Workload-oriented Programming

Silviu S. Craciunas

Department of Computer Sciences
University of Salzburg, Austria
-
joint work with
C. Kirsch, H. Röck, and A. Sokolova

July 2008

Tiptoe RTOS

Tiptoe

Prototype real-time operating system.

`http://tiptoe.cs.uni-salzburg.at`

- Processes are sequences of actions
- Actions may have workload parameters that determine the amount of work involved
- We are interested in the temporal performance of actions with respect to the workload involved

Example

Consider a process that:

- allocates memory for a video stream
- reads the video stream from a network connection
- compresses the stream
- stores it on disk
- deallocates the used memory

Example

Pseudo-code

```
loop {
  int number_of_frames=determine_rate();
  allocate_memory(number_of_frames);
  read_from_network(number_of_frames);
  compress_data(number_of_frames);
  write_to_disk(number_of_frames);
  deallocate_memory(number_of_frames);
} until (done); <– requirements
```

Example

Pseudo-code

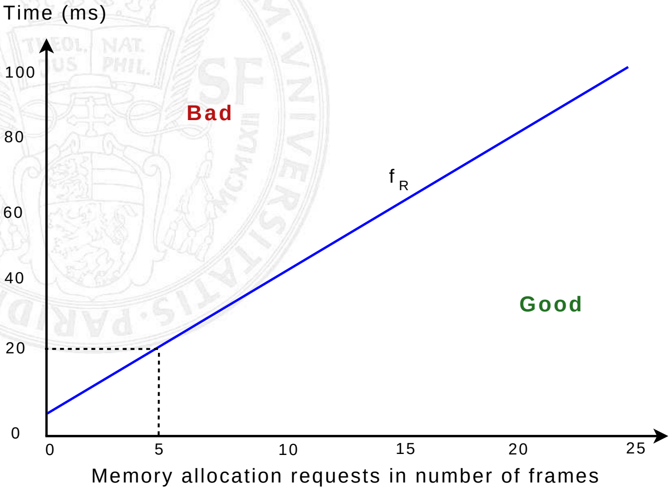
```
loop {
  int number_of_frames=determine_rate();

  allocate_memory(number_of_frames);<– requirements 1
  read_from_network(number_of_frames);<– requirements 2

  compress_data(number_of_frames);<– requirements 3

  write_to_disk(number_of_frames);<– requirements 4
  deallocate_memory(number_of_frames);<– requirements 5
} until (done);
```

Example

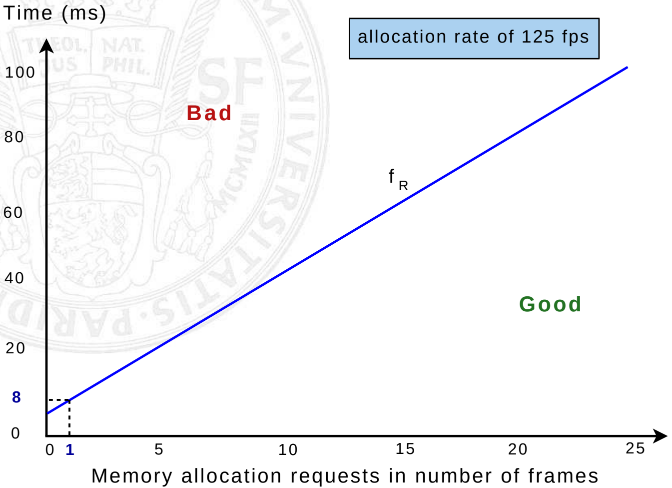
Pseudo-code

```
loop {
  int number_of_frames=determine_rate();
> allocate_memory(number_of_frames);
  read_from_network(number_of_frames);
  compress_data(number_of_frames);
  write_to_disk(number_of_frames);
  deallocate_memory(number_of_frames);
} until (done);
```

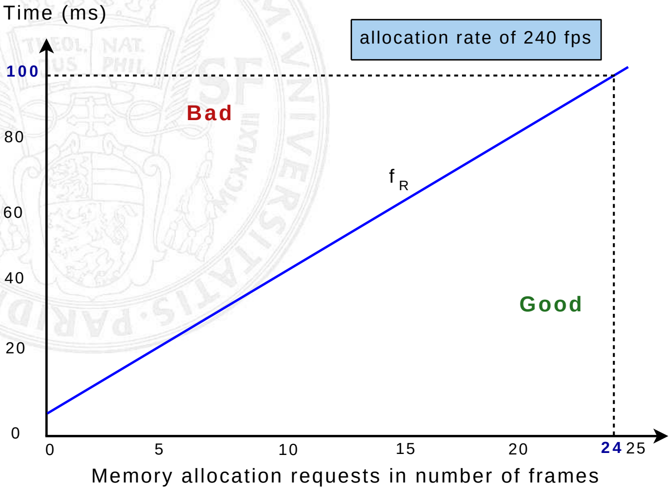
Response-time function



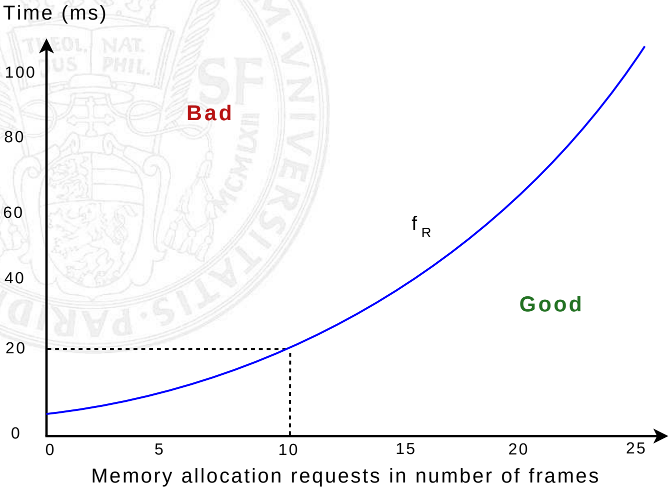
Improving latency over throughput



Improving throughput over latency



Response-time function



Response-time function

Informally

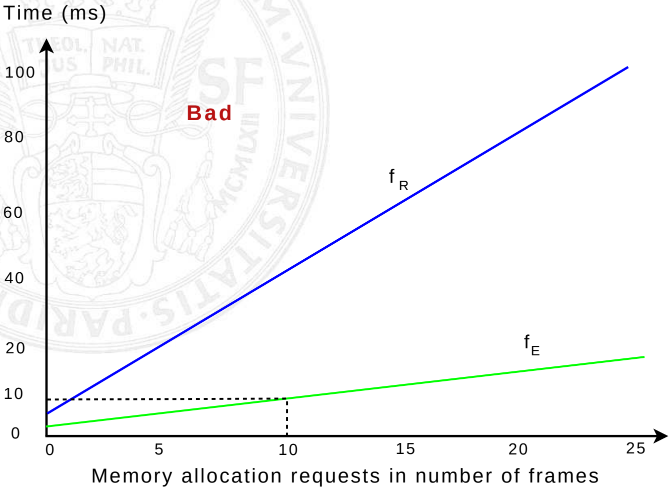
The response-time (RT) function f_R characterizes the action's response time bound for a given workload, independently of any previous or concurrent actions.

Formally

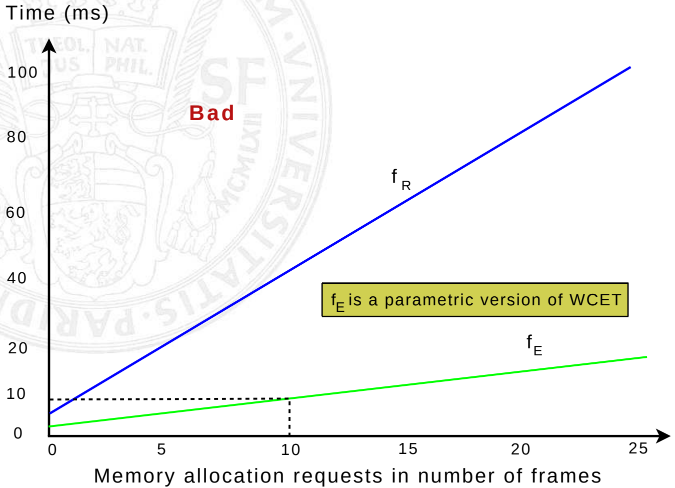
A response-time (RT) function is a discrete function

$$f_R : \mathbb{N} \longrightarrow \mathbb{Q}^+$$

Execution-time function



Execution-time function



Execution-time function

Informally

The execution-time (ET) function f_E characterizes the action's execution time bound for a given workload, in the absence of any concurrent actions.

Formally

An execution-time (ET) function is a discrete function

$$f_E : E_D \rightarrow \mathbb{Q}^+.$$

where $E_D \subseteq \mathbb{N}$ is called the execution domain.

Scheduling and Admission

Process scheduling

How do we efficiently schedule processes on the level of individual process actions?

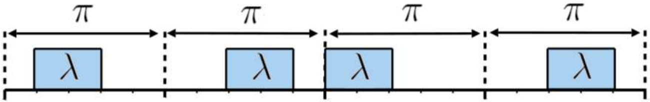
Process admission

How do we efficiently test schedulability of newly arriving processes?

Virtual periodic resources

Virtual periodic resources

Period π Limit λ Utilization $\frac{\lambda}{\pi}$



We use our modified version of the CBS[1] which allows different π and λ for a process.

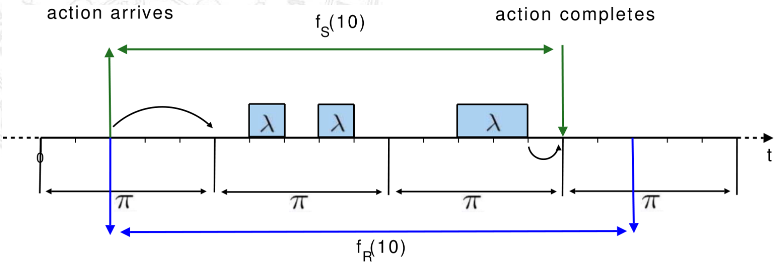
[1] Giorgio Buttazzo and Enrico Bini - *Optimal Dimensioning of a Constant Bandwidth Server* in Proc. RTSS 2006

Virtual periodic resources

- Each process declares a finite set of virtual periodic resources
- Each process action of a process uses exactly one virtual periodic resource declared by the process

Scheduled response time

The scheduled response time $f_S(w)$ is the time from the arrival of an action until it completes.



Schedulabilty result

A set of processes is schedulable under EDF if

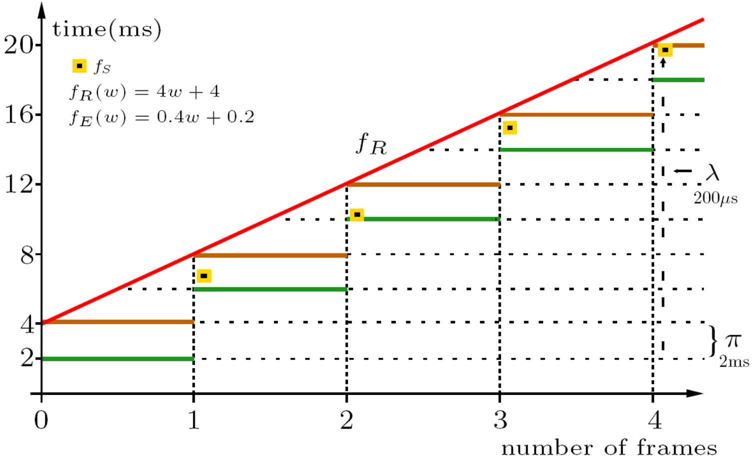
- For every action of every process, λ and π "sample" f_R such that

$$\forall w \in E_D, f_S(w) \leq f_R(w)$$

- The schedulability test holds for the set of processes

$$\sum_P \max_R \frac{\lambda_{PR}}{\pi_{PR}} \leq 1$$

Response time sampling

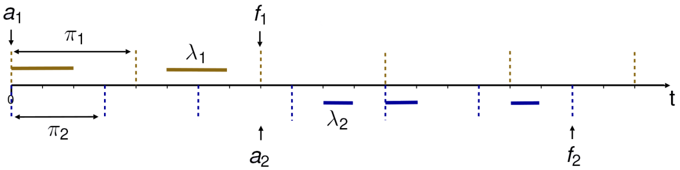


Tiptoe compositionality

$$\forall f_S, f_S{}', \forall w \in E_D$$

$$0 \leq |f_S(w) - f_S{}'(w)| \leq \pi_a - 1$$

if the schedulability test holds



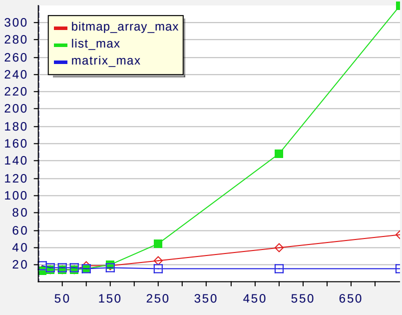
Tiptoe compositionality

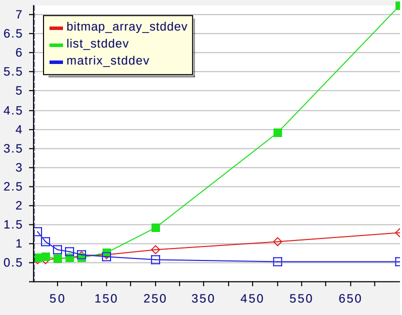
Response-time jitter

Any individual action a of a process maintains its response time within $\pi_a - 1$ even when adding/removing concurrent processes.

Implementation and performance

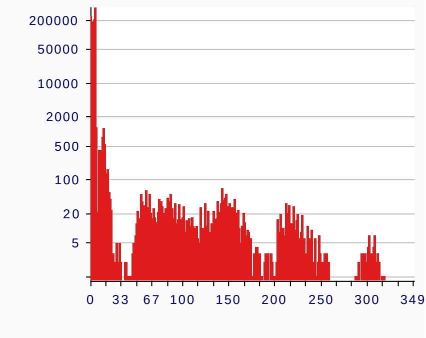
	list	array	matrix
time	$O(n^2)$	$O(\log(t) + n \cdot \log(t))$	$\Theta(t)$
space	$\Theta(n)$	$\Theta(t + n)$	$\Theta(t^2 + n)$





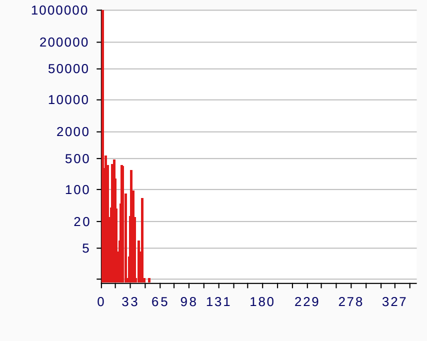
Implementation and performance

	list	array	matrix
time	$O(n^2)$	$O(\log(t) + n \cdot \log(t))$	$\Theta(t)$
space	$\Theta(n)$	$\Theta(t+n)$	$\Theta(t^2+n)$



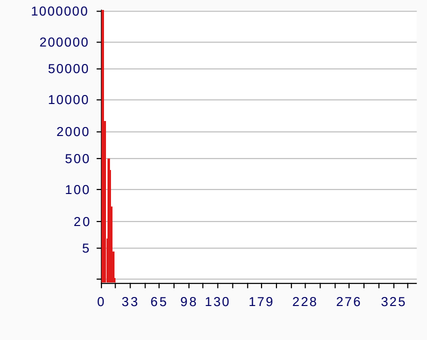
Implementation and performance

	list	array	matrix
time	$O(n^2)$	$O(\log(t) + n \cdot \log(t))$	$\Theta(t)$
space	$\Theta(n)$	$\Theta(t + n)$	$\Theta(t^2 + n)$



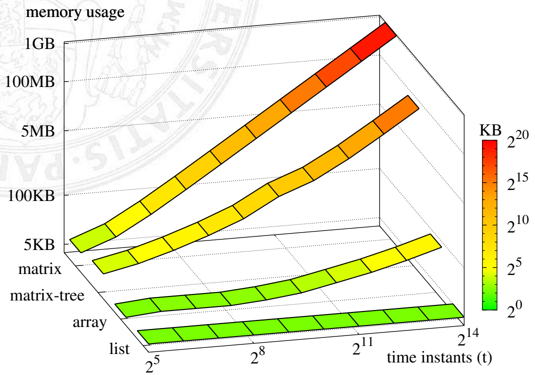
Implementation and performance

	list	array	matrix
time	$O(n^2)$	$O(\log(t) + n \cdot \log(t))$	$\Theta(t)$
space	$\Theta(n)$	$\Theta(t + n)$	$\Theta(t^2 + n)$



Implementation and performance

	list	array	matrix
time	$O(n^2)$	$O(\log(t) + n \cdot \log(t))$	$\Theta(t)$
space	$\Theta(n)$	$\Theta(t + n)$	$\Theta(t^2 + n)$



Conclusion

The model enables

- trading off throughput and latency
- sequential and concurrent process composition
- controlling the response-time variance (jitter) of individual process actions

Conclusion

The model enables

- trading off throughput and latency
- sequential and concurrent process composition
- controlling the response-time variance (jitter) of individual process actions

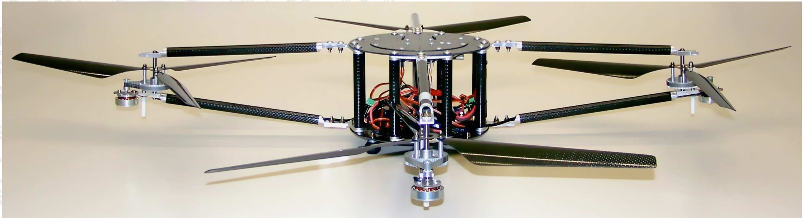
Conclusion

The model enables

- trading off throughput and latency
- sequential and concurrent process composition
- controlling the response-time variance (jitter) of individual process actions

Current work

Fly the JAviator with Tiptoe



Homepage

`http://javiator.cs.uni-salzburg.at`

Thank you!

Scheduled response time

$\forall w \in U_D, f_S(w) \leq f_R(w) + \pi$ if

- $\dfrac{\lambda}{\pi} = c_U$
- π divides $f_R(w)$ evenly
- $\sum_P \max_R \frac{\lambda_{PR}}{\pi_{PR}} \leq 1$

Scheduled response time

$\forall w \in U_D, f_S(w) \leq f_R(w)$ if

- $\dfrac{\lambda}{\pi} = c_U$
- $0 < \pi \leq d_R - \frac{d_E}{c_U}$
- π divides d_R and $f_R(w) - d_R$ evenly
- $\sum_P \max_R \frac{\lambda_{PR}}{\pi_{PR}} \leq 1$