Why real-time scheduling theory still matters

Sanjoy Baruah

The University of North Carolina at Chapel Hill

Our discipline = Systems + Theory

is about systems
…that require formal/ theoretical analysis

Has over-emphasized the theory

- A distinguishing characteristic of the discipline

Is starting to remedy this

-The "only theory?" test
-Hard to get a uniprocessor paper into ECRTS/ RTSS!
-Special issues/ invited talks are systems-oriented

Let's not over-compensate

Scheduling theory remains relevant to real-time systems.

Scheduling theory remains relevant to real-time systems. Some forms of scheduling theory are particularly important.

Scheduling theory remains relevant to real-time systems. Some forms of scheduling theory are particularly important. These important forms can be identified.

Thesis

Scheduling theory remains relevant to real-time systems. Some forms of scheduling theory are particularly important. These important forms can be identified.

Outline of presentation

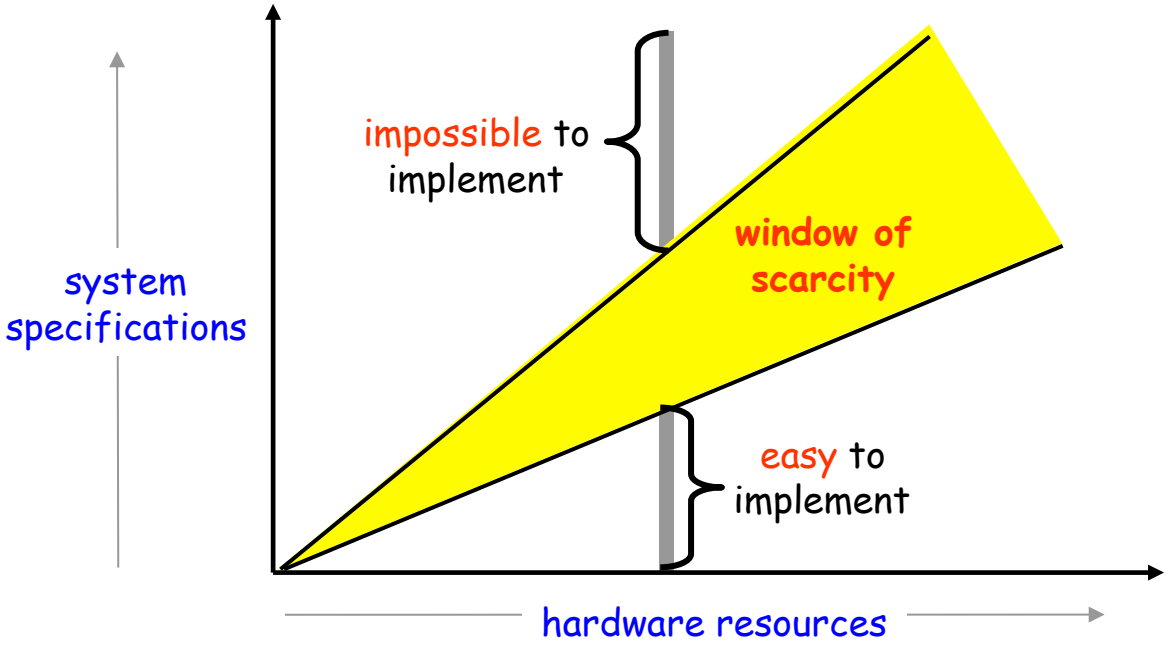
1. Why theory mattered (and matters)
2. What forms of scheduling theory are important
3. What specific areas are most important

Why theory?

Our discipline = Systems + Theory

The window of scarcity

Real-time systems often have resource constraints



Why theory?

Our discipline = Systems + Theory

The window of scarcity

Real-time systems often have resource constraints

Often safety-critical

- Very high cost of error
- Must be validated correct by Certification Authorities

Use theory to prove correctness

Why theory?

The window of scarcity

Often safety-critical

What counts as good theory?

Should potentially be useful

The "pure" sciences legitimately take the discovery of facts and laws as an end in itself. A new fact, a new law is an accomplishment, worthy of publication. But in [computer science and engineering] novelty in itself has no merit. We test our artifacts by their

"Industrially relevant research is … research that is usable when industry decides it is relevant"

-Bjorn Andersson

1. new facts and laws are needed

   - but present them elsewhere (e.g., theory conferences)

2. need not be useful immediately

What counts as good theory?

In that empire, the art of cartography attained such perfection that […] the cartographers guilds struck a map of the empire whose size was that of the empire, and which coincided point for point with it.

In the deserts of the west, still today, there are tattered ruins of that map, inhabited by animals and beggars.

-Jorge Luis Borges

systems

dels

* Using the appropriate abstractions
  - highlights a few salient features or principles
* Computationally tractable abstractions
  - E.g., for hard-real-time schedulability analysis

Obtaining appropriate abstractions is an important research area

Promising research areas

Multiprocessors

Component-based design

Mixed criticalities

Promising research areas

Multiprocessors

* Future RT systems will be multiprocessor ones
  -the multicore revolution

Component-based design

* Multiprocessor scheduling theory is not mature enough

Mixed criticalities

* Important questions
  - models
  - metrics
  - scheduling algorithms

Big-picture question: what critical insights are needed for multiproc. scheduling?

Promising research areas

Multiprocessors

Component-based design

Mixed criticalities

Open systems and componentization

- more powerful platforms
- SWaP considerations
- software engineering issues

Size, Weight, and Power (i.e., Energy)

Some work has been done… but much remains

- abstraction and interface specification
- an algebra for composition

Mixed criticalities: Promising research areas

Multiprocessors

Component-based design

Mixed criticalities

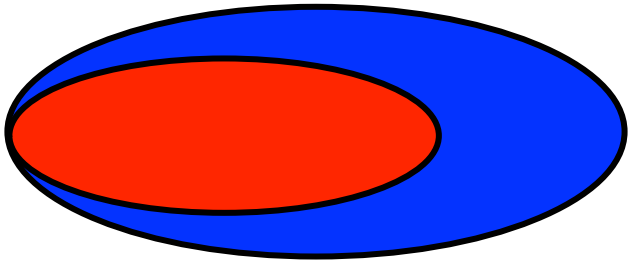
Mixed criticalities: Promising research areas

Some sub-systems are more important than others

- Automotive example: ABS vs car stereo

Different sub-systems have different certification requirements

- Defense avionics example. Flight-critical and mission-critical functionalities



Flight critical: certified by Certification Authorities

Mission-critical: validated by design team

Example: Determining worst-case execution time (WCET)

- flight-critical certification: cycle-counting under pessimistic assumptions
- mission-critical validation: extensive experimentation

Mixed criticalities: Promising research areas

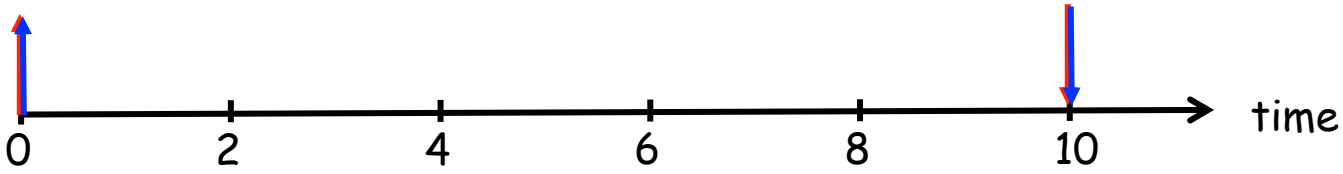
J_1 is flight-critical; J_2 is mission-critical
Both arrive at t=0; have deadlines at t=10
WCET of J_1 is 6; WCET of J_2 is 5

$6 + 5 > 10 \Rightarrow$ not schedulable

But…

- flight-criticality certification does not need J_2 to meet its deadline
- for mission-critical validation, J_1's WCET of 6 may be too pessimistic
  * Suppose J_1's WCET, obtained by extensive experimentation, is 5

Priority-based scheduling: $J_1 > J_2$



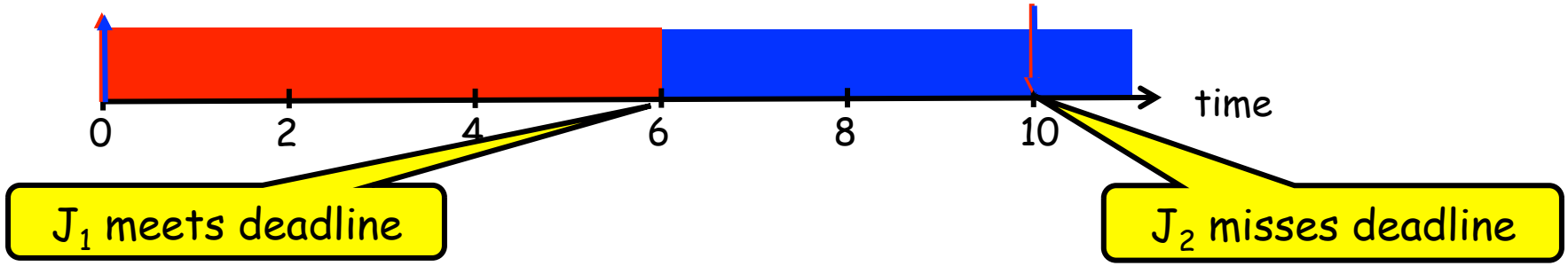
Mixed criticalities: Promising research areas

J_1 is flight-critical; J_2 is mission-critical
Both arrive at t=0; have deadlines at t=10
WCET of J_1 is 6; WCET of J_2 is 5

But…
- flight-criticality certification does not need J_2 to meet its deadline
- for mission-critical validation, J_1's WCET of 6 may be too pessimistic
  * Suppose J_1's WCET, obtained by extensive experimentation, is 5

Priority-based scheduling: $J_1 > J_2$

Flight-criticality certification



Mixed criticalities: Promising research areas

J_1 is flight-critical; J_2 is mission-critical
Both arrive at t=0; have deadlines at t=10
WCET of J_1 is 6; WCET of J_2 is 5

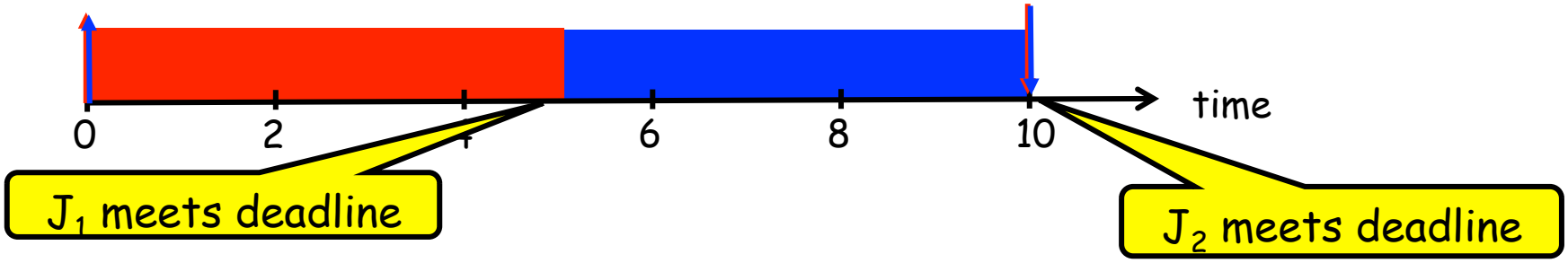
Validated at both criticalities

But…

- flight-criticality certification does not need J_2 to meet its deadline
- for mission-critical validation, J_1's WCET of 6 may be too pessimistic
  - * Suppose J_1's WCET, obtained by extensive experimentation, is 5

Priority-based scheduling: $J_1 > J_2$

Mission-critical validation



Mixed criticalities: Promising research areas

J_1 is flight-critical; J_2 is mission-critical
Both arrive at t=0; have deadlines at t=10
WCET of J_1 is 6; WCET of J_2 is 5

Validated at both criticalities

The same system is being validated, twice

Flight-critical certification	Mission-critical validation
at a very high level of assurance	at a lower level of assurance
of only a subset of the system	of the entire system

Interesting open issues:

- How do we represent MC systems?
- How do we reason about them?
  - "parallel worlds"? space-time partitioning?
- What scheduling strategies are suitable?

Summary

A thesis…

Scheduling theory remains relevant to real-time systems.

…and its justification

- The window of scarcity; safety-critical nature of applications

Summary

A thesis…

Scheduling theory remains relevant to real-time systems. Some forms of scheduling theory are particularly important.

…and its justification

- The window of scarcity; safety-critical nature of applications
- Must be relevant; must be at appropriate level of abstraction

Summary

A thesis…

Scheduling theory remains relevant to real-time systems. Some forms of scheduling theory are particularly important. These important forms can be identified.

…and its justification

- The window of scarcity; safety-critical nature of applications
- Must be relevant; must be at appropriate level of abstraction
- Multiprocessors; component-based design; and mixed criticalities

REAL-TIME SYSTEMS

SCHEDULING THEORY

Summary

A thesis…

Scheduling theory remains relevant to real-time systems. Some forms of scheduling theory are particularly important. These important forms can be identified.

…and its justification

- The window of scarcity; safety-critical nature of applications
- Must be relevant; must be at appropriate level of abstraction
- Multiprocessors; component-based design; and mixed criticalities