Analysis of real-time avionics systems using tracing and sampling data

François Rajotte

Laboratoire DORSAL
Département de génie informatique

POLYTECHNIQUE MONTRÉAL
AFFILIÉE À L'UNIVERSITÉ DE MONTRÉAL

Outline

- **Real-time and tracing characteristics**
- **Case study of a typical real-time problem**
- **Early trace analysis prototypes**
- **Advanced techniques**
- **Planning and next steps**

Real-time

- **Response time is bounded**
- **Characteristics**
  - **Determinism**
  - **Hard vs soft real-time**
- **Challenges**
  - **Timing bugs**
  - **Diffculty to debug**

Tracing

- **Logging low-level information about a program's execution**
- **LTTng**
  - **Low performance impact**
  - **Multi-core Scalability**
  - **Information logged :**
    - **Kernel events and system calls**
    - **Userspace tracepoints**

Exploring a trace

- **Challenges**
  - **Large amount of data**
    - **Filtering**
    - **Searching**
  - **Compromise between details and the big picture**

- **Visualisation**
  - **Organising the information**
    - **Use of domain-specific knowledge**
    - **Use of color**
  - **Drill-down approach**

Case study

- **Unexpected latency while running cyclictest**
- **Characteristics**
  - **Simple program**
  - **Reproducible**

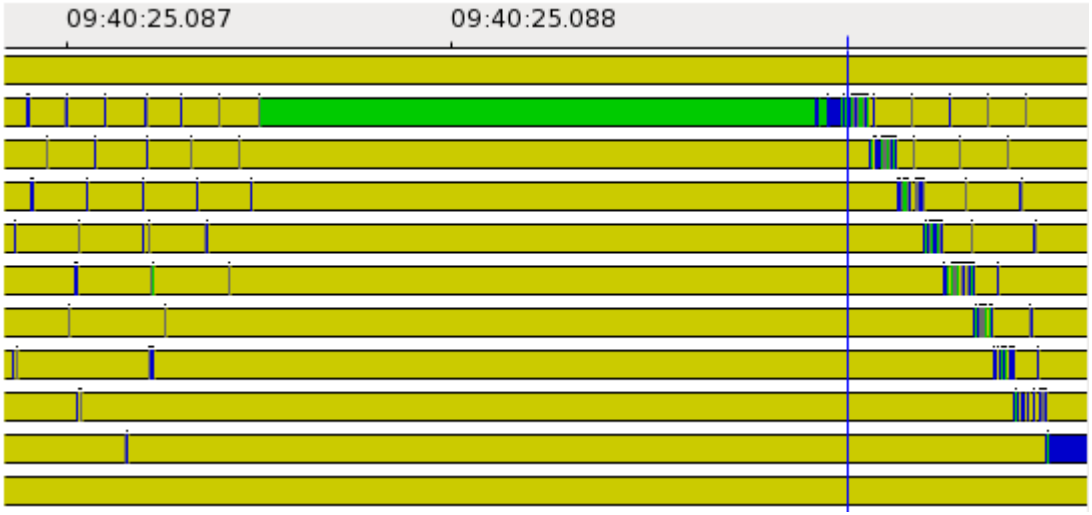
- **Using LTTng to locate the problem**

Case study - Steps

- **Harvesting data**
  - **Add tracepoint in cyclictest code that triggers when a deadline is missed**
  - **Start tracing + cyclictest**
  - **Execute latency-inducing action**
  - **Stop tracing**
- **Analysing data**
  - **Trace Monitoring Framework (TMF)**

Case study – Steps (cont.)

- **Search for missed-deadline event**
- **Findings :**
  - **Call to Nvidia driver**
  - **Privileged instruction to clear CPU cache**



Research questions

- **How can we use real-time system knowledge to find more rapidly these anomalous events ?**
  - **Periodic / aperiodic tasks**
  - **Focus on latency and task priorities**

- **Can we avoid using manually-inserted tracepoints ?**
  - **Tracepoints are statically defined for performance reasons**

Available tracing data

- **Events of interest**
  - **Scheduling events**
    - **sched_switch**
    - **sched_wakeup**
  - **System calls**
    - **sleep**
    - **Mutex operations**

- **Caveat : userspace mutex are “futexes”**
  - **Do not need to reach kernel space if no contention**

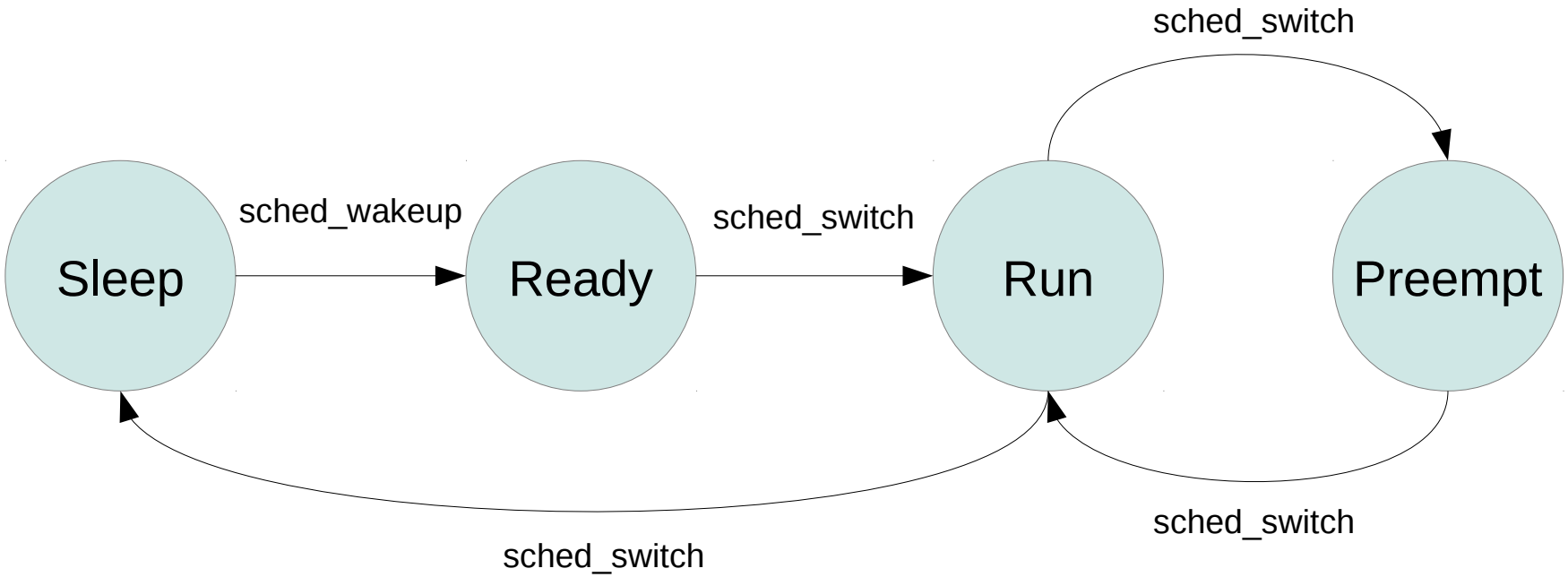
Gathering task statistics

- **Basic case : periodic task**
- **What we need :**
  - **Cycle start / end**
- **What we will use :**
  - **Scheduling events**
  - **Finite state machine**

Gathering task statistics – states description

- **States**
  - **Sleeping : task is sleeping until next cycle**
  - **Ready : task has woken up and wants to execute**
  - **Running : task is executing**
  - **Preempted : task has been preempted by a higher priority task**

Gathering task statistics – FSM description

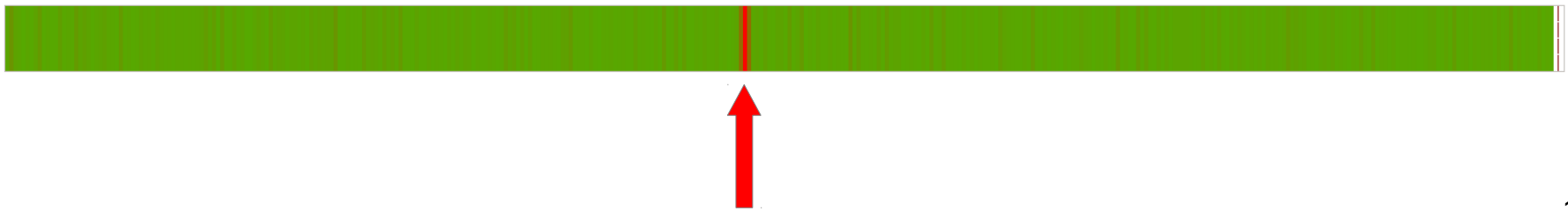


Gathering task statistics – results

- **Recoverable statistics**
  - **Per cycle**
    - **Interarrival time**
    - **Response time**
    - **Execution time**
  - **Globally**
    - **Period**
    - **CPU usage**
- **Using these statistics to enhance the visualization**
  - **Heatmap**

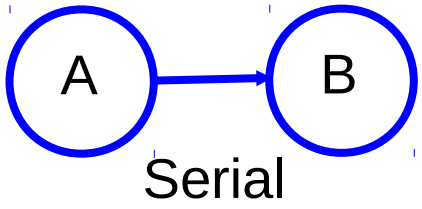
Heatmap

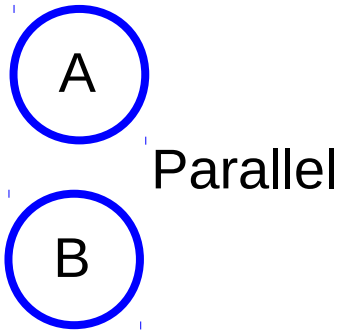
- **Goal : Finding abnormal statistics at a glance**
- **Problem : Not enough room to show all individual cycle statistics at a higher level.**
  - **Merge cycle statistics together**
- **Hypotheses**
  - **“Good” statistics are uninteresting**
  - **“Bad” statistics are interesting**
- **When merging, keep the worst statistic**
- **Use color gradient from green(good) to red (bad)**

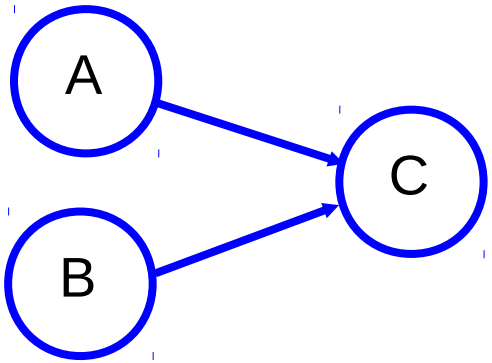


Data mining techniques

- **Finding frequent patterns**
  - **A priori algorithm**
    - **Transaction-based**
    - **Frequent patterns are formed from frequent subpatterns**
    - **Bottom-up approach**
  - **Winepi algorithm**
    - **Stream-based**
    - **Sliding-window approach**
    - **More complex episodes**







Applying these techniques to a trace

- **Segmenting the trace**
  - **Per process**
  - **Per cycle**
- **Variable-size windows to find short and long patterns**
  - **Time sensitive episodes**
- **Extract frequent patterns**
  - **Baseline signature / normal execution**
- **Cycles not exhibiting these patterns are potentially interesting and/or abnormal**

Planning

- **Short term**
  - **Creating an application to generate test traces**
    - **Mutex operations**
    - **Periodic and aperiodic tasks**
- **Long term**
  - **Implementing data mining techniques to trace analysis**
    - **Test them on generated test traces**
      - **Quality of results**
      - **Computational performance**

Conclusion

- **Recap**
  - **Case study showing the rationale of the research**
  - **Statistics harvesting using finite state machines**
  - **Future work using data mining approaches**

- **Questions ?**