Requirements for flexible scheduling in complex embedded systems

Real Time Systems Lab
(ReTiS)

Scuola Superiore di Studi Universitari e Perfezionamento Sant'Anna
(SSSA)

Requirements overview

Resource level
- Per-application requirements
- Global resource mangement requirements

Application level
- Multi-resource management requirements

User level (mostly qualitative)

Design support requirements
- Composability/integration
  - Possibility to design subcomponents in isolation
  - Guarantees should remain in integrated system

Requirements (resource level)

Resource level requirements

- Related to how the system
  - manages and schedules resources at a low level
  - allows an application to specify its allocation requirements
  - accepts or denies start of new applications, so to have the guarantee to respect requirements
  - modulates a resource allocation of an application
    - considering the measured/expected actual resource occupation
    - considering requests made by other applications as well, in “overload” conditions
    - considering the constraints of the platform (energy)
  - Coordinates with application for multi-level adaptation

Resource level scenarios 1/2

Multimedia application

- expects an allocation at least able to sustain a given data rate (MPEG DVD/HDTV, mp3 192 kbps, etc…)
  - CBR versus VBR applications (highly variable loads)
  - Don’t want to tie system design to WCET, but rather consider average requirements (soft, non-guaranteed RT)
- within given latency tolerance thresholds
- within given jitter tolerance thresholds (buf mgmt)
- may be able to adapt to varying “external” conditions

Control application

- expects an allocation for a given data rate as well
  - need for guaranteed allocations
  - need for upper bounded latency
- may be able to adapt as well (decreasing sensor sampling periods, speed of actuators, etc…)

Resource level scenarios 2/2

Service provider with guarantees

- customer paid for at least a min allocation of a percentage of the available resources
- reclamation of spare allocation may or may not be done (upper bounded allocations)
- multimedia/interactive case: system may adapt allocation to real usage
- batch case: application "backlogs" resource, so guarantee the minimum and use a global compression (or expansion) policy

Timing range of requirements

Short-term vs long-term requirements

- application may expect it within a given deadline, or at known (periodical) intervals
- application may expect the promised allocation in the average (at infinity)
- application may not have expectations on when and how its request will actually be satisfied

How it could work

Application specifies

- minimum required allocation
- preferred management of optionally available additional resource time (i.e. if adaptive or not, what kind of adaptation)
- its ability to decrease requests and QoS levels for dealing with
  - temporary overload situations
  - dynamically reconfigurable energy-aware constraints

System enforces

- per-application/user/group max allocations
- saturation of exceeding requests
- priority-based or weight-based reduction of preferred requests in case of unsatisfiability of all of them
- global optimisation policies for trading off QoS sacrifications
  - for facing with a temporary overload
  - for saving battery, if requested

Requirements

Application level (AL) requirements

- Related to how the system
  - manages AL requirements through
    - a mapping to resource level allocations (through probes, benchmarks, etc…)
    - a multi-resource evolution/consumption model (someway provided by the application)
  - allows specification of AL QoS requirements
    - by specifying constraints on application-specific metrics
  - performs admission control and provides guarantees based on such knowledge(s)
  - coordinates application-level adaptations, resource-level adaptations, multi-resource cross-interrelations by exploiting knowledge of application data flow / structure

QoS Metrics

Starting point is defining QoS metrics

- Measurable significant quantities dealing with experienced QoS by the user

RL QoS metrics

- Scheduling error (latency, jitter)
- Statistics on deadline misses

AL QoS metrics

- Multimedia: video/audio related parameters (also mixed, e.g. AV delay)
- Control: ?

AL Requirements

AL Requirements

- Expressed in terms of metrics values
  - Min, max, avg values
  - Timing of metrics values
- Application-specificity
- Identification of classes of applications
- Provide modularity/pluggability of application-specific AL to RL mappings of contracts/metrics

Energy issues in ALCs

Flexible / multi-level ALCs (/RLCs)

- Not absolute values for min guarantees, but availability to adapt/reduce based on energy requirements (e.g. power unplugged)
- Or at least provide means for immediate renegotiation of parameters by reacting to events related to such "external" conditions
- Possibly use a homogeneous adaptation interface from the standpoint of application for
  - energy-aware adaptation
  - overload-aware adaptation