artist

The ArtistDesign
European Network of Excellence
on Embedded Systems Design

http://www.artist-embedded.org/

Showcase of the Main Results

DATE Conference, March 15th, 2012

SEVENTH FRAMEWORK PROGRAMME

ArtistDesign European NoE: Showcase of the Main Results

DATE Conference, March 15th, 2012

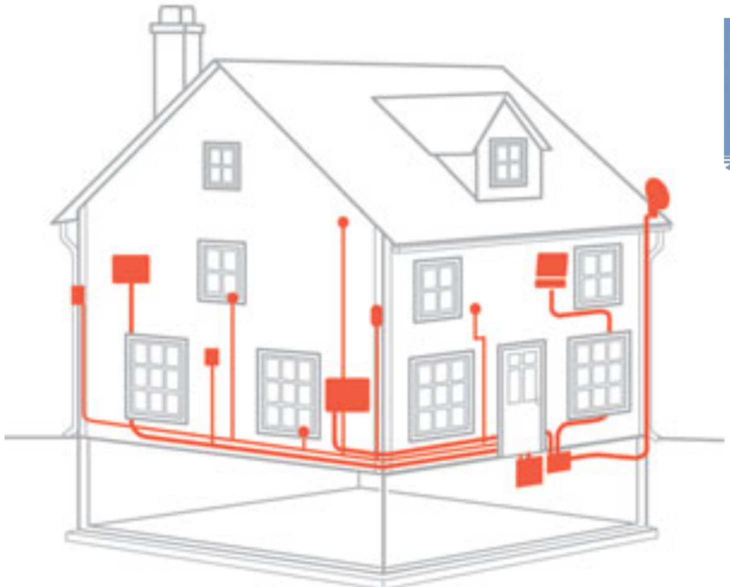
Achievements and Perspectives

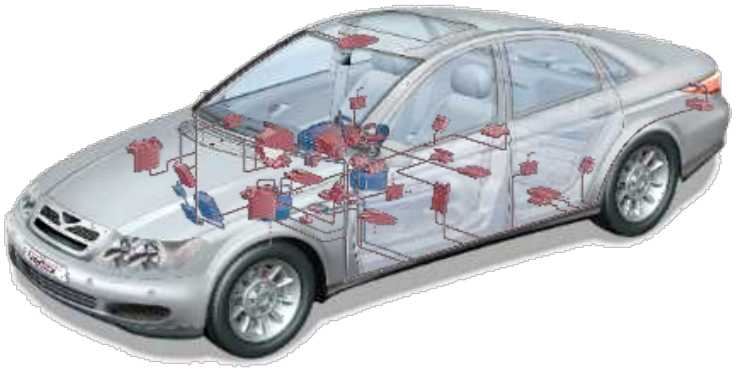
Operating Systems and Networks

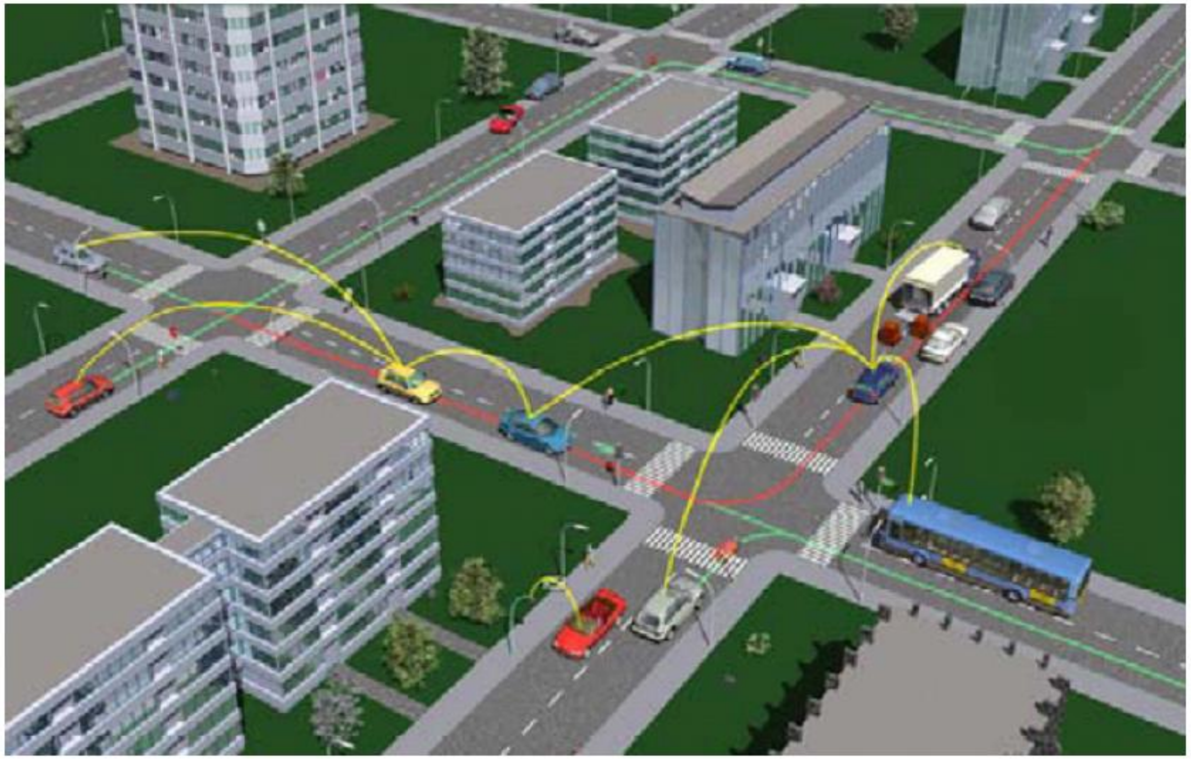
leader: Giorgio Buttazzo

Scuola Superiore Sant'Anna, Pisa, Italy

Embedded Systems have become more complex, and characterized by dynamic behavior and distributed organization







Overall High-Level Objectives and Vision

Provide a more efficient and predictable support (at the OS and Network level) to the development of future embedded systems. In particular:

- Allow simple and flexible resource management to cope with the growing complexity;
- Take advantage of multi-core platforms;
- Support distributed computing to deal with the ubiquitous nature of the computing infrastructure;
- Increase system adaptivity to react to environmental changes.

Cluster activities

Real-Time Networks

Scheduling and Resource Management

Resource-Aware Operating Systems

Year 4 Review
Dresden, March 16th, 2012

Achievements and Perspectives:

Resource-Aware Operating Systems

Activity leader: *Giorgio Buttazzo*
Scuola Superiore Sant'Anna
Pisa, Italy

Objectives

Exploit the expertise in the NoE to make operating systems more

- predictable (in terms of timing behavior)
- efficient (in terms of resource usage)
- robust (to tolerate overload conditions)
- easy to use (to simplify user interface)

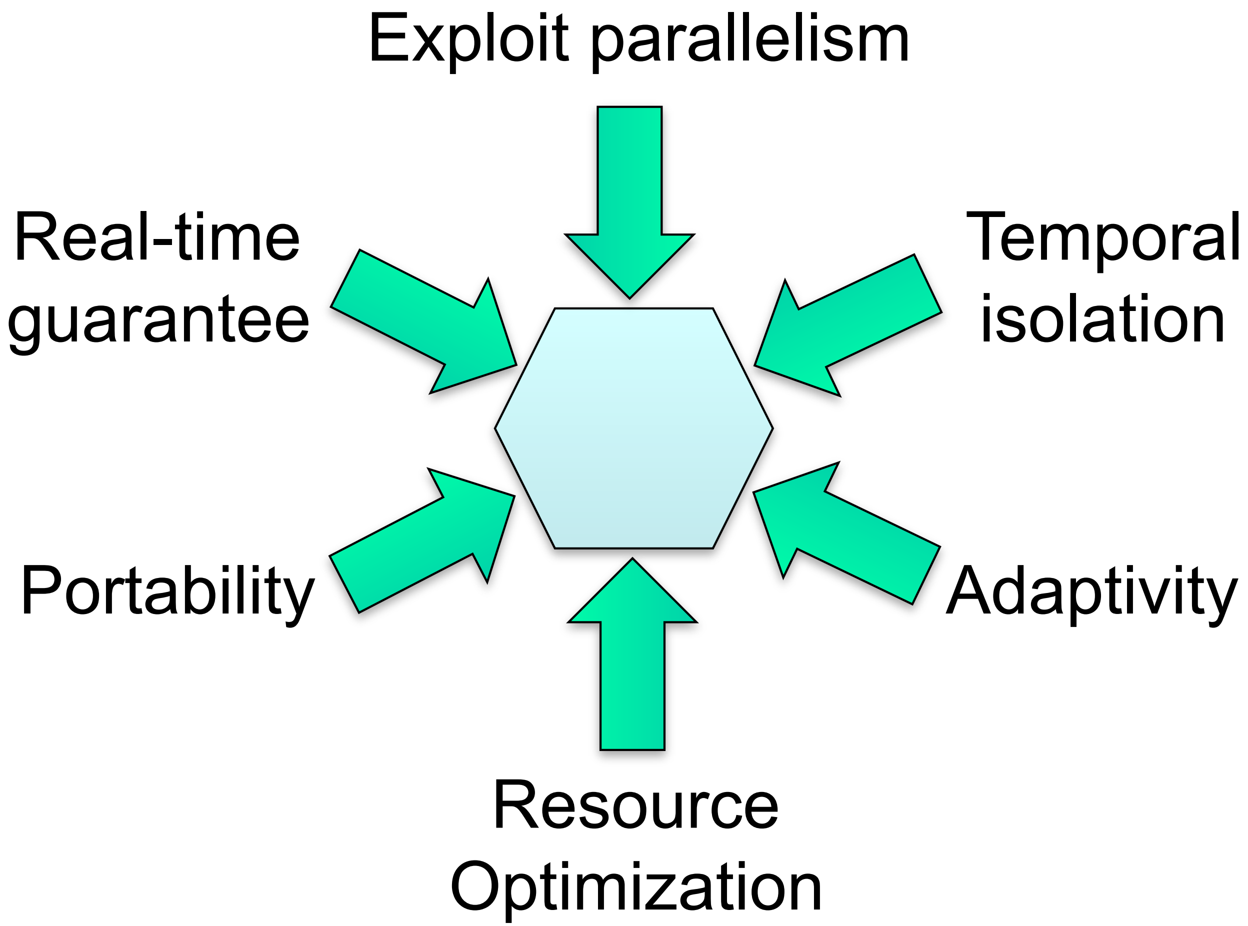
Approach

- Investigate novel kernel **methodologies**
  - Resource reservation
  - Contract-based scheduling
  - Limited preemptive scheduling
  - Energy-aware policies

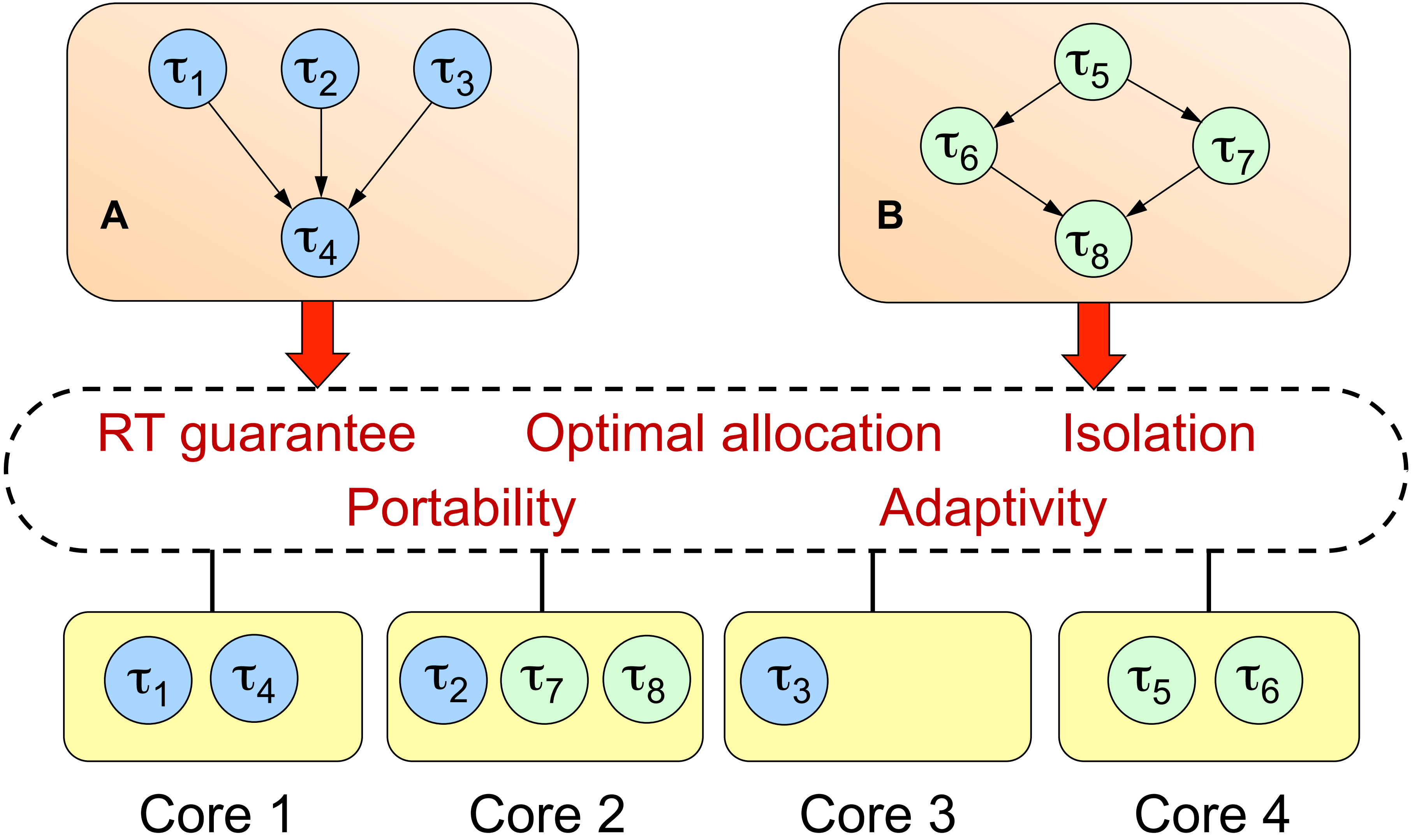
Emphasis on multi-core platforms

- Implement these techniques in **existing RTOSes**
- Provide **appropriate tools**

Multiple goals



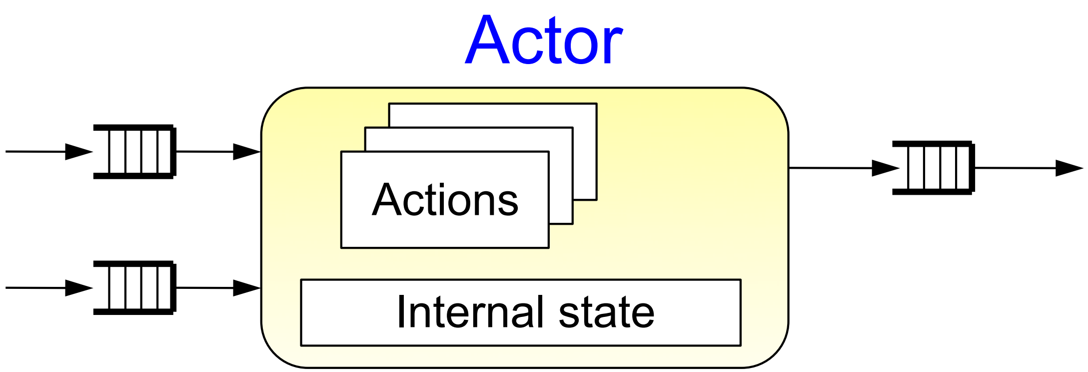
System model



Expressing parallelism

Parallelism can be expressed by using a suitable dataflow language, like CAL [UC Berkeley, 2003].

- It describes algorithms through a set of modular components (actors), communicating through I/O ports:



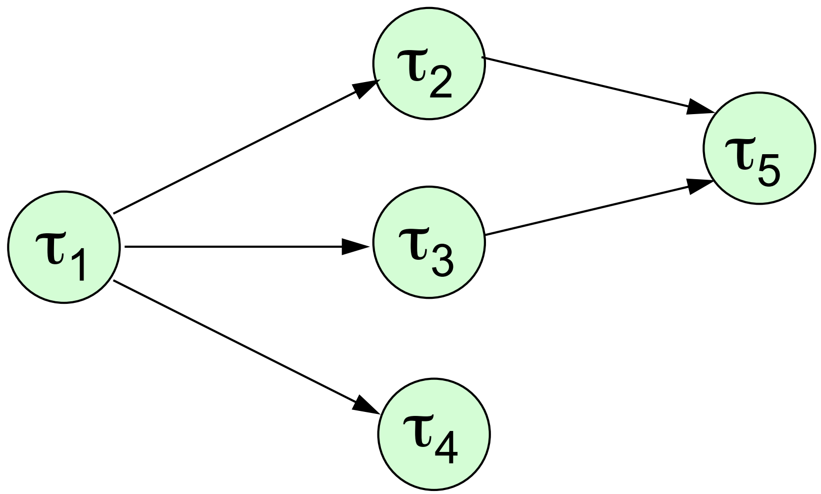
- Actions read input tokens, modify the internal state, and produce output tokens.

Application model

- An application can be modeled as a task graph with precedence relations:

Task τ_i

A sequential portion of code that cannot be further parallelized



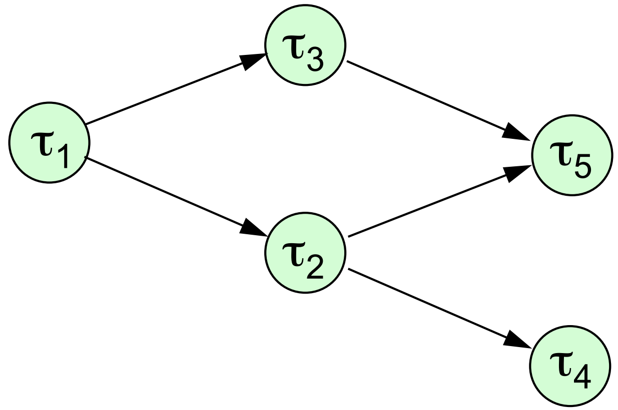
A task graph specifies the maximum level of parallelism

Application model

- Each application is sporadic:
  - Minimum interarrival time (period) T
  - relative deadline D
- Tasks are fully preemptive

Application parameters:

$\{C_1, C_2, C_3, C_4, C_5\}$, D, T



Achieving Temporal Isolation

Temporal Isolation

Property of a multi-application system in which the performance of an application does not depend on the execution of the others.

The performance of an application only depends on:

- Its own computational demand;
- The amount of allocated resources.

→ Resource Reservation

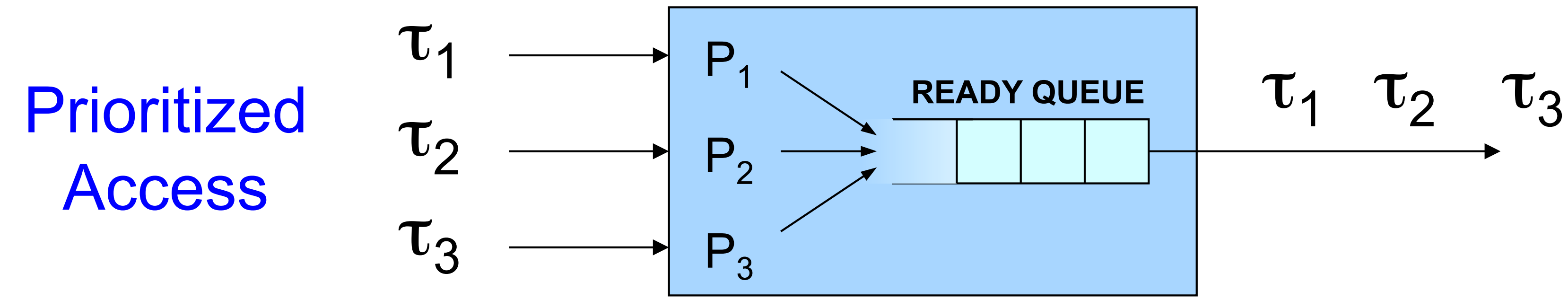
Achieving Temporal Isolation

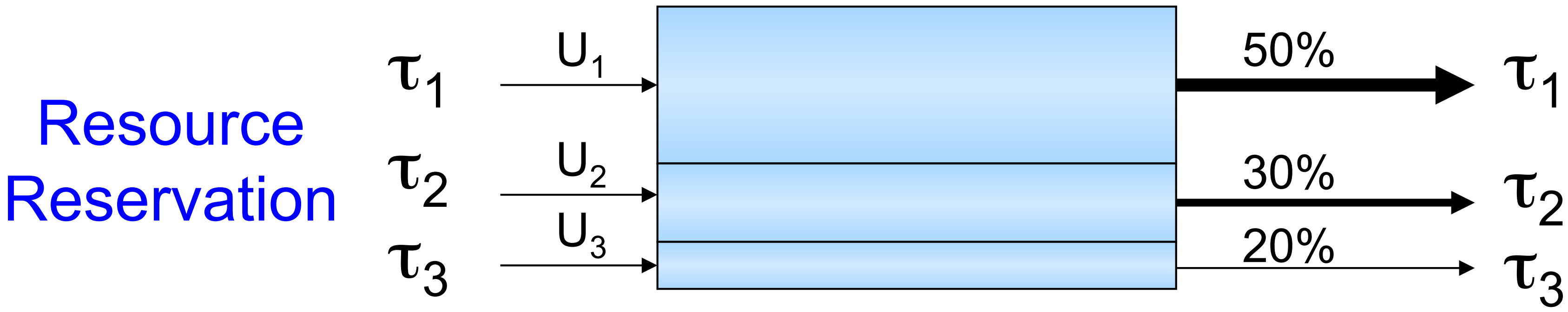
An isolated application executes as it were executing alone on a slower dedicated processor of speed s equal to the reserved fraction.

Advantages

- Predictability: A misbehavior of an application does not affect the others.
- Modular analysis: RT constraints can be verified independently of the knowledge of other applications.

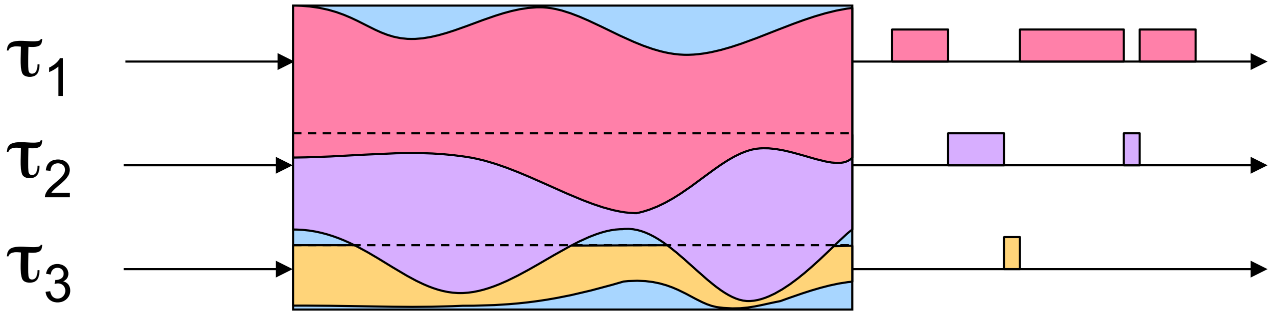
Priorities vs. Reservations



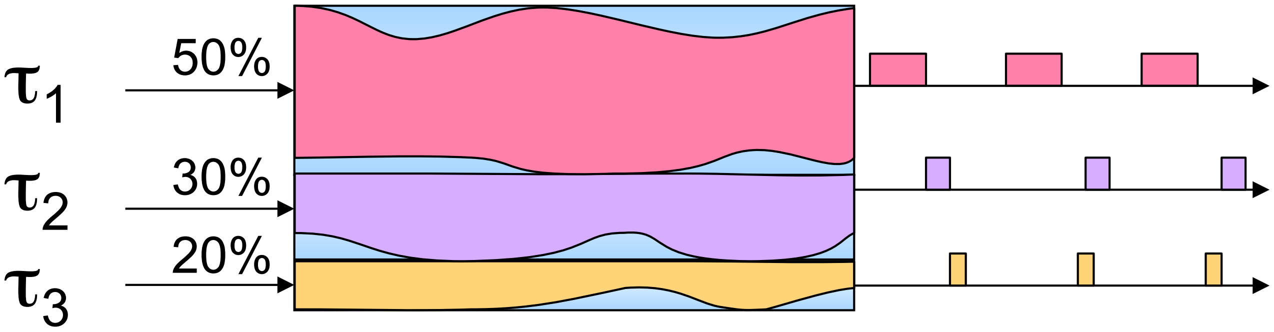


Priorities vs. Reservations

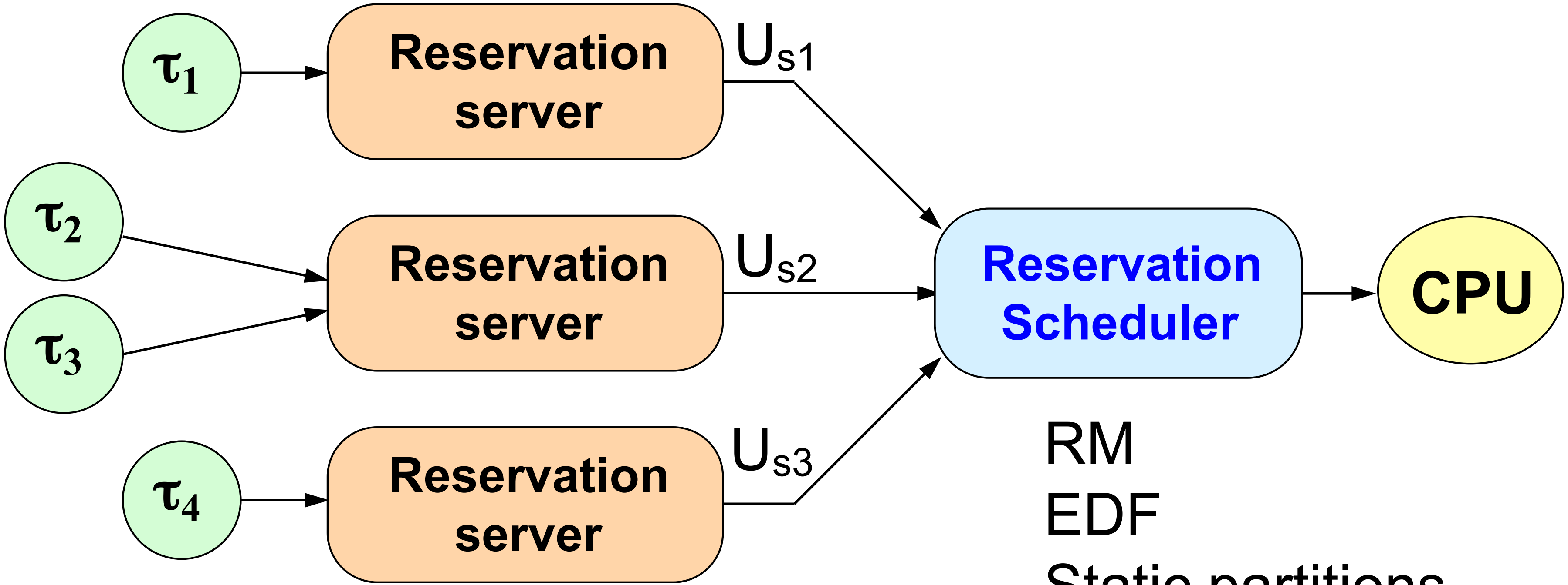
Prioritized Access



Resource Reservation



Implementing Resource Reservation

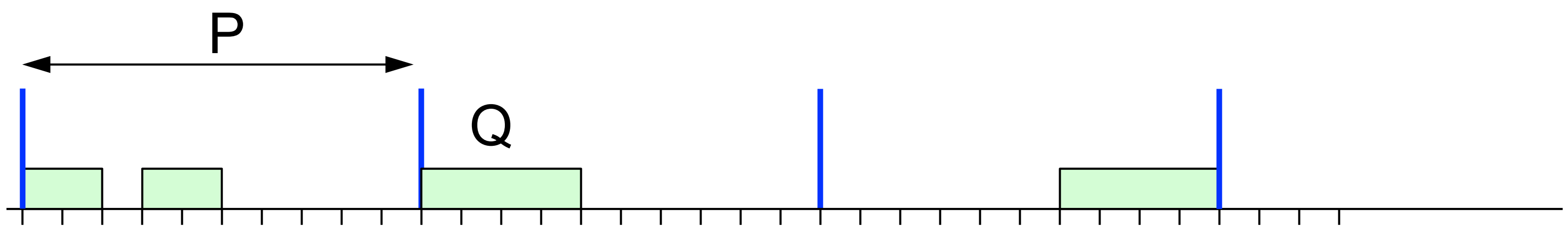


RM
EDF
Static partitions

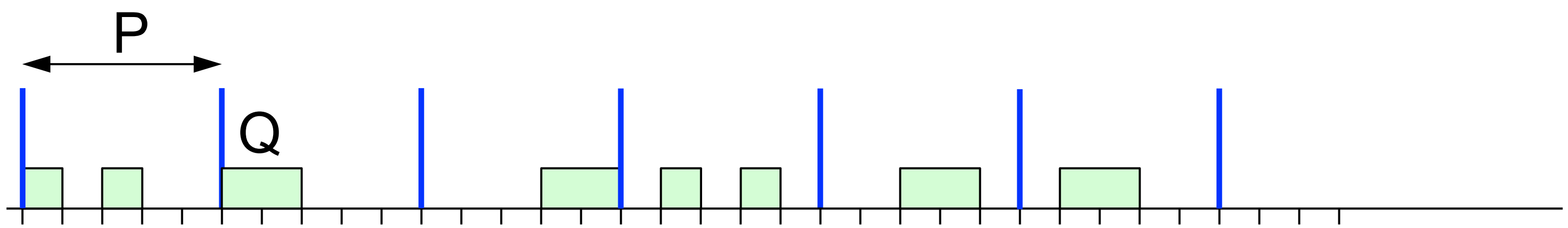
Polling
Deferrable Server
CBS

Reservation server

A way to implement a reservation is through a periodic server providing a budget **Q** every period **P**:



Reserved bandwidth: $\alpha = Q/P = 4/10$

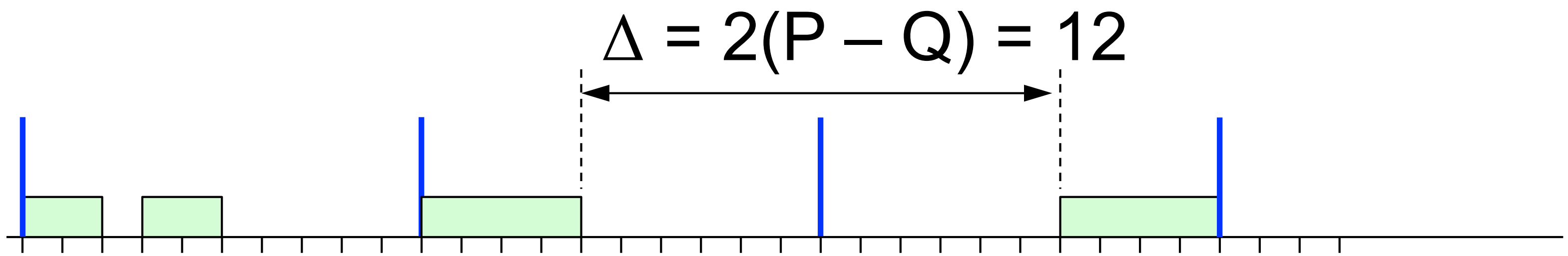


Reserved bandwidth: $\alpha = Q/P = 2/5$

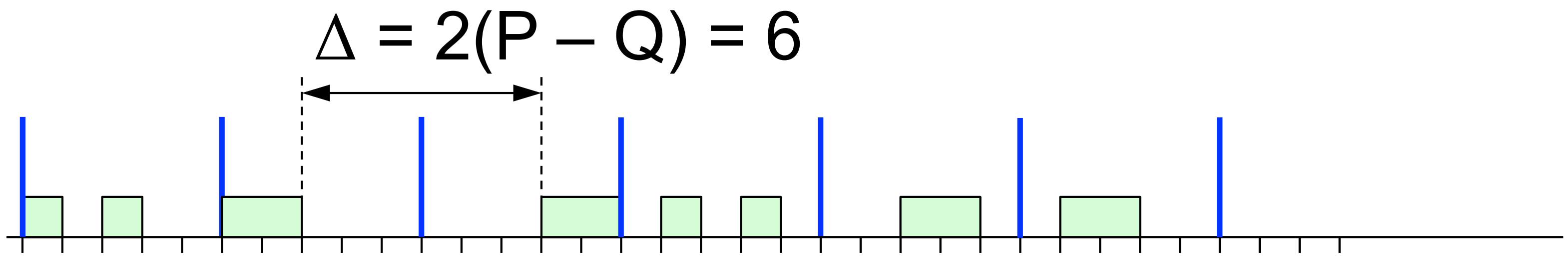
Which one is better?

Reservation server

Observe the worst-case delay:



Reserved bandwidth: $\alpha = Q/P = 4/10$

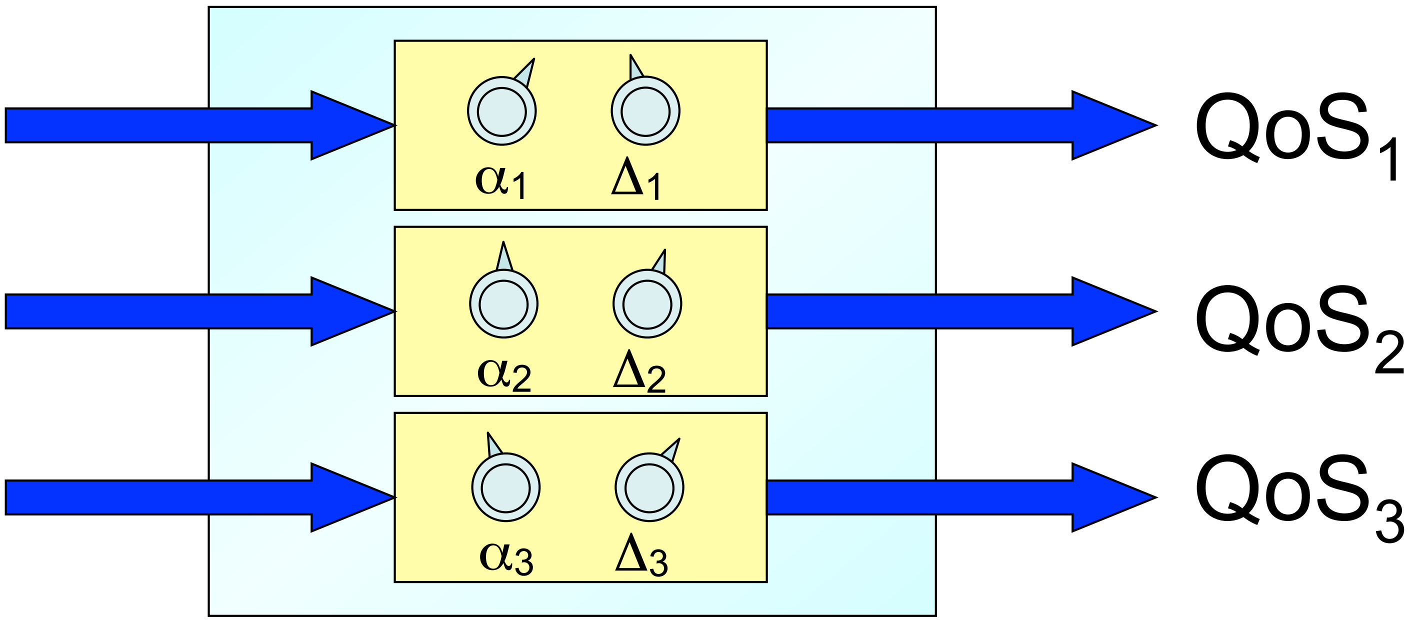


Reserved bandwidth: $\alpha = Q/P = 2/5$

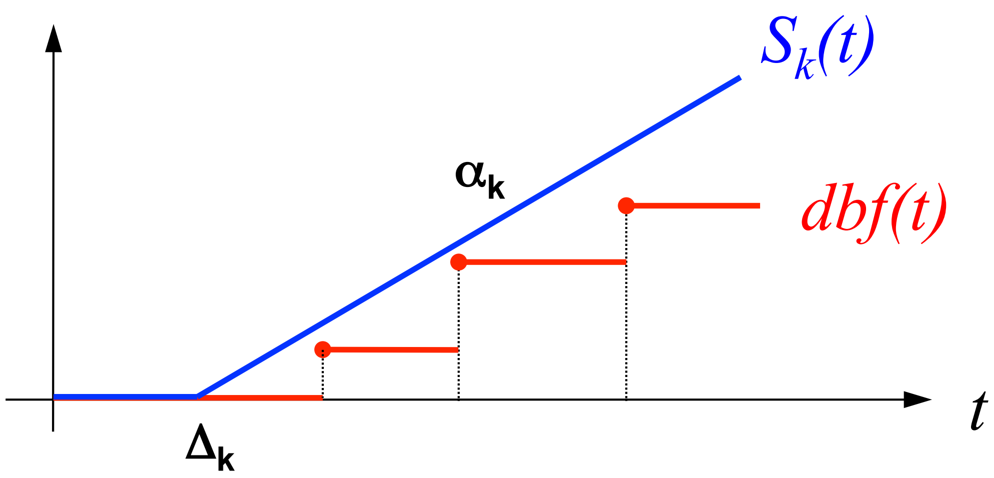
Abstracting Reservations

Hence, two key parameters to describe a reservation are:

- Bandwidth: α
- Worst-case delay: Δ



Real-Time guarantee



Supply function of a reservation

Demand Bound function of an Applⁿ

$$dbf(t) = \sum_{i=1}^{n} \left\lfloor \frac{t + T_i - D_i}{T_i} \right\rfloor C_i$$

Guarantee test

$$\forall t > 0 \quad dbf(t) \le S_k(t)$$

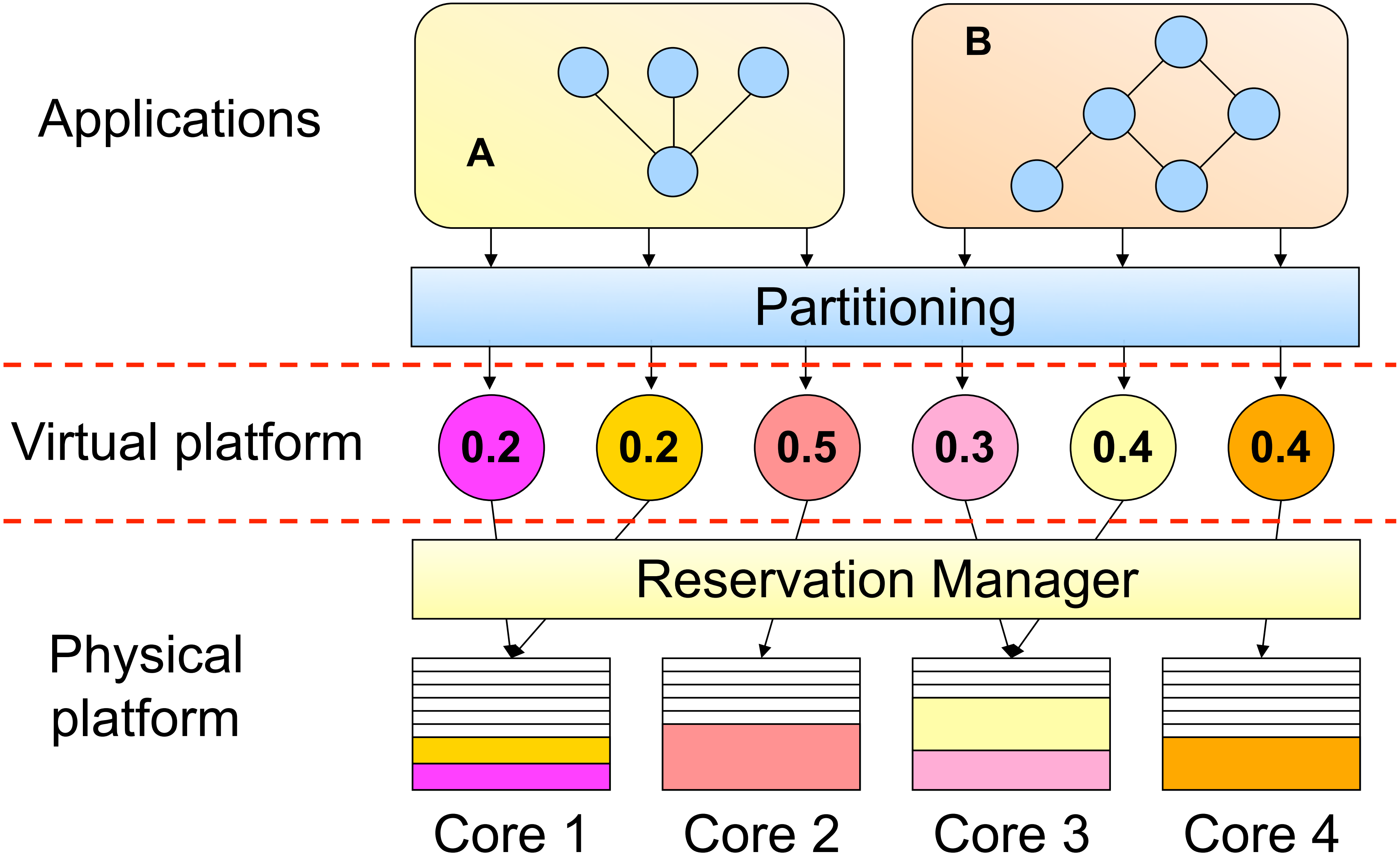
Multicore Reservations

- What is a reservation on a multi-core platform?
- Does it make sense to have a bandwidth $\alpha > 1$?

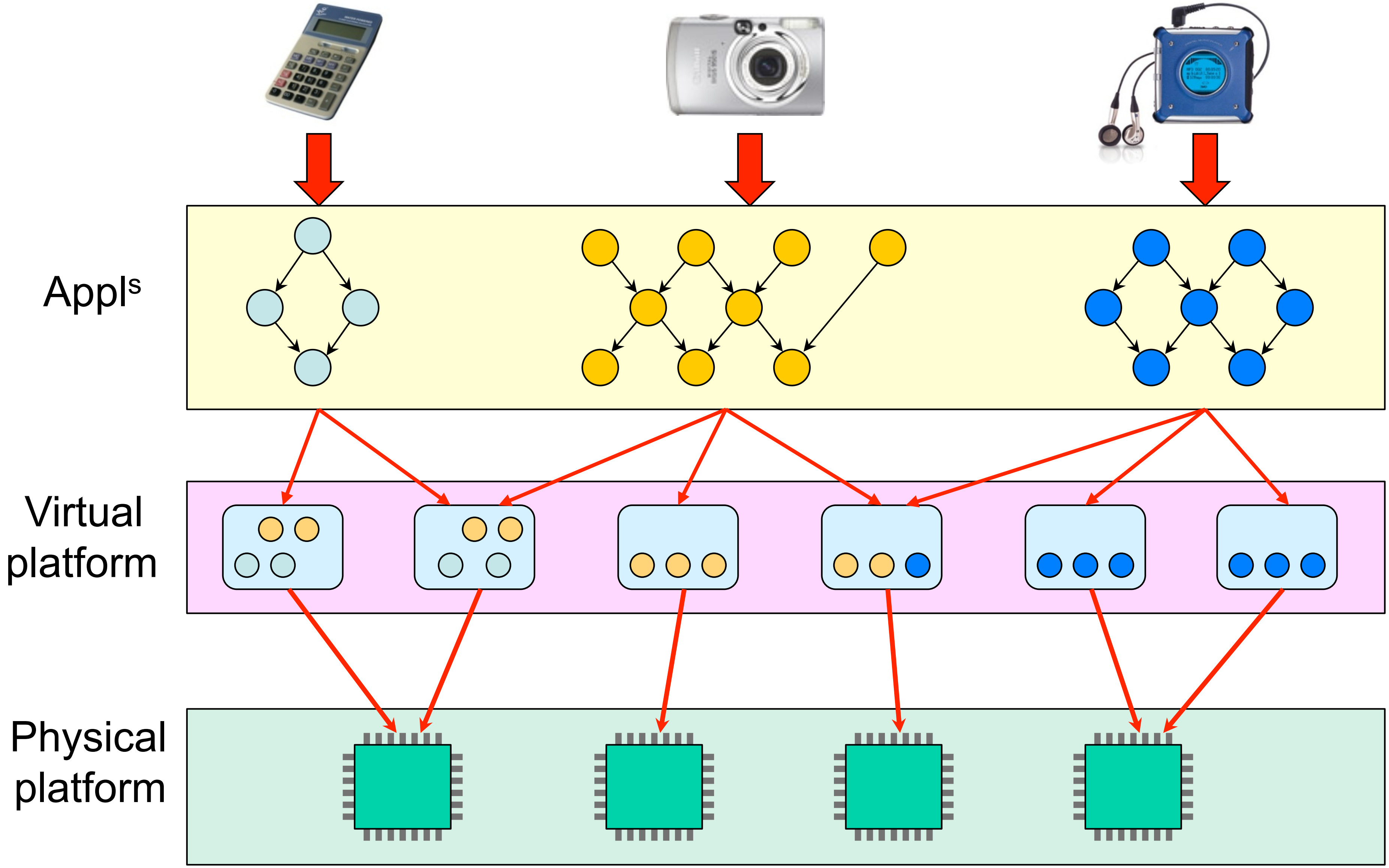
⇒ A multicore reservation cannot be specified by the overall supplied bandwidth

A multicore reservation must be specified as
a set of uniprocessor reservations

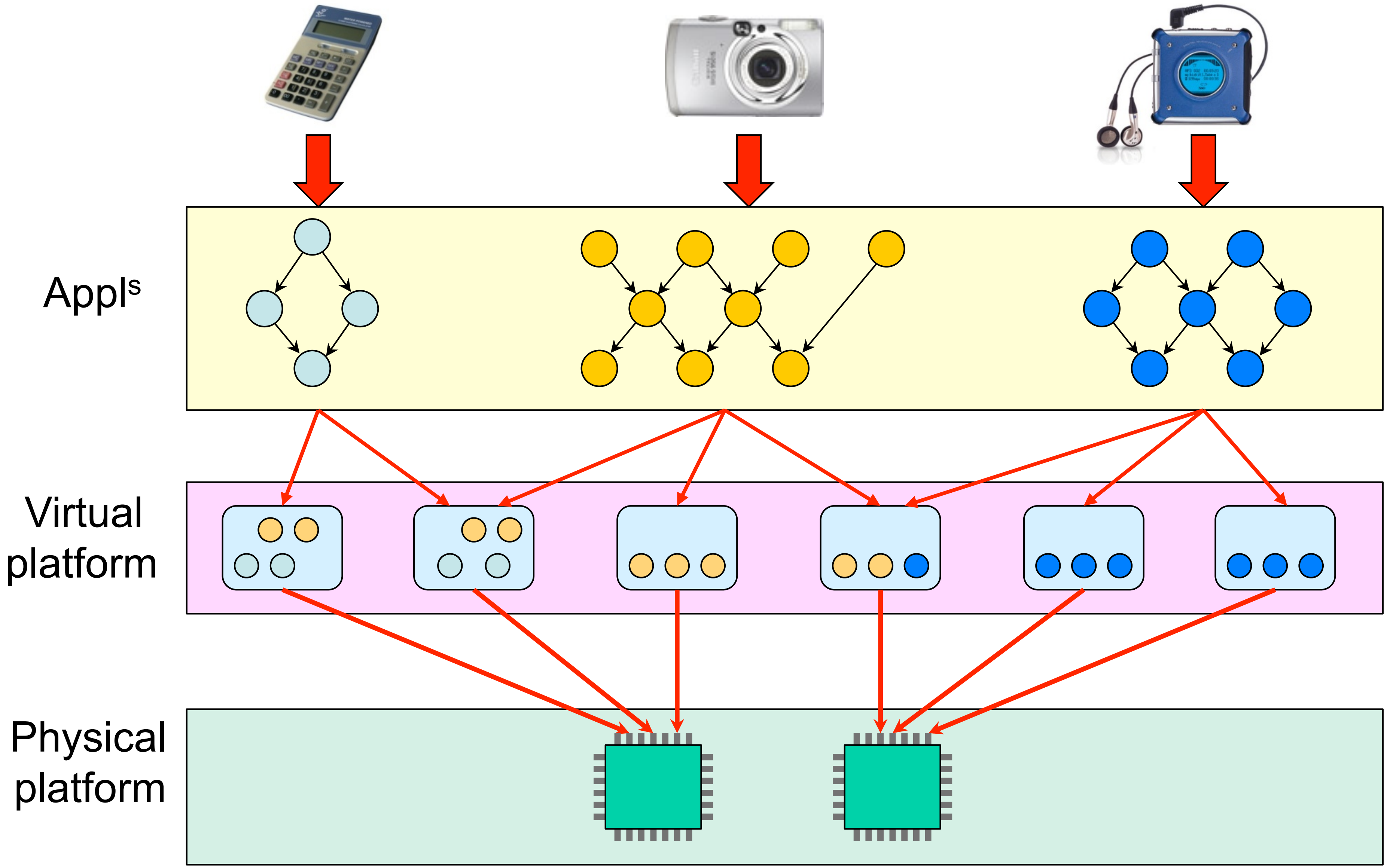
Abstracting the platform



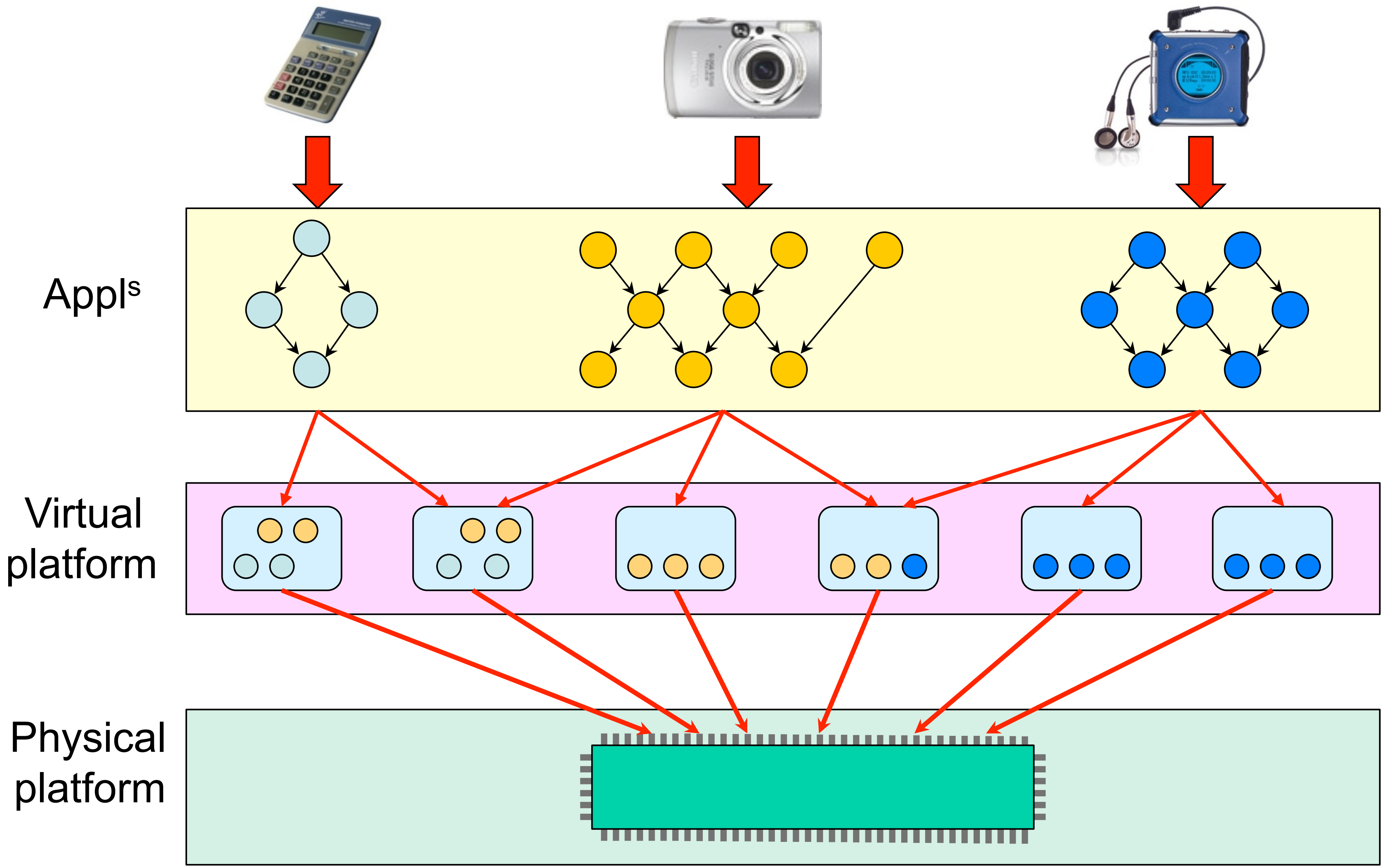
Abstracting the platform



Abstracting the platform



Abstracting the platform



Target RTOSes

ERIKA ENTERPRISE

EVIDENCE

http://erika.tuxfamily.org/

Linux EDF + **SCHED_DEADLINE**

http://www.evidence.eu.com/sched_deadline.html

- Small platforms (1-2 Kbytes)
- OSEK compliance

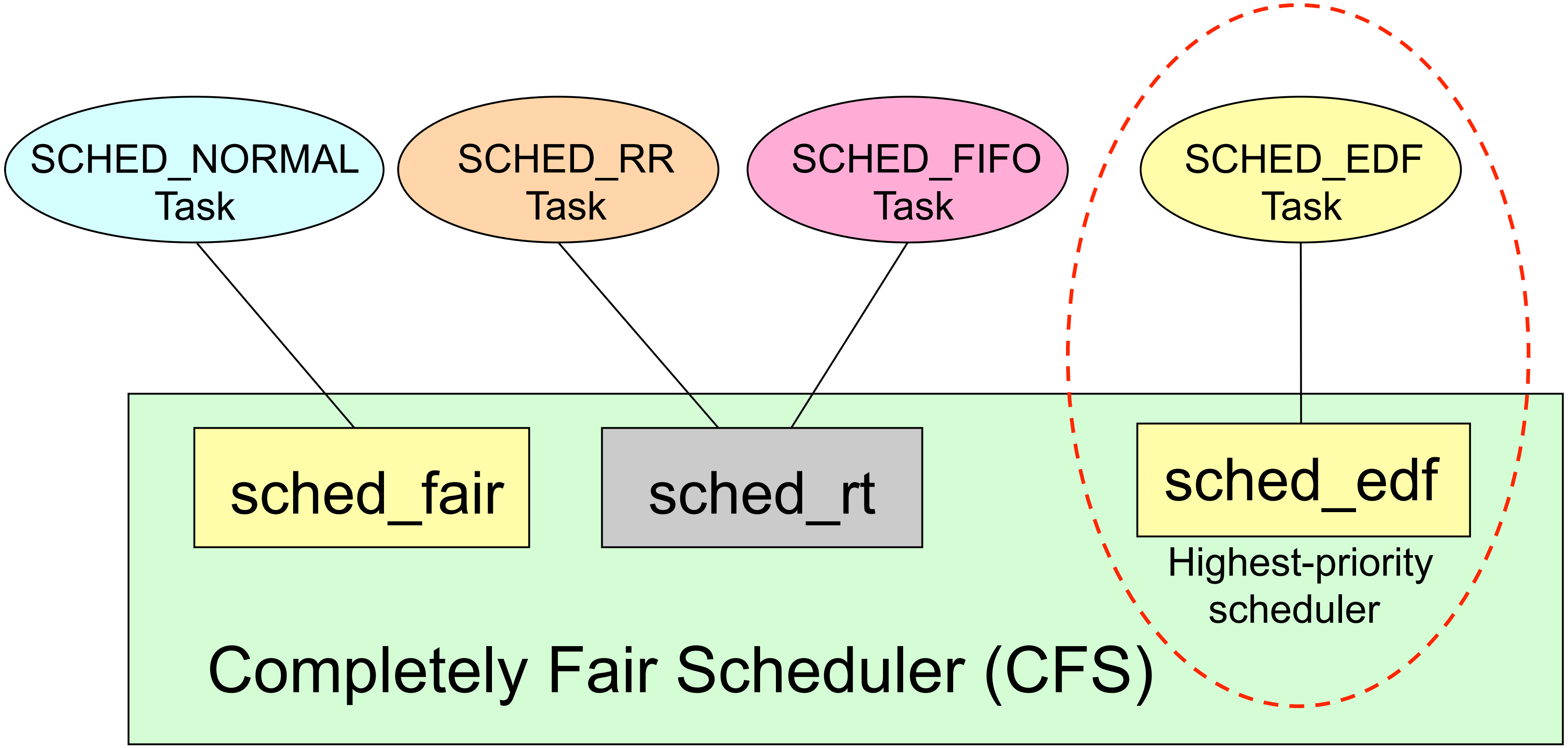
- PC-like platforms
- POSIX compliance

Both support

- Multi-core platforms
- Resource reservation
- Deadline-based scheduling

Integrating EDF in Linux

Deadline Scheduling on Linux (*Pisa, Evidence*)



Resource optimization

Partition the application into virtual cores in order to minimize

- the overall bandwidth
- the active number of processors
- the power consumption
- the maximum finishing time (makespan)

Bandwidth minimization

$$\begin{cases} \alpha = Q/P \\ \Delta = 2(P - Q) \end{cases}$$

Overhead: σ / P

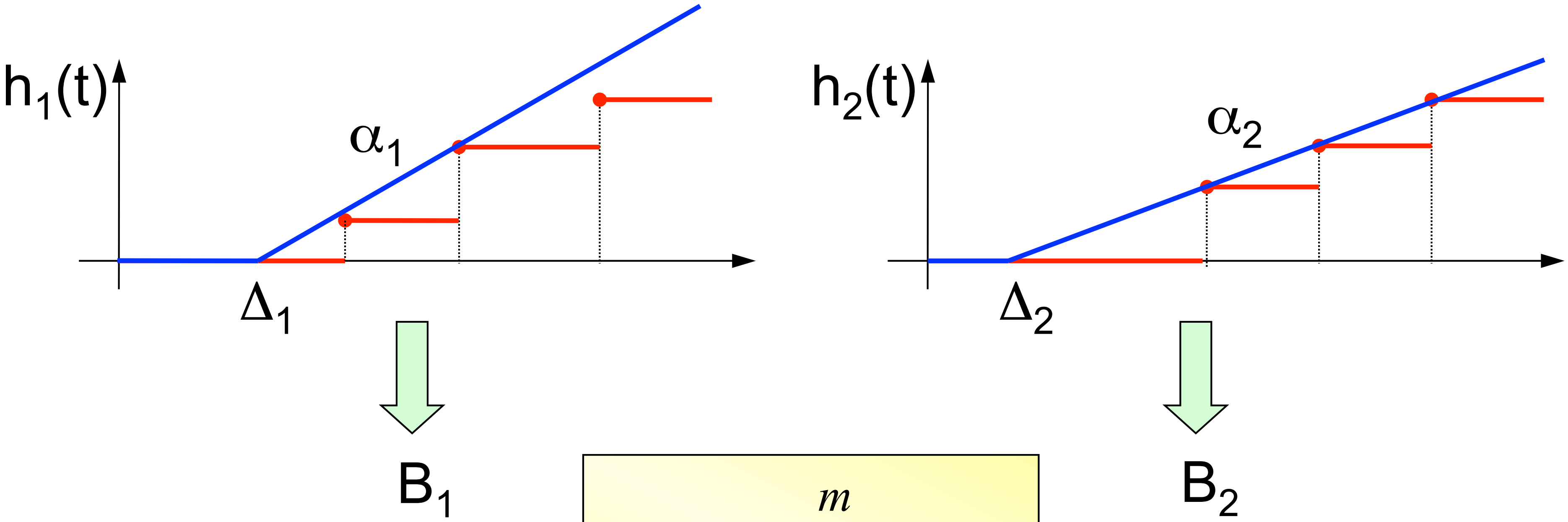
Actual Bandwidth: $B = \alpha + \sigma/P$

$$B = \alpha + 2\sigma \frac{1-\alpha}{\Delta}$$

Taking overhead into account, it is possible to compute the (α,Δ) that minimizes B.

Optimal bandwidth

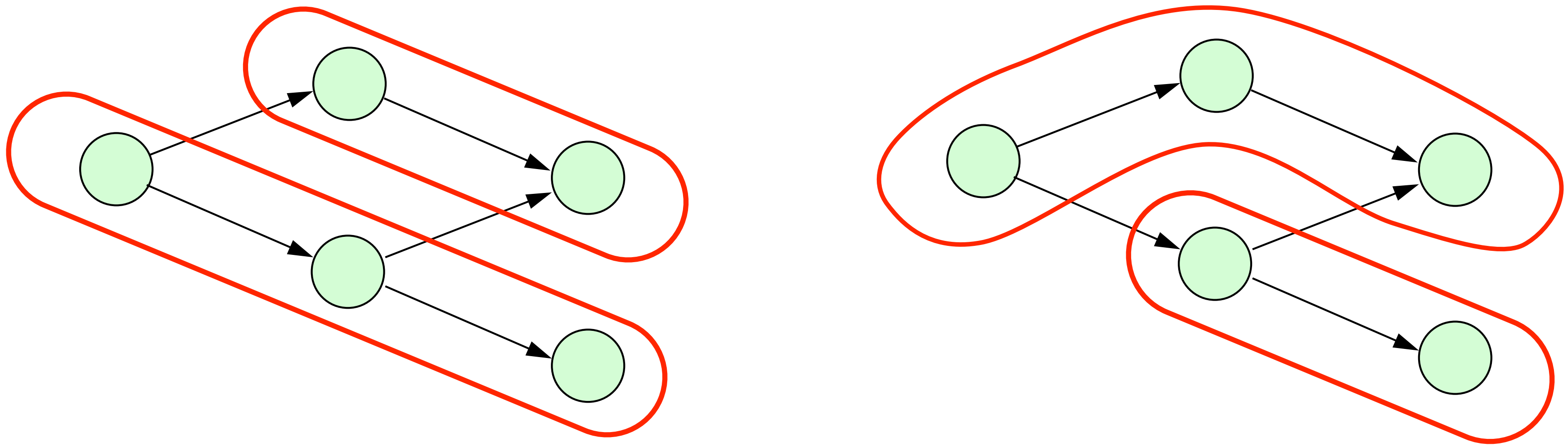
- Once the best (α,Δ) have been computed for each flow, the total bandwidth required by the application is:



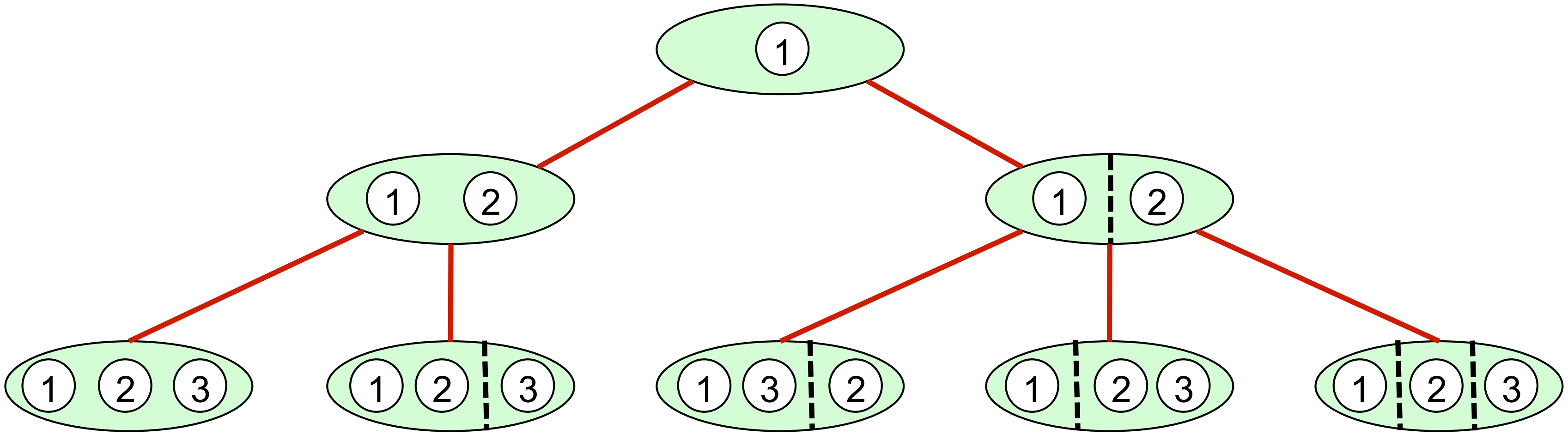
$$B = \sum_{k=1}^{m} B_k$$

Search for the best partition

- Different partitions require different bandwidth:



Complete vs. heuristic search



Pruning is used to cut

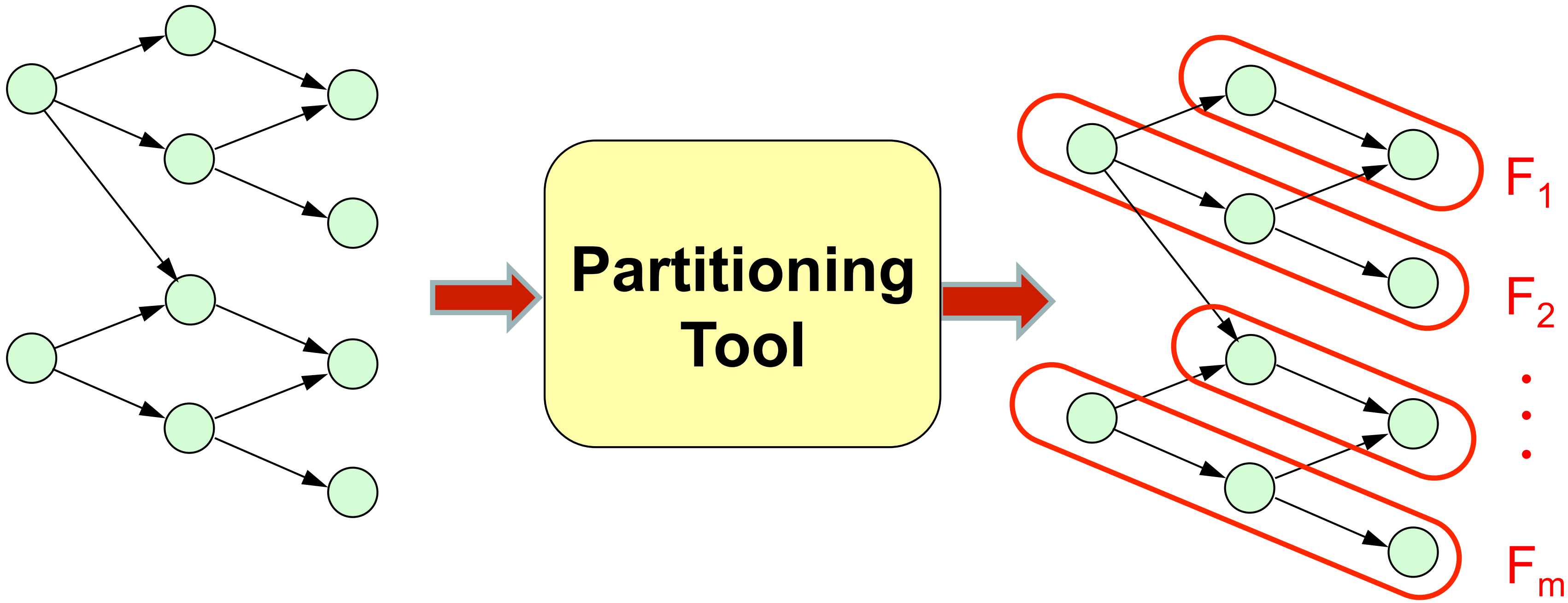
- unfeasible branches ($B_k > 1$)
- redundant branches ($m > M$)

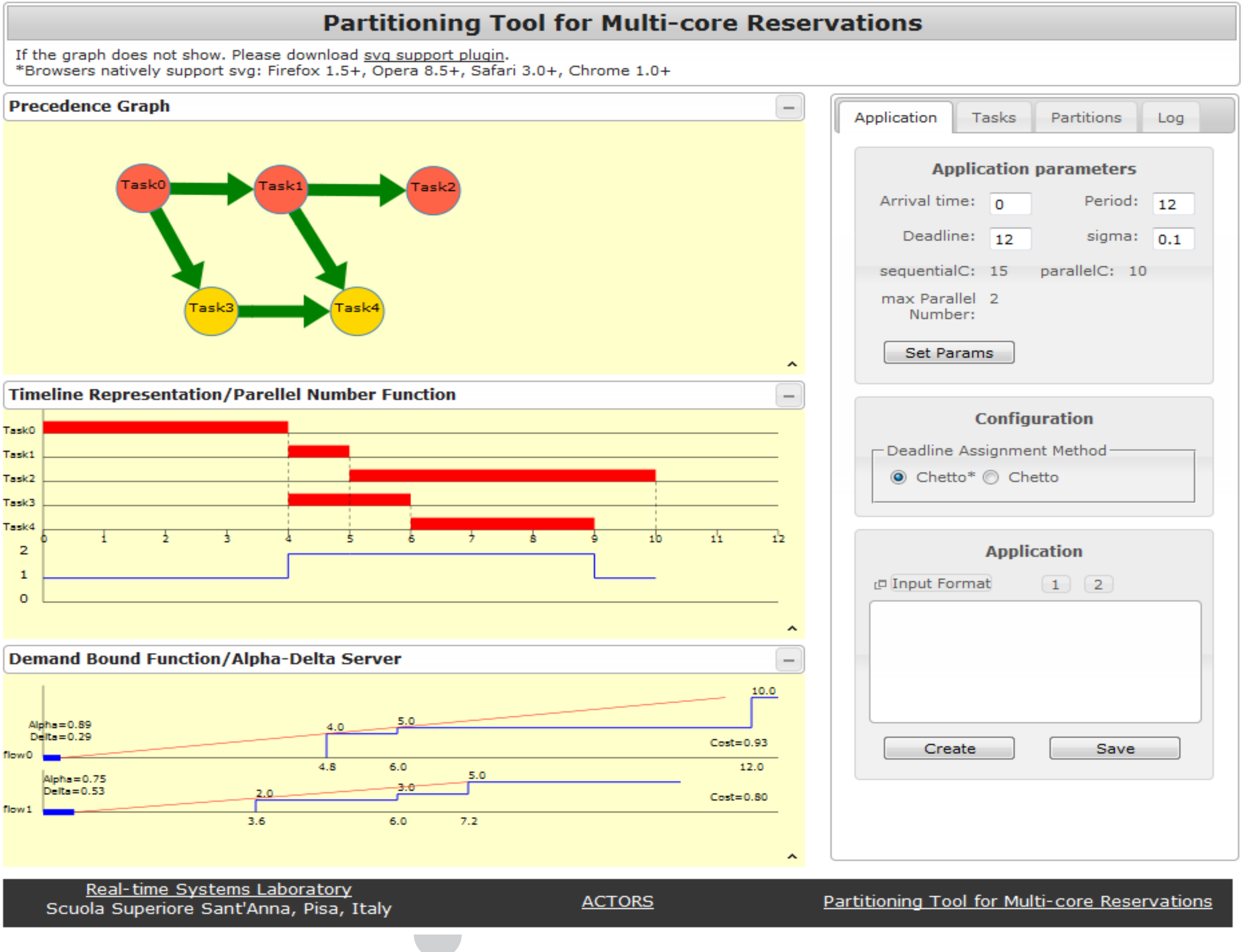
$$M = \left\lceil \delta \frac{C^s}{D} \right\rceil$$

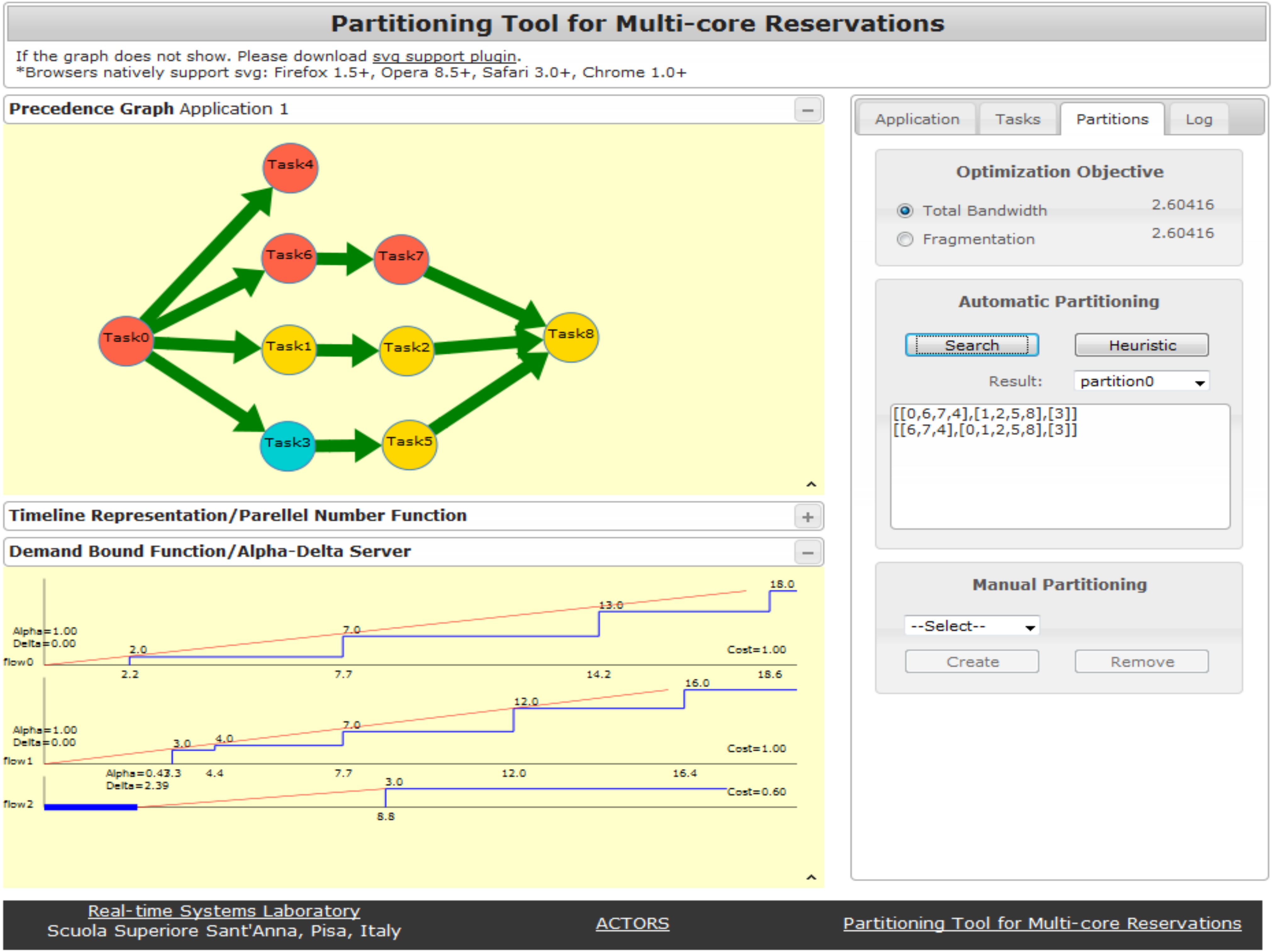
Exponential complexity (tractable for n < 20)

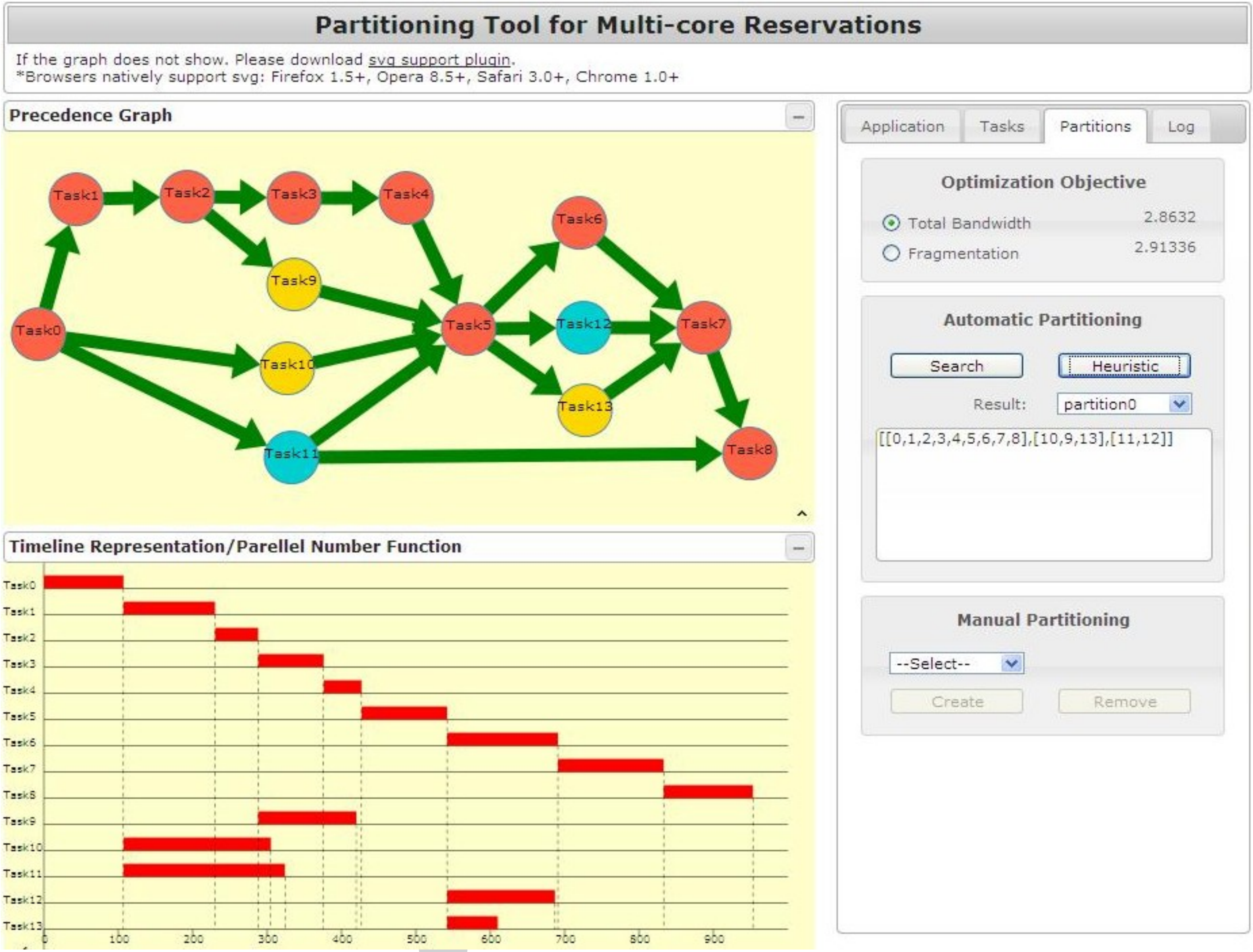
PartiCore: Partitioning tool for multi-core platforms

URL: http://particore.sssup.it/



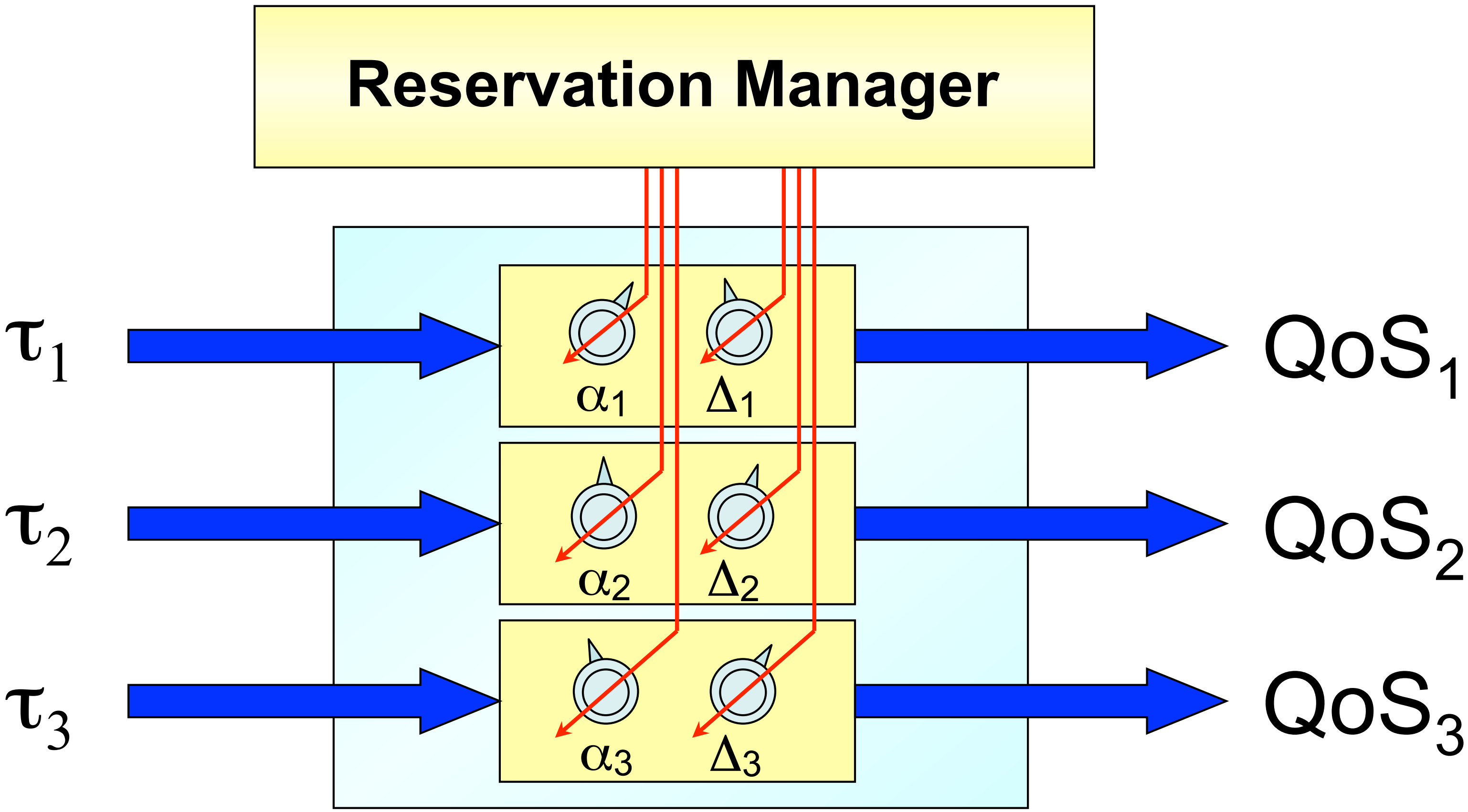

Partitioning Tool for Multi-core Reservations
If the graph does not show. Please download svg support plugin.
*Browsers natively support svg: Firefox 1.5+, Opera 8.5+, Safari 3.0+, Chrome 1.0+
Precedence Graph
Task0
Task1
Task2
Task3
Task4
Timeline Representation/Parellel Number Function
Task0
Task1
Task2
Task3
Task4
0 1 2 3 4 5 6 7 8 9 10 11 12
2
1
0
Demand Bound Function/Alpha-Delta Server
Alpha=0.89
Delta=0.29
flow0
4.0
5.0
10.0
Cost=0.93
4.8
6.0
12.0
Alpha=0.75
Delta=0.53
flow1
2.0
3.0
5.0
Cost=0.80
3.6
6.0
7.2
Application
Tasks
Partitions
Log
Application parameters
Arrival time: 0
Period: 12
Deadline: 12
sigma: 0.1
sequentialC: 15
parallelC: 10
max Parallel Number: 2
Set Params
Configuration
Deadline Assignment Method
Chetto*
Chetto
Application
Input Format
1
2
Create
Save
Real-time Systems Laboratory
Scuola Superiore Sant'Anna, Pisa, Italy
ACTORS
Partitioning Tool for Multi-core Reservations


Partitioning Tool for Multi-core Reservations
If the graph does not show. Please download svg support plugin.
*Browsers natively support svg: Firefox 1.5+, Opera 8.5+, Safari 3.0+, Chrome 1.0+
Precedence Graph Application 1
Task4
Task6
Task7
Task0
Task1
Task2
Task8
Task3
Task5
Timeline Representation/Parellel Number Function
Demand Bound Function/Alpha-Delta Server
Alpha=1.00
Delta=0.00
flow0
2.0
7.0
13.0
18.0
Cost=1.00
2.2
7.7
14.2
18.6
Alpha=1.00
Delta=0.00
flow1
3.0
4.0
7.0
12.0
16.0
Cost=1.00
3.3
4.4
7.7
12.0
16.4
Alpha=0.43
Delta=2.39
flow2
3.0
Cost=0.60
8.8
Application
Tasks
Partitions
Log
Optimization Objective
Total Bandwidth 2.60416
Fragmentation 2.60416
Automatic Partitioning
Search
Heuristic
Result: partition0
[[0,6,7,4],[1,2,5,8],[3]]
[[6,7,4],[0,1,2,5,8],[3]]
Manual Partitioning
--Select--
Create
Remove
Real-time Systems Laboratory
Scuola Superiore Sant'Anna, Pisa, Italy
ACTORS
Partitioning Tool for Multi-core Reservations


Partitioning Tool for Multi-core Reservations
If the graph does not show. Please download svg support plugin.
*Browsers natively support svg: Firefox 1.5+, Opera 8.5+, Safari 3.0+, Chrome 1.0+
Precedence Graph
Task0
Task1
Task2
Task3
Task4
Task5
Task6
Task7
Task8
Task9
Task10
Task11
Task12
Task13
Timeline Representation/Parellel Number Function
0
100
200
300
400
500
600
700
800
900
Application
Tasks
Partitions
Log
Optimization Objective
Total Bandwidth
2.8632
Fragmentation
2.91336
Automatic Partitioning
Search
Heuristic
Result:
partition0
[[0,1,2,3,4,5,6,7,8],[10,9,13],[11,12]]
Manual Partitioning
--Select--
Create
Remove

Adaptivity

- Achieved through a Reservation Manager that uses feedback scheduling to deal with dynamic changes:



Year 4 Review
Dresden, March 16th, 2012

Scheduling and Resource Management

Activity leader: Alan Burns
University of York
York, UK

Year 4 Review
Dresden, March 16th, 2012

Real-Time Networks

Activity leader: *Luis Almeida*
University of Porto
Portugal