TIME AND SPACE PARTITIONING COMPUTATIONAL MODEL

A. Crespo1, M. Masmano1, J. Coronel1, P. Balbastre1, M.H. Deredempt, A. Rossignol2, and J. Windsor3

1Instituto de Informática Industrial, Universidad Politécnica de Valencia, Spain
2Astrium, France
3ESTEC, The Nederlands

ABSTRACT

This paper presents a computational model for partitioned systems. The computational model has as goal a particular way of conceptualizing computations and temporal constraints and predetermines the techniques to be used to analyse the schedulability of the system. The model is based on the entities used to specify the systems.

Key words: Partitioning systems, hypervisors, scheduling.

1. INTRODUCTION

In the last years, embedded systems have increased their processing capabilities and its computational resources in such a way that they are capable of executing several concurrent real-time applications that would formerly have required several embedded systems. Scheduling several applications on a single processor not only reduces cost but assures better reliability, because the overall complexity of the system is reduced. The design of such systems can be solved by means of virtualization, where the processor is subdivided in multiple partitions, each of them with its own operating system and applications running on top. Switching between partitions is the role of a hypervisor. When the virtualization technique is used in real-time embedded systems, the performance plays the most important role.

The Aeronautic industry is using the Integrated Modular Avionics (IMA) approach based on the fundamentals of [Rus81, Rus99] allowing to manage higher level of software complexity and efficiency. This approach is supported by ARINC-653 standard [ari96] which defines the interfaces of a partitioned system. The extension of this approach to space domain [WH09, AM08], automotive [aut] and other is under development.

In the last years, the space domain has been promoting some solutions for partitioned systems based on different technologies as hypervisor based [MRC+09, AMGC09] or based on RTEMS air09 and based on the pikeOS micro-kernel [KW07]. In the framework of the project IMA-SP [IMA11a], a study of the computational model for partitioned systems has been carried out a set of studies to the analyse and propose different aspects of the called IMA-SP platform. One of these studies has been the definition of the computational model [IMA11c] taking into account a set of requirements defined in [IMA11b].

From the computational model, one of the main aspects is the analysis of the global system. The analysis of the system involves the fulfillment of the basic properties: temporal and spatial isolation and the schedulability analysis of the temporal constraints. In order to perform a system analysis there is needed the definition of a computational model that copes the definition of the system and permits the system evaluation.

This project assumes the IMA approach and the ARINC-653 specication [ari96] as basis for the IMA-SP. ARINC-653 standard defines two clearly separated scheduling spaces: the partition scheduling, where the scheduled elements are the partitions, and the Application/Executive (APEX) software interface between the operating system and the application software, where the processes are the scheduled entities. ARINC 653 defines a cyclic scheduling for the global scheduler and a preemptive fixed priority policy for the local scheduler.

In this paper, we present a synthesis of the work carried out in the computational model study. In this work, a proposal of a computational model for partitioned systems describing the entities involved in the model and the expected outputs is presented.

The paper is organised as follows, next section present a summary of the scheduling techniques. Section 3 states some system requirements that permit to define in section 4 the computational model. Finally, some conclusions are presented.

2. SCHEDULING TECHNIQUES FOR PARTITIONED SYSTEMS

Partitioned systems are software architectures that define two software layers: a partitioning kernel in charge of partition scheduling and partitions that can include internal task scheduling. These two levels of scheduling in a
system are known as hierarchical scheduling.

In a hierarchical system, a virtualisation layer is in charge of partition scheduling, also called the global scheduler. On the second level, the local scheduler schedules partitions' tasks.

In recent years, considerable effort has been made in order to provide exact schedulability analysis to hierarchical systems. Specifically, for fixed-priority hierarchical systems (where fixed-priority scheduling policy is used in both global and local schedulers) the work by Davis and Burns [DB05a] provides an exact worst case response time analysis.

A generic real-time system is composed of a number interdependent tasks working together to meet some common purpose. It can be viewed as a collection of tasks, each one responsible for a specific portion of the system requirements. The computational model or task model defines the set of tasks, resources and temporal constraints of the system. The periodic task model has been traditionally used to express timing requirements in real-time systems and it has been the basis of real-time scheduling theory. This model has shown to be practical and rigorous. It properly represents the fundamental timing behaviour of the real world, and also, it has been shown to be a formal model where rigorous theoretic analysis can be done.

Real-time scheduling policies can be categorised in three classes:

- **Static table-driven approaches:** The system is analysed off-line and a static schedule (or table, as it is usually called) is generated, which is used at run-time to decide when a task must be executed. The length of the table is defined as the MAF (Major time frame) and it is the least common multiple of task periods. These techniques are used in time triggered systems, and cyclic schedulers.

  The problem of generation of a static table has been studied and it is equivalent to a bin packing problem. This problem is a NP-complete [GJ79] which means that no polynomial-time solutions are known. In the case of generation of a cyclic scheduler, several authors propose different techniques like heuristic search, simulated annealing or genetic algorithms [BHR95].

- **Fixed priority preemptive approaches:** These family of schedulers assigns a fixed priority to each task, at run-time tasks are scheduled according to the priority. The most representative policies are RM (Rate Monotonic) and DM (Deadline Monotonic) where resources, jitter control, O.S. overheads, etc. can be taken into account in the schedulability analysis.

- **Dynamic planning-based approaches:** These perform also an off-line schedulability analysis. At run-time, the priority of the tasks is calculated dynamically. The EDF (Earliest Deadline First) and LLF (Least Laxity First) are the dynamic scheduling policies par excellence. Although not directly used a the system scheduler, the EDF is widely used as a reference to build table driven schedules [BHR95] [GP08].

### 2.1. Scheduling techniques for partitioned systems

The main characteristics of the partitioned systems scheduling are:

- The scheduling unit is a partition.
- Partitions have no priority.
- The scheduling algorithm is predetermined, repetitive with a fixed periodicity, and is configurable by the system integrator only. At least one partition window is allocated to each partition during each cycle.

Within the IMA concept, a partition comprises one or more processes or tasks that provide the functions associated with that partition. According to the characteristics associated with the partition, the tasks may operate concurrently in order to achieve their functional and real-time requirements.

To achieve a hierarchical scheduling several strategies can be used:

- **Server-based schedulers:** A server is a periodic tasks, commonly defined with the tuple (budget, period), that allocates budget units of time every period to the server.

  The implementation of most servers except the periodic (and the background server), require the modification of the global scheduling policy to maintain the budget of the servers.

  Exact schedulability tests for fixed priority systems (both the global and the local scheduler are fixed priority) for the periodic, sporadic and deferrable servers where presented in [DB05b] (a correction to this work was presented in [BRC09]).

  The improvement of the servers and a better understanding of the isolation properties of this mechanisms refocused the application of these servers to what was called bandwidth servers or resource reservation protocols, which allows real-time tasks to execute in a dynamic environment under a temporal protection mechanism, so that each server will never exceed a predefined bandwidth, independently of its actual workload requests.

- **Compositional approach** One of the goals of the compositional scheduling model is to avoid performing a global schedulability analysis that considers the timing requirements of all the tasks in all the partitions (or task group in the original approach). Ideally, each partition or task group can be analysed by itself for schedulability in an independent way. Once all partitions have been analysed, the final schedule is a composition or integration of each one guaranteeing the correct temporal operation of the composed system.
The work presented by Insik Shin et al. [SL08] propose a compositional real-time scheduling framework where global (system level) timing properties are established by composing together independently analyzed local (component-level) timing properties.

All the workload executed by a subsystem (which is assumed to be periodic) is modeled as a single execution task. Basically they use the demand bound function $dbf_\tau(t)$ (where $\tau$ represents the subsystem task set) function as defined in [IAA96]. This function represents the amount of computation time that has been requested by all the activations whose deadline is less than or equal to the function’s argument.

The subsystem can be modeled as a single periodic task (interface task, $\Phi$) such that the demand bound function of this interface task $dbf_\Phi(t)$ is an upper bound of $dbf_\tau(t)$. This task acts as a periodic server. If the periodic server is described as a budget, deadline and period it is called an “Explicit Deadline Periodic” (EDP) resource server.

In [AIOS09] the compositional scheduling is adapted to the ARINC model.

Flat model In many cases is difficult to hide the internal tasks of partitions because of the need of specify execution flows conducted by IO operations that involve tasks of different partitions.

Flat model approach considers all tasks, independently of the partition, as a global system. Then it is supposed that a single global scheduler is in charge of managing all the tasks, and conduct the corresponding schedulability analysis.

The last step is to adapt the solution back to the partitioned system by grouping (trying to put together) the tasks of each partition in order to reduce the number of partition context switches.

Comparing the previous models, a list of advantages and drawbacks can be extracted:

- Compositional approach permits to abstract partitions and hide internals details which is in the line of isolation principles. However, the interface generated for each partition (bandwitch) is an upper bound of the partition needs that can be a very pessimistic result. Additionally, inter-partition communications and shared resources may difficult the analysis and the implementation.

- Flat model is based in a consolidated theory and permits to specify in detail the execution sequences which is very appropriated to deal with IO operations and tasks involving hard real-time constraints. The final schedule can be optimise taking into account the partition execution. On the other hand, the number of tasks can be very high with the analysis complexity. If an optimized solution is desired, a deep knowledge of the timing attributes of all the tasks is needed to do the scheduling analysis. As final comment, there is not a clean separation of concerns between partition developers and system integrator, or even among partition developers.

3. SYSTEM REQUIREMENTS

IMA-SP assumes the ARINC-653 [ari96] approach which defines a partitioning scheme, a set of interfaces and a scheduling policy. For the computational model, two basic initial assumptions are considered: the partitioning scheme involving the static resource allocation to partitions, and the cyclic scheduling policy for the virtualisation layer.

In the IMA-SP requirements [IMA11b] it is defined the allocation of computing resources to software functions. From these requirements the following conclusions can be extracted:

- Applications can be associated to partitions. Activities or functions (the same semantic is assumed) are internal to applications.
- Worst case execution time of the activities assumes the specific use of resources as FPU and cost of the memory access.
- Activities are periodic. Aperiodic activities should be transformed in periodic due to the underlaying scheduling policy (cyclic scheduling)
- CPU budget can be expressed as a maximum computation every period. A partition interested in a CPU bandwidth should specify the maximum time using the CPU in an interval (period).
- An entity to model the function-to-function communication and synchronisation as a sequence of activities involving different partitions should be considered
- The communication between partitions, derived from the previous conclusion, should be modeled. (ARINC-653 defines sampling and queuing channels). Other mechanisms, as shared memory, are not included in the standard.
- Shared memory between applications is required. It involves the need of shared resources (memory area) and mechanisms to prevent their use in a safe way
- Cache memory should be managed and propagated to the partition in order to produce more exact worst case execution time analysis.

4. COMPUTATIONAL MODEL

In general, a computational model can be understood as a definition of the entities and their relationship in order to study and analyse the system. In classical real-time systems, the entities use to be "tasks" and "shared resources" which define the computational needs in terms of CPU and the temporal constraints. The analysis involves the decision about the schedulability of the system. In this chapter, a proposal of a computational model
for partitioned systems describing the entities involved in
the model and the expected outputs is presented.

The assumption of a model means to subscribe a partic-
ular way of conceptualizing computations and temporal
constraints and predetermines the techniques to be used
to analyse the schedulability of the system.

In classical real-time systems, there exists a set of well
known scheduling policies with a large number of paper
describing scheduling techniques under different model
assumptions. For instance, assuming a fixed priority
scheduling, there is a technique (RMA) that permits to
determine the schedulability of the system. In this case,
the result of the analysis is a boolean answer: the system
is /is not schedulable. The final schedule is not provided
because of no heuristic decision have been taken in the
schedulability analysis.

In the IMA-SP approach, the underlying scheduling at
SEP level is a cyclic scheduler (ARINC-653 scheduling
policy). It means that final schedule is a sequence of tem-
poral windows or time slots where the resources (CPU)
are allocated to a specific entity. The problem of ad-
justing a set of computations into a collection of tem-
poral slots has been studied by several authors and it is
a NP-complete [GJ79]. NP-complete problems are a set
of equivalent problems for which no polynomial-time so-
lutions are known. In the case of generation of a cyclic
scheduler, different authors propose different techniques
like heuristic search, simulated annealing or genetic al-
gorithms.

This imply that the result of a system analysis can not be
schedulable or not schedulable due to the schedule is not
a consequence of a deterministic algorithm. The result
has to include the schedule plan considered. In this case,
the result should be "the system is schedulable under the
achieved schedule".

The computational model defines the different entities
involving computation and their relationship in order to
study and analyse the system. These entities are:

- System: models the global system.
- Partition: models the partition as a container of com-
  putation units
- Task: is the basic unit of computation. A task is allo-
  cated to partition.
- End to end flow (ETEF): permits to specify a se-
  quence of computations with temporal constraints.
- Channel: captures the communication needs be-
  tween partitions.
- Mutual exclusion resources (MER): models the
  shared resources used by partitions in a exclusive
  way.
- Overhead parameters: models the overhead intro-
  duced by the computation activities.
- Slot: temporal window where the resources are allo-
  cated to a partition
- Scheduling plan: slot sequence to be executed in a
  MAF

A system can be described in terms of a tuple

\[ \Sigma = (\Theta, \Pi, \Xi, \Omega, \Delta, \Lambda) \]

where \( \Theta \) represents the partitioning kernel, \( \Pi \) a set of par-
titions, \( \Xi \) a set of End-to-End flows (ETEF), \( \Omega \) a the set
of channels to inter-partition communications, \( \Delta \) a set of
shared resources and \( \Lambda \) represents the global schedule.

The underlying platform (\( \Theta \)) provides the infrastruc-
ture to execute the partitions. From the computational
model, it can be defined in terms of

\[ \Theta = (CF, \phi, Ov) \]

where \( CF \) represents the static definition of the system
with the allowed data flows which will be used to guide
the system execution, \( \phi \) defines the scheduling policy
(cyclic policy), and \( Ov \) defines the platform overheads.

\( \Pi \) is a set of partitions which are defined in terms of:

\[ P_i = (\Gamma_i, \rho_i) \]

where \( \Gamma \) is the set of tasks or processes in ARINC-653
terminology, and \( \rho \) the list of ports available for this par-
tition. Each task \( (\tau_i) \) in the partition is characterised by
\( \tau_i = (WCET_i, LD_i, Ri) \), being \( WCET \) the worst case
execution time of the task, \( LD \) the local deadline and \( Ri \)
the set of shared resources required \( (Ri \in \Delta) \).

Inter-partition communication channels (\( \Omega \)) defines the
way of connecting partitions through ports. There are two
types of channels, as defined in ARINC-653: sampling
channels and queuing channels. A channel is charac-
terised by \( Ch\i = (ChType, ChIn, ChOut, Att) \). The channel
type \( (ChType) \) is sampling or queuing, \( ChIn \) and \( ChOut \) are
determined by a pair \( < P_i, \rho > \), and \( Att \) permits to specify specific attributes of the channel.
The cardinality of this set has to be \( \geq 1 \). In the case of
queuing channels the cardinality is 1.

\( \Xi \) defines a set of End-to-End flows. Each ETEF denes
basically a sequence of tasks involved in a computation
that have to be executed under some glogal timing con-
straints. Each element of this sequence \( (\xi) \) is defined as:

\[ \xi_i = (Comp_i, Per_i, GD_i, Off_i, Prec_i) \]

where \( Comp_i \) is a pair \( < P_i, \tau_i > \), \( Per_i \) is the period
of the computation, \( GD_i \) is the relative global deadline and
\( Prec_i \) the precedence relation between \( Comp \).

Shared resources \( \Delta \) models the resources used by several
partitions that are defined in terms of synchronisation.

The goal of the analysis is to determine the system fea-
sibility which in cyclic scheduling implies the generation
of a schedule that fulfils the temporal requirements. In
the model, \( \Lambda \) represents the final schedule and is defined
in terms of a scheduling plan. A scheduling plan is a se-
quence of execution slots defined as: \( \Lambda = (MAF, S) \)

\[ s_i = (P_i, Org_i, Dur_i) \]

where \( P_i \) is a partition, \( Org_i \) is the relative distance from
the MAF origin.
4.1. Modeling partitions

The proposed model permits to consider different partitions requirements for the system architect.

- Partitions with low level detail. This corresponds to partitions that the system architect does not know the internals (or these details are not relevant) and the goal is to provide a bandwidth of the system resources to the partition.

This case can be modeled defining a unique task in the partition and an ETEF. The task defines the WCET that is understood as "maximum computation time" for this task. The ETEF includes this task with a Period. The bandwidth of this partition will be the relation between the WCET and the Period.

- Partitions with high level detail. The system architect knows the internal task of one or several partitions and is interested in schedule sequences of tasks with precedence relations. In this case, partitions have several tasks and the system architect defines several ETEF’s to cope these relations.

When all partitions are defined with one task and one ETEF, it matches the compositional scheduling model. When all partitions are defined in detail, the approach is the flat model.

5. CONCLUSIONS

This paper presents a computational model for partitioned real-time systems in the scope of the project IMA for Space. The proposed model is based on the needs expressed in the Requirement Baseline document [IMA11b] and the study and analysis of the state of art of real-time scheduling and tools for configuration and scheduling partitioned systems.

The proposed model tries to capture the entities and their characteristics involved in the computation to be performed by the different activities under temporal constraints.

The computational model is based on a set of entities as: platform, partitions, channels, ETEFs, etc, that permit to specify the global system. The goal of the computational model is the analysis of basic properties as spatial and temporal isolation and the system schedulability.

Partitioned based scheduling tools is one of the weak areas in the integration of partitioned systems with real-time contraints. The improvement of the scheduling tools to optimise the scheduling plan in another important activity to be done.

REFERENCES


[IMA11c] IMA-SP-D03 - Time and space partitioning computational model, January 2011. IMA-SP Integrated Modular Avionics for Space.


