TECHNISCHE UNIVERSITÄT DRESDEN

**Faculty of Computer Science** Institute of Systems Architecture, Operating Systems Group

# REAL-TIME vs. SYSTEMS

MICHAEL ROITZSCH

# DEFINITION

system whose quality depends on the **functional correctness** of computations and the **time** those results are produced

# LINES OF THOUGHT

| **real-time school** | **systems school** |
|---|---|
| mathematically sound | practical systems |
| simplifying assumptions | baroque details |
| random task sets | actual applications |
| strong contract | usable interface |
| theory | implementation |

# THE OLD DAYS

W. A. Horn: Some Simple Scheduling Algorithms. Naval Research Logistics Quarterly, Volume 21(1): pp. 177–185, March 1974

- scheduling was part of plant operation
- work items were scheduled to machines
- algorithms were known, but not by today's names

# THEORY

Chang L. Liu, James W. Layland: Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. Journal of the ACM, Volume 20(1): pp. 46–61, January 1973

- periodic task model
- first description of RMS and EDF
- including the utilization bounds
- with proof

# PRACTICE

Clifford W. Mercer, Stefan Savage, Hideyuki Tokuda: Processor Capacity Reserves: Operating System Support for Multimedia Applications. Proceedings of the IEEE International Conference on Multimedia Computing and Systems (MCS), pp. 90–99, May 1994

- introduced the idea of tracking actual execution time to police overruns
- the reservation concept was born
- implemented in RT-Mach

# RESOURCE KERNEL

Raj Rajkumar, Kanaka Juvva, et al.: Resource Kernels: A Resource-Centric Approach to Real-Time and Multimedia Systems. Proceedings of the 1998 Multimedia Computing and Networking Conference (MMCN), pp. 150–164, January 1998

- time as a first-class, globally managed shared resource
- based on the periodic task model
- extended with inheritance, reservations

# DISASTER STRIKES

- the 90's: multimedia driving practical real-time work
- this could have been a wonderful world, but then…


**… CPUs got fast.**

# DIRECTIONS

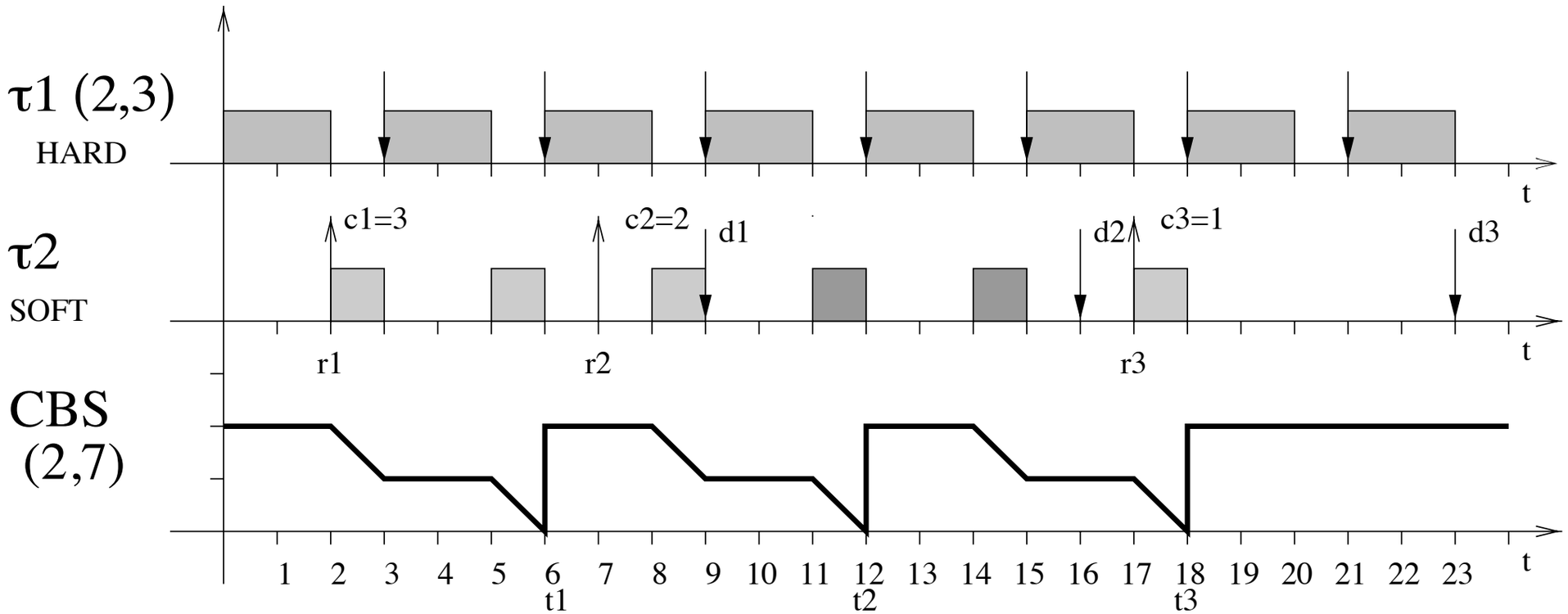
**real-time school** ⟷ **systems school**

**real-time folks building systems**

**systems folks designing time contracts**

Luca Abeni, Giorgio Buttazzo: Integrating Multimedia Applications in Hard Real-Time Systems. Proceedings of the 19th IEEE Real-Time Systems Symposium (RTSS), pp. 4–13, December 1998

- “server” as a real-time concept
- allocation of time with period and budget
- jobs enqueued to be run by the server
- deadlines postponed on budget overrun




- illusion of a dedicated, slower processor
- virtual fluid-flow of CPU time
- individual job deadlines hard to manage

# SLACK

Luca Marzario, Giuseppe Lipari, et al.: IRIS: A New Reclaiming Algorithm for Server-Based Real-Time Systems. Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), pp. 211–218, May 2004

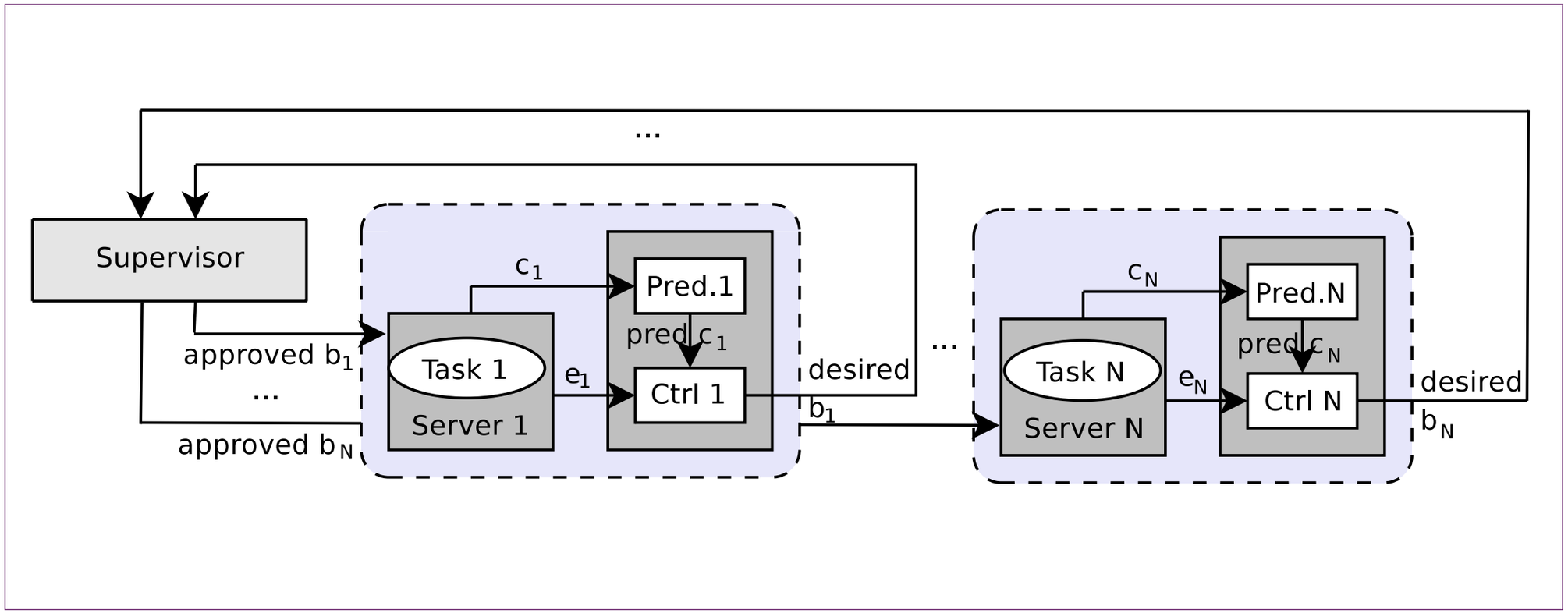
- "slack time" ≈ nothing urgent to do
- redirect slack to help out on overruns

# ADAPTING

Luigi Palopoli, Tommaso Cucinotta, et al.: AQuoSA – Adaptive Quality of Service Architecture. Software: Practice and Experience, Volume 39(1): pp. 1–31, January 2009

- adapt server budget to varying application demand
- control loop observes tasks, QoS manager tunes parameters

# ADAPTING


...
Supervisor
approved $b_1$
...
approved $b_N$
$c_1$
Pred.1
pred $c_1$
Task 1
$e_1$
Ctrl 1
Server 1
desired
$b_1$
...
$c_N$
Pred.N
pred $c_N$
Task N
$e_N$
Ctrl N
Server N
desired
$b_N$

# SUMMARY

**adaptation mechanisms**

- ✔ serve jobs by demand
- ✘ complex infrastructure

↓

**fluid flow abstraction**

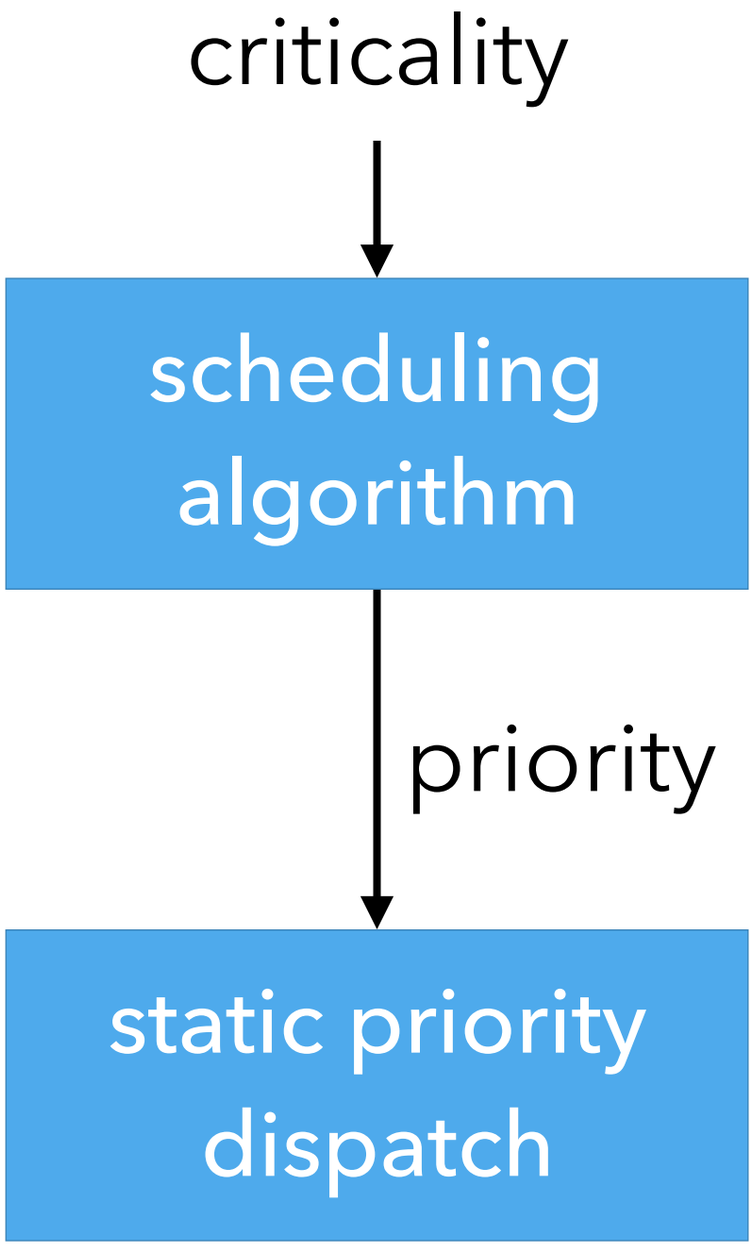
- ✔ flexible on overruns
- ✘ job behavior less clear

↓

**periodic tasks**

- ✔ predictable job behavior
- ✘ inflexible

# PRIORITIES


criticality
scheduling algorithm
priority
static priority dispatch

# MIXED CRITICALITY

Steve Vestal: Preemptive Scheduling of Multi-Criticality Systems with Varying Degrees of Execution Time Assurance. Proceedings of the 28th IEEE Real-Time Systems Symposium (RTSS): pp. 239–243, December 2007

- criticality describes which tasks should be preferred under resource overload
- overload is allowed because of tiered assurance of timing parameters

# DIRECTIONS

**real-time school** ⟷ **systems school**

**real-time folks building systems**

**systems folks designing time contracts**

Kenneth J. Duda, David R. Cheriton: Borrowed-Virtual-Time (BVT) Scheduling: Supporting Latency-Sensitive Threads in a General-Purpose Scheduler. Proceedings of the 17th ACM Symposium on Operating Systems Principles (SOSP), pp. 261–276, December 1999

- each thread carries a virtual timestamp
- increases when thread runs, inversely proportional to its weight
- thread with smallest timestamp runs

- this is also the principle behind the Completely Fair Scheduler (CFS) in Linux
- **warp time** controls dispatch latency
- effective virtual time =
  actual virtual time – warp time
- effective time is used for scheduling
- warping describes short-term urgency

# WEIGHT POLICE

Ion Stoica, Hussein Abdel-Wahab, et al.: A Proportional Share Resource Allocation Algorithm for Real-Time, Time-Shared Systems. Proceedings of the 17th IEEE Real-Time Systems Symposium (RTSS), pp. 288–299, December 1996

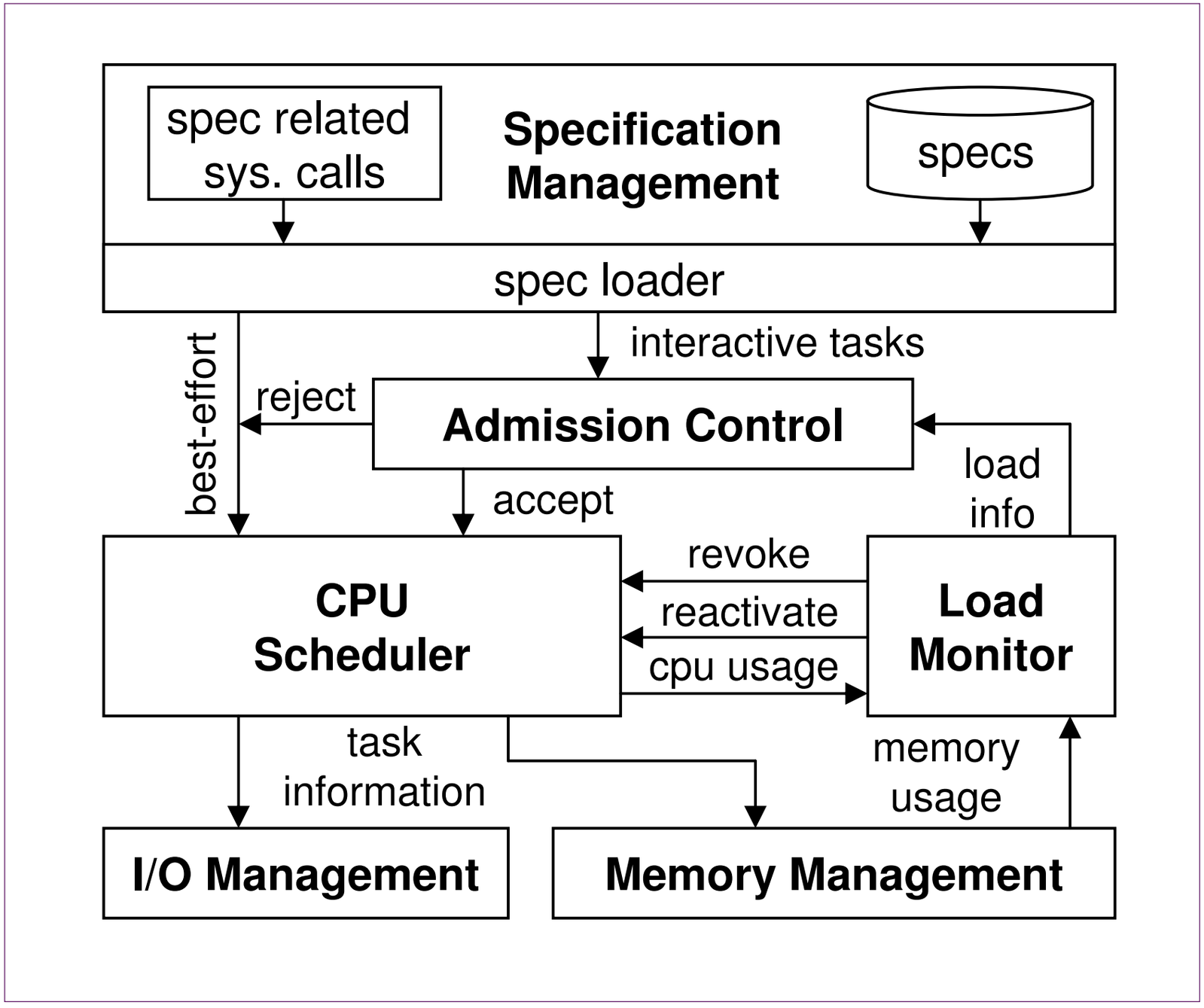
- using fair-share schedulers for real time
- careful policing of sharing weights

# REDLINE

Ting Yang, Tongping Liu, et al.: Redline: First Class Support for Interactivity in Commodity Operating Systems. Proceedings of the 8th USENIX Symposium on Operating Systems Design and Implementation (OSDI), pp. 73–86, December 2008

- integrated management of CPU, memory and disk to improve responsiveness
- external specification files

# REDLINE


spec related sys. calls
Specification Management
specs
spec loader
interactive tasks
best-effort
reject
Admission Control
accept
load info
CPU Scheduler
revoke
reactivate
cpu usage
Load Monitor
task information
memory usage
I/O Management
Memory Management

# REDLINE

```
</usr/bin/mplayer:Iact:5:30:IR:—:—>
```

application

class

budget

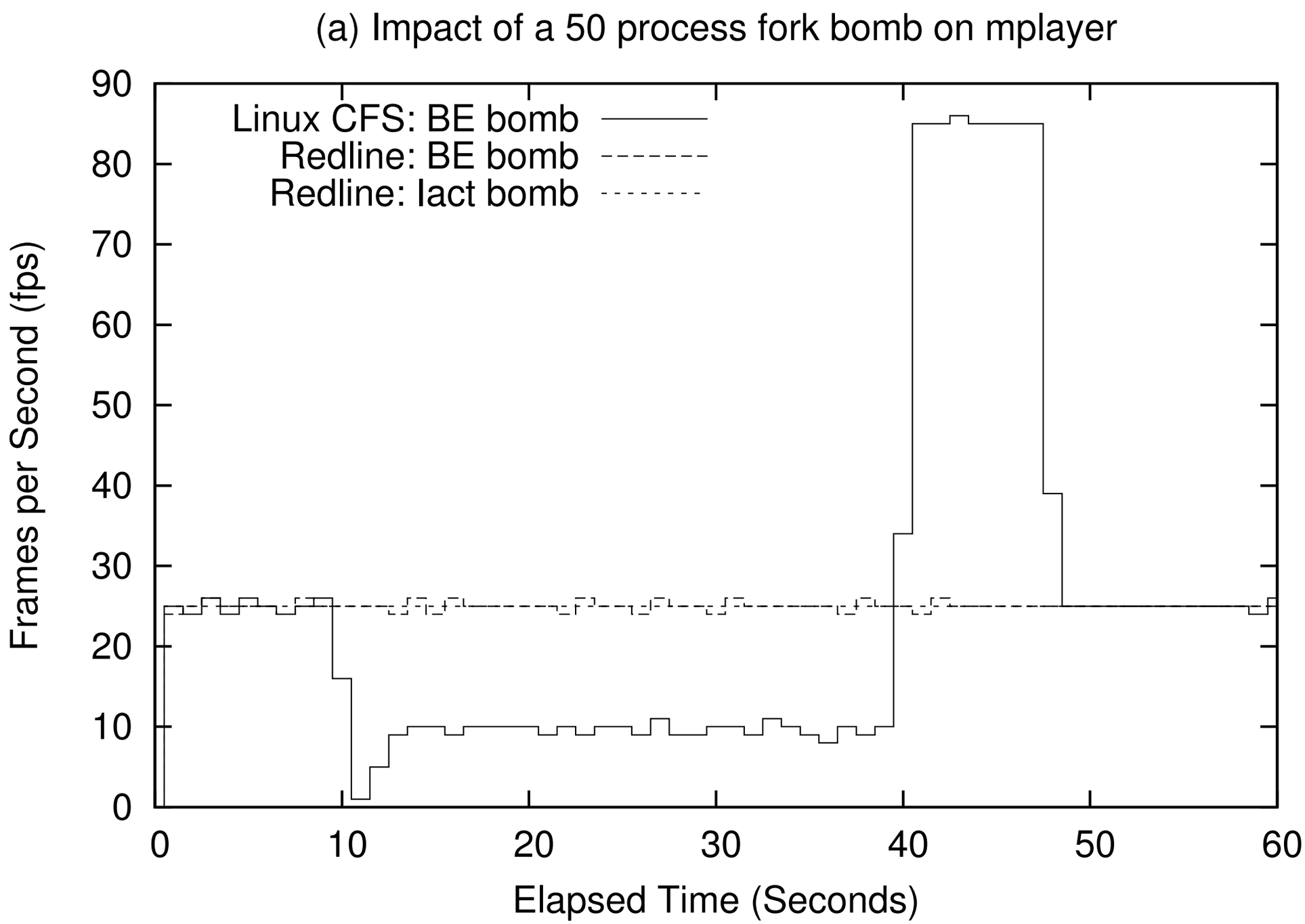
period

flags

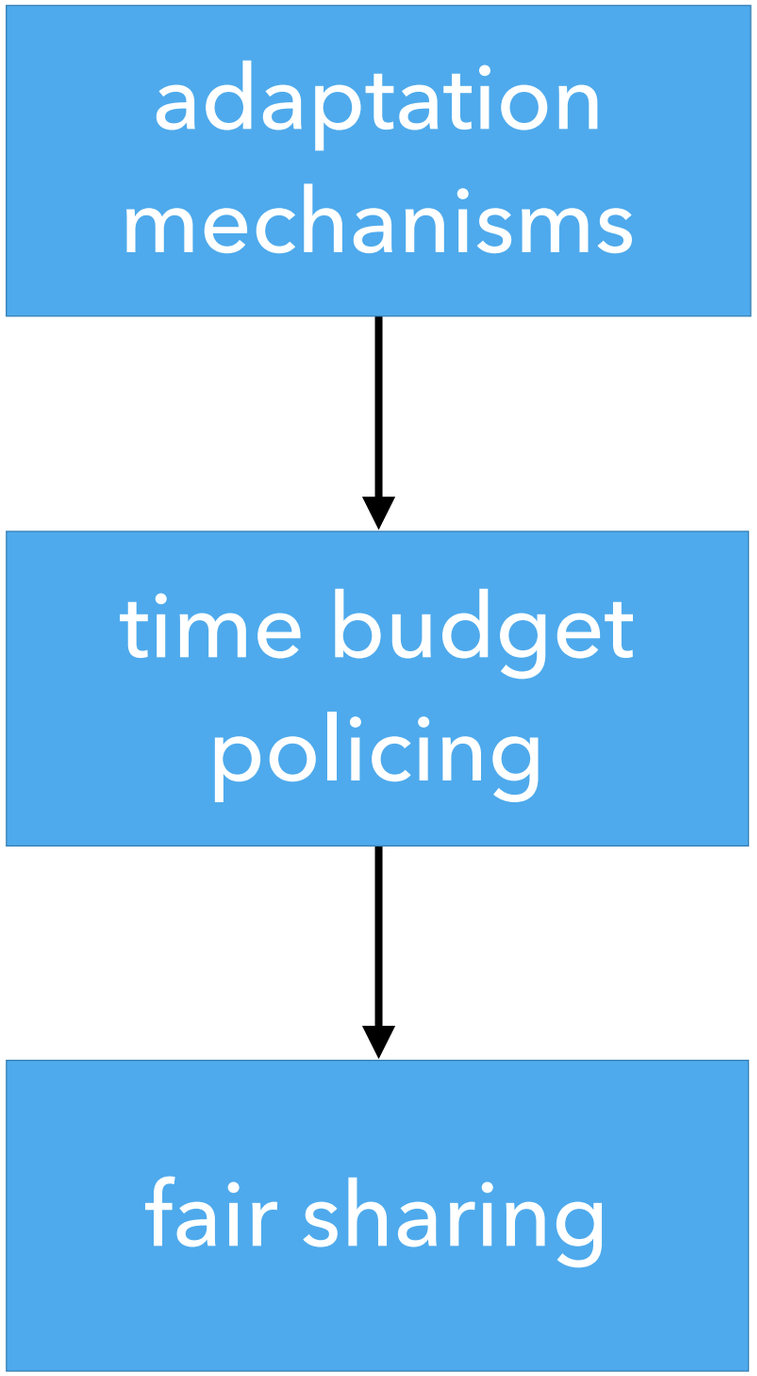
working-set keep-alive

I/O priority

# REDLINE


(a) Impact of a 50 process fork bomb on mplayer
Linux CFS: BE bomb
Redline: BE bomb
Redline: Iact bomb
Frames per Second (fps)
Elapsed Time (Seconds)
0
10
20
30
40
50
60
70
80
90

# SUMMARY




✔ serve jobs by demand

✘ complex enforcement

✔ timing guarantees

✘ only long-term control

✔ intuitive and simple

✘ no timing control

# SUMMARY

- real-time work tends to start with **theory** and **strong enforcement**
  - practical use often is only an afterthought
- systems work tends to start with **best effort systems**
  - strong guarantees are just an afterthought
- golden middle ground still not found