Power-Aware Temporal Isolation with Variable-Bandwidth Servers

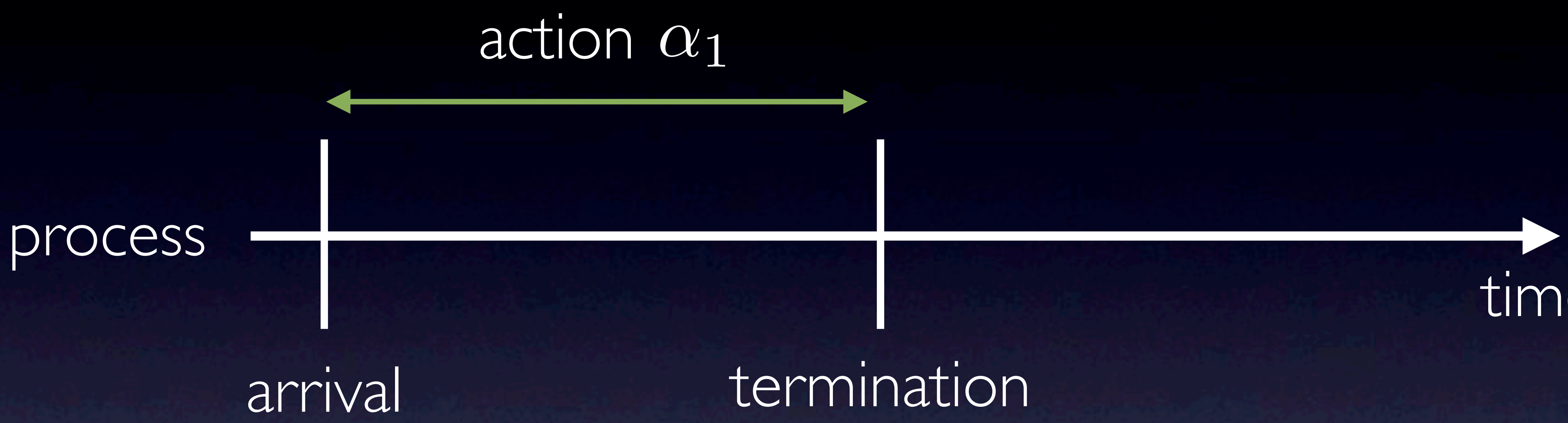
Silviu S. Craciunas, Christoph M. Kirsch, Ana Sokolova
Department of Computer Sciences
University of Salzburg

Process Model

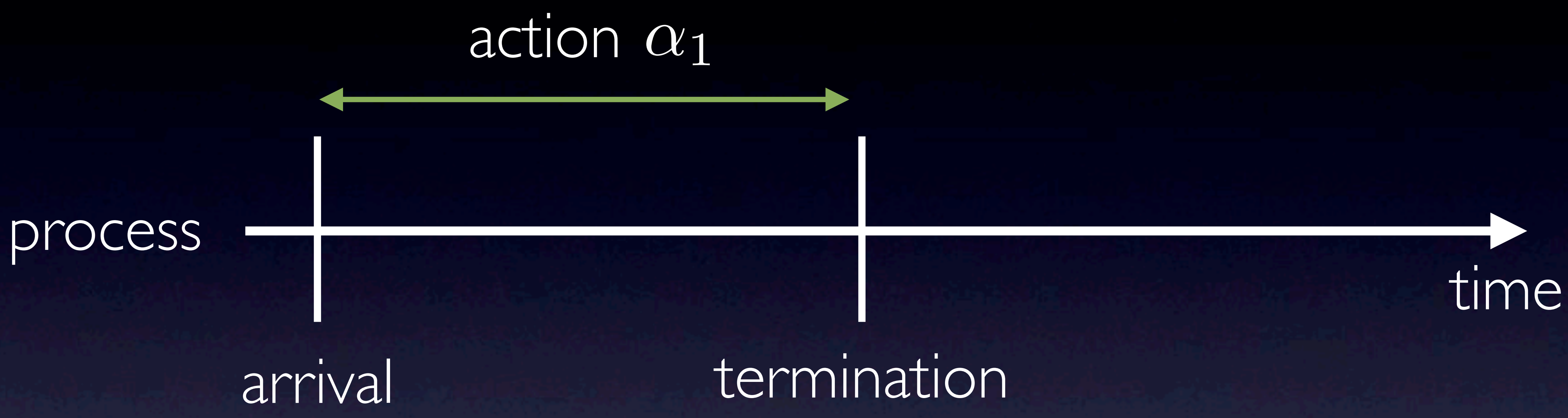
process

time

Process Model

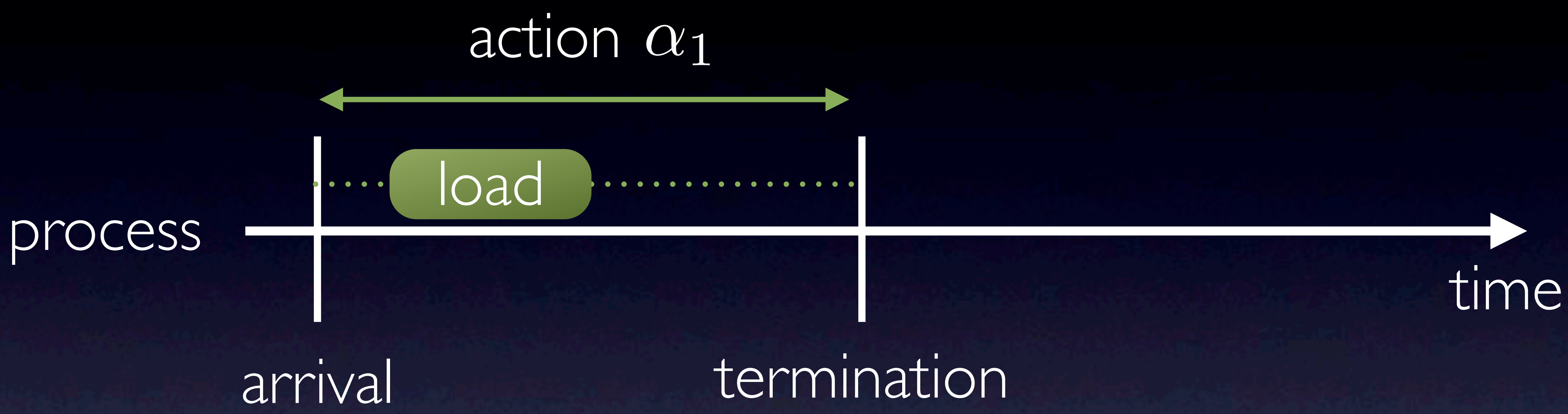


Process Model



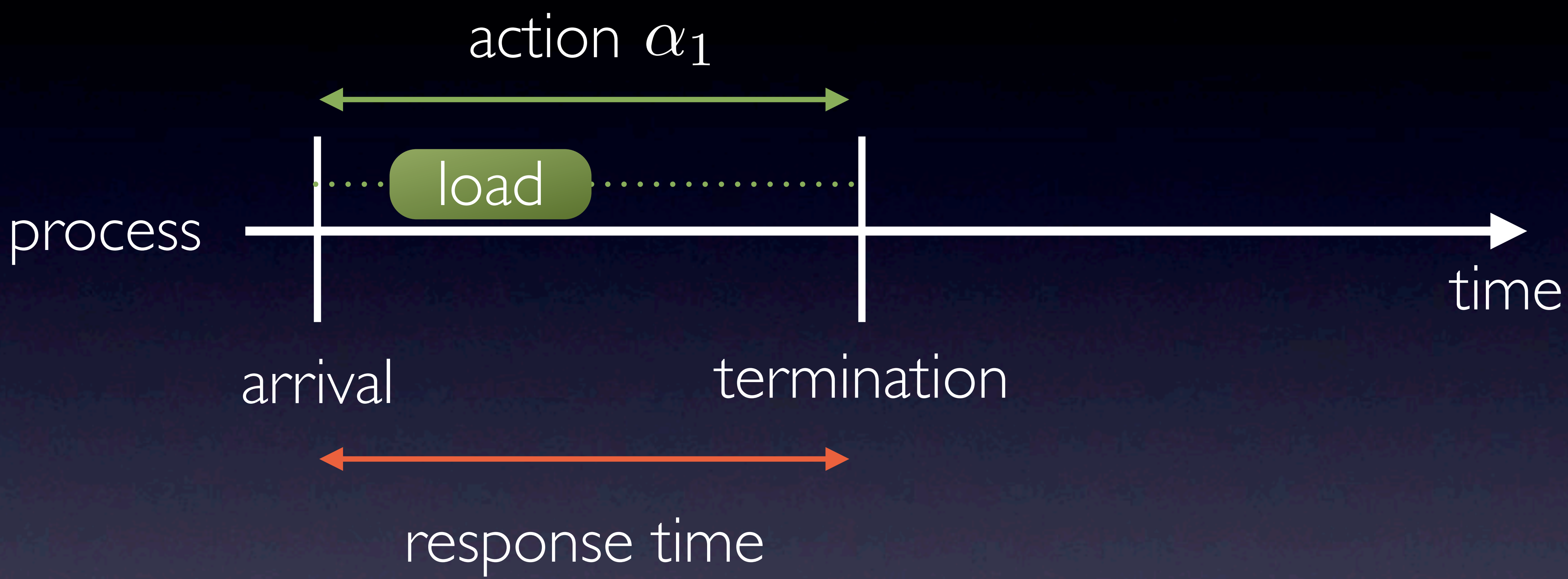
- action is a piece of code

Process Model



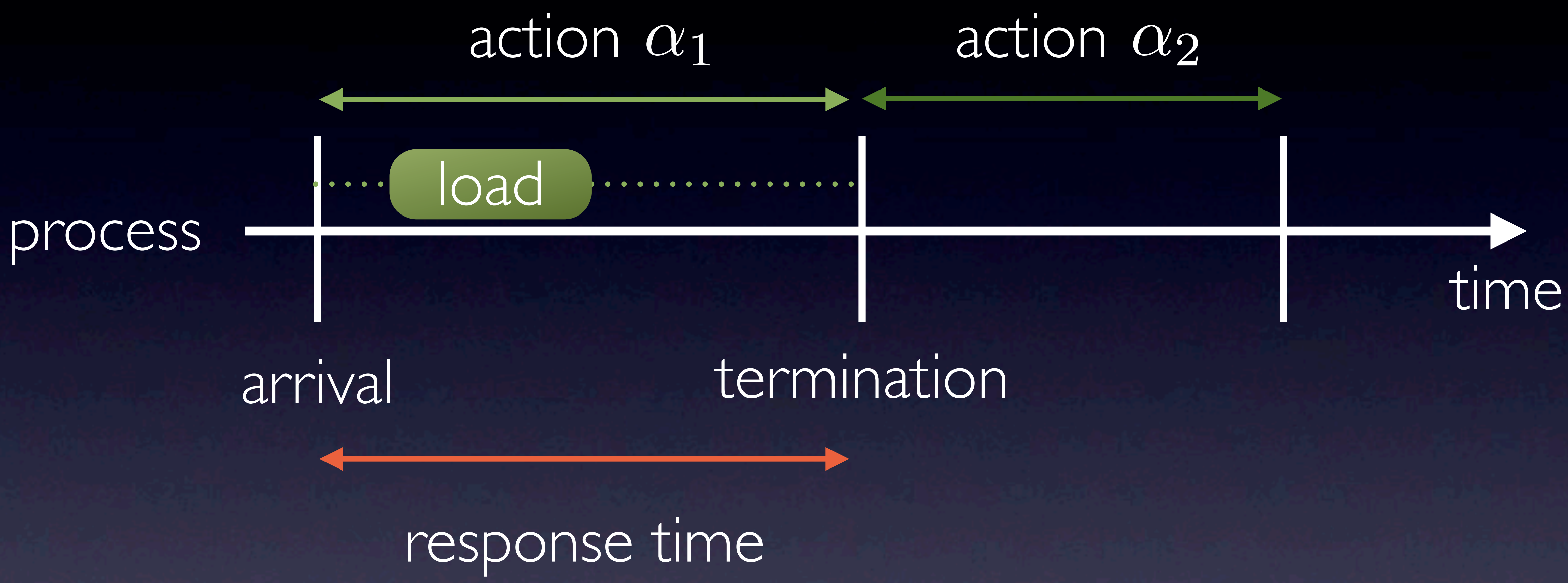
- action is a piece of code

Process Model



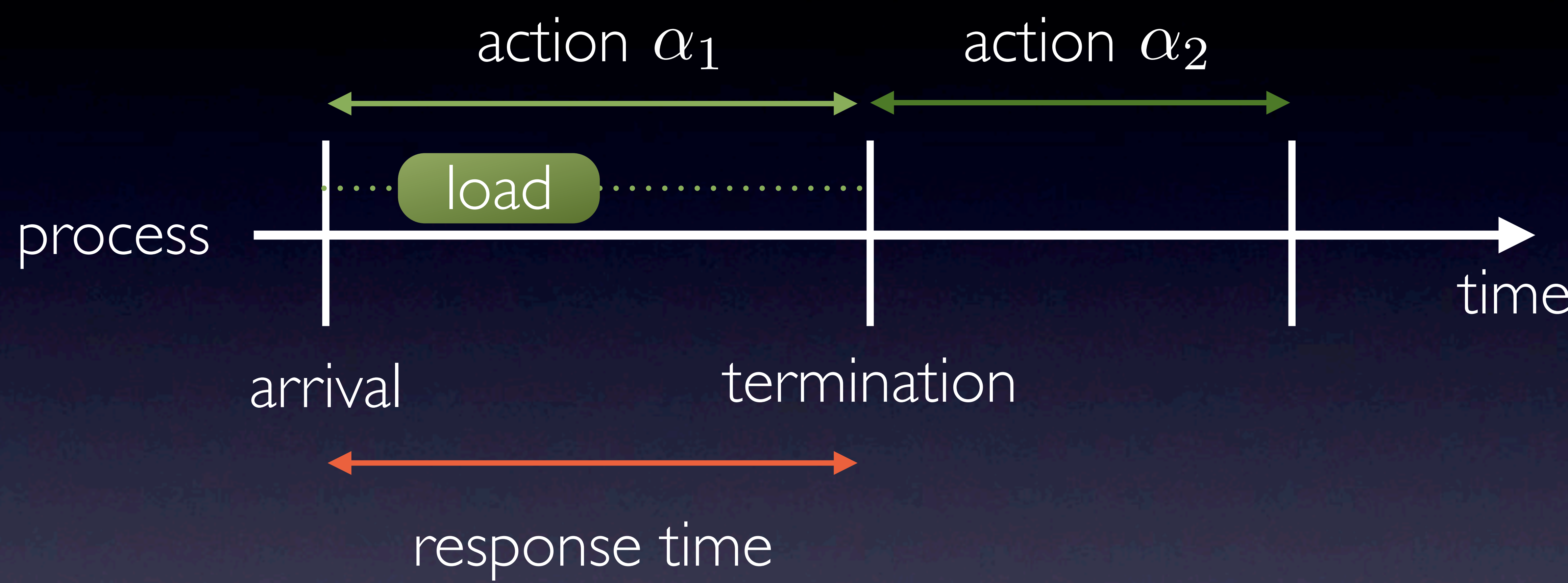
- action is a piece of code

Process Model



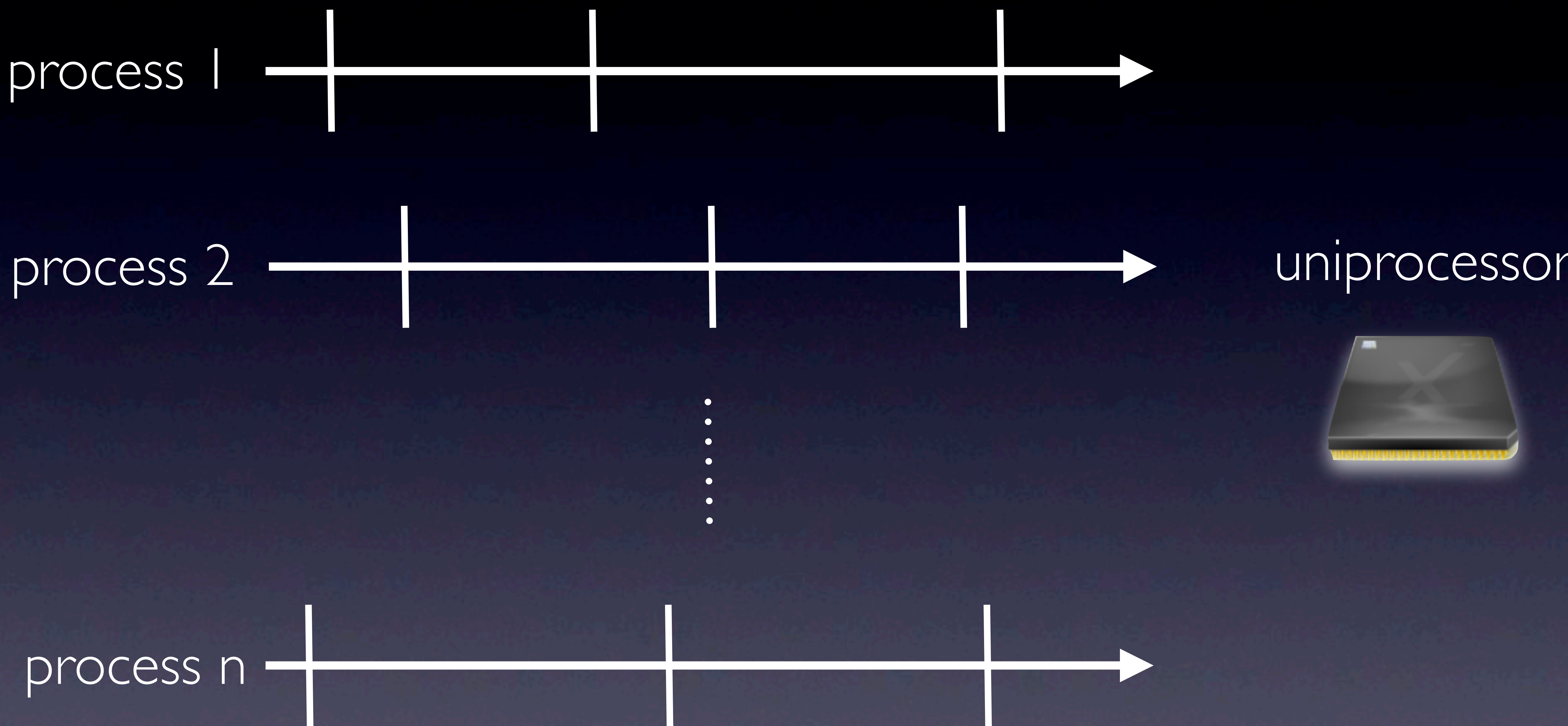
- action is a piece of code
- process is a sequence of actions

Process Model



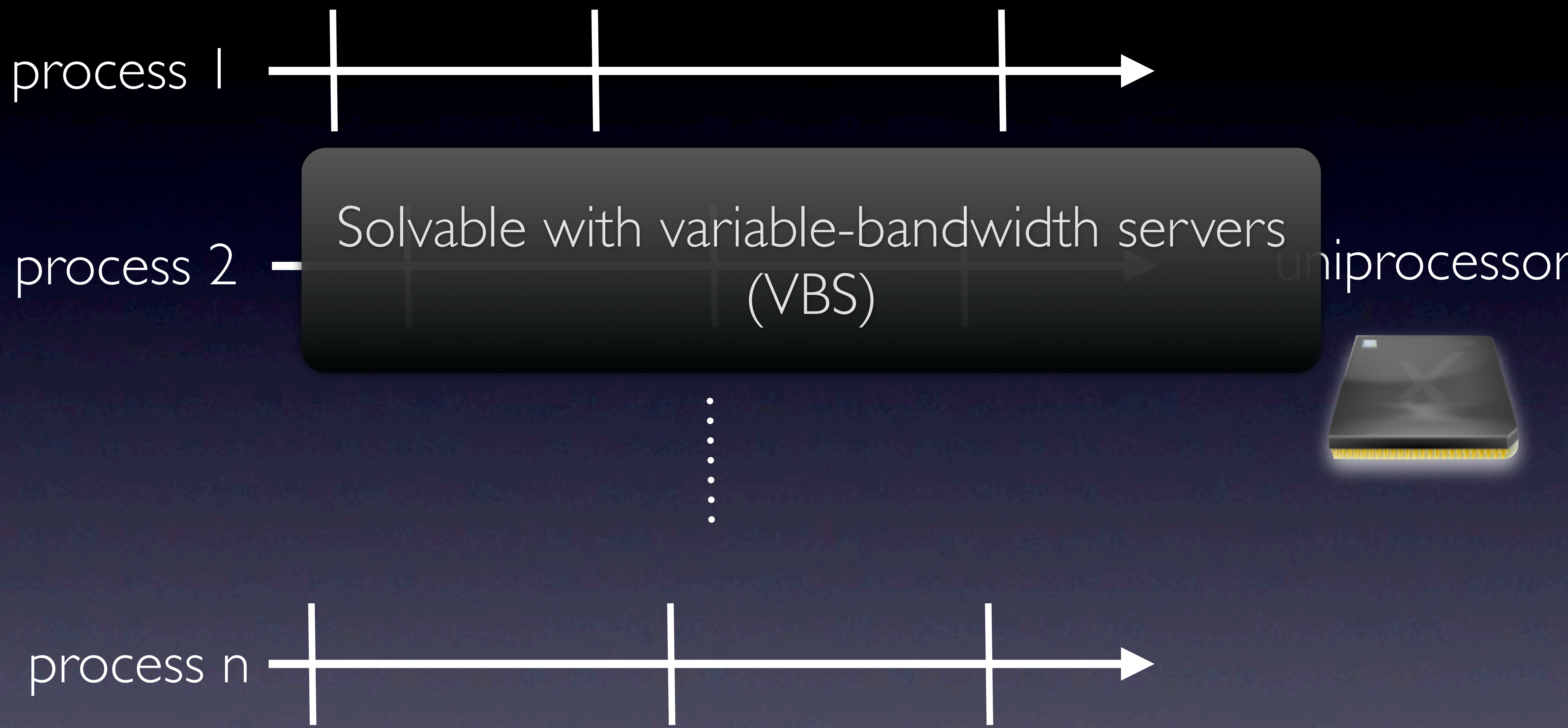
- action is a piece of code
- process is a sequence of actions
- throughput vs latency of process execution

Scheduling Problem



schedule the processes so that each of their actions maintains its response time

Scheduling Problem



Solvable with variable-bandwidth servers (VBS)

schedule the processes so that each of their actions maintains its response time

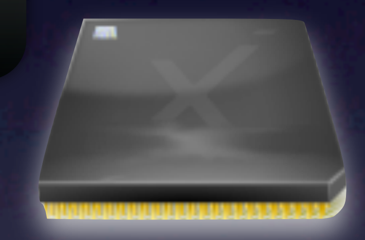
Scheduling Problem

process 1

Solvable with variable-bandwidth servers (VBS)

process 2

uniprocessor



Results [SIES09]:
- constant-time scheduling algorithm
- constant time admission test

process n

schedule the processes so that each of their actions maintains its response time

Resources and VBS

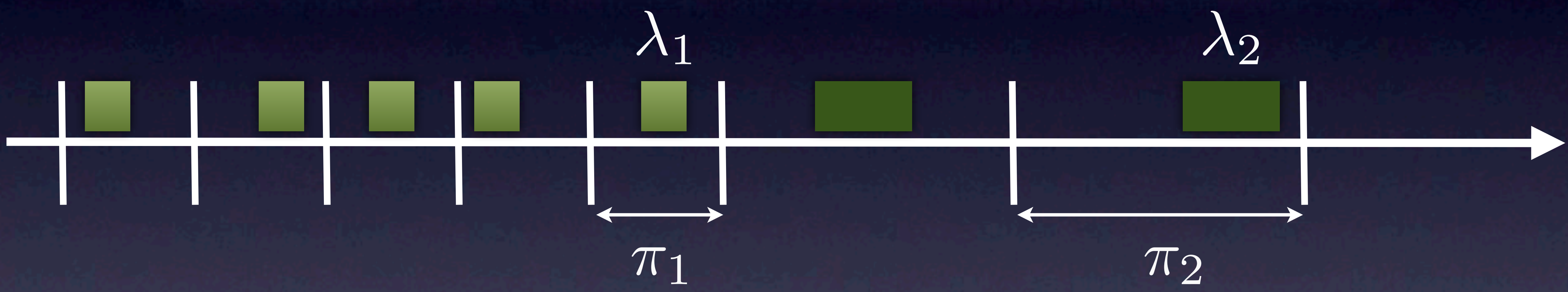
virtual periodic resources

period π limit λ utilization $\dfrac{\lambda}{\pi}$

Resources and VBS

virtual periodic resources

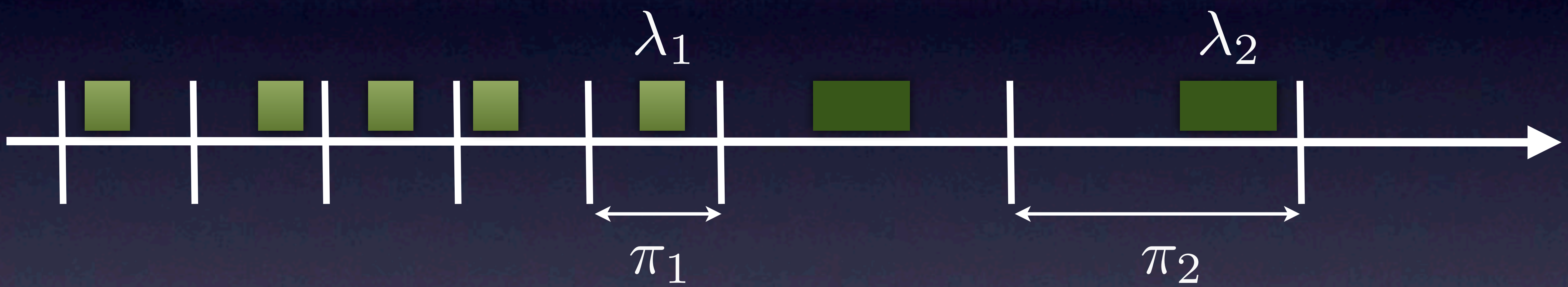
period π limit λ utilization $\frac{\lambda}{\pi}$



Resources and VBS

virtual periodic resources

period π limit λ utilization $\dfrac{\lambda}{\pi}$



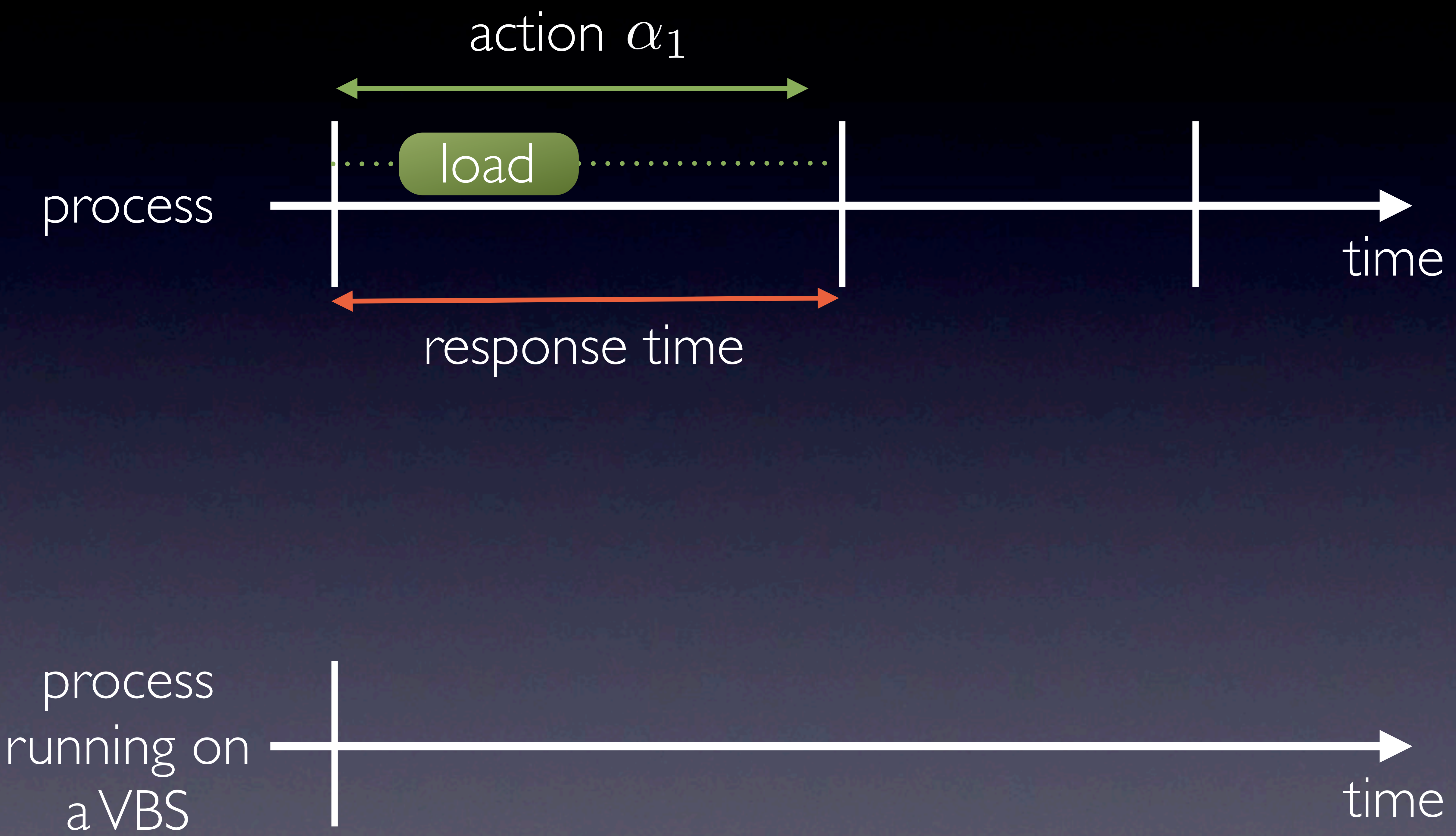
- VBS is determined by a bandwidth cap (u)
- VBS processes dynamically adjust speed (change resources)

$$\frac{\lambda_1}{\pi_1} \leq u \qquad \frac{\lambda_2}{\pi_2} \leq u$$

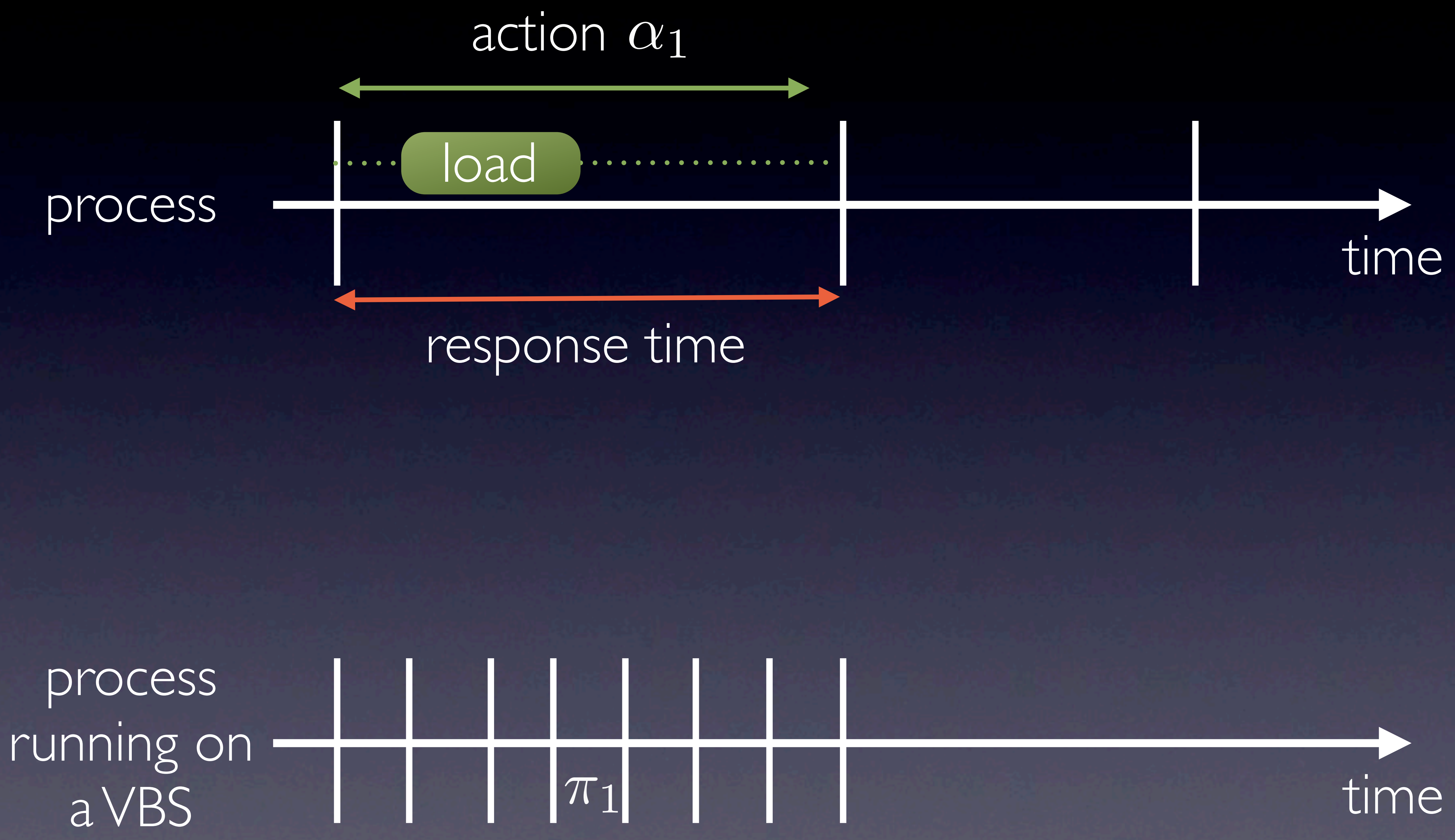
- generalization of constant bandwidth servers (CBS)

[Abeni and Buttazzo 2004]

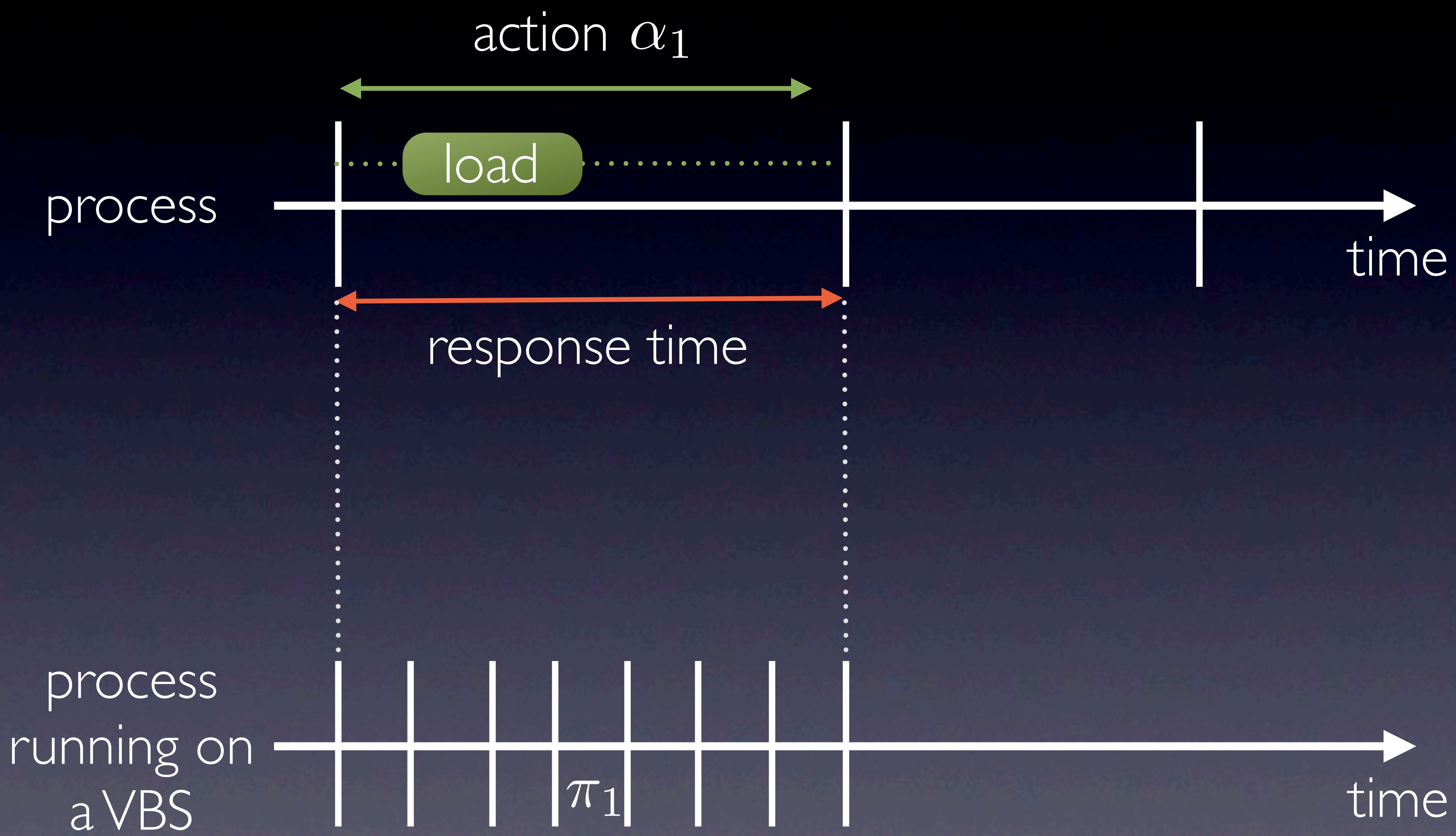
One Process on a VBS



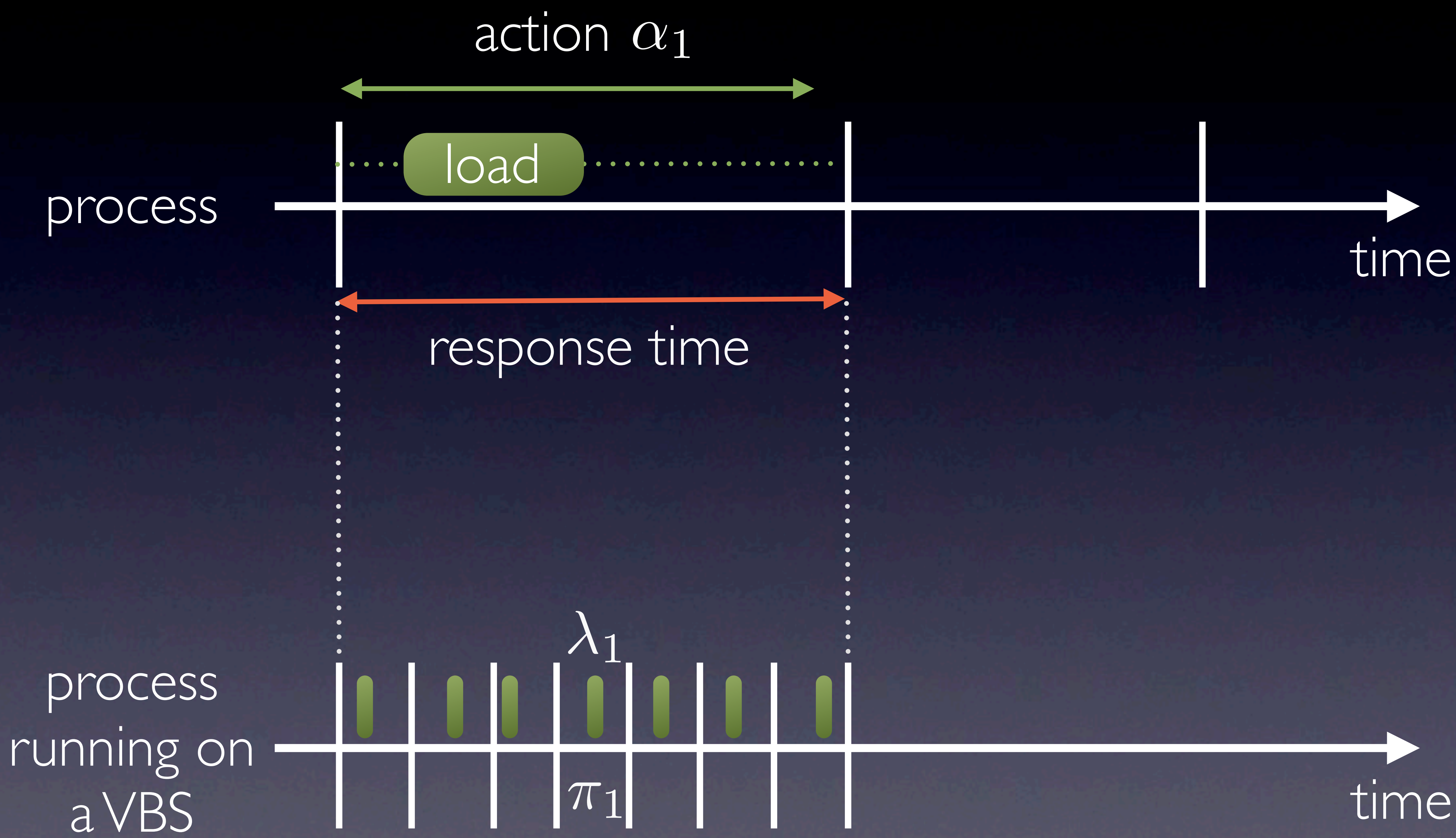
One Process on a VBS



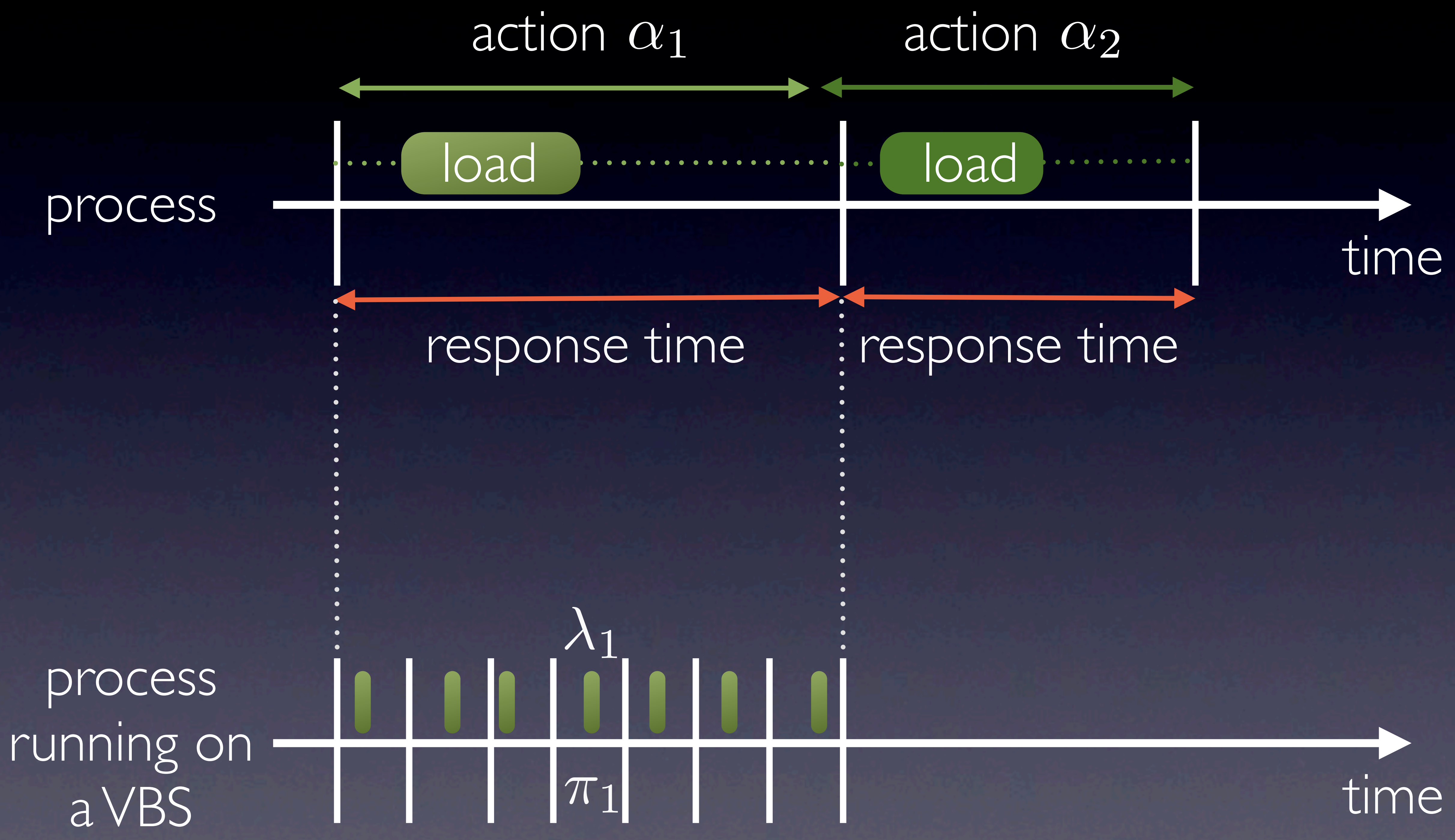
One Process on a VBS



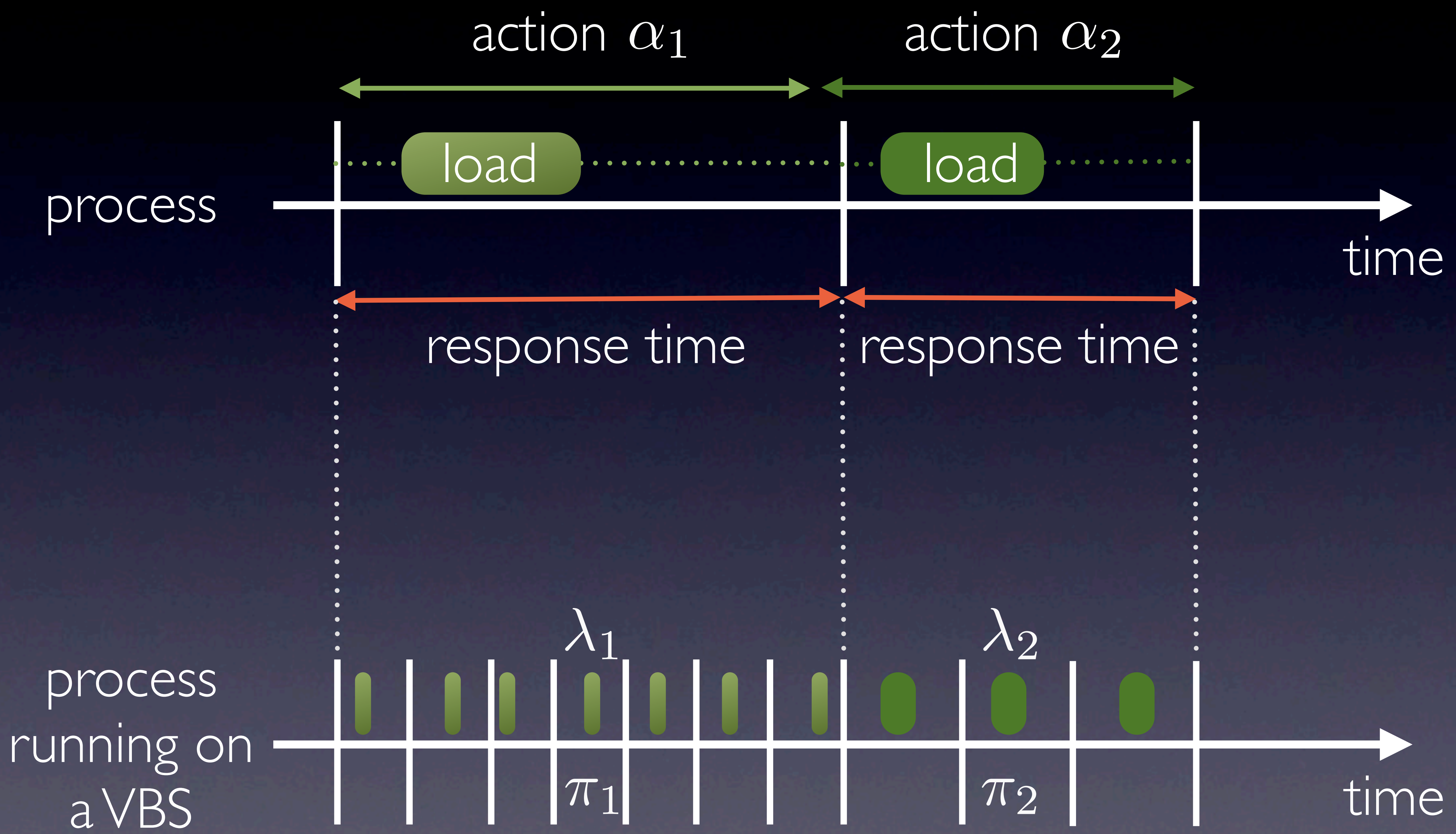
One Process on a VBS



One Process on a VBS



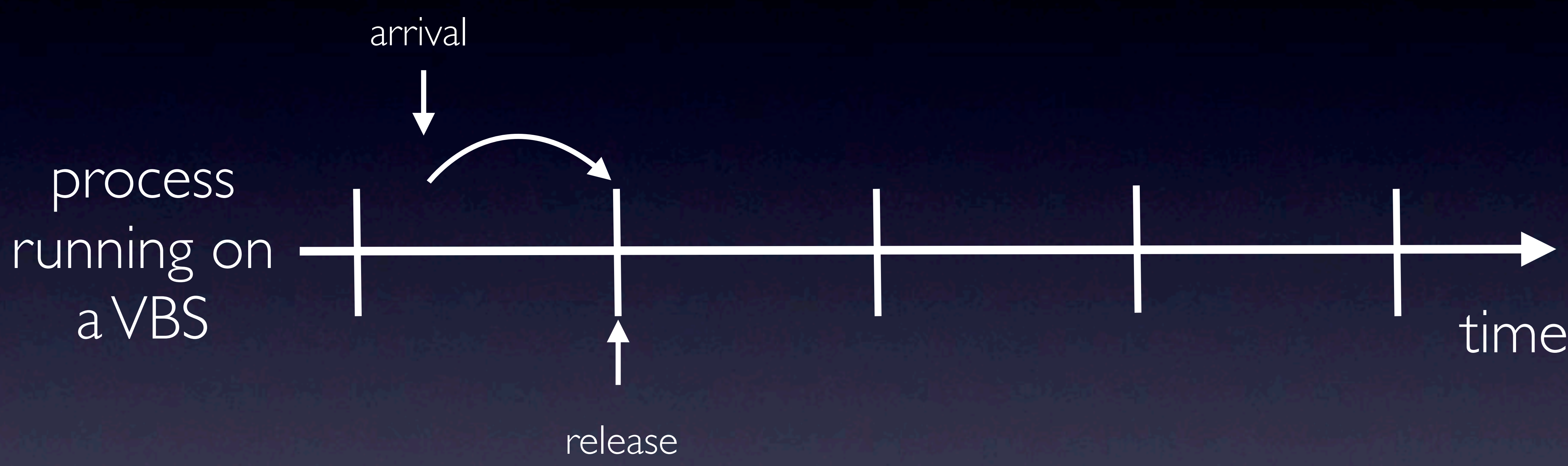
One Process on a VBS



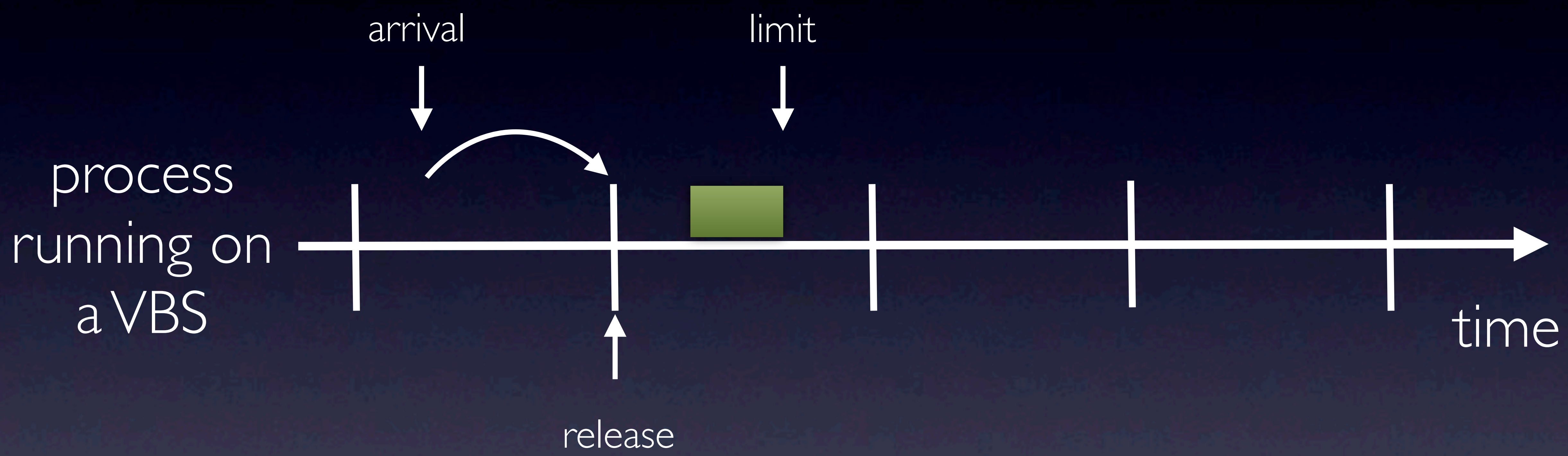
VBS

VBS

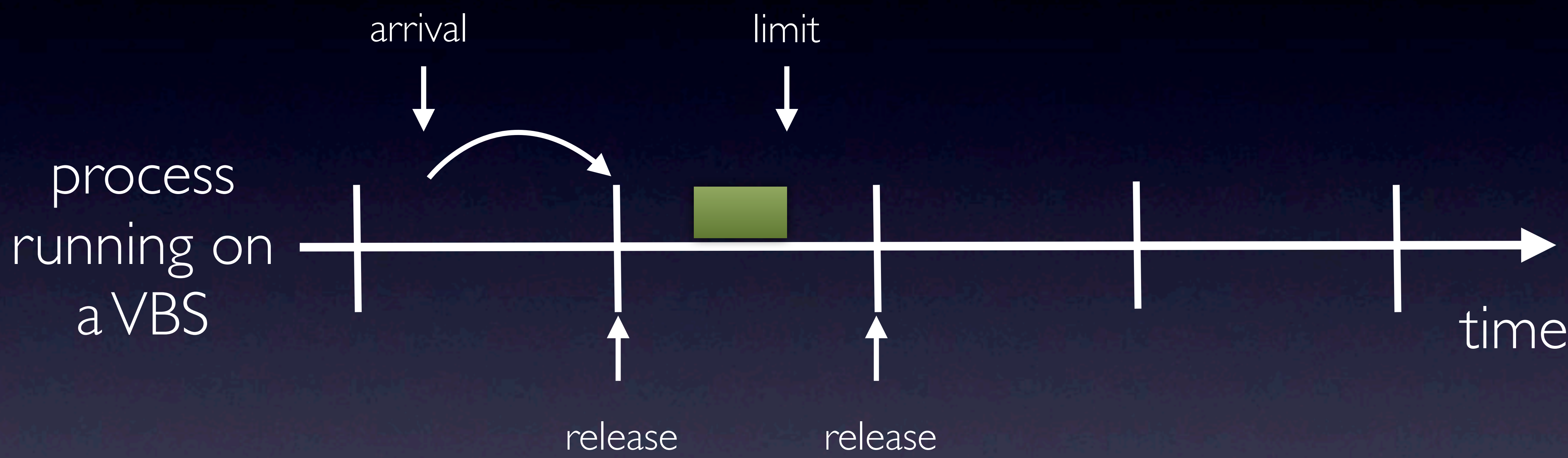
VBS



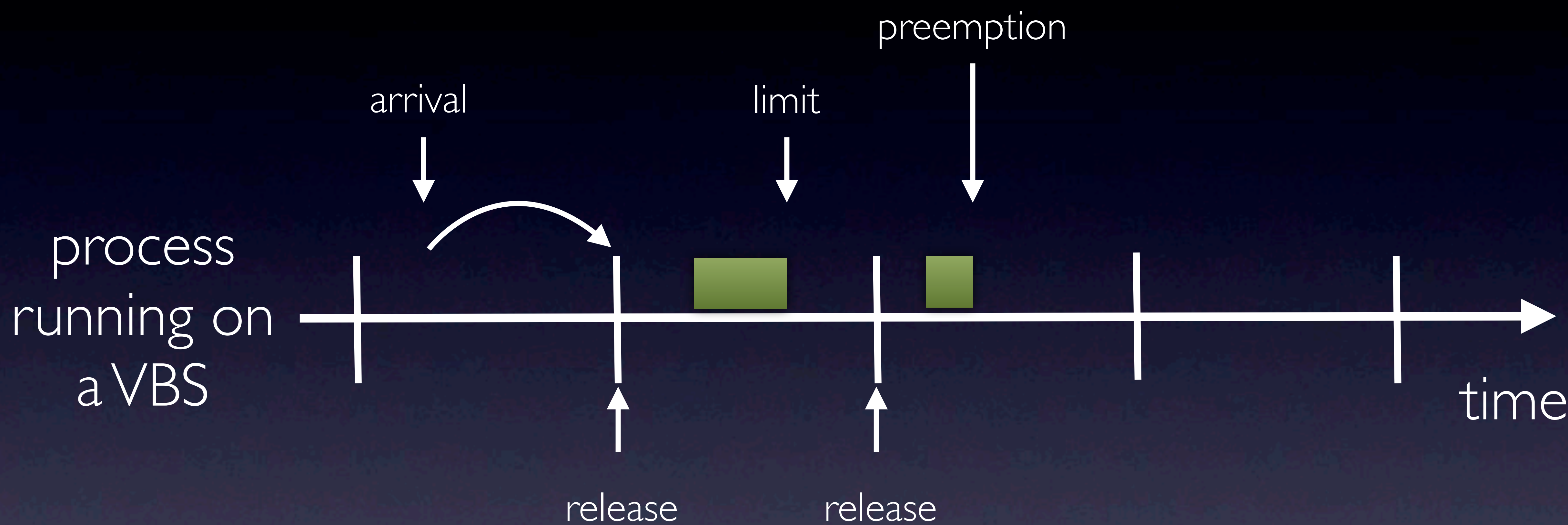
VBS



VBS


arrival
limit
process
running on
a VBS
time
release
release

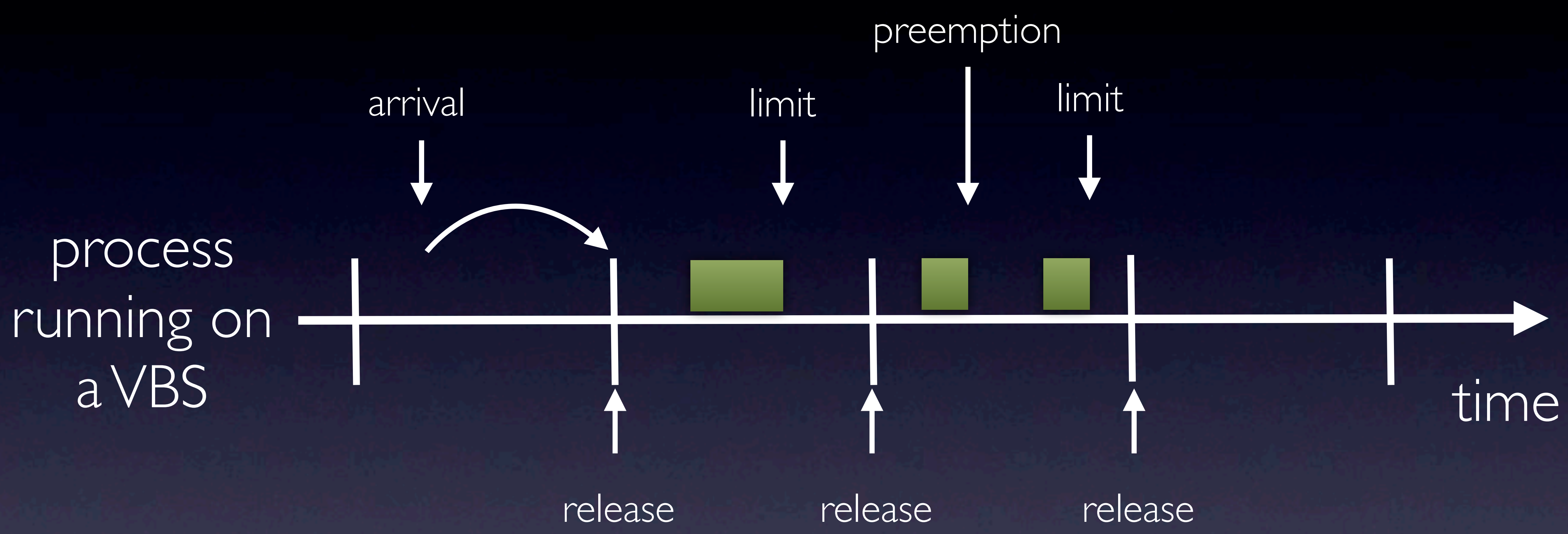
VBS


preemption
arrival
limit
process
running on
a VBS
time
release
release

VBS


preemption
arrival
limit
limit
process running on a VBS
time
release
release

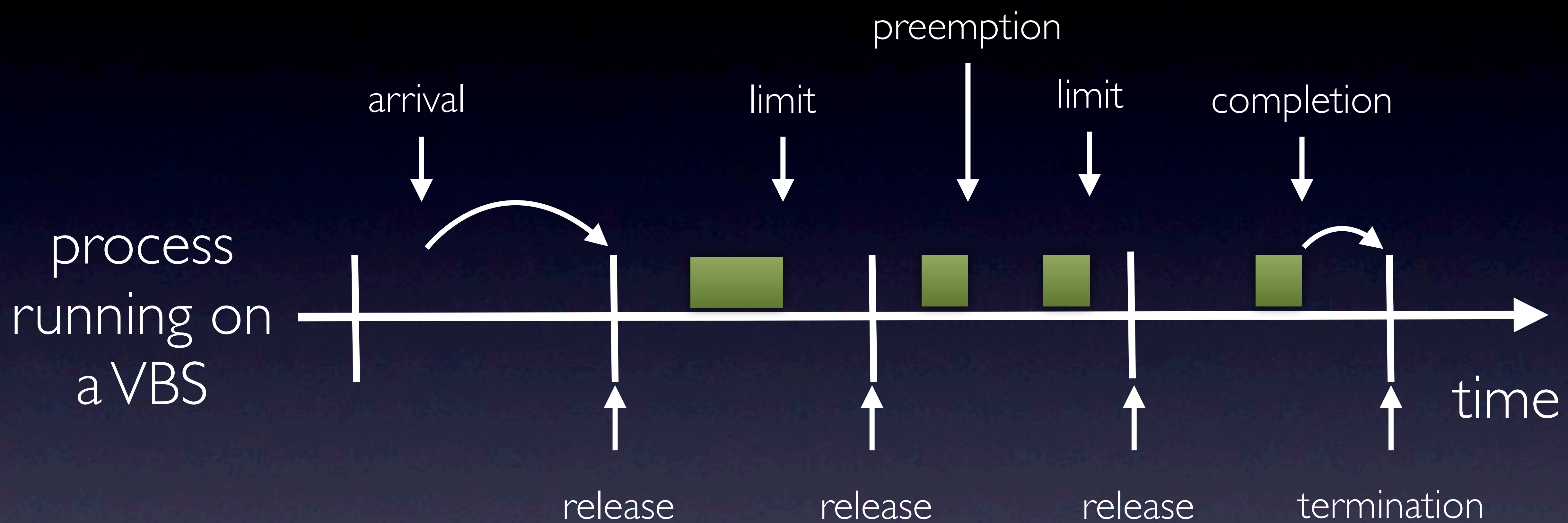
VBS


preemption
arrival
limit
limit
process running on a VBS
time
release
release
release

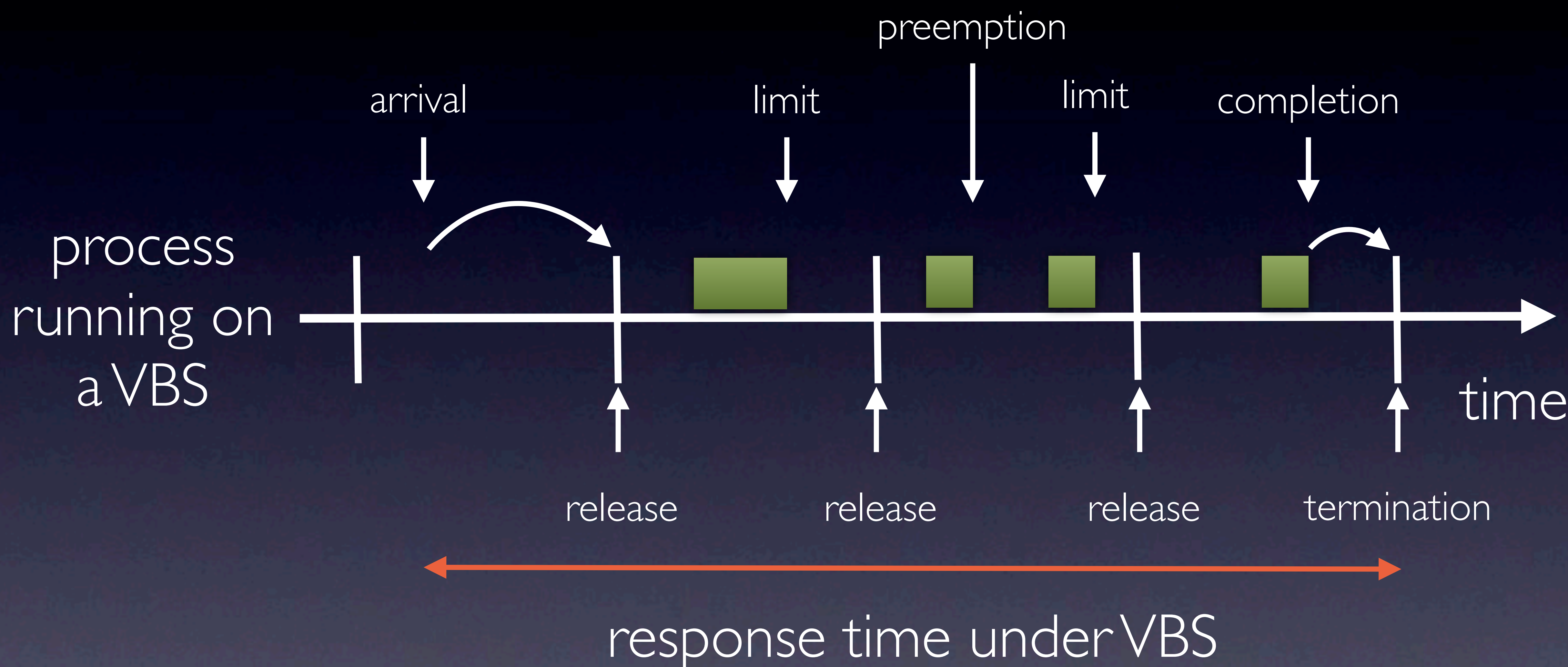
VBS


preemption
arrival
limit
limit
completion
process running on a VBS
time
release
release
release

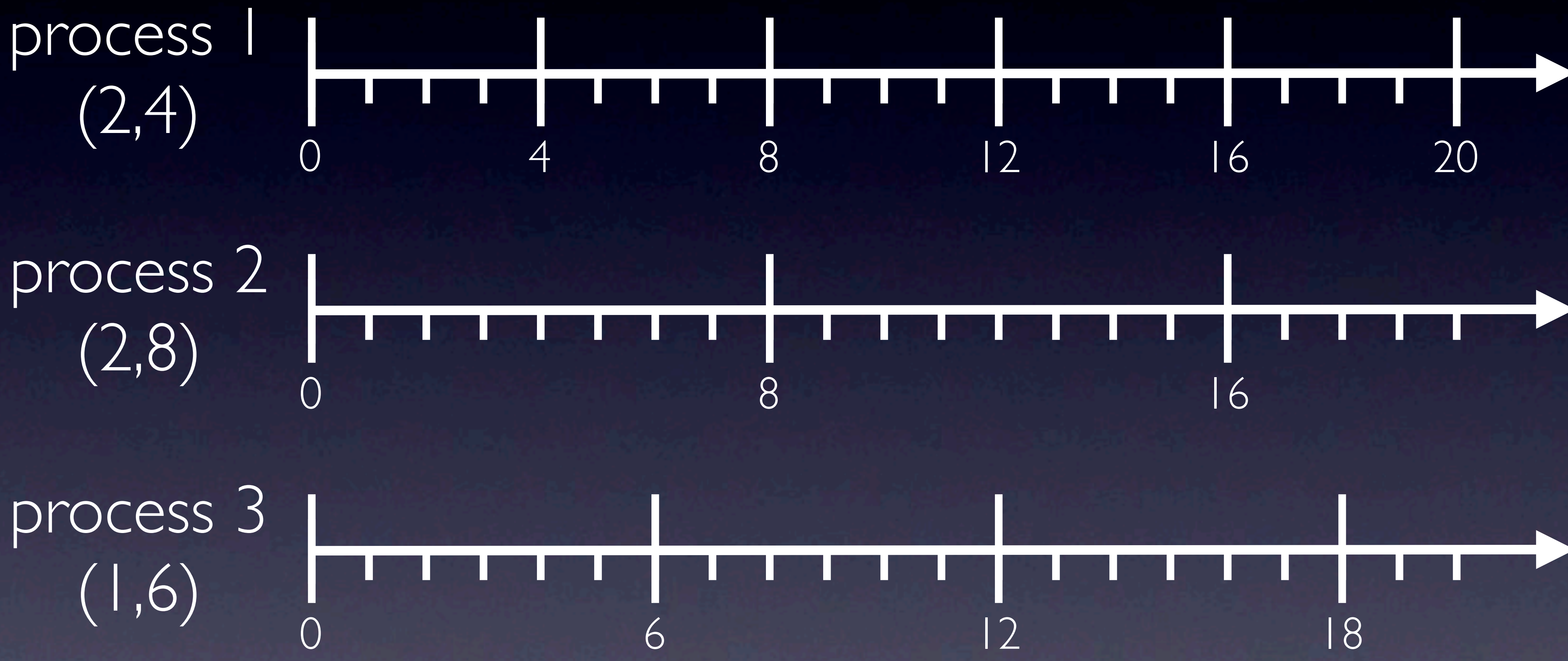
VBS


preemption
arrival
limit
limit
completion
process running on a VBS
time
release
release
release
termination

VBS

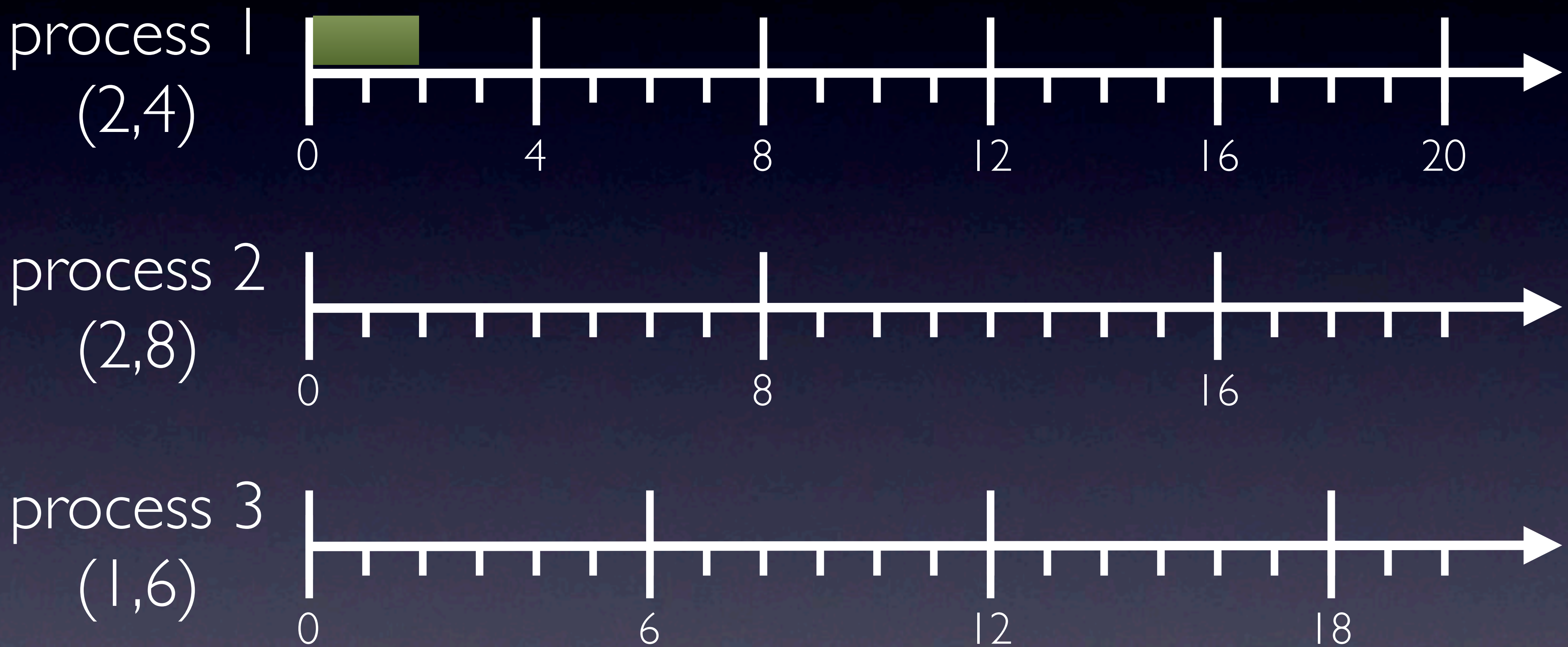

preemption
arrival
limit
limit
completion
process
running on
a VBS
time
release
release
release
termination
response time under VBS

VBS



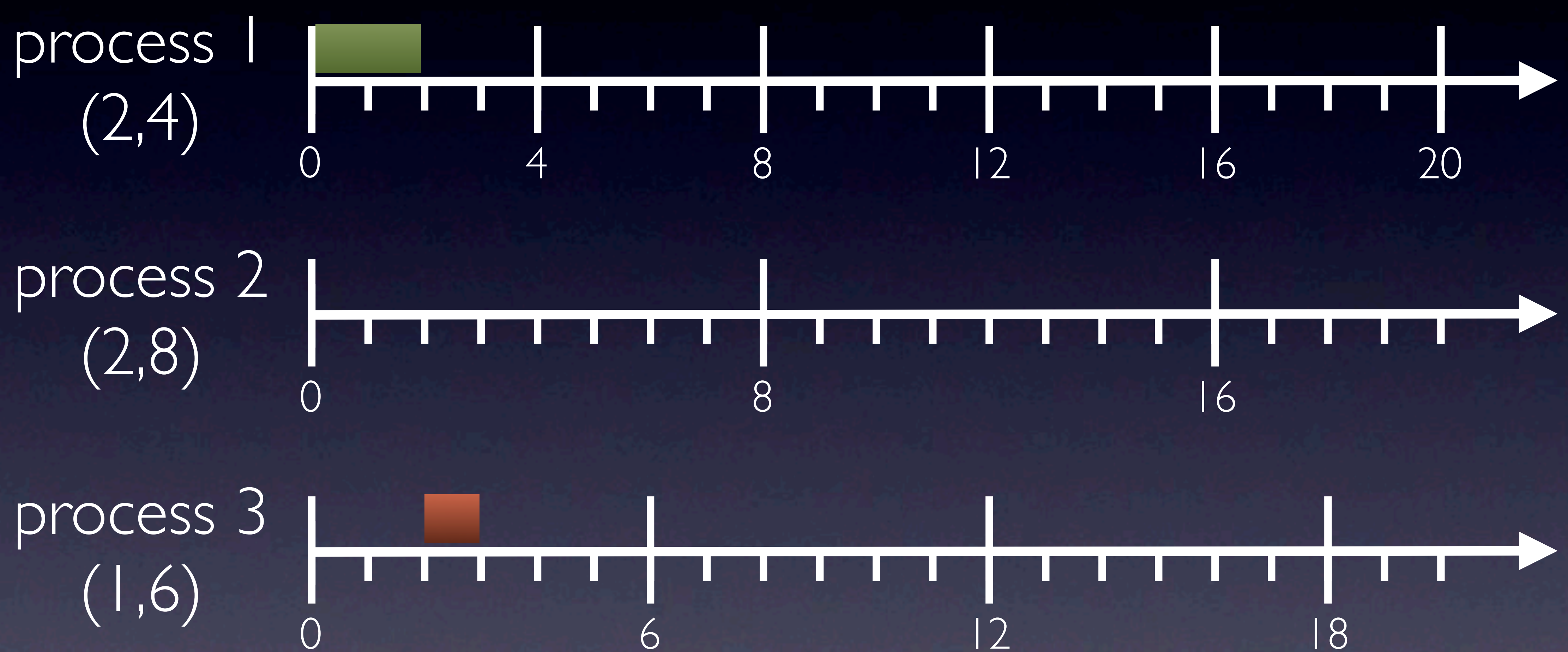
multiple processes are EDF-scheduled

VBS



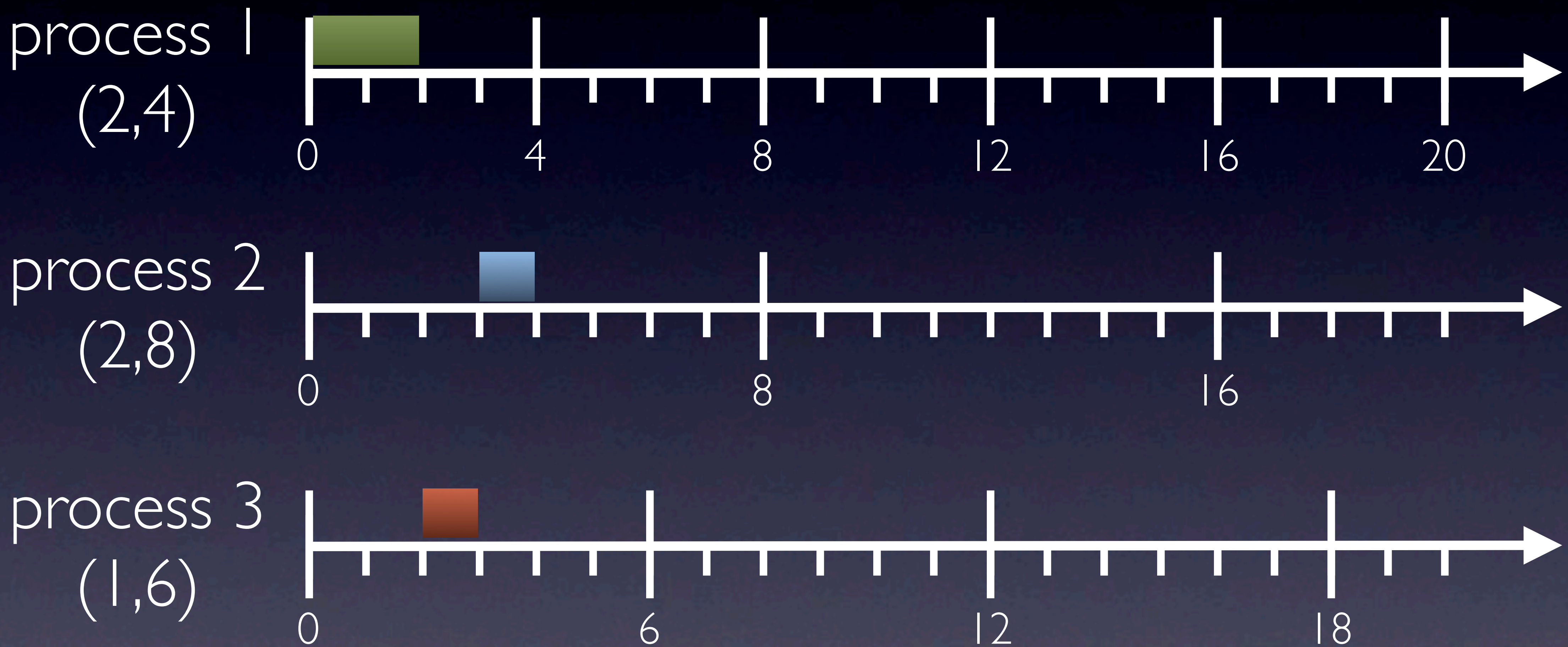
multiple processes are EDF-scheduled

VBS



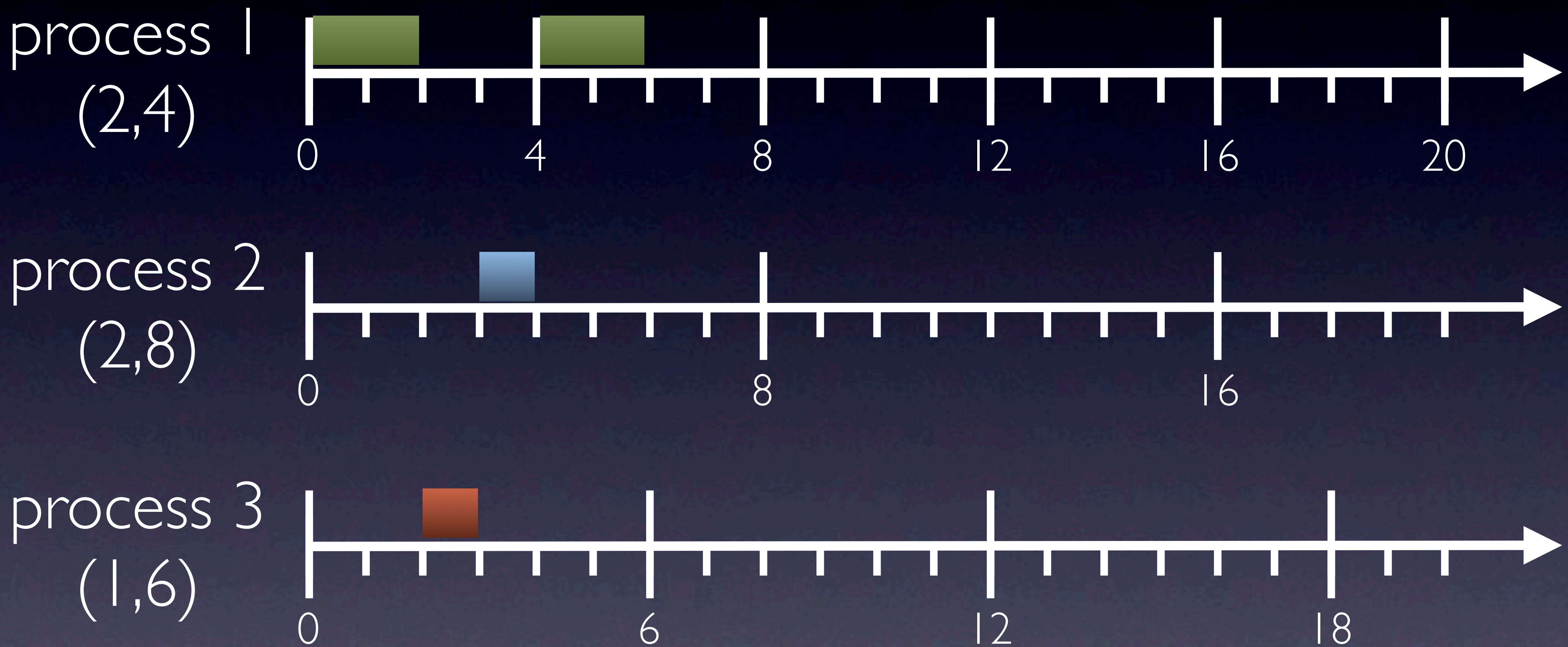
multiple processes are EDF-scheduled

VBS



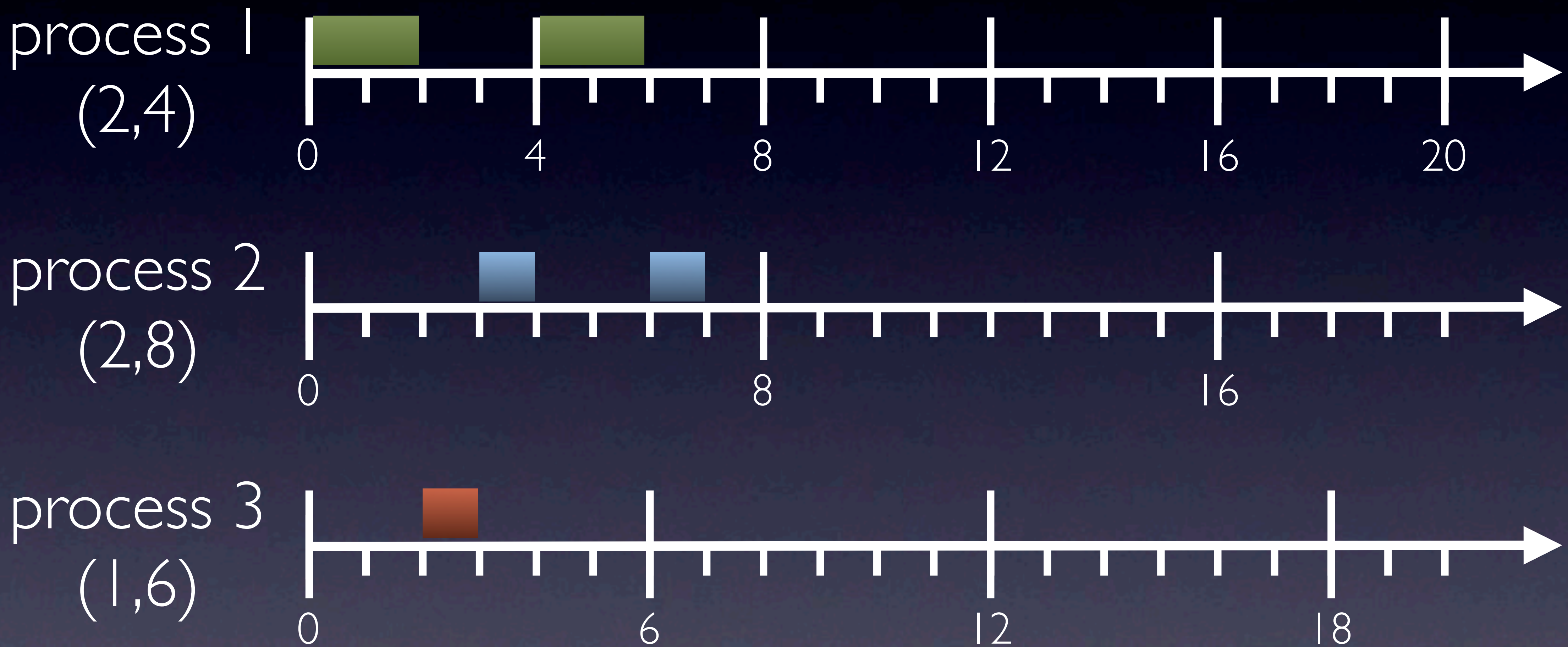
multiple processes are EDF-scheduled

VBS



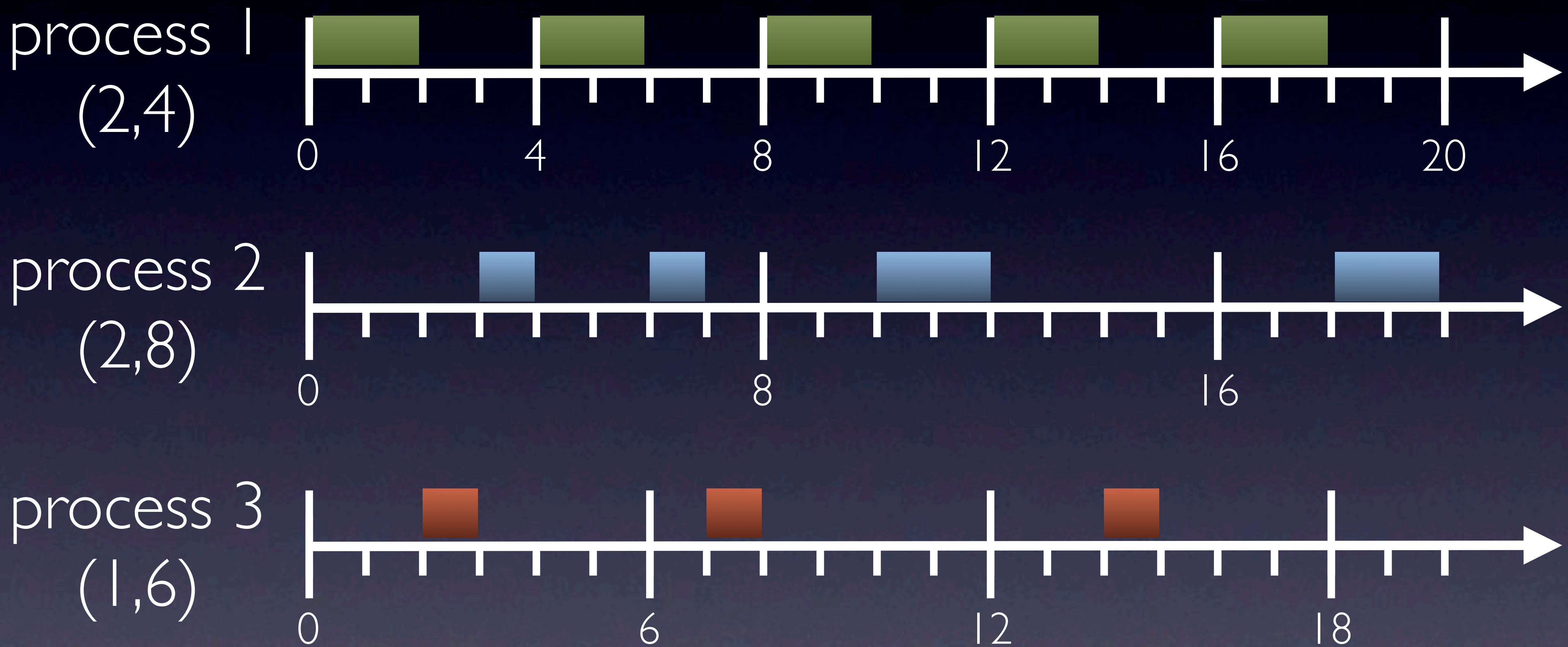
multiple processes are EDF-scheduled

VBS



multiple processes are EDF-scheduled

VBS



multiple processes are EDF-scheduled

Scheduling Result and Bounds

[SIES09]

Processes $P_1, P_2, \ldots, P_n$ on VBSs $u_1, u_2, \ldots, u_n$ are schedulable if $\sum_{i=1}^{n} u_i \leq 1$

Scheduling Result and Bounds
[SIES09]

Processes $P_1, P_2, \ldots, P_n$ on VBSs $u_1, u_2, \ldots, u_n$ are schedulable if $\sum_{i=1}^{n} u_i \leq 1$

For any action α on a resource (λ, π) we have:

- upper response-time bound $\left\lceil \frac{load}{\lambda} \right\rceil \pi + \pi - 1$
- lower response-time bound $\left\lceil \frac{load}{\lambda} \right\rceil \pi$
- jitter $\pi - 1$

Scheduling Result and Bounds

[SIES09]

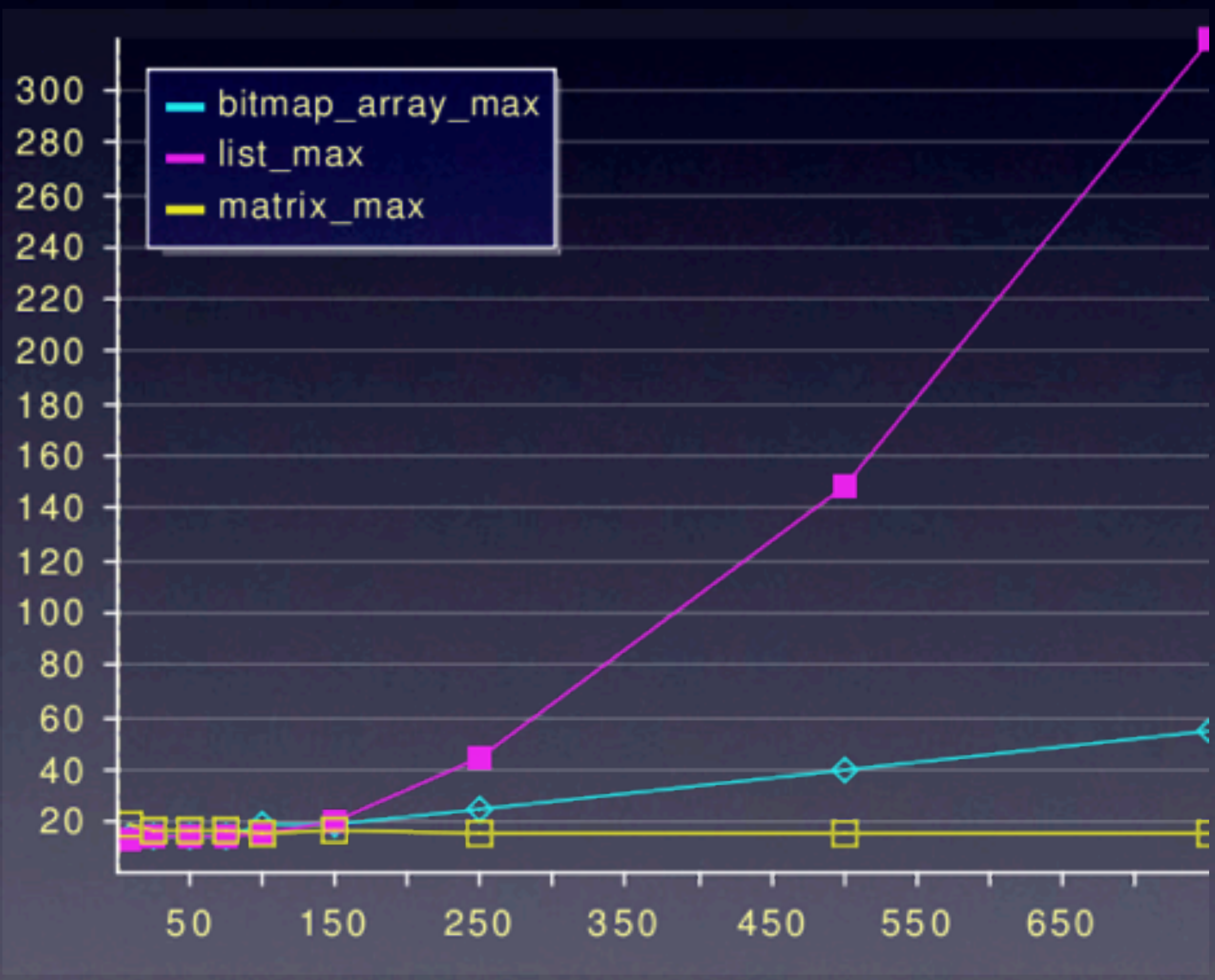
Processes $P_1, P_2, \ldots, P_n$ on VBSs $u_1, u_2, \ldots, u_n$ are schedulable if $\sum_{i=1}^{n} u_i \leq 1$

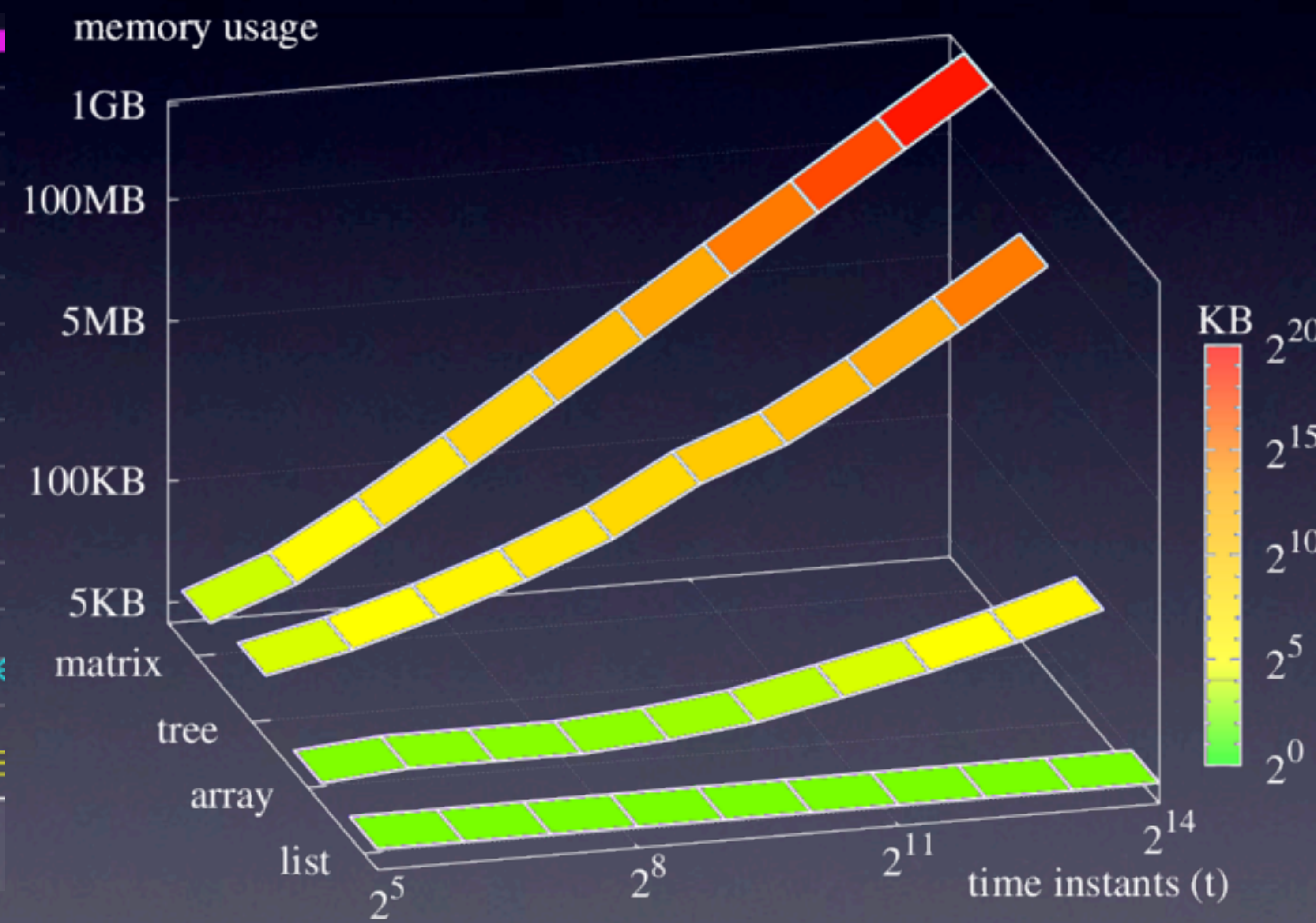
For any action α on a resource (λ, π) we have:

- upper response-time bound $\lceil \frac{load}{\lambda} \rceil \pi + \pi - 1$
- lower response-time bound $\lceil \frac{load}{\lambda} \rceil \pi$
- jitter $\pi - 1$

temporal isolation

Scheduler Overhead
[SIES09]





Power-Aware VBS

Dynamic Voltage and Frequency Scaling

Power-Aware VBS

Dynamic Voltage and Frequency Scaling

Maintain VBS properties (temporal isolation, bounds)

Power-Aware VBS

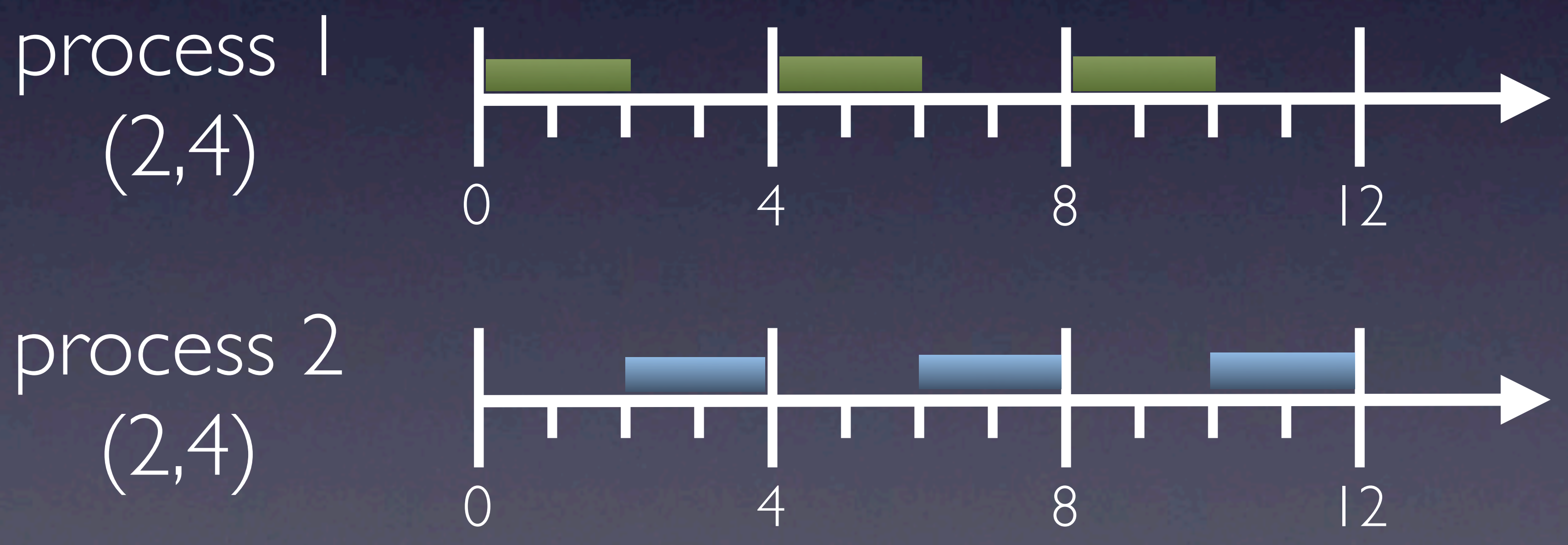
Dynamic Voltage and Frequency Scaling

Maintain VBS properties (temporal isolation, bounds)

Power-Aware VBS

Dynamic Voltage and Frequency Scaling

Maintain VBS properties (temporal isolation, bounds)



Power-Aware VBS

Dynamic Voltage and Frequency Scaling

Maintain VBS properties (temporal isolation, bounds)

Possible whenever there is slack in the system

Power-Aware VBS

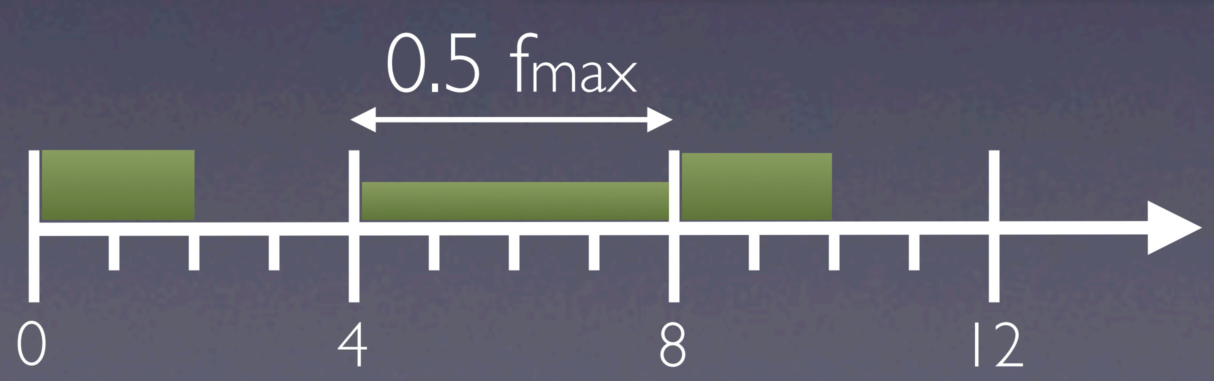
EDF frequency scaling result:

An EDF-schedulable set of tasks is still schedulable if the processor frequency in between any two release times is set to at least

$$U_c \cdot f_{max}$$

current total utilization of all released tasks in the considered interval of time between two releases

process 1
(2,4)



Frequency-scaling VBS

Slack

Frequency-scaling VBS

Static Slack

Slack

Action Slack

Dynamic Slack

Termination Slack

Frequency-scaling VBS

Frequency is scaled to the sum of the bandwidth caps and not changed at runtime

Slack

- Static Slack
- Dynamic Slack
  - Action Slack
  - Termination Slack

Frequency-scaling VBS

Frequency is scaled to the sum of the bandwidth caps and not changed at runtime

Static Slack

Slack

Frequency is scaled at release time to the sum of the utilizations of the released actions

Action Slack

Dynamic Slack

Termination Slack

Frequency-scaling VBS

Static Slack

Frequency is scaled to the sum of the bandwidth caps and not changed at runtime

Action Slack

Frequency is scaled at release time to the sum of the utilizations of the released actions

Dynamic Slack

Termination Slack

New limits are computed for each action such that the upper response-time bound is maintained

Frequency-scaling VBS

Frequency-scaling VBS

Static slack

$$f = \sum_{i=1}^{n} u_i \cdot f_{max}$$

Frequency-scaling VBS

Static slack

$$f = \sum_{i=1}^{n} u_i \cdot f_{max}$$

Action slack

$$f = \sum_{i=1}^{n} \frac{\lambda_{i,j}}{\pi_{i,j}} \cdot f_{max}$$

Frequency-scaling VBS

Static slack

$$f = \sum_{i=1}^{n} u_i \cdot f_{max}$$

Action slack

$$f = \sum_{i=1}^{n} \frac{\lambda_{i,j}}{\pi_{i,j}} \cdot f_{max}$$

Termination slack

$$f = \sum_{i=1}^{n} \frac{\lambda^*_{i,j}}{\pi_{i,j}} \cdot f_{max} \qquad \lambda^*_{i,j} = \left\lceil \frac{l_{i,j}}{n_{i,j}} \right\rceil \qquad n_{i,j} = \left\lceil \frac{l_{i,j}}{\lambda_{i,j}} \right\rceil$$

Frequency-scaling VBS

Static slack

$$f = \sum_{i=1}^{n} u_i \cdot f_{max}$$

Action slack

$$f = \sum_{i=1}^{n} \frac{\lambda_{i,j}}{\pi_{i,j}} \cdot f_{max}$$

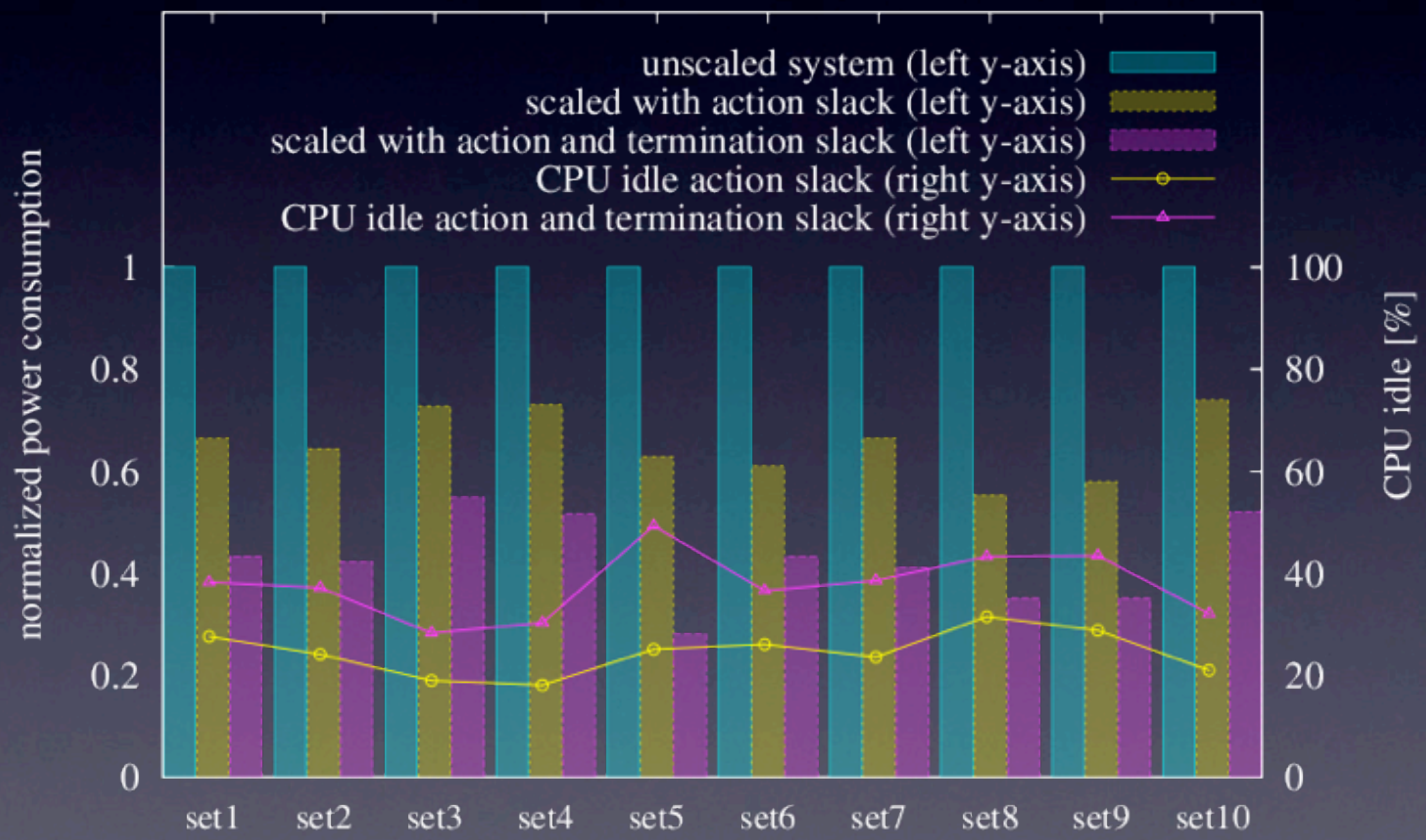
Termination slack

$$f = \sum_{i=1}^{n} \frac{\lambda^*_{i,j}}{\pi_{i,j}} \cdot f_{max} \qquad \lambda^*_{i,j} = \left\lceil \frac{l_{i,j}}{n_{i,j}} \right\rceil \qquad n_{i,j} = \left\lceil \frac{l_{i,j}}{\lambda_{i,j}} \right\rceil$$

Termination and action slack can be used separately or together

Power-Aware VBS

Assuming a simple power model ($P \propto V^2$)



Look-ahead FS-VBS

Look-ahead FS-VBS

With knowledge of future events:

- redistribute computation time between periods

Look-ahead FS-VBS

With knowledge of future events:

- redistribute computation time between periods
- optimal offline method

Look-ahead FS-VBS

With knowledge of future events:

- redistribute computation time between periods
- optimal offline method
- feasible online method

Look-ahead FS-VBS

With knowledge of future events:

- redistribute computation time between periods
- optimal offline method
- feasible online method

May help to handle:

more complex power models

Look-ahead FS-VBS

With knowledge of future events:

- redistribute computation time between periods
- optimal offline method
- feasible online method

May help to handle:

more complex power models

frequency switching cost (time and power)

Look-ahead FS-VBS

With knowledge of future events:

- redistribute computation time between periods
- optimal offline method
- feasible online method

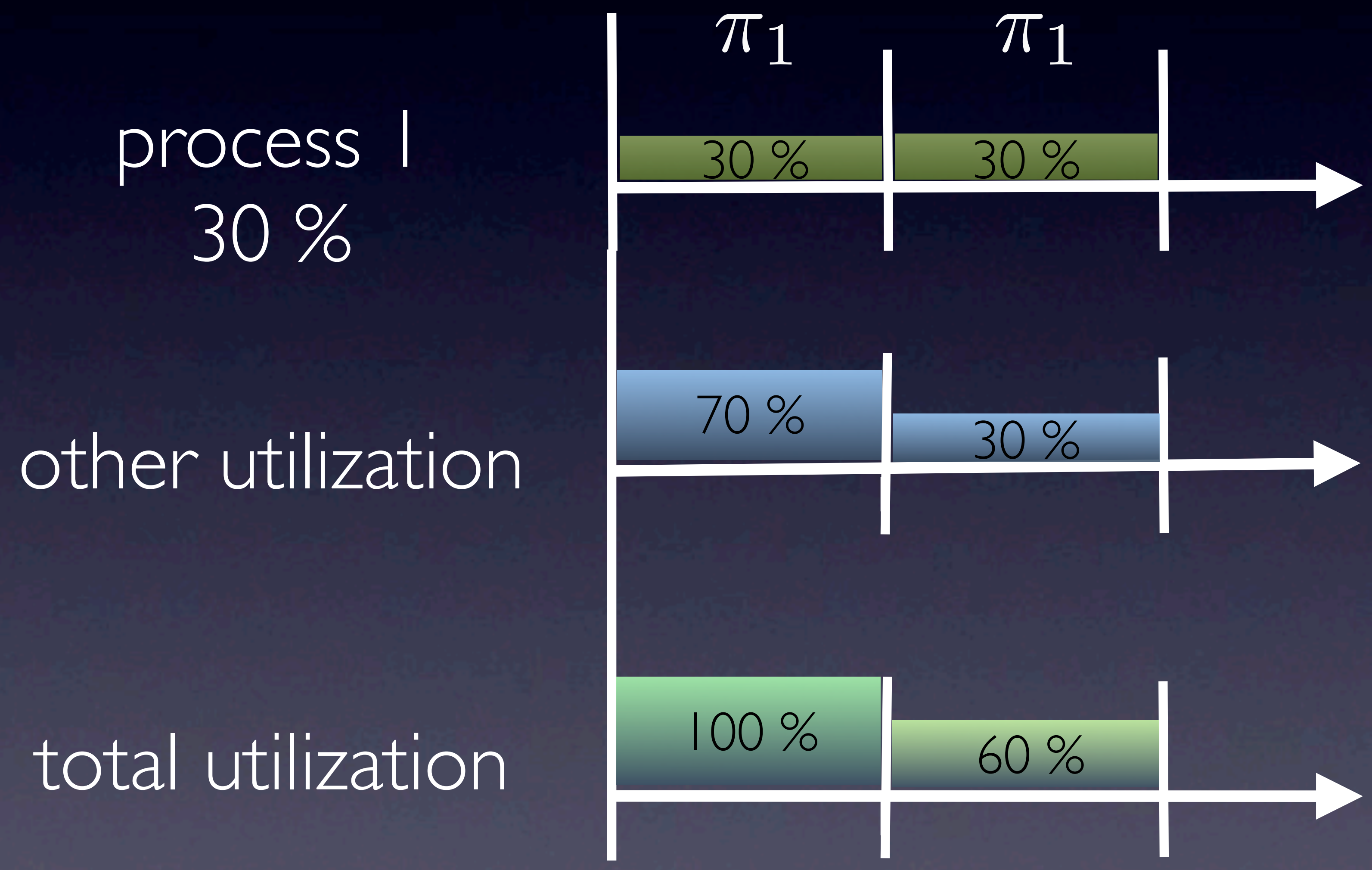
May help to handle:

more complex power models

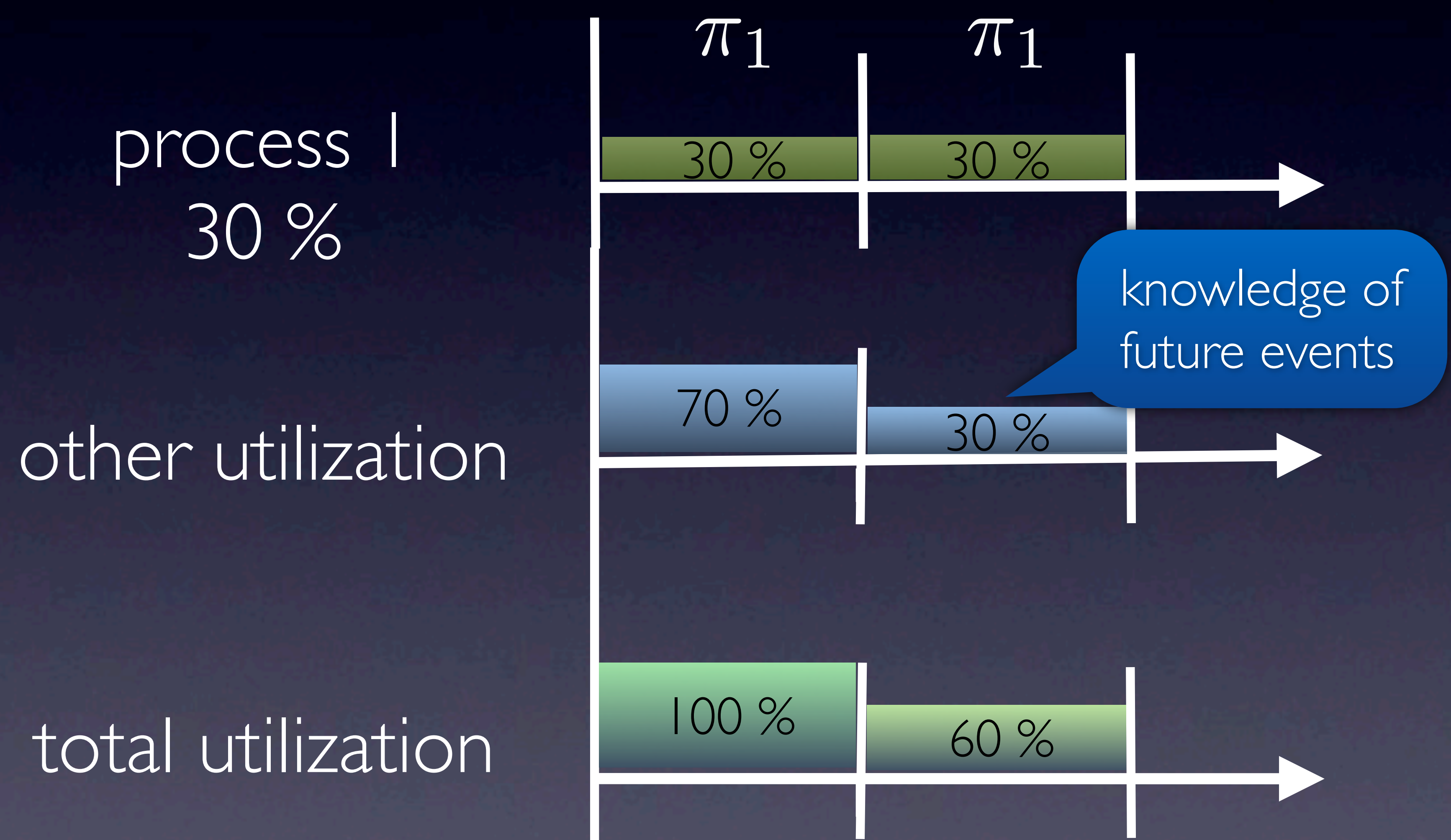
frequency switching cost (time and power)

- time overhead included using overhead accounting [RTAS10]

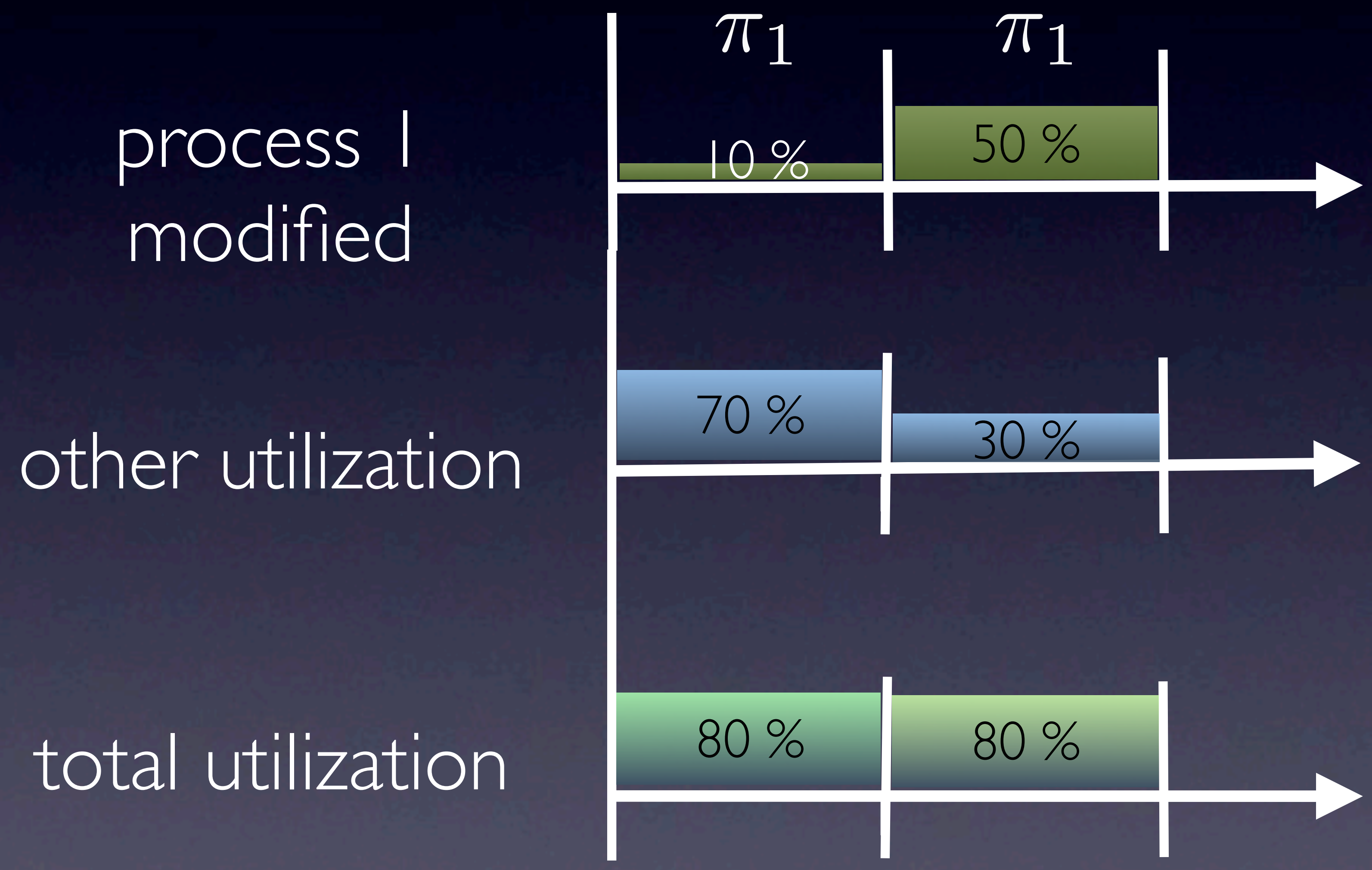
Look-ahead FS-VBS



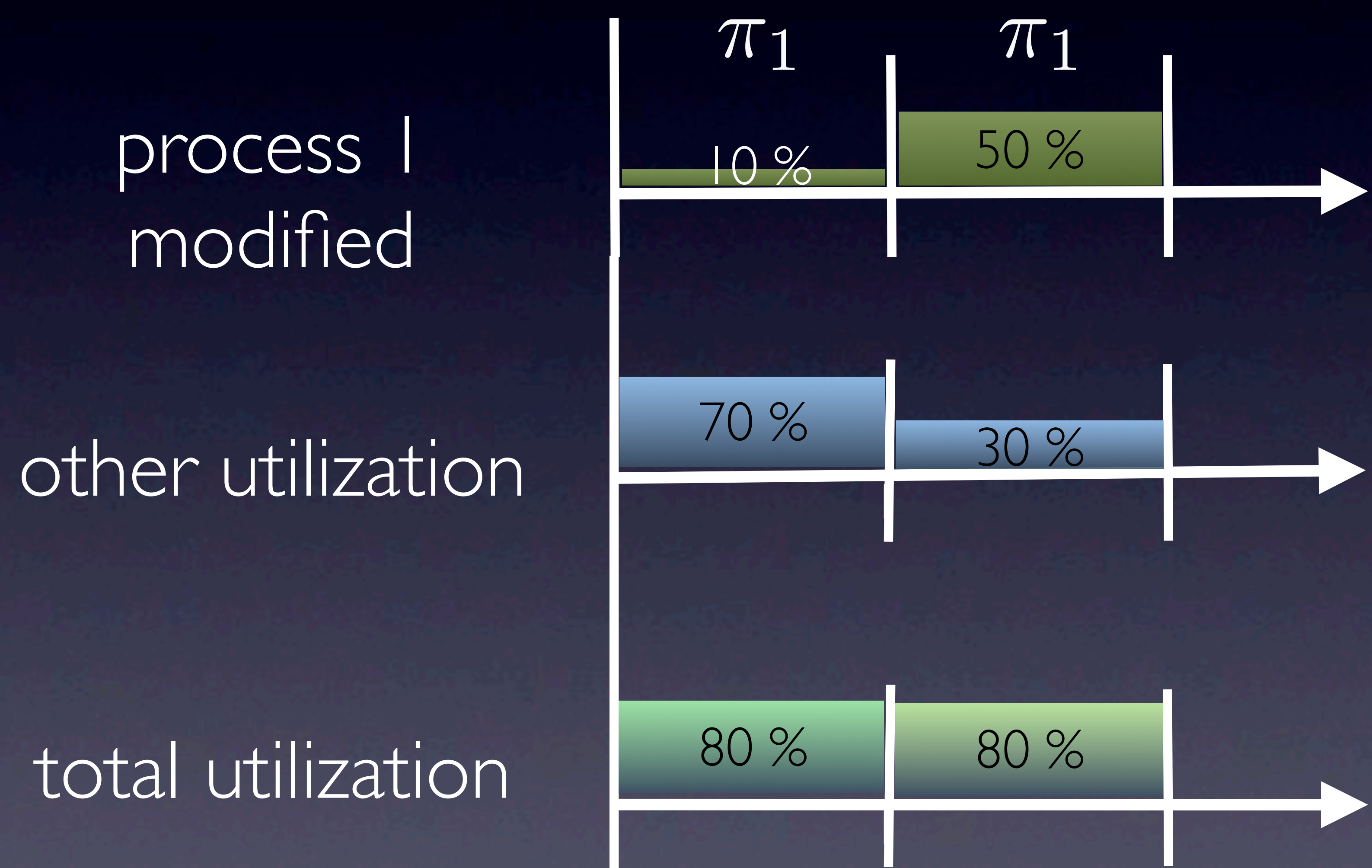
Look-ahead FS-VBS



Look-ahead FS-VBS



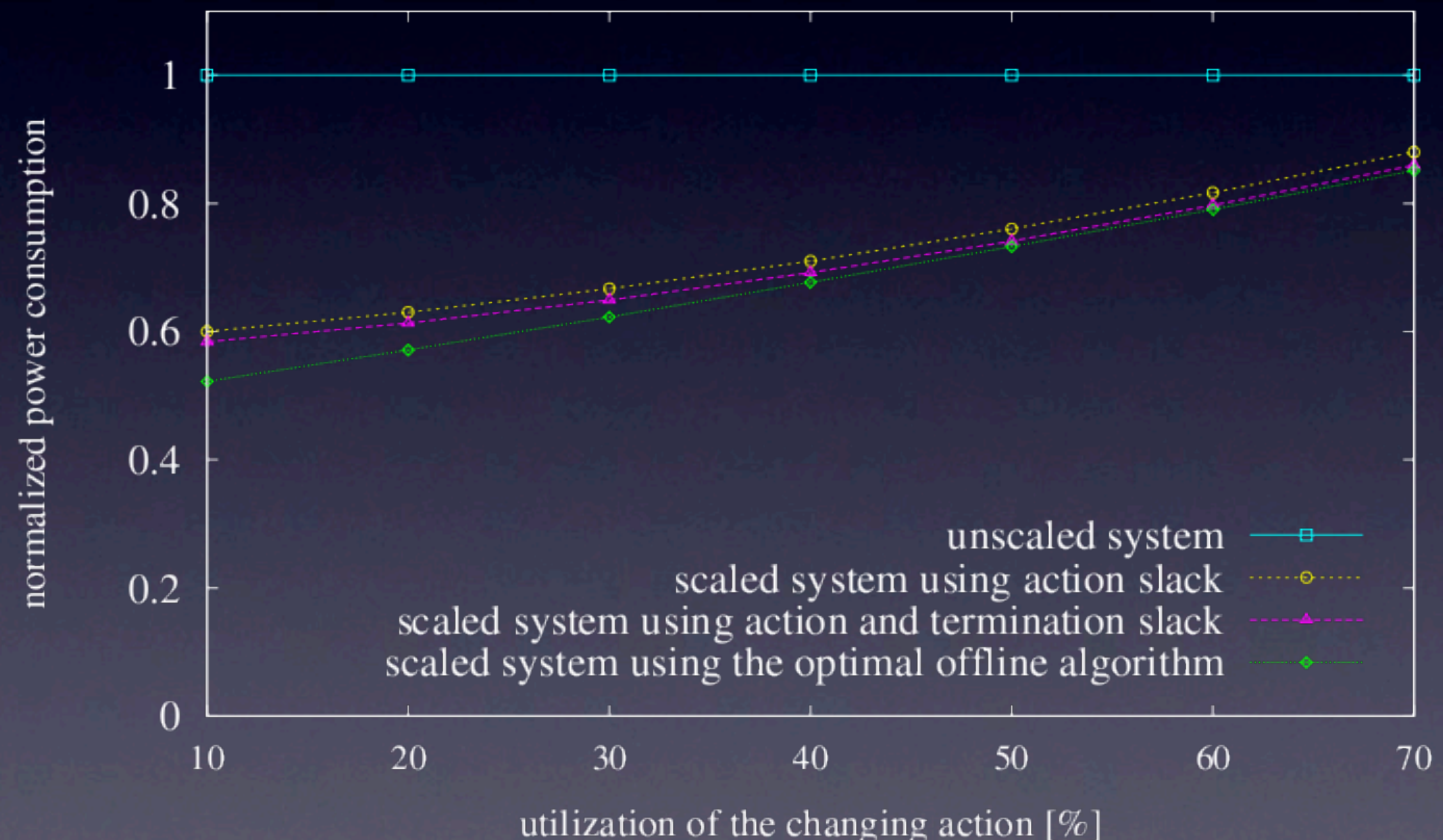
Look-ahead FS-VBS



actual improvement depends on the power model

Look-ahead FS-VBS

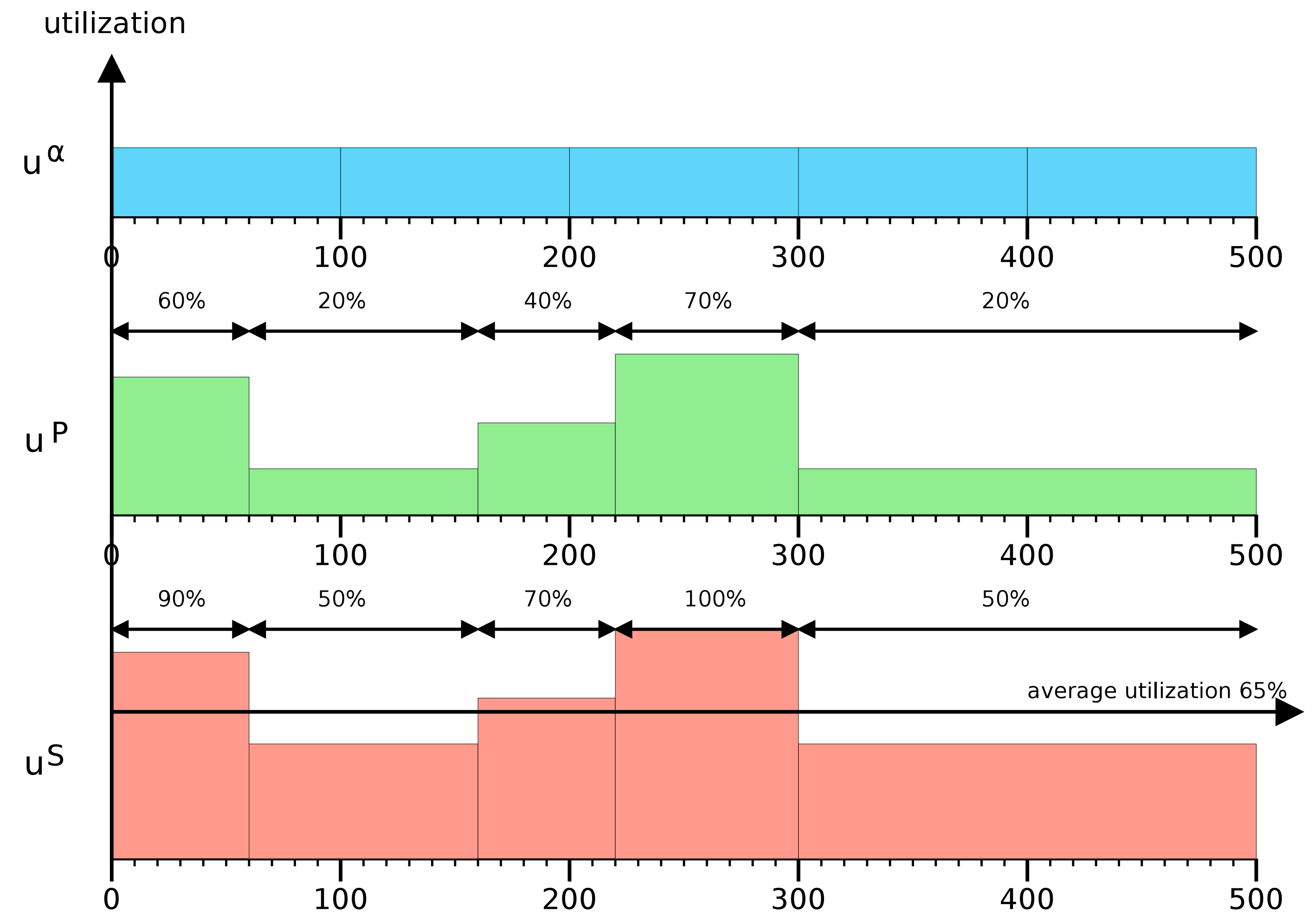
Assuming a simple power model ($P \propto V^2$)



Look-ahead online FS-VBS

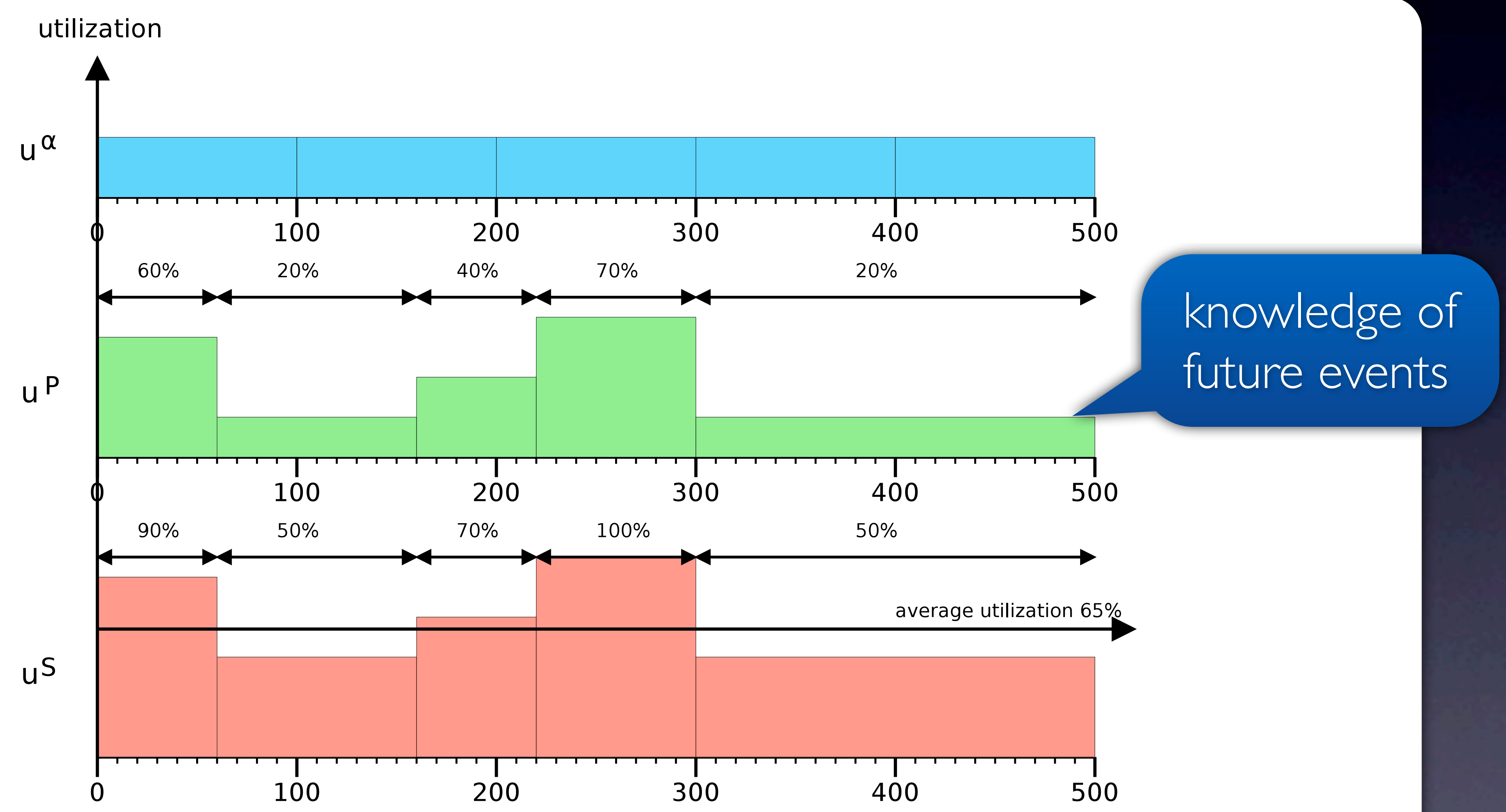
Look-ahead online FS-VBS

Assume a simple power model ($P \propto V^2$)



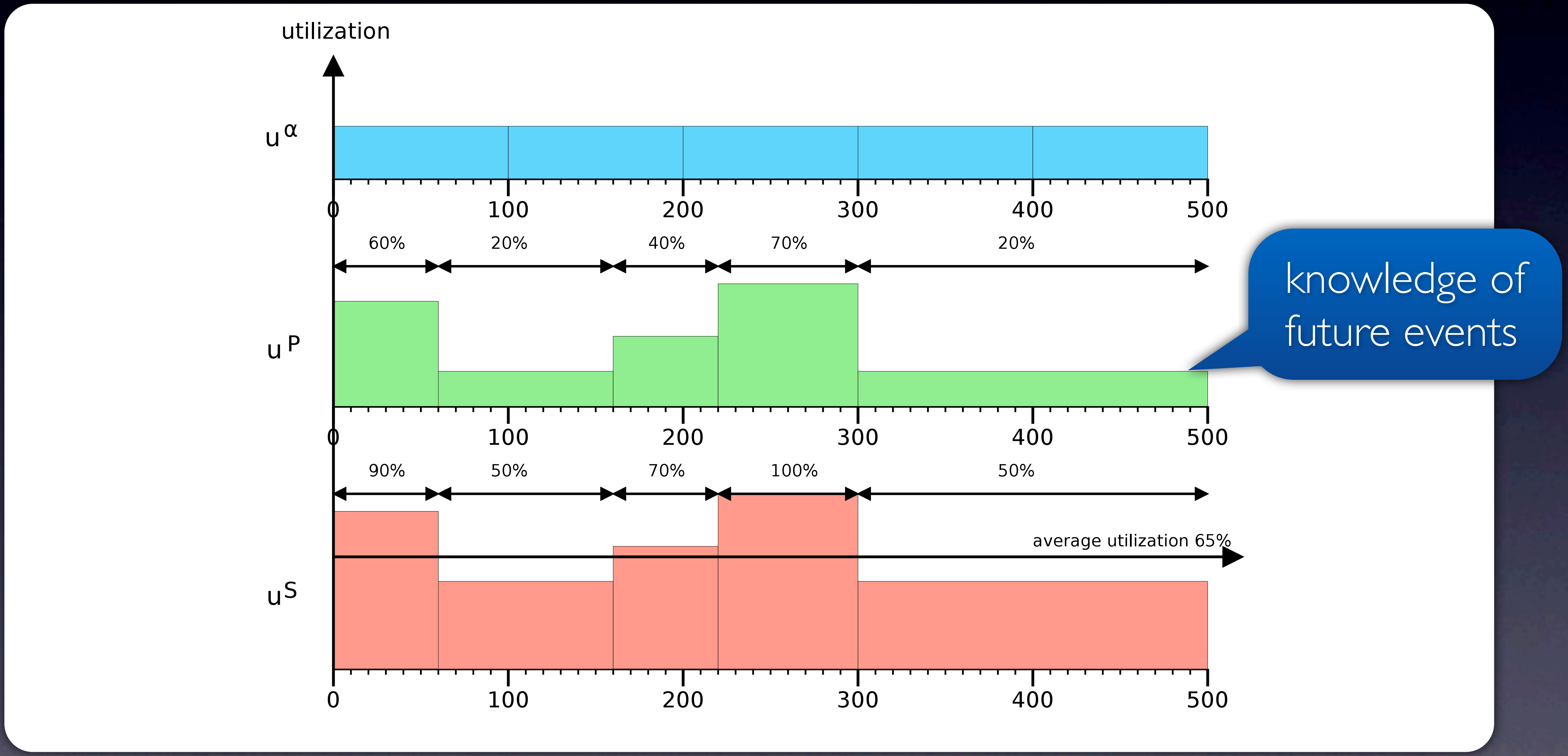
Look-ahead online FS-VBS

Assume a simple power model ($P \propto V^2$)



Look-ahead online FS-VBS

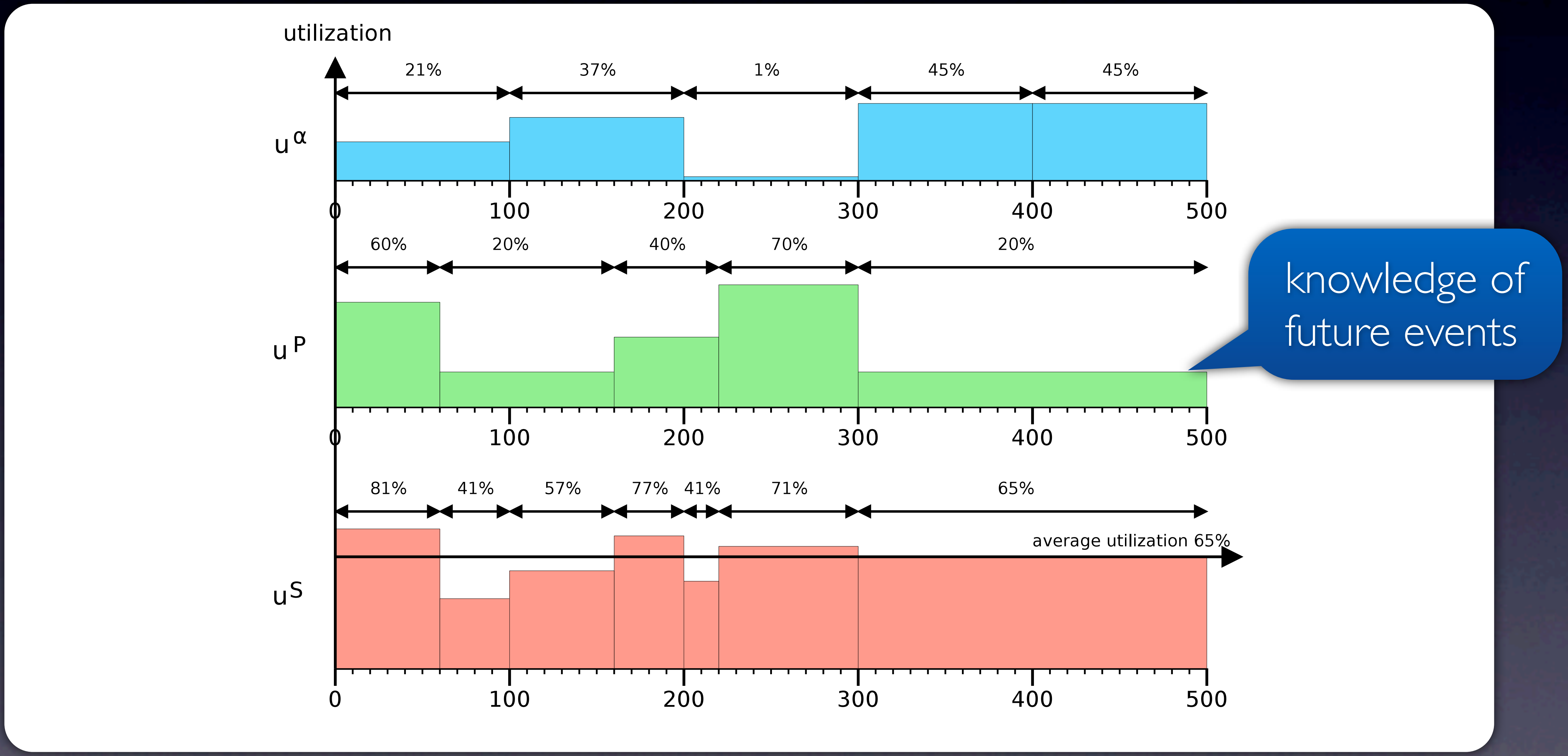
Assume a simple power model ($P \propto V^2$)



Modify the limits in each period (whenever possible) s.t. the utilization approximates the average utilization

Look-ahead online FS-VBS

Assume a simple power model ($P \propto V^2$)

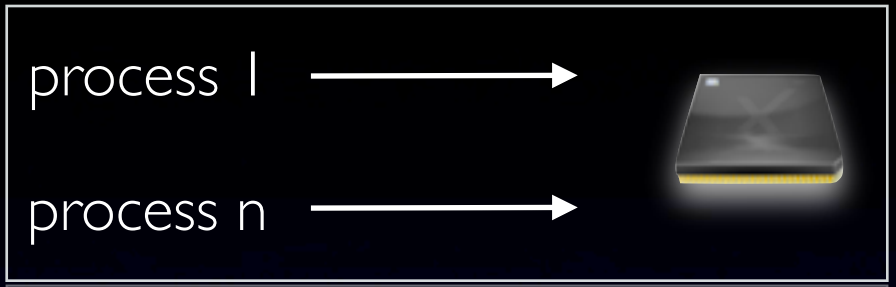


Modify the limits in each period (whenever possible) s.t. the utilization approximates the average utilization

Conclusions

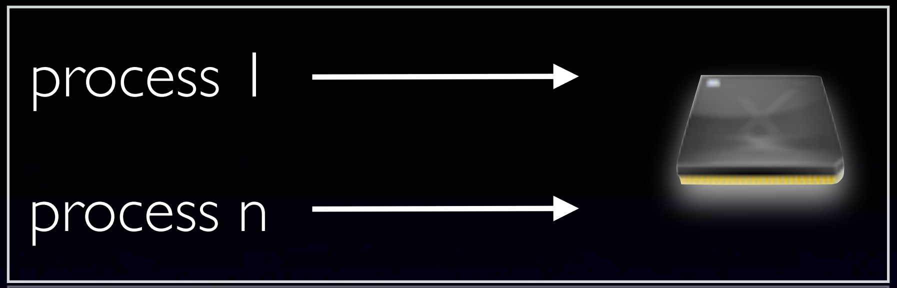
Conclusions

- Server-based scheduling for temporal isolation

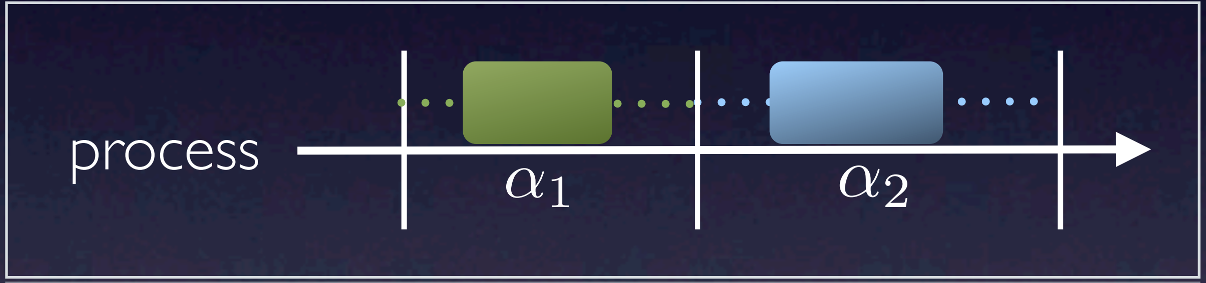


Conclusions

- Server-based scheduling for temporal isolation

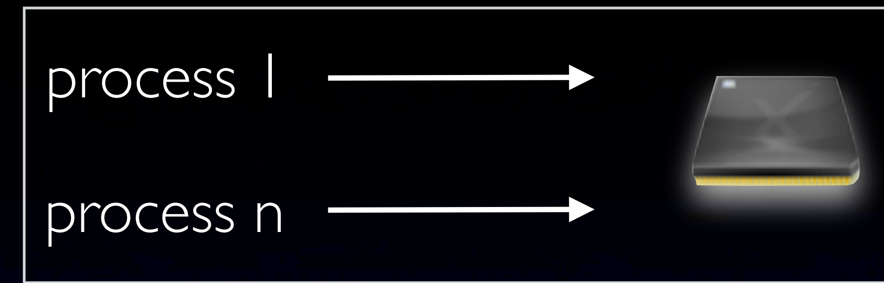


- VBS for variable execution speed

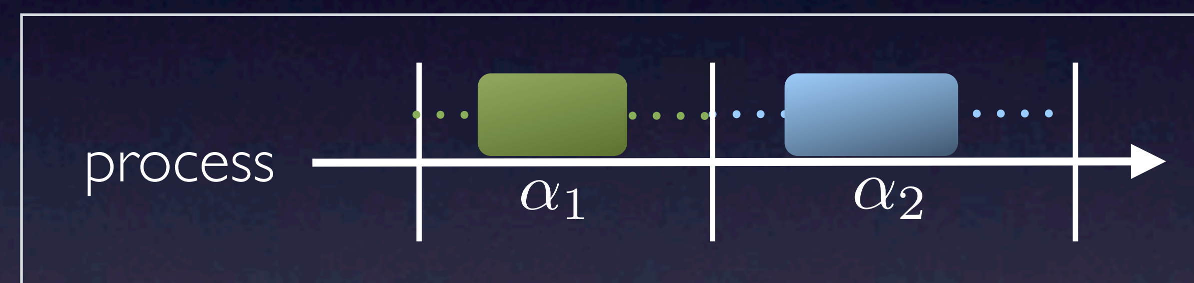


Conclusions

- Server-based scheduling for temporal isolation



- VBS for variable execution speed



- Power-aware VBS

Static Slack

optimal offline FS-VBS

Dynamic Slack

Termination Slack

Action Slack

feasible online FS-VBS