

CRM: Middleware for Managing CPU Time

John Regehr Jay Lepreau
Flux Group
University of Utah

Oct. 5, 2001

1

Outline of Talk

- ◆ Context
- ◆ Motivation
- ◆ CRM
 - Structure
 - Converting Guarantees
 - Enforcing Rules
 - Other Functionality
- ◆ Related work
- ◆ Conclusions

2

Context

- ◆ Focus on
 - Single-node
 - CPU time as critical resource
- ◆ Open system model
 - Independently developed apps running concurrently
 - Requires enforcement
 - e.g. CPU reservations or proportional share

3

Motivation

- ◆ OS support for multimedia not used much in practice
- ◆ Why?
 - Users would rather run just one app at a time?
 - Field is immature?
 - Bill Gates, Linus Torvalds, and Steve Jobs don't care?
 - Usability issues?

4

Our answer: Usability Issues!

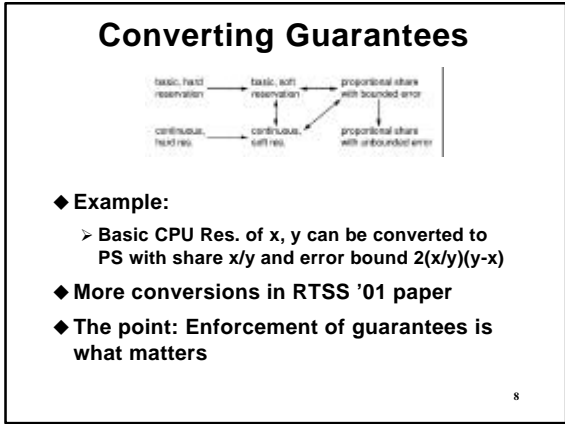
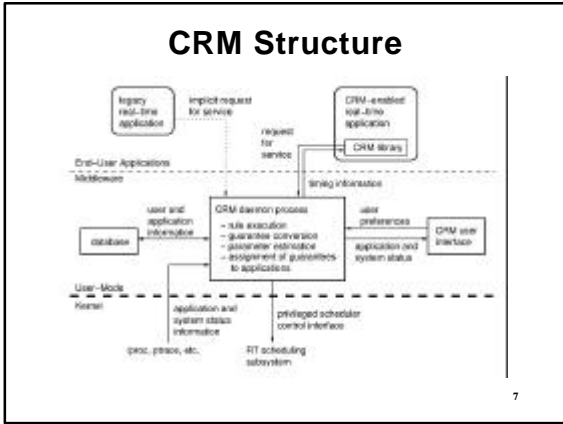
- ◆ For end-users and developers

5

How can middleware help?

- ◆ Store user preferences
- ◆ Interact with user (minimally)
- ◆ Determine app. requirements
- ◆ Support legacy apps.
- ◆ Hide differences between schedulers

6



- ### Enforcing Rules
- ◆ Entities are: resource principals, guarantees, requests, rules and events
 - ◆ Events cause rules to be evaluated
 - ◆ Examples:
 - Enforcing fairness between users
 - Running feasible set of apps. with highest value
 - Selecting mode for adaptive app.

- ### Other Functionality
- ◆ Determine application requirements
 - Store them
 - Apply them to legacy apps
 - ◆ GUI
 - ◆ These are hard!

- ### Related Work
- ◆ QoS Broker (Nahrstedt and Smith '95)
 - ◆ Rialto resource manager (Jones et al. '95)
 - ◆ Adaptive resource manager (Oparah '99)
 - ◆ RT-CORBA

- ### Status
- ◆ Not implemented (yet)
 - ◆ Targets:
 - Linux/RT (TimeSys)
 - QLinux (UMass)
 - Linux-SRT (AT&T Research)
 - Linux + HLS (Utah)

Conclusions

- ◆ Hacking the scheduler is not enough
- ◆ Middleware can help solve remaining problems

13

The End

- ◆ More info and papers here:
<http://www.cs.utah.edu/~regehr>
- ◆ Let's talk...

14