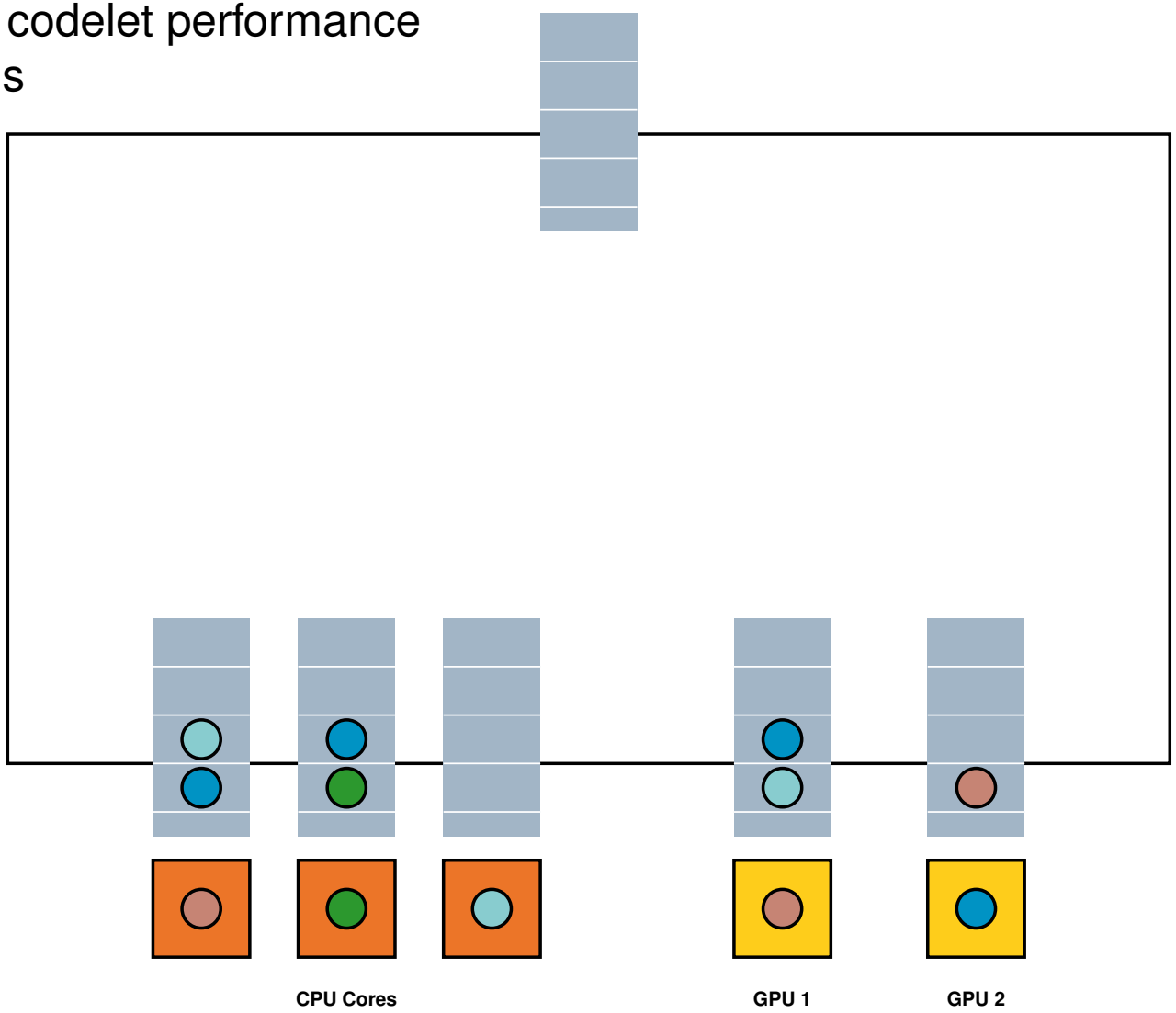
- Inspired by HEFT popular scheduling algorithm
  - Heterogeneous Earliest Finish Time
- Try to get the best from accelerators **and** CPUs
- Using codelet performance models
  - Kernel calibration on each available computing device
  - **Raw** history model of kernels' past execution times
  - **Refined** models using regression on kernels' execution times history

The Deque Model (dm) Scheduler

- Inspired by HEFT popular scheduling algorithm
  - Heterogeneous Earliest Finish Time
- Try to get the best from accelerators **and** CPUs
- Using codelet performance models
  - Kernel calibration on each available computing device
  - **Raw** history model of kernels' past execution times
  - **Refined** models using regression on kernels' execution times history
- Model parameter
  - **Data size** by default
  - **User-defined** for more complex cases
    - Sparse data structures
    - Iteratives kernels

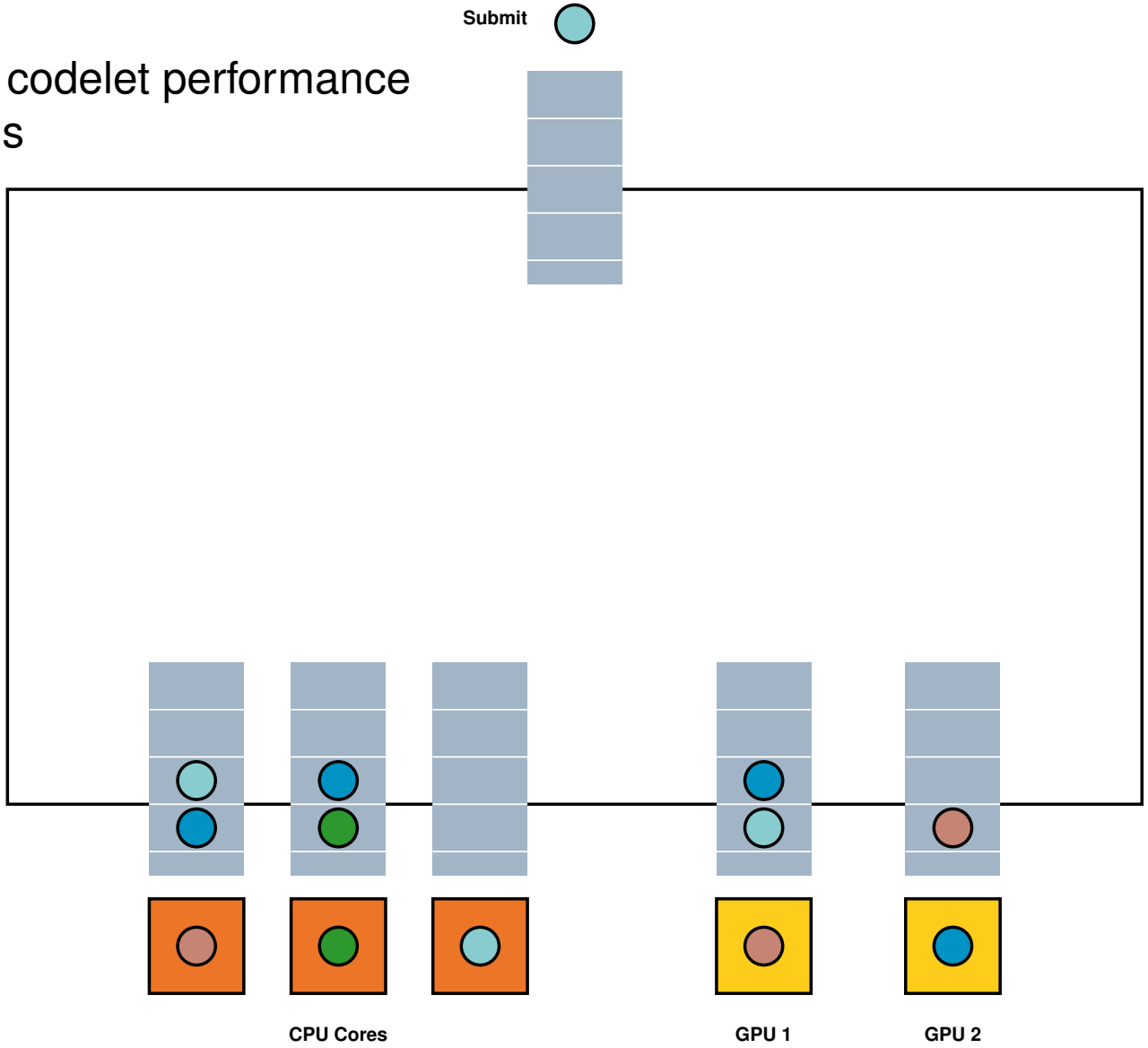
The Deque Model (dm) Scheduler

- Using codelet performance models



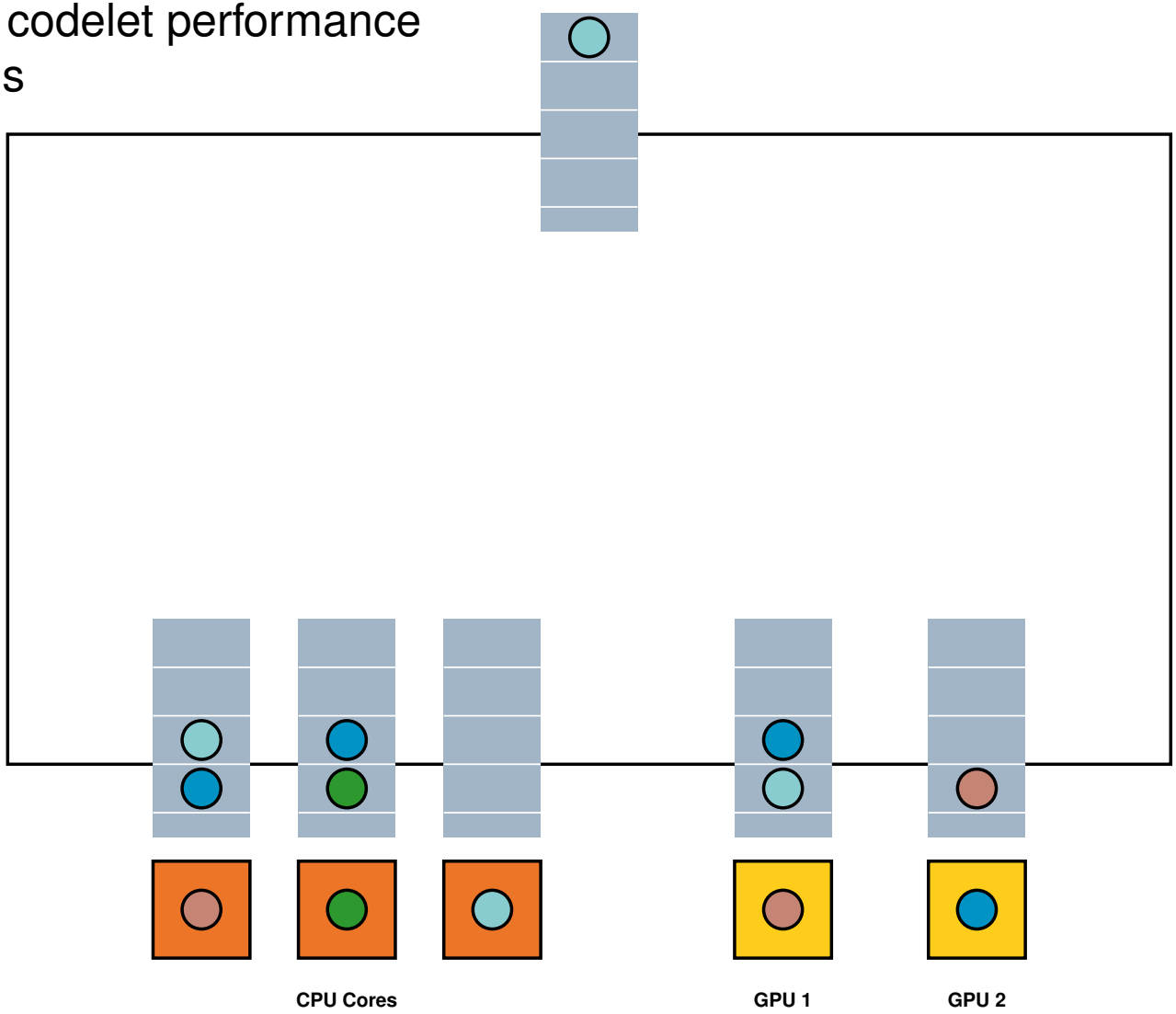
The Deque Model (dm) Scheduler

- Using codelet performance models



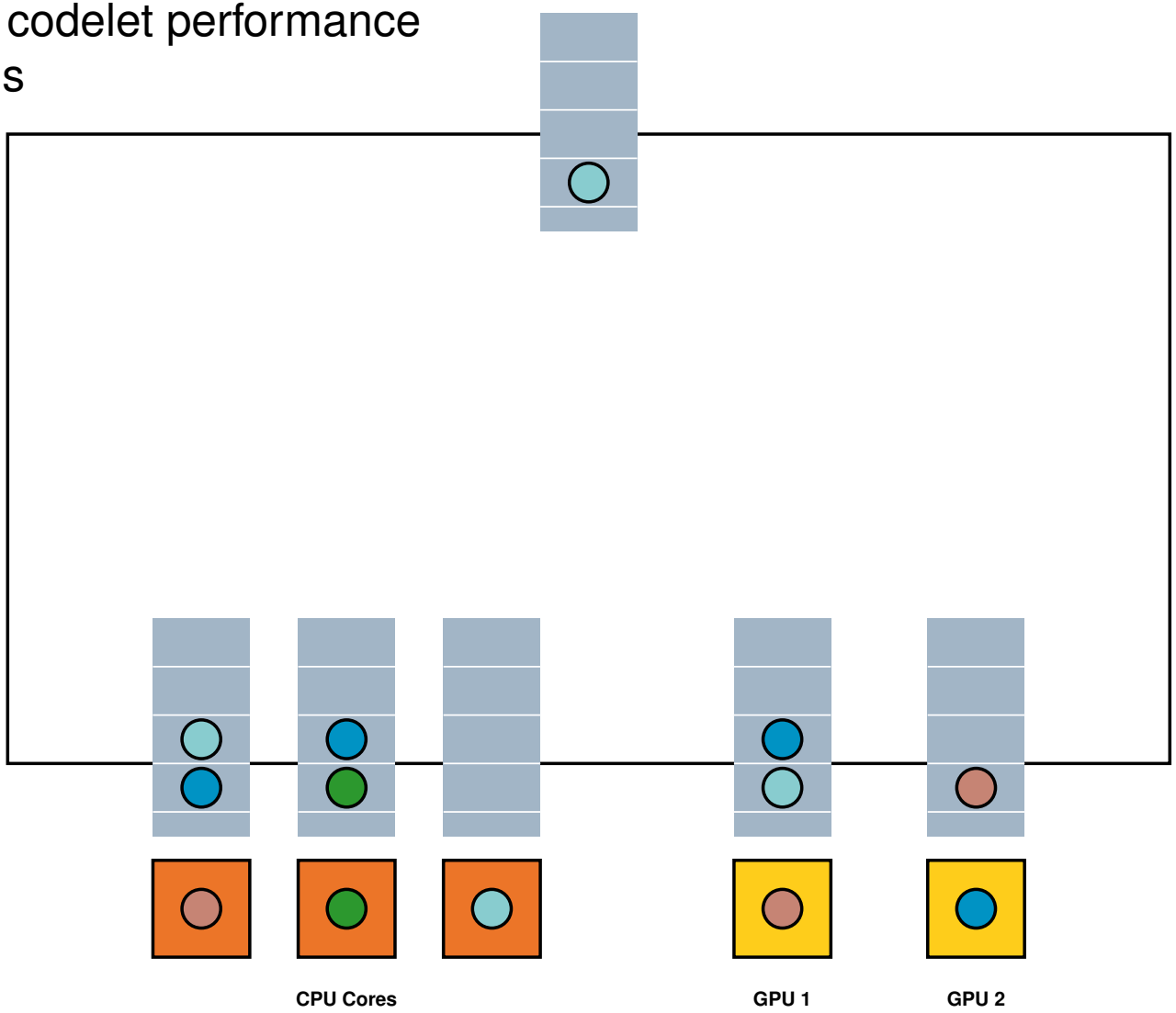
The Deque Model (dm) Scheduler

- Using codelet performance models



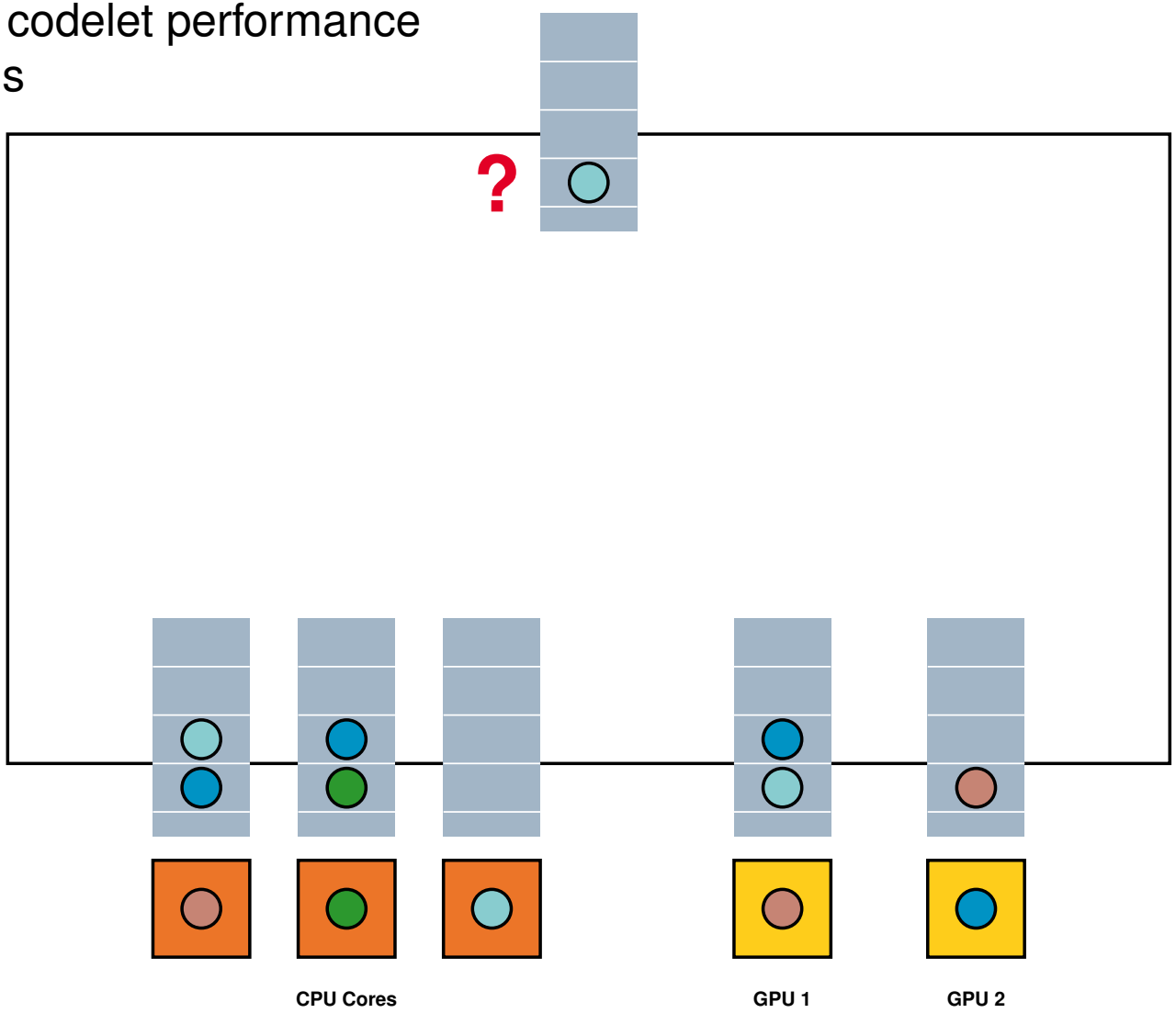
The Deque Model (dm) Scheduler

- Using codelet performance models



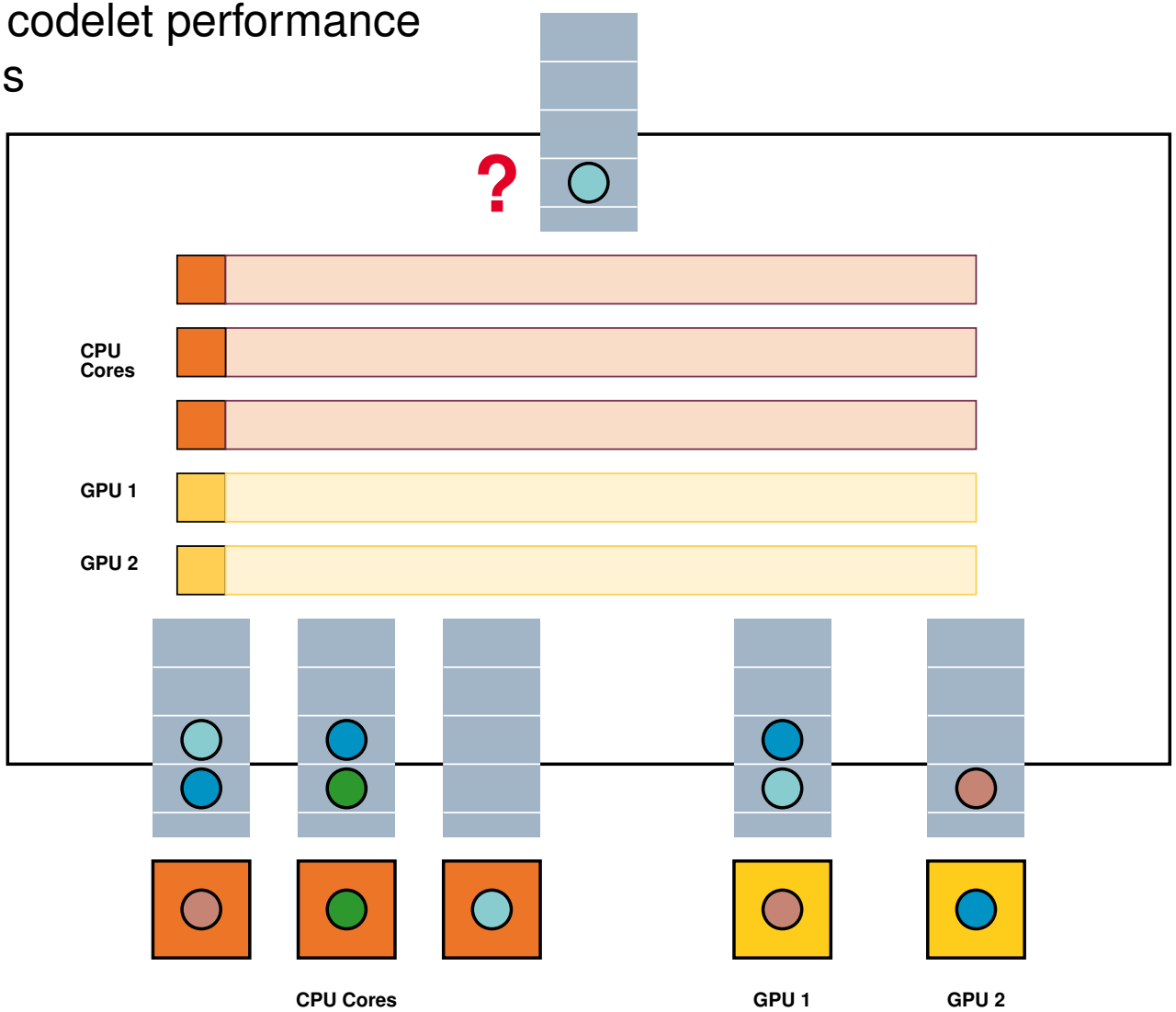
The Deque Model (dm) Scheduler

- Using codelet performance models



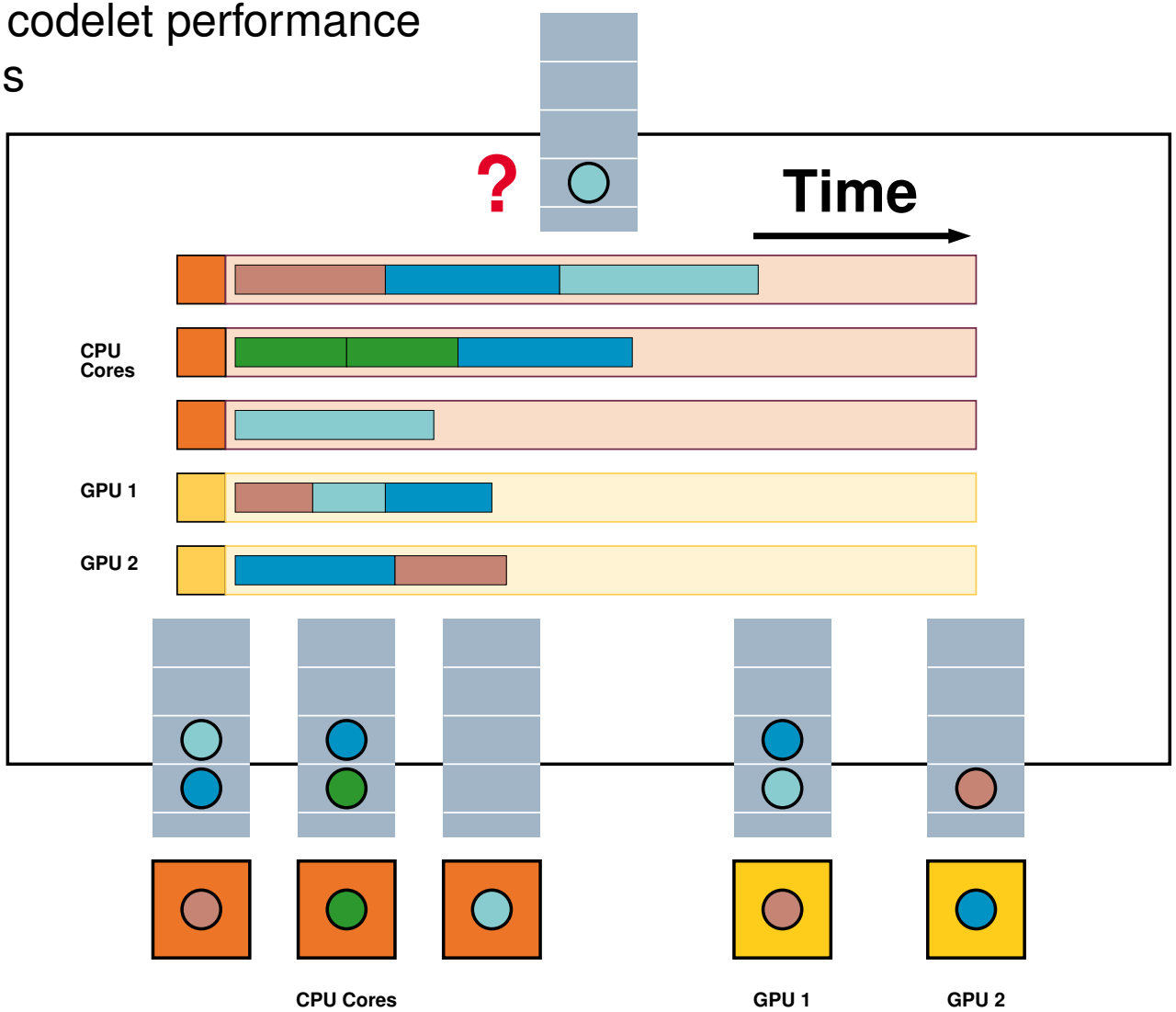
The Deque Model (dm) Scheduler

- Using codelet performance models



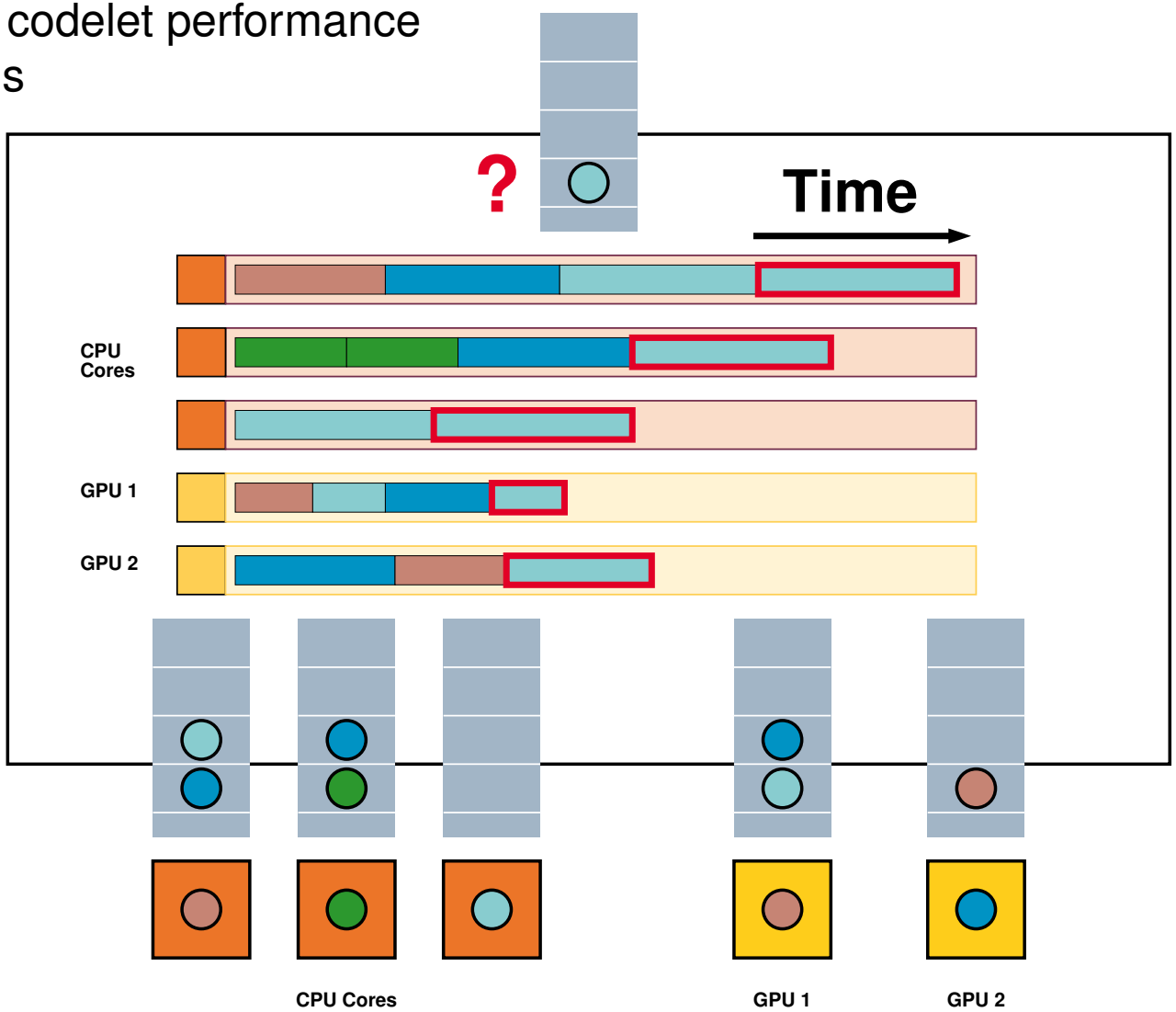
The Deque Model (dm) Scheduler

- Using codelet performance models



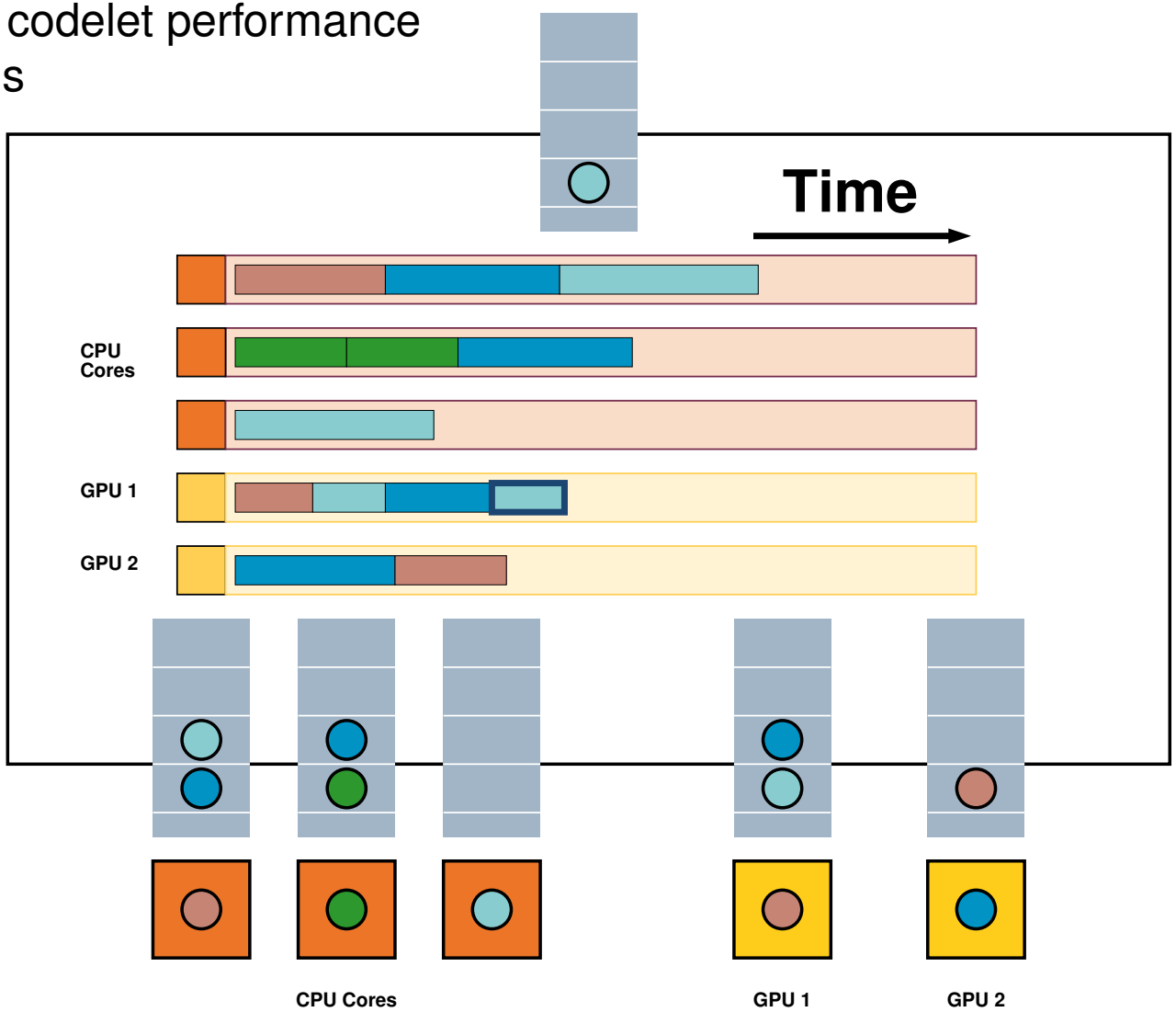
The Deque Model (dm) Scheduler

- Using codelet performance models



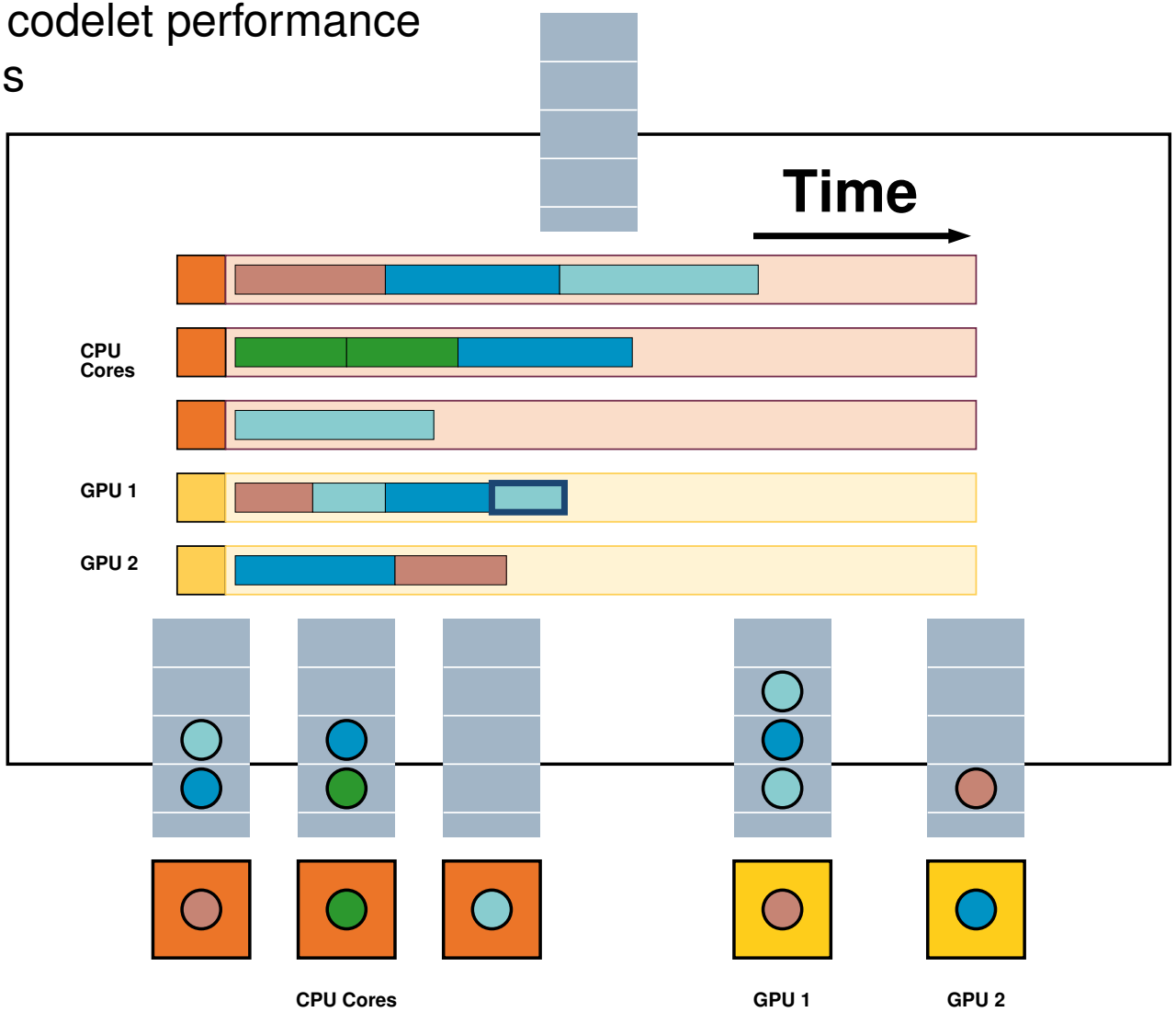
The Deque Model (dm) Scheduler

- Using codelet performance models



The Deque Model (dm) Scheduler

- Using codelet performance models



Selecting a Scheduling Policy

- Use the `STARPU_SCHED` environment variable

Selecting a Scheduling Policy

- Use the `STARPU_SCHED` environment variable
- Example 1: selecting the prio scheduler

```
1 $ export STARPU_SCHED=prio
2 $ my_program
3 ...
```

Selecting a Scheduling Policy

- Use the `STARPU_SCHED` environment variable
- Example 1: selecting the prio scheduler
- Example 2: selecting the dm scheduler

```
1 $ export STARPU_SCHED=prio
2 $ my_program
3 ...
```

```
1 $ export STARPU_SCHED=dm
2 $ my_program
3 ...
```

Selecting a Scheduling Policy

- Use the `STARPU_SCHED` environment variable
- Example 1: selecting the prio scheduler
- Example 2: selecting the dm scheduler
- Example 3: resetting to default scheduler eager

```
$ export STARPU_SCHED=prio
$ my_program
...
```

```
$ export STARPU_SCHED=dm
$ my_program
...
```

```
$ unset STARPU_SCHED
$ my_program
...
```

Selecting a Scheduling Policy

- Use the `STARPU_SCHED` environment variable
- Example 1: selecting the prio scheduler
- Example 2: selecting the dm scheduler
- Example 3: resetting to default scheduler eager
- No need to recompile the application

```
$ export STARPU_SCHED=prio
$ my_program
...
```

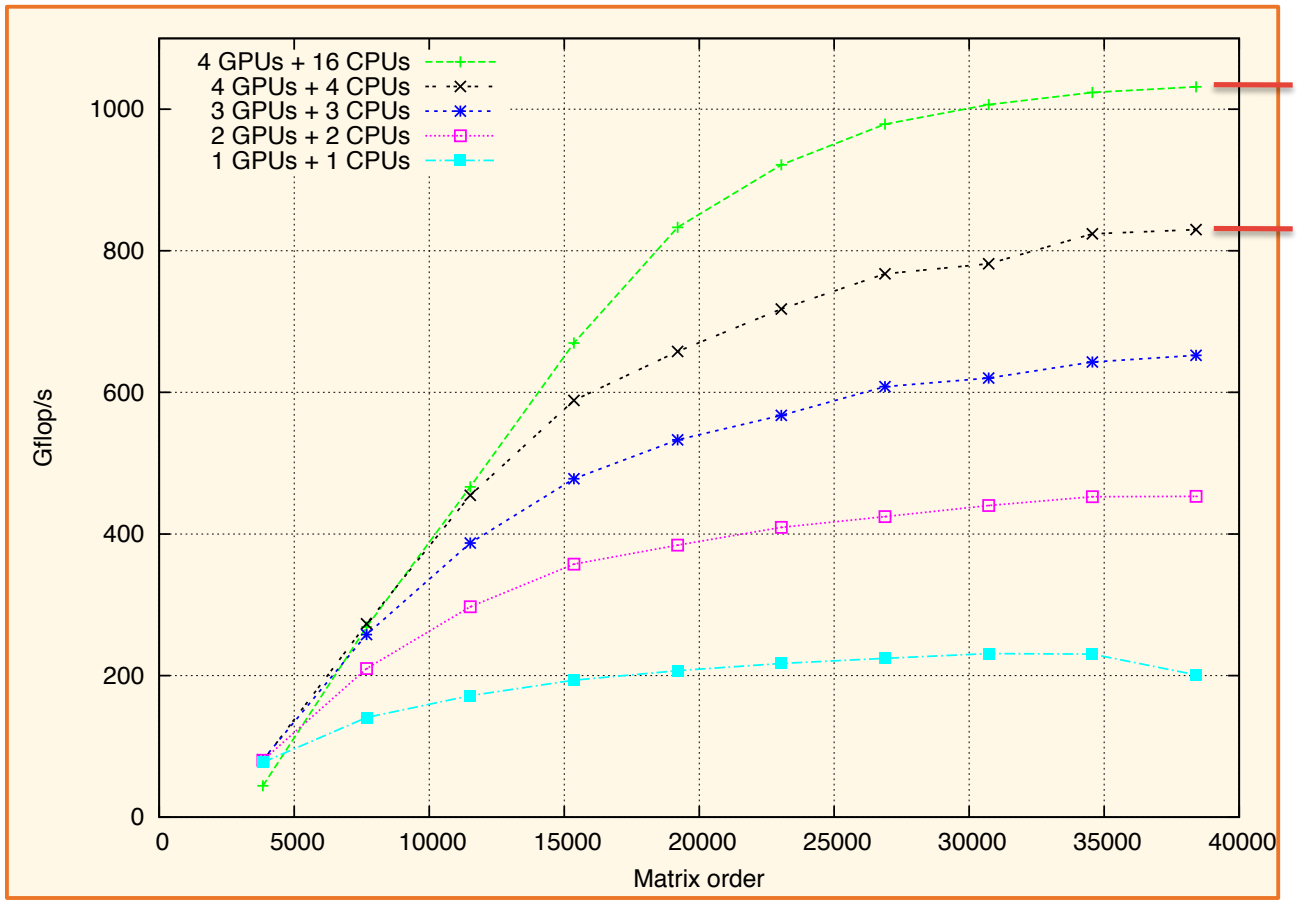
```
$ export STARPU_SCHED=dm
$ my_program
...
```

```
$ unset STARPU_SCHED
$ my_program
...
```

Showcase with the MAGMA Linear Algebra Library

University of Tennessee, INRIA HIEPACS, INRIA RUNTIME

- QR decomposition on 16 CPUs (AMD) + 4 GPUs (C1060)



Measured increase:
+12 CPUs
~200 GFlops

Expected increase:
+12 CPUs
~150 Gflops

Showcase with the MAGMA Linear Algebra Library

QR kernel properties

Kernel SGEQRT					
CPU:	9 GFlop/s	GPU:	30 GFlop/s	Speed-up:	3
Kernel STSQRT					
CPU:	12 GFlop/s	GPU:	37 GFlop/s	Speed-up:	3
Kernel SOMQRT					
CPU:	8.5 GFlop/s	GPU:	227 GFlop/s	Speed-up:	27
Kernel SSSMQ					
CPU:	10 GFlop/s	GPU:	285 GFlop/s	Speed-up:	28

Consequences

- Task distribution
  - SGEQRT: 20% Tasks on GPU
  - SSSMQ: 92% tasks on GPU
- Taking advantage of heterogeneity!
  - Only do what you are good for
  - Don't do what you are not good for

Beyond StarPU's Predefined Scheduling Policies

Predefined set of popular policies

- No *one size fits all* policy
- Selectable scheduling policy

Beyond StarPU's Predefined Scheduling Policies

Predefined set of popular policies

- No *one size fits all* policy
- Selectable scheduling policy

Extensible policy set

- You can write your own, specifically tailored policy
- Modular scheduler writing toolbox