A General Framework for Average-Case Performance Analysis of Shared Resources

Sahar Foroutan[1], Benny Akesson[2], Kees Goossens[2], and Frederic Petrot[1]
[1]TIMA Laboratory, Grenoble, France
[2]Eindhoven University of Technology, Eindhoven, the Netherlands
{sahar.foroutan, frederic.petrot}@imag.fr
{k.b.akesson, k.g.w.goossens}@tue.nl

EUROMICRO DSD/SEAA
SANTANDER, SPAIN SEPTEMBER 4-6, 2013

Soft Real-Time Requirements in Shared Memory MPSoCs

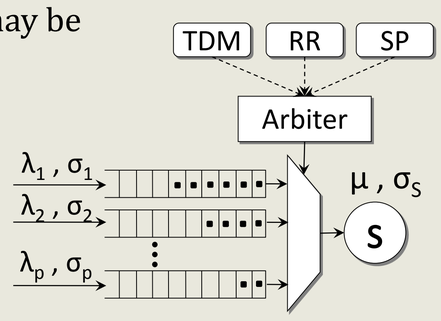
- Embedded systems are growing in complexity as more and more applications are integrated into MPSoCs.
- Applications share resources (e.g. processors , interconnect, and memories) to reduce cost, resulting in temporal interference.
- Some applications, e.g. video decoders, have **soft real-time (SRT) requirements**, where deadlines must be satisfied with high probability.
- Cost-efficient performance analysis requires the average execution time of the SRT applications to be accurately estimated.

Average Performance Analysis

- To determine the average execution time of application:
  - Average latency of shared resources is required.
  - Probabilistic models, e.g. queuing theory, can be used.
- Any performance analysis needs to:
  1. Model shared resources (i.e. arbitration mechanism)
  2. Characterize the application (i.e. traffic)
- Existing queuing models:
  - Are either based on an exponential arrival traffic or address only a single arbitration mechanism, thus failing to recognize the diversity of arbiters in complex systems.

1) Resource Modeling

In an MPSoC, access to the shared resources may be provided by different arbiters, e.g. :
- Time-Division Multiplexing (**TDM**),
- Static-Priority (**SP**), or
- Round-Robin (**RR**)

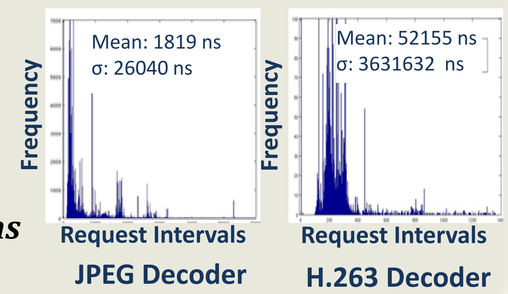


Contribution 1:
A high-level model for resource sharing that can be used with different arbiters.

2) Application Characterization

Models assuming exponentially distributed traffic do not cover dynamic applications executing on MPSoCs.

E.g. histograms of request intervals of JPEG and H.263 clearly show a non-exponential distribution.



Contribution 2:
Our high-level model is based on general distributions of request interval and service times (i.e. G/G/1).

General Resource Model Based on Queuing Theory

• The analytical model aims to estimate the average waiting time a request arriving to a multiple-queue resource with ***p*** queues, spends in any queue ***i***. This is called the ***queuing delay (W_i)*** :

$$W_i = \underbrace{n_i T_{S_i}}_{(I)} + \underbrace{\sum_{j=1,\ j\neq i}^{p} T_{S_j} \,.\, A(n_i, n_j)}_{(II)} + \underbrace{R}_{(III)}$$

(I) is the waiting time due to requests in the same queue, depends on:
- n_i: average number of requests already waiting in queue ***i***
- Ts_i: Average service time of queue ***i***

(II) represents the interference from other queues. At the arrival of the considered request to queue ***i,*** there is a number of requests (n_j) in any other queue ***j*** that depending on arbitration may be served earlier.

- $A(n_i,n_j)$=**Arbitration Indicator**: determines the interference of other queues on queue ***i*** and thus models the arbitration.

(III) R is the average residual times (remaining service times) of all queues in multiple G/G/1 models.

Arbitration Indicators for TDM, SP, and, RR

$$A(n_i, n_j)^{TDM} = n_i \quad for\ 1 \le j \le p$$

$$A(n_i, n_j)^{SP} = \begin{cases} n_j, & \text{if } \boldsymbol{j} \text{ has a higher priority than } \boldsymbol{i} \\ 0, & otherwise \end{cases}$$

$$A(n_i, n_j)^{RR} = Min\,(n_i, n_j) \quad \text{for all } \boldsymbol{j}$$

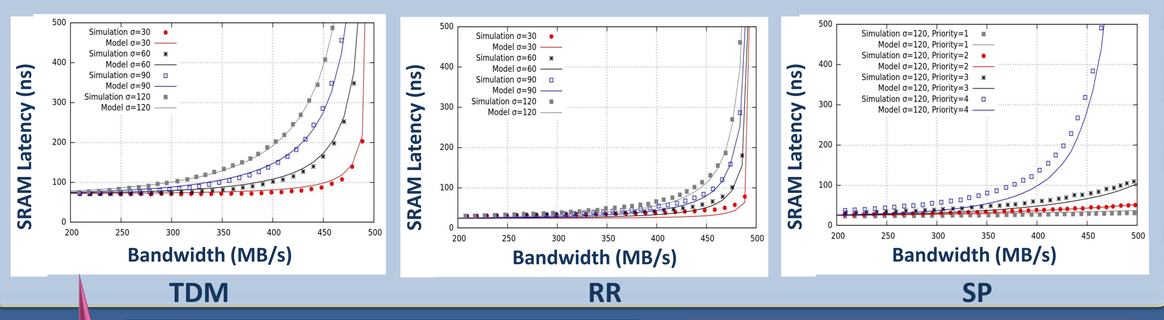
Analytical vs. Simulation Performance Results

Simulation Platform

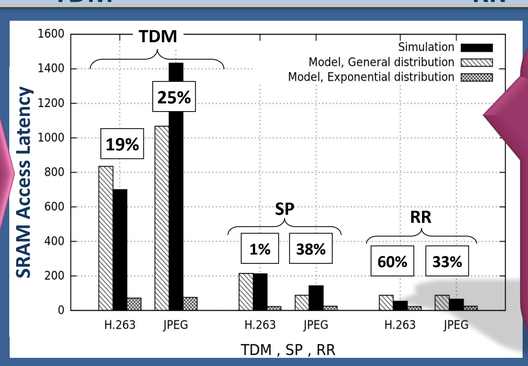
- A cycle-accurate SystemC model of a real-time memory controller, supporting a variety of memories and arbiters.
- Experiments consider a 32-bit **SRAM** with a peak bandwidth of 2 GB/s.
- **Four** requestors share the bandwidth of the memory.
- Traffic (read and write requests toward the SRAM) is injected by traffic generators issuing either **synthetic** traffic or **real application traces**.
- **Request size** is set to 64 B (16 words), resulting in uniform service times of 16 cycles for both reads and writes.

Synthetic Traffic Results

- Average memory latency (queuing + architectural delays), plotted against bandwidth.
- Synthetic traffic: Request intervals follow a normal distribution with:
  - **Mean** request intervals = 1/bandwidth
  - **Standard deviation** ***σ=30, 60, 90, 120 ns*** for TDM and RR, and ***σ=120*** for SP
- For **SP**, 4 requestors with different priority levels.
- Analytical vs. simulation **average error**: 4.1% for TDM, 12.9% for SP, and 14.2% for RR.



Real Application Results



- H.263 & JPEC decoders
- More significant error compared to synthetic results
  - Applications have very bursty traffic that cannot be precisely captured by G/G/1 models

However, our model provides more accurate results than models assuming **exponential** arrivals!