Virtual Execution Platforms for Mixed Time-Criticality Applications

Martijn Koedam Gabriela Breaban Karthik Chandrasekar[1]
Manil Dev Gomony Sven Goossens Rachana Kumar Yonghui Li
Davit Mirzoyan[1] Andrew Nelson Shubhendu Sinha Benny Akesson[2]
Kees Goossens

Eindhoven University of Technology; [1] Delft University of Technology; [2] Czech Technical University in Prague

System-on-Chip (SOC) design gets increasingly complex, as a growing number of applications are integrated in such systems. These applications have mixed time-criticality, i.e., some have firm-, some soft-, and others non-real-time requirements. Executing such a mix of applications on a SOC poses several challenges.

First, to reduce cost, platform resources, e.g., processors, interconnect, memories, are shared between applications. However, sharing causes interference between applications, making their behaviors interdependent. This results in two problems for SOC design and verification: 1) accurate system-level simulation and several approaches to formal verification are infeasible, because of the explosion in the number of possible combinations of applications, inputs, and resource states and 2) verification becomes a circular process that must be repeated if an application is added, removed, or modified, making integration and verification dominant parts of SOC development, in terms of time and money.

The CompSOC platform addresses these problems by *executing each application on an independent virtual execution platform* (VEP). The VEPs are *composable*, i.e., cannot affect each other's behaviors. In the temporal domain an applications actual execution never varies by even a single clock cycle. Similarly, the energy and power behaviors of applications are also composable. As a result, applications can be designed, developed, verified, and executed in isolation.

The VEPs are also *predictable*, meaning that all interference is bounded. This makes them virtualized also in terms of performance bounds, which enables firm real-time applications to be verified using formal performance analysis frameworks.

The CompSOC platform uses the CoMiK *microkernel* to implement virtual processors on each processor time through temporal partitioning. Each application can use its own operating system (e.g. Compose, μcOS-III) and model of computation (e.g. CSDF, KPN, TT) in its VEP, to suit its level of time criticality.

As more applications are integrated on a single SOC, the need arises for more dynamic behaviour. The system should be able to start, modify and stop applications at run time without affecting running applications. For this purpose the CompSOC platform has been extended with a predictable and composable *resource management framework*. It manages *application bundles* that contain 1) an application in the form of executables (ELFs on multiple processors), and also 2) the specifications of the (one or more) particular VEPs that the application executes in, consisting of virtual processors, NOC connections, virtualised memories, etc. At run time, the resource management framework can dynamically load and start application bundles by creating a VEP and then loading, booting, and executing an application within it. VEPs can also be modified, stopped, and deleted at run time.

Our University Booth will present virtual-execution-platform and application-bundle concepts using an interactive demonstrator. It will show that the CompSOC has been extended with dynamic functionality, without sacrificing its key strengths: composability and predictability. We will demonstrate this through the use of the resource management framework and application bundles, showing that we can create, modify and delete virtual execution platforms running a mixed time-criticality application dynamically at run-time.

This work was partially funded by projects EU FP7 288008 T-CREST and 288248 Flextiles, Catrene CA104 Cobra, CA505 BENEFIC, CA703 OpenES, ARTEMIS-2013-1 621429 EMC2 and 621353 DEWI, NL STW 10346 NEST, and Czech Ministry of Education CZ.1.07/2.3.00/30.0034.