

# Programmable Temporal Isolation for Real-Time Systems

Silviu Craciunas  
Department of Computer Sciences  
University of Salzburg





# Structure



# Structure

VBS process model

varying real-time constraints while maintaining temporal isolation



# Structure

VBS process model

varying real-time constraints while maintaining temporal isolation

VBS scheduling result

EDF-based scheduler, response time bounds, queue mechanism



# Structure

VBS process model

varying real-time constraints while maintaining temporal isolation

VBS scheduling result

EDF-based scheduler, response time bounds, queue mechanism

Overhead accounting

include scheduler overhead in the schedulability analysis



# Structure

VBS process model

varying real-time constraints while maintaining temporal isolation

VBS scheduling result

EDF-based scheduler, response time bounds, queue mechanism

Overhead accounting

include scheduler overhead in the schedulability analysis

Power-aware VBS

reduce CPU power consumption



# Process model

process  time



# Process model



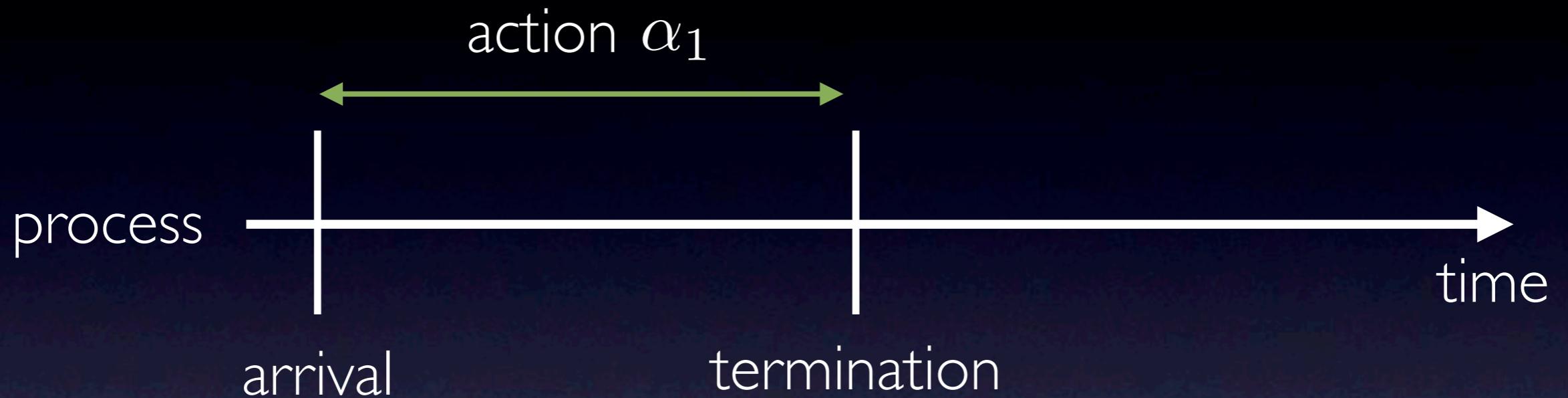


# Process model





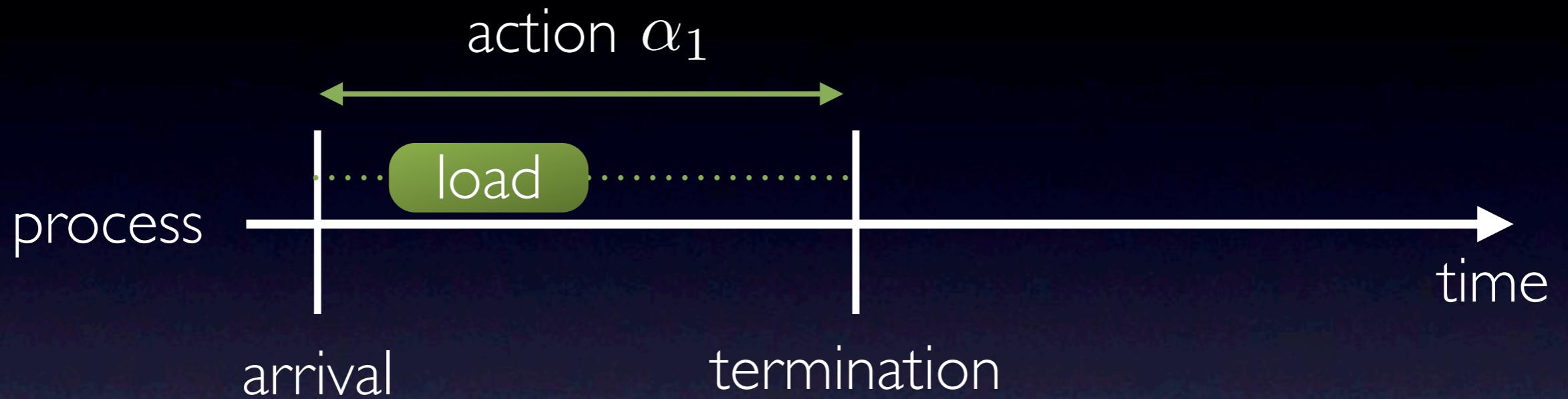
# Process model



- action is a piece of code



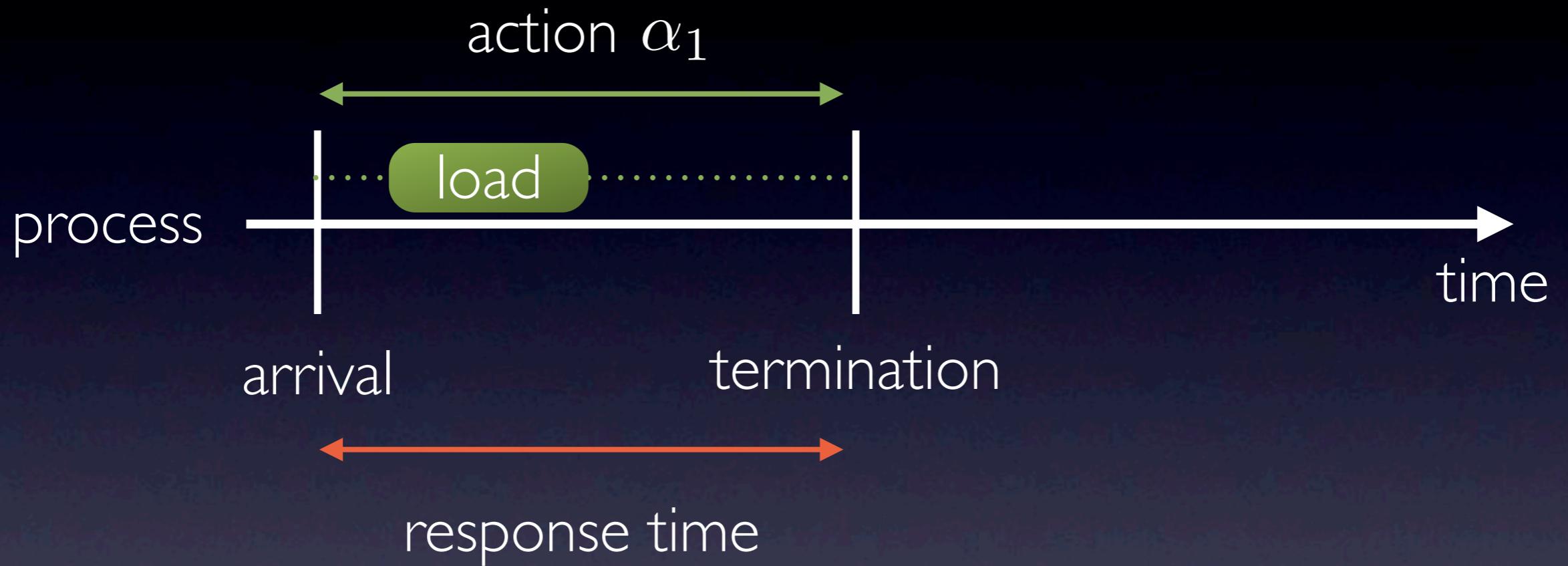
# Process model



- action is a piece of code



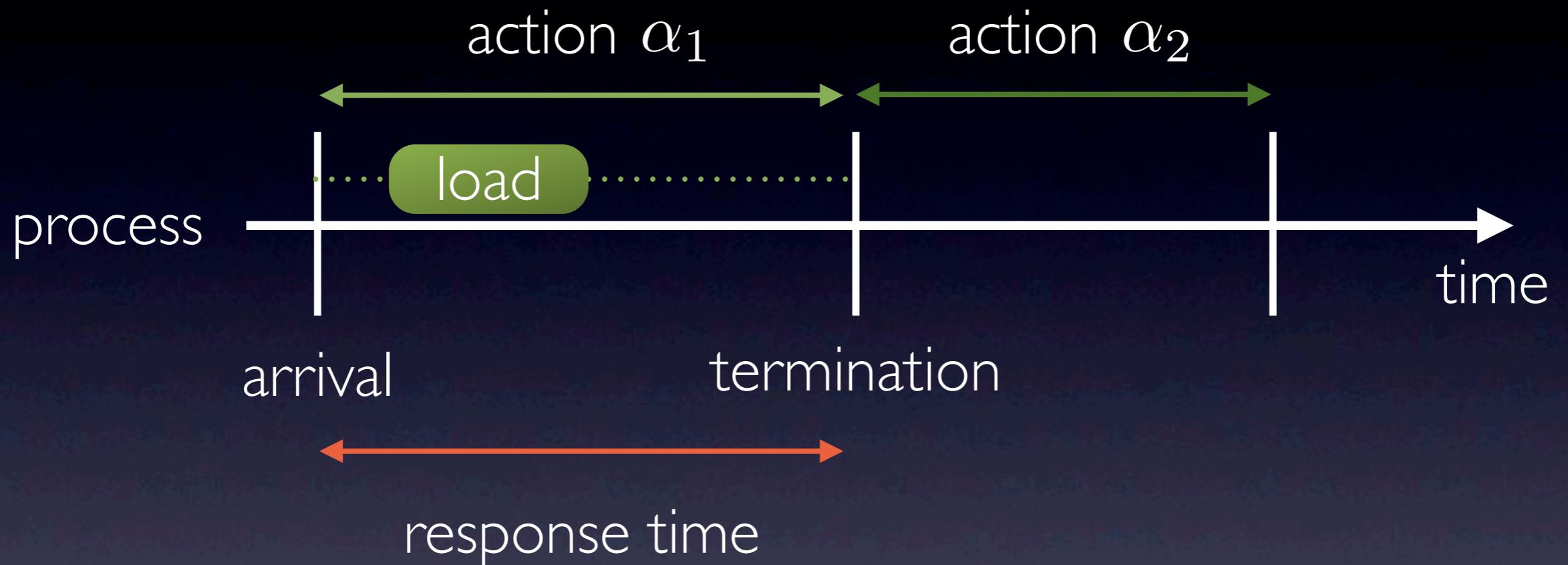
# Process model



- action is a piece of code



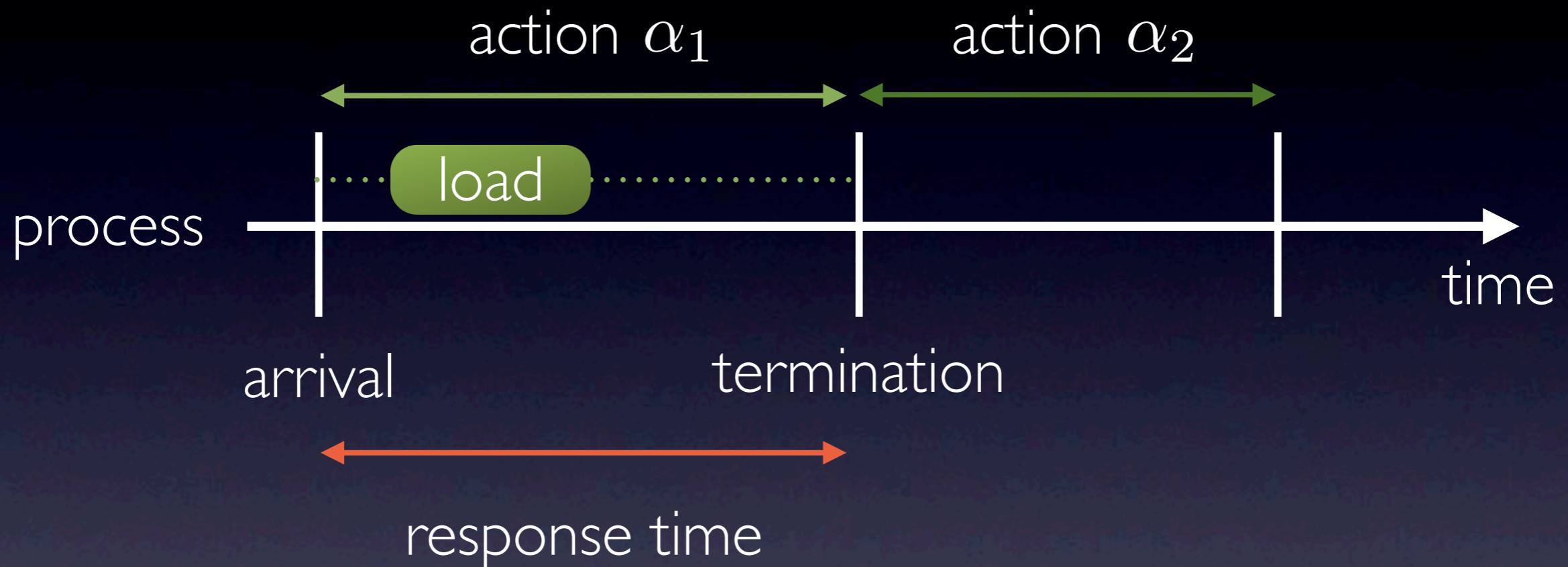
# Process model



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



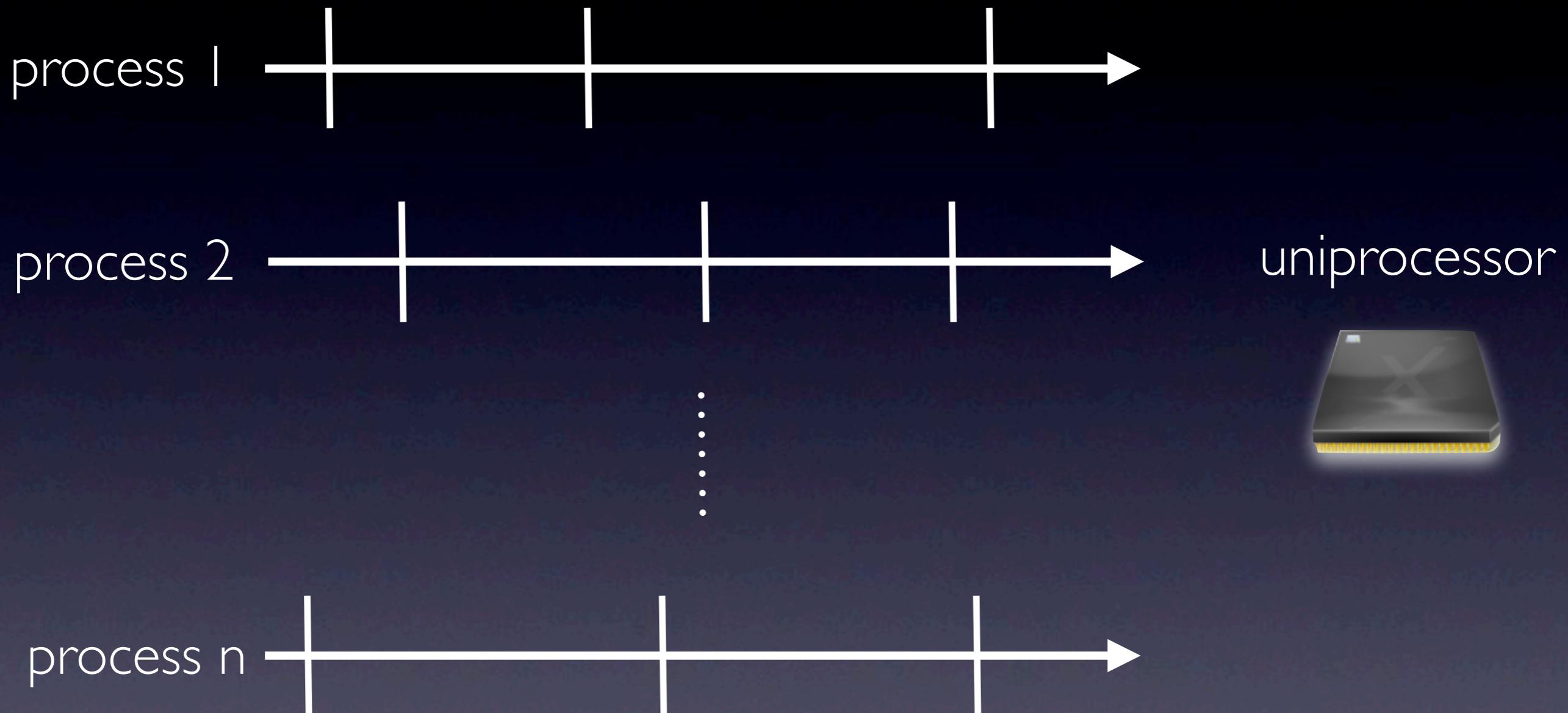
# Process model



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



# Scheduling problem

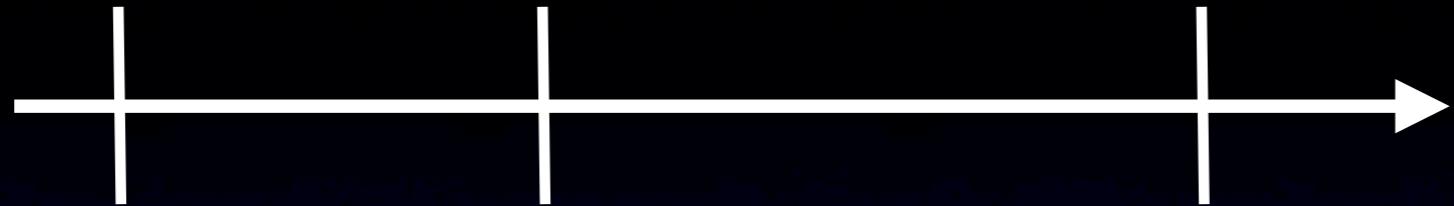


schedule the processes so that each of their actions maintains its response time



# Scheduling problem

process 1



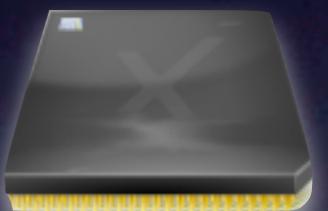
process 2

Solvable with variable-bandwidth servers  
(VBS)

process n



⋮



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



# Scheduling problem

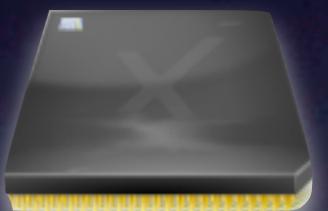
process 1



process 2

Solvable with variable-bandwidth servers  
(VBS)

uniprocessor



Results:

- 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  $\pi$    limit  $\lambda$    utilization  $\frac{\lambda}{\pi}$

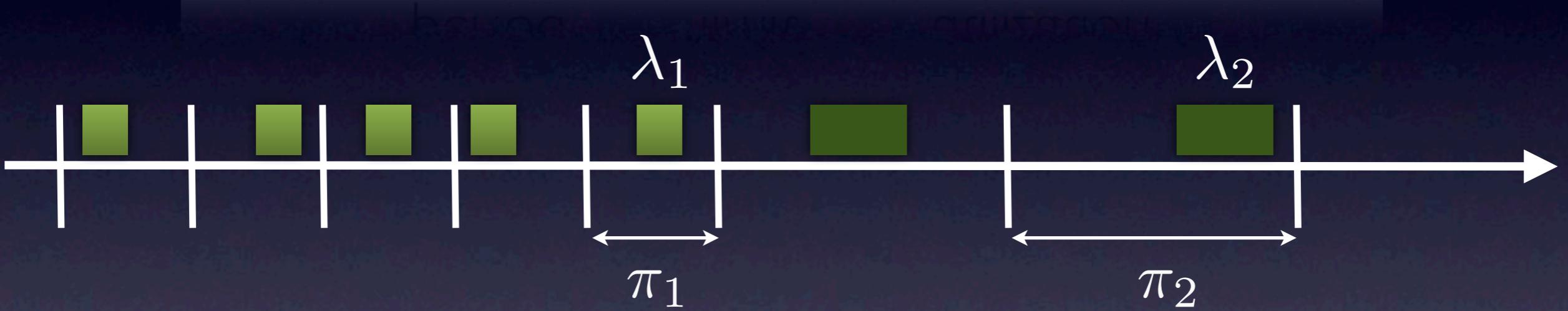
homogeneous periodic resources



# Resources and VBS

virtual periodic resources

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

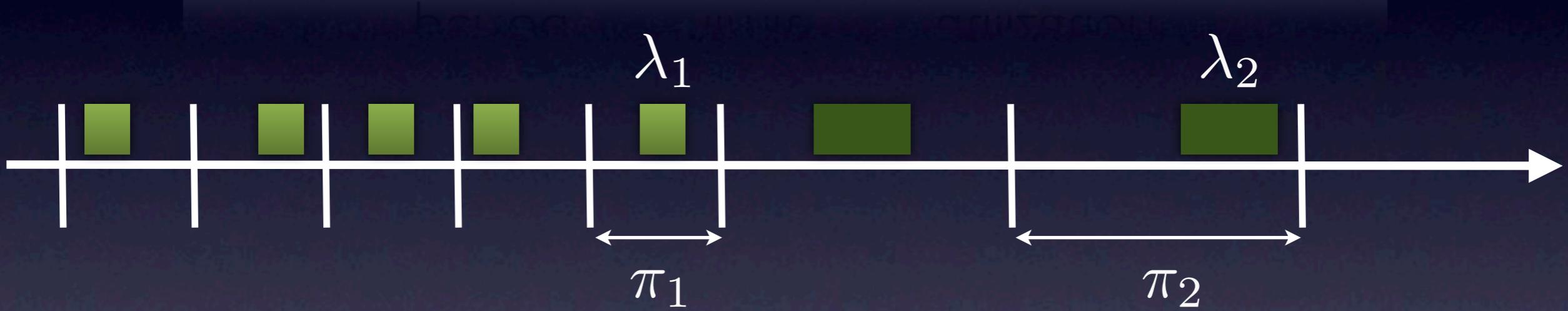




# Resources and VBS

virtual periodic resources

$$\text{period } \pi \quad \text{limit } \lambda \quad \text{utilization } \frac{\lambda}{\pi}$$



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

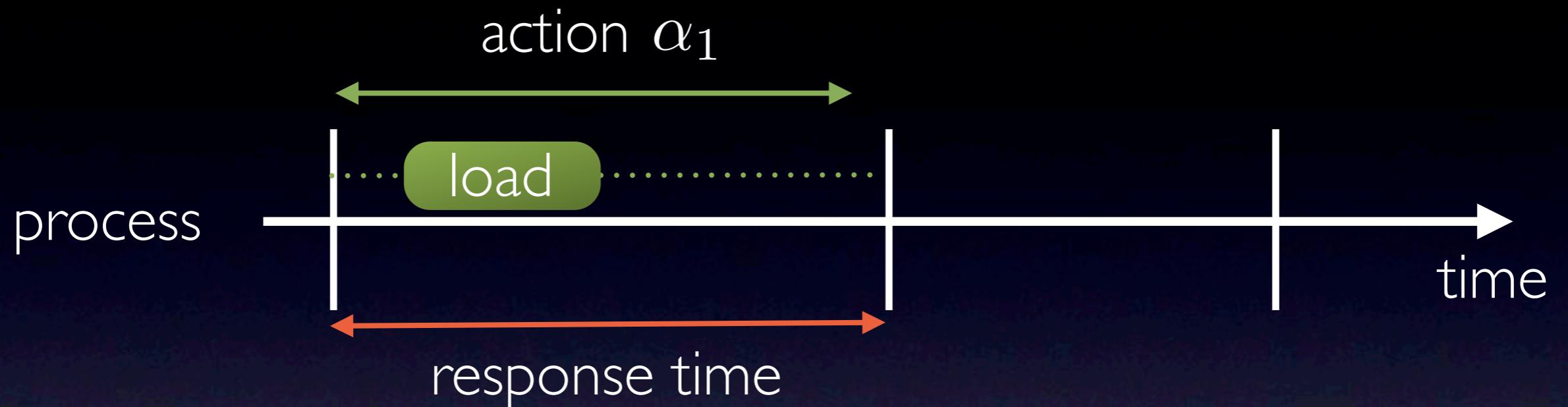
$$\frac{\lambda_1}{\pi_1} \leq u \quad \frac{\lambda_2}{\pi_2} \leq u$$

- generalization of constant bandwidth servers (CBS)

[Abeni and Buttazzo 2004]

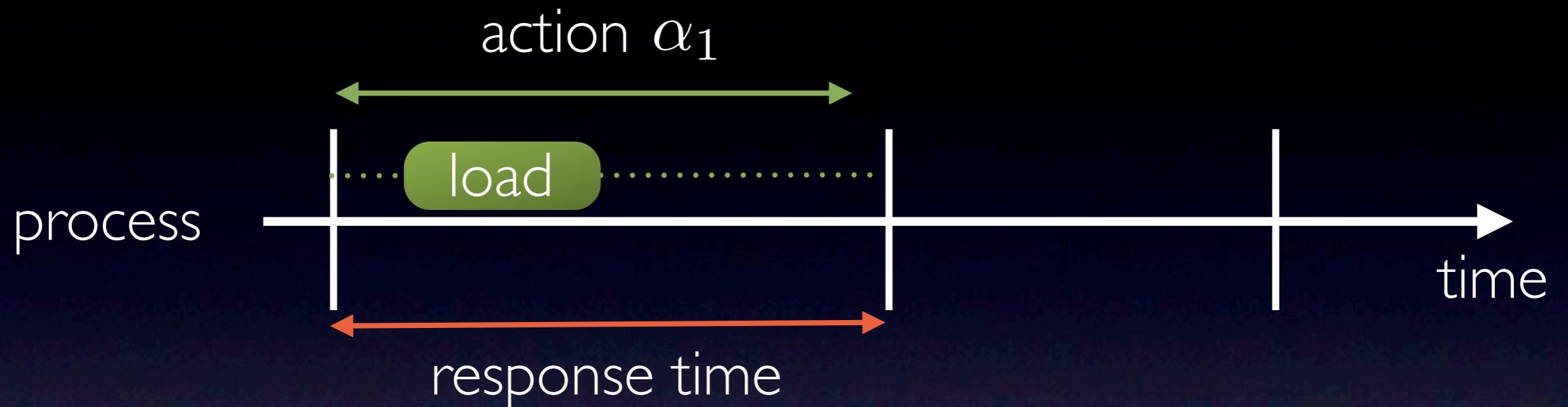


# One process on a VBS



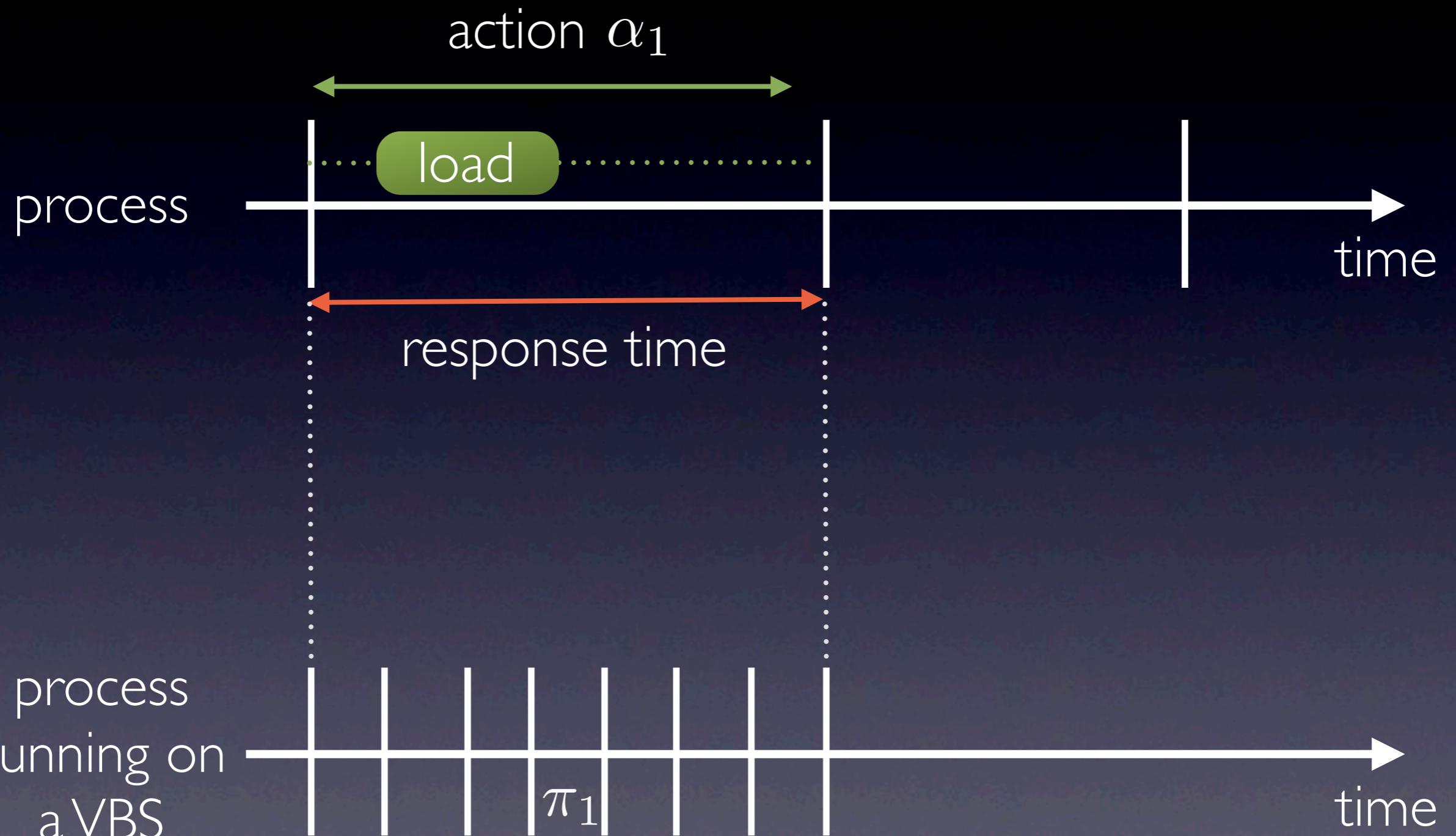


# One process on a VBS



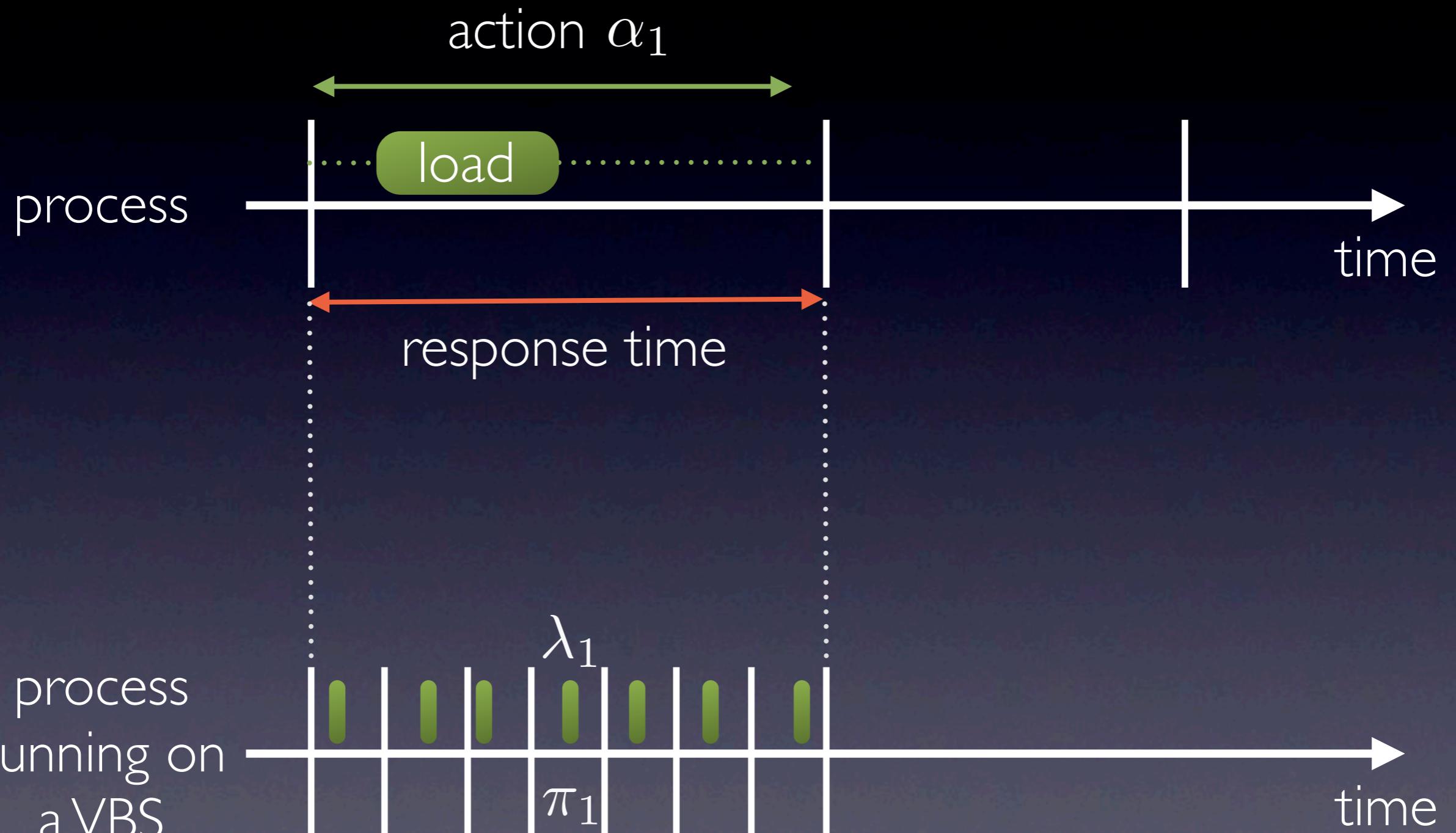


# One process on a VBS



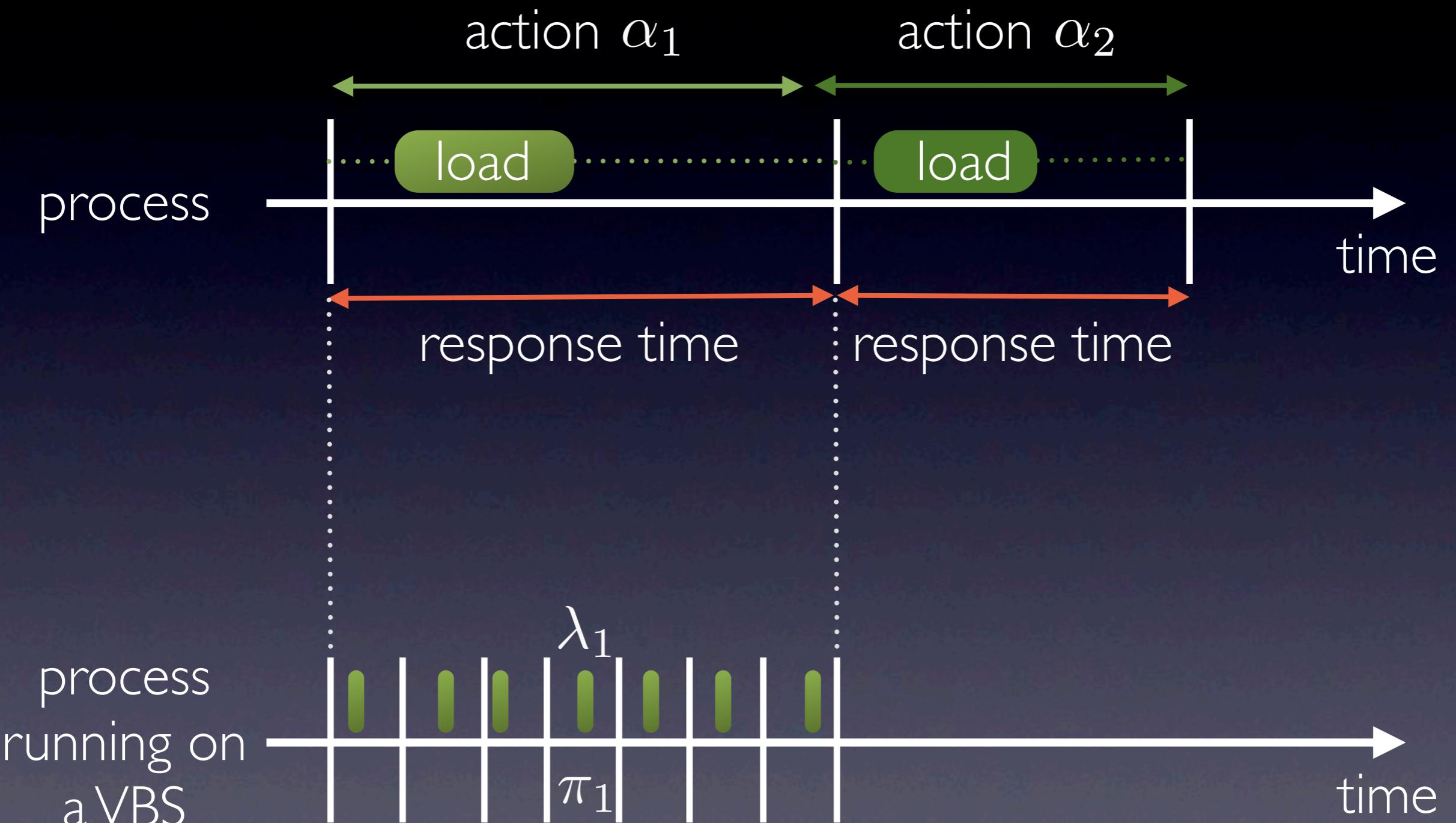


# One process on a VBS



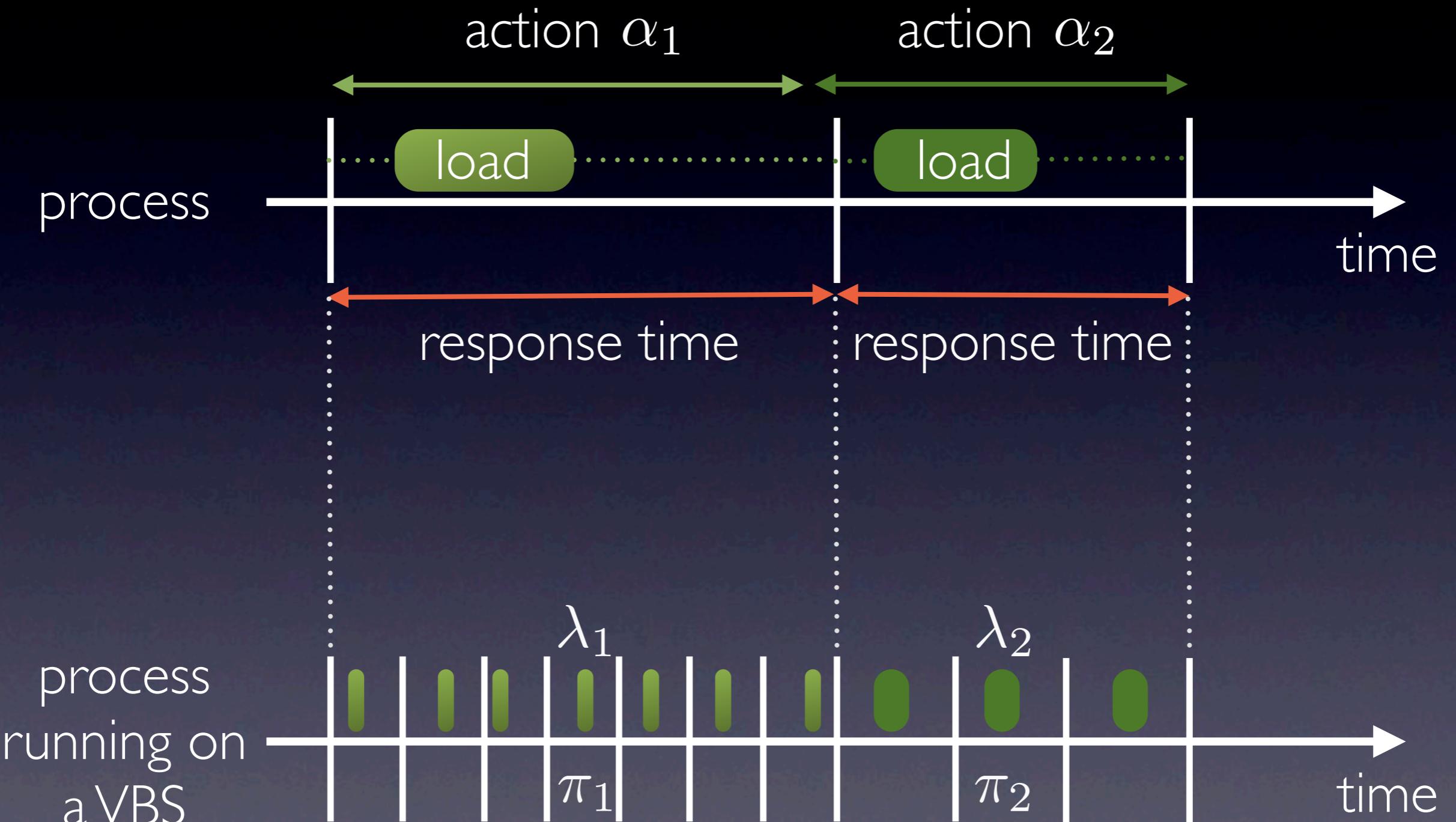


# One process on a VBS





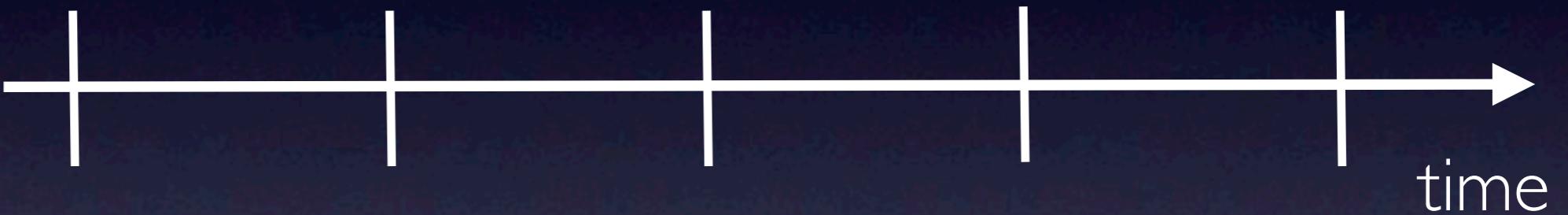
# One process on a VBS





# VBS

process  
running on  
a VBS



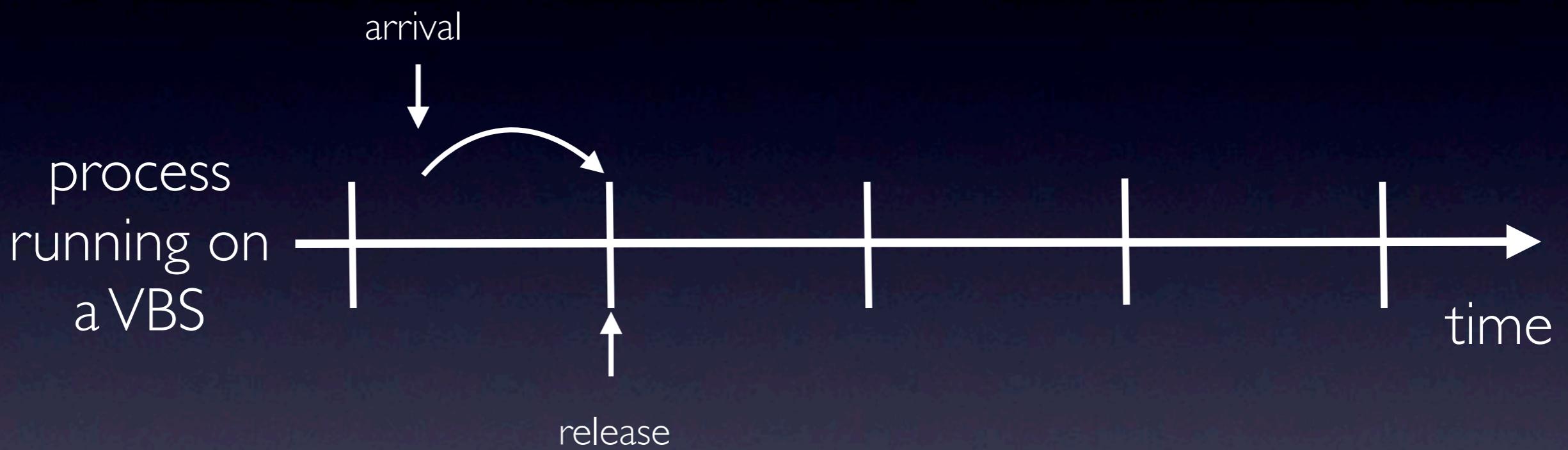


# VBS



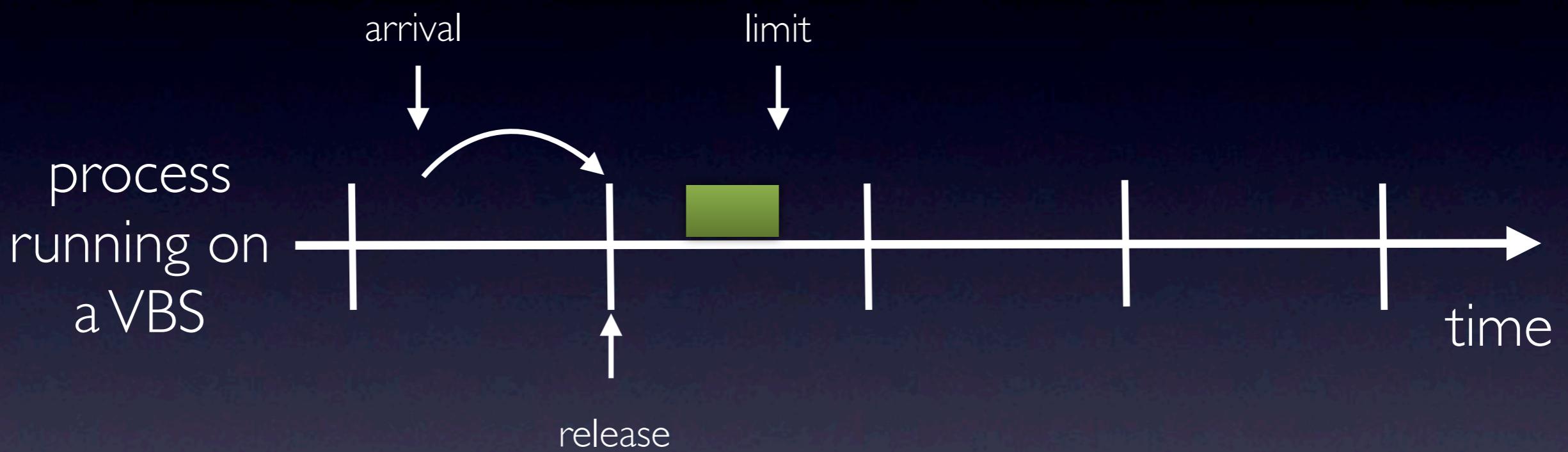


# VBS



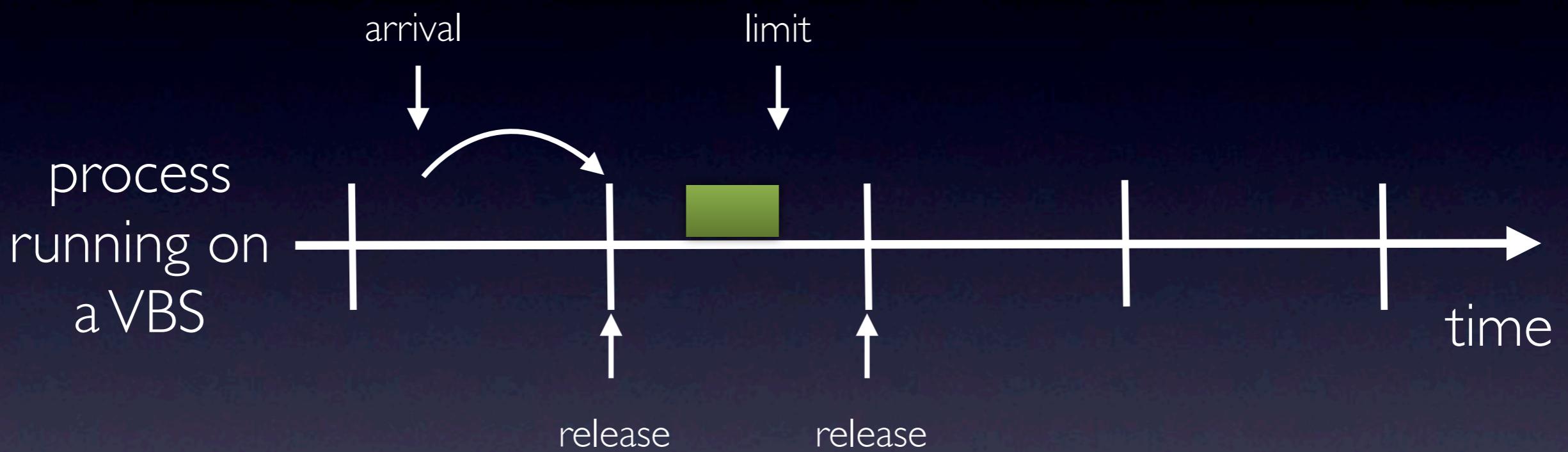


# VBS





# VBS



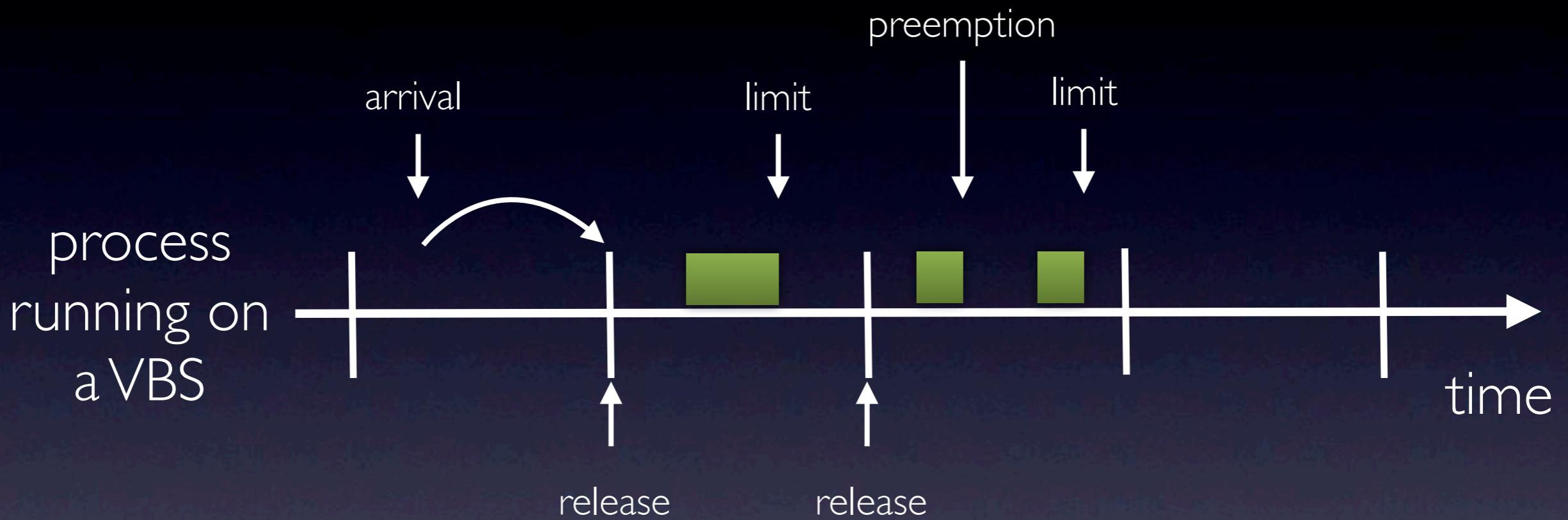


# VBS



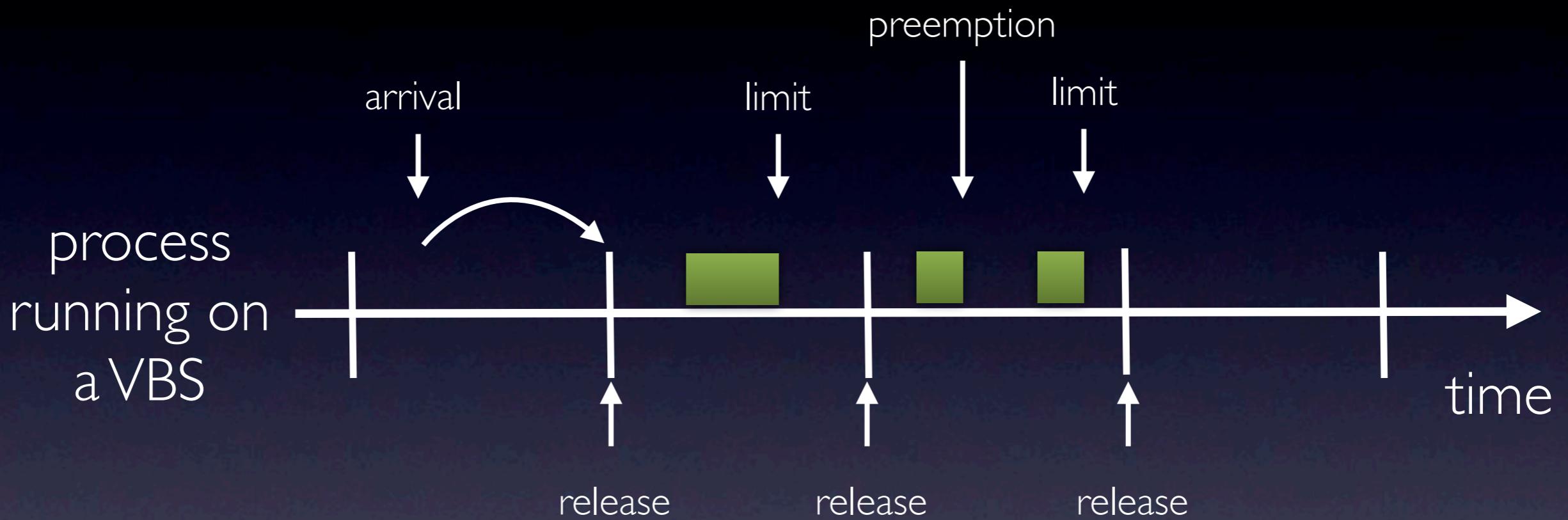


# VBS



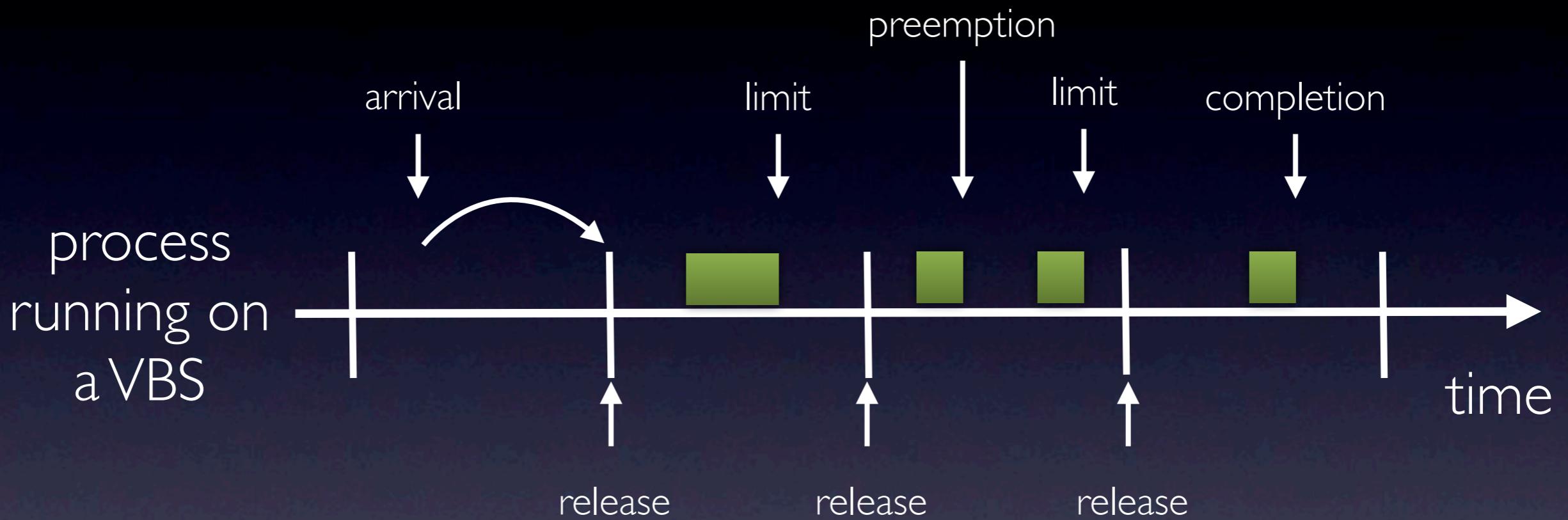


# VBS



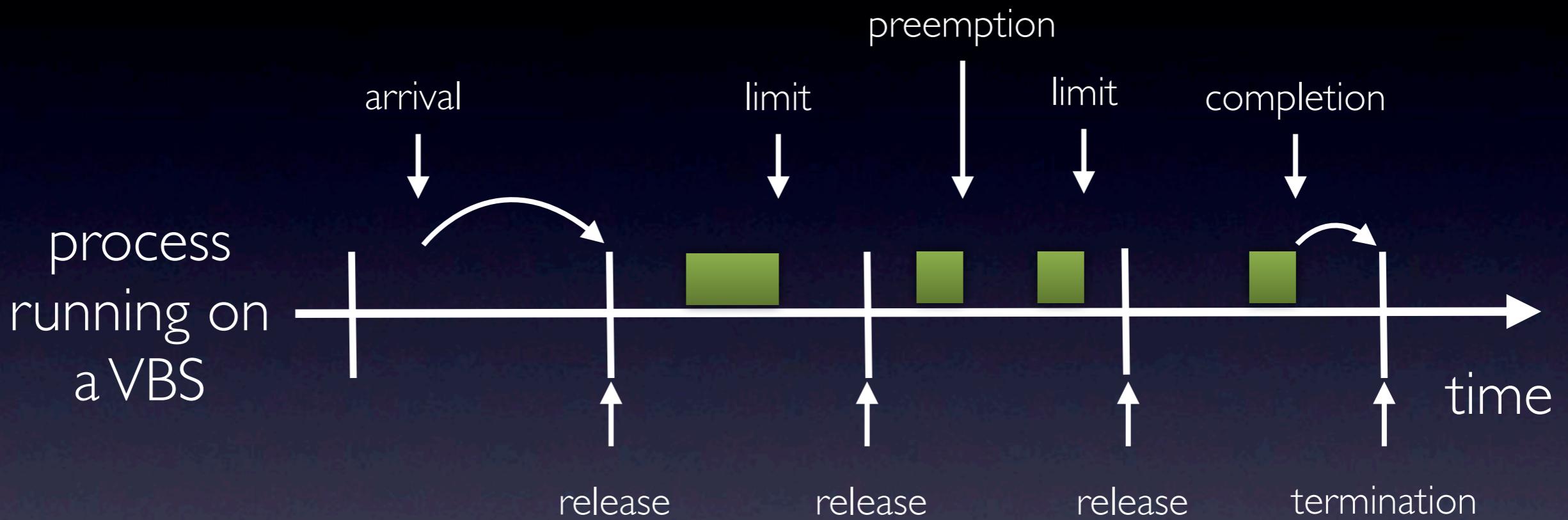


# VBS



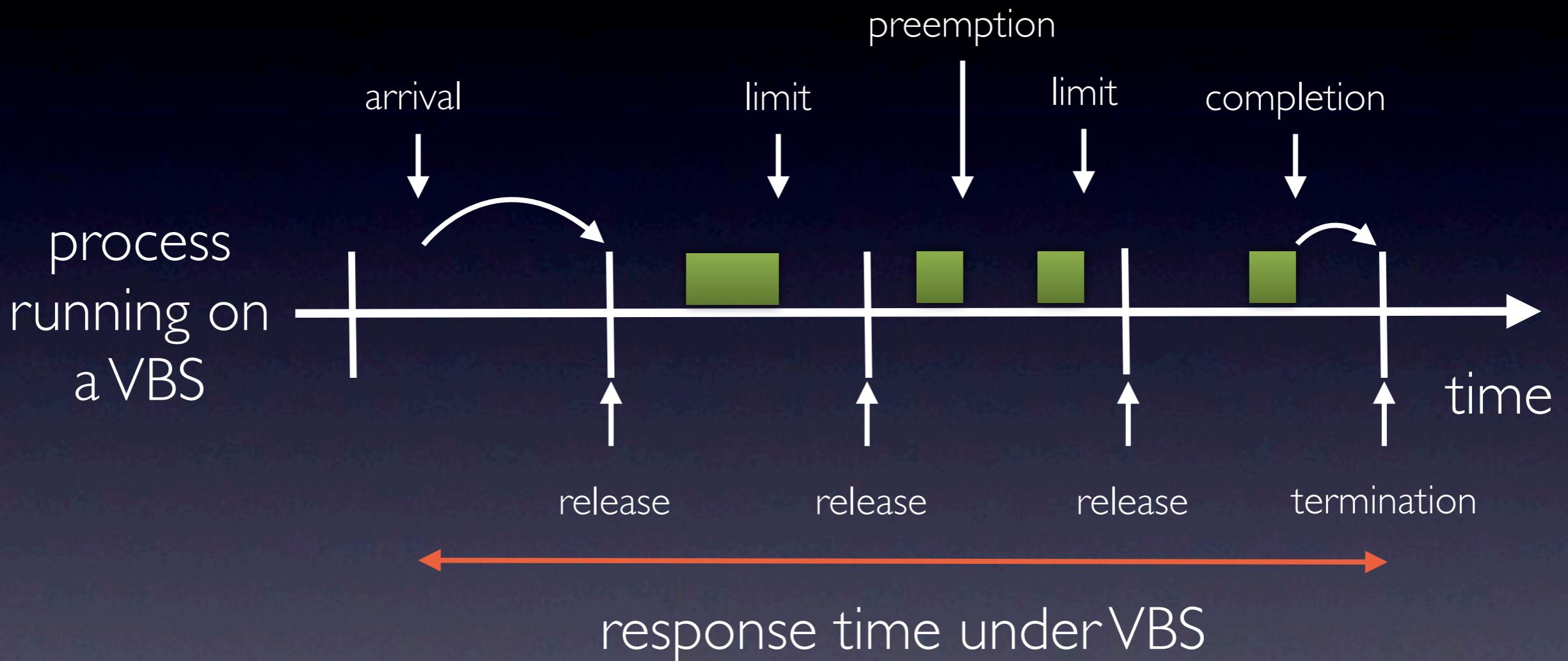


# VBS



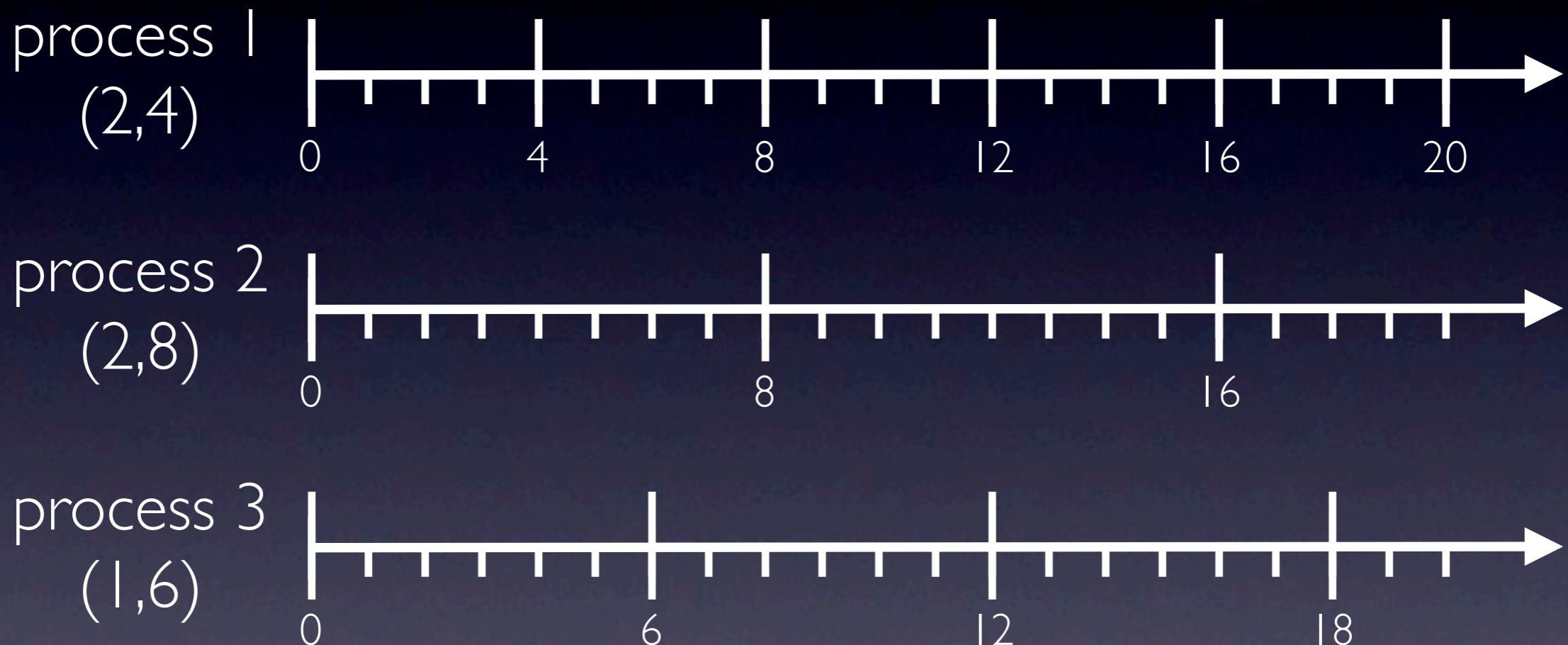


# VBS





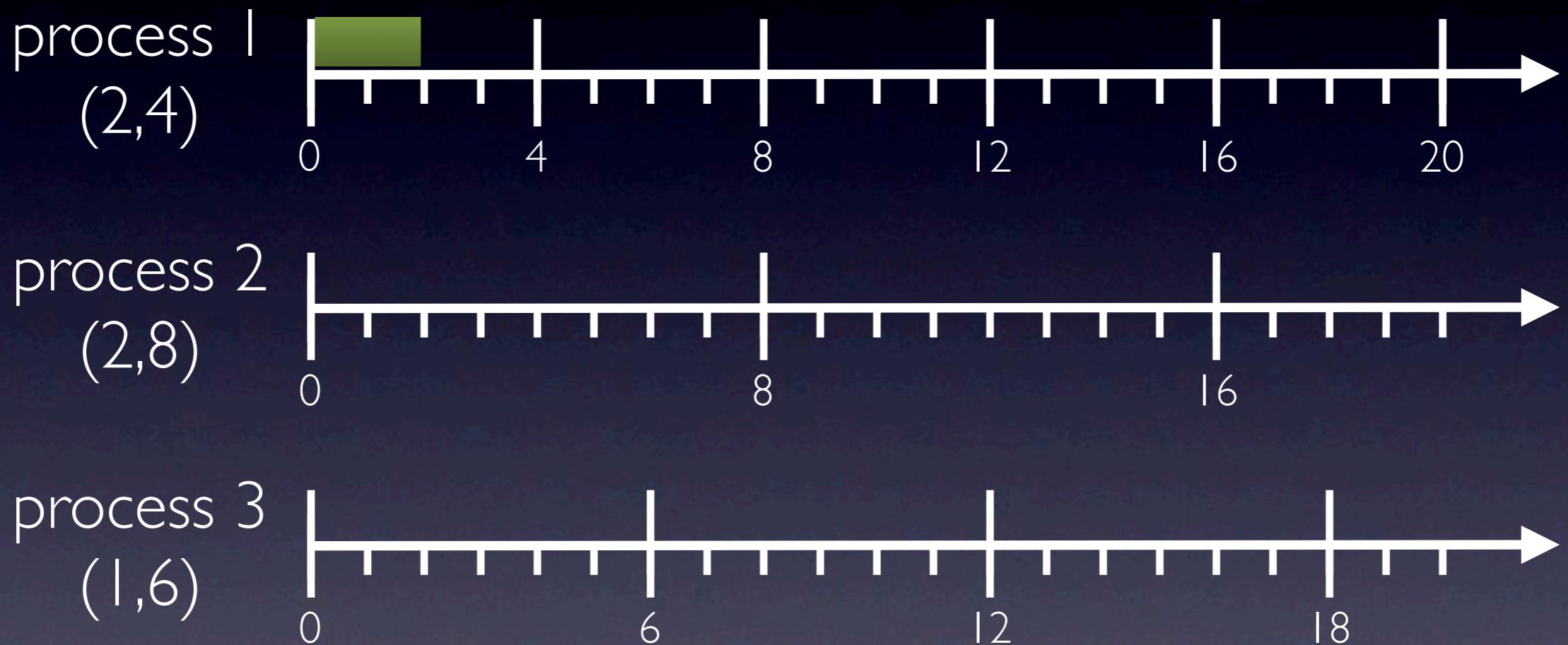
# VBS



multiple processes are EDF-scheduled



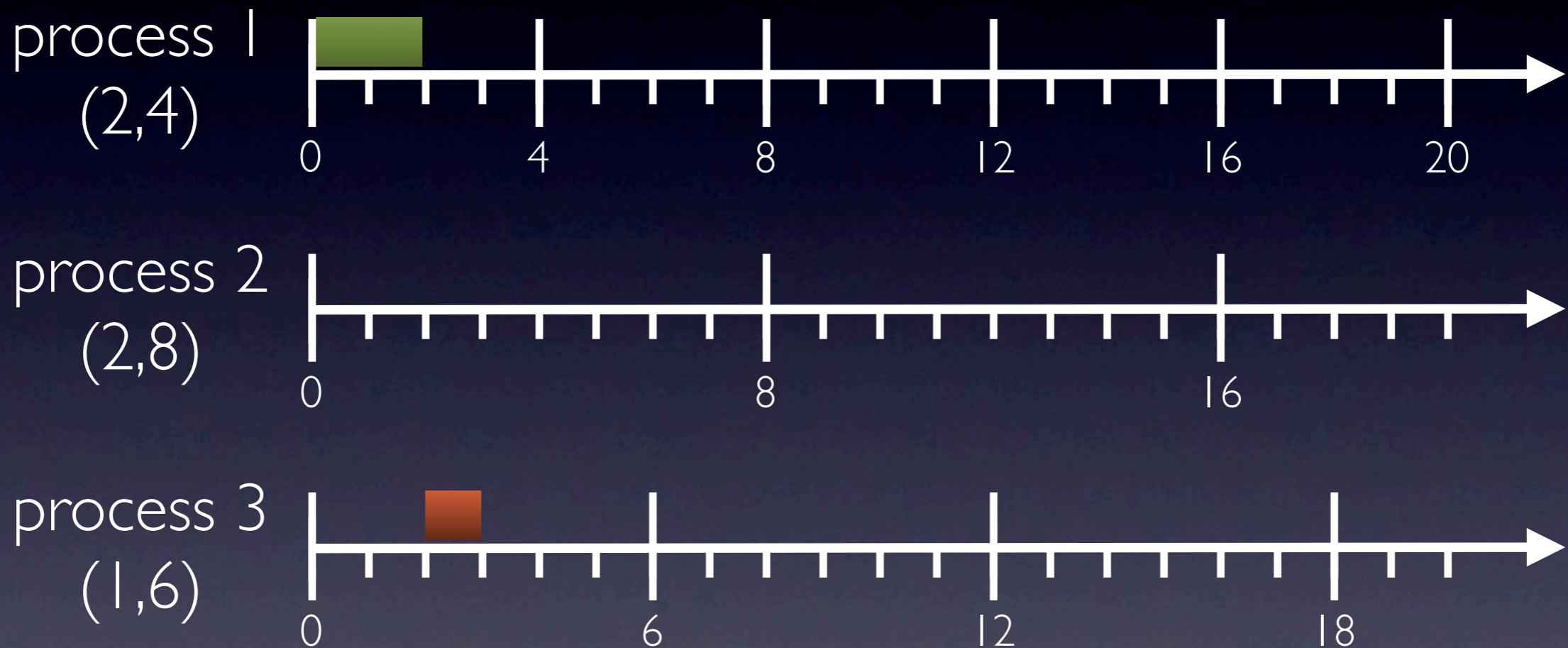
# VBS



multiple processes are EDF-scheduled



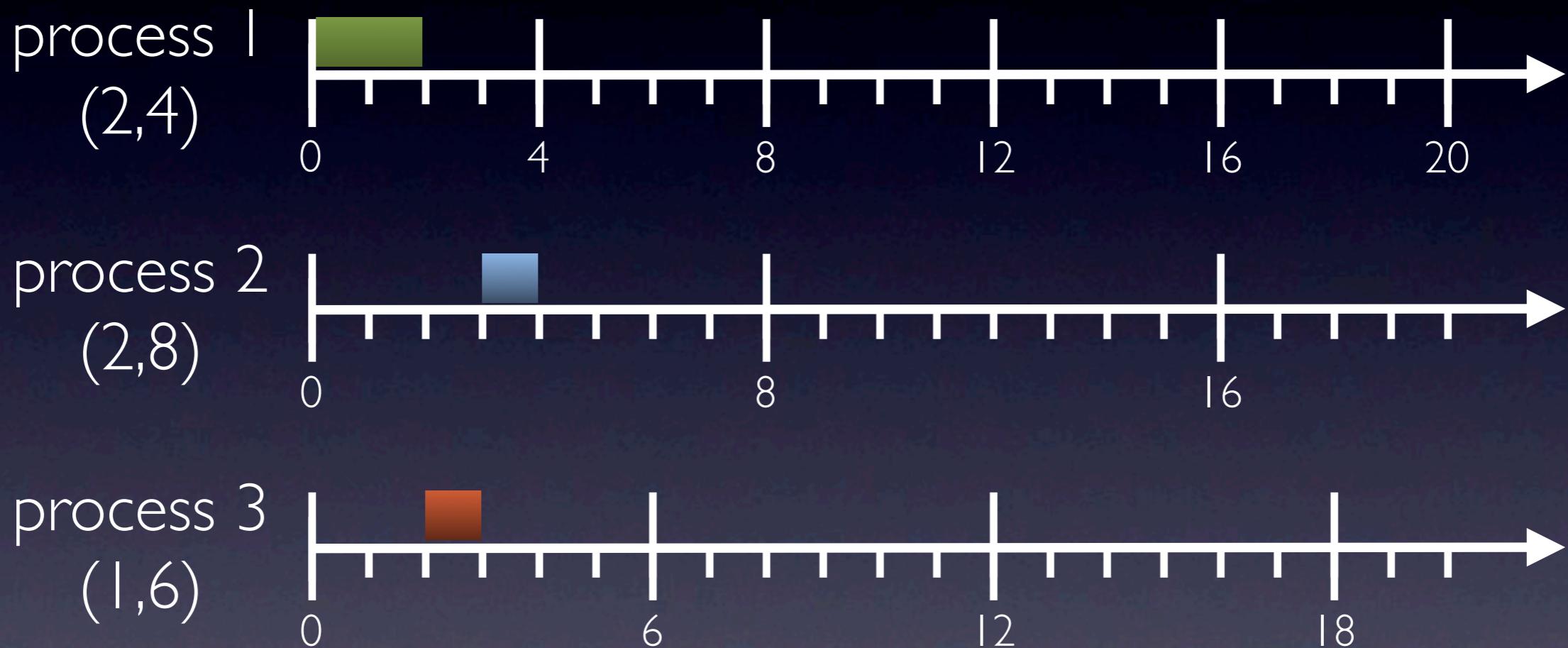
# VBS



multiple processes are EDF-scheduled



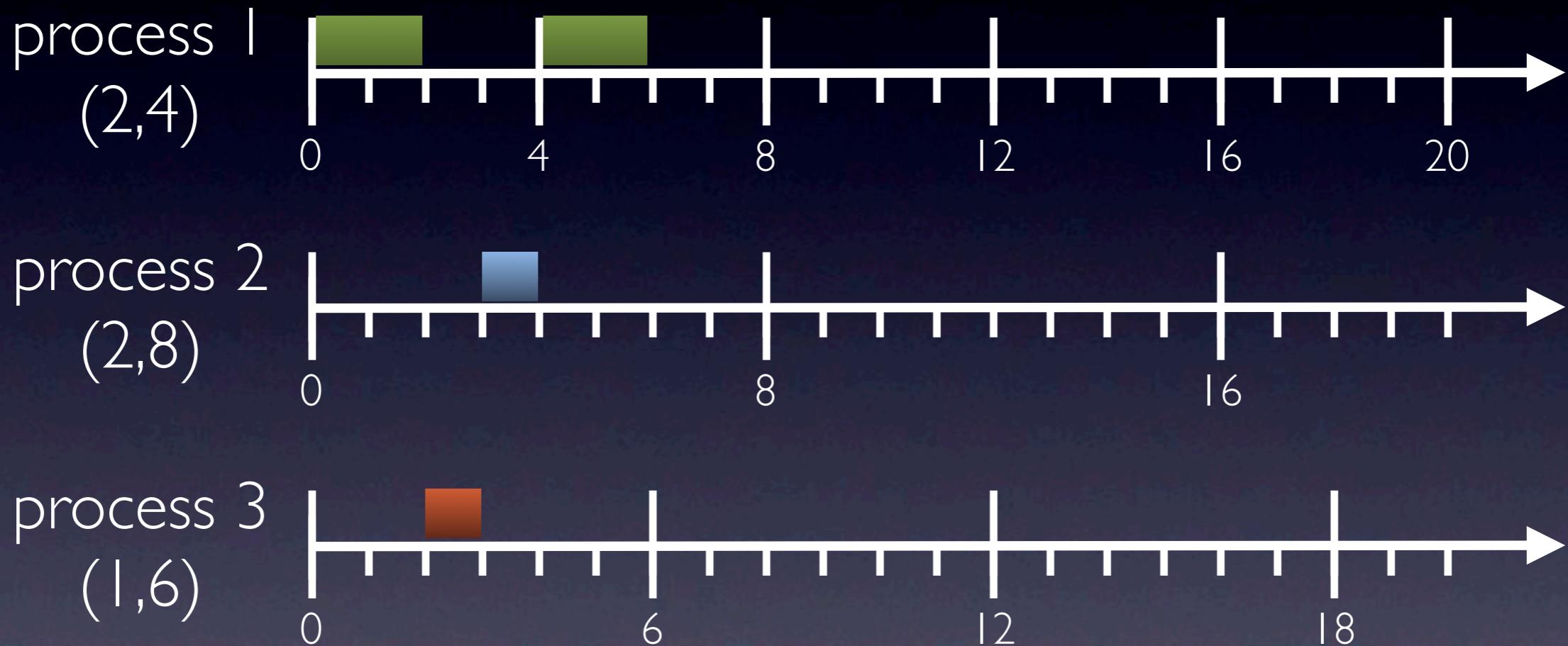
# VBS



multiple processes are EDF-scheduled



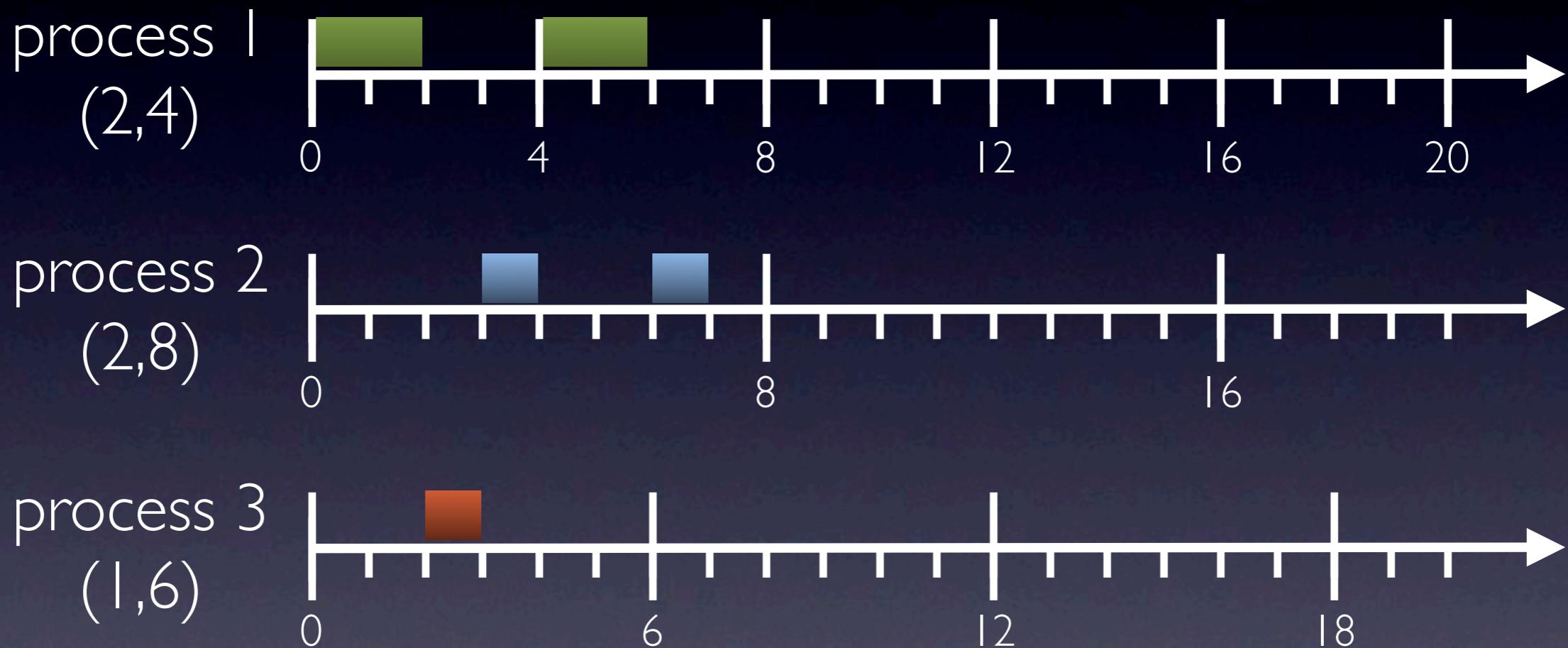
# VBS



multiple processes are EDF-scheduled



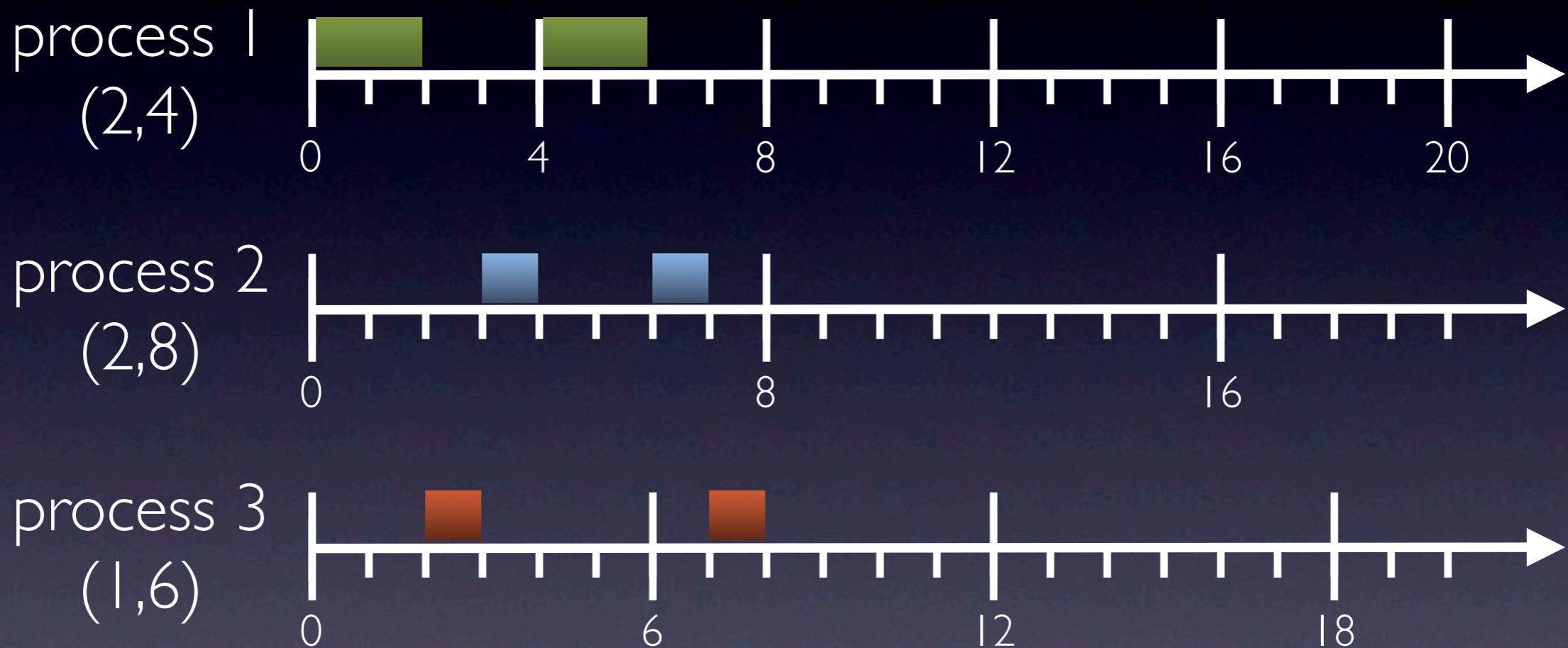
# VBS



multiple processes are EDF-scheduled



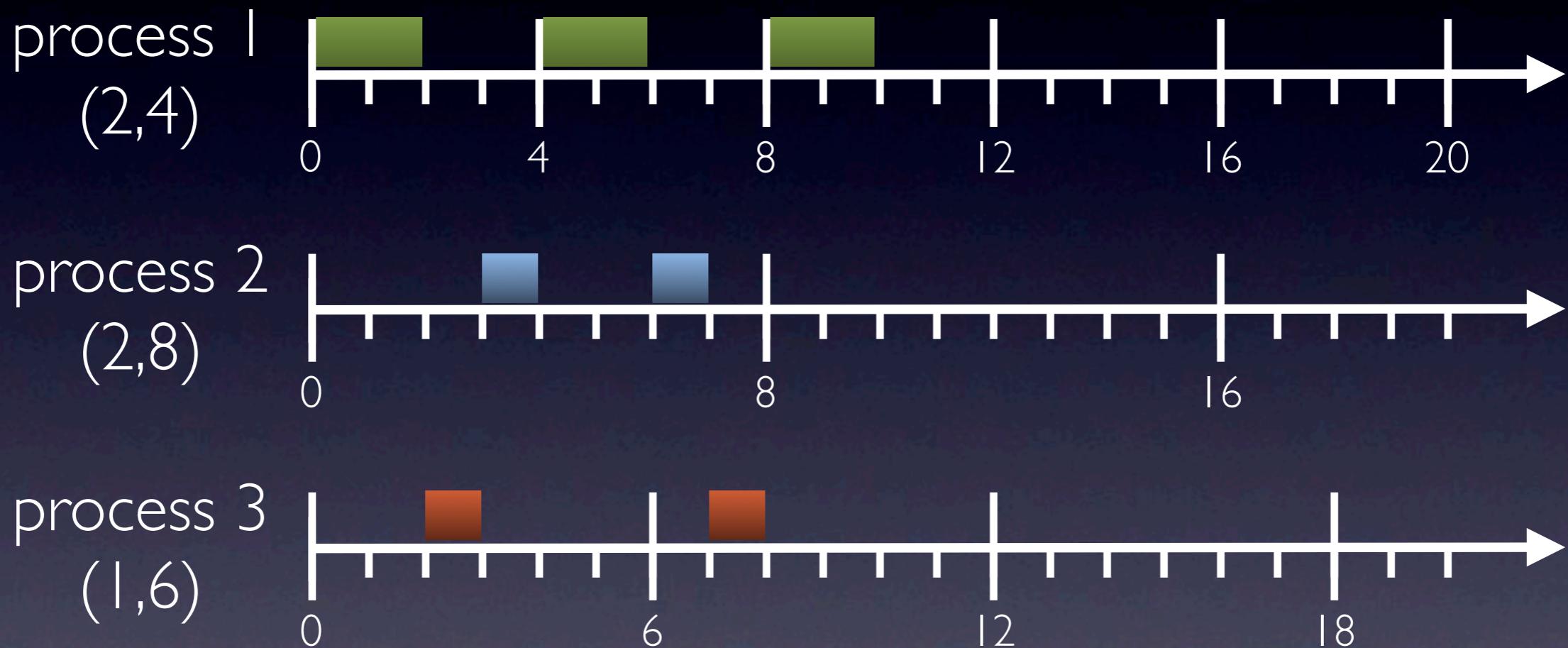
# VBS



multiple processes are EDF-scheduled



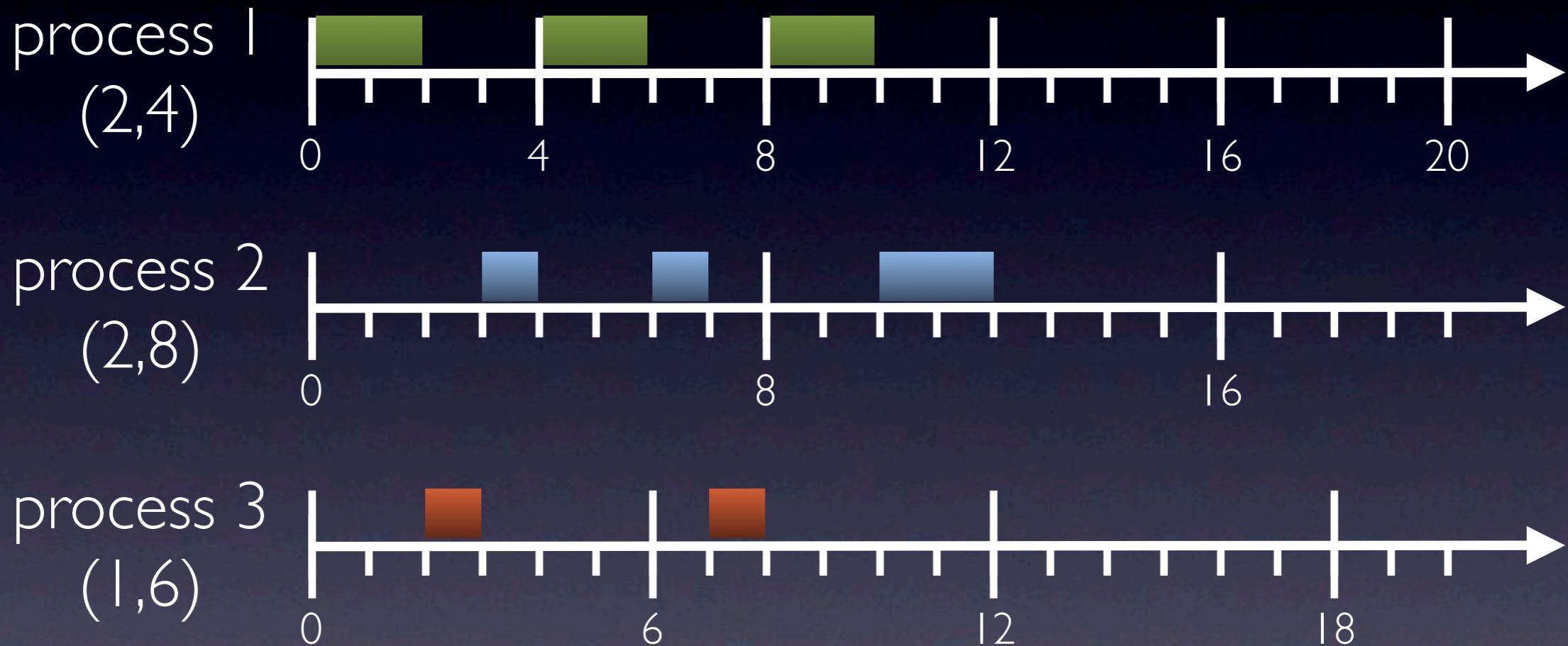
# VBS



multiple processes are EDF-scheduled



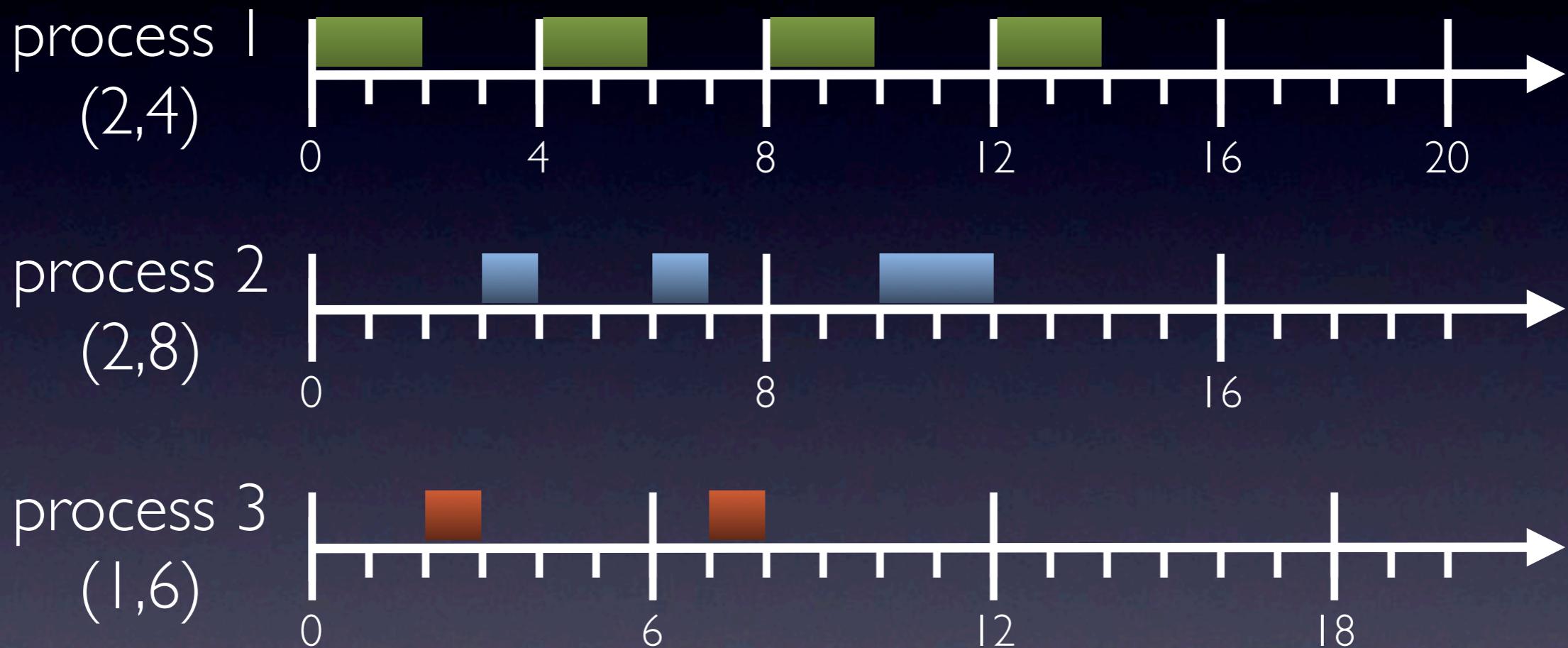
# VBS



multiple processes are EDF-scheduled



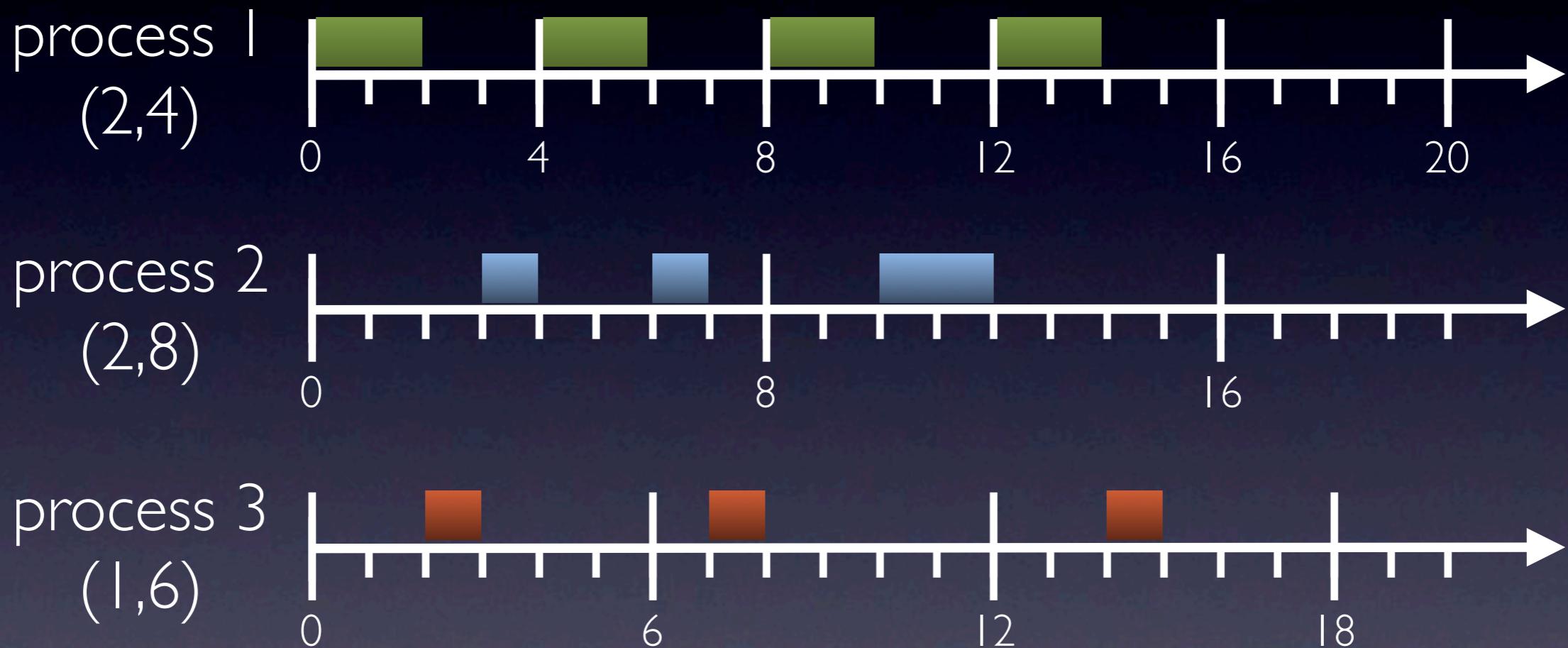
# VBS



multiple processes are EDF-scheduled



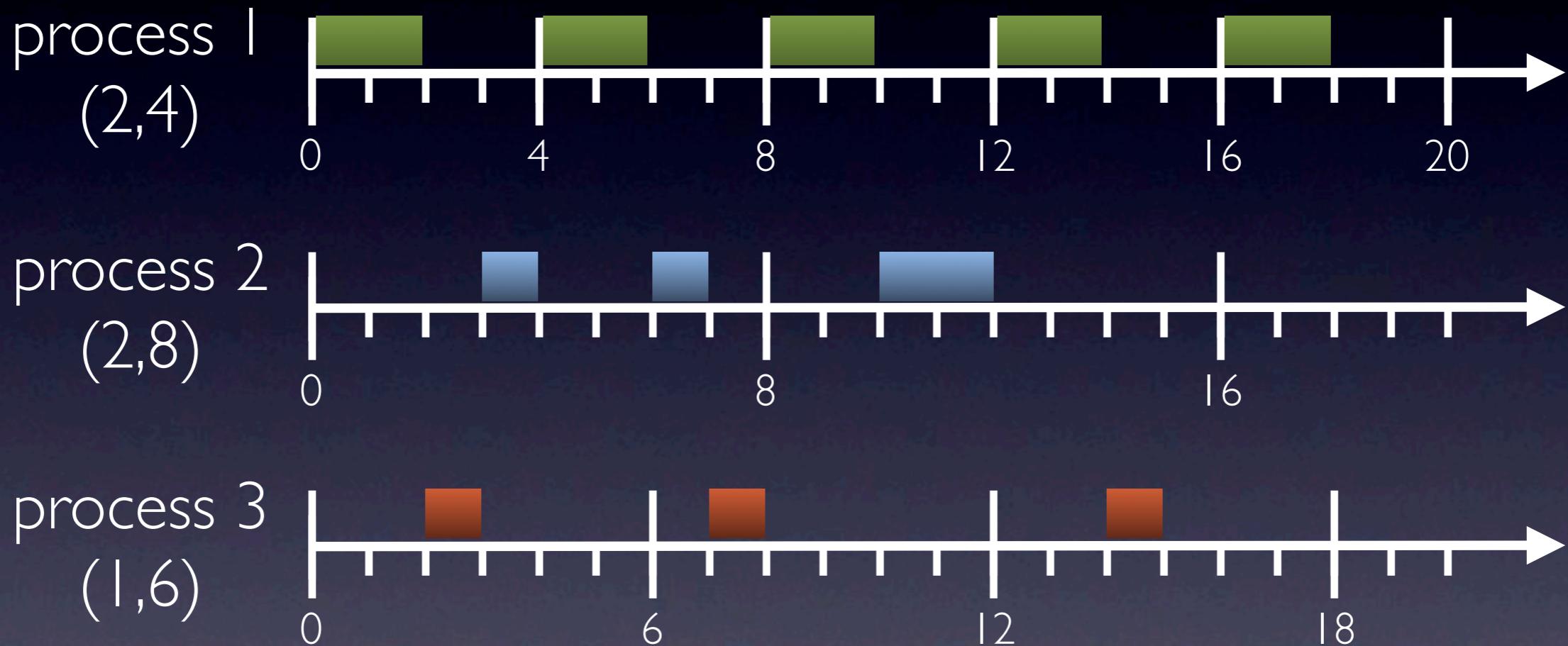
# VBS



multiple processes are EDF-scheduled



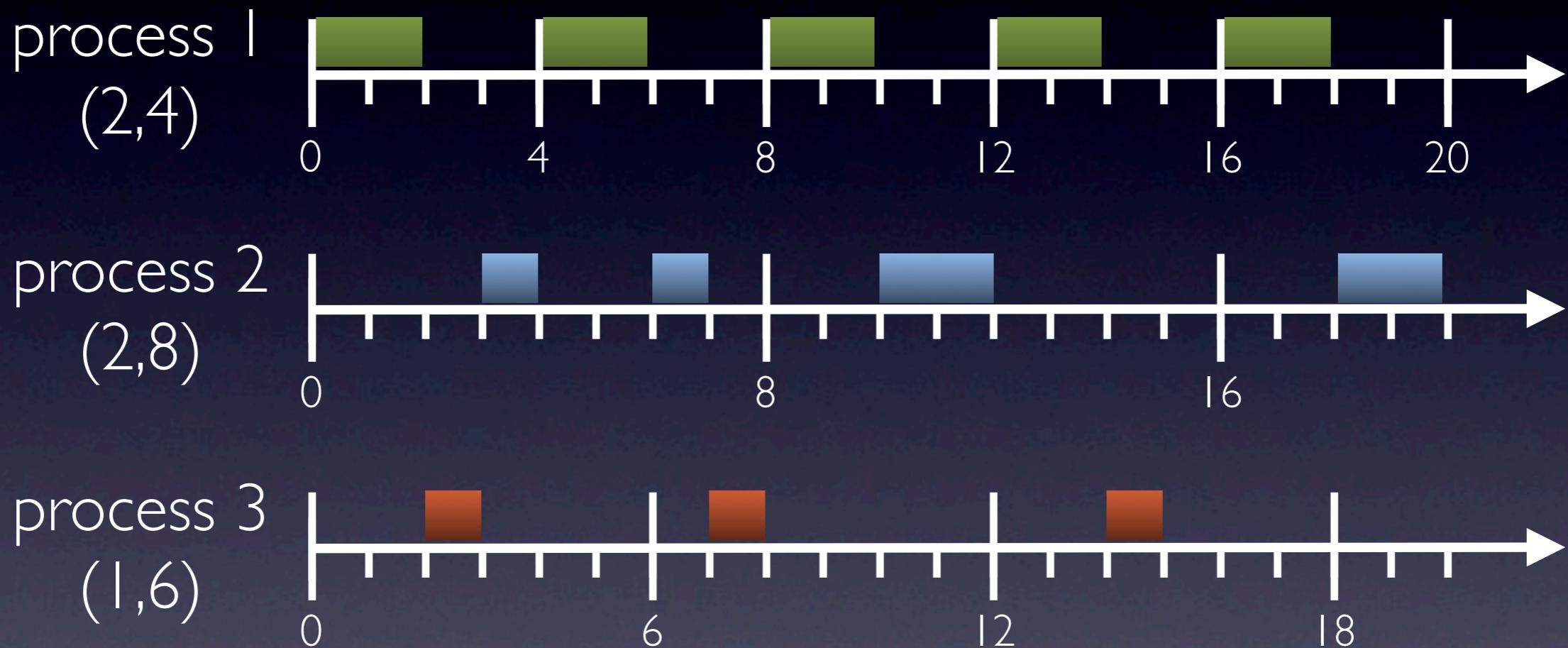
# VBS



multiple processes are EDF-scheduled



# VBS



multiple processes are EDF-scheduled



# Scheduling result and bounds

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



# Scheduling result and bounds

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

For any action  $\alpha$  on a resource  $(\lambda, \pi)$  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

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

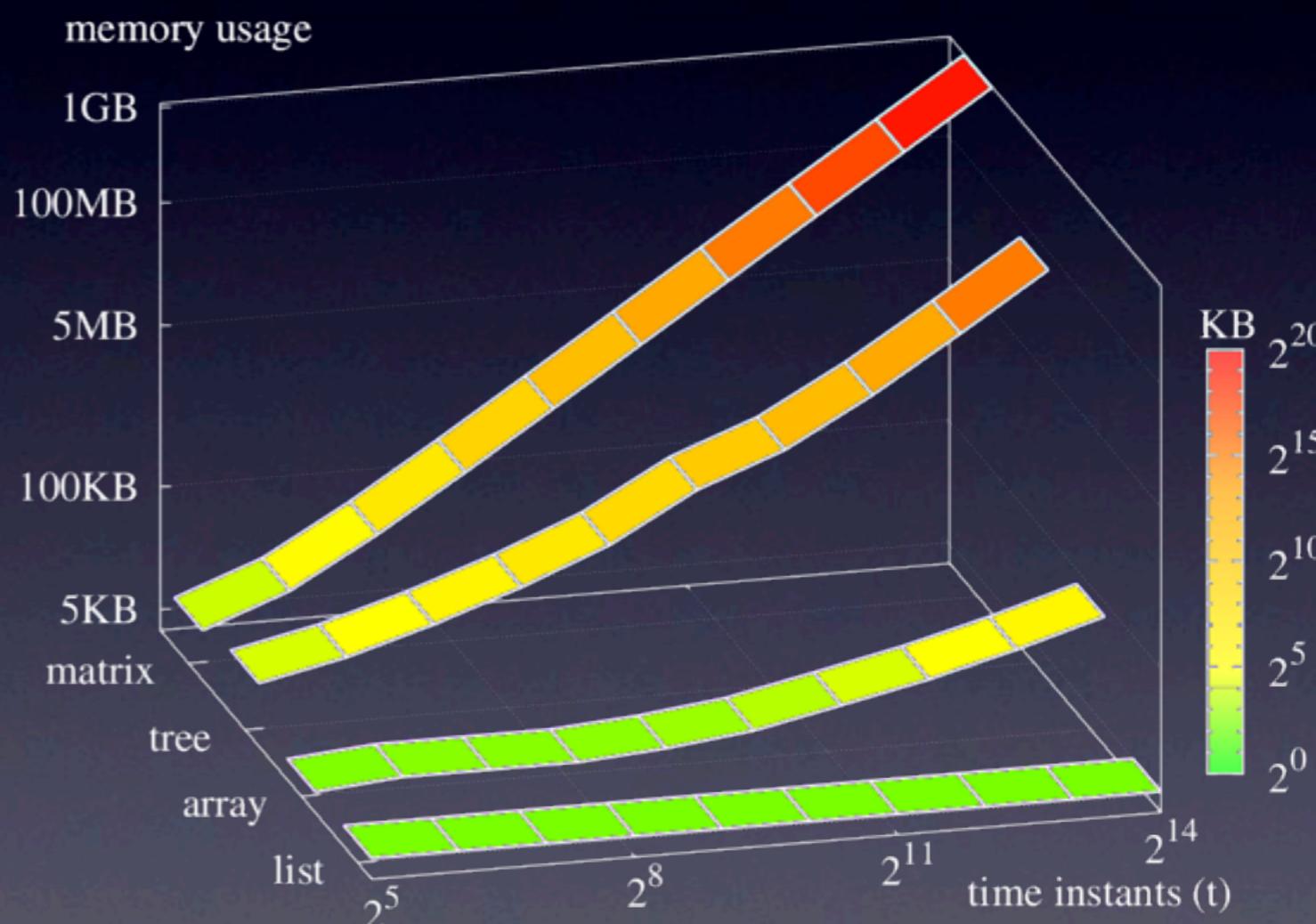
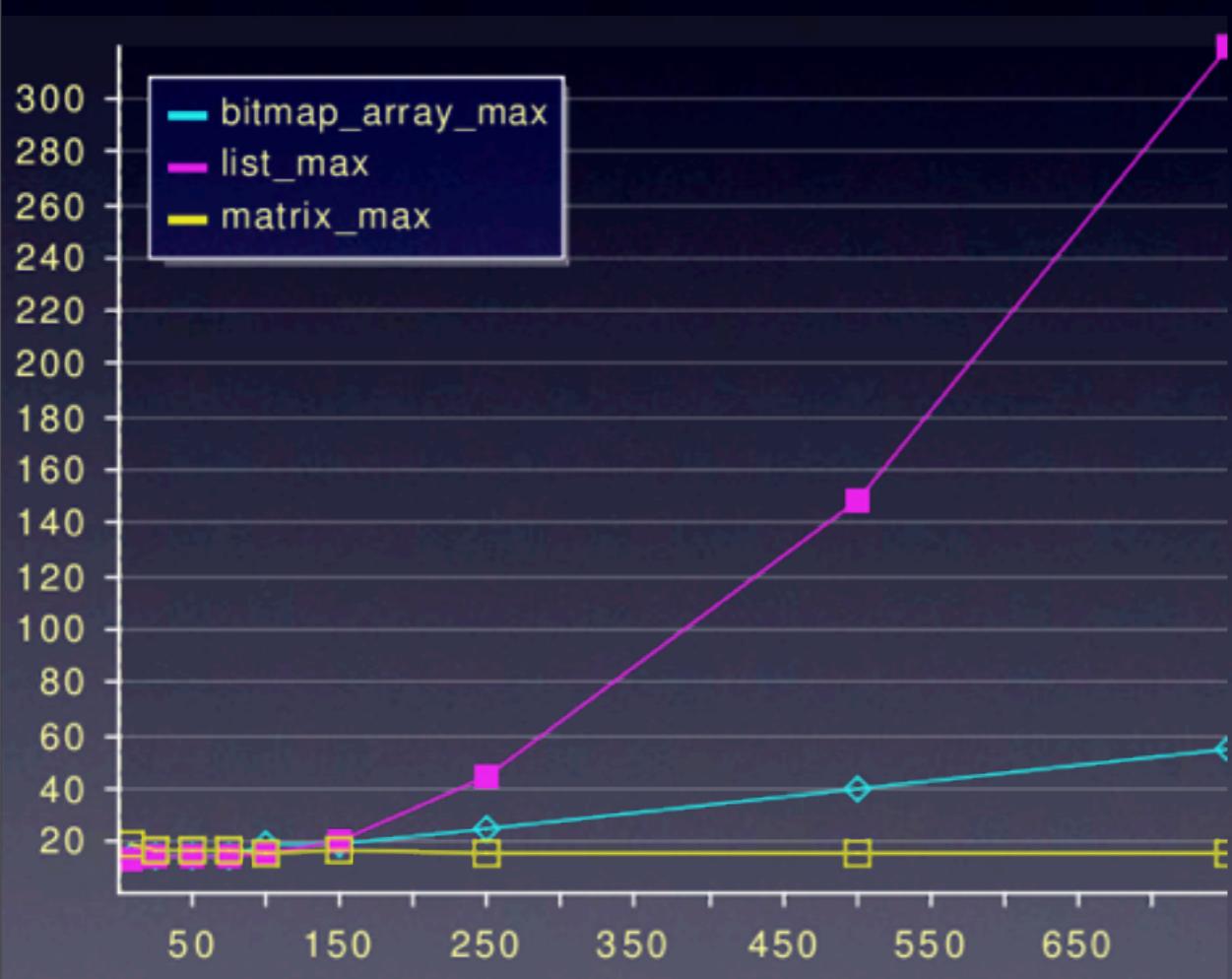
For any action  $\alpha$  on a resource  $(\lambda, \pi)$  we have:

- 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  $\pi - 1$

temporal isolation

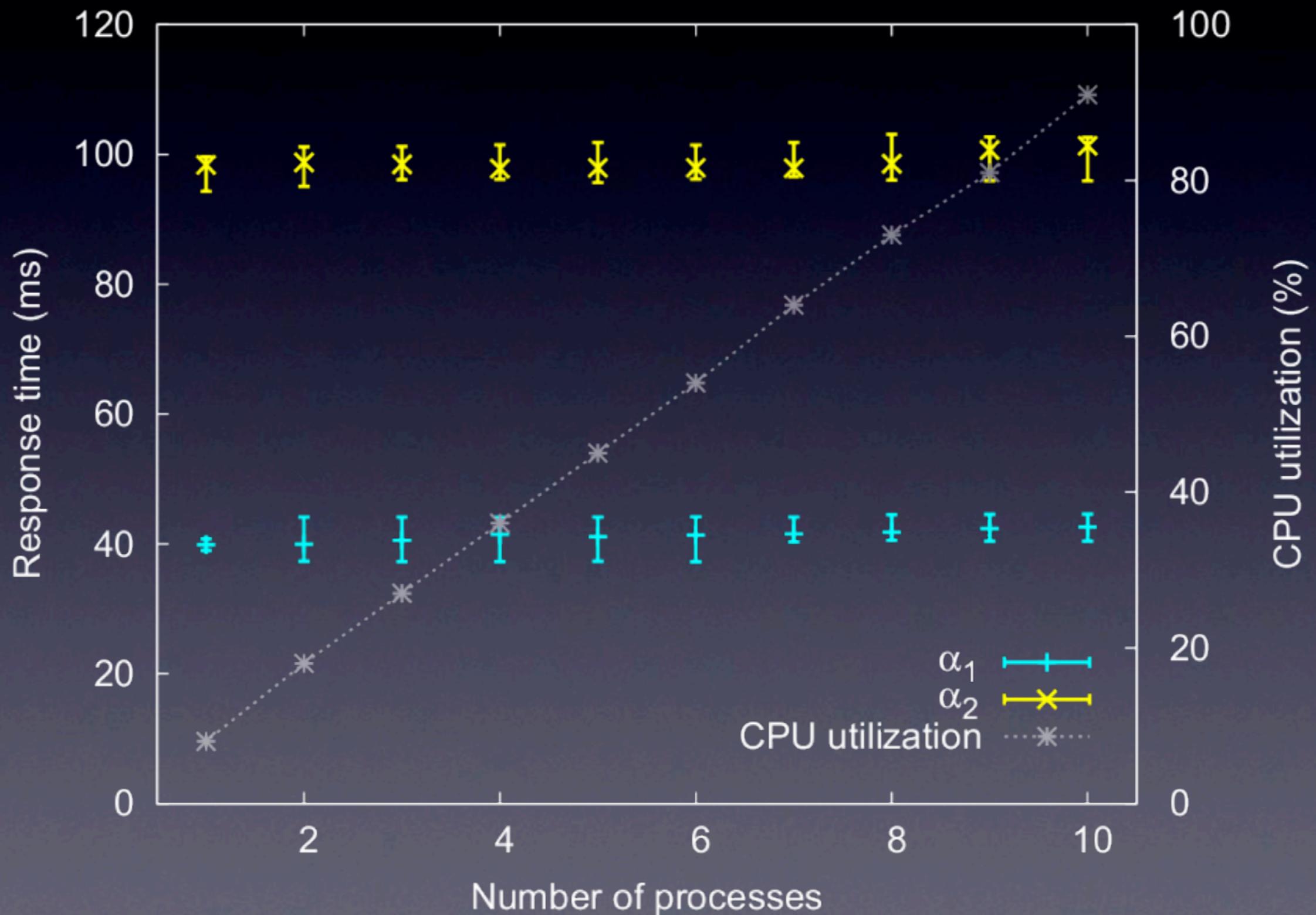


# Scheduler overhead



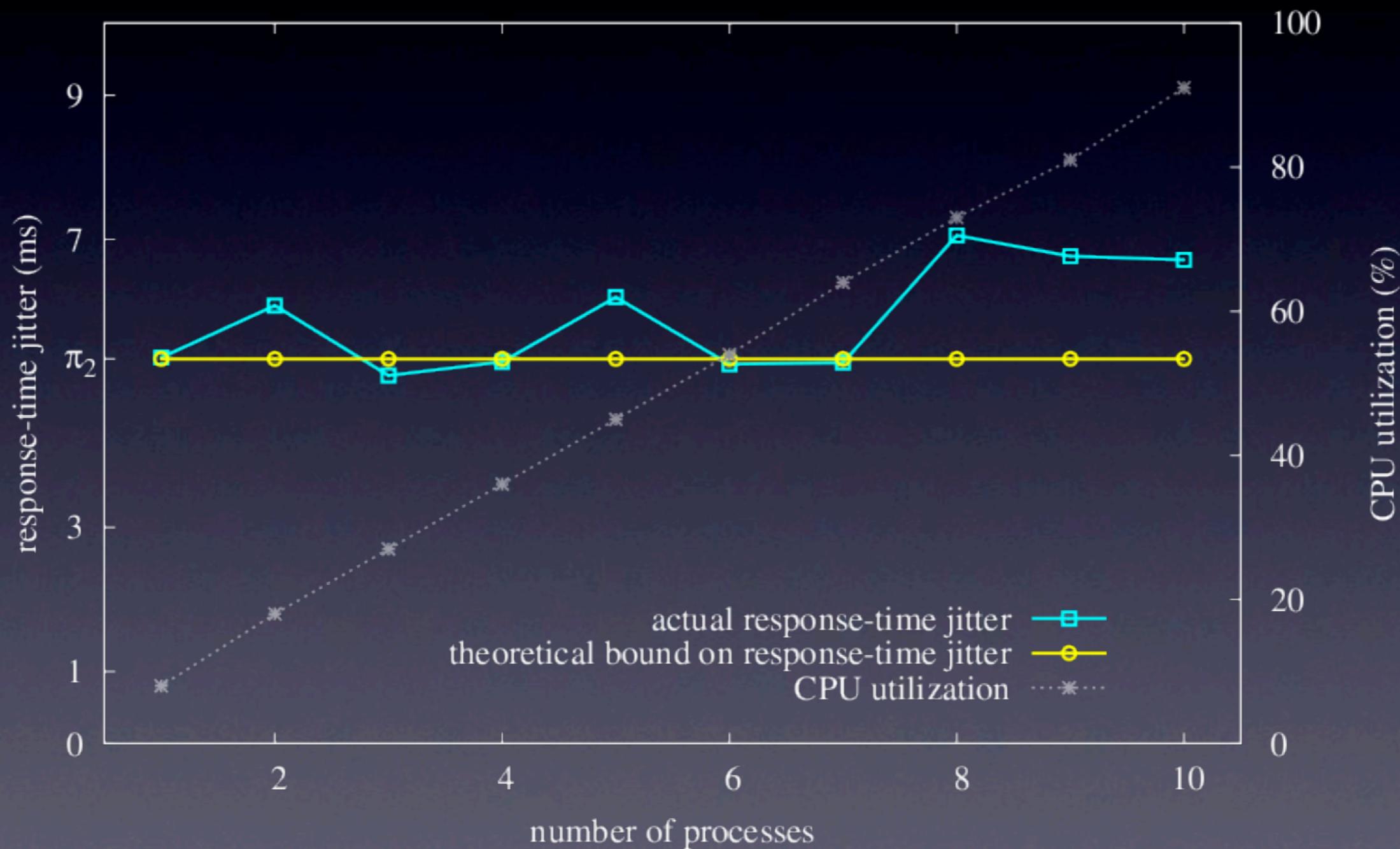


# Bare-metal experiment



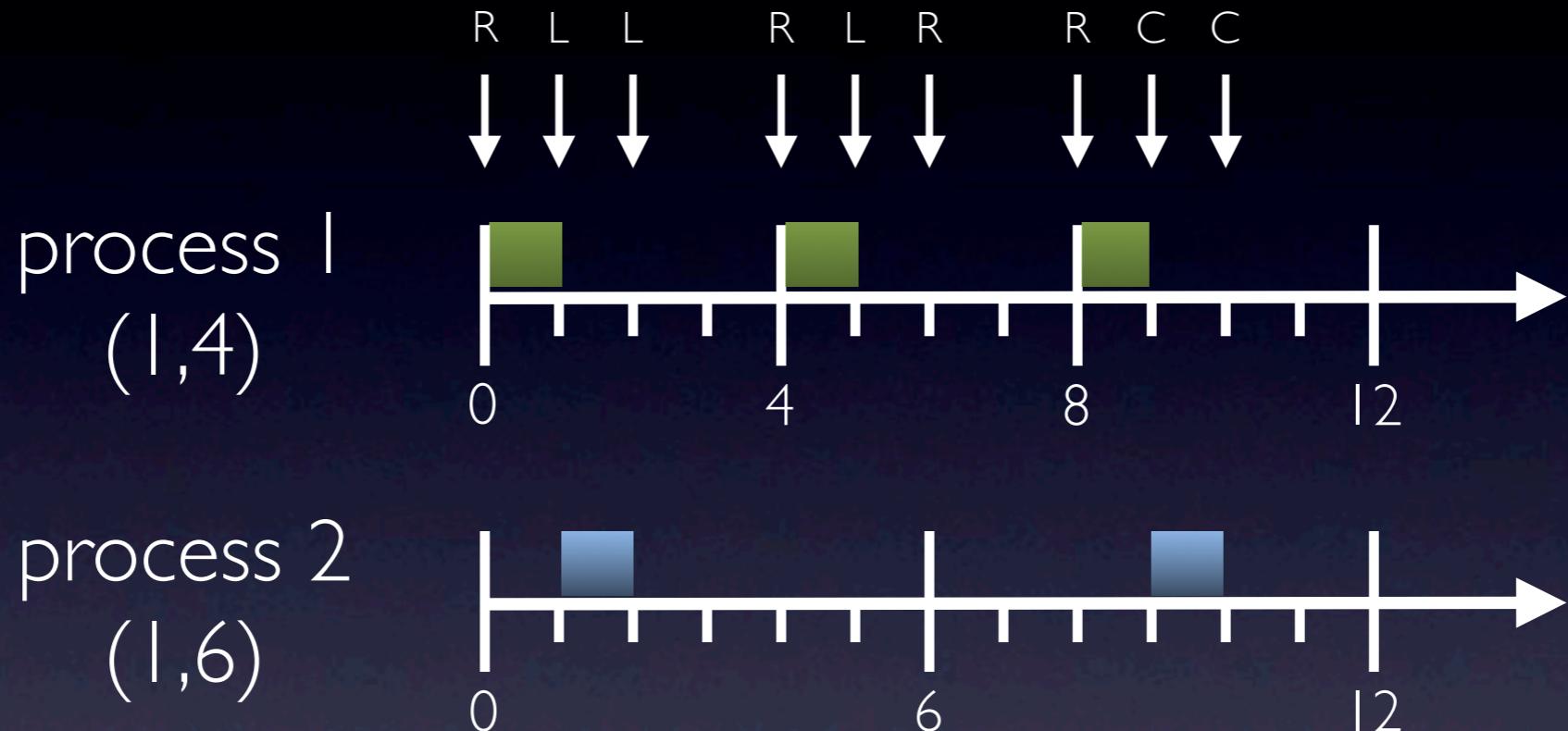


# Bare-metal experiment



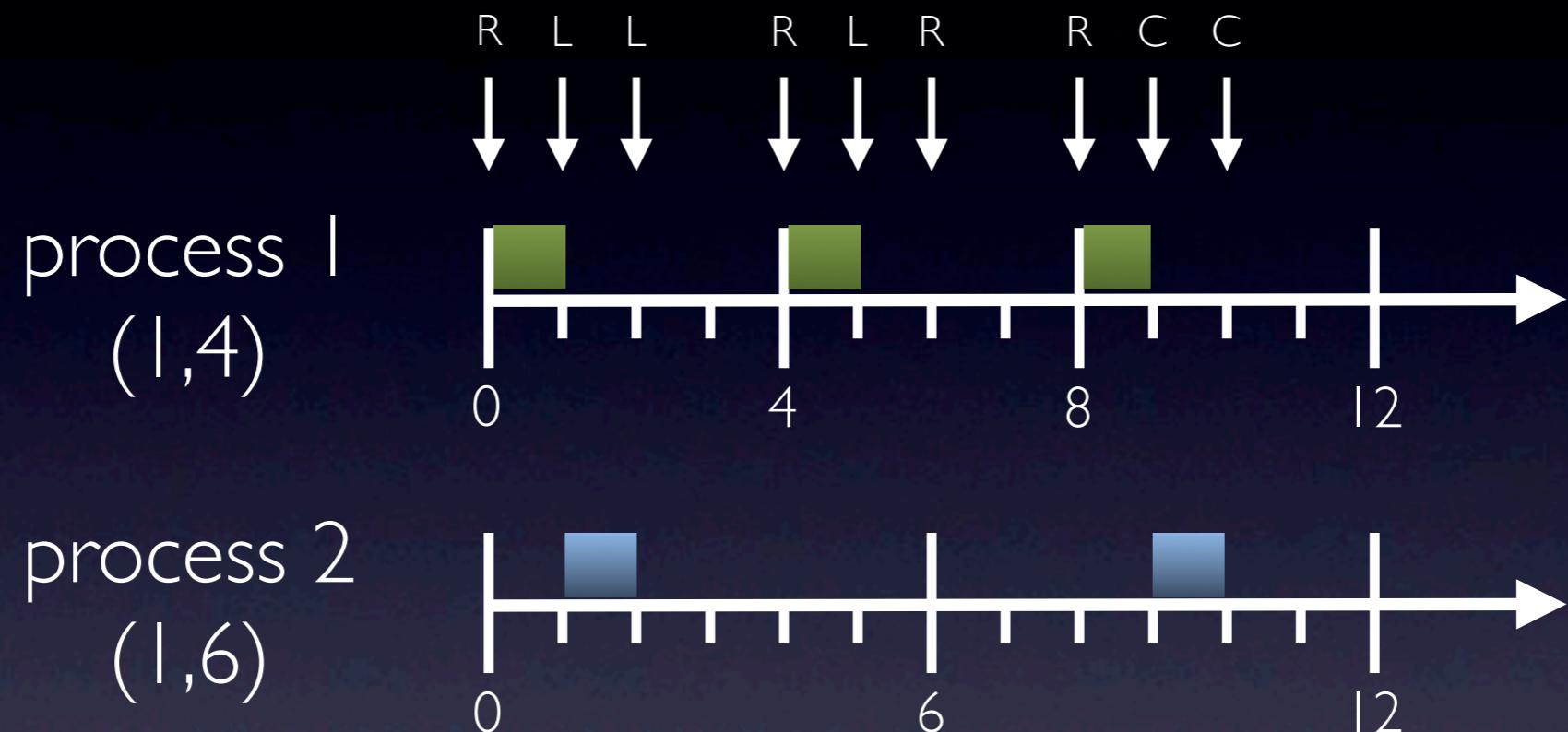


# Overhead accounting





# Overhead accounting

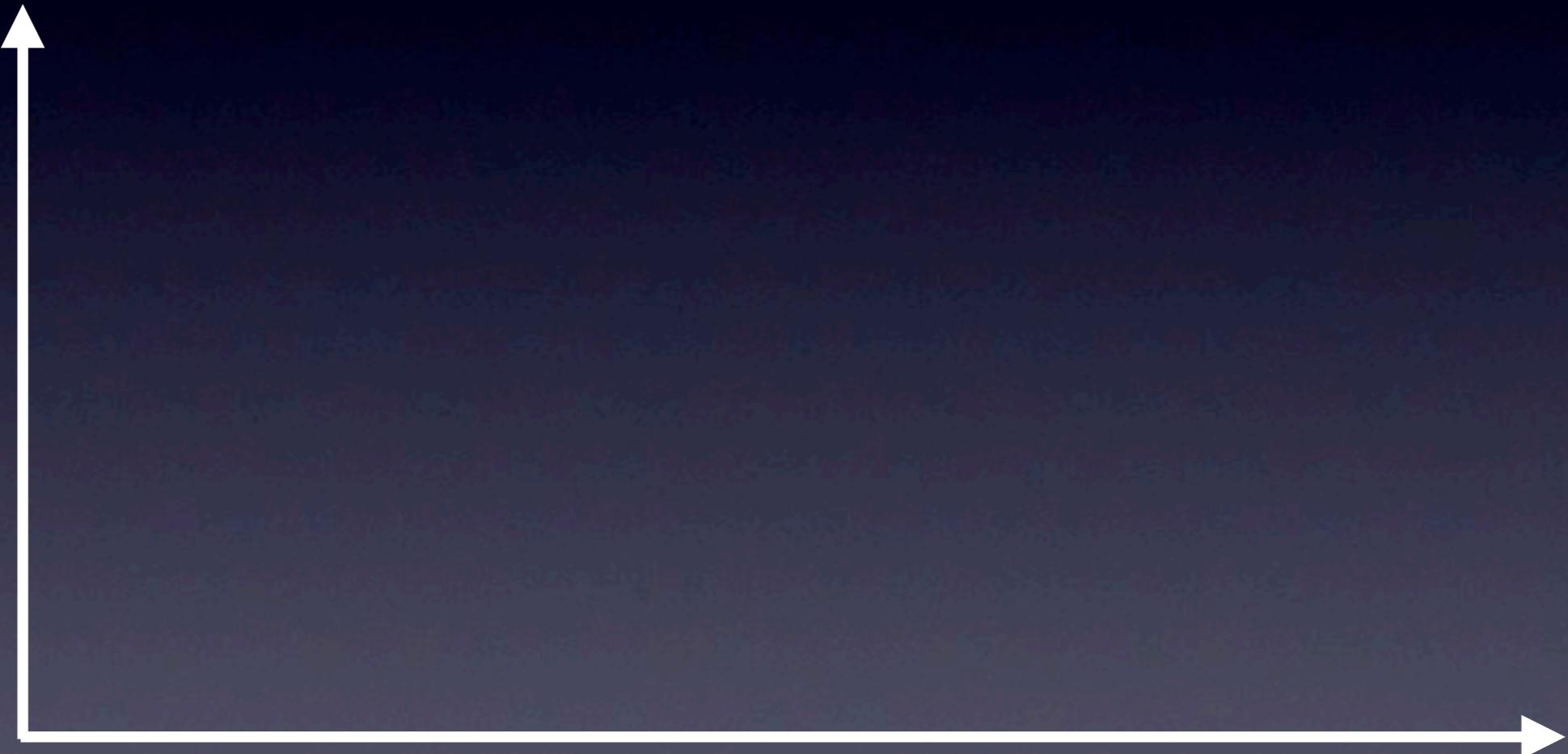


$$N = \left\lceil \frac{\pi}{\gcd(\text{all periods})} \right\rceil + 1$$



# Overhead accounting

utilization

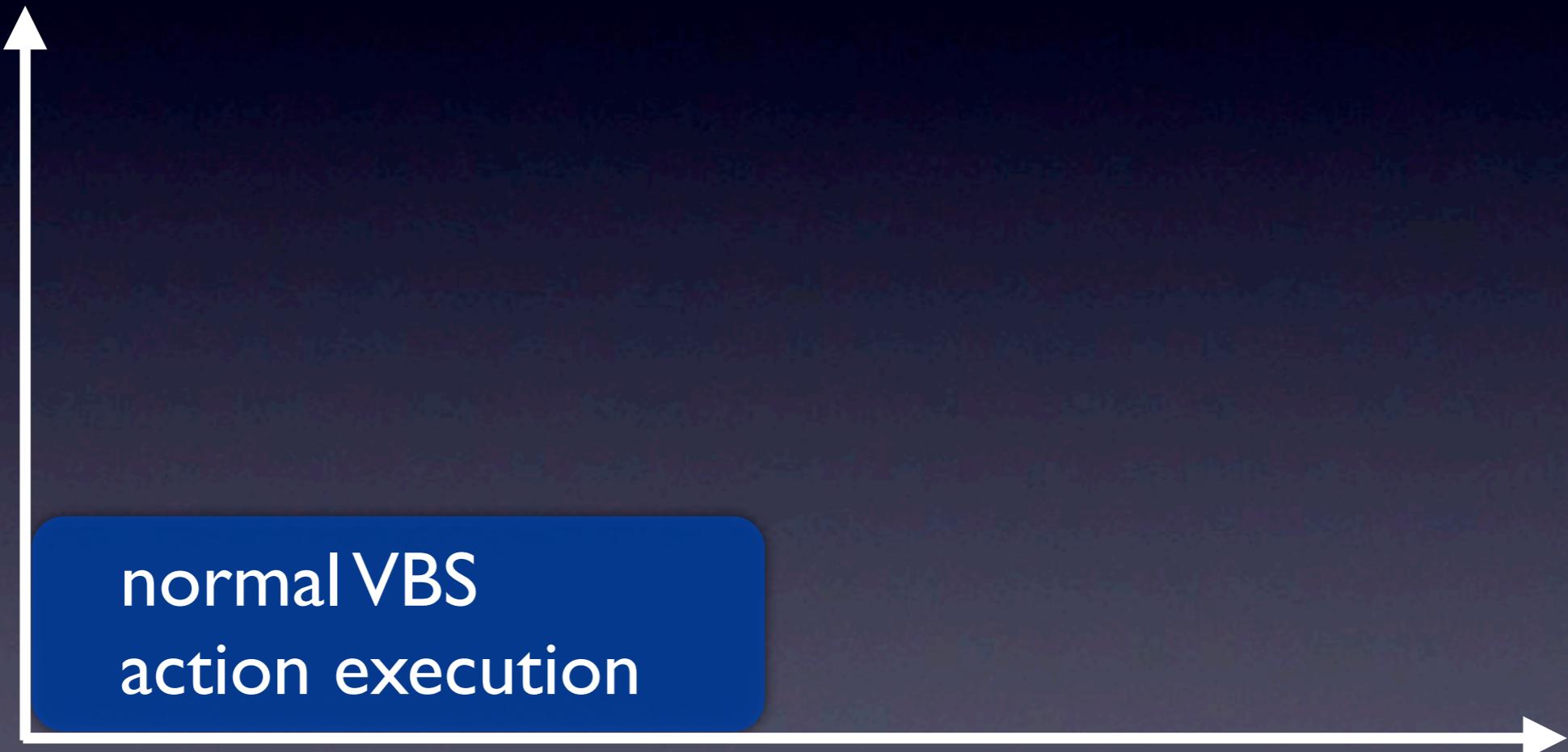


response time



# Overhead accounting

utilization

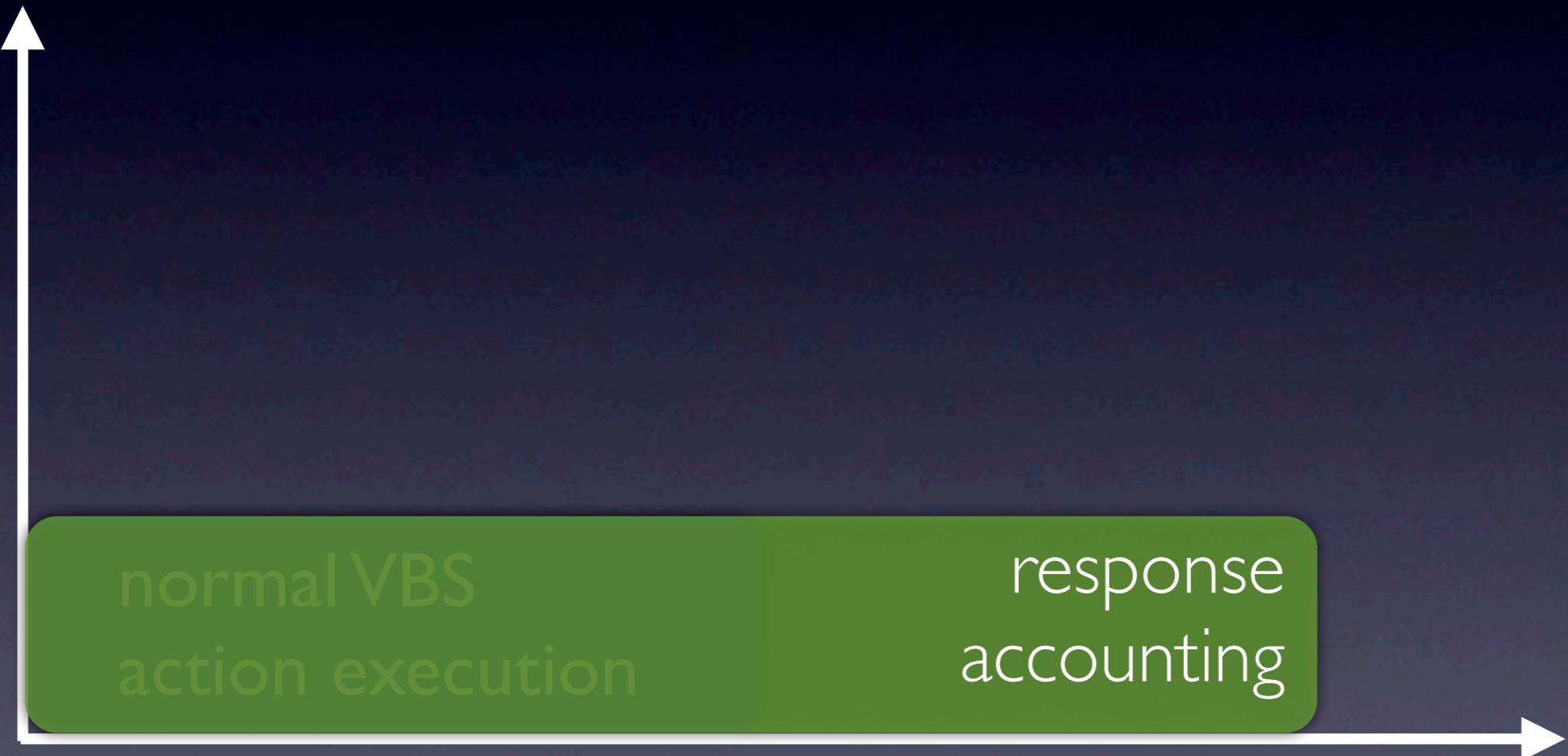


response time



# Overhead accounting

utilization

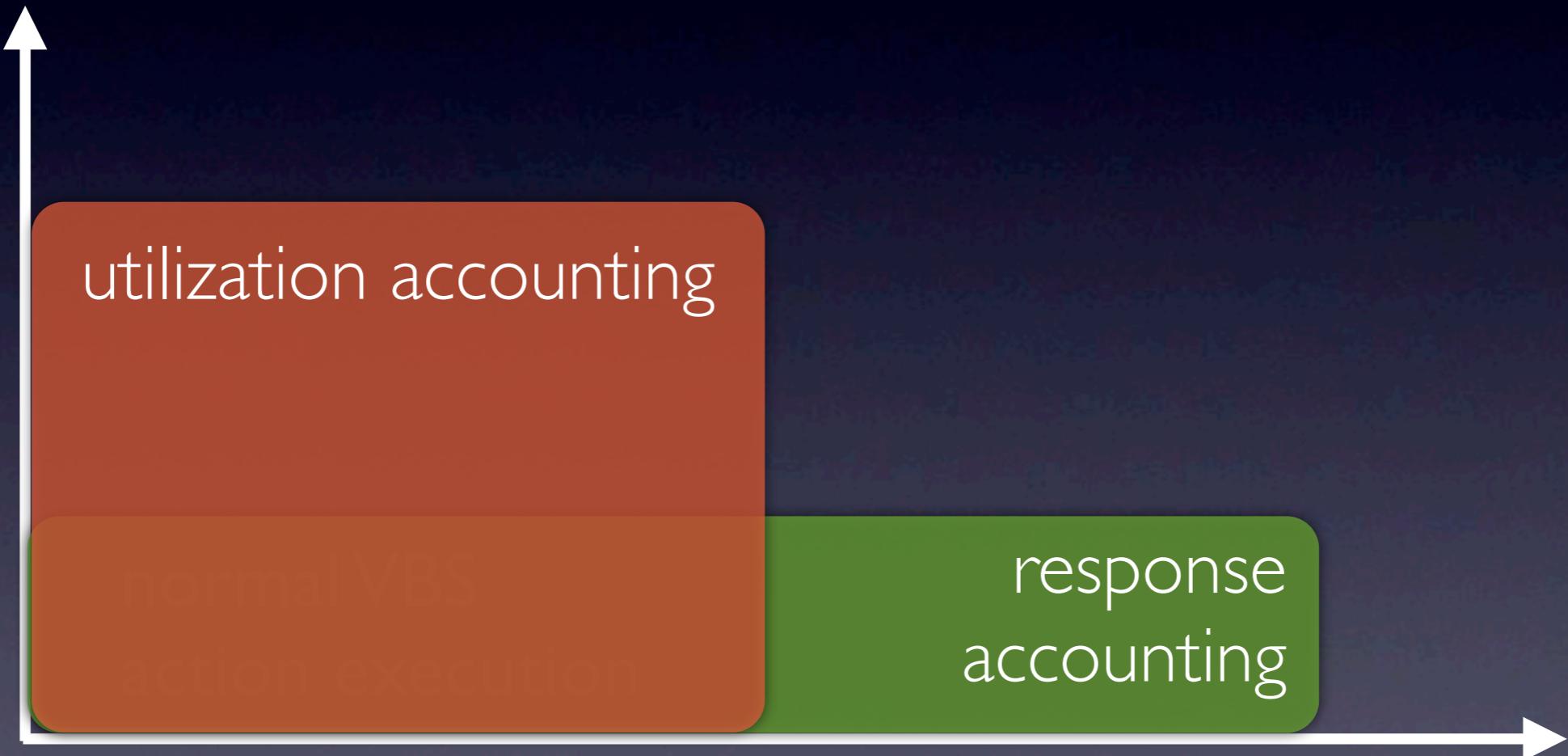


response time



# Overhead accounting

utilization



response time



# Overhead accounting

utilization

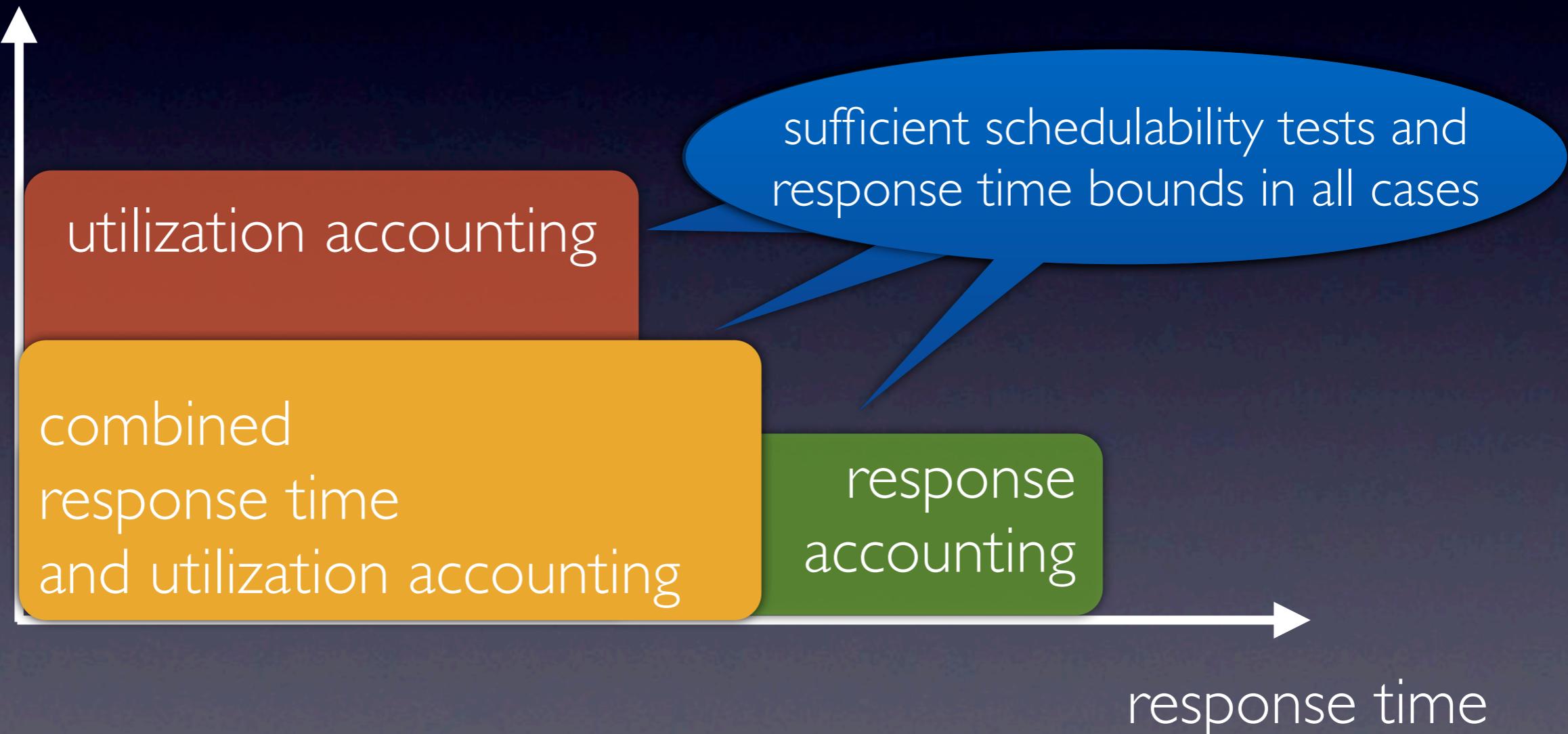


response time



# Overhead accounting

utilization



response time



# Without overhead

utilization



normal VBS  
action execution



response time



# Without overhead

utilization



normal VBS  
action execution

$$\text{test: } \sum_i u_i \leq 1$$

bounds:

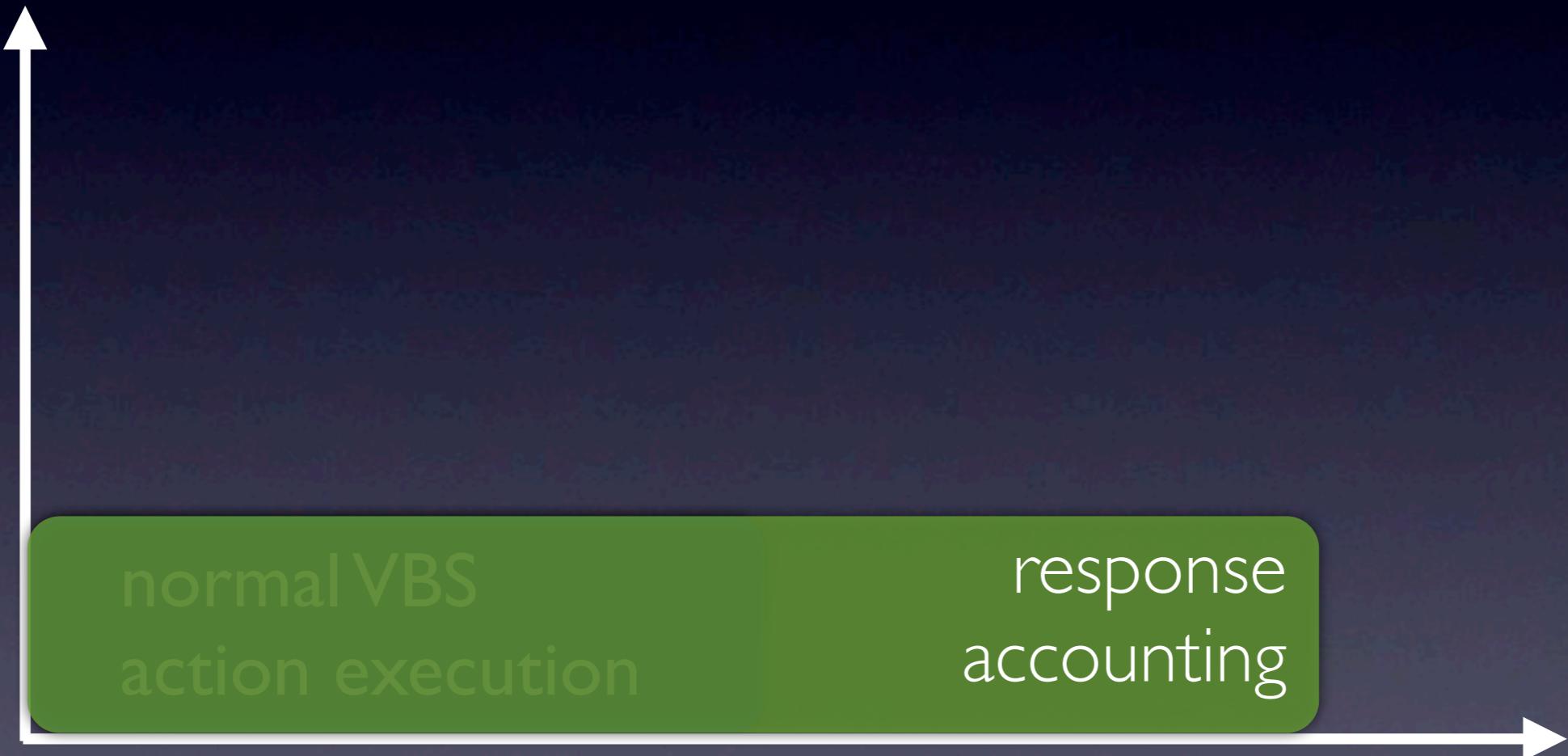
$$\left\lceil \frac{\text{load}}{\lambda} \right\rceil \pi \leq RT \leq \left\lceil \frac{\text{load}}{\lambda} \right\rceil \pi + \pi - 1$$

response time



# Response accounting

utilization



response time



# Response accounting

utilization



test:  $\sum_i u_i \leq 1$

bounds:

$$\text{load} + \left\lceil \frac{\text{load}}{\lambda - \delta} \right\rceil \delta$$

$$\left\lceil \frac{\text{load}^*}{\lambda} \right\rceil \pi \leq \text{RT} \leq \left\lceil \frac{\text{load}^*}{\lambda} \right\rceil \pi + \pi - 1$$

normal VBS  
action execution

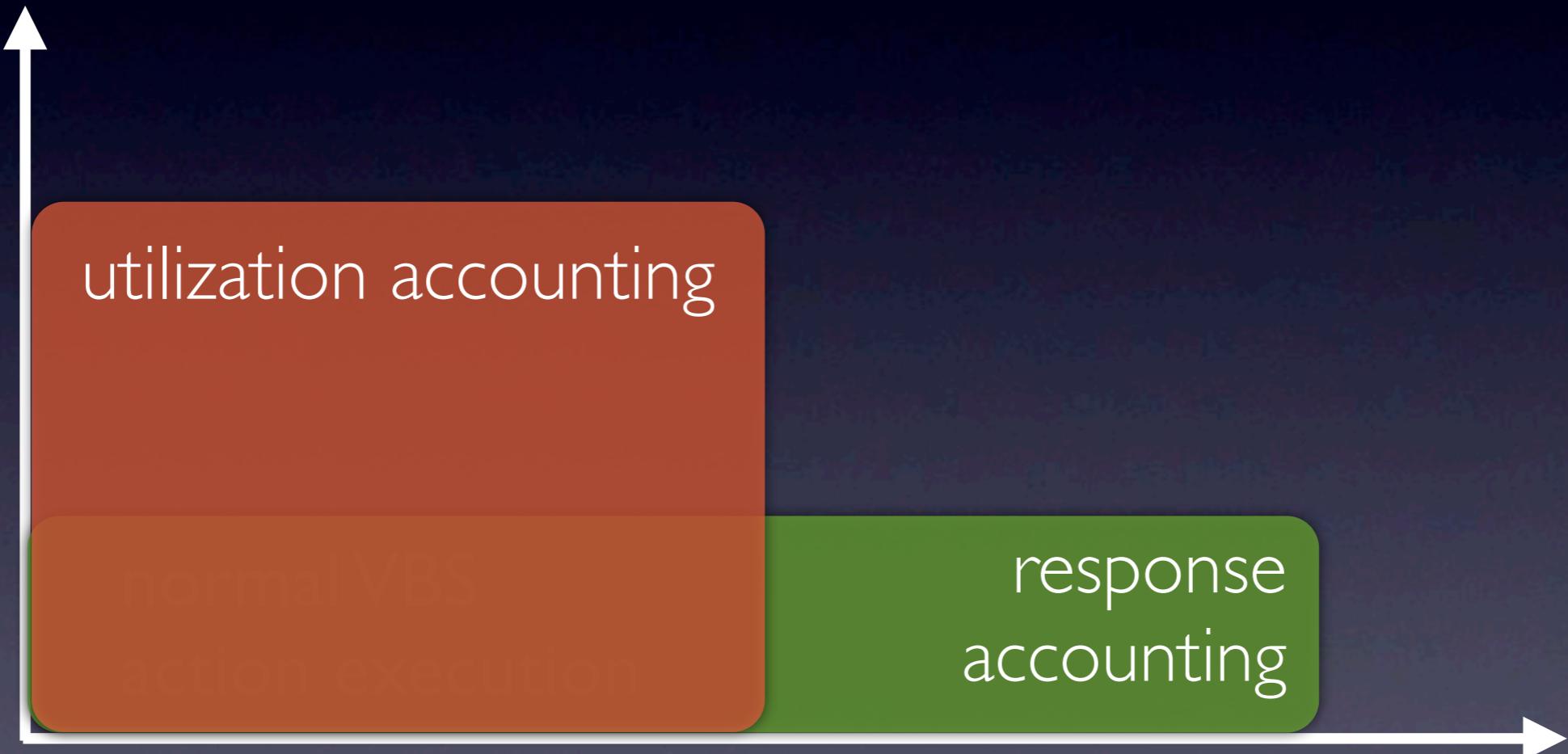
response  
accounting

response time



# Utilization accounting

utilization



response time



# Utilization accounting

utilization



utilization accounting

normal VBS  
action execution

test:  $\sum_i \max_j \frac{\lambda_{i,j} + \delta_{i,j}}{\pi_{i,j}} \leq 1$

bounds:

$$\left\lceil \frac{\text{load}}{\lambda} \right\rceil \pi \leq RT \leq \left\lceil \frac{\text{load}}{\lambda} \right\rceil \pi + \pi - 1$$

response  
accounting



response time



# Utilization accounting

utilization



response time



# Utilization accounting

utilization



utilization accounting

combined  
response time  
and utilization accounting

test:  $\sum_i \max_j \frac{\lambda_{i,j} + \delta_{i,j}^u}{\pi_{i,j}} \leq 1$

bounds:

$$\left\lceil \frac{\text{load}^*}{\lambda} \right\rceil \pi \leq \text{RT} \leq \left\lceil \frac{\text{load}^*}{\lambda} \right\rceil \pi + \pi - 1$$

$$\text{load}' + \left\lceil \frac{\text{load}'}{\lambda} \right\rceil \delta^u$$

$$\text{load} + \left\lceil \frac{\text{load}}{\lambda - \delta^b} \right\rceil \delta^b$$

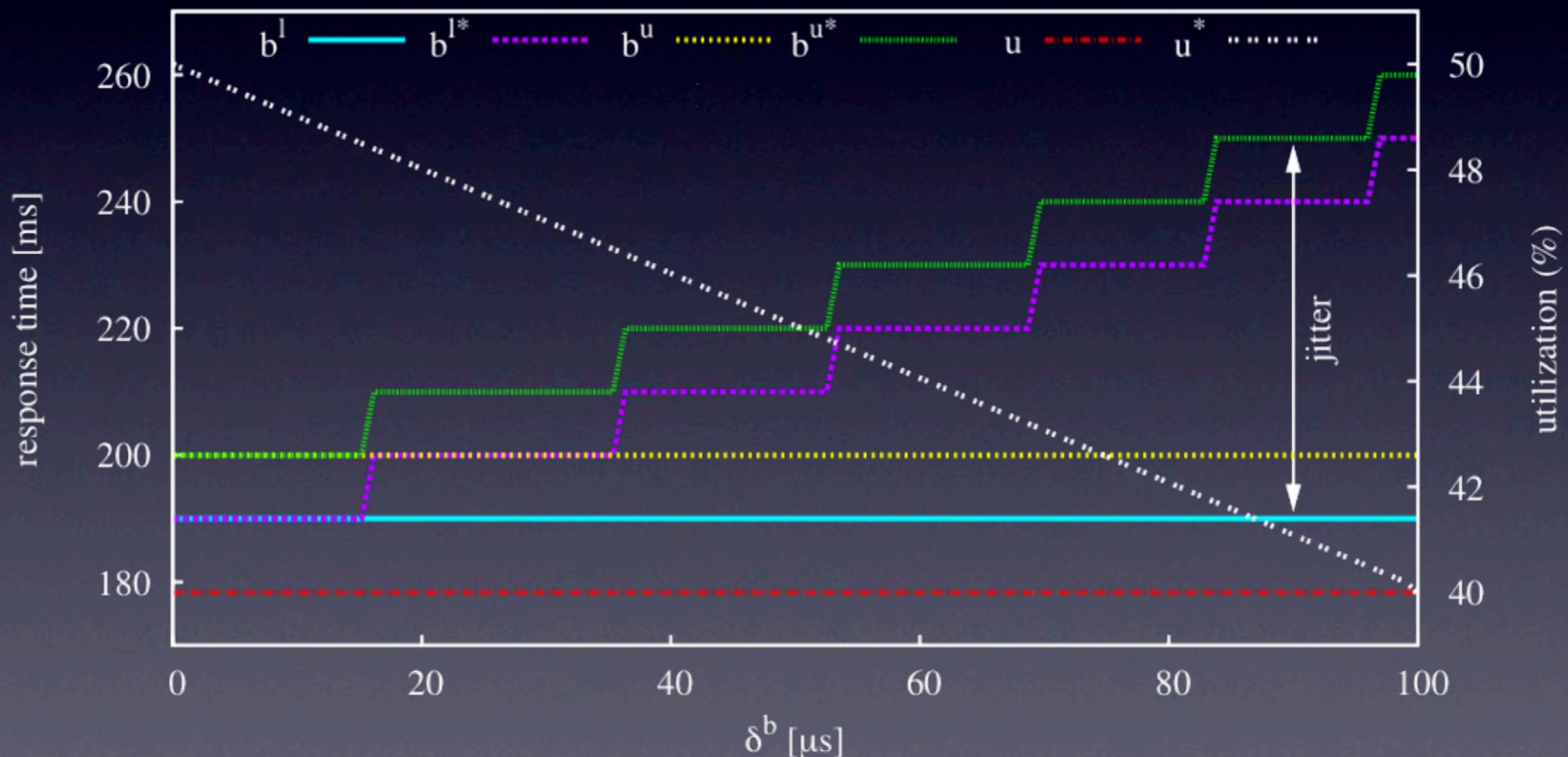
$$\delta = \delta^u + \delta^b$$

response  
accounting

response time

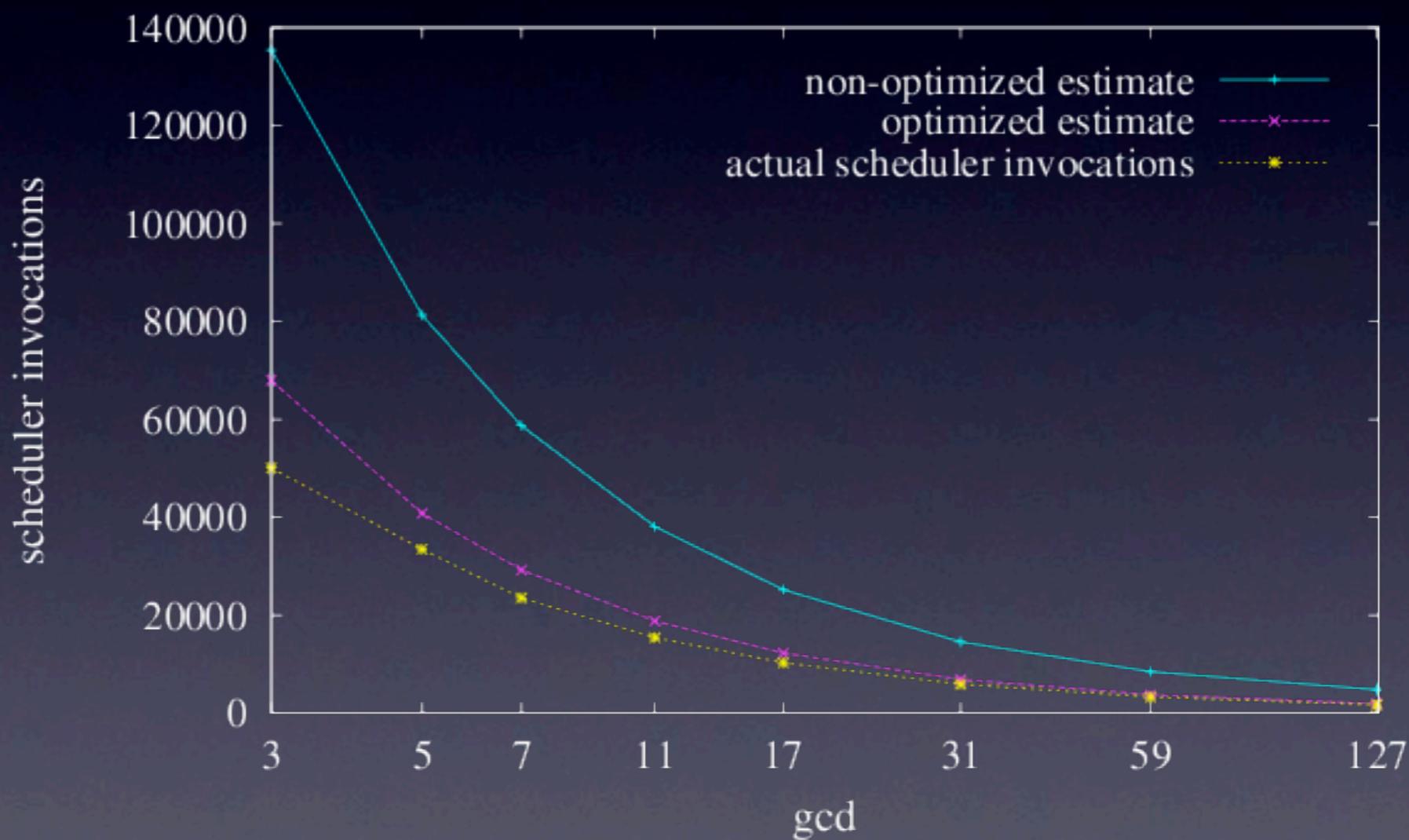


# Overhead accounting





# Experiments





# Power-Aware VBS

## Dynamic Voltage and Frequency Scaling



# Power-Aware VBS

Dynamic Voltage and Frequency Scaling

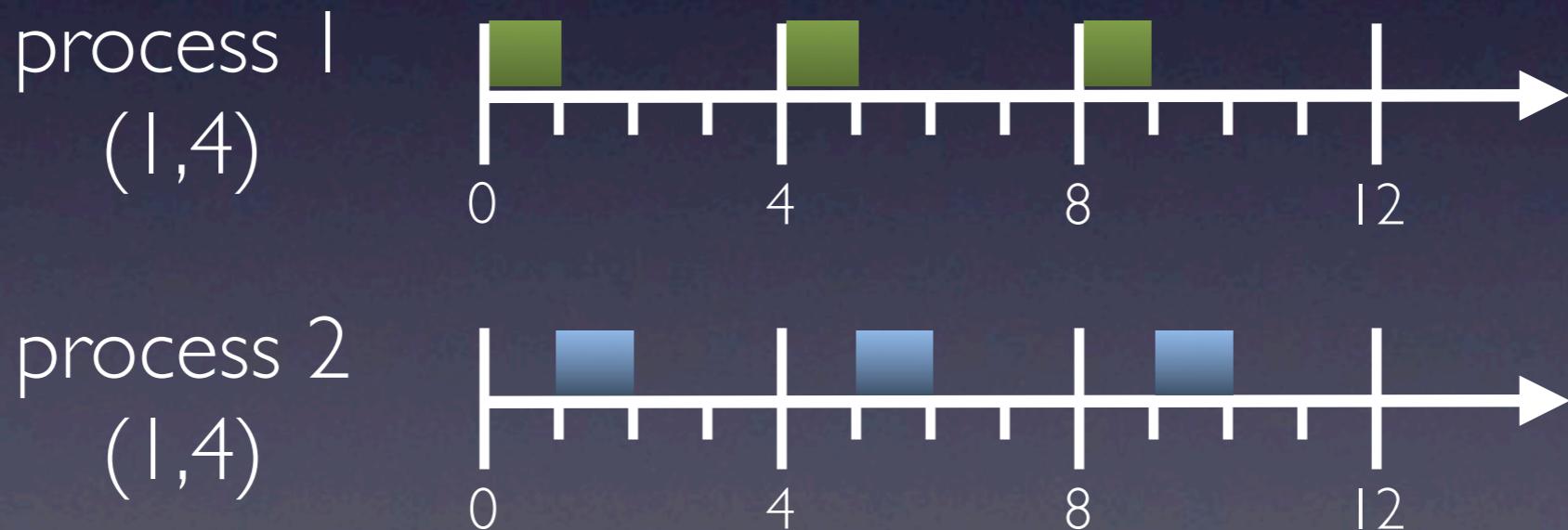
Maintain VBS properties (temporal isolation, bounds)



# Power-Aware VBS

Dynamic Voltage and Frequency Scaling

Maintain VBS properties (temporal isolation, bounds)

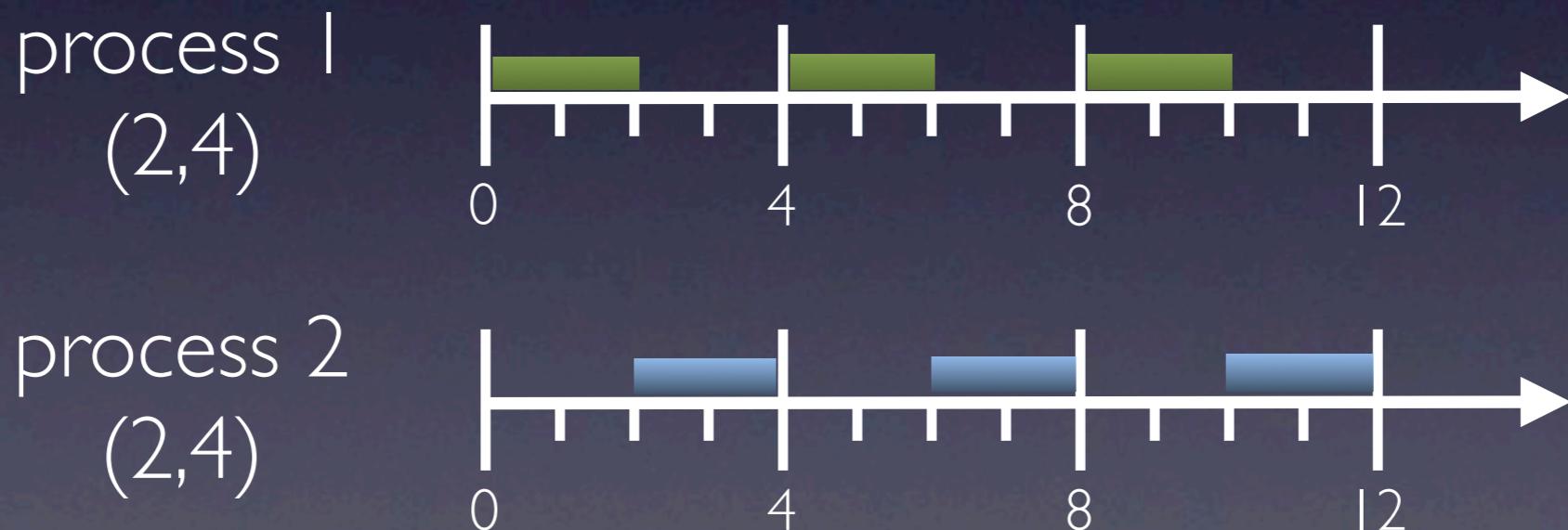




# Power-Aware VBS

Dynamic Voltage and Frequency Scaling

Maintain VBS properties (temporal isolation, bounds)

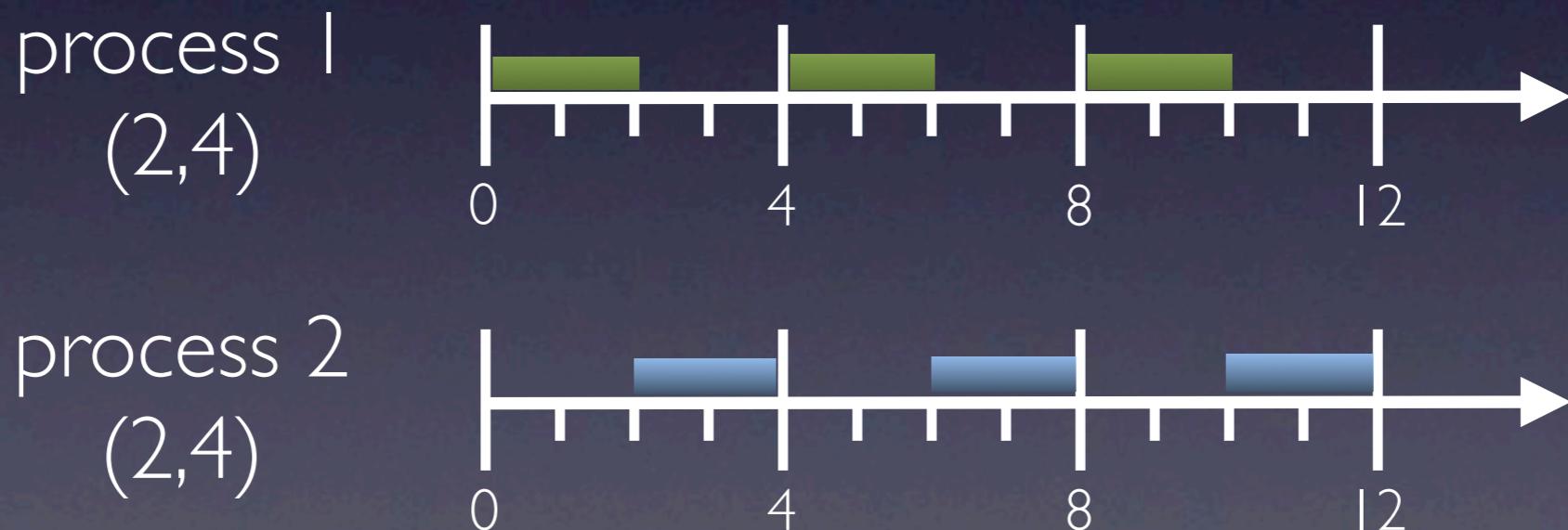




# Power-Aware VBS

Dynamic Voltage and Frequency Scaling

Maintain VBS properties (temporal isolation, bounds)



Possible whenever there is slack in the system



# Power-Aware VBS

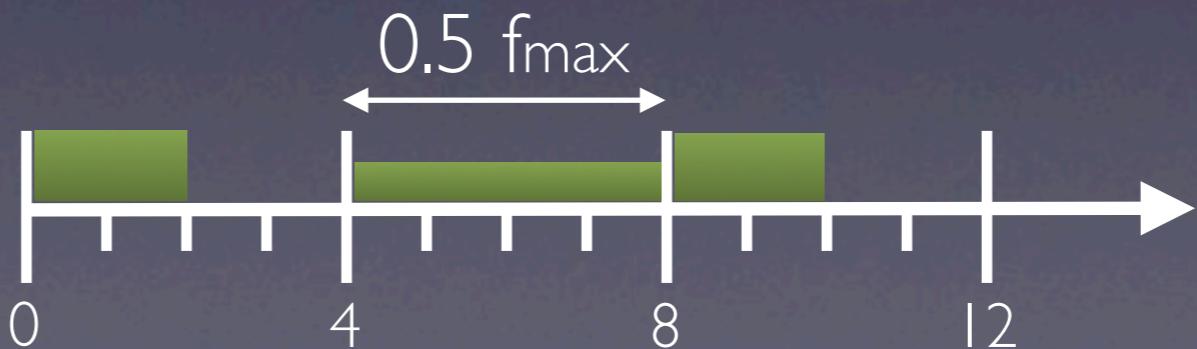
EDF frequency scaling result:

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 I  
(2,4)



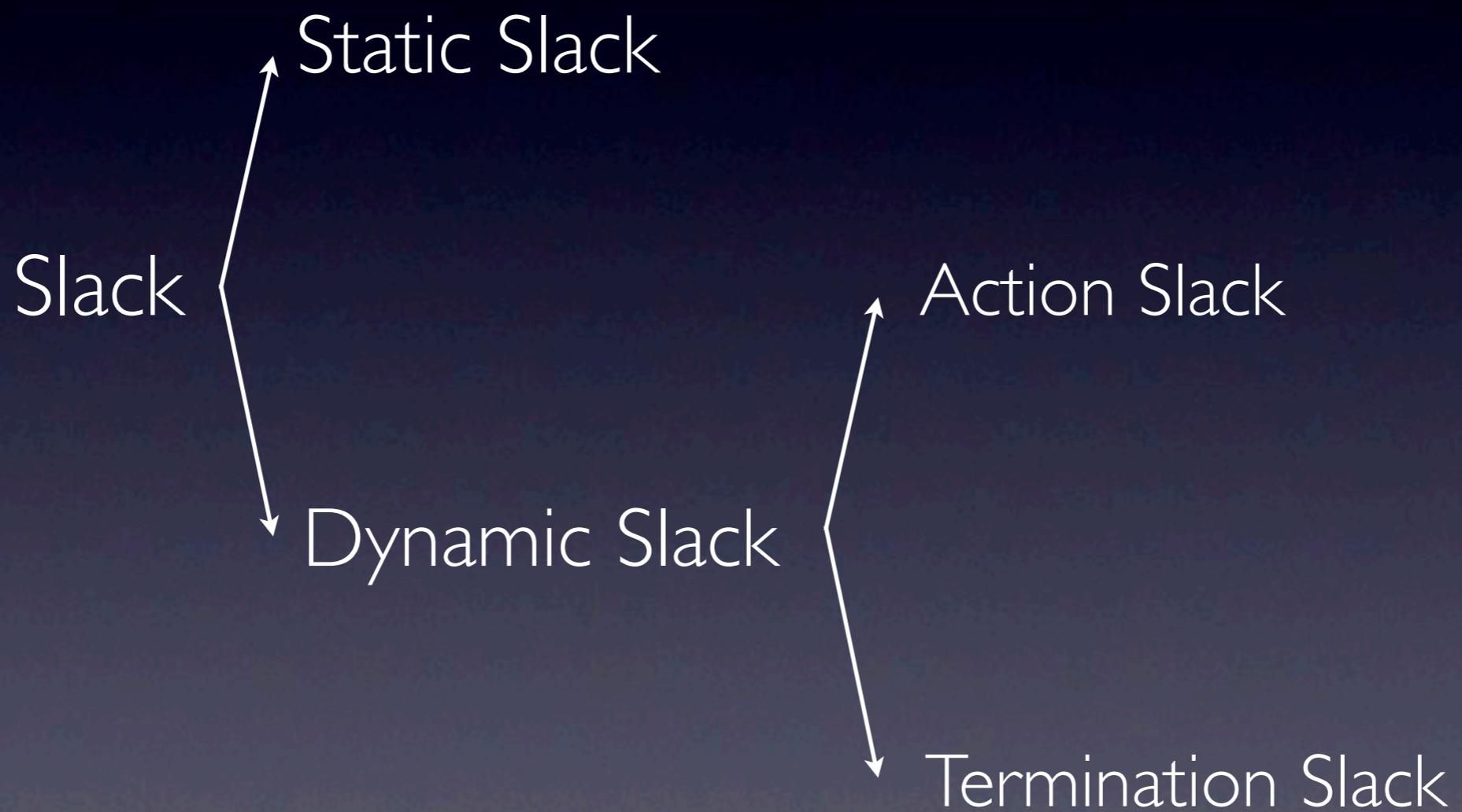


# Frequency-scaling VBS

Slack



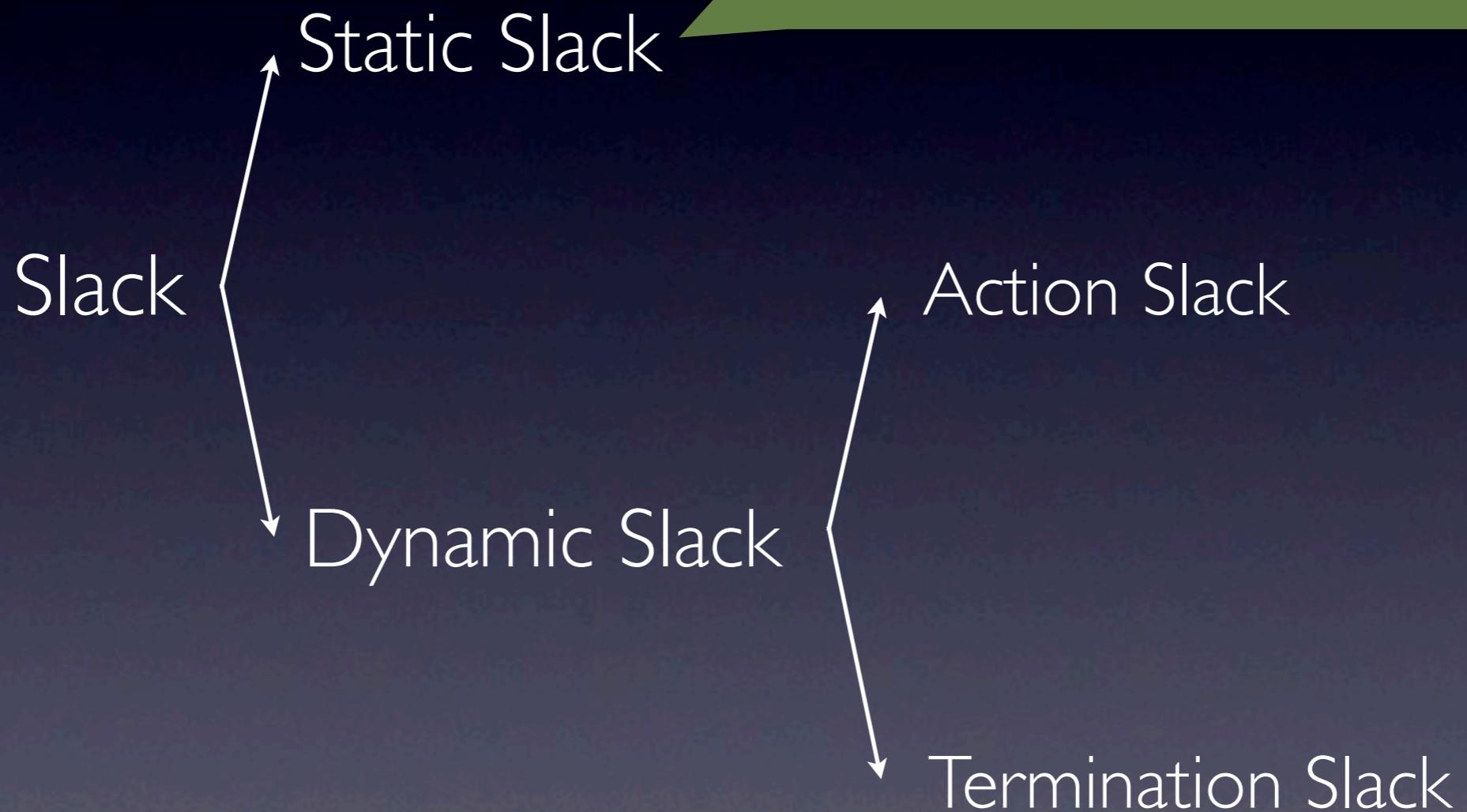
# Frequency-scaling VBS





# Frequency-scaling VBS

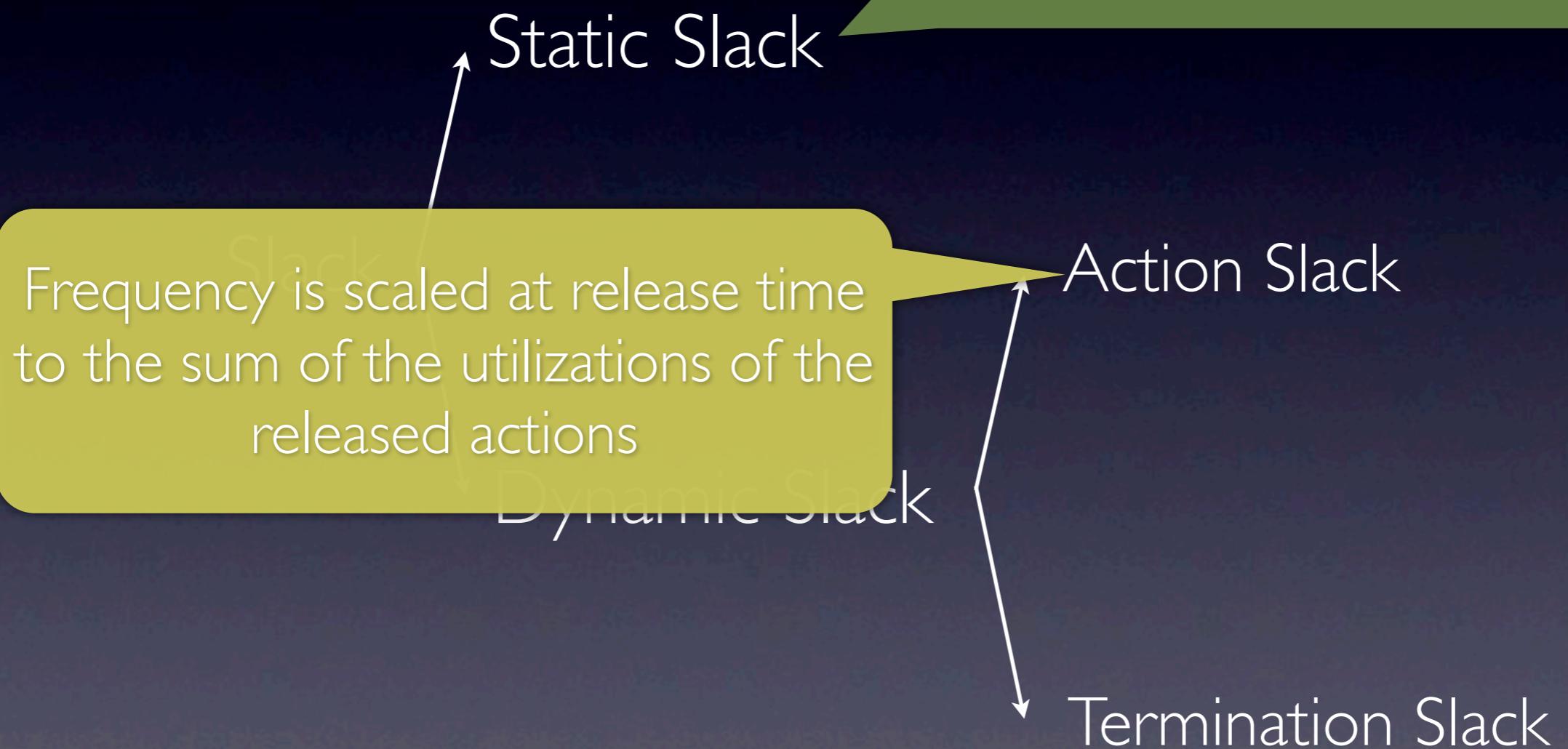
Frequency is scaled to the sum of the bandwidth caps and not changed at runtime





# Frequency-scaling VBS

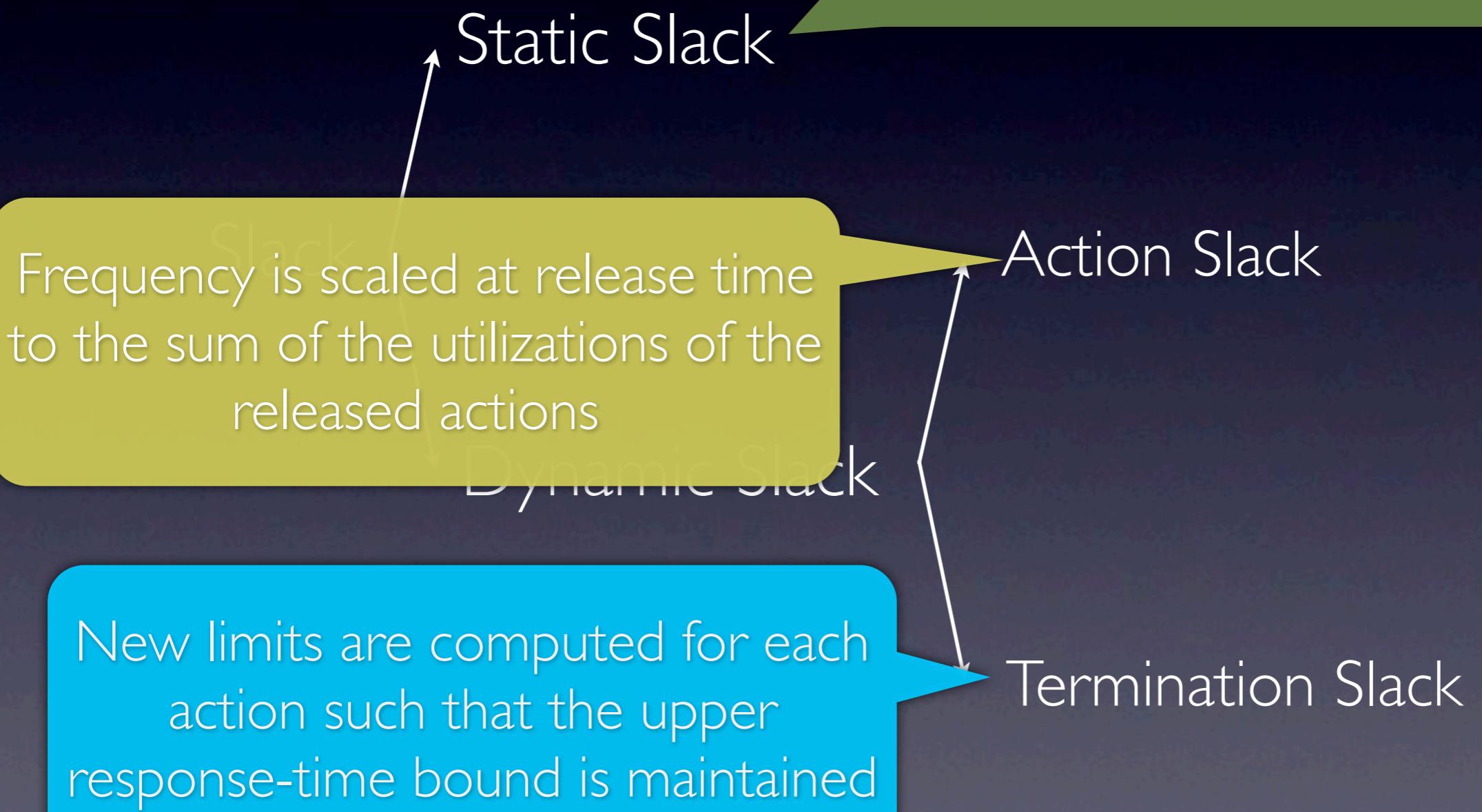
Frequency is scaled to the sum of the bandwidth caps and not changed at runtime





# Frequency-scaling VBS

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





# 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

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



# 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

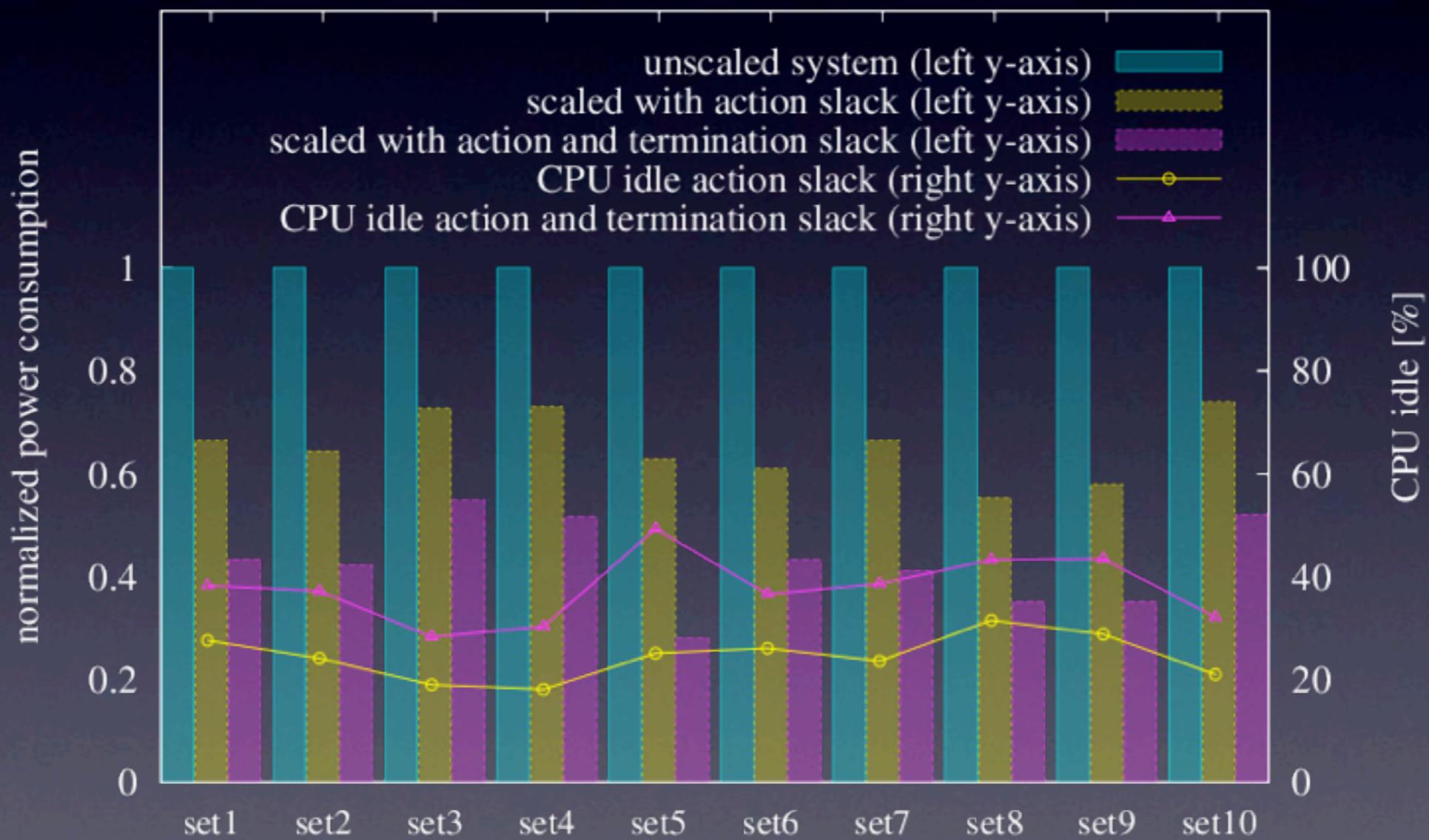
$$f = \sum_{i=1}^n \frac{\lambda_{i,j}^*}{\pi_{i,j}} \cdot f_{max} \quad \lambda_{i,j}^* = \left\lceil \frac{l_{i,j}}{n_{i,j}} \right\rceil \quad n_{i,j} = \left\lceil \frac{l_{i,j}}{\lambda_{i,j}} \right\rceil$$

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:

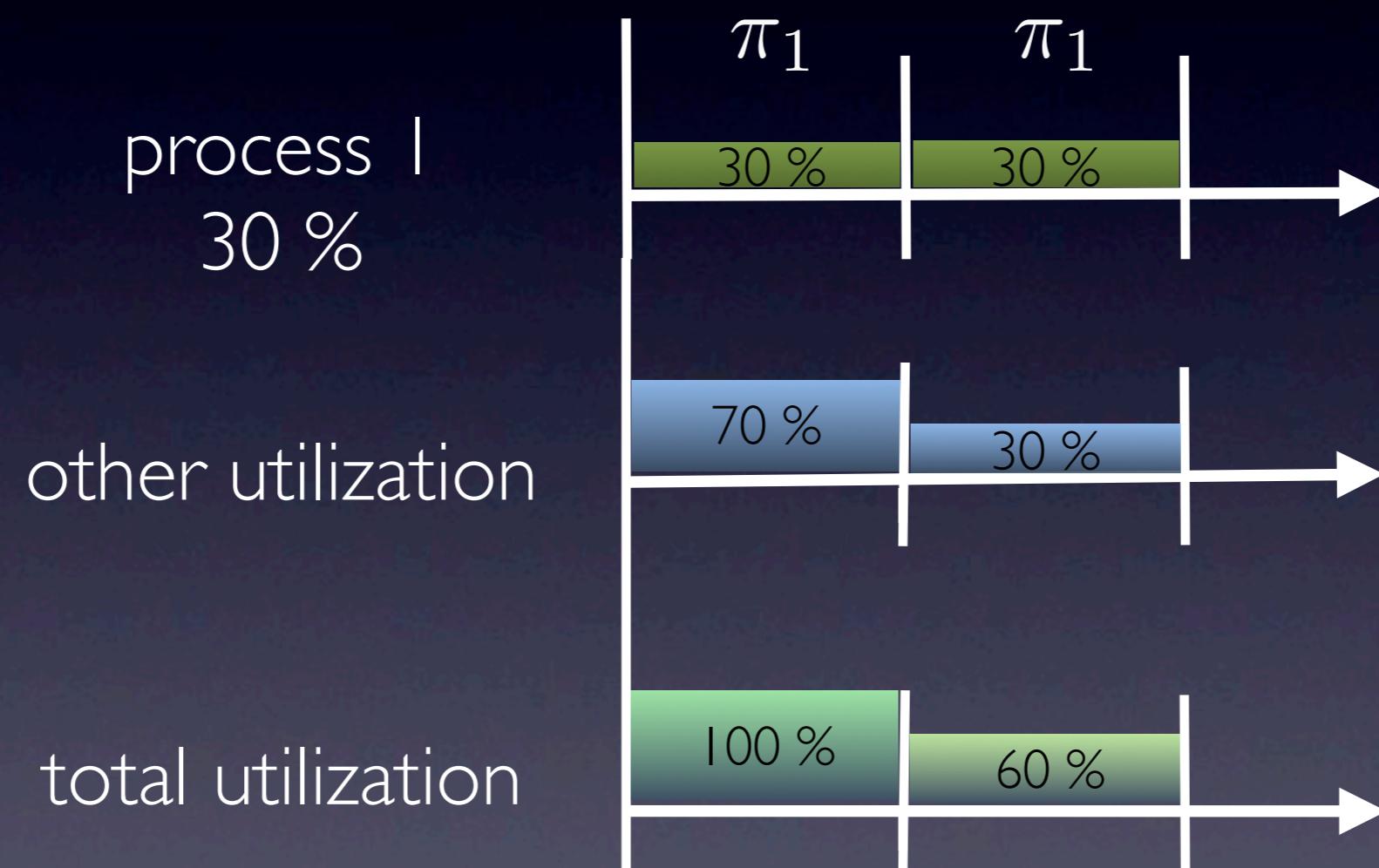
- more complex power models

- frequency switching cost (time and power)

- time overhead included using overhead accounting

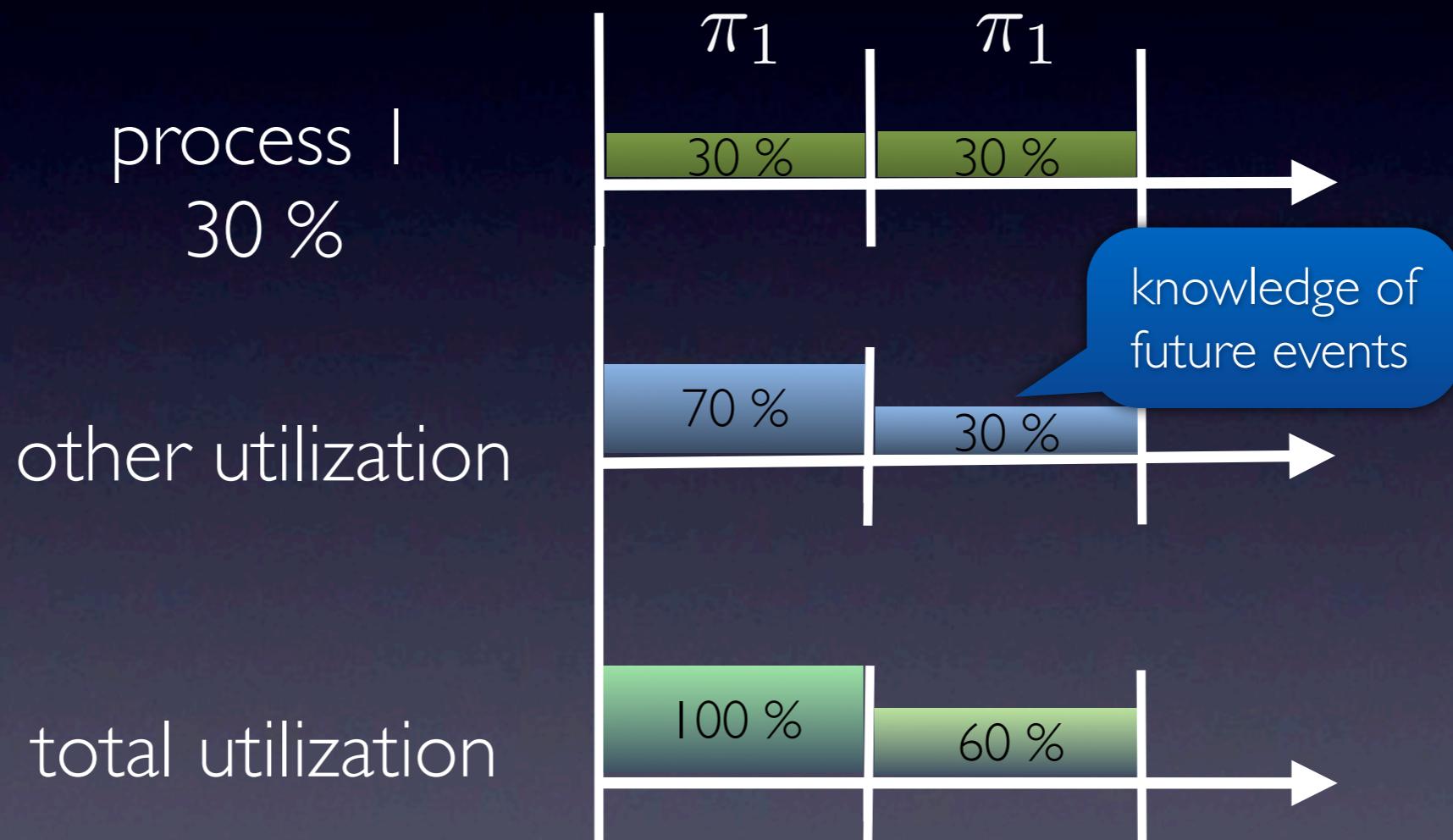


# Look-ahead FS-VBS



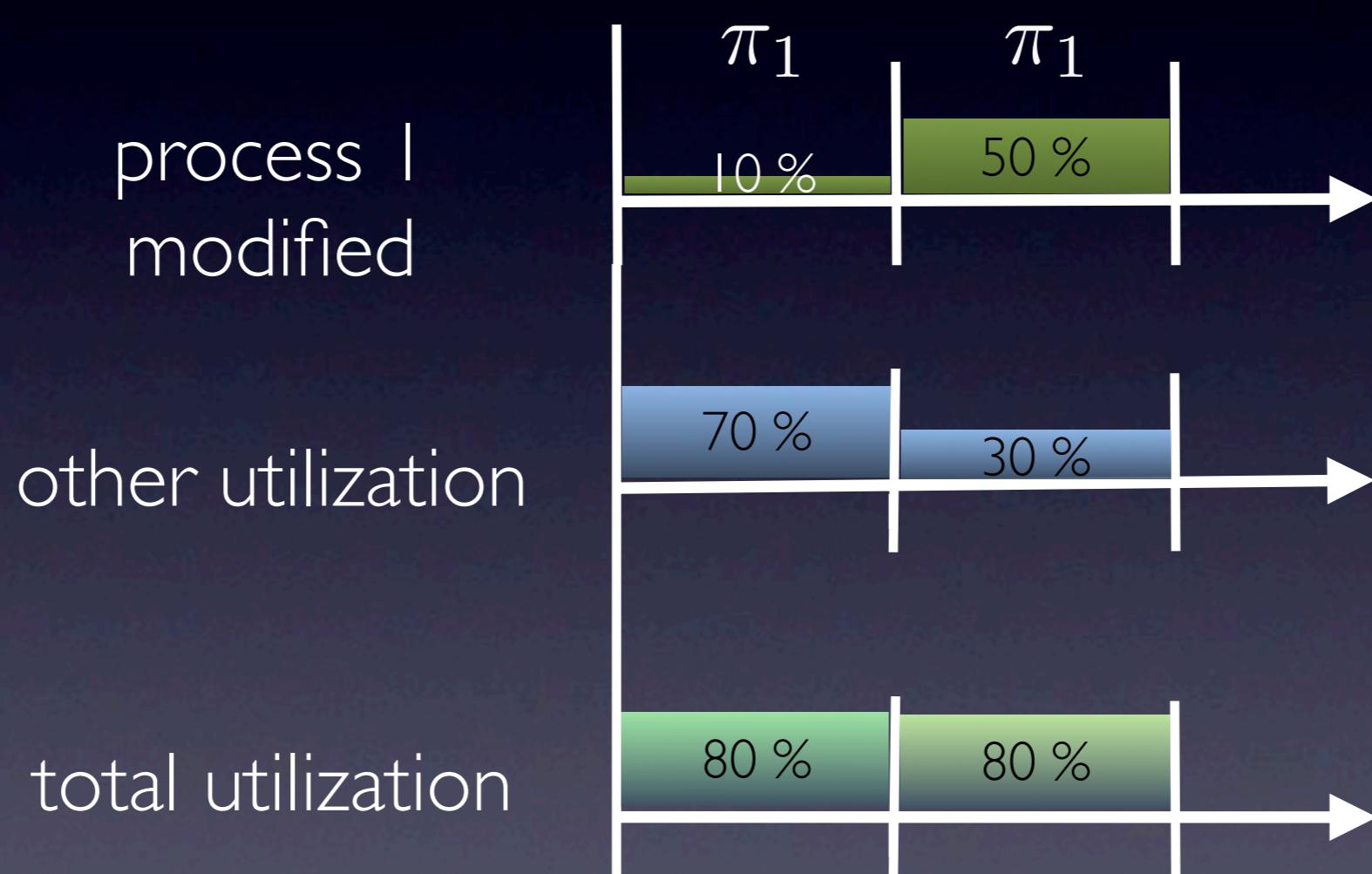


# Look-ahead FS-VBS



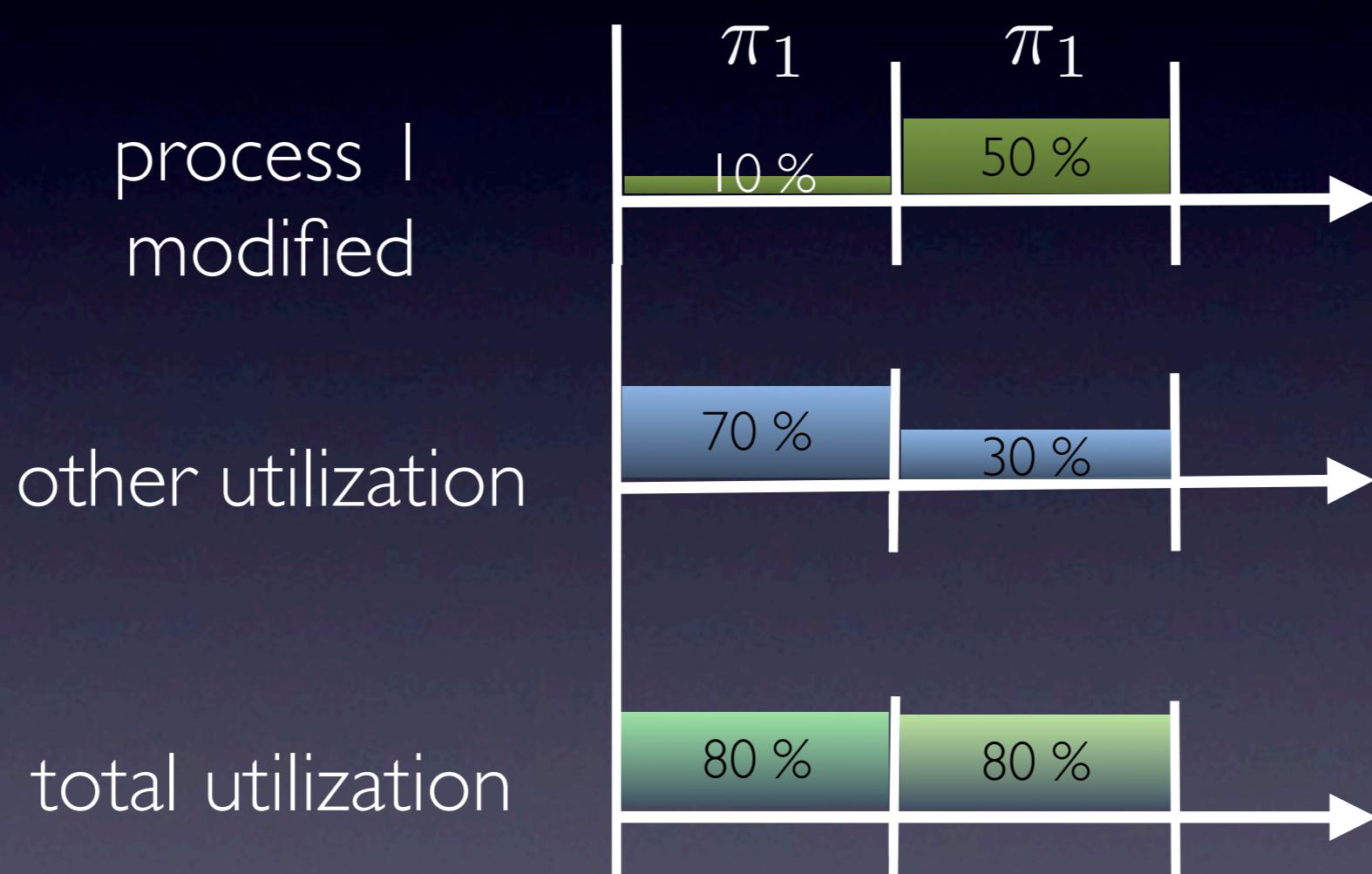


# Look-ahead FS-VBS





# Look-ahead FS-VBS

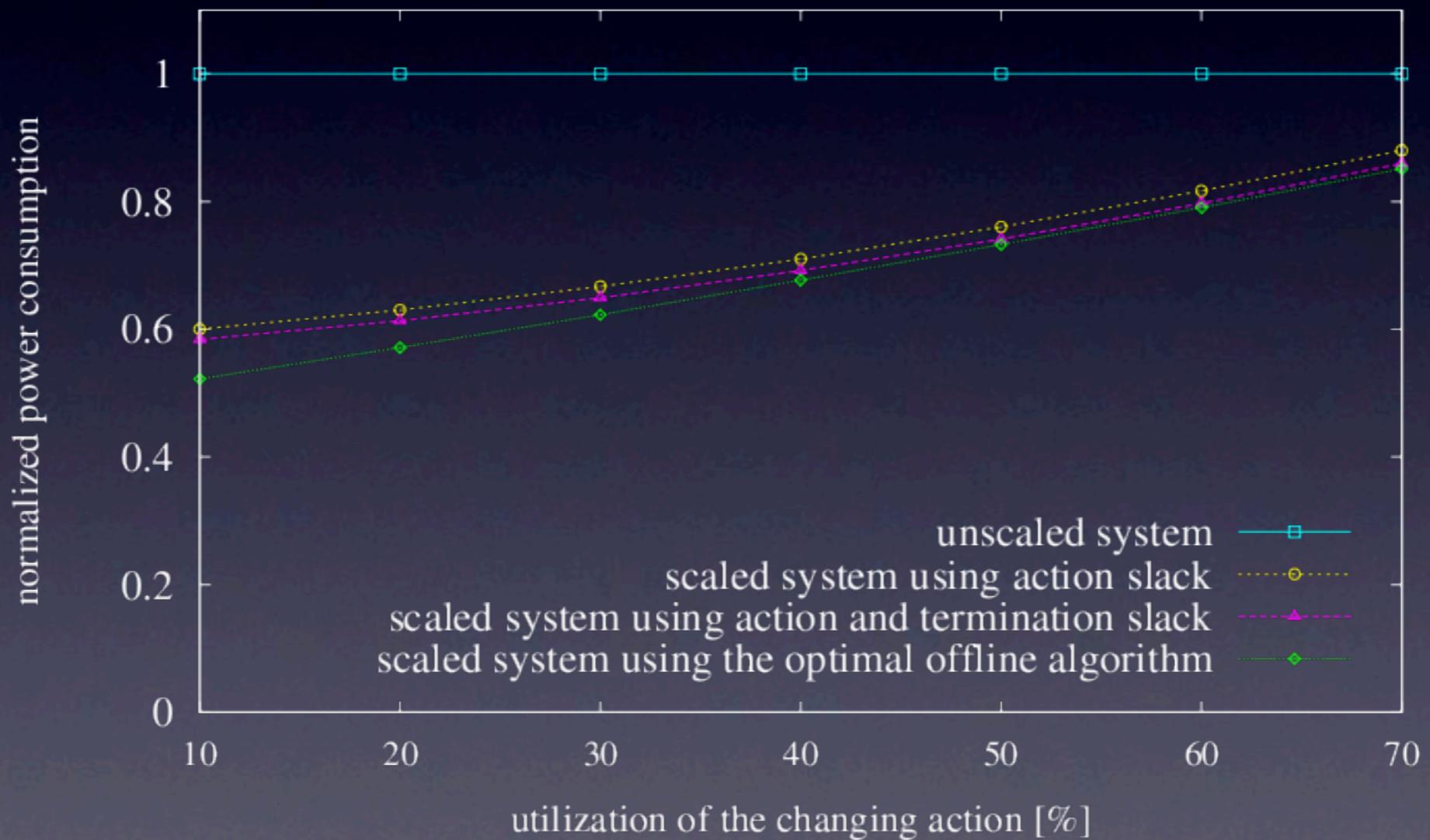


actual improvement depends  
on the power model



# Look-ahead FS-VBS

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



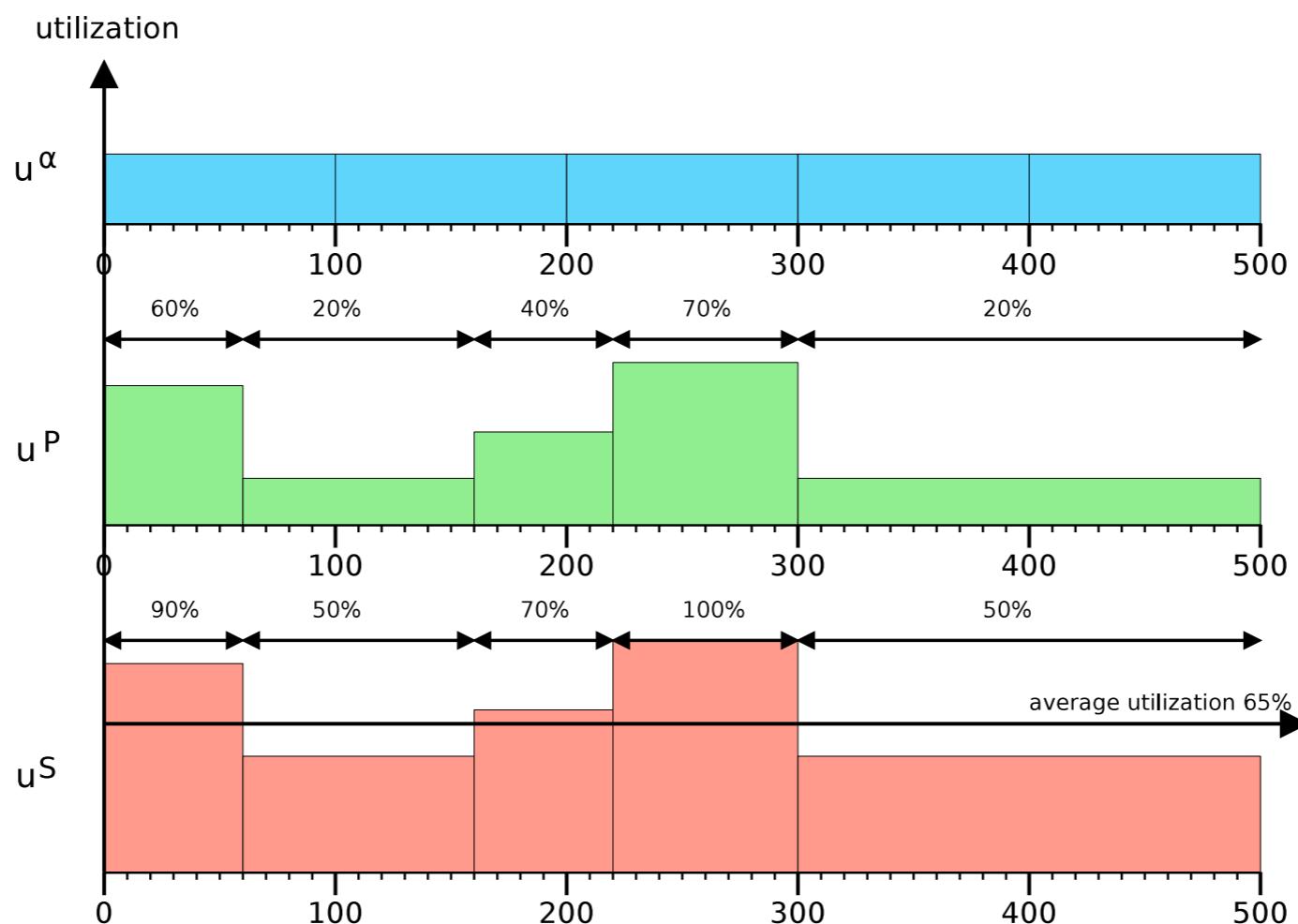


# Look-ahead online FS-VBS



# Look-ahead online FS-VBS

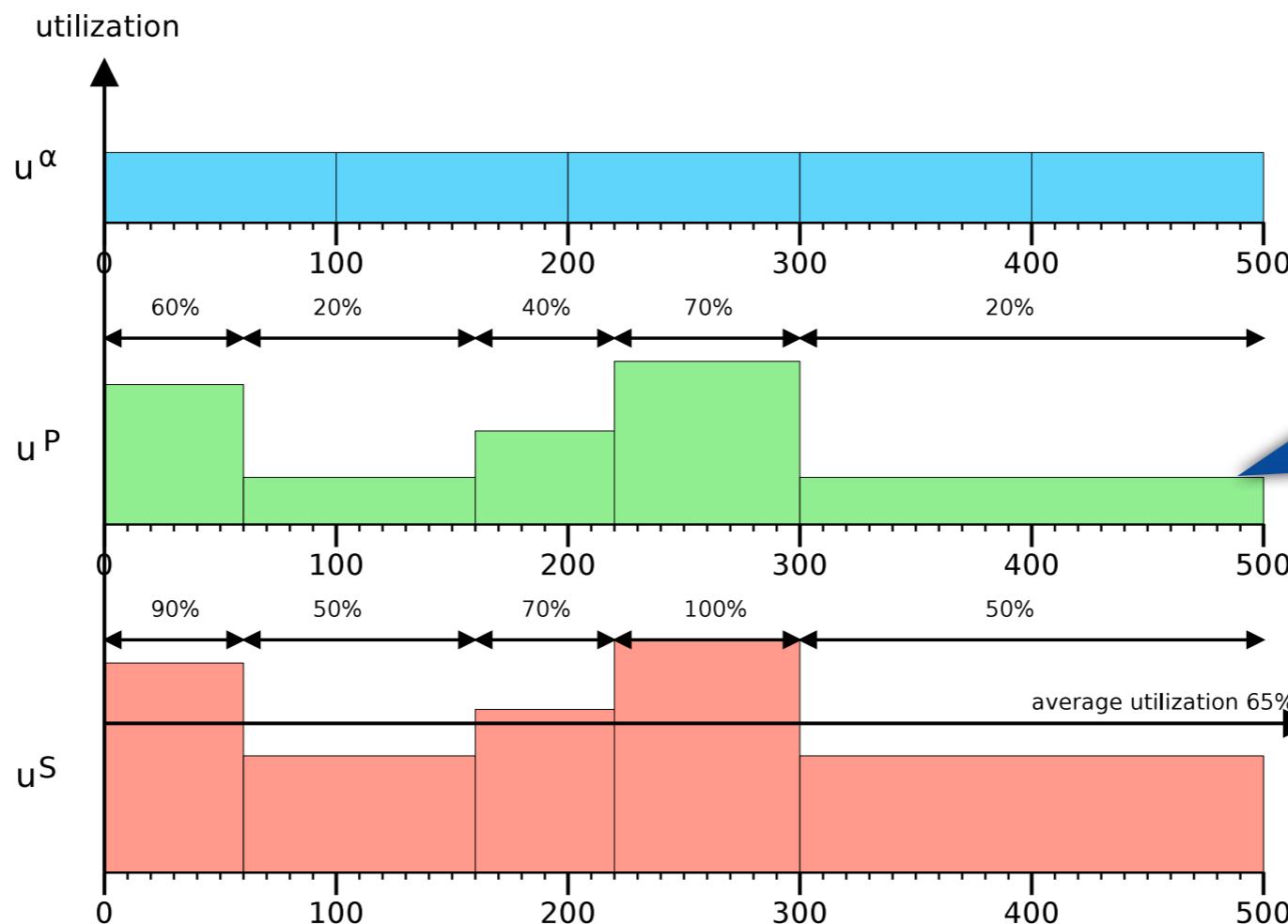
Assume a simple power model ( $P \propto V^2$ )





# Look-ahead online FS-VBS

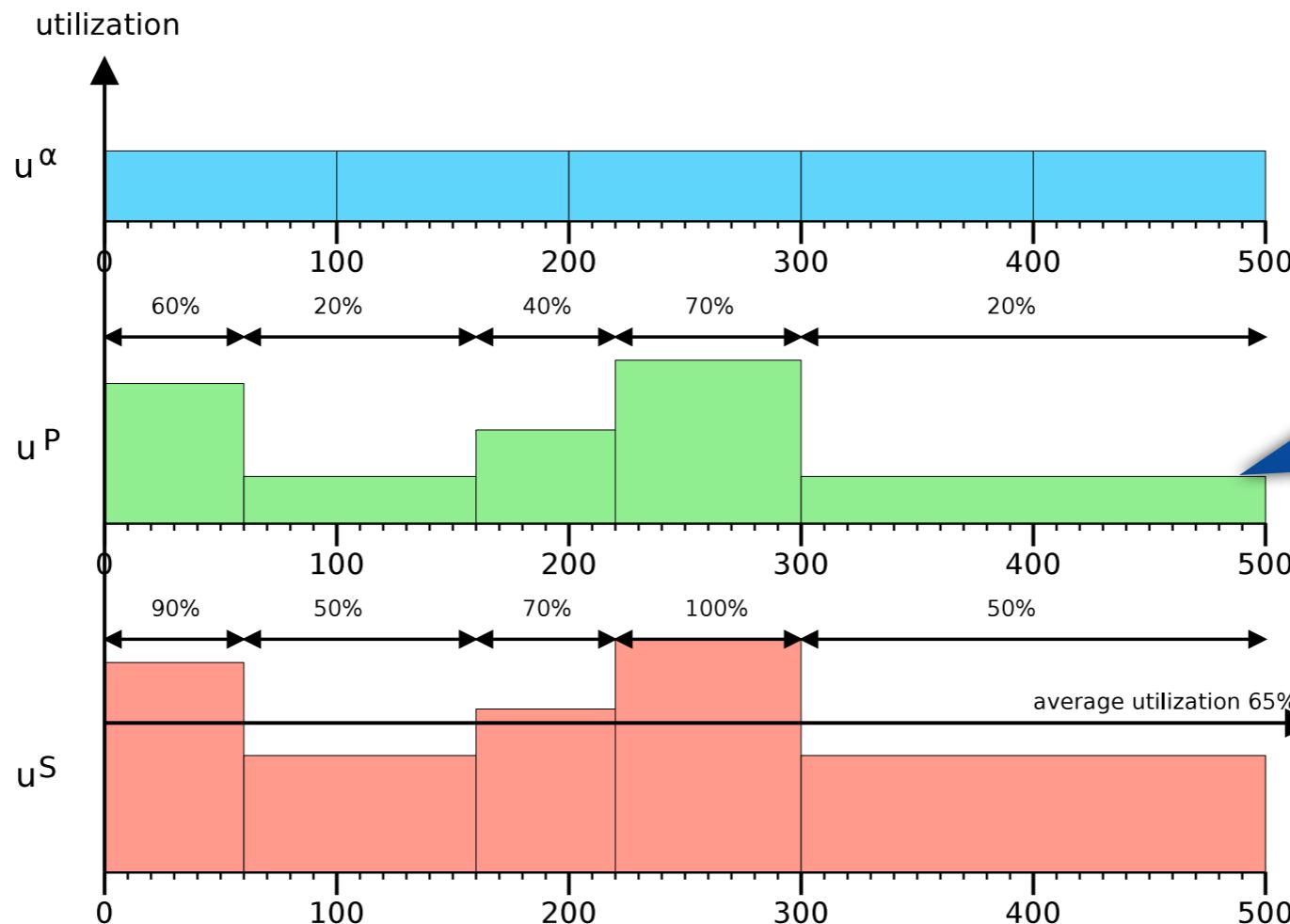
Assume a simple power model ( $P \propto V^2$ )





# Look-ahead online FS-VBS

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

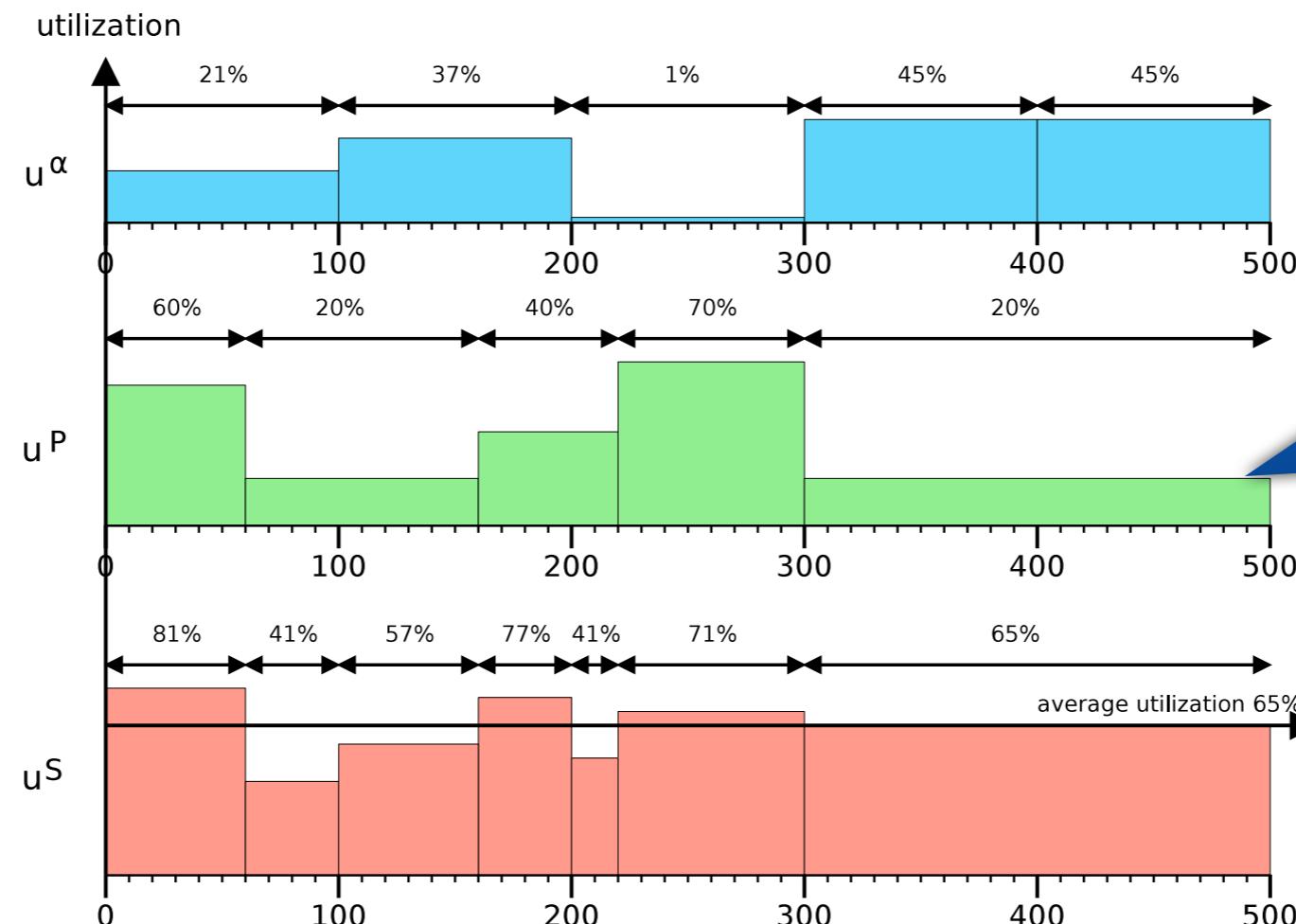


Modify the limits in each period (whenever possible)  
s.t. the utilization approximates the average utilization



# 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

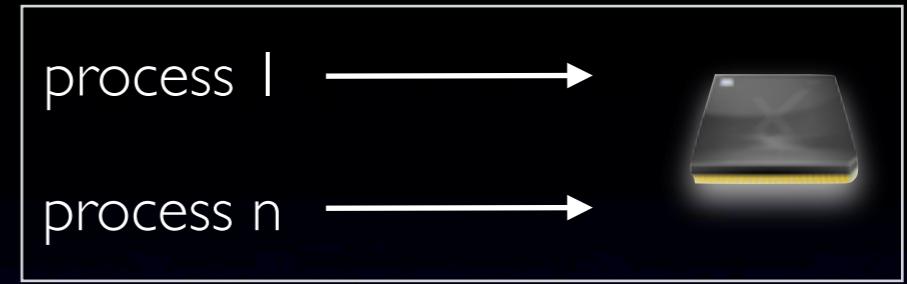


# Conclusions



# Conclusions

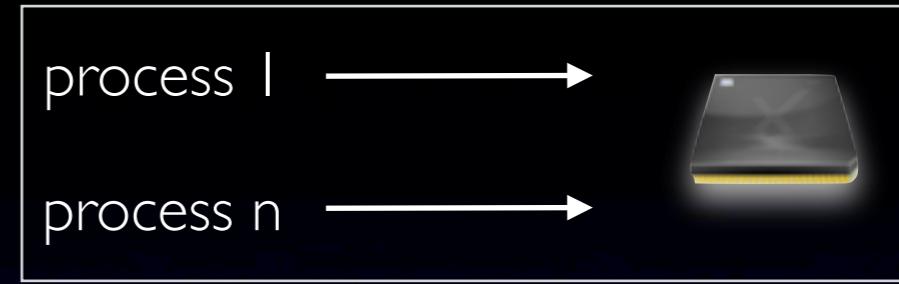
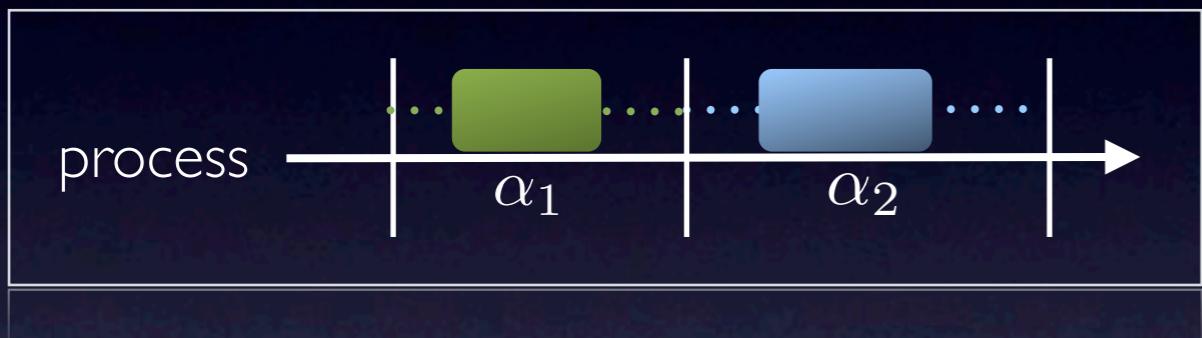
- Reservation-based scheduling for temporal isolation





# Conclusions

- Reservation-based scheduling for temporal isolation

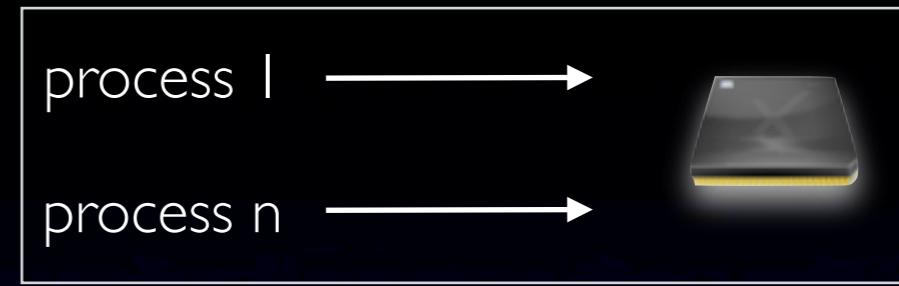
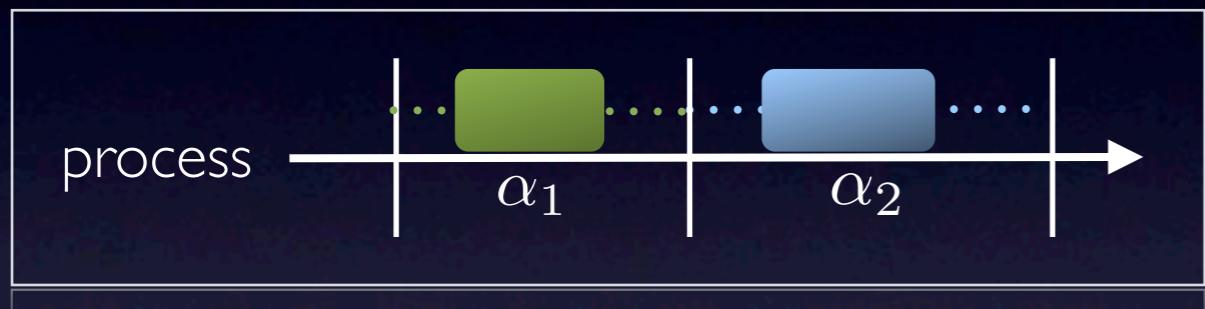


- VBS enables variable execution speed



# Conclusions

- Reservation-based scheduling for temporal isolation



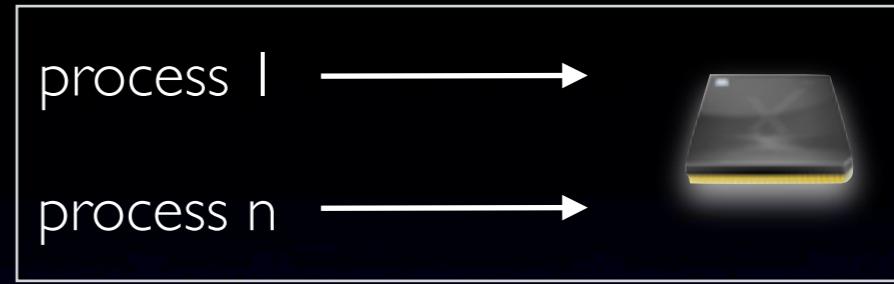
- Overhead accounting



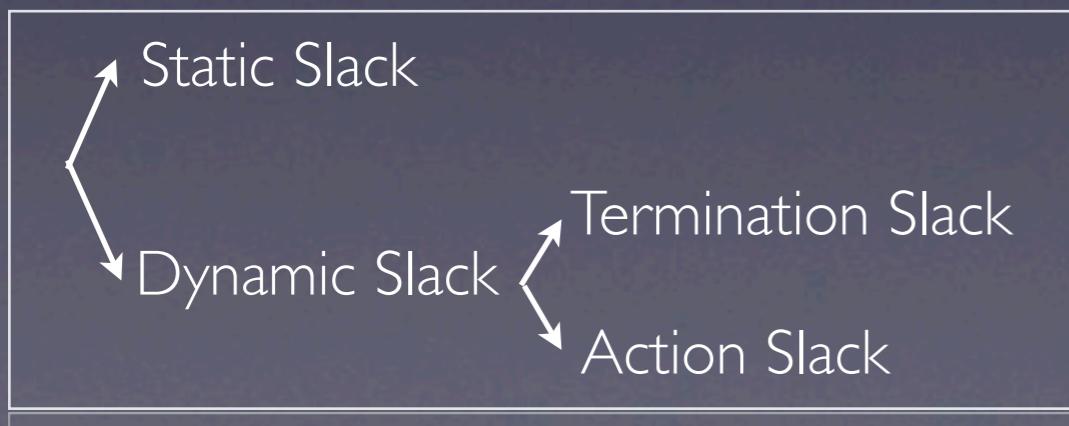


# Conclusions

- Reservation-based scheduling for temporal isolation



- Overhead accounting



- Power-aware VBS



# Thank you

S.S. Craciunas, C.M. Kirsch, and A. Sokolova - Power-Aware Temporal Isolation with Variable-Bandwidth Servers. In International Conference on Embedded Software (EMSOFT), ACM, 2010.

S.S. Craciunas, C.M. Kirsch, and A. Sokolova - Response Time versus Utilization in Scheduler Overhead Accounting. In Real-Time and Embedded Technology and Applications Symposium (RTAS), pp. 291-300, IEEE, 2010.

S.S. Craciunas, C.M. Kirsch, H. Payer, H. Röck, and A. Sokolova - Programmable Temporal Isolation through Variable-Bandwidth Servers. In IEEE Symposium on Industrial Embedded Systems (SIES), pp. 171-180, IEEE, 2009.