

RT.js: Practical Real-Time Scheduling for Web Applications

Stefan Naumann and Daniel Lohmann, Leibniz Universität Hannover

For billions of deployed browsers, JavaScript provides the platform-independent lingua franca that enabled the triumphal march of web-based applications. Originally intended for simple UI-event processing, JavaScript comes with an event-driven programming model, where event-callback functions are executed in strict sequential order. However, with applications getting more complex and tasks becoming more computation intensive, its first-come-first-served and run-to-completion semantic is hitting a limit, when reactions to user inputs are delayed beyond the human perception threshold. With the rise of the Internet of Things, this leads to friction-filled user experiences in everyday situations.

With RT.js, we selectively introduce pseudo-preemption points into JavaScript functions and sequence the execution of event callbacks with well-known real-time scheduling policies, like EDF. Thereby, we provide a soft real-time abstraction that mitigates the described short comings of the JavaScript execution model without modifying the actual engine; making RT.js compatible with billions of devices. Applied to generated real-time task sets, we can almost eliminate the 30-percent deadline-miss ratios of baseline JavaScript at moderate costs. In a browser-based macro benchmark, we could diminish the influence of computation-intensive background tasks on the page-rendering performance.