Augmented CPU Reservations

John Regehr John A. Stankovic
University of Virginia

May 31, 2001

1

Outline of Talk

- Background
  - Open real-time systems
  - Rez and HLS
- Stolen time
- Rez-C and Rez-FB
  - Design
  - Performance
- More stolen time data
- Related work
- Conclusions

2

Background: Soft Real-Time in an Open System

- Goal: Coexisting, independently developed real-time applications
  - Digital video and audio, voice recognition, vision, soft modem, games, etc.
- A solution: add CPU reservations to general-purpose OS
  - Applications scheduled at specified rate and granularity
  - E.g. 1 ms / 7.5 ms, 15 ms / 250 ms

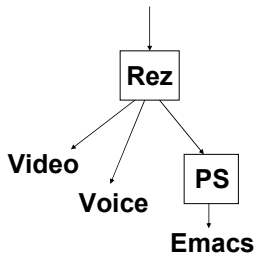
3

Rez: A Reservation Scheduler

- Algorithm:
  - EDF
  - Budgets
- Implementation:
  - In Windows 2000 kernel
  - Uses HLS hierarchical scheduler infrastructure
  - 400 lines of C

4

HLS Example



Schedulers communicate using virtual processors

HLS = Windows 2000 + 3100 lines of C

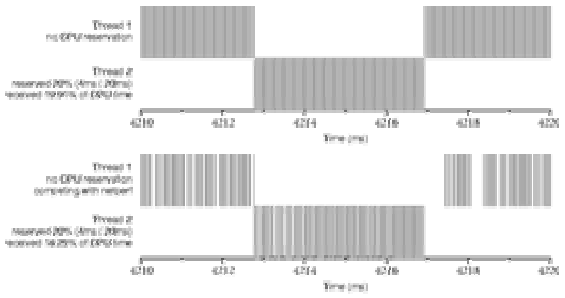
5

A Problem: Stolen Time

- OS may steal CPU time from applications, causing missed deadlines
- Stolen time sources:
  - DPCs in Windows NT / 2000
  - Bottom half handlers in Unix
- Stolen time mechanisms: high priority, not preemptible, not accounted for

6

Time Stolen by Network Receive Processing



Stolen Time Solutions

- Move CPU-intensive tasks into threads
- Make stolen time mechanisms preemptible
- Account for worst-case amount of stolen time
- Augmented CPU reservations

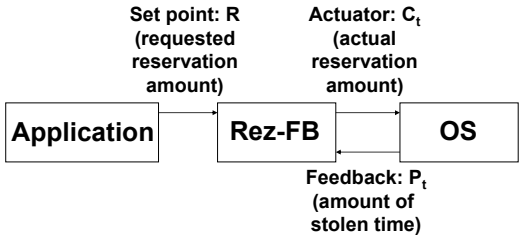
8

Augmented Reservations

- Strategy: accurately measure stolen time
  - Instrument Windows 2000 dispatch interrupt handler
- Rez-C: avoid deducting stolen time from budgets
- Rez-FB: feedback control
  - Goal: actual CPU time == requested CPU time

9

Rez-FB



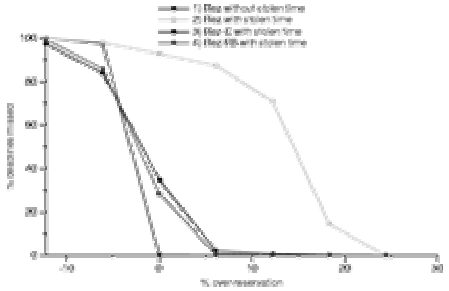
Feedback equation:

$$C_{t+1} = C_t + G(R-P_t)$$

Evaluated each period for each reservation

10

Augmented Reservation Performance



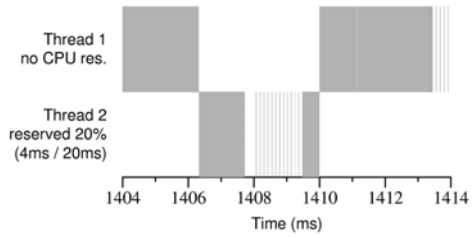
11

More Stolen Time Data

- Test machine: 500 MHz PIII
- Receive processing for 100 Mbps Ethernet:
  - More than 20% of reservation in Linux and Windows 2000
- Software modem:
  - 9.9% in Windows 2000
- USB 1.1
  - 5.7% in Windows 2000
- USB 2.0, Firewire
  - ??

12

Time Stolen by Disk Driver



- 49% of reservation stolen by Linux IDE disk driver in default mode (PIO)

13

Related Work

- Moving code into scheduled contexts
  - Soft modems [Jones and Saroiu 01]
- Scheduling bottom-half activity
  - Mach [Rashid et al. 89]
  - Nemesis [Leslie et al. 96]
  - FreeBSD [Jeffay et al. 98]
- Including stolen time in schedulability analysis
  - Accounting for interrupt costs [Jeffay and Stone 93]
- Feedback-based scheduling
  - FC-EDF [Lu et al. 99]

14

Conclusion

- Stolen time is a serious problem
  - Experiments show up to 50% of CPU being stolen
  - OSs have hundreds of drivers, many of which may steal time
- Augmented CPU reservations:
  - Simple and non-intrusive
  - Increase application scheduling predictability during stolen time

15

The End

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

16

Augmented Reservation Contributions

- Rez-C and Rez-FB
  - 6% over-reservation to eliminate most deadline misses due to network traffic
  - vs. 24% over-reservation for plain Rez
- Quantified severity of stolen time
  - Windows 2000 + Rez and Linux/RT
  - Network, disk, software modem, USB

17

OS Design Rule

- Mechanisms that are invoked often must be lightweight
  - Interrupts
    - Highest priority
    - Fixed-priority scheduler
  - DPCs, bottom-half handlers
    - Medium priority
    - FIFO scheduler
  - Threads
    - Lowest priority
    - Time-sharing scheduler

18