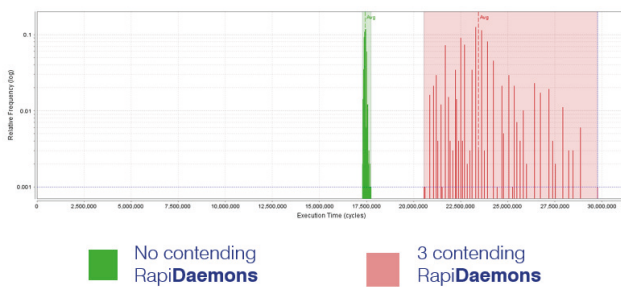


# Interference channel analysis support for DO-178 and CAST-32A with RapiDaemons

## RapiDaemons

### How can RapiDaemons help you?

RapiDaemons allow you to provide the evidence to identify interference channels and quantify interference effects for multicore systems.



### RapiDaemons use cases

- Platform characterization and comparison.
- Software characterization and optimization.
- Interference channel characterization and quantification.
- Automatically discover worst-case interference profile.
- Simulate the interference of your application.
- Produce evidence for DO-178 (CAST-32A).
- Robustness and sensitivity analysis.
- Verification of Performance Monitoring Counters.
- Verification of hypotheses on multicore behavior, for example effectiveness of cache partitioning and fairness of bus arbitration policies.

### Benefits of using RapiDaemons

- Reduce the cost and effort of analyzing multicore hardware for timing behavior, hardware characterization and selection.
- Understand the sensitivity of your application to interference when running in a multicore environment.
- Path to multicore certification for DO-178 and CAST-32A.

### How do RapiDaemons work?

RapiDaemons are small applications that run on multicore hardware and generate accesses on specific resources within the system. Each RapiDaemon targets a specific resource for example caches, interconnects, memory and other on-chip or off-chip shared resources, allowing you to identify and quantify the effects of multicore interference by running different RapiDaemons in different configurations.

RapiDaemon execution can be specified through tests created using RapiTest, allowing the automatic initialization and execution of RapiDaemons on your multicore platform. Meanwhile, RapiTime and RapiTask can be used to collect execution time results, values from Performance Monitoring Counters in your system, and scheduling information.

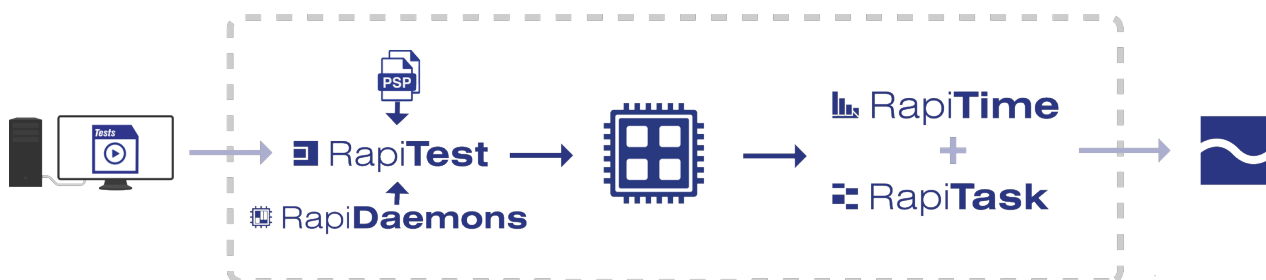


Figure 1. Using RapiDaemons within a multicore timing analysis workflow

## Types of RapiDaemons

There are different types of Rapi**Daemons**, which support the analysis of multicore interference in different ways:

- Standard Rapi**Daemons** either generate large load or are sensitive to load on a specific hardware resource. Standard Rapi**Daemons** target common interference channels, for example multi-level caches, interconnects and memory.
- Advanced Rapi**Daemons** either generate large load or are sensitive to load on a specific hardware resource with greater accuracy and precision than Standard Rapi**Daemons**. Advanced Rapi**Daemons** target complex resources and interference channels such as complex I/O devices, chip-specific devices and GPUs. Some Advanced Rapi**Daemons** support the analysis of complex sources of interference such as cache coherency protocols, thermal behavior and the effectiveness of cache partitioning mechanisms.
- Tuneable Rapi**Daemons** generate configurable load on a specific hardware resource and are customized to a specific multicore setup. As they are tuneable, these Rapi**Daemons** support fine-grained analysis of interference effects.

In addition to the Rapi**Daemons** listed above, special tools generate configurations of Rapi**Daemons** for specific use cases:

- The Discovery Rapi**Daemon** tool automatically generates a combination of Rapi**Daemons** that generate worst-case interference for your system.
- The Surrogate Rapi**Daemon** tool automatically generates a Surrogate Application, a combination of Rapi**Daemons** that mimics a specified interference profile, often one that is similar to the profile generated by your real application as it runs on your multicore system (see *Surrogate Applications*).

## How are RapiDaemons configured for my system?

As part of our *Target Integration Service*, we select appropriate Rapi**Daemons** for your system, port these to work with your system and perform additional configuration activities.

Through this service, we deliver a Platform Support Package that provides a tracing mechanism to extract results from Performance Monitoring Counters from your system and integration into the **RVS** analysis toolchain.

## Key features

### *Interference channel analysis*

- Interference generators for e.g.:
  - Memory bandwidth
    - Access direction: access direction, paging, memory partitioning, memory level parallelism
  - Buses/interconnects
    - Fairness assessment, contention analysis, routing impact
  - Caches
    - Different levels: L1, L2, L3..., inter-level inclusion, coherence, partitioning
  - Shared functional resources
    - Floating point units, GPUs, DSP
  - Thermal/control features
    - Dynamic DVFS behaviors, dynamic power/thermal caps
  - I/O devices
    - DMA, Ethernet
- Sensitive Rapi**Daemons**
- Tuneable resource accesses with Tuneable Rapi**Daemons**
- Identify the maximum interference for your system easily using Discovery Rapi**Daemons**
- Mimic the interference of your multicore applications with Surrogate Rapi**Daemons**
- Verify Performance Monitoring Counters
- Verifiable using Performance Monitoring Counters
- Integration with Rapi**Test**, Rapi**Time** and Rapi**Task**
- Documentation describing how to use Rapi**Daemons**

### *Supported platforms*

- See our website for a list of supported architectures, CPUs and RTOSs
- Support for other platforms is available on request

### *Licensing*

- Annual or perpetual licenses
- Licenses available for one or more configurations

## Surrogate Applications

Surrogate Applications are executable applications that mimic a specified interference profile, which is often a similar profile to that of a real application. Generated by the Surrogate Rapi**Daemon** tool, Surrogate Applications comprise a combination of Rapi**Daemons**. They support the analysis of multicore interference by allowing realistic interference scenarios to be analyzed without needing all applications in the system to be run during testing.

To generate Surrogate Applications, a profile (“fingerprint”) of a target application is first created by running that application and collecting metrics from Performance Monitoring and Debug Support Units on the hardware. This fingerprint represents the non-functional behavior of the application, including metrics such as cache hits, cache misses and memory accesses.

Next, the Surrogate Rapi**Daemon** tool is configured to match a specific interference profile when generating a Surrogate Application. This includes two things: specifying the non-functional metrics that must be matched, which can include any of those in the fingerprint for an application, and specifying a value for each of these metrics. The value for each metric is tuneable and can represent the value captured from the real application or can be increased to create a more “aggressive” Surrogate Application.

Finally, the Surrogate Rapi**Daemon** tool is run to generate a Surrogate Application that mimics the desired interference profile. The tool also produces a metric that represents the similarity between the desired fingerprint and that of the generated Surrogate Application.

**Rapita Systems Inc.**

41131 Vincenti Ct.  
Novi, MI 48375

Tel (USA):

**+1 248-957-9801**

**Rapita Systems Ltd.**

Atlas House, Osbaldwick Link Road  
York, YO10 3JB

Tel (UK/International):

**+44 (0)1904 413945**

Registered in England & Wales: 5011090