Resource-Efficient Real-Time Scheduling Using Credit-Controlled Static-Priority Arbitration

Firew Siyoum, Benny Akesson, Sander Stuijk, Kees Goossens, Henk Corporaal

Eindhoven University of Technology,
The Netherlands

RTCSA-2011, Toyama, Japan

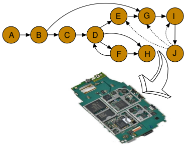
August 31, 2011

Overview

Embedded Systems

- run multiple applications with real-time requirements such as throughput and latency.

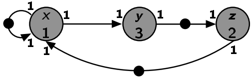
Dataflow Modeling



- For design-time analyzability, application models, such as dataflow techniques, are used.

Homogeneous Synchronous Dataflow (HSDF)

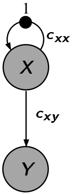
- Actors connected through channels.

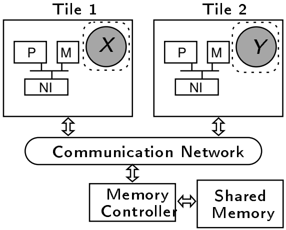


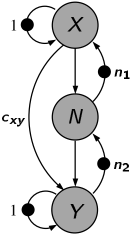
Overview

Multiprocessor System-on-Chip (MPSoc)

- architecture-aware model captures various system aspects.

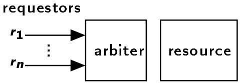






Resource sharing

- MPSoC resources are shared to reduce cost.
- Service guarantee is the minimum service an arbiter guarantees each requestor.

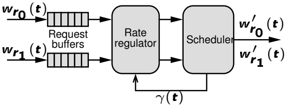


Technische Universiteit Eindhoven University of Technology

Overview

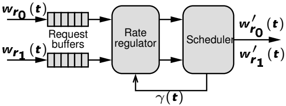
Credit-Controlled Static-Priority (CCSP) Arbiter

- consists of a rate regulator and scheduler.
- schedules the highest-priority eligible requestor.
- the rate-regulator guarantees an allocated fraction of the resource.



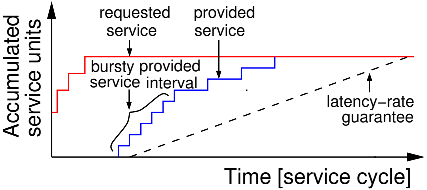
Credit-Controlled Static-Priority (CCSP) Arbiter

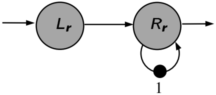
- consists of a rate regulator and scheduler.
- schedules the highest-priority eligible requestor.
- the rate-regulator guarantees an allocated fraction of the resource.



Latency-rate ($\mathcal{LR}$) Service Guarantee

- is a linear model on the provided service.

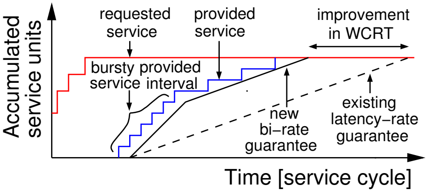


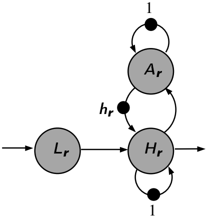


Overview

Contributions

- A piece-wise linear service guarantee and its dataflow model.





Implications

- A given MPSoC resource can support more requestors, or
- Accommodate a given set of requestors with less resource capacity.
- Experimental result: savings from 26%-67% in memory bandwidth.

Outline

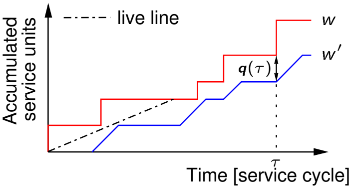
The Formal Model

Service allocation for $r \in R$

- Abstract resource view: service unit and service cycle
- allocated service is the pair (burstiness, rate)$= (\sigma'_r, \rho'_r) \in \mathbb{R}^+ \times \mathbb{R}^+$.
- $\sum_{\forall r \in R} \rho'_r \leq 1$ and $\sigma'_r \geq 1$.

Service curves

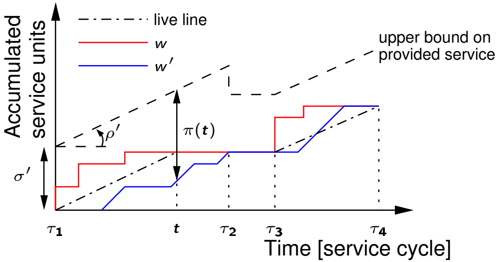
- w - the requested service curve
- w' - the provided service curve
- backlog - $q(t) = w(t) - w'(t)$
- live line - $\rho'_r \cdot (t - \tau_1 + 1)$
- a requestor is live if w_r is above the live line.



Replenishment Policy

Active period

- is the maximum interval of time a requestor is backlogged and/or live.



Potential ($\pi_r(t)$)

- is the amount of budget a requestor has.

Service guarantee

- How do we compute the minimum guaranteed service for any active period?
  - by considering the active period with the maximum interference

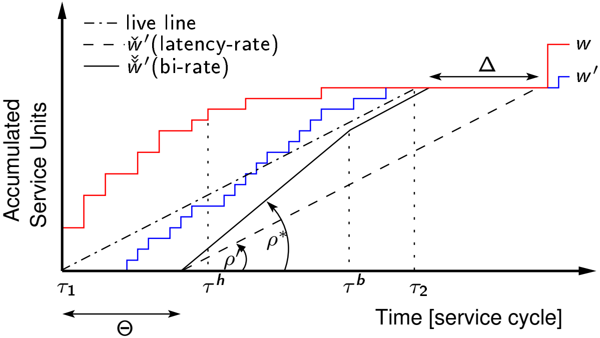
$\mathcal{LR}$ Service Guarantee

- For every active period $[\tau_1, \tau_2]$, it guarantees a minimum service at a rate of ρ', after a maximum latency, Θ.

- However, it does not take into account bursty provided service; hence, it gives pessimistic WCRT.

Bi-rate Service Guarantee

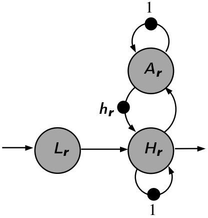
- A higher rate (ρ^*) interval, followed by the regular service rate (ρ').
- The higher rate, $\rho_r^* = 1 - \sum_{\forall s \in R_r^+} \rho_s'$.



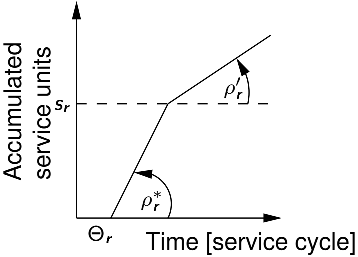
Dataflow Model

Dataflow Model of Bi-rate Service Guarantee

- For n requested service units within an active period,
- s of them are served at the higher rate, and
- $n - s$ of them are served at the regular allocated rate.



CCSP dataflow model



Rate of provided service

Experiment Setup

Given

- a CCSP arbitrated SRAM memory controller,
- a video decoder application running on a GPP of a given frequency

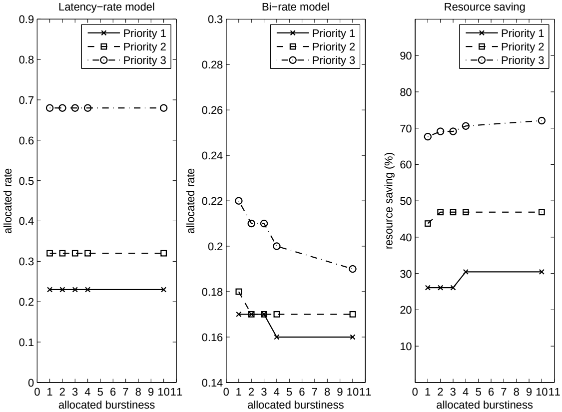
We need to find

- the arbiter configuration that satisfies a throughput requirement,
    - both according to the latency-rate and the bi-rate service guarantees

Experimental Result

Resource utilization

- E.g. At priority 3, where $\sigma = 1$, ρ drops from 0.68 to 0.22. This is a memory bandwith saving of 67%.



Conclusions

- Pessimistic service guarantees lead to resource over-allocation.
- A tight service guarantee enables
  - a given MPSoC resource to support more requestors, or
  - a given set of requestors to be accommodated with less resource.
- We present a tight service guarantee for CCSP
  - a piece-wise linear guarantee for accurately capturing the provided service
  - a dataflow model for real-time timing analysis at system-level
- Experiments show memory bandwidth savings from 26% - 67%.

Thank you! Questions?